
# CS 725 Autumn 2024 | Endsem Exam

**Instructor: Preethi Jyothi Date/Time: Nov 23, 2024, 5.30 pm to 8.30 pm**

Total points: **50**

Name: _____

Roll number: _____

## Instructions

- This is an open notes exam which should be completed individually. No form of collaboration or discussion is allowed.
- Use of laptops or cell phones are not allowed during the exam.
- Write your name and roll number on the top of this page.
- This exam consists of 6 problems with sub-parts. The maximum possible score is 50.
- Work efficiently. The questions are not sorted in any order of difficulty.
- Write your answers legibly with a pen (not a pencil) in the space provided on the exam sheet. If necessary, use/ask for extra sheets to work out your solutions. These sheets will not be graded, so **please make sure your final answers appear on the exam sheet**.
- Good luck!

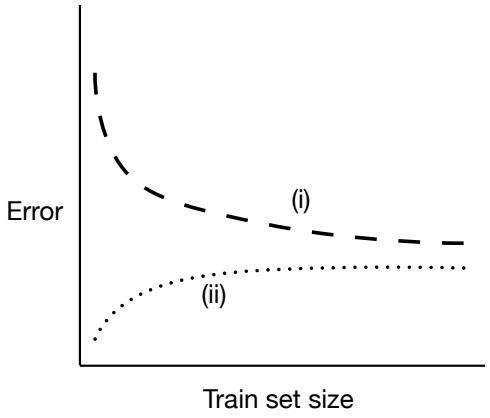
| Question | Score |
|---|---|
| Mixed Bag (I) | /8 |
| Mixed Bag (II) | /14 |
| Invariant or not? $k$-means Algorithm | /6 |
| Simple Backpropagation | /8 |
| SVMs | /7 |
| Kernels: Valid or not? | /7 |
| Total | /50 |

# Problem 1: Mixed Bag (I) (8 points)

(A) For a given threshold on the minimum number of instances at a leaf node, does increasing the maximum allowed depth of a decision tree make it less or more likely to overfit? Briefly justify your answer. **[1 pts]**

**Solution:** More likely to overfit. Fixed depth is a regularizer that prevents the tree from overfitting to specific properties of the training data.

(B) Consider a logistic regression classifier trained on a dataset with $n$ samples. Consider the following plot that shows how train and test error varies as a function of increasing $n$. Which of curve (i) or curve (ii) is more likely to represent training error? Briefly justify your answer. **[1 pts]**




**Solution:** Curve (ii) is the training error. Training error increases as the train set increases in size, and eventually plateaus once the model is able to generalize well enough.

(C) In a random forest classifier, would decreasing the number of randomly selected features at each node tend to increase or decrease the bias? Briefly justify your answer. **[1 pts]**

**Solution:** Decreasing the number of randomly selected features we consider for splitting at each tree node tends to increase the bias, since each tree would now tend to underfit.

(D) Briefly explain why we use weak learners for boosting. **[1 pts]**

**Solution:** To prevent overfitting; the boosted classifier grows in complexity with each training iteration.

(E) Consider a labeled training dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ for which we want to estimate a linear regression model parameterized by $\mathbf{w}$. We want to find a MAP estimate for $\mathbf{w}$. Let the prior over the weights $\mathbf{w} = [w_1, \ldots, w_d]$ be the following Laplace distribution for each dimension:

$$w_j \sim \text{Laplace}(0, \alpha) \qquad \text{where} \qquad \text{Laplace}(w; \mu, \alpha) = \frac{1}{2\alpha} \exp\left(-\frac{|w - \mu|}{\alpha}\right)$$

Let the datapoints be drawn from the following Gaussian distribution:

$$y_i \mid \mathbf{x}_i \sim \mathcal{N}(\mathbf{w}^T \mathbf{x}_i, \sigma^2)$$

Write down the cost function we should minimize in order to find the MAP estimate for $\mathbf{w}$. Just writing down the final expression will suffice. This expression should be expanded as far as possible. **[1 pts]**

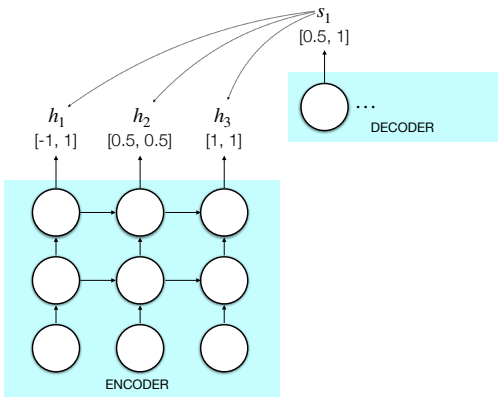
**Solution:**

$$L(\mathcal{D}; \mathbf{w}, \sigma, \alpha) = \frac{1}{2\sigma^2} \sum_i (y_i - \mathbf{w}^t \mathbf{x}_i)^2 + \frac{1}{\alpha} \sum_j |w_j|$$

(F) Say we train a perceptron classifier on a dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$. Let its initial weight vector be the zero vector. Then, the final weight vector learned by the perceptron can be written as a linear combination of $\mathbf{x}_i$'s. True or False? Justify your answer. **[1 pts]**

**Solution:** True. Because weight update rule is $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$, where $\mathbf{w}$ is initially the zero vector and $y$ is either 1 or -1.

(G) In the following encoder-decoder model, the 2-dimensional encoder and decoder states for a test example are shown.




Let the attention probabilities between $s_1$ and $h_1, h_2, h_3$ be $\alpha_1$, $\alpha_2$ and $\alpha_3$, respectively. Sort $\alpha_1$, $\alpha_2$ and $\alpha_3$ in descending order. Show your work. **[1 pts]**

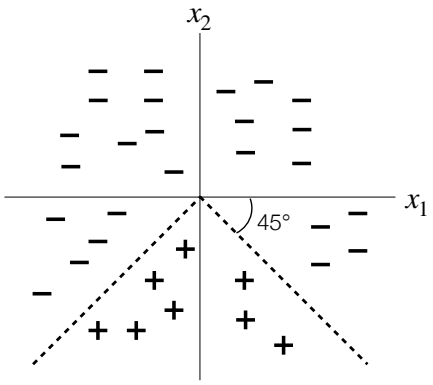
**Solution:** $\alpha_3 \propto 1.5 > \alpha_2 \propto 0.75 > \alpha_1 \propto 0.5$.

(H) Consider a feedforward neural network for binary classification (0 or 1) with a single output neuron. Let the output from this neuron be $o$. The predicted probability from this network should have been $\hat{y} = \sigma(o)$ where $\sigma$ denotes the sigmoid function. However, by mistake, the predicted probability is computed as $\hat{y} = \sigma(ReLU(o))$. Note that all instances with $\hat{y} \geq 0.5$ will be classified as 1. What problem would this mistake cause? **[1 pts]**

**Solution:** ReLU followed by sigmoid would result in all predictions being classified as 1.

# Problem 2: Mixed Bag (II) (14 points)

(A) Consider the 2D binary classification dataset in the figure below.




Pick as few features as possible from the options below for a logistic regression classifier such that all the datapoints in the figure are perfectly classified. Justify your answer. **[2 pts]**

1. 1
2. $x_1$
3. $x_2$
4. $|x_1|$
5. $|x_2|$

**Solution:** (i) $\phi(x) = [x_2, |x_1|]$. The v-shape curve corresponds to $x_2 + |x_1| = 0$. So, $x_2 + |x_1| \geq 0$ corresponds to the region enclosed within the v-shape.

Grading guide

The answer should contain equations of the lines. If the lines are incorrect with a bias term, give 0.5/2.

(B) Consider the following two convolution kernels:

$$K_1 = \begin{bmatrix} -0.5 & 0 & 0.5 \\ -1 & 0 & 1 \\ -0.5 & 0 & 0.5 \end{bmatrix} \quad K_2 = \begin{bmatrix} -0.5 & -0.5 & -0.5 \\ -1 & 6 & -1 \\ -0.5 & -0.5 & -0.5 \end{bmatrix}$$

1. What is the effect on convolving each filter above with an input black-and-white image? **[1 pts]**
2. Specify what effect each of these kernels will have on monochromatic (i.e., single-color) patches in the image. **[1 pts]**

**Solution:** Convolving with $K_1$ accentuates vertical edges. Convolving with $K_2$ sharpens the image. On monochromatic patches in the image, $K_1$ will blacken it out while $K_2$ will leave it as-is.

(C) Consider a multilabel multiclass classification task where each training example can have more than one correct class label, and the maximum number of correct labels is $C$. Let a training example $(\mathbf{x}, \mathbf{y})$ have $M$ correct classes. That is, the ground-truth vector $\mathbf{y} \in \{0,1\}^C$ is a binary vector of length $C$ with $M$ non-zero values. Prove the following bound for the cross-entropy loss $L_{\text{ce}}(\hat{\mathbf{y}}, \mathbf{y})$ on this example, where $\hat{\mathbf{y}}$ is a predicted probability distribution: **[2 pts]**

$$L_{\text{ce}}(\hat{\mathbf{y}}, \mathbf{y}) \geq M \log M$$

Note that you will need to use the following inequality: $\log\left(\frac{\sum_{i=1}^n x_i}{n}\right) \geq \frac{\sum_{i=1}^n \log(x_i)}{n}$.

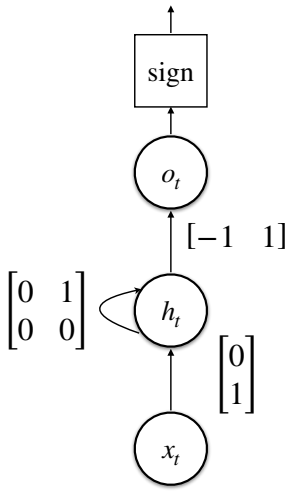
**Solution:** Let $S$ be the set of indices with non-zero values (i.e., 1) in $\mathbf{y}$.

$$\begin{aligned}
L_{\text{ce}}(\hat{\mathbf{y}}, \mathbf{y}) &= -\sum_{i \in S} \log \hat{y}_i \\
&= -M \cdot \frac{1}{M} \sum_{i \in S} \log \hat{y}_i \\
&\geq -M \log\left(\frac{1}{M} \sum_{i \in S} \hat{y}_i\right) \\
&\geq -M \log\left(\frac{1}{M}\right) \text{ since } \sum_{i \in S} \hat{y}_i \leq 1 \\
&= M \log M
\end{aligned}$$

Grading guide

Give 0.5/2 for writing down the correct expression for the loss.

(D) Consider the RNN below with its trained weight matrices marked on the respective edges. This RNN is a binary classifier consisting of a single output unit with a sign activation function that predicts 1 if $o_t \geq 0$ and -1 otherwise. Say the input is a sequence of integers. Describe the output of this RNN at the final time-step. (Ignore outputs from all other time-steps.) Show your work. Assume that all the biases are 0, and the initial hidden state at time-step 0 is the zero-vector. Assume a linear activation function for the recurrent unit. **[3 pts]**




**Solution:**

$$
\begin{aligned}
h_0 &= 0, 0 \\
h_1 &= 0, x_1 \\
h_2 &= x_1, x_2 \\
h_t &= x_{t-1}, x_t \\
o_t &= \text{sign}(x_t - x_{t-1})
\end{aligned}
$$

Output is checking whether the last two inputs depict an increasing or decreasing trend.

Grading guide

2 points for the correct expression for $h_t$, and one point for $o_t$. This assumes an identity mapping for the recurrent mapping. Assuming a linear mapping with a linear transformation and with the correct expressions for $h_t$ and $o_t$ will also get full 3 points. Assuming a non-linear mapping (tanh, ReLU) and deriving the correct expressions will also get full 3 points.

(E) Consider a linear regression model in one dimension with no bias terms, $y = wx$ where $y$ and $x$ are scalars. Weight $\mathbf{w}$ is initialized to 0. Consider a dataset with a single point $x = 1$, $y = 1$ and the loss function is mean squared error. Compute the value of the weight $w$ for one step of gradient descent with momentum. Recall the expressions for gradient descent with momentum:

$$\mathbf{v}_t \leftarrow \beta \mathbf{v}_{t-1} + \eta \mathbf{g}_t$$
$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \mathbf{v}_t$$

where $\mathbf{v}$, $\mathbf{g}$ and $\mathbf{w}$ are momentum, gradient and weight terms, respectively. Set $\beta = 0.99$, $\eta = 0.5$ in your computations. Assume $\mathbf{v}_0 = 0$. Show your work. **[2 pts]**

**Solution:**

1. $\frac{\partial (wx-y)^2}{\partial w} = 2(wx - y)x = -2$.
   $\mathbf{v}_0 = 0, \mathbf{v}_1 = 0.99 \times 0 + 0.5 \times -2 = -1$
   $\mathbf{w}_1 = 1$

Grading guide

1 point each for correct computations in the two steps outlined in the solution. Not assuming $\mathbf{v}_0 = 0$, but rest being correct, will also fetch you full points.

(F) Consider $\mathbf{X} \in \mathbb{R}^{d \times n}$ to be a design matrix consisting of $n$ mean-centered datapoints. Recall that the principal components in PCA can be computed using an eigendecomposition of the covariance matrix of the data, $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$. $\mathbf{S}$ can also be decomposed as:

$$\mathbf{S} = \mathbf{Q}\Lambda\mathbf{Q}^T$$

where $\mathbf{Q}$ is an orthogonal matrix and $\Lambda$ is a diagonal matrix. The projection matrix $\mathbf{U} \in \mathbb{R}^{d \times k}$ for PCA will then be the first $k$ columns of $\mathbf{Q}$. Then, the low-dimensional projection of the data would be $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$. Show that the coordinates of the projected datapoints are uncorrelated with each other; that is, the covariance matrix of $\mathbf{Z}$ is a diagonal matrix. **[3 pts]**

**Solution:** The covariance matrix of $\mathbf{Z}$: $\frac{1}{n}\mathbf{Z}\mathbf{Z}^T = \mathbf{U}^T\left(\frac{1}{n}\mathbf{X}\mathbf{X}^T\right)\mathbf{U} = \mathbf{U}^T\mathbf{S}\mathbf{U} = \mathbf{U}^T\mathbf{Q}\Lambda\mathbf{Q}^T\mathbf{U} == \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}\Lambda\begin{pmatrix}\mathbf{I} \\ \mathbf{0}\end{pmatrix}$.
The top left $k \times k$ block of this covariance matrix is diagonal, which means that the coordinates of the projected datapoints are uncorrelated with each other.

Grading guide

Correct expression for the covariance matrix of $\mathbf{Z}$ gets 1 point.

# Problem 3: Invariant or not? $k$-means clustering (6 points)

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a data matrix comprising $n$ training points. Given an affine transformation $(\mathbf{A}, \mathbf{b})$, $\mathbf{A} \in \mathbb{R}^{d \times d}, \mathbf{b} \in \mathbb{R}^d$, that acts on $\mathbf{X}$ to give $\mathbf{XA} + \mathbf{b}$, an algorithm is said to be $(\mathbf{A}, \mathbf{b})$**-invariant** if it gives exactly the same output for $\mathbf{XA} + \mathbf{b}$ as it gives for $\mathbf{X}$ when the initialization is appropriately modified.

(A) Show that the $k$-means algorithm is $(\mathbf{I}, \mathbf{v})$-invariant, where $\mathbf{I}$ is the $d \times d$ identity matrix and $\mathbf{v} \in \mathbb{R}^d$. Note that for $k$-means, the output is the partition of $n$ point indices into $k$ clusters. Show how the initialization should be modified to establish this invariance. That is, given an initialization of $k$ cluster centroids for $\mathbf{X}$, say $\boldsymbol{\mu}_1^0, \ldots, \boldsymbol{\mu}_k^0$, mention the initialization for $\mathbf{XI} + \mathbf{v}$ that results in the same output. **[3 pts]**

**Solution:** Yes. $\|\mathbf{x}_j - \boldsymbol{\mu}_k\|^2 = \|(\mathbf{x}_j + \mathbf{v}) - (\boldsymbol{\mu}_k + \mathbf{v})\|^2$. So, the distances from each point to the nearest cluster centroid will be unaltered. Initialization will become $\boldsymbol{\mu}_1^0 + \mathbf{v}, \ldots, \boldsymbol{\mu}_k^0 + \mathbf{v}$. (Cluster centroids with the translated points will be: $\frac{1}{|\mathbb{C}_k|} \sum_{j \in \mathbb{C}_k} (\mathbf{x}_j + \mathbf{v}) = \boldsymbol{\mu_k} + \mathbf{v}$.)

Grading guide

Showing how distances are unaltered gets 1 point. Correct reasoning about the cluster centroid gets 1 point. Correct initialization gets 1 point. Even if what happens to cluster centroids isn't mentioned, but the remaining two points are correct, full 3 points can be granted.

(B) Consider a matrix $\mathbf{U} \in \mathbb{R}^{d \times d}$ such that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is the $d \times d$ identity matrix. Is the $k$-means algorithm $(\mathbf{U}, \mathbf{b})$-invariant for all $\mathbf{b} \in \mathbb{R}^d$? If the answer is yes, justify your answer using mathematically precise statements and also specify the initialization for $\mathbf{XU} + \mathbf{b}$. If the answer is no, construct an example showing why it isn't. Note that for $k$-means, the output is the partition of $n$ point indices into $k$ clusters. **[3 pts]**

**Solution:** Yes.

$$\begin{aligned}
\|\mathbf{x}_j\mathbf{U} + \mathbf{b} - \boldsymbol{\mu}_k\mathbf{U} - \mathbf{b}\|^2 &= (\mathbf{x}_j\mathbf{U} - \boldsymbol{\mu}_k\mathbf{U})^T(\mathbf{x}_j\mathbf{U} - \boldsymbol{\mu}_k\mathbf{U}) \\
&= \mathbf{U}^T(\mathbf{x}_j - \boldsymbol{\mu}_k)^T(\mathbf{x}_j - \boldsymbol{\mu}_k)\mathbf{U} \\
&= \mathbf{U}^T\|(\mathbf{x}_j - \boldsymbol{\mu}_k)\|^2\mathbf{U} \\
&= \|(\mathbf{x}_j - \boldsymbol{\mu}_k)\|^2 \text{ since } \mathbf{U}^T\mathbf{U} = \mathbf{I}
\end{aligned}$$

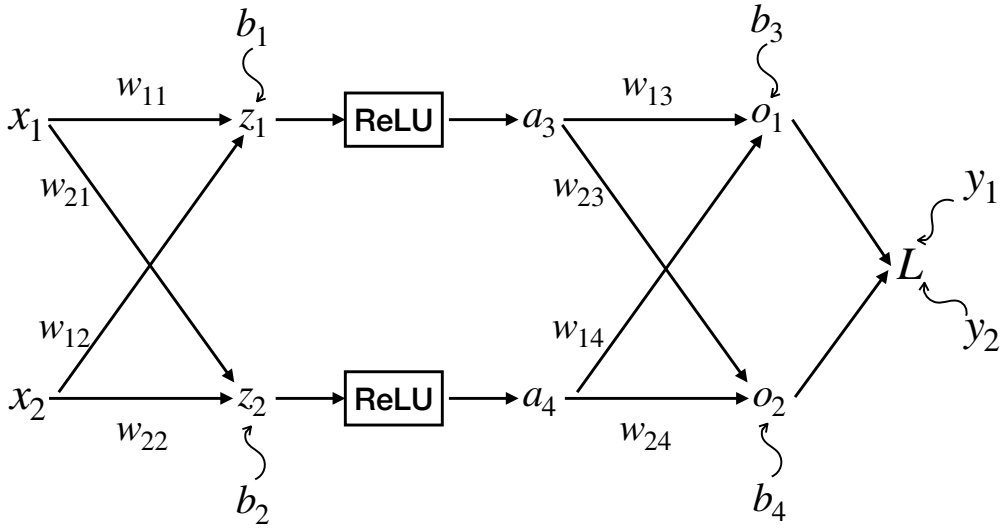
Distances remain unaltered. Initialization will become $\boldsymbol{\mu}_1^0\mathbf{U} + \mathbf{b}, \ldots, \boldsymbol{\mu}_k^0\mathbf{U} + \mathbf{b}$. (Cluster centroid with the rotated points will be: $\frac{1}{|\mathbb{C}_k|}\sum_{j \in \mathbb{C}_k} \mathbf{x}_j\mathbf{U} + \mathbf{b} = \boldsymbol{\mu_k}\mathbf{U} + \mathbf{b}$.)

Grading guide

Showing distances are unaltered gets 1.5 points. Correct initialization gets 1 point. Correct reasoning about the cluster centroid gets 0.5 point. Even if what happens to cluster centroids isn't mentioned, but the remaining two points are correct, full 3 points can be granted.

# Problem 4: Simple Backpropagation (8 points)

Consider the following neural network (NN) that has two input units $(x_1, x_2)$, one hidden layer with two units $(z_1, z_2)$ and ReLU activations $(a_3, a_4)$, and two output units $(o_1, o_2)$. The loss function $L = \frac{1}{2}\sum_{k=1}^{2}(o_k - y_k)^2$, where $(y_1, y_2)$ denote the ground-truth labels corresponding to $(x_1, x_2)$. The weights of the NN are all marked in the figure; the biases of different units, $b_1, b_2, b_3, b_4$ are marked using squiggly arrows next to the relevant units.




The NN is initialized with input layer weights: $\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ -0.5 & -1 \end{bmatrix}$ and hidden layer weights: $\begin{bmatrix} w_{13} & w_{14} \\ w_{23} & w_{24} \end{bmatrix} = \begin{bmatrix} 1 & -0.5 \\ 0.5 & -1 \end{bmatrix}$, and the biases $b_1, b_2, b_3, b_4$ are $-1, -1, 0.5, 1$, respectively.

Consider a training example $(x_1, x_2) = (2, 1)$ and its corresponding labels $(y_1, y_2) = (1.5, 1)$.

(A) What are the values for $o_1, o_2$ for the given example? Show your work. **[2 pts]**

**Solution:**

$$o_1 = w_{13}a_3 + w_{14}a_4 + b_3 = 1 \times 1.5 + -0.5 \times 0 + 0.5 = 2$$

$$o_2 = w_{23}a_3 + w_{24}a_4 + b_4 = 0.5 \times 1.5 + -1 \times 0 + 1 = 1.75$$

Grading guide

1 point each for $o_1$ and $o_2$.

Common Wrong Solutions

1. Does not consider the bias in the forward propagation equation.
2. Students who are computing the outputs using matrix multiplication, often mistakenly transpose the matrix where it is not required to be transposed.
3. Forgets to apply ReLU function on the intermediate activation.
4. Common calculation mistakes such as wrongly copying the values of weights and biases, incorrect handling of negative values, incorrect multiplication and/or addition etc.

(B) Use stochastic gradient descent with a learning rate of 0.1 to compute the updated weight for $w_{12}$. Show all your work, including the gradient computation of the loss $L$ with respect to $w_{12}$. **[4 pts]**

**Solution:**

$$\begin{aligned}\frac{\partial L}{\partial w_{12}} &= \left(\frac{\partial L}{\partial o_1}\cdot\frac{\partial o_1}{\partial a_3}\cdot\frac{\partial a_3}{\partial z_1}\cdot\frac{\partial z_1}{\partial w_{12}}\right)+\left(\frac{\partial L}{\partial o_2}\cdot\frac{\partial o_2}{\partial a_3}\cdot\frac{\partial a_3}{\partial z_1}\cdot\frac{\partial z_1}{\partial w_{12}}\right)\\ &= (o_1-y_1)\times w_{13}\times 1\times x_2+(o_2-y_2)\times w_{23}\times 1\times x_2\\ &= 0.875\end{aligned}$$

Thus, $w_{12}\leftarrow w_{12}-(0.1\times 0.875)=0.4125$

Grading guide

2 point for writing down the correct expression for $\frac{\partial L}{\partial w_{12}}$. Evaluate to correct value of 0.875 gets 1 point. Modify $w_{12}$ to get its updated value gets 1 point.

Common Wrong Solutions

1. Does not consider all the paths in the backward propagation equation. For example the gradient would flow from two paths: $L \longrightarrow o_1 \longrightarrow a_3 \longrightarrow ReLU(.) \longrightarrow z_1 \longrightarrow x_2$ and $L \longrightarrow o_2 \longrightarrow a_3 \longrightarrow ReLU(.) \longrightarrow z_1 \longrightarrow x_2$. Many students have made this mistake of not considering both the paths for gradient flow in backward propagation equation.
2. Students who are computing the outputs using matrix multiplication, often wrote the incorrect formula for backward propagation where the matrices are not correctly handled.
3. Forgets to update the weight after computing its gradient.
4. Uses cross entropy loss instead of the MSE loss given in the question.
5. If the forward computation is wrong then the backward computation would definitely be wrong and for many students, this is the case.

(C) What is the updated weight for $w_{12}$ if the loss $L$ is L2-regularized and defined as $L = \frac{1}{2}\sum_{k=1}^{2}(o_k - y_k)^2 + \|\mathbf{w}\|^2$ where $\mathbf{w} = [w_{11}, w_{12}, w_{21}, w_{22}, b_1, b_2, w_{13}, w_{23}, w_{14}, w_{24}, b_3, b_4]$. You can reuse calculations from part (B). **[2 pts]**

**Solution:**

$$\begin{aligned}\frac{\partial L}{\partial w_{12}} &= \left(\frac{\partial L}{\partial o_1}\cdot\frac{\partial o_1}{\partial a_3}\cdot\frac{\partial a_3}{\partial z_1}\cdot\frac{\partial z_1}{\partial w_{12}}\right) + \left(\frac{\partial L}{\partial o_2}\cdot\frac{\partial o_2}{\partial a_3}\cdot\frac{\partial a_3}{\partial z_1}\cdot\frac{\partial z_1}{\partial w_{12}}\right) + 2w_{12} \\ &= (o_1 - y_1)\times w_{13}\times 1\times x_2 + (o_2 - y_2)\times w_{23}\times 1\times x_2 + 1 \\ &= 1.875\end{aligned}$$

Thus, $w_{12} \leftarrow w_{12} - (0.1 \times 1.875) = 0.3125$.

Common Wrong Solutions

1. Does not understand the fact that $\frac{\partial\|\mathbf{w}\|^2}{\partial w_{12}} = 2 \times w_{12}$. Many students have missed the scalar "2" before the term $w_{12}$.
2. Many students simply wrote that the weight won't change after adding a regularization term.
3. Forgets to update the weight after computing its gradient.
4. Does not reuse the computation from part B and this led to recomputation of the entire gradient where calculation mistake happened.
5. If the gradient computation in part B is wrong then the answer of this part would definitely be wrong and for many students, this is the case.

# Problem 5: SVMs (7 points)

(A) The soft-margin SVM can also be written as minimizing an L2-regularized hinge loss. That is,

$$\mathcal{L}(\mathbf{w}, b) = \frac{C}{N}\sum_{i=1}^{N}\mathcal{L}_h(\hat{y}_i, y_i) + \frac{\|\mathbf{w}\|^2}{2}$$
$$\text{where } \mathcal{L}_h(\hat{y}, y) = \max(0, 1 - y\hat{y}) \text{ and } \hat{y} = \mathbf{w}^T\mathbf{x} + b$$

where $y \in \{-1, +1\}$.

1. Specify the condition on $\mathbf{w}$ such that $\nabla_{\mathbf{w}}\mathcal{L}$ is well-defined. **[1 pts]**
2. Give an expression for $\nabla_{\mathbf{w}}\mathcal{L}$ when it is well-defined. Your final expression should not contain any unexpanded (partial) derivatives. Show your work. **[2 pts]**

**Solution:** $\nabla_{\mathbf{w}}\mathcal{L}$ is not well-defined if for any training example $y_i\hat{y}_i = 1$, thus yielding the point at which max of hinge loss is not differentiable.

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} + \frac{C}{N}\sum_{i=1}^{N}\frac{\partial\mathcal{L}_h^i}{\partial\mathbf{w}}$$
$$\frac{\partial\mathcal{L}_h^i}{\partial\mathbf{w}} = \frac{\partial\mathcal{L}_h^i}{\partial\hat{y}}\mathbf{x}$$
$$\frac{\partial\mathcal{L}_h^i}{\partial\hat{y}} = \begin{cases} -y_i & \text{if } 1 - y_i\hat{y}_i > 0 \\ 0 & \text{if } 1 - y_i\hat{y}_i < 0 \end{cases}$$

**Grading guide**

1 point for the correct expression of $\frac{\partial\mathcal{L}}{\partial\mathbf{w}}$ using $\frac{\partial\mathcal{L}_h^i}{\partial\mathbf{w}}$, and 1 point for the correct expansion of $\frac{\partial\mathcal{L}_h^i}{\partial\mathbf{w}}$.

(B) Recall the soft-margin SVM objective function. Which value of $C$ is most likely to overfit the training data? Pick one answer and briefly justify. **[1 pts]**

1. C = 0.001
2. C = 0.1
3. C = 10
4. C = 1000

**Solution:** C = 1000. Makes the $\xi_i$ values very small, thus leaving very little slack for each training instance and hence is more likely to overfit the training data.

(C) What can you say about the soft-margin SVM solution when $C \to \infty$? Justify your answer. **[1 pts]**

**Solution:** When $C \to \infty$, all the slack variables $\xi_i \to 0$. Thus, as $C \to \infty$ the solution of the soft-margin SVM approaches the solution of the hard-margin SVM.

Mention of hard-margin SVM gets full 1 point, and nothing else.

(D) Describe a 2D dataset that will be perfectly separated by a simple SVM with a linear kernel but will perform poorly (in terms of training error) using a decision tree with bounded depth. Assume the decision tree learns axis-parallel boundaries. (Also, recall that with a continuous attribute, say $x_i$, decision trees pick a threshold $\tau$ to create a binary question by splitting on $x_i > \tau$ and $x_i \leq \tau$.) You can draw a sketch of the dataset if that helps further clarify your description. **[2 pts]**

**Solution:** Dataset that is concentrated along the diagonal line. Pluses on one side of the line, and minuses on the other side; both close to the separating plane.

There could be multiple correct solutions.

# Problem 6: Kernels: Valid or not? (7 points)

(A) Consider $K(x, x') = \log \gcd(x, x')$ where $\gcd(x, x')$ returns the greatest common divisor of the two numbers, and $x, x'$ are integers in the range $[1, 100]$. Is $K$ a valid kernel? If yes, provide the corresponding feature mapping; otherwise, provide a short proof or a counter-example. **[4 pts]**

**Solution:** True.

$\phi(x)$ is a vector of length 25, with headers $2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 5, 5, 7, 7, 11, 13, \ldots, 47$ for each of the 25 dimensions. That is, six dimensions for 2, four for 3, two for 5, two for 7 and 1 dimension for each of the primes from 11 to 47. $\phi(1)$ is the zero-vector of length 25. For a given $x \neq 1$, find its prime factorization and for every prime factor $k$, add a corresponding value $v = \sqrt{\log k}$ to $\phi(x)$. For example, if $x = 45 = 3 \times 3 \times 5$, then $\phi(x) = [0, 0, 0, 0, 0, 0, \sqrt{\log 3}, \sqrt{\log 3}, 0, 0, \sqrt{\log 5}, 0, 0, 0, \ldots]$. If $x' = 25 = 5 \times 5$, then $\phi(x') = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \sqrt{\log 5}, \sqrt{\log 5}, 0, 0, \ldots]$. Then, $\phi(x)^T \phi(x') = \log 5$.

1/4 points for answering True. Remaining three points for correct construction of $\phi(x)$.

Common Wrong Solutions

1. **Trying to use Mercer's theorem** Many students have tried to use Mercer's theorem to prove that the kernel is valid. The second condition of Mercer's theorem states that $k(\cdot, \cdot)$ is a valid kernel iff the kernel matrix $K \in \mathbb{R}^{n \times n}$ where $K_{ij} = k(x_i, x_j)$ is PSD for **all possible subsets** $\{x_1, x_2, \ldots x_n\}$ of elements in $\mathbb{R}^d$.

   **If you have proven positive semi-definiteness for some particular values of the kernel matrix $K$, you will get no marks. Please do not crib for the same.**

   Another mistake in the use of Mercer's theorem is simply not understanding what PSD means. **Just because all the elements in the matrix are positive, doesn't mean the matrix becomes PSD.** A matrix $K \in \mathbb{R}^{n \times n}$ is PSD if

   $$\boldsymbol{v}^T K \boldsymbol{v} \geq 0 \quad \forall \boldsymbol{v} \in \mathbb{R}^n$$

   The vectors $\boldsymbol{v}$ here, are not limited to the original domain of scalars in $[1, 100]$.

2. **Have vaguely mentioned "primes" and "factorisation" in the solution** You will receive no marks for vague mentions of these terms, which are in no way related to the solution. Appropriate marks (up to 1) have been awarded for designing feature maps $\phi(x)$ that are somewhat in the direction of the solution.

   **If you have simply given the formula of the gcd in terms of prime factorisations without making meaningful progress on feature maps, no marks will be awarded.**

3. **Assumed feature maps to be scalars**. Feature maps are vectors, and conclusions drawn from treating **dot product of two vectors** as multiplication of two scalar quantities is obviously wrong, and will fetch no marks.

(B) Let $K(a,b) = a + b$ where $a, b$ are integers in the range $[0, n]$. Is $K$ a valid kernel? If yes, provide the corresponding feature mapping; otherwise, provide a short proof or a counter-example. **[3 pts]**

**Solution:** False. Here is a proof by contradiction.

Let $K(a,b) = \phi(a)^T\phi(b)$. $K(a,a) = \phi(a)^T\phi(a) = 2a$. Thus, $\|\phi(a)\| = \sqrt{2a}$. $K(0,0) = 0 = \phi(0)^T\phi(0) \Rightarrow \|\phi(0)\| = 0$, $\phi(0)$ is the zero vector. $K(a,0) = \phi(a)^T\phi(0) = a$ by the definition of the kernel, except $K(a,0) = 0$ by the definition of $\phi(0)$. This is a contradiction.

1/3 points for answering False. Remaining 2 points for the proof.

Common Wrong Solutions

1. **Not deriving why** $\phi(0) = 0$. If you have simply written $\phi(0) = 0$ without the argument below, **1 mark will be deducted** if the rest of the solution follows the proof by contradiction.

$$K(0,0) = 0 + 0 = \phi(0)^T\phi(0) = ||\phi(0)||^2 \Rightarrow \phi(0) = 0$$

   If your argument is not exactly the one written above, **please do not present any cribs.**

2. **Counter-example of a non-invertible matrix** If you have chosen a couterexample of a matrix, whose determinant is **negative**, well and good. It is a valid counter example, since Mercer's theorem says that **all** kernel matrices $K$ should be positive semi definite (eigenvalues $\geq 0$, and if a particular matrix has $\det(K) < 0 \Rightarrow$ some eigenvalue $\lambda_i < 0$.

   **However, examples showing a matrix is not invertible because it is not full rank or** $\det(K) = 0$ **are not valid counter-examples, since it is possible for a matrix to have some zero eigenvalues, and be positive semi-definite.** The most trivial example is of the $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ matrix, which is indeed PSD.

3. **Any mention of** $\phi(a) = \sqrt{a+b}$ **or** $\phi(a) = \sqrt{a}$. Any mention of obviously wrong feature maps in your solution, even when used to arrive at correct conclusions, **will fetch you no marks. Please do not crib for the same.**