


An Overview on Deep Neural Networks: Part 2



Backward Propagation

Calculating the Loss and Cost

$$A^{[0]} = X$$

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$$

$$\forall l = 1, 2, \dots, L$$

$$A^{[l]} = \sigma^{[l]}(Z^{[l]})$$

$$\forall l = 1, 2, \dots, L$$

$$\hat{Y} = A^{[L]}$$

$$C = \frac{1}{m}J(Y, \hat{Y})$$

Gradient Descent and Chain Rule

$$A^{[l]} = \sigma^{[l]}(Z^{[l]})$$

$$\forall l = 1, 2, \dots, L$$

$$\hat{Y} = A^{[L]}$$

$$C = \frac{1}{m} J(Y, \hat{Y})$$

$$\frac{\partial C}{\partial \hat{Y}} = \frac{1}{m} \frac{\partial J}{\partial \hat{Y}}$$

$$\frac{\partial C}{\partial A^{[L]}} = \frac{\partial C}{\partial \hat{Y}}$$

$$\frac{\partial A^{[l]}}{\partial Z^{[l]}} = \sigma'^{[l]}(Z^{[l]})$$

$$\forall l = L, L - 1, \dots, 1$$

Gradient Descent and Chain Rule

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]} \quad \forall l = 1, 2, \dots, L$$

$$\frac{\partial Z^{[l]}}{\partial W^{[l]}} = A^{[l-1],T} \quad \forall l = L, L-1, \dots, 1$$

$$\frac{\partial Z^{[l]}}{\partial b^{[l]}} = 1_{m \times 1} \quad \forall l = L, L-1, \dots, 1$$

$$\frac{\partial Z^{[l]}}{\partial A^{[l-1]}} = W^{[l],T} \quad \forall l = L, L-1, \dots, 2$$

Putting It All Together

$$\frac{\partial C}{\partial W^{[l]}} = \left(\frac{\partial C}{\partial A^{[l]}} \div \frac{\partial A^{[l]}}{\partial Z^{[l]}} \right) \frac{\partial Z^{[l]}}{\partial W^{[l]}} = \left(\frac{\partial C}{\partial A^{[l]}} \div \sigma'^{[l]}(Z^{[l]}) \right) A^{[l-1],T} \quad \forall l = L, L-1, \dots, 1$$

$$\frac{\partial C}{\partial b^{[l]}} = \left(\frac{\partial C}{\partial A^{[l]}} \div \frac{\partial A^{[l]}}{\partial Z^{[l]}} \right) \frac{\partial Z^{[l]}}{\partial b^{[l]}} = \left(\frac{\partial C}{\partial A^{[l]}} \div \sigma'^{[l]}(Z^{[l]}) \right) 1_{m \times 1} \quad \forall l = L, L-1, \dots, 1$$

$$\frac{\partial C}{\partial A^{[l-1]}} = \frac{\partial Z^{[l]}}{\partial A^{[l-1]}} \left(\frac{\partial C}{\partial A^{[l]}} \div \frac{\partial A^{[l]}}{\partial Z^{[l]}} \right) = W^{[l],T} \left(\frac{\partial C}{\partial A^{[l]}} \div \sigma'^{[l]}(Z^{[l]}) \right) \quad \forall l = L, L-1, \dots, 2$$



Updating the Weights and the Biases

$$\Delta W^{[l]} = -\alpha \frac{\partial \mathcal{C}}{\partial W^{[l]}}$$

$$\forall l = L, L - 1, \dots, 1$$

$$\Delta b^{[l]} = -\alpha \frac{\partial \mathcal{C}}{\partial b^{[l]}}$$

$$\forall l = L, L - 1, \dots, 1$$



Loss Function Examples

Loss Function Examples

- Sum of Squared Errors (SSE)
- Binary Cross Entropy (BCE)
- Categorical Cross Entropy (CCE)

SSE

- Commonly used in regression.
- The sum is divided by two for easier derivative calculation.

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

BCE

- Commonly used in binary or multi-label classification.

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

CCE

- Commonly used in multi-class classification.

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

Summary of Activation and Loss Functions

Summary of Activation and Loss Functions

Task	Output Layer Activation Function	Loss Function
Regression	ReLU	SSE
Binary or Multi-Label Classification	Sigmoid	BCE
Multi-Class Classification	Softmax	CCE

