Your grade: 100%

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

Next item →

1. What are the names of the two functions which are part of the built in solution for training a model in TensorFlow and Keras? 1 / 1 point

- ○ model.build() and model.train()
- ◉ model.compile() and model.fit()
- ○ model.structure() and model.compute()
- ○ model.configure() and model.map()

Correct
Correct!

2. With regards to the loss function, the derivative that gives you the gradient points away from the minimum loss value, but you can still use it for direction by adding the gradient value to your weight, instead of subtracting from it. 1 / 1 point

- ◉ False
- ○ True

Correct
Correct! You subtract the gradient value from your weights, instead of adding it.

3. During training one of the parameters which you have to set is the *Learning Rate*.Which of the following are true statement(s) for *Learning Rate*? Check all that are true. 1 / 1 point

- ☐ A larger learning rate (0.1 as opposed to 0.0001) leads to a bigger update and reaches the minimum loss with fewer iterations, so a larger learning rate is better.
- ☐ It defines the direction in which we should jump while finding the minimum loss value
- ☑ With a smaller learning rate (0.0001 as opposed to 0.1), the update to the weight is smaller, and will take more iterations to minimize the loss.

Correct
Correct!

- ☑ It defines how big a step to take in the given direction during the update step.

Correct
Correct!

4. For creating your own custom training loop, arrange the following steps in the correct order for each training batch: 1 / 1 point

1. Accumulate accuracy metric
2. Calculate loss
3. Calculate gradients of loss with respect to the model trainable weights
4. Calculate logits
5. Apply gradients on model using optimizer

- ○ 2, 4, 5, 3, 1
- ◉ 4, 2, 3, 5, 1
- ○ 4, 1, 5, 2, 3
- ○ 2, 1, 3, 4, 5

Correct
Correct!

5. Consider the code below used in custom *batch training*: 1 / 1 point

```
def apply_gradient(optimizer, model, x, y):
  with tf.GradientTape() as tape:
    loss_value, logits = # some code for setting up these values

  gradients = tape.gradient(loss_value, model.trainable_weights)
  # code for optimizing

  return loss_value, logits
```

Which of the following lines of code performs optimization (should appear where you see the comment "# code for optimizing" ?

- ○ optimizer = apply_gradients(zip(gradients, model.trainable_weights))
- ○ optimizer = apply_gradients(gradients, model.trainable_weights)
- ○ optimizer.apply_gradients(gradients, model.trainable_weights)
- ◉ optimizer.apply_gradients(zip(gradients, model.trainable_weights))

Correct
Correct!

6. Metrics in Keras can only be called as functions and not instantiated as Classes. **1 / 1 point**

- ◉ False
- ○ True

Correct
Correct! They can be instantiated as *Class* as well.

7. Which of the following is the correct syntax for querying the current value of metric for display ? **1 / 1 point**

- ○ metric.get_result
- ◉ metric.result
- ○ metric.state
- ○ metric.get_state

Correct
Correct!