


An Overview on Deep Neural Networks: Part 3



Activation Functions' Derivatives

Activation Functions' Derivatives

$$\sigma'^{[l]}(Z^{[l]})_{i,j,u,v} = \frac{\partial}{\partial Z_{u,v}^{[l]}} \sigma^{[l]}(Z^{[l]})_{i,j}$$

ReLU

$$\sigma^{[l]}(Z^{[l]})_{i,j} = \max\{0, Z_{i,j}^{[l]}\}$$

$$\sigma'^{[l]}(Z^{[l]})_{i,j,u,v} = \begin{cases} 1 & i = u \text{ and } j = v \text{ and } Z_{u,v}^{[l]} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Sigmoid

$$\sigma^{[l]}(Z^{[l]})_{i,j} = \frac{1}{1 + \exp(-Z_{i,j}^{[l]})}$$

$$\sigma'^{[l]}(Z^{[l]})_{i,j,u,v} = \begin{cases} A_{i,j}^{[l]} (1 - A_{i,j}^{[l]}) & i = u \text{ and } j = v \\ 0 & \text{otherwise} \end{cases}$$

Softmax

$$\sigma^{[l]}(Z^{[l]})_{i,j} = \frac{\exp(Z_{i,j}^{[l]})}{\sum_{k=1}^{m_h^{[l]}} \exp(Z_{i,k}^{[l]})}$$

$$\sigma'^{[l]}(Z^{[l]})_{i,j,u,v} = \begin{cases} A_{i,j}^{[l]} (1 - A_{i,j}^{[l]}) & i = u \text{ and } j = v \\ -A_{i,j}^{[l]} A_{i,v}^{[l]} & i = u \text{ and } j \neq v \\ 0 & i \neq u \end{cases}$$



Loss Functions' Derivatives

SSE

$$J(Y, \hat{Y}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{m_y} (\hat{Y}_{i,j} - Y_{i,j})^2$$

$$\frac{\partial \mathcal{L}}{\partial \hat{Y}_{i,j}} = \hat{Y}_{i,j} - Y_{i,j}$$

$$\frac{\partial \mathcal{L}}{\partial \hat{Y}} = \hat{Y} - Y$$

BCE

$$J(Y, \hat{Y}) = - \sum_{i=1}^n \sum_{j=1}^{m_y} \left(Y_{i,j} \log(\hat{Y}_{i,j}) + (1 - Y_{i,j}) \log(1 - \hat{Y}_{i,j}) \right)$$

$$\frac{\partial \mathcal{L}}{\partial \hat{Y}_{i,j}} = \frac{\hat{Y}_{i,j} - Y_{i,j}}{\hat{Y}_{i,j}(1 - \hat{Y}_{i,j})}$$

$$\frac{\partial \mathcal{L}}{\partial \hat{Y}} = (\hat{Y} - Y) \oslash (\hat{Y} \odot (1 - \hat{Y}))$$

CCE

$$J(Y, \hat{Y}) = - \sum_{i=1}^n \sum_{j=1}^{m_y} (Y_{i,j} \log(\hat{Y}_{i,j}))$$

$$\frac{\partial \mathcal{L}}{\partial \hat{Y}_{i,j}} = - \frac{Y_{i,j}}{\hat{Y}_{i,j}}$$

$$\frac{\partial \mathcal{L}}{\partial \hat{Y}} = -Y \oslash \hat{Y}$$

Different Types of Gradient Descent

Different Types of Gradient Descent

- Batch
- Stochastic
- Mini-Batch



Some Mathematical Notations and Their NumPy Equivalents

Some Mathematical Notations and Their NumPy Equivalents

AB	\equiv	$A @ B$
$A:B$	\equiv	<code>np.tensordot(A, B)</code>
$A \odot B$	\equiv	$A * B$
$A \oslash B$	\equiv	A / B