



Data Science Academy

www.datascienceacademy.com.br

Matemática Para Machine Learning

Otimização RMSProp



Esta é uma ligeira variação do AdaGrad e funciona melhor na prática, uma vez que aborda os problemas deixados em aberto pelo AdaGrad.

Similar a AdaGrad, aqui também manteremos a estimativa do gradiente quadrado, mas ao invés de deixar que a estimativa quadrada acumule ao longo do treinamento, nós deixamos essa estimativa decair gradualmente. Para conseguir isso, multiplicamos a estimativa atual de gradientes quadrados com a taxa de decaimento (decay rate).

```
squared_gradients = 0
for i in range(iterations_count):
    param_gradients = evaluate_gradients(loss_function,
                                        data,
                                        params)

    squared_gradients = decay_rate * squared_gradients + (1 -
                                                         decay_rate) * param_gradients * param_gradients
    params -= learning_rate * param_gradients /
              (np.sqrt(squared_gradients) + 1e-8)
```

Referências:

<http://cs229.stanford.edu/notes/cs229-notes1.pdf>

<https://stanford.edu/~rezab/classes/cme323/S15/notes/lec11.pdf>

<https://arxiv.org/abs/1609.04747>

<http://runder.io/optimizing-gradient-descent/>