Project Report for the AI

Hao Sun
Student ID:516030910362
Shanghai JiaoTong University
haosun_sjtu@qq.com

***Abstract*—The report provides a solution for the classification problem under the dataset *Gene_Chip_Data*. The dataset contains genes(features) and observations(gene chip experiments). After pre-processing the data, We dig out more useful and meaningful data in the dataset, and apply our classfication methods including machine learning and deep learning to cluster them. The method we used includes SVM(Support Vector Machine), LR(Logistic Regression), and MLP(Multi-Layer Perceptron, a deep learning algorithm with neural network). We optimize each method by adjusting finely the parameters, and compare among them, so as to obtain the applicability, pros and cons of different classification methods in different situations.**

***Index Terms*—Classification, Machine Learning, Deep Learning**

I. Pre-Processing data

A. Principle Component Analysis

IN the largest text file *mircoarray.original.txt*, The rows represents the features of each sample. Without any dimension reduction, the approximate 23000 dimensions are extremely difficult for subsequent classification. Thus first of the whole project, we use PCA(Principle Component Analysis) to reduce dimension of the dataset.

PCA[**?**] is a technique that can be used to simplify a dataset .It is a linear transformation that chooses a new coordinate system for the data set such that greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on. PCA can be used for reducing dimensionality by eliminating the later principal components.

Consider a data matrix X, where each of the n rows represents a different repetition of the experiment, and each of the p columns gives a particular kind of feature.

Let $x_1, x_2 \cdots, x_n$ be a set of p vectors of dimension $n \times 1$ and let $\bar{x}$ be their average Let X be the $N \times n$ matrix with columns $x_1 - \bar{x}, x_2 - \bar{x}, \cdots x_n - \bar{x}$:

$$X = \begin{bmatrix} \mathbf{x}_1 - \overline{\mathbf{x}} & \mathbf{x}_2 - \overline{\mathbf{x}} & \cdots & \mathbf{x}_n - \overline{\mathbf{x}} \end{bmatrix} \tag{1}$$

Note that subtracting the mean is equivalent to translating the coordinate system to the location of the mean.

My student ID is 516030910362.
My teammate is Mingfei Li.

Let $Q = XX^T$ be the $N \times N$ matrix:

$$\begin{aligned} Q &= XX^T \\ &= \begin{bmatrix} \mathbf{x}_1 - \overline{\mathbf{x}} & \mathbf{x}_2 - \overline{\mathbf{x}} & \cdots & \mathbf{x}_n - \overline{\mathbf{x}} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_1 - \overline{\mathbf{x}})^T \\ (\mathbf{x}_2 - \overline{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \overline{\mathbf{x}})^T \end{bmatrix} \end{aligned} \tag{2}$$

Note that Q is square and symmetric, and Q is the covariance matrix.

Therefore we get that each x_j can be written as:

$$\mathbf{x}_j = \overline{\mathbf{x}} + \sum_{i=1}^{i=n} g_{ji} \mathbf{e}_i \tag{3}$$

where $\mathbf{e_i}$ are the n eigenvectors of Q with non-zero eigenvalues, the scalars g_{ji} are the coordinates of are the coordinates of x_j in the space. in the space. Note that the eigenvectors $\mathbf{e_1 e_2} \ldots \mathbf{e_n}$ span and eigenspace. And $\mathbf{e_1 e_2} \ldots \mathbf{e_n}$ are $N \times 1$ orthonormal vectors(directions in N-Dimensional space)

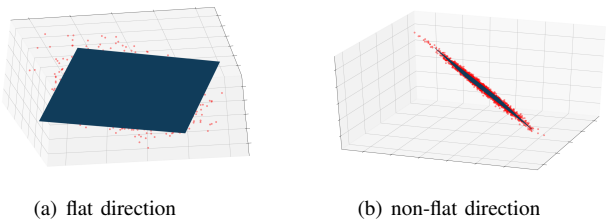
Expressing x in terms of $\mathbf{e_1}, \cdots \mathbf{e_n}$ has not changed the size of the data. However, if the points are highly correlated many of the coordinates of x will be zero or closed to zero.

Sort the eigenvectors e_i according to their eigenvalue: $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$

Assume that $\lambda_i \approx 0$ if $i > k$,

Then we have the final conclusion

$$\mathbf{x}_j \approx \overline{\mathbf{x}} + \sum_{i=1}^{k} g_{ji} \mathbf{e}_i \tag{4}$$



(a) flat direction (b) non-flat direction

Fig. 1: The two figures aid in illustrating how a point cloud can be very flat in one direction–which is where PCA comes in to choose a direction that is not flat [**?**].

We select the number of components such that the amount of variance that needs to be explained is greater than the percentage specified by 0.9. In other words, we can save at

least 90 percent main information of the whole features after dimension reduction.

After implement of PCA, the original data which is a 22286×5896 matrix became a matrix of dimension 453×5896. The text file size was reduced from 2G to 48.8MB.

B. Generating Training Set

In another text file *E-TABM-185.sdrf.txt*, there is a 5896×30 matrix where each row corresponds to a sample, which is also an observation in *microarray.original.txt*. And each sample has at most 30 attributes, which can have one or more values(categories).

For generating training set, we must choose an attribute first. The corresponding samples will be picked out in this process (for each attribute, Not all samples are valid)

In order to compare our classification methods more comprehensively, and considering that we tend to pay more attention to some specific attributes, we finally chose these four attributes as follows.

Attributes	Number of valid samples	Number of sample categories under this attribute
Material Type	5896	2
Characteristics [BioSourceType]	3087	12
Characteristics [DiseaseStage]	964	17
Characteristics [DiseaseState]	4028	193

TABLE I: The chosen attributes and the related information

We shuffled, and divided the valid samples into training set and test set according to the ratio of 9:1.

II. Method

A. Support Vector Machine

In machine learning, support-vector machines [**?**](SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an traditional SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Given a training dataset $D(x_i, y_i)$, where x_i denotes n observations of malware signatures, $x_i \in R^N$,$i = 1, \cdots, N$; and y_i is the corresponding class label whose value is either 1 or -1(i.e., malicious or benign), indicating the class to which the point x_i belongs, $y_i \in \{1, 1\}$, assigned to each observation x_i . Each behavioural signature x_i is of dimension d corresponding to the number of propositional variables.

$$\mathrm{D} = \left\{ (x_i, y_i) \,|x_i \in R^N, y_i \in \{1, -1\} \right\}_{i=1}^{N} \tag{5}$$

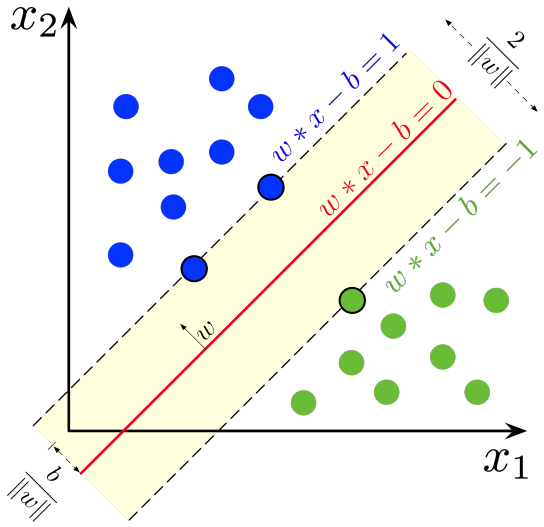


Fig. 2: Support Vector Machine [**?**]

A typical clustering problem is identifying the maximum margin of a hyperplane that divides the points exhibiting $y_i = 1$ from those exhibiting $y_i = -1$. Any hyperplane can be written as the set of points, x, satisfying the following formula:

$$\mathbf{w} \cdot \boldsymbol{x}_i + \boldsymbol{b} = 0 \tag{6}$$

where $\cdot$ denotes the inner product and $\mathbf{w}$ denotes the normal vector of the hyperplane. The parameter $\frac{b}{\|w\|}$ determines the offset of a hyperplane from the origin along the normal vector W. Generally, a decision function $D(x)$ is defined for clustering as $D(x) = \mathbf{w} \cdot \mathbf{x} + \mathbf{b}$.

Other than performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

In addition to performing binary classification, SVMs can perform a multi-classification by the "one-against-one" approach or "one-vs-the-rest" multi-class strategy.

B. Logistic Regression

In statistics, the logistic model[**?**] is a widely used statistical model that, in its basic form, uses a logistic function to model a binary dependent variable. In regression analysis, logistic regression is estimating the parameters of a logistic model.In addition, it is a form of binomial regression.

An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is useful because it can take any real input $t(t \in R)$, whereas the output always takes values between 0 and 1 and hence is interpretable as a probability. The logistic function $\sigma(t)$ is defined as follows:

$$\sigma(\boldsymbol{t}) = \frac{1}{1 + e^{-t}} \tag{7}$$

A graph of the logistic function of the t-interval $(-6, 6)$ is shown in Fig.3. Assume that $\mathbf{t}$ is a linear function of a single explanatory variable $\mathbf{x}$. We can then express $\mathbf{t}$ as follows

$$\mathbf{t} = \boldsymbol{w}_0 + \boldsymbol{w}_1 \boldsymbol{x}_1 + \cdots + \boldsymbol{w}_n \boldsymbol{x}_n = \boldsymbol{W} \boldsymbol{x} \tag{8}$$

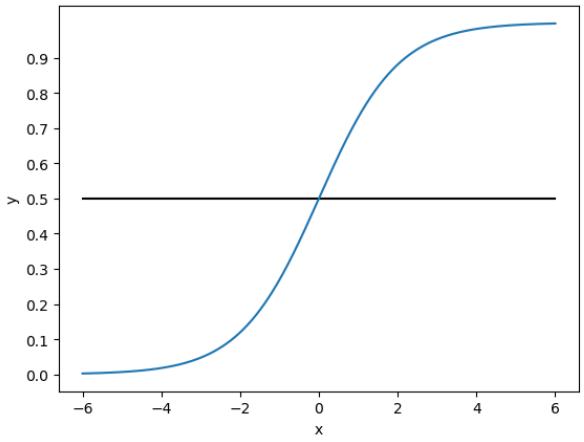


Fig. 3: Sigmoid function: $f(x) = \frac{1}{1+e^{-x}}$

And the logistic function can now be written as:

$$F(x) = \frac{1}{1 + e^{-wx}} \tag{9}$$

C. MLP

A multi-layer perceptron(MLP)[**?**] is a class of feed-forward artificial neural network. An MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

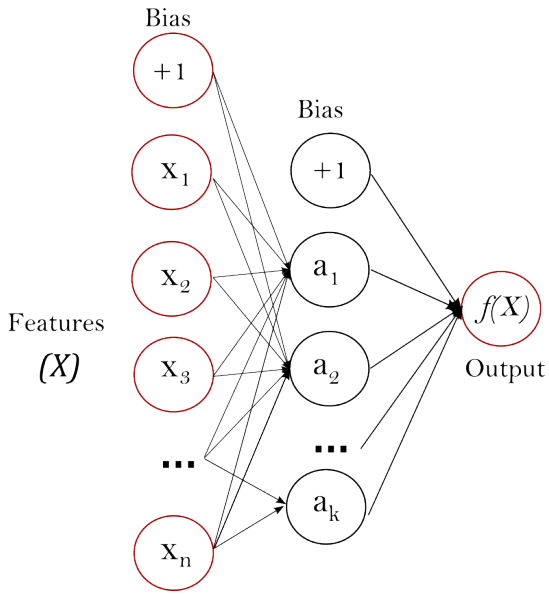


Fig. 4: multilayer perceptron network(one hidden layer) [**?**]

The leftmost layer, known as the input layer, consists of a set of neurons $\{x_i | x_1, x_2, \ldots, x_m\}$ representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + \ldots + w_mx_m$, followed by a non-linear activation function $g(\cdot) : R \to R$ - like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values.

Given a set of training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in \mathbf{R}^n$ and $y_i \in \{0, 1\}$, a one hidden layer one hidden neuron MLP learns the function $f(x) = W_2 g\left(W_1^T x + b_1\right) + b_2$ where $W_1 \in \mathbf{R}^m$ and $W_2, b_1, b_2 \in \mathbf{R}$ are model parameters. W_1, W_2 represent the weights of the input layer and hidden layer, respectively; and b_1, b_2 represent the bias added to the hidden layer and the output layer, respectively. $g(\cdot) : R \to R$ is the activation function, there are four common activation function as follows:

1) hyperbolic tan: $g(z) = (e^z - e^{-z})/(e^z + e^{-z})$
2) identity function: $g(z) = z$
3) logistic sigmoid function: $g(z) = 1/(1 + e^{-z})$
4) rectified linear unit function: $g(z) = \max(0, z)$

Similar to other classification method, MLP takes different measures between binary and multiple classification.

For binary classification, $f(x)$ passes through the logistic function $g(z) = 1/\left(1 + e^{-z}\right)$ to obtain output values between zero and one. A threshold, set to 0.5, would assign samples of outputs larger or equal 0.5 to the positive class, and the rest to the negative class.

If there are more than two classes, $f(x)$ itself would be a vector of size being the number of classes. Instead of passing through logistic function, it passes through the softmax function, which is written as,

$$\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_{l=1}^{k} \exp(z_l)} \tag{10}$$

The loss function of MLP for classification is Cross-Entropy, which in binary case is given as,

$$\text{Loss}(\hat{y}, y, W) = -y \ln \hat{y} - (1 - y) \ln(1 - \hat{y}) + \alpha \|W\|_2^2 \tag{11}$$

where $\alpha\|W\|_2^2$ is an L2-regularization term (aka penalty) that penalizes complex models; and $\alpha > 0$ is a non-negative hyperparameter that controls the magnitude of the penalty.

Starting from initial random weights, multi-layer perceptron (MLP) minimizes the loss function by repeatedly updating these weights. After computing the loss, a backward pass propagates it from the output layer to the previous layers, providing each weight parameter with an update value meant to decrease the loss.

In gradient descent, the gradient ∇Loss_W of the loss with respect to the weights is computed and deducted from W. More formally, this is expressed as,

$$W^{i+1} = W^i - \epsilon \nabla \text{Loss}_W^i \tag{12}$$

where i is the iteration step, and ϵ is the learning rate with a value larger than 0.

The algorithm stops when it reaches a preset maximum number of iterations; or when the improvement in loss is below a certain, small number.

III. Results

In this section, the implementations and the results of the models mentioned above will be shown.

A. Support Vector Machine

We input training data set in SVM and observe the accuracy of each iteration in SVM, from which we drew a figure Fig.5.

The figure shows that in four different attributes, SVM has a significant difference in accuracy. Here are a series of detailed discovers and conclusions as follows.

Fig. 5: SVM accuracy rate vs iterations in four attributes

1) SVM perform pretty well in binary classification. From the blue line which represents the material type, we can see that after sufficient iterations, the accuracy rate reaches very close to 1. In fact, according ro our observations, the accuracy rate can be exactly 1.
2) SVM can sometimes perform well in multi-categories attribute such as BioSource Type. On the other hand, we tend to consider this high accuracy rate of the attribute is caused by the specificity of the data. The number of each classification may be relatively average. Conversely, the number of each classification in attribute Disease State, is pretty distinct, which means we have some very few samples for some category, making the prediction progress more difficult.
3) Consistent with the previous expectation, it is more difficult to classify the attributes of multiple categories, especially those attributes containing a small number of valid samples, resulting in a different degree of underfitting. The accuracy distance between multi-classification and binary classification is about 0.15, which we think is an acceptable number. In attribute Disease State, .there are 193 categories in total and more than 100 categories with sample size less than 10. In the case of such a small training set, it is really hard for us to improve our accuracy without changing the data set.
4) Before the accuracy of SVM converges, there will be different degrees of jitter, which seems to have nothing to do with attribute complexity. From Fig.5, it seems that there is still a set of data that has not yet converged. In addition, SVM runs very fast compared to the other methods especially under the condition of large number of categories.

B. Logistic Regression

Similar to SVM, we draw a figure Fig.6 which shows how classification accuracy is changed as iterations increases of Logistic Regression. However, we only drew three curves because the rest one is too time-consuming. Here come some conclusions from the figure.

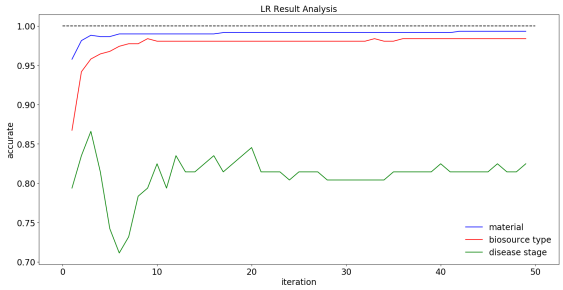


Fig. 6: LR accuracy rate vs iterations in three attributes

1) The reason for only three attributes is that running logistic regression is too time-consuming especially the number of classification is large. For the multi-classification LR, as we mentioned before, the decision tree has to be built layer by layer. which means the running time of the undrawn attribute disease state is as 200 times running time as the material type.
2) Though LR is time-comsuming, the effect of classification is not any obviously worse than SVM. As we can see, the attribute material type and biosource type accuracy rate is very close to 1. The disease stage accuracy rate maintains around 0.82, not much different from SVM.
3) The accuracy of LR can converge at a very small number iterations, (average 25 iterations). And the curves are much smoother than SVM's. Meanwhile, the curves perform well abnormally at the very beginning, we consider this is determined by the initial parameters of LR.

C. Multi-Layer Perceptron

We construct a fully-connected four layers structure neural network. The number of neurons in the neural network has a relatively large impact on the result. Through continuous testing and filtering, we finally set the hidden layer structure of 256×256.

Running MLP is pretty time-consuming and we didn't compare the four attributes(meanwhile we thought it unnecessary because in SVM and LR we have already compared the differences among them). Instead, we decided a specific attribute, and observe the influences of other parameters such as activation function and L2 penalty(regularization term).

We use the attribute Disease Stage to study what influence the activation function take for the loss in the MLP, the result is shown in Fig.7 as below.

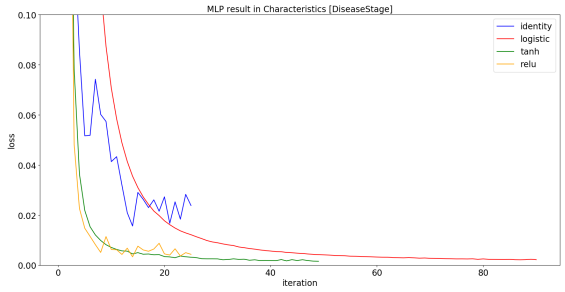


Fig. 7: MLP loss vs iteration in different activation function (if training loss does not improve more than tol=0.000100 for 10 consecutive epochs, then Stop).

From the image above,we know that under the two activation function: identity function $g(z) = z$ and rectified linear unit function $g(z) = \max(0, z)$, the loss converges very fast. However, that is nothing with the converging value, and the loss is greater correspondingly than the logistic function and hyperbolic tan function.

And from Table 2(in Discussions part)we can know that there exists some differences among them in accuracy rate, note that $loss$ is Not necessarily equal to $1 - accucacy\ rate$, Instead, they have very little to do with each other. It can be seen that when we choose tanh activation function, the accuracy rate is the highest and the curve is really smoother than identity function and rectified linear unit function.

Other than the activation function, we did an experiment about L2 penalty(regularization term). We observed the final accuracy rate by adjusting α=L2 penalty.

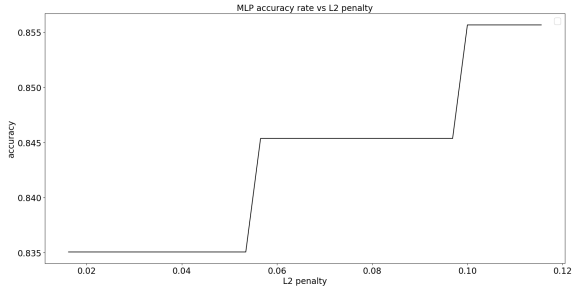


Fig. 8: MLP accuracy rate vs L2 penalty

From the figure above, We found that the accuracy rate increases by leaps and bounds with the L2 penalty α increasing(no position higher than the highest position in the figure).

Note that in the figure, the two segments of the straight line with a rising fixed slope are not from actual tests, but due to insufficient step size(the MLP is time-consuming). In fact, I guess this is a sudden change or a sharp rise.

IV. Discussions

All methods we did and their accuracies are shown in Table 2, we used cross validation to get more accurate result , which means the accuracy does not change by at least the third decimal places as we divide different training sets and test sets.

We can get many conclusions and inferences by comparing between different classifications.

In Material Type attribute, each method has excellent performance. The accuracy of all methods is over 99.5%, and in deep learning, Multi-Layer Perceptron with logistic activation function and α being 10^{-4} performs the best. With this accuracy, we could say that we can almost decide the correct classification in this attribute.

As for the second row, Characteristics [BioSourceType] is a multi-classification attribute. However, the attribute is relatively easy to be clustered by these methods, which is caused by the reason that the distinction between the features of different classification is obvious. We found that the MLP with logistic activation function performed still the best in this attribute. And it is the only one whose accuracy is over 99%.

In Characteristics DiseaseStage, the accuracy of all the methods has dropped a lot. We consider the main reason is that the number of samples is not enough. The number is not exceeded 1000. The models may be still in underfitting stage. The MLP with logistic activation function performs still best with 86.392% accuracy. We guess in a common classification problem, the number of samples are limited, just like this situation. We have tried some special ways to improve such as ignoring PCA or remaining a more number of features. But the result is still not ideal.

In Characteristics DiseaseState, the most difficult attribute to cluter, all method perform not as well as those in Characteristics BioSourceType. The number of samples is large but the number of classification is 193, also very large. Logistic Regression even cannot get a complete result in 2 minute. Meanwhile, we find that SVM handles this task still very fast,almost a third of the time as other methods.

Of all the methods, MLP with logistic activation function and α being 0.0001 performs the best, though it has a slight accuracy behind MLP with L2 penalty α being 0.1 in the attribute Characteristics DiseaseState. The time complexity of this model is acceptable and if we prefer the slight forward in accuracy in some aspects, MLP with these parameters mentioned above is the first one to consider.

From the view of comparison between machine learning and deep learning, SVM and LR belongs to machine learning method, and MLP belongs to deep learning method. Only from the accuracy perspective, deep learning method performs a little better. However, as for the efficiency perspective, SVM of machine learning is much more efficient than other method. Specially, Logistic Regression performs very badly in time complexity in multi-classification task. Also, deep learning needs an amount of computation and feedback, and the iteration number is also more than machine learning ways.

All these methods will be consider to be improved in future especially deep learning. The neural network we built contains

Attributes	SVM	Logistic Regression	MLP (tanh,$\alpha = 0.0001$)	MLP (tanh,$\alpha = 0.1$ *)	MLP (relu,$\alpha = 0.0001$)	MLP (logistic,$\alpha = 0.0001$)	MLP (identity,$\alpha = 0.0001$)
Material Type	99.797%	99.695%	99.628%	99.797%	99.798%	**99.831%**	99.593%
Characteristics [BioSourceType]	98.964%	98.705%	98.317%	98.058%	98.317%	**99.288%**	97.929%
Characteristics [DiseaseStage]	84.536%	81.443%	82.474%	86.186%	84.124%	**86.392%**	83.093%
Characteristics [DiseaseState]	83.275%	83.747%*	83.275%	**84.665%**	83.623%	84.069%	82.730%

TABLE II: Accuracy of all methods in each attribute

1. tanh,relu,identity,logistic represents the activation funtion, α represents the L2 penalty.
2. Logistic Regression method is not fit for multi-classification because it is too time-consuming. This accuracy is not sufficiently cross-verified, may have some errors.

only two hidden layer (256,256). As far as we know, the deep learning method can be improved much if the hidden layer is (256,512,256). Meanwhile, the kernel trick in SVM can be considered(the method is not useful in these four attributes in our experiments).

V. Conclusion

In this project, the task is to classify the samples with lots of features. We used PCA to reduce dimension and train a classifier of machine learning including SVM and LR, deep learning approach including MLP. The final result is that deep learning has a better accuracy but it is time-consuming. And SVM remains to be a good choice for both lightweight and high performance.

Finally, I want to express my gratitude to Professor Bo Yuan, who guided me a lot in Artificial Intelligence, let me fall in love with AI. Meanwhile, I want to express my gratitude to teacher assistant and my teammate.

References

[1] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, pp. 559–572, 1901.
[2] https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_3d.html.
[3] V. N. Cortes, Corinna; Vapnik, "Support-vector networks," *Machine Learning*, pp. 273–297, 1995.
[4] https://en.wikipedia.org/wiki/Support-vector_machine.
[5] D. Walker, SH;Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, pp. 167–178, 1967.
[6] F. x. Rosenblatt, "Principles of neurodynamics: Perceptrons and the theory of brain mechanisms," *Spartan Books*, 1961.
[7] https://scikit-learn.org/stable/modules/neural_networks_supervised.html.