Homework 4

Machine Learning

Due: 11:59 pm, April 4

Exercise 1

Perceptrons. A perceptron learns a linear classifier with weight vector w. It predicts

$$\hat{y} = h_w = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} >= 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

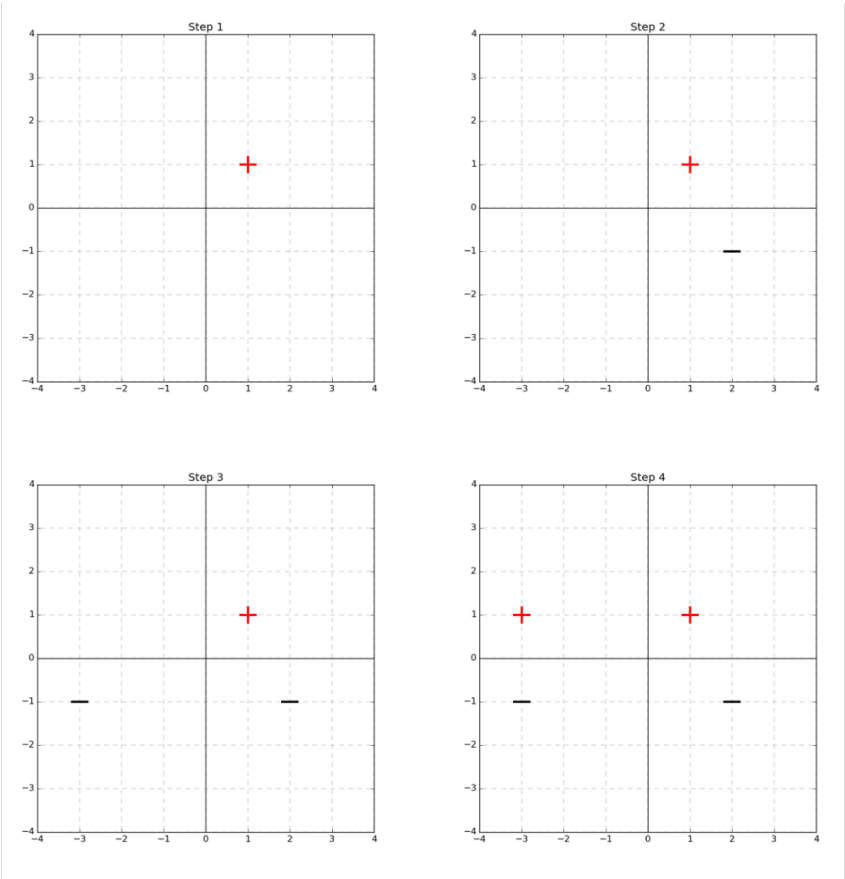
(Note that we are not using a bias weight w_0, for simplicity) When the perceptron makes a mistake, it updates the weights using the formula ($\alpha = 1.0$).

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (y_t - \hat{y}_t)\mathbf{x}_t$$

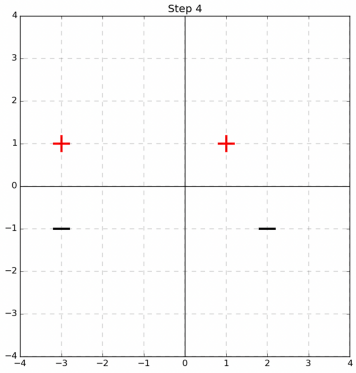
Imagine that we have $\mathbf{x}_t \in \mathbb{R}^2$, and we encounter the following data points. (15 points)

$\mathbf{x}[1]$	$\mathbf{x}[2]$	y
1	1	1
2	-1	0
-3	-1	0
-3	1	1

Q1.1. Starting with $\mathbf{w} = [w_1\ w_2]^T = [1\ 0]^T$, use the perceptron algorithm to learn on the data points in the order from top to bottom. Show the perceptron's linear decision boundary after observing each data point in the graphs below. Be sure to show which side is classified as positive.



Q1.2. Does our learned perceptron maximize the margin between the training data and the decision boundary? If not, draw the maximum-margin decision boundary on the graph below.



Q1.3. What is the margin of the final decision boundary in **Q1.1**? What is the margin of the decision boundary in **Q1.2**? Show the process to calculate the margins. (Ignore the incorrectly classified samples when calculating the margins.)

Exercise 2

The table below provides a set of training examples that can help predict a customer's likelihood of buying a used car. (The Car ID is not an attribute; rather, it serves the purpose of identifying each individual training sample.)

Car ID	Model>2020	Color=Black	Mileage<=30k	HadAccident	Buy
1	Yes	Yes	No	Yes	Yes
2	Yes	Yes	Yes	No	Yes
3	No	No	Yes	No	Yes
4	No	Yes	No	Yes	No
5	Yes	No	Yes	Yes	Yes
6	No	Yes	Yes	Yes	No

Using this data and the pseudocode below, build a decision tree to decide whether a customer buys a used car or not. Show all steps including the information gain of variables. Use the table below to compute entropy. (15 points)

x	1	1/4	2/3	3/4	1/3	1/2
$\log_2 x$	0	-2	-0.6	-0.4	-1.6	-1

function DTL(*examples, attributes, default*) **returns** a decision tree

  if *examples* is empty **then return** *default*
  else if all *examples* have the same classification **then return** the classification
  else if *attributes* is empty **then return** MODE(*examples*)
  else
    $best \leftarrow$ CHOOSE-ATTRIBUTE(*attributes, examples*)
    $tree \leftarrow$ a new decision tree with root test *best*
    for each value v_i of *best* **do**
      $examples_i \leftarrow$ {elements of *examples* with $best = v_i$}
      $subtree \leftarrow$ DTL($examples_i, attributes - best,$ MODE(*examples*))
      add a branch to *tree* with label v_i and subtree *subtree*
    return *tree*

Exercise 3

Adaboost. The AdaBoost algorithm can be summarized as follows:

```
function ADABOOST(examples, L, K) returns a weighted-majority hypothesis
  inputs: examples, set of N labeled examples (x_1, y_1), ..., (x_N, y_N)
          L, a learning algorithm
          K, the number of hypotheses in the ensemble
  local variables: w, a vector of N example weights, initially 1/N
                   h, a vector of K hypotheses
                   z, a vector of K hypothesis weights

  for k = 1 to K do
      h[k] ← L(examples, w)
      error ← 0
      for j = 1 to N do
          if h[k](x_j) ≠ y_j then error ← error + w[j]
      for j = 1 to N do
          if h[k](x_j) = y_j then w[j] ← w[j] · error/(1 − error)
      w ← NORMALIZE(w)
      z[k] ← log (1 − error)/error
  return WEIGHTED-MAJORITY(h, z)
```

Consider the following training set and initial weights:

x	0.1	0.3	0.6	0.9
y	-1	1	1	-1

Assume that the ensemble classifier has two hypotheses ($K = 2$). (15 points)

Q3.1. Assume that $\mathbf{h}[1]$ generates the following results:

x_j	0.1	0.3	0.6	0.9
$\mathbf{h}[1][x_j]$	1	1	1	-1

After the first iteration of the outermost for loop, what are the values of *error*, $\mathbf{w}$, and $\mathbf{z}$?

Q3.2. Assume that $\mathbf{h}[2]$ generates the following results:

x_j	0.1	0.3	0.6	0.9
$\mathbf{h}[2][x_j]$	-1	1	1	1

After the second iteration of the outermost for loop, what are the values of *error*, $\mathbf{w}$, and $\mathbf{z}$?

Q3.3. Given that $\mathbf{h}[1]$ classifies a particular point as -1 and $\mathbf{h}[2]$ classifies it as 1, what class will be selected by the ensemble? Justify your answer.

Exercise 4

Kernel Functions. Support vector machines are able to produce non-linear decision boundaries by, in a sense, transforming low-dimensional inputs into a high-dimensional space, then performing classification in that high-dimensional space. This usually works because high-dimensional data is much more likely to be linearly separable than low-dimensional data.

As an example, consider the following two-dimensional data points: A = (1, 2), B = (2, 4). Use the following function to map these two data points into the indicated six-dimensional space:

$$\Phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

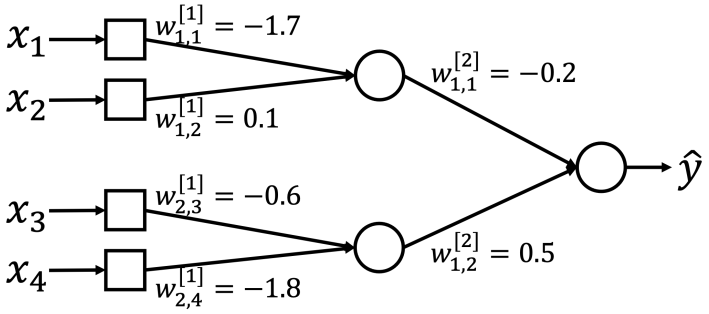
Q4.1. What are $\Phi(A)$ and $\Phi(B)$?

Q4.2. Now calculate the dot product between these two points in the space defined by Φ.

Q4.3. What is the Kernel function for Φ? Calculate $K(A, B)$ using the Kernel function. (Hint: polynomial kernel.)

Exercise 5

Artificial neural network. Consider the following neural network. In all neurons, the input function is $z^{[l]} = w^{[l]T} a^{[l-1]} + b^{[l]}$ and the activation function is sigmoid function, $g(x) = \sigma(x) = \frac{1}{1+e^{-x}}$. The neurons do not have a connection when the weight is zero (*e.g.*, $w_{1,3}^{[1]} = 0$).



$$b_1^{[1]} = b_2^{[1]} = b_1^{[2]} = 0$$

Suppose we have an L2 loss $L(y, \hat{y}) = \|y - \hat{y}\|_2^2$. We are given a data point $(x_1, x_2, x_3, x_4) = (-0.7, 1.2, 1.1, -2)$ with true label 0.5 (*i.e.*, $y = 0.5$). Use the backward propagation equations below to compute the partial derivative $\frac{\partial L}{\partial w^{[1]}}$. (15 points)

$$dz^{[l]} = da^{[l]} * g^{[l]\prime}(z^{[l]})$$
$$dw^{[l]} = dz^{[l]} \cdot a^{[l-1]^T}$$
$$db^{[l]} = dz^{[l]}$$
$$da^{[l-1]} = w^{[l]T} \cdot dz^{[l]}$$

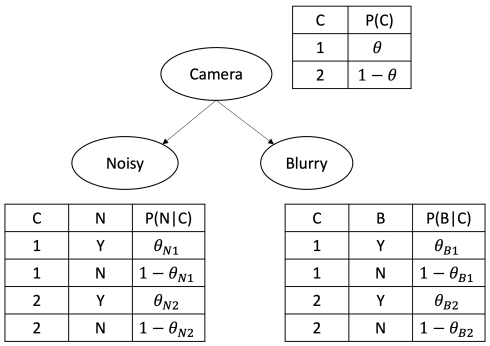
* in the first equation is element-wise multiplication.

Hints:

- The derivative of an L2 loss function is $L'(y, \hat{y}) = 2\|y - \hat{y}\|$.
- The derivative of the sigmoid function $\sigma'(x) = \sigma(x)(1 - \sigma(x))$.
- You can get the value of the sigmoid function from this website.
  `https://www.vcalc.com/wiki/vCalc/Sigmoid+Function`

Exercise 6

Expectation-maximization algorithm. Consider the following Bayesian network. The Bayesian network describes the possibility of capturing noisy and blurry photos with different cameras.



C	P(C)
1	θ
2	$1-\theta$

C	N	P(N|C)
1	Y	θ_{N1}
1	N	$1-\theta_{N1}$
2	Y	θ_{N2}
2	N	$1-\theta_{N2}$

C	B	P(B|C)
1	Y	θ_{B1}
1	N	$1-\theta_{B1}$
2	Y	θ_{B2}
2	N	$1-\theta_{B2}$

Assume that you collected 1000 images with a missing variable, *Camera*, as shown in the following table.

	$Noisy = Yes$	$Noisy = No$
$Blurry = Yes$	34	102
$Blurry = No$	193	671

Given that the parameters are initialized with the following values, complete the first iteration of the expectation-maximization process.

$$\theta^{(0)} = 0.5, \theta_{N1}^{(0)} = \theta_{B1}^{(0)} = 0.1, \theta_{N2}^{(0)} = \theta_{B2}^{(0)} = 0.2$$

Show all steps. Report the values of $\theta^{(1)}, \theta_{N1}^{(1)}, \theta_{B1}^{(1)}, \theta_{N2}^{(1)}, \theta_{B2}^{(1)}$. (25 points)