**Urszula Król,**

**grupa projektowa 2**

**IS WIMiIP**

**Sprawozdanie nr 5**

**1. Temat ćwiczenia:**

Budowa i działanie sieci Kohonena dla WTA.

**2. Cel ćwiczenia:**

Celem ćwiczenia jest poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły

WTA do odwzorowywania istotnych cech kwiatów.

**3. Zadania wykonane w ramach ćwiczenia:**

a) Przygotowanie danych uczących zawierających numeryczny opis cech kwiatów.

b) Przygotowanie (implementacja) sieci Kohonena i algorytmu uczenia opartego o regułę Winner Takes All (WTA).

c) Uczenie sieci dla różnych współczynników uczenia.

d) Testowanie sieci.

**4. ​Syntetyczny opis budowy oraz wykorzystanych sieci i algorytmów uczenia:**

Sieci Kohonena – są szczególnym przypadkiem algorytmu realizującego uczenie się bez nadzoru. Ich głównym zadaniem jest organizacja wielowymiarowej informacji w taki sposób, żeby można ją było prezentować i analizować w przestrzeni o znacznie mniejszej liczbie wymiarów, czyli mapie.

Zasady działania sieci Kohonena:

* Wejścia (ich liczba zależy od tego, iloma parametrami opisano obiekty) połączone są ze wszystkimi węzłami sieci
* Każdy węzeł przechowuje wektor wag o wymiarze identycznym z wektorami wejściowymi
* Każdy węzeł oblicza swój poziom aktywacji jako iloczyn skalarny wektora wag i wektora wejściowego (podobnie jak w zwykłym neuronie)
* Ten węzeł, który dla danego wektora wejściowego ma najwyższy poziom aktywacji, zostaje zwycięzcą i jest uaktywniony
* Wzmacniamy podobieństwo węzła-zwycięzcy do aktualnych danych wejściowych poprzez dodanie do wektora wag wektora wejściowego (z pewnym współczynnikiem uczenia)
* Każdy węzeł może być stowarzyszony z pewnymi innymi, sąsiednimi węzłami - wówczas te węzły również zostają zmodyfikowane, jednak w mniejszym stopniu.

Wagi w sieci Kohonena są inicjalizowane jest losowa. Wektory wejściowe stanowią próbę uczącą, podobnie jak w przypadku zwykłych sieci rozpatrywaną w pętli podczas budowy mapy. Wykorzystanie utworzonej w ten sposób mapy polega na tym, że zbiór obiektów umieszczamy na wejściu sieci i obserwujemy, które węzły sieci się uaktywniają. Obiekty podobne powinny trafiać w zbliżone miejsca na mapie.

Schemat:



Zasada WTA - tylko neuron zwycięski (oznaczany indeksem c, od conqueror) uaktualnia swe wagi, tzn. zbliża się do wektora x(k). Wagi zwycięskiego neuronu mogą zostać uaktualnione np. na podstawie wzoru Kohonena:



Współczynnik uczenia η(k) jest na ogół malejącą funkcją numeru iteracji k



Numer *k* wygrywającego neuronu ustala się na podstawie kryterium:



Zależność ta wykrywa największy spośród skalarnych iloczynów wektorów wag z wektorem wejściowym, czyli wskazuje wektor wag najbliższy wejściowemu.



Gdy *φ* = 0, cos(*φ* ) = 1 = max

*x*

*w*

φ

W myśl tej reguły następuje korekta wektora wag sygnałów zbiegających się do wejścia zwycięskiego neuronu *k* (linie pogrubione na rysunku)



Algorytm uczenia WTA:

1. Na wstępie przyjmuje się losowe, znormalizowane względem 1 wartości wag poszczególnych neuronów.
2. Po podaniu pierwszego wektora wejściowego **x** wyłaniany jest zwycięzca o numerze *k*.
3. Aktualizacja wag neuronu zwycięzcy (neurony przegrywające mają na wyjściu stan zero, co blokuje proces aktualizacji ich wag).

Aktualizacja wag według tzw. **reguły Kohonena** :



Wektor wag neuronu zwycięzcy jest zwiększany o ułamek różnicy *x-w*, w wyniku czego w następnych krokach lepiej odtwarza rozpatrywany wektor wejściowy (α > 0 jest dodatnim współczynnikiem, malejącym w miarę postępu nauki)

Przed rozpoczęciem uczenia wektory wag są normalizowane:



Do korekcji kwalifikowany jest wektor taki, że:



Korekcja wektora ma na celu zbliżenie go do typowego reprezentanta grupy *m*.

Sygnał wyjściowy *ai* *i-tego* neuronu, można opisać zależnością wektorową:



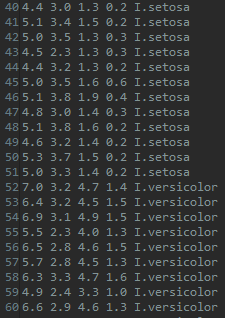
Ponieważ ||**w**|| = ||**x**|| = 1, o wartości sygnału wyjściowego *ai* decyduje różnica kątowa między wektorami **x** oraz **w**,   
bo *ai = cos(φi)*.

Zwycięża zatem ten neuron, którego wektor wag jest najbliższy aktualnemu wektorowi uczącemu **x**. Neurony przegrywające nie zmieniają swoich wag, dopóki nie zwyciężą przy następnej prezentacji wektora wejściowego.

Przy podaniu na wejście sieci wielu wektorów zbliżonych do siebie, zwyciężać będzie ciągle ten sam neuron, w wyniku czego jego wagi odpowiadać będą uśrednionym wartościom wektorów wejściowych, dla których dany neuron był zwycięzcą. W wyniku takiego współzawodnictwa następuje samoorganizacja się sieci. Neurony dopasowują swoje wagi w ten sposób, że przy prezentacji grup wektorów wejściowych zbliżonych do siebie zwycięża zawsze ten sam neuron.

Format danych

Fragment zestawu danych uczących:



Zestaw uczący składa się ze 150 linijek danych zawierających numeryczny opis cech 3 gatunków irysów: setosa, versicolor i virginica. Zestaw testowy składa się z 60 linijek danych, po 20 z każdego gatunku.

**5. ​Zestawienie otrzymanych wyników i obserwacje:**

Tabela 1 – Uzyskane wyniki w próbach różniących się wartością współczynnika uczenia:

|  |  |  |  |
| --- | --- | --- | --- |
| Lp. | Współczynnik uczenia | Liczba błędów | Liczba powstałych grup |
| 1 | 0.01 | 20 | 2 |
| 2 | 0.05 | 20 | 2 |
| 3 | 0.1 | 0 | 3 |
| 4 | 0.2 | 11 | 3 |
| 5 | 0.3 | 1 | 3 |
| 6 | 0.4 | 3 | 3 |
| 7 | 0.5 | 21 | 3 |
| 8 | 0.6 | 3 | 3 |
| 9 | 0.7 | 4 | 3 |
| 10 | 0.8 | 6 | 4 |
| 11 | 0.9 | 27 | 8 |
| 12 | 1.0 | 45 | 16 |

Wykres 1 – Wpływ wartości współczynnika uczenia na liczbę błędów:

Wykres 2 – Liczba powstałych grup w zależności od współczynnika uczenia:

Tabela 2 – Zależność liczby powstałych grup od liczby iteracji dla współczynnika uczenia 0.1:

|  |  |  |
| --- | --- | --- |
| Lp. | Liczba iteracji | Liczba powstałych grup |
| 1 | 10 | 2 |
| 2 | 100 | 3 |
| 3 | 1000 | 2 |
| 4 | 10000 | 2 |
| 5 | 100000 | 2 |

Dla współczynnika uczenia równego 0.1 otrzymano bezbłędne wyniki, powstały 3 grupy podziału i wszystkie gatunki zostały poprawnie rozpoznane.

Dla współczynników mniejszych od 0.1 zaobserwowano, że powstawały tylko dwie grupy: gatunki versicolor i virginica zostały potraktowane jako jeden gatunek.

Dla współczynników o wartościach od 0.2 do 0.7 otrzymywano podział na 3 grupy, jednak pojawiały się liczne błędy – największe przy współczynniku uczenia równym 0.5.

Dla współczynników większych od 0.7 liczba grup podziału wzrastała aż do 16 grup przy współczynniku równym 1.0, zaobserwowano także znaczny wzrost liczby błędów.

Przy liczbie iteracji równej 100 otrzymywano najbardziej optymalne wyniki.

**6. Przykłady działania programu:**

Learning rate: 0.1

Correct learning!

Testing...

1. Type: I.setosa Result: 7

2. Type: I.setosa Result: 7

3. Type: I.setosa Result: 7

4. Type: I.setosa Result: 7

5. Type: I.setosa Result: 7

6. Type: I.setosa Result: 7

7. Type: I.setosa Result: 7

8. Type: I.setosa Result: 7

9. Type: I.setosa Result: 7

10. Type: I.setosa Result: 7

11. Type: I.setosa Result: 7

12. Type: I.setosa Result: 7

13. Type: I.setosa Result: 7

14. Type: I.setosa Result: 7

15. Type: I.setosa Result: 7

16. Type: I.setosa Result: 7

17. Type: I.setosa Result: 7

18. Type: I.setosa Result: 7

19. Type: I.setosa Result: 7

20. Type: I.setosa Result: 7

21. Type: I.versicolor Result: 29

22. Type: I.versicolor Result: 29

23. Type: I.versicolor Result: 29

24. Type: I.versicolor Result: 29

25. Type: I.versicolor Result: 29

26. Type: I.versicolor Result: 29

27. Type: I.versicolor Result: 29

28. Type: I.versicolor Result: 29

29. Type: I.versicolor Result: 29

30. Type: I.versicolor Result: 29

31. Type: I.versicolor Result: 29

32. Type: I.versicolor Result: 29

33. Type: I.versicolor Result: 29

34. Type: I.versicolor Result: 29

35. Type: I.versicolor Result: 29

36. Type: I.versicolor Result: 29

37. Type: I.versicolor Result: 29

38. Type: I.versicolor Result: 29

39. Type: I.versicolor Result: 29

40. Type: I.versicolor Result: 29

41. Type: I.virginica Result: 16

42. Type: I.virginica Result: 16

43. Type: I.virginica Result: 16

44. Type: I.virginica Result: 16

45. Type: I.virginica Result: 16

46. Type: I.virginica Result: 16

47. Type: I.virginica Result: 16

48. Type: I.virginica Result: 16

49. Type: I.virginica Result: 16

50. Type: I.virginica Result: 16

51. Type: I.virginica Result: 16

52. Type: I.virginica Result: 16

53. Type: I.virginica Result: 16

54. Type: I.virginica Result: 16

55. Type: I.virginica Result: 16

56. Type: I.virginica Result: 16

57. Type: I.virginica Result: 16

58. Type: I.virginica Result: 16

59. Type: I.virginica Result: 16

60. Type: I.virginica Result: 16

Winners:

7

29

16

Learning rate: 0.01

Correct learning!

Testing...

1. Type: I.setosa Result: 26

2. Type: I.setosa Result: 26

3. Type: I.setosa Result: 26

4. Type: I.setosa Result: 26

5. Type: I.setosa Result: 26

6. Type: I.setosa Result: 26

7. Type: I.setosa Result: 26

8. Type: I.setosa Result: 26

9. Type: I.setosa Result: 26

10. Type: I.setosa Result: 26

11. Type: I.setosa Result: 26

12. Type: I.setosa Result: 26

13. Type: I.setosa Result: 26

14. Type: I.setosa Result: 26

15. Type: I.setosa Result: 26

16. Type: I.setosa Result: 26

17. Type: I.setosa Result: 26

18. Type: I.setosa Result: 26

19. Type: I.setosa Result: 26

20. Type: I.setosa Result: 26

21. Type: I.versicolor Result: 12

22. Type: I.versicolor Result: 12

23. Type: I.versicolor Result: 12

24. Type: I.versicolor Result: 12

25. Type: I.versicolor Result: 12

26. Type: I.versicolor Result: 12

27. Type: I.versicolor Result: 12

28. Type: I.versicolor Result: 12

29. Type: I.versicolor Result: 12

30. Type: I.versicolor Result: 12

31. Type: I.versicolor Result: 12

32. Type: I.versicolor Result: 12

33. Type: I.versicolor Result: 12

34. Type: I.versicolor Result: 12

35. Type: I.versicolor Result: 12

36. Type: I.versicolor Result: 12

37. Type: I.versicolor Result: 12

38. Type: I.versicolor Result: 12

39. Type: I.versicolor Result: 12

40. Type: I.versicolor Result: 12

41. Type: I.virginica Result: 12

42. Type: I.virginica Result: 12

43. Type: I.virginica Result: 12

44. Type: I.virginica Result: 12

45. Type: I.virginica Result: 12

46. Type: I.virginica Result: 12

47. Type: I.virginica Result: 12

48. Type: I.virginica Result: 12

49. Type: I.virginica Result: 12

50. Type: I.virginica Result: 12

51. Type: I.virginica Result: 12

52. Type: I.virginica Result: 12

53. Type: I.virginica Result: 12

54. Type: I.virginica Result: 12

55. Type: I.virginica Result: 12

56. Type: I.virginica Result: 12

57. Type: I.virginica Result: 12

58. Type: I.virginica Result: 12

59. Type: I.virginica Result: 12

60. Type: I.virginica Result: 12

Winners:

26

12

**7. Wnioski:**

- Uczenie sieci Kohonena wymagają dobrania właściwego współczynnika uczenia oraz odpowiedniej liczby iteracji.

- Główny wpływ na uczenie ma współczynnik uczenia: jeżeli jest zbyt mały, występuje problem w odróżnieniu dwóch gatunków od siebie, natomiast jeśli jest za duży, gatunki są błędnie rozpoznawane i tworzy się znacznie więcej grup.

- Im większy współczynnik uczenia tym więcej tworzy się grup podziału.

- Liczba iteracji ma wpływ na uczenie sieci, zarówno zbyt mała jak i zbyt duża ilość iteracji powoduje złe rozpoznanie grup.

- Analizując uzyskane wyniki można stwierdzić, iż sieć ma problem z odróżnieniem podobnych danych, tak jak w przypadku gatunków Virginica i Versicolor, które często były traktowane jako jedna grupa.

- Algorytm uczenia WTA wymaga normalizacji wektorów wag.

- W metodzie WTA zwycięża ten neuron, którego wektor wag jest najbliższy aktualnemu wektorowi uczącemu. Neurony przegrywające nie zmieniają swoich wag, dopóki nie zwyciężą przy następnej prezentacji wektora wejściowego.

**8. Listing kodu programu:**

Pełny listing kodu programu znajduje się w repozytorium: <https://github.com/ukrol/PSI_GP02_zima_2017-2018_Urszula_Krol>