

## Podstawy Sztucznej Inteligencji

### Laboratorium

#### Ćwiczenie 3.

Klasyfikacja wielokryterialna za pomocą jednowarstwowej sieci neuronowej.

## Wstęp.

Sztuczne sieci neuronowe to rodzaj maszyn uczących inspirowanych sposobem działania ludzkiego mózgu, który możemy postrzegać jako układ przetwarzający pewne wejścia (bodźce, sygnały z receptorów) i wytwarzający na ich podstawie pewne wyjścia.

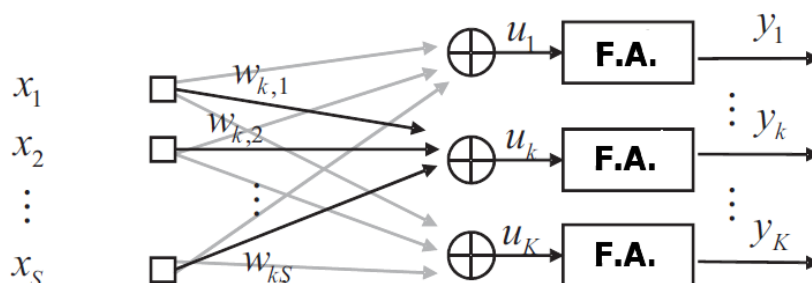
Sztuczne sieci neuronowe, jako uproszczone modele oparte na zasadach działania mózgu, można traktować jako układy typu wejście–wyjście. Sieci neuronowe są zbudowane z pojedynczych neuronów – pojedynczych bardzo prostych procesorów, ułożonych w warstwy. Sygnały są przesyłane do kolejnych warstw, w których zadanie do rozwiązania jest kolejno upraszczane.

W ogólności sieci nie rozwiązują stawianych przed nimi problemów w sposób algorytmiczny, czyli nie sposób ich zaprogramować do rozwiązywania jakiegoś zadania. Sieci neuronowe zdobywają umiejętność rozwiązywania zadań na podstawie przykładów pokazywanych im w trakcie uczenia. Przykładami tymi są zestawy par (wejście-wyjście), zawierające prawidłowe odpowiedzi sieci na pobudzenia.

Niniejsza instrukcja traktować będzie o sieciach złożonych z jednej lub dwóch warstw neuronów, tzw. sieciach MLP (Multilayer Perceptron).

## Jednowarstwowe sieci neuronowe.

W odróżnieniu od instrukcji dotyczącej działania pojedynczego neuronu ze skokową funkcją aktywacji sieć jednowarstwowa składa się z wielu neuronów ( $k$ -neuronów) ułożonych w jednej warstwie. Przedstawia ją rys. 1.



Rys. 1. Jednowarstwowa sieć neuronowa.

Sieć jednowarstwowa to zespół  $k$  - neuronów, przetwarzających sygnały z tych samych wejść.

Symbole na rys. 1 oznaczają:

$[x_1 \ x_2 \ \dots \ x_s]^T$  – wektor  $s$ -wejść,

$$W = [w_1 \ w_2 \ \dots \ w_k] = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{k,1} \\ w_{1,2} & w_{2,2} & \dots & w_{k,2} \\ \vdots & \vdots & & \vdots \\ w_{1,s} & w_{2,s} & \dots & w_{k,s} \end{bmatrix}_{s \times k} \quad \text{– macierz wag neuronów}$$

**F.A.** – funkcje aktywacji neuronów,

$[y_1 \ y_2 \ \dots \ y_k]^T$  – wektor wyjść neuronów.

Uczenie sieci jest procesem dochodzenia wag do wartości optymalnych – zapewniających odpowiednie reakcje sieci na pokazywane jej sygnały wejściowe, czyli prawidłowe rozwiązanie zadań. Wagi sieci jednowarstwowej zostały zapisane w macierzy o wymiarach  $s \times k$ . Z punktu widzenia uczenia sieci jako procesu optymalizacji, wagi sieci jednowarstwowej są punktem w  $s$ -na- $k$ -wymiarowej przestrzeni parametrów, zaś samo uczenie polega na znalezieniu optymalnego punktu w tej przestrzeni, dla którego funkcja celu, zwykle określana jako średnia kwadratów błędów na wyjściach sieci osiągnie najmniejszą wartość. Im bliżej punktu optymalnego rozpoczynane są poszukiwania, czyli im „lepsze” będą wagi początkowe sieci, tym efektywniejszy będzie proces ich uczenia. Najprostszą metodą wyboru wag początkowych jest losowe ustalanie wag. Ich zakres można przyjąć z zakresu  $\langle 0;1 \rangle$  lub  $\langle -0.1;0.1 \rangle$ .

## Symulacja działania sieci.

Z modelu sieci neuronowej przedstawionej na rys. 1 widać, że obliczenie odpowiedzi neuronu  $Y$  na zadany sygnał wejściowy  $X$  następuje w dwóch etapach:

1. Obliczenie pobudzenia neuronu  $u$ , czyli sumy ważonej sygnałów wejściowych z wagami neuronu,
2. Obliczenie wyjścia  $Y$  jako wartości funkcji aktywacji w punkcie  $u$ .

Zatem pobudzenie neuronu jest sumą ważoną wszystkich sygnałów wejściowych, które można obliczyć ze wzoru:

$$u = \sum_{s=1}^S w_s \cdot x_s = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_S \cdot x_S = \mathbf{W}^T \cdot \mathbf{X}$$

jako iloczyn transponowanego wektora wag neuronu  $\mathbf{W}^T$  z wektorem jego wejść  $\mathbf{X}$ .

W przypadku warstwy sieci składającej się z  $K$  neuronów, pobudzenia poszczególnych neuronów zapisuje się w jednym wektorze o wymiarach  $K \times 1$ :

$$\mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_K \end{bmatrix}_{K \times 1}$$

Następnym etapem obliczania wyjścia sieci  $Y$  jest przejście sygnału pobudzenia  $\mathbf{U}$  neuronu przez jego funkcję aktywacji. W podstawowym modelu sztucznego neuronu - perceptronie McCullocha-Pittsa, jest to równoznaczne z przejściem sygnału pobudzenia (czyli ważonej sumy sygnałów wejściowych) przez progową funkcję aktywacji. W sztucznych sieciach z reguły przyjmuje się funkcję sigmoidalną o wzorach:

$$\begin{aligned} y &= f_u(u) = \frac{1}{1 + e^{-\beta u}} \\ y &= f_b(u) = 2 \cdot f_u(u) - 1 = \tanh(\beta u) \end{aligned}$$

Ważnym warunkiem wyboru postaci funkcji aktywacji jest ciągłość funkcji w każdym punkcie poszukiwanej przestrzeni rozwiązań oraz prosta postać pierwszej pochodnej tejże funkcji.

$$f'_u(u) = \dots = \beta \cdot y \cdot (1 - y) \quad - \text{I pochodna funkcji unipolarnej danej wzorem}$$

$$f'_b(u) = \dots = \beta \cdot (1 - y^2) \quad - \text{I pochodna funkcji bipolarnej}$$

Obliczenie wyjścia warstwy sieci neuronowej polega na obliczeniu wyjść wszystkich jej neuronów. Zakładając, że wszystkie neurony w warstwie posiadają taką samą funkcję aktywacji – sigmoidalną, unipolarną wyjścia wszystkich neuronów danej warstwy sieci zostaną zebrane w jednym wektorze:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix}_{K \times 1} = \begin{bmatrix} f(u_1) \\ f(u_2) \\ \vdots \\ f(u_K) \end{bmatrix}$$

## Uczenie jednowarstwowej sieci neuronowej.

Aby rozpocząć uczenie sieci należy zdefiniować odpowiednio zagadnienie do rozwiązania przez sieć. W poprzedniej instrukcji pojedynczy neuron był w stanie rozwiązać zagadnienie klasyfikacji dwóch zbiorów na płaszczyźnie kartezjańskiej. Obecnie zadaniem do rozwiązania będzie rozpoznawanie na podstawie danych wektora wejściowego i nauczyciela, czy mamy z jakim rodzajem zwierzęcia mamy do czynienia?

Na podstawie pięciu informacji:

1. ile zwierzę ma nóg,
2. czy żyje w wodzie,
3. czy umie latać,
4. czy ma pióra,
5. czy jest jajorodne,

sieć rozpoznaje, czy jest to ssak, ptak, czy ryba.

Informacje te podane zostaną na pięć wejść jednowarstwowej sieci neuronowej, złożonej z trzech neuronów. Każdy z nich będzie „odpowiedzialny” za jedną gromadę zwierząt, tzn. pierwszy ma się nauczyć rozpoznawać ssaki, drugi – ptaki, a trzeci – ryby.

Dobór poszczególnych cech wzorcowych przedstawicieli każdej klasy jest w oryginale ([5]) argumentowany w następujący sposób (porównaj z wartościami wektora P):

- ssak – ma 4 nogi; czasem żyje w wodzie, ale nie jest to dla niego typowe (foka, delfin), czasem umie latać (nietoperz), ale również nie jest to dla niego typowe; nie ma piór; jest żyworodny i jest to dość ważne;
- ptak – ma 2 nogi; nie żyje w wodzie (co najwyżej na wodzie); zwykle umie latać i jest to bardzo ważna jego cecha; jak również ta, że ma pióra; oraz że jest jajorodny;
- ryba – nie ma nóg (płetwy się nie liczą); żyje w wodzie i jest to jej najważniejsza cecha; nie umie latać (choć niektóre próbują); nie jest pokryta piórami; jest zwykle jajorodna, ale nie jest to aż tak ważne, jak u ptaków.

Do tego celu należy zdefiniować wektor wejściowy **P** o postaci:

Zwierzę 1	Zwierzę 2	Zwierzę 3	Opis wejścia
4	2	-1	Liczba nóg
0.01	-1	3.5	Czy żyje w wodzie?
0.01	2	0.01	Czy umie latać?
-1	2.5	-2	Czy ma pióra?
-1.5	2	1.5	Czy jest jajorodne?

Oraz wektor nauczyciela **T**:

1	0	0	Ssak
0	1	0	Ptāk
0	0	1	Ryba

Można zauważyć, że jedynie pierwsza cecha ma charakter ilościowy. Nie jest ona tu w żaden sposób skalowana, ani normalizowana – nie jest to konieczne przy tak prostym zadaniu, jednak w praktyce bardzo często stosowane. O wiele łatwiejszym zadaniem dla sieci jest bowiem analiza wpływu poszczególnych wejść na żądany wynik, gdy wszystkie wejścia przyjmują wartości z podobnego zakresu, niż gdy jedno wejście ma zakres np. od 0 do 1, a drugie od 1000 do 5000. Kolejne cztery wejścia neuronu są typu logicznego. Również i one zostały w oryginale potraktowane dosyć swobodnie. Wartości logiczne koduje się zwykle albo jako 0 i 1, albo -1 i 1 (odpowiednio: fałsz i prawda). Tu autor zdecydował się na podkreślenie znaczenia niektórych cech przez wyjście poza ten zakres.

W prezentowanym przykładzie wykorzystano trzy neurony wyjściowe do klasyfikacji sygnałów wejściowych na trzy klasy. Przedstawiony sposób kodowania klas nazywany jest metodą „jeden z N” i jest jednym z kilku możliwych. Do oznaczenia trzech klas wystarczyłyby bowiem dwa wyjścia. Moglibyśmy próbować nauczyć dwa neurony reagowania następującymi wartościami wyjść: 01 dla ssaków, 10 dla ptaków, 11 dla ryb, zaś 00 oznaczałoby dodatkową klasę – „brak decyzji”. Zaproponowany sposób kodowania „jeden z N” jest jednak łatwiejszy do nauczania przez sieć. Jest również łatwiejszy w interpretacji – wielkości poszczególnych wyjść można bowiem odczytywać jako stopień „przekonania” sieci do danej klasy, czyli prawdopodobieństwo rozpoznania danej klasy. Za rozpoznaną klasę można uznać tę przypisaną do neuronu, który najsilniej zareagował na sygnał wejściowy.

Gdy wartości na wyjściach żadnego neuronu nie przekraczają założonego progu, rezultatem działania sieci może być „brak decyzji”. Podobny wynik możemy przyjąć w przypadku, gdy kandydatów do zwycięstwa jest dwóch lub więcej (wysokie poziomy wyjść na kilku neuronach i mała różnica między nimi). Postępowanie takie jest często stosowane w zadaniach klasyfikacji, szczególnie w takich zastosowaniach (np. medycznych), w których złe rozpoznanie jest gorsze w skutkach niż wstrzymanie się od decyzji.

## Zadanie.

Napisz funkcję, która będzie uczyć sieć jednowarstwową o danej macierzy wag  $W$  przez zadaną liczbę epok  $n$  na ciągu uczącym podanym w postaci macierzy przykładów  $P$  (wejścia) i  $T$  (żądane wyjścia). Sieć zbudowana jest z neuronów o sigmoidalnej funkcji aktywacji. Wynikiem działania funkcji powinna być macierz wag nauczonej sieci. Jako parametry wejściowe wprowadzić macierz  $P$  postaci:

```
P =      4.0000      2.0000     -1.0000
        0.0100     -1.0000      3.5000
        0.0100      2.0000      0.0100
       -1.0000      2.5000     -2.0000
       -1.5000      2.0000      1.5000
```

Wektor nauczyciela:  $T$ :

```
T =      1      0      0
        0      1      0
        0      0      1
```

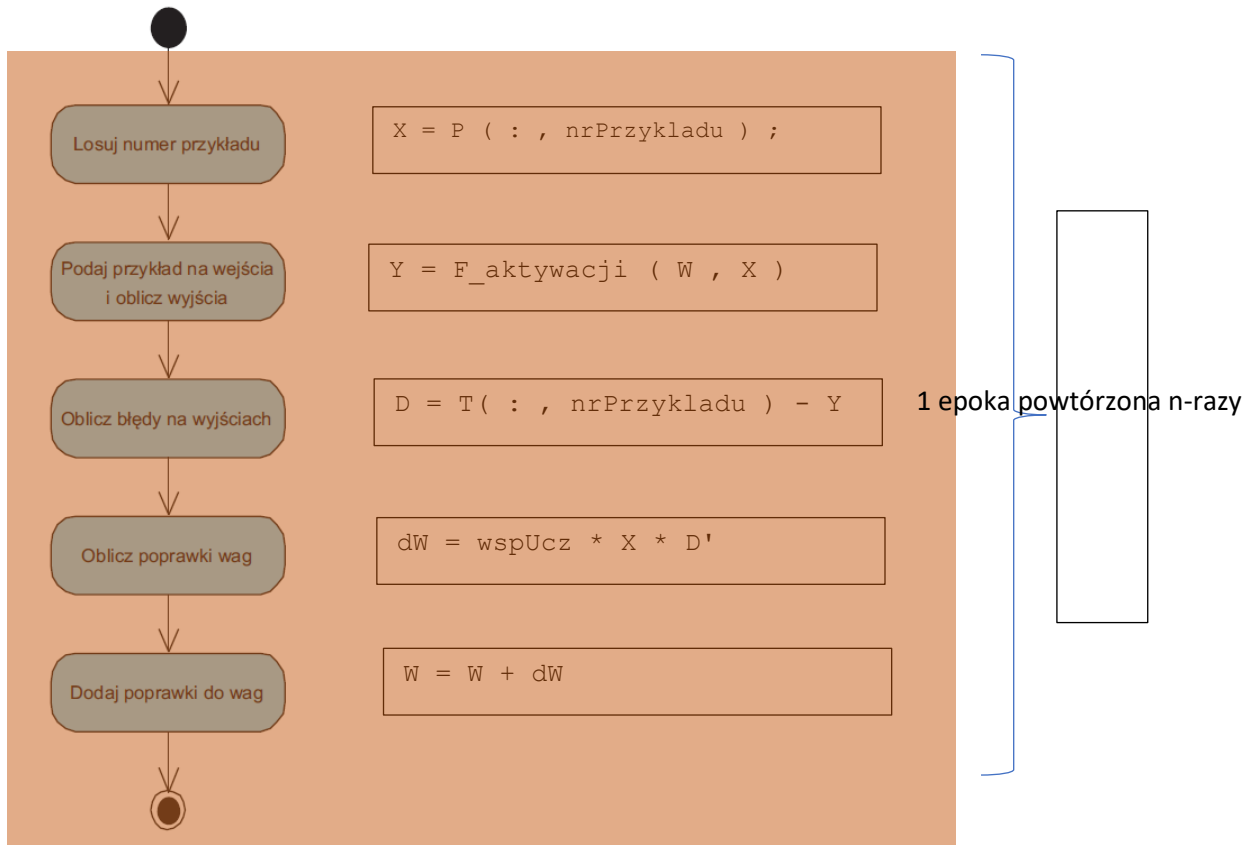
Macierz wylosowanych wstępnie wag  $W_{prz}$ :

```
Wprz =    0.0629   -0.0805   -0.0685
          0.0812   -0.0443    0.0941
         -0.0746    0.0094    0.0914
          0.0827    0.0915   -0.0029
          0.0265    0.0930    0.0601
```

Oraz liczbę epok  $n$ . Epoki to iteracje, w których wprowadzamy do sieci wylosowany numer kolumny macierzy  $P$  (czyli dane dla zwierzęcia, które klasyfikujemy) i uczymy go sieć. Jeśli  $n=10$ , to statystycznie każda kolumna macierzy  $P$  będzie pokazana sieci 3 lub więcej razy, dzięki czemu sieć nauczy się lepiej każdego wzorca.

### Rozwiązanie.

Uczenie sieci polega na wielokrotnym „pokazywaniu” jej kolejnych przykładów z ciągu uczącego. Podczas każdego takiego pokazu wyznaczana jest odpowiedź sieci, po czym jest ona porównywana z żądaną odpowiedzią dla danego przykładu, zawartą w ciągu uczącym. Na podstawie obliczonej różnicy, czyli błędu popełnionego przez sieć, ustala się następnie poprawki wszystkich wag sieci, zgodnie z przyjętym algorytmem uczenia. Poprawki te dodaje się do wag po to, aby w następnych krokach, gdy sieci zostanie przedstawiony ten sam przykład, jej odpowiedź była bliższa żądanej odpowiedzi z ciągu uczącego.



Rys. 2. Pojedynczy krok uczenia sieci jednowarstwowej.

Jako funkcję aktywacji przyjmij funkcję unipolarną o współczynniku  $\beta=5$ . Współczynnik uczenia przyjmij wstępnie jako  $\eta = 0.1$ .