The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

L1 正則化を用いたCapsule Networkの学習法の一提案

L1 Regularization based Learning Method for Capsule Network

太田 望 *1　河合 新 *1　延原 肇 *1
Nozomu Ohta　Shin Kawai　Hajime Nobuhara

*1筑波大学大学院システム情報工学研究科知能機能システム専攻
Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba

Capsule Network is a new neural network proposed to overcome the shortcomings of CNN. However, the Capsule Network has many learnable parameters and is prone to over-fitting. In this research, we aim to improve generalization ability by reducing parameters using L1 regularization. We evaluate our method by comparing the accuracy and the reconstructed image with the conventional method.

1. はじめに

畳み込みニューラルネットワーク (Convolutional Neural Network、以下 CNN) は画像認識の分野で一般的な手法として認識されている。このネットワークでは、特徴検出を行う畳み込み層とシフト不変性のためのプーリング層を繰り返す構造によって線の傾き、太さや色などの低レベルの特徴から、顔画像などにおける目や口などの高レベルの特徴を認識している。CNN の問題点として、プーリング層が抽出した特徴間の位相構造を保存しないため、同じオブジェクトであっても姿勢が異なると全く異なる内部表現になり、大規模なデータ拡張と非常に深いネットワークが必要になることが挙げられる。

本研究では、この CNN の問題を解決するために、Google Brain のチームによって提案されている Capsule Network(以下 CapsNet) に着目する。[Hinton 11, Sabour 17]。通常の CNN の各層は、特徴の存在確率をスカラーとして出力するのに対し、CapsNet はオブジェクトの存在と特徴をベクトルとして表現する。下位のカプセルはオブジェクトの一部の特徴をベクトルとして保持し、複数の下位カプセルからオブジェクト全体の特徴と存在を表現する上位カプセルを計算する。

CapsNet に関して拡張の余地がある点としては、CNN はプーリング層によってパラメータが減るのに対し CapsNet はパラメータが増えるため、ネットワークの大規模化が難しく過学習に陥りやすいことが挙げられる。特に大きな画像を入力した場合や分類クラスを増やした場合、カプセルが増加し学習が困難になる。本論文では CapsNet のパラメータを L1 正則化によって縮小した後、閾値以下の値を 0 とすることでスパース化をはかり、より単純で過学習に強いモデルを学習することを目指す。

提案手法と元の CapsNet を MNIST、fashion MNIST、SVHN データセットを用いて学習させ、テストデータの正答率を比較することで評価を行う。

2. CapsNet

CapsNet は入力を画像とし、分類クラスの確率ベクトルと再構成画像を出力する。CapsNet の概要図を図 1 に示す。

CapsNet の計算手順を以下に示す。

連絡先: 太田望, 筑波大学大学院システム情報工学研究科知能機能システム専攻, ohta@cmu.iit.tsukuba.ac.jp

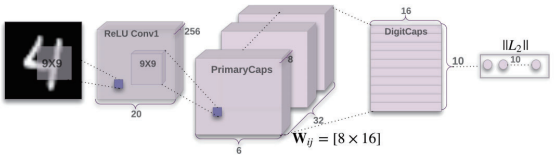

図 1: CapsNet の概要図

1. 入力画像を 2 層の畳み込み層によって 256 チャンネル、6×6 の特徴マップへ変換する。図 1 の入力画像と Conv1 間、Conv1 と PrimaryCaps 間の変換に相当する。

2. 特徴マップを画素ごと、256 チャンネルを 8 チャンネルごとに分割する。すなわち $6 \times 6 \times 32$ 個の 8 次元ベクトルができる。i 番目のベクトルを下位カプセル $\boldsymbol{u}_i$ と定義する。図 1 の PrimaryCaps に相当する。

3. $\boldsymbol{u}_i$ と重み行列 $W_{ij} \in R^{8\times16}$ の積を $\hat{\boldsymbol{u}}_{j|i}$ とする。この W_{ij} は誤差逆伝播法により学習される。

$$\hat{\boldsymbol{u}}_{j|i} = W_{ij}\boldsymbol{u}_i \tag{1}$$

4. $\hat{\boldsymbol{u}}_{j|i}$ と $c_{ij} \in R$ の重み付き和を計算する。c_{ij} は後述する Dynamic Routing によって計算する。

$$\boldsymbol{s}_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i} \tag{2}$$

5. 以下の squash 関数によって上位カプセル $\boldsymbol{v}_j = \text{squash}(\boldsymbol{s}_j)$ を計算する。squash 関数はベクトルの向きを保存したままノルムを $[0, 1)$ の範囲に変換する活性化関数である。上位カプセルは図 1 の DigitCaps に対応する。

$$\text{squash}(\boldsymbol{s}) = \frac{||\boldsymbol{s}||^2}{1 + ||\boldsymbol{s}||^2}\frac{\boldsymbol{s}}{||\boldsymbol{s}||} \tag{3}$$

6. 3–5 の手順を繰り返し 16 次元の上位カプセルを 10 個構成する。

Dynamic Routing のアルゴリズムを Algorithm 1 に示す。r はルーティング回数、l はレイヤー数を表す。先行研究 [Sabour 17] では $r = 3$、$l = 1$ で固定している。Dynamic

Routing によって下位カプセルから一度上位カプセルを計算し、計算された上位カプセルに似た下位カプセルを内積によって抽出し、抽出された下位カプセルを使用してサイド上位カプセルを計算する。このように、EM アルゴリズムのように上位カプセルの推定と下位カプセルの選択を同時に行う。上位カプセルは分類対象の各クラスに対応し、画像の分類と再構成に使われる。全体の損失関数は以下に説明する分類誤差と再構成誤差の和である。

各上位カプセルのノルムを対応するオブジェクトの存在確率とする。分類の損失関数に以下の margin loss 関数を使用する。

$$L_{class} = \sum_k (T_k \max(0, m^+ - ||\boldsymbol{v}_k||)^2 + \lambda(1 - T_k)\max(0, ||\boldsymbol{v}_k|| - m^-)^2) \quad (4)$$

k は分類クラス番号であり、T_k は真のクラスの場合 1、それ以外の場合は 0 をとる。m^+ は真のクラスの誤差に対する閾値、m^- は誤ったクラスの誤差に対する閾値である。先行研究 [Sabour 17] では $m^+ = 0.9$ と $m^- = 0.1$ と $\lambda = 0.5$ である。

各上位カプセルに対し全結合層を 3 層結合し入力画像を再構成する。先行研究 [Sabour 17] はカプセルの要素は対応するオブジェクトがどのような姿勢やスケールであるかを表現していると主張している。その根拠としてカプセルの各要素を少しずつ増減させた場合に再構成されたオブジェクトの変化が要素ごとに異なることを示している。

3. L1 正則化に基づく CapsNet 学習法の提案

CapsNet では下位カプセルと 重み行列 W_{ij} の積によって上位カプセルが計算されるが、下位カプセルの全特徴が上位カプセルの全特徴の予測の役に立つわけではない。顔の識別の例で考えると、目の傾きの情報は顔の大きさとは無関係である。したがって W_{ij} の要素の大半は 0 であると考えられる。そこで先行研究 [Han 15] を参考に重み行列 W_{ij} のスパース化を図る。

CapsNet を通常の方法で 50epoch 学習した後、L1 正則化を使って 50epoch 学習する。L1 正則化とは誤差関数にパラメータの絶対値の総和に係数をかけたものである正則化項を加えることでパラメータの値を縮小する手法である。最初から L1 正則化項を誤差関数に加えた場合、係数をどれだけ小さくしても正則化項が大きくなり、学習がうまく進まなかった。設定された閾値以下の重み行列 W_{ij} のパラメータの値を 0 とする。閾値は今回は 10^{-4} とした。

従来手法である CapsNet に対して通常のテストデータと微変形したテストデータの分類精度によって提案手法を評価する。また、W のパラメータのうち、何%が 0 になったのか、再構成された画像が妥当なものかを評価する。

4. 実験

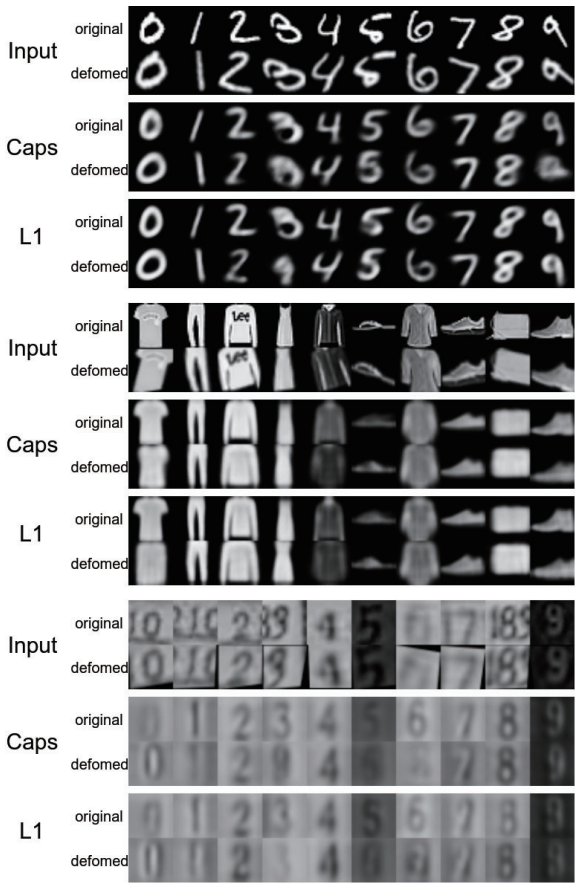

図 2: MNIST の再構成画像

参考文献 [Nair 18] に従い、MNIST、fashion MNIST、SVHN データセットとその微小変形したデータを使用して提案手法の評価を行う。MNIST は 0 から 9 の手書き数字 28×28 白黒画像のデータセットであり画像認識の評価に使用される。fashion MNIST は洋服や靴などの 10 クラスの 28×28 白黒画像のデータセットである。SVHN は google street view に写った家の表札の数字を切り抜いた 32×32 の画像データセットである。それぞれの学習データのうち 15%をバリデーションデータとして使用した。

データの微小変形は以下の手順で行う。

1. 画像を 1.2 倍に拡大
2. 画像中心を回転中心として $[-20, 20]$ 度から一様ランダムに回転
3. x 軸、y 軸方向に [-0.2,0.2] の範囲でせん断変形

実装には Python3 と Keras を使用し、計算には Geforce GTX 1080 Ti を使用した。最適化アルゴリズムは Adam を使用した。L1 正則化はモデル中の全てのパラメータに対して適用した。

表 1 に実験結果を示す。ここで、Deformed は、テストデータに微小変形を加えた場合、Original は微小変化を加えていない場合を表す。いずれのデータセットでも微小変形の有無にかかわらず提案手法によって正答率が改善されていることか

Algorithm 1 Routing algorithm

procedure ROUTING($\hat{\boldsymbol{u}}_{j|i}$,$r$,$l$)
  for all capsule i in layer l and capsule j in layer$(l+1)$:$b_{ij} \leftarrow 0$
  for r iterations **do**
    for all capsule i in layer l : $\boldsymbol{c_i} \leftarrow \mathrm{softmax}(\boldsymbol{b_i})$
    for all capsule j in layer $(l+1)$: $\boldsymbol{s_j} \leftarrow c_{ij}\hat{\boldsymbol{u}}_{j|i}$
    for all capsule i in layer $(l+1)$: $\boldsymbol{v_j} \leftarrow \mathrm{squash}(\boldsymbol{s_j})$
    for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$
  end for
  return $\boldsymbol{v_j}$
end procedure

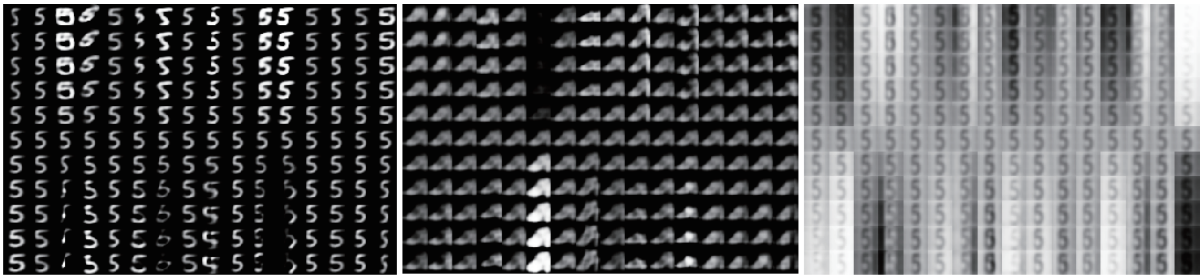

図 3: カプセルの 16 次元の要素の変化させた結果：MNIST（左）、fashion MNIST（中央）、SVHN（右）

表 1: 従来手法 (CapsNet) と提案手法 (L1) の正答率の比較

Dataset	Original		Deformed	
Network	CapsNet	L1	CapsNet	L1
MNIST	99.13	**99.30**	91.37	**93.10**
fashion MNIST	88.29	**90.49**	61.57	**62.27**
SVHN	91.91	**93.29**	70.86	**74.17**

表 2: パラメータのスパース性

	Sparsity[%]
MNIST	68.57
fashion MNIST	62.90
SVHN	68.01

ら、提案手法の有効性が確認できる。W のパラメータのうち 0 になったものの割合を表 2 に示す。どのデータセットでも 6 割以上のパラメータが 0 になっており、先行研究 [Han 15] と同様にネットワークのスパース化に成功している。また、各データセット、各クラスの入力画像と再構成画像を図 2 に示す。MNIST データセットでは従来手法、提案手法によらず、変形されたデータに対応して再構成された画像が変化していることが確認できる。提案手法の方が再構成画像が少し明確になっている。他のデータセットでは従来手法と提案手法の違いは見られない。図 3 はカプセルの 16 次元の要素のうち、一つを選んで変動させ、他の 15 個を 0 に固定した場合の再構成画像である。ここで横軸がカプセルの次元、縦軸は変動量を表す。MNIST データセットでは選んだ要素によって数字の位置や形、大きさなどが変化している様子が確認できる。SVHN データセットでは数字の変化より背景の変化に大きく影響されていることがわかる。

5. おわりに

従来手法に比べ提案手法ではテストデータの微小変形の有無に関わらず精度が向上した。これは提案手法によって過学習が抑えられ、CapsNet の汎化性能が向上したためだと考えられる。今後の課題としては、テストデータをオブジェクトを様々な角度から撮影したものに変えて CapsNet がオブジェクトの姿勢の変化に対応する能力をテストすること、他のスパース化手法を試すことなどが挙げられる。

参考文献

[Han 15] Han, S., Pool, J., Tran, J., and Dally, W.: Learning both Weights and Connections for Efficient Neural Network, in Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. eds., *Advances in Neural Information Processing Systems 28*, pp. 1135–1143, Curran Associates, Inc. (2015)

[Hinton 11] Hinton, G. E., Krizhevsky, A., and Wang, S. D.: Transforming Auto-Encoders, in Honkela, T., Duch, W., Girolami, M., and Kaski, S. eds., *Artificial Neural Networks and Machine Learning – ICANN 2011*, pp. 44–51, Berlin, Heidelberg (2011), Springer Berlin Heidelberg

[Nair 18] Nair, , Prem, , Doshi, R., and Keselj, S.: Pushing the limits of capsule networks (2018), Technical note

[Sabour 17] Sabour, S., Frosst, N., and Hinton, G. E.: Dynamic Routing Between Capsules, in Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. eds., *Advances in Neural Information Processing Systems 30*, pp. 3856–3866, Curran Associates, Inc. (2017)