Max Marmer
Carmel Ron
Odelia Hochman
Efrat Cohen

Deep learning

The project deals with the voice recognition of women and men.
Identification is done by logistic regression.

We first took wav files of women and men and cut them all out for about a second.

The features we used are features of the mfcc directory, the number of features is 1287 for one second.

We first took one type of female voice and one type of male voice.

We divided the data into two parts: train, test. In the train folder we put 700 audio files per gender, and in the test folder we put 300 audio files per gender.

In the function- def prepare_data (path):

We read the dataset and created a variable- **data_y** ,which is a vector of the number of samples and contains zeros and unities according to the audio files.

if the file is a man's audio we put 0 and if the file is a woman's audio we put 1.

We created a variable of **data_x** which is a matrix, the number of columns as the size of the number of features and the number of rows as the size of the examples.

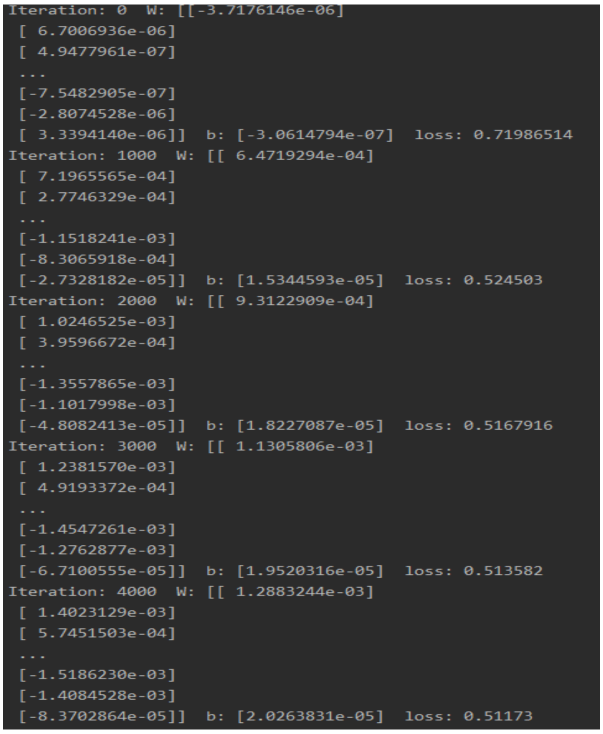
The first time we ran the model we got in the loss function: **nan**.

To deal with this problem, we used tensorflow function:

y = tf.nn.sigmoid(tf.matmul(x,W)+b)

loss1 = tf.nn.sigmoid_cross_entropy_with_logits(labels=y_, logits=y)

loss = tf.reduce_mean(loss1)



```
Iteration: 0  W: [[-3.7176146e-06]
 [ 6.7006936e-06]
 [ 4.9477961e-07]
 ...
 [-7.5482905e-07]
 [-2.8074528e-06]
 [ 3.3394140e-06]]  b: [-3.0614794e-07]  loss: 0.71986514
Iteration: 1000  W: [[ 6.4719294e-04]
 [ 7.1965565e-04]
 [ 2.7746329e-04]
 ...
 [-1.1518241e-03]
 [-8.3065918e-04]
 [-2.7328182e-05]]  b: [1.5344593e-05]  loss: 0.524503
Iteration: 2000  W: [[ 9.3122909e-04]
 [ 1.0246525e-03]
 [ 3.9596672e-04]
 ...
 [-1.3557865e-03]
 [-1.1017998e-03]
 [-4.8082413e-05]]  b: [1.8227087e-05]  loss: 0.5167916
Iteration: 3000  W: [[ 1.1305806e-03]
 [ 1.2381570e-03]
 [ 4.9193372e-04]
 ...
 [-1.4547261e-03]
 [-1.2762877e-03]
 [-6.7100555e-05]]  b: [1.9520316e-05]  loss: 0.513582
Iteration: 4000  W: [[ 1.2883244e-03]
 [ 1.4023129e-03]
 [ 5.7451503e-04]
 ...
 [-1.5186230e-03]
 [-1.4084528e-03]
 [-8.3702864e-05]]  b: [2.0263831e-05]  loss: 0.51173
```

Results	Classified As Female	Classified As Male
Really Female	297	3
Really Male	0	300

Accuracy: 297+300/600=0.995

Recall: 297/300=0.99

Precision:297/297=1

F-Measure: 2*1*0.99/1+0.99=0.99497

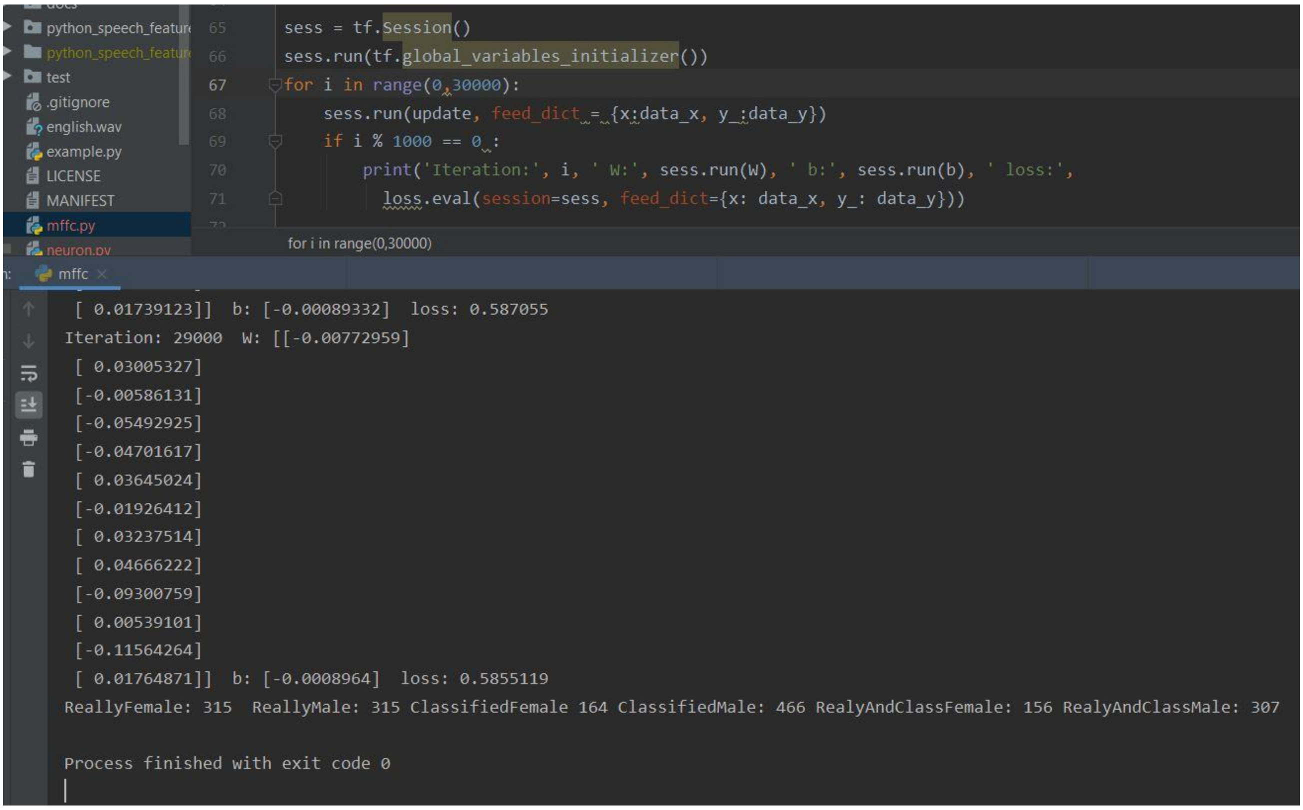
To complicate the model, we added more voices.

To the train folder we added two more types of voices for each gender and to the test folder we added one type of voice that does not exist in the train.

The train contains 2100 examples and the test contains 630 examples.

In addition, we change the number of features from 1287 to 13.

We ran the examples in the logistic regression and we got that the accuracy is 73.4%.



```
sess = tf.Session()
sess.run(tf.global_variables_initializer())
for i in range(0,30000):
    sess.run(update, feed_dict = {x:data_x, y_:data_y})
    if i % 1000 == 0 :
        print('Iteration:', i, ' W:', sess.run(W), ' b:', sess.run(b), ' loss:',
          loss.eval(session=sess, feed_dict={x: data_x, y_: data_y}))
```

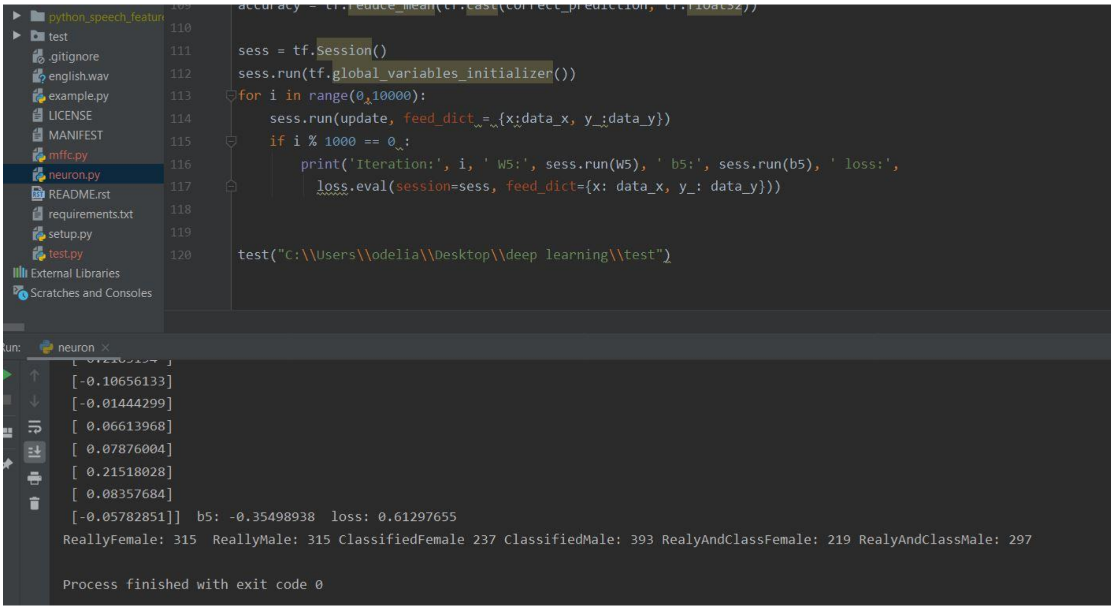
```
 [ 0.01739123]]  b: [-0.00089332]  loss: 0.587055
Iteration: 29000  W: [[-0.00772959]
 [ 0.03005327]
 [-0.00586131]
 [-0.05492925]
 [-0.04701617]
 [ 0.03645024]
 [-0.01926412]
 [ 0.03237514]
 [ 0.04666222]
 [-0.09300759]
 [ 0.00539101]
 [-0.11564264]
 [ 0.01764871]]  b: [-0.0008964]  loss: 0.5855119
ReallyFemale: 315  ReallyMale: 315 ClassifiedFemale 164 ClassifiedMale: 466 RealyAndClassFemale: 156 RealyAndClassMale: 307

Process finished with exit code 0
```

Neuron network

In the network construction, changes were made each time in the number of layers, the number of neurons in each layer, the alpha, and the number of iterations.

we did iterations of four hidden layers with different number of neurons, in
this example we choose neurons(70,100,60,30).
Only after about 10,000 iterations, the model improved and the accuracy was 81.9%.



```
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

sess = tf.Session()
sess.run(tf.global_variables_initializer())
for i in range(0,10000):
    sess.run(update, feed_dict = {x:data_x, y_:data_y})
    if i % 1000 == 0 :
        print('Iteration:', i, ' W5:', sess.run(W5), ' b5:', sess.run(b5), ' loss:',
          loss.eval(session=sess, feed_dict={x: data_x, y_: data_y}))


test("C:\\Users\\odelia\\Desktop\\deep learning\\test")
```

```
  [-0.10656133]
  [-0.01444299]
  [ 0.06613968]
  [ 0.07876004]
  [ 0.21518028]
  [ 0.08357684]
  [-0.05782851]]  b5: -0.35498938  loss: 0.61297655
ReallyFemale: 315  ReallyMale: 315 ClassifiedFemale 237 ClassifiedMale: 393 RealyAndClassFemale: 219 RealyAndClassMale: 297

Process finished with exit code 0
```

We used the ReLU activation function to calculate the gradient in simply.

We also used the sigmoid function of logistic regression.

Results	Classified As Female	Classified As Male
Really Female	219	96
Really Male	18	297

Accuracy: 219+297/630=0.819

Recall: 219/315=0.695

Precision:219/237=0.92

F-Measure: 2*0.92*0.695/0.92+0.695=0.7918

Conclusions:

When we ran one type of sound, the accuracy of the model was very high (99%), and as we complicated the model and added more sound types then we had to increase the number of iterations in order to improve the model.

At first we used 1287 features and saw no significant improvement between the logistic regression and neurons (only 2% improvement), so we decided to calculate the average of all features.

After several iterations, we noticed that the number of neurons in each hidden layer had to be increased from the number that had been so far, because it does not identify most of the women's voices.
And we noticed that the number of neurons in the second layer should be higher than the number of neurons in the first layer.

In the neuronal network that we built, we saw that when we ran more iterations with a high number of neurons our loss dropped, but the test error was high (overfitting).