

# Excursion to Artificial Neural Network

Himanshu Rawlani

Machine Learning and Deep Learning Engineer

February 10, 2019





# Agenda

1. Neural Network Playground Demo
2. Building your first neural network in Python using Numpy
3. Introduction to Deep learning frameworks
4. Build and train CNN in using Keras



# TensorFlow Neural Network Playground

Let's visually understand the following

1. Representation of neural networks
2. Forward propagation
3. Backward propagation
4. Overall training dynamics



# Building your first neural network using Numpy

Steps involved:

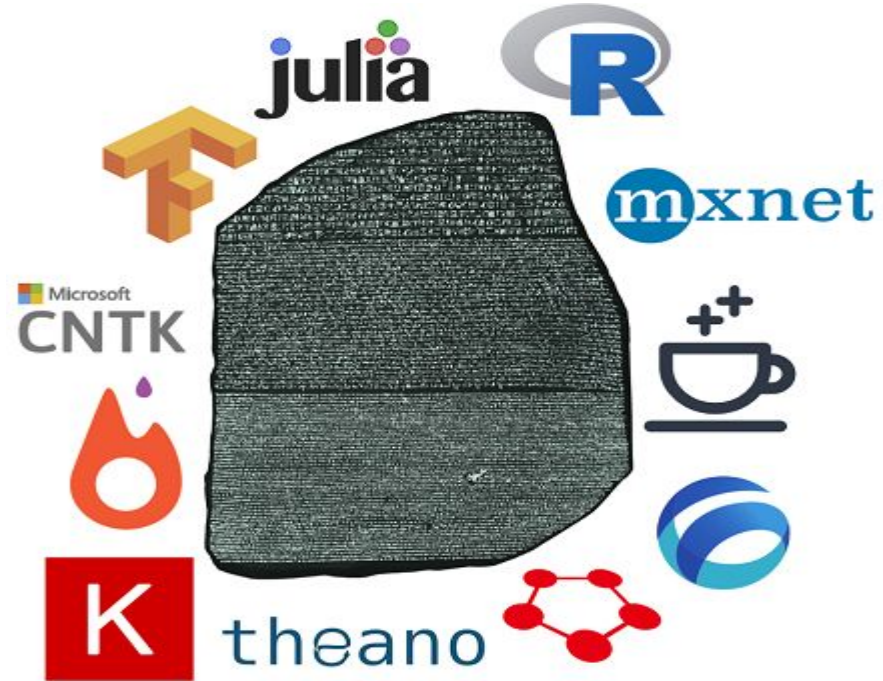
1. Importing the required libraries
2. Getting the training dataset
3. Defining helper functions
4. Define the training loop
5. Perform any input data preprocessing and pass it to the train function
6. Do the predictions and evaluation of trained model



**Any doubts?**

# Introduction to Deep learning frameworks

1. Keras
2. TensorFlow
3. PyTorch
4. Sonnet
5. MXNet
6. CNTK
7. DL4J





# 1. TensorFlow (Google)

1. TensorBoard
2. TensorFlow serving
3. TensorFlow Lite
4. TensorFlow Debugger
5. Default mode: Static computation graphs
6. For dynamic architectures TensorFlow has Eager Execution mode (Dynamic computation graphs)



## 2. PyTorch (Facebook)

1. Default mode: Dynamic computation graphs a.k.a. Define by run mode
2. Can be easily used by commons debugging tools like pdb and PyCharm
3. Modular and numpy like code
4. Lacks native serving and visualizations tools
5. Best for prototyping and small scale projects





### 3. Sonnet (Deepmind)

1. Built on top of TensorFlow
2. First constructs python object which represents some part of neural network, then separately connects these objects into the computational graph
3. These modules simplify training process and can be combined to implement higher level networks
4. Can be easily extended for custom implementation



## 4. MXNet (Amazon)

1. Adopted by AWS
2. Huge variety of APIs
3. Ability to scale linearly (i.e. on scaling on 128 GPUs the algorithm performs more than 100 times better than on a single GPU)
4. High performance imperative API
5. Simplicity of Keras and dynamic like PyTorch



## 5. CNTK (Microsoft Cognitive Toolkit)

1. Used in Skype, Xbox and Cortana
2. Support for Apache spark
3. Easiest to integrate with Azure
4. Handles passing of sequences better than other frameworks (TensorFlow requires padding and masking, etc.)
5. Community mostly consists of windows laptop and mobile developers (Xamarin)
6. No conventional open source license



## 6. DL4J (Deep learning 4 Java)

1. Mainly Java and Scala
2. Works with Apache, Hadoop and Spark
3. Hard to integrate it with other ML libraries
4. Good for Android apps if written in Java
5. Possible to write Full Stack DL pipeline in Java
6. No conventional open source license



## 7. ONNX (Microsoft and Facebook)

1. ONNX - Open Neural Network Exchange
2. Enables tools by allowing to share models between frameworks
3. Train with one tool stack and export it in other (required to export model to ONNX format )
4. ONNX format is a serialized representation of model in ProtoBuf file



# Summary of Deep learning frameworks

1. For beginners - Use Keras
2. For production on GCP - Use TensorFlow
3. For research - Use PyTorch
4. For research - Also try Sonnet
5. For production on AWS - Use MXNet
6. For production on Azure - Use CNTK
7. For Java Developers - Use DL4J



# Building your first neural network using Keras

1. We will train a simple deep CNN on CIFAR10 small image dataset
2. CIFAR10 consists of following 10 classes: 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck'

The image features a solid orange background. In the top-left corner, there are three vertical bars of varying heights, each composed of three overlapping circles. In the bottom-right corner, there are four vertical bars of varying heights, each composed of four overlapping circles. The text "Gotta clear 'Em all!" is centered in the middle of the image in a white, bold, sans-serif font.

**Gotta clear 'Em all!**



# Thank You!

Himanshu Rawlani

Twitter: [https://twitter.com/raw\\_himanshu](https://twitter.com/raw_himanshu)

LinkedIn: <https://www.linkedin.com/in/himanshurawlani/>

GtiHub: <https://github.com/himanshurawlani/>

