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

Métodos de otimização aplicados ao aprendizado de máquina: um estudo do problema de classificação de caracteres

Matheus Becali Rocha[1]
Leonardo Delarmelina Secchin[2]
UFES, São Mateus, ES

1 Classificação de caracteres

Problemas de classificação consistem em "treinar" o computador de modo a categorizar objetos segundo padrões pré-definidos. Em nosso caso, estamos interessados no reconhecimento automatizado de caracteres numéricos ("0" a "9") escritos à mão. Para tanto, dispomos de um banco de dados com imagens para as quais sabemos o respectivo número associado (*dados de treinamento*). O problema é representado em um grafo (*rede neural*), cujo fluxo vai da camada de entrada (representação da imagem), passando por camadas intermediárias, até a camada de saída (resposta).

À cada neurônio são associadas as variáveis *peso* (w) e *viés* (b). O chamado "aprendizado de máquina" consiste na otimização dessas variáveis tendo em vista uma medida de qualidade sobre os dados de treinamento. O *treinamento* é a minimização dessa medida por um algoritmo de otimização, geralmente estocástico. Treinada a rede, calculamos o percentual de acertos relativos à dados não treinados (*dados de teste*), pois é de interesse que a rede dê respostas corretas sobre imagens desconhecidas. Este trabalho dedica-se ao estudo, sobretudo numérico, de diferentes algoritmos de otimização para o problema de classificação de caracteres numéricos, utilizando o banco de imagens MNIST [1]. Essa coletânea é bastante utilizada na literatura, e contém 60.000 imagens para treinamento e 10.000 para teste, coletadas de estudantes e censos dos EUA. Todos os algoritmos foram implementados em Python. Ainda, realizamos testes preliminares em imagens próprias, escritas e tratadas pelos autores, de modo a verificar a eficácia dos métodos.

2 Treinamento por algoritmos de otimização

O método do gradiente consiste na iteração $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$, onde $\alpha_k > 0$ é o tamanho do passo (*taxa de aprendizagem*). Os métodos mais utilizados em aprendizado de máquina consistem em variações desse esquema. Porém, dada à quantidade de variáveis, são utilizadas versões estocásticas que usam aproximações do gradiente a cada iteração. Deste modo, em cada iteração usamos apenas uma amostra reduzida dos dados de treinamento. Um desses métodos é o *Gradiente Estocástico Descendente* (do inglês, SGD), que pode ser aplicado para minimizar o erro quadrático $C(w,b) = \frac{1}{2}\sum_x \|y(w,b;x) - a(x)\|^2$. Note que $C(w,b) \approx 0$ quando o vetor resposta $y(w,b;x) \in [0,1]^{10}$ é próximo à saída real $a(x) \in \{0,1\}^{10}$, para todas as imagens de treinamento x. O vetor $y(w,b;x)$ pode ser visto como o fluxo de saída da rede, iniciado com a imagem x. A resposta é o caractere $i \in \arg\max_{i=0,\dots,9} |y_{i+1}(w,b;x)|$. Neurônios da camada de entrada recebem

[1] matheusbecali@gmail.com
[2] leonardo.secchin@ufes.br

a escala de cinza de um *pixel* da imagem, e cada neurônio seguinte recebe uma média ponderada das saídas dos neurônios da camada anterior. Especificamente, o neurônio i da camada j recebe o fluxo $s(\sum_j w_{ij}x_{j-1} - b_i)$, onde s é uma *função de ativação* que escala a saída entre 0 e 1.

A Figura 1 ilustra a representação do problema. As imagens de entrada têm 28×28 *pixels*, totalizando $28^2 = 784$ neurônios na camada de entrada (em verde). O vetor $y(w, b; x)$ da camada de saída (em amarelo) é computado por um processo *forward pass*, calculando o fluxo em cada neurônio da esquerda para a direita. Após esse passo, propagamos o fluxo na rede no sentido inverso (*backward pass*), possibilitando o cálculo de ∇C (onde a soma é sobre uma amostra dos dados). Essa técnica é conhecida como *backpropagation*, e se justifica pela regra da cadeia. É uma maneira barata para computar ∇C, e é essencial em uma boa implementação.

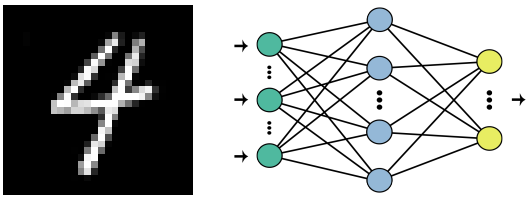

Figura 1: Representação do problema.

Outros métodos são aplicados no aprendizado de máquina. Dentre eles está o SGD com *momento*, que adiciona um vetor "velocidade" à direção de busca. Uma limitação do SGD e do SGD com momento é que a taxa de aprendizagem é constante. Outros métodos mais modernos, e aplicados com sucesso à vários contextos, empregam taxas de aprendizagem adaptativas. Dentre eles, encontram-se Adam, AdaGrad e AdaDelta. Para mais informações, veja por exemplo [2].

3 Resultados, trabalhos futuros e conclusões

Testes preliminares foram realizados considerando uma rede com 1 camada intermediária. Com 30 neurônios, SGD alcança 83% de correção para os dados de teste da MNIST. Quando a rede treinada é avaliada sobre dígitos próprios, chegamos a uma taxa de apenas 42%. O SGD com momento apresenta melhores resultados (87% na MNIST e 68% nas imagens próprias). Realizamos ainda testes com outros métodos, como o Adam, obtendo resultados melhores que SGD. Isso corrobora resultados relatados na literatura. Testes com mais neurônios na camada intermediária também foram realizados. Cabe ressaltar que este trabalho de iniciação científica está em desenvolvimento. Iniciamos testes em grandes redes, executando-os em um máquina robusta com acesso remoto ininterrupto. Pretendemos utilizar dados da MNIST "estendida" [3], diversificando ainda as imagens por rotações e translações a fim de testar a robustez das respostas. Finalmente, este trabalho representa um ponto de partida no estudo de outros problemas (de classificação) e aplicações.

Referências

[1] LeCun, Y., Bottou, L., Bengio, Y. e Haffner, P. Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, v. 86, n. 11, p. 2278-2324, 1998.

[2] Sun, S., Cao, Z., Zhu, H., e Zhao, J. A survey of optimization methods from a machine learning perspective. *IEEE Transactions on Cybernetics*, v. 50, n. 8, p. 3668-3681, 2020.

[3] Cohen, G., Afshar, S., Tapson, J., e van Schaik, A. EMNIST: an extension of MNIST to handwritten letters, *International Joint Conference on Neural Networks*, p. 2921-2926, 2017.