Neural Network I: Fundamental Theory and Applications

Project II

Project member: ①Atsushi Takamiya（s1260013）
②Tomonori Suzuki （s1260032）
③Taiki Watanabe （m5251152）
④Kaito Suzuki （m5251144）
⑤Kaito Iwamatsu （m5251115）

• Project 2:

a,b,c)

a) Make a computer program for the BP algorithm. Test the program using the 4-bit parity check problem.

b) BP algorithm

c)

・Increasing number of hidden neurons shorten learning time.

・Also, it may improve accuracy because a classifier can use more information to classify.

・ In case of the number of hidden neurons is 4, the accuracy is low. Therefore, we concluded it is difficult to solve this problem with 4 hidden neurons because it is not enough boundaries to classify.

The number of hidden neurons is 4.

```
Error in the 132258-th learning cycle = 0.001000


The connection weights in the output layer:
23.898368 14.007705 -39.629622 -7.145658


The connection weights in the hidden layer:
-9.898973 -9.894098 -9.891405 -9.892650 -24.593709
-7.401936 -7.428954 -7.435708 -7.419668 -2.287090
-1.528985 -1.529109 -1.528681 -1.527434 -3.722874


Check the output results:
Input {0,0,0,0,-1} Output {1}
Input {0,0,0,1,-1} Output {0}
Input {0,0,1,0,-1} Output {0}
Input {0,1,0,0,-1} Output {0}
Input {1,0,0,0,-1} Output {0}
Input {0,0,1,1,-1} Output {0}
Input {0,1,1,0,-1} Output {0}
Input {1,1,0,0,-1} Output {0}
Input {0,1,0,1,-1} Output {0}
Input {1,0,1,0,-1} Output {0}
Input {1,0,0,1,-1} Output {0}
Input {0,1,1,1,-1} Output {1}
Input {1,0,1,1,-1} Output {1}
Input {1,1,0,1,-1} Output {1}
Input {1,1,1,0,-1} Output {1}
Input {1,1,1,1,-1} Output {1}
```

The number of hidden neurons is 6.

```
Error in the 22038-th learning cycle = 0.001000


The connection weights in the output layer:
0.922975 -12.047839 12.662219 -12.775470 11.877464 -5.378598


The connection weights in the hidden layer:
-1.252344 -2.040781 -2.068052 -2.093854 1.620458
-3.214726 -3.217241 -3.216340 -3.215458 -11.126956
-6.783770 -6.812683 -6.811545 -6.810427 -16.190931
-6.650884 -6.648417 -6.657647 -6.660015 -9.171266
-5.927486 -5.953684 -5.979712 -5.971880 -3.009295


Check the output results:
Input {0,0,0,0,-1} Output {1}
Input {0,0,0,1,-1} Output {0}
Input {0,0,1,0,-1} Output {0}
Input {0,1,0,0,-1} Output {0}
Input {1,0,0,0,-1} Output {0}
Input {0,0,1,1,-1} Output {1}
Input {0,1,1,0,-1} Output {1}
Input {1,1,0,0,-1} Output {1}
Input {0,1,0,1,-1} Output {1}
Input {1,0,1,0,-1} Output {1}
Input {1,0,0,1,-1} Output {1}
Input {0,1,1,1,-1} Output {0}
Input {1,0,1,1,-1} Output {0}
Input {1,1,0,1,-1} Output {0}
Input {1,1,1,0,-1} Output {0}
Input {1,1,1,1,-1} Output {1}
```

The number of hidden neurons is 8.

```
Error in the 27848-th learning cycle = 0.001000


The connection weights in the output layer:
-13.529778 -9.213486 -8.367947 12.701961 3.072105 3.110826 12.306624 -3.914031


The connection weights in the hidden layer:
-6.848348 -6.443674 -6.440201 -6.494663 -9.995439
7.422092 -5.869096 -5.839608 -5.887143 -6.661017
-6.370728 0.811977 0.530315 1.037875 -0.218006
-6.115949 -6.132749 -6.130760 -6.075984 -3.076393
-2.566094 -2.533029 -2.688243 -2.802247 -0.434685
-2.222209 -2.592056 -2.697718 -2.825996 -0.308560
-5.741422 -7.575100 -7.641279 -7.522927 -16.402300


Check the output results:
Input {0,0,0,0,-1} Output {1}
Input {0,0,0,1,-1} Output {0}
Input {0,0,1,0,-1} Output {0}
Input {0,1,0,0,-1} Output {0}
Input {1,0,0,0,-1} Output {0}
Input {0,0,1,1,-1} Output {1}
Input {0,1,1,0,-1} Output {1}
Input {1,1,0,0,-1} Output {1}
Input {0,1,0,1,-1} Output {1}
Input {1,0,1,0,-1} Output {1}
Input {1,0,0,1,-1} Output {1}
Input {0,1,1,1,-1} Output {0}
Input {1,0,1,1,-1} Output {0}
Input {1,1,0,1,-1} Output {0}
Input {1,1,1,0,-1} Output {0}
Input {1,1,1,1,-1} Output {1}
```

The number of hidden neurons is 10.

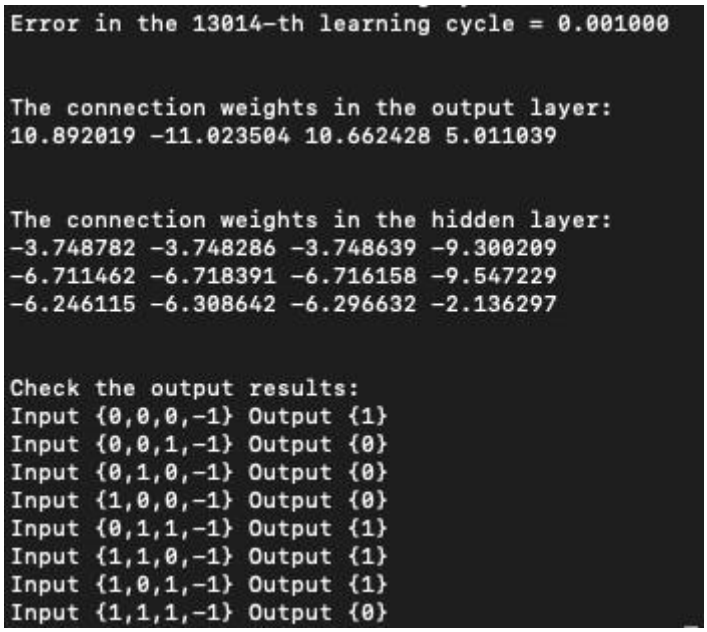
```
Error in the 22516-th learning cycle = 0.001000


The connection weights in the output layer:
-0.131491 -8.136651 1.792174 -10.164666 12.957578 12.364125 0.697661 -12.032424 -2.077469 -4.548972


The connection weights in the hidden layer:
-1.027784 -1.419498 -1.445453 -1.379935 1.611321
-6.382677 0.721982 1.453282 1.201857 0.906571
-1.815886 -2.087010 -2.214566 -2.218853 0.727978
4.753719 -4.565830 -4.478232 -4.570353 -6.115825
-6.535914 -6.067164 -6.089062 -6.059007 -3.229788
-6.046864 -7.018321 -7.006784 -6.975167 -15.432667
-1.469454 -1.738050 -1.775606 -1.803468 1.526701
-6.278492 -6.617527 -6.631656 -6.642510 -9.550682
-2.071674 0.427317 -0.423299 0.274394 0.116451


Check the output results:
Input {0,0,0,0,-1} Output {1}
Input {0,0,0,1,-1} Output {0}
Input {0,0,1,0,-1} Output {0}
Input {0,1,0,0,-1} Output {0}
Input {1,0,0,0,-1} Output {0}
Input {0,0,1,1,-1} Output {1}
Input {0,1,1,0,-1} Output {1}
Input {1,1,0,0,-1} Output {1}
Input {0,1,0,1,-1} Output {1}
Input {1,0,1,0,-1} Output {1}
Input {1,0,0,1,-1} Output {1}
Input {0,1,1,1,-1} Output {0}
Input {1,0,1,1,-1} Output {0}
Input {1,1,0,1,-1} Output {0}
Input {1,1,1,0,-1} Output {0}
Input {1,1,1,1,-1} Output {1}
```

d,e,f)

d) Is it enough to have 4 hidden neurons if the input dimension is lowered?

e) We ran the program to solve 3-bit parity problem. (Input:4, hidden neurons:4)

f) From result, the required number of hidden neurons depend on difficulty in judgment.



Project Program (Major changes)

```
#define I                5
#define J                10
#define n_sample         16

double x[n_sample][I]={
   {0,0,0,0,-1},
   {0,0,0,1,-1},
   {0,0,1,0,-1},
   {0,1,0,0,-1},
   {1,0,0,0,-1},
   {0,0,1,1,-1},
   {0,1,1,0,-1},
   {1,1,0,0,-1},
   {0,1,0,1,-1},
   {1,0,1,0,-1},
   {1,0,0,1,-1},
   {0,1,1,1,-1},
   {1,0,1,1,-1},
   {1,1,0,1,-1},
```

```
    {1,1,1,0,-1},
    {1,1,1,1,-1}
};
double d[n_sample][K]={1,0,0,0,0,1,1,1,1,1,1,0,0,0,0,1};

void PrintResult(void){
    int i,j,k;

    printf("¥n¥n");
    printf("The connection weights in the output layer:¥n");
    for(k=0; k<K; k++){
        for(j=0; j<J; j++)
            printf("%5f ",w[k][j]);
        printf("¥n");
    }

    printf("¥n¥n");
    printf("The connection weights in the hidden layer:¥n");
    for(j=0; j<J-1; j++){
        for(i=0; i<I; i++)
            printf("%5f ",v[j][i]);
        printf("¥n");
    }
    printf("¥n¥n");

//output check
    int temp[J-1];
    int temp2;
    printf("Check the output results:¥n");
    for(i=0; i<n_sample; i++){

        printf("Input {");
        for(j=0; j<I; j++){
            if(j == I-1){
printf("%1.f} ",x[i][j]);
            } else {
```

```
printf("%1.f,",x[i][j]);
        }
      }

      printf("Output {"); // v[J][I]
      for(j=0; j<J-1; j++){
        if((v[j][0]*x[i][0] + v[j][1]*x[i][1]+ v[j][2]*x[i][2]+ v[j][3]*x[i][3]) > v[j][4])
temp[j] = 1;
        else
temp[j] = 0;
      }
      temp2=0;
      for(j=0; j<J-1; j++){
        temp2 += w[0][j]*temp[j];
      }
      if(temp2 > w[0][J-1])
        printf("1}¥n");
      else
        printf("0}¥n");
   }
}
```