Take Home Assignment

Harikesh Kushwaha (2021PHS7181)

February 12, 2023

Contents

1 Problem Statement

Derive systematically the steps involving the backpropagation of a neural network considered in the class with two hidden layers using the cross-entropy loss function. The activation function for each layer can be considered to be ReLU. Prepare a LaTeXreport that briefly describes the derivation of each step along with the algorithmic framework

2 Solution

The backward propagation derived in this document will follow vector notations. We'll take input as a matrix and work with matrices and vectors thorughout the detivation. This approach is taken becuase this makes the derivation both easy to derive and easy to understand. Let's start with the notations.

2.1 Notations

Here are the notations that will be used throughout the report.

> The notation is borrowed from the course Neural Networks and Deep Learning by Andrew NG with some modifications.

2.1.1 General Notations

- A lowercase letter represents a scalar. eg. a
- A lowercase bold letter represents a vector. eg. $\mathbf{a}$
- An uppercase letter represents a matrix. eg. A

2.1.2 Shape Related Notations

- m is the number of training examples.
- n_x is the number of features. (input size)
- n_y is the number of classes. (output size)
- $n^{[l]}$ is the number of neurons in layer l.

- L is the number of layers in the network. (Excluding input layer)
- $X \in \mathbb{R}^{n_x \times m}$ is the input matrix, where each column is a training example. So, X is a matrix with shape
- $Y \in \mathbb{R}^{n_y \times m}$ is the output matrix, where each column is a training example. So, Y is a matrix with shape (n_y, m).
- $\mathbf{y} \in \mathbb{R}^{1 \times m}$ is the true y-labels.
- $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ is the weight matrix of layer l. This means that $W^{[l]}$ is a matrix with shape $(n^{[l]}, n^{[l-1]})$.
- $b^{[l]} \in \mathbb{R}^{n^{[l]}}$ is the bias vector of layer l. This means that $b^{[l]}$ is a column vector with shape $(n^{[l]},)$.

2.1.3 Forward Propagation Related Notations

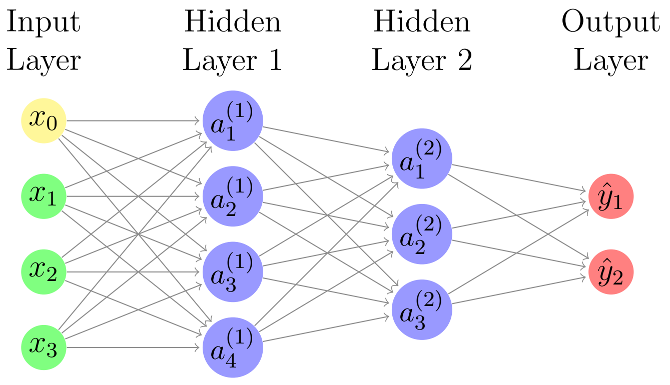
- $Z^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$ is the linear output of layer l. This means that $Z^{[l]}$ is a matrix with shape $(n^{[l]}, m)$.
- $A^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$ is the activation output of layer l. Its shape is the same as $Z^{[l]}$.
- $g^{[l]}$ is the activation function of layer l.
- $\hat{y} \in \mathbb{R}^{m}$ is the predicted output label. This is an exception where I'll use lowercase, normal font for a vector.

2.1.4 Backward Propagation Related Notations

- $\mathcal{J}(X, W, \mathbf{b}, \mathbf{y}) \in R^1$ or $\mathcal{J}(\hat{y}, \mathbf{y}) \in R^1$ is the cost function.
- $dW^{[l]}$ is the partial derivative of $\mathcal{J}$ with respect to W, $\frac{\partial \mathcal{J}}{\partial W^{[l]}}$.
- $db^{[l]}$ is the partial derivative of $\mathcal{J}$ with respect to $b^{[l]}$, $\frac{\partial \mathcal{J}}{\partial b^{[l]}}$.

2.2 Architectire of the Neural Network

With notations in place, let's look at the architecture of the neural network. Problem statement says that the neural network has two hidden layers. So, taking the third layer as output and following our notations we have the following architecture.



The neural network architecture

So, we have $L = 3$. Though the figure above shows only three fatures in the input layer, we can have any number of features. So, n_x is the number of features. The output layer has two neurons, so $n_y = 2$. The hidden layers have four and three neurons respectively, so $n^{[1]} = 4$ and $n^{[2]} = 3$. Note that these are all arbitrary numbers and the formula derived below will work for any value of n_x, n_y, $n^{[1]}$ and $n^{[2]}$. We'll also give a general recurrence relation which will work for any number of hidden layers.

2.3 Forward Propagation

Though the problem does not want forward propagation, since we are going to use it, we'll write the forward propagation too. The forward propagation is given by the following equations.

$$
\begin{aligned}
A^{[0]} &= X \\
Z^{[1]} &= W^{[1]}A^{[0]} + \mathbf{b^{[1]}} \\
A^{[1]} &= g^{[1]}(Z^{[1]}) \\
Z^{[2]} &= W^{[2]}A^{[1]} + \mathbf{b^{[2]}} \\
A^{[2]} &= g^{[2]}(Z^{[2]}) \\
Z^{[3]} &= W^{[3]}A^{[2]} + \mathbf{b^{[3]}} \\
A^{[3]} &= g^{[3]}(Z^{[3]})
\end{aligned}
$$

2.4 Loss Function

The loss function to be used is the categorical cross entropy loss function. The loss function, in vectorized form, is given by the following equation.

$$\mathcal{J}(\hat{y}, Y) = -\frac{1}{m} Y \log(\hat{y})$$

The derivative of it with respect to $\hat{y}$ which is nothing but $A^{[3]}$ is given by the following equation.

$$\frac{\partial \mathcal{J}}{\partial A^{[3]}} = Y - A^{[3]}$$

2.5 Backward Propagation

The idea behind the backward propagation (or backpropagation) as to calculate the derivative from the output layer to the input layer. The derivatives are calculated using the chain rule and they are propagated backwards.

The goal is to determine the partial derivatives of the cost function with respect to the weights and biases. Once these are determined, the weights can be updated using the update rule:

$$W^{[l]} = W^{[l]} - \alpha dW^{[l]}$$

and

$$\mathbf{b}^{[l]} = \mathbf{b}^{[l]} - \alpha d\mathbf{b}^{[l]}$$

where α is the learning rate. Let's start from the last layer.

2.5.1 Output Layer

At last layer, we need to determine the partial derivative of the cost function with respect to $W^{[3]}$ and $\mathbf{b}^{[3]}$. This can be determined using the chain rule:

$$\frac{\partial \mathcal{J}}{\partial W^{[3]}} = \frac{\partial \mathcal{J}}{\partial A^{[3]}} \frac{\partial A^{[3]}}{\partial Z^{[3]}} \frac{\partial Z^{[3]}}{\partial W^{[3]}}$$
$$\frac{\partial \mathcal{J}}{\partial \mathbf{b}^{[3]}} = \frac{\partial \mathcal{J}}{\partial A^{[3]}} \frac{\partial A^{[3]}}{\partial Z^{[3]}} \frac{\partial Z^{[3]}}{\partial \mathbf{b}^{[3]}}$$

The first term for both the weight and bias are already calculated. Note that the drivative for $\mathbf{b}^{[3]}$ is almost the same as that for $W^{[3]}$. The only difference is that the derivative of $Z^{[3]}$ with respect to $\mathbf{b}^{[3]}$ is $\mathbf{1}$ while it is $A^{[2]}$ for $W^{[3]}$ (as we will see later). That's why I'm going to derive the formula in detail for just the first one and the second one will be almost the same.

For W, we have

$$\begin{aligned}\frac{\partial A^{[3]}}{\partial Z^{[3]}} &= \frac{\partial}{\partial Z^{[3]}} g^{[3]}(Z^{[3]}) \\ &= g^{[3]\prime}(Z^{[3]})\end{aligned}$$

and

$$\begin{aligned}\frac{\partial Z^{[3]}}{\partial W^{[3]}} &= \frac{\partial}{\partial W^{[3]}} \left(W^{[3]} A^{[2]} + \mathbf{b^{[3]}}\right) \\ &= A^{[2]}\end{aligned}$$

This gives

$$\frac{\partial \mathcal{J}}{\partial W^{[3]}} = \frac{1}{m} \left(A^{[3]} - Y\right) g^{[3]\prime}(Z^{[3]}) A^{[2]T}$$

So, for the $\mathbf{b^{[3]}}$, we have

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b^{[3]}}} = \frac{1}{m} \sum \left(A^{[3]} - Y\right) g^{[3]\prime}(Z^{[3]})$$

This completes the derivation for the last layer. Before we procede to the second layer, we define a term

$$\delta^{[3]} = \frac{\partial \mathcal{J}}{\partial Z^{[3]}} = \left(A^{[3]} - Y\right) g^{[3]\prime}(Z^{[3]})$$

which will be used later.

2.5.2 Second Layer

Just for $W^{[2]}$, we have

$$\frac{\partial \mathcal{J}}{\partial W^{[2]}} = \frac{\partial \mathcal{J}}{\partial A^{[3]}} \frac{\partial A^{[3]}}{\partial Z^{[3]}} \frac{\partial Z^{[3]}}{\partial A^{[2]}} \frac{\partial A^{[2]}}{\partial Z^{[2]}} \frac{\partial Z^{[2]}}{\partial W^{[2]}}$$

The first two terms are the same as the equation for $\delta^{[3]}$. Using this and rewriting gives

$$\frac{\partial \mathcal{J}}{\partial W^{[2]}} = W^{[3]T} \delta^{[3]} g^{[2]\prime}(Z^{[2]}) A^{[1]T}$$

While for $\mathbf{b^{[2]}}$, we have

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b^{[2]}}} = \frac{\partial \mathcal{J}}{\partial A^{[3]}} \frac{\partial A^{[3]}}{\partial Z^{[3]}} \frac{\partial Z^{[3]}}{\partial A^{[2]}} \frac{\partial A^{[2]}}{\partial Z^{[2]}} \frac{\partial Z^{[2]}}{\partial \mathbf{b^{[2]}}}$$

The first two terms are the same as the equation for $\delta^{[3]}$. Using this and rewriting gives

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b^{[2]}}} = W^{[3]T} \delta^{[3]} g^{[2]\prime}(Z^{[2]})$$

This completes the derivation for the second layer. Before we procede to the first layer, we define a term

$$\delta^{[2]} = \frac{\partial \mathcal{J}}{\partial Z^{[2]}} = W^{[3]T} \delta^{[3]} g^{[2]\prime}(Z^{[2]})$$

which will be used later.

2.5.3 First Layer

Just for $W^{[1]}$, we have

$$\frac{\partial \mathcal{J}}{\partial W^{[1]}} = \frac{\partial \mathcal{J}}{\partial A^{[3]}} \frac{\partial A^{[3]}}{\partial Z^{[3]}} \frac{\partial Z^{[3]}}{\partial A^{[2]}} \frac{\partial A^{[2]}}{\partial Z^{[2]}} \frac{\partial Z^{[2]}}{\partial A^{[1]}} \frac{\partial A^{[1]}}{\partial Z^{[1]}} \frac{\partial Z^{[1]}}{\partial W^{[1]}}$$

This can be rewritten as

$$\frac{\partial \mathcal{J}}{\partial W^{[1]}} = W^{[2]T} \delta^{[2]} g^{[1]\prime}(Z^{[1]}) A^{[0]T}$$

Here $A^{[0]}$ is the input data.
While for $\mathbf{b^{[1]}}$, we have

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b^{[1]}}} = \frac{\partial \mathcal{J}}{\partial A^{[3]}} \frac{\partial A^{[3]}}{\partial Z^{[3]}} \frac{\partial Z^{[3]}}{\partial A^{[2]}} \frac{\partial A^{[2]}}{\partial Z^{[2]}} \frac{\partial Z^{[2]}}{\partial A^{[1]}} \frac{\partial A^{[1]}}{\partial Z^{[1]}} \frac{\partial Z^{[1]}}{\partial \mathbf{b^{[1]}}}$$

This can be rewritten as

$$\frac{\partial \mathcal{J}}{\partial \mathbf{b^{[1]}}} = W^{[2]T} \delta^{[2]} g^{[1]'}(Z^{[1]})$$

This completes the derivative calculation for the first layer and the whole network. Next we need to update parameters.

2.5.4 Update Equations

The update equations are

$$W^{[l]} \leftarrow W^{[l]} - \alpha \frac{\partial \mathcal{J}}{\partial W^{[l]}}$$
$$\mathbf{b^{[l]}} \leftarrow \mathbf{b^{[l]}} - \alpha \frac{\partial \mathcal{J}}{\partial \mathbf{b^{[l]}}}$$

where α is the learning rate. This completes the derivation of the back-propagation algorithm.

2.5.5 The Activation Function

During derving the above equations, we did not considered any specific format for $g^{[l]}$ and $g^{[l]'}$. The assignment asks for the ReLu activation function which is defined as

$$g^{[l]}(Z^{[l]}) = \max\left(0, Z^{[l]}\right)$$

The derivative of this is very simple and is given by

$$g^{[l]'}(Z^{[l]}) = \begin{cases} 1 & Z^{[l]} > 0 \\ 0 & Z^{[l]} \leq 0 \end{cases}$$

2.6 General Equations

General equation of backpropagation is

$$\frac{\partial \mathcal{J}}{\partial W^{[l]}} = \delta^{[l]} \mathbf{A}^{[l-1]T}$$
$$\frac{\partial \mathcal{J}}{\partial b^{[l]}} = \delta^{[l]}$$

where

$$\delta^{[l]} = (W^{[l+1]T} \delta^{[l+1]}) g^{[l]'}(Z^{[l]})$$

where $g^{[l]'}(Z^{[l]})$ is the derivative of the activation function of the l^{th} layer.

The value of $\delta^{[l]}$ is calculated using the value of $\delta^{[l+1]}$ which is calculated using the value of $\delta^{[l+2]}$ and so on. This is called the backpropagation. The value of $\delta^{[L]}$ can be calculated using the following equation

$$\delta^{[L]} = \frac{1}{m} \frac{\partial \mathcal{J}}{\partial \mathbf{A}^{[L]}} g^{[L]'}(Z^{[L]})$$

where $\frac{\partial \mathcal{J}}{\partial \mathbf{A}^{[L]}}$ is the derivative of the cost function with respect to the output of the L^{th} layer.

In our case, $\delta^{[L]}$ is

$$\delta^{[3]} = \frac{\partial \mathcal{J}}{\partial Z^{[3]}} = \left(A^{[3]} - Y\right) g^{[3]'}(Z^{[3]})$$

with

$$g^{[3]'}(Z^{[3]}) = \begin{cases} 1 & Z^{[3]} > 0 \\ 0 & Z^{[3]} \leq 0 \end{cases}$$

3 Extra: A Note on Implementation

While implementing the backpropagation algorithm, we need to be careful about the dimensions of the matrices. We also need to keep in mind when should we do elemt wise multiplication and when should we do matrix multiplication. It can be shown that for the backpropagation to work in the correct way, when implemented in numpy with the dimensions defined above, we have to use the following formulae for derivatives:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^{[l]}} &= \delta^{[l]} \mathbf{A}^{[l-1]T} \\
\frac{\partial \mathcal{L}}{\partial b^{[l]}} &= \sum_{i=1}^{m} \delta^{[l](i)} \\
\delta^{[l]} &= (W^{[l+1]T} \odot \delta^{[l+1]}) g^{[l]\prime}(Z^{[l]}) \\
\delta^{[L]} &= \frac{1}{m} \frac{\partial \mathcal{L}}{\partial \mathbf{A}^{[L]}} g^{[L]\prime}(Z^{[L]})
\end{aligned}$$

here $\odot$ is the dot product.