# Python Deep Learning

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

## About the Tutorial

Python is a general-purpose high level programming language that is widely used in data science and for producing deep learning algorithms.

This brief tutorial introduces Python and its libraries like Numpy, Scipy, Pandas, Matplotlib; frameworks like Theano, TensorFlow, Keras. The tutorial explains how the different libraries and frameworks can be applied to solve complex real world problems.

## Audience

This tutorial has been prepared for professionals aspiring to learn the basics of Python and develop applications involving deep learning techniques such as convolutional neural nets, recurrent nets, backpropagation, etc.

## Prerequisites

Before you proceed with this tutorial, we assume that you have prior exposure to Python, Numpy, Pandas, Scipy, Matplotib, Windows, any Linux distribution, prior basic knowledge of Linear Algebra, Calculus, Statistics and basic machine learning techniques.

## Copyright &Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

# Table of Contents

# 1. Python Deep Learning – Introduction

Deep structured learning or hierarchical learning or deep learning in short is part of the family of machine learning methods which are themselves a subset of the broader field of Artificial Intelligence.

Deep learning is a class of machine learning algorithms that use several layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.

Deep neural networks, deep belief networks and recurrent neural networks have been applied to fields such as computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, and bioinformatics where they produced results comparable to and in some cases better than human experts have.

Deep Learning Algorithms and Networks -

- are based on the unsupervised learning of multiple levels of features or representations of the data. Higher-level features are derived from lower level features to form a hierarchical representation.

- use some form of gradient descent for training.

# 2. Python Deep Learning – Environment

In this chapter, we will learn about the environment set up for Python Deep Learning. We have to install the following software for making deep learning algorithms.

- Python 2.7+
- Scipy with Numpy
- Matplotlib
- Theano
- Keras
- TensorFlow

It is strongly recommend that Python, NumPy, SciPy, and Matplotlib are installed through the Anaconda distribution. It comes with all of those packages.

We need to ensure that the different types of software are installed properly.

Let us go to our command line program and type in the following command:

```
$ python
Python 3.6.3 |Anaconda custom (32-bit)| (default, Oct 13 2017, 14:21:34)
[GCC 7.2.0] on linux
```

Next, we can import the required libraries and print their versions:

```
import numpy
print numpy.__version__
```

### Output

```
1.14.2
```

## Installation ofTheano, TensorFlow and Keras

Before we begin with the installation of the packages - Theano, TensorFlow and Keras, we need to confirm if the **pip** is installed. The package management system in Anaconda is called the pip.

To confirm the installation of pip, type the following in the command line:

```
$ pip
```

Once the installation of pip is confirmed, we can install TensorFlow and Keras by executing the following command:

```
$pip install theano
$pip install tensorflow
$pip install keras
```

Confirm the installation of Theano by executing the following line of code:

```
$python –c “import theano: print (theano.__version__)”
```

## Output

```
1.0.1
```

Confirm the installation of Tensorflow by executing the following line of code:

```
$python –c “import tensorflow: print tensorflow.__version__”
```

## Output

```
1.7.0
```

Confirm the installation of Keras by executing the following line of code:

```
$python –c “import keras: print keras.__version__”
Using TensorFlow backend
```

## Output

```
2.1.5
```

# 3. Python Deep Learning – Basic Machine Learning

Artificial Intelligence (AI) is any code, algorithm or technique that enables a computer to mimic human cognitive behaviour or intelligence. Machine Learning (ML) is a subset of AI that uses statistical methods to enable machines to learn and improve with experience. Deep Learning is a subset of Machine Learning, which makes the computation of multi-layer neural networks feasible. Machine Learning is seen as shallow learning while Deep Learning is seen as hierarchical learning with abstraction.

Machine learning deals with a wide range ofconcepts. The concepts are listed below:

- supervised
- unsupervised
- reinforcement learning
- linear regression
- cost functions
- overfitting
- under-fitting
- hyper-parameter, etc.

In supervised learning, we learn to predict values from labelled data. One ML technique that helps here isclassification, where target values are discrete values; for example,cats and dogs. Another technique in machine learning that could come of help is regression.Regression works onthe target values. The target values are continuous values; for example, the stock market data can be analysedusing Regression.

In unsupervised learning, we make inferences from the input data that is not labelled or structured. If we have a million medical records and we have to make sense of it, find the underlying structure,outliers or detect anomalies, we use clustering technique to divide data into broad clusters.

Data sets are divided into training sets, testing sets, validation sets and so on.

A breakthrough in 2012 brought the concept of Deep Learning into prominence. An algorithm classified 1 million images into 1000 categories successfully using 2 GPUs and latest technologies like Big Data.

## Relating Deep Learning and Traditional Machine Learning

One of the major challenges encountered in traditional machine learning models is a process called feature extraction. The programmer needs to be specific and tell the computer the features to be looked out for. These features will help in making decisions.

Entering raw data into the algorithm rarely works, so feature extraction is a critical part of the traditional machine learning workflow.

This places a huge responsibility on the programmer, and the algorithm's efficiency relies heavily on how inventive the programmer is. For complex problems such as object recognition or handwriting recognition, this is a huge issue.

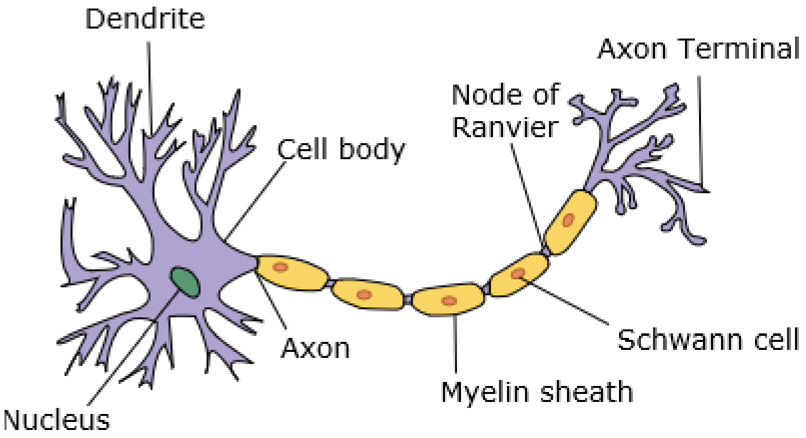
Deep learning, with the ability to learn multiple layers of representation, is one of the few methods that has help us with automatic feature extraction. The lower layers can be assumed to be performing automatic feature extraction, requiring little or no guidance from the programmer.

# 4. Python Deep Learning – Artificial Neural Networks

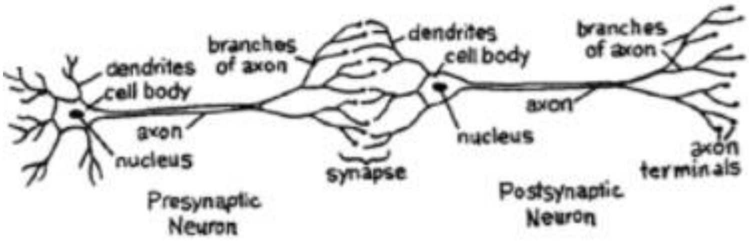
The Artificial Neural Network, or just neural network for short, is not a new idea. It has been around for about 80 years.

It was not until 2011, when Deep Neural Networks became popular with the use of new techniques, huge dataset availability, and powerful computers.

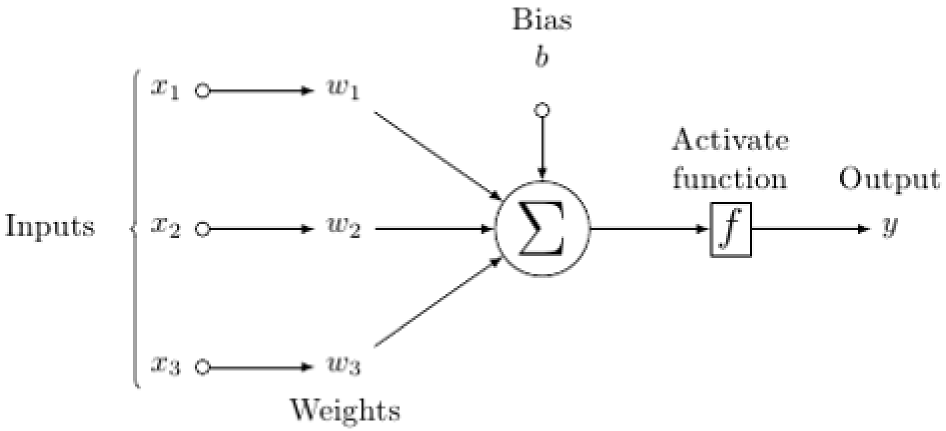
A neural network mimics a neuron, which has dendrites, a nucleus, axon, and terminal axon.



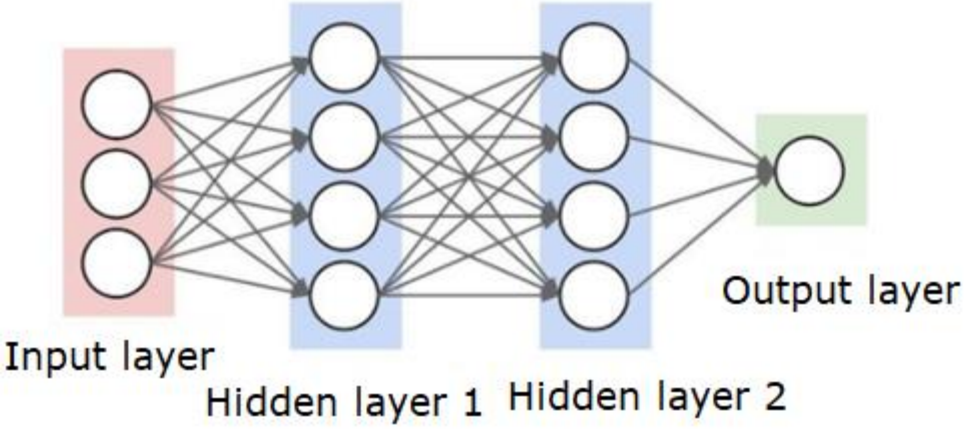
For a network, we need two neurons. These neurons transfer information via synapse between the dendrites of one and the terminal axon of another.



A probable model of an artificial neuron looks like this:



A neural network will look like as shown below:



The circles are neurons or nodes, with their functions on the data and the lines/edges connecting them are the weights/information being passed along.

Each column is a layer. The first layer of your data is the input layer. Then, all the layers between the input layer and the output layer are the hidden layers.

If you have one or a few hidden layers, then you have a shallow neural network. If you have many hidden layers, then you have a deep neural network.

In this model, you have input data, you weight it, and pass it through the function in the neuron that is called threshold function or activation function.

Basically, it is the sum of all of the values after comparing it with a certain value. If you fire a signal, then the result is (1) out, or nothing is fired out, then (0). That is then weighted and passed along to the next neuron, and the same sort of function is run.

We can have a sigmoid (s-shape) function as the activation function.

As for the weights, they are just random to start, and they are unique per input into the node/neuron.

In a typical "feed forward", the most basic type of neural network, you have your information pass straight through the network you created, and you compare the output to what you hoped the output would have been using your sample data.

From here, you need to adjust the weights to help you get your output to match your desired output.

The act of sending data straight through a neural network is called a **feed forward neural network.**

Our data goes from input, to the layers, in order, then to the output.

When we go backwards and begin adjusting weights to minimize loss/cost, this is called **back propagation**.

This is an **optimization problem**. With the neural network, in real practice, we have to deal with hundreds of thousands of variables, or millions, or more.

The first solution was to use stochastic gradient descent as optimization method. Now, there are options like AdaGrad, Adam Optimizer and so on. Either way, this is a massive computational operation. That is why Neural Networks were mostly left on the shelf for over half a century. It was only very recently that we even had the power and architecture in our machines to even consider doing these operations, and the properly sized datasets to match.

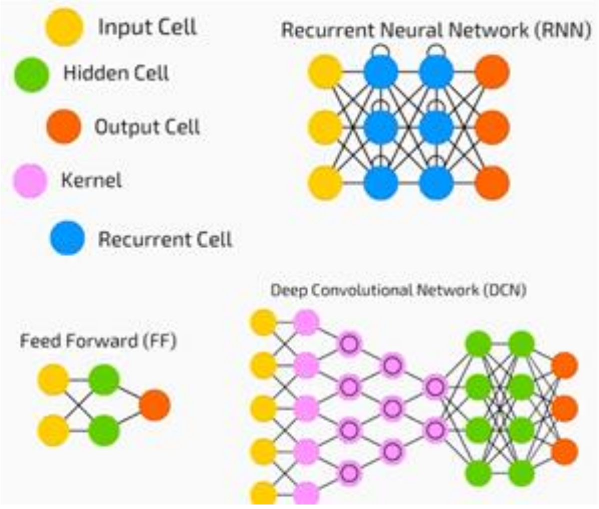
For simple classification tasks, the neural network is relatively close in performance to other simple algorithms like K Nearest Neighbors. The real utility of neural networks is realized when we have much larger data, and much more complex questions, both of which outperform other machine learning models.

# 5. Python Deep Learning – Deep Neural Networks

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks.

We have an input, an output, and a flow of sequential data in a deep network.



Neural networks are widely used in supervised learning and reinforcement learning problems. These networks are based on a set of layers connected to each other.

In deep learning, the number of hidden layers, mostly non-linear, can be large; say about 1000 layers.

DL models produce much better results than normal ML networks.

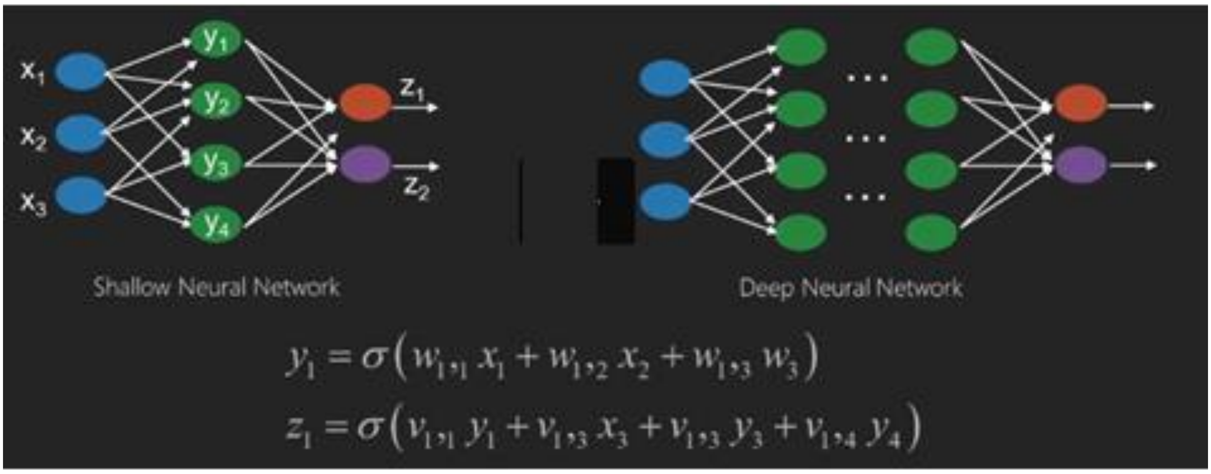
We mostly use the gradient descent method for optimizing the network and minimising the loss function.

We can use the **Imagenet**, a repository of millions of digital images to classify a dataset into categories like cats and dogs. DL nets are increasingly used for dynamic images apart from static ones and for time series and text analysis.

Training the data sets forms an important part of Deep Learning models. In addition, Backpropagation is the main algorithm in training DL models.

DL deals with training large neural networks with complex input output transformations.

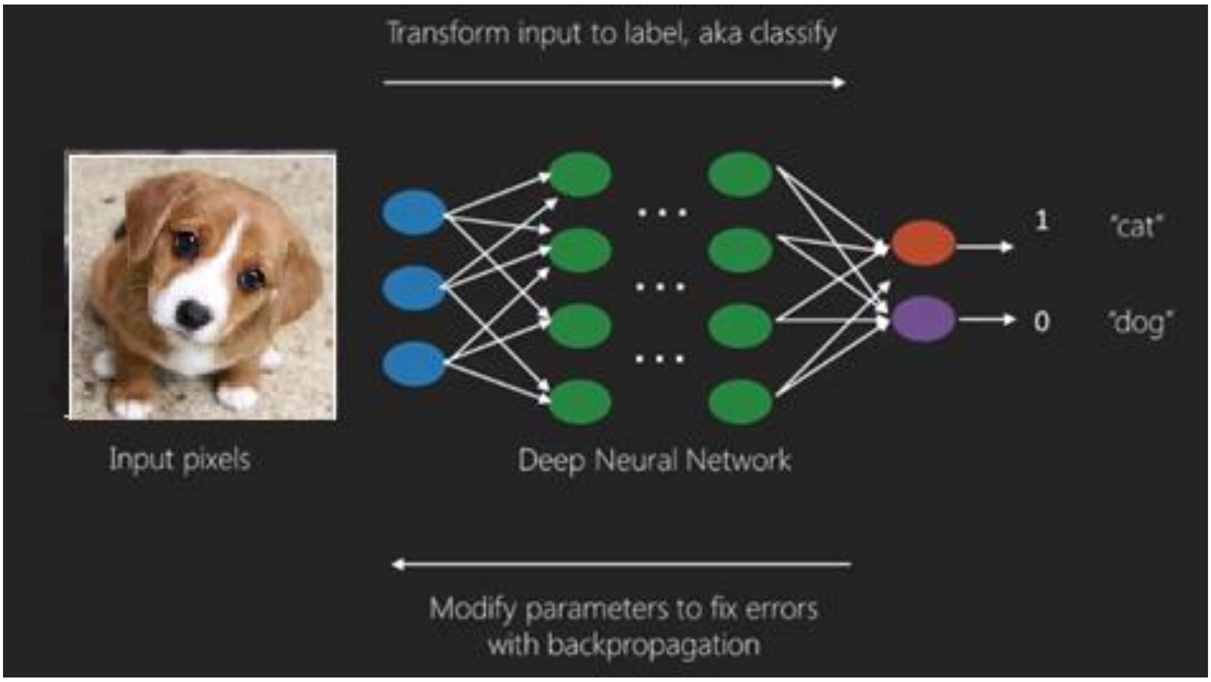
One example of DL is the mapping of a photo to the name of the person(s) in photo as they do on social networks and describing a picture with a phrase is another recent application of DL.


$x_1$
$x_2$
$x_3$
$y_1$
$y_2$
$y_3$
$y_4$
$z_1$
$z_2$
Shallow Neural Network
Deep Neural Network
$y_1 = \sigma\left(w_{1,1} x_1 + w_{1,2} x_2 + w_{1,3} w_3\right)$
$z_1 = \sigma\left(v_{1,1} y_1 + v_{1,3} x_3 + v_{1,3} y_3 + v_{1,4} y_4\right)$

Neural networks are functions that have inputs like x1,x2,x3…that are transformed to outputs like z1,z2,z3 and so on in two (shallow networks) or several intermediate operations also called layers (deep networks).

The weights and biases change from layer to layer. 'w' and 'v' are the weights or synapses of layers of the neural networks.

The best use case of deep learning is the supervised learning problem.Here,we have large set of data inputs with a desired set of outputs.


Transform input to label, aka classify
Input pixels
Deep Neural Network
1
"cat"
0
"dog"
Modify parameters to fix errors
with backpropagation

Here we apply backpropagation algorithm to get correct output prediction.

The most basic data set of deep learning is the MNIST, a dataset of handwritten digits.

We can train deep a Convolutional Neural Network with Keras to classify images of handwritten digits from this dataset.

The firing or activation of a neural net classifier produces a score. For example,to classify patients as sick and healthy,we consider parameters such as height, weight and body temperature, blood pressure etc.

A high score means patient is sick and a low score means he is healthy.

Each node in output and hidden layers has its own classifiers. The input layertakes inputs and passes on its scores to the next hidden layer for further activation and this goes on till the output is reached.

This progress from input to output from left to right in the forward direction is called **forward propagation**.

Credit assignment path (CAP) in a neural network is the series of transformations starting from the input to the output. CAPs elaborate probable causal connections between the input and the output.

CAP depth for a given feedforward neural network or the CAP depth is the number of hidden layers plus one as the output layer is included. For recurrent neural networks, where a signal may propagate through a layer several times, the CAP depth can be potentially limitless.

## Deep Nets and Shallow Nets

There is no clear threshold of depth that divides shallow learning from deep learning; but it is mostly agreed that for deep learning which has multiple non-linear layers, CAP must be greater than two.

Basic node in a neural net is a perceptron mimicking a neuron in a biological neural network. Then we have multi-layered Perceptron or MLP. Each set of inputs is modified by a set of weights and biases; each edge has a unique weight and each node has a unique bias.

The prediction **accuracy** of a neural net depends on its **weights and biases**.

The process of improving the accuracy of neural network is called **training**. The output from a forward prop net is compared to that value which is known to be correct.

The **cost function or the loss function** is the difference between the generated output and the actual output.

The point of training is to make the cost of training as small as possible across millions of training examples.To do this, the network tweaks the weights and biases until the prediction matches the correct output.

Once trained well, a neural net has the potential to make an accurate prediction every time.

When the pattern getscomplex and you want your computer to recognise them, you have to go for neural networks.In such complex pattern scenarios, neural network outperformsall other competing algorithms.

There are now GPUs that can train them faster than ever before. Deep neural networks are already revolutionizing the field of AI

Computers have proved to be good at performing repetitive calculations and following detailed instructions but have been not so good at recognising complex patterns.

If there is the problem of recognition of simple patterns, a support vector machine (svm) or a logistic regression classifier can do the job well, but as the complexity of patternincreases, there is no way but to go for deep neural networks.

Therefore, for complex patterns like a human face, shallow neural networks fail and have no alternative but to go for deep neural networks with more layers. The deep nets are able to do their job by breaking down the complex patterns into simpler ones. For example, human face; adeep net would use edges to detect parts like lips, nose, eyes, ears and so on and then re-combine these together to form a human face

The accuracy of correct prediction has become so accurate that recently at a Google Pattern Recognition Challenge, a deep net beat a human.

This idea of a web of layered perceptrons has been around for some time; in this area, deep nets mimic the human brain. But one downside to this is that they take long time to train, a hardware constraint

However recent high performance GPUs have been able to train such deep nets under a week; while fast cpus could have taken weeks or perhaps months to do the same.