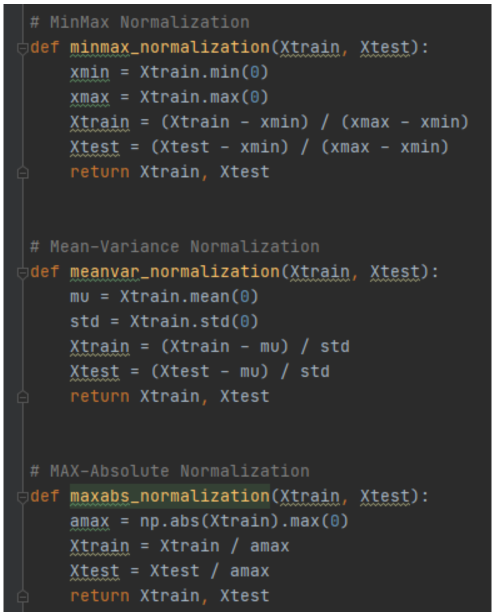
Ayah Hamdan

Lab 3: Speech Commands

The goal of this lab is to infer the words expressed by a speaker from a voice recording by building multilayer perceptron.

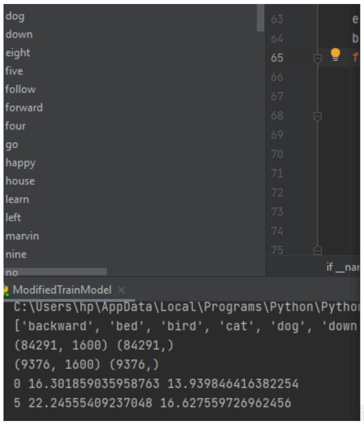
Normalization is used to reduce the range of values because we have a wide range. And to test the different types of Feature Normalization, multiple functions were written, as shown in the figure below:



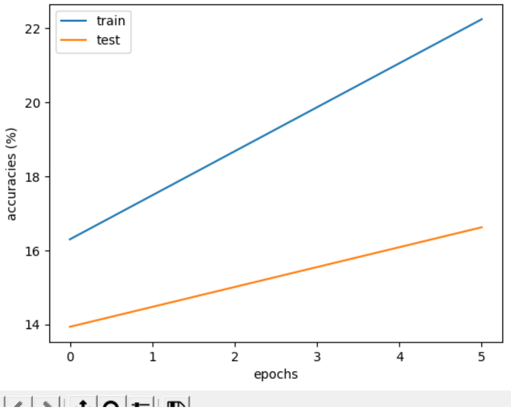
```python
# MinMax Normalization
def minmax_normalization(Xtrain, Xtest):
    xmin = Xtrain.min(0)
    xmax = Xtrain.max(0)
    Xtrain = (Xtrain - xmin) / (xmax - xmin)
    Xtest = (Xtest - xmin) / (xmax - xmin)
    return Xtrain, Xtest


# Mean-Variance Normalization
def meanvar_normalization(Xtrain, Xtest):
    mu = Xtrain.mean(0)
    std = Xtrain.std(0)
    Xtrain = (Xtrain - mu) / std
    Xtest = (Xtest - mu) / std
    return Xtrain, Xtest


# MAX-Absolute Normalization
def maxabs_normalization(Xtrain, Xtest):
    amax = np.abs(Xtrain).max(0)
    Xtrain = Xtrain / amax
    Xtest = Xtest / amax
    return Xtrain, Xtest
```

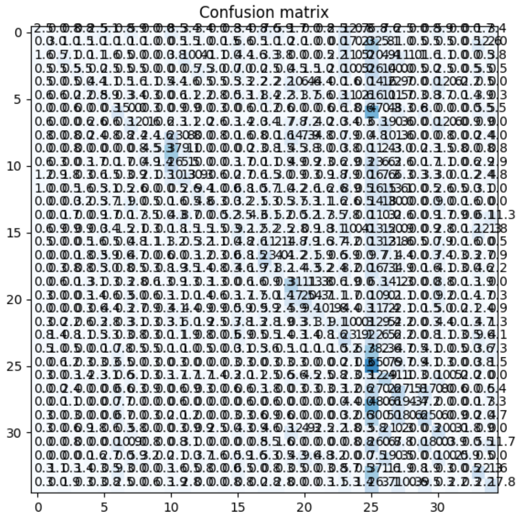
And the results for each Feature Normalization are as shown in the figures below, where it can be noticed that the best results occurred when using the Mean/Variance Normalization where it had the biggest accuracies.

Mean/ Variance Normalization:

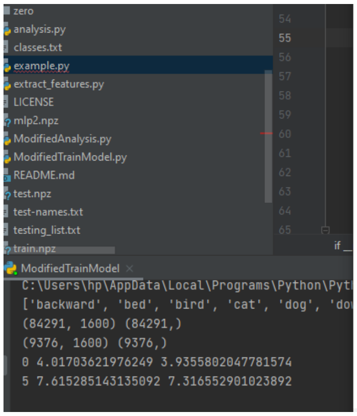


```
C:\Users\hp\AppData\Local\Programs\Python\Pytho
['backward', 'bed', 'bird', 'cat', 'dog', 'down
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 16.301859035958763 13.939846416382254
5 22.24555409237048 16.627559726962456
```

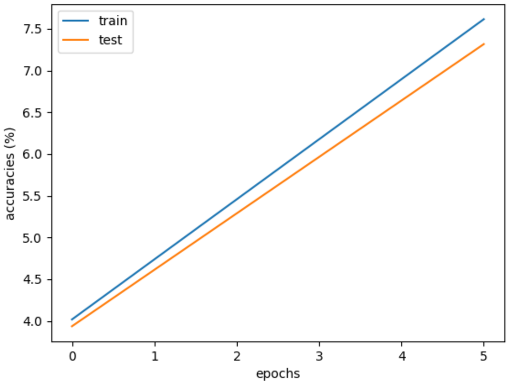


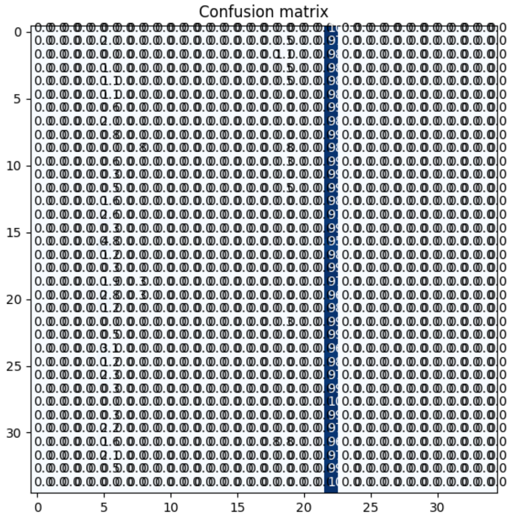


Min/ Max Normalization:

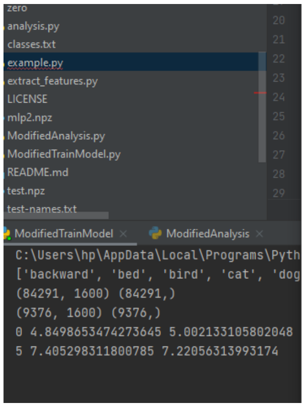


```
C:\Users\hp\AppData\Local\Programs\Python\Pyt
['backward', 'bed', 'bird', 'cat', 'dog', 'do
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 4.01703621976249 3.9355802047781574
5 7.615285143135092 7.316552901023892
```

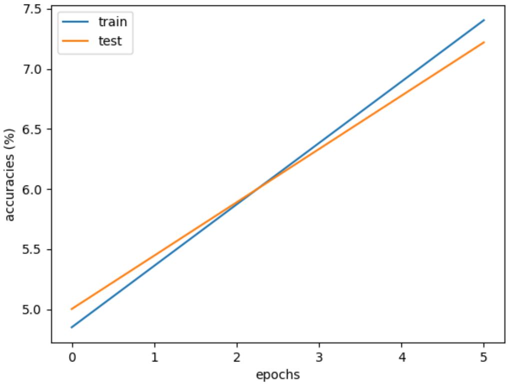


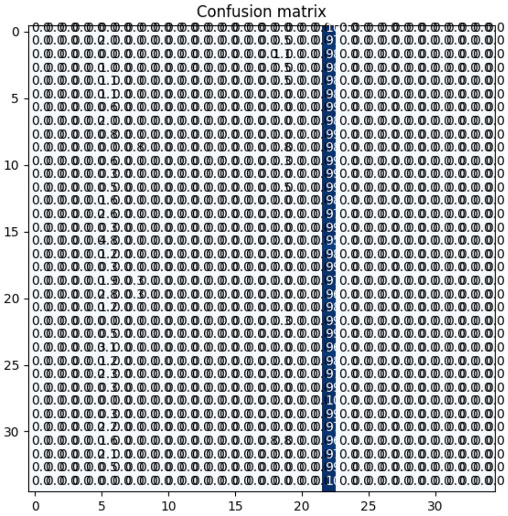


Max/Absolute Normalization:

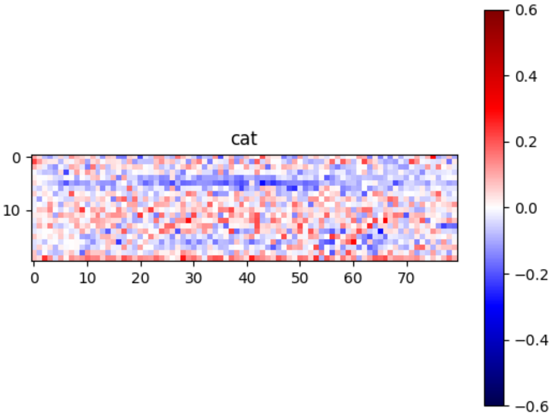


```
C:\Users\hp\AppData\Local\Programs\Pyth
['backward', 'bed', 'bird', 'cat', 'dog
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 4.8498653474273645 5.002133105802048
5 7.405298311800785 7.22056313993174
```





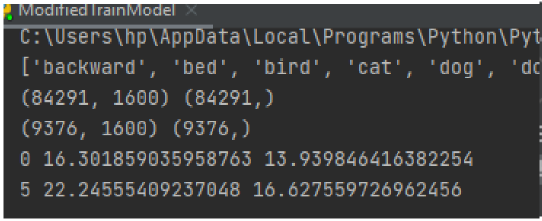
The following image shows the set of weights of the MLP without hidden layers for class 3, and it can be noticed that the positive weight occurs in the second part of the wave while the negative weight occurs as a line in the first part and the speaker isn't saying much.



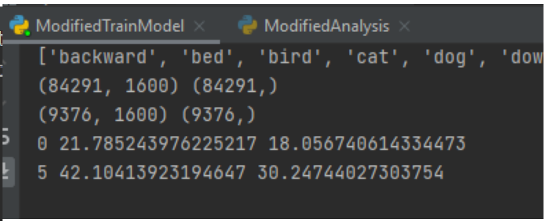
Different Architectures:

Adding multiple layers with multiple neurons to the neural network will enhance the results, until reaching a point where overfitting will occur. The following pictures and table show the different architecture consisting of different hidden layers, and it can be noticed that the accuracy has increased each time we increased the number of hidden layers, and when increasing the number of neurons to the same number of layers.

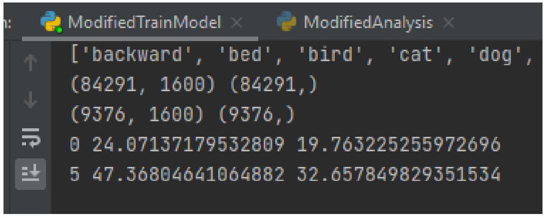
Number of neurons in layers	Training Accuracy	Test Accuracy
[1600, 1]	22.26	16.86
[1600, 200, 1]	42.10	30.25
[1600, 600, 1]	47.368	32.658
[1600, 200, 130, 1]	47.68	37.98



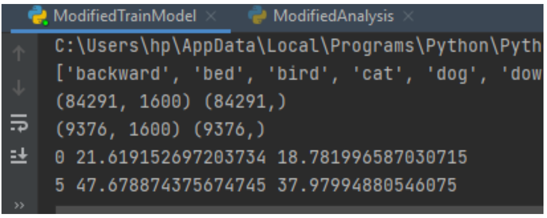
```
C:\Users\hp\AppData\Local\Programs\Python\Pyt
['backward', 'bed', 'bird', 'cat', 'dog', 'dc
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 16.301859035958763 13.939846416382254
5 22.24555409237048 16.627559726962456
```



```
['backward', 'bed', 'bird', 'cat', 'dog', 'dow
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 21.785243976225217 18.056740614334473
5 42.10413923194647 30.24744027303754
```

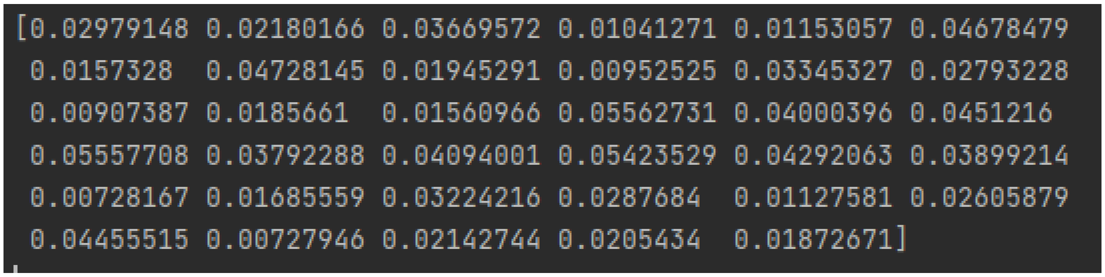


```
['backward', 'bed', 'bird', 'cat', 'dog',
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 24.07137179532809 19.763225255972696
5 47.36804641064882 32.657849829351534
```



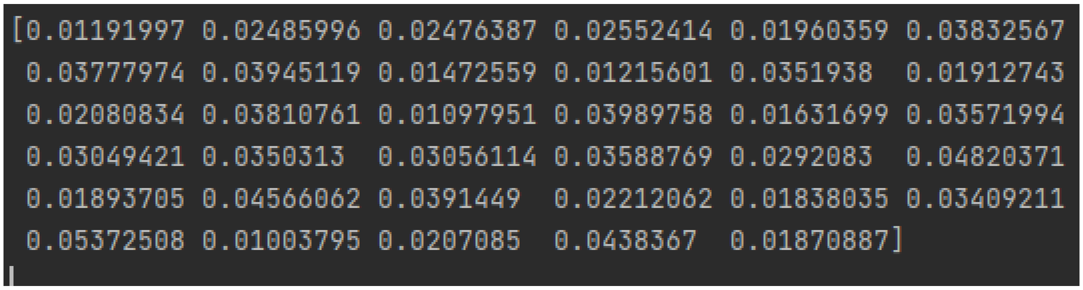
```
C:\Users\hp\AppData\Local\Programs\Python\Pyth
['backward', 'bed', 'bird', 'cat', 'dog', 'dow
(84291, 1600) (84291,)
(9376, 1600) (9376,)
0 21.619152697203734 18.781996587030715
5 47.678874375674745 37.97994880546075
```

To find the worst mistakes in finding the words, the probability from the inference function can be used, which is the probability estimated by the multilayer perceptron. For example, in class 1 ('bed'), the lowest probability is for class 32 ('visual).



```
[0.02979148 0.02180166 0.03669572 0.01041271 0.01153057 0.04678479
 0.0157328  0.04728145 0.01945291 0.00952525 0.03345327 0.02793228
 0.00907387 0.0185661  0.01560966 0.05562731 0.04000396 0.0451216
 0.05557708 0.03792288 0.04094001 0.05423529 0.04292063 0.03899214
 0.00728167 0.01685559 0.03224216 0.0287684  0.01127581 0.02605879
 0.04455515 0.00727946 0.02142744 0.0205434  0.01872671]
```

And for class 30('two'), the lowest probability is for class 32 ('visual):



```
[0.01191997 0.02485996 0.02476387 0.02552414 0.01960359 0.03832567
 0.03777974 0.03945119 0.01472559 0.01215601 0.0351938  0.01912743
 0.02080834 0.03810761 0.01097951 0.03989758 0.01631699 0.03571994
 0.03049421 0.0350313  0.03056114 0.03588769 0.0292083  0.04820371
 0.01893705 0.04566062 0.0391449  0.02212062 0.01838035 0.03409211
 0.05372508 0.01003795 0.0207085  0.0438367  0.01870887]
```

And the latter can happen due to many reasons such as existence of a heavy accent in the clip, or the clip is damaged for some reason, or there is wrong in the data set (someone put a wrong label).

I affirm that this report is the result of my own work and that I did not share any part of it with anyone else except the teacher.