CS559: Machine Learning Name: Tarquin Bennett Due: May 4, 2021

# Assignment 4

Homework assignments will be done individually: each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited. Electronic submission on Canvas is mandatory.

1. **Nearest Neighbors** (10 points) Implement a basic k-NN model on the yeast dataset. The task is to predict the compartment in a cell that a yeast protein will localize to based on properties of its sequence. Apply cross-validation and report your best performance.
   With k-NN models, the only hyperparameter is k. Therefore, to get the best accuracy we need to test with different k's. With this data set, the best k I found was 15. This means that we are taking the 15 closest Neighbors into account when determining the label. This yeild us an accuracy of 61.8%.

2. **Clustering** (5 points) Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can a few (say 3) points that are assigned to different clusters in the kmeans solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example or explain in 1-2 sentences.

   Yes, k-means assigns each data point to a unique cluster based on its distance to the cluster center. Gaussian mixture clustering gives soft (probabilistic) assignment to each data point. Therefore, even if cluster centers are identical in both methods, if Gaussian mixture components have large variances (components are spread around their center), points on the edges between clusters may be given different assignments in the Gaussian mixture solution.

3. **Bayesian Networks** (10 points) Do the following statements hold in each of the above networks ? Please explain your reasoning

   - $A \perp C | B, D$

   For network a, this is true because all paths are inactive between A and C. For network b, this is false because there is an active between A and C once B and D are observed.

   - $B \perp D | A, C$

   For network a, this is false because once C is ovserved the path between B and D is active. For network b, this is true becuase A and C are inactive.

Figure 1: Scatter plot of datasets and the initialized centers of 3 clusters

4. **K-means** (30 points) Given the matrix $X$ whose rows represent different data points, you are asked to perform a k-means clustering on this dataset using the Euclidean distance as the distance function. Here $k$ is chosen as 3. The Euclidean distance d between a vector $x$ and a vector $y$ both in $\mathcal{R}^d$ is defined as $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$. All data in X were plotted in Figure. The centers of 3 clusters were initialized as $\mu_1 = (6.2, 3.2)$(red), $\mu_2 = (6.6, 3.7)$(green), $\mu_3 = (6.5, 3.0)$(blue).

   (a) What's the center of the first cluster (red) after one iteration? (Answer in the format of $[x_1, x_2]$, round your results to three decimal places)
   $[5.633, 3.067]$

   (b) What's the center of the second cluster (green) after two iteration?
   $[5.167, 3.700]$

   (c) What's the center of the third cluster (blue) when the clustering converges?
   $[6.033, 3]$

   (d) How many iterations are required for the clusters to converge?
   4

5. **Expectation Maximization (EM)** (25 points) In this question you will implement the EM algorithm for Gaussian Mixture Models. A good read on gaussian mixture EM can be found at this link. A sample dataset for this problem can be downloaded in canvas files. For this problem:

   - $n$ is the number of training points
   - $f$ is the number of features
   - $k$ is the number of gaussians
   - $X$ is an $n \times f$ matrix of training data
   - $w$ is an $n \times k$ matrix of membership weights. $w(i,j)$ is the probability that $x_i$ was generated by gaussian $j$
   - $\pi$ is a $k \times 1$ vector of mixture weights (gaussian prior probabilities). $\pi_i$ is the prior probability that any point belongs to cluster $i$
   - $\mu$ is a $k \times f$ matrix containing the means of each gaussian
   - $\Sigma$ is an $f \times f \times k$ tensor of covariance matrices. $\Sigma(:,:,i)$ is the covariance of gaussian $i$

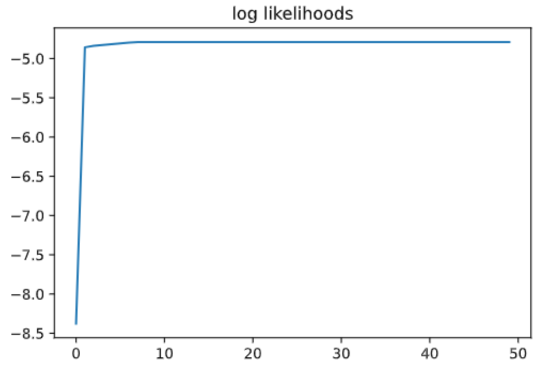
   (a) **Expectation**: Complete the function $[w] = \text{Expectation}(X, k, \pi, \mu, \Sigma)$. This function takes in a set of parameters of a gaussian mixture model, and outputs the membership weights of each data point

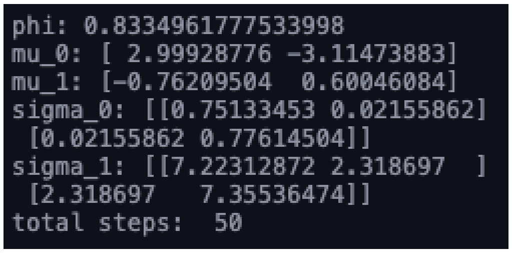
   (b) **Maximization of Means**: Complete the function $[\mu] = \text{MaximizeMean}(X, k, w)$. This function takes in the training data along with the membership weights, and calculates the new maximum likelihood mean for each gaussian.

   (c) **Maximization of Covariances**: Complete the function $[\Sigma] = \text{MaximizeCovariance}(X, k, w, \mu)$. This function takes in the training data along with membership weights and means for each gaussian, and calculates the new maximum likelihood covariance for each gaussian

   (d) **Maximization of Mixture Weights** : Complete the function $[\pi] = \text{MaximizeMixtures}(k, w)$. This function takes in the membership weights, and calculates the new maximum likelihood mixture weight for each gaussian.

   (e) **EM**: Put everything together and implement the function $[\pi, \mu, \Sigma] = \text{EM}(X, k, \pi_0, \mu_0, \Sigma_0, \text{nIter})$. This function runs the EM algorithm for nIter steps and returns the parameters of the underlying GMM. Note: Since this code will call your other functions, make sure that they are correct first. A good way to test your EM function offline is to check that the log likelihood, $\log P(X|\pi, \mu, \Sigma)$ is increasing for each iteration of EM.





```
phi: 0.83349617775339998
mu_0: [ 2.99928776 -3.11473883]
mu_1: [-0.76209504  0.60046084]
sigma_0: [[0.75133453 0.02155862]
 [0.02155862 0.77614504]]
sigma_1: [[7.22312872 2.318697  ]
 [2.318697   7.35536474]]
total steps:  50
```

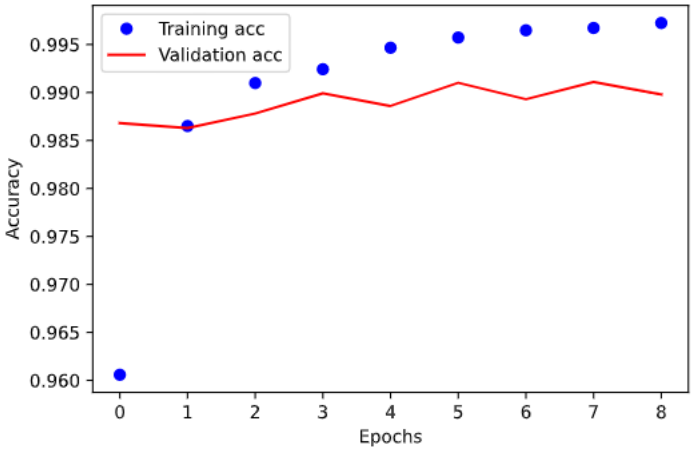
See notebook for code.

6. **Convolutional Neural Networks** (20 points) Develop a Convolutional Neural Network (CNN) model to predict a handwritten digit images into 0 to 9 (You can use Keras or other packages). The pickled file represents a tuple of 3 lists: the training set, the validation set and the testing set. Each of the three lists is a pair formed from a list of images and a list of class labels for each of the images. An image is represented as numpy 1-dimensional array of 784 (28 x 28) float values between 0 and 1 (0 stands for black, 1 for white). The labels are numbers between 0 and 9 indicating which digit the image represents. The code block below shows how to load the dataset.

```
import pickle, gzip, numpy

# Load the dataset
f = gzip.open('mnist.pkl.gz', 'rb')
u = pickle._Unpickler(f)
u.encoding = 'latin1'
train_set, valid_set, test_set = u.load()
f.close()
```

- Choose the proper activation and loss function.
- Plot the train, validation, and test errors as a function of the epochs.
- Report the best accuracy on the validation and test data sets. Discuss the parameter choices such as the filter size, number of filters etc.
- Apply early stopping using the validation set to avoid overfitting.
- Give a brief description of your observations.
- Does pooling make the model more or less sensitive to small changes in the input images? Why? By small changes, we mean moving the input images to the left or right, rotating them slightly etc.



The best validation accuracy is 0.9898 with the test accuracy being 0.989799976348877. By using early stopping we were able to prevent the model from overfitting. The model took 8 epochs to reach a high account. This is due to the high amount of data that each epoch had. Pooling makes the model less sensitive to small changes in the input image. This is because it takes either the average value for each patch on the feature map or the maximum value for each patch of the feature map. Even though small changes change the feature map, when pooling is used it is found that the values of the pooling output do not change due to taking the average or maximum.