

## A Implementacja Sieci Hopfielda

*Poniżej opisano projekt do realizacji na pewne kategorie ocen.*

*Aby uzyskać ocenę 4 należy wykonać projekt tylko dla trybu synchronicznego.*

*Natomiast, aby otrzymać ocenę 5 należy opracować także część dotyczącą działania asynchronicznego.*

1. Zapoznaj się z przykładowym skryptem<sup>1</sup> `zad_2a.sce` symulującym działanie Sieci Hopfielda w trybie synchronicznym.
2. Przy wykorzystaniu skryptu symulującego działanie Sieci Hopfielda (`zad_2a.sce`) w trybie synchronicznym napisz parametryzację procesu uruchomienia i uczenia sieci (reguła Hebb'a i metoda pseudoinwersji, obie metody do zaimplementowania) w postaci funkcji `synchronicznaSH()`:

$$Z = \text{synchronicznaSH}(W, \text{pattern}, V, \text{activation\_function})$$

która w wyniku działania zwraca wektory rozpoznane przez sieć (punkty stałe lub rozpoznane wektory dla których sieć osiągnęła warunek stopu<sup>2</sup> dla wektorów niezbieżnych). Ponadto powyższa funkcja w trakcie działania wyświetla w konsoli kroki badania wektorów (zupełnie jak skrypt `zad_2a.sce`).

Parametrami funkcji `synchronicznaSH(W, pattern, V, activation_function)` są:

- a) Macierz wag:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \dots & \dots & \dots & \dots \\ w_{N1} & w_{N2} & \dots & w_{NN} \end{bmatrix}$$

- b) Macierz wektorów wzorcowych  $x^{(1)}, x^{(2)}, \dots, x^{(P)}$ :

$$\text{pattern} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \dots \\ x^{(P)} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_N^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(P)} & x_2^{(P)} & \dots & x_N^{(P)} \end{bmatrix}$$

- c) Macierz wektorów dla których chcemy zbadać zbieżność (aby sprawdzić co sieć rozpozna):

$$V = \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \dots \\ v^{(L)} \end{bmatrix} = \begin{bmatrix} v_1^{(1)} & v_2^{(1)} & \dots & v_N^{(1)} \\ v_1^{(2)} & v_2^{(2)} & \dots & v_N^{(2)} \\ \dots & \dots & \dots & \dots \\ v_1^{(L)} & v_2^{(L)} & \dots & v_N^{(L)} \end{bmatrix}$$

- d) Parametr wyboru funkcji aktywacji: unipolarnej (1) oraz bipolarnej (-1).

gdzie  $N$  to liczba neuronów,  $P$  to liczba wektorów wzorcowych oraz  $L$  to liczba wektorów których zbieżność chcemy badać.

<sup>1</sup> Skrypt napisany w środowisku Scilab 5.3.3

<sup>2</sup> maksymalną liczbę kroków z góry określoną

Funkcję *synchronicznaSH()* możemy uruchomić w dwóch konfiguracjach (tj. trybach) badania rozpoznawania wektorów:

- a) dla zadanych wag bez uczenia, np.  $Z = \text{synchronicznaSH}(W_1, -1, V_1, -1)$ , co oznacza, że sieć ma wagi  $W_1$ , oraz sieć rozpoznaje wektory z macierzy  $V_1$  (wektory w wierszach) i zapisuje rozpoznane do macierzy  $Z$ . Natomiast parametr -1 na 2 miejscu oznacza, że nie przekazujemy wzorców do uczenia, ponieważ sieć już uczyła się pilnie i ma wiedzę zapisaną w  $W_1$ . Czwarty parametr -1 oznacza, że wybrano funkcję aktywacji bipolarną<sup>3</sup>.
- b) z wzorcami do nauczania bez zadanej macierzy wag, np.  $Z = \text{synchronicznaSH}(-1, \text{pattern}_1, V_1, -1)$ , gdzie -1 na pierwszym miejscu oznacza, że sieć nie ma zapisanej wiedzy w macierzy wag. Macierz  $\text{pattern}_1$  zawiera  $P$  wzorcowych wektorów do nauczania. Macierz  $V_1$ , -1 na ostatnim miejscu oraz macierz  $Z$  analogicznie jak poprzednio.

3. **(tryb asynchroniczny)** Przekształć funkcję *synchronicznaSH()* w funkcję *asynchronicznaSH()*, która symuluje proces działania sieci w trybie asynchronicznym z klasycznym sposobem modyfikacji neuronów. W przypadku wektorów, które nie są zbieżne należy przyjąć, że proces rozpoznawania kończy działanie po co najwyżej  $4^{2^{P-1}}$  krokach, gdzie  $P$  to liczba wektorów wzorcowych.

4. Zastosuj zaprogramowaną funkcję *synchronicznaSH()* (oraz funkcję *asynchronicznaSH()*) do nauczania sieci trzech różnych wybranych znaków (np. liter, cyfr, itp.). Następnie określ trzy inne znaki, podobne na pewien sposób do znaków wzorcowych i sprawdź jaki będzie rezultat rozpoznania.

Znak wzorcowy  $x^{(1)} = (x_1^{(1)}, x_2^{(1)}, \dots, x_N^{(1)})$  może przybrać postać tablicy 35 elementowej (macierzy 5x7) jak na poniższym rysunku.

$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_4^{(1)}$	$x_5^{(1)}$
$x_6^{(1)}$	$x_7^{(1)}$	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	$x_{33}^{(1)}$	$x_{34}^{(1)}$	$x_{35}^{(1)}$

Rysunek nr 1. Przykład znaku<sup>4</sup> wzorcowego dla cyfry 1 dla Sieci Hopfielda o  $N = 5 \times 7 = 35$  neuronach.

<sup>3</sup> uwaga, w takim wypadku sieć powinna przyjmować wektory wejściowe o wartościach  $\{-1, 1\}$

<sup>4</sup> Znak jest tutaj rozumiany jako graficzno-tekstowa postać zakodowanej cyfry

Aby ułatwić wyświetlanie znaku zakodowanego jako wektor  $x^{(1)}$  w konsoli należy zaprogramować funkcję *printC*( $x^{(1)}$ ) wyświetlającą w konsoli interpretację tekstowo-graficzną tegoż wektora (funkcja działa zarówno dla wektorów o wartościach  $\{-1,1\}$  jak i dla  $\{0,1\}$ ). Np. dla wektora z Rysunek nr 1 funkcja w konsoli wyświetli:

```
...#.
..##.
.#.#.
#..#.
...#.
...#.
...#.
```

## B Przygotuj sprawozdanie z przeprowadzonego badania wg wzoru:

- 0.Strona tytułowa
- I. Spis treści
- II. Analityczne działanie Sieci Hopfielda
- III. Symulacyjne działanie Sieci Hopfielda
- IV. Zastosowanie sieci Hopfielda do rozpoznawania znaków.
- V. Podsumowanie

Do sprawozdania należy dołączyć płytę CD z projektem (skrypty z dokumentacją oraz postać cyfrowa sprawozdania). Sprawozdanie oddajemy w formie wydrukowanej.

UWAGA: Osoby starające się o ocenę 4 z tegoż projektu mogą pominąć implementację asynchronicznego trybu działania sieci.

### Omówienie sprawozdania:

Ad.II. Przedstawienie analitycznego działania<sup>5</sup> Sieci Hopfielda w trybie synchronicznym (oraz asynchronicznym) na przykładzie macierzy 3x3 z funkcją aktywacji unipolarną i bipolarną (a więc dwa przykłady wag dla każdego trybu (synch. i asynch.) działania sieci). A więc należy wybrać dwie macierze wag dla trybu synchronicznego (oraz dwie dla trybu asynchronicznego). Dla obu trybów należy zbadać wszystkie możliwe wektory wejściowe.

Ad.III. Dla obu powyższych przykładów należy zastosować funkcję *synchronicznaSH()* (oraz *asynchronicznaSH()*), aby sprawdzić czy obliczenia analityczne oraz te powstałe w wyniku symulacji zgadzają się. Należy przedstawić tekstowo proces badania zbieżności (to co generuje funkcja podczas działania w kolejnych krokach) ze szczególnym podkreśleniem procesów zbieżnych.

Ad.IV. Należy opisać sposób kodowania wybranych znaków (np. dla rozmiaru 5x7 wybrać 3 znaki), przedstawić graficzną interpretację znaków oraz przedstawić wynik rozpoznania znaków przez zaprogramowaną funkcję *synchronicznaSH()* (oraz *asynchronicznaSH()*). Następnie, należy podać przykłady 3 podobnych znaków do tych wzorcowych i sprawdzić czy sieć rozpoznała znaki wzorcowe, które są podobne do tych znaków. (Podobny zestaw czynności należy wykonać dla zaprogramowanej funkcji *asynchronicznaSH()*)

---

<sup>5</sup> badacz analizuje zbieżność sieci bez użycia programu, tak jak na ćwiczeniach, rozpisujemy badanie wszystkich możliwych wektorów w poszczególnych krokach.

Ad.V. W podsumowaniu należy streścić co zostało opisane w sprawozdaniu, czy zostały zaprogramowane jakieś rozszerzenia oraz przedstawić wnioski z przeprowadzonych symulacji.

#### Wzory:

- i) Wzór na metodę uczenia wg *reguły Hebba* zgodnie z którą na podstawie wektorów wzorcowych  $X = \{x^{(1)}, x^{(2)}, \dots, x^{(P)}\}$  (gdzie  $x^{(s)} = (x_1^s, x_2^s, \dots, x_N^s)$ , a  $x_i^s$  to  $i$ -ta współrzędna wektora  $x^{(s)}$ ) iteracyjnie obliczamy elementy macierzy wag  $W = \{w_{ij}\}_{i,j=1}^N$ :

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{s=1}^P x_i^{(s)} x_j^{(s)} & \text{dla } i \neq j \\ 0 & \text{dla } i = j \end{cases}$$

Uwaga: Należy pamiętać, iż efektywność metody Hebb'a zależy od liczby neuronów i liczby wzorców. Przyjmuje się, że liczba wektorów wzorcowych nie powinna przekroczyć 13,8% liczby neuronów sieci<sup>6</sup>.

- ii) Wzór na metodę uczenia przez pseudoinwersję<sup>7</sup>. Zakładamy, że dysponujemy macierzą  $X$ , w której wektory wzorcowe znajdują się w wierszach. Jeżeli wektory uczące są liniowo niezależne to macierz wag  $W$  może być wyznaczona ze wzoru:

$$W = X^T * (X * X^T)^{-1} * X$$

#### Propozycje rozszerzeń dla chętnych osób (dla grup 3 osobowych do wyboru jedno obowiązkowe rozszerzenie).

- teoretyczne opisanie działania Sieci Hopfielda w trybach synchronicznym i asynchronicznym;
- rozszerzenie funkcji asynchronicznaSH() o możliwość określania kolejności modyfikacji neuronów
- przekształcenie funkcji wyświetlającą interpretację tekstowo-graficzną wektora na taką, która wyświetla dwa wektory, początkowy (po lewej) oraz ten rozpoznany (po prawej):

```
#.#.    ...#
..##.   ..##.
.#.#.   .#.#.
#.#. -> #.#.
...#    ...#
..##.   ...#
.#.#.   ...#
```

#### UWAGI

1. Prowadzący sprawdza Internet pod kątem wystąpienia kodu w Internecie i w projekcie. Za odnalezienie znaczącej ilości podobnego kodu w projekcie i w sieci Internet jest automatycznie negatywna ocena za projekt. Projekt należy wykonać samodzielnie (w grupach co najwyżej 3 osobowych).
2. Zaprogramowane funkcje będą sprawdzane przez prowadzącego także dla innych wartości wag.
3. Należy określić w projekcie procent wkładu każdego członka grupy projektowej.

<sup>6</sup> Rutkowski, Leszek. *Metody i techniki sztucznej inteligencji: inteligencja obliczeniowa*. Wydawnictwo Naukowe PWN, 2005, s.201

<sup>7</sup> jak wyżej