APPENDIX A

MP – ANN : This software can be used for implementation of M-P neural network model discussed in Chapter II

```
                                /* MP-ANN */

                        /*Program using rule 2.2.2    */

#include <stdio.h>);
#include <stdlib.h>;
#include <math.h>;
#include <conio.h>;
#include <time.h>;

 void main()
  {
       int     i,j,k,p,I,P;  float z1[10];
       int     o[4],d[4],z[10][10];
       float   w[10],out1,t,e; int qp1;

                              FILE *fp1,*fp2,*fp3;

                 clrscr();
            printf( "\t This softwre needs two Data files:\n ");
            printf("\n\ti) INPUT.dat : Input data file containing P set of");
            printf("                    Inputs vectors each of dim I. \n");
            printf("\n\tii) D_OUT.dat : File containing K dim vector of ");
            printf(" Target values.\n");


            fp1= fopen("INPUT.DAT","r");
            fp2= fopen("D_OUT.DAT","r");
            fp3= fopen("SMR.DAT","w");

              if(fp1==NULL)
              {
                        printf("\n Error open file-1");
                          exit(0);
                 }

              if(fp2==NULL)
                 {
                        printf("\nError open file-2");
                          exit(0);
                  }
```

```
            /* enter the data....*/

   printf( "\n\n Enter values of : ");
   printf("\n I= No. of Input neurons ");
   printf(" P= No of Patterns\n\t");
       scanf("%d",&I);
       printf("\t\t\n");
       scanf("%d",&P);
      printf("\n The val of I,P are =%d %d",I,P);


   printf("\n Reading the values of z[][] and d[]\n");

   for (p=1;p<=P;p++)

      {
            for(i=1;i<=I-1;i++)
            fscanf(fp1,"%d",&z[p][i]);
                  z[p][I]=-1;
        }

            for (p=1;p<=P;p++)
             fscanf(fp2,"%d",&d[p]);
                  clrscr();


      randomize();

      for (i=1;i<=I;i++)
      {
             w[i]=2.0*((float)rand()/RAND_MAX)-1.0;
             printf("\n w[%d]=%f",i,w[i]);
       }

       /*Iteratien starts */

qp1=1;
 while (e != 0 )

       {
         /* while (p<=P)
         {
               for(i=1;i<=I;i++)
              {
                    z1[i]=z[p][i];
                }*/


              /* compute net*/
                   e=0.0;
```

```
                for (p=1;p<=P;p++)
                {
                              out1=0;
                     for (i=1;i<=I;i++)
                     out1=out1+w[i]*z[p][i];
                          if (out1 > 0)
                               o[p]=1;
                          else
                               o[p]=-1;
                if( d[p] > o[p])
                     for(i=1;i<=I;i++)
                     {
                          w[i]=w[i]+0.5*(d[p]-o[p])*z[p][i];
                          printf("\n w[i]=%f",w[i]);
                     }



                else
                    if( d[p] < o[p])
                      for(i=1;i<=I;i++)
                          {
                              w[i]=w[i]-0.5*(d[p]-o[p])*z[p][i];
                              printf("\n w[%d]=%f",i,w[i]);
                          }

           }
        for (p=1;p<=P;p++)
           e=e+0.5*(d[p]-o[p])*(d[p]-o[p]);
              printf("\n e=%f",e);
              qp1=qp1+1;

    }   /* End of while loop */

     for(p=1;p<=P;p++)
          printf("\n d[%d]=%d  o[%d]=%d",p,d[p],p,o[p]);
          printf("\n");
          printf("\n No. of iterations=%d\n",qp1);

      printf("\n final wts.");
          for (i=1;i<=I;i++)
               printf("\n w[%d]=%f\t%f",i,w[i],e);

}  /* Program end */
```

APPEXDIX

SANN : This software can be used for training single layer ANN with continuous activation function.

```
/* SANN.C */

/* Delta learning rule - single layer ANN  */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <conio.h>
#include <time.h>

void main()
  {

      float z[100][12],d[100][12],v[12][12],z1[100],d1[100];
      float e; int p,i,j,k,I,J,K,P,q; float m[100]; double y[100];
      float ed[100],ds[100];float ds1,s,ita,emax; int r1,ip,qmax,t1,t2;
      int t,T;float y1[300],pre; char c;

            FILE *fp1,*fp2,*fp3,*fp5;


            fp1= fopen("INPUT.DAT","r");
            fp2= fopen("D_OUT.DAT","r");
            fp3= fopen("TEST.DAT","r");
            fp5= fopen("SM.DAT","w");

                  if(fp1==NULL)
                    {
                         printf("\n Error open file-1");
                         exit(0);
                        }

                   if(fp2==NULL)
                    {
                         printf("\nError open file-2");
                         exit(0);
                        }

                   if(fp3==NULL)
                    {
                         printf("\Error open file-3");
                         exit(0);
                     }
```