Exercises
Deep Learning
Fall 2018

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Fernando Pérez-Cruz
Web http://www.da.inf.ethz.ch/teaching/2018/DeepLearning/

Series Monday, September 24, 2018 (Deep Learning, Exercise series 1)

Problem 1 (Basic computations in Tensorflow):

While tensorflow is usually used in the context of deep learning, it's foundation is simply mathematical computations encapsulated in a graph structure. Let's do some basic computations to point out how Variables are used in tensorflow and how/when computations are performed.

We are given input vectors $\boldsymbol{x} \in \mathbb{R}^d$ that we want to compress (and decompress) to $\boldsymbol{x}' \in \mathbb{R}^{d'}, d' \ll d$. In principle we could perform $\boldsymbol{W}\boldsymbol{x}$ with $\boldsymbol{W} \in \mathbb{R}^{d' \times d}$. Here we want to perform this process in n steps where we will assume that $d - d'$ is divisible by n evenly for simplicity. That means

$$\boldsymbol{x}' = \boldsymbol{W}_n \boldsymbol{W}_{n-1} \dots \boldsymbol{W}_1 \boldsymbol{x} \qquad \boldsymbol{W}_1 \in \mathbb{R}^{d - \frac{d-d'}{n} \times d} \dots \boldsymbol{W}_n \in \mathbb{R}^{d' \times d' + \frac{d-d'}{n}} \tag{1}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{W}_1^\top \boldsymbol{W}_2^\top \dots \boldsymbol{W}_n^\top \boldsymbol{x}' \tag{2}$$

1. Implement the compression and decompression chained together $\boldsymbol{x} \to \boldsymbol{x}' \to \tilde{\boldsymbol{x}}$ in one tensorflow model. Given an $\boldsymbol{x}$, the model outputs $\tilde{\boldsymbol{x}}$. Follow these steps:

   - Create a `tf.placeholder` with shape[1] $[1, d]$ for $\boldsymbol{x}$.
   - Create the matrices $\boldsymbol{W}_1 \dots \boldsymbol{W}_n$ as tensorflow Variables and use `tf.get_variable` instead of `tf.Variable` to create them.
   - Use a random initialization for $\boldsymbol{W}_1 \dots \boldsymbol{W}_n$ such as random normal.

   Also note that we use *the same* $\boldsymbol{W}_i$ in the compression and in the decompression.

   Your code should roughly look like this

```
# Create graph...
x = ...
x_tilde = ...

# shape [1, d]
input = np.random(...)
feed_dict={x:input}
output = session.run(x_tilde, feed_dict=feed_dict)
```

2. The situation when re-using $\boldsymbol{W}_i$ to compute $\boldsymbol{W}_i^\top$ is a very common one. It's called *variable sharing* and there is a whole tutorial dedicated to it:

   `https://www.tensorflow.org/programmers_guide/variables`

   You have probably stored the variables in an array when creating them using `tf.get_variable` for $\boldsymbol{W}_i$ and used the array content when computing $\boldsymbol{W}_i^\top$. However, often it is quite cumbersome to carry around such arrays of references which is why tensorflow allows you to use `get_variable` multiple times with the same name to share variables. Read the tutorial and adapt your code so that sharing is performed without sharing python references explicitly.

[1] This will require you to think about $\boldsymbol{x}$ as a row vector. Typically we think of vectors as column vectors when we write math, but in DL models we will be using mini-batching (more on this later) which requires the above shape. You'll have to go through this for all material in the class and all papers/code that you might read.