Politechnika Warszawska

OKNO

MSI

C6. Sieci neuronowe

Włodzimierz Kasprzak

Zadanie 1. Implementacja funkcji logicznych

Pojedynczy neuron może realizować liniowe funkcje logiczne. Przedstawić budowę pojedynczego neuronu implementującego funkcje logiczne (Boole'a):

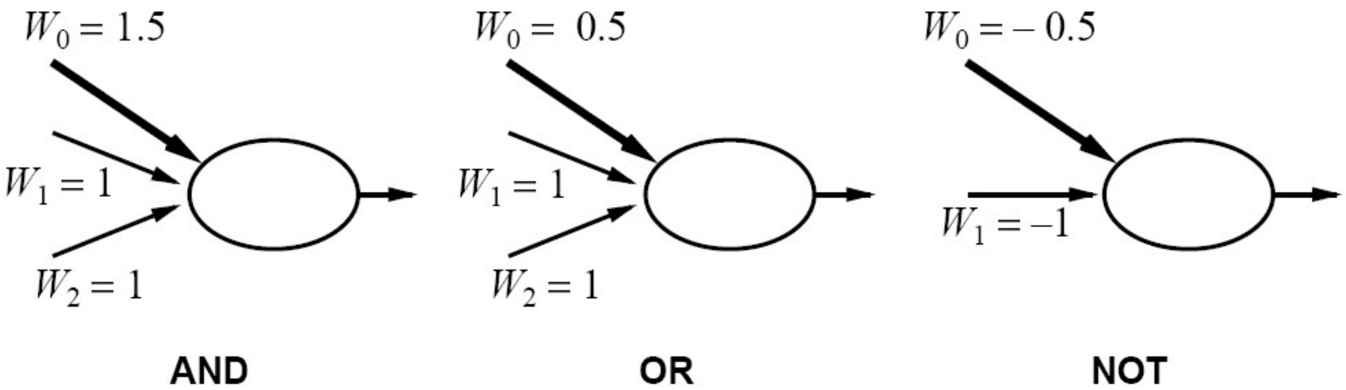
- AND,
- OR,
- NOT.
- Przedstawić budowę sieci implementującej funkcję XOR.

Rozwiązanie 1

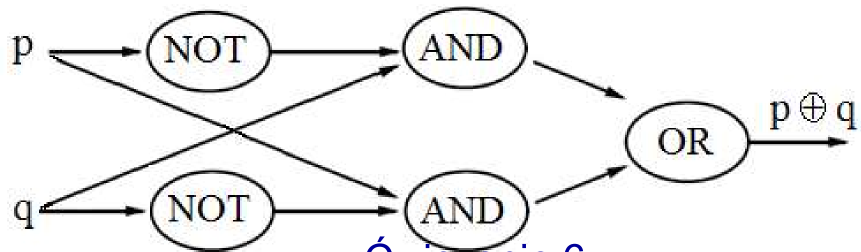
Neuron realizuje funkcję iloczynu AND, gdy ustalimy dla niego wagi: w_0 = 1.5, w_1=1, w_2=1, a jako funkcję aktywacji wybierzemy funkcję skoku.

Neuron realizuje funkcję sumy OR, gdy ustalimy dla niego wagi: w_0 = 0.5, w_1=1, w_2=1, a jako funkcję aktywacji wybierzemy funkcję skoku.

Neuron realizuje funkcję negacji NOT, gdy ustalimy dla niego wagi: w_0= -0.5, w_1= -1,, a jako funkcję aktywacji wybierzemy funkcję skoku.



Sieć wielowarstwowa realizuje funkcję XOR**.**



Zadanie 2. Uczenie sieci MLP

Wykonać jedną pełną iterację procesu uczenia 3-warstwowego perceptronu (jedna warstwa ukryta), z 3 neuronami w warstwie ukrytej i po dwóch wejściach i wyjściach. Uczenie przebiega według podstawowej metody wstecznej propagacji błędu, przy następujących założeniach:

- stały współczynnik uczący $= 0.1$,
- *sigmoidalna* funkcja aktywacji, $z = \quad \theta(y) = \dfrac{1}{1+\exp(-y)}$
- początkowe wartości wag w macierzach $\mathbf{W}^{(1)}$ i $\mathbf{W}^{(2)}$ wynoszą

$$\mathbf{W}^{(1)} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \mathbf{W}^{(2)} = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}$$

- pierwsza próbka ucząca: $\mathbf{x}=(1,2)$; $f(\mathbf{x}) = (1, 0)$.

Dla ułatwienia można przybliżyć wartości funkcji aktywacji zgodnie z wartościami w poniższej tabelce:

y	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	2.0	3.0
z=θ(y)	0.525	0.55	0.575	0.60	0.623	0.646	0.668	0.69	0.71	0.73	0.75	0.88	0.95

y	-3	-2	-1	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0
z=θ(y)	0.05	0.12	0.27	0.29	0.31	0.33	0.35	0.375	0.40	0.425	0.45	0.475	0.50

Rozwiązanie 2 (1)

1) Propagacja wprzód - z wejścia do wyjścia

$$\mathbf{y}^{(1)}(1) = \mathbf{W}^{(1)}(0) \cdot \mathbf{x}(1) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

$$\mathbf{z}^{(1)}(1) = \psi(\mathbf{y}^{(1)}(1)) = \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}$$

$$\mathbf{y}^{(2)}(1) = \mathbf{W}^{(2)}(0) \cdot \mathbf{z}^{(1)}(1) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} = \begin{bmatrix} 0.05 \\ -0.05 \end{bmatrix}$$

$$\mathbf{z}^{(2)}(1) = \psi(\mathbf{y}^{(2)}(1)) = \begin{bmatrix} 0.5125 \\ 0.4875 \end{bmatrix}$$

Rozwiązanie 2 (2)

2) Propagacja błędu wstecz

2.A) Wektor korekcji dla warstwy wyjściowej

$$\mathbf{d}^{(2)}(1) = (\mathbf{s}(1) - \mathbf{z}^{(2)}(1)).*\,(1 - \mathbf{z}^{(2)}(1)).*\,\mathbf{z}^{(2)}(1) =$$

$$= \left(\begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}0.5125\\0.4875\end{bmatrix}\right).*\left(\begin{bmatrix}1\\1\end{bmatrix} - \begin{bmatrix}0.5125\\0.4875\end{bmatrix}\right).*\begin{bmatrix}0.5125\\0.4875\end{bmatrix} \cong\cdot \begin{bmatrix}0.1218\\-0.1218\end{bmatrix}$$

2.B) Wektor korekcji dla warstwy ukrytej

$$\mathbf{d}^{(1)}(1) = [\mathbf{W}^{(2)}(0)]^T \cdot \mathbf{d}^{(2)}(1).*\,(1 - \mathbf{z}^{(1)}(1)).*\,\mathbf{z}^{(1)}(1) =$$

$$= \begin{bmatrix}1.0 & -1.0\\-1.0 & 1.0\\1.0 & -1.0\end{bmatrix} \cdot \begin{bmatrix}0.1218\\-0.1218\end{bmatrix}.*\left(\begin{bmatrix}1\\1\\1\end{bmatrix} - \begin{bmatrix}0.27\\0.95\\0.73\end{bmatrix}\right).*\begin{bmatrix}0.27\\0.95\\0.73\end{bmatrix} =$$

$$= \begin{bmatrix}0.2436\\-0.2436\\0.2436\end{bmatrix} \cdot * \begin{bmatrix}0.1971\\0.0475\\0.1971\end{bmatrix} \cong \begin{bmatrix}0.04801\\-0.01157\\0.04801\end{bmatrix}$$

Rozwiązanie 2 (3)

3) Modyfikacja wag

3.A) Modyfikacja macierzy wag $\mathbf{W}^{(2)}$

$$\Delta\mathbf{W}^{(2)}(1) = 0.1 \cdot \begin{bmatrix} 0.1218 \\ -0.1218 \end{bmatrix} \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}^T = 0.1 \cdot \begin{bmatrix} 0.0329 & 0.1157 & 0.0889 \\ -0.0329 & -0.1157 & -0.0889 \end{bmatrix}$$

$$\mathbf{W}^{(2)}(1) = \mathbf{W}^{(2)}(0) + \Delta\mathbf{W}^{(2)}(1)$$

$$= \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.0329 & 0.1157 & 0.0889 \\ -0.0329 & -0.1157 & -0.0889 \end{bmatrix} =$$

$$\cong \begin{bmatrix} 1.0033 & -0.9884 & 1.0089 \\ -1.0033 & 0.9884 & -1.0089 \end{bmatrix}$$

Rozwiązanie 2 (4)

3.B) Modyfikacja macierzy wag $\mathbf{W}^{(1)}$

$$\Delta\mathbf{W}^{(1)}(1) = 0.1 \cdot \begin{bmatrix} 0.04801 \\ -0.01157 \\ 0.04801 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T = 0.1 \cdot \begin{bmatrix} 0.04801 & 0.09602 \\ -0.01157 & -0.02314 \\ 0.04801 & 0.09602 \end{bmatrix}$$

Macierz wag $\mathbf{W}^{(1)}$ po modyfikacji:

$$\mathbf{W}^{(1)}(1) = \mathbf{W}^{(1)}(0) + \Delta\mathbf{W}^{(1)}(1)$$

$$= \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.04801 & 0.09602 \\ -0.01157 & -0.02314 \\ 0.04801 & 0.09602 \end{bmatrix} \cong \begin{bmatrix} 1.00480 & -0.99040 \\ 0.99884 & 0.99769 \\ -0.99520 & 1.00960 \end{bmatrix}$$

Zadanie 3. Entropia krzyżowa

Wykonać pierwszą iterację algorytmu uczenia sieci w pełni połączonej o jednej warstwie ukrytej, realizującej klasyfikację binarną – tzn. o pojedynczym wyjściu, sigmoidalnej funkcji aktywacji i entropii krzyżowej jako funkcji straty).

Założyć 3 neurony w warstwie ukrytej i dwa wejścia. Dalsze założenia:

Stała ucząca, $\mu = 0.1$;

Sigmoidalna funkcja aktywacji: $z = \theta(y) = \frac{1}{1+exp(-y)}$

Początkowa macierz wag $\mathbf{W}^{(1)}$ i początkowy wektor wag $\boldsymbol{w}^{(2)}$:

$$\boldsymbol{W}^{(1)}(0) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix}; \qquad \boldsymbol{w}^{(2)^T}(0) = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}$$

Pierwsza próbka danych, $\mathbf{x}(1)=[1,\ 2]$, należy do klasy dodatniej, tzn. oczekiwane wyjście, $z(1)=f(\mathbf{x}(1)) = 1$.

Rozwiązanie 3 (1)

1. Propagacja wprzód

$$\mathbf{y}^{(1)}(1) = \mathbf{W}^{(1)}(0) \cdot \mathbf{x}(1) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

$$\mathbf{z}^{(1)}(1) = \theta(\mathbf{y}^{(1)}(1)) = \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}$$

$$y^{(2)}(1) = \boldsymbol{w}^{(2)^T}(0) \cdot \mathbf{z}^{(1)}(1) = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} = 0.05$$

$$z^{(2)}(1) = \theta(y^{(2)}(1)) \cong 0.5125$$

2. Wartość korekty dla wyjścia

$$d^{(2)}(1) = \left(s(1) - z^{(2)}(1)\right) = 1 - 0.5125 = 0.4875$$

Rozwiązanie 3 (2)

3. Wektor korekty dla wag warstwy ukrytej

$$\mathbf{d}^{(1)}(1) = \boldsymbol{w}^{(2)}(0) \cdot d^{(2)}(1).*\left(1 - \mathbf{z}^{(1)}(1)\right).*\mathbf{z}^{(1)}(1) =$$

$$= \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \cdot 0.4875 \ .*\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}\right).*\begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.4875 \\ -0.4875 \\ 0.4875 \end{bmatrix} \cdot * \begin{bmatrix} 0.2241 \\ 0.0475 \\ 0.1971 \end{bmatrix} \cong \begin{bmatrix} 0.10925 \\ -0.02316 \\ 0.09609 \end{bmatrix}$$

4. Modyfikacja wektora wag neuronu wyjściowego

$$\Delta \boldsymbol{w}^{(2)^T}(1) = \mu \cdot d^{(2)}(1) \cdot \mathbf{z}^{(1)^T}(1) = 0.1 \cdot 0.4875 \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}^T$$

$$= 0.1 \cdot [0.1316 \quad 0.4631 \quad 0.3559]$$

$$\boldsymbol{w}^{(2)^T}(1) = \boldsymbol{w}^{(2)^T}(0) + \Delta \boldsymbol{w}^{(2)^T}(1)$$

$$= [1 \quad -1 \quad 1] + 0.1 \cdot [0.1316 \quad 0.4631 \quad 0.3559] =$$

$$\cong [1.0132 \quad -0.9537 \quad 1.0356]$$

Rozwiązanie 3 (3)

5. Modyfikacja macierzy wag warstwy ukrytej

$$\Delta\mathbf{W}^{(1)}(1) = \mu \cdot \mathbf{d}^{(1)}(1) \cdot \mathbf{z}^{(0)^T}(1) = \mu \cdot \mathbf{d}^{(1)}(1) \cdot \mathbf{x}(1)$$
$$= 0.1 \cdot \begin{bmatrix} 0.10925 \\ -0.02316 \\ 0.09609 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T = 0.1 \cdot \begin{bmatrix} 0.10925 & 0.21850 \\ -0.02316 & -0.04632 \\ 0.09609 & 0.19218 \end{bmatrix}$$

$$\mathbf{W}^{(1)}(1) = \mathbf{W}^{(1)}(0) + \Delta\mathbf{W}^{(1)}(1) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.10925 & 0.21850 \\ -0.02316 & -0.04632 \\ 0.09609 & 0.19218 \end{bmatrix} =$$
$$\cong \begin{bmatrix} 1.0109 & -0.9782 \\ 0.9977 & 0.9954 \\ -0.9904 & 1.0192 \end{bmatrix}$$

Zadanie 4. Softmax

Wykonać pierwszą iterację algorytmu uczenia dwu-warstwowej sieci w pełni połączonej realizującej klasyfikację wieloklasową (przyjąć dwa wyjścia) z aktywacją typu *softmax* i entropią krzyżową jako funkcją straty.

Założyć 3 neurony w warstwie ukrytej i dwa wejścia. Dalsze założenia:

- Stała ucząca, $\mu = 0.1$;
- Sigmoidalna funkcja aktywacji: $z = \theta(y) = \frac{1}{1+exp(-y)}$
- Początkowe macierze wag $\mathbf{W}^{(1)}$ i $\mathbf{W}^{(2)}$:

$$\boldsymbol{W}^{(1)}(0) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix}; \qquad \boldsymbol{W}^{(2)}(0) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}$$

- Pierwsza próbka danych, $\mathbf{x}(1)=[1, 2]$ i oczekiwane wyjście, $k(1)= f(\mathbf{x}(1)) = [1, 0]$.

Rozwiązanie 4 (1)

1. Propagacja wprzód

$$\mathbf{y}^{(1)}(1) = \mathbf{W}^{(1)}(0) \cdot \mathbf{x}(1) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

$$\mathbf{z}^{(1)}(1) = \theta(\mathbf{y}^{(1)}(1)) = \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}$$

$$\mathbf{y}^{(2)}(1) = \mathbf{W}^{(2)}(0) \cdot \mathbf{z}^{(1)}(1) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} = \begin{bmatrix} 0.05 \\ -0.05 \end{bmatrix}$$

Aktywacja typu softmax w warstwie wyjściowej:

$$\mathbf{z} = \mathbf{y}, \qquad \mathbf{z}^{(2)}(1) = \mathbf{y}^{(2)}(1) = \begin{bmatrix} 0.05 \\ -0.05 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$s_1 = \frac{\exp(z_i)}{\sum_{j=1}^{2} \exp(z_j)}, \qquad i = 1,2$$

$$s_1 = \frac{\exp(0.05)}{\exp(0.05) + \exp(-0.05)} \cong \frac{1.05127}{1.05127 + 0.95123} = \frac{1.05127}{2.0025} \cong 0.525$$

$$s_2 = 1 - s_1 = 0.475 \qquad \mathbf{s}^{(2)}(1) \cong \begin{bmatrix} 0.525 \\ 0.475 \end{bmatrix}$$

Rozwiązanie 4 (2)

2. Wektor korekty dla neuronów warstwy wyjściowej

$$\mathbf{d}^{(2)}(1) = \boldsymbol{k}(1) - \boldsymbol{s}^{(2)}(1) = \left(\begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}0.525\\0.475\end{bmatrix}\right) =\cdot \begin{bmatrix}0.475\\-0.475\end{bmatrix}$$

3. Wektor korekty dla neuronów warstwy ukrytej

$$\mathbf{d}^{(1)}(1) = [\mathbf{W}^{(2)}(0)]^T \cdot \mathbf{d}^{(2)}(1).*(1 - \mathbf{z}^{(1)}(1)).*\mathbf{z}^{(1)}(1) =$$

$$= \begin{bmatrix}1 & -1\\-1 & 1\\1 & -1\end{bmatrix} \cdot \begin{bmatrix}0.475\\-0.475\end{bmatrix}.*\left(\begin{bmatrix}1\\1\\1\end{bmatrix} - \begin{bmatrix}0.27\\0.95\\0.73\end{bmatrix}\right).*\begin{bmatrix}0.27\\0.95\\0.73\end{bmatrix} =$$

$$= \begin{bmatrix}0.95\\-0.95\\0.95\end{bmatrix} \cdot * \begin{bmatrix}0.2241\\0.0475\\0.1971\end{bmatrix} \cong \begin{bmatrix}0.2128\\-0.04513\\0.1872\end{bmatrix}$$

4. Modyfikacja wag warstwy wyjściowej

$$\Delta\mathbf{W}^{(2)}(1) = \mu \cdot \mathbf{d}^{(2)}(1) \cdot \mathbf{z}^{(1)^T}(1) = 0.1 \cdot \begin{bmatrix}0.475\\-0.475\end{bmatrix} \cdot \begin{bmatrix}0.27\\0.95\\0.73\end{bmatrix}^T$$

$$= 0.1 \cdot \begin{bmatrix}0.12825 & 0.45125 & 0.34675\\-0.12825 & -0.45125 & -0.34675\end{bmatrix}$$

Rozwiązanie 4 (3)

$$\mathbf{W}^{(2)}(1) = \mathbf{W}^{(2)}(0) + \varDelta\mathbf{W}^{(2)}(1)$$
$$= \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.12825 & 0.45125 & 0.34675 \\ -0.12825 & -0.45125 & -0.34675 \end{bmatrix} =$$
$$\cong \begin{bmatrix} 1.0128 & -0.9549 & 1.0347 \\ -1.0128 & 0.9549 & -1.0347 \end{bmatrix}$$

5. Modyfikacja macierzy wag warstwy ukrytej

$$\varDelta\mathbf{W}^{(1)}(1) = \mu \cdot \mathbf{d}^{(1)}(1) \cdot \mathbf{z}^{(0)^T}(1) = \mu \cdot \mathbf{d}^{(1)}(1) \cdot \mathbf{x}(1)$$
$$= 0.1 \cdot \begin{bmatrix} 0.2128 \\ -0.04513 \\ 0.1872 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T = 0.1 \cdot \begin{bmatrix} 0.2128 & 0.4256 \\ -0.04513 & -0.09026 \\ 0.1872 & 0.3744 \end{bmatrix}$$

$$\mathbf{W}^{(1)}(1) = \mathbf{W}^{(1)}(0) + \varDelta\mathbf{W}^{(1)}(1) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.2128 & 0.4256 \\ -0.04513 & -0.09026 \\ 0.1872 & 0.3744 \end{bmatrix} =$$
$$\cong \begin{bmatrix} 1.0213 & -0.9574 \\ 0.9955 & 0.9910 \\ -0.9813 & 1.0374 \end{bmatrix}$$

Zadanie 5. RNN

Wykonać dwie iteracje algorytmu uczenia sieci rekurencyjnej z jedną warstwą ukrytą, realizującej klasyfikację wieloklasową. Przyjąć dwa wyjścia z aktywacją typu *softmax* i entropią krzyżową jako funkcją straty.

Założyć 3 neurony w warstwie ukrytej i dwa wejścia. Dalsze założenia:

Stała ucząca, $\mu = 0.1$;

Sigmoidalna funkcja aktywacji w warstwie ukrytej: $z = \theta(y) = \frac{1}{1+exp(-y)}$

Początkowe wagi sieci:

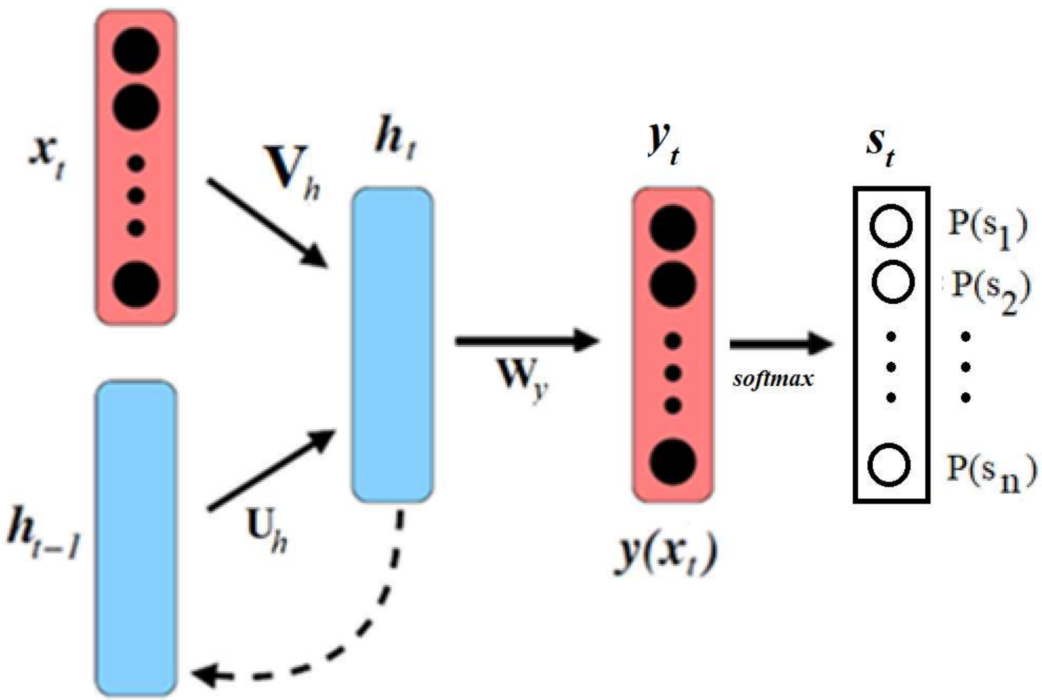
$$\boldsymbol{V}_h(0) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix}; \quad \boldsymbol{W}_y(0) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}; \quad \boldsymbol{U}_h(0) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix};$$

Pierwsze dwie próbki uczące: $\boldsymbol{x}(1)=[1, 2]^T$; $\boldsymbol{x}(2)=[2, 1]^T$;

Etykiety wyjść: $\boldsymbol{k}(1) = [1, 0]^T$; $\boldsymbol{k}(2) = [0, 1]^T$

Zadanie 5 (c.d.)

Sieć RNN z jedną warstwa ukrytą:



Rozwiązanie 5 (1)

Iteracja 1.

1.1 Propagacja wprzód

$$\boldsymbol{h}(1) = \boldsymbol{V}_h(0) \cdot \boldsymbol{x}(1) + \boldsymbol{U}_h(0) \cdot \boldsymbol{h}(0) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

$$\boldsymbol{z}^{(1)}(1) = \theta(\boldsymbol{h}\ (1)) = \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}$$

$$\boldsymbol{y}(1) = \boldsymbol{W}_y\ (0) \cdot \boldsymbol{z}^{(1)}(1) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} = \begin{bmatrix} 0.05 \\ -0.05 \end{bmatrix}$$

$$\boldsymbol{z}(1) = \boldsymbol{y}(1) = \begin{bmatrix} 0.05 \\ -0.05 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$s_i = \frac{\exp(z_i)}{\sum_{j=1}^{2} \exp(z_j)}, \qquad i = 1,2$$

$$s_1 = \frac{\exp(0.05)}{\exp(0.05) + \exp(-0.05)} \cong \frac{1.05127}{1.05127 + 0.95123} = \frac{1.05127}{2.0025} \cong 0.525$$

Rozwiązanie 5 (2)

Iteracja 1) 1.1 Propagacja wprzód (c.d.)

$$s_1 \cong 0.525$$

$$s_2 = 1 - s_1 = 0.475$$

Wyjście po iteracji 1: $\boldsymbol{s}^{(2)}(1) \cong \begin{bmatrix} 0.525 \\ 0.475 \end{bmatrix}$

1.2 Propagacja błędu wstecz

Korekta dla wyjścia: $\boldsymbol{d}^{(2)}(1) = \boldsymbol{k}(1) - \boldsymbol{s}^{(2)}(1) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.525 \\ 0.475 \end{bmatrix} \right) =\cdot \begin{bmatrix} 0.475 \\ -0.475 \end{bmatrix}$

Korekta dla warstwy ukrytej: $\boldsymbol{d}^{(1)}(1) = [\boldsymbol{W}_y\,(0)]^T \cdot \boldsymbol{d}^{(2)}(1).*\,(1 - \boldsymbol{z}^{(1)}(1)).*\,\boldsymbol{z}^{(1)}(1) =$

$$= \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0.475 \\ -0.475 \end{bmatrix} .* \left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} \right) .* \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.95 \\ -0.95 \\ 0.95 \end{bmatrix} \cdot* \begin{bmatrix} 0.2241 \\ 0.0475 \\ 0.1971 \end{bmatrix} \cong \begin{bmatrix} 0.2128 \\ -0.04513 \\ 0.1872 \end{bmatrix}$$

Rozwiązanie 5 (3)

Iteracja 1) 1.3 Modyfikacja wag

$$\Delta \boldsymbol{W}^{(2)}(1) = \mu \cdot \mathbf{d}^{(2)}(1) \cdot \mathbf{z}^{(1)^T}(1) = 0.1 \cdot \begin{bmatrix} 0.475 \\ -0.475 \end{bmatrix} \cdot \begin{bmatrix} 0.27 \\ 0.95 \\ 0.73 \end{bmatrix}^T$$

$$= 0.1 \cdot \begin{bmatrix} 0.12825 & 0.45125 & 0.34675 \\ -0.12825 & -0.45125 & -0.34675 \end{bmatrix}$$

$$\boldsymbol{W}^{(2)}(1) = \boldsymbol{W}^{(2)}(0) + \Delta \boldsymbol{W}^{(2)}(1)$$
$$= \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.12825 & 0.45125 & 0.34675 \\ -0.12825 & -0.45125 & -0.34675 \end{bmatrix} =$$
$$\cong \begin{bmatrix} 1.0128 & -0.9549 & 1.0347 \\ -1.0128 & 0.9549 & -1.0347 \end{bmatrix}$$

$$\Delta \boldsymbol{V}_h(1) = \mu \cdot 0.5 \cdot \mathbf{d}^{(1)}(1) \cdot \boldsymbol{x}(1)^{\boldsymbol{T}} = 0.05 \cdot \begin{bmatrix} 0.2128 \\ -0.04513 \\ 0.1872 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T$$

$$= 0.05 \cdot \begin{bmatrix} 0.2128 & 0.4256 \\ -0.04513 & -0.09026 \\ 0.1872 & 0.3744 \end{bmatrix}$$

Rozwiązanie 5 (4)

Iteracja 1) 1.3 Modyfikacja wag (c.d.)

$$\boldsymbol{V}_h(1) = \boldsymbol{V}_h(0) + \Delta\boldsymbol{V}_h(1) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} 0.1064 & 0.2128 \\ -0.02256 & -0.04513 \\ 0.0936 & 0.1872 \end{bmatrix} =$$

$$\cong \begin{bmatrix} 1.0106 & -0.9787 \\ 0.9977 & 0.9955 \\ -0.9904 & 1.0187 \end{bmatrix}$$

$$\Delta\boldsymbol{U}_h(1) = \mu \cdot 0.5 \cdot \mathbf{d}^{(1)}(1) \cdot \boldsymbol{h}(0)^{\boldsymbol{T}} = 0.05 \cdot \begin{bmatrix} 0.2128 \\ -0.04513 \\ 0.1872 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T = 0.05 \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{U}_h(1) = \boldsymbol{U}_h(0) + \Delta\boldsymbol{U}_h(1) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Rozwiązanie 5 (5)

Iteracja 2. 2.1 Propagacja wprzód

$$\boldsymbol{h}(2) = \boldsymbol{V}_h(1) \cdot \boldsymbol{x}(2) + \boldsymbol{U}_h(1) \cdot \boldsymbol{h}(1) = \begin{bmatrix} 1.0106 & -0.9787 \\ 0.9977 & 0.9955 \\ -0.9904 & 1.0187 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1.0425 \\ 6.9909 \\ -0.9621 \end{bmatrix}$$

$$\boldsymbol{z}^{(1)}(2) = \theta(\boldsymbol{h}\ (2)) = \begin{bmatrix} 0.7393 \\ 0.9991 \\ 0.2765 \end{bmatrix}$$

$$\boldsymbol{y}(2) = \boldsymbol{W}_y\ (1) \cdot \boldsymbol{z}^{(1)}(2) = \begin{bmatrix} 1.0128 & -0.9549 & 1.0347 \\ -1.0128 & 0.9549 & -1.0347 \end{bmatrix} \cdot \begin{bmatrix} 0.7393 \\ 0.9991 \\ 0.2765 \end{bmatrix} = \begin{bmatrix} 0.0808 \\ -0.0808 \end{bmatrix}$$

$$\boldsymbol{z}(1) = \boldsymbol{y}(1) = \begin{bmatrix} 0.0808 \\ -0.0808 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$s_i = \frac{\exp(z_i)}{\sum_{j=1}^{2} \exp(z_j)}, \qquad i = 1,2$$

$$s_1 = \frac{\exp(0.0808)}{\exp(0.0808) + \exp(-0.0808)} \cong \frac{1.0842}{1.0842 + 0.9224} = \frac{1.0842}{2.0065} \cong 0.5403$$

Rozwiązanie 5 (6)

Iteracja 2) 2.1 Propagacja wprzód (c.d.)

$$s_1 \cong 0.540$$

$$s_2 = 1 - s_1 = 0.460$$

Wyjście po iteracji 2: $\boldsymbol{s}^{(2)}(2) \cong \begin{bmatrix} 0.540 \\ 0.460 \end{bmatrix}$

2.2 Propagacja błędu wstecz

Korekta dla wyjścia: $\boldsymbol{d}^{(2)}(2) = \boldsymbol{k}(2) - \boldsymbol{s}^{(2)}(2) = \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.540 \\ 0.460 \end{bmatrix} \right) = \begin{bmatrix} -0.54 \\ 0.54 \end{bmatrix}$

Korekta dla warstwy ukrytej:

$$\boldsymbol{d}^{(1)}(2) = [\boldsymbol{W}_y\,(1)]^T \cdot \boldsymbol{d}^{(2)}(2) .* \left(1 - \boldsymbol{z}^{(1)}(2)\right) .* \boldsymbol{z}^{(1)}(2) =$$

$$= \begin{bmatrix} 1.0128 & -1.0128 \\ -0.9549 & 0.9549 \\ 1.0347 & -1.0347 \end{bmatrix} \cdot \begin{bmatrix} -0.54 \\ 0.54 \end{bmatrix} .* \left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.7393 \\ 0.9991 \\ 0.2765 \end{bmatrix} \right) .* \begin{bmatrix} 0.7393 \\ 0.9991 \\ 0.2765 \end{bmatrix} =$$

$$\cong \begin{bmatrix} -0.2108 \\ 0.0009 \\ -0.2235 \end{bmatrix}$$

Rozwiązanie 5 (7)

Iteracja 2) 2.3 Modyfikacja wag

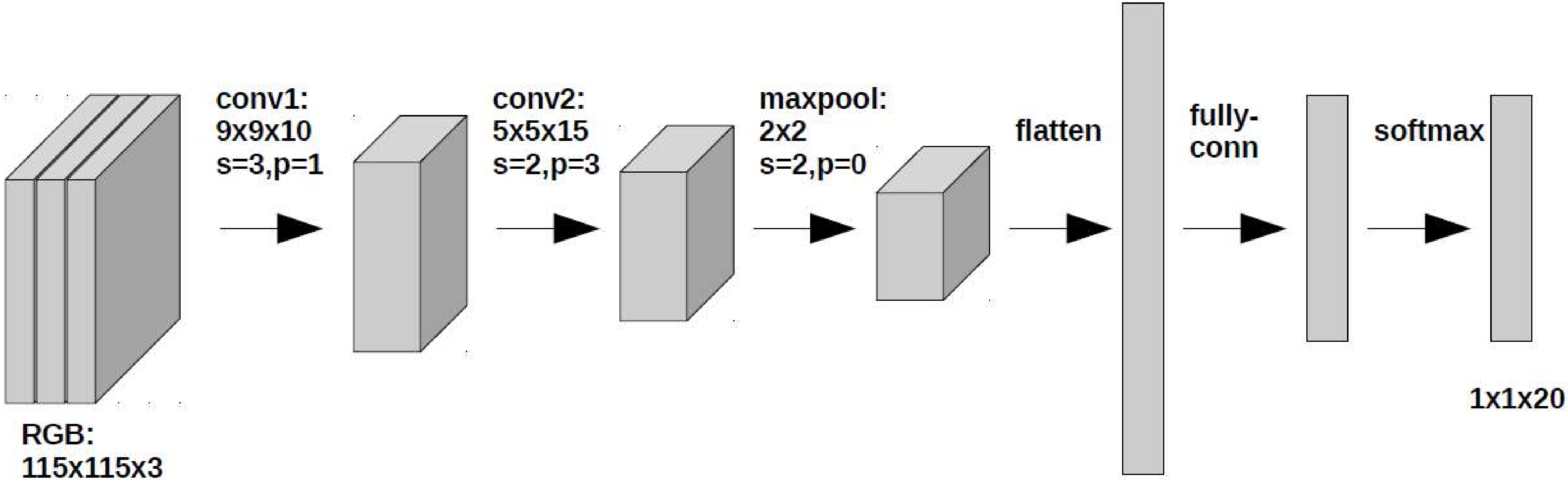
$$\Delta \boldsymbol{W}^{(2)}(2) = \mu \cdot \mathbf{d}^{(2)}(2) \cdot \mathbf{z}^{(1)^{T}}(2)$$
$$\boldsymbol{W}^{(2)}(2) = \boldsymbol{W}^{(2)}(1) + \Delta \boldsymbol{W}^{(2)}(2)$$

$$\Delta \boldsymbol{V}_h(2) = \mu \cdot 0.5 \cdot \mathbf{d}^{(1)}(2) \cdot \boldsymbol{x}(2)^{\boldsymbol{T}}$$
$$\boldsymbol{V}_h(2) = \boldsymbol{V}_h(1) + \Delta \boldsymbol{V}_h(2)$$

$$\Delta \boldsymbol{U}_h(2) = \mu \cdot 0.5 \cdot \mathbf{d}^{(1)}(2) \cdot \boldsymbol{h}(1)^{\boldsymbol{T}}$$
$$\boldsymbol{U}_h(2) = \boldsymbol{U}_h(1) + \Delta \boldsymbol{U}_h(2)$$

Zadanie 6. Analiza sieci CNN

Dany jest model Splotowej Sieci Neuronowej (CNN) o następującej strukturze (funkcje aktywacji zostały pominięte):



Parametry splotu są podane jako *filterH×filterW×numFilter;* warstwy max-pooling są określone jako *filterH × filterW; s* jest parametrem *stride*, a *p* jest parametrem *symmetrical padding*.

Warstwa *flatten* przekształca (zachowując liczbę elementów) wolumen danych do wektora. Początkowy i końcowy rozmiar wolumenów został podany na rysunku.

Zadanie 6 (c.d.)

Zadania

- Obliczyć rozmiary wolumenów danych dla każdej warstwy sieci
- Obliczyć liczby wag dla każdej warstwy sieci (włączając wagę wartości stałej ’bias’).
- Obliczyć całkowitą liczbę wag sieci.

Rozwiązanie 6 (1)

Warstwa splotowa

- Volumen wejściowy: $V_{in} = H_{in} \times W_{in} \times C_{num}$

(wysokość obrazu × szerokość obrazu × liczba kanałów – składowe koloru)

- „ Stride”: (S_h, S_w) - jądro filtru przesuwa się po obrazie o (S_h, S_w) zamiast $(1, 1)$;
- „Padding”: (P_h, P_w) - poszerzenie obrazu (kanału) wejściowego o (P_h, P_w) w celu pełnego wykorzystania brzegowych danych;
- Liczba filtrów : f_{num} , rozmiar filtrów $f_w \times f_h \times C_{num}$

Volumen wyjściowy warstwy, $H_{out} \times W_{out} \times f_{num}$, wyznaczony jest jako:

$$H_{out} = \frac{H_{in} - f_h + 2P_h}{S_h} + 1$$

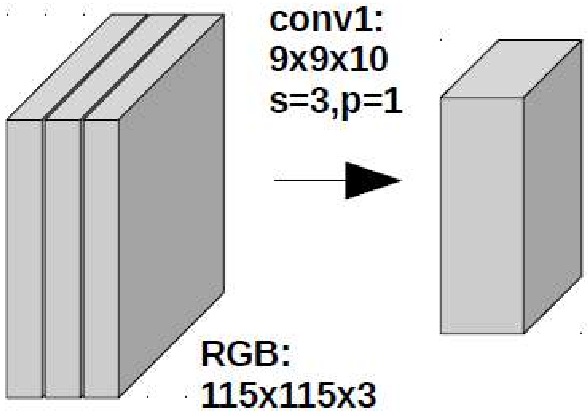
$$W_{out} = \frac{W_{in} - f_w + 2P_w}{S_w} + 1$$

a liczba kanałów na wyjściu wynosi f_{num}.

Liczba wag warstwy, $L_{weights} = (f_h \times f_w \times C_{num} + 1) \times f_{num}$

Rozwiązanie 6 (2)

Warstwa splotowa nr 1 (conv1)



- Volumen wejściowy: $V_{in} = H_{in} \times W_{in} \times C_{num} = 115 \cdot 115 \cdot 3 = 39.675$
- „Stride”: $(S_h, S_w) = (3, 3)$
- „Padding”: $(P_h, P_w) = (1, 1)$
- Liczba filtrów : f_{num}=10, rozmiar filtrów $f_{\mathrm{w}} \times f_{\mathrm{h}} \times C_{num}$=9×9×3

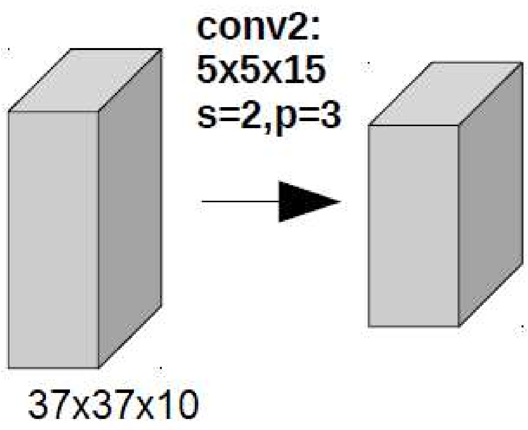
Volumen wyjściowy warstwy, $V_{out}(conv1) = H_{out} \times W_{out} \times f_{num} = 37 \cdot 37 \cdot 10 = 13.690$, gdzie :

$$H_{out} = \frac{115 - 9 + 2 \cdot 1}{3} + 1 = 37\,; \quad W_{out} = \frac{115 - 9 + 2 \cdot 1}{3} + 1 = 37$$

Liczba wag warstwy, $L_{weights}(conv1) = (f_h \times f_w \times C_{num} + 1) \times f_{num} = (9 \cdot 9 \cdot 3 + 1) \cdot 10 = 2.440$

Rozwiązanie 6 (3)

Warstwa splotowa nr 2 (conv2)



- Volumen wejściowy: $V_{in} = H_{in} \times W_{in} \times f_{num} = 37 \cdot 37 \cdot 10 = 13.690$
- „Stride": $(S_h, S_w) = (2, 2)$
- „Padding": $(P_h, P_w) = (3\ , 3)$
- Liczba filtrów : $f_{num} = 15$, rozmiar filtrów $f_w \times f_h \times C_{num} = 5 \times 5 \times 10$

Volumen wyjściowy warstwy, $V_{out}(conv2) = H_{out} \times W_{out} \times f_{num} = 20 \cdot 20 \cdot 15 = 6.000$, gdzie :

$$H_{out} = \frac{37 - 5 + 2 \cdot 3}{2} + 1 = 20\ ; \quad W_{out} = \frac{37 - 5 + 2 \cdot 3}{2} + 1 = 20$$

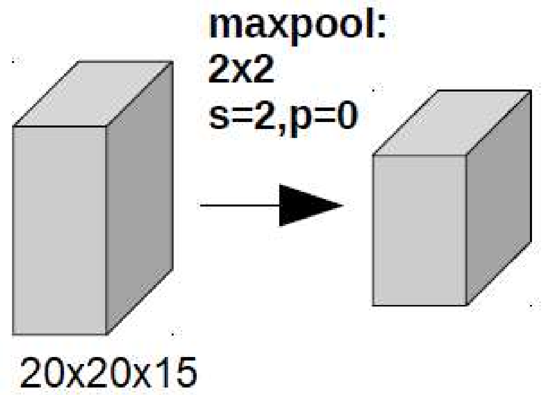
Liczba wag warstwy, $L_{weights}(conv1) = (f_h \times f_w \times C_{num} + 1) \times f_{num} = (5 \cdot 5 \cdot 10 + 1) \cdot 15 = 3.765$

Rozwiązanie 6 (4)

Warstwa „pooling” (pod-próbkowanie)

- Zadaniem warstwy „pooling” jest ograniczenie rozmiaru danych. Stosowana jest filtracja lokalna realizująca ustalone funkcje (brak wag/parametrów do trenowania). Typowe funkcje to MAX (wybiera maksymalną wartość) lub AVERAGE (wyznacza średnią wartość).
- Filtracja wykonywana jest osobno dla każdego kanału.

Warstwa „maxpool”



- Volumen wejściowy: $V_{in} = H_{in} \times W_{in} \times C_{num} = 20 \cdot 20 \cdot 15 = 6.000$
- „ Stride”: $(S_h, S_w) = (2, 2)$
- „Padding”: $(P_h, P_w) = (0, 0)$
- Liczba filtrów : $f_{num}=15$, rozmiar filtrów $f_w \times f_h = 2\times2$

Rozwiązanie 6 (5)

Warstwa „maxpool" (c.d.)

Volumen wyjściowy warstwy, $V_{out} = H_{out} \times W_{out} \times f_{num} = 10 \cdot 10 \cdot 15 = 1.500$, gdzie H_{out}, W_{out} wyznaczone są identycznie jak dla warstwy splotowej jako:

$$H_{out} = \frac{H_{in} - f_h + 2P_h}{S_h} + 1 = \frac{20 - 2 + 2 \cdot 0}{2} + 1 = 10$$

$$W_{out} = \frac{W_{in} - f_w + 2P_w}{S_w} + 1 = \frac{20 - 2 + 2 \cdot 0}{2} + 1 = 10$$

Liczba wag warstwy, $L_{weights}(maxpool) = 0$

Funkcja „flatten" (skanowanie)

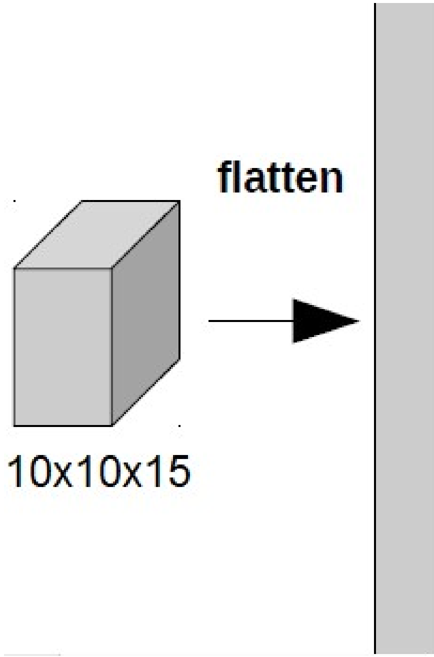
Przekształcenie volumenu wejściowego,

$V_{in} = H_{in} \times W_{in} \times C_{num}$, na wektor 1-wymiarowy,

$V_{out} = 1 \times 1 \times H_{in} \cdot W_{in} \cdot C_{num}$.

W tym zadaniu: $V_{out} = 1 \times 1 \times 10 \cdot 10 \cdot 15 = 1.500$

Liczba wag warstwy, $L_{weights}(flatten) = 0$



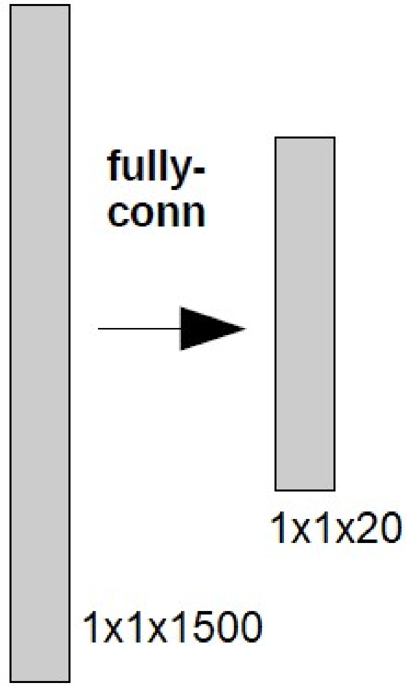
Rozwiązanie 6 (6)

Warstwa „w pełni połączona” (FC)

Wektor wejściowy: $V_{in}(FC) = 1 \times 1 \times 1.500.$

Wektor wyjściowy warstwy FC odpowiada wyjściu funkcji „softmax”: $V_{out}(FC) = 1 \times 1 \times 20 = 20.$

Liczba wag warstwy, $L_{weights}(FC) = 20 \cdot (1.500 + 1) = 30.020$

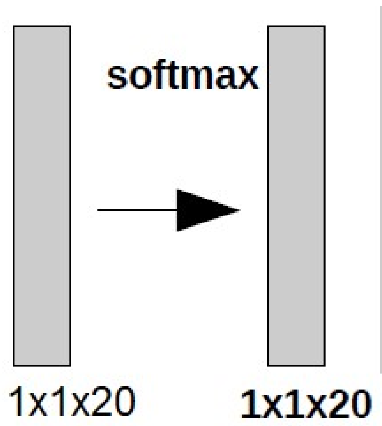


Funkcja „softmax”

Wektor wejściowy: $V_{in}(softmax) = 1 \times 1 \times 20$

Wektor wyjściowy: $V_{out}(softmax) = 1 \times 1 \times 20$

Liczba wag warstwy, $L_{weights}(softmax) = 0$



Całkowita liczba wag tej sieci:

2.440+3.765+0+0+30.020+0 = 36.225

Zadanie 7. Sieć dla problemu kWTA

Napisać program symulujący działanie sieci rekurencyjnej rozwiązującej problem znalezienia k zwycięzców (kWTA) ze zbioru N dodatnich liczb.

Rozwiązanie 7 (1)

Program w Matlabie:

```
function [z, iternum] = kWTA( u, maxiter, k )
% Argumenty: u -  wektor „n” liczb (sygnały wejściowe dla sieci),
%                 maxiter – maksymalna liczba iteracji,
%                 k – liczba szukanych zwycięzców.
% Zwracany wynik: z – wektor „n” liczb (sygnały wyjściowe) –
% niezerowa wartość na i-tym wyjściu wskazuje, że odpowiednie
% i-te wejście należy do zbioru „k zwycięzców”.
      [c, n] = size(u);
      if (k> (n-1)) % błędna wartość argumentu k
        iternum = 0;
        z = u;
       return;
      end
      eps = 1.0/(n+k);
```

Rozwiązanie 7 (2)

```
% Inicjalizacja
z = u;   newz = z;
iternum = maxiter; % maksymalna liczba iteracji
for i=1 : maxiter % ewentualnie przerwij wcześniej, jeśli problem zostanie
rozwiązany
   sumuvec = sum(z);      zeronum = 0;
   for j=1:n
      newz(j) = u(j) + z(j) - eps *(sumuvec - z(j)); % reguła modyfikacji sieci RNN
      if newz(j) < 0  % nieliniowość RELU
         newz(j) = 0;         zeronum = zeronum +1;
      end
   end
   z = newz; % synchroniczna modyfikacja wszystkich wyjść
   % Sprawdź aktualną liczbę niezerowych wyjść u(j):
   if zeronum >= (n-k)
      iternum = i;       break; // zakończ
   end
end
```