**RNA Sequence:**  It is a technique that examines the quantity and sequences of RNA in a sample. (using next-generation sequencing)

**RNA:** It is a nucleic acid present in all living cells that has structural similarities to DNA  
  
**Working of CNN:   
  
Normalization:** is means transforming the data to appear on the same scale across all the records

*(data – np.min(data)) / (np.max(data) – np.min(data))*  
 **NumPy:** It is a library for Python, it adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to   
 operate on these arrays.

**Heat Map:** Itis a data visualization technique that shows magnitude of a phenomenon as color in two dimensions. The variation in color may be by hue or intensity.  **Matplotlib:** Itis a plotting library for Python and its numerical mathematics extension NumPy **imshow() function:** used to display an image in a window  
 **Convolutional Layer:** It is the main building block of a CNN. It contains a set of filters/kernels, parameters of which are to be learned throughout the training. **Kernal Size (3\*3):** It “slides” over the 2D input data, performing an elementwise multiplication. As a result, it will be summing up the results into a single output pixel. The kernel will perform the same operation for every location it slides over, transforming a 2D matrix of features into a different 2D matrix of features. **Pooling Layer:** It is used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn, and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. **Batch Normalization Layer:** It is a process to make neural networks faster and more stable through adding extra layers in a deep neural network. The new layer performs the standardizing and normalizing operations on the input of a layer coming from a previous layer.

**Need for Activation Function:** We need the activation function to introduce nonlinear real-world properties to artificial neural networks. Basically, in a simple neural network, x is defined as inputs, w weights, and we pass f (x) that is the value passed to the output of the network. This will then be the final output or the input of another layer.

**Activation Function:** Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

**ReLu (**Rectified Linear Unit)**:** The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

**Softmax:**performs well when used as a classifier. The most important difference is that it is preferred in the output layer of deep learning models. It is used in multiple classification logistic regression model. I[t allows determining the probability that the input belongs to a particular class by producing values in the range 0-1.](https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/) So, it performs a probabilistic interpretation.

**Dropout:** is a [regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics)) technique for reducing [overfitting](https://en.wikipedia.org/wiki/Overfitting) in [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) by preventing complex co-adaptations on [training data](https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets). It randomly drops units (both hidden and visible) during the training process of a neural network.

**Transcriptome** is the full range of messenger RNA, or mRNA, molecules expressed by an organism. The term "transcriptome" can also be used to describe the array of mRNA transcripts produced in a particular cell or tissue type.

[2] unsupervised learning algorithms include **K-Means Clustering, Principal Component Analysis and Hierarchical Clustering**

**Feature Selection:** It is used to reduce the number of input variables.

**RFE:** It removes the weakest features until the specified number of features is reached.

The scikit-learn library provides an implementation of RFE for machine learning.

**Normalization:** is means transforming the data to appear on the same scale across all the records

*(data – np.min(data)) / (np.max(data) – np.min(data))*

**Convolutional Layer:** It is the main building block of a CNN. It contains a set of filters/kernels, parameters of which are to be learned throughout the training. **Kernal Size (3\*3):** It “slides” over the 2D input data, performing an elementwise multiplication. As a result, it will be summing up the results into a single output pixel. The kernel will perform the same operation for every location it slides over, transforming a 2D matrix of features into a different 2D matrix of features. **Pooling Layer:** It is used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn, and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. **Batch Normalization Layer:** It is a process to make neural networks faster and more stable through adding extra layers in a deep neural network. The new layer performs the standardizing and normalizing operations on the input of a layer coming from a previous layer.

**Need for Activation Function:** We need the activation function to introduce nonlinear real-world properties to artificial neural networks. Basically, in a simple neural network, x is defined as inputs, w weights, and we pass f (x) that is the value passed to the output of the network. This will then be the final output or the input of another layer.

**Activation Function:** Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

**ReLu (**Rectified Linear Unit)**:** The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

**Softmax:**performs well when used as a classifier. The most important difference is that it is preferred in the output layer of deep learning models. It is used in multiple classification logistic regression model. I[t allows determining the probability that the input belongs to a particular class by producing values in the range 0-1.](https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/) So, it performs a probabilistic interpretation.

**Dropout:** is a [regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics)) technique for reducing [overfitting](https://en.wikipedia.org/wiki/Overfitting) in [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) by preventing complex co-adaptations on [training data](https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets). It randomly drops units (both hidden and visible) during the training process of a neural network.

**Epochs:** One **Epoch** is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE.

**Batch:** It is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.

**Accuracy:** It is calculated as the ratio between the number of correct predictions to the total number of predictions.

**Cohen Kappa:**

It a statistical measure that is used to check if two raters rating the same quantity are reliable and agree with each other.

**(**it tells us how much better our classifier is performing over the performance of a classifier that simply guesses at random according to the frequency of each class.)

**F1-Score:** It combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

(Harmonic Mean: is calculated by dividing the number of observations by the reciprocal of each number in the series)

P**recision:** The precision measures the model trustiness in classifying positive samples, it takes into account how both the positive and negative samples were classified

**Recall:** measures how many positive samples were correctly classified by the model, recall is dependent only on the positive samples