Congratulations! You passed!
TO PASS 80% or higher

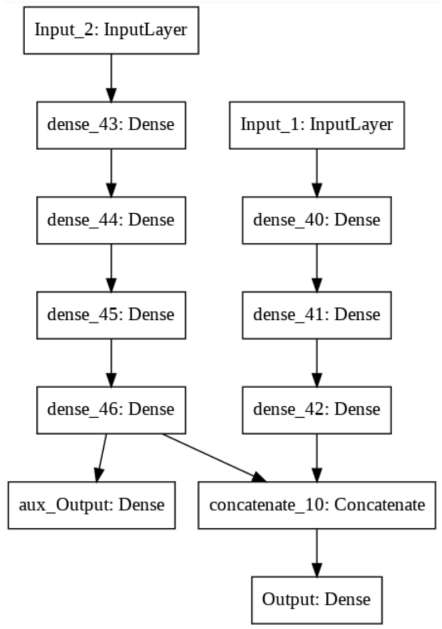
Keep Learning

GRADE
100%

Custom Models

LATEST SUBMISSION GRADE
100%

1. Following is an example of a deep and wide network structure. **1 / 1 point**



- ○ True
- ◉ False

✓ **Correct**
Correct! This model structure does not have an input path that go through a shallow, or a wide layer.

2. Consider the following code and check all that are true: **1 / 1 point**

```
class MyModel(Model):
  def __init__(self, units=30, activation='relu', **kwargs):
    super().__init__(**kwargs)
    self.hidden1 = Dense(units, activation=activation)
    self.hidden2 = Dense(units, activation=activation)
    self.main_output = Dense(1)
    self.aux_output = Dense(1)

  def call(self, inputs):
    input_A, input_B = inputs
    hidden1 = self.hidden1(input_B)
    hidden2 = self.hidden2(hidden1)
    concat = concatenate([input_A, hidden2])
    main_output = self.main_output(concat)
    aux_output = self.aux_output(hidden2)
    return main_output, aux_output
```

- ☑ The *init* function initializes the *MyModel* Class objects, as well as the attributes that are inherited from the *Model* Class.

✓ **Correct**
Correct!

- ☐ The code is incomplete in the sense that you can only initialize and construct your model, you cannot perform training or inference.
- ☑ The output layers cannot give more than 1 result each.

✓ **Correct**
Correct! They each hold only 1 unit.

- [ ] The *concat* should be defined within the *init* function instead of the *call* function as it is also a hidden layer.

3. You have learned that Sequential and Functional APIs have their limitations. **1 / 1 point**

   How can you build dynamic networks where the architecture changes on the fly, or networks where recursion is used? Check all that are true:

   - [x] Using Functional API

     ✓ **Correct**
     Correct! With Functional APIs it is possible to build these networks, but it would require a lot of coding.

   - [x] Using model subclassing

     ✓ **Correct**
     Correct! With model subclassing it is relatively easier to build these complex networks.

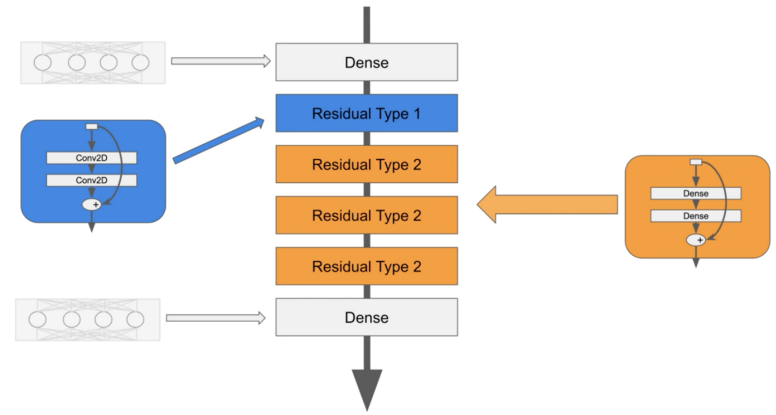
   - [ ] Using Sequential API

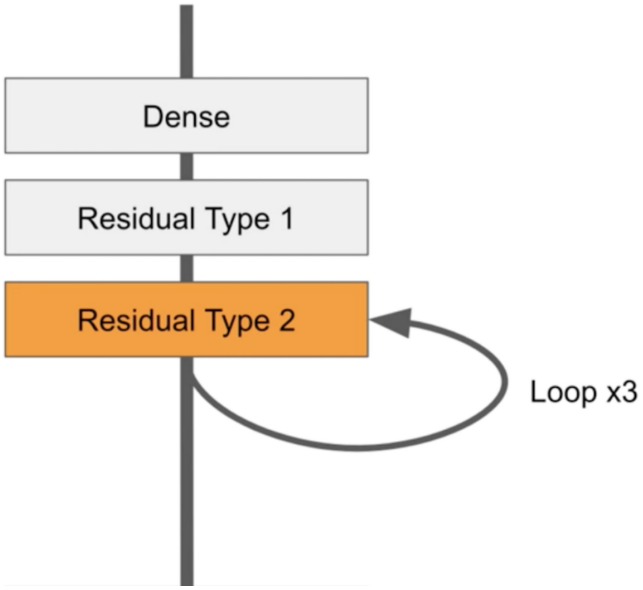
4. Which one of the following is a false statement regarding model subclassing? **1 / 1 point**

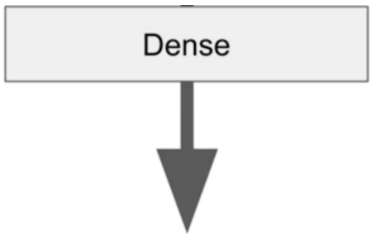
   - ( ) Instead of tweaking the entire architecture, you can have different modules and make changes in them as required, as opposed to entirely rewriting the structure.
   - ( ) You can make use of Functional and Sequential APIs when writing code for model subclassing.
   - ( ) You can have modular architectures
   - (•) You cannot introduce a branch structure in the architecture when doing model subclassing.

     ✓ **Correct**
     Correct! You can have branches within your network

5. Consider the following two images: **1 / 1 point**







Check all that are true:

- [x] When you make a loop of *Residual Type 2* blocks, each block could have the same weights.

Correct
Correct!

- [x] Each Residual block has two hidden layers and one add layer in it.

Correct
Correct!

- [x] You make a loop of *Residual Type 2* blocks because you want to reduce the depth of the network (making it less complex of an architecture)

Correct
Correct!

- [ ] You loop *Residual Type 2* (*Dense* layers) because you cannot make a loop of *Conv2D* layers (*Residual Type 1)*