DSC 140B - Quiz 08

March 12, 2026

Name: ______________________ PID: ______________

About the quizzes:

- Quizzes in DSC 140B are *optional* and graded pass/fail.
- A score of 70% or higher earns a "pass" and 1.5 credits toward your final grade.
- If you don't pass, no credits are earned, but it doesn't hurt your grade.
- You have 30 minutes to complete the quiz.
- At least one of the questions below will be on an exam (probably with slight changes, such as different numbers).
- Unfortunately, we can't answer clarifying questions during the quiz. If you think a question has a bug or is unclear, please let us known in a private post on Campuswire after the quiz, and we'll take it into account when grading.

Problem 1.

A color image of size $64 \times 64 \times 3$ is convolved with a single filter of size $7 \times 7 \times 3$. No padding is applied, and the stride is 1. What is the shape of the output response map?

- ○ $64 \times 64 \times 3$
- ○ $58 \times 58 \times 3$
- ● $58 \times 58 \times 1$
- ○ $57 \times 57 \times 1$

Solution: $58 \times 58 \times 1$.

With no padding and stride 1, the output height and width are each $64 - 7 + 1 = 58$. A single filter produces a single response map (depth 1), regardless of the number of input channels. The filter's depth of 3 matches the input's 3 channels, but the dot product across all channels produces one scalar per spatial position.

This was similar to Practice Problem 161.

Problem 2.

Suppose you have a tabular dataset of 10,000 houses. Each row represents a house, and the columns contain features such as the number of bedrooms, the square footage, the year built, the lot size, and the sale price. You would like to predict the sale price from the other features.

True or False: a convolutional neural network (CNN) would be an appropriate model for this task.

- ○ True
- ● False

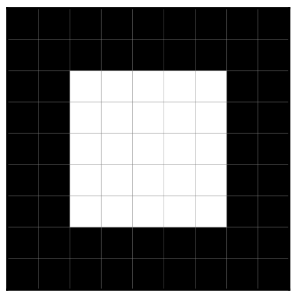
Solution: False.

CNNs are designed to exploit *spatial structure* in the input: nearby entries (e.g., neighboring pixels in an image) are meaningfully related, and the same local patterns can appear at different positions. In a tabular dataset, the columns represent distinct, unrelated features (e.g., number of bedrooms and year built) with no spatial relationship. Reordering the columns would not change the meaning of the data, but it would change the output of a convolutional filter. A standard fully connected neural network (or another model suited to tabular data) would be more appropriate.

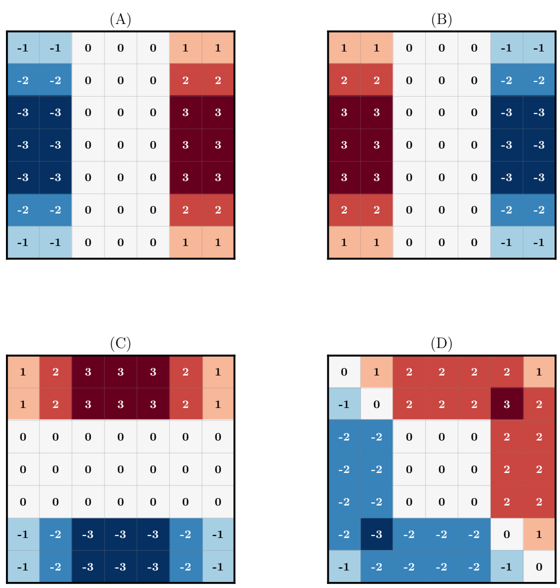
Problem 3.

The image below (where black pixels are 0 and white pixels are 1) is convolved with the following 3×3 filter (stride 1, no padding):

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$



Which of the following is the resulting response map?



- ◯ (A)
- ● (B)
- ◯ (C)
- ◯ (D)

Solution: (B).

The filter has negative values on the left and positive values on the right, making it a vertical edge detector that responds to transitions from dark (left) to light (right). It produces a positive response at the left edge of the white square (where the image transitions from black to white, left to right) and a negative response at the right edge (where the image transitions from white to black). This matches response map (B).

Problem 4.

Consider a convolutional neural network with the following architecture. The input is a $12 \times 12 \times 1$ grayscale image. It passes through Conv layer 1 (4 filters of size 3×3, stride 1, no padding), producing an output of shape $10 \times 10 \times 4$. Then 2×2 max pooling is applied, producing an output of shape $5 \times 5 \times 4$. Next is Conv layer 2 (6 filters of size $3 \times 3 \times 4$, stride 1, no padding), producing an output of shape $3 \times 3 \times 6$. This is flattened and fed into a fully connected layer with n nodes, followed by an output layer with 1 node.

a) What is the value of n?

- ○ 18
- ○ 36
- ● 54
- ○ 150

Solution: $n = 54$.

The output of Conv layer 2 is $3 \times 3 \times 6$. Flattening this gives $3 \times 3 \times 6 = 54$ values, so the fully connected layer has 54 nodes.

This was similar to Practice Problem 164.

b) What is the total number of learnable parameters in the network, excluding biases?

- ○ 144
- ○ 252
- ○ 258
- ● 306

Solution: 306.

Conv layer 1 has 4 filters of shape 3×3, each with 9 weights, for $4 \times 9 = 36$ parameters. Conv layer 2 has 6 filters of shape $3 \times 3 \times 4$, each with 36 weights, for $6 \times 36 = 216$ parameters. The fully connected layer connects to the output: $54 \times 1 = 54$ parameters. The grand total is $36 + 216 + 54 = 306$.

Note that max pooling has no learnable parameters.

This was similar to Practice Problem 164.

Problem 5.

A neural network with 3 output nodes uses the softmax activation function. The pre-activation values (logits) at the output layer are $\vec{z} = (1, 3, 1)$. What is the softmax output $\vec{h} = (h_1, h_2, h_3)$?

- ○ $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$
- ○ $\left(\frac{e}{2+e^3}, \frac{e^3}{2+e^3}, \frac{e}{2+e^3}\right)$
- ● $\left(\frac{e}{2e+e^3}, \frac{e^3}{2e+e^3}, \frac{e}{2e+e^3}\right)$
- ○ $\left(\frac{1}{2+e^3}, \frac{e^3}{2+e^3}, \frac{1}{2+e^3}\right)$

Solution: $\vec{h} = \left(\frac{e}{2e+e^3}, \frac{e^3}{2e+e^3}, \frac{e}{2e+e^3}\right)$.

By the softmax formula, $h_k = \frac{e^{z_k}}{\sum_{j=1}^{3} e^{z_j}}$. The denominator is:

$$e^1 + e^3 + e^1 = 2e + e^3$$

Therefore:

$$\begin{aligned} h_1 &= \frac{e^1}{2e+e^3} = \frac{e}{2e+e^3} \\ h_2 &= \frac{e^3}{2e+e^3} \\ h_3 &= \frac{e^1}{2e+e^3} = \frac{e}{2e+e^3} \end{aligned}$$

This was similar to Practice Problem 166.

Problem 6.

A multi-label classifier has 3 output nodes with sigmoid activations. The true labels are $\vec{y} = (0, 1, 1)$ and the predicted probabilities are $\vec{h} = (0.3, 0.7, 0.9)$.

What is the binary cross-entropy loss?

- ● $-\log(0.7) - \log(0.7) - \log(0.9)$
- ○ $-\log(0.3) - \log(0.7) - \log(0.9)$
- ○ $-\log(0.7) - \log(0.9)$
- ○ $-\log(0.3) - \log(0.3) - \log(0.1)$

Solution: $-\log(0.7) - \log(0.7) - \log(0.9)$.

By the binary cross-entropy formula:

$$\ell(\vec{h}, \vec{y}) = -\sum_{k=1}^{3} \begin{cases} \log h_k, & \text{if } y_k = 1 \\ \log(1 - h_k), & \text{if } y_k = 0 \end{cases}$$

Evaluating each term:

$$\begin{aligned} k = 1: &\quad y_1 = 0, \text{ so } -\log(1 - 0.3) = -\log(0.7) \\ k = 2: &\quad y_2 = 1, \text{ so } -\log(0.7) \\ k = 3: &\quad y_3 = 1, \text{ so } -\log(0.9) \end{aligned}$$

The total is $-\log(0.7) - \log(0.7) - \log(0.9) = -2\log(0.7) - \log(0.9)$.

Note that option (c), $-\log(0.7) - \log(0.9)$, is what you would get if you mistakenly computed the *categorical* cross-entropy loss (which only sums over k where $y_k = 1$) instead of the binary cross-entropy loss.

This was similar to Practice Problem 169.

Problem 7.

A multi-class classifier has 4 output nodes with softmax activation. The true label is $\vec{y} = (0, 1, 0, 0)$ and the softmax outputs are $\vec{h} = (0.15, 0.5, 0.25, 0.1)$.

What is the categorical cross-entropy loss?

- ○ $-\log(0.15) - \log(0.5) - \log(0.25) - \log(0.1)$
- ○ $-\log(0.85) - \log(0.5) - \log(0.75) - \log(0.9)$
- ● $-\log(0.5)$
- ○ $-\log(0.15) - \log(0.5)$

Solution: $-\log(0.5)$.

By the categorical cross-entropy formula:

$$\ell(\vec{h}, \vec{y}) = -\sum_{k=1}^{4} \begin{cases} \log h_k, & \text{if } y_k = 1 \\ 0, & \text{if } y_k = 0 \end{cases}$$

Only $y_2 = 1$ contributes, so the loss is $-\log(h_2) = -\log(0.5)$.

Note that option (b), $-\log(0.85) - \log(0.5) - \log(0.75) - \log(0.9)$, is what you would get if you mistakenly computed the *binary* cross-entropy loss instead of the categorical cross-entropy loss.

This was similar to Practice Problem 172.

Problem 8.

A neural network classifies images into 5 categories.

a) Suppose the categories are mutually exclusive (each image belongs to exactly one category). Which pair of output activation function and loss function would be most appropriate?

- ○ Sigmoid activation with categorical cross-entropy loss
- ● Softmax activation with categorical cross-entropy loss
- ○ Softmax activation with binary cross-entropy loss
- ○ Sigmoid activation with binary cross-entropy loss

Solution: Softmax activation with categorical cross-entropy loss.

Since the categories are mutually exclusive, we want the output probabilities to represent a single probability distribution over the 5 classes. Softmax enforces this by ensuring the outputs sum to 1. Categorical cross-entropy is the corresponding loss function: it penalizes the predicted probability of the true class.

b) Now suppose an image can belong to multiple categories simultaneously. Which pair of output activation function and loss function would be most appropriate?

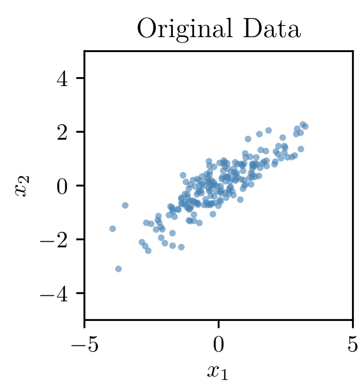
- ● Sigmoid activation with binary cross-entropy loss
- ○ Softmax activation with categorical cross-entropy loss
- ○ Sigmoid activation with categorical cross-entropy loss
- ○ Softmax activation with binary cross-entropy loss

Solution: Sigmoid activation with binary cross-entropy loss.

Since the categories are not mutually exclusive, each output node independently predicts the probability that the image belongs to that category. Sigmoid is applied independently to each output, and binary cross-entropy is used to evaluate each output independently.

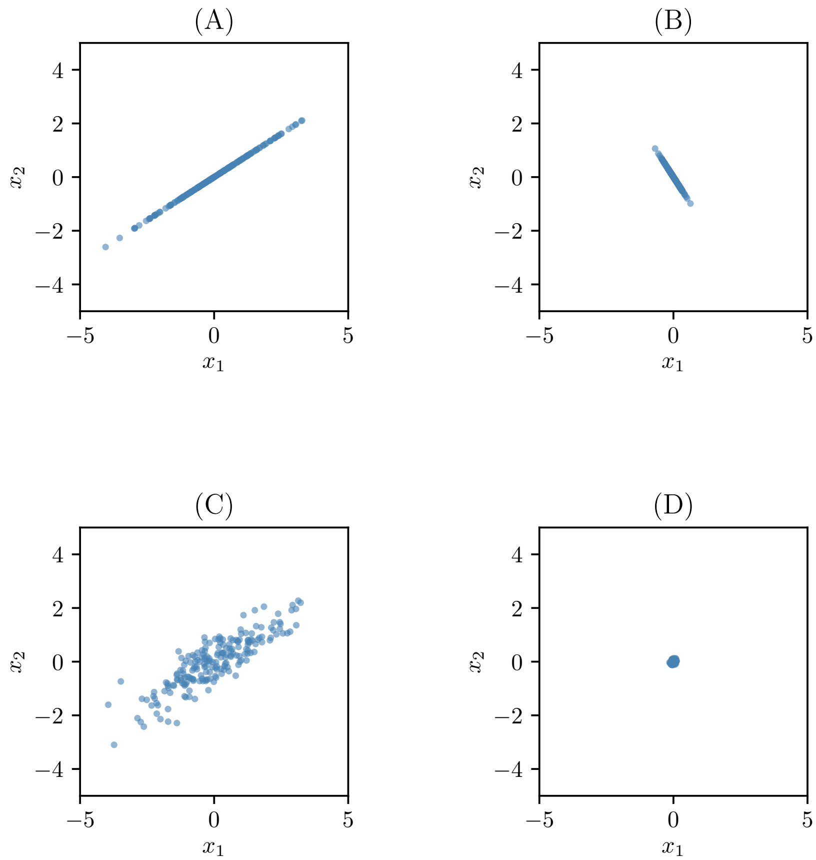
Problem 9.

The plot below shows a dataset of 200 points in $\mathbb{R}^2$.



An autoencoder with 2 input nodes, 1 hidden node, and 2 output nodes (all using linear activations) is trained on this dataset to minimize the mean squared reconstruction error, and the global minimum is achieved.

Which of the following plots shows the decoded data?



● (A) ○ (B) ○ (C) ○ (D)

Solution: (A).

With 1 hidden node and linear activations, the autoencoder learns to project the data onto a single direction and then reconstruct from that projection. To minimize reconstruction error, it will choose the direction of greatest variance—the first principal component. The reconstructed points therefore lie along a line in the direction of the first principal component, which is exactly what plot (A) shows.

Plot (B) is incorrect because it shows points along the direction of *least* variance (the second principal component). Plot (C) is incorrect because it shows the original data with no compression—but the bottleneck has only 1 node, so the autoencoder cannot perfectly reconstruct 2D data. Plot (D) is incorrect because it shows all points collapsed near the origin, which would have very high reconstruction error.

Problem 10.

a) Consider an autoencoder with 10 input nodes, a single hidden layer with 10 nodes, and 10 output nodes. All activation functions are linear (i.e., the identity function).

True or False: given any dataset, the smallest reconstruction error this autoencoder can possibly achieve is zero. You may assume that, when trained, the global minimum of the cost function is achieved.

● True

○ False

Solution: True.

Since the hidden layer has the same dimensionality as the input, the autoencoder can learn the identity function, mapping each input to itself exactly. This results in zero reconstruction error.

b) Now consider an autoencoder with 10 input nodes, 5 hidden nodes, and 10 output nodes. All activation functions are linear (i.e., the identity function), and the network has a single hidden layer.

True or False: there exists a dataset in $\mathbb{R}^{10}$ on which this autoencoder can achieve zero reconstruction error.

- ● True
- ○ False

Solution: True.

Even though the hidden layer has fewer nodes than the input, zero reconstruction error is possible if the data lies on a subspace of dimension at most 5. For example, if every data point in $\mathbb{R}^{10}$ has its last 5 components equal to zero, then the data lies in a 5-dimensional subspace. A linear autoencoder can learn to encode and decode within this subspace perfectly, achieving zero reconstruction error.

More generally, a linear autoencoder with k hidden nodes can perfectly reconstruct any dataset that lies in a k-dimensional (or lower) subspace of $\mathbb{R}^n$.