擬似タグと線形移動ベクトルを用いた単一モデルによる擬似モデルアンサンブル

桑原 亮介 [1]　鈴木 潤 [2]　中山 英樹 [1]
[1] 東京大学 大学院情報理工学系研究科　[2] 東北大学 大学院情報科学研究科
{kuwabara, nakayama}@nlab.ci.i.u-tokyo.ac.jp
jun.suzuki@ecei.tohoku.ac.jp

1 序論

深層ニューラルネットワークにおけるモデルアンサンブルは，同一の訓練データを使用し，乱数シードなどを変更して独立に訓練された複数モデルの出力を組み合わせることで性能を向上させる広く知られた方法論である [1, 2]．例えば，自然言語処理研究分野で取り組まれている様々な評価型ワークショップやシェアードタスクなどのコンペティションで上位を獲得するシステムでは，モデルアンサンブルを用いないシステムはないという程，当たり前の技術として利用されている [3, 4]．

一方，モデルアンサンブルは複数モデルを使用するため，訓練と推論に多くの時間とメモリを要し，ハイパーパラメーターなどの管理に手間がかかるという課題がある．特に，推論時には複数のモデルを一つのシステムで同時に利用することになるため，それらのモデルの管理や計算のコストが増大が，実システムにおいては時に大きな問題となりえる．例えば，エッジデバイスなどの限られた計算リソースで動作することが求められるシステムを考えた場合，利用可能なメモリ量などが限定されるという観点で，モデルアンサンブルの方法論を利用するのが困難となる．

これらの課題を克服することを目的とし，本研究では モデルアンサンブルの効果を単一のモデルで再現する手法を提案する．通常のモデルアンサンブルが個々に独立した空間で学習した複数モデルを集約するのに対し，提案手法では単一空間内で複数のモデルを仮想的に作成しそれを集約する．具体的には，仮想モデルを K 個作成することを想定した場合，K 倍に拡大した訓練データの全てに K 個の疑似タグを付与する．各疑似タグ $k \in \{1, \ldots, K\}$ は k 番目の仮想モデルに対応し入力文の先頭に付加される．また，それぞれの疑似タグに対応する K 個のベクトル（以後このベクトルを線形移動ベクトルと呼ぶ）をそれぞれのサンプルの入力に相当する単語埋め込みベクトル（word embedding）に加算する．直感的にはこの操作は，同じデータの単語埋め込みベクトルを k 番目に指定した空間内の局所領域にシフトすることに相当する．よって，このような操作により，単一空間内に K 個の仮想モデルを明示的に作成するものと解釈ができる．したがって，単一のモデルから生成された K 個の仮想モデルを使用して，それぞれの出力を集約することで K 個のモデルで構成されるモデルアンサンブルと同様の効果が得られることが期待される．

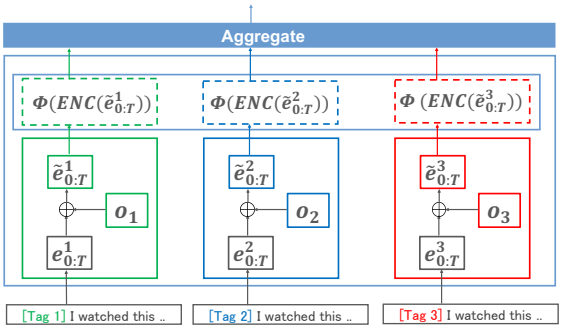

図 1: 提案法の概要．単一モデルで先頭に擬似タグのついた同一データを処理する．それぞれの擬似タグは k 番目の仮想モデルを定義し，対応する線形移動ベクトル $\boldsymbol{o}_k$ を単語埋め込みベクトルに足すことで，共通の関数 $\phi(\mathrm{ENC}(\cdot))$ を用いた場合にも別の出力が得られる設計としている．

本稿では，自然言語処理の典型的なタスクであるテキスト分類と系列ラベリングの標準的なベンチマークデータを用いて提案法の有効性を検証する．また，実験により，提案法がアンサンブルと同様に，特定のタスクに依存せず通常の単一モデルの性能を向上できることを示す．加えて，一部のデータセットにおいては $1/K$ 倍のパラメタ数で通常のアンサンブルに比肩する，または凌駕する性能を達成することができることを示す．

2 関連研究

ニューラルネットワークにおけるアンサンブルは，古くから研究されている方法論の一つである [1, 2, 5]．一方，アンサンブルを用いることで発生する追加の様々なコスト（計算コスト，メモリ，管理コスト）に関しては，これまであまり多く議論されてこなかった．しかし，近年発表されたいくつかの論文では，従来のアン

サンブルの欠点であるこれらのコストを部分的に克服する方法論も提案されている．例えば，スナップショットアンサンブル [6] では，最適化パスに沿った複数の局所解に収束させることにより，単一のモデルを使用して複数のモデルを構築する．文献 [7] の蒸留による方法論では，アンサンブルモデルの知識を単一のモデルに伝承できることを示した．

本稿の提案法で用いる擬似タグを活用するという考え方は，同一の入力に対して何らかの条件付けなどを行う一般的な手法でおり，本研究はこの考え方に基づく（例：[8, 9]）. 我々の知る限り，擬似タグを仮想モデルの識別マーカーとして単一モデルに組み込む手法は本手法が初である．

本研究の考え方に類似する研究はドロップアウト [10] であると考える. ドロップアウトは，訓練中各ミニバッチ中の隠れユニットを確率的に取り除き，推論時にはすべてのユニットが使用される. Gao ら [6] は，ドロップアウトを，同一空間内で多数の仮想モデルを暗黙的に使用するものとして説明している. 提案手法では，同一空間内で複数の仮想モデルを明示的に使用する．実験の章 5 で述べる通り，ドロップアウトと相補的な関係にある．

3 基本モデル

本稿で提案する方法は特定のタスクを仮定しない一般的な方法論であるが，自然言語処理分野で広く用いられているテキスト分類と系列ラベリングで近年よく用いられているモデルを対象として議論をおこなう．

ここで，入力はトークン列（文）を仮定する．$\boldsymbol{x}_t$ を t 番目のトークンのワンホットベクトルとする．$\boldsymbol{E} \in \mathbb{R}^{D \times |\mathcal{V}|}$ を単語埋め込みベクトルの行列とする．ここで，D が単語埋め込みベクトルの次元，$\mathcal{V}$ が入力の語彙数である．入力の t 番目の単語埋め込みベクトルを以下で表す．

$$\boldsymbol{e}_t = \boldsymbol{E}\boldsymbol{x}_t. \tag{1}$$

ここで，$\boldsymbol{e}_{1:T}$ を $(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_T)$ 長さ T の入力文に対応するベクトルのリストとする．$\boldsymbol{e}_{1:T}$ に基づき，$t \in \{1, \ldots, T\}$ のそれぞれに対応する特徴（隠れ）ベクトル $\boldsymbol{h}_t \in \mathbb{R}^H$ がエンコーダニューラルネットワークによって計算される．また H は特徴ベクトルの次元である. したがって特徴ベクトル $\boldsymbol{h}_t$ は下記のように表すことができる．

$$\boldsymbol{h}_{1:T} = \mathrm{ENC}\left(\boldsymbol{e}_{1:T}\right). \tag{2}$$

ここで，ENC はエンコーダニューラルネットワークを表す関数である．

そして入力 $\boldsymbol{x}_{1:T}$ に基づく出力 $\widehat{\boldsymbol{y}}$ は以下の式で推定される．

$$\widehat{\boldsymbol{y}} = \phi\left(\boldsymbol{h}_{1:T}\right). \tag{3}$$

ここで，$\phi(\cdot)$ はタスク固有の関数である．例えば．テキスト分類ではソフトマックス関数，系列ラベリングでは条件付き確率場などが用いられる．

4 擬似タグと線形移動ベクトルを用いた擬似アンサンブル

本章では提案手法である SINGLEENS を説明する．図 1 に提案法の直感的な概要図を示す．提案法の基本的な考え方は複数の仮想モデルを単一モデル内部に作成することである．

そこで，提案法では**擬似タグ** および **線形移動ベクトル**を用いる．擬似タグは，入力語彙にスペシャルトークン $\{\boldsymbol{\ell}_k\}_{k=1}^K$ として追加され，単一モデル内部の k 番目の仮想モデルを定義する．K は仮想モデルの個数を示すハイパーパラメーターである．線形移動ベクトルでは，互いに直交するベクトルを用いる．ここで，‘互いに直交する’ とは，$\boldsymbol{o}_k \cdot \boldsymbol{o}_{k'} \simeq 0$ を全ての (k, k'), $k \neq k'$ について満たすことをいう[1].

全ての入力系列は $\{\boldsymbol{\ell}_k\}_{k=1}^K$ のうちの 1 つの擬似タグから開始される．そして，擬似タグ $\boldsymbol{\ell}_k$ に対応した線形移動ベクトル $\boldsymbol{o}_k$ を全てのステップにおける単語埋め込みベクトルに足しこむ．したがって，提案法を反映した新しい単語埋め込みベクトルは次のように表せる．

$$\tilde{\boldsymbol{e}}_{0:T}^{(k)} = (\boldsymbol{\ell}_k, \boldsymbol{e}_1 + \boldsymbol{o}_k, \boldsymbol{e}_2 + \boldsymbol{o}_k, \ldots, \boldsymbol{e}_T + \boldsymbol{o}_k). \tag{4}$$

提案法では，ベクトル $\tilde{\boldsymbol{e}}_{0:T}^{(k)}$ を式 3 における $\boldsymbol{e}_{1:T}$ に代入する．直感的な解釈として，擬似タグの役割は，単一モデルに対して同一データにおける違いを明示的に認識させることである．また，直交ベクトルの目的は，同一データに対する単語埋め込みベクトルを同一空間内で仮想モデル毎に定義された局所領域の方向へ線形移動させることである．

このように提案法は，入力層に相当する単語埋め込みベクトルのみに特別な処理を加えるという比較的シンプルな構成になっている．しかし，本研究の目的であるモデルアンサンブルの代替法という位置付けを考えた場合に，複雑な方法ではないことが非常に重要となる．それは，シンプルであれば，近年発達が著しい深層ニューラルネットワークの他の様々な方法論と親和性高く相補的に同時に取り入れられる可能性が高くなると考えられるためである．これは，従来のモデルアンサンブルの利点と同様に，適用するモデルの構成にかかわらず，総じて効果が得られるという点につながる．

5 実験

提案法の有効性評価のため，テキスト分類と系列ラベリングの 2 つのタスクで実験を行った．データセッ

[1] 実装上 $\boldsymbol{o}_k \cdot \boldsymbol{o}_{k'} \simeq 0$ としている

データセット	モデル	手法	パラメタ数	正解率
IMDB	TFM: GLOVE	SINGLE	12 M	87.03
		1/K ENS	14 M	81.93 (−5.10)
		SINGLEENS	12 M	87.30 (+0.27)
		NORMALENS	108 M	**87.67** (+0.64)
	TFM: BERT	SINGLE	400 M	91.99
		1/K ENS	1000 M	90.63 (−1.36)
		SINGLEENS	400 M	**92.91** (+0.92)
		NORMALENS	3600 M	92.75 (+0.76)
Rotten	TFM: BERT	SINGLE	400 M	81.75
		1/K ENS	1000 M	82.67 (+0.92)
		SINGLEENS	400 M	**85.01** (+3.26)
		NORMALENS	3600 M	82.57 (+0.82)
RCV1	TFM: BERT	SINGLE	400 M	87.18
		1/K ENS	1000 M	80.27 (−6.91)
		SINGLEENS	400 M	89.16 (+1.98)
		NORMALENS	3600 M	**90.01** (+2.83)

表 1: テキスト分類における正解率とパラメーターサイズ．提案法の **SINGLEENS** は，SINGLE と 1/K ENS の性能を超えるだけでなく，IMDB および Rotten データセットでは NORMALENS も超えた．

トとして，テキスト分類には IMDB [11], Rotten [12], そして RCV1 [13]，系列ラベリングには CoNLL-2003 [14] および CoNLL-2000 [15] を使用した．

基本モデルとして Transformer[16] を全ての実験において用いた．また，それぞれの基本モデルは，GloVe[17], BERT[18] または ELMo [19] のいずれかの事前学習された単語ベクトルを使用している．それぞれのモデルは以後，TFM:GLOVE, TFM:BERT[2], そして TFM:ELMO と表す. すべてのモデルは，ドロップアウトとの相補性を確認するため，ドロップアウト層を備えている．

本実験では，提案法 (**SINGLEENS**) を，単一モデル (**SINGLE**)，通常のアンサンブル (**NORMALENS**)，そしてアンサンブルを構成する単一モデルのパラメーター数が $1/K$ であるアンサンブル (**1/K ENS**) と比較をした．NORMALENS，1/K ENS，および SINGLEENS については，$K = 9$ を使用した．擬似タグは他の単語埋め込みベクトルと同じ訓練，初期化方法で 9 つ $\{\boldsymbol{\ell}_k\}_{k=1}^{9}$ 用意し，学習で値の変化しない固定の線形移動ベクトル $\{\boldsymbol{o}_k\}_{k=1}^{9}$ を Andrew らの手法 [20] により作成した．アンサンブルの集約方法については，テキスト分類と系列ラベリングのそれぞれにおいて，平均化，Voting を用いた．全ての実験において異なるランダムシードを使用した 5 つの結果の平均を示す．

5.1 テキスト分類における評価

データ　データの分割方法は清野ら [21] の実装に従った．提案法である SINGLEENS はデータを K 倍に拡大させる. また，データを拡大させる際，k 番目のサブセットには対応する k 番目の擬似タグを付与するとともに，ブートストラップサンプリングによりデータをサンプリングする．NORMALENS および 1/K ENS については，ブートストラップサンプリングと通常のサンプリングを用い，よりスコアの高いものを最終スコアとした．

結果　表 1 に本実験結果を記載する．TFM:GLOVE と TFM:BERT の両方の場合について，SINGLEENS は SINGLE の性能を超えた．さらに，IMDB と Rotten において，それぞれ 92.91% と 85.01% という精度を記録し，パラメーター数を約 1/9 に抑えつつ通常のアンサンブルの精度を 0.16% および 2.44% 上回った．実験結果により，K 個の仮想モデルを明示的に作成する提案法の効果が優位な影響を与えると考えることができる．

[2]詳細な実装は [22] を参照.

データセット	モデル	手法	パラメタ数	F1 スコア
CoNLL 2003	TFM: ELMO	SINGLE	100 M	91.93
		1/K ENS	150 M	91.65 (−0.28)
		SINGLEENS	100 M	92.37 (+0.44)
		NORMALENS	900 M	**92.86** (+0.93)
CoNLL 2000	TFM: ELMO	SINGLE	100 M	96.42
		1/K ENS	150 M	95.67 (−0.75)
		SINGLEENS	100 M	96.56 (+0.14)
		NORMALENS	900 M	**96.67** (+0.25)

表 2: 系列ラベリングタスクにおける F1 スコアおよびパラメーターサイズ．提案法である **SINGLEENS** が，SINGLE の性能より優っている．

5.2 系列ラベリングにおける評価

データ　基本的な実験設定は CoNLL-2000 および 2003 に基づいている. SINGLEENS についてはデータを 9 倍に拡大し NORMALENS および 1/K ENS については通常のサンプリング手法を適用した．

結果　表 2 に示す通り， TFM:ELMO における SINGLEENS が SINGLE をそれぞれのデータセットにおいて，0.44%， 0.14% ずつ上回った．一方で，本タスクにおいては NORMALENS が最高性能となった．

6 分析

本章では，提案法の挙動について分析を行った．分析においては，特に言及のない限り，TFM:BERT および TFM:ELMO を IMDB と CoNLL-2003 のそれぞれに対して適用した．

提案法は単なるデータ拡張またはランダムノイズと等価であるか？　擬似タグと線形移動ベクトルの両方を使うことの重要性を検証するために，1) タグのみ, 2) 線形移動ベクトルのみ，3) ランダムノイズ の 3 つを用いた手法との比較を行った．1) はタグに対応した線形移動ベクトルを用いずに，擬似タグのみをデータの先頭につけた 2) では，擬似タグの番号に対応しない線形移動ベクトルをランダムに単語埋め込みベクトルに加えており，3) は線形移動ベクトルの代替と考え得るランダムノイズを単語埋め込みベクトルに加えた手

手法	IMDB 正解率	CoNLL-2003 F1 スコア
SINGLE	91.99	91.93
タグのみ	89.84	92.20
線形移動ベクトルのみ	92.06	92.21
ランダムノイズ	92.38	92.32
タグ + 線形移動ベクトル	**92.91**	**92.37**

表 3: 提案法（擬似タグ+線形移動ベクトル）とその他手法の比較.

手法	IMDB 正解率	CoNLL-2003 F1 スコア
SINGLE	91.99	91.93
単語埋め込みベクトルのみ	**92.91**	92.37
Hidden のみ	90.68	**92.45**
単語埋め込みベクトル + Hidden	92.64	92.19

表 4: IMDB および CoNLL-2003 における線形移動ベクトルを足した場所ごとのテストメトリクス.

法である.

表 3 が示す通り，擬似タグと線形移動ベクトルの両方を使う手法において最高性能を確認することができた．片方のみを使う手法では，むしろ性能が下がることも確認できた．したがって，擬似タグと線形移動ベクトルの両方を用いることは単なるデータ拡張やランダムノイズとは本質的な違いがあると考えることができる．また，この結果がドロップアウトに対する補完性を説明できる．ドロップアウトはランダムにモデル内部のサブネットワークを使用しているため，明示的な仮想モデルの指定を行なっていない．提案法では，擬似タグと個別のベクトルを用いて仮想モデルの定義と入力に変化を加えているため，仮想モデルの生成に明示的な操作を加えている．したがってこの違いが両手法における相補性を保つことに寄与していると考えられる.

どこにベクトルを足すべきか　提案法では単語埋め込みベクトルのみに線形移動ベクトルを足したが，その他の箇所を加えた場合についても結果を検証した．表 4 が示す通り，単語埋め込みベクトルのみに足す手法が一番性能が高く，むしろ他の箇所に加えた場合には性能が下がるケースも確認された．これは本実験で用いた Transformer のアーキテクチャが関係するものと思われる．Transformer は残差ネットワーク [23] を用いているため，単語埋め込みベクトルに対する小さな違いでもモデル内部に再帰的に影響を及ぼすことができる．したがって単語埋め込みベクトルに対する小さな入力な違いも最終的な出力に大きな影響を与えることができると考える.

7　結論

本研究では単一モデルで擬似的なアンサンブルを行う手法 SINGLEENS を提案した．SINGLEENS の考え方は，同一空間中に複数の仮想モデルを明示的に生成することである．テキスト分類と系列ラベリングタスクの実験により，提案法が単一モデルの性能を上回ることを確認できた．さらに，TFM:BERT を用いた実験では，パラメーターサイズを $1/K$ に抑えつつ，IMDB と Rotten のデータセットにおいて，提案法が通常のアンサンブルの性能を上回った．本手法は今回実験で用いたタスクに限らず，その他 NLP タスクや画像タスクなどにも応用可能である.

謝辞　本研究は JSPS 科研費 19H04162 の助成を受けたものです．実験データの準備に関してご協力をいただきました Preferred Networks の佐藤元紀氏および理化学研究所の清野舜氏へ，記して感謝いたします.

参考文献

[1] Hansen.L. K and Salamon, P. Neural Network Ensembles. IEEE Trans. Pattern Anal. Mach. Intell., 1990.

[2] Krogh. A and Vedelsby. J Neural Network Ensembles, Cross Validation and Active Learning In *NeurIPS*, 1994.

[3] Bojar. O, Federmann. C, Fishel. M, Graham. Y, Haddow. B, Koehn. P and Monz. C Findings of the 2018 Conference on Machine Translation (WMT18). In *WMT*, 2018.

[4] Barrault. L, Bojar, and others Findings of the 2019 conference on machine translation (WMT19). In *WMT*, 2019.

[5] Sherif. H, Optimal Linear Combinations of Neural Networks NEURAL NETWORKS, pages 599-614, 1994.

[6] Gao. Huang, Yixuan. L, Geoff. P, Zhuang. L, John E.H and Kilian. Q.W Snapshot Ensembles: Train 1, get M for free *arXiv*, 1704.00109., 2017.

[7] Hinton. G, Vinyals. O, and Dean. J Distilling the knowledge in a neural network *arXiv*, 1503.02531., 2015.

[8] Sennrich. R, Haddow. B and Birch. A Controlling Politeness in Neural Machine Translation via Side Constraints In *NAACL*, 2016.

[9] Johnson. M, Schuster. M, Le. Q.V, Krikun. M, Wu. Y, Chen. Z, Thorat. N, Viégas. F, Wattenberg. M, Corrado. G, Hughes. M and Dean. J Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation Transactions of the Association for Computational Linguistics, 2017.

[10] Srivastava. N, Hinton. G, Krizhevsky. A, Sutskever. I and Salakhutdinov. R Dropout: A Simple Way to Prevent Neural Networks from Overfitting Journal of Machine Learning Research, 2014.

[11] Maas. A.L, Daly. R.E, Pham. P.T, Huang. D, Ng. A.Y and Potts.C Learning Word Vectors for Sentiment Analysis In *ACL*, 2011.

[12] Pang. B and Lee. L Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales In *ACL*, 2005.

[13] Lewis. D.D, Yang. Y, Rose. T.G and Li. F RCV1: A New Benchmark Collection for Text Categorization Research J. Mach. Learn. Res., 2004.

[14] Tjong. K.S, Erik F and Buchholz.S Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition In *NAACL*, 2003.

[15] Tjong. K.S, Erik F and Buchholz.S Introduction to the CoNLL-2000 Shared Task Chunking In *CoNLL*, 2000.

[16] Ashish. V, Noam. S, Niki. P, Jakob. U, Llion. J, Aidan. N.G, Lukasz. K and Illia. P Attention Is All You Need arXiv, 1706.03762., 2017.

[17] Pennington. J, Socher. R and Manning. C Glove: Global Vectors for Word Representation In *EMNLP*, 2014.

[18] Jacob. D and Ming-Wei. C, Kenton. L and Kristina. T BERT: Pre-training of Deep Bidirectional Transformers for Language *arXiv*, 1810.04805., 2018.

[19] Peters. M, Neumann. M, Iyyer. M, Gardner. M, Clark. C, Lee. K and Zettlemoyer. L Deep Contextualized Word Representations In *NAACL*, 2018.

[20] Andrew. M.S, James. L.M and Surya. G Exact solutions to the nonlinear dynamics of learning in deep linear neural networks arXiv., 1312.6120., 2013.

[21] Kiyono. S, Suzuki .J and Inui. K Mixture of Expert/Imitator Networks: Scalable Semi-supervised Learning Framework *arXiv*, 1810.05788., 2018.

[22] Jinhua. Z, Yingce. X, Lijun. W, Di. H, Tao. Q, Wengang. Z, Houqiang. L and Tieyan. L Incorporating BERT into Neural Machine Translation In *ICLR*, 2020.

[23] Kaiming. H, Xiangyu. Z, Shaoqing. R and Jian. S Deep Residual Learning for Image Recognition arXiv, 1512.03385., 2015.