An Overview on Deep Neural Networks: Part 2

Backward Propagation

Calculating the Loss and Cost

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

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

$$A^{[l]} = \sigma^{[l]}\left(Z^{[l]}\right) \qquad \forall l = 1,2,\ldots,L$$

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

$$C = \frac{1}{n}J\left(Y,\hat{Y}\right)$$

Gradient Descent and Chain Rule

$$A^{[l]} = \sigma^{[l]}\left(Z^{[l]}\right) \qquad \forall l = 1,2,\dots,L$$

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

$$C = \frac{1}{n} J\left(Y, \hat{Y}\right)$$

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

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

$$\frac{\partial A^{[l]}}{\partial Z^{[l]}} = \sigma'^{[l]}\left(Z^{[l]}\right) \qquad \forall l = L, L-1, \dots, 1$$

Gradient Descent and Chain Rule

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

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

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

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

Putting It All Together

- $\frac{\partial C}{\partial A^{[L]}} = \frac{1}{n}\frac{\partial J}{\partial \hat{Y}}$
- $\frac{\partial C}{\partial W^{[l]}} = \frac{\partial Z^{[l]}}{\partial W^{[l]}}\left(\frac{\partial C}{\partial A^{[l]}} : \frac{\partial A^{[l]}}{\partial Z^{[l]}}\right) = A^{[l-1],T}\left(\frac{\partial C}{\partial A^{[l]}} : \sigma'^{[l]}\left(Z^{[l]}\right)\right) \qquad \forall l = L, L-1, \dots, 1$
- $\frac{\partial C}{\partial b^{[l]}} = \frac{\partial Z^{[l]}}{\partial b^{[l]}}\left(\frac{\partial C}{\partial A^{[l]}} : \frac{\partial A^{[l]}}{\partial Z^{[l]}}\right) = 1_{1\times n}\left(\frac{\partial C}{\partial A^{[l]}} : \sigma'^{[l]}\left(Z^{[l]}\right)\right) \qquad \forall l = L, L-1, \dots, 1$
- $\frac{\partial C}{\partial A^{[l-1]}} = \left(\frac{\partial C}{\partial A^{[l]}} : \frac{\partial A^{[l]}}{\partial Z^{[l]}}\right)\frac{\partial Z^{[l]}}{\partial A^{[l-1]}} = \left(\frac{\partial C}{\partial A^{[l]}} : \sigma'^{[l]}\left(Z^{[l]}\right)\right)W^{[l],T} \qquad \forall l = L, L-1, \dots, 2$

Updating the Weights and the Biases

$$\Delta W^{[l]} = -\alpha \frac{\partial C}{\partial W^{[l]}} \qquad \forall l = L, L-1, \dots, 1$$

$$\Delta b^{[l]} = -\alpha \frac{\partial C}{\partial b^{[l]}} \qquad \forall l = L, L-1, \dots, 1$$

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.

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

BCE

- Commonly used in binary or multi-label classification.

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

CCE

- Commonly used in multi-class classification.

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

Summary of Activation and Loss Functions

Summary of Activation and Loss Functions

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