Team 11 Final Project

MLP vs SVM from Project 6

Member

m5261108 Kazuki Fujita m5261110 Mizuki Goto

m5261132 Kaito Ogino m5261147 Haruki Maeda

Past Activities

We decided a leader for each Project to divide the amount of Tasks.
We did our self-assessment in Table 2.

Project number	Leader of the project	Report author
Project 1	Kazuki Fujita	Kazuki Fujita
Project 2	Kaito Ogino	Kaito Ogino
Project 3	Haruki Maeda	Haruki Maeda
Project 4	Mizuki Goto	Mizuki Goto (Bonus: Kazuki Fujita)
Project 5	Kazuki Fujita	Kazuki Fujita
Project 6	Kaito Ogino Haruki Maeda	Kaito Ogino Haruki Maeda

Table 1: Task assignment list

Member	Rate
m5261108 Kazuki Fujita	30%
m5261110 Mizuki Goto	10%
m5261108 Kaito Ogino	30%
m5261108 Haruki Maeda	30%

Table 2: Team contribution rate
(total: 100%)

Introduction: Support Vector Machine

- SVM(Support Vector Machine)
  - A supervised learning model with an algorithm for pattern classification [1]
  - It select a hyperplane so that the distance from it to the nearest data point on each side is maximized.

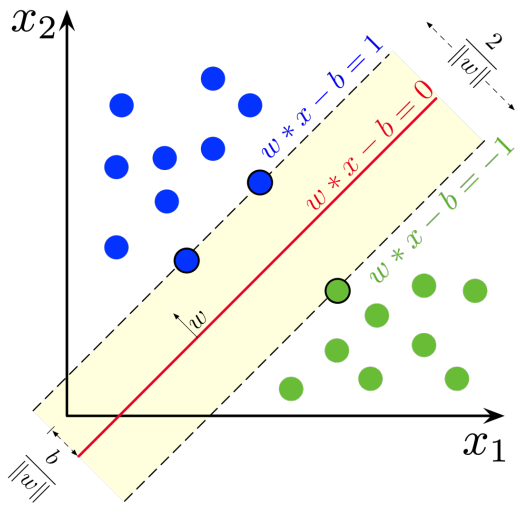

Figure 1: Maximum margin hyperplanes and margins for SVMs trained on a 2-class sample. [2]

[1] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20.3 (1995): 273-297.
[2] Wikipedia. "Support-vector machine" *https://commons.wikimedia.org/wiki/File:SVM_margin.png*

Introduction: Multilayer Perceptron

- Multilayer Perceptron (MLP)
  - A fully connected class of feedforward artificial neural network (ANN)
  - It consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
  - It utilizes a supervised learning technique called backpropagation for training.

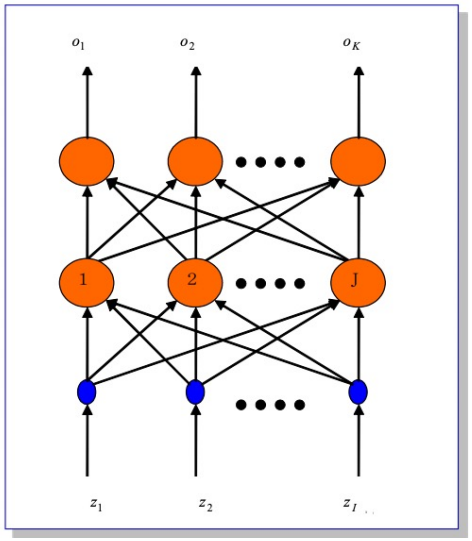

Figure 1: Network structure of MLP

Experiments

Procedure:

1. Compare MLP with SVMs
2. Change karnel function in SVM:
   - Poly(polynomial): $K(x, y) = (\boldsymbol{x} \cdot \boldsymbol{y} + c)^d$
   - RBF(radial basis function): $K(x, y) = e^{\frac{-\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}$
   - Sigmoid: $K(x, y) = \tanh(\kappa \mathbf{x} \cdot \boldsymbol{y} - \delta)$
   - Linear: $K(x, y) = \boldsymbol{x} \cdot \boldsymbol{y}$
3. Check the accuracy with following formula:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Data Sets

- From UCI machine learning repository:
  - Iris Data Set
  - Wine Data Set
  - Breast Cancer Wisconsin Data Set
  - Optical Recognition of Handwritten Digits Data Set

DataSet	#Classes	#Train	#Test	#Features
Iris	3	105	45	4
wine	3	124	54	13
cancer	2	398	171	30
degits	10	1257	540	64

Table 3: Number of classes, Number of instances in Train and Test split in each datasets

Tools

- Programing Language:
  - Python 3.9.9
- Scikit-learn:
  - Python package for machine learning
  - This pachage produces modules to establish and to evaluate SVM and MLP.

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_breast_cancer
from sklearn.datasets import load_iris
from sklearn.datasets import load_wine
from sklearn.datasets import load_digits
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.pipeline import Pipeline
```

Figure 1: Modules of sklearn which were used in the experiment

Hyper Parameter in MLP

- Setting Hyper parameters:
  - SVM: default setting of scikit-learn
  - MLP: hidden layer size, other is default setting of scikit-learn
- Evaluate hidden layer size:
  - Parameter set: [3, 5, 7, 10, 100]

DataSet	size 3	size 5	size 7	size 10	size 100
Iris	0.84	0.96	0.96	0.98	0.96
wine	0.98	1	1	0.98	0.98
cancer	0.97	0.95	0.96	0.95	0.96
digits	0.84	0.9	0.94	0.98	0.98

Table 4: Evaluate hidden layer size

  - The average of accuracy if hidden layer size set 10 was the best score.
  - So, hidden layer size in experiment was defined 10.

Results

DataSet	Algorithm	Accuracy	time (ms)
Iris	Poly	0.98	2.7483
	sigmoid	0.93	2.7284
	Linear	0.98	2.7382
	RBF	0.96	2.8370
	MLP	0.96	293.1680

DataSet	Algorithm	Accuracy	time (ms)
wine	Poly	0.94	2.7650
	sigmoid	0.98	2.6602
	Linear	0.98	2.5990
	RBF	0.98	2.7600
	MLP	1.00	204.0756

DataSet	Algorithm	Accuracy	time (ms)
cancer	Poly	0.90	4.8982
	sigmoid	0.94	4.4854
	Linear	0.95	4.1840
	RBF	0.97	4.9770
	MLP	0.95	238.4200

DataSet	Algorithm	Accuracy	time (ms)
digits	Poly	0.95	51.0530
	sigmoid	0.95	27.8043
	Linear	0.98	20.2355
	RBF	0.98	44.5010
	MLP	0.96	1108.4915

Result 1: Results for each dataset

Discussion

- MLP takes time more than SVMs. However, the accuracy of SVMs are little higher than MLP's.
- We consider if we implement with more neurons and layers, we will get higher MLP's accuracy.

Number of hidden layer: 1 -> 4
Each hidden layer size: 10 -> 1000

```
'MLP':
    Pipeline([('scl',StandardScaler()),
              ('est',MLPClassifier(hidden_layer_sizes=(10,),
                                   max_iter=1000,
                                   random_state=1))])
```

```
'MLP':
    Pipeline([('scl',StandardScaler()),
              ('est',MLPClassifier(hidden_layer_sizes=(1000, 1000, 1000, 1000),
                                   max_iter=1000,
                                   random_state=1))])
```

Figure 1: Increase in hidden layers

Result of the changes

- The more neurons and layers we used, the higher accuracy we got.
- MLP takes huger time than before the changes, but there are no differences between RBF performance and MLP one.

	Accuracy			time(ms)		
DataSet	**RBF**	**MLP(Before the changes)**	**MLP(After the changes)**	**RBF**	**Before(MLP)**	**After(MLP)**
iris	0.96	0.96	0.96	2.837	293.168	5112.6446
wine	0.98	1	1	2.76	204.0756	1762.4261
cancer	0.97	0.95	0.96	4.977	238.42	6816.1918
digits	0.98	0.96	0.99	44.501	1108.4915	17596.4272

Result 2: Comparison before and after increasing the hidden layer

Conclusion

- SVM achieved similar performance with less time than MLP.
- It means if we train huge dataset,
- We couldn’t optimize hyperparameters (number of hidden layers and each hidden layer size) of MLP and SVM.
- If we do that, we will be able to get better performance (accuracy and run time).