Problem Sheet 1

1 Nearest Neighbour Classification

In the lectures, we studied the perceptron, a linear classifier of the form $y = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x} + w_0)$, where $\mathrm{sign}(z) = 1$ if $z \geq 0$ and $\mathrm{sign}(z) = 0$ otherwise. The parameters to be learnt are $\mathbf{w}$ and w_0. The "Nearest neighbour classifier" (NN) is a different approach to learning from data. Suppose we are given N points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ where $y_i \in \{0, 1\}$; for a parameter k and given a new point $\mathbf{x}^*$, the k-NN approach does the following: find $\mathbf{x}_{j_1}, \ldots \mathbf{x}_{j_k}$ the k-closest points to $\mathbf{x}^*$, then output $\widehat{y}^*$ as the majority label from the set $\{y_{j_1}, \ldots, y_{j_k}\}$, *i.e.,* the most commonly occurring label among the k-nearest neighbours.

1. What advantage does the k-NN approach offer over a linear classifier like the perceptron?

2. How many parameters does the nearest neighbour model have? How much memory do you need to store the model? What is the computational cost of predicting the label $\widehat{y}^*$?

3. In this part, we'll look at the setting where the vectors $\mathbf{x}$ are points on the boolean hypercube, *i.e.,* $\mathbf{x} \in \{0, 1\}^D$. Fix $\mathbf{x}^* = (0, 0, \ldots, 0)$ to be the origin and imagine that data consists of points drawn uniformly at random from the boolean hypercube. What is the distribution of the Hamming distance of data points from $\mathbf{x}^*$? What happens as $D \to \infty$? (*Hint*: Use the central limit theorem.)

4. Let us now fix some numbers. Suppose the dimension of the data $D = 10{,}000$; let $\mathbf{x}^* = (0, 0, \ldots, 0)$ and suppose we generated $N = 10{,}000$ data points. What do you expect the distance of $\mathbf{x}^*$ from the nearest data-point to be? the furthest? How large does N need to be to get points that are reasonably close to $\mathbf{x}^*$, say within Hamming distance 50?

Remark: You do not have to write precise numbers or even mathematical expressions for the answers to part 4 above. Make sure you understand the behaviour qualitatively. The phenomenon explored in the last two parts of the question is referred to as the *curse of dimensionality.*

2 Logical Gates Using Perceptrons

Recall that a perceptron with input features $x_1, \ldots, x_D$, weights $w_1, \ldots, w_D$ and bias w_0 outputs the value:

$$y = \begin{cases} 1 & \text{if } w_0 + \sum_{i=1}^{D} w_i x_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

1. Suppose there are at most two inputs and the inputs always take binary values, *i.e.,* $x_i \in \{0, 1\}$. Show how to construct AND, OR and NOT gates by suitably adjusting weights.

2. The constructions for AND and OR gates required only the bias term w_0 to be negative, all other weights were positive. Can you achieve a similar construction for the NOT gate? Why?

3. Can you construct an XOR (exclusive or) gate? If not, give reasons.

4. Often, instead of using a hard threshold we would like to use a continuous approximation. Recall the hyperbolic tangent function $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. We consider another type of *artificial neuron* whose output is defined as

$$y = \tanh\left(w_0 + \sum_{i=1}^{D} w_i x_i\right). \tag{2.2}$$

   Suppose you treat outputs above 0.99 as true and those below -0.99 as false. Show that similar constructions to the ones you had earlier can still be used to construct logic gates.

3 Share Price Prediction using Linear Regression

Note: This example is for illustrative purposes to help you understand linear regression. It is not recommended that you use this for actual share price prediction.

Suppose that $x_0, x_1, \ldots, x_t, x_{t+1}, \ldots,$ denote the (daily) share prices of a particular stock over time. Answer the following questions (you should add a bias term as necessary):

1. Write a linear model to predict x_{t+1} using the share prices on the two preceeding days, *i.e.*, x_t and x_{t-1} .

2. The more useful quantity to predict is $\Delta_{t+1} := x_{t+1} - x_t$, the change in share value. Write a linear model to predict Δ_{t+1} using x_t and x_{t-1}.

3. Write a linear model to predict Δ_{t+1} using Δ_t.

4. Write a linear model to predict Δ_{t+1} using Δ_t and x_t.

5. Which of the above four models, if any, are equivalent? Justify your answer briefly.

6. Given that the only observations you make is the sequence $(x_0, x_1, \ldots, x_t, x_{t+1}, \ldots, x_T)$ for some T, explain how you would train the model in Part (4) above.