

Your grade: 95.96%

Your latest: 95.96% • Your highest: 95.96% • To pass you need at least 80%. We keep your highest score.

1. Which of the following is an advantage of using PyTorch for deep learning projects?

1 / 1 point

- ☒ Dynamic computation graph
- ☐ Built-in support for web development
- ☐ Statically-typed language features
- ☐ Limited support for neural network models

✔ Correct
Correct! PyTorch's dynamic computation graph allows for more flexibility and easier debugging.

2. What is a key benefit of using machine learning over classical programming approaches?

1 / 1 point

- ☐ Easier to implement
- ☐ Guaranteed to be error-free
- ☐ Does not require labeled data
- ☒ Ability to generalize from data

✔ Correct
Correct! Machine learning models can generalize from data, making them adaptable to new, unseen situations.

3. Which factor has significantly contributed to the recent popularity and improvement of deep learning?

1 / 1 point

- ☐ Introduction of microservices architecture
- ☐ Advancements in classical machine learning algorithms
- ☒ Increased computational power
- ☐ Development of new programming languages

✔ Correct
Correct! Increased computational power, especially with GPUs, has significantly contributed to the improvement and popularity of deep learning.

4. Identify the best practices for configuring the output layer in a neural network based on the type of problem.

0.6 / 1 point

- ☐ Use a large number of nodes with ReLU activation for regression problems
- ☐ Use a single node with Tanh activation for binary classification
- ☒ Use multiple nodes with Softmax activation for binary classification

✘ This should not be selected
Incorrect. Softmax activation is used for multi-class classification, not binary classification.

- ☐ Use Softmax activation for multi-class classification
- ☒ Use a single node with Sigmoid activation for binary classification

✔ Correct
Correct! A single node with Sigmoid activation is appropriate for binary classification problems.

5. What is the primary role of an optimizer in training a deep learning model?

1 / 1 point

- ☐ Increase the number of layers in the model
- ☐ Select the appropriate activation function
- ☐ Reduce overfitting by adding regularization
- ☒ Minimize the loss function

✔ Correct
Correct! The primary role of an optimizer is to minimize the loss function by adjusting the model parameters.

6. Which of the following activation functions is most likely to cause the vanishing gradient problem in deep neural networks?

1 / 1 point

- ☐ Tanh
- ☐ Softmax
- ☐ ReLU
- ☒ Sigmoid

☒ **Correct**
Correct! Sigmoid activation functions can cause the vanishing gradient problem because their gradients tend to be very small for large input values.

7. Select all the correct statements about the process of gradient descent and the impact of learning rate on the optimization process in deep learning models. 1 / 1 point

☒ Gradient descent updates model parameters in the direction of the gradient of the loss function with respect to the parameters.

☒ **Correct**
Correct! Gradient descent updates model parameters in the direction of the gradient of the loss function with respect to the parameters.

☐ The learning rate does not affect the convergence speed of the gradient descent algorithm.

☒ A very low learning rate can slow down the optimization process, potentially causing it to get stuck in local minima.

☒ **Correct**
Correct! A very low learning rate can indeed slow down the optimization process and cause it to get stuck in local minima.

☐ The optimal learning rate is the same for all types of neural network architectures.

☒ A very high learning rate can cause the optimization process to overshoot the minimum of the loss function.

☒ **Correct**
Correct! A very high learning rate can indeed cause the optimization process to overshoot the minimum.

☐ Gradient descent is only used in deep learning and not in other machine learning algorithms.

☒ Gradient descent aims to minimize the loss function by iteratively adjusting model parameters.

☒ **Correct**
Correct! Gradient descent minimizes the loss function by making iterative adjustments to the model parameters.

8. Which of the following are advantages of K-fold Cross-Validation? 1 / 1 point

☐ Eliminates the need for a separate test set.

☒ Provides a more reliable estimate of model performance.

☒ **Correct**
Correct! By averaging results from multiple folds, it gives a better estimate of model performance.

☒ Reduces the risk of overfitting by testing the model on multiple data sets.

☒ **Correct**
Correct! K-fold Cross-Validation helps in assessing how the model generalizes to an independent data set.

☐ Increases the complexity of the model by using more parameters.

9. Why is the derivative of the Sigmoid function significant during the neural network's forward pass? 1 / 1 point

☐ It initializes the network's parameters.

☐ It improves the computational efficiency of the forward pass.

☐ It reduces the dimensionality of the input data.

☒ It helps in updating the weights and biases during backpropagation.

☒ **Correct**
Correct! The derivative of the Sigmoid function is used in backpropagation to update weights and biases.

10. During the forward pass of a simple neural network, which mathematical operation is performed between the input features and the weights? 1 / 1 point

☐ Matrix inversion

☐ Cross product

☒ Dot product

☐ Element-wise multiplication

☒ **Correct**
Correct! The dot product operation is essential for computing the weighted sum of inputs in a neural network.

11. Which of the following tasks are part of setting up the backward pass in a neural network?

0.8 / 1 point

- ☐ Initializing parameters
- ☐ Preprocessing data
- ☒ Updating weights and biases

✔ **Correct**
Correct! Updating weights and biases based on the gradients is part of the backward pass.

- ☒ Calculating gradients

✔ **Correct**
Correct! Calculating gradients is a crucial step in setting up the backward pass.

- ☐ Calculating loss

You didn't select all the correct answers

12. Which of the following steps are involved in the training loop of a neural network?

1 / 1 point

- ☒ Forward pass

✔ **Correct**
Correct! The forward pass is a key step in the training loop to compute the network's predictions.

- ☒ Backward pass

✔ **Correct**
Correct! The backward pass is performed to calculate gradients and update the model's parameters.

- ☐ Hyperparameter tuning

- ☒ Updating weights

✔ **Correct**
Correct! Updating the weights is part of the training loop, based on calculated gradients.

- ☐ Preprocessing data

13. In the context of training a neural network from scratch, which of the following is a key step in preprocessing the training data for heart attack detection?

1 / 1 point

- ☐ Initializing weights to zero
- ☐ Randomly shuffling the labels
- ☒ Scaling the features to a standard range
- ☐ Removing outliers by deleting rows with missing values

✔ **Correct**
Correct! Scaling features helps improve the convergence rate of the training algorithm.

14. Which method would you use to create a tensor filled with zeros in PyTorch?

1 / 1 point

- ☒ torch.zeros()
- ☐ torch.fill_zeros()
- ☐ torch.empty()
- ☐ torch.zeros_like()

✔ **Correct**
Correct! The torch.zeros() method fills a tensor with zeros.

15. When setting up a neural network model from scratch in PyTorch, which of the following components is essential to define the architecture of the model?

1 / 1 point

- ☐ The torch.nn.functional module
- ☐ The torch.optim module
- ☐ The DataLoader class
- ☒ The nn.Module class

✔ **Correct**
Correct! The nn.Module class is fundamental in PyTorch to define the architecture of your neural network model.

16. In the context of linear regression using PyTorch, what is the primary purpose of the Model class?

1 / 1 point

- ☐ To handle data preprocessing steps like normalization and scaling.
- ☐ To save and load the model's parameters.
- ☒ To define the structure of the linear regression model and its forward pass.

☐ To implement the training loop and loss calculation.

☒ **Correct**
Correct! The primary purpose of the Model class is to define the structure of the linear regression model and its forward pass.

17. Which factors should be considered when experimenting with different hyperparameter values in PyTorch?

1 / 1 point

☒ Batch size

☒ **Correct**
Correct! Batch size is also an important hyperparameter to consider.

☐ Model architecture

☒ Learning rate

☒ **Correct**
Correct! Learning rate is a critical hyperparameter that impacts model training.

☐ Data normalization technique

☒ Number of layers in the model

☒ **Correct**
Correct! The number of layers in the model can significantly affect performance.

☒ Type of activation function

☒ **Correct**
Correct! The type of activation function used can impact the model's performance.

18. Which of the following are steps involved in the training loop of a PyTorch model?

0.8333333333333334
/ 1 point

☒ Update the model's parameters using the optimizer.

☒ **Correct**
Correct! Updating the model's parameters using the optimizer is essential in the training loop.

☒ Compute the loss based on the model's predictions and the true labels.

☒ **Correct**
Correct! Computing the loss is a fundamental step in the training loop.

☐ Initialize the model's parameters.

☐ Set the model to evaluation mode.

☐ Save the model's state dictionary.

☒ Perform backpropagation to compute gradients.

☒ **Correct**
Correct! Backpropagation is a key step to compute gradients in the training loop.

You didn't select all the correct answers

19. In PyTorch, which method is used to manually modify the parameters of a neural network model?

1 / 1 point

☐ model.state_dict()

☐ model.parameters().grad

☒ model.parameters()

☐ model.load_state_dict()

☒ **Correct**
Correct! The model.parameters() method can be used to access and manually modify the parameters of a neural network model.