【CVPR2024】Adaptive Random Feature Regularization on Fine-tuning Deep Neural Networks【論文解説】

※下記ブログにも掲載した内容より抜粋・一部改変して転載しています

https://qiita.com/hotekagi/items/5ba59dd0aac57d4d0030

要点

- Fine-tuningにおいて、特徴空間上の混合ガウス分布を用いた新たな正則化手法を提案
- 事前学習データの情報が無いにも関わらず、それを用いた手法も含んだ多数の手法より精度や計算コストの優位性を確認
- 勾配の大きさや特徴空間の様子、相互情報量の観点からも有効性を観察

論文の背景と目的

- 大規模なResNetやViT、CLIPの出現により、事前学習された深層学習モデルをtarget datasetにfine-tuningすることが一般的
- target datasetが小さくとも、source domainの知識を有効活用してtarget domainに対して優れた汎化性能を発揮することもあれば、一方でtarget datasetにoverfittingしてしまうこともある

→ そのため適切な正則化手法を用いることが重要

いくつかの正則化手法はsource datasetやlabelなどの情報を必要とするが、sourceとtargetでタスクの種類が異なる場合(自己教師あり→分類)やsource datasetが公開されていない場合(CLIP)では利用できない

→ sourceの情報をせずに、かつ軽い計算コストで適用可能な正則化手法が必要

問題設定

- Kクラス分類
- 予測器$f_\theta : \mathcal{X} \to \mathcal{Y}$

ここで、
f_θは特徴抽出器$g_\phi : \mathcal{X} \to \mathbb{R}^d$と出力直前の線形層$W \in R^{d \times K}$に分かれるとし、
$f_\theta = W^T g_\phi$、訓練可能なパラメータは$\theta = [\phi,\ W]$

事前学習によりパラメータの初期値
$\theta_s = [\phi_s,\ W_s]$が得られており、
そこからtarget dataset$\mathcal{D} = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^N$を用いて訓練

既存の正則化手法

target datasetにoverfittingさせないためには、事前学習時の知識を上手く保持することが重要だと考えられる

source domainの情報を利用しない正則化手法

- **Feature Norm Penalty (FNP) [Hariharan and Girshick, ICCV, 2017]**
  - $g_\phi(x)$のL1/L2ノルムで正則化
- **L^2-SP [Li et al., ICML, 2018]**
  - パラメータについてStarting PointからのL2ノルム$||\theta - \theta_s||_2^2$で正則化
- **DELTA [Li et al., ICLR, 2019]**
  - f_θの中間層における出力が、$\theta = \theta_s$のときとあまり変わらないようにL2正則化
- **Batch Spectral Shrinkage (BSS) [Chen et al., NeurIPS, 2019]**
  - ミニバッチ内の特徴ベクトルを並べた行列の固有値に対する正則化
- **RandReg [Zhong and Maki, CVPR, 2020]**
  - 後述
- **Contrast-regularized tuning (Core-tuning) [Zhang et al., NeurIPS, 2021]**
  - 対照学習に基づき、特徴ベクトルに対してfocal contrastive lossを適用

- **DR-Tune [Zhou et al., ICCV, 2023]**
  - Wが元のf_{θ_s}の特徴ベクトルの分布を利用しても分類できるように正則化項を付け加える

source domainの情報を利用した正則化手法

- **Co-tuning [You et al., NeurIPS, 2020]**
  - 元のsource特化のヘッドW_sの情報を活用する
    - 通常のfine-tuningでは(出力次元が一致しないこともあり)元のヘッドは破棄され新たに初期化される
  - sourceとtargetのラベルと関係を鑑みて、source datasetをソフトラベルによるpseudo source datasetにマッピングし、target datasetと同時に学習
- **Unbalanced Optimal Transport (UOT) [Liu et al., NeurIPS, 2022]**
  - 最適輸送に基づくsourceとtargetのマッピングにより、target datasetに関連する部分的なsource datasetを同時に学習
- **Borrowing Treasures From the Wealthy [Ge and Yu, CVPR, 2017]**
  - targetに関連するsource datasetを用いる

RandReg

- 特徴ベクトルと、あるランダムなベクトルの差の最小化を目指す
- ある分布$p(z)$を用意し、 $||g_\theta(x) - z||_2^2$ で正則化
  - $z = 0$ の場合にはFNPと一致
  - 勾配における更新式を展開すれば、FNPの正則化の項と、摂動により局所解に陥りにくくする項と見做せる

RandRegの問題点

- $p(z)$の設定を上手く行う必要がある
  - $g_\phi(x)$が実際にどれくらいのノルムになるかはデータセットやモデルごとに異なる
    →
    ϕ_sにおけるtarget datasetに対する特徴ベクトルの平均 $\bar{\mu}_s$ 分散 $\bar{\sigma}_s^2$ で初期化したガウス分布を用いる

- それでも上手く動かずFNPに劣る場合がある
  - $g_\phi(x)$ のノルムが特に小さくなる場合

Pre-trained Method	$\|g_{\phi_0}(x)\|_2^2$	$(\bar{\mu}_s, \bar{\sigma}_s^2)$	Fine-tuning	FNP [10]	RandReg-$U(0,1)$	RandReg-$\mathcal{N}(0,1)$	RandReg-$\mathcal{N}(\mu_s, \sigma_s^2)$
ImageNet Classification	19.58	$(4.18\times10^{-1}, 2.69\times10^{-1})$	$89.14^{\pm.42}$	$90.27^{\pm.10}$	$90.59^{\pm.24}$	$90.42^{\pm.12}$	$\mathbf{90.61}^{\pm.24}$
ImageNet SimCLR [4]	1.34	$(2.60\times10^{-2}, 3.97\times10^{-2})$	$83.73^{\pm.73}$	$\mathbf{84.53}^{\pm.32}$	$84.08^{\pm.21}$	$84.03^{\pm.07}$	$83.91^{\pm.04}$
ImageNet Barlow Twins [34]	3.67	$(5.53\times10^{-2}, 6.94\times10^{-2})$	$86.98^{\pm.16}$	$87.44^{\pm.15}$	$87.24^{\pm.30}$	$\mathbf{87.74}^{\pm.33}$	$87.65^{\pm.48}$
CLIP [27]	11.70	$(5.92\times10^{-4}, 3.12\times10^{-2})$	$88.72^{\pm.24}$	$89.96^{\pm.05}$	$90.19^{\pm.40}$	$90.59^{\pm.24}$	$\mathbf{90.78}^{\pm.07}$

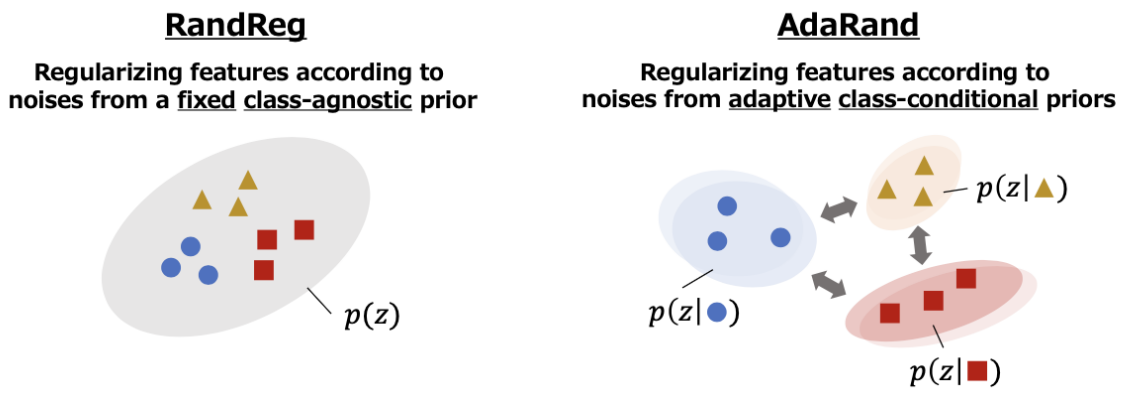
RandRegでは後述に実験のように、
$g_\phi(x)$のノルムが制限され、特徴ベクトルの多様性が失われてしまう

- $||g_\phi(x)||_2^2$の減少
  - cross entropy誤差を用いる場合、$g_\phi(x)$のノルムが小さいとWに関する勾配のノルムも小さくなり、上手くtarget datasetを学習できない恐れがある
- 特徴ベクトルに関するdifferential entropy $H(g_\phi(x))$の減少
  - $X \subset \mathbb{R}^d$上の確率密度が$f(x)$であるときのdifferential entropy
    - $H(X) = -\int f(x) \log f(x) dx$
  - これを手元のdatasetから推定するdifferential entropy estimatorを考える
  - x_iまわりのϵ-ball: $p(\epsilon) = \int_{||x-x_i||<\epsilon} p(x)dx$が定数になる近似できるとき、differential entropy estimatorは$H(X) = const + \frac{d}{n(n-1)} \sum_{i \neq j} ||x_i - x_j||$ で表される [Faivishevsky and Goldberger, NeurIPS, 2008]
  - entropyが減少すると、相互情報量が減少する
    - $I(g_\varphi(x); y) = H(g_\varphi(x)) - H(g_\varphi(x)|y)$
    - 相互情報量は、全体では多様(第1項)ながら各ラベルに関しては密集しているとき、つまり各ラベルごとのクラスタが離れていると大きくなる

論文の提案手法：AdaRand

RandRegでは1つの固定の$p(z)$を使っていたところに混合ガウス分布を用いる。

$p(x) = \sum_{k=1}^K p(z|y_k)p(y_k),\ \ p(z|y_k) = \mathcal{N}(\mu_k, \sigma_k^2 I)$



パラメータμ_k, σ_k^2は$\{(g_{\phi_s}(x_i), y_i)\}_{i=1}^N$で初期化されたのち、$\mu_k$は以下の目的関数を最小化するように更新されていく。

1. 現在のバッチにおける各クラスの平均$\bar{\mu}_k$を移動平均により計算する

$$\hat{\mu}_k = \frac{1}{B_k}\sum_{i=1}^{B_k} g_\varphi(x_i),\ \bar{\mu}_k \leftarrow \alpha\bar{\mu}_k + (1-\alpha)\hat{\mu}_k$$

2. 距離$D(u,v)$（たとえばコサイン距離 $1 - u^T v/||u||_2||v||_2$）を用いて目的関数を計算する

$$\mathcal{L}_{ada} = \frac{1}{K}\sum_{k=1}^{K} D(\mu_k, \bar{\mu}_k) - \frac{1}{K(K-1)}\sum_k \sum_{k\neq l} D(\mu_k, \mu_l)$$

3. 勾配法によりμ_kを更新

目的関数の第1項は分布の平均を現在のデータに対する特徴ベクトルの平均に近づけるように、第2項は分布間を離すようにしている。

アルゴリズム全体としては、各ミニバッチに対して

1. ランダムベクトルの生成
   - $z_i \sim p(z)$
2. モデルのパラメータθの更新
   - 目的関数：$CE(y_i, f_\theta(x_i)) + \lambda||g_\phi(x_i) - z_i||_2^2$
   - 勾配法
3. $p(z)$のパラメータ$\bar{\mu}$の更新
   - 目的関数：$\mathcal{L}_{ada} = \frac{1}{K}\sum_{k=1}^{K} D(\mu_k, \bar{\mu}_k) - \frac{1}{K(K-1)}\sum_k \sum_{k\neq l} D(\mu_k, \mu_l)$

- 勾配法

実験結果

- どの事前学習モデルに対してもAdaRandが他の(source datasetを用いる手法も含めた)手法を抑えて最も精度が良い

Table 2. Top-1 test accuracy (%) of various combinations of pre-training methods and neural network architectures (Cars).

Pre-training Method	Architecture	Fine-tuning	FNP [10]	DR-Tune [38]	RandReg-Best	AdaRand (Ours)
ImageNet Classification	RN-50	$89.14^{\pm.42}$	$90.27^{\pm.10}$	$90.38^{\pm.59}$	$90.61^{\pm.24}$	$\mathbf{91.17}^{\pm.13}$
ImageNet Classification	ViT-B/32	$78.56^{\pm1.3}$	$81.77^{\pm.21}$	$79.49^{\pm.51}$	$82.46^{\pm.20}$	$\mathbf{83.84}^{\pm.13}$
ImageNet Classification	ViT-B/16	$87.35^{\pm.53}$	$88.75^{\pm.44}$	$88.19^{\pm.26}$	$88.88^{\pm.26}$	$\mathbf{89.54}^{\pm.17}$
ImageNet SimCLR [4]	RN-50	$83.73^{\pm.73}$	$84.53^{\pm.32}$	$84.05^{\pm.17}$	$84.08^{\pm.21}$	$\mathbf{85.51}^{\pm.05}$
ImageNet Barlow Twins [34]	RN-50	$86.98^{\pm.16}$	$87.44^{\pm.15}$	$86.69^{\pm.23}$	$87.74^{\pm.33}$	$\mathbf{88.23}^{\pm.39}$
CLIP [27]	RN-50	$88.72^{\pm.24}$	$90.19^{\pm.40}$	$90.16^{\pm.22}$	$90.78^{\pm.07}$	$\mathbf{91.25}^{\pm.63}$
CLIP [27]	ViT-B/32	$83.56^{\pm.60}$	$85.79^{\pm.52}$	$85.71^{\pm.59}$	$86.83^{\pm.34}$	$\mathbf{87.40}^{\pm.48}$
CLIP [27]	ViT-B/16	$90.35^{\pm.23}$	$91.24^{\pm.03}$	$90.47^{\pm.26}$	$91.33^{\pm.44}$	$\mathbf{92.84}^{\pm.48}$

- 少ないデータセットに対してもoverfittingしにくい

Table 4. Top-1 test accuracy (%) on small training datasets (Cars, ResNet-50 pre-trained with ImageNet classification).

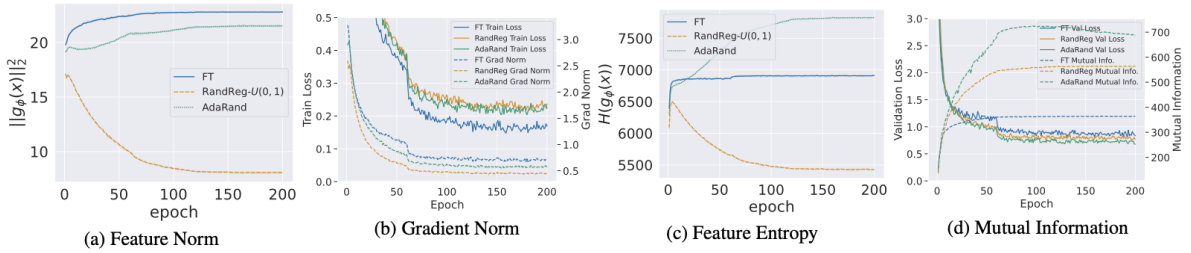
Method / Dataset Size (%)	10 %	25 %	50%
Fine-tuning	$19.58^{\pm.07}$	$53.10^{\pm.08}$	$77.86^{\pm.09}$
FNP [10]	$23.68^{\pm.34}$	$57.07^{\pm.34}$	$79.93^{\pm.21}$
DR-Tune [38]	$23.68^{\pm.34}$	$57.07^{\pm.34}$	$79.93^{\pm.21}$
RandReg-$U(0,1)$	$25.35^{\pm.14}$	$58.33^{\pm.45}$	$80.91^{\pm.51}$
RandReg-$\mathcal{N}(0,1)$	$26.15^{\pm.21}$	$59.94^{\pm.58}$	$81.74^{\pm.41}$
RandReg-$\mathcal{N}(\mu_0,\sigma_0^2)$	$24.95^{\pm.05}$	$59.66^{\pm.24}$	$81.31^{\pm.17}$
RandReg-CP	$26.45^{\pm.26}$	$59.37^{\pm.46}$	$81.13^{\pm.23}$
AdaRand (Ours)	$\mathbf{27.55}^{\pm.10}$	$\mathbf{61.09}^{\pm.54}$	$\mathbf{82.19}^{\pm.19}$

- 他手法と比較しても、計算時間をかけすぎずメモリを使い過ぎず、ある程度の計算コストに留めている

Method	Test Accuracy (%)	Time / Epoch (sec.)	GPU Mem. (MiB)
Fine-tuning	$88.29^{\pm.12}$	**9.737**	**8,073**
FNP [10]	$90.27^{\pm.06}$	9.916	8,073
L2SP [18]	$88.50^{\pm.25}$	12.405	8,153
DELTA [19]	$89.00^{\pm.24}$	14.733	9,211
BSS [5]	$89.70^{\pm.06}$	11.403	8,227
Core-tuning [36]	$90.01^{\pm.13}$	16.872	24,223
DR-Tune [38]	$90.38^{\pm.59}$	25.475	8,845
Co-Tuning [33]	$90.66^{\pm.34}$	11.546	8,125
UOT [21]	$90.82^{\pm.19}$	12.405	14,293
RandReg-$U(0,1)$	$90.42^{\pm.24}$	9.988	8,075
RandReg-$\mathcal{N}(0,1)$	$90.32^{\pm.41}$	9.971	8,075
RandReg-$\mathcal{N}(\mu_s, \sigma_s^2)$	$90.61^{\pm.24}$	9.963	8,075
RandReg-CP	$90.55^{\pm.17}$	11.552	8,075
AdaRand w/o ℓ_{intra}	$90.81^{\pm.21}$	12.410	8,585
AdaRand w/o ℓ_{inter}	$90.99^{\pm.06}$	12.841	8,585
AdaRand (Ours)	$\mathbf{91.17}^{\pm.13}$	13.824	8,585

RandRegとAdaRandの比較

- 特徴ベクトルや勾配のノルムが小さくなりすぎていない
- entropyや相互情報量を最大化できている



(a) Feature Norm (b) Gradient Norm (c) Feature Entropy (d) Mutual Information

- 特徴ベクトルをPCAした結果の分布は、AdaRandではクラスごとのまとまった混合ガウス分布となっていて分類タスクで有効な特徴ベクトルの分布になっている

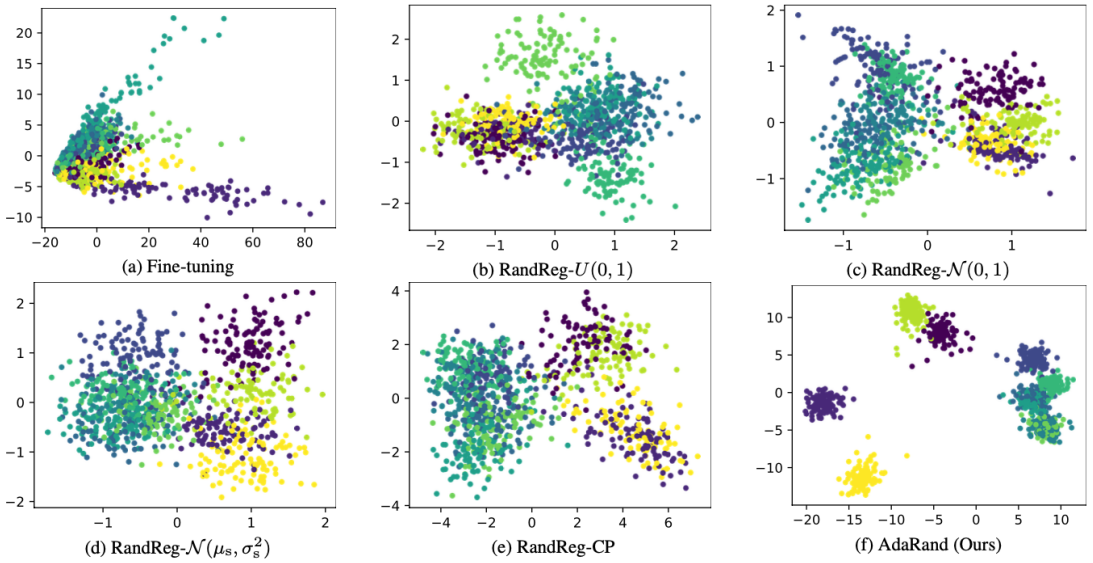


Figure 3. PCA visualization of feature spaces of trained models (CIFAR-10, ResNet-50). The colors in the sample plot correspond to that class. AdaRand clearly forms well-separated clusters, which can be useful for solving downstream classification tasks.

参考文献

Yamaguchi, S. Y., Kanai, S., Adachi, K., & Chijiwa, D. (2024). Adaptive Random Feature Regularization on Fine-tuning Deep Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 23481-23490)

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In International Conference on Machine Learning, 2018.

Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, Zeyu Chen, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. In International Conference on Learning Representations, 2019.

Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In Advances in Neural Information Processing Systems, 2019.

Yang Zhong and Atsuto Maki. Regularizing cnn transfer learning with randomised regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13637–13646, 2020.

Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. Advances in Neural Information Processing Systems, 2021.

Nan Zhou, Jiaxin Chen, and Di Huang. Dr-tune: Improving fine-tuning of pretrained visual models by distribution regularization with semantic calibration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.

Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. Advances in Neural Information Processing Systems, 2020.

Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Xiangyang Ji, Antoni B. Chan, and Rong Jin. Improved fine-tuning by better leveraging pre-training data. In Advances in Neural Information Processing Systems, 2022.

Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1086–1095, 2017.

Lev Faivishevsky and Jacob Goldberger. Ica based on a smooth estimation of the differential entropy. In Advances in neural information processing systems, 2008.