DEEP LEARNING (P-1)

techworldthink • March 08, 2022

1. Describe sigmoid activation functions

An activation function is a very important feature of an artificial neural network , they basically decide whether the neuron should be activated or not.

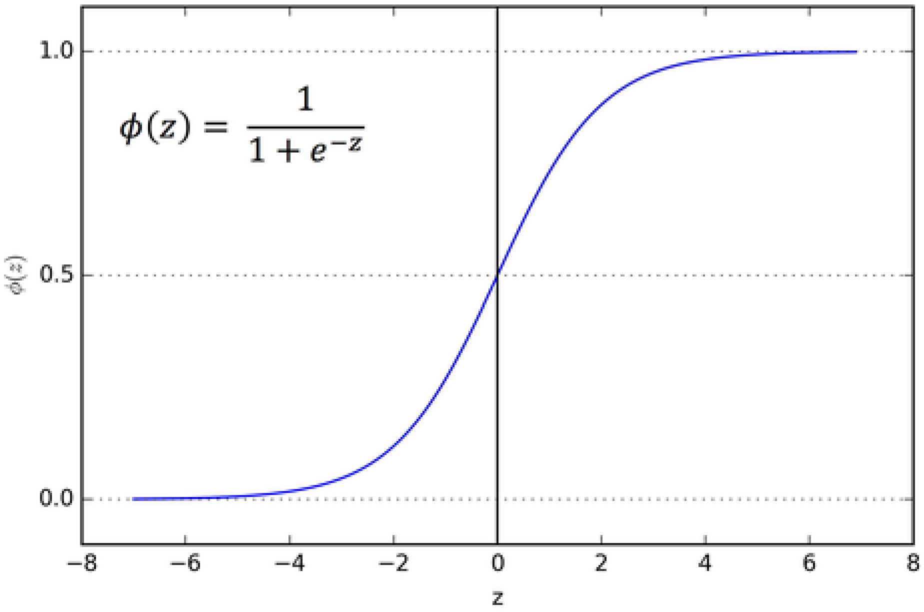
The Activation Functions can be basically divided into 2 types-

1. Linear or Identity Activation Function
2. Non-linear Activation Functions

The Nonlinear Activation Functions are the most used activation functions.It makes it easy for the model to generalize or adapt with variety of data and to differentiate between the output.

The Nonlinear Activation Functions are mainly divided on the basis of their range or curves-

The Sigmoid Function curve looks like a S-shape.



Equation : f(x) = 1 / 1 + exp(-x)

Range : (0 to 1)

Pros:

1.The function is differentiable.That means, we can find the slope of the sigmoid curve at any two points

2.The function is monotonic but function's derivative is not

Cons:

1.It gives rise to a problem of "vanishing gradients", since the Y values tend to respond very less to changes in X

2.Secondly , its output isn't zero centered. It makes the gradient updates go too far in different directions. 0 < output < 1, and it makes optimization harder.

3. Sigmoids saturate and kill gradients.

4. Sigmoids have slow convergence.

2. Write the gradient descent algorithm.

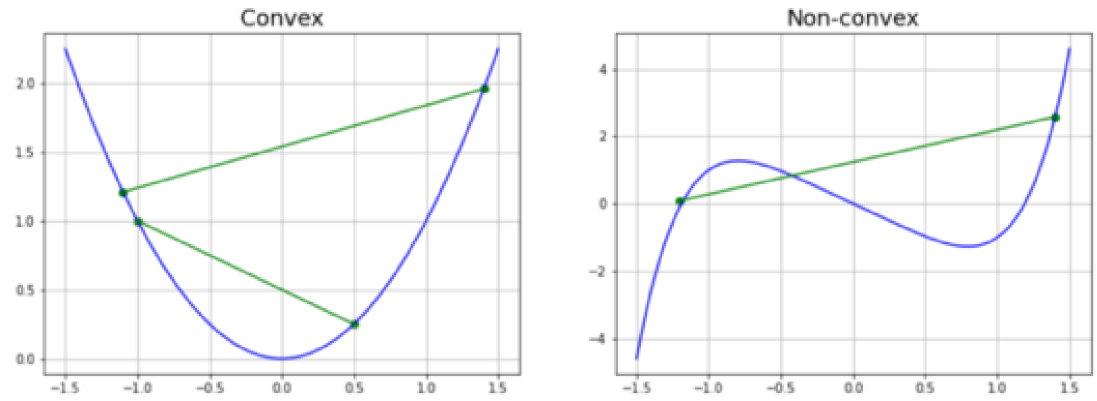
Gradient descent (GD) is an iterative first-order optimisation algorithm used to find a local minimum/maximum of a given function. This method is commonly used in *machine learning* (ML) and *deep learning*(DL) to minimise a cost/loss function (e.g. in a linear regression).

Gradient descent algorithm does not work for all functions. There are two specific requirements. A function has to be:

- differentiable
- convex

If a function is differentiable it has a derivative for each point in its domain — not all functions meet these criteria.

function has to be convex. For a univariate function, this means that the line segment connecting two function's points lays on or above its curve (it does not cross it). If it does it means that it has a local minimum which is not a global one.



To find the local minimum of a function using Gradient Descent, take steps proportional to the negative of the gradient(move away from the gradient) of the function at the current point.

It takes steps proportional to the positive of the gradient (moving towards the gradient), we will approach a local maximum of the function. This procedure is called gradient ascent.

Gradient Descent Algorithm iteratively calculates the next point using gradient at the current position, then scales it (by a learning rate) and subtracts obtained value from the current position (makes a step). It subtracts the value because we want to minimise the function (to maximise it would be adding). This process can be written as:

$$p_{n+1} = p_n - \eta \nabla f(p_n)$$

There's an important parameter η which scales the gradient and thus controls the step size. In machine learning, it is called learning rate and have a strong influence on performance.

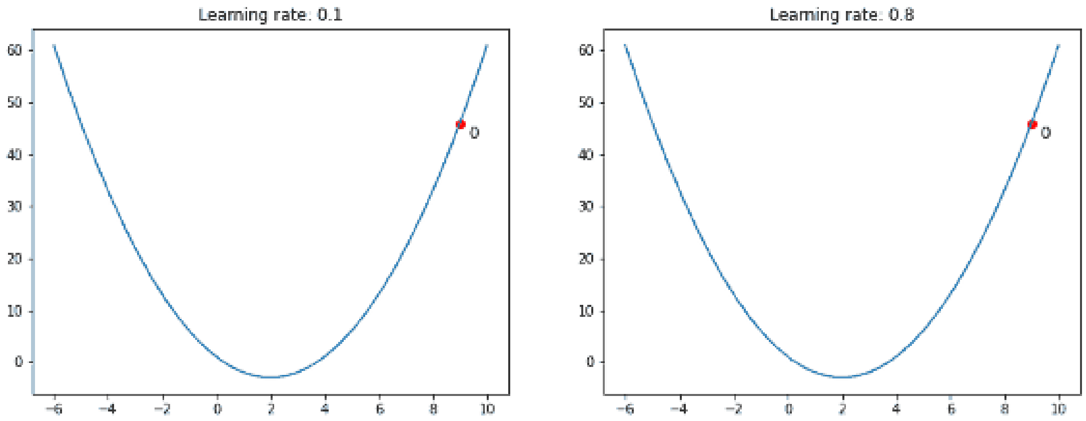
- The smaller learning rate the longer GD converges, or may reach maximum iteration before reaching the optimum point
- If learning rate is too big the algorithm may not converge to the optimal point (jump around) or even to diverge completely.

- Choose an optimal learning rate so that model converges to the minimum.

In summary, Gradient Descent method’s steps are:

1. choose a starting point (initialisation)
2. calculate gradient at this point
3. make a scaled step in the opposite direction to the gradient (objective: minimise)
4. repeat points 2 and 3 until one of the criteria is met:

- maximum number of iterations reached
- step size is smaller than the tolerance.



https://telegra.ph/DEEP-LEARNING-P-1-03-08 (original version)

https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a (types)

3. Explain with an example how graphs are stored and represented in TensorFlow.

TensorFlow is a free and open-source software library for machine learning and artificial intelligence.

It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

TensorFlow was developed by the Google Brain team for internal Google use in research and production.

TensorFlow is Google Brain's second-generation system.

It can be used in a wide variety of programming languages, most notably Python, as well as Javascript, C++, and Java.

TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

TensorFlow serves as the core platform and library for machine learning. TensorFlow’s APIs use Keras to allow users to make their own machine learning models.

TensorFlow makes use of a graph framework. The graph gathers and computations done during the training. The graph has lots of advantages:

•It was done to run on multiple CPUs or GPUs and even mobile operating

•The portability of the graph allows to preserve the computations for graph can be saved to be executed in the future.

•All the computations in the graph are done by connecting tensors together

• A tensor has a node and an edge. The node carries the mathematical endpoints outputs. The edges explain the input/output

> *In order to be a highly efficient, flexible, and production-ready library, TensorFlow uses dataflow graphs to represent computation in terms of the relationships between individual operations. Dataflow is a programming model widely used in parallel computing and, in a dataflow graph, the nodes represent units of computation while the edges represent the data consumed or produced by a computation unit.*