E9 205 – Machine Learning for Signal Processing

Homework # 5
Due date: Nov. 20, 2019

Submit report in class and the codes as single zip file to mlsp19 doT iisc aT gmail doT com

November 6, 2019

1. **M**NIST data - Download the dataset of hand-written digits
   http : //yann.lecun.com/exdb/mnist/
   containing 10 classes. [Reduce the number of samples in training data if your computing powers are limited as required by random subsampling]. The Keras or PyTorch package is needed for the rest of the question

   (a) **I**mplementing CNNs - Use the Keras package to implement a CNN model with one layer of convolutions (kernel size of 3×3 with a 2-D convolutional layer and having 128 filters) followed by two dense layers of 128 neurons. Use the rectified linear unit (ReLU) non-linearity. Compare the performance of the CNN with the DNN used in Assignment # 4.

   (b) Compare emperically for a fixed neural network architecture different learning algorithsm - momentum parameter, Nesterov accelerated gradient, Adagrad, Adam optimizers. Does the training, validation and test loss depend on the learning scheme used ?

   (c) Provide your answers with analysis for different choices of number of kernels $(128, 256, 512)$ (size of feed forward layers is kept at 128 in all these experiments), maxpooling choices in the convolution layer (no max pooling and (2×2) max pooling) and convolutional kernel sizes (3×3 and 5×5) in the CNN.

   (d) For the two dense layers, compare emperically the benefits of regularization using, L2-weight gradient, dropout strategy with 0.05, 0.1 and 0.2 dropout parameter values. How does the training, validation and test loss depend on the regularization scheme used.

   (e) Pick your favorite choice of neural network parameters (number of kernels, number of hidden neuroons in the feed-forward layer, learning algorithm, presence/absence of regularization) and visualize the hidden activity in the fully connected layers (two layers) for a random subset of 500 test images. Plot the tSNE scatter for hidden activations before training (iter=0), middle of training (niter/2) and the final model using 2-D. Mark the color of the points according to the true class of the test images.

   (**Points** 50)

2. **Comparing Dimensionality Reduction Methods for MNIST** For the MNIST handwritten digit dataset, choose a random subset of 10000 images for training and 1000

images for validation and original MNIST testset for blind testing. Vectorize the images as 784×1. Use one of the following dimensionality reduction methods trained with the 10000 image training set,

(a) Apply PCA to 25 dimensions, and train a DNN with dimensions $(25, 256, 256, 10)$ with softmax non-linearity and two hidden layers to classify the 10 handwritten digits.

(b) Apply LDA on the PCA reduced images to 9 dimensions, and train a DNN with dimensions $(9, 256, 256, 10)$ with same output non-linearity for digit classification.

(c) Train a Gaussian Bernoulli RBM model with PCA input and 9 dimensional hidden activation. Once the RBM is trained, forward pass the PCA input and use the 9 dimensional hidden activation from RBM for training the digit classification similar to LDA case (using the same DNN configuration).

(d) Train an autoencoder using 25 PCA reduced image with size $(25, 128, 9, 128, 25)$. Use a ReLU activation and MSE loss for the Autoencoder. Following the training of the autoencoder, generate the embeddings from 9 dimensional latent layer and use them for training a DNN similar to the LDA case.

Implement all the above classifier models. Use the validation data to check the accuracy of the training on each epoch. Plot the training validation and test accuracy for each of the above methods for every epoch of training. Comment on classification performance on the test data of each of the dimensionality reduction methods using the same MNIST test data. Does any of the methods show advantage over the others. If so, why ? (**Points** 50)