

Sequential Model

Learn how a neural network is built in Keras.

Chapter Goals:

- Initialize an MLP model in Keras

A. Building the MLP

In Keras, every neural network model is an instance of the `Sequential` object. This acts as the container of the neural network, allowing us to build the model by stacking multiple layers inside the `Sequential` object.

The most commonly used Keras neural network layer is the `Dense` layer. This represents a fully-connected layer in the neural network, and it is the most important building block of an MLP model.

When building a model, we start off by initializing a `Sequential` object. We can initialize an empty `Sequential` object and add layers onto the model using the `add` function, or we can directly initialize the `Sequential` object with a list of layers.

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



Adding two Dense layers to a Sequential model.

```
layer1 = Dense(5, input_dim=4)
layer2 = Dense(3, activation='relu')
model = Sequential([layer1, layer2])
```



Initializing a Sequential model with two Dense layers.

The `Dense` object takes in a single required argument, which is the number of neurons in the fully-connected layer. The `activation` keyword argument specifies the activation function for the layer (the default is no activation). In the code snippets above, we used no activation for `layer1` and ReLU activation for `layer2`.

The first layer of the `Sequential` model represents the input layer. Therefore, in the first layer we need to specify the feature dimension of the input data for the model, which we do with the `input_dim` keyword argument.

In the code snippets above, we set the input feature dimension to 4, meaning that the input data has shape `(batch_size, 4)` (where `batch_size` is the data's batch size, decided at runtime).

Time to code!

The coding exercise for this chapter involves setting up a Keras `Sequential` model with a single `Dense` layer. We start off with an empty initialized `Sequential` object.

Set `model` equal to `Sequential` initialized with no arguments.

```
# CODE HERE
```



We'll build a three layer MLP model. The first layer will consist of 5 neurons and use ReLU activation. It will also act as the input layer for the model.

To create the input layer, we'll initialize a `Dense` object with the requisite number of neurons and activation. We'll also set the `input_dim` keyword argument to `2`, which represents the feature dimension of the input data for the model.

Set `layer1` equal to a `Dense` with `5` as the required argument, `'relu'` for the `activation` keyword argument, and `2` for the `input_dim` keyword argument.

Then call `model.add` on `layer1`.

CODE HERE

