E1 213 Pattern Recognition and Neural networks

Problem Sheet 6

1. Consider a learning problem where $\mathcal{X} = \{x_1, x_2, \cdots\}$ and $\mathcal{Y} = \{0,\ 1\}$. That is, it is a two class problem with feature space being countable. Consider a family of classifiers: $\mathcal{C} = \{h_z : z \in \mathcal{X}\} \cup \{h^-\}$ where h^- classifies all patterns into class-0 and for any $x_i \in \mathcal{X}$, h_{x_i} classifies x_i into class-1 and all others into class-0. That is, this class $\mathcal{C}$ consists of classifiers each of which can put at most one x_i in class-1. We consider PAC learning framework where we assume realizability; that is, there is always one classifier in $\mathcal{C}$ that correctly classifies all training examples. Design a PAC learning algorithm for this problem.

2. A monomial over Boolean variables is a conjunction of literals. A literal is a variable or its compliment. For example, x_1, x_2x_3, $\bar{x}_1x_2x_3$ are all monomials over three Boolean variables. Here, $\bar{x}_1$ denotes the literal which is complement of x_1. Consider a 2-class problem with n Boolean features. Suppose we know that all patterns can be correctly classified by some monomial. (That is, the correct monomial would have value 1 on all feature vectors of C_0 and would have value 0 on all feature vectors from C_1). We want to learn the monomial given some examples. Consider a learning algorithm for this as given below. We start with the monomial $x_1\bar{x}_1x_2\bar{x}_2 \cdots x_n\bar{x}_n$. (Note that this monomial classifies all patterns as C_1). The algorithm is an incremental algorithm processing one example in each iteration. At each iteration we modify the current monomial as follows. If the next example is from C_1 we do nothing. If the next example is from C_0, then, for each $i, 1 \leq i \leq n$, if the example has value 1 for i^{th} feature, then we delete the literal $\bar{x}_i$ (if present) from the current monomial; if the example has value 0 for i^{th} feature, then we delete the literal x_i (if present) from the current monomial. For $n = 3$, take some example target monomial and show how the algorithm works if you take each of the 8 possible feature vectors one by one. Next, show that this is a PAC-learning algorithm. That is, show that given any ϵ and δ, we can find n such that after n random examples, the probability that the error of the classifier learnt by the algorithm is greater than ϵ is less than δ.

3. Consider a 2-class classification problem. Let X be the feature vector and let $y \in \{0, 1\}$ be the class label. Let W^* be the minimizer of

$J(W) = E\left[(W^T X - y)^2\right]$ where the expectation is with respect a joint density P_{Xy}. Now suppose that our process for observing class label is noisy. Hence, when we draw samples, we can only observe $(X, \tilde{y})$ where $\tilde{y}$ is a noisy version of y such that, on any X, with probability $(1-\eta)$, $\tilde{y} = y$ and with probability η we have $\tilde{y} = |1 - y|$. That is, randomly labels are flipped with probability η. Since this is the only thing we can observe, we can, at best, find minimizer of $J'(W) = E\left[(W^T X - \tilde{y})^2\right]$ where now the expectation is with respect to the joint distribution of $X, \tilde{y}$ which is related to P_{Xy} as described above (in terms of the probabilistic relation between y and $\tilde{y}$). Let $\tilde{W}$ be the minimizer of $J'(W)$. Show that, for a random sample (X, y) drawn according to P_{Xy}, the probability of misclassification of W^* and $\tilde{W}$ are the same.

4. Suppose X has the following mixture density:

$$f_X(x) = \sum_{i=1}^{N} \lambda_i f_i(x),\ \lambda_i \geq 0,\ \sum_i \lambda_i = 1$$

Suppose the component densities f_i have mean μ_i and variance σ_i^2, $i = 1, \cdots, N$. Find the mean and variance of X.

5. Consider a three layer feed-forward network with d nodes in the input layer, m nodes in the hidden layer and K nodes in the output layer. All nodes use sigmoidal activation function. How many weights and biases does the network have? How much computation (in terms of number of multiplications etc) is needed for one forward pass (that is to calculate output for a given input vector) and for updating weights using the backpropagation algorithm?

6. Consider a three layer feed-forward network with sigmoidal activation functions. Suppose the input layer and the hidden layer have 2 units each while the output layer has one unit. Suppose all weights and biases are zero. Taking an arbitrary vector in $\Re^2$ as input and an arbitrary value as the desired output, calculate the output of network and the updating of weights using backpropagation. Repeat for a couple of iterations. Based on all this can we say something about initialization of all weights to zero in a feedforward network?

7. Consider feedforward network with one hidden layer and K nodes in

the output layer. Write down the weight update equations if you want to minimize empirical risk with cross entropy loss function.

8. Suppose we want to learn a feedforward network with five hidden layers. We want to use autoencoders for initializing the weights of the network. Suppose we have ten training examples and we use ten iterations of batch-mode backpropagation for training each autoencoder. How many matrix-vector multiplication steps do we have to do for initializing all the weights? (Note that, in any feedforward network, to compute the output of a layer we need to do one matrix-vector multiplication step. Similarly, during backpropagation, for finding the errors at each of the hidden layers we need to do one matrix-vector multiplication step).