CS 445/545: Machine Learning

Portland State UNIVERSITY

Midterm (49 pts. Possible),

Spring 2024 Rhodes

SRIHARI TANMAY KARTHIK TADALA

PSU ID - 918597835

---

Please show a sufficient amount of work for full credit on exercises. If the question prompts you to simply provide an answer, it is acceptable to simply submit the answer (without explanation, if none is necessary). The exam is open book and open notes. **However, you are not allowed to confer with fellow students or, nor are you permitted to use "Google" to seek out a solution**.

*Email your exam solutions to our grader by the assigned due date. You can submit typed or hand-written solutions (or a combination of both if this is preferred); please make an effort to ensure that your solutions are clear and legible.

---

1. (1 pt.) Soft-margin SVMs, defined with slack variables $\xi_i$ always admit of a solution. **True False**

**True**

2. (1 pt.) A zero training set error necessarily indicates good generalization performance. **True False**

**False**

The efficiency of a model is determined by its ability to accurately predict outcomes on new, unfamiliar data, rather than solely relying on its similarity to the data it was trained on.

3. (1 pt.) Recall the general expression for backprop weight updates: $\Delta w_{ji}^{t} = \eta \delta_{j} x_{ji} + \alpha \Delta w_{ji}^{t-1} - \lambda w_{ji}^{t-1}$. Explain the role of the following terms (just in a sentence):

$\alpha \Delta w_{ji}^{t-1}$:-

$\lambda w_{ji}^{t-1}$:-

Solution-

The general expression for backpropagation weight updates with weight

decay (L2 regularization) can be represented as follows:

$$\Delta w_{ji}^{t} = \eta \delta_{j} x_{i} + \alpha \Delta w_{ji}^{t-1} - \lambda w_{ji}^{t-1}$$

Where,

- [Δωj(t)]: This represents weight updates for weight "i" at time "t". It's the change that will be applied to the weight to update its value.
- [η]: The learning rate, which controls the step size during weight updates.
- [∂jxi]: This term represents the gradient of the loss function "j" with respect to the input "xi."

It quantifies how much the loss changes concerning the input.

- [αΔωji(t-1)]: The term "α" represents the momentum coefficient. Δωji(t-1) represents the change in weight at the previous time step. Momentum in the weight update takes into account the direction and velocity of previous weight changes, which aids in achieving convergence.
- [λωji(t-1)]: The symbol "λ" denotes the weight decay or L2 regularization term. It imposes a penalty on significant weights by deducting a portion of the weight "ji" from the previous time step "t-1." Weight decay mitigates overfitting by promoting the regularization of weight values towards smaller magnitudes.

This formula combines the gradient of the loss with respect to the input, momentum from the previous weight update, and weight decay to calculate the weight update.

4. (1pt.) The PLA (perceptron learning algorithm) always converges in a finite number of steps. **True** **False**

**False**

5. (1 pt.) Given a finite data set, there exists a finite dimensional vector space for which the data is linearly separable. **True False**

**True**

6. (1 pt.) A model with infinite VC dimension can have a finite number of parameters.
**True** **False**

**True**

7. (1 pt.) Suppose we wish to calculate $P(H|X_1, X_2)$ and we have no conditional independence information. Which of the following sets of numbers are sufficient for the calculation? (circle)

(i) $P(X_1, X_2)$, $P(H)$, $P(X_1|H)$, $P(X_2|H)$

(ii) $P(X_1, X_2)$, $P(H)$, $P(X_1, X_2|H)$

(iii) $P(H)$, $P(X_1|H)$, $P(X_2|H)$

8. (1 pt.) Consider the sigmoid function: $f(x) = \frac{1}{1+e^{-x}}$. Which expression is equal to $f'(x)$ ? (circle)

(i) $f(x)\log(1-f(x))$

(ii) $f(x)(1-f(x))$

(iii) $\frac{1}{f(x)}$

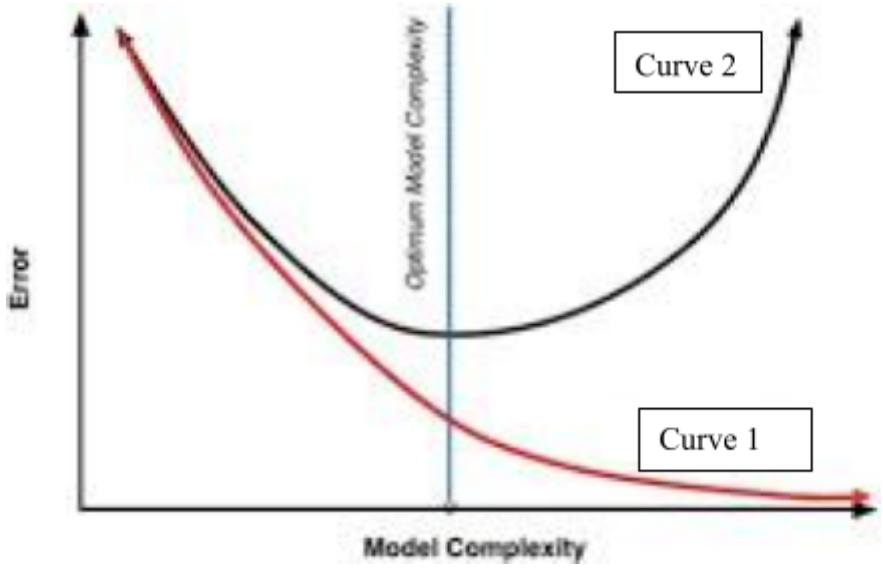
(iv) $\aleph_0$

Option 2 $f(x)(1-f(x))$

9. (1 pt.) Suppose you are given a dataset of cellular images from patients with and without cancer and that you are required to train a classifier that predicts the probability that the patient has cancer. The dataset has 900 cancer-free images and 100 images from cancer patients. If I train a classifier which achieves 95% accuracy on this dataset, should we consider this to be a good classifier? Explain your answer in a sentence or two.

Ans) Even if the classifier attains a 95% accuracy rate, it may still produce inaccurate outcomes due to the inadequate distribution of the dataset. With only 100 cancer images and 900 images of cancer-free patients, the dataset is not well-balanced.

10. (1 pt.) The plots of training and test error are shown as a function of model complexity below. Appropriately identify the plots.




Training Error: Curve 1 Curve 2 (circle)

Test Error: Curve 1 Curve 2

11. (2 pts.) Suppose that the prevalence of a disease is 2%. This disease can be screened by a medical test that is 85% accurate. This means that the test result is positive about 85% of the times when it is applied on patients who have the disease and that the test result is negative about 85% of the time when it is applied on patients who do not have the disease. Suppose that you take the test and the test shows a positive result. How likely is it that you have the disease?

11) $H_1$ be the event of having disease
$\neg H$ be the event of not having disease
$D+$ be the event of testing positive
$D-$ be the event of testing negative
Here we can find the possibility of having the disease given the positive test results which is $P(H/\neg H)$

Probability of having disease $P(H) = 0.02$
Probability of not having disease $P(\neg H) = 0.98$
Probability of getting positive of having the disease $P(D+/H) = 0.85$.
Probability of testing negative of not having a disease $P(D-/\neg H) = 0.85$.

From Bayes Theorm :- $P(H/D+) = \frac{P(D+/H) \times P(H)}{P(D+)}$

To find the value of $P(D+)$

$$P(D+) = P(D+/H) * P(H) + P(D+/\neg H) * P(\neg H)$$

The value of $P(D+/\neg H) = 1 - P(D-/\neg H) = 1 - 0.85 = 0.15$.

So $P(D+) = P(D+/H) * P(H) + P(D+/\neg H) * P(\neg H)$
$= 0.85 \times 0.02 + 0.15 \times 0.98$
$= 0.164$

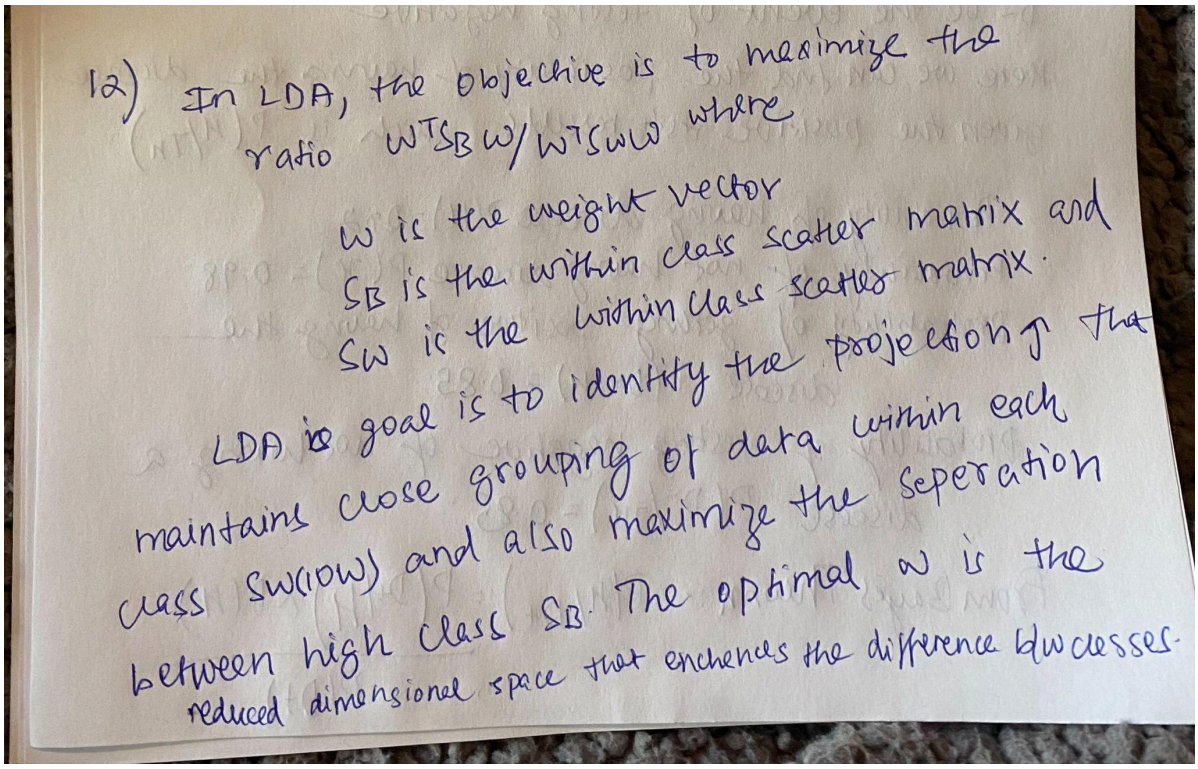
From bayes theorm $(P(H/D+) = \frac{P(D+/H) * P(H)}{P(D+)}$

$= \frac{0.85 \times 0.02}{0.164} = 0.103$

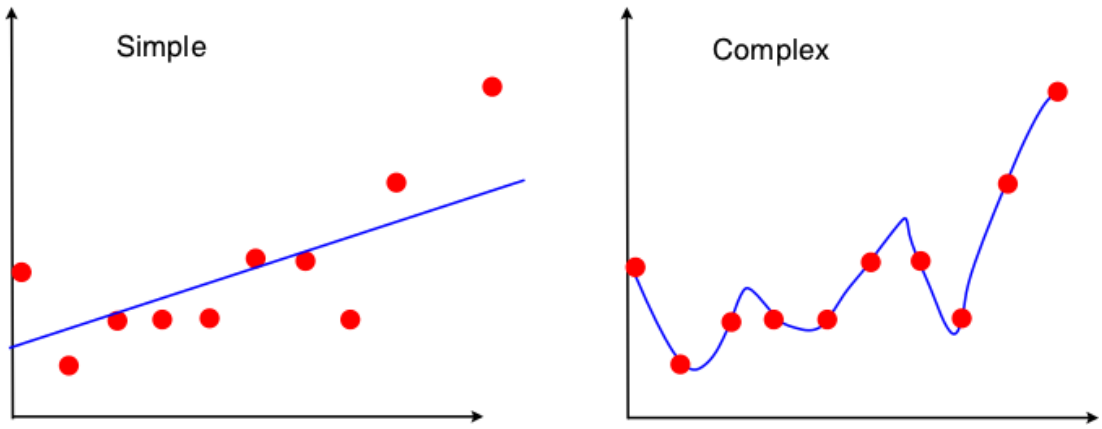
The chances of having disease is 10.3%.

12. (2pts.) Recall that the with LDA dimensionality reduction we wish to solve the following

optimization problem: $\arg\max_{w} \frac{w^T S_B w}{w^T S_W w}$. In a few simple sentences, explain the meaning of this expression with respect to LDA.




13. (2 pts.) Using the following mathematical models (i.e. the curves shown), explain briefly the idea of the "*bias-variance tradeoff*" in machine learning, how it relates to model *complexity* and the notions of *overfitting* and *underfitting*. In relation to these ideas, elaborate on what is means to have a "good" predictive model.



The diagram on the left depicts a model that is underfitting, meaning it does not accurately represent the trend in the data points. This model shows a high level of bias and a low level of variance. Conversely, the diagram on the right illustrates the phenomenon of overfitting, characterized by a model that exhibits low bias but high variance. This means that the model captures excessive noise from the data. An effective predictive model is one that achieves a good balance between bias and variance, resulting in minimal error. This model effectively captures the fundamental attributes of the data while also demonstrating strong performance on new, unseen data.

14. (1 pt.) "t-SNE" is an example of which type of general ML algorithm: (circle)

(i) classification (ii) regression (iii) dimensionality reduction (iv) backpropagation

(iii) dimensionality reduction

15. (2 pts.) Let $\mathbf{x} = (x_1, x_2)$. Using the feature mapping

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

show that

$$\Phi((2, 3)) \cdot \Phi((4, 4)) = ((2, 3) \cdot (4, 4))^2$$

15. $x=(x_1,x_2)$ $y=(y_1,y_2)$

there we have, $\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

we need to substitute (2,3) and (4,4) in the equation to get,

$$\phi(2,3) * \phi(4,4) = (2^2, \sqrt{2}(2)(3), 3^2)$$
$$= (4, \sqrt{2}(4)(4), 4^2)$$

$$\phi(2,3) \cdot \phi(4,4)) = 64 + 192 + 144$$
$$= 400.$$

Now, to prove it, we shall calculate the value

$$\phi((2,3)(4,4))^2 = (8+12)^2 = 20^2 = 400$$

They are equal and Hence proved!!!

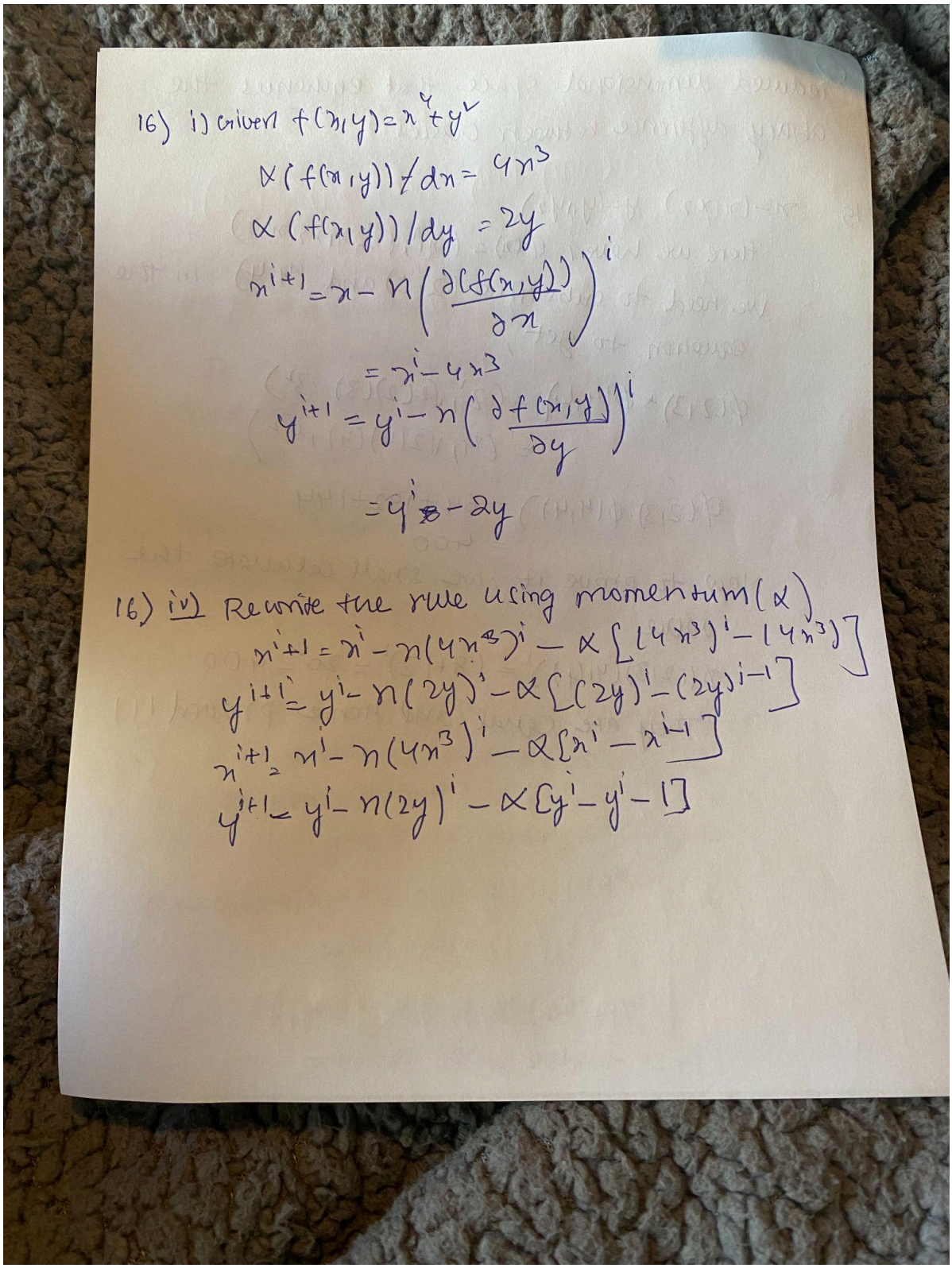
16. (5 pts.) ***Gradient Descent***. Consider the multivariate function: $f(x, y) = x^4 + y^2$

Devise an iterative rule using gradient descent that will iteratively move closer to the minimum of this function. Assume we start our search at an arbitrary point: $(x_o, y_o)$. Give your update rule in the conventional form for gradient descent, using $\eta$ for the learning rate.

(i) Write the explicit x-coordinate and y-coordinate updates for step (i+1) in terms of the x- coordinate and y-coordinate values for the ith step.

$x^{(i+1)} \leftarrow$

$y^{(i+1)} \leftarrow$



16) i) Given $f(x,y) = x^4 + y^2$

$\propto (f(x,y))/dx = 4x^3$

$\propto (f(x,y))/dy = 2y$

$x^{i+1} = x - n\left(\frac{\partial(f(x,y))}{\partial x}\right)^i$

$= x^i - 4x^3$

$y^{i+1} = y^i - n\left(\frac{\partial f(x,y)}{\partial y}\right)^i$

$= y^i - 2y$

16) iv) Rewrite the rule using momentum ($\propto$)

$x^{i+1} = x^i - n(4x^3)^i - \propto [(4x^3)^i - (4x^3)]$

$y^{i+1} = y^i - n(2y)^i - \propto [(2y)^i - (2y)^{i-1}]$

$x^{i+1} = x^i - n(4x^3)^i - \propto [x^i - x^{i-1}]$

$y^{i+1} = y^i - n(2y)^i - \propto [y^i - y^{i-1}]$

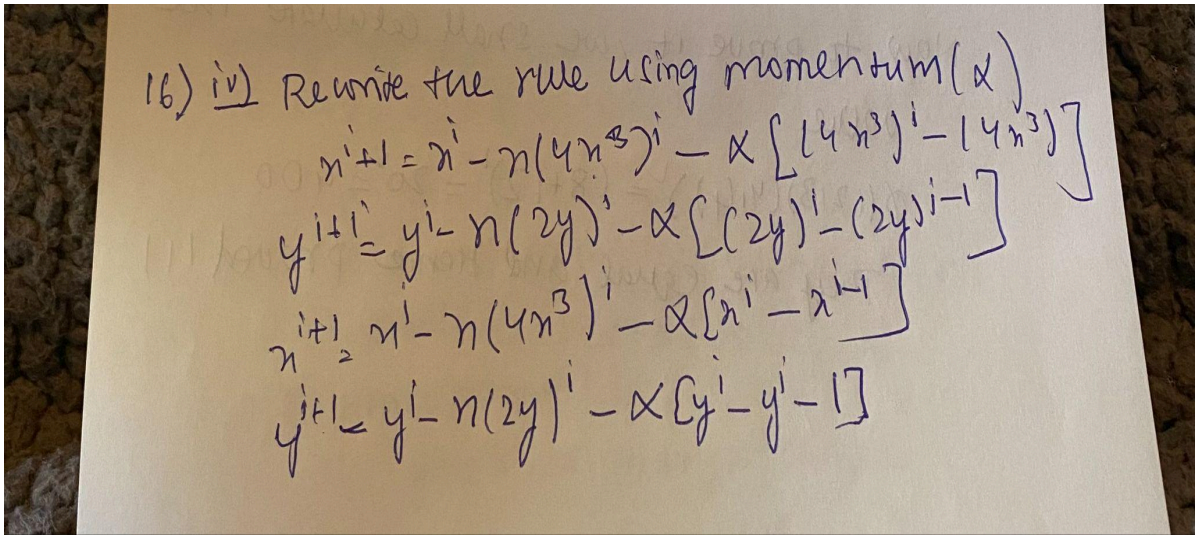
(ii) Briefly explain how G.D. works, and the purpose of the learning rate.

Gradient Descent is an iterative algorithm employed to minimize a cost function by identifying the direction of steepest descent. Initially, the weights are assigned arbitrary values and subsequently adjusted through incremental changes referred to as the learning rate. The mentioned rate influences the magnitude of the increments made in each update to approach the local minimum. Augmenting the magnitude of the steps can accelerate the convergence process, although there is a potential risk of exceeding the intended objective. Conversely, a decreased learning rate results in a slower and more gradual progression towards the local minimum.

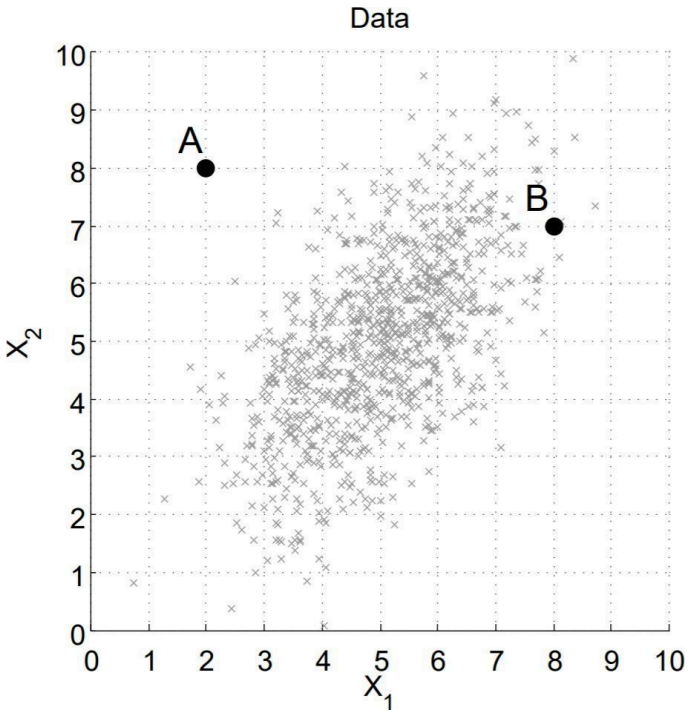
(iii) Is your algorithm guaranteed to converge to the minimum of $f$ (you are free to assume that the learning rate is sufficiently small)? Why or why not?

**Answer**: Yes, it will converge to the minimum of f if we use a small learning rate because it has only one local minima which is the global minimum.

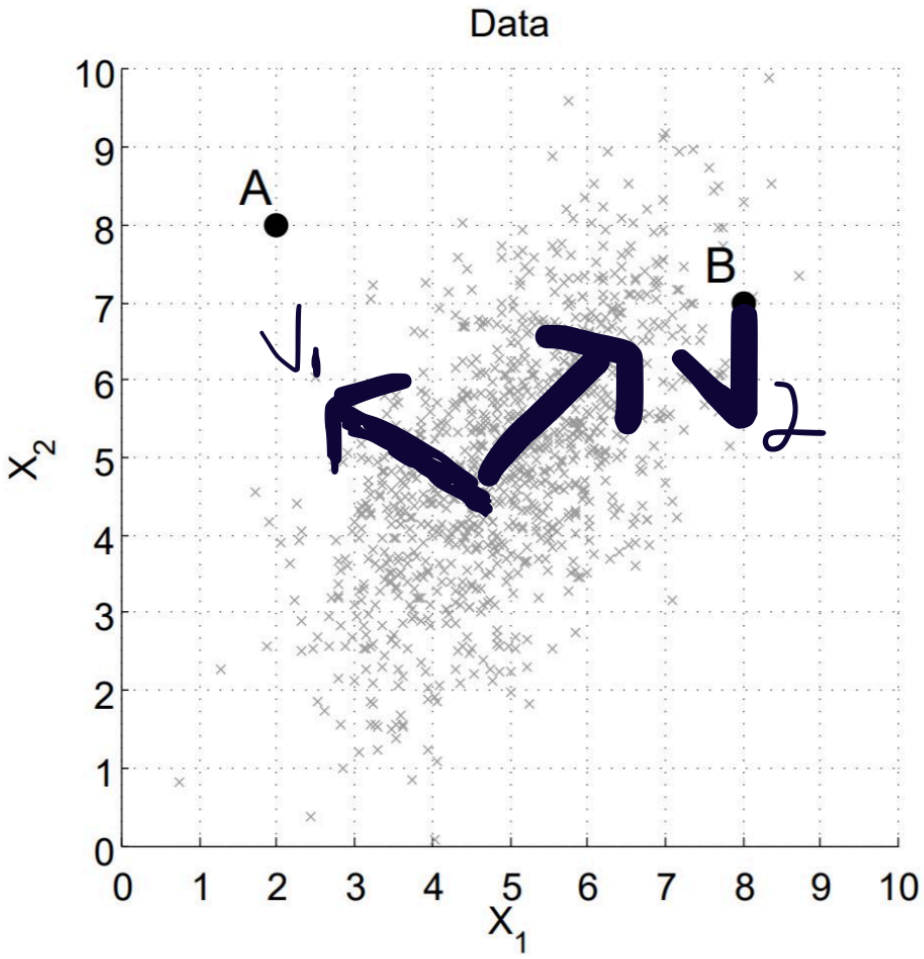
(iv) Re-write your rule from part (i) with a momentum term, including a momentum parameter α.



17. (3 pts.) *PCA*. The plot below shows a sample drawn from a two dimensional multivariate Normal (Gaussian) distribution. Define vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ as the directions of the first and second principal components, after applying PCA to the dataset, where $\|\mathbf{v}_1\| = \|\mathbf{v}_2\| = 1$.




(i) Sketch and label $\mathbf{v}_1$ and $\mathbf{v}_2$ in the figure above. The arrows should originate from the mean of the distribution. You do not need to compute the actual PCA procedure, instead simply visually estimate the directions of the arrows.

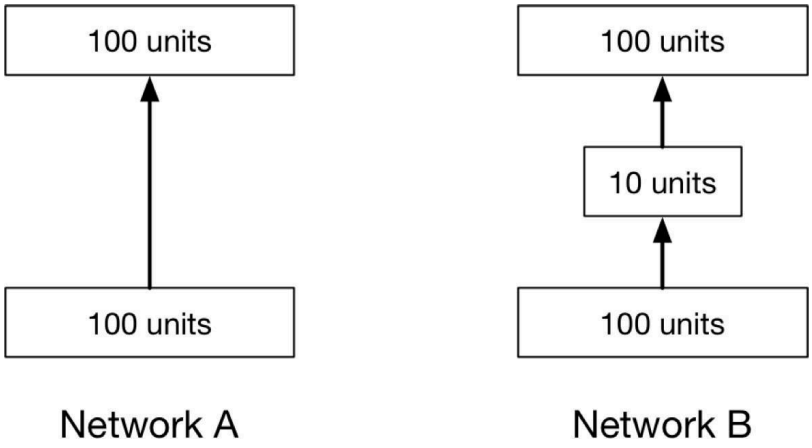



(ii) Which point (A or B) would have the higher reconstruction error after projecting onto the first principal component direction v1? Circle one:

Point A Point B

**Answer**: Point A

18. (2 pts.) Consider the following two MLPs, where all of the layers use linear activation functions.




(i) Give one advantage of Network A over Network B.

Implementing Network A is easier than Network B due to its smaller number of units and lower computational complexity.

(ii) Give one advantage of Network B over Network A.

Network B has the capacity to acquire a greater number of features compared to Network A. Thus, the likelihood of overfitting is lower compared to Network A.

19. (2 pts.) Suppose that we have trained an MLP on the MNIST dataset, and pass a datum **x** (the encoding of a handwritten digit) through our network and the corresponding output neuron activations consist of the following values:

{0.10, 0.40, 0.45, 0.10, 0.20,0.90,0.05,0.20, 0.10, 0.35}

Apply the softmax transformation to this set of activations and list the transformed values (preserving the order of the original activations).

In practice, what is the purpose of using the softmax transformation?

(3)

(9) We know that

$$y_n = g(h_n) = \frac{\exp(h_n)}{\sum_{k=1}^{N} \exp(h_k)}$$

{0.10, 0.40, 0.45, 0.10, 0.20, 0.90, 0.05, 0.20, 0.10, 0.35}

$\exp(0.10) = 1.1052$

$\exp(0.40) = 1.4918$

$\exp(0.45) = 1.5683$

$\exp(0.10) = 1.1052$

$\exp(0.20) = 1.2214$

$\exp(0.30) = 2.4596$

$\exp(0.05) = 1.0513$

$\exp(0.20) = 1.2214$

$\exp(0.10) = 1.1052$

$\exp(0.35) = 1.4191$

$$\sum_{k=1}^{N} \exp(h_k) = \exp[0.10 + 0.40 + 0.45 + 0.10 + 0.20 + 0.98 + 0.05 + 0.20]$$

$$\exp[0.10] + \exp[0.35]$$

$$= 13.7485$$

$$\text{softmax } h(x) = \left[ \frac{1.1052}{13.748}, \frac{1.4318}{13.74}, \frac{1.5683}{13.7485}, \right.$$

$$\frac{1.1052}{13.748}, \frac{1.2214}{13.7485}, \frac{1.0513}{13.7485}, \frac{1.2214}{13.7485}, \frac{1.1052}{13.748}, \left. \frac{1.4191}{13.7} \right]$$

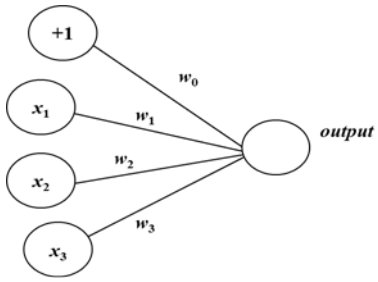
Softmax(h(x)) = [0.0804, 0.1085, 0.1140, 0.0804, 0.0888,
0.4789, 0.0765, 0.0888, 0.0804, 0.103]

The softmax functions rescales the outputs by calculating the exponential of the inputs to the neuron.

It is basically used for normalizing values which helps in predicting the probability of each class

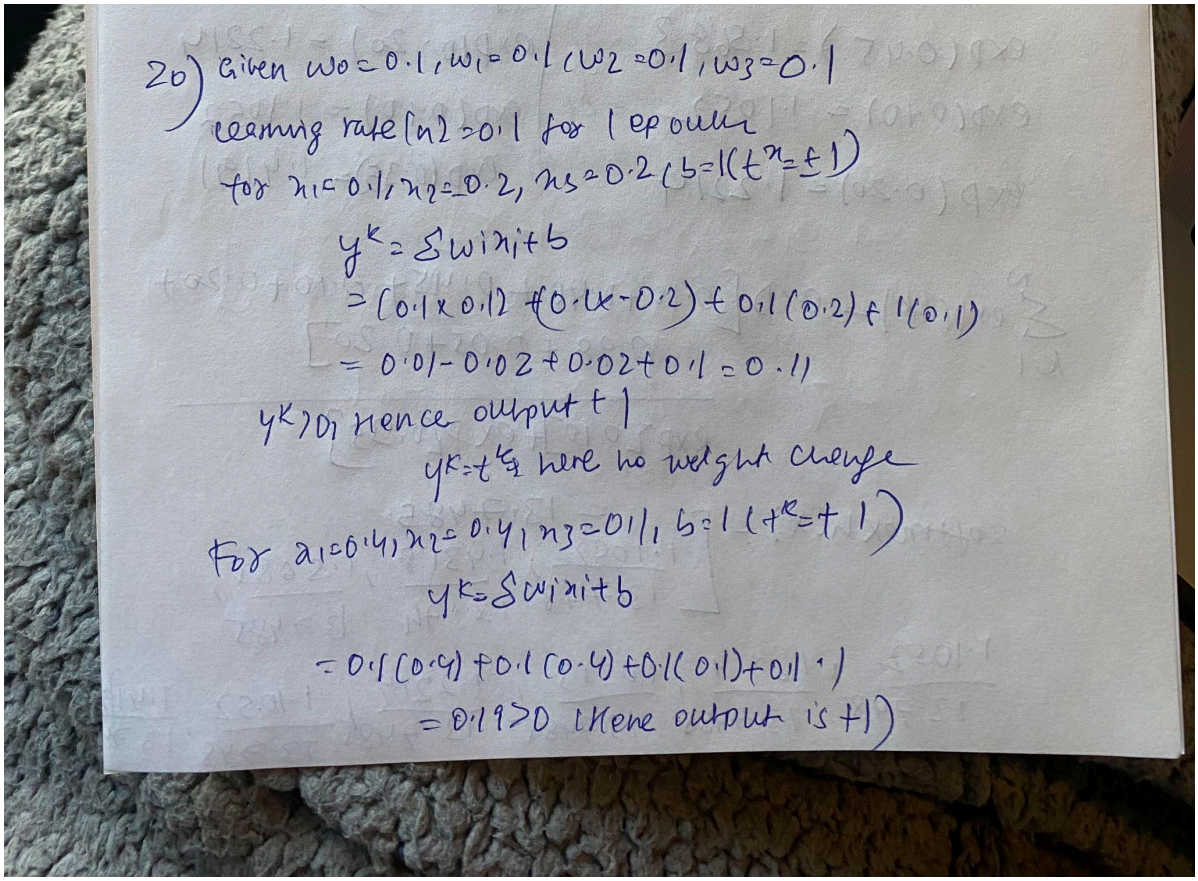
20. (4 pts) Consider the training set, perceptron, and initial weights given below.

| $x_1$ | $x_2$ | $x_3$ | Class |
|---|---|---|---|
| .1 | −.2 | .2 | +1 |
| .4 | .4 | .1 | +1 |
| −.3 | .1 | .3 | −1 |
| −.2 | .2 | −.1 | −1 |




Initial Weights:
$w_0 = .1$
$w_1 = .1$
$w_2 = .1$
$w_3 = .1$

(i) Using stochastic gradient descent (i.e., update the weights after processing each training example), and learning rate $\eta = 0.1$, simulate, by hand, the perceptron learning algorithm for one epoch, showing the new values of the weights after completing the epoch.



(4)

$y^k = t^k$ (weight change not required here also)

For $x_1 = 0.3, x_2 = 0.1, x_3 = 0.3, b = 1$ ($t^k = -1$)

$y^k = \sum w_i x_i + b$

$= 0.1(0.3) + 0.1(0.1) + 0.1(0.3) + 0.1$

$= 0.16 > 0$

perceptron fires +1

$y^k \neq t^k$, update weights

$\Delta w = 0.1(-1-(1))(1) = 0.1(-2)$

$\boxed{\Delta w_0 = -2}$

$\Delta w_1 = (0.1)(-2)(-0.3) = -0.06$

$\Delta w_2 = (0.1)(-1-1)(0.1) = -0.02$

$\Delta w_3 = (0.1)(-1-1)(0.3) = -0.06$

New weights are

$w_0 = w_0 + \Delta w_0 = 0.1 - 0.2 = 0.1$

$w_1 = w_1 + \Delta w_1 = 0.1 + 0.06 = 0.16$

$w_2 = w_2 + \Delta w_2 = 0.1 - 0.02 = 0.08$

$w_3 = w_3 + \Delta w_3 = 0.1 - 0.06 = 0.04$

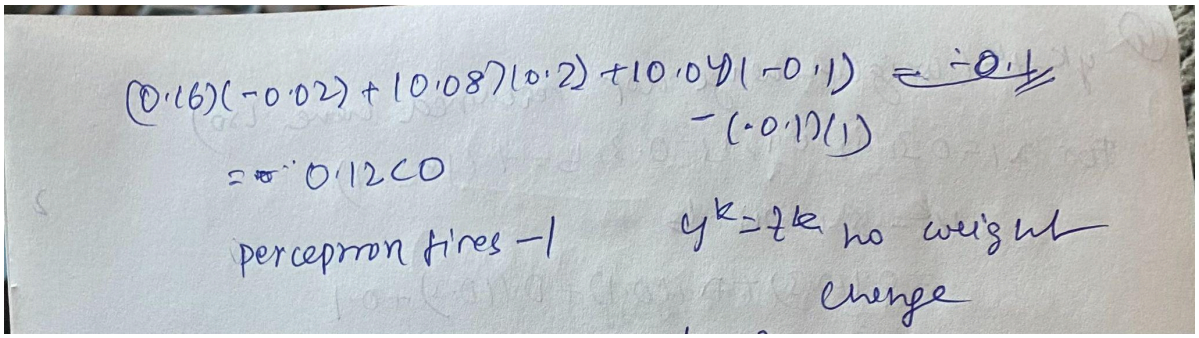
Now consider the 4th output

$x_1 = 0.2, x_2 = 0.2, x_3 = 0.1, b = 1$ ($t_k = -1$)

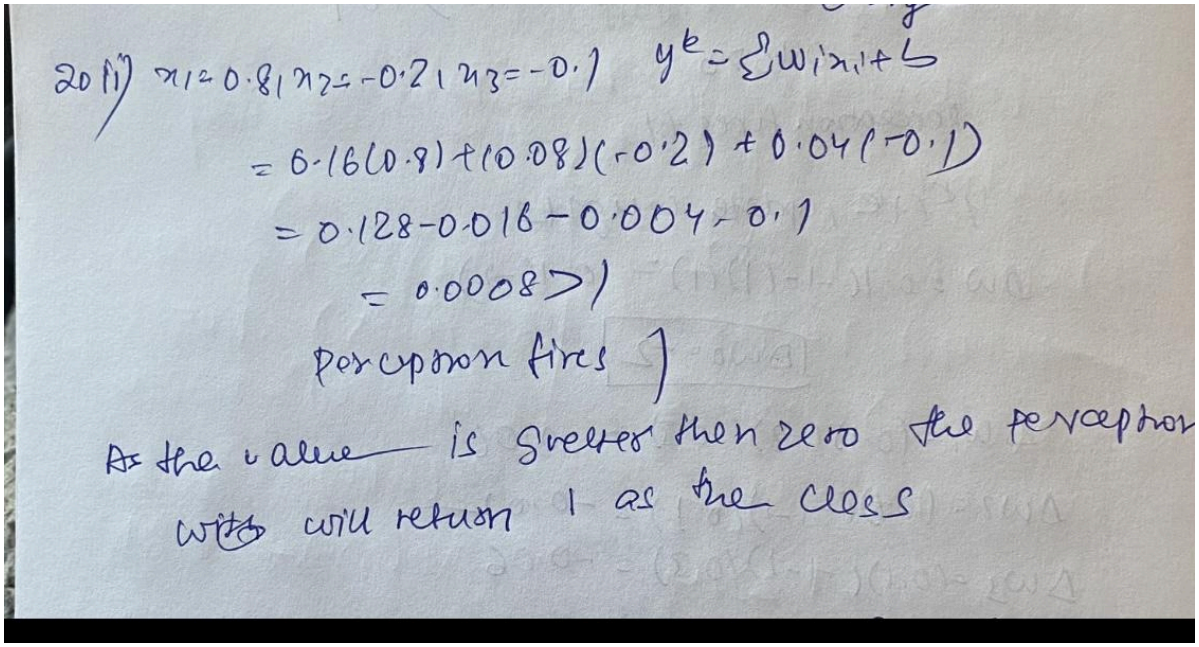
$y^k = \sum w_i x_i + b$

2

1




(ii) At the end of the epoch, how would your resulting perceptron classify a new example with $x_1 = 0.8$, $x_2 = -0.2$, $x_3 = -0.1$ ?

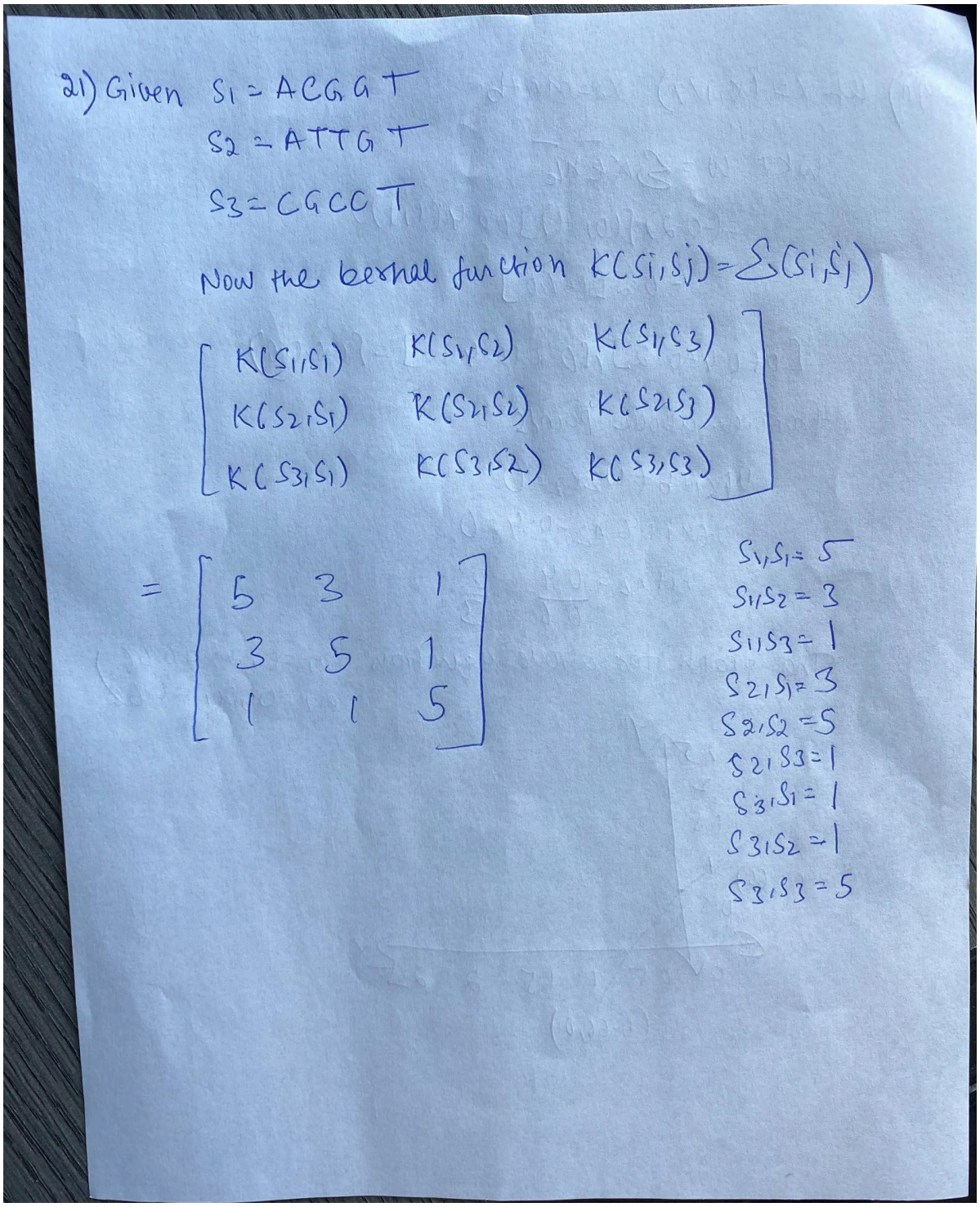



21. (2 pts.) Suppose you have the following short DNA sequences in your training set:

$s_1$ = ACGGT $s_2$ = ATTGT $s_3$ = CGCCT

Suppose you are using these to train an SVM with a "match-count" kernel, where $k(s_i, s_j)$ returns the number of locations strings $s_1$ and $s_2$ at which the symbols match.

Give the Kernel matrix $K$ for this training set and kernel function.

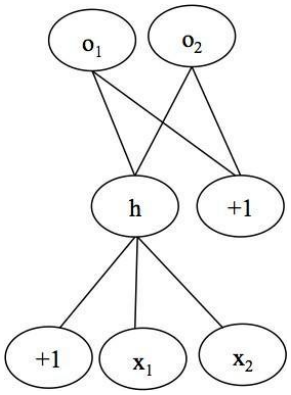



21. (2 pts.) Explain in a few sentences why kernel functions allow SVMs to classify non- linearly separable problems in an efficient way.

Kernel functions enable SVMs to efficiently classify nonlinearly separable problems by transforming the dataset into a higher-dimensional space. In this transformed space, the previously inseparable classes become linearly separable, allowing for the use of a linear

classifier. The effectiveness of this approach heavily depends on the selection of the kernel function, as it determines the characteristics of the high-dimensional space in which the data is organized, thereby influencing the performance of the SVM

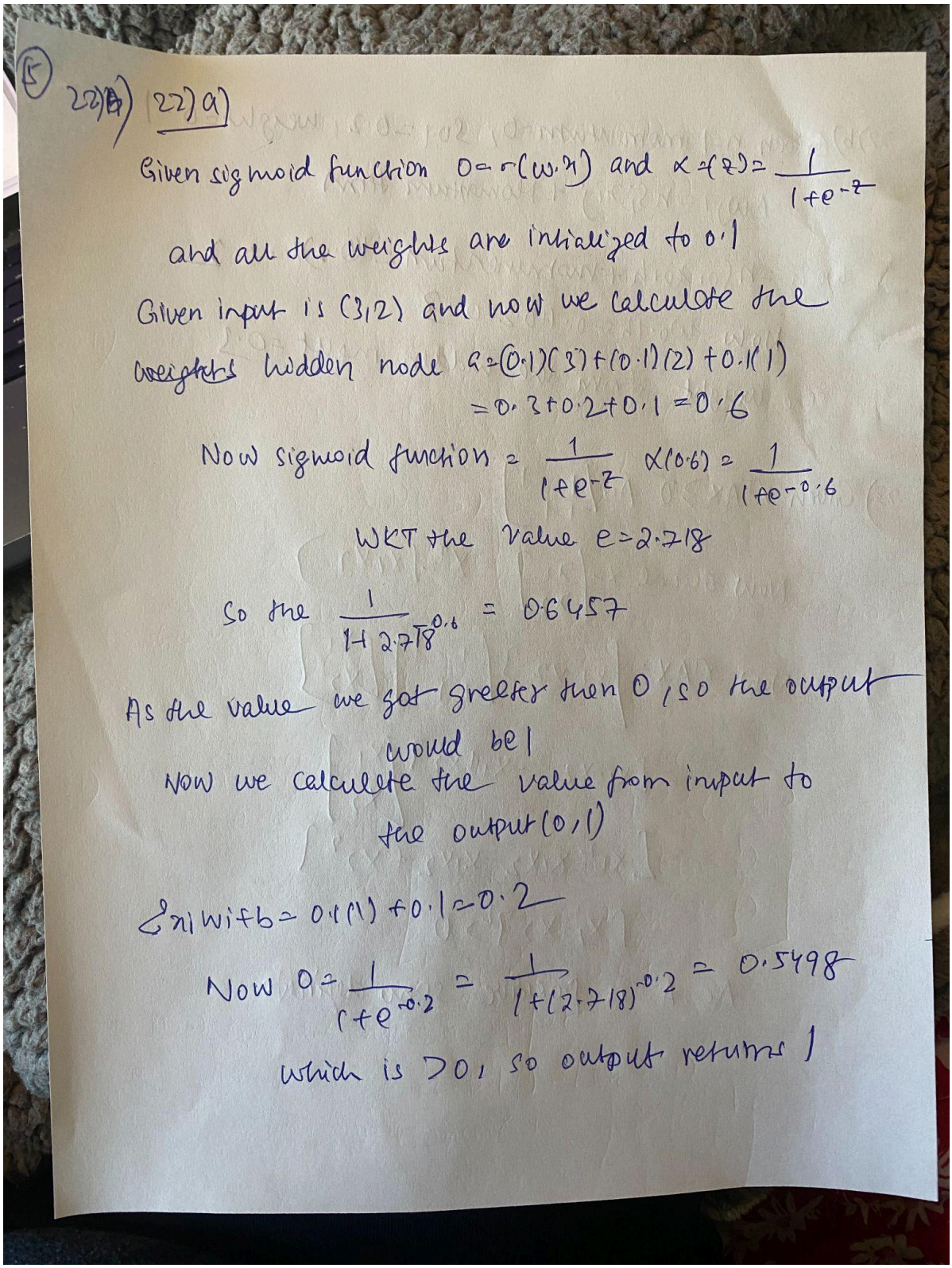
22. (4 pts.) Consider the multilayer neural network given below.




Suppose all the weights are initialized to 0.1. Assume the sigmoid activation function for hidden and output nodes:

$$o = \sigma(\boldsymbol{w} \cdot \boldsymbol{x}), \text{ where } \sigma(z) = \frac{1}{1 + e^{-z}}.$$

(a) Given input **x** = (3,2), what is the activation at output node $o_1$?



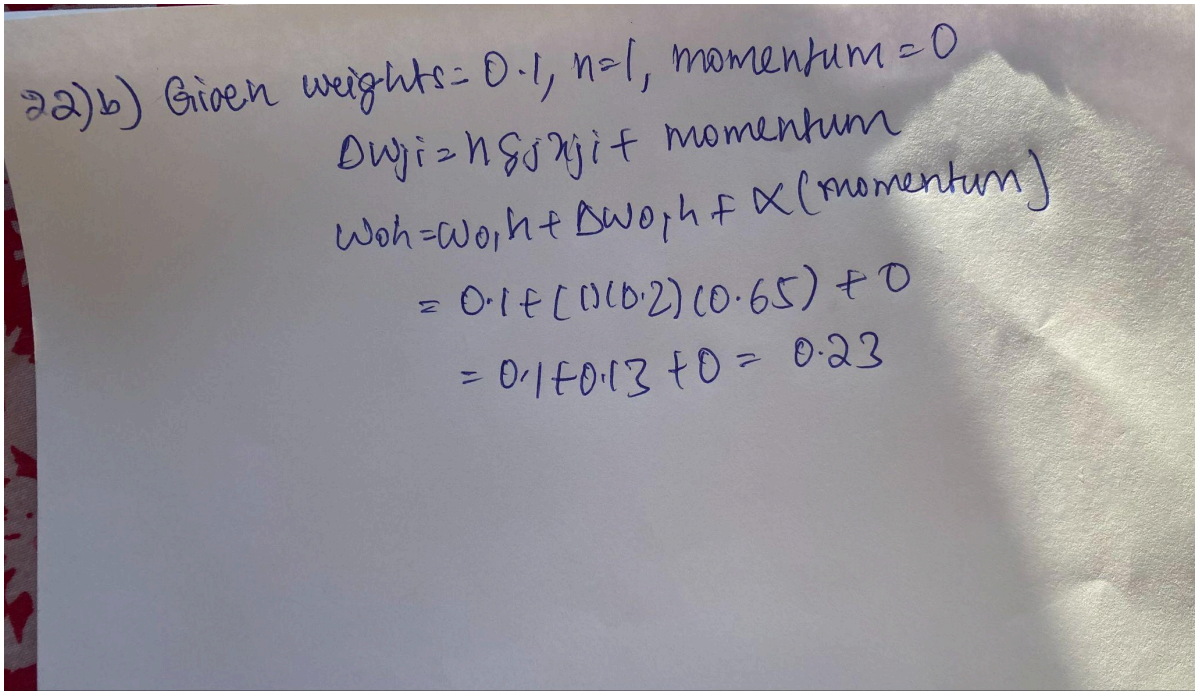

(b) Recall that the weight update rule for backpropagation is

$$\triangle w_{ji} = \eta \delta_j x_{ji} + \textit{momentum-term}$$

Let $\eta = 1$ and *momentum* = 0.

Suppose that after the weights are initialized to 0.1 and the input in part (a) is given, the error term for $o_1$ is calculated to be $\delta_{o1} = 0.2$.

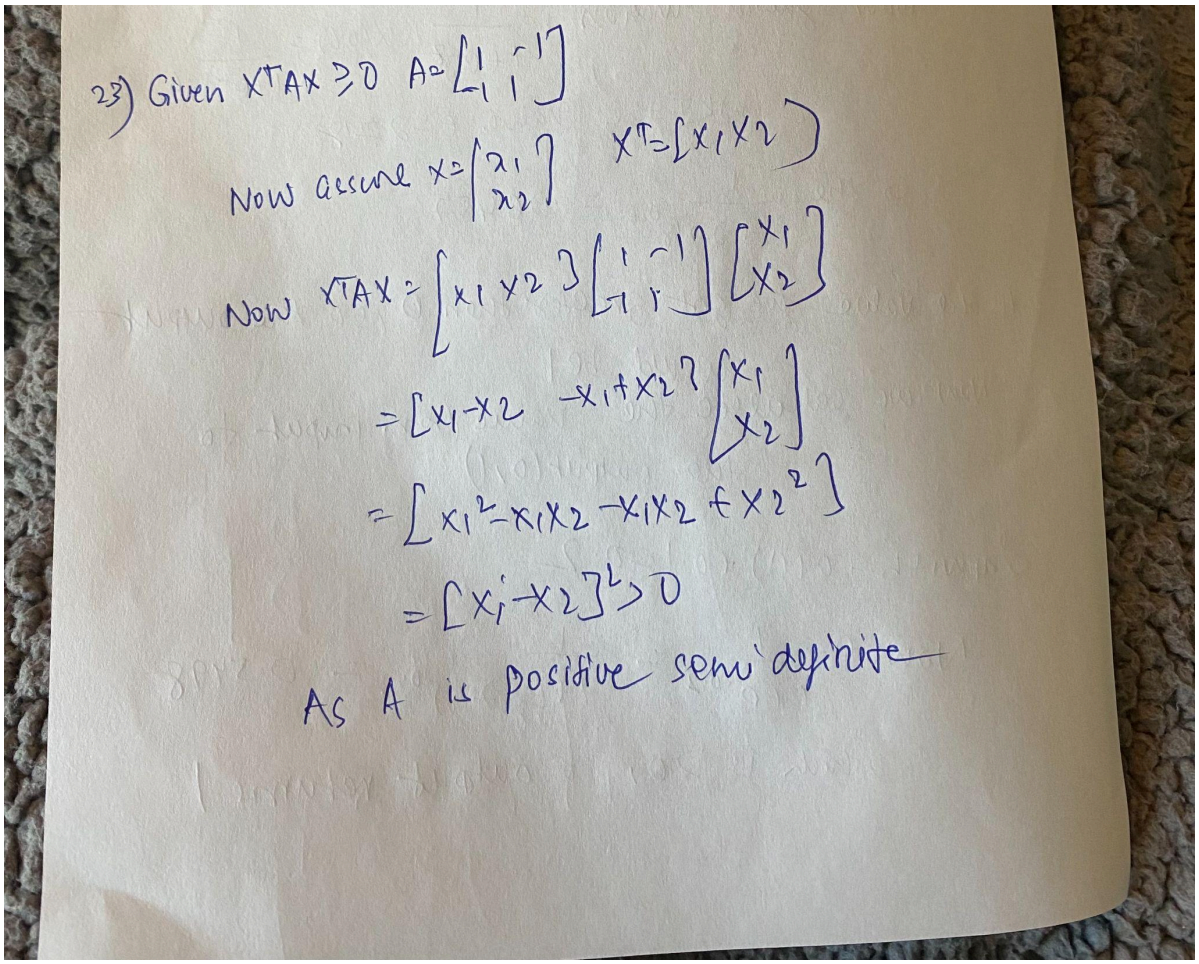
What is the new value of $w_{o1h}$, the weight from the hidden unit to $o_1$?



23. (3 pts.) **Definition**. A real-valued symmetric matrix A is positive semi-definite if:

$\mathbf{x}^T A\mathbf{x} \geq 0$ for all real-valued vectors $\mathbf{x} \neq \mathbf{0}$ .

(i) Let's consider a simple matrix as an example; let $A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$. Prove that A in this case is positive semi-definite. In other words, you need to show that the necessary condition holds in all cases.




(ii) Briefly, explain the significance of positive semi-definite matrices in relation to the *Mercer's Theorem* in ML.

Positive semi-definite matrices play a critical role in machine learning, as emphasized by Mercer's Theorem. This theorem is fundamental for assessing the appropriateness of kernel functions in support vector machines (SVMs) and other related applications. According to Mercer's Theorem, a kernel function is deemed suitable for use if the resulting kernel matrix is positive semi-definite. The condition is crucial as it guarantees the mathematical integrity

of the kernel method, enabling the creation of dependable and efficient machine learning models. The effectiveness of the kernel method, which is essential for producing accurate machine learning outcomes, depends on the utilization of specific matrices to ensure the validity of the underlying mathematics.

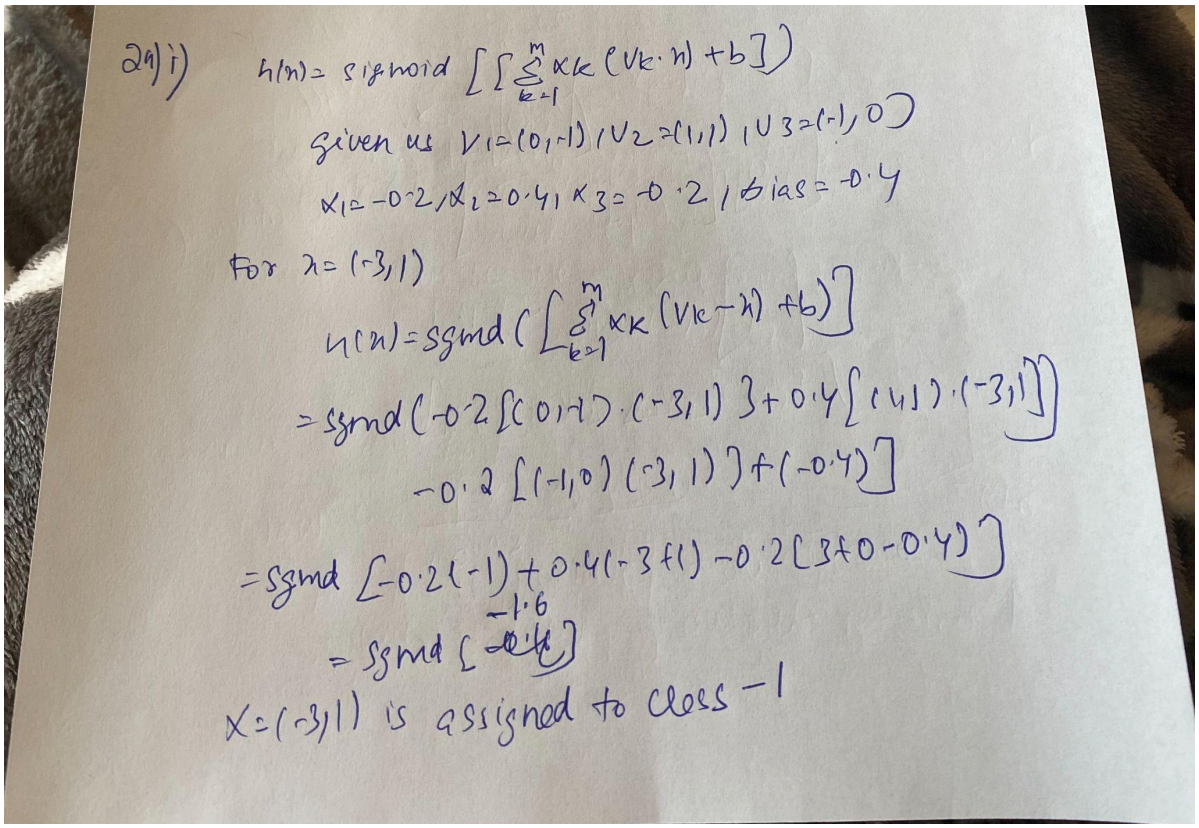
24. (3 pts.) Recall that the classification formula used by support vector machines is

$$h(\mathbf{x}) = \text{sgn}\left(\sum_{k=1}^{m} \alpha_k (\mathbf{v}_k \cdot \mathbf{x}) + b\right)$$

for support vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$.

Let $\mathbf{v}_1 = (0, -1)$, $\mathbf{v}_2 = (1, 1)$, and $\mathbf{v}_3 = (-1, 0)$ be three support vectors defining a separating line, and let the corresponding coefficients and bias be $\alpha_1 = -.2$, $\alpha_2 = .4$, $\alpha_3 = -.2$, and $bias = -.4$.

(i) Using the formula for $h$ above, give the class the $h$ would assign to $\mathbf{x} = (-3, 1)$. Show your work.




(ii) Letting $\mathbf{x} = (x_1, x_2)$, give the equation of the separating line in the form $x_2 = m\,x_1 + b$, where $m$ is the slope of the line and $b$ is the vertical-axis intercept. (Note, here, $b$ denotes the vertical-axis intercept, **not** the bias). Sketch the line described by this equation.

(7)

24)ii) Let $x = (x_1, x_2)$ $x_2 = mx_1 + b$

WKT $w = \sum \alpha_i x_i$

$= (-0.2)[(0,-1)] + 0.4[(1,1)]$

$= -0.2[1,1,0]$

$= [0 + 0.4 + 0.2], [0.2 + 0.4 + 0] = [0.6, 0.6]$

from the above points

$w_1 x_1 + b + w_2 x_2 = 0$

$0.6x + 0.6 x_2 - 0.4 = 0$

$x_1 + x_2 = \frac{0.4}{0.6} = \frac{2}{3}$

The graph for above equation ($x_1 = 0, x_2 = 0.66$)
($x_2 = 0, x_1 = 0.66$)