CS229 Final Exam - Summer 2022

Total time to take the exam is 4 hours. The 4-hour window can be started at the earliest 12:01am on Friday, August 12, and end before 11:59pm on Saturday, August 13

[0 points] The Stanford University Honor Code

At the top of your solution file, please write/type the following Honor Code statement and attest it (i.e. sign or print your name below it), thereby implying your consent to follow the code:

"*I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.*"

Submissions without the attested Honor Code statement will NOT be graded.

1 [10 points] True or False

Each question is for 1 point. To earn 1 point per question, provide the correct True or False answer along with the correct justification. Incorrect answers or incorrect justifications will earn 0 points. Justifications must be short (a couple of sentences). There will be no negative points.

1. **True or False**. Consider a training dataset $\{x^{(1)}, .., , x^{(n)}\}$, where we fit a latent variable model $p(x, z; \theta)$ to the data using the EM algorithm. This algorithm can't overfit to the data.

   Answer:

2. **True or False**. Consider two models, model A trained with the loss function $L(\theta)$ and model B trained with $L(\theta) + \lambda||\theta||_2^2$, where $\lambda > 0$. It is possible that the generalization error for model A is smaller then the generalization error of model B.

   Answer:

3. **True or False**. For a given training dataset, there exist a kernel such that a Support Vector Machine trained on the dataset with that kernel will output class probabilities.

   Answer:

4. **True or False**. All generative models learn the joint probability distribution of the data.

   Answer:

5. **True or False**. For the k-means clustering algorithm, with fixed k, and number of data points evenly divisible by k, the number of data points in each cluster for the final cluster assignments is deterministic for a given dataset and does not depend on the initial cluster centroids.

   Answer:

6. **True or False.** Suppose we use two approaches to optimize the same problem: Newton's method and stochastic gradient descent. Assume both algorithms eventually converge to the global minimizer. Suppose we consider the total run time for the two algorithms (the number of iterations multiplied by

the time complexity per iteration), then stochastic gradient descent may converge more quickly than Newton's method.

Answer:

7. **True or False.** All Markov decision process $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \gamma, R)$, where $\mathcal{S}$ is the states, γ is the discount factor, R is the reward function, $\mathcal{A}$ is the actions, and $\{P_{sa}\}$ is the transition probabilities, have an unique optimal policy.

   Answer:

8. **True or False.** Let π_1 be a greedy policy with respect to V^{π_0} for some policy π_0, i.e., $\pi_1(s) = \text{argmax}_{a \in A} \sum_{s \in S} P_{s\pi_0(s)} V^{\pi_0}(s)$. Then $V^{\pi_1}(s) \geq V^{\pi_0}(s)$ for all $s \in S$.

   Answer:

9. **True or False.** Consider the linear regression problem without an intercept term, with $x^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \mathbb{R}$, for all $i \in \{1, ..., n\}$. Let $\dim(\text{span}\{x^{(1)}, x^{(2)}, ..., x^{(n)}\}) = k$, then the optimization problem $\text{argmin}_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2$ has a unique solution if and only if $k = d$.

   Answer:

10. **True or False**. Consider a dataset $\{x^{(i)}\}_{i=1}^n$ where $x^{(i)} = [x_1^{(i)}, x_2^{(i)}]^T \in \mathbb{R}^2$, with $x_2^{(i)} = x_1^{(i)} \times x_1^{(i)}$. PCA with the top eigenvector will capture all or almost all variance in this dataset.

    Answer:

2 [10 points] Short Answers

Each sub-question is worth 2 points. Provide a short justification for each answer.

1. **Short Answer**. For which values of $a \in \mathbb{R}$ is the matrix $A = \begin{bmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}$ a valid covariance matrix?

   Answer:

2. **Short Answer**. Consider some data $s \in \mathbb{R}^2$, generated via 2 independent, non gaussian, sources. This generates a datapoint x via $x = As$, for some matrix A. Repeated three times this gives gives us a dataset $\{x^1, x^2, x^3\}$, where $x_1 = [1, 2]^T$, $x_2 = [2, 3]^T$, $x_3 = [0, 4]^T$.

   Construct two matrices A_1, A_2, and sources $s_1^1, s_1^2, s_1^3, s_2^1, s_2^2, s_2^3 \in \mathbb{R}^2$, such that $x^j = A_i s_i^j$ for $i \in \{1, 2\}$, $j \in \{1, 2, 3\}$, and such that there does not exist $t \in \mathbb{R}$ such that $A_1[k, :] = tA_2[k, :]$ for any $k \in \{1, 2\}$, where $A_i[k, :]$ denotes row k of A_i.

   Answer:

3. **Short Answer**. Consider two models: one trained with Gaussian Process Regression, and the other with Bayesian Linear Regression. Assume that the number of training samples, n, is very large. What is one possible drawback of the Gaussian Process Regression model, compared to the Bayesian linear regression model, when making predictions on new data?

   Answer:

4. **Short Answer**. Consider an input attribute $x \in \mathbb{R}^d$, and consider the feature mapping $\phi(x)$ consisting of all monomials of degree 1 and degree 2 of x. What is the dimension of $\phi(x)$?

   Answer:

5. **Short Answer**. Consider a neural network trained for a multi-class classification task:

$$\begin{aligned} z^{[1]} &= W^{[1]}x + b^{[1]} \\ a^{[1]} &= \sigma(z^{[1]}) \\ z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\ \hat{y} &= \text{softmax}(z^{[2]}) \end{aligned}$$

   where $W^{[1]} \in \mathbb{R}^{p\times n}$, $b^{[1]} \in \mathbb{R}^p$, $W^{[2]} \in \mathbb{R}^{K\times p}$, $b^{[2]} \in \mathbb{R}^K$, and for any vector z, $\text{softmax}(z) = \exp(z)/\sum_k \exp(z_k)$ (where the division is performed elementwise). Describe how the neural network can be viewed as learning a feature mapping for softmax regression.

   Answer:

3 [10 points] Representer Theorem

In the supervised learning setting, we have input data $x \in \mathbb{R}^n$ and label $y \in \mathcal{Y}$.

1. Specify the set $\mathcal{Y}$ in the case of:

   (a) **[1 point]** linear regression
   Answer:

   (b) **[1 point]** binary classification
   Answer:

   (c) **[1 point]** multi-class (k-class) classification
   Answer:

2. *(Representer theorem)*

   For each of the above problems, we constructed a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ with $\ell(\theta^T x, y)$ measuring the loss we suffer for predicting $\theta^T x$. For example, linear regression uses the squared residual for the loss: $\ell(\theta^\top x, y) = \frac{1}{2}(\theta^\top x - y)^2$. Let $\Omega : \mathbb{R}_+ \to \mathbb{R}$ be a non-decreasing function. Consider the set of training examples $\{(x^{(i)}, y^{(i)}) \mid i \in \{1, 2, ..., n\}$. Let $\tilde{J} : \mathbb{R}^d \to \mathbb{R}$ be the regularized objective across n examples:

   $$\tilde{J} : \theta \mapsto \sum_{i=1}^{n} \ell(\theta^\top x^{(i)}, y^{(i)}) + \Omega\left(\|\theta\|_2\right)$$

   Let $\theta \in \mathbb{R}^d$. In this question, we would like to show that there exists $\alpha \in \mathbb{R}^n$ such that:

   $$\tilde{J}(\hat{\theta}) \leq \tilde{J}(\theta) \tag{1}$$

   where

   $$\hat{\theta} := \sum_{i=1}^{n} \alpha_i x^{(i)}$$

   (a) **[2 points]** Recall that for any subspace $\mathcal{S}$ of $\mathbb{R}^d$ (i.e. $\mathcal{S} \subset \mathbb{R}^d$) we can define its orthogonal complement $\mathcal{S}^\perp = \left\{u \in \mathbb{R}^d \mid \forall v \in \mathcal{S}, u^T v = 0\right\}$. Also recall that the orthogonal decomposition theorem states that any vector $x \in \mathbb{R}^d$ can be uniquely written as:

   $$x = x_{\mathcal{S}} + x_{\mathcal{S}^\perp}$$

   where $x_{\mathcal{S}}$ and $x_{\mathcal{S}^\perp}$ are the projections of x onto $\mathcal{S}$ and $\mathcal{S}^\perp$, respectively.

   Let $\theta \in \mathbb{R}^d$ and $\mathcal{S}_x := \text{span}\left\{x^{(i)} \mid i \in \{1, 2, ..., n\}\right\}$. Denote $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{S}^\perp}$ the projections of θ onto $\mathcal{S}_x$ and $\mathcal{S}_x^\perp$, respectively. Write the orthogonal decomposition of θ and express $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{S}^\perp}$ in bases of $\mathcal{S}_x$ and $\mathcal{S}_x^\perp$, respectively. The values of the coefficients don't need to be explicitly written, but please do introduce relevant notation and specify the dimensions of each basis in terms of any of the following values: $\dim S_x$, n and d.
   Answer:

   (b) **[3 points]** Show that $\|\theta\|_2 \geq \|\theta_{\mathcal{S}}\|_2$.
   Answer:

(c) **[2 points]** Now prove the theorem. That is, show that there exists $\alpha \in \mathbb{R}^n$ such that:

$$\tilde{J}(\hat{\theta}) \leq \tilde{J}(\theta) \tag{2}$$

where

$$\hat{\theta} := \sum_{i=1}^{n} \alpha_i x^{(i)}$$

.

Answer:

The implications of this theorem are far-reaching. In particular, if $\tilde{J}$ can be minimized, then by virtue of the representer theorem, its minimizer admits the following representation:

$$\theta^* = \sum_{i=1}^{n} \alpha_i x^{(i)}$$

and we can replace any occurrence of $\theta^\top x$ with $\theta^\top x = \sum_{i=1}^{n} \alpha_i x^\top x^{(i)}$ which is handy because the kernel trick $\phi(x)^\top \phi(x^{(i)}) := K(x, x^{(i)})$ can then be applied for some high-dimensional mapping ϕ, and we can directly solve for α instead.

4 [10 points] Neural networks

In this problem, we'll perform classification using a modified two-layer neural network. For any input vector $x \in \mathbb{R}^n$, our neural network outputs a probability distribution over K classes following the forward propagation rules:

$$\begin{aligned} z^{[1]} &= W^{[1]}x + b^{[1]} \\ a^{[1]} &= \sigma(z^{[1]}) \\ z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\ \hat{y} &= \text{softmax}(z^{[2]}) \end{aligned}$$

where $W^{[1]} \in \mathbb{R}^{p \times n}$, $b^{[1]} \in \mathbb{R}^p$, $W^{[2]} \in \mathbb{R}^{K \times p}$, $b^{[2]} \in \mathbb{R}^K$, and for any vector z, $\text{softmax}(z) = \exp(z)/\sum_k \exp(z_k)$ (where the division is performed elementwise).

We evaluate our model using the cross entropy loss (CE). For a single example (x, y), the cross entropy loss is:

$$\text{CE}(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k,$$

where $\hat{y} \in \mathbb{R}^K$ is the vector of softmax outputs for a single training example x, and $y \in \mathbb{R}^K$ is the ground-truth vector for training example x such that $y = [0, ..., 0, 1, 0, ..., 0]^\top$ contains a single 1 at the position of the correct class. For m training examples, we average the cross entropy loss over the m examples:

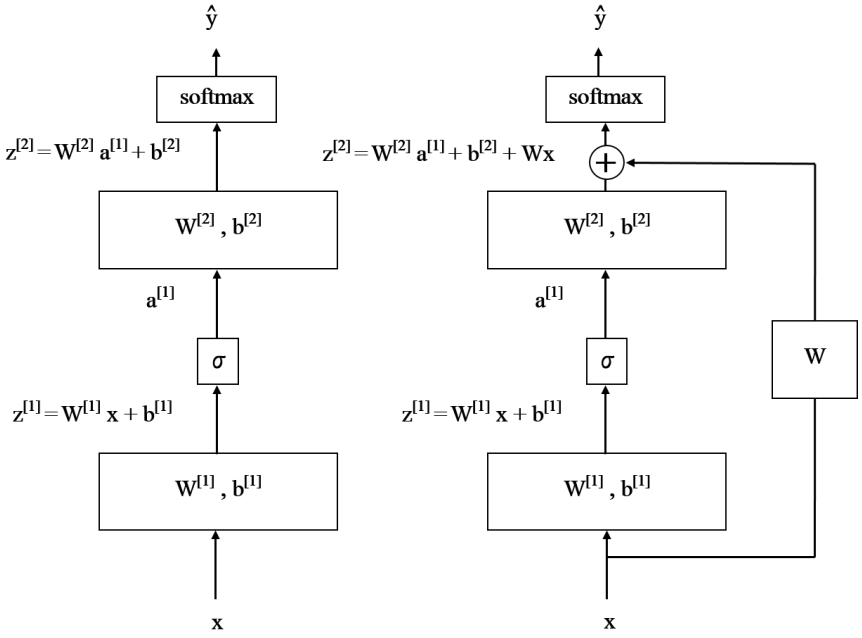
$$J(W^{[1]}, W^{[2]}, b^{[1]}, b^{[2]}) = \frac{1}{m}\sum_{i=1}^{m} \text{CE}(y, \hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k \log \hat{y}_k.$$

We modify the described network by adding a "shortcut" connection between the input x and the second layer. The forward propagation equations then become:

$$\begin{aligned} z^{[1]} &= W^{[1]}x + b^{[1]} \\ a^{[1]} &= \sigma(z^{[1]}) \\ z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} + Wx \\ \hat{y} &= \text{softmax}(z^{[2]}) \end{aligned}$$

where $W \in \mathbb{R}^{K \times n}$, and $J(W^{[1]}, W^{[2]}, b^{[1]}, b^{[2]}, W)$ is defined as before.

Figure 4 (on the next page) shows the two-layer neural network, before and after adding the shortcut connection. In practice, it is often observed that shortcut connections improve the learning of neural networks.



CE *ntering*

Figure 1: On the left, a two-layer neural network without shortcut connection. On the right, the same two-layer neural network with a shortcut connection.

1. **[2 point]** How many parameters does the model including the shortcut connection have? Your answer should be expressed in terms of n, p, and K.

   Answer:

2. **[4 points]** In this part of the question, we'll consider a single input vector $x \in \mathbb{R}^n$ with true label vector y. Show that: $\nabla_{z^{[2]}} \text{CE}(y, \hat{y}) = \nabla_{z^{[2]}} \text{CE}(y, \text{softmax}(z^{[2]})) = \hat{y} - y$. *Hint*: To simplify your answer, it might be convenient to denote the true label of x as $l \in \{1, \ldots, K\}$. Hence l is the index such that that $y = [0, ..., 0, 1, 0, ..., 0]^\top$ contains a single 1 at the l-th position. You may also wish to compute $\partial \text{CE}(y, \hat{y}) / \partial z_j^{[2]}$ for $j \neq l$ and $j = l$ separately.

   If you get stuck, the next part of this question can be done independently of this part.

   Answer:

3. **[4 points]** Find the expressions for $\nabla_{W^{[2]}} J$, $\nabla_{W^{[1]}} J$, $\nabla_{b^{[1]}} J$, and $\nabla_x J$ for a single training example (x, y). We've already provided $\nabla_{b^{[2]}} J$ for you. You may assume the result from Part (b) is true for this part of the question.

   Using the result from Part (b), we have: $\delta_{z^{[2]}} = \nabla_{z^{[2]}} \text{CE}(y, \hat{y}) = \hat{y} - y$.

   $$\begin{aligned} \nabla_{b^{[2]}} \text{CE}(y, \hat{y}) &= \delta_{z^{[2]}} \circ \nabla_{b^{[2]}} z^{[2]} \\ &= (\hat{y} - y) \circ \nabla_{b^{[2]}} (W^{[2]} a^{[1]} + b^{[2]} + Wx) \\ &= \hat{y} - y \end{aligned}$$

   Answer: