

## Laborator 6

### Rețele neurale

**Rețelele neurale** caracterizează ansambluri de elemente de procesare simple, interconectate și operând în paralel, care urmăresc să interacționeze cu mediul înconjurător într-un mod asemănător creierelor biologice și care prezintă capacitatea de a învăța. Nu există o definiție general acceptată a acestor tipuri de sisteme, dar majoritatea cercetătorilor sunt de acord cu **definirea rețelelor artificiale ca rețele de elemente simple puternic interconectate prin intermediul unor legături numite interconexiuni prin care se propagă informație numerică.**

Originea acestor rețele trebuie căutată în studierea rețelelor bioelectrice din **creier** formate din **neuroni** și **sinapsele** acestora. Principala trăsătură a rețelelor constă în capacitatea de a *învăța din exemple într-un mod conexiunist*, folosindu-se de experiența anterioară pentru a-și îmbunătăți performanțele.

Rețelele neurale au fost modelate astfel încât să efectueze funcții complexe în domenii diverse de aplicabilitate, incluzând recunoașterea, identificarea și clasificarea formelor, sisteme de vorbire, imagine și control.

**Metodologia rețelelor neurale ne permite să proiectăm sisteme neliniare care recunosc un număr mare de intrări, fiind proiectate numai pe baza relațiilor dintre instanțe de tipul intrare-ieșire ( adică perechi  $\{(x_i, t_i)\}$  formate din vectorul caracteristicilor  $x_i$  și respectiv categoria structurii  $t_i$  ). Vectorul de intrare  $x_i$  poate fi văzut ca un vector care conține măsurători ale caracteristicilor unui domeniu de interes, iar ieșirea dorită  $t_i$  poate reprezenta clasele structurii, valori precise sau valori țintă.**

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

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

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

obiectivul este de a determina o ipoteză ( o aplicație )  $h$  dintr-un anumit spațiu de ipoteze astfel încât  $h(p^i) = t^i$  ( ipoteza  $h$  este denumită funcție de predicție, deoarece prezice eticheta  $t_i$  corespunzătoare vectorul caracteristicilor  $x_i$  ).

Pocedeul folosit pentru a executa procesul de antrenare se numește algoritm de învățare, care are funcția de a modifica ponderile sinaptice ale rețelei într-un mod sistematic pentru a atinge obiectivul dorit de proiectare. Există două tipuri importante de învățare: supervizată și nesupervizată. Învățarea supervizată presupune aplicarea unei intrări rețelei, după care se compară ieșirea produsă de rețea cu ieșirea dorită și se modifică ponderile astfel încât să se minimizeze diferența dintre cele două. În învățarea nesupervizată mulțimea de antrenare constă numai din vectori de intrare. Iar scopul algoritmului este de a produce vectori consistenți, în sensul că două semnale foarte apropiate să producă răspunsuri identice sau foarte asemănătoare. Astfel perechile ( vector de intrare, vector de ieșire ) similare sunt grupate în clase, proces numit și clusterizare.

În cazul învățării supervizate distingem două modalități de antrenare:

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

## Caracteristici

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

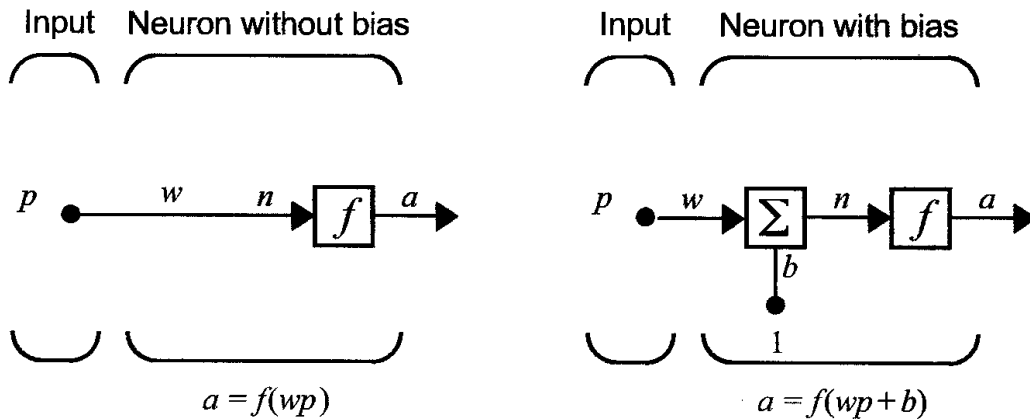
- **modelul** adoptat pentru elementul de procesare individual,
- structura particulară de interconexiuni (**arhitectura**)

- regulile de ajustare a legăturilor (**algoritmii de învățare**).

Modelul neuronului și arhitectura acestuia descriu modul în care o rețea transformă datele de intrare în date de ieșire. 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 funcției de transfer  $f$ , care produce scalarul de ieșire  $a$ . 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  $\Sigma$  sau ca pe o deplasare a funcției  $f$  la stânga cu o valoare  $b$ . Bias-ul poate fi considerat ca fiind o pondere corespunzătoare intrării constante 1.

Atât  $w$ , cât și  $b$  sunt parametri reglabili ai neuronului. Ideea centrală a rețelelor neurale constă în faptul că parametrii pot fi modificați astfel încât rețeaua să ilustreze un

comportament dorit. Astfel, putem antrena rețeaua să realizeze un anumit lucru ( problema de clasificare pentru rețeaua de tip perceptron ) prin ajustarea ponderilor și a bias-ului sau rețeaua însăși poate ajusta acești parametri pentru a produce un anumit comportament.

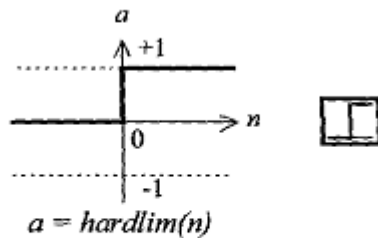
## Funcții de transfer

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

1. **Funcția de transfer hard-limit** limitează ieșirea ( output-ul ) unui neuron la două valori 0 și 1, după formula:

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

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



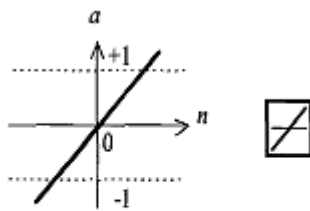
Hard-Limit Transfer Function

Toolbox-ul **nnet** conține o funcție numită **hardlim** pentru a obține funcția matematică de transfer hard-limit. Codul următor generează graficul funcției hard-limit pe intervalul [-5, 5].

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

## 2. Funcția de transfer liniară

$$purelin(n)=n$$



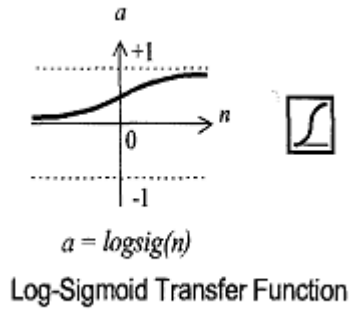
$$a = purelin(n)$$

Linear Transfer Function

Neuronii care au ca funcție de transfer această funcție sunt folosiți ca aproximatori liniari în cadrul filtrelor liniare.

**3. Funcția de transfer sigmoidală** transformă argumentul într-un număr cu valoarea cuprinsă între 0 și 1; este folosită în rețelele backpropagation, datorită proprietății de diferențiabilitate a funcției.

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



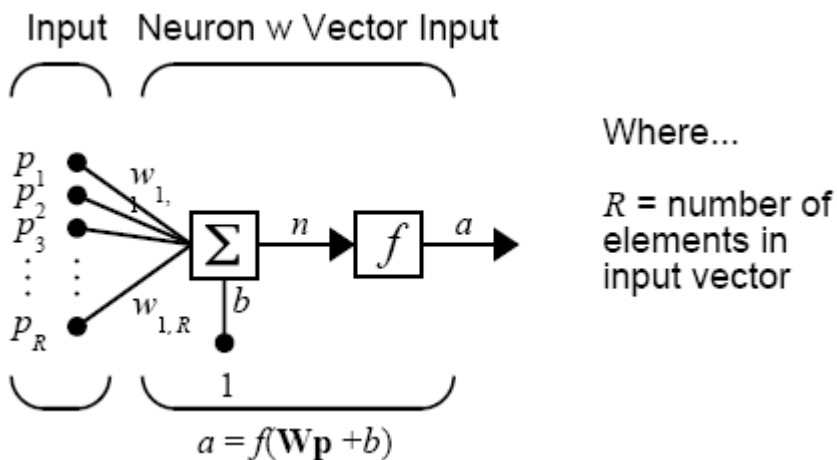
Simbolul din dreapta fiecărui grafic arată reprezentarea funcției de transfer; aceste reprezentări vor fi folosite în diagramele rețelelor pentru a sugera tipul funcției de transfer folosit.

**Exercițiu:** Studiați comportarea unui neuron cu o singură intrare folosind programul demonstrativ *nnd2n1*.

**Exercițiu:** Reprezentați grafic funcțiile 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:



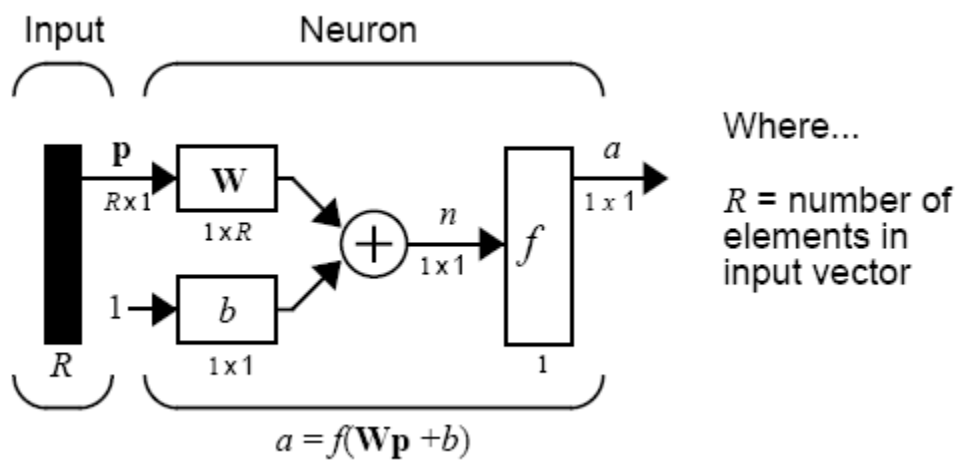
Elementele vectorului de intrare  $p_1, \dots, p_R$  sunt multiplicare cu ponderile  $w_{1,1}, w_{1,2}, \dots, w_{1,R}$  și apoi sumate. Suma lor poate fi scrisă ca produsul scalar dintre matricea  $\mathbf{W}$  ( matrice cu o linie și  $R$  coloane ) și vectorul  $\mathbf{p}$ . Neuronul are bias-ul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

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

Expresia poate fi scrisă în MATLAB astfel:

$$n = W * p + b$$

În reprezentarea rețelelor cu mai mulți neuroni și mai multe nivele de neuroni, detaliile nu sunt foarte importante și prin urmare va fi preferată următoarea notație pentru a reprezenta un neuron:



$\mathbf{p}$  reprezintă vectorul intrărilor și are dimensiunile  $R \times 1$ ;

$\mathbf{W}$  reprezintă matricea ponderilor;

$b$  reprezintă bias-ul;

$n$  reprezintă intrarea rețelei,  $n = \mathbf{W} * \mathbf{p} + b$  ;

$a$  este ieșirea rețelei,  $a = f(n)$ ; trebuie făcută observația că dacă avem o rețea cu mai mult de un neuron ieșirea este un vector.

**Nivelul** unei rețele neurale este un 4-uplu format din:

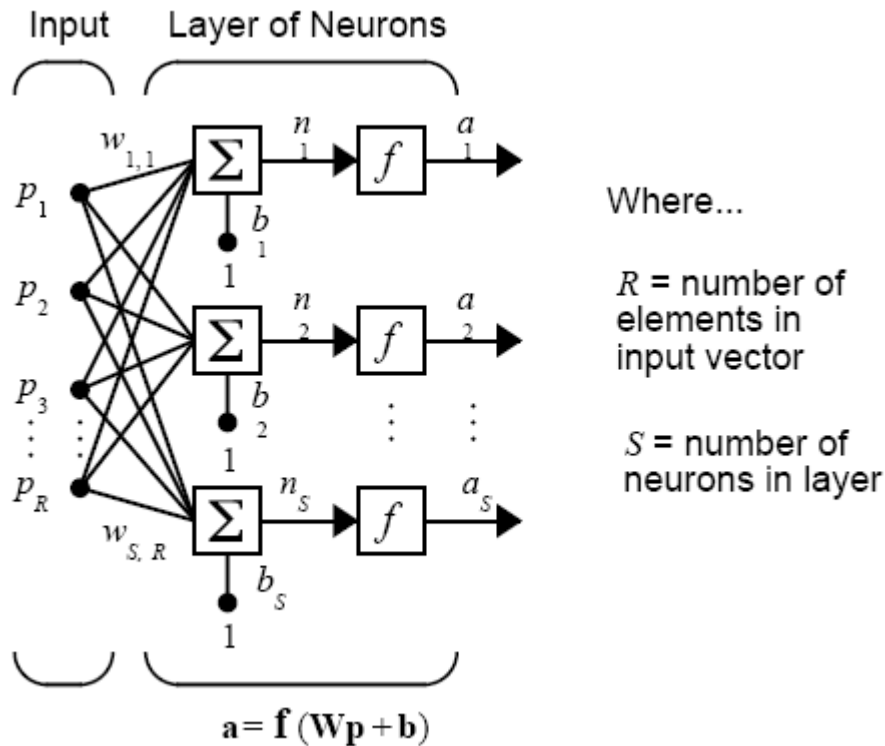
- matricea ponderilor,
- bias-ul  $b$
- funcția de integrare (operațiile de multiplicare și adunare) și
- funcția de transfer  $f$ .

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

## Arhitecturi neurale

Mai mulți neuroni pot fi grupați pe un nivel, iar o rețea neurală poate conține unul sau mai multe nivele. Mai jos este reprezentată o rețea cu un singur nivel ce conține  $S$  neuroni:

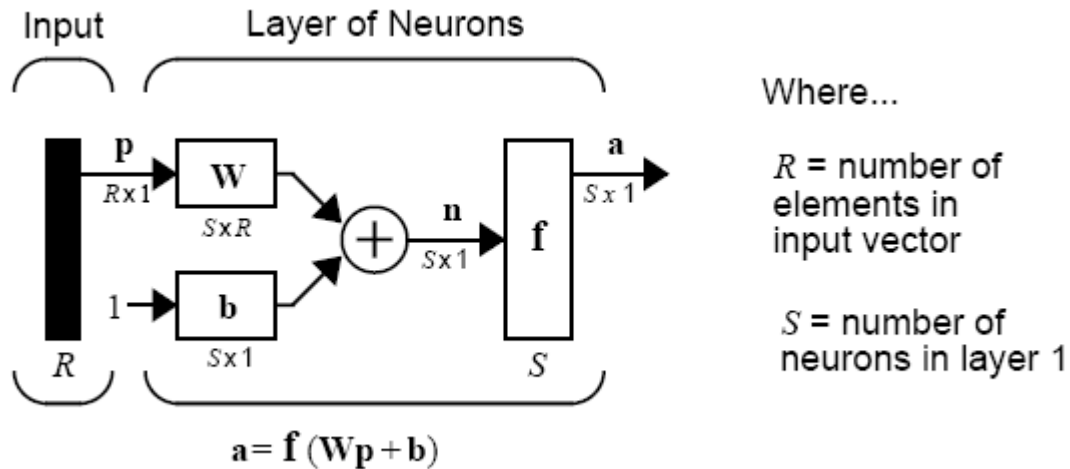




În rețeaua de mai sus, fiecare element al vectorului de intrare  $\mathbf{p}$  va constitui o intrare pentru fiecare neuron, ponderată de componentele matricei de ponderi  $\mathbf{W}$ . Starea neuronul  $i$ ,  $n(i)$ , este dată de produsul scalar dintre intrările vectorului  $\mathbf{p}$  și coloana  $i$  a matricei ponderilor. Cele  $S$  valori  $n(i)$  determină intrarea rețelei,  $n$ . Ieșirile nivelului de neuroni sunt reținute în vectorul  $a$ . Matricea ponderilor este:

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

Rețeaua de mai sus cu un singur nivel, având  $S$  neuroni și  $R$  intrări poate fi reprezentată prescurtat astfel:



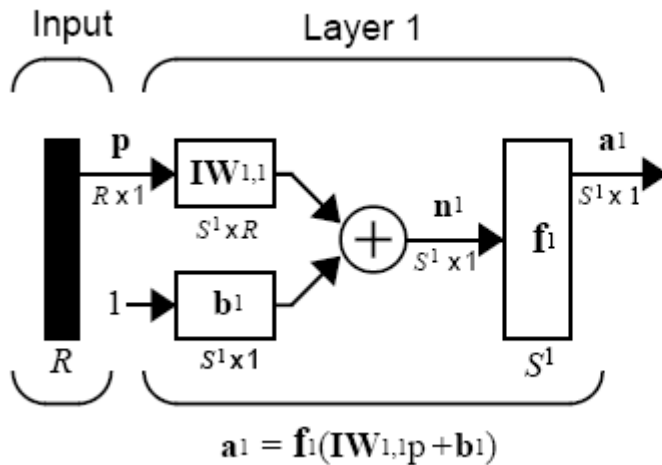
## Input-uri și nivele

Vom defini imediat rețele având mai multe nivele, de aceea este necesară distincția dintre matricea ponderilor asociată input-urilor și matricea ponderilor care conectează două nivele. De asemenea, trebuie să identificăm sursa și destinația 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 destinația ponderii; de asemenea bias-ul va conține un indice care indică nivelul în care apare.

**Matricea ponderilor asociată intrării de tip  $j$  și nivelului  $i$  este reținută în celula aflată pe linia  $i$  și coloana  $j$  a unei matrice de celule notată  $IW$  de dimensiune  $\text{numărul\_nivelelor} \times \text{numărul\_intrărilor}$  ( exemplu:  $IW^{1,1}$  este matricea de ponderi care leagă nivelul 1 ( al doilea indice ) și vectorii de intrare de tip 1 ( primul indice ), adică primul vector de intrare de primul nivel, iar  $b^1$  reprezintă celula conținând vectorul de bias-uri asociate nivelului 1 ).**

În mod similar, **matricea ponderilor asociată nivelului sursă  $j$  și nivelului destinație  $i$  este reținută în celula aflată pe linia  $i$  și coloana  $j$  a unei matrice de celule notată  $LW$  de dimensiune  $\text{numărul\_nivelelor} \times \text{numărul\_nivelelor}$ .**

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



Where...

$R$  = number of elements in input vector

$S$  = number of neurons in Layer 1

### Observații:

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

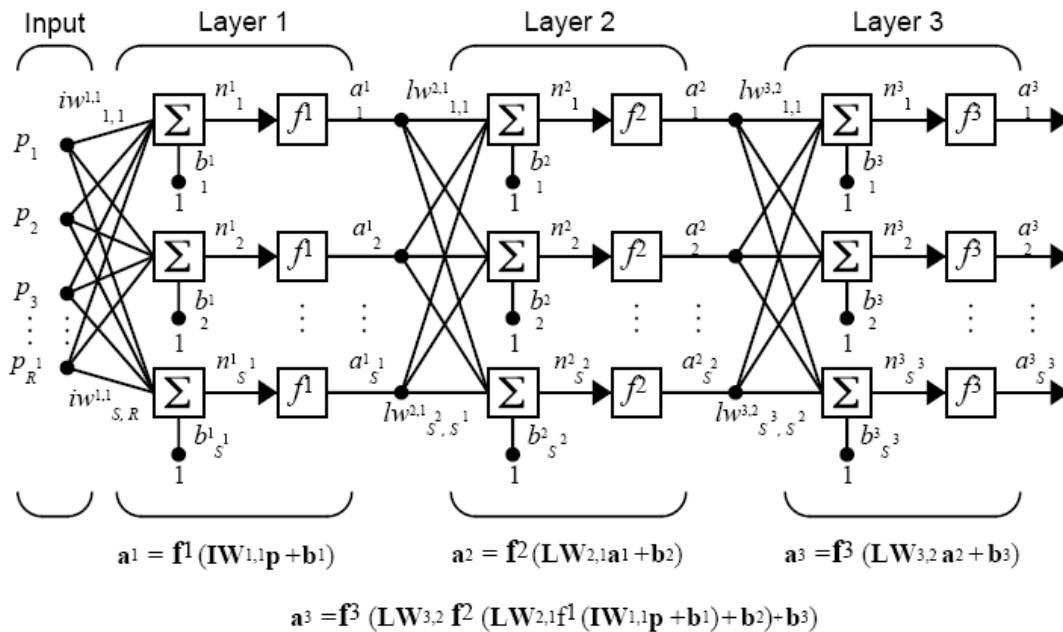
$$IW^{1,1} \rightarrow \text{net.IW}\{1,1\}$$

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

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

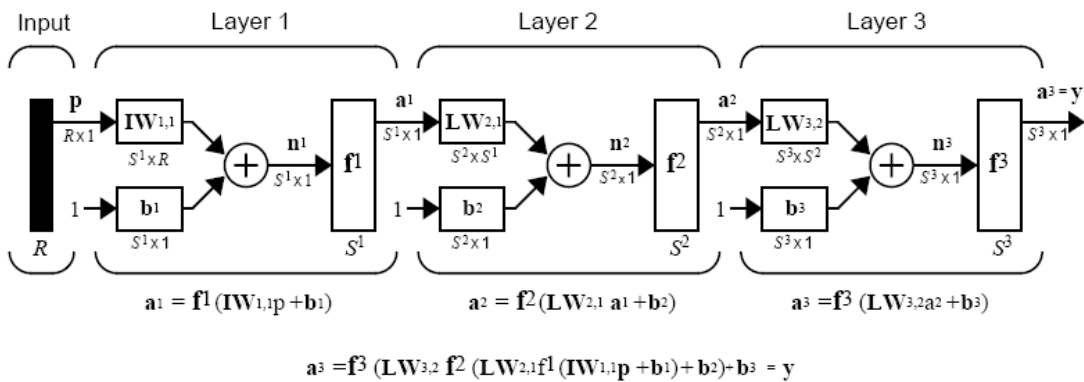
### Rețele cu mai multe nivele

O rețea poate avea mai multe nivele, fiecărui nivel corespunzându-i o matrice de ponderi  $W$ , un vector bias  $b$  și un vector rezultat  $a$ . 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 rețea cu trei nivele, fiecare nivel conținând  $S^1$ ,  $S^2$ , respectiv  $S^3$  neuroni.



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

Reprezentarea prescurtată a rețelei de mai sus este următoarea:



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