Report on Homework 3 – SVM vs. Neural Networks

CS420, Machine Learning, Shikui Tu, Summer 2018

Zelin Ye 515030910468

1 Introduction

In machine learning, SVM (Support Vector Machine) is a commonly used classfication method due to its high efficiency and accuracy. Recent years, the neural network has been attracting more and more attention, and also used to solve classification problems. In this homework, I would investigate the performances of SVM and neural network (e.g. MLP) on some classification datasets under different experimental settings (e.g. pass).

2 Methodology

In this section, I would introduce the datasets and models in my experiments.

2.1 Datasets

In my experiments, I use two datasets that are from **LIBSVM Data** [1] to investigate the performances of SVM and neural network. One is **splice**, a binary classification dataset with 60 features, 1000 training samples and 2175 testing samples. Another is called **satimage**, which is for multi-class classification and has 36 features, 6 classes, 3104 training samples and 2000 testing samples.

Additionally, I choose a dataset called **ConvexNonConvex** from LISA [2] to conduct comparison between SVM and deep learning algorithm benchmarks. It is a binary classification dataset that contains 784 features, 8000 training samples and 50000 testing samples.

More details about these datasets can refer to Appendix A.1.

2.2 Models

For neural network, considering the complexity of features and scale of samples, I choose MLP (Multilayer Perceptron) instead of popular DNN or CNN.

In experiments, I would investigate the performances of MLP under different architectures or parameter settings (e.g. number of hidden layers or hidden neurons).

3 Experiments and Results

3.1 Preprocess

Most datasets are likely to have missing data, and those in LIBSVM Data are no exception. Therefore, I first make up for the omission in the datasets, replacing empty data with corresponding mean values. Afterwards, I convert the labels from numbers to one-hot vectors for the calculation of loss.

3.2 SVM

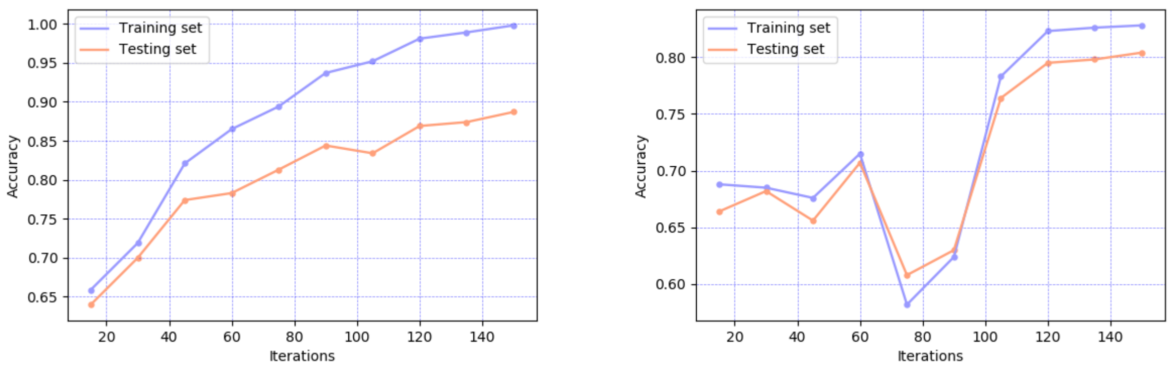
In this section, I would show the performances of SVM in both *splice* and *satimage* datasets under different configurations (e.g. pass). Note that the two datasets have different problem settings, sample sizes and etc. I would thus conduct some pertinent discussion based on each experiment result.

Without special explanation, except for the target parameter, the experiments are based on default parameters of *sklearn* [3] (a Python package for machine learning), which can refer to Tab. 1.

Table 1: Some important default parameters of the experiments

Name	Meaning	Value
C	penalty parameter of the error term	1.0
degree	degree of the polynomial kernel function	3
gamma	kernel coefficient for 'rbf', 'poly' and 'sigmoid'	1/n_features
coef0	independent term in kernel function	0.0
shrinking	whether to use the shrinking heuristic	True
tol	tolerance for stopping criterion	1e-3
decision_function_shape	one-vs-rest or one-vs-one	ovr
hidden_layer_sizes	the size of hidden layers	(100,)
activation	activation function for the hidden layer	relu
solver	the solver for weight optimization	adam
alpha	L2 penalty (regularization term) parameter	0.0001
batch_size	size of minibatches for stochastic optimizers	min(200, n_samples)
learning_rate_init	the initial learning rate used	0.001
power_t	the exponent for inverse scaling learning rate	0.5
tol	tolerance for the optimization	1e-4

With these default parameters, I train a baseline model as a reference (Fig. 1). According to the normal principle, I should measure the performance with f1-score, but I tend to make the results more intuitive and I use the **accuracy** of classification as the criterion.

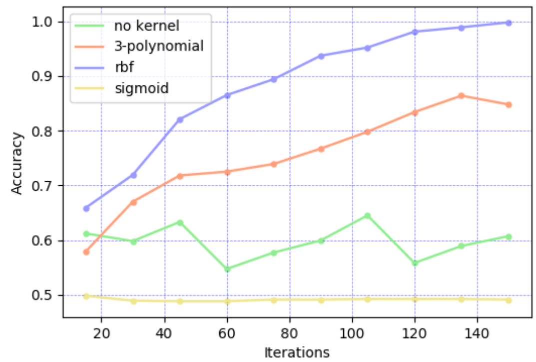


(a) Splice (b) Satimage

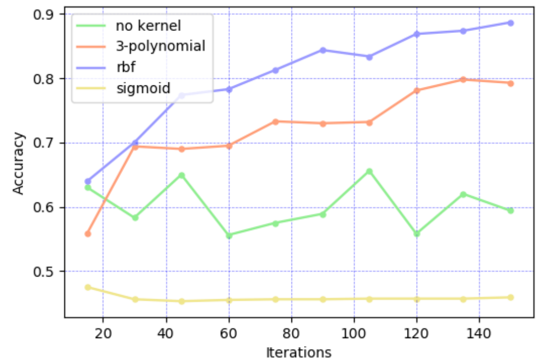
Figure 1: The baseline of SVM.

3.2.1 Kernel

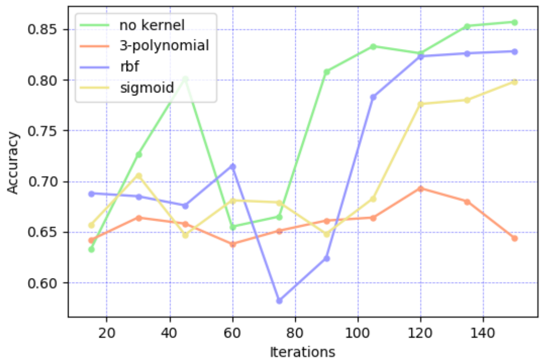
The ability of SVM might be constrainted when dealing with non-linear features, where kernel trick could resolve this issue. Nevertheless, a successful choice of kernel is still based on experience or luck. I apply several kernels to SVM and get the following results (Fig. 2).



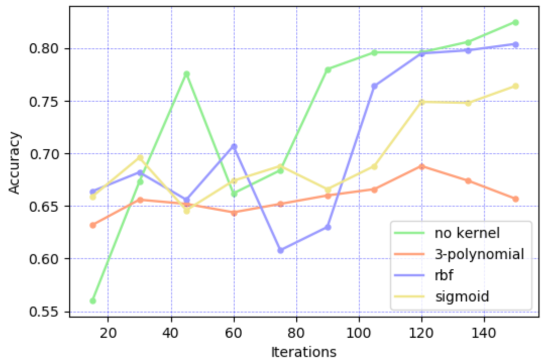
(a) Splice-Training Set



(b) Splice-Tetsing Set



(c) Satimage-Training Set



(d) Satimage-Testing Set

Figure 2: The performances of SVM under different kernels.

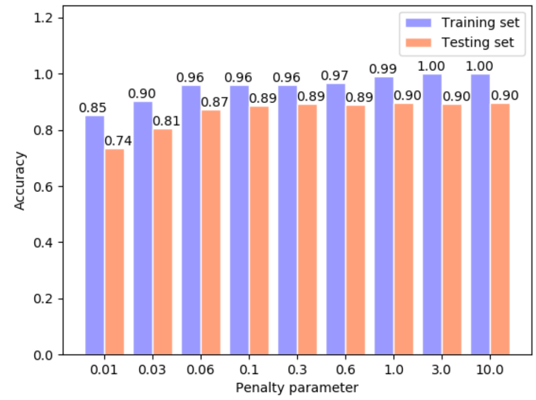
It is apparent that the effect of these kernels differ greatly. Rbf (Radial Basis Function) kernel is best for *splice* while no kernel function is the best choice for *satimage*.

Additionally, the selection of specific parameters of a kernel would also affect the results. I test polynomial kernel with different numbers of power, the test error can refer to Tab. 2.

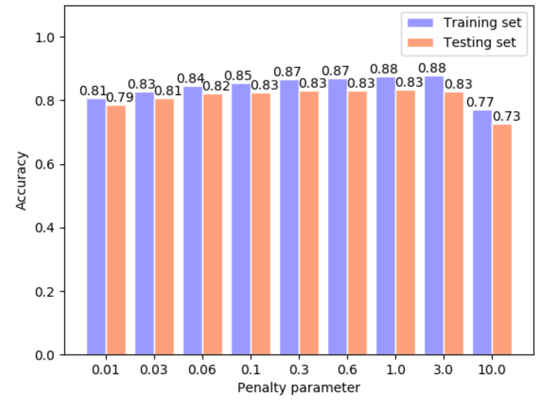
Table 2: The performances of polynomial kernel under different powers

Number of Power	Testing Error(Splice)	Testing Error(Satimage.scale)
2	0.786	0.635
3	0.857	0.666
4	0.865	0.505
5	0.871	0.554
6	0.880	0.488
7	0.874	0.512
8	0.864	0.477
9	0.867	0.528
10	0.851	0.403

3.2.2 Penalty Parameter



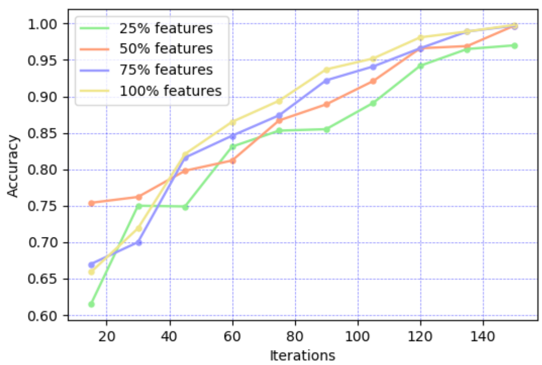
(a) Splice



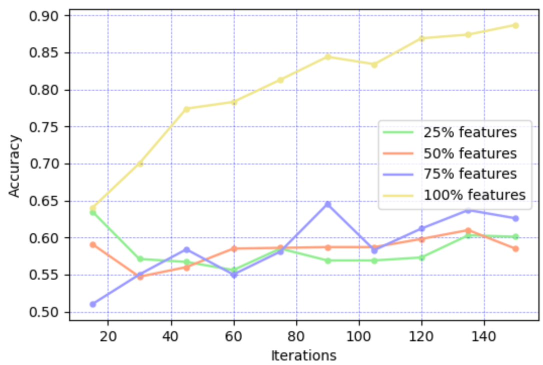
(b) Satimage

Figure 3: The performances of SVM under different penalty parameter C.

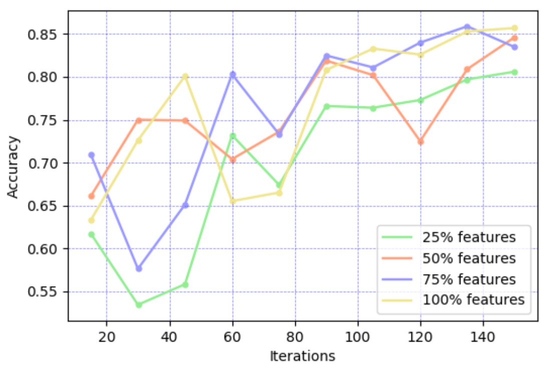
3.2.3 Dimension of Features



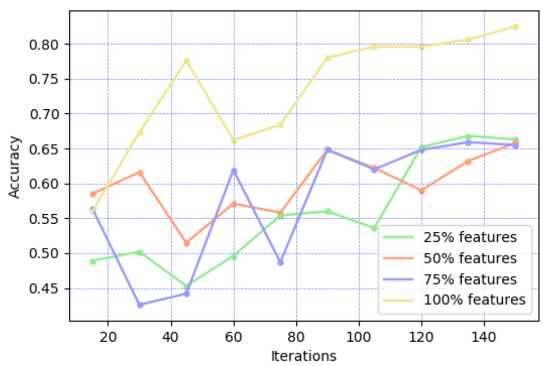
(a) Splice-Training Set



(b) Splice-Tetsing Set



(c) Satimage-Training Set

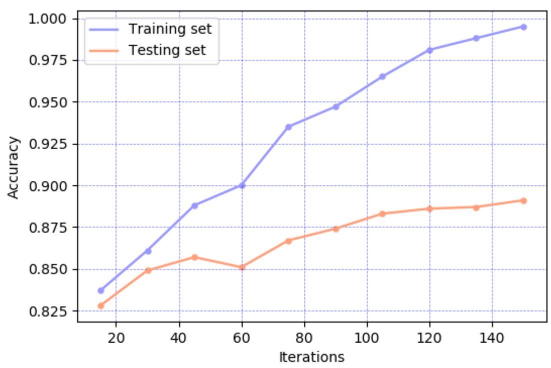


(d) Satimage-Testing Set

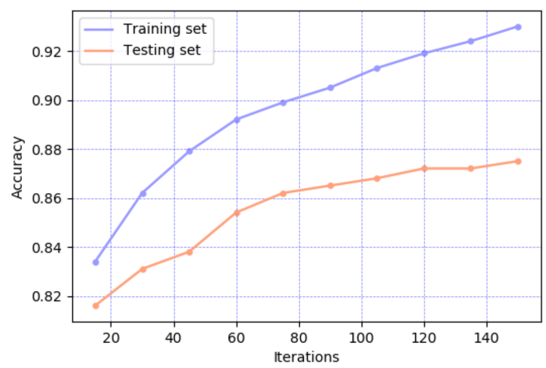
Figure 4: The performances of SVM under different dimensions of features.

3.3 Neural Network

In this section, I would also show the performances of MLP in the two datasets under different configurations (e.g. optimization algorithms, network architectures). Similarly, I would analyze each result like Sec. 3.2.



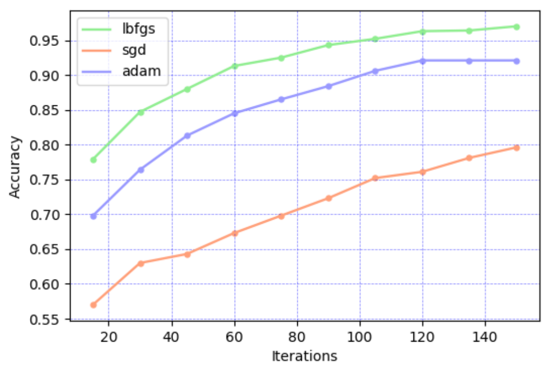
(a) Splice



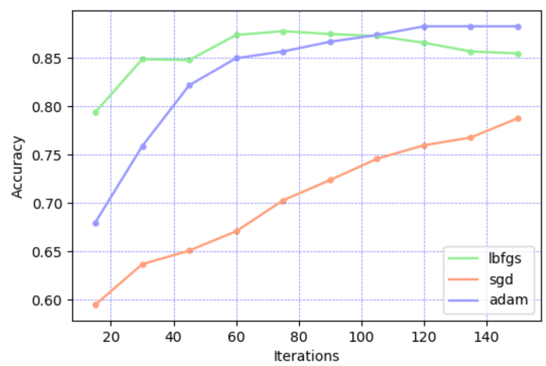
(b) Satimage

Figure 5: The baseline of MLP.

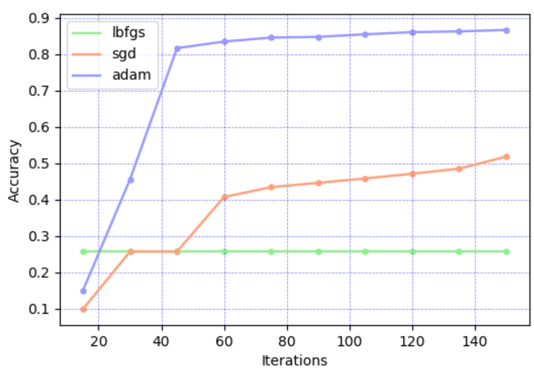
3.3.1 Optimization Algorithm



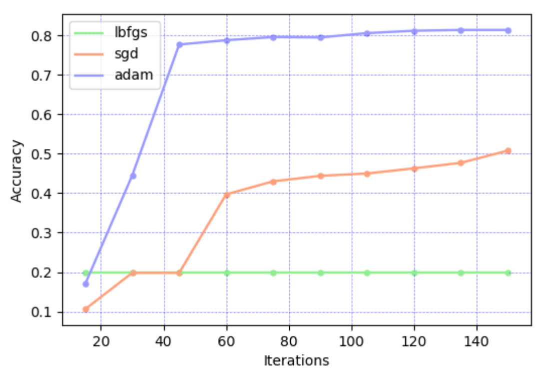
(a) Splice-Training Set



(b) Splice-Tetsing Set



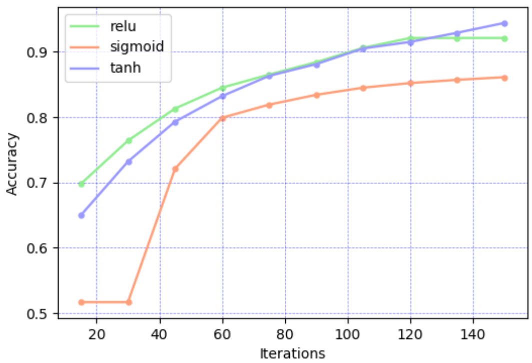
(c) Satimage-Training Set



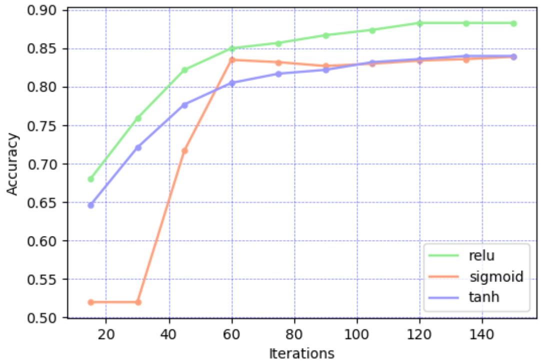
(d) Satimage-Testing Set

Figure 6: The performances of MLP under different optimization algorithms.

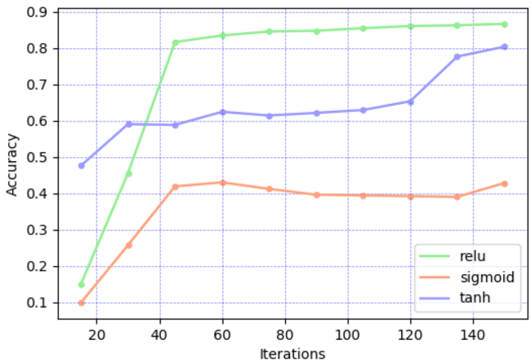
3.3.2 Activation Function



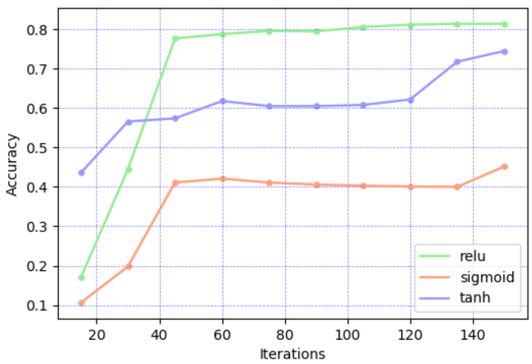
(a) Splice-Training Set



(b) Splice-Tetsing Set



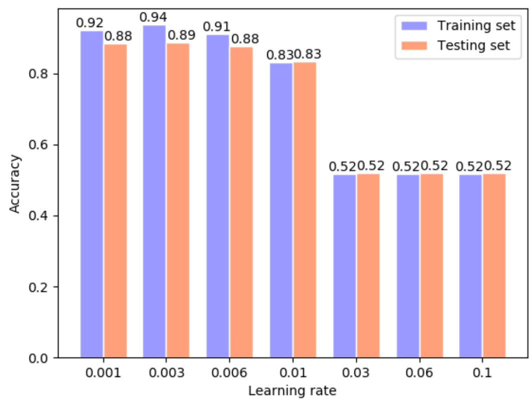
(c) Satimage-Training Set



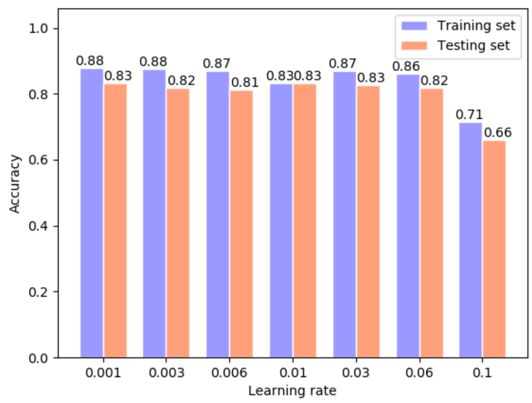
(d) Satimage-Testing Set

Figure 7: The performances of MLP under different activation functions.

3.3.3 Learning Rate



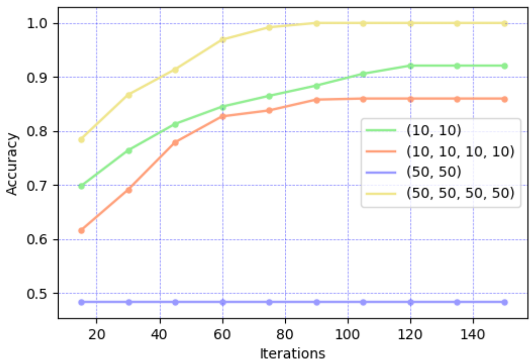
(a) Splice



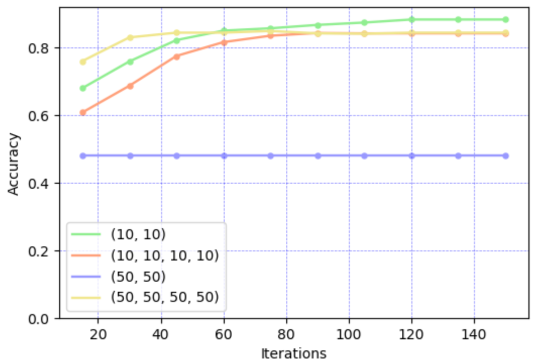
(b) Satimage

Figure 8: The performances of MLP under different learning rate.

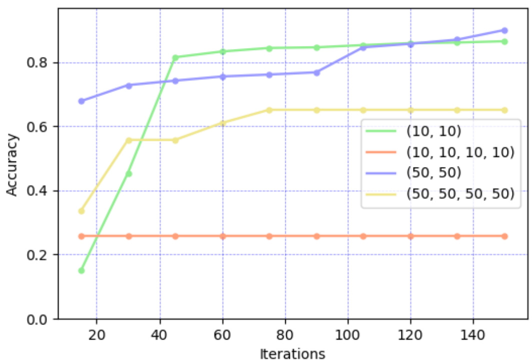
3.3.4 Network Architecture



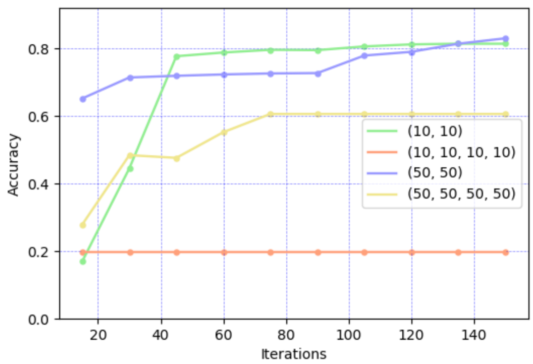
(a) Splice-Training Set



(b) Splice-Tetsing Set



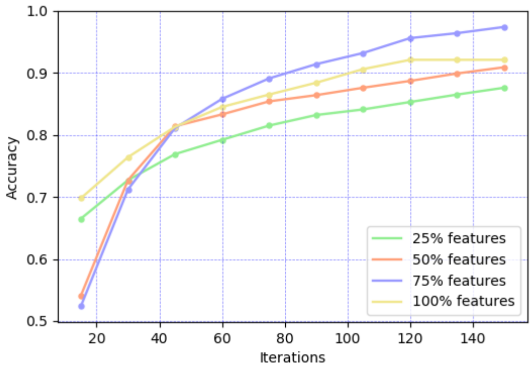
(c) Satimage-Training Set



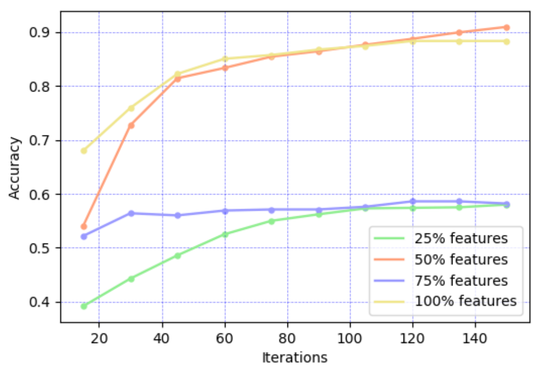
(d) Satimage-Testing Set

Figure 9: The performances of MLP under different network architectures.

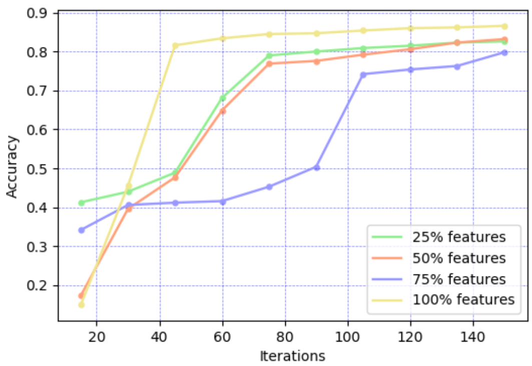
3.3.5 Dimension of Features



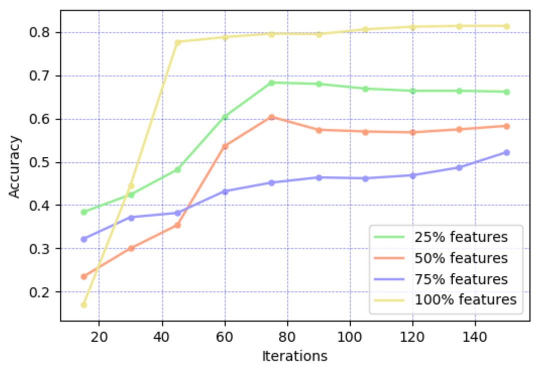
(a) Splice-Training Set



(b) Splice-Tetsing Set



(c) Satimage-Training Set



(d) Satimage-Testing Set

Figure 10: The performances of MLP under different dimensions of features.

4 Comparison Between SVM and Deep Learning Algorithm Benchmark

A Appendix

A.1 Details of Datasets in Experiments

A.1.1 Splice [4]

Basically, according to the biological knowledge, splice junctions are points on a DNA sequence at which 'superfluous' DNA is removed during the process of protein creation in higher organisms.

Splice dataset aims to recognize two classes of splice junctions, given a DNA sequence, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out).

A.1.2 Satimage [5]

Satimage dataset is also called **satalog** dataset. It contains multi-spectral values of pixels in 3×3 neighbourhoods in satellite images, and the classification associated with the central pixel in each neighbourhood.

As a classification dataset, the aim of satimage is to predict the classification, given the multi-spectral values. In the sample database, the class of a pixel is coded as a number.

A.1.3 ConvexNonConvex [6]

The ConvexNonConvex dataset consists of convex regions with pixels of value 255. Such regions are constructed via the intersection of a number of half-planes whose location and orientation were chosen uniformly at random. In the meanwhile, the number of half-planes is also sampled randomly according to a geometric distribution with parameter 0.195.

To ensure the validity of the data, a candidate convex image is rejected once there are less than 19 pixels in it.

References

[1] "Libsvm data: Classification, regression, and multi-label." `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

[2] "Lisa." `http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/WebHome)`.

[3] "scikit-learn." `http://scikit-learn.org/stable/`.

[4] "Molecular biology (splice-junction gene sequences) data set." `http://archive.ics.uci.edu/ml/datasets/Molecular+Biology+%28Splice-junction+Gene+Sequences%29`.

[5] "Statlog (landsat satellite) data set." `https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)`.

[6] "Convexnonconvex." `http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/ConvexNonConvex`.