

---

# Getting started with TensorFlow (TF)

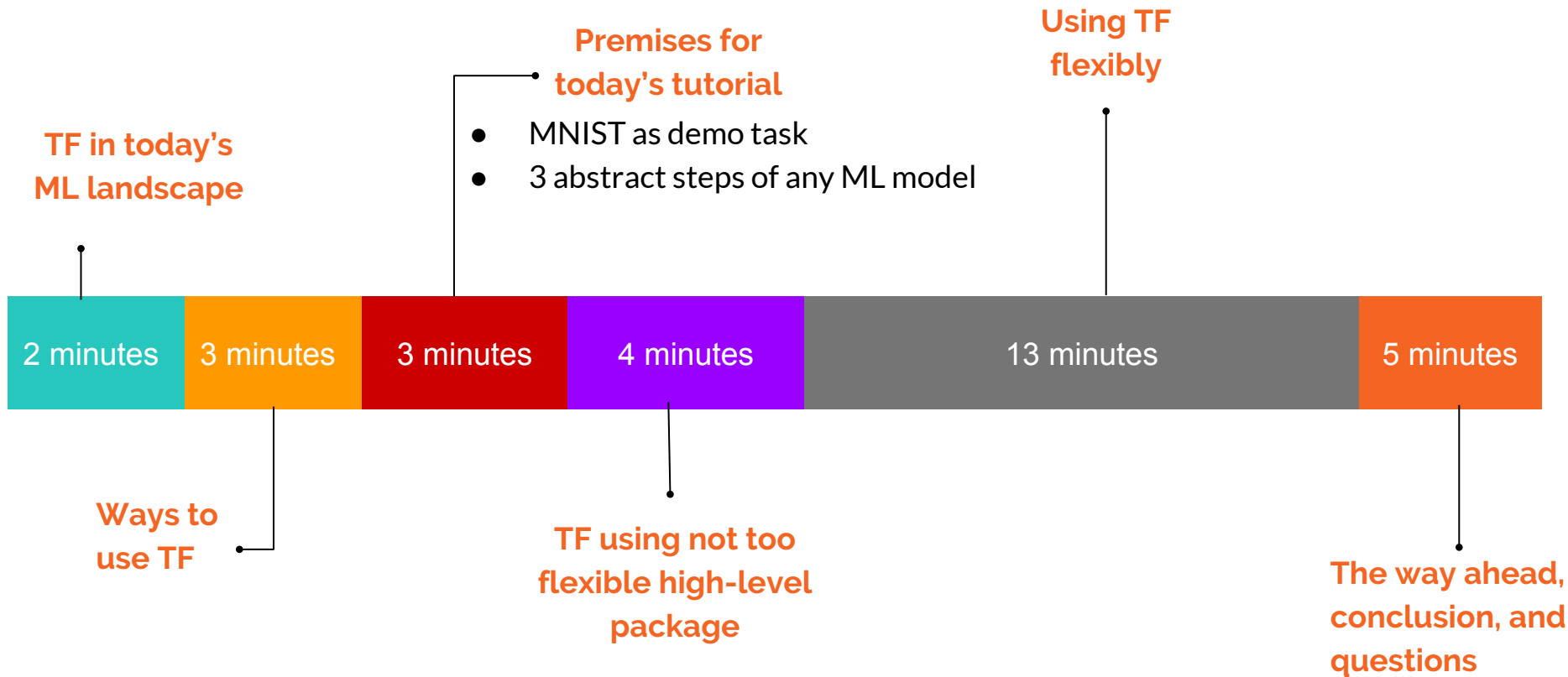


Target audience - Beginners in TF

- Sanjay Thakur

---

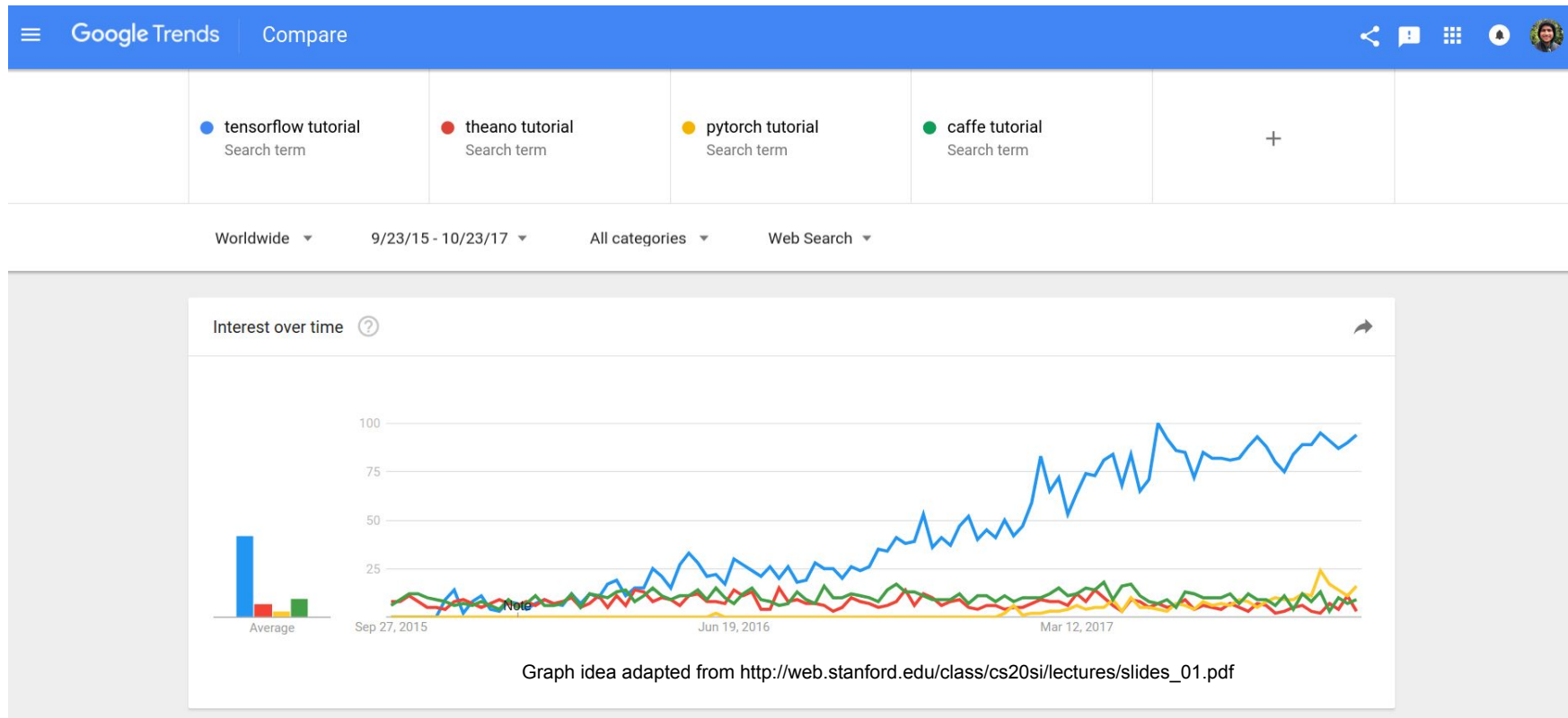
# Plan for today



# TensorFlow(TF) in today's ML landscape

- TensorFlow (TF) is an open-source numerical computation python library maintained by Google Inc.
- Supports efficient ways to play with Machine Learning ideas, especially that of Deep Learning.
- Has awesome benefits of using it, like
  - TensorBoard
  - Checkpoints
  - Huge and helpful community support

# TensorFlow in today's ML landscape(2)



This graph is testimony to TensorFlow's popularity.

# Ways to use TensorFlow

There are at-least six direct or indirect ways that I know of to use TensorFlow

- As a Keras Backend.
- Using Keras wrapper to call TF functions as its high-level library.
- Using `tensorflow.contrib.keras`
- Using TFLearn as a high-level TF library.
- By using TF's high-level functions.
- By manually defining TF's computation graph and setting a session to run it.

# Ways to use TensorFlow

Easy, not flexible

There are at-least six direct or indirect ways that I know of to use TensorFlow

- As a Keras Backend.
- Using Keras wrapper to call TF functions as its high-level library.
- Using `tensorflow.contrib.keras`
- Using TFLearn as a high-level TF library.
- By using TF's high-level functions.
- By manually defining TF's computation graph and setting a session to run it.

Flexible, but difficult

# Ways to use TensorFlow

Easy, not flexible

There are at-least six direct or indirect ways that I know of to use TensorFlow

- As a Keras Backend.
- Using Keras wrapper to call TF functions as its high-level library.
- Using tensorflow.contrib.keras
- Using TFLearn as a high-level TF library.
- By using TF's high-level functions.
- By manually defining TF's computation graph and setting a session to run it.

Covered in Tutorial 2 already

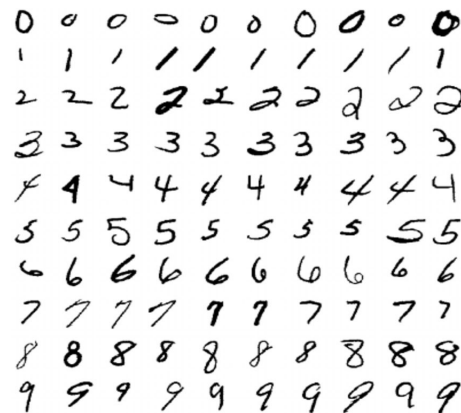
We won't cover these. Their links are given at the end of these slides

This is what we'll be talking about for next the remainder of my presentation.

Flexible, but difficult

# The premise for this tutorial

- **Demonstration task:** MNIST handwritten digit recognition task





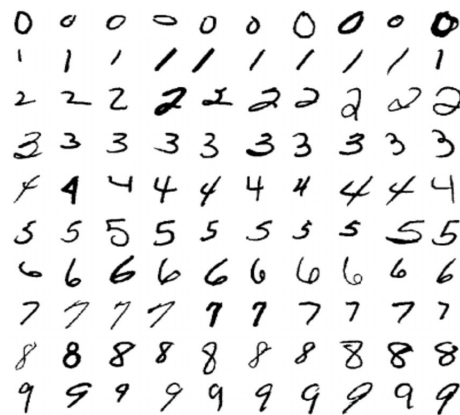
# The premise for this tutorial

- **Demonstration task:** MNIST handwritten digit recognition task
- **Goal:** To identify digits from handwritten digits.



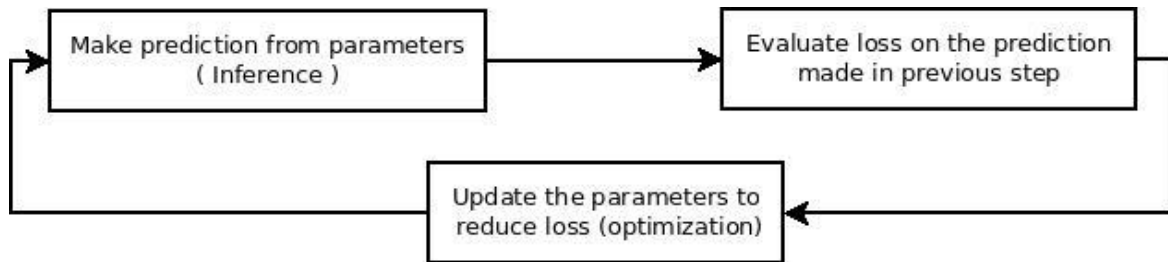
# The premise for this tutorial

- **Demonstration task:** MNIST handwritten digit recognition task
- **Goal:** To identify digits from handwritten digits.
- It is a classic dataset for benchmarking different algorithms mainly for computer vision task.



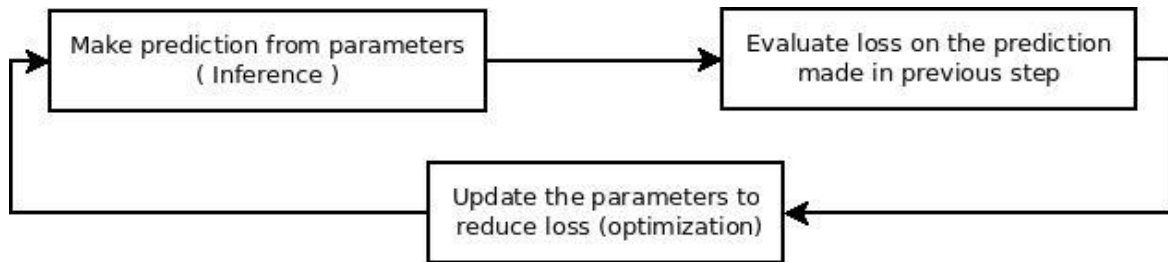
# The premise for this tutorial(2)

Remember these three steps of creating any Machine Learning model from this moment until your last moment on earth.



# The premise for this tutorial(2)

Remember these three steps of creating any Machine Learning model from this moment until your last moment on earth.

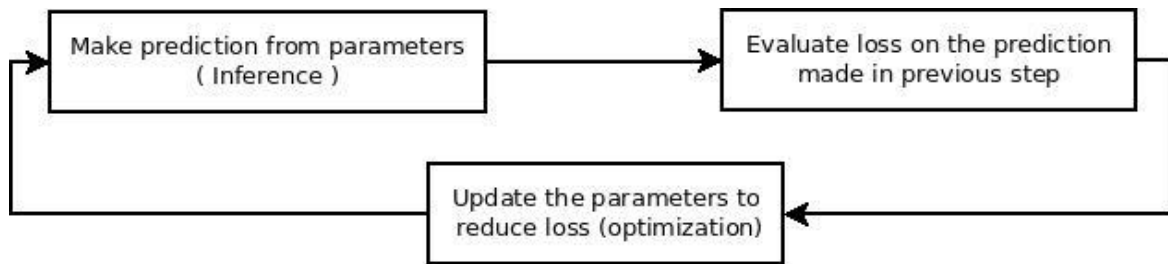


Until **SUFFICIENTLY LOW LOSS** is obtained, do:

1. Make **INFERENCE**

# The premise for this tutorial(2)

Remember these three steps of creating any Machine Learning model from this moment until your last moment on earth.

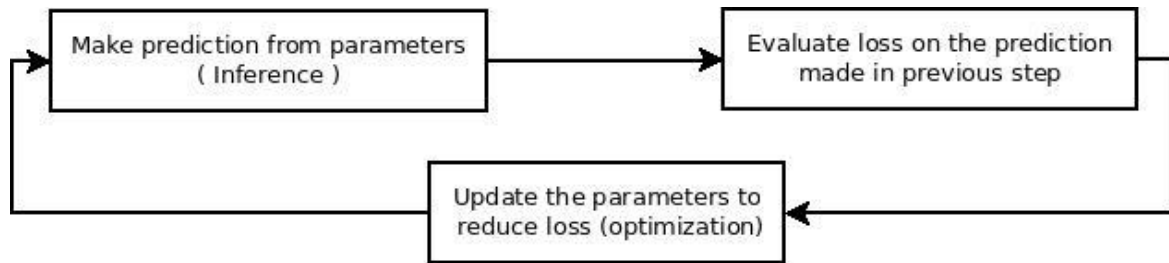


Until **SUFFICIENTLY LOW LOSS** is obtained, do:

1. Make **INFERENCE**
2. Determine **LOSS**

# The premise for this tutorial(2)

Remember these three steps of creating any Machine Learning model from this moment until your last moment on earth.



Until **SUFFICIENTLY LOW LOSS** is obtained, do:

1. Make **INFERENCE**
2. Determine **LOSS**
3. **UPDATE** the **PARAMETERS** wrt loss to make better inference in next iteration

NOTE:: By the term **PARAMETERS**, I mean weights and biases of a Deep Learning architecture

# TensorFlow using not too flexible high-level library

- We'll use ***tf.contrib.learn.DNNClassifier*** which is one of TF's many high-level classification packages.
- Let's switch to the jupyter notebook - **TF using high-level APIs.ipynb** in the github repository
- or, just click [here](#).

—

It turns out that this is **not as flexible as what hard-core ML enthusiasts would like it to be**. So let's dive deeper!!



# Using TF flexibly

**TF works in two phases::**

1. Creating a highly efficient computational graph, and
2. Running a session to compute the value of nodes of this graph.

## Using TF flexibly (2)

This feature of separate graph definition and computation is what makes TF so popular but there are both pros and cons of separating these two steps completely. We won't talk about it as it is outside the scope of this tutorial.

## Using TF flexibly (3)

### What is a Tensor

This is what TF transforms and see every data as. All TF computations occur on Tensors, and all data flow happens as Tensors. It is the soul of TF.

## Using TF flexibly (4)

- **Types of nodes in TF's computational graph::**  
Constants, Placeholders, Variables
- **Defining a session to run computations and fetch values from the TF nodes.**
- Switch to the jupyter notebook - **Finer Controls in TF - 1.ipynb** in the github repository to see usage.
- or, just click [here](#).

## Using TF flexibly (5)

**Keep in mind of most common pitfalls:**

1. Print a node directly to get its value.
2. Not initializing the variables explicitly before trying to run computations on top of them.

## Using TF flexibly (6)

- Switch to the jupyter notebook- **Finer Controls in TF - Demo.ipynb** in the github repository to see usage.
- or, just click [here](#).

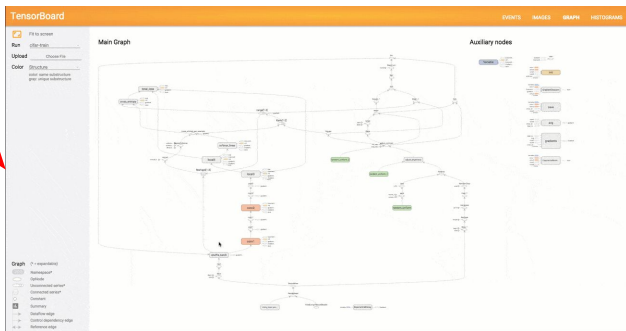
---

# Phew!! That's it.

We were limited by time-constraints, but before we end, here are a few pointers to move ahead.

# Other useful things to know about TF

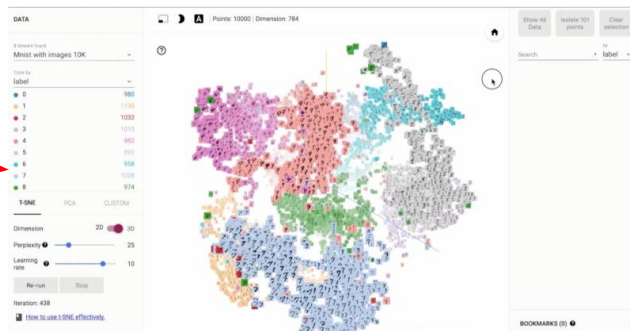
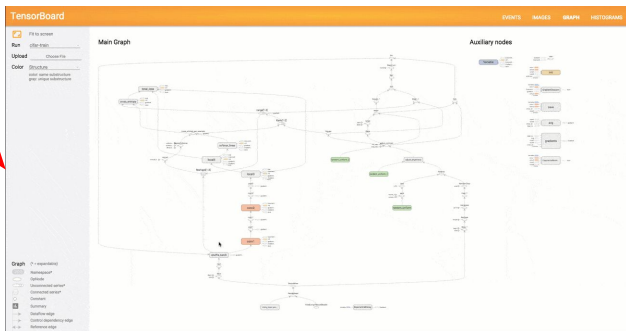
- Using TensorBoard effectively
  - Graph definition





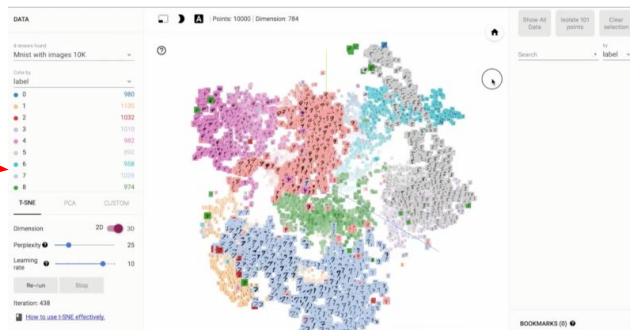
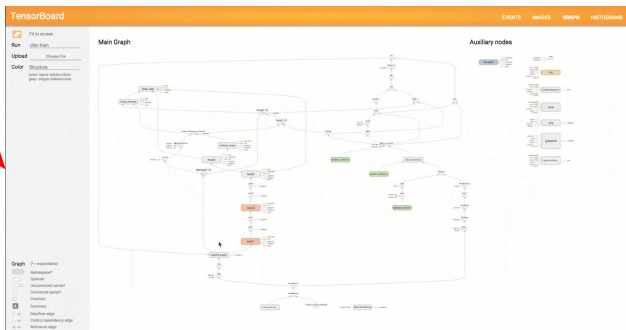
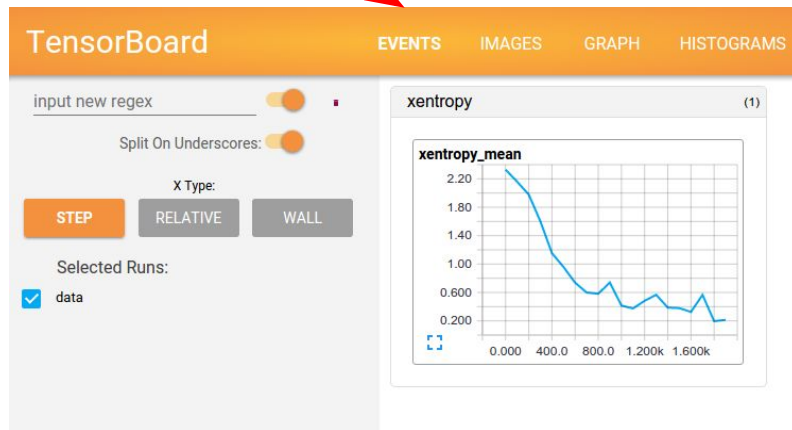
# Other useful things to know about TF

- Using TensorBoard effectively
  - Graph definition
  - 3-D visualization



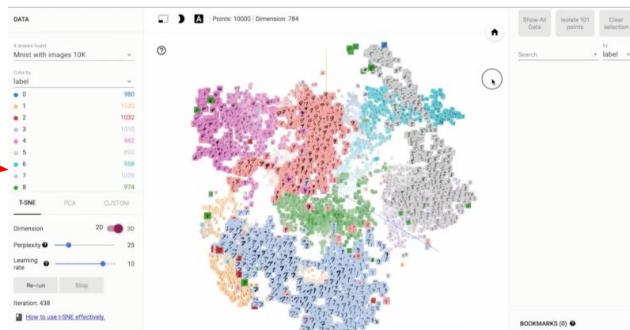
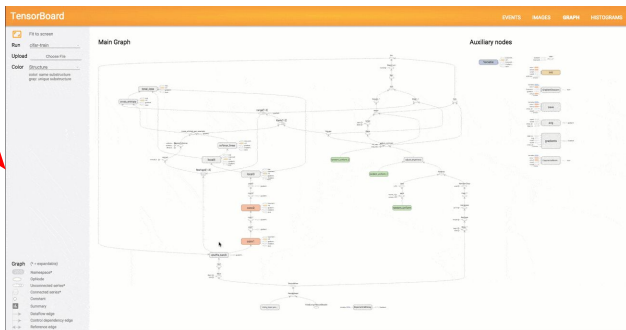
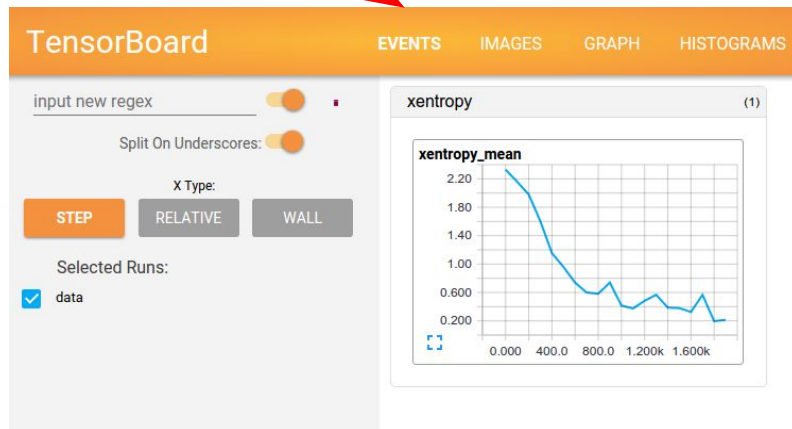
# Other useful things to know about TF

- Using TensorBoard effectively
  - Graph definition
  - 3-D visualization
  - Training progress on the fly



# Other useful things to know about TF

- Using TensorBoard effectively
  - Graph definition
  - 3-D visualization
  - Training progress on the fly
- Saving network parameters for later reuse



## Other relevant/suggested resources


- A comprehensive and easy to follow course on TF.  
<http://web.stanford.edu/class/cs20si/syllabus.html>
- How to use TF in an object-oriented fashion.  
<https://danijar.com/structuring-your-tensorflow-models/>
- TFLearn  
<http://tflearn.org/>
- Keras to call TF functions without using it as its backend.  
<https://blog.keras.io/keras-as-a-simplified-interface-to-tensorflow-tutorial.html>
- Keras as official part of TF  
[https://www.tensorflow.org/api\\_docs/python/tf/contrib/keras](https://www.tensorflow.org/api_docs/python/tf/contrib/keras)  
<https://news.ycombinator.com/item?id=13413487>
- My favorite place to look up the basics of deep learning and neural network.  
<http://neuralnetworksanddeeplearning.com/>



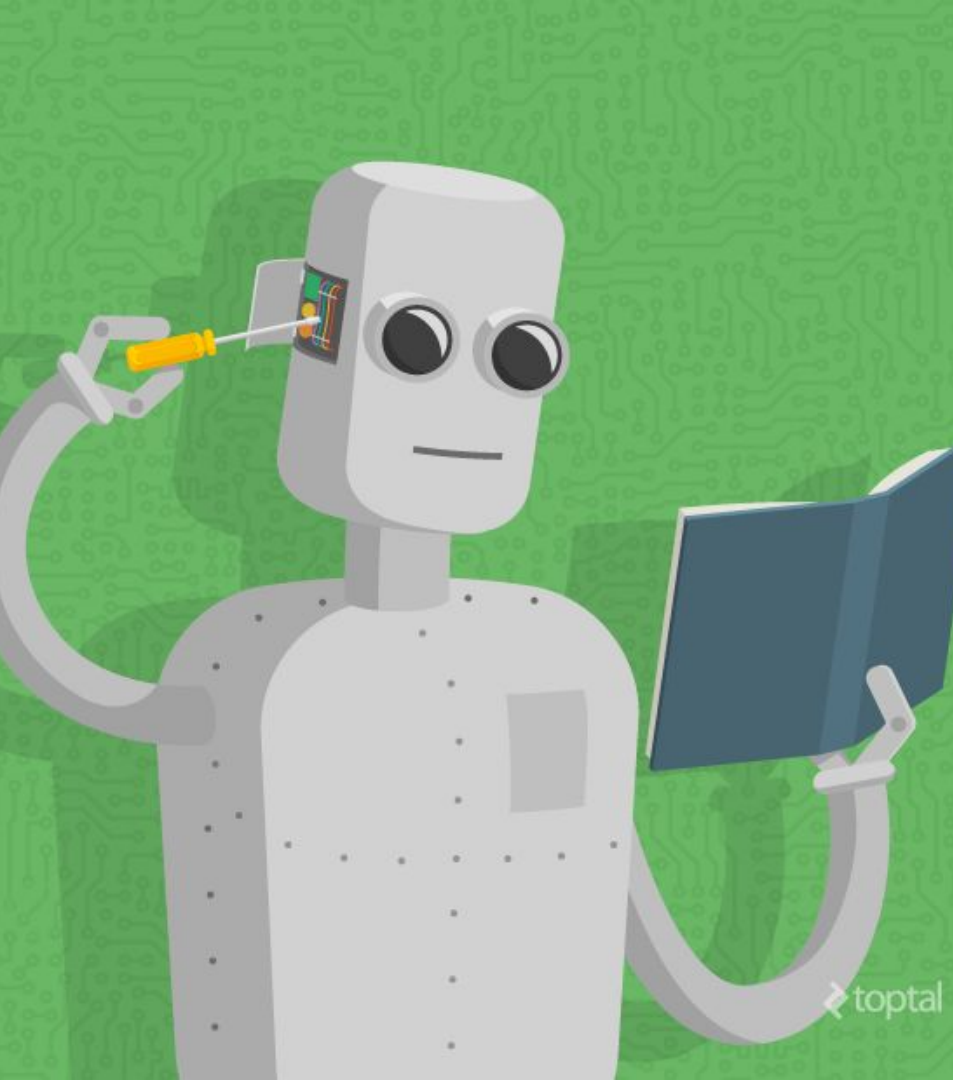
# That's it!!

- Slides and codes are available here:  
[https://github.com/sanjaythakur/TensorFlow\\_Tutorial](https://github.com/sanjaythakur/TensorFlow_Tutorial)
- Everything covered here has been pencilled down here in these notes as well.  
<https://sanjaykthakur.wordpress.com/2017/06/17/interfacing-tensorflow-to-noobs/>

# Questions, concerns, suggestions, or just want to say something

- Find me in MC105. I might look like something like this   
(without the puppy of course)
- or, shoot me an email at [sanjay.thakur@mail.mcgill.ca](mailto:sanjay.thakur@mail.mcgill.ca)
- or, if you have nothing better to do, visit my blog  
<https://sanjaykthakur.wordpress.com/>





Good luck and  
happy learning  
Machine Learning