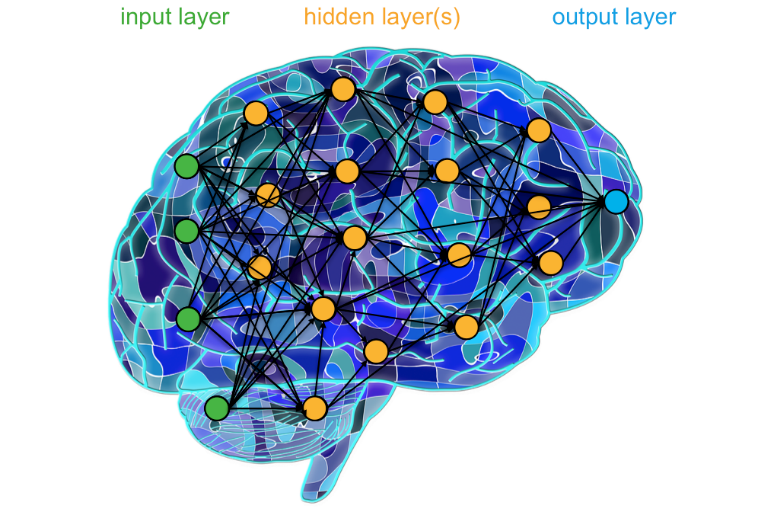
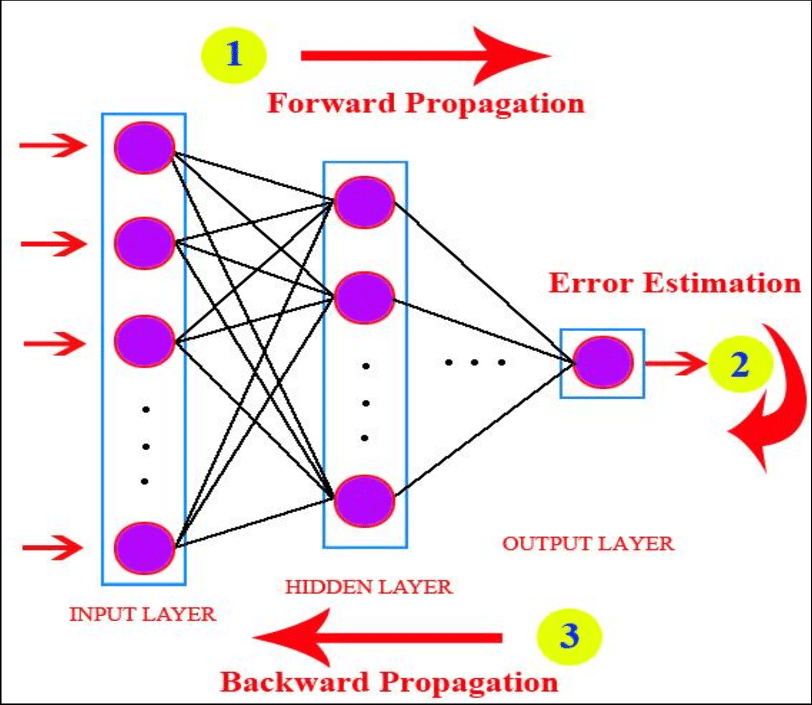
**Artificial Neural Networks**

**Artificial Neural Networks** are a mathematical model to mimics the brain of a human as it uses the concept of neurons to send information.



Every neural network has **hidden layers** to process the data from the **input layer** until it reaches the **output layer**. Every layer has some number of nodes like **neurons**, and they are connected to each node in the next layer. The neural network can learn from the data by adjusting its weights. the weights are the lines that connect each node with the other nodes.

there are two stages in the neural network, first called **forward propagation**, and it means the data are processed from the input layer to produce the output.



**Example:**

To transmit the input vector to the next layer, it is multiplied by the weights of this layer. In the figure below, to forward propagate to the node (**h1)** we need to multiply the output of (**i1**) with (**w1**) + the output of (**i2**) times (**w3**).

A diagram of a diagram

Description automatically generated

Output (**h1**) after taking it’s input as we mentioned before using its **activation function** to produce the output that will be used in the hidden. by this mechanism, the values propagated until they reach the output layer. the output layer is the prediction of the neural network to some given values.

**so, what is activation function?**

**activation function** helps the neural network to learn non-linearity and complex patterns. As discussed earlier in forward propagation, each neuron takes the input from the previous layers and then multiplies them by their weights, you can think of activations functions as function graphs that can be transformed into different shapes to fit the training examples. The transformations are done by multiplication and addition of weights.

There are different types of activation function like **relu, linear, and SoftMax** activation functions. In each layer, the type of the activation function must be specified.

**RELU:**

**relu** stands for Rectified Linear Unit. It returns the output if it is more than zero and zero elsewhere. It solves problems that arise in other activation functions such as saturation and sensitivity. The graph of the function is shown below.

(we use it)

**A blue line graph with numbers

Description automatically generated** A black background with a black square

Description automatically generated with medium confidence

**SoftMax:**

SoftMax is used in multiclassification tasks as it outputs a predicted probability given the weights in the neural network. Usually, it exists in the final layer of the network. The predicted probability is not the actual probability of the classes, it is just an estimation of it. the equation of SoftMax is shown below:

A mathematical equation with a number of symbols

Description automatically generated with medium confidence

For each class, the probability is given by exponentiation of the numerical prediction of the class in the last layer divided by the summation of the exponentiations of the other classes.

**Linear activation function:**

The linear activation function is just the identity function f(x) = x. it maps the input to the output. usually, it is used as the last layer in regression neural networks.

A graph with a green line

Description automatically generated

**Loss function**

The loss function or error function is the function that neural networks use to know if its prediction is true or false. Minimizing the loss function helps the neural network to learn from the training examples by adjusting its weights to make its predictions accurate in the future. The loss function used in multi-classification problems called categorical cross-entropy.

**categorical cross-entropy**

Note :(it used for multi-classification. In our case we use **binary cross-entropy** A white lines on a black background

Description automatically generated)  
It is SoftMax activation plus cross-entropy. The equation of cross-entropy is shown below:

where it is the actual truth of the classification and the prediction probability of the neural network. In the categorical cross-entropy, the final layer passed to SoftMax then the cross-entropy.

As in multi-classification, one class is only true, and its truth is one and the rest.

A diagram of a mathematical equation

Description automatically generatedis zero. Therefore, its loss only remained making the equation.

A black and white image of a log

Description automatically generated

by taking derivatives to this function and minimizing it, the neural network can update the weights.

**Implementation Details**

**Dataset Preparation**

1. **Data Collection**: The dataset used for this project was obtained from Kaggle.com.
2. **Data Cleaning**: Pre-processing steps involved handling missing values, encoding categorical variables, and scaling numerical features.

**Model Architecture**

**Neural Network Structure**: The artificial neural network (ANN) architecture consisted of 4 layers, including input, hidden, and output layers.

**Activation Functions:** ReLU (Rectified Linear Unit) was used as the activation function for hidden layers, while a sigmoid function was employed for the output layer to obtain binary predictions.

**Optimization Algorithm**: The Adam optimizer was utilized to minimize the binary cross-entropy loss function during training.

**Regularization Techniques**: L2 regularization was applied to prevent overfitting.

**Model Evaluation**

**Performance Metrics:** Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score on the test set.

**Confusion Matrix:** A confusion matrix was generated to analyze the model's true positive, true negative, false positive, and false negative predictions.

**ROC Curve Analysis:** The Receiver Operating Characteristic (ROC) curve was plotted to assess the trade-off between true positive rate and false positive rate at different classification thresholds.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  | **Data Description:** | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | ID | Customer ID | |  |  |  |  |  |  |  |
|  | Age | Customer's age in completed years | | | |  |  |  |  |  |
|  | Experience | #years of professional experience | | | |  |  |  |  |  |
|  | Income | Annual income of the customer ($000) | | | | |  |  |  |  |
|  | ZIPCode | Home Address ZIP code. | | |  |  |  |  |  |  |
|  | Family | Family size of the customer | | | |  |  |  |  |  |
|  | CCAvg | Avg. spending on credit cards per month ($000) | | | | | |  |  |  |
|  | Education | Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional | | | | | | | |  |
|  | Mortgage | Value of house mortgage if any. ($000) | | | | |  |  |  |  |
|  | Mortgage | Value of house mortgage if any. ($000) | | | | |  |  |  |  |
|  | Personal Loan | Did this customer accept the personal loan offered in the last campaign? | | | | | | | |  |
|  | Securities Account | Does the customer have a securities account with the bank? | | | | | | |  |  |
|  | CD Account | Does the customer have a certificate of deposit (CD) account with the bank? | | | | | | | |  |
|  | Online | Does the customer use internet banking facilities? | | | | | |  |  |  |
|  | Credit Card | Does the customer use a credit card issued by Universal Bank? | | | | | | |  |  |

**Dataset Description:**  
The file Bank.xls contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

**Results**

**Accuracy:**

accuracy: 0.9851 - loss: 0.0710

**Classification report:**

**A screenshot of a graph

Description automatically generated**

**Confusion Matrix:**

**A number in a row

Description automatically generated with medium confidence**