

Model Output

Complete a multilayer perceptron model in Keras.

Chapter Goals:

- Add the final layers to the MLP for multiclass classification

A. Final layer activation

In the **Intro to Deep Learning** section, we built the MLP classification models such that each model produced logits (<https://en.wikipedia.org/wiki/Logit>). This is because the TensorFlow cross-entropy (https://en.wikipedia.org/wiki/Cross_entropy) loss functions applied the sigmoid/softmax function to the output of the MLP.

In Keras, the cross-entropy loss functions only calculate cross-entropy, without applying the sigmoid/softmax function to the MLP output. Therefore, we can have the model directly output class probabilities instead of logits (i.e. we apply sigmoid/softmax activation to the output layer).

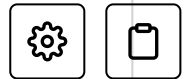
```
1 model = Sequential()
2 layer1 = Dense(5, activation='relu', input_dim=4)
3 model.add(layer1)
4 layer2 = Dense(1, activation='sigmoid')
5 model.add(layer2)
```




Creating an MLP model for binary classification (sigmoid activation).

```
1 model = Sequential()
2 layer1 = Dense(5, input_dim=4)
```

```
3 model.add(layer1)
4 layer2 = Dense(3, activation='softmax')
5 model.add(layer2)
```



 Code Execution In Progress...

Creating an MLP model for multiclass classification with 3 classes (softmax activation).

Time to code!

The coding exercise will complete the Keras `Sequential` model that was set up in the previous chapter. Note that the output size of the model will be 3 (there are 3 possible classes for each data observation).

Set `layer2` equal to a `Dense` with 5 as the required argument and `'relu'` for the activation keyword argument. Then call `model.add` on `layer2`.

Set `layer3` equal to a `Dense` with 3 as the required argument and `'softmax'` for the activation keyword argument. Then call `model.add` on `layer3`.

```
1 model = Sequential()
2 layer1 = Dense(5, activation='relu', input_dim=2)
3 model.add(layer1)
4 # CODE HERE
```



Solution



```
1 model = Sequential()
2 layer1 = Dense(5, activation='relu', input_dim=2)
3 model.add(layer1)
4 layer2 = Dense(5, activation='relu')
5 model.add(layer2)
6 layer3 = Dense(3, activation='softmax')
7 model.add(layer3)
```

[← Back](#)

Sequential Model

[Next →](#)

Model Configuration

**Mark as Completed**

94% completed, meet the [criteria](#) and claim your course certificate!

[Buy Certificate](#)[Report an Issue](#)[Ask a Question](#)

(https://discuss.educative.io/tag/model-output__deep-learning-with-keras__machine-learning-for-software-engineers)