

疑似訓練サンプル最適化による ニューラルネットワークの少数ショット学習

Deep few-shot learning with pseudo example optimization

木村 昭悟 *1*²
Akisato Kimura

Zoubin Ghahramani*²
Zoubin Ghahramani

竹内 孝 *1
Koh Takeuchi

岩田 具治 *1
Tomoharu Iwata

上田 修功 *1
Naonori Ueda

*1 日本電信電話 (株) コミュニケーション科学基礎研究所
Communication Science Laboratories, NTT Corporation

*2 ケンブリッジ大学 工学部
Department of Engineering, University of Cambridge

This paper proposes a novel method for training neural networks with a limited amount of training data. Our approach is based on knowledge distillation that transfers knowledge from a deep reference neural network to a shallow target one. The proposed method employs this idea to mimic predictions of reference models more robust against overfitting than the target neural network. Different from almost all the previous work for knowledge distillation that requires a large amount of labeled training data, the proposed method requires only a small amount of training data. Instead, we introduce pseudo training data that is optimized as a part of model parameters.

1. はじめに

画像認識に関連する様々なタスクの state-of-the-art は、数多くの層を持つ深いニューラルネットワーク (neural networks; NN) モデルと、大規模教師付データセットによって支えられている。しかし、解くべき問題に合わせて大規模教師付データセットを都度準備することは非常に多くの人的労力が必要となると共に、医療応用などいくつかの実世界の問題においては、そもそもデータそのものを大量に獲得すること自体が困難である場合も珍しくない。少量の訓練データで NN を学習すると、訓練データに過剰に適合する過学習を引き起こし、タスクの性能が大幅に低下する。NN モデルの過学習を避ける方策として、大量の教師なしデータを追加で用いる半教師付学習 [Goodfellow 14] や、大量の教師付データから学習したモデルを追加で利用する転移学習 [Caruana 94, Koch 15] が広く用いられている。しかし、いずれの方策においても、大量の教師なしデータや学習済モデルなど何らかの追加データを準備する必要があり、少数の訓練データのみから NN を適切に学習する方法は、依然として重要な研究課題の一つとなっている。

一方、SVM やガウス過程回帰など、従来から用いられている機械学習モデルのいくつかは、ニューラルネットワークと比較して過学習を引き起こしにくいことが知られている。本研究では、この点に着目した新しいニューラルネットワークの学習方法を提案する。図 1 に提案手法の概要を示す。提案手法では、半教師付学習や転移学習とは異なり、所与の少量訓練データ以外のデータ資源は仮定しない。提案手法では、まず NN よりも過学習に強いとされる NN 以外の予測器を所与の訓練データで学習し、この予測器を参照モデルとして、目的の NN モデルを知識蒸留 (knowledge distillation) [Buciluă 06, Hinton 15] により学習する。すなわち、参照モデルの出力と目標 NN モデルの出力が同様となることを目指して、目標モデルを学習する。従来の知識蒸留手法では、参照モデル・目標モデルの双方の学習において共通の大規模訓練データを用いることが一般的であり、訓練データが少量の場合には目標モデルを十分に

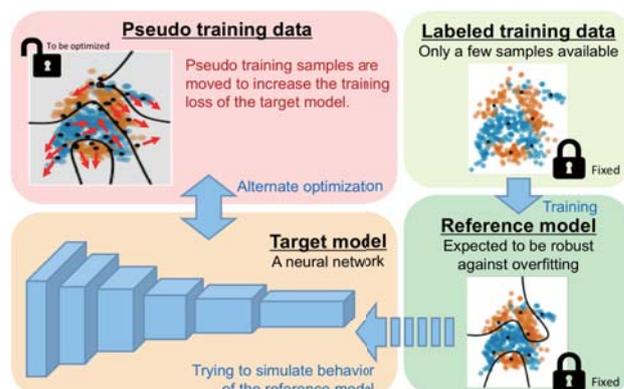


図 1: 提案手法の概略

学習することができない。提案手法では、この問題を解決するために、少数の訓練データから生成した大量の疑似訓練データ (pseudo training data) を導入すると共に、この疑似訓練データを目標モデルの学習の過程で学習損失がより大きくなるように追加更新を行う。これにより、現在の目標モデルで十分に学習が行われていない領域を探し出し、その領域における学習を促進させることを目指す。さらに、提案手法では、参照モデルによる予測の不確かさに基づいて疑似訓練サンプルの重みを調整する fidelity weighting [Dehghani 17] を導入する。これにより、参照モデルの予測が確からしい領域のみを目標モデルに転移でき、目標モデルの予測性能のさらなる向上が期待できる。

2. モデル最適化

2.1 知識蒸留

知識蒸留 [Buciluă 06, Hinton 15] は、訓練済の参照モデル $g(\mathbf{x}; \theta_g)$ が持つ予測性能を、目標モデル $f(\mathbf{x}; \theta_f)$ に転移させるための枠組である。ここで、 \mathbf{x} は入力サンプル、 θ_g と θ_f はそれぞれ参照モデル・目標モデルのモデルパラメータである。以降では簡単のため、特に必要な場合を除いてモデルパラメー

タを省略する。多くの知識蒸留手法では、参照モデルとして層が多いもしくは幅が広い NN モデルを仮定し、層が少ないもしくは幅の狭い目標 NN モデルに参照モデルと同等程度の予測性能を持たせることを目的としている。その際に用いられる訓練損失の最も一般的な定式化 [Hinton 15] を以下に示す。以降では、この訓練損失を蒸留損失 (distillation loss) と呼ぶ。

$$L_{\text{dis}}(\mathbf{X}^L, \mathbf{Y}^L) = \frac{\lambda_1}{N_L} \sum_{n=1}^{N_L} D_1(\mathbf{y}_n^L, f(\mathbf{x}_n^L)) + \frac{\lambda_2}{N_L} \sum_{n=1}^{N_L} D_2(g(\mathbf{x}_n^L), f(\mathbf{x}_n^L)), \quad (1)$$

ここで、 $\mathbf{X}^L = \{\mathbf{x}_1^L, \dots, \mathbf{x}_{N_L}^L\}$ は教師付訓練サンプルの集合、 $\mathbf{Y}^L = \{\mathbf{y}_1^L, \dots, \mathbf{y}_{N_L}^L\}$ はそれに対応する教師情報の集合、 $D_1(\mathbf{y}, \hat{\mathbf{y}})$ は教師情報 \mathbf{y} とモデル予測 $\hat{\mathbf{y}}$ との乖離を評価する損失関数、 $D_2(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2)$ は 2 つの異なるモデル予測 $\hat{\mathbf{y}}_1$ と $\hat{\mathbf{y}}_2$ との乖離を評価する損失関数である。知識蒸留は分類問題に適用する、すなわち教師情報 \mathbf{y} を one-hot vector とするのが一般的であるが、教師情報や損失関数を適切に設定することで回帰問題など他の問題にも適用可能である。

2.2 提案手法の損失関数

式 (1) に示した蒸留損失を含め、従来の知識蒸留手法の多くは、教師付訓練サンプル ($\mathbf{X}^L, \mathbf{Y}^L$) を用いて目標モデル $f(\cdot)$ を学習する。しかし、少量の教師付訓練データのみでモデル学習を行うと過学習を引き起こす。ここで、蒸留損失の式 (1) を確認すると、右辺第 2 項は教師情報 \mathbf{Y}^L がなくても計算できることに気がつく。提案手法では、この点を考慮して、新しい訓練損失 L_{imi} を定義する。この損失では、少数の教師付訓練サンプル ($\mathbf{X}^L, \mathbf{Y}^L$) に加えて、新たに疑似訓練サンプル $\mathbf{X}^P = \{\mathbf{x}_1^P, \dots, \mathbf{x}_{N_P}^P\}$ を導入し、この疑似訓練サンプルから蒸留損失の第 2 項に相当する項を計算する。以降では、この訓練損失を模倣損失 (imitation loss) と呼ぶ。

$$L_{\text{imi}}(\mathbf{X}^L, \mathbf{Y}^L, \mathbf{X}^P) = \frac{\lambda_1}{N_L} \sum_{n=1}^{N_L} D_1(\mathbf{y}_n^L, f(\mathbf{x}_n^L)) + \frac{\lambda_2}{N_P} \sum_{n=1}^{N_P} D_2(g(\mathbf{x}_n^P), f(\mathbf{x}_n^P)). \quad (2)$$

2.3 疑似訓練サンプルの重み付け

先の模倣損失では、いずれの疑似訓練サンプルも同様に重要であることを仮定しているが、実際には目標モデルの学習に悪影響を与えるサンプルも含まれている。例えば、参照モデルの予測の不確実性 (uncertainty) が大きい場合、その参照モデルの予測を目標モデル学習の教師情報として用いることは望ましくない。そこで提案手法では、fidelity weighting [Dehghani 17] の考え方に基づく疑似訓練サンプルの適応的重み付けを導入する。Fidelity weighting では参照モデルの予測の不確実性に応じてモデル更新の学習率を調整しているが、本論文では模倣損失における疑似訓練サンプルの重みを調整する。

$$L_{\text{imi}}(\mathbf{X}^L, \mathbf{Y}^L, \mathbf{X}^P) = \frac{\lambda_1}{N_L} \sum_{n=1}^{N_L} D_1(\mathbf{y}_n^L, f(\mathbf{x}_n^L)) + \frac{1}{N_P} \sum_{n=1}^{N_P} \lambda_2(g, \mathbf{x}_n^P) D_2(g(\mathbf{x}_n^P), f(\mathbf{x}_n^P)). \quad (3)$$

ここで、 $\lambda_2(g, \mathbf{x}_n^P)$ は疑似訓練サンプル \mathbf{x}_n^P の重みであり、この疑似訓練サンプルを入力とした参照モデルの予測 $g(\mathbf{x}_n^P)$ の不確実性 $\sigma_g(\mathbf{x}_n^P)$ から以下のように計算される。

$$\lambda_2(g, \mathbf{x}_n^P) = \hat{\lambda}_2 \exp(-\bar{\lambda}_2 \sigma_g(\mathbf{x}_n^P) / \bar{\sigma}_g), \quad (4)$$

ここで、 $\hat{\lambda}_2$ は重みの上限値、 $\bar{\sigma}_2$ は不確実性の平均値、 $\bar{\lambda}_2$ は不確実性が平均値であったときの重みを示すパラメータである。上記の重みを計算するためには、参照モデルから予測の不確実性を求める必要がある。そこで本論文では、fidelity weighting と同様、ガウス過程 (Gaussian process; GP) 分類器を参照モデルとして採用する。

3. 疑似訓練サンプル最適化

ここまで、提案手法で用いる訓練損失について説明してきたが、高い予測性能を持つ目標モデルを学習するためには、疑似訓練サンプルの配置が非常に重要となる。本節では、疑似訓練サンプルの配置方法について述べる。

疑似訓練サンプルの考え方は、ガウス過程のスケラブルな推論を目的とする誘導点法 (inducing point method) [Snelson 06] から着想を得ている。誘導点は学習の過程で更新される疑似訓練サンプルであり、モデルパラメータと同様に、目的関数である経験尤度の下限 (evidence lower bound; ELBO) を最大化するように更新される。一方で、本手法では、目標モデルのパラメータは前述の模倣損失を小さくするように更新するが、疑似訓練サンプルは逆に模倣損失を大きくするように更新する。この戦略により、疑似訓練サンプルを現在の目標モデルの学習が十分に進んでいない領域へ移動させ、目標モデルの学習をより進めることを目指す。以下では、疑似訓練サンプルの更新方法について、具体的に説明する。

提案する疑似訓練サンプルの更新方法は、敵対的サンプルを用いた学習 [Szegedy 15] から着想を得ている。敵対的サンプルは、訓練サンプルに注意深く設計されたわずかな摂動を与えることで、元の訓練サンプルから得られるモデル予測とは大きく異なる予測を与えるサンプルのことを指す。最も基本的な敵対的なサンプルの生成手法として、最急勾配符号法 (fast gradient sign method; FGSM) [Goodfellow 15] が知られている。この方法では、所与の教師付訓練サンプル ($\mathbf{x}^L, \mathbf{y}^L$) から以下の式で敵対的サンプル $\mathbf{x}_{\text{AT}}(\mathbf{x}^L, \mathbf{y}^L)$ を計算する。

$$\mathbf{x}_{\text{AT}}(\mathbf{x}^L, \mathbf{y}^L) = \mathbf{x}^L + \epsilon \text{sign}\{\nabla_{\mathbf{x}^L} D_1(\mathbf{y}^L, f(\mathbf{x}^L))\}, \quad (5)$$

ここで $\nabla_{\mathbf{x}}$ は \mathbf{x} についての偏微分、 $\epsilon \geq 0$ は定数である。

上記の式 (5) は、定数 ϵ を学習率とみなし、勾配符号項 $\text{sign}\{\nabla_{\mathbf{x}^L} D_1(\mathbf{y}^L, f(\mathbf{x}^L))\}$ を単なる勾配 $\nabla_{\mathbf{x}^L} D_1(\mathbf{y}^L, f(\mathbf{x}^L))$ に置き換えると、確率的勾配降下法 (stochastic gradient descent method; SGD) によるモデルパラメータの更新則に極めて類似していることに気がつく。ただし、この更新は目的関数 $D_1(\mathbf{y}^L, f(\mathbf{x}^L))$ を増加させる方向に進むことに注意する。教師付訓練サンプル \mathbf{x}^L に代えて疑似訓練サンプル \mathbf{x}^P を、教師情報 \mathbf{y}^L に代えて参照モデルの出力 $g(\mathbf{x}^L)$ をそれぞれ用いると、以下の式によって疑似訓練サンプル \mathbf{x}^P を更新できる。

$$\mathbf{x}_{\text{imi}}(\mathbf{x}^P) = \mathbf{x}^P + \epsilon \nabla_{\mathbf{x}^P} D_2(g(\mathbf{x}^P), f(\mathbf{x}^P)). \quad (6)$$

上記は最急勾配符号法及び確率的勾配降下法を基礎とする更新式であるが、深層学習で発展した各種の確率的最適化手法や、対象サンプルが画像である場合にはより自然な敵対的サンプルを生成する方法 [Zhao 17] などを適用することも可能である。

目標モデルのパラメータと疑似訓練サンプルとを交互に更新する学習手順は、敵対的生成モデル (generative adversarial networks; GAN) の生成器と識別器との交互更新と方針が似ており、このような敵対的なモデル学習は一般的に不安定になりやすいことが知られている [Salimans 16]. そこで、本論文では、疑似訓練サンプルを逐次更新するのではなく、実際のモデル学習に用いるサンプルと更新するサンプルとを分離して、適切なタイミングで更新用サンプルを学習用サンプルに統合する方法を採用する. 疑似訓練サンプルの更新を含む、提案手法の具体的なアルゴリズムは、以下の通りとなる.

1. (少数の) 教師付訓練サンプル (X^L, Y^L) を用いて参照モデル $g(\cdot)$ を学習する.
2. 初期疑似訓練サンプル $X^{P(0)}$ 及びそのコピー $X^{P(1)}$ を教師付訓練サンプル X^L から作成, ステップを表現するインデックスを $t=0$ とする.
3. 教師付訓練サンプル (X^L, Y^L), 現在の疑似訓練サンプル $X^{P(t)}$ 及び参照モデル $g(\cdot)$ を用いて, 目標モデル $f(\cdot)$ のモデルパラメータ θ_f を更新 (式 (3)).
4. 参照モデル $g(\cdot)$ 及び現在の目標モデル $f(\cdot)$ を用いて, 次の疑似訓練サンプル $X^{P(t+1)}$ を更新 (式 (6)).
5. 3-4 を所定のエポック数だけ繰り返す.
6. 次の疑似訓練サンプル $X^{P(t+1)}$ のコピー $X^{P(t+2)}$ を作成, 現在の疑似訓練サンプル $X^{P(t)}$ を次の疑似訓練サンプル $X^{P(t+1)}$ に統合 $X^{P(t+1)} \leftarrow X^{P(t)} \cup X^{P(t+1)}$, インデックスをインクリメント $t \leftarrow t+1$.
7. 3-6 を所定のステップ数だけ繰り返す.

4. 実験

4.1 定性的評価

提案手法の挙動を確認するために、まず最初に 2 次元 2 値分類データセットである Banana dataset *1 を用いた実験を行った. 総計 400 個の 2 次元サンプルを事前に固定したランダム行列により 100 次元空間に線形変換を行い, 各クラスから 5 個ずつサンプルを選択し, この 10 個のサンプルを教師付訓練サンプルとして採用した. 図 2 (a) に, 教師付訓練サンプル (橙・青×印, 色がクラスに対応) 及びその他の (未使用の) サンプル (橙・青点) の配置を示す. 放射基底関数 (radial basis function; RBF) カーネルを用いたガウス過程分類器を参照モデルとして採用し, 変分ガウス近似 [Opper 09] を用いてモデル学習及びカーネルパラメータ最適化を行った. 目標モデルとして, 7 層の全結合層からなる NN を採用し, 中間層のユニット数はすべて 1,000 とした. 最適化手法として, モデルパラメータの更新には Nesterov 加速勾配を用いた Adam (Nadam) を, 疑似訓練サンプルの更新には Adam をそれぞれ使い, 初期学習率をそれぞれ 0.001 及び 0.05 に設定した. NN 学習のバッチサイズは 100, エポック数は 200 とした.

まず, 最も極端な例として, 図 2 (b) に示すように, 特徴空間をグリッド上に敷き詰めるように疑似訓練サンプルを配置した場合を検証した. この設定では, 疑似訓練サンプルを最適化せず, 予測の不確実性に基づく疑似訓練サンプルの重み付けも行わなかった. 損失関数として, D_2 には KL ダイバージェンスを用い, D_1 は使用しなかった. すなわち, この設定では, 教師付訓練サンプルを目標モデルの学習に用いなかった. 図 2 (c)(d) に参照モデルと目標モデルの識別境界をそれぞれ示す. この図から, 提案手法を用いることで, 極限的には参照モデル

表 1: MNIST (上) 及び fashion MNIST (下) に対する分類性能. “Distill”, “opt” 及び “fidelity” はそれぞれ, 模倣損失の利用, 疑似訓練サンプル最適化, 予測の不確実性を考慮した疑似訓練サンプルの重み付けに対応.

Methods / #labeled	10	20	50	100	200
NN plain	37.9	46.0	66.0	78.3	86.7
GP plain	39.9	51.6	64.6	73.2	80.0
Distill	43.5	51.2	67.7	78.1	86.1
Distill, opt	44.1	53.5	70.0	79.3	86.7
Distill, opt, fidelity	44.1	53.9	70.4	80.0	86.4

Methods / #labeled	10	20	50	100	200
NN plain	39.3	47.9	58.3	64.9	71.3
GP plain	44.6	52.4	59.9	65.7	71.4
Distill	43.6	50.9	60.0	67.3	72.5
Distill, opt	41.2	49.6	60.1	67.3	72.2
Distill, opt, fidelity	44.8	52.7	62.1	68.0	72.5

の挙動をほぼ完全に目標モデルに転移することができることがわかる.

続いて, より現実的な例として, 教師付訓練サンプルを用いて 250 個の初期疑似訓練サンプルを生成し, 提案手法のアルゴリズムを用いて最終的に 1,000 個まで疑似訓練サンプルを増やす場合を検証した. すなわち, 目標モデル学習は 4 ステップに別れ, 各ステップではそれぞれ 250, 500, 750 及び 1000 個の疑似訓練サンプルを用いて 50 エポックの学習を行った. 初期疑似訓練サンプルは, 半分を教師付訓練サンプルの線形内挿により, 残り半分を教師付訓練サンプルに小さなガウスノイズを加えることにより, それぞれ生成した. 図 2 (e) に初期疑似訓練サンプルの配置を示す. 損失関数として, D_1 にはヒンジロス, D_2 には KL ダイバージェンスを用い, 両者の重み λ_1 及び λ_2 はいずれも 1.0 とした. 図 2 (f)(g)(h) に学習済目標モデルの識別境界及び最終的な疑似訓練サンプルの配置を示す. (f)(g)(h) はそれぞれ, 疑似訓練サンプルのみを学習に用いた場合, 加えて教師付訓練サンプルを用いた場合, さらに予測の不確実性に基づく重み付けを用いた場合を示している. いずれの場合においても, 疑似訓練サンプルが特徴空間を覆い尽くしていなくても, 参照モデルの挙動をある程度目標モデルに転移できていることがわかる. また, 図 2 (g)(h) から, 教師付訓練サンプルを学習に含めることで, 参照モデルの識別境界をできる限り維持しつつ, 教師付訓練サンプルの予測をより良くできていることがわかる (中央下部の橙×印に着目).

4.2 定量的評価

次に, 標準的ベンチマークである MNIST [Lecun 98] 及び fashion MNIST [Xiao 17] を用いて, 提案手法の分類性能を定量的に評価した. 参照モデルとして再びガウス過程分類器を用い, 参照モデルに関する設定はすべて前節と同じものを用いた. 目標モデルとして, 3 層の畳み込み NN を用い, 各畳み込み層はカーネルサイズを 3×3 , ストライドを 2 (奇数層) もしくは 1 (偶数層), 第 l 層のチャンネル数を $16 \times (\text{floor}(l/2) + 1)$ と設定した. また, 各畳み込み層の後に batch normalization, pReLU 活性化層 及び dropout (0.5) を導入すると共に, 最終畳み込み層の後に 2 層の全結合層 (batch normalization, pReLU 及び dropout を第 1 層の後に追加) を配置した. 最適化手法は前節と同様であり, 初期学習率をいずれも 0.02 に設定した. 各データセットに含まれる 60,000 個の訓練サンプルの中から各クラス 1 から 20 個のサンプルをランダムに 20 通り選択し, これを教師付訓練サンプルとして用いた. また, 教師付訓練サンプルの線形内挿により 1,250 個の初期疑似訓練サンプルを生成し, 提案手法のアルゴリズムにより最終的に 10,000 個まで疑似訓練サンプルを増やした. すなわち, 目標モデル学習は 8 ステップで構成され, 各ステップで 25 エポ

*1 <https://github.com/GPflow/GPflow/tree/master/doc/source/notebooks/data>

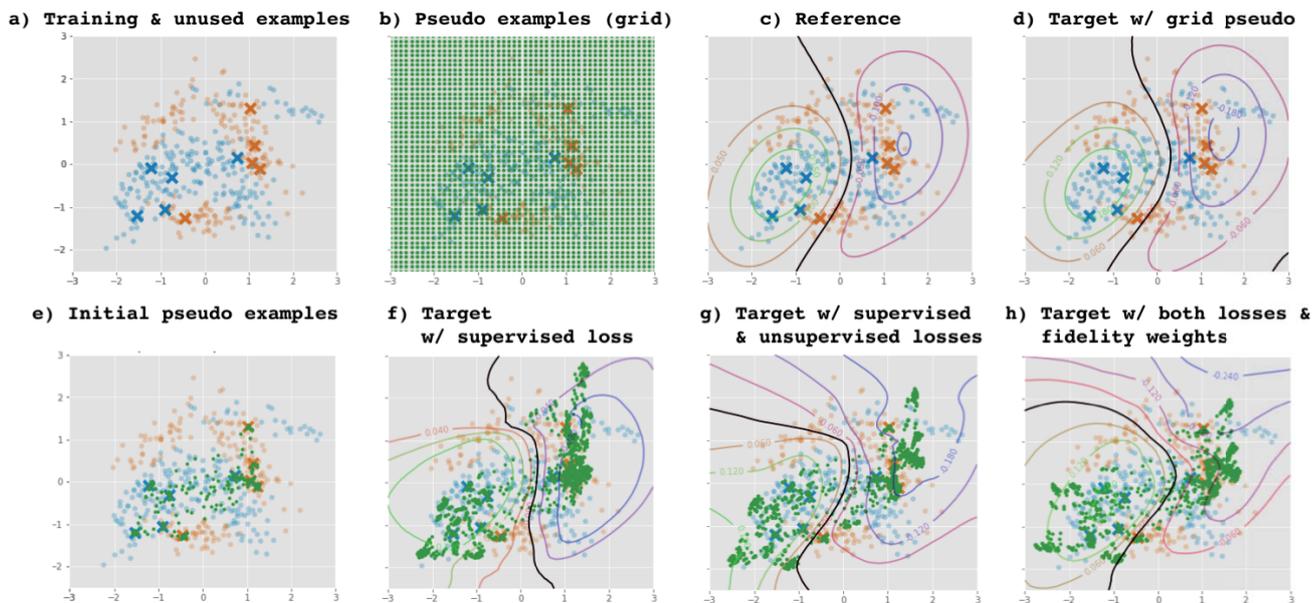


図 2: Banana データセットを用いた際の提案手法の予測結果の可視化。×印: 教師付訓練サンプル, 橙・青点: データセット内の他のサンプル (学習未使用), 緑点: 疑似訓練サンプル, 黒線: 分類境界。

クの学習を行った。損失関数として、 D_1 にはヒンジロス、 D_2 にはKLダイバージェンスを用い、重み λ_1 は1.0に、 $\bar{\lambda}_2$ は教師付訓練サンプルの増加に伴って100.0から1.0まで徐々に減らした。評価は、各データセットに含まれる10,000個のテストサンプルを用いて、20通りの教師付訓練サンプルの選択に対する分類正解率の平均値で評価した。その他の実験設定は、前節と同様である。この実験では、以下の5手法を比較する。(1) 教師付訓練サンプルのみで学習したNN分類器、(2) 参照モデルであるガウス過程分類器、(3) 固定した疑似訓練サンプル10,000個と模倣損失を用いて学習した目標モデル、(4) 模倣損失と疑似訓練サンプルの更新を用いて学習した目標モデル、(5) 4に加えて予測の不確実性を考慮した疑似訓練サンプルの重み付けを用いて学習した目標モデル。

表1に実験結果を示す。この結果から、提案手法(3)-(5)のいずれもが、単純なNN学習を上回るもしくはほぼ同等の分類性能を発揮できたことがわかる。また、提案手法は、参照モデルの挙動を目標モデルに転移する戦略を採用しているにもかかわらず、ほぼすべての場合において参照モデルの分類性能を同等もしくは上回る分類性能を示したことがわかる。さらに、MNISTデータセットでは、教師付訓練サンプルの数が増えると、教師付訓練サンプルのみで学習したNN分類器が、参照モデルであるガウス過程分類器よりも優れた分類性能を持つ。この設定は、参照モデルの挙動を模擬しようとする提案手法にとって望ましくないが、このような状況であっても、提案手法はさらに優れた分類性能を示した。提案手法の個別要素について確認すると、模倣損失の導入のみでもある程度の分類性能を担保できるが、疑似訓練データの最適化と予測の不確実性に基づく重み付けを導入することでさらに分類性能を向上できる。

5. Conclusion

本論文では、少量の訓練データのみからNNを学習するための手法として、疑似訓練サンプルの導入及びその最適化に基づく手法を提案した。提案手法の基本的な枠組は非常に一般的であり、予測の不確実性に基づく疑似訓練サンプルの重み付けを除けば、どのようなタイプの参照モデル・目標モデルにも適

用が可能である。例えば、従来の知識蒸留と同様に、深いNNモデルを参照モデル、浅いNNモデルを目標モデルとすることも可能であり、その逆の場合にも適用可能である。さらには、SVMやrandom forestsなど、別の種類の予測器を参照モデルとして利用することも可能である。疑似訓練サンプルの最適化についても、最新の最適化手法や敵対的サンプル生成手法を導入することで、さらなる性能向上の可能性が見込める。

参考文献

- [Buciluă 06] Buciluă et al.: Model Compression, in *Proc. KDD.* (2006)
- [Caruana 94] Caruana et al.: Learning Many Related Tasks at the Same Time with Backpropagation, in *Proc NIPS* (1994).
- [Dehghani 17] Dehghani et al.: Fidelity-Weighted Learning, *ArXiv* (2017)
- [Goodfellow 14] Goodfellow et al.: Generative Adversarial Nets, in *Proc. NIPS.* (2014)
- [Goodfellow 15] Goodfellow et al.: Explaining and Harnessing Adversarial Examples, in *Proc. ICLR.* (2015)
- [Hinton 15] Hinton et al.: Distilling the Knowledge in a Neural Network, in *NIPS Workshop.* (2015)
- [Koch 15] Koch et al.: Siamese Neural Networks for One-shot Image Recognition, in *ICML Workshop.* (2015)
- [Lecun 98] Lecun et al.: Gradient-based learning applied to document recognition, *Proc. IEEE.* (1998)
- [Opper 09] Opper et al.: The Variational Gaussian Approximation Revisited, *Neural Computation.* (2009)
- [Salimans 16] Salimans et al.: Improved Techniques for Training GANs, in *Proc. NIPS.* (2016)
- [Snelson 06] Snelson et al.: Sparse Gaussian Processes using Pseudo-inputs, in *Proc. NIPS.* (2006)
- [Szegedy 15] Szegedy et al.: Going Deeper with Convolutions, in *Proc. CVPR.* (2015)
- [Xiao 17] Xiao et al.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, *ArXiv* (2017)
- [Zhao 17] Zhao et al.: Generating Natural Adversarial Examples, *ArXiv* (2017)