

Your grade: 100%

[Next item →](#)

Your latest: 100% • Your highest: 100% • To pass you need at least 80%. We keep your highest score.

1. What does the loss function measure during training?

1 / 1 point

- The error between predictions and actual values.
- The rate at which the model is learning
- The accuracy of the model's predictions.
- How confident the model is in its predictions

Correct

Yes. The loss function quantifies prediction error, showing how far off the model's predictions are from the actual values.

2. Why do neural networks need activation functions like ReLU between layers?

1 / 1 point

- To automatically adjust the learning rate during training.
- To reduce the amount of data needed for training.
- To speed up the training process.
- To enable the network to learn non-linear patterns.

Correct

Yes. Without activation functions, stacking layers just produces another linear equation—still a straight line. Activation functions add the non-linearity needed to model curves.

3. If your training process runs for 1000 epochs, what does one epoch represent?

1 / 1 point

- One update to all the model's weights
- One forward and backward pass through the network
- One complete pass through the entire training dataset.
- One pass through a single batch of training data.

Correct

Yes. An epoch is one full cycle through all the training data during learning.

4. In PyTorch, what does `loss.backward()` accomplish?

1 / 1 point

- Makes predictions using the neural network.
- Saves the trained model parameters to disk.
- Cleans and prepares data before training.
- Calculates gradients to reduce the loss.

Correct

Yes. `loss.backward()` calculates how to adjust the model's parameters to reduce the loss—this happens using calculus behind the scenes.

5. What does the `dtype` parameter control when creating a PyTorch tensor?

1 / 1 point

- The device where the tensor is stored (CPU or GPU).
- The type of numbers stored in the tensor.
- How many samples are in the batch.
- The number of dimensions in the tensor.

Correct

Yes. The `dtype` parameter specifies what kind of numbers the tensor contains—integers, floats, and their precision.

6. What will be printed by the following code?

```
1 x = torch.tensor([2.0, 4.0, 6.0])
```

1 / 1 point