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: 1](#_Toc70112266)

[a,b,c) 1](#_Toc70112267)

[d,e,f) 5](#_Toc70112268)

[Project Program 5](#_Toc70112269)

# 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

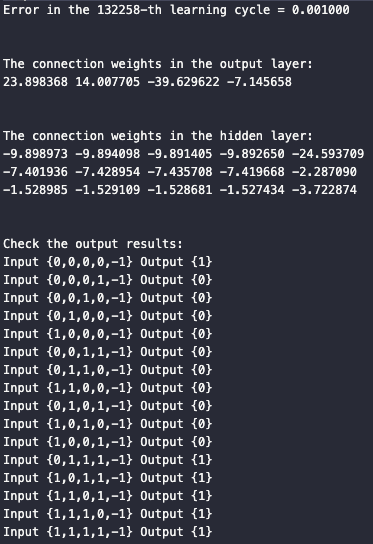
c)

・Increasing number of hidden neurons shorten learning time.

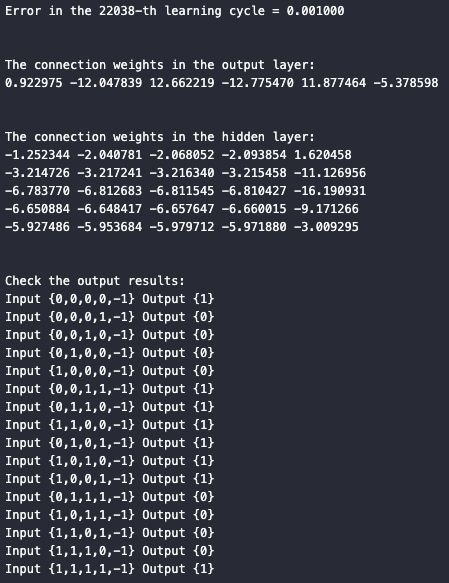
・Also, it may improve accuracy because a classifier can use more information to classify.

・ 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.

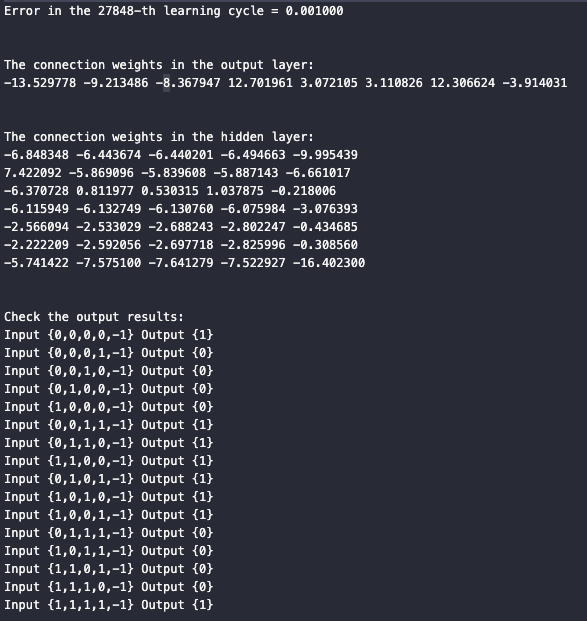
The number of hidden neurons is 4.



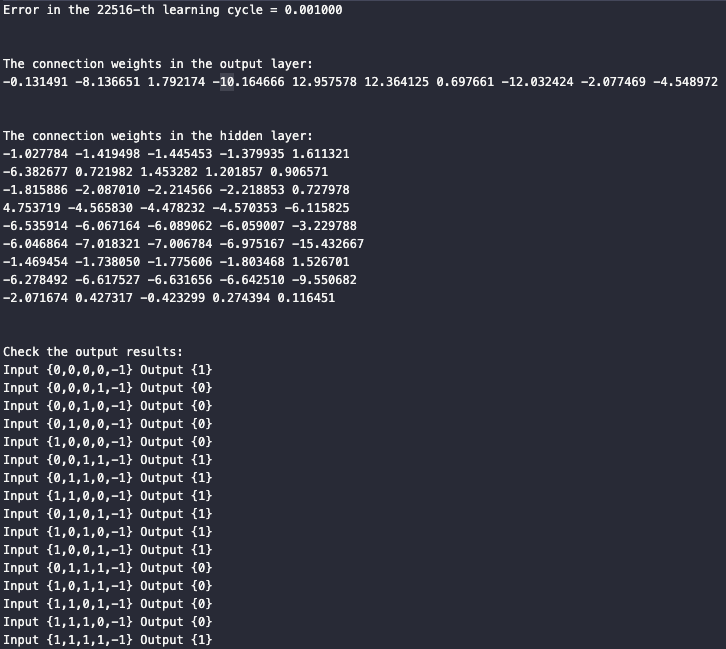
The number of hidden neurons is 6.



The number of hidden neurons is 8.



The number of hidden neurons is 10.

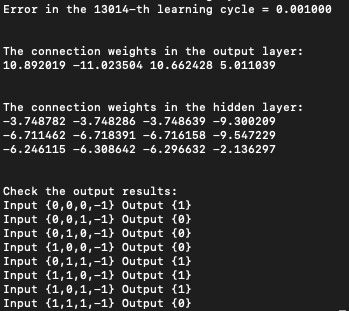


## d,e,f)

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");

}

}