

CS230 Hands-on session 3: “Tensorflow Tutorial”

Kian Katanforoosh, Andrew Ng

Spring 2018

In this hands-on session, you will use two files:

- Tensorflow_tutorial.py (Part I)
- [CS230 project example code](#) repository on github (Part II)

Part I - Tensorflow Tutorial

The goal of this part is to quickly build a tensorflow code implementing a Neural Network to classify hand digits from the MNIST dataset.

The steps you are going to implement are:

- Load the dataset
- Define placeholders
- Define parameters of your model
- Define the model's graph (including the cost function)
- Define your accuracy metric
- Define the optimization method and the training step
- Initialize the tensorflow graph
- Optimize (loop)
- Compute training and testing accuracies

Question 1: Open the starter code “tensorflow_tutorial.py”. Tensorflow stores the MNIST dataset in one of its dependencies called “tensorflow.examples.tutorials.mnist”. This part is very specific to MNIST so we have coded it for you. Please read the code that loads MNIST.

Question 2: Define the tensorflow placeholders X (data) and Y (labels). Recall that the data is stored in 28x28 grayscale images, and the labels are between 0 and 9.

Question 3: For now, we are going to implement a very simple 2-layer neural network (*LINEAR->RELU->LINEAR->SOFTMAX*). Define the parameters of your model in tensorflow. Make sure your shapes match.

Question 4: Using the parameters defined in question (3), implement the forward propagation (from the input X to the output probabilities A). Don't forget to reshape your input, as you are using a fully-connected neural network.

Question 5: Recall that this is a 10-class classification task. What cost function should you use? Implement your cost function.

Question 6: What accuracy metric should you use? Implement your accuracy metric.

Question 7: Define the tensorflow optimizer you want to use, and the tensorflow training step. Running the training step in the tensorflow graph will perform one optimization step.

Question 8: As usual in tensorflow, you need to initialize the variables of the graph, create the tensorflow session and run the initializer on the session. Write code to do these steps.

Question 9: Implement the optimization loop for 20,000 steps. At every step, have to:

- Load the mini-batch of MNIST data (including images and labels)
- Create a feed dictionary to assign your placeholders to the data.
- Run the session defined above on the correct graph nodes to perform an optimization step and access the desired values of the graph.
- Print the cost and iteration number.

Question 10: Using your accuracy metric, compute the accuracy and the value of the cost function both on the train and test set.

Run the code from your terminal using: `python tensorflow_tutorial.py`

Question 11: Look at the outputs, accuracy and logs of your model. What improvements could be made? Take time at home to play with your code, and search for ideas online.

Part II - Project Code Examples

The goal of this part is to become more familiar with the [CS230 project example code](#) that the teaching staff has provided. It's meant to help you prototype ideas for your projects.

Question 1: Please start by git cloning the "[cs230-code-examples](#)" repository on your local computer.

This repository contains several project examples:

- Tensorflow vision (SIGNS dataset classification)
- Tensorflow nlp (Named Entity Recognition)
- Pytorch vision (SIGNS dataset classification)
- Pytorch nlp (Named Entity Recognition)

Question 2: You will start with the Tensorflow vision project example. On your terminal, and inside the cloned repository, navigate to `./tensorflow/vision`. Follow the instructions described in the "Requirements" section of the readme.

Question 3: Follow the guidelines described in the "Download the SIGNS dataset".

Question 4: Follow the guidelines described in the "Quickstart".

Question 5: Read through the code and find what you could modify. All the project examples are structured the same way, we invite you to try the one that matches your needs the best. This project example code has been coded to help you insert your dataset quickly, and start prototyping some results. It is meant to be a helper code for your project and for you to learn in-depth Tensorflow/Pytorch, it is not a starter code.