



Universidade Federal de Juiz de Fora
Programa de Pós-Graduação em
Engenharia Elétrica

Mateus Hufnagel Maranhã de Faria

Estimação de energia no calorímetro hadrônico do ATLAS para o primeiro nível de
trigger utilizando redes neurais artificiais

Juiz de Fora
2017

Mateus Hufnagel Maranha de Faria

Estimação de energia no calorímetro hadrônico do ATLAS para o primeiro nível de *trigger* utilizando redes neurais artificiais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora.

Orientador: Prof. Dr. Luciano Manhães de Andrade Filho

Juiz de Fora
2017

Faria, Mateus Hufnagel Maranha de.

Estimação de energia no calorímetro hadrônico do ATLAS para o primeiro nível de *trigger* utilizando redes neurais artificiais/ Mateus Hufnagel Maranha de Faria. - 2017.

135 f. : il.

Orientador: Luciano Manhães de Andrade Filho
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Juiz de Fora, 2017

1. Estimação. 2. Calorimetria. 3. Rede Neural. 4. Processamento online. 5. FPGA. I. Andrade Filho, Luciano Manhães de, orient. II Dsc.

Mateus Hufnagel Maranhã de Faria

Estimação de energia no calorímetro hadrônico do ATLAS para o primeiro nível de *trigger* utilizando redes neurais artificiais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora.

Área de concentração: Sistemas Eletrônicos

Aprovada em

BANCA EXAMINADORA:

Prof. Dr. Luciano Manhães de Andrade Filho

Universidade Federal de Juiz de Fora, UFJF

Orientador

Prof. Dr. Augusto Santiago Cerqueira

Universidade Federal de Juiz de Fora, UFJF

Prof. Dr. Leandro Rorigues Manso Silva

Universidade Federal de Juiz de Fora, UFJF

Prof. Dr. Danton Diego Ferreira

Universidade Federal de Lavras, UFLA

AGRADECIMENTOS

Agradeço, em primeiro lugar, à minha família pelo apoio incondicional e incentivo em todas as etapas da minha vida acadêmica. Sem o suporte recebido por eles, esta conquista em hipótese alguma seria possível.

A todos meus amigos, que sempre estiveram ao meu lado contribuindo com correções, traduções e hospedagem nesse período de trabalho intensivo. Vocês foram fundamentais para que este trabalho se concretizasse em tempo hábil.

Ao meu orientador, Luciano, sempre paciente e dedicado, com o qual aprendi muito nesta jornada. Seus incentivos e a confiança depositada em mim foram fundamentais.

Aos colegas do LAPTEL, os quais formam uma grande família, sempre solícitos e dispostos a ajudar, independente das dificuldades enfrentadas.

A todos os professores do PPEE, pelos ensinamentos e apoio neste curto, porém intenso período de estudos.

À Universidade Federal de Juiz de Fora e à Faculdade de Engenharia Elétrica, que proporcionaram um ambiente propício para o florescimento deste trabalho.

Existem três classes de pessoas que são infelizes: a que não sabe e não pergunta, a que sabe e não ensina e a que ensina e não faz.

Siddhartha Gautama

RESUMO

Este trabalho tem como objetivo a implementação de um sistema de estimação de energia, adequado às condições de alta luminosidade do maior e mais potente colisor de partículas já construído, o LHC. Agendadas em seu programa de atualização, estas condições aumentam a probabilidade de colisões próton-próton dentro de seus detectores, como o ATLAS. A resposta do calorímetro hadrônico do detector ATLAS, Tilecal, necessita de um tempo de sete cruzamentos de feixe de partículas para gerar um pulso de amplitude proporcional à energia depositada. Neste sentido, aumentar a probabilidade de ocorrência das colisões também intensifica a presença do empilhamento de sinais. Este efeito faz com que o erro aumente na estimativa da amplitude dos sinais gerados na eletrônica de leitura do Tilecal neste ambiente. A interpretação do calorímetro como um canal de comunicação de característica dispersiva faz-se necessária para a recuperação do impulso de entrada, removendo-se, assim, o efeito de empilhamento e proporcionando uma medida direta da deposição de energia em cada cruzamento de feixe. A utilização de técnicas lineares para este propósito mostraram-se eficientes quando comparadas ao método atualmente implementado no sistema de *trigger*. No entanto, não-linearidades intrínsecas ao canal não podem ser modeladas por estas técnicas. Assim sendo, neste trabalho é proposto um equalizador de canal não-linear baseado em redes neurais artificiais. Através de treinamentos exaustivos, os principais parâmetros que compõem a arquitetura de uma rede neural *feedforward* foram determinados com o propósito de aliar alta performance e baixa complexidade da rede. O estimador neural mostrou-se robusto em ambientes com alta ocorrência de empilhamento de sinais, além de atender aos requisitos de implementação em *hardware* dedicado, no qual foi utilizado uma FPGA. Apesar da implementação de funções de ativação não-lineares em *hardware* com boa precisão consumir bastante recurso computacional, esta limitação foi contornada com a utilização de uma *look-up table* altamente compacta. Portanto, a rede neural foi quantizada em ponto fixo, visando reduzir ao máximo o número de elementos lógicos utilizados, sem comprometer sua capacidade ao compará-la com a rede implementada em *software*.

ABSTRACT

This work aims to implement an energy estimation system, which is adequate to the high luminosity conditions of the most powerful collider ever built, the LHC. Scheduled in its update program, these conditions increase the probability of proton-to-proton collisions within its detectors, like ATLAS. The response from the hadronic calorimeter of the ATLAS detector, Tilecal, needs time corresponding to seven crossing particle beams to generate pulse amplitude proportional to the deposited energy. In this sense, increasing the likelihood of collision occurrence also intensifies the signal pile-up. This effect increases the amplitude estimation error of the signals, generated in the readout electronics of Tilecal in this environment. The interpretation of the calorimeter as a communication channel of dispersive character is necessary for the recovery of the input impulse, removing, this way, the pile-up effect and providing a direct measure of the energy deposition in each bunch-crossing. The utilization of linear techniques for this purpose have shown to be efficient when compared to the method currently being employed in the trigger system. However, non-linearities intrinsic to the channel cannot be modulated through these techniques. Therefore, it is proposed in this work a non-linear channel equalizer based on artificial neural networks. Through exhaustive training, the main parameters that compose the architecture of a feedforward neural network have been determined with the purpose of allying high performance with low network complexity. The neural estimator has shown to be robust in environments with high amounts of signal pile-up, as well as meeting the requirements for implementation of dedicated hardware, in which an FPGA was used. In spite of the implementation of non-linear activation functions in optimal precision hardware requiring high computational resources, this limitation was circumvented by utilizing a highly compact look-up table. Therefore, the neural network was quantified at a fixed point, aiming to reduce as much as possible the number of logical elements used without compromising its capacity to compare it with the network implemented in software

LISTA DE ILUSTRAÇÕES

Figura 1	Os vinte e dois países membros do CERN (MELIS, 2016).	23
Figura 2	Visão geral dos experimentos do LHC e sua localização geográfica (TEAM, 1999).	25
Figura 3	Em (a) visão geral do detector ATLAS e seus sub-sistemas (ATLAS, 2016). Em (b) é possível ver, em corte transversal, como cada detector interage com cada tipo de partícula gerada nas colisões (PEQUENAO, 2008b).	28
Figura 4	Sistema de coordenadas da mão direita tridimensional do ATLAS (CERN, 2016e).	29
Figura 5	Detector de trajetória do ATLAS e seus sub-sistemas (CERN, 2016f).	30
Figura 6	Sistema de calorimetria do ATLAS (PEQUENAO, 2008a).	32
Figura 7	Esboço de um módulo do barril do calorímetro eletromagnético, mostrando suas três camadas e a granularidade nas coordenadas η e ϕ (NIKIFOROU, 2013).	33
Figura 8	Disposição dos conjuntos de telhas cintiladoras e aço no calorímetro hadrônico, e a aquisição do sinal feita através de fibras óticas, agrupadas pelas PMTs (AAD et al., 2008a).	33
Figura 9	Em (a) é representada a localização do barril e barril estendido do Tilecal no sentido positivo de η , e sua divisão em células (ATLAS, 2008).	

Em (b) é mostrada uma foto tirada durante a fase de construção do Tilecal. Também é possível visualizar sua divisão em células e as PMTs (GUIRAUD, 1999).	34
Figura 10 Pulso analógico típico do Tilecal digitalizado por um conjunto de 7 amostras, representadas por pontos ao longo da curva (PERALVA et al., 2013).	36
Figura 11 Esquema geral do sistema de <i>trigger</i> e DAQ durante a <i>Run-2</i> (NAKAHAMA, 2015).	37
Figura 12 Cadeia eletrônica atual do Tilecal (CARRIÓ et al., 2013).	41
Figura 13 Cadeia eletrônica do Tilecal na Fase-II (CARRIÓ et al., 2013).	43
Figura 14 Ilustração do efeito de empilhamento de sinais (<i>pile-up</i>) (KLIMEK, 2012).	43
Figura 15 Desenho esquemático de neurônios biológicos (HAGAN et al., 1996).	45
Figura 16 Neurônio de McCulloch e Pitts.	47
Figura 17 Modelo matemático de um neurônio (HAYKIN, 1999).	50
Figura 18 Funções de ativação típicas.	51
Figura 19 Rede <i>feedforward</i> de camada simples (HAYKIN, 1999).	53
Figura 20 Representação de uma rede <i>feedforward</i> multicamadas, possuindo duas camadas escondidas. Nota-se que a saída da segunda camada escondida é conectada como entrada da camada de saída da rede.	54

Figura 21	Exemplo de uma rede neural recorrente, possuindo apenas uma camada. Nota-se que a saída dos neurônios retornam às suas entradas através dos blocos de <i>delay</i> .	56
Figura 22	Aprendizado supervisionado.	57
Figura 23	Diagrama de fluxo de sinal para o neurônio j , adaptado de Haykin (1999).	61
Figura 24	<i>Overtraining</i> .	66
Figura 25	Conceito de um treinamento não-supervisionado.	68
Figura 26	Diagrama de blocos no domínio z que representa a modelagem do Tilleal como um sistema linear.	69
Figura 27	Sistema inverso utilizando filtro adaptativo.	71
Figura 28	Comparação entre os métodos propostos para a Fase-II e o atualmente implementado.	72
Figura 29	Filtro FIR em aplicação de tempo real com janela par.	75
Figura 30	Rede neural <i>feedforward</i> em aplicação de tempo real com janela par.	76
Figura 31	Desvio de fase do pulso característico.	77
Figura 32	Esquemas de detecção de sinal combinados com a estimação de energia.	80

Figura 33	Curva ROC (WOODS; BOWYER, 1997).	81
Figura 34	Influência da proporção do subconjunto de validação em relação ao conjunto de treinamento.	85
Figura 35	Histogramas para os conjuntos de inicializações.	86
Figura 36	Média e dispersão do erro RMS para cada conjunto dos 10 melhores resultados.	87
Figura 37	Dispersão do erro RMS para cada conjunto dos 10 melhores resultados para sinais com ocupâncias distintas.	88
Figura 38	Dispersão do erro RMS para cada conjunto dos 10 melhores resultados para sinais com ocupâncias distintas.	89
Figura 39	Dispersão do erro RMS para cada conjunto dos 10 melhores resultados para sinais com ocupâncias distintas.	90
Figura 40	Performance média para todos os cenários de teste.	92
Figura 41	Influência da ocupância do sinal de treinamento em diferentes cenários.	93
Figura 42	Comparação entre redes treinadas com diferentes conjuntos.	95
Figura 43	Performance média para todas as combinações possíveis dentro do espaço de busca $m \times h_1$ para sinais com diferentes ocupâncias.	97
Figura 44	Rede neural 10-4-1.	98

Figura 45	Comparação entre as performances dos filtros FIR, o filtro casado e o estimador neural.	99
Figura 46	Neurônios na camada oculta da rede classificadora.	100
Figura 47	Curvas ROC para diferentes sinais de teste.	101
Figura 48	Neurônios na camada oculta da rede classificadora.	102
Figura 49	Comparação entre as performances dos filtros FIR, o filtro casado e o estimador neural com patamar em zero.	103
Figura 50	Dados de entrada/saída dos neurônios da camada escondida para sinal teste com 30 % de ocupância.	104
Figura 51	Pesos da camada de entrada (sem bias).	104
Figura 52	Pesos de entrada do neurônio 4 comparados aos de um filtro FIR de ordem 9.	105
Figura 53	Regressão linear.	106
Figura 54	Contribuição de cada neurônio antes da combinação linear de saída com bias.	107
Figura 55	Histograma do sinal de saída da rede.	107
Figura 56	Quantização dos pesos de entrada.	112
Figura 57	Quantização dos pesos da camada de entrada levando em consideração	

	a diferença de escala entre os neurônios 1, 2 e 3 e o neurônio 4.	113
Figura 58	Lógica de decisão para leitura da LUT.	115
Figura 59	Exemplo de truncamento de palavra binária.	116
Figura 60	Quantização da função de ativação dos neurônios 1, 2 e 3.	116
Figura 61	Função de ativação em uma LUT.	117
Figura 62	Quantização da função de ativação do neurônio 4.	118
Figura 63	Função de ativação em uma LUT.	118
Figura 64	Quantização dos pesos da camada escondida.	119
Figura 65	Representação da rede quantizada.	120
Figura 66	Classificação dos circuitos VLSI (MEYER-BAESE, 2007).	122
Figura 67	Classificação dos circuitos VLSI (MEYER-BAESE, 2007).	123
Figura 68	Arquitetura do estimador neural em FPGA.	124

LISTA DE TABELAS

Tabela 1	Condições de funcionamento do LHC durante <i>Run-1</i> e <i>Run-2</i> (NAKAHAMA, 2015).	37
Tabela 2	Pesos sinápticos na camada escondida.	106
Tabela 3	Representações com sinal e sem sinal para palavra binária de três bits.	110
Tabela 4	Influência da posição do ponto na interpretação de uma palavra binária.	111
Tabela 5	Pesos da camada de entrada da rede estimadora com 5 algarismos significativos.	112
Tabela 6	Comparação entre as arquiteturas implementadas em FPGA.	125

SUMÁRIO

1	Introdução	19
1.1	Motivação	19
1.2	Objetivos	21
1.3	Organização do texto	22
2	O CERN, LHC e o Experimento ATLAS	23
2.1	O CERN	23
2.2	O acelerador de partículas LHC	24
2.3	O experimento ATLAS	26
2.3.1	O sistema de coordenadas do ATLAS	27
2.3.2	O detector de trajetória	29
2.3.3	O sistema de calorimetria	30
2.3.3.1	O calorímetro eletromagnético	31
2.3.3.2	O calorímetro hadrônico	32
2.3.4	O detector de múons	35
2.4	O sistema de <i>trigger</i> do ATLAS	36
2.4.1	O primeiro nível de trigger	37
2.4.2	<i>High Level Trigger</i>	39
2.5	O programa de atualização do ATLAS	39
2.5.1	A eletrônica atual do Tilecal	40
2.5.2	A eletrônica do Tilecal na Fase-II	41
3	Redes neurais artificiais	44

3.1	Origem bioinspirada	44
3.1.1	Histórico	46
3.2	Modelagem matemática de um neurônio	49
3.2.1	Funções de ativação	50
3.3	Topologias de redes	52
3.3.1	Redes de camada simples	53
3.3.2	Redes multicamadas	53
3.3.3	Redes recorrentes	55
3.4	Processos de aprendizagem de uma RNA	56
3.4.1	Aprendizado supervisionado	57
3.4.1.1	O algoritmo <i>backpropagation</i>	59
3.4.2	Aprendizado não-supervisionado	67
4	Método proposto e base de dados	69
4.1	Estimação de energia utilizando filtros FIR	69
4.1.1	FIR determinístico	70
4.1.2	FIR utilizando <i>Least Squares</i>	70
4.2	Filtragem inversa utilizando redes neurais <i>feedforward</i>	73
4.3	Estimação não-linear e <i>online</i> de energia	74
4.4	Metodologia utilizada	76
4.4.1	Rede estimadora	78
4.4.2	Detecção	79
5	Simulações, resultados e discussões	83
5.1	Determinando o número de inicializações	83
5.2	Conjuntos para treinamento e validação	86
5.3	Ajuste dos critérios de parada	90
5.4	Influência da ocupância do sinal no treinamento	92

5.5	Definindo o número de atrasos de entrada e neurônios na camada oculta . . .	95
5.6	Detecção utilizando rede neural	98
5.7	Interpretação dos pesos sinápticos	102
6	Implementação em FPGA	108
6.1	Quantização do estimador neural	108
6.1.1	Aritmética de ponto fixo	109
6.1.2	Pesos de entrada	111
6.1.3	Função de ativação	114
6.1.4	Pesos da camada escondida e saída da rede	118
6.2	Estimador neural implementado em FPGA	121
7	Considerações Finais	127
	Referências	129

1 INTRODUÇÃO

Quando se deseja medir a alteração ou estado de uma variável em um processo físico ou químico, faz-se o uso de um sistema de instrumentação, que é composto desde os transdutores, que mantêm contato direto com o processo, até à eletrônica de condicionamento do sinal e tomada de decisão, sendo, esta última, responsável por selecionar apenas os dados ou eventos de maior relevância para o objetivo da aplicação industrial ou experimento. Para atingir estes objetivos, é necessário um sistema de aquisição eficiente, projetado de acordo com as condições operacionais da variável que se pretende medir. Nas últimas décadas, a conversão dos sinais analógicos em digitais, gerados pelos transdutores, permitiu que estes pudessem ser enviados aos computadores, para o uso de técnicas de processamento digital de sinais.

O processamento deste sinal é de suma importância em um sistema de aquisição de dados, visto que é fundamental amenizar as distorções que o ambiente de medição cria na variável que se deseja obter para atingir os critérios exigidos pelo processo em questão. Em ambientes que possuem uma alta taxa de eventos, a filtragem de sinais de interesse, através de um sistema de processamento digital, torna-se essencial para reduzir o fluxo de dados para armazenamento. Ocorre também que, a informação pretendida, muitas vezes, encontra-se imersa em ruído, o que dificulta tanto sua detecção, quanto a recuperação do valor real da variável medida.

1.1 MOTIVAÇÃO

Em experimentos nos quais se deseja atender critérios de precisão desafiadores, como em Física de Altas Energias, naturalmente, exige-se um complexo sistema de instrumentação. Nestes ambientes, a busca por eventos físicos que ocorrem raramente necessita de uma alta taxa de dados para que tais interações possam ser observadas em tempo hábil. A Organização Europeia para Pesquisas Nucleares (CERN), fundada na década de 1950, é responsável por explorar, principalmente, as estruturas que constituem a matéria e a força que as une. Encontrar as pistas que podem levar os

pesquisadores às respostas para os questionamentos de natureza fundamental da matéria, tem como requisito a utilização do maior e mais potente acelerador de partículas, criando as condições necessárias para que essa busca seja possível. Chamado de Grande Colisor de Hádrons, o LHC acelera prótons próximos à velocidade da luz, em um túnel subterrâneo na fronteira franco-suíça. Estas partículas são direcionadas por intensos campos magnéticos a percorrerem trajetórias em sentidos contrários, colidindo umas às outras em pontos específicos do LHC, dentro dos detectores de partículas.

O experimento ATLAS é responsável por extrair as variáveis necessárias para a reconstrução dos eventos que ocorrem em seu interior, permitindo observar com detalhes as sub-partículas criadas nas colisões. Isso só é possível devido aos seus sub-sistemas, nos quais os calorímetros, tanto o hadrônico, quanto o eletromagnético, desempenham papel fundamental. O calorímetro hadrônico (ou Tilecal) é um instrumento responsável por absorver e amostrar a energia de partículas classificadas como hádrons, constituídas de quarks e glúons. Possui alta segmentação, ou seja, para fornecer uma cobertura total do ponto de colisão com boa precisão, tem grande volume e quase 10.000 canais de leitura, agrupados em células. As células, ao interagirem com uma partícula, convertem a energia da mesma em um sinal elétrico de amplitude proporcional à energia depositada.

A taxa de colisões do LHC, que é de 40 milhões por segundo, gera uma quantidade imensa de dados em todos os sub-detectores do ATLAS. Para reduzir essa taxa de dados, emprega-se um sistema de filtragem *online* que subdivide-se em dois níveis, com a finalidade selecionar apenas os eventos de interesse. No primeiro nível de filtragem do Tilecal, atualmente, utiliza-se uma combinação de dois algoritmos: Filtro Casado e um circuito detector de pico. Este sistema de processamento de sinais digitais é responsável por detectar e estimar a energia nas células do calorímetro. Devido à alta taxa do LHC, isto deve ser feito em um período de tempo muito pequeno, encorajando a utilização de algoritmos simples, implementados em hardware dedicado (DSPs e FPGAs).

O LHC passa por um programa de atualização de suas condições operacionais, visando prepará-lo para um aumento da sua luminosidade, que é uma medida proporcional à probabilidade de colisões em um cruzamento de feixes de partículas. O aumento na probabilidade de ocorrência de uma colisão acarreta em um problema de sobreposição (ou empilhamento) de sinais nos canais de leitura do Tilecal. Isso ocorre, pois o pulso de amplitude proporcional à energia depositada na célula espalha a informação da energia no tempo. Este efeito reduz o desempenho da técnica atualmente implementada, aumentando a probabilidade de falso alarme e estimação incorretas.

Devido ao efeito de sobreposição de sinais, técnicas de equalização visando remover interferência inter-simbólica em canais dispersivos utilizando filtros FIR, foram propostas em Duarte (2015), mostrando-se mais eficazes que o método atualmente implementado. No entanto, estas técnicas utilizam de uma aproximação linear para realizar o processo de estimação e características não-lineares intrínsecas ao sinal de leitura reduzem a performance destes métodos.

1.2 OBJETIVOS

Este trabalho tem como principal proposta a implementação de um sistema de filtragem inversa baseado em redes neurais implementado em FPGA, capaz de levar em conta os parâmetros estatísticos de ordem superior que os filtros lineares não conseguem. O estimador neural deve possuir uma topologia fixa e de baixa complexidade computacional, atendendo aos pré-requisitos de latência e número de elementos lógicos exigidos pelo sistema de filtragem *online* do ATLAS.

Para alcançar os objetivos listados, o calorímetro hadrônico passa a ser interpretado como um sistema linear, no qual a amostra de energia desejada convolui com um sistema de fase não-mínima, que tem características dispersivas no tempo e não-linearidades, devido a desvio de fase aleatórios do sinal decorrente de diferentes tempos de voo das partículas que atingem o calorímetro. Logo, o método proposto deve realizar a deconvolução não-linear deste sistema, através da aproximação de um modelo inverso ao do calorímetro, que é visto como um sistema desconhecido. Foram avaliadas as topologias de um sistema de detecção e estimação da energia dos sinais. Os coeficientes das redes neurais foram encontrados utilizando algoritmos já consolidados para este propósito, sendo os parâmetros do treinamento ajustados através de buscas exaustivas dentro de um intervalo de valores limitadamente plausível. Para treinamento das redes, foi utilizada uma simulação de *Toy* Monte-Carlo que aproxima as características dos sinais de leitura do TileCal em ambientes de alta luminosidade. As análises sobre as melhores topologias, tanto para o detector, quanto para o estimador neural, levarão em consideração a performance das redes em todos os cenários de *ocupância das células*. A ocupância, neste trabalho, foi utilizada de forma indissociável de luminosidade, visto que também é uma medida proporcional à probabilidade de ocorrência de um evento. As figuras de mérito utilizadas foram a raiz do erro médio quadrático (RMSe), para as redes estimadoras, e o critério da curva ROC, juntamente com o produto SP, para os detectores baseados em redes neurais.

A melhor rede encontrada, segundo as figuras de mérito utilizadas, será quantizada em ponto fixo de forma a viabilizar sua implementação em múltiplos canais em paralelo nas FPGAs da nova eletrônica de processamento do Tilecal.

1.3 ORGANIZAÇÃO DO TEXTO

No capítulo que se segue, será apresentada uma visão mais detalhada sobre o LHC, o detector ATLAS e seus sub-sistemas, dando ênfase ao sistema de *trigger* do calorímetro hadrônico, destacando as mudanças que ocorrerão na atualização do experimento para o cenário de alta luminosidade.

No Capítulo 3 foi realizada uma revisão bibliográfica sobre as principais topologias e métodos de treinamento das redes neurais artificiais, culminando em sua aplicação para estimação de energia em tempo real.

O Capítulo 4, enfatiza-se de forma detalhada toda a metodologia utilizada no trabalho, incluindo a geração do banco de dados de treinamento, teste e avaliação dos resultados.

O Capítulo 5 é organizado de maneira a exibir todos os resultados obtidos, comentando cada um deles, finalizando com a topologia escolhida como ótima para implementação em FPGA.

Já no Capítulo 6, apresenta-se o processo de quantização dos pesos e função de ativação, otimizando-os no quesito performance e complexidade computacional, visando à implementação em FPGA. No final, avalia-se sobre a viabilidade, finalizando com uma avaliação sobre a viabilidade deste sistema em ambiente com alta taxa de eventos.

Por fim, são apresentadas as considerações finais e as propostas para trabalhos futuros.

2 O CERN, LHC E O EXPERIMENTO ATLAS

Neste capítulo, apresenta-se o ambiente, assim como as condições operacionais em que se desenvolveram este trabalho. Além disso, uma visão geral acerca do CERN e os experimentos que são, atualmente, desenvolvidos no LHC, em especial, no programa de atualização para alta luminosidade, com ênfase no experimento ATLAS, seus calorímetros e o sistema de *trigger*.

2.1 O CERN

A Organização Europeia para Pesquisas Nucleares, CERN (do francês, *Conseil Européen pour la Recherche Nucléaire*), está situada na fronteira franco-suíça próximo a Genebra. Fundada em 1954, a instituição foi um dos primeiros empreendimentos conjuntos da Europa, possuindo hoje vinte e dois Estados membros, como se mostra na Figura 1.

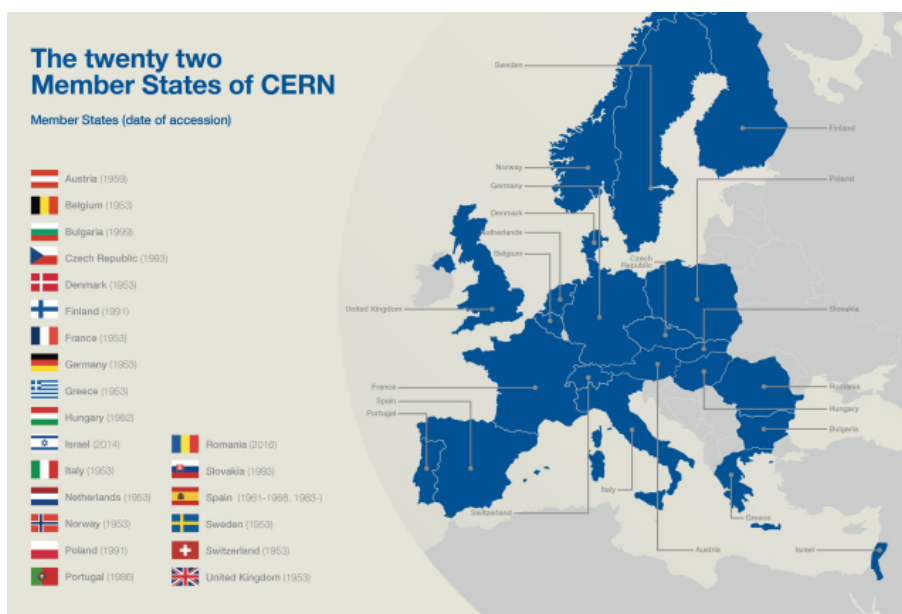


Figura 1: Os vinte e dois países membros do CERN (MELIS, 2016).

Estes países têm o dever de contribuir financeiramente para cobrir os custos opera-

cionais dos programas em funcionamento no CERN, além de possuírem o privilégio de terem representação no conselho da instituição, o qual possui autoridade máxima, controlando atividades de caráter científico, técnico e administrativo. Vale ressaltar que cada Estado membro indica dois delegados oficiais, sendo um representante da administração de seu governo e outro, interesses científicos nacionais. Há também países ou instituições em condições de “observador”, os quais podem frequentar o conselho, porém, não fazem parte dos procedimentos de tomada de decisão da organização. E aqueles Estados não-membros, que possuem acordos de cooperação com o CERN, como é o caso do Brasil. Atualmente, emprega-se pouco mais de 2.500 pessoas, recebendo cerca de 12.000 cientistas visitantes de mais de 120 nacionalidades (CERN, 2016a, 2016h).

O CERN tem por objetivo explorar as estruturas constituintes da matéria e as forças que a mantém unida. Para atingir tal propósito, faz-se o uso de complexas estruturas para acelerar feixes de partículas próximos à velocidade da luz, conhecidos como aceleradores de partículas, e colidindo esses feixes dentro de enormes experimentos. Estes, por sua vez, são responsáveis por absorver, amostrar a energia e identificar as partículas subatômicas geradas após cada colisão através de seu sistema de aquisição e gravação que, posteriormente, são estudadas por pesquisadores de todo o mundo. Esse processo dá pistas de como as partículas interagem, proporcionando novas compreensões sobre as leis da natureza (CERN, 2016a; ANDRADE FILHO, 2009).

2.2 O ACELERADOR DE PARTÍCULAS LHC

Na década de 1930, foram desenvolvidos os primeiros aceleradores com o propósito de investigar a estrutura do núcleo atômico e, a partir daí, vários outros aspectos da física de partículas. Sua função principal é acelerar feixes de partículas utilizando campos elétricos e guiando-os através de túneis por vários quilômetros com o auxílio de campos magnéticos. Podem ter uma configuração de anel, no qual os feixes circulam, ou uma linha reta (CERN, 2016g).

O Grande Colisor de Hádrons, LHC (do inglês, *Large Hadron Collider*), é o maior acelerador de partículas já construído, o qual entrou em operação no dia 10 de setembro de 2008. Durante sua primeira grande tomada de dados, chamada de *Run-1*, que ocorreu entre 2009 e 2012, ao analisar dados coletados de colisões entre abril e junho de 2012, com energia de centro de massa de 8 TeV, foi detectado o Bóson de Higgs, fortalecendo ainda mais o Modelo Padrão, que por sua vez, descreveu com eficiência interações de partículas em altas energias nas últimas quatro décadas (CERN, 2016g;

EVANS; BRYANT, 2008; AAD et al., 2012; MAEDA; ATLAS et al., 2015).

O LHC opera acelerando dois feixes de prótons em sentidos contrários dentro de um anel com 27,4 km de circunferência a 100 m de profundidade, na fronteira franco-suíça, juntamente com seus principais experimentos, como ilustrado na Figura 2. Projetado para observar colisões de 14 TeV de energia no centro de massa do feixe, atualmente o LHC encontra-se na sua segunda grande tomada de dados (*Run-2*), que teve início em 2015 e deve durar até 2018, colidindo prótons a 13 TeV (PERALVA, 2012; MAEDA; ATLAS et al., 2015). O cronograma de *upgrades* do LHC será detalhado na Seção 2.5.

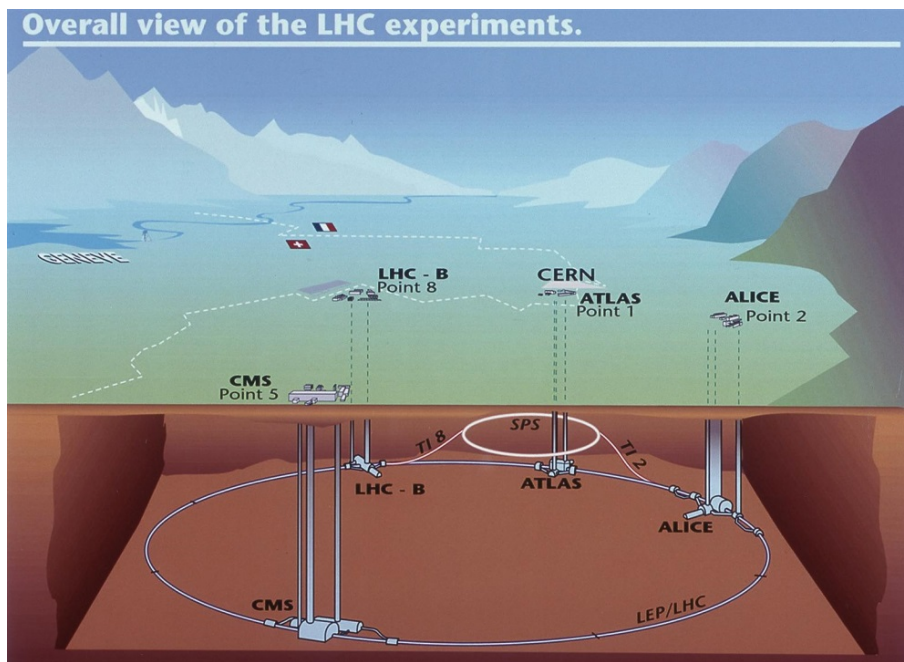


Figura 2: Visão geral dos experimentos do LHC e sua localização geográfica (TEAM, 1999).

Além da *Run-2* possuir uma energia de colisão de centro de massa maior, outro fator que caracteriza um aumento no desempenho das colisões é o aumento da luminosidade (L), que pode ser calculada pela Equação 2.1.

$$L = \frac{1}{\sigma} \frac{dN}{dt} \quad (2.1)$$

Esta é a quantidade que mede a capacidade de um acelerador de partículas produzir um número requerido de interações, sendo definida como o número de partículas (N) que se cruzam em um feixe com seção σ em certo tempo (t), e é dada em $cm^{-2}s^{-1}$ (HERR; MURATORI, 2003). Logo, com o acréscimo de luminosidade, o número médio de interações que ocorrem em cada cruzamento de feixe também aumenta, indo de 35 para 50 durante essa tomada de dados, com o objetivo de se atingir a casa de 200

interações com futuros *upgrades* do LHC. Anteriormente, o LHC, em funcionamento, tinha em seu interior milhares de feixes de prótons que circulavam espaçados a cada 50 ns. Atualmente, esse espaçamento foi alterado para 25 ns, aumentando a taxa de colisão para 40 milhões por segundo (40 MHz) no ponto de interesse dos detectores (MAEDA; ATLAS et al., 2015; MARSHALL; COLLABORATION et al., 2014).

Os quatro experimentos principais do LHC possuem sistemas síncronos com a taxa de colisão, adaptados à atual tomada de dados (*Run-2*) do acelerador. Cada um deles também possui objetivos específicos:

- ALICE (*A Large Ion Collider Experiment*)

Experimento projetado para colisão de íons pesados, dedicado às colisões nucleares para o estudo da interação entre a matéria e o plasma de quark-glúon (ALICE, 2008).

- LHCb (*Large Hadron Collider beauty*)

É um experimento dedicado à procura de uma física além do Modelo Padrão, através de violação de CP (assimetria entre quantidade de matéria e antimatéria no universo) e decaimentos de mésons B e D (LHCb, 2008).

- CMS (*Compact Muon Solenoid*)

É um experimento de propósito geral, que tem como principal objetivo o estudo do mecanismo de Higgs, assim como possíveis alternativas ao Modelo Padrão, o qual propõe uma teoria unificada que envolve supersimetria e dimensões extras (CMS, 2008).

- ATLAS (*A Toroidal LHC Apparatus*)

Também é um experimento de propósito geral projetado, de forma similar ao CMS, para aproveitar o máximo potencial e a grande variedade de fenômenos físicos que o LHC pode oferecer (ATLAS, 2008).

O experimento ATLAS é o ambiente no qual este trabalho foi desenvolvido e será detalhado a seguir.

2.3 O EXPERIMENTO ATLAS

Para determinar quais partículas foram geradas em uma colisão dentro de um acelerador, os pesquisadores devem se atentar às particularidades dos produtos dos decai-

mentos das partículas. Para isso, são projetados experimentos que possuem estágios, com o objetivo de identificar e mensurar aspectos de um evento como massa, velocidade e carga das partículas que interagem com ele. Os experimentos modernos são constituídos basicamente de três camadas de sub-sistemas: dispositivos de rastreamento, indicam a trajetória da partícula, localizados próximos ao ponto de colisão; calorímetros, destinados a absorver e medir a energia das partículas incidentes; e detectores de múons, localizados na camada mais externa e que utilizam técnicas variadas para determinar a trajetória e a energia dos múons, partículas que ultrapassam as camadas mais internas (CERN, 2017).

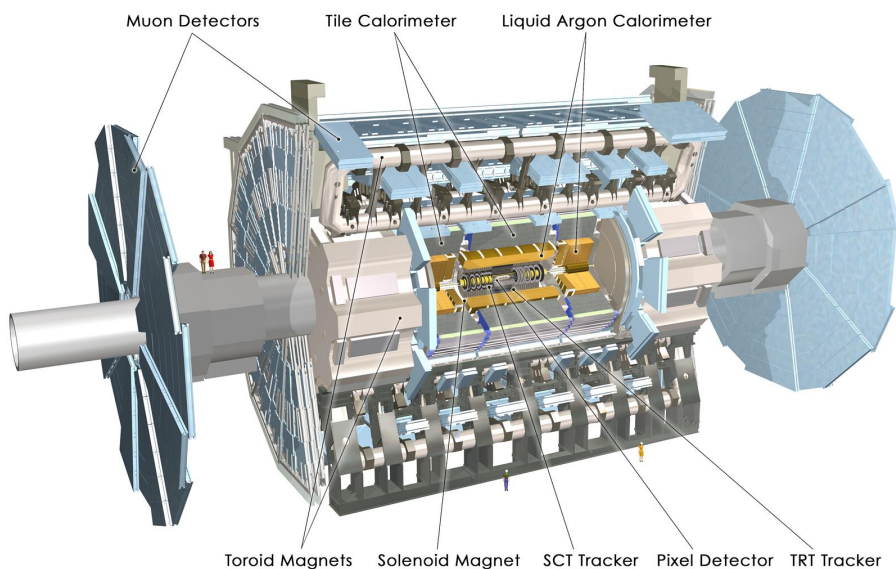
O ATLAS é o maior experimento do LHC. Possui formato cilíndrico com 46 m de comprimento e 25 m de diâmetro, pesando 7.000 toneladas (CERN, 2016b). Foi projetado para explorar uma grande variedade de fenômenos físicos, dando maior ênfase ao aprimoramento sobre os constituintes fundamentais da matéria, abordando questões como: quais os blocos constituintes da matéria? Quais forças que governam suas interações? O que aconteceu com a antimatéria no universo? O que é a matéria escura? Como era o universo primitivo e como ele irá evoluir? Como a gravidade se encaixa no Modelo Padrão? Para responder a essas perguntas, trabalham em colaboração no experimento ATLAS, mais de 3.000 cientistas de 174 institutos em 38 países (CERN, 2016d, 2016b).

Uma visão geral de todo o detector pode ser vista na ilustração da Figura 3(a). Na Figura 3(b) é demonstrado um corte transversal no detector. Nesta figura, pode-se observar que, dependendo do padrão da trajetória da partícula e de como ela deposita sua energia nos calorímetros, é possível discriminá-las. Partindo do ponto de colisão (camada mais interna do detector), tem-se o detector de trajetória, seguido pelos calorímetros eletromagnético e hadrônico e, por último, o espectrômetro de múons (ATLAS, 2008). Todos esses sub-sistemas serão descritos com mais detalhes nas Seções 2.3.2, 2.3.3 e 2.3.4.

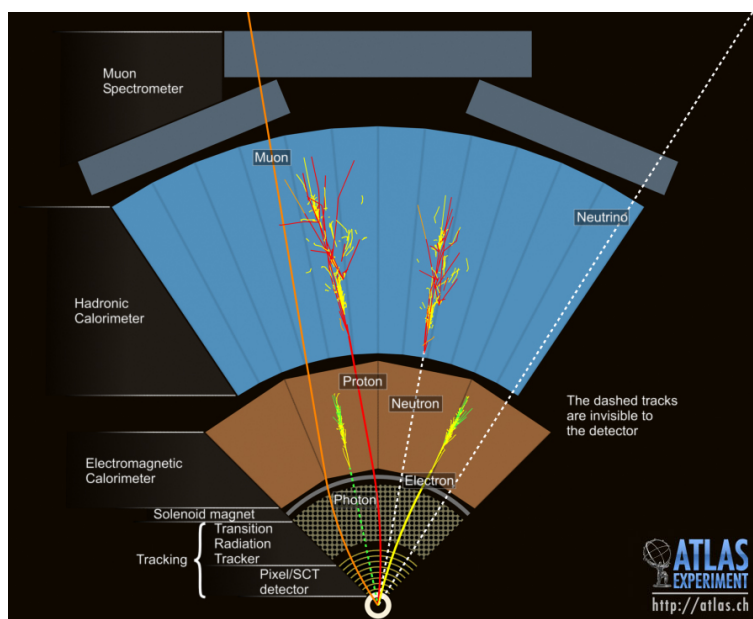
2.3.1 O SISTEMA DE COORDENADAS DO ATLAS

O sistema de coordenadas do ATLAS e das partículas que surgem da interação entre prótons tem como origem o ponto de colisão. São três eixos, assim como o sistema de coordenadas da mão direita. No entanto, o eixo z é situado na direção do feixe e o plano xy encontra-se numa posição transversal à direção do feixe. O eixo x tem sentido positivo partindo da origem para o centro do anel do LHC, e o eixo y é positivo no

sentido da origem para cima (fora da caverna do LHC). A metade do detector que fica no sentido positivo do eixo z é referenciada como ‘lado A’ (*Cote A*), sendo a outra metade o ‘lado C’ (*Cote C*), como se mostra na Figura 4.



(a)



(b)

Figura 3: Em (a) visão geral do detector ATLAS e seus sub-sistemas (ATLAS, 2016). Em (b) é possível ver, em corte transversal, como cada detector interage com cada tipo de partícula gerada nas colisões (PEQUENAO, 2008b).

O plano transversal à direção do feixe (x - y) é dado pelo ângulo ϕ , que é medido ao redor do feixe, começando no eixo x positivo. O ângulo medido a partir do eixo z positivo é chamado de θ . Ambos os sistemas têm a mesma origem e, em aceleradores em que não são colididas partículas muito massivas, θ é frequentemente relatado em

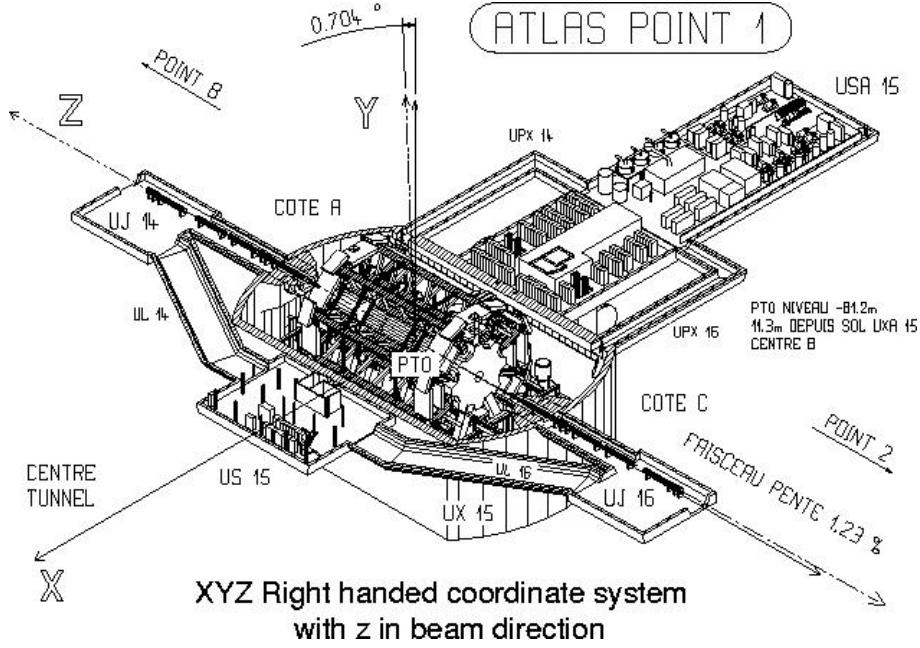


Figura 4: Sistema de coordenadas da mão direita tridimensional do ATLAS (CERN, 2016e).

termos da chamada pseudo-rapidez (η) (ATLAS, 2008), definida como:

$$\eta = -\ln \left[\tan \left(\frac{\theta}{2} \right) \right] \quad (2.2)$$

Ao analisar os feixes de partículas gerados nas colisões do LHC, têm-se valores nulos de η quando estes são perpendiculares ao ponto de colisão, assumindo um valor infinito quando os feixes são paralelos a z .

Nos detectores de partículas situados no LHC, ou em outros aceleradores, a maioria das colisões é tangencial, ou seja, gera desvios das partículas para as regiões com altos valores de η . Já as colisões frontais são mais raras e seus vestígios tendem a ocupar as regiões com menores valores de η . Conseqüentemente, a taxa de eventos nas células dos detectores por unidade de tempo aumenta com o valor do $|\eta|$. Os detectores serão vistos mais detalhadamente nas seções subsequentes e, ao decorrer deste trabalho, a relação entre $|\eta|$ e a taxa de eventos será chamada de ocupância (PERALVA, 2015; DUARTE, 2015).

2.3.2 O DETECTOR DE TRAJETÓRIA

O detector de trajetória (*Inner Detector*) é o primeiro detector do ATLAS a observar o grande número de decaimentos de partículas provenientes das colisões, ou seja,

está localizado na camada mais interna e, portanto, além de compacto, necessariamente possui uma fina granularidade, contendo aproximadamente 80 milhões de canais. É formado por três sistemas de sensores imersos em um campo magnético paralelo ao eixo dos feixes, medindo a direção, o momento e a carga de partículas eletricamente carregadas, geradas nas colisões entre prótons (AAD et al., 2008b; CERN, 2016c).

Os sistemas que o compõem são o detector de pixel (*Pixel Detector*), sendo o elemento mais interno do detector de trajetória, contido em uma área de $1,7 \text{ m}^2$. Externo ao detector de pixel, têm tiras semicondutoras, chamadas de *Semiconductor Tracking* (SCT), que juntamente com o detector de pixel, possuem uma alta granularidade, contendo milhões de *pixels*, responsáveis por dar a direção da maioria das partículas que são geradas após a colisão. O detector de radiação de transição (*Transition Radiation Tracker - TRT*) utiliza finos detectores em forma de tubo preenchidos com gás com 4 mm de diâmetro, e um fio de ouro com diâmetro de $30 \mu\text{m}$ em seu interior, responsáveis por detectar partículas eletricamente carregadas, como elétrons e píons, e, assim, ele completa a função desse sistema de detecção que cobre uma pseudo-rapidez de $|\eta| = 2,5$, como se ilustra na Figura 5 (ATLAS, 2008; PEETERS, 2003).

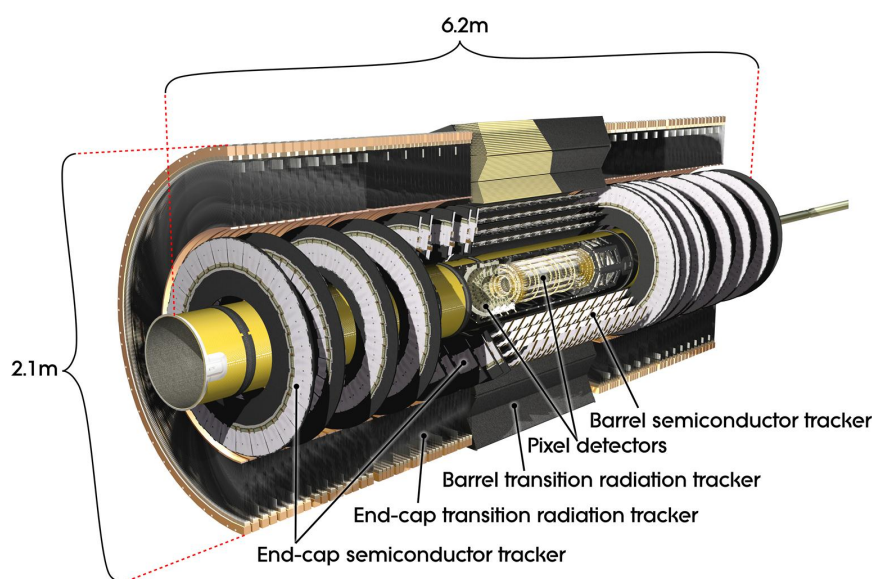


Figura 5: Detector de trajetória do ATLAS e seus sub-sistemas (CERN, 2016f).

2.3.3 O SISTEMA DE CALORIMETRIA

Em Física de Altas Energias, os calorímetros são responsáveis por medir a energia dos hádrons, elétrons e fótons, absorvendo-os através de um processo de geração de chuviscos de partículas dentro de materiais de alta densidade (passivos), e amostrando

parte de sua energia ao utilizar camadas intercaladas entre os materiais absorvedores e materiais ativos, geralmente argônio líquido ou plástico. Uma característica peculiar dos calorímetros é o seu volume, pois devem ser grandes o suficiente para que as partículas que o atravessam possam se extinguir em chuveiros dentro do mesmo. No ATLAS, os calorímetros são do tipo “amostradores”, no qual os chuveiros de partículas produzidos pelo material absorvedor têm apenas parte da energia medida pelo material sensor (ativo), assim, a energia do chuveiro total é estimada através da energia observada pelos detectores. Fazendo o uso de diferentes tecnologias de calorimetria, o sistema cobre uma pseudo-rapidez de até $|\eta| = 4,9$ (ATLAS, 2008).

Os calorímetros são peças fundamentais nos experimentos de Física de Partículas, sendo normalmente segmentados de forma transversa, permitindo medir, com alta eficiência, tanto a energia quanto a trajetória das partículas, sendo que estas podem ser eletricamente carregadas ou neutras. Possuem também rápida resposta, o que é uma característica positiva em um ambiente com alta taxa de eventos, além de a informação de energia medida por eles poder ser utilizada para filtrar eventos de interesse (PERALVA, 2015).

O sistema de calorimetria do ATLAS é ilustrado em corte transversal na Figura 6. É possível observar que os calorímetros eletromagnético e hadrônico são posicionados de acordo com o valor da pseudo-rapidez, que para valores pequenos de $|\eta|$ estão localizados o barril (*barrel*) e, para valores grandes, o barril estendido (*extended barrel*) e a tampa (*end-cap*). Os calorímetros do ATLAS serão mostrados com mais detalhes nas Seções 2.3.3.1 e 2.3.3.2.

2.3.3.1 O CALORÍMETRO ELETROMAGNÉTICO

O calorímetro eletromagnético, ou de argônio líquido (do inglês, *Liquid Argon - LAr*), é constituído de material absorvedor de chumbo e aço, intercalados com eletrodos de cobre encrustados em um material polímero altamente resistente às baixas temperaturas (kapton) e imersos em argônio líquido, agindo como material ativo. Tem o formato de acordeão (ver Figura 7) e os eletrodos amostram a energia dos fótons e das partículas eletricamente carregadas, quando estas atravessam as regiões imersas em argônio líquido. Este processo ocorre através da ionização das células do calorímetro. É composto de duas partes: o barril (do inglês, *barrel*) e duas tampas (do inglês, *end caps*), cobrindo juntas um valor de $|\eta| < 3,2$. O barril situa-se na região de $|\eta| < 1,475$ e consiste de dois meio-barris idênticos, separados por um pequeno espaço de 4 mm em

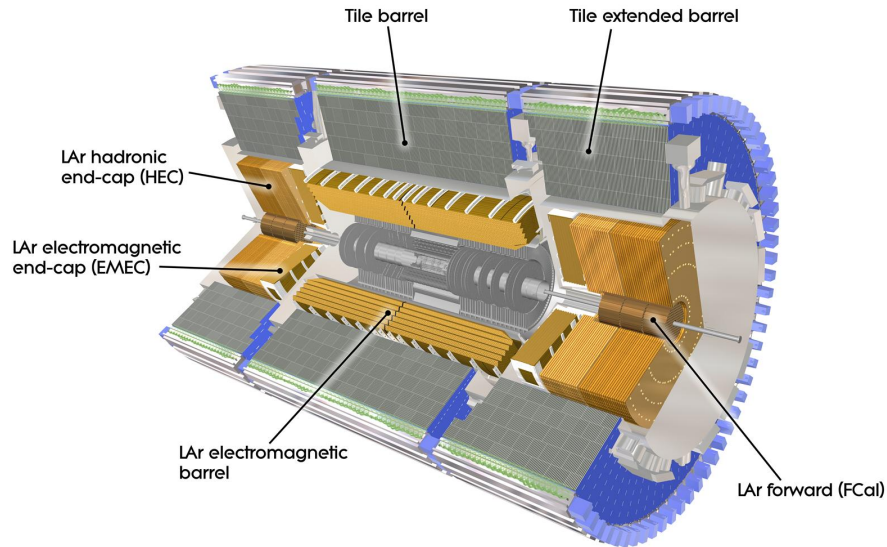


Figura 6: Sistema de calorimetria do ATLAS (PEQUENAO, 2008a).

$z = 0$. Cada tampa é mecanicamente dividida em duas rodas coaxiais, uma na região de $1,375 < |\eta| < 2,5$, chamada de tampa interior, e outra na região de $2,5 < |\eta| < 3,2$, chamada de tampa exterior (ATLAS, 2008; PERALVA, 2015). As partes que compõem o calorímetro eletromagnético podem ser vistas na Figura 6.

Para a maior parte do calorímetro de LAr, cada módulo possui três camadas com diferentes granularidades em profundidade, como esquematizado na Figura 7. É projetado para que a maior parcela de energia seja coletada na camada intermediária, deixando para a terceira e última camadas apenas a extremidade do chuveiro de partículas criado pelo material passivo. A primeira camada possui a mais fina segmentação dentre as três, tornando-a essencial na dedução da direção de partículas incidentes e, totalizando aproximadamente 200.000 canais de leitura em todo o calorímetro eletromagnético (NIKIFOROU, 2013).

2.3.3.2 O CALORÍMETRO HADRÔNICO

O calorímetro hadrônico do ATLAS é conhecido também como calorímetro de telhas (do inglês, *Tile Calorimeter* - Tilecal), e está situado externo ao invólucro do calorímetro de LAr. Seu nome se dá por serem utilizadas placas cintiladoras em forma de telha como material ativo. Assim como o calorímetro eletromagnético, também possui um material passivo de aço para absorver parte da energia das partículas que o atravessam, como ilustrado na Figura 8. É constituído de um barril (*Tile Barrel*) e dois barris estendidos (*Tile extended barrel*), localizados, respectivamente, do lado de

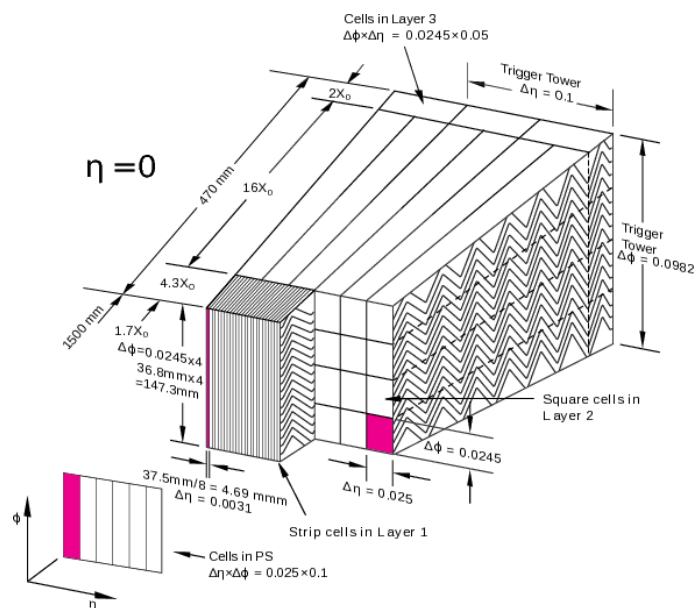


Figura 7: Esboço de um módulo do barril do calorímetro eletromagnético, mostrando suas três camadas e a granularidade nas coordenadas η e ϕ (NIKIFOROU, 2013).

fora do barril e da tampa do calorímetro de LAr, como pode ser visto na Figura 6. O barril cobre a região de $|\eta| < 1,0$, e seus dois barris estendidos, a extensão de $0,8 < |\eta| < 1,7$, como ilustrado na Figura 9(a). (GRAHN, 2008; PERALVA, 2015).

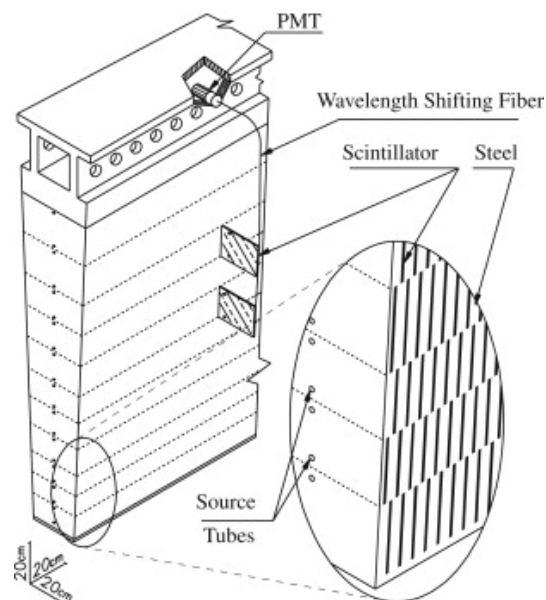
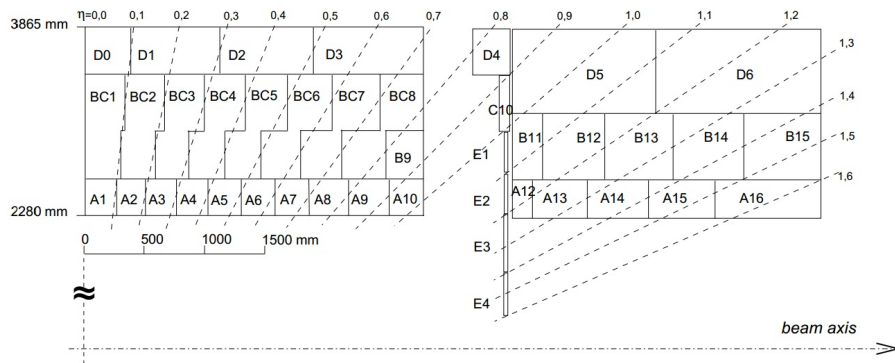


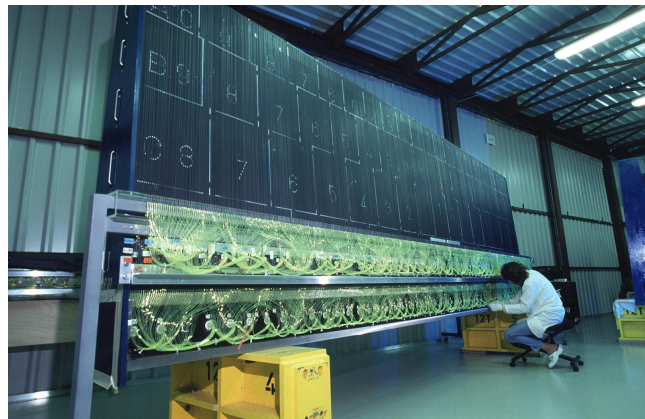
Figura 8: Disposição dos conjuntos de telhas cintiladoras e aço no calorímetro hadrônico, e a aquisição do sinal feita através de fibras ópticas, agrupadas pelas PMTs (AAD et al., 2008a).

O Tilecal tem como principal objetivo a identificação e separação de jatos ao medir sua energia e direção através de uma faixa dinâmica de medição extensa, caracterizando-o como um calorímetro de alta precisão. Por possuir boa cobertura do ponto de colisão,

absorve e mede a maior parte dos jatos hadrônicos emitidos do ponto de interação, sendo, também, peça chave para inferir a presença de neutrinos pelo momento faltante em alguma direção (ANDRADE FILHO, 2009). Um módulo do barril e do barril estendido estão representado na Figura 9. Os mesmos são constituídos, na sua totalidade, por 64 destes módulos, possuindo respectivamente, 23 e 16 células cada, totalizando, aproximadamente, 10.000 canais devido à dupla leitura, obtendo redundância e maior confiabilidade na medida da energia. Cada módulo deste está disposto lado a lado no sentido radial e juntos, circundando o plano transversal à direção do feixe de colisão, cobrindo todos os valores de ϕ com granularidade de $\Delta\phi = 0,1$. Cada módulo também é segmentado em profundidade, divididos em três camadas com uma granularidade de $\Delta\eta = 0,1$ para as duas primeiras camadas, e $\Delta\eta = 0,2$ para a terceira e mais externa das três, como pode ser visto na Figura 9(a) (PERALVA, 2015).



(a)



(b)

Figura 9: Em (a) é representada a localização do barril e barril estendido do Tilecal no sentido positivo de η , e sua divisão em células (ATLAS, 2008). Em (b) é mostrada uma foto tirada durante a fase de construção do Tilecal. Também é possível visualizar sua divisão em células e as PMTs (GUIRAUD, 1999).

Ao serem atravessadas por partículas ionizantes, as telhas cintiladoras emitem fótons altamente energéticos. Essa luz é coletada por duas fibras óticas independentes

(redundantes) localizadas uma de cada lado de cada telha. As fibras são agrupadas formando células. A luz resultante é enviada a tubos fotomultiplicadores (do inglês, *photomultiplier tubes* ou PMTs), resultando em um sinal elétrico para posterior processamento (GRAHN, 2008). O sinal das PMTs é então amplificado e condicionado através de um circuito de conformação, ou *shaper*, fornecendo um padrão de formato conhecido que tem sua amplitude proporcional à energia depositada (ANDERSON et al., 2005). Esse pulso é enviado a dois amplificadores de ganho relativo 64 (*high-gain* e *low-gain*) para melhorar a faixa dinâmica. Então, esse sinal analógico é digitalizado por conversores analógico-digital (ADC) com frequência de amostragem de 40 MHz que, durante o funcionamento do LHC, é sincronizado com o cruzamento de feixes de prótons.

Com essa configuração, o sinal, amostrado a cada 25 ns, é transferido para circuitos de *buffer* localizados em gavetas no próprio detector, contabilizando uma janela de 150 ns, que resulta em 7 amostras (ver Figura 10). Após aceitação pelo primeiro nível de *trigger*, as 7 amostras referentes ao evento de interesse são recortadas dos circuitos de *buffer* e enviadas através de fibras óticas para as ROD (*Read Out Drivers*) (PERALVA, 2015). As ROD recebem as 7 amostras de todas as PMTs para os eventos selecionados e realizam um pré-processamento, calculando o valor da energia em todas as células de leitura do detector, através da estimação da amplitude do sinal. Como o sistema é síncrono com o *clock* de colisão do LHC, o sinal é recortado de modo que a amostra central (quarta amostra) sempre corresponda ao pico do sinal. O sistema de *trigger* será explicado com mais detalhes na seção 2.4, e a eletrônica do Tilecal na Seção 2.5.1.

Em ambientes de alta luminosidade, algumas células do calorímetro apresentam uma alta ocupância de sinais. Como a resposta das células do calorímetro para uma interação dura mais de um *bunch crossing*, pode ocorrer a sobreposição de sinais. Este efeito de empilhamento de sinais no calorímetro, conhecido como *pile-up*, distorce a forma padrão do pulso, ocasionando erros de estimação da amplitude. Vale ressaltar que o mesmo é tema central deste trabalho, sendo discutido mais detalhadamente ao longo da Seção 2.5.2.

2.3.4 O DETECTOR DE MÚONS

Os múons são partículas similares aos elétrons, porém possuem muito mais massa e a sobrevida mais longa dentre as sub-partículas geradas no LHC, que não são completamente absorvidas pelos calorímetros. O detector de múons situa-se na camada

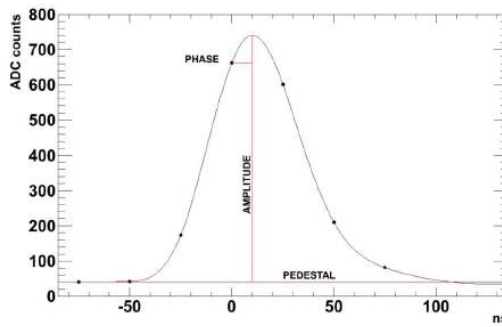


Figura 10: Pulso analógico típico do Tilecal digitalizado por um conjunto de 7 amostras, representadas por pontos ao longo da curva (PERALVA et al., 2013).

mais externa do ATLAS. É composto de grandes ímãs supercondutores e câmaras de traço altamente segmentadas, com o objetivo de medir a deflexão destas partículas através do intenso campo magnético, sendo, desta forma, possível inferir a energia dos mesmos (SCHLAGER et al., 2006; PERALVA, 2012).

2.4 O SISTEMA DE TRIGGER DO ATLAS

Em Física de Altas Energias, a busca por interações raras cria um ambiente único no qual uma alta taxa de eventos se faz necessária para aumentar a probabilidade de observação de tais interações. Nesse ambiente, os detectores geram sinais que devem ser processados com restrições de tempo, discriminando entre eventos de interesse ou ruído (eventos ordinários), reduzindo, assim, a quantidade de memória necessária para armazenar os dados. Tal sistema é denominado sistema de *trigger* ou filtragem *online* (ANDRADE FILHO, 2009).

O sistema de *trigger* do ATLAS deve estar adequado com as características de funcionamento do LHC. Durante a primeira grande tomada de dados, chamada também de *Run-1*, que ocorreu entre 2009 e 2013, condições como a energia de colisão de centro de massa, luminosidade e espaçamento entre os feixes de prótons fizeram com que os algoritmos e a eletrônica implementados fossem projetados para atender estes requisitos. Ao final da *Run-1*, o LHC passou pelo primeiro período de atualização (*first long shutdown - LS1*) de dois anos, que visava preparar tanto os experimentos quanto o acelerador para as novas condições de operação, chamada de *Run-2*, que teria início entre o fim de 2014 e começo de 2015 (PIANORI, 2014). As condições de funcionamento do final da *Run-1* e o esperado para a *Run-2* são comparadas na Tabela 1.

Durante o LS1, a estrutura de filtragem do ATLAS também se modificou. Antes

possuindo três estágios, os filtros de alto nível (HLT, do inglês, *high-level trigger*) da *Run-1*, chamados de L2 (*Level 2*) e EF (*Event Filter*), até então separados, foram unidos em um único HLT (NAKAHAMA, 2015; DUARTE, 2015). Essa nova configuração, possuindo apenas dois estágios, é composta pelo L1, no qual este trabalho está inserido, e o HLT, ambos esquematizados na Figura 11 e explicados com mais detalhes nas seções 2.4.1 e 2.4.2.

Tabela 1: Condições de funcionamento do LHC durante *Run-1* e *Run-2* (NAKAHAMA, 2015).

Período: Ano	Espaçamento entre os feixes	\sqrt{s}	Luminosidade Máxima	Colisões por Cruzamento
<i>Run-1</i> : 2012	50 ns	8 TeV	$8 \times 10^{33} \text{cm}^{-2} \text{s}^{-1}$	40
<i>Run-2</i> : 2015-2018	25 ns	13 TeV	$1 - 2 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$	25 - 50

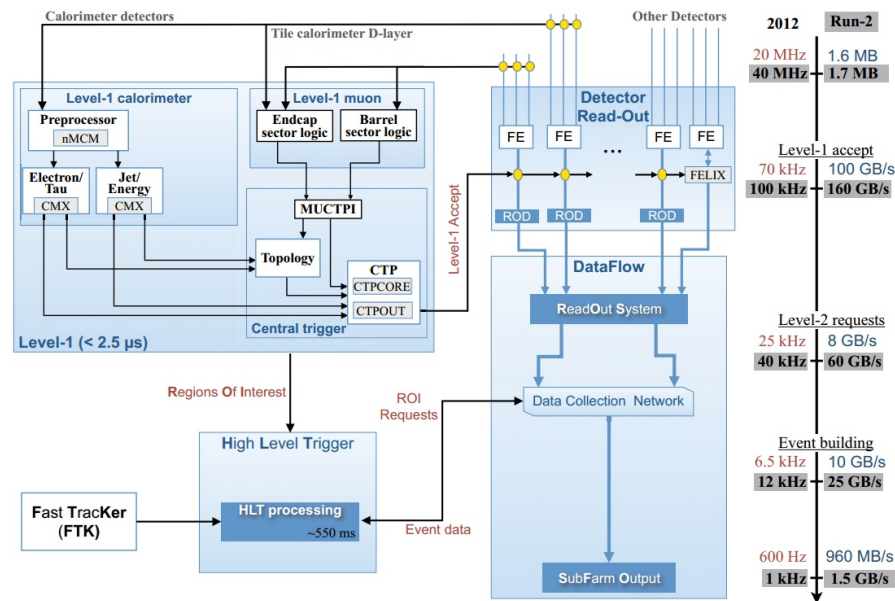


Figura 11: Esquema geral do sistema de *trigger* e DAQ durante a *Run-2* (NAKAHAMA, 2015).

2.4.1 O PRIMEIRO NÍVEL DE TRIGGER

A primeira etapa do sistema de filtragem *online* é responsável por filtrar eventos que chegam a 40 milhões por segundo, para cerca de 100.000 (100 kHz). Por lidar com uma taxa de eventos de 40 MHz, o L1 deve, necessariamente, possuir uma resposta rápida, evitando problemas de latência ao manter o tempo de processamento e tomada de decisão menor que $2,5 \mu\text{s}$ (ATLAS, 2008). Atualmente, o L1 tem como propósito

fundamental identificar unicamente o *bunch-crossing* que há um evento de interesse, no qual é associado um identificador (BCID). Para isso, é implementado em eletrônica dedicada e faz o uso da informação dos calorímetros (L1Calo) e do detector de múons (L1Muon) com reduzida granularidade, procurando por assinaturas físicas de interesse locais como candidatos a elétron, fóton ou múon, ou globais, como energia transversa faltante e energia total, através do processador de *trigger* central (do inglês, *Central Trigger Processor* ou CTP) (ATLAS, 2008).

Nos calorímetros, essa informação, com reduzida granularidade, é gerada pelas chamadas torres de *trigger*. Elas são formadas através da soma de sinais analógicos de células ao longo de η , cobrindo, geralmente, uma área de $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$. Através da Figura 9(a), é possível observar no Tilecal a disposição das células que formam uma torre de *trigger*. Nas células A1, BC1 e D0, pertencentes a um mesmo módulo e que, separadamente, fornecem 6 sinais distintos, formam uma das cerca de 2.000 torres de *trigger*. Outro exemplo é o grupo de células em $0,1 < \eta < 0,2$, no qual apenas a leitura de um lado da célula D1 seria utilizada para formar outra torre com BC2 e A2, resultando em 5 sinais distintos somados. Buscando simplificar a implementação e a manutenção, os circuitos somadores são os mesmos em todo o Tilecal, suportando até 6 sinais de entrada (CERQUEIRA; SEIXAS; CALÔBA, 2007; ANDRADE FILHO, 2009).

A decisão tomada pelo L1 em cada *bunch-crossing* é baseada na comparação da energia estimada dos sinais das torres de *trigger* a patamares pré-determinados de acordo com a assinatura física que se deseja procurar (como decaimentos de Higgs). A partir da *Run-2*, faz-se o uso de informações adicionais dos jatos ou múons no espaço, através de novos módulos topológicos (L1Topo), o que aumenta a eficiência na seleção de eventos, mantendo as taxas de *trigger* no nível atual sem ter que aumentar os patamares do decisor primário (o que poderia suprimir eventos de interesse físico) (SIMIONI et al., 2015).

Ao gerar um sinal de *trigger* pelo L1, os sinais de todo o detector — traço, calorímetros e múons —, armazenados nas memórias do tipo *pipeline* com granularidade nominal, passarão já digitalizados da eletrônica de *front end* para as ROD (eletrônica de *back end*), explicadas em detalhes na Seção 2.5.1.

Assim, os dados de saída das RODs do Tilecal são processados de modo a se estimar a amplitude do sinal proveniente de cada PMT para que se possa inferir a deposição de energia em cada célula do calorímetro, bem como uma estimativa da fase do pulso (para medidas de tempo de voo das partículas), além de uma informação do fator de quali-

dade (do inglês, *Quality Factor*) de cada medida. Esses dados reconstruídos são então formatados e enviados para o DAQ (do inglês, *Data Acquisition*). O primeiro estágio do DAQ é o sistema de leitura de saída (do inglês, *Read Out System*), que recebe os dados e armazena-os temporariamente em *buffers* locais para então serem processados pelo HLT, visando uma análise mais refinada (ATLAS, 2008; ANDRADE FILHO, 2009; DUARTE, 2015).

2.4.2 HIGH LEVEL TRIGGER

No HLT, faz-se o uso de algoritmos rápidos funcionando *online*, processando os eventos selecionados pelo L1, ou algoritmos que funcionam quase *offline*. Nele, é utilizada a informação do evento como um todo, rodando em *cluster* de PCs dedicados, com tempo de processamento em torno de 0,2 s. Ao final, o HLT reduz de 100.000 para aproximadamente 1.000 eventos por segundo (1 kHz), que serão gravados em mídia permanente. Mais detalhes técnicos sobre esta etapa de *trigger* e seus algoritmos podem ser encontradas nas referências (NAKAHAMA, 2015; SCHIAVI, 2015).

2.5 O PROGRAMA DE ATUALIZAÇÃO DO ATLAS

No presente trabalho aborda-se a eletrônica de estimação da amplitude dos pulsos do calorímetro hadrônico do ATLAS e da informação que será enviada para o primeiro nível de *trigger* em uma nova etapa de atualização do LHC. Mais especificamente, verifica-se processamento do sinal nas células do Tilecal em condições de alta luminosidade do LHC, chamado de HL-LHC. Esse estágio será alcançado após uma série de *upgrades* tanto na topologia quanto na eletrônica, preparando os detectores para essas condições únicas de radiação e fluxo de dados.

O programa de atualização do ATLAS se dá em três fases, que ocorrem durante os longos períodos de desligamento do LHC. Após a *Run-1*, deu-se início a Fase-0 durante o LS1, preparando o LHC e seus detectores para as condições descritas na Tabela 1. Para o período de 2019-2020 foi planejada a LS2, dando início à chamada Fase-I de atualização que, durante a *Run-3*, alcançará valores de luminosidade da ordem de $2 - 3 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$, o que corresponde de 55 à 80 interações por *bunch crossing*, no mesmo período de 25 ns. A Fase-2 termina a série de atualizações durante a LS3, prevista para acontecer entre 2024-2026, na qual, durante a fase de funcionamento, o LHC alcançará uma luminosidade máxima de $5 - 7 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$ (HL-LHC) (CERQUEIRA, 2015).

A atualização da eletrônica do Tilecal na Fase-2 será detalhada após uma breve revisão do que há atualmente.

2.5.1 A ELETRÔNICA ATUAL DO TILECAL

A cadeia eletrônica do Tilecal está ilustrada na Figura 12 e pode ser subdividida em duas partes: eletrônica de *front-end* (FE) e digitalizadores, e eletrônica de *back-end* (BE). Após a ocorrência de um evento físico no LHC, sub-partículas geradas na colisão, que atingem as células compostas de chapas de aço, geram chuveiros de partículas que, ao passarem pelo material cintilador, produzem luz, cuja captação é realizada por meio de fibras óticas (*wavelength shifting fibers*). Na eletrônica de FE, as fibras são agrupadas pelos misturadores (do inglês, *mixers*), formando células e, assim, otimizando a detecção da luz de maneira uniforme. Esse sinal luminoso é convertido em elétrico através das PMTs, e então é processado pela placa conhecida como *3-in-1 card*. Ela é responsável pelo condicionamento e amplificação do sinal, de tarefas de calibração e teste, além de amplificar o sinal conformado das PMTs com dois amplificadores de ganho relativo 64 (*low-gain* e *high-gain*), permitindo boa resolução e precisão de medida em uma vasta faixa dinâmica de medição. Após a amplificação, os sinais são enviados para as placas digitalizadoras (do inglês, *digitizer board*). A *3-in-1 card* também envia o sinal amplificado para uma placa somadora analógica, responsável por criar as torres de *trigger* (Seção 2.4.1), na qual o sinal analógico é enviado para o L1 através de cabos de par trançado de 70 m para a caverna de serviços principais (USA15), localizada a uma distância segura da radiação do detector (CERQUEIRA, 2015; TANG et al., 2013). Todos esses sistemas descritos acima ficam alojados em placas mãe (do inglês, *mother-boards*) no interior das gavetas (*drawers*), que são responsáveis por distribuir a alimentação para todo o circuito da FE, além de fornecer níveis de tensão de referência precisos para os circuitos de calibração e injeção de carga dos *3-in-1 card* (ATLAS, 2008).

Os sinais que saem da *3-in-1 card* são digitalizados a 40 MHz por placas digitalizadoras, contendo dois ADC dedicados de 10 bits, cada um deles destinado a digitalizar os sinais provenientes dos amplificadores de *low* e *high-gain*. Cada placa digitalizadora pode ser utilizada por até 6 canais de PMTs. Em condições normais de funcionamento, após o evento ser aceito pelo L1, as 7 amostras do pulso digitalizado armazenadas em memórias do tipo *pipeline* são recebidas pela placa de interface (do inglês, *interface board*). Esta placa recebe e distribui os sinais de TTC (*timing, trigger and control*), responsáveis por sincronizar todos os sub-detectores com o LHC (BC), além de receber e distribuir o sinal de *trigger* do L1. Este último é responsável por digitalizar o sinal

das torres analógicas, medir a energia, procurar grupos de células de alta energia e selecionar o evento que passará para o próximo nível com granularidade completa. A placa de interface também formata os dados para enviá-los através de links ópticos de alta velocidade para a eletrônica de BE quando um sinal de *trigger* é recebido, localizada na USA15. Neste ponto, encerra-se a cadeia de eletrônica de FE, que está distribuída em 256 gavetas (*drawers*) removíveis na parte mais externa do Tilecal, suportando, cada uma, até 48 blocos de PMTs. Os módulos do barril utilizam 44 das 48 posições disponíveis, e as gavetas dos barris estendidos são ocupadas com 32 posições (ATLAS, 2008; ASK et al., 2008; TANG et al., 2013).

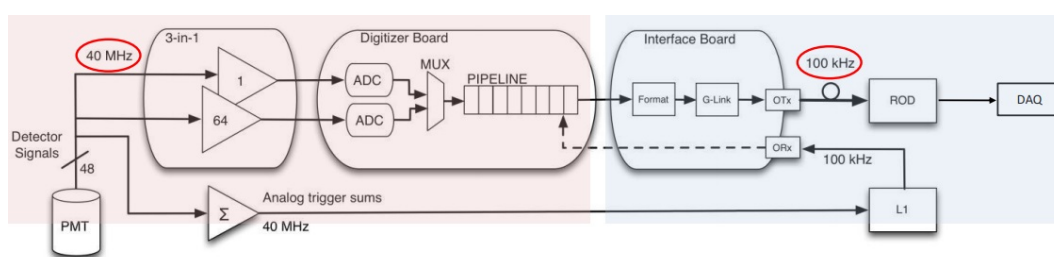


Figura 12: Cadeia eletrônica atual do Tilecal (CARRIÓ et al., 2013).

As RODs são os principais componentes da eletrônica de BE, realizando um pré-processamento do sinal da eletrônica de FE na taxa máxima de filtragem do L1 (100 kHz), como se mostra na Figura 12. Elas recebem os sinais da FE através de receptores ópticos (ORx), que são redirecionados por FPGAs (do inglês, *Field Programmable Gate Arrays*) para unidades de processamento (do inglês, *Processing Units* ou PUs) correspondentes, que são os principais núcleos de processamento das ROD. Uma PU contém dois processadores de sinal digital (do inglês, *digital signal processor* ou DSP) da Texas Instruments (TMS320C6414) (CARRIO et al., 2013). Nestes DSPs são implementados algoritmos de reconstrução *online* do sinal para obter a amplitude, a fase e um fator de qualidade do pulso reconstruído a partir das 7 amostras para cada PMT, ou seja, de todos os canais do Tilecal, em um tempo inferior a $10 \mu\text{s}$ (VALERO et al., 2008).

2.5.2 A ELETRÔNICA DO TILECAL NA FASE-II

Os principais objetivos dessa atualização são a substituição da antiga eletrônica por uma nova mais tolerante a um ambiente com alta radiação, buscar melhorias na confiabilidade com menos conectores, aumentar a precisão dos dados, melhorar o primeiro nível de *trigger* através da disponibilidade completa do detector, além de um incremento na relação sinal-ruído (SNR, do inglês, *signal to noise ratio*), e adequar suas fontes de energia devido à grande taxa de eventos nas PMTs, que irão requerer

mais corrente elétrica (CERQUEIRA, 2015). Para atender a estes requisitos, três opções de eletrônica estão em teste para o novo FE (CERQUEIRA, 2013):

1. *Front-End Modified 3-in-1 card.*
2. *Front-End - QIE Chip.*
3. *Front-End - FATALIC ASIC.*

A nova eletrônica de FE será composta de novas placas para condicionar o sinal das PMTs, assim como funcionalidades de calibração, amplificação e digitalização (dependendo da opção de FE), novas *Main Boards* (MB), que deverão realizar o controle de dados e também funções de digitalização (dependendo da opção de FE) e enviar para as novas *Daughter Boards* (DB). Será adaptada para um ambiente com alta taxa de transmissão de dados e fornecerá também uma interface com a eletrônica de BE através de links ópticos (CERQUEIRA, 2013). O novo sistema também deverá ser compatível a um *trigger* de primeiro nível totalmente digital, no qual o Tilecal será responsável por prover uma palavra digital com o valor da energia estimada para todos os canais (total granularidade) em cada BC (TANG et al., 2013; CARRIO et al., 2013). O circuito de geração desta informação é o tema central deste trabalho.

As DBs enviam os dados digitalizados para a eletrônica de BE, que os receberão através de receptores ópticos (ORx). A nova eletrônica de BE é composta principalmente pela chamada *super* ROD (sROD), que substituirá a atual ROD. As sROD realizarão todas as principais funções que a ROD já executa (reconstrução dos dados, TTC etc), porém serão adicionadas novas funcionalidades, como um pré-processamento para o primeiro nível de *trigger*. As sROD ainda encontram-se em fase de projeto e, atualmente, uma versão está em fase de teste, com as devidas modificações de interface, detalhadas nas referências (CARRIO et al., 2013; CERQUEIRA, 2013).

Devido à mudança de arquitetura necessária para a Fase-II, as sROD deverão lidar com a taxa de eventos nominal do LHC (40 MHz) onde os sinais de todos os canais do Tilecal serão pré-processados (PP) e armazenados temporariamente (*pipeline*) pela sROD, e enviados para o sistema de *trigger* (L1), que agora tomará sua decisão utilizando a granularidade completa dos calorímetros e detector de múons, como representado na Figura 13. Essa é uma das principais diferenças para a arquitetura atualmente implementada, em que as ROD lidavam apenas com os dados recortados pelo L1. O primeiro nível de *trigger* não receberá mais os sinais das torres de *trigger*, que deixarão de existir (CARRIO et al., 2013).

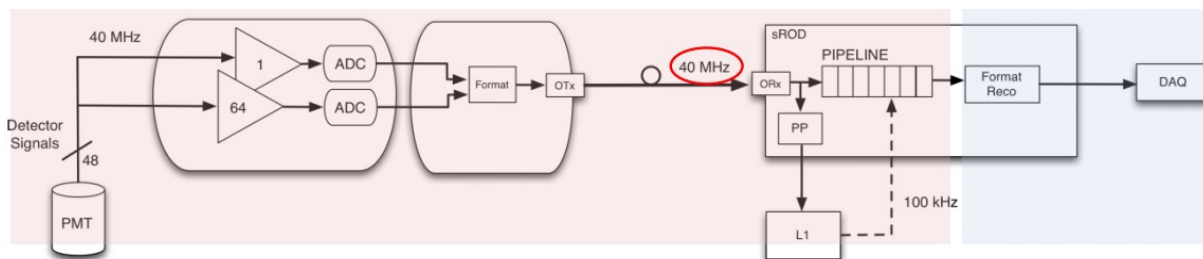


Figura 13: Cadeia eletrônica do Tilecal na Fase-II (CARRIÓ et al., 2013).

O pré-processamento realizado pelas sROD é o objetivo deste trabalho. Nele, FPGAs instaladas nas sROD fornecerão ao L1Calo uma estimativa da energia depositada nas células por *bunch-crossing*. Visto que o número de eventos por segundo é de 40 milhões, os algoritmos implementados neste sistema devem ser de simples implementação e de rápida resposta, evitando criar problemas de latência em todo o detector.

Durante a Fase-II, a configuração do HL-LHC criará cenários com alta ocupância nas células do Tilecal, o que ocasionará em um aumento considerável do ruído de *pile-up*. O *pile-up*, explicado de forma objetiva na Seção 2.3.3.2 é representado na Figura 14. Como pode ser visto, o sinal lateral referente à deposição de energia em um *bunch-crossing* anterior atrapalha a estimação da amplitude do sinal central do pulso característico. Soluções baseadas em equalização de canal usando filtros FIR (do inglês, *Finite Impulse Response*), detalhadas na Seção 4.1, foram propostas para substituir a técnica de estimação atual empregada no primeiro nível de *trigger*, o Filtro Casado. A técnica atual é altamente dependente da forma do pulso do Tilecal e tem seu desempenho drasticamente reduzida em ambientes de alta luminosidade (DUARTE, 2015).

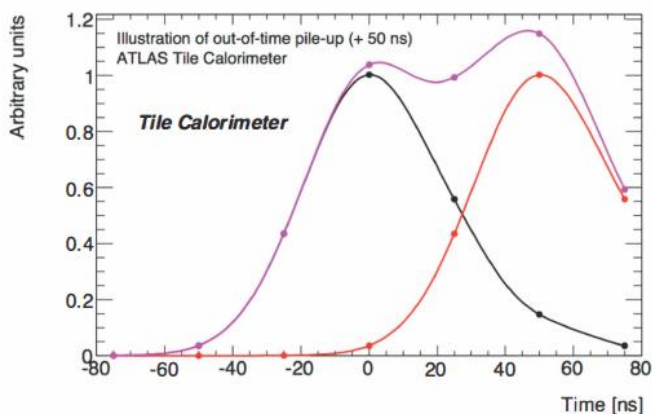


Figura 14: Ilustração do efeito de empilhamento de sinais (*pile-up*) (KLIMEK, 2012).

3 REDES NEURAIS ARTIFICIAIS

Neste capítulo é feita uma revisão da origem histórica das redes neurais artificiais, assim como suas aplicações, métodos de treinamento e principais topologias, no qual será dada uma maior ênfase à rede *feedforward*, topologia utilizada como base para este trabalho.

3.1 ORIGEM BIOINSPIRADA

Os estudos que envolvem as redes neurais artificiais ou simplesmente redes neurais (RNA) surgiram da constatação na qual o cérebro humano processa a informação de forma completamente diferente de um computador digital convencional. Do ponto de vista da computação, o cérebro humano é um sistema de processamento de informação altamente complexo, não-linear e paralelo, capaz de auto-organizar seus constituintes estruturais básicos, conhecidos como neurônios, de forma a executar funções como reconhecimento de padrões e controle motor. A enorme plasticidade dos neurônios permite que o sistema nervoso se adapte às mudanças do meio ambiente, sendo essencial para o funcionamento dos mesmos como unidades de processamento de informação e armazenamento de conhecimento (HAYKIN, 1999).

O tecido nervoso é composto basicamente de dois tipos de células: os neurônios e as neuróglia. O neurônio é a unidade fundamental, tendo como função básica receber, processar e enviar informações. Já as neuróglia ocupam o espaço entre os neurônios e são responsáveis pela sustentação, nutrição e defesa dos neurônios. A maioria dos neurônios é composta por três componentes principais: dendritos, corpo celular (ou soma) e axônios. Os neurônios trocam informações entre si na extremidade dos axônios em locais denominados sinapses. Eles não tocam uns aos outros, por isso, neste local há a chamada fenda sináptica, que separa a região entre a membrana pré-sináptica e a pós-sináptica (MACHADO, 2005).

Os dendritos são excitados pelos neurônios pré-sinápticos, gerando um potencial

elétrico. Este, por sua vez, é somado no corpo celular e, caso seja maior que um limiar, constitui um potencial de ação ou impulso nervoso. Esse sinal elétrico é transmitido através do axônio até suas extremidades, local onde são liberadas substâncias químicas chamadas de neurotransmissores para dentro da fenda sináptica. O tipo de neurotransmissor resultará em um impulso inibitório ou excitatório, gerando um potencial elétrico no dendrito de um neurônio pós-sináptico. A resultante desses impulsos nos dendritos de cada neurônio dá início a um novo impulso nervoso, que é disparado através do axônio até a liberação de novos neurotransmissores em outras conexões sinápticas. O efeito das sinapses pode variar e é esta capacidade que dá aos neurônios a habilidade de se adaptarem (MACHADO, 2005; BRAGA; FERREIRA; LUDERMIR, 2007). Está ilustrado na Figura 15 um modelo simplificado de um neurônio.

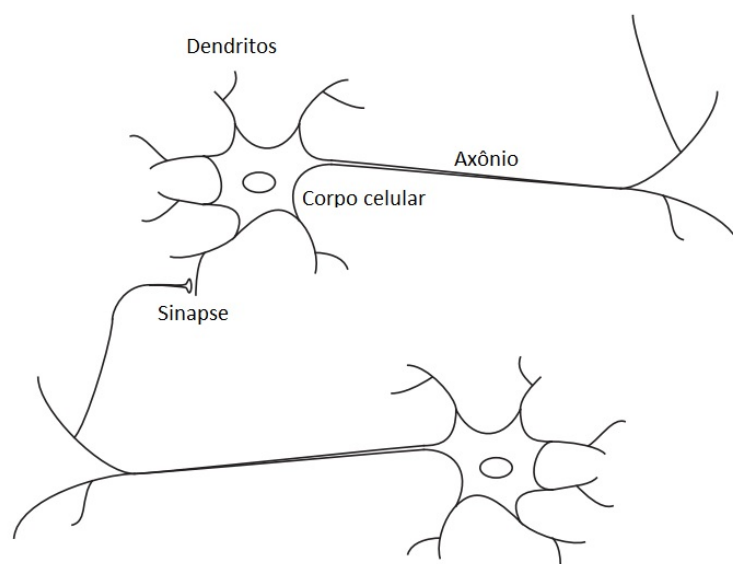


Figura 15: Desenho esquemático de neurônios biológicos (HAGAN et al., 1996).

Esse simples mecanismo, ilustrado na Figura 15, não é capaz de executar tarefas complexas. Porém, ao se formarem redes de comunicação entre neurônios, operando em paralelo, seu poder de processamento mostra-se evidente. Há, no cérebro humano, uma quantidade de neurônios da ordem de 10^{10} (dezenas de bilhões), altamente interconectados, cada um podendo ter de 1.000 à 10.000 sinapses, chegando a 150 trilhões destas em um adulto saudável (MACHADO, 2005; DRACHMAN, 2005).

Algumas estruturas neurais são adquiridas ao nascer e, outras partes, são desenvolvidas através de aprendizado. É durante este processo que conexões sinápticas são criadas, fortalecidas ou enfraquecidas, dando origem às novas memórias e/ou habilidades. Através de pesquisas realizadas no campo da neurociência, como em Maguire (2000), por exemplo, na qual se observou que motoristas de táxi de Londres têm a

área do cérebro relacionada à memória espacial mais desenvolvida devido ao fato de memorizarem muitas rotas. Acredita-se, atualmente, que novas memórias são formadas por modificações na intensidade de conexões sinápticas (HAGAN et al., 1996; MAGUIRE et al., 2000).

As redes neurais artificiais não se aproximam nem um pouco da complexidade do cérebro. No entanto, o surgimento da RNA baseou-se em duas características do tecido nervoso biológico: os blocos básicos são elementos computacionais simples e altamente interconectados; e a forma na qual os neurônios se conectam determina a função da rede. O campo de estudo das redes neurais artificiais desenvolveu-se através da colaboração de pesquisadores de várias áreas do conhecimento. Sua história progrediu tanto em nível conceitual, quanto em desenvolvimento de novas formas de implementação. Porém, como em muitas áreas da ciência, estes avanços não ocorreram de forma contínua (HAGAN et al., 1996).

3.1.1 HISTÓRICO

O trabalho que fomentou o pano de fundo para o surgimento das RNAs deu-se no fim do século XIX e início do XX, com colaboração multidisciplinar da física, psicologia e neurofisiologia pelos cientistas Hermann von Helmholtz, Ernst Mach e Ivan Pavlov. Este trabalho reforçou teorias como aprendizagem e visão, porém não incluiu nenhum modelo matemático que envolvesse a operação de um neurônio (HAGAN et al., 1996).

A partir do trabalho de Warren McCulloch, psiquiatra e neuroanatomista, e Walter Pitts, matemático, em 1943, criou-se a visão moderna da rede neural artificial, na qual, esta poderia computar qualquer função lógica e aritmética, descrevendo o funcionamento do primeiro modelo de um neurônio e dando início à área de inteligência artificial. Neste mesmo trabalho, foram lançadas também novas ideias sobre máquinas de estados finitos e representações lógicas de várias formas de comportamento e memória, influenciando um dos pioneiros da computação moderna, John von Neumann. O modelo de neurônio de McCulloch e Pitts é uma versão simplificada do funcionamento de um neurônio biológico e está ilustrado na Figura 16, sendo y as entradas que representam os estímulos recebidos pelos dendritos, p , que representa um axônio, e w , emulando o comportamento das sinapses. Neste modelo, o efeito de uma sinapse i no neurônio pós-sináptico é computado por $y_i w_i$. O efeito de todas as sinapses em cada neurônio pós-sináptico é somado (Σ) e comparado a um limiar (θ), ativando a saída ou não, de acordo com a Equação 3.1 (MCCULLOCH; PITTS, 1943; BRAGA; FERREIRA;

LUDERMIR, 2007).

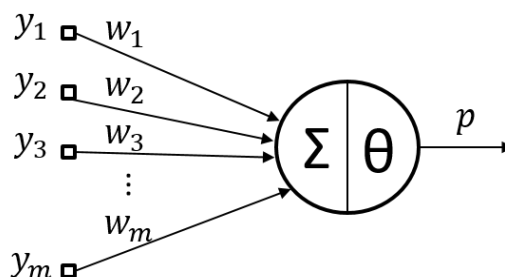


Figura 16: Neurônio de McCulloch e Pitts.

$$p = \sum_{i=1}^n y_i w_i \geq \theta \quad (3.1)$$

Em 1949, Donald Hebb propôs que o condicionamento clássico está presente devido à atuação de neurônios individuais. Sua teoria descreve um mecanismo da plasticidade sináptica, na qual a partir da estimulação repetida e persistente de células pós-sinápticas, fortalece-se esta interação sináptica, indicando que enquanto o indivíduo está aprendendo novas tarefas, essas sinapses se alteram continuamente (HEBB, 1949; HAGAN et al., 1996; HAYKIN, 1999).

A primeira aplicação prática de uma rede neural surgiu em 1958 com a invenção da rede *perceptron* e uma forma de aprendizagem, por Frank Rosenblatt. Fortemente influenciado pelas ideias de Hebb e outros teóricos do mesmo campo, como Hayek, Uttley e Ashby, propôs um método de aprendizagem em um experimento prático de reconhecimento de padrões (ROSENBLATT, 1958). Praticamente ao mesmo tempo, Bernard Widrow e Ted Hoff desenvolveram um novo algoritmo, que foi utilizado para treinar uma rede neural adaptativa linear com estruturas e potencial de aplicação similares ao *perceptron* de Rosenblatt, chamada de ADALINE. Este algoritmo é usado ainda nos dias atuais e conhecido como aprendizagem de Widrow-Hoff, ou algoritmo LMS (*Least Mean Square*). O LMS obteve mais sucesso em aplicações práticas do que o algoritmo de aprendizagem do *perceptron*, como, por exemplo, na área de processamento digital de sinais, na qual redes ADALINE foram utilizadas para cancelamento de eco em redes telefônicas. Essas demonstrações de sucesso atraíram grande interesse nas pesquisas relacionadas às redes neurais artificiais na época, no entanto, só conseguiam solucionar certa classe de problemas. No livro, publicado por Marvin Minsky e Seymour Papert, em 1969, intitulado *Perceptrons*, enfatizaram-se as limitações inerentes às redes de Rosenblatt e Widrow quanto às suas aplicações, as quais os mesmos estavam

cientos e propuseram novas arquiteturas. Essa modificação trouxe problemas quanto aos algoritmos de treinamento, que se mostraram ineficientes em redes mais complexas. O trabalho de Minsky e Papert influenciou uma grande parcela dos pesquisadores da época, chegando, até mesmo, a desencorajar agências a apoiá-los, acreditando, assim, que pesquisas com as redes neurais artificiais seriam frustradas. Isto, aliado o fato de, naquela época, não existirem computadores potentes para a realização de novos experimentos, fez com que muitos pesquisadores deixassem tal campo de estudos, fazendo com que pesquisas sobre as RNAs ficassem praticamente suspensas por cerca de uma década (MINSKY; PAPERT, 1969; HAGAN et al., 1996; HAYKIN, 1999).

Sob a perspectiva da Engenharia, a década de 1970 foi de grande dormência para as redes neurais. No entanto, trabalhos como o de Teuvo Kohonen (1972) e James Anderson (1972) desenvolveram, de forma independente, novas redes neurais que poderiam agir como memórias. Nesse mesmo período, Stephen Grossberg (1976), motivado pelo sistema visual humano, investigava uma categoria de redes, chamadas de redes auto-organizáveis (do inglês, *self-organizing networks*). Nos dias atuais, o trabalho de Grossberg deu origem a teoria de ressonância adaptativa (do inglês, *adaptive resonance theory*), ou redes ART (KOHONEN, 1972; ANDERSON, 1972; GROSSBERG, 1976; HAGAN et al., 1996).

Durante a década de 1980, os principais empecilhos que frearam o desenvolvimento das RNAs ao final dos anos de 1960 foram sobrepostos a partir do desenvolvimento dos computadores pessoais, fazendo com que fosse possível realizar experimentos para testar novas ideias que estavam surgindo, como em Hopfield (1982). O trabalho de John Hopfield fez ressurgir o interesse pelas redes neurais, pois além de ser um físico reconhecido da Caltech (Instituto de Tecnologia da Califórnia), reforçando credibilidade para o campo, possuía contatos na *AT&T Bell Laboratories*, na qual uma parceria permitiu o desenvolvimento de redes neurais em chips de silício, baseadas nas redes de Hopfield. A forma com que atacava os problemas o diferenciava dos pesquisadores que o precederam em tais estudos, pois enfatizava a praticidade tanto na implementação de sua rede, quanto no tipo de problema a ser resolvido. Outro fator determinante responsável pelo renascimento das redes neurais durante a década de 1980 foi o algoritmo para treinamento de redes *perceptron* multicamadas *backpropagation*, descoberto independentemente por vários pesquisadores diferentes. Esse algoritmo foi a resposta às críticas realizadas por Minsky e Papert (1969) quanto às limitações das redes *perceptron*. A publicação mais influente do *backpropagation* deu-se através do livro *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, editado por

James McClelland e David Rumelhart (1986), baseado no relatório de Rumelhart, Hinton e Williams (1985). (HOPFIELD, 1982; RUMELHART; HINTON; WILLIAMS, 1985; MCCLELLAND; RUMELHART; HINTON, 1986; HAGAN et al., 1996).

Ao final da década de 1980, o campo de pesquisa que envolvia o desenvolvimento de novas aplicações e ideias conceituais sobre redes neurais artificiais estava retomado, continuando, assim, pelas décadas seguintes. As maiores contribuições até os dias atuais se deram em novas topologias de redes e métodos de treinamentos, podendo ser encontrados na literatura (HAGAN et al., 1996; HAYKIN, 1999; BRAGA; FERREIRA; LUDERMIR, 2007).

3.2 MODELAGEM MATEMÁTICA DE UM NEURÔNIO

Um neurônio é uma unidade de processamento de informação essencial ao funcionamento de uma rede neural. Seu modelo simplificado está representado na Figura 17, e é constituído de quatro elementos básicos (HAYKIN, 1999):

- Conjunto de sinapses ou conexões

Cada sinapse é representada por um peso (constante), que pode assumir valores positivos ou negativos. Os principais autores utilizam a nomenclatura que representa os pesos com duas letras subscritas em uma variável, cuja primeira indica a qual neurônio esse peso está conectado e, a segunda, qual das entradas esse peso está relacionado. Usando a Figura 17 de exemplo, vê-se que um sinal de entrada (a_j), conectado a um neurônio k , está sendo multiplicado pelo peso sináptico w_{kj} , na qual j representa qualquer valor entre 1 e m . São ajustáveis de acordo com o propósito do neurônio ou da rede neural.

- Elemento somador

É responsável por somar todos os sinais de entrada, ponderados pelas respectivas sinapses em um neurônio, constituindo um combinador linear.

- Função de ativação (ou transferência)

Tem como objetivo limitar a amplitude do sinal de saída do neurônio a um valor finito e, geralmente, nos intervalos fechados $[0, 1]$ ou $[-1, 1]$. Os principais tipos utilizados serão detalhados na Seção 3.2.1.

- Bias

É um valor independente das entradas e que tem por objetivo deslocar o valor de entrada na função de ativação através de uma constante positiva ou negativa. É representado por b_k na Figura 17 e, assim como os pesos sinápticos, são ajustáveis de acordo com a aplicação.

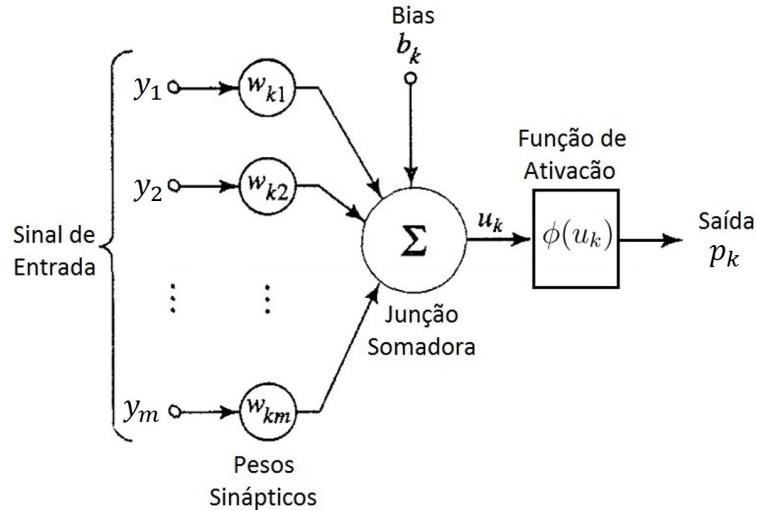


Figura 17: Modelo matemático de um neurônio (HAYKIN, 1999).

Logo, o neurônio k pode ser descrito pelas Equações 3.2 e 3.3.

$$u_k = \sum_{j=1}^m w_{kj} y_j + b_k \quad (3.2)$$

$$p_k = \phi(u_k) \quad (3.3)$$

Na qual u_k é o combinador linear seguido de uma translação (b_k), realizando uma transformação afim do hiperplano que resulta dessa combinação. Já p_k é o sinal de saída do neurônio k , que é o resultado de passar a função afim u_k como domínio para ϕ , que é, na maioria dos casos, uma função não-linear (HAYKIN, 1999).

3.2.1 FUNÇÕES DE ATIVAÇÃO

O projetista de uma rede neural clássica, antes de ajustar os pesos sinápticos através de algum método de treinamento (Seção 3.4), normalmente define a função de ativação que melhor se adéqua ao problema que a rede, potencialmente, deve resolver. Esta função pode ser linear ou não-linear e, dentre as mais populares, estão a degrau (ou

limiar), linear, logística e tangente hiperbólica (HAYKIN, 1999; HAGAN et al., 1996). Todas estas funções estão representadas na Figura 18.

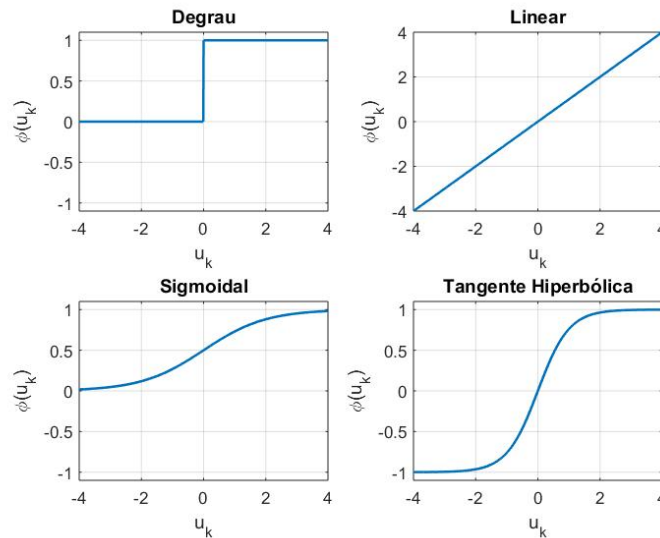


Figura 18: Funções de ativação típicas.

A função degrau tem como saída zero, caso o resultado da soma ponderada com o bias seja menor que zero e, +1, caso maior ou igual zero, sendo descrita pela Equação 3.4.

$$\phi(u_k) = \begin{cases} 1, & \text{se } u_k \geq 0 \\ 0, & \text{se } u_k < 0 \end{cases} \quad (3.4)$$

Já na função puramente linear, a saída nada mais é que uma combinação linear das entradas, representada pela Equação 3.5:

$$\phi(u_k) = u_k \quad (3.5)$$

As funções sigmóides, que possuem a forma de um “S”, são utilizadas na grande maioria de aplicações com redes neurais artificiais, possuindo um equilíbrio entre o comportamento linear e não-linear. São funções de crescimento monotônico assintótico em direção uma constante, quando o domínio da função tende a $\pm\infty$. A função logística é um exemplo de sigmóide, na qual sua imagem varia entre zero e +1, sendo descrita pela Equação 3.6. Ao variar o parâmetro α , são obtidas funções sigmóides com diferentes inclinações na origem, porém mantendo um limiar de saturação. Enquanto α se torna cada vez maior, a sigmóide se aproxima da função degrau, retornando apenas zero ou

+1, ao invés de valores contínuos neste intervalo. Outra característica da sigmóide que a torna importante em aplicações com RNA é por ela ser diferenciável, ao contrário da função degrau. Este fato será elucidado na Seção 3.4 (HAYKIN, 1999; LECUN et al., 2012).

$$\phi(u_k) = \frac{1}{1 + \exp(\alpha u_k)} \quad (3.6)$$

Em certos casos, é desejável uma função de ativação antissimétrica (ou ímpar), pois tendem a produzir resultados com média próxima de zero, acelerando o processo de treinamento (Seção 3.4). Para isso, existem funções que são correspondentes às citadas acima. No caso da função degrau tem-se a função *signal*, representada pela Equação 3.7. Já para a sigmóide utiliza-se a função tangente hiperbólica, descrita pela Equação 3.8 (HAYKIN, 1999; LECUN et al., 2012).

$$\phi(u_k) = \begin{cases} +1, & \text{se } u_k > 0 \\ 0, & \text{se } u_k = 0 \\ -1, & \text{se } u_k < 0 \end{cases} \quad (3.7)$$

$$\phi(u_k) = \tanh(u_k) \cong \frac{2}{1 + \exp(-2u_k)} - 1 \quad (3.8)$$

Outras funções de ativação clássicas podem ser vistas em Haykin (1999) e Hagan (1996). Há várias maneiras de interligar neurônios para formar uma rede, caracterizando topologias distintas. Em He (2015), são abordadas novas propostas de funções que pretendem resolver problemas de implementação e custo computacional das citadas nesta seção, porém, com a utilização de topologias de rede que não são abordadas neste trabalho (HE et al., 2015).

3.3 TOPOLOGIAS DE REDES

De forma geral, têm-se duas classes de redes neurais: as *feedforward* (ou acíclicas), que não possuem realimentação, e as redes recorrentes, que têm pelo menos um laço de realimentação. Como a utilização de apenas um neurônio é insuficiente para problemas que envolvem uma complexidade maior, faz-se necessária a associação desse neurônio a outros, formando uma rede. Essa rede pode possuir uma ou mais camadas, caracterizando, respectivamente, em redes de camada única (do inglês, *single layer networks*)

e redes de camadas múltiplas (do inglês, *multilayer networks*) (HAYKIN, 1999).

3.3.1 REDES DE CAMADA SIMPLES

As redes neurais de camada simples ou *perceptron* de camada simples (do inglês, *single layer perceptron* ou SLP) são denominadas as formas mais simples de rede utilizada para classificação de padrões que são linearmente separáveis. Possui uma camada de nós entrada (do inglês, *node input layer*), que se projetam em direção aos neurônios da camada de saída (do inglês, *output layer*), como representado na Figura 19 (HAYKIN, 1999). Os neurônios representados nesta figura são as unidades de processamento descritas na Seção 3.2. Cada seta representa uma conexão sináptica, logo, tem associada a ela um peso (constante), que irá multiplicar o valor do estímulo externo — através de um nó de entrada — correspondente. Seu nome se dá pois há somente uma camada de neurônios em toda sua extensão, que é a camada de saída.

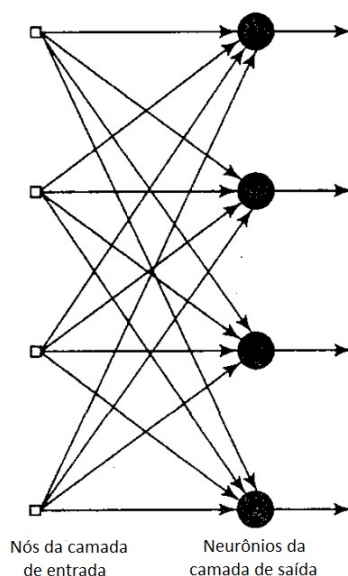


Figura 19: Rede *feedforward* de camada simples (HAYKIN, 1999).

3.3.2 REDES MULTICAMADAS

As redes multicamadas ou *perceptron* multicamadas (do inglês, *multilayer perceptron* ou MLP) são consideradas uma generalização das redes de camada simples por possuírem uma ou mais camadas entre os nós de entrada e os neurônios de saída. A estas camadas dá-se o nome de camadas escondidas (do inglês, *hidden layers*), sendo os neurônios nestas camadas chamados de neurônios escondidos (do inglês, *hidden neurons*). A modelagem dos neurônios de redes *perceptron* multicamadas envolve funções

de ativação não-linear, como detalhado na Seção 3.2.1.

De acordo com o que é visto em Haykin (1999), ao adicionar uma ou mais camadas escondidas, a rede é capaz de extrair parâmetros estatísticos de ordem superior, o que caracteriza a possibilidade de estimação ótima de parâmetros em ruído não-gaussiano, tarefa primordial para o presente trabalho. Enfatiza-se também que, apesar da rede possuir apenas conexões locais e um novo conjunto de pesos sinápticos, ela adquire uma perspectiva global sobre o problema através dessa dimensionalidade extra. Este fato torna as redes multicamadas poderosas ferramentas quando comparadas às de camada simples.

Hagan (1996) exemplifica que uma rede com apenas duas camadas, possuindo uma função de ativação sigmóide na primeira camada (camada escondida) e linear na camada de saída, pode ser treinada para realizar a aproximação de funções ou mapeamentos não-lineares de forma bastante eficiente. Na Figura 20 é ilustrada uma rede neural multicamadas com duas camadas escondidas.

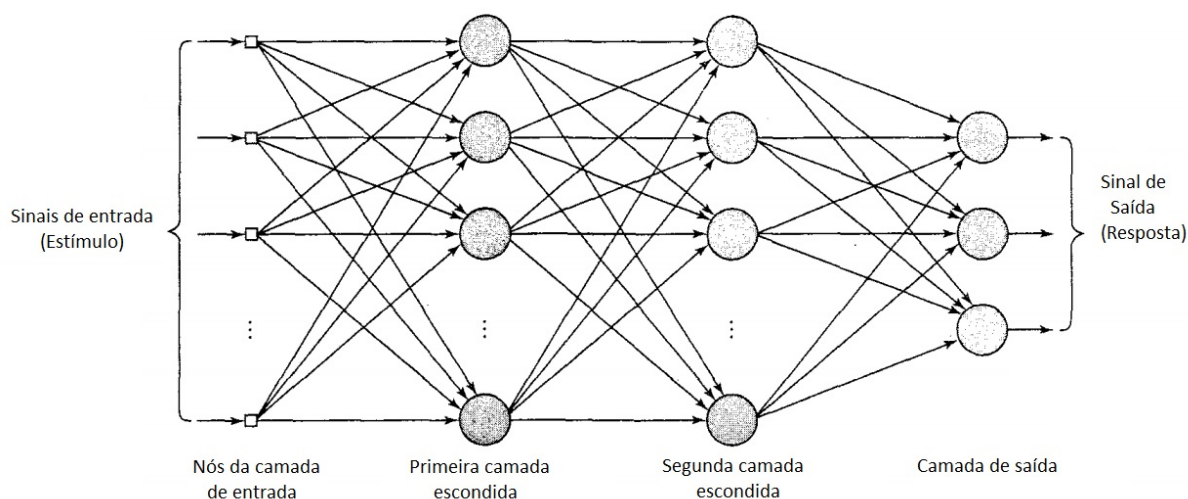


Figura 20: Representação de uma rede *feedforward* multicamadas, possuindo duas camadas escondidas. Nota-se que a saída da segunda camada escondida é conectada como entrada da camada de saída da rede.

Como em todas as redes do tipo *feedforward*, a rede multicamadas, após receber um estímulo externo através dos nós de entrada, o propaga — utilizando a referência da Figura 20 — sempre da esquerda para a direita, ou seja, dos nós de entrada, passando pelas camadas escondidas, até a camada de saída na qual o resultado computado por toda a rede é obtido. Há nomenclaturas para descrever uma rede *feedforward* multicamadas quanto à sua arquitetura de maneira mais objetiva, como a sugerida em Haykin (1999). Nela, uma sequência de números seria o suficiente para diferenciar as várias topologias de redes. Utiliza-se uma sequência de variáveis, $m-h_1-h_2-o$, na qual m seria

o número de nós de entrada, h_1 e h_2 o número de neurônios na primeira e segunda camada escondida, respectivamente, e o , o número de neurônios na camada de saída. Logo, uma rede PMC 4-4-3 representaria uma rede com apenas uma camada escondida, possuindo quatro neurônios escondidos e três de saída, com quatro nós entrada. Esta nomenclatura também pode ser estendida para representar redes com número de camadas escondidas maiores que duas.

Quando uma rede *feedforward* possui todos os pesos sinápticos, ou seja, cada nó em cada camada está conectado aos nós adjacentes na camada seguinte, diz-se que esta rede é uma rede completamente conectada (do inglês, *fully connected*). Caso algumas dessas conexões não estiverem presentes, denomina-se que a rede é parcialmente conectada (do inglês, *partially connected*). As redes *feedforwards* são utilizadas em aplicações como reconhecimento de padrões, filtragem adaptativa e controle automático de processos.

3.3.3 REDES RECORRENTES

As redes recorrentes baseiam-se na ideia de que nosso cérebro não possui a capacidade de responder a estímulos de forma sequencial apenas, mas também é capaz de realizar análises de caráter temporal através de elementos de memória, encontrando dependências entre eventos (HAGAN et al., 1996; HAYKIN, 1999). Uma rede recorrente difere-se claramente das redes *feedforward* multicamadas pelo fato de possuírem pelo menos um laço de realimentação. Um laço reconecta a saída de um neurônio para a entrada do mesmo e/ou de outros neurônios. Para realizar isto, este tipo de rede faz o uso de elementos de atraso (do inglês, *delay units*). A saída desses blocos sempre é a amostra de entrada uma unidade de tempo atrasada, implicando que as redes recorrentes devem, necessariamente, funcionar síncronas a um tempo discreto, que assume apenas valores inteiros. Na Figura 21 é ilustrada a topologia de uma rede recorrente, possuindo neurônios escondidos, nos quais os blocos de *delay* são representados por um atraso no tempo discreto (z^{-1}) (HAYKIN, 1999).

As redes neurais com laços de realimentação (recorrentes) são utilizadas em aplicações como: problemas de otimização combinatória, previsão de séries temporais e reconhecimento de fala e escrita. Elas se mostram bons modelos nestas aplicações pois têm como característica uma memória temporal (tanto da entrada como da saída) que persiste por um tempo relativamente maior que as redes tradicionais (*feedforward*), sendo uma poderosa ferramenta quando aplicadas aos sinais de natureza não estacionária. Dentre as topologias que utilizam realimentação, a rede de Hopfield é uma

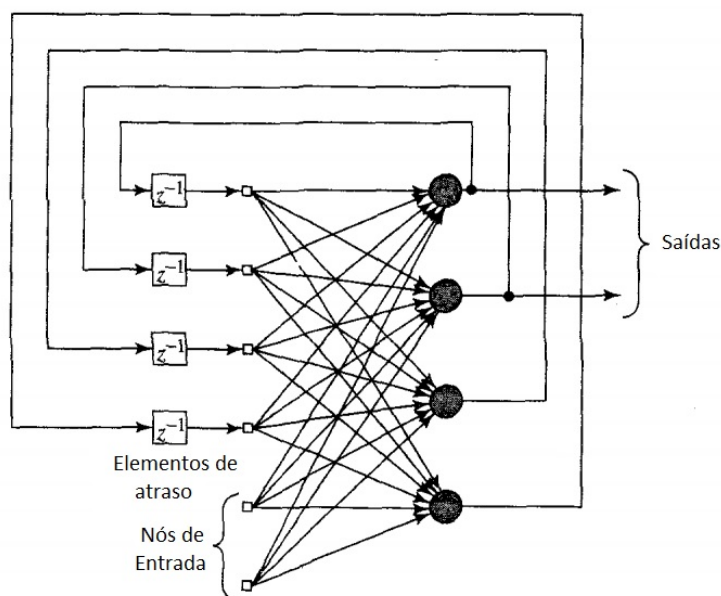


Figura 21: Exemplo de uma rede neural recorrente, possuindo apenas uma camada. Nota-se que a saída dos neurônios retornam às suas entradas através dos blocos de *delay*.

das mais conhecidas, trabalhando com um conceito profundo de armazenamento da informação (HAYKIN, 1999).

Neste trabalho optou-se pela utilização de redes *feedforward* multicamadas, pois em aplicações de processamento de sinais, elas apresentam uma estrutura análoga aos filtros FIR, chamadas de FIR MLP, diferentemente das redes recorrentes, que, por possuírem realimentação, são, neste caso, comparadas a filtros IIR. Esta consideração é importante do ponto de vista da estabilidade inerente às redes *feedforward*. Esta estrutura será implementada em um hardware dedicado (FPGA), utilizando aritmética de representação em ponto fixo. Apesar do treinamento das redes *feedforward* ser demorado, o mesmo não será um problema, pois tal mecanismo será realizado de maneira *offline*, sendo apenas sua estrutura com os pesos sinápticos já ajustados ao ambiente, implementada de maneira *online*. Ademais, estas são capazes de realizar mapeamentos não-lineares com boa eficiência, o que se encaixa com a proposta.

3.4 PROCESSOS DE APRENDIZAGEM DE UMA RNA

A maneira que uma rede é estruturada está ligada intimamente com o algoritmo de aprendizagem utilizado para treiná-la. Define-se aprendizagem de uma rede neural como sendo um processo em que seus parâmetros livres são adaptados através de uma estimulação pelo ambiente o qual ela é inserida. O tipo de aprendizado determinará

como seus parâmetros serão ajustados. Um conjunto de procedimentos com o objetivo de fazer com que a rede “aprenda” determinada função é chamado de algoritmo de aprendizado. Cada algoritmo executa essa tarefa de maneira diferente, existindo vários deles, com objetivos similares, porém, variando de acordo com a topologia de rede utilizada. Na literatura há basicamente dois paradigmas de aprendizado: o supervisionado e o não-supervisionado (HAYKIN, 1999; BRAGA; FERREIRA; LUDERMIR, 2007).

3.4.1 APRENDIZADO SUPERVISIONADO

É o método de aprendizado mais comumente utilizado para o treinamento de redes neurais artificiais. Recebe este nome por basear-se na presença de um agente supervisor (ou professor), que é incumbido de fornecer um parâmetro de qualidade da resposta da rede ao compará-la a um valor desejado e é ilustrado na Figura 22. Este valor desejado está associado a um padrão de entrada e é pertencente a um conjunto previamente conhecido chamado de conjunto de treinamento, devendo este, necessariamente, possuir pares entrada-saída associados. A rede é treinada de forma a atender um critério de erro, no qual é produzido a cada padrão ou conjunto de padrões apresentados pelo supervisor. Este processo é cíclico (iterativo), no qual em cada iteração são ajustados os pesos sinápticos de forma proporcional ao critério mencionado acima. Este processo leva a um problema de otimização, no qual uma função custo baseada no erro deve ser minimizada pelo algoritmo de treinamento. Geralmente, usa-se o erro médio quadrático como medida de desempenho neste processo, também sendo utilizada como função custo por vários algoritmos de treinamento (HAYKIN, 1999; BRAGA; FERREIRA; LUDERMIR, 2007).

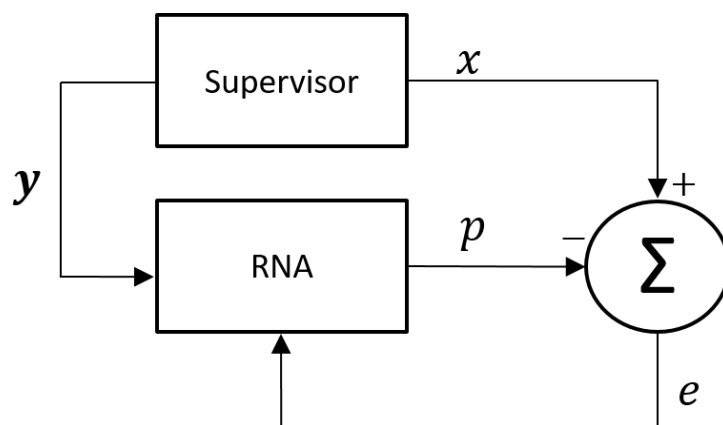


Figura 22: Aprendizado supervisionado.

O processo de ajuste dos pesos de um neurônio k de uma rede *perceptron* de camada simples é descrito através do diagrama de blocos da Figura 22 para uma RNA

constituída apenas pelo neurônio k . O erro produzido pela saída do neurônio k é dado pela Equação 3.9:

$$e_k(n) = x_k(n) - p_k(n) \quad (3.9)$$

sendo n o índice da iteração a qual este erro foi gerado. O erro atua como um mecanismo de controle para reduzir a diferença entre x_k e p_k , sendo utilizado para realizar ajustes nos pesos sinápticos do neurônios k passo-a-passo. Para isso, como explicitado anteriormente, uma função custo (Equação 3.10) deve ser minimizada, que é o valor do erro médio quadrático instantâneo.

$$E(n) = \frac{1}{2} e_k^2(n) \quad (3.10)$$

A minimização da Equação 3.10 resulta em uma regra de aprendizagem conhecida como regra delta ou regra de Widrow-Hoff. Escrevendo p em função de w_{kj} na Equação 3.9, é possível obter uma relação entre w_{kj} e o erro e_k . Assim, são usados algoritmos baseados em gradiente descendente para atualizar os pesos na direção contrária ao gradiente desta função, ou seja, na direção que minimiza o parâmetro de erro e_k . De acordo com esta regra, o ajuste $\Delta w_{kj}(n)$, aplicado ao peso sináptico w_{kj} na iteração n , é definido por

$$\Delta w_{kj}(n) = \eta e_k(n) y_j(n) \quad (3.11)$$

que é conhecido também como o algoritmo LMS. A variável η é uma constante positiva chamada de taxa de aprendizagem, responsável por fazer um ajuste proporcional aos pesos sinápticos. Logo, a atualização do peso sináptico do neurônio k é realizada por

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (3.12)$$

na qual $w_{kj}(n+1)$ recebe a atualização dos pesos para a próxima iteração do processo de treinamento. Vale ressaltar também que o bias é atualizado da mesma maneira que os demais pesos sinápticos e a Equação 3.12 aparece no algoritmo de treinamento do *perceptron* e ADALINE. É possível observar que a regra delta necessita do valor desejado de saída do neurônio para que seus pesos sejam adaptados aos padrões de entrada. Isso implica que ela só pode ser usada quando os neurônios são visíveis, isto é, quando a rede não possui camadas de neurônios ocultos, restringindo suas aplicações

somente às redes de camada única (Seção 3.3.1), e a apenas resolver de forma ótima, problemas de natureza linear (HAGAN et al., 1996).

Em virtude dessa limitação para aplicações com redes *perceptron* e ADALINE, alguns problemas requeriam a utilização de redes multicamadas, as quais na época (décadas de 1960 e 1970), não possuíam um método de treinamento eficaz que a consagrasse. Ao surgir na década de 1980, o *backpropagation* (Seção 3.1.1) trouxe novamente o entusiasmo para o campo de aplicações utilizando redes neurais artificiais, pois as redes multicamadas ofereciam uma enorme quantidade de aplicações quando comparadas a sua versão simplificada, as redes de camada simples.

3.4.1.1 O ALGORITMO BACKPROPAGATION

O algoritmo de retro-propagação do erro, ou simplesmente algoritmo de *backpropagation*, é uma generalização da regra delta para treinar redes *perceptron* multicamadas com qualquer número de neurônios e camadas escondidas. Vale ressaltar que além dos neurônios de uma MLP serem formados por funções de ativação não-linear, as mesmas devem ser diferenciáveis para atender aos critérios de funcionamento deste algoritmo, que tem duas premissas fundamentais (HAYKIN, 1999):

1. a partir de um conjunto de amostras apresentadas aos nós de entrada da rede MLP — previamente inicializada com pesos sinápticos arbitrários e mantidos fixos — computa-se o resultado desses estímulos de entrada serem propagados no *sentido direto*, ou seja, da entrada para a saída.
2. É gerado um sinal de erro entre a saída da rede após a passagem de um conjunto de amostra no sentido direto, e um valor desejado associado a este conjunto. Este erro é então propagado no *sentido reverso*, isto é, da camada de saída para as entradas, ajustando cada um dos pesos com o objetivo de fazer com que a saída da rede real aproxime-se do valor desejado, no sentido estatístico.

O objetivo do *backpropagation* é calcular o gradiente de uma função custo em relação a qualquer peso sináptico da rede. Logo, o sinal de erro na saída de um neurônio da camada de saída j em uma dada iteração n é (HAYKIN, 1999; BRAGA; FERREIRA; LUDERMIR, 2007):

$$e_j(n) = x_j(n) - p_j(n) \quad (3.13)$$

Sendo a energia instantânea do neurônio j definida como $\frac{1}{2}e_j^2(n)$, o correspondente para todos os neurônios da camada de saída, que são os únicos “visíveis”, é dado por:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (3.14)$$

em que C é o conjunto que engloba todos os neurônios da camada de saída. A função custo é dada pela média do erro ao serem passados todas as N amostras de treinamento para a rede:

$$E_{med} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (3.15)$$

A Equação 3.15, por ser composta da soma do erro ao fim da etapa de propagação dos estímulos na entrada, é função de todos os parâmetros livres da rede, pois a soma das resposta na camada de saída contém a atuação de todos os elementos da própria rede. Observado isto, ela pode ser usada como uma medida de qualidade do aprendizado, visto que a cada elemento da rede influencia no erro médio quadrático. A função custo, de acordo com as funções de ativação da RNA, gera uma superfície do erro. Se os neurônios são todos lineares, essa superfície toma forma quadrática, possuindo apenas um mínimo. Caso há elementos não-lineares em meio a rede, essa superfície poderá ter, além de um mínimo global, vários mínimos locais, dificultando métodos como o gradiente descendente clássico a aproximar-se deste mínimo global. O algoritmo de *backpropagation* utiliza o gradiente da superfície de erro para a busca da melhor solução, visando minimizar a função custo para todo o conjunto de treinamento. Logo, essa estimativa do erro é utilizada para decidir como cada um dos pesos serão ajustados (HAYKIN, 1999; BRAGA; FERREIRA; LUDERMIR, 2007).

Visando simplificar como a retro-propagação do erro ocorre, na Figura 23 é ilustrado o fluxo de sinal para um neurônio de saída j para uma rede MLP qualquer, na qual o bias é conectado a uma entrada que sempre é +1 e, w_{j0} representa seu valor. Para a amostra de treinamento n , após a camada anterior computar seus resultados, a soma ponderada entre esses valores e os pesos sinápticos dessa camada de saída é dado por:

$$u_j(n) = \sum_{i=0}^m w_{ji}(n)p_i(n) \quad (3.16)$$

sendo m o número de entradas (bias não incluso) aplicadas ao neurônio j , sendo sua

função de saída dada por:

$$y_j(n) = \phi(u_j(n)) \quad (3.17)$$

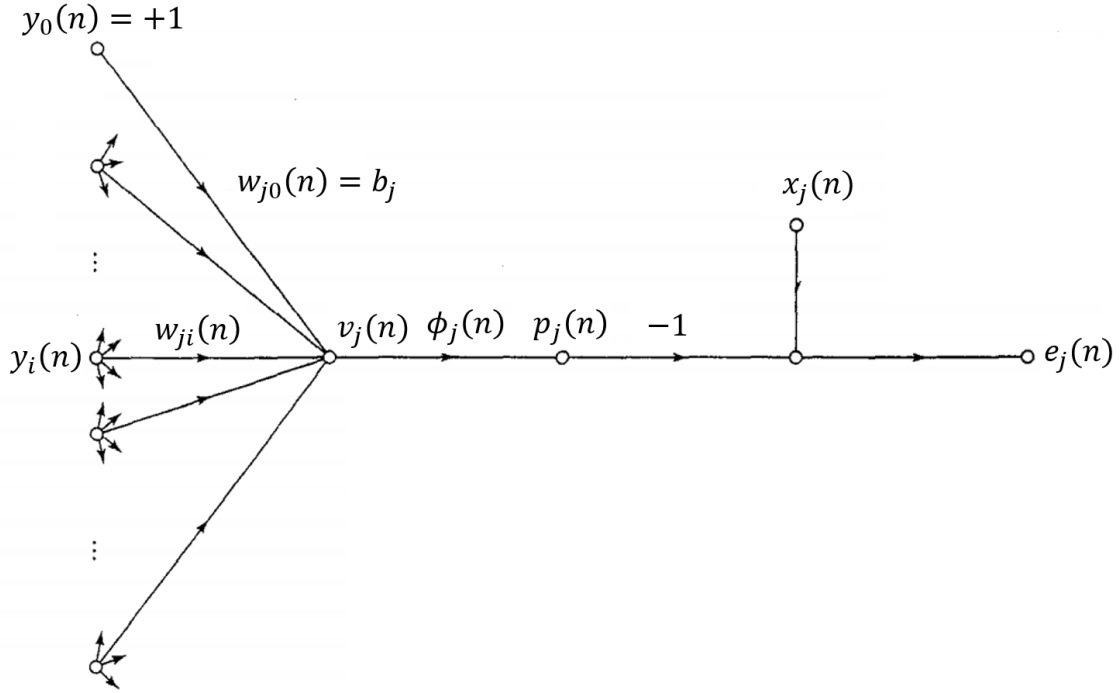


Figura 23: Diagrama de fluxo de sinal para o neurônio j , adaptado de Haykin (1999).

De forma similar ao algoritmo LMS, aplica-se uma correção os pesos sinápticos que é proporcional à estimativa do erro em cada um dos pesos. Essa estimativa é dada pela derivada parcial do erro de cada amostra do conjunto de treinamento, com respeito a cada um dos pesos $\partial E(n)/\partial w_{ij}(n)$. Como o erro é uma função composta, aplica-se a regra da cadeia, resultando em:

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial p_j(n)} \frac{\partial p_j(n)}{\partial u_j(n)} \frac{\partial u_j(n)}{\partial w_{ij}(n)} \quad (3.18)$$

O termo $\partial u_j(n)/\partial w_{ij}(n)$ representa um fator que indica uma direção de busca no espaço de pesos para o peso sináptico w_{ij} .

Ao diferenciar ambos os lados da Equação 3.13 em função de $y_j(n)$, tem-se:

$$\frac{\partial e_j(n)}{\partial u_j(n)} = -1 \quad (3.19)$$

Realizando o mesmo procedimento na Equação 3.14, porém, em relação a $e_j(n)$, obtém-se:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \quad (3.20)$$

Derivando agora a Equação 3.17 em função de $u_j(n)$,

$$\frac{\partial p_j}{\partial u_j(n)} = \phi'(u_j(n)) \quad (3.21)$$

e, por fim, a Equação 3.16, derivada com relação a $w_{ij}(n)$:

$$\frac{\partial u_j(n)}{\partial w_{ij}(n)} = y_i(n) \quad (3.22)$$

Substituindo as Equações 3.19, 3.20, 3.21 e 3.22 na equação obtida ao aplicar a regra da cadeia na função custo (Equação 3.18), obtém-se:

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = -e_j(n)\phi'(u_j(n))p_i(n) \quad (3.23)$$

De acordo com a regra delta, a correção aplicada em $w_{ij}(n)$ é dada por:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial u_j(n)} \quad (3.24)$$

$$= -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial p_j(n)} \frac{\partial p_j(n)}{\partial u_j(n)} \quad (3.25)$$

$$= e_j(n)\phi'(u_j(n)) \quad (3.26)$$

$$\Delta w_{ij}(n) = \eta \delta_j(n) p_i(n) \quad (3.27)$$

na qual η é a taxa de aprendizagem do algoritmo de *backpropagation* e o sinal negativo indica que a correção deve ocorrer no sentido contrário ao gradiente (descendente), sendo $\delta_j(n)$ o gradiente local, definido pela Equação 3.26 como o produto do sinal de erro correspondente $e_j(n)$ para o neurônio j e a derivada, $\phi'(u_j(n))$, da sua função de ativação (HAYKIN, 1999).

Partindo para uma generalização deste algoritmo — sua principal contribuição para o campo de redes neurais — o neurônio j , descrito anteriormente, pode estar localizado na camada de saída, ou em qualquer uma das camadas escondidas. O gradiente local ($\delta_j(n)$) depende da localização do neurônio para que a correção dos pesos seja feita de forma correta. Se ele está na *camada de saída*, o ajuste dos pesos é dado pela

Equação 3.26. Caso ele esteja em uma *camada oculta*, o fato de não se ter um sinal de erro previamente conhecido para ele, como para os neurônios da camada de saída, o gradiente local é dado por:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial p_j(n)} \frac{\partial p_j(n)}{\partial u_j(n)} \quad (3.28)$$

$$= -\frac{\partial E(n)}{\partial p_j(n)} \phi'(u_j(n)) \quad (3.29)$$

Como a derivada parcial na Equação 3.29 depende do sinal de erro (Equação 3.13) da camada subsequente (saída), há a representação através de um neurônio k ,

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad (3.30)$$

na qual k pertence ao conjunto de neurônios de saída C . O objetivo aqui é encontrar o sinal de erro na saída do neurônio j , que se encontra na camada escondida anterior à de saída. Portanto, a Equação 3.30 é derivada, gerando:

$$\frac{\partial E(n)}{\partial p_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial p_j(n)} \quad (3.31)$$

Aplicando a regra da cadeia para expandir a função composta (Equação 3.31), tem-se:

$$\frac{\partial E(n)}{\partial p_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial u_k(n)} \frac{\partial u_k(n)}{\partial p_j(n)} \quad (3.32)$$

sendo $e_k(n)$ o sinal de erro previamente conhecido na camada de saída (Equação 3.13). No entanto, sendo agora representado pelo neurônio de saída k . Portanto:

$$\frac{\partial e_k(n)}{\partial u_k(n)} = -\phi'(u_k(n)) \quad (3.33)$$

$$u_k(n) = \sum_{j=0}^m w_{kj}(n) p_j(n) \quad (3.34)$$

na qual m é o total de conexões de entrada no neurônio k , desconsiderando o bias. Ao derivar a Equação 3.34 em função de $y_j(n)$, tem-se:

$$\frac{\partial u_k(n)}{\partial p_j(n)} = w_{kj}(n) \quad (3.35)$$

Logo, ao substituir as Equações 3.33 e 3.35 na Equação 3.31, obtém-se:

$$\frac{\partial E(n)}{\partial p_j(n)} = - \sum_k e_k(n) \phi'(u_k(n)) w_{kj}(n) \quad (3.36)$$

$$= - \sum_k \delta_k(n) w_{kj}(n) \quad (3.37)$$

em que, na Equação 3.37, $\delta_k(n)$ é o gradiente local obtido para a camada de saída. Portanto, ao substituir essa expressão na Equação 3.29, é encontrada a fórmula do *backpropagation* para o gradiente local do neurônio j , na camada escondida anterior à de saída:

$$\delta_j(n) = \phi'(u_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (3.38)$$

Portanto, a correção do erro dos pesos que conectam um neurônio i — em uma camada anterior a j — ao neurônio j é dado por:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (3.39)$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) p_i(n) \quad (3.40)$$

sendo $p_i(n)$ o sinal de entrada do neurônio j . O termo $\phi_j'(u_j(n))$ é a derivada da função de ativação do neurônio j . Logo, de forma resumida, é computado o erro da camada de saída, na qual seus neurônios são visíveis, e é possível comparar o resultado da primeira premissa do algoritmo (propagação do estímulo no sentido direto) com os valores desejados. Logo, os pesos que se conectam diretamente a esses neurônios são ajustados devido à segunda premissa (retro-propagação do erro – Equação 3.27). As camadas escondidas, as quais não se pode medir diretamente a influência de cada neurônio, têm seus pesos ajustados de acordo com a retro-propagação desse sinal de erro gerado na saída, utilizando a Equação 3.39. E esse processo se repete até não haver mais camadas escondidas, chegando à entrada da rede (HAYKIN, 1999).

O algoritmo de *backpropagation* possui muitos parâmetros para se configurar antes

mesmo de iniciar um treinamento. O fato de não haver objetivamente uma maneira clara de ajustá-los na literatura é um fator desestimulante para a sua utilização, pois uma pequena alteração na combinação destes, muitas vezes, pode modificar bastante o andamento e o resultado do treinamento, fazendo a rede convergir para um mínimo local e/ou demorando muito tempo para convergir. São listados abaixo alguns dos principais cuidados que devem ser tomados ao treinar uma rede MLP (HAGAN et al., 1996; BRAGA; FERREIRA; LUDERMIR, 2007; LECUN et al., 2012):

- **o modo com que os pesos da rede são atualizados** pode acontecer, essencialmente, de duas maneiras: sequencial (*online* ou estocástico) e por lote (ou batelada). No primeiro, os pesos são ajustados a medida que as amostras são apresentadas para a rede, na qual a influência de um par de amostras entrada-saída é computado em uma iteração do algoritmo de treinamento. No modo por lote, os pesos são ajustados após todo o conjunto de amostras N ser passado como estímulo para a rede, cuja média da influência de cada um dos pares de amostra entrada-saída é utilizada para realizar o ajuste de uma só vez, ao fim desse período denominado época de treinamento. Esta última é geralmente mais estável, porém necessita de mais memória caso o conjunto de treinamento seja grande e redundante.
- **A necessidade de parar o treinamento segundo algum critério** é uma forma de prevenir que a rede demore muito tempo (épocas ou iterações) para convergir, e é uma forma de estabelecer o quanto a rede deve se aproximar de um modelo ideal, evitando problemas como *overtraining*, mas podendo causar outros como *undertraining*, sendo este último, resultado direto do ajuste de parada antes da rede chegar a um mínimo próximo ao global.
- ***Overtraining*** ocorre quando, após um certo ciclo de treinamento, a rede começa a apresentar uma performance ruim ao serem apresentadas novas amostras, até então, desconhecidas para a mesma. Este fato ocorre devido à rede acabar por memorizar os dados e o ruído, que compõem o conjunto de treinamento, reduzindo sua capacidade de generalização, conhecido como *overfitting*. *Overfitting* pode ocorrer também quando a rede é muito mais complexa que o próprio modelo que a mesma viria a representar, ou seja, possuindo um número de neurônios — consequentemente, pesos sinápticos — extremamente alto. Caso a rede seja simples demais para o modelo a que deve representar, ocorrerá um fenômeno chamado de *underfitting*. Uma solução para esse problema é dividir o conjunto

de treinamento em dois sub-conjuntos: treino e validação. O primeiro é utilizado para o ajuste dos pesos sinápticos normalmente, já o segundo, será utilizado para testar a rede entre os ciclos ou épocas de treinamento apenas. Este processo torna possível visualizar quando a rede começa a apresentar características de *overtraining* e está ilustrado na Figura 24.

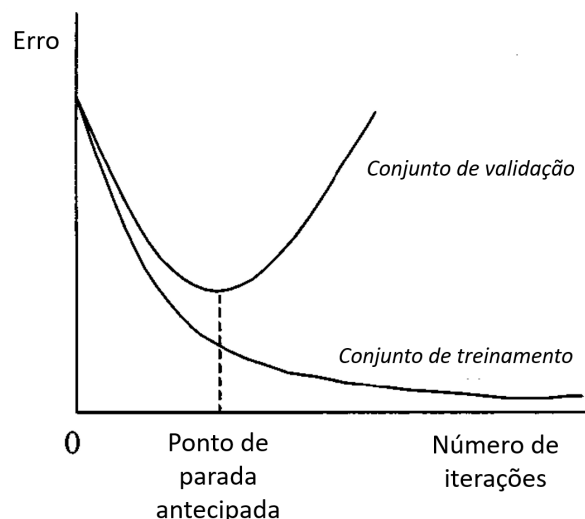


Figura 24: *Overtraining*.

- **A capacidade de generalização** de uma rede neural é, em grande parte, o principal objetivo do treinamento. Ela permite que a rede faça estimações corretas sobre padrões nunca antes apresentados ou com ruído. Isto torna o sistema robusto a falhas, quando o processo de treinamento é bem executado.
- **Evitar que a solução final fique presa a um mínimo local** na superfície de erro, mesmo que ele apresente uma solução estável. O *backpropagation* apresenta uma série de deficiências que o torna lento para convergir. Uma das soluções é utilizar uma taxa de aprendizagem adaptativa (decrecente com o tempo ou outro critério). São alternativas também alterar o número de camadas intermediárias (escondidas) ou neurônios nestas camadas. Em McClelland (1986), sugere-se a utilização do termo chamado de *momentum*, visando acelerar o processo de treinamento e evitar as regiões mais planas da superfície de erro.
- **O número de neurônios e camadas** a serem incluídos no modelo é uma das variáveis que mais adicionam dificuldade na utilização de uma rede MLP. Esse parâmetro varia de acordo com o problema e, como a rede utiliza de alta conectividade para realizar generalização, alterar estes parâmetros torna a busca por uma combinação ideal dos mesmos exaustiva e sem uma solução genérica.

O teorema da aproximação universal diz que apenas uma camada escondida em uma rede MLP é o suficiente para representar uma aproximação de qualquer função contínua, dado um conjunto de treinamento com uma relação entrada/saída apropriada (GALLANT; WHITE, 1988). Pontos de partida são sugeridos para a escolha de um número ótimo desses parâmetros por vários autores. Atualmente, são propostos métodos que incluem o crescimento do número de neurônios (*growing*), eliminação de pesos em uma rede inicializada com um número excessivo de neurônios (*pruning*), além de abordagens com vieses estatístico baseados no critério da informação de Akaike (MURATA; YOSHIZAWA; AMARI, 1994; HAYKIN, 1996; HU; HWANG, 2001). Outros métodos abordados pela literatura são encontrados em Sheela (2013).

O algoritmo *backpropagation* torna-se muito lento para problemas que envolvem redes maiores ou que são muito complexos. Desde a sua concepção, suas deficiências foram sendo contornadas com variações do próprio algoritmo. Dentre estas, as mais utilizadas são o *backpropagation* com *momentum*, *Quickprop*, Levenberg-Marquadt, *momentum* de segunda ordem, Newton e Resilient *Backpropagation*. Mais detalhes sobre o funcionamento destes algoritmos são encontrados nas referências (MCCLELLAND; RUMELHART; HINTON, 1986; FAHLMAN, 1988; HAGAN; MENHAJ, 1994; PEARLMUTTER, 1991; BATTITI, 1992; RIEDMILLER, 1994). Há uma outra forma de aprendizado supervisionado conhecida como aprendizado por reforço, explicado de forma mais aprofundada em Haykin (1999) e Braga; Ferreira e Ludemir (2007).

3.4.2 APRENDIZADO NÃO-SUPERVISIONADO

Como o próprio nome sugere, a forma de aprendizado não-supervisionado é aquele que não depende de um supervisor (professor) que acompanhe o processo em si. Devido ao fato do aprendizado supervisionado ser mais facilmente relacionado com a forma que os seres humanos aprendem, a forma de aprendizagem não-supervisionada é também inspirada em sistemas biológicos, como por exemplo, os estágios iniciais dos sistemas de visão e audição, nos quais para estes organismos apenas os padrões de entrada estão disponíveis, ou seja, apenas os estímulos do ambiente, não possuindo pares entrada-saída, como no modo de aprendizado anteriormente detalhado (BRAGA; FERREIRA; LUDERMIR, 2007). Um modelo básico de aprendizagem não-supervisionada é ilustrado na Figura 25.

Existem várias outras formas de aprendizagem não-supervisionada, como o apren-

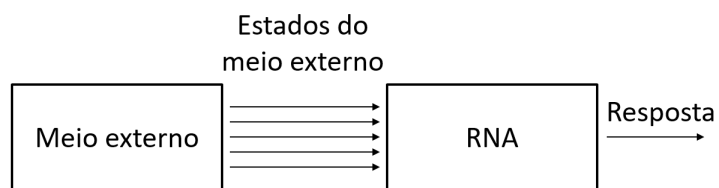


Figura 25: Conceito de um treinamento não-supervisionado.

dizado Hebbiano, o modelo de Linsker e o aprendizado por competição que, por não fazerem parte do escopo desta dissertação, não serão abordados com detalhes. Estes e outros métodos são detalhados em maior profundidade nas referências (HAYKIN, 1999; BRAGA; FERREIRA; LUDERMIR, 2007).

4 MÉTODO PROPOSTO E BASE DE DADOS

Neste capítulo, é feita uma revisão de técnicas aplicadas para atenuar os efeitos do empilhamento de sinais, tanto em calorimetria, quanto em outras áreas. São pontuados os métodos com uma abordagem linear, que visam a simplicidade de implementação, assim como não-lineares, que buscam alto desempenho do sistema implementado. Após, serão apresentados o método proposto juntamente com base de dados utilizada.

4.1 ESTIMAÇÃO DE ENERGIA UTILIZANDO FILTROS FIR

As técnicas propostas em Duarte (2015) baseiam-se em interpretar os canais de leitura do Tilecal como uma convolução entre a função delta de Dirac, que representaria a energia a ser estimada em cada colisão $x[n]$, com o pulso característico do Tilecal $s[n]$, somado a um ruído branco gaussiano $q[n]$, advindo da eletrônica de leitura. Esse modelo resulta em $r[n]$, que é um sinal típico de um dos canais do Tilecal. Na Figura 26 é possível observar essa relação através de um diagrama de blocos no domínio z , no qual um novo bloco $G(z)$ é incluído após $R(z)$. Este bloco tem o propósito de realizar a deconvolução do sistema $S(z)$, recuperando uma estimativa do sinal de entrada, denominada por de $\hat{X}(z)$.

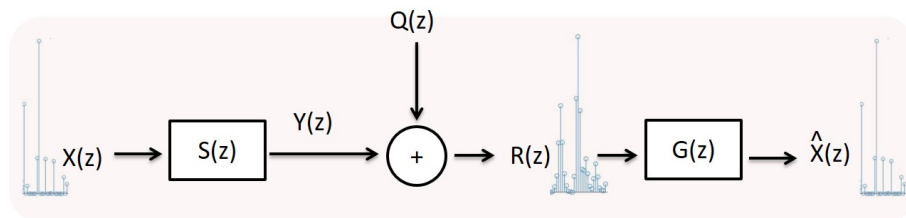


Figura 26: Diagrama de blocos no domínio z que representa a modelagem do Tilecal como um sistema linear.

Com o intuito de atender os requisitos de latência do primeiro nível de *trigger* na Fase-II, as técnicas baseadas em filtros FIR são adequadas pois são de simples implementação. Foram abordados dois métodos: um determinístico e outro adaptativo, detalhados nas Seções 4.1.1 e 4.1.2.

4.1.1 FIR DETERMINÍSTICO

Neste método, a partir do conhecimento prévio da resposta do sistema (canal) ao impulso unitário (resultando no pulso de 7 amostras, como ilustrado na Figura 10), projeta-se um filtro inverso $G_D(z)$, descrito pela Equação 4.1, na qual $S(z)$ é obtido aplicando-se a transformada z no sinal determinístico do Tilecal, amostrado a 40 MHz.

$$G_D(z) = \frac{1}{S(z)} \quad (4.1)$$

Como esse sistema é inverso de um sistema de resposta finita (7 amostras), logo, ele apresenta pólos e é classificado como um filtro IIR (do inglês, *Infinite Impulse Response*) (MITRA, 1998). Para se obter um filtro estável e uma aproximação por resposta finita (filtro FIR), alguns procedimentos devem ser seguidos (DUARTE, 2016):

1. o sistema inverso gerado G_D deve ser separado em duas partes. Uma deve conter os pólos que estejam fora do círculo unitário ($G_{D1}(z)$), e a outra, os pólos que estejam dentro do círculo unitário ($G_{D2}(z)$), resultando, respectivamente, em um filtro instável e outro estável em cascata. Para tal, a condição $G_D(z) = G_{D1}(z).G_{D2}(z)$ deve ser satisfeita.
2. Inverte-se a região de convergência do filtro causal instável, neste caso $G_{D1}(z)$, tornando estável, porém, não-causal. A mesma deve incluir também o círculo unitário, resultando em $G'_{D1}(z)$.
3. Truncam-se as respostas dos filtros IIR $G'_{D1}(z)$ e $G_{D2}(z)$, com o intuito de produzir filtros FIR com resposta aproximada, resultando em $G''_{D1}(z)$ e $G'_{D2}(z)$.
4. Pelo fato de $G''_{D1}(z)$ ser não-causal, ao atrasar a sua resposta, ele se torna um sistema causal, gerando $G'''_{D1}(z)$.
5. Ao realizar a convolução da resposta ao impulso dos filtros $G'''_{D1}(z)$ e $G'_{D2}(z)$, tem-se um filtro FIR $F(z)$ que realiza a deconvolução aproximada de $G_D(z)$.

4.1.2 FIR UTILIZANDO LEAST SQUARES

Nesta abordagem, parte-se do princípio que o calorímetro é um sistema desconhecido. Baseando-se em simulações de Monte Carlo, na qual toda a estatística da física que envolve o Tilecal é considerada, conhecendo-se os valores de $x[n]$ e, observando a

resposta do canal (calorímetro) a essas entradas impulsivas $r[n]$, consegue-se projetar um filtro FIR, que é a deconvolução aproximada do sistema desconhecido. Para isso, utiliza-se um critério de otimização, como o erro médio quadrático. Os pesos são ajustados de modo a minimizar o erro entre $v[n]$, que é a saída do filtro FIR e $x[n]$, que é o valor desejado (DUARTE, 2015; KAY, 1998). Na Figura 27 ilustra-se esse processo.

Ao utilizar o critério citado acima, a função custo do sistema é dada por:

$$J(\hat{\mathbf{g}}) = \sum_{n=0}^{M-1} (x[n] - v[n])^2 \quad (4.2)$$

sendo M o total de amostras disponíveis e, na forma matricial, \mathbf{x} representa o número de amostras consecutivas do sinal desejado. A matriz \mathbf{v} depende linearmente de \mathbf{r} e $\hat{\mathbf{g}}$. Representando a Equação 4.2 na forma matricial, tem-se:

$$J(\hat{\mathbf{g}}) = (\mathbf{x} - \mathbf{R}\hat{\mathbf{g}})^T (\mathbf{x} - \mathbf{R}\hat{\mathbf{g}}) \quad (4.3)$$

na qual \mathbf{R} é a matriz de observação dos dados de saída do canal, que é formada por amostras deslocadas de r . Obtendo-se a derivada de J em função dos parâmetros de $\hat{\mathbf{g}}$, tem-se:

$$\frac{\partial J(\hat{\mathbf{g}})}{\partial \hat{\mathbf{g}}} = -2\mathbf{R}\mathbf{x} + 2\mathbf{R}\mathbf{R}^T \hat{\mathbf{g}} \quad (4.4)$$

Ao igualar a Equação 4.4 a zero e isolar $\hat{\mathbf{g}}$, assumindo que $\mathbf{R}\mathbf{R}^T$ seja invertível, $\hat{\mathbf{g}}$ pode ser estimado de acordo com a Equação 4.5.

$$\hat{\mathbf{g}} = (\mathbf{R}\mathbf{R}^T)^{-1} \mathbf{R}\mathbf{x} \quad (4.5)$$

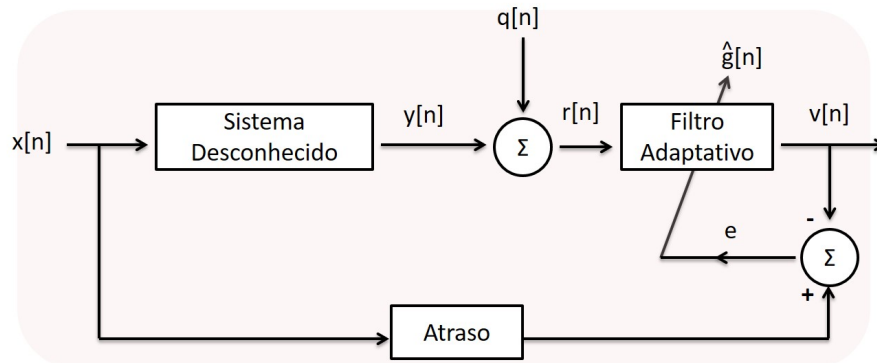


Figura 27: Sistema inverso utilizando filtro adaptativo.

O vetor de coeficientes $\hat{\mathbf{g}}$ corresponde à estimação da resposta ao impulso de um sistema inverso ao canal desconhecido, podendo ser interpretado como um filtro FIR que realiza a deconvolução de tal sistema (DUARTE, 2015; KAY, 1998).

Na Figura 28 mostra-se um comparativo entre o desempenho dos métodos propostos por Duarte (2015) para a Fase-II e o Filtro Casado, atualmente implementado no primeiro nível de *trigger*. As técnicas utilizando filtros FIR mostraram bom desempenho em comparação com o Filtro Casado (DUARTE, 2015).

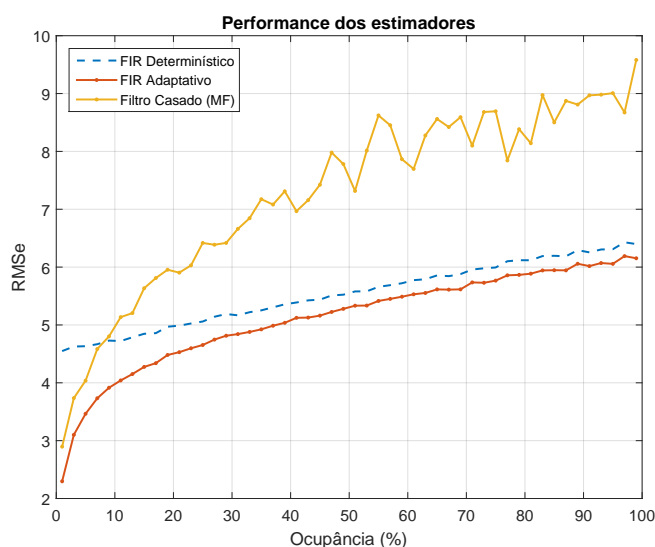


Figura 28: Comparação entre os métodos propostos para a Fase-II e o atualmente implementado.

Os métodos propostos por Duarte (2015) atendem os principais requisitos do primeiro nível de *trigger*, sendo de simples implementação (composto por somas e produtos apenas), além de possuírem baixa latência (12 a 13 *bunch-crossing*). No entanto, encontram-se alguns problemas:

- os sinais laterais à amostra que se deseja estimar sua amplitude, provenientes de outros *bunch-crossings* (anterior e/ou posterior), são caracterizados como um ruído que distorce o valor real da amplitude. Como todas as amostras dos sinais do Tilecal são positivas, este ruído tem uma distribuição assimétrica, sendo então, não-gaussiana. Logo, estimadores lineares são ótimos apenas quando o ruído é gaussiano, fazendo com que, necessariamente, quando o ruído é não-gaussiano, o estimador ótimo deva ser não-linear (KAY, 1993).
- As partículas têm tempos de voos — entre o ponto de colisão e o detector — diferentes, ocasionando em mais um parâmetro não determinístico que carac-

teriza o sinal de leitura. Ao utilizar um método linear para estimar a fase, é necessário fazer uma aproximação de primeira ordem do sinal por série Taylor, desconsiderando os termos polinomiais não-lineares de maior ordem (PERALVA, 2015).

- Fazer uma aproximação do filtro IIR por um filtro FIR pode gerar filtros com ordem muito grande, criando um atraso também alto.

Ao reforçar estes pontos, este trabalho propõe a implementação de uma filtragem não-linear baseada em redes neurais artificiais *feed-forward*, visando realizar a reconstrução da energia de forma mais eficiente.

4.2 FILTRAGEM INVERSA UTILIZANDO REDES NEURAS FEED-FORWARD

A tarefa básica de um sistema de comunicação é transmitir informação de um emissor para um receptor, através de um meio. O meio de transmissão inclui parâmetros à informação enviada, sendo um deles, o canal. Portanto, o sinal a ser transmitido passa por um canal de comunicação antes de chegar ao receptor. Pode-se dizer também que o sinal transmitido convolve com este canal. Quando o canal é de natureza dispersiva, ou seja, espalhando temporalmente o dado, a resposta de transmissões sucessivas pode causar interferências entre esses sinais, chamado de interferência inter-simbólica — em telecomunicações — ou empilhamento de sinais (*pile-up*) — em processamento de sinais (RAGHAVENDRA; TRIPATHY, 2006). É representado na Figura 15 um exemplo de sistema de transmissão de sinais por um canal de comunicação.

O efeito de *pile-up* causa problemas para o receptor, que deve ser projetado de forma a reverter ou amenizar seus efeitos. Muitas vezes, deseja-se obter um modelo matemático do canal para poder compensar sua influência. A qualquer técnica de processamento de sinais aplicada para realizar essa compensação, dá-se o nome de equalização ou deconvolução. Quando o modelo do sistema é desconhecido ou muito complexo, uma estimativa é feita de forma a projetar o equalizador ou filtro inverso. A aplicação de filtros inversos, também chamado de deconvolução, pode ser realizada de diversas formas, como a utilização de algoritmos adaptativos ou determinísticos, baseados em filtragem linear ou não-linear (HAYKIN, 2001; RAGHAVENDRA; TRIPATHY, 2006).

Quanto a equalização linear, os métodos mais comuns são a utilização de filtros li-

neares baseados na resposta ao impulso do canal, além de algoritmos adaptativos, como LMS, RLS (do inglês, *recursive least square*), gradiente RLS, dentre outros (BURSE; YADAV; SHRIVASTAVA, 2010). A utilização de filtros lineares para realizar a deconvolução aproximada de sistemas, que apresentam características não-lineares intrínsecas, pode, muitas vezes, ser uma escolha viável no que tange sua complexidade de implementação, considerada baixa. No entanto, essa escolha é feita causando uma redução do desempenho da operação inversa. Nestes casos, a abordagem em que se utiliza técnicas de filtragem não-linear torna-se uma alternativa a ser explorada, como a utilização de redes neurais artificiais.

Nos últimos anos, a aplicação de redes neurais artificiais e suas várias topologias, em filtragem inversa não-linear, é explorada largamente em sistemas de telecomunicações, como para equalização de redes *wireless*. Em Burse, Yadav e Shrivastava (2010), é feito um comparativo de topologias como MLP *feedforward*, redes recursivas, redes de base radial (RBF) e estruturas derivadas de redes FLANN (do inglês, *functional link neural network*). Neste trabalho, são comparados aspectos problemáticos de filtragem não-linear, como sua complexidade de implementação, assim como o tempo de treinamento (BURSE; YADAV; SHRIVASTAVA, 2010). Redes MLP *feedforward* foram aplicadas também em problemas de filtragem inversa em sistemas de processamento de sinais em rádio-astronomia, além de equalização de distorções acústicas (GLANZ; MILLER, 1989; CHANG; LIN; YEH, 1994).

Em calorimetria para Física de Altas Energias, aplicações de métodos filtragem inversa encontrados na literatura são vistos em Duarte (2015), como o Filtro Casado, assim como o estudo de outras técnicas de deconvolução linear baseadas em filtros FIR. No entanto, a utilização de redes neurais artificiais para deconvolução não linear nesse ambiente não foi encontrada na literatura, sendo, então, a proposta central deste trabalho.

4.3 ESTIMAÇÃO NÃO-LINEAR E ONLINE DE ENERGIA

No caso de sinais no tempo discreto, o processamento digital de sinais pode ser implementado de maneira *offline* ou em tempo real (*online*) (MITRA, 1998). Um filtro FIR aplicado para estimação *online* é ilustrado na Figura 29. Ele é composto de uma série de registradores de deslocamento, que geram um atraso do sinal do canal de comunicação (calorímetro hadrônico). A esta sequência de sinais deslocados temporalmente dá-se o nome de janela de observação ou simplesmente atrasos do filtro, que têm

largura m . Cada um dos valores da janela (y_m) é multiplicado por uma constante (c_m), que são os coeficientes do filtro. Sabe-se que $y[n]$ é a amostra de saída do calorímetro referente a $x[n]$, sendo esta última, o valor e energia real (ou teórico) a ser recuperado pelo sistema de estimação. Após os resultados do filtro serem combinados linearmente, obtém-se o valor estimado da energia referente à amostra central da janela ($y[n - \frac{m}{2}]$), antes dela sofrer as distorções inerentes ao calorímetro ($x[n - \frac{m}{2}]$), representado por $\hat{x}[n - \frac{m}{2}]$. Devido à adição de uma janela de atrasos, necessária à estimação do elemento central, esse tipo de filtro causa uma latência de $\frac{m}{2}$ amostras em relação aos dados de saída do calorímetro. Vale ressaltar que, para valores de m ímpares, o elemento central à janela é dado pela amostra $\frac{(m+1)}{2}$.

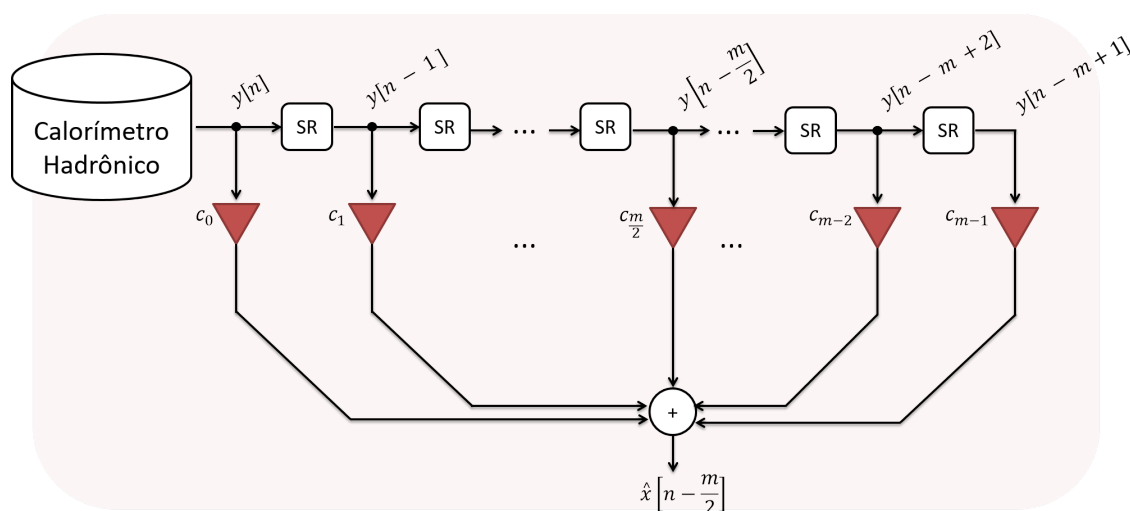


Figura 29: Filtro FIR em aplicação de tempo real com janela par.

O propósito deste trabalho é substituir o filtro FIR por uma rede neural *feedforward perceptron* multi-camadas, até este momento, de arquitetura desconhecida, posteriormente implementada em hardware, sendo ilustrada na Figura 30. Como pode ser observado, ela irá substituir todos os coeficientes do filtro FIR, no entanto, também fará o uso de uma janela de atrasos para executar o procedimento de estimação da energia real referente à amostra central da mesma. A metodologia utilizada na construção da rede para um ambiente de alta luminosidade e a seleção de cada um de seus principais parâmetros de treinamento, serão discutidos na Seção 4.4.

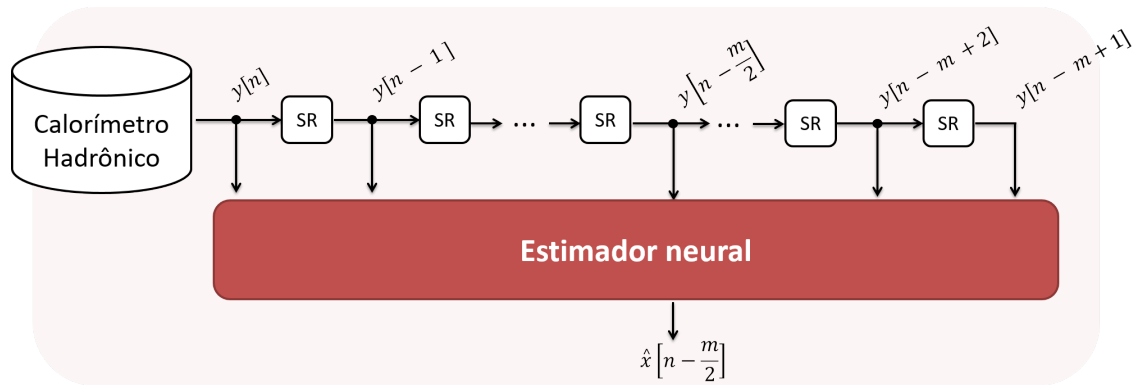


Figura 30: Rede neural *feedforward* em aplicação de tempo real com janela par.

4.4 METODOLOGIA UTILIZADA

O treinamento da rede foi realizado de maneira *offline*, permitindo que o desempenho de cada topologia e demais parâmetros sejam avaliadas detalhadamente, antes de sua implementação em hardware dedicado (FPGA). Os dados das simulações foram gerados visando criar condições operacionais para células localizadas em diferentes regiões do Tilecal. Essas condições serão reproduzidas através de sinais com ocupâncias distintas imersos em ruído gaussiano. Como já foi comentado no Capítulo 1, o ruído de *pile-up* se intensifica enquanto a ocupância do sinal aumenta. Serão realizadas simulações utilizando *Toy Monte Carlo*, que geram sinais com características similares às encontradas para o Tilecal (CHAPMAN, 2011). A simulação de *Toy Monte Carlo* foi realizada da seguinte forma:

- de acordo com a ocupância do sinal que se deseja criar, é realizado um sorteio de posições (*bunch-crossings*), nas quais houveram depósito de energia. Se é desejado criar um sinal com 20% de ocupância, por exemplo, isso indica que, em média, ocorreria 1 depósito de energia nesta célula a cada 5 *bunch-crossings*.
- A energia por *bunch-crossing* é obtida através de um sorteio utilizando uma distribuição exponencial de valor médio 30 unidades de conversão do ADC. Este valor se origina de um *fitting* exponencial feito em distribuições de *pile-up* de simulações de Monte Carlo do Tilecal.
- Após os sorteios dos *bunch-crossings*, nos quais teve depósito e a amplitude de cada um destes valores de energia, é realizado outro sorteio para simular o desvio de fase do pulso. Esta condição está relacionada com o tempo de voo das par-

tículas geradas em cada colisão, e foi utilizada uma distribuição uniforme de ± 5 ns.

- Logo, para cada amostra de energia gerada em cada *bunch-crossing*, é sorteado um pulso característico normalizado (resposta ao impulso unitário do Tilecal), constituído de 7 amostras, com um desvio de fase específico. É realizada a operação de convolução de cada uma dessas amostras, simulando a característica dispersiva do calorímetro.
- Ao final, são geradas N amostras de ruído branco gaussiano, com desvio padrão σ e média zero. Estas amostras de ruído visam reproduzir o ruído eletrônico existente no Tilecal em um de seus canais a 40 MHz. As amostras digitalizadas deste ruído são adicionadas ao sinal do item anterior.

O período de amostragem do ADC, por ser sincronizado com o cruzamento de feixes do LHC, é de 25 ns. Isso implica que, como o desvio de fase utilizado compreende-se numa faixa de ± 5 ns, por ser menor que este período, o ADC irá converter valores de energia em que a amostra central pode não representar a amplitude máxima do pulso. Este efeito é representado na Figura 31, na qual pode ser observado que, para o mesmo pulso, são obtidas amostras com valores diferentes, criando uma condição que aumenta o erro de medição. Os dados gerados a partir das descrições acima são utilizados para o treinamento das redes neurais e, parâmetros como o tamanho do conjunto de treinamento e a intensidade do ruído de fundo eletrônico são discutidos na próxima seção.

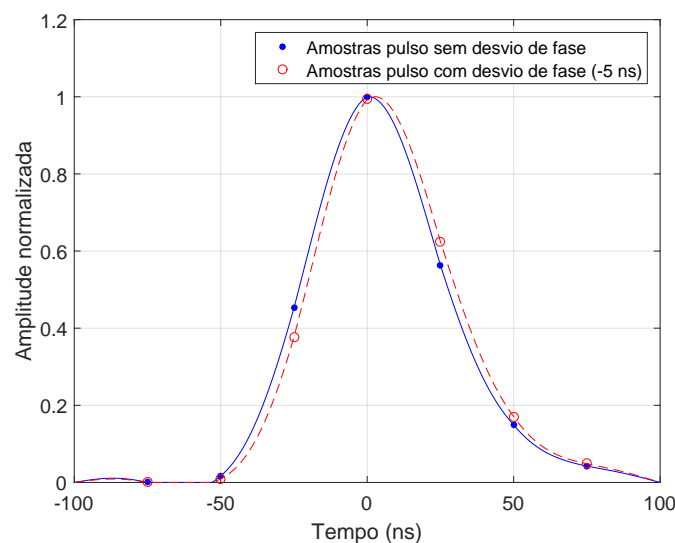


Figura 31: Desvio de fase do pulso característico.

Para criar os dados utilizados no treinamento da rede estimadora, foi fixado o ruído eletrônico com um desvio padrão (σ) de 1,4 ADC, que é um valor típico encontrado em canais de leitura do Tilecal. A intensidade do ruído eletrônico implica diretamente na relação entre o sinal e o ruído, conhecida como SNR, que também pode ser definida como o recíproco do coeficiente de variação (EVERITT, 1998):

$$\text{SNR} = \frac{\mu_{\text{sinal}}}{\sigma_{\text{ruído}}} \quad (4.6)$$

sendo $\sigma_{\text{ruído}}$ o desvio padrão do ruído branco gaussiano, e μ_{sinal} o valor médio da distribuição exponencial do sinal de uma célula do Tilecal. Analisando a Equação 4.6, é possível observar que quanto mais ruidoso é o canal, menor é o valor da SNR. Logo, a relação sinal-ruído é maximizada quando $\sigma \rightarrow 0$. O valor da SNR utilizado em todos os bancos de dados de treino, teste e validação é de 21,43.

4.4.1 REDE ESTIMADORA

Os demais parâmetros que envolvem o treinamento desta rede, visando se aproximar de uma configuração ótima para o problema apresentado no Capítulo 2, são testados separadamente, buscando observar o impacto de cada um deles no desempenho do estimador neural. Logo, a metodologia utilizada neste trabalho parte do pressuposto que os parâmetros que estiverem fixos em uma dada simulação irão influenciar pouco ou nada no ajuste do parâmetro livre. Esta observação é importante, pois, como já foi explicitado anteriormente, o processo de ajuste dos parâmetros de treinamento de uma rede neural pode ser visto como um problema de otimização. Por isso, neste caso, o ajuste dos parâmetros de treinamento para obter uma rede neural com o melhor desempenho possível torna-se um problema multi-dimensional, dificultando a análise dos parâmetros ótimos de forma conjunta. Então, foram realizadas cinco simulações para encontrar a melhor topologia e os pesos sinápticos da rede estimadora:

1. Determinação do número de inicializações (método estocástico).
2. Influência do tamanho do conjunto de treinamento.
3. Ajuste dos critérios de parada.
4. Influência da ocupância do sinal no treinamento.
5. Número de atrasos de entrada e neurônios na camada oculta.

A figura de mérito utilizada para a rede estimadora é o erro RMS (do inglês, *root mean square error*), que é a raiz quadrada do erro médio quadrático (MSE), calculado de acordo com a Equação 4.7:

$$RMSe = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \hat{X}_i)^2} \quad (4.7)$$

sendo N o número de amostras do sinal, X_i um valor de saída conhecido desejado, e \hat{X}_i o valor de saída real, estimado, sendo neste caso, aquele fornecido pela rede após o ajuste dos pesos sinápticos. Também foi avaliada a capacidade da rede de detectar o sinal desejado, combinando esta detecção com a estimação, visando aumentar seu desempenho.

4.4.2 DETECÇÃO

Ao ajustar os pesos sinápticos de uma rede neural PMC com um algoritmo de treinamento supervisionado com o intuito de detectar um sinal em meio a um ruído, esta aproxima-se de um detector ótimo. Devido ao fato do detector de máxima verossimilhança (ótimo) necessitar do conhecimento das funções de densidade de probabilidade dos dados *a priori*, sua implementação em sistemas práticos torna-se extremamente limitada. Logo, a utilização de redes neurais é encorajada, dado que não se faz necessário nenhum conhecimento prévio dos parâmetros estatísticos dos dados, para que este sistema, após ser ajustado corretamente, gere resultados próximos do ótimo (WATTERSON, 1990; TREES, 2001; PERALVA et al., 2012).

Neste trabalho são abordadas duas formas de realizar a detecção de um sinal contendo amostras de ruído: detecção por patamar na rede estimadora e rede neural classificadora. No método de detecção por patamar, o valor de saída do estimador é comparado a uma constante. Caso esta amostra seja maior, o valor estimado é aceito como sendo de um sinal e, caso seja menor ou igual, este sinal estimado é classificado como a estimação de uma amostra de ruído, sendo descartada, forçando a saída a zero. Ao utilizar uma rede neural especializada em classificação, a mesma trabalha em paralelo à rede detectora, classificando a amostra em análise na janela de atrasos como sinal ou ruído. Na saída dessa rede, é utilizado um sistema de detecção por patamar similar ao descrito anteriormente, forçando a saída do sistema de estimação a zero caso um sinal de ruído seja detectado. Ambos sistemas são ilustrados na Figura 32.

Para realizar a diferenciação de um sinal de entrada entre sinal real e ruído eletrô-

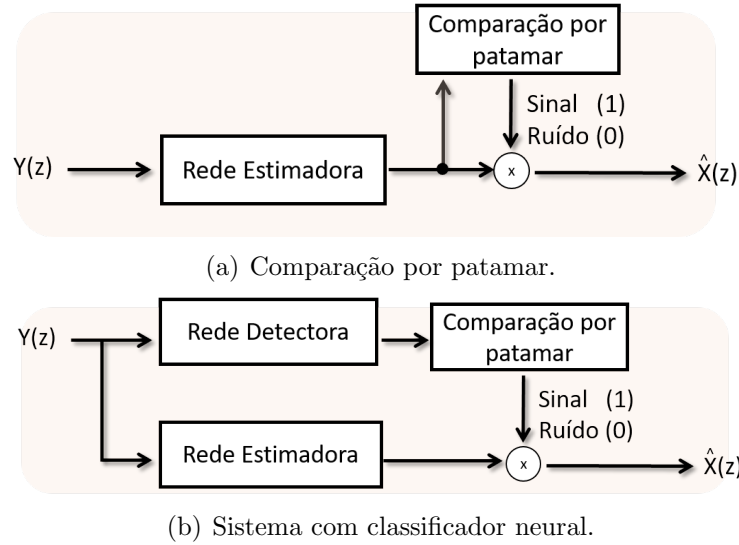


Figura 32: Esquemas de detecção de sinal combinados com a estimação de energia.

nico, primeiramente, deve-se alocá-los em classes distintas. Os alvos da rede durante o treinamento não serão mais o valor de energia ideal, como utilizado na rede estimadora, e sim, valores referentes às classes as quais os dados de entrada pertencem. Logo, têm-se, em um treinamento supervisionado, pares de dados $\{\mathbf{y}, x\}$, no qual \mathbf{y} é composto das amostras dos sinais simulados de saída de um canal do Tilecal, observados em uma janela de largura m . Já x é composto dos valores que representam as classes, sendo $+1$ para *senal* (qualquer valor de energia depositada na célula no *bunch crossing* referente ao centro da janela) e -1 para *ruído* (quando não houve depósito de energia na célula no *bunch crossing* referente ao centro da janela). As redes treinadas para realizar a classificação têm diferenças quanto à arquitetura quando comparadas às redes utilizadas para estimação. Como o objetivo aqui é dar como resposta $+1$ ou -1 , a função de ativação do neurônio de saída, assim como os neurônios da camada escondida, será a tangente hiperbólica.

Como o neurônio de saída deve possuir uma função de ativação diferenciável (tangente hiperbólica) para que o algoritmo de *backpropagation* possa realizar o ajuste dos pesos sinápticos, o resultado de saída da rede não fica confinado apenas aos valores das classes, mas compreendidos entre $[-1, +1]$. A rede após o treinamento não dá exatamente a qual classe aquele sinal pertence, e sim, uma distribuição de valores, os quais a maioria deles está concentrada nas saturações da função de ativação. Para inferir a qual classe o sinal pertence, o valor de saída deve ser comparado a um patamar de decisão, ajustado em um ponto que maximize a performance de classificação da rede. Uma forma de avaliar um classificador é, após comparar a um patamar de decisão, através da observação das suas taxas de positivos verdadeiros (do inglês, *true positive* - TP)

e falsos positivos (do inglês, *false positive* - FP). A taxa de TP é a porcentagem das amostras de entrada que foram corretamente classificadas de acordo com a classe-alvo analisada. Já a taxa de FP é a porcentagem de amostras da classe que não é o alvo analisado, neste caso, as amostras de ruído classificadas como sendo um sinal-alvo. A essa taxa, em sistemas de detecção, dá-se o nome também de falso alarme (do inglês, *false alarm* - FA). Em determinadas aplicações, pode ser desejado que o classificador trabalhe em pontos de operação distintos, o que modifica a relação entre TP e FA. Ao variar o patamar de decisão em uma faixa de valores entre -1 e $+1$, para cada valor do patamar, é encontrado um conjunto de valores (TP,FA) diferente, resultando em pontos de operação distintos. O conjunto de valores de TP e FA para toda a faixa de decisores é ilustrado em um gráfico, como na Figura 33, fornecendo uma análise ampla sobre o classificador em questão, chamada de curva ROC (do inglês, *receiver operating characteristics*) (WOODS; BOWYER, 1997).

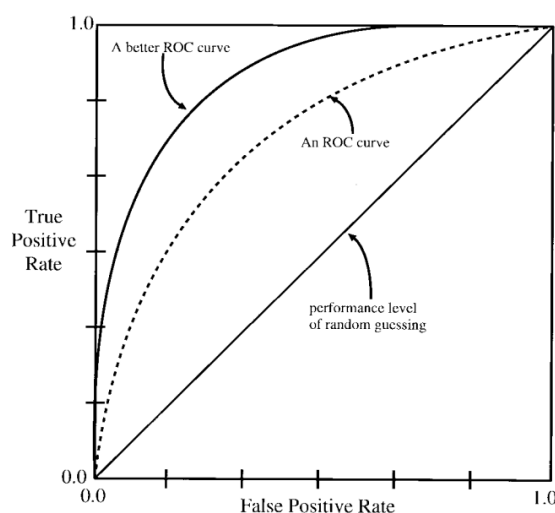


Figura 33: Curva ROC (WOODS; BOWYER, 1997).

Neste tipo de gráfico, observa-se que a linha diagonal é de um sistema o qual não possui capacidade de classificação, retornando valores aleatórios para cada amostra testada. Agora, quanto mais a curva se aproxima do canto superior esquerdo, indicando alta taxa de verdadeiros positivos e baixa taxa de falsos positivos (falso alarme), melhor é o classificador, segundo o critério da curva ROC. A figura de mérito utilizada neste trabalho para determinar o número de neurônios da camada oculta é o produto SP, definido pela Equação 4.8 (FILHO; SEIXAS; CALOBA, 2008),

$$SP = \frac{(\text{efic}_{\text{classe}_1} + \text{efic}_{\text{classe}_2}) \times (\text{efic}_{\text{classe}_1} \times \text{efic}_{\text{classe}_2})}{2} \quad (4.8)$$

na qual as eficiências de classificação atingem no máximo a unidade, e o produto SP,

consequentemente, ao ser dividido por 2, também alcança no máximo uma unidade. Isso ocorre quando a eficiência de classificação é máxima, tanto para a classe 1, quanto para a classe 2, sendo um bom indicador para avaliar a performance da rede classificadora, logo que, o patamar decisor será escolhido em uma posição que maximize o produto SP.

5 SIMULAÇÕES, RESULTADOS E DISCUSSÕES

Neste capítulo serão abordados os resultados dos treinamentos das redes neurais *feedforward* com diferentes configurações, como discutido no capítulo anterior, buscando aquela que alie boa performance com baixa complexidade. Após ser definida, a melhor topologia é então comparada com os métodos lineares detalhados nos capítulos anteriores.

5.1 DETERMINANDO O NÚMERO DE INICIALIZAÇÕES

O processo de treinamento de uma rede neural é, muitas vezes, computacionalmente custoso, podendo demorar um tempo considerável para convergir, dependendo do problema. Mas, nem sempre, a convergência é uma garantia de bom desempenho da rede. Não há métodos analíticos aplicados para encontrar um conjunto-solução de pesos sinápticos ótimos. Sendo assim, soluções baseadas no gradiente descendente, como algoritmo de *backpropagation* e suas derivações, incluindo o algoritmo de Levenberg-Marquardt, são aplicadas com sucesso neste campo. Por se tratarem de procedimentos de otimização iterativos, sua performance é fortemente dependente das condições iniciais do problema (ERDOGMUS et al., 2005).

Como o objetivo final deste problema de otimização (treinamento) é o ajuste dos pesos sinápticos, são eles que devem ser inicializados de forma a criar condições para que o processo de minimização da função custo seja o mais rápido e obtenha a melhor performance possível. São encontrados na literatura maneiras de realizar a inicialização dos pesos de maneira analítica, como em Erdogmus et al. (2005), porém, a maneira mais difundida, e adotada por este trabalho, é a inicialização através de uma distribuição uniforme, de com valores limitados no intervalo $[-1, +1]$ (HAYKIN, 1999).

Esta escolha, mesmo com todos os outros parâmetros de treinamento fixos, acarreta mudanças de performance a cada inicialização do processo de treinamento. Isso se deve à mudança das condições iniciais do mesmo, associando valores diferentes aos

pesos sinápticos a cada vez que o treinamento é inicializado. Como é desejado que a rede retorne resultados com erros que sejam os menores possíveis, este problema é enfrentado com múltiplas inicializações do treinamento, nas quais serão escolhidas as melhores redes no quesito performance. Para evitar um excesso de inicializações que demandaria um tempo computacional cada vez maior, foram realizadas simulações que avaliam a relação entre o número de inicializações e o erro RMS. Nestas simulações, foram fixados os seguintes parâmetros de treinamento:

- o conjunto de *treino* é composto de 55.000 amostras, referentes a um sinal com 50 % de ocupância;
- o conjunto de *teste* é composto de 900.000 amostras, divididas em nove subconjuntos de 100.000 amostras cada, referentes a sinais com ocupância de 10 % a 90 %;
- critérios de parada:
 - épocas: máximo 1000;
 - validação cruzada: 5;
- neurônios na camada oculta: 10;
- atrasos de entrada: 10.

Para avaliar a capacidade de generalização do modelo da rede será utilizada a técnica de validação cruzada. Ela consiste em dividir o conjunto de dados de teste em dois: treino e teste. Feito isto, o conjunto de treinamento é particionado aleatoriamente em dois subconjuntos:

1. um subconjunto utilizado para *estimação do modelo* (treinamento da rede);
2. um subconjunto utilizado para *avaliação da performance do modelo* (validação do treinamento).

Segundo Haykin (1999), o subconjunto de validação costuma ser de 10 % a 20 % do conjunto de treinamento. Para esta simulação, foram utilizadas 30 % das amostras para validação, pois o número de amostras disponíveis para treino é grande, sendo que, para proporções pequenas do conjunto de treinamento, pouco se altera na performance final da rede, como pode ser observado na Figura 34, na qual foi simulada a influência deste parâmetro, dadas as condições acima.

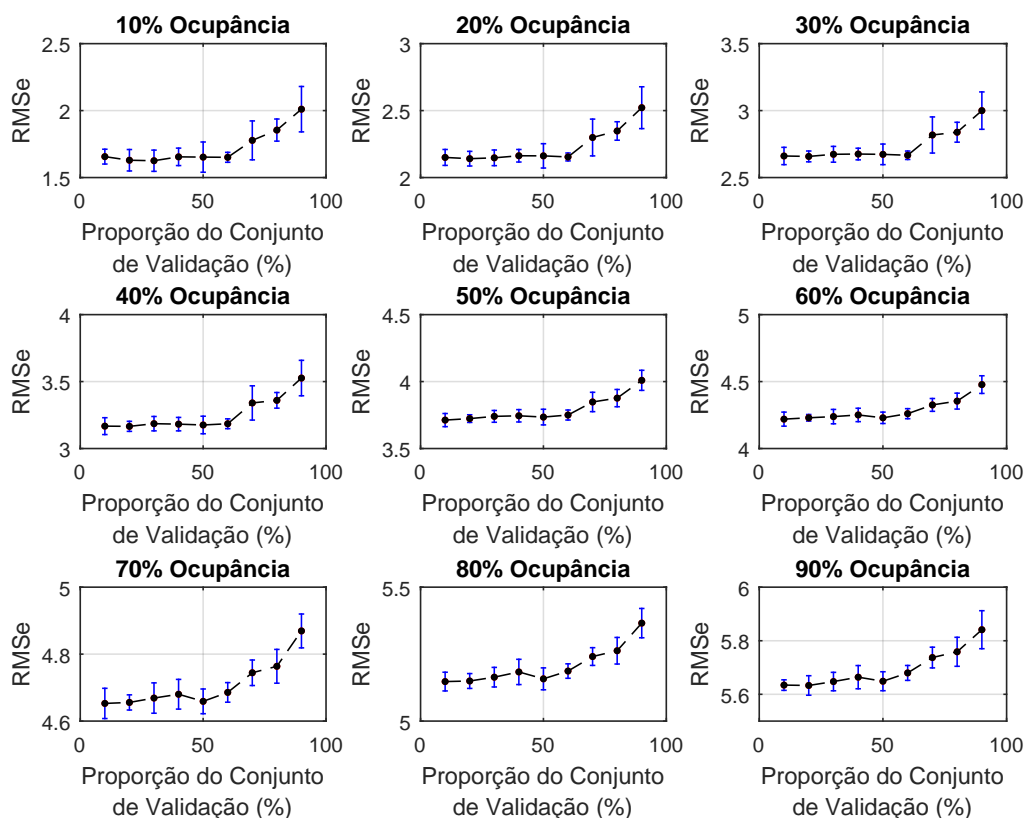


Figura 34: Influência da proporção do subconjunto de validação em relação ao conjunto de treinamento.

Cada conjunto descrito acima é constituído de pares de sinal de entrada/saída desejada $\{y, x\}$ para cada amostra, devido ao processo de treinamento ser supervisionado. Os valores armazenados em y são dados que simulam a leitura em um canal do Tilecal com determinada ocupância, constituído de uma janela de m amostras, com características definidas pelo conjunto ao qual pertence. Já x , é o valor de energia teórico, depositado em cada célula, desprovido de ruído e das demais distorções do calorímetro. O padrão de descrição dos parâmetros de treinamento desta seção será utilizado ao longo deste trabalho.

Os resultados de simulação da influência do número de inicializações encontram-se ilustrados na Figura 35. O processo de treinamento foi inicializado 50 vezes no total, no qual a cada 10 inicializações era montado um histograma da performance das redes geradas. É possível observar que, em algumas destas inicializações, a rede convergiu para uma solução ruim, resultando em um erro RMS mais alto. Com base nestas informações, foi analisada a capacidade da rede encontrar as melhores soluções dentro de conjuntos dos 10 menores RMSe para cada conjunto de inicializações, ou seja, para

20 inicializações. Os 10 melhores resultados foram salvos, e os demais descartados, e assim sucessivamente. Os resultados podem ser observados na Figura 36.

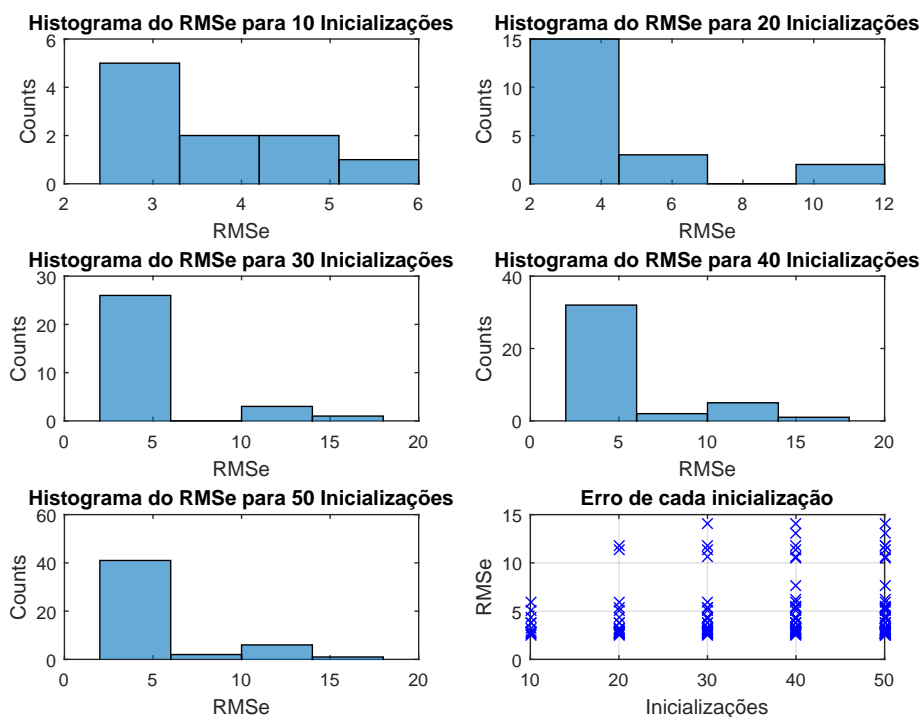


Figura 35: Histogramas para os conjuntos de inicializações.

Cada coluna do gráfico de barras é constituída de um subconjunto dos 10 menores RMSe para 10, 20, 30, 40 e 50 inicializações, com a média do erro e sua dispersão. O sinal utilizado de teste nas Figuras 35 e 36 tem 30 % de ocupância. Para certificar se o ruído de *pile-up* tem influência nesta análise, o mesmo processo realizado anteriormente, ilustrado pela Figura 36, foi feito para sinais com as demais ocupâncias do conjunto de teste, ilustrados na Figura 37.

Foi observado que o padrão para sinais com outras ocupâncias se repete e que, a partir de 30 inicializações, tanto a média quanto a dispersão do RMSe pouco se alteram. Logo, é escolhido realizar todas as simulações que determinarão os outros parâmetros da rede realizando-se 30 inicializações.

5.2 CONJUNTOS PARA TREINAMENTO E VALIDAÇÃO

É evidente que o número de amostras disponíveis para realizar o treinamento tem grande influência sobre a generalização da rede. Em problemas em que os dados são obtidos através de coletas experimentais, possuindo número de amostras limitado, deve-

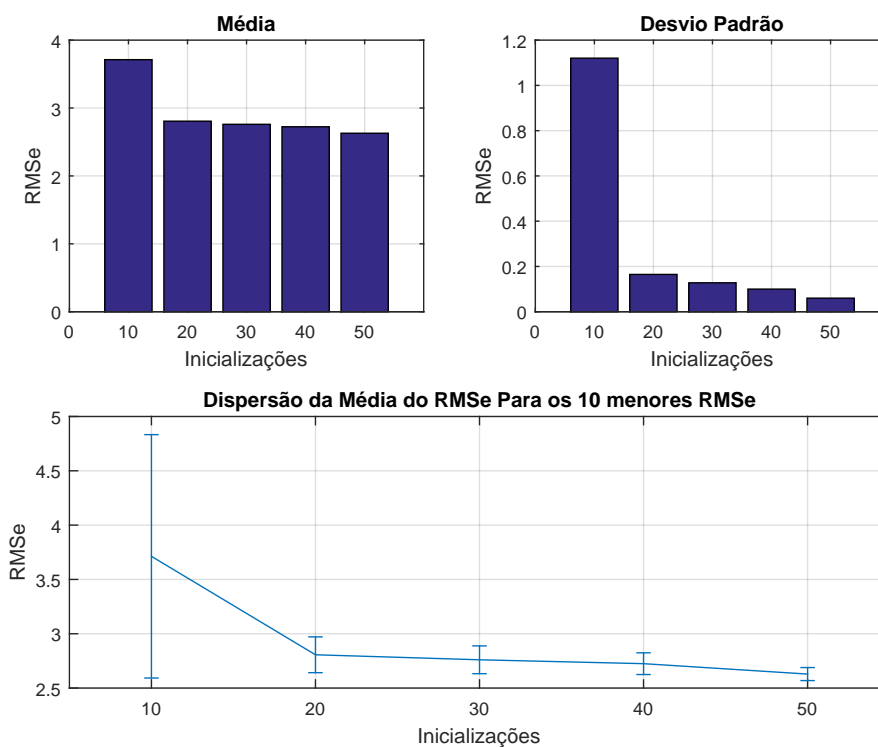


Figura 36: Média e dispersão do erro RMS para cada conjunto dos 10 melhores resultados.

se ficar atento ao tamanho da rede, relacionando o número de pesos a serem ajustados (graus de liberdade) com o número de amostras para realizar este ajuste. Em Haykin (1999) é sugerida uma regra, conhecida como regra de ouro de *Widrow*, na qual:

$$N > \frac{W}{\epsilon} \quad (5.1)$$

em que para um erro de generalização (ϵ) de 10 %, o número de amostras para treinamento (N) deve ser 10 vezes maior que o número de pesos sinápticos da rede (W).

Outro fator que tem influência considerável sobre a capacidade de generalização da rede é a complexidade da sua arquitetura. Neste trabalho, como os dados são gerados através de simulações de *Toy Monte Carlo*, foram criados conjuntos de treinamento, contendo números de amostras distintas, variando de 1.000 a 70.000. Para estas simulações, são utilizados os seguintes parâmetros:

- o conjunto de *treino* é composto de de 70 % das 55.000 amostras, referentes a um sinal com 50 % de ocupância;
- o conjunto de *validação* é composto de de 30 % das 55.000 amostras, referentes

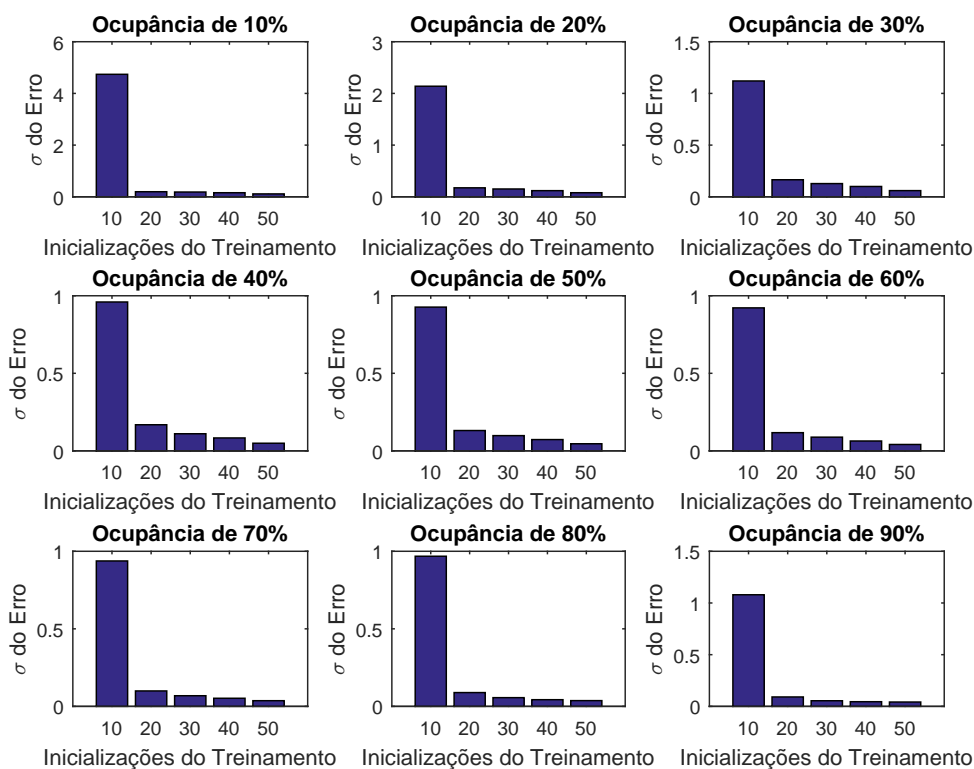


Figura 37: Dispersão do erro RMS para cada conjunto dos 10 melhores resultados para sinais com ocupâncias distintas.

a um sinal com 50 % de ocupância;

- o conjunto de *teste* é composto de 900.000 amostras, divididas em nove subconjuntos de 100.000 amostras cada, referentes a sinais com ocupância de 10 % a 90 %;
- critérios de parada:
 - épocas: máximo 1000;
 - parada por validação: 5;
- neurônios na camada oculta: 10;
- atrasos de entrada: 10;
- inicializações: 30.

O resultado das simulações para um sinal de teste com ocupância de 30 % está representado na Figura 38. Resultados para sinais de teste com as demais ocupâncias podem ser vistos na Figura 39. É possível observar que para conjuntos de treinamento

pequenos ($N < 10.000$), a capacidade de generalização da rede foi bastante comprometida, quando comparada a resultados de treinamentos com conjuntos de amostras maiores ($N = 50.000$). A arquitetura utilizada possui 121 pesos a serem ajustados. Logo, se o critério de erro (ϵ) for restringido a 1 %, N deve ser maior que 100 vezes o número de pesos da rede, ou seja, 12.100 amostras, mostrando-se coerente com a regra de ouro de *Widrow*.

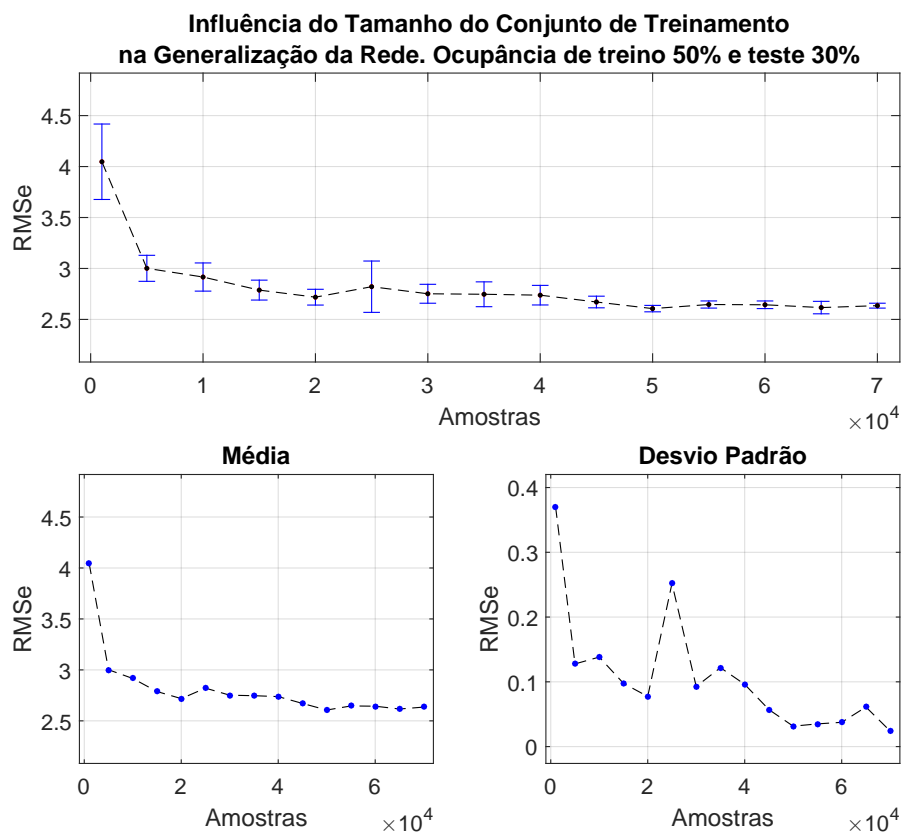


Figura 38: Dispersão do erro RMS para cada conjunto dos 10 melhores resultados para sinais com ocupâncias distintas.

É possível observar também que, a partir de 50.000 amostras, pouco se ganha em termos de performance. Vale salientar que, ao utilizar o algoritmo de Levenberg-Marquardt, em média, são conseguidos bons resultados em poucas épocas de treinamento. No entanto, ao utilizar grandes conjuntos no treinamento, ele se torna lento para realizar o ajuste dos pesos sinápticos referentes a uma época de treinamento. Escolhendo $N = 50.000$, de acordo com o sistema de processamento disponível para treinamento, ainda eram obtidos resultados em um tempo razoavelmente hábil, e com boa performance, em média. Outro fator que influencia na capacidade de generalização da rede são os critérios de parada, detalhados na Seção 5.3.

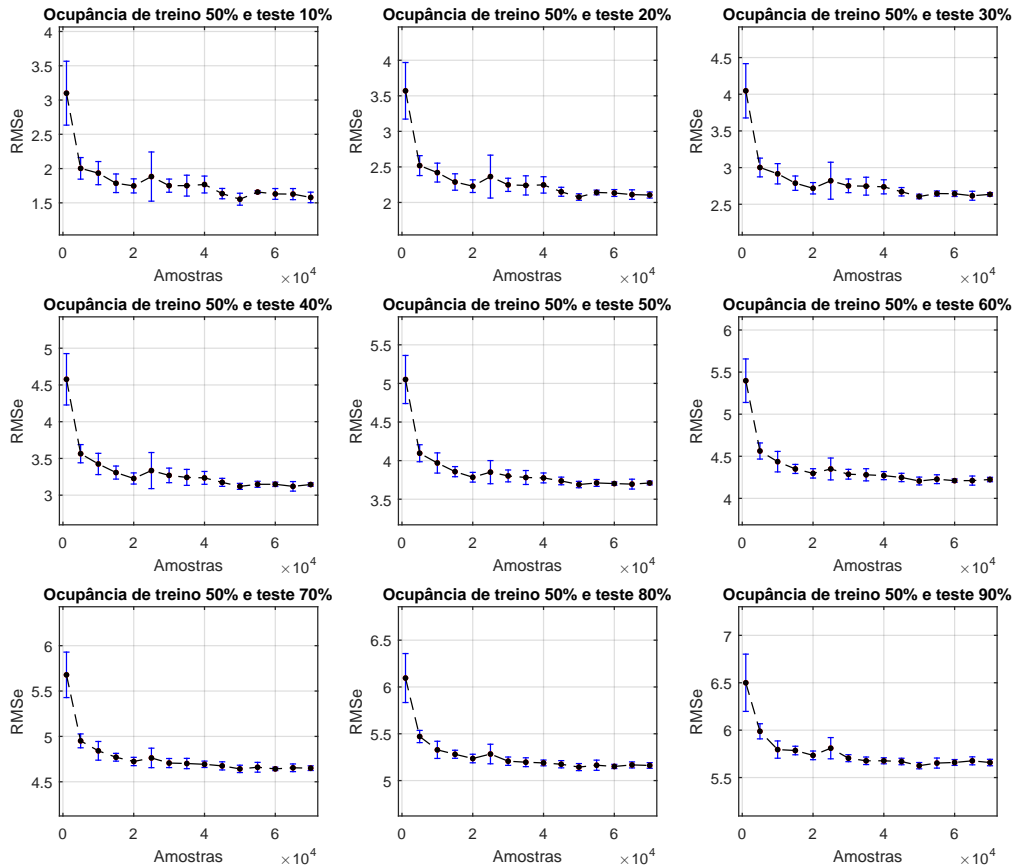


Figura 39: Dispersão do erro RMS para cada conjunto dos 10 melhores resultados para sinais com ocupâncias distintas.

5.3 AJUSTE DOS CRITÉRIOS DE PARADA

Os critérios de parada do treinamento, quando corretamente ajustados, são de grande importância para se evitar *overtraining* e *undertraining*. Muitas vezes, são utilizados uma combinação de vários critérios, ajustados para não permitir que tais condições de adaptação da rede ocorram. Um dos critérios mais utilizados é impor um limite ao número de épocas para que o treinamento se encerre. Devido à aleatoriedade das condições iniciais, ajustar este parâmetro se torna difícil. No entanto, ele é geralmente utilizado com algum outro critério, como as checagens por validação cruzada. Este último consiste em realizar a validação a cada época, guardando a medida de performance do processo de otimização (MSE), comparando-a com a performance das épocas subsequentes. Caso o MSE não decresça, ou aumente nas próximas i épocas, o processo de treinamento é interrompido. Essa combinação de parâmetros é utilizada nas simulações para encontrar um valor i de checagens por validação que pouco influen-

cie na performance final da rede. O número de épocas foi fixado em um valor alto, com o intuito de testar somente a influência da parada por validação cruzada. Os demais parâmetros encontram-se abaixo.

- O conjunto de *treino* é composto de 70 % das 50.000 amostras, referentes a um sinal com 50 % de ocupância;
- o conjunto de *validação* é composto de de 30 % das 50.000 amostras, referentes a um sinal com 50 % de ocupância;
- o conjunto de *teste* é composto de 900.000 amostras, divididas em nove subconjuntos de 100.000 amostras cada, referentes a sinais com ocupância de 10 % a 90 %;
- critérios de parada:
 - épocas: máximo 10.000;
 - parada por validação: $i = \{1, 2, 3, 4, 5, 10, 15, 20, 25, 30\}$;
- neurônios na camada oculta: 10;
- atrasos de entrada: 10;
- inicializações: 30.

Na Figura 40, é possível observar que para um sinal de teste com ocupância de 30 %, por exemplo, quando se utiliza um número de checagens pequeno ($i < 4$), o treinamento, em média, converge para uma solução ruim, indicando *underfitting*. Para valores altos de i , o erro também pouco se alterou. Este padrão se repete para todo o conjunto de teste, no qual para sinais com ocupâncias distintas, com 4 checagens por validação cruzada se obteve a melhor performance.

O número de épocas máximo fixo em 10.000 pouco influenciou nos resultados. Pode-se afirmar isto, pois, apenas 4 % dos treinamentos se encerraram por excederem este limite, resultando em baixa performance. Portanto, este valor de erro foi tratado como um *outlier* do conjunto de resultados e eliminado dos dados utilizados para análise. Até agora, o banco de dados de treinamento era composto de sinais com ocupância fixa em 50 %. Na Seção 5.4 será avaliada a influência da ocupância deste sinal na performance da rede.

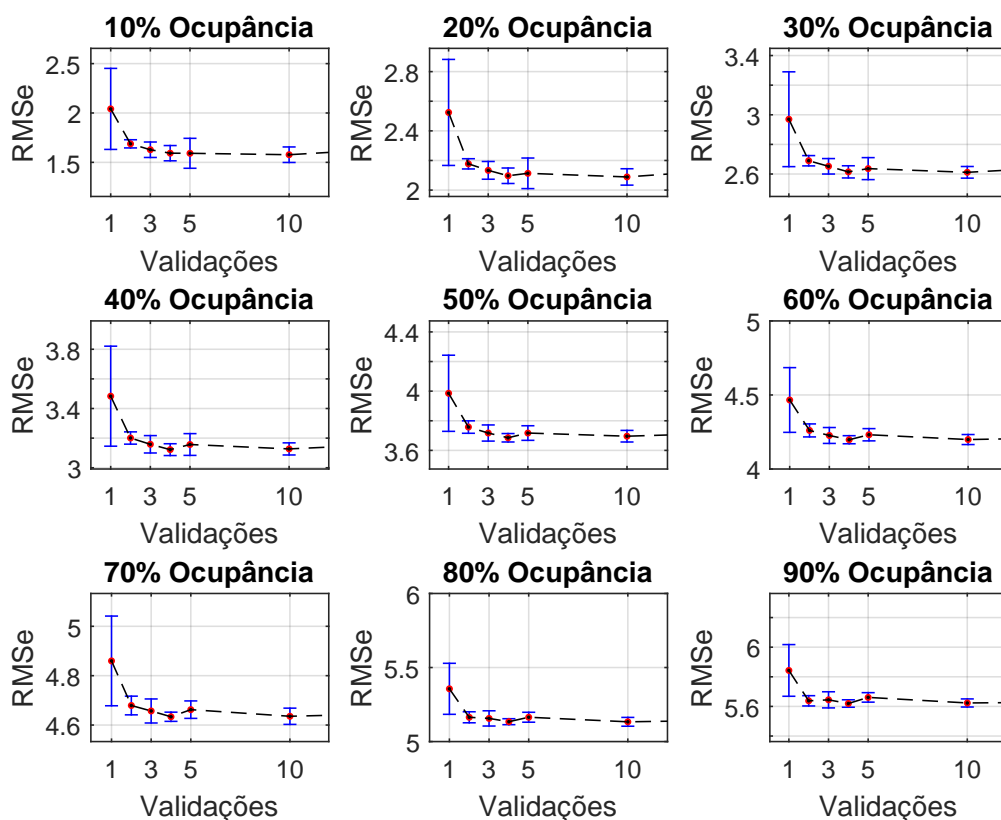


Figura 40: Performance média para todos os cenários de teste.

5.4 INFLUÊNCIA DA OCUPÂNCIA DO SINAL NO TREINAMENTO

O conjunto utilizado para treinamento de uma rede neural tem grande impacto no resultado do mesmo. Esta seção tem como objetivo avaliar treinamentos com sinais que apresentam ocupâncias distintas, na performance final da rede para diferentes cenários (conjuntos de teste). Essa análise é encorajada pelo fato de que sinais com ocupâncias distintas apresentam níveis de ruído de *pile-up* também distintos. A hipótese testada aqui é que se é desejável uma rede neural que generalize para qualquer cenário de ocupância, caso seja adequada ao nível de ruído apenas do sinal de treinamento, a rede terá sua capacidade de generalização comprometida ou reduzida. Na Figura 41 é ilustrado o resultado das simulações para os parâmetros listados abaixo:

- o conjunto de *treino* é composto de 9 sub-conjuntos contendo 70 % de 50.000 amostras, no qual cada sub-conjunto é composto de um sinal com ocupância distinta, variando de 10 % a 90 %;
- o conjunto de *validação* é composto de 9 sub-conjuntos contendo 30 % de 50.000

amostras, no qual cada sub-conjunto é composto de um sinal com ocupância distinta, variando de 10 % a 90 %;

- o conjunto de *teste* é composto de 900.000 amostras, divididas em nove sub-conjuntos de 100.000 amostras cada, referentes a sinais com ocupância de 10 % a 90 %;
- critérios de parada:
 - épocas: máximo 1.000;
 - prada por validação: 4;
- neurônios na camada oculta: 10;
- atrasos de entrada: 10;
- inicializações: 30.

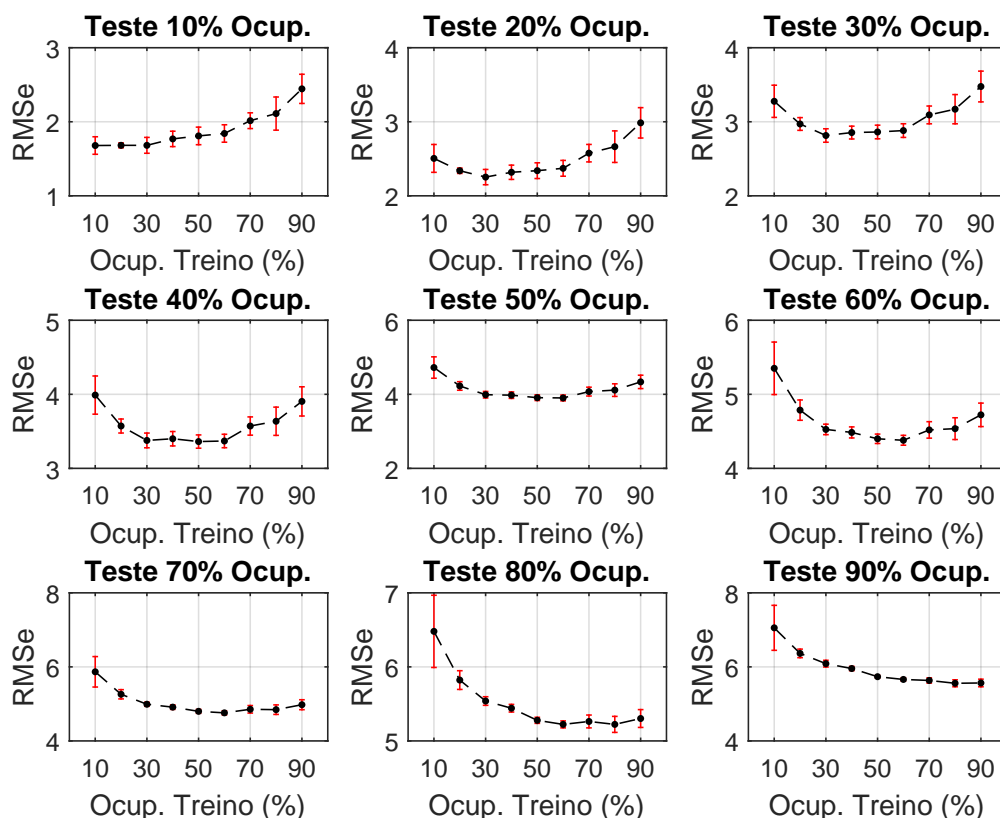


Figura 41: Influência da ocupância do sinal de treinamento em diferentes cenários.

Cada um dos gráficos na Figura 41 representa a performance média de redes treinadas com sinais que apresentam ocupâncias variando de 10 % a 90 %, testadas em um

cenário distinto. É possível observar que há influência da ocupância do sinal de treinamento na performance da rede, em média. Pode-se inferir a partir desses resultados que para sinais de treinamento com ocupância em 50 % há a menor variação dentre os cenários testados na situação a qual se deseja obter uma rede com pesos fixos, que possa atuar em todos os cenários de ocupância. No caso em que seja possível montar um sistema em que os pesos possam ser carregados de acordo com a ocupância que a célula venha apresentar durante o seu funcionamento, foram montados dois sistemas de rede especialistas por faixa de ocupância:

- **Especialista 1:** Resultados que compõem um sistema de rede que contempla nove conjuntos de coeficientes, cada um deles ajustado através do treino da rede com um sinal que apresenta ocupância diferente, abrangendo os nove cenários em análise neste trabalho.
- **Especialista 2:** Ao invés de utilizar um conjunto de pesos para cada faixa de ocupância do sinal, foi feita uma tentativa de reduzir a complexidade do modelo Especialista 1, baseando-se na capacidade de generalização de uma rede treinada com um sinal em certa faixa de ocupância, que generalize bem em faixas vizinhas de ocupância do sinal. Este sistema constitui-se de três conjuntos de pesos, treinados com ocupâncias de 20 %, 50 % e 80 %.

Os resultados para os sistemas propostos são comparados na Figura 42(a), na qual os outros parâmetros utilizados nas simulações anteriores se mantiveram e, nesta figura consta apenas o resultado da melhor rede encontrada para cada modelo proposto. O estimador chamado de **Genérico** é a rede neural treinada com um sinal de 50 %. É possível observar que os estimadores especialistas alcançaram resultados semelhantes ao estimador genérico para ocupâncias menores que 60 %. No entanto, para sinais de teste com ocupância maior que este valor, o modelo Especialista 1 mostrou uma performance ligeiramente superior. Com base nestes resultados e, em virtude de desejar um sistema menos complexo que possa ser instalado em ambientes que apresentem sinais com qualquer nível de ruído de *pile-up*, é proposto a utilização de um sinal para treinamento com apenas um conjunto de dados, contendo amostras de todos os nove cenários de ocupância em análise. À rede treinada com ele é dado o nome de **Ocupância Variável**, sendo comparado aos modelos Especialista 1 e Genérico na Figura 42(b).

Ao observar os resultados mostrados na Figura 42(b), ressalta-se o fato de que ao treinar a rede com dados que contém amostras de sinais com níveis de ruído de

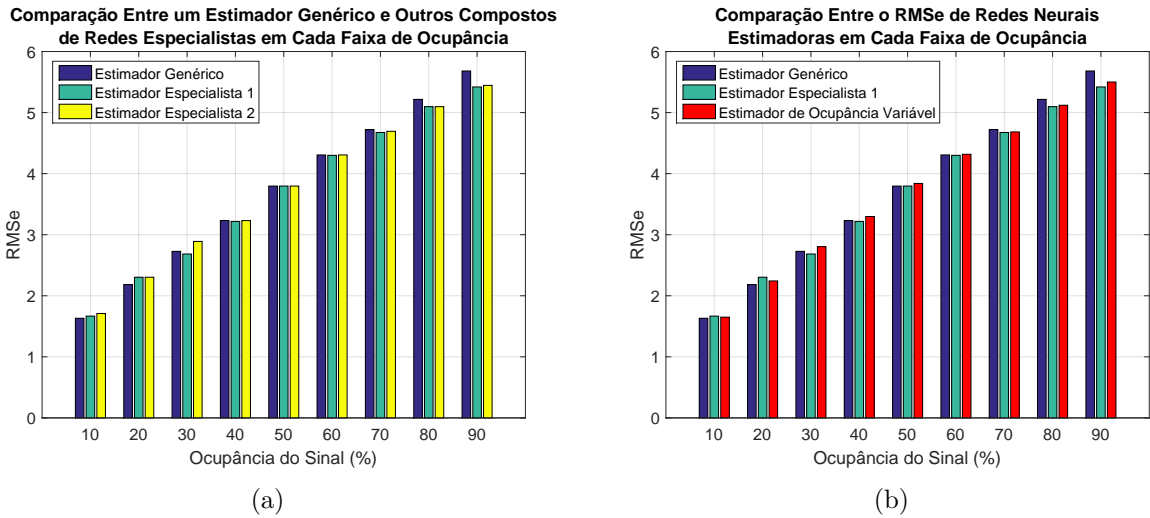


Figura 42: Comparação entre redes treinadas com diferentes conjuntos.

pile-up distintos, possibilitou que o processo de treinamento encontrasse um conjunto-solução de pesos sinápticos, permitindo reduzir o erro ao ser testada com sinais de alta ocupância. Além disso, sua performance foi mantida próxima aos demais modelos de treinamento propostos para os todos os cenários de ocupância. Logo, este modelo com ocupância variável se mostrou mais eficiente que treinar a rede com ocupância fixa em 50 %, além de ser constituído por um modelo mais simples, com apenas um conjunto-solução para qualquer cenário tratado neste trabalho, diferentemente dos modelos especialistas. Estes resultados abarcam a hipótese descrita anteriormente em que o nível de ruído de *pile-up* do sinal de treinamento pode comprometer a capacidade de generalização da rede para outros cenários, sendo este problema atenuado ao utilizar um sinal multi-ocupância.

Uma observação importante a ser feita é que os treinamentos utilizando o sinal com ocupância variável, em média, demoraram mais épocas para convergir do que utilizando os sinais com ocupância fixa. Em virtude desta informação, apenas a rede com a arquitetura de neurônios e atrasos de entrada já definida será treinada com este sinal, visando reduzir o tempo computacional para as simulações realizadas na Seção 5.5, as quais envolvem a combinação de várias destas arquiteturas.

5.5 DEFININDO O NÚMERO DE ATRASOS DE ENTRADA E NEURÔNIOS NA CAMADA OCULTA

A última etapa para definir a estrutura da rede estimadora é encontrar uma combinação entre o número de nós de entrada (m), e o número de neurônios na camada oculta

(h_1). Estes parâmetros se mostraram dependentes entre si, inviabilizando analisá-los separadamente, como feito anteriormente com os demais. Por isso, o impacto deles na performance da rede estimadora será realizado de forma conjunta, criando redes com cada combinação possível, dentro de um espaço de busca $m \times h_1$. O treinamento de cada arquitetura é inicializado 30 vezes, assim como nas simulações anteriores, porém, apenas o resultado da rede que generalizou melhor para todos os cenários de teste será utilizado para comparar as diferentes arquiteturas. Ao serem testadas em sinais com ocupâncias diferentes, a performance de cada uma delas pode ser vista na Figura 43.

- O conjunto de *treino* é composto de 70 % das 50.000 amostras, referentes a um sinal com 50 % de ocupância;
- o conjunto de *validação* é composto de 30 % das 50.000 amostras, referentes a um sinal com 50 % de ocupância;
- o conjunto de *teste* é composto de 900.000 amostras, divididas em nove sub-conjuntos de 100.000 amostras cada, referentes a sinais com ocupância de 10 % a 90 %;
- critérios de parada:
 - épocas: máximo 1.000;
 - parada por validação: 4;
- neurônios na camada oculta: $h_1 = \{1, 2, 3, 4, \dots, 10, 11, 12\}$;
- nós de entrada: $m = \{1, 2, 3, 4, \dots, 13, 14, 15\}$;
- inicializações: 30.

Em cada uma das sub-figuras acima é destacado um ponto de RMSe mínimo encontrado. A este ponto está relacionada uma combinação entre m e h_1 , sendo que, para sinais com baixa ocupância, o aumento do número de neurônios proporcionou uma melhoria significativa da performance dessas redes, como observado nas sub-figuras 43(a) e 43(b). Já para sinais com alta ocupância, pouco se ganhou neste quesito quando o número de neurônios foi aumentado. No entanto, a janela de observação do sinal (número de atrasos) foi o principal fator para a otimização da figura de mérito, como observado nas sub-figuras 43(c) e 43(d). A partir desta análise, foi encontrada uma rede que generalize com boa performance para toda a faixa de ocupância, constituída de 10-4-1, ilustrada na Figura 44.

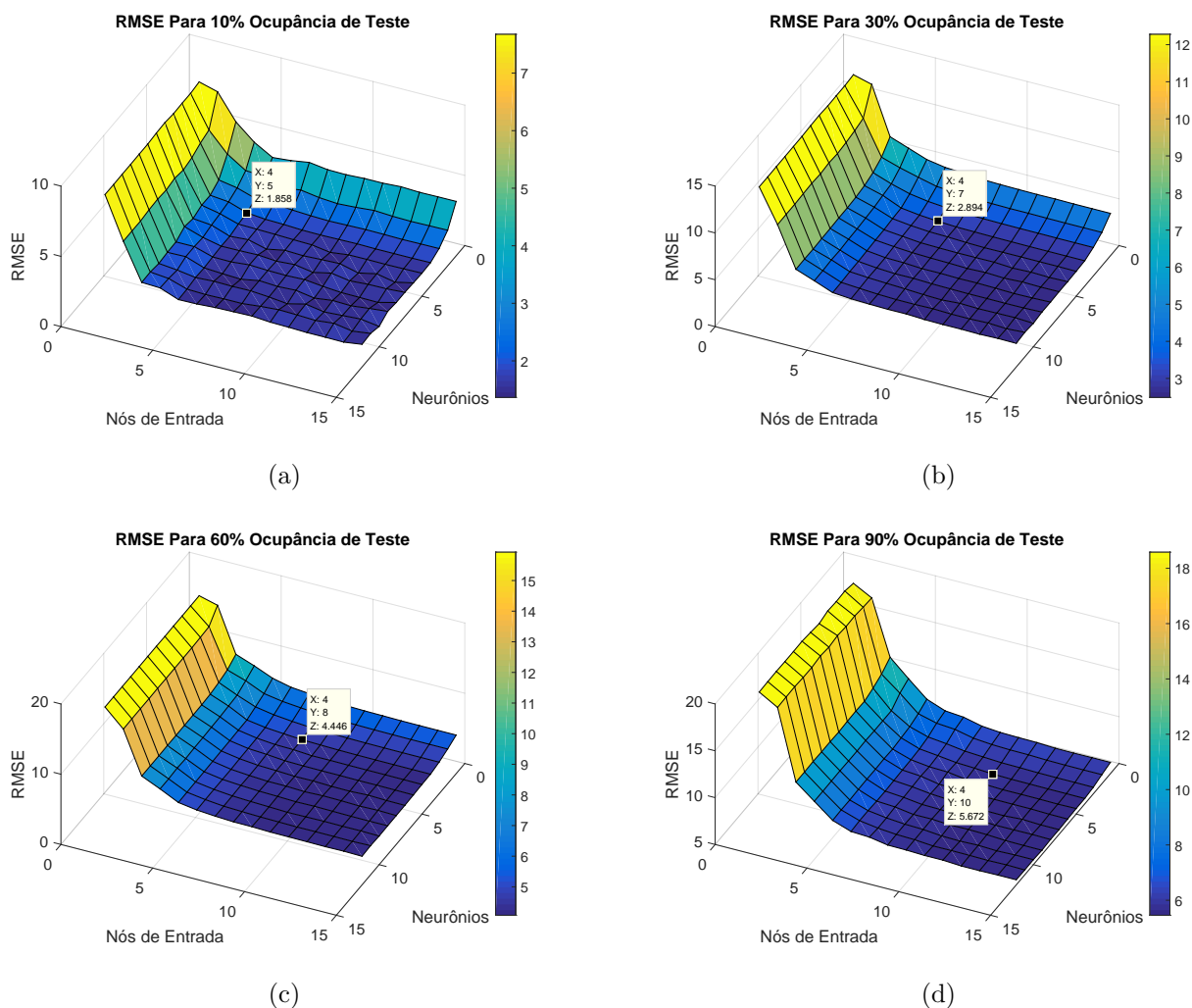


Figura 43: Performance média para todas as combinações possíveis dentro do espaço de busca $m \times h_1$ para sinais com diferentes ocupâncias.

Visto isso, esta rede foi treinada com o sinal que apresenta ocupância variável, contendo 450.000 amostras e o processo de treinamento foi inicializado 100 vezes, no qual a rede com a melhor performance foi selecionada. Na Figura 45(a) é ilustrada uma comparação entre a rede PMC estimadora e os métodos lineares ilustrados na Figura 17. É possível observar claramente que a rede neural estimadora alcança uma performance superior aos demais métodos considerados, principalmente em ocupâncias inferiores a 40 %, aproximando-se da performance dos filtros FIR quando a ocupância do canal tende a 100 %. Já na Figura 45(b), é ilustrada uma comparação sobre a robustez a intensidade de ruído eletrônico em um canal com ocupância de 30 %. Pode-se observar que o método determinístico deteriora mais intensamente enquanto a SNR diminui. O filtro casado, por conter a matriz de covariância do sinal com o ruído para projetar seus coeficientes, apresenta um menor deterioramento da performance. O filtro FIR adaptativo se sai bem em ambientes bastante ruidosos, pois o ruído eletrônico

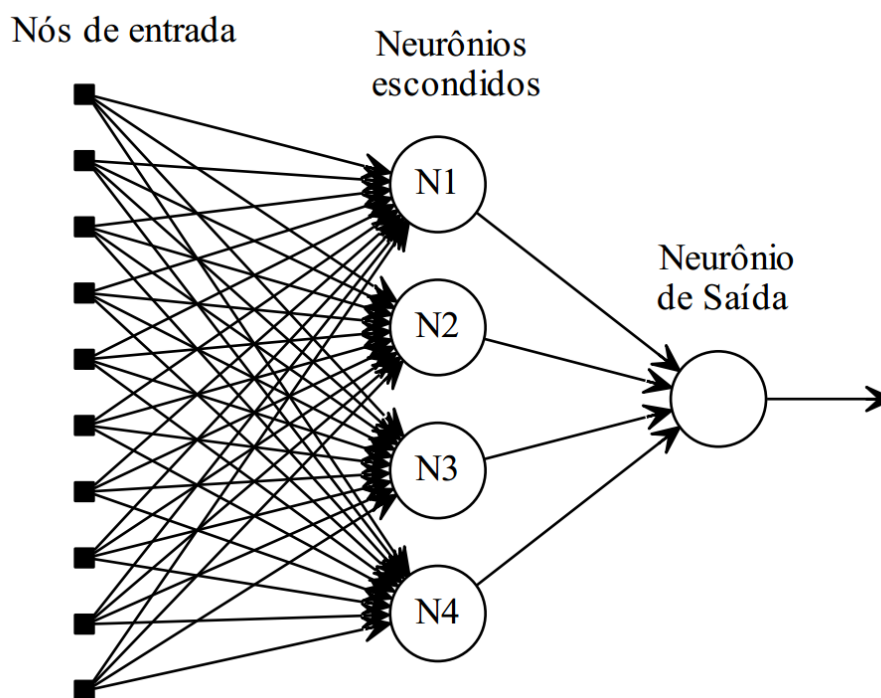


Figura 44: Rede neural 10-4-1.

gaussiano tende a ser mais influente que o de *pile-up*. Já a rede estimadora, mesmo com coeficientes fixos para os diferentes cenários de intensidade de ruído, mostrou a maior robustez dentre os métodos comparados. Esta condição ocorre devido a sua grande capacidade de extração de parâmetros estatísticos sobre os dados durante o treinamento, sendo um dos principais motivadores para aplicações de redes neurais em modelagem de sistemas que lidam com dados corrompidos por ruído.

Até o momento, foram realizadas comparações entre os métodos linear e não-linear sem a adição do conhecimento especialista sobre a diferenciação entre sinal e ruído. Com isso, muitas vezes, a rede e os demais estimadores podem estimar um sinal de ruído como se fosse um sinal real, constituindo em um falso alarme. Na Seção 5.6, serão abordadas formas de detectar o sinal antes de estimar a energia do mesmo, procurando melhorar a performance do sistema de estimação.

5.6 DETECÇÃO UTILIZANDO REDE NEURAL

Os parâmetros de treinamento, algoritmo, ocupância do sinal e critérios de parada utilizados aqui serão os mesmos encontrados ao fim da Seção ??, com exceção do número de neurônios na camada escondida. O número de atrasos de entrada deve ser o mesmo, pois como a rede detectora irá trabalhar em paralelo com o estimador

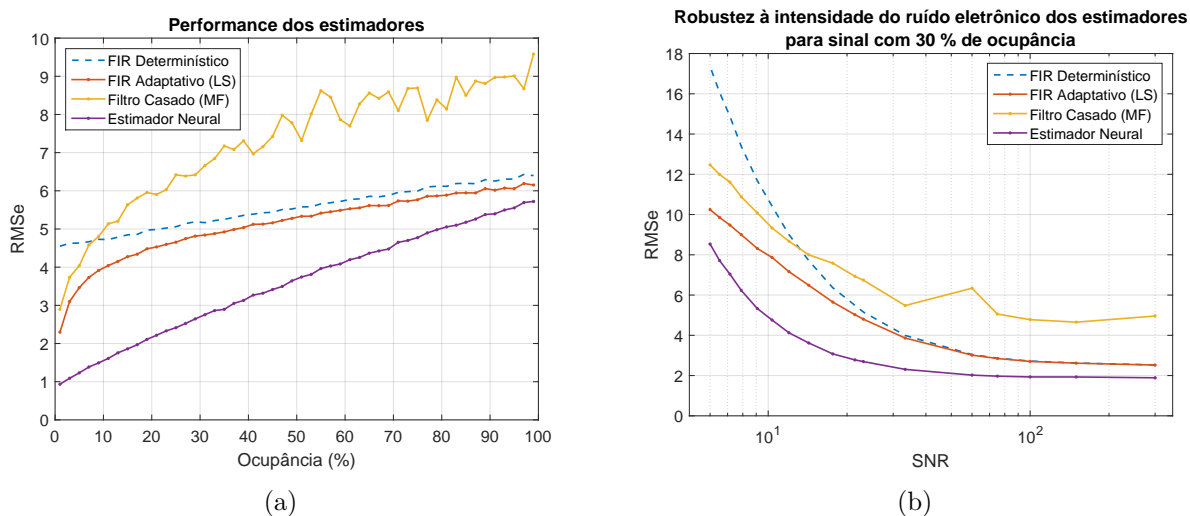


Figura 45: Comparação entre as performances dos filtros FIR, o filtro casado e o estimador neural.

neural, ambos devem ter como saída o mesmo elemento central da janela de dados. Um resumo dos parâmetros utilizados nos treinamentos encontra-se abaixo, e o resultado das simulações são ilustrados na Figura 46.

- O conjunto de *treino* constitui-se de 70 % das 50.000 amostras para treinamento, sendo que é composto de sinais referentes a todos os nove cenários de ocupância;
- o conjunto de *validação* constitui-se de 30 % das 50.000 amostras para treinamento, sendo que é composto de sinais referentes a todos os nove cenários de ocupância;
- o conjunto de *teste* é composto de 900.000 amostras, divididas em nove subconjuntos de 100.000 amostras cada, referentes a sinais com ocupância de 10 % a 90 %;
- critérios de parada:
 - épocas: máximo 1.000;
 - parada por validação: 4;
- neurônios na camada oculta: $h_1 = \{1, 2, 3, 4, \dots, 13, 14, 15\}$;
- atrasos de entrada: 10;
- inicializações: 30;

- alvos do treinamento:

 sinal: +1;

 ruído: -1.

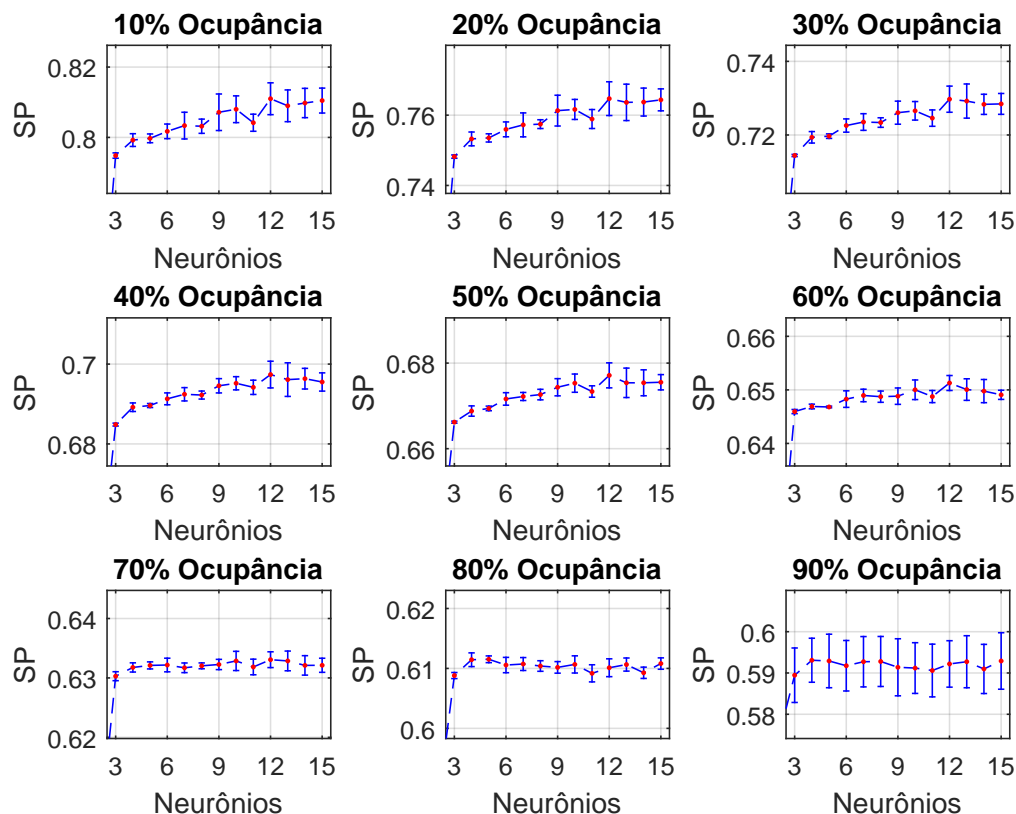


Figura 46: Neurônios na camada oculta da rede classificadora.

Pode-se claramente observar que para sinais com altas ocupâncias ($> 60\%$), a eficiência da rede não melhora significativamente acima ao treiná-la com mais de 4 neurônios na camada oculta. No entanto, a eficiência máxima alcançada para sinais com ocupâncias mais baixas ($\leq 60\%$) ocorreu ao treinar a rede com 12 neurônios na camada oculta. Esta arquitetura com 12 neurônios será considerada para realizar os testes em conjunto com o sistema de estimação propostos nesta seção, visto que a hipótese motivadora para realizar estes procedimentos é de que a estimação no primeiro nível pode se beneficiar desta combinação.

Paralelamente, foi testada a eficiência de detecção da rede estimadora encontrada na Seção ???. As eficiências dos dois sistemas de detecção propostos são comparadas na Figura 47, juntamente com o filtro FIR, no qual também é submetido a uma análise

por patamar decisor. As curvas ROC mostram que, para sinais com ocupância menor que 70 %, a rede especialista em detecção sobressaie em relação aos demais sistemas. No entanto, para sinais com altas ocupâncias, ($\geq 70\%$), a rede detectora nada acrescenta em eficiência em relação aos demais métodos. Outro teste realizado foi sobre a influência destes sistemas de detecção na performance de estimação final do estimador neural, sendo ilustrado para toda a faixa de ocupâncias na Figura 48. É possível observar que há uma relativa melhora na estimação ao implementar a rede classificadora em conjunto com a estimadora (Estimação + Detecção) para baixas ocupâncias ($\leq 45\%$), quando comparada a apenas a utilização da rede estimadora com patamar decisor. A implementação do sistema de estimação composto por duas redes neurais torna-se inviável devido à sua alta complexidade computacional (12 neurônios), ressaltando também que a rede neural estimadora apenas, já alcança relativo incremento na performance em relação aos métodos lineares. O sistema escolhido para implementação em hardware dedicado (FPGA) será apenas uma rede neural estimadora, visto que o patamar decisor não trouxe benefícios expressivos à performance do estimador, ficando a cargo do sistema de *trigger* realizar o corte no sinal. A influência dos decisores nos demais sistemas em análise nesta dissertação é ilustrada na Figura 49.

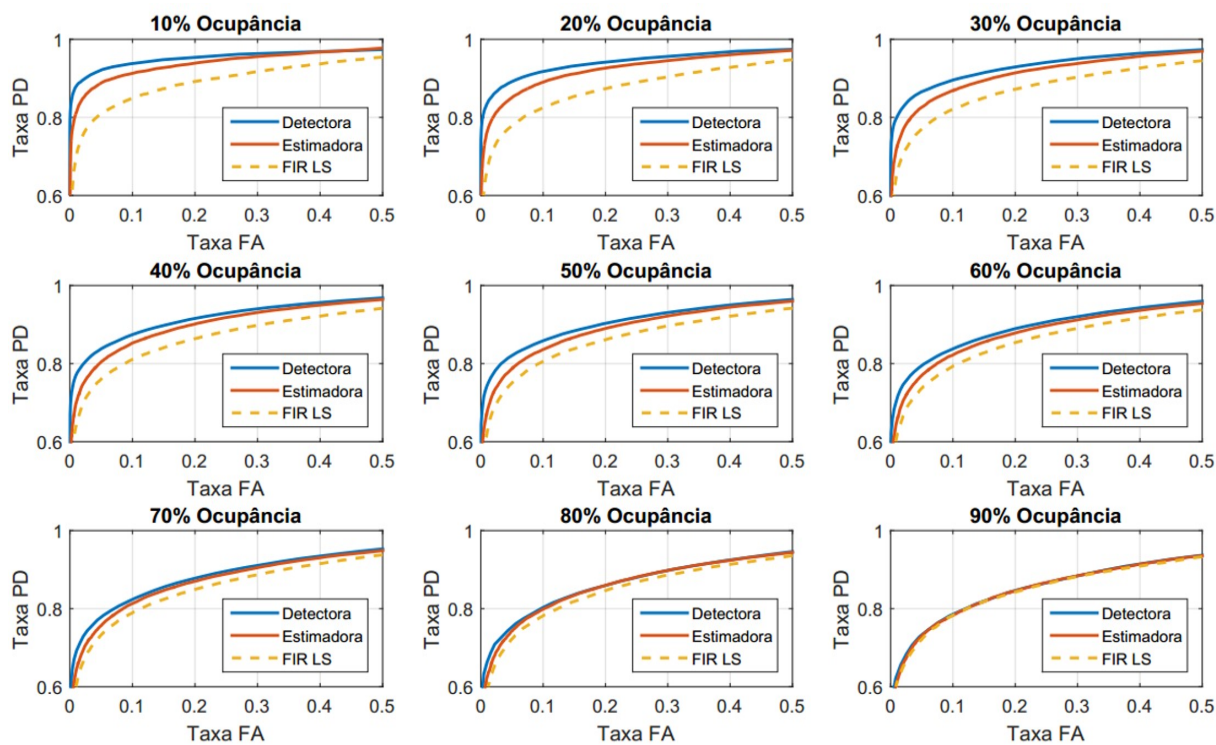


Figura 47: Curvas ROC para diferentes sinais de teste.

Observa-se que a partir da adição de um patamar decisor em zero, todos os sistemas de estimação alcançaram uma relativa melhora (em comparação à Figura 45). Isto se

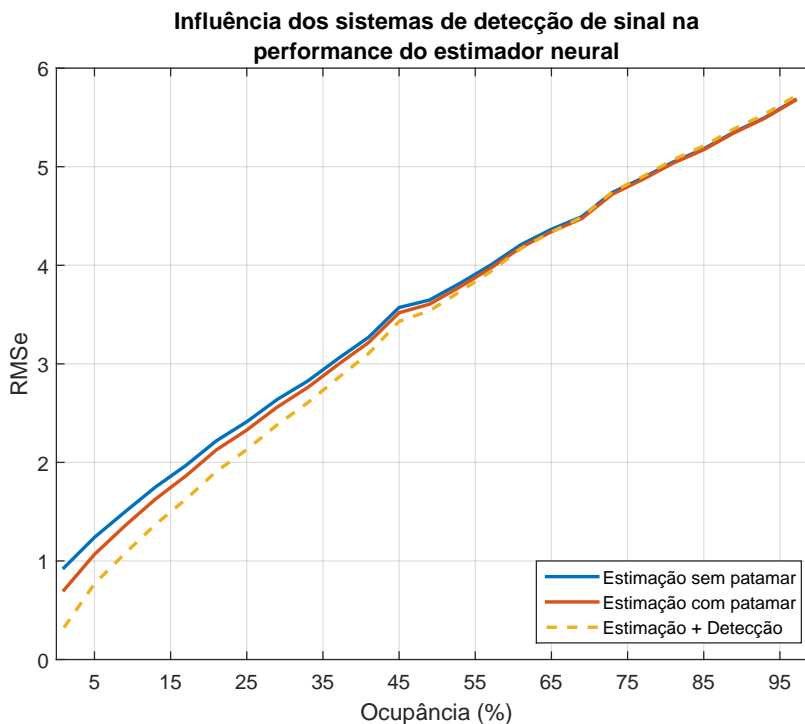


Figura 48: Neurônios na camada oculta da rede classificadora.

apresenta, pois não ocorrem deposições de energia negativas. Com base nisto, qualquer sinal de saída do estimador que seja menor que zero é interpretado como um ruído eletrônico, sendo esta amostra forçada a um valor nulo na saída do estimador. Para sinais com uma SNR de 21,43, o filtro FIR determinístico se aproxima, na maioria dos cenários, ao FIR adaptativo. Porém, em baixas ocupâncias ($< 20\%$), nos quais as amostras de ruído sobressaem-se, o FIR adaptativo apresentou evidente melhora na estimação. Isto ocorreu, pois o processo de ajuste dos pesos leva em consideração o ambiente em que está atuando, adaptando-se às condições do sinal a ser filtrado. Vale ressaltar também que, para valores de ocupância próximos a 100%, a performance dos métodos lineares e do estimador neural tendem a se igualar. No entanto, a superioridade do estimador neural é evidenciada para qualquer cenário discutido neste trabalho.

5.7 INTERPRETAÇÃO DOS PESOS SINÁPTICOS

Por serem baseadas nos sistemas neuronais biológicos, as redes neurais artificiais têm como principal característica o paralelismo e o processamento distribuído entre suas unidades constituintes, os neurônios. Esta última característica, na grande maioria das aplicações, dificulta a realização de uma análise que não seja apenas sua relação entrada/saída. De forma geral, as redes *feed-forward* com uma camada escondida

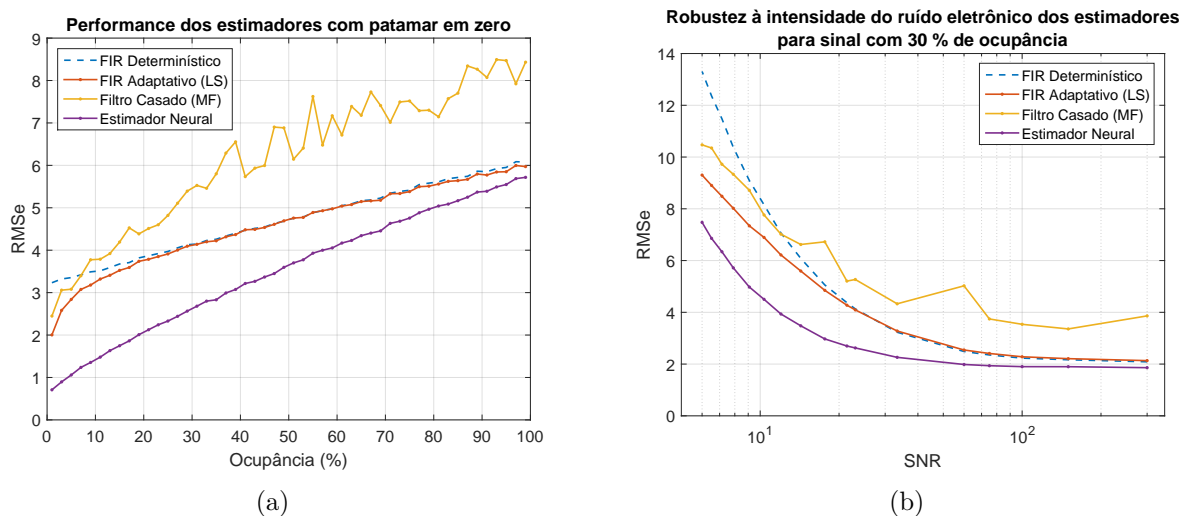


Figura 49: Comparação entre as performances dos filtros FIR, o filtro casado e o estimador neural com patamar em zero.

realizam três transformações nos dados de entrada:

1. transformação linear através dos pesos sinápticos;
2. deslocamento utilizando *bias*;
3. aplicação ponto-a-ponto de uma função de ativação monotônica.

Realizar uma análise de baixo nível em uma rede neural pode trazer uma nova perspectiva sobre um problema, proporcionando interpretações de como os dados são processados por suas unidades. Seguindo esta linha de pensamento, a rede 10-4-1, após ter seus pesos ajustados pelos processos descritos nas seções anteriores, foi submetida a uma análise entre os dados de entrada e saída de cada neurônio da camada escondida, conhecida como *scatter plot*. Neste tipo de análise, cada amostra de entrada no neurônio, após passar pela função de ativação, terá sua saída correspondente, constituindo um ponto no gráfico, ilustrado na Figura 50.

Nota-se que, os neurônios 1, 2 e 3 têm amostras ativando uma grande faixa de saturação negativa da tangente hiperbólica e, também, no domínio o qual a função possui maior derivada, como normalmente espera-se ao utilizar funções de ativação não-lineares. No entanto, o neurônio 4 teve apenas uma faixa aproximadamente linear da tangente hiperbólica ativada. Esses resultados estão intimamente conectados aos pesos da camada de entrada para cada um dos neurônios ilustrados na Figura 50.

Os pesos do neurônio 4 possuem uma configuração diferente dos demais, além de sua escala de valores estar confinada na ordem de 10^{-3} . A amplitude dos pesos

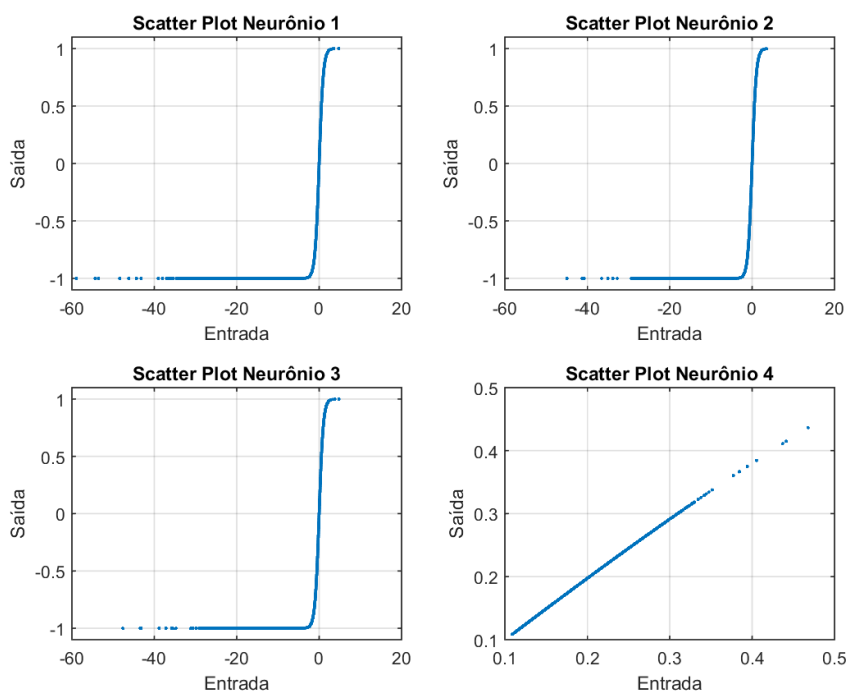


Figura 50: Dados de entrada/saída dos neurônios da camada escondida para sinal teste com 30 % de ocupância.

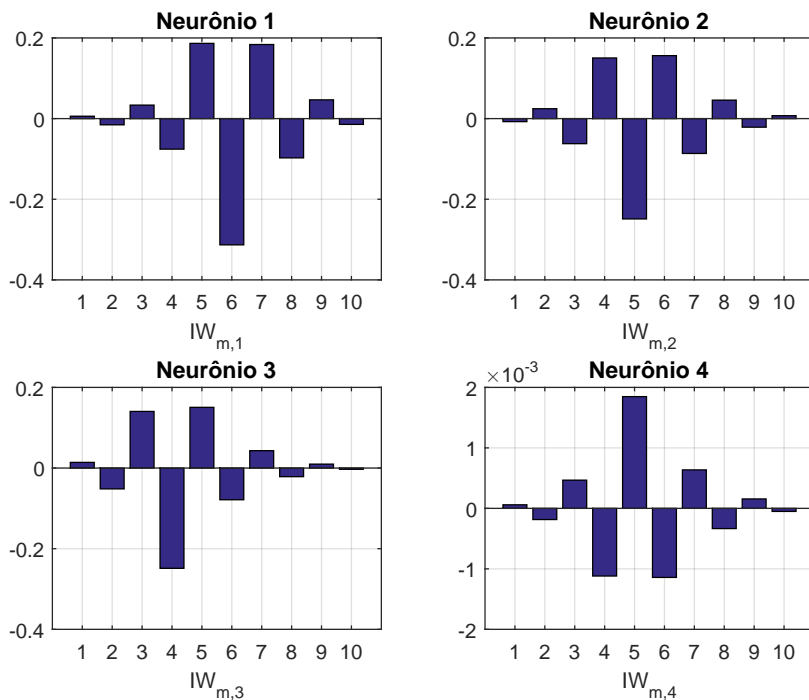


Figura 51: Pesos da camada de entrada (sem bias).

revela que este neurônio, em particular, está modificando a amplitude do sinal de entrada, confinando-o no domínio de derivada aproximadamente constante da tangente

hiperbólica. Uma hipótese para essa constatação é que este neurônio pode ter sido configurado no treinamento para lidar com as características intrinsecamente lineares do sinal em questão, aproximando-se de um filtro linear FIR de ordem 9, projetado para a equalização do canal do Tilecal usando a técnica de *Least Squares*, como descrito na Seção 4.1.2. A comparação entre os pesos do filtro FIR de ordem 9 e os pesos do quarto neurônio é mostrada na Figura 52. Para melhor comparação, os pesos foram normalizados pelo coeficiente central. Salvo uma constante de escala entre os dois conjuntos de valores, suas estruturas são bastante similares, no qual este padrão se repetiu para as redes que apresentaram os melhores resultados. Isto reforça a hipótese de que a rede neural, durante o treinamento encontrou uma solução pela qual realiza uma filtragem linear através de um neurônio, com uma correção não-linear, realizada pelos demais em paralelo. Uma regressão linear para o *scatter plot* do neurônio 4 foi realizada e está ilustrada na Figura 53. É possível observar que a maioria dos dados está contida na região na qual a tangente hiperbólica é aproximadamente linear.

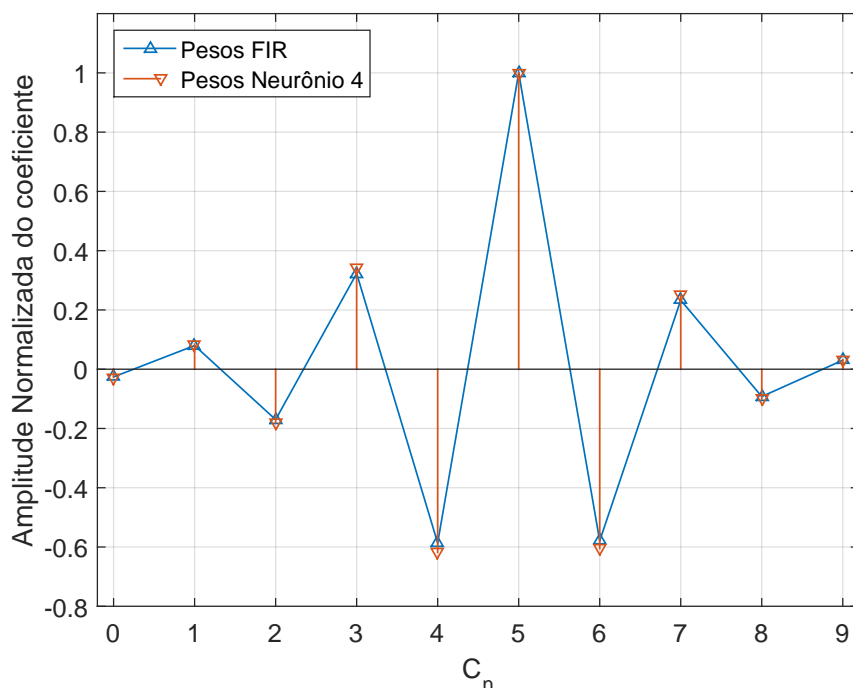


Figura 52: Pesos de entrada do neurônio 4 comparados aos de um filtro FIR de ordem 9.

Os pesos da camada escondida são um outro tipo de elemento reforçador para esta hipótese, como se mostra na Tabela 2; novamente destacando o papel do neurônio 4. O treinamento compensou os valores de saída pequenos do neurônio devido a sua escala dos pesos de entrada, com um peso muito grande nesta camada intensificando a ação que este neurônio tem na resposta final da rede neural, aproximando, em escala, dos

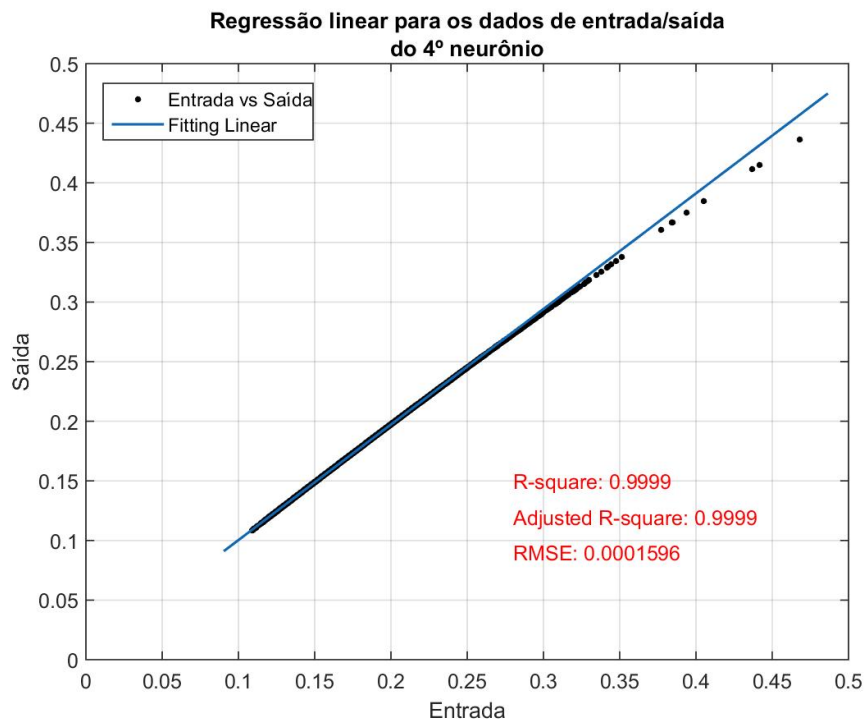


Figura 53: Regressão linear.

pesos do filtro FIR citado anteriormente.

Tabela 2: Pesos sinápticos na camada escondida.

Neurônio	1	2	3	4
Peso sináptico	-5,05	10,25	-7,48	1.358,23

Ao observar o histograma do sinal de cada neurônio, antes de serem combinados pelo neurônio linear de saída com bias, ilustrado na Figura 54, é possível inferir que a maior parte da informação de saída da rede vem do neurônio com ativação linear, enquanto que os outros três neurônios aplicam somente um ajuste fino não-linear à resposta final. Logo, o bias do combinador linear de saída, que é de $-189,21$, faz um deslocamento da contribuição dos quatro neurônios, como pode ser observado na Figura 55. Em Duarte (2015) foi demonstrado que um filtro FIR com apenas 10 coeficientes não realiza a estimação de forma muito eficiente, sendo utilizado 26 coeficientes para alcançar a melhor performance no ambiente em questão. Ao se utilizarem neurônios em paralelo para correção não-linear, a rede neural consegue estimar a energia em uma janela de sinal mais estreita. Essa característica, como relatado ao fim do Capítulo 2, é de grande importância para trabalhar em um ambiente com alta taxa de eventos.

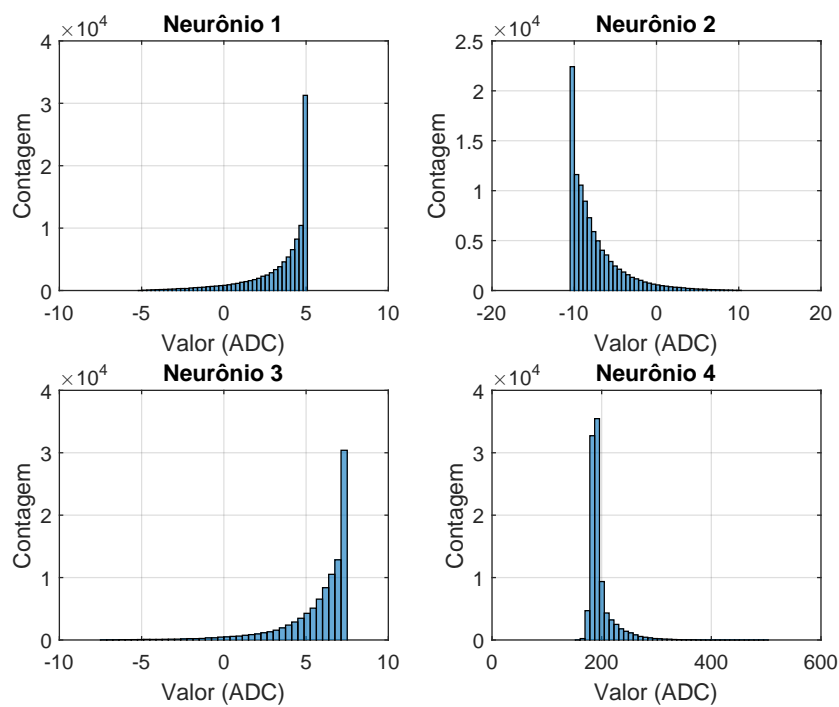


Figura 54: Contribuição de cada neurônio antes da combinação linear de saída com bias.

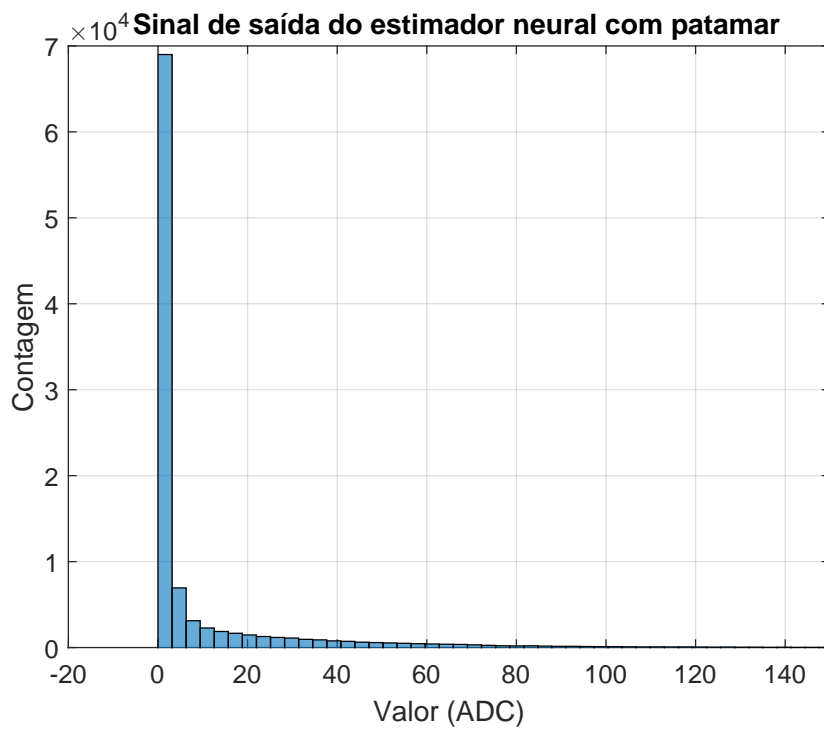


Figura 55: Histograma do sinal de saída da rede.

6 IMPLEMENTAÇÃO EM FPGA

Neste capítulo serão detalhados os métodos utilizados para quantização do estimador neural, assim como sua implementação e análise temporal, com o objetivo de realizar o seu funcionamento em um ambiente com alta taxa de eventos.

6.1 QUANTIZAÇÃO DO ESTIMADOR NEURAL

Redes neurais artificiais são utilizadas como soluções para diversos problemas práticos. No entanto, muitas vezes, sua aplicação em *software* não atende aos requisitos de performance desejados, visto que um *hardware* dedicado pode realizar operações aritméticas de maneira mais rápida que em um *software*. Este incremento de velocidade pode ser alcançado caso o paralelismo inerente de processamento das redes neurais seja explorado adequadamente (WOLF; ROMERO; MARQUES, 2001; DIAS; ANTUNES; MOTA, 2004).

Ao projetar um circuito para implementação em *hardware*, inicialmente, deve-se ter os cuidados de discretizar todas as operações lógicas e aritméticas. Pode-se realizar a representação numérica de duas formas: ponto fixo e flutuante. O ponto fixo apresenta precisão numérica pré-estabelecida, no entanto, necessita de menos recursos para se realizar operações, como multiplicações e adições. Já a utilização de ponto flutuante é capaz de adequar a precisão numérica às magnitudes dos números envolvidos, apresentando menores erros de quantização. Contudo, consome uma maior quantidade de recursos ao utilizar este formado em *hardware* em comparação com a representação em ponto fixo. Logo, visando implementações em *hardware* dedicado, sempre haverá uma busca por um equilíbrio chamado de área *versus* precisão. Na qual área representa a quantidade de recursos necessários para a implementação de acordo com as escolhas de representação numérica, e a precisão é a consequência dessa escolha, devendo estar de acordo com os requisitos do projeto (PADGETT; ANDERSON, 2009; SAVICH; MOUSSA; AREIBI, 2007).

Segundo Padget; Anderson (2009), mais de 90 % dos sistemas de processamento digital de sinais utilizam aritmética de precisão em ponto fixo. Isso ocorre, pois *hardware* em ponto fixo são mais baratos e energeticamente eficientes, além de mais rápidos, em comparação aqueles em ponto flutuante, pois exigem circuitos aritméticos menores. O método proposto neste trabalho visa sua implementação em FPGAs nas sROD do Titecal na Fase-II. As FPGAs sobressaem-se por serem mais rápidas e, quando se utiliza aritmética de ponto fixo, possibilitam a construção de palavras binárias em qualquer tamanho desejado, tornando o circuito necessário para realizar as operações altamente compacto. Logo, optou-se pela aritmética em ponto fixo para representar o estimador neural em uma FPGA, devido ao fato de que nas sROD, em uma mesma FPGA, serão lidos vários canais em paralelo na taxa de 40 MHz.

6.1.1 ARITMÉTICA DE PONTO FIXO

Em aplicações de processamento digital de sinais, a utilização de números decimais com sinal (diferenciação entre negativos e positivos) é a mais adequada. Existem três formas de representar um número levando em consideração seu sinal: magnitude sinalizada, complemento de um e complemento de dois. A representação por magnitude sinalizada utiliza um sinal (+ ou -) separado do próprio número, no qual se reserva um bit para isto, sendo 0 para números positivos e 1 para os negativos. Como os bits de magnitude são os mesmos para os positivos e negativos, algumas dificuldades de implementação são criadas, visto que circuitos diferentes serão necessários para somar números com o mesmo sinal e com sinais opostos. Além disso, possui duas formas de representar o zero (+0 e -0). O complemento de um também possui um bit reservado para o sinal, porém, os números negativos são representados pelo complemento da magnitude dos bits positivos. Esta representação resolve o problema de se necessitar de circuitos diferentes para realizar a soma, no entanto, ainda existem duas formas de representar o número zero. Já o complemento de dois é largamente utilizado na computação pela forma a qual trata números positivos e negativos, além do fato de possuir apenas uma representação para o zero. Os números positivos são representados da mesma forma dos demais métodos, porém, os negativos, são o resultado do complemento de um, no qual a palavra binária resultante é somada a '1' (PADGETT; ANDERSON, 2009). Na Tabela 3 são comparadas as formas de representação de números detalhados acima para uma palavra binária de três bits.

Tabela 3: Representações com sinal e sem sinal para palavra binária de três bits.

Palavra binária	Representação	Magnitude	Complemento	Complemento
	sem sinal	senalizada	de um	de dois
000	0	0	0	0
001	1	1	1	1
010	2	2	2	2
011	3	3	3	3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

Para representar qualquer número decimal, sem sinal em binário, utiliza-se expressão abaixo:

$$\chi_{(10)} = \sum_{i=-j}^{n-1} \beta_i 2^i \quad (6.1)$$

no qual n é o número de dígitos à esquerda do ponto (ou *radix*), j é o número de dígitos à direita do ponto, β_i é o valor do dígito binário (0 ou 1) e $\chi_{(10)}$ é o número na base 10 (decimal) o qual esta sequência binária representa. Com poucas exceções, o *hardware* para executar cálculos aritméticos trata todos os números como inteiros, ficando a cargo do projetista/programador interpretar cada um dos valores armazenados e resultantes ao fixar o *radix* na posição desejada. Dependendo da localização do ponto, uma mesma sequência binária pode representar números diferentes, como mostrados na Tabela 4. Logo, para definir conceitualmente uma variável de ponto fixo, utilizam-se dois parâmetros (PADGETT; ANDERSON, 2009; TOCCI; WIDMER; MOSS, 2003):

- número de bits para representação do algarismo decimal;
- posição do ponto na sequência binária.

Como o valor de escala utilizado pode mudar no mesmo circuito, em Engenharia, geralmente, utiliza-se a chamada notação Q para especificar a proporção de dígitos binários destinados à parte fracionária. Por exemplo, a sequência binária 01,10 tem seu *radix* representado por $Q2$. Essa notação clássica não informa qual o tamanho da

Tabela 4: Influência da posição do ponto na interpretação de uma palavra binária.

Sequência binária com ponto	Interpretação na base 10
1 1 0 1 0 1, 0 0 0	53,0
1 1 0 1 0, 1 0 0 0	26,5

palavra e é mais utilizada em microprocessadores, visto que o tamanho das palavras binárias é fixo. Com o surgimento das FPGAs, a grande flexibilidade na modelagem em ponto fixo criou a necessidade de uma notação que informasse com mais detalhes a escala utilizada para interpretação dos dados no sistema digital. A esta notação deu-se o nome de notação Q modificada. Ao representar o mesmo exemplo dado anteriormente com essa notação, teria-se $Q2.2$, onde o primeiro número indica quantos bits estão sendo interpretados como inteiros (n), e o segundo, quantos deles serão lidos como fracionários (j). A notação Q modificada será adotada por esta dissertação para representar a construção do estimador neural em aritmética de ponto fixo (PADGETT; ANDERSON, 2009).

A quantização de toda a rede neural será executada da camada de entrada para a de saída, em que a cada simulação são testadas palavras binárias de tamanhos variados, representadas em complemento de dois, avaliando a performance final da rede quantizada. O RMSe é a figura de mérito, no qual foi calculado o erro relativo (e_r) entre a rede ideal (simulada no *software* Matlab[®]) e a rede quantizada, dado pela Equação 6.2. O critério utilizado para a seleção da melhor quantização é a de que a rede quantizada deve se aproximar ao máximo da rede implementada em *software*, porém, será tolerado um erro final em cada etapa da quantização, que deve estar de acordo com a margem escolhida de 1 %. O sinal do Tilecal utilizado para testar as redes, com arquitetura e pesos encontrados no capítulo anterior, é dado em $Q10.0$ (os digitalizadores do Tilecal são de 10 bits inteiros), tanto para rede em *software* quanto para a rede quantizada, com uma ocupância de 30 %.

$$e_r = \frac{RMSe_{\text{quant}} - RMSe_{\text{sim}}}{RMSe_{\text{sim}}} \times 100 \% \quad (6.2)$$

6.1.2 PESOS DE ENTRADA

Para esta etapa de quantização apenas os pesos da camada de entrada da rede serão discretizados, mantendo-se todos os demais elementos como estruturas ideais. Ao fazer desta forma, é possível avaliar apenas o impacto da representação em ponto fixo das

constantes em análise. Na Tabela 5 são mostrados os pesos da camada de entrada para os quatro neurônios, juntamente com seus bias. Observa-se que, para representar a parte inteira, seriam necessários no mínimo 2 bits, sendo que um deles é utilizado para codificar o sinal. A precisão fracionária foi testada, variando o número de bits total a partir de 2. Este processo encontra-se ilustrado na Figura 56.

Tabela 5: Pesos da camada de entrada da rede estimadora com 5 algarismos significativos.

	Neu. 1	Neu. 2	Neu. 3	Neu. 4
w_1	0,00593	-0,00757	0,01406	0,00005
w_2	-0,01545	0,02455	-0,05172	-0,00018
w_3	0,03347	-0,06221	0,14044	0,00046
w_4	-0,07584	0,15040	-0,24859	-0,00111
w_5	0,18658	-0,24887	0,15059	0,00184
w_6	-0,31315	0,15613	-0,07865	-0,00114
w_7	0,18371	-0,08649	0,04296	0,00063
w_8	-0,09745	0,04581	-0,02139	-0,00033
w_9	0,04655	-0,02138	0,00976	0,00015
w_{10}	-0,01441	0,00725	-0,00275	-0,00005
b	-0,99390	-1,05974	-1,16549	0,13887

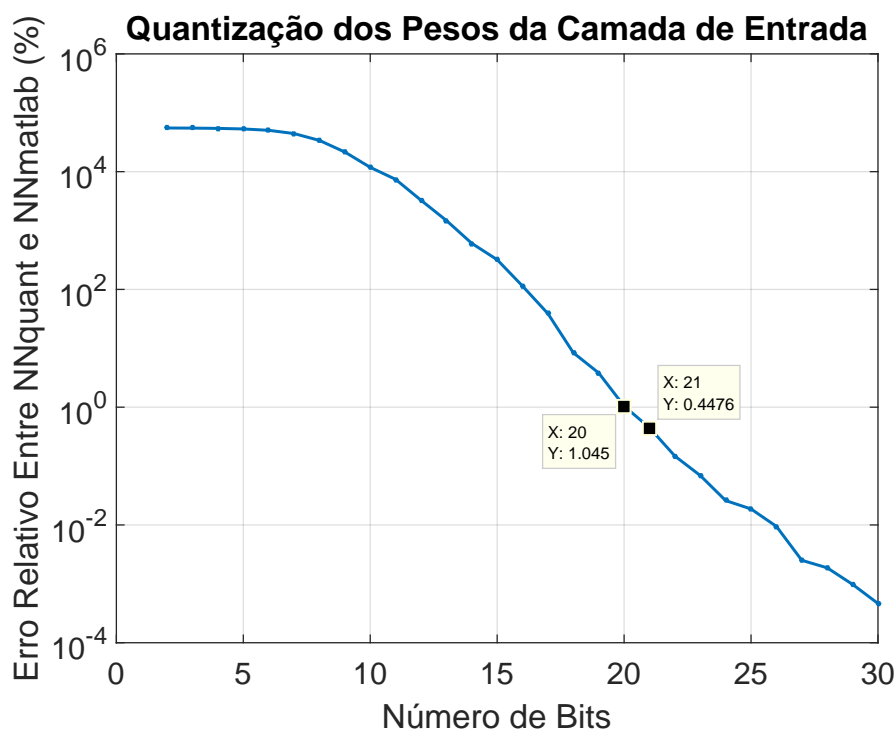


Figura 56: Quantização dos pesos de entrada.

Observou-se que o número de bits que atendeu ao critério de erro anteriormente explicitado foi de 21, resultando em uma representação em ponto fixo de $Q2.19$. No

entanto, ao analisar os pesos que ligam as entradas a cada neurônio, é possível ver claramente que os pesos sinápticos do neurônio 4 são até 1000 vezes menores que os pesos dos demais neurônios. Este fato gera uma utilização excessiva de bits para a representação dos pesos dos neurônios 1 a 3. A hipótese de que se os pesos dos neurônios 1 a 3 forem representados com o ponto e o número de bits diferentes do neurônio 4, aqueles referentes ao neurônio 4 necessitariam de mais bits fracionários, foi testada e está ilustrada na Figura 57.

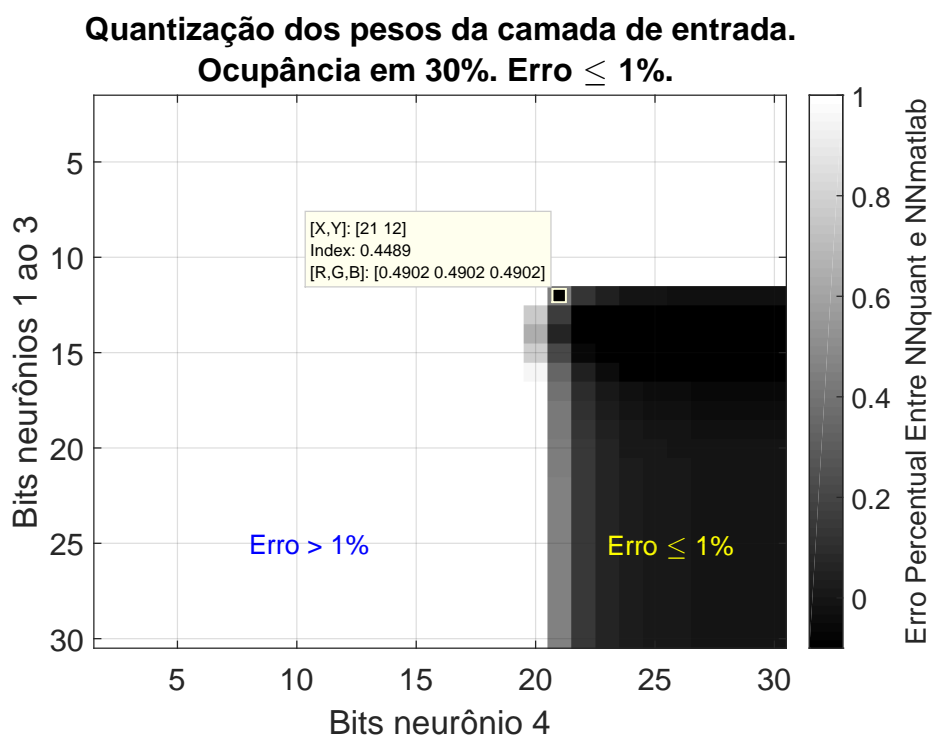


Figura 57: Quantização dos pesos da camada de entrada levando em consideração a diferença de escala entre os neurônios 1, 2 e 3 e o neurônio 4.

Pode-se observar que o número de bits necessário para representar os pesos de entrada do neurônio 4, com uma margem de erro da rede quantizada de 1 %, foi de $Q1.20$. Já para os pesos dos neurônios 1 a 3 foi de $Q2.10$. A diferença na parte inteira ocorreu pois não há pesos em que a parte inteira tem módulo maior que zero no neurônio 4. Essa separação valida a hipótese acima, surtindo uma grande redução no número de bits para essa camada da rede. Com isto, este método será utilizado também para a próxima etapa de quantização, visto que o neurônio 4 é ativado em uma região diferente dos demais (Figura 50).

6.1.3 FUNÇÃO DE ATIVAÇÃO

Grande parte da dificuldade de se implementar uma rede neural em um *hardware* dedicado vem da função de ativação. Por ser na maioria das vezes uma tangente hiperbólica, não pode ser calculada por elementos aritméticos básicos, sendo substituída por um circuito que gere um resultado aproximado. Novamente, a ideia é otimizar a dualidade área *versus* precisão (BAPTISTA; DIAS, 2012). Este problema gerou soluções variadas ao longo dos anos, como: aproximação por partes através de funções lineares (VILLANUEVA, 2005), uma tabela de pesquisa previamente endereçada (*lookup table*) (MEHER, 2010), aproximação polinomial (PANICKER; BABU, 2012) e por série de Taylor (LEE; BURGESS, 2003).

A implementação da função tangente hiperbólica através de uma *lookup table* (LUT) é a mais rápida dentre os diferentes métodos citados anteriormente. Devido ao fato dos valores da imagem da função estarem armazenados previamente em uma memória, gera-se um atraso de apenas um acesso à memória para que se tenha um resultado, tornando esta técnica bastante atrativa para implementação no contexto o qual este trabalho está inserido. Esse tempo de atraso é bem menor quando comparado à utilização de circuitos aritméticos para realizar uma aproximação polinomial ou por partes lineares. No entanto, o tamanho da LUT cresce exponencialmente à medida que uma maior precisão é requerida. Outra característica motivadora para a utilização de uma LUT é que, na maioria das aplicações, uma alta precisão da função de ativação não se faz necessária. Além disso, FPGAs modernas vêm com muitos recursos para implementação de memórias dedicadas.

Esta aplicação concentra-se na implementação da função de ativação tangente hiperbólica através de uma LUT. Para economizar recursos com o armazenamento em memória de dados redundantes, as seguintes características são extraídas da \tanh :

- é uma função ímpar, fazendo com que $f(-x) = -f(x)$.
- Para domínios $] -\infty, -3]$ e $[3, +\infty[$, a imagem da função para estes valores é aproximadamente -1 e $+1$, respectivamente.

Com base nisto, é possível realizar um tratamento nos valores de entrada e saída da LUT, de acordo com o fluxograma representado na Figura 58, objetivando armazenar em memória apenas a cauda positiva, compreendendo o domínio $[0, \text{Dom}]$. Essa lógica será implementada a partir do resultado da soma ponderada de entrada como

um circuito combinacional em hardware, visto na Seção 6.2. Resultando em valores maiores que $+3$ ou menores que -3 , a saída do neurônio será dada como $+1$ ou -1 , respectivamente, valores de saturação da tangente hiperbólica. Se o valor de entrada não satisfazer as condições anteriores e for menor que zero, seu módulo é calculado. Neste ponto do circuito, os bits deveriam ser interpretados como um endereço de memória para fazer a busca pelo valor armazenado. No entanto, como se sabe que o sinal do Tilecal possui $Q10.0$ e os pesos sinápticos da camada de entrada $Q2.10$ e $Q1.20$, utilizando a aritmética de ponto fixo, após a multiplicação de cada amostra de entrada tem-se para os neurônios 1 ao 3, $Q10.0 \times Q2.10 \rightarrow Q12.10$, e, para o neurônio 4, $Q10.0 \times Q1.20 \rightarrow Q11.20$. Após somar estes resultados com os bias, tem-se $Q17.10$ e $Q16.20$, respectivamente. Se estas palavras fossem lidas como endereços, seriam necessárias memórias com 2^{27} e 2^{36} palavras, mostrando-se um fator limitador para implementar essas LUTs. Com base nesta observação, o tamanho da memória será avaliado a partir do truncamento dos bits resultantes dessas somas ponderadas, analisando quais bits poderiam ser descartados sem causar maiores impactos na performance final da rede, como representado na Figura 59. De acordo com a lógica descrita na Figura 58, só serão necessários os truncamentos em valores que estiverem dentro do domínio da LUT. Com isso, a janela de bits é ajustada de acordo com o número de bits inteiros no domínio de cada uma das LUTs.

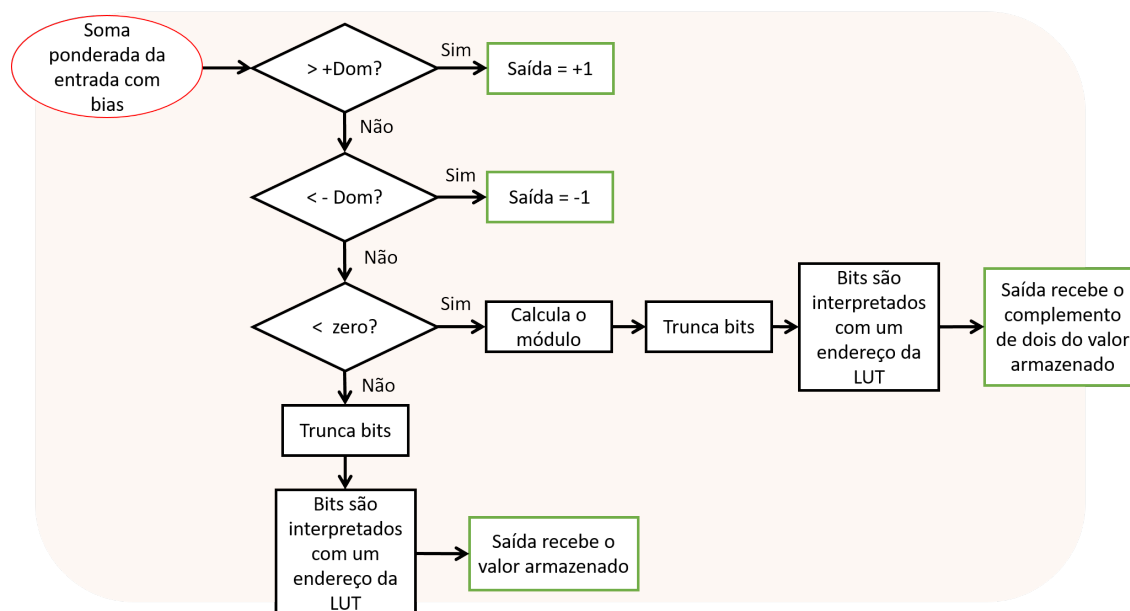


Figura 58: Lógica de decisão para leitura da LUT.

Como não haverá a necessidade de distinção entre valores positivos e negativos armazenados na memória, todos os bits inteiros serão usados para determinar a extensão do domínio (representação binária sem sinal). Como se determinou que o mesmo

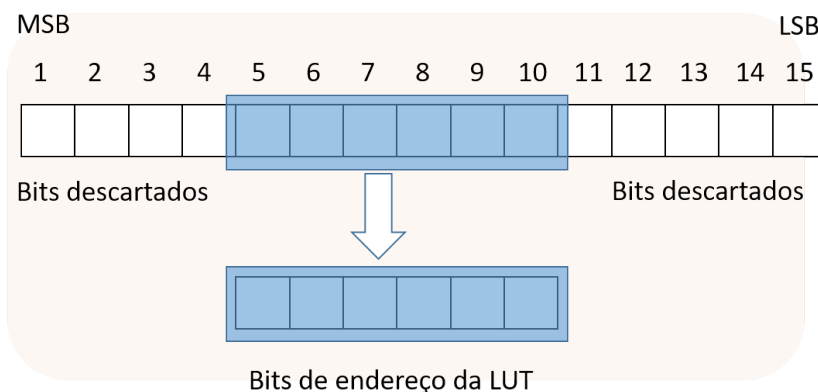


Figura 59: Exemplo de truncamento de palavra binária.

deve ir de zero a três, apenas dois bits serão necessários para representá-lo. A precisão da imagem se dá com 1 bit inteiro, visto que apenas serão usados valores positivos e menores ou iguais a 1 (saturação da tanh). A partir disso será realizada a quantização da função de ativação dos neurônios 1 ao 3 em duas perspectivas distintas: número de bits do domínio e da imagem, ilustrados na Figura 60.

Quantização da função de ativação dos neurônios 1 ao 3.

Ocupância em 30%. Erro $\leq 1\%$.

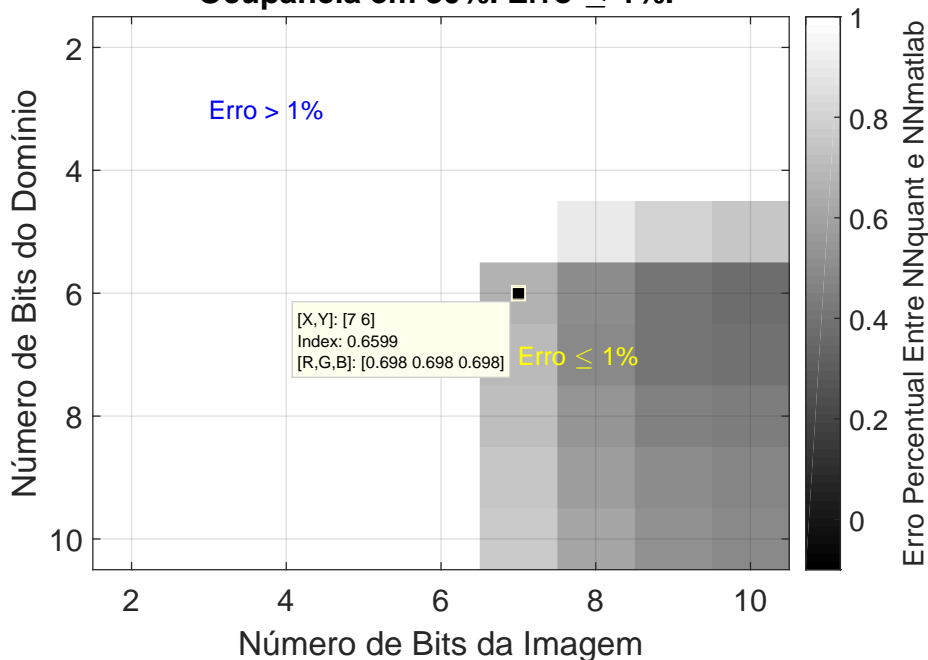


Figura 60: Quantização da função de ativação dos neurônios 1, 2 e 3.

Observa-se que o requisito de performance foi alcançado para o domínio em $Q2.4$, e para a imagem em $Q1.6$. Esta aproximação está representada na Figura 61. Nota-se que o domínio não se limita a três apenas, pois a parte fracionária estende o domínio para valores menores que quatro.

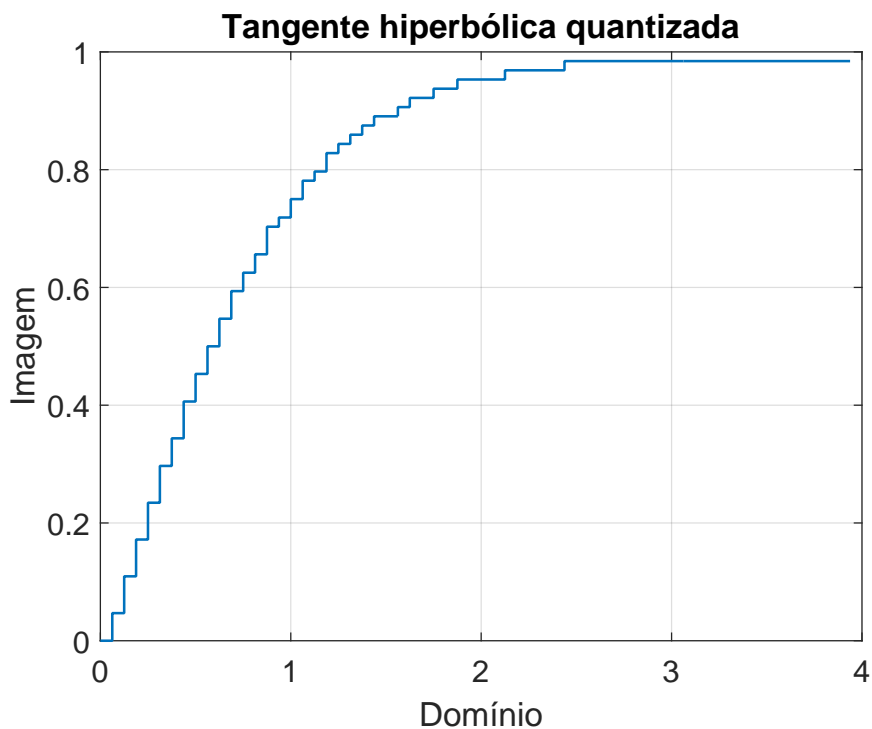


Figura 61: Função de ativação em uma LUT.

A quantização da função de ativação do neurônio 4 se dá de maneira similar aos demais, com uma pequena mudança na extensão do domínio, visto que esta só é ativada em uma pequena faixa linear da tangente hiperbólica. O domínio será limitado a 1 bit inteiro, o que dá à parte inteira uma representação de $[0, 1[$. Há a opção de não utilizar bits inteiros, o que restringiria ainda mais o domínio. No entanto, isso dá uma margem muito pequena para os valores que podem ser empregados como entrada nessa LUT, podendo gerar erros, caso o valor de entrada esteja fora do domínio. O processo de quantização, partindo de um ponto de baixa precisão, aumentando o domínio e a imagem bit a bit está representado na Figura 62.

Pode-se observar que o requisito de performance foi atingido para o domínio em $Q1.14$, e para a imagem em $Q1.14$. Esta aproximação está representada na Figura 63. Nota-se, em comparação com a Figura 61, que ao utilizar um número maior de bits para representar a função de ativação deste neurônio, mais precisa se tornou a aproximação.

A análise da quantização da função de ativação realizada nesta seção, ao fazer o uso do algoritmo descrito, resultou em uma configuração reduzida, porém, com boa performance, viabilizando a implementação dos neurônios.

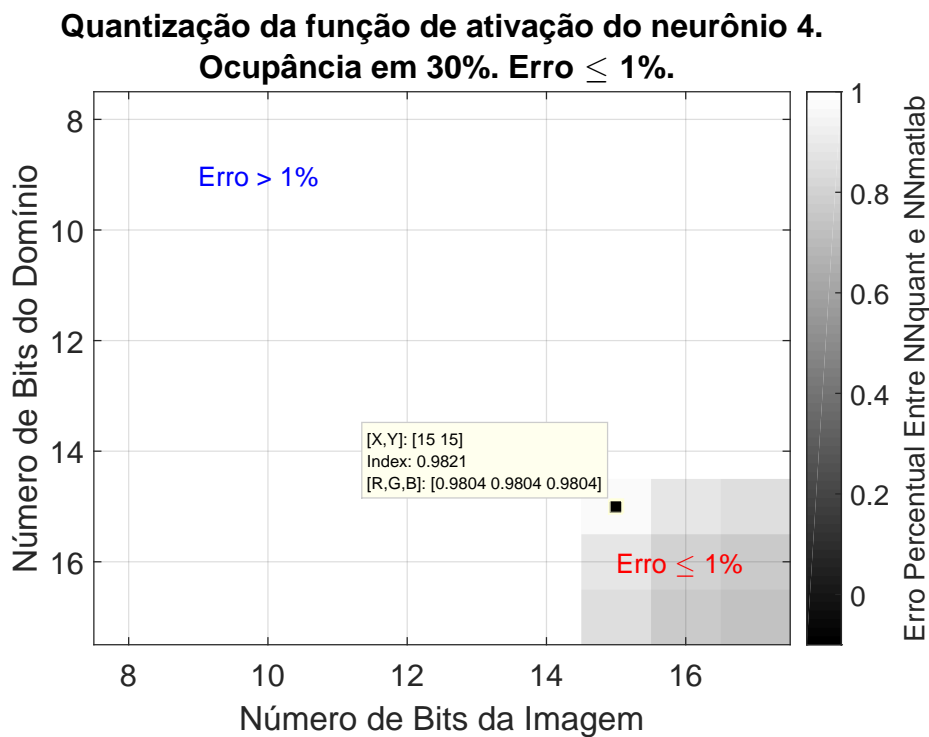


Figura 62: Quantização da função de ativação do neurônio 4.

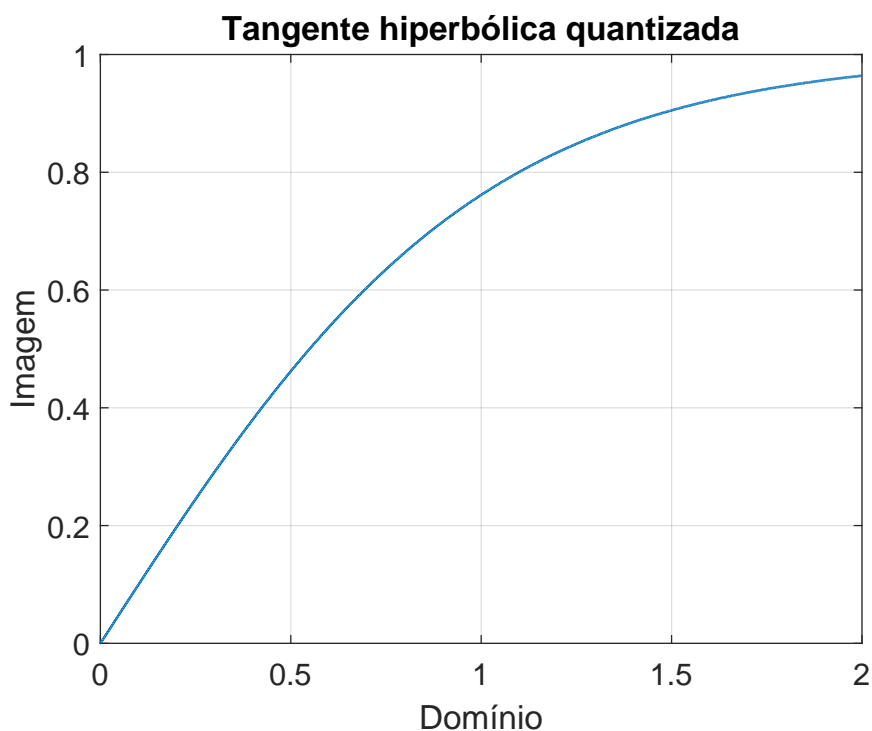


Figura 63: Função de ativação em uma LUT.

6.1.4 PESOS DA CAMADA ESCONDIDA E SAÍDA DA REDE

Seguindo a mesma metodologia da camada de entrada, os pesos sinápticos da camada escondida serão quantizados, no entanto, serão utilizados todos os parâmetros

anteriormente definidos de acordo com o critério de erro para a quantização da rede. Como são apenas quatro pesos, todos serão quantizados com o mesmo ponto e número de bits, sendo que sua distinção por faixa de valores não traria benefícios relevantes. De acordo com a Tabela 2, seriam necessários no mínimo 12 bits para representar a parte inteira do peso do neurônio 4. Visto isso, e a partir dos resultados ilustrados na Figura 64, a configuração que alcançou o critério de erro foi de $Q12.1$. O bias do combinator linear de saída foi quantizado com o mesmo ponto e número de bits do resultado da soma ponderada de saída, que é dada pela soma do valor de saída da LUT e o peso sináptico da camada escondida de cada neurônio. Para os três primeiros neurônios, obteve-se $Q1.6 \times Q12.1 \rightarrow Q13.7$, e, para o quarto neurônio, $Q1.14 \times Q12.1 \rightarrow Q13.15$. Aplicou-se uma complementação por zeros (zero *padding*) às variáveis representadas com $Q13.7$, visando adequá-las às somas da camada de saída. Com isso, o bias estará representado com $Q13.15$ e, ao ser somado com o resultado da soma ponderada, tem-se $Q15.13$.

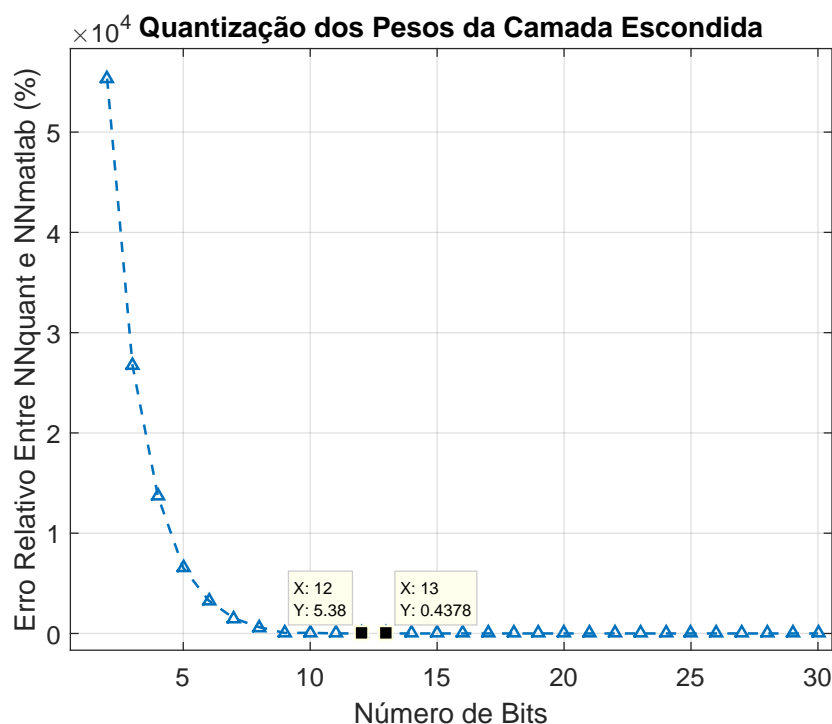


Figura 64: Quantização dos pesos da camada escondida.

Logo, a saída da rede teria 28 bits. Um rápido teste de quantização da variável de saída da rede, mostrou que com $Q10.3$ o erro entre a rede totalmente quantizada e a ideal foi de 0,26 %. A rede quantizada está representada na Figura 65, sendo implementada em uma FPGA na próxima seção.

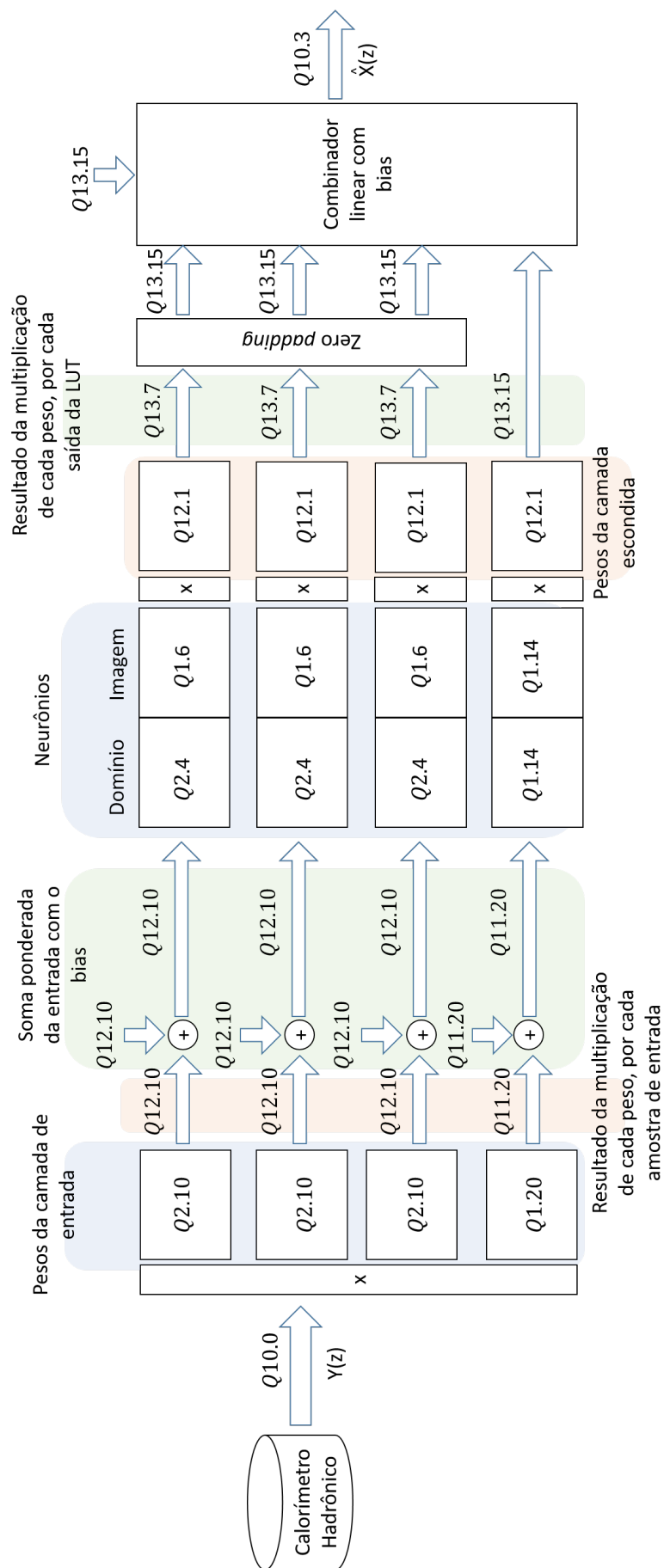


Figura 65: Representação da rede quantizada.

6.2 ESTIMADOR NEURAL IMPLEMENTADO EM FPGA

Implementar uma rede neural artificial em um *hardware* dedicado gera redução de custo, além do benefício em velocidade de processamento, visto que seu paralelismo intrínseco pode ser completamente aproveitado. No entanto, as alternativas de dispositivos físicos para suportar esse tipo de implementação devem ser flexíveis, permitindo que alterações sejam feitas de forma rápida e economicamente viável (DIAS; ANTUNES; MOTA, 2004). Em virtude destas características, a plataforma escolhida para implementação do estimador neural é a FPGA, que por consistir-se de um dispositivo reconfigurável, permite o reuso de seus componentes internos inúmeras vezes, além da praticidade com que modificações do sistema embarcado são realizadas.

De acordo com a classificação dos circuitos de integração em larga escala (do inglês, *very large-scale integration* — VLSI) ilustrada na Figura 66, as FPGAs são pertencentes a classe de dispositivos lógicos programáveis em campo (do inglês, *field-programmable logic* — FPL). De maneira geral, os FPLs são dispositivos que contêm uma grande quantidade de pequenos blocos lógicos e elementos de circuito digitais elementares, repetidamente distribuídos em campos. Estes dispositivos podem ser classificados quanto sua granularidade, ou seja, a complexidade dos blocos lógicos constituintes da placa: granularidade fina, que compreende dispositivos contendo blocos lógicos com apenas uma porta NAND e um latch, conhecida como arquitetura de “mar de portas” (do inglês, *sea of gates*). Dispositivos de média granularidade, como as FPGAs, possuem blocos lógicos que contêm pequenas tabelas com 4 ou 5 bits de entrada e até 2 bits de saída, um somador completo e um flip-flop. Já os dispositivos de alta granularidade compreendem os CPLDs (do inglês, *complex programmable logic devices*), formados pela combinação de dispositivos lógicos programáveis simples (do inglês, *simple programmable logic devices* — SPLDs), compostos de um arranjo lógico programável com portas AND/OR (MEYER-BAESE, 2007).

A tecnologia de arranjo de portas programável em campo constitui-se basicamente de blocos lógicos e de entrada e saída, chaves de interconexão e canais de roteamento. Uma matriz bidimensional é formada pelos blocos lógicos, sendo as chaves de interconexão organizadas de forma a criar canais de roteamento vertical e horizontal, como ilustrado na Figura 67. Essas chaves são programáveis, ou seja, permitem que os blocos lógicos básicos sejam conectados de maneiras distintas, de acordo com necessidade de cada aplicação. Toda a lógica que descreve o comportamento do circuito a ser implementado é feita utilizando linguagens de descrição de *hardware* (do inglês, *hardware*

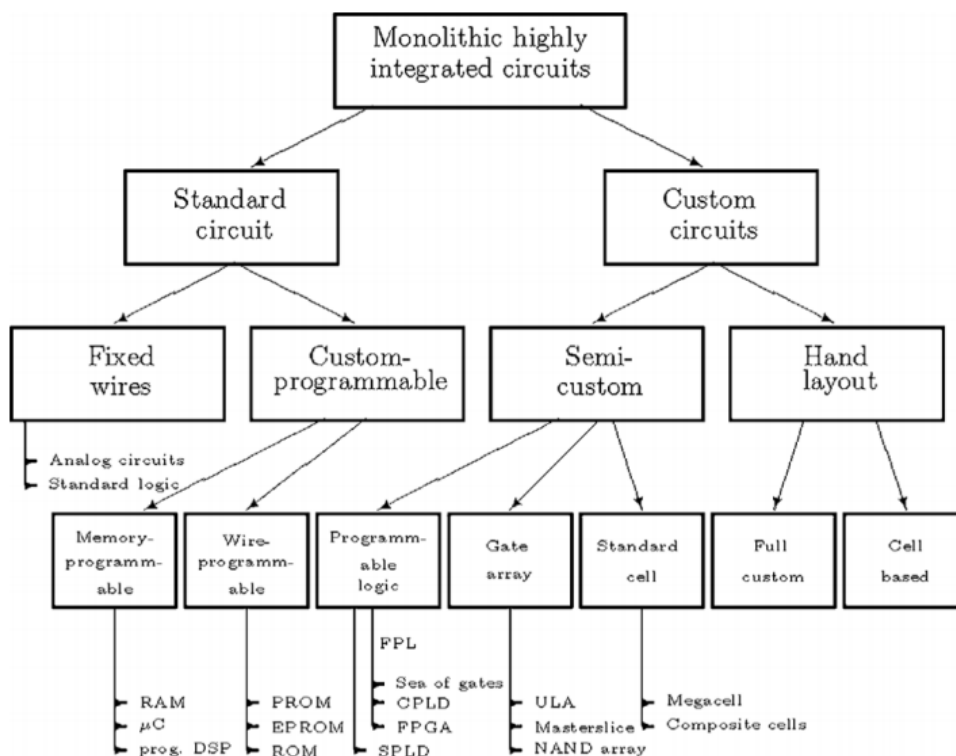


Figura 66: Classificação dos circuitos VLSI (MEYER-BAESE, 2007).

description language — HDL) como VHDL (do inglês, *Very High Speed Integrated Circuit HDL*) e Verilog, ou através de um projeto esquemático, sendo este último não recomendado para implementações mais complexas. As HDLs são recomendadas para circuitos maiores e mais complexos, dado que não há a necessidade de desenhar cada parte do mesmo de forma gráfica, possuindo a capacidade de descrever cada parte do mesmo de forma numérica e em alto nível de abstração. O projetista do circuito descreve o *hardware*, que é adequado ao modelo físico da FPGA de acordo com a ferramenta de EDA (o inglês, *eletronic design automation*) de cada fabricante. Este *software* fará a sintetização do *hardware* descrito pelo usuário, realizando o processo de roteamento entre os blocos constituintes do modelo escolhido para implementação. A ferramenta EDA também possui formas de validação do modelo através de simulações de análise lógica e temporal, antes da implementação física do circuito sintetizado (MEYER-BAESE, 2007).

Para desenvolver o circuito com a quantização propostos nesta dissertação, foram utilizadas as EDAs: Quartus II versão 15.0 *Web Edition*, para sintetização em Verilog pertencente à empresa Altera *Corporation*, e ModelSim Altera *Starter Edition* 10.3d, da empresa Mentor Graphics, para a realização das análises de simulação lógica e temporal. O modelo da FPGA utilizado foi de baixo custo (EP4CE115F29I8L), contendo

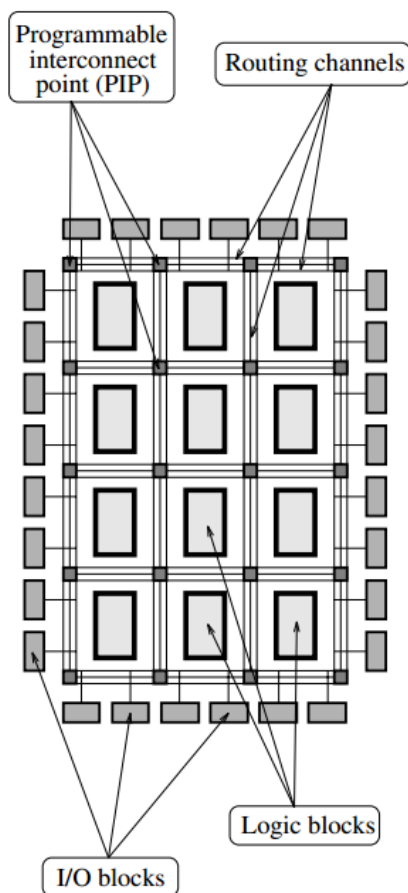


Figura 67: Classificação dos circuitos VLSI (MEYER-BAESE, 2007).

114.480 elementos lógicos com 3,888 Mbits de memória e máximo de 528 I/O (ALTERA, 2009). O *top-level* do circuito é ilustrado na Figura 68. O sinal do ADC foi atrasado em uma janela de 9 registradores de deslocamento de 10 bits cada. Esse atraso é necessário para realizar as somas ponderadas com bias (`soma_pond`) com os pesos da camada de entrada. O resultado da soma ponderada da entrada de cada neurônio (`neuronio`) é passado para um circuito (`wind`) encarregado de realizar verificações de sinal e magnitude, além da busca de endereço em uma memória (`lut`) através do janelamento da palavra binária de entrada. A saída do circuito `wind` retorna um valor que é, aproximadamente, a tangente hiperbólica do valor de saída do circuito `soma_pond`, para cada um dos neurônios. A saída do circuito de cada `neuronio` é combinada em outra soma ponderada com bias na camada escondida da rede, na qual tem como resultado uma estimativa da amplitude da amostra armazenada pelo 4º registrador de deslocamento da entrada.

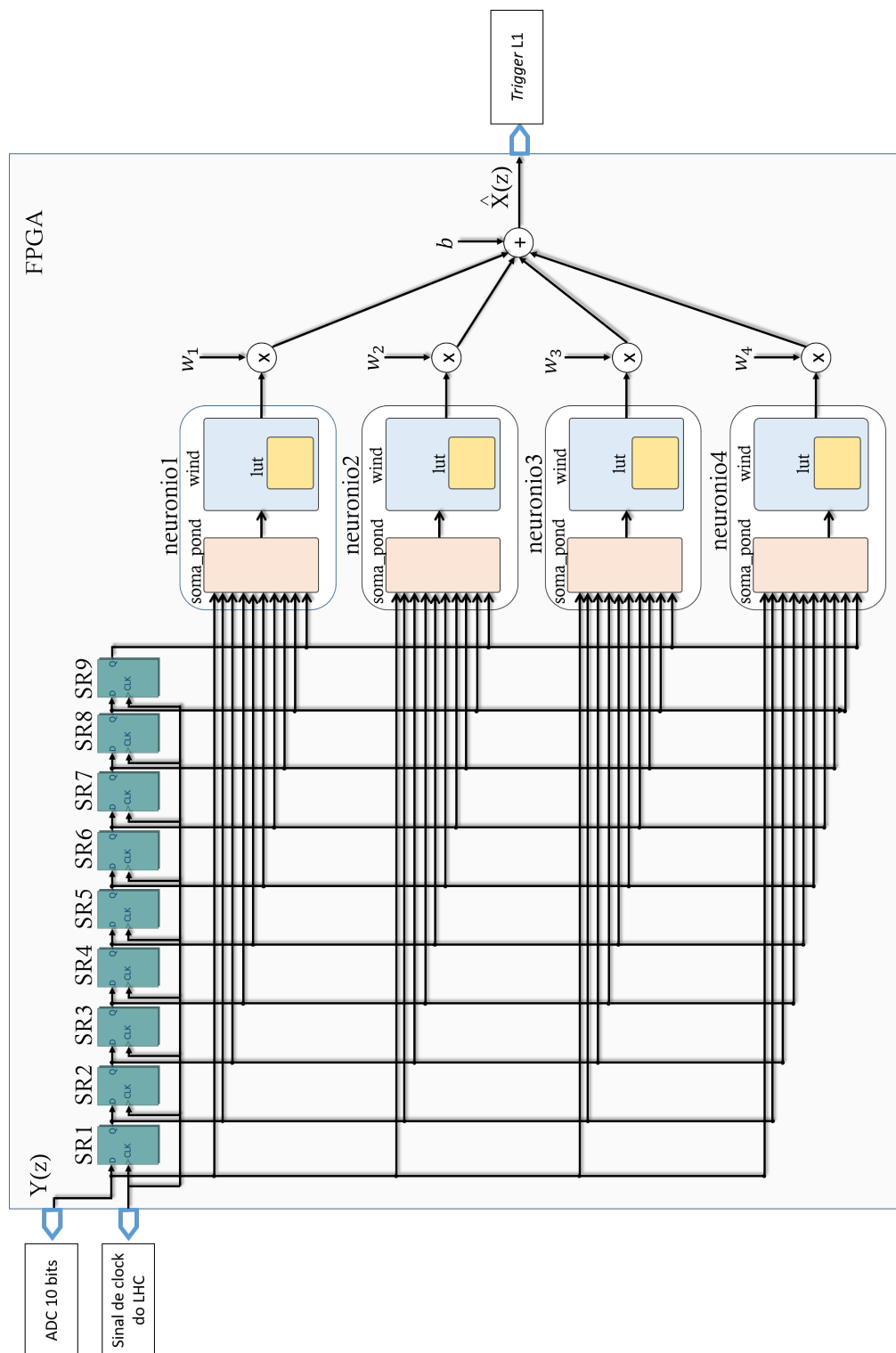


Figura 68: Arquitetura do estimador neural em FPGA.

O principal objetivo dessa implementação é atender os pré-requisitos impostos pela SROD, que são a necessidade de se possuir, em uma mesma FPGA, vários canais em paralelo, além de realizar o processamento da informação a uma taxa de 40 MHz. Inicialmente, foi testada uma rede puramente combinacional, na qual sua topologia

Tabela 6: Comparação entre as arquiteturas implementadas em FPGA.

<i>Pipelines</i>	Elementos lógicos	Fmáx (MHz)	Uso de memória (Kbits)	Latência em bunch-crossing
0	13.319	15,07	0	5
1	1.413	34,24	491,52	6
2	1.419	41,93	491,52	7

encontra-se ilustrada na Figura 68. A LUT não foi implementada em uma memória separada, mas sim utilizando elementos lógicos padrão. Além disso, a frequência máxima de processamento alcançada foi de 16,78 MHz, muito abaixo do necessário para funcionar no ambiente desejado, sendo que a a quantidade de recursos consumida torna essa implementação inviável, uma vez que as LUTs ocupam uma grande quantidade de elementos lógicos. Visto isso, utilizou-se uma arquitetura em *pipeline*. A utilização deste recurso é intensamente explorada em soluções de processamento digital de sinais, podendo ser aplicado em FPGAs sem excessiva utilização de elementos lógicos adicionais, visto que utilizam apenas *flip-flops*. Essa abordagem permite a realização de operações em paralelo, sincronizando partes do circuito. Para cada *pipeline*, aumenta-se um ciclo de *clock* necessário à obter o resultado do processamento. No entanto, esta técnica possibilita a implementação desse circuito à taxas mais altas, além de possibilitar a utilização de memórias internas síncronas para implementação das LUTs (MEYER-BAESE, 2007). Foram propostas duas arquiteturas com 1 e 2 *pipelines*, no qual os resultados das simulações encontram-se na Tabela 6.

Os resultados apresentados acima foram alcançados com a utilização dos módulos de multiplicação disponíveis pelo modelo utilizado, os quais contêm 9 bits cada, sendo necessários 59 destes dos 532 disponíveis. É possível observar que foram necessários dois *pipelines* para que o circuito funcionasse na frequência requerida (40 MHz). Vale ressaltar também que ao implementar a LUT em uma memória dedicada, reduziu-se a utilização de recursos lógicos em, aproximadamente, 7 vezes. De acordo com o relatado por Carrio et al (2013), os módulos da sROD incluem duas FPGAs da fabricante Xilinx, Série 7. Na FPGA de alta performance Virtex-7 XC7VX485T, serão realizadas, principalmente, a separação dos sinais das PMTs e decodificação do protocolo de comunicação da eletrônica de FE. O sinal decodificado será enviado à uma FPGA Kintex-7 XC7K420T, encarregada pelo pré-processamento do primeiro nível de *trigger*, sendo a plataforma na qual se destina a implementação do método proposto nesta dissertação. Este modelo contém 416.960 elementos lógicos, 901 pinos de I/O e 34,38 MBbits de memória total, se mostrando suficiente para a operação vários canais em paralelo do circuito com 2 *pipelines* com relativa manutenção da taxa de processamento (XILINX,

2016). Além disso, a família Kintex permite a implementação de circuitos com atrasos menores do que o da família testada Cyclone IV E, de modo que espera-se uma frequência máxima de operação superior a 41.1 MHz na FPGA da sROD. Deve-se ressaltar que o teste foi feito na Cyclone pelo fato da universidade não possuir a licença para teste nas FPGAs da Xilinx. Porém, determinando a correta operação em uma FPGA de baixo custo, espera-se o seu funcionamento também em uma FPGA superior. Com os resultados da Tabela 6 pode-se concluir que a aritmética de ponto fixo utilizada manteve o desempenho dentro da tolerância esperada, respeitando os requisitos de latência do L1, com um uso relativamente baixo de LEs.

7 CONSIDERAÇÕES FINAIS

Esta dissertação teve como ambiente de pesquisa o Tilecal, que devido ao programa de atualização do LHC, terá células com alta ocupância de sinal. Em virtude disto, a ocorrência de *pile-up* se intensificará neste período, comprometendo o método de estimação de energia atualmente implementado. Fornecer uma informação da amplitude da energia depositada nas células do calorímetro é de suma importância para a reconstrução final dos eventos selecionados pelo sistema de *trigger*. O pré-processamento do sinal de leitura das células do calorímetro tem como objetivo principal atenuar ao máximo os efeitos inerentes ao sistema de instrumentação, provendo uma estimação por *bunch-crossing* que otimize a seleção dos eventos de relevância para a física realizada pelo primeiro nível de *trigger*.

A analogia na qual este trabalho se baseou, que é a interpretação do calorímetro como um canal de comunicação que distorce a informação da energia depositada nas células, permitiu que se propusesse uma técnica de equalização não-linear utilizando redes neurais artificiais. No Capítulo 5 foi desenvolvido o método de equalização, englobando os principais parâmetros que envolveram os ajustes dos pesos sinápticos da rede em uma arquitetura que aliasse boa performance e baixa complexidade computacional, em comparação com o método atualmente implementado e os propostos em Duarte (2015). Ao fim, pôde-se concluir que o equalizador não-linear apresentou melhor performance em todos os cenários de ocupância testados, ao compará-lo com os métodos lineares. É ressaltado também o fato de que o estimador neural conseguiu reduzir a latência necessária para se obter a estimação de uma amostra de 13, para 7 *bunch-crossings*, ao ser comparado com o filtro FIR adaptativo.

A complexidade computacional foi avaliada no Capítulo 6, no qual foi proposta uma implementação em FPGA da rede neural, com arquitetura e pesos sinápticos otimizados no capítulo anterior. Este equalizador deve prover uma estimativa da energia à uma taxa de 40 MHz, além do processo de quantização dos elementos constituintes do estimador neural proporcionar a implementação de vários estimadores neurais (canais)

em paralelo na mesma FPGA. Estas condições impostas pela sROD na Fase-II são analisadas em um *hardware* de baixo custo, na qual a taxa de dados desejada foi alcançada com sucesso. O número de elementos lógicos utilizados pela rede neural quantizada foi baixo o suficiente para a implementação na sROD, que possui uma FPGA com mais recursos computacionais, atingindo, assim, os objetivos propostos por este trabalho.

A maior parte dos recursos utilizados pela rede neural implementada em FPGA é advindo das quatro somas ponderadas de entrada. Constatado isto, destina-se a trabalhos futuros a implementação de um núcleo para processamento das somas ponderadas funcionando a 160 MHz, no qual seriam executadas estas quatro somas em série, trazendo uma redução de recursos lógicos da FPGA de aproximadamente quatro vezes. Outra alternativa, seria a utilização de apenas dois núcleos funcionando a 80 MHz cada, neste caso, reduzindo o número de elementos lógicos pela metade. Estas alternativas poderiam ser realizadas com relativa facilidade por modelos de FPGA de alta performance, como a família Kintex, da fabricante Xilinx. Como outro trabalho futuro, fica para se realizar uma análise precisa do número de canais em paralelo a FPGA da Xilinx citada, verificando a viabilidade da proposta no ambiente real.

REFERÊNCIAS

- AAD, G. et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, Elsevier, v. 716, n. 1, p. 1–29, 2012.
- AAD, G. et al. The ATLAS experiment at the cern large hadron collider. *Journal of Instrumentation*, IOP Publishing, v. 3, n. 08, p. S08003, 2008.
- AAD, G. et al. Atlas pixel detector electronics and sensors. *Journal of Instrumentation*, IOP Publishing, v. 3, n. 07, p. P07007, 2008.
- ALICE, C. The alice experiment at the cern lhc. *Journal of Instrumentation*, IOP Publishing, v. 3, n. 08, p. S08002, 2008.
- ALTERA. *Cyclone IV FPGAs*. 2009. Disponível em: <https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/po/ss-civfpga.pdf>.
- ANDERSON, J. A. A simple neural network generating an interactive memory. *Mathematical biosciences*, Elsevier, v. 14, n. 3-4, p. 197–220, 1972.
- ANDERSON, K. et al. Design of the front-end analog electronics for the atlas tile calorimeter. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier, v. 551, n. 2, p. 469–476, 2005.
- ANDRADE FILHO, L. M. *Detecção e Reconstrução de Raios Cósmicos usando Calorimetria de Altas Energias*. Tese (Doutorado) — COPPE UFRJ, 2009.
- ASK, S. et al. The atlas central level-1 trigger logic and ttc system. *Journal of Instrumentation*, v. 3, n. 08, p. P08002, 2008. Disponível em: <<http://stacks.iop.org/1748-0221/3/i=08/a=P08002>>.
- ATLAS, C. The atlas experiment at the cern large hadron collider. *Journal of Instrumentation*, v. 3, n. 08, p. S08003, 2008. Disponível em: <<http://stacks.iop.org/1748-0221/3/i=08/a=S08003>>.
- ATLAS, C. Acessado em Dezembro de 2016. Disponível em: <<https://cds.cern.ch/record/1095924>>.
- BAPTISTA, D.; DIAS, M. On the implementation of different hyperbolic tangent solutions in fpga. In: *CONTROLO 2012*. [S.l.: s.n.], 2012.
- BATTITI, R. First-and second-order methods for learning: between steepest descent and newton’s method. *Neural computation*, MIT Press, v. 4, n. 2, p. 141–166, 1992.
- BRAGA, A. de P.; FERREIRA, A. C. P. de L.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: LTC Editora, 2007.

BURSE, K.; YADAV, R. N.; SHRIVASTAVA, S. Channel equalization using neural networks: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 40, n. 3, p. 352–357, 2010.

CARRIO, F. et al. *The sROD Module for the ATLAS Tile Calorimeter Phase-II Upgrade Demonstrator*. Geneva, Nov 2013. Disponível em: <<https://cds.cern.ch/record/1628753>>.

CARRIÓ, F. et al. The srod module for the atlas tile calorimeter phase-2 upgrade demonstrator (poster). In: *TWEPP Conf.* [S.l.: s.n.], 2013.

CERN, C. *About CERN*. Acessado em Dezembro de 2016. Disponível em: <<http://home.cern/about>>.

CERN, C. *ATLAS - The Experiment*. Acessado em Dezembro de 2016. Disponível em: <<http://home.cern/about/experiments/atlas>>.

CERN, C. *ATLAS - The Inner Detector*. Acessado em Dezembro de 2016. Disponível em: <<http://atlas.cern/discover/detector/inner-detector>>.

CERN, C. *ATLAS - The Physics*. Acessado em Dezembro de 2016. Disponível em: <<http://atlas.cern/discover/physics>>.

CERN, C. *ATLAS Coordination System*. Acessado em Dezembro de 2016. Disponível em: <http://atlas.web.cern.ch/Atlas/GROUPS/MUON/TDR/pdf_final/naming.pdf>.

CERN, C. *ATLAS Inner Detector*. Acessado em Dezembro de 2016. Disponível em: <<http://cds.cern.ch/images/CERN-GE-0803014-01>>.

CERN, C. *How an Accelerator Works*. Acessado em Dezembro de 2016. Disponível em: <<http://home.cern/about/how-accelerator-works>>.

CERN, C. *Member States*. Acessado em Dezembro de 2016. Disponível em: <<http://home.cern/about/member-states>>.

CERN, C. *How a detector works*. Acessado em Janeiro de 2017. Disponível em: <<https://home.cern/about/how-detector-works>>.

CERQUEIRA, A.; SEIXAS, J.; CALÔBA, L. Analog system for building the first-level triggering signal provided by the hadronic calorimeter of atlas detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier, v. 570, n. 1, p. 117–125, 2007.

CERQUEIRA, A. S. Atlas tile calorimeter readout electronics upgrade program for the high luminosity lhc. *arXiv preprint arXiv:1305.0859*, 2013.

CERQUEIRA, A. S. *Tile Calorimeter Upgrade Program for the Luminosity Increasing at the LHC*. Geneva, Oct 2015. Disponível em: <<https://cds.cern.ch/record/2056628>>.

CHANG, P.-R.; LIN, C.; YEH, B.-F. Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks. *The Journal of the Acoustical Society of America*, ASA, v. 95, n. 6, p. 3400–3408, 1994.

- CHAPMAN, J. ATLAS simulation computing performance and pile-up simulation in ATLAS. *LPCC*, 2011.
- CMS, C. The cms experiment at the cern lhc. *JInst*, Citeseer, v. 3, n. 08, p. S08004, 2008.
- DIAS, F. M.; ANTUNES, A.; MOTA, A. M. Artificial neural networks: a review of commercial hardware. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 17, n. 8, p. 945–952, 2004.
- DRACHMAN, D. A. Do we have brain to spare? *Neurology*, AAN Enterprises, v. 64, n. 12, p. 2004–2005, 2005.
- DUARTE, J. B. d. S. *Estudo de Técnicas de Deconvolução para Reconstrução de Energia online no Calorímetro Hadrônico do ATLAS*. Dissertação (Mestrado) — PPEE UFJF, 2015.
- DUARTE, J. B. d. S. Técnicas de deconvolução aplicadas à estimação de energia online em calorimetria de altas energias em condições de alta taxa de eventos. In: SBA/PPGEE-UFES. *Anais do XXI Congresso Brasileiro de Automática*. [S.l.], 2016. p. 2612–2617.
- ERDOGMUS, D. et al. Linear-least-squares initialization of multilayer perceptrons through backpropagation of the desired response. *IEEE Transactions on Neural Networks*, IEEE, v. 16, n. 2, p. 325–337, 2005.
- EVANS, L.; BRYANT, P. LHC machine. *Journal of Instrumentation*, IOP Publishing, v. 3, n. 08, p. S08001, 2008.
- EVERITT, B. *The Cambridge Dictionary of Statistics*. Cambridge, UK New York: Cambridge University Press, 1998. ISBN 0521593468.
- FAHLMAN, S. E. An empirical study of learning speed in back-propagation networks. 1988.
- FILHO, E. F. S.; SEIXAS, J. M. de; CALOBA, L. P. Online neural filtering operating over segmented discriminating components. In: IEEE. *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*. [S.l.], 2008. p. 530–533.
- GALLANT, A. R.; WHITE, H. There exists a neural network that does not make avoidable mistakes. In: *Proceedings of the Second Annual IEEE Conference on Neural Networks, San Diego, CA, I*. [S.l.: s.n.], 1988.
- GLANZ, F. H.; MILLER, W. T. Deconvolution and nonlinear inverse filtering using a neural network. In: IEEE. *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. [S.l.], 1989. p. 2349–2352.
- GRAHN, K.-J. *ATLAS Calorimetry : Hadronic Calibration Studies*. [S.l.]: KTH, 2008. v, 136 p. (Trita-FYS, 2008:43). QC 20101111.
- GROSSBERG, S. Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biological cybernetics*, Springer, v. 23, n. 3, p. 121–134, 1976.

GUIRAUD, L. Various phases of the instrumentation of the ATLAS barrel tile calorimeter at CERN: Phase III. Oct 1999. Disponível em: <<http://cds.cern.ch/record/40849>>.

HAGAN, M. T. et al. *Neural network design*. [S.l.]: PWS publishing company Boston, 1996.

HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, IEEE, v. 5, n. 6, p. 989–993, 1994.

HAYKIN, S. *Adaptive Filter Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0-13-322760-X.

HAYKIN, S. *Neural networks, a comprehensive foundation*. [S.l.]: Pearson Education, 1999.

HAYKIN, S. *Communication Systems*. [S.l.]: 4th ed. New York: Wiley, 2001.

HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 1026–1034.

HEBB, D. O. *The organization of behavior: A neuropsychological approach*. [S.l.]: John Wiley & Sons, 1949.

HERR, W.; MURATORI, B. Concept of luminosity. In: *proceedings of CERN Accelerator School*. [S.l.: s.n.], 2003. v. 361.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982.

HU, Y. H.; HWANG, J.-N. *Handbook of neural network signal processing*. [S.l.]: CRC press, 2001.

KAY, S. M. Fundamentals of statistical signal processing, volume i: estimation theory. Prentice Hall, 1993.

KAY, S. M. *Fundamentals of statistical signal processing: Detection theory, vol. 1*. [S.l.]: Prentice Hall Upper Saddle River, NJ, USA:, 1998.

KLIMEK, P. *Signal reconstruction performance with the ATLAS Hadronic Tile Calorimeter*. Geneva, Aug 2012. Disponível em: <<https://cds.cern.ch/record/1473499>>.

KOHONEN, T. Correlation matrix memories. *IEEE transactions on computers*, IEEE, v. 100, n. 4, p. 353–359, 1972.

LECUN, Y. A. et al. Efficient backprop. In: *Neural networks: Tricks of the trade*. [S.l.]: Springer, 2012. p. 9–48.

LEE, B.; BURGESS, N. Some results on taylor-series function approximation on fpga. In: IEEE. *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*. [S.l.], 2003. v. 2, p. 2198–2202.

- LHCB, C. The lhc detector at the lhc. *Journal of instrumentation*, IOP Publishing, v. 3, n. 08, p. S08005, 2008.
- MACHADO, A. B. *Neuroanatomia funcional*. [S.l.]: Atheneu, 2005.
- MAEDA, J.; ATLAS, C. et al. *The ATLAS Trigger System: Ready for Run-2*. [S.l.], 2015.
- MAGUIRE, E. A. et al. Navigation-related structural change in the hippocampi of taxi drivers. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 97, n. 8, p. 4398–4403, 2000.
- MARSHALL, Z.; COLLABORATION, A. et al. Simulation of pile-up in the atlas experiment. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2014. v. 513, n. 2, p. 022024.
- MCCLELLAND, J. L.; RUMELHART, D. E.; HINTON, G. E. The appeal of parallel distributed processing. *MIT Press, Cambridge MA*, p. 3–44, 1986.
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- MEHER, P. K. An optimized lookup-table for the evaluation of sigmoid function for artificial neural networks. In: IEEE. *VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP*. [S.l.], 2010. p. 91–95.
- MELIS, C. D. *The twenty two Member States of CERN*. Acessado em Dezembro de 2016. Disponível em: <<http://cds.cern.ch/images/OPEN-PHO-CHART-2016-005-1>>.
- MEYER-BAESE, U. *Digital Signal Processing with Field Programmable Gate Arrays*. 3rd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2007. ISBN 3540726128, 9783540726128.
- MINSKY, M.; PAPERT, S. *Perceptrons*. MIT press, 1969.
- MITRA, S. K. *Digital Signal Processing: A Computer Based Approach*. [S.l.]: McGraw-Hill, 1998.
- MURATA, N.; YOSHIZAWA, S.; AMARI, S.-i. Network information criterion-determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, IEEE, v. 5, n. 6, p. 865–872, 1994.
- NAKAHAMA, Y. The atlas trigger system: Ready for run-2. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2015. v. 664, n. 8, p. 082037.
- NIKIFOROU, N. Performance of the ATLAS Liquid Argon Calorimeter after three years of LHC operation and plans for a future upgrade. In: *Proceedings, 3rd International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA 2013): Marseille, France, June 23-27, 2013*. [s.n.], 2013. Disponível em: <<https://inspirehep.net/record/1240499/files/arXiv:1306.6756.pdf>>.
- PADGETT, W. T.; ANDERSON, D. V. *Fixed-Point Signal Processing*. [S.l.]: Morgan & Claypool Publishers, 2009.

PANICKER, M.; BABU, C. Efficient fpga implementation of sigmoid and bipolar sigmoid activation functions for multilayer perceptrons. *IOSR Journal of Engineering*, p. 1352–1356, 2012.

PEARLMUTTER, B. Gradient descent: Second order momentum and saturating error. In: *NIPS*. [S.l.: s.n.], 1991. v. 2, p. 887–894.

PEETERS, S. J. M. *The ATLAS semiconductor tracker endcap*. Tese (Doutorado) — NIKHEF, 2003.

PEQUENAO, J. Computer Generated image of the ATLAS calorimeter. Mar 2008. Disponível em: <<https://cds.cern.ch/record/1095927>>.

PEQUENAO, J. Event Cross Section in a computer generated image of the ATLAS detector. Mar 2008. Disponível em: <<https://cds.cern.ch/record/1096081>>.

PERALVA, B. S.-M. *Detecção de sinais e estimação de energia para calorimetria de altas energias*. Dissertação (Mestrado) — PPEE UFJF, 2012.

PERALVA, B. S.-M. *Reconstrução de Energia para Calorímetros Finamente Segmentados*. Tese (Doutorado) — PPEE UFJF, 2015.

PERALVA, B. S.-M. et al. Implementacao em fpga de detectores baseados em filtragem casada e redes neurais em ambientes de física experimental de altas energias. *Anais do XIX Congresso Brasileiro de Automãtica*, p. 4690–4696, 2012.

PERALVA, B. S. M. et al. *The TileCal Energy Reconstruction for Collision Data Using the Matched Filter*. Geneva, Nov 2013. Disponível em: <<https://cds.cern.ch/record/1629575>>.

PIANORI, E. *The ATLAS Trigger System: Past, Present and Future*. [S.l.], 2014.

RAGHAVENDRA, K.; TRIPATHY, A. K. An efficient channel equalizer using artificial neural networks. *Neural Network World*, Institute of Computer Science, v. 16, n. 4, p. 357, 2006.

RIEDMILLER, M. Rprop-description and implementation details. Citeseer, 1994.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation*. [S.l.], 1985.

SAVICH, A. W.; MOUSSA, M.; AREIBI, S. The impact of arithmetic representation on implementing mlp-bp on fpgas: A study. *IEEE transactions on neural networks*, IEEE, v. 18, n. 1, p. 240–252, 2007.

SCHIAVI, C. ATLAS High-Level Trigger Algorithms for Run-2 Data Taking. Apr 2015. Disponível em: <<https://cds.cern.ch/record/2007824>>.

SCHLAGER, G. et al. *The Energy Response of the ATLAS Calorimeter System*. Tese (Doutorado) — Vienna, Tech. U., Vienna, 2006. Presented on 04 Dec 2006. Disponível em: <<http://cds.cern.ch/record/998248>>.

SIMIONI, E. et al. Upgrade of the atlas level-1 trigger with event topology information. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2015. v. 664, n. 8, p. 082052.

TANG, F. et al. Design of the front-end readout electronics for atlas tile calorimeter at the slhc. *IEEE Transactions on Nuclear Science*, IEEE, v. 60, n. 2, p. 1255–1259, 2013.

TEAM, A. The four main LHC experiments. Jun 1999. Disponível em: <<https://cds.cern.ch/record/40525>>.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. *Sistemas digitais: princípios e aplicações*. [S.l.]: Prentice Hall, 2003.

TREES, H. L. V. *Detection, Estimation and Modulation Theory, Part I*. [S.l.]: Wiley, 2001.

VALERO, A. et al. Dsp online algorithms for the atlas tilecal read-out drivers. *IEEE Transactions on Nuclear Science*, IEEE, v. 55, n. 1, p. 158–164, 2008.

VILLANUEVA, J. M. M. *Determinação de aproximação linear por partes de funções não lineares para sistemas embarcados utilizando algoritmos genéticos*. Dissertação (Mestrado) — Programa de pós-graduação em engenharia de eletricidade UFMA, 2005.

WATTERSON, J. W. An optimum multilayer perceptron neural receiver for signal detection. *IEEE Transactions on Neural Networks*, IEEE, v. 1, n. 4, p. 298–300, 1990.

WOLF, D. F.; ROMERO, R. A.; MARQUES, E. Using embedded processors in hardware models of artificial neural networks. In: *V Simposio Brasileiro de automação inteligente, Brasil*. [S.l.: s.n.], 2001.

WOODS, K.; BOWYER, K. W. Generating roc curves for artificial neural networks. *IEEE Transactions on medical imaging*, IEEE, v. 16, n. 3, p. 329–337, 1997.

XILINX. *7 Series FPGAs Data Sheet: Overview*. 2016. Disponível em: <https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf>.