Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 8, n. 1, 2021.

Trabalho apresentado no XL CNMAC, Evento Virtual - Co-organizado pela Universidade do Mato Grosso do Sul (UFMS).

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Um algoritmo de otimização de pontos interiores e direções viáveis para treinamento de redes neurais.

Vitor M. A. Goulart [1]
DM-ICE-UFJF, Juiz de Fora, MG
Wilhelm P. Freire [2]
DM-ICE-UFJF, Juiz de Fora, MG
Sandro R. Mazorche [3]
DM-ICE-UFJF, Juiz de Fora, MG
Luís Henrique S. Ribeiro [4]
DCC-ICE-UFJF, Juiz de Fora, MG

As redes neurais artificiais são modelos matemáticos inspirados no funcionamento das redes neurais biológicas do ser humano. Elas são formadas por nós, divididos em camadas, e arestas ligando esses nós que representam os neurônios e as conexões por onde a informação é passada adiante de forma ponderada por pesos contidos nessas conexões.

Dada uma entrada na rede, essa informação é propagada adiante para a próxima camada através de uma soma ponderada com os pesos contidos nas ligações e ainda, é feita a aplicação de uma função não linear, chamada função de ativação.

Com isso, conseguimos uma função $f : \mathbb{R}^m \to \mathbb{R}^n$ que descreve a computação da rede. Entre os tipos de aprendizado que existem, está o aprendizado supervisionado que é quando temos um conjunto de dados de entrada e sabemos quais saídas devem ser retornadas para esse conjunto. Tomando um conjunto de dados, conseguimos escrever uma função de erro utilizando a nossa função f da rede, como por exemplo, a função do erro quadrático

$$C(w, b) = \frac{1}{2k} \sum_{i=1}^{k} ||f(x_i) - y_i||^2, \tag{1}$$

onde x_i são os dados de entrada que temos disponíveis, y_i são as respectivas saídas esperadas, k é a quantidade de dados que temos disponíveis e w e b são os parâmetros que definem a função f.

O nosso objetivo é realizar o treinamento da rede utilizando esse conjunto de dados e, para isso, desejamos encontrar um conjunto de parâmetros w e b, que são nossos pesos e vieses (*'biases'*) de modo a minimizar uma função de custo como a exemplificada na equação 1.

Uma das formas de realizar tal treinamento é através do uso da descida de gradiente e suas variações. Porém, tal método necessita do cálculo do gradiente da função que computa a rede neural que é uma função com muitas variáveis e com uma série de composições dependendo da estrutura da rede.

Para realizar esse cálculo do gradiente utiliza-se o método chamado *backpropagation*, o qual é um método mais eficiente que utiliza a regra da cadeia para calcular o gradiente da função de custo em cada camada, e aproveita essa informação para calcular o gradiente relativo à camada anterior.

[1] vitormgou@gmail.com
[2] wilhelmfreire@ice.ufjf.br
[3] sandro.mazorche@ufjf.edu.br
[4] luishenrique@ice.ufjf.br

Nesse trabalho, iremos utilizar o método FDIPA [1] (Feasible Direction Interior-Point Algorithm), o qual é um algoritmo de otimização que resolve problemas restritos da forma:

$$\begin{cases} \min & f(x) \\ \text{sujeito à} & g(x) \leq 0 \end{cases}, \tag{2}$$

onde $f : \mathbb{R}^n \to \mathbb{R}$ e $g : \mathbb{R}^n \to \mathbb{R}^m$ são funções diferenciáveis.

O FDIPA, é um algoritmo baseado nas condições de otimalidade de primeira ordem de Karush-Kuhn-Tucker (KKT). Daí, realizando algumas manipulações do sistema baseado nas condições de KKT e utilizando a fórmula de Sherman–Morrison conseguimos chegar nas seguintes expressões para as direções de busca utilizadas pelo algoritmo, no caso em que g é uma função real, que dependem apenas do gradiente da função f assim como os métodos de descida de gradiente:

$$d_\alpha = -\Big(B^{-1} + \frac{B^{-1}\nabla g \nabla g^T B^{-1}}{\frac{G}{\lambda} - \nabla g^T B^{-1} \nabla g}\Big)\nabla f; \quad d_\beta = \frac{\lambda}{G}\Big(B^{-1} + \frac{B^{-1}\nabla g \nabla g^T B^{-1}}{\frac{G}{\lambda} - \nabla g^T B^{-1} \nabla g}\Big)\nabla g \tag{3}$$

$$d = d_\alpha + \rho d_\beta$$

onde B deve ser uma matriz simétrica e positiva definida e G é uma matriz diagonal tal que $G_{ii}(x) = g_i(x)$.

Com isso, utilizamos um passo t e atualizamos nosso ponto x_n na forma $x_n = x_{n-1} + td$, onde d é a direção de descida considerando o ponto x_{n-1} e o tamanho de passo t é escolhido de modo a respeitar a restrição.

Adaptando o FDIPA a esse contexto, para realizar o treinamento de uma rede neural, adicionaremos uma restrição de modo que esta delimite os pesos dentro de uma bola de certo raio, uma vez que o problema de treinamento é considerado irrestrito na maioria das vezes.

Para realizar o teste do nosso algoritmo, utilizamos um conjunto de dados de dígitos manuscritos, MNIST [2]. Em testes iniciais, foram realizados os treinamentos considerando 1000 épocas, uma taxa de treinamento $\eta = 0.01$ e restrição dos pesos dentro de uma bola de raio 200, no qual obtivemos uma acurácia de 0.9956 no conjunto de treinamento e 0.9304 no conjunto de teste . Foram feitos testes para os algoritmos Adadelta, RMSProp e Adam por 1000 épocas os quais coletamos o resultado da melhor época, em cada caso, como a seguir:

- O algoritmo Adadelta obteve uma acurácia de 0.98194 no conjunto de treinamento e 0.9527 no conjunto de teste com parâmetros $\eta = 0.01$, $\rho = 0.9$ e $\epsilon = 10^{-6}$.
- O algoritmo RMSProp obteve uma acurácia de 0.9803 no conjunto de treinamento e 0.948 no conjunto de teste com parâmetros $\eta = 0.01$, $\alpha = 0.99$ e $\epsilon = 10^{-8}$.
- O algoritmo Adam obteve uma acurácia de 0.9835 no conjunto de treinamento e 0.9475 no conjunto de teste com parâmetros $\eta = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ e $\epsilon = 10^{-8}$.

Obtivemos resultados inciais com o FDIPA próximos aos obtidos por algoritmos mais comumente utilizados para o problema de treinamento. Com isso, mais testes, variando os parâmetros do algoritmo, serão realizados com o objetivo de explorar mais o seu potencial em problemas de treinamento de redes neurais.

Referências

[1] Herskovits, J. Feasible Direction Interior-Point Technique for Nonlinear Optimization. *Journal of Optimization Theory and Applications*, 1998. DOI : 10.1023/A:1021752227797

[2] LeCun, Y. and Cortes, C. MNIST handwritten digit database, 2010. Disponível em: http://yann.lecun.com/exdb/mnist/. Acesso: 29 de março de 2021.