

2023.11.15

Arduino.

Экспертная Система (ЭС) байесовского типа

Распознавание объектов по готовым правилам.

Иллюстрирующий пример.

Версия 1.01.

Оглавление

1 «Об этом».....	2
2 Текст скетча.....	4

1 «Об этом»...

В качестве «подопытного кролика» используется Arduino UNO (без всяких дополнительных устройств, датчиков и т.д. – поэтому, схема не приводится).

Информация об ЭС байесовского типа в подробностях приведена в книге К. Нейлора «Как построить свою экспертную систему».

(https://roamer55.ru/my_download/progs/os_win/spTrain_ES/kak_postrait_svou_es_nejlor.pdf).

В этом документе тема рассматривается лишь как шуточный, иллюстрирующий пример... Поэтому, в скетче (см. ниже) «все зашито жестко». Т.е., ровно **НОЛЬ практического эффекта**.

Но вот если:

1. Вместо **Arduino UNO** использовать (например) **ESP-12F WeMos D1 WiFi...**
С целью: оперативно передавать на WeMos через Wi-Fi задачу (таблицу правил) и признаки распознавания объекта.
2. Каждому признаку будет соответствовать определенный датчик.
3. «Это» встроить в адаптивную и гетерогенную СЕТЬ ЭС...

То очень может быть – будет практическая польза...

:-)

В рамках данного скетча (см. ниже) принято.

Распознаваемые объекты:

- 0 - Самолет;
- 1 - Планер;
- 2 - Птица.

Признаки распознаваемых объектов:

- 0 - Крылья есть;
- 1 - Клюв есть;
- 2 - Хвост есть;
- 3 - Оперение есть;
- 4 - Двигатель есть;
- 5 - Шасси есть.

Таблица правил (результат обучения ЭС) - получена в рамках программы **spTrainES.exe** (ее можно скачать бесплатно на сайте https://roamer55.ru/my_download/progs/os_win/spTrain_ES/).

Результат распознавания объекта для приведен на рисунке ниже:

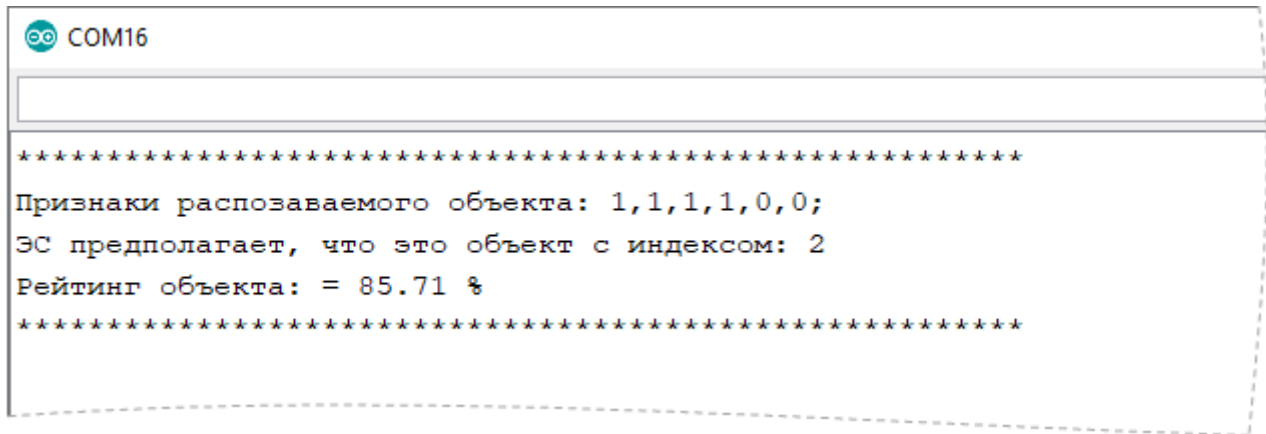


Рисунок 1 – Результат распознавания объекта по заданным признакам
(esPropS = "1,1,1,1,0,0;")

Признаки (см. текущее значение переменной esPropS в скетче):

Крылья есть – да;
Клюв есть – да;
Хвост есть – да;
Оперение есть – да;
Двигатель есть – нет;
Шасси есть – нет.

2 Текст скетча

Текст скетча для Arduino UNO приведен ниже.
Все пояснения внесены в текст скетча.

```

/*
Чем бы дите не тешилось, лишь бы... :-)
-----

2023.11.15
Arduino. Экспертная Система (ЭС) байесовского типа.
Распознавание объектов по готовым правилам.
Иллюстрирующий пример.
Версия 1.01.

Информация об ЭС байесовского типа в подробностях приведена в книге
К. Нейлора "Как построить свою экспертную систему".

В данном скетче "все зашито жестко" потому, что это - всего лишь иллюстрирующий пример
и не более.

-----
Но вот если:
1. Вместо Arduino UNO использовать (например) ESP-12F WeMos D1 WiFi...
   С целью: оперативно передавать на WeMos через Wi-Fi задачу (таблицу правил)
   и признаки распознавания объекта.
2. Каждому признаку будет соответствовать определенный датчик.
3. "Это" встроить в адаптивную и гетерогенную СЕТЬ ЭС...
То очень может быть - будет практическая польза...
:-)
-----

В рамках данного скетча принято.
Распознаваемые объекты:
0 - Самолет;
1 - Планер;
2 - Птица.

Признаки распознаваемых объектов:
0 - Крылья есть;
1 - Клов есть;
2 - Хвост есть;
3 - Оперение есть;
4 - Двигатель есть;
5 - Шасси есть.

Таблица правил (результат обучения ЭС) - получена в рамках программы spTrainES.exe
(ее можно скачать бесплатно на сайте roamer55.ru).
*/

//-----
//Исходные данные
String esRuleS = ""; //Таблица правил текущей задачи по распознаванию объектов
String esPropS = ""; //Строка значений признаков распознаваемого объекта
//-----
//-----
//Вычисляемые значения
//Кол-во распознаваемых объектов:
int ObjCount = 0;

```

```

//Кол-во значений признаков распознаваемых объектов:
int PropCount = 0;
//Индекс распознанного объекта (объект с наилучшим рейтингом):
int Obj_Best_Index = -1;
//Рейтинг (%) распознанного объекта:
float Obj_Best_Rating = 0.0;
//Кол-во распознанных объектов с одинаковым (наилучшим) рейтингом:
int Obj_Best_Rating_Count = 0;
//-----

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600);

  //-----
  //Таблица правил текущей задачи распознавания объектов (Самолет, Планер, Птица)
  esRuleS = "";
  esRuleS = esRuleS + "2,4,5;";
  esRuleS = esRuleS + "-1,-1,8;";
  esRuleS = esRuleS + "2,4,5;";
  esRuleS = esRuleS + "-1,-1,8;";
  esRuleS = esRuleS + "8,-3,-1;";
  esRuleS = esRuleS + "3,5,-3;";
  //-----

  //-----
  //Строка значений признаков распознаваемого объекта
  //esPropS = "1,0,1,0,1,1;"; //Самолет
  //esPropS = "1,0,1,0,0,1;"; //Планер
  esPropS = "1,1,1,1,0,0;"; //Птица
  //esPropS = "1,1,1,1,0,1;"; //Птица с шасси :-)
  //esPropS = "0,0,0,0,0,0;"; //???
  //-----

  Serial.println("*****");
  Serial.print("Признаки распознаваемого объекта: ");
  Serial.println(esPropS);
  if (ES_Obj_Recognize() == true)
  {
    if (Obj_Best_Rating_Count == 1)
    {
      Serial.print("ЭС предполагает, что это объект с индексом: ");
      Serial.println(Obj_Best_Index);
      Serial.print("Рейтинг объекта: = ");
      Serial.print(Obj_Best_Rating);
      Serial.println(" %");
    }
    else
    {
      Serial.println("ЭС запуталась! ");
      Serial.print("Потому что объектов с наивысшим рейтингом = ");
      Serial.println(Obj_Best_Rating_Count);
    }
  }
  else
  {
    Serial.print("ЭС не смогла распознать объект!");
  }
  Serial.println("*****");
}

void loop()

```

```

{
// put your main code here, to run repeatedly:

}

//*****
//Вспомогательные функции (для создания динамических массивов)
int ES_Prop_Count() { return PropCount; }
int ES_Obj_Count() { return ObjCount; }
//*****

//*****
//Вычисление значений PropCount и ObjCount
//на основе таблицы правил esRuleS
bool ES_Prop_Obj_Count()
{
bool res = false;
PropCount = 0; //кол-во признаков
ObjCount = 0; //кол-во объектов
String s1 = "";
String s = esRuleS; //таблица правил

//-----
//Цикл по строкам (признаки) таблицы правил
//Определение кол-ва признаков
int m = 0;
int n = 1;
while (n>0)
{
n = s.indexOf(";");
if (n>0)
{
PropCount = PropCount + 1; //кол-во признаков
s1 = s.substring(0, n);
s = s.substring(n+1, s.length());
if (PropCount == 1) //Если 1-я строка таблицы правил, то вычисляется кол-во объектов
{
//-----
//Цикл по столбцам (объекты) таблицы правил
//Определение кол-ва объектов
s1.trim();
m = 1;
while (m>0)
{
m = s1.indexOf(",");
if (m <= 0) { m = s1.indexOf(";"); }
if (m <= 0)
{
if (s1.length()>0) { m = s1.length(); }
}
}
if (m > 0)
{
ObjCount = ObjCount + 1; //кол-во объектов
s1 = s1.substring(m+1, s1.length());
}
}
}
//-----
}
}
}
//-----
//Оценка корректности таблицы правил esRuleS

```

```

if (PropCount > 1)
{
    if (PropCount >= ObjCount)
    {
        res = true;
    }
}
//-----
return res;
}
//*****

//*****
//Распознавание объекта по заданным признакам esPropS
//в рамках таблицы правил esRuleS текущей задачи
bool ES_Obj_Recognize()
{
    bool res = false;
    Obj_Best_Index = -1;
    Obj_Best_Rating = 0.0;
    Obj_Best_Rating_Count = 0;
    if (ES_Prop_Obj_Count() == true)
    {
        //Массив рейтингов распознаваемых объектов (целочисленные значения):
        int esObjRatN[ES_Obj_Count()];
        //Массив значений признаков распознаваемого объекта:
        int esPropN[ES_Prop_Count()];
        //Массив. Таблица правил текущей задачи по распознаванию объектов:
        int esRuleN[ES_Prop_Count()][ES_Obj_Count()];

        //=====
        //Парсинг строки значений признаков распознаваемого объекта.
        //Формирование массива esPropN.
        String s1 = "";
        String s2 = "";
        int i = -1;
        int j = -1;
        int m = 0;
        int n = 1;
        String s = esPropS;
        //Serial.println("Признаки:");
        //Serial.println(s);
        while (n>0)
        {
            n = s.indexOf(",");
            if (n <= 0) { n = s.indexOf(";"); }
            if (n <= 0)
            {
                if (s.length()>0) { n = s.length(); }
            }
            if (n>0)
            {
                i = i + 1;
                s1 = s.substring(0, n);
                s = s.substring(n+1, s.length());
                s1.trim();
                esPropN[i] = s1.toInt();
                //Serial.print(esPropN[i]);
                //Serial.print(" ");
            }
        }
        //Serial.println("");
    }
}

```

```

//=====

//=====
//Парсинг строки правил задачи распознавания объектов.
//Формирование массива esRuleN.
//Serial.println(".....");
//Serial.println("Таблица правил:");
s1 = "";
s2 = "";
s = esRuleS;
i = -1;
j = -1;
m = 0;
n = 1;
while (n>0)
{
  n = s.indexOf(";");
  if (n>0)
  {
    i = i + 1;
    s1 = s.substring(0, n);
    s = s.substring(n+1, s.length());
    s1.trim();
    //Serial.println(s1);
    j = -1;
    m = 1;
    while (m>0)
    {
      m = s1.indexOf(";");
      if (m <= 0) { m = s1.indexOf(";"); }
      if (m <= 0)
      {
        if (s1.length()>0) { m = s1.length(); }
      }
      if (m > 0)
      {
        j = j + 1;
        s2 = s1.substring(0, m);
        s1 = s1.substring(m+1, s1.length());
        s2.trim();
        esRuleN[i][j] = s2.toInt();
        //Serial.print("[ "+s2+" ");
        //Serial.print(esRuleN[i][j]);
        //Serial.print(", ");
      }
    }
    //Serial.println("");
  }
}
//Serial.println(".....");
//=====

//=====
//Обнуление массива целочисленных значений рейтингов распознаваемых объектов
for (j = 0; j < ObjCount; j = j + 1)
{
  esObjRatN[j] = 0;
}
//=====

//=====

```



```

//Расчет целочисленных значений рейтингов распознаваемых объектов esObjRatN
//по таблице правил esRuleN и признакам esPropN
int v;
int d;
int r;
for (i = 0; i < PropCount; i = i + 1)
{
v = esPropN[i];
for (j = 0; j < ObjCount; j = j + 1)
{
d = esObjRatN[j];
r = esRuleN[i][j];
d = d + v*r;
esObjRatN[j] = d;
}
}
//=====

//=====
//Поиск объекта с наилучшим целочисленным рейтингом
int RatBest = -999999;
int ObjBest = -999999;
for (j = 0; j < ObjCount; j = j + 1)
{
if (esObjRatN[j] > RatBest)
{
RatBest = esObjRatN[j];
ObjBest = j;
}
}
if (ObjBest >= 0) //объект найден и значит - распознан
{
res = true;
Obj_Best_Index = ObjBest;
Obj_Best_Rating = RatBest;
}
//=====
//=====
//Перерасчет целочисленных рейтингов объектов - в проценты (тип float)
if (ObjBest >= 0)
{
//-----
//Поиск минимального значения рейтинга
float vMin = 999999999;
float vNorm = 0;
float vSum = 0;
for (j = 0; j < ObjCount; j = j + 1)
{
if (vMin > esObjRatN[j]) { vMin = esObjRatN[j]; }
}
//-----

//Массив рейтингов объектов в процентах:
float esObjRatF[ES_Obj_Count()];

//-----
//Нормирование целочисленных значений рейтингов объектов
for (j = 0; j < ObjCount; j = j + 1)
{
v = esObjRatN[j];
vNorm = v - vMin;
vSum = vSum + vNorm;
}
}

```

```

    esObjRatF[j] = vNorm;
}
//-----
//-----
//Вычисление рейтингов объектов в процентах
float vF = 0.0;
for (j = 0; j < ObjCount; j = j + 1)
{
    vF = esObjRatF[j];
    if (vSum != 0)
    {
        vF = (vF/vSum)*100;
        esObjRatF[j] = vF;
    }
}
//-----

//Рейтинг (в процентах) распознанного объекта:
Obj_Best_Rating = esObjRatF[ObjBest];

//-----
//Оценка достоверности результата распознавания
//Если (Obj_Best_Rating_Count == 1), то результата однозначен
//В противном случае - сомнительно
for (j = 0; j < ObjCount; j = j + 1)
{
    if (esObjRatF[j] == Obj_Best_Rating) { Obj_Best_Rating_Count = Obj_Best_Rating_Count + 1; }
}
//-----
}
//=====
}
return res;
}
//*****

```