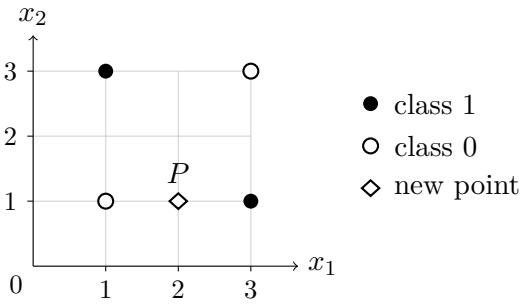
Question	**Page**
1. Perceptrons and backpropagation	3
2. Convolutional networks	5
3. Downsampling and pooling	6
4. Diagnosing neural networks	7
5. Reinforcement learning	8
6. Genetic algorithms	10
7. Language models	12

1. Perceptrons and backpropagation

A perceptron has weights $(w_1, w_2) = (1, -2)$ and bias $b = -1$. The circles are labeled training points. The diamond marks a new point $P = (2, 1)$.



(a) Which class does the perceptron predict for P?

(b) Perform one update using P with learning rate $\frac{1}{2}$. For each possible label, start with the original weights and bias. Give the updated weights and bias. Does the prediction for P change?

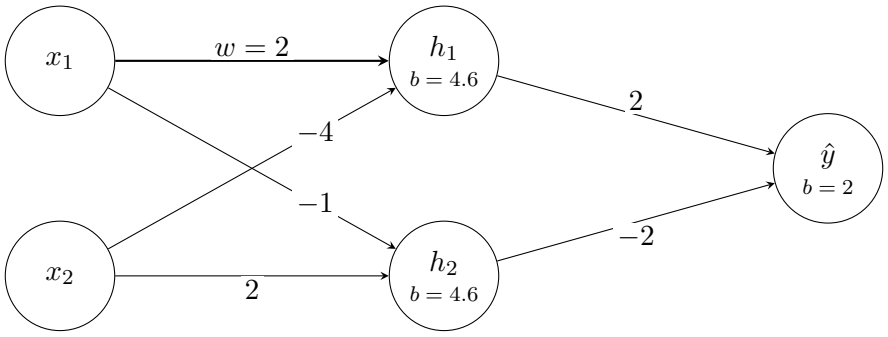
If the true label is 0

If the true label is 1

(c) Can a single perceptron correctly classify all four circles? Explain.

1. Perceptrons and backpropagation (continued)

For $P = (2, 1)$, the true label is $y = 1$. The network below has a linear output $\hat{y}$ (so there is no activation function applied for the output neuron). It predicts class 1 if $\hat{y} \geq 0$ and class 0 otherwise.



For each part, start from the weights shown and use loss $= \frac{1}{2}(y - \hat{y})^2$ with learning rate $\frac{1}{4}$.

(d) Both hidden neurons use ReLU. Show the forward pass for P. Is the prediction correct? Using the output error $y - \hat{y} = -1$, find w after one update.

$$f(z) = \max(0, z), \qquad f'(z) = \begin{cases} 1 & z > 0, \\ 0 & z < 0. \end{cases}$$

(e) Repeat part (d) using sigmoid hidden neurons and the same output error. Use $\sigma(4.6) \approx 0.99$. Why do the two activations give different-sized updates to w?

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \qquad \sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

2. Convolutional networks

The layer below uses stride 1 and a linear activation. Ignore the bias. There is no padding, so the kernel must fit entirely inside the input image.

Input image (6×8)

0	−1	−1	−1	0	0	+1	0
0	+1	+1	+1	0	−1	0	0
0	−1	−1	−1	0	0	0	+1
+1	0	0	0	+1	+1	+1	0
0	0	−1	0	−1	−1	−1	0
0	+1	0	0	+1	+1	+1	0

Kernel (3×3)

−1	−1	−1
+1	+1	+1
−1	−1	−1

Output image

(a) In the output image, write **H** in the cell with the highest activation and **L** in the cell with the lowest activation.

(b) How many output neurons does this layer have? How many kernel weights?

(c) How many multiplications does this layer perform per image? How many would a fully connected layer perform with the same input and the same number of output neurons? Count all input–weight products.

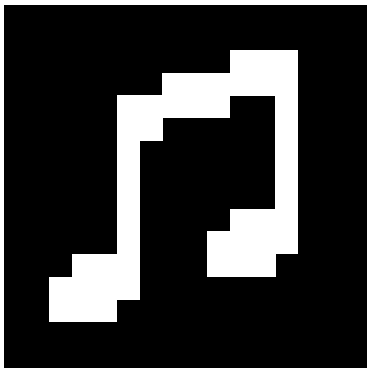
(d) If the stride changes to 2, how many output neurons are there?

(e) With stride 1, suppose the image has three input channels and the layer has two output channels. How many kernel weights and output neurons does the layer have?

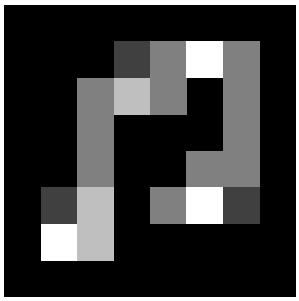
3. Downsampling and pooling

All three methods use stride 2, starting at the upper left. The pooling windows are 2×2.

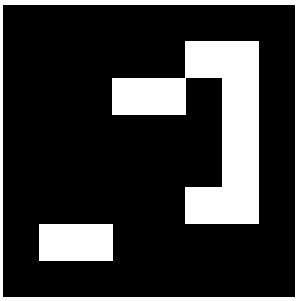
Input image



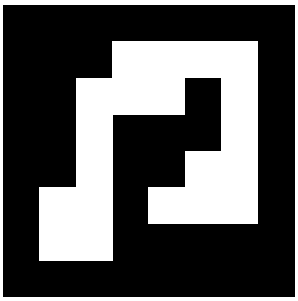
A



B



C



(a) Match A, B, and C to **strided downsampling**, **max pooling**, and **average pooling**.

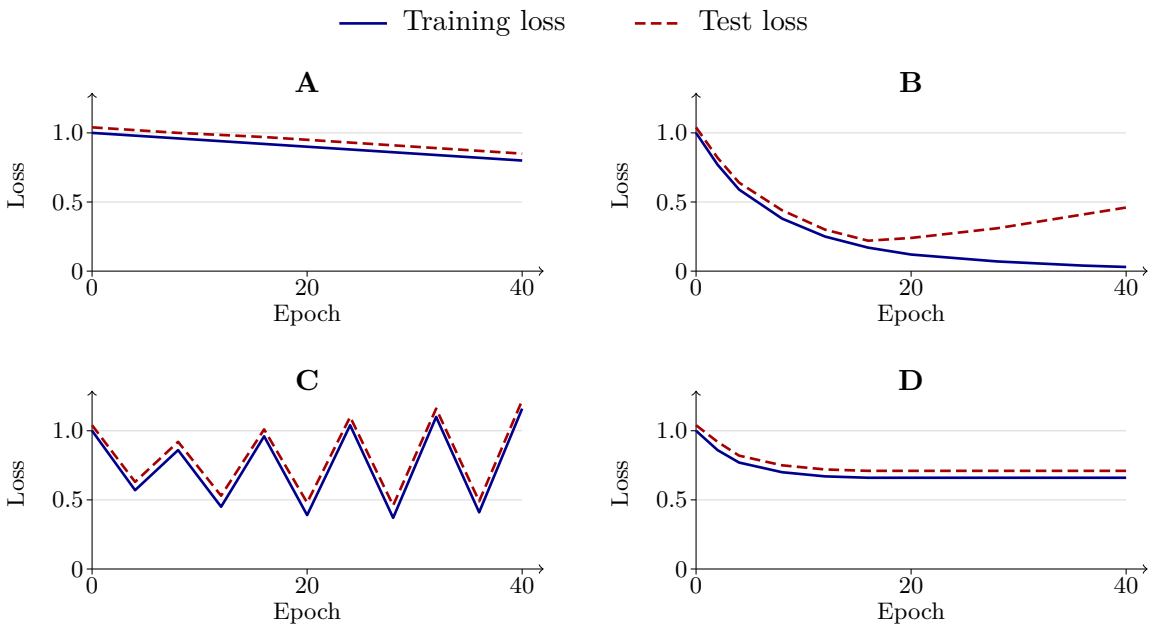
A: _____ B: _____ C: _____

(b) Why might a classifier benefit from **max pooling** on this image? Compare with **average pooling**.

(c) What is the purpose of a **flattening layer** in a convolutional neural network?

4. Diagnosing neural networks

The plots show loss for four classifiers on the same task.



(a) How do training loss and test loss differ? Why measure both?

(b) Match each diagnosis to a plot and give one reason. You may reuse plots.

Diagnosis	Plot	Evidence from the curves
Learning rate too high		
Underfitting		
Learning rate too low		
Overfitting		

(c) Suggest one change to reduce overfitting in the run you identified. Why would it help?

(d) Which network would you expect to have the **highest test accuracy** at epoch 40? Explain.

5. Reinforcement learning

(a) Why use an ϵ-greedy policy when learning values? Why can REINFORCE explore without one?

(b) Start at S with $\epsilon = 0$ and $\alpha = \gamma = 0.5$. Acting in an ordinary square earns -0.2. Entering an exit ends the episode. Its printed payoff is its fixed value. Moves are deterministic. A blocked move leaves the agent in place.

A 0.2	**B** 0.4	**C** 0.6	**Exit** +1
S (start) 0.1	Wall	**D** -0.2	**Exit** -1
E 0.0	**F** 0.1	**G** 0.8	**H** 0.0

Trace the agent's path to an exit. Use a state-value TD update after each move. Show the new value of each state you update.

Path: _____

Value updates:

5. Reinforcement learning (continued)

(c) Start again at S, this time using Q-learning. Each entry is the Q-value for the move in that direction. Use the same settings as in part (b), including the printed exit value for the final update.

0 0 **A** 0.1 0	0 0.4 **B** 0.2 0	0.1 0.6 **C** 0.8 0	Exit +1
0.2 0 **S** 0 0.4	Wall	0.6 −0.1 **D** −0.6 0.4	Exit −1
0.1 0 **E** 0.6 0	0 0.1 **F** 0.4 0	0.6 0.2 **G** 0.1 0	−0.4 0.2 **H** 0 0

Trace the path to an exit and show each Q-value update.

Path: _____

Q-value updates:

(d) In the cart game, the agent observes continuous positions and velocities and learns about the dynamics only by playing. Why can it train a Q-network from this experience? What would it need to apply value iteration?

(e) How does experience replay help when training the cart's Q-network?

6. Genetic algorithms

Use a genetic algorithm to evolve a controller for the maze robot from Lab 1. Its goal is to reach the light as quickly as possible.

```
def controller(robot):
    left  = robot["left-ir"].get_distance()
    right = robot["right-ir"].get_distance()
    if left < 7:
        robot.move(0.2, -0.6)
    elif right < 9:
        robot.move(0.3,  0.5)
    else:
        robot.move(0.9,  0.0)
```

Sensor readings range from 0 to 20 cm. Both arguments of `move(speed, turn)` range from -1 to 1. Use the rule structure shown.

(a) Write a **chromosome** for this controller. What does each gene represent?

(b) Describe a **mutation** for your encoding. Give one example.

(c) Describe a **crossover** operation for your encoding.

6. Genetic algorithms (continued)

In each run, the robot records the total distance traveled.

(d) What **fitness function** would you maximize for this task?

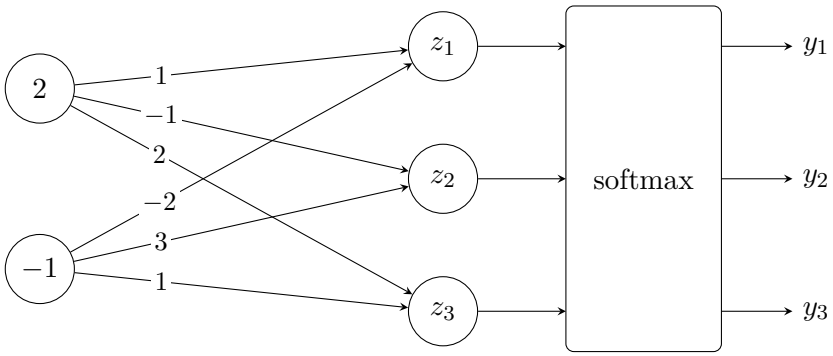
(e) How would you use fitness to **select parents**?

(f) Why might a GA find a better controller than hill climbing with single-gene changes? Give an example from this task.

(g) Based on the building block hypothesis, how would you order the genes in your robot's chromosome for one-point crossover? Justify your choice.

7. Language models

(a) For the network below, what is $y_1 + y_2 + y_3$? _____



(b) Use byte-pair encoding to build a vocabulary of 10 tokens for the text below.

```
The geese see bees
```

Start with one token per distinct character, including the space. Break ties by choosing the tied pair that appears first from the left. Show your merges and list the final vocabulary.

(c) Why might you prefer this tokenizer to a whole-word tokenizer when you encounter the new word `bee`?

(d) With the vocabulary from part (b), how many possible actions does a language-model agent have at each step of next-token generation? _____

(e) How is a token represented at the input to a neural language model? Where does that representation come from?