Machine Learning - deep and structured Hw2

B00902006 何主恩
B00902053 馮　硯
B00902064 宋昊恩
R03922163 姜佳昀

May 8, 2015

1 Problem Statement

For each sequential sound record, said S, is consisted with N frames. It it worth to note that n is a variable for different sound records. S can be described as a list of $(\mathbf{x}, y)$, where $\mathbf{x}$ is a vector of length k, and y is a scalar label. To be clear, let $S = [(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)]$ be the trainig data, and our target is to model the latent pattern of these sequential data.

2 Proposed Solution

To solve this structured learning problem, we propose three solutions to it. One is the implementation of structured SVM provided by SVM$^{\text{struct}}$. Another one is based on the work combined of deep learning and structured learning, called DNN-HMM. The last one is based on the recurrent of data, called RNN.

2.1 Structured SVM

2.1.1 Algorithms

Our target is to maximize the joint probability of given sequential data. It can be denoted as $p(\mathbf{seq}) = p(y_1)p(\mathbf{x}_1|y_1) \bullet \prod_{i=2}^{n} p(y_i|y_{i-1})p(\mathbf{x}_i|y_i)$. With the SVM$^{\text{struct}}$, we can train the probability for both observevation and transition table with given observed labels. After the training stage, we can simple use dynamic programming algorithm (DP algorithm) to obtain the most possible sequence label for each sequential sound record in testing data.

2.1.2 Experiments Results

Changeable factors include

- features for training and testing data
- parameter c in *svm_empty_learn*
- omitted trimming length

Firstly, we changed our origin feature length from 69 to 1024 and 2048. The latter two feature data are extracted from the last hidden layer of neural networks of 1024 and 2048 nodes, respectively.

The second factor mentioned above is used to determine the effort SVM spend to fit the given data.

What's more, we can utilize the information leaked from the empty baseline that the average sequence length is 36.24324. To be more specific, we can determine an *omitted trimming length* that we finally eliminate all continuous labels below that length. In the following statistics, most of the submissions have already been trimmed to a suitable length, which means approaching average sequence length.

feature length	paramter c	trimming length	edit distance
69	1	2	29.47973
1024	1	3	19.84797
1024	2	1	18.93919
2048	1	1	22.75338

Table 2.1: Experiment Results Table

It is observed that the larger the parameter c is chosen, the longer time is needed. Since the machine which our experiments ran on was very busy, we cannot tell the accurate execution time for each experiment. However, we found out that the experiment with 69 features ran less than 20 minutes, whereas, 2048 features used more than 1000 minutes for model training.

Even though we spent a lot of time on implementing the whole structured SVM algorithm, its behvior was just leave-much-to-be-desired!

2.2 DNN-HMM

2.2.1 Algorithms

Use hw1 DNN result as observation probabilities in HMM, and use statistical count of bi-gram to approximate transition probabilities in HMM. Then use Viterbi to find the best prediction.

2.2.2 Features

Use fbank provided in the data.

2.2.3 Parameter tuning

We want to improved DNN based on hw1 work. Firstly, we replaced sigmoid unit with ReLU. The performance did not improved but take less training time.

Secondly, we apply dropout regularization, and the accuracy of hw1 test data is improved from about 0.7 to 0.733. Due to the limited computation power of our machine, we use parameters advised by the proposer of dropout. However, we found it is better to use a relatively small learning rate(0.001) rather than recommended large learning. Also, we found it is important to use smaller batch size(128) rather than bigger one(1024) witch used in hw1.

Thirdly, we apply max-norm regularization and experiment with different max-norm parameters(0.3 0.4), but no improvement have shown.

2.2.4 Loss functions

The DNN is using a softmax output layer with cross entropy loss.

2.2.5 Inference methods

We use Viterbi to predict the sequence, but we found it would stuck in a state and would not transit to another state, perhaps due to the fact that the count of one state to same state is much larger than one state to another. So, we first tried to scaled down the counts with logarithm, but the situation remain the same. Then we truncate the count of one state to same state to the max count of the state to others, but nothing improved.

2.2.6 Experiments Results

batch size	learning rate	drop rate	frame accuracy
128	0.001	0.2	0.72
256	0.001	0.2	0.67
1024	0.001	0.2	0.64
128	0.1	0.2	0.19
128	0.01	0.2	0.61
128	0.001	0.2	0.72
128	0.0001	0.2	0.525
128	0.001	0.2	0.71
128	0.001	0.5	0.735
128	0.001	0.6	0.70

Table 2.2: Experiment Validation Results in Frame Accuracy

2.3 DNN-HMM

2.3.1 Algorithms and brief result

Based on the DNN model, we also try to implement a simple RNN. We record the hidden layer output of the previous instance and make this output vector as the input to the hidden layer with features of current instance. But due to the limit of computation, it requires long training time, so we cannot use large amount of hidden nodes, and the best result we have is only 19.05405 in edit distance.

3 Conclusion

Our final experiment of each model shows that the best performance came from the trimmed Hw1 output. In other words, the concept of structured feature does not provide any improvement to our models.