Solutions

10-601 Machine Learning	**Name:**
Spring 2019	**Andrew Email:**
Final Worksheet	**Room:**
05/01/2019	**Seat:**
Time Limit: 180 minutes	**Exam Number:**

Instructions:

- Fill in your name and Andrew ID above. Be sure to write neatly, or you may not receive credit for your exam.
- Clearly mark your answers in the allocated space **on the front of each page.** If needed, use the back of a page for scratch space, but you will not get credit for anything written on the back of a page. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- No electronic devices may be used during the exam.
- Please write all answers in pen.
- You have 180 minutesto complete the exam. Good luck!

Instructions for Specific Problem Types

For "Select One" questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

● Matt Gormley

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

● Matt Gormley

○ Marie Curie

⊗ Noam Chomsky

For "Select all that apply" questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

⊠ I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-601 10-~~7~~601

1 Worksheet Questions

1.1 K - Nearest Neighbors

1. (1 point) **Select one:** A k-Nearest Neighbor model with a large value of K is analogous to...

   ◯ A *short* Decision Tree with a *low* branching factor

   ◯ A *short* Decision Tree with a *high* branching factor

   ◯ A *long* Decision Tree with a *low* branching factor

   ◯ A *long* Decision Tree with a *high* branching factor

   A short Decision Tree with a low branching factor

2. (1 point) **Select one.** Imagine you are using a k-Nearest Neighbor classifier on a data set with lots of noise. You want your classifier to be *less* sensitive to the noise. Which is more likely to help and with what side-effect?

   ◯ Increase the value of k => Increase in prediction time

   ◯ Decrease the value of k => Increase in prediction time

   ◯ Increase the value of k => Decrease in prediction time

   ◯ Decrease the value of k => Decrease in prediction time

   Increase the value of k => Increase in prediction time

3. (1 point) **Select all that apply:** Identify the correct relationship between bias, variance, and the hyperparameter k in the k-Nearest Neighbors algorithm:

   ☐ Increasing k leads to increase in bias

   ☐ Decreasing k leads to increase in bias

   ☐ Increasing k leads to increase in variance

   ☐ Decreasing k leads to increase in variance

   A and D

1.2 Perceptron

Suppose you are given the following dataset:

Example Number	X_1	X_2	Y
1	-1	2	-1
2	-2	-2	+1
3	1	-1	+1
4	-3	1	-1

You wish to perform the Batch Perceptron algorithm on this data. Assume you start with initial weights $\theta^T = [0, 0]$, bias $b = 0$ and that we pass all of our examples through in order of their example number.

1. (1 point) **Numerical answer:** What would be the updated weight vector θ be after we pass example 1 through the perceptron algorithm?

   $[1, -2]$

2. (1 point) **Numerical answer:** What would be the updated bias b be after we pass example 1 through our the Perceptron algorithm?

   -1

3. (1 point) **Numerical answer:** What would be the updated weight vector θ be after we pass example 2 through the Perceptron algorithm?

   $[1, -2]$

4. (1 point) **Numerical answer:** What would be the updated bias b be after we pass example 2 through the Perceptron algorithm?

   -1

5. (1 point) **Numerical answer:** What would be the updated weight vector θ be after we pass example 3 through the Perceptron algorithm?

   $[1, -2]$

6. (1 point) **Numerical answer:** What would be the updated bias b be after we pass example 3 through the Perceptron algorithm?

   −1

7. (1 point) **True or False:** You friend stops you here and tells you that you do not need to update the Perceptron weights or the bias anymore, is this true or false?

   ◯ True

   ◯ False

   True, all points are classified cor

8. (2 points) **True or False:** Data (X,Y) has a non-linear decision boundary. Fortunately there is a function $\mathcal{F}$ that maps (X,Y) to ($\mathcal{F}$(X),Y) such that ($\mathcal{F}$(X),Y) is linearly separable. We have tried to build a modified perceptron to classify (X,Y). Is the given (modified) perceptron update rule correct ?

   if sign$(w\mathcal{F}(x^{(i)}) + b)$!= $y^{(i)}$:
   $w' = w + y^{(i)}\mathcal{F}(x^{(i)})$
   $b' = b + y^{(i)}$

   ◯ True

   ◯ False

   True

1. (1 point) **Select all that apply:** Which of the following are considered as inductive bias of perceptron.

   ☐ Assume that most of the cases in a small neighborhood in feature space belong to the same class

   ☐ Decision boundary should be linear

   ☐ Prefer to correct the most recent mistakes

   ☐ Prefer the smallest hypothesis that explains the data

   BC

2. (1 point) **True or False:** If the training data is linearly separable and representative of the true distribution, the perceptron algorithm always finds the optimal decision boundary for the true distribution.

   ◯ True

   ◯ False

False.

1.3 Linear Regression

1. (1 point) **Select one:** The closed form solution for linear regression is $\theta = (X^T X)^{-1} X^T y$. Suppose you have n = 35 training examples and m = 5 features (excluding the bias term). Once the bias term is now included, what are the dimensions of X, y, θ in the closed form equation?

   ◯ X is 35×6, y is 35×1, θ is 6×1

   ◯ X is 35×6, y is 35×6, θ is 6×6

   ◯ X is 35×5, y is 35×1, θ is 5×1

   ◯ X is 35×5, y is 35×5, θ is 5×5

   A.

2. (1 point) (True or False) A multi-layer perceptron model with linear activation is equivalent to linear regression model.

   ◯ True

   ◯ False

   True

3. (2 points) **Select all that apply:** You are given a variable z to predict with 2 covariates x_1 and x_2. Which of the following equations relating x_1 and x_2 **could** lead to a closed-form solution $\boldsymbol{\theta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ if you run a linear-regression of x_1 and x_2 on z : $z = \beta_1 x_1 + \beta_2 x_2$?

   Note: The i^{th} row of $\mathbf{X}$ contains the i^{th} data point $(x_{i,1}, x_{i,2})$ while the i^{th} row of $\mathbf{y}$ contains the i^{th} data point y_i. Ignore the bias term in $\mathbf{X}$.

   ☐ $x_2 = 2x_1 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$

   ☐ $x_2 = 3x_1$

   ☐ $x_2 = 2x_1^2 + x_1$

   ☐ $x_2 = x_1^3$

   ☐ $x_2 = \sin x_1$

   ☐ None of the Above

   (a), (c), (d), (e). Only in (b) are we guaranteed that the columns corresponding to x_1 and x_2 in the X matrix are linearly dependent. In (c), (d) and (e), the columns corresponding to x_1 and x_2 in the X matrix are not linearly independent. Additionally, in (a), due to the ϵ, x_1 and x_2 may not be linearly dependent.

1.4 Regularization, Optimization

1. (1 point) **Select one:** Which of the following is true about the regularization parameter λ (the parameter that controls the extent of regularization):

   ◯ Larger values of λ can overfit the data.

   ◯ Larger λ does not affect the performance of your hypothesis

   ◯ Adding a regularization term to a classifier, ($\lambda \neq 0$), may cause some training examples to be classified incorrectly.

   C

1.5 Decision Trees

1. (2 points) ID3 algorithm is a greedy algorithm for growing Decision Tree and it suffers the same problem as any other greedy algorithm that finds only locally optimal trees. Which of the following method(s) can make ID3 "less greedy"? **Select all that apply:**

   ☐ Use a subset of attributes to grow the decision tree

   ☐ Use different subsets of attributes to grow many decision trees

   ☐ Change the criterion for selecting attributes from information gain (mutual information) to information gain ratio (mutual information divided by entropy of splitting attributes) to avoid selecting attributes with high degree of randomness

   ☐ Keep using mutual information, but select 2 attributes instead of one at each step, and grow two separate subtrees. If there are more than 2 subtrees in total, keep only the top 2 with the best performance (e.g., top 2 with lowest training errors at the current step)

   2nd and 4th choices; 1st choice should be wrong as the best performance will be determined by the deepest tree. Any shallower tree will make more mistakes, so ensemble learning can only make performance worse and it won't change the local optimality of the forest.

1.6 Probabilistic Learning

2. (1 point) Consider a logistic regression model for a binary classification problem $Y \in \{0, 1\}$. We have observed a training data set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ where the N training data points are mutually independent. Write down an expression for the likelihood, i.e. the probability of the observed training data,

$$P(Y^{(1)} = y^{(1)}, \ldots, Y^{(N)} = y^{(N)} \mid \mathbf{x}^{(1)} = x^{(1)}, \ldots, \mathbf{x}^{(N)} = x^{(N)}, \Theta)$$

expressed in terms of the class-1 probabilities,

$$p(x, \Theta) := P(Y = 1 | \mathbf{x} = x, \Theta).$$

Since all data points are independent we can write the full probability as a product i.e.,

$$P(Y^{(1)} = y^{(1)}, \ldots, Y^{(N)} = y^{(N)} \mid \mathbf{x}^{(1)} = x^{(1)}, \ldots, \mathbf{x}^{(N)} = x^{(N)}, \Theta) = \prod_{i=1}^{N} P(Y^{(i)} = y^{(i)} \mid \mathbf{x}^{(i)} = x^{(i)}, \Theta).$$

using the class-1 probabilities we have

$$= \prod_{i=1}^{N} p(x^{(i)}, \Theta)^{y^{(i)}} (1 - p(x^{(i)}, \Theta))^{1-y^{(i)}}$$

3. (1 point) Write down an expression for the log-likelihood function from the derivation above:

$$= \log \left[\prod_{i=1}^{N} p(x^{(i)}, \Theta)^{y^{(i)}} (1 - p(x^{(i)}, \Theta))^{1-y^{(i)}} \right]$$

$$= \sum_{i=1}^{N} y^{(i)} \log p(x^{(i)}, \Theta) + (1 - y^{(i)}) \log(1 - p(x^{(i)}, \Theta))$$

4. Let $\mathcal{D} = X_1, X_2, X_3, X_4, X_5$ be a set of i.i.d. random variables which can be modelled as a Gaussian. The mean and standard deviation of the distribution is dependent on both the output class and the attribute. The result is a probability distribution over 10 classes. George wishes to train a Naive Bayes classifier to accomplish this task. How many parameters must George estimate to train such a classifier?

Select one:

- ◯ 100
- ◯ 110
- ◯ 109
- ◯ 111

2 parameters (mean and standard deviation) need to be computed for each of the m features and K classes. This yields 2mK parameters. Here m=5 and K=10, that results in 100 parameters. We also need to estimate the priors for P(Y). There would be 9 such parameters since sum of all of these is 1. All in all we need to estimate 109 parameters.

5. Sally however feels unsatisfied with George's approach to the problem. She makes the bold assumption that since all the noise in the data arises from a single common source, one should consider the standard deviation to be independent of both the class and the attribute. She then proceeds to train the Naive Bayes classifier. How many parameters must she estimate to train the classifier?

   Select one:

   ◯ 50

   ◯ 51

   ◯ 60

   ◯ 61

   Solution: 1 parameter (mean) needs to be computed for each of the m features and K classes. This yields mK parameters. Here m=5 and K=10, that results in 50 parameters. There would be one common standard deviation for all the Gaussians.That makes the parameter count 51.
   We also need to estimate the priors for P(Y). There would be 9 such parameters since sum of all of these is 1.
   All in all we need to estimate 60 parameters.

6. Let $\mathcal{D} = X_1, X_2, \ldots, X_n$ be a set of i.i.d. random variables with a Poisson Probability distribution.

$$p_X(x) = \mathbb{P}(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

   Derive the Maximum Likelihood estimate for the same.

   Solution:

$$L(D : \lambda) = \prod_{i=1}^{n} p\left(x_i|\lambda\right) = \prod_{i=1}^{n} \lambda^{x_i} e^{-\lambda} \left(x_i!\right)^{-1}$$

   Take log of both sides. RHS will be

$$\sum_{i=1}^{n} \log \left[\lambda^{x_i} e^{-\lambda} \left(x_i!\right)^{-1}\right]$$

$$\sum_{i=1}^{n} x_i \log \lambda - \lambda - \log\left(x_i!\right)$$

Differentiate wrt λ to get:

$$\sum_{i=1}^{n} \left(\frac{x_i}{\lambda} - 1 \right)$$

Simplify further to get:

$$\lambda = \frac{\sum_{i=1}^{n} x_i}{N}$$

1.7 Logistic Regression

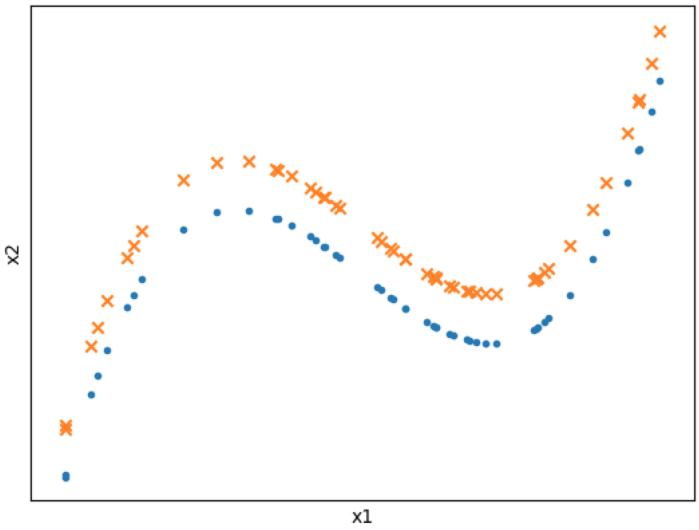

Figure 1: 'x' and 'o' are examples from different classes

7. (2 points) Consider the input data as shown in figure 1. (Select all that apply) Which of the following sets of features would you use to apply logistic regression to this data?

   ◯ x_1, x_2

   ◯ x_1, x_2, x_1^2, x_2^2

   ◯ x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3

   ◯ x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1^4, x_2^4

C. A and B would undrfit while D would overfit.

8. (1 point) **True or False:** Logistic regression gives a probability distribution over the output labels for an input example.

   ◯ True

   ◯ False

   True.

9. (1 point) **True or False:** Before applying Logistic regression, normalizing the input data i.e., rescaling inputs so that each feature has values between a fixed range [a, b], is necessary for the algorithm to converge even on linearly separable data.

   ◯ True

   ◯ False

   False.

10. (2 points) Select all that apply. Suppose you train a logistic regression module for binary classification with 2 outputs, use a softmax activation over it and use the cross entropy loss with non-negligible L2 regularization. Which activation function can I use to replace the softmax with and still learn the same decision boundary. Assume the data is linearly separable

    ☐ softmax activation with 1 output unit

    ☐ sigmoid activation with 1 output unit

    ☐ softmax activation with 3 output units

    ☐ sigmoid activation with 2 output units, that is applied indepently to each unit

    B

1.8 Deep Learning

11. (2 points) Suppose you are given an input image and a convolution filter.

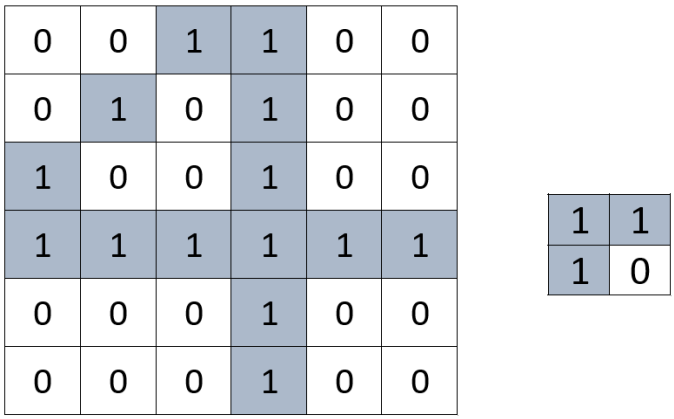

Figure 2: Input image (left) and Convolution filter (right)

Perform a convolution with a stride of 2 over this image using the given filter and put the output in the table below.

0	2	0
2	2	1
0	1	0

1.9 Optimization

1. (3 points) **Derivation.** The objective function of a perceptron function can be thought of as:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(-y^{(i)} \left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right)_+$$

. where

$$(x)_+ = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

In the update rule of a perceptron, we only update the weights θ when $y_i \neq (\boldsymbol{\theta}^T \mathbf{x}^{(i)})$, where y_i is the actual label of i^{th} data point and $(\boldsymbol{\theta}^T \mathbf{x}^{(i)})$ is the predicted label given by the perceptron model.
Given the objective function, derive this update rule.

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(-y^{(i)} \left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right)_+$$

$$\frac{dJ(\theta)}{d\theta} = \sum_{n}^{i=1} Z_i$$

, where

$$Z_i = \begin{cases} -y^{(i)} x^{(i)}, & \text{if } y^{(i)}(\theta^T x^{(i)}) \leq 0 \\ 0, & \text{if } y^{(i)}(\theta^T x^{(i)}) = 0 \end{cases}$$

We can see that $y^{(i)}(\theta^T x^{(i)}) = 1$ iff $y_i = (\boldsymbol{\theta}^T \mathbf{x}^{(i)})$ and $y^{(i)}(\theta^T x^{(i)})$ = -1 iff $y_i \neq (\boldsymbol{\theta}^T \mathbf{x}^{(i)})$.

$\therefore \theta$ is only updated with $y^{(i)}(\theta^T x^{(i)})$ when $y_i \neq (\boldsymbol{\theta}^T \mathbf{x}^{(i)})$.

2. (3 points) **Derivation.** We introduce a modified method of linear regression where we now put different importance on each feature.

We introduce a new $p \times p$ diagonal matrix: Q:

$$\begin{pmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & q_p \end{pmatrix}$$

Each diagonal entry q_i is a quantified "importance" that we give each of the p features.

The new objective function is given as:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \boldsymbol{\theta}^T Q x^{(i)})^2$$

,where $\boldsymbol{\theta} \in \mathbb{R}^k$

Derive the update rule for this new modified regression.

$$\frac{dJ(\theta)}{d\theta} = -QX\sum_{i=1}^{N}((y^{(i)} - \theta^T Q x^{(i)})$$

, where X is a matrix with $x^{(i)}$ as rows.

Given the general update rule:

$$\theta^{(k+1)} \leftarrow \theta^{(k)} + \frac{dJ(\theta)}{d\theta}$$

, the new update rule should be:

$$\theta^{(k+1)} \leftarrow \theta^{(k)} - QX\sum_{i=1}^{N}((y^{(i)} - \theta^T Q x^{(i)})$$

1.10 Neural Networks

Assume you have a Neural Network made up of hidden layer, with 4 neurons all with a Perceptron non-linearity applied and a single output neuron.

3. (2 points) Draw the decision boundaries that your neurons may find on the below graph, be sure to show the orientation of these decision boundaries.

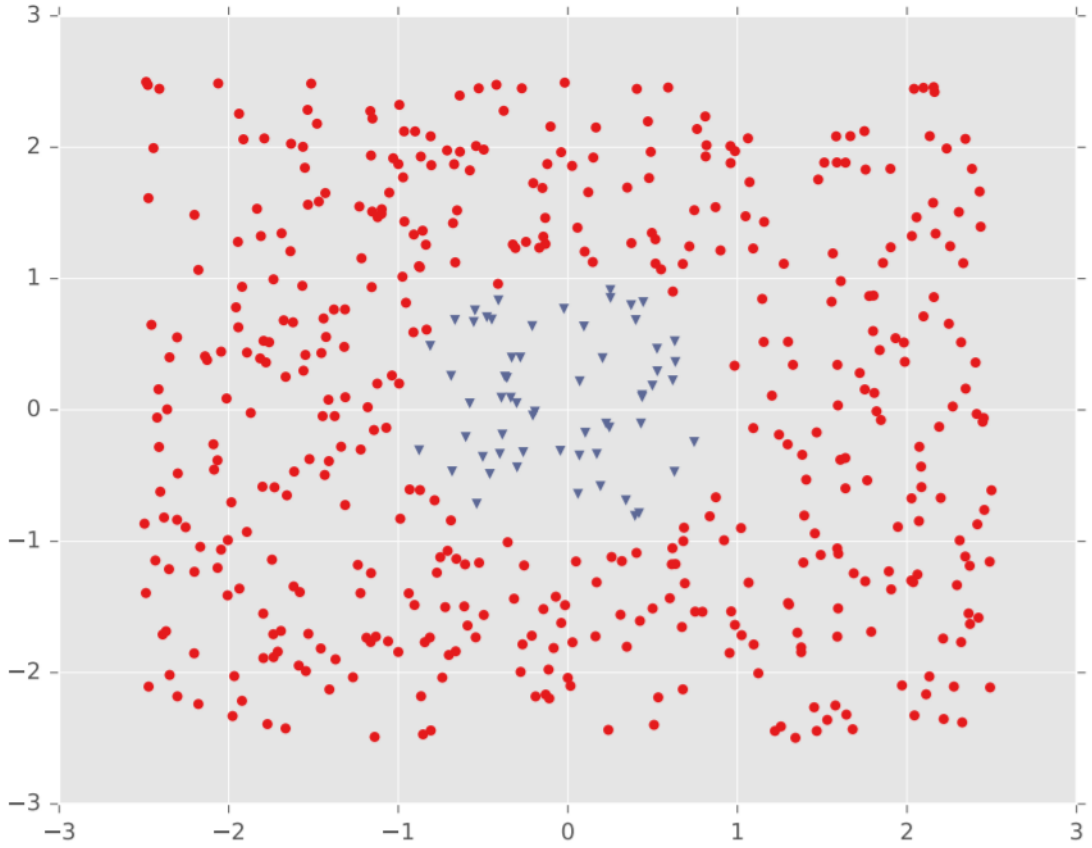

Figure 3: Neural Network Data

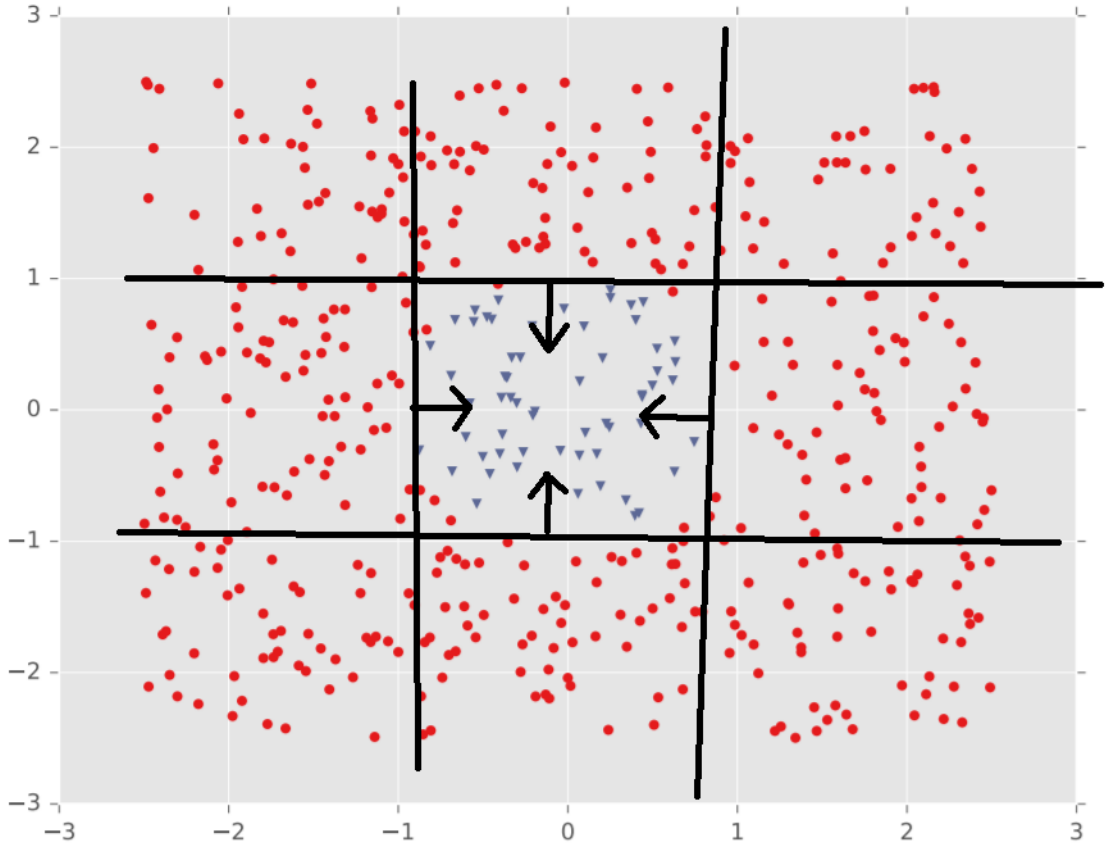

Figure 4: Neural Network Data

4. (2 points) Explain what kind of function the output neuron represents?

   The output neuron represents the function that is all of the hidden neurons in the hidden layer return 1 then the output neuron will return 1. If any of the hidden neurons returns 0 then it will return 0.

5. (2 points) **True or False:** Gradient descent would suitable to find us this solution. Briefly explain your choice.

   ◯ True

   ◯ False

False: Perceptron is not a differentiable function as such Gradient decent would not be suitable.

1.11 HMM's

6. (3 points) Suppose we are Samsung data scientists trying to estimate iPhone X sales of Apple. Since daily sale information is not available, we decide to focus on information we can gather on Apple, such as the daily closing stock price of Apple. To simplify our job, we round the stock price to the nearest multiple of 50s $(0, 50, 100, ...)$ and judging by the historic prices, we can safely assume that the price never exceeds ≤ 300. Knowing a bit of finance, we think that the stock price is determined by the iPhone sales, and the daily sale should be in the range of 50,000 phones, rounded to the nearest 10,000 (so we consider possible daily sales of $10000, 20000, \cdots, 50000$). Seems like HMM is a good model for this job, but in order to account for the dependence of stock price on past prices, we decide to use 2nd-order HMM (recall that second-order HMM means the transition probability is given by $P(s_t|s_{t-1}, s_{t-2})$ instead of $P(s_t|s_{t-1})$; s_t being the latent state at time t).

   (a) (1 point) If our job is to predict iPhone X sales for 10 consecutive days, and we can only observe Apple stock price, how many parameters do we need to learn?

   $|o| = 7, |s| = 5$, prior requires 24, emission requires $5 \times (7 - 1) = 30$, transition requires $5 \times 5 \times (5 - 1) = 100$, total is 154.

   (b) (1 point) Same setup as previous question, but we only need to estimate iPhone sales of today; how many parameters do we need to learn?

   $24 + 100 = 124$, no transition probability needed.

   (c) (1 point) Suppose we have gathered iPhone X sale information for the past 2 days, and we now want to predict sales of next day. How many parameters do we need now?

   130. No prior is needed

1.12 SVMs

Supposed you learned a two class linear SVM for linearly separable input data. Let $\mathbf{w}$ and b be the parameters we obtained for the primal SVM formulation.

In the standard SVM formulation (SVM1) we use the following constraints for all x in class 1:

$$\mathbf{w}^T\mathbf{x} + b \geq 1$$

and for all x in class 0:

$$\mathbf{w}^T\mathbf{x} + b \leq -1$$

Assume that we learned a new SVM model (SVM2) using the following constraints instead, for all x in class 1:

$$\mathbf{w}^T\mathbf{x} + b \geq 0$$

and for all x in class 0:

$$\mathbf{w}^T\mathbf{x} + b < 0$$

7. (2 points) If we compare the margin of SVM2 to that of SVM1 we can say that:

   ◯ The margin increased

   ◯ The margin decreased

   ◯ The margin stayed the same

   ◯ Impossible to tell

   Decrease (2). If we set the threshold at 0 then there will be no margin and since this is a linearly separable dataset the margin will decrease

   Assume that we are using a new SVM, SVM3 which uses $\frac{\mathbf{w}}{2}$ and $\frac{b}{2}$ where $\mathbf{w}$ and b are the parameters learned for SVM1. With these new parameters

8. (2 points) Are we guaranteed that SVM3 would not make any mistakes on the training data? (recall that an SVM classifier determines the class based on the sign of $\mathbf{w}^T\mathbf{x} + b$ where x is the input).

   ◯ Yes

   ◯ No

   Yes. This is a linearly separable problem and everything that was higher than 0 before remains higher now and similarly for lower than 0.

9. (2 points) How would the margin for SVM3 compare to the margin of SVM1?

   ◯ The margin would increase

   ◯ The margin would decrease

   ◯ The margin would stay the same

◯ Impossible to tell

The Margin would increase. The margin is $\frac{2}{\sqrt{(\mathbf{w}^T\mathbf{w})}}$ and since $\mathbf{w}$ is divided by 2 it would increase.

10. (2 points) What can we say about the number of support vectors in SVM3 compared to the number in SVM1?

◯ The number of Support Vectors would increase

◯ The number of Support Vectors would decrease

◯ The number of Support Vectors would stay the same

◯ Impossible to tell

Impossible to tell, it entirely depends on the distribution of x.

11. (2 points) Which of the following do not converge to a classification of the data if the data is not linearly separable (assuming no kernel trick or projection into higher dimensions):

☐ Logistic Regression

☐ Linear soft-margin SVM

☐ Linear hard-margin SVM

☐ Perceptron

☐ Neural network with two hidden layers

C, D: Linear hard-margin SVM and Perceptron cannot classify linearly inseparable data.

12. (2 points) **Select all that apply:** Given a set of input features x, where $x \in \mathbb{R}^n$, you are tasked with predicting a label for y, where $y = 1$ or $y = -1$. You have no knowledge of about the distribution of x and of y. Which of the following methods are appropriate?

☐ Perceptron

☐ k-Nearest Neighbors

☐ Linear Regression

☐ Decision Tree with unlimited depth

☐ None of the Above

Kth Nearest Neighbours and Decision Tree with unlimited depth since these two methods do not making the assumption of linear separation.

1.13 Kernels

13. (2 points) Explain in one short sentence the purpose of using Kernels in Machine Learning.

The purpose is to project your data into a higher dimension in the hopes to find a more simple way of classifying the data.

14. (4 points) Assume we have data $\mathbf{X} \in \mathbb{R}^2$ and we have our kernel ϕ such that for:

$$\mathbf{x} = [a, b], \phi(\mathbf{x}) = \begin{bmatrix} a^2 \\ b^2 \\ \sqrt{2}ab \\ \sqrt{2}a \\ \sqrt{2}b \\ 1 \end{bmatrix}$$

Prove that the kernel function for this transformation is: $k(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^2$

$$\langle \phi(\mathbf{x}_1)\phi(\mathbf{x}_2) \rangle = \begin{bmatrix} a_1^2 \\ b_1^2 \\ \sqrt{2}a_1b_1 \\ \sqrt{2}a_1 \\ \sqrt{2}b_1 \\ 1 \end{bmatrix}^T \begin{bmatrix} a_2^2 \\ b_2^2 \\ \sqrt{2}a_2b_2 \\ \sqrt{2}a_2 \\ \sqrt{2}b_2 \\ 1 \end{bmatrix} ** = (a_1a_2)^2 + (b_1b_2)^2 + 2a_1b_1a_2b_2 + 2a_1a_2 + 2b_1b_2 + 1$$

Similarly: $(\langle x_1, x_2 \rangle + 1)^2 = (a_1a_2 + b_1b_2 + 1)^2 = (a_1a_2)^2 + (b_1b_2)^2 + 2a_1b_1a_2b_2 + 2a_1a_2 + 2b_1b_2 + 1$
hence equivalent.

1.14 K-Means

15. (2 points) Akin to the k-Means algorithm, we can define the k-Medians algorithm and the k-Modes algorithm (where we partition observations into k clusters in which each observation belongs to the cluster with the nearest median and mode respectively). For the same dataset and value of k, which of these methods is most sensitive to outliers?

- ◯ k-Means
- ◯ k-Medians
- ◯ k-Modes

k-Means. Median and mode are less sensitive to outliers for any dataset.

16. (2 points) Say we run the k-Means algorithm with $k = 5$ on a 2-dimensional dataset, i.e. each datapoint is of the form (x_1, x_2). The cluster centers will be in a straight line if the correlation between x_1 and x_2 is:

    ◯ 0

    ◯ 0.25

    ◯ 0.5

    ◯ 1

    Correlation = 1. If all data points are in a straight line, the cluster centers will also be along that line.

1.15 PCA

17. (2 points) Which of the following is/are good way(s) to reduce dimensionality of a dataset?

    ☐ Removing a column of the design matrix which is identical to another column

    ☐ Removing a column of the design matrix where almost all values are missing

    ☐ Removing a column of the design matrix which has high variance

    ☐ PCA

    A, B, D. Removing a column which has lots of missing values, or is linearly dependent with another column, as well as PCA, are all valid dimension reduction techniques.

18. (2 points) Which of the following is/are true about PCA?

    ☐ PCA searches for directions that have the least variance

    ☐ The maximum number of principal components is always at most the number of features in the original dataset

    ☐ Every principal component is orthogonal to every other principal component

    ☐ If we reduce our dataset into r dimensions, a lower value of r implies more regularization

    B, C, D. A: PCA searches for directions that have the greatest variance. B and C are true. D: Lower the value of r, more is the smoothening as we preserve fewer characteristics in data.