CS 349 Homework 3

Note: all graphs, results, and matrices are outputed on the notebook. Please contact us if you cannot see them.

1. Learning curves are displayed as outputs on the notebook. Confusion matrix and accuracy score are present in the notebook as well. **(iii)** Using a neural network to "predict" health using only 3 metrics is not a bad idea. It's a *horrible* idea. First of all, the labels are as subjective as can be. Who determines what makes someone healthy versus neutral? Additionally, we're assuming that there's a direct correlation between someone's age, exercise, and smoking patterns and someone's health. We need more than just these three attributes to determine someone's health correctly. **(iv)** There were many decisions that we made about our model. In addition to our loss function, we also have to decide our optimizer's learning rate. We settled on 0.01. This makes the model less noisy and note changes its parameters drastically from epoch to epoch. Our loss graph would be more jagged if the learning rate was say 1. Another hyperparameter we chose is the number of epochs we want our model to run through. We chose 50 because that's when we found that our validation set's loss plateaued and stopped decreasing. This impacts our performance because too few epochs and we underfit; too many and we overfit. A very important hyperparameter is the number of nodes in our hidden layer. We chose 32. We noticed that this dataset wasn't too complicated, so we settled on one layer and not too many nodes.
2. Like in 1, our learning rate was 0.01. We did this to make learning a smoother process and minimize the impact of a training example's loss. Another decision we made was batch size. We found that a bigger batch size makes the model learn slower. We chose 16. We have two hidden layers, as opposed to 1 from question 1, with 16 and 32 nodes each, respectively. We chose to go with Leaky ReLU for our activation function and SGD for our loss function. We chose 30 epochs because any more would result in diminishing validation set accuracy. Our training and validation loss graphs are way smoother than the ones from question 1. The training loss is almost perfectly exponentially decaying. The validation loss is way smoother and we can see it starting to increase after the 25 epoch mark. Our results were overall more accurate. This could be because this dataset is objectively better and more accurate than the one in question 1. We see that our model confuses 4s with 9s and also 5s with 8s.
3. We added an L2-regularizer. We took the square root of the sum of all the weight vectors squared. This penalized high-weight values by adding it to our loss function. This resulted in a less stochastic loss graph and more spread-out misclassifications in our confusion matrix. Performance-wise, it was quite similar. We might need to do more epochs for this model.