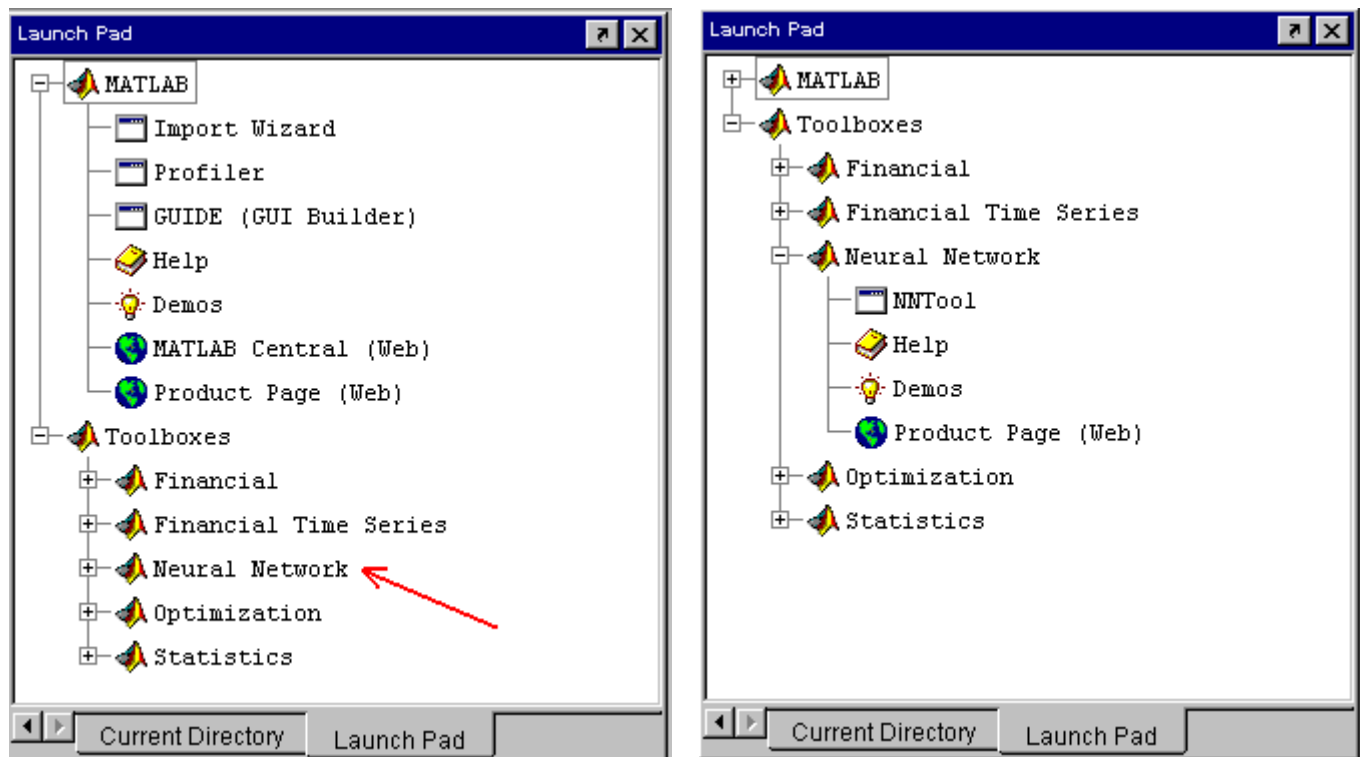


1. WPROWADZENIE – PROGRAMY DEMONSTRACYJNE

Pakiet Matlab może zawierać dodatkową bibliotekę *Neural Network* zawierającą zestaw funkcji pozwalających na budowę i badanie sztucznych sieci neuronowych. Biblioteka ta jest dostępna tylko w przypadku, gdy użytkownik wykupił odpowiednią licencję.

Obecność biblioteki można sprawdzić na kilka sposobów:

- można sprawdzić zawartość podkatalogu Matlab: **toolbox**
(jeżeli biblioteka jest zainstalowana w podkatalogu tym znajduje się katalog: **nnet**),
- można wywołać w oknie **Command Window** polecenie **ver**
(jeżeli biblioteka jest zainstalowana wyświetlona zostanie między innymi informacja o wersji biblioteki, np.: „*Neural Network Toolbox Version 4.0.2 (R13)*”),
- można przejrzeć zawartość okna **LaunchPad** (jeżeli jest niewidoczne, można je wyświetlić wybierając opcję menu głównego: **View|LaunchPad**)
(jeżeli biblioteka jest zainstalowana w oknie tym widoczna jest pozycja: **Neural Network**).



Rys. 1. Okno Launch Pad z widoczną biblioteką Neural Network.

Biblioteka ta zawiera szereg programów demonstracyjnych pokazujących możliwości sieci neuronowych począwszy od pojedynczego neuronu kończąc na praktycznych przykładach zastosowań sieci. Programy demonstracyjne można wywoływać na kilka sposobów:

- można wybrać opcję **Demos** z okna **LaunchPad** (wybór opcji powoduje wyświetlenie pierwszego tematu pomocy opisującego ogólnie programy demonstracyjne biblioteki, kolejne tematy opisują dostępne programy demonstracyjne, zawierają odnośniki [Run this demo](#) pozwalające na uruchomienie opisywanego programu demonstracyjnego),
- można wywołać okno zawierające odwołania do wszystkich programów demonstracyjnych, wpisując w oknie **Command Window** polecenie **nndtoc**,
- można wpisać nazwę właściwej funkcji demonstracyjnej w oknie **Command Window**.

1.1. Neuron z jednym wejściem

Funkcja **nnd2n1** demonstruje sposób w jaki neuron przetwarza sygnał wejściowy w sygnał wyjściowy. Funkcję tą można wybrać:

- pisząc w oknie **Command Window** polecenie **nnd2n1**,
- wybierając pierwszy z przykładów do rozdziału drugiego (**Chapter 2 demos**) z okna wyświetlonego poleceniem **nndtoc**,
- wskazując tekst [Run this demo](#) w temacie pomocy: „Simple neuron and transfer functions”.

Po uruchomieniu funkcji **nnd2n1** wyświetlane jest okno (przedstawione na rys. 2.), pozwalające użytkownikowi na badanie możliwości pojedynczego neuronu.

W górnej części okna wyświetlany jest model neuronu. Na rysunku tym przyjęto następujące oznaczenia:

p – sygnał wejściowy,

w – waga odpowiadająca sygnałowi **p**,

b – bias (dodatkowe wejście neuronu, na które stale podawany jest sygnał o wartości równej 1, wartość parametru **b** odpowiada wartości wagi neuronu dla tego wejścia),

a – sygnał wyjściowy.

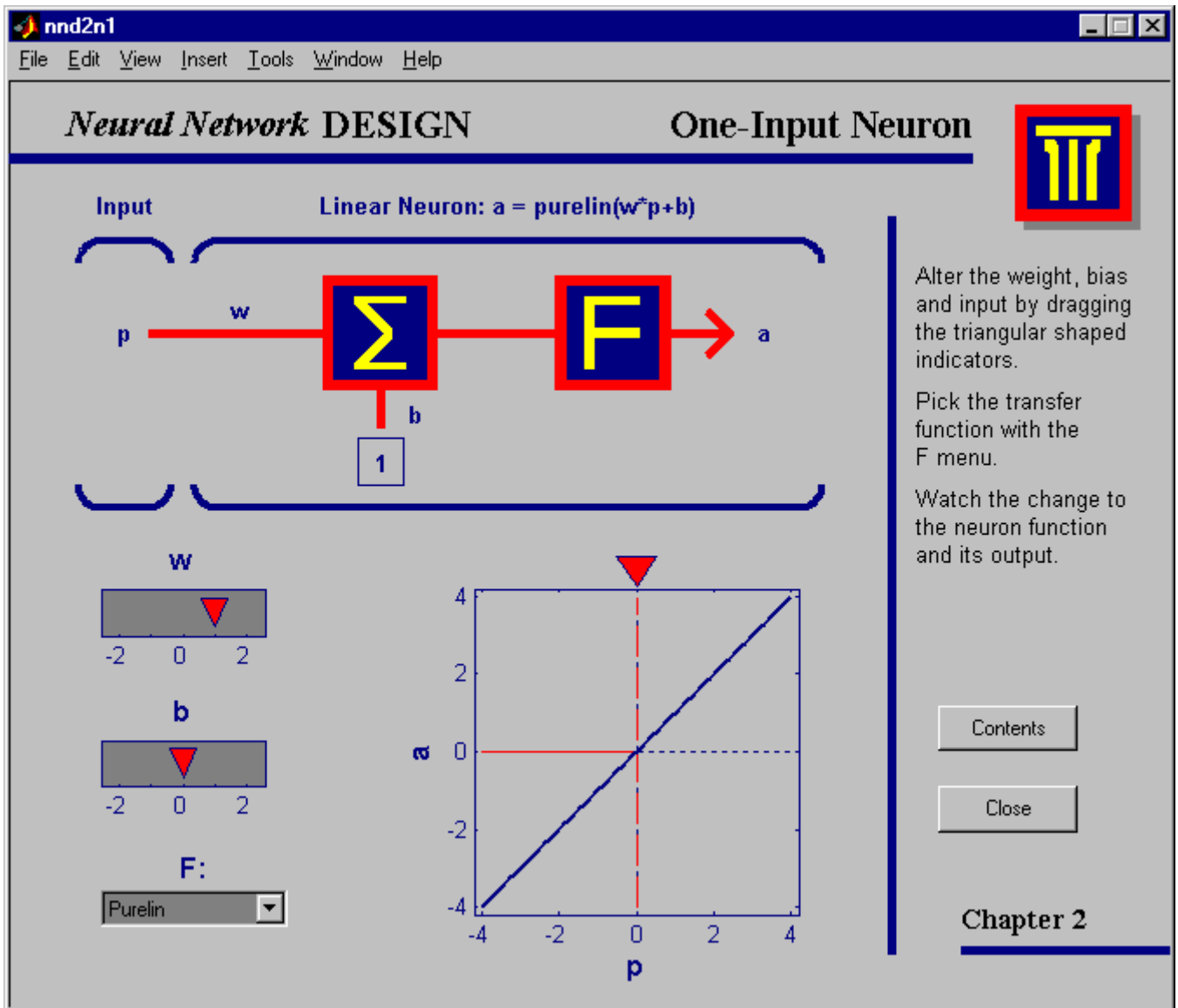


– blok sumujący,



– blok aktywacji,

Nad rysunkiem umieszczany jest zawsze wzór opisujący sposób przekształcania sygnału wejściowego **p** w sygnał wyjściowy **a**. Na powyższym rysunku zależność ta opisana jest wzorem: **a = purelin(w*p+b)**. Zależność w nawiasie jest zawsze taka sama - wyznacza ona wartość potencjału membranowego ($\varphi = w*p+b$), nazwa **purelin** opisuje zastosowaną funkcję aktywacji.

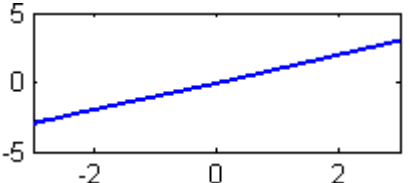
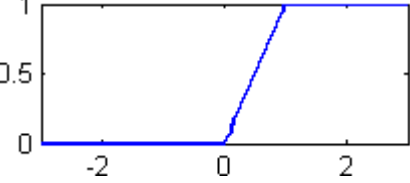
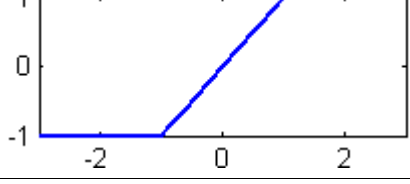
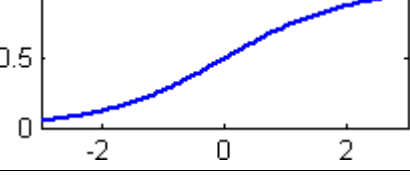
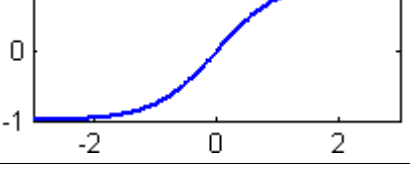


Rys. 2. Funkcja nnd2n1

W poniższej tabeli zostały zebrane funkcje często wykorzystywane jako funkcje aktywacji neuronu.

Funkcja	Wykres	Funkcja Matlab-a
$a = \begin{cases} 1, & \varphi \geq 0 \\ 0, & \varphi < 0 \end{cases}$		hardlim
$a = \begin{cases} 1, & \varphi \geq 0 \\ -1, & \varphi < 0 \end{cases}$		hardlims



$a = \varphi$		<p>purelin</p>
$a = \begin{cases} 1, & \varphi > 1 \\ \varphi, & 0 \leq \varphi \leq 1 \\ 0, & \varphi < 0 \end{cases}$		<p>satlin</p>
$a = \begin{cases} 1, & \varphi > 1 \\ \varphi, & -1 \leq \varphi \leq 1 \\ -1, & \varphi < -1 \end{cases}$		<p>satlins</p>
$a = \frac{1}{1 + e^{-\varphi}}$		<p>logsig</p>
$a = \tanh(\varphi)$		<p>tansig</p>

Parametrami wpływającymi na sposób w jaki neuron przetwarza sygnał wejściowy są:

- waga,
- bias,
- funkcja aktywacji.

Parametry te można zmieniać posługując się kontrolkami umieszczonymi pod rysunkiem neuronu w lewej kolumnie.

Aby zmienić wartość wagi **w** lub biasu **b** wystarczy zmienić położenie czerwonego wskaźnika na linijce (waga i bias mogą przyjmować wartości całkowite z przedziału [-2, 2])



Funkcją aktywacji neuronu może być jedna z omówionych wcześniej funkcji, odpowiednią funkcję można wybrać posługując się listą rozwijalną.



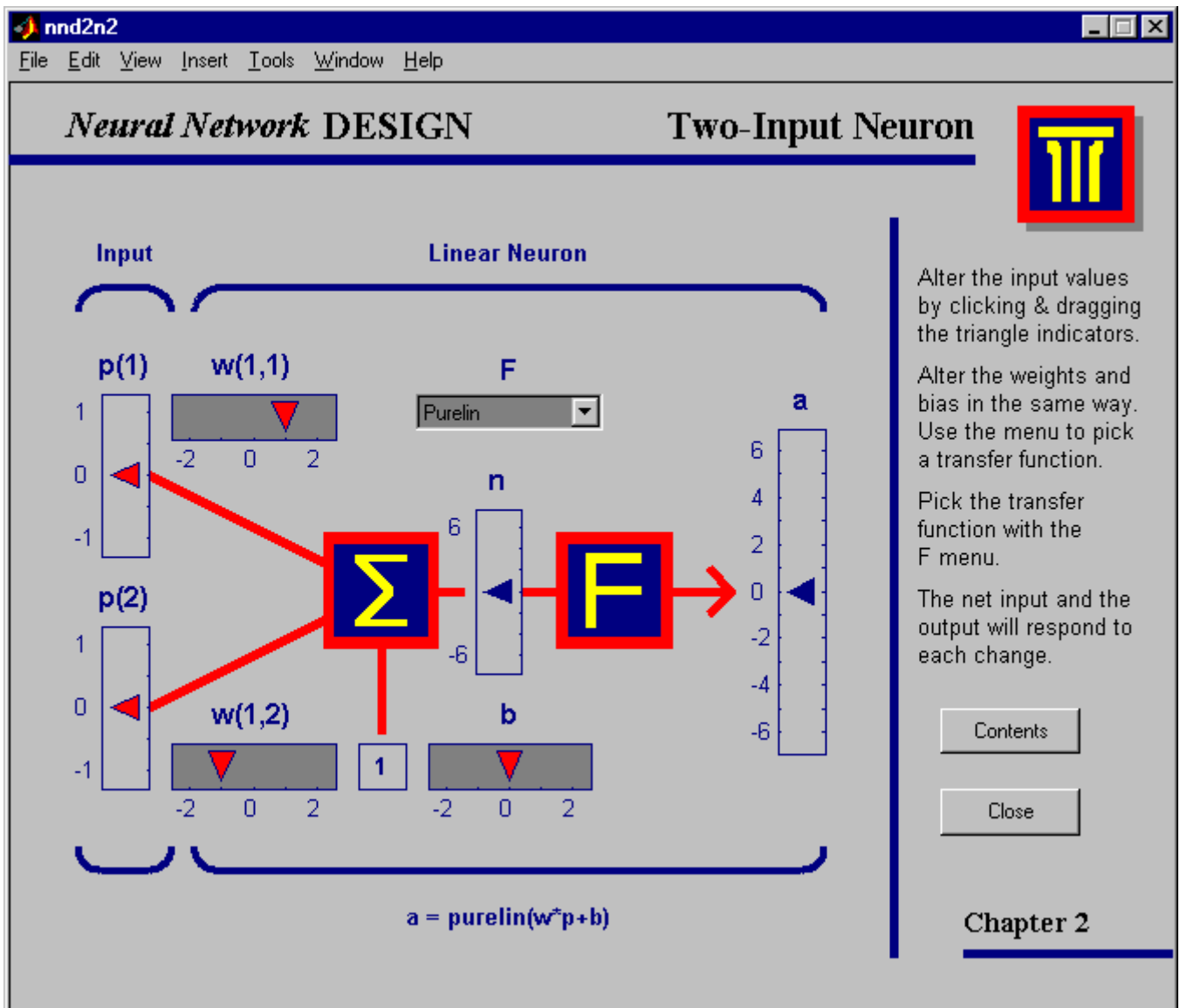
Zmiana wartości jednego z parametrów powoduje aktualizację wykresu umieszczonego pod rysunkiem neuronu z prawej strony. Wykres ten przedstawia sposób w jaki neuron przekształca sygnał wejściowy **p** (oś pozioma) w sygnał wyjściowy **a** (oś pionowa).

1.2. Neuron z dwoma wejściami

Funkcja **nnd2n2** demonstruje sposób w jaki neuron przetwarza sygnały wejściowe w sygnał wyjściowy. Funkcję tą można wybrać:

- pisząc w oknie **Command Window** polecenie **nnd2n2**,
- wybierając drugi z przykładów do rozdziału drugiego (**Chapter 2 demos**) z okna wyświetlonego poleceniem **nndtoc**,
- wskazując tekst **Run this demo** w temacie pomocy: „Neuron with vector input”.

Uruchomienie funkcji **nnd2n2** powoduje wyświetlenie okna przedstawionego na poniższym rysunku.



Rys. 3. Funkcja **nnd2n2**

Całą zawartość okna zajmuje rysunek neuronu posiadającego dwa wejścia oraz bias. Znaczenie występujących na rysunku symboli i oznaczeń jest takie samo jak w przypadku omawianego wcześniej przykładu neuronu z jednym wejściem. Rozszerzone zostało jedynie znaczenie parametrów **p** i **w**.



Parametr \mathbf{p} określał poprzednio wartość sygnału wejściowego, w omawianym przykładzie \mathbf{p} jest wektorem, którego pierwszy element $\mathbf{p}(1)$ definiuje wartość sygnału podawanego na pierwsze wejście i odpowiednio drugi element $\mathbf{p}(2)$ definiuje wartość sygnału podawanego na drugie wejście. Podobnie, parametr \mathbf{w} określał poprzednio wartość wagi, w omawianym przykładzie \mathbf{w} jest macierzą o jednym wierszu i dwóch kolumnach, element $\mathbf{w}(1,1)$ definiuje wartość wagi dla pierwszego wejścia a $\mathbf{w}(1,2)$ definiuje wartość wagi dla drugiego wejścia.

Wartości parametrów: $\mathbf{p}(1)$, $\mathbf{p}(2)$, $\mathbf{w}(1,1)$, $\mathbf{w}(1,2)$ oraz \mathbf{b} , można modyfikować zmieniając położenie czerwonych wskaźników na odpowiednich liniijkach. Odpowiednią funkcję można wybrać posługując się listą rozwijalną, analogicznie jak w przykładzie poprzednim. Zmiana wartości jednego z parametrów powoduje aktualizację położenia wskaźników umieszczonych odpowiednio za blokiem sumującym i blokiem aktywacji.

1.3. Powierzchnia decyzyjna

Funkcja **nnd4db** demonstruje sposób w jaki neuron o dwóch wejściach i funkcji aktywacji w postaci bipolarnej funkcji skoku klasyfikuje sygnały wejściowe.

Funkcję tą można wybrać:

- pisząc w oknie **Command Window** polecenie **nnd4db**,
- wybierając pierwszy z przykładów do rozdziału czwartego (**Chapter 4 demos**) z okna wyświetlonego poleceniem **nndtoc**,
- wskazując tekst **Run this demo** w temacie pomocy: „*Decision Boundaries*”.

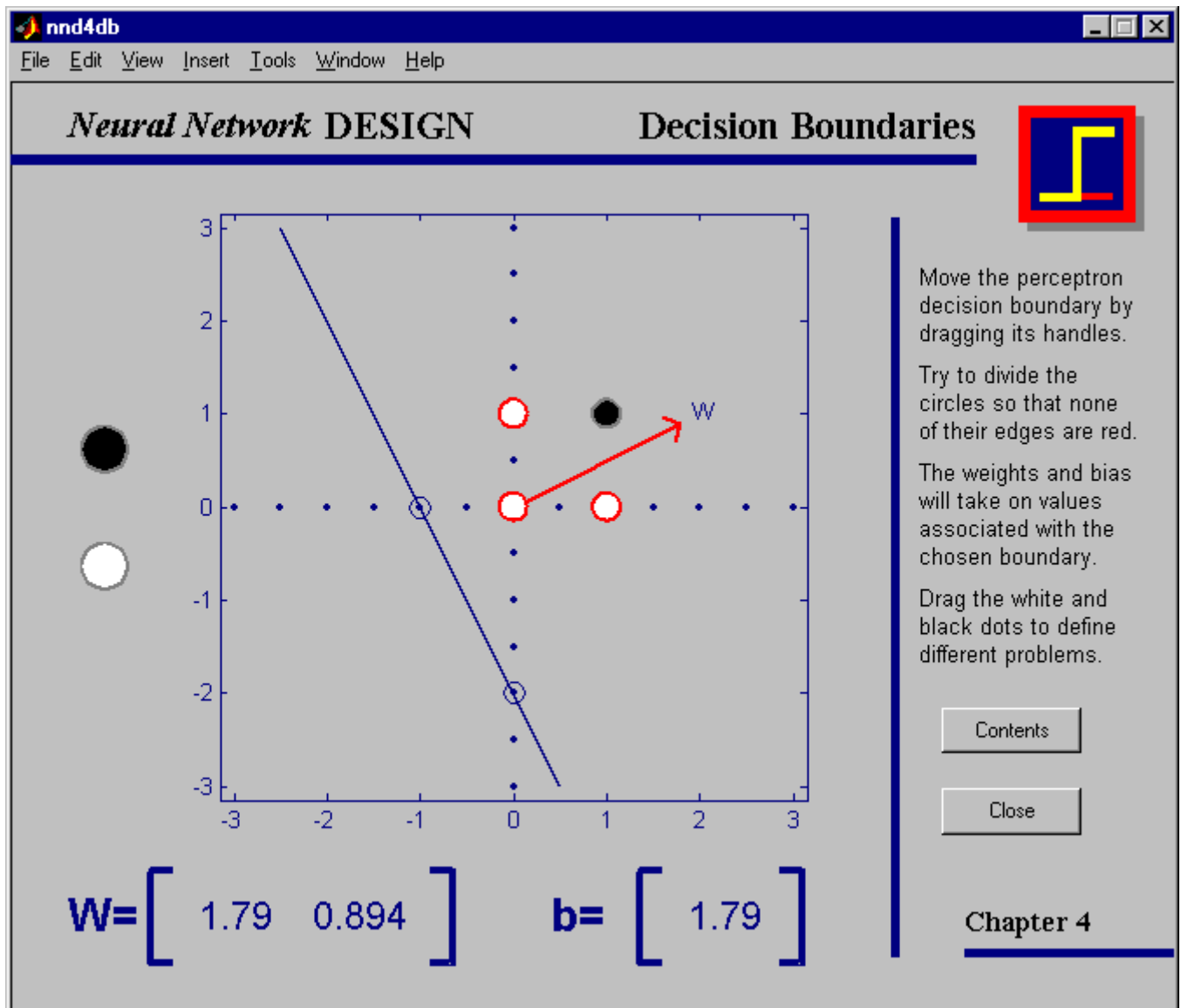
Uruchomienie funkcji **nnd4db** powoduje wyświetlenie okna przedstawionego na rys. 4.

Widoczna na rysunku linia odpowiada powierzchni decyzyjnej wyznaczonej przez neuron o podanych pod rysunkiem wartościach wag \mathbf{w} i biasu \mathbf{b} . Linia ta dzieli przestrzeń wejściową na dwa obszary decyzyjne. Odpowiedź neuronu dla wartości wejściowych należących do jednego z obszarów wynosi 1 (półpłaszczyzna wskazywana przez widoczny na rysunku wektor wag \mathbf{w}), dla drugiego obszaru sygnał wyjściowy wynosi (-1). Użytkownik może zmieniać położenie powierzchni decyzyjnej zmieniając położenie punktów leżących na linii (punkty te otoczone są niebieskimi okręgami). Zmiana położenia linii powoduje automatyczną aktualizację wyświetlanych pod rysunkiem wag i biasu.

Posługując się omawianą w tym punkcie funkcją można spróbować rozdzielić widoczne na rysunku punkty białe i czarne (punkty czarne odpowiadają sygnałowi wyjściowemu o wartości 1, punkty białe sygnałowi o wartości (-1)). Posługując się językiem sieci neuronowych można stwierdzić, że użytkownik może nauczyć neuron określonego zachowania (klasyfikacja wskazanych punktów) – uczenie sprowadza się do wyznaczenia odpowiednich wag i biasu. Program demonstracyjny dodatkowo otacza określone punkty czerwonym lub szarym okręgiem. Czerwona otoczka oznacza, że punkt znajduje się w niewłaściwym obszarze decyzyjnym.



Przykładowa funkcja umożliwia rozwiązywanie wielu problemów klasyfikacji – użytkownik może przesuwać punkty białe i czarne. Można również dodawać nowe punkty, posługując się odpowiednio białym i czarnym polem umieszczonym przy lewej krawędzi okna.



Rys. 4. Funkcja nnd4db

Zadania

Zad.1. Posługując się funkcją demonstrującą działanie neuronu o jednym wejściu (**nnd2n1**) porównaj wykresy $a=F(p)$ dla dostępnych funkcji aktywacji. Badanie wpływu parametrów (**w**, **b**, **F**) przeprowadź zgodnie z podanym poniżej schematem:

- wybierz jedną z dostępnych funkcji aktywacji,
- ustaw bias na 0,
- zmieniaj wartość wagi (Jak wpływa na wykres waga o wartości dodatniej a jak waga o wartości ujemnej? Porównaj parami wykresy dla wagi $w=1$ i $w=2$ oraz $w=-1$ i $w=-2$),
- zbadaj wpływ biasu.

Zad.2. Dany jest neuronu o dwóch wejściach i wagach $w=[1, 2]$. Oblicz wartość sygnału wyjściowego dla sygnałów wejściowych $x=[1, -1]$, przyjmując kolejno jako funkcję aktywacji:

- funkcję liniową,
- unipolarną funkcję skoku,
- bipolarną funkcję skoku,
- tangens hiperboliczny (parametr $\beta=1$),
- funkcja sigmoidalna (parametr $\beta=1$).

Sprawdź poprawność wyników posługując się funkcją demonstrującą działanie neuronu o dwóch wejściach (**nnd2n2**).

Zbadaj wpływ biasu na uzyskiwaną wartość sygnału wyjściowego.

Zad.3. Posługując się funkcją **nnd4db** dobierz wagi i bias neuronu tak, aby poprawnie klasyfikował punkty: $x_I = [1, 1]^T$, $x_{II} = [-1, 1]^T$, $x_{III} = [-1, -1]^T$, $x_{IV} = [1, -1]^T$. Rozważ następujące przypadki:

- neuron zwraca sygnał wyjściowy o wartości 1 dla punktów: x_I , x_{II} , x_{IV} i wartości (-1) dla x_{III} ,
- neuron zwraca sygnał wyjściowy o wartości (-1) dla punktów: x_I , x_{II} , x_{IV} i wartości 1 dla x_{III} ,
- neuron zwraca sygnał wyjściowy o wartości 1 dla punktu: x_I i wartości (-1) dla x_{II} , x_{III} , x_{IV} ,
- neuron zwraca sygnał wyjściowy o wartości 1 dla punktów: x_I , x_{III} i wartości (-1) dla x_{II} , x_{IV} .