1. Introduction

- RNNs use the softmax activation function in the last layer

\[ P(s|\mathbf{x}_t) = \frac{\exp(w_s^T h_t)}{\sum_{j=1}^N \exp(w_j^T h_t)} \]

Denominator is constant for classification

\[ \arg \max_s P(s|\mathbf{x}_t) = \arg \max_s w_s^T h_t \]

- Multiclass SVM, same classification:

\[ \arg \max_s \tilde{w}_s^T \phi(\mathbf{x}_t) \]

Replace softmax layer in RNN with SVM \( \rightarrow \) Recurrent SVM

2. Recurrent SVM

Architecture: LSTM+SVM

- 1st-step: fixed LSTM, training SVM using the quadratic programing.
- 2nd-step: fixed SVM, training LSTM using the subgradient methods

3. Max-Margin Sequence Training

What is the Margin? Objective Function?

\[ \log P(S_{1:T}|\mathbf{x}_{1:T}) \]

reference state sequence \( S \)

Margin

Most competing state sequence \( \tilde{S} \)

How to train the last layer \( w_{SVM} \)?

1st-step: fixed \( w_{LSTM} \), \( F(w_{SVM}, w_{LSTM}) \) is convex,
training \( w_{SVM} \) \( \leftrightarrow \) the structured SVM

How to train the previous layers \( w_{LSTM} \)?

2nd-step: fixed \( w_{SVM} \), \( F(w_{SVM}, w_{LSTM}) \) is non-differentiable,
training \( w_{LSTM} \) requires subgradients

4. Experiments & Conclusion

- Training data: 60 hours of US-English Windows Phone Short Message Dictation
- Testing data: 3 hours of data from same task

Model | WER (%)
---|---
6-layer DNN (MMI training) | 21.1
4-layer LSTM (MMI training) | 20.8
Recurrent SVM (Max Margin training) | 19.8

- 4.8% WERR. More results in the paper.

Model | WER (%)
---|---
Recurrent SVM (only train last layer) | 20.2
Recurrent SVM (+ previous layer) | 19.8

- 65% gains are from updating the LSTM layers

References