

Podstawy Sztucznej Inteligencji

Przemysław Bartkowski Gr 1 Inżynieria Obliczeniowa

Uczenie sieci regułą Hebba

Cel ćwiczeń:

Celem ćwiczenia było poznanie działania reguły Hebba na przykładzie rozpoznawania emotikon.

Zadania do realizacji:

- a) Wygenerowanie danych uczących i testujących, zawierających 4 różne emotikony np. czarno-białe, wymiar 8x8 pikseli dla jednej emotikony.
- b) Przygotowanie (implementacja lub wykorzystanie gotowych narzędzi) sieci oraz reguły Hebba w wersji z i bez współczynnika zapominania.
- c) Uczenie sieci dla różnych współczynników uczenia i zapominania.
- d) Testowanie sieci.

Zasada Hebba i siła powiązań między neuronami:

W neuro-fizjologii zauważono, że siła (waga) powiązań między dwoma neuronami wzrasta przy jednoczesnym pobudzeniu obu neuronów, w przeciwnym przypadku maleje. Zaobserwowano również, że im częściej jakiś bodziec dochodzi do neuronu, tym silniejsza jest odpowiedź neuronu na bodziec. W konsekwencji najczęściej powtarzające się mają największy wpływ na samo-adaptację wag. W sieciach neuronowych neurony 'uczą się' rozpoznawać bodźce (w naszym rozumieniu dane podawane na wejście sieci), czyli dane. Zmiany aktualnych wag powinny iść w kierunku dostosowywania wag do najczęściej przedstawianych wzorców, jakimi są wektory danych. Sygnały napływające do neuronu noszą również nazwę sygnałów presynaptycznych. Sygnały wytwarzane przez neuron to sygnały postsynaptyczne.

Podczas procesu uczenia, w miarę napływających wzorców (wektorów danych w naszym zrozumieniu) sieć 'uczy się' tych danych, czyli adaptuje sekwencyjnie swoje wagi aby umieć prawidłowo rozpoznawać te dane. Adaptacja ta iteracyjna, tzn po przedstawieniu k – tego wzorca $x(k)$ wytworzone już wagi w zamieniają ogólnej zasady:

$$w(k + 1) = w(k) + \Delta w(k).$$

Jak wyrazić zmianę wektora wag, czyli wielkość $\Delta w(k)$? Zagadnienie to było przedmiotem intensywnych dyskusji, w rezultacie których pojawiło się kilka propozycji, które przetrwały do dzisiaj, a nawet są dalej rozwijane. W latach 1949 ukazała się książka Hebba dyskutująca to zagadnienie. Z książki tej pochodzą dwie zasady, znane dzisiaj pod nazwą ogólnej i prostej reguły Hebba. Ogólna reguła Hebba mówi, że zmiany wag powinny odbywać się według reguły:

$$\Delta w(k) = F(x(k), y(k)),$$

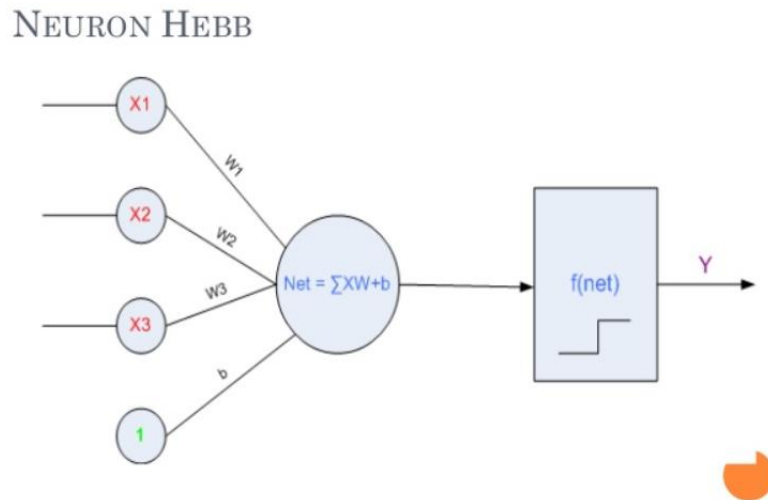
$\Delta w(k)$ – przyrost wag.

$x(k)$ – wielkość wzorca presynaptycznego.

$y(k)$ – wielkość wzorca postsynaptycznego.

F -funkcja wyrażana poprzez prostą regułę Hebba : $\Delta w(k) = \eta \cdot x(k) \cdot y(k)$ (stała η oznacza tu współczynnik proporcjonalności)

Model neuronu Hebba:



Listing kodu:

%wejścia do sieci z min i max wartościami

```
minmax=[0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
 0 1; 0 1; 0 1; 0 1];
```

%ilość wyjść z sieci

```
ilosc_wyj = 64;
```

%użycie funkcji tworzącej sieć

```
net = newff(minmax, ilosc_wyj,{'tansig'}, 'trainlm', 'learnh');
```

%kolumnowe wprowadzenie emotikon w formie 0-1

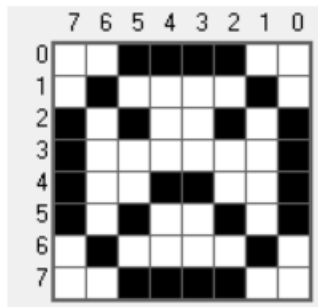
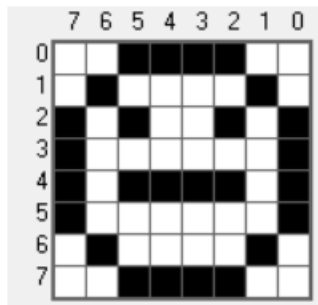
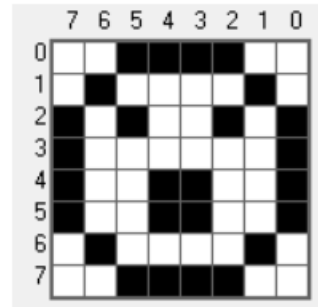
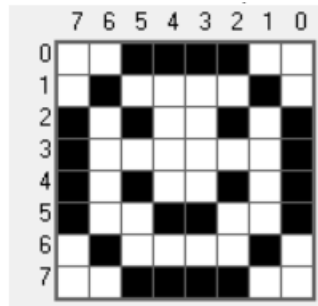
% smile/shock/confuse/sad

```
input = [ 0 0 0 0;
 0 0 0 0;
 1 1 1 1;
 1 1 1 1;
 1 1 1 1;
 1 1 1 1;
 1 1 1 1;
 0 0 0 0;
 0 0 0 0;
 0 0 0 0;
 1 1 1 1;
 0 0 0 0;
 0 0 0 0;
 0 0 0 0;
 0 0 0 0;
 0 0 0 0;
 1 1 1 1;
 0 0 0 0;]
```

```

1 1 1 1;
0 0 0 0;
1 1 1 1;
0 0 0 0;
0 0 0 0;
1 1 1 1;
0 0 0 0;
1 1 1 1;
1 1 1 1;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
1 1 1 1;
1 1 1 1;
0 0 0 0;
1 0 1 0;
0 1 1 1;
0 1 1 1;
1 0 1 0;
0 0 0 0;
1 1 1 1;
1 1 1 1;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
1 1 1 1;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
1 1 1 1;
0 0 0 0;
0 0 0 0;
0 0 0 0;
0 0 0 0;
1 1 1 1;
1 1 1 1;
1 1 1 1;
1 1 1 1;
0 0 0 0;
0 0 0 0;
];

```



0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1
1	0	1	0	0	1	0	1
1	0	0	1	1	0	0	1
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1
1	0	0	1	1	0	0	1
1	0	0	1	1	0	0	1
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1
1	0	1	1	1	1	0	1
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1
1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	1
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

%zmienna zawierająca 1 gdy trafimy w emotikon i 0 gdy chybimy

```

output = [ 1 0 0 0 %smile
           0 1 0 0 %shock
           0 0 1 0 %confuse
           0 0 0 1]; %sad

```

%parametry reguły Hebba

```
lp.dr = 0.5; %wsp. zapominania  
lp.lr = 0.9; %wsp. uczenia
```

%użycie reguły Hebba

```
hebb = learnh( [], input, [], [], output, [], [], [], [], lp, []);  
heb=hebb';
```

```
net.trainParam.epochs = 1000;  
net.trainParam.goal = 0.01;
```

%trenowanie sieci z użyciem reguły Hebba

```
net = train(net, input, heb);
```

%DANE TESTUJACE

```
smile = [ 0 0 1 1 1 1 0 0; 0 1 0 0 0 0 1 0; 1 0 1 0 0 1 0 1; 1 0 0 0 0 0 0 1; 1 0 1 0 0 1 0 1; 1 0 0 1 1 0 0 1; 0 1 0 0 0 0 1  
0; 0 0 1 1 1 1 0 0];  
shock = [ 0 0 1 1 1 1 0 0; 0 1 0 0 0 0 1 0; 1 0 1 0 0 1 0 1; 1 0 0 0 0 0 0 1; 1 0 0 1 1 0 0 1; 1 0 0 1 1 0 0 1; 0 1 0 0 0  
1 0; 0 0 1 1 1 1 0 0];  
confuse = [ 0 0 1 1 1 1 0 0; 0 1 0 0 0 0 1 0; 1 0 1 0 0 1 0 1; 1 0 0 0 0 0 0 1; 1 0 1 1 1 1 0 1; 1 0 0 1 1 0 0 1; 0 1 0 0 0  
0 1 0; 0 0 1 1 1 1 0 0];  
sad = [ 0 0 1 1 1 1 0 0; 0 1 0 0 0 0 1 0; 1 0 1 0 0 1 0 1; 1 0 0 0 0 0 0 1; 1 0 0 1 1 0 0 1; 1 0 1 0 0 1 0 1; 0 1 0 0 0 0 1  
0; 0 0 1 1 1 1 0 0];
```

%sprawdzenie poprawności wytrenowanej sieci

```
test = sim(net, smile);  
test1 = sim(net, shock);  
test2 = sim(net, confuse);  
test3 = sim(net, sad);
```

%wypisanie wartości

```
disp('SMILE ='), disp(test(1));  
disp('SHOCK ='), disp(test1(1));  
disp('CONFUSE ='), disp(test2(1));  
disp('SAD ='), disp(test3(1));
```

Dane zaprezentowane na poglądowym obrazku zostały wprowadzone jako dane kolumnowe dla zmiennej input, oraz te same dane, ale w konfiguracji wierszowej zostały wprowadzone do danych testowych.

Poniżej wyniki uzyskane dla każdej emotikony. Przyjąłem współczynnik uczenia równy: 0.01 oraz współczynnik zapominania 0.01.

SMILE:

Próba	0.01/0.01
1	-0.7096
2	1
3	0.4374
4	-1
5	-0.8830

SHOCK:

Próba	0.01/0.01
1	-0.1986
2	1
3	0.1491
4	-1
5	-0.2091

CONFUSE:

Próba	0.01/0.01
1	-0.5272
2	1
3	0.6064
4	-1
5	-0.6268

Wnioski:

- Zgodnie z regułą Hebba wagi mają duży wpływ na wyniki uczenia sieci.
- Metoda Hebba opiera się o uczenie bez nauczyciela w wyniku czego sieć jest zmuszona zdecydować o tym jakie będą skutki działania sieci dla różnych danych wejściowych.
- Sieć sama musi wyciągać wnioski na podstawie posiadanych informacji co nie zawsze skutkuje oczekiwanym rezultatem.
- Sieć dla poprawnego działania potrzebuje znacznie więcej czasu niż taka sama sieć uczona z nauczycielem. Jednak zbyt powolne uczenie sieci może prowadzić do błędów.