**10-315 Intro to ML**
**Spring 2025**
**Practice Midterm Exam 2**

**Name:** ____________________

**Andrew ID:** ____________________

**Instructions:**

- Please fill in your name and Andrew ID above.
- Be sure to write neatly and dark enough or you may not receive credit for your exam.
- Repeated from Ed post: You may use TWO two-sided, hand-written sheets of paper containing your notes to use during the exam. Note: this may NOT be created digitally and printed out. No calculators will be necessary or allowed.
- Note: All vectors on this exam are column vectors unless we state otherwise.
- Note: All logs are natural logs unless we state otherwise.

Below is a space for you to write any assumptions you make as you go through the exam that you would like us to consider. If you have nothing to state, you can leave this box blank.

Note: If at all possible, you should raise your hand to ask a clarifying question during the exam rather than relying on any assumptions.

# 1 Short Answer / Multiple Choice

1. (8 points) Which of the following machine learning algorithms are probabilistic generative models?

   Select all that apply.

   □ Logistic regression

   □ Logistic regression with $\ell_1$ regularization

   □ Linear regression with a Gaussian prior on the parameters

   □ Convolutional neural network

   □ Naive Bayes with Categorical class distributions and Gaussian class-conditional distributions

   □ Naive Bayes with Categorical class distributions and Bernoulli class-conditional distributions with add-one smoothing

   □ Linear Discriminant Analysis

   □ Quadratic Discriminant Analysis

   □ None of the above

2. (2 points) True or False: Generative models allow us to formulate the joint distribution of the input features and the output, e.g. $p(y, x_1, x_2, x_3)$.

   ○ True ○ False

3. (10 points) Probabilistic Models

   For each machine learning model, select the probabilistic model that it attempts to fit. Assume that $\mathbf{x}$ is the input, $y$ is the output, and $\boldsymbol{\theta}$ is the vector of all parameters.

   (The set of options is the same for all of the following questions.)

   (a) Linear regression (without regularization)

   ○ $p(y \mid \boldsymbol{\theta})$ ○ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

   ○ $p(\mathbf{x} \mid \boldsymbol{\theta})$ ○ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

   ○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ ○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

   ○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$ ○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

   ○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$ ○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$

(b) Linear regression with L2 regularization

○ $p(y \mid \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$

○ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$

(c) Neural networks for classification (no regularization)

○ $p(y \mid \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$

○ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$

(d) Classification using categorical class distributions and Gaussian class-conditional distributions. No additional assumptions including no naive Bayes assumption.

○ $p(y \mid \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$

○ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$

(e) Classification using categorical class distributions and Gaussian class-conditional distributions with a naive Bayes assumption but no other assumptions.

○ $p(y \mid \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$

○ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$

○ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$

4. (10 points) Consider a single-headed causal attention block in a transformer language model that has already been trained and is being used at inference (test) time, i.e., no parameters are being updated:

$$V = XW_V \qquad K = XW_K \qquad Q = XW_Q \qquad S = QK^\top/\sqrt{d_k}$$

$$A = g_{softmax}(S) \qquad Z = AV$$

where $d_k$ is the length of the value, key, and query vectors, and $d_k$ is also the dimension of the input vectors in the rows of $X$.

Which of the following matrix dimensions change as the context size $T$ grows? Select all that apply

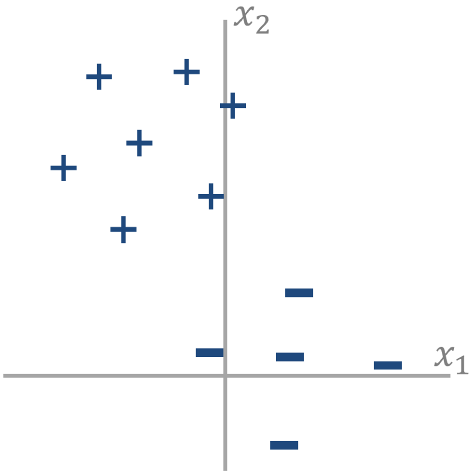
| | | | |
|---|---|---|---|
| $X$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $W_V$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $W_K$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $W_Q$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $V$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $K$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $Q$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $S$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $A$: | □ Number of rows | □ Number of columns | □ Size doesn't change |
| $Z$: | □ Number of rows | □ Number of columns | □ Size doesn't change |

# 2 Regularization

1. We attempt to solve the binary classification task depicted in the following figure with the simple logistic regression model.

$$P(Y = 1 \mid \boldsymbol{x}, \boldsymbol{w}) = g(w_0 + w_1x_1 + w_2x_2) = \frac{1}{1 + \exp\left(-w_0 - w_1x_1 - w_2x_2\right)}$$

In the figure below, the + values represent class $Y = 1$, and the $-$ values represent class $Y = 0$. Notice that the training data can be separated with zero training error with a linear separator.



(a) (6 points) Consider training a version of regularized logistic regression models where we try to maximize:

$$\sum_{i=1}^{N} \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, w_0, w_1, w_2) - C(w_j)^2$$

where hyperparameter $C >> 0$, and the penalty is on only one fixed $j$ from $\{0, 1, 2\}$. In other words, only one of the parameters is penalized.

For the regularization of each $w_j$ and $C >> 0$, state whether the training error on the above dataset increases, decreases, or stays the same when compared to the unpenalized logistic regression model. Provide a brief justification for each of your answers.

By penalizing $w_0$, the training error:

○ Increases ○ Remains the Same ○ Decreases

Justification

By regularizing $w_1$, the training error:

○ Increases ○ Remains the Same ○ Decreases

Justification

By regularizing $w_2$, the training error:

○ Increases ○ Remains the Same ○ Decreases

Justification

(b) (3 points) If we change the form of regularization to L1-norm (absolute value) and regularize $w_1$ and $w_2$ only (but not $w_0$), we get the following penalized log-likelihood:

$\sum_{i=2}^{N} \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, w_0, w_1, w_2) - C(|w_1| + |w_2|)$

Consider again the problem in Figure 1 and the same logistic regression model $P(y = 1 \mid \boldsymbol{x}, \boldsymbol{w}) = g(w_0 + w_1 x_1 + w_2 x_2)$

As we increase the regularization parameter C, which of the following scenarios do you expect to observe? (Choose only one) Briefly explain your choice:

○ First $w_1$ will become 0, then $w_2$

○ First $w_2$ will become zero then $w_1$

○ $w_1$ and $w_2$ will become zero simultaneously

○ None of the weights will become exactly zero, only smaller as $C$ increases.

Justification

# 3 MLE and MAP

1. (6 points) Assume we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ and assume our $x^{(i)}$ are i.i.d from a distribution with the following density function: $f(x; \lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$

   Write the expression for the likelihood of the data $\mathcal{D}$ in terms of $x^{(i)}$, $\lambda$ and $N$?

   Write the expression for the log-likelihood of the data $\mathcal{D}$ in terms of $x^{(i)}$, $\lambda$ and $N$?

   Derive the equation for the MLE of $\lambda$ in terms of $x^{(i)}$ and $N$. You must show your work.

2. Let binary $Y$ be a random variable representing a coin flip. The probability of outcome heads, $Y = 1$ is modeled by a Bernoulli distribution with parameter $\phi \in [0, 1]$, i.e. $p(Y = 1 \mid \phi) = \phi$.

   Consider the observed outcomes of four coin flips being *tails*, *heads*, *tails*, *tails*, $\mathcal{D}_A = [y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}] = [0, 1, 0, 0]$.

   You may include basic arithmetic in your answers.

   (a) (4 points) MLE

   Write an expression for the likelihood for this dataset $p(\mathcal{D}_A \mid \phi)$ in terms of $\phi$. Don't include any symbols for the data such as $\mathcal{D}_A$ or $y$.

   $p(\mathcal{D} \mid \phi)$:

   For this data, $\mathcal{D}_A$, what is the maximum likelihood estimate of $\phi$?

   $\hat{\phi}_{\text{MLE}}$:

   (b) (6 points) MAP

   We now add a prior on the parameter $\phi$, specifically $p(\phi) = 2\phi$.

   Write an expression for $p(\mathcal{D}_A, \phi)$ in terms of $\phi$. Don't include any symbols for the data such as $\mathcal{D}_A$ or $y$.

   $p(\mathcal{D}, \phi)$:

   Write an expression for the objective function $J(\phi) = -\log p(\mathcal{D}_A, \phi)$ in terms of $\phi$. Don't include any symbols for the data such as $\mathcal{D}_A$ or $y$.

   $J(\phi) = -\log p(\mathcal{D}, \phi)$:

Write an expression for the derivative of the objective function with respect to $\phi$, $dJ/d\phi$.

$dJ/d\phi$:

For this data, $\mathcal{D}_A = [0, 1, 0, 0]$, what is the MAP estimate of $\phi$?

$\hat{\phi}_{\text{MAP}}$:

(c) (4 points) MAP with more points

Now, we flip the coin a total of 99 times and end up with dataset $\mathcal{D}_B$ containing 11 heads and 88 tails.

Using the same prior as before, $p(\phi) = 2\phi$, compute the MAP estimate for this new dataset, $\mathcal{D}_B$.

$\hat{\phi}_{\text{MAP}}$:

Finally, we flip the coin a total of 999 times and end up with dataset $\mathcal{D}_C$ containing 111 heads and 888 tails.

Using the same prior as before, $p(\phi) = 2\phi$, compute the MAP estimate for this new dataset, $\mathcal{D}_C$.

$\hat{\phi}_{\text{MAP}}$:

# 4 Naive Bayes

1. (8 points) We are given a database of vehicles and their features with a theft record of whether they were stolen or not.

| **Example** | **Color** | **Type** | **Origin** | **Stolen?** |
|---|---|---|---|---|
| 1 | Red | Sports | Domestic | Yes |
| 2 | Red | Sports | Domestic | No |
| 3 | Red | Sports | Domestic | Yes |
| 4 | Yellow | Sports | Domestic | No |
| 5 | Yellow | Sports | Imported | Yes |
| 6 | Yellow | SUV | Imported | No |
| 7 | Yellow | SUV | Imported | Yes |
| 8 | Yellow | SUV | Domestic | No |
| 9 | Red | SUV | Imported | No |
| 10 | Red | Sports | Imported | Yes |

Use Bernoulli naive Bayes without smoothing to compute the following probabilities given this dataset.

*Note*: Leave your answers as unsimplified arithmetic expressions, i.e. you don't have to simplify your arithmetic down to numerical value. Please show any work.

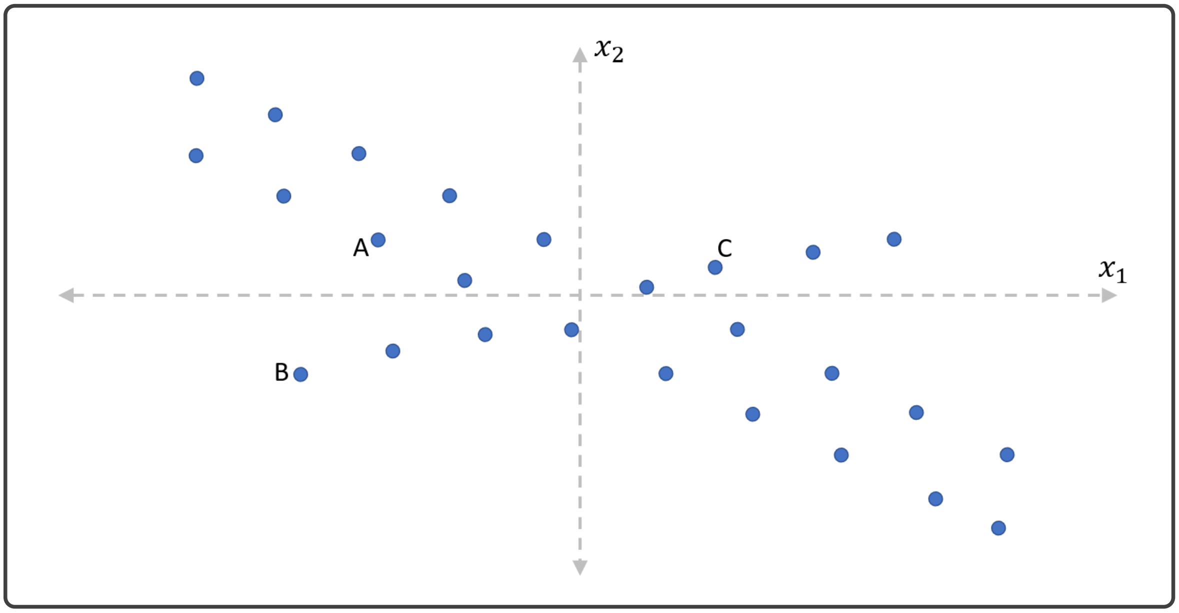
Given a new car $\mathbf{x} = [Red, SUV, Domestic]$, what is $P(Stolen = Yes, \mathbf{x})$?

Given the same new car $\mathbf{x} = [Red, SUV, Domestic]$, what is $P(Stolen = Yes \mid \mathbf{x})$?

# 5 PCA

1. (6 points) On the figure below, draw the following:

   - The first principal component: as an arrow labeled 1
   - The second principal component: as an arrow labeled 2
   - The reconstructed points for the 3 data points labeled A, B, C after projecting onto the first principal component
   - The reconstruction error lines connecting points A, B, C and their reconstructed points after projecting onto the first principal component



# 6 Recommender Systems

1. (3 points) Given a sparse set of training recommendations, $r_{i,j}$, for $N$ users and $M$ items, assume that you have already solved for the matrix factorization of $R \in R^{N \times M}$ and now have matrices $U \in R^{N \times K}$ and $V \in R^{M \times K}$.

   Now, you are given a new user, user $N+1$, that was not considered at all during your training. You collect reviews from that user on the first 5 items. Describe precisely how you would go about recommending the single item that the user is most likely to rate the highest (other than the first 5 items of course).

2. (6 points) Consider the following ratings matrix, where '?' indicates that no rating has been given:

$$R = \begin{bmatrix} 4 & ? & ? \\ ? & 2 & ? \\ ? & 5 & ? \\ ? & ? & 3 \end{bmatrix}$$

   We would like to use matrix factorization to embed our users and items in a $K = 2$ dimensional space. Write specific $U$ and $V$ matrices that optimize the matrix factorization objective function $J(U, V)$ for $K = 2$:

$$J(U, V) = \frac{1}{2} \sum_{i,j \in \mathcal{S}} (r_{i,j} - \mathbf{u}_i^T \mathbf{v}_j)^2$$

   where, as usual, $\mathbf{u}_i^T$ is the $i$-th row of $U$, $\mathbf{v}_j^T$ is the $j$-th row of $V$, and $\mathcal{S}$ is the set of indices, $i, j$, for ratings that are not '?'. *Note*: There is more than one correct answer.

   Write your answers as matrices filled with numerical values.

   $U$: $V$:

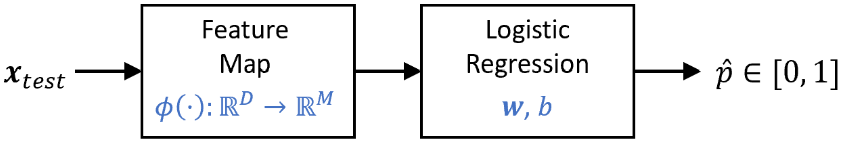
# 7 Applying ML

Once a machine learning system has been trained, it is important to understand how to use it. For the following trained systems, give a precise mathematical equation to compute the output for a given input.

If you would like to use any temporary variables, be sure to give the equation for that variable AND define the size of the variable, e.g. scalar or $\in R^N$, etc.

We'll start with an example, so you know what to expect.

**Logistic Regression** (example)



System: We applied a feature map to our input data, $\phi(\cdot)$, and then trained a binary logistic regression model on the mapped features. The logistic regression model included a bias term.

Test: For a new input, $\mathbf{x}_{test}$, give an equation for $\hat{p}$, the predicted probability of $\mathbf{x}_{test}$ belonging to class $Y = 1$.

Example acceptable answer:

$\hat{p} = \frac{1}{1+e^{-(w^T\phi(x_{test})+b)}}$

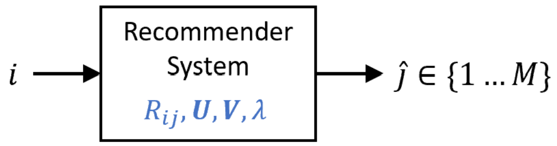
Example insufficient answers:

$\hat{p} = g(w^T\phi(x_{test}) + b)$ // Issue: Using undefined function $g$

$\hat{p} = \frac{1}{1+e^{-z}}$ // Issue: Using undefined variable $z$

$\hat{p} =$ the logistic function of the weight vector dotted with the feature map vector plus the bias term // Issue: Using words, not precise math

1. (3 points) **Recommender Systems**


$i$
Recommender System
$R_{ij}, \boldsymbol{U}, \boldsymbol{V}, \lambda$
$\hat{j} \in \{1 \dots M\}$

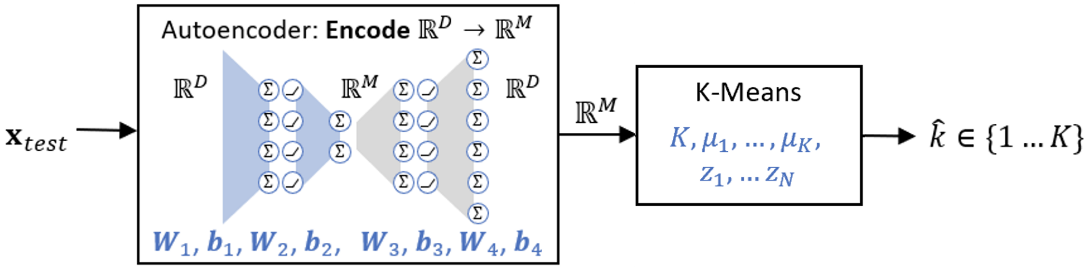
System: We solved the following regularized matrix factorization optimization for a given sparse set of ratings $\{R_{ij}\}_{i,j \in \mathcal{S}}$, where $\mathcal{S}$ is the set of all pairs $(i, j)$ where there exists a rating from the $i$-th user on the $j$-th item.

$\text{argmin}_{\mathbf{U},\mathbf{V}} \sum_{i,j \in \mathcal{S}} (R_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \lambda(\sum_i ||\mathbf{u}_i||_2^2 + \sum_j ||\mathbf{v}_j||_2^2)$

For hyperparameter $\lambda$, the optimization returned an $N \times K$ user matrix, $\mathbf{U}$, and an $M \times K$ item matrix, $\mathbf{V}$, where $\mathbf{u}_i^T$ is the $i$-th row of $\mathbf{U}$ and $\mathbf{v}_j^T$ is the $j$-th row of $\mathbf{V}$.

Test: For user index $i$, give an equation for $\hat{j}$, the index with the highest predicted rating for that user. To simplify things, it is ok if this predicted index is of an item that user $i$ already rated.

2. (6 points) **Grouping Photos**


Autoencoder: **Encode** $\mathbb{R}^D \rightarrow \mathbb{R}^M$
$\mathbb{R}^D$
$\mathbb{R}^M$
$\mathbb{R}^D$
$\mathbf{x}_{test}$
$W_1, b_1, W_2, b_2, W_3, b_3, W_4, b_4$
$\mathbb{R}^M$
K-Means
$K, \mu_1, \ldots, \mu_K,$
$z_1, \ldots z_N$
$\hat{k} \in \{1 \ldots K\}$

System: Identify which cluster an input image is likely to belong to.

To do this, we first trained an autoencoder network to map the $D$ number of (vectorized) input pixels into an $M$-dimensional space. Second, we trained a k-means model with $K$ classes on the encoded $M$-dimensional points (not the $D$-dimensional recovered points). Finally, we use the k-means to determine the index, $\hat{k}$, of the cluster that the encoded test point belongs to.

The neural network (encoder then decoder) has a total of four fully-connected layers. The first and third layers use a ReLU activation function; the second and output layers have no activation functions. The trained autoencoder network contains the following parameters: $W_1 \in \mathbb{R}^{L \times D}$, $\mathbf{b}_1 \in \mathbb{R}^L, W_2 \in \mathbb{R}^{M \times L}$, $\mathbf{b}_2 \in \mathbb{R}^M$, $W_3 \in \mathbb{R}^{L \times M}$, $\mathbf{b}_3 \in \mathbb{R}^L, W_4 \in \mathbb{R}^{D \times L}$, $\mathbf{b}_4 \in \mathbb{R}^D$.

The k-means model produced cluster centers, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and cluster assignments, $z_1, \ldots, z_N$, where $N$ is the number of training images.

Test: For an input image in vector form, $\mathbf{x}_{test}$, give an equation for $\hat{k}$, the cluster index that it belongs to.

You may use the following functions in your answer:

- $ReLU(\mathbf{z})$, which applies a ReLU to each entry in the input vector, $\mathbf{z}$, and returns the resulting output vector

*Note*: Given the length of the answer, you should use temporary variables for this question. Be sure to give the equation for that variable AND define the size of the variable, e.g. scalar or $\in R^N$, etc.

# 8 Word Embeddings

1. Consider the following word embeddings SGD objective function for a single pair of tokens in a training corpus with $N$ tokens, where $(i, j)$ is the corresponding pair of indices into a vocabulary of $M$ tokens:

$$J(U, V)^{(i,j)} = -\log g_{softmax}\left(U\mathbf{v}_i\right)$$

where $g_{softmax}$ is the softmax function, $\mathbf{v}_i$ is the $i$-th row of $V$ as a column vector, and $\mathbf{u}_j$ is the $j$-th row of $U$ as a column vector.

Let's define the following:

- $\mathbf{s} = U\mathbf{v}_i$, the score vector for each vocab token
- $\hat{\mathbf{y}} = g_{softmax}\left(\mathbf{s}\right)$, the predicted probability for each vocab token
- $\mathbf{y}$ is the one-hot vector with a one at index $j$.

The following are the corresponding SGD gradient update expressions, derived using the partial derivative of the cross-entropy and softmax functions, $\partial J/\partial\mathbf{s} = \hat{\mathbf{y}} - \mathbf{y}$:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i - \alpha U^{\top}\left(\hat{\mathbf{y}} - \mathbf{y}\right)$$
$$U \leftarrow U - \alpha\left(\hat{\mathbf{y}} - \mathbf{y}\right)\mathbf{v}_i^{\top}$$

(a) (3 points) Give your answers to the following in terms of $N$, $M$, $K$, where $K$ is the number of dimensions of embedded space.

What is the size of $V$:

□

What is the size of $U$:

□

How many iterations are in one epoch of SGD (batch size one)?

□

2. Now, we would like to train a word embeddings model to predict the next word from a context of exactly $T = 3$ previous words rather than just one, as in the model above. To accommodate the larger input context, we'll incorporate uniform attention weights $1/T$ (rather than learned attention) to our model above.

   (a) (2 points) How many parameters are in this model in terms of $N$, $M$, $K$?

   How many iterations are in one epoch of SGD (batch size one)in terms of $N$, $M$, $K$?

   (b) (2 points) Write the equation for the vector $\hat{\mathbf{y}}$ containing predicted next token probabilities for each token in the vocabulary, given input context indices, $i_1, i_2, i_3$. You may use $g_{softmax}$ in your answer.

   (c) (2 points) Write the SGD objective function for four consecutive tokens in the training corpus where $i_1, i_2, i_3, i_4$ are the corresponding vocabulary indices, where $i_4$ is the index of the fourth word. You may write your answer in terms of your definition for $\hat{\mathbf{y}}$ above.

(d) (3 points) Write the SGD gradient update expressions for one tuple of vocabulary indices, $i_1, i_2, i_3, i_4$, and learning rate $\alpha$. You may write your answer in terms of your definition for $\hat{\mathbf{y}}$ above as well as the one-hot vector $\mathbf{y}$ that has a one at index $i_4$.

# 9 Variational Autoencoders

In class, we derived the evidence lower bound (ELBO) of the autoencoder log likelihood using variational inference, including Jensen's inequality:

$$\log p(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log p_\theta(x \mid z)\right] - D_{\mathrm{KL}}\left(q_\phi(z \mid x) \,\|\, p(z)\right)$$

where the above KL divergence is:

$$D_{\mathrm{KL}}(q(z \mid x) \parallel p(z)) = \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log\left(\frac{q(z \mid x)}{p(z)}\right)\right]$$

This ELBO is composed of a reconstruction loss and the KL divergence of $q_\phi(z \mid x)$ from $p(z)$.

1. (10 points) In this question, we will derive a different relationship between the log likelihood, $\log p(x)$, and the ELBO, starting with a different KL divergence than the one used above.

   Prove that the KL divergence of $q(z \mid x)$ from $p(z \mid x)$ (not $p(x)$) is equal to the log likelihood minus the ELBO.

   Specifically, derive the following equality:

   $$D_{\mathrm{KL}}(q(z \mid x) \parallel p(z \mid z)) = \log p(x) - \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log p_\theta(x \mid z)\right] + D_{\mathrm{KL}}\left(q_\phi(z \mid x) \,\|\, p(z)\right)$$

   *Hint 1*: It is an equality, not an inequality, so no need for Jensen's inequality or other lower bounds. *Hint 2*: You will need to apply Bayes rule.

This page intentionally left blank.