Name: ________________________________ NetID: ____________

S&DS 365 / 665

Intermediate Machine Learning

Final Exam

Tuesday, December 20, 2022

Complete all of the problems. You have 2.5 hours (150 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.

Please use a black pen or dark pencil so that your exam scans clearly.

1. ***Multinomial choice*** (25 points)

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

1.1. Suppose the lasso is carried out for a response vector Y on a design matrix $\mathbb{X}$ that satisfies $\mathbb{X}\mathbb{X}^T = I_n$, where I_n is the $n \times n$ identity matrix. Let $\text{Soft}_\lambda(\cdot)$ denote the soft thresholding operator with threshold λ. If the ℓ_1 penalty of the lasso is λ, then the solution $\widehat{\beta}$ satisfies which of the following?

(a) $\widehat{\beta} = \text{Soft}_\lambda(Y)$

(b) $\widehat{\beta} = \text{Soft}_\lambda(\mathbb{X}^T Y)$

(c) $\widehat{\beta} = \text{Soft}_\lambda(\mathbb{X} Y)$

(d) $\widehat{\beta}$ does not have a closed form

(e) None of the above

1.2. Consider the regression estimator computed as

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

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) $\widehat{\beta} = -\frac{1}{2}$

(d) $\widehat{\beta} = 1$

(e) None of the above

1.3. Suppose that we have a kernel regression technique in one dimension with bandwidth parameter h for which the squared bias scales as $O(\sqrt{h})$ and the variance scales as $O\left(\frac{1}{\sqrt{nh}}\right)$ 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) $O(n^{-1/4})$

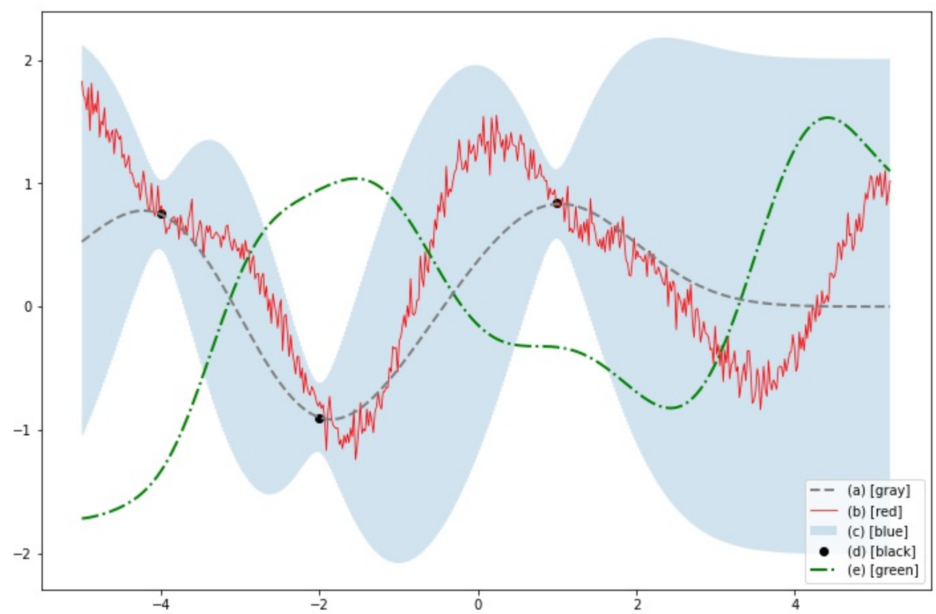
(c) $O(n^{-1/2})$

(d) $O(n^{-1})$

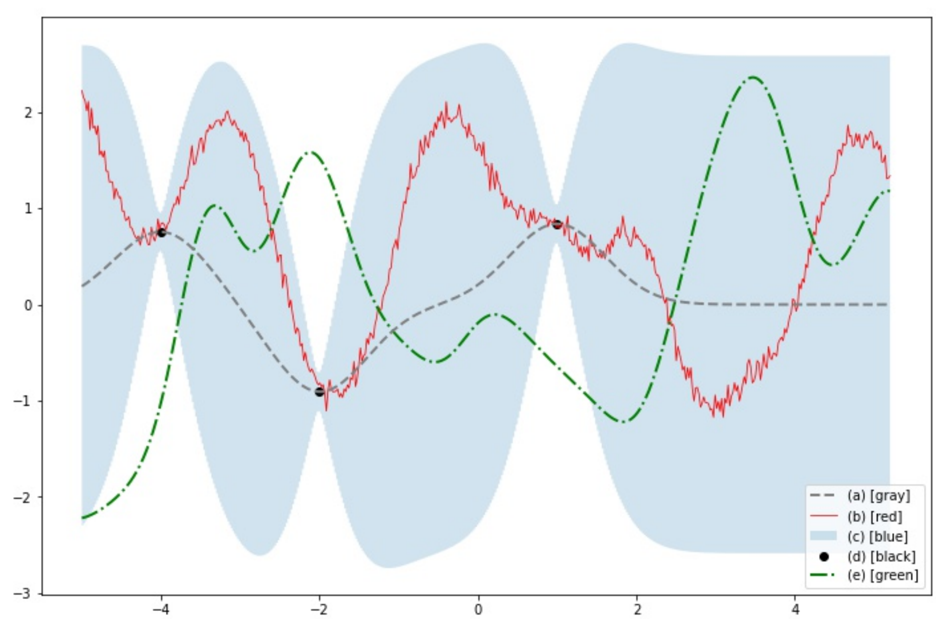
(e) None of the above

1.4. Consider the following two plots associated with a Gaussian process. The data are the same in the two plots; only the hyperparameters are different.

Plot A



Plot B



(1) What quantities are shown in the plots? Write the appropriate label (a), (b), (c), (d), or (e) next to each of the following.

____ Sample from prior

____ Sample from posterior

____ Data

____ Posterior mean

____ Posterior confidence interval

(2) The two plots are generated with different hyperparameters, kernel bandwidths h_A and h_B and measurement noise levels σ_A^2 and σ_B^2. Circle the appropriate relations:

Kernel bandwidth:

$h_A < h_B$ $\qquad$ $h_A > h_B$

Measurement noise:

$\sigma_A^2 < \sigma_B^2$ $\qquad$ $\sigma_A^2 > \sigma_B^2$

(3) Briefly explain your answers to (2) above.

1.5. A convolutional neural network is constructed to classify color images, each with 3 color channels (red, blue, green). The network has two convolutional layers followed by two dense layers.

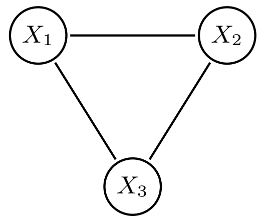
The first convolutional layer uses 10 filters, each 3×3, constructed with a command such as `model.add(layers.Conv2D(10,(3,3)))`. How many trainable parameters does this layer have?

(a) 90
(b) 100
(c) 270
(d) 280
(e) 900
(f) 910
(g) It depends on the number of pixels in the images
(h) It depends on the pooling operation after the first layer

The second convolutional layer also uses 10 filters, each 3×3, constructed with a command such as `model.add(layers.Conv2D(10,(3,3)))`. How many trainable parameters does this layer have?

(a) 90
(b) 100
(c) 270
(d) 280
(e) 900
(f) 910
(g) It depends on the number of pixels in the images
(h) It depends on the pooling operation after the first layer

1.6 Consider a graph neural network for binary classification of inputs $x = (x_1, x_2, x_3)^T$ on the graph below:

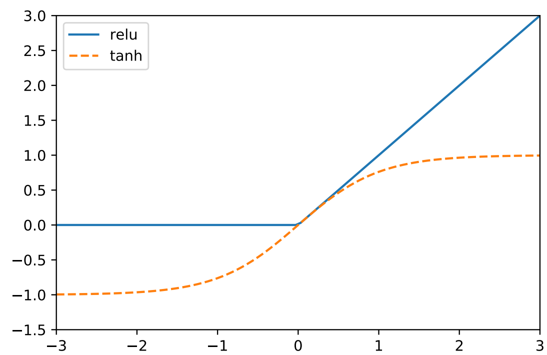


The graph neural network defines a classifier with discriminant function

$$\log\left(\frac{p(Y=1\,|\,x)}{p(Y=0\,|\,x)}\right) = \beta^T h(x)$$

with $h(x) = \varphi(Lx)$ where φ is an activation function and L is the graph Laplacian (with edge weights 1), and $\beta = (1, -1, 1)^T$.

Recall the relu and tanh activation functions look like this, with $\tanh(-x) = -\tanh(x)$:



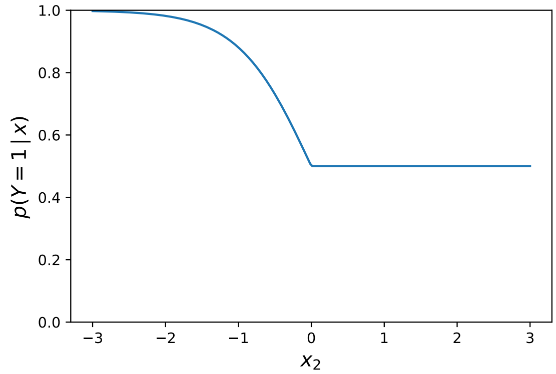
TRUE FALSE (1) If φ is relu and $x = (1, 2, 1)$ then $p(Y=1\,|\,x) > \frac{1}{2}$

TRUE FALSE (2) If φ is tanh and $x = (1, -2, 1)$ then $p(Y=1\,|\,x) > \frac{1}{2}$

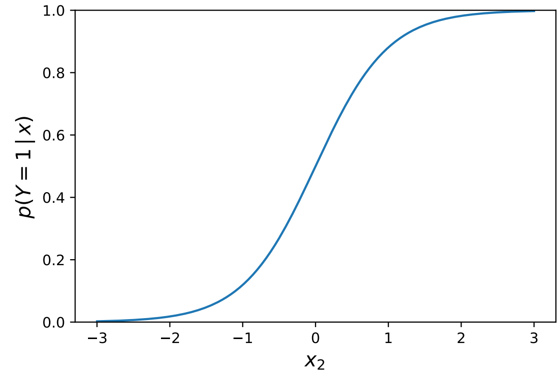
TRUE FALSE (3) If φ is tanh and $x = (1, 1, 1)$ then $p(Y=1\,|\,x) > \frac{1}{2}$

(4) If φ is relu and $x = (0, x_2, 0)^T$, which of the following is a plot of the probability $p(Y = 1 \mid x)$ as a function of x_2?

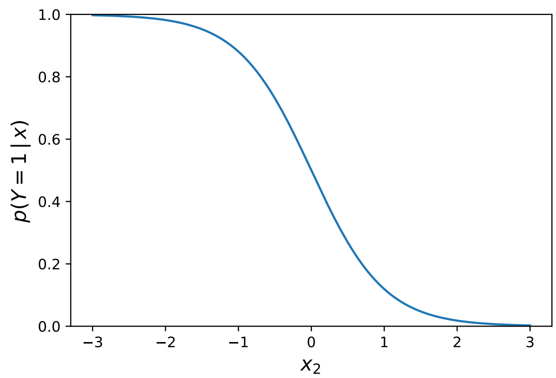
(a)



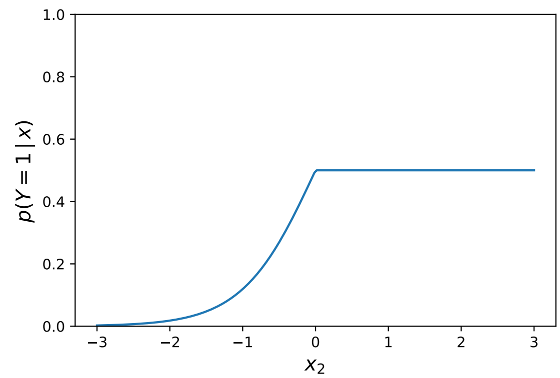
(b)



(c)



(d)



2. ***Short answer*** (18 points)

The following three subproblems ask you to explain important concepts associated with topics in the course.

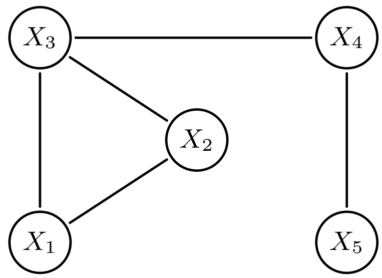
(a) ***Policy-gradient algorithm for reinforcement learning***. (1) What problem is the policy gradient algorithm designed to solve, compared with the policy iteration algorithm? (2) How is it different from actor-critic and Q-learning? (3) Write down mathematical expressions for the parameter update equations used by the policy-gradient algorithm.

(b) ***Gated recurrent units (GRUs)***. (1) Describe what problem GRUs are designed to solve. (2) Explain the two types of gates used in a GRU, and how they are implemented. (3) Write down mathematical expressions for the update equations for computing the hidden state vector, assuming one hidden layer.

(c) ***Attention and transformers***. (1) Explain the idea of attention, and the problem it is designed to solve. (2) Write down how attention is implemented in terms of queries, keys, and values. (3) Write down a mathematical expression for how kernel regression can be expressed in terms of attention — what are the queries, keys, and values?

3. ***Graphical models*** (10 points)

Consider the following graph on five variables:

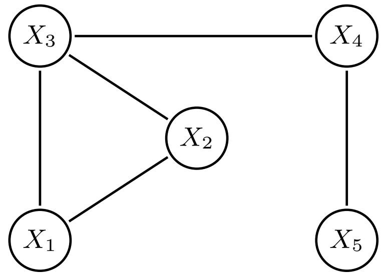


(a) Write down the general form for a strictly positive probability density that factors (is Markov) with respect to this graph.

$p(x_1, x_2, x_3, x_4, x_5) =$

(b) If a multivariate Gaussian has this graph, what is the form of the sparsity pattern of the precision matrix Ω? Enter 0 for a zero entry and $*$ for a non-zero entry.

$$\Omega = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \left[\begin{array}{ccccc} * & & & & \\ & * & & & \\ & & * & & \\ & & & * & \\ & & & & * \end{array}\right] \end{array}$$



(c) Based on this graph (repeated above) which of the following independence relations hold? Circle all that apply.

(1) $X_1 \perp\!\!\!\perp X_5$

(2) $X_3 \perp\!\!\!\perp X_5 \mid X_4$

(3) $X_3 \perp\!\!\!\perp X_4 \mid X_5$

(4) $X_2 \perp\!\!\!\perp X_5 \mid X_3$

(5) $X_2 \perp\!\!\!\perp X_5 \mid X_3, X_4$

(6) $X_1 \perp\!\!\!\perp X_5 \mid X_2, X_4$

4. ***Coding: Deep Q-learning*** (10 points)

In this problem you are asked to complete the implementation of some code for deep Q-learning, which only looks at the current reward, not future rewards. This corresponds to discount factor $\gamma = 0$.

The environment is specified by functions that begin `env_` such as `env_num_actions()` and `env_state_dim()` that specify the number of actions and dimension of the state. The function `env_step()` returns the reward and next state.

You need to provide the missing five lines in the code below.

```
def construct_q_network(state_dim, num_actions):
  inputs = layers.Input(shape=(state_dim,))
  hidden1 = layers.Dense(32, activation="relu")(inputs)
  hidden2 = layers.Dense(16, activation="relu")(hidden1)
  q_values = ... # line (a)
  deep_q_network = keras.Model(inputs=inputs, outputs=q_values)
  return deep_q_network

def loss_function(q_value, reward):
  loss = ... # line (b)
  return loss

exploration_rate = 0.1
num_steps = 2000

num_actions = env_num_actions()
q_network = construct_q_network(env_state_dim(), num_actions)
opt = tf.keras.optimizers.Adam(learning_rate=0.01)
state = env_init_state()

for i in range(num_steps):
  with tf.GradientTape() as tape:
    q_values = q_network(state)

    u = np.random.uniform()
    if u <= exploration_rate:
      action = np.random.choice(num_actions)
    else:
      action = ... # line (c)

    reward, next_state = env_step(action)
    q_value = q_values[0, action]
    loss = loss_function(q_value, reward)

    grads = ... # line (d)
    opt.apply_gradients(zip(grads, q_network.trainable_variables))

    state = ... # line (e)
```

Complete the implementation by giving *a single line of Python code* for each of the five missing lines above. For full credit your Python expressions should be syntactically correct.

`line (a):`

`line (b):`

`line (c):`

`line (d):`

`line (e):`

5. ***Reinforcement learning*** (15 points)

The following three subproblems are on the topic of reinforcement learning using policy iteration.

(1) Give the algorithm for policy iteration, and outline a proof that it converges.

(2) Recall the Taxi problem discussed in class. A taxi navigates a 5×5 grid, picking up passengers and delivering them to their desired destinations. There is a default per-step reward of -1, and a reward of $+20$ for delivering the passenger. An illegal pickup or drop-off action has a reward of -10.
The state can be represented as a tuple

`(row, col, passenger_location, destination).`

The "ascii art" figure on the left below shows the grid environment the taxi navigates. Vertical lines | are barriers, which prevent the taxi from moving horizontally. The agent (taxi) can take the following six actions shown below.

```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

0: move south (down)
1: move north (up)
2: move east (right)
3: move west (left)
4: pickup passenger
5: drop off passenger

The passenger is either waiting to be picked up at one of four locations, or is in the taxi being driven to their desired destination. So, for a fixed passenger location and destination, an optimal policy $\pi^*(\texttt{row}, \texttt{col})$ assigns an action to each of the $25 = 5 \times 5$ grid points.

For each of the two tables below, write an integer in each cell for the action to be taken by an optimal policy π^*. We have filled in two actions, which indicate whether the passenger is in the taxi or waiting to be picked up.

				4

passenger is outside the taxi

5				

passenger is inside the taxi

(3) Suppose that an environment has three states, $s = 0$, $s = 1$, and $s = 2$, and an agent can take two actions $a = 0$ and $a = 1$. The environment is deterministic and has the following `reward` and `next` functions:

$$\texttt{next}(s,a): \begin{array}{c} \\ {\scriptstyle s=0} \\ {\scriptstyle s=1} \\ {\scriptstyle s=2} \end{array} \begin{array}{c} \begin{array}{cc} {\scriptstyle a=0} & {\scriptstyle a=1} \end{array} \\ \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \end{pmatrix} \end{array} \qquad \texttt{reward}(s,a): \begin{array}{c} \\ {\scriptstyle s=0} \\ {\scriptstyle s=1} \\ {\scriptstyle s=2} \end{array} \begin{array}{c} \begin{array}{cc} {\scriptstyle a=0} & {\scriptstyle a=1} \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

Give an optimal (deterministic) policy $\pi^*(s)$ using discount $\gamma = \frac{2}{3}$. Write your answer in the following table, showing which action to take in each state. Show your work for partial credit.

$$\pi^*(s): \begin{array}{c} {\scriptstyle s=0} \\ {\scriptstyle s=1} \\ {\scriptstyle s=2} \end{array} \begin{pmatrix} \quad \\ \quad \\ \quad \end{pmatrix}$$

Extra work space