**Project Report**

**Objective**:

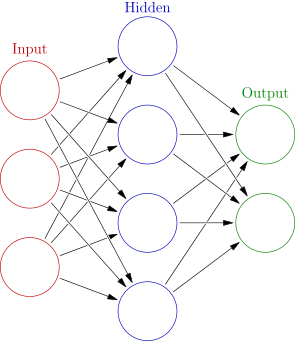
The objectives of the project are:

1. Running Neural Networks on the Prudential Insurance Data Set.
2. Fit Convolutional Neural Network (CNN) to Semeion Handwritten Digit Data by finding the optimum number of hidden layers and related nodes.

**Part 1: Neural Networks**

Neural Network is a Non-linear model characterized by an activation function, which is used by interconnected information processing units to transform input into output.

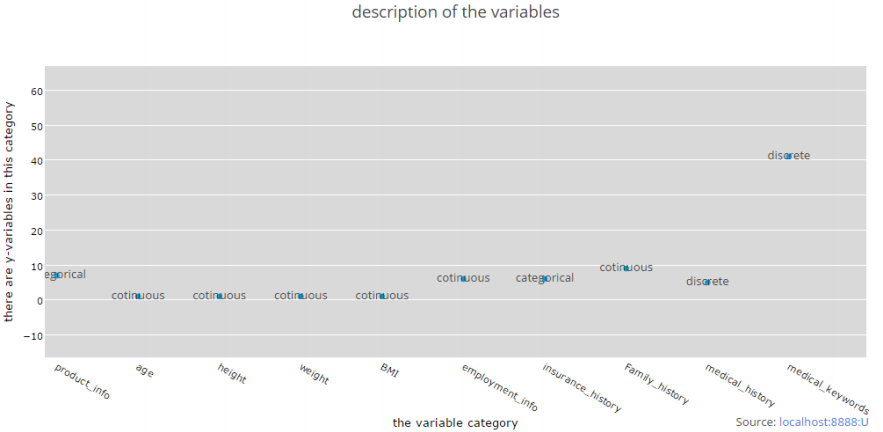
**The layers**: The input layer connects with hidden layer/s, which in turn connects to the output layer.



**Prudential Dataset:**

1. The data contains 110 categorical variables, 13 continuous variables and 5 discrete variables
2. Categorical variables contain a finite number of categories, such as (Product\_Info\_2)
3. Discrete variables are numeric variables that have a countable number of values between two values, such as (Medical\_History\_10)
4. Continuous variables are numeric variables that have an infinite number of values between two values, such as (BMI)

**Predictive Variables:**



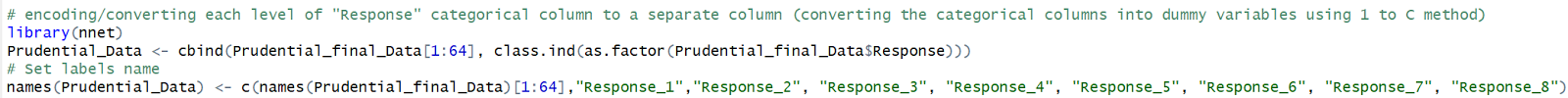
**Our Approach:**

**Data Presentation and Transformation:**

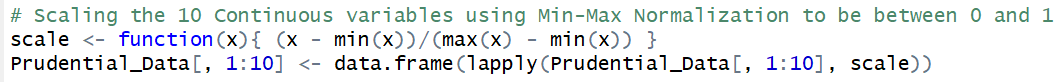
1. Removing the columns that have more than or equal 70% of null values; and removing unnecessary Id column
2. Replacing the null values in each column with the computed mean of that column
3. Setting categorical columns to as a factor to get factors with n levels
4. Converting the categorical columns into dummy variables using 1 to C method
5. Fitting the linear regression model and taking the significant variables only

**Model 1: Neural Network Multinomial Classification**

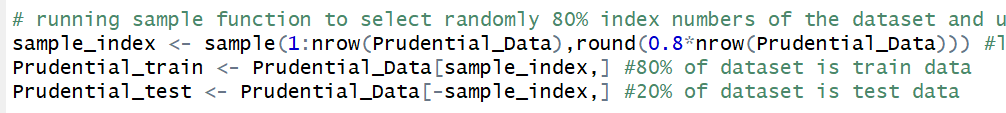
1. Converting each level of "Response" categorical column into dummy variables using 1 to C method.



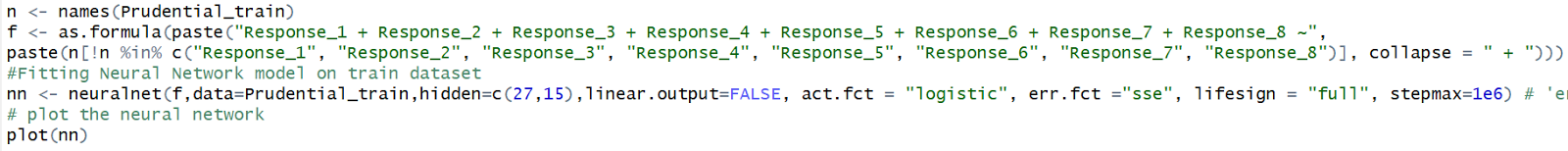
1. Scaling the 10 Continuous variables using Min-Max Normalization to be between 0 and 1



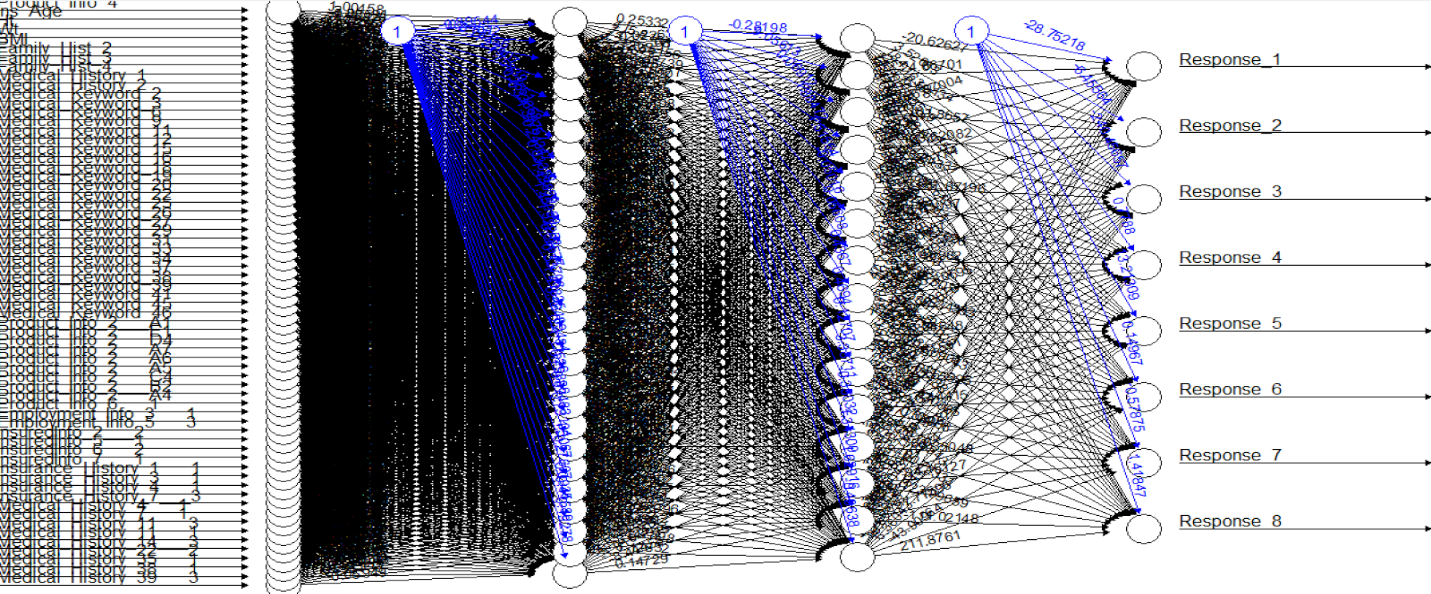
1. Splitting dataset into train and test datasets using sample function



1. Fitting Neural Network on train dataset by choosing the hidden layer to be ⅔ of the input size.



**Plotting the Result:**

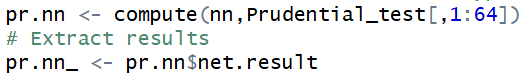


**Model 1**:

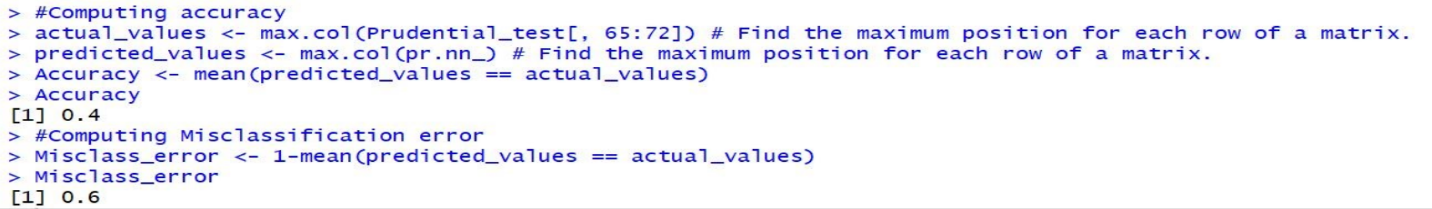
1. Computing predictions for multi-class classification:

The typical approach is to have ‘n’ output neurons represent the different classes

In the end, the neuron which has the highest prediction 'wins' and that class is predicted



1. Computing the accuracy and the misclassification error

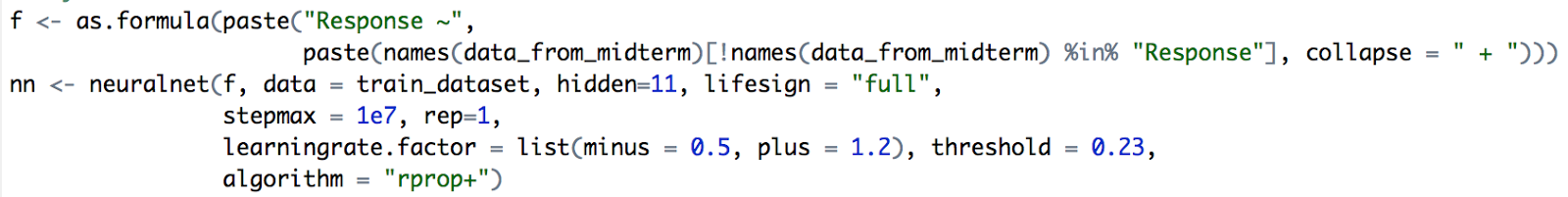


**Hidden Layers:**

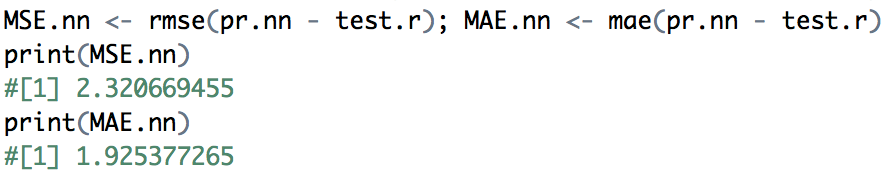
|  |  |  |
| --- | --- | --- |
| **Hidden Layers** | (27,15) | (23,19) |
| **Accuracy** | 40% | 41% |
| **Misclassification Error** | 60% | 59% |

**Model 2: Neural Network Regression**

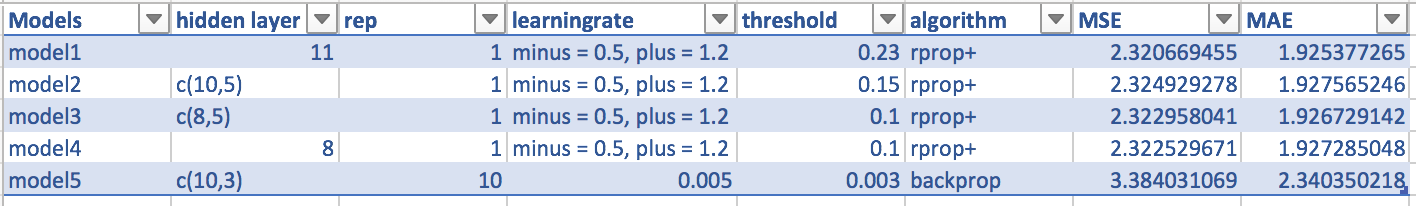
1. Scale the dataset and split it into train and test data
2. Set parameters and train the train dataset



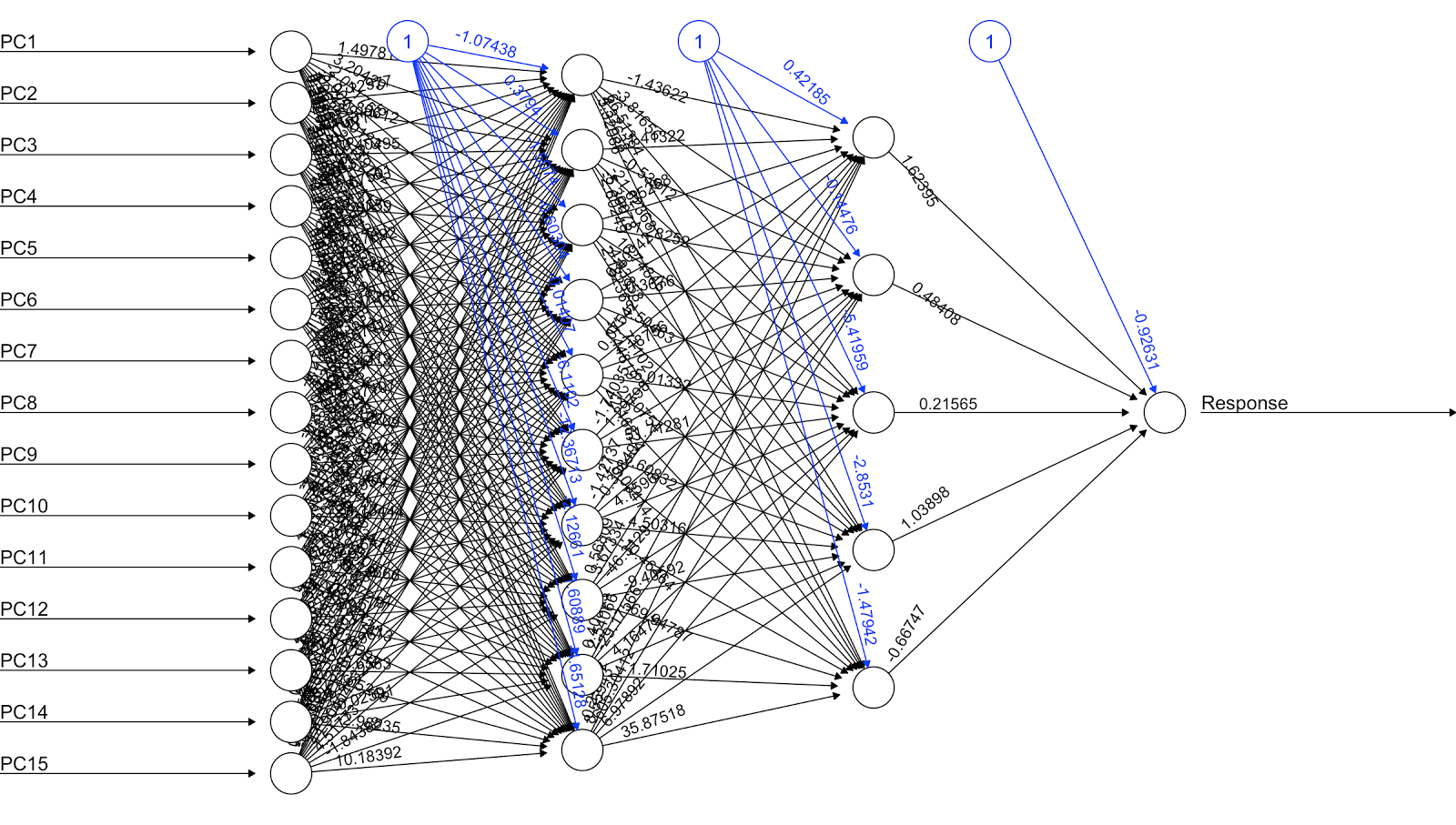
1. Evaluate the performance



**Models and Summary:**

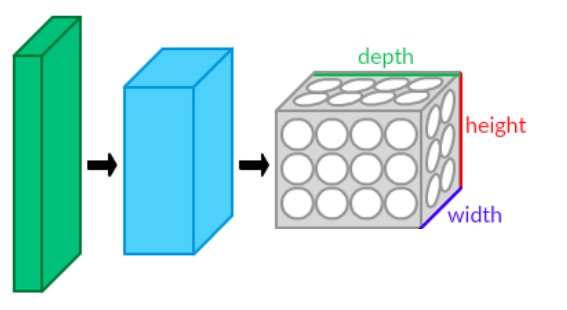


**Demo Plot Output:**



**Part 2: Convolutional Neural Network**

1. Convolutional neural network (CNN, or ConvNet) is a class of deep, [feed-forward](https://en.wikipedia.org/wiki/Feedforward_neural_network) [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) that has successfully been applied to analyzing visual imagery.
2. They have applications in image and video recognition, recommender systems and natural language processing.
3. A CNN consists of an input and an output layer, as well as multiple [hidden layers](https://en.wikipedia.org/w/index.php?title=Hidden_layer_(neural_network)&action=edit&redlink=1). The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers.
4. Convolutional
5. Pooling
6. ReLU (Rectified Linear Units) layer
7. Fully connected layer
8. Loss layer



**Exploratory Data Analysis:**

Title: Semeion Handwritten Digit

* Title: Semeion Handwritten Digit
* Title: Semeion Handwritten Digit

1. Abstract: 1593 handwritten digits from around 80 persons were scanned, stretched in a rectangular box 16x16 in a gray scale of 256 values.
2. Data Set Characteristics: Multivariate
3. Number of Instances: 1593
4. Number of Attributes: 256
5. Associated Tasks: Classification

**Data Set Information**:

1593 handwritten digits from around 80 persons were scanned, stretched in a rectangular box 16x16 in a gray scale of 256 values. Then each pixel of each image was scaled into a boolean (1/0) value using a fixed threshold.

Each person wrote on a paper all the digits from 0 to 9, twice. The commitment was to write the digit the first time in the normal way (trying to write each digit accurately) and the second time in a fast way (with no accuracy).

The best validation protocol for this dataset seems to be a 5x2CV, 50% Tune (Train +Test) and completely blind 50% Validation.

**Attribute Information:**

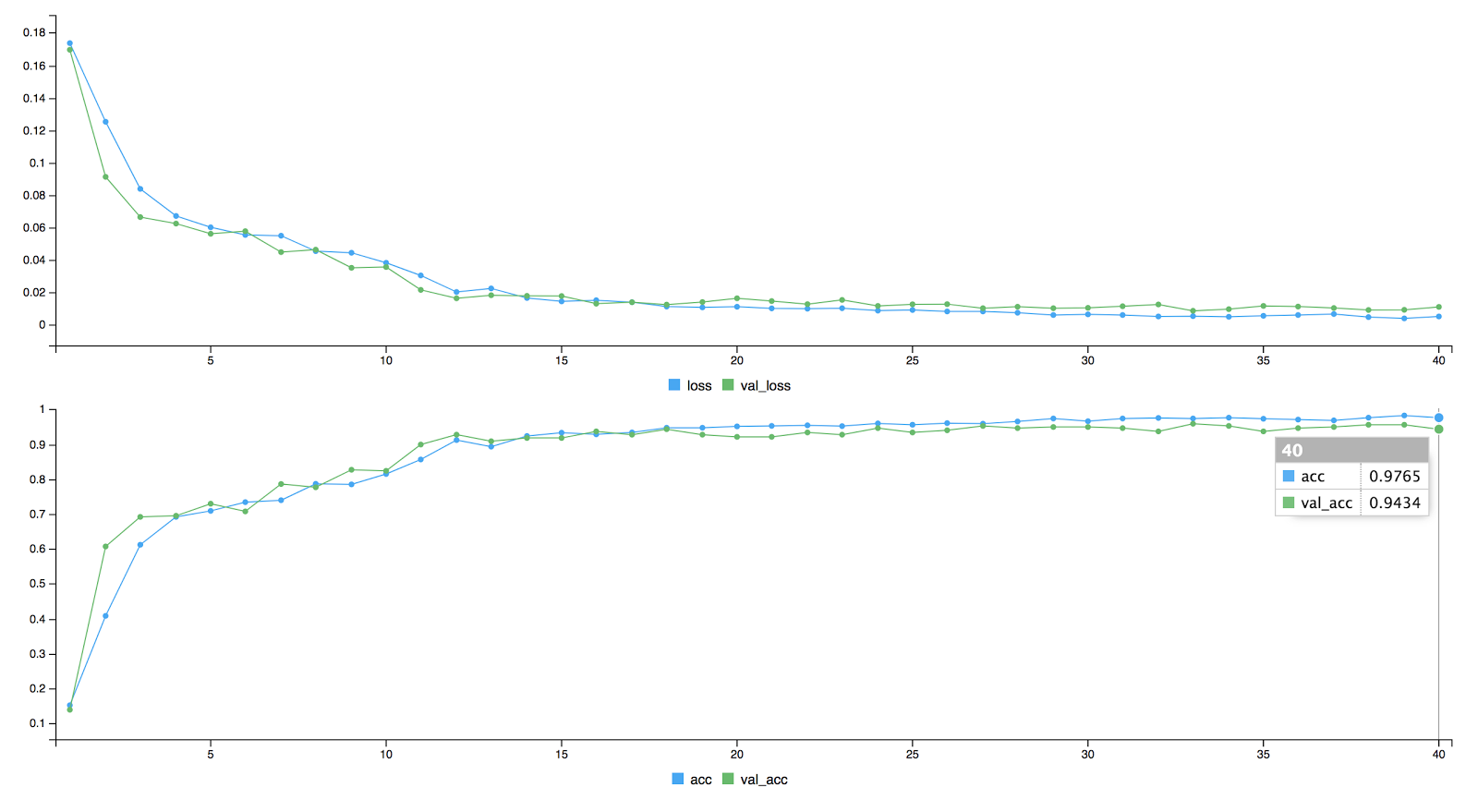
This dataset consists of 1593 records (rows) and 256 attributes (columns). Each record represents a handwritten digit, originally scanned with a resolution of 256 grays scale (28). Each pixel of the each original scanned image was first stretched, and after scaled between 0 and 1 (setting to 0 every pixel whose value was under the value 127 of the grey scale (127 included) and setting to 1 each pixel whose ordinal value in the grey scale was over 127). Finally, each binary image was scaled again into a 16x16 square box (the final 256 binary attributes).

**Process and Code:**

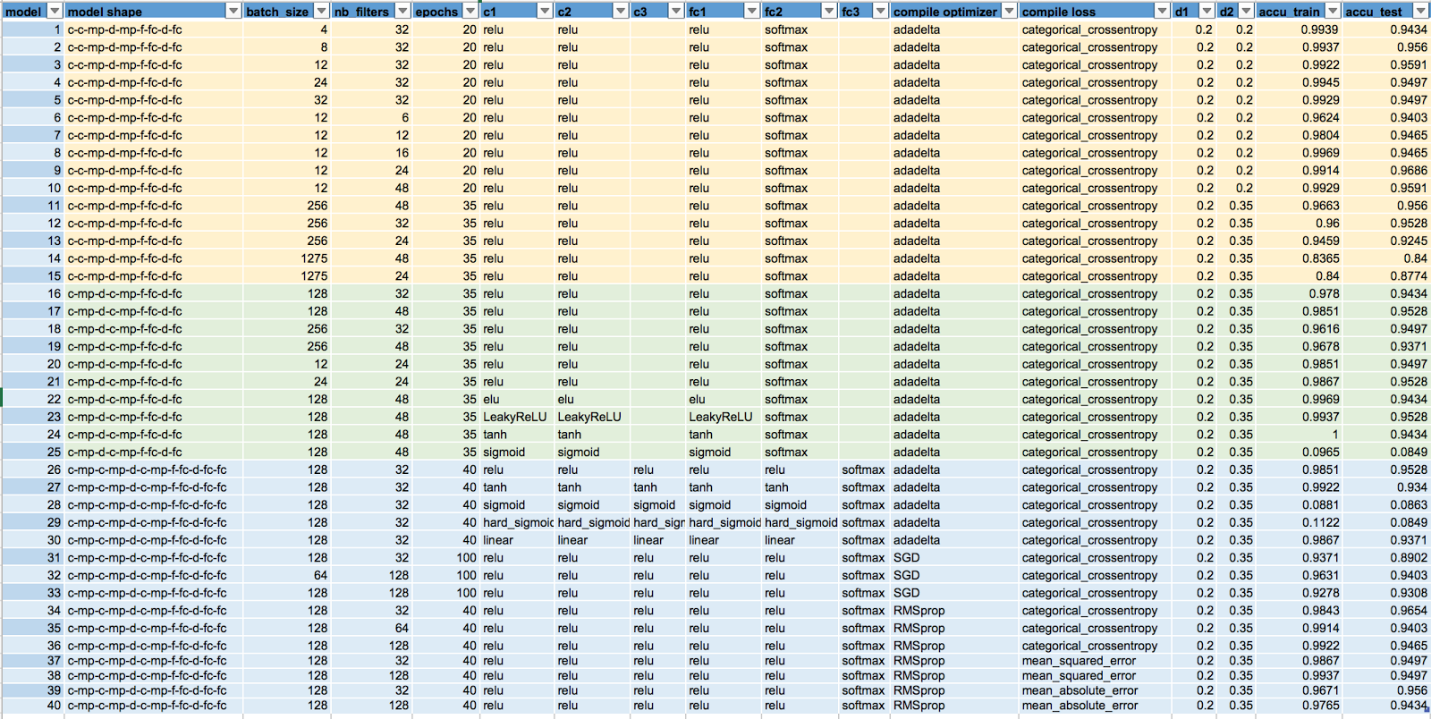
1. Setup keras environment in R
2. Read semeion dataset
3. Split into train and test dataset
4. Add CNN layers



**Sample Output:**



**Models and Summary:**



**Conclusion and Future Work:**

**NEURAL NETWORK:**

|  |  |
| --- | --- |
| **Multinomial Classification** | **Regression** |
| Takes longer time to process | Processes in shorter time |
| Result is straightforward | Requires additional operation |

1. Less variables reduce the runtime
2. Each dataset has different parameters and algorithms that works best

**CONVOLUTIONAL NEURAL NETWORK:**

* Data is the key

**FUTURE WORK:**

1. deepnet and mxnet Package
2. h20 package