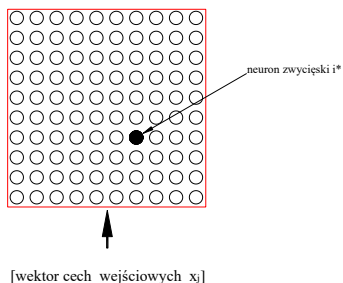


Schemat wyjściowej warstwy sieci o algorytmie uczenia o konkurencji ostrej



1

Algorytm uczenia Kohonena

Kwantowanie wektorowe

Modyfikacja wag połączeń sieci:

$$w_{ij} = w_{ij} + \eta(t) (x_i^{\mu} - w_{ij}),$$

gdzie:

$\eta(t)$ – współczynnik uczenia,

t – numer iteracji,

x_i^{μ} – wartość j -tej cechy μ -tego wzorca wejściowego,

w_{ij} – wartość wagowa połączenia wejściowego węzła j z i -tym neuronem wyjściowym.

Wartość wyjściowa:

$$h_i^{\mu} = \sum_j w_{ij} x_i^{\mu} = \mathbf{w}_i \cdot \mathbf{x}^{\mu}$$

Dla miary Euklidesowej:

$$|\mathbf{w}_{i^*} - \mathbf{x}^{\mu}| \leq \mathbf{w}_i \cdot \mathbf{x}^{\mu}$$

Dla innych miar:

$$d(\mathbf{x}^{\mu}, \mathbf{w}_{i^*}) = \min d(\mathbf{x}, \mathbf{w}_i), \quad 0 \leq i \leq n$$

2

Algorytm uczenia LVQ Kohonena

Learning quantization vector – adaptacyjne kwantowania wektorowe

Podstawowe założenia:

Algorytmy WTA i WTM dzielą obszar wejściowy na określone strefy wpływów poszczególnych neuronów. Wagi neuronu zwyciężającego wyznaczają punkt centralny dla grupy wektorów wejściowych, który oznaczamy jako \mathbf{w}_c (vektor Voronoia).

Zbiór wszystkich wektorów centralnych nosi nazwę książki kodowej.

VQ - w każdym obszarze wejściowym R^n wyróżnia się tzw. „wektory kodujące” (codebook vectors), gdzie przy zastosowaniu reguły najbliższego sąsiada można określić klasę dla nowego wzorca \mathbf{x} .

Każdemu wzorcowi wejściowemu przypisujemy klasę C co stanowi algorytm uczenia LVQ.

Kolejne wektory wejściowe porównuje się z wektorami centralnymi. Jeżeli klasy wektora centralnego i wejściowego są zgodne to następuje zmiana wartości wektora centralnego tak aby zbliżyć się do wektora wejściowego, w przeciwnym przypadku wektor centralny jest odsuwany od wejściowego.

3

Algorytm uczenia LVQ Kohonena

Learning quantization vector – adaptacyjne kwantowania wektorowe

Podstawowy algorytm LVQ1:

1. Wprowadzenie kolejnego wektora wejściowego $\mathbf{x}_i, i = 1..p$

$$c = \arg \min_k \|\mathbf{x} - \mathbf{w}_k\|$$

2. Porównanie klasy C_{x_i} przypisanej wektorowi \mathbf{x}_i i klasy C_{w_c} przypisanej do \mathbf{w}_c .

• Jeżeli $C_{x_i} = C_{w_c}$, to $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha_i [\mathbf{x}_i - \mathbf{w}_c(t)]$

• Jeżeli $C_{x_i} \neq C_{w_c}$, to $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha_i [\mathbf{x}_i - \mathbf{w}_c(t)]$

3. Pozostałe wektory nie ulegają zmianie.

4

Algorytm uczenia LVQ Kohonena

Learning quantization vector – adaptacyjne kwantowania wektorowe

Algorytm LVQ2.1:

Wyznaczenie okna do którego wpada wektor wejściowy wg zależności:

$$\min \left(\frac{d_i}{d_j}, \frac{d_j}{d_i} \right) > s, \quad s = \frac{1-e}{1+e}, \quad e = 0.2..0.3$$

Adaptacja wag:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) - \alpha_i [\mathbf{x}_i - \mathbf{w}_i(t)]$$

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha_j [\mathbf{x}_i - \mathbf{w}_j(t)]$$

Gdzie: i - wektor o innej klasie niż \mathbf{x} ,

j - wektor o tej samej klasie co \mathbf{x} .

5

Algorytm uczenia LVQ Kohonena

Learning quantization vector – adaptacyjne kwantowania wektorowe

Algorytm LVQ3:

Wyznaczenie okna do którego wpada wektor wejściowy wg zależności:

$$\min \left(\frac{d_i}{d_j}, \frac{d_j}{d_i} \right) > s, \quad s = \frac{1-e}{1+e}, \quad e = 0.2..0.3$$

Adaptacja wag:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) - \alpha_i [\mathbf{x}_i - \mathbf{w}_i(t)]$$

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha_j [\mathbf{x}_i - \mathbf{w}_j(t)]$$

Gdzie: i - wektor o innej klasie niż \mathbf{x} ,

j - wektor o tej samej klasie co \mathbf{x} .

Natomiast, jeżeli $\mathbf{x}_i, \mathbf{w}_i$ oraz \mathbf{w}_j należą do tej samej klasy:

$$\mathbf{w}_h(t+1) = \mathbf{w}_h(t) + e\alpha_i [\mathbf{x}_i - \mathbf{w}_h(t)], \quad h \in \{i, j\}$$

6