CSC311 midterm

1 Midterm A

1. In this question, your job is to implement a 1-nearest-neighbour classifier in NumPy. However, unlike our KNN algorithm from lecture, which used Euclidean distance, you will instead use the L^1 distance. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$, this is defined as: $\|\mathbf{x} - \mathbf{y}\|_1 = \sum_{j=1}^{D} |x_j - y_j|$. (The bars $|\cdot|$ represent absolute value.)

   Assume you are given an $N \times D$ data matrix `X`, where each row corresponds to one of the training input vectors, and an N-dimensional vector `y` containing the labels. You are given a query vector `x_query`, and your code should output the nearest neighbour prediction.

   If you don't remember the API for a NumPy operation, then for partial credit, explain what you are trying to do. Your answer should not use a for loop, but solutions that use a for loop will receive partial credit.

   ```
   Solution:
   Xq  = np.concatenate(x_query.reshape((1, -1)), 0)   # various alternatives are possible here
   dist= np.abs(X-Xq).sum(-1)
   return y[np.argmin(dist)]
   ```

2. We have a random variable $X \in \{0, 1\}$. In particular, we have $P(X = 0) = 0.4$ and $P(X = 1) = 0.6$. For another random variable Y and we know the following condition probabilities:

$$P(Y = 0|X = 0) = 0.8, \quad P(Y = 1|X = 0) = 0.2$$
$$P(Y = 0|X = 1) = 0.3, \quad P(Y = 1|X = 1) = 0.7$$

   Please compute the entropy $H(Y)$ and conditional entropy $H(Y|X)$.

   ```
   Solution:
    P(Y=1) = 0.5
    H(y) = -plog p - (1-p)log (1-p)
         = -log 0.5=1
    H(Y|X) = - (0.32 log 0.8+
                0.08 log 0.2+
                0.18 log 0.3+
                0.42 log 0.7)

    Note: a numerical answer here is OK.
   ```

3. You are given access to the following dataset for binary-classification. Your goal is to build a decision tree which classifies Y as T/F using A, B, C as features.

   (a) Write down a decision tree that would be learned via the greedy algorithm that thresholds features based on information gain.

```
Calculations for information gain:
IG[Feat1] = 1-1
IG[Feat2] = 1-0
IG[Feat3] = 1-0

Step 1 - pick Feat1

A = Feat1
C = return T

Step 2 - either Feat2 or Feat3 will have the same IG
[assume student chooses Feat2]
B = Feat2
D = Feat3
H = return T
I = return F
E = return F
```

   (b) An optimal binary decision tree is one that achieves zero training error with minimal *depth*. Is the above tree optimal? If not draw the optimal binary decision tree for this dataset.

```
The tree above is not optimal.

Notice that Y = Feat2 XOR Feat3
A tree with either B at the root and C below (or vice versa) will be optimal.

Optimal tree 1:
A=Feat2
B=Feat3
C=Feat3
D=return T
E=return F
F=return F
G=return T

Optimal tree 2:
A=Feat3
B=Feat2
```

```
C=Feat2
D=return T
E=return F
F=return F
G=return T


Edit (thanks Stephen Zhao)
Some students may have split on feature 2
and then feat3 if feat2=T and feat1 if feat2=F.
This will indeed result in an optimal tree with
the same depth as the above two.
```

Feat1	Feat2	Feat3	Y
F	F	F	T
T	F	T	F
T	T	F	F
T	T	T	T

4. In lecture, we discussed polynomial regression in one variable. However, it's also possible to fit polynomials of more variables. A degree-2 polynomial in two variables u and v is a sum of monomials which are at most quadratic in u and v; an example is

$$u^2 + 3uv + v + 5.$$

Define a feature map $\phi(u, v)$ which you can use to fit a degree-2 polynomial in u and v with linear regression. No justification is required. (Note: we'd ordinarily denote the two input dimensions as x_1 and x_2, but we used u and v instead to save you some formatting difficulty.)

```
Solution:
    phi(u,v) = [1,u,v,u**2,uv,v**2]
```

5. Consider the following dataset representing the XNOR boolean function.

x_1	x_1	y
0	0	1
0	1	0
1	0	0
1	1	1

(a) Is the data linearly separable?

(b) Briefly justify your choice on whether the data is (or is not) linearly separable.

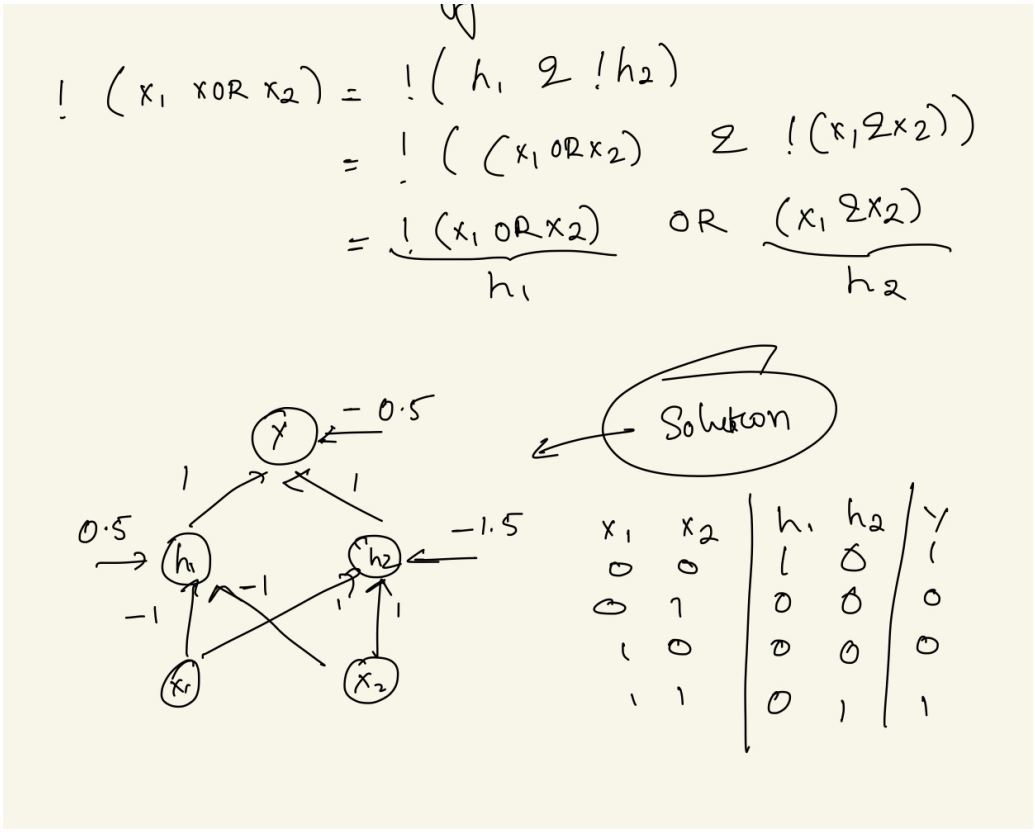


Figure 1: Solution

```
Not linearly separable.
Justification -- can prove via contradiction using argument in class.
Alternatively, notice that this is simply XOR with the labels flipped,
so if XOR is not linearly separable, then neither is XNOR.
```

(c) Design a two layer multi-layer perceptron using the hard threshold activation function $\phi(x) = \mathbb{I}[x > 0]$ to perfectly classify the above dataset. While no justification is required, explaining what you intend the hidden units to do may help you receive partial credit.

```
Solution: See Figure 1 for weights.

In practice, any scalar multiplier on those weights
would also work so you should be able to manually verify that the
four training examples are a perfect classifier.
```

6. In lecture, we observed that when doing classification with cross-entropy loss , it can be useful

to compress the predictions to be in the interval $[0, 1]$ using the logistic function σ. Here is a different function which compresses the predictions to be in $[0, 1]$:

$$\phi(z) = \max(0, \min(z, 1)).$$

What would go wrong if we try to do least-squares classification using ϕ in place of σ? RBG: I think this question might have been updated since the solution was written.

```
Solution:
    Anything value of z above 1 is ignored
    Any value of z below 0 is ignored
    Issues: no gradient signal beyond [0,1] -- cannot learn with this activation
```

7. Is the composition of a convex function and a quadratic function (e.g., $y = x^2$) is still a convex function. If yes, justify your answer. Otherwise, give a counterexample.

```
Solution: it is not convex.

For instance, take g(z) = (z - 1)^2 and f(x) = x^2. The
function g(f(x)) is minimized when x = +/- 1, while the
value at x=0 is larger.

We forgot to specify that the quadratic is a convex quadratic.
So another valid solution would be g(z) = z and f(x) = -x^2.
```

8. Finite differences is useful for testing the implementation of a gradient computation. Give one reason why we don't simply use finite differences to compute all our gradients when we run gradient descent.

```
Possible answers:
(a) It requires a separate function evaluation for every coordinate of
the gradient, which is very expensive.
(b) It only returns an approximation to the gradient, not the exact
value.
```

9. We said in lecture that it's important to tune some hyperparameters on the validation set rather than the training set since the optimal choice on the training set could overfit. Suppose you have a dataset where all of the input vectors are distinct. For which of the following pairs of algorithm and hyperparameter is it *guaranteed* that the optimal hyperparameter setting on the training set will get perfect training accuracy? For some parts, either a YES or a NO answer can be justified.

   (a) algorithm = K-nearest-neighbours, hyperparameter = K

   (b) algorithm = decision tree, hyperparameter = depth

(c) algorithm = logistic regression with L^2 regularization, hyperparameter = λ the weight on the L^2 penalty

(d) algorithm = multilayer perceptron, hyperparameter = layer width

```
Solution:
    (a) Yes - for K=1, each training example's nearest neighbor is itself, so
    the accuracy is perfect.
    (b) Yes - with large enough depth, the leaves keep getting split until they
    only have a single label.
    (c) No. - it may be impossible to fit the data with a linear model.
    (d) Possible answers: Yes, because MLPs are nonlinear function approximators. No,
    because the optimization algorithm might get stuck in a local optimum. It
    depends on the activation function (linear activation function would make the
    architecture non-universal).
```

2 Midterm B

1. You have discovered a new machine learning for binary classification called "Not K farthest strangers". The algorithm first finds the labels of the K most distant examples in the training set of size N. Then, the most frequently occurring label is found and flipped (i.e. if its a 1, it is flipped to 0 and vice-versa).

   (a) Is this a parametric or non-parametric learning algorithm? Justify your choice.
   Solution: Non-parametric, because the hypothesis space depends on the dataset and cannot be described by finitely many parameters.

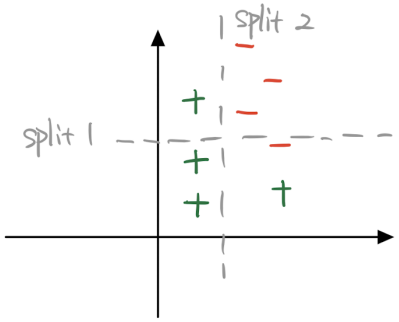
   (b) What is the test-time complexity of the algorithm, i.e. the computational cost of making a classification? Give your answer in big-O notation in terms of N (the number of training examples), K (the number of strangers), and D (the input dimension). Briefly justify your answer.
   Solution: $O(ND + N \log N)$ for computing the distance and the sorting. It is also OK for $O(ND + N \log N + K)$

2. We are considering the following two splits for decision tree model. The green plus, and red minus represent positive and negative examples, respectively.

   (a) What's the entropy before splitting?
   Solution: 1 bit.

   (b) What's the information gain of each split (in bits)? Show your work.
   Solution: for split 1, the information gain is $H(Y) - H(Y|X) = 1 - 0.25 \times 2 - 0.75 \times 0.415 = 0.189$. For split 2, the information gain is $H(Y) - H(Y|X) = 1 - 0.675 \times (0.2 \times \log_2 5 + 0.8 \times \log_2 1.25) = 0.549$.

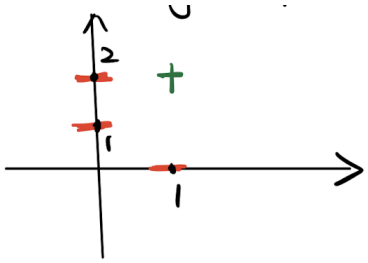
   (c) Which split is better in terms of information gain? Report the accuracy of each split.
   Solution: The split 2 is better. The accuracy of split 1 is 0.75 while the accuracy of split 2 is 0.875.



3. You are trying to solve a binary classification problem with inputs x and targets $t \in \{0, 1\}$. You are lucky enough to know the full data generating distribution $p(x, t)$. The loss function is zero-one loss, i.e. $\mathcal{L}_{0-1}(y, t) = \mathbb{I}[y \neq t]$. Determine the Bayes optimal prediction y_* for a given x, and briefly explain why this is Bayes optimal.

   Solution: the Bayes optimal prediction y_* is $\mathbb{I}[p(x, 1) > p(x, 0)]$, or equivalently, $\arg\max_t p(t|x)$ (ignoring the corner case of $p(x, 0) = p(x, 1)$). Basically, one minimizes the expected loss $\mathbb{E}_t[\mathcal{L}(y, t)|x]$, which has the form $p(1|x)\mathbb{I}[y = 0] + p(0|x)\mathbb{I}[y = 1]$.

4. Given the following input space picture, please write down all the constraints in the weight space and give a solution if it is feasible. We assume no dummy feature for this model. **Solution:**



$$\begin{aligned} w_1 &< 0 \\ w_2 &< 0 \\ 2w_2 &< 0 \\ w_1 + 2w_2 &\geq 0 \end{aligned} \tag{1}$$

   This is infeasible. (Any constraints equivalent to the above get full credit, so it's OK to combine the second and third inequalities.)

5. You are fitting a (least squares) regression model where you expect the targets to have a sinusoidal dependency on a scalar-valued input x. Therefore, you make predictions as $y = \sin(wx + b)$. The loss is squared error, $\mathcal{L}(y, t) = \frac{1}{2}(y - t)^2$. Determine the stochastic gradient descent update for w on a single training example (x, t) with learning rate α. (You don't need to worry about the gradient descent update for b.)

Solution: $w_{t+1} \leftarrow w_t - \alpha(y - t) \times \cos(w_t x + b) \times x$.

6. Samir and Anna are both asked to build a model for binary classification. Anna trains a logistic regression model with $y = \sigma(w^\top x)$, where "cat" is the positive class and "dog" is the negative class. Samir trains a multi-class classification model (softmax regression) with two classes using the matrix $W \in \mathbb{R}^{2 \times D} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, where label 1 is "dog" and label 2 is "cat".

   Under what condition do the two classifiers make the same predictions (interpreted as probabilities)? Give your answer as an equation relating $w, w1, w2$. Briefly justify your answer.

   Solution: $w_2 - w_1 = w$

7. True or false: in fitting a linear regression model, if the current value of the loss is 0 on the entire training set, then the gradient descent update will be a no-op (i.e. the weights won't change). Briefly justify your answer.

   Solution: True. If the loss is 0, then the current weights are an optimum, which means the gradient is 0.

8. Alice and Bob plan to train the same softmax regression model for image classification (same architecture and same initialization). At initialization, the training accuracy is 10%. Alice decides to use stochastic gradient descent while Bob decides to use batch gradient descent. Assuming both select the optimal learning rate, then to achieve 50% accuracy (assume the network is capable of getting this accuracy)

   (a) Would Alice takes more iterations or less?

   (b) Would Alice takes more epochs or less (the number of epochs indicates the number of passes of the entire training dataset)?

   Solution: Alice will require more iterations because each iteration is noisier. She will require fewer epochs because she will be able to do more weight updates per epoch, and hence make faster progress.

9. For a one-layer neural network with identity activation and one hidden unit (i.e., $y = w_2 w_1 x$).

   (a) Please represent the same function with a one-layer neural network with ReLU activation $\phi(x) = \max\{x, 0\}$ and two hidden units. **Hint:** choosing the weights of ReLU network (w_{11}, w_{12} for the first layer, w_{21}, w_{22} for the second layer).

   (b) Does this indicate that one-layer ReLU networks are at least as expressive as one-layer linear networks? Justify your answer.

   Solution: (a) $w_{11} = -w_{12} = w_1$ and $w_{21} = -w_{22} = w_2$. **There are other possible answers. Basically one can rescale the weights in the first layer and second layer.** (b) Yes, for any one-layer linear networks, we can simulate the same function by doubling the number of neurons.