Neural Network I: Fundamental Theory and Applications

Project IV

Project member:

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

[ Project Ⅳ: 1](#_Toc71998705)

[a,b,c) 1](#_Toc71998706)

[d,e,f) 2](#_Toc71998707)

[Project Program (Major changes) 3](#_Toc71998708)

[a,b,c) 3](#_Toc71998709)

[d,e,f) 4](#_Toc71998710)

# Project Ⅳ:

## a,b,c)

a) Modify the program, and test the program using the database Iris.

b) NNC (Nearest Neighbor Classifier) Winner-take-all learning

c)

カレンダー

中程度の精度で自動的に生成された説明カレンダー

中程度の精度で自動的に生成された説明テキスト

自動的に生成された説明カレンダー

中程度の精度で自動的に生成された説明

As a result, they were roughly classified according to the class of the dataset.

I think there are better parameter.

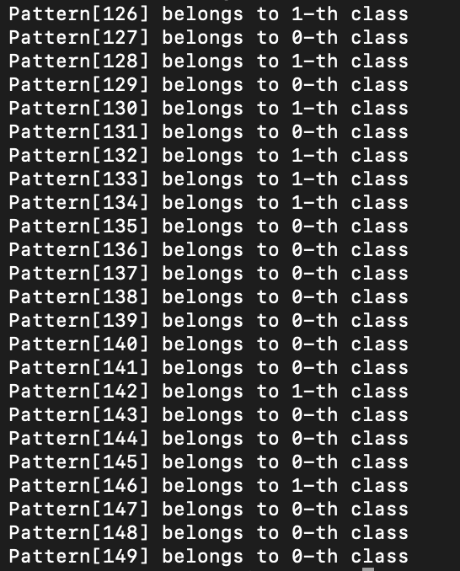
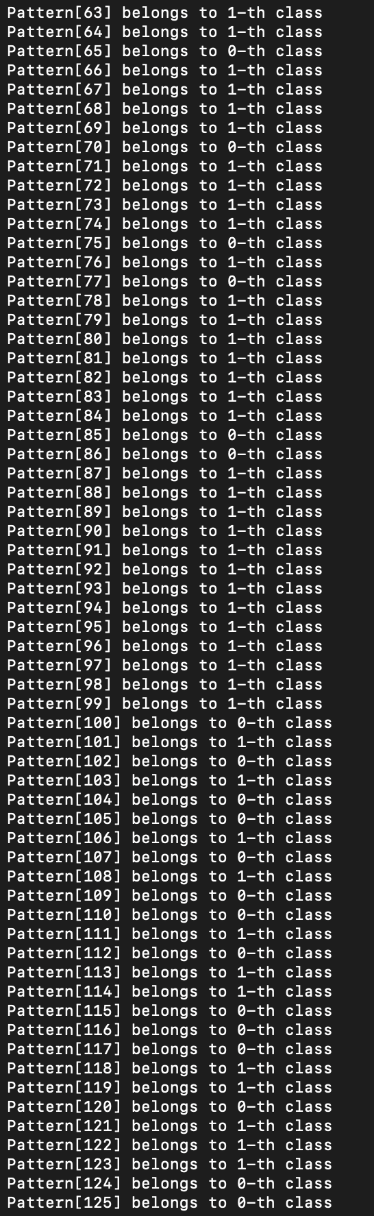
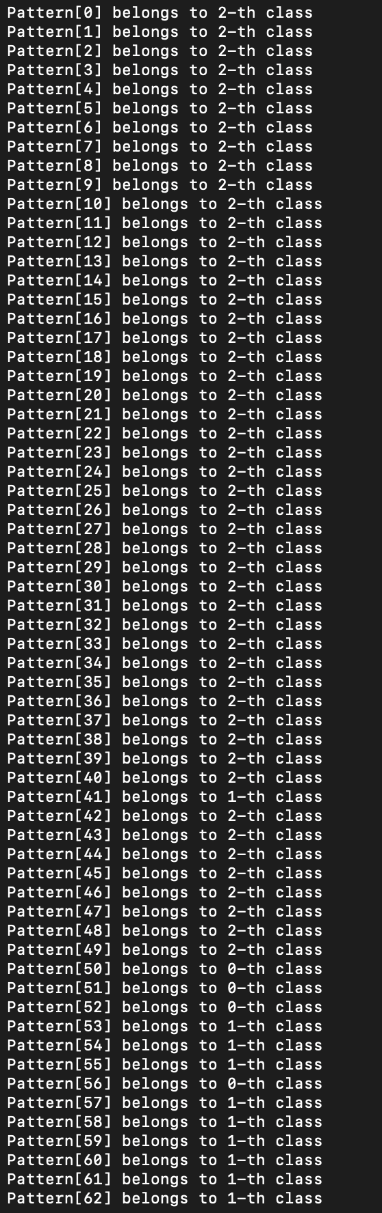
Learning result depends on initial values of weight.

## d,e,f)

d) Is it possible to classify by the number of classes == the number of neurons?

e) Make the program efficient by using feature scaling (standardization)

f)



Learning was able to be performed efficiently by feature scaling.

## Project Program (Major changes)

## a,b,c)

#define I 4

#define M 400

#define P 150

#define alpha 0.1

#define n\_update 500

Change the input data “double x[P][I]” to Iris database.

## d,e,f)

#define I 4

#define M 3

#define P 150

#define alpha 0.1

#define n\_update 1000

// standardization

double sum[I];

double ave[I];

double b[I];

// sum of each feature

for(i=0; i<I; i++){

sum[i]=0;

ave[i]=0;

for(int j=0; j<P; j++){

sum[i]+=x[j][i];

}

}

// average

for(i=0; i<I; i++){

ave[i]=sum[i]/P;

}

// standard deviation

for(i=0; i<I; i++){

b[i]=0;

for(int j=0; j<P; j++){

b[i]+=pow(x[j][i]-ave[i],2.0);

// b[i]+=x[j][i]-ave[i];

}

b[i] = sqrt(b[i]/=P);

}

// standardization

for(i=0; i<I; i++){

for(int j=0; j<P; j++){

x[j][i] = (x[j][i] - ave[i]) / b[i];

}

}