REPO1(階層型ニューラルネット)

氏名; 津波古正輝
学籍番号;075739
提出日;(7/18 日曜日)

課題内容

1: サンプルソース 1 にコメントをつけよ.
2: サンプルソース 1 を動かして ExOR の学習がうまくいくかどうかを確認せよ.
3: サンプルソース 1 で，中間層のユニット数を変更した場合にどのように挙動が変化するかを調べよ.(誤差の収束をグラフ化し，隠れ層のユニット数についての設計方針について考察せよ)
4: サンプルソース 2 は“0,1,2,3”の数字文字認識のための NN である．学習用の数字パターンファイルを作成し，認識力の高い NN を学習により完成させよ．評価データを独自に作成し未知のデータに対する認識率ができるかどうかを確認せよ.

1:サンプルソース 1 にコメントをつけよ。

主要な変数と定数----------

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

#define LAYER 3//3 段階の階層型ネットワーク
#define INPUT 2//入力信号数
#define HIDDEN 2//中間層の数
#define OUTPUT 1//出力層の数
#define CTG 4//

#define ITERATIONS 30000//繰り返し数

#define ETA 1.20//η
#define EPSILON 1.00//ε
#define ALPHA 0.90//α
#define WD 4.00//重みの幅
#define MIN_ERR 0.000000001

#define ON 0.9
#define OFF 0.1
#define SMAX 15
#define SMIN -15
#define MAX 0.99995
#define MIN 0.00005

#ifdef _Cdecl
#define drand() ((double)(rand()+1)/RAND_MAX-0.5)
#else
#define drand() ((double)(rand()+1)/RAND_MAX-0.5)
#endif

#define sq(x) (x)*(x)

double i_lay[CTG][INPUT+1],//入力層
 h_lay[HIDDEN+1],//中間層
 o_lay[OUTPUT],//出力層
 teach[CTG][OUTPUT];

/*各層の重み*/
double ih_w[INPUT+1][HIDDEN],//INPUT から HIDDEN への重み
 ho_w[HIDDEN+1][OUTPUT];//HIDDEN から OUTPUT への重み

/*各層の誤差*/
double h_del[HIDDEN],//中間層の誤差
 o_del[OUTPUT];//出力層の誤差

/*各層の修正量*/
double ih_lw[INPUT+1][HIDDEN],//INPUT から HIDDEN の修正量
 ho_lw[HIDDEN+1][OUTPUT];//HIDDEN から OUTPUT の修正量

/*慣性項*/
```

```
double ih_dw[INPUT+1][HIDDEN],//INPUT から HIDDEN の慣性項
  ho_dw[HIDDEN+1][OUTPUT];//HIDDEN から OUTPUT の慣性項

  /*収束判定*/
double sigmoid(double s)
{
  double sm;

  sm=EPSILON*s;
  if(sm>SMAX)
    return(MAX);
  else if(sm<SMIN)
    return(MIN);
  else
    return(1./(1.+exp(-sm)));
}

-------------------------------------------------------------------------
/*一括修正法*/
double sigmoid(double s);

main(argc,argv)
int argc;
char **argv;
{
  int i,j,ite,ctg;
  double err,sum;
  srand((int)time((long *)0));

  /* ExOR Problem */
  /*入力パターンの提示
  ilay[x][0] と ilay[x][1] → ilay[x][2](本当の答え teach[x][0]
  0 と 0 → 1(本当の答え:0
  1 と 0 → 1(本当の答え:1
  0 と 1 → 1(本当の答え:1
  1 と 1 → 1(本当の答え:0
  */
  i_lay[0][0]=OFF; i_lay[0][1]=OFF; i_lay[0][2]=ON; teach[0][0]=OFF;
  i_lay[1][0]=ON;  i_lay[1][1]=OFF; i_lay[1][2]=ON; teach[1][0]=ON;
  i_lay[2][0]=OFF; i_lay[2][1]=ON;  i_lay[2][2]=ON; teach[2][0]=ON;
  i_lay[3][0]=ON; i_lay[3][1]=ON;  i_lay[3][2]=ON; teach[3][0]=OFF;

  h_lay[HIDDEN]=ON;

  /*重みをランダムに設定*/
  for(j=0;j<HIDDEN;j++)
    for(i=0;i<=INPUT;i++)
      ih_w[i][j]=WD*drand();//INPUT から HIDDEN への重み
  for(j=0;j<OUTPUT;j++)
    for(i=0;i<=HIDDEN;i++)
      ho_w[i][j]=WD*drand();//HIDDEN から OUTPUT への重み

  /*学習の開始*/
  for(ite=0;ite<=ITERATIONS; ite++){
```

```
/*修正量の初期化*/
for(j=0;j<HIDDEN;j++)
  for(i=0;i<=INPUT;i++)
    ih_lw[i][j]=0.;//INPUT から HIDDEN への修正量
for(j=0;j<OUTPUT;j++)
  for(i=0;i<=HIDDEN;i++)
    ho_lw[i][j]=0.;//HIDDEN から OUTPUT への修正量


/*繰り返し学習*/
for(ctg=0,err=0.;ctg<CTG;ctg++){
  ////////////////////////////
  /*学習*/
  /*INPUT から HIDDEN*/
  for(j=0;j<HIDDEN;j++){
    for(i=0,sum=0.;i<=INPUT;i++)
      sum+=i_lay[ctg][i]*ih_w[i][j];
    h_lay[j]=sigmoid(sum);
  }

  /*HIDDEN から OUTPUT*/
  for(j=0;j<OUTPUT;j++){
    for(i=0,sum=0.;i<=HIDDEN;i++)
      sum+=h_lay[i]*ho_w[i][j];
    o_lay[j]=sigmoid(sum);
  }
  ////////////////////////////



  ////////////////////////////
  /*誤差の計算 (逆方向)*/
  /*誤差の計算 (OUTPUT)*/
  for(i=0;i<OUTPUT;i++){
    err+=sq(teach[ctg][i]-o_lay[i])/2./(double)CTG;
    o_del[i]=EPSILON*(teach[ctg][i]-o_lay[i])*o_lay[i]*(1.-o_lay[i]);
  }

  /*誤差の計算 (HIDDEN)*/
  for(j=0;j<=HIDDEN;j++){
    h_del[j]=0.;
    for(i=0;i<OUTPUT;i++)
      h_del[j]+=o_del[i]*ho_w[j][i];
    h_del[j]*=EPSILON*h_lay[j]*(1.-h_lay[j]);
  }

  ////////////////////////////


  ////////////////////////////
  /*重みの修正*/
  for(j=0;j<=HIDDEN;j++)
    for(i=0;i<OUTPUT;i++)
      ho_lw[j][i]+=ETA*o_del[i]*h_lay[j];//HIDDEN から OUTPUT の重みの修正
```

```
      for(j=0;j<=INPUT;j++)
        for(i=0;i<HIDDEN;i++)
          ih_lw[j][i]+=ETA*h_del[i]*i_lay[ctg][j];//INPUT から HIDDEN の重みの修正
      ////////////////////////////
    }


    /*学習の結果*/
    fprintf(stderr,"iteration = %4d, error = %1.5f\n",ite,err);


    if((err<MIN_ERR)||(ite==ITERATIONS)){
      for(ctg=0;ctg<CTG;ctg++){
        fprintf(stderr,"ctg[%d] : ",ctg);
        for(i=0;i<INPUT;i++)
          fprintf(stderr,"i[%d] = %1.1f  ",i,i_lay[ctg][i]);
          for(j=0;j<HIDDEN;j++){
            for(i=0,sum=0.;i<=INPUT;i++)
              sum+=i_lay[ctg][i]*ih_w[i][j];
            h_lay[j]=sigmoid(sum);
          }
          for(j=0;j<OUTPUT;j++){
          for(i=0,sum=0.;i<=HIDDEN;i++)
            sum+=h_lay[i]*ho_w[i][j];
          o_lay[j]=sigmoid(sum);
          fprintf(stderr,"o[%d] = %1.5f, t{%d] = %1.1f\n",j,o_lay[j],j,teach[ctg][j]);
          }
        }
      fprintf(stderr,"iteration = %4d, error = %1.5f\n",ite,err);
      break;
    }

    /*前回の重みの修正の影響の考慮*/
    for(j=0;j<=HIDDEN;j++)
      for(i=0;i<OUTPUT;i++)
        ho_w[j][i]+=ho_lw[j][i];
    for(j=0;j<=INPUT;j++)
      for(i=0;i<HIDDEN;i++)
        ih_w[j][i]+=ih_lw[j][i];
      }
    }

--------------------------------------------------------------------------
/*逐次修正法*/
main(argc,argv)
int argc;
char **argv;
{
  int i,j,ite,ctg;
  double err,sum;
  srand((int)time((long *)0));

  /* ExOR Problem */
  i_lay[0][0]=OFF; i_lay[0][1]=OFF; i_lay[0][2]=ON; teach[0][0]=OFF;
```

```
  i_lay[1][0]=ON;  i_lay[1][1]=OFF; i_lay[1][2]=ON; teach[1][0]=ON;
  i_lay[2][0]=OFF; i_lay[2][1]=ON;  i_lay[2][2]=ON; teach[2][0]=ON;
  i_lay[3][0]=ON; i_lay[3][1]=ON;  i_lay[3][2]=ON; teach[3][0]=OFF;

  h_lay[HIDDEN]=ON;

  /*重みをランダムに初期化*/
  for(j=0;j<HIDDEN;j++)
    for(i=0;i<=INPUT;i++)
      ih_w[i][j]=WD*drand();
    for(j=0;j<OUTPUT;j++)
      for(i=0;i<=HIDDEN;i++)
        ho_w[i][j]=WD*drand();

  /*学習の開始*/
  for(ite=0;ite<=ITERATIONS; ite++){
    for(ctg=0,err=0.;ctg<CTG;ctg++){
      for(j=0;j<HIDDEN;j++){
        for(i=0,sum=0.;i<=INPUT;i++)
          sum += i_lay[ctg][i]*ih_w[i][j];//sum=入力層の値*入力層から中間層の重み
        h_lay[j]=sigmoid(sum);//中間層の決定
      }
      for(j=0;j<OUTPUT;j++){
        for(i=0,sum=0.;i<=HIDDEN;i++)
          sum+=h_lay[i]*ho_w[i][j];//sum=中間層の値*中間層から出力層の重み
        o_lay[j]=sigmoid(sum);//出力層の決定
      }

      //誤差の計算
      for(i=0;i<OUTPUT;i++){
        err+=sq(teach[ctg][i]-o_lay[i])/2./(double)CTG;//エラーの計算
        o_del[i]=EPSILON*(teach[ctg][i]-o_lay[i])*o_lay[i]*(1.-o_lay[i]);//出力層
の誤差=(ε*教師データ)*出力層*(1-出力層)
      }
      for(j=0;j<=HIDDEN;j++){
        h_del[j]=0.;//中間層の誤差の初期化
        for(i=0;i<OUTPUT;i++)
          h_del[j]+=o_del[i]*ho_w[j][i];//中間層の誤差=出力層の誤差*中間層と出力層
の重み
        h_del[j]*=EPSILON*h_lay[j]*(1.-h_lay[j]);//中間層の誤差=ε*中間層*(1-中間
層)
      }

      for(j=0;j<=HIDDEN;j++)
        for(i=0;i<OUTPUT;i++)
          ho_w[j][i]+=ETA*o_del[i]*h_lay[j];//中間層から出力層への重み=η*出力層の
誤差*中間層
      for(j=0;j<=INPUT;j++)
        for(i=0;i<HIDDEN;i++)
          ih_w[j][i]+=ETA*h_del[i]*i_lay[ctg][j];//入力層から中間層への重み=η*中
間層の誤差*入力層
    }

    fprintf(stderr,"iteration = %4d, error = %1.5f\n",ite,err);
```

```
    /*終了条件*/
    if((err<MIN_ERR)||(ite==ITERATIONS)){
      for(ctg=0;ctg<CTG;ctg++){
        fprintf(stderr,"ctg[%d] : ",ctg);
        for(i=0;i<INPUT;i++)
          fprintf(stderr,"i[%d] = %1.1f  ",i,i_lay[ctg][i]);
        for(j=0;j<HIDDEN;j++){
          for(i=0,sum=0.;i<=INPUT;i++)
            sum+=i_lay[ctg][i]*ih_w[i][j];//sum=入力層の値*入力層から中間層の重み
          h_lay[j]=sigmoid(sum);//中間層の決定
        }
        for(j=0;j<OUTPUT;j++){
          for(i=0,sum=0.;i<=HIDDEN;i++)
            sum+=h_lay[i]*ho_w[i][j];//sum=中間層の値*中間層から出力層の重み
          o_lay[j]=sigmoid(sum);//出力層の決定
          fprintf(stderr,"o[%d] = %1.5f, t{%d] = %1.1f\n",j,o_lay[j],j,teach[ctg][j]);
        }
      }
      fprintf(stderr,"iteration = %4d, error = %1.5f\n",ite,err);
      break;
    }
  }
}

--------------------------------------------------------------------------
/*逐次修正法＋慣性項*/
/*慣性項; 過去の荷重修正を次の修正時に反映させる*/
main(argc,argv)
int argc;
char **argv;
{
  int i,j,ite,ctg;
  double err,sum;
  srand((int)time((long *)0));

  /* ExOR Problem */
  i_lay[0][0]=OFF; i_lay[0][1]=OFF; i_lay[0][2]=ON; teach[0][0]=OFF;
  i_lay[1][0]=ON;  i_lay[1][1]=OFF; i_lay[1][2]=ON; teach[1][0]=ON;
  i_lay[2][0]=OFF; i_lay[2][1]=ON;  i_lay[2][2]=ON; teach[2][0]=ON;
  i_lay[3][0]=ON; i_lay[3][1]=ON;  i_lay[3][2]=ON; teach[3][0]=OFF;

  h_lay[HIDDEN]=ON;

  /*重みをランダムに設定*/
  for(j=0;j<HIDDEN;j++)
    for(i=0;i<=INPUT;i++){
      ih_w[i][j]=WD*drand();
      ih_dw[i][j]=0.;//慣性項の初期化
    }

  for(j=0;j<OUTPUT;j++)
    for(i=0;i<=HIDDEN;i++){
      ho_w[i][j]=WD*drand();
```

```
      ho_dw[i][j]=0.;//慣性項の初期化
    }

  /*学習の開始*/
  for(ite=0;ite<=ITERATIONS; ite++){
    for(ctg=0,err=0.;ctg<CTG;ctg++){
      for(j=0;j<HIDDEN;j++){
        for(i=0,sum=0.;i<=INPUT;i++)
          sum += i_lay[ctg][i]*ih_w[i][j];
        h_lay[j]=sigmoid(sum);
      }
      for(j=0;j<OUTPUT;j++){
        for(i=0,sum=0.;i<=HIDDEN;i++)
          sum+=h_lay[i]*ho_w[i][j];
        o_lay[j]=sigmoid(sum);
      }

      //誤差の計算
      for(i=0;i<OUTPUT;i++){
        err+=sq(teach[ctg][i]-o_lay[i])/2./(double)CTG;
        o_del[i]=EPSILON*(teach[ctg][i]-o_lay[i])*o_lay[i]*(1.-o_lay[i]);
      }
      for(j=0;j<=HIDDEN;j++){
        h_del[j]=0.;
        for(i=0;i<OUTPUT;i++)
          h_del[j]+=o_del[i]*ho_w[j][i];
        h_del[j]*=EPSILON*h_lay[j]*(1.-h_lay[j]);
      }

      /*慣性項の計算 (+αの計算が逐次修正法にはない)*/
      for(j=0;j<=HIDDEN;j++)
        for(i=0;i<OUTPUT;i++){
          ho_dw[j][i]=ETA*o_del[i]*h_lay[j]+ALPHA*ho_dw[j][i];//中間層から出力層
への慣性項=η*出力層の誤差*中間層+α*中間層から出力層の慣性項
          ho_w[j][i]+=ho_dw[j][i];//中間層から出力層の重みに慣性項を加算
        }


      for(j=0;j<=INPUT;j++)
        for(i=0;i<HIDDEN;i++){
          ih_dw[j][i]=ETA*h_del[i]*i_lay[ctg][j]+ALPHA*ih_dw[j][i];//入力層から中
間層への慣性項=η*中間層の誤差*入力層+α*入力層から中間層の慣性項
          ih_w[j][i]+=ih_dw[j][i];//入力層から中間層の重みに慣性項を加算
        }
      }
    fprintf(stderr,"iteration = %4d, error = %1.5f\n",ite,err);
    if((err<MIN_ERR)||(ite==ITERATIONS)){
      for(ctg=0;ctg<CTG;ctg++){
        fprintf(stderr,"ctg[%d] : ",ctg);
        for(i=0;i<INPUT;i++)
          fprintf(stderr,"i[%d] = %1.1f  ",i,i_lay[ctg][i]);
        for(j=0;j<HIDDEN;j++){
          for(i=0,sum=0.;i<=INPUT;i++)
            sum+=i_lay[ctg][i]*ih_w[i][j];
```

```
        h_lay[j]=sigmoid(sum);
      }
      for(j=0;j<OUTPUT;j++){
        for(i=0,sum=0.;i<=HIDDEN;i++)
          sum+=h_lay[i]*ho_w[i][j];
        o_lay[j]=sigmoid(sum);
        fprintf(stderr,"o[%d] = %1.5f, t{%d] = %1.1f\n",j,o_lay[j],j,teach[ctg][j]);
      }
    }
    fprintf(stderr,"iteration = %4d, error = %1.5f\n",ite,err);
    break;
  }
 }
}
```

2. サンプルソース 1 を動かして ExOR の学習がうまくいくかどうかを確認せよ.

```
ex_bp_l.c
ctg[0] : i[0] = 0.1  i[1] = 0.1  o[0] = 0.10004, t{0] = 0.1
ctg[1] : i[0] = 0.9  i[1] = 0.1  o[0] = 0.89996, t{0] = 0.9
ctg[2] : i[0] = 0.1  i[1] = 0.9  o[0] = 0.89996, t{0] = 0.9
ctg[3] : i[0] = 0.9  i[1] = 0.9  o[0] = 0.10005, t{0] = 0.1
iteration = 3133, error = 0.00000

ex_bp_o.c
ctg[0] : i[0] = 0.1  i[1] = 0.1  o[0] = 0.10218, t{0] = 0.1
ctg[1] : i[0] = 0.9  i[1] = 0.1  o[0] = 0.89645, t{0] = 0.9
ctg[2] : i[0] = 0.1  i[1] = 0.9  o[0] = 0.89620, t{0] = 0.9
ctg[3] : i[0] = 0.9  i[1] = 0.9  o[0] = 0.10668, t{0] = 0.1
iteration = 1375, error = 0.00001

ex_bp_mo.c
ctg[0] : i[0] = 0.1  i[1] = 0.1  o[0] = 0.11897, t{0] = 0.1
ctg[1] : i[0] = 0.9  i[1] = 0.1  o[0] = 0.88811, t{0] = 0.9
ctg[2] : i[0] = 0.1  i[1] = 0.9  o[0] = 0.88907, t{0] = 0.9
ctg[3] : i[0] = 0.9  i[1] = 0.9  o[0] = 0.10915, t{0] = 0.1
iteration =   83, error = 0.00010
```

3. サンプルソース 1 で，中間層のユニット数を変更した場合にどのように挙動が変化するかを調べよ．(誤差の収束をグラフ化し，隠れ層のユニット数についての設計方針について考察せよ)

中間層が 2 の場合は下図のようになっている。

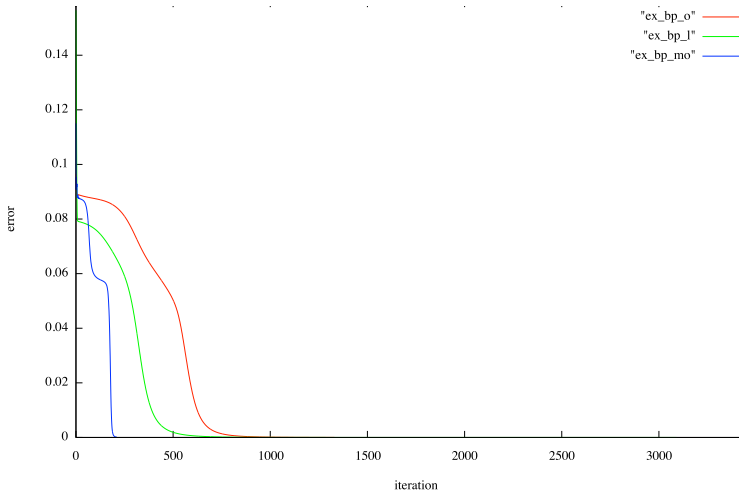

FIGURE 1. pdf ファイルの引用

中間層を増加させ、その収束具合を観測した。左が中間層が 3 、右が中間層が 10 である。

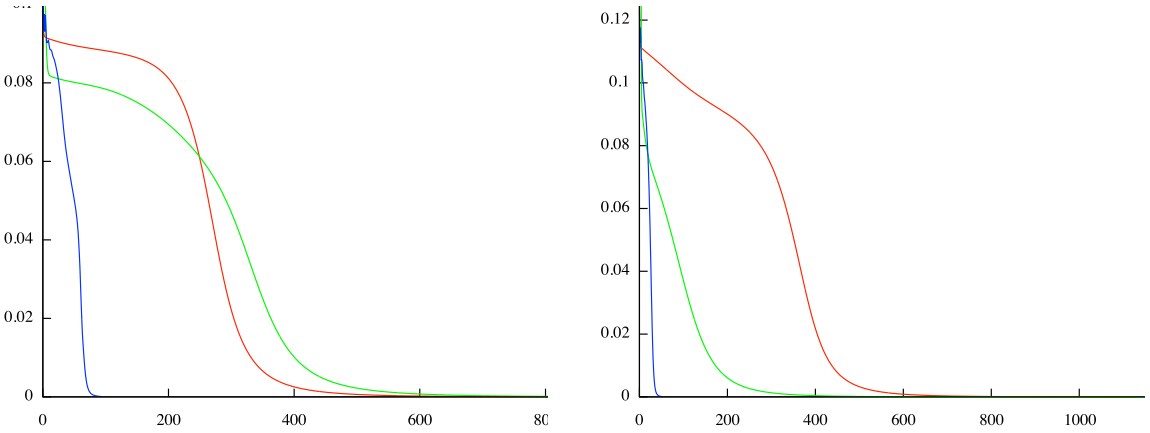

FIGURE 2. pdf ファイルの引用

中間層を 3 にしたところ、収束が早くなったので、単に中間層を増加させれば学習早くなると考えた。なので、中間層を 10 にして実行したところ、ex_bp_l,ex_bp_mo は収束が早くなったが、ex_bp_o は増加してしまった。

4. サンプルソース 2 は“0,1,2,3”の数字文字認識のための NN である．学習用の数字パターンファイルを作成し，認識力の高い NN を学習により完成させよ．評価データを独自に作成し未知のデータに対する認識率ができるかどうかを確認せよ．

以下の学習データを用意した。

```
0      1      2      3      4
11111  00100  11111  11110  10010
10001  01100  00001  00001  10010
10001  00100  01111  11111  10010
10001  00100  11000  00001  11111
11111  11111  11111  11110  00010
```

学習を開始

```
filename? --> ./data/eva1-0.txt
correct = 10000
01110
10001
10001
10001
01110
CHECK filename ./data/eva1-0.txt
EVA o[0] = 0.00716, correct[0] = 1.0
EVA o[1] = 0.00067, correct[1] = 0.0
EVA o[2] = 0.01313, correct[2] = 0.0
EVA o[3] = 0.98004, correct[3] = 0.0
EVA o[4] = 0.00102, correct[4] = 0.0
EVA sum_error = 1.98770
nn> ./data/eva1-1.txt
nn> e
filename? --> ./data/eva1-1.txt
correct = 01000
00100
01100
00100
00100
00100
CHECK filename ./data/eva1-1.txt
EVA o[0] = 0.01600, correct[0] = 0.0
EVA o[1] = 0.98346, correct[1] = 1.0
EVA o[2] = 0.00009, correct[2] = 0.0
EVA o[3] = 0.04062, correct[3] = 0.0
EVA o[4] = 0.00191, correct[4] = 0.0
EVA sum_error = 0.07516
nn> e
filename? --> ./data/eva1-2.txt
correct = 00100
11111
10001
00111
01000
11111
CHECK filename ./data/eva1-2.txt
EVA o[0] = 0.00524, correct[0] = 0.0
EVA o[1] = 0.00138, correct[1] = 0.0
EVA o[2] = 0.97275, correct[2] = 1.0
```

```
EVA o[3] = 0.03349, correct[3] = 0.0
EVA o[4] = 0.00199, correct[4] = 0.0
EVA sum_error = 0.06935
nn> e
filename? --> ./data/eva1-3.txt
correct = 00010
11111
10001
01111
10001
11111
CHECK filename ./data/eva1-3.txt
EVA o[0] = 0.00535, correct[0] = 0.0
EVA o[1] = 0.00074, correct[1] = 0.0
EVA o[2] = 0.94821, correct[2] = 0.0
EVA o[3] = 0.09793, correct[3] = 1.0
EVA o[4] = 0.00179, correct[4] = 0.0
EVA sum_error = 1.85816
nn> e
filename? --> ./data/eva1-4.txt
correct = 00001
00010
00110
01010
11111
00010
CHECK filename ./data/eva1-4.txt
EVA o[0] = 0.00740, correct[0] = 0.0
EVA o[1] = 0.01656, correct[1] = 0.0
EVA o[2] = 0.52655, correct[2] = 0.0
EVA o[3] = 0.04051, correct[3] = 0.0
EVA o[4] = 0.00167, correct[4] = 1.0
EVA sum_error = 1.58935
```

2つしか学習がうまくいかなかった。この失敗を活かし、新しく、教師データを用意、学習させたが、やはり正答率はあがらなかった。
『0』:曲線の部分が多く、どうしても 5×5 のマスでは足りない。
『3』:『2』と被ってしまう (↑『0』と同じ理由?)
『4』:ヨン、という書き方は、2種類ある。ヨンの上を開く場合と、閉じる場合である。今回は教師データを閉じるタイプ、学習データを開くタイプで行った。結果、学習できていなかったので、教師データを2種類用意する必要がある。