品詞タグ付けニューラルネットワークの深層化

坪井 祐太
日本アイ・ビー・エム株式会社
東京基礎研究所
yutat@jp.ibm.com

1 はじめに

品詞タグ付けは自然言語処理において基盤となる処理であり，解析精度だけでなく解析速度やメモリ使用量も広く使われるために重要な要素である．一方，英語品詞タグ付けのベンチマークデータで現在最高精度を示しているニューラルネットワークに基づく品詞タグ付け手法 [15] は，高精度だが解析速度が遅くメモリ使用量も多い課題があった．本研究では，ニューラルネットワークを深層化することでパラメータ数を減少させ解析を高速化した．

2 節でニューラルネットワークに基づく品詞タグ付けモデルを導入し，3 節で深層化方法について述べる．4 節で実験設定を説明し，5 節で結果を示す．

2 ニューラルネットワークによる品詞タグ付け

文献 [15] は，a) 離散値疎ベクトルを入力とした線形モデルと b) 連続値密ベクトルを入力としたニューラルネットワークを組み合わせた品詞タグ付けモデルを提案した．離散値疎ベクトルが品詞予測対象の単語の周辺の局所的な情報を表現するのに対して，連続値密ベクトルはコーパス全体の統計情報を反映するように設計されている．離散値疎ベクトルには単語 N グラムなど文献 [2] に定義されている特徴量を用いている．一方，連続値密ベクトルには窓幅中の単語のコーパス全体から計算される次の 4 つの値を用いている．

1. 埋め込みベクトル: *word2vec* [10] と *GloVe* [12] を用いて単語を低次元空間に射影した連続値ベクトル表現．
2. 品詞分布: 単語や接頭・接尾・文字種に対するアノテーションされた品詞の分布．
3. スーパータグ分布: 単語のアノテーションされた係り先・係り元の方向と係り受けラベル [11] の分布．
4. 隣接語分布: 単語の左右に隣接する単語の分布．ただし，隣接語は頻度上位 500 単語のみを使用

なお，品詞分布やスーパータグ分布は訓練データを用いてあらかじめ計算できるため，解析時には品詞や係り受けは必要ない．また，品詞分布・スーパータグ分布・隣接語分布は加算スムージングを使用しているため，未観測な値があっても密ベクトル表現となる．

離散値は 2 値ベクトルで表現される事が多い．その積で組み合わせ特徴量を表せば共起を自然に表現できるため，組み合わせ特徴量は直感的で設計が比較的容易である．また，疎ベクトルであるため低頻度の組み合わせを省くための頻度閾値などを設けることで組み合わせ特徴次元の増大を防ぐ事が可能であった．

一方，連続値同士の組み合わせを表現する特徴量は離散値の組み合わせ特徴量にくらべて直感的でなく設計が困難である．また密ベクトルであるため値を持たない組み合わせが存在せず，頻度による削減ができない．そのため連続値密ベクトルに対してはニューラルネットワークを用いて高次の特徴量を学習することが有効である．

使われているニューラルネットワークはフィードフォーワード型で，プーリング (pooling) 型の活性化関数 (activation function) が使われている点が特徴的である．プーリング操作は隠れ変数推定のばらつきを減らす事が知られている [1]．階層 l の活性化関数の入力ベクトル $\boldsymbol{v}^l \in \mathbb{R}^{|V^l|}$ は，$|V^l|$ 個の変数を G^l 個毎のグループに分けてあり，i 番目のグループの j 番目の要素を $\left\{v^l_{ij} \,|\, 1 \leq i \leq |V^l|/G^l \wedge 1 \leq j \leq G^l\right\}$ とする．$v^l_{ij} = {\boldsymbol{\theta}^l_{ij}}^\top \boldsymbol{h}^{l-1}$ は階層 l のためのパラメータベクトル $\boldsymbol{\theta}^l_{ij}$ と，一つ下の階層 $l-1$ の $|H^{l-1}|$ 次元の隠れ変数ベクトル $\boldsymbol{h}^{l-1} \in \mathbb{R}^{|H^{l-1}|}$ の内積によって計算される値である．プーリング型活性化関数は次の 2 種類が

使われている．

1. 正規化 L_p プーリング (L_p) [7]:

$$h_i = \left(\frac{1}{G} \sum_{j=1}^{G} |v_{ij}|^p \right)^{\frac{1}{p}} \text{ for all } \left\{ i \,\middle|\, 1 \le i \le \frac{|V|}{G} \right\}.$$

2. 最大値出力ネットワーク (MAXOUT) [6]:

$$h_i = \max_{1 \le j \le G} v_{ij} \text{ for all } \left\{ i \,\middle|\, 1 \le i \le \frac{|V|}{G} \right\}.$$

なお，$v_{ij} > 0$ を仮定すると MAXOUT は $p = \infty$ とした L_p の特殊形といえる．

線形モデルとニューラルネットワークの統合モデルの学習はマルチクラスヒンジ損失関数 [3] に L_1 と L_2 の正則化を加えた関数を目的関数としている．目的関数の最小化にはオンライン学習アルゴリズムである *Follow the Proximally Regularized Leader* [9] を用いている．またニューラルネットワークの学習にはバックプロパゲーションを使用して微分を計算している．

3　ニューラルネットワークの深層化

文献 [15] では隠れ層は 1 層のみの浅いニューラルネットワークであったが， 本研究では図 1 に示すように隠れ層を 2 階層に拡張する．深層化の目的は，各層の隠れ変数の数を減らして全体のパラメータ数を減らすことによるメモリ使用量の削減と解析の高速化である．なお，ニューラルネットワークの計算量はパラメータ数に比例するため，パラメータ数削減によって解析の高速化が期待できる．

理論的に，隠れ層を一つしか持たないニューラルネットワークにとって指数的なパラメータ数が必要になる関数を，深い隠れ層を使うことで線形のパラメータ数で表現できることが示されている [4]．つまり，1 層のみに比べてのニューラルネットワークの深層化によって全体のパラメータ数の削減が期待できる．特に，文献 [15] で用いている入力ベクトルの次元が相対的に高いため，1 層目の隠れ変数の数 $|H^1|$ と活性化関数のグループ数 G^1 を減らす事がパラメータ数削減に有効である．

本研究では，1 層目の活性化関数には文献 [15] で最も精度の高かった $L_p (p = 2)$ を使用し，2 層目の活性化関数には訓練時の微分計算が速い MAXOUT を用いた．また，深層化による訓練データへの過学習を避けるためにドロップアウト (Dropout) [14] を導入した [1]．ドロップアウトは訓練時にドロップアウト確率 p^{l-1} で h_i^{l-1} の値を 0 に置換する．ドロップアウト使用時の v_{ij} の計算は次式で行う：

$$\begin{aligned} r_i^{l-1} &\sim \text{Bernoulli}(1 - p^{l-1}), \\ \tilde{\boldsymbol{h}}^{l-1} &= \boldsymbol{h}^{l-1} \circ \boldsymbol{r}^{l-1}, \\ v_{ij}^{l} &= {\boldsymbol{\theta}_{ij}^{l}}^{\top} \tilde{\boldsymbol{h}}^{l-1}, \end{aligned}$$

ただし，$\circ$ は要素積である．確率的に一部の隠れ変数を無効化することで，訓練中には毎回異なるネットワーク構造を評価していることに相当し，ある種の正則化の効果がある事が知られている．なお，予測時のパラメータは $(1-p)\boldsymbol{\theta}$ のように調整することで，異なる複数のネットワーク構造を平均して使う効果がある．

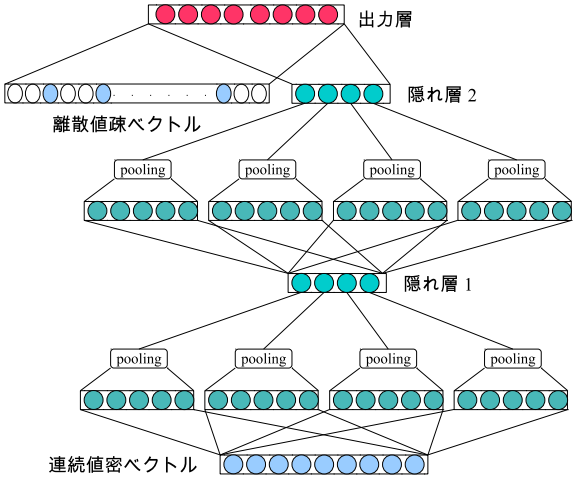

図 1: 線形モデルと隠れ層が 2 層のニューラルネットワークの統合構成

4　実験設定

文献 [15] と同じく，Penn ツリーバンク [8] に対して標準的な分割を実施して，訓練セット (0–18)・開発セット (19–21)・テストセット (22–24) として使用した．評価指標には全単語および未知語の精度を使用した．なお，未知語は訓練セットに出現していない単語を示す．

離散値疎ベクトルは文献 [2] で定義されているものを使用し，連続値密ベクトルのための各特徴量の構成方法や窓幅は文献 [15] の実験設定に従った．

ニューラルネットワークの階層が増えることによってハイパーパラメータ（隠れ変数の数・活性化関数のグループ数・ドロップアウト確率）が増える．なお，

[1] 隠れ層が 1 層だけのニューラルネットワークではドロップアウトの導入効果は見られなかった．

	#隠れ変数	#グループ	ドロップアウト確率	開発 全単語	開発 未知語	テスト 全単語	テスト 未知語	#パラメータ	解析速度	試行回数
文献 [15]	48	8	0, 0	**97.52**	90.91	**97.51**	**91.64**	205 万	1	2
提案法	16, 64	8, 8	0.2, 0, 0.5	**97.52**	**91.07**	**97.51**	91.45	69 万	0.25	60

表 2: 開発・テストセットでの全単語と未知語の精度 (%) : #隠れ変数，#グループ，ドロップアウト確率は下層から上層の順でカンマ区切りで表示した． #パラメータはニューラルネットワーク部分のパラメータ数，解析速度は文献 [15] の開発セットでの解析時間を 1 としたときの比率，試行回数はハイパーパラメータ選択の回数である．

変数	層	候補
隠れ変数の数	1	{4, 8, 16}
	2	{8, 16, 32, 64, 128}
活性化関数の	1	{4, 8, 16}
グループ数	2	{4, 8, 16, 32}
ドロップアウト確率	0	{0, 0.2, 0.4}
	1	{0, 0.5}
	2	{0, 0.5}

表 1: ハイパーパラメータの選択候補: ドロップアウト確率の 0 層は入力層を示す.

ハイパーパラメータは訓練の前に決定する必要のあるパラメータである．本研究では，各ハイパーパラメータの候補値をあらかじめ決めて，候補値の任意の組み合わせでの訓練結果を開発セットで評価した値を確認しながら，次に試す組み合わせを決定した．表 1 はハイパーパラメータの選択候補を示す．なお，パラメータ数の削減が目的であるため，パラメータ数に大きな影響のある第 1 層の隠れ変数の数の上限は文献 [15] で使われた値より少ない値のみを候補とした．上記のハイパーパラメータの組み合わせを設定した下では，文献 [15] と同様にその他のハイパーパラメータはランダムに候補を生成し開発セットを使用して選択した．

5 実験結果

開発セットの中で最高の全単語精度を示したハイパーパラメータの組み合わせを表 2 に示す．なお，全単語精度が同じ値の設定が複数あったため，それらの中から未知語精度が最も高い結果だけを示した．

深層化したモデルは隠れ層が一層だけの文献 [15] と同程度の精度が達成できた．また，ニューラルネットワークのパラメータ数およびタグ付け速度は文献 [15] の約 30%および 25%にそれぞれ改善した．ハイパーパラメータ選択の試行回数は多く必要で全体の学習時間は増えたが，同程度の精度かつコンパクトで高速な解析器が実現できた．

深層化結果で興味深いのは，高次元の入力ベクトル (5342 次元) を隠れ層 1 層目ではたった 16 個の隠れ変数だけで表現していることである．隠れ層 1 層目が表現する空間を 2 層目がより細かく分割することで，複雑な言語現象の表現が可能になったと考えられる．

深層化したモデルは 1 層目の隠れ変数が非常に少なく，既知語に対して隠れ層の 1 層目を計算しておくことにより解析時の計算時間とメモリ使用量をさらに減少できると考えられる [5]．深層化したモデルでは $384(= 16(\text{隠れ変数}) \times 8(\text{グループ数}) \times 3(\text{窓幅}))$ 次元の値を保持しておくことで，既知語の 1 層目の計算が不要になる．また入力変数として使用している単語埋め込みベクトル (600 次元) や隣接語分布 (1004 次元) などを既知語に対して保持する必要が無くなるため解析時のメモリ使用量を減らす事も可能になる．

6 おわりに

深層化によるニューラルネットワークに基づく品詞タグ付けの高速化・省メモリ化を提案した．

本研究ではハイパーパラメータ群は開発セットで測定した精度に基づいて選択したが，ハイパーパラメータ数に対してハイパーパラメータの組み合わせは指数的に増えるため，更なる深層化を実施した場合にグリッドサーチなどのしらみつぶしの探索方法は現実的ではなくなる．特にニューラルネットワークの学習には時間がかかるため，ベイズ的最適化 [13] などよりハイパーパラメータの効率的な探索方法が望まれる．

参考文献

[1] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 111–118, 2010.

[2] Jinho D. Choi and Martha Palmer. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (ACL)*, pp. 363–367, 2012.

[3] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, Vol. 2, pp. 265–292, 2001.

[4] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 666–674, 2011.

[5] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1370–1380, 2014.

[6] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C. Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1319–1327, 2013.

[7] Caglar Gulcehre, Kyunghyun Cho, Razvan Pascanu, and Yoshua Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2014.

[8] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The Penn treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[9] H. Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 525–533, 2011.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pp. 3111–3119, 2013.

[11] Hiroki Ouchi, Kevin Duh, and Yuji Matsumoto. Improving dependency parsers with supertags. In *Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 154–158, 2014.

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: global vectors for word representation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[13] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, pp. 1929–1958, 2014.

[15] Yuta Tsuboi. Neural networks leverage corpus-wide information for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 938–950, 2014.