

Contents

- [Delivery report Pt 2](#)
- [Deliverable 3ci](#)
- [Deliverable 3cii](#)
- [Part 3 Training a machine to understand emotion \(The Sweep\)](#)
- [Part 4 Methods to reduce complexity](#)
- [Part 5 Sweep ROC](#)

```
clc; clear all; close all;
```

Delivery report Pt 2

This report contains the answers to questions posed in the Deliverable

Deliverable 3ci

```
% The number of neurons that worked better for step 3 was 400 for a sample
% size of 9000. This number of neurons worked better because increasing the
% number of neurons showed a decrease in accuracy and favorable readings
% from the ROC (Receiver Operating Characteristic). This is most commonly
% because of overfitting to the training set of the data.
```

Deliverable 3cii

```
%The extraction method that worked better was the one with less inputs and
%more information. The extraction of the frequency magnitude components
%showed an increased accuracy at different levels of neurons in the sweep.
%This was because the features that were fed to the input had more
%information in them for the neural network.
```

Part 3 Training a machine to understand emotion (The Sweep)

```
%Let it be known that the whole training dataset was not used in training
%this neural network. It was modified to only take 15,000 training samples
%and use 9,000 of it to sweep for the number of hidden neurons.

image1 = imresize(imread('./Final/100.jpg'), 0.5);
image2 = imresize(imread('./Final/200.jpg'), 0.5);
image3 = imresize(imread('./Final/300.jpg'), 0.5);
image4 = imresize(imread('./Final/400.jpg'), 0.5);
image5 = imresize(imread('./Final/500.jpg'), 0.5);
image6 = imresize(imread('./Final/600.jpg'), 0.5);
image7 = imresize(imread('./Final/700.jpg'), 0.5);
image8 = imresize(imread('./Final/800.jpg'), 0.5);
image9 = imresize(imread('./Final/900.jpg'), 0.5);
image10 = imresize(imread('./Final/1000.jpg'), 0.5);
final = imread('./Final/1000.jpg');
plot = [image1 image2 image3 image4 image5; image6 image7 image8 image9 image10];
figure; imshow(plot); title('Accuracy through each iteration');
figure; imshow(final); title('Final System Accuracy');

%Below shows each sweep iteration from 100 to 1000, incrementing by 100
%each time. The fluctuation that we are seeing is due to
%the fact that the neural network is being trained with a different number
%of neurons each iteration. As can be seen, the accuracy per neurons
%decreases until it reaches 400 where it spikes to a 25% accuracy and then
%continues to decrease until the final iteration. This means that training
%the neural network with 400 neurons would give us the most accurate
%outputs. The final system accuracy is shown below also. As you can see,
%400 neurons is the most accurate in our sweep.

image_p = imread('./Final/percentage.jpg');
figure; imshow(image_p); title('Accuracy after percentage change');

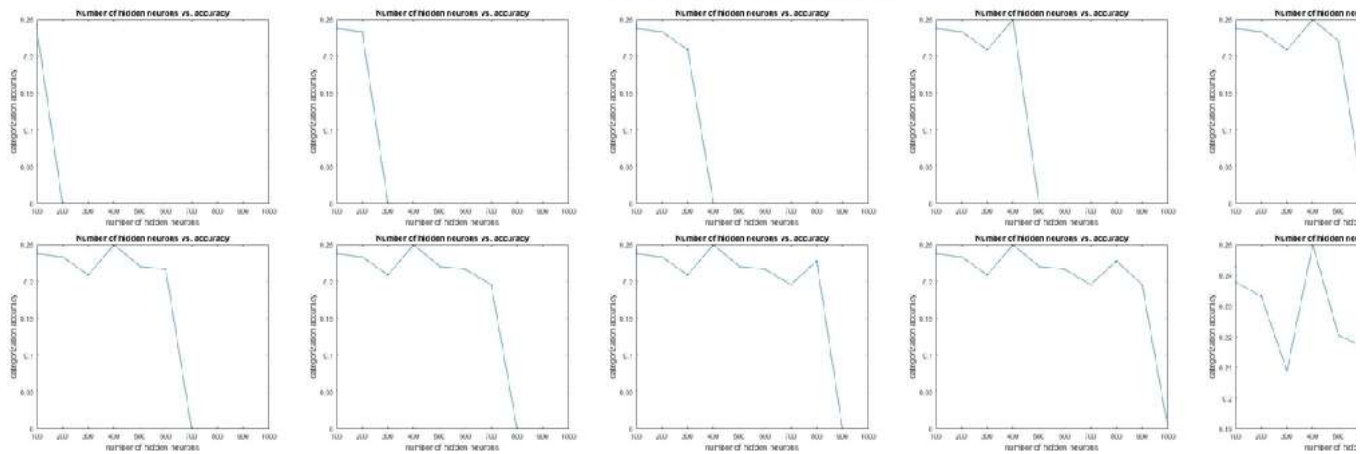
%We also manipulated the percentage of data going to training and testing
%in the sweep iteration for loop in the hopes that we would correct
%overfitting more. However, the results yielded less accuracy as seen
%below.

image1_n = imresize(imread('./Final/100 neurons.jpg'), 0.5);
image2_n = imresize(imread('./Final/200 neurons.jpg'), 0.5);
image3_n = imresize(imread('./Final/300 neurons.jpg'), 0.5);
image4_n = imresize(imread('./Final/400 neurons.jpg'), 0.5);
image5_n = imresize(imread('./Final/500 neurons.jpg'), 0.5);
image6_n = imresize(imread('./Final/600 neurons.jpg'), 0.5);
image7_n = imresize(imread('./Final/700 neurons.jpg'), 0.5);
image8_n = imresize(imread('./Final/800 neurons.jpg'), 0.5);
image9_n = imresize(imread('./Final/900 neurons.jpg'), 0.5);
image10_n = imresize(imread('./Final/1000 neurons.jpg'), 0.5);
final_n = imread('./Final/ROC_sweep.jpg');
figure; imshow(image1_n);
figure; imshow(image2_n);
```

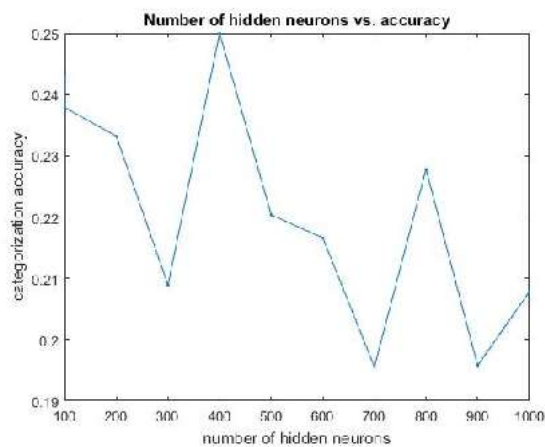
```
figure; imshow(image3_n);
figure; imshow(image4_n);
figure; imshow(image5_n);
figure; imshow(image6_n);
figure; imshow(image7_n);
figure; imshow(image8_n);
figure; imshow(image9_n);
figure; imshow(image10_n);
figure; imshow(final_n);
```

```
%Each of the ROC curves shown below represent the performance of the neural
%network when being trained with the specified number of hidden neurons.
%Each emotion is represented by a class, as can be seen on each graph:
%(7=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)
%What we are looking for is the ROC curve with the most classes
%located primarily in the upper left hand quadrant of the plot. This will
%indicate that the neural networks performance, with respect to each
%class, is good. It can be seen that the plot utilizing 400 hidden neurons
%shows the best ROC curve.
%The final ROC curve is the overall ROC curve for the neural network.
```

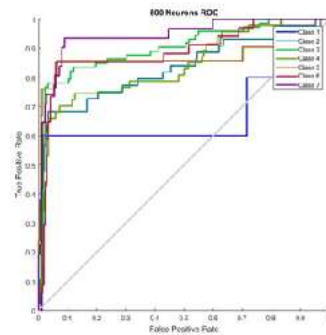
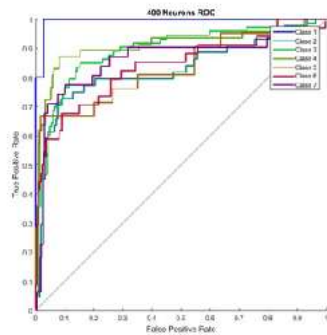
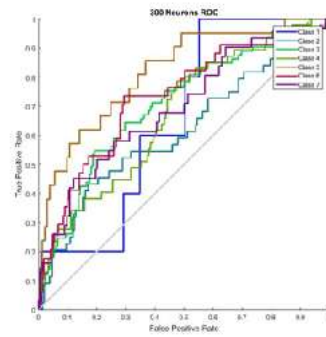
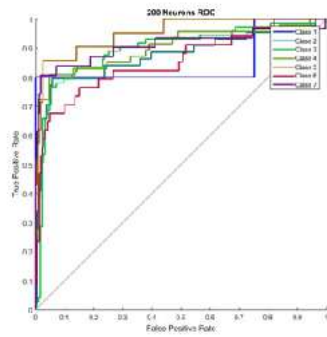
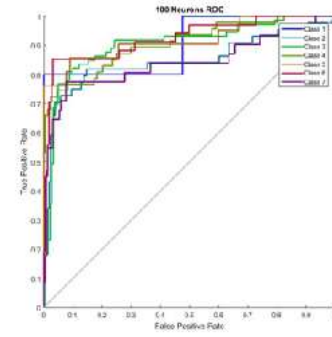
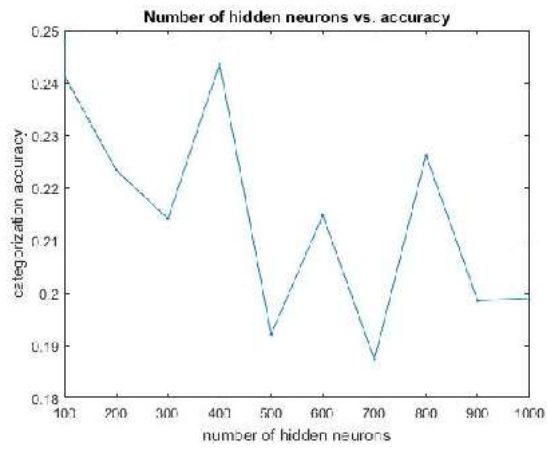
Accuracy through each iteration

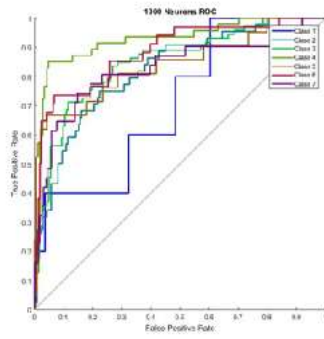
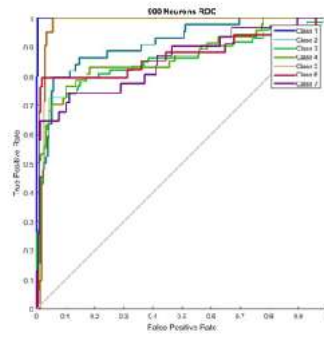
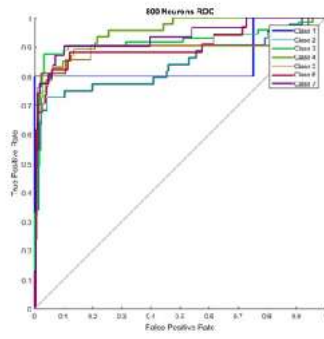
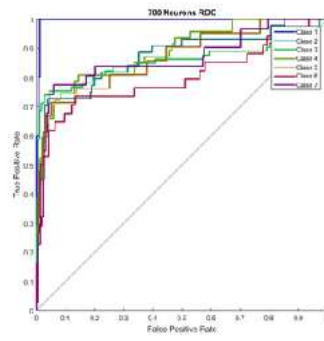
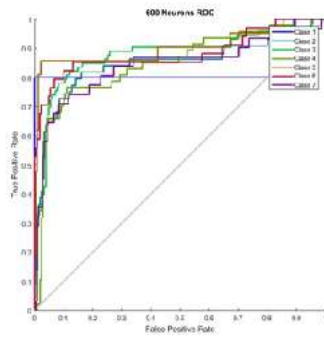


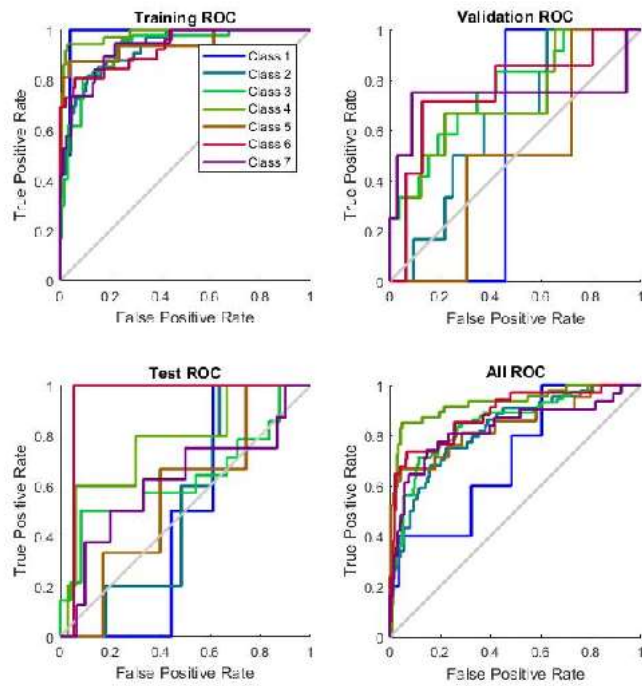
Final System Accuracy



Accuracy after percentage change







Part 4 Methods to reduce complexity

```
%Three different methods that could be used to reduce the complexity of the
%system are:
%1. wavelet: using wavelet transform would take out a lot of the noise in
%the images, allowing the neural network to grab better pixel values from
%the data sets.
%2. frequency domain: by taking the frequency domain, we could take lower
%values of an image and create a more precise data set for training and
%testing
%3. downsize image: downsizing an image would decrease the size of the
%dataset which could decrease the chance of overfitting the training set.
%This could reduce the complexity of the training set and increase the
%accuracy.

%All three of the above methods would manipulate the input images and
%create a more precise training and testing set for the neural network to
%be trained with. This would create a better performing and more accurate
%neural network.
```

Part 5 Sweep ROC

```
sweep = [10,10:10:250];

for i = 1:21
    formatSpec = './Q5figSaves/N%dRoc';
    savefigpath = sprintf(formatSpec,sweep(i));
    openfig(savefigpath);

end
% close all
```

