Neural Networks

Oliver Temple

March 8, 2022

What a Neural Network Is

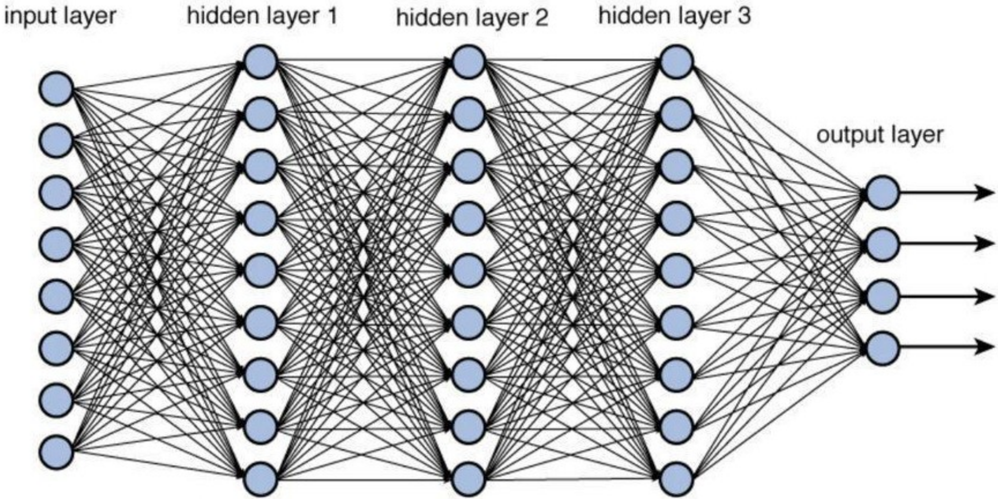


Figure: Representation of a neural network

Types of Neural Network

- Perceptron
- Feed Forward Neural Network
- Multi-layer Perceptron

Types of Neural Network

Perceptron

$$\mathbf{y} = \sigma(\mathbf{W} \cdot \mathbf{x} + b) \qquad (1)$$

Where **y** is the output, **W** is a vector of the weights, **x** is a vector of the inputs, b is the bias and σ is the activation function.

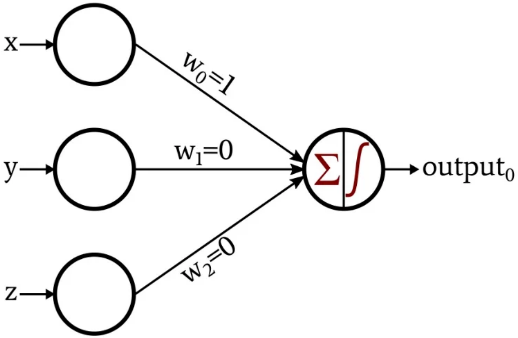


Figure: Perceptron

Types of Neural Network

Feed Forward Neural Network

$$\mathbf{y}_n = \sigma(\mathbf{W}_n \cdot \mathbf{y}_{n-1} + \mathbf{b}_n) \qquad (2)$$

Where $\mathbf{y}_n$ is the output vector of the nth layer, $\mathbf{W}_n$ is the weight matrix of the nth layer, $\mathbf{y}_{n-1}$ is the output vector of the previous layer, $\mathbf{b}_n$ is the bias vector of the nth layer, and σ is the activation function.

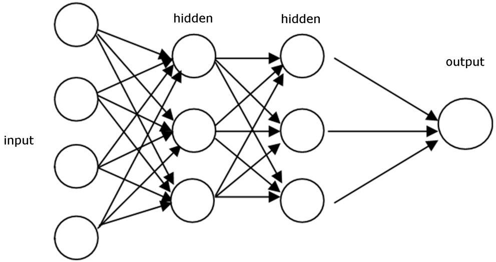


Figure: Feed Forward Neural Network

Types of Neural Network

Multi-layer Perceptron

$$\mathbf{y}_n = \sigma(\mathbf{W}_n \cdot \mathbf{y}_{n-1} + \mathbf{b}_n) \tag{3}$$

Where $\mathbf{y}_n$ is the output vector of the nth layer, $\mathbf{W}_n$ is the weight matrix of the nth layer, $\mathbf{y}_{n-1}$ is the output vector of the previous layer, $\mathbf{b}_n$ is the bias vector of the nth layer, and σ is the activation function.

How Neural Networks Work

- Collection of interconnected neurons
- Each neuron has a bias
- Each connection has a weight

How Neural Networks Work

Structure

- Input Layer
- Output Layer
- Hidden Layers

How Neural Networks Work

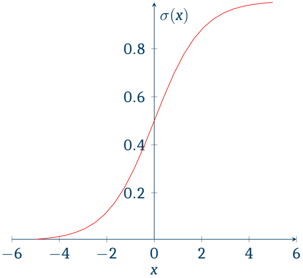
Forward Propagation

- Inputs are passed through the network to get a prediction
- $\mathbf{y}_n = \sigma(\mathbf{W}_n \cdot \mathbf{y}_{n-1} + \mathbf{b}_n)$
- Where $\mathbf{y}_n$ is the output vector of the nth layer, $\mathbf{W}_n$ is the weight matrix of the nth layer, $\mathbf{y}_{n-1}$ is the output vector of the previous layer, $\mathbf{b}_n$ is the bias vector of the nth layer, and σ is the activation function.

How Neural Networks Work

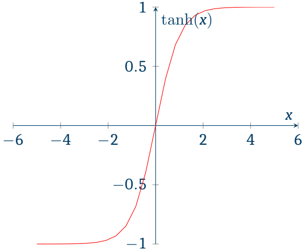
Activation Functions

Sigmoid



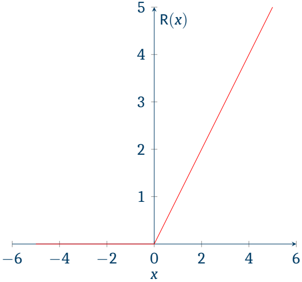
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

tanh



$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

ReLU



$$\mathrm{ReLU}(x) = \max(0, x)$$

How Neural Networks Work

Loss Functions

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

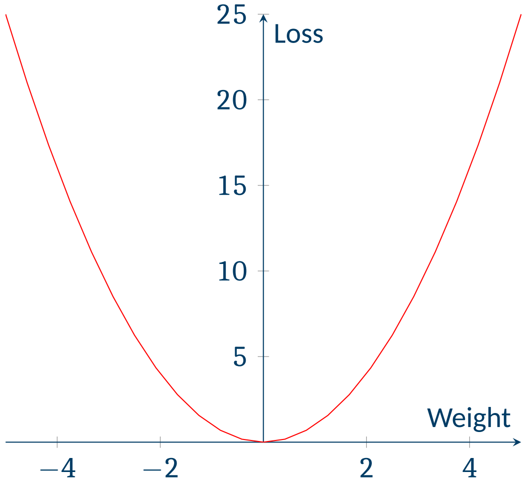
$$(CEL) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \times \log(\hat{y}_i))$$

Where n is the number if sample, y_i is the desired output of the network and $\hat{y}_i$ is the actual output of the network.

How Neural Networks Work

Backpropagation

- Data moves backwards through the network
- Weights and biases adjusted
- Loss must be calculated
- Weights and biases can be changed proportionally to the gradient
- Process is repeated



How Neural Networks Work

Training

- Forward Propagation
- Backpropagation
- Repeat

Coding a Neural Network

- Data
- Defining the Model
- Training the Model
- Testing the Model

Coding a Neural Network

Data

- Data is very important
- MNIST dataset
- Dataset must be split into ‘test’ and ‘train’ subsets
- 50000 images in the ‘train’ subset
- 10000 images in the ‘test’ subset

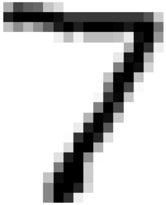


Figure: MNIST Example

Defining the Model

Layers

- 784 input neurons
- 350 hidden neurons
- 10 output neurons

Defining the Model

Loss Function

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{4}$$

Defining the Model

Activation Function

$$2\sigma'(x) = x(1 - x) \tag{6}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

Coding a Neural Network

Training the Model

- Split the data into equal sized batches
- Forward Propagation for the whole epoch
- Backpropagation after each epoch

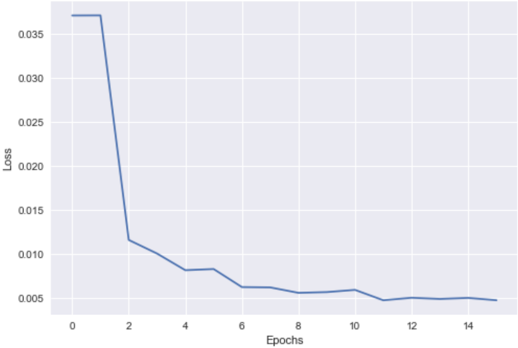


Figure: Loss over epochs

Coding a Neural Network

Testing the Model

- Test the model on unseen data
- Run the test dataset through the model
- Calculate accuracy by keeping track of how many correct predictions were made
- 95.32% accuracy

Coding a Neural Network

Testing the Model

- Input labels: 8, 5, 8, 9, 1, 9, 7, 2, 8
- predictions: 5, 5, 8, 9, 1, 9, 7, 2, 8
- 88% accuracy

Figure: Inputs for example predictions

Conclusion

- Complex models used for many different tasks
- Easier and quicker to use a library
- I have gained knowledge of neural networks