```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace Neural_Network
12 {
13     public partial class Visualizer : Form
14     {
15         public Visualizer()
16         {
17             InitializeComponent();
18         }
19
20         Trainer trainer = new Trainer();
21
22         private void runToolStripMenuItem_Click(object sender, EventArgs e)
23         {
24             for (int i = 0; i < trainer.training.Count-1; ++i) {
25                 g1.DrawLine(p1,
26                     Convert.ToSingle(trainer.training[i].inputVector[0]),
27                     Convert.ToSingle(trainer.training[i].expectedOutput),
28                     Convert.ToSingle(trainer.training[i + 1].inputVector[0]),
29                     Convert.ToSingle(trainer.training[i + 1].expectedOutput)
30                     );
31                 //g1.DrawRectangle(p,Convert.ToSingle(trainer.training[i].inputVector[0]), Convert.
       ToSingle(trainer.training[i].expectedOutput), 0.001f, 0.001f);
32             }
33         }
34
35         Graphics g1;
36         int w, h;
37         Pen p1, p2;
38         protected override void OnLoad(EventArgs e)
39         {
40             base.OnLoad(e);
41
42             g1 = pictureBox1.CreateGraphics();
43
44             w = pictureBox1.Width;
45             h = pictureBox1.Height;
46
47
48             g1.TranslateTransform(pictureBox1.Width / 2, pictureBox1.Height / 2);
49             g1.ScaleTransform(pictureBox1.Width / 20.0F, -pictureBox1.Height / 6.0F);
50
51
52             p1 = new Pen(Color.Green, 0.05F); // target function
53             p2 = new Pen(Color.Black, 0.05F); // coord axis
54
55         }
56
57         int outputCounter = 0;
58
59         private void showNetworkOutputToolStripMenuItem_Click(object sender, EventArgs e)
60         {
61             List<List<double>> tr = trainer.trainingResults();
62
63             //vary output color
64             Pen p = new Pen(Color.FromArgb((255*5-3*outputCounter)%255,(outputCounter)%255,(10*
       outputCounter++)%255), 0.001F);
65
66             //axis
67             g1.DrawLine(p2, -10f, 0f, 10f, 0f);
68             g1.DrawLine(p2, 0f, -3f, 0f, 3f);
69
70             for (int i = 0; i < tr.Count-1; ++i)
71             {
72               g1.DrawLine(p,
```

```
73                      Convert.ToSingle(trainer.training[i].inputVector[0]),
74                      Convert.ToSingle(tr[i][0]),
75                      Convert.ToSingle(trainer.training[i + 1].inputVector[0]),
76                      Convert.ToSingle(tr[i+1][0])
77                      );
78              }
79
80              toolStripStatusLabel1.Text = trainer.meanSquareError().ToString();
81          }
82
83          private void trainOutputLayerToolStripMenuItem_Click(object sender, EventArgs e)
84          {
85              trainer.trainOutputLayer();
86              showNetworkOutputToolStripMenuItem_Click(sender, e);
87          }
88
89          private void trainHiddenLayer0ToolStripMenuItem_Click(object sender, EventArgs e) {
90              trainer.trainHiddenLayer();
91              showNetworkOutputToolStripMenuItem_Click(sender, e);
92          }
93      }
94  }
95
```