

Małgorzata Bień

Sprawozdanie do scenariusza nr 4

Temat ćwiczenia: Uczenie sieci regułą Hebba.

1. Cel ćwiczenia: Poznanie działania reguły Hebba na przykładzie rozpoznawania emotikon.
2. Opis syntetyczny budowy sieci oraz algorytmu uczenia
  - Model neuronu Hebba

Wykorzystuje regułę matematyczną Hebba, w której zmiana wagi  $w_{ij}$  neuronu odbywa się proporcjonalnie do iloczynu sygnału wejściowego oraz wyjściowego :

$$\Delta w_{ij} = \eta y_j y_i$$

Gdzie  $\eta$  to stała uczenia z przedziału  $(0,1)$  ,  $y_j$  sygnał wejściowy a  $y_i$  sygnał wyjściowy.

Hebb zauważył podczas badań działania komórek nerwowych, iż połączenie pomiędzy dwiema komórkami jest wzmacniane, jeżeli w tym samym czasie obie komórki są aktywne.

Reguła Hebba może być stosowana do różnego typu struktur sieci neuronowych i różnych funkcji aktywacji zastosowanych w modelu neuronu. Model ten ma identyczną strukturę jak model typu Adaline oraz neuronu sigmoidalnego, ale charakteryzuje się specyficzną metodą uczenia.

Uczenie neuronu z zastosowaniem reguły Hebba może odbywać się w trybie z bez nauczyciela lub z nauczycielem. W pierwszym przypadku w regule Hebba używa się aktualnej wartości  $y_i$  sygnału wyjściowego neuronu.

W uczeniu z nauczycielem wartość sygnału wyjściowego  $y_i$  zastępuje się wartością zadaną  $d_i$  dla danego neuronu. Regułę Hebba można wówczas zapisać w postaci:

$$\Delta w_{ij} = \eta y_j d_i$$

Uaktualnianie wag w regule Hebba:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}$$

Wagi mogą przyjąć dowolnie duże wartości, gdyż w każdym cyklu uczącym następuje proces sumowania aktualnych wartości wag i skończonego przyrostu  $\Delta w_{ij}$ .

Poprawa stabilności procesu uczenia:

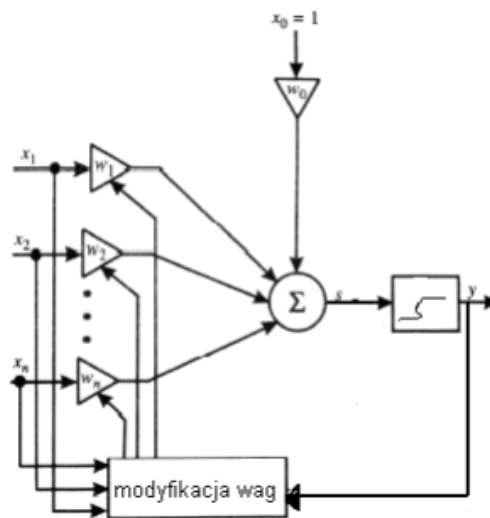
$$w_{ij}(k+1) = (1 - \gamma)w_{ij}(k) + \Delta w_{ij}$$

Przyjęcie przy aktualizacji wag nie ostatniej wartości  $w_{ij}$ , ale wartości tej zmniejszonej o współczynnik zapominania  $\gamma$ . Zawiera on się zwykle w przedziale  $(0,1)$  i stanowi najczęściej

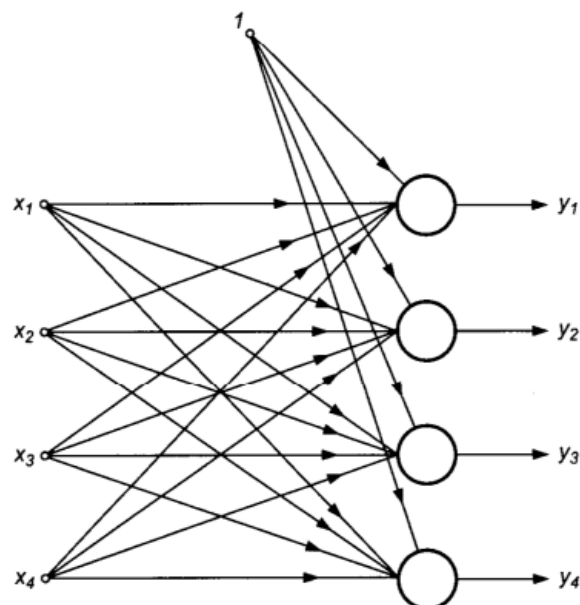
niewielki procent stałej uczenia  $\eta$ . Przyjęcie dużej wartości  $\gamma$  powoduje, że neuron zapomina większość tego co zdołał się nauczyć. Typowa wartość współczynnika zapominania to  $\gamma < 0.1$ , przy której neuron zachowuje większość informacji zgromadzonej podczas procesu uczenia.

W projekcie wykorzystałam sigmoidalną funkcję aktywacji. Sieć składała się z warstwy wejściowej, która posiada 100 danych wejściowych oraz 6 danych wyjściowych.

## I. Model neuronu

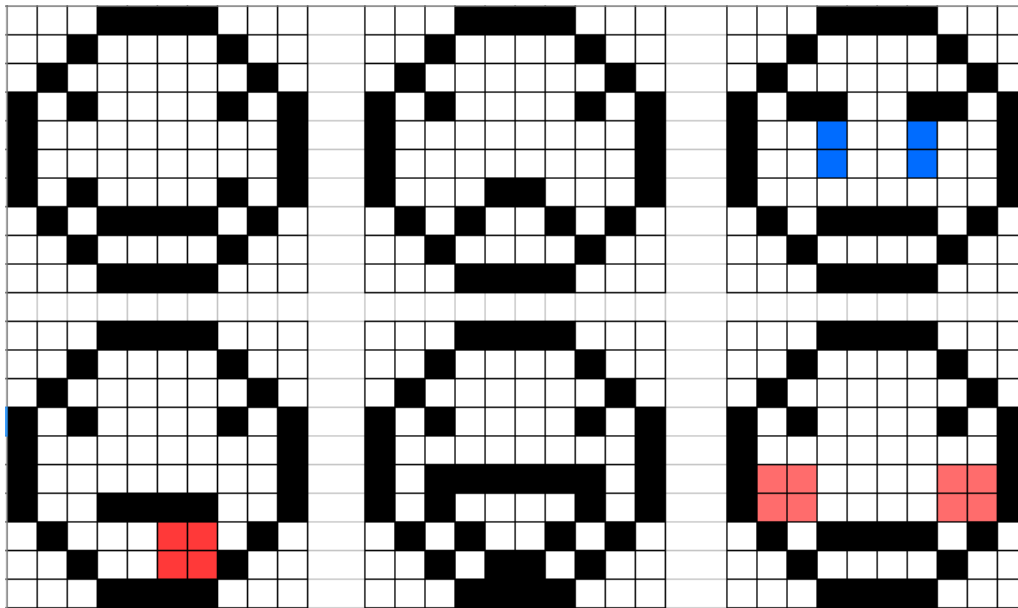


## II. Schemat sieci Hebba



### 3. Dane uczące:

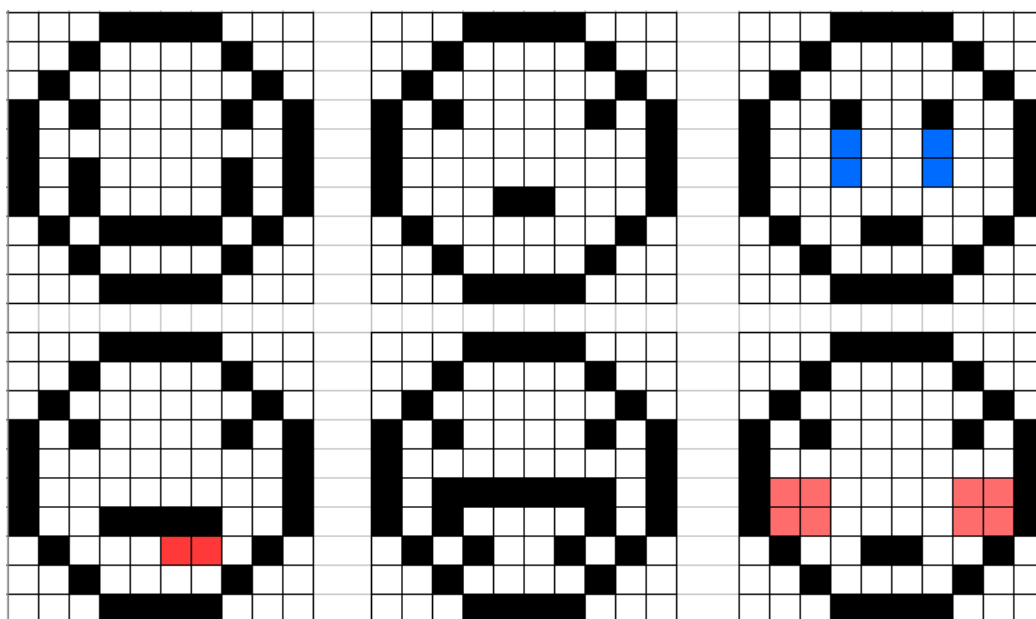
Dane uczące to 6 emotikon o wymiarze 10x10 pikseli. Do przetworzenia w sieci zostały zapisane jako 100 cyfr (0 lub 1) które są danymi wejściowymi plus 6 cyfr – prawidłowe dane wyjściowe (pięć 0 i jedna 1 określająca daną emotikonę).



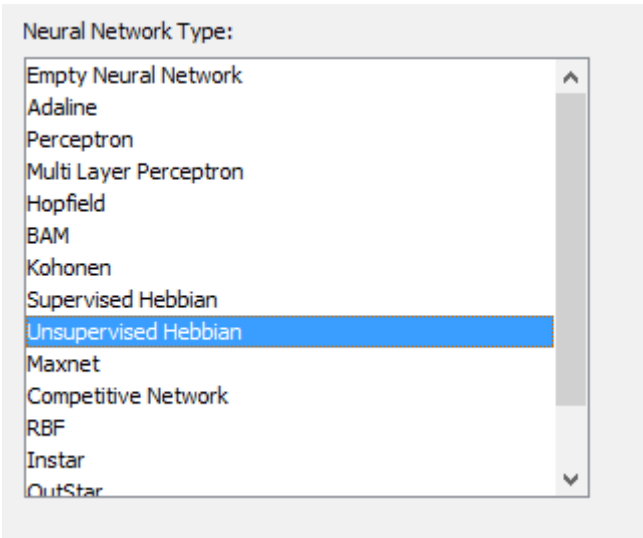
```
0001111000001000010001000000101010000101100000000110000000011010000101010111101000100001000001111000100000//smile
0001111000001000010001000000101010000101100000000110000000011000110001010100101000100001000001111000010000//sad
00011110000010000100010000001010110011001100100110010010011000000001010111101000100001000001111000001000//cry
000111100000100001000100000010101000010110000000011000000001100111100101000101000100111000001111000000100//tongue
000111100000100001000100000010101000010110000000011011111101101000010101010010100010110100000111100000010//veryhappy
0001111000001000010001000000101010000101100000000111100001111110000111010111101000100001000001111000000001//love
```

### Dane testujące:

W zestawie danych testujących zmieniłam część pikseli by sprawdzić w jakim stopniu sieć rozpoznaje daną emotikonę.



4. Konfiguracja programu:

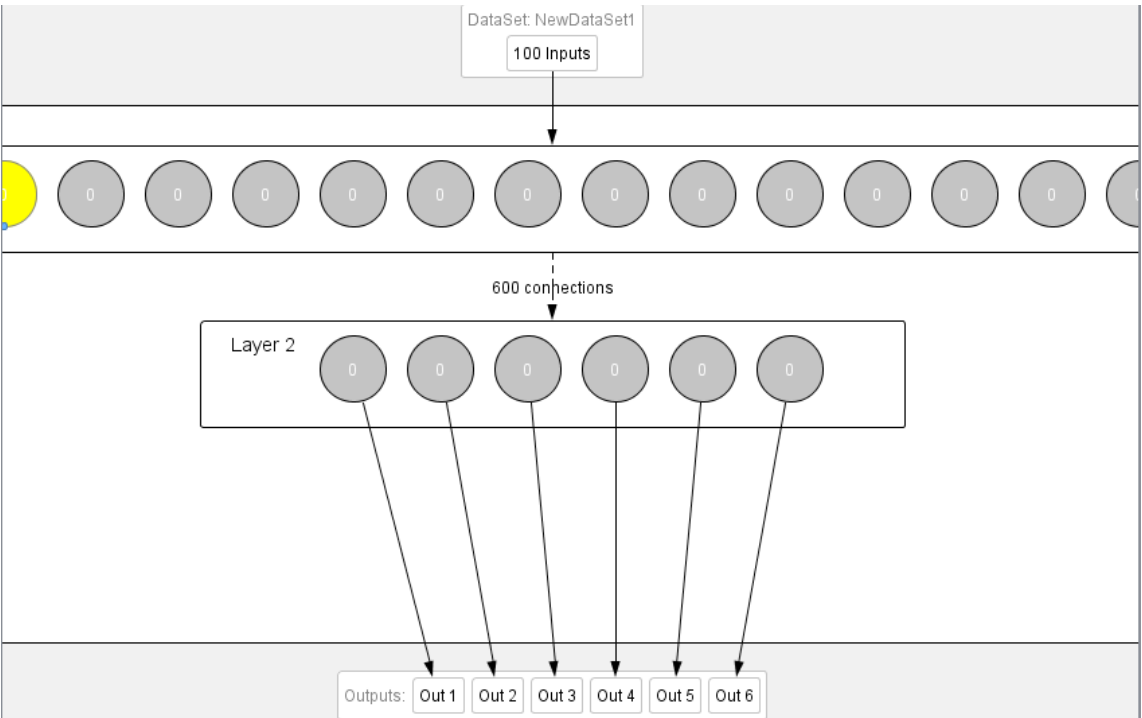


**Number of input neurons, output neurons and transfer function**

Inputs num

Outputs num

Transfer function

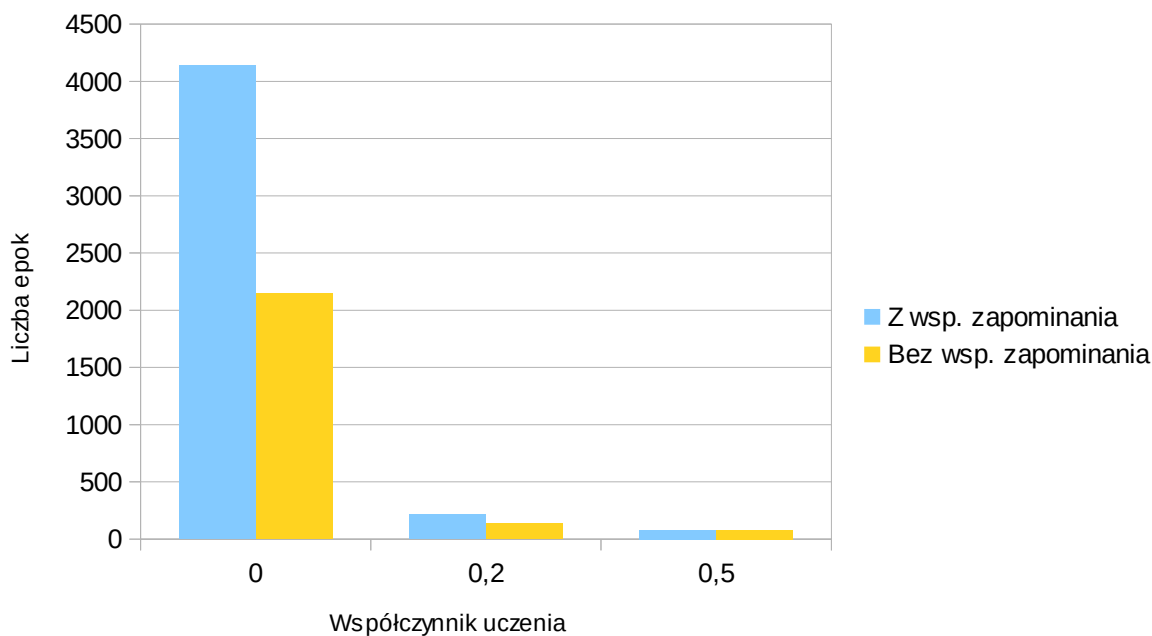


## 5. Wyniki

Wyniki w załączniku w arkuszu kalkulacyjnym.

## 6. Analiza

Wykres zależności w liczbie epok a współczynnikiem uczenia oraz uczeniem ze współczynnikiem zapominania i bez:



Learning Rate	Forgetting Rate	Epochs	Error
0,001	0,003	4140	0,003325555609134
0,2	0,03	214	0,003174370750372
0,5	0,1	82	0,003033256979368

Learning Rate	Epochs	Error
0,001	2147	0,003316422665173
0,2	135	0,003033256979368
0,5	79	0,00289928693552

Sieć uczyłam z wykorzystaniem trzech współczynników uczenia i zapominania. Te dwa współczynniki uzależniłam od siebie. Jeśli wartość Learning Rate była mała również Forgetting Rate był odpowiednio mały, zachowując założenie, że jego typowa wartość jest mniejsza niż 0,1. Najwolniej proces uczenia następował przy najmniejszym Learning Rate i były to 4140 epoki, uwzględniając współczynnik zapominania. Bez niego liczba epok zmalała niemal o połowę. We wszystkich przypadkach otrzymałam prawie identyczny całkowity błąd średniokwadratowy. Minimalnie mniejsze wartości uzyskałam przy uczeniu bez współczynnika zapominania.

Podczas uczenia zawsze otrzymywałam poprawne wyniki, tj. poprawne wskazanie emotikony. Do testowania użyłam zestawu testującego, w którym część pikseli była inna, zmodyfikowany był wygląd emotikony. W każdym teście sieć wykazywała 100% poprawność, minimalnie wzrósł błąd. Błędne wskazania podczas testów otrzymałam dopiero po większej modyfikacji emotikon.

Błędne wskazanie dotyczyło trzech emotikon 1,4 i 6. Emotikona testowa nr 1 rozpoznana została jako emotikona nr 5, emotikona nr 4 jako nr 2 oraz emotikona nr 6 jako nr 1. Przy teście przeprowadzonym na bardziej zmodyfikowanym zestawie testowym całkowity błąd wyniósł około 0,06 co jest wartością niemal dwa razy większą od wartości błędów otrzymanych przy uczeniu i pierwszym testowaniu.

## 7. Wnioski

Uczenie sieci neuronowej przebiegało poprawnie. Sieć prawidłowo rozpoznaje emotikony, wynik ich rozpoznania dąży do 1, niezależnie od współczynnika uczenia i zapominania. Od współczynników zależy prędkość uczenia sieci. Z wykresu zależności liczby epok można wnioskować że uczenie przebiegało szybciej bez współczynnika zapominania oraz wraz ze wzrostem współczynnika uczenia. Otrzymałam też wtedy minimalnie mniejszy błąd średniokwadratowy. Testowanie sieci na niewiele różniących się od zestawu uczącego danych testujących wskazywało poprawność nauczania sieci. Przeprowadzenie testów na bardziej zmodyfikowanym zestawie testującym pokazało błędy sieci w rozpoznawaniu zadanych emotikon. Nie wskazuje to jednak na jej nieprawidłowe działanie lecz na umiejętność rozpoznania różnych cech nauczonych emotikon. Sieć wskazywała błędne rozpoznanie dla emotikon których wygląd był zbliżony do innych. Przykładowo emotikona nr 6 z zestawu testującego przypomina emotikonę nr 1 z zestawu uczącego, dlatego nie można stwierdzić, że sieć podała niepoprawny wynik.