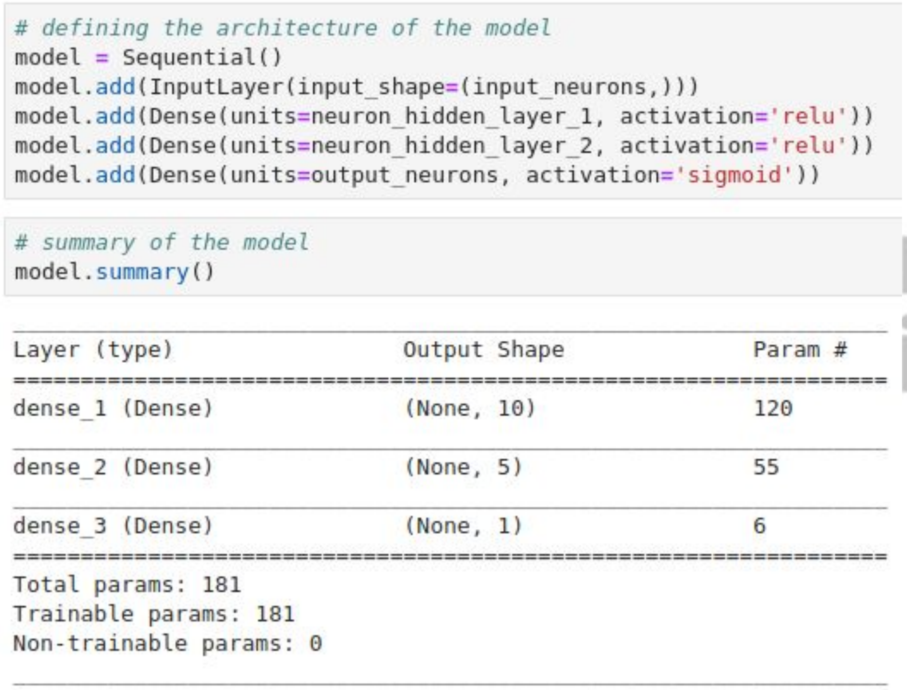
Functional API for Deep Learning

Sequential API

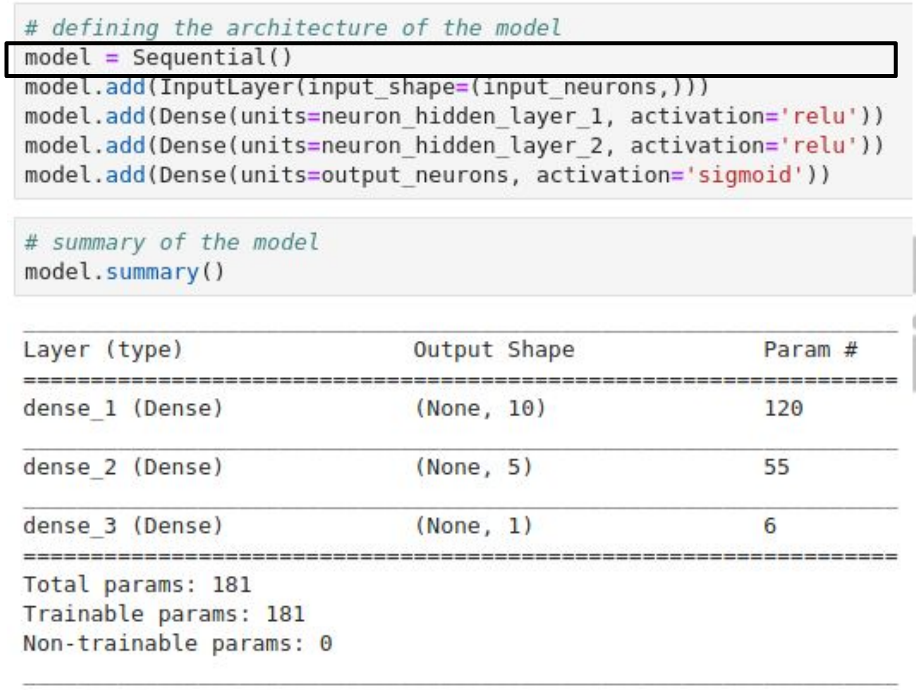


```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 10)                120       
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55        
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6         
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```

Sequential API

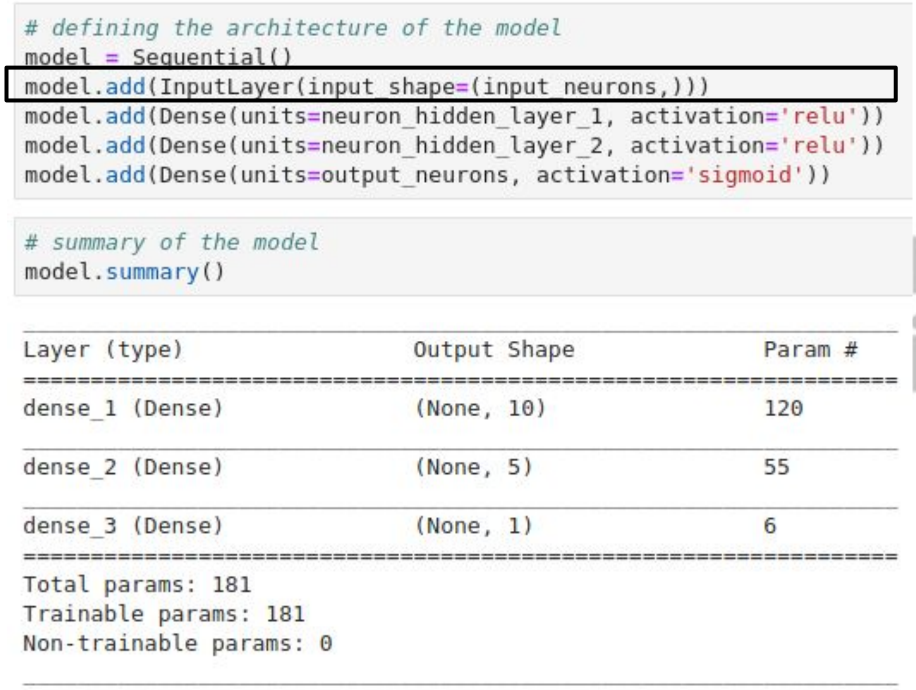


```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                120
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```

Sequential API



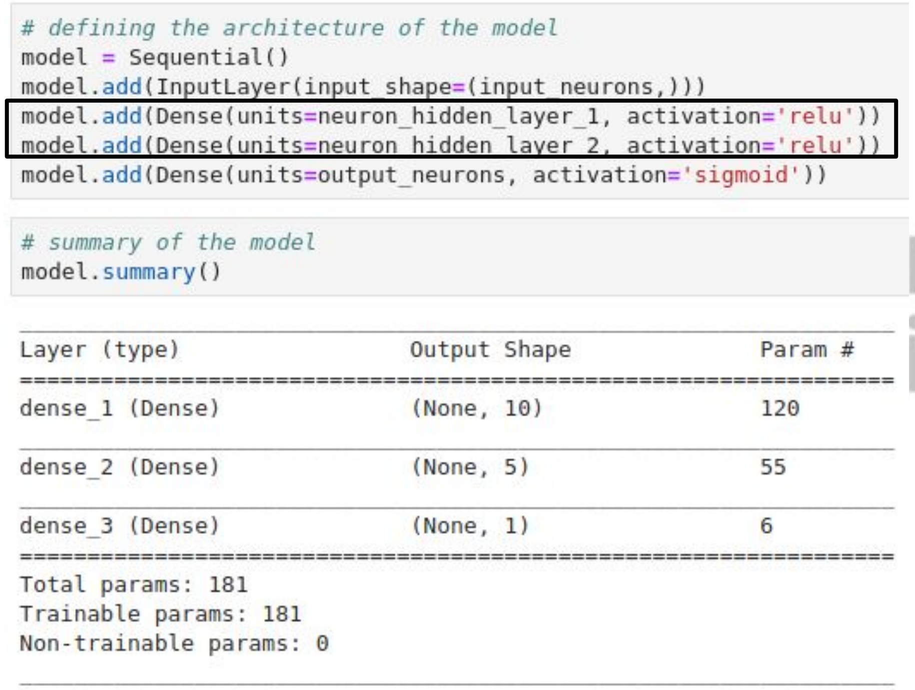
```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                120
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```

Input Layer

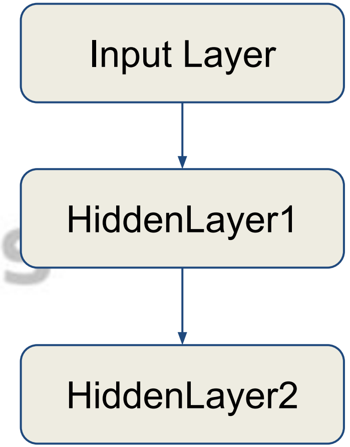
Sequential API



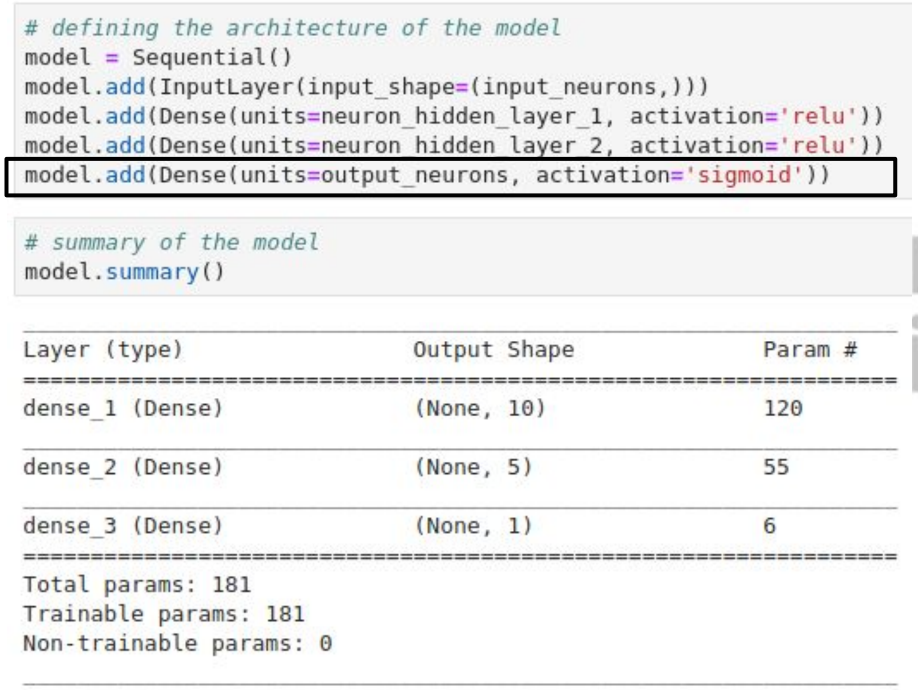
```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 10)                120       
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55        
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6         
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



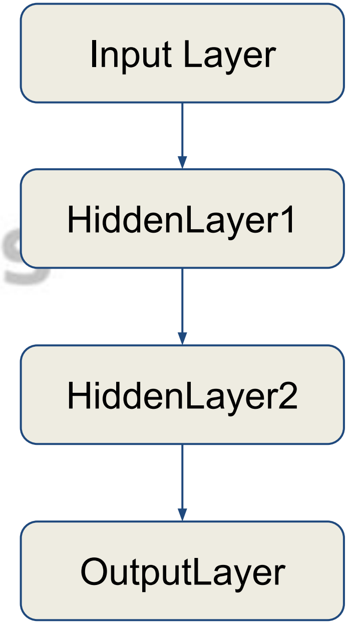
Sequential API



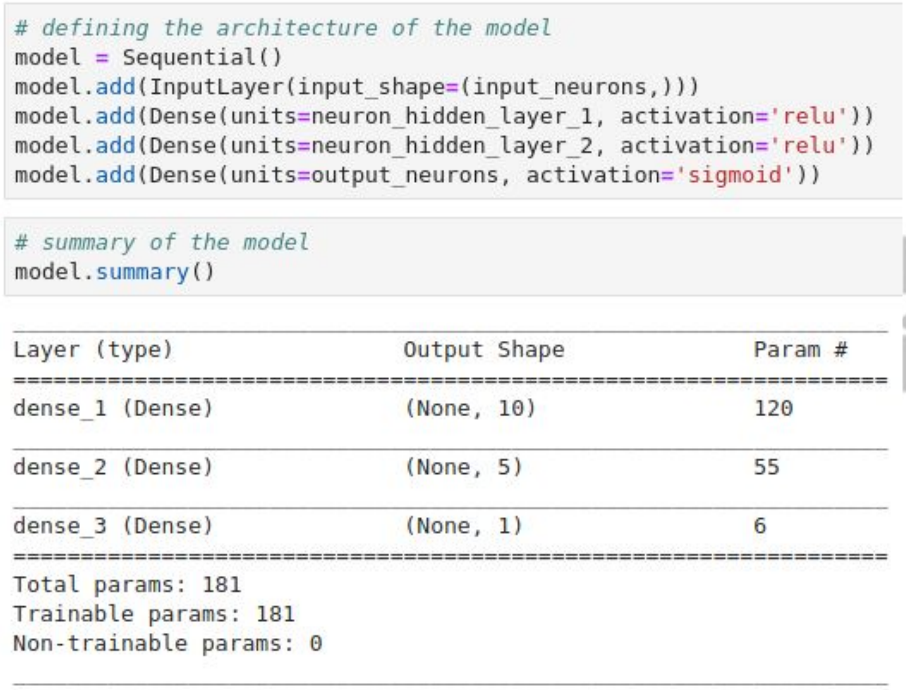
```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                120
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



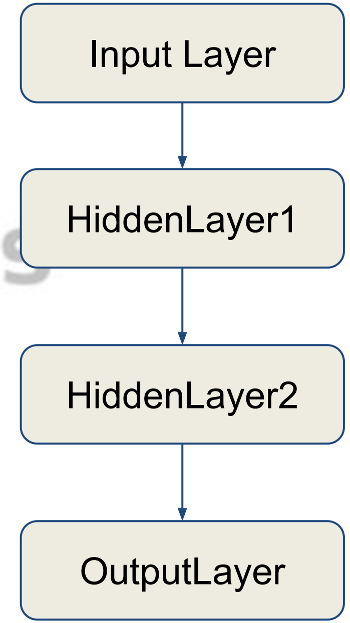
Sequential API



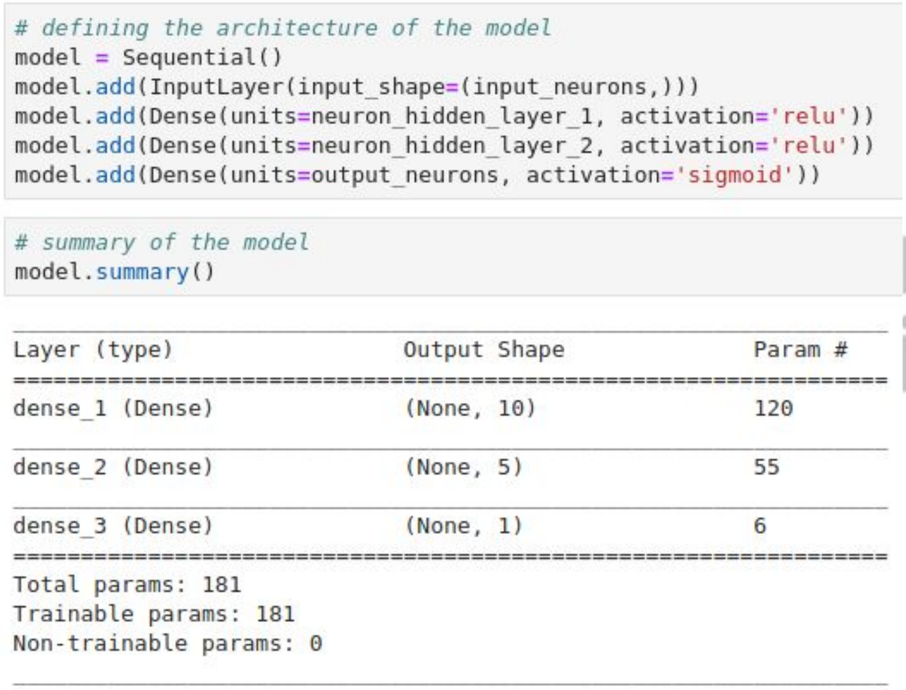
```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                120
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



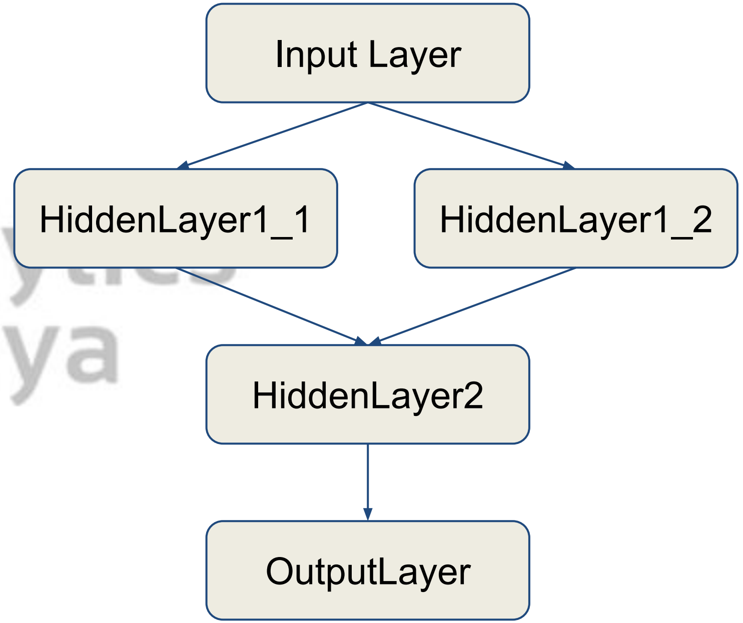
Sequential API



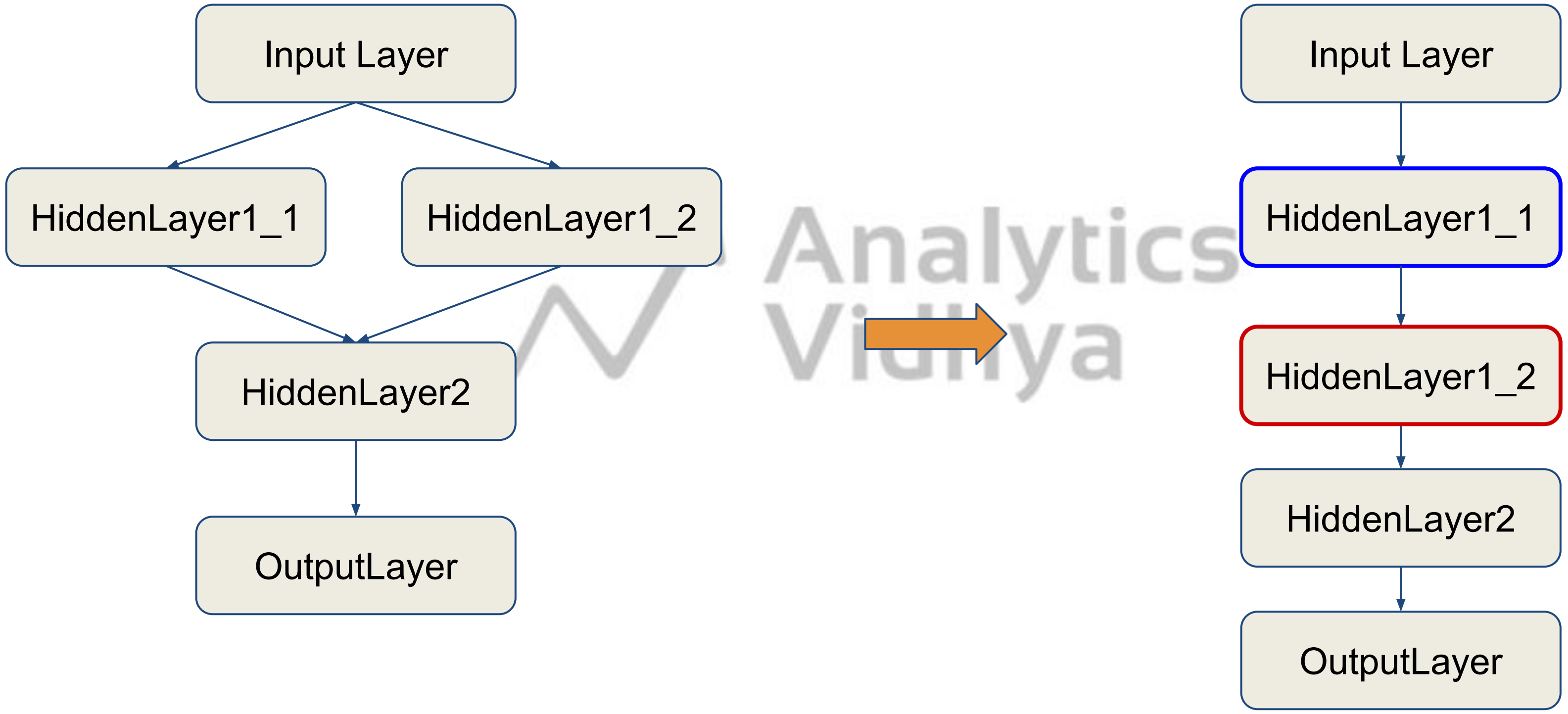
```
# defining the architecture of the model
model = Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=neuron_hidden_layer_1, activation='relu'))
model.add(Dense(units=neuron_hidden_layer_2, activation='relu'))
model.add(Dense(units=output_neurons, activation='sigmoid'))
```

```
# summary of the model
model.summary()
```

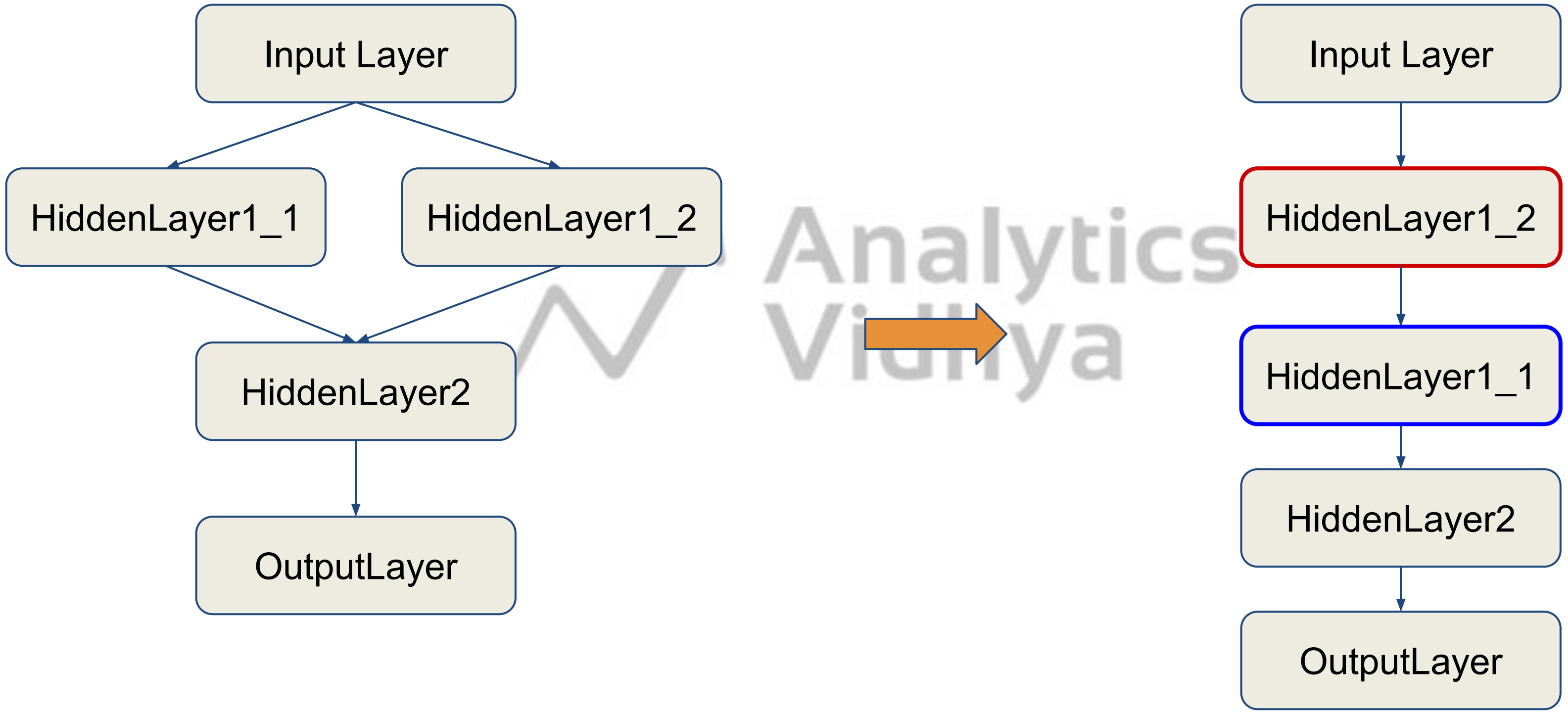
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                120
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



Sequential API



Sequential API



Functional API

1. Provides more flexibility to define models

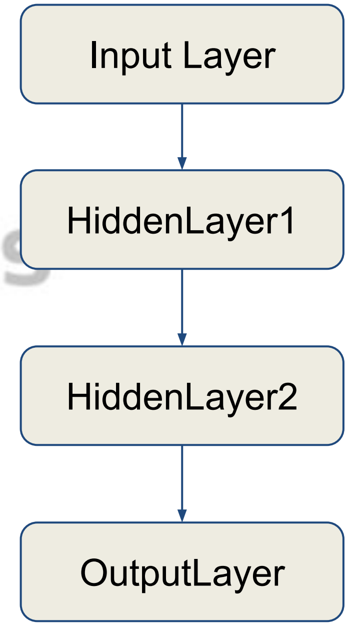
2. Can define multiple inputs, outputs models

3. Split and share the intermediate layers

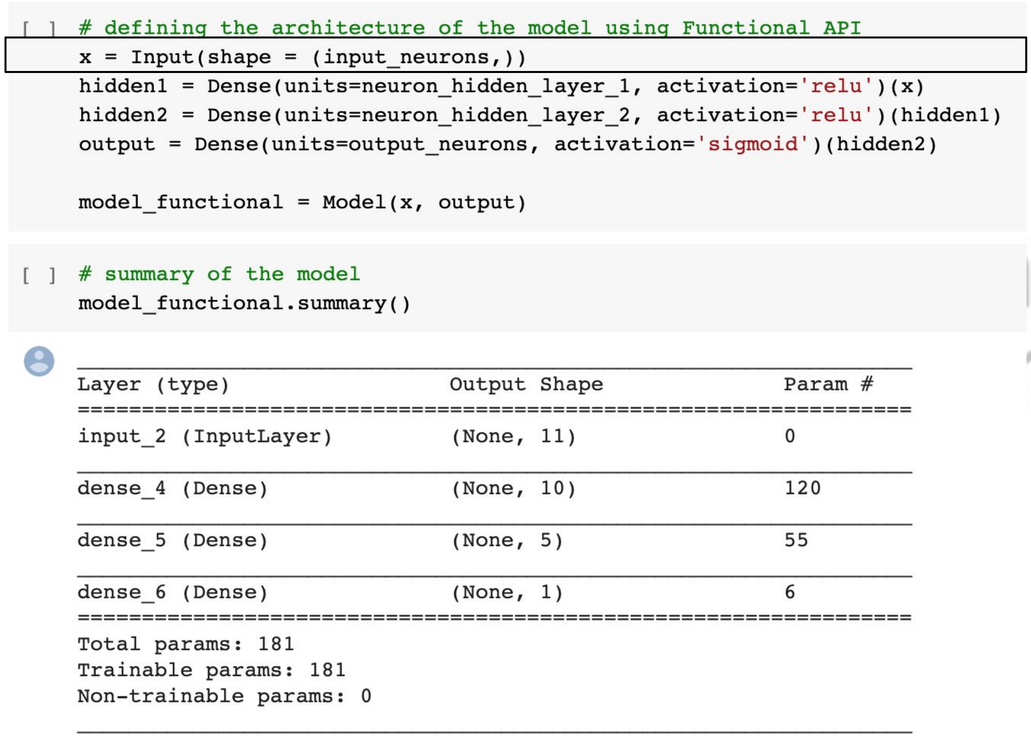
4. Build State-of-the-Art model architectures or Custom architecture

Build Simple NN model using Functional API

Create Layers and Connect



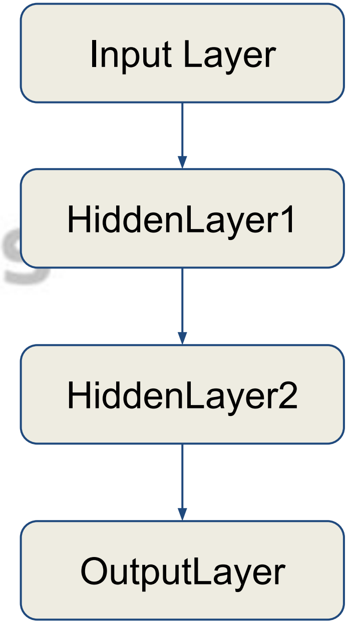
Build Simple NN model using Functional API



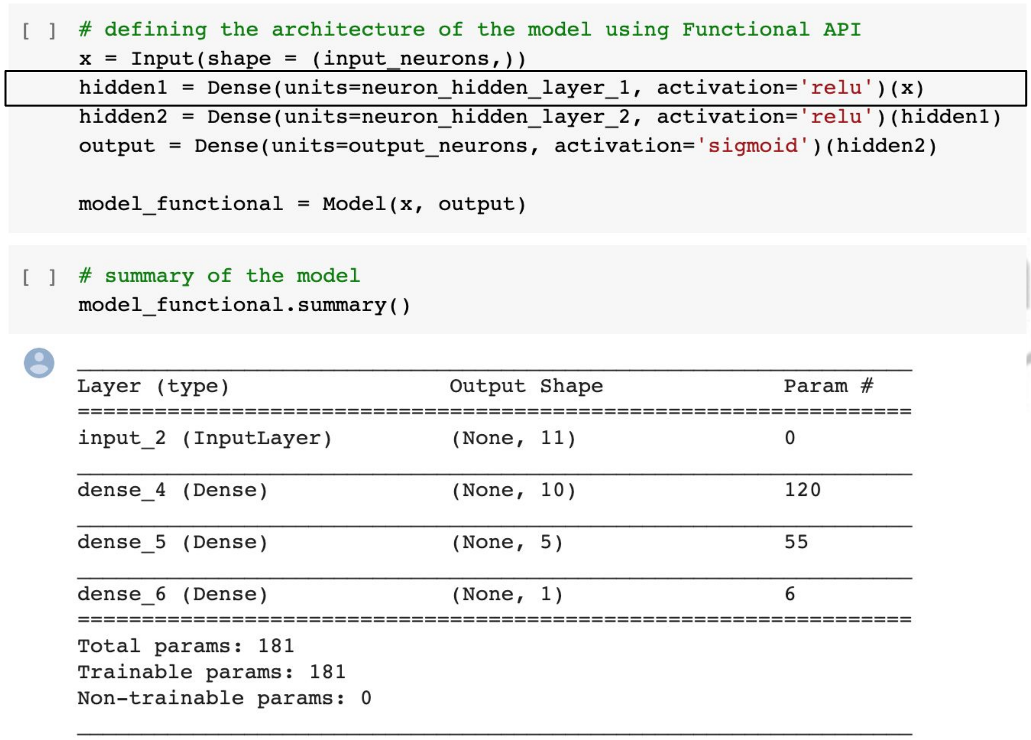
```
[ ] # defining the architecture of the model using Functional API
    x = Input(shape = (input_neurons,))
    hidden1 = Dense(units=neuron_hidden_layer_1, activation='relu')(x)
    hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(hidden1)
    output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

    model_functional = Model(x, output)
```

```
[ ] # summary of the model
    model_functional.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 11)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                120
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



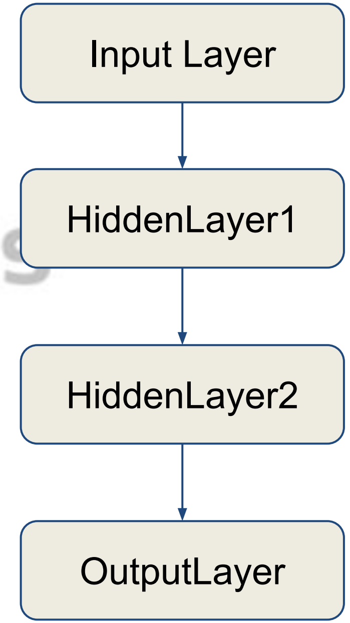
Build Simple NN model using Functional API



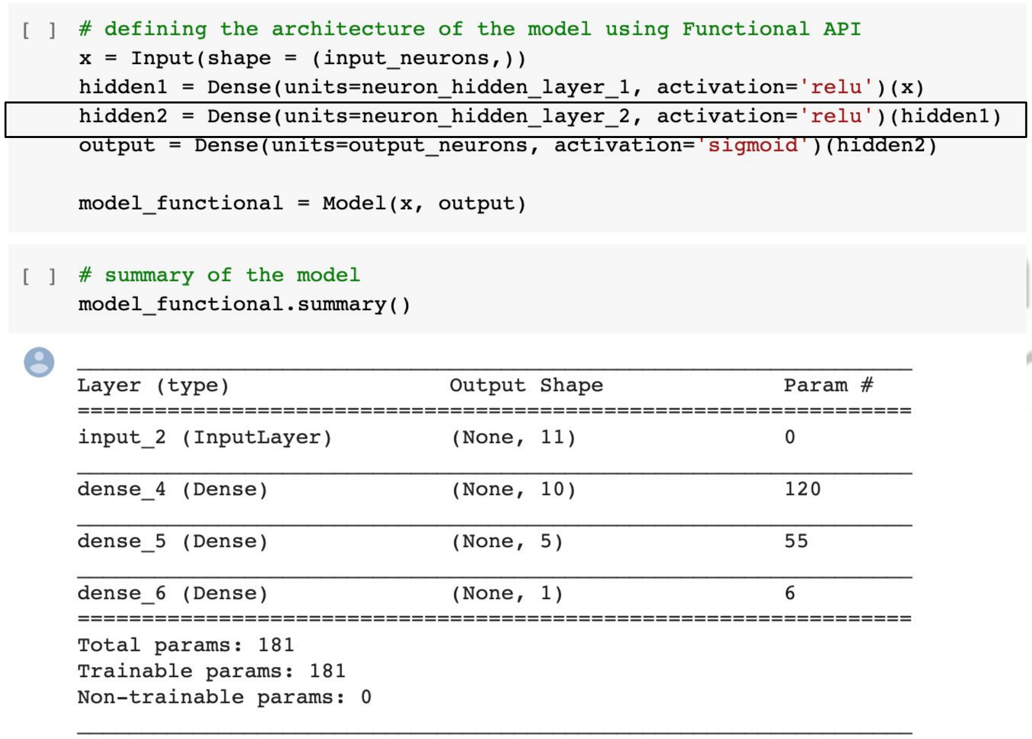
```
[ ] # defining the architecture of the model using Functional API
    x = Input(shape = (input_neurons,))
    hidden1 = Dense(units=neuron_hidden_layer_1, activation='relu')(x)
    hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(hidden1)
    output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

    model_functional = Model(x, output)
```

```
[ ] # summary of the model
    model_functional.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 11)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                120
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



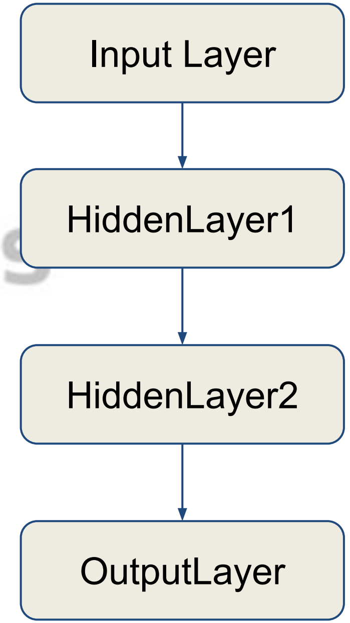
Build Simple NN model using Functional API



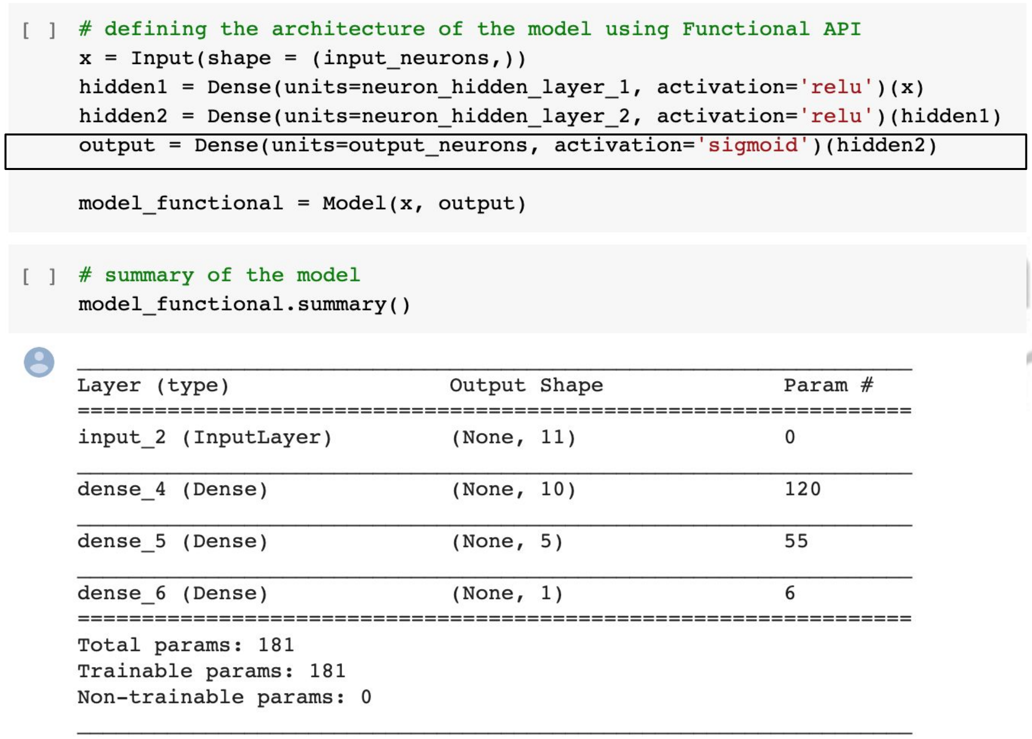
```
[ ] # defining the architecture of the model using Functional API
    x = Input(shape = (input_neurons,))
    hidden1 = Dense(units=neuron_hidden_layer_1, activation='relu')(x)
    hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(hidden1)
    output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

    model_functional = Model(x, output)
```

```
[ ] # summary of the model
    model_functional.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 11)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                120
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



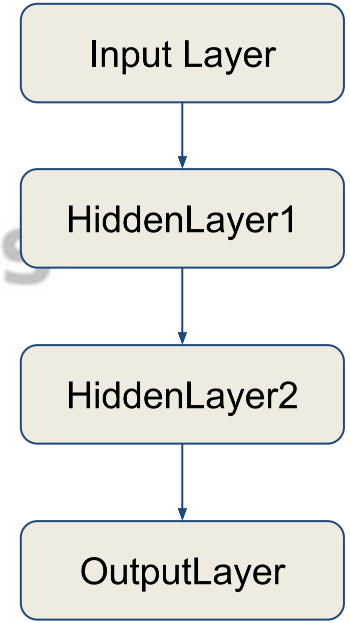
Build Simple NN model using Functional API



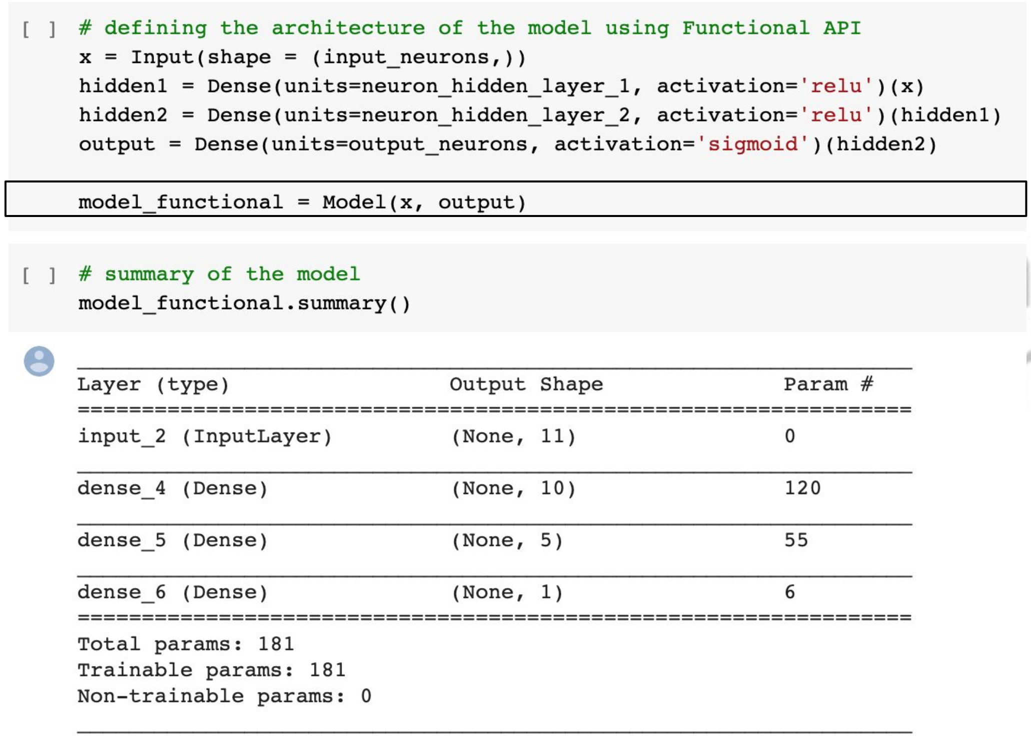
```
[ ]  # defining the architecture of the model using Functional API
     x = Input(shape = (input_neurons,))
     hidden1 = Dense(units=neuron_hidden_layer_1, activation='relu')(x)
     hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(hidden1)
     output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

     model_functional = Model(x, output)
```

```
[ ]  # summary of the model
     model_functional.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 11)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                120
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



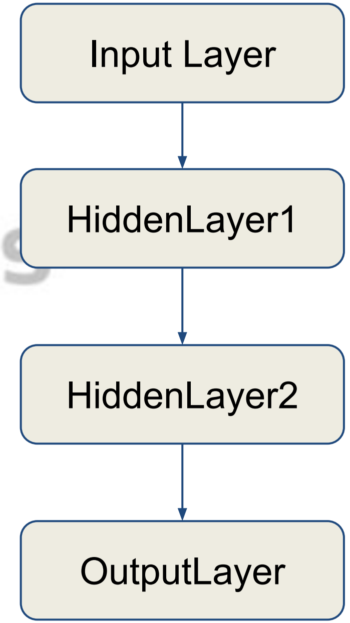
Build Simple NN model using Functional API



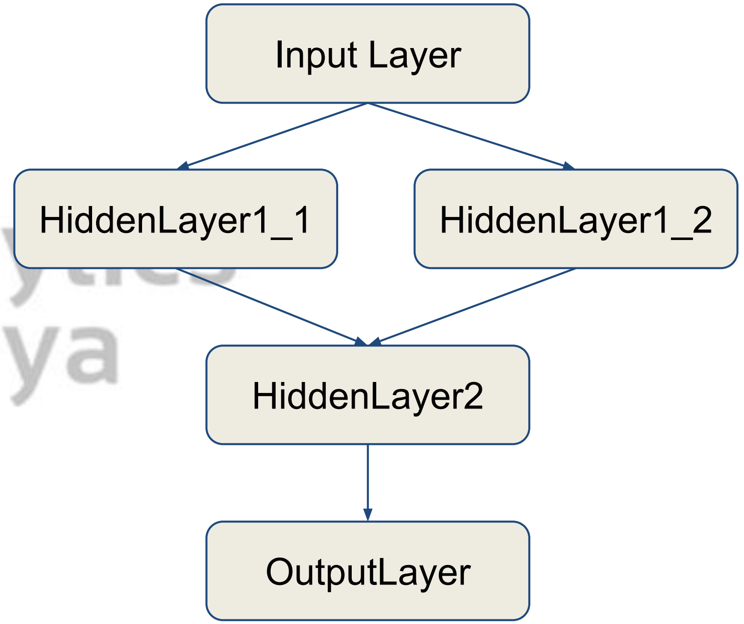
```
[ ] # defining the architecture of the model using Functional API
    x = Input(shape = (input_neurons,))
    hidden1 = Dense(units=neuron_hidden_layer_1, activation='relu')(x)
    hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(hidden1)
    output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

    model_functional = Model(x, output)
```

```
[ ] # summary of the model
    model_functional.summary()
```

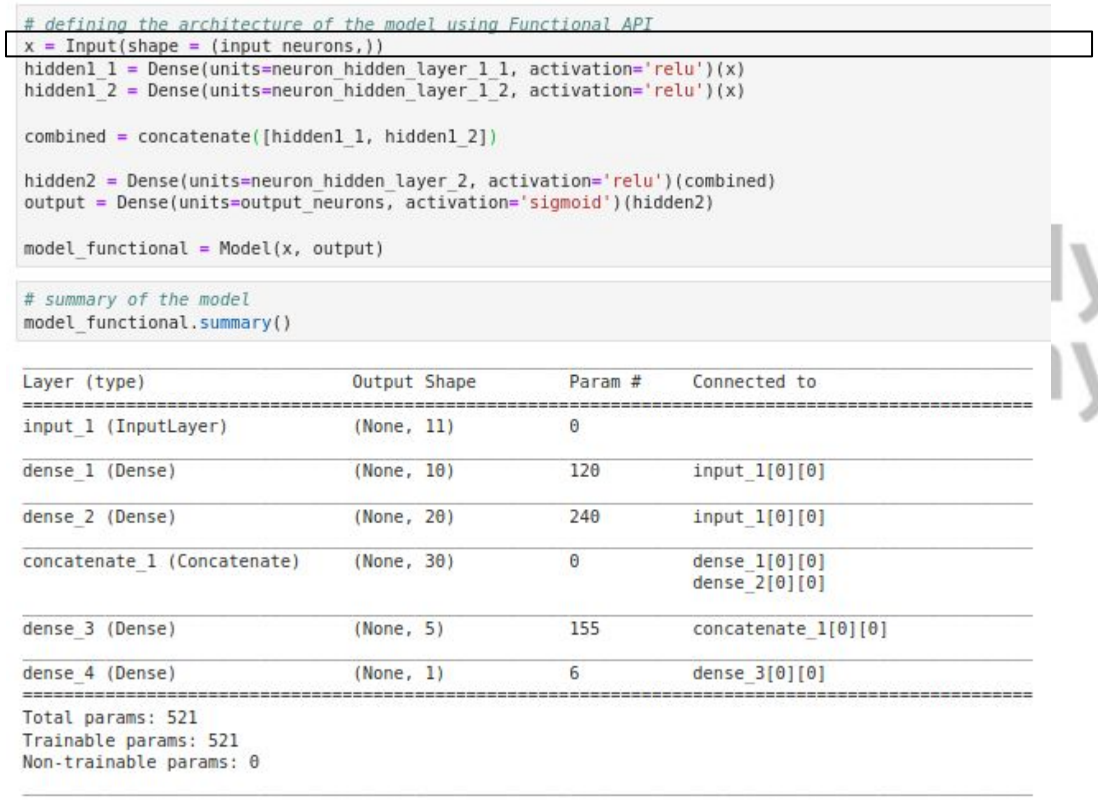
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 11)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                120
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 55
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 6
=================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
_________________________________________________________________
```



Build a Ad-hoc architecture using Functional API



Build a Ad-hoc architecture using Functional API



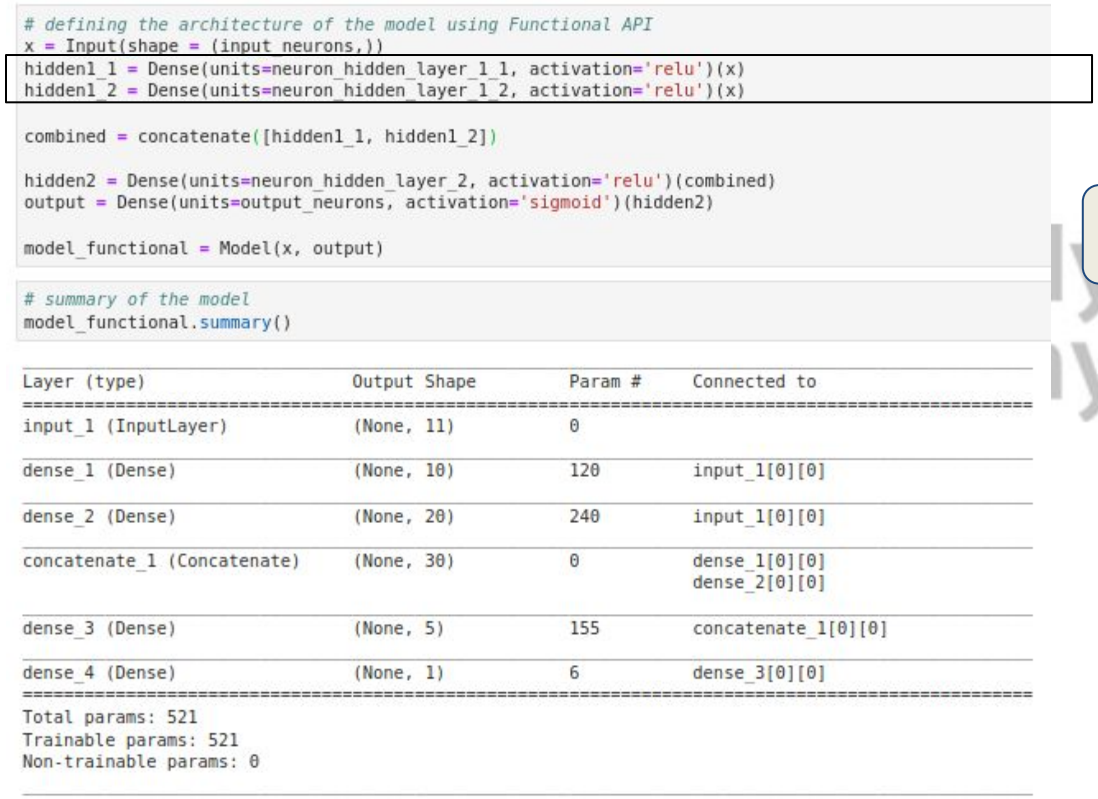
```
# defining the architecture of the model using Functional API
x = Input(shape = (input_neurons,))
hidden1_1 = Dense(units=neuron_hidden_layer_1_1, activation='relu')(x)
hidden1_2 = Dense(units=neuron_hidden_layer_1_2, activation='relu')(x)

combined = concatenate([hidden1_1, hidden1_2])

hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(combined)
output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

model_functional = Model(x, output)
```

```
# summary of the model
model_functional.summary()
```

```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_1 (InputLayer)             (None, 11)            0
____________________________________________________________________________________________________
dense_1 (Dense)                  (None, 10)            120         input_1[0][0]
____________________________________________________________________________________________________
dense_2 (Dense)                  (None, 20)            240         input_1[0][0]
____________________________________________________________________________________________________
concatenate_1 (Concatenate)      (None, 30)            0           dense_1[0][0]
                                                                   dense_2[0][0]
____________________________________________________________________________________________________
dense_3 (Dense)                  (None, 5)             155         concatenate_1[0][0]
____________________________________________________________________________________________________
dense_4 (Dense)                  (None, 1)             6           dense_3[0][0]
====================================================================================================
Total params: 521
Trainable params: 521
Non-trainable params: 0
____________________________________________________________________________________________________
```

Input Layer

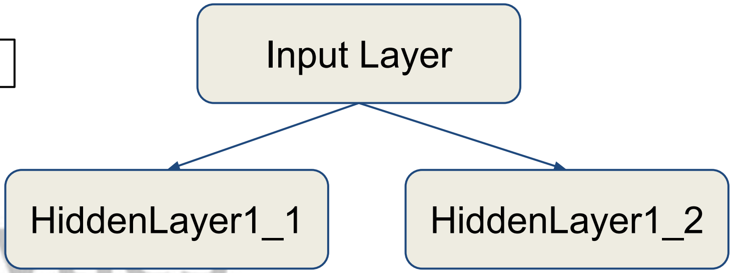
Build a Ad-hoc architecture using Functional API



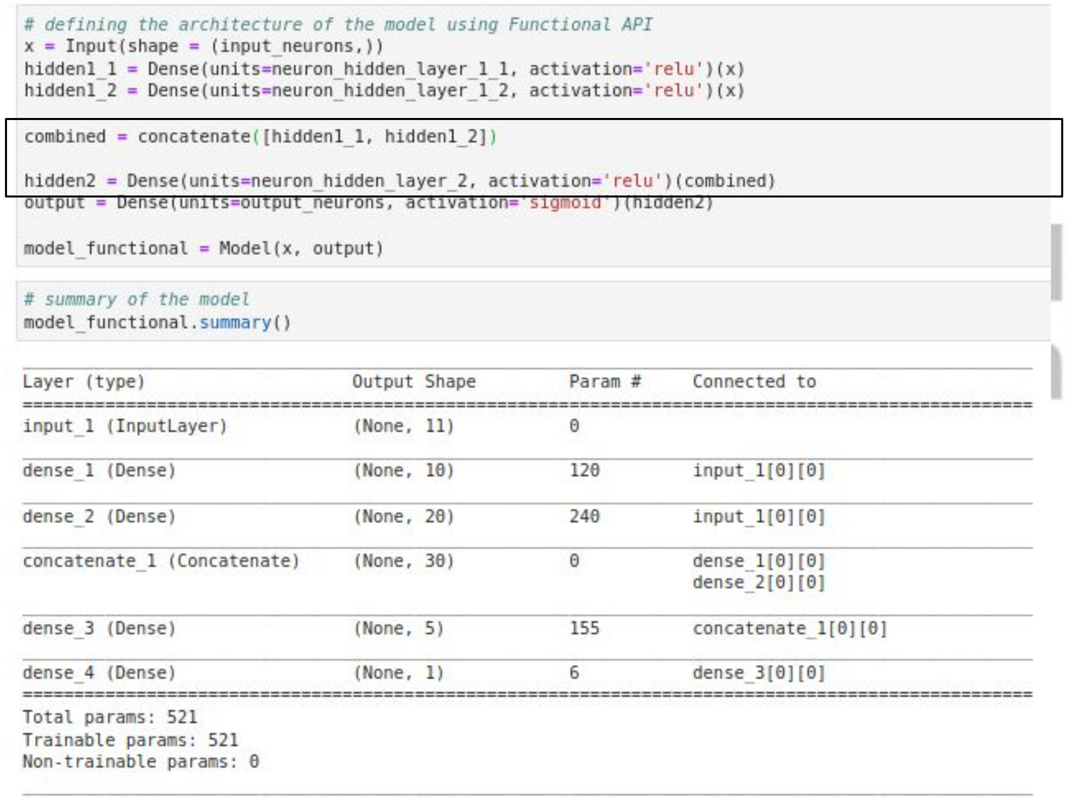
```
# defining the architecture of the model using Functional API
x = Input(shape = (input_neurons,))
hidden1_1 = Dense(units=neuron_hidden_layer_1_1, activation='relu')(x)
hidden1_2 = Dense(units=neuron_hidden_layer_1_2, activation='relu')(x)

combined = concatenate([hidden1_1, hidden1_2])

hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(combined)
output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

model_functional = Model(x, output)
```

```
# summary of the model
model_functional.summary()
```

```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_1 (InputLayer)             (None, 11)            0
____________________________________________________________________________________________________
dense_1 (Dense)                  (None, 10)            120         input_1[0][0]
____________________________________________________________________________________________________
dense_2 (Dense)                  (None, 20)            240         input_1[0][0]
____________________________________________________________________________________________________
concatenate_1 (Concatenate)      (None, 30)            0           dense_1[0][0]
                                                                   dense_2[0][0]
____________________________________________________________________________________________________
dense_3 (Dense)                  (None, 5)             155         concatenate_1[0][0]
____________________________________________________________________________________________________
dense_4 (Dense)                  (None, 1)             6           dense_3[0][0]
====================================================================================================
Total params: 521
Trainable params: 521
Non-trainable params: 0
____________________________________________________________________________________________________
```



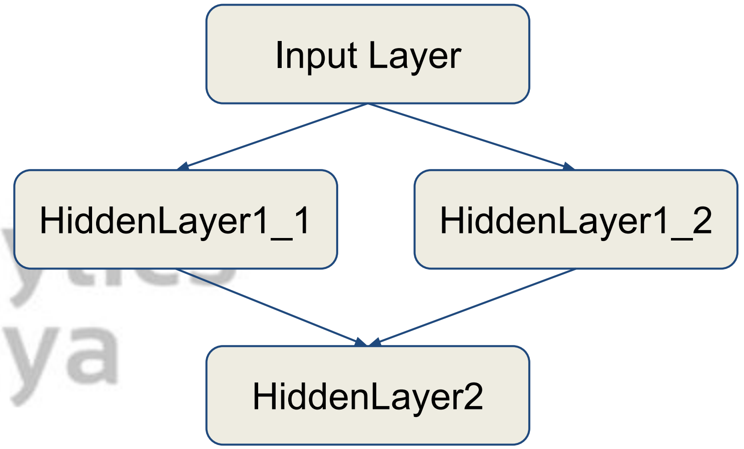
Build a Ad-hoc architecture using Functional API



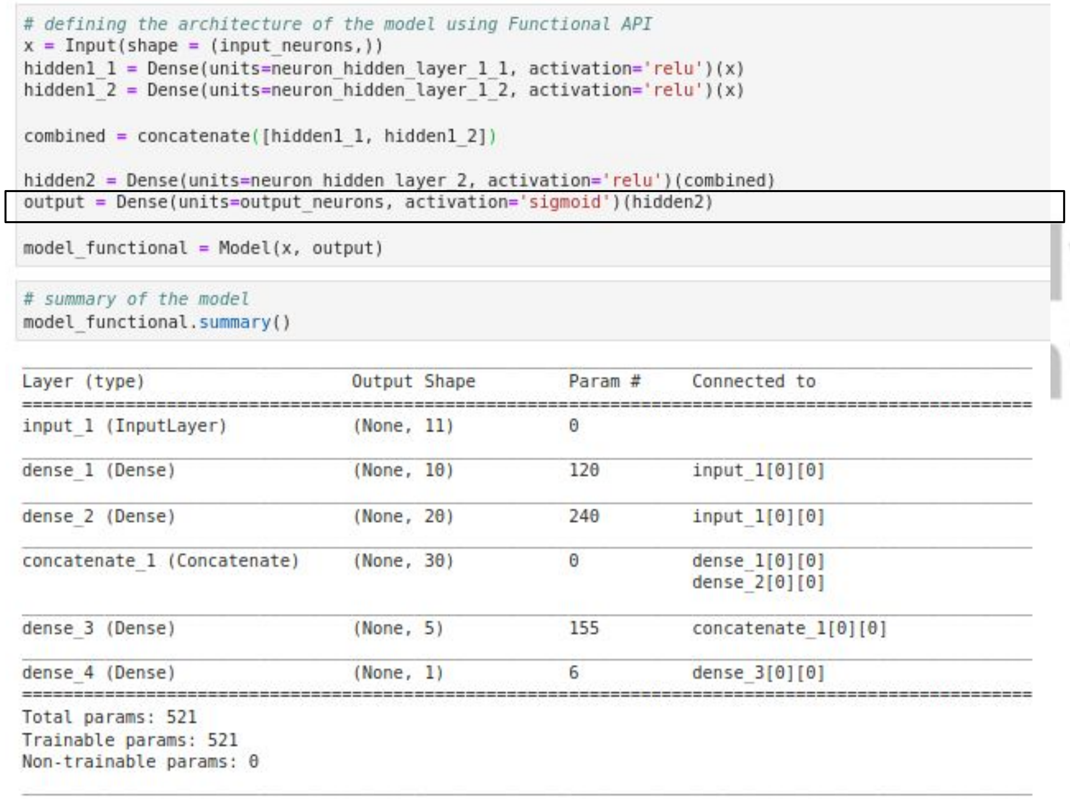
```
# defining the architecture of the model using Functional API
x = Input(shape = (input_neurons,))
hidden1_1 = Dense(units=neuron_hidden_layer_1_1, activation='relu')(x)
hidden1_2 = Dense(units=neuron_hidden_layer_1_2, activation='relu')(x)

combined = concatenate([hidden1_1, hidden1_2])

hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(combined)
output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

model_functional = Model(x, output)
```

```
# summary of the model
model_functional.summary()
```

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 11)           0
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 10)           120         input_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 20)           240         input_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 30)           0           dense_1[0][0]
                                                                 dense_2[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 5)            155         concatenate_1[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 1)            6           dense_3[0][0]
==================================================================================================
Total params: 521
Trainable params: 521
Non-trainable params: 0
__________________________________________________________________________________________________
```



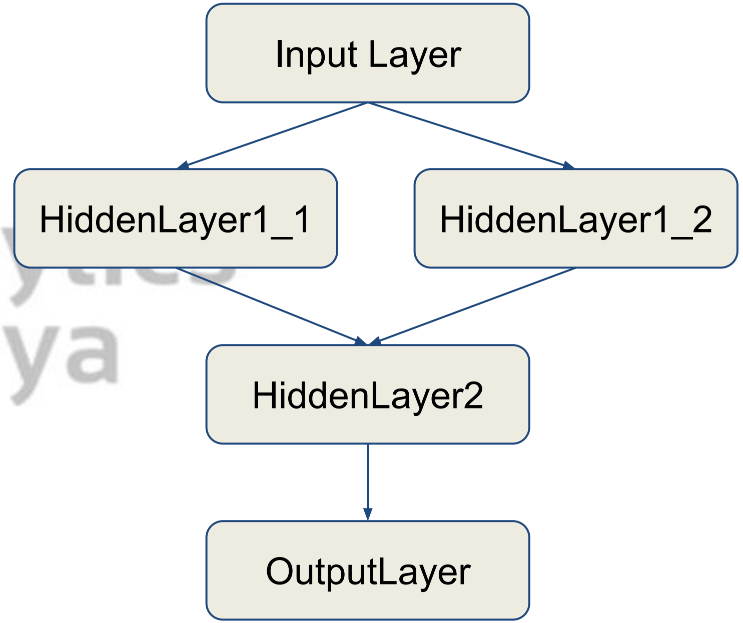
Build a Ad-hoc architecture using Functional API



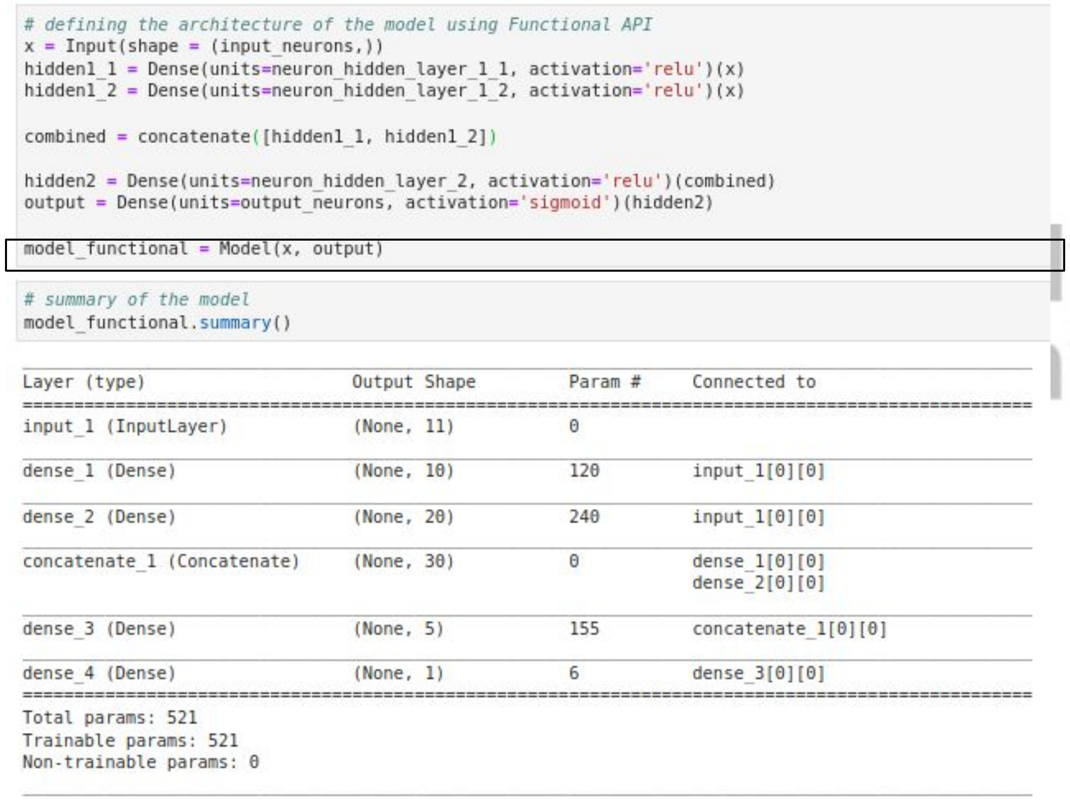
```
# defining the architecture of the model using Functional API
x = Input(shape = (input_neurons,))
hidden1_1 = Dense(units=neuron_hidden_layer_1_1, activation='relu')(x)
hidden1_2 = Dense(units=neuron_hidden_layer_1_2, activation='relu')(x)

combined = concatenate([hidden1_1, hidden1_2])

hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(combined)
output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

model_functional = Model(x, output)
```

```
# summary of the model
model_functional.summary()
```

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 11)           0

dense_1 (Dense)                 (None, 10)           120         input_1[0][0]

dense_2 (Dense)                 (None, 20)           240         input_1[0][0]

concatenate_1 (Concatenate)     (None, 30)           0           dense_1[0][0]
                                                                 dense_2[0][0]

dense_3 (Dense)                 (None, 5)            155         concatenate_1[0][0]

dense_4 (Dense)                 (None, 1)            6           dense_3[0][0]
==================================================================================================
Total params: 521
Trainable params: 521
Non-trainable params: 0
```



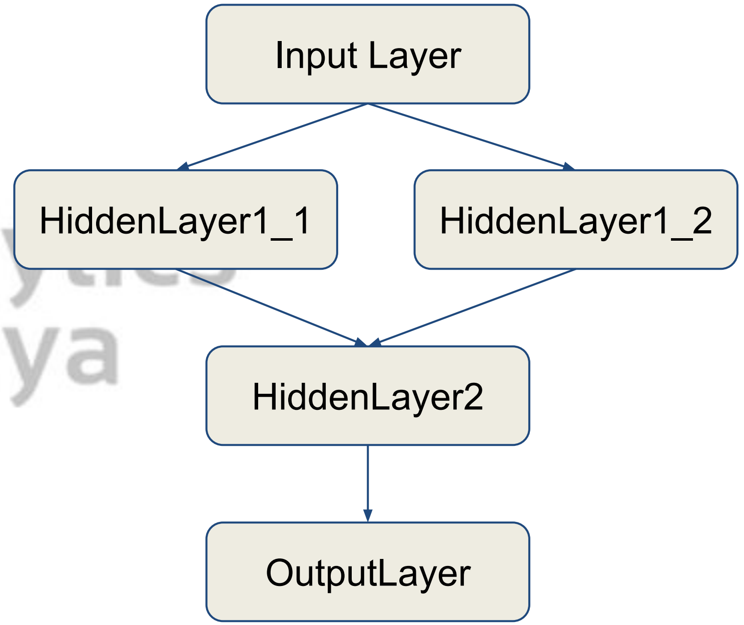
Build a Ad-hoc architecture using Functional API



```
# defining the architecture of the model using Functional API
x = Input(shape = (input_neurons,))
hidden1_1 = Dense(units=neuron_hidden_layer_1_1, activation='relu')(x)
hidden1_2 = Dense(units=neuron_hidden_layer_1_2, activation='relu')(x)

combined = concatenate([hidden1_1, hidden1_2])

hidden2 = Dense(units=neuron_hidden_layer_2, activation='relu')(combined)
output = Dense(units=output_neurons, activation='sigmoid')(hidden2)

model_functional = Model(x, output)
```

```
# summary of the model
model_functional.summary()
```

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 11)           0

dense_1 (Dense)                 (None, 10)           120         input_1[0][0]

dense_2 (Dense)                 (None, 20)           240         input_1[0][0]

concatenate_1 (Concatenate)     (None, 30)           0           dense_1[0][0]
                                                                 dense_2[0][0]

dense_3 (Dense)                 (None, 5)            155         concatenate_1[0][0]

dense_4 (Dense)                 (None, 1)            6           dense_3[0][0]
==================================================================================================
Total params: 521
Trainable params: 521
Non-trainable params: 0
```



Thank You