一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
PRMU2017-94(2017-10)

階層型 Knowledge Distillation による DNN のコンパクト化

西行　健太† 　山下　隆義† 　藤吉　弘亘†

† 中部大学　〒 487–8501 愛知県春日井市松本町 1200
E-mail: †nishiyuki@mprg.cs.chubu.ac.jp, ††{yamashita,hf}@cs.chubu.ac.jp

あらまし　Knowledge Distillation による DNN のコンパクト化は，Student Network の学習に中間層の情報を十分に活かせていないため，精度の低下を招く場合がある．本稿では，Teacher/Student Network の中間層に対して，下位層から順に Knowledge Distillation を行う階層型 Knowledge Distillation を提案する．中間層の Knowledge Distillation を行う際に，既存手法の Soft Target だけでなく，確率分布間の距離推定を適用することで，高精度な Student Network を生成することが可能であることを示す．CIFAR-10 を用いた評価実験により，従来手法と比較して，約 2%の精度向上を確認した．

キーワード　Knowledge Distillation, Deep Neural Network, 高速化, 省メモリ化, 確率分布間の距離推定

1. ま え が き

Deep Neural Network(DNN) は様々なコンペティションで記録を塗り替えるほどの高精度化を実現し，近年注目されている [1] [2]DNN は計算量とメモリ消費量が多いため，組み込み機器などの計算資源が制限される環境では，DNN を利用することが難しい場合がある．そのため，精度を下げることなく，DNN のモデルサイズを小さくすることで，高速化・省メモリ化を行う手法が提案されている [3]-[15]．DNN の高速化手法は，大きく分けると，下記 2 つに大別される．

（1） パラメータ削減による学習なしの高速化・省メモリ化
（2） ネットワークの再構築による学習ありの高速化・省メモリ化

学習なしの高速化・省メモリ化は，DNN が盛んになる前の 80 年代後半から研究が行われている．具体的な手法としては，重要度の低いノードを削除する Pruning [3] [4] [5] [6] [7] や，重みの共有化 [7] [8] などがある．学習なしの高速化・省メモリ化は，既存のネットワークに対して，縮小やメモリ消費の効率化を図る手法であるため，ネットワークの構成自体を大きく変更することは難しい．一方，学習ありの高速化・省メモリ化は，既存の大きなネットワーク (Teacher Network) の情報を用いて，新しく小さなネットワーク (Student Network) を再構築する手法であるため，ネットワークの構成を比較的自由に変更することが可能である．本研究では，ネットワークの構成を大きく変更することが可能な学習ありの高速化・省メモリ化を対象とする．

学習ありの高速化・省メモリ化は，学習なしの手法に比べて，比較的新しい手法である．2006 年に提案された Model Compression [9] を初めとして，2014 年に提案された Knowledge Distillation [10] [11] をきっかけとして，近年盛んに研究されている．

本研究では，Teacher Network と Student Network の中間層に対して，下位層から順に Knowledge Distillation を行う手法を提案する．また，提案手法では，Teacher Network と Student Network の中間層に対して，Knowledge Distillation を行う際に，既存手法の Soft Target だけではなく，確率分布間の距離推定手法を適用することで，より効率的な Knowledge Distillation を実現する．

2. Knowledge Distillation

高精度なネットワークを学習するためには，層を深くしたり，1 層あたりのノード数を増やす必要がある．そのため，高精度かつ小さなネットワークを初めから作成することは難しい．そのため，Knowledge Distillation [10] [11] では，Teacher Network と呼ばれる大きなネットワークを学習した後，Teacher Network の出力値などの情報をヒントとして，Student Netowork と呼ばれる小さなネットワークを学習することで，精度を下げることなく，小さなネットワークを生成する．

Hinton らは，Soft Target [10] [11] を用いて Knowledge Distillation を効果的に行い，高精度かつ小さなネットワークを作成している．図 1 のように，cow, dog, cat, car の 4 クラスの識別を行うネットワークの学習を行うとする．正解クラスが dog の場合は，上段のように教師信号 (Hard Target) は dog が 1 になり，それ以外の値は 0 となる．従って，このような教師信号を用いて，学習されたネットワークの出力値は，中段のように，dog が 0.9 と高い値になり，それ以外の値は 0 に近い値となる．しかし，ネットワークの出力値を確認すると，不正解クラスの値でも，cat は 0.1, car は 10^{-9} と値に差がある．これは dog と cat は似ているが，dog と car は似ていないことを示している．教師信号 (Hard Target) を用いた教師あり学習手法では，当然ながら，不正解クラス間の値の違いには着目していなかった．しかし，人間の学習の場合には，このような dog に対する cat

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2017 by IEICE

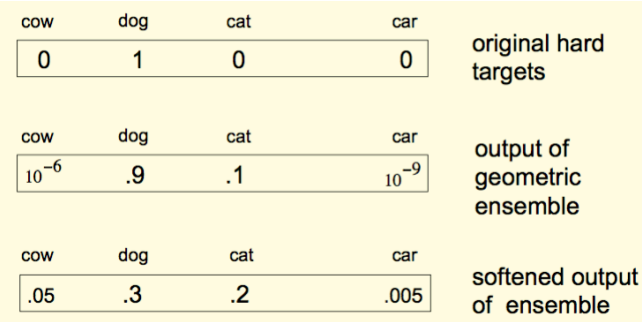

図 1　Soft Target の例 [11]

と car の違いのような不正解クラス間の値の違いを活用していると考えられる．そこで，Hinton らは，Teacher Network の出力値を用いて，Student Network を学習する際，図 1 の下段のように，Teacher Network の出力を滑らかにすることで，このような不正解クラス間の値の違いを強調した．Soft Target を Student Network の学習に使用することで，高精度な Student Network を生成している．なお，この出力値の滑らかさは温度 T というパラメータにより調整される．出力層の Softmax の計算を行う代わりに，下式を用いてネットワークの出力値を計算することで，Soft Target が計算される．

$$p_i = \frac{exp(\frac{z_i}{T})}{\sum_j exp(\frac{z_j}{T})} \tag{1}$$

Hinton らの Soft Target を用いた Knowledge Distillation は，DNN 高速化の研究に大きなインパクトを与えその後，Knowledge Distillation のように，Teacher Network の情報をヒントとして Student Network を如何に上手く学習させるかに着目した研究が提案されている．

Li らは，[12] にて，確率分布間の距離推定手法である KL-divergence を用いて，高精度かつ小さなネットワークを再構築している．Teacher Network の出力分布と，Student Network の出力分布間の距離を KL-divergence を用いて定義し，その KL-divergence を最小化するように，Student Network を学習することで，高精度に音声認識を行う事のできる小さなネットワークを生成している．

[10] [11] [12] では Teacher Network の出力だけを用いて，Student Network を学習していたため，Teacher Network の中間層の情報を，Student Network の学習に活かせているとは言えない．しかし，Teacher Network の中間層と，Student Network の中間層は幅 (パラメータ数) が異なるため，中間層に対して，Knowledge Distillation を行うことは出来ない．FitNets [13] では，Student Network の中間層の最上位に，Teacher Network の中間層と同じ幅の Regressor と呼ばれる層を新たに付け足すことで，中間層に対しても Knowledge Distilaltion を行い，Hinton らの Knowledge Distillation よりもより高精度で小さな Student Network の学習に成功している．なお，FitNets では，Teacher Network の中間層，Student Network の中間層，それぞれをまとめて 1 つのネットワークとみなして，Knowledge Distillation を行っているため，Teacher Network を構成する複数の中間層の情報全てを Student Network の学習に活かしきれているとはいえない．

Junho らは，[14] にて，複数ある中間層同士の関係を Knowledge Distillation に利用することで，FitNets [13] よりも，高精度な Student Network の学習に成功している．[14] では，複数の中間層の出力を FSP Matrix と呼ばれる行列に変換することで，中間層同士の関係を 1 つの行列として表現している．Teacher Network と Student Network それぞれで，FSP Matrix を生成し，FSP Matrix 間の L2 Loss を計算することで，中間層の情報を十分に活かした Student Network の学習を行っている．しかし，FSP Matrix 間の Loss を計算する際には，L2 距離しか用いておらず Teacher Network と Student Network の中間層間の類似度を効果的に測れているとは言えない．

3. 提 案 手 法

本研究では，Teacher Network の中間層の情報を下位層から順に Distillation を実施する階層型 Knowledge Distillation を提案する．階層型 Knowledge Distillation では，[13], [14] と同様に，中間層の Knowledge Distillation を行う．従来法では，中間層の Knowledge Distillation を行う際の Loss は L2 距離を用いているが，階層型 Knowledge Distillation では，Wassestein Loss を用いる．

3.1 階層型 Knowledge Distillation の流れ

提案手法の概要は図 2 に示す．主な手順は下記の 4 つになる．なお，階層型 Knowledge Distillation では，Teacher Network の中間層に出力層を付ける場合と，付けない場合を提案しており，出力層を付けない場合には手順 2 が省略される．

手順 1　Teacher Network の学習
手順 2　(Teacher Network の出力層の再学習)
手順 3　階層型 Knowledge Distillation を
　　用いた Student Network の学習

以降で，各手順の詳細について，説明する．

手順 1 では，Teacher Network の学習を行う．なお，Teacher Network は Student Network よりも大きな（パラメータ数の多い）Network であり，必ずしも Teacher Network と Student Network は同じ深さである必要はない．

手順 2 では，Teacher Network の中間層に，出力層を追加した後，Teacher Network の学習に使用されたものと同じデータセットを用いて再学習を行い新たに追加した出力層の重みを更新する．この再学習の際に，Teacher Network の学習で更新された中間層の重みは更新しない．また，出力層のみの重みを更新するので，学習 epoch 数は，Teacher Network を学習した際の epoch 数の半分以下の epoch 数でよい．Teacher Network の中間層に出力層を付けない場合には，手順 2 が省略される．

手順 3 にて，本研究で提案する階層型 Knowledge Distillation を用いて Student Network の学習を行う．図 3 のように，本研究で提案する階層型 Knowledge Distillation では，Teacher Network と Student Network 両方の中間層に出力層をつけて，Knowledge Distillation を行う手法と FitNets [13] のように，Student Network の中間層のみ，Teacher Network の中間層と

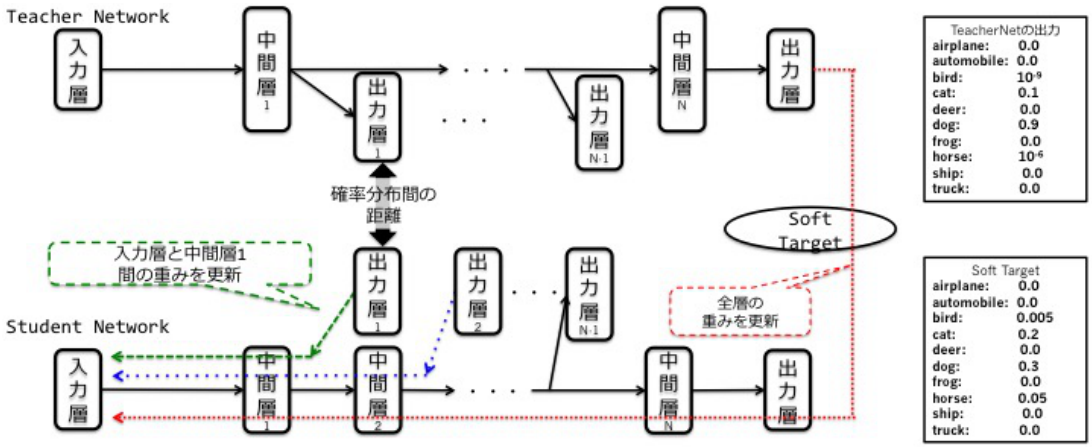

図 2　階層型 Knowledge Distillation

同じ幅の出力層をつけて，Knowledge Distillation を行う手法がある．前者では，Teacher Network の中間層に追加した出力層の重みを更新するため手順 2 の再学習が必要となるが，後者は必要ない．そのため，本研究では，以後，前者を“再学習あり”の手法，後者を“再学習なし”の手法と呼ぶ．

再学習ありの手法では，Soft Target を用いた Knowledge Distillation を行う．Hinton らの提案した Soft Target [10] [11] では式のように Student Network の出力値と，Teacher Network の出力値を滑らかにした Soft Target から計算される Loss と Student Network の出力値と，Ground Truth(Hard Target) から計算される Loss を混合する割合をパラメータ λ によって調整する．

$$P_T^\tau = softmax(\frac{a_T}{\tau}) \tag{2}$$

$$P_S^\tau = softmax(\frac{a_S}{\tau}) \tag{3}$$

$$L_{KD}(W_S) = H(y_{true}, P_S) + \lambda H(P_T^\tau, P_S^\tau) \tag{4}$$

本研究では，出力層から行われる SoftTarget を用いた Knowledge Distillation では，λ を 1.0 としている．従って，Soft Target と Hard Target が同等に重視される．また，本研究では，中間層の階層型 Knowledge Distillation を行う際，出力層の Knowledge Distillation と同様に Soft Target と Hard Target を λ=1.0 で使用する場合と，Hard Target を使用せず，Soft Target のみを使用する場合の 2 種類がある．そのため，本研究では，以後，前者を“再学習あり，Hard Target あり”，後者を“再学習あり，Hard Target なし”の手法と呼ぶ．

3.2　Wasserstein 距離の導入

再学習なしの手法では Teacher Network の中間層と Student Network の中間層を確率分布間の距離を用いて双方の Network の中間層の類似度を測り，その類似度を Loss として Student Network の中間層の重みを学習する．著名な確率分布間の距離推定手法として，下記がある．

- L2 距離

$$L^2(\mathbb{P}_r, \mathbb{P}_g) = \int (P_r(x) - P_g(x))^2 d\mu(x)$$

- カルバック・ライブラー (KL) 距離

$$KL(\mathbb{P}_r \parallel \mathbb{P}_g) = \int \log(\frac{P_r(x)}{P_g(x)}) P_r(x) d\mu(x)$$

- ジェンセン・シャノン (JS) 距離

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \parallel \mathbb{P}_g) + KL(\mathbb{P}_g \parallel \mathbb{P}_r)$$

- 全変動距離 (Total Variation Distance)

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup |\mathbb{P}_r(A) - \mathbb{P}_g(A)|$$

- Wasserstein 距離 (Earth-Mover Distance) [15]

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \prod(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

階層型 Knowledge Distillation の際に使用する Loss Function として L2 距離と Wasserstein 距離を用いる．L2 距離は，[13], [14] にて Knowledge Distillation の際に利用されており，効果があることが示されている．Wasserstein 距離は，最近画像生成の分野で注目されている技術である GAN [16] を拡張した Wasserstein GAN [15] で利用されている手法である．GAN では JS 距離を近似した確率分布間の距離推定手法を使用しているが，Wasserstein 距離を用いた Wasserstein 距離は GAN よりも高精細な画像を生成可能であることが示されている．そこで，本研究の提案手法である階層型 Knowledge Distillation でも Wasserstein 距離を Loss Function として利用する．

本研究では，以後，L2 距離を用いた階層型 Knowledge Distillation を“再学習なし，L2 距離”，Wasserstein 距離を用いた階層型 Knowledge Distillation を“再学習なし，Wasserstein 距離”と呼ぶ．

4.　評 価 実 験

提案手法の有効性を確認するため，下記 2 通りの実験を行う．

（1）　既存手法との精度の比較

（2）　Feature Map の可視化

4.1　実験データ

実験データには，CIFAR-10 [17] の 2 種類のデータセットを用いて評価を行う．CIFAR-10 は，Krizhevsky らによって作成されたデータセットであり，どちらのデータセットも解像度

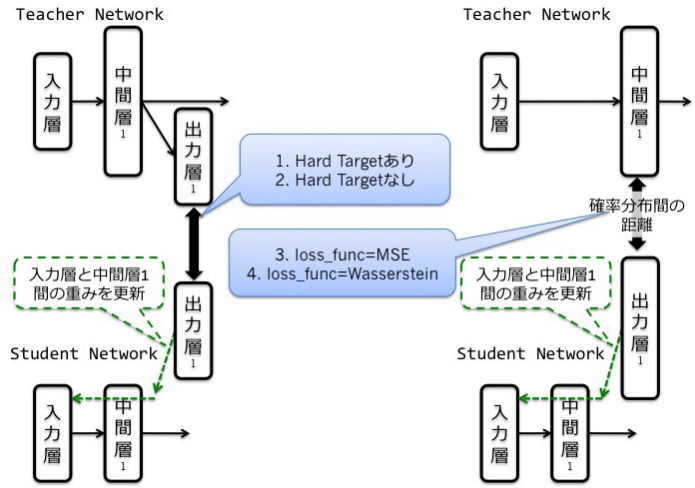

図 3　4 種類の階層型 Knowledge Distillation

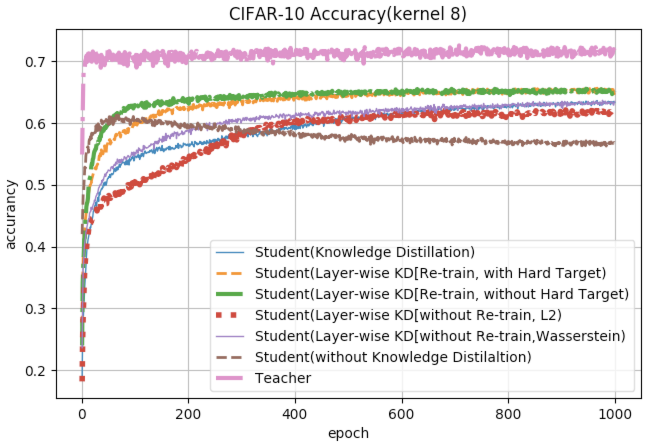

図 4　CIFAR-10 評価結果

32 x 32，RGB チャンネルの 60,000 枚の画像から構成されている．そのうち，50,000 枚を学習用，10,000 枚を評価用として用いる．また，CIFAR-10 は 10 種類のカテゴリを分類するタスクである．

4.2　評 価 方 法

実験では，畳み込み層とプーリング層が 2 層ずつ，全結合層 1 層からなるネットワークを用いた．実験に使用した Teacher Network と Student Network のパラメータを表 1 に示す．表で示されている「Conv(3 × 3 × 128)」は，3 × 3 のウィンドウサイズ，出力チャンネル数が 128，ストライドが 1，パディングが 1 の畳み込み層を示している．「AvgPool(2 × 2)」は，2 × 2 のウィンドウサイズ，ストライドはウィンドウサイズと同サイズ，　パディングが 0 の平均プーリングを行う層を示している．「FC1200」は出力ノード数が 1200 の全結合層を示している．活性化関数は畳み込み層と全結合層の後に適用する．Teacher Network のメモリサイズは 10.5MB, Student Network のメモリサイズは 0.02MB である．なお，1 つのパラメータは，浮動小数点型の 4Byte の変数として，メモリサイズを計算している．

実験に使用したパラメータの詳細を表 2 に示す．

表 1　実験に使用したネットワークの構成

	Teacher	Student
CNN (CIFAR-10)	Conv(3 × 3 × 128) AvgPool(2 × 2) Conv(3 × 3 × 128) AvgPool(2 × 2) FC1200	Conv(3 × 3 × 8) AvgPool(2 × 2) Conv(3 × 3 × 8) AvgPool(2 × 2) FC50
メモリサイズ [MB]	10.5	0.02

表 2　実験に使用したパラメータ一覧

項目	内容
最適化手法	Adam
学習エポック数	1000
活性化関数	Leaky RELU
バッチサイズ	128
前処理	輝度値を 0.0〜1.0 に正規化， 並行移動とミラーリング (CIFAR-100 のみ)

4.3　既存手法との精度の比較

本節では，CIFAR-10 に対して，提案手法である階層型 Knowledge Distillation と既存手法 [10] [11] で精度を比較する．

評価実験では，Teacher Network で通常の学習を行った場合，Student Network に対して，Knowledge Distillation を適用せずに通常の学習を行った場合，Soft Target [10] [11] による Knowledge Distillation を適用した場合，提案手法である階層型 Knowledge Distillation を“再学習あり，Hard Target なし”，“再学習あり，Hard Target あり”，“再学習なし，L2 距離”，“再学習なし，Wasserstein 距離”の 4 種類を適用した場合の計 7 種類を比較する．

CIFAR-10 の評価結果を，図 4，表 3 に示す．Teacher Network の精度が 72%程度であるのに対して，Knowledge Distillation を適用しなかった場合の Student Network の精度は 61%程度である．既存手法である Soft Target を持ちいた Knowledge Distillation の精度は 63%程度であり，Knowledge Distillation を用いない場合よりも改善している．提案手法である階層型 Knowledge Distillation の場合は，“再学習あり”の精度はいずれも精度が 65%程度となっており，既存手法よりも高い精度を持つネットワークを学習できた．また，“再学習なし”の場合には，“L2 距離”を用いた場合は，既存手法よりも少し低い精度であるが，“Wasserstein 距離”を用いた場合には，既存手法よりも僅かに高い精度となっている．

表 3　CIFAR-10 各手法の精度

手法	精度
Teacher	72.57%
Student(Knowledge Distillation なし)	61.30%
Student(Knowledge Distillation)	63.64%
Student(提案手法 [再学習あり, Hard Target あり)	65.65%
Student(提案手法 [再学習あり, Hard Target なし)	65.57%
Student(提案手法 [再学習なし, L2 距離)	62.22%
Student(提案手法 [再学習なし,Wasserstein 距離)	63.69%

4.4　Feature Map の可視化

本節では，中間層の出力である Feature Map を可視化する

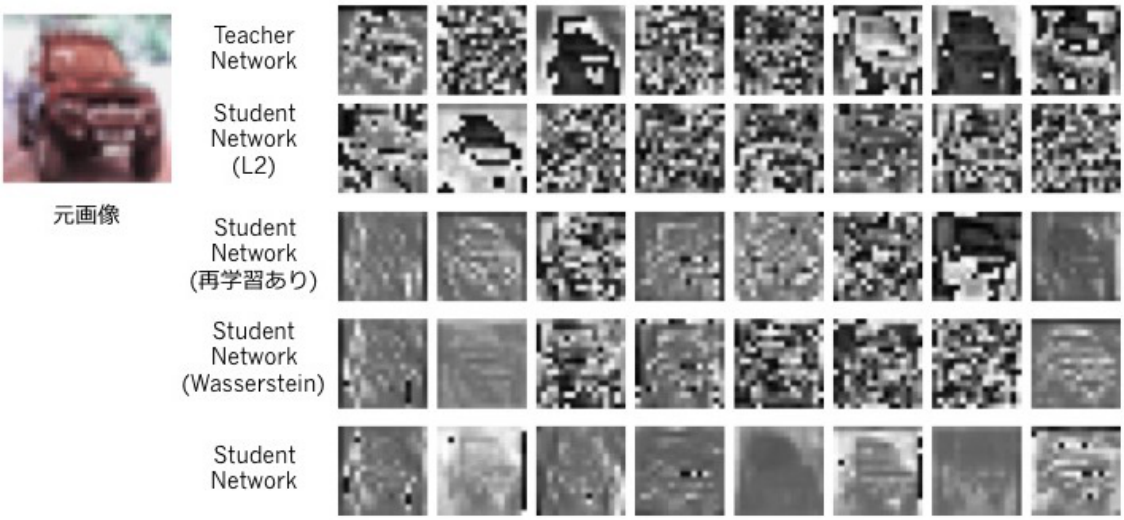

図 5　Feature Map の可視化

ことで，提案手法の有効性を示す．

中間層の出力である Feature Map を可視化したものが，図 5 となる．左側のカラー画像が可視化の際に使用した車カテゴリに属する CIFAR-10 の画像である．右側の画像は，上から，Teacher Network，Student Network(再学習なし，L2 距離)，Student Network(再学習あり，Hard Target なし)，Student Network(再学習なし，Wasserstein 距離)，Student Network(Knowledge Distillation なし) の Feature Map を可視化した画像である．ただし，Teacher Network は，Student Network よりも Feature Map の数が多いため代表的な画像だけを抜き出している．これを見ると，Student Network(Knowledge Distillation なし) の結果では，全体的に薄い画像しかないため，特定の特徴のみしか捉えられていないと考えられる．一方の，提案手法である階層型 Knowledge Distillation を用いた結果では，Student Network(再学習なし，L2 距離) では，Student Network(Knowledge Distillation なし) よりもわずかに画像の傾向が偏っていないだけであるが，Student Network(再学習あり，Hard Target なし)，Student Network(再学習なし，Wasserstein 距離) では画像の傾向が偏っておらず，多様な特徴を捉えられていると考えられる．

また，図 6-図 10 は中間層の出力を t-SNE [18] を用いて 2 次元に縮約して可視化した結果である．0〜9 はクラス番号を示す．Knowledge Distillation を用いなかった結果 (図 7) と比べ，提案手法を用いた結果 (図 8-図 10) は分布が散らばっており，クラスの分類に有効な特徴を捉えられていると考えられる．二通りの可視化による結果からも，4.3 での実験結果のように，提案手法の精度が Knowledge Distillation を使用しない場合に比べて高くなったことがわかる．

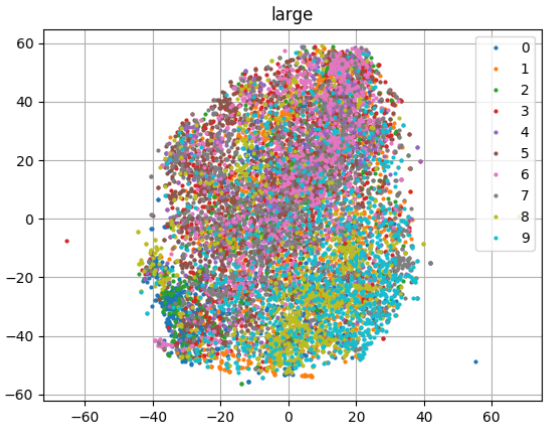

図 6　t-SNE の結果 (Teacher Network)

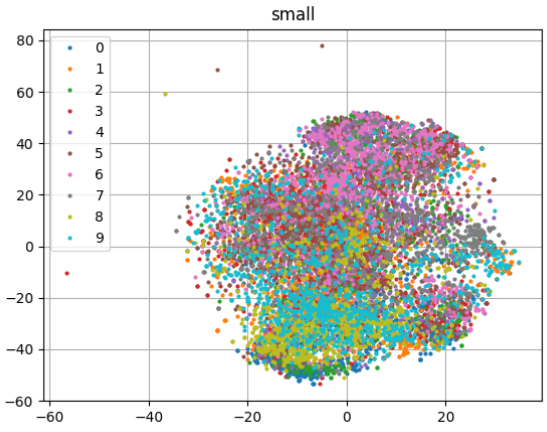

図 7　t-SNE の結果 (Student Network[Knowledge Distillation なし])

5. む　す　び

本研究では，Teacher Network と Student Network の中間層に対して，確率分布間の距離推定手法を用いた Knowledge Distilaltion として，階層型 Knowledge Distillation を提案した．階層型 Knowledge Distilaltion を用いることで，Knowledge Distillation よりも，効率的な学習を行い，より高精度な

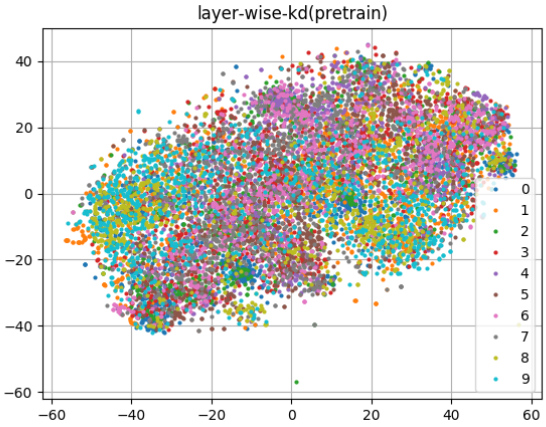

図 8　t-SNE の結果 (Student Network[階層型 Knowledge Distillation, 再学習あり])

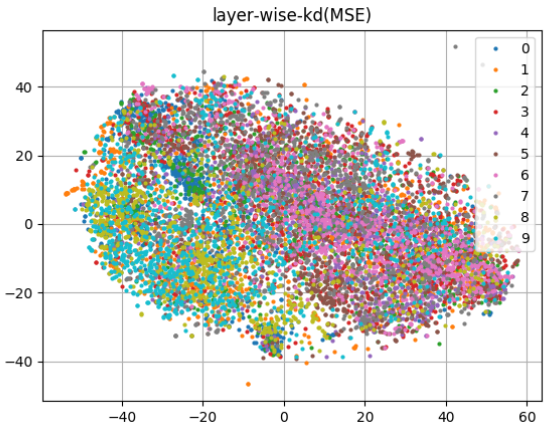

図 9　t-SNE の結果 (Student Network[階層型 Knowledge Distillation, 再学習なし，L2 距離])

Student Network を生成出来ることを示した．本研究では，比較的層の浅い CNN に対する実験のみ実施したため，今後はより層の深いネットワークに対して，実験による効果を示す．また，実験に使用した Teacher Network と Student Network の層の深さは同じであったため，今後の実験では，層の深さが異なるネットワークに対して適用可能かどうかを確認する．

文　　献

[1] ImageNet Large Scale Visual Recognition Challenge. `http://www.image-net.org/challenges/LSVRC/`.

[2] Merck Molecular Activity Challenge. `https://www.kaggle.com/c/MerckActivity`.

[3] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *NIPS*, Vol. 2, pp. 598–605, 1989.

[4] Babak Hassibi and David G Stork. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann Pub, 1993.

[5] Tianxing He, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. Reshaping deep neural network for fast decoding by node-pruning. In *ICASSP*, pp. 245–249, 2014.

[6] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NIPS*, pp. 1135–1143, 2015.

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR, abs/1510.00149*, Vol. 2, , 2015.

[8] Wenlin Chen, James T Wilson, Stephen Tyree, Kilian Q Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. *CoRR, abs/1504.04788*, 2015.

[9] Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, pp. 535–541. ACM, 2006.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *stat*, Vol. 1050, p. 9, 2015.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Dark knowledge. *Presented as the keynote in BayLearn*, Vol. 2, , 2014.

[12] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong. Learning small-size dnn with output-distribution-based criteria. In *INTERSPEECH*, pp. 1910–1914, 2014.

[13] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *In ICLR*, 2015.

[14] Ji-Hoon Bae Junho Yim, Donggyu Joo and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *In CVPR*, 2017.

[15] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[18] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

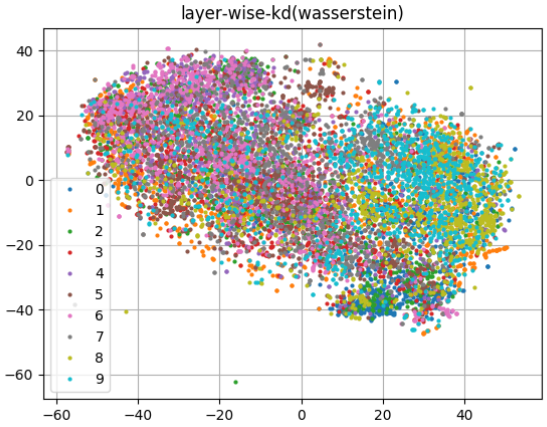

図 10　t-SNE の結果 (Student Network[階層型 Knowledge Distillation, 再学習なし，Wasserstein 距離])