

1. はじめに

複数の深層学習モデルによるアンサンブル学習では、各ネットワークの出力分布の平均から推論を行うことで、1モデルの推論と比べて認識性能が向上する。このとき、各ネットワークの出力分布間に多様性が生まれると、アンサンブルの効果が高くなると考えられる。しかし、ネットワーク間で出力分布を離すように学習すると、出力分布の多様化が期待できるが、性能低下が生じる。

そこで本研究では、性能低下を引き起こすことなく多様性を得るために Attention map に着目し、出力分布と Attention map を知識とした知識蒸留を複数のネットワーク間で行う。そのために、従来の知識蒸留にネットワーク間の知識を離す設計を導入する。これにより、アンサンブル学習に適した多様な知識蒸留を実現できる。また、知識蒸留を用いたアンサンブル学習にグラフ表現 [2] を導入することで、多様なアンサンブル学習方法を獲得可能とする。

2. 知識蒸留

知識蒸留 [1] は、ネットワークが獲得した知識を他のネットワークが模倣することで、知識を転移する学習手法である。知識の転移方向として一方向や双方向、知識として確率分布や特徴マップ、Attention map など様々な方法が提案されている。南ら [2] は、知識蒸留をグラフで表現し、知識の転移方法をハイパーパラメータとして最適化することで、従来の様々な知識蒸留方法を内包し、その中から効果的な知識蒸留方法を自動設計可能な手法を提案した。従来の知識の転移方法は、ネットワークの認識性能が向上する一方、ネットワーク間で多様性が生まれないため、知識蒸留を行わないネットワークと比べて同程度のアンサンブル性能となる。

3. 提案手法

本研究では、ネットワークの多様性を獲得するために、確率分布と Attention map の観点から各ネットワークが異なる知識を獲得できるような知識蒸留を行う。また、グラフ表現 [2] をアンサンブル学習に拡張し、グラフ構造をハイパーパラメータ最適化することでモデル数に応じた効果的なアンサンブル学習方法を自動設計する。

3.1 多様な知識蒸留を用いたアンサンブル学習

アンサンブル学習のための知識蒸留として、知識を近づける損失と離す損失を設計する。損失設計は、最小化問題として学習を行うために近づける・離す場合で異なる設計を用いる。この時、知識の転移先をターゲットネットワーク、知識の元をソースネットワークとする。

確率分布を近づける場合は KL-divergence、離す場合はコサイン類似度を利用する。KL-divergence を用いた損失関数は次のように定義する。

$$L_p(x) = \sum_{c=1}^C p_s^c(x) \log \frac{p_s^c(x)}{p_t^c(x)} \quad (1)$$

ここで、 C はクラス数、 x は入力サンプル、 $p_s^c(\cdot)$ はソースネットワークのクラス c の確率値、 $p_t^c(\cdot)$ はターゲットネットワークのクラス c の確率値である。コサイン類似度を用いた損失関数は次のように定義する。

$$L_p(x) = \frac{p_s(x)}{\|p_s(x)\|_2} \cdot \frac{p_t(x)}{\|p_t(x)\|_2} \quad (2)$$

ここで、 $p_s(\cdot)$ はソースネットワークの確率分布、 $p_t(\cdot)$ はターゲットネットワークの確率分布である。

Attention map は、入力サンプルの認識に有効な領域に強く反応する。対象物体の大きさはサンプルごとに異なるため、対象物体の異なる箇所に強く反応しているのに関わらず、類似度が高くなる場合がある。そこで、Attention map のクロップを行う。ソースネットワークの Attention

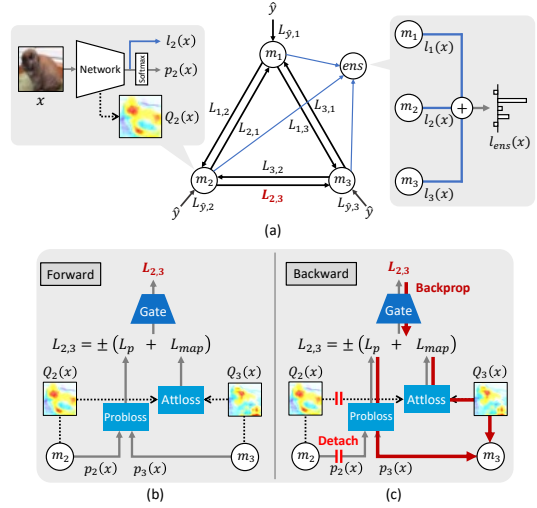


図 1: グラフ表現を導入した多様な知識蒸留を用いたアンサンブル学習

map において最も値の高い位置を中心に正方形のクロップを行い、ターゲットネットワークはソースネットワークと同じ位置をクロップする。複数のサイズでクロップを行い、それぞれのサイズにおける損失値の平均値を最終的な損失値とする。Attention map を近づける場合は平均二乗誤差、離す場合はコサイン類似度を利用する。平均二乗誤差を用いた損失関数は次のように定義する。

$$L_{map}(x) = \frac{1}{K} \sum_{k=1}^K \left(\frac{Q_s^k(x)}{\|Q_s^k(x)\|_2} - \frac{Q_t^k(x)}{\|Q_t^k(x)\|_2} \right)^2 \quad (3)$$

ここで、 K はクロップ数、 $Q_s^k(\cdot)$ はクロップしたソースネットワークの Attention map、 $Q_t^k(\cdot)$ はクロップしたターゲットネットワークの Attention map である。コサイン類似度を用いた損失関数は次のように定義する。

$$L_{map}(x) = \frac{1}{K} \sum_{k=1}^K \frac{Q_s^k(x)}{\|Q_s^k(x)\|_2} \cdot \frac{Q_t^k(x)}{\|Q_t^k(x)\|_2} \quad (4)$$

3.2 アンサンブル学習のためのグラフ表現

アンサンブル学習にとって効果的な知識蒸留方法を自動設計するために、グラフ表現 [2] を利用する。グラフ表現を導入した知識蒸留を用いたアンサンブル学習を図 1(a) に示す。グラフはノードとエッジで構成される。ノードは、ネットワークを表すネットワークノードとアンサンブルを行うアンサンブルノードを定義する。エッジは、損失計算を表し、ネットワークノード間のエッジは知識蒸留、ネットワークノードとラベル $\hat{y}$ 間のエッジは教師ラベルを用いた交差エントロピー損失を表す。

アンサンブルノードは、全てのネットワークノードの出力を用いたアンサンブルを行う。アンサンブルノードにおける処理は次のように定義する。

$$l_{ens} = \frac{1}{M} \sum_{m=1}^M l_m(x) \quad (5)$$

ここで M はネットワークノード数、 $l_m(\cdot)$ はネットワークノードの logit 関数、 x は入力サンプルである。

ネットワークノード間のエッジは、ノード間の知識蒸留を行う。エッジでは、図 1(b) のように知識蒸留の損失値に対してゲート [2] を用いて重み付けを行うことで、知識蒸留を制御する。知識蒸留の損失計算は次のように定義する。

$$L_{s,t}(x) = G_{s,t}(L_p(x) + L_{map}(x)) \quad (6)$$

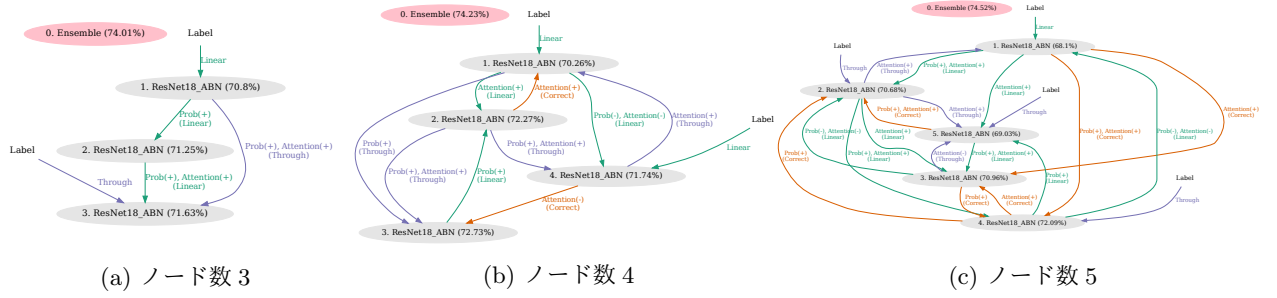


図 2: 最適化後のグラフ構造: 赤色のノードはアンサンブルノード, 灰色のノードはネットワークノード, “Label” は教師ラベル, “Prob” と “Attention” は損失設計の種類を表す. エッジの色はゲートの種類を表す.

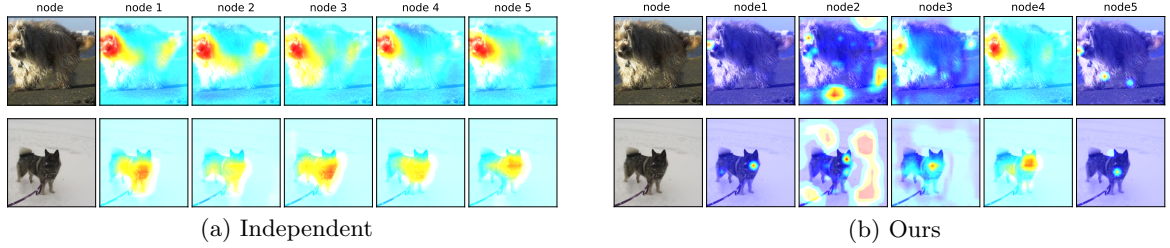


図 3: Attention map の可視化

表 1: 従来手法との精度比較

手法	ネットワーク	精度 [%]	
		ノード平均	アンサンブル
Independent	ABN × 3	68.04 ± 0.28	71.41 ± 0.34
DML	ABN × 3	70.50 ± 0.26	72.08 ± 0.42
Ours	ABN × 3	70.95 ± 0.16	73.41 ± 0.30
Independent	ABN × 4	68.30 ± 0.27	72.06 ± 0.53
DML	ABN × 4	71.50 ± 0.31	72.87 ± 0.29
Ours	ABN × 4	71.46 ± 0.22	74.16 ± 0.22
Independent	ABN × 5	68.24 ± 0.26	72.32 ± 0.18
DML	ABN × 5	71.15 ± 0.28	72.50 ± 0.16
Ours	ABN × 5	70.23 ± 0.33	74.14 ± 0.50

表 2: 異なるデータセットにおけるアンサンブル精度

学習・評価用 データセット	最適化用 データセット	ノード数		
		3	4	5
CUB-200-2011	Stanford Dogs	71.82	73.03	72.13
CUB-200-2011	CUB-200-2011	74.17	72.22	74.05
Stanford Cars	Stanford Dogs	89.94	89.98	90.41
Stanford Cars	Stanford Cars	90.04	89.57	90.73

ここで $G_{s,t}(\cdot)$ はゲートを表す. 知識蒸留の損失から計算された勾配は, エッジの向きによって勾配を伝えるノードが変化する. $L_{2,3}$ の場合は図 1(c) のように, ノード m_3 にのみ勾配を伝えるためにノード m_2 側の計算グラフをカットすることで, ノード m_2 からノード m_3 への知識蒸留が行われる.

グラフのハイパーパラメータは, ネットワークノード間のエッジにおける損失設計, 各エッジの損失値に適用されるゲートである. 損失設計は, 確率分布を近づける・離す, Attention map を近づける・離す, 確率分布と Attention map を同時に近づける・同時に離すの計 6 種類とする. ゲートは, 従来法 [2] で提案された 4 種類とする. ハイパーパラメータ最適化は, ランダムサーチと枝刈りを用いて 6,000 組のグラフ構造を評価する.

4. 評価実験

ネットワークノード数を 3 から 5 としてハイパーパラメータ最適化をしたグラフの評価を行う. ネットワークは, Attention Branch Network(ABN) [3] を用いる. データセットは, 詳細画像識別問題である Stanford Dogs, Stanford Cars, Caltech-USCD Birds-200-2011(CUB-200-2011) を用いる. 損失計算時に Attention map は, 3×3 , 7×7 , 11×11 のサイズにクロップする.

4.1 最適化したグラフの評価

Stanford Dogs を用いて最適化を行ったグラフ構造を図 2, Stanford Dogs における従来手法との比較を表 1 に示す. Independent は個別に学習を行ったネットワーク, DML は従来の知識蒸留を行ったネットワーク, Ours は最適化したグラフの結果である. Ours は, Independent や DML を

上回るアンサンブル精度を達成した.

Independent の Attention map を図 3(a), 図 2(c) のグラフにおける Ours の Attention map を図 3(b) に示す. Ours は, 多様な知識蒸留によってネットワークごとに異なる注視領域を獲得したと言える.

4.2 グラフ構造の汎化性

Stanford Dogs を用いて最適化したグラフを最適化時と異なるデータセットを用いて評価する. CUB-200-2011, Stanford Cars における評価結果を表 2 に示す. 最適化に用いたデータセットによらず同程度のアンサンブル精度となっていることから, 最適化したグラフ構造に汎化性があると言える.

5. おわりに

本研究では, アンサンブル学習のための多様な知識蒸留とグラフ表現を提案した. 評価実験により, 従来の知識蒸留やアンサンブル学習を超える性能を発揮するアンサンブル学習方法を獲得したことを確認した. 今後は, 獲得したグラフ構造の分析やグラフの最適化方法について検討する.

参考文献

- [1] G. Hinton, *et al.*, “Distilling the knowledge in a neural network”, NeurIPS workshop, 2015.
- [2] S. Minami, *et al.*, “Knowledge Transfer Graph For Deep Collaborative Learning”, ACCV, 2020.
- [3] H. Fukui, *et al.*, “Attention Branch Network: Learning of Attention Mechanism for Visual Explanation”, CVPR, 2019.

研究業績

- [1] 岡本直樹 等, “知識蒸留グラフによるアンサンブル学習”, 画像の認識・理解シンポジウム, 2020.
- [2] 岡本直樹 等, “詳細画像識別における知識蒸留グラフによるアンサンブル学習”, 画像の認識・理解シンポジウム, 2021.