COMPSCI 590S Systems for Data Science **Fall 2016**

Lecture 26: Tensor Flow

Lecturer: Emery Berger *Scribe(s): Lurdh Pradeep Reddy Ambati, Nabanita De*

26.1 Neural Networks

Neural Network is a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.

- During training phase (using Neural networks), loss function is computed(back propagation) and it is used for readjusting the weights to achieve higher accuracy.
- In Neural Networks, **ReLU**(Rectifying linear unit) are used to prevent the negative values.
- Sigmoid function is applied to the loss-function as **0-1** loss is not differential.

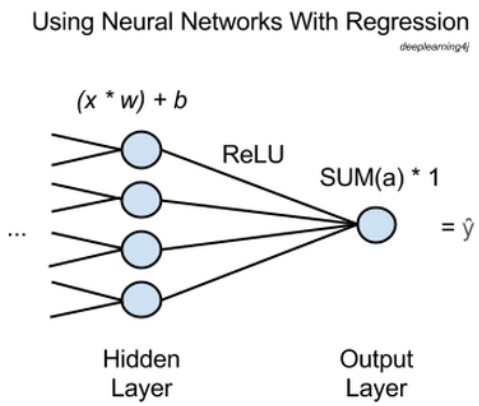

Figure 26.1: Neural Network

26.2 Deep Learning

Deep learning refers to artificial neural networks that are composed of many layers. Earlier 40 years back, neural networks were only 2 layers deep as it was not computationally feasible to build larger networks. But, now with better computing parts like Faster CPU, Faster GPU, large storage, now large neural networks are feasible.

And to accelerate the computing of these multi-layer neural networks, **ASIC**'s(Application specific Integrated Circuits) have been developed like **TPU**(Tensor Processing Units), which are designed to compute vector multiplications and additions.(AI accelerators)

26.3 Parameter Server Architecture

In Parameter server architecture, a job comprises two disjoint sets of processes- stateless worker processes and stateful parameter server. In this model, workers perform the bulk of computations and fetch the data from the parameter server and computed data is sent back to the parameter.

26.4 DistBelief

DistBelief was predecessor to TensorFlow, was implemented in C++. DistBelief uses Parameter Server Architecture.

- DistBelief was not flexible - It was hard to implement new algorithms in DistBelief like Softmax classifier, RandomForest algorithms etc.
- DistBelief doesn't scale down - it is developed to be deployed in a cluster and cannot be used on a single pc.

26.5 TensorFlow

TensorFlow is a machine learning system that operates at large scale and in hetrogenous environments.

- It uses data flow graphs(dfg) to represent computation shared state and the operations as well.
- Data is represented in the form of **Tensors**- multi dimensional arrays.
- Nodes are stateful which do bulk of computations.
- Defers the execution until the entire program is available so that TensorFlow can optimize the execution phase by using global information about the computation.
- Constant folding and common sub expression elimination are few of the optimizations that TensorFlow can perform.
- TensorFlow represents the data/matrices in the form of the dense matrices only.(No support for sparse matrices)

26.5.1 Dynamic Control Flow - Switch and Merge

TensorFlow implements *Switch* and *Merge* primitives, which are borrowed from classic dynamic data flow architectures.

- Switch : is a demultiplexer. It takes a input data and a control input, and uses the control input to route the input data to one of its two outputs. The Switch utput not taken receives a special dead value, which propagates through the rest of graph until it reaches a Merge Operation.
- Merge : is multiplexer. It forwards atmost one non-dead input to its output or if two inputs are dead values, then it produces a dead output.

26.5.2 Operators - Variadic and Polymorphic

- Polymorphic : Polymorphic mean that the operators in TensorFlow can operate on multiple types of data. For example, a single Add operator can add Ints, Floats or Vectors of values.(Like Java generic functions)
- Variadic : Variadic mean that the operators in TensorFlow can operate on different number of arguments. For example, a single Add operator can add 2 variables or 3 variables etc.

26.5.3 Stateful operations : Queue

Queues(stores Tensors) allows concurrent access to tensors in the respective orders. These queue can block if the queue is full, thus providing backpressure and also supports synchronization ensuring deterministic results.