Learning Compact Features via In-Training Representation Alignment

Xin Li, Xiangrui Li, Deng Pan, Yao Qiang, Dongxiao Zhu

Department of Computer Science, Wayne State University
Detroit, Michigan 48202, USA
{xinlee, xiangruili, pan.deng, yao, dzhu}@wayne.edu

Abstract

Deep neural networks (DNNs) for supervised learning can be viewed as a pipeline of the feature extractor (i.e., last hidden layer) and a linear classifier (i.e., output layer) that are trained jointly with stochastic gradient descent (SGD) on the loss function (e.g., cross-entropy). In each epoch, the true gradient of the loss function is estimated using a mini-batch sampled from the training set and model parameters are then updated with the mini-batch gradients. Although the latter provides an unbiased estimation of the former, they are subject to substantial variances derived from the size and number of sampled mini-batches, leading to noisy and jumpy updates. To stabilize such undesirable variance in estimating the true gradients, we propose In-Training Representation Alignment (ITRA) that explicitly aligns feature distributions of two different mini-batches with a matching loss in the SGD training process. We also provide a rigorous analysis of the desirable effects of the matching loss on feature representation learning: (1) extracting compact feature representation; (2) reducing over-adaption on mini-batches via an adaptive weighting mechanism; and (3) accommodating to multi-modalities. Finally, we conduct large-scale experiments on both image and text classifications to demonstrate its superior performance to the strong baselines.

Introduction

Recently, deep neural networks (DNNs) have achieved remarkable performance improvements in a wide range of challenging tasks in computer vision (He et al. 2016; Huang et al. 2019; Pan, Li, and Zhu 2021; Qiang et al. 2022a), natural language processing (Sutskever, Vinyals, and Le 2014; Chorowski et al. 2015; Qiang et al. 2022b) and healthcare informatics (Miotto et al. 2018; Li, Zhu, and Levy 2020). For supervised learning, DNNs can be viewed as a feature extractor followed by a linear classifier on the latent feature space, which are jointly trained using stochastic gradient descent (SGD). Specifically, in each iteration of SGD, a mini-batch of m samples $\{(x_i, y_i)\}_{i=1}^m$ is sampled from the training data $\{(x_i, y_i)\}_{i=1}^n (n > m)$. The gradient of loss function $L(x, \theta)$ is calculated on the mini-batch, and network parameter θ is updated via one step of gradient descent (learning rate α):

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(x_i, \theta) \approx \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta L(x_i, \theta), \\ &\theta \leftarrow \theta - \alpha \cdot \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta L(x_i, \theta). \end{aligned} \tag{1}$$

This update in Eq.(1) can be interpreted from two perspectives. First, from the conventional approximation perspective, the true gradient of the loss function (i.e., gradient on the entire training data) is approximated by the mini-batch gradient. As each mini-batch gradients are unbiased estimators of the true gradient of the loss function and the computation is inexpensive, large DNNs can be efficiently and effectively trained with modern computing infrastructures. Second, Eq. (1) can also be interpreted as an exact gradient descent update on the mini-batch. In other words, SGD updates network parameters θ to achieve maximum improvement in fitting the mini-batch. As each mini-batch is often uniformly sampled from each class of the training data, such exact update inevitably introduces the undesirable variance in gradients calculation via backpropagation, resulting in the over-adaption of model parameters to that mini-batch.

A natural question then to ask is, *"can we reduce the over-adaption to mini-batches?"*, to reduce the mini-batch dependence on SGD update in Eq. (1). In this paper, we propose In-Training Representation Alignment (ITRA) that aims at reducing the mini-batch over-adaption by aligning feature representation of different mini-batches that is learned by the feature extractor in SGD. Our motivation for feature alignment is: *if the SGD update using one mini-batch A is helpful for DNNs learning good feature representations with respect to the entire data, then for another mini-batch B, their feature representation should align well with each other*. In this way, we can reduce mini-batch over-adaption by forcing accommodation of SGD update to B and reducing dependence of the parameter update on A. Ideally, if the distribution $P(h)$ of latent feature h is known as a prior, we could explicitly match the mini-batch feature h_{mb} with $P(h)$ via maximum likelihood. However, in practice, $P(h)$ is not known or does not even have an analytic form. To achieve this, we utilize the maximum mean discrepancy (MMD) (Gretton et al. 2012) from statistical hypothesis testing for the two-sample problem. MMD is differentiable that can be trained via back prop-

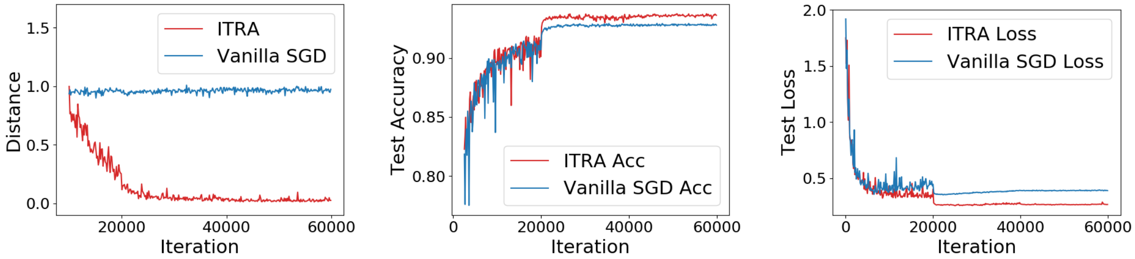


Figure 1: A comparison of ITRA and vanilla SGD training on the CIFAR10 testing data. Left: normalized distance between samples of the same class from different mini-batches used in training; middle: testing accuracy; right: testing cross-entropy loss. The model is Resnet18.

agation. Moreover, we show in an analysis that the gradient of MMD enjoys several good theoretical merits. Based on the analysis, ITRA reduces SGD update adaption to the mini-batch by implicitly strengthening the supervision signal of high-density samples via an adaptive weighting mechanism (see details in Section 4), where high-density samples are closely clustered to form modalities for each class.

To check effect of gradient update on feature representation learning, an illustrative example is presented in Figure 1. The model is Resnet18 with BN layers trained with cross-entropy (CE) loss. We calculate the distance between a pair of same-class samples from two mini-batches respectively and plot the normalized distance in the left panel of Figure 1, after model training stabilizes and achieves relatively good performance. We see that when model is trained only with CE loss in vanilla SGD, the distance stabilizes while the training makes progresses. This is due to that after the model capturing the classification pattern for each class, vanilla SGD adapts to mini-batch samples to achieve gain for the loss function yet does not further encourage feature alignment to learn compact feature representations. Hence, vanilla SGD has little effect on the compactness of feature representations. However, in ITRA, the distance between a pair of samples keeps decreasing. This implies ITRA indeed helps DNN to learn more compact feature representations by aligning different mini-batches and thus achieves higher accuracy and lower loss (Figure 1 middle and right panels).

We summarize our original contributions as follows. (1) We propose a novel and general strategy ITRA for training DNNs. ITRA augments conventional SGD with regularization by forcing feature alignment of different mini-batches to reduce variance in estimating the true gradients using mini-batches. ITRA can enhance the existing regularization approaches and is compatible with a broad range of neural network architectures and loss functions. (2) We provide theoretical analysis on the desirable effects of ITRA and explains why ITRA helps reducing the over-adaption of vanilla SGD to the mini-batch. With MMD, ITRA has an adaptive weighting mechanism that can help neural networks learn more discriminative feature representations and avoid the assumption of uni-modality on data distribution. Results on benchmark datasets demonstrate that training with ITRA can significantly improve DNN performance, compared with other state-of-the-art methods.

Related Work

Modern architectures of DNNs usually have an extremely large number of model parameters, which often outnumbers the available training data. To reduce overfitting in training DNNs, regularizations are needed. Those regularization methods include classic ones such as L_1/L_2-norm penalties and early stopping (Li, Zhu, and Dong 2018; Li and Zhu 2018). For deep learning, additional useful approaches are proposed motivated by the SGD training dynamics (Li et al. 2022). For example, dropout (Srivastava et al. 2014) and its variants (Gao, Pei, and Huang 2019; Ghiasi, Lin, and Le 2018) achieve regularization by reducing the co-adaption of hidden neurons of DNNs. (Ioffe and Szegedy 2015) proposes batch normalization (BN) to reduce the internal covariate shift caused by SGD. For image classification, data-augmentation types of regularization are also developed (DeVries and Taylor 2017; Gastaldi 2017; Li et al. 2020, 2021). Different from those approaches, our proposed ITRA is motivated by the perspective of exact gradient update for each mini-batch in SDG training, and achieves regularization by encouraging the alignment of feature representations of different mini-batches. Those methods are compatible with ITRA for training DNNs and hence can be applied in conjunction with ITRA.

Another line of regularization are loss function based that the supervision loss is augmented with other penalties under different considerations. One example is label smoothing (Szegedy et al. 2016), which corrupts the true label with a uniformly-distributed noise to discourage DNNs' over-confident predictions for training data. *The work that is closest to ours is Center loss* (Wen et al. 2016), which reduces the intra-class variation by aligning feature of each class to its "center". With the assumption of distribution uni-modality for each class, it explicitly encourages the feature representations clustering around its center. However, this assumption may be too strict since true data distribution is generally unknown and can be multi-model. On the contrary, ITRA reduces variances and encourages intra-class compactness by aligning a pair of features from two minibatches to each other, which avoids the distribution assumption and is accommodating to multi-modalities.

To match the distribution of features learned from different mini-batches, ITRA uses MMD as its learning objective. MMD (Gretton et al. 2007, 2012) is a probability metric for testing whether two finite sets of samples are generated from the same distribution. Using a universal kernel (i.e., Gaussian kernel), minimizing MMD encourages to match all moments of the empirical data distribution. MMD has been widely applied in many machine learning tasks. For example, (Li, Swersky, and Zemel 2015) and (Li et al. 2017) use MMD to train unsupervised generative models by matching the generated distribution with the data distribution. Another application of MMD is for the domain adaption. To learn domain-invariant feature representations, (Long et al. 2015) uses MMD to explicitly match feature representations from different domains. There are also other probability-based distance metrics applied in domain adaption such as $\mathcal{A}$-divergence (Ben-David et al. 2007) and Wasserstein distance (Shen et al. 2018). However, these metrics are *non-differentiable* while the differentiability of MMD enables the adaptive weighting mechanism in ITRA. Moreover, our goal is different from those applications. In ITRA, we do not seek exact distribution matching. Instead, we use class-conditional MMD as a regularization to improve SGD training.

Preliminary: Maximum Mean Discrepancy

Given two finite sets of samples $S_1 = \{x_i\}_{i=1}^n$ and $S_2 = \{y_i\}_{i=1}^m$, MMD (Gretton et al. 2007, 2012) is constructed to test whether S_1 and S_2 are generated from the same distribution. MMD compares the sample statistics between S_1 and S_2, and if the discrepancy is small, S_1 and S_2 are then likely to follow the same distribution.

Using the kernel trick, the empirical estimate of MMD (Gretton et al. 2007) w.r.t. S_1 and S_2 can be rewritten as:

$$\begin{aligned}\text{MMD}(S_1, S_2) =& \Big[\frac{1}{n^2}\sum_{i,j=1}^{n}\mathcal{K}(x_i, x_j) + \frac{1}{m^2}\sum_{i,j=1}^{m}\mathcal{K}(y_i, y_j) \\ &- \frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathcal{K}(x_i, y_j)\Big]^{1/2},\end{aligned}$$

where $\mathcal{K}(\cdot,\cdot)$ is a kernel function. (Gretton et al. 2007) shows that if $\mathcal{K}$ is a characteristic kernel, then asymptotically MMD = 0 if and only S_1 and S_2 are generated from the same distribution. A typical choice of $\mathcal{K}$ is the Gaussian kernel with bandwidth parameter σ: $\mathcal{K}(x, y) = \exp(-\frac{||x-y||^2}{\sigma})$. With Gaussian kernel, minimizing MMD is equivalent to matching all orders of moments of the two datasets (Li, Swersky, and Zemel 2015).

In-Training Representation Alignment

The Proposed ITRA The idea of ITRA is to reduce the DNN over-adaption to a mini-batch if we view the SGD iteration as an exact update for that mini-batch. In terms of feature learning, we attempt to train the feature extractor to encode less mini-batch dependence into the feature representation. From the distribution point of view, the latent feature distribution of the mini-batch should approximately match with, or more loosely, should not deviate much from that of the entire data. However, aligning a mini-batch with the global statistics from entire data may not be available, we sample a pair of mini-batch to match each other to reduce the variance. It is possible to sample more mini-batches to further reduce variances but is computationally expensive.

More formally, let $f_\theta(x)$ be a convolutional neural network model for classification that is parameterized by θ. It consists of a feature extractor $h = E_{\theta_e}(x)$ and a linear classifier $C_{\theta_c}(h)$ parameterized by θ_e and θ_c respectively. Namely, $f_\theta(x) = C_{\theta_c}(E_{\theta_e}(x))$ and $\theta = \{\theta_e, \theta_c\}$. Without ambiguity, we drop θ in f, E and C for notational simplicity. In each iteration, let $S_{(1)} = \{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^{m_1}$ be the mini-batch of m_1 samples. Then the loss function using cross-entropy (CE) on $S_{(1)}$ can be written as

$$L_{mb}(\theta) = -\frac{1}{m_1}\sum_{i=1}^{m_1}\log f_{y_i^{(1)}}(x_i^{(1)}), \tag{2}$$

where $f_{y_i^{(1)}}(x_i^{(1)})$ is the predicted probability for $x_i^{(1)}$'s true label $y_i^{(1)}$. SGD performs one gradient descent step on L_{mb} w.r.t. θ using Eq. (1). To reduce θ's dependence on S_1 in this exact gradient descent update, we sample from the training data another mini-batch $S_{(2)} = \{(x_i^{(2)}, y_i^{(2)})\}_{i=1}^{m_2}$ to match the latent feature distribution between $S_{(1)}$ and $S_{(2)}$ using MMD:

$$\begin{aligned}&H_{(1)} = \{h_i^{(1)} = E(x_i^{(1)}) : i = 1, \cdots, m_1\}, \\ &H_{(2)} = \{h_i^{(2)} = E(x_i^{(2)}) : i = 1, \cdots, m_2\}, \\ &\text{Match}(\theta_e; H_{(1)}, H_{(2)}) = \text{MMD}(H_{(1)}, H_{(2)}).\end{aligned} \tag{3}$$

Our proposed ITRA modifies the conventional gradient descent step in SGD by augmenting the CE loss (Eq. (2)) with the matching loss, which justifies the name of ITRA:

$$\theta \leftarrow \theta - \alpha\nabla_\theta\big[L_{mb}(\theta) + \lambda\text{Match}(\theta_e; H_{(1)}, H_{(2)})\big], \tag{4}$$

where λ is the tuning parameter controlling the contribution of the matching loss. Note that mini-batch $S_{(2)}$ is not used in the calculation of cross-entropy loss $L_{mb}(\theta)$.

Class-conditional ITRA For classification tasks, we could also utilize the label information and further refine the match loss as a sum of class-conditional matching loss, termed as **ITRA-c** ($k = 1, \cdots, K$):

$$\begin{aligned}&H_{(1)}^k = \{h_i^{(1)} = E(x_i^{(1)}) : y_i = k, i = 1, \cdots, m_1\} \\ &H_{(2)}^k = \{h_i^{(2)} = E(x_i^{(2)}) : y_i = k, i = 1, \cdots, m_2\} \\ &\text{Match}_c(\theta_e; H_{(1)}, H_{(2)}) = \frac{1}{K}\sum_{k=1}^{K}\text{MMD}(H_{(1)}^k, H_{(2)}^k),\end{aligned} \tag{5}$$

where K is the total number of classes and $y_i = k$ the true label of sample x_i. The ITRA-c update is

$$\theta \leftarrow \theta - \alpha\nabla_\theta\big[L_{mb}(\theta) + \lambda\text{Match}_c(\theta_e; H_{(1)}, H_{(2)})\big]. \tag{6}$$

Analysis on ITRA

On learning compact feature representations To further gain insight on the desirable effects of ITRA on the SGD training procedure, we analyze the matching loss at the sample level. With the same notation in Eq. (5), the matching loss for class k is

$$M := \text{Match}_k = \text{MMD}(H_{(1)}^k, H_{(2)}^k).$$

Since MMD is symmetric with respect to $H^k_{(1)}$ and $H^k_{(2)}$, without loss of generality, we consider sample $x^{(1)}_i$ with its feature representation $h^{(1)}_i = E(x^{(1)}_i)$ from $H^k_{(1)}$ (but the CE loss is not symmetric and only calculated on the first mini-batch $H_{(1)}$). Then the gradient of matching loss with respect to $h^{(1)}_i$ is (superscript (1) in $x^{(1)}_i$ and $h^{(1)}_i$ are dropped.)

$$\begin{aligned}\nabla_{h_i} M =& \frac{1}{\sqrt{M}} \nabla_{h_i} \big[\frac{1}{m_1^2} \sum_{j=1}^{m_1} \mathcal{K}(h_i, h_j^{(1)}) \\ &- \frac{2}{m_1 m_2}\sum_{j=1}^{m_2}\mathcal{K}(h_i, h_j^{(2)})\big].\end{aligned}$$

For Gaussian kernel $\mathcal{K}(x, y)$, its gradient with respect to x is $\nabla_x \mathcal{K}(x, y) = -2\exp(-\frac{||x-y||^2}{\sigma})\frac{x-y}{\sigma}$. Note that σ is data-dependent and treated as hyperparameter. Hence, it is not back propagated in the training process and in practice set as the median of sample pairwise distances (Gretton et al. 2012; Long et al. 2015; Li, Swersky, and Zemel 2015). By the linearity of gradient operator, we have

$$\begin{aligned}\nabla_{h_i} M =& -\frac{2}{\sqrt{M}}\big[\frac{1}{m_1^2}\sum_{j=1}^{m_1}\exp(-\frac{||h_i - h_j^{(1)}||^2}{\sigma})\frac{h_i - h_j^{(1)}}{\sigma} \\ &- \frac{2}{m_1 m_2}\sum_{j=1}^{m_2}\exp(-\frac{||h_i - h_j^{(2)}||^2}{\sigma})\frac{h_i - h_j^{(2)}}{\sigma}\big].\end{aligned} \tag{7}$$

We notice that for function $g_a(x) = \exp(-x^2/a)x/a$ (a is some constant), $g_a(x) \to 0$ exponentially as $x \to \infty$. Hence, for fixed σ, using the triangle inequality of L_2 norm,

$$\begin{aligned}||\nabla_{h_i} M|| \leq & \frac{2}{\sqrt{M}}\big[\frac{1}{m_1^2}\sum_{j=1}^{m_1} g_\sigma(||h_i - h_j^{(1)}||) \\ &+ \frac{2}{m_1 m_2}\sum_{j=1}^{m_2} g_\sigma(||h_i - h_j^{(2)}||)\big].\end{aligned} \tag{8}$$

Within the mini-batch, $\sqrt{M}$ remain as constant for all samples. From Eq. (8), we observe that when x_i deviates significantly away from the majority of samples of the same class, i.e., noisy samples or outliers, $||h_i - h_j^{(1)}||$ and $||h_i - h_j^{(2)}||$ are large, the magnitude of its gradient in matching loss diminishes. In other words, x_i will only provide signal from the supervision loss (e.g., CE loss) and its impact on matching loss is negligible. On the other hand, training ITRA with matching loss promotes the alignment of feature representations of samples that stay close in the latent feature space. From the data distribution perspective, samples deviating from the majority are likely of low-density or even outliers. Then such behavior of ITRA implies that it can help DNNs to better capture information from high density areas and reduce the distraction of "low density" samples in learning feature representations on the data manifold.

On reducing over-adaption to mini-batches The analysis above shows that low-density samples only provide supervision signal in ITRA, we now analyze how ITRA reduces the over-adaption to mini-batches. It turns out that this effect is achieved by an adaptively weighted feature alignment mechanism, which *implicitly* boosts the supervision signal from high-density samples and resultantly downweights relatively the contribution of low-density samples.

To understand this, we examine the full gradient of supervision loss L and matching loss MMD. Note that in ITRA, the gradient of supervision loss is only calculated on one mini-batch. Without loss of generality, we consider sample x_i from the first mini-batch. The full gradient of $L(x_i)$ and M=MMD$(x_i, H^k_{(2)})$ with respect to h_i is (using the same notation as above)

$$\begin{aligned}\nabla_{h_i}(M + L) =& \frac{4}{\sqrt{M} m_2}\sum_{j=1}^{m_2}\exp(-\frac{||h_i - h_j^{(2)}||^2}{\sigma})\frac{h_i - h_j^{(2)}}{\sigma} \\ &+ \nabla_{o_i} L \cdot \frac{\partial o_i}{\partial h_i},\end{aligned}$$

where o_i is the output for x_i. Let $A = \sum_{j=1}^{m_2}\exp(-||h_i - h_j^{(2)}||^2/\sigma)$ and $w_j = \exp(-||h_i - h_j^{(2)}||^2/\sigma)/A$ ($\sum_{j=1}^{m_2} w_j = 1$), then equivalently:

$$\nabla_{h_i}(M + L) = \frac{4A}{\sqrt{M} m_2 \sigma}(h_i - \sum_{j=1}^{m_2} w_j h_j^{(2)}) + \nabla_{o_i} L \cdot \frac{\partial o_i}{\partial h_i}. \tag{9}$$

When ITRA converges and DNNs is well trained with good performance, $||\nabla_{h_i}(M + L)|| \approx 0$ and $||\nabla_{o_i} L \cdot \partial o_i / \partial h_i||$ is close to zero, we have $||h_i - \sum_{j=1}^{m_2} w_j h_j^{(2)}|| < \epsilon$ (ϵ is a small scalar). In other words, ITRA promotes the feature representation h_i of x_i to align with the weighted average $\sum_{j=1}^{m_2} w_j h_j^{(2)}$ ($\sum_{j=1}^{m_2} w_j = 1$), where each w_j is adaptively adjusted in the training process based on similarity between h_i and $h_j^{(2)}$ in the latent feature space. As mini-batch samples are uniformly sampled from the training data, it is expected that on average, the majority of $\{h_j^{(2)}\}_{j=1}^{m_2}$ are from high-density area of the data distribution. For DNNs with good generalizability, DNNs must perform well for samples from those areas (as testing samples are more likely to be generated from high-density areas in the data manifold). Hence, provided that sample x_i is of high-density that already provides useful supervision signal, ITRA further boosts its contribution by aligning h_i with $\sum_{j=1}^{m_2} w_j h_j^{(2)}$ of other high-density samples in the 2nd mini-batch. *The adaptive weight w_j is critical*: if sample $h_j^{(2)}$ is of low-density and deviates far from x_i, its weight w_j is automatically adjusted small, having vanishing contribution in the gradient. This in turn downweights relatively the contribution of low-density samples in SGD, resulting in the reduction of over-adaption to mini-batches.

Accommodating multi-modalities The adaptively weighting mechanism brings another benefit: if the data distribution (for each class) is multi-modality in the latent feature space, ITRA automatically aligns x_i with its corresponding modality. Specifically, without loss of generality, assume two modalities md_1 and md_2, $\{h_j^{(2)}\}$ consists of samples from md_1 and md_2 and x_i is generated from md_1. We can rewrite $h_i -$

$\sum_{j=1}^{m_2} w_j h_j^{(2)} = h_i - (\sum_{j \in md_1} w_j h_j^{(2)} + \sum_{j \in md_2} w_j h_j^{(2)})$. As x_i is generated from md_1 and deviates from md_2, implying that x_i is closer to samples from the same modality than those from the other modality. Hence, with the adaptively weighting mechanism in Eq. (9), $w_j \approx 0$ $(j \in md_2)$ and $h_i - \sum_{j=1}^{m_2} w_j h_j^{(2)} \approx h_i - \sum_{j \in md_1} w_j h_j^{(2)}$. That is, align x_i only with samples from the same modality. Therefore, ITRA avoids the uni-modality assumption on data distribution as in (Wen et al. 2016; Wan et al. 2018) and justifies the advantage of nonparametric MMD for feature alignment.

Experiments

In this Section, we extensively evaluate the ITRA performance using benchmark datasets on both image classification (i.e., KMNIST (Clanuwat et al. 2018), FMNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR10, CIFAR100 (Krizhevsky, Hinton et al. 2009), STL10 (Coates, Ng, and Lee 2011) and ImageNet (Deng et al. 2009)) and text classification (i.e., AG's News, Amazon Reviews, Yahoo Answers and Yelp Reviews) tasks. In our experiments, class-conditional ITRA-c is tested as it exploits implicit label information with better supervision in the training process. In addition to using vanilla SGD training as the baseline (i.e., w/o ITRA), we also compare ITRA with more widely used *loss-function based regularization methods* as the strong baselines for comparison: label smoothing (LSR) (Szegedy et al. 2016) and center loss (Center) (Wen et al. 2016). For evaluation metrics, we report the Top-1 accuracy and CE loss value for all methods. The optimal hyperparameter value λ for each method is also reported. Results on other tuning parameter values as well as experimental details are provided in supplementary materials.

Image classification Table 1 shows the performance for KMNIST and FMNIST testing data. From the Table, we see that training with ITRA achieves better results in terms of higher accuracy and lower CE. In terms of the testing loss, ITRA has a smaller loss value compared with other methods. The testing loss with respect to different λ values are shown in Supplementary Materials. As CE is equivalent to negative log-likelihood, smaller CE value implies that the network makes predictions on testing data with higher confidence on average. In each iteration of ITRA, there is a trade-off between the CE and matching loss. This leads to that ITRA has a regularization effect by alleviating the over-confident predictions on training data. As a result, the smaller gap between training and testing losses implies that ITRA has better generalization performance. When trained with vanilla SGD, we observe that the increasing testing loss exhibits an indication of overfitting, which is due to that FMNIST has a significant number of hard samples (e.g., those from pullover, coat and shirt classes). However, ITRA is capable of regularizing the training process hence prevents overfitting and stabilizes the testing loss as shown in the Figure 2.

Additionally, in Table 2, we present the performance of Resnet18, VGG13 and MobilenetV2 on CIFAR10, STL10 and CIFAR100. From the Table, we see that ITRA achieves the best performance compared among all the four methods. Especially for the relatively more challenging (lower accuracy) STL10 data set, ITRA outperforms the baseline with a significant margin, i.e., Resnet 1.9%, VGG13 1.4% and MobilenetV2 2.9%. In terms of CE loss, all methods have similar training losses that are close to zero. However, ITRA and Center have significantly better testing loss than other the baseline and LSR. A closer gap between training and testing losses indicates a better generalization of the DNN models enabled by regularization capability of ITRA.

Table 1: Accuracy (in %, larger is better) and CE (smaller is better) on KMNIST and FMNIST data. The optimal performances are obtained by tuning multiple λs according to existing literature (Szegedy et al. 2016; Wen et al. 2016).

	KMNIST			FMNIST		
	λ	Acc ↑	CE ↓	λ	Acc ↑	CE ↓
Baseline	-	95.57	0.183	-	92.43	0.294
LSR	0.1	95.60	0.181	0.1	92.47	0.292
Center	0.1	94.90	0.214	0.1	92.10	0.263
ITRA	0.8	**95.79**	**0.170**	0.6	**92.57**	**0.224**

Larger-scale experiment on image classification Table 3 shows results on the large-scale ImageNet dataset and the deeper Resnet-101 network architecture. Note that compared with performance of Resnet-18 in Table 2, the deeper Resnet-101 indeed demonstrates a better performance over the CIFAR100 dataset. For both larger dataset and deeper network, ITRA consistently achieves better accuracy and lower CE value than other methods. Markedly, for the ImageNet dataset, ITRA improves the accuracy by $5.0\%(3.12/62.92)$ and CE value by $9.7\%(0.15/1.55)$ over the standard baseline. When compared with the strong baseline Center loss, ITRA also improves the accuracy by $2.1\%(1.36/64.68)$ and CE value by $4.8\%(0.07/1.47)$. The training time of ResNet-101 (CIFAR-100) and ResNet-50 (ImageNet) using ITRA are 7.5% and 3.9% more than baseline on an RTX 3090 GPU. Despite the moderate increase in training time for large-scale experiments due to the extra computation incurred by sampling additional mini-batch, it demonstrates a reasonable trade-off between the increase in computational cost and gaining attractive analytic properties of ITRA.

Large-scale experiment on text classification ITRA performance is also evaluated on large-scale text classification experiments. We use different loss functions for fine-tuning the pre-trained Bert-base, DistillBert and XLNet models from Huggingface transformers library (Wolf et al. 2019). Table 4 shows that the models fine-tuned with ITRA achieve a better performance in terms of accuracy and CE value on most datasets. Specifically, for Bert-base, DistillBert, XLNet models, ITRA achieves an average accuracy improvement of 1.3%, 0.3%, 0.5%, respectively, and an average CE value improvement of 22.4%, 16.1%, and 15.6%, respectively. It is worth noting that the Center loss also reduces CE value occasionally in the experiments, its accuracy performance is nevertheless compromised. The potential reason behind this phenomenon could be the multi-modality of natural language. The Center loss's uni-modality assumption helps model to minimize the distances within the class (hence CE), but can therefore lead to sub-optimal feature learning for hard samples near the boundary between modals if the class condi-

Table 2: Accuracy and CE loss on CIFAR10, STL10 and CIFAR100 datasets.

		CIFAR10			STL10			CIFAR100		
		λ	Acc ↑	CE ↓	λ	Acc ↑	CE ↓	λ	Acc ↑	CE ↓
Resnet18	Baseline	-	92.99	0.40	-	70.88	1.63	-	74.19	1.05
	LSR	0.1	92.73	0.42	0.1	71.08	1.55	0.1	74.21	1.04
	Center	0.1	92.30	0.35	0.1	70.97	1.10	0.05	73.98	0.98
	ITRA	0.8	**93.70**	**0.27**	0.6	**72.78**	**1.05**	0.6	**74.88**	**0.97**
VGG13	Baseline	-	92.49	0.47	-	74.40	1.55	-	71.72	1.46
	LSR	0.1	92.53	0.46	0.1	74.50	1.51	0.1	71.75	1.43
	Center	0.05	92.11	0.38	0.05	74.04	1.16	0.05	71.65	1.31
	ITRA	0.8	**92.72**	**0.33**	0.8	**75.80**	**0.93**	0.6	**72.55**	**1.22**
MobileV2	Baseline	-	88.55	0.62	-	59.09	2.14	-	66.42	1.57
	LSR	0.1	88.77	0.61	0.1	59.01	2.12	0.1	66.60	1.55
	Center	0.1	88.81	0.53	0.1	58.24	**1.46**	0.05	66.39	1.51
	ITRA	1.0	**89.37**	**0.43**	0.6	**62.02**	1.60	0.6	**67.23**	**1.49**

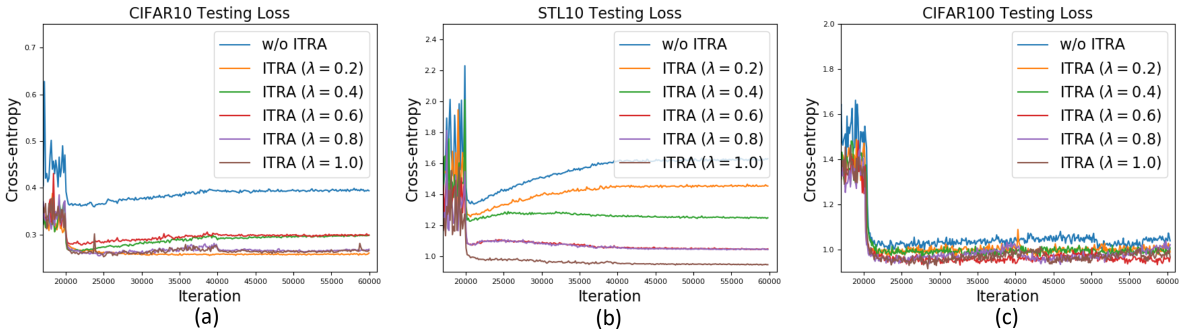


Figure 2: Testing loss of Resnet18 w.r.t. different λ values on CIFAR10, STL10 and CIFAR100.

Table 3: Accuracy and CE loss of Resnet-101 on CIFAR-100 and Resnet-18 on ImageNet.

	CIFAR-100 (Resnet-101)			ImageNet (Resnet-18)		
	Batch Size	Acc	CE	Batch Size	Acc	CE
Baseline	200	75.85	1.04	256	50.01	2.24
Center ($\lambda = 0.05$)	200	75.23	1.05	256	51.83	2.21
ITRA ($\lambda = 0.6$)	50	75.97	1.01	64	51.32	2.19
	100	**77.35**	1.01	128	51.69	2.17
	200	76.85	**0.94**	256	**53.13**	**2.07**

tional distribution is indeed multi-modal.

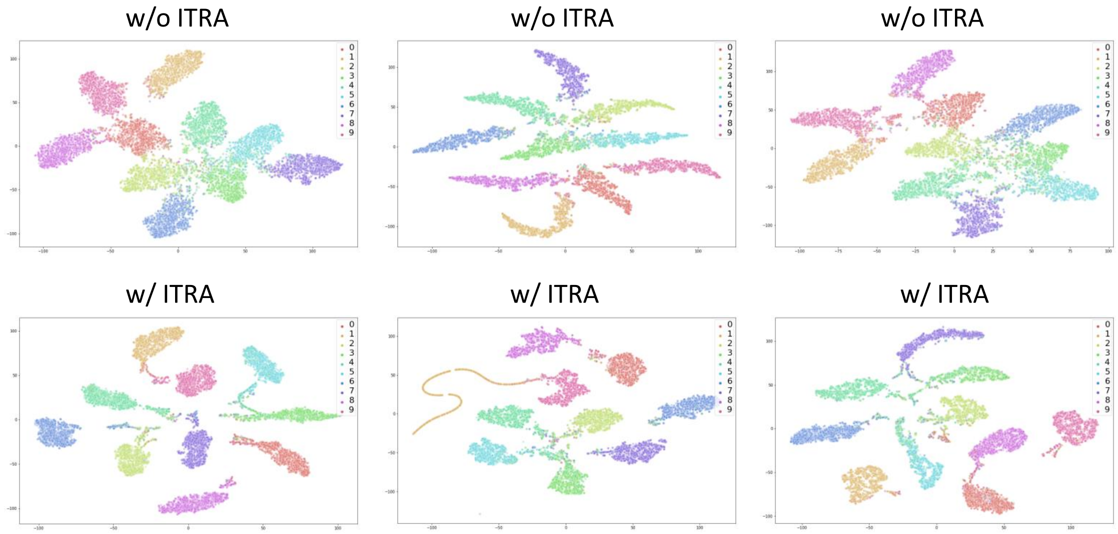
Comparing with Center loss As discussed in Section 2, center loss (Wen et al. 2016) is the closest work to ours. It effectively characterizes the intra-class variations by aligning features of each to its "center" which is designed to reduce variance in feature learning and results in compact feature representations. Different from ITRA which aligns a pair of features from two minibatches to each other, Center loss explicitly assumes uni-modality of data distribution at feature level for each class, which may be valid in face recognition task where the Center loss is initially proposed for, but can be too stringent in classification task as class-conditional density can be multi-modal. On the contrary, ITRA is capable of accommodating the multi-modalities supported by a rigorous analysis in Section 4.1. In Figure 3, the model trained with ITRA effectively captures the "typical pattern" of each class at feature level and misses some hard samples to improve generalizability. The results on both image and text classifications concur with our analysis: as shown in the Tables 1, 2 and 4, although Center loss can occasionally reduce the CE value, it is still outperformed by our method due to its strong assumption.

Hyperparameter λ and batch size Here we also investigate the influence of hyperparameter λ and batch size of ITRA. As shown in the Tables 1&2, when λ is set with a relatively large value of 0.8 or 1, ITRA can outperform other methods in terms of both accuracy and testing loss, which is due to that larger λs incorporate stronger implicit supervision information as mini-batches from the same class are matched. We also plot the CE loss for different λs in Figure 2 w.r.t. Resnet. Comparing with baseline, we see that training with ITRA results in significant gain in CE, regardless of network architecture. Looking at Figure 2 (b) in more detail, when trained with the baseline, the testing loss shows an increasing trend as a sign of overfitting while ITRA can alleviate this trend as λ increases. For batch size, Table 3 demonstrates that the increase of batch size indeed helps the reduction of variation in feature learning (lower CE loss), however, it usually requires advanced large-scale GPU clusters. Table 3 illustrates that ITRA achieves a better performance even with $1/4$ batch size compared to the baseline and thus avoid this hardware restriction.

Learning compact feature representations From the geometric perspective, samples from the same class should stay

Table 4: Accuracy and CE loss on text classification task.

		AG's News			Amazon Full			Yahoo Answers			Yelp Full		
		λ	Acc ↑	CE ↓	λ	Acc ↑	CE ↓	λ	Acc ↑	CE ↓	λ	Acc ↑	CE ↓
Bert-base	Baseline	-	92.10	0.42	-	58.53	1.66	-	66.60	1.36	-	61.12	1.52
	Center	0.1	91.70	0.45	0.1	59.38	**1.23**	0.1	66.95	1.25	0.1	60.58	1.17
	ITRA	0.5	**92.45**	**0.29**	0.2	**60.08**	1.29	0.6	**67.47**	**1.25**	0.4	**61.68**	**1.09**
DistillBert	Baseline	-	92.13	0.38	-	58.10	1.38	-	66.50	1.28	-	60.02	1.32
	Center	0.1	91.33	0.47	0.1	57.60	1.21	0.1	66.57	1.26	0.1	59.58	**1.14**
	ITRA	0.2	**92.27**	**0.26**	0.4	**58.30**	**1.20**	0.1	**66.65**	**1.17**	0.1	**60.30**	1.17
XLNet	Baseline	-	91.43	0.40	-	60.65	1.34	-	66.90	1.29	-	62.78	1.21
	Center	0.1	90.95	0.45	0.1	59.88	**1.18**	0.1	66.93	1.28	0.1	62.82	1.10
	ITRA	0.8	**91.85**	**0.27**	0.5	**60.90**	1.23	0.5	**67.05**	**1.15**	1.0	**63.32**	**1.08**



(a)Resnet18 (b)VGG13 (c)MobilenetV2

Figure 3: T-SNE plot for CIFAR10 testing data. Networks are trained with λ that achieves best accuracy in Table 2.

close (i.e., intra-class compactness) and those from different classes are expected to stay far apart (i.e., inter-class separability) in the feature space (so that f_k output by softmax is large). We visualize the distribution of CIFAR10 testing samples with T-SNE (Maaten and Hinton 2008) in Figure 3. From the figure, we have the following observations: (1) ITRA learns feature representation that is much tighter with clearer inter-class margin than that learned by vanilla SGD training. (2) The data distribution in the latent space learned by ITRA exhibits a consistent pattern that for each class, the majority of testing samples are closely clustered to form a data manifold, while a small subset of samples deviate from the majority. This phenomenon concurs with our analysis that the matching loss provides diminishing gradient signals for "low-density" samples while encourages the closeness of "high-density" samples. Hence, ITRA can effectively capture the "typical pattern" of each class but can miss some hard samples that overlap with other classes. This explains why ITRA achieves impressive improvement in CE value but not as much in accuracy. Overall, ITRA still outperforms vanilla SGD training and can be used as a promising training prototype that enjoys theoretical merits as shown in the analysis for matching loss.

Conclusion

In this paper, we propose a new training strategy, ITRA, as a loss function based regularization approach that can be embedded in the standard SGD training procedure. ITRA augments vanilla SGD with a matching loss that uses MMD as the objective function. We show that ITRA enjoys three theoretical merits that can help DNN learn compact feature representations without assuming uni-modality on the feature distribution. Experimental results demonstrate its excellent performance on both image and text classification tasks, as well as its impressive feature learning capacity. We outline two possible directions for future studies. The first is to improve ITRA that can learn hard sample more effectively. The second is the ITRA application in learning from poisoned datasets as ITRA is able to capture the high density areas (i.e., modalities) for each class where poisoned samples deviates far from those areas (e.g., erroneously labeled samples from other classes).

Acknowledgements

This work is supported by the National Science Foundation under grant IIS-2211897.

Ethical Impact

This paper proposes ITRA to improve the performance of feature representation learning for training DNNs. As a general training strategy for supervised classification problems, ITRA can be used as a drop-in replacement in wherever the vanilla SGD is used. Most modern deep learning model training utilizes SGD as the standard optimization algorithm where researchers have already proposed various alternatives and/or enhancements to overcome its intrinsic limitations; the over adaption to mini-batch is more pronounced. The proposed in-training regularization via aligning representations enables learning more compact features thus to improve the generalizability of the predictive models. With our and many other effective feature representation learning approaches, manual feature engineering requiring profound domain knowledge and expertise will eventually phase out. As such, our research has positive impacts to a broad range of machine learning and artificial intelligence domains where domain adaption and generalization become the primary concern. For example, in medical imaging based diagnosis, leveraging ITRA on a smaller labeled and heterogeneous training set is expected to demonstrate a competitive and consistent performance to other medical imaging data sets.

References

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, 137–144.

Chorowski, J. K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, 577–585.

Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; and Ha, D. 2018. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Gao, H.; Pei, J.; and Huang, H. 2019. Demystifying Dropout. In *The 36th International Conference on Machine Learning (ICML 2019)*.

Gastaldi, X. 2017. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*.

Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2018. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, 10727–10737.

Gretton, A.; Borgwardt, K.; Rasch, M.; Schölkopf, B.; and Smola, A. J. 2007. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, 513–520.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar): 723–773.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, G.; Liu, Z.; Pleiss, G.; Van Der Maaten, L.; and Weinberger, K. 2019. Convolutional Networks with Dense Connectivity. *IEEE transactions on pattern analysis and machine intelligence*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, C.; Dong, Z.; Fisher, N.; and Zhu, D. 2022. Coupling User Preference with External Rewards to Enable Driver-centered and Resource-aware EV Charging Recommendation. *arXiv preprint arXiv:2210.12693*.

Li, C.-L.; Chang, W.-C.; Cheng, Y.; Yang, Y.; and Póczos, B. 2017. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, 2203–2213.

Li, X.; Li, X.; Pan, D.; and Zhu, D. 2020. On the learning property of logistic and softmax losses for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4739–4746.

Li, X.; Li, X.; Pan, D.; and Zhu, D. 2021. Improving adversarial robustness via probabilistically compact loss with logit constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8482–8490.

Li, X.; and Zhu, D. 2018. Robust feature selection via l2,-norm in finite mixture of regression. *Pattern Recognition Letters*, 108: 15–22.

Li, X.; Zhu, D.; and Dong, M. 2018. Multinomial classification with class-conditional overlapping sparse feature groups. *Pattern Recognition Letters*, 101: 37–43.

Li, X.; Zhu, D.; and Levy, P. 2020. Predicting Clinical Outcomes with Patient Stratification via Deep Mixture Neural Networks. *AMIA Summits on Translational Science Proceedings*, 2020: 367.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *International Conference on Machine Learning*, 1718–1727.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; and Dudley, J. T. 2018. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6): 1236–1246.

Pan, D.; Li, X.; and Zhu, D. 2021. Explaining Deep Neural Network Models with Adversarial Gradient Integration. In *IJCAI*, 2876–2883.

Qiang, Y.; Li, C.; Brocanelli, M.; and Zhu, D. 2022a. Counterfactual interpolation augmentation (cia): A unified approach to enhance fairness and explainability of dnn. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, 732–739.

Qiang, Y.; Pan, D.; Li, C.; Li, X.; Jang, R.; and Zhu, D. 2022b. AttCAT: Explaining Transformers via Attentive Class Activation Tokens. In *Advances in Neural Information Processing Systems*.

Shen, J.; Qu, Y.; Zhang, W.; and Yu, Y. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Wan, W.; Zhong, Y.; Li, T.; and Chen, J. 2018. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9117–9126.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, 499–515. Springer.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.