

INTELIGÊNCIA ARTIFICIAL: FUNÇÕES DE ATIVAÇÃO

ARTIFICIAL INTELLIGENCE: ACTIVATION FUNCTIONS

51

Igor Vilela Rizzo¹, Robson Leandro Carvalho Canato²

1- Formado em Ciência da Computação pela Faculdade Municipal Prof. Franco Montoro – Mogi Guaçu (FMPFM/Mogi Guaçu); 2- Coordenador e docente do curso de Ciência da Computação da FMPFM/Mogi Guaçu, Mestre em Ciência da Computação pela Universidade Estadual de Campinas (UNICAMP).

Contatos: igorrizzo1@hotmail.com¹; canato.fmpfm@gmail.com²

RESUMO

Consideradas um elemento de extrema importância nas Redes Neurais Artificiais, as funções de ativação são responsáveis basicamente por decidir se um neurônio será ou não ativado, de acordo com a relevância da informação recebida por esse neurônio. Com base nesta afirmação, este trabalho propõe apresentar uma avaliação sobre a performance e eficiência das funções de ativação. As funções de ativação consideradas na avaliação são: Identidade, Sigmóide, ReLU e TanH mediante a operação lógica XOR. Tal avaliação contou com a utilização da plataforma Anaconda, linguagem de programação Python, biblioteca TensorFlow e do ambiente de desenvolvimento integrado Jupyter Notebook.

Palavras-chave: Inteligência Artificial, Aprendizado de máquina, Funções de Ativação.

ABSTRACT

Considered an extremely important element in Artificial Neural Networks, activation functions are primarily responsible for deciding whether or not a neuron will be activated, according to the relevance of the information received by that neuron. Based on this statement, this paper proposes to present an evaluation on the performance and efficiency of activation functions. The activation functions considered in the evaluation are: Identity, Sigmoid, ReLU and TanH through XOR logic operation. This evaluation included the use of the Anaconda platform, Python programming language, TensorFlow library and Jupyter Notebook integrated development environment.

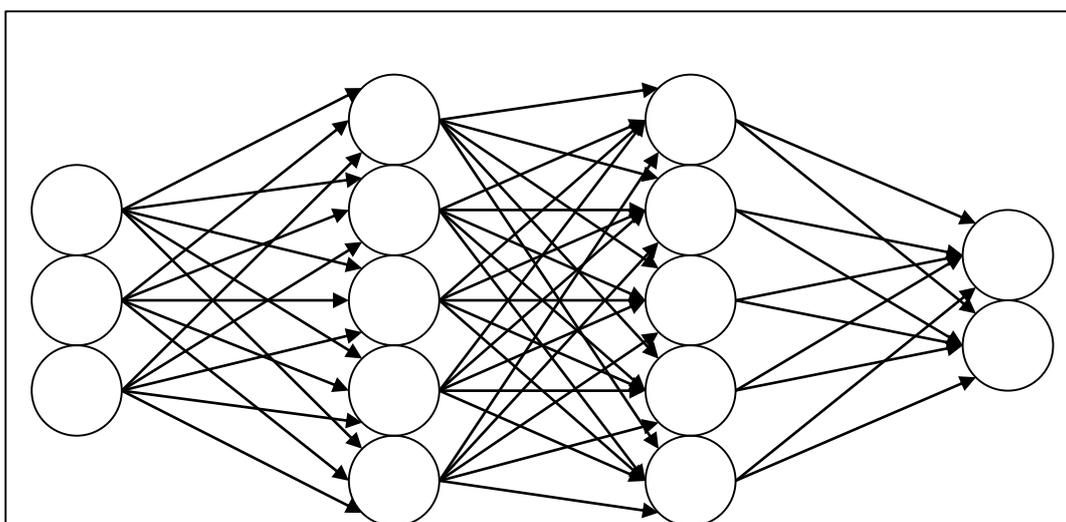
.Keywords: Artificial Intelligence, Machine learning, Activation Functions.

INTRODUÇÃO

Todo processo de aprendizado, seja ele qual for, tem seus mecanismos e fundamentos. Com a Inteligência Artificial (IA) o processo é o mesmo, o que se difere são suas ferramentas de aprendizado. Uma delas, são as que conhecemos como Funções de Ativação. Elas são uma das várias ferramentas usadas pela IA para transmitir e analisar os dados em sua rede neural. Tais fórmulas são as responsáveis pelo estímulo de seus neurônios e são eles que permitem a aprendizagem de IA's.

Na Figura abaixo é possível observar como é estruturada uma rede neural. No *input* da rede neural a IA obtém os dados a serem analisados. Nas fases seguintes, a soma é transformada usando a função de ativação específica a aquela situação. Conforme pode ser verificado a seguir, a Figura 1 ilustra um exemplo de uma Rede Neural Artificial de quatro camadas, três entradas, duas saídas e duas camadas ocultas, incluindo dez neurônios.

Figura 1. Exemplo de uma Rede Neural Artificial.



Fonte: elaborado pelos autores.

Todo problema apresentado a Inteligência Artificial é analisado e calculado de diferentes formas. Tudo depende da pergunta feita, ou seja, cada tipo de problema requer um tipo específico de função de ativação. Para entender na prática qual função a IA aprende de forma mais rápida e também treiná-la escolhi a operação lógica XOR. As possíveis entradas e saídas utilizando a lógica XOR podem ser verificadas na Tabela 1:

Tabela 1. Tabela de operação lógica XOR.

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

Fonte: (PORTAS, 2011).

O motivo de ter escolhido essa operação em específico é sua simplicidade tanto nos cálculos quanto para ser aprendida pelos que ainda estão iniciando no assunto. Baseado nesses dados, a minha hipótese é a de que dentre as quatro funções de ativação escolhidas para este artigo, a Sigmóide é a que obtém os melhores resultados, sendo a que proporciona um maior aprendizado a IA. Quanto a função identidade, que será analisada abaixo, penso que a IA não será capaz de passar por todas as épocas visto que a operação lógica XOR não é uma problemática linear.

Para tanto, este artigo tem como objetivo analisar as funções de ativação Identidade, Sigmóide, ReLU e Sigmóide a fim de descobrir qual função se adequa melhor ao problema apresentado a IA utilizando a operação lógica XOR.

MOTIVAÇÃO

Enquanto estudava sobre Inteligência Artificial me deparei com redes neurais e uma das curiosidades surgiu mediante a agilidade com que a IA lê, cruza tantas informações entre si, sendo capaz de chegar a um resultado rapidamente. O intrigante e que tanto me fez pensar a respeito, é que há problemas que a mente humana não é capaz de resolver tão rápido como a IA, e outros tantos que não é nem capaz de resolver.

Diante disso, passei a estudar sobre a aprendizagem de IA's e quando finalmente encontrei materiais que pudessem me dar um maior esclarecimento para essa questão, percebi que tudo que a IA é capaz de fazer é definido por fórmulas que determinam seu aprendizado. Tal processo só é possível através dessas funções. Atualmente há incontáveis fórmulas disponíveis a IA e não há restrição em quantidade de funções a ser usadas por um mesmo programa visto que cada uma delas tem uma finalidade.

Descobri que um dos fatores limitantes à eficiência da IA é a aplicabilidade da fórmula estruturada, que em alguns casos, são muito específicas. Tais equações são as ferramentas de aprendizagem da IA. Essas fórmulas são chamadas de funções de ativação e possuem esse nome pois os neurônios artificiais da IA são ativados sempre que uma determinada informação passa por eles.

Colher tais informações não foi uma tarefa tão simples, e durante esse processo pude perceber que há pouco material a respeito. Minha motivação para estudar esse assunto se dá pelo fato de que a IA é um universo complexo e os estudos a respeito em português são raros e com linguagem muito específica do meio científico. Devido a esse fato, decidi tornar esse o tema do meu presente trabalho, com o intuito de analisar algumas funções de ativação com uma linguagem mais fácil de ser compreendida pelos que estão iniciando no assunto.

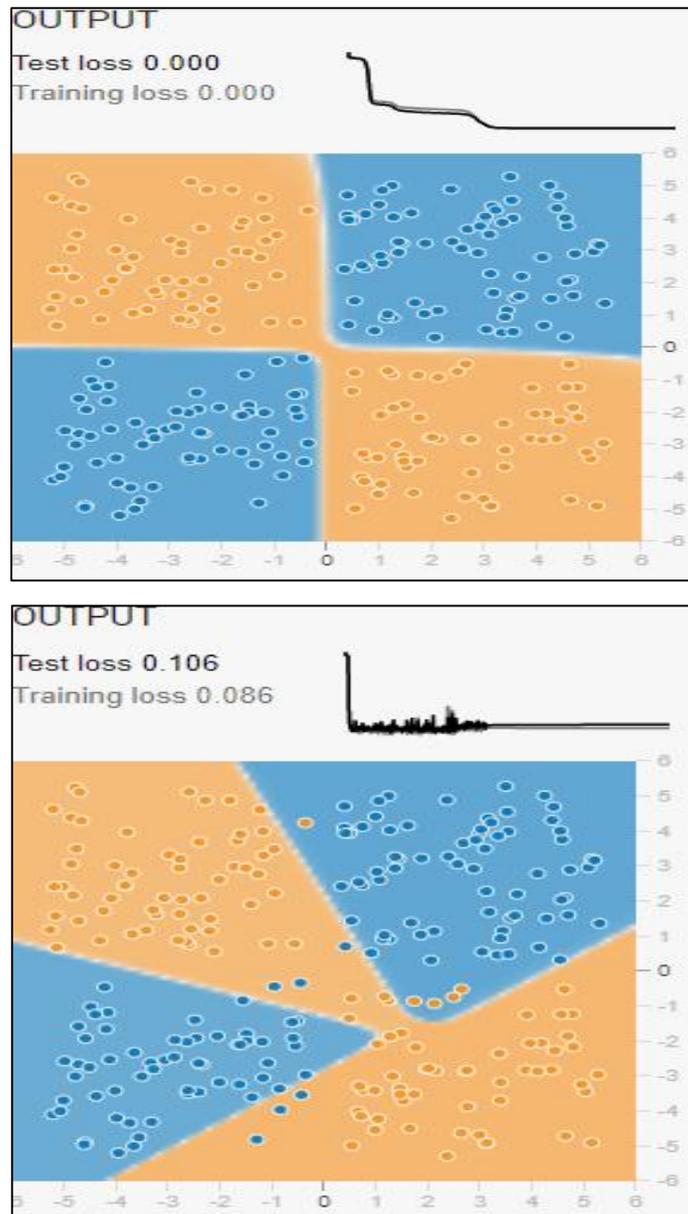
CONCEITOS FUNDAMENTAIS

Funções de ativação são as fórmulas usadas como ferramenta de aprendizado de IA's. Assim como todo processo possui seus métodos de análise de dados, a IA possui essas fórmulas para calcular as informações de entrada e chegar a um resultado correto. Entretanto, cada problema requer um cálculo específico, ou seja, das várias funções que existem umas podem ser eficientes a IA para um determinado cálculo e outras podem não chegar a nenhum resultado.

Uma forma simples e prática para perceber se a IA conseguiu ou não calcular os dados é por meio de gráficos. Através deles é mais fácil observar as entradas da rede neural, representadas por pontos, e então esperar a IA separar os dados de forma correta, ou não. Assim, é possível observar se ela conseguiu separar os dados corretamente ou se a função usada não era a melhor opção diante do problema apresentado.

Para exemplificar, usamos a Função Sigmóide e a TanH para separar círculos azuis dos alaranjados, ambas com a mesma quantidade de camadas, neurônios e tentativas. Na Figura 2 é possível perceber que a IA conseguiu chegar a um resultado correto com 0% de erro. Já na Figura 3 a IA não teve êxitos em separar os dados completamente.

Figura 2. Teste usando Função Sigmoide.



Fonte: Adaptado de (SMILKOV; CARTER, 2019).

Entretanto, para que esse processo aconteça é necessário que todos os dados passem pelo que chamamos de Redes Neurais Artificiais. Esse é um dos conceitos mais importantes que precisa ficar bem claro antes de adentrarmos nas análises e avaliações das funções de ativação.

Uma Rede Neural Artificial é um modelo de aprendizado de máquina que, dados determinados vetores de entrada e saída, tentará ajustar as saídas às entradas. Ou seja, dado um conjunto de instâncias observadas com certos valores que desejamos prever e alguns dados que temos em cada instância, ele tentará generalizar esses dados a fim de prever os valores corretamente para novas instâncias do problema.

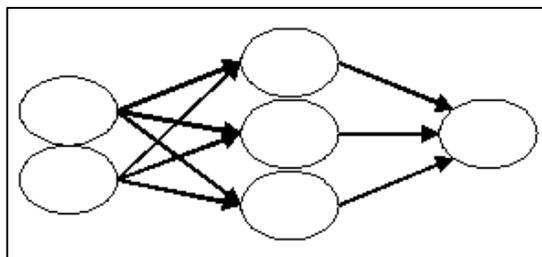
A flexibilidade potencial de uma Rede Neural Artificial está proporcionalmente relacionada à quantidade de camadas ocultas e seus tamanhos. Isso quer dizer que quanto mais camadas ocultas ela tiver e quanto mais neurônios possuir, maior será sua capacidade de aprendizagem e maiores serão as chances de separar os dados corretamente.

DESENVOLVIMENTO

Embora o objetivo deste artigo seja analisar funções de ativação, não iremos abordar todas, pois existem quase trinta funções. Para este estudo, escolhemos quatro funções de ativação: Identidade, Sigmóide, ReLU e TanH.

Antes de iniciar os testes com cada uma das funções tivemos que estabelecer alguns parâmetros base para a IA. Primeiramente carregamos uma matriz x_1 e x_2 com os dados de entrada segundo a operação lógica XOR, em seguida inserimos também as respostas em outra matriz. O segundo passo foi criar uma Rede Neural Artificial com dois neurônios de entrada, uma camada oculta com três neurônios e um neurônio de saída. Para os testes desenvolvidos ao longo deste trabalho utilizaremos como modelo a Rede Neural Artificial esboçada pela Figura 4:

Figura 4. Modelo da Rede Neural Artificial usada nos testes



Fonte: elaborado pelos autores.

Com a primeira parte da programação concluída, prossegui para a definição dos *placeholders*, camada oculta, camada de saída e dos neurônios de ativações. Em seguida ajustei a taxa de aprendizado da IA para 0,3. Também definimos um milhão de épocas. Dessa forma a IA consegue extrair informações

mais claras utilizando todas as funções, levando em consideração a diferença no tempo de execução de cada uma. Isso significa que devido a diferença no tempo de execução de cada função, ter uma grande quantidade de épocas ajuda a ter um melhor contraste entre as funções ao se comparar as taxas de aprendizagem, o tempo e as margens de erro. Com esses dados ajustados, definimos também o otimizador, taxa de erro e o timer para cronometrar o tempo que a IA leva executando as funções. Com a IA devidamente configurada podemos começar a treiná-la.

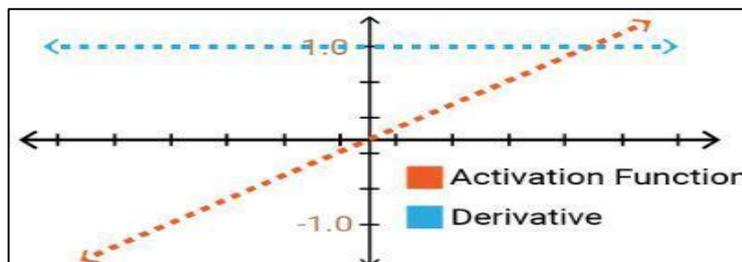
Função Identidade

É a função mais simples, porém a mais específica também. Sua eficiência só é possível com cálculos exponenciais. Dishashree Gupta (2017) relata que sempre que tentarmos fazer uma propagação de retorno obteremos o mesmo gradiente e dessa forma não estaremos melhorando o erro.

Segundo Gharat (2019): “A função identidade pega as entradas, multiplicadas pelos pesos de cada neurônio e cria um sinal de saída proporcional à entrada.” Ou seja, imagine que a entrada seja 3 e o peso dos neurônios da camada oculta seja 4. Então na função $F(x) = 4x$ o resultado será 12 e crescerá exponencialmente com múltiplos de 4 sempre que passar pelo mesmo neurônio outra vez. A representação gráfica para esta função de ativação pode ser verificada na Figura 5. Abaixo os dados da função identidade:

- **Equação:** $f(x) = x$
- **Derivada:** $f'(x) = 1$
- **Intervalo:** $(-\infty, +\infty)$

Figura 5. Representação gráfica da função identidade.



Fonte: (GHARAT, 2019).

Avaliação da Função Identidade

Não foi possível testar a função identidade na IA porque a operação lógica de XOR necessita de uma função não linear para calcular pois o seu resultante varia entre 0 e 1.

58

Vantagens e Desvantagens

Essa função é muito prática quando a resolução do problema envolve regressão linear. Não há outra fórmula tão simples quanto esta para esse tipo de cálculo. Entretanto, diante das principais problemáticas a função identidade não está entre as mais eficazes devido a sua especificidade.

Função Sigmóide

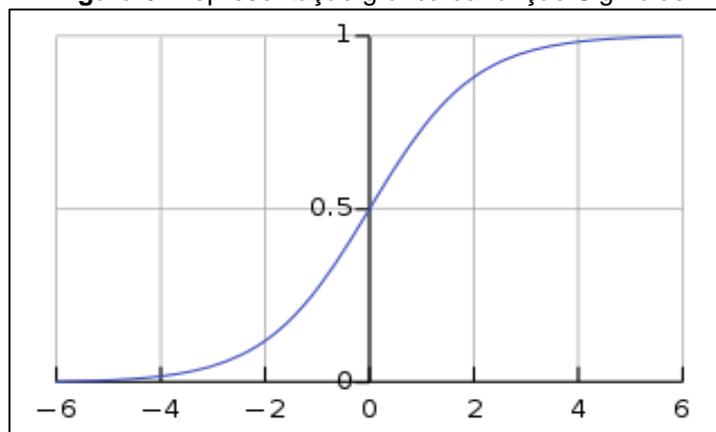
Gharat (2019) caracteriza a Sigmóide como: “uma curva em ‘S’ na representação gráfica. Para mapear valores previstos para probabilidades, usamos a função Sigmóide. Ela mapeia qualquer valor real em outro valor entre 0 e 1.” A Figura 6 ilustra a representação gráfica desta função de ativação. Abaixo os dados da função Sigmóide:

→ **Equação:** $f(x) = s = 1 / (1 + e^{-x})$

→ **Derivada:** $f'(x) = s * (1-s)$

→ **Intervalo:** (0,1)

Figura 6. Representação gráfica da função Sigmóide.



Fonte: (GHARAT, 2019).

Avaliação da função sigmoide

No teste com a Sigmóide a IA conseguiu obter bons resultados em poucas tentativas. Essa função se aplica muito bem com a operação lógica XOR visto que ela trabalha entre 0 e 1 e a resposta do operador lógico também é 0 e 1. A margem de erro da IA foi diminuindo cada vez mais de maneira rápida e contínua levando 466 segundos e 78 milissegundos para passar por todas as épocas. Os resultados utilizando a função de ativação Sigmóide são apresentados na Tabela 2 resultados da função Sigmóide, a seguir:

59

Tabela 2. Resultados da função Sigmóide.

X1	X2	XOR	Resultado do teste
0	0	0	0.00353575
0	1	1	0.9991415
1	0	1	0.99552155
1	1	0	0.00351021

Fonte: elaborado pelos autores.

Vantagens e desvantagem

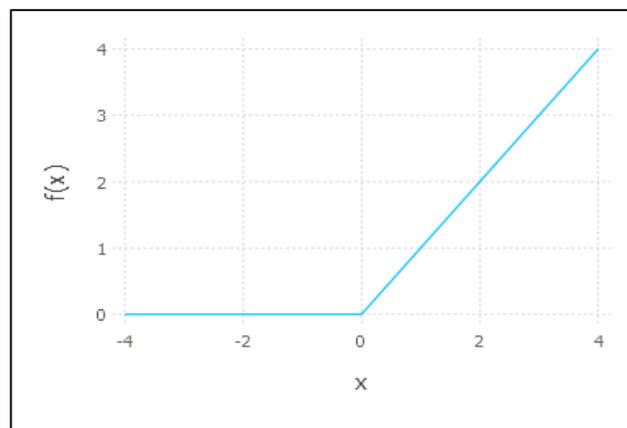
De acordo com Gharat (2019), a Sigmóide não é tão eficiente em todos os cálculos. Se os valores são muito distantes de zero - muito altos ou muito baixos - não há quase nenhuma alteração na previsão e isso pode causar um problema de gradiente de fuga. Nessas circunstâncias os Sigmóides têm convergência lenta. Entretanto, se os valores forem próximos de zero a Sigmóide é a função que consegue os resultados com a menor margem de erro e em menos tempo quando comparada a outras funções.

Função ReLU (Unidade Linear Retificada)

Essa função é inspirada nos neurônios do cérebro que retornam a um valor positivo ou a 0. Atualmente, a ReLU é a função de ativação mais usada no mundo. Em redes neurais convolucionais ou para aprendizado profundo é a mais eficiente. A representação gráfica para a função ReLU é esboçada através da Figura 7. Abaixo os dados da função ReLU:

- **Equação:** $f(x) = a = \max(0, x)$
- **Derivada:** $f'(x) = \{1; \text{ se } z > 0, 0; \text{ se } z < 0 \text{ e indefinido se } z = 0\}$
- **Intervalo:** $(0, +\infty)$

Figura 7. Representação gráfica da função ReLU.



Fonte: (GHARAT, 2019).

Avaliação da função ReLU

No teste com a ReLU a IA não conseguiu obter bons resultados em nenhuma época. A função se manteve constante com a taxa de erro em 0,125. Em um milhão de épocas ela repetiu a mesma margem de erro de forma que não estava evoluindo seu aprendizado. O tempo total para passar um milhão de épocas foi de 529 segundos e 51 milissegundos, obtendo um número muito distante do correto. A Tabela 3 apresenta os resultados utilizando a função de ativação ReLU.

Tabela 3. Resultados da função ReLU.

X1	X2	XOR	Resultado do teste
0	0	0	0,613202
0	1	1	0,6139185
1	0	1	0,5932282
1	1	0	0,5726325

Fonte: elaborado pelos autores.

Vantagens e Desvantagens

Para Gharat (2019) a ReLU tem a vantagem computacional de convergir rapidamente entre as redes neurais. No entanto, por mais que pareça que ela seja uma função linear, não é, pois tem uma função derivada que permite a propagação traseira. Entretanto, quando as entradas se aproximam de zero ou são negativas, o gradiente da função se torna zero impedindo a rede de executar a propagação de retorno.

61

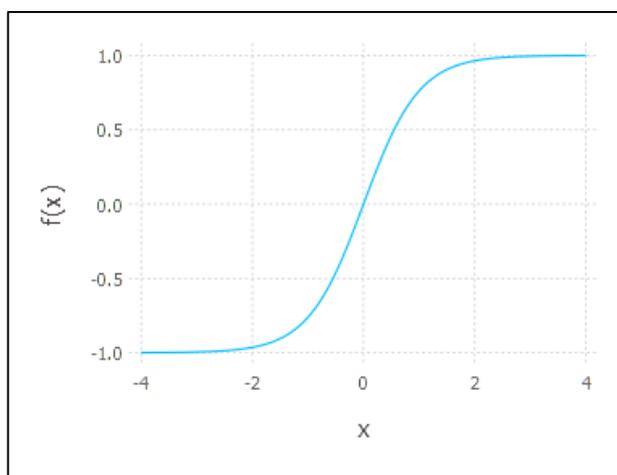
Função TanH (Tangente Hiperbólica)

Facure (2017) caracteriza a TanH como: “similar a função Sigmóide, a função TanH também tem um formato de ‘S’, mas varia de -1 a 1, em vez de 0 a 1 como na Sigmóide.” Em sua opinião se assemelha também à identidade, sendo assim uma alternativa mais atraente do que a Sigmóide para ativar as camadas ocultas. A Figura 8 é uma representação gráfica da função de ativação TanH.

Abaixo os dados da função TanH:

- **Equação:** $f(x) = a = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$
- **Derivada:** $(1 - a^2)$
- **Intervalo:** $(-1, 1)$

Figura 8. Representação gráfica da função TanH.



Fonte: (GHARAT, 2019)

Avaliação da função TanH

No teste com a TanH a IA não conseguiu obter bons resultados embora tenha tido um aprendizado constante em todas as épocas. Em comparação com a Sigmóide - de onde se origina sua fórmula - ela teve um resultado bom, porém pior. Isso já era esperado porque mesmo que sua função tenha se originado com a Sigmóide a TanH tem o diferencial do intervalo entre -1 e 1. A TanH demorou 439 segundos e 4 milissegundos para passar por todas as épocas com a taxa de erro diminuindo cada vez mais. Os resultados obtidos utilizando a função TanH podem ser verificadas na Tabela 4.

62

Tabela 4. Resultados da função TanH.

X1	X2	XOR	Resultado do teste
0	0	0	0.4874975
0	1	1	0.72930384
1	0	1	0.70005774
1	1	0	0.54866654

Fonte: elaborado pelos autores.

Vantagens e Desvantagens

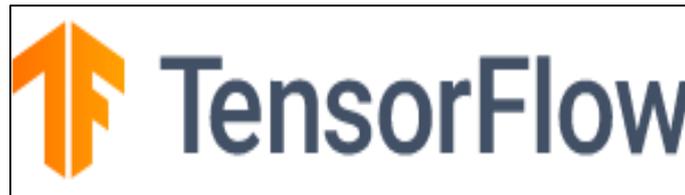
Para Snehal Gharat (2019) a TanH se destaca em problemas de separação com valores neutros, fortemente positivos e fortemente negativos. Segundo o autor, “a função e sua derivada são monotônicas (preservam a relação de ordem) e também sofrem um problema de gradiente de fuga e, portanto, possuem lenta convergência” (GHARAT, 2019).

FERRAMENTAS

Enquanto buscava as melhores ferramentas para estudar o método de aprendizagem de IA's me deparei com a biblioteca TensorFlow. Após uma análise cuidadosa, percebi que ela é uma das mais aclamadas no estudo de IA's. Devido

a isso decidi que ela seria meu referencial neste estudo, bem como suas ferramentas para estudar IA's.

Figura 9. Logo da biblioteca TensorFlow.



Fonte: (TENSORFLOW, 2019)

Para interpretar a biblioteca do TensorFlow e facilitar meus estudos escolhi o IDE (Ambiente de Desenvolvimento Integrado) Jupyter Notebook. Essa escolha se deu pelo fato de ser o IDE utilizado pela Google - proprietária da biblioteca TensorFlow.

Figura 10. Logo da IDE Jupyter.



Fonte:(JUPYTER, 2019).

Para o gerenciamento de ambientes, pacotes e bibliotecas foi utilizada a plataforma Anaconda. Ela é responsável por reunir compiladores e IDE's para cientistas de dados que utilizam Python e R como linguagem de programação. Para este estudo escolhi a Python por ser a mais usada pelos cientistas de dados, entusiastas no aprendizado de máquina e/ou aprendizado profundo.

Figura 11. Logo da plataforma Anaconda.



Fonte: (ANACONDA, 2019)

CONSIDERAÇÕES FINAIS

Diante dos resultados obtidos testando cada uma das funções, pude comprovar a minha hipótese. A função Identidade se provou incompatível com a operação lógica XOR por ser uma fórmula linear. Quanto a ReLU e TanH observei que são compatíveis com a XOR pois são funções não lineares. A ReLU se mostrou ineficiente porque quando o valor de entrada foi igual a 0 a função não conseguiu executar a propagação de retorno.

A TanH conseguiu demonstrar avanços na aprendizagem da IA e apresentou resultados que tendem a resposta correta. Talvez com muito mais épocas ela seria capaz de obter números mais aproximados, mas isso é inviável computacionalmente. Como havia previsto, a Sigmóide foi a função que melhor se adaptou a operação lógica XOR, proporcionando a IA melhores valores de peso entre seus neurônios.

REFERÊNCIAS

ANACONDA. **Solutions for Data Science Practitioners and Enterprise Machine Learning**. Disponível em:<<https://www.anaconda.com/>>. Acesso em: 14 de novembro de 2019.

Data Science Academy. **Deep Learning Book**, 2019. Disponível em:<<http://deeplearningbook.com.br/funcao-de-ativacao/>>. Acesso em: 10 de novembro de 2019.

FACURE, M. **Funções de Ativação**. 2017. Disponível em:<<https://matheusfacure.github.io/2017/07/12/activ-func/>>. Acesso em: 13 de novembro de 2019.

GHARAT, S. **O quê, por que e qual ?? Funções de Ativação**. 2019. Disponível em:<<https://medium.com/@snaily16/what-why-and-which-activation-functions-b2bf748c0441>>. Acesso em: 13 de novembro de 2019.

GUPTA, D. **Fundamentos da aprendizagem profunda - funções de ativação e quando usá-las?** 2017. Disponível em:<<https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/>>. Acesso em: 13 de novembro de 2019.

JUPYTER. **Project Jupyter**. Disponível em:<<https://jupyter.org/index.html>>. Acesso em: 14 de novembro de 2019.

PORTAS lógicas. In: **WIFI-PROMPT**. Blogspot.com. 2011. Disponível em:<<http://wifi-prompt.blogspot.com/p/portas-logicas-and-or-xor-not-nand-nor.html>>. Acesso em: 5 de novembro de 2019.

SMILKOV, D.; CARTER, S. **TensorFlow**. 2019. Disponível em:<<https://playground.tensorflow.org/>>. Acesso em: 13 de novembro de 2019.

TENSORFLOW. Disponível em:<<https://www.tensorflow.org/>>. Acesso em: 14 de novembro de 2019.

Os autores declararam não haver qualquer potencial conflito de interesses referente a este artigo.