Derivation of a Two-Layer Multilayer Perceptron (MLP)

Overview

We will be creating a 2 layer MLP as such:

$$\text{input\_layer} \longrightarrow \text{hidden\_layer} \longrightarrow \text{output\_layer}$$

We'll now elaborate on what actually happens. Remainder, this is the flow for predicting a value, to learn, we'll have to apply a loss function on the output and use backwards propagation to calculate our gradients and optimize our weights.

Notation

Our input will be represented as the matrix $\mathrm{X} \in \mathbb{R}^{m\times d}$ (in our case d=784 and m is the number of samples).

Each of our samples can be classified into $K = 10$ classes.

Our weights are going to be $\mathrm{W}^{(1)} \in \mathbb{R}^{d\times h}$ and $\mathrm{b}^{(1)} \in \mathbb{R}^{1\times h}$ for the pass from the input layer to the hidden layer and $\mathrm{W}^{(2)} \in \mathbb{R}^{h\times K}$ and $\mathrm{b}^{(2)} \in \mathbb{R}^{1\times K}$ for the pass from the hidden layer to the output layer.

We'll use the sigmoid function as our first non-linear activation function and softmax as our second non-linear activation function to get our probabilities vector (we can then decode it into one-hot to find out the actual class our model predicted).

Note. *The sigmoid and softmax functions are defined as such:*

$$\forall z \in \mathbb{R}\colon \sigma(z) = \frac{1}{1+e^{-z}}$$

$$\forall \mathrm{z} \in \mathbb{R}^K \forall 1 \leq i \leq K\colon\ (\mathrm{softmax}(\mathrm{z}))_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$$

Where, v_i is the ith entry in the vector v. *The softmax function returns a vector.*

Activation

Note. *Activating a function $f\colon \mathbb{R} \longrightarrow \mathbb{R}$ on a vector behaves as activating the function on each of the vector components. Thus we can activate the sigmoid function on a vector the get a sigmoid vector.*

For the pass from the input to the hidden layer we'll have the following pre-activation and activation:

$$\mathrm{Z}^{(1)} = \mathrm{X}\mathrm{W}^{(1)} + \mathrm{b}^{(1)}, \qquad \mathrm{A}^{(1)} = \sigma\left(\mathrm{Z}^{(1)}\right) \qquad \in \mathbb{R}^{m\times h}$$

For the second pass we are doing something similar

$$\mathrm{Z}^{(2)} = \mathrm{A}^{(1)}\mathrm{W}^{(2)} + \mathrm{b}^{(2)}, \qquad \mathrm{A}^{(2)} = \text{softmax}\left(\mathrm{Z}^{(1)}\right) \qquad \in \mathbb{R}^{m\times K}$$

where $\text{Row}_i\left(\mathrm{A}^{(2)}\right) = \text{softmax}\left(\text{Row}_i\left(\mathrm{Z}^{(2)}\right)\right)$ ($\mathbb{A}^{(2)}$ is a column vector).

Note. *We are adding a matrix to a vector. Because our* b *vectors are a row vector, the addition is defined as adding to each of the rows. This is the behavior numpy defines (and a lot of other computational modules for a just reason) and this is how we'll define it.*

Output

We'll have our labels be represented as $\mathrm{Y} \in \{0,1\}^{m\times K}$ where each row is a **one-hot** representation of the true label.

Loss

We will be using the *Cross-Entropy Loss* which looks like:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{K}\mathrm{Y}_{i,j}\ln \mathrm{A}^{(2)}_{i,j}$$

Backwards Propagation

Derivatives w.r.t. $\mathrm{Z}^{(1)}$

For a single sample i, we can define our loss to be:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k \ln a_k, \qquad a_k = \left(\text{softmax}\left(\mathrm{z}\right)\right)_k$$

First,

$$\frac{\partial \mathcal{L}}{\partial a_k} = -\frac{y_k}{a_k}$$

Next, we calculate the Jacobian of softmax to get

$$\frac{\partial a_k}{\partial z_j} = a_k\left(\delta_{kj} - a_j\right)$$

where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kj} = 1$ if $k = j$.
Now get get that:

$$\frac{\partial \mathcal{L}}{\partial z_j} = \sum_{k=1}^{K}\frac{\partial \mathcal{L}}{\partial a_k}\frac{\partial a_k}{\partial z_j} = -\sum_{k=1}^{K}\frac{y_k}{a_k}a_k\left(\delta_{kj} - a_j\right) = -\sum_{k=1}^{K} y_k\delta_{kj} + \sum_{k=1}^{K} y_k a_j = a_j - y_j$$

as a_k cancels out and δ_{kj} gives us 1 only when $k = j$ and $y_k = 1$ only when $j = k$ (as per our one-hot encoding).

And for all m samples we get

$$\boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{Z}^{(2)}} = \frac{1}{m}\left(\mathrm{A}^{(2)} - \mathrm{Y}\right)}$$

Gradients for $\mathrm{W}^{(2)}, \mathrm{b}^{(2)}$

Using

$$\mathrm{Z}^{(2)} = \mathrm{A}^{(1)}\mathrm{W}^{(2)} + \mathrm{b}^{(2)}$$

$$\frac{\partial \mathrm{Z}^{(2)}}{\partial \mathrm{W}^{(2)}} = \mathrm{A}^{(1)} \qquad \frac{\partial \mathrm{Z}^{(2)}}{\partial \mathrm{b}^{(2)}} = I_{h\times K}$$

Thus, we get

$$\boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{W}^{(2)}} = \left(\mathrm{A}^{(1)}\right)^{\top} \cdot \frac{\partial \mathcal{L}}{\partial \mathrm{Z}^{(2)}}} \qquad \boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{b}^{(2)}} = \sum_{i=1}^{m} \frac{\partial \mathcal{L}}{\partial \mathrm{Z}_i^{(2)}}}$$

Where $\mathrm{Z}_i^{(2)}$ is the ith row.

Backprop into layer 1

Propagate through weights

$$\boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{A}^{(1)}} = \frac{\partial \mathcal{L}}{\partial \mathrm{Z}^{(2)}} \cdot \left(\mathrm{W}^{(2)}\right)^{\top}}$$

With $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ we get

$$\boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{Z}^{(1)}} = \frac{\partial \mathcal{L}}{\partial \mathrm{A}^{(1)}} \odot \sigma'\left(\mathrm{Z}^{(1)}\right)}$$

Where $\odot$ is element-wise multiplication.

Gradients for $\mathrm{W}^{(1)}, \mathrm{b}^{(1)}$

Since $\mathrm{Z}^{(1)} = \mathrm{X}\mathrm{W}^{(1)} + \mathrm{b}^{(1)}$, we get that

$$\boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{W}^{(1)}} = \mathrm{X}^{\top} \frac{\partial \mathcal{L}}{\partial \mathrm{X}^{\top}}} \qquad \boxed{\frac{\partial \mathcal{L}}{\partial \mathrm{b}^{(1)}} = \sum_{i=1}^{m} \frac{\partial \mathcal{L}}{\partial \mathrm{Z}_i^{(1)}}}$$

Parameter update

With our learning rate being η we get:

$$\mathrm{W}^{(2)} \leftarrow \mathrm{W}^{(2)} - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{W}^{(2)}} \qquad \mathrm{b}^{(2)} \leftarrow \mathrm{b}^{(2)} - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{b}^{(2)}}$$

$$\mathrm{W}^{(1)} \leftarrow \mathrm{W}^{(1)} - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{W}^{(1)}} \qquad \mathrm{b}^{(1)} \leftarrow \mathrm{b}^{(1)} - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{b}^{(1)}}$$