```
/*************************************************************************/
/* NEURONE.PRO *  Réseau de 60 neurones en couches  * Christian JOUSSELIN */
/*************************************************************************/
DOMAINS    point=integer*  listR=real*  key=cr;esc;up;down;left;right;other
DATABASE   wji(point,point,real)       ni(point,real,real,real,real)
        status(integer,real)
PREDICATES whileF              menu(char)           initWji(integer)
        initNi(integer)      initLi(point)        initCi(point)
        calcul(integer)      sortie(point)        apprend(integer)
        moment(point,real)   retrom(point,real)   erreur(point,real)
        modN(point,real)     modW(point,real)     sum(listR,real,real)
        f(real,real,real)    df(real,real,real)   f1(real,real,real)
        pos(integer)         position(key,integer)  touche(key,integer)
        afficher(integer)    energie(real)        energiep(real)
        tempm(integer)       filtre(real,char)    rand(real,real)
        Z(real,real)         N(real,real)
/*------------------------------------------------------------ OK 27/10/91 -*/
GOAL       makewindow(10,31,7,"R,seau de neurones en couches",0,0,25,80),
        makewindow( 1,7,31,"Entr",2,1 ,6,7),shiftwindow(10),
        makewindow( 2,7,31,"Cac1",2,19,6,7),shiftwindow(10),
        makewindow( 3,7,31,"Cac2",2,37,6,7),shiftwindow(10),
        makewindow( 4,7,31,"Sort",2,54,6,7),shiftwindow(10),
        makewindow( 5,7,31,"Appr",2,72,6,7),shiftwindow(10),
        makewindow( 6,7,31,"Observations",9,1,12,78),shiftwindow(10),
        makewindow( 7,7,31,"Commandes",21,1,3,78),asserta(status(0,0)),
write("         i:Init  l:Load  s:Save  a:Apprend  c:Calcul  q:Quit"),
shiftwindow(6),whileF,write("Entrez la commande :\n"),readchar(C),menu(C),!.
/*------------------------------------------------------------ OK 27/10/91 -*/
CLAUSES
menu('c') :- write("Initialisez la couche: Entr,e\n"),pos(0),calcul(1),fail.
menu('a') :- not(menu('c')),write("Apprentissage: Initialisez 'Appr'\n"),
          pos(4),energie(_),whileF,apprend(3),energie(E),E<0.001,!,fail.
menu('l') :- write("Chargement de fichier \n"),consult("Neurone.DB"),fail.
menu('s') :- write("Sauvegarde sur fichier\n"),save("Neurone.DB"),fail.
menu('i') :- write("Initialisation\n"),not(initNi(0)),initWji(1),fail.
menu('q').
/*------------------------------------------------------------ OK  5/12/91 -*/
initNi(Co)      :- Co < 5,write("Initialisation des neurones couche:",Co),
                nl,not(initLi([Co,0,0])),Co1 = Co+1, initNi(Co1).

initLi([Co,L,C]) :- L < 4,not(initCi([Co,L,C])),L1 = L+1,initLi([Co,L1,C]).

initCi([Co,L,C]) :- C < 5,rand(4,S),T=0.9,assertz(ni([Co,L,C],T,S,-0.95,0)),
                C1 = C+1,initCi([Co,L,C1]).

initWji(Coi)    :- write("Initialisation des coefficients Wj",Coi),nl,
                Coj = Coi-1,ni([Coj|J],_,_,_,_),ni([Coi|I],_,_,_,_),
                rand(4,Wji),assertz(wji([Coj|J],[Coi|I],Wji)),fail.
initWji(Co)     :- Co < 3,Co1 = Co+1,initWji(Co1).
/*------------------------------------------------------------ OK 23/10/91 -*/
calcul(4)       :- !.
calcul(Co)      :- write("Calcul des neurones couche:",Co),nl,
                ni([Co|I],_,_,_,_),sortie([Co|I]),fail.
calcul(Co)      :- not(afficher(Co)),shiftwindow(6),Co1 = Co+1,calcul(Co1).
/*------------------------------------------------------------ OK 27/11/91 -*/
apprend(0)      :- !,calcul(1).
apprend(Co)     :- write("Calcul des erreurs couche :",Co),nl,
                ni([Co|I],_,_,_,_),erreur([Co|I],E),modN([Co|I],E),fail.
apprend(Co)     :- Co1 = Co-1,apprend(Co1).
/*------------------------------------------------------------ OK 27/11/91 -*/
sortie(Ni)      :- findall(Mji,moment(Ni,Mji),L),retract(ni(Ni,T,S,_,D)),!,
                S1 = -S,sum(L,S1,V),f(T,V,Oi),asserta(ni(Ni,T,S,Oi,D)).
/*------------------------------------------------------------ OK 30/10/91 -*/
```

```
/*************************************************************************/
/* NEURONE.PRO *  Réseau de 60 neurones en couches  * Christian JOUSSELIN */
/*************************************************************************/
Déclarations des types point, liste de réels et touches.
Déclarations des él,ments de la base de données.
  wji(Nj;c-1,Ni;c,wji;c) Ni(Ni,Temp,Seuil,Sortie,Erreur) status(It,.,Energ.)
Déclarations des prédicats









/*-----------------------------------------------------------------------*/
Boot          Fenêtre principale   :"Réseau de neurones en couches"
              Fenêtre neurones     :"Entr"
              Fenêtre neurones     :"Cac1"
              Fenêtre neurones     :"Cac2"
              Fenêtre neurones     :"Sort"
              Fenêtre apprentisage :"Appr"
              Fenêtre observation  :"Observations"
              Fenêtre commandes    :"Commandes"
              Liste des commandes
              Lancement de l'interpréteur de commande
/*-----------------------------------------------------------------------*/
Interpr,teur de commande:
     'c'  :  Lancement du mode "Calcul" du réseau de neurones.
     'a'  :  Lancement du mode "Apprentissage" du réseau de neurones
             avec arr^t si l',nergie est < à 0.001.
     'l'  :  Chargement de la base de donn,es à partir de "Neurone.DB".
     's'  :  Sauvegarde de la base de données dans "Neurone.DB".
     'i'  :  Initialisation de la base de données du réseau de neurones.
     'q'  :  Sortie du programme.
/*-----------------------------------------------------------------------*/
Initialisation des neurones couche par couche + zone d'apprentissage.


Initialisation des neurones ligne par ligne.

Initialisation des neurones colonnes par colonnes,
              avec Température = 0,9 et Sortie = -0,95 et un Seuil aléatoire.

Initialisation des coefficients Wji
              avec une valeur aléatoire.


/*-----------------------------------------------------------------------*/
Mode calcul: arr^t du calcul … la couche nø4.
              Calcul des neurones de la couche: Co
              par appel au pr,dicat: sortie.
              Affichage des nouvelles valeurs de la couche Co et poursuite.
/*-----------------------------------------------------------------------*/
Mode apprentissage: arrêt de l'apprentissage à la couche n°0.
              Calcul des erreurs de la couche: Co
              par appel au prédicat: erreur.
              Poursuite de l'apprentissage sur la couche précédente.
/*-----------------------------------------------------------------------*/
Définition de la sortie d'un neurone par le calcul de la somme de moments
              - seuil, puis filtrage.
```

```
moment(Ni,Mji)   :- wji(Nj,Ni,Wji),ni(Nj,_,_,Oj,_),Mji = Wji*Oj.

retrom(Ni,MDi)   :- wji(Ni,Nj,Wij),ni(Nj,_,_,_,Dj),MDi = Wij*Dj.
/*------------------------------------------------------------ OK 27/11/91 -*/
erreur([3|I],Er) :- ni([4|I],_,_,A,_),ni([3|I],_,_,O,_),Er = O-A,!.
erreur(Ni,Er)    :- findall(MD,retrom(Ni,MD),LMD),sum(LMD,0,Er).
/*------------------------------------------------------------ OK 27/11/91 -*/
modN(N,Er)       :- retract(ni(N,T,S,O,_)),!,f1(T,O,V),df(T,O,O1),
                    Di = O1*Er,S1 = 0.9*S + 1.1*Di,T1 = 0.9*T - 1.1*Di*V,
                    Z(T1,T2),asserta(ni(N,T2,S1,O,Di)),modW(N,Di).

modW([Coi|I],Di) :- Coj = Coi-1,ni([Coj|J],_,_,Oj,_),
                    retract(wji([Coj|J],[Coi|I],Wji)),
                    Wji1 = 0.9*Wji - 1.1*Di*Oj,
                    asserta(wji([Coj|J],[Coi|I],Wji1)),fail.
/*------------------------------------------------------------ OK 27/11/91 -*/
energie(E)       :- retract(status(N,_)),!,findall(Ep,energiep(Ep),LE),
                    N1 = N+1,sum(LE,0,E1),E = E1/2,asserta(status(N1,E)),
                    write("It,ration ",N,". Energie = ",E),nl,tempm(1).

energiep(Ep)     :- ni([4|I],_,_,A,_),ni([3|I],_,_,O,_),Ep = (O-A)*(O-A).
/*------------------------------------------------------------ OK  5/12/91 -*/
tempm(4)         :- !.
tempm(Co)        :- findall(T,ni([Co|_],T,_,_,_),LT),sum(LT,0,S),Tm = S/20,
                    write("Couche ",Co,". Temp,rature moyenne = ",Tm),nl,
                    Co1 = Co + 1,tempm(Co1).
/*------------------------------------------------------------ OK  5/12/91 -*/
f(T,V,O)         :- V1 = exp(-V/T), O1 = (1-V1)/(1+V1),N(O1,O).
df(T,O,O1)       :- O1 = (1-O*O)/(2*T).
f1(_,O,V)        :- V  = ln((1-O)/(1+O)).
/*------------------------------------------------------------ OK 27/10/91 -*/
afficher(Co)     :- W = Co+1,shiftwindow(W),ni([Co,L,C],_,_,O,_),
                    filtre(O,M),scr_char(L,C,M),fail.

filtre(O,' ')    :- O < -0.6,!. filtre(O,'°') :- O < -0.2,!.
filtre(O,'±')    :- O <  0.2,!. filtre(O,'²') :- O <  0.6,!.  filtre(_,'Û').
/*------------------------------------------------------------ OK 27/10/91 -*/
pos(Co)          :- not(afficher(Co)),whileF,touche(K,0),position(K,Co),!.

position(up,_)   :- cursor(L,C),L1 = L+3, L2 = L1 mod 4,cursor(L2,C),!,fail.
position(down,_) :- cursor(L,C),L1 = L+1, L2 = L1 mod 4,cursor(L2,C),!,fail.
position(left,_) :- cursor(L,C),C1 = C+4, C2 = C1 mod 5,cursor(L,C2),!,fail.
position(right,_):- cursor(L,C),C1 = C+1, C2 = C1 mod 5,cursor(L,C2),!,fail.
position(cr,Co)  :- cursor(L,C),retract(ni([Co,L,C],T,S,O,D)),O1 = -O,
                    asserta(ni([Co,L,C],T,S,O1,D)),filtre(O1,M),
                    scr_char(L,C,M),!,fail.
position(esc,_)  :- shiftwindow(6).
/*------------------------------------------------------------ OK 27/10/91 -*/
touche(Key,0)    :- readchar(C),char_int(C,V),touche(Key,V),!.
touche(up,72)    :- !.       touche(down,80)  :- !.    touche(cr,13)  :- !.
touche(left,75)  :- !.       touche(right,77) :- !.    touche(esc,27) :- !.
touche(other,_).
/*------------------------------------------------------------ OK 26/10/91 -*/
sum([],V,V).       sum([V|T],A,Vs) :- S = A+V,sum(T,S,Vs).
whileF.            whileF :- whileF.

rand(N,W)        :- random(Y), X = (Y - 0.5)/N,Z(X,W).

Z(X,0.05)        :- X<+0.05,X>=0,!.   Z(X,-0.05) :- X>-0.05,X<0,!.   Z(X,X).
N(X,0.95)        :- X>+0.95,!.        N(X,-0.95) :- X<-0.95,!.       N(X,X).
/*--------------------------------- FIN ---------------------- OK 27/11/90 -*/
```

```
/*-------------------------------------------------------------------------*/
Définition du moment:       Mji = Wji*Oj.

Définition du rétro-moment: MDi = Wij*Dj.
/*-------------------------------------------------------------------------*/
Définition de l'erreur couche nø4.
Définition de l'erreur pour les autres couches.
/*-------------------------------------------------------------------------*/
Formules de modification des coefficients liés aux neurones:
            S1 = 0.9*S + 1.1*Di et T1 = 0.9*T - 1.1*Di*V
            puis arrondis de la valeur trouvée pour la température.

Formule de modification du coefficient synaptique
            Wji1 = 0.9*Wji - 1.1*Di*Oj.


/*-------------------------------------------------------------------------*/
Calcul de l'énergie partielle du réseau de neurones
            et affichage du nombre d'itération et de la valeur de cette
            énergie dans la fen^tre d'observation.

Définition de l'énergie partielle locale Ep = (O-A)ý.
/*-------------------------------------------------------------------------*/
Calcul de la température moyenne des neurones avec arrêt à la couche n°4,
            la température moyenne est la somme des températures divisée
            par le nombre de neurone par couche: 20.

/*-------------------------------------------------------------------------*/
Définition de la fonction de filtrage
Définition de fonction auxilliaire
Définition de fonction auxilliaire
/*-------------------------------------------------------------------------*/
Affichage d'une couche de neurone à l'écran
            sous forme semi-graphique.

Transformation de la sortie d'un neurone:
            forme numérique => forme semi-graphique.
/*-------------------------------------------------------------------------*/
Gestionnaire de clavier pour les fenêtres "neurones":

            gestion de la touche 'up'   : déplacement vers le haut.
            gestion de la touche 'down' : déplacement vers le bas.
            gestion de la touche 'left' : déplacement vers la gauche.
            gestion de la touche 'right': déplacement vers la droite.
            gestion de la touche 'cr':
                    inversion de la valeur de sortie du neurone.

            gestion de la touche 'esc': sortie du gestionnaire.
/*-------------------------------------------------------------------------*/
Gestionnaire de touche.



/*-------------------------------------------------------------------------*/
Sommateur.
Prédicat: Tant que faux.

Election d'une valeur aléatoire comprise entre -1/(4*N) et 1/(4*N).

Arrondi autour de 0.
Ecrêteur au dela de +0,95 et en dessous de -0,95.
```