AI534—Dimension reduction and Neural networks—Problem set

1. **Dimension reduction.** One interpretation of PCA is to seek to find projection directions such that reconstruction error is minimized. Consider a set of data points $\mathbf{x}_i \in R^d, i = 1, ..., n$ that are already centered, i.e., $\sum_i \mathbf{x}_i = 0$. Let $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ be k projection vectors such that $\mathbf{v}_t^T \mathbf{v}_t = 1$ for all $t = 1, \cdots, k$. The projection of point $\mathbf{x}_i$ is $\mathbf{y}_i = [\mathbf{v}_1^T \mathbf{x}_i, \mathbf{v}_2^T \mathbf{x}_i, ..., \mathbf{v}_k^T \mathbf{x}_i]^T \in R^k$. The reconstruction of $\mathbf{x}_i$ is expressed as $\hat{\mathbf{x}}_i = \sum_{t=1}^k (\mathbf{v}_t^T \mathbf{x}_i)\mathbf{v}_t$. The reconstruction error is measured as

$$\sum_{i=1}^{n} |\mathbf{x}_i - \hat{\mathbf{x}}_i|^2,$$

where $|\cdot|^2$ denotes squared L_2 norm.

   (a) Show that minimizing this objective is equivalent to maximizing

$$\sum_{t=1}^{k} \mathbf{v}_t^T \Sigma \mathbf{v}_t,$$

   where $\Sigma = \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T$.

   (b) Show that the optimizing the objective of $\sum_{t=1}^k \mathbf{v}_t^T \Sigma \mathbf{v}_t$ subject to $\mathbf{v}_t^T \mathbf{v}_t = 1$ leads to the top k eigen-vectors (with largest k eigen-values) of Σ.

   (c) Given three data points, (0,0), (1,2), (-1, -2) in a 2-d space. What is the first principal component direction (please write down the actual vector)? If you use this vector to project the data points, what are their new coordinates in the new 1-d space? What is the variance of the projected data?

2. **Neural network expressiveness**
In class, we have discussed that neural network can express any arbitrary boolean functions. Please answer the following question about neural networks. You can assume a step function for the activation function.

   (a) It is impossible to implement a XOR function $y = x_1 \oplus x_2$ using a single unit (neuron). However, you can do it with a neural net. Use the smallest network you can. Draw your network and show all the weights.

   (b) Explain how can we construct a neural network to implement a Naive Bayes Classifier with Boolean features.

   (c) Recall that the decision function of kernelized SVM can be written as:

$$f(\mathbf{x}) = sign(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}))$$

   , where $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ are the training examples, α_i's are the dual variables and $K(\cdot, \cdot)$ is the kernel function. Consider the following polynomial kernel:

$$K(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T \mathbf{x}_2 + 1)^p$$

   Give a neural net with a single hidden layer that represents the above SVM decision function with the polynomial kernel. Hing: for this problem, you do not need to restrict to activation functions introduced in class.