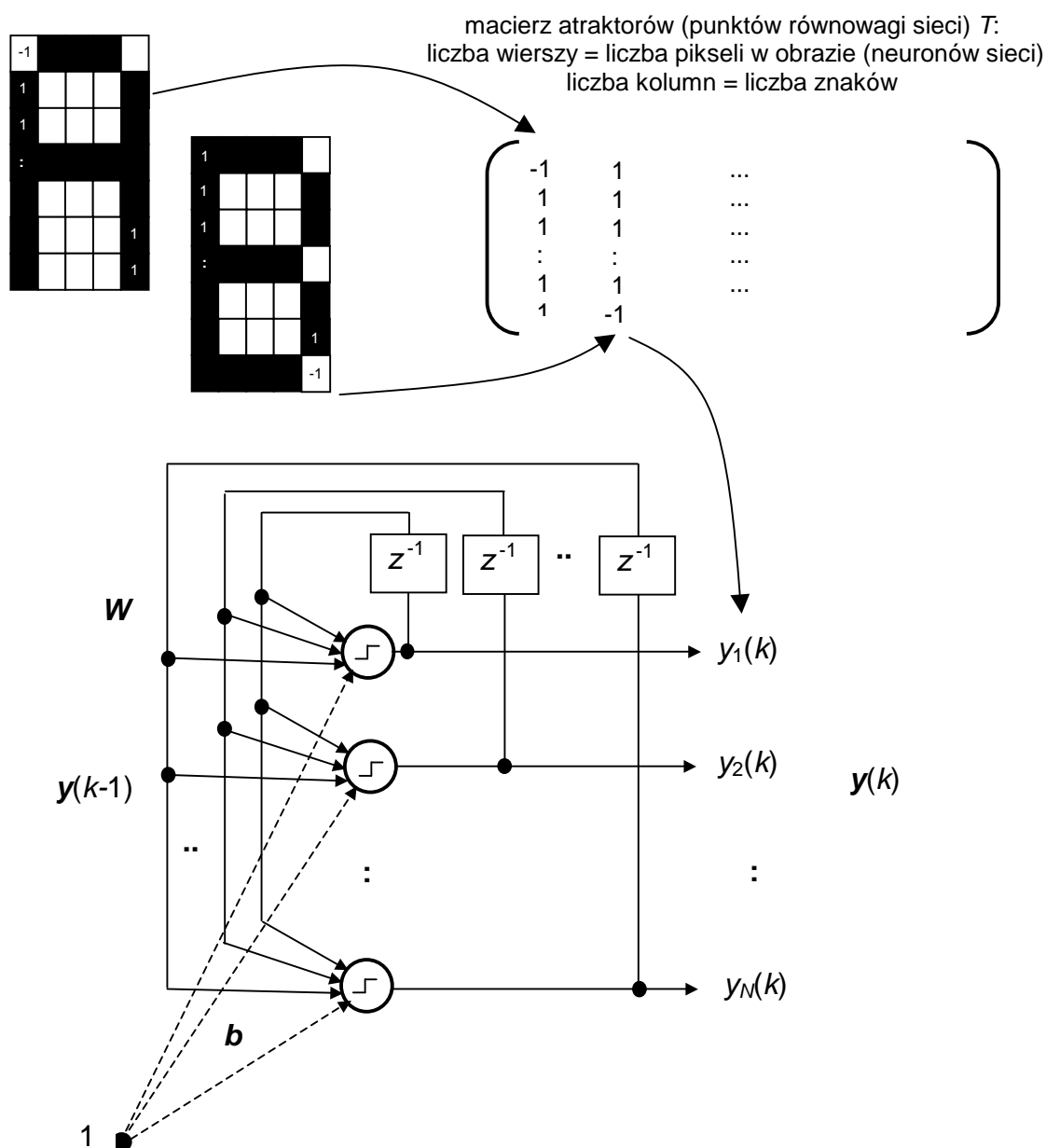


ĆWICZENIE 5

Rekurencyjna sieć Hopfielda jako pamięć skojarzeniowa – – rekonstrukcja wzorców znakowych

Celem ćwiczenia jest zastosowanie rekurencyjnej sieci neuronowej Hopfielda do usuwania zakłóceń z map bitowych, zawierających obrazy wybranych liter alfabetu łańskiego. Działanie sieci umożliwia rekonstrukcję zniekształconego obrazu, poprzez skojarzenie go z jego oryginalnym wzorcem, pełniącym rolę tzw. atraktora (punktu stabilnego) sieci. Przed wykonaniem ćwiczenia należy przypomnieć sobie wiadomości dotyczące sieci rekurencyjnych, zaś w szczególności zaś sieci Hopfielda.



Rys. 5.1. Rekurencyjna sieć Hopfielda jako pamięć skojarzeniowa

Oznaczenia: $y(k)$ – wektor sygnałów wyjściowych w k -tym kroku działania sieci; W – macierz współczynników wagowych sieci;
 b – wektor współczynników progowych; z^{-1} – opóźnienie sygnału o 1 krok

W ćwiczeniu wykorzystane zostaną mapy bitowe, zawierające obrazy liter przydzielonych przez prowadzącego w ćwiczeniu 2.

Utworzona zostanie neuronowa sieć Hopfielda, złożona z pojedynczej warstwy neuronów o liniowej, symetrycznej funkcji aktywacji z nasyceniem ('satlins'). Liczba neuronów sieci (i jednocześnie jej wyjść) odpowiadać będzie liczbie pikseli w obrazie ($N=35$). Rolę atraktorów, czyli stanów stabilnych sieci, będą pełniły wektory reprezentujące wzorcowe obrazy liter (rys. 5.1).

W roli stanów początkowych sieci wykorzystane zostaną zniekształcone (zasmurowane) obrazy poszczególnych liter. W kolejnych krokach działania sieci powinniśmy zaobserwować zmiany jej stanu w kierunku określonego atraktora, widoczne w oknie graficznym jako stopniowe odtwarzanie oryginalnego wzorca znakowego.

Czynności wstępne

Zapoznać się z działaniem dołączonych do biblioteki *Neural Network Toolbox* pakietu *Matlab* skryptów przykładowych, ilustrujących działanie sieci Hopfielda. Dokonać ich edycji (poleceniem `edit nazwa_skryptu`) oraz zapoznać się z ich zawartością i działaniem:

`demohop1` – Ilustracja działania sieci Hopfielda złożonej z dwóch neuronów, dla której zdefiniowano dwa atraktory.

`demohop2` – Przykład identyczny jak powyżej; dodatkowo zademonstrowano istnienie tzw. „fałszywego atraktora”.

Uruchomić także skrypt przykładowy:

`nnd18db` – wizualizacja funkcji energetycznej (Lapunowa) oraz atraktorów sieci Hopfielda złożonej z dwóch neuronów. Możliwa jest zmiana wartości współczynników wagowych sieci i obserwacja wpływu tych zmian na kształt jej funkcji energetycznej.

Przebieg ćwiczenia

W pierwszej części ćwiczenia utworzymy sieć Hopfielda, w której rolę atraktorów będą pełniły cztery 35-elementowe wektory, reprezentujące wzorcowe obrazy liter w postaci prostych map bitowych.

1. Otworzyć nowe okno edytora *m-plików Matlaba* (*File/New/M-File*), zapisać w nim polecenia `close all` oraz `clear`.
2. Zdefiniować cztery macierze o wymiarach 7×5 , reprezentujące mapy bitowe wzorcowych liter (identycznych jak w ćwiczeniu 2).

3. Dokonać wizualizacji obrazów liter. Tym razem wszystkie cztery litery wyświetlić w jednym oknie graficznym, w dwóch wierszach po dwa znaki. Wykorzystać w tym celu funkcje `subplot` oraz `hintonw`.
4. Każdy z obrazów będzie stanowił jeden z atraktorów (punktów przyciągania, stanów stabilnych) sieci Hopfielda. Przed utworzeniem sieci należy zdefiniować macierz atraktorów T , przy czym poszczególne atraktory powinny znaleźć się w kolejnych kolumnach tej macierzy. W celu zmiany rozmiarów macierzy wzorcowych można wykorzystać funkcję `reshape`.
5. Zapisać i uruchomić skrypt. Sprawdzić czy powstała macierz T i jakie są jej rozmiary. Wyświetlić jej zawartość. Jeśli wymiary (lub zawartość) macierzy T nie są właściwe, poprawić w skrypcie jej definicję.
6. Wykorzystując funkcję `newhop` utworzyć sieć Hopfielda złożoną z 35 neuronów, dla której atraktorami będą nasze wzorce znakowe. Zapoznać się z zawartością zmiennej `net` reprezentującej utworzoną sieć.
7. Sprawdzić rozmiary macierzy współczynników wagowych oraz progowych sieci. Wyświetlić ich zawartość:

```
disp('Rozmiary macierzy wag: ')
disp(net.LW)
disp('Zawartość macierzy wag: ')
disp(net.LW{1})
disp('Rozmiar wektora wsp. progowych: ')
disp(net.b)
disp('Zawartość wektora wsp. progowych: ')
disp(net.b{1})
```

W drugiej części ćwiczenia zbadamy „skojarzeniowe” właściwości utworzonej sieci Hopfielda. W tym celu sprawdzimy, czy sieć będzie potrafiła odtwarzać oryginalne wzorce liter na podstawie ich zniekształconych (zaszumionych) obrazów.

8. Skopiować definicję macierzy z obrazami wzorcowymi i wkleić je w dolnej części skryptu. Wprowadzić „ręcznie” do każdego obrazu po jednym zakłóceniu, zmieniając wartość wybranego piksela z -1 na 1 lub odwrotnie. Zmienić nazwy macierzy z zakłóconymi literami. Wszystkie zakłócone obrazy umieścić w macierzy TZ (w taki sam sposób, w jaki z obrazów wzorcowych utworzyliśmy macierz T).
9. Zasymulować (`sim`) działanie sieci Hopfielda, podając jako warunki początkowe stany reprezentujące zakłócone obrazy. Symulację można przeprowadzić w sposób wsadowy, tzn. przekazując do funkcji całą macierz TZ . Składnia funkcji `sim` w rozpatrywanym przypadku jest nieco inna niż w poprzednich ćwiczeniach:

```
Y = sim(net, {LS KS}, {}, {TZ})
```

gdzie:

- `net` jest zmienną reprezentującą utworzoną w punkcie 6. sieć,
- zmienna `LS` reprezentuje liczbę różnych stanów początkowych, dla których chcemy przeprowadzić symulację (w naszym przypadku mamy po jednym zakłóconym obrazie dla każdego wzorca, zatem `LS=4`),
- zmienna `KS` reprezentuje liczbę kroków czasowych symulacji – na początek przyjmujemy `KS=5`, zakładając, że sieć zdoła w pięciu krokach odtworzyć obrazy wzorcowe, czyli odszukać odpowiedni atraktor.

10. W wyniku wywołania funkcji `sim` otrzymujemy zmienną `Y`, która jest tablicą macierzy (*cell array*). W naszym przypadku będzie się ona składała z 5 macierzy – bo tyle kroków trwa symulacja działania sieci (`KS=5`). Każda z tych macierzy będzie miała rozmiary 35×4 i zawierać będzie informacje o stanach kolejnych wyjść sieci (wiersze) dla poszczególnych stanów początkowych (kolumny).

Po wywołaniu funkcji `sim` sprawdzić, czy powstała w przestrzeni roboczej *Matlaba* zmienna `Y`. Sprawdzić jej typ oraz rozmiary (`whos`). Wyświetlić na ekranie zawartość tablicy oraz kolejne macierze wchodzące w jej skład:

```
disp(Y)
disp(Y{1})
:
disp(Y{5})
```

11. Przeprowadzić wizualizację stanów sieci w kolejnych krokach jej działania. Najpierw powinny wyświetlić się, analogicznie jak w punkcie 3., wszystkie obrazy zakłócone przed podaniem ich do sieci. Po kolejnych naciśnięciach klawisza, w tym samym oknie będziemy chcieli zobaczyć obrazy odpowiadające stanom na wyjściach sieci w kolejnych krokach jej działania.

W tym celu należy:

- dodać nowe okno graficzne (`figure`),
- umieścić w skrypcie instrukcję pętli `for .. end`, wyświetlającą w dwóch wierszach okna graficznego (`subplot(2,2,i)`) cztery zakłócone obrazy przed podaniem ich sieci, czyli kolejne kolumny macierzy `TZ`, przekształcone przy pomocy funkcji `reshape` do macierzy o rozmiarach 7×5 ,
- pod instrukcją `end` kończącą pętlę umieścić instrukcję `pause`,
- pod spodem umieścić dwie zagnieżdżone instrukcje `for`, wyświetlające w tym samym oknie po cztery obrazy odpowiadające stanom wyjść sieci w bieżącym kroku.

Uwaga: Zewnętrzna instrukcja `for` powinna „zliczać” kolejne kroki działania sieci, zaś instrukcja wewnętrzna – kolejne wzorce (stany początkowe). Odwołanie do odpowiedniego wektora wyjściowego sieci powinno wyglądać następująco: `Y{k}(:,j)` (`k` – numer kroku, `j` – numer stanu początkowego). Instrukcję `pause` umieścić tylko raz, w pętli zewnętrznej.

12. Zapisać i uruchomić skrypt. Prześledzić zmiany stanów (odpowiedzi) sieci w kolejnych krokach jej działania. Zastanowić się nad interpretacją rozmiarów poszczególnych kwadratów na wykresach graficznych. Jaki jest związek tych rozmiarów z funkcją aktywacji (`satlins`) rozpatrywanej sieci Hopfielda ?
13. Zwiększać stopniowo (np. do 3, 5, 7, itd.) liczbę zakłóceń w obrazach liter. Obserwować odpowiedzi sieci w kolejnych krokach jej działania – w razie potrzeby zwiększyć ich liczbę (np. `KS=10`). Czy zaobserwowano istnienie „fałszywych atraktorów” ?

Zadanie dla ochotników: Obliczyć, zgodnie z poniższą zależnością

$$E(k) = -\frac{1}{2} \mathbf{y}^T(k) \mathbf{W} \mathbf{y}(k) - \mathbf{b}^T \mathbf{y}(k) \quad (5.1)$$

wartość funkcji energetycznej $E(k)$ sieci Hopfielda w każdym kolejnym k -tym kroku jej działania, dla każdego ze stanów (obrazów) początkowych. Jak zmieniają się wartości tej funkcji w kolejnych krokach ? Ile wynoszą jej wartości dla poszczególnych atraktorów ? Obliczanie wartości funkcji $E(k)$ można zrealizować wewnątrz instrukcji pętli `for` z punktu 11.