Deep Learning Frameworks – Tensorflow and Pytorch

***Part I: Tensorflow***

1. Tensorflow basics –
2. In order to run a Tensorflow program one should define a graph.

Tensorflow is based on the idea that numerical computations are done by computational graph. Each computation in the sequence has input and output and connected to the next computation, which creates a directed graph. The purpose of graphs is to represent the data flow of all the computations, when each one is represented by a node in the graph.

1. Tensorflow has introduced a feature that enables the users to execute a program without compiling a graph. This is called Tensorflow XLA JIT – Accelerated Linear Algebra Just-in-time. This compilation method will optimize a graph so that computations will result in a single GPU kernel, and instead of allocating memory for mediate computations of the graph, it streams these intermediate results into the next node, while saving them in the GPU registers. Removing memory operations will improve the performance of the algorithm.

Also, Tensorflow has ‘eager execution’ which is programming environment that evaluates operation immediately, without build the computational graph – which is easy to debug.

1. Tensor graphs that contain dependent operations, does not automatically ensure the correct order of the operations execution.