Neural Networks Leverage Corpus-wide Information for Part-of-speech Tagging

Yuta Tsuboi <yutat@jp.ibm.com>
IBM Research – Tokyo

Overview

- Using a feature combination of
  - local context information and
  - *corpus-wide* information
- State-of-the-art POS tagging accuracies
  - PTB-WSJ: 97.51% (ours) vs. 97.50% (Søggard, 2011)
  - CoNLL2009: 98.02% (ours) vs. 97.84% (Bohnet and Nivre, 2012)

Four types of corpus-wide features

- Word embeddings (w2v and glv)
  - word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)
- POS tag distribution (pos)
  - Pr(pos | w_t); Pr(pos | $affix_t$); Pr(pos | $spelling_t$)
- Supertag distribution (stag)
  - Pr(stag | w_t); Supertags are dependency labels and directions of parent/children, e.g. "nn/L" (Ouchi et al., 2014)
- Context word distribution(cw)
  - Pr(w_{t-1} | w_t); Pr(w_{t+1} | w_t); (Schnabel and Sch¨utze, 2014)

Activation Functions

- Let **v** be a linear filter: $\mathbf{v} = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}$
- **Rectified Linear Units (ReLUs)**

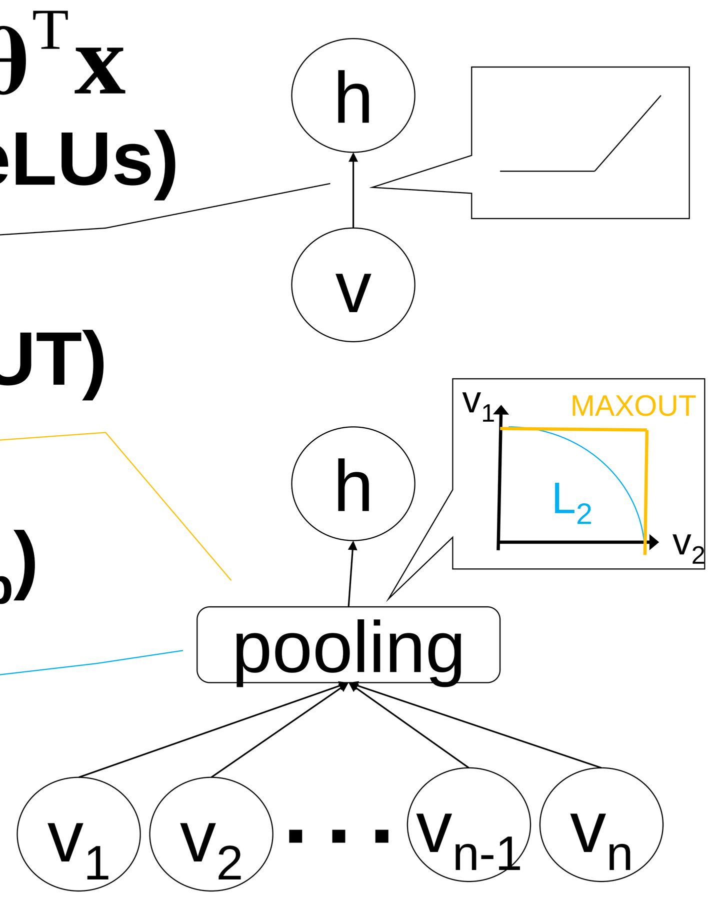
$$h = \max(v, 0)$$

- **Maxout networks (MAXOUT)**

$$h = \max(v_1, v_2, \ldots, v_n)$$

- **Normalized L_p pooling (L_p)**

$$h = \left(\frac{1}{G}\sum_{j=1}^{G} |v_j|^p\right)^{1/p}$$



Results on Penn Treebank (PTB-WSJ)

- Evaluation of the hybrid model

Neural Network Settings			Development Set		Test Set	
Activation functions	**#Hidden**	**Group size (G)**	**All**	**Unk.**	**All**	**Unk.**
Linear model	-	-	97.45	90.22	97.46	91.39
ReLUs	384	1	97.45	90.87	97.42	91.04
$L_p(p=2)$	48	8	**97.52**	90.91	**97.51**	91.64
$L_p(p=3)$	32	8	97.51	90.91	**97.51**	91.53
MAXOUT	48	8	97.50	90.89	97.50	**91.67**
$L_p(p=2)$ (w/o linear part)	48	8	97.39	**91.18**	97.40	91.23

Feature engineering using linear model

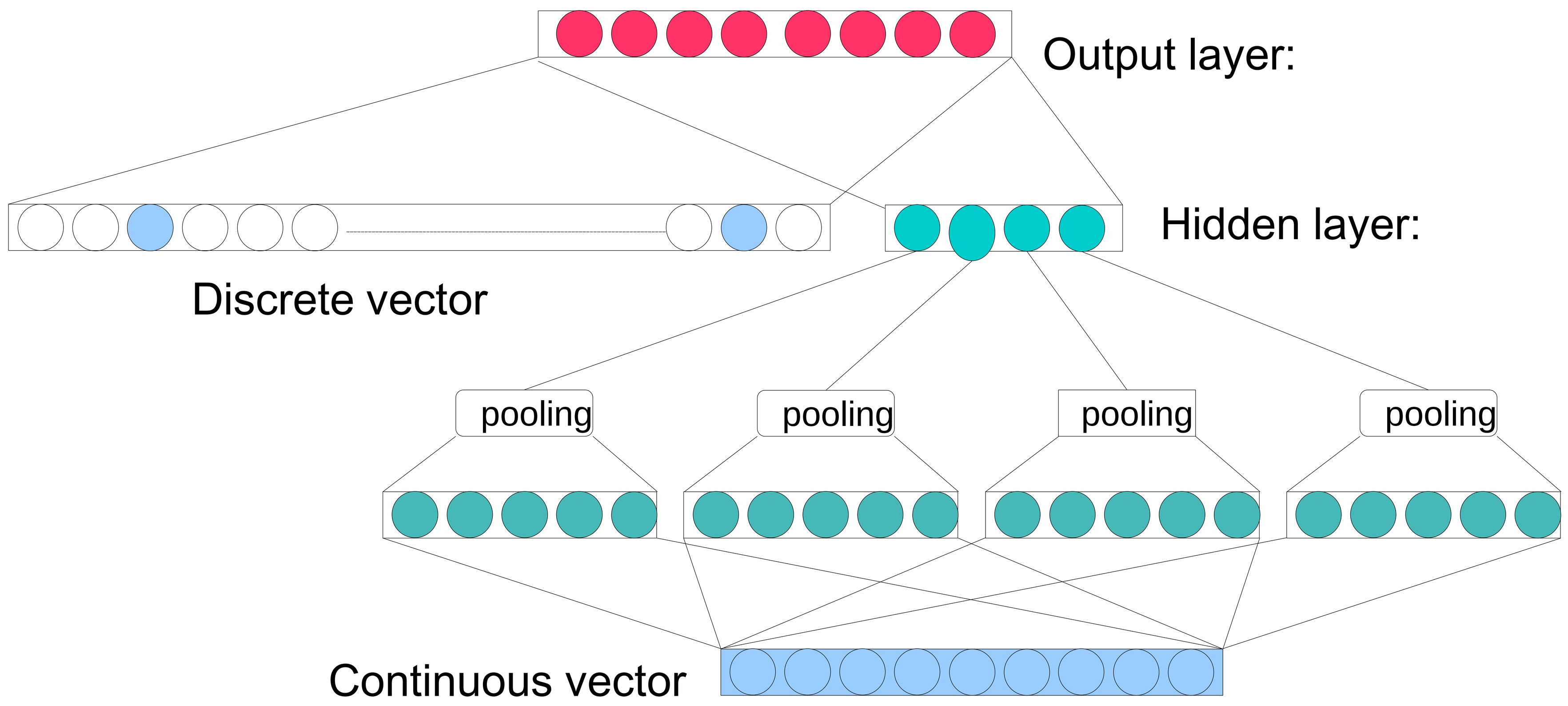
- Evaluation results of corpus-wide features on dev. set.

All token accuracy (%). / Unknown token accuracy (%)

Binary feature only: 97.15 / 86.81
- +w2v → 97.36 / 88.96
- +glv → 97.34 / 89.55
- +w2v+glv → 97.40 / 90.44
  - +pos +cw → 97.44 / 90.17
    - +stag(w=1) → 97.44 / **90.53**
    - +stag(w=3) → **97.45** / 90.22

A hybrid architecture

- Linear model for **local context features**, e.g. the neighborhood of the target word
  - Sparse discrete vectors
- Neural nets for **corpus-wide features**, e.g. the distribution of neighbor words
  - Dense continuous vectors



Why neural net. for continuous features?

- The non-linearity of discrete features has been exploited by the simple conjunction of the discrete features.
- In contrast, the non-linear feature design of continuous features is **not intuitive.**

Online learning of a left-to-right tagger

- Deterministically predicts each tag using prediction history (Choi and Palmer, 2012)
  - Binary features: N-grams, affix, spelling types, etc.
- A variant of the on-the-fly example generation algorithm (Goldberg and Nivre, 2012)
  - Using the prediction of the previously learned model as prediction history to overcome error propagation.
- FTRLProximal algorithm (McMahan, 2011) with Adagrad (Duchi et al., 2010)
  - Multi-class hinge loss + L1/L2 regularization terms
- Random hyper-parameter searches (Bergstra and Bengio, 2012)
  - Initial weights; initial weight range; momentum; learning rate; regularization, epoch to start the regularizations, etc. (256 initial weights are tried!)

Learned representations

Scatter plots of verbs for all combinations between the first 4 principal components of the raw features and the activation of hidden variables.