Practical work 07 – 01/11/2018 Deep Learning Frameworks

Objectives

The main objective of this PW is to understand the overall functioning of TensorFlow, a computational graph framework used for machine learning and deep learning. More specifically, we want to experiment with the strategy of static graph used by Tensorflow which is in two steps : (1) declaration of the graph and (2) use of the graph.

This practical work is supposed to be shorter than the previous ones and we then ask you to submit the solution for next week.

Submission

- **Deadline** : Wednesday 7 November, 12am
- **Format** :
  - Zip with report and iPython notebook.

Exercice 1 Computational graph exercise

The logistic regression (LR)[1] is a well known classification algorithm that outputs values between 0 and 1.0 with the following equation :

$$\hat{y} = h_\theta(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x} + b)} \tag{1}$$

where the parameters θ of the model are defined with a weight vector $\mathbf{w}$ and a scalar value b. Assuming an input space with 2 dimensions, the LR function becomes :

1. LR is actually similar to the generalised perceptron - cf week 2 slide 27.

$$h_\theta(x_1, x_2) = \frac{1}{1 + \exp(-(w_1 x_1 + w_2 x_2 + b))} \tag{2}$$

a) Revise slides 28–32 from the class and propose a computational graph with an input space of 2 dimensions (Eq. 2). Use operators as granular as you can. You should have about 7 nodes in your graph.

b) On the graph, illustrate the forward pass using numerical values.

c) Illustrate the back-propagation of a gradient value of 1.0 at the output of the graph using the numerical values of b).

Exercice 2 MLP with TensorFlow

The Figure 1 below illustrates a Multi-Layer Perceptron. The network takes inputs of dimension D, has H hidden neurons and K output neurons. In this case we assume that all activation functions of the neurons are sigmoids. The objective of the exercise is to implement computational graphs in TensorFlow to train and use such an architecture. The constraints we put ourselves is to use "low-level" functions of TensorFlow, i.e. we will not use high-level functions to compose layers and to train the parameters.

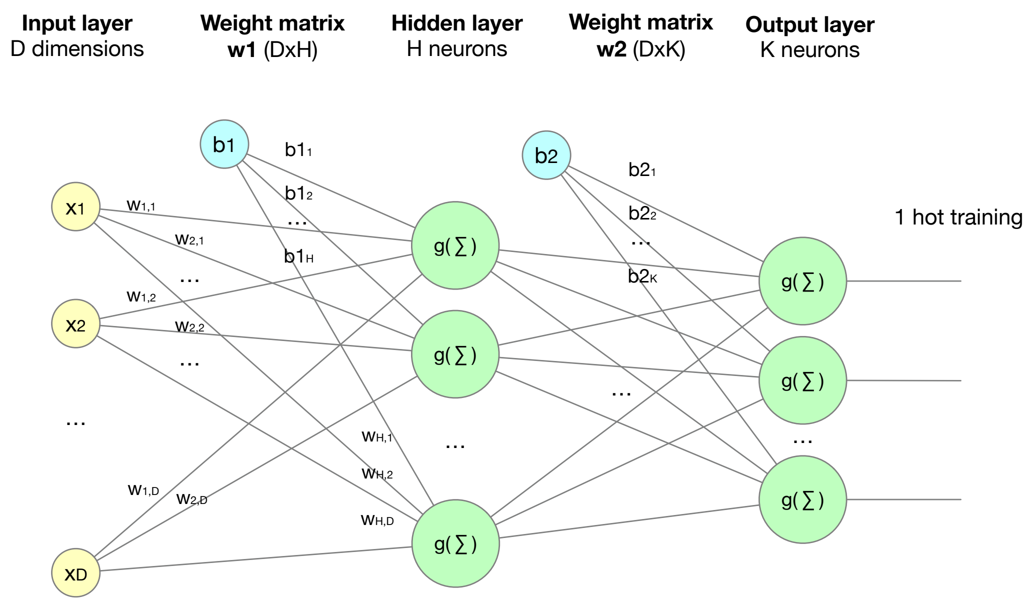

FIGURE 1 – Multi-layer perceptron with one hidden layer.

a) Revise slides 40–55 from the class on TensorFlow and make sure you understand the principles and the different steps that were described.

b) Use the iPython notebook provided on Moodle and complete the missing parts of the code in order to perform a training and testing phase. We assume that the loss function used will be the Mean Squared Error (MSE) loss function (see slide 30 week 2).

c) Make sure that the network is converging and compute the error rate on the test set. You should observe an error rate below 10.0%.

Optional 1. Implement a simple learning schedule strategy (piecewise) as explained in slide 41 of last week's class.

Optional 2. Redo this exercise using a softmax activation at the output layer and with a cross-entropy loss function.

Exercice 3 Optional : Review Questions

a) Explain the differences between cpu, gpu and tpu.

b) What do we mean by gpu bottleneck ?

c) Explain the advantages of computational graphs for learning strategies.

d) What are the expected advantages / disadvantages of a static graph strategy (TensorFlow) versus a dynamic graph stategy (Pytorch) ?