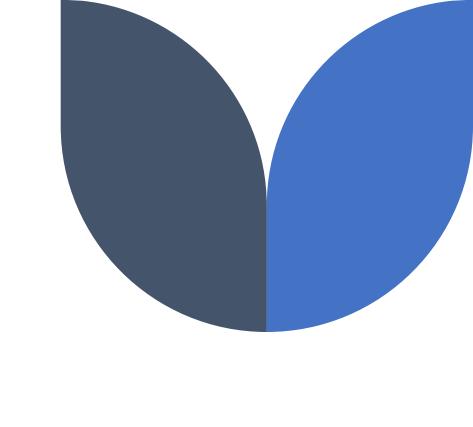
An Overview on Deep Neural Networks: Part 3



# Activation Functions' Derivatives

#### Activation Functions' Derivatives

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

#### ReLU

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

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

### Sigmoid

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

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

#### Softmax

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

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

### Loss Functions' Derivatives

#### SSE

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

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

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

#### BCE

$$J(Y,\widehat{Y}) = -\sum_{i=1}^{n_y} \sum_{j=1}^{m} \left( Y_{i,j} \log \left( \widehat{Y}_{i,j} \right) + \left( 1 - Y_{i,j} \right) \log \left( 1 - \widehat{Y}_{i,j} \right) \right)$$

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

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

CE

$$J(Y,\widehat{Y}) = -\sum_{i=1}^{n_y} \sum_{j=1}^m \left( Y_{i,j} \log \left( \widehat{Y}_{i,j} \right) \right)$$

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

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

# 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

*A*: *B* 

 $A \odot B$ 

 $A \oslash B$ 

 $\equiv$ 

 $\equiv$ 

A a B

np.tensordot(A, B)

A \* B

A / B