APPENDIX 1

A.1 NEURAL NETWORK DESIGN AND TRAINING

The network architecture or features such as number of neurons and layers are very important factors that determine the functionality and generalization capability of the network. For this model standard multilayer feed forward–back propagation neural networks are designed with neural fitting tool software. The network consists of three layers, they are the input layer, hidden layer and output layer. In order to determine the optimal architecture, different network with different number of layers and neuron in the hidden layer are designed and tested. The ANN models developed for the prediction of TDS have 12 inputs and 1 output. The number of neurons is adjusted in steps containing the multiples of inputs, as a trial. The various stages of training of premonsoon model are summarized in Table A1.1

Table A1.1 Training stages of ANN model for premonsoon season

<table>
<thead>
<tr>
<th>S. No</th>
<th>Number of hidden layers</th>
<th>MSE</th>
<th>R value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>1284720</td>
<td>4.21</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>206692</td>
<td>8.12</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>59011</td>
<td>5.41</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>519.8</td>
<td>0.9878</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>3.38 e^{-24}</td>
<td>0.9999</td>
</tr>
<tr>
<td>6</td>
<td>66</td>
<td>179.2</td>
<td>0.9984</td>
</tr>
</tbody>
</table>
The first stage of training with 12 neurons has given a very high values of MSE and R. Each stage of training is carried out by 10 trials. The minimum value of MSE and R in all 10 trials is considered and tabulated. The training is further continued by adjusting the neurons as given in Table A1.1 until a very low value of MSE and R is acquired. The value of MSE is started reducing when numbers of neurons are increased. A very low value of MSE is when the number of neurons is 60. The value of MSE started increasing on further increase in number of neurons. The type of result is seen in R value also. Hence the apt number of neuron is considered as “60” for the premonsoon model.

The procedure followed in training ANN model is explained with the model created for premonsoon season. The same procedure is followed for postmonsoon season also. The postmonsoon model has given a low value of MSE and R value of 0.999 in the fourth stage of training. The number of neurons selected is 48 for best postmonsoon model. The various stages of training of postmonsoon model are summarized in Table A1.2.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Number of hidden layers</th>
<th>MSE</th>
<th>R value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>425544</td>
<td>7.2</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>193413</td>
<td>0.89</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>8666</td>
<td>0.995</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>8.15 e^{-24}</td>
<td>0.9999</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>9.7 e^{-1}</td>
<td>0.9888</td>
</tr>
</tbody>
</table>
APPENDIX 2

A.1 DESIGN PROGRAM “ARD PRO 1.0”

```c
#include<stdio.h>
#include<math.h>
main()
{
float  Aq,Nrf,Sy,Ir, K, gwt, GWT,
Dw,Diaw,rw,TDS,PerTDS,Qdis,Qnet,Nw,FQf;
float  PWL,ho,hw,Qnet1,Qnet2,Qr,Qrf,Qf,Rd;
float  Qgr,Qwelly,Qsup,Qdisr,Q1,Qrw, diarw, Qy,hrw,hro,dw;
clrscr();
printf("n Enter TDS content of the water sample from the selected well: ");
scanf("%f", & TDS);
printf("n Enter Allowable TDS content in drinking water: ");
scanf("%f", & PerTDS);
printf("n Enter Overall depth of the existing well in m: ");
scanf("%f", & Dw);
printf("n Enter Diameter of the well / Bore well in m: ");
scanf("%f", & Diaw);
printf("n Enter Hydraulic conductivity of the aquifer: ");
scanf("%f", & K);
printf("n Enter Depth of groundwater table: ");
scanf("%f", & GWT);
printf("n Enter Normal fluctuation of groundwater table before and after 
rains: ");
scanf("%f", & gwt);
```
printf("\n Enter Depth of pumping water level:");
scanf("%f", &PWL);
printf("\n Enter Radius of cone of depression (For alluvial areas: 200-300 m &
For Hard rock areas: 75-150 m)");
scanf("%f", &Rd);
printf("\n Enter Normal rainfall of the locality in mm:");
scanf("%f", &Nrf);
printf("\n Enter Specific yield of the aquifer:");
scanf("%f", &Sy);
printf("\n Enter Infiltration rate of the substratum:");
scanf("%f", &Ir);
printf("\n Enter Depth of proposed recharge well in m : ");
scanf("%f", &dw);
printf("\n Enter Diameter of proposed recharge well :");
scanf("%f", &diarw);

hw = (Dw - PWL);
ho = (Dw - GWT);

Q1 = ((3.14 * K * (ho * ho) - (hw * hw)) / (log (Rd / (Diaw / 2))));
Qwelly = Q1 * 365;
Qf = (TDS / PerTDS);
FQf = Qf * 1.5;
Qdis = (Qwelly * FQf);
Aq = (3.14 * Rd * Rd);
Qy = (Aq * gwt * (Sy / 100));
Qrf = (Aq * (Ir / 100) * (Nrf / 1000));
if (Qrf < Qy)
    printf("Rainfall is insufficient ");
else
    Qsup = (Qy - Qrf);
Qsup=0;
if (Qrf<Qy)
printf("Additional Quantity of water required for recharge \%.2f cum \n",Qsup);
if (Qrf<Qy)
printf("Collect Additional Quantity of water required for recharge from Roof Top, Refer Table 9.1 for \%.2f cum \n",Qsup);
if (Qrf>Qy)
Qdisr=(Qrf-Qy);
else
Qdisr=0;
Qnet1=(Qdis+Qsup);
Qnet2=(Qdis-Qdisr);
if ( Qnet1 > Qnet2)
Qnet=Qnet1;
else
Qnet=Qnet2;
hrw=(dw);
hro=(dw-GWT);
Qrw=((3.14*K*((hrw*hrw)-(hro*hro)))/(log(Rd/(diarw/2))))*365;
Nw=(Qnet/Qrw);
printf("Number of wells required in upstream side of flow direction \%.2f Nos \n",Nw);
printf("Diameter of the recharge well \%.2f m \n",diarw);
printf("Depth of the recharge well \%.2f m \n",dw);
getch();
return;