FAET630004: Computer & AI Architecture **(Due: 05/15/19)**

Homework Assignment #3

Instructor: Chixiao Chen *Name:* , *FudanID:*

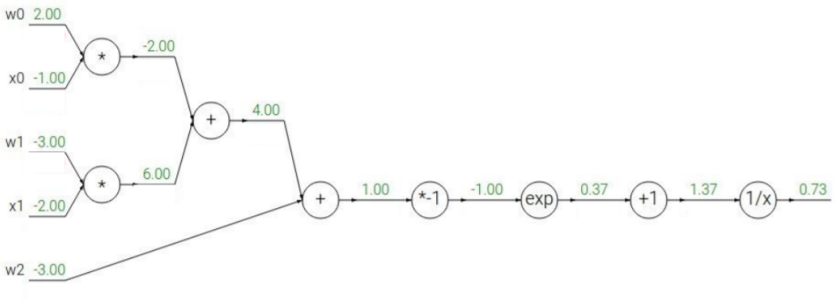
- This HW counts 15% of your final score, please treat it carefully.
- Please submit the electronic copy via mail: cxchen@fudan.edu.cn before 05/05/2019 11:59pm.
- It is encouraged to use LaTeX to edit it, the source code of the assignment is available via: https://www.overleaf.com/read/fwgzmxzkxrtd
- You can also open it by Office Word, and save it as a .doc file for easy editing. Also, you can print it out, complete it and scan it by your cellphone.
- Problem 2 needs python and numpy. If you do not have a local python environment, please use an online version https://colab.research.google.com/.
- You can answer the assignment either in Chinese or English

Problem 1: Gradient Computing (30 points)

Assuming that one loss function in a classifier has the following output expression:

$$f(x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}},$$

and the current state is shown below:



Please compute all the weight gradients $\frac{\partial f}{\partial w_i}, i = 0, 1, 2$.

Problem 2: Training a two-layer neural network using Numpy (70 points)

Assuming you have a tiny dataset which has 8 inputs, 4 classes and 500 samples. Please design a two-layer neural network as the classifier. Both forward (inference) and backward (training) propagation are required. The first 400 samples are for training, and the last 100 samples are for test. The dataset is available via: https://cihlab.github.io/course/dataset.txt. The activation function is ReLU in the case.

The following table is an example interpretation of the dataset file. (The first two lines of the file is illustrated.)

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Class Label
0.4812	0.7790	0.8904	0.7361	0.9552	0.2119	0.7992	0.2409	4
0.4472	0.5985	0.7859	0.5035	0.6912	0.4038	0.0787	0.2301	1

Please submit you code and a brief report with the loss function definition, the final accuracy results, the neuron number in the hidden layers, etc. Also include your strategy for batch size and learning rate. (Hint: It is encouraged to use python and numpy (https://www.numpy.org/). You can refer to the slides 34 in the lecture 7 notes. The problem does not encourage you to use Tensorflow/caffe/pytorch, but if you have no idea about numpy, you can also using these frameworks.)