ECE 283: Homework 2

Topics: Classification using neural networks
Assigned: Wednesday April 17
Due: Tuesday April 30
Reading: Posted reading on neural networks; Tensorflow/PyTorch tutorials

Use the same 2D synthetic data for binary classification as in HW1
Class 0: Gaussian with mean vector $\mathbf{m} = (0,0)^T$ and covariance matrix $\mathbf{C}$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 2$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = 0$.
Class 1: Gaussian mixture with two components:
Component A: $\pi_A = \frac{1}{3}$, $\mathbf{m}_A = (-2,1)^T$, $\mathbf{C}_A$ with eigenvalue, eigenvector pairs: $\lambda_1 = 2$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1/4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = -\frac{3\pi}{4}$.
Component B: $\pi_B = \frac{2}{3}$, $\mathbf{m}_A = (3,2)^T$, $\mathbf{C}_B$ with eigenvalue, eigenvector pairs: $\lambda_1 = 3$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{4}$.
Variant of the data model: In addition to this dataset, where the classes are rather close to each other, also consider a variant in which the means under class 1 are doubled, so the classes become easy to distinguish.

1) **Use Tensorflow or PyTorch** to implement a fully connected neural network with (a) 1 hidden layer, (b) 2 hidden layers. Use ReLU nonlinearities in the layers, and a sigmoid at the output. Play with the number of neurons in each layer, and use L2 weight regularization.

2) Repeat 1) when you use tanh nonlinearities in the layers, and a sigmoid at the output. Comment on any differences you see in training and inference performance.

3) While these are small neural networks that should be relatively easy to train, discuss the impact of various training tricks such as input pre-processing (e.g., scaling to zero mean and unit variance via empirical estimates across the dataset); weight initialization (e.g., uniform versus zero mean Gaussian, and how you need to scale the initialization based on the number of input connections–this is more relevant for layer 2); learning rates.

4) Comment on the number of training and validation samples you used, how the misclassification error in the training and validation sets evolved, and how/when you stopped adapting the weights.

5) Compare the performance of the neural networks with the results from HW1 (MAP, kernelized logistic regression, and logistic regression with explicit feature engineering). You only need to do this for the original data model, not the variant with better class separation.