ECE 283: Homework 2

Topic: Classification using neural networks
Reading: Posted reading and lecture videos on neural networks; PyTorch tutorial material

0) Use the same 2D synthetic data for binary classification as in HW1
Class 0: Gaussian mixture with two components:
Component A: $\pi_A = \frac{3}{4}$, $\mathbf{m}_A = (0,0)^T$, $\mathbf{C}_A$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 4$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = 0$.
Component B: $\pi_B = \frac{1}{4}$, $\mathbf{m}_B = (6,4)^T$, $\mathbf{C}_B$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{3}$.

Class 1: Gaussian mixture with two components:
Component C: $\pi_C = \frac{2}{3}$, $\mathbf{m}_C = (2,3)^T$, $\mathbf{C}_C$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 2$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{4}$.
Component D: $\pi_D = \frac{1}{3}$, $\mathbf{m}_B = (2,-2)^T$, $\mathbf{C}_D$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 4$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{6}$.

1) Use 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.
1.1) Plot the decision boundaries provided by the neural network side by side with the MAP decision boundaries, and comment on what you see.
1.2) Estimate the probability of misclassification conditioned on each class, and the overall probability of misclassification. Compare with the results from HW1 (MAP benchmark, logistic regression with explicit feature engineering, and kernelized logistic regression).
1.3) 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.
Remark: You are welcome to use Tensorflow if you are proficient with it, but please use PyTorch if you need guidance from the TA. We can only support one framework.

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.