Paweł Put, Informatyka Stosowana gr. 3

Sprawozdanie nr 4.

Temat ćwiczenia: Uczenie sieci regułą Hebba.

Cel ćwiczenia: Celem ćwiczenia jest poznanie działania reguły Hebba dla sieci jednowarstwowej na przykładzie grupowania liter alfabetu.

Przebieg: Zaimplementowałem uczenie regułą Hebba w javie i nauczyłem reagowanie sieci na kolejne dane wejściowe. Danymi wejściowymi było 20 dużych liter polskiego alfabetu w wektorach 35cio elementowych, które po podzieleniu wyglądały jak tablice 5x7. Przykład litery A:

* Wektor:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* Tablica utworzona z wektora:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* Interpretacja numeryczna wektora:

(-1, 1, 1, 1, -1, 1, -1, -1, -1, 1, 1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, 1, 1, -1, -1, -1, 1, 1, -1, -1, -1, 1)

* Interpretacja numeryczna tabeli:

(-1, 1, 1, 1, -1,

1, -1, -1, -1, 1,

1, -1, -1, -1, 1,

1, 1, 1, 1, 1,

1, -1, -1, -1, 1,

1, -1, -1, -1, 1,

1, -1, -1, -1, 1 )

Następnie utworzyłem wektor uczący 20sto elementowy, którego kolejne wyjścia miały odpowiadać kolejnym literom. Początkowo wypełniłem go samymi 1, aby nie wpływał na obliczanie wag początkowych:

(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 )

Potem zrobiłem 35 elementowy wektor wag i rozlosowałem wagi z zakresu (-0.05, 0.05).

Zostało mi tylko utworzenie sieci metodą Hebba. Jest to pierwsza sieć bez nauczyciela, z którą przyszło mi się zmierzyć. D. Hebb, badając działąnie komórek nerwowych zauważył, że powiązanie dwóch komórek jest wzmacniane jeśli obie komórki są pobudzone w tym samym czasie. Dzięki temu jeżeli będziemy działali na sieć konkretnymi sygnałami wejściowymi, to niektóre z połączeń będą się wzmacniać, niektóre osłabiać, a niektóre pozostaną neutralne. Spowoduje to rozpoznawanie przez naszą sieć kolejnych przykładów na zasadzie wzmacniania reakcji na nie. Aby uzyskać taki stan rzeczy, podawałem litery jako przykłady i za każdym przejściem epoki przestawałem uczyć jedną literę, czyli 20 literę nauczałem 20 razy, 19stą 19 razy, a pierwszą raz.

Obliczanie wartości wag bez współczynnika zapominania:

* weights[i+1] = weights[i] + learningrate \* letters \* output

Obliczanie wartości wag ze współczynnikiem zapominania:

* Weights[i+1] = weights[i] \* (1 - forgetrate) + learningrate \* letters\* output

Obliczanie wyjścia dla danej litery

* output = ∑ weights[i] \* letters[i]

Oznaczenia:

* weights[i] - wektor wag w aktualnym kroku
* weights[i + 1] - wektor wag w następnym kroku
* learningrate - współczynnik nauczania
* letters - wektor liter
* output - odpowiedni output literze
* forgetrate - współczynnik zapominania
* letters[i] - konkretny output dla danej wagi

Współczynnik nauczania i zapominania dobiera się z przedziału (0, 1).

Przetestujemy działanie programu dla różnego zestawu danych, a także dla danych z współczynnikami zapominania i bez nich:

1. Współczynniki nauczania:

* 0,01
* 0,1
* 0,5

1. Współczynniki zapominania:

* 0,001
* 0,01
* 0,1

Oznaczenia z tabeli:

1. Współczynnik nauczania
2. Współczynnik zapominania
3. Output litery U trenowanej 1 raz
4. Output litery K trenowanej 10 razy
5. Output litery A trenowanej 20 razy
6. Waga[0] na koniec uczenia
7. Waga[17] na koniec uczenia
8. Waga[33] na koniec uczenia

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 0,01 | - | 0, 9272 | -3.3329 | 5.0848 | -0.6117 | 0.4678 | 0.2365 |
| 0,01 | - | 0.7730 | -2.7786 | 4.2393 | -0.2594 | 0.3713 | 0.0435 |
| 0,1 | - | 0.9495 | -1316.37 | 25468.806 | -18341.85 | 1479.702 | -9438.952 |
| 0,1 | - | 0.7360 | -1020.357 | 19741.56 | -14216.42 | 1147.295 | -7316.26 |
| 0,5 | - | 0.0736 | -1006865 | 2.72\*10­9 | -1.06\*1010 | 1.81\*109 | -7.54\*109 |
| 0,5 | - | 0.3674 | -3494760 | 9.45\*109 | -8.97\*109 | 1.53\*109 | -6.36\*109 |
| 0,01 | 0,001 | 0.3743 | -1.1851 | 1.7334 | -0.271 | 0.421 | 0.515 |
| 0,01 | 0,001 | 0.7943 | -2.515 | 3.6791 | -0.2524 | 0.4137 | 0.3341 |
| 0,1 | 0,01 | 0.01809 | -9.921 | 141.377 | -102.93 | 8.360 | -52.88 |
| 0,1 | 0,01 | 0.3908 | -251.84 | 3588.86 | -2616.57 | 210.685 | -1346.465 |
| 0,5 | 0,05 | 0.1688 | -132466 | 1.36\*108 | -1.31\*107 | 2.27\*107 | -9.47\*107 |
| 0,5 | 0,05 | 0.3787 | -374576 | 3.86\*108 | -3.71\*108 | 6.44\*107 | -2.67\*108 |
| 0,1 | 0,001 | 0.1427 | -182.70 | 3428 | -2470 | 199.54 | -1271 |
| 0,1 | 0,001 | 0.80559 | -1033 | 19388 | -13980 | 1128 | -7194 |
| 0,1 | 0,01 | 0.01809 | -9.921 | 141.377 | -102.93 | 8.360 | -52.88 |
| 0,1 | 0,01 | 0.3908 | -251.84 | 3588.86 | -2616.57 | 210.685 | -1346.465 |
| 0,1 | 0,1 | 0.1151 | -0.23183719 | 0.231837109 | -0.18603 | 0.01310 | -0.09723 |
| 0,1 | 0,1 | 0.1359 | -0.2862848 | 0.2862846 | -0.22971 | 0.0161863 | -0.12006 |

Omówienie wyników:

* Wyniki wskazują, że im więcej razy podajemy sieci dane wejście, tym mocniej na nie reaguje. Różnica pomiędzy literą U podawaną jeden raz, a literą A jest kolosalna,
* Pomimo testowania sieci dla różnych współczynników uczenia i zapominania zauważalna jest pewna prawidłowość. Weights[0] osiągał na koniec zawsze silnie ujemną wartość, natomiast Weights[17] za każdym razem był skierowany w stronę dodatnią i około 10krotnie mniejszy niż Weights[0]. To pokazuje, że za każdym razem pobudzane były te same neurony w podobnym stopniu
* Przy zbyt dużym współczynniku zapominania sieć nie będzie nam się uczyła, co jest dość logiczne. Współczynnik zapominania powinien być około 10krotnie razy mniejszy niż wartość współczynnika nauczania.
* Im większy współczynnik nauczania tym silniej reagowały neurony i wyniki były bardziej wyraziste. Sieć uczyła się wtedy lepiej reagować na przykłady.

Wnioski:

* Hebb miał rację. Nie potrzeba nauczyciela, żeby sieć się uczyła. Reaguje ona na przykłady, które otrzymuje
* Uczenie Hebba jest bardzo intensywne, dlatego warto korygować je współczynnikiem zapominania, by nasze liczby nie były tak ogromne.

Bibliografia:

* Żurada J., Barski M., Jędruch W. - Sztuczne sieci neuronowe
* Stanisław Osowski - Sieci neuronowe do przetwarzania informacji
* http://home.agh.edu.pl/~asior/stud/doc/Kohonen\_12.pdf
* https://www.ii.uni.wroc.pl/~aba/teach/NN/w6pca.pdf
* http://www.poltynk.pl/marcin/reguly.html
* http://sknbo.ue.poznan.pl/neuro/ssn/pliki/samoucz/samoucz1.html
* http://aragorn.pb.bialystok.pl/~gkret/SSN/Ssn\_w2.PDF
* http://galaxy.agh.edu.pl/~vlsi/AI/hopf/hopfield\_pl.html