Page1:

・We would like to start our presentation.

Page2:

・We explain in this outline.

・First is introduction about what we done.

・Second is explanation about the main technique of this presentation.

・Next is the effect of standardization on Project 4 and 5.

・Finally, we show the result with other dataset.

Page3:

・First, We done 4 and 5 projects, which is the classification problem of iris variety.

・However, the results weren't good.

・And, we concluded that the difference in each feature scale of the data is one of cause .

・Therefore, we improved the results by standardizing the dataset.

Page4:

・Next, let's explain about standardization.

・This can be concluded that the working process to fit the average to 0 and the standard deviation to 1.

・The formula is below.

・Thereby, the scale of each feature are arranged.

・For example, the iris dataset used the projects have four features: sepal length and width, petal length and width.

・And there are obviously difference of size.

・If standardization is not performed, the influence of sepal length and width which are larger in four features will be strong regardless of the weight value.

・Therefore, if the project depends on petal length and width, a model can not learn well.

Page5:

Problem of Project 4 is Classify Iris dataset using Winner-take-all algorithm.

Iris data set contains 3 classes of 50 instances each.

To explain the winner-take-all learning algorithm simply,

when the weight vector of a neuron is close to the input pattern, it is moved towards this pattern.

This is flow of the algorithm.

Step1: Initialize all weights with random numbers.

Step2: Identify the winner.

Step3: Update the weights of the winner.

As Predict result of this project, we want to classify the input into three classes.

Page6:

We first set the number of neurons to 3, the learning rate to 0.1(ゼロポイントワン), and the number of training sessions to 20, and ran the program. However, as you can see on the left side of the slide, with those parameters, all the patterns were classified into the same class and were not correctly classified. So, we increased the number of neurons little by little, and around 400, the patterns started to be classified into three classes.

However, since the input data was classified into three classes, increasing the number of clusters too much would increase the amount of computation and take a long time, so we thought of other methods. So, we standardized the input data.

Page7:

As a result, we were able to classify them into three classes with three numbers of neurons. Please see the result for the number of neurons 3 on the right side of the slide.

In the case of no standardization, the classification results are biased towards class 0.

150 are not perfectly classified, but if you look at the maximum value of the patterns, they are classified into three classes.

Page8:

I would like to talk Project V “SOFM”.

SOFM is Self-organizing feature map. SOFM is algorithm extended the Kohonen network and it is useful for visualizing the pattern space. This is achieved by reducing the dimensionality of the pattern space.

The neurons are usually arranged in a 2 dimensional planar array with hexagonal neighborhoods and the weights of all neurons in the neighborhood of the winner are updated. The size of the neighborhood is reduced during learning.

This is SOFM algorithm steps.

In step1, select a sample at random from the training set .

In step2, find the neuron closest to the input .

And in last step, update the weight of the neuron if it is in the neighborhood of the neuron .

In project V case, the SOFM predict result is that Iris dataset will be classified into three types and visualized the pattern space by reducing the dimensionality.

Page9:

This is the results of SOFM using the iris dataset.

One alphabet represents one group.

The left image is the non-standardization one, and the right image is the standardized one.

For the non-standardized result, we can see B and C are mixed together.

For the standardized(すたんだだいずど) result, we can see A, B, and C are clearly separated.

Page10:

We will verify the effectiveness of standardization by modifying the dataset and checking the results.

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.

This dataset contains 3 classes of almost 60 instances for each. And it has 13 attributes.

In the experiment, run in each project program and check the result.

Page11:

We tried Winner-take-all Algorithm with wine dataset.

This is the result.

Non-standardized data is classified into one class.

For the standardized data, patterns 60~130 cannot be classified well, but overall, the dataset can be classified into three classes.

Page12:

This is the results of SOFM using the wine data set.

The image on the left shows the data with non-standardized and the image on the right shows the data with standardized.

For the non-standardized result, we can see that A and B are separated, but C is mixed in with the A and B groups.

For the standardized result, we can see that the grouping is cleaner than the non-standardized result.

Page13:

We used standardization to solve the problems, and it brought good results.

Machine learning depends on data, so preprocessing data is important.

And, there are various methods of preprocessing, from basic methods such as normalization and standardization to applied methods using statistics.

Therefore, using a method that matches with data is also important rather than simply performing(パフォーミング) preprocessing.

We are going to continue to deepen our understanding.

Thank you for listening.

Page14:

Page15:

Page16: Parameters of Project4

**ニューロンの数を400にした理由：**

・When we increased the number of neurons, it happened to work well at 400.

・Else parameters is almost the default.

n\_update: we try change it but, outputs didn't change.

In case of not using standardization, we increased

Page17: Parameters of Project5

**重み変更のスケールを小さくした理由：**

By reducing effect on surroundings weight, classifying become neatly.

SOFMのパラメータの設定  
学習率と範囲について

最初は重みがランダムに初期化されるため、学習率と近傍サイズには大きな値を設定します。ある程度学習した後、学習率と近傍サイズの値を小さく設定して、ある値に収束させます。

Initially, the weight is initialized randomly, so we set a large value for the learning rate and neighborhood size. After learning to some extent, we set the learning rate and neighborhood size values small so that they would converge to a certain value.

neighborhood size の値は大きく設定すると、うまく分類できなかったので、値を小さくしました。 When we set the neighborhood size value to a large value, it could not be classified well, so we reduced the value.