

ARHITECTURI DE REȚELE NEURONALE FEEDFORWARD. STUDIUL PRIN SIMULARE AL TRANSFERULUI INTRARE-IEȘIRE

1. Considerații generale, motivație și obiectiv

Rețelele neuronale feedforward sunt sisteme statice ce realizează un transfer instantaneu intrare-ieșire, descriabil prin compunerea unor funcții dependente de topologia (arhitectura) rețelei. Astfel, odată ce a fost definită arhitectura unei rețele, comportarea acesteia poate fi investigată prin simulare într-un mediu software cu facilități specifice calculelor tehnico-științifice. Un atare mediu software este oferit de MATLAB, care, pe lângă nucleul general, pune la dispoziție colecția de funcții destinate operării cu rețele neuronale, înglobate în pachetul *Neural Network Toolbox*.

Prin parcurgerea acestei ședințe de laborator, studentul va acumula cunoștințele fundamentale privitoare la arhitectura rețelelor feedforward și la descrierea matematică a transferului intrare-ieșire realizat. De asemenea, făcând apel la simularea în MATLAB, studentul va explora în detaliu modul de operare al unor arhitecturi particulare, în scopul creării unui suport de factură intuitivă, deosebit de util pentru consolidarea abordărilor teoretice, generale.

2. Breviar de terminologie și descriere matematică

Trecerea în revistă a noțiunilor fundamentale privitoare la arhitectura rețelelor feedforward se realizează prin extinderea gradată a complexității configurației studiate, conform următorului plan:

- neuron cu o singură intrare;
- neuron cu mai multe intrări;
- rețea neuronală cu un singur strat;
- rețea neuronală cu un două straturi;
- rețea neuronală cu mai multe straturi.

Sunt, de asemenea, prezentate câteva funcții din pachetul MATLAB *Neural Network Toolbox*, a căror utilizare este recomandată pentru exercițiile de simulare.

3. Modelul neuronului artificial

3.1. Neuronul artificial cu o singură intrare

În figura 1 se prezintă schema bloc a unui neuron cu o singură intrare (intrare scalară), notată $x \in \mathbb{R}$ și ieșirea scalară $y \in \mathbb{R}$. Constanta $w \in \mathbb{R}$ se numește *pondere* (eng. *weight*), iar constanta $b \in \mathbb{R}$ poartă denumirea de *deplasare* (eng. *bias*). Intrarea cu valoarea precizată “1” arată că, indiferent de semnalul de intrare x , valoarea cu care este alimentat blocul b este tot

timpul 1. Referirea comună a constantelor $w, b \in \mathbb{R}$ se face prin termenul de “parametrii neuronului”. În utilizarea neuronului, valorile parametrilor $w, b \in \mathbb{R}$ sunt *ajustabile*. Pe ajustarea adecvată a acestor valori se bazează capacitatea de învățare a neuronului artificial.

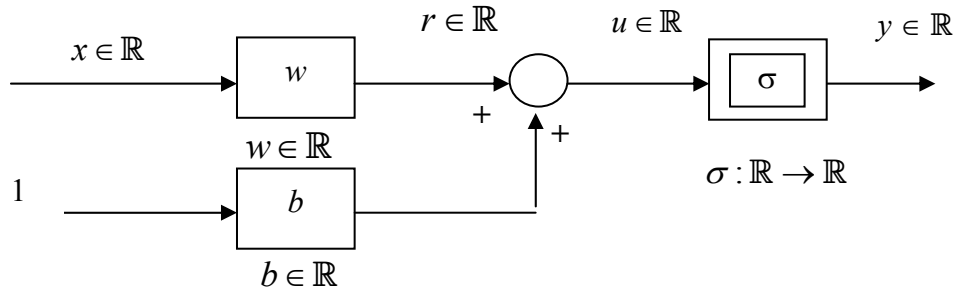


Fig. 1. Schema bloc a unui neuron cu o singură intrare

Funcția $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ se numește *funcția de activare a neuronului* (eng. *activation function*). Pentru funcția de activare σ se folosesc *funcții analitice standard*, liniare sau neliniare. Blocul σ definește *nodul* neuronului, iar ansamblul care generează semnalul u definește partea *liniară*.

În tabelul 1 sunt prezentate analitic și grafic funcțiile de activare cu cea mai frecventă utilizare în practică; de asemenea sunt furnizate *numele funcțiilor MATLAB* aferente, disponibile în Neural Network Toolbox.

Revenind cu aceste informații la studierea schemei bloc din figura 1, se constată că neuronul artificial cu o singură intrare este *un sistem static* (asigură transferul *instantaneu* al semnalului de intrare către ieșire), a cărui funcționare este descrisă prin aplicația:

$$f : \mathbb{R} \rightarrow \mathbb{R}, \quad (1)$$

$$y = f(x) = \sigma(w \cdot x + b). \quad (2)$$

Așadar f este o funcție realizată prin *compunere*, în următoarea manieră ilustrată sugestiv de schema bloc din fig. 1:

$$y = f(x) = \sigma(u(x)), \quad (3)$$

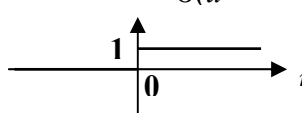
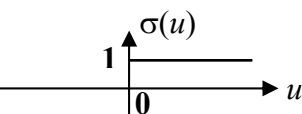
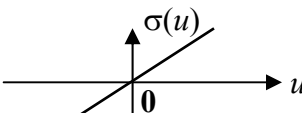
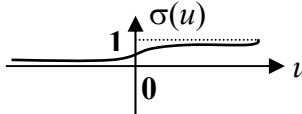
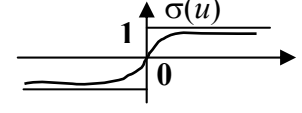
unde:

$$u(x) = r(x) + b, \quad (4)$$

$$r(x) = w \cdot x. \quad (5)$$

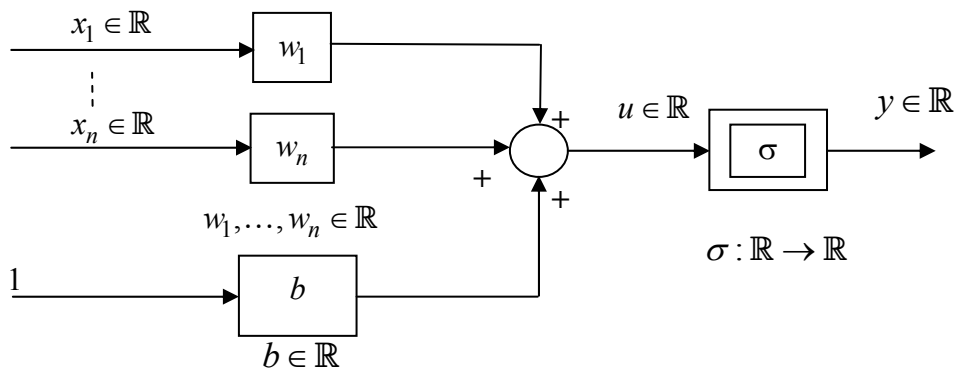
Din expresia lui $y = f(x)$ dată în relația (2), se observă că *transferul instantaneu* intrare-ieșire $x \rightarrow y$ este *parametrizat* de $w, b \in \mathbb{R}$ sau, cu alte cuvinte, *dependența* lui y de x este *parametrizată* de valorile ponderii $w \in \mathbb{R}$ și deplasării $b \in \mathbb{R}$. Subliniem faptul că, deși conform (2), ieșirea y se prezintă ca funcție de trei variabile reale (x, w, b), *din punct de vedere sistemic*, variabila x este *singura* cu semnificație de *semnal de intrare*, remarcă pusă în evidență de către schema bloc din fig. 1. Dintr-un semnal de intrare x nu se poate obține orice formă a semnalului de ieșire y (parametrizat de w și b), cu alte cuvinte, y va păstra informații specifice nodului.

Tabelul 1. Funcții de activare uzuale

Nume Funcție	Expresie analitică	Reprezentare Grafică	Nume funcție MATLAB
Treaptă unipolară	$\sigma(u) = \begin{cases} 0, u < 0 \\ 1, u \geq 0 \end{cases}$		hardlim
Treaptă Bipolară	$\sigma(u) = \begin{cases} -1, u < 0 \\ 1, u \geq 0 \end{cases}$		hardlims
Funcție Liniară	$\sigma(u) = u$		purelin
Sigmoid Unipolar	$\sigma(u) = \frac{1}{1 + e^{-u}}$		logsig
Sigmoid Bipolar	$\sigma(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$		tansig

3.2. Neuronul artificial cu mai multe intrări

Neuronul cu mai multe intrări generalizează modelul prezentat în paragraful anterior, în sensul că *semnalul de intrare este vectorial* (posedă mai multe componente). În figura 2 se prezintă schema detaliată a unui neuron cu ieșirea $y \in \mathbb{R}$ și cu n intrări, notate $x_i \in \mathbb{R}, i = 1, \dots, n$:

Fig. 2. Schema detaliată a unui neuron cu n intrări

Funcția de activare rămâne neschimbată, dar intrarea ei, u , are expresia:

$$u = w_1 \cdot x_1 + \dots + w_n \cdot x_n + b = \sum_{i=1}^n w_i x_i + b, \quad (6)$$

care se poate scrie în formă vectorial-matriceală:

$$u = \mathbf{w} \cdot \mathbf{x} + b, \quad (7)$$

unde $\mathbf{x}$ este *vectorul coloană* conținând intrările, $\mathbf{x} = [x_1 \dots x_n]^T \in \mathbb{R}^n$, iar $\mathbf{w}$ este *vectorul linie* (adică un caz particular de matrice) conținând ponderile, $\mathbf{w} \in \mathbb{R}^{1 \times n}$.

Această scriere ne arată că reprezentarea grafică compactă a neuronului cu mai multe intrări se poate realiza ca în figura 3:

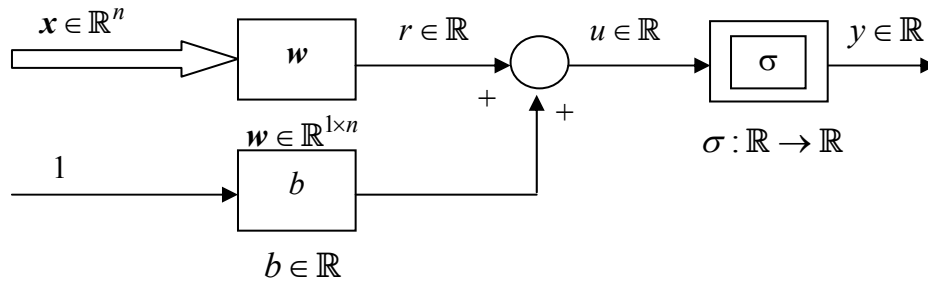


Fig. 3. Schema bloc a unui neuron cu n intrări

Așadar, neuronul cu n intrări este un *sistem static* a cărui funcționare este descrisă prin *aplicația de variabilă vectorială*:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}, \quad (8)$$

$$y = f(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b), \quad (9)$$

ce rezultă din următoarea compunere de funcții (pusă în evidență de schema din fig. 2):

$$y = f(\mathbf{x}) = \sigma(u(\mathbf{x})) \quad (10)$$

unde:

$$u(\mathbf{x}) = r(\mathbf{x}) + b, \quad (11)$$

$$r(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}. \quad (12)$$

Se observă că transferul $x \rightarrow y$ este *parametrizat* de $n+1$ elemente: n elemente conținute în vectorul linie al ponderilor $\mathbf{w} \in \mathbb{R}^{1 \times n}$ și scalarul $b \in \mathbb{R}$ (deplasarea fiind unică pentru un neuron).

4. Rețele de neuroni artificiali

4.1. Rețea cu un singur strat

În figura 4 se prezintă schema bloc a unei rețele neuronale cu un singur strat (eng. *layer*) conținând p neuroni. Rețeaua posedă n intrări, notate compact $\mathbf{x} \in \mathbb{R}^n$, și p ieșiri, notate compact $\mathbf{y} \in \mathbb{R}^p$:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix} \in \mathbb{R}^p. \quad (13)$$

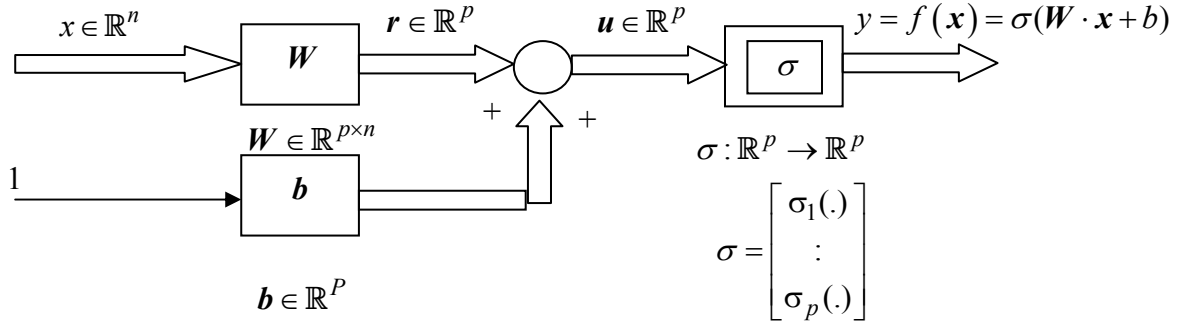


Fig. 4. Schema bloc a unei rețele neuronale cu un singur strat de p neuroni

Cele n intrări sunt comune pentru toți cei p neuroni și fiecare dintre cele p ieșiri corespunde funcției de activare a neuronului corespunzător. În acest caz $\mathbf{W} \in \mathbb{R}^{p \times n}$ este o *matrice*, iar $\mathbf{b} \in \mathbb{R}^p$ este un *vector coloană*. Rețeaua neuronală cu un singur strat este un *sistem static* a cărui funcționare este descrisă prin *aplicația vectorială, de variabilă vectorială*:

$$\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^p, \quad (14)$$

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}), \quad (15)$$

ce rezultă din următoarea compunere de funcții (pusă în evidență de schema din fig. 4)

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \sigma(\mathbf{u}(\mathbf{x})), \quad (16)$$

unde:

$$\mathbf{u}(\mathbf{x}) = \mathbf{r}(\mathbf{x}) + \mathbf{b}, \quad (17)$$

$$\mathbf{r}(\mathbf{x}) = \mathbf{W} \cdot \mathbf{x}. \quad (18)$$

Se observă că transferul $\mathbf{x} \rightarrow \mathbf{y}$ este *parametrizat* de cele $p \cdot n$ elemente ale matricei $\mathbf{W} \in \mathbb{R}^{p \times n}$ și cele p elemente ale vectorului $\mathbf{b} \in \mathbb{R}^p$.

4.2. Rețea cu două straturi

În figura 5 se prezintă schema bloc a unei rețele neuronale cu două straturi, conținând p_1 și, respectiv, p_2 neuroni. Rețeaua posedă n_1 intrări, notate compact $\mathbf{x}^1 \in \mathbb{R}^{n_1}$, și p_2 ieșiri, notate compact $\mathbf{y}^2 \in \mathbb{R}^{p_2}$:

$$\mathbf{x}^1 = \begin{bmatrix} x_1^1 \\ \vdots \\ x_{n_1}^1 \end{bmatrix} \in \mathbb{R}^{n_1}, \quad \mathbf{y}^2 = \begin{bmatrix} y_1^2 \\ \vdots \\ y_{p_2}^2 \end{bmatrix} \in \mathbb{R}^{p_2}. \quad (19)$$

Arhitectura acestei rețele rezultă prin înserierea a două rețele cu câte un singur strat fiecare (de tipul celor prezentate în paragraful 4.1), în așa manieră încât ieșirile primului strat să constituie intrările celui de-al doilea strat.

Primul strat al rețelei mai este referit cu denumirea de *strat de intrare* (eng. *input layer*) iar al doilea cu denumirea de *strat de ieșire* (eng. *output layer*). Conectarea impune ca numărul ieșirilor primului strat (adică p_1) să coincidă cu numărul intrărilor în cel de al doilea strat (adică n_2). Semnalul vectorial de la ieșirea primului strat, $\mathbf{y}^1 \in \mathbb{R}^{p_1}$, constituie semnalul vectorial de intrare pentru al doilea strat, $\mathbf{x}^2 \in \mathbb{R}^{n_2}$, adică are loc identitatea de semnale:

$$\mathbf{y}^1 (\in \mathbb{R}^{p_1}) \equiv \mathbf{x}^2 (\in \mathbb{R}^{n_2}). \quad (20)$$

Rețeaua neuronală cu două straturi este un sistem static a cărui funcționare este descrisă prin *aplicația vectorială, de variabilă vectorială*:

$$\mathbf{f}: \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{p_2}, \quad (21)$$

$$\mathbf{y}^2 = \mathbf{f}(\mathbf{x}^1) = \sigma^2 \left(\mathbf{W}^2 \sigma^1 (\mathbf{W}^1 \mathbf{x}^1 + \mathbf{b}^1) + \mathbf{b}^2 \right). \quad (22)$$

Această aplicație rezultă din următoarea compunere de funcții (pusă în evidență de schema bloc din figura 5):

- *La nivelul celui de al doilea strat:*

$$\mathbf{y}^2 = \mathbf{f}^2(\mathbf{x}^2) = \sigma^2 \left(\mathbf{u}^2(\mathbf{x}^2) \right), \quad (23)$$

unde:

$$\mathbf{u}^2(\mathbf{x}^2) = \mathbf{r}^2(\mathbf{x}^2) + \mathbf{b}^2, \quad (24)$$

$$\mathbf{r}^2(\mathbf{x}^2) = \mathbf{W}^2 \mathbf{x}^2. \quad (25)$$

- *La nivelul primului strat:*

$$\mathbf{x}^2 = \mathbf{y}^1 = \mathbf{f}^1(\mathbf{x}^1) = \sigma^1 \left(\mathbf{u}^1(\mathbf{x}^1) \right), \quad (26)$$

unde:

$$\mathbf{u}^1(\mathbf{x}^1) = \mathbf{r}^1(\mathbf{x}^1) + \mathbf{b}^1, \quad (27)$$

$$\mathbf{r}^1(\mathbf{x}^1) = \mathbf{W}^1 \mathbf{x}^1. \quad (28)$$

Se observă că transferul $\mathbf{x}^1 \rightarrow \mathbf{y}^2$ realizat de rețeaua cu două straturi de neuroni este parametrizat de elementele matricelor $\mathbf{W}^1 \in \mathbb{R}^{p_1 \times n_1}$, $\mathbf{W}^2 \in \mathbb{R}^{p_2 \times n_2}$ (ponderile celor două straturi) și elementele vectorilor $\mathbf{b}^1 \in \mathbb{R}^{p_1}$, $\mathbf{b}^2 \in \mathbb{R}^{p_2}$ (deplasările celor două straturi).

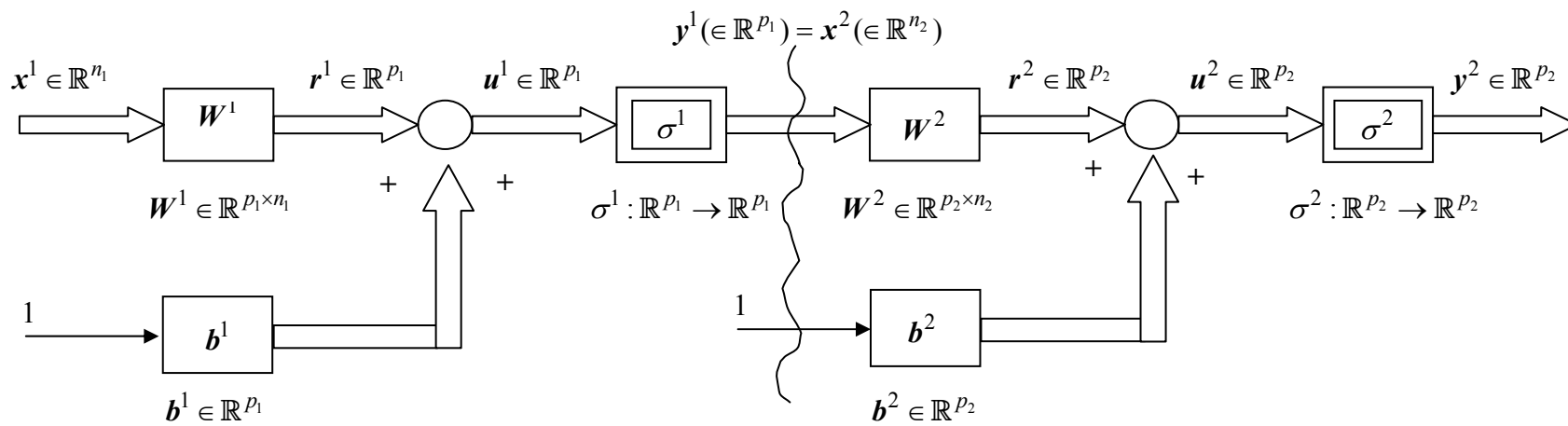


Fig. 5. Schema bloc a unei rețele neuronale cu două straturi, conținând p_1 și, respectiv, p_2 neuroni

4.3. Cazul general - rețea cu mai multe straturi

Într-o manieră similară celei prezentate în paragraful 3.2.2 se poate construi o rețea cu trei (sau mai multe) straturi de neuroni, conectând trei (sau mai multe) rețele cu un singur strat, astfel încât intrările fiecărui strat (exceptând primul) să coincidă cu ieșirile stratului care îl precede. Cele amintite mai înainte cu privire la terminologia prin care se referă straturile rețelei rămân valabile. Existența mai multor straturi are drept efect atât posibilitatea de a defini transferul intrare-ieșire cu ajutorul a mai mulți parametri, cât și mai multe nivele de neliniarități care se compun în acest transfer. Asupra acestor aspecte se va reveni în detaliu atunci când se va discuta despre capacitatea de aproximare pe care o posedă unele rețele neuronale. În practica uzuală se face apel la arhitecturi *cu cel mult trei straturi*.

Straturile de neuroni cuprinse între stratul de intrare și cel de ieșire se numesc *straturi ascunse* (eng. *hidden*). În unele texte, orice strat al rețelei care nu este strat de ieșire este referit drept *strat ascuns* (inclusiv stratul de intrare). Ca urmare, exprimarea fără nici un fel de echivoc este cea care atribuie numere de ordine straturilor, astfel în cazul rețelei cu două straturi, stratul I desemnează stratul de intrare și stratul II este stratul de ieșire.

5. Problematica propusă pentru studiu

Problema 1

Se consideră un neuron cu o intrare (vezi fig. 1). Să se elaboreze un program MATLAB pentru a studia modul de compunere a funcțiilor, considerând la intrarea neuronului, valorile:

$$x = -v : v_s : v;$$

unde valorile v și v_s vor fi precizate în cursul ședinței de aplicații, pentru fiecare colectiv de lucru în parte.

Se vor prezenta grafic următoarele trei funcții:

1. $u(x)$ (relația (4)) pentru $x \in [-v, v]$;
2. $\sigma(x)$ pentru $u \in [u_{\min}, u_{\max}]$, acest interval reprezentând mulțimea valorilor lui $u(x)$, în condițiile de la punctul 1. Pasul de discretizare u_s al intervalului $[u_{\min}, u_{\max}]$, se va alege astfel încât să asigure calitatea plotării;
3. $f(x) = \sigma(u(x))$ (relația (3)) pentru $x \in [-v, v]$.

Se vor considera cazurile funcției de activare:

- a) sigmoid bipolar;
- b) treaptă unipolară.

Pentru fiecare din cazurile a) și b), se vor studia următoarele situații:

- (i) $w = 1, b = 0$;
- (ii) $w = 1, b = b^*$;
- (iii) $w = w^*, b = 0$;
- (iv) $w = w^*, b = b^*$;

unde valorile w^* și b^* vor fi precizate în cursul ședinței de aplicații.

Se vor comenta legăturile existente între graficele rezultate la (i) - (iv), iar atunci când este posibil, aceste legături vor fi utilizate pentru *trasarea directă* a graficelor, fără a mai face apel la programul elaborat în MATLAB.

Notă: Pentru exemplificare putem considera $v_1 = 10$, $v_s = 0.01$.

Problema 2

Se consideră un neuron cu două intrări (vezi fig. 2 pentru $n = 2$). Să se elaboreze un program MATLAB pentru a studia modul de compunere a funcțiilor, considerând la intrarea neuronului, vectorii:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

unde:

$$x_1 = -v_1 : v_s : v_1;$$

$$x_2 = -v_2 : v_s : v_2;$$

valorile v_1 , v_2 , v_s , v_s urmând a fi precizate în cursul ședinței de aplicații, pentru fiecare colectiv de lucru în parte.

Se vor prezenta grafic următoarele trei funcții:

1. $u(x)$ (relația (11)) pentru $(x_1, x_2) \in [-v_1, v_1] \times [-v_2, v_2]$

2. $\sigma(u)$ pentru $u \in [u_{\min}, u_{\max}]$, acest interval reprezentând mulțimea în care $u(x)$ ia valori, în condițiile de la punctul 1. Pasul de discretizare u_s al intervalului $[u_{\min}, u_{\max}]$, se va alege astfel încât să asigure calitatea plotării.

3. $f(x) = \sigma(u(x))$ (relația (10)) pentru $(x_1, x_2) \in [-v_1, v_1] \times [-v_2, v_2]$

Se vor considera cazurile funcției de activare:

a) sigmoid bipolar;

b) treaptă unipolară.

Pentru fiecare din cazurile a) și b), se vor studia următoarele situații:

(i) $\mathbf{w} = [1 \ 0]$, $b = 0$;

(ii) $\mathbf{w} = [1 \ 0]$, $b = b^*$;

(iii) $\mathbf{w} = [0 \ 1]$, $b = 0$;

(iv) $\mathbf{w} = [0 \ 1]$, $b = b^*$;

(v) $\mathbf{w} = [1 \ 1]$, $b = 0$;

(vi) $\mathbf{w} = [1 \ 1]$, $b = b^*$;

(vii) $\mathbf{w} = [w_1^* \ w_2^*]$, $b = 0$;

(viii) $\mathbf{w} = [w_1^* \ w_2^*]$, $b = b^*$;

unde valorile w_1^* , w_2^* și b^* vor fi precizate în cursul ședinței de aplicații.

Se vor comenta legăturile existente între graficele existente la (i) - (viii), iar atunci când este posibil, aceste legături vor fi utilizate pentru *trasarea directă* a graficelor, fără a mai face apel la programul elaborat în MATLAB.

Notă: Pentru exemplificare putem alege $v_1 = 10$, $vs_1 = 0.01$, $v_2 = 20$, $vs_2 = 0.02$.

Problema 3

Se consideră o rețea neuronală cu un singur strat cu doi neuroni (vezi fig.3 - cu $n = 1$, $p = 2$).

Să se elaboreze un program MATLAB pentru a studia modul de compunere a funcțiilor, considerând la intrarea rețelei, valorile:

$$x = -v : vs : v;$$

unde valorile v și vs vor fi precizate în cursul ședinței de aplicații, pentru fiecare colectiv de lucru în parte.

Se vor prezenta grafic următoarele șase funcții:

1. $u_1(x), u_2(x)$ (relația (17)) pentru $x \in [-v, v]$
2. $\sigma_1(u_1)$ pentru $u_1 \in [u_{1\min}, u_{1\max}]$, acest interval reprezentând mulțimea valorilor lui $u_1(x)$, în condițiile de la punctul 1. Pasul de discretizare us_1 al intervalului $[u_{1\min}, u_{1\max}]$, se va alege astfel încât să asigure calitatea plotării.
3. $\sigma_2(u_2)$ pentru $u_2 \in [u_{2\min}, u_{2\max}]$, acest interval reprezentând mulțimea valorilor lui $u_2(x)$, în condițiile de la punctul 1. Pasul de discretizare us_2 al intervalului $[u_{2\min}, u_{2\max}]$, se va alege astfel încât să asigure calitatea plotării.
4. $f_1(x) = \sigma_1(u_1(x))$, $f_2(x) = \sigma_2(u_2(x))$ (relația (16)), pentru $x \in [-v, v]$.

Se vor considera cazurile funcției de activare:

- a) sigmoid bipolar;
- b) treaptă unipolară.

Pentru fiecare din cazurile a) și b), se vor studia următoarele situații:

- (i) $W = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;
- (ii) $W = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $b = \begin{bmatrix} b_1^* \\ b_2^* \end{bmatrix}$;
- (iii) $W = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;
- (iv) $W = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $b = \begin{bmatrix} b_1^* \\ b_2^* \end{bmatrix}$;
- (v) $W = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;
- (vi) $W = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $b = \begin{bmatrix} b_1^* \\ b_2^* \end{bmatrix}$;
- (vii) $W = \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;

$$(viii) \quad \mathbf{W} = \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1^* \\ b_2^* \end{bmatrix};$$

unde valorile w_1^* , b_1^* , w_2^* , b_2^* vor fi precizate în cursul ședinței de aplicații.

Se vor comenta legăturile existente între graficele existente la (i) – (viii), iar atunci când este posibil, aceste legături vor fi utilizate pentru *trasarea directă* a graficelor, fără a mai face apel la programul elaborat în MATLAB. De asemenea, se vor comenta legăturile cu rezultatele obținute la Problema 1.

Notă: Pentru exemplificare putem considera $v = 10$, $vs = 0.1$.

Problema 4

Se consideră o rețea neuronală cu două straturi, cu câte un neuron pe fiecare strat (vezi fig. 5 - cu $n_1 = 1$, $p_1 = n_2 = 1$, $p_2 = 1$). Să se elaboreze un program MATLAB pentru a studia modul de compunere a funcțiilor, considerând la intrarea rețelei, valorile:

$$x = -v : vs : v;$$

unde valorile v și vs vor fi precizate în cursul ședinței de aplicații, pentru fiecare colectiv de lucru în parte.

Se vor prezenta grafic următoarele trei funcții:

1. $y^1(x^1)$ (relația (26)) pentru $x \in [-v, v]$

2. $y^2(x^2)$ (relația (23)) pentru $x^2 \in [y_{\min}, y_{\max}]$, acest interval reprezentând mulțimea valorilor lui $y^1(x^1)$, în condițiile de la punctul 1. Pasul de discretizare ys al intervalului $[y_{\min}, y_{\max}]$ se va alege astfel încât să asigure calitatea plotării.

3. $y^2(x^1)$ (relația (22)) pentru $x \in [-v, v]$.

Se vor considera cazurile funcției de activare:

- a) sigmoid bipolar pe ambele straturi;
- b) treaptă unipolară pe ambele straturi.

Pentru fiecare din cazurile a) și b), se vor considera valorile $w^2 = w^{2*}$, $b^1 = b^{1*}$, $w^2 = w^{2*}$, $b^2 = b^{2*}$, ce vor fi precizate în cursul ședinței de aplicații.

Se vor comenta legăturile cu rezultatele obținute la Problema 1.

Notă: Pentru exemplificare putem considera $v = 10$, $vs = 0.1$.