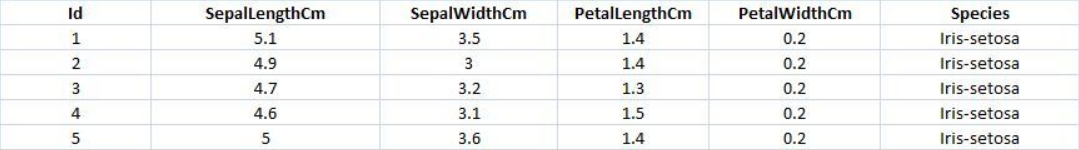
**Sample Data**:

The Iris dataset includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.



**Key Terminologies and Parameters:**

**1. Neural Network:**

A neural network is a computer system inspired by the brain. It uses interconnected nodes to process information and learn from data, much like how our brains learn. This allows them to tackle complex tasks like image recognition and speech understanding. In the mentioned case it can be used for classification of the iris flower. A neural network has input layers, hidden layers and output layers.

**2. Neuron:**

Neurons are nodes through which data and computations flow. The neurons receive one or many input signals , these signals can be raw data or processed data from the previous layer neurons. They perform the calculations and send output to the next layer.

**3. Layer:**

A neural network has three layers- input layer, hidden layer and output layer.

**Input layer:**

It is the first layer in the neural network and it receives the input data.In this case, we need 4 neurons since we have 4 input features [sepal length, sepal width, petal length, and petal width]

**Hidden layer:**

Intermediate layers between input and output layers. They perform transformations on the input data to find patterns. It can have as many neurons as we desire.

**Output layer:**

It is the final layer of the neural network and it produces the output. For the iris dataset we need 3 neurons, one for each species [Setosa, Versicolor, Virginica].

**7. Convolutional Layer:**

A convolutional layer is the fundamental component of the Convolutional Neural Network. It is mainly used for image and spatial data processing. It is used to detect patterns, edges and textures in the images.

**8. Convolutional Neural Network:**

It is a type of deep learning model that is mainly used for processing the images. It is used especially for image detection, classification and facial recognition. It uses filters that slide over the image to extract the features in the images.

**9. Recurrent Neural Network:**

It is a type of deep learning model that is mainly used for handling sequential data and capturing temporal dependencies. They are mainly suited for text generation since they are based on context learning. In this model the output of a neuron is given as the input of another neuron, so it can track the history of the conversation and use them in future data processing.

**10. Activation Function:**

It is a mathematical function applied to the output of the neuron to introduce non linearity in the model. It allows the network to learn and represent complex patterns in the data.

There are several types of activation functions, some of the major ones are

**11.Relu:**

It stands for Rectified linear unit. It is the most widely used activation function. It is used in the hidden layers.

Equation: A(x) = max(0,x)

Value : 0 to inf

**12.Sigmoid:**

Usually used in the output layer of the binary classification. Where the result is either one or zero.

Equation: A = 1/(1+e^-x)

Value: 0 to 1

**13.14Tanh:**

It stands for Tangent Hyperbolic function. It is a mathematically shifted version of the Sigmoid function. And performs better in most of the cases.

Equation : tanh(x) = 2/(1+e-2x)-x

Value: -1 to +1

**14.Softmax:**

Softmax is also a sigmoid function but it is widely used in multi class classification. It converts a vector of raw scores into probabilities, which sum to one, allowing the network to make probabilistic predictions for each class.

**15.Forward Propagation:**

It is the process to train neural networks by moving inputs through the layers of a neural network to generate the output. The aim is to transform the input data into meaningful classification or predictions. It involves matrix multiplication, addition, and applying0 activation functions at each layer.

**16. Backward Propagation:**

This process is used to train neural networks by minimizing the error between the predicted output and actual output. It involves calculating the gradient of the loss function with respect to each weight in the network, using the chain rule of calculus.

**17. Loss Function:**

It is a mathematical function used to quantify the difference between the predicted output of a model and the actual output. The purpose of loss function is to provide a measure of how well a model is performing.

**18. Cost Function:**

It is a measure of how well the model is performing over the entire training dataset. It is usually the sum of the loss function for all the training examples.

**19. Gradient Descent:**

It is an optimization algorithm used to minimize the cost function in the models by iteratively adjusting the model parameters. The aim is to find the parameter values that result in the lowest possible cost.

**20. Learning Rate:**

The learning rate is a hyperparameter in optimization algorithms. It controls the size of the steps the algorithm takes towards minimizing the cost function.

**21. Batch size:**

It is a hyperparameter that defines the number of training examples used in one iteration of updating the model's parameters. The choice of batch size can significantly impact the training process and performance.

**22. Epoch:**

It refers to one complete pass through the entire training dataset. During an epoch, the learning algorithm processes each training example once, updating the model parameters (weights and biases) based on the computed gradients.

**23. Overfitting:**

When a model trains exceptionally well on the training dataset but poorly on test or validation data, overfitting occurs. This indicates that the model has become too complex and tailored to the training data rather than generalizing well to new data.

**24. Underfitting:**

Underfitting occurs when the model is too simple to capture the underlying patterns in the training data. The model performs poorly on both the training data and unseen test or validation data.

**25. Training set:**

It is a set of dataset which is used for training the model. It contains the majority of the data samples and their corresponding labels

**26. Validation set:**

A validation set is used to evaluate the model’s performance during the process, allowing for hyperparameter tuning, model selection, and performance assessment.

**27. Test set:**

A test set is used to evaluate the performance of a trained machine learning model. It is distinct from both the training and validation set.\

**28. Cross validation:**

It is used to provide a more reliable and less biased estimate of a model's performance by using multiple subsets of the data for training and validation.

**29. Hyperparameters:**

They are configuration settings used to tune models, set before the learning process begins. Unlike model parameters, which are learned from the training data, hyperparameters are set manually and can significantly impact the performance and behavior of a mode

**30. Model Parameter:**

They are the internal variables of a model that are learned from the training data during the training process. These parameters define the model's function and determine how it transforms input data into the desired output

**31. Regularization:**

Used to prevent overfitting by adding a penalty to the loss function. This penalty discourages the model from fitting the noise in the training data and helps it generalize better to unseen data.

**32. Dropout:**

Dropout is a regularization technique used primarily in training neural networks to prevent overfitting. It works by randomly "dropping out" (setting to zero) a subset of neurons during the training process, which forces the network to learn more robust features that are not reliant on any specific neurons.

**33. Weight Initialization:**

It sets the initial values of the weights before training begins. Proper weight initialization can help ensure faster convergence, avoid vanishing/exploding gradient problems, and lead to better overall model performance.

**34. Normalization:**

Normalization is a preprocessing technique used to adjust the scale of input data to ensure that all features contribute equally to the model training process. This is important because features with different scales can disproportionately influence the model, potentially leading to poorer performance or longer training times.

**35. Standardization**:

Standardization is a preprocessing technique used to center and scale feature values so that they have a mean of zero and a standard deviation of one.This process is often essential when working with machine learning algorithms that are sensitive to the scale of the data, such as gradient descent-based algorithms or models that use distance metrics