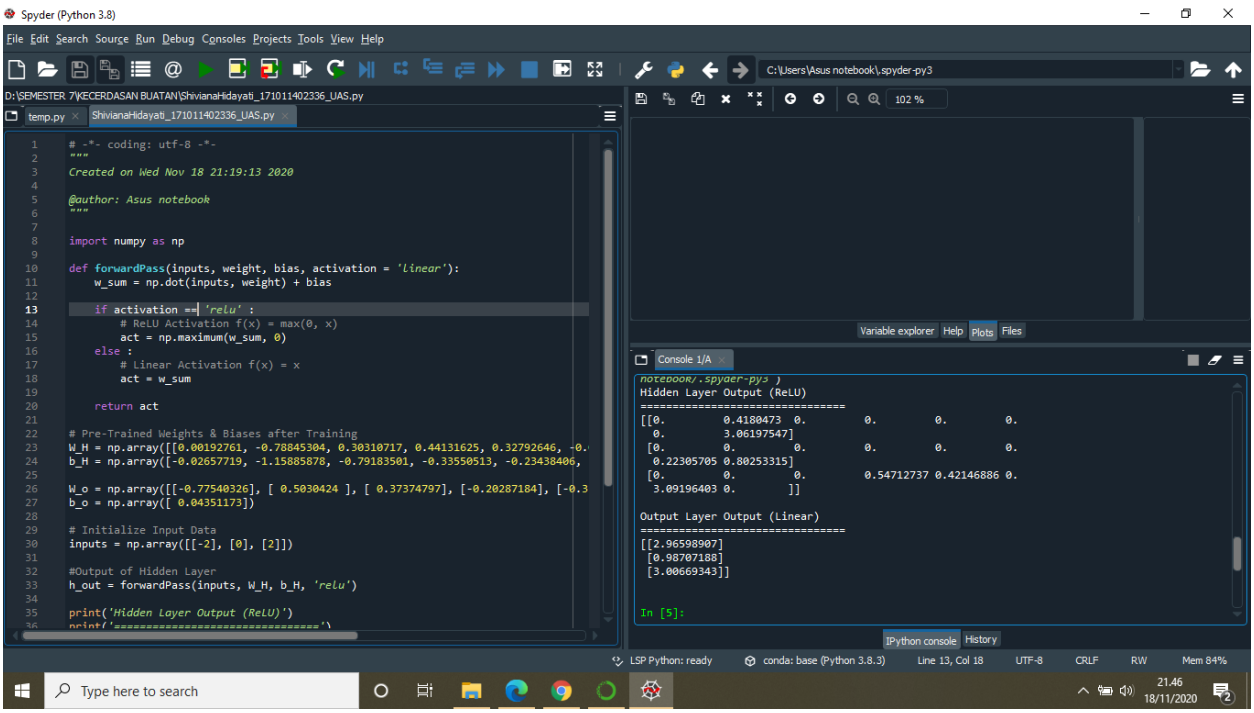
Metode Multilayer Perceptron

Shiviana Hidayati (171011402336)

```
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 18 21:19:13 2020

@author: Asus notebook
"""

import numpy as np

def forwardPass(inputs, weight, bias, activation = 'linear'):
    w_sum = np.dot(inputs, weight) + bias

    if activation == 'relu' :
        # ReLU Activation f(x) = max(0, x)
        act = np.maximum(w_sum, 0)
    else :
        # Linear Activation f(x) = x
        act = w_sum

    return act

# Pre-Trained Weights & Biases after Training
W_H = np.array([[0.00192761, -0.78845304, 0.30310717, 0.44131625, 0.32792646, -0.02451803,
1.43445349, -1.12972116]])
```

```
b_H = np.array([-0.02657719, -1.15885878, -0.79183501, -0.33550513, -0.23438406, -0.25078532, 0.22305705, 0.80253315])


W_o = np.array([[-0.77540326], [ 0.5030424 ], [ 0.37374797], [-0.20287184], [-0.35956827], [-0.54576212], [ 1.04326093], [ 0.8857621 ]])

b_o = np.array([ 0.04351173])


# Initialize Input Data

inputs = np.array([[-2], [0], [2]])


#Output of Hidden Layer

h_out = forwardPass(inputs, W_H, b_H, 'relu')


print('Hidden Layer Output (ReLU)')

print('================================')

print(h_out, "\n")


# Output of Output Layer

o_out = forwardPass(h_out, W_o, b_o, 'linear')


print('Output Layer Output (Linear)')

print('================================')

print(o_out, "\n")


"""[[ 2.96598907]

 [ 0.98707188]

 [ 3.00669343]]"""
```


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Asus notebook\.spyder-py3
D:\SEMESTER 7\KECERDASAN BUATAN\ShivianaHidayati_171011402336_UAS.py
temp.py
ShivianaHidayati_171011402336_UAS.py
-*- coding: utf-8 -*-
"""
Created on Wed Nov 18 21:19:13 2020
@author: Asus notebook
"""
import numpy as np
def forwardPass(inputs, weight, bias, activation = 'linear'):
w_sum = np.dot(inputs, weight) + bias
if activation == 'relu' :
# ReLU Activation f(x) = max(0, x)
act = np.maximum(w_sum, 0)
else :
# Linear Activation f(x) = x
act = w_sum
return act
# Pre-Trained Weights & Biases after Training
W_H = np.array([[0.00192761, -0.78845304, 0.30310717, 0.44131625, 0.32792646, -0.
b_H = np.array([-0.02657719, -1.15885878, -0.79183501, -0.33550513, -0.23438406,
W_o = np.array([[-0.77540326], [0.5030424], [0.37374797], [-0.20287184], [-0.3
b_o = np.array([0.04351173])
# Initialize Input Data
inputs = np.array([[-2], [0], [2]])
#Output of Hidden Layer
h_out = forwardPass(inputs, W_H, b_H, 'relu')
print('Hidden Layer Output (ReLU)')
Variable explorer Help Plots Files
Console 1/A
Hidden Layer Output (ReLU)
[[0. 0.4180473 0. 0. 0. 0.
0. 3.06197547]
[0. 0. 0. 0. 0. 0.
0.22305705 0.80253315]
[0. 0. 0. 0.54712737 0.42146886 0.
3.09196403 0.]]
Output Layer Output (Linear)
[[2.96598907]
[0.98707188]
[3.00669343]]
In [5]:
IPython console History
LSP Python: ready
conda: base (Python 3.8.3)
Line 13, Col 18
UTF-8
CRLF
RW
Mem 84%
Type here to search
21.46
18/11/2020