

Material Suplementar para “Introdução às redes neurais para Físicos”

Apêndice A - Cálculo da função custo

Cada tipo de tarefa (como regressão, classificação, etc.) geralmente tem funções de custo mais adequadas. Aqui estão algumas das principais funções de custo usadas em ML:

Função de Ativação	$f(x)$	Derivada
Sigmoide	$\frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh	$\tanh(x)$	$f'(x) = 1 - f(x)^2$
ReLU	$\max(0, x)$	$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$
Leaky ReLU	$\max(0.01x, x)$	$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0.01 & \text{if } x \leq 0 \end{cases}$

Tabela 1: Comparação das funções de ativação e suas derivadas

Para ilustrar melhor ao leitor apresentamos os gráficos das funções da ativação e sua derivadas.

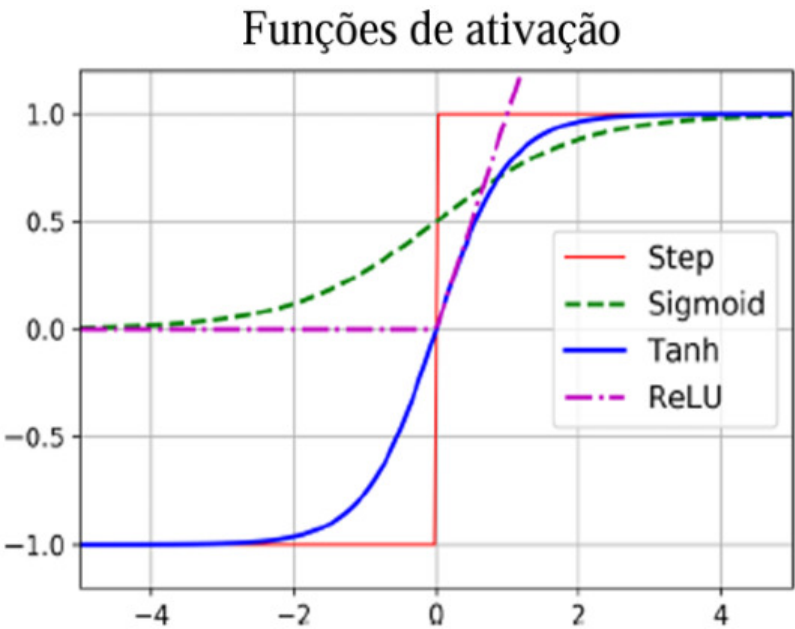


Figura 1S: Funções de ativação step, sigmoid, tanh e relu, citadas na Tabela 1. Fonte: [1].

Podemos notar que para valores grandes algumas funções(sigmoid, tanh e step) tendem 1S a zero em suas derivadas 2S, que por consequência afetar o processo de treinamento.

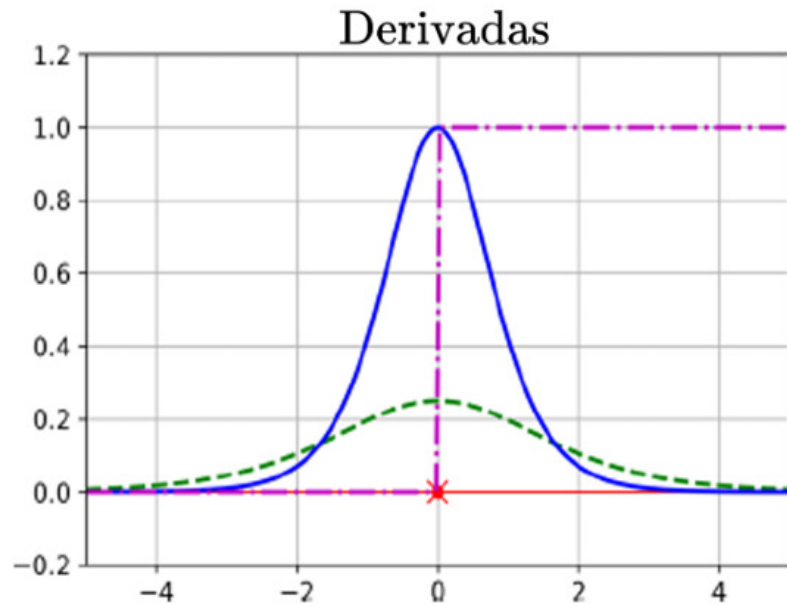


Figura 2S: Derivadas das funções de ativação step, sigmoid, tanh e relu, citadas na tabela 1. Fonte: [1].

Loss	Expressão Matemática
MSE	$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
MAE	$\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
BCE	$-\sum [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$
CCE	$-\sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(\hat{y}_{ij})$

Tabela 2: Exemplos de funções de ativação e Funções Custo.

A tabela II apresenta algumas das funções custo, como Erro Quadrático Médio (*Mean Squared Error - MSE*) uma das funções de custo mais comuns para tarefas de regressão, ela mede a média dos quadrados das diferenças entre os valores preditos e os reais. Erro Absoluto Médio (*Mean Absolute Error - MAE*): Também usado em regressão, o MAE mede a média das diferenças absolutas entre os valores preditos e os reais, proporcionando uma medida de erro que é robusta a outliers. Entropia Cruzada (*Cross-Entropy Loss-CCE*), muito usada em problemas de classificação. Esta função de custo mede a diferença entre duas distribuições de probabilidade, a prevista pelo modelo e a distribuição verdadeira. Para a classificação binária, é conhecida como Entropia Cruzada Binária - BCE (*Binary Cross-Entropy*). Observação, a *Loss BCE* é um caso específico da CCE, onde temos apenas duas classes.

Durante o cálculo da atualização dos pesos, precisamos do gradiente da função custo em relação aos parâmetros. Então, apresentaremos um exemplo da derivação da expressão utilizando como exemplos a função custo MSE e a

função de ativação sigmoide.

$$\begin{aligned}
\nabla_w \mathcal{L} &= \frac{d}{dw} (y - \hat{y})^2, \\
\nabla_w \mathcal{L} &= -2(y - \hat{y}) \frac{d}{dw} \hat{y}, \\
\nabla_w \mathcal{L} &= -2(y - \hat{y}) \frac{d}{dw} f(w^T \mathbf{x}), \\
\nabla_w \mathcal{L} &= -2(y - \hat{y}) f' \frac{d}{dw} (w^T \mathbf{x}), \\
\nabla_w \mathcal{L} &= -2(y - \hat{y}) f' \mathbf{x} \frac{d}{dw} w^T, \\
\nabla_w \mathcal{L} &= -2(y - \hat{y}) f' \mathbf{x}.
\end{aligned}$$

Fornecida a expressão, para os w podemos fazer o mesmo para b .

$$\begin{aligned}
\nabla_b \mathcal{L} &= \frac{d}{db} (y - \hat{y})^2, \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) \frac{d}{db} \hat{y}, \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) \frac{d}{db} f(w^T \mathbf{x}), \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) f' \frac{d}{db} (w^T \mathbf{x}), \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) f' \mathbf{x} \frac{d}{db} w^T, \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) f'.
\end{aligned}$$

Assim,

$$\begin{aligned}
\nabla_w \mathcal{L} &= -2(y - \hat{y}) f' \mathbf{x}, \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) f',
\end{aligned}$$

onde f' depende de qual função de ativação está sendo usada. Na tabela 1 temos alguns exemplos, para ilustrar ao leitor usaremos o Sigmoid $f' = f(1 - f)$, onde $f = \hat{y}$

$$\begin{aligned}
\nabla_w \mathcal{L} &= -2(y - \hat{y}) \hat{y}(1 - \hat{y}) \mathbf{x}, \\
\nabla_b \mathcal{L} &= -2(y - \hat{y}) \hat{y}(1 - \hat{y}).
\end{aligned}$$

Referências

- [1] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* (O'Reilly Media, Sebastopol, 2017).