BASIC TERMS:

- **TRADING:** Trade is a basic economic concept involving the buying and selling of goods and services, with compensation paid by a buyer to a seller, or the exchange of goods or services between parties. Trade can take place within an economy between producers and consumers.

- **STOCK:** Stocks are securities that represent an ownership share in a company. For companies, issuing stock is a way to raise money to grow and invest in their business. For investors, stocks are a way to grow their money and outpace inflation over time.

- **STOCK EXCHANGE:** The stock exchange is a virtual market where buyers and sellers trade in existing securities. It is a market hosted by an institute or any such government body where shares, stocks, debentures, bonds, futures, options, etc are traded. A stock exchange is a meeting place for buyers and sellers. Examples of stock exchanges are New York Stock Exchange (NYSE), NASDAQ, Tokyo Stock Exchange (JPX), Bombay Stock Exchange (BSE), National Stock Exchange (NSE).

- **BROKER:** A stockbroker is a financial professional who executes orders in the market on behalf of clients. A stockbroker may also be known as a registered representative (RR) or an investment advisor. Examples: Zerodha, UpStox etc.

- **FACTORS THAT CAN AFFECT STOCK:** https://www.5paisa.com/blog/9-factors-that-affects-the-indian-stock-market.

DIFFERENT TYPES OF ALGORITHM:

- **MATHEMATICAL METHODS:**
  - Random Walk (RW)
  - Auto Regressive Integrated Moving Average (ARIMA)
  - Auto Regressive Conditional Heteroskedacity (ARCH)
  - Generalized Auto Regressive Conditional Heteroskedacity (GARCH)
- **MACHINE LEARNING MODELS:**
  - Logistic Regression (LR)
  - Decision Tree (DT)
  - Support Vector Machine (SVM)
  - Random Forest (RF)
  - Artificial Nueral Network (ANN)

- **DEEP LEARNING MODELS:**
  - Long Short-Term Memory (LSTM)
  - Recurrent Nueral Networks (RNN)
  - Gated Recurrent Unit (GRU)

MACHINE LEARNING: Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

DEEP LEARNING: It is just a type of Machine Learning, inspired by the structure of a human brain. Deep learning algorithms attempt to draw similar conclusions as humans would by continually analyzing data with a given logical structure. To achieve this, deep learning uses a multi-layered structure of algorithms called neural networks.

MATHEMATICAL METHODS:

- **Random Walk:** It basically states that returns on a stock tomorrow can be calculated using the return today plus an error term. An error term is the deviation of reality from your model that cannot be calculated by your model. If it was possible to be calculated, it should have been integrated into your model.

  Equation: $R(t+1) = R(t) + e$.

- **ARIMA:** This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:
  - **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
  - **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
  - **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

  Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

  The parameters of the ARIMA model are defined as follows:

  - **p**: The number of lag observations included in the model, also called the lag order.
  - **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
  - **q**: The size of the moving average window, also called the order of moving average.

  A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model.

- **ARCH:** An ARCH (autoregressive conditionally heteroscedastic) model is a model for the variance of a time series. ARCH models are used to describe a changing, possibly volatile variance. Although an ARCH model could possibly be used to describe a gradually increasing variance over time, most often it is used in situations in which there may be short periods of increased variation. (Gradually increasing variance connected to a gradually increasing mean level might be better handled by transforming the variable.)
    - An ARCH(m) process is one for which the variance at time *t* is conditional on observations at the previous *m* times, and the relationship is

$$Var(y_t \vee y_{t-1}, \ldots, y_{t-m}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \ldots + \alpha_m y_{t-m}^2.$$

- **GARCH:** A GARCH (generalized autoregressive conditionally heteroscedastic) model uses values of the past squared observations and past variances to model the variance at time . As an example, a GARCH(1,1) is

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

MACHINE LEARNING MODELS:

- **Logistic Regression:** The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits. Sigmoid function:

$$1/(1+e^{-x})$$

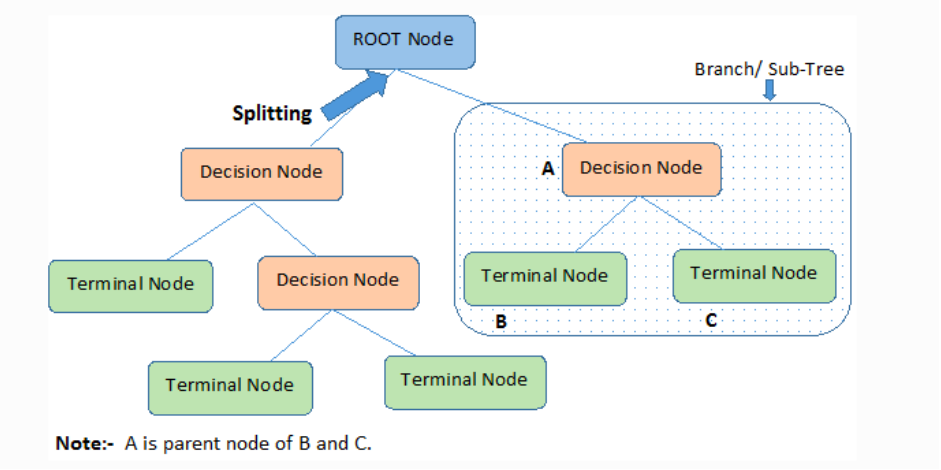
  Logistic regression uses an equation as the representation, very much like linear regression. Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value. Below is an example logistic regression equation:

$$y = e^{(b0+b1*x)}/(1+e^{(b0+b1*x)})$$

- **Decision Tree:** Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

  The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

  In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

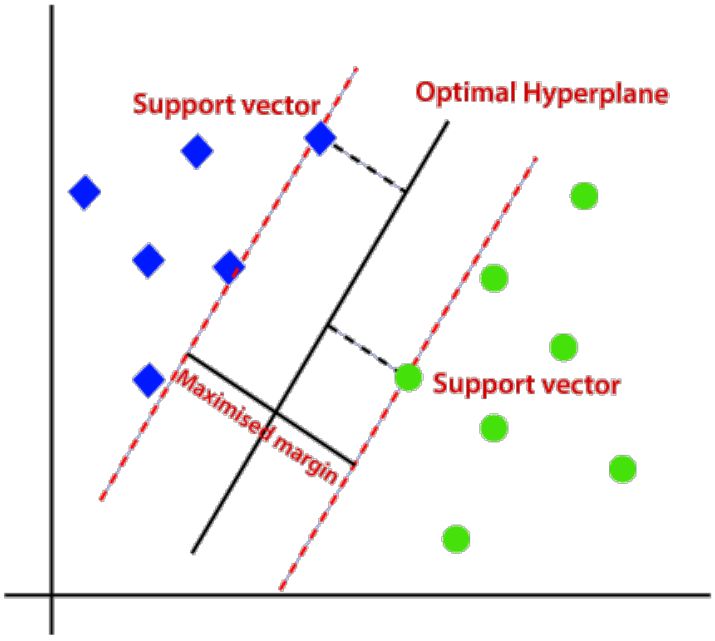


Source: https://medium.datadriveninvestor.com/decision-tree-algorithm-with-hands-on-example-e6c2afb40d38

- **Support Vector Machine:** SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

  At first approximation what SVMs do is to find a separating line(or hyperplane) between data of two classes. SVM is an algorithm that takes the data as an input and outputs a line that separates those classes if possible.

  According to the SVM algorithm we find the points closest to the line from both the classes.These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane.



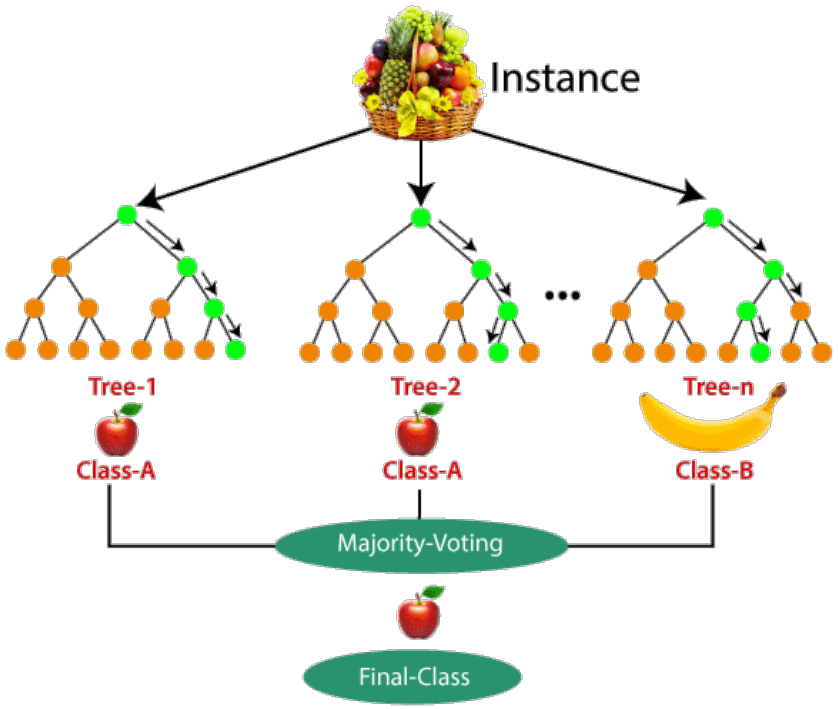
Source: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

- **Random Forest:** A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

  A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

  The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction.

  Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.



Source: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

- **Artificial Nueral Network:** Artificial Neural Networks are a special type of machine learning algorithms that are modeled after the human brain. That is, just like how the neurons in our nervous system are able to learn from the past data, similarly, the ANN is able to learn from the data and provide responses in the form of predictions or classifications.

  ANNs are nonlinear statistical models which display a complex relationship between the inputs and outputs to discover a new pattern. A variety of tasks such as image recognition, speech recognition, machine translation as well as medical diagnosis makes use of these artificial neural networks.

An important advantage of ANN is the fact that it learns from the example data sets. Most commonly usage of ANN is that of a random function approximation. With these types of tools, one can have a cost-effective method of arriving at the solutions that define the distribution. ANN is also capable of taking sample data rather than the entire dataset to provide the output result. With ANNs, one can enhance existing data analysis techniques owing to their advanced predictive capabilities.

ANNs consists of three layers:

Input Layers

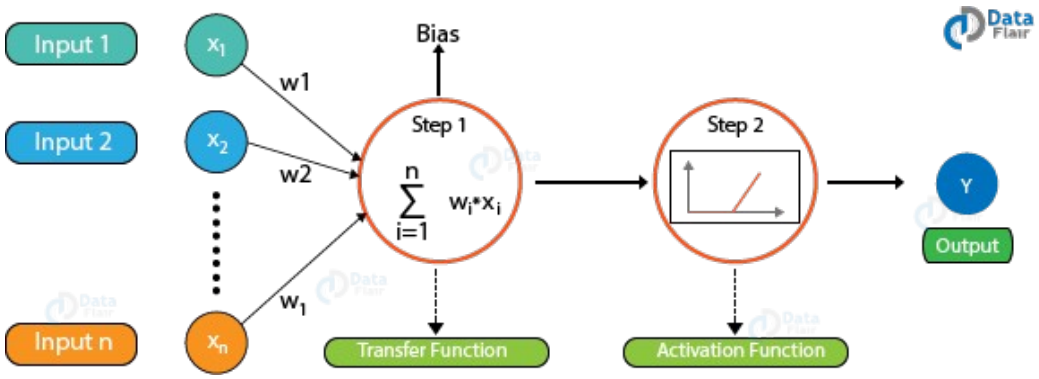
The *input layer* is the first layer of an ANN that receives the input information in the form of various texts, numbers, audio files, image pixels, etc.

Hidden Layers

In the middle of the ANN model are the *hidden layers*. There can be a single hidden layer, as in the case of a perceptron or multiple hidden layers. These hidden layers perform various types of mathematical computation on the input data and recognize patterns.

Output Layer

In the *output layer*, we obtain the result that we obtain through rigorous computations performed by the middle layer.

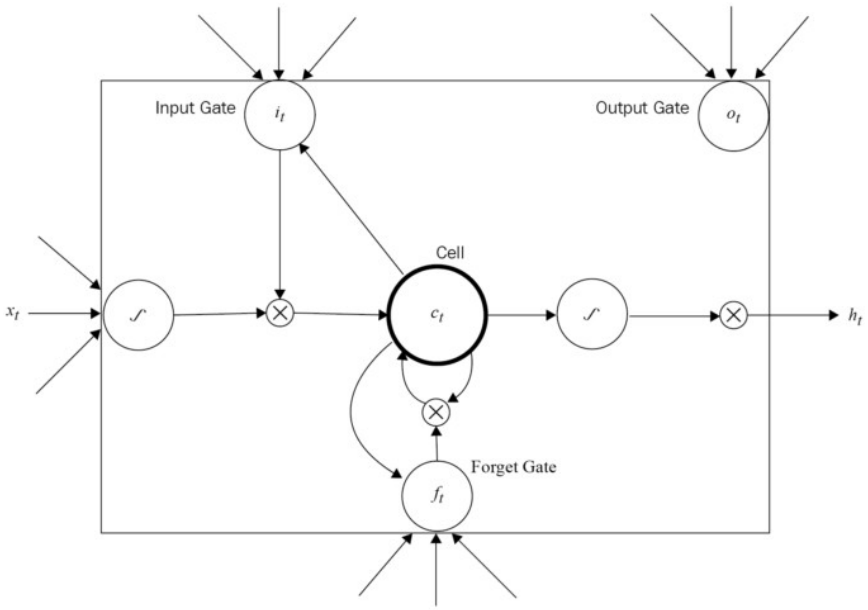


Source: https://data-flair.training/blogs/artificial-neural-networks-for-machine-learning/

DEEP LEARNING MODELS:

- **LSTM:** LSTM stands for **long short term memory**. It is a model or architecture that extends the memory of recurrent neural networks. Typically, recurrent neural networks have 'short term memory' in that they use persistent previous information to be used in the current neural network. Essentially, the *previous information is used in the present task*. That means we do not have a list of all of the previous information available for the neural node.

  LSTM introduces long-term memory into recurrent neural networks. It mitigates the vanishing gradient problem, which is where the neural network stops learning because the updates to the various weights within a given neural network become smaller and smaller. It does this by using a series of 'gates'. These are contained in memory blocks which are connected through layers, like this:



Source: https://hub.packtpub.com/what-is-lstm/

There are three types of gates within a unit:

- Input Gate: Scales input to cell (write)
- Output Gate: Scales output to cell (read)
- Forget Gate: Scales old cell value (reset)

Each gate is like a switch that controls the read/write, thus incorporating the long-term memory function into the model.

- **Recurrent Nueral Networks:** Recurrent neural networks (RNN) are a class of neural networks that are helpful in modeling sequence data. Derived from feedforward networks, RNNs exhibit similar behavior to how human brains function. Simply put: recurrent neural networks produce predictive results in sequential data that other algorithms can’t. RNNs are a powerful and robust type of neural network, and belong to the most promising algorithms in use because it is the only one with an internal memory.

  Like many other deep learning algorithms, recurrent neural networks are relatively old. They were initially created in the 1980’s, but only in recent years have we seen their true potential. An increase in computational power along with the the massive amounts of data that we now have to work with, and the invention of long short-term memory (LSTM) in the 1990s, has really brought RNNs to the foreground.

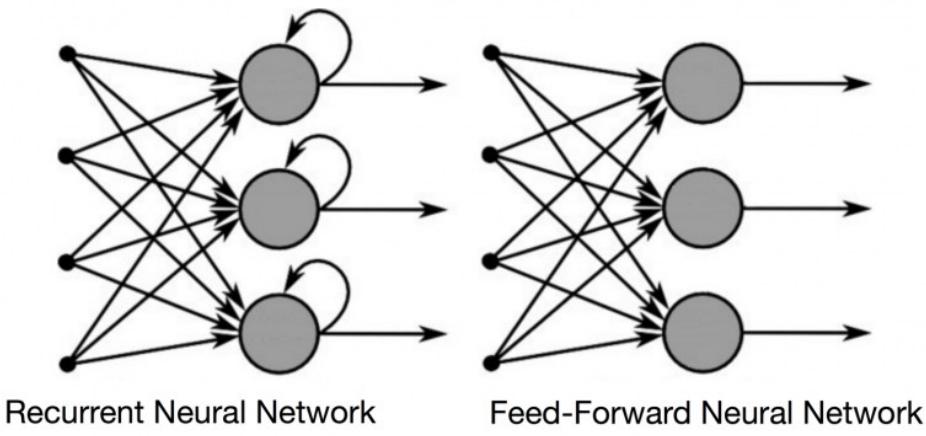
  Because of their internal memory, RNN’s can remember important things about the input they received, which allows them to be very precise in predicting what’s coming next. This is why they're the preferred algorithm for sequential data like time series, speech, text, financial data, audio, video, weather and much more. Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms.

  To understand RNNs properly, you'll need a working knowledge of "normal“ feed-forward neural networks and sequential data.

In a feed-forward neural network, the information only moves in one direction — from the input layer, through the hidden layers, to the output layer. The information moves straight through the network and never touches a node twice.

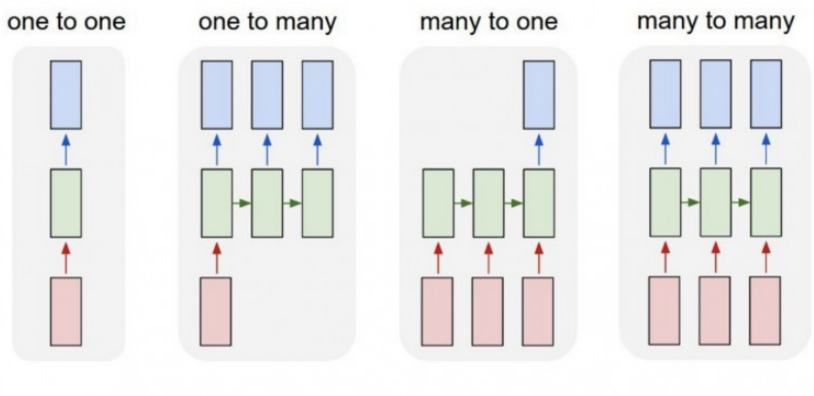
Feed-forward neural networks have no memory of the input they receive and are bad at predicting what's coming next. Because a feed-forward network only considers the current input, it has no notion of order in time. It simply can't remember anything about what happened in the past except its training.

In a RNN the information cycles through a loop. When it makes a decision, it considers the current input and also what it has learned from the inputs it received previously.



Source: https://builtin.com/data-science/recurrent-neural-networks-and-lstm

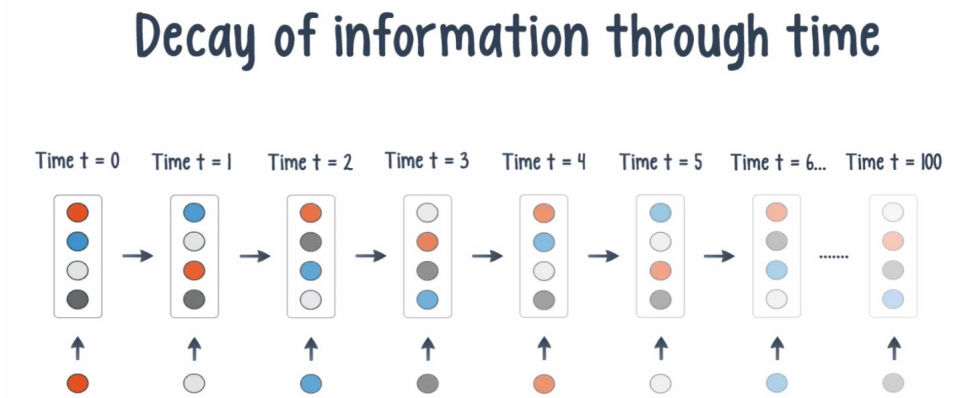
Also note that while feed-forward neural networks map one input to one output, RNNs can map one to many, many to many (translation) and many to one (classifying a voice).



Source: https://builtin.com/data-science/recurrent-neural-networks-and-lstm

- **Gated Recurrent Unit:** GRU (Gated Recurrent Unit) aims to solve the **vanishing gradient problem** which comes with a standard recurrent neural network.

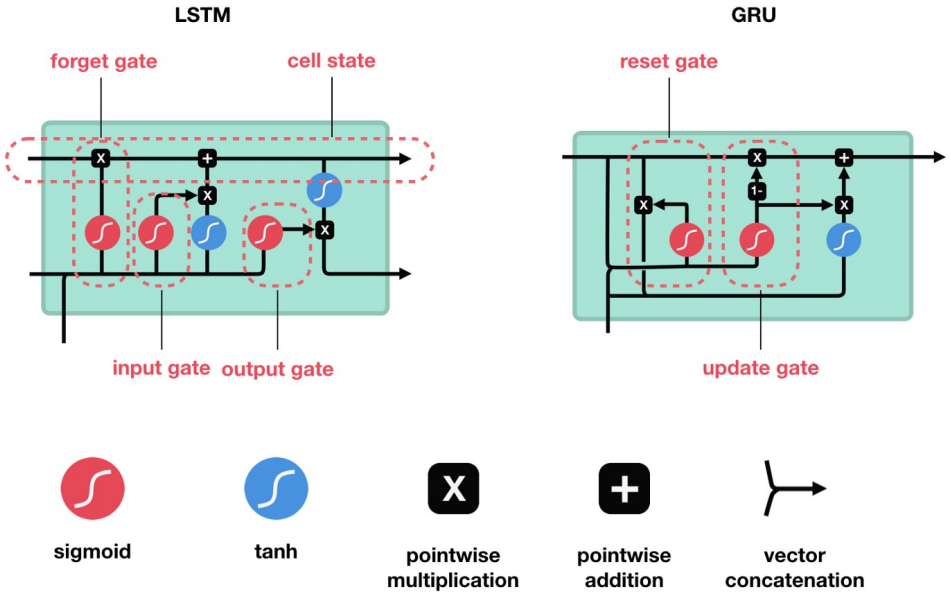
Vanishing gradient problem is a difficulty found in training artificial neural networks with gradient-based learning methods and backpropagation. In such methods, each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. As one example of the problem cause, traditional activation functions such as the hyperbolic tangent function have gradients in the range (0, 1), and backpropagation computes gradients by the chain rule. This has the effect of multiplying n of these small numbers to compute gradients of the "front" layers in an n-layer network, meaning that the gradient (error signal) decreases exponentially with n while the front layers train very slowly.



Source: https://www.kaggle.com/thebrownviking20/intro-to-recurrent-neural-networks-lstm-gru

In simple words, the GRU unit does not have to use a memory unit to control the flow of information like the LSTM unit. It can directly makes use of the all hidden states without any control. GRUs have fewer parameters and thus may train a bit faster or need less data to generalize. But, with large data, the LSTMs with higher expressiveness may lead to better results.

They are almost similar to LSTMs except that they have two gates: reset gate and update gate. Reset gate determines how to combine new input to previous memory and update gate determines how much of the previous state to keep. Update gate in GRU is what input gate and forget gate were in LSTM. We don't have the second non linearity in GRU before calculating the outpu, .neither they have the output gate.



Source: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21