Homework 1 for
Introduction to Natural Language Processing

Deadline:2019.10.27 23:59:59

● Part I: Review on Deep Learning

1. For the multiple-layer neural network model below, show that nonlinearity will not be achieved if the activation function f(.) is chosen as a linear function.

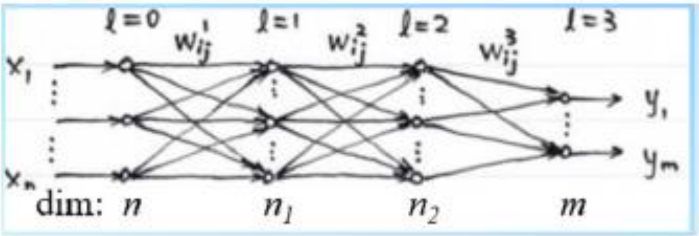
$$g_k(\mathbf{x}) \equiv y_k = f\left(\sum_j w_{kj} f\left(\sum_i w_{ji} x_i + w_{j0} \right) + w_{k0} \right),$$

2. Consider a single neuron with Sigmoid activation function $s(z) = 1/(1+e^{-z})$. The input of this neuron is $X = (x_0, \ldots, x_n)^T$ and the output is $y = s(W^T X)$, whose weight vector being $W = (w_0, \ldots, w_n)^T$. The error function is $E = 0.5(g-y)^2$, where g is the true label of samples.

(1)Write the weight-updating formula (Denote the learning rate as λ)

(2) Initially, the weight vector $X = (0.5,1,1)^T$. If $X = (1,2,0.5)^T$, g=1, λ=0.1. Write the new values of weight vector updated by one-step error back propagation.

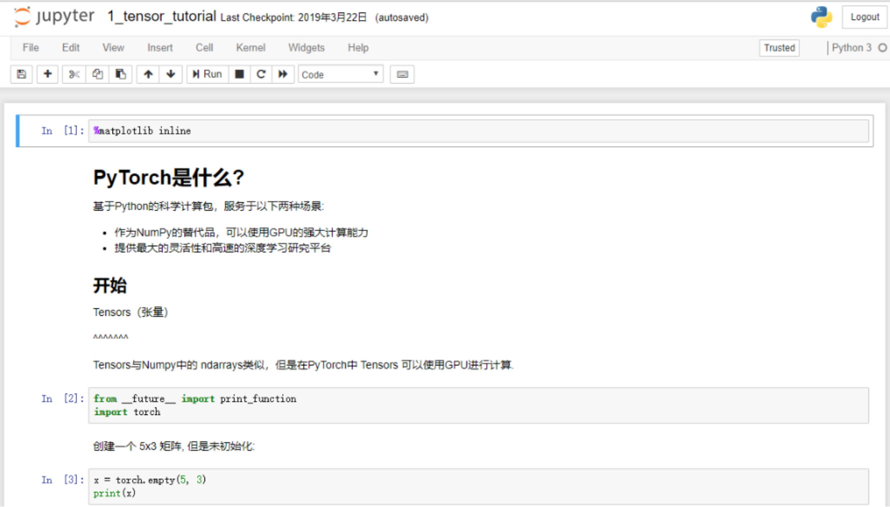
3. (Optional) For a 4-layered MLP(with 3 hidden layers), derive the BP algorithm one by one layer.



● Part II: Programming Practice

1. Install Pytorch and Jupyter notebook. And open the reference materails of Lesson1

(Path：/资料/Pytorch-中文学习手册/Chapter1) by Jupyter notebook.



```
In [1]: %matplotlib inline
```

PyTorch是什么?

基于Python的科学计算包，服务于以下两种场景:

- 作为NumPy的替代品，可以使用GPU的强大计算能力
- 提供最大的灵活性和高速的深度学习研究平台

开始

Tensors（张量）

^^^^^^^

Tensors与Numpy中的 ndarrays类似，但是在PyTorch中 Tensors 可以使用GPU进行计算.

```
In [2]: from __future__ import print_function
        import torch
```

创建一个 5x3 矩阵, 但是未初始化:

```
In [3]: x = torch.empty(5, 3)
        print(x)
```

Reading all ipynb files in Chapter 1 (and running the codes in them) is highly recommended if you are not familiar with Pytorch yet.

The MNIST database of handwritten digits (http://yann.lecun.com/exdb/mnist/) has a training set of 60,000 examples, and a test set of 10,000 examples. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. Let's start our travel from here!

2. Please construct a Deep-learning model to classify the handwritten digits in MNIST **using pytorch**. (Massive resources for this toy-task are available on Internet. You can attach to and view them if it'a a little tough for you. But copying source codes directly from websites is prohibited)

3. (Optional) A more challenging task is from the *Assignment 1, Introduction to Deep Learning, Fall 2019, Tsinghua University*. In this task, you are required to build several basic blocks in Neural Networks (Forward Propagation, Backward Propagation, Softmax Classifier, etc.) **without the help of Pytorch**. The framework of this network and more detailed instruction have been provided in appendix. Though the direct help for your Pytorch skill is limited, you will have a much more thorough understanding of Deep-learning mechanism by completing this task.