Neural Networks Math

Asher Labovich

November 2024

1 Introduction

L layers, weights are defined as w^l_{jk} in = weight going from neuron j in layer (l - 1) to neuron k in layer (l). n_l = number of neurons in layer l

Dimensions

W = n x m matrix, with n = number of neurons in layer (l-1), m number of neurons in layer l.

δ = 1 x n matrix

z_l = 1 x n matrix

The Four Fundamental Theorems of Feed-Forward Networks

1. **Output Layer Error:**

$$\delta^L = \frac{\partial C}{\partial z^L} = \nabla C \odot \sigma'(z^L)$$

2. **Hidden Layer Error:**

$$\delta^l = (\delta^{l+1} W^{l+1}) \odot \sigma'(z^l)$$

3. **Gradient of Cost w.r.t Weights:**

$$\frac{\partial C}{\partial W^l} = (a^{l-1})^T \delta_l$$

4. **Gradient of Cost w.r.t Biases:**

$$\frac{\partial C}{\partial b^l} = \delta^l$$

Explanation of Notation

- C is the cost function.
- δ^L and δ^l represent the error in the output layer and the l-th hidden layer, respectively.
- $\sigma'(z^l)$ is the derivative of the activation function applied to z^l.
- W^l and b^l are the weights and biases for the l-th layer.
- ∇C represents the gradient of the cost function.
- a^{l-1} represents the activations from the previous layer.
- $\odot$ represents the element-wise (Hadamard) product.

Formula 1: Final Layer

$\delta_j^L = \frac{\partial C}{\partial z_j^L} = \frac{\partial C}{\partial \sigma(z_j^L)} \frac{\partial \sigma(z_j^L)}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L)$

So, $\delta^L = \frac{\partial C}{\partial z^L} = \nabla C \odot \sigma'(z^L)$

Formula 2: Previous Layer

$$
\begin{aligned}
\delta_j^l &= \\
\frac{\partial C}{\partial z_j^l} &= \\
\sum_{k=1}^{n_{l+1}} \frac{\partial C}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} &= \\
\sum_{k=1}^{n_{l+1}} \delta_k^l \frac{\partial z_k^{l+1}}{\partial \sigma_k(z_j^l)} \frac{\partial \sigma_l(z_j^l)}{\partial z_j^l} &= \\
\sum_{k=1}^{n_{l+1}} \delta_k^{l+1} \sigma_l'(z_j^l) \frac{\partial z_k^{l+1}}{\partial a_j^l} &= \\
\sum_{k=1}^{n_{l+1}} \delta_k^{l+1} \sigma_l'(z_j^l) \frac{\partial \sum_{m=1}^{n_l} w_{mk}^{l+1} a_m^l + B_k^{l+1}}{\partial a_j^l} &= \\
\sum_{k=1}^{n_{l+1}} \delta_k^{l+1} \sigma_l'(z_j^l) w_{jk}^{l+1} &= \\
\sigma_l'(z_j^l) \sum_{k=1}^{n_{l+1}} \delta_k^{l+1} w_{jk}^{l+1} &= \\
\sigma_l'(z_j^l) \delta^{l+1} w_{j:}^{l+1}
\end{aligned}
$$

So, $\delta^l = \delta^{l+1} W^{l+1^T} \odot \ \sigma_l'(z^l)$

Formula 3: Weights

$\frac{\partial C}{\partial w_{kj}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{kj}^l} = \delta_j^l \frac{\partial \sum_{i=1}^{n_{l-1}} w_{ij}^l a_i^{l-1} + B_j^l}{\partial w_{kj}^l} = \delta_j^l a_k^{l-1}$

So, W = $\begin{bmatrix} \delta_1^l a_1^{l-1} & \delta_2^l a_1^{l-1} & \delta_3^l a_1^{l-1} \\ \delta_1^l a_2^{l-1} & \delta_2^l a_2^{l-1} & \delta_3^l a_2^{l-1} \\ \delta_1^l a_3^{l-1} & \delta_2^l a_3^{l-1} & \delta_3^l a_3^{l-1} \end{bmatrix} = (a^{l-1})^T \delta_l$

Formula 4: Bias

$\frac{\partial C}{\partial B_j^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial B_j^l} = \delta_j^l \frac{\partial z_j^l}{\partial B_j^l} = \delta_j^l \frac{\partial \sum_{i=1}^{n_{l-1}} w_{ij}^l a_i^{l-1} + B_j^l}{\partial B_j^l} = \delta_j^l.$

So, $\frac{\partial C}{\partial B^l} = \delta_l$