Mathematics of a Feedforward Neural Network with Backpropagation

Alejandro Gomez Rivas

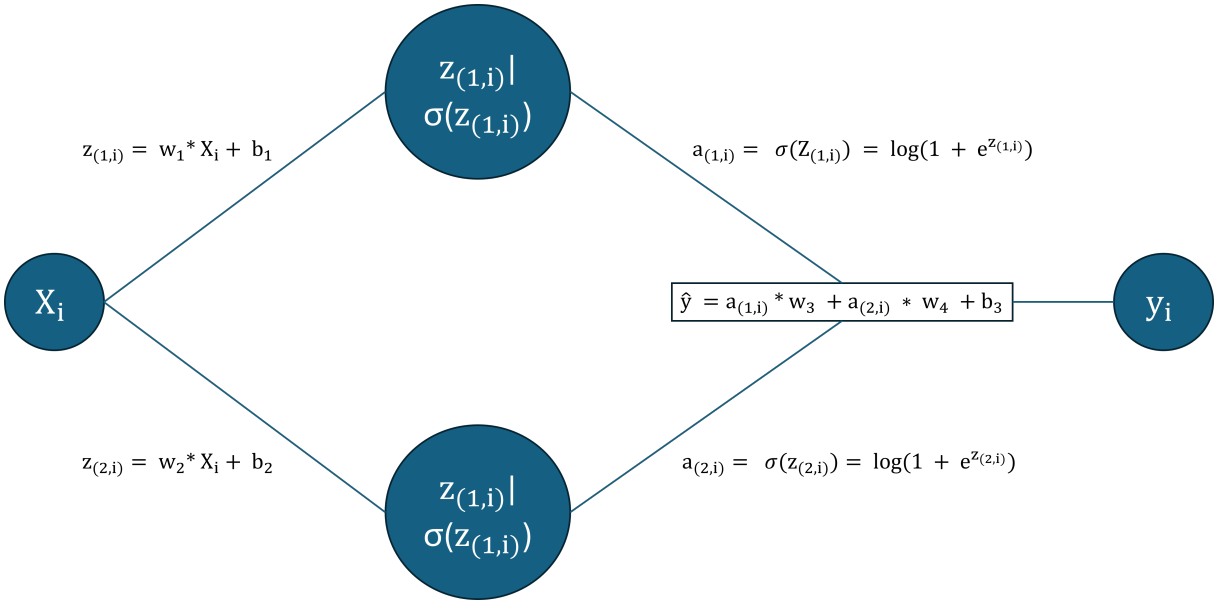
Abstract

This paper describes the mathematics of a simple feedforward neural network. The neural network consists of two neurons, which are activated by the softplus function, and then the prediction is made.

Contents

1. Neural Network

This diagram shows a simple feedforward neural network with one hidden layer. The input goes through two neurons in the hidden layer, each applying a non-linear activation function to its weighted input. The outputs of these hidden neurons are then combined linearly, added to a bias, and passed forward to produce the final prediction. Essentially, it's a basic two-layer network transforming inputs through activation functions to generate an output.



Symbol	Description
X_i	i-th input sample dataset
w_k	Weight parameters
b_k	Bias parameters
$z_{(n,j)}$	neuron
$a_{(n,j)}$	neuron activation
$\hat{y}_i$	Output prediction
y_i	Output observed
L_i	Loss for input x_i
$\sum_{i=1}^{N} L_i$	Sum of losses over all inputs

2. All the Formulas in the Neural Network

First, the network takes the input data, which is just your training samples. Each sample is fed into the neurons. Each neuron first does a simple weighted sum of the inputs plus a bias, this is the linear step. Then, to make the output more flexible and able to model complex patterns, an activation function is applied, which transforms the linear output into a non-linear one.

Once the neurons have produced their activated outputs, these outputs are combined using another set of weights and a bias to produce the final prediction. This final step essentially sums the contributions of all neurons to generate the network's guess for the target value.

Finally, the loss function measures how far off the network's predictions are from the actual observed values. It calculates the difference for every training sample, squares it to penalize larger errors, and sums everything up. The goal of training is to adjust the weights and biases so that this loss becomes as small as possible, meaning the predictions are as close as possible to the true values.

2.1. Input data

- X_{input} is the input data that for training is used. Each data sample has a index i

$$X_{\text{input}} = \text{data sample} = X_i$$

2.2. Define neural network formulas

- Xinput data will be subsituted into the first nueron. The first neuron is a linear equation with a slope of w1 and an bias of b1, the output is described as X(1,i). This is the linear transformation

$$z_{(1,i)} = X_i w_1 + b_1$$

- The same is done for the second nueron. this is the linear transformation

$$z_{(2,i)} = X_i w_2 + b_2$$

2.3. activation function formulas

- The first neuron is described as a linear equetion. An activation function is applied to make it non linear. The output of the linear equation is the input of the activation function. the activation function gives a non linear output. This is the activation function

$$a_{(1,i)} = f(z_{(1,i)}) = \log(1 + e^{z_{(1,i)}})$$

- The same is done for the second nueron. This is the activation function

$$a_{(2,i)} = f(z_{(2,i)}) = \log(1 + e^{z_{(2,i)}})$$

2.4. Prediction formula

- After the activation formula, a new weight will be added to the activation outputs. The prediction for the output is the sum of the neurons output*weight and a bias. The bias is added to shift the final shape of the curve. This is the prediction formula

$$\hat{y}_i = a_{(1,i)} w_3 + a_{(2,i)} w_4 + b_3$$

2.5. Loss function formula

- Loss function describes how far the predicted value is from the observed value. The residual is the diffrence of the observed value and the predicted value. there will be a prediction for every training data sample. The total loss function is the sum of all the residuals. The loss function is minimized optimized if the prediction output is the same as the observed output

$$\sum_{i=1}^{n} L_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

3. Computing the Derivatives

All the parameters $w_1, b_1, w_2, b_2, w_3, w_4, b_3$ need to be optimized with respect to the loss function. The parameters are not directly connected with the loss function; therefore, the chain rule needs to be applied.

To train a neural network, all the weights and biases need to be adjusted to minimize the loss. However, these parameters don't directly control the loss—they influence it through the network's computations. That's why we use derivatives and the chain rule: it lets us see how changing each weight or bias affects the final error.

For each parameter, we trace the effect step by step: how it changes the linear combination in its neuron, how that affects the neuron's activation, how the activated output contributes to the prediction, and finally how the prediction changes the loss. This chain of effects is captured by the derivative, which tells us both the direction and size of the change needed.

The process is repeated for every training sample, and the results are summed up to give the overall derivative with respect to each parameter. Once we have these derivatives, they guide the optimization process, telling us how to update the weights and biases to make the network's predictions closer to the actual outputs.

3.1. Overview formulas

3.1.1. Derivative of the neurons formulas with respect to the loss function

- Derivative of the loss function (L_i^2) with respect to the first weight (w_1)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_1} = \sum_{i=1}^{n} \frac{dL_i^2}{dw_1} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(1,i)}} \cdot \frac{da_{(1,i)}}{dz_{(1,i)}} \cdot \frac{dz_{(1,i)}}{dw_1}$$

- Derivative of the loss function (L_i^2) with respect to the first bias (b_1)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_1} = \sum_{i=1}^{n} \frac{dL_i^2}{db_1} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(1,i)}} \cdot \frac{da_{(1,i)}}{dz_{(1,i)}} \cdot \frac{dz_{(1,i)}}{db_1}$$

- Derivative of the loss function (L_i^2) with respect to the second weight (w_2)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_2} = \sum_{i=1}^{n} \frac{dL_i^2}{dw_2} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(2,i)}} \cdot \frac{da_{(2,i)}}{dz_{(2,i)}} \cdot \frac{dz_{(2,i)}}{dw_2}$$

- Derivative of the loss function (L_i^2) with respect to the second bias (b_2)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_2} = \sum_{i=1}^{n} \frac{dL_i^2}{db_2} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(2,i)}} \cdot \frac{da_{(2,i)}}{dz_{(2,i)}} \cdot \frac{dz_{(2,i)}}{db_2}$$

3.1.2. Derivative of the prediction formula with respect to the loss function

- Derivative of the loss function (L_i^2) with respect to the third weight (w_3)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_3} = \sum_{i=1}^{n} \frac{dL_i^2}{dw_3} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{dw_3}$$

- Derivative of the loss function (L_i^2) with respect to the fourth weight (w_4)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_4} = \sum_{i=1}^{n} \frac{dL_i^2}{dw_4} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{dw_4}$$

- Derivative of the loss function (L_i^2) with respect to the third bias (b_3)

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_3} = \sum_{i=1}^{n} \frac{dL_i^2}{db_3} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{db_3}$$

3.2. Derivative of the loss function with respect to weight 1

- Derivative of the loss function (L_i^2) with respect to the first weight (w_1) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_1} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(1,i)}} \cdot \frac{da_{(1,i)}}{dz_{(1,i)}} \cdot \frac{dz_{(1,i)}}{dw_1}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to activation ($a_{(1,i)}$)

$$\frac{d\hat{y}_i}{da_{(1,i)}} = w_3$$

- Derivative of the activation function with respect to linear output ($z_{(1,i)}$)

$$\frac{da_{(1,i)}}{dz_{(1,i)}} = \frac{e^{z_{(1,i)}}}{1 + e^{z_{(1,i)}}}$$

- Derivative of the linear transformation with respect to w_1

$$\frac{dz_{(1,i)}}{dw_1} = X_i$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_1} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot w_3 \cdot \frac{e^{z_{(1,i)}}}{1 + e^{z_{(1,i)}}} \cdot X_i\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_1} = -2\Big[(y_1 - \hat{y}_1)w_3 \frac{e^{z_{(1,1)}}}{1 + e^{z_{(1,1)}}} X_1 + (y_2 - \hat{y}_2)w_3 \frac{e^{z_{(1,2)}}}{1 + e^{z_{(1,2)}}} X_2 + \cdots + (y_n - \hat{y}_n)w_3 \frac{e^{z_{(1,n)}}}{1 + e^{z_{(1,n)}}} X_n\Big]$$

3.3. Derivative of the loss function with respect to bias 1

- Derivative of the loss function (L_i^2) with respect to the first bias (b_1) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_1} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(1,i)}} \cdot \frac{da_{(1,i)}}{dz_{(1,i)}} \cdot \frac{dz_{(1,i)}}{db_1}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to activation ($a_{(1,i)}$)

$$\frac{d\hat{y}_i}{da_{(1,i)}} = w_3$$

- Derivative of the activation function with respect to linear output ($z_{(1,i)}$)

$$\frac{da_{(1,i)}}{dz_{(1,i)}} = \frac{e^{z_{(1,i)}}}{1 + e^{z_{(1,i)}}}$$

- Derivative of the linear transformation with respect to b_1

$$\frac{dz_{(1,i)}}{db_1} = 1$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_1} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot w_3 \cdot \frac{e^{z_{(1,i)}}}{1 + e^{z_{(1,i)}}} \cdot 1\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_1} = -2\Big[(y_1 - \hat{y}_1)w_3 \frac{e^{z_{(1,1)}}}{1 + e^{z_{(1,1)}}} + (y_2 - \hat{y}_2)w_3 \frac{e^{z_{(1,2)}}}{1 + e^{z_{(1,2)}}} + \cdots + (y_n - \hat{y}_n)w_3 \frac{e^{z_{(1,n)}}}{1 + e^{z_{(1,n)}}}\Big]$$

3.4. Derivative of the loss function with respect to weight 2

- Derivative of the loss function (L_i^2) with respect to the second weight (w_2) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_2} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(2,i)}} \cdot \frac{da_{(2,i)}}{dz_{(2,i)}} \cdot \frac{dz_{(2,i)}}{dw_2}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to activation ($a_{(2,i)}$)

$$\frac{d\hat{y}_i}{da_{(2,i)}} = w_4$$

- Derivative of the activation function with respect to linear output ($z_{(2,i)}$)

$$\frac{da_{(2,i)}}{dz_{(2,i)}} = \frac{e^{z_{(2,i)}}}{1 + e^{z_{(2,i)}}}$$

- Derivative of the linear transformation with respect to w_2

$$\frac{dz_{(2,i)}}{dw_2} = X_i$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_2} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot w_4 \cdot \frac{e^{z_{(2,i)}}}{1 + e^{z_{(2,i)}}} \cdot X_i\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_2} = -2\Big[(y_1 - \hat{y}_1)w_4 \frac{e^{z_{(2,1)}}}{1 + e^{z_{(2,1)}}} X_1 + (y_2 - \hat{y}_2)w_4 \frac{e^{z_{(2,2)}}}{1 + e^{z_{(2,2)}}} X_2 + \cdots + (y_n - \hat{y}_n)w_4 \frac{e^{z_{(2,n)}}}{1 + e^{z_{(2,n)}}} X_n\Big]$$

3.5. Derivative of the loss function with respect to bias 2

- Derivative of the loss function (L_i^2) with respect to the second bias (b_2) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_2} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{da_{(2,i)}} \cdot \frac{da_{(2,i)}}{dz_{(2,i)}} \cdot \frac{dz_{(2,i)}}{db_2}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to activation ($a_{(2,i)}$)

$$\frac{d\hat{y}_i}{da_{(2,i)}} = w_4$$

- Derivative of the activation function with respect to linear output ($z_{(2,i)}$)

$$\frac{da_{(2,i)}}{dz_{(2,i)}} = \frac{e^{z_{(2,i)}}}{1 + e^{z_{(2,i)}}}$$

- Derivative of the linear transformation with respect to b_2

$$\frac{dz_{(2,i)}}{db_2} = 1$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_2} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot w_4 \cdot \frac{e^{z_{(2,i)}}}{1 + e^{z_{(2,i)}}} \cdot 1\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_2} = -2\Big[(y_1 - \hat{y}_1)w_4 \frac{e^{z_{(2,1)}}}{1 + e^{z_{(2,1)}}} + (y_2 - \hat{y}_2)w_4 \frac{e^{z_{(2,2)}}}{1 + e^{z_{(2,2)}}} + \cdots + (y_n - \hat{y}_n)w_4 \frac{e^{z_{(2,n)}}}{1 + e^{z_{(2,n)}}}\Big]$$

3.6. Derivative of the loss function with respect to weight 3

- Derivative of the loss function (L_i^2) with respect to the third weight (w_3) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_3} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{dw_3}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to weight (w_3)

$$\frac{d\hat{y}_i}{dw_3} = \frac{d}{dw_3}\left(a_{(1,i)}w_3 + a_{(2,i)}w_4 + b_3\right) = a_{(1,i)}$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_3} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot a_{(1,i)}\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_3} = -2\Big[(y_1 - \hat{y}_1)a_{(1,1)} + (y_2 - \hat{y}_2)a_{(1,2)} + \cdots + (y_n - \hat{y}_n)a_{(1,n)}\Big]$$

3.7. Derivative of the loss function with respect to weight 4

- Derivative of the loss function (L_i^2) with respect to the fourth weight (w_4) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_4} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{dw_4}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to weight (w_4)

$$\frac{d\hat{y}_i}{dw_4} = \frac{d}{dw_4}\left(a_{(1,i)}w_3 + a_{(2,i)}w_4 + b_3\right) = a_{(2,i)}$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_4} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot a_{(2,i)}\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{dw_4} = -2\Big[(y_1 - \hat{y}_1)a_{(2,1)} + (y_2 - \hat{y}_2)a_{(2,2)} + \cdots + (y_n - \hat{y}_n)a_{(2,n)}\Big]$$

3.8. Derivative of the loss function with respect to bias 3

- Derivative of the loss function (L_i^2) with respect to the third bias (b_3) using the chain rule

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_3} = \sum_{i=1}^{n} \frac{dL_i^2}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{db_3}$$

- Derivative of the loss function with respect to prediction ($\hat{y}_i$)

$$\frac{dL_i^2}{d\hat{y}_i} = -2(y_i - \hat{y}_i)$$

- Derivative of the prediction formula with respect to bias (b_3)

$$\frac{d\hat{y}_i}{db_3} = \frac{d}{db_3}\left(a_{(1,i)}w_3 + a_{(2,i)}w_4 + b_3\right) = 1$$

- Full derivative substituted

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_3} = \sum_{i=1}^{n} \left[-2(y_i - \hat{y}_i) \cdot 1\right]$$

- Expanded sum over all data points

$$\frac{d\left(\sum_{i=1}^{n} L_i^2\right)}{db_3} = -2\Big[(y_1 - \hat{y}_1) + (y_2 - \hat{y}_2) + \cdots + (y_n - \hat{y}_n)\Big]$$

4. Softplus Activation Function