Machine Learning

NIRDOSH BHATNAGAR

Deep Learning Tutorial

1. Introduction

Deep learning is a subset of machine learning, and machine learning in turn is a subset of artificial intelligence. See Figure 1.

- Artificial intelligence (AI): AI tries to imitate or simulate behavioral patterns of human beings or any other living organism.
- Machine learning (ML): ML is a scheme by which computing engines can "learn" from data. This learning is supposed to be accomplished without using a complex set of external rules. Further, this scheme consists of training, improving, and fine-tuning a model from data sets.
- Deep learning (DL): DL performs ML tasks which are inspired by a human being brain's complex network of neurons. Neurons, also called "neurones," are nerve cells. DL uses artificial neurons.

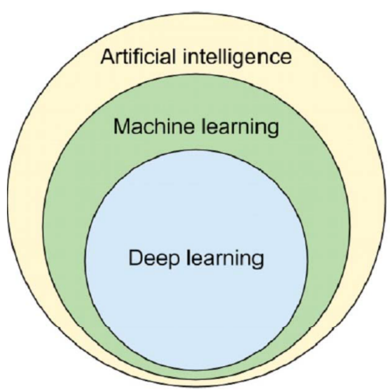

Figure 1. Relationship between AI, ML, and DL (Wikipedia: Machine Learning, 2021).

See Table 1, for a comparison of machine learning and deep learning schemes.

Machine Learning	Deep Learning
+ Good results with small data sets	— Requires very large data sets
+ Quick to train a model	— Computationally intensive
— Need to try different features and classifiers to achieve best results	+ Learns features and classifiers automatically
— Accuracy plateaus	+ Accuracy is unlimited

Table 1. Comparison of machine learning and deep learning schemes (MATLAB).

In this very brief tutorial, the following topics are discussed:

– Biological and artificial neurons.

– Comparison of conventional computers and biological neural networks

– Informal definition of deep learning.

– Performance of deep learning algorithms.

– History of deep learning.

– Applications of deep learning.

– Terminology for neural network architectures

– Examples artificial neural networks.

– Types of neural networks architectures.

– Training algorithms for artificial neural networks.

– Advantages and limitations of neural networks.

– Problem formulation

– References

2. Biological and Artificial Neurons

Biological neurons are initially discussed. This is followed by the discussion of an artificial neuron. An artificial neuron mimics a biological neuron.

Biological Neuron

As mentioned earlier, biological neurons are nerve cells. These are the fundamental units of the brain and nervous system. Their purpose is to detect patterns. The neurons receive sensory input from the outside world, and are responsible for transmitting motor commands to human beings' muscles. There are approximately 10^{11} neurons in a human brain, which in turn interact with cells of other types. The processing speed of the brain is 100 Hz

A neuron connects to other neurons to form a network of neurons. Note however, that each neuron cell communicates with only $1,000$ to $10,000$ other neurons. The components of interest of a neuron are: dendrites, synapses, axons, and the cell body (also called soma). See Figure 2.

- *Dendrites*: Dendrites manage inputs to the neuron cells. These receive input (called action potential) from the output (axon) of other neurons, and sometimes from the external world. Thus a dendrite is the receiving part of the neuron. A neuron might have thousands of dendrites. Note that the action potential is a brief electrical event.
- *Synapses*: These act as the weights to the input messages. These occur at the point of interconnection of one neuron with other neurons. The weights are also termed "synaptic weights." There are approximately 10^{14} synapses in the brain.
- *Cell body*: Cell body (called soma) is the computational unit of the biological neuron. Soma contains the nucleus. The nucleus can range from 3 to 18 μm in diameter. Soma sums all the incoming signals to generate input.
- *Axons*: When the sum reaches a certain threshold value, the neuron gets activated, and the signal travels down the axon towards other neurons. Thus axons carry the output of the biological neurons. Neurons are connected to other neurons via strands of fiber called axons (transmission lines). Consequently, the purpose of axons is to transmit nerve impulses between two neurons whenever the stimulation of neurons occur.

Artificial Neuron

An artificial neuron is supposed to mimic a biological neuron. The components of an artificial neuron are: inputs, weights, output, and summation and threshold operation.

- *Inputs*: The inputs are provided from an external sources, or from the output of other artificial neurons.
- *Weights*: These are the weights applied to the inputs.
- *Summation and threshold operation*: This function sums the weighted inputs, and then performs the thresholding operation, and produces the output.

– *Output*: The output of the artificial neuron sometimes provides input to other such neurons.

A single artificial neuron is also called a perceptron. A comparison of the biological and artificial neurons is provided in the following Table 2.

Biological neuron	Artificial neuron
Dendrites	Inputs
Synapses	Weights
Axons	Output
Cell body	Summation and thresholding

Table 2. Comparison of biological and artificial neuron.

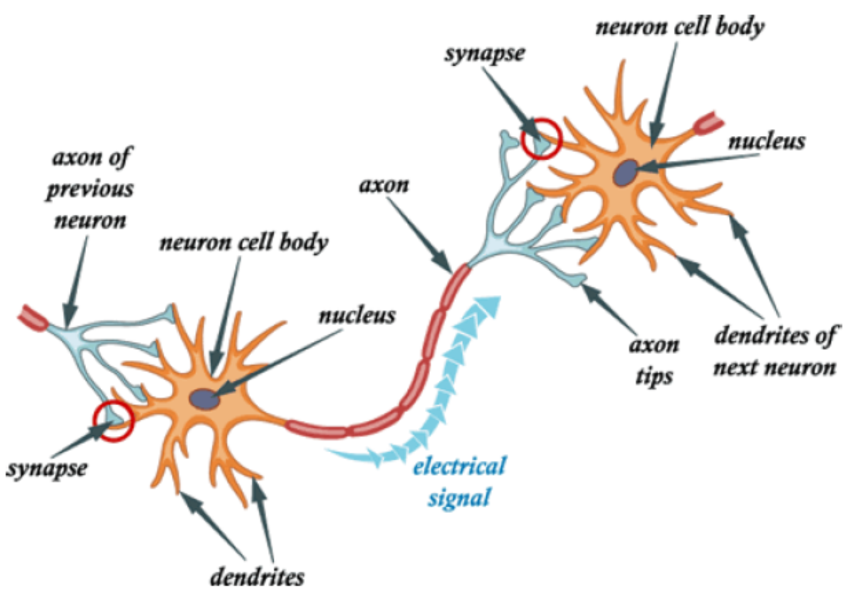

Figure 2. Components of a biological neuron, Lou (2019).

Implementation of an Artificial Neuron

Deep learning is implemented via a configuration of neural network. It is assumed that a logistic function can model a single neuron inside our brain. See Figure 3 (Lou, 2019), for an implementation of a single artificial neuron.

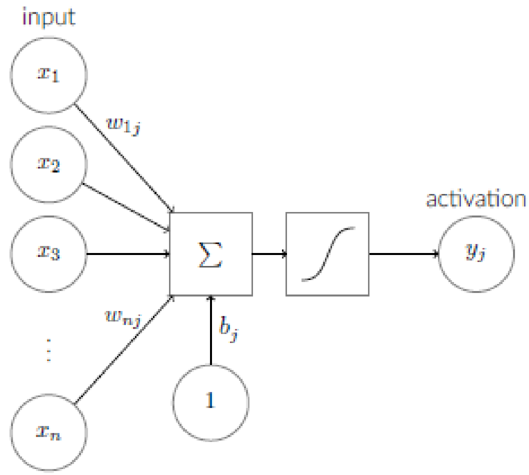

Figure 3. Components of an artificial neuron.

A neuron is a nonlinear transformation $\phi(\cdot)$ of a linear sum of its input.

- Let $x \in \mathbb{R}^n$ be the input vector of size $n \in \mathbb{P}$.
- Let $w \in \mathbb{R}^n$ be the weight vector of size $n \in \mathbb{P}$.
- Let $b \in \mathbb{R}$ be a scalar. It is also called the bias.
- Let $\phi : \mathbb{R} \to \mathbb{R}$ be the nonlinear function which models the neuron. Then

$$y = \phi\left(w^T x + b\right)$$

The function $\phi(\cdot)$ is often called the *activation function*.

Some possible examples of the activation function are as follows. Let $t \in \mathbb{R}$.

- Rectifiable linear unit.

$$\phi(t) = \max(t, 0)$$

- Heaviside function.

$$\phi(t) = \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{if } t \leq 0 \end{cases}$$

The neurons which use this function for activation are called perceptrons.

- Sigmoid function $\sigma(\cdot)$.

$$\phi(t) = \frac{1}{1 + e^{-t}} \triangleq \sigma(t)$$

The neurons which use this function for activation are called sigmoid neurons.

– Tan hyperbolic function.

$$\phi(t) = \tanh(t)$$

Note that the tan hyperbolic and the sigmoid function are related as

$$2\sigma(2t) = 1 + \tanh(t)$$

– Softmax function $\phi(\cdot)$.

$$\phi(t) = \ln\left(1 + e^t\right)$$

– Linear function.

$$\phi(t) = t$$

3. Comparison of Conventional Computers and Biological Neural Networks

The following comparison of conventional computers and biological neural networks in Table 3 is due to Coolen (1998).

Conventional computers	Brain
Processors	Biological neural networks
Silicon transistors	Neurons
Size of a single transistor ~ 9 *to* 10 *microns*	*Size* ~ 6 *to* 10 *microns*
Operation speed $\sim 10^8$ *Hz*	*Operation speed* $\sim 10^2$ *Hz*
Signal / Noise $\sim \infty$	*Signal / Noise* ~ 1
Signal velocity $\sim 10^8$ *m/sec*	*Signal velocity* ~ 1 *m/sec*
Connections ~ 10	*Connections* $\sim 10^4$
Operation	Operation
Sequential operation	*Parallel operation*
Program and data	*Connections, neuron thresholds*
External programming	*Self-programming & adaptation*
Programming capability	Learning capability
Hardware failure	Hardware failure
Fatal	*Robust against hardware failure*
No unforseen data	*Messy, unforseen data*

Table 3. Comparison of conventional computers and biological neural networks.

4. Informal Definitions of Deep Learning

- Deep learning is a subset of machine learning algorithms.
- Deep learning is also known as *deep structured learning* or *hierarchical learning*.
- Deep learning algorithms use a cascade of multiple layers of nonlinear processing units. Each layer uses the output of the previous layer in this cascade (except the first layer) as its input. The first layer in the cascade uses the input data.
  - The multiple layers can be mapped to different levels of abstraction of the input data, where the levels form a hierarchy of concepts.
  - The goal of the algorithm is feature abstraction and transformation.
- The phrase "deep" in deep learning alludes to the number of layers in the deep learning algorithm.
- The multiple layers between input and output layers, perform feature identification and processing in a series of stages. It is assumed that this is similar to the way our brain appears to process information.
- Each layer is made up of several artificial neurons.

5. Performance of Deep Learning Algorithms

Why do we want to learn deep learning? Deep learning is good at clustering algorithms, classification tasks, and predictive analysis. The success of deep learning algorithms is next discussed.

- Success of deep learning algorithms depends upon:
  - Larger data sets.
  - More computing power.
  - Better optimization techniques.
  - Greedy training in each layer of the neural network
- Learning can be supervised, unsupervised, or semisupervised.
- To a certain extent the deep learning models have drawn inspiration from models of biological nervous systems.

The performance of deep learning algorithms, when compared to old machine learning algorithms is depicted in the Figure 4. This figure is due to Alom et al, 2019.

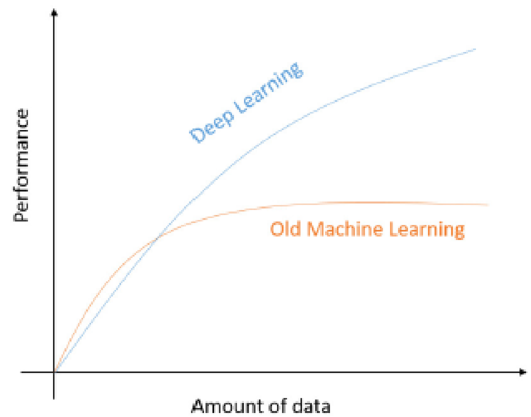

Figure 4. Performance of deep learning with respect to the amount of data.

6. History of Deep Learning

History of deep learning in tabular (Table 4) form due to Wang and Raj (2017).

Table 1: Major milestones that will be covered in this paper

Year	Contributer	Contribution
300 BC	Aristotle	introduced Associationism, started the history of human's attempt to understand brain.
1873	Alexander Bain	introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule.
1943	McCulloch & Pitts	introduced MCP Model, which is considered as the ancestor of Artificial Neural Model.
1949	Donald Hebb	considered as the father of neural networks, introduced Hebbian Learning Rule, which lays the foundation of modern neural network.
1958	Frank Rosenblatt	introduced the first perceptron, which highly resembles modern perceptron.
1974	Paul Werbos	introduced Backpropagation
1980	Teuvo Kohonen	introduced Self Organizing Map
	Kunihiko Fukushima	introduced Neocogitron, which inspired Convolutional Neural Network
1982	John Hopfield	introduced Hopfield Network
1985	Hilton & Sejnowski	introduced Boltzmann Machine
1986	Paul Smolensky	introduced Harmonium, which is later known as Restricted Boltzmann Machine
	Michael I. Jordan	defined and introduced Recurrent Neural Network
1990	Yann LeCun	introduced LeNet, showed the possibility of deep neural networks in practice
1997	Schuster & Paliwal	introduced Bidirectional Recurrent Neural Network
	Hochreiter & Schmidhuber	introduced LSTM, solved the problem of vanishing gradient in recurrent neural networks
2006	Geoffrey Hinton	introduced Deep Belief Networks, also introduced layer-wise pretraining technique, opened current deep learning era.
2009	Salakhutdinov & Hinton	introduced Deep Boltzmann Machines
2012	Geoffrey Hinton	introduced Dropout, an efficient way of training neural networks

Table 4. History of deep learning in a tabular form.

Important landmarks in the deep learning golden era, as per Gavves (2020), are shown in the Figure 5.

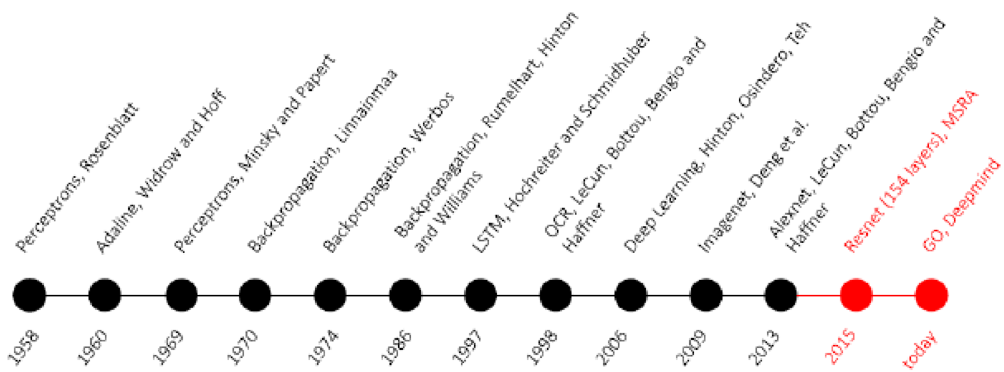

Figure 5. Deep learning landmarks.

A pictorial history of the important contributors of deep learning is shown in the Figure 6 (Beam, 2021).

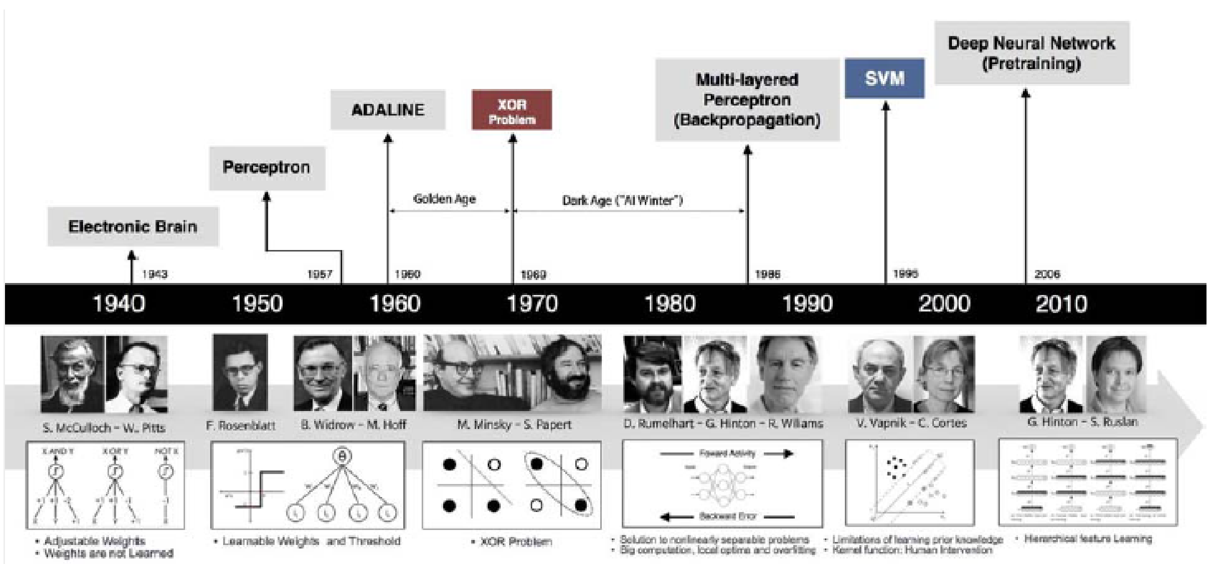

Figure 6. An incomplete pictorial history of the important contributors of deep learning.

The 2018 Turing prize was awarded to Yoshio Bengio, Geoffery Hinton, and Yann Lecun; for their work on deep learning techniques. Turing prize is the highest prize a person can get for their work in computer science. It it equivalent of the Noble prize in medicine, physics, chemistry, peace, literature, and economics.

7. Applications of Deep Learning

The following figures (Figures 7 & 8) specify the variegated applications of deep learning. These figures are due to Texas Instruments (2018).

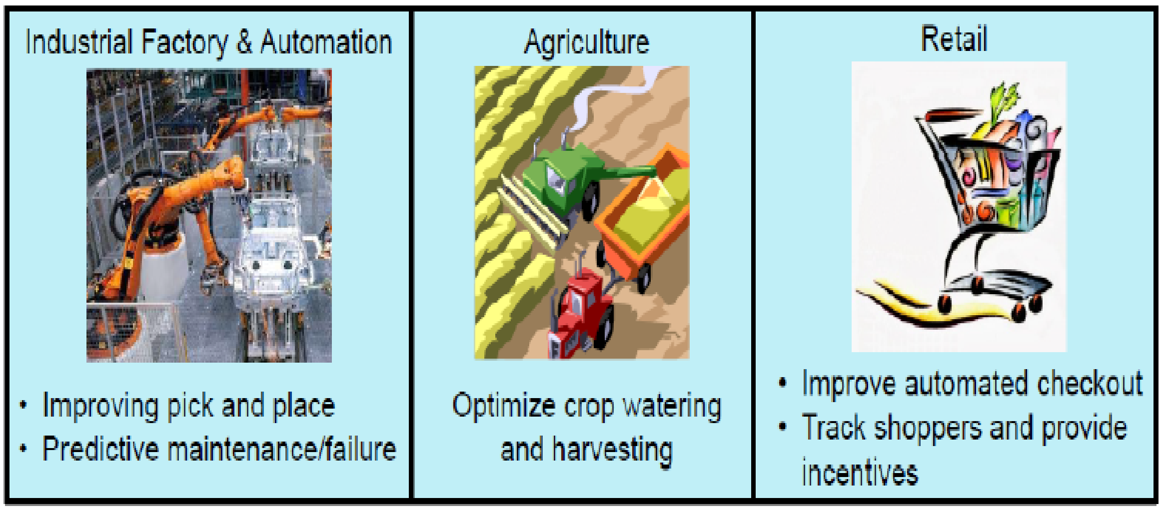

Figure 7. Possible applications of deep learning.

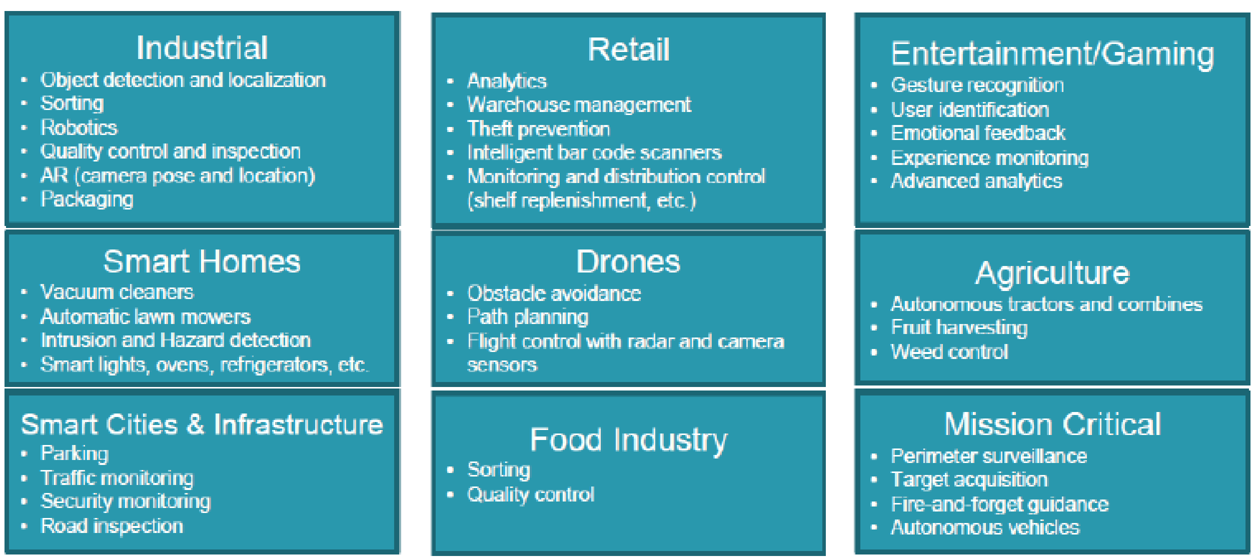

Figure 8. Some applications of deep learning.

Some specific applications of deep learning.

- Audio recognition
- Bioinformatics
- Board game programs
- Computer vision
- Customer relationship management
- Drug discovery and toxicology
- Financial fraud detection
- Image recognition and restoration, object recognition, optical character recognition
- Machine translation
- Military
- Mobile advertising
- Natural language processing
- Recommendation systems
- Social network filtering
- Speech recognition, music retrieval, transcription
- Text domain: Parsing, sentiment analysis, machine translation
- Visual art processing

8. Terminology for Neural Network Architectures

An artificial neural network consists of a large number of neurons. These neurons are generally organized in a sequence of layers. A typical such ANN consists of three types of layers. These are: the input layer, the output layer, and the hidden layers.

– Input layer: This layer consists of neurons which receive input directly from the external world. Based upon this external input, the ANN will either learn, or recognize, or process.

– Output layer: The neurons in this layer provide the output of the ANNs.

– Hidden layer: A hidden layer is between the input and output layers of the ANN. There can be more than a single hidden layer in an ANN.

In a *connected neural network*, a neuron is connected to each and every neuron in the immediately preceding layer.

9. Examples of Artificial Neural Networks

Some examples of multilayered artificial neural networks are shown.

Example 1 A three-layered artificial neural network (ANN) is shown in the Figure 9 (Maarten, 2021). A neuron (or equivalently an activation function) is specified by a circle.

The circles in the first column specify the input layer. This layer has three neurons.

The circles in the second column represent the hidden layer of the ANN. This layer has four neurons.

The circles in the third column represent the output layer. This layer has two neurons.

Note that, in this example, artificial neurons are always processed in the forward direction. Therefore, this is an example of a *feed-forward neural network.* It is also a connected neural network.

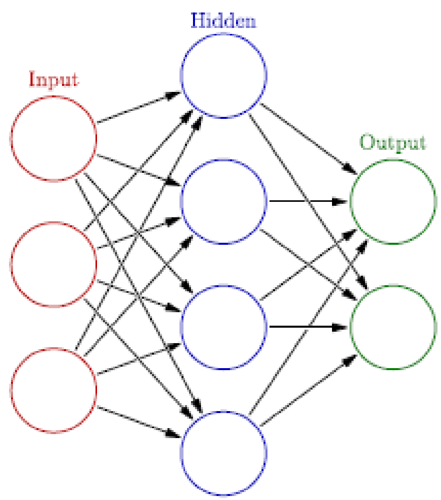

Figure 9. A three-layered ANN.

□

Example 2. A five-layered artificial neural network (ANN) is shown in the Figure 10 (Gavves, 2020). A neuron is specified by a circle.

The circles in the first column specify the input layer. This layer has eight neurons.

The circles in the second, third, and fourth columns represent the hidden layers of the ANN. Thus there are three hidden layers in this ANN. Each hidden layer has nine neurons.

The circles in the fifth column represent the output layer. This layer has four neurons.

This network is also a feed-forward neural network. It is also a connected neural network.

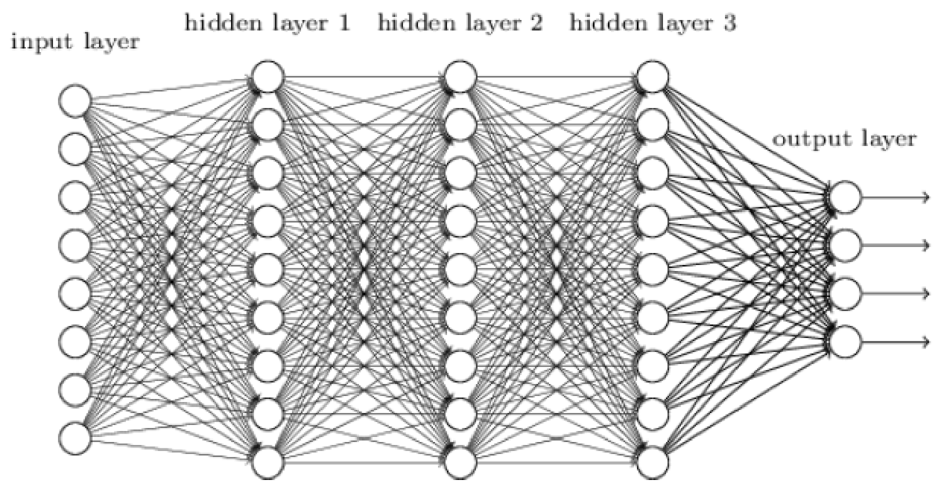

Figure 10. A fully-connected, five-layered ANN.

□

10. Types of Neural Network Architectures

A neural network is a configuration of artificial neurons, which are connected in some useful and interesting manner.

Neural networks are often organized in layers. The series of layers between the input and output of the neural network, perform feature identification and processing in stages. It is conjured that our brains perform in a similar manner.

We shall see that 'deep learning' means using a neural network with multiple layers of artificial neurons between the input and output layers.

Some of the different types of deep learning architectures are:

- Feed-forward neural network (FNN)
  - Single-layer feed-forward neural network
  - Multilayer feed-forward neural network
- Recurrent neural network (RNN)
- Convolutional deep neural network (CNN)
- Deep belief neural network (DBN)
- Long short-term memory neural network (LSTM)
- Hopfield network
- Boltzmann machine neural network
- Modular neural network
- Radial basis function neural network

Feed-Forward Neural Network

– There are two types of feed-forward neural networks. These are single-layer and multi-layer feed-forward neural networks.

– Single-layer feed-forward neural networks have only a single layer of neurons.

– Multi-layer feed-forward neural networks have more than a single layer of neurons. Such networks are also called multi-layer perceptrons (MLPs), and deep feed-forward neural networks.

– MLP architecture is one of the most traditional types of architectures.

– Each neuron of the previous layer is connected to each element of the subsequent layer of the configuration. Layers of this type are called dense layer.

– Such networks are hard to train. This is a significant disadvantage.

Recurrent Neural Network

– In RNNs, feedback is added to feed-forward neural networks.

– Hidden-layer neurons have self-connections.

– Possess memory.

– Internal memory is used to process data sequences of arbitrary length.

– Use

  – It is eminently suitable for processing time series data where current characteristics depend upon the past features.

  – Suitable for text and speech analysis.

See Figure 11 (Pai, 2020) for a comparative study of FNN and RNN.

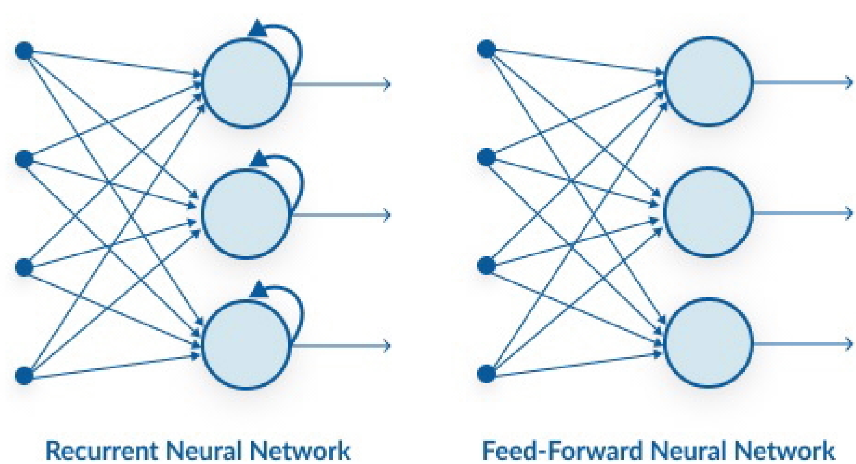

Figure 11. Comparison of feed-forward neural networks and recurrent neural networks.

Convolutional Deep Neural Network

- Convolution deep neural networks are a special type of feed-forward neural network.
- Process fixed-size inputs and generate fixed-size outputs.
- Unsupervised pre-training is not necessary.
- Learning is discriminative.
- Utilizes spatial / temporal relationships in order to reduce the load on learning.
- Use
  - Suitable for image and video processing of signals.
  - Useful for computer visual applications for object detection, classification, and semantic segmentation.

A comparison of MLP, RNN, and CNN is shown in the Table 5 (Pai, 2020).

	MLP	RNN	CNN
Data	Tabular data	Sequential data	Image data
Recurrent connections	No	Yes	No
Parameter sharing	No	Yes	Yes
Spatial relationship	No	No	Yes
Vanishing & Exploding Gradient	Yes	Yes	Yes

Table 5. Comparison of MLP, RNN, and CNN.

Deep Belief Neural Network

- Unsupervised pre-training is helpful.
- Learning is generative.
- Uses multi-layered recurrent neural network which uses energy minimizing methods.

Long Short-Term Memory Neural Network

- Memory cell is added to hidden-layer neurons of LSTM type of neural networks.

Hopfield Network

- Hopfield network, is a completely connected network of neurons, in which each neuron is connected to every other neuron.
- The neurons in this network are trained with input patterns by setting a value of neurons to the specified design pattern. The weights are then computed.

Boltzmann Machine Neural Network

- Boltzmann machine neural network are analogous to the Hopfield neural network, except that some neurons are input, and other neurons are hidden.
- The weights in this neural network are initialized randomly, and learning takes place via the backpropagation algorithm.

Modular Neural Network

- Modular neural network is a combination of different types of neural networks like: MLP, RNN, CNN, and Hopfield network.
- These neural networks of different types are amalgamated into a single network module to execute independent subtasks of the complete neural network.

Radial Basis Function Neural Network

- Radial basis function neural networks are analogous to the feed-forward neural network, except that radial function is used as a neuron's activation function.

11. Training Algorithms for Artificial Neural Networks

In order for the ANNs to execute successfully, the weights in the network have to be determined efficiently. Two popular training algorithms for ANNs are:

- Gradient descent algorithm
- Back propagation algorithm

Gradient Descent Algorithm

Characteristics of the gradient descent algorithm:

- Gradient descent algorithm is a simple and popular method for training weights in ANNs.
- It is used in the case of supervised training model. If the computed output is different from the target output, the difference of the error is determined.
- The gradient descent algorithm modifies the weights of the network so that the error in the output is minimized.

Back-Propagation Algorithm

Characteristics of the back-propagation algorithm:

- The back-propagation algorithm is an extension of the gradient-based delta learning rule.
- In this case, after the determination of an error (which is the difference between the computed and the target output), the error is propagated backward from the output layer to the input layer via the hidden layer.
- As can be observed, this algorithm finds use in the case of multi-layer neural networks.

12. Advantages and Limitations of Neural Networks

We discuss advantages and limitations of neural networks in this section

Advantages of Neural Networks

Some of the advantages of neural networks are as follows.

- A neural network can execute a task, which a linear model cannot.
- If a neuron in the network fails, then the parallelism of the ANNs will not curtail the execution of the specified task.
- A neural network learns, and reprogramming is generally unnecessary.
- Neural networks can be used for any application.
- It is not problematic in general.

Limitations of Neural Networks

Some of the limitations of neural networks are as follows.

- Training is necessary for the neural networks to operate properly.
- Processing time is high for large neural networks.
- As the architecture of microprocessors and neural networks are different, emulation is required.

13. Problem Formulation

An attempt at a problem formulation of deep learning is ventured upon in this section. In order to characterize an ANN, the weights have to be determined. Before, this is done, the ANN has to be described carefully.

As per Shai Shalev-Shwartz (2014), the goal of *learning* is to get an accurate mapping $f : X \to Y$ based upon the data set

$$\mathcal{D} = \{(x_i, y_i) \mid x_i \in X, y_i \in Y; 1 \leq i \leq n\}$$

In deep learning, each mapping $f : X \to Y$ is parameterized by a weight vector $w \in \mathbb{R}^d$. So the immediate goal of deep learning is to learn the vector w.

As per Gavves (2020), the goal of deep learning via artificial neural networks is to generate a family of parametric, nonlinear and hierarchical representation learning functions, which are massively optimized with stochastic gradient to encode domain knowledge, which are domain invariances.

A definition of a neural network, in which artificial neurons are arranged in layers is given as per Sun (2020).

Definition. *Feed-Forward Neural network with layers (MLP). Let*:

- *X is the set of input vectors (data points), where $X \subset \mathbb{R}^{d_X}$, and $d_X \in \mathbb{P}$ is the dimension of the data vector.*

- *Y is the set of output vectors, where $Y \subset \mathbb{R}^{d_Y}$, and $d_Y \in \mathbb{P}$ is the dimension of the output vector.*
- *The set of data points is: $\mathcal{D} = \{(x_i, y_i) \mid x_i \in X, y_i \in Y; 1 \leq i \leq n\}$.*
- *The neural network mapping function is $f_\theta : \mathbb{R}^{d_X} \to \mathbb{R}^{d_Y}$. It maps an input x to a predicted output $\widehat{y}$. Note that θ is the parameter of the mapping.*
- *$\phi : \mathbb{R} \to \mathbb{R}$ is the nonlinear function which models a single neuron. It is the activation function.*
- *L is the total number of layers. The input layer is the layer number 1, and the output layer is layer number L. The hidden layers are numbered 2 through $(L-1)$.*
- *W^j is a weight matrix of dimension $d_j \times d_{j-1}$, where $j = 1, 2, \ldots, L$, where $d_0 = d_X$, and $d_L = d_Y$.*
- *$W = \left(W^1, W^2, \ldots, W^L\right)$ is the collection of all weight matrices.*
- *b^j is the bias vector of size d_j. Thus $b^j \in \mathbb{R}^{d_j}$, where $j = 1, 2, \ldots, L$.*
- *$B = (b_1, b_2, \ldots, b_L)$ is the collection of all bias vectors.*
- *$\theta = (W, B)$ represents the collection of all parameters.*
- *$\phi^j : \mathbb{R}^{d_j} \to \mathbb{R}^{d_j}$, where if $a = \left(a_1, a_2, \ldots, a_{d_j}\right)^T \in \mathbb{R}^{d_j}$, then*

  $$\phi^j(a) = \left(\phi(a_1), \phi(a_2), \ldots, \phi\left(a_{d_j}\right)\right)^T$$

  and $j = 1, 2, \ldots, L$.

  The neural network is specified via the following recursion. Let $x \in X$, the

  $$\begin{aligned} z^0 &= x \\ z^l &= \phi^l\left(W^l z^{l-1} + b^l\right), \quad l = 1, 2, \ldots, L \end{aligned}$$

  Note that $z^l \in \mathbb{R}^{d_l}$, where $l = 0, 1, 2, \ldots, L$.
- *Thus the complete neural network mapping is parametrized by θ.* □

The neural network parameters are picked so that for a data point (x, y), the predicted output $\widehat{y} = f_\theta(x)$ should be such that it is close to the true output y. Thus the distance between y and $\widehat{y}$ has to be minimized. Let $\ell(\cdot, \cdot)$ be a distance metric. The problem of determining the optimal value of the parameter θ is specified as

$$\min_\theta F(\theta), \quad \text{where} \quad F(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f_\theta(x_i))$$

For regression problem, $\ell(y, z)$ is typically the quadratic metric $\|y - z\|^2$. For binary classification, $\ell(y, z) = \ln(1 + \exp(-yz))$.

14. **References**

1. Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., and Asari, V. K.; 2019. "A State-of-the-Art Survey on Deep Learning Theory and Architectures," Electronics 2019, 8, 292; doi:10.3390/electronics8030292, www.mdpi.com/journal/electronics.

2. Beam, A., 2021. "Deep Learning 101 - Part 1: History and Background." Retrieved on July 2021, from URL, https://www.programmersought.com/article/54453361399/

3. Coolen, A. C. C., 1998. "A Beginner's Guide to the Mathematics of Neural Networks," in *Concepts for Neural Networks*, - Springer, 10.1.1.161.3556.pdf.

4. Gavves, E., 2020. "Overview," UVA Deep Learning course, University of Amsterdam.

5. Gill, N. S., 2021. "Artificial Neural Networks Applications and Algorithms," Xenonstack, A Stack Innovator.

6. Grachten, Maarten, 2021. "A very brief overview of deep learning," overview.pdf.

7. Kutynick, G., 2018. "Mathematics of Deep Neural Network," AIT Workshop of the Information Theory Group, Technische Universität Berlin, May 4, 2018.

8. Lou, C., 2019. "Artificial Neural Networks: their Training Process and Applications," Lou-Hundley.pdf.

9. Mathworks, 2012. "Mathworks: Introducing Deep Learning with MATLAB," deep_learning_ebook.pdf, Retrieved on July 2021.

10. Pai, A., 2020. ANN vs CNN vs RNN _ Types of Neural Networks, Retrieved on July 2021, from URL: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

11. Shalev-Shwartz, S., 2014. "Accelerating Stochastic Optimization," "Master Class at Tel-Aviv," Tel-Aviv University, November 2014.

12. Sun, R., 2020. "Optimization for Deep Learning: An Overview," University of Illinois at Urbana-Champaign, Urbana, IL, USA.

13. Texas Instruments, 2018. "Introduction to Deep Learning," introduction-to-deep-learning.pdf, Retrieved in July 2021.

14. Wikipedia: Machine Learning, 2021. Machine_learning.pdf, Retrieved on July 2021, from URL: https://en.wikipedia.org/w/index.php?title=Special:DownloadAsPdf&page=Machine_learning&action=show-download-screen