

Implementação de Rede Neural com Otimizador RMSProp para uso no banco de dados MNIST

Matheus Farias

Abstract—O presente projeto busca analisar a implementação de uma rede neural MLP com o uso do otimizador RMSProp para identificação de dígitos por imagens escritas de números (banco de dados MNIST).

I. INTRODUÇÃO

O projeto proposto foi para que se elabore uma rede neural MLP *from scratch* utilizando o otimizador RMSProp, de tal forma que se analise a implementação. Para validar a rede neural, deve ser utilizado a validação *10-Folds*. O conjunto de dados que será aplicado a rede neural é o conjunto de dados MNIST, onde tem-se 70046 imagens dos algarismos de 0 a 9 manuscritos. O objetivo do projeto é ter uma boa porcentagem de acertos no reconhecimento dos números utilizando uma rede neural simples. Segue abaixo a expressão de atualização dos pesos numa rede neural com otimização RMSProp:

$$E[g^2]_t = E[g^2]_{t-1} + (1 - \beta) \left(\frac{\partial C}{\partial W} \right)^2$$

$$W_t = W_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\partial C}{\partial W}$$

II. PROJETO

A. Base de Dados

Primeiramente, a base de dados oferecida consistia em 10 arquivos *.jpg* de teste e 10 arquivos *.jpg* de treino, o *i*-ésimo arquivo se refere a um grande conjunto de imagens do dígito *i*, portanto dentro do arquivo *mnist_train_0* tem vários dígitos 0, cada dígito 0 está contido em um quadrado de 28 por 28 pixels, discretizado pela intensidade de sua cor entre 0 e 255. Cada imagem foi separada em um vetor coluna de dimensão $28 \times 28 = 784$.

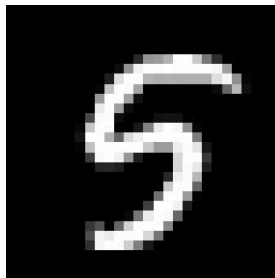


Fig. 1. Exemplo do dígito 5 do conjunto MNIST

Em *machine learning* é interessante que se normalize os dados para processamento, uma vez que os classificadores binários julgam valores de 0 e 1, portanto, os dados de imagem de 0 a 255 foram normalizados pelo fator de 255.

B. Hiperparâmetros

A rede neural MLP foi elaborada de forma totalmente conectada vide Figura 3 utilizando 2 camadas, onde a primeira camada possui 15 neurônios e a segunda possui 10 neurônios, onde a primeira camada possui função de ativação ReLU, e a segunda camada, que é a de saída possui a função de ativação sigmoide, mostradas na Figura 2. A justificativa para essas escolhas é que um dos objetivos é mostrar que uma rede neural simples pode ter um bom rendimento, e também a resolução da lista 2 utilizou a mesma rede, porém a saída era diferente, apenas classificava se estava contido ou não em um conjunto, no presente projeto, criou-se um vetor de dimensão 10, onde se o elemento *i* é 1 e o resto 0, então o dígito representado por essa saída é *i*, logo se adaptou a rede para ter 10 neurônios na saída.

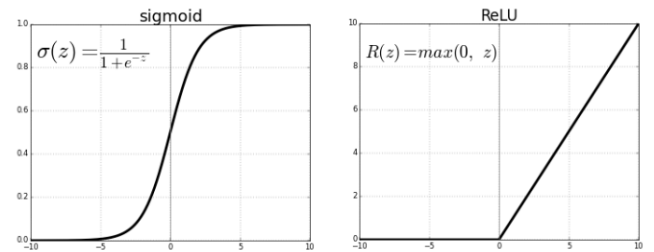


Fig. 2. Funções de ativação usadas

A taxa de aprendizagem usada no projeto foi $\alpha = 5 \times 10^{-4}$ decidida pós análise da validação e de um teste de 5 valores randomizados entre 10^{-4} e 10^{-6} . Ademais, o modo de treinamento foi o Gradiente Descendente Estocástico, uma vez que buscava-se também que o treinamento fosse mais rápido, e o número de épocas, decidido também ao randomizar 5 valores entre 1 e 10, a quantidade de épocas que melhor se adequou ao projeto foi 10.

III. VALIDAÇÃO

O conjunto de dados MNIST foi separado utilizando o *Cross-Validation 10-Folds*, portanto tem-se 54024 imagens para treino, 6003 imagens para validação e 10019 imagens para teste, o método de validação consiste em dividir o conjunto de imagens em 10 conjuntos, onde 1 dessas 9 partes são destinadas à validação, para análise dos hiperparâmetros

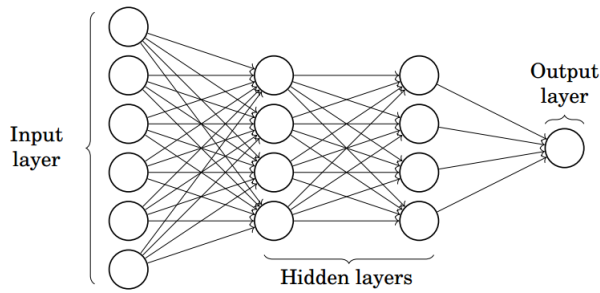


Fig. 3. Arquitetura de uma MLP completamente conectada

e sua mudança, uma vez que são determinados de forma subjetiva para buscar o resultado mais satisfatório. O Cross-Validation foi implementado usando a biblioteca sci-kit, e portanto se assegura que o projeto não está sob overfit, a porcentagem de acertos com o conjunto de dados de validação foi de aproximadamente 82%.

IV. TESTE E CONCLUSÃO

Por fim, foi possível obter uma porcentagem de acertos de aproximadamente 91% em relação ao conjunto total de teste (de 10019 imagens) para um treinamento de aproximadamente 500 segundos e de 90,5% para o conjunto total de treino (de 54024 imagens), o que mostra que a rede neural, implementada de forma bastante simples, chegou a uma taxa de acertos bastante razoável.