

ANTRENAREA SUPERVIZATĂ A REȚELELOR FEEDFORWARD CU FUNCȚII DE ACTIVARE LINIARE ÎN PROBLEME DE APROXIMARE LINIARĂ

1. Considerații generale, motivație și obiectiv

Rețelele neuronale feedforward cu un singur strat cu funcții de activare liniare pot fi utilizate pentru a realiza aproximarea liniară a unor funcții definite cu ajutorul unei mulțimi de vectori prototip (eng. *pattern*) și respectiv a unei mulțimi de vectori țintă (eng. *target*). Arhitectura rețelei (adică numărul de neuroni, precum și numărul de intrări) este impusă de dimensiunea vectorilor țintă și respectiv dimensiunea vectorilor prototip.

Prin parcurgerea acestei ședințe de aplicații, studentul își va însuși cunoștințele necesare antrenării supervizate a acestor rețele cu ajutorul algoritmului Widrow-Hoff. Pentru desfășurarea practică a experimentelor se va face apel la funcții disponibile în **Neural Network Toolbox**. Un accent deosebit se va pune pe vizualizarea tridimensională a progresului antrenării în cazul unui neuron cu o singură intrare.

2. Cunoștințele prealabile necesare

- Capitolul „Antrenarea supervizată a rețelelor neuronale”, secțiunea „Rețele feedforward cu funcții de activare liniare” din cursul „Aplicații ale rețelelor neuronale în automatică”.
- Experiență de programare în MATLAB.

3. Breviar teoretic

Utilizarea în problemele de aproximare liniară a rețelelor neuronale feedforward cu funcții de activare liniare se bazează pe antrenarea rețelelor prin algoritmul Widrow-Hoff. Aceste rețele mai poartă denumirea de ADALINE (ADaptive LINEar Element), denumire propusă de Widrow și Hoff pentru că algoritmul de antrenare are efectul adaptării (ADA) valorii parametrilor rețelei cu funcții liniare (LIN) în scopul aproximării.

3.1. Transferul liniar intrare-ieșire și proprietatea de aproximare

Pentru o rețea feedforward cu un singur strat cu n intrări, p neuroni cu funcții de activare liniare, transferul intrare-ieșire este descris prin dependența liniară:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}, \quad \mathbf{W} \in \mathbb{R}^{p \times n}, \quad \mathbf{b} \in \mathbb{R}^p, \quad (1)$$

unde $\mathbf{x} \in \mathbb{R}^n$ notează vectorul de intrare iar $\mathbf{y} \in \mathbb{R}^p$ notează vectorul de ieșire. Acest transfer este parametrizat de ponderile $\mathbf{W} \in \mathbb{R}^{p \times n}$ și deplasările $\mathbf{b} \in \mathbb{R}^p$.

Se consideră o mulțime de N vectori prototip:

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \in \mathbb{R}^n, \quad (2)$$

și o mulțime de N vectori țintă:

$$\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N \in \mathbb{R}^p, \quad (3)$$

care corespund vectorilor prototip în sensul că există o aplicație:

$$\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^p, \quad (4)$$

care asigură:

$$\varphi(\mathbf{x}_i) = \mathbf{z}_i, \quad i = 1, \dots, N. \quad (5)$$

Aplicând funcția (1) pentru cei N vectori prototip $\mathbf{x}_i$, putem scrie:

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{W}\mathbf{x}_1 + \mathbf{b} = [\mathbf{W} \ \mathbf{b}] \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{\Theta} \underline{\mathbf{x}}_1, \\ &\vdots \\ \mathbf{y}_N &= \mathbf{W}\mathbf{x}_N + \mathbf{b} = [\mathbf{W} \ \mathbf{b}] \begin{bmatrix} \mathbf{x}_N \\ 1 \end{bmatrix} = \mathbf{\Theta} \underline{\mathbf{x}}_N, \end{aligned} \quad (6)$$

unde s-au folosit notațiile:

$$\mathbf{\Theta} = [\mathbf{W} \ \mathbf{b}] \in \mathbb{R}^{p \times (n+1)}, \quad (7)$$

$$\underline{\mathbf{x}}_i = \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1}, \quad i = 1, \dots, N. \quad (8)$$

Partajând pe linii matricea parametrilor $\mathbf{\Theta} \in \mathbb{R}^{p \times (n+1)}$ sub forma:

$$\mathbf{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^T \\ \vdots \\ \boldsymbol{\theta}_p^T \end{bmatrix}, \quad \boldsymbol{\theta}_k \in \mathbb{R}^{n+1}, \quad k = 1, \dots, p, \quad (9)$$

se constată că relația ce corespunde unui indice oarecare i în (6) poate fi rescrisă astfel:

$$\mathbf{y}_i = \begin{bmatrix} \boldsymbol{\theta}_1^T \underline{\mathbf{x}}_i \\ \vdots \\ \boldsymbol{\theta}_p^T \underline{\mathbf{x}}_i \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{x}}_i^T \boldsymbol{\theta}_1 \\ \vdots \\ \underline{\mathbf{x}}_i^T \boldsymbol{\theta}_p \end{bmatrix} = \mathbf{A}_i \mathbf{v}, \quad i = 1, \dots, N. \quad (10)$$

În relația (10) matricea $\mathbf{A}_i$ este definită prin:

$$\mathbf{A}_i = \begin{bmatrix} \underline{\mathbf{x}}_i^T & \mathbf{O}_{1 \times (n+1)} & \cdots & \mathbf{O}_{1 \times (n+1)} \\ \mathbf{O}_{1 \times (n+1)} & \underline{\mathbf{x}}_i^T & \cdots & \mathbf{O}_{1 \times (n+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_{1 \times (n+1)} & \mathbf{O}_{1 \times (n+1)} & \cdots & \underline{\mathbf{x}}_i^T \end{bmatrix} \in \mathbb{R}^{p \times p(n+1)}, \quad i = 1, \dots, N, \quad (11)$$

iar $\mathbf{v} \in \mathbb{R}^{p(n+1)}$ este vectorul tuturor parametrilor rețelei:

$$\mathbf{v} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_p \end{bmatrix} \in \mathbb{R}^{p(n+1)}. \quad (12)$$

Într-o scriere globală, cele N relații (10) (toate conținând același vector al parametrilor rețelei, $\mathbf{v} \in \mathbb{R}^{p(n+1)}$) conduc la relația vectorială (compusă din pN egalități scalare):

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \mathbf{v}. \quad (13)$$

Se observă imediat că (13) descrie comportarea rețelei (pentru *valori arbitrare* ale parametrilor rețelei) în cazul când la intrare se prezintă vectorii prototip $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \dots, N$.

Dacă în (13), vectorii $\mathbf{y}_i \in \mathbb{R}^p$, $i = 1, \dots, N$, se înlocuiesc cu vectorii țintă $\mathbf{z}_i \in \mathbb{R}^p$, $i = 1, \dots, N$, și dacă se schimbă membrul stâng cu cel drept, se obține relația:

$$\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{bmatrix}. \quad (14)$$

Această relație poate fi privită ca un sistem algebric liniar de pN ecuații cu $p(n+1)$ necunoscute (necunoscutele fiind parametrii rețelei, grupați în vectorul $\mathbf{v} \in \mathbb{R}^{p(n+1)}$). În sistemul respectiv, matricea coeficienților se construiește din elementele vectorilor prototip $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \dots, N$ (conform celor N egalități (11)), iar vectorul termenilor liberi se construiește din elementele vectorilor țintă $\mathbf{z}_i \in \mathbb{R}^p$, $i = 1, \dots, N$.

După cum este cunoscut din algebra liniară, sistemul (14) posedă întotdeauna o *soluție în sensul celor mai mici pătrate* (unică sau nu), care se va nota prin $\mathbf{v}^* \in \mathbb{R}^{p(n+1)}$. Introducând vectorii de eroare:

$$\mathbf{e}_i = \mathbf{z}_i - \mathbf{y}_i \in \mathbb{R}^p, \quad i = 1, \dots, N, \quad (15)$$

valorile parametrilor rețelei definiți prin cele $p(n+1)$ elemente ale vectorului $\mathbf{v}^* \in \mathbb{R}^{p(n+1)}$ asigură *minimizarea erorii pătratice globale*, adică:

$$0 \leq \sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i \Big|_{\mathbf{v}^*} \leq \sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i \Big|_{\mathbf{v}_{\text{oarecare}}}. \quad (16)$$

Să notăm prin $\mathbf{W}^* \in \mathbb{R}^{p \times n}$ și $\mathbf{b}^* \in \mathbb{R}^p$ valorile parametrilor rețelei corespunzători elementelor vectorului $\mathbf{v}^* \in \mathbb{R}^{p(n+1)}$ (vezi relațiile (7), (9) și (12)). Astfel, pentru aceste valori $\mathbf{W}^*$ și $\mathbf{b}^*$, rețeaua va furniza prin funcția $\mathbf{f}$ din (1) o aproximare liniară, în sensul celor mai mici pătrate, a funcției ϕ (5) (care definește corespondența dintre vectorii prototip și țintă).

Observație:

Rezolvarea sistemului de ecuații algebrice (14), comportă următoarea discuție, care decurge din Teorema Kronecker-Capelli:

1° Dacă:

$$\text{rang} \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} = \text{rang} \left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{z}_1 \\ \vdots & \vdots \\ \mathbf{A}_N & \mathbf{z}_N \end{array} \right] < \min(pN, p(n+1)), \quad (17)$$

atunci sistemul (14) posedă o *infinitate de soluții exacte* $\mathbf{v}^* \in \mathbb{R}^{p(n+1)}$, pentru care vectorii eroare $\mathbf{e}_i$, $i = 1, \dots, N$, definiți prin (15) se anulează și implicit, rezultă:

$$\sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i \Big|_{\mathbf{v}^*} = 0. \quad (18)$$

2° Dacă:

$$\text{rang} \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} = \text{rang} \left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{z}_1 \\ \vdots & \vdots \\ \mathbf{A}_N & \mathbf{z}_N \end{array} \right] = \min(pN, p(n+1)), \quad (19)$$

atunci sistemul (14) posedă o *singură soluție exactă* $\mathbf{v}^*$, pentru care vectorii eroare $\mathbf{e}_i$, $i = 1, \dots, N$, definiți prin (15) se anulează și, implicit, rezultă satisfacerea egalității (18).

3° Dacă:

$$\text{rang} \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} < \text{rang} \left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{z}_1 \\ \vdots & \vdots \\ \mathbf{A}_N & \mathbf{z}_N \end{array} \right], \quad (20)$$

atunci sistemul (14) posedă numai *pseudo-soluții* $\mathbf{v}^*$, sau *soluții în sensul celor mai mici pătrate* conform (16), pentru care avem inegalitatea strictă:

$$\sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i \Big|_{\mathbf{v}^*} > 0. \quad (21)$$

Se disting următoarele două subcazuri:

a) Dacă

$$\text{rang} \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} < \min(pN, p(n+1)), \quad (22)$$

atunci există o *infinitate de pseudo-soluții* $\mathbf{v}^*$ satisfăcând (16).

b) Dacă

$$\text{rang} \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} = \min(pN, p(n+1)), \quad (23)$$

atunci există o *singură pseudo-soluție* $\mathbf{v}^*$ satisfăcând (16).

Această discuție formulată cu privire la vectorul $\mathbf{v}^* \in \mathbb{R}^{p(n+1)}$, se transpune direct în termenii parametrilor rețelei $\mathbf{W}^* \in \mathbb{R}^{p \times n}$ (ponderi) și $\mathbf{b}^* \in \mathbb{R}^p$ (deplasări), parametri care constituie elementele vectorului $\mathbf{v}^* \in \mathbb{R}^{p(n+1)}$.

3.2. Obiectivul antrenării rețelei

Pornind de la mulțimea vectorilor prototip (2) și mulțimea vectorilor țintă (3), prin antrenarea rețelei trebuie să se determine parametrii acesteia $\mathbf{W}^* \in \mathbb{R}^{p \times n}$ și $\mathbf{b}^* \in \mathbb{R}^p$ care să asigure minimizarea erorii pătratică globale, adică satisfacerea inegalității (16). Funcția f definită prin (1) pentru $\mathbf{W} = \mathbf{W}^*$ și $\mathbf{b} = \mathbf{b}^*$ realizează o aproximare liniară în sensul celor mai mici pătrate a funcției φ definită conform (5).

3.3. Algoritmul de antrenare Widrow-Hoff

În baza celor discutate în paragraful 3.1, parametrii rețelei $\mathbf{W}^* \in \mathbb{R}^{p \times n}$ și $\mathbf{b}^* \in \mathbb{R}^p$ pot fi determinați prin rezolvarea sistemului algebric (14) cu pN ecuații și $p(n+1)$ necunoscute. Totuși, din punct de vedere numeric, acest procedeu nu este recomandabil întrucât acuratețea soluției depinde de condiționarea numerică a sistemului (14). Din acest motiv, se preferă pentru antrenare, *algoritmul iterativ Widrow-Hoff* (denumit și *regula delta*), care se bazează pe minimizarea erorii pătratice globale prin metoda gradientului.

Considerând vectorii eroare $\mathbf{e}_i$, $i = 1, \dots, N$, definiți prin (15) ca funcții de parametri rețelei (care sunt conținuți în vectorul $\mathbf{v} \in \mathbb{R}^{p(n+1)}$) se definește funcția obiectiv:

$$J(\mathbf{v}) = \frac{1}{N} \sum_{i=1}^N \mathbf{e}_i^T(\mathbf{v}) \mathbf{e}_i(\mathbf{v}). \quad (24)$$

Minimizarea prin metoda gradientului a lui $J(\mathbf{v})$ (care înseamnă satisfacerea condiției (16)) constă în modificarea vectorului $\mathbf{v} \in \mathbb{R}^{p(n+1)}$ la fiecare iterație, în conformitate cu formula:

$$\mathbf{v}_{\text{new}} = \mathbf{v}_{\text{old}} - \eta \nabla J(\mathbf{v}_{\text{old}}), \quad (25)$$

unde $\eta > 0$ notează *pasul metodei de optimizare*, iar $\nabla J = [\partial J / \partial \mathbf{v}]^T$ notează vectorul gradient, normal la suprafața $J(\mathbf{v})$ definită prin (24). Printr-o exprimare adecvată a vectorului gradient, relația (25) poate fi formulată *direct în termenii parametrilor rețelei* $\mathbf{W}^* \in \mathbb{R}^{p \times n}$ (ponderi) și $\mathbf{b}^* \in \mathbb{R}^p$ (deplasări), furnizând, astfel, *mecanismul de efectuare a unei iterații pentru algoritmul de antrenare Widrow-Hoff*.

Pentru algoritmul de antrenare se stabilesc de către utilizator un *număr maxim de iterații* sau *epoci*, precum și o valoare ε care se dorește a fi atinsă în minimizarea funcției obiectiv (24) (adică un *prag pentru eroarea pătratică globală*). Atragem atenția asupra faptului că funcția obiectiv $J(\mathbf{v})$ definită prin (24) satisface condiția (18), dacă și numai dacă sistemul de ecuații algebrice liniare (14) posedă o soluție exactă (sau soluții exacte). În cazul când relația (18) nu este satisfăcută, atingerea unui prag *oarecare* de eroare ε impus de utilizator *nu este întotdeauna posibilă* (întrucât ne situăm în situația caracterizată prin (21)).

De asemenea, se fixează o valoare pentru pasul metodei de optimizare (η), care, în contextul problemei de antrenare, este referită sub denumirea de *rată de învățare* (eng. *learning rate*).

Algoritmul de antrenare Widrow-Hoff se bazează pe faptul că $J(\mathbf{v})$ este o formă pătratică definită pe spațiul parametrilor $\mathbb{R}^{p(n+1)}$:

$$\begin{aligned} J(\mathbf{v}) &= \frac{1}{N} \sum_{i=1}^N \mathbf{e}_i^T(\mathbf{v}) \mathbf{e}_i(\mathbf{v}) = \frac{1}{N} \sum_{i=1}^N [\mathbf{z}_i - \mathbf{A}_i \mathbf{v}]^T [\mathbf{z}_i - \mathbf{A}_i \mathbf{v}] = \\ &= \frac{1}{N} \sum_{i=1}^N [\mathbf{z}_i^T \mathbf{z}_i - 2 \mathbf{z}_i^T \mathbf{A}_i \mathbf{v} + \mathbf{v}^T \mathbf{A}_i^T \mathbf{A}_i \mathbf{v}] = c + \mathbf{d}^T \mathbf{v} + \frac{1}{2} \mathbf{v}^T \mathbf{R} \mathbf{v}, \end{aligned} \quad (26)$$

unde:

$$c = \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i^T \mathbf{z}_i \in \mathbb{R}, \quad \mathbf{d} = -2 \frac{1}{N} \sum_{i=1}^N \mathbf{A}_i^T \mathbf{z}_i \in \mathbb{R}^{p(n+1)}, \quad \mathbf{R} = 2 \frac{1}{N} \sum_{i=1}^N \mathbf{A}_i^T \mathbf{A}_i \in \mathbb{R}^{p(n+1) \times p(n+1)}, \quad (27)$$

astfel că $\nabla J = \mathbf{d} + \mathbf{R} \mathbf{v}$ și $D^2 J = \mathbf{R}$ (matricea Hessian).

La *fiecare iterație* a algoritmului Widrow-Hoff se parcurg etapele descrise mai jos.

Valorile parametrilor rețelei la începutul unei iterații sunt notate prin $\mathbf{W}_{\text{old}} \in \mathbb{R}^{p \times n}$ (pentru ponderi) și $\mathbf{b}_{\text{old}} \in \mathbb{R}^p$ (pentru deplasări).

Etapa I (Etapa de prezentare)

Se prezintă vectorii prototip $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \dots, N$, sub forma matricei:

$$\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N] \in \mathbb{R}^{n \times N}. \quad (28)$$

Pentru valorile curente ale parametrilor rețelei ($\mathbf{W}_{\text{old}} \in \mathbb{R}^{p \times n}$, $\mathbf{b}_{\text{old}} \in \mathbb{R}^p$) se calculează ieșirile rețelei:

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i) = \mathbf{W}_{\text{old}} \mathbf{x}_i + \mathbf{b}_{\text{old}}, \quad \mathbf{y}_i \in \mathbb{R}^p, \quad i = 1, \dots, N, \quad (29)$$

care se organizează sub forma matricei:

$$\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_N] \in \mathbb{R}^{p \times N} \quad (30)$$

Etapa II (Etapa de verificare)

Se calculează vectorii eroare $\mathbf{e}_i$, $i = 1, \dots, N$, definiți prin (15) ca diferența între vectorii țintă $\mathbf{z}_i \in \mathbb{R}^p$, $i = 1, \dots, N$, și vectorii ieșire la iterația curentă $\mathbf{y}_i \in \mathbb{R}^p$, $i = 1, \dots, N$.

Algoritmul se oprește dacă este îndeplinită una din următoarele condiții:

(χ1) eroarea pătratică globală este inferioară valorii de prag (eng. *threshold*) ε stabilit la startarea algoritmului, adică:

$$\sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i < \varepsilon; \quad (31)$$

sau:

(χ2) a fost atins numărul maxim de iterații stabilit la startarea algoritmului.

La oprire, algoritmul va furniza valorile parametrilor rețelei rezultate din antrenare, care vor fi notați prin $\mathbf{W}_f \in \mathbb{R}^{p \times n}$ (ponderi) și $\mathbf{b}_f \in \mathbb{R}^p$ (deplasări).

Dacă nici una din condițiile ($\chi 1$) sau ($\chi 2$) nu sunt îndeplinite, se continuă cu etapa următoare a iterației curente.

Etapa III (Etapa de actualizare a parametrilor)

Se construiește matricea erorilor:

$$\mathbf{E} = [\mathbf{e}_1 \dots \mathbf{e}_N] \in \mathbb{R}^{p \times N}, \quad (32)$$

cu ajutorul căreia se calculează matricele de actualizare al parametrilor:

- pentru *ponderi*:

$$\mathbf{D}_W = \mathbf{E} \cdot \mathbf{X}^T \in \mathbb{R}^{p \times n}, \quad (33-a)$$

- pentru *deplasări*:

$$\mathbf{D}_b = \mathbf{E} \cdot \underbrace{[1 \dots 1]^T}_{N \text{ elemente}} \in \mathbb{R}^p. \quad (33-b)$$

Se calculează noile valori ale parametrilor:

- pentru *ponderi*:

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + \eta \mathbf{D}_W, \quad (34-a)$$

- pentru *deplasări*:

$$\mathbf{b}_{\text{new}} = \mathbf{b}_{\text{old}} + \eta \mathbf{D}_b. \quad (34-b)$$

Cu aceste calcule se încheie iterația curentă și se trece la o nouă iterație efectuând atribuirile: $\mathbf{W}_{\text{new}} \rightarrow \mathbf{W}_{\text{old}}$, $\mathbf{b}_{\text{new}} \rightarrow \mathbf{b}_{\text{old}}$.

Observații:

1° Inițializarea parametrilor rețelei (adică $\mathbf{W}_{\text{old}}$ și $\mathbf{b}_{\text{old}}$ la prima iterație a algoritmului) se face cu valori arbitrare.

2° Este demonstrat faptul că algoritmul Widrow-Hoff converge, după caz, *către soluția unică* sau *către una din soluțiile* sistemului algebric de ecuații liniare (în accepțiunea discuției din paragraful 3.1). Convergența se realizează în condițiile specifice *metodei gradientului* (ca tehnică de optimizare), fiind puternic dependentă de valoarea ratei de învățare η și anume:

- pentru η inadecvat mic, deplasarea către optim, în spațiul parametrilor, este foarte lentă,
- pentru η inadecvat mare, deplasarea către optim, în spațiul parametrilor, se realizează cu oscilații (eventual cu pierderea stabilității algoritmului).

3° În cazul când algoritmul se încheie prin satisfacerea condiției ($\chi 1$), acuratețea parametrilor furnizați $\mathbf{W}_f \in \mathbb{R}^{p \times n}$ și $\mathbf{b}_f \in \mathbb{R}^p$ depinde de valoarea de prag ε stabilită pentru eroarea pătratică globală. Uzual, condiția ($\chi 1$) devine operantă atunci când poate fi satisfăcută relația (18), adică atunci când valoarea minimă a lui $J(\mathbf{v})$ este 0.

4° Algoritmul Widrow-Hoff, privit ca metodă de căutare a minimului funcției $J(\mathbf{v})$ (24), nu se confruntă cu problema eșuării în minime locale, deoarece $J(\mathbf{v})$ este o formă

pătratică definită pe spațiul parametrilor $\mathbb{R}^{p(n+1)}$ care posedă numai un punct de minim global egal cu 0, sau o infinitate de astfel de puncte (în accepțiunea discuției din paragraful 3.1).

4. Facilități software oferite de Neural Network Toolbox

Funcția **newlin** este folosită pentru crearea unei rețele ADALINE cu un singur strat. Modul de alegere a parametrilor de antrenare se poate observa în urma consultării help-ului aferent. Funcția **maxlinlr** poate fi folosită pentru determinarea ratei de învățare maxime care asigură stabilitatea algoritmului de antrenare.

Pentru simularea rețelei antrenate se poate utiliza funcția **sim**.

În cazul unui neuron cu o singură intrare și funcție de activare liniară, pentru reprezentarea grafică tridimensională a suprafeței erorii pătratice globale (ca funcție de pondere și deplasarea neuronului) se pot utiliza funcțiile **errsurf** și **plotes** (apelate în această ordine).

Funcția **newlind** permite determinarea parametrilor unei rețele ADALINE prin rezolvarea sistemului (14) în sensul celor mai mici pătrate.

5. Problematica propusă pentru studiu

Problema 1

Se va studia și lansa în execuție programul **arna_wh** elaborat în MATLAB care antrenează un neuron ADALINE cu o singură intrare, utilizând algoritmul Widrow-Hoff, pentru vectorii prototip:

$$x_1 = 0, \quad x_2 = 1,$$

și, respectiv, vectorii țintă:

$$z_1 = 0, \quad z_2 = 2.$$

1. Să se determine valorile parametrilor neuronului utilizând funcția **newlind**.

2. Să se determine valorile parametrilor neuronului făcând modificările necesare în programul **arna_wh** în următoarele situații:

- a) - număr de iterații maxime 30,
 - pragul erorii 0.01,
 - rata de învățare dată de funcția **maxlinlr** (se va preciza valoarea furnizată de aceasta);
- b) - număr de iterații maxime 30,
 - pragul erorii 0.000001,
 - rata de învățare dată de funcția **maxlinlr** (se va preciza valoarea furnizată de aceasta);
- c) - număr de iterații maxime 40,
 - pragul erorii 0.01,
 - rata de învățare 0.2;
- d) - număr de iterații maxime 40,
 - pragul erorii 0.01,
 - rata de învățare 0.6;

- e) - număr de iterații maxime 20,
 - pragul erorii 0.01,
 - rata de învățare 0.8.

Pentru fiecare situație se vor preciza valorile selectate inițial pentru parametrii rețelei, valorile rezultate în final pentru parametrii rețelei, numărul de iterații cât a durat optimizarea și valoarea finală a erorii pătratice globale. Se va preciza condiția care determină încheierea antrenării. De asemenea se va schița suprafața erorii pătratice globale și traseul parcurs în planul parametrilor.

În cazul când se consideră relevant, se pot testa mai multe condiții inițiale pentru fiecare din situațiile a) - e).

Problema 2

Se va relua Problema 1 pentru un singur vector prototip $x_1 = 1$ și, respectiv, un singur vector țintă $z_1 = 2$, pentru situațiile 2.a) și 2.e).

Problema 3

Se va relua Problema 1 pentru vectorii prototip:

$$x_1 = -1, x_2 = 0, x_3 = 1,$$

și, respectiv, vectorii țintă:

$$z_1 = -1, z_2 = 0, z_3 = 2,$$

pentru situațiile 2.a), 2.c) și 2.d).

Problema 4

Se consideră amplificatorul operațional din fig. 1 care posedă în reacție rezistența $R = 100\text{ k}\Omega$ și la intrare rezistențele R_1, R_2, R_3 .

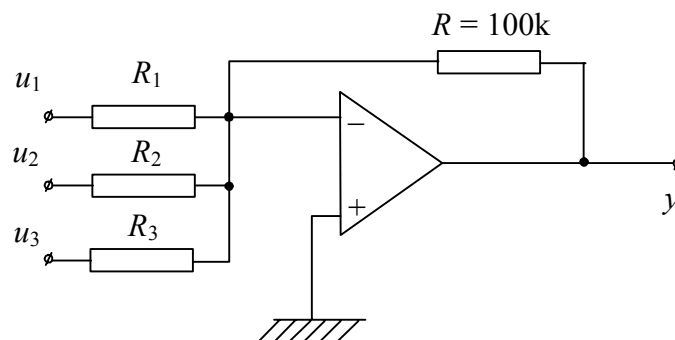


Fig. 1. Schema amplificatorului operațional utilizat în Problema 4.

La intrarea u_1 se aplică o tensiune constantă de 2V. Pe intervalul de timp cuprins între 0 s și 5 s, pe intrările u_2 și u_3 se aplică două semnale de frecvență joasă, care, eșantionate cu perioada $T = 0.01$ s, sunt memorate în fișierul de date *opamp_wh.mat* în vectorii *uuu2* și

respectiv **uuu3**. Semnalul obținut la ieșirea y , eșantionat cu aceeași perioadă, este memorat în vectorul **yyy** din fișierul de date **opamp_wh.mat**.

Utilizând datele din fișierul **opamp_wh.mat** să se determine valorile rezistențelor necunoscute R_1 , R_2 , R_3 prin următoarele metode:

- a) antrenând o rețea ADALINE cu arhitectură adecvată, utilizând funcțiile **newlin** și **train**;
- b) apelând funcția **newlind**.