



**Data Science  
Academy**

[www.datascienceacademy.com.br](http://www.datascienceacademy.com.br)

**Deep Learning I**

**O Que é Exatamente a Taxa de  
Aprendizagem?**



## Mas como descemos o morro afinal?

Estamos no topo do morro e precisamos descer até a base, na superfície plana, mas como realizamos isso? (Se você não faz ideia do que estamos falando aqui, acesse e leia o item de aprendizagem anterior).

Intuitivamente, queremos descer o morro. Mas quando estamos falando sobre nossa função de custo, como sabemos de que forma é o declive? Felizmente, o gradiente fornece exatamente esta informação.

Tecnicamente, o gradiente realmente avança, na direção da subida mais íngreme. Mas se colocarmos um sinal negativo na frente deste valor, obtemos a direção da descida mais íngreme, que é o que precisamos. Pense nisso como um vetor de números. Cada número representa a quantidade pela qual devemos ajustar um peso ou bias correspondentes na rede neural. Ajustar todos os pesos e bias pelos valores de gradiente reduz o custo (ou erro) da rede.

## Taxa de Aprendizagem

Agora sabemos como descer o morro. Precisamos então considerar a quantidade de força a ser aplicada ao impulso. Isso é conhecido como a taxa de aprendizagem, que é um nome apropriado, pois esse valor determina a rapidez ou lentidão que a rede neural aprende.

Você pode querer estabelecer uma taxa de aprendizado grande, para que a rede neural aprenda mais rápido certo? Errado! Se o valor for muito grande, você pode superar o alvo e eventualmente divergir. Então, qual é a taxa de aprendizagem ideal?

Este é mais um jogo de adivinhação do que qualquer outra coisa. Não existe uma única resposta a esta pergunta e precisamos testar diferentes valores (e esse é seu trabalho). Empiricamente o intervalo de 0,1 a 0,0001 funciona bem na maioria dos casos. A faixa de 0,001 a 0,0001 é popular, pois 0,1 e 0,01 às vezes são muito grandes.

Aqui está a fórmula para descida gradiente (pseudocódigo):

$$x = x - \text{taxa\_aprendizado} * \text{gradiente\_de\_x}$$

Onde  $x$  é um parâmetro usado pela rede neural (ou seja, um único peso ou bias). Multiplicamos **gradiente\_de\_x** (a direção ascendente) por **taxa\_aprendizado** (a força do “empurrão”) e depois subtraímos de  $x$  para fazer o “empurrão” ir para baixo.

Impressionante não? Redes Neurais são um tema fascinante! Agora, vamos estudar como implementar esses conceitos.