

[Few Hours To Go] Top Companies are Hiring 300+ WOMEN Data Scientists! Are you on their list?

[Register Now](#)

00 D 11 H 14 M 39 S

[Home](#)

# Absolute Beginner's Series To Implement Neural Networks(Part-1)

 Adwait Dathan R – July 22, 2021[Advanced](#) [Deep Learning](#) [Maths](#)

This article was published as a part of the [Data Science Blogathon](#)

**JOB-A-THON**

India's Largest Data Science Hiring Event

Exclusively for Women

[Hiring Companies](#)**Deloitte**, **fractaloo** **TIGER ANALYTICS** and many more...

8-13 March 2022

[Register Now](#)

## Overview

This is the first of the three articles in the series covered, explaining some of the basic theories that you have to know while implementing the neural network. In the upcoming blogs, you will get to know how to apply the learned theory to code your custom neural network model. Maximum effort has been made to make this article interactive and simple so that everyone can understand. I hope you will enjoy this. Happy learning!!



## Introduction

[Image Source](#)

Neural networks, that were actually inspired by the human brain's neurons are a buzzword these days because of their ability to do an amazing amount of work in various domains.

Here I am not going to dive into the in-depth theory of neural networks, but for understanding the code some basic concepts will be explained. For a deeper understanding, this article won't be just enough, you have to go through some standard materials that are present. Saying so, let me concisely explain some concepts.

Some Basic Concepts Related To Neural networks

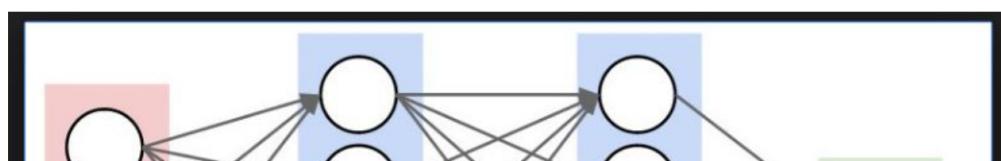
Conclusion

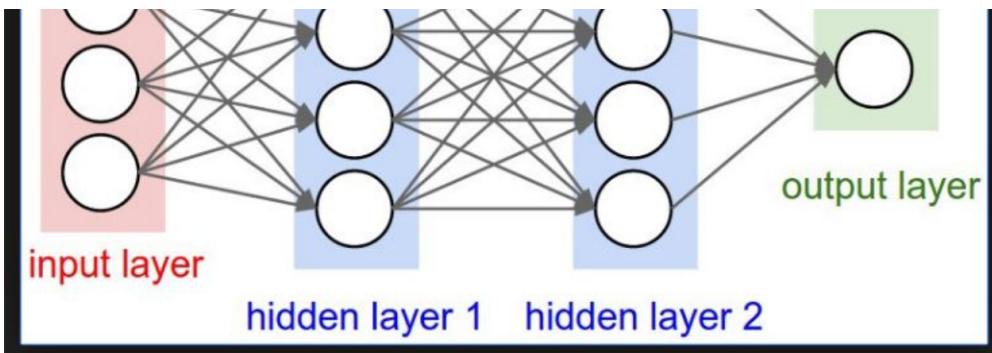
References

About the Author

## Some Basic Concepts Related To Neural networks

### 1.Different Layers of a Neural Network



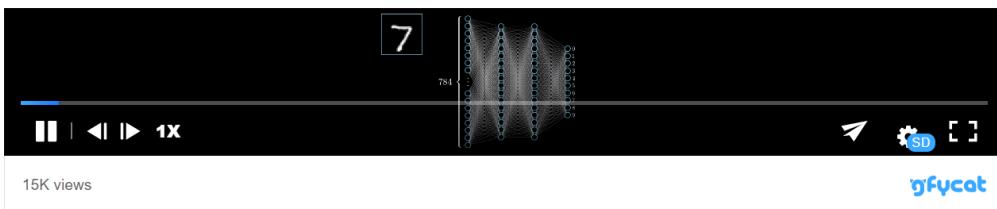


*Image Source*

1. **Input Layer** – This is the gateway for the inputs that we feed.
2. **Hidden layers** – The layer where complex computations happen. The more your model has hidden layers, the more complex the model will be. This is kind of like a black box of the neural network where the model learns complex relations present in the data.
3. **Output Layer** – From the diagram given above there is only one node in the output layer, but don't think that it is always like that in every neural network model. The number of nodes in the output layer basically depends upon the problem that we have taken. If it's a classification model, the number of nodes in the output layer will be equal to the number of classes that we want to predict.

Now you might have got a better idea. If not please check out this [video](#). Now we will quickly see some of the concepts that are necessary to build and train a neural network.

## 2. Forward propagation

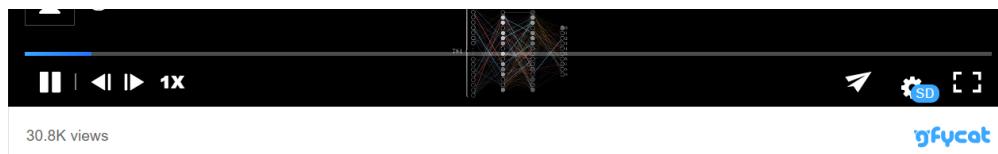


*Image Source*

Keenly observe the above animation. This is a number classification problem and our input is the digit seven. During the forward propagation, we are pushing the input to the output layer through the hidden layers. There is also another interesting observation: only some neurons in the hidden layer get fired during the entire process. Finally, we can observe that the model gave a higher probability/prediction that the input is number seven. But if the model is not well trained there is a good chance that the model may give the wrong prediction bringing in error.

## 3. Backpropagation





[Image Source](#)

The backpropagation algorithm has mainly two steps :

1. A forward propagation step, where the input is pushed through the neural network to get the output.
2. During the backward pass we calculate the derivative of the loss function at the output layer. Then we are applying the chain rule, to calculate the gradient of the weights present in the other layers. The gradients calculated are then used for the process of parameter update.

From the video above, during the first few seconds, you are seeing the forward propagation as explained before but after that, you can see that something is getting propagated in a backward direction from the output layer which is actually the error that the model has made while predicting the output. To understand the mathematics related to backpropagation please check [here](#).

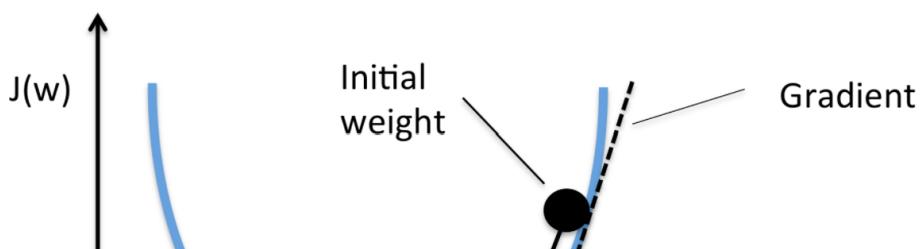
## 4. Updating weights and biases

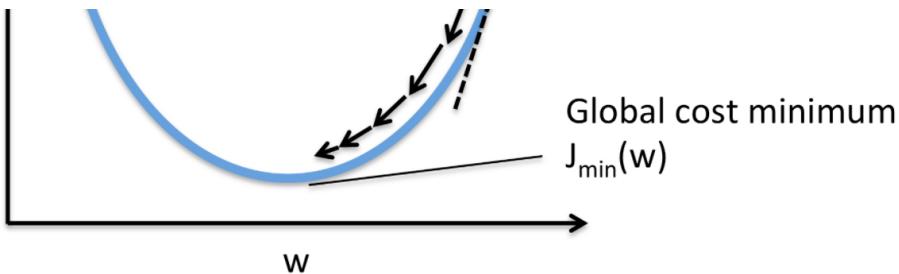
$$*W_x = W_x - a \left( \frac{\partial \text{Error}}{\partial W_x} \right)$$

↓                              ↓  
Old weight      Derivative of Error  
                        with respect to weight  
↑  
New weight      Learning rate

[Image Source](#)

This is one of the critical steps in the training phase wherein the neural network learns from its mistakes and adjusts its features to reduce the loss. If you don't know what weights and biases are please click [here](#) to get a better intuition. The neural network uses an optimization algorithm such as gradient descent to update its weights and finds a particular point at which the loss will be minimal.





[Image Source](#)

While coding the neural network, we don't have to worry about these complex updatons because there is a function called Autograd which will help us to calculate the gradients of the weights and biases.

## 5. Activation Functions

Activation functions play a key role in neural networks. They are being extensively used in the outputs of hidden layers and as well as the output layer.

Commonly Used Activation Functions	Range
1. Step function : $f(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases}$	① $\{0, 1\}$
2. Signum function: $f(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	② $\{-1, 1\}$
3. Linear function: $f(z) = z$	③ $(-\infty, \infty)$
4. ReLU function : $f(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$	④ $(0, \infty)$
5. Sigmoid function: $f(z) = \frac{e^z}{1+e^z}$	⑤ $(0, 1)$
6. Hyperbolic tan : $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	⑥ $(-1, 1)$

by Dr. Pankaj Kumar Porwal (BTech - IIT Mumbai, PhD - Cornell University) : Principal, Techno India NJR Institute of Technology, Udaipur

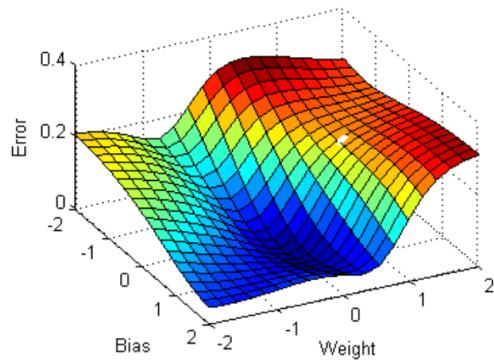
[Image Source](#)

The Rectified linear activation Function (Relu) has been commonly used in many neural network models, it has been a kind of default activation function. The softmax and sigmoid activation functions are being used mainly in the outputs of a model built for classification type problems. Sigmoid is used for computing the probability of classes for binary classification while softmax activation function has been used for calculating the probability for multi-class classification problems.

## Some simple optimization functions

In the previous sections, a small intuition was provided about the gradient descent algorithm. But the normal gradient descent algorithm has a big disadvantage because it requires a huge amount of memory to load the entire dataset for computing the gradients. So some variants were introduced in the gradient descent algorithm which produced better results.

## 1. Stochastic Gradient Descent

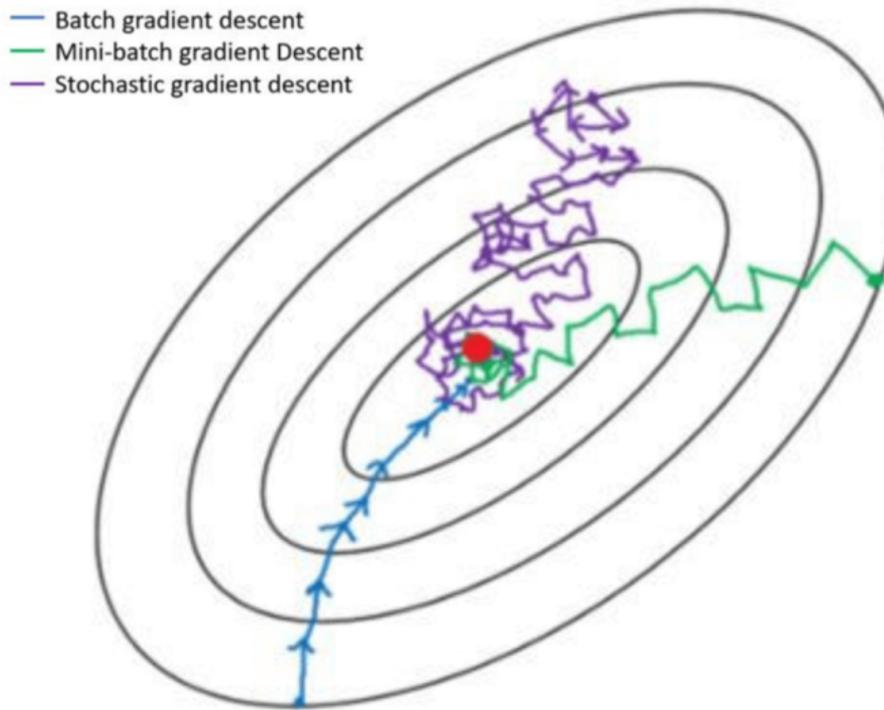


*Image Source*

Stochastic gradient descent(SGD), a variant of the ordinary gradient descent algorithm computes the derivative of the parameters by considering one individual/training sample at a time instead of loading the whole of the dataset. But the time complexity/ time taken for the SGD to converge to the global minima was relatively high compared to the normal gradient descent algorithm.

## 2. Mini Batch Stochastic Gradient Descent (MB-SGD)

Mini Batch SGD is a slight variant of the normal SGD. Here instead of taking a single sample at a time, a batch of samples are taken and the parameters are updated.



*Image Source*

Observing the above diagram we can see how each optimization algorithm approaches the global minima. We can see here that SGD is somewhat noisy while compared to batch gradient descent and mini-batch gradient descent.

## Conclusion

In this article, a concise explanation has been done for some of the basic concepts that are necessary while implementing a neural network. In the upcoming parts of this series, we will be covering how to implement a simple neural network with the help of the facts which we have discussed here. The missed concepts if any, will be explained in the next article. In the final article, we will be discussing transfer learning and solve problems using pre-trained state-of-the-art models.

## References

1. <https://unsplash.com/s/photos/network> – Article cover Image.
2. <https://mlfromscratch.com/optimizers-explained/> – Brief explanation on optimizers.
3. <https://www.pyimagesearch.com/2021/05/06/backpropagation-from-scratch-with-python/> – Backprop.

## About the Author

Hello, myself Adwait Dathan currently doing my master's in Artificial Intelligence and Data Science. Please feel free to connect with me through [Linkedin](#).

*The media shown in this article on Neural Networks are not owned by Analytics Vidhya and are used at the Author's discretion.*

---

### Related



[Introduction to Neural Networks](#)



[My playlist – Top YouTube Videos on Machine Learning, Neural Network & Deep Learning](#)



[Deep Learning Guide: Introduction to Implementing Neural Networks using TensorFlow in Python](#)

---

[blogathon](#) [neural networks](#)

---

### About the Author



Adwait Dathan R



## Our Top Authors



## Download

Analytics Vidhya App for the Latest blog/Article



Previous Post

Must-Know Text Operations in Python before you dive into NLP!

Next Post

Top 3 Ways to Write Your Tensorflow Code

## Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

Name\*

Email\*

Website

Notify me of follow-up comments by email.

Notify me of new posts by email.

**Submit**

# Top Resources



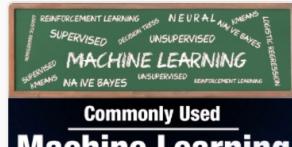
Python Tutorial:  
Working with CSV file  
for Data Science

👑 Harika Bonthu -  
AUG 21, 2021



3 Interesting Python  
Projects With Code  
for Beginners!

👉 Gaurav Sharma -  
JUL 18, 2021



Commonly Used  
**Machine Learning**

ALGORITHMS Analytics Vidhya

Commonly used  
Machine Learning  
Algorithms (with  
Python and R Codes)

sunil - SEP 09, 2017



6 Easy Steps to Learn  
Naive Bayes  
Algorithm with codes..

sunil - SEP 11, 2017



Download App



Analytics Vidhya

About Us

Our Team

Careers

Contact us

Data Scientists

Blog

Hackathon

Discussions

Apply Jobs

Companies

Post Jobs

Trainings

Hiring Hackathons

Visit us

