

Reconhecimento de padrões com rede Backpropagation

IZABELLA THAÍS OLIVEIRA GOMES

LETÍCIA CÂMARA VAN DER PLOEG

Universidade de Brasília
igomesizabella@gmail.com
leticiacvdploeg@gmail.com

Resumo

Neste documento é descrito o processo de implementação de uma rede neural Backpropagation, ou Perceptron Multicamada, na solução de um problema de reconhecimento de padrões. Estes padrões são obtidos da base de dados MNIST, que fornece imagens manuscritas dos dígitos de 0 a 9.

I. INTRODUÇÃO

Os sistemas inteligentes desempenham papel essencial no desenvolvimento da tecnologia associada à inteligência artificial. Sistemas desse tipo possuem requisitos não atendidos pela arquitetura clássica computacional, como processamento paralelo e distribuído, processamento numérico, capacidade de aprendizado, tolerância a falhas e aplicável a partir de exemplos. Estes requisitos são completados pelas redes neurais artificiais.

As redes neurais artificiais (RNAs) são sistemas de computação adaptativos inspirados nas características de processamento de informação encontradas nos neurônios reais e nas características de suas interconexões [1].

As RNAs são formadas por um conjunto de neurônios artificiais que interagem entre si, de maneira semelhante ao funcionamento dos neurônios biológicos. Um dos algoritmos de treinamento utilizados para reconhecimento de padrões, problema tratado neste documento,

é o backpropagation, ou perceptron multicamada. O backpropagation procura iterativamente a mínima diferença entre as saídas desejadas e as saídas obtidas pela rede neural minimizando o erro. Neste algoritmo o aprendizado é realizado por retropropagação, etapa onde os pesos sinápticos são ajustados.

II. METODOLOGIA

O banco de dados utilizado (MNIST) fornece um conjunto de 60000 imagens para treinamento e 10000 imagens para testes. Cada imagem é composta por um conjunto de 28 x 28 pixels em níveis de cinza de 0 a 255 centralizadas pelo método de centro de gravidade, configurando 748 entradas para a rede, com exceção do bias. O número de saídas é fixo em 10 (dígitos de 0 a 9). Foi adotada a seguinte metodologia:

- Desenvolvimento da rede neural em Matlab considerando variação de camadas, número de neurônios, entradas, saídas, taxa de aprendizado e número de épocas;
- Verificação do aprendizado com o conjunto de 60000 imagens;
- Alteração de parâmetros da rede para verificação do aprendizado;
- Verificação da rede neural desenvolvida a partir dos testes com o conjunto de 10000 imagens.

Além dos parâmetros citados, a rede neural pode ser definida para as funções de ativação logística e tangente hiperbólica. As equações (1) e (2) apresentam a função logística e sua derivada, enquanto que as equações (3) e (4) apresentam a função tangente hiperbólica e sua derivada, respectivamente.

$$g(s) = \frac{1}{1 + \exp(-s)} \quad (1)$$

$$g'(s) = g(s)(1 - g(s)) \quad (2)$$

$$f(s) = \tanh(s) \quad (3)$$

$$f'(s) = (1 - f(s))(1 + f(s)) \quad (4)$$

Nas seções seguintes são apresentados os resultados e a análise considerando a função de ativação como a logística (1) e alterando os parâmetros da rede. A função de ativação tangente hiperbólica é aplicada para os valores de parâmetros que proporcionam o melhor caso obtido para a função logística.

III. RESULTADOS

Utilizando a seguinte configuração inicial, é possível obter os gráficos apresentados nas figuras 1 e 2, que indicam os erros associados ao treinamento.

- Função de ativação: logística;
- Número de épocas: 200;
- Taxa de aprendizado: 0,7;
- Número de camadas: 4;
- Neurônios por camada: 25, 20, 15, 10 (saídas).

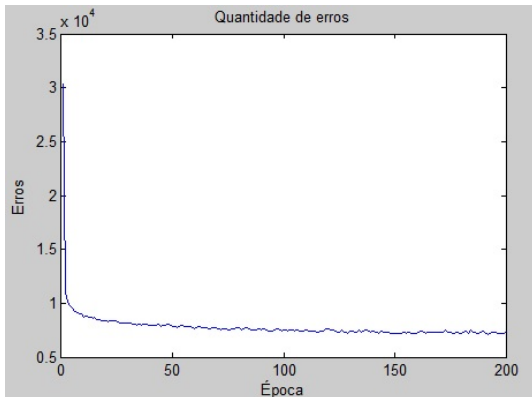


Figura 1: Quantidade de erros ao longo do aprendizado do MLP na configuração inicial

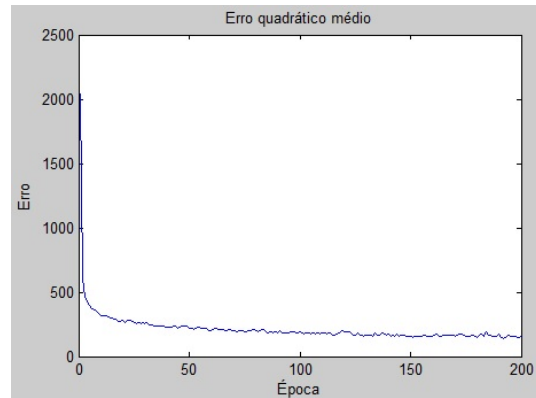


Figura 2: Erro quadrático médio para MLP na configuração inicial

A configuração foi modificada de forma a encontrar valores de erro menores.

- Função de ativação: logística;
- Número de épocas: 500;
- Taxa de aprendizado: 0,3;
- Número de camadas: 3;
- Neurônios por camada: 20, 25, 10 (saídas).

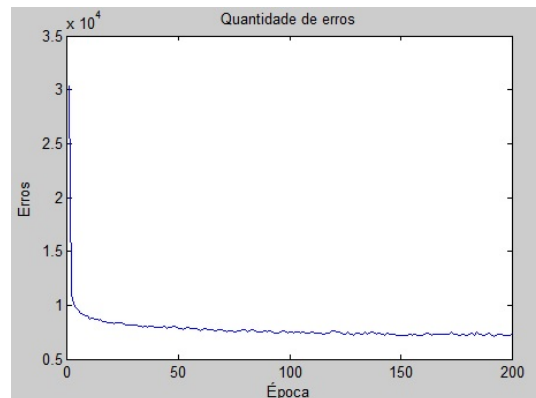


Figura 3: Quantidade de erros ao longo do aprendizado do MLP para segunda configuração

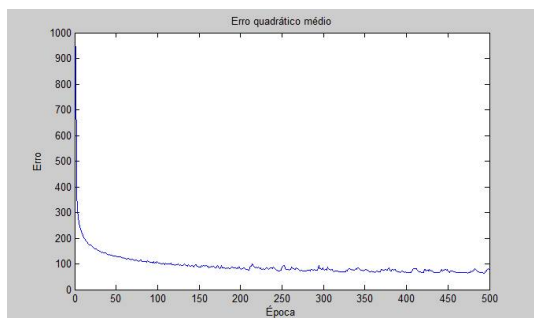


Figura 4: Erro quadrático médio para MLP na segunda configuração

Desta vez, reduzindo bastante o número de épocas como intuito principal, obtém-se a seguinte configuração:

- Função de ativação: logística;
- Número de épocas: 20;
- Taxa de aprendizado: 0,05;
- Número de camadas: 2;
- Neurônios por camada: 100, 10 (saídas).

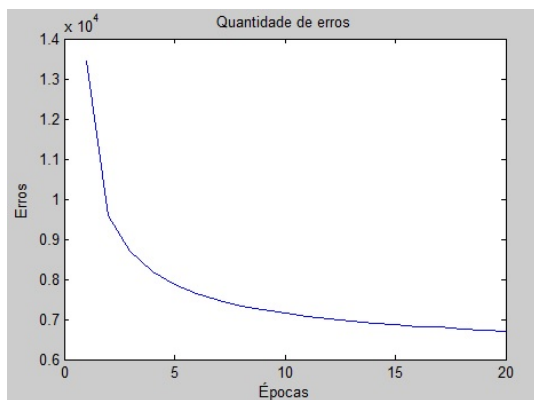


Figura 5: Quantidade de erros ao longo do aprendizado do MLP para terceira configuração

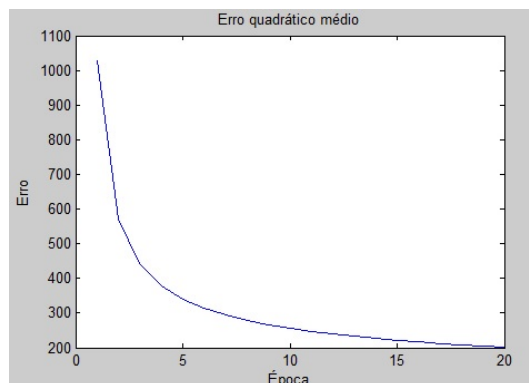


Figura 6: Erro quadrático médio para MLP na terceira configuração

Utilizando a terceira configuração, que apresenta a priori um erro menor, ou seja, possui melhor desempenho, pode-se utilizar a tangente hiperbólica como função de ativação.

- Função de ativação: tangente hiperbólica;
- Número de épocas: 20;
- Taxa de aprendizado: 0,05;
- Número de camadas: 2;
- Neurônios por camada: 100, 10 (saídas).

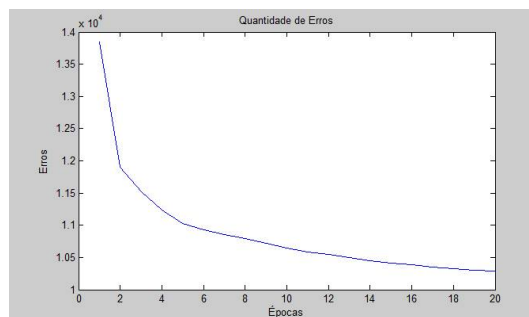


Figura 7: Quantidade de erros ao longo do aprendizado do MLP para quarta configuração

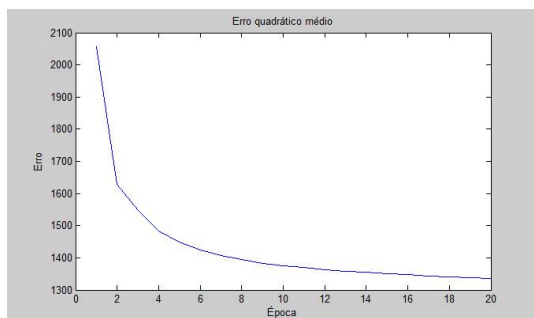


Figura 8: Erro quadrático médio para MLP na quarta configuração

IV. ANÁLISE

I. Treinamento da rede

Utilizando os gráficos obtidos para a quantidade de erros e o erro quadrático médio, é possível observar que na configuração inicial a quantidade de erros ao fim do treinamento é de 7330, ou seja, 12,22 por cento. Da mesma forma, tem-se que na segunda, na terceira e na quarta configuração os percentuais foram iguais a 10,87, 11,18 e 17,17. O erro quadrático médio é reduzido ao longo das épocas para cada configuração. Observe que o maior número de épocas, ou seja, mais tempo de treino, não indica necessariamente um melhor desempenho da rede. O menor erro correspondeu à segunda configuração com 200 épocas, uma taxa alta de aprendizado e o maior número de camadas. Este erro, porém, ficou muito próximo ao erro obtido com a rede configurada para 20 épocas, ou seja, 10 vezes menos tempo de treinamento. Nesta configuração porém a taxa de aprendizado é menor, mas o número de neurônios na camada oculta é bem maior. Há um equilíbrio dos parâmetros número de neurônios por camada e número de épocas. Perceba que ao utilizar a função tangente hiperbólica como função de ativação ao invés da função logística na terceira configuração, foi obtido um erro maior. Isso indica que apesar de simples, a função logística como função de ativação é suficiente para o problema de reconhecimento de padrão no nível atingido de desempenho.

II. Testes

Após a realização do treinamento, a rede Perceptron multicamada está pronta para lidar com o reconhecimento de padrões do conjunto de 10000 imagens disponibilizadas no MNIST. Aplicando as configurações anteriores os testes mostram os seguintes percentuais de erro:

- Configuração 1: 1419 erros ao fim do teste, percentual de erro 14,19;
- Configuração 2: 1482 erros ao fim do teste, percentual de erro 14,82;
- Configuração 3: 1213 erros ao fim do teste, percentual de erro 12,13;
- Configuração 4: 1893 erros ao fim do teste, percentual de erro 18,93.

Perceba que ao realizar os testes a configuração que indicou menor quantidade de erros foi a configuração 3, que apresenta menor número de épocas, menor taxa de aprendizado, porém muitos neurônios na camada escondida. Sendo assim, pode-se concluir que o número de neurônios na camada é um fator crucial para o desempenho da rede.

V. CONCLUSÃO

A rede perceptron multicamada desenvolvida conseguiu realizar o reconhecimento de padrões proposto sob um erro mínimo de 12,13 por cento, porém, o desempenho de MLPs com uma camada escondida situa-se na faixa de 4 por cento de erro. Isso indica que o treinamento não possui o melhor desempenho que poderia ter diante dos parâmetros alterados. A função de ativação logística se mostrou, em comparação com a tangente hiperbólica, mais precisa, fornecendo um resultado com menor erro. Já os parâmetros como número de neurônios, camadas escondidas, taxa de aprendizado se relacionaram equilibrando os resultados. Para a melhoria da rede, portanto, um número maior de neurônios associado a um número mais alto de épocas, ou maior tempo de treino, pode proporcionar uma taxa de erro mais próxima do desejado. O reconhecimento de padrões pode

ser adaptado também para um algoritmo de rede neural artificial mais refinado.

REFERÊNCIAS

- [1] S. Haykin, Neural Networks: a comprehensive foundation. New York: Mac-Millan College Publishing Co., 1999.