Appendix

IT2FNN-1 (Source Code)

```
function [Y,Yl,Yr,sigma,M1,M2,C,S,c1,c2,RMSE,MSE,MAE,ME,MAPE,
MPE] = TRAINIT2TSKNNGMF1(X,D,sigma,M1,M2,C,S,alpha,TOL,ITERMAX)
%
[L,n] = size(X) ;
[N,~] = size(M1);
OK   = false;
iter  = 0;
while (~OK),
  for i=1:L % data
    xe  = [X(i,:) D(i)];
    Z   = [xe' ; abs(xe')];
    UU=[];
    LL=[];
    for j=1:N % rules
      Uu=1;
      Ll=1;
      for m=1:n % variables
        Param=[sigma(j,m),M1(j,m),M2(j,m)];
        mf=evalimftype2(X(i,m),Param,'igaussmtype2');
        ll = mf.LU{1}(:,1);
        uu = mf.LU{1}(:,2);
        Uu=Uu*uu;
        Ll=Ll*ll;
      end % variables
      UU=[UU,Uu];
      LL=[LL,Ll];
    end % reglas
    %
    c2  = [C +S]*Z;
    c1  = [C -S]*Z;
    %
```


J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2

```
    [r_out,I2l,I2u,wr]= rightpoint(c2',LL,UU);
    [l_out,I1u,I1l,wl]= leftpoint(c1',LL,UU);
    %
    Yl(i)=l_out;
    Yr(i)=r_out;
    Y(i) =(l_out+r_out)/2;
    e(i) =D(i)-Y(i);

    % Seleciona las MFs que contribuyen al punto a la derecha (yr)
    ME10=M1;
    ME20=M2;
    sigma0=sigma;
    LE=length(I2u); % numero de fu(I2u)
    for t=1:LE % fu(I2u)
      for k=1:n % variables
        if X(i,k)<M1(I2u(t),k)
          l=I2u(t);
          M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
        elseif X(i,k)>M2(I2u(t),k)
          l=I2u(t);
          M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
^2)/(sigma0(l,k)^3))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
        end
      end % variables
    end % fu(I2u)
    %
    LE=length(I2l); % numero de fl(I2l)
    %
    for t=1:LE % numero de fl(I2l)
      for k=1:n % variables
        if X(i,k)<(M1(I2l(t),k)+M2(I2l(t),k))/2
          l=I2l(t);
          M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
          *(c2(l)-r_out)*wr(l)/sum(wr);
          sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
```

```
^2)/(sigma0(l,k)^3))...
            *(c2(l)-r_out)*wr(l)/sum(wr);
          else
            l=I2l(t);
            M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))...
            *(c2(l)-r_out)*wr(l)/sum(wr);
            sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))...
            *(c2(l)-r_out)*wr(l)/sum(wr);
          end
        end % variables
      end % fl(I2u)
      % Seleciona las MFs que contribuyen al punto a la izquierda (yl)
      LE=length(I1l); % numero de fl(I1l)
      for t=1:LE % fl(I1l)
        for k=1:n % variables
          if X(i,k)<(M1(I1l(t),k)+M2(I1l(t),k))/2
            l=I1l(t);
            M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
            sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
^2)/(sigma0(l,k)^3))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
          else
            l=I1l(t);
            M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
            sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
          end
        end % variables
      end % fl(I1l)
      %
      LE=length(I1u); % numero de fu(I1u)
      %
      for t=1:LE % fu(I1u)
        for k=1:n % variables
          if  X(i,k)< M1(I1u(t),k)
            l=I1u(t);
            M1(l,k)=M1(l,k)+alpha*e(i)*0.5*((X(i,k)-ME10(l,k))/(sigma0
(l,k)^2))...
```

```
            *(c1(l)-l_out)*wl(l)/sum(wl);
            sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME10(l,k))
^2)/(sigma0(l,k)^3))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
          elseif X(i,k)> M2(I1u(t),k)
            l=I1u(t);
            M2(l,k)=M2(l,k)+alpha*e(i)*0.5*((X(i,k)-ME20(l,k))/(sigma0
(l,k)^2))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
            sigma(l,k)=sigma(l,k)+alpha*e(i)*0.5*(((X(i,k)-ME20(l,k))
^2)/(sigma0(l,k)^3))...
            *(c1(l)-l_out)*wl(l)/sum(wl);
          end
        end % variables
      end % fu(I1u)
      %
      for l=1:N % reglas
        for k=1:n % variables
          if sigma(l,k) < 0
            sigma(l,k)=abs(sigma(l,k));
          end
        end
      end
      %
      % fa1=wl/sum(wl); % wl = {fu(1),...,fu(L),fl(L+1),...,fl(N)} = {fu(I1u),fl
(I1l)}
      % fa2=wr/sum(wr); % wr = {fl(1),...,fl(R),fu(R+1),...,fu(N)} = {fl(I2l),fu
(I2u)}
      %
      fa1=wr'/sum(wr);
      fa2=wl'/sum(wl);
      % Termino de constantes de la funcion lineal
      C(:,n+1)=C(:,n+1)+alpha*e(i)*(fa1+fa2)/2;
      S(:,n+1)=S(:,n+1)+alpha*e(i)*(fa1-fa2)/2;
      for j=1:n, % variables
        C(:,j)=C(:,j)+alpha*e(i)*X(i,j)   *(fa1+fa2)/2;
        S(:,j)=S(:,j)+alpha*e(i)*abs(X(i,j))*(fa1-fa2)/2;
      end % variables
      % Reconstruye las funciones de membresia tipo-2 por intervalos
      S=abs(S);
      ME1=[];
      ME2=[];
      for t=1:N % reglas
```

```
        P=[M2(t,:)',M1(t,:)'];
        m2=max(P);
        m1=min(P);
        ME1=[ME1,m1'];
        ME2=[ME2,m2'];
      end % reglas
      M1=ME1';
      M2=ME2';
    end % end for i (datos)

    iter = iter + 1;

    % Calcular errores RMSE, MSE, ME, MAE, MPE, MAPE
    RMSE(iter) = sqrt(mse(e));
%   RMSE(iter)=errperf(D',Y,'rmse');
    MSE(iter)=errperf(D',Y,'mse');
    ME(iter)=sum(D'-Y)/size(D,1);
    MAE(iter)=errperf(D',Y,'mae');
    MPE(iter)=(sum((D'-Y)./D')/size(D,1))*100;
    for p=1:length(D),
     if (isnan(D(p,1))) || (D(p,1)==0),
       D(p,1)=1;%D(p-1,1);
     end
    end
    MAPE(iter)=errperf(D',Y,'mape');

    fprintf('*** Epochs %d *** RMSE %f *** MSE %f *** ME %f *** MAE %
f ***** MPE %f ***** MAPE %f *****\n', iter, RMSE(iter),MSE(iter),ME
(iter),MAE(iter),MPE(iter),MAPE(iter));

    if (iter==ITERMAX)
      %% Plot evolution error RMSE
      h = figure;
      plot(RMSE,'-
r','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','r',
'MarkerSize',8);
      axis([-inf inf -0.05 .4]);
      xlabel('Epochs','fontsize',8,'fontweight','b','color','k');
      ylabel('Errors','fontsize',8,'fontweight','b','color','k');
      legend(['\fontsize{12}RMSE..: ',num2str(RMSE(iter))],'loca-
tion','best');
    %   legend(['Best..: ', num2str(BestResultMinIter
(Iter,1))],'location','best');
```

```
    title('Plot the Error (RMSE) Evolution of the IT2FNN-1');
    % Best Errors (MSE) for each Evolution in the GAs
    print(h, '-dpng','-r150',[folders,'EvolutionRMSE']);
  %   drawnow;
    close(h);

    %% Plotear evolucion de los errores por Iteracion Metrics (RMSE, MSE,
MAE y ME)
    h=figure;
    plot(RMSE,'-
r','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','r',
'MarkerSize',8); hold on
    plot(MSE,'-
b','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g',
'MarkerSize',8);
    plot(MAE,'-.
g','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','b',
'MarkerSize',8);
  %   plot(ME,'-
cs','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','b',
'MarkerSize',8);
    axis([-inf inf -0.05 .4]);
    hold off
    xlabel('Epochs','fontsize',8,'fontweight','b','color','k');
    ylabel('Errors','fontsize',8,'fontweight','b','color','k');
    legend(['\fontsize{12}RMSE..: ',num2str(RMSE(iter))],['\fontsize
{12}MSE..: ',num2str(MSE(iter))],['\fontsize{12}MAE..: ',num2str(MAE
(iter))],'location','best');
  %   legend(['- STD..: ',num2str(BestResultMinIterSTDMen(Iter))],
['Best..: ',num2str(BestResultMinIter(Iter,1))],['+ STD..: ',num2str
(BestResultMinIterSTDMas(Iter))],'location','best');
    title
('Comparison the Error Metrics for the Performance of the IT2FNN-1');
    print(h, '-dpng','-r150',[folders,'BestMetricsPlotEvolution']);
  %   drawnow;
    close(h);

    %% Plotear evolucion de los errores por Iteracion Metrics (MAPE y MPE)
    h=figure;
  %   plot(MPE,'-
ro','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','r',
'MarkerSize',8); hold on
    plot(MAPE,'-
b','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g',
'MarkerSize',8);
```

```
  %   hold off
    xlabel('Epochs','fontsize',8,'fontweight','b','color','k');
    ylabel('Errors','fontsize',8,'fontweight','b','color','k');
  %   legend('\fontsize{12}MAPE','location','best');
    legend(['\fontsize{12}MAPE..: ',num2str(MAPE(iter))],'loca-
tion','best');
    title('Plot the Error (MAPE) Evolution of the IT2FNN-1');
    print(h, '-dpng','-r150',[folders,'BestMAPEEvolution']);
  %   drawnow;
    close(h);
  end

  % fprintf(1,'%6i %12.8f\n',iter,rmse(iter));
  if iter >= ITERMAX
    OK = false;
    disp('procedimiento excede el numero maximo de iteraciones');
    break;
  end
  if RMSE <= TOL
    OK = true;
    disp('procedimiento terminado con exito');
    break;
  end
end
```

Index

A
Antecedents, 6, 22, 25

B
Backpropagation algorithm, 24, 37, 68
Bio-inspired algorithms, 1, 12

C
Chromosome, 11, 12, 30, 31
Consequents, 6, 7, 9, 23–26

D
Defuzzification, 23–26
Dow Jones, 2, 17, 19–22, 28, 73–78, 87

E
Ensemble learning, 9, 21, 35, 68, 73, 79, 87

F
Fuzzification, 22, 25, 26
Fuzzy inference system, 28, 30, 31
Fuzzy integrators, 1, 17, 18, 28–33, 35, 45, 46, 53, 54, 56–58, 64, 66, 83, 85–88
Fuzzy Logic System (FLS), 10

G
Gaussian MFs, 7, 26, 28, 31, 56, 59, 60
Generalized bell MFs, 7, 28, 87
Genetic Algorithm (GA), 2, 9–12, 17, 30, 31, 45, 87

I
igaussmtype2, 26, 28, 31, 32, 35, 45, 46, 51, 52, 56–58, 63, 64, 68, 73, 79, 87
igbelltype2, 28, 45, 46, 53, 54, 57, 58, 64, 65, 86, 87
IT2FNN, 1, 2, 6, 17, 18, 21–24, 26, 27, 35, 36, 68, 72–74, 79, 80, 87, 88
itritype2, 28, 45, 46, 54, 55, 57, 66, 67, 86

K
KM algorithm, 23–25

M
Mackey-Glass, 17, 19, 35, 36, 40, 45, 50, 56, 59, 60, 63, 83
Mamdani, 28
Membership functions, 6, 7, 9, 10, 28, 87
Mexican stock exchange (BMV), 2, 17, 19, 20, 68, 87

N
NASDAQ, 2, 17, 19, 21, 22, 28, 79–84, 87

P
Particles, 12, 13, 32, 33
Particle swarm optimization, 2, 12, 17, 54, 87

T
Takagi-Sugeno-Kang (TSK), 6, 22, 25, 26
Time series, 5, 17, 18, 54, 56, 60, 66, 70, 73, 74
Triangular MFs, 28, 45, 50, 51, 56–58, 62, 63, 86, 87

J. Soto et al., *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, SpringerBriefs in Computational Intelligence, https://doi.org/10.1007/978-3-319-71264-2