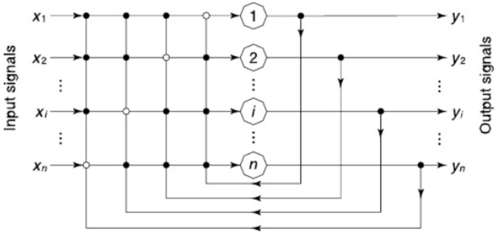
Sieci rekurencyjne

Ewa Adamus

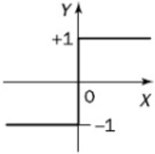
ZUT
Wydział Informatyki
Instytut Sztucznej Inteligencji i Metod Matematycznych

7 maja 2012

Sieć Hopfielda



Jednowarstwowa sieć Hopfielda, z n–neuronami



Bipolarna funkcja przejścia

W wariancie dyskretnym sieci stosuje się zazwyczaj neurony ze skokowymi, bipolarnymi funkcjami przejścia:

$$Y = \begin{cases} +1 & \text{if } X \geq 0 \\ -1 & \text{if } X < 0 \end{cases}$$

Bieżący stan sieci jest określony przez zestaw wyjść neuronów $y_1, y_2, \ldots, y_n$:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Wagi w macierzowej formie:

$$\mathbf{W} = \sum_{m=1}^{M} \mathbf{Y}_m \mathbf{Y}_m^T - M * \mathbf{I},$$

gdzie M to liczba stanów do zapamiętania przez sieć, $\mathbf{Y}_m$ to n wymiarowy binarny wektor, $\mathbf{I}$ $n \times n$ macierz identycznościowa.

Przykład

Załóżmy, że mamy do zapamiętania dwa stany $(1, 1, 1)$ oraz $(-1, -1, -1)$:

$$\mathbf{Y1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ and } \mathbf{Y1} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

.

Możemy określić wagi:

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} * \begin{bmatrix} -1 & -1 & -1 \end{bmatrix} - 2 * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \end{aligned}$$

Przykład cd.

Po określeniu wag można przetestować sieć podając wektory wejściowe $\mathbf{X_1}$ oraz $\mathbf{X_2}$ (które są równoważne wektorom docelowym $\mathbf{Y_1}$, $\mathbf{Y_2}$).
Po podaniu wektora wejściowego $\mathbf{X}$ oraz obliczeniu aktualnego wyjścia $\mathbf{Y}$ porównujemy wynik z wektorem wejściowym. W celu uzyskania wyjścia:

$$\mathbf{Y_m} = sign(\mathbf{W} * \mathbf{X_m} - \theta), \quad m = 1, 2, \ldots, M$$

gdzie θ jest wartością progową, w przypadku naszej funkcji przejścia równą 0.
Dla naszego przykładu, wyjścia sieci:

$$\mathbf{Y_1} = sign\left\{ \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

oraz

$$\mathbf{Y_2} = sign\left\{ \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} * \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Wszystko się zgadza, obydwa stany możemy określić jako *stabilne*.

Przykład cd.

Jak wygląda sytuacja z pozostałymi stanami? Dla sieci złożonej z trzech neuronów (o skokowych funkcjach przejścia) możliwych jest osiem stanów. W naszym przypadku pozostałe sześć stanów jest niestabilnych. Natomiast stabilne stany (**fundamentalna pamięć** powinny „przyciągać" stany im bliskie. **W ramach ćwiczeń obliczeniowych proszę sprawdzić, czy tak jest rzeczywiście, a uzyskane wyniki umieścić w tabeli.**

Algorytm treningowy sieci Hopfielda

- **Zapamiętywanie**. Wagi n – neuronowa sieć Hopfielda, wymaganej do zapamiętania M wzorców: $\mathbf{Y_1}, \mathbf{Y_2}, \dots, \mathbf{Y_M}$ w formie macierzowej:

$$\mathbf{W} = \sum_{m=1}^{M} \mathbf{Y}_m \mathbf{Y}_m^T - M * \mathbf{I}. \tag{1}$$

  Raz obliczone wagi pozostają stałe.

- **Testowanie**. W celu potwierdzenia, że sieć prawidłowo odtwarza wszystkie wzorce *fundamentalnej macierzy* $\mathbf{Y_m}$, które są równoważne wejściom $\mathbf{X_m}$:

$$\mathbf{Y_m} = sign(\mathbf{W} * \mathbf{X_m} - \theta), \quad m = 1, 2, \dots, M. \tag{2}$$

  Jeżeli wszystkie wzorce odtwarzane są prawidłowo, możemy przejść do następnego kroku.

Algorytm treningowy sieci Hopfielda cd.

- **Odzyskiwanie danych**. Zaprezentujmy sieci nieznaną próbkę $\mathbf{X}$, w celu uzyskania stanu stabilnego. Zazwyczaj, w takim przypadku próbka jest niekompletną bądź uszkodzoną wersją stanu z pamięci fundamentalnej: $\mathbf{X} \neq \mathbf{Y_m}$ dla $m = 1, 2, \ldots, M$.
  - Obliczmy odpowiedź sieci dla naszej próbki $\mathbf{X}(0)$, w iteracji $p = 0$:
  $$\mathbf{Y}(0) = sign(\mathbf{W} * \mathbf{X}(0) - \theta)$$
  - Obliczony sygnał wyjściowy podajemy z powrotem na wejście sieci i w ramach kolejnej iteracji ponownie obliczmy wyjście. Porównujemy obydwa wektory $\mathbf{Y}(p)$ oraz $\mathbf{Y}(p+1)$ czyli wyjście uzyskane w bieżącej iteracji z wcześniejszym. Jeżeli obydwa wektory są niezmienione, oznacza to, że uzyskaliśmy stan stabilny. Warunek stabilności:
  $$\mathbf{Y}(p+1) = sign(\mathbf{W} * \mathbf{Y}(p) - \theta)$$

Okazuje się, że wprzypadku sieci Hopfielda uzyskany stan stabilny niekoniecznie musi odpowiadać wzorcowi z fundamentalnej pamięci, a jeżeli odpowiada to nie musi to być stan najbliższy badanej próbce

Przykład nr 2

Dla sieci zlożonej z pięciu neuronów umieść w pamięci fundamentalnej trzy wzorce:

$$
\begin{aligned}
\mathbf{X}_1 &= (& 1, && 1, && 1, && 1, && 1) \\
\mathbf{X}_2 &= (& 1, && -1, && 1, && -1, && 1) \\
\mathbf{X}_3 &= (& -1, && 1, && -1, && 1, && -1)
\end{aligned}
$$

Następnie sprawdź reakcję sieci na nową próbkę testową:$\mathbf{X} = (1, 1, -1, 1, 1)$.

Pojemność pamięci

Hopfield wykazał eksperymentalnie, że dla n– neuronowej sieci maksymalna liczba możliwych wzorców do zapamiętania wynosi:

$$M_{max} = 0.15 * n \tag{3}$$

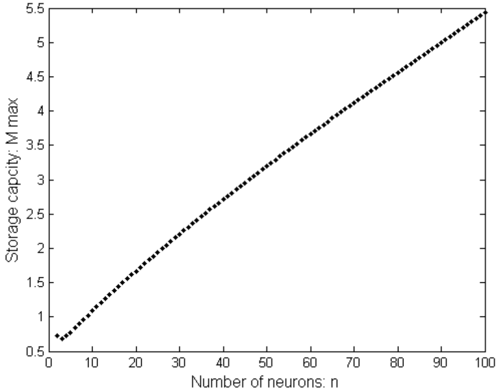
Generalnie, większość wzorców może być odzyskana z pamięci o maksymalnej pojemności:

$$M_{max} = \frac{n}{2 * \ln n} \tag{4}$$

Natomiast wszystkie próbki odzyskamy perfekcyjne dla pamięci o pojemności zmniejszonej o połowę:

$$M_{max} = \frac{n}{4 * \ln n} \tag{5}$$

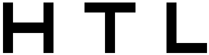
Pojemność pamięci



Rysunek: Zależność liczby neuronów n w sieci Hopfielda od maksymalnej pojemności: $M_{max} = \frac{n}{4 * \ln n}$

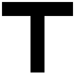
Przykłady zastosowań sieci Hopfielda. Zadanie do samodzielnej realizacji – odtwarzanie znaków. Przygotowanie wzorców do zapamiętania

Należy przygotować własny zbiór **X** wektorów przeznaczonych do zapamiętania w sieci. Np.



Rysunek: Przykładowe bitmapy (rozmiar $[5 \times 5]$ pikseli) ze znakami do zapamiętania

Przykładowo, litera `T` może być przedstawiona w postaci macierzy o wartościach: 1 piksel zapalony oraz -1 zgaszony.



$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 \end{matrix}$$

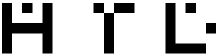
W postaci wektorowej (przykładowo kolumnowej) wzorzec dla `T`:

```
X=[1,-1,-1,-1,-1,1,-1,-1,-1,-1,1,1,1,1,1,1,-1,-1,-1,-1,1,-1,-1,-1 ,-1]
```

Odtwarzanie znaków, cd.

Dla przygotowanej macierzy **X** ze wzorcami do zapamiętania, należy zrealizować przedstawiony wcześniej algorytm sieci Hopfielda. Jak wygląda macierz wag? Z ilu neuronów składa się sieć? Jaka maksymalna liczba wzorców może zostać przechowana w pamięci?

Zweryfikować możliwości odtworzeniowe sieci. W tym celu należy przygotować znaki zaszumione, uszkodzone w stosunku do zapamietanych wzorców, np.



Rysunek: Przykładowe bitmapy (rozmiar [5×5] pikseli) ze znakami do zapamiętania

Jakich odpowiedzi udzieliła sieć? Czy są zgodne z zapamiętanymi wzorcami?

Zadanie, dla własnego zbioru znaków, należy zrealizować w formie sprawozdania.