



**UNIVERSIDADE FEDERAL DO PARÁ  
CURSO DE LICENCIATURA PLENA EM MATEMÁTICA  
FACULDADE DE MATEMÁTICA**

**REDES NEURAIS ARTIFICIAIS E SUAS APLICABILIDADES:  
MODELAGEM DO VALOR DA TEMPERATURA MÁXIMA NA  
CIDADE DE CASTANHAL**

**FERNANDA KELLY MARTINS DOS SANTOS**

**CASTANHAL- PARÁ**

**2018**



**UNIVERSIDADE FEDERAL DO PARÁ**  
**CURSO DE LICENCIATURA PLENA EM MATEMÁTICA**  
**FACULDADE DE MATEMÁTICA**

**REDES NEURAIS ARTIFICIAIS E SUAS APLICABILIDADES:  
MODELAGEM DO VALOR DA TEMPERATURA MÁXIMA NA  
CIDADE DE CASTANHAL**

**FERNANDA KELLY MARTINS DOS SANTOS**

**PROFESSOR ORIENTADOR Dr. EDILBERTO OLIVEIRA ROZAL**

**CASTANHAL-PARÁ**

**2018**

**FERNANDA KELLY MARTINS DOS SANTOS**

**REDES NEURAIS ARTIFICIAIS E SUAS APLICABILIDADES:  
MODELAGEM DO VALOR DA TEMPERATURA MÁXIMA NA  
CIDADE DE CASTANHAL**

Trabalho de Conclusão de Curso apresentado a Faculdade de Matemática da Universidade Federal do Pará, campus de Castanhal, como pré-requisito para a obtenção do Grau de Licenciado Pleno em Matemática. Orientador: Prof. Dr. Edilberto Oliveira Rozal.

**CASTANHAL - PARÁ**

**2018**

**FERNANDA KELLY MARTINS DOS SANTOS**

**REDES NEURAIS ARTIFICIAIS E SUAS APLICABILIDADES:  
MODELAGEM DO VALOR DA TEMPERATURA MAXIMA NA  
CIDADE DE CASTANHAL**

Este trabalho de Conclusão de Curso foi julgado e aprovado para a obtenção do Grau de Licenciado Plena em Matemática, no Curso de Licenciatura em Matemática, da Universidade Federal do Pará.

Castanhal, \_\_\_\_\_ de Junho de 2018.

---

**Coordenador do Curso de Matemática**

BANCA EXAMINADORA

---

Prof. Dr. Edilberto Oliveira Rozal

Orientador

---

Prof. Dr. Arthur da Costa Almeida

Membro

---

Prof. Msc. Marcos Vinicius Orguen Gouvea

Membro

## **DEDICATÓRIA**

Dedico este trabalho a todos que de alguma forma contribuíram para elaboração desta pesquisa.

## **AGRADECIMENTOS**

Agradeço a Deus por tudo que ele realizou em minha vida e há de realizar, agradeço também aos meus pais que ajudaram financeiramente para a conclusão do curso, agradeço aos meus tios, Gleison Martins e Gal Martins, pelo companheirismo e por sempre estarem presente. Agradeço meu namorado Cris pelo apoio e incentivo. Sou grata também ao meu orientador Edilberto Oliveira Rozal por todo auxílio no meu trabalho de conclusão e no decorrer do curso.

“Devemos admitir com humildade que, ao passo que os números são puramente produtos de nossas mentes, o espaço tem uma realidade fora de nossas mentes, de modo que não podemos descrever completamente suas propriedades a priori”

**Carl Friedrich Gauss**

## RESUMO

No atual contexto de aquecimento global, torna-se relevante analisar e modelar a Temperatura máxima do ar visando identificar os possíveis sinais ou evidências de tendências das mudanças climáticas, neste caso abordamos a cidade de Castanhal (Pará). A inteligência artificial tem sido utilizada com êxito para modelar dados climáticos, por meio das Redes Neurais Artificiais (RNA). Entretanto, ainda é escassa a utilização das RNA para obtenção da previsão de temperatura disponível. Com isso, esta pesquisa apresenta estudos da série temporal da temperatura máxima (°C) acumulada no tempo com a utilização de redes neurais na tentativa de gerar modelos de previsões. Foi utilizada uma série obtida junto à Estação Meteorológica da cidade de Castanhal/PA de 2013 a 2017 de registros diários da temperatura máxima (°C) que originou a série acumulada diária. O objetivo neste estudo foi testar o desempenho, para previsões, dos modelos de redes neurais artificiais através do algoritmo de treinamento, Backpropagation. Os resultados mostram que o erro médio dos valores simulados fica em torno de 0,6 %, mostrando um desempenho aceitável do método aplicado.

**Palavras-chave:** Algoritmo Backpropagation. Redes neurais artificiais. Temperatura máxima.

## ABSTRACT

In the current context of global warming, it is relevant to analyze and model the maximum air temperature in order to identify possible signs or evidences of trends in climate change, in this case we approach the city of Castanhal (Pará). Artificial intelligence has been used successfully to model climatic data through Artificial Neural Networks (ANNs). However, the use of ANNs to obtain the available temperature forecast is still scarce. Thus, this research presents studies of the time series of the maximum temperature ( $^{\circ}$  C) accumulated in time with the use of neural networks in the attempt to generate prediction models. A series obtained from the Meteorological Station of the city of Castanhal / PA from 2013 to 2017 of daily records of the maximum temperature ( $^{\circ}$  C) that originated the daily cumulative series was used. The objective of this study was to test the performance, for predictions, of artificial neural network models through the training algorithm, Backpropagation. The results show that the mean error of the simulated values is around 0.6%, showing an acceptable performance of the applied method.

**Keywords:** Backpropagation algorithm. Artificial neural networks. Maximum temperature

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	<b>12</b>
<b>1. INTELIGÊNCIA ARTIFICIAL (RESUMO HISTÓRICO – REDES NEURAIS)</b> .....	<b>14</b>
1.1. Neurônio Biológico.....	16
1.2. Neurônio Artificial.....	18
<b>1.2.1. Funções de Ativação</b> .....	<b>21</b>
<b>2. ARQUITETURA DAS REDES NEURAIS ARTIFICIAIS E PROCESSOS DE TREINAMENTO</b> .....	<b>27</b>
2.1. Principais Arquiteturas de Redes Neurais Artificiais.....	27
<b>2.1.1. Arquitetura <i>feedforward</i> de Camada Simples</b> .....	<b>29</b>
<b>2.1.2. Arquitetura <i>feedforward</i> de Camada Múltiplas</b> .....	<b>29</b>
<b>2.1.3. Arquitetura Recorrente ou Realimentada</b> .....	<b>30</b>
2.2. Processos de Treinamento.....	32
<b>2.2.1. Treinamento supervisionado</b> .....	<b>32</b>
<b>2.2.2. Treinamento não supervisionado</b> .....	<b>33</b>
<b>3. REDES PERCEPTRON</b> .....	<b>35</b>
3.1. Princípio de Funcionamento de Perceptron.....	37
3.2. Análise Matemática do Perceptron.....	38
3.3. Princípio de Funcionamento do Adaline.....	40
3.4. Processo de Treinamento do Adaline.....	40
3.5. Comparação entre o Processo de Treinamento de Adaline e Perceptron.....	42
3.6. Redes Perceptron Multicamadas.....	42
<b>3.6.1. Princípio de Funcionamento de Perceptron Multicamadas</b> .....	<b>43</b>
<b>3.6.2. Processo de Treinamento do Perceptron Multicamadas</b> .....	<b>44</b>
<b>3.6.3. Derivação do Algoritmo Backpropagation</b> .....	<b>45</b>
<b>3.6.4. Implementação do Algoritmo Backpropagation</b> .....	<b>47</b>
<b>4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS</b> .....	<b>49</b>
4.1. Materiais e métodos.....	49
4.2. Treinamento da RNA.....	50
4.3 Tipo de rede.....	51
4.4 Estrutura da rede.....	51
<b>5 CONCLUSÃO</b> .....	<b>55</b>
<b>6 REFERÊNCIAS</b> .....	<b>56</b>

## LISTA DE FIGURAS

Figura 1 – Esquema dos constituintes da célula neural.....	17
Figura 2 – Esquema dos constituintes da rede neural artificial.....	20
Figura 3 – Função de ativação rampa simétrica. ....	22
Figura 4 – Função degrau.....	22
Figura 5 – Função degrau bipolar.....	23
Figura 6 – Função gaussiana.....	24
Figura 7 – Função linear.....	24
Figura 8 – Função tangente hiperbólica.....	25
Figura 9 – Função logística.....	26
Figura 10 – Rede feedforward camada única.....	29
Figura 11 – Rede feedforward múltiplas camadas.....	30
Figura 12 – Rede realimentada.....	31
Figura 13 – Perceptrons de Rosenblatt.....	35
Figura 14 – Rede perceptron.....	36
Figura 15 - Fronteira de separação do neurônio de entrada.....	38
Figura 16 - Fronteira não linearmente separável.....	39
Figura 17 - Perceptron de duas entradas.....	39
Figura 18 - Configuração do neurônio com derivação do algoritmo de Backpropagation.....	45
Figura 19 – Vista aérea da cidade de Castanhal/PA.....	49
Figura 20 – Janela de comandos NNTool.....	50
Figura 21 – Topologia de uma das redes simuladas com 1 neurônio na camada oculta.....	52
Figura 22 – Modelo simulado e dados de campo para rede simulada com 5 neurônios na camada oculta.....	52
Figura 23 – Propagação do erro para rede simulada com 5 neurônios na camada oculta.....	53
Figura 24 – Modelo simulado e dados de campo para rede simulada com 30 neurônios na camada oculta.....	53
Figura 25 – Propagação do erro para rede simulada com 30 neurônios na camada oculta.....	53
Figura 26 – Modelo simulado e dados de campo para rede simulada com 60 neurônios na camada oculta.....	54
Figura 27 – Propagação do erro para rede simulada com 60 neurônios na camada oculta.....	54

## INTRODUÇÃO

O conhecimento acerca da temperatura máxima ( $^{\circ}\text{C}$ ) de determinada região é de grande importância para toda a população que compõe aquela localidade; principalmente para os profissionais cujas atividades estão sujeitas a oscilações desta variável, notadamente para o produtor agrícola, uma vez que ele está intimamente relacionado a quantidade de água, que é elemento essencial para o crescimento das plantas e a radiação solar desempenha importante papel na fotossíntese e, portanto, na produção. Essa importância se torna maior nas regiões tropicais úmidas, como é o caso da cidade de Castanhal/PA, onde ao contrário das regiões fora dos trópicos, em que a estação de crescimento das plantas é regulada pela temperatura. No estado do Pará, mais precisamente na cidade de Castanhal, altas temperaturas têm sido caracterizadas envolvendo principalmente montantes anuais e mensais associados à evapotranspiração de referência para fins de planejamento agrícola ou a eventos resultantes de fenômenos de interação atmosfera-oceano, conhecidos como El Niño.

A escassez de estudos sobre a distribuição climática na região nordeste paraense, é uma consequência da dificuldade encontrada pelos pesquisadores em trabalhar com séries históricas reduzidas e uma densidade baixa de estações de coleta espalhadas nesta região. Tal fato dificulta a estudos relativos à sazonalidade das chuvas, além de impedir uma minuciosa análise estatística de dados. Neste sentido, através da obtenção de dados da temperatura máxima ( $^{\circ}\text{C}$ ) coletados em uma única estação localizada na cidade de Castanhal/PA, decidiu-se criar um modelo em redes neurais com treinamento supervisionado com o algoritmo backpropagation para simular a temperatura máxima ( $^{\circ}\text{C}$ ) nesta cidade. Os resultados mostram que o erro médio dos valores simulados é satisfatório, mostrando um desempenho aceitável do método aplicado.

O objetivo geral desta pesquisa é identificar se para a base de dados em questão, os métodos de Redes Neurais Artificiais, com o algoritmo de Backpropagation, são adequados para modelar adequadamente a série temporal da temperatura máxima ( $^{\circ}\text{C}$ ) acumulada no tempo.

Para tal, os objetivos específicos são:

- Testar o desempenho, para previsões, dos modelos de redes neurais artificiais através do algoritmo de treinamento, Backpropagation.
- Construir seis configurações diversas de redes com 1, 5, 10, 20, 30,60;
- Apontar dos seis sistemas propostos, qual a rede que apresenta menor erro;

Este trabalho está composto por quatro capítulos: O capítulo 1 apresenta um breve resumo histórico acerca da rede neural biológica e também artificial; no capítulo 2 é apresentada a arquitetura e funcionamento da rede neural artificial; no capítulo 3, é apresentada a rede perceptron; no capítulo 4 são apresentados estudos da série temporal da temperatura máxima (°C) acumulada no tempo com a utilização de redes neurais na tentativa de gerar modelos de previsões onde foi utilizada uma série obtida junto à Estação Meteorológica da cidade de Castanhal/PA de 2013 a 2017 de registros diários da temperatura máxima (°C) que originou a série acumulada diária; neste capítulo são apresentados os resultados de qual rede é mais adequada para o estudo de caso.

## 1. INTELIGÊNCIA ARTIFICIAL (RESUMO HISTÓRICO – REDES NEURAIS)

Redes neurais artificiais (RNA) podem ser consideradas como uma metodologia para resolver problemas característicos da inteligência artificial. Neste caso, não se trata apenas de programar um computador digital e fazê-lo imitar um comportamento inteligente (resolver problemas matemáticos, manter diálogo, jogar xadrez e outros procedimentos que podem ser considerados como inteligentes), mas sim, elaborar um computador que modelando os circuitos cerebrais também possua seus próprios circuitos de maneira que se possa ver um comportamento inteligente emergindo, realizando tarefas, errando e também (na maioria das vezes) excedendo seu professor. Dessa maneira, estas redes neurais artificiais poderão inventar suas próprias representações interiores e apresentar comportamento inesperado.

Nas últimas décadas, a evolução de diferentes tecnologias de modelagem proporcionou uma melhor reprodução dos fenômenos naturais por meio, também, das redes neurais artificiais (RNA). Em consequência de novos fundamentos teóricos e evolução tecnologia, ressurgiu o interesse científico pela temática proporcionando avanços no desenvolvimento no estudo na área de inteligência artificial com o uso da computação intensiva. A seguir, apresenta-se um breve histórico dos estudos em redes neurais, ressaltando alguns conceitos e teorias consideradas proeminentes para este trabalho.

Em 1943, Mcculloch & Pitts (1943) desenvolveram uma máquina inspirada no cérebro humano e um modelo matemático de neurônio biológico artificial designado *psychon*. Este modelo, mesmo que muito simples, trouxe uma grande contribuição para os debates sobre a construção dos primeiros computadores digitais, consentindo a elaboração dos modelos matemáticos de dispositivos artificiais iniciais, que tinham a intenção de buscar analogias biológicas. Foi considerado o marco inicial do estudo de redes neurais, entretanto, este modelo não era capaz de exercer uma de seus principais trabalhos: o aprendizado.

Em 1949, Donald O. Hebb no livro “The Organization of Behavior” (A organização inteligente) definiu o conceito de atualização de pesos sinápticos. Suas ideias não eram completamente novas, porém Hebb foi o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios.

A construção do primeiro neuro computador, denominado Snark, foi em 1951, por Marvin Lee Minsky. O Snark, operava bem, ajustando os seus pesos automaticamente, mas ele não executava qualquer função de processamento de informação interessante, porém serviu de inspiração para estruturas futuras.

Em 1958, surgiu o primeiro neuro computador a obter sucesso, criado por Frank Rosenblatt, Charles Wightman e outros. Devido suas contribuições técnicas estes pesquisadores são designados como fundadores da neuro computação na forma em que se tem atualmente. As suas pesquisas apoiaram os modelos do tipo perceptron (redes de um nível) e MLP (Perceptrons de múltiplas camadas). Seu interesse inicial para a criação do perceptron era o reconhecimento de padrões. Redes muito similares, chamadas “adalines”, foram inventadas por volta da mesma época por Widrow e Hoff (1960-1962, p.54,55).

Entretanto, em 1969, Minsky e Papert na obra “An Introduction to computational geometry”, mostraram matematicamente que estes modelos, na forma como estavam, não eram capazes de aprender a função lógica do OU Exclusivo, por exemplo. O teorema se aplicava apenas aos problemas cuja estrutura é capaz de aprender. Esta constatação impactou negativamente as pesquisas que vinham sendo realizadas sobre este assunto nas décadas de 60 e 70.

Em 1986, pesquisadores desenvolveram algoritmos para o reajuste dos pesos das redes neurais multicamadas, a teoria de redes em multinível treinadas com algoritmo de aprendizado por retro propagação (*Backpropagation*) foram baseadas em *perceptrons* de Rosenblatt. Este método, tornou-se conhecido após a publicação do livro “Parallel Distributed Processing” editado por J. L. McClelland e D. E. Rumelhat (1986a – 1986b)

A partir da descoberta do *Backpropagation* o interesse na linha de redes neurais voltou a vigorar. Nos últimos anos, a maioria das teorias sobre redes neurais transformaram se em aplicações, algumas dedicadas a comercialização. Aumentou as atividades de pesquisa e novas áreas de atuação das redes neurais têm surgido, a saber: reconhecimento de padrões, processamento de imagem e voz, sistema de potência, diagnósticos médicos e previsões de series temporais.

### 1.1. Neurônio Biológico

A maioria das funções de controle do corpo é proporcionada pelo sistema nervoso e sistema endócrino. O sistema endócrino tem atuação mais lenta, controla muitas funções do metabolismo do corpo. Em contrapartida, o sistema nervoso (em relação ao cérebro), monitora as reações rápidas do corpo, como uma contração muscular, controla a velocidade e equilibra o percentual de secreção de muitas glândulas endócrinas (BARRETO, 2002).

Estima-se que o cérebro humano possui, cerca de 10 bilhões de neurônios. São eles as células mais distintas do organismo, pois exibem a maior complexidade estrutural e funcional. Durante o período da vida do embrião incidem as divisões de células nervosas imprescindíveis à vida humana e, em virtude disso, nenhuma diversa célula procriará após o nascimento do bebê, conservando-se, portanto, sua quantidade idêntica sempre, ocorrendo somente alterações de volume e prolongamento de conexões com outras células (determinados pesquisadores atuais têm defendido que neurônios podem se procriar) (TAFNER et al, 1995);

O neurônio é a célula elementar do sistema nervoso cerebral, sua função sob determinadas condições de operação, limita-se em conduzir impulsos (estímulos elétricos ocorridos de reações físico-químicas) (SILVA et al, 2010). Os neurônios são divididos em três seções: o corpo da célula (também conhecido como soma), os dendritos, e o axônio, cada um com funções específicas porém elementares.

O papel dos dendritos é receber as informações, os impulsos nervosos, provenientes de outro neurônio e conduzi-las até o corpo celular. Na soma, é processada e novos impulsos são gerados. Esses estímulos são transmitidos a outros neurônios, passando do axônio até os dendritos dos neurônios seguintes. Sinapse é o ponto de contato entre a terminação axônica de um neurônio e o dendrito. É exatamente pelas sinapses que os neurônios se ligam funcionalmente, formando a rede neural. As sinapses trabalham como válvulas, são elas capazes de controlar a difusão dos impulsos entre os nodos e rede neural. O resultado da sinapse é variável, por isso a possibilidade de adaptação dos neurônios (BRAGA et al, 2000).

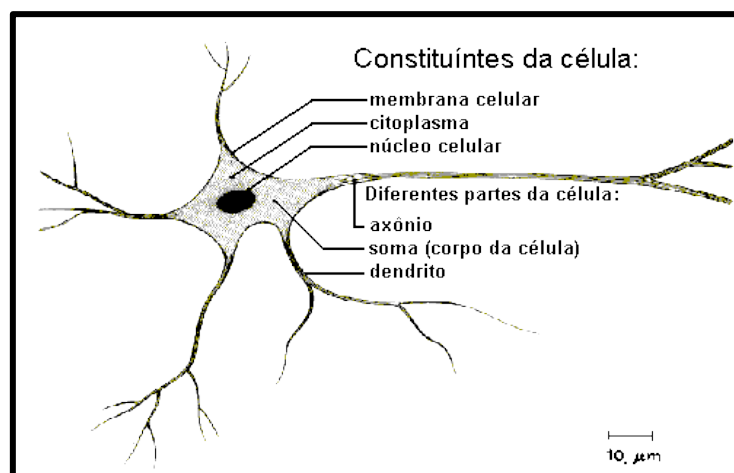
Sendo assim, com base no estudo de Barreto (2002), o uso de um órgão favorece seu desenvolvimento (seguindo um princípio biológico válido), isto é, cada vez que uma sinapse é ativada ou consegue ativar outro neurônio a quantidade de neurotransmissores liberados eleva na seguinte vez que o neurônio for ativado. Isto representa uma ampliação da conexão entre os dois neurônios, processo denominado facilitação. Cada neurônio tem de 1.000 a 10.000 sinapses e pode receber informação de próximo 1.000 outros neurônios. O método de facilitação motivou a Lei de Hebb: A intensidade de uma conexão sináptica entre dois neurônios cresce quando os dois neurônios estão excitados ao mesmo tempo. Nota-se que a Lei de Hebb é básica de grande parte dos algoritmos de aprendizagem de RNA.

Segundo Tatibana e Kaetsu (2012), os principais componentes do neurônio biológico são:

- Os dendritos, que tem por função, receber os estímulos transmitidos pelos outros neurônios;
- O corpo de neurônio, também chamado de Soma, que é responsável por coletar e combinar informações vindas de outros neurônios;
- E finalmente o axônio, que é constituído de uma fibra tubular que pode alcançar até alguns metros, e é responsável por transmitir os estímulos para outras células.

A Figura 1 ilustra os principais componentes de um neurônio natural.

Figura 1 – Esquema dos constituintes da célula neural.



Fonte: Tatibana e Kaetsu 2012).

Em suma, ainda que as atividades desenvolvidas pelos neurônios biológicos sejam simples, tais subsídios, operando em conjunto são os principais responsáveis por todos os processamentos adimplidos e gerenciados pelo cérebro humano.

## 1.2. Neurônio Artificial

A partir do modelo de sistemas nervosos biológicos foi elaborado a estrutura das redes neurais artificiais. Visto que os neurônios artificiais (ou elemento processador), são modelos simplificados se comparados ao neurônio biológico que possui estrutura e funcionamento complexo. O modelo mais bem aceito, conhecido como *Perceptron*, foi proposto por McCulloch e Pitts (1943), o qual implementa de modo simplificado os componentes e o funcionamento de um neurônio biológico. Era um dispositivo binário, bem como a saída do neurônio poderia ser ativo ou não, e as múltiplas entradas tinham um ganho arbitrário, podendo ser excitatórias ou inibitórias. A saída do neurônio era determinada pelo cálculo da soma ponderada das entradas com os respectivos ganhos como fatores de ponderação.

Do mesmo modo que o neurônio biológico, o neurônio artificial possui um ou mais sinais de início e apenas um sinal de saída. Os dados podem ser recebidos através de sensores ou de outros neurônios artificiais que englobam as Redes Neurais Artificiais. Estes sinais são verificados e enviados para a saída. Já os sinais de entrada devem surgir até o neurônio simultaneamente, ou seja, todas as informações devem chegar ao mesmo tempo no núcleo do neurônio. Em outras palavras, Barreto (2002) ressalta a ideia dizendo:

Quando o potencial de ação se propaga pelo axônio, chega a uma de suas terminações. Aí ele provoca modificações na membrana destas terminações, as sinapses. Isto permite a liberação de moléculas, de vários tipos com o nome genérico de neurotransmissores que se difundem no espaço entre o terminal do axônio e um outro neurônio, geralmente o terminal de um dendrito. Moléculas de neurotransmissor ao se colar ao dendrito provocam uma modificação na membrana deste que acaba, algumas vezes, provocando um potencial de ação, outras vezes, dificultando seu aparecimento. Este potencial de ação, se criado, se propaga até a soma do neurônio que recebeu o sinal

alterando sua frequência de disparo (Barreto, 2002, p.13). dessa forma, mudando valores destas conexões muda-se o comportamento da rede. E estas mudanças de comportamento representam um aprendizado da rede.

Uma característica de grande importância do neurônio artificial é o peso (ou peso sináptico), os pesos sinápticos são representados pela letra  $w$  (*weight*) e concebem o grau de importância de que determinada entrada possui se tratando ao neurônio em questão. O valor do peso é modificado em função da intensidade do sinal de entrada pelo processo de aprendizagem.

Os diferentes sinais de entrada são originados do meio externo, representados pelo conjunto  $\{X_1, X_2, \dots, X_n\}$  que são semelhantes aos impulsos elétricos externos no neurônio biológico. Já os pesos sinapses são vistos matematicamente, como vetores de valores  $[W_1, W_2, \dots, W_n]$  para um neurônio, ou uma matriz de pesos, para um conjunto de neurônios. Esses são ponderações exercidas pelas junções sinápticas do modelo biológico.

De forma equivalente, a importância das individuais entradas  $\{X_1\}$  do neurônio é então executada por meio de suas multiplicações pelos respectivos pesos sinápticos  $\{W_1\}$ , ponderando-se, portanto, todas as informações externas que chegam ao neurônio. Dessa forma, torna-se possível verificar que a saída do modelo celular artificial, denominada por  $u$ , é a soma ponderada de suas entradas (Silva et al, 2010, p.33).

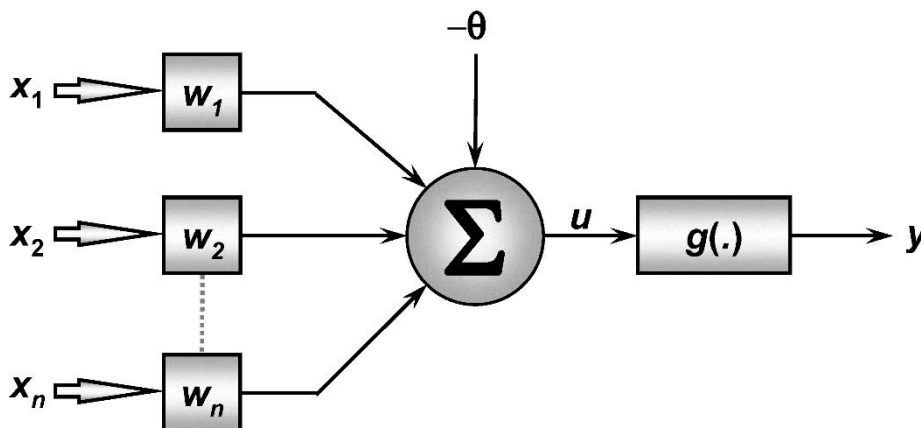
A próxima tarefa a ser executada pelo neurônio é a de analisar se o resultado da soma entre o produto dos sinais de entrada pelos referentes pesos sinápticos atingiu ou não o valor predeterminado, denominado de limiar. Se o valor calculado conseguiu alcançar o limiar, ele é repassado adiante através da saída. De outro modo, se o valor não conseguiu alcançar o limiar, o sinal não será transferido. Esse processo de verificação é conhecido como função de transferência.

Além da função de transferência, existe a função de ativação que tem como principal tarefa, suceder um nível de ativação dentro do próprio neurônio. Em alguns modelos simples de redes neurais artificiais, a função de ativação pode ser a própria função de soma das entradas ponderadas do neurônio. Entretanto, em modelos mais elaborados, a função de ativação pode ter um

processamento atribuído, no qual por exemplo, a utilização de um valor prévio de saída como a entrada para o próprio neurônio, servindo de “auto excitação” para o mesmo (TAFNER, 1996).

Nesse espelho, cada neurônio da rede pode ser produzido conforme a Figura 2 representada abaixo.

Figura 2 – Esquema dos constituintes da rede neural artificial.



(fonte: SILVA et al, 2010, p.34).

Para Silva et al (2010), o neurônio artificial é constituído de sete elementos básicos, são eles:

1. *Sinais de entrada*  $\{X_1, X_2, \dots, X_n\}$ , são os sinais advindos do meio externo e que representam os valores assumidos pelas variáveis de uma aplicação específica. Os sinais de entrada são usualmente normalizados visando incrementar a eficiência computacional dos algoritmos de aprendizagem;
2. *Pesos sinápticos*  $\{W_1, W_2, \dots, W_n\}$ , são os valores que serviram para ponderar cada uma das variáveis de entrada da rede, permitindo-se quantificar as suas relevâncias em relação á funcionalidade do respectivo neurônio;
3. *Combinador linear*  $\{\Sigma\}$ , sua função é agregar os sinais de entrada que foram ponderados pelos respectivos pesos sinápticos a fim de produzir um valor de potencial de ativação;
4. *Limiar de ativação*  $\{\theta\}$  é uma variável que especifica qual será o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção à saída do neurônio.

5. *Potencial de ativação*  $\{u\}$ , é o resultado produzido pela diferença do valor produzido entre o combinador linear e o limiar de ativação. Se tal valor é positivo, ou seja,  $u \geq \theta$  então o neurônio produz um potencial excitatório, caso contrário, o potencial será inibitório;
6. *Função de ativação*  $\{g\}$ , seu objetivo é limitar a saída do neurônio dentro de um intervalo de valores razoáveis a serem assumidos pela sua própria imagem funcional;
7. *Sinal de saída*  $\{y\}$  consiste no valor final produzido pelo neurônio em relação a um determinado conjunto de sinais de entrada, podendo ser também utilizado por outros neurônios que estão sequencialmente interligados.

Em uma rede neural as referências a serem considerados são os pesos sinápticos e o polarizador. Como em cada neurônio chega a soma ponderada de todas as entradas, então o polarizador surgirá associado a uma entrada fixa +1 ou -1.

### 1.2.1. Funções de Ativação

Ainda sobre o que se encontra na estrutura interna de cada neurônio, sobre a função de ativação, podem ser divididas em dois grupos principais, a saber: função parcialmente diferenciável e função totalmente diferenciável, tendo em vista todo o domínio da função.

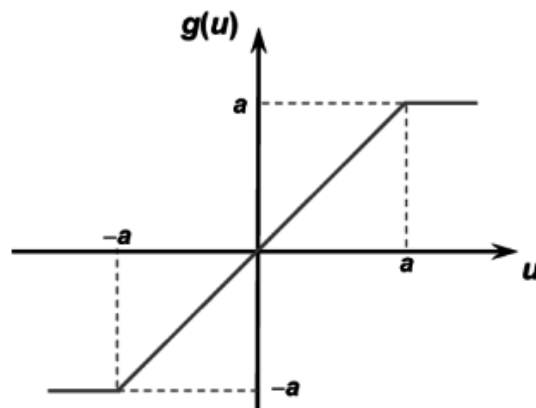
Silva et al (2010) relata que as funções que possui pontos cujas derivadas de primeira ordem são inexistentes, são denominadas de função parcialmente diferenciável. Sendo que as três principais funções dessa ordem são: *função rampa simétrica, função degrau e função degrau bipolar.*

- 1) *Função Rampa simétrica*: os valores devolutos são iguais aos próprios valores dos potenciais de ativação quando os mesmos estão definidos no intervalo  $[-a, a]$ . Em caso contrário, restringem-se aos respectivos valores. Matematicamente tem a forma:

$$g(u) = \begin{cases} a, & \text{se } u < -a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u > a \end{cases}$$

A representação do gráfico da função rampa simétrica é mostrada na Figura 3.

Figura 3 – Função de ativação rampa simétrica.



(fonte: SILVA et al, 2010, p.38).

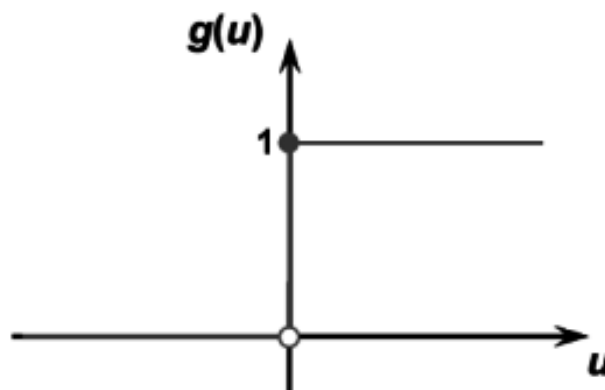
- 2) *Função degrau* (ou *heavyside bard limiter*): o efeito produzido pela aplicação da função degrau obterá valores unitários positivos se o potencial de ativação do neurônio em questão for maior ou igual a zero, de outro modo, o resultado terá valores nulos. Então:

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases}$$

Onde  $u$  é o potencial de ativação da função.

O gráfico dessa função é representado pela Figura 4.

Figura 4 – Função degrau.



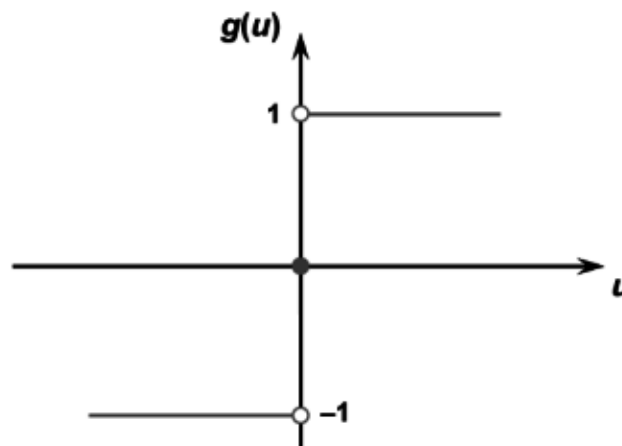
(fonte: SILVA et al, 2010, p.38).

3) *Função degrau bipolar* ou *função sinal* (*symmetric hard limiter*): A resposta obtida pelo uso desta terá valores unitários positivos quando o potencial de ativação for maior que zero, assumirá valor nulo quando o potencial também for nulo e valores unitários negativos quando o potencial de ativação do neurônio for menor que zero. Logo, matematicamente tem-se:

$$g(u) = \begin{cases} 1, & \text{se } u > 0 \\ 0, & \text{se } u = 0 \\ -1, & \text{se } u < 0 \end{cases}$$

O modelo do gráfico da função degrau bipolar é representado pela Figura 5.

Figura 5 – Função degrau bipolar.



(fonte: SILVA et al, 2010, p.37).

Além disso, a função degrau bipolar pode ser aproximado pela função a seguir, em alguns problemas envolvendo classificação de padrões.

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases}$$

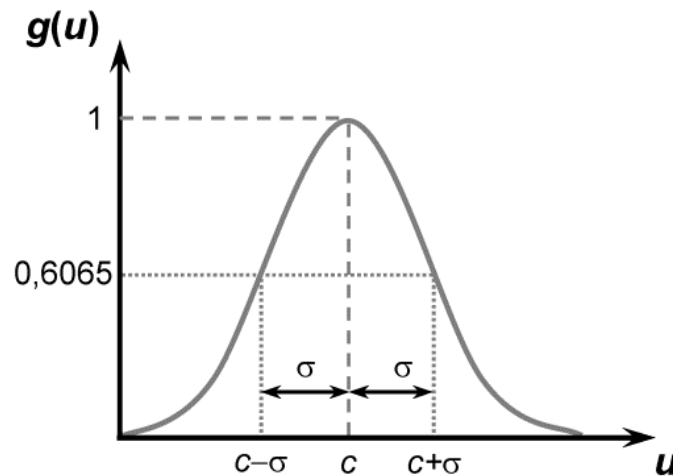
A função de ativação totalmente diferencial é aquela cuja derivada de primeira ordem é conhecida, as principais funções desse grupo são quatro: função gaussiana, função linear, função tangente hiperbólica e a função logística, que serão sucessivamente apresentadas a seguir.

a) *Função gaussiana*: nesta função de ativação a saída do neurônio produzirá resultados iguais para aos valores de potencial de ativação  $\{u\}$  que estejam posicionados na mesma distância que o seu centro, sendo que a curva deste é simétrica. Matematicamente representada a função é dada por:

$$g(u) = e^{\frac{-(u-c)^2}{2\sigma^2}}$$

$c$  é o parâmetro que define a função gaussiana e  $\sigma$  refere ao desvio padrão, ou seja, a quanto dispersada está a sua curva em relação ao ponto médio da função. O gráfico da função gaussiana é representado pela Figura 2.6.

Figura 6 – Função gaussiana.



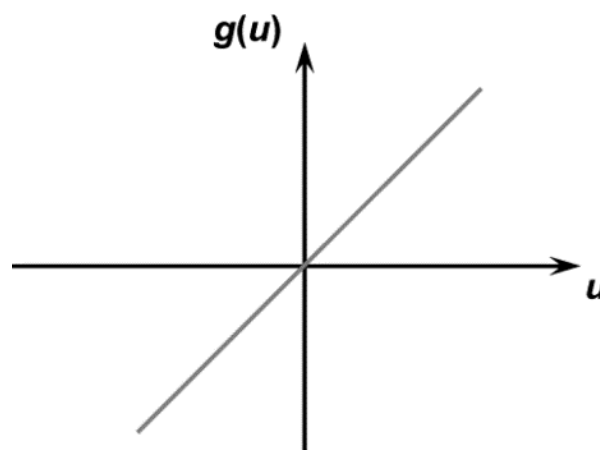
(fonte: SILVA et al, 2010, p.41).

b) Função linear: a função linear também chamada de função identidade gera resultados de saída idênticos aos valores do potencial de ativação  $\{u\}$ , a expressão que representa matematicamente essa abordagem é:

$$g(u) = u$$

O gráfico da função linear pode ser representado pela Figura 7:

Figura 7 – Função linear.



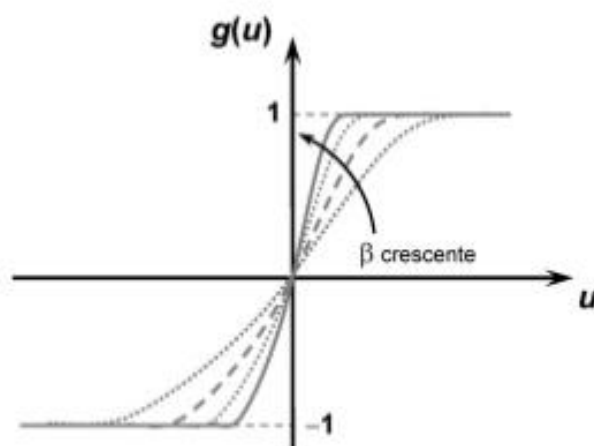
(fonte: SILVA et al, 2010, p.44).

- c) Função tangente hiperbólica: seu resultado da função de ativação sem estará em um intervalo real entre -1 e 1, é sua expressão é determinada por:

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}}$$

considerando  $\beta$  associado ao nível de inclinação da função tangente hiperbólica em relação ao seu ponto de inflexão. A Figura 8 representa o gráfico da função tangente hiperbólica para diversos valores impostos ao parâmetro  $\beta$ .

Figura 8 – Função tangente hiperbólica



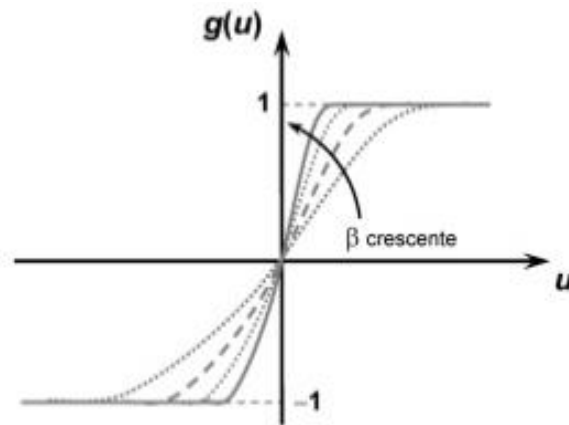
(fonte: SILVA et al, 2010, p.40)

- d) Função logística: sua saída sempre implicará em um valor real positivo, delimitado entre 0 e 1, diferente da função tangente hiperbólica. Assumindo sua expressão matemática como:

$$g(u) = \frac{1}{1 + e^{-ku}}$$

Tomando  $k$  como um constante real agregada ao nível de inclinação da função logística ao seu ponto de inflexão. A Figura 9 representa a projeção da função logística em relação a variação do parâmetro de inclinação  $k$  (constante).

Figura 9 – Função logística.



(fonte: SILVA et al, 2010, p.39).

A descrição do neurônio permitiu compreender melhor o que são as unidades básicas que compõem uma RNA e agora o próximo passo é falar sobre os parâmetros de desempenho associado ao funcionamento do neurônio artificial.

## **2. ARQUITETURA DAS REDES NEURAI ARTIFICIAIS E PROCESSOS DE TREINAMENTO**

Serão abordadas agora as distintas topologias e arquiteturas de Redes Neurais Artificiais existentes, bem como as suas formas de treinamento. Referindo-se à arquitetura de uma rede neural estamos nos aludindo sobre a acomodação dos neurônios, um em relação ao outro, seguindo as conexões sinápticas relatadas anteriormente. Por outro lado, a topologia da rede, refere-se aos diferentes acordos estruturais possíveis com diferentes quantidades de neurônios nas camadas de entrada, intermediária e de camadas de saída da rede.

O aprendizado de uma Rede Neural Artificial é concretizado por meio de processos iterativos de adequações aplicadas aos pesos sinápticos, denominado treinamento. O aprendizado só ocorre quando a rede neural atinge uma solução generalizada para um determinado problema. Em resumo, treinar uma rede é ajustar a sua matriz de pesos sinápticos de forma que o vetor de saída coincida com um valor desejado para cada vetor de entrada (MIRANDA et al., 2009).

O exercício de uma determinada arquitetura equivale da aplicação de uma série de passos ordenados com a intenção de ajustar os pesos e os limiares de seus neurônios. Com isso, este procedimento de ajuste, conhecido também como algoritmo de aprendizagem, pretende adequar a rede de modo que as respostas estejam próximas dos supostos valores almejados.

Sendo assim, a seguir serão apresentados alguns tipos de arquiteturas da Redes Neurais Artificiais e suas respectivas denominações literárias.

### **2.1. Principais Arquiteturas de Redes Neurais Artificiais**

Um aspecto importante a ser acatado é o modo como os neurônios artificiais são agrupados. No cérebro humano, este agrupamento é feito de maneira que as informações possam ser processadas de maneira dinâmica ou iterativa. As redes neurais biológicas são organizadas e elaboradas de maneira tridimensional microscópica. Existe uma forte restrição no número de camadas que uma rede neural pode ter, dessa forma, uma rede neural pode ter uma ou várias camadas. As redes que possuem um nó entre uma entrada e uma saída

da rede é a rede de uma única camada. Já as redes multicamadas possuem mais de uma camada entre as já existentes camadas de entrada e saída.

As redes artificiais multicamadas possuem as camadas escondidas (hidden), que também são chamadas de intermediárias ou ocultas. Esse número de camadas pode ser indeterminado, e estão situadas entre a camada de entrada e a camada de saída da rede neural (CARVALHO e LUDEMIR, 1998).

Segundo Silva et al (2010) uma rede neural artificial pode ser dividida em três partes, chamadas de camadas, e são nomeadas:

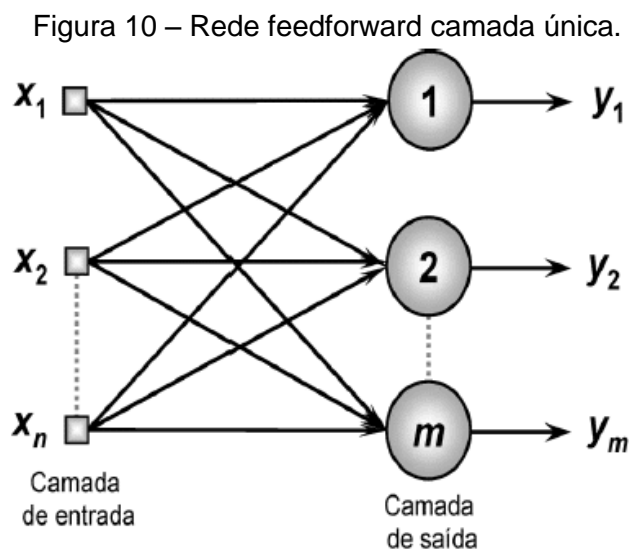
- 1) Camadas de entradas: É a camada responsável pelo recebimento de dados, sinais, características advindas externamente, sendo que tais entradas são geralmente normalizadas em relação as faixas de variações dinâmicas produzidas pelas funções de ativação. Esta normalização implica em uma melhor precisão numérica frente as operações matemáticas realizadas pela rede.
- 2) Camadas escondidas, intermediárias, ocultas ou invisíveis: são aquelas compostas de neurônios que possuem a responsabilidade de extrair as características associadas ao processamento interno da rede é realizado nessa camada.
- 3) Camada de saída: esta camada também Constituída de neurônios, sendo responsável pela produção e apresentação dos resultados finais da rede, os quais são advindos dos processamentos efetuados pelo neurônio das camadas anteriores.

Um exemplo desse tipo de arquitetura são as redes do tipo “feedforward”, em que o sinal é sempre propagado para a frente, da entrada para a saída. As redes feedforward, fundamentalmente estão organizadas em camadas e são largamente utilizadas atualmente. Podem ser divididas em: redes feedforward de camada simples, redes feedforward de camadas múltiplas, redes recorrentes e redes reticuladas.

### 2.1.1. Arquitetura *feedforward* de Camada Simples

Uma camada de entrada diretamente conecta a um ou mais neurônios que vão provocar a resposta de saída. O fluxo de informações segue sempre em direção à camada de saída, são empregadas em problemas de classificação de padrões e filtragem, a saber: Perceptron e Adaline, cujos algoritmos de aprendizados utilizados em seus processos de treinamento são respectivamente baseados na regra Hebb e na regra Delta. Logo as quantidades de saídas nas redes pertencentes a arquitetura sempre será equivalente ao número de neurônios.

A Figura 10 mostra uma rede feedforward de camadas simples composta de n entradas e m saídas.



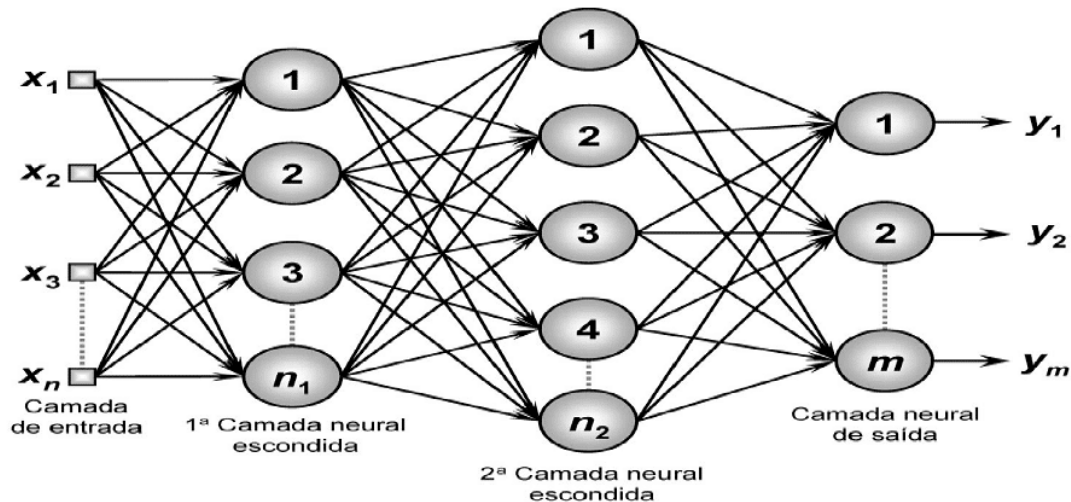
(fonte: SILVA et al, 2010, p.47).

### 2.1.2. Arquitetura *feedforward* de Camada Múltiplas

A rede feedforward de camadas múltiplas são constituídas por uma ou mais camadas escondidas de neurônios. São utilizadas em problemas de aproximação de funções, classificação de padrões, identificação de sistemas, otimização, robótica e controle de processos, a saber: Perceptron Multicamadas, Redes de Base Radial. Esta rede é composta de uma camada de entrada por n sinais, duas camadas neurais escondidas constituídas respectivamente de  $n_1$  e  $n_2$  neurônios e, por fim, uma camada neural de saída constituída de m neurônios representando os respectivos valores de saída.

A Figura 11 ilustra uma rede feedforward de camadas múltiplas de acordo com as informações relatadas anteriormente.

Figura 11 – Rede feedforward múltiplas camadas



(fonte: SILVA et al, 2010, p.48).

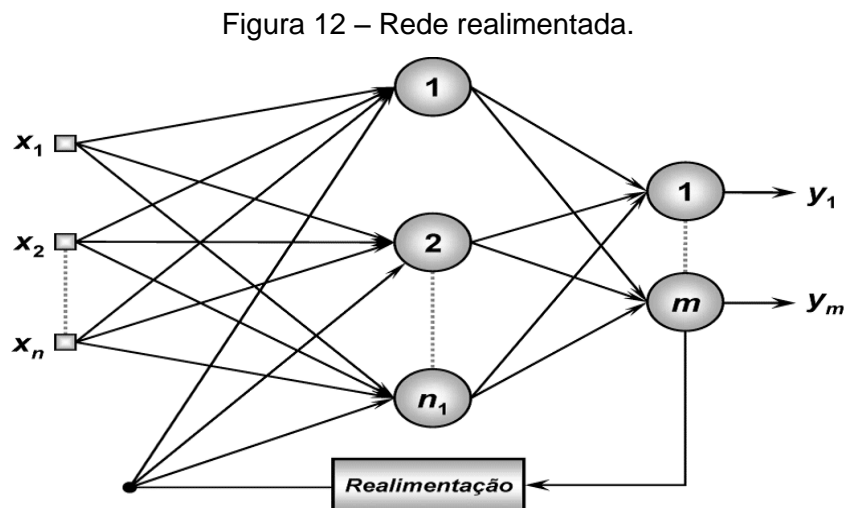
Segundo Silva et al (2010), sobre as redes neurais com arquitetura feedforward de camadas simples, é possível abstrair que a quantidade de neurônios que compõem a primeira camada escondida é normalmente diferente do número de sinais que compõem a camada de entrada da rede. De fato, o número de camadas escondidas e seus respectivos neurônios constituintes dependem do tipo e da complexidade do problema a ser mapeado pela rede, bem como da quantidade e qualidade dos dados disponíveis do problema.

### 2.1.3. Arquitetura Recorrente ou Realimentada

As Redes Neurais Artificiais recorrentes são redes mais gerais que possuem realimentação das saídas para as entradas, respectivamente, e suas saídas são determinadas pelas entradas atuais e pelas saídas anteriores. Também, a estrutura dessa rede não é necessariamente ordenada em camadas e se forem as redes podem apresentar interligações entre neurônios da mesma camada e inclusive camadas não consecutivas. Estas redes, pelo fato de possuírem realimentação, respondem a impulsos dinamicamente, isto é, depois de aplicar a nova entrada, calcula-se a saída e por fim, realimentada para modificar a entrada. Por esse motivo, denomina-se redes “com memória”.

Silva et al (2010) ressalta o conceito de arquitetura das redes neurais artificiais recorrente como, redes em que as saídas dos neurônios são realimentadas como sinais de entrada para outros neurônios. A característica da realimentação qualifica tais redes para o processamento dinâmico de informações, ou seja, estas podem ser utilizadas em sistemas variantes em relação ao tempo, como previsão de series temporais, otimização e identificação de sistemas, controle de processos, etc.

A Figura 12 mostra uma arquitetura recorrente de uma rede neural artificial, de acordo com os conceitos e características apresentadas anteriormente.



(fonte: SILVA et al, 2010, p.49).

Reafirmando o conceito de rede realimentada, Vellasco (2007) ressalta que as RNA's recorrentes, também chamadas de redes neurais artificiais com memória por possuírem realimentação, respondem a estímulos dinamicamente, ou seja, após aplicar uma nova entrada, a saída é calculada e então realimentada para modificar a entrada. Para as redes estáveis, este processo é repetido várias vezes, produzindo penas mudanças nas saídas, até ficarem constantes. Entretanto, as redes neurais recorrentes nem sempre são estáveis, mesmo com entradas constantes.

## 2.2. Processos de Treinamento

Para Silva et al (2010), o processo de treinamento de uma rede neural consiste na aplicação de passos ordenados que sejam necessários para a sincronização dos pesos sinápticos e limiares de seus neurônios, tendo-se como objetivo final a geração de soluções a serem produzidas pelas suas saídas, cujas respostas são representativas do sistema físico em que estas estão mapeando.

Um conceito muito importante na caracterização de redes neurais artificiais é o aprendizado. Este, é definido como uma modificação realizada na matriz de pesos  $w$  de uma rede, no sentido de otimizar um mapeamento dos padrões de entrada e de saída. Mesmo que em alguns casos simples fosse possível ajustar a matriz  $w$  manualmente, de modo a obter um mapeamento único, é desejável que este processo se dê de forma auto organizada, ou seja, a rede, através de um algoritmo apropriado, auto ajusta os seus pesos durante a progressão do aprendizado (SABBATINI, 1992).

No processo de aprendizado das redes neurais artificiais, são propostas diversas maneiras de treinamento, sendo estes subdivididos em dois paradigmas principais, são eles: o aprendizado não supervisionado e o supervisionado, que serão definidos e caracterizados a seguir.

### 2.2.1. Treinamento supervisionado

No treinamento supervisionado, treina-se as redes neurais artificiais antes de serem utilizadas. Este aprendizado consiste da apresentação dos sinais de entrada e de saída à rede. Grande parcela do tempo é direcionada a esse aprendizado. Quando a rede alcança um certo nível de performance o treinamento em questão é considerado completo. Quando não existe a necessidade de aprendizado, os pesos são praticamente inativos para a aplicação. O conjunto de aprendizados necessita ser suficientemente grande para conter as informações necessárias para que então a rede aprenda os moldes e as relações importantes. De acordo com o resultado, o sistema precisa aprender com todos os fatos em conjunto, provendo em seguida o ajuste mais adequado dos pesos para todo o conjunto.

Segundo Silva et al (2010), os pesos sinápticos e limiares são então continuamente ajustados mediante a aplicação de ações comparativas, executadas pelo próprio algoritmo de aprendizagem, que supervisionam a defasagem entre as respostas produzidas pela rede em relação aquelas desejadas, sendo esta diferença usada no procedimento de ajuste. A rede será considerada treinada quando tal defasagem estiver dentro de valores aceitáveis, levando em consideração os propósitos de generalização de soluções.

O algoritmo de aprendizado supervisionado mais utilizado é o Backpropagation (que será detalhado mais adiante). Portanto, o aprendizado supervisionado pressupõe um erro de saída.

Na aprendizagem supervisionada existem três componentes a serem destacados para o aprendizado.

1. Ambiente (estacionário): as características estatísticas não mudam com o tempo. Dessa forma, se o ambiente não é estacionário as estatísticas variam com o tempo. Ou seja, a rede neural é estimulada pelo ambiente através da apresentação de padrões na sua camada de entrada.
2. O professor (resposta desejada): o aprendizado supervisionado funciona como se existisse um professor que conhece a resposta almejada, isto é, sua estrutura interna é alterada em função destes padrões e da saída desejada.
3. Algoritmo de aprendizado: ajusta os pesos para fazer o mapeamento entrada-saída. Sua resposta futura ao estímulo do ambiente será diferente devido a alteração da estrutura interna.

Existem vários algoritmos de ajuste dos parâmetros da rede neural artificial, baseados neste conceito, a saber: Adaline, madaline, retro propagação do erro (“backpropagation”), entre outros.

### **2.2.2. Treinamento não supervisionado**

Contrário ao aprendizado supervisionado, o treinamento não supervisionado durante a aplicação de um algoritmo de aprendizado inexistem as saídas desejadas. Em virtude disso, a própria rede deve se auto estabelecer em relação as individualidades existentes em relação aos elementos

constituintes do conjunto total de amostras, identificando os conjuntos que apresentam semelhanças.

Isto é, uma vez que a rede esteja reajustada aos parâmetros estatísticos das informações de entrada, ela amplifica a habilidade de formar representações internas para codificar as características de entrada e elaborar novas classes. A regra de aprendizagem competitiva pode ser utilizada para este tipo de treinamento.

Estas redes não toleram influências externas para alteração dos seus pesos sinápticos. Pois existe um monitoramento interno na rede, ponderando as regularidades dos sinais de entrada, e com isso adaptando-se as necessidades da rede automaticamente.

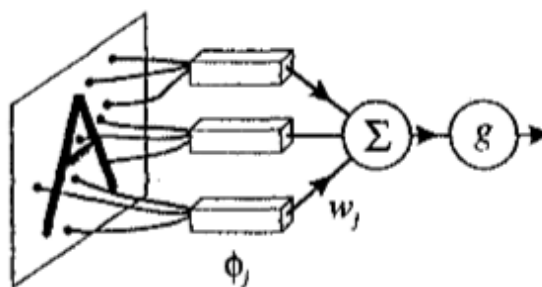
Quando a rede assimila uma quantidade excessiva de modelos, esta acaba memorizando as informações do treinamento, isto é, uma rede treinada em exagero perde a aptidão de generalização dos modelos de entrada-saída análogos. Com isso, consegue-se um ótimo resultado *in-sample* e um péssimo ajustamento *out-of-sample*. Além disso, quando se emprega mais neurônios ocultos que o necessário, faz com que subsídios indesejáveis da entrada fiquem contidos nos pesos sinápticos e a rede assimila ruídos.

### 3. REDES PERCEPTRON

Tem-se em latim a palavra para o verbo compreender denominada de “*percipio*”, e sua forma demasiada é “*perceptum*”, em virtude disso, pode se concluir que, a rede deve ser capaz de compreender o mundo exterior. Visto que, esse algoritmo de aprendizagem supervisionada considera um período de treinamento (com valores de entrada e saída) para definir se uma nova entrada pertence a alguma classe específica ou não. O Perceptron é o ancião das redes neurais. Ele foi elaborado em 1957 por Frank Rosenblatt. O Perceptron é o mais simples tipo de rede neural direta (Feedforward), conhecido como classificador linear. Isto quer dizer que os tipos de problemas solucionados por esta rede neural devem ser linearmente separáveis. Uma de suas aplicações consiste em identificar padrões geométricos.

O perceptron em sua origem consistia em uma simulação computacional da retina, a qual demonstrou como o sistema nervoso visual possui habilidade de reconhecer padrões, afirma Bishop (1995). A Figura 13 ilustra a maneira que os sinais elétricos obtidos de fotocélulas que mapeiam padrões geométricos são ponderados por resistores sintonizáveis, podendo ser ajustados durante o treinamento. Logo após um somador realizava as conciliações dos sinais em geral, com isso, o perceptron é capaz de reconhecer padrões geométricos.

Figura 13 – Perceptrons de Rosenblatt.

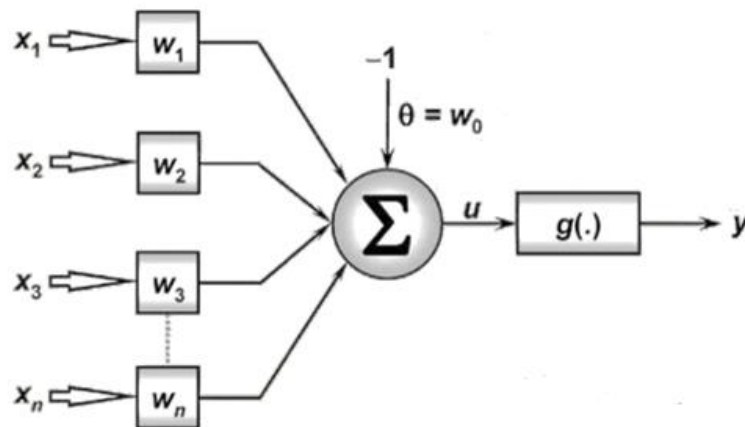


(Fonte: Bishop 1995, p. 98).

Entretanto, enquanto Rosenblatt desenvolvia o perceptron, Widrow e alguns colegas empenhavam-se ao Adaline uma linha semelhante à do perceptron. A nomenclatura ADaptive LINear Element provem de uma única unidade de processamento com um limiar não linear.

De acordo com a Figura 14, observa-se que o perceptron se refere a uma rede de uma única camada. Neste caso, na maioria das vezes, a rede é composta por um único neurônio e um polarizador. Uma rede neural que não possui camada oculta apenas consegue classificar padrões que sejam linearmente separáveis. A rede a seguir é constituída de  $n$  sinais de entrada, e somente uma saída, pois é formada de um neurônio.

Figura 14 – Rede perceptron.



(Fonte: SILVA et al, 2010, p.58.)

Na figura 4.1 o valor do limiar de ativação  $\{\theta\}$  foi assumido como sendo também um termo de ponderação  $\{W_0\}$ , tendo-se então o valor unitário negativo como respectiva entrada (SILVA et al, 2010). Estas redes utilizam a função de transferência do tipo *hard-limiter*, no qual a função de ativação do limiar tem resultado de saída 1 ou -1. O perceptron computa os valores de saída da seguinte maneira:

$$\begin{cases} 1 & \text{se } \sum x_i w_i \geq \theta \\ -1 & \text{se } \sum x_i w_i < \theta \end{cases}$$

Estas redes utilizam como configuração, o treinamento supervisionado. O perceptron modifica os seus respectivos pesos sinápticos, com a intenção de reduzir o erro.

Com bases nas informações expostas anteriormente tomamos como características básicas do perceptron:

- Função de ativação: degrau
- Topologia: uma única camada de neurônios
- Algoritmo de aprendizado: supervisionado.
- Valores entrada/saída: binários [-1, +1].

### 3.1. Princípio de Funcionamento de Perceptron

Para elaborar o aprendizado de um padrão, é imprescindível, de imediato, transformá-lo para a configuração de -1 e 1. Também pode ser encontrado arquivos de modelos binários, exemplo de resultados de sinais digitais. Os 0 do padrão binário necessitam ser colocado na forma de -1 para que não apresente erros no implemento das equações do algoritmo. O processamento torna-se desnecessário se o padrão for contínuo. A maneira em que são proporcionados os padrões é de suma relevância. Devem ser apresentados de maneira aleatória e/ou alternada, caso haja mais de um. Caso contrário, isso deixaria os pesos da rede sempre tendendo a apenas um dos padrões.

Cada uma das entradas  $\{X\}$ , as quais representam informações sobre o comportamento do processo a ser mapeado, será inicialmente ponderada pelos pesos sinápticos  $\{W_i\}$  a fim de quantificar a importância de cada uma frente aos objetivos funcionais atribuídos ao neurônio, cujo propósito será então mapear o comportamento entrada/saída do referido processo. Posteriormente, o valor resultante da composição de todas as entradas já devidamente ponderadas pelos seus respectivos pesos, adicionando ainda do limiar de ativação  $\{\theta\}$ , é repassado como argumento da função de ativação, cujo resultado de retorno será a saída  $\{y\}$  produzida pelo perceptrons (SILVA et al, 2010, p. 59).

Para realizar o reconhecimento de um padrão, é necessário que o algoritmo seja executado uma vez, sem que haja o ajuste dos pesos. Apenas isso basta para que alcançamos uma saída quantizada na qual indique a que padrão treinado pertence o padrão atual.

Em resumo, explica-se os aspectos característicos dos parâmetros relacionados com a dinâmica de funcionamento de um perceptrons:

- Entrada  $\{X_i\}$ : advindas externamente, reais ou binárias.
- Pesos sinápticos  $\{W_i\}$ : iniciados aleatoriamente, reais.
- Limiar de ativação  $\{\theta\}$ : iniciados aleatoriamente, reais.
- Saída  $\{y\}$ : binária.

-Função de ativação  $\{g(\cdot)\}$ : degrau ou degrau bipolar.

-Processo de treinamento supervisionado.

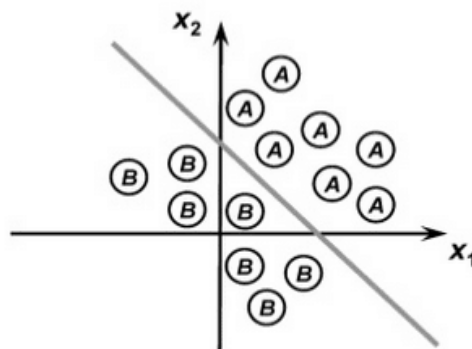
-Regra de aprendizado de Hebb.

A emprego do perceptron está limitado a quantidade de padrões envolvidos e sua complexidade. Este pode ser utilizado em reconhecimentos de sinais digitais, grandes redes e caracterização de apenas um padrão por neurônio.

### 3.2. Análise Matemática do Perceptron

Matematicamente o perceptron, tendo em vista a função de ativação sinal, pode ser considerado um caso de discriminador linear, ou seja, como ilustra a Figura 15, o Perceptron é um classificador linear e só irá lidar com problemas de classificação onde o conjunto de dados seja linearmente separável.

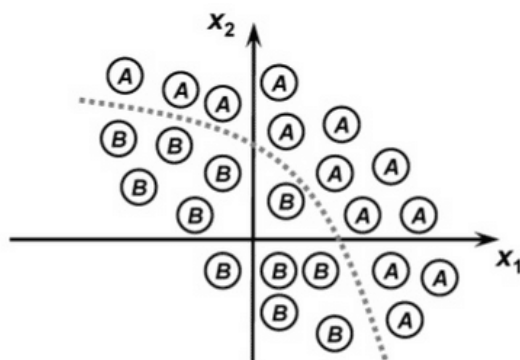
Figura 15 - Fronteira de separação do neurônio de entrada.



Fonte: SILVA et al, 2010, p.62.

Em particular para a figura 15, o perceptrons divide duas classes linearmente separáveis, matematicamente tem-se que quando a resposta for positiva (1) os padrões serão da classe B (localizados a cima da fronteira); no entanto, se a resposta for negativa (-1), os padrões serão da classe A (localizados abaixo da fronteira). Além disso, tem-se uma configuração (Figura 16) em que a reta será incapaz de separar essas classes, com isso, diz-se que ela é linearmente separável.

Figura 16 - Fronteira não linearmente separável.

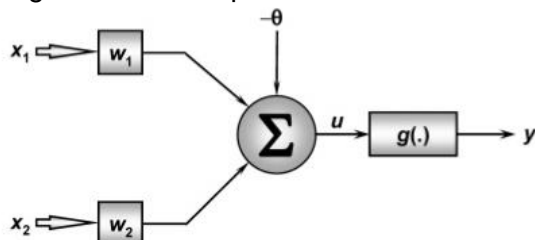


Fonte: Silva et al 2010, p.62.

Apesar dessa limitação ele não é um algoritmo fraco. Pois, por mais que seja difícil achar um conjunto de dados linearmente separável em aplicações reais, ele pode ser combinado com outros algoritmos para se aumentar a precisão.

Analisando o algoritmo (Figura 17), tem-se  $x_1$  e  $x_2$  sendo os inputs e  $w_1$  e  $w_2$  são os respectivos pesos sinápticos de cada input. E  $\Sigma$  sendo uma combinação linear desses inputs ponderada pelos pesos que gera o output, cuja representação é mostrada na Figura 17.

Figura 17 - Perceptron de duas entradas.



Fonte: SILVA et al, 2010, p.62).

Onde  $\theta$  é output desse. Depois disso ele é avaliado por uma função de perda tendo a função de ativação, dada por:

$$Y = \begin{cases} 1, & \text{se } \sum w_i * x_i - \theta \geq 0 \Leftrightarrow w_1 * x_1 + w_2 * x_2 - \theta \geq 0 \\ -1, & \text{se } \sum w_i * x_i - \theta < 0 \Leftrightarrow w_1 * x_1 + w_2 * x_2 - \theta < 0 \end{cases}$$

Pode-se analisar a semelhança entre as dimensões de um conjunto de dados pois dependendo da forma em que agrupamos as classes é aceitável que ele seja linearmente separável em uma combinação específica das dimensões dadas. Com algumas alterações também é possível aplicá-lo em problemas com mais de duas classes.

### 3.3. Princípio de Funcionamento do Adaline

O princípio de funcionamento do Adaline é semelhante ao do Perceptron. Nota-se que, assim como a rede perceptron, a aquisição da saída do Adaline é a mesma e tem a mesma sequência definida para o modelo Perceptron, representado por:

$$\begin{cases} u = \sum_{i=1}^n w_i * x_i - \theta \\ y = g(u) \end{cases}$$

Partindo do pressuposto da presença de um bloco associador junto à estrutura Adaline, o sinal de erro obtido pelo bloco é:

$$\text{Erro} = d - u$$

Cujo mesmo modelo tem seus respectivos pesos sinápticos  $\{\theta, W_1, X_2, \dots, W_n\}$  ajustados de acordo com o erro potencial de ativação elaborado pela rede com seu valor previamente desejado.

Para que obtenha completa identificação, as classes são linearmente separáveis, já que análise matemática executada para demonstração das condições de convergência da rede Perceptron também pode ser aplicada ao modelo Adaline.

### 3.4. Processo de Treinamento do Adaline

A rede Adaline possui também arquitetura feedforward de camada única, pois o fluxo de informações é realizado sempre adiante, isto é, partindo-se das entradas em direção a saída da rede. Observa-se ainda que existe qualquer tipo de realimentação de valores produzidos pelo seu único neurônio (SILVA, et al 2010, p.74).

O intuito da regra Delta é de se alcançar o ponto de mínimo da diferença entre a saída desejada  $\{d\}$  e a resposta do combinador linear  $\{u\}$  usando um exemplo de conjunto de cada vez, e interações locais, a partir daí ajustar os pesos sinápticos e as bias do neurônio. Isto é, essa metodologia faz referência a minimização do erro quadrático entre  $u$  e  $d$  com o objetivo de ajustar o vetor de pesos sinápticos  $w = \{\theta, w_1, w_2, \dots, w_n\}$  da rede.

O intuito deste algoritmo é obter  $w^*$  ótimo para que o erro quadrático definido por  $(E(w^*))$  sobre o conjunto por completo de amostras seja o mínimo. Esta configuração pode ser expressada por:

$$E(w^*) \leq E(w), \forall w \in R^{n+1}$$

Se tratando de uma função erro quadrática, as  $p$  amostras de treinamento é dada por:

$$E(w) = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - u)$$

A partir daí obtém-se:

$$E(w) = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - (\sum_{i=1}^n w_i \cdot x_i^{(k)} - \theta))^2$$

$$E(w) = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - (w^t \cdot x^{(k)} - \theta))^2$$

Utiliza-se o operador gradiente em analogia ao vetor de pesos  $w$ , com o intuito de encontrar o valor ótimo para o erro quadrático médio. Dessa forma:

$$\nabla E(w) = \frac{\partial E(w)}{\partial w}$$

Com isso:

$$\nabla E(w) = \sum_{k=1}^p (d^{(k)} - (w^t \cdot x^{(k)} - \theta)) \cdot (-x^{(k)})$$

Então:

$$\nabla E(w) = - \sum_{k=1}^p (d^{(k)} - u) \cdot x^{(k)}$$

Pois:

$$u = \sum_{i=1}^n w^i \cdot x^{(k)} - \theta$$

O processo de aprendizado do vetor peso necessita ser feito em direção a um determinado ponto de mínimo, neste ponto caminha ao sentido oposto ao gradiente, pois o intuito da otimização seria minimizar o erro quadrático médio. Este método também é conhecido como método do valor gradiente descendente. Considerando  $\mu$  como parâmetro da taxa de aprendizado. Sendo assim  $\Delta w$  (variação) no vetor Adaline é dada por:

$$\Delta w = -\mu \cdot \nabla E(-w)$$

Substituindo o resultado da penúltima expressão na última expressão, tem-se:

$$\Delta w = \mu \cdot \sum_{k=1}^p (d^{(k)} - u) \cdot x^{(k)}$$

Com isso:

$$w^{atual} = w^{anterior} + \mu \cdot \sum_{k=1}^n (d^{(k)} - u) \cdot x^{(k)}$$

Simplificando o resultado, atualizamos  $w$  depois de cada  $k$ -ésimo padrão de aprendizado, então:

$$w^{atual} = w^{anterior} + \mu \cdot (d^{(k)} - u) \cdot x^{(k)}, \text{ onde } k = 1, \dots, p$$

### 3.5. Comparação entre o Processo de Treinamento de Adaline e Perceptron

O Adaline é similar ao Perceptron, com contestação apenas pelo seu algoritmo de treinamento. Enquanto o Perceptron ajusta os pesos somente quando um padrão é rotulado incorretamente, o Adaline, por outro lado, utiliza a regra Delta para minimizar o erro médio após cada padrão ser apresentado, ajustando os pesos proporcionalmente ao erro.

Ao contrário do Adaline o procedimento de treinamento do Perceptron é feito pela regra de Hebb, a qual se analisa a resposta produzida após a apresentação particular de cada modelo de treinamento. Sendo assim, alguns hiperplanos dispostos dentro da faixa de separabilidade entre as classes são consideradas respostas adequadas, pelo perceptrons, ao propósito de classificação de padrões.

Nota-se que o processo de treinamento do Adaline é efetivado por meio da regra Delta, que tem por objetivo tornar mínimo a diferença entre a saída desejada e a resposta do combinador linear, tendo em vista todas as amostras de treinamento disponíveis. Por isso, independentemente dos valores iniciais impostos ao seu vetor de pesos, o hiperplano de separabilidade adquirido após a convergência da rede será o mesmo.

### 3.6. Redes Perceptron Multicamadas

Redes Perceptron de Múltiplas Camadas (PMC), também conhecidas como redes MLP (Multiple Layer Perceptron), são caracterizadas pela presença de pelo menos uma camada intermediária de neurônios. Essas camadas

intermediárias são aquelas localizadas entre a camada de entrada e a respectiva camada neural de saída, conseqüentemente, as redes Perceptron Múltiplas Camadas possuem no mínimo duas camadas de neurônios, os quais são distribuídos entre as camadas intermediárias e a camada de saída.

A rede PMC é uma das mais versáteis quanto às suas aplicações, podendo ser utilizadas em alguns tipos de problemas, a saber: Aproximação universal de funções, classificação de padrões, identificação e controle de processos, previsão de séries temporais e também otimização de sistemas.

A respeito da arquitetura do Perceptron de Múltiplas Camadas, este pertence à arquitetura feedforward de camadas múltiplas. Seu treinamento é executado de forma supervisionada. No fluxo de informações, primeiramente, inicia-se na camada de entrada, depois, percorre, em seguida, as camadas intermediárias, e por fim, finaliza-se na camada neural de saída. No PMC convencional inexistente qualquer tipo de realimentação de valores produzidos pela camada neural de saída ou pelas próprias camadas neurais intermediárias.

### **3.6.1. Princípio de Funcionamento de Perceptron Multicamadas**

As entradas do Perceptron de Múltiplas Camadas, representando os sinais advindos de determinada aplicação, serão propagadas, camada-a-camada, em direção à camada neural de saída. Estas partidas dos neurônios da primeira camada neural de saída serão as próprias entradas daqueles neurônios pertencentes à segunda camada neural escondida. Os saimentos dos neurônios da segunda camada neural escondida serão as respectivas entradas dos neurônios pertencentes à sua camada neural de saída.

Diferentemente do modelo Perceptron e Adaline, além da presença de camadas escondidas, a camada neural de saída do PMC pode ser composta por diversos neurônios. Cada um destes neurônios de saída representaria uma das saídas do processo a ser mapeado. As camadas intermediárias, por sua vez, extraem a maioria das informações referentes ao seu comportamento e as codificam por meio dos pesos sinápticos e limiares de seus neurônios.

O plano de um PMC depende dos seguintes aspectos, a saber: Classe de problema a ser tratado, disposição espacial das amostras de treinamento, valores iniciais atribuídos tanto aos parâmetros de treinamento como para as matrizes de pesos, nível de ruído presente nas amostras de treinamento. O processo de treinamento do PMC é feito mediante o algoritmo backpropagation. Este processo é realizado por meio das aplicações sucessivas de duas fases bem específicas que serão mostradas posteriormente.

### 3.6.2. Processo de Treinamento do Perceptron Multicamadas

O Perceptron múltiplas camadas em seu processo de treinamento aplica-se um algoritmo Backpropagation ou algoritmo da retropropagação. Este treinamento constitui-se de duas fases bem específicas. A primeira fase dita forward, tem o intuito de obter resultado da rede. Os sinais de entrada  $\{x_1, x_2, \dots, x_n\}$ , nesta fase, de uma amostra do um conjunto de treinamentos são inseridos nos inícios da rede e também são propagados camada por camada até obter a produção das saídas (respostas) que serão comparadas com as saídas desejadas. SILVA et al (2010) afirma que este processo é repetido, sequencialmente para todas as amostras de treinamento, até que a saída produzida pelo perceptron seja similar à saída desejada de cada amostra. Em termos matemáticos, as regras de ajustes dos pesos sinápticos  $\{w_i\}$  e do limiar  $\{\theta\}$  do neurônio podem ser expressas, respectivamente, pelas seguintes expressões:

$$\begin{cases} w_i^{atual} = w_j^{anterior} + \mu \cdot (d^{(k)} - y) \cdot x^{(k)} \\ \theta_i^{atual} = \theta_j^{anterior} + \mu \cdot (d^{(k)} - y) \cdot x^{(k)} \end{cases}$$

Em uma rede Perceptron múltipla camada que seja composta de  $n$  neurônios em sua camada de saída, os  $n$  erros são então calculados, e ajustados posteriormente os pesos sinápticos e limiares de todos os seus neurônios. Sequencialmente utiliza-se a segunda fase do método backpropagation chamada de propagação reversa ou backward. A diferença desta para a outra fase é que nessa são alterados os pesos sinápticos e limiares de todos os neurônios da rede.

Estas duas fases (forward e backward) alteram automaticamente os pesos sinápticos e limiares dos neurônios em questão, em cada interação,

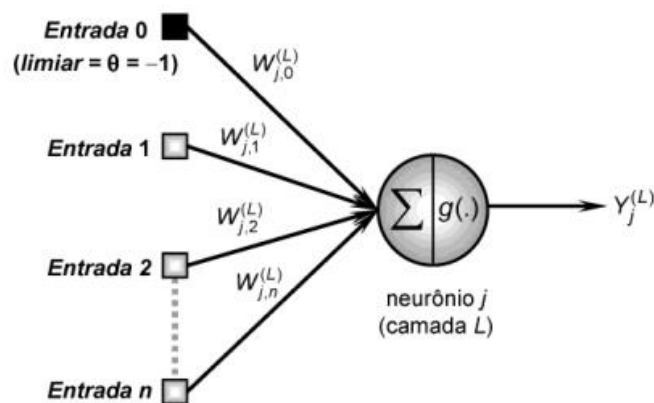
implicando aos poucos a diminuição da soma dos erros das respostas da rede com a saída almejada.

### 3.6.3. Derivação do Algoritmo backpropagation

O algoritmo apresentado neste momento representa uma maneira competente de prática computacional da regra da cadeia, sobretudo quando a instrumento empregado tem aptidão de processamento matricial, como os pacotes Matlab (que será utilizado nesse estudo).

Para simplificar o desenvolvimento do algoritmo de retropropagação, utilizar-se a notação abreviada para uma arquitetura genérica. Como mostra a Figura 18, cada neurônio  $\{j\}$  pertencente a uma das classes  $\{L\}$  pode ser configurado, onde  $g(\cdot)$  representa uma função de ativação que deve ser contínua e diferenciável em todo o seu domínio. Definindo variáveis e parâmetros (vetores de entrada) tem-se:

Figura 18 - Configuração do neurônio com derivação do algoritmo de backpropagation.



(Fonte: SILVA et al, 2010, p.96.)

$l_j^{(1)}$  são vetores cujos elementos denotam a entrada ponderada em relação ao  $j$ -ésimo neurônio da camada  $l$ , os quais são definidos por:

- $I_j^{(1)} = \sum_{i=0}^n W_{ji}^{(1)} \cdot x_i \Leftrightarrow I_j^{(1)} = w_{j,0}^{(1)} \cdot x_0 + w_{j,1}^{(1)} \cdot x_1 + \dots + w_{j,n}^{(1)} \cdot x_n$
- $I_j^{(2)} = \sum_{i=0}^{n_1} W_{ji}^{(2)} \cdot y_i^{(1)} \Leftrightarrow I_j^{(2)} = w_{j,0}^{(2)} \cdot y_0^{(1)} + w_{j,1}^{(2)} \cdot y_1^{(1)} + \dots + w_{j,n_1}^{(2)} \cdot y_{n_1}^{(1)}$
- $I_j^{(3)} = \sum_{i=0}^{n_2} W_{ji}^{(3)} \cdot y_i^{(2)} \Leftrightarrow I_j^{(3)} = w_{j,0}^{(3)} \cdot y_0^{(2)} + w_{j,1}^{(3)} \cdot y_1^{(2)} + \dots + w_{j,n_2}^{(3)} \cdot y_{n_2}^{(2)}$

A terminologia para tais parâmetros constituintes são:  $w_{ji}^{(L)}$  significa matrizes de pesos cujos elementos significam o valor do peso sináptico atrelando o j-ésimo neurônio da camada L ao i-ésimo neurônio da camada L-1.  $W_{ji}^{(3)}$  denota o peso sináptico acoplado o j-ésimo neurônio da camada de saída ao i-ésimo neurônio da camada 2;  $W_{ji}^{(2)}$  representa o peso sináptico conectando o j-ésimo neurônio da camada escondida 2 ao i-ésimo neurônio da camada 1;  $W_{ji}^{(1)}$  refere-se ao peso sináptico conectando o j-ésimo neurônio da camada escondida 1 ao i-ésimo neurônio da camada de entrada.

A saída do j-ésimo neurônio em relação a camada L, é representado por  $y_j^{(L)}$  que são vetores definidos por:

$$Y_j^{(1)} = g(l_j^{(1)})$$

$$Y_j^{(2)} = g(l_j^{(2)})$$

$$Y_j^{(3)} = g(l_j^{(3)})$$

Onde,  $Y_j(k)$  é a resposta calculada para o neurônio j. também  $d_j(k)$  é a resposta desejada para o neurônio j. O processo de treinamento do Perceptron múltiplas camadas é em função do sinal de erro, representado adiante, onde  $Y_i(k)$  refere-se a resposta calculada para o neurônio j e  $d_j(k)$  representa a resposta desejada para o neurônio j.

$$e_j(k) = d_j(k) - Y_i(k)$$

A função utilizada para medir o desempenho local conexo aos resultados produzidos pelos neurônios de saída a referida amostra é a denominada função erro quadrático. Onde  $y_j^{(3)}(k)$  remete ao valor produzido pelo j-ésimo neurônio de saída da rede tendo em vista a k-ésima amostra de treinamento, também  $d_j(k)$  é o valor almejado. Isto é:

$$E(k) = \frac{1}{2} \sum_{j=1}^{n_2} (d_j(k) - y_j^{(3)}(k))^2$$

Sendo assim, o erro quadrático médio de um conjunto composto com  $p$  amostras é representado por:

$$E_M = \frac{1}{p} \sum_{k=1}^p E(k)$$

A metodologia aplicada para o ajuste dos pesos sinápticos e bias é fundamentado no gradiente da função erro quadrático mencionado anteriormente.

#### **3.6.4. Implementação do Algoritmo Backpropagation**

O modelo de retropropagação ou Backpropagation, é utilizado para reconhecimento de caracteres, series temporais, composição musical, aprendizagem de robôs, reconhecimento de sinais em geral, etc.

A elaboração do modelo de abertura tem os mesmos princípios do modelo perceptron/Adaline, isto é, não há muito pré-processamento, dependendo da aplicação desejada. Esta aplicação, além disso, define a quantidade de neurônios existentes no conjunto intermediário. Frequentemente emprega-se um neurônio para cada camada de padrão. Além do mais é comum o emprego de somente uma camada intermediária para grande parte das aplicações. A retropropagação tem a mesma série de apresentação dos padrões perceptron/Adaline, ou seja, aleatória ou alternada. Sequencialmente definidos os padrões e a topologia, faz-se imprescindíveis algumas iterações de aprendizado para que então possa decidir os parâmetros. Assim como no modelo perceptron/Adaline,  $\alpha$  e  $\theta$  serem funções variáveis, estes podem ser definidos pelo projetista. O desenvolvimento do seguinte modelo mencionado, está no momento  $\mu$ , coeficiente no qual, permite a aceleração do processo de convergência e seu valor fica no intervalo  $[0;1]$ .

O treino se consolidará quando finalmente as saídas esperadas estiverem numa margem segura de proximidade à saída da função sigmoid. A aproximação existe em meio à dificuldade de se chegar a um resultado esperado da saída e, então, estima-se um valor de erro para que se obtenha a convergência. Grande parte das aplicações são imprescindíveis várias iterações até a convergência.

Uma das maiores vantagens deste modelo (Backpropagation) é sua aptidão de compreender as mais variadas classificações de padrões, podendo ser utilizado numa grande gama de aplicações. De outro modo, as desvantagens também são muitas. O tempo de treinamento é bastante longo para a grande parte das aplicações de uso prático, como por exemplo o reconhecimento de caracteres. Existe também alguns tipos de padrões, como aqueles envolvidos em séries temporais, que exige uma enorme necessidade de pré-processamento, dada a complexidade da análise e classificação dos padrões em questão.

Na retropropagação o reconhecimento é somente uma iteração do algoritmo para verificação da saída. Tendo o padrão que se quer reconhecer, faz-se uma iteração do algoritmo, ajustando os seus respectivos pesos sinápticos, e fazendo a comparação da saída do sigmoid com as saídas desejadas aprendidas. Sobre o resultado desta comparação nota-se a qual classe pertence o padrão.

## 4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

### 4.1. Materiais e Métodos

Castanhal/PA pertence à Mesorregião Metropolitana de Belém. Com uma área de aproximadamente 1.028,889 km<sup>2</sup>, localiza-se no nordeste paraense, com Latitude 01°18'03" S Longitude 047°55'51" W, altitude de 41 metros e distante 68 quilômetros da capital estadual, Belém, e 2.078 km da capital federal, Brasília. Com uma população de 175.000 habitantes (IBGE/2017). A Figura 19 mostra uma vista aérea do centro urbano da cidade de Castanhal.

A cidade de Castanhal/PA tem um período chamado de inverno, que é caracterizado como mais chuvoso, que vai dezembro a maio e um período menos chuvoso chamado de verão, que vai de junho a novembro. A área urbana da cidade, de acordo com sua posição geográfica (latitude e longitude), apresenta um aspecto importante do ponto de vista climático que é a baixa latitude. A base de dados foi obtida junto ao "SEOMA/PA" ([seoma.pa@inmet.gov.br](mailto:seoma.pa@inmet.gov.br)) e é constituída de 1462 pontos, que correspondem a dados diários da temperatura máxima (°C) medida na cidade de Castanhal/PA.

Figura 19 – Vista aérea da cidade de Castanhal/PA.



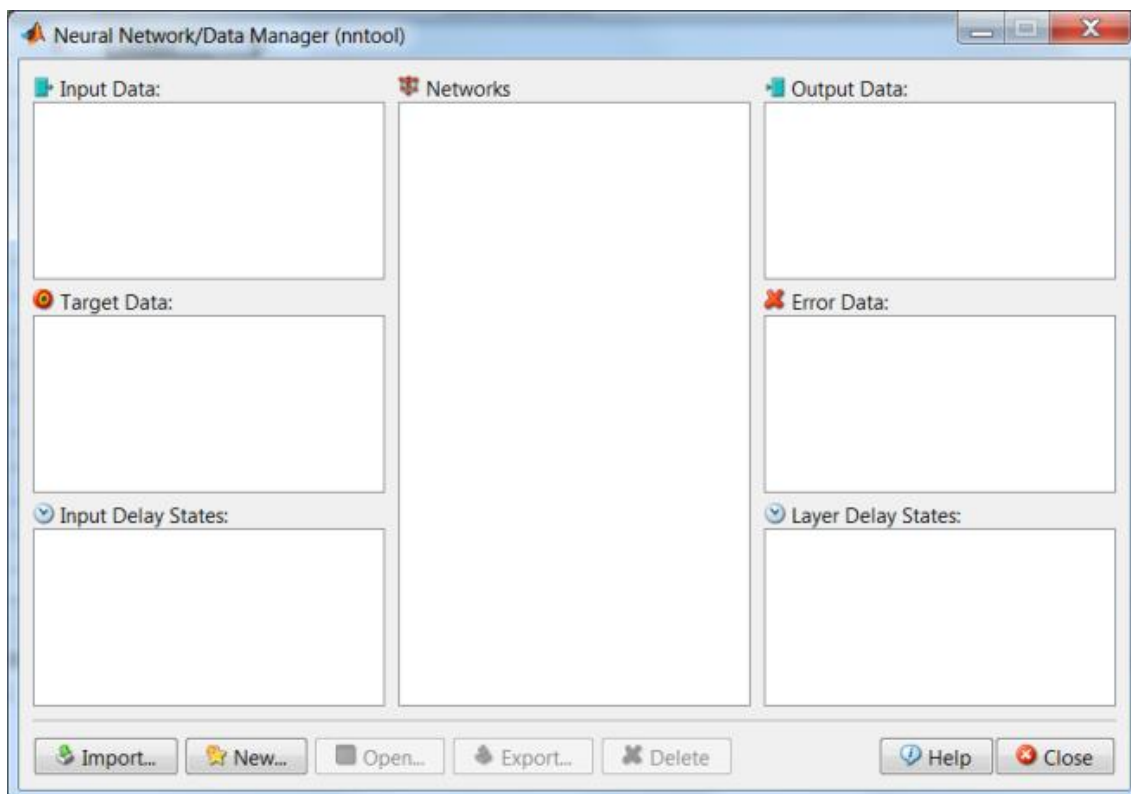
Fonte:

[https://www.google.com.br/search?q=altitude+castanhal+par%C3%A1&rlz=1C1NHXL\\_ptBRBR735BR735&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiljLbpnYzWAhXFIFQKHx6VAXMQ\\_AUICigB&biw=1366&bih=662#imgdii=Ciy4y3B03RkeiM:&imgsrc=DvlbBaDJ2koufM::,](https://www.google.com.br/search?q=altitude+castanhal+par%C3%A1&rlz=1C1NHXL_ptBRBR735BR735&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiljLbpnYzWAhXFIFQKHx6VAXMQ_AUICigB&biw=1366&bih=662#imgdii=Ciy4y3B03RkeiM:&imgsrc=DvlbBaDJ2koufM::,) maio de 2018).

## 4.2. Treinamento da RNA

Partindo do estudo de redes neurais artificiais, as mesmas foram empregadas com o fim de modelar a temperatura máxima ( $^{\circ}\text{C}$ ) na cidade de Castanhal/PA. Uma das características marcantes do uso das RNAs é, exatamente, a não necessidade da existência de um modelo matemático pré-existente. Para que seja eficiente, a RNA deve ser treinada de maneira adequada com vetores de entrada reais, de forma que a mesma adquira experiência. Quanto melhor a qualidade desses vetores e maior o seu treinamento, com cuidado para não saturar, a RNA tende a ficar cada vez mais funcional, até que comece a ter capacidade de tomar decisões com altas taxas de acerto, baseada nos padrões reconhecidos em seu treinamento. Neste trabalho, o treinamento da RNA foi realizado em software Matlab, utilizando-se a interface gráfica NNTool (Neural Network Toolbox), a qual permite importar, criar, utilizar e exportar dados de redes neurais. O acesso ao NNTool é realizado digitando NNTool na janela de comandos do MATLAB. Uma janela denominada Network/ Data Manager será aberta, conforme a Figura 20.

Figura 20 – Janela de comandos NNTool.



Fonte: A autora.

Os padrões (dados de entrada e de saída) e a arquitetura da rede são definidos de acordo com a Figura 20. É possível visualizar a rede neural pressionando o botão "View". Após a criação da rede, passa-se para a fase de treinamento. Nesta etapa, determina-se cada parâmetro de treinamento, tais como dados de validação e teste, número de épocas, valores dos pesos ou bias, etc. Para iniciar o processo de treinamento pressiona-se o botão Train Network.

#### 4.3 Tipo de rede.

Foi utilizada uma rede de múltiplas camadas Multilayer Perceptron (MLP) com algoritmo de treinamento Backpropagation, já implementado no Matlab. O algoritmo usado no ajuste dos pesos da rede MLP é um fator relevante para o seu desempenho, pois este estabelece um conjunto de regras bem definidas para a solução de um problema de aprendizagem.

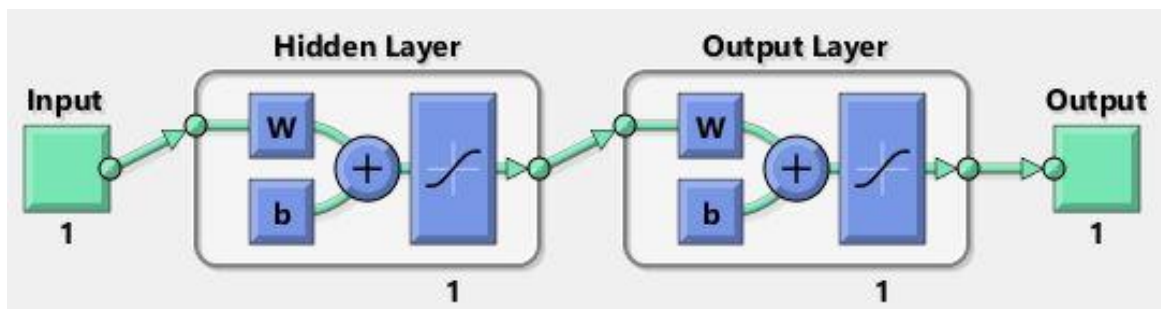
#### 4.4 Estrutura da rede.

Após uma análise de arquitetura da MLP, foram utilizados 1, 5, 10, 20, 30 e 60 neurônios na camada escondida. São fornecidos dados de entrada e de saída, que são processados pela rede gerando uma única saída preditiva. Foram feitos alguns treinamentos para que a rede pudesse ajustar seus pesos e apresentasse um bom desempenho com alta taxa de acerto.

Para a criação da rede neural foi utilizado o soft Matlab versão R2015a. Foram coletados os valores dos erros mínimo e máximo, os gráficos contendo a propagação do erro cometido e outro gráfico que exhibe os resultados da simulação versus os dados de campo.

As redes simuladas possuem uma camada oculta que possuem 1, 5, 10, 20, 30 e 60 neurônios, sendo 100 o número de épocas para cada simulação. Portanto as simulações foram realizadas para 6 (seis) topologias diferentes. É importante frisar que para cada topologia foram realizadas 5 simulações; perfazendo um total de 30 simulações. Onde foram coletados os erros médio e mínimo para cada topologia que apresentou melhor desempenho. A Figura 21 mostra a topologia de uma das redes simuladas com 1 neurônio na camada oculta.

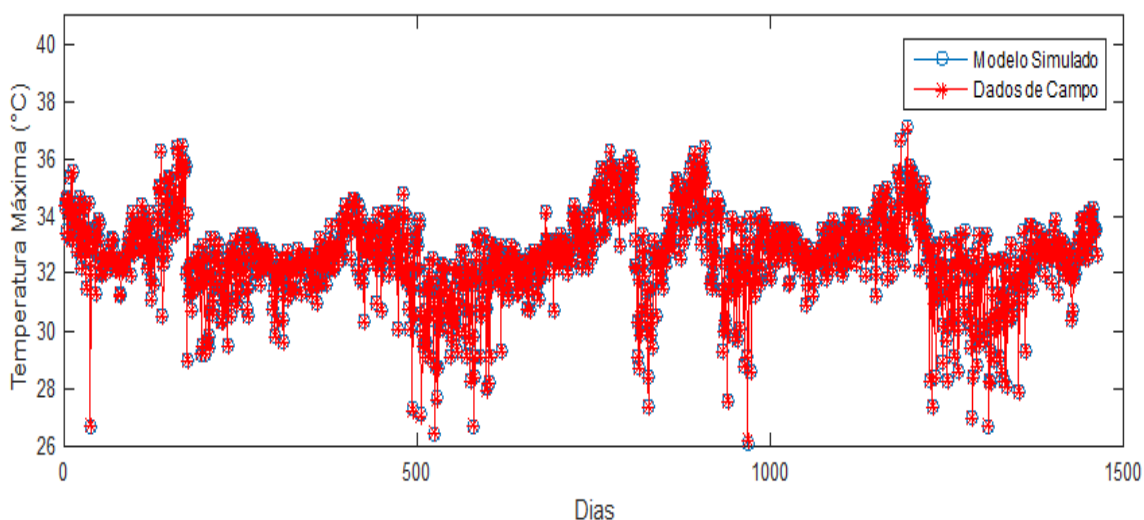
Figura 21 – Topologia de uma das redes simuladas com 1 neurônio na camada oculta.



Fonte: A autora

A Figura 22 apresenta os resultados obtidos pela rede simulada e os valores de campo com 5 neurônios na camada escondida, enquanto que a Figura 23 apresenta a propagação do erro para esta topologia. As Figuras 24 e 25 apresentam os resultados obtidos pela a rede simulada com 30 neurônios na camada escondida, enquanto que as Figuras 26 e 27 apresentam os resultados obtidos pela a rede simulada com 60 neurônios na camada escondida. A Tabela 1 apresenta o melhor desempenho de cada topologia de rede entre as seis simulações. Observe as redes simuladas com 5, 30 e 60 neurônios na camada escondida apresentaram os melhores erros mínimos (em negrito).

Figura 22 – Modelo simulado e dados de campo para rede simulada com 5 neurônios na camada oculta.



Fonte: A autora

Figura 23 – Propagação do erro para rede simulada com 5 neurônios na camada oculta.

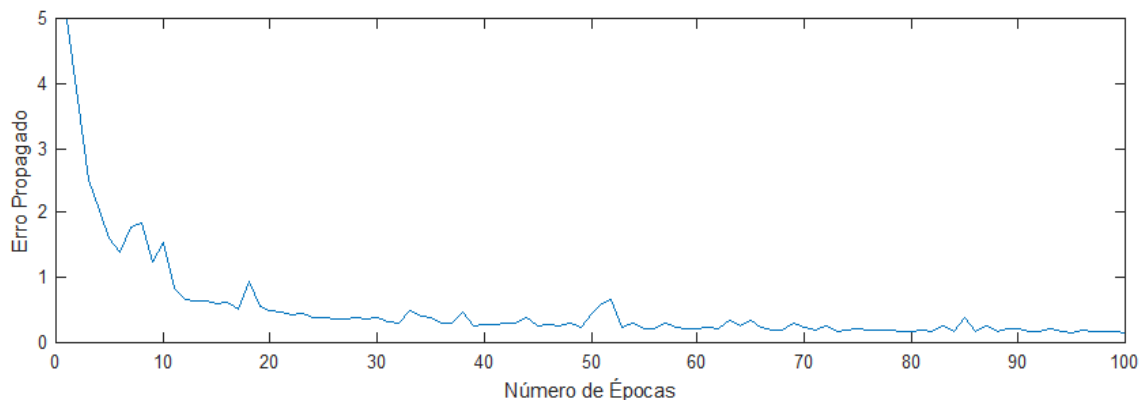


Figura 24 – Modelo simulado e dados de campo para rede simulada com 30 neurônios na camada oculta.

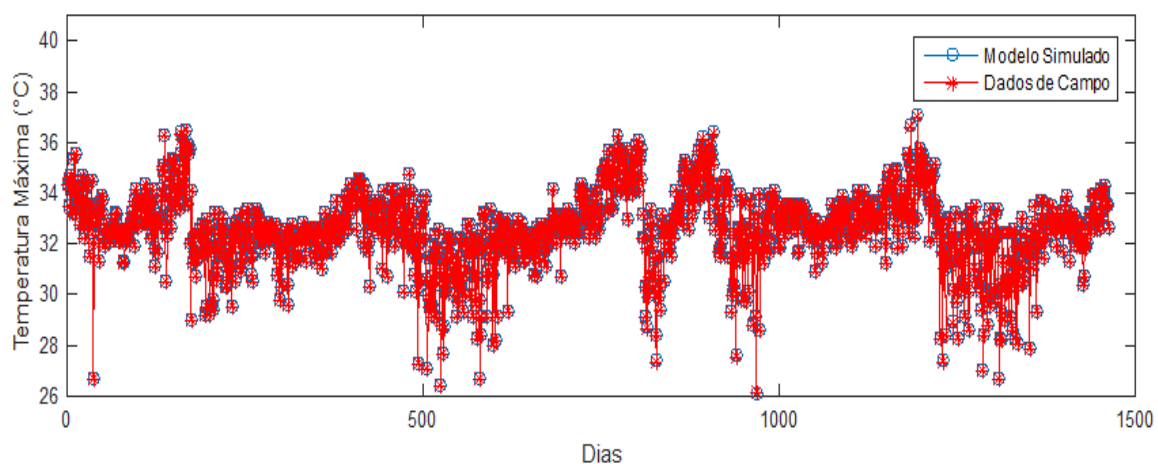


Figura 25 – Propagação do erro para rede simulada com 30 neurônios na camada oculta.

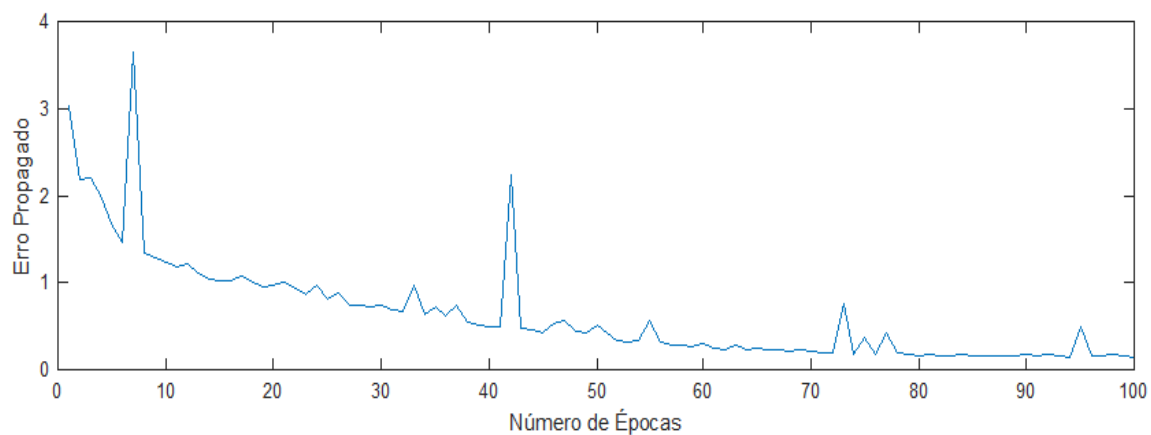
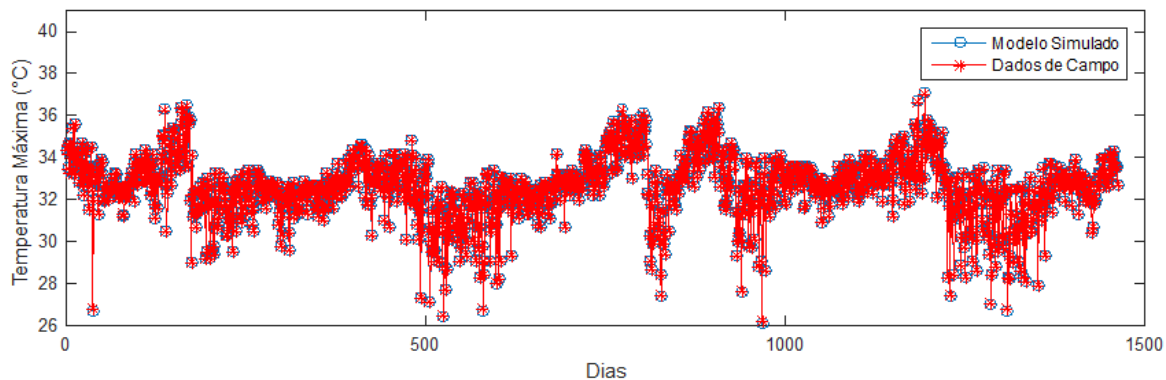
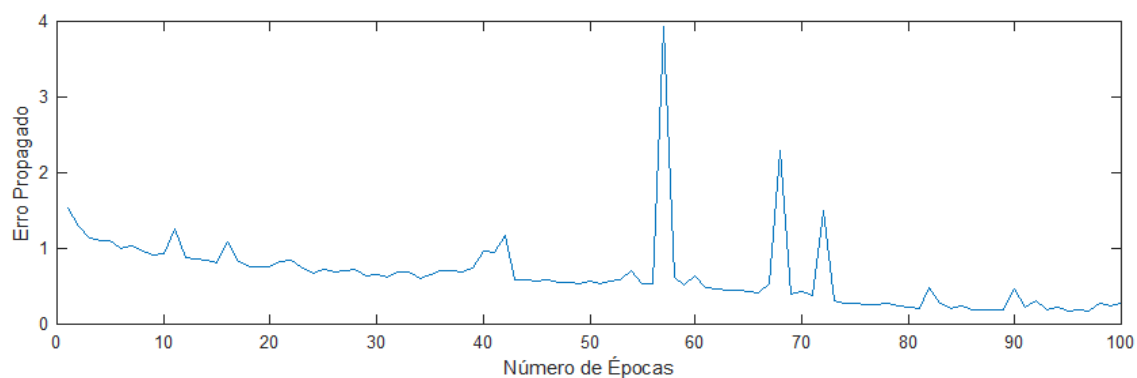


Figura 26 – Modelo simulado e dados de campo para rede simulada com 60 neurônios na camada oculta.



Fonte: A autora

Figura 27 – Propagação do erro para rede simulada com 60 neurônios na camada oculta.



Fonte: A autora

Tabela 1 – Erro mínimo e médio cada topologia de rede.

	Número de Neurônios na Camada Escondida					
	1	5	10	20	30	60
<b>Erro Médio</b>	0,271	0,5147	0,675	0,534	0,642	0,653
<b>Erro Mínimo</b>	0,492	<b>0,154</b>	0,288	0,192	<b>0,146</b>	<b>0,176</b>

## 5. CONCLUSÃO

Neste trabalho foi estudada a variação da temperatura máxima (°C) na cidade de Castanhal/PA através do uso de redes neurais artificiais. Foram criadas diferentes configurações de RNAs com o objetivo de tentar simular através do algoritmo backpropagation esta variável. Os resultados obtidos permitiram concluir que a temperatura máxima (°C) pode ser prevista com erros médios em torno de 0,6 %. Apenas uma camada oculta foi suficiente para a previsão da variável estudada, a qual obteve resultados mais precisos quando se utilizou uma topologia com 30 neurônios na camada oculta. Isto mostra que o uso de RNA para a previsão a temperatura máxima (°C) é, certamente, uma ferramenta valiosa. Para trabalhos futuros, seria interessante investigar a precipitação pluviométrica na cidade estudada, tendo como camada de entrada a temperatura máxima (°C), mínima, velocidade do vento e a umidade relativa do ar; além da temperatura máxima (°C). É importante salientar que além da modelagem realizada nesta pesquisa, é necessário que se faça a previsão da série estudada para tempos futuros. Para isso, além de utilizar a interface gráfica NNTool, haverá a necessidade da elaboração de um código para realizar esta operação.

## REFERENCIAS

BARRETO, Jorge M. Introdução às Redes Neurais Artificiais. Departamento de Informática e Estatística, UFSC, 2002.

BRAGA, A. P.; CARVALHO, A. C. P. F.; LUDERMIR, T. B. Fundamentos de Redes Neurais Artificiais. Rio de Janeiro: XI Escola de Computação, 1998, v. 1. 245 p.

BRAGA, A. P. Redes Neurais Artificiais: Teoria e Aplicações. Rio de Janeiro: LTC, 2000.

BISHOP, C. M. Neural Networks for Pattern Recognition. Oxford University Press. 1995

DA SILVA, I.N., SPATTI, D.H. & FLAUZINO, R.A. (2010) “Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas”, Artliber Editora Ltda., ISBN: 9788588098534.

Fundação Amazônia de Amparo a Estudos e Pesquisas (FAPESPA) Estatísticas Municipais Paraenses: Castanhal. / Diretoria de Estatística e de Tecnologia e Gestão da Informação. – Belém, 2016.

FRANK ROSENBLATT, The Perceptron, A Perceiving and Recognizing Automaton, Project Para Report No. 85-460-1, Cornell Aeronautical Laboratory (CAL), Jan. 1957

HEBB, Donald O. The Organization of Behavior: A Neuropsychological Theory. New York: Wiley, 1949. 335p.

HINRICHS, Roger A.; KLEINBACH, Merlin; DOS REIS, Lineu Belico. Energia e Meio Ambiente. Tradução técnica: Lineu Belico dos Reis, Flávio Maron Vichi, Leonardo Freire Mello. São Paulo: Cengage Learning, 2010.

MCCULLOR, W. S., PITTS, W., “A logical calculus of ideas Immanent in Nervous Activity” Bulletin of Mathematical Biophysics 5, pp. 115-133. Reprinted in Anderson and Rosenfeld [1988]. 1943.

MEKKI, H.; MELLIT, A.; SALHI, H.; KHALED, B. Modeling and simulation of photovoltaic panel based on artificial neural networks and VHDL-language BT - 14th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2007, December 11, 2007 - December 14, 2007. , p. 58–1424413788, 2007. Disponível em:

MINSKY, M. L. Theory of neural-analog reinforcement systems and its application to the brain model problem. Thesis (Doctor of Philosophy) – Princeton University, University Microfilms, Ann Arbor, 1954.

MINSKY, M. L.; PAPERT, S. A. Perceptrons: an introduction to computational geometry. Cambridge, MA: MIT Press, 1969.

MIRANDA, F. A.; FREITAS, S. R. C.; FAGGION, P. L. Integração e interpolação de dados de anomalias ar livre utilizando-se a técnica de RNA e krigagem. Boletim de Ciências Geodésicas, v.15, n.3, p. 428-443, 2009.

NREL. National renewable energy laboratory: research cell efficiency records. 2016. Disponível em: <<https://www.nrel.gov/pv/>> . Acesso em: 24/04/18

RUMELHART, D. E.; MCCLELLAND, J. L. Parallel distributed processing: explorations in the microstructure of cognition foundations. MIT Press, 1986.

SABERIAN, A.; HIZAM, H.; RADZI, M. A M.; KADIR, M. Z. A.; MIRZAEI, M. Modelling and Prediction of Photovoltaic Power Output Using Artificial Neural Networks., v. 2014, 2014.

SABBATINI RME – Aplicações biomédicas de redes neurais artificiais. Não publicado, 1991.

TAFNER, M.; XEREZ, M.; RODRIGUES, I. Redes neurais artificiais: Introdução e princípios de neurocomputação. Blumenau: EKO/FURB, 1996.

TATIBANA, Cassia Yuri; KAETSU, Deisi Yuki. Redes Neurais. Disponível em: . Acesso em: 09 de novembro 2017.

VELLASCO, M. M. B. R.,2007. Redes Neurais Artificiais. Laboratório de Inteligência Computacional Aplicada, PUC, Rio de Janeiro, RJ, Brasil.

TATIBANA, Cassia Yuri; KAETSU, Deisi Yuki. Redes Neurais. Disponível em: . Acesso em: 09 novembro 2017.

WIDROW, B.; HOFF, M. E. Adaptive switching circuits. In: IRE WESCON CONVENTION RECORD, v. 4, n. 1, p. 96-104. 1960.

Matlab, Software, versão 2015.