COL774: Assignment 4

Nikhil Goyal
2015CS50287

1 Part 1: Fixed Algorithms

1.a K-Means

- Results
  - Training Accuracy = 36.905%
  - Test Accuracy = 36.607%
- Observations
  - K-Means converges after about 20 iterations. After this only slight variance occurs in train and test accuracies.
  - Clusters do not form a 1-1 mapping with labels as the data is not separable into 20 clusters in linear space.

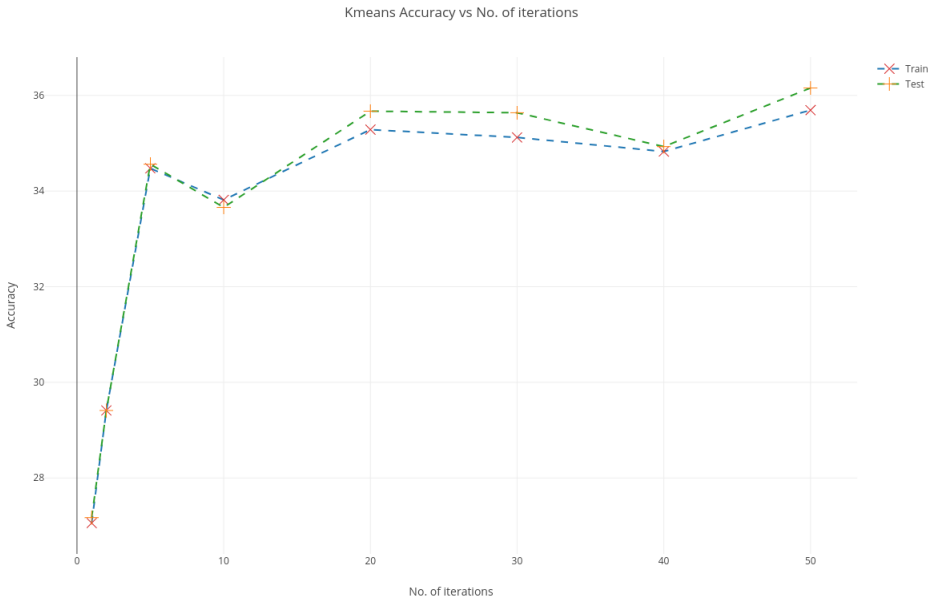


Figure 1: Accuracy vs No. of iterations (K-Means)

1.b PCA + SVM

- Best Parameters
  - C = 0.1
- Results
  - Training Accuracy = 78.180%

- Test Accuracy = 65.455%

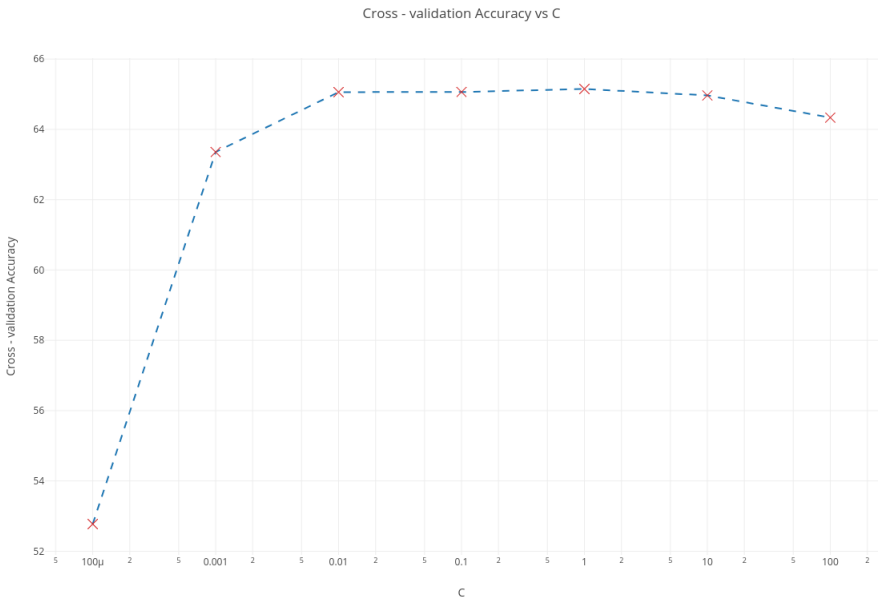


Figure 2: Cross - validation accuracy vs C value (SVM)

1.c Neural Network

- Best Parameters
  - No. of hidden units = 1000
- Results
  - Training Accuracy = 97.168%
  - Test Accuracy = 80.392%
- Observations
  - Validation accuracy increases with number of hidden units initially. After a certain limit it plateaus and starts to decrease as the neural network starts to overfit the data.

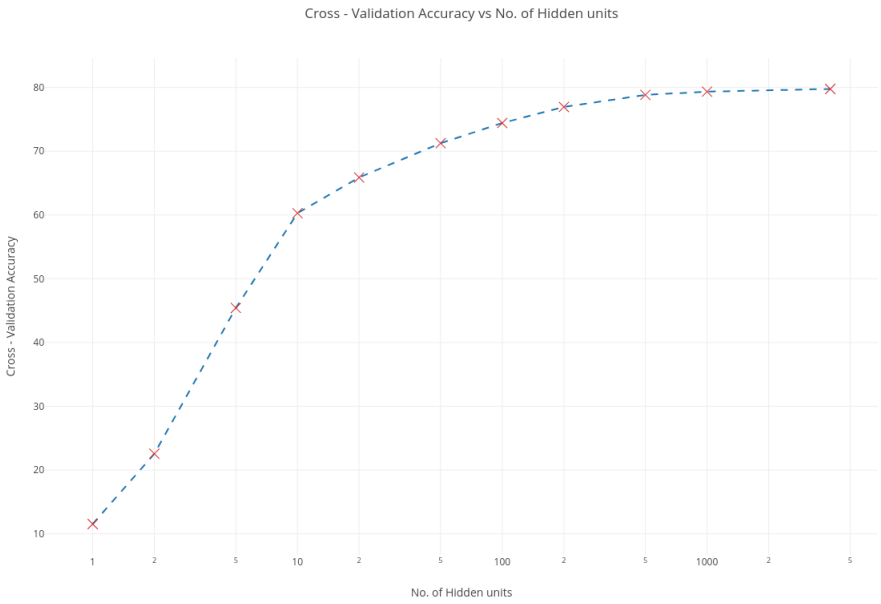


Figure 3: Cross - validation accuracy vs No. of hidden units (Neural Net)

1.d Convolutional Neural Network

- Best Parameters
  - No. of hidden units = 500
  - Kernel size = 5×5
  - No. of Kernels = 64
- Results
  - Training Accuracy = 94.772%
  - Test Accuracy = 86.692%
- Observations
  - A small kernel size of 2 captures very fine details but is unable to store information of higher level details like raw strokes which form the drawing, resulting in low validation accuracies. Also kernel size of 7 generally doesn't fare well as it ignores at lot of details.
  - Validation accuracy increases with number of hidden units initially. After a certain limit it plateaus and starts to decrease as the neural network starts to overfit the data.

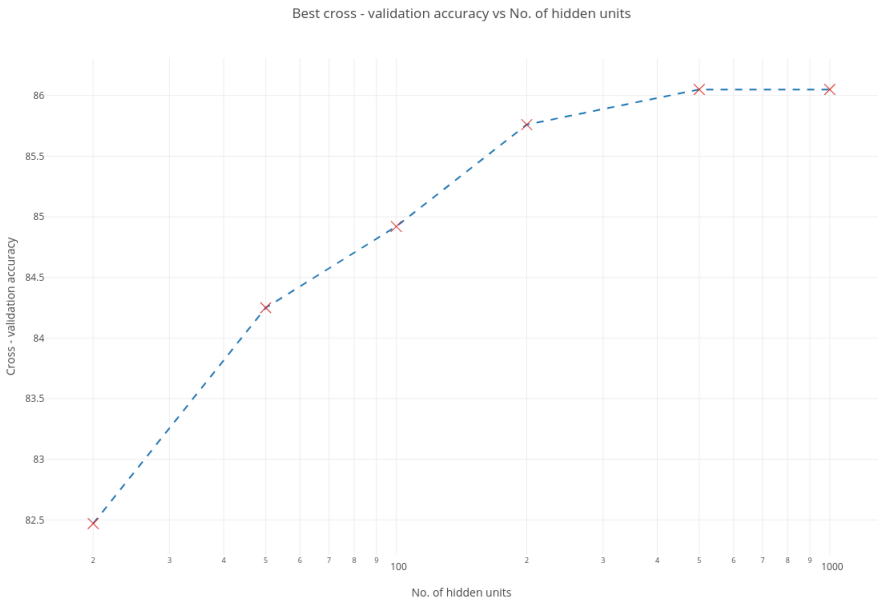
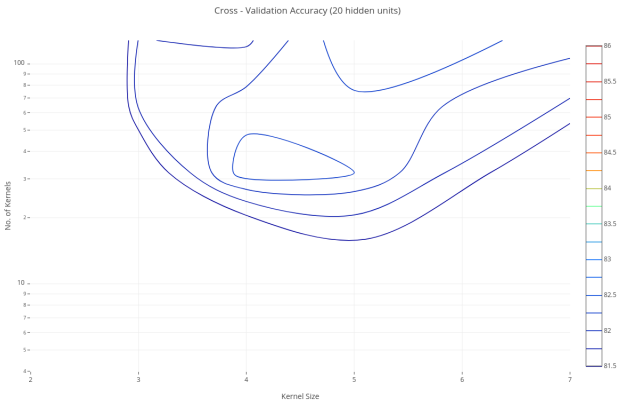
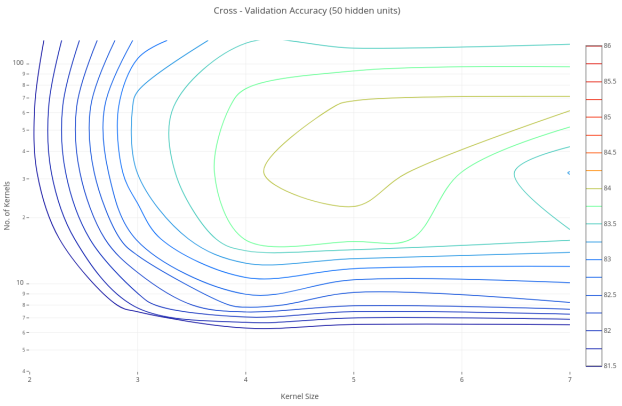


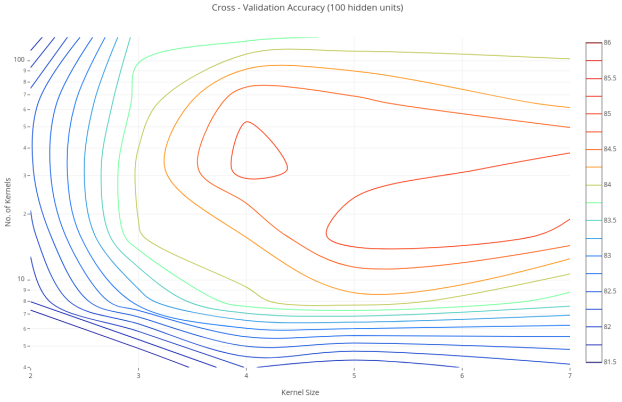
Figure 4: Best Cross - validation accuracy vs No. of hidden units (CNN))



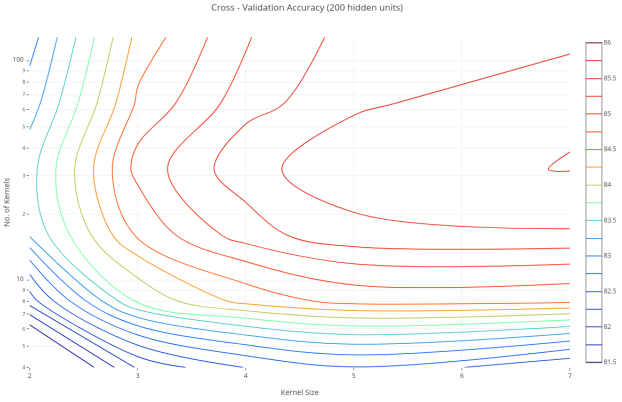
(a) 20 Hidden units



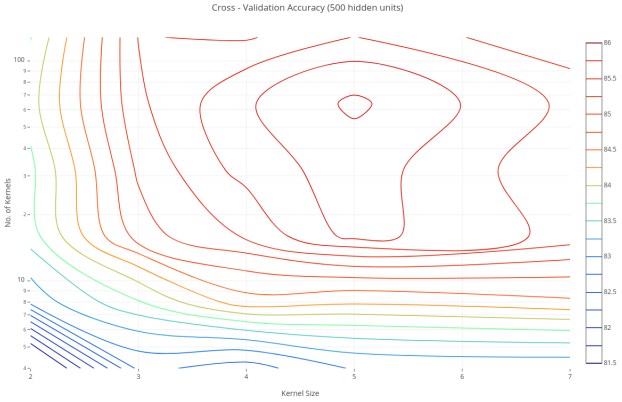
(b) 50 Hidden units



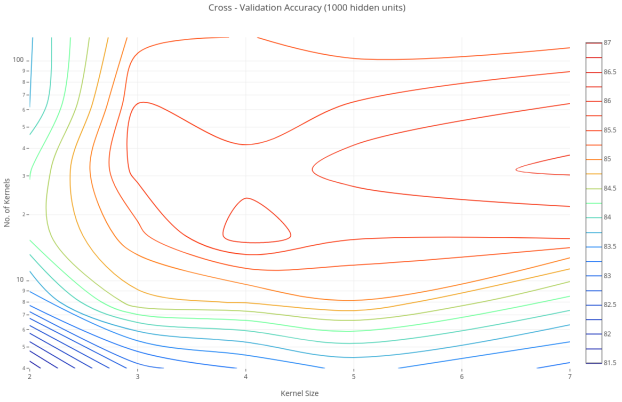
(c) 100 Hidden units



(d) 200 Hidden units



(e) 500 Hidden units



(f) 1000 Hidden units

Figure 5: Cross - validation accuracy for different Kernel sizes and no. of Kernels (CNN)

1.e Comparison

- Observations
  - K-means is an unsupervised learning algorithm which probably won't give great results for image classification. The classes it creates will likely mix similar objects like, create multiple classes for the same object drawn in different styles, and ignore other objects completely.
  - PCA decreases the noise in the data and reduces computation. It does well than K-means as it leverages

the information of truth label for each data point. The data is not separable in linear space so accuracy is not considerably good.

- Neural network learns the training data very well and does good on test data as it can fit very complex non linear boundaries.
- Convolutional Neural Network performs the best among all four algorithms. CNN's exploit the fact that features show spatial locality in images. Convolving multiple kernels with the image extract useful features which are absent when each pixel is considered as separate independent entity.

2 Competition

- Libraries used
  - Keras-gpu
  - Tensorflow-gpu
- Preprocessing
  - Randomly flipping the images horizontally
- Results
  - Train Accuracy = 96.173%
  - Test Accuracy = 93.693%
- Architecture

Layer	Description	Activation	Output Shape
Conv2D	64 - (3 x 3 x 1) Kernels	ReLU	(28, 28, 64)
Batch Normalization	NA	NA	(28, 28, 64)
Conv2D	64 - (3 x 3 x 64) Kernels	ReLU	(28, 28, 64)
Batch Normalization	NA	NA	(28, 28, 64)
Max Pooling	Pool size - (2 x 2)	NA	(14, 14, 64)
Conv2D	96 - (3 x 3 x 64) Kernels	ReLU	(14, 14, 96)
Batch Normalization	NA	NA	(14, 14, 96)
Conv2D	96 - (3 x 3 x 96) Kernels	ReLU	(14, 14, 96)
Batch Normalization	NA	NA	(14, 14, 96)
Max Pooling	Pool size - (2 x 2)	NA	(7, 7, 96)
Conv2D	128 - (3 x 3 x 96) Kernels	ReLU	(7, 7, 128)
Batch Normalization	NA	NA	(7, 7, 128)
Conv2D	128 - (3 x 3 x 128) Kernels	ReLU	(7, 7, 128)
Batch Normalization	NA	NA	(7, 7, 128)
Max Pooling	Pool size - (2 x 2)	NA	(4, 4, 128)
Dropout	Drop probability = 0.2	NA	(4, 4, 128)
Flatten	NA	NA	(2048)
Dense	NA	ReLU	(1024)
Dropout	Drop probability = 0.2	NA	(1024)
Dense	NA	Softmax	(20)

Table 1: Deep CNN Architecture

- Other Details
  - Kernel initializer : Glurot Uniform
  - Max Pooling padding : Zero padding
- Hyper-parameters
  - Batch size = 100

  - Epochs = 12
  - Loss function = Negative Log-Likelihood
  - Optimizer = Adam
- Link to model