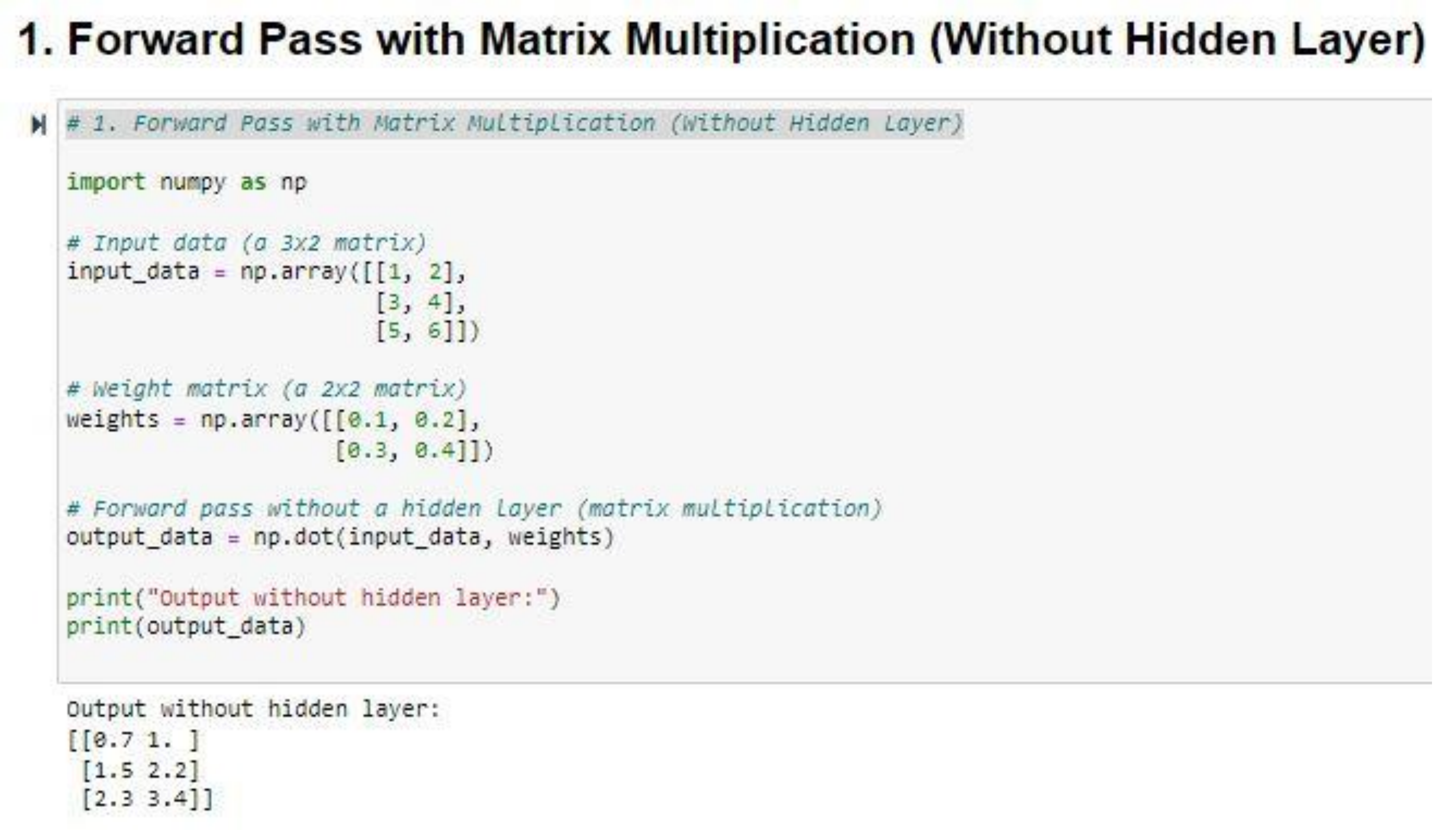
Experiment 6

Title: Forward pass with matrix multiplication Forward pass with hidden layer (matrix multiplication) Forward pass with matrix multiplication with Keras Forward passes with hidden layer (matrix multiplication) with Keras.

Aim: To Implement Forward pass with matrix multiplication Forward pass with hidden layer Forward pass with matrix multiplication with Keras Forward passes with hidden layer with Keras.

Overview of implementing forward passes using matrix multiplication and Keras for both a single-layer neural network (perceptron) and a multi-layer neural network (with a hidden layer):

1) Forward Pass with Matrix Multiplication in Keras (Single-Layer Neural Network).
2) Define a Keras model with a single layer: model
3) To perform the forward pass and obtain the output.
4) Forward Pass with Hidden Layer in Keras (Multi-Layer Neural Network):
5) Create input data.
6) Define a Keras model with a hidden layer and an output layer.
7) To perform the forward pass and obtain the output.

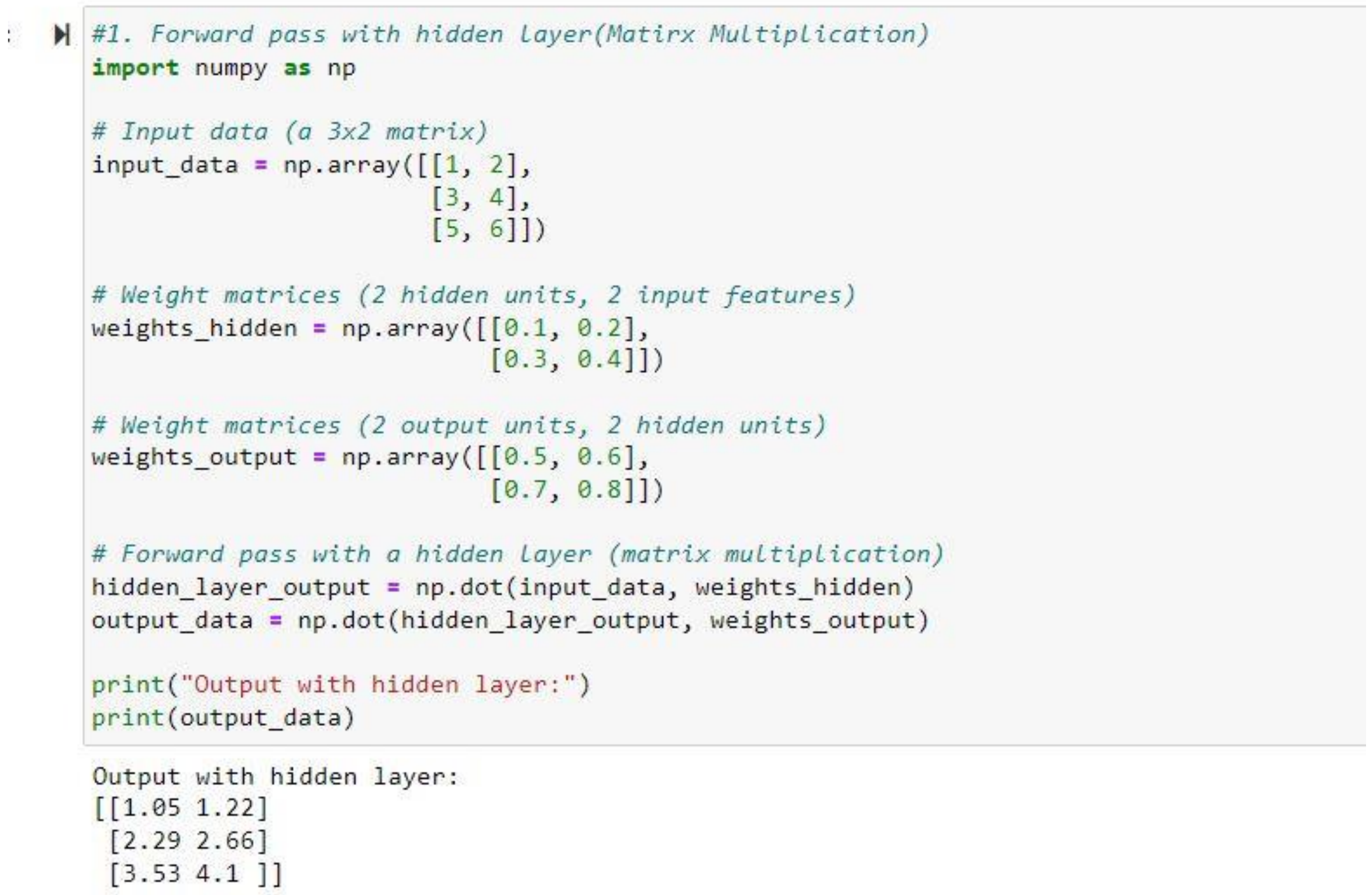


1. Forward Pass with Matrix Multiplication (Without Hidden Layer)

```
# 1. Forward Pass with Matrix Multiplication (Without Hidden Layer)

import numpy as np

# Input data (a 3x2 matrix)
input_data = np.array([[1, 2],
                       [3, 4],
                       [5, 6]])

# Weight matrix (a 2x2 matrix)
weights = np.array([[0.1, 0.2],
                    [0.3, 0.4]])

# Forward pass without a hidden layer (matrix multiplication)
output_data = np.dot(input_data, weights)

print("Output without hidden layer:")
print(output_data)
```

```
Output without hidden layer:
[[0.7 1. ]
 [1.5 2.2]
 [2.3 3.4]]
```

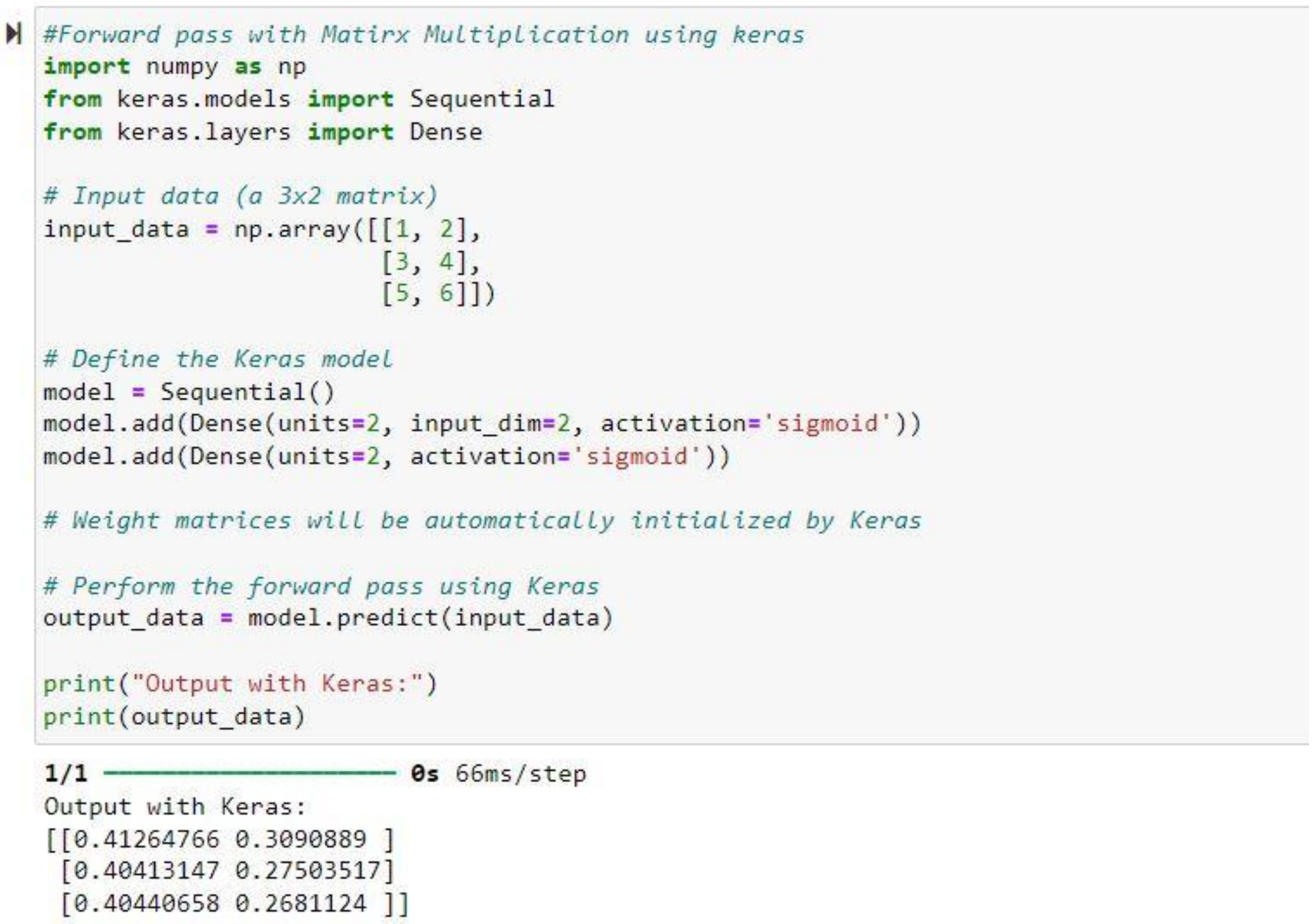
2. Forward pass with hidden layer(Matrix Multiplication)



```
#1. Forward pass with hidden layer(Matirx Multiplication)
import numpy as np

# Input data (a 3x2 matrix)
input_data = np.array([[1, 2],
                       [3, 4],
                       [5, 6]])

# Weight matrices (2 hidden units, 2 input features)
weights_hidden = np.array([[0.1, 0.2],
                           [0.3, 0.4]])

# Weight matrices (2 output units, 2 hidden units)
weights_output = np.array([[0.5, 0.6],
                           [0.7, 0.8]])

# Forward pass with a hidden layer (matrix multiplication)
hidden_layer_output = np.dot(input_data, weights_hidden)
output_data = np.dot(hidden_layer_output, weights_output)

print("Output with hidden layer:")
print(output_data)
```

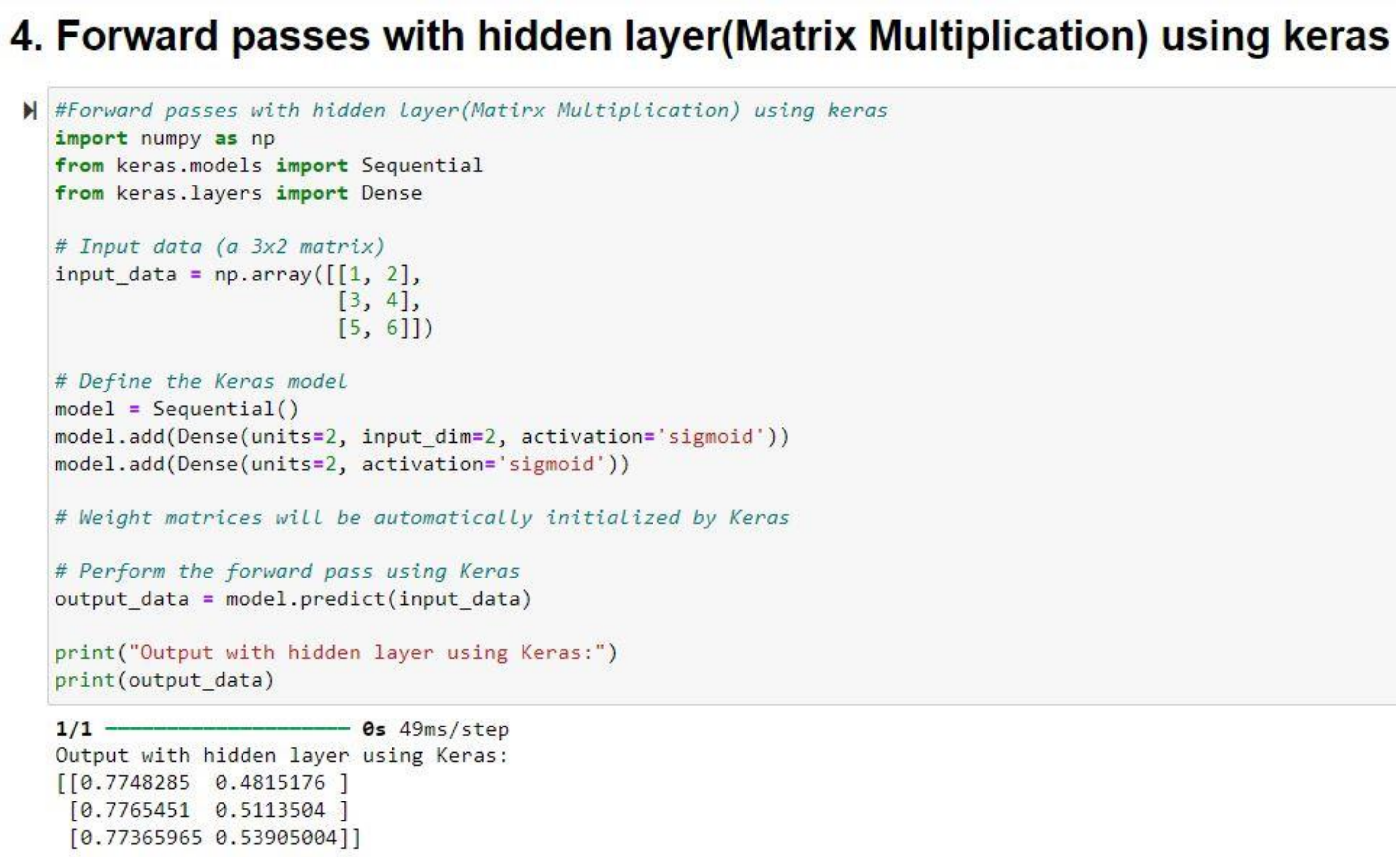
```
Output with hidden layer:
[[1.05 1.22]
 [2.29 2.66]
 [3.53 4.1 ]]
```

3. Forward pass with Matrix Multiplication using keras



```
#Forward pass with Matirx Multiplication using keras
import numpy as np
from keras.models import Sequential
from keras.layers import Dense

# Input data (a 3x2 matrix)
input_data = np.array([[1, 2],
                       [3, 4],
                       [5, 6]])

# Define the Keras model
model = Sequential()
model.add(Dense(units=2, input_dim=2, activation='sigmoid'))
model.add(Dense(units=2, activation='sigmoid'))

# Weight matrices will be automatically initialized by Keras

# Perform the forward pass using Keras
output_data = model.predict(input_data)

print("Output with Keras:")
print(output_data)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
Output with Keras:
[[0.41264766 0.3090889 ]
 [0.40413147 0.27503517]
 [0.40440658 0.2681124 ]]
```



4. Forward passes with hidden layer(Matrix Multiplication) using keras

```python
#Forward passes with hidden layer(Matirx Multiplication) using keras
import numpy as np
from keras.models import Sequential
from keras.layers import Dense

# Input data (a 3x2 matrix)
input_data = np.array([[1, 2],
                       [3, 4],
                       [5, 6]])

# Define the Keras model
model = Sequential()
model.add(Dense(units=2, input_dim=2, activation='sigmoid'))
model.add(Dense(units=2, activation='sigmoid'))

# Weight matrices will be automatically initialized by Keras

# Perform the forward pass using Keras
output_data = model.predict(input_data)

print("Output with hidden layer using Keras:")
print(output_data)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Output with hidden layer using Keras:
[[0.7748285  0.4815176 ]
 [0.7765451  0.5113504 ]
 [0.77365965 0.53905004]]
```

Conclusion: Successful Implement Forward pass with matrix multiplication Forward pass with hidden layer Forward pass with matrix multiplication with Keras Forward passes with hidden layer.