Laborator 3: Reţele neurale

Reţelele neurale caracterizează ansambluri de elemente de procesare simple, interconectate şi operând în paralel, care urmăresc să interacţioneze cu mediul înconjurător într-un mod asemănător creierelor biologice şi care prezintă capacitatea de a învăţa. Nu există o definiţie general acceptată a acestor tipuri de sisteme, dar majoritatea cercetătorilor sunt de acord cu **definirea reţelelor artificiale ca reţele de elemente simple puternic interconectate prin intermediul unor legături numite interconexiuni prin care se propagă informaţie numerică.**

Originea acestor reţele trebuie căutată în studierea reţelelor bioelectrice din **creier** formate din **neuroni** şi **sinapsele** acestora. Principala trăsătură a reţelelor constă în capacitatea de a *învăţa din exemple într-un mod conexionist,* folosindu-se de experienţa anterioară pentru a-şi îmbunătăţi performanţele.

Reţelele neurale au fost modelate astfel încât să efectueze funcţii complexe în domenii diverse de aplicabilitate, incluzând recunoaşterea, identificarea şi clasificarea formelor, sisteme de vorbire, imagine şi control.

Metodologia reţelelor neurale ne permite să proiectăm sisteme neliniare care recunosc un număr mare de intrări, fiind proiectate nunai pe baza relaţiilor dintre instanţe de tipul intrare-ieşire (adică perechi $\{(x_i, t_i)\}$ formate din vectorul caracteristicilor x_i şi respectiv categoria structurii t_i). Vectorul de intrare x_i poate fi vazut ca un vector care conţine măsurători ale caracteristicilor unui domeniu de interes, iar ieşirea dorită t_i poate reprezenta clasele structurii, valori prezise sau valori ţintă.

Problema pe care ne propunem să o rezolvăm folosind o reţea neurală poate fi formulată astfel:

Dându-se o mulţime de date, denumită mulţime de antrenare, de forma

$$S = \left\{ \left(p^i, t^i \right) \middle| \; p^i \in R^R, t^i \in R^S, i = \overline{1,m} \right\}$$

obiectivul este de a determina o ipoteză (o aplicaţie) *h* dintr-un anumit spaţiu de ipoteze astfel încât $h(p^i) = t^i$ (ipoteza *h* este denumită funcţie de predicţie, deoarece prezice eticheta t_i corespunzătoare vectorul caracteristicilor x_i).

Pocedeul folosit pentru a executa procesul de antrenare se numeşte algoritm de învăţare, care are funcţia de a modifica ponderile sinaptice ale reţelei într-un mod sistematic pentru a atinge obiectivul dorit de proiectare. Există două tipuri importante de învăţare: supervizată şi nesupervizată. Învăţarea supervizată presupune aplicarea unei intrări reţelei, după care se compară ieşirea produsă de reţea cu ieşirea dorită şi se modifică ponderile astfel încât să se minimizeze diferenţa dintre cele două. În învăţarea nesupervizată mulţimea de antrenare constă numai din vectori de intrare. Iar scopul algoritmului este de a produce vectori consistenţi, în sensul că două semnale foarte apropiate să producă răspunsuri identice sau foarte asemănătoare. Astfel perechile (vector de intare, vector de ieşire) similare sunt grupate în clase, proces numit şi clusterizare.

În cazul învăţării supervizate distingem două modalităţi de antrenare:

- antrenarea de tip *bach* constă în modificarea ponderilor şi a bias-ului presupunând ca fiind dată o mulţime de vectori de intrare
- antrenarea *incrementală* modifică ponderile şi bias-ul unei reţele imediat după introducerea unui vector de intrare. Antrenarea incrementală poartă numele şi de antrenare “on-line” sau “adaptivă”.

Caracteristici

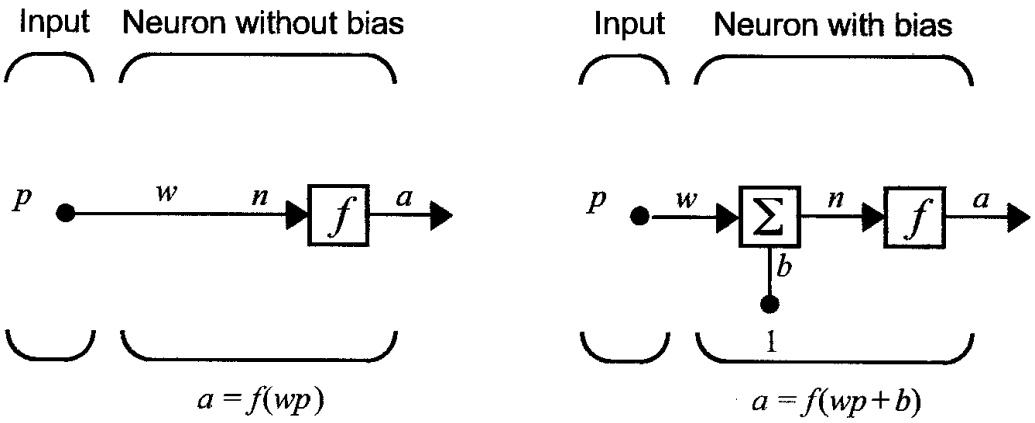
Reţelele neurale artificiale se pot caracteriza pe baza a 3 elemente:

- **modelul** adoptat pentru elementul de procesare individual,
- structura particulară de interconexiuni (**arhitectura**)
- regulile de ajustare a legăturilor (**algoritmii de învăţare**).

Modelul neuronului şi arhitectura acestuia descriu modul în care o reţea transformă datele de intrare în date de ieşire. Această transformare poate fi privită ca pe o compunere.

Modelul neuronului

Structura unui neuron având ca dată de intrare un scalar este următoarea:



Intrarea *p*, transmisă printr-o conexiune, este multiplicată de ponderea *w*, iar produsul wp este argumentul funcţiei de transfer *f*, care produce scalarul de ieşire *a*. Neuronul din partea stângă nu are bias (este nedeplasat), iar cel din dreapta are un bias (deplasare), notat *b*, care este un scalar. Bias-ul poate fi privit ca o simplă adunare la valoarea *wp*, reprezentată prin simbolul ∑ sau ca pe o deplasare a funcţiei *f* la stânga cu o valoare *b*. Bias-ul poate fi considerat ca fiind o pondere corespunzătoare intrării constante 1.

Atât *w*, cât şi *b* sunt parametri reglabili ai neuronului. Ideea centrală a reţelelor neurale constă în faptul că parametrii pot fi modificaţi astfel încât reţeaua să ilustreze un comportament dorit. Astfel, putem antrena reţeaua să realizeze un anumit lucru (problema de clasificare pentru reţeaua de tip perceptron) prin ajustarea ponderilor şi a

bias-ului sau reţeaua însăşi poate ajusta aceşti parametri pentru a produce un anumit comportament.

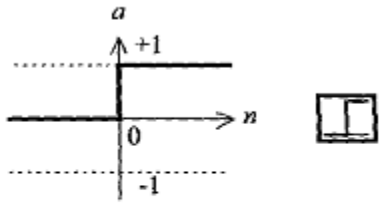
Funcţii de transfer

Multe dintre funcţiile de transfer sunt incluse în toolbox-ul **nnet**. În continuare voi prezenta cele mai utilizate trei funcţii.

1. **Funcţia de transfer hard-limit** limitează ieşirea (output-ul) unui neuron la două valori 0 şi 1, după formula:

$$hardlim(n) = \begin{cases} 0, \text{ dacă } n < 0 \\ 1, \text{ dacă } n \geq 0 \end{cases}$$

Această funcţie este utilizată în reţeaua cu un singur neuron (perceptron) pentru a lua decizii legate de misclasificarea unui punct (problema clasificării).



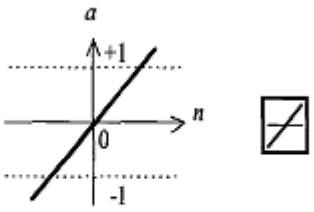
Hard-Limit Transfer Function

Toolbox-ul **nnet** conţine o funcţie numită **hardlim** pentru a obţine funcţia matematică de transfer hard-limit. Codul următor generează graficul funcţiei hard-limit pe intervalul [-5, 5].

```
n = -5:0.1:5;
plot(n,hardlim(n),'*r:');
```

2. **Funcţia de transfer liniară**

   $purelin(n)=n$



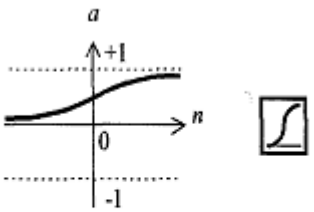
a = purelin(n)

Linear Transfer Function

Neuronii care au ca funcţie de transfer această funcţie sunt folosiţi ca aproximatori liniari în cadrul filtrelor liniare.

3. **Funcţia de transfer sigmoidală** transformă argumentul într-un număr cu valoarea cuprinsă între 0 şi 1; este folosită în reţelele backpropagation, datorită proprietăţii de diferenţiabilitate a funcţiei.

   $logsig(n) = 1 / (1 + \exp(-n))$



a = logsig(n)

Log-Sigmoid Transfer Function

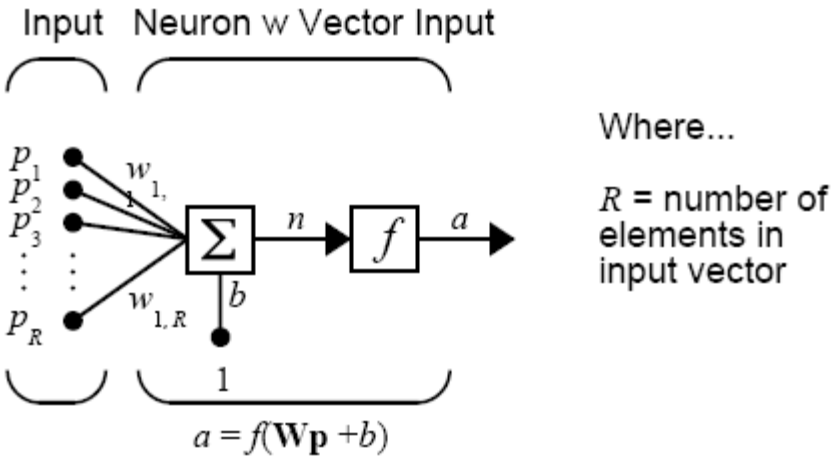
Simbolul din dreapta fiecărui grafic arată reprezentarea funcţiei de transfer; aceste reprezentări vor fi folosite în diagramele reţelelor pentru a sugera tipul funcţiei de transfer folosit.

Exerciţiu: Studiaţi comportarea unui neuron cu o singură intrare folosind programul demonstrativ ***nnd2n1***.

Exerciţiu: Reprezentaţi grafic funcţiile de transfer exemplificate în programul demonstrativ.

Neuron având ca intrare un vector

Un neuron având ca intrare un vector cu R elemente este reprezentat mai jos:



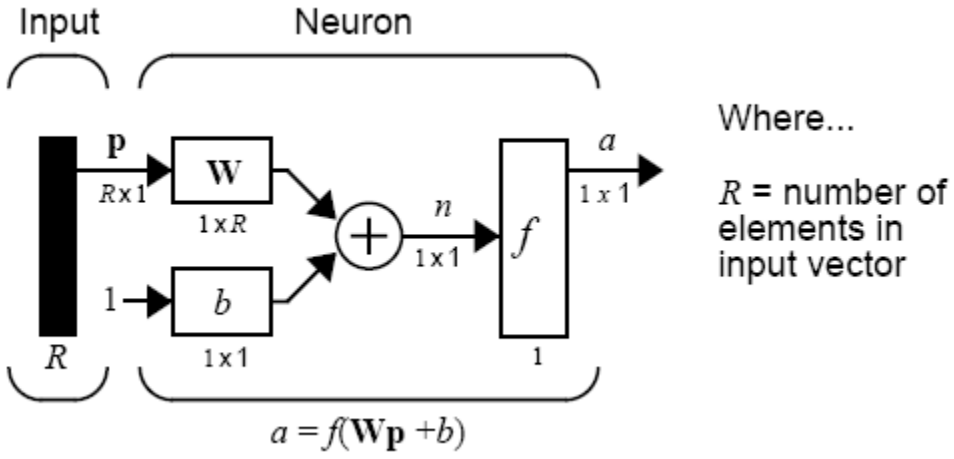
Elementele vectorului de intrare $p_1,...,p_R$ sunt multiplicate cu ponderile $w_{1,1},w_{1,2},...,w_{1,R}$ şi apoi sumate. Suma lor poate fi scrisă ca produsul scalar dintre matricea **W** (matrice cu o linie şi R coloane) şi vectorul **p**. Neuronul are bias-ul *b*, care este adunat intrărilor ponderate, rezultând intrarea reţelei, *n*, care reprezintă argumentul funcţiei de transfer:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + ... + w_{1,R}p_R + b$$

Expresia poate fi scrisă în MATLAB astfel:

$n = W*p + b$

În reprezentarea reţelelor cu mai mulţi neuroni şi mai multe nivele de neuroni, detaliile nu sunt foarte importante şi prin urmare va fi preferată următoarea notaţie pentru a reprezenta un neuron:



p reprezintă vectorul intrărilor şi are dimensiunile R × 1;

W reprezintă matricea ponderilor;

b reprezintă bias-ul;

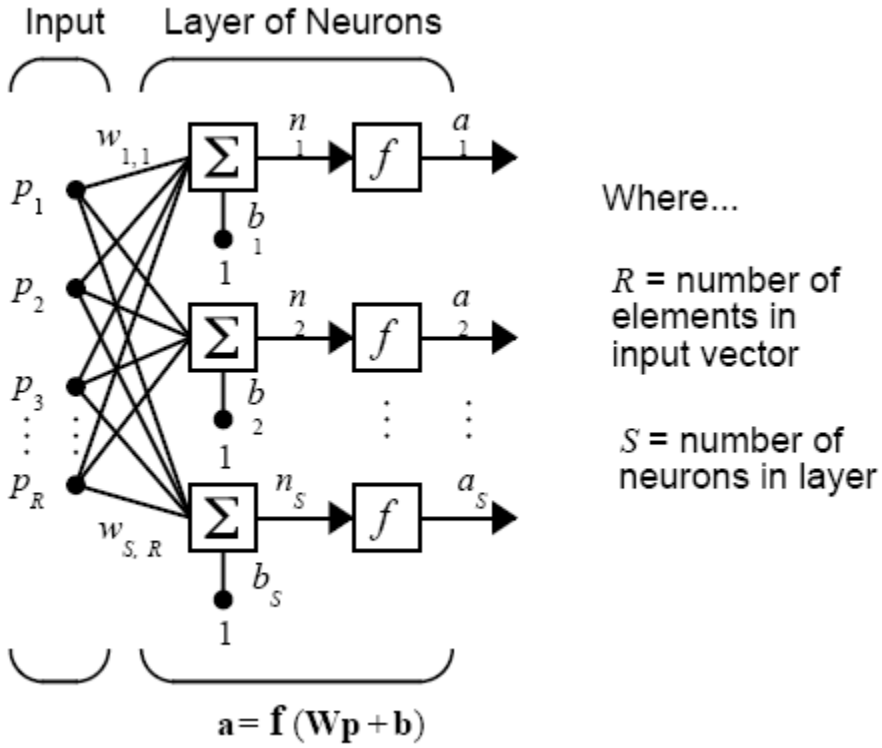
n reprezintă intrarea reţelei, *n*= **W*p** + *b* ;

a este ieşirea reţelei, *a* = *f*(*n*); trebuie făcută observaţia că dacă avem avem o reţea cu mai mult de un neuron ieşirea este un vector.

Exerciţiu: Experimentaţi folosind programul experimental **nnd2n2** neuronul având ca intrare un vector cu două elemente.

Arhitecturi neurale

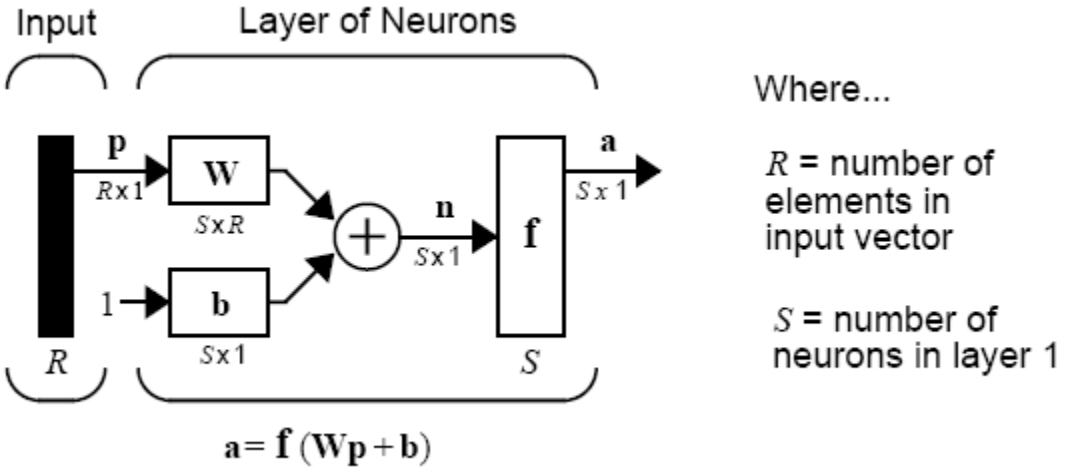
Mai mulţi neuroni pot fi grupaţi pe un nivel, iar o reţea neurală poate conţine unul sau mai multe nivele. Mai jos este reprezentată o reţea cu un singur nivel ce conţine S neuroni:



În reţeaua de mai sus, fiecare element al vectorului de intrare **p** va constitui o intrare pentru fiecare neuron, ponderată de componentele matricei de ponderi **W**. Starea neuronul *i*, *n*(*i*), este dată de produsul scalar dintre intrările vectorului **p** şi coloana *i* a matricei ponderilor. Cele S valori *n*(*i*) determină intrarea reţelei, *n*. Ieşirile nivelului de neuroni sunt reţinute în vectorul *a*. Matricea ponderilor este:

$$\mathbf{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \dots & \dots & \dots & \dots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{pmatrix}$$

Reţeaua de mai sus cu un singur nivel, având S neuroni şi R intrări poate fi reprezentată prescurtat astfel:



Astfel, ***nivelul*** unei reţele neurale este un 4-uplu format din:

- matricea ponderilor,
- bias-ul *b*
- funcţia de integrare (operaţiile de multiplicare şi adunare) şi
- funcţia de transfer *f*.

Input-uri şi nivele

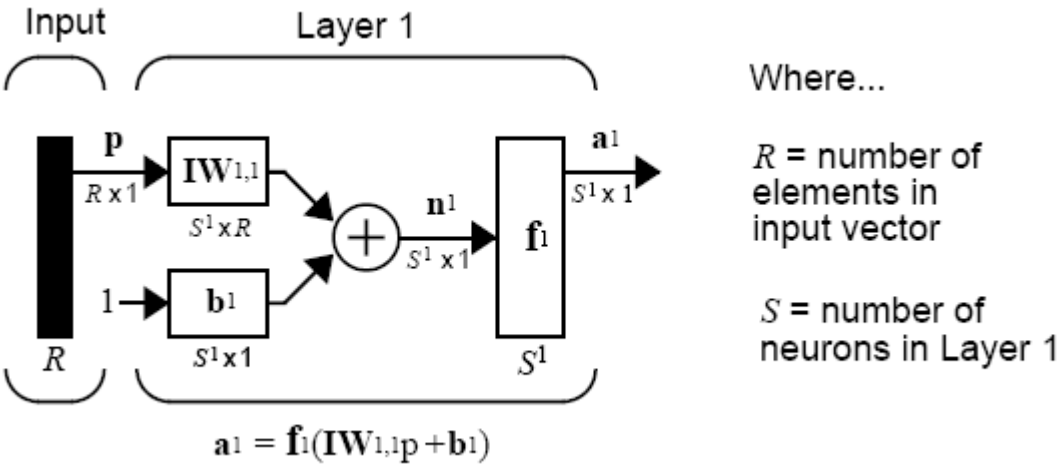
Vom defini imediat reţele având mai multe nivele, de aceea este necesară distincţia dintre matricea ponderilor asociată input-urilor şi matricea ponderilor care conectează două nivele. De asemenea, trebuie să identificăm sursa şi destinaţia ponderilor matricei. Vom numi matricea ponderilor asociată intrărilor *InputWeights*, iar matricea ponderilor asociate arcelor care conectează două nivele de neuroni o vom numi *LayerWeights*. Vom considera că al doilea indice reprezintă sursa, iar primul destinaţia ponderii; de asemenea bias-ul va conţine un indice care indică nivelul în care apare.

Matricea ponderilor asociată intrării de tip *j* şi nivelului *i* este reţinută în celula aflată pe linia *i* şi coloana *j* a unei matrice de celule notată IW de dimensiune numărul_nivelelor × numărul_intrărilor (exemplu: $IW^{1,1}$ este matricea de ponderi

care leagă nivelul 1 (al doilea indice) şi vectorii de intrare de tip 1 (primul indice), adică primul vector de intrare de pe primul nivel, iar b^1 reprezintă celula conţinând vectorul de bias-uri asociate nivelului 1).

În mod similar, **matricea ponderilor asociată nivelului sursă *j* şi nivelului destinaţie *i* este reţinută în celula aflată pe linia *i* şi coloana *j* a unei matrice de celule notată LW de dimensiune numărul_nivelelor × numărul_nivelelor.**

Reţeaua cu un singur nivel de mai jos ilustrează ceea am spus mai sus:



Observaţii:

1. Structurii $IW^{1,1}$ pentru o reţea *net* (creată) îi este asociat următorul cod MATLAB folosind matricea ponderilor:

   $IW^{1,1} \rightarrow$ net.IW{1,1}

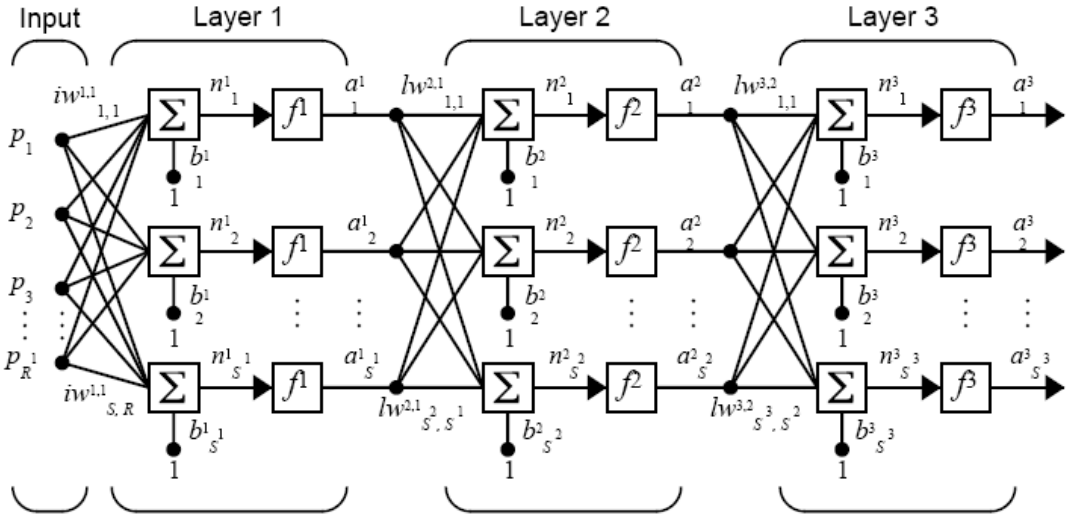
2. Pentru a calcula în MATLAB intrarea funcţiei de transfer vom folosi codul:

   n{1} = net.IW{1,1}*p + net.b{1}

Reţele cu mai multe nivele

O reţea poate avea mai multe nivele, fiecărui nivel corespunzându-i o matrice de ponderi W, un vector bias b şi un vector rezultant a. Pentru a distinge matricele de ponderi pentru fiecare nivel în parte vom adăuga fiecărei variabile ca superscript indicele nivelului din care face parte. Figura de mai jos ilustrează modul în care este

reprezentată o reţea cu trei nivele, fiecare nivel conţinând S^1, S^2, respectiv S^3 neuroni.

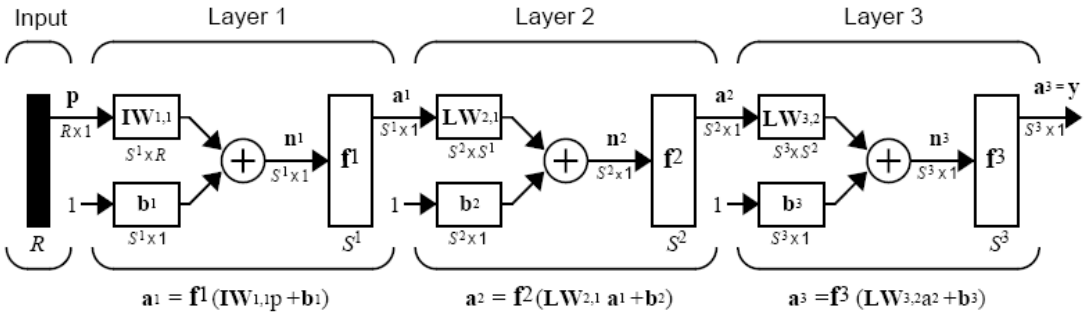


$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1 + \mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\,\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

Trebuie remarcat faptul că ieşirile unui nivel intermediar reprezintă intrările nivelului următor. Astfel, nivelul 2 poate fi analizat ca o reţea având S^1 intrări, S^2 neuroni şi matricea ponderilor de dimensiune $S^2 \times S^1$.

Reprezentarea prescurtată a reţelei de mai sus este următoarea:



$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\,\mathbf{a}^1 + \mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\,\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3 = \mathbf{y}$$

Nivelul care produce ieşirea reţelei se numeşte **nivel de ieşire** (output layer), iar nivelele interioare se numesc **nivele ascunse** (hidden layers). Reţelele neurale cu mai multe nivele sunt foarte utilizate în probleme care necesită aplicarea algoritmului “backpropagation” pe care îl vom discuta mai târziu.