# Dominik Kędzior

## Inżynieria Obliczeniowa gr.1

### Nr 293094

Sprawozdanie 4

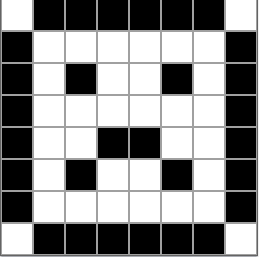
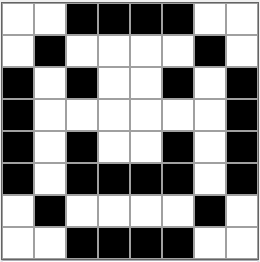
**Uczenie sieci metoda Hobba**

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

Do wykonania zadania wykorzystałem jednowarstwową sieć neuronową składająca się z 64 neuronów. Następnie została ona nauczona rozpoznawania czterech emotikon, które były zapisane kolumnowo w postaci binarnej. Do ich reprezentacji użyłem tablicy 8x8.

Wykonałem zadanie z pomocą programu Matlab oraz biblioteki Neural Network Toolbox (dzięki tej bibliotece wygenerowałem wcześniej wspomnianą sieć, która później była przeze mnie testowana).

Umieszczanie emotikon w tablicy polegało na zinterpretowaniu danego pola przez sieć. Tablica składa się z 64 pól, które odpowiednio są czarne - 1, lub białe - 0. Czarne pole oznaczało, że w danym miejscu występował fragment emoticony.



**Reguła Hebba**

Zasada działania reguły Hebba polega na interpretacji zachowań aktywnych neuronów. Jeśli aktywny neuron A jest cyklicznie pobudzany przez drugi neuron B, to staje się on jeszcze bardziej czuły na pobudzenie tego neuronu. Reguła Hebba składa się z następujących punktów:

1. Jeżeli połączone synapsą neurony A i B są pobudzane jednocześnie (synchronicznie) to połączenie synaptyczne między nimi jest wzmacniane.

2. Jeżeli neurony A i B połączone synapsą nie są pobudzane jednocześnie (asynchronicznie) to połączenie pomiędzy nimi jest osłabiane.

Reguła Hebba jest jedną z najpopularniejszych metod samouczenia sieci neuronowych. Polega ona na tym, że sieci pokazuje się kolejne przykłady sygnałów wejściowych nie podając żadnych informacji o tym, co z tymi sygnałami należy zrobić. Sieć obserwuje otoczenie i odbiera różne sygnały nie zostają określone jednak jakie znaczenie mają pokazujące się obiekty i jakie są pomiędzy nimi zależności.

2

Sieć na podstawie obserwacji występujących sygnałów stopniowo sama odkrywa, jakie jest ich znaczenie i również sama ustala zachodzące między sygnałami zależności. Po podaniu do sieci neuronowej każdego kolejnego zestawu sygnałów wejściowych tworzy się w tej sieci pewien rozkład sygnałów wyjściowych - niektóre neurony sieci są pobudzone bardzo silnie inne słabiej, a jeszcze inne mają sygnały wyjściowe wręcz ujemne.

Proces samouczenia ma niestety wady. W porównaniu z procesem uczenia z nauczycielem samouczenie jest zwykle znacznie powolniejsze. Co więcej bez nauczyciela nie można z góry określić, który neuron wyspecjalizuje się w rozpoznawania której klasy sygnałów. Stanowi to pewną trudność przy wykorzystywaniu i interpretacji wyników pracy sieci. Co więcej - nie można określić, czy sieć uczona w ten sposób nauczy się wszystkich prezentowanych jej wzorców. Dlatego sieć przeznaczona do samouczenia musi być większa niż sieć wykonująca to samo zadanie, ale trenowana w sposób klasyczny z udziałem nauczyciela.

Metoda Hebba posiada wiele wad, są nimi m. in.:

- niska efektywność uczenia,

- przemnożony wpływ początkowych wartości wag sieci,

- wagi w przypadku tej reguły mogą rosnąć bez ograniczeń,

- proces uczenia sieci nigdy nie zakończy się samodzielnie,

- nie ma pewności, że jednej klasie wzorców będzie odpowiadał jeden neuron,

- nie ma również gwarancji, że wszystkie klasy wzorców będą miały oddzielne reprezentacje w postaci oddzielnych zbiorów aktywnych neuronów.

Bardzo istotną kwestią jest wybór początkowych wartości wag neuronów sieci przeznaczonej do samouczenia. Wartości te mają bardzo silny wpływ na ostateczne zachowanie sieci, ponieważ proces uczenia jedynie pogłębia i doskonali pewne tendencje istniejące w sieci od samego początku. Od jakości początkowych właściwości sieci silnie zależy do czego sieć dojdzie na końcu procesu uczenia.

**Opis wykorzystanych metod oraz zmiennych**

Wykorzystanie funkcji newff(start, output\_s, {'tansig'}, 'trainlm', 'learnh');

która tworzy sieć neuronową złożoną z jednej warstwy, a jej argumentami są:

– start – wartości minimalne i maksymalne dla elementów wejściowych sieci dla funkcji tworzących sieć neuronową, składa się z 20 par, które odpowiednio oznaczają wartość maksymalną - 1 oraz minimalną - 0 na wejściu,

– output\_s – liczba elementów wektora wyjściowego sieci (w przypadku tego ćwiczenia jest równa 20 neuronom na wyjściu),

– tansig – parametru określającego funkcję aktywacji neuronów (w tym przypadku tangens hiperboliczny),

– trainlm – funkcja szkolenia sieciowego, która aktualizuje wartości wagi i odchylenia,

– learnh – jest wagą uczenia dla funkcji Hebba.

3

Inne użyte zmienne oraz funkcje:

– net – zmienna, do której będzie przypisywana nowa tworzona sieć neutronowa,

– Input – dane uczące sieci neuronowej

- InputT – transpozycja Input

– Outout – zmienna przechowujące dane wyjściowe odpowiadające danym uczącym (gdzie 1 odpowiada trafieniu w motikonę, a 0 oznacza chybienie),

– lp.dr – wartość współczynnika zapominania dla metody Hebba,

– lp.lr – wartość współczynnika uczenia dla metody Hebba,

– HebianWeights – dostosowanie wartości wag dla metody Hebba za pomocą learnh,

- net.trainParam.epochs – maksymalnej liczby epok,

- net.trainParam.goal – celu wydajności sieci,

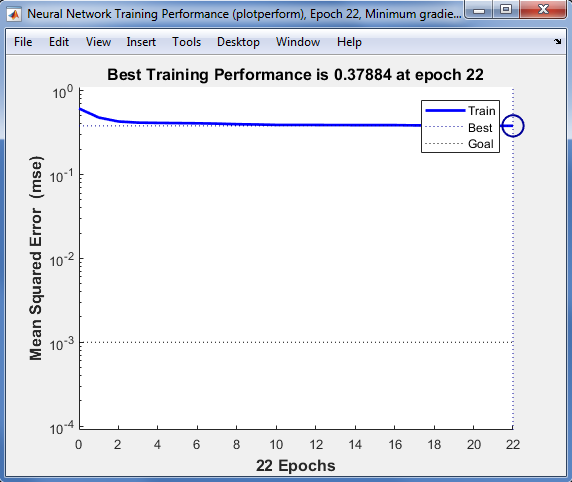
- net.trainParam.lr – wartości współczynnika uczenia się sieci.

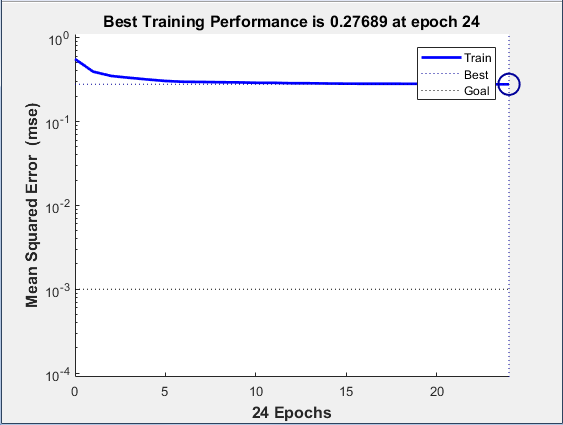
– train(net, InputT, HebbianWeights) – uczenie (trening) sieci net z wykorzystaniem danych wejściowych i wartości określonych wcześniej wag dla reguły Hebba,

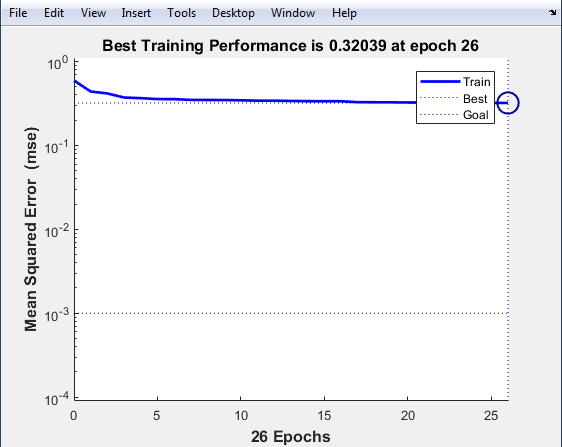
– Test\_emot – zmienna dla danych testujących

- result – końcowy wynik symulacji

**Wykresy zależności błędu od ilości epok dla różnych wartości lr**

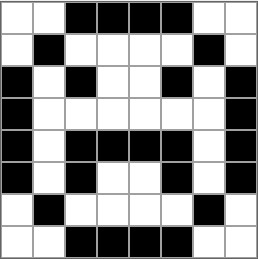
** lr = 0.001**

**lr=0.01**

****

**Lr = 0.1**

**Emotikon użyty do testów:**

****

**Wnioski:**

– Na podstawie powyższych danych można stwierdzić, że funkcja newff poradziła sobie z zadaniem rozpoznawania dużych liter alfabetu nawet w przypadku reguły Hebba.

– Wysoki współczynnik uczenia skutkował szybszym wzrostem wartości wag metody. Dobranie nieodpowiedniego współczynnika uczenia sieci może zwiększać ryzyko występowania błędów podczas treningu sieci (np. dane mogą wyjść poza swój zakres).

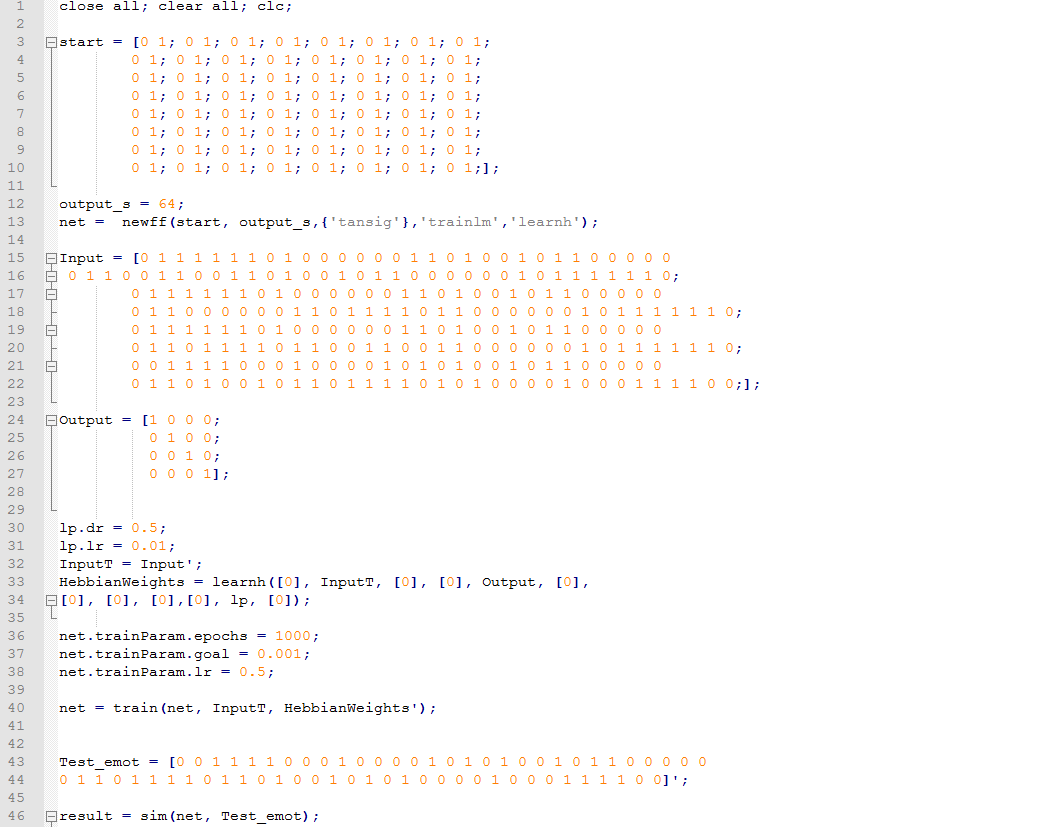
– Ograniczenie epok treningowych wynikało z własności reguły Hebba, która mówi o tym, że użycie jej w algorytmie nie pozwala na samoistne zakończenie programu - wagi w tej metodzie rosną w nieskończoność (przez to program może nigdy się nie skończyć).

– Podczas analizowania wydajności można zauważyć również miejsce, w którym jakość treningu przestaje znacząco wzrastać - może to świadczyć o tym, że jest to punkt, w którym sieć przestaje się uczyć bez większego progresu oraz wzrasta prawdopodobieństwo na zjawisko przeuczenia sieci.

– Reguła Hebba jest dobrą alternatywą w porównaniu do poprzednich metod, ponieważ pozawala na uczenie sieci bez nauczyciela (jest jedną z najbardziej zbliżonych metod do prawdziwej, biologicznej sieci neuronowej).

- Interpretacja danych przez algorytm nie gwarantuje poprawności. Reguła Hebba opiera się na uczeniu bez nauczyciela, przez co sieć jest zmuszona samodzielnie decydować o skutkach w oparciu o dane wejściowe. To zdecydowanie zwiększa możliwość wystąpienia błędu w działaniu algorytmu.

**Listing kodu**

****