**Final Project Report**

**INFO 7390 Advances in Data Science and Architecture**

**Fall 2017**

**Team 2**

**Soumiya Roy (NUID – 001220681)**

**Bharat Parwani (NUID – 001222816)**

**Yuanyuan Zhou (NUID – 001233834)**

**Jiamin Wang (NUID – 001289959)**

**Final Project Report**

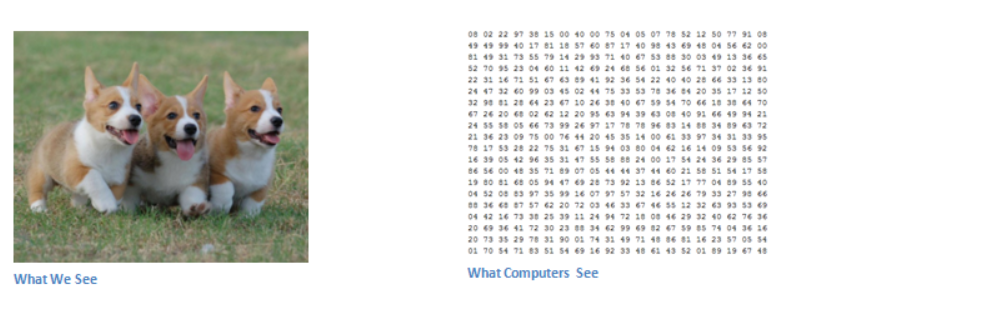
**Advances in Data Science/Architecture**

**Convolutional Neural Networks –**

1. Convolutional Neural Networks (ConvNets or CNNs) are the category of Neural Networks that are very effective in areas such as image recognition and classification.
2. CNN’s have been very successful in identifying the faces, objects and trafiic signs along with powering vision in robots and self-driving cars.

**CNN’s for Image Classification –**

Image classification is the task of taking an input image and outputting a class (a cat, a dog, etc) or a probability of classes that best describes the image. For humans, this task of recognition is one of the qualities we have as our ability to identify things with our eyes and which comes naturally and effortlessly as adults. When we see an image or just when we look at the world around us, most of the time we are able to immediately characterize the scene and give each object a label, all without even consciously noticing. These skills of being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are ones that we do not share with our fellow machines.



**Inputs and Outputs for Convolutional Neural Network –**

When a computer sees an image (takes an image as input), it will see an array of pixel values. Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB which refers to the red, green and blue values). Just to give a clear explanation, let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer.  The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (.80 for cat, .15 for dog, .05 for bird, etc).

**Working Structure –**

A more detailed overview of what CNNs do would be that you take the image, pass it through a series of convolutional, nonlinear, pooling (downsampling), and fully connected layers, and get an output. As we said earlier, the output can be a single class or a probability of classes that best describes the image. Now, the hard part is understanding what each of these layers do.

**Difference between CNN and NN –**

**Neural Network –**

1. Uses the entire image to train the network
2. Works well for simple centered image (for example a centered handwritten digit image) but fails to recognize image with more complex variation (for example a running cat in backyard)
3. Having more hidden layers to learn abstract features would help but impractical as we need far too many neurons to train and store in memory.

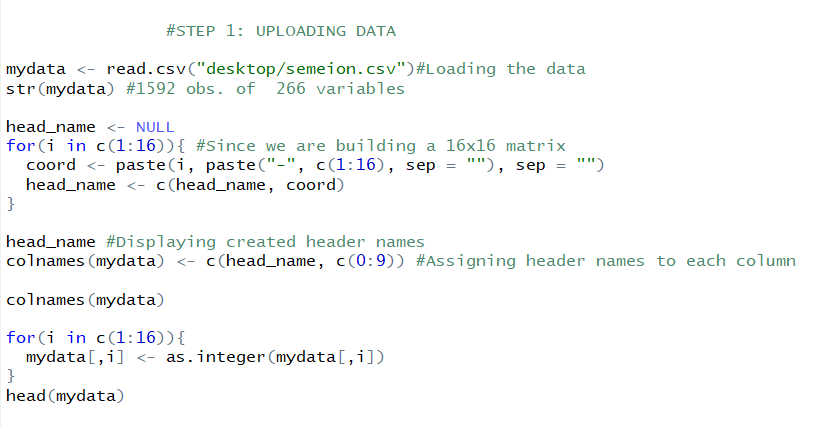
**Convolutional Neural Network –**

1. Recognizes an object by first looking for low level features such as edges, lines and curves, and then building up more abstract features through series of convolutional layers.
2. Network learns individual parts of an object by shared filters on small regions of the image, and add them up to build abstract features. The use of shared filters largely reduce the actual parameter learning.
3. Generalizes better for complex image recognition as learned filters can be reused to detect abstract features compositionally through convolutional layers.

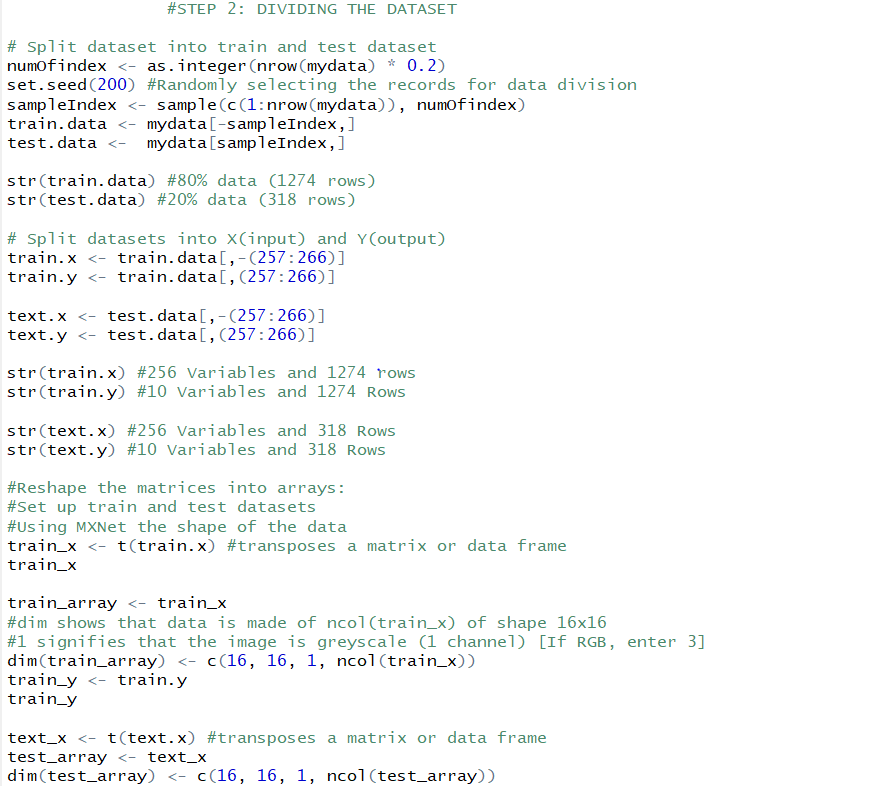
**Our approach of working on the project –**

We used R packages “mxnet”, “mlbench”, “lattice” “caret” and “ggplot2” for implementation of this project.

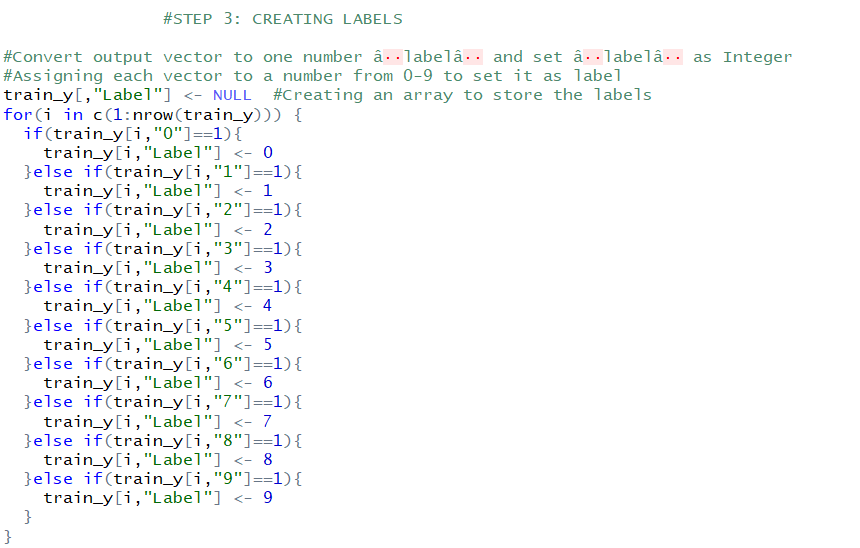
Step 1. Loading the data

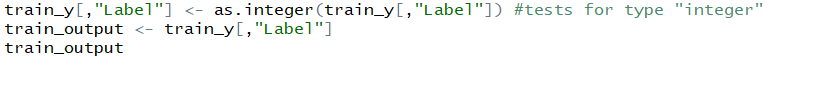


Step 2. Dividing the Dataset



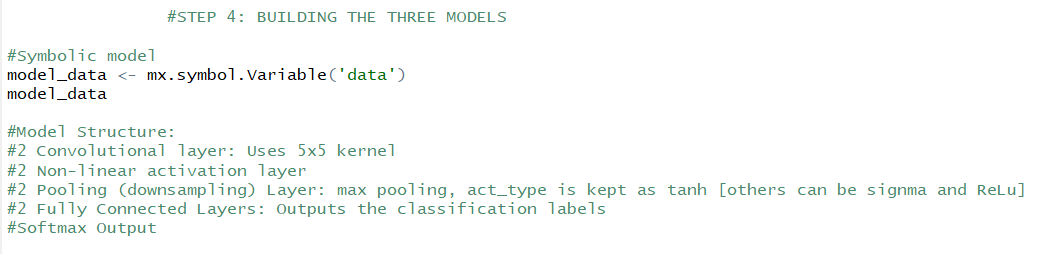
Step 3. Creating the labels



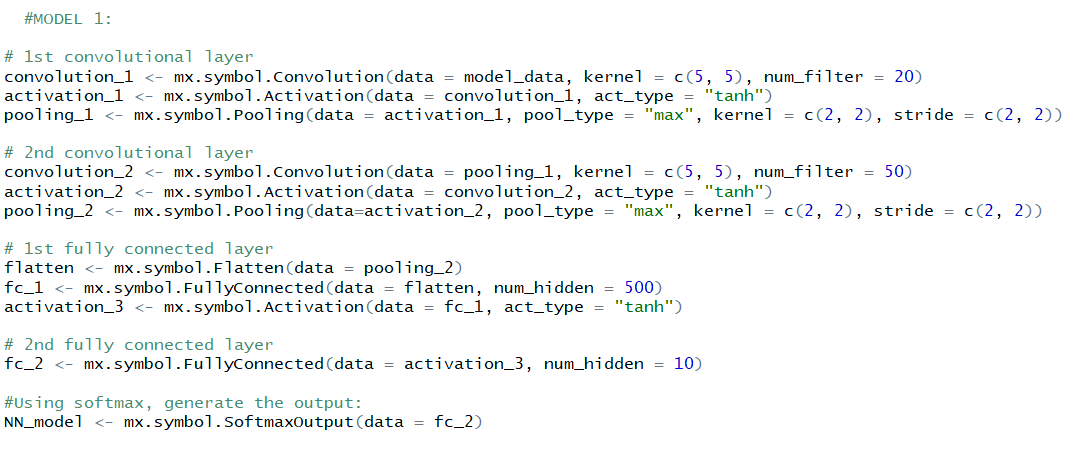




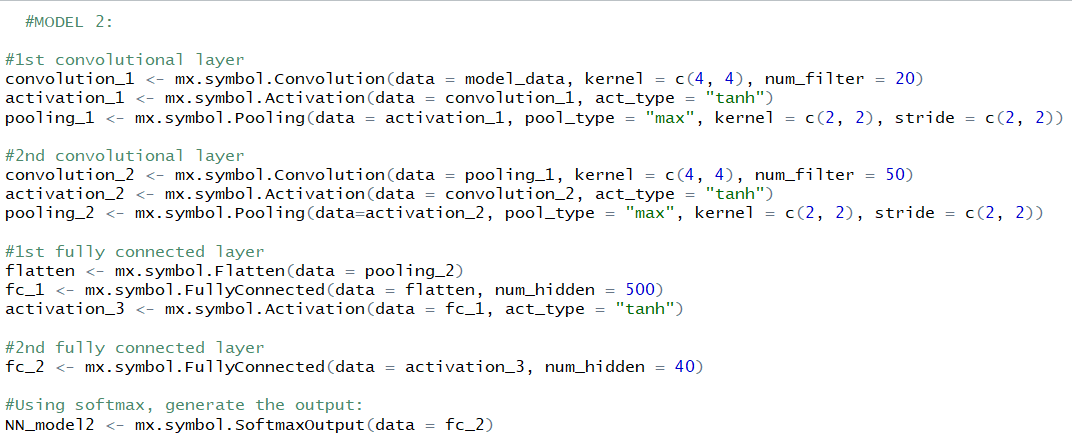
Step 4. Building the Models



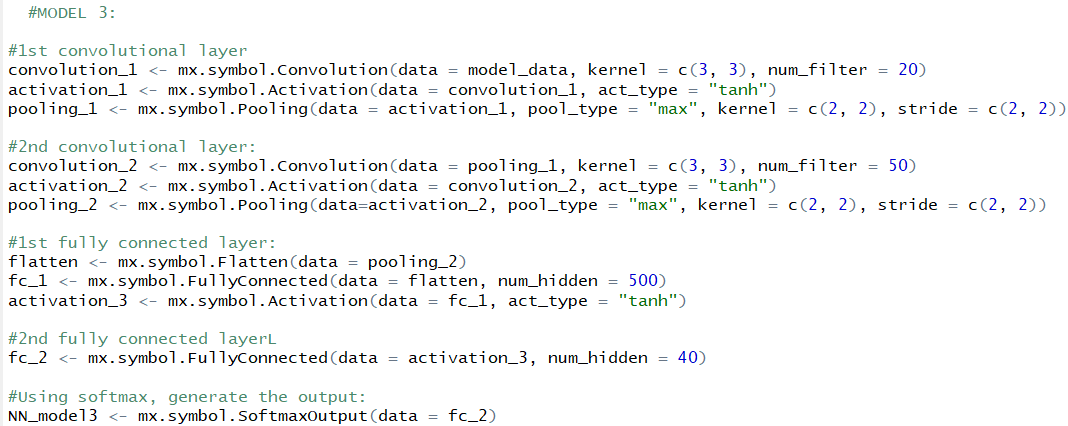
Step 4(A). Model 1



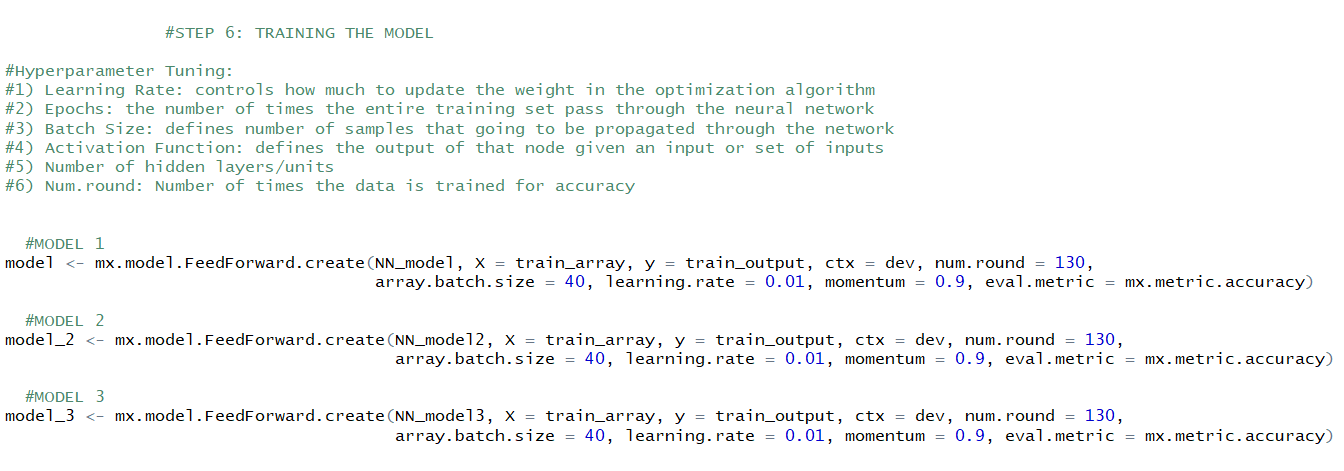
Step 4(B). Model 2



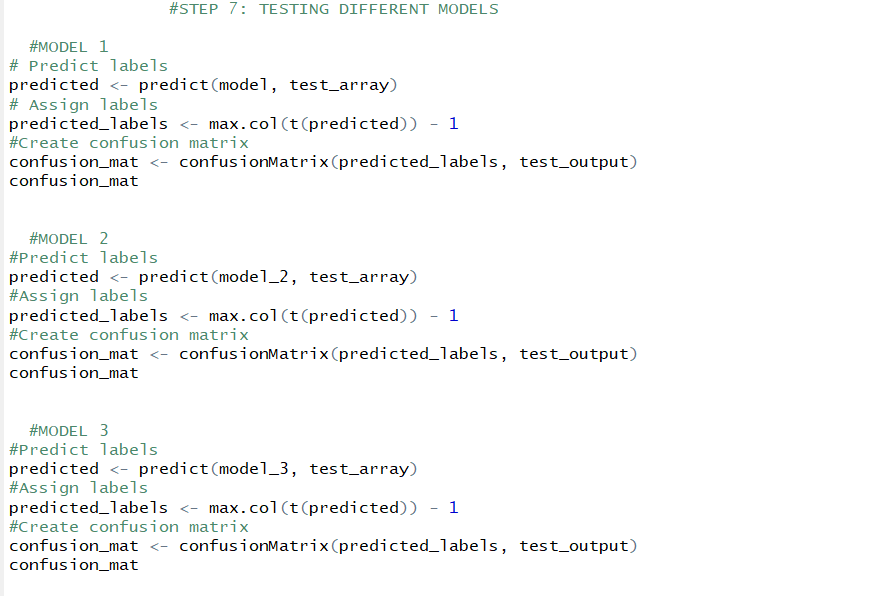
Step 4(C). Model 3



Step 5. Training the Model



Step 6. Testing the Model



Recommendation/Future Work –

1. Gather handwritten samples from different demographics and age groups to analyze the pattern
2. Based on the mass handwritten samples, find out the emotional sentiments of the individual at the time of sample gathering
3. Apply facial recognition and image processing techniques on politicians and participants to decide the outcome based on artificial intelligence



PPT Link:

https://docs.google.com/presentation/d/1Zcls6ZIjsW-b3PC5iR7MGUEhpWfIxk2759XfrTCfsuk/edit#slide=id.p