Comparison Study of SVM and MLP

Yingxue Su, yingxue@math.uh.edu
Qianfan Bai,qfbaiwei@gmail.com

April 2020

Supported Vector Machine (SVM) and Multilayer Perceptron (MLP) are two most popular algorithms for classification tasks. In this report, we will first introduce a simple MLP with only one hidden layer in section 1. In section 2, we will explain that the MLP is actually a mixture of SVMs under some assumptions. Then, we will compare the performance of SVM and MLP on the EEG data set in section 3.

1 The Structure of MLP

Suppose we have L cases in the training set, for each case (x_l, y_l), y_l is the true label of $x_l \in \mathbf{R}^k$ for $l = 1, ..., L$. MLP normally contains three parts, input layer, hidden layers and output layer. We will only talk about the MLP with one hidden layer in this report like the structure in figure 1. The numbers of units in the input layer and output layer are k and m, respectively. The number of units in the hidden layer N is chosen by us.

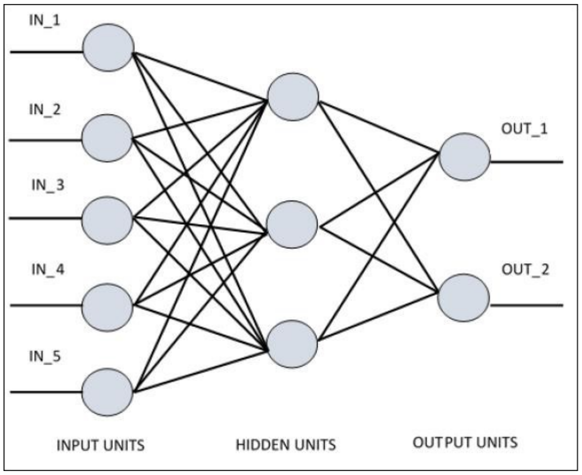
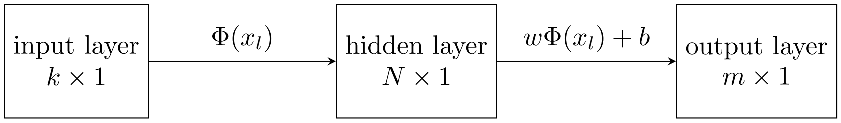


Figure 1: MLP with one hidden layer



As the graph above, in the input layer, we have the vector x_l. We use function $\Phi : \mathbf{R}^k \to \mathbf{R}^N$ to get the values of units in the hidden layer. Function $\Phi(x) = (\Phi_1(x), ..., \Phi_N(x))$ is defined as $\Phi_i(x) = h(v_i x + b)$ where h is the response function, $v_i \in \mathbf{R}^k$ and $b \in \mathbf{R}$. h is normally sigmoid function, Relu function or hyperbolic tangent function. After we get the values of units in the hidden layer, we use the decision function $f_\theta(x) = w\Phi(x) + b$ to get the values of units in the output layer, where $w \in \mathbf{R}^{m\times N}$, $b \in \mathbf{R}^m$ and θ represents all the variables in MLP.

The MLP is achieved by the minimization of a loss function using the gradient descent method. The loss function usually look like

$$loss = \frac{\mu}{2}||\theta||^2 + \frac{1}{L}\sum_{l=1}^{L} Q(f_\theta(x_l), y_l), \tag{1}$$

where $Q(f_\theta(\cdot), \cdot)$ is the criterion function. We usually choose Q to be the Cross-Entropy criterion for the classification task. Replace Q with the Cross-Entropy criterion in (1), we will get the following optimization problem:

$$\min \quad loss = \frac{\mu}{2}||\theta||^2 + \frac{1}{L}\sum_{l=1}^{L} \log(1 + \exp(-f_\theta(x_l) \cdot y_l)). \tag{2}$$

The optimal θ^* is achieved by the gradient descent method:

$$\theta_{t+1} = \theta_t - \epsilon_t \frac{\partial}{\partial\theta} loss_t,$$

where ϵ_t is the learning rate at step t and $\epsilon_t \to 0$ when $t \to \infty$.

2 Links Between SVM and MLP[1]

Recall that the decision function of SVM also has the form

$$f_\theta(x) = w\Phi(x) + b.$$

The SVM problem is equivalent to the following optimization problem:

$$\min \quad \frac{\mu}{2}||w||^2 + \frac{1}{L}|1 - y_l f_\theta(x_l)|_+,$$

where $|1-z|_+ = max(0, 1-z)$ is the margin criterion. The optimization problem is equivalent to:

$$\begin{aligned} \min \quad & \frac{\mu}{2}||w||^2 + \frac{1}{L}\xi_l \\ \text{subject to} \quad & \xi_l \geq 0 \\ & 1 - \xi_l - y_l f_\theta(x_l) \leq 0 \end{aligned} \tag{3}$$

We plot the margin criterion and the Cross-Entropy criterion in figure 2, we can say that the margin criterion is a 'hard' version of the Cross-Entropy criterion. Therefore, if we replace the Cross-Entropy criterion in the the optimization problem (2) with the margin criterion and rewrite it we get

$$\begin{aligned} \min \quad & \frac{\theta}{2}||\theta||^2 + \frac{1}{L}\xi_l \\ \text{subject to} \quad & \xi_l \geq 0 \\ & 1 - \xi_l - y_l f_\theta(x_l) \leq 0 \end{aligned} \tag{4}$$

which is very similar to the optimization problem (3) of SVM.

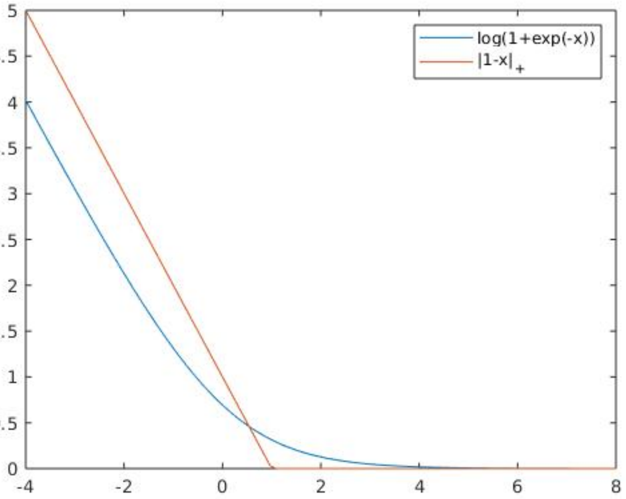


Figure 2: Margin criterion and Cross-Entropy criterion

If we compare the KKT conditions of optimization problem (4) and optimization problem (3), we notice that the optimal (w^*, b^*, Φ^*) which satisfy the KKT condition of MLP optimization problem also satisfy the KKT condition of SVM optimization problem. Therefore, (w^*, b^*) are the optimal weights for SVM using Φ^* as the future space. So MLP maximizes the margin of the hidden layer space defined by Φ. If we look at the cases x_l such that $|v_i^* x_l + d_i^*| \leq 1$, then we can say that the units i on the hidden layer form a linear SVM for cases x_l. Hence, we conclude that MLP with one hidden layer is a mixture of SVMs.

3 Numerical Task and Results

We will perform a numerical task using the EEG data set. We want to compare the performance of SVM and MLP on the same data set. The original EEG data set has 5 classes. Since 5 classes is hard to classify, we rearranged the original data set to the following 3 classes:

Class 1: the EEG signal is related to a tumor. (4600 cases)

Class 2: the EEG signal is recorded during an eye activity. (4600 cases)

Class 3: the EEG signal is recorded during a seizure activity.(4600 cases)

We use the RBF kernel for the SVM and choose the hidden layer size of MLP to be $N = 82$. We get the following results:

- The accuracy of this SVM on the training set and test set are 0.93126 and 0.89034, respectively.
- The confusion matrix of this SVM on the test set is

$$C\_M_{test} = \begin{bmatrix} 0.9116 & 0.0667 & 0.0217 \\ 0.2154 & 0.7809 & 0.0037 \\ 0.0197 & 0.0071 & 0.9732 \end{bmatrix}$$

- The accuracy of this MLP on the traning set and test set are 0.8694 and 0.8490, respectively.
- The confusion matrix of this MLP on the test set is

$$C\_M_{test} = \begin{bmatrix} 0.8055 & 0.1529 & 0.0416 \\ 0.1507 & 0.8240 & 0.0253 \\ 0.0298 & 0.0082 & 0.9620 \end{bmatrix}$$

- Suppose the average activity of units in the hidden layer of MLP is $PROF_i = \frac{1}{4600}\sum_j \Phi(x_j)$ for cases $x_j \in class_i$, we plot the following figure 3.

Based on the above results, we get the following conclusions:

- The accuracy on the whole test set of SVM has the accuracy interval $[0.8855, 0.8952]$. Therefore, SVM has a better performance on classifying the 3 classes than MLP.
- By looking at the confusion matrix of SVM on the test set, the accuracy intervals of classifying 3 classes of SVM are $[0.9040, 0.9192]$, $[0.6689, 0.7921]$ and $[0.9689, 0.9775]$, respectively. SVM has better performance in classifying class 1 and class 3. MLP has better performance in classifying class 2.
- By looking at the hidden layer activity of the MLP after training, we can see that the hidden layer units are much more active to the cases belonging to class 3 and they are similarly active to cases from class 1 and 2, which explains why MLP has a higher accuracy in classifying class 3.
- Even though MLP has a more complicated structure than SVM, the performance of MLP is not guaranteed to be better than SVM.

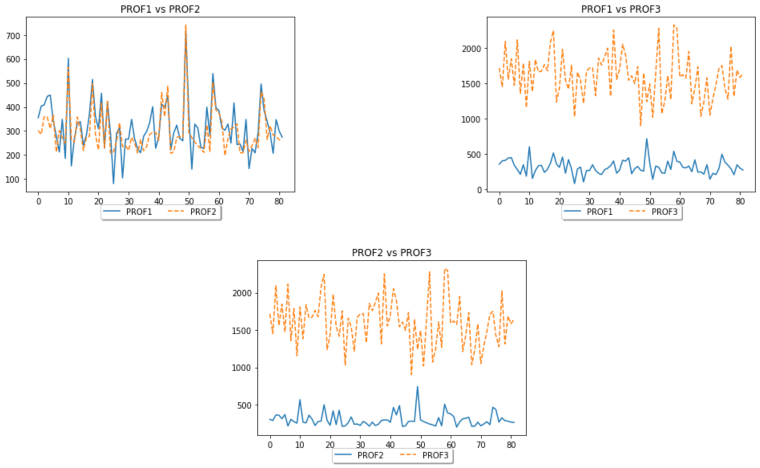


Figure 3: hidden layer activity

References

[1] Samy Bengio Ronan Collobert. *Links Between Perceptrons, MLPs and SVMs*. Feb 6,2004.