There are 2 major types of neural networks: biological and artificial. Both types of neural networks form a computing architecture that is very different from that of the conventional digital computer: They are massively parallel, use very simple processing units, and have a vast array of interconnections among these units.

Artificial neural networks represent an attempt to mimic the neural networks that exist in biological brains. The attempt takes a double approach: It is an effort aimed at a better understanding of the functioning of biological brains, and it is also an effort aimed at building machines that can duplicate many of the amazing abilities of biological brains.

Biological brain is able to break down many tasks into a number of small steps that are then processed simultaneously, by a large number of identical processing units distributed throughout the brain. This approach to problem solving is called "parallel distributed processing." It can result in very dramatic increases in problem solving speed compared to the speed obtained using traditional computer algorithms in which one processing unit goes through a long series of problem-solving steps in a sequential manner.

Artificial neural networks can modify their behaviour as a result of inputs. This learning capacity is one of their most interesting capabilities. They accomplish this by adjusting themselves until their responses are consistent.

ANN can have millions of neurons connected into one system, which makes it extremely successful at analysing and even memorizing various information.

Many training procedures exist, and the selection of any one of them is often determined by the tasks envisioned for the network. The ability of artificial neural networks to recognize patterns in the presence of distortions or noise is similar to the capabilities of biological networks and represents a major improvement over the conventional computer. The structure of the network, rather than a computer program that has been produced by a human, is responsible for this recognition ability.

A neural network, is composed of a complex interconnected web of individual, biological "neurons" .Each neuron, although separated from the others, may form extremely close associations with other neurons for the purpose of signal transmission from one neuron to another. These communicating structures are called "synapses." It is neither the very simple processing units (neurons) nor their large numbers that allow neural networks to attain their performance. Instead, it is the tremendous number of synapses (interconnections) that give them their computing capabilities.

Artificial neural networks are naturally less complex in structure than their biological counterparts. Nevertheless, the artificial networks resemble the brain closely enough to allow their use as practical models in studies of information processing by the brain.

The artificial neuron receives a set of inputs, each of which represents the synaptic input either from another artificial neuron in the network or from an input device of some type. Each signal that arrives from an input line is multiplied by an assigned synaptic weight (representing the strength of that particular synaptic connection). The resulting product (arriving signal times synaptic weight) is called the input signal. All the input signals are added, which produces a net change in the activation level of the receiving neuron. If this activation level exceeds the neuron's assigned threshold level, the neuron generates an output signal of its own and sends it to all neurons to which its axon is connected. The form of the output signal from an artificial neuron depends on the specific neural models that are used. Many models generate an output signal that takes as a value either 1 or 0, while in other models the value can be either +1 or -1.

Applications

Neural networks can be used in a wide variety of applications. They are already being used in many commercial applications, and new uses are being researched in laboratories throughout the world. They are most appropriately applied to problems demanding pattern recognition, pattern completion, or pattern mapping. Other areas that are appropriate for the use of neural networks include applications that must deal with noisy data: Traditional computers are very sensitive to noisy data, while neural networks are, by comparison, quite insensitive. Such situations are common in speech recognition systems in which background noise often cannot be eliminated, in image processing and analysis, and in sonar signal classification.

One of the first commercial applications of artificial neural networks was in the identification of handwritten numerals and letters, such as those entered on bills to indicate the amount being paid or printed on envelopes to indicate the address. The challenge is to be able to identify correctly the characters in different people's handwriting. The variations in handwritten characters among writers are numerous enough to present great difficulties to traditional computer program developers. Neural networks, however, are first trained on a sample set of characters for which they are given the correct responses. After the training period has been completed, the ability of neural networks to generalize and to categorize enables the correct identification of handwritten characters from any writer.

In the biochemical field, pattern classification can become a monumentally complex task. There are thousands of organic compounds, each of which has a set of biochemical characteristics (the pattern), some of which are unique to each substance but are shared with other substances. Neural networks are used to automate the pattern classification process, thereby eliminating the human drudgery associated with this tedious procedure.

Artificial neural networks are used to obtain complicated financial analyses. Area of financial forecasting requires the analysis of numerous parameters and the recognition of particular patterns of these parameters. Capacity to form associative memory systems with artificial neural networks has been of great value in this application. Artificial neural networks are able to use their learning and pattern recognition skills to achieve accurate & useful results.

Speech recognition : All major commercial speech recognition systems (like Microsoft Cortana, Alexa, Google Assistant, Apple Siri) are based on deep learning

Pattern recognition : Pattern recognition systems are already able to give more accurate results than the human eye in medical diagnosis.

NLP: Neural networks have been used to implement language models since 2000s. Invention of LSTM helped improve machine translation & language modelling.

Discovery of new drugs : For EG - AtomNet neural network has been used to predict new biomolecules that can potentially cure diseases such as Ebola and multiple sclerosis.

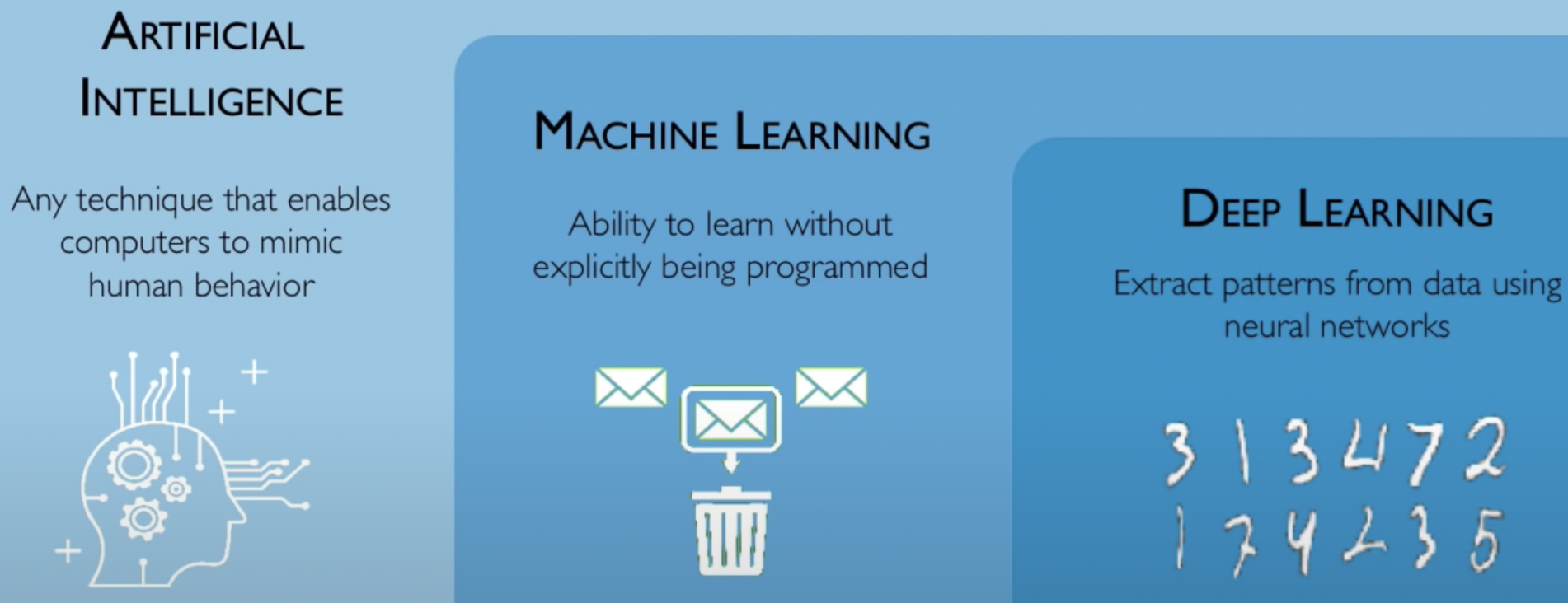
Recommender systems : Deep learning is being used to study user preferences across many domains. EG - Netflix

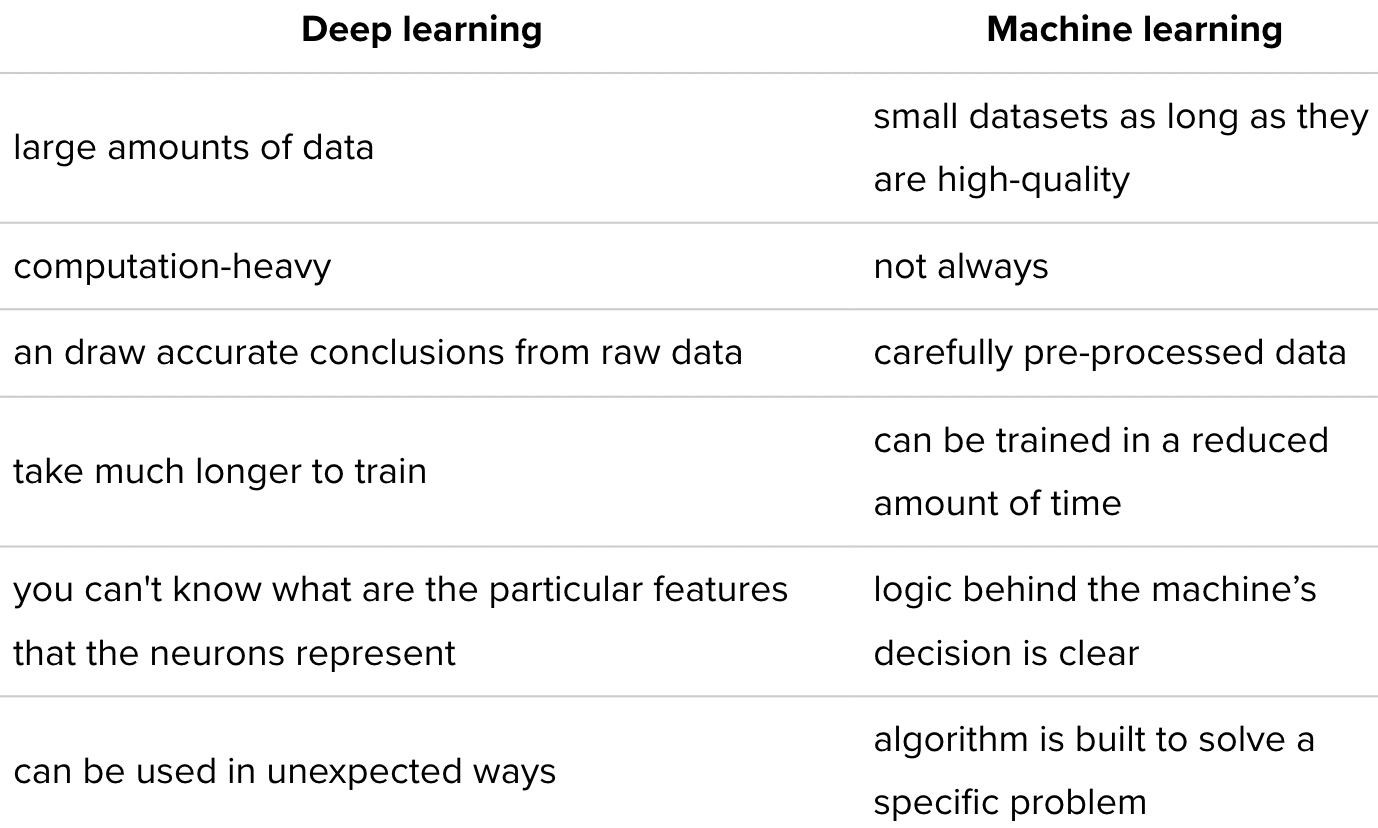
Heart of the network is its architecture which is at the designer's discretion. The design choices include factors as the number of layers of neurons in the network, the number of neurons in each layer, the patterns of interconnections between layers, and the types of learning and recall rules to be used during the training and use of the network.

Once network has been constructed, training period begins. During training it is determined whether the performance is sufficient, whether additional training is required, or whether design changes may be necessary. If performance is adequate, it is switched to recall mode.

Artificial neural networks are the first machines that learn in much the same way as human brains, without following a program of instructions, as do conventional digital computers. Neural networks require neither programming nor the use of complex cognitive rules, as does the conventional "expert system" branch of artificial intelligence; therefore, neural networks do not need to be reprogrammed when they are used in new applications.

Deep learning is one of the subsets of machine learning that uses deep learning algorithms to implicitly come up with important conclusions based on input data. Usually, deep learning is unsupervised or semi-supervised.





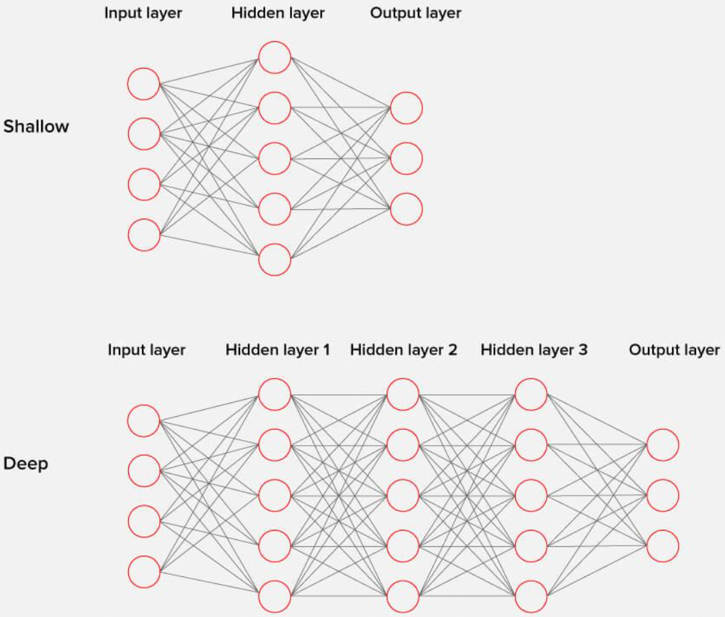
Advantages of deep learning

* The ability to identify patterns and anomalies in large volumes of raw data enables deep learning to efficiently deliver accurate and reliable analysis results to professionals.
* Deep learning doesn’t rely on human expertise as much as traditional machine learning. DL allows us to make discoveries in data even when the developers are not sure what they are trying to find.

Problems of deep learning

* Large amounts of quality data are resource-consuming to collect.
* Another difficulty with deep learning technology is that it cannot provide reasons for its conclusions. Therefore, it is difficult to assess the performance of the model if you are not aware of what the output is supposed to be.
* It is very costly to build deep learning algorithms. It is impossible without qualified staff who are trained to work with sophisticated maths. Moreover, deep learning is a resource-intensive technology. It requires powerful GPUs and a lot of memory to train the models. A lot of memory is needed to store input data, weight parameters, and activation functions as an input propagates through the network. Sometimes deep learning algorithms become so power-hungry that researchers prefer to use other algorithms, even sacrificing the accuracy of predictions.

“Artificial neural networks” and “deep learning” are often used interchangeably, which isn’t really correct. Not all neural networks are “deep”, meaning “with many hidden layers”, and not all deep learning architectures are neural networks.



Components of Neural Networks

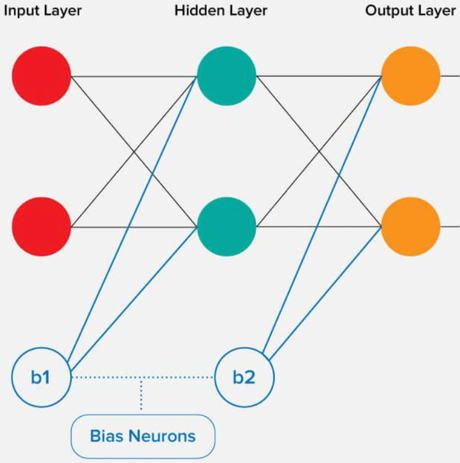
Neuron : Neuron/node is a basic unit of neural network that receives information, performs simple calculations, & passes it further. All neurons in a net are divided into 3 groups:

* Input neurons that receive information from the outside world
* Hidden neurons that process that information
* Output neurons that produce a conclusion

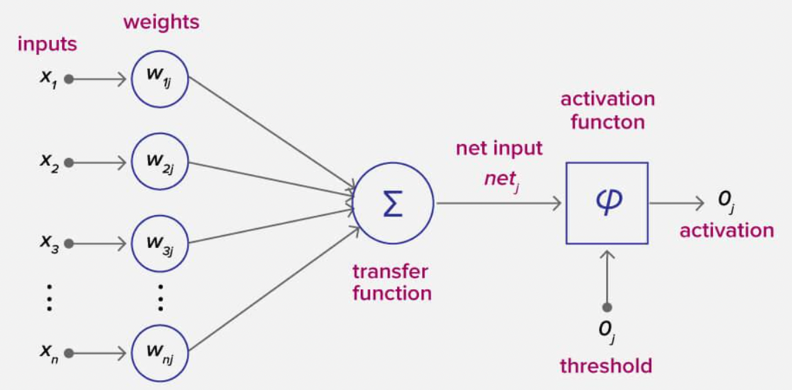
In a large neural network with many neurons and connections between them, neurons are organized in layers. Input layer receives information, a number of hidden layers, and output layer that provides valuable results. Every neuron performs transformation on the input information. Neurons only operate numbers in the range [0,1] or [-1,1].

Synapses and weights : A synapse connects neurons. Every synapse has a weight. The weights also add to the changes in the input information. The results of the neuron with the greater weight will be dominant in the next neuron, while information from less ‘weighty’ neurons will not be passed over. Matrix of weights governs the whole neural system.

Bias : A bias neuron allows for more variations of weights to be stored. Biases add richer representation of the input space to the model’s weights. In the case of neural networks, a bias neuron is added to every layer. It plays a vital role by making it possible to move the activation function to the left or right on the graph. ANNs can work without bias neurons. However, they are always added and counted as an indispensable part of the overall model.



How ANNs work : Every neuron processes input data to extract a feature. Let’s imagine we have 3 features and 3 neurons, each of which is connected with all these features. Each neurons has its own weights that are used to weight the features. During the training you need to select such weights for each of the neurons that the output provided by the whole network would be true-to-life. To perform transformations and get an output, every neuron has an activation function. This combination of functions performs a transformation that is described by a common function F — this describes the formula behind the NN’s magic. There are a lot of activation functions. The most common ones are linear, sigmoid, and hyperbolic tangent. Their main difference is the range of values they work with.



How do you train an algorithm?

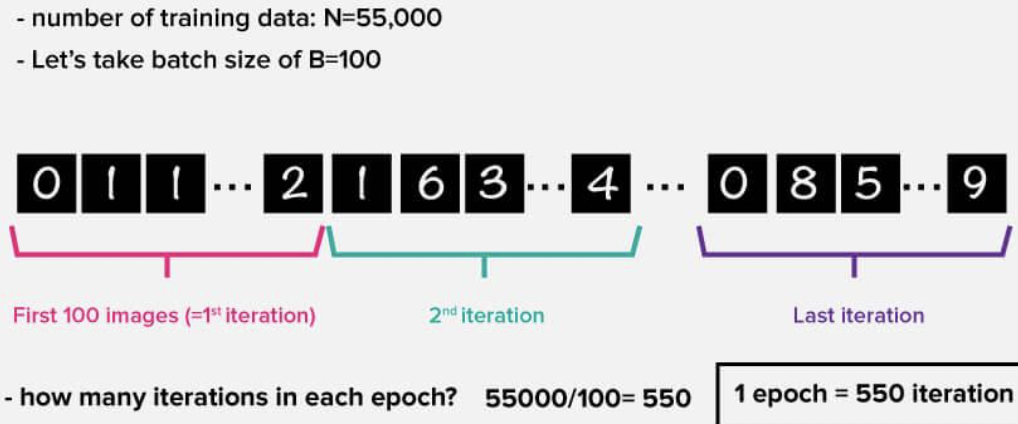
Neural networks are trained like any other algorithm. You want to get some results and provide information to the network to learn from.

Delta is the difference between the data and the output of the neural network. We use calculus magic and repeatedly optimize the weights of the network until the delta is zero. Once the delta is zero or close to it, our model is correctly able to predict our example data.

Iteration : A kind of counter that increases every time neural network goes through 1 training set. It is the total number of training sets completed by the neural network. Number of iterations is a number of passes

Epoch : The epoch increases each time we go through the entire set of training sets. The more epochs there are, the better is the training of the model. 1 epoch is 1 forward pass and 1 backward pass of all the training examples

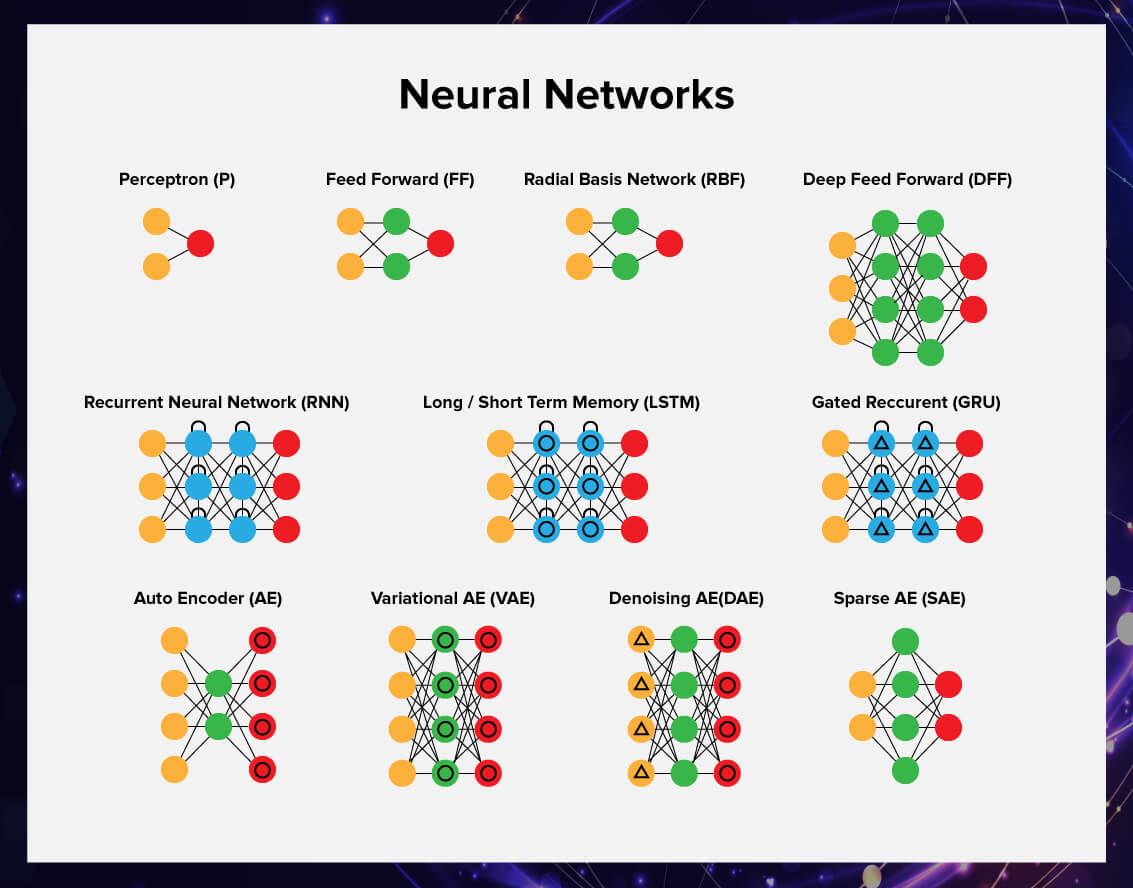
Batch : Batch size is equal to the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you’ll need.



Error is a deviation that reflects the discrepancy between expected and received output. The error should become smaller after every epoch. If this does not happen, then you are doing something wrong. The error can be calculated in different ways. Some EG :

With Arctan, the error will almost always be larger

MSE is more balanced and is used more often



<https://www.asimovinstitute.org/neural-network-zoo/>

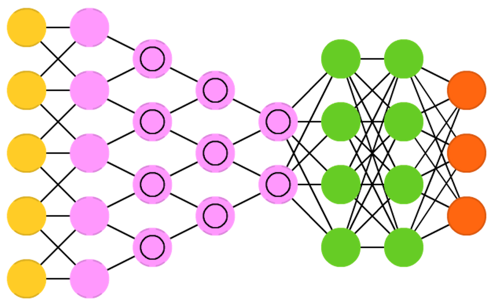
Feed-forward neural networks : Simplest neural network algorithm. A feed-forward network doesn’t have any memory. That is, there is no going back in a feed-forward network. In many tasks, this approach is not very applicable. For example, when we work with text, the words form a certain sequence, and we want the machine to understand it. Feedforward neural networks can be applied in supervised learning when the data that you work with is not sequential or time-dependent.

Radial basis function (RBF) networks are FFNNs with radial basis functions as activation functions.

Recurrent neural networks : A recurrent neural network can process texts, videos, or sets of images and become more precise every time because it remembers the results of the previous iteration and can use that information to make better decisions. Neurons are fed information not just from the previous layer but also from themselves from the previous pass. One big problem with RNNs is the vanishing (or exploding) gradient problem where, depending on the activation functions used, information rapidly gets lost over time, just like very deep FFNNs lose information in depth. Recurrent neural networks are widely used in natural language processing and speech recognition.

Long / short term memory : LSTM networks try to combat the vanishing /exploding gradient problem by introducing gates and an explicitly defined memory cell. Each neuron has a memory cell and 3 gates: input, output and forget. The function of these gates is to safeguard the information by stopping or allowing the flow of it. The input gate determines how much of the information from the previous layer gets stored in the cell. The output gate takes the job on the other end and determines how much of the next layer gets to know about the state of this cell. The forget gate seems like an odd inclusion at first but sometimes it’s good to forget: if it’s learning a book and a new chapter begins, it may be necessary for the network to forget some characters from the previous chapter. LSTMs have been shown to be able to learn complex sequences, such as writing like Shakespeare or composing primitive music. Note that each of these gates has a weight to a cell in the previous neuron, so they typically require more resources to run.

Convolutional neural networks (CNN or deep convolutional neural networks, DCNN) are quite different from most other networks. Convolutional neural networks are the standard of today’s deep machine learning and are used to solve the majority of problems. They can be either feed-forward or recurrent. They are primarily used for image processing but can also be used for other types of input such as audio. A typical use case for CNNs is where you feed the network images and the network classifies the data. CNNs tend to start with an input “scanner” which is not intended to parse all the training data at once. This input data is then fed through convolutional layers instead of normal layers, where not all nodes are connected to all nodes. Each node only concerns itself with close neighbouring cells. These convolutional layers also tend to shrink as they become deeper, mostly by easily divisible factors of the input. Powers of two are very commonly used here, as they can be divided cleanly and completely by definition: 32, 16, 8, 4, 2, 1. Besides these convolutional layers, they also often feature pooling layers. Pooling is a way to filter out details: a commonly found pooling technique is max pooling. To apply CNNs for audio, we basically feed the input audio waves and inch over the length of the clip, segment by segment. Real world implementations of CNNs often glue an FFNN to the end to further process the data, which allows for highly non-linear abstractions. These networks are called DCNNs but the names and abbreviations between these two are often used interchangeably.

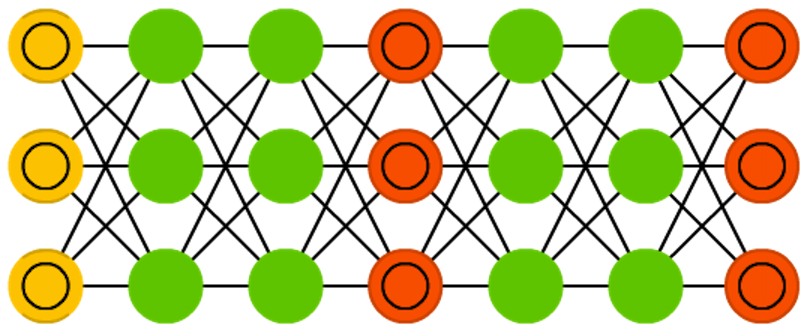


Generative adversarial networks (GAN) are from a different breed of networks, they are twins: two networks working together. GANs consist of any 2 networks (although often a combination of FFs and CNNs), with one tasked to generate content and the other has to judge content. The discriminating network receives either training data or generated content from the generative network. How well the discriminating network was able to correctly predict the data source is then used as part of the error for the generating network. This creates a form of competition where the discriminator is getting better at distinguishing real data from generated data and the generator is learning to become less predictable to the discriminator.

GANs can be quite difficult to train, as you don’t just have to train two networks but their dynamics need to be balanced as well. If prediction or generation becomes to good compared to the other, a GAN won’t converge as there is intrinsic divergence.

A generative adversarial network is an unsupervised machine learning algorithm that is a combination of two neural networks, one of which (network G) generates patterns and the other (network A) tries to distinguish genuine samples from the fake ones. Since networks have opposite goals – to create samples and reject samples – they start an antagonistic game that turns out to be quite effective.

GANs are used, for example, to generate photographs that are perceived by the human eye as natural images or deepfakes (videos where real people say and do things they have never done in real life).



Neural networks are used to solve complex problems that require analytical calculations similar to those of the human brain.

The most common uses for neural networks are:

1. Classification. NNs label the data into classes by implicitly analysing its parameters. For example, a neural network can analyse the parameters of a bank client such as age, solvency, credit history and decide whether to loan them money.
2. Prediction : The algorithm has the ability to make predictions. For example, it can foresee the rise or fall of a stock based on the situation in the stock market.
3. Recognition : This is currently the widest application of neural networks. For example, a security system can use face recognition to only let authorized people into the building.

References :

<https://serokell.io/blog/deep-learning-and-neural-network-guide>