Exercise week 47-48

November 17-28, 2025

Date: **Deadline is Friday November 28 at midnight**

Overarching aims of the exercises this week

The exercise set this week is meant as a summary of many of the central elements in various machine learning algorithms we have discussed throught the semester. You don't need to answer all questions.

Linear and logistic regression methods

Question 1:

Which of the following is not an assumption of ordinary least squares linear regression?

- There is a linearity between predictors/features and target/outout
    - The inputs/features distributed according to a normal/gaussian distribution

Answer:

Not an assumption: The inputs/features are distributed according to a normal distribution

Question 2:

The mean squared error cost function for linear regression is convex in the parameters, guaranteeing a unique global minimum. True or False? Motivate your answer.

Answer:

True, because the MSE cost function is quadratic in the parameters and therefore convex, ensuring a unique global minimum when $\boldsymbol{X}^T\boldsymbol{X}$ is full rank.

Question 3:

Which statement about logistic regression is false?

- Logistic regression is used for binary classification.
    - It uses the sigmoid function to map linear scores to probabilities.

- It has an analytical closed-form solution.
- Its log-loss (cross-entropy) is convex.

Answer:

"It has an analytical closed-form solution" is false.

Question 4:

Logistic regression produces a linear decision boundary in the input space. True or False? Explain.

Answer:

True, because the sigmoid is applied to a linear function of the inputs, so the point where the model predicts class 1 (probability = 0.5) forms a straight line or plane.

Question 5:

Give two reasons why logistic regression is preferred over linear regression for binary classification.

Answer:

1) The outputs in logistic regression are between 0 and 1 (probabilites), while you can get impossible probabilites in linear regression (negative or larger than 1)

2) Logistic regression is designed for classification, using a proper loss function (cross-entropy), while linear regression assumes continuous targets.

Neural networks

Question 6:

Which statement is not true for fully-connected neural networks?

- Without nonlinear activation functions they reduce to a single linear model.
- Training relies on backpropagation using the chain rule.
- A single hidden layer can approximate any continuous function on a compact set.
- The loss surface of a deep neural network is convex.

Answer:

False: The loss surface of a deep neural network is convex.

Question 7:

Using sigmoid activations in many layers of a deep neural network can cause vanishing gradients. True or False? Explain.

Answer:

True, because the sigmoid squeezes values into a small range where its derivative is very small, so gradients shrink as they pass through many layers.

Question 8:

Describe the vanishing gradient problem: Why does it occur? Mention one technique to mitigate it and explain briefly.

Answer:

The vanishing gradient problem occurs when gradients become extremely small as they are backpropagated through many layers, especially with activations like sigmoid or tanh, causing early layers to learn very slowly. A common technique to mitigate it is using ReLU activations, which do not squash values into a small range and therefore keep gradients larger and easier to train.

Question 9:

Consider a fully-connected network with layer sizes n_0 (the input layer) ,n_1 (first hidden layer), $\ldots, n_L$, where n_L is the output layer. Derive a general formula for the total number of trainable parameters (weights + biases).

Answer:

The total number of trainable parameters is:

$$\sum_{l=1}^{L} (n_{l-1} \cdot n_l + n_l)$$

because each layer l has $n_{l-1} \cdot n_l$ weights and n_l biases.
Equivalently:

$$\sum_{l=1}^{L} n_l(n_{l-1} + 1).$$

Convolutional Neural Networks

Question 10:

Which of the following is not a typical property or advantage of CNNs?

- Local receptive fields
    - Weight sharing
    - More parameters than fully-connected layers
    - Pooling layers offering some translation invariance

Answer:

The statement “More parameters than fully-connected layers” is not a typical property of CNNs.

Question 11:

Using zero-padding in convolutional layers can preserve the input spatial dimensions when using a 3×3 kernel/filter, stride 1, and padding $P = 1$. True or False?a

Answer:

True, because padding by 1 on each side with a 3×3 filter and stride 1 keeps the output the same height and width as the input.

Question 12:

Given input width W, kernel size K, stride S, and padding P, derive the formula for the output width $W_{\text{out}} = \frac{W-K+2P}{S} + 1$.

Answer:

The output width is found by counting how many times the kernel fits when sliding across the padded input.

Padding expands the input width to:

$$W + 2P$$

Each kernel application uses (K) units of width, and the kernel moves by stride (S).
So the number of valid positions is:

$$W_{\text{out}} = \frac{(W + 2P) - K}{S} + 1$$

which simplifies to:

$$W_{\text{out}} = \frac{W - K + 2P}{S} + 1.$$

Question 13:

A convolutional layer has: C_{in} input channels, C_{out} output channels (filters) and kernel size $K_h \times K_w$. Compute the number of trainable parameters including biases.

Answer:

Each filter has:

$$C_{\text{in}} \cdot K_h \cdot K_w$$

weights, and each output channel has one bias term.
With C_{out} filters, the total number of parameters is:

$$C_{\text{out}} \left(C_{\text{in}} K_h K_w + 1 \right).$$

Recurrent Neural Networks

Question 14:

Which statement about simple RNNs is false?

- They maintain a hidden state updated each time step.
    - They use the same weight matrices at every time step.
    - They handle sequences of arbitrary length.
    - They eliminate the vanishing gradient problem.

Answer:

The false statement is: “They eliminate the vanishing gradient problem.”

Question 15:

LSTMs mitigate the vanishing gradient problem by using gating mechanisms (input, forget, output gates). True or False? Explain.

Answer:

True, because LSTMs use gates to control how information flows and is stored, allowing gradients to pass through many time steps without shrinking too much.

Question 16:

What is Backpropagation Through Time (BPTT) and why is it required for training RNNs?

Answer:

Backpropagation Through Time (BPTT) is the training method where an RNN is “unrolled” over all time steps, and gradients are computed across the entire sequence. It is required because RNN outputs depend on previous hidden states, so the model must backpropagate errors through time to update the shared weights correctly.

Question 17:

What does a sliding window do? And why would we use it?

Answer:

A sliding window extracts overlapping subsequences from a longer sequence or time series by moving a fixed-size window step by step. We use it to create training samples, capture local patterns, or make predictions based on recent history.