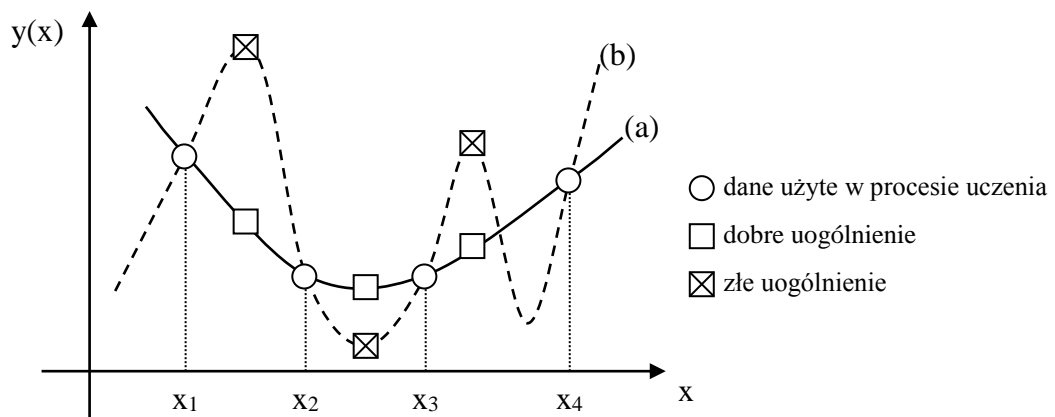


4. UCZENIE SIECI NEURONOWEJ

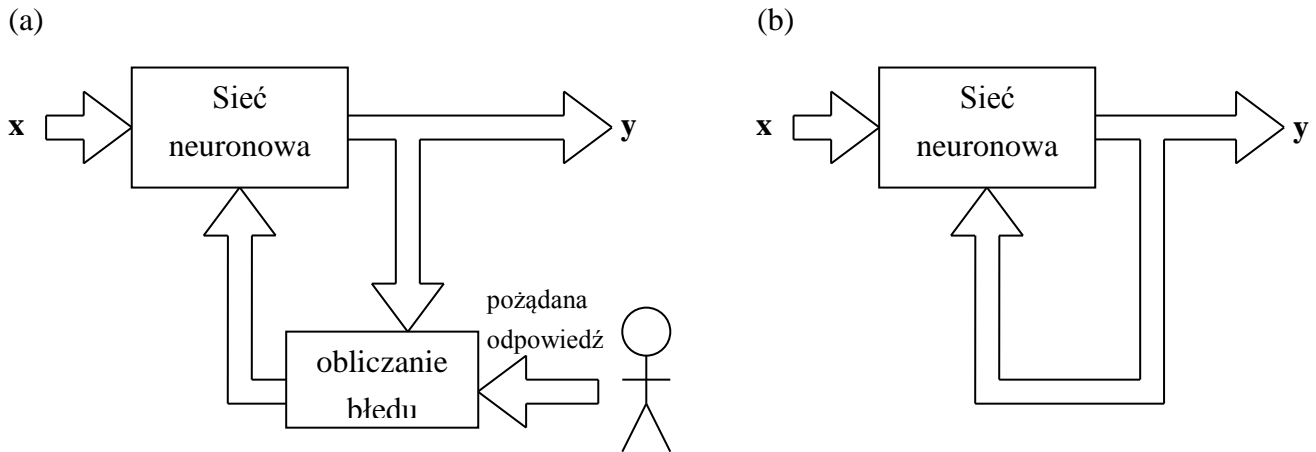
Do realizacji dowolnego zadania konieczne jest nauczenie sieci prawidłowego przetwarzania informacji. W tym celu określa się zbiór przykładowych danych wejściowych i odpowiadających im, pożądaných wartości wyjściowych (tzw. par uczących), a następnie dobiera się wartości wag w taki sposób, aby sieć uzyskiwała założone wartości wyjściowe dla wszystkich wartości wejściowych. Zadanie to można opisać jako przybliżenie pewnej ciągłej funkcji $h(\mathbf{x})$ za pomocą funkcji $H(\mathbf{w}, \mathbf{x})$, gdzie $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ jest wektorem wartości wejściowych, natomiast $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ jest wektorem wag. Stąd problem uczenia można określić jako poszukiwanie wektora \mathbf{w} zapewniającego najlepszą aproksymację funkcji $h(\mathbf{x})$ na podstawie wyników pewnej serii eksperymentów (zbioru par uczących).

Ważnym elementem procesu uczenia jest odpowiedni dobór funkcji aproksymującej $H(\mathbf{w}, \mathbf{x})$. Jedną z cech sieci neuronowych jest zdolność uogólniania, czyli umiejętność rozsądnej interpolacji wyników dla nieznaných wartości wyjściowych. W wyniku nieodpowiedniego wyboru funkcji $H(\mathbf{w}, \mathbf{x})$ sieć może utracić zdolność uogólniania. Rysunek 1. pokazuje przykład dobrego (krzywa a) i złego (krzywa b) uogólniania nowych wartości leżących pomiędzy punktami używanymi jako dane uczące (punkty x_1, \dots, x_4).



Rys. 1. Przykład dobrego i złego uogólnienia

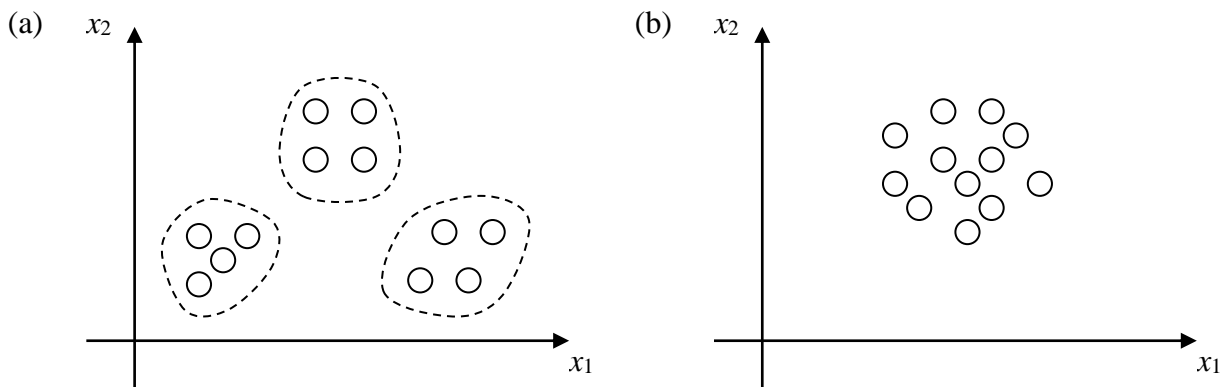
Rozróżnia się dwa rodzaje uczenia: **z nauczycielem** i **bez nauczyciela (samouczenie)**. Obydwa sposoby zostały przedstawione schematycznie na rys. 2.



Rys. 2. *Uczenie z nauczycielem (a) i bez nauczyciela (b)*

W przypadku uczenia z nauczycielem sieć przetwarza dane wejściowe ze zbioru uczącego, natomiast nauczyciel podpowiada prawidłową odpowiedź. Różnica pomiędzy odpowiedzią uzyskaną i pożądaną jest miarą błędów, która zostaje wykorzystana do skorygowania wartości wag sieci. Proces uczenia sprowadza się więc do minimalizacji wartości błędów w przestrzeni wag.

Podczas uczenia bez nauczyciela pożądana odpowiedź nie jest znana, sieć musi więc sama wykrywać wszystkie zależności, linie podziału i inne charakterystyki danych wejściowych. Samouczenie może pozwolić na znalezienie granic pomiędzy klasami obiektów wyraźnie od siebie oddzielonych (rys. 3a), sieć musi jednak posiadać odpowiedni mechanizm autoadaptacji. Przykładem takiego mechanizmu jest zasada przypisywania obiektu do tego skupiska, od centrum którego leży najbliżej. Uczenie bez nauczyciela nie jest możliwe w każdym przypadku. Sieć nie będzie mogła dokonać prawidłowej klasyfikacji obiektów, jeżeli nie będą one tworzyły wyraźnie odseparowanych skupisk. Na rys. 3b pokazany został zbiór obiektów, których nie można sklasyfikować bez dokładnej znajomości zasad przynależności do konkretnej klasy.



Rys. 3. *Przykład obiektów w skupiskach (a) i bez wyraźnych skupisk (b)*

W dalszej części tego opracowania zostaną przedstawione metody uczenia z nauczycielem.



4.1. Metoda uczenia pojedynczego neuronu

Proces uczenia sieci (a więc również pojedynczego neuronu) sprowadza się do modyfikacji wektora wag w taki sposób, aby zwracane odpowiedzi były zgodne z odpowiedziami oczekiwanymi dla wszystkich wartości wejściowych.

W procesie uczenia z nauczycielem sieci przedstawiane są przykładowe wartości wejściowe x_1, x_2, \dots, x_p , nazywane ciągiem uczącym, oraz prawidłowe odpowiedzi (dostarczane przez nauczyciela). Sieć zmienia swoje wagi jeżeli uzyskiwana odpowiedź nie jest zgodna z odpowiedzią prawidłową. Analizę przebiegu procesu uczenia pojedynczego neuronu można przeprowadzić posługując się pojęciem **przestrzeni wag** czyli przestrzeni, której współrzędnymi są wszystkie wagi tego neuronu. Koncepcja przestrzeni wag jest podobna do koncepcji przestrzeni wejściowej przedstawionej w poprzednim opracowaniu (Laboratorium 02), podobne są również jej własności. Dla dowolnego wektora wejściowego można wyróżnić obszar, w którym potencjał membranowy jest większy od zera i obszar, w którym jest mniejszy od zera. Granica pomiędzy tymi obszarami przebiega przez początek układu współrzędnych, wzdłuż hiperpłaszczyzny wyznaczonej przez punkty, dla których potencjał membranowy jest równy zero. Korzystając z własności iloczynu skalarnego można pokazać, że wektor wejściowy jest prostopadły do hiperpłaszczyzny separującej i wskazuje na ten obszar, w którym potencjał membranowy jest większy od zera.

Przykład 1.

Dany jest neuron bez biasu o dwóch wejściach, z bipolarną funkcją aktywacji. Na wejściu podany został wektor $\mathbf{x}=[1, 2]^T$, należy przeprowadzić analizę przestrzeni wag tego neuronu.

Neuron ma dwie wagi (każda odpowiada jednemu wejściu), nie ma biasu, więc wymiar przestrzeni wag wynosi 2. Przyjmując wartości sygnału wejściowego jako $\mathbf{x}=[1, 2]^T$ całą przestrzeń można podzielić na dwie części: pierwsza zawiera wagi dla których potencjał membranowy jest większy od zera (neuron zwraca 1), druga wagi dla których potencjał membranowy jest mniejszy od zera (neuron zwraca -1). Granica pomiędzy tymi obszarami przebiega w miejscu, gdzie potencjał membranowy jest równy zero:

$$\varphi = w_1 x_1 + w_2 x_2 = w_1 \cdot 1 + w_2 \cdot 2 = w_1 + 2w_2 = 0, \quad (4)$$

stąd:

$$w_2 = -\frac{1}{2} w_1. \quad (5)$$

Potencjał membranowy można wyrazić jako iloczyn skalarny wektora wag, można więc zapisać:

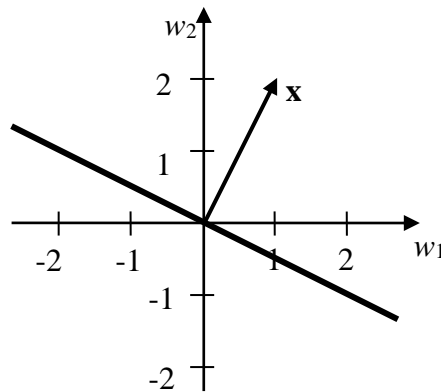
$$\varphi = \langle \mathbf{w}, \mathbf{x} \rangle = |\mathbf{w}| |\mathbf{x}| \cos(\alpha), \quad (6)$$

gdzie α - kąt pomiędzy wektorem \mathbf{w} i \mathbf{x} .

Z własności iloczynu skalarnego wynika, że potencjał membranowy jest równy zero gdy wektor wejściowy i wektor wag są prostopadłe, stąd wektor wejściowy musi być prostopadły do powierzchni dzielącej przestrzeń wag. Dodatkowo można zauważyć, że potencjał membranowy będzie większy od zera dla tych wektorów z przestrzeni wag, które tworzą z wektorem wejściowym kąt z przedziału



$(-90^\circ, 90^\circ)$. Wynika stąd, że wektor wejściowy znajduje się w obszarze zawierającym wagi, dla których potencjał membranowy jest większy od zera (neuron zwraca wartość 1). Rysunek 4. przedstawia przestrzeń wag omawianego neuronu z zaznaczonym wektorem wejściowym i granicą obszarów wag.



Rys. 4. Przestrzeń wag



Korzystając z przedstawionych własności można w łatwy sposób wyznaczyć wartości wag, które zapewnią prawidłowe funkcjonowanie neuronu. Każdy wektor wejściowy wprowadza do przestrzeni wag hiperpłaszczyznę oddzielającą wagi dla których $\varphi > 0$ od wag dla których $\varphi < 0$. Po zaznaczeniu wszystkich hiperpłaszczyzn odpowiadających wektorom wejściowym należy wybrać ten obszar przestrzeni wag, który zapewnia $\varphi > 0$ dla obiektów należących do pierwszej i $\varphi < 0$ dla obiektów należących do drugiej kategorii, czyli prawidłową klasyfikację.

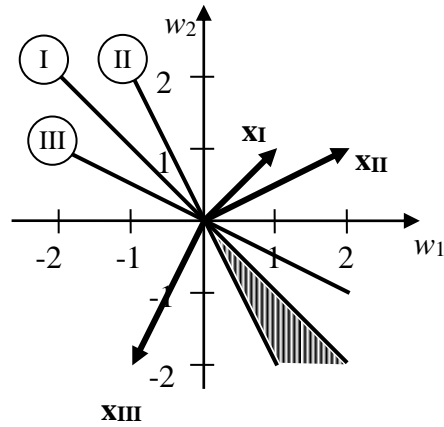
Przykład 2.

Dany jest neuron bez biasu o dwóch wejściach i bipolarnej funkcji aktywacji. Należy dobrać wagi neuronu w taki sposób, aby klasyfikował punkty: $\mathbf{x}_I = [1, 1]^T$, $\mathbf{x}_{II} = [2, 1]^T$, $\mathbf{x}_{III} = [-1, -2]^T$, przypisując \mathbf{x}_{II} i \mathbf{x}_{III} do pierwszej klasy (wartość wyjściowa 1), a \mathbf{x}_I do drugiej klasy (wartość wyjściowa -1).

Hiperpłaszczyzny dzielące przestrzeń wag dla kolejnych punktów są opisane zależnościami:

$$\begin{aligned} \text{punkt } \mathbf{x}_I : & \quad w_2 = -w_1 \\ \text{punkt } \mathbf{x}_{II} : & \quad w_2 = -2w_1 \\ \text{punkt } \mathbf{x}_{III} : & \quad w_2 = -0.5w_1 \end{aligned}$$

Przestrzeń wag z zaznaczonymi wektorami wejściowymi i odpowiadającymi im hiperpłaszczyznami przedstawia rys. 5.

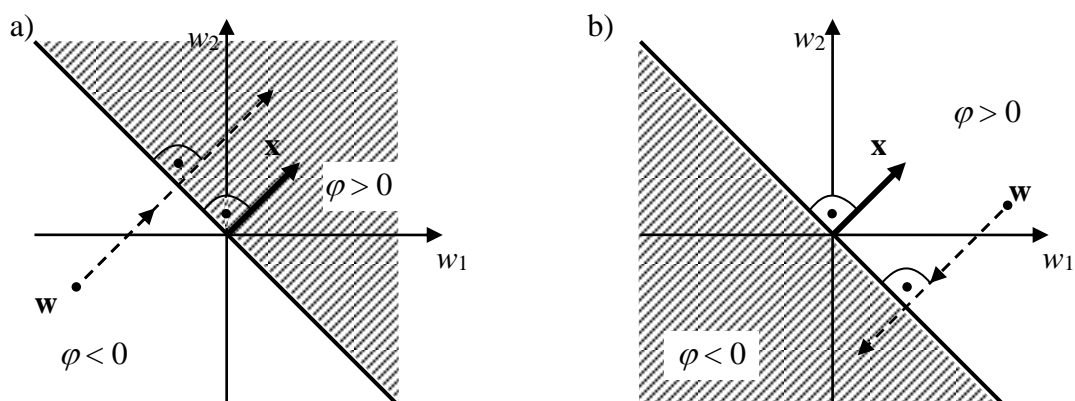


Rys. 5. Wektory wejściowe w przestrzeni wag

Analiza rysunku pozwala określić obszar rozwiązań, czyli obszar przestrzeni, który zawiera wagi zapewniające prawidłowe funkcjonowanie neuronu – musi znajdować się on ponad hiperpłaszczyznami odpowiadającymi wektorom x_{II} i x_{III} oraz poniżej hiperpłaszczyzny wyznaczonej przez wektor x_I (na rysunku został on zakreślony). ■

Stosując koncepcję przestrzeni wag można sformułować podstawowy algorytm poszukujący właściwych wartości wag na drodze iteracyjnej. Podstawowa zasada takiego algorytmu sprowadza się do wyznaczenia wartości wyjściowej neuronu dla wektora z ciągu uczącego i dokonania korekty wag jeżeli jest ona niezgodna z wartością żadaną. Wielokrotne powtarzanie tego procesu dla kolejnych wartości uczących doprowadzi do znalezienia wag znajdujących się w obszarze rozwiązań, czyli zapewniających prawidłową klasyfikację wszystkich elementów ze zbioru uczącego (o ile jest to możliwe).

Realizacja opisanego powyżej algorytmu wymaga sformułowania zasady, zgodnie z którą w każdym kroku będzie przeprowadzana modyfikacja wag. W przypadku neuronu o bipolarnej funkcji aktywacji możliwe są dwie sytuacje, w których wagi nie są poprawne i wymagają korekty: dla danego wektora wejściowego neuron powinien zwracać 1 ($\varphi > 0$), natomiast aktualne wagi znajdują się na półpłaszczyźnie $\varphi < 0$, lub dla danego wektora wejściowego neuron powinien zwracać -1 ($\varphi < 0$), natomiast aktualne wagi znajdują się na półpłaszczyźnie $\varphi > 0$ (rys. 6).



Rys. 6. Modyfikacja wag a) wartość wejściowa x należy do klasy 1,
b) wartość wejściowa x należy do klasy -1

Modyfikacja wag powinna spowodować przesunięcie punktu \mathbf{w} do obszaru rozwiązań. Jak widać na rys. 6. najkrótsza droga od obszaru rozwiązań prowadzi w kierunku prostopadłym do hiperpłaszczyzny dzielącej przestrzeń wag, a więc wyznaczonym przez wektor wejściowy. Jeżeli \mathbf{x} należy do klasy 1, a punkt \mathbf{w} znajduje się po ujemnej stronie hiperpłaszczyzny (rys. 6a), to w celu korekty wartości wag należy przesunąć go w kierunku zgodnym ze zwrotem wektora \mathbf{x} , czyli:

$$\tilde{\mathbf{w}} = \mathbf{w} + c\mathbf{x}, \quad (7)$$

gdzie: $\tilde{\mathbf{w}}$ - nowe przybliżenie wektora wag; c - współczynnik korekcji, dodatni skalar, który określa wielkość zmian w pojedynczym kroku.

Jeżeli \mathbf{x} należy do klasy -1, a punkt \mathbf{w} znajduje się po dodatniej stronie hiperpłaszczyzny (rys. 6b), to w celu korekty wartości wag należy przesunąć go w kierunku przeciwnym do zwrotu wektora \mathbf{x} , czyli:

$$\tilde{\mathbf{w}} = \mathbf{w} - c\mathbf{x}, \quad (8)$$

Powtarzając wielokrotnie operację opisaną zależnością (7) lub (8) dla każdej wartości ze zbioru uczącego można wyznaczyć wartości wag zapewniające założoną klasyfikację.

Jeżeli neuron ma bipolarną funkcję skoku jako funkcję aktywacji zależności (7), (8) można przedstawić w postaci:

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \frac{1}{2}c(d^k - y^k)\mathbf{x}^k, \quad (9)$$

gdzie: \mathbf{w}^k - aktualne przybliżenie wektora wag; \mathbf{w}^{k+1} - nowe przybliżenie wektora wag; \mathbf{x}^k - k -ty wektor ze zbioru par uczących; $y^k = \text{sign}(\mathbf{w}^k \mathbf{x}^k)$ - rzeczywista odpowiedź neuronu na sygnał \mathbf{x}^k ; d^k - oczekiwana odpowiedź neuronu na sygnał \mathbf{x}^k ; c - współczynnik korekcji, dodatni skalar, który określa wielkość zmian w pojedynczym kroku.

W przypadku neuronu z unipolarną funkcją skoku jako funkcją aktywacji odpowiednia zależność przyjmuje postać:

$$\mathbf{w}^{k+1} = \mathbf{w}^k + c(d^k - y^k)\mathbf{x}^k, \quad (10)$$

Przykład 3.

Dany jest neuron bez biasu o dwóch wejściach i bipolarnej funkcji aktywacji. Należy dobrać wagi neuronu w taki sposób, aby klasyfikował punkty: $\mathbf{x}_I = [1, 1]^T$, $\mathbf{x}_{II} = [2, 1]^T$, $\mathbf{x}_{III} = [-1, -2]^T$, przypisując \mathbf{x}_{II} i \mathbf{x}_{III} do pierwszej klasy (wartość wyjściowa 1), a \mathbf{x}_I do drugiej klasy (wartość wyjściowa -1). Należy przeprowadzić analizę procesu uczenia neuronu. Wartość współczynnika korekcji c wynosi 0.25, początkowe przybliżenie wektora wag $\mathbf{w}^0 = [1.5, 0.5]^T$.



Na podstawie zależności (9) wyznacza się kolejne przybliżenia wag:

Krok 1.

$$y_I = \text{sign}\left(\begin{bmatrix} 1.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \text{sign}(2) = 1, \quad d_I = -1, \quad d_I - y_I = -2$$

$$\mathbf{w}^1 = \mathbf{w}^0 + \frac{1}{2} 0.25(d_I - y_I) \mathbf{x}_I = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} - 0.25 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 0.25 \end{bmatrix}$$

Krok 2.

$$y_{II} = \text{sign}\left(\begin{bmatrix} 1.25 & 0.25 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{sign}(2.75) = 1, \quad d_{II} = 1, \quad d_{II} - y_{II} = 0$$

$$\mathbf{w}^2 = \mathbf{w}^1 + \frac{1}{2} 0.25(d_{II} - y_{II}) \mathbf{x}_{II} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \mathbf{w}^1$$

Uwaga: Rzeczywista odpowiedź neuronu była zgodna z oczekiwaną, korekta wag nie została przeprowadzona.

Krok 3.

$$y_{III} = \text{sign}\left(\begin{bmatrix} 1.25 & 0.25 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix}\right) = \text{sign}(-1.75) = -1, \quad d_{III} = 1, \quad d_{III} - y_{III} = 2$$

$$\mathbf{w}^3 = \mathbf{w}^2 + \frac{1}{2} 0.25(d_{III} - y_{III}) \mathbf{x}_{III} = \begin{bmatrix} 1.25 \\ 0.25 \end{bmatrix} + 0.25 \begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.25 \end{bmatrix}$$

Krok 4. $y_I = \text{sign}(0.75) = 1, \quad d_I = -1, \quad d_I - y_I = -2, \quad \mathbf{w}^4 = \mathbf{w}^3 + \frac{1}{2} 0.25(d_I - y_I) \mathbf{x}_I = \begin{bmatrix} 0.75 \\ -0.5 \end{bmatrix}$

Uwaga: W krokach 1–3 zostały użyte wszystkie wektory ze zbioru uczącego, kolejne etapy algorytmu wykorzystują ponownie te same wzorce.

Krok 5. $y_{II} = \text{sign}(1) = 1, \quad d_{II} = 1, \quad d_{II} - y_{II} = 0, \quad \mathbf{w}^5 = \mathbf{w}^4$

Krok 6. $y_{III} = \text{sign}(0.25) = 1, \quad d_{III} = 1, \quad d_{III} - y_{III} = 0, \quad \mathbf{w}^6 = \mathbf{w}^5$

Krok 7. $y_I = \text{sign}(0.25) = 1, \quad d_I = -1, \quad d_I - y_I = -2, \quad \mathbf{w}^7 = \mathbf{w}^6 + \frac{1}{2} 0.5(d_I - y_I) \mathbf{x}_I = \begin{bmatrix} 0.5 \\ -0.75 \end{bmatrix}$

Krok 8. $y_{II} = \text{sign}(0.25) = 1, \quad d_{II} = 1, \quad d_{II} - y_{II} = 0, \quad \mathbf{w}^8 = \mathbf{w}^7$

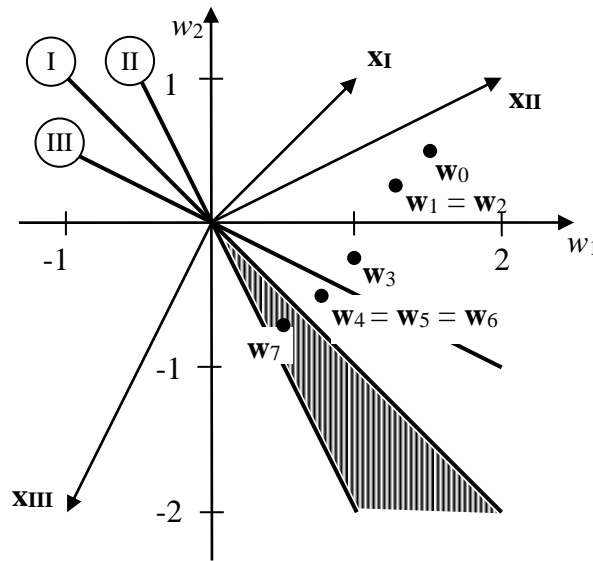
Krok 9. $y_{III} = \text{sign}(1) = 1, \quad d_{III} = 1, \quad d_{III} - y_{III} = 0, \quad \mathbf{w}^9 = \mathbf{w}^8$

Krok 10. $y_I = \text{sign}(-0.25) = -1, \quad d_I = -1, \quad d_I - y_I = 0, \quad \mathbf{w}^{10} = \mathbf{w}^9$

Uwaga: Neuron udziela poprawnych odpowiedzi dla wszystkich ciągów uczących, proces uczenia został zakończony.



Na rysunku 7. zostały przedstawione kolejne przybliżenia wag uzyskiwane w poszczególnych krokach algorytmu.



Rys. 7. Kolejne przybliżenia wag uczonego neuronu



4.2. Reguły uczenia sieci neuronowych

Ogólna zasada uczenia sieci neuronowych mówi, że wektor wag rośnie proporcjonalnie do iloczynu sygnałów wejściowego i uczącego. W ogólnym przypadku sygnał uczący jest funkcją wektora wag i wartości wejściowej, a w pewnych przypadkach również wartości oczekiwanej, określanej przez nauczyciela. Podstawowa zasada uczenia sieci jest więc zgodna z koncepcją uczenia pojedynczego neuronu przedstawianą w punkcie poprzednim.

4.3.1. Reguła perceptronu (perceptronowa)

Reguła ta dotyczy nauki z nauczycielem, sieci z neuronami dyskretnymi (z uni- lub bipolarną funkcją aktywacji). Sygnał uczący jest różnicą pomiędzy odpowiedzią pożądaną, a rzeczywistą:

$$r = d_i - y_i, \tag{11}$$

gdzie: d_i - pożądana odpowiedź dla i -tej wartości uczącej, y_i - rzeczywista odpowiedź dla i -tej wartości uczącej.

Korekcja wag odbywa się zgodnie z zależnością:

$$\Delta \mathbf{w}_i = c(d_i - y_i)\mathbf{x} \tag{12}$$

Korekta wag jest przeprowadzana tylko wtedy, gdy odwzorowanie wykonane przez neuron jest błędne (błąd sieci jest różny od zera). Początkowe wartości wag mogą być dowolne.



4.3.2. Reguła delta

Reguła dotyczy nauki z nauczycielem sieci z ciągłymi funkcjami aktywacji. Sygnał uczący (nazywany w tej regule sygnałem delta) jest definiowany jako:

$$r = \delta = (d_i - f(\mathbf{w}_i^T \mathbf{x}))f'(\mathbf{w}_i^T \mathbf{x}), \quad (13)$$

gdzie: $f(\cdot)$ - funkcja aktywacji, $f'(\cdot)$ - pochodna funkcji aktywacji.

Reguła opiera się na minimalizacji kwadratowego kryterium błędu, co prowadzi do zależności opisującej korektę wag w postaci:

$$\delta \mathbf{w}_i = c(d_i - y_i)f'(\mathbf{w}_i^T \mathbf{x}), \quad (14)$$

gdzie: c jest dowolnie przyjętą stałą.

Reguła delta stanowi odpowiednik reguły perceptronu dla neuronów o ciągłej funkcji aktywacji. Zależność (14) dotyczy sieci jednowarstwowych, ale może być również uogólniona na przypadek sieci wielowarstwowych. Początkowe wartości wag mogą być dowolne.

4.3.3. Reguła Widrowa-Hoffa

Reguła dotyczy nauki z nauczycielem sieci o dowolnych funkcjach aktywacji. Stanowi ona szczególnie przypadek reguły delta. Korekta wag jest obliczana w wyniku minimalizacji średniego błędu kwadratowego pomiędzy pożądaną odpowiedzią, a pobudzeniem. Sygnał uczący jest zdefiniowany jako:

$$r = d_i - \mathbf{w}_i^T \mathbf{x}. \quad (15)$$

Korekta wektora wag ma postać:

$$\Delta \mathbf{w}_i = c(d_i - \mathbf{w}_i^T \mathbf{x})\mathbf{x}. \quad (12)$$

Wagi początkowe mogą mieć dowolne wartości.

Zadania

Zad. 1. Przeanalizuj rozwiązanie zadania przedstawionego w przykładzie 2. dla neuronu o unipolarnej funkcji aktywacji.

Zad. 2. Dany jest neuron bez biasu o dwóch wejściach i bipolarnej funkcji aktywacji. Zadaniem neuronu jest klasyfikacja czterech punktów: $\mathbf{x}_I = [1, 1]^T$, $\mathbf{x}_{II} = [-1, 1]^T$, $\mathbf{x}_{III} = [1, -1]^T$, $\mathbf{x}_{IV} = [-1, -1]^T$. Punkty \mathbf{x}_I i \mathbf{x}_{IV} mają być przypisane do klasy 1, \mathbf{x}_{II} i \mathbf{x}_{III} do klasy -1. Wyznacz przestrzeń wag, hiperpłaszczyzny związane z wartościami wejściowymi i obszar rozwiązań. Wyjaśnij uzyskane wyniki.

Zad. 3. Dany jest neuron bez biasu o dwóch wejściach i unipolarnej funkcji aktywacji. Zadaniem neuronu jest klasyfikacja dwóch punktów: $\mathbf{x}_I = [2, 1]^T$, $\mathbf{x}_{II} = [-2, 1]^T$. Punkt \mathbf{x}_I ma być przypisany do klasy pierwszej (wartość wyjściowa 1), punkt \mathbf{x}_{II} do klasy drugiej (wartość wyjściowa 0). Narysuj przestrzeń wag i obszar rozwiązań. Przeprowadź analizę działania algorytmu uczenia przedstawionego w punkcji drugim, przyjmując $\mathbf{w}^0 = [-3, 1]^T$ i $c=1$.

