少数データから構成されるタスクに対する深層学習モデルに関する研究

1X16C070-4　清水瑛貴
指導教員　後藤正幸

1. 研究背景・目的

深層学習技術の発展に伴い，画像認識や囲碁ゲームなどのタスクにおいて電算機は人間を超える性能を示すようになっている．一方で，深層学習モデルが高い性能を発揮するためには，ラベル付きの訓練データが大量に必要となる．しかし，多くの実問題においては，ラベル付きのデータが相対的に少なく，大量のデータを集めてラベルを付与するためには多大なコストがかかるという課題がある．そのため，少数のデータからどのように大量のパラメータを有する深層学習モデルを効率的に学習させるかという課題が重要視されている．

このような問題に対し，有用な事前知識をモデルに組み込むことが効果的であることが知られている．近年では，複数の物事に対する共通の概念を自ら学習し，得られた知識を未知の問題に適用する人間の学習過程を模したメタ学習という手法が注目されている．メタ学習は，類似したタスクの集合から，機械学習モデルが自らタスク間に共通の概念 (メタ知識) を学習することを目的とする．

深層学習モデルに対してメタ学習を適用した代表的な手法として Model-Agnostic Meta-Learning [1] (以下，MAML) という手法がある．MAML では，メタ知識としてタスク間に共通の初期パラメータを学習することにより，少数データしか存在しない新たなタスクに対しても，少ないパラメータ更新で効率的に適応することを可能とする．

しかし，MAML では各タスクに適応する際に統一的にパラメータの更新を行っており，タスク毎の特徴が十分に考慮されていない．実際には，タスク毎にメタ知識がどの程度有用であるかは問題により異なると考えられる．例えば，学習済のタスクと新たなタスクが類似している場合や，新タスク内のデータ数が極めて少ない場合には過学習を防ぐためにメタ知識を重視して学習する方が合理的である．一方で，類似性が低い場合や新タスクの学習データ数が比較的多い場合には十分なパラメータの更新を行うことが望ましい．

そこで本研究では，MAML においてタスク毎に学習内容を調整し，より性能の高いモデルを得る学習手法を構築することを目的とする．具体的には，提案手法ではデータを圧縮することで得られるタスクの特徴を表す表現ベクトルを用いて，タスク毎の学習率とパラメータの初期値に対する重みを導出する．また，各パラメータの推定は確率モデルにより定式化する．最後に，画像のデータセットである MiniImagenet を用いて実験を行い，提案手法の有効性をメタ学習におけるベンチマークデータを用いて示す．

2. 準備

2.1. 問題設定

メタ学習では，類似したタスクの集合 $\mathcal{T}$ を考える．$\mathcal{T}$ には，メタ知識を獲得するためのメタ訓練用のタスク集合 $\mathcal{T}_{train}$ と，モデルの性能を検証するメタテスト用のタスク集合 $\mathcal{T}_{test}$ が存在する．また両タスク集合に含まれる各タスク T_i は，タスク毎の学習に用いる訓練データ $\mathcal{D}^{T_i} = \left\{\left(x_n^{T_i}, y_n^{T_i}\right)\right\}_{n=1}^{N}$ とタスク毎の検証に用いるテストデータ $\tilde{\mathcal{D}}^{T_i} = \left\{\left(\tilde{x}_n^{T_i}, \tilde{y}_n^{T_i}\right)\right\}_{n=1}^{N'}$ から構成される．ここで，$x_n^{T_i}, \tilde{x}_n^{T_i}$ はタスク T_i におけるデータ，$y_n^{T_i}, \tilde{y}_n^{T_i}$ はそのラベルである．

2.2. MAML

学習器である深層学習モデルのパラメータを $\boldsymbol{\theta}$ とする．MAML では，$\mathcal{T}_{train}$ でタスク間に共通の初期パラメータ $\boldsymbol{\theta}_0$ を学習することで，$\mathcal{T}_{test}$ の新たなタスクに対しても少数のデータから効率的に適応することを目的とする．$\mathcal{T}_{train}$ における学習では，B 個のタスクの集合からなるバッチ $\boldsymbol{B}$ を用いる．各バッチでは，各タスクにおける $\mathcal{D}^{T_i}$ に対する Inner Loop と，全タスクにおける $\tilde{\mathcal{D}}^{T_i}$ に対する Outer Loop という二つの手順により $\boldsymbol{\theta}_0$ を更新する．以下に，各手順における更新式を示す．ここで，α, β は学習率，$L(\cdot,\cdot)$ はクロスエントロピーなどの損失関数である．

Step1) Inner Loop：各タスクへの適応

$$\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}_0 - \alpha \bigtriangledown_{\boldsymbol{\theta}} L\left(\boldsymbol{\theta}_0, \mathcal{D}^{T_i}\right) \tag{1}$$

Step2) Outer Loop：初期パラメータの更新

$$\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_0 - \frac{\beta}{B} \bigtriangledown_{\boldsymbol{\theta}} \sum_{T_i \in \boldsymbol{B}} L\left(\boldsymbol{\theta}', \tilde{\mathcal{D}}^{T_i}\right) \tag{2}$$

これらのパラメータの更新は，$\mathcal{T}_{train}$ 内のタスクに対して損失関数が収束するまで繰り返し行われる．

3. 提案モデル

3.1. 概要

MAML では，Inner Loop における学習率 α と初期パラメータ $\boldsymbol{\theta}_0$ は全タスクについて共通であると仮定している．しかし，タスクによってはメタ知識が有用にならず，パラメータ更新の加減を大きくしなければタスクに適応できないことも考えられる．この場合には，初期パラメータの寄与度を小さくし，学習率を大きくするなどの調整が必要である．そこで，α と $\boldsymbol{\theta}_0$ をタスクごとに重み付けする方法を考える．

提案手法では，まず訓練データ $\mathcal{D}^{T_i}$ に対して複数のデータを同時に圧縮することで一つのベクトルを出力する Deep Sets [2] を用いる．これにより，タスクの特徴を表す表現ベクトル $\boldsymbol{v}$ を得る．そして，$\boldsymbol{v}$ から α と $\boldsymbol{\theta}_0$ に対する重みを生成する．最後に，MAML と同様の手順でパラメータの更新を行う．これらの一連の流れは確率モデルにより定式化され，ベイズ推論によりパラメータの推定を行う．

3.2. パラメータの更新方法

提案モデルでは，学習率 α と初期パラメータ $\boldsymbol{\theta}_0$ に対する重みを生成するために，タスクごとに特有の新たな変数 $\boldsymbol{\gamma}_i$ と $\boldsymbol{\omega}_i$ を導入する．これらの変数は，表現ベクトル $\boldsymbol{v}$ を全結合層や ReLu 関数により適切な次元のベクトルに変換するこ

とで生成する．そして，得られた変数 $\boldsymbol{\gamma}_i$ と $\boldsymbol{\omega}_i$ を非負の値を出力する SoftPlus 関数 g により変換することで α と $\boldsymbol{\theta}_0$ に対する重みを求める．したがって，学習率は $g(\boldsymbol{\gamma}_i)\cdot\alpha$，初期パラメータは $g(\boldsymbol{\omega}_i)\cdot\boldsymbol{\theta}_0$ のように調整される．以上より，Inner Loop における更新式は以下のようになる．

$$\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}_0 - g(\boldsymbol{\gamma}_i)\cdot\alpha \bigtriangledown_{\boldsymbol{\theta}} L\left(g(\boldsymbol{\omega}_i)\cdot\boldsymbol{\theta}_0, \mathcal{D}^{T_i}\right) \quad (3)$$

これにより，タスク T_i 毎に α と $\boldsymbol{\theta}_0$ を変更することができ，変数 $\boldsymbol{\gamma}_i$ と $\boldsymbol{\omega}_i$ を適切に学習することで，よりタスク間の類似性を加味したメタ学習が可能になると考えられる．

3.3. 確率モデルによる定式化

新たに導入した変数 $\boldsymbol{\gamma}_i$ と $\boldsymbol{\omega}_i$ は各タスクに対する潜在変数であると解釈できる．したがって，これらの変数は確率モデルにより定式化され，変分ベイズ法により推定することができる．図 1 に，グラフィカルモデルを示す．ここで，$\boldsymbol{\phi}=(\boldsymbol{\gamma}_i,\boldsymbol{\omega}_i)$，$N$ は $\mathcal{D}^{T_i}$ 内のデータ数，N' は $\tilde{\mathcal{D}}^{T_i}$ 内のデータ数，M は $\mathcal{T}_{train}$ 内のタスク数とする．

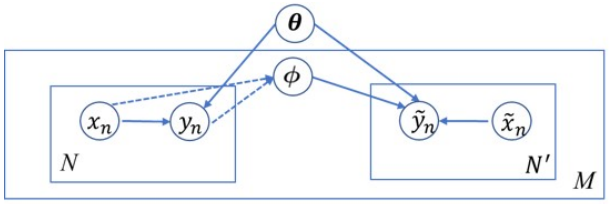

図 1: グラフィカルモデル

$\boldsymbol{\phi}$ は Ravi ら [3] と同様の手順で $\mathcal{D}^{T_i}$ のみから推論する．このとき，事後分布 $p\left(\boldsymbol{\phi}\mid\mathcal{D}^{T_i}\right)$ を解析的に計算することは困難であり，またタスク数に比例した潜在変数の増加を抑えるため，パラメータ ξ をもつニューラルネットワークにより近似分布 $q\left(\boldsymbol{\phi}\mid\mathcal{D}^{T_i};\xi\right)$ を算出する．以上より，Outer Loop は以下のような最適化問題に帰着される．

$$min_{\boldsymbol{\theta},\boldsymbol{\phi}}\frac{1}{M}\sum_{M}\left\{\frac{1}{N'}\sum_{N'}-\log p(\tilde{y}_n\mid\tilde{x}_n,\boldsymbol{\phi};\boldsymbol{\theta}) + KL\left[q\left(\boldsymbol{\phi}\mid\mathcal{D}^{T_i};\xi\right)\parallel p(\boldsymbol{\phi})\right]\right\} \quad (4)$$

ここで，事前分布 $p(\boldsymbol{\phi})$ は多次元正規分布 $\mathcal{N}(\mathbf{0},\mathbf{I})$ と仮定する．KL はカルバック・ライブラー情報量を表す．

4. 評価実験

4.1. 実験条件

提案手法の有効性を示すため画像データの識別実験を行った．データセットとしてはメタ学習におけるベンチマークデータである MiniImagenet を用いる．MiniImagenet は，100 カテゴリに対して各 600 枚の計 60,000 枚の画像から構成されている．そのうち，64 カテゴリがメタ訓練用に，20 カテゴリがメタテスト用に事前に割り当てられている．

各タスク内の訓練データ $\mathcal{D}^{T_i}$ において，5 カテゴリの画像が 1 枚ずつある 5-way 1-shot と，5 カテゴリの画像が 5 枚ずつある 5-way 5-shot の設定で実験を行った．両実験ともタスク内のテストデータ $\tilde{\mathcal{D}}^{T_i}$ は各カテゴリの画像を 15 枚ずつ用意した．なお，各タスク T_i は該当するカテゴリの画像からランダムに，必要な枚数だけ抽出して作成される．

実験条件によってはメタ知識を直接的に活用した方が良い場合と，十分な適応が必要な場合とがあると考えられる．これを確認するために，提案手法に対する比較手法としては，通常の MAML に加えて，MAML において Inner Loop の更新を省いたモデル (以下，No Inner) を採用する．ベースとなる学習器としては 4 つのモジュールからなる Convolutional Neural Network(以下，CNN) を用いた．

メタ訓練時において，バッチ内のタスク数は 4，バッチの合計は 30,000，α は 0.01，β は 0.001 とした．また，メタテスト時は 3,000 の異なるタスクにおいて，タスク内のテストデータ $\tilde{\mathcal{D}}^{T_i}$ に対する精度を求め，その平均を識別精度として算出した．なお，提案手法における初期パラメータの重み付けは CNN の 3 番目のモジュールにのみ適用した．

4.2. 結果と考察

表 1 に，各手法の識別精度を示す．

表 1: MiniImagenet における実験結果 (%)

手法	5-way 1-shot	5-way 5-shot
MAML	43.1	60.7
No Inner	44.8	59.5
提案手法	**46.3**	**61.6**

まず MAML と No Inner を比較する．No Inner は Inner Loop が省略されているため，$\mathcal{T}_{test}$ に対してメタ知識を直接的に適用している．表 1 より，1-shot の場合には No Inner が，5-shot の場合には MAML の識別精度が高いことが分かる．これは，各タスクにデータ数が少ない場合はメタ知識をそのまま適応する方が良く，多い場合には初期パラメータの十分な更新が必要であることを示唆している．

一方で，提案手法は 1-shot と 5-shot ともに比較手法よりも高い精度を示した．したがって，各タスクの特性を考慮したうえで学習率 α や初期パラメータ $\boldsymbol{\theta}_0$ を適切に調整できていると考えられる．以上より，提案手法はメタ知識の活用の加減をタスク毎にコントロールすることで MAML よりも高い識別精度を示すことが確認できた．

5. まとめと今後の課題

本研究では，MAML よりも性能の高いモデル構築することを目的として，タスク毎にメタ知識の活用の度合いを適正化するために，確率モデルを用いて学習率や初期パラメータを調整するモデルを提案した．そして，画像データセットを用いた実験により手法の有効性を示した．今後の課題としては，提案モデルで得られたタスクの表現ベクトルを用いて，ドメイン外のタスクの検知に応用することなどが考えられる．

参考文献

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-agnostic Meta-learning for Fast Adaptation of Deep Networks”, *International Conference on Machine Learning*, pp. 1126–1135, 2017.

[2] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov and Alexander J Smola, “Deep Sets”, *Advances in Neural Information Processing Systems*, 2014.

[3] Sachin Ravi and Alex Beatson,“Amortized Bayesian Meta-Learning”, *International Conference on Learning Representations*, 2019.