Seung Jun Baek

## 1 Outline

In this assignment, you are asked to design softmax classifier.

## 2 Specification

**In this assignment, you are asked to write a Python code for softmax classifier.** The dataset is based on randomly generated points in $\mathbb{R}^2$ according to Gaussian distribution. Each class will have its mean vector and covariance matrix. An example plot of the synthesized dataset is shown in Fig. 1.

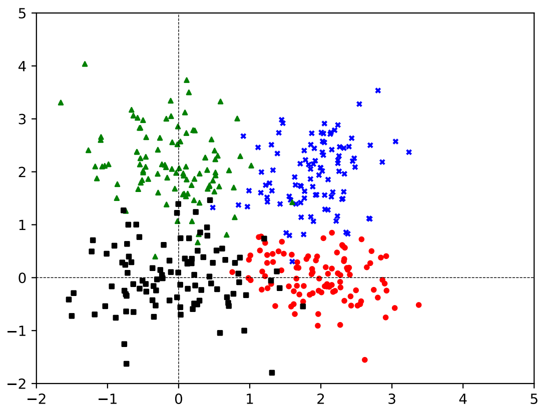

Figure 1: Plot of sample training dataset points with 4 classes.

The module for data generation is given in the file `data_generator.py`. You will need to put this file in the same directory as `hw2.py`. You will need to modify `hw2.py` file to implement the classifiers.

The module `data_generator.py` contains the following function:

```
generate(number=100, seed=None, plot=True, dataset=True,
num_class=3,sigma=1.0)
```

The parameters are:

- `number`: the total number of data points to generate. This number will be divided into the number of classes in the dataset. For example, if there are 4 classes, the data points per class will be 100/4=25.
- `seed`: to control the seed for random number generator.
- `plot`: toggles the plotting of data points.
- `num_class`: number of classes in the dataset.
- `sigma`: Sigma parameter controls the degree of scattering of data points around the mean. Larger sigma means larger variance in the dataset, which will result in lower accuracy. Default value is 1.0. For default value of sigma=1.0, you will see the accuracy will be around 80–90% for all classifiers. When sigma=2.0, you will see the accuracy drops to 60–70%.
- returns `input_value`,`output_value` where `input_value` is the input dataset, a matrix of shape `number` by 2; `output_value` is the output dataset, which contains the labels from 0 to `num_class`-1, a list of size `number`.

## 2.1 softmax classifier

You are asked to design a softmax classifier. You need to complete the function
`cross_entropy_softmax_loss(Wb, x, y, num_class, n, feat_dim)`

- `n`: dataset size.
- `num_class`: the number of classes to classify
- `feat_dim`: the dimension of input data. In this assignment, this value will be set to 2.
- `x`: input data. a matrix of shape `(n,feat_dim)`
- `y`: output data. a list of shape `(n,)`. Contains ground truth labels, that is, a number ranging from 0 to `num_class`-1.
- `Wb`: a column vector of size `num_class*feat_dim` + `num_class`. The first `num_class*feat_dim` elements of the vector are the parameters for $W$ which is `num_class` by `feat_dim` matrix. The last `num_class` elements of the vector are the parameters for $b$, the bias.

```
Wb = np.reshape(Wb, (-1, 1))
b = Wb[-num_class:]
W = np.reshape(Wb[range(num_class * feat_dim)], (num_class, feat_dim))
x=np.reshape(x.T, (-1, n))

# this will give you a score matrix s of size (num_class)-by-(n)
# the i-th column vector of s will be
# the score vector of size (num_class)-by-1, for the i-th input data point
# performing s=Wx+b

s=W@x+b
```

- returns the cross-entropy loss (should be nonnegative), averaged over the dataset.

**If necessary, you can define and use whatever extra functions that is needed to implement this function.**

## 3 What to submit

You will be given `data_generator.py` and `hw2.py`. Place them on the same directory, and run `hw2.py` with the properly implemented functions.

- Submit a modified Python file `hw2.py`. The requirement is to implement the function `cross_entropy_softmax_loss`.
- Upload your `hw2.py` file at Blackboard before deadline. (Please submit the file in time, no late submission will be accepted).

## 4 How to test your module

Run your completed code `hw2.py`.
Example results

```
number of classes:  4  sigma for data scatter: 2.0
generating training data
400 data points generated. Seed is random.
generating test data
100 data points generated. Seed is random.
training softmax classifier...
testing softmax classifier...
accuracy of softmax loss:  75.0 %
```

The errors in the output may vary. However, with `num_class` equal to 4 and `sigma` equal to 1.0, your accuracy should be around 90%.

## 5 Grading

We will test your classifier with `num_class` equal to 4 and `sigma` equal to 1.0.

- 10 points if your module works correctly. Specifically, if your accuracy is more than 85%.
- 3 points if the accuracy is below 85%.

- 0 point if you do not submit the file by deadline.

In the blackboard, you can upload your file as many times as you like, before the deadline. The last uploaded file will be used for grading. After deadline, the submission menu will be closed and you will not be able to make submission.