# 修士論文の和文要旨

| | | | |
|---|---|---|---|
| 研究科・専攻 | 大学院　情報理工学 研究科　情報・ネットワーク工学 専攻　博士前期課程 | | |
| 氏　　　名 | ZHENG ZIDE | 学籍番号 | 2031109 |
| 論 文 題 目 | Sparse feature selection with non-convex matrix regularization in deep neural network<br>(非凸行列正則化によるスパース特徴選択を用いた深層学習) | | |

要　　旨

機械学習において、ニューラルネットワーク（neural network, NN）は優れたデータフィッティング能力を持つため広く使われているが、過学習しやすい問題がある。過学習の二つの主な原因は、ノイズ、と多くの無関係な特徴がモデル訓練に使われることである。センサーやIoT（Internet of Things）技術がデータ収集に重要な役割を果たすようになった一方、環境条件などの制約により、データ収集の過程にノイズの混入が不可避である。ニューラルネットワークモデルを訓練するとき、識別などの訓練目的と無関係な特徴が用いられる。

特徴選択は、収集されたデータの特徴集合から有用な特徴を選択する。特徴選択は学習過程を加速すること、データ記憶コストを減らすこととノイズや過学習の緩和などが可能であり、頑健なモデルを構築するために不可欠なプロセスである。L2,1-2 ノルムのスパースを利用した特徴選択が Miao らによって提案され、良いパフォーマンスを示した。

本研究は、L2,1-2 ノルムを用いて、ニューラルネットワークに特徴選択を導入する手法を提案する。スパース性を持つ L2,1 ノルムと比べて、L2,1-2 ノルムはよりスパースな解が得られる。このため、L2,1-2 ノルムを用いるとき、より強い特徴選択効果とノイズの影響を減少することができると考えられる。提案法は特徴選択が目的であるため、モデルの構造として、入力層だけ L2,1-2 ノルムを加える。そして、より頑健なモデルを得るため、他の層に L2,2 ノルム（Frobenius norm）正則化項を加える。提案法は、L2,2 ノルム正則化より高い分類精度を得られる方法と考えられる。

五つのオープンデータセットを用いて実験を行った。正則化項なし、全部層が L2,2 ノルム正則化、入力層 L2,1 ノルム正則化他の層が L2,2 ノルム正則化と提案法合わせて四組の実験を行った。五分割交差検定の結果より、提案法が一番良い分類精度が得られた。L2,1 ノルムのスパース特性に関する実験も行い、提案法がよりスパースな解とより高い分類精度が得られた。提案法と L2,2 ノルム正則化について t 検定も行い、有意差が示された。結論として、提案法が L2,2 ノルム正則化と比べて、分類精度がより高く、より頑健なモデルを構築することができた。

# Sparse feature selection with non-convex matrix regularization in deep neural network

Washizawa Lab

2031109

Zheng Zide

Supervisor: Washizawa Yoshikazu associate professor

2022/01/20

# Contents

# Chapter 1

# Introduction

In machine learning, an artificial neural network (ANN) is a model which is enlightened from the concept of neurons in the human brain. With long time of continuous development and improvement, ANN has become an important part of machine learning and applied to many fields such as computer vision, natural language processing, and image classification [1, 2].

The back propagation [3] is an important and widely used algorithm to update the parameters while we train the model through samples. The back propagation neural network is widely used because of its excellent data fitting ability, however it is prone to over-fit.

As sensors of the internet of things (IoT) play an important role in data collection, meanwhile due to the limitation of environmental conditions and other factors, noise is inevitable in the process of data collection. In machine learning, the over-fitting is a common problem that the model fails to fit or predict new data reliably but performs well for the training data. The main reasons for the over-fitting problem are noise and using too much irrelevant features during the model training which results in the model being too complex. We can avoid the over-fitting problem by reducing impact of noise and selecting relevant features.

In the neural networks, several methods are used to avoid the over-fitting problem such like the dropout [4], the early stopping, and the regularization. In order to simplify the model, the dropout tries to shut-down some neurons during the training process. In detail, the dropout closes a certain proportion of neurons randomly in each layer of the network. The shortcoming of the dropout is the uncertainty that the cost function cannot be defined explicitly. The early stopping tries to stop the training process when it is considered that the data is over-fitted. The regularization prevents the over-fitting problem by adding a regularization term to the optimization problem. In the neural network, a common practice is using the regularization in all layers, and a most widely used method is using the $\ell_2$ norm regularization [5]. However, all the features are used to train the models which means that the influence of irrelevant features are considered in the training process.

Feature selection aims to select relevant subset features from the original feature set [6]. Nowadays, feature selection is an indispensable part in obtaining robust model because it can speed up the learning process, reduce the data storage cost, and relax noise influence and the over-fitting problems [7]. Sparse feature selection aims to obtain a sparse matrix or vector to select features by adding a sparsity regularization term to the loss function. The sparsity regularization term forces the weight of some features to be very small or zero, thus those corresponding features can be ignored and the rest of the features will be selected. Due to the interpretability, convenience and the good performance, sparse feature selection is widely used to select relevant features and build robust models [8, 9].

The $\ell_{2,1}$ norm of a matrix is equal to the sum of the $\ell_2$ norm of all row vectors in the matrix, while the Frobenius ($\ell_{2,2}$) norm is the square root of the square sum of all elements. In [10], the $\ell_{2,1-2}$ norm is proposed to select features and achieved a excellent classification performance, and the $\ell_{2,1-2}$ norm is defined as the difference of the $\ell_{2,1}$ norm and the Frobenius norm. More important, comparing with the $\ell_{2,1}$ norm, the $\ell_{2,1-2}$ norm is more likely to obtain sparser solution. The $\ell_{2,1-2}$ norm is non-convex but Lipschitz continuous which makes it optimizing easily. Besides, Lipschitz continuous means that the gradient of the $\ell_{2,1-2}$ norm is bounded.

In this paper, motivated by the sparsity of the $\ell_{2,1-2}$ norm and the good performance of using the $\ell_{2,1-2}$ norm to select features, we apply it to the back propagation neural network model to select relevant features and make the model more robust. In details, considering about the sparsity of the $\ell_{2,1-2}$ norm, we propose the $\ell_{2,1-2}$ norm in the input layer. Meanwhile, in order to enhance the robustness of the model and reduce the impact of noise further, we use the Frobenius norm regularization in the rest of the layers. As a result, the proposed method is considered to exhibit better classification performance than the Frobenius norm regularization.

The advantages of using the $\ell_{2,1-2}$ norm are 1) sparser solution lead to better feature selection effect; and 2) reducing the impact of noise. As we mentioned, the $\ell_{2,1-2}$ norm regularization can obtain a sparser solution, it means that more irrelevant features are not involved in the classification process. At the same time, a sparser solution also means a better reduction in the intake of noise. We can consider both noise and feature selection because we propose the $\ell_{2,1-2}$ norm only in the input layer. It can reduce the impact of noise and select relevant features.

The thesis consists of five chapters and the rest parts are organized as follows. Chapter 2 introduces the method to prevent the over-fitting problem in the back propagation neural network and some feature selection methods. Chapter 3 introduces the proposed method. Chapter 4 presents the experimental results of the proposed method and compares with the other methods. Chapter 5 concludes the thesis and discusses the future work.

# Chapter 2

# Related Work

In this chapter, we introduce some basic definitions and concepts, related research, and some necessary knowledge. We use bold uppercase English letter to represent matrices, bold lowercase English letter to represent vectors, and non-bold English letter to represent scalars.

## 2.1 Feature selection with sparse learning

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbf{R}^{d \times n}$ be a data matrix with $n$ samples of $d$ features. Suppose all the samples can be classified to either one of $c$ classes. Feature selection with sparse learning aims to obtain a row sparse solution $\mathbf{W}$. Mathematically, it is described as

$$\min_{\mathbf{W}} L(\mathbf{W}) + \alpha R(\mathbf{W}), \tag{2.1}$$

where $L(\mathbf{W})$ is the loss function, $R(\mathbf{W})$ is the regularization term with sparsity, and $\alpha > 0$ is called the regularization parameter that controls the proportion of those two terms. The effect of the regularization can be changed by adjusting $\alpha$. The regularization parameter $\alpha$ becomes lager, the regularization effect becomes stronger.

As shown in Figure 2.1, once we obtain the row sparse solution $\mathbf{W}$, then features can be selected by $\mathbf{W}^T\mathbf{x}$. In detail, $\mathbf{W}$ is supposed to be row sparse that the *i-th* row of $\mathbf{W}$ is all zero, then in $\mathbf{W}^T\mathbf{x}$, the *i-th* feature in sample $\mathbf{x}$ is ignored (multiply by zero), the rest of the features are selected.

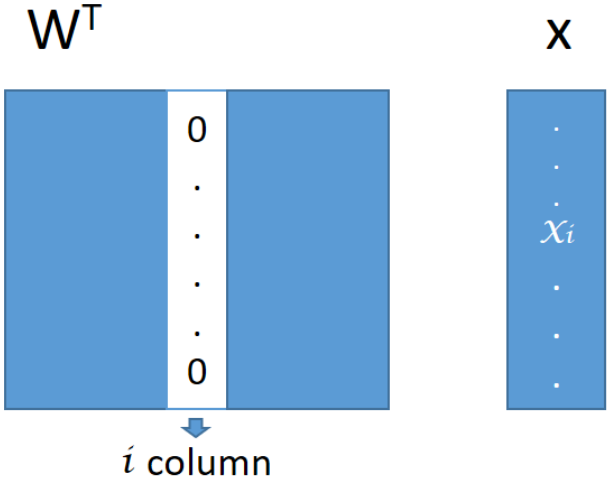


Figure 2.1: Feature selected by sparse matrix

## 2.2 Matrix norm and its gradient

For a vector $\mathbf{w} = [w_1, \ldots, w_n]$, the $\ell_2$ norm of $\mathbf{w}$ is defined as

$$\|\mathbf{w}\|_2 = \sqrt{\sum_{i=1}^{n} w_i^2} \ . \tag{2.2}$$

The $\ell_{2,1}$ norm and the Frobenius norm of matrix $\mathbf{W} \in \mathbf{R}^{d \times c}$ are defined as

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{c} \mathbf{W}_{i,j}^2}\ , \tag{2.3}$$

and

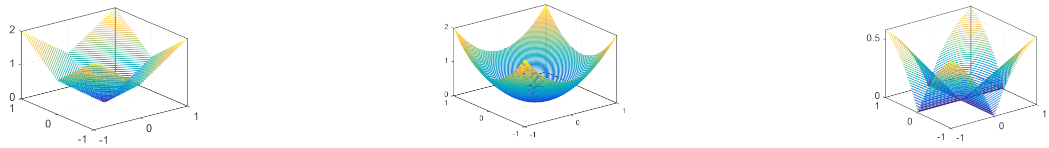
$$\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{c} \mathbf{W}_{i,j}^2}\ . \tag{2.4}$$

From Eq. (2.3), we know that the $\ell_{2,1}$ norm of $\mathbf{W}$ is the sum of $\ell_2$ norm of all row vectors in the matrix. As the difference of the $\ell_{2,1}$ norm and the Frobenius norm, the $\ell_{2,1-2}$ norm of $\mathbf{W}$ is defined as

$$\|\mathbf{W}\|_{2,1-2} = \|\mathbf{W}\|_{2,1} - \|\mathbf{W}\|_F. \tag{2.5}$$

As the $\ell_{1-2}$ norm (difference of the $\ell_1$ norm and $\ell_2$ norm of the vector) is non-convex and Lipschitz continuous, the $\ell_{2,1-2}$ norm is also non-convex and Lipschitz continuous.

Figure 2.2[1] is the contour plot of the $\ell_{2,1}$ norm, the Frobenius norm, and the $\ell_{2,1-2}$ norm. As shown in those figures, the $\ell_{2,1-2}$ norm is more likely to obtain a sparse solution during the minimizing process.



(a) $\ell_{2,1}$ (b) $\ell_{2,2}$ (c) $\ell_{2,1-2}$

Figure 2.2: Contour plot of three matrix norm

[1] Figure from [10]

For a given matrix $\mathbf{W} \in \mathbf{R}^{d\times c}$, the gradient of $\|\mathbf{W}\|_F$ is

$$\frac{\partial}{\partial \mathbf{W}}(\|\mathbf{W}\|_F) = \begin{cases} \frac{\mathbf{W}}{\|\mathbf{W}\|_F}, & \text{if } \mathbf{W} \neq 0 \\ 0, & \text{if } \mathbf{W} = 0, \end{cases} \tag{2.6}$$

and the gradient of $\|\mathbf{W}\|_{2,1}$ is

$$\frac{\partial}{\partial \mathbf{W}}(\|\mathbf{W}\|_{2,1}) = [\phi(\mathbf{w}_1)^T, \phi(\mathbf{w}_2)^T, \ldots, \phi(\mathbf{w}_c)^T]^T, \tag{2.7}$$

where

$$\phi(\mathbf{w}_i) = \begin{cases} \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2}, & \text{if } \mathbf{w}_i \neq 0 \\ 0, & \text{if } \mathbf{w}_i = 0, \end{cases} \tag{2.8}$$

and $\mathbf{w}_i$ is the *i-th* row vector of $\mathbf{W}$. Thus, the gradient of $\|\mathbf{W}\|_{2,1-2}$ is given by Eqs. (2.6) and (2.7), it is

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}}(\|\mathbf{W}\|_{2,1-2}) &= \frac{\partial}{\partial \mathbf{W}}(\|\mathbf{W}\|_{2,1}) - \frac{\partial}{\partial \mathbf{W}}(\|\mathbf{W}\|_F) \\ &= \begin{cases} [\phi(\mathbf{w}_1)^T, \phi(\mathbf{w}_2)^T, \ldots, \phi(\mathbf{w}_c)^T]^T - \frac{\mathbf{W}}{\|\mathbf{W}\|_F}, & \text{if } \mathbf{W} \neq 0 \\ 0, & \text{if } \mathbf{W} = 0. \end{cases} \end{aligned} \tag{2.9}$$

## 2.3 Gradient descent

The gradient descent (GD) is one of the most widely used optimization algorithms. GD finds a local extremum of a differentiable function. As the gradient represents the direction in which the value of the function increases the fastest, the idea of GD is using the negative gradient to find the optimal solution. For a differentiable function $F(\mathbf{W})$ in $\mathbf{W}^t$, the update rule is

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \beta \nabla F(\mathbf{W}^t), \tag{2.10}$$

where $t$ is the number of iteration, $\nabla F(\mathbf{W}^t)$ is the gradient of function $F(\mathbf{W})$ at $\mathbf{W}^t$, and the parameter $\beta > 0$ is called the learning rate. For a given initial value, we keep updating it by using Eq. (2.10) until the stopping condition is satisfied, then we can get the optimal solution $\mathbf{W}^*$.

## 2.4 Frobenius norm regularization

The Frobenius norm regularization is a widely used method to prevent the over-fitting problem. Considering a differentiable loss function $L(\mathbf{W})$ with the Frobenius norm regularization, then we have

$$\min_{\mathbf{W}} L(\mathbf{W}) + \frac{\alpha}{2} \|\mathbf{W}\|_F^2. \tag{2.11}$$

If we use the gradient descent to solve this problem, then for an element $W_{ij}^t$ in $\mathbf{W}^t$, the update is

$$\begin{aligned} W_{ij}^{t+1} &= W_{ij}^t - \beta\Big(\nabla L(W_{ij}^t) + \alpha W_{ij}^t\Big) \\ &= \big(1 - \alpha\beta\big) W_{ij}^t - \beta \nabla L(W_{ij}^t). \end{aligned} \tag{2.12}$$

It is considered as the weight decay. We can control the effect of the regularization by adjusting the regularization parameter $\alpha$. Without the Frobenius norm regularization, the update is

$$W_{ij}^{t+1} = W_{ij}^t - \beta \nabla L(W_{ij}^t). \tag{2.13}$$

Comparing Eqs. (2.12) and (2.13) , the value of the weight will be smaller in the Frobenius norm regularization, which can reduce the complexity of the model and make the model more robust (theory of Ockham's razor), and it is also proved in practice.

## 2.5 Back propagation neural network

Neural network is widely used in the information technology (IT) industry. It is one of the most popular and widely used models in classification task such as image recognition. In practical applications, most of the neural network models are multi nodes feed forward structure.

### 2.5.1 Forward propagation

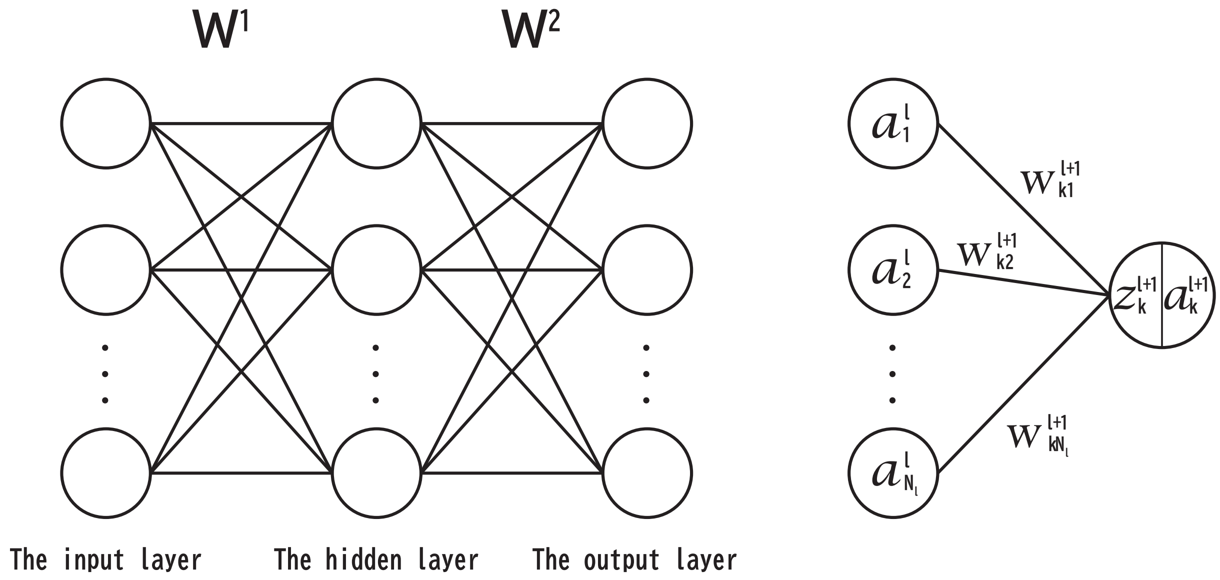


Figure 2.3: A two layer neural network

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ represent the data matrix with $n$ samples of $d$ features, and the label matrix $\mathbf{Y} \in \mathbf{R}^{n\times c}$ is represented by the one-hot coding. We use $l = 0, \ldots, L$ to denote the layer index of the network and $N_l$ denotes the number of neurons in the *l-th* layers. According to the feature dimension and the number of classes, we know that the input of the network is a $d$-dimensional vector and the output of the network is a $c$-dimensional vector. Let $W^l_{jk}$ denotes the weight of the *k-th* neuron in the *(l-1)-th* layer to the *j-th* neuron in the *l-th* layer, $b^l_j$ denotes the bias of the *j-th* neuron in the *l-th* layer, $z^l_j$ be the linear result of the *j-th* neuron in the *l-th* layer, and $a^l_j$ be the output of the *j-th* neuron in the *l-th* layer. In the forward propagation, neurons deliver data from the former layer to the next layer. Considering the *j-th* neuron in the *l-th* layer, we have

$$z^l_j = \sum_k W^l_{jk} a^{l-1}_k + b^l_j, \tag{2.14}$$

and

$$a^l_j = \sigma^l(z^l_j), \tag{2.15}$$

where $\sigma^l(\cdot)$ is the element-wise activation function of the *l-th* layer. We convert it into a matrix form, we have

$$\mathbf{z}^l = \mathbf{W}^l \mathbf{a}^{l-1} + \mathbf{b}^l, \tag{2.16}$$

and

$$\mathbf{a}^l = \sigma^l(\mathbf{z}^l), \tag{2.17}$$

where $\mathbf{W}^l$ is the weight matrix of the *l-th* layer. $\mathbf{a}^l$ and $\mathbf{a}^{l-1}$ are the output vectors of the *l-th* layer and the *(l-1)-th* layer respectively, $\mathbf{z}^l$ and $\mathbf{b}^l$ are the linear result vector and the bias vector of the *l-th* layer respectively. The output of the network is $\mathbf{a}^L \in \mathbf{R}^c$. Figure 2.3 shows a neural network with one hidden layer (a two layer neural network) and the detail of a neuron in the network.

### 2.5.2 Loss function and activation function

A loss function is used to measure how well the model fits the training sample to its label. In machine learning, the choice of the loss function should consider the type of the optimization problem. There are three popular loss functions for classification tasks; the multi-class hinge loss, the logistic loss, and the cross entropy loss.

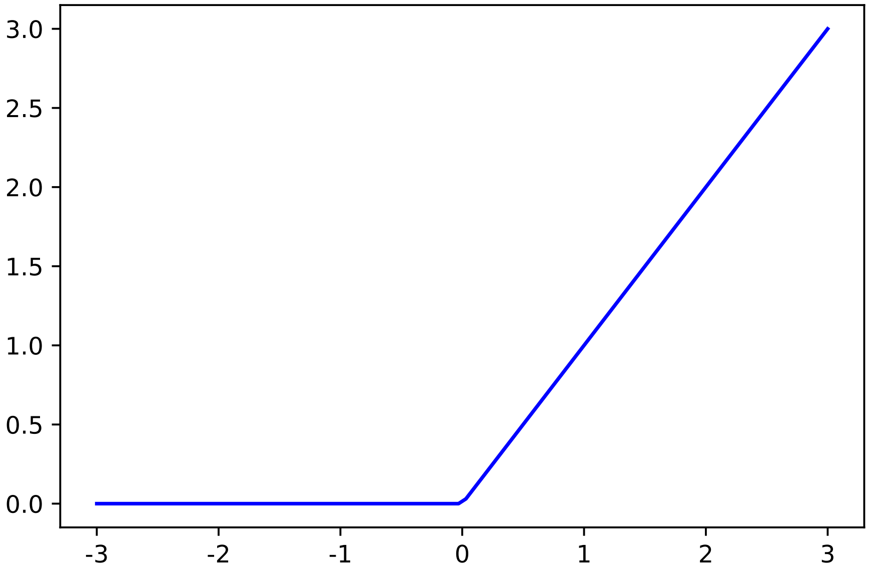


Figure 2.4: ReLu

We consider a multi class classification task. As we mentioned before, in one-hot coding, if a sample $\mathbf{x}$ belongs to the *i-th* class, then the *i-th* element of output $\mathbf{a}^L$ should be one and the rest of the elements are all zero. We choose the cross entropy loss as the loss function and the softmax as the activation function of the output layer. As for reason, using the cross entropy loss combined with the softmax can help to process data easily. Besides, we use the rectified linear unit (ReLU) as the activation function in the rest of layers. The rectified linear unit is defined as

$$\sigma(z) = \max(0, z), \tag{2.18}$$

and Figure 2.4 is a graph of the rectified linear unit. The softmax is defined as

$$y_j = \frac{e^{a_j}}{\sum_{j=1}^{c} e^{a_j}}, \tag{2.19}$$

where $a_j$ is the *j-th* element of $\mathbf{a}^L$, $\mathbf{y} = [y_1, \ldots, y_c]$. As $\sum_{j=1}^{c} y_j = 1$, the value of $y_j$ is also considered as the probability that the input sample belongs to the *j-th* class. If the *j-th* element of $\mathbf{y}$ has the biggest probability, then for the input sample $\mathbf{x}_i$, it will be classified to the *j-th* class. As $\mathbf{y}$ is considered as the classification prediction of $\mathbf{x}_i$, and the true label of $\mathbf{x}_i$ is $\mathbf{y}_i = [Y_{i1}, \ldots, Y_{ic}]$, the cross entropy loss is

$$-\sum_{j=1}^{c} Y_{ij} \log(y_j). \tag{2.20}$$

### 2.5.3 Back propagation

The weight which minimizes the error of the network can be obtained by the back propagation algorithm. In detail, suppose one sample input case, the loss function of a network without regularization is $L(\mathbf{W})$. For $W^l_{jk}$ in $\mathbf{W}^l$ and $b^l_j$ in $\mathbf{b}^l$, from Eq. (2.14), we have

$$\frac{\partial L(\mathbf{W})}{\partial W^l_{jk}} = \frac{\partial L(\mathbf{W})}{\partial z^l_j}\frac{\partial z^l_j}{\partial W^l_{jk}} = \frac{\partial L(\mathbf{W})}{\partial z^l_j} a^{l-1}_k, \tag{2.21}$$

and

$$\frac{\partial L(\mathbf{W})}{\partial b^l_j} = \frac{\partial L(\mathbf{W})}{\partial z^l_j}\frac{\partial z^l_j}{\partial b^l_j} = \frac{\partial L(\mathbf{W})}{\partial z^l_j}. \tag{2.22}$$

From Eqs. (2.21) and (2.22), we know that in the same layer, if we get $\frac{\partial L(\mathbf{W})}{\partial z^l_j}$, then $\frac{\partial L(\mathbf{W})}{\partial W^l_{jk}}$ and $\frac{\partial L(\mathbf{W})}{\partial b^l_j}$ are given, thus the key is how to get $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^l}$. With the chain rule, once we get the relation between $\frac{\partial L(\mathbf{W})}{\mathbf{z}^{l+1}}$ and $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^l}$, the problem is solved. With Eq. (2.16), we have

$$\mathbf{z}^{l+1} = \mathbf{W}^{l+1}\sigma(\mathbf{z}^l) + \mathbf{b}^{l+1}. \tag{2.23}$$

For $z^l_j$, using the chain rule, we have

$$\frac{\partial L(\mathbf{W})}{\partial z^l_j} = \sum_k \frac{\partial L(\mathbf{W})}{\partial z^{l+1}_k}\frac{\partial z^{l+1}_k}{\partial z^l_j}. \tag{2.24}$$

For $z^{l+1}_k$ we have

$$z^{l+1}_k = \sum_{i=1}^{N_l} W^{l+1}_{ki}\sigma^l(z^l_i) + b^{l+1}_k, \tag{2.25}$$

and only $i = j$ is meaningful, thus

$$\frac{\partial z^{l+1}_k}{\partial z^l_j} = W^{l+1}_{kj}\sigma'^l(z^l_j). \tag{2.26}$$

With Eq. (2.26), we can rewrite Eq. (2.24) into

$$\frac{\partial L(\mathbf{W})}{\partial z_j^l} = \sum_k \frac{\partial L(\mathbf{W})}{\partial z_k^{l+1}} W_{kj}^{l+1} \sigma'^l(z_j^l), \tag{2.27}$$

consider a matrix form, then we have

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^l} = (\mathbf{W}^{l+1})^T \Big( \frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^{l+1}} \Big) \odot \sigma'^l(\mathbf{z}^l), \tag{2.28}$$

where the operation symbol $\odot$ represents the same position element multiplication between two terms that have the same size.

As Eq. (2.28) is the relation between $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^{l+1}}$ and $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^l}$, once we have the gradient of the last layer $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^L}$, we can get the gradient of weight matrix in all layers. $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^L}$ is given by

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^L} = \frac{\partial L(\mathbf{W})}{\partial \mathbf{a}^L} \odot {\sigma'}^L(\mathbf{z}^L). \tag{2.29}$$

Since we have the relation between $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^{l+1}}$ and $\frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^l}$, we can get the derivative of any weight and bias in any layer. For convenience, let $\boldsymbol{\delta}^l = \frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^l}$ and $\boldsymbol{\delta}^L = \frac{\partial L(\mathbf{W})}{\partial \mathbf{z}^L}$. As a summary, for $W_{jk}^l$ and $b_j^l$, we have

$$\frac{\partial L(\mathbf{W})}{\partial W_{jk}^l} = \delta_j^l a_k^{l-1}, \tag{2.30}$$

and

$$\frac{\partial L(\mathbf{W})}{\partial b_j^l} = \delta_j^l, \tag{2.31}$$

where $\delta_j^l$ is the *j-th* element of $\boldsymbol{\delta}^l$. We calculate $\boldsymbol{\delta}^L$, and get the gradient of the weight matrix in the output layer. Then using the relation between $\boldsymbol{\delta}^{l+1}$ and $\boldsymbol{\delta}^{l}$, from the output layer to the input layer, all the gradient of the weight matrix can be obtained. Therefore, it is called the back propagation process.

# Chapter 3

# Proposed Method

## 3.1 Model structure of proposed method

As our purpose is selecting relevant features through a sparse learning method, we propose a sparse regularization method to the neural network. We propose the $\ell_{2,1-2}$ norm as the regularization term in the input layer of the network because we can get a sparser solution.

Besides, in order to deal with the over-fitting problem, we consider using a regularization in the rest of the layers. As the Frobenius norm regularization is widely used to prevent the over-fitting problem and exhibited good results, we use it in the rest of the layers. Thus, the structure of our model is; the input layer with the $\ell_{2,1-2}$ norm regularization and the rest of layers with the Frobenius norm regularization.

## 3.2 Minimization problem

As a sparse feature selection, the model is Eq. (2.1). The optimization problem of the proposed method is

$$\begin{aligned}
&\min_{\mathbf{W}} L(\mathbf{W}) + \alpha\Big\{\|\mathbf{W}^1\|_{2,1-2} + \frac{1}{2}\sum_{l=2}^{L}\|\mathbf{W}^l\|_F^2\Big\} \\
&= \min_{\mathbf{W}} L(\mathbf{W}) + \alpha\Big\{\|\mathbf{W}^1\|_{2,1} - \|\mathbf{W}^1\|_F + \frac{1}{2}\sum_{l=2}^{L}\|\mathbf{W}^l\|_F^2\Big\},
\end{aligned} \tag{3.1}$$

where $L(\mathbf{W})$ represents the cross entropy loss, $\|\mathbf{W}^1\|_{2,1} - \|\mathbf{W}^1\|_F$ is the $\ell_{2,1-2}$ norm regularization of the input layer, and $\sum_{l=2}^{L}\|\mathbf{W}^l\|_F^2$ represents the Frobenius norm regularization of the other layers.

## 3.3 Gradient descent for solving minimization problem

We use the gradient descent to update the weight of every layer. In the loss function part of (3.1), the update rule of $W_{jk}^l$ is the same as Eq. (2.30). It is

$$\frac{\partial L(\mathbf{W})}{\partial W_{jk}^l} = \delta_j^l a_k^{l-1}, \quad l = 1, \dots, L. \tag{3.2}$$

In the regularization part of (3.1), for $W^l_{jk}$ of the input layer, with Eq. (2.9), we can easily have

$$\frac{\partial}{\partial W^l_{jk}}(\|\mathbf{W}^l\|_{2,1-2}) = \begin{cases} \frac{W^l_{jk}}{\|\mathbf{w}^l_j\|_2} - \frac{W^l_{jk}}{\|\mathbf{W}^l\|_F}, \quad l = 1,, & \text{if } W^l_{jk} \neq 0 \\ 0, & \text{if } W^l_{jk} = 0, \end{cases} \tag{3.3}$$

where $\mathbf{w}^l_j$ is the *j-th* row vector of the weight matrix $\mathbf{W}^l$. Besides, for $W^l_{jk}$ of the others layers, we have

$$\frac{\partial(\frac{1}{2}\|\mathbf{W}^l\|^2_F)}{\partial W^l_{jk}} = W^l_{jk}, \quad l = 2, \dots, L. \tag{3.4}$$

Thus, for the optimization problem Eq. (3.1), the update of the weight in the input layer is

$$W^{l,t+1}_{jk} = W^{l,t}_{jk} - \beta(\theta^t_1 + \alpha\theta^t_2), \quad l = 1, \tag{3.5}$$

where $\theta^t_1$ represents $\delta^{l,t}_j a^{l-1,t}_k$ and $\theta^t_2$ represents $\frac{W^{l,t}_{jk}}{\|\mathbf{w}^{l,t}_j\|_2} - \frac{W^{l,t}_{jk}}{\|\mathbf{W}^{l,t}\|_F}$. For the others layers, the update of the weight is

$$W^{l,t+1}_{jk} = W^{l,t}_{jk} - \beta(\theta^t_1 + \alpha\theta^t_3), \quad l = 2, \dots, L, \tag{3.6}$$

where $\theta^t_3$ represents $W^{l,t}_{jk}$, $t$ is the number of iteration. As for the update of bias $b^l_j$ of all layers, the update rule is

$$b^{l,t+1}_j = b^{l,t}_j - \beta\delta^{l,t}_j, \quad l = 1, \dots, L. \tag{3.7}$$

# Chapter 4

# Experiments and Results

In this chapter, we verified the effectiveness of the proposed method through conducting experiments on image classification task, and all the results were base on five real word datasets. Besides, we also compared the results with other methods.

## 4.1 Experiment

### 4.1.1 Implementation

All the programs of our experiments were implemented by Python 3.7, and the program ran on PC with i5-7200U CPU, 8.00 GB memory.

We used two types of neural network models for the classification task. One is a three hidden layer structure with 100, 50, and 50 units separately, another is one hidden layer structure with 120 units. The number of the hidden layer and unit are decided by the samples of the dataset and the feature dimension of the sample[1]. Also, we trained the data by mini-batch[2].

### 4.1.2 Dataset

We used five popular open datasets to conduct our experiments. Among those datasets, including three small datasets and two large datasets. The information of the datasets we used on experiments are shown in Table 4.1.

Table 4.1: Open dataset information

| Dataset information | | | |
|---|---|---|---|
| Dataset | Features | Samples | Classes |
| COIL20 | 1024 | 1440 | 20 |
| Mnist | 784 | 70000 | 10 |
| ORL | 1024 | 400 | 40 |
| USPS | 256 | 9298 | 10 |
| Yale | 1024 | 165 | 15 |

And all the open datasets were downloaded from the internet. Among five datasets used in our experiment, Mnist[3] and USPS[4] are handwritten digits datasets, ORL[5] and Yale[6] are face recognition datasets, and COIL20[7] is a image recognition dataset.

[1] Mnist and USPS were trained by the three hidden layer network and the rest of the datasets were trained by the one hidden layer network.

[2] For USPS, COIL20 and Yale, the batchsize is 32, for Mnist and ORL, the batch size is 64.

[3] From http://yann.lecun.com/exdb/mnist/

[4] From http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

[5] From http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

[6] From http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

[7] From https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

## 4.2 Details of the experiment

### 4.2.1 Parameters

In the experiment of the proposed method, we need to decide two parameters; the regularization parameter $\alpha$ and the learning rate $\beta$.

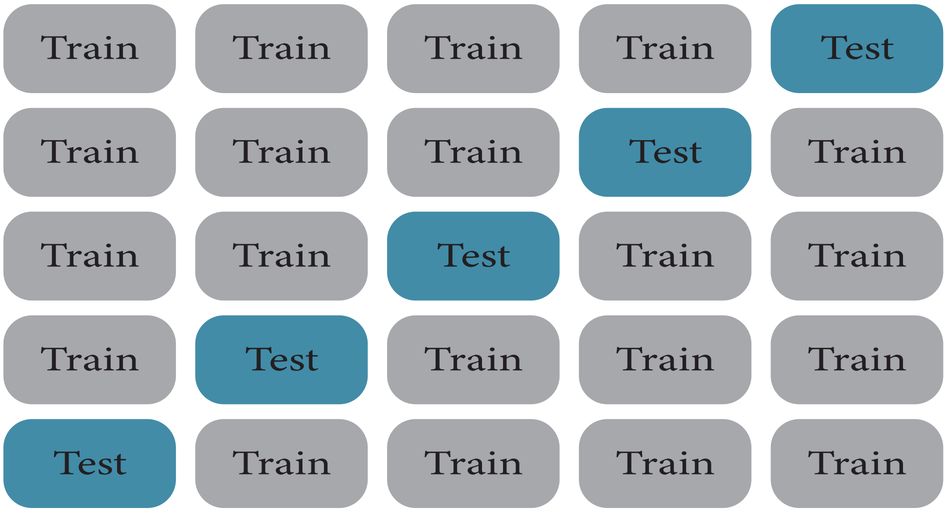
We set the learning rate $\beta$ into 0.1, and the regularization parameter $\alpha$ is tuned from the range of $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$. We fixed the learning rate $\beta$ and used all the regularization parameter candidates to run a five fold cross validation. Results were base on six models corresponding to all the regularization parameter candidates, and the best regularization parameter $\alpha$ was determined by the best model. Besides, We also made minor adjustments[8] to the above parameters to see if there were better results.

### 4.2.2 Cross validation and student's t-test

We used classification accuracy to evaluate the performance of all methods. We used the five fold cross validation to run the experiment because we considered that in a small sample dataset, a large fold of the cross validation might lead to a large fluctuation in the results. Figure 4.1 shows the partition of dataset with five fold cross validation.

As the proposed method is considered to exhibit better classification performance than the Frobenius norm regularization, we also did the paired samples student's t-test between the classification accuracy of the Frobenius norm regularization and that of the

[8] For example, in the experiment of dataset Mnist, if $\alpha = 0.0005$ showed the best result from all parameter candidates, then we plus and minus 0.00005 (10% of the best parameter) to get two new parameter 0.00045 and 0.00055, and used two new parameters to run the program. Repeated it if there were better result.



| | | | | |
|---|---|---|---|---|
| Train | Train | Train | Train | Test |
| Train | Train | Train | Test | Train |
| Train | Train | Test | Train | Train |
| Train | Test | Train | Train | Train |
| Test | Train | Train | Train | Train |

Figure 4.1: Five fold cross validation

proposed method to verify the difference. We denoted the null hypothesis as $H_0$; the average classification accuracy of the Frobenius norm regularization is higher than the average classification accuracy of the proposed method, and denoted the alternative hypothesis as $H_1$; the average classification accuracy of the proposed method is higher than the average classification accuracy of the Frobenius norm regularization. We used 0.05 as the statistical significance, and the $t$ value is given by

$$t = \frac{\bar{d}}{\sqrt{s^2/n}}, \tag{4.1}$$

where $\bar{d}$ is the mean of the difference of two groups, $n$ is five, the number of folds in cross validation, $s^2$ is the unbiased estimator of the population variance, it is given by

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2, \tag{4.2}$$

and $d_i$ is the difference of the *i-th* fold result.

## 4.3 Experimental results

Table 4.2 lists the average classification accuracy and the standard deviation of the five fold cross validation of all datasets. There are four cases: 1)no regularization; 2)the Frobenius norm; 3)$\ell_{2,1}$ norm; and 4)the proposed method, the $\ell_{2,1-2}$ norm.

No regularization means that trained data and obtained model without using any regularization term. The Frobenius norm means that using the Frobenius norm as the regularization term in all the layers of the network. As the $\ell_{2,1}$ norm regularization and the $\ell_{2,1-2}$ norm regularization induce a sparse solution, they were used only in the input layer, and the rest of the layers used the Frobenius norm regularization to enhance the robustness of the model.

Table 4.2: Classification accuracy of four methods

| Average classification accuracy [%] | | | | |
|---|---|---|---|---|
| Dataset | no regularization | the Frobenius norm | the $\ell_{2,1}$ norm | the $\ell_{2,1-2}$ norm |
| COIL20 | 90.44±5.038 | 91.18±4.784 | 91.32±4.691 | 91.94±4.857 |
| Mnist | 96.82±0.185 | 97.06±0.314 | 97.59±0.145 | 97.96±0.110 |
| ORL | 93.75±4.031 | 94.25±3.921 | 95.00±3.536 | 95.25±3.391 |
| USPS | 95.83±1.554 | 96.14±1.810 | 96.66±1.925 | 96.81±1.547 |
| Yale | 80.13±0.976 | 81.21±1.024 | 81.79±1.067 | 84.56±2.160 |

As the results shown in Table 4.2, no regularization case showed the lowest average classification accuracy because it took no measure to deal with the over-fitting problem. The Frobenius norm case showed higher classification accuracy because it can prevent the over-fitting problem which might be caused by noise. As the $\ell_{2,1}$ norm and the $\ell_{2,1-2}$ norm were used to select relevant features, these two cases showed higher classification accuracy than the Frobenius norm case. The classification accuracy of the $\ell_{2,1-2}$ norm was higher than the $\ell_{2,1}$ norm, and we consider the $\ell_{2,1-2}$ norm is more effective in feature selection. Figure 4.2 is the comparison of all four methods.

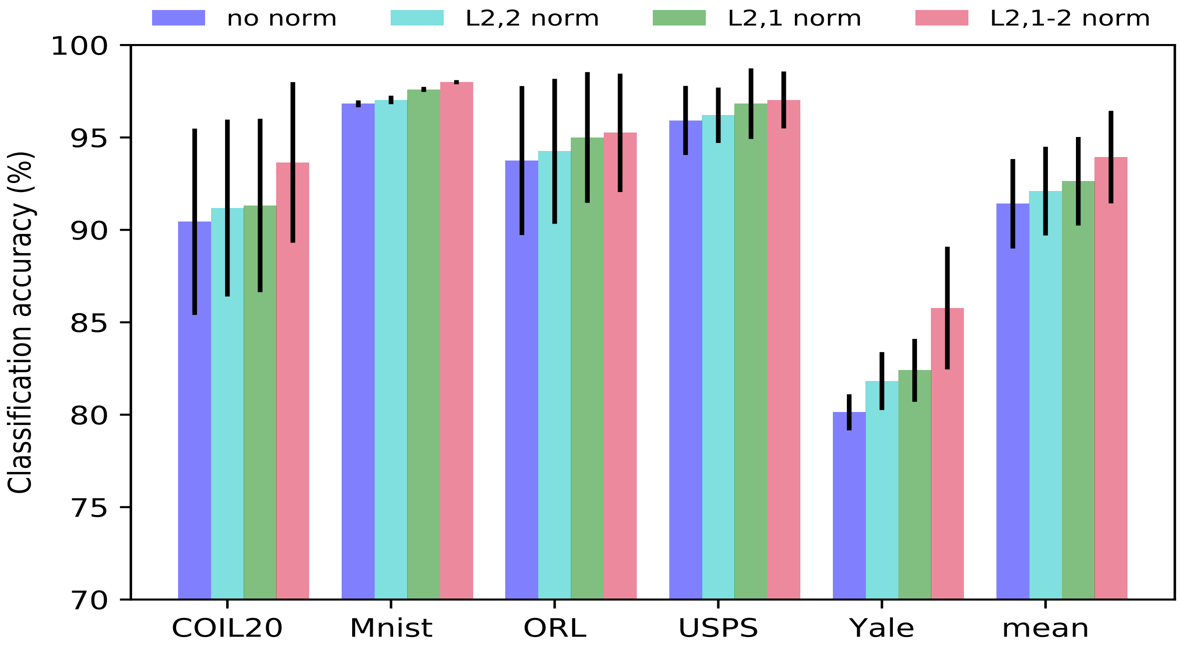


Figure 4.2: Results of five open datasets

We also verified the sparsity of the $\ell_{2,1}$ norm and the the $\ell_{2,1-2}$ norm. As for detail, we used the row sparse matrix solution to conducted the experiment, and also calculated the sparse rate of both the $\ell_{2,1}$ norm and the the $\ell_{2,1-2}$ norm. The results are shown in Table 4.3 and Table 4.4. Figure 4.3 is the comparison of sparse rate between the $\ell_{2,1}$ norm and the $\ell_{2,1-2}$ norm of all the datasets. Sparse rate is the ratio of zero row vectors and the number of the row vector in the weight matrix. A higher sparse rate means less noise and less features are selected, which can lead to a better regularization effect. From Figure 4.3, we know that the proposed method showed a higher sparser rate.

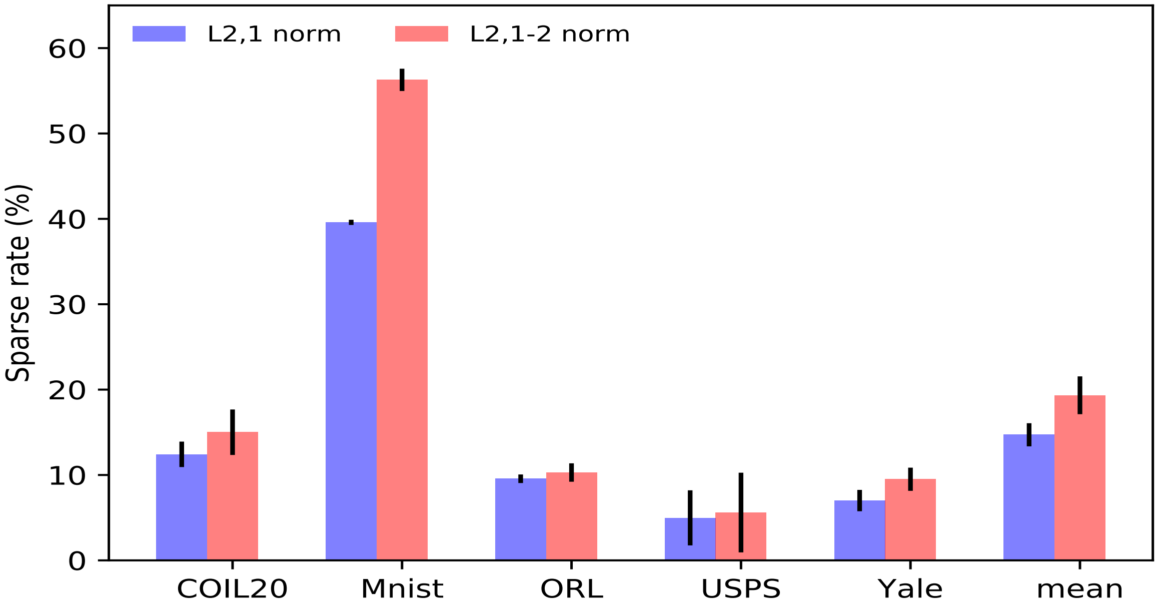


Figure 4.3: Sparse rate between the $\ell_{2,1}$ norm and the $\ell_{2,1-2}$ norm

Table 4.3: Results of sparse rate

| Average sparse rate [%] | | |
|---|---|---|
| Dataset | $\ell_{2,1}$ norm | $\ell_{2,1-2}$ norm |
| COIL20 | 12.43±1.477 | 16.51±1.040 |
| Mnist | 39.67±0.628 | 53.94±3.337 |
| ORL | 3.083±2.569 | 3.200±2.867 |
| USPS | 5.292±2.867 | 6.148±2.659 |
| Yale | 8.449±0.864 | 9.112±1.223 |

Table 4.4: Results of sparse classification accuracy

| Average sparse classification accuracy [%] | | |
|---|---|---|
| Dataset | $\ell_{2,1}$ norm | $\ell_{2,1-2}$ norm |
| COIL20 | 90.63±5.272 | 91.39±5.040 |
| Mnist | 97.24±0.142 | 97.74±0.159 |
| ORL | 94.75±3.687 | 94.75±3.687 |
| USPS | 96.38±1.810 | 96.52±1.962 |
| Yale | 81.35±4.876 | 84.19±4.693 |

Table 4.4 is the sparse classification accuracy of the $\ell_{2,1}$ norm and the the $\ell_{2,1-2}$ norm. Different from classification accuracy, sparse classification accuracy means that we used the sparse solution[9] to conduct experiments and obtain the corresponding classification accuracy.

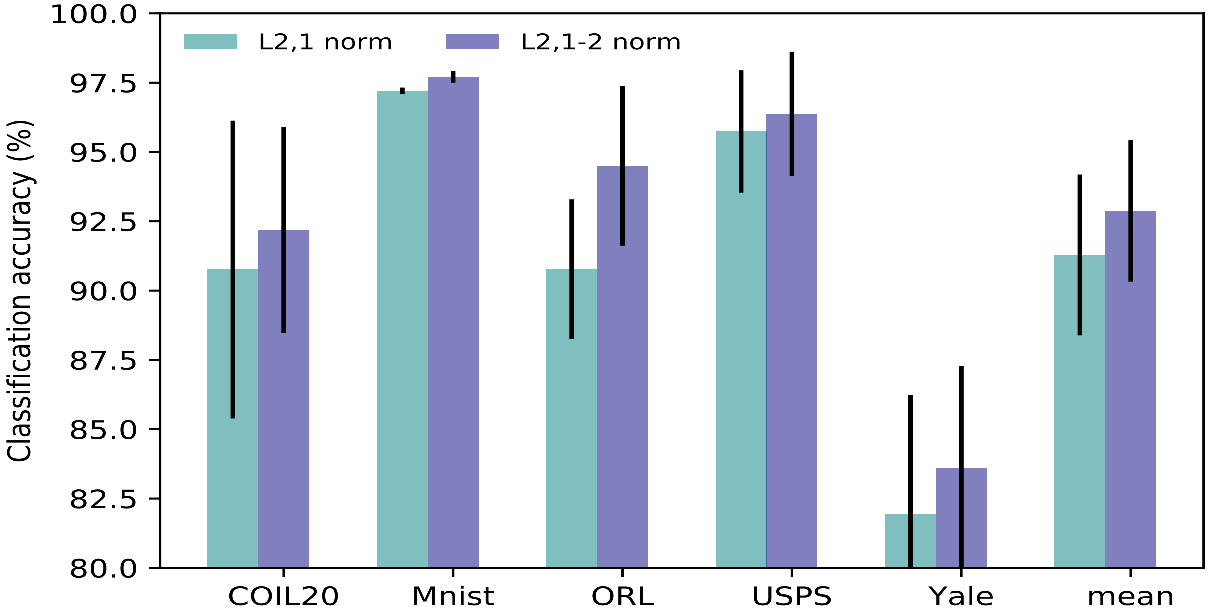


Figure 4.4: Sparse classification accuracy between the $\ell_{2,1}$ norm and the $\ell_{2,1-2}$ norm

From Figure 4.4 the $\ell_{2,1-2}$ norm case showed a higher sparse classification accuracy. Comparing with the $\ell_{2,1}$ norm regularization, the proposed method showed higher sparse rate and higher sparse classification accuracy. It means that comparing with the $\ell_{2,1}$ norm regularization, the proposed method is a more effective feature selection method.

Table 4.5 shows the results of the paired samples student's t-test. In the student's t-test, we set 0.05 as the statistical significance. As a result, hypothesis $H_0$ is rejected

[9] For the row vector $\mathbf{w}_i, i = 1, \ldots, d$ in the optimal solution $\mathbf{W}^* \in \mathbf{R}^{d\times c}$, sparse solution was obtained by setting $\mathbf{w}_i$ into zero. For example, if $\|\mathbf{w}_i\|_2/c$ is less than a very small value such like 0.0001, then $\mathbf{w}_i$ will be set to zero.

Table 4.5: Results of student's t-test

| Student's t-test between the $\ell_{2,1-2}$ norm and the Frobenius norm | | | | |
|---|---|---|---|---|
| Dataset | $t$-value | $t_{0.05}(4)$ | measure | conclusion |
| COIL20 | 5.412 | 2.132 | $5.412 > 2.132$ | $H_1$ |
| Mnist | 14.563 | 2.132 | $14.563 > 2.132$ | $H_1$ |
| ORL | 1.372 | 2.132 | $1.372 < 2.132$ | $H_0$ |
| USPS | 5.416 | 2.132 | $5.416 > 2.132$ | $H_1$ |
| Yale | 3.225 | 2.132 | $3.225 > 2.132$ | $H_1$ |

while hypothesis $H_1$ is adopted on most of the datasets. It means that we can consider the $\ell_{2,1-2}$ norm regularization is better than the Frobenius norm regularization in classification accuracy.

# Chapter 5

# Conclusion

## 5.1 Conclusion

### 5.1.1 Discussion

From Figure 4.2, the proposed method showed the highest classification accuracy among all regularization methods. For the $\ell_{2,1}$ norm regularization which also induces a sparse solution, in addition to classification accuracy, we also compared the sparsity performance. We used the sparse rate and the sparse classification accuracy to evaluate their performance. Sparse classification accuracy is obtained by using the sparse solution, and the sparse rate is the ratio of zero row vectors in the weight matrix (sparse solution). As shown in Figures 4.3 and 4.4, the proposed method obtains a sparser solution and achieved a higher sparse classification accuracy on all datasets. Besides, as we consider the $\ell_{2,1-2}$ norm case was higher classification accuracy than the Frobenius norm case, we did the student's t-test with 0.05 as the statistical significance, and the result shows that the classification accuracy of the proposed method is significantly higher.

### 5.1.2 Conclusion

In this paper, motivated by the advantage of the $\ell_{2,1-2}$ norm, we proposed it as the regularization term in neural network to select features and build a more robust model. We compared the proposed method with the $\ell_{2,1}$ norm regularization and the Frobenius norm regularization. As the $\ell_{2,1}$ norm can also lead to a sparse solution, we evaluated their performance by comparing sparse classification accuracy and sparse rate. As a result, the proposed method can obtain a sparser solution and achieved higher sparse classification accuracy. Besides, combining the result of classification accuracy and the result of the student's t-test between the the proposed method and the Frobenius norm regularization, we can also draw a conclusion that the proposed method is higher classification accuracy than the Frobenius norm regularization.

## 5.2 Future work

The dropout tries to shut-down some neurons randomly in each layer of the network during the training process. Considering that the $\ell_{2,1-2}$ norm leads to a sparser solution during the optimization process, we apply the $\ell_{2,1-2}$ norm to all the layers of the network to "shut-down" some neurons discriminantly.

# References

[1] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. *MIT Press*, 2016.

[3] D. E. Rumelhart, G. E. Hinton, R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088): 533-536, 1986.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[5] K. Anders, and J. Hertz. A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, pp. 950-957, 1992.

[6] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu. Advancing feature selection research. *ASU Feature Selection Repository*, 1-28 , 2010.

[7] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection, in *Proc. 23rd AAAI Conf. Artif. Intell.*, pp. 671–676, 2008.

[8] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint $\ell_{2,1}$ -norms minimization, in *Proc. Advances in Neural Information Processing Systems*, pp. 1813–1821, 2010.

[9] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou. $\ell_{2,1}$ -norm regularized discriminative feature selection for unsupervised learning, in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, pp. 1589–1594, 2011.

[10] Shi, Y., Miao, J., Wang, Z., Zhang, P., Niu, L. Feature selection with $\ell_{2,1-2}$ regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10), 4967-4982, 2018.

# Acknowledgement

Here, I want to say thank you to my supervisor: Washizawa Yoshikazu associate professor. The successful completion of this thesis is based on his comments, suggestions and others support. Also, his professional knowledge and serious scientific attitude have a great impact on me, and I wish him all the best for the future !