Mathematical Comprehension of Neural Networks

Introduction

Opening Statement

In our modern world technology surrounds us, it automates and makes life easier. As years pass, AI researchers constantly develop new ways of interpreting data, more specifically; more advanced methods of understanding the world around us. Currently, neural networks are being used not only understand and predict events but they are being used to further develop artificial intelligence. This being said it is left evident that neural networks are an important part of the modern world.

Idea

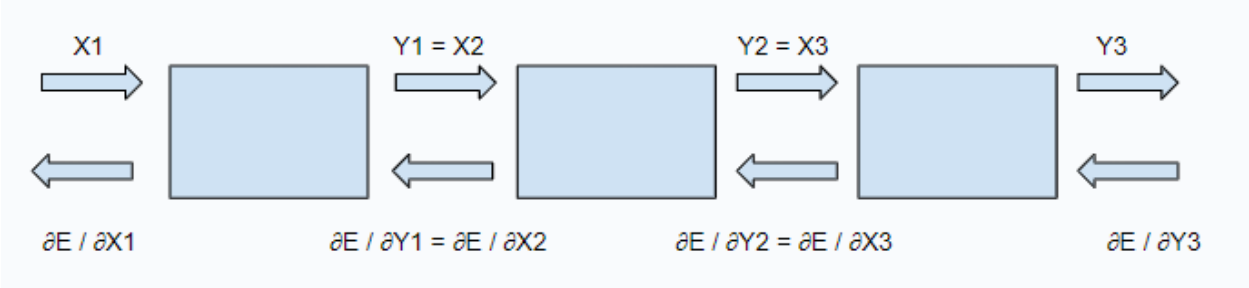
This report will highlight the mathematical process that lies within the creation and implementation of this neural network, not only looking at the calculus and mathematical aspects but, why each part is important and how it contributes towards the whole idea of a neural network.

What is a Neural Network

In essence, a neural network is a machine learning algorithm / model that makes predictions. The way it works is by training itself on data and then based on that training it can make predictions on unseen data. There are many variations of neural networks, each specified to address a different type of problem, the neural network which will be implemented during this report is the base foundation of all other types of neural networks.

Implementation Design

What you see below is the general design of the neural network. You can see the forward propagation and the backward propagation. The main takeaway is that the output of a layer is the input of another.



Chain Rule

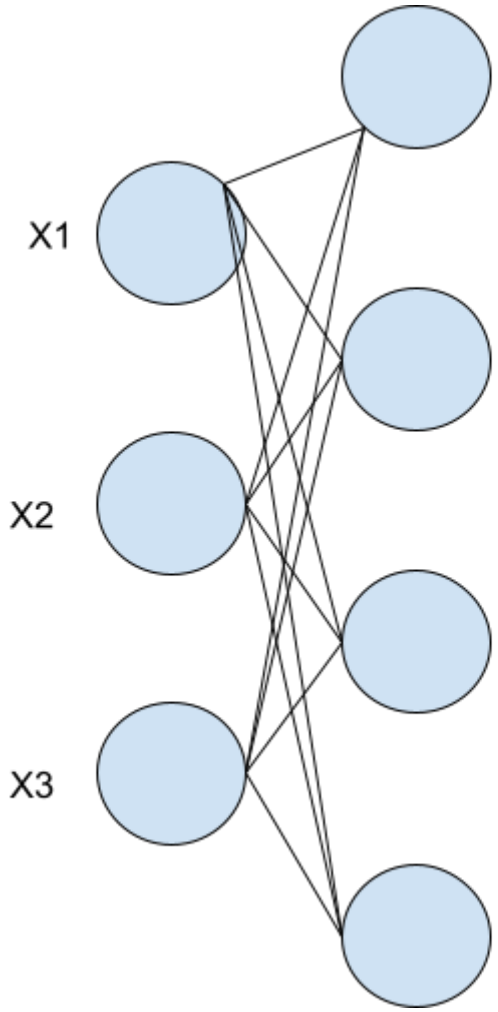
Now we have to find the derivative of the error with respect to the output or input depending on that section, you can find the basic implementation of the chain rule below.

$$\frac{\partial E}{\partial W} = \frac{\partial E}{\partial Y} \times \frac{\partial E}{\partial W}$$

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} \times \frac{\partial E}{\partial X}$$

Forward Propagation

In a neural network forward propagation is in essence the way that the weighted sum of the inputs is found, then that gives an output which is then further passed on through the layers.



$$= (X_1 \cdot W_{11}) + (X_2 \cdot W_{21}) + (X_3 \cdot W_{31})$$

$$= (X_1 \cdot W_{21}) + (X_2 \cdot W_{22}) + (X_3 \cdot W_{23})$$

$$= (X_1 \cdot W31) + (X_2 \cdot W32) + (X_3 \cdot W33)$$

$$= (X_1 \cdot W_{41}) + (X_2 \cdot W_{42}) (X_3 \cdot W_{43})$$

Backward Propagation

Backward propagation is a very common algorithm in machine learning, the idea of it is to backward propagate the error/output of the forward propagation in order to update the weights and biases of a neural network. During this section we will be finding the equation for the weights, biases, and for the inputs using the chain rule.

Finding the equation for X:

What you see below is the multiplication of the derivative of the error with respect to y1 by the derivative of the error with respect to x1. Then every time y increases and x stays the same. We come to find that when we express this as a matrix multiplication we get a final answer.

$$\frac{\partial E}{\partial X1} = \frac{\partial E}{\partial Y_1} \cdot \frac{\partial E}{\partial X_1} + \frac{\partial E}{\partial Y_2} \cdot \frac{\partial E}{\partial X_1} + \frac{\partial E}{\partial Y_3} \cdot \frac{\partial E}{\partial X_1}$$

$$W11 \qquad 0 \qquad 0$$

$$\frac{\partial E}{\partial X} = W \cdot \frac{\partial E}{\partial Y}$$

Finding the equation for W:

$$\frac{\partial E}{\partial W21} = \frac{\partial E}{\partial Y_1} \cdot \frac{\partial E}{\partial W_{21}} + \frac{\partial E}{\partial Y_2} \cdot \frac{\partial E}{\partial W_{21}} + \frac{\partial E}{\partial Y_3} \cdot \frac{\partial E}{\partial W_{21}}$$

$$0 \qquad X1 \qquad 0$$

$$\frac{\partial E}{\partial W} = X \cdot \frac{\partial E}{\partial Y}$$

Finding the equation for B:

$$\frac{\partial E}{\partial B_1} = \frac{\partial E}{\partial Y_1} \cdot \frac{\partial E}{\partial B_1} + \frac{\partial E}{\partial Y_2} \cdot \frac{\partial E}{\partial B_1} + \frac{\partial E}{\partial Y_3} \cdot \frac{\partial E}{\partial B_1}$$

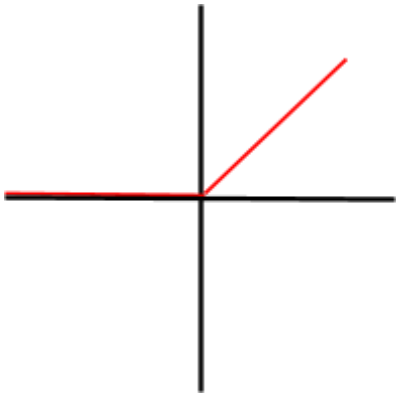
$$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$

Collection of Equations:

$$\frac{\partial E}{\partial X} = W \cdot \frac{\partial E}{\partial Y}$$
$$\frac{\partial E}{\partial W} = X \cdot \frac{\partial E}{\partial Y}$$
$$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$

Activation Functions

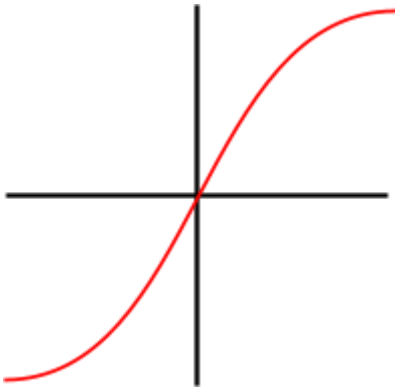
In a neural network activation functions server a very important role. The idea of them is to remove the linearity of the outputs. There are many activation functions including tanh and sigmoid which all carry their advantages and disadvantages but during this project we will be using ReLU for the hidden layers and Softmax for the output layers.



Rectified Linear Unit

The idea of ReLU is that when a number is equal or less than zero then the output is zero, otherwise it is linear.

$$Y > 0 = Y$$
$$Y \leq 0 = 0$$



Softmax Activation Function

The softmax activation function basically converts outputs from a neural network into an array of probabilities.

$$\frac{exp(y)}{\sum exp(y)}$$

Calculating the Error

As you can probably figure, calculating the error of the model is essential, as it is part of not only finding the derivatives for backward propagation but also to investigate the performance of a model. There are many different methods of calculating the loss but we will be using sparse categorical crossentropy as it is optmized for tasks like classification.

$$-\sum_{i} y_i \cdot log(\widehat{y_i})$$

The equation above represents the negative sum of the true values multiplied by the logarithmic function of the predicted values.

Gradient Descent

Gradient descent is the method in which the weights and the biases get updated. What is going on here is first we are finding the weights gradient by finding the weighted sum of X transposed and the output gradient. Then, we are finding the input gradient by finding the weighted sum of the weight and the output gradient. Finally, we are updating our parameters using gradient descent.

Gradient Descent Mathematical Visualisation

$$d_1 = (X \cdot \frac{\partial E}{\partial Y})_{What\ you\ find\ above\ is\ the\ weighted\ sum}$$

$$d_2 = (W \cdot \frac{\partial E}{\partial Y})_{What\ you\ find\ above\ is\ the\ weighted\ sum}$$

$$w = w - ld_1$$

$$b = b - l\frac{\partial E}{\partial Y}$$