Assignment 5 Report
Liyang Ru

Part 1

Describe the dataset of faces. In particular, provide at least three examples of the images in the dataset as well as at least three examples of cropped out faces.

A good description will:

- **Comment on the quality of the dataset: are the provided bounding boxes accurate?**
- **Explain why the preprocessing is important in the context of classifying faces**
- **Explain the effect of resizing on the images**

1. Most of the provided bounding boxes are accurate (only 1 or 2 pictures wrongly cropped) and it can roughly crop out face from the pictures in a box.
2. The preprocessing is important in the context of classifying faces because when we do face recognizing, we only need our model learn (and test) from the faces of the people instead of the whole picture. If we do not crop out the faces correctly, the model may form wrong memories and reduce the accuracy when it recognizes faces.
3. The effects of resizing on the images are:
   (1). Standardize the cropped pictures, then the image information we get will be the same size and the learned features of the faces are bounded to the locations.
   (2). Reduce the size of each pictures, and we can do the learning algorithm more rapidly and more efficiently. And it will reduce the accuracy of recognizing.


bracco0.jpg


bracco1.jpg


bracco2.jpg


bracco0.jpg


bracco1.jpg

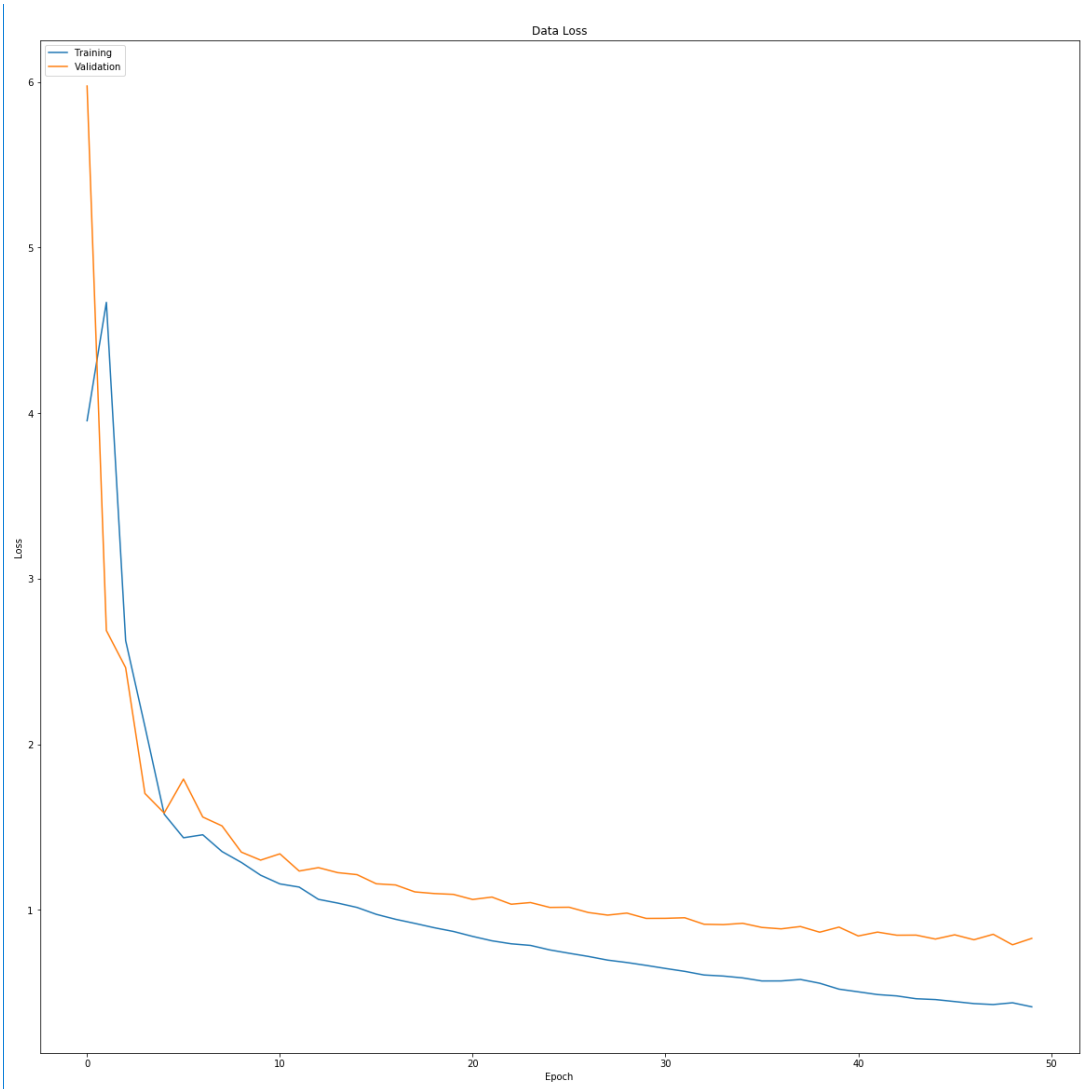

bracco2.jpg

Part 2

- **Plot the loss graphs by epoch (as shown in the class slides) This should have two lines one for the training loss and one for the validation loss.**
- **Evaluate your final model on the test set.**
- **Include a description of your model. In particular, describe how you separated the data into training, text, and validation sets, how you preprocessed the inputs and**

initialized the weights, what activation function you used, and how many nodes in the hidden layer you selected.

1& 2.




```
Test loss: 0.8309500133782103
Test accuracy: 0.7543859481811523
```

3.

We read the images from the folder “Processed” and convert the image into numpy array, storing all the image into a 3-dimensional numpy array as X. Then we read the filenames and cut the filename string into alphabet string. (For example, “bracco0.jpg” will be convert into “bracco”) Then save the string into a list as y. Therefore, we get the features and the labels.

We split the data into Train and Test with 4:1 after we import data, so we get train data + validation data (80%) and test data (20%). In the function, we split the data again into Train and Validation with 4:1, so we have Train data (64%), Validation data (16%) and Test data (20%) in total.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(preProcessedImages, labels, test_size=0.2, random_state=1)
```

After we split the data, we build the input vector from the 60x60 pixels and flatten the data array into 2 dimensions. We also normalize the data to help with the training by dividing the matrix with 255. At the same time, we will do one-hot encoding using keras' numpy-related utilities on the labels.

```
train_num = X_train.shape[0]
val_num = X_val.shape[0]
X_train = X_train.reshape(train_num, 3600)
X_val = X_val.reshape(val_num, 3600)
X_train = X_train.astype('float32')
X_val = X_val.astype('float32')

X_train /= 255
X_val /= 255

Y_train = keras.utils.to_categorical(y_train, n_classes)
Y_val = keras.utils.to_categorical(y_val, n_classes)
```

When we build the model, we use the default weight initialization. We use “relu” activation function in the hidden layer and “softmax” activation function in the output layer. We select 512 neurons (nodes) in the hidden layer.

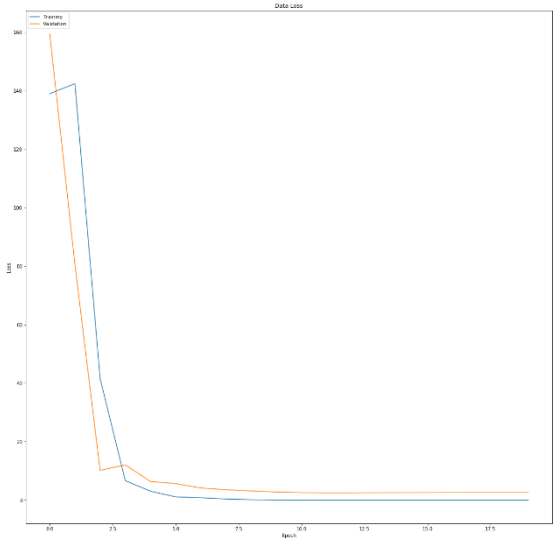
Part 3

- Explain your choice of VGG16 layer used as input.

- Compare the performance of your new network which used features from VGG6 to your network using the raw images from part 2. Is it better or worse? Why do you think this is?

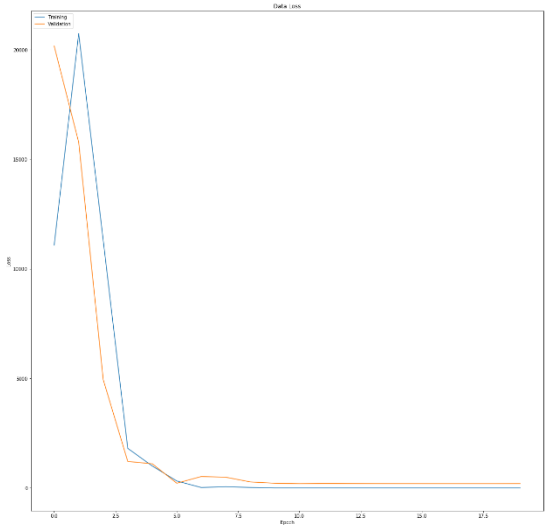
1. We choose “block5_pool” as the VGG16 layer as input. “block5_pool” runs much faster and efficiently than “block4_pool” since the size of extracted features is smaller. And we can get much lower loss and higher accuracy by using “block5_pool” than “block4_pool”.

block5_pool



```
Test loss: 3.339356037608364
Test accuracy: 0.8070175647735596
```

block4_pool



```
Test loss: 179.55335115131578
Test accuracy: 0.7543859481811523
```

2. VGG is better than the network in part 2 since the accuracy of VGG is higher than the network in part2. The test loss of VGG is also higher than the network of part2 since the features of VGG is much more than that of network in part2, so the percent of loss of VGG is also lower than that of the network in part2. On the other hand, the validation loss and training loss decrease more rapidly in VGG than that in the network in part2.

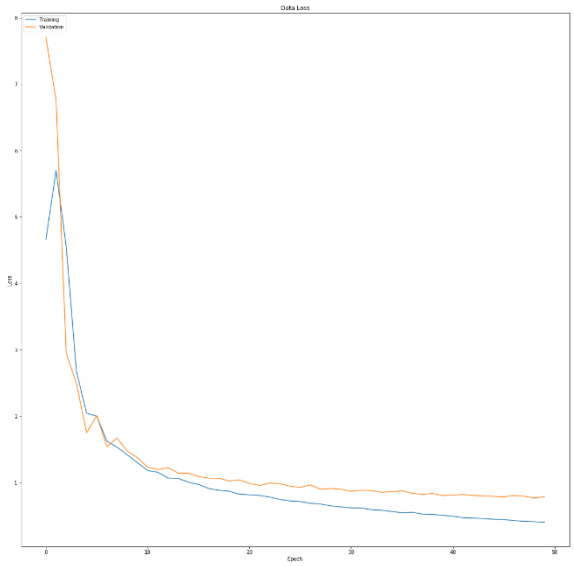
Part 4

- **Write a 1-2 page summary of your experiments, including the parameter settings that produced the best strategy.**
- **You should discuss how efficient the model is as well as the variation between experiments.**
- **Be sure to use evidence (screenshots, graphs, etc) to back your claims.**
- **Compare the best models from these experiments to performance in parts 2 and 4 – if you saw an improvement, why do you think that is? Likewise if you were not able to get an improvement, why do you think this was the case?**

Comparison in dropout and number of hidden layers

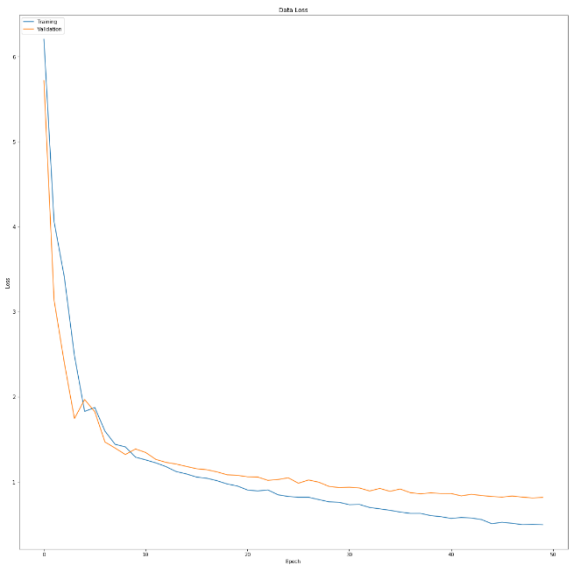
	Test	1 hidden layer	2 hidden layers	3 hidden layers
With Dropout	Loss	0.86	0.80	0.96
	Accuracy	0.70	0.78	0.70
Without Dropout	Loss	0.81	0.84	0.90
	Accuracy	0.73	0.72	0.75

Without dropout, 1 hidden layer



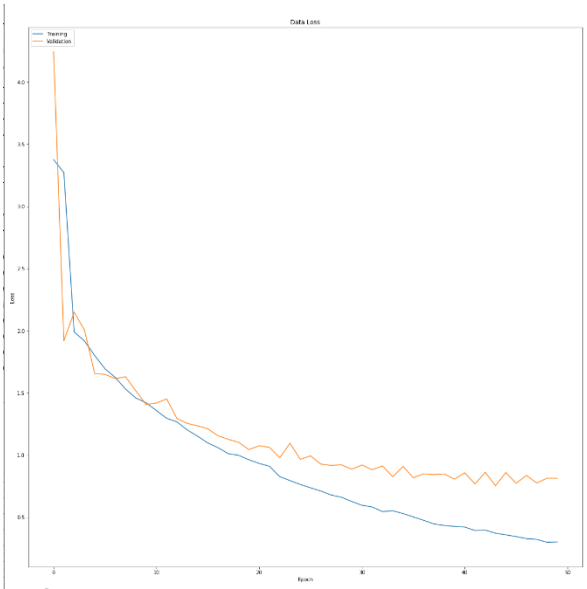

```
Test loss: 0.8059776101196021
Test accuracy: 0.7280701994895935
```

With dropout, 1 hidden layer



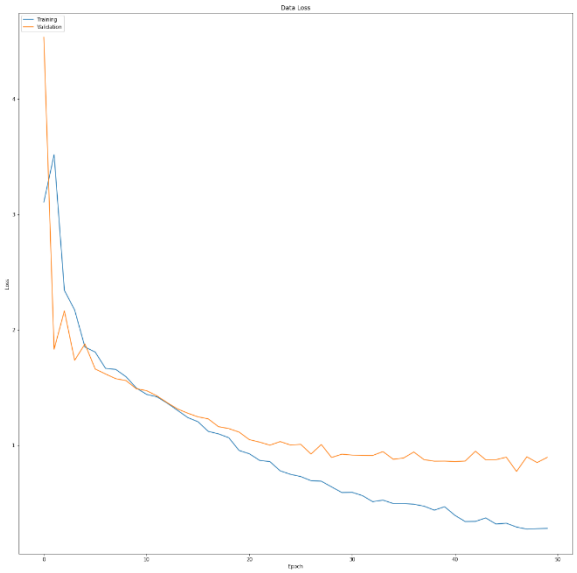

```
Test loss: 0.8560045612485785
Test accuracy: 0.7017543911933899
```

Without dropout, 2 hidden layers



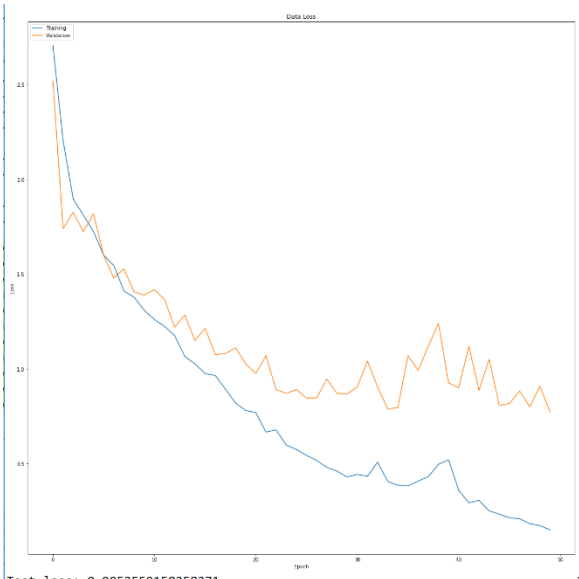

```
Test loss: 0.8467752734820048
Test accuracy: 0.719298243522644
```

With dropout, 2 hidden layers



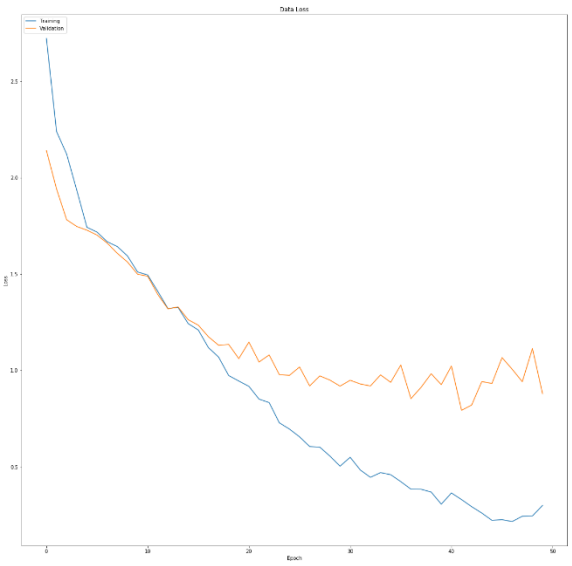

```
Test loss: 0.8022190478810093
Test accuracy: 0.780701756477356
```

Without dropout, 3 hidden layers




```
Test loss: 0.9052559158258271
Test accuracy: 0.7456140518188477
```

With dropout, 3 hidden layers




```
Test loss: 0.9628729537913674
Test accuracy: 0.7017543911933899
```

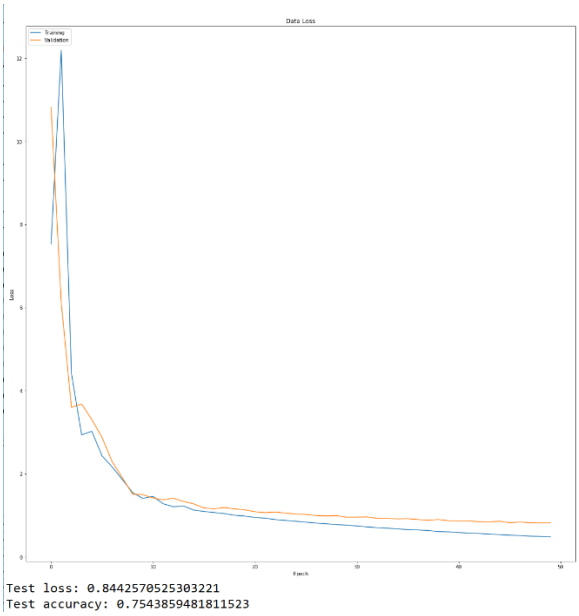
From the form and plot, we find:

1. Dropout does not affect the test loss and accuracy a lot. The loss and accuracy change in the interval and are unstable. We cannot find the value range changes a lot.
2. 3 hidden layers is best. As the number of hidden layers increase, the performances become better. For the diagram of training loss and validation loss, when we add more hidden layers, the difference between training loss and validation loss becomes larger, since the validation go stop at around 1 while the training loss continue to decrease and approach to 0.

Comparison in nodes in the hidden layer

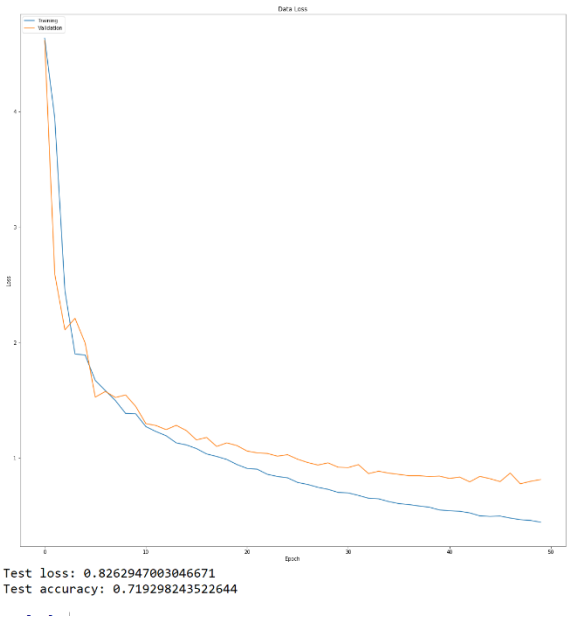
Num of nodes →	256	512	1024	2048
Loss	0.83	0.80	0.84	0.86
Accuracy	0.72	0.73	0.75	0.72

Without dropout, 1 hidden layer, 1024



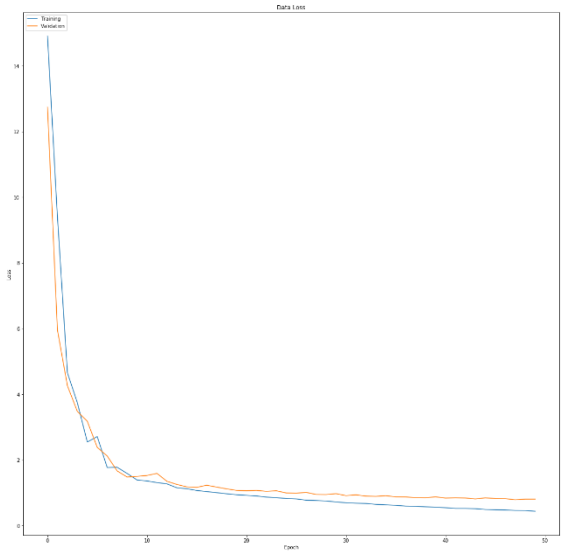

```
Test loss: 0.8442570525303221
Test accuracy: 0.7543859481811523
```

256



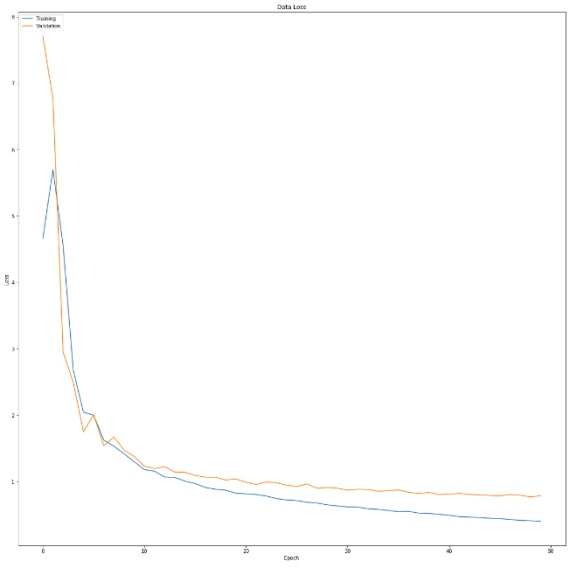

```
Test loss: 0.8262947003046671
Test accuracy: 0.719298243522644
```

2048




```
Test loss: 0.8569353264674806
Test accuracy: 0.719298243522644
```

512




```
Test loss: 0.8059776101196021
Test accuracy: 0.7280701994895935
```

From the form and plot, we find:

256 nodes is the best. When we decrease the number of nodes, the plot goes more like to best model since the difference between training loss and validation loss becomes larger. And the training loss will decrease when we decrease the number of nodes from 2048 to 256.

As a result:
The best model we find is：
1. 3 hidden layers
2. 256 nodes in each hidden layer
3. Dropout is trivial

Compare to the other result, this model has similar test loss and accuracy, but has better curve of (lower) training loss. The plot of training loss and validation loss is closer to the best model plot. (Training loss decreases over time, while validation loss saturates but not increases)

Part 5

- **Include your visualizations. Label your visualizations with the name of the actor, and display any other relevant information.**
- **Explain how you selected the hidden nodes**
- **Give a brief description (2-3 sentences) of any insights provided by the visualisations**

1. Figure 1 is Lorraine Bracco

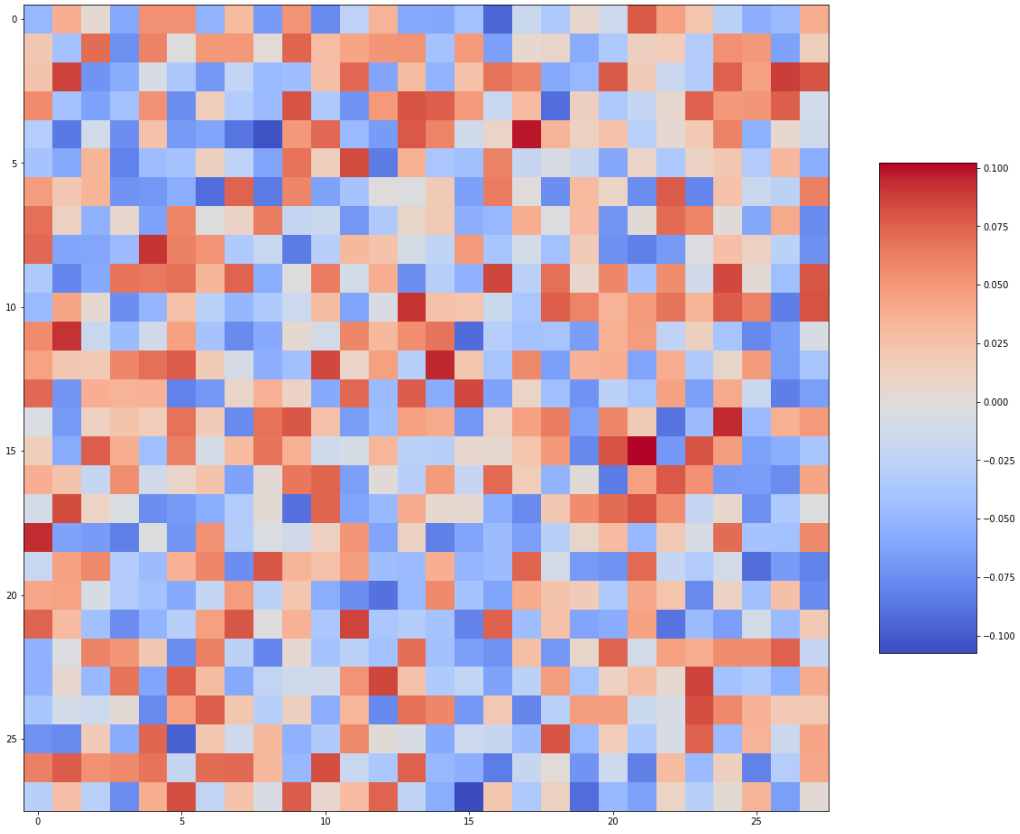
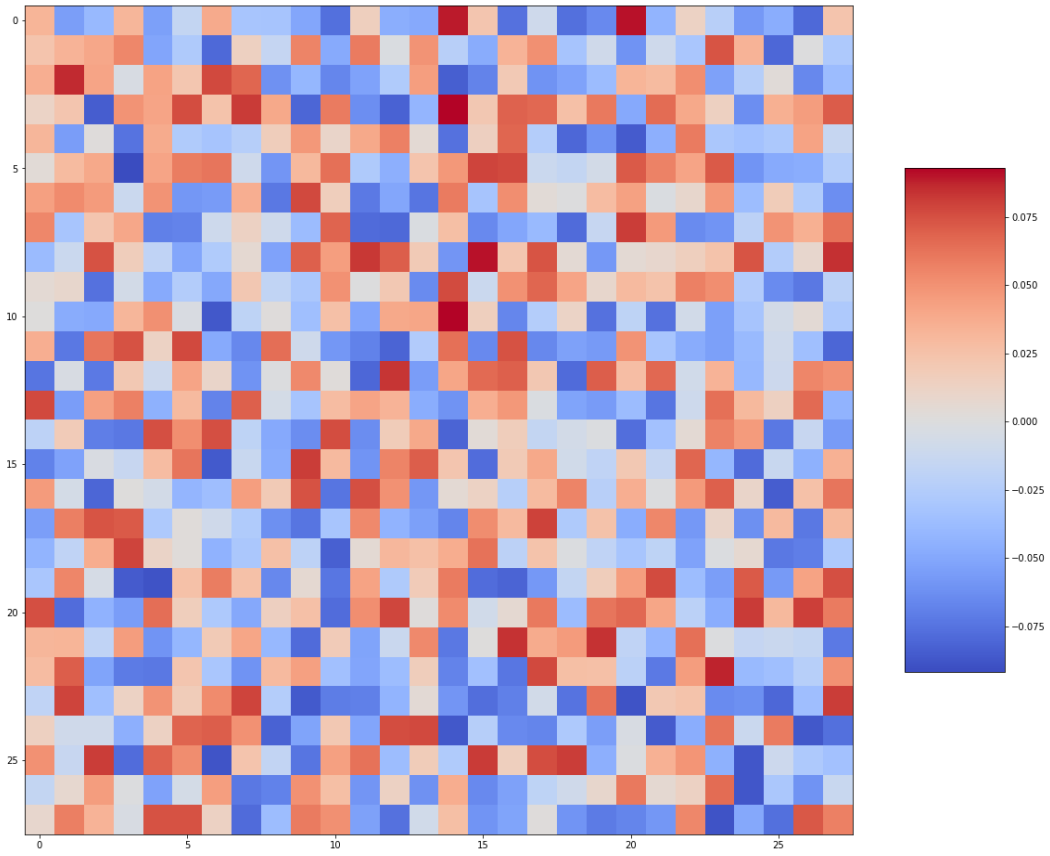


Figure 2 is Peri Gilpin



2. I choose the first and second neurons from the last list of weights. And the weights of connection between the hidden layer and output layer is the third list of weights. The shape of the list is (768, 6). Then I pick the first column and second column to save in the list and show the plot.

3. I can't get any useful information since the weights data is too abstract and it represent how important for each neurons in the hidden layer.