**Podstawy Sztucznej Inteligencji – Laboratorium nr 6**

Wykonał: Kamil Wieniecki

Temat ćwiczenia: Budowa i działanie sieci Kohonena dla WTM.

1. Cel ćwiczenia

Celem ćwiczenia jest poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTM do odwzorowywania istotnych cech liter alfabetu.

1. Realizacja ćwiczenia

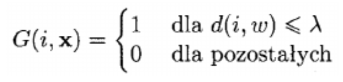
Wybrany przeze mnie język programowania do wykonania laboratorium to **Python.**

Sieć Kohonena nazywana jest inaczej mapą samoorganizującą. Bazuje na algorytmie uczenia bez nauczyciela, wagi na początku są losowane. Charakterystyczne jest tutaj uczenie konkurencyjne tzn. neurony konkurują ze sobą o prawo do reprezentacji danych wejściowych.

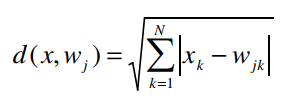
Metoda WTM (Winner Takes Most – „Zwycięzca bierze większość”) działa na zasadzie aktualizacji wag nie tylko neuronu zwycięskiego, ale również neuronów z jego sąsiedztwa. Neuron sąsiadujący to taki, który znajduje się w określonym promieniu od neuronu zwycięskiego. Wartość promienia definiuje się indywidualnie. Neuron którego wagi najmniej różnią się od składowych wektora wejściowego jest wygranym. Relacja opisująca ten stan wygląda następująco:



Neurony, które nie zostają uznane za sąsiednie nie zmieniają swoich wag. Jest to opisane następującą funkcją:



Wykorzystana przeze mnie miara odległości między wektorami to norma Manhattan, która przedstawia się następująco:



Adaptacja wag opisana jest wzorem:



, gdzie **η – współczynnik uczenia**

**Xk – wartość składowej wektora wejściowego**

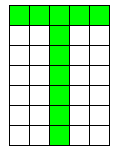
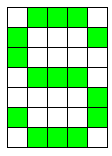
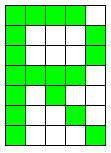
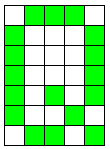
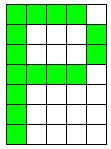
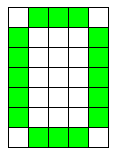
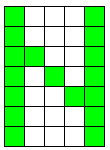
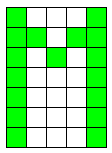
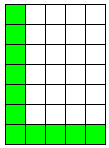
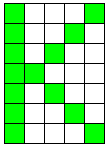
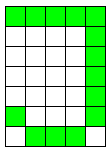
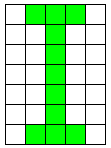
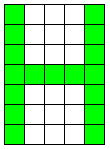
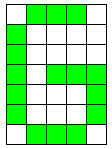
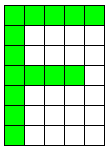
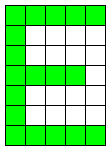
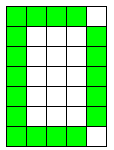
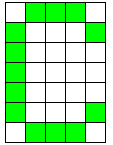
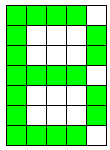
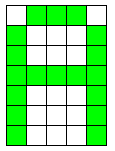
**Wj\*k – wartość składowej wektora wag**

Nadmiarowa ilość neuronów wynika z inicjalizacji wag losowo – przez co część neuronów może znaleźć się w strefie, gdzie nie ma danych lub ich liczba jest znikoma. Neurony te automatycznie zostają wykluczone z sieci, ponieważ nie mają szans na wygraną, nazywamy je neuronami martwymi.

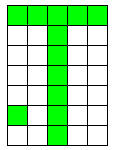
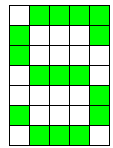
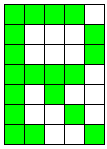
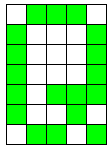
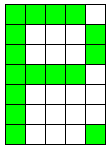
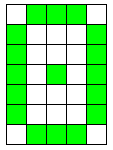
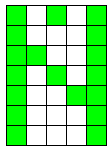
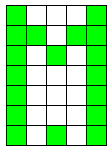
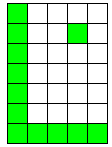
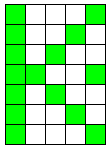
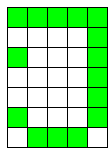
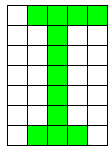
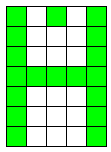
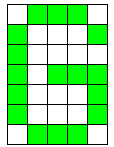
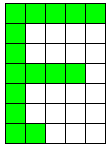
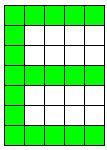
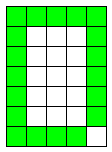
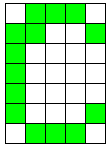
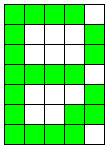
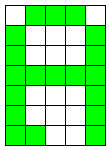
Dane wejściowe tworzą samoistną „bazę wiedzy” na podstawie której sieć uczy się i podejmuje decyzje.

Dane wejściowe zostały zaczerpnięte ze strony zaproponowanej w instrukcji laboratorium. Litery „zaszumione” (testowe) zostały przeze mnie.

Dane wejściowe prezentują się następująco:



Natomiast „zaszumione” dane testowe wyglądają następująco:



Do każdej litery dodany został dokładnie jeden nadmiarowy pixel.

Dane te przedstawione są w systemie binarnym, gdzie zielone pola opisane są liczbą 1, natomiast białe są puste, czyli mają wartość 0.

1. Wyniki

Wyniki przeprowadzone zostały dla trzech różnych współczynników uczenia – 0,01, 0,1 oraz 0,3. Ilość wykorzystanych neuronów do nauki to XXX, natomiast promienie sąsiedztwa wynosiły XXXXX dla każdej z prób. Wyniki zestawiłem w tabeli:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Współczynnik uczenia | 0,01 | | 0,1 | | 0,3 | |
| Promień sąsiedztwa |  |  |  |  |  |  |
| Lp. | Poprawność nauki [%] | Liczba epok | Poprawność nauki [%] | Liczba epok | Poprawność nauki [%] | Liczba epok |
| 1. |  |  |  |  |  |  |
| 2. |  |  |  |  |  |  |
| 3. |  |  |  |  |  |  |
| 4. |  |  |  |  |  |  |
| 5. |  |  |  |  |  |  |
| 6. |  |  |  |  |  |  |
| 7. |  |  |  |  |  |  |
| 8. |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Współczynnik uczenia | 0,01 | | 0,1 | | 0,3 | |
| Promień sąsiedztwa |  |  |  |  |  |  |
| Maksymalna ilość epok |  |  |  |  |  |  |
| Minimalna ilość epok |  |  |  |  |  |  |
| Średnia ilość epok |  |  |  |  |  |  |
| Maksymalna poprawność nauki [%] |  |  |  |  |  |  |
| Minimalna poprawność nauki [%] |  |  |  |  |  |  |
| Średnia poprawność nauki [%] |  |  |  |  |  |  |

1. Analiza wyników

TODO

1. Wnioski

TODO

1. Listing kodu

**Main.py**

**Kohonen.py**

from random import uniform

class Kohonen:

def \_\_init\_\_(self, inputs):

self.inputs = inputs

self.weights = []

for i in range(0, inputs):

self.weights.append(uniform(0, 1))

def learn(self, vector, learning\_rate):

"Metoda uczenia"

for i in range(0, self.inputs):

self.weights[i] += learning\_rate \* (vector[i] - self.weights[i])

**Letters.py**

learn\_letters = [

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 1

],

[

1, 1, 1, 1, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 0,

1, 0, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1

],

[

0, 1, 1, 1, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

0, 0, 0, 0, 1,

0, 0, 0, 0, 1,

0, 0, 0, 0, 1,

0, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 0, 0, 0, 1,

1, 0, 0, 1, 0,

1, 0, 1, 0, 0,

1, 1, 0, 0, 0,

1, 0, 1, 0, 0,

1, 0, 0, 1, 0,

1, 0, 0, 0, 1

],

[

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 1

],

[

1, 0, 0, 0, 1,

1, 1, 0, 1, 1,

1, 0, 1, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1

],

[

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 0, 0, 1,

1, 0, 1, 0, 1,

1, 0, 0, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 1, 0, 1,

1, 0, 0, 1, 0,

0, 1, 1, 0, 1

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0,

1, 0, 1, 0, 0,

1, 0, 0, 1, 0,

1, 0, 0, 0, 1

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 0,

0, 1, 1, 1, 0,

0, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0

]

]

test\_letters = [

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 0, 0, 1

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 1, 1,

1, 1, 1, 1, 0

],

[

0, 1, 1, 1, 0,

1, 1, 0, 0, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 1

],

[

1, 1, 1, 1, 1,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 0, 0, 0

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 0,

1, 0, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 1

],

[

1, 0, 1, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1

],

[

0, 1, 1, 1, 1,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

0, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 0, 0, 0, 1,

0, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 0, 0, 0, 1,

1, 0, 0, 1, 0,

1, 0, 1, 0, 0,

1, 1, 0, 0, 1,

1, 0, 1, 0, 0,

1, 0, 0, 1, 0,

1, 0, 0, 0, 1

],

[

1, 0, 0, 0, 0,

1, 0, 0, 1, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 1, 1, 1, 1

],

[

1, 0, 0, 0, 1,

1, 1, 0, 1, 1,

1, 0, 1, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 1, 0, 1

],

[

1, 0, 1, 0, 1,

1, 0, 0, 0, 1,

1, 1, 0, 0, 1,

1, 0, 1, 0, 1,

1, 0, 0, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 1, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 0,

1, 0, 0, 0, 1

],

[

0, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 0, 1, 1, 1,

1, 0, 0, 1, 0,

0, 1, 1, 0, 1

],

[

1, 1, 1, 1, 0,

1, 0, 0, 0, 1,

1, 0, 0, 0, 1,

1, 1, 1, 1, 0,

1, 0, 1, 0, 0,

1, 0, 0, 1, 0,

1, 1, 0, 0, 1

],

[

0, 1, 1, 1, 1,

1, 0, 0, 0, 1,

1, 0, 0, 0, 0,

0, 1, 1, 1, 0,

0, 0, 0, 0, 1,

1, 0, 0, 0, 1,

0, 1, 1, 1, 0

],

[

1, 1, 1, 1, 1,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

1, 0, 1, 0, 0,

0, 0, 1, 0, 0

]

]

1. Bibliografia

<http://aragorn.pb.bialystok.pl/~gkret/SSN/SSN_w10.PDF>

<https://pl.wikipedia.org/wiki/Sie%C4%87_Kohonena>

<http://galaxy.agh.edu.pl/~vlsi/AI/koho_t/>