Name: ______________________________ NetID: ____________

S&DS 365 / 665

Intermediate Machine Learning

Midterm Exam

October 17, 2022

Complete all of the problems. You have 75 minutes to complete the exam.

The exam is closed book, computer, phone, etc. You are allowed one double-sided $8\frac{1}{2} \times 11$ sheet of paper with hand-written notes. No calculators—one problem requires some multiplication and addition that you can do on paper.

The following facts may (or may not) be helpful:

- If (X_1, X_2) are jointly Gaussian with distribution

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} A & C \\ C^T & B \end{pmatrix}\right)$$

  then the conditional distributions are also Gaussian and given by

$$\begin{aligned} X_1 \,|\, x_2 &\sim N\left(\mu_1 + CB^{-1}(x_2 - \mu_2),\ A - CB^{-1}C^T\right) \\ X_2 \,|\, x_1 &\sim N\left(\mu_2 + C^T A^{-1}(x_1 - \mu_1),\ B - C^T A^{-1} C\right) \end{aligned}$$

- The function `numpy.random.choice(a, size, p)` generates a random sample of size `size` by sampling elements of a given array `a`, with weights `p`.

1. ***Multinomial choice*** (10 points)

For each of the following questions, circle the *single best* answer.

1.1. Consider the toy lasso problem

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{2}(Y - \beta)^2 + |\beta|$$

where Y is a random variable and β is a scalar. If $Y = -2$ the solution is

(a) $\widehat{\beta} = 0$
(b) $\widehat{\beta} = -1$
(c) $\widehat{\beta} = -3$
(d) $\widehat{\beta} = 1$

1.2. Suppose that we have a kernel regression technique in one dimension with bandwidth parameter h for which the squared bias scales as $O(h^3)$ and the variance scales as $O\left(\frac{1}{nh}\right)$ as $h \to 0$ with $nh \to \infty$, for a sample of size n, under certain assumptions. What is the fastest rate at which the risk (expected squared error) will decrease with sample size for this technique?

(a) $O(n^{-1/3})$
(b) $O(n^{-1/4})$
(c) $O(n^{-3/4})$
(d) $O(n^{-1})$

1.3. Suppose that we fit a Gaussian process regression model using training data X and y with noise level σ^2, and we use the model to predict for a test set X'. Let $\mathbb{K}_{ij} = K(X_i, X_j)$ and $\mathbb{K}'_{ij} = K(X'_i, X_j)$. Then the posterior mean of the regression function $\widehat{m}(X')$ is given by

(a) $\mathbb{K}(\mathbb{K} + \sigma^2 I)^{-1} Y$
(b) $\mathbb{K}(\mathbb{K}' + \sigma^2 I)^{-1} Y$
(c) $\mathbb{K}'(\mathbb{K} + \sigma^2 I)^{-1} Y$
(d) None of the above

1.4. Consider the minimum norm linear model $\widehat{\beta}_{\text{mn}} = \mathbb{X}^T(\mathbb{X}\mathbb{X}^T)^{-1}y$ defined for $p > n$, where $\mathbb{X}$ is the $n \times p$ design matrix. Any other interpolating solution $\mathbb{X}\widehat{\beta} = y$ satisfies

(a) $(\widehat{\beta} - \widehat{\beta}_{\text{mn}})^T \widehat{\beta}_{\text{mn}} > 0$

(b) $(\widehat{\beta} - \widehat{\beta}_{\text{mn}})^T \widehat{\beta}_{\text{mn}} = 0$

(c) $(\widehat{\beta} - \widehat{\beta}_{\text{mn}})^T \widehat{\beta}_{\text{mn}} < 0$

(d) $(\widehat{\beta} - \widehat{\beta}_{\text{mn}})^T \widehat{\beta}_{\text{mn}}$ could be positive or negative

1.5. Consider a model $p(x, z) = p(z)\, p(x \,|\, z)$ where x is observed data and z is a latent variable. In variational inference one forms a variational distribution $q(z \,|\, x)$ to approximate $p(z \,|\, x)$. Consider the following expression:

$$\text{LBO} = \mathbb{E}_q \log p(x \,|\, z) + H(q)$$

where $H(q)$ is the entropy. What is the missing expectation term from the ELBO? That is, LBO $+\ ?\ =$ ELBO. Choose from the following.

(a) $\mathbb{E}_q \log p(x, z)$

(b) $\mathbb{E}_q \log p(z)$

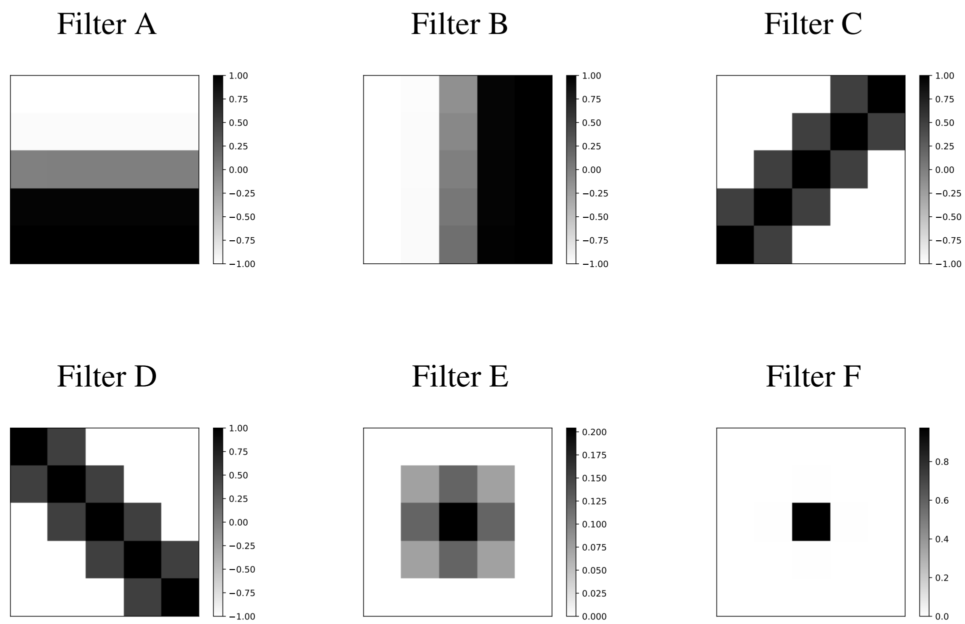
(c) $-\mathbb{E}_q \log p(z)$

(d) $-\mathbb{E}_p \log q(z \,|\, x)$

2. ***Convolutional neural networks*** (10 points)

Convolutional neural networks (CNNs) work by learning a set of kernel functions or "filters" that are swept across an image to create a "feature map." Some of the filters can be difficult to interpret; others are more interpretable.

(a) The figure below shows six 5×5 filters:

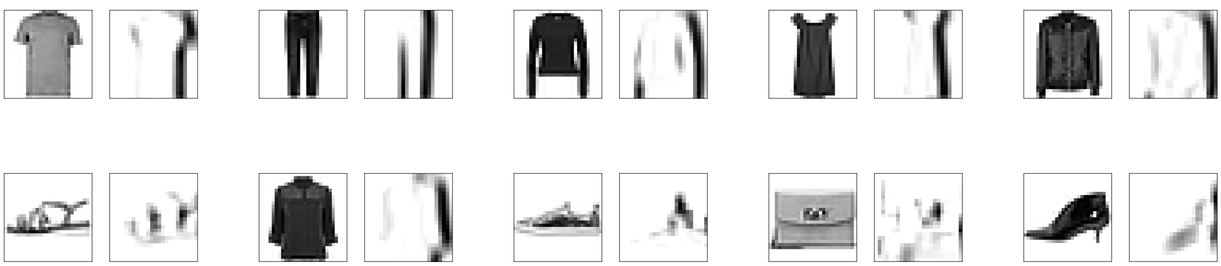


On the following two pages, a set of images is shown, together with the feature map generated by applying one of these filters. Each filter is applied using the ReLU activation function, with a particular value of an intercept b.

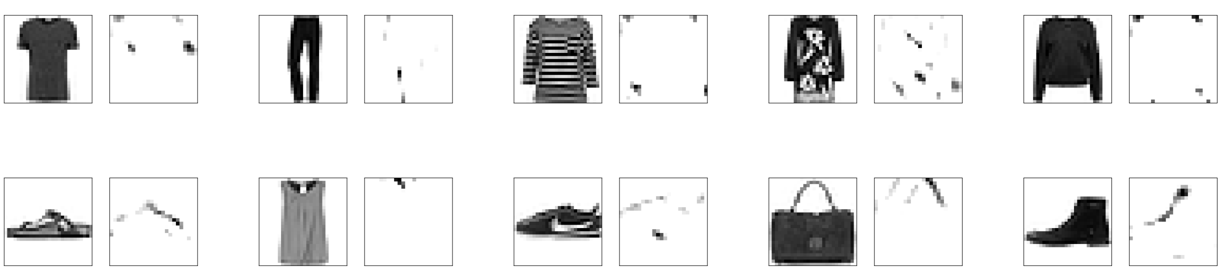
Write the name of the filter that generated each group of maps. Each filter is applied to only one group of images.

Feature map 1. Filter used: ____________

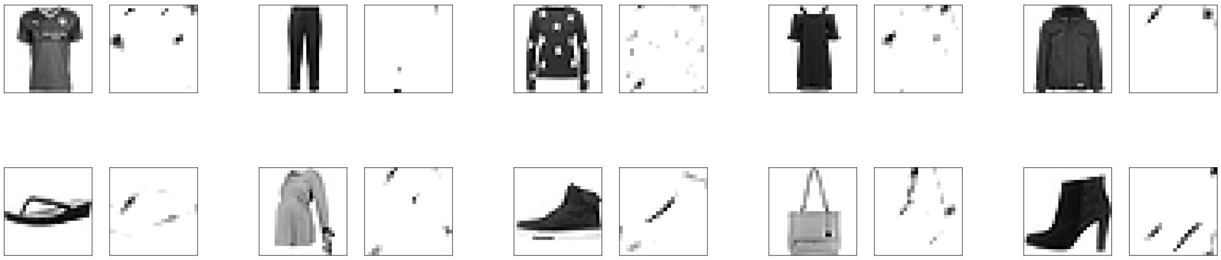
Feature map 2. Filter used: ____________



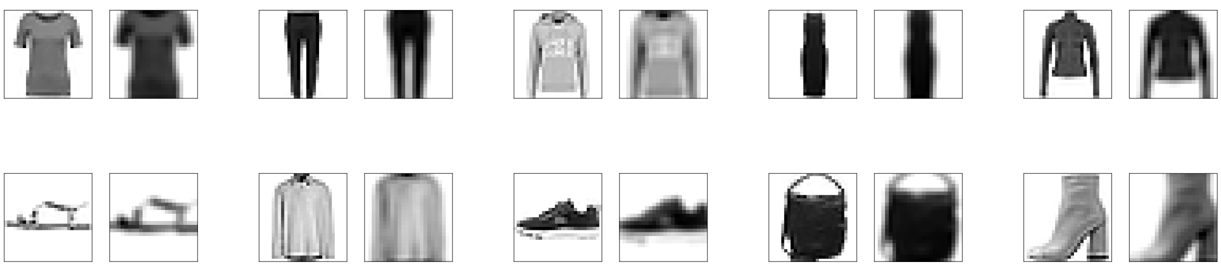
Feature map 3. Filter used: ____________



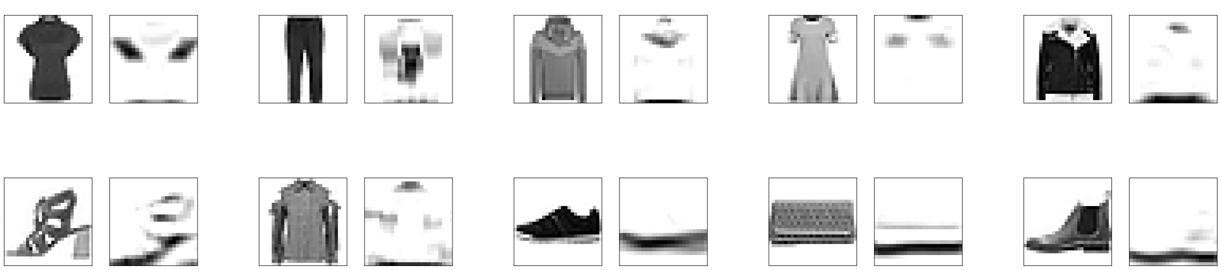
Feature map 4. Filter used: ____________



Feature map 5. Filter used: ____________



Feature map 6. Filter used: ____________



(b) The following TensorFlow code, similar to that used on assignmnent 2 and in class, constructs a CNN to classify 64×64 color images, using two convolutional layers followed by a dense layer.

```
from tensorflow.keras import layers, models

model = models.Sequential()
model.add(layers.Conv2D(10, (5, 5), input_shape=(64, 64, 3)))
model.add(layers.MaxPooling2D((4, 4)))
model.add(layers.Conv2D(20, (6, 6)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(rate=.5))
model.add(layers.Flatten())
model.add(layers.Dense(2))
```

Indicate the shape of the output tensor for each layer, together with the number of trainable parameters, by filling in the two missing fields for each of the rows below. For partial credit if an answer is wrong, you may show your work below the table. No calculators.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)		
max_pooling2d (MaxPooling2D)		
conv2d_1 (Conv2D)		
max_pooling2d_1 (MaxPooling2D)		
dropout (Dropout)		
flatten (Flatten)		
dense (Dense)		

3. ***Short answer*** (6 points)

The following two subproblems ask you to explain the important concepts associated with two topics that have been central to the first part of the course.

Answer only one of the following two questions. If two answers are given only the Lasso problem will be graded.

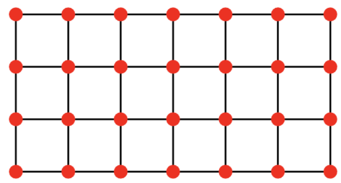
(a) ***Lasso***. Describe the role of the Lasso in machine learning, answering each of the following questions: (1) What is the definition of the Lasso? (2) What is the main purpose of the Lasso? (3) What are the steps used in applying the Lasso to a data set?

(b) ***Smoothing vs. Mercer kernels***. Compare smoothing kernels and Mercer kernels, answering each of the following questions. (1) What are two similarities and two differences between the two types of kernels? (2) What is an example of the use of smoothing kernels? (3) What is an example of the use of Mercer kernels?

4. ***Implementation and Gibbs sampling*** (10 points)

The Ising model, as discussed in class, is a model that is central to physics, social network analysis, and many other areas.

For this problem, we will consider a grid graph, such as the one shown below, where each node is connected to the nodes immediately above and below it, and to its left and right:



Each node position (x, y) in the graph, $Z_{(x,y)} \in \{-1, 1\}$ is a random variable that is either 1 or -1. The joint probability takes the form

$$p(Z) \;\propto\; \exp\left(\sum_{(x,y)\in V} \alpha Z_{(x,y)} + \sum_{((x,y),(x',y'))\in E} \beta Z_{(x,y)} Z_{(x',y')} \right)$$

where α is a scalar parameter, assigning a weight α to each variable (indexed by vertices V), and β is second scalar parameter, assigning a weight β to the product of each pair of neighboring variables in the grid graph (indexed by edges E).

Recall that the Gibbs sampling algorithm draws from this distribution by repeatedly visiting nodes (x, y), and sampling $Z_{(x,y)}$ while holding all of the other $Z_{(x',y')}$ values fixed.

(a) Suppose that we are running a Gibbs sampling step on a node (x, y) that has four neighbors, with

$$Z_{(x-1,y)} = Z_{(x+1,y)} = 1$$
$$Z_{(x,y-1)} = Z_{(x,y+1)} = -1.$$

In this Gibbs sampling step, what is the probability that $Z_{(x,y)}$ is set to 1?

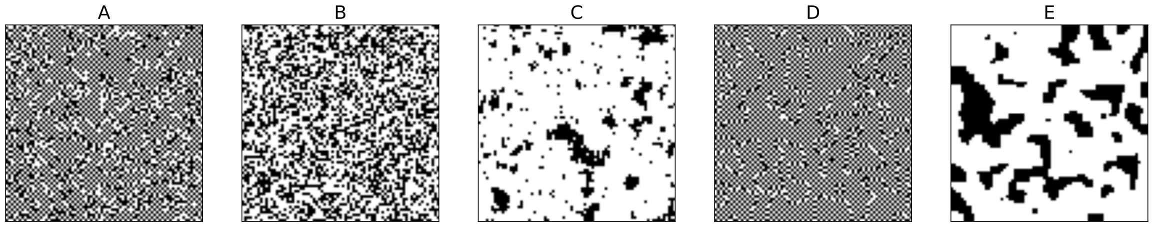
(a) $1/(1 + e^{-2\alpha-2\beta})$
(b) $1/(1 + e^{-\alpha-3\beta})$
(c) $1/(1 + e^{-2\alpha})$
(d) $1/2$

The following code partially implements Gibbs sampling for this model. Your job is to complete the implementation by providing three additional lines of code.

```
import numpy as np

def gibbs_step(Z, alpha, beta, x, y):
    # initialize the exponent variable with one line
    # line 1:
    exponent = ...
    for dx, dy in [(-1,0), (1,0), (0,-1), (0,1)]:
        if ((x + dx < 0) | (x + dx >= Z.shape[0]) |
             (y + dy < 0) | (y + dy >= Z.shape[1])):
            continue
        # finish the loop with one line
        # line 2:
        exponent = ...

    # set the probability p
    # line 3:
    p = ...

    # np.random.rand() draws a random uniform variable
    if np.random.rand() <= p:
        Z_new = 1
    else:
        Z_new = -1
    return Z_new


def run_gibbs_sampling(Z, alpha, beta, steps):
   for _ in np.arange(steps*np.prod(Z.shape)):
      x = np.random.choice(range(Z.shape[0]))
      y = np.random.choice(range(Z.shape[0]))
      Z[x, y] = gibbs_step(Z, alpha, beta, x, y)
```

(b) Complete the implementation, by writing the three missing lines below.

```
line 1:

line 2:

line 3:
```



(c) The figure above shows the result of running Gibbs sampling on a 75×75 grid, using a fixed value of α and five different values of β:

$$\beta \in \left\{-5, -\tfrac{1}{2}, 0, \tfrac{1}{2}, 5\right\}$$

Which is which? Explain your answer.

A : $\beta =$

B : $\beta =$

C : $\beta =$

D : $\beta =$

E : $\beta =$

(d) In the above plots, white pixels correspond to $Z[x, y] = 1$ and black pixels correspond to $Z[x, y] = -1$. Was the Gibbs sampling algorithm run with $\alpha > 0$, $\alpha < 0$, or $\alpha = 0$? Explain your answer.

5. ***Kernel hedge*** (6+2 points)

Consider a loss function that combines linear regression and Mercer kernel regression:

$$\widehat{\alpha}, \widehat{\beta} = \underset{\alpha,\beta}{\operatorname{argmin}} \left\{ \frac{1}{n} \|Y - \mathbb{K}\alpha - \mathbb{X}\beta\|^2 + \lambda_1 \alpha^T \mathbb{K} \alpha + \lambda_2 \|\beta\|^2 \right\} \tag{1}$$

where λ_1 and λ_2 are two regularization penalities. Here $\mathbb{X}$ is an $n \times p$ design matrix for data $X_1, \ldots, X_n$ with predictor variables $X_i \in \mathbb{R}^p$, and $\mathbb{K}$ is an $n \times n$ Gram matrix for a Mercer kernel, with entries $\mathbb{K}_{ij} = K(X_i, X_j)$, and Y is an n-vector of response values.

(a) What infinite dimensional optimization over the reproducing kernel Hilbert space for K is expression (1) above solving? Explain.

(b) What is the relationship between this estimator and a Gaussian process? Explain.

(c) (2 points extra credit) Give an iterative two-step algorithm for computing $\widehat{\alpha}$ and $\widehat{\beta}$ that estimates α holding β fixed, and then estimates β holding α fixed. Give explicit formulas for the parameter updates.

Extra work space