OSCAR GABRIEL FILHO

INTELIGÊNCIA ARTIFICIAL NEURAL E GENERATIVA



Blucher

Oscar Gabriel Filho

Inteligência Artificial neural e generativa

Inteligência Artificial neural e generativa
© 2026 Oscar Gabriel Filho
Editora Edgard Blücher Ltda.

Publisher Edgard Blücher
Editor Eduardo Blücher
Coordenador editorial Rafael Fulanetti
Coordenadora de produção Ana Cristina Garcia
Preparação de texto Maurício Katayama
Diagramação Muiraquitã Editoração Gráfica
Revisão de texto Equipe editorial
Capa Juliana Midori Horie
Imagem da capa iStockphoto

Blucher

Rua Pedroso Alvarenga, 1245, 4º andar
04531-934 – São Paulo – SP – Brasil
Tel.: 55 11 3078-5366
contato@blucher.com.br
www.blucher.com.br

Segundo o Novo Acordo Ortográfico, conforme 6. ed.
do *Vocabulário Ortográfico da Língua Portuguesa*,
Academia Brasileira de Letras, julho de 2021.

É proibida a reprodução total ou parcial por quaisquer
meios sem autorização escrita da editora.

Todos os direitos reservados pela Editora Edgard Blücher Ltda.

Dados Internacionais de Catalogação na Publicação (CIP)
Heytor Diniz Teixeira, CRB-8/10570

Gabriel Filho, Oscar
Inteligência artificial neural e generativa / Oscar Gabriel Filho.
– São Paulo : Blucher, 2026.

292 p.: il

Bibliografia
ISBN 978-85-212-2974-2 (Impresso)
ISBN 978-85-212-3053-3 (Eletrônico – Epub)
ISBN 978-85-212-2973-5 (Eletrônico – PDF)

1. Inteligência artificial. 2. Redes neurais artificiais. 3. Inteligência artificial neural. 5. Inteligência artificial generativa. I. Título.
CDU 004.8

Índice para catálogo sistemático:
1. Inteligência artificial CDU 004.8

Prefácio

Este é um compilado atualizado e ampliado de publicações feitas por este autor nos últimos anos, procurando difundir diversos temas da área da inteligência artificial (IA), neste livro delimitada às áreas conexionista (redes neurais artificiais) e generativa, com o objetivo de proporcionar ao leitor um entendimento amplo sobre alguns aspectos fundamentais dessas instigantes áreas do conhecimento, para permitir o acompanhamento do estágio de desenvolvimento em que se encontram na atualidade.

Em algumas pesquisas de ponta promovidas pelas grandes empresas de tecnologia no mundo, notadamente nas áreas de IA, existem passagens no desenvolvimento teórico e/ou prático que não são devidamente esclarecidas, supostamente motivadas por interesses envolvendo a confidenciabilidade do assunto, o que só faz despertar a curiosidade científica dos estudiosos no preenchimento dessas lacunas, visando a compreensão completa, passo a passo, de maneira a propiciar um melhor aproveitamento e o desenvolvimento de novas aplicações nessas áreas do conhecimento.

É o que ocorreu no famoso artigo "Attention is all you need", publicado em 2017, por um grupo de cientistas do Google Brain, ao não descrever como realmente são obtidas as matrizes de consulta W_q, de chave W_k e de valor W_v, fundamentais para a implementação computacional da revolucio-

nária técnica de Atenção, desenvolvida por aqueles cientistas. Aqui você conhecerá uma maneira de calcular essas matrizes, sem que se possa afirmar categoricamente que são iguais às matrizes do trabalho original, a não ser pela coerência e consistência matemática dos resultados obtidos, mesmo porque não foi divulgado o método original de como essas matrizes foram obtidas.

Esta é a proposta principal deste livro – suprir lacunas com informações técnicas, objetivando fornecer uma compreensão completa do método aplicado, isto é, como se desencadeia passo a passo, sempre fundamentado na matemática, como garantidora dos resultados obtidos nas aplicações das ferramentas de IA.

Entretanto, faz-se imprescindível incentivar o leitor a adotar uma postura de verificação diante das propostas inovadoras contidas neste livro, com a recomendação de testá-las antes num ambiente de aplicação, para somente depois aplicá-las ao problema real de interesse.

A intenção é que o seu conhecimento seja enriquecido com a leitura deste livro, com informações minuciosas sobre alguns aspectos teóricos e práticos relevantes, que aqui se encontram selecionados. Tenho a convicção de que alguma parte deste livro vai contribuir para enriquecer o seu conhecimento sobre a IA neural e generativa.

Sugestões e comentários enviados à editora Blucher serão bem-vindos.

O Autor
Natal (RN), julho de 2026

Sumário

CAPÍTULO 1
Inteligência artificial 17
CAPÍTULO 2
Redes neurais artificiais 23
CAPÍTULO 3
O algoritmo *Backpropagation* 43
CAPÍTULO 4
RNA: assegurando a qualidade da aprendizagem 57
CAPÍTULO 5
RNA: acelerando a convergência da rede neural 63
CAPÍTULO 6
RNA: fugindo dos mínimos locais 73
CAPÍTULO 7
IA generativa 79
CAPÍTULO 8
Similaridade e/ou autorregressividade 109

CAPÍTULO 9

Transformer: (uma) descrição do funcionamento 117

CAPÍTULO 10

Interpretabilidade, explicabilidade e aplicabilidade 149

CAPÍTULO 11

Identificação dos erros na aprendizagem de máquina 153

CAPÍTULO 12

IA generativa: uma percepção sobre alucinação 181

CAPÍTULO 13

Aprendizagem de máquina: refinamento e retreinamento 185

CAPÍTULO 14

Agentes inteligentes 189

CAPÍTULO 15

Aleatoriedade é tudo que a criatividade precisa 197

APÊNDICE A

A matemática do *Backpropagation* 207

APÊNDICE B

O algoritmo *Backpropagation* 219

APÊNDICE C

Redes neurais em sistemas generativos 233

APÊNDICE D

Implementação de atenção multicabeças 251

Referências 277

Glossário 279

Índice remissivo 287

CAPÍTULO 1
Inteligência artificial

1.1 INTRODUÇÃO

É bastante notória a importância que a inteligência artificial (IA) tem adquirido nos últimos anos, notadamente a partir de 2017, com a publicação do artigo "Attention is all you need", por Vaswani e outros cientistas do Google Brain, revolucionando, a partir de então, a maneira de capturar e processar as similaridades e as relações de proximidade dos objetos dentro de um determinado contexto, visando principalmente dar soluções para problemas de interesse envolvendo a comunicação através da linguagem natural, considerada uma das manifestações mais inteligentes do ser humano.

Neste ponto inicial, surge a seguinte questão: o que significa a palavra "inteligente". Ela serve para designar um pensamento ou ação resultante do uso da inteligência, cujo significado pode ser compreendido analisando a sua origem, que vem do latim, escrita como *intelligere*, a qual é formada por dois termos: *inter*, que significa entre, e *legere*, que significa capacidade de fazer escolhas, portanto, podemos concluir que "inteligência" é a capacidade de fazer boas escolhas, isto é, de tomar decisões acertadas.

Pois bem, foi exatamente nessa linha de raciocínio que o cientista britânico Allan Turing propôs, em 1950, um teste de inteligência, para identificar se

uma determinada máquina consegue imitar o ser humano, a ponto de enganá-lo como se também fosse uma pessoa, em uma conversação cega, onde o avaliador humano tenta distinguir se está se comunicando com outro ser humano ou com uma máquina, por meio de conversas em linguagem natural, na qual os dois falam entre si, porém não se veem.

É exatamente nessa perspectiva de discussão que acho razoável manter este livro, quando a questão é definir se algo tem comportamento inteligente ou não, ou seja, se a máquina consegue apresentar um desempenho similar ou mesmo melhor do que o ser humano em tarefas úteis, tornando a sua utilização atrativa em certas áreas de atuação no mundo real.

Consequentemente, torna-se árido discutir o que venha a ser a inteligência da máquina, notadamente quando se usam para comparação certas habilidades manifestadas pelos seres humanos, sob as quais se processa a inteligência humana, embora reconheça serem de grande valia como fonte de inspiração para a evolução da IA, como tem ocorrido desde o início do conexionismo, em 1943, com a criação do neurônio artificial por McCulloch e Pitts, mas sem o futurismo de virem a se igualar em algum dia vindouro.

Até porque é temerário fazer predições nessa área do conhecimento sem possuir evidências concretas, considerando que, na história da evolução da ciência e tecnologia, alguns cientistas, mesmo as tendo, foram inicialmente contestados nas suas descobertas e, não raras vezes, desacreditados perante a sociedade, somente sendo reconhecidos tempos depois. Foi o caso de Lofti A. Zadeh, professor premiado da Universidade da Califórnia, Estados Unidos, nascido no Azerbaijão, que teve a sua reputação contestada quando propôs a teoria dos sistemas e conjuntos *fuzzy*, ferramenta da abordagem simbólica da IA, publicada em 1965, hoje muito importante para a comunicação do ser humano com a máquina, feita através da modelagem matemática de variáveis linguísticas ditas nebulosas (*fuzzy*) ou imprecisas, antes inimaginável acreditar que poderia acontecer.

O que reforça a ciência da IA é que, além de seus resultados parecerem uma magia, eles têm seu respaldo na matemática, que possibilita demonstrar antecipadamente as regras básicas que asseguram o seu desempenho, na solução de um determinado problema de interesse, o que não impede de cometer erros. Se, por um lado, a matemática é exata e possibilita o desenho de boas soluções, por outro lado, a máquina digital não é, pois essa estará sempre sujeita a cometer erros provenientes de sua finitude e dos arredonda-

mentos da digitalização, por ínfimos que possam vir a ser, mas que em circunstâncias críticas podem ocasionar a troca de "gato por lebre".

Acima de tudo, a matemática acaba sendo um divisor de águas entre as pessoas interessadas exclusivamente na sua utilização e outras interessadas também em compreender o funcionamento das ferramentas de IA.

Por um lado, temos uma multidão ávida por saber usá-la em suas atividades, desde o entretenimento até aplicações comerciais e industriais, existindo entre essas pessoas algumas poucas interessadas no que está por trás que a faz dar bons resultados. Pelo outro lado, estão os estudiosos interessados em desenvolver novas tecnologias para o mercado, embasadas nas descobertas das grandes empresas de tecnologia e das universidades existentes pelo mundo, sendo estes últimos o grupo detentor de sólidos conhecimentos em matemática.

Os capítulos que virão vão remeter o leitor para entender os fundamentos da IA neural e a IA generativa, não sendo possível escapar de fornecer alguns conceitos essenciais da matemática para a compreensão completa do conteúdo, principalmente no que diz respeito aos aspectos relacionados com critérios de otimização numérica, por exemplo, o cálculo de derivadas.

Para os leitores com pouco ou nenhum conhecimento sobre derivada matemática, aqui vai uma dica: nas fórmulas em que ela aparece, ela quer apenas indicar que está sendo levado em conta no cálculo a variação da grandeza principal, a que está sendo derivada, em relação a uma outra grandeza de referência. Por exemplo, se você se deparar com a expressão do tipo $\partial erro/\partial peso$, isso representa que está sendo levado em conta o quanto varia o erro em relação ao peso (derivada parcial), mantidas as demais grandezas constantes, isto é, sem que possam variar simultaneamente com as grandezas citadas na fórmula, pois as grandezas que podem variar são somente o erro e o peso. O símbolo ∂ (lê-se del) indica a variação de uma grandeza e a barra inclinada não é uma operação de divisão, apenas indica uma relação entre duas grandezas que estão variando simultaneamente.

Enfim, o propósito desta obra é: 1) responder ou indicar um caminho confiável para obter respostas, no que se refere à configuração e desempenho do treinamento de uma rede neural artificial (RNA), principal ferramenta para a construção dos sistemas generativos baseados nos mecanismos de Atenção, os quais vêm predominando nos aplicativos voltados para essa área

do conhecimento, desde 2017, e 2) mostrar o funcionamento de uma IA generativa, baseada no *Transformer* de Vaswani e outros.

A seguir, serão apresentados alguns conceitos que ajudarão na compreensão para onde caminha a IA no futuro próximo, apenas para provocar a curiosidade sobre possíveis áreas de atuação e desenvolvimento.

1.2 SENSIBILIDADE, RACIOCÍNIO E INTUIÇÃO HUMANOS

A **sensibilidade** nos faz detectar os objetos ao nosso redor. Entretanto, o entendimento só é possível se existir no nosso cérebro um conjunto de conhecimentos *a priori* (anteriores) capazes de pensar a natureza dos objetos, como as noções de tamanho, cor, posição, deslocamento. Assim, destituído de sensibilidade nenhum objeto nos seria dado e, isento de entendimento, nenhum objeto seria pensado (Kant, 2003).

Segundo Aristóteles, nada está no intelecto, sem antes passar pelos sentidos. Intelecto, ainda conforme o filósofo, "é a parte da alma com a qual ela conhece e pensa" (*apud* Tomás de Aquino, [s.d.]). Pelo dicionário online Oxford Languages and Google, intelecto é sinônimo de entendimento, mente.

Ainda dentro desse contexto, pode-se inferir que o **raciocínio** humano nada mais é do que a manipulação do conhecimento sobre os fenômenos ou fatos, com o objetivo de chegar a alguma conclusão interessante.

E de uma sequência de fenômenos mentais surge a **intuição**, uma sensação na forma de uma predição resultante do entendimento extrapolado para prever o que futuramente poderá ocorrer com os objetos manipulados pelo pensamento, caso a dinâmica dos acontecimentos se mantenha no seu curso. Interessante registrar que intuição é um tipo de predição que está sendo realizada fora do domínio atual do fenômeno observado, tratando-se, pois, de uma extrapolação, enquanto a generalização é uma espécie diferente de predição, pois é realizada dentro do domínio do fenômeno observado, ou seja, mais se assemelha com a interpolação.

De tudo o que foi mencionado anteriormente, destaca-se a sensibilidade humana que extrai informações (dados) do ambiente, experimentando-o e usando os sensores da visão, odor, som, paladar e tato, na forma de dados que serão transformados em conhecimentos do mundo real. Outro destaque é a forma de processar os conhecimentos existentes, o que denominamos de

raciocínio, o qual leva a formular novos conhecimentos provenientes das experiências realizadas de modo autônomo.

Construir uma IA que, no futuro não muito distante, possa tomar decisões baseada, em parte ou totalmente, no seu próprio conhecimento, fruto de ações independentes de "atuar sobre o mundo e raciocinar", provavelmente será o caminho a ser explorado para atribuir à máquina certa autonomia, ao dotá-la de iniciativa para obter o seu próprio conhecimento e, assim, poder raciocinar de forma independente ou conjuntamente com o conhecimento humano previamente armazenado.

Atualmente, esta é uma das vertentes do campo de atuação na qual está sendo desenvolvida a inteligência artificial geral (*artificial general intelligence* – AGI), procurando aproximar-se do comportamento humano, sendo um desafio a ser superado nos próximos anos, isto é, gerar e utilizar, de alguma forma inteligente, o seu próprio conhecimento para resolver problemas do mundo real.

1.3 CONCLUSÕES

A cada dia que passa, a IA vem atraindo a atenção e sendo utilizada por muitas pessoas e empresas mundo afora, existindo quem diga que ela será a tecnologia dominante nas próximas décadas. A aposta intelectual e financeira vem escalando muito rapidamente, sendo que nos últimos anos a unidade de tempo para medir o desenvolvimento tecnológico tem encurtado cada vez mais, e praticamente todo dia se tem uma novidade na área da IA.

Dominar os seus fundamentos, quer como usuário quer como desenvolvedor, tornou-se imperativo, em face dos desafios impostos por mercados de trabalho exigentes, onde as tarefas aumentam de complexidade e ficam cada vez mais dependentes dos dispositivos digitais, para mantermo-nos competitivos diante de uma concorrência universal acirrada.

A competência hoje certamente passa pelo domínio e aplicação de ferramentas inteligentes que permitam extrair a máxima eficiência dos processos, favorecendo consequentemente a sua automatização, e é nesse campo de atuação que a IA tem muito para contribuir e um papel importante a desempenhar – promover a execução automática das atividades de lazer e produtivas, ajudando os seres humanos a atingirem altos patamares de eficiência na execução dos processos.

Um próximo passo que se vislumbra na evolução da IA será dotar as máquinas de iniciativa para "agir sobre o mundo e raciocinar", como sendo uma ação voluntária, segura no sentido de não causar danos e de caráter excitativo para provocar uma resposta, com o objetivo de adquirir novos conhecimentos, visando usá-los de modo independente ou em conjunto com o conhecimento humano previamente armazenado na máquina, de forma inteligente, isto é, na busca de escolhas mais eficientes.

CAPÍTULO 2
Redes neurais artificiais

2.1 INTRODUÇÃO

As redes neurais artificiais (RNAs) atualmente são a ferramenta de aprendizagem mais importante da inteligência artificial (IA). Pensadas originalmente para imitar o comportamento do cérebro humano, como uma estratégia para emular a nossa maneira de aprender e armazenar o conhecimento, as RNAs são usadas em tarefas que exigem a predição de eventos futuros.

No cérebro humano, essas redes são compostas por muitos neurônios conectados entre si, disparando impulsos elétricos através das ramificações de saída chamadas de axônio, fazendo com que os disparos se repitam neurônio após neurônio localizados à frente, até produzirem um efeito resultante, por exemplo, recuperar uma informação armazenada na memória e transformá-la numa manifestação sonora (voz), num processo de comunicação entre seres humanos.

2.2 O NEURÔNIO ARTIFICIAL DE McCULLOCH E PITTS

Pensando numa forma de reproduzir matematicamente o processo de aprendizagem humano, em 1943, o médico neurofisiologista Warren McCulloch e o estatístico Walter Pitts propuseram o primeiro modelo matemático para o

funcionamento do neurônio biológico (Kovács, 1996), que resultou no neurônio artificial de McCulloch e Pitts (MCP), mostrado na Figura 2.1, composto de: 1) '*n*' entradas, 2) a soma das '*n*' entradas menos uma entrada especial chamada de *threshold* ou limiar de operação, cujo resultado final é chamado de NET, 3) uma função matemática que modela o comportamento do neurônio, chamada de Função de Ativação do Neurônio (FAN), e, finalmente, 4) a saída do neurônio artificial, dada pelo valor da função calculado para cada valor de NET, simbolicamente calculado por f(NET).

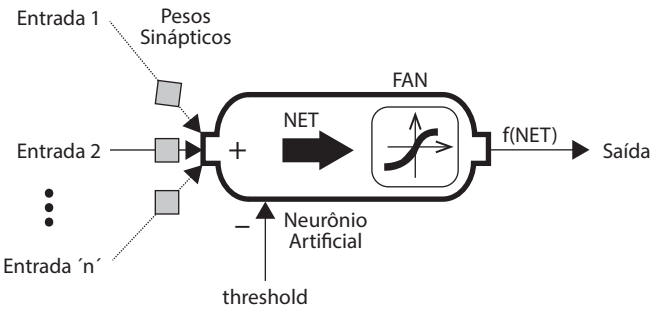


Figura 2.1 – Neurônio artificial de McCulloch e Pitts (MCP).

Importante ressaltar que a estrutura do neurônio artificial é formada basicamente pela operação de soma (+) e pela Função de Ativação do Neurônio (FAN), lembrando que o sinal interno chamado de NET é a soma de todas as entradas do neurônio artificial, após passarem pelos pesos sinápticos externos de ligação do neurônio com os neurônios anteriores. A Figura 2.1 mostra o neurônio original de McCulloch e Pitts, em que consta a subtração do *threshold*, indicando que o neurônio somente se torna ativo após ultrapassar este patamar, ou seja, após exceder numericamente o valor do limiar de operação, em cada neurônio – claro que isto ocorre somente nos neurônios do tipo "tudo ou nada". Para facilitar o cálculo do limiar de operação para cada neurônio artificial, utiliza-se o artifício matemático que consiste em transferir o valor do *threshold* para dentro da estrutura matricial dos pesos sinápticos que alimentam o referido neurônio, ou seja, acrescentar um peso sináptico extra à matriz dos pesos posicionados antes do neurônio. Este peso extra recebe o nome de *bias* (Figura 2.2), em português viés ou desvio, sendo atribuído a sua entrada o valor constante igual a -1, apenas com o objetivo de torná-los numericamente iguais entre si (Gabriel Filho, 2023), isto é, forçar que o *bias* seja igual ao *threshold*.

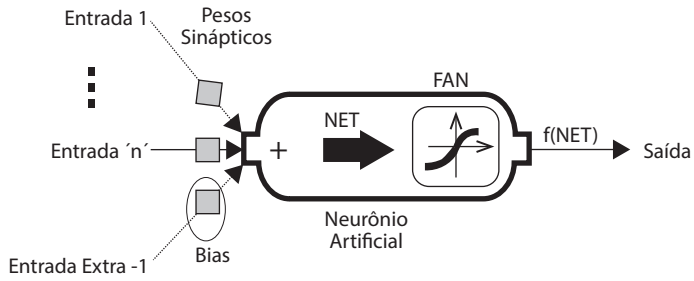


Figura 2.2 – Neurônio artificial MCP modificado.

Assim, o cálculo do *bias* (Figura 2.2) é forçado a ser numericamente igual ao *threshold* (Figura 2.1) e passa a ser realizado automaticamente, como um elemento adicional da matriz dos pesos sinápticos. Portanto, a justificativa para atribuir à entrada extra o valor -1 é exclusivamente matemático, somente para fazer com que a modificação imposta pelo *bias* preserve o mesmo valor numérico do limiar de operação (*threshold*), coerente com o que foi proposto originalmente no artigo de McCulloch e Pitts.

2.3 A REDE NEURAL ARTIFICIAL

A justaposição de vários neurônios interligados entre si forma o que chamamos de rede neural artificial (RNA), projetada com a finalidade de aprender a relação existente entre os pares de entrada-saída desejada, obtidos de experimentos previamente tabelados. Portanto, esse tipo de aprendizagem, denominada de aprendizagem supervisionada, consiste na apresentação de pares de dados de treinamento à rede neural, sendo repetida várias vezes num processo contínuo, até que a rede consiga extrair um modelo matemático, capaz de reproduzir com certa precisão, dentro de uma tolerância previamente estabelecida pelo usuário, a relação entre as entradas e respectivas saídas.

Portanto, uma RNA é um conjunto de neurônios artificiais dispostos em camadas e ligados entre si por meio de pesos sinápticos, os quais são frequentemente designados pela letra *w* (inicial de *weight*, peso em inglês), e que são os responsáveis por dosar ou regular os sinais que propagam de um neurônio para outro.

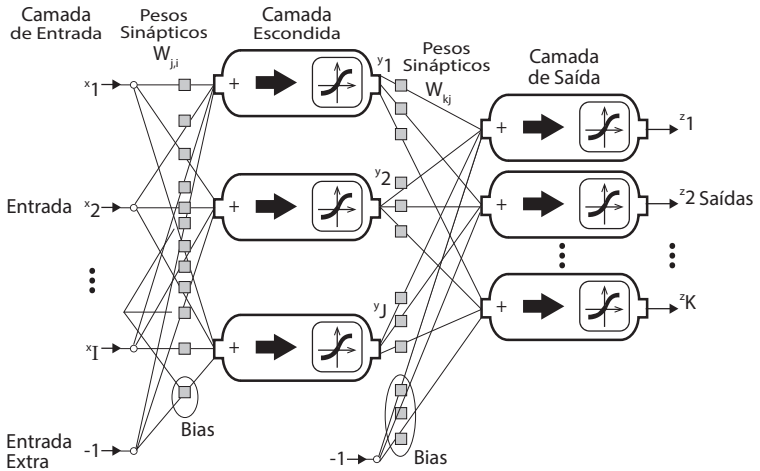


Figura 2.3 – Uma RNA com três camadas: entrada, escondida e saída.

É importante ressaltar que:

- Os pesos sinápticos não fazem parte do neurônio artificial (Figura 2.1), pois eles são as ligações entre dois neurônios consecutivos de uma rede neural (Figura 2.3).
- As entradas dos neurônios de uma rede neural (Figura 2.3) devem incluir uma entrada extra para o *bias*, sendo que o *bias* é definido como sendo um peso sináptico extra, em substituição ao limiar de operação (*threshold*) do neurônio artificial (Figura 2.1) proposto originalmente no modelo matemático de McCulloch e Pitts, devido à vantagem de o *bias* ser um valor extra ajustado automaticamente, por ter sido incluído dentro da estrutura matricial dos pesos sinápticos existentes entre todas as camadas da rede.
- A rede neural, durante a fase do treinamento, evolui tanto para frente (*forward*), no cálculo direto da saída ou das saídas, quanto para trás (*backward*), quando está corrigindo ou ajustando os seus pesos sinápticos, sendo esta etapa chamada de retropropagação do erro (*error back-propagation*), enquanto na fase de inferência (generalização ou predição), após ser treinada, a rede somente evolui para frente.

Portanto, constata-se na Figura 2.3 que uma rede neural é formada por várias camadas justapostas, dispostas obedecendo à ordem da camada de entrada (*input*), camadas escondidas (*hiddens*) e camada de saída (*output*), sendo que a camada de entrada não possui neurônios, isto é, somente serve

como uma porta de entrada para a rede neural e, portanto, não realiza processamento.

Uma decisão importante a ser tomada é definir quantas camadas escondidas a rede possuirá, podendo ter somente uma camada escondida, o que resultará no tipo denominado de rede neural rasa (*shallow neural network*), ou mais de uma camada escondida, tipo de rede denominado de rede neural profunda (*deep neural network*). Fixada a quantidade de camadas escondidas, o próximo passo é definir quantos neurônios a rede terá em cada camada, exceto na camada de entrada que não tem neurônio. É importante ressaltar que a quantidade de entradas e a quantidade de neurônios na camada de saída, tendo em vista a utilização do paradigma da aprendizagem supervisionada, dependem exclusivamente da aplicação específica da rede neural.

Um balizador para ajudar a definir a quantidade de neurônios na camada escondida ou nas camadas escondidas deve levar em consideração dois aspectos fundamentais: um é a quantidade de dados a serem treinados, ou seja, de pares de treinamento, e o outro aspecto é a complexidade do problema a ser modelado (linear, quadrático, exponencial etc.), para prevenir a ocorrência das situações anormais de funcionamento denominadas de subajuste (*underfitting*) e de sobreajuste (*overfitting*).

O subajuste (*underfitting*) ocorre quando surge um viés ou desvio acentuado (em inglês, *bias*; não confundir com o *bias* do neurônio artificial da Figura 2.2) nos dados inferidos ou preditos pela rede, que indica um modelo mal treinado, isto é, com baixa precisão, sendo resultante de um treinamento utilizando muitos dados, porém uma arquitetura de baixa complexidade (poucos neurônios), que não consegue nem bons resultados (validação) nos pontos usados no seu próprio treinamento, revelando um comportamento indesejável, no qual o modelo apresenta erros com valores altos, tanto na fase de treinamento quanto na fase de generalização ou predição.

No extremo oposto, pode ocorrer o sobreajuste (*overfitting*), que é identificado pelo surgimento de uma variância de alto valor nos dados estimados, resultante da utilização de uma arquitetura de alta complexidade (muitas camadas contendo muitos neurônios) para poucos dados de treinamento, que produz um erro de treinamento muito pequeno e um erro de generalização muito alto. Neste caso, diz-se que a rede aprendeu muito de pouco, chegando

até mesmo a aprender o ruído normalmente embutido nos dados de entrada durante a medição, em detrimento da sua capacidade de generalização ou predição.

Uma visão completa desses dois comportamentos indesejáveis, que levam a uma rede neural cometer erros de aprendizagem inaceitáveis, é apresentada no Capítulo 11, em que são analisados os aspectos relacionados aos erros cometidos, que são devidos à quantidade de dados de treinamento disponíveis e à complexidade do modelo escolhido.

A ideia agora é propor uma arquitetura de rede neural e ilustrar o seu funcionamento, com a finalidade de aprender a relação existente entre os pares de entradas e suas respectivas saídas desejadas (rótulos), definidos pelos pontos: (1;1), (2;4) e (3;9). Vê-se, portanto, que se trata da operação de elevar um número inteiro ao quadrado, o que implicará na adoção de um modelo não linear. Uma possível rede neural poderia ter a seguinte configuração:

- Arquitetura 1:6:1, indicando que se trata de uma rede com 1 entrada, 6 neurônios na camada escondida e 1 neurônio na camada de saída, lembrando que a camada de entrada não executa processamento.
- Funções de ativação Tangente Hiperbólica (Sigmoide Bipolar) na camada escondida e Linear na camada de saída (a camada de entrada não possui neurônio, apenas serve como um ponto de alimentação da rede); de acordo com o que foi mencionado anteriormente, a função Tangente Hiperbólica foi escolhida por ser não linear (uma curva na forma de um 'S', por isso também chamada de Sigmoide) e, além disso, possuir todas as suas saídas com valores negativos e positivos.
- Tolerância escolhida = 0,001 (10^{-3}).

É uma tarefa relativamente simples treinar uma rede neural para aprender a operação de elevar ao quadrado, embora seja considerada uma função matemática de média complexidade, por se tratar de uma função não linear, motivo pelo qual foi escolhida a função de ativação Tangente Hiperbólica, também conhecida como Sigmoide Bipolar, isto é, que tem a propriedade de gerar apenas números positivos.

Então, a RNA anteriormente proposta foi implementada num computador pessoal, gerando uma sequência de telas produzidas pelo programa desenvolvido por este autor lá pelo ano de 1994, em que foi utilizada a linguagem científica de alto nível mais usada na época, a linguagem Pascal, como

fruto de um trabalho acadêmico de pós-graduação, e que foi sendo melhorado e adaptado conforme a evolução das redes neurais com o passar do tempo, até a atualidade.

Na Figura 2.4 se apresenta apenas a tela de abertura do programa, em que é mencionada a utilização do algoritmo de Propagação Retroativa do Erro (*Error Backpropagation*), mais conhecido simplesmente como *backpropagation*, e que será exaustivamente explicado em vários capítulos à frente, cada um tratando de uma determinada peculiaridade durante o processo de treinamento.



Figura 2.4 – Tela de abertura do programa.

A Figura 2.5 mostra a configuração da rede neural, pois observe que foram impostos limites para a quantidade de camadas e de neurônios por camada (nós), a ocorrência de erros de estouro relacionados com a capacidade de armazenamento e de ponto flutuante do computador pessoal usado.

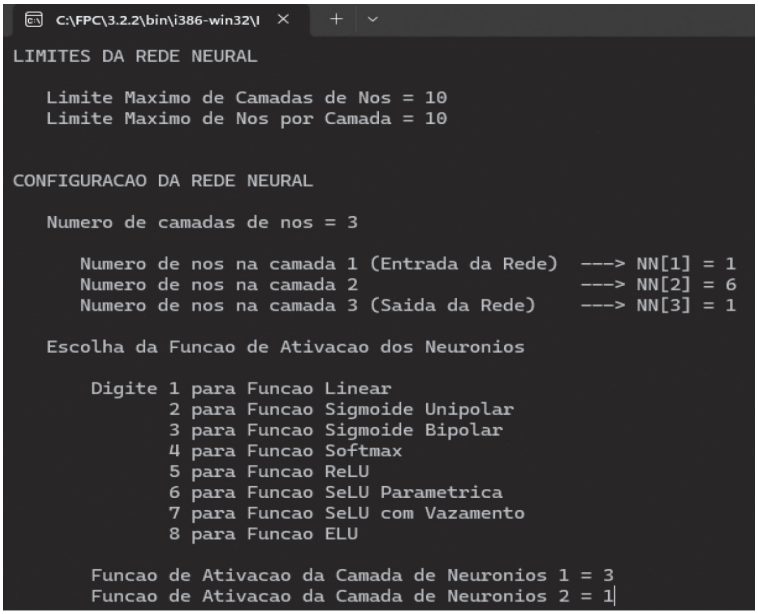


Figura 2.5 – Tela de configuração da arquitetura.

A Figura 2.6 mostra a tela de entrada de dados de treinamento, notando-se o valor da entrada extra –1 para todos os neurônios, o que corresponde à entrada extra na Figura 2.3, ou seja, a entrada do *bias*.

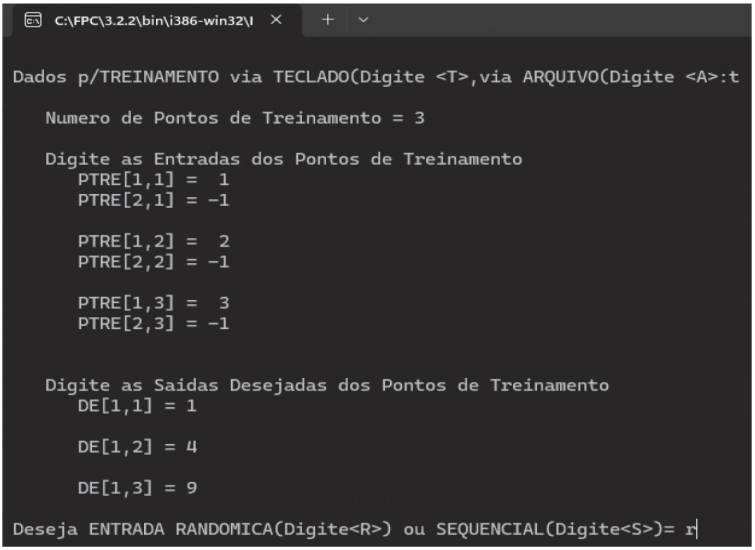


Figura 2.6 – Tela de configuração das entradas e saídas desejadas.

A Figura 2.7 mostra a tela de ativação do método ETA-adaptativo (ver o Capítulo 5), responsável pelo ajuste automático da taxa de aprendizagem durante a execução do programa, com o objetivo de encontrar uma taxa que proporcione melhoria da velocidade de convergência, sem comprometer a estabilidade do processo de treinamento. Neste caso específico, o valor do ETA inicial escolhido foi 0,01 (recomenda-se valores menores que 1), sendo que o treinamento foi concluído certamente com um outro valor (ETA calculado), o que melhor pode ser obtido usando esse método adaptativo no processo de aprendizagem (ver a Figura 2.8).

Caso a resposta para a ativação do ETA-adaptativo seja não (N), a tela mostrará uma linha para o usuário digitar o valor de um ETA constante, sendo que o valor escolhido pelo usuário manter-se-á inalterado até encerrar o treinamento.

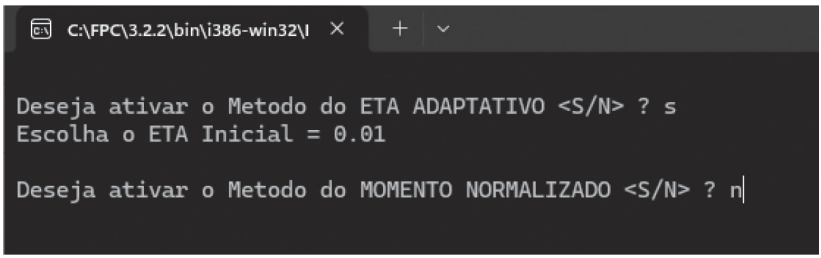


Figura 2.7 – Tela de configuração do ETA-adaptativo e do momento.

Além de buscar a melhor taxa de aprendizagem em tempo de execução, a Figura 2.7 também oferece a opção de ativar o Método do Momento (ver o Capítulo 6), sendo que este serve para permitir que o processo de convergência não fique preso nas armadilhas dos pontos de mínimos locais, que dificultam ou até mesmo impedem o treinamento da rede neural. Aqui, da mesma forma do ETA-adaptativo, caso a resposta seja sim (S), a tela mostrará uma linha para o usuário digitar o valor do Momento (recomenda-se um valor entre 0 e 1, de preferência próximo de 1), sendo que o valor escolhido manter-se-á inalterado até encerrar o treinamento.

Finalmente, a Figura 2.8 apresenta alguns resultados importantes para avaliar a qualidade do treinamento, assim como possibilitar a realização de testes de validação e de generalização do modelo neural obtido.

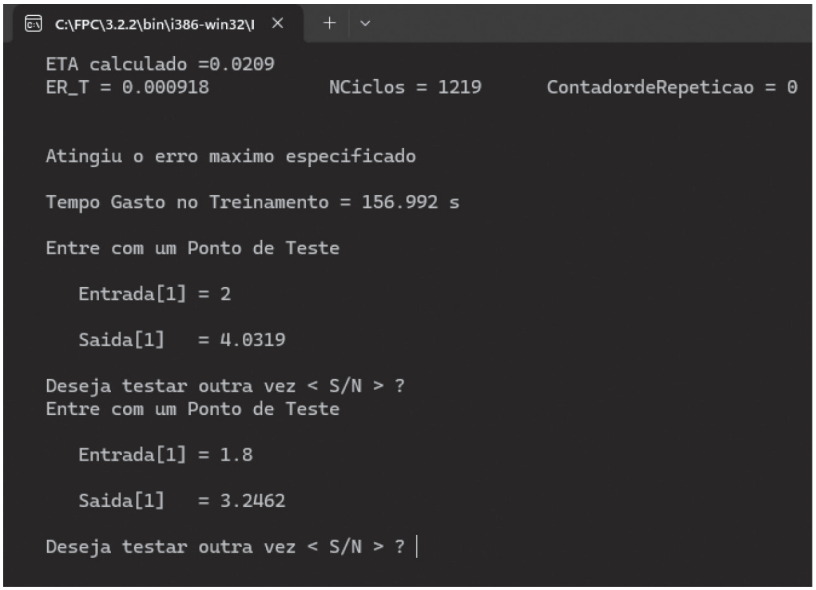


Figura 2.8 – Tela final: resultados e testes.

Note na última figura que a rede neural projetada convergiu após 1219 ciclos de treinamento ou épocas (repetições de cálculos feitos para frente e para trás), após o que apresentou um erro total de 0,000918, portanto menor do que a tolerância previamente escolhida de 0,001. Entretanto, a quantidade dispendida de ciclos revela o grau de dificuldade de modelar funções não lineares, no caso uma função quadrática.

2.4 O PROCESSO DE APRENDIZAGEM DE UMA REDE NEURAL

Desde 1943, com o surgimento do modelo matemático do neurônio de McCulloch e Pitts, fazer com que uma rede neural aprendesse a partir de exemplos ou amostras sempre despertou o interesse da comunidade cientifica, resultando na alternância de períodos de entusiasmo e de descrédito.

As dificuldades somente foram superadas definitivamente em 1986, com a divulgação do livro intitulado *Parallel Distributed Processing* (Rumelhart e McClelland), que divulgou o método da Propagação Retroativa do Erro (*Error Backpropagation*), mais conhecido simplesmente como *Backpropagation*. Desde então, as redes neurais passaram a dominar as aplicações de IA, mais do que as ferramentas simbólicas, por exemplo, os sistemas especialistas, e as evolucionistas, por exemplo, os algoritmos genéticos, embora essas ferramentas

tenham o seu nicho de aplicação na solução de problemas específicos, motivo pelo qual também mantêm o interesse da comunidade cientifica.

2.4.1 Descrição do algoritmo *Backpropagation*

O algoritmo *Backpropagation* é a estratégia responsável pelo ajuste dos pesos sinápticos de uma RNA, que garante o funcionamento conjunto das suas várias pequenas unidades de processamento, dispostas em camadas justapostas e conectadas entre si de alguma forma – essas unidades são os neurônios artificiais.

Seu funcionamento, embora num primeiro contato possa parecer complexo, na verdade é de uma simplicidade incrível, iniciando-se por receber uma entrada numérica, efetuar uma sequência de operações aritméticas em direção às saídas, de modo a fornecer um resultado em cada saída, o qual será comparado com o resultado ideal (desejado), gerando assim um erro individual. Esse foi o caminho de ida, isto é, para frente (*forward*). O erro calculado então retorna pelo mesmo caminho em que foi gerado, só que nesse percurso de volta (*backward*) ele vai corrigindo proporcionalmente cada um dos pesos sinápticos que deram origem ao erro individual, isto é, cada peso é corrigido conforme a sua parcela de "culpa" pelo erro ocorrido na correspondente saída da rede neural.

O *Backpropagation* surgiu para possibilitar o treinamento de várias camadas de neurônios artificiais numa mesma rede neural e, de maneira definitiva, superar as limitações impostas pela falta de um algoritmo capaz de ajustar todos os pesos sinápticos. Logo após o seu surgimento, a partir de 1986 até o início da década de 2010, as aplicações das redes neurais se restringiam praticamente a resolver problemas complexos, envolvendo relações não lineares, isto é, que exigiam a utilização de funções de ativação do neurônio não lineares, como as funções matemáticas Logística e Tangente Hiperbólica.

O emprego dessas funções de ativação não lineares sempre resultava em processos de aprendizagem lentos, devido às repetidas operações matemáticas envolvendo exponenciais, agravados na ocasião pela utilização de computadores com baixa velocidade de processamento, o que acabava por limitar a quantidade de neurônios artificiais. Ademais, existia também uma crença de que a utilização de redes neurais só seria compensatória em aplicações complexas, respaldada pela disponibilidade de uma teoria linear eficiente para tratar os problemas mais simples.

Assim, a necessidade de processar muitos dados fez surgir nos últimos anos as redes neurais profundas (*deep neural networks* – DNNs), dando origem à área de aprendizagem de máquina (*machine learning*) denominada de aprendizagem profunda (*deep learning*), cujo treinamento das redes somente se tornou viável com a criação de novas funções de ativação menos complexa e mais rápidas, por exemplo, a função ReLU (*Rectifier Linear Unit*), entre outras, a serem apresentadas mais adiante neste capítulo.

Na esteira da aprendizagem profunda veio uma demanda crescente por aplicações de processamento da linguagem natural (*natural language processing* – NLP), principalmente em tarefas de comunicação relacionadas com a geração de conteúdos, assim como a necessidade de automatizar os processos de gestão das atividades empresariais, que exigiam a construção de ferramentas capazes de processar uma quantidade muito grande de dados.

Seguindo esse ritmo acelerado de evolução da tecnologia, não somente no que diz respeito ao desenvolvimento de algoritmos sofisticados, como também na fabricação de máquinas com grande poder de computação, eis que surge uma nova abordagem para implementar mecanismos mais eficientes de Atenção (Vaswani *et al.*, 2017), viabilizando assim o desenvolvimento dos modelos grandes de linguagem (*large language models* – LLMs), já a partir do final do ano de 2022 (ChatGPT), e, desde aquele evento, muitas outras ferramentas de IA têm surgido no mercado (Gemini, Claude, DeepSeek etc.), utilizando técnicas avançadas de processamento, e escalando cada vez mais na quantidade de dados e de parâmetros (pesos sinápticos).

2.4.2 As novas funções de ativação do neurônio (FAN)

A grande vantagem trazida por essas funções de ativação é a rapidez no processamento, por possuírem uma formulação matemática simples, pois são funções parcialmente lineares ou lineares em parte. O componente linear favorece bastante a rapidez no cálculo, considerando que é a região ativa da curva, cujos cálculos são repetidos muitas vezes durante o treinamento da rede neural. Vale frisar que, na sua integralidade, não são funções lineares, devido à presença também de algum componente não linear.

*2.4.2.1 Unidade Linear Retificada (*Rectified Linear Unit – *ReLU)*

Consta ter sido a primeira função de ativação do neurônio linear por partes da IA, tendo sido proposta por Hahnloser *et al.* (2000), mas somente de-

monstrada matematicamente em 2011, e a partir dela outras variantes foram surgindo.

Define-se como Unidade Linear Retificada (em inglês *Rectified Linear Unit* – ReLU) a função linear por partes, contendo a parte negativa retificada (constante), dada por

$$f(NET) = \begin{cases} 0, & para\ NET < 0 \\ \beta NET, & para\ NET \geq 0 \end{cases} \quad \textbf{(2.1)}$$

em que a saída $f(NET)$ é igual a zero para valores da entrada menor que zero e é igual a β vezes a própria entrada NET para valores da entrada maiores ou iguais a zero. Aqui, β representa a declividade da parte inclinada da reta, que é matematicamente igual à tangente do ângulo que essa reta faz com o semieixo horizontal positivo, sendo que $\beta = 1{,}0$ corresponde ao ângulo de 45°. Evidentemente, outro ângulo poderá ser escolhido, desde que possa trazer alguma vantagem significativa, como melhorar a convergência da rede neural. O gráfico da função de ativação ReLU é mostrado na Figura 2.9.

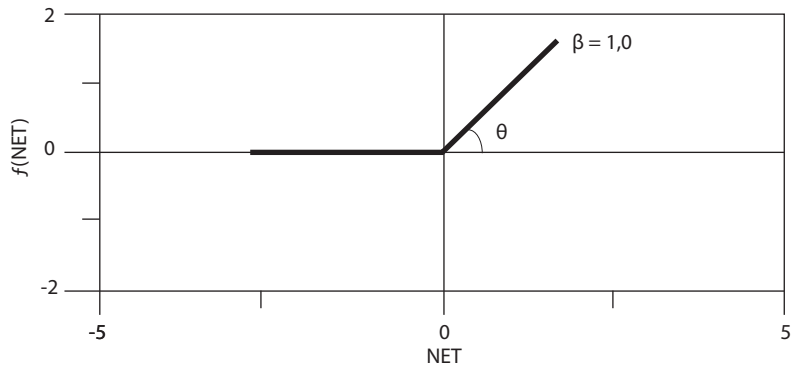


Figura 2.9 – Função de Ativação ReLU.

É imediato notar que a não linearidade consiste no ponto de quebra existente no gráfico da função. Sua derivada primeira é dada por

$$f'(NET) = \begin{cases} 0, & para\ NET < 0 \\ \beta, & para\ NET \geq 0 \end{cases} \quad \textbf{(2.2)}$$

Observe que a função ReLU possui apenas uma região ativa, por exemplo, a reta definida pela declividade ou coeficiente angular $\beta = 1{,}0$, correspondendo ao ângulo $\theta = 45°$. Essa função de ativação possui a vantagem de poder gerar vetores de saída dos neurônios esparsos, isto é, com muitos elementos nulos, o que facilita ainda mais o cálculo iterativo. Entretanto, a ocorrência da saída nula implica na morte do respectivo neurônio (*dying neuron*), o que certamente impedirá a correção dos pesos que chegam ao neurônio, tornando assim a convergência da rede neural mais oscilatória e, consequentemente, mais difícil.

As próximas funções de ativação que serão apresentadas são variantes da função linear retificada (ReLU) e têm a propriedade de evitar a morte da sinapse, simplesmente substituindo a parte nula da função de ativação por uma função com valores negativos próximos de zero.

2.4.2.2 *Unidade Linear Segmentada Paramétrica (*Parametric Segmented Linear Unit – Parametric *SeLU)*

Define-se como Unidade Linear Segmentada Paramétrica, em inglês *Parametric Segmented Linear Unit* (*Parametric* SeLU), a função linear por partes dada por

$$f(NET) = \begin{cases} \alpha NET, & para\ NET < 0 \\ \beta NET, & para\ NET \geq 0 \end{cases} \qquad \textbf{(2.3)}$$

em que a saída $f(NET)$ é igual a α vezes a entrada NET para valores da entrada menores que zero (NET < 0) e é igual a β vezes a entrada NET para valores da entrada maiores ou iguais a zero (NET ≥ 0), observando que α é muito menor que β, ou seja, $\alpha << \beta$.

O gráfico da função de ativação SeLU Paramétrica é mostrado na Figura 2.10.

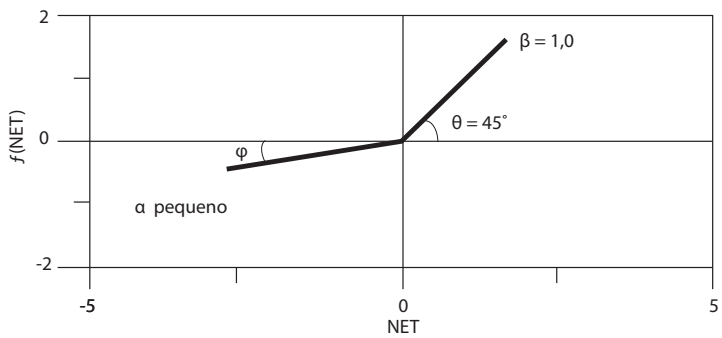


Figura 2.10 – Função de Ativação SeLU Paramétrica.

Aqui, também, a não linearidade é introduzida pela existência de um ponto de quebra no gráfico da função. Sua derivada primeira é dada por

$$f'(NET) = \begin{cases} \alpha, & para\ NET < 0 \\ \beta, & para\ NET \geq 0 \end{cases} \quad \textbf{(2.4)}$$

Na Figura 2.10, α e β representam as declividades das retas situadas no terceiro e primeiro quadrantes, respectivamente, sendo α um valor relativamente pequeno escolhido pelo desenvolvedor. Essa função de ativação elimina a morte do neurônio para $NET < 0$, assim como permite ajustar a influência do sinal de erro retroagido entre as correções negativas e positivas, o que pode acarretar uma melhora na convergência da rede.

Apenas para efeito de comparação, uma declividade igual a 1,0 corresponde ao ângulo de 45°, considerando que a declividade é definida matematicamente como a tangente do ângulo que a reta faz com o semieixo horizontal positivo.

2.4.2.3 *Unidade Linear Segmentada com Vazamento (*Leaky Segmented Linear Unit – Leaky *SeLU)*

Define-se como Unidade Linear Segmentada com Vazamento (em inglês *Leaky Segmented Linear Unit – Leaky* SeLU) a função linear por partes representada por

$$f(NET) = \begin{cases} 0{,}01NET, & para\ NET < 0 \\ \beta NET, & para\ NET \geq 0 \end{cases} \quad \textbf{(2.5)}$$

em que a saída $f(NET)$ é igual a 0,01 vez a entrada NET para valores da entrada menores que zero ($NET < 0$) e é igual a β vezes a entrada NET para valores da entrada maiores ou iguais a zero ($NET \geq 0$), em que β é a declividade da reta situada no primeiro quadrante. Observe que esta função de ativação é um caso particular da função Unidade Linear Segmentada Paramétrica (*Parametric* SeLU), para $\alpha = 0{,}01$.

Esta função de ativação também elimina a morte do neurônio para valores negativos de NET, e suaviza o impacto das correções negativas em relação às correções positivas. Apenas para efeito de comparação, no caso da declividade β igual a 1,0, o ângulo vale 45°, considerando que a declividade é

definida matematicamente como sendo a tangente do ângulo que a reta faz com o semieixo horizontal positivo.

O gráfico da função de ativação SeLU com Vazamento é mostrado na Figura 2.11.

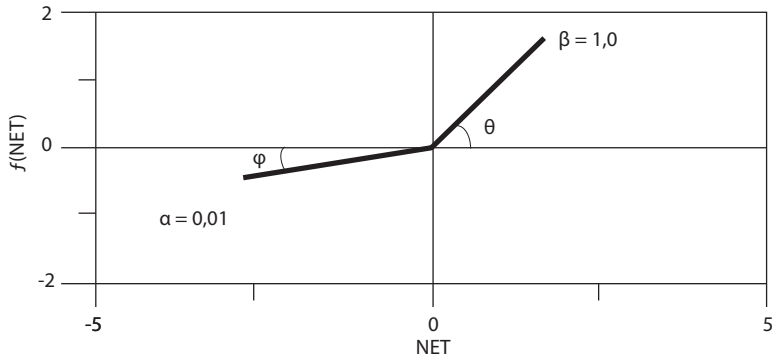


Figura 2.11 – Função de ativação SeLU com Vazamento.

Aqui, também, a não linearidade é introduzida pela existência de um ponto de quebra no gráfico da função. Sua derivada primeira é

$$f'(NET) = \begin{cases} 0{,}01, & para\ NET \leq 0 \\ \beta, & para\ NET > 0 \end{cases} \quad \textbf{(2.6)}$$

2.4.2.4 *Unidade Linear Exponencial (*Exponential Linear Unit – *ELU)*

Define-se como Unidade Linear Exponencial (em inglês *Exponential Linear Unit* – ELU) a função não linear representada pela equação

$$f(NET) = \begin{cases} \alpha(e^{NET} - 1), & para\ NET < 0 \\ \beta NET, & para\ NET \geq 0 \end{cases} \quad \textbf{(2.7)}$$

em que *NET* é o valor de entrada, α é um fator de redução ($\alpha < 1$) ou de ampliação ($\alpha > 1$) fazendo a parte exponencial se aproximar ou se afastar do eixo horizontal, respectivamente, e β é a declividade da reta situada no primeiro quadrante. O gráfico da função de ativação ELU é mostrado na Figura 2.12.

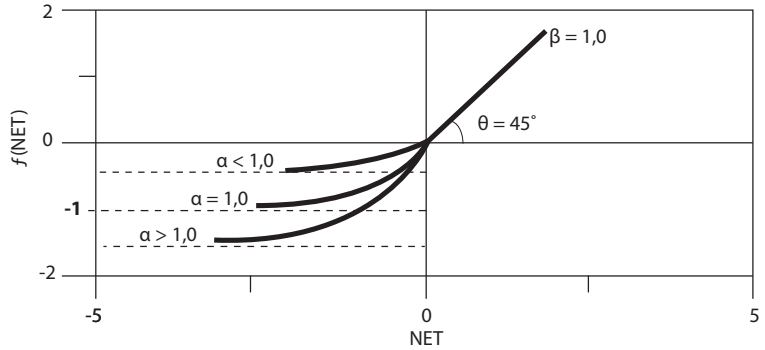


Figura 2.12 – Função de Ativação Linear Exponencial.

Diferentemente das anteriores, a não linearidade é introduzida pela função exponencial concordante com a função Linear. Sua derivada primeira é

$$f'(NET) = \begin{cases} \alpha e^{NET}, & para\ NET < 0 \\ \beta, & para\ NET \geq 0 \end{cases} \quad \textbf{(2.8)}$$

Como já foi mencionado antes, a declividade β igual a 1,0 corresponde ao ângulo de 45°, considerando que a declividade é definida como a tangente do ângulo que a reta faz com o semieixo horizontal positivo.

A função ELU possui a desvantagem de poder implicar na extinção prematura de gradientes (*vanishing gradient*), a qual está relacionada ao zeramento das derivadas existentes no método da descida do gradiente, isso durante a fase de retropagação do erro (*error backpropagation*) para sinais de erro muito negativos, devido à ocorrência de saturação.

As funções de ativação apresentadas aqui são as mais usadas na literatura, nada impedindo de se usar outras funções, desde que obedeçam à condição para ser uma função de ativação do neurônio, que é ser uma função contínua e derivável por partes.

2.4.2.5 Softmax

É uma função de ativação que não se enquadra na classe de funções matemáticas simples, como vem sendo a principal característica das novas FANs apresentadas até aqui.

O motivo de incluir a função Softmax é por ser utilizada nos LLMs baseados no *Transformer*, uma estrutura que implementa os mecanismos de

Atenção, conforme será visto mais adiante nos capítulos de IA generativa. Então, a ideia é apresentar logo a função Softmax, como parte do assunto relativo às novas funções de ativação, embora o seu propósito seja diferente das demais apresentadas até aqui. Neste caso, estamos priorizando os cálculos estatísticos, ao invés da velocidade de processamento!

Define-se como Softmax a função estatística não linear representada pela equação

$$f(NET_i) = \frac{e^{NET_i}}{\sum_{j=1}^{N} e^{NET_j}} \quad \textbf{(2.9)}$$

em que i, $j = 1,\ldots,N$. É denominada de estatística pelo fato de suas saídas serem expressas como sendo a probabilidade de uma saída ocorrer comparativamente às demais saídas restantes.

Exemplo: suponha que se tem os seguintes valores de entradas: NET_1 = 0,9, NET_2 = 1,5 e NET_3 = 2,1. Calcular os valores das respostas aplicando-se a função Softmax para as entradas fornecidas.

Solução: primeiro vamos calcular os valores de e^{NET_i}, fazendo i = 1, 2, 3, que resulta em:

$e^{NET_1} = e^{0,9} = 2,46$ $\quad e^{NET_2} = e^{1,5} = 4,48$ $\quad e^{NET_3} = e^{2,1} = 8,17$

Agora, basta aplicar a fórmula (2.9) da função Softmax, obtendo:

$$f(NET_1) = \frac{e^{NET_1}}{\sum_{j=1}^{3} e^{NET_j}} = \frac{2,46}{2,46 + 4,48 + 8,17} = \frac{2,46}{15,11} = 0,16$$

$$f(NET_2) = \frac{e^{NET_2}}{\sum_{j=1}^{3} e^{NET_j}} = \frac{4,48}{2,46 + 4,48 + 8,17} = \frac{4,48}{15,11} = 0,30$$

$$f(NET_3) = \frac{e^{NET_3}}{\sum_{j=1}^{3} e^{NET_j}} = \frac{8,17}{2,46 + 4,48 + 8,17} = \frac{8,17}{15,11} = 0,54$$

Nota-se que todas as respostas se encontram entre 0 e 1, cuja soma vale 1.

A principal característica da função Softmax é que as suas respostas fazem parte de uma distribuição de probabilidades, pelo fato de satisfazer as seguintes condições: I) valores das respostas compreendidos no intervalo fechado entre 0 e 1; II) a soma das respostas vale 1.

Portanto, o valor de cada resposta corresponde a quanto cada uma delas representa a probabilidade de ocorrência em relação às demais. No exemplo

INTELIGÊNCIA ARTIFICIAL NEURAL E GENERATIVA

REDE NEURAL ■ *BACKPROPAGATION* ■ IA GENERATIVA ■ *TRANSFORMER* ■ MECANISMOS DE ATENÇÃO

Este livro proporciona uma compreensão ampla das técnicas que regem o funcionamento do *Transformer*, cérebro dos modelos grandes de linguagem (*large language models* – LLMs), construído basicamente usando as redes neurais artificiais. Seus detalhes operacionais ficaram "escondidos" na publicação do famoso artigo *Attention Is All You Need* (2017), escrito por cientistas do Google Brain, principalmente com relação aos aspectos funcionais internos dos mecanismos de Atenção (*Attention*) e Atenção Multicabeças (*Multi-head Attention*). A obra é destinada a todos os interessados em conhecer detalhes do funcionamento da IA Neural e Generativa, principalmente os profissionais de TI, ciências exatas e engenharias.

O que você vai encontrar:

- Redes neurais e *Backpropagation.*
- IA Generativa e o *Transformer*: codificador e decodificador.
- Mecanismos de Atenção e Atenção Multicabeças, incluindo ilustrações e exemplos numéricos sobre o seu funcionamento, para facilitar a compreensão por parte do leitor.

Este é o segundo livro do autor sobre IA. No primeiro, ele aborda de forma magistral a IA simbólica e conexionista, as RNAs, entre outros temas. Nesta obra, apresenta, de forma didática e precisa, contribuições significativas para a compreensão do funcionamento da IA Neural e Generativa.

André Marcorin de Oliveira, D.Sc.

OSCAR GABRIEL FILHO

Engenheiro, pesquisador e escritor nas áreas de IA e Sistemas de Controle de Processos. Graduado pela Universidade Federal de Juiz de Fora (UFJF), mestre e doutor em Engenharia Elétrica pela Universidade Federal do Rio Grande do Norte (UFRN). Vencedor do prêmio David A. Wilson 2009, *Excellence in Teaching and Learning*, da Laureate International Universities, EUA, com o trabalho *Adaptive e-Learning System*, baseado em técnicas de Inteligência Artificial (lógica *fuzzy* e redes neurais artificiais). Finalista do prêmio Jabuti Acadêmico 2024, na categoria Ciência da Computação, com o livro *Inteligência artificial e aprendizagem de máquina: aspectos teóricos e aplicações* (Blucher, 2023).

Saiba mais sobre o livro

www.blucher.com.br

Blucher

ISBN 978-85-212-2974-2