知能システム前期末レポート

15422 篠田拓樹

2019 年 7 月 22 日

1　XOR を学習するニューラルネットワーク

2　魚を識別するニューラルネットワーク

3　プログラムリスト

リスト 1　sampleBP.c

```c
/*
 * 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;
  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]);
      }

      /* ------------------ */
      /* 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-outa)*outa;
      deltab = (deltaa * wab)*EPSILON*(1-outb)*outb;
      deltac = (deltaa * wac)*EPSILON*(1-outc)*outc;

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

  }

  fclose(fp);

  return 0;
}
```