

✓ **Congratulations! You passed!**

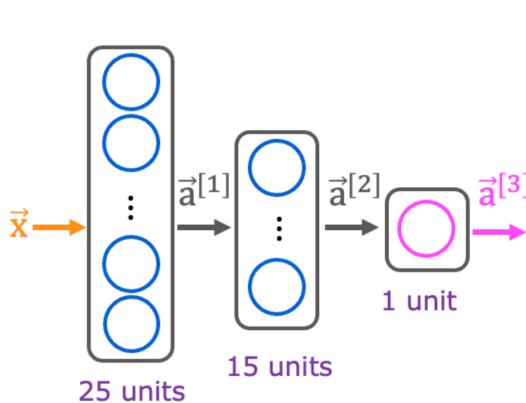
Grade received 100% Latest Submission Grade 100% To pass 80% or higher

Go to next item

1.

1 / 1 point

Train a Neural Network in TensorFlow



```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([
    Dense(units=25, activation='sigmoid')
    Dense(units=15, activation='sigmoid')
    Dense(units=1, activation='sigmoid')
])

from tensorflow.keras.losses import
BinaryCrossentropy
```

```
model.fit(X,Y,epochs=100)
```

Here is some code that you saw in the lecture:

```
...

model.compile(loss=BinaryCrossentropy())
...
```

For which type of task would you use the binary cross entropy loss function?

- ☒ binary classification (classification with exactly 2 classes)
- ☐ A classification task that has 3 or more classes (categories)
- ☐ BinaryCrossentropy() should not be used for any task.
- ☐ regression tasks (tasks that predict a number)

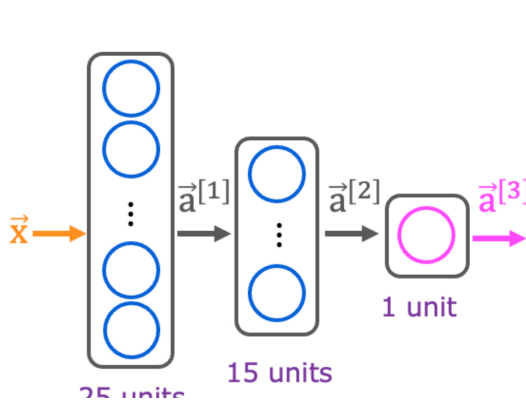
✓ **Correct**

Yes! Binary cross entropy, which we've also referred to as logistic loss, is used for classifying between two classes (two categories).

2.

1 / 1 point

Train a Neural Network in TensorFlow



```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([
    Dense(units=25, activation='sigmoid')
    Dense(units=15, activation='sigmoid')
    Dense(units=1, activation='sigmoid')
])

from tensorflow.keras.losses import
BinaryCrossentropy
```

```
model.fit(X,Y,epochs=100)
```

Here is code that you saw in the lecture:

```
...  
  
model = Sequential([  
  
Dense(units=25, activation='sigmoid'),  
  
Dense(units=15, activation='sigmoid'),  
  
Dense(units=1, activation='sigmoid')  
  
])  
  
  
model.compile(loss=BinaryCrossentropy())  
  
model.fit(X,y,epochs=100)  
  
...
```

Which line of code updates the network parameters in order to reduce the cost?

- ☒ model.fit(X,y,epochs=100)
- ☐ None of the above -- this code does not update the network parameters.
- ☐ model.compile(loss=BinaryCrossentropy())
- ☐ model = Sequential([...])

✓ **Correct**

Yes! The third step of model training is to train the model on data in order to minimize the loss (and the cost)