

知能システムレポート

高野大悟

2018 年 8 月 3 日

1 問 1 XOR を学習するニューラルネットワークを作成しなさい

作成したプログラムをレポートの末尾に添付する。実行したところ、errorSum の最終的な値が 0.001129 であった。また分類結果を正しい値が 0 の時には 0.05 以下、1 の時には予想値が 0.95 以上が正解として実行すると全て正解だった。xor はパターンが四つに決まっているため、学習しやすい問題だと思うのでこのような高い結果を示したと考える。

2 問 2 大迫データを学習し、半端ない A と半端な B を識別するニューラルネットワークを作成しなさい

作成したプログラムをレポートの末尾に添付する。errorSum は 1.677633 であり問 1 と同様に正答率を調べたところ 84% であった。問 1 と比べ、学習するデータが少数であり複雑になっているため xor より正答率が落ちたと考える。

3 応用

応用問題として問 2 をモーメント法で重みを更新するプログラムを作成した。しかし errorSum が 2.592927 で正答率も 38% と大きく下がってしまった。原因として更新式を間違えている可能性がある。またバッチ法のようにランダムに学習するデータを選び学習するプログラムを作成したが、こちらも正答率がとても低くなってしまい、プログラムをミスが原因と考えられる。

4 プログラムリスト

問 1 のプログラムを以下に示す。

```
/*  
 * NeuralNetwork For XOR  
 *  
 * Input layer: 2  
 * Hidden layer: 2  
 * Output layer: 1
```

```

*/

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

#define EPSILON 4.0
#define ETA 0.1
#define TIMES 1000
#define INIT_WEIGHT 0.3

double randNum(void)
{
    return ((double)rand())/RAND_MAX-0.5)*2.0*INIT_WEIGHT;
}

double sigmoid(double x)
{
    return 1/(1+exp(-1*EPSILON*x));
}

int main(void)
{
    double data[4][3] = {
        {0.0, 0.0, 0.0},
        {0.0, 1.0, 1.0},
        {1.0, 0.0, 1.0},
        {1.0, 1.0, 0.0}
    };
    double wbd, wbe, wcd, wce, wab, wac;
    double offb, offc, offa;
    double outd, oute, outb, outc, outa;
    double xb, xc, xa;
    double deltab, deltac, deltaa;
    int r;
    double error;

```

```

double errorSum;
double accuracy = 0;
int times;
int seed;
FILE *fp;

fp = fopen("error.dat", "w");
if (fp==NULL) {
    printf("can't open file.\n");
    exit(1);
}

//seed = (unsigned int)time(NULL);
//printf("seed = %d\n", seed);
seed = 0;
srand(seed);

wbd = randNum();
wbe = randNum();
wcd = randNum();
wce = randNum();
wab = randNum();
wac = randNum();
offb = randNum();
offc = randNum();
offa = randNum();

for (times=0; times<TIMES; times++) {

    errorSum = 0.0;

    for (r=0; r<4; r++) {

        /* ----- */
        /* Feedforward */
        /* ----- */

```

```

/* Input layer output */
outd = data[r][0];
oute = data[r][1];

/* Hidden layer output */
xb = wbd*outd + wbe*oute + offb;
outb = sigmoid(xb);

xc = wcd*outd + wce*oute + offc;
outc = sigmoid(xc);

/* Output layer output */
xa = wab*outb + wac*outc + offa;
outa = sigmoid(xa);

if(times==TIMES-1) {
    printf("[%d]=%.10f, (%f)\n", r, outa, data[r][2]);

    if(data[r][2] == 0 && outa < 0.05)
        accuracy += 1.0;
    else if(data[r][2] == 1 && outa > 0.95)
        accuracy += 1.0;
}

/* ----- */
/* Back Propagation */
/* ----- */
error = ((outa-data[r][2])*(outa-data[r][2]));
errorSum += error;

/*
 *
 * ここに更新式を書く
 *
 * deltaa = ...
 * wab = wab + ...

```

```

        *
        */
    deltaa = (outa - data[r][2]) * EPSILON * (1.0 - outa) * outa;
    deltab = deltaa * wab * EPSILON * (1.0 - outb) * outb;
    deltac = deltaa * wac * EPSILON * (1.0 - outc) * outc;

    wab = wab - ETA * deltaa * outb;
    wac = wac - ETA * deltaa * outc;
    offa -= ETA * deltaa;
    wbd = wbd - ETA * deltab * outd;
    wbe = wbe - ETA * deltab * oute;
    offb -= ETA * deltab * 1;
    wcd = wcd - ETA * deltac * outd;
    wce = wce - ETA * deltac * oute;
    offc -= ETA * deltac * 1;
}
printf("errorSum = %f\n", errorSum/4.0);
fprintf(fp, "%f\n", errorSum/4.0);

}

printf("accuracy: %lf\n", accuracy/r);
fclose(fp);

return 0;
}

```

問2のプログラムを以下に示す。

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "inputData.h"

#define EPSILON 4.0
#define ETA 0.1
#define TIMES 1000
#define INIT_WEIGHT 0.3

```

```

#define RAW 100

double randNum(void)
{
    return ((double)rand())/RAND_MAX-0.5)*2.0*INIT_WEIGHT;
}

double sigmoid(double x)
{
    return 1/(1+exp(-1*EPSILON*x));
}

int main(void)
{
    double data[RAW*2][3];

    inputData(data);

    double wbd, wbe, wcd, wce, wab, wac;
    double offb, offc, offa;
    double outd, oute, outb, outc, outa;
    double xb, xc, xa;
    double deltab, deltac, deltaa;
    int r;
    double error;
    double errorSum;
    double accuracy = 0.0;
    int times;
    int seed;
    FILE *fp;

    double a, b, c = 0;

    fp = fopen("error.dat", "w");
    if (fp==NULL) {
        printf("can't open file.\n");
    }

```

```

    exit(1);
}

//seed = (unsigned int)time(NULL);
//printf("seed = %d\n", seed);
seed = 0;
srand(seed);

wbd = randNum();
wbe = randNum();
wcd = randNum();
wce = randNum();
wab = randNum();
wac = randNum();
offb = randNum();
offc = randNum();
offa = randNum();

for (times=0; times<TIMES; times++) {

    errorSum = 0.0;

    for (r=0; r<RAW*2; r++) {

        /* ----- */
        /* Feedforward */
        /* ----- */

        /* Input layer output */
        outd = data[r][0];
        oute = data[r][1];

        /* Hidden layer output */
        xb = wbd*outd + wbe*oute + offb;
        outb = sigmoid(xb);

        xc = wcd*outd + wce*oute + offc;

```

```

outc = sigmoid(xc);

/* Output layer output */
xa = wab*outb + wac*outc + offa;
outa = sigmoid(xa);

if(times==TIMES-1) {
    printf("[%d]=%.10f, (%f)\n", r, outa, data[r][2]);
    if(data[r][2] == 0 && outa < 0.1)
        accuracy += 1.0;
    else if(data[r][2] == 1 && outa > 0.95)
        accuracy += 1.0;
}

/* ----- */
/* Back Propagation */
/* ----- */
error = ((outa-data[r][2])*(outa-data[r][2]));
errorSum += error;

deltaa = (outa - data[r][2]) * EPSILON *(1.0 - outa) * outa;
deltab = deltaa * wab * EPSILON *(1.0 - outb) * outb;
deltac = deltaa * wac * EPSILON *(1.0 - outc) * outc;

wab = wab - ETA * deltaa * outb;
wac = wac - ETA * deltaa * outc;
offa -= ETA * deltaa;
wbd = wbd - ETA * deltab * outd;
wbe = wbe - ETA * deltab * oute;
offb -= ETA * deltab * 1;
wcd = wcd - ETA * deltac * outd;
wce = wce - ETA * deltac * oute;
offc -= ETA * deltac * 1;

}
printf("errorSum = %f\n", errorSum/4.0);
fprintf(fp, "%f\n", errorSum/4.0);

```



```
}  
printf("accuracy: %lf\n", accuracy/r);  
fclose(fp);  
return 0;  
}
```