LAB (Autoenkoder)

$\square = 0.0$, $\blacksquare = 1.0$

Wejścia (input): $u(p) \in \mathbb{R}^{10}$ $(1 \le p \le 2)$ wektor kolumnalny

$$u(1) = \begin{bmatrix} u_1(1) \\ \vdots \\ u_{10}(1) \end{bmatrix} = \begin{array}{|c|c|c|} \hline u_1(1) & u_2(1) & u_3(1) \\ \hline u_4(1) & u_5(1) & u_6(1) \\ \hline u_7(1) & u_8(1) & u_9(1) \\ \hline \end{array} \; u_{10}(1) = \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \\ \hline & \blacksquare & \\ \hline & \blacksquare & \\ \hline \end{array} \; \blacksquare \in \mathbb{R}^{10}$$

$$u(2) = \begin{bmatrix} u_1(2) \\ \vdots \\ u_{10}(2) \end{bmatrix} = \begin{array}{|c|c|c|} \hline u_1(2) & u_2(2) & u_3(2) \\ \hline u_4(2) & u_5(2) & u_6(2) \\ \hline u_7(2) & u_8(2) & u_9(2) \\ \hline \end{array} \; u_{10}(2) = \begin{array}{|c|c|c|} \hline & & \blacksquare \\ \hline & & \blacksquare \\ \hline & & \blacksquare \\ \hline \end{array} \; \blacksquare \in \mathbb{R}^{10}$$

<u>Enkoder</u> $u(p) \mapsto x(p)$ $(1 \le p \le 2)$

Sygnały szyfrowane (*ang.* ciphered) lub kompresowane: $x(p) \in \mathbb{R}^3$ $(1 \le p \le 2)$ wektor kolumnalny

$$x(p) = \begin{bmatrix} x_1(p) \\ x_2(p) \\ x_3(p) \end{bmatrix} = \begin{array}{|c|} \hline x_1(p) \\ \hline x_2(p) \\ \hline \end{array} \; x_3(p) = \begin{array}{|c|} \hline ? \\ \hline ? \\ \hline \end{array} \; \blacksquare \in \mathbb{R}^3$$

$w_1 = \begin{bmatrix} w_{1\,1} & \cdots & w_{1\,10} \end{bmatrix} \in (\mathbb{R}^{10})^*$ wektor wag rządowy

$x_1(p) = f(w_1 u(p)) = f(\sum_{j=1}^{10} w_{1\,j} u_j(p))$

$w_2 = \begin{bmatrix} w_{2\,1} & \cdots & w_{2\,10} \end{bmatrix} \in (\mathbb{R}^{10})^*$ wektor wag rządowy

$x_2(p) = f(w_2 u(p)) = f(\sum_{j=1}^{10} w_{2\,j} u_j(p))$

$x_3(p) \equiv 1$

$f(x) = \frac{1}{1+e^{-\beta x}}$

<u>Dekoder</u> (*ang.* decipherment) $x(p) \mapsto y(p)$ $(1 \le p \le 2)$

Wyjścia (output): $y(p) \in \mathbb{R}^9$ $(1 \le p \le 2)$ wektor kolumnalny

$$y(p) = \begin{bmatrix} y_1(1) \\ \vdots \\ y_9(p) \end{bmatrix} = \begin{array}{|c|c|c|} \hline y_1(p) & y_2(p) & y_3(p) \\ \hline y_4(p) & y_5(p) & y_6(p) \\ \hline y_7(p) & y_8(p) & y_9(p) \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline \end{array} \in \mathbb{R}^9$$

$$y_j(p) = f(\sum_{i=1}^{3} s_{j\,i} x_i(p)) \quad (1 \le j \le \mathbf{9}) \quad (1 \le p \le 2)$$

$f(x) = \frac{1}{1+e^{-\beta x}}$

Zadania.

Zadanie (1) Podamy parametry wag $w_{i\,j}$, $s_{j\,i}$ następująco:

$w_{1\,1} = 20$, $w_{1\,10} = -10$; $w_{1\,j} = 0$ gdy $j \neq 1, 10$

$w_{2\,3} = 20$, $w_{2\,10} = -10$; $w_{2\,j} = 0$ gdy $j \neq 3, 10$

$s_{j\,i} = 20 u_j(i)$ dla $i = 1, 2$; $s_{j\,3} = -10$

($\beta = 2.5$)
Wyświetlić **obrazy** z liczbami

$$x(p) = \begin{bmatrix} x_1(p) \\ x_2(p) \\ x_3(p) \end{bmatrix} = \begin{array}{|c|}\hline x_1(p) \\ \hline x_2(p) \\ \hline \end{array} \; x_3(p) \quad = \begin{array}{|c|}\hline ? \\ \hline ? \\ \hline \end{array} \; \blacksquare$$

$$y(p) = \begin{bmatrix} y_1(1) \\ \vdots \\ y_9(p) \end{bmatrix} = \begin{array}{|c|c|c|}\hline y_1(p) & y_2(p) & y_3(p) \\ \hline y_4(p) & y_5(p) & y_6(p) \\ \hline y_7(p) & y_8(p) & y_9(p) \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline \end{array}$$

Uwaga: $y(p)$ powinien być $\approx u(p)$ $(1 \le p \le 2)$.

Zadanie (2)

Cel: Za pomocą metody gradientu

$$E = E(w_{i\,j}, s_{j\,i}) = \frac{1}{2}\sum_{p=1}^{2}\sum_{t=1}^{9}(y_t(p) - u_t(p))^2 \to \text{minimum (lokalne)}.$$

Gradienty

$f'(x) = \frac{\beta e^{-\beta x}}{(1+e^{-\beta x})^2} = \beta f(x)\{1 - f(x)\}$

$$\frac{\partial E}{\partial s_{j\,i}} = \sum_{p=1}^{2}(y_j(p) - u_j(p))f'(\sum_{k=1}^{3} s_{j\,k}x_k(p))x_i(p) \quad (1 \le j \le 9, 1 \le i \le 3)$$

$$\frac{\partial E}{\partial w_{i\,j}} = \sum_{p=1}^{2}\sum_{t=1}^{9}(y_t(p) - u_t(p))f'(\sum_{k=1}^{3} s_{t\,k}x_k(p))s_{t\,i}f'(\sum_{\ell=1}^{10} w_{i\,\ell}u_\ell(p))u_j(p) \quad (1 \le i \le 2, 1 \le j \le 10)$$

(*Por.* Gradienty z LAB (BackProp dla XOR))

Implementować algorytm metody gradientu dla autoenkodera z warunkiem początkowym **podanym w Zadaniu (1)**.
(np. $c = 0.8$, $\varepsilon = 0.0001$, $\beta = 1.0$)
Po zakończeniu iteracji metody gradientu wyświetlić **obrazy** z liczbami $x(p)$ i $y(p)$ $(p = 1, 2)$.
Uwaga: $y(p)$ powinien być $\approx u(p)$ $(1 \le p \le 2)$.

Zadanie (3)

Implementować algorytm metody gradientu dla autoenkodera z warunkiem początkowym **losowanym z przedziału** $[-N, N]$. (np. $N = 0.5 \sim 10$?)
Po zakończeniu iteracji metody gradientu wyświetlić **obrazy** z liczbami $x(p)$ i $y(p)$ $(p = 1, 2)$.
Uwaga: $y(p)$ powinien być $\approx u(p)$ $(1 \le p \le 2)$.

Notacja dla programu. (propozycja)
$u_j(p) \rightsquigarrow u[p][j],\ x_i(p) \rightsquigarrow x[p][i],\ y_j(p) \rightsquigarrow y[p][j]$
$s_{j\,i} \rightsquigarrow s[j][i],\ w_{i\,j} \rightsquigarrow w[i][j]$
$\frac{\partial E}{\partial s_{j\,i}} \rightsquigarrow DE\_s[j][i],\ \frac{\partial E}{\partial w_{i\,j}} \rightsquigarrow DE\_w[i][j]$
$f \rightsquigarrow f,\ f' \rightsquigarrow Df,\ e^{-\beta x} \rightsquigarrow math.exp((-1) * beta * x)$ (Python?)
$c \rightsquigarrow c,\ \varepsilon \rightsquigarrow epsilon,\ \beta \rightsquigarrow beta$