**รายงาน**

**เรื่อง นิวรอลเน้ตเวิร์คแบบหลายเลเยอร์**

จัดทำโดย

นายวัสพร คนงาม รหัสนักศึกษา 541733023026-0

นายเอนก กันสันเทียะ รหัสนักศึกษา 541733023010-4

นายธนวัฒน์ ปลั่งกลาง รหัสนักศึกษา 541733023013-8

นายพัฐพงษ์ ปานประชาติร หัสนักศึกษา 541733023010-4

เสนอ

อาจารย์ธรรมกร ครองไตรภพ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา **ปัญญาประดิษฐ์(Artificial Intelligence)**

สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน

ภาคเรียนฤดูร้อน ปีการศึกษาที่ 2556

**คำนำ**

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา **ปัญญาประดิษฐ์(Artificial Intelligence)** ทางคณะผู้จัดทำได้ค้นคว้าหาข้อมูลต่างๆจากหนังสือของห้องสมุด และ ค้นคว้าหาจากอินเตอร์เน็ต เกี่ยวกับ นิวรอลเน็ตเวิร์คแบบหลายเลเยอร์นำมาประยุกต์ใช้โดยมีเนื้อหา ซึ่งจะเป็นประโยชน์กับผู้สนใจเกี่ยวกับนิวรอลเน็ตเวิร์คแบบหลายเลเยอร์ หากผิดพลาดประการใดทางคณะผู้จัดทำขออภัยไว้ ณ ที่นี้ด้วย

**นิวรอลเน้ตเวิร์คแบบหลายเลเยอร์**

นิวรอลเน็ตเวิร์คแบบหลายเลเยอร์ เป็นนิวรอลเน็ตเวิร์คที่ประกอบด้วยเลเยอร์ตั้งแต่ 2 เลเยอร์ขึ้นไป เอาท์พุทแพทเทิร์นในเลเยอร์หนึ่งจะเป็นค่าอินพุทของเลเยอร์ถัดไป

เลเยอร์ที่อยู่ระหว่างอินพุทและเอาท์พุทถูกเรียกว่า ฮิดเดนเลเยอร์หรือเลเยอร์ภายใน (Internal layer)

ค่าเอาท์พุทในเลเยอร์สุดท้ายจะถูกเปรียบเทียบกับค่าเอาท์พุทจริงที่ต้องการ (Desired output) จากผู้สอน

นิวรอลเน็ตเวิร์คแบบหลายเลเยอร์จะถูกนำไปใช้กับอัลกอริทึมของแบ็คพร็อพพาเกชัน (Back Propagation)

**แบ็คพร็อพพาเกชัน (Back Propagation)** เป็นอัลกอริทึมสำหรับการสอนที่นำไปใช้กับนิวรอลเน็ตเวิร์คแบบหลายเลเยอร์

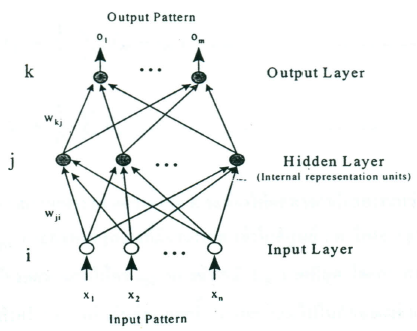
โดยอาศัยประสบการณ์ที่ได้จากความรู้ของการแปลงสัญญาณระหว่างอินพุทและเอาท์พุทภายในเน็ตเวิรค์แบบหลายเลเยอร์ด้วยรูปแบบการเรียนรู้แบบมีผู้สอน

**แบ็คพร็อพพาเกชันแบ่งขั้นตอนการดำเนินการเป็นสองส่วนหลัก**

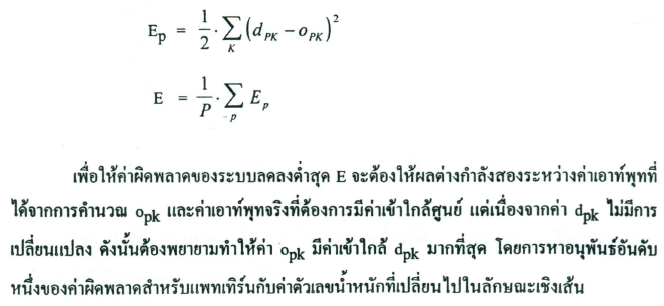
1.ส่วนการป้อนไปหน้า

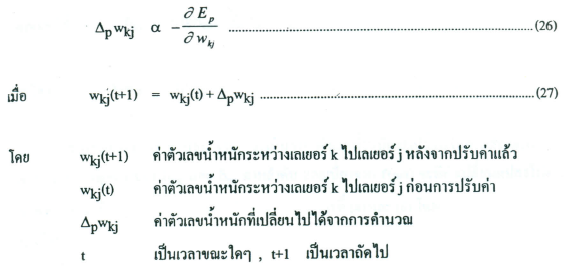
2.ส่วนการป้อนกลับ

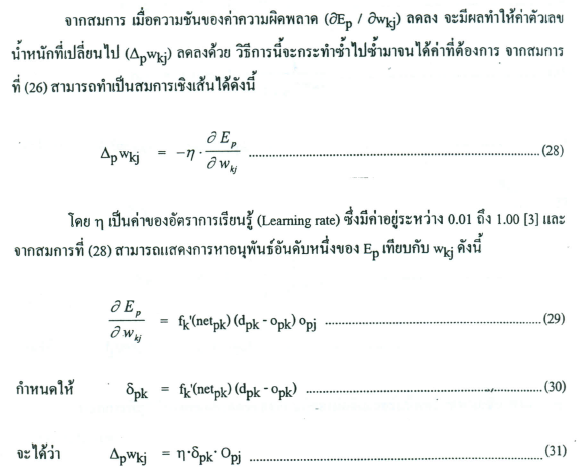
เป็นนิวรอลเน็ตเวิร์คแบบหลายเลเยอร์ ที่มีเลเยอร์ภายในเพียงเลเยอร์เดียว โดยวงกลมในรูปแสดงถึงโหนดหรือนิวรอล และลูกศรจะแสดงทิศทางการเชื่อมต่อของการเคลื่อนที่ของสัญญาณระหว่างนิวรอลในแต่ละเลเยอร์ ซึ่งจะถูกแทนด้วยค่าตัวเลขน้ำหนักระหว่างนิวรอล

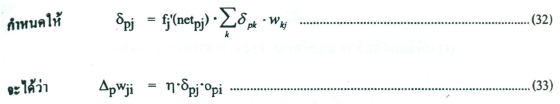
\

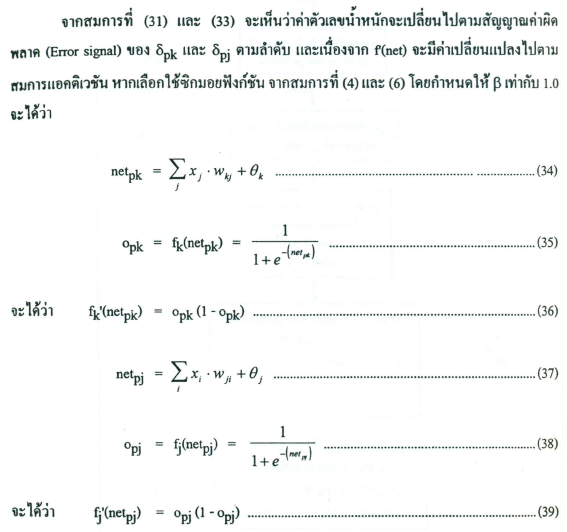
ค่าผิดพลาดของสำหรับแพทเทิร์น (Ep) และค่าผิดพลาดของระบบโดยรวม (E) สามารถหาได้โดยสมการ

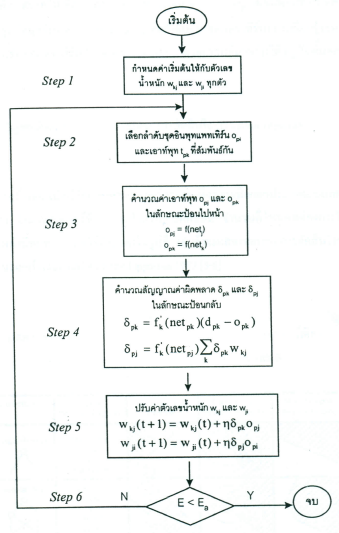




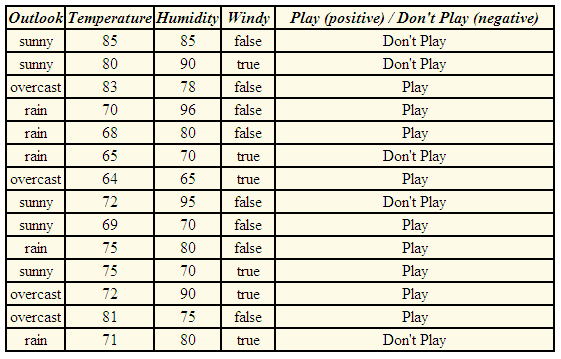


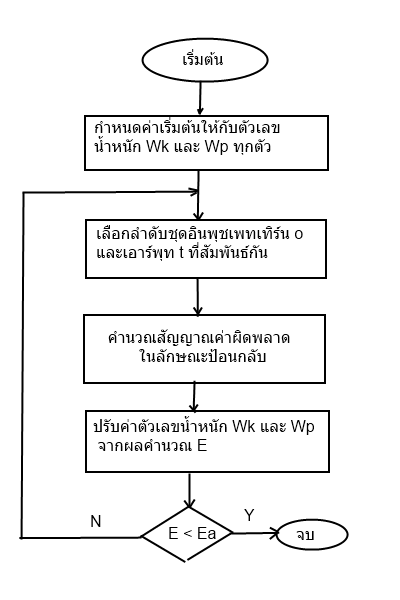






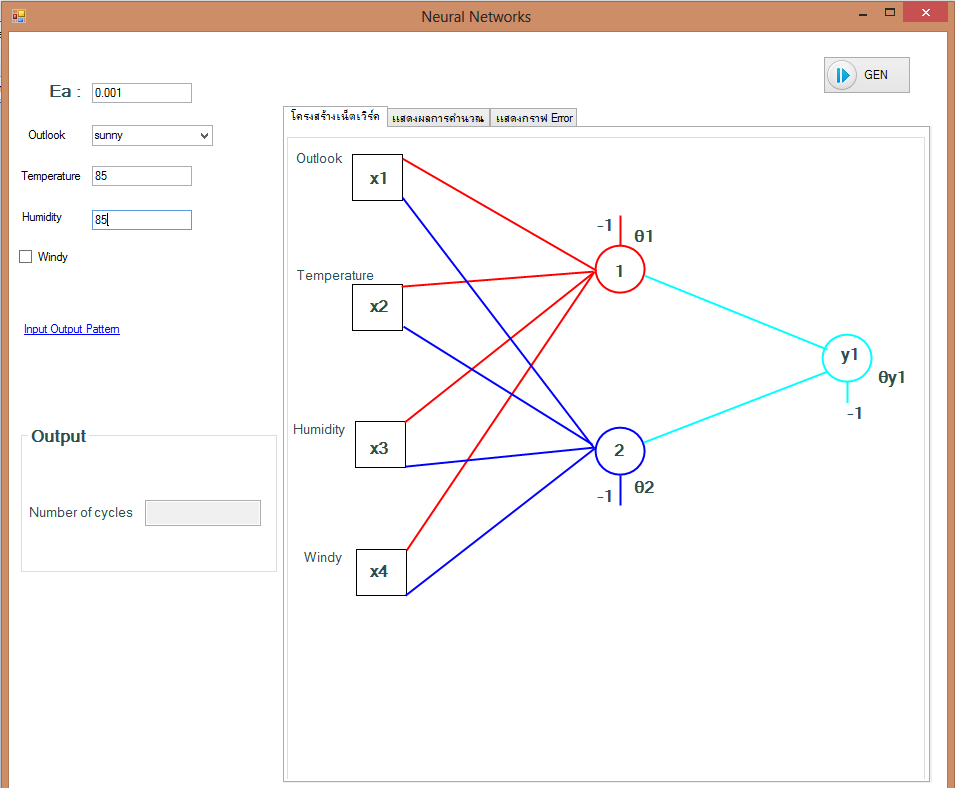
เขียนโปรแกรมนิวรอลเน็ตเวิร์คแบบหลายเลเยอร์ที่ใช้ในการตัดสินใจปัญหาการเล่นกอลฟ์?



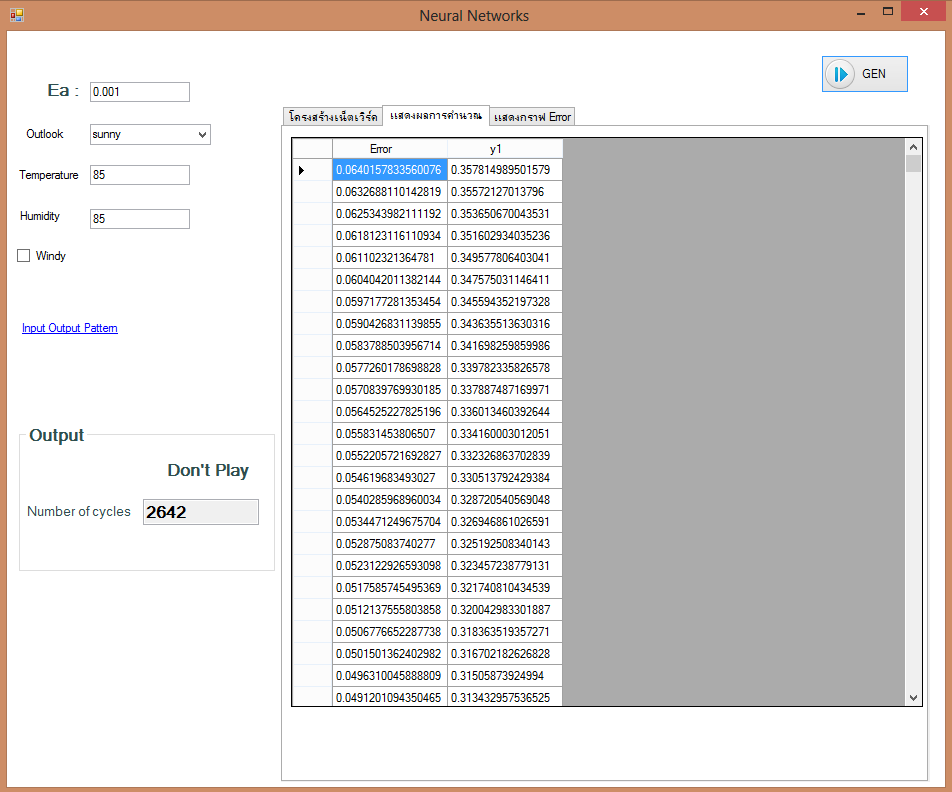


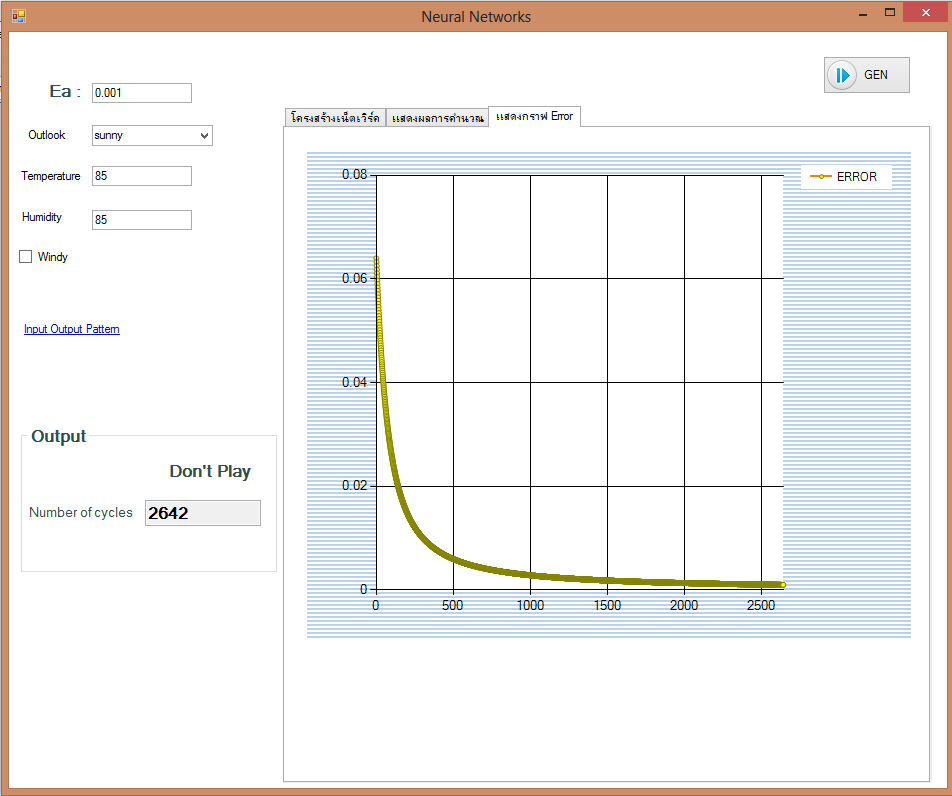
ภาพแสดงโครงสร้าง นิวรอลเน้ตเวิร์คแบบหลายเลเยอร์การเล่นกอล์ฟ

หน้าแสดงการป้อนค่า



แสดงการคำนวณนิวรอลเน้ตเวิร์คแบบหลายเลเยอร์



กราฟแสดงการคำนวณ ค่า Error

**ส่วนโค้ดโปรแกรม**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Collections;

namespace AI\_Back\_Propagation

{

public partial class Form1 : Form

{

int ti = -1;

public Form1()

{

InitializeComponent();

}

public void display7seg(PictureBox p, NumericUpDown num)

{

if (num.Value == 1)

{

p.BackColor = Color.Red;

}

else {

p.BackColor = Color.White;

}

}

private void linkLabel2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

new Input\_Output\_Pattern().Show();

}

private void button1\_Click(object sender, EventArgs e)

{

string outp=chkPattern();

if (outp == "") MessageBox.Show("Pattren ไม่ถูกต้อง");

else {

calculate(outp);

Char[] sc = outp.ToCharArray();

String d = outp.ToString();

if (d == "1")

{

d = "Play";

}

else

{

d = "Don't Play";

}

label1.Text = d.ToString();

}

}

public string chkPattern()

{

int WindyNum;

int outNum = int.Parse(Outlook.SelectedIndex.ToString());

Double TempertNum = int.Parse(Temperrature.Text.ToString());

Double HumiNum = int.Parse(Humidity.Text.ToString());

Double TemIn = TempertNum / 100;

Double HumiIn = HumiNum / 100;

if (Windy.Checked == false)

WindyNum = 0;

else

WindyNum = 1;

string Input = outNum.ToString() + TemIn.ToString() + HumiIn.ToString() + WindyNum.ToString();

ArrayList In\_pattern=new ArrayList(){"00.850.850","00.80.91","10.830.780","20.70.960","20.680.80","20.650.71","10.640.651","00.720.950","00.690.70","20.750.80","00.750.71","10.720.91","10.810.750","20.710.81"};

string[] Out\_pattern = { "0", "0", "1", "1", "1","0","1","0","1","1","1","1","1","0" };

int ou = In\_pattern.IndexOf(Input);

if (ou != -1) return Out\_pattern[ou];

else return "";

}

private void calculate(string ou)

{

int i = 0;

Char[] sc = ou.ToCharArray();

int WindyNum;

int outNum = int.Parse(Outlook.SelectedIndex.ToString());

Double TempertNum = int.Parse(Temperrature.Text.ToString());

Double HumiNum = int.Parse(Humidity.Text.ToString());

Double TemIn = TempertNum / 100;

Double HumiIn = HumiNum / 100;

if(Windy.Checked==false)

WindyNum = 0;

else

WindyNum = 1;

double x1 = Convert.ToDouble(0), x2 = Convert.ToDouble(TemIn), x3 = Convert.ToDouble(HumiIn), x4 = Convert.ToDouble(WindyNum);

double Y1 = Convert.ToDouble(sc[0] + "");

Random r = new Random();

double w11 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w11 = -1 \* w11;

double w21 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w21 = -1 \* w21;

double w31 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w31 = -1 \* w31;

double w41 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w41 = -1 \* w41;

double w12 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w12 = -1 \* w12;

double w22 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w22 = -1 \* w22;

double w32 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w32 = -1 \* w32;

double w42 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1); if (r.Next(1, 3) == 1) w42 = -1 \* w42;

double W11 = randomWpk(), W21 = randomWpk();

double teta1 = randomWpk(), teta2 = randomWpk(), teta3 = randomWpk();

// teta4 = Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1);

double tetay1 = randomWpk();

double d1, d2;

double y1;

double e1;

double sumE;

Random rr = new Random();

int nloop=rr.Next(5000,6000);

dataGridView1.Rows.Clear();

chart1.Series[0].Points.Clear();

while (true)

{

i++;

d1 = 1 / (1 + Math.Exp(-(x1 \* w11 + x2 \* w21 + x3 \* w31 + x4 \* w41 - 1 \* teta1)));

d2 = 1 / (1 + Math.Exp(-(x1 \* w12 + x2 \* w22 + x3 \* w32 + x4 \* w42 - 1 \* teta2)));

y1 = 1 / (1 + Math.Exp(-(d1 \* W11 + d2 \* W21 - 1 \* tetay1)));

e1 = Y1 - y1;

sumE = Math.Pow(e1, 2);

sumE = sumE / 2;

chart1.Series[0].Points.AddXY(i, sumE);

dataGridView1.Rows.Add(sumE, y1);

if (sumE < double.Parse(Ep.Text) || (i == nloop))

{

textBox1.Text = i+"";

break;

}

else {

double ro = 0.1;

double deltay1 = y1 \* (1 - y1) \* e1;

double deltaW11 = ro \* d1 \* deltay1; W11 += deltaW11;

double deltaW21 = ro \* d2 \* deltay1; W21 += deltaW21;

double deltatetay1=ro\*(-1)\*deltay1;tetay1+=deltatetay1;

}

}

}

private double randomWpk()

{

Random r = new Random();

if(r.Next(1,3)==1)

return Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1);

else return -1\*Math.Round((double)(r.Next(1, 3) - r.NextDouble()), 1);

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Ep\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == 8) ;

else if (e.KeyChar == 46 && Ep.Text.Contains(".")) e.Handled = true;

else if (e.KeyChar < 46 || e.KeyChar > 57 || e.KeyChar == 47) e.Handled = true;

}

private void chart1\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void numericUpDown9\_ValueChanged(object sender, EventArgs e)

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}