

Inteligência Computacional

Rede neural artificial para identificação de caracteres

Gabriel Tambara Rabelo
Departamento de engenharia elétrica
Faculdade de Tecnologia - UnB
Brasília, Brasil
gtambarab@gmail.com

Abstract—Artificial neural networks are used for various functions. One of these, the ability to identify patterns in characters, is explored in this document, addressing issues such as accuracy, functions used and noise resistance. The results obtained in the laboratory corroborate its utility and practicality as well as the design method and demonstrate ways to circumvent the problems commented.

Keywords: identification; characters; intelligence; artificial; neural; network.

Resumo—Redes neurais artificiais são utilizadas para diversas funções. Uma dessas, a capacidade de identificação de padrões em caracteres, é explorada neste documento, abordando questões como acurácia, funções utilizadas e resistência à ruído. Os resultados obtidos em laboratório corroboram a sua utilidade e praticidade bem como demonstram o método de projeto e formas de se contornar os problemas comentados.

Palavras-chave: identificação; caracteres; inteligência; artificial; rede; neural.

I. INTRODUÇÃO

No contexto atual, diante do cenário da indústria 4.0, e portanto da conectividade de diversos dispositivos diferentes, percebe-se a necessidade de constante inovação e adaptação dos ambientes físicos e virtuais, de forma a melhorar e otimizar ferramentas para resolver problemas otimamente fazendo uso da grande quantidade de dados disponíveis. Nesse cenário, a rede neural artificial é uma ferramenta preponderante no quesito crescimento em suas aplicações, utilizando de um sistema semelhante ao sistema neural animal, de modo a criar aprendizado análogo ao humano, por exemplo.

Diante desta ferramenta, o presente artigo detalha um estudo realizado visando esclarecer os fundamentos e questões pertinentes quanto ao uso de redes neurais artificiais, em específico, com uma aplicação voltada para identificação de alguns caracteres alfanuméricos simples específicos, à vista de praticidade e recursos limitados.

II. FUNDAMENTAÇÃO TEÓRICA

A. Rede Neural Artificial

Uma rede neural, baseada no modelo inicial do Perceptron de Rosenblatt, é dita como uma ferramenta que visa modelar

o funcionamento do cérebro na resolução de problemas específicos, podendo ser interpretada como uma identificadora de padrões. Isso é possível através da melhora iterativa de seus resultados, que faz uso de pequenas unidades de processamento ditas "neurônios", que se relacionam entre si por pesos. A calibração desses pesos determina a acurácia da ferramenta. Ela coleta dados do exterior e realiza o aprendizado dessas informações realizando processamentos sinápticos, ou interneurais.

Para que uma rede neural seja implementada no ambiente virtual de um computador, é necessária a modelagem matemática dos neurônios, descrito pela Fig.[3]. O modelo abordado, perceptron multicamadas, trabalha com pesos em cada um dos neurônios interconectados. Percebe-se o comportamento da soma ponderada dos sinais de entrada x com os pesos w , somada a um viés, ou *bias*, que, ao serem inseridos em uma função dita função de ativação, geram o resultado y . Essa função de ativação se responsabiliza por filtrar e limitar o sinal dentro de um intervalo discreto e fechado. A função de ativação mais comum é a *sigmoid* pelo seu caráter bioinspirado que permite soluções de sistemas não linearmente separáveis, ou seja, sistemas mais complexos e reais.

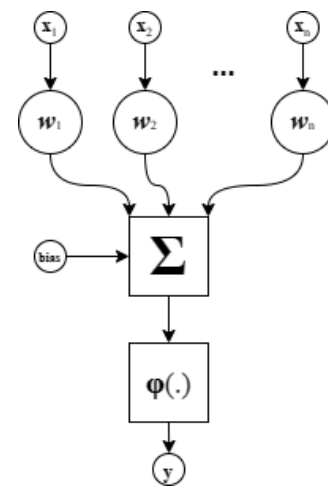


Figura 1: Modelo de neurônio

Para garantir uma mínima solução viável, redes neurais precisam ter ao menos uma camada de entrada, e uma camada de saída. Contudo, redes para obterem resultados factíveis com bons recursos de memória e processamento, fazem uso de diversas camadas ditas camadas escondidas, que residem entre as outras duas camadas. Quanto mais camadas, mais a rede é capaz de extrair estatísticas de maior ordem dos dados. Na prática, a saída de uma camada de neurônios é a entrada de uma nova, sendo que cada neurônio recebe todas as saídas dos neurônios da camada anterior.

Para que o sistema seja treinado, é necessário que existam chutes iniciais para os vetores de bias e pesos, de modo que conforme cada iteração dos processos, seja realizada uma verificação como se segue;

$$w(n+1) = w(n) \Leftrightarrow w^T x > 0 \quad (1)$$

$$w(n+1) = w(n) \Leftrightarrow w^T x \leq 0 \quad (2)$$

Essas igualdades são válidas para o cenário em que o vetor de entradas x é adequadamente identificado. Caso contrário, é necessário que sejam feitas correções nos pesos, como se segue, respectivamente para cada cenário:

$$w(n+1) = w(n) - \eta x \Leftrightarrow w^T x > 0 \quad (3)$$

$$w(n+1) = w(n) + \eta x \Leftrightarrow w^T x \leq 0 \quad (4)$$

Sendo η o parâmetro de aprendizado. Estes ajustes ocorrem em duas etapas. Na primeira, na propagação direta, os valores de entrada iniciais percorrem no sentido positivo da rede até gerar um sinal na saída, onde é calculado um erro que, durante a segunda etapa, a propagação reversa, é utilizado para corrigir os parâmetros conforme o modelo definido.

Este modelo é um exemplo simples e funcional de otimização dos pesos da rede, contudo, métodos mais avançados, baseados em SGD ou *Stochastic gradient descent*, ou descida do gradiente estocástico, costumam realizar a correção com o parâmetro de aprendizado multiplicado pelo oposto do vetor gradiente de alguma função, podendo ser a composição da soma ponderada com o viés multiplicado por um parâmetro de interesse relacionados à imperfeições no sistema ou medidas de aleatoriedade.

B. ADAM: Adaptive Moment Estimation

Adam é um método para otimização estocástica capaz de consumir pouca memória e que utiliza apenas gradientes de primeira ordem. O método vai computar taxas adaptativas individuais para diferentes parâmetros das estimativas dos primeiro e segundo momentos dos gradientes.

Definem-se β_1 e β_2 como os parâmetros de decaimento exponencial, ou hiper-parâmetros, para as estimativas dos momentos. Define-se uma função objetiva $f(\theta)$, capaz de calcular máximos e mínimos estocásticos que seja diferenciável, de parâmetros θ .

O algoritmo utilizará da média móvel exponencial dos momentos, os atualizando com os hiper-parâmetros, que atuarão como estimativas. Por fim, é realizada uma correção de viés, já que os momentos são inicializados com vetores que enviesam o resultado. O procedimento se repete até que os parâmetros θ converjam.

C. Situação problema

Para testar a viabilidade do método da rede neural artificial, propõe-se uma sequência de 16 caracteres alfanuméricos, a serem representados por um vetor binário. Estes vetores terão seus padrões aprendidos pela inteligência computacional e então será feita uma verificação de sua capacidade de reconhecimento de caracteres, desde iguais até bem diferentes com base em ruído. Então, a rede será treinada com caracteres com taxas progressivas de ruído, e os resultados do reconhecimento serão comparados.

A fins de ilustração, seguem os 16 caracteres a serem ensinados para a rede, bem como alguns exemplos de caracteres com ruído.

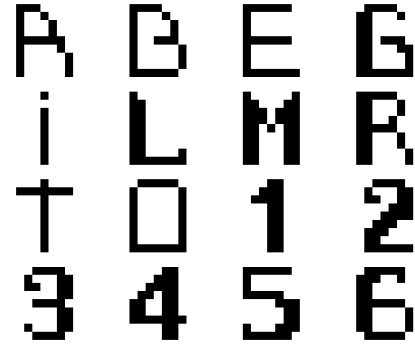


Figura 2: Caracteres alfanuméricos ensinados

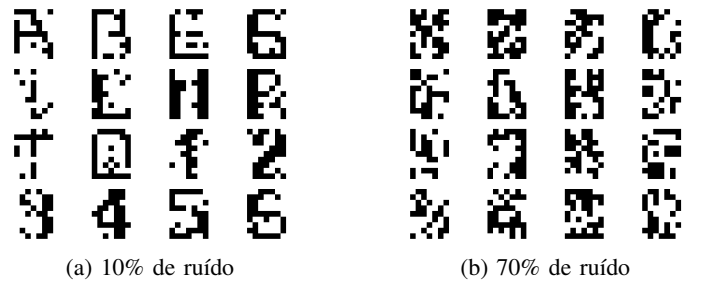


Figura 3: Caracteres alfanuméricos com ruído

III. LABORATÓRIO

Para realizar a avaliação da rede neural artificial, foi utilizado um programa em Python, com as bibliotecas numpy, matplotlib, keras e tensorflow, objetivando facilitar e permitir os desenvolvimentos estatísticos e de aprendizado de máquina em ambientes já bem desenvolvidos e consolidados na comunidade Python.

Criou-se um programa com uma topologia de rede neural artificial sequencial, com 5 camadas cuja função de ativação designada é a sigmoide, e com inicialização de vetor de bias e do kernel como identidades. O número de neurônios em cada camada e sua designação geral é apresentada na Fig.[4].

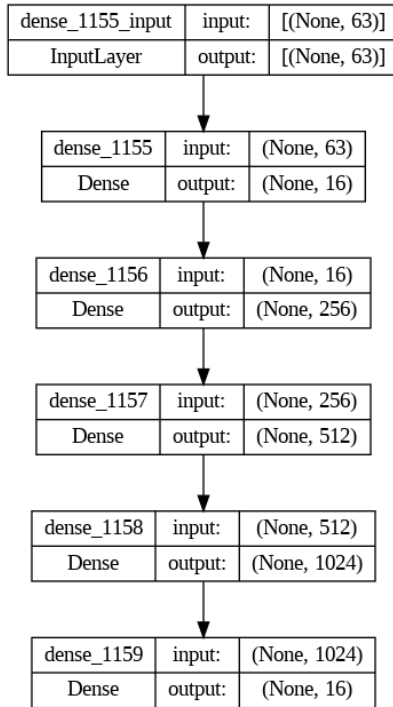


Figura 4: Topologia da RNA

O projeto desta topologia se deu de forma iterativa, com experimentos em pequenas iniciais mudanças em parâmetros de treinamento, como também em mudanças na quantidade de camadas e número de neurônios em cada camada da rede. O ciclo de teste e validação eventualmente resultou no modelo proposto que obteve uma acurácia de 100% na identificação de caracteres ideais dentro de metade do seu tempo de treinamento.

Diante dessa topologia, a rede neural passou por uma sequência de aprendizado definida por 1000 épocas de 2048 amostras dos caracteres ideais sendo processadas, ou *batch size*. O otimizador utilizado foi o ADAM cujo parâmetro escolhido foi a acurácia. Em seguida, definiu-se uma quantidade de 70 níveis diferentes de ruído em que para cada nível houve uma sequência nova de aprendizado a fim de ensinar à rede como reconhecer os caracteres com ruído. Para cada uma dessas sequências, houveram 40 iterações de aprendizado com ruído de 60 épocas e *batch size* de 256, seguido por um reforço ao fim de aprendizado dos caracteres ideais de 80 épocas e 256 de *batch size*.

O aprendizado totalizou 678,672 parâmetros, todos treinados, e cujos resultados foram validados ao se comparar a capacidade de reconhecimento da rede neural sem o treinamento para ruído com a com o treinamento para ruído. O resultado

pode ser observado pela Fig.[5], e conclui-se imediatamente que ambas as redes foram capazes de reconhecer com acurácia de 100% os caracteres ideais, e a rede ensinada com ruído foi capaz de melhorar o erro de 90% para 60% no pior dos casos, cujo ruído chegou a 70%.

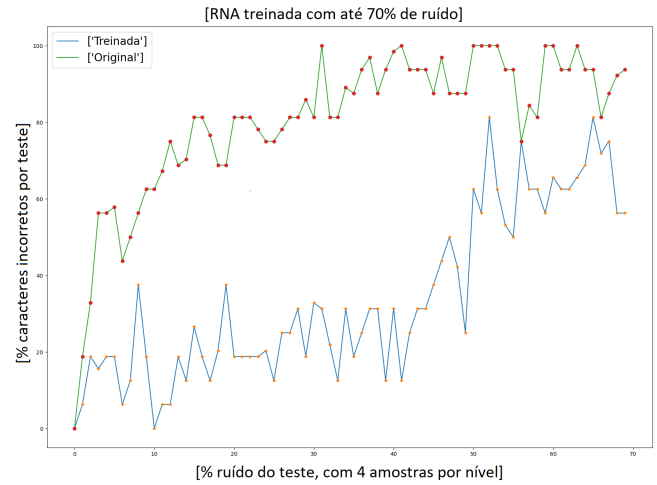


Figura 5: Análise comparativa de acurácia na identificação das redes

IV. CONCLUSÃO

As redes neurais artificiais possuem diversas aplicações e umas delas é o reconhecimento de padrões em caracteres alfanuméricos. Essa capacidade foi posta à prova e gerou resultados positivos pertinentes ao objetivo inicial, como registrado no presente documento, sendo capaz de reduzir em 30% o erro de estimação para o pior dos casos de ruído nas amostras. Foram também elucidadas questões pertinentes à resistência ao ruído e interferências para esse tipo de rede, o que pôde ser contornado com uma rotina especial de aprendizado, também detalhada, responsável pela redução de erro abordada. Ademais, reafirma-se o uso da rede modelo para problemas semelhantes e recomenda-se a sua utilização sempre que dispor-se-á dos recursos computacionais necessários para se realizar o treinamento.

Para fins de trabalhos futuros no tópico, recomenda-se uma exploração maior das topologias, visando reduzir o número de neurônios, ajudando também a reduzir o espaço na memória necessário para o processamento; e reduzir o tempo necessário para alcançar a acurácia desejada, além de reduzir eventualmente o erro percentual no pior dos cenários de ruído, criando uma maior robustez à interferência e melhorando a qualidade da identificação.

REFERÊNCIAS

- [1] S. Haykin, "Neural Networks and Learning Machines," 3rd ed., 2009.
- [2] D. P. Kingma, J. Lei Ba, ADAM a method for stochastic optimization, ICLR 2015.