Lekcja 3 Uczenie perceptronu

November 7, 2018

1 Polish Japanese School of Information Technology

1.1 Subject: Artificial Intelligence Tools

1.1.1 Autor: Mateusz Gruszka

1.1.2 Index: s16251

Date: 7.11.18

Exercise 6 Zadanie 6. Dla sieci neuronowych podanych na Rysunku 6, - a) napisac macierze wag. - b) wyznaczyc sygna wyjsciowy z sieci, jesli wektor wejsciowy jest [1,1].

Rozpatrywac nastepujace przypadki: * funkcja aktywacji jest funkcja progowa unipolarna. * funkcja aktywacji jest funkcja sigmoidalna unipolarna (= 1). * funkcja aktywacji jest funkcja progowa bipolarna. * funkcja aktywacji jest funkcja sigmoidalna bipolarna (= 1).

Data:

```
In [171]: import numpy as np
          import math
          X_input = np.array([1,-1,0])
          B_input = np.array([0,0,1])

          alpha = 1
```

Activation functions:

```
In [172]: def f_bin_unipolar(net):
              if(net >= 0):
                  return 1
              else:
                  return 0

          def f_bin_bipolar(net):
              if(net >= 0):
                  return 1
              else:
                  return -1
```

```
def f_sig_unipolar(net):
    return 1 / (1 + math.exp(-1*net*alpha))


def f_sig_bipolar(net):
    return (2 / (1 + math.exp(-1*net*alpha))) - 1
```

1.1.3 Neuron

a)

```
In [173]: W_1 = np.array([-1,3,0])
          bias_weight_1 = np.array([0,0,2])
          net_1 = W_1@X_input + B_input@bias_weight_1

          print("\x1b[31m OUTPUT: net array result: \x1b[0m")
          print(net_1)
```

```
 OUTPUT: net array result:
-2
```

b)

- funkcja aktywacji jest funkcja progowa unipolarna

```
In [174]: Y_1 = f_bin_unipolar(net_1)
          print("\x1b[31m OUTPUT: activation f_bin_unipolar function result: \x1b[0m")
          print(Y_1)
```

```
 OUTPUT: activation f_bin_unipolar function result:
0
```

- funkcja aktywacji jest funkcja sigmoidalna unipolarna (= 1).

```
In [175]: Y_1 = f_sig_unipolar(net_1)
          print("\x1b[31m OUTPUT: activation f_sig_unipolar function result: \x1b[0m")
          print(Y_1)
```

```
 OUTPUT: activation f_sig_unipolar function result:
0.11920292202211755
```

- funkcja aktywacji jest funkcja progowa bipolarna.

```
In [176]: Y_1 = f_bin_unipolar(net_1)
          print("\x1b[31m OUTPUT: activation f_bin_unipolar function result: \x1b[0m")
          print(Y_1)
```

```
 OUTPUT: activation f_bin_unipolar function result:
0
```

- funkcja aktywacji jest funkcja sigmoidalna bipolarna (= 1).

```
In [177]: Y_1 = f_sig_bipolar(net_1)
          print("\x1b[31m OUTPUT: activation f_sig_bipolar function result: \x1b[0m")
          print(Y_1)
```

```
 OUTPUT: activation f_sig_bipolar function result:
-0.7615941559557649
```

1.1.4 Jedno-warstwowa sie neuronowa

- funkcja aktywacji jest funkcja progowa unipolarna

a)

```
In [178]: W_2 = np.array([[2,-1,0],[-4,-5,0],[0,1,0]])
          bias_weight_2 = np.array([[0,0,2],[0,0,3],[0,0,-3]])
          net_2 = W_2@X_input + B_input@bias_weight_2
          print("\x1b[31m  OUTPUT: net array result: \x1b[0m")
          print(net_2)
```

```
  OUTPUT: net array result:
[ 3  1 -4]
```

b)

- funkcja aktywacji jest funkcja progowa unipolarna

```
In [179]: Y_2 = []
          for element in net:
              result = f_bin_unipolar(element)
              Y_2.append(result)
          print("\x1b[31m OUTPUT: activation f_bin_unipolar function result: \x1b[0m")
          print(Y_2)
```

```
 OUTPUT: activation f_bin_unipolar function result:
[1, 1, 0]
```

- funkcja aktywacji jest funkcja sigmoidalna unipolarna (= 1).

```
In [180]: Y_2 = []
          for element in net:
              result = f_sig_unipolar(element)
              Y_2.append(result)
          print("\x1b[31m OUTPUT: activation f_sig_unipolar function result: \x1b[0m")
          print(Y_2)
```

```
 OUTPUT: activation f_sig_unipolar function result:
[0.9525741268224334, 0.7310585786300049, 0.01798620996209156]
```

- funkcja aktywacji jest funkcja progowa bipolarna.

```
In [181]: Y_2 = []
          for element in net:
              result = f_bin_unipolar(element)
              Y_2.append(result)
          print("\x1b[31m OUTPUT: activation f_bin_unipolar function result: \x1b[0m")
          print(Y_2)
```

```
 OUTPUT: activation f_bin_unipolar function result:
[1, 1, 0]
```

- funkcja aktywacji jest funkcja sigmoidalna bipolarna (= 1).

```
In [182]: Y_2 = []
          for element in net:
              result = f_sig_bipolar(element)
              Y_2.append(result)
          print("\x1b[31m OUTPUT: activation f_sig_bipolar function result: \x1b[0m")
          print(Y_2)
```

```
 OUTPUT: activation f_sig_bipolar function result:
[0.9051482536448667, 0.4621171572600098, -0.9640275800758169]
```

1.1.5 Dwu-warstwowa sie neuronowa

- funkcja aktywacji jest funkcja progowa unipolarna

a)

```
In [183]: W_3_1 = np.array([[2,-1,0],[-4,-1,0],[0,1,0]])
          W_3_2 = np.array([[-3,-1,4,0],[1,-1,2,0]])
          bias_weight_3_1 = np.array([[0,0,0],[0,0,2],[0,0,-1]])
          bias_weight_3_2 = np.array([[0,0,0,0],[0,0,0,0]])
          B_input_3_2 = np.array([0,0,0,1])

          net_3_1 = W_3_1@X_input + B_input@bias_weight_3_1
```

b)

- funkcja aktywacji jest funkcja progowa unipolarna

```
In [184]: Y_3_1 = []
          for element in net_3_1:
              result = f_bin_unipolar(element)
              Y_3_1.append(result)

          Y_3_1.append(0)

          net_3_2 = W_3_2@Y_3_1+ bias_weight_3_2@B_input_3_2

          Y_final_layer_output = []
          for element in Y_3_1:
              result = f_bin_unipolar(element)
              Y_final_layer_output.append(result)


          print("\x1b[31m OUTPUT: activation f_bin_unipolar function result: \x1b[0m")
          print(Y_final_layer_output)

 OUTPUT: activation f_bin_unipolar function result:
[1, 1, 1, 1]
```

- funkcja aktywacji jest funkcja sigmoidalna unipolarna (= 1).

```
In [185]: Y_3_1 = []
          for element in net_3_1:
              result = f_sig_unipolar(element)
              Y_3_1.append(result)

          Y_3_1.append(0)

          net_3_2 = W_3_2@Y_3_1+ bias_weight_3_2@B_input_3_2

          Y_final_layer_output = []
          for element in Y_3_1:
              result = f_sig_unipolar(element)
              Y_final_layer_output.append(result)

          print("\x1b[31m OUTPUT: activation f_sig_unipolar function result: \x1b[0m")
          print(Y_final_layer_output)

 OUTPUT: activation f_sig_unipolar function result:
[0.7216325609518421, 0.5118542464834632, 0.5297654931903004, 0.5]
```

- funkcja aktywacji jest funkcja progowa bipolarna.

```
In [186]: Y_3_1 = []
          for element in net_3_1:
```

```
            result = f_bin_unipolar(element)
            Y_3_1.append(result)

        Y_3_1.append(0)

        net_3_2 = W_3_2@Y_3_1+ bias_weight_3_2@B_input_3_2

        Y_final_layer_output = []
        for element in Y_3_1:
            result = f_bin_unipolar(element)
            Y_final_layer_output.append(result)

        print("\x1b[31m OUTPUT: activation f_bin_unipolar function result: \x1b[0m")
        print(Y_final_layer_output)
```

```
 OUTPUT: activation f_bin_unipolar function result:
[1, 1, 1, 1]
```

- funkcja aktywacji jest funkcja sigmoidalna bipolarna (= 1).

```
In [187]: Y_3_1 = []
          for element in net_3_1:
              result = f_sig_bipolar(element)
              Y_3_1.append(result)

          Y_3_1.append(0)

          net_3_2 = W_3_2@Y_3_1+ bias_weight_3_2@B_input_3_2

          Y_final_layer_output = []
          for element in Y_3_1:
              result = f_sig_bipolar(element)
              Y_final_layer_output.append(result)

          print("\x1b[31m OUTPUT: activation f_sig_bipolar function result: \x1b[0m")
          print(Y_final_layer_output)
```

```
 OUTPUT: activation f_sig_bipolar function result:
[0.42401264054072985, -0.42401264054072973, -0.36339948438905245, 0.0]
```