Feature Variance Regularization: A Simple Way to Improve the Generalizability of Neural Networks

Ranran Huang,[1,2*] **Hanbo Sun,**[1] **Ji Liu,**[2] **Lu Tian,**[2]
Li Wang,[2] **Yi Shan,**[2] **Yu Wang**[1†]
[1]Tsinghua University, [2]Xilinx
{hrr17, sun-hb17}@mails.tsinghua.edu.cn, {jiliu1, lutian, liwa, yishan}@xilinx.com
yu-wang@tsinghua.edu.cn

Abstract

To improve the generalization ability of neural networks, we propose a novel regularization method that regularizes the empirical risk using a penalty on the empirical variance of the features. Intuitively, our approach introduces confusion into feature extraction and prevents the models from learning features that may relate to specific training samples. According to our theoretical analysis, our method encourages models to generate closer feature distributions for the training set and unobservable true data and minimize the expected risk as well, which allows the model to adapt to new samples better. We provide a thorough empirical justification of our approach, and achieves a greater improvement than other regularization methods. The experimental results show the effectiveness of our method on multiple visual tasks, including classification (CIFAR100, ImageNet, fine-grained datasets) and semantic segmentation (Cityscapes).

Introduction

By virtue of a large number of parameters and multiple nonlinear layers, deep neural networks have powerful abilities to learn the complex relationship between inputs and outputs, and have demonstrated much impressive success on a variety of visual tasks in recent years (Simonyan and Zisserman 2014; He et al. 2016; Sandler et al. 2018; Ioffe and Szegedy 2015; Paszke et al. 2016). To train neural networks, Empirical Risk Minimization (ERM) (Vapnik 1995) is widely adopted as a learning scheme to minimize prediction errors over training samples. However, through ERM, large models may memorize some unique feature patterns relating to training samples, especially when available data is limited. For instance, in order to distinguish visually similar samples with different labels, ERM encourages models to be as confident as possible of the predictions, thus force models to capture the most obviously discriminative feature representations to separate samples well, which may be the less generalized feature patterns that only relate to specific training samples (*i.e.* background noises), and this phenomenon is especially severe when lacking rich information from sufficient data. Therefore training only with ERM may provide poor generalization on test data that only slightly deviate the training data.

Intuitively, we propose to bring confusion to the feature extraction by pulling features of training samples closer to each other to penalize the strongly discriminative feature extraction under ERM training, and this will prevent the feature extractors from over-representing training samples. Specifically, we formalize this intuition and propose Feature Variance Regularization (FVR) that regularizes the empirical risk using a penalty on the empirical variance of the features.

We also provide a theoretical explanation of FVR based on previous statistical learning theories (Maurer and Pontil 2009; Namkoong and Duchi 2017; Tolstikhin and Seldin 2013). We prove that through FVR, the discrepancy between the features of training samples and unobservable true data will be reduced, and the expected risk will be minimized, which means better generalizability of the model.

Despite its simplicity, our experiments demonstrate FVR outperforms many other regularizers, and the performance is consistent across multiple tasks, including classification (CIFAR100 (Krizhevsky and Hinton 2009), ImageNet (Deng et al. 2009), fine-grained datasets (Wah et al. 2011; Yang et al. 2015; Maji et al. 2013)), and semantic segmentation (Cityscapes (Cordts et al. 2016)). And our ablation study experiments also show our approach could improve the transfer ability, make the feature map less responsive to environmental noises and is robust to label noise and the choices of hyperparameters.

Related works

Regularization

Numerous methods have been proposed to improve the generalization abilities of neural networks. $L1$ and $L2$ regularization (Ng 2004) are widely used to regularize ERM by constraining model weights. Dropout (Srivastava et al. 2014) ensembles exponentially many thinned networks efficiently to avoid over-fitting. Adversarial training (Goodfellow, Shlens, and Szegedy 2014) is a regularization method

[*]Work done during an internship at Xilinx.
[†]Corresponding author

applied on the input level and consider the input perturbation towards the direction that increases the loss most. Virtual adversarial training (Miyato et al. 2018) develops it and use current probabilities generated by networks as virtual labels to replace true labels, so that it can be applied to semi-supervised learning as well. Another approach is label smooth regularization (Szegedy et al. 2016), which prevents the largest logit from becoming much larger than all others by changing the construction of ground-truth label distributions.

Different from their concerns, our proposed method directly acts on the features extracted by the network and bring confusion into the feature extraction to prevent the feature extractors from over-representing training samples. Therefore, penalty on feature variance could also help to prevent the prediction from being too confident on training samples which is similar to label smooth. Also, our regularization does not involve in any labels, so could still work in unsupervised manners or existence of label noise. In our theoretical part, we prove that the discrepancy between the features of training samples and unobservable true data will be reduced through FVR.

Variance-based theories

Some variance-based bounds have been explored in a number of researches (Hoeffding 1994; Maurer and Pontil 2009; Namkoong and Duchi 2017; Tolstikhin and Seldin 2013; Audibert, Munos, and Szepesvári 2009). Hoeffding's inequality (Hoeffding 1994) is independent of the hypothesis, to improve it, Bennett's inequality (Hoeffding 1994) provides us with estimates of lower accuracy for hypotheses of large variance, and higher accuracy for hypotheses of small variance. However, the upper bound of Bennett's inequality depends on the unobservable variance. To overcome this drawback, some empirical variance-based bounds are derived in (Maurer and Pontil 2009; Audibert, Munos, and Szepesvári 2009), in which the upper bound depends on the observable empirical variance. Thus the basic idea is that hypotheses with small variance could better estimate the true quantity. In previous works (Maurer and Pontil 2009; Namkoong and Duchi 2017), the empirical variance bound is applied to the loss functions. Our presented confidence bound builds on previous analysis. Instead, we focus on confidence bounds based on the empirical variance of features and develop confidence bounds for our designed variance calculation form. Additionally, we extend formulations from single-dimensional random variables to the multi-dimensional scenario.

Algorithm

We propose to regularize the empirical risk using a penalty on the empirical variance of the features. In this section, we will give a theoretical explanation of our proposed method. Firstly we will give a definition of *empirical feature variance*. Then we propose *feature deviation* to measure the stability and generalizability of extracted features. And via the presented confidence upper-bounds, we demonstrate that *feature deviation* could be restrained by *empirical feature variance*, thus better generalization performance of features can be obtained. We further discuss the relationship between the penalty on *empirical feature variance* and the expected risk.

Consider the classification problem. Let $\mathcal{X}$ be the input domain, and P a distribution on $\mathcal{X}$. The training samples are given by N i.i.d samples $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ drawn from $\mathcal{X}$ according to the true data-generating distribution P. Each data point $\mathbf{x}$ is associated with with a ground-truth label $\mathbf{y}$. During training, we learn parameters of the classifier $\mathbf{w}$ and feature extractor $\Phi(\cdot)$. We assume $\Phi(\cdot)$ maps the input $\mathbf{x}$ to a M-dimensional feature $\Phi(\mathbf{x})$, which can be described as $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \cdots, \phi_M(\mathbf{x}))$, and each $\phi(\cdot)$ corresponds to a one-dimensional feature extractor.

Feature variance

Firstly, to characterizes the variations of the feature distributions among training samples, we define *empirical feature variance* in Definition 1. $Var[\phi(\mathbf{X})]$ measures the dispersion among different samples along each feature dimension, and $Var[\Phi(\mathbf{X})]$ sums up the dispersion of all feature dimensions.

Definition 1 *Let P be a distribution on $\mathcal{X}$, and for each one-dimensional feature extractor $\phi(\cdot)$, the empirical feature variance on training samples is defined as:*

$$Var[\phi(\mathbf{X})] = \frac{1}{N-1}\sum_{i=1}^{N}\left(\phi(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(\mathbf{x}_j)\right)^2 \quad (1)$$

for the M-dimensional feature extractor $\Phi(\cdot)$, the empirical feature variance on training samples can be defined as:

$$Var[\Phi(\mathbf{X})] = \sum_{j=1}^{M} Var[\phi_j(\mathbf{X})] \quad (2)$$

Feature deviation

We now propose *feature deviation* to measure the generalizability and stability of extracted features. We are motivated by similar ideas in domain adaption (Ganin et al. 2016; Tzeng et al. 2014) where feature distributions of training samples and test samples from different domains are aligned to get better performance on test samples.

Different from it, we are concerned about training and test set belonging to the same input domain $\mathcal{X}$ with P distribution. Considering training data are sampled from the unobservable true data subject to P distribution, we encourage the feature extractor to generate similar feature distribution for the training set and true data. This allows the classifier designed for the training features to adapt better to the unobservable true data, thus better generalization performance can be obtained. To achieve this goal, we use the distance between the empirical mean of training features and their corresponding expected true population mean as the measure of discrepancy, and define the form of *feature deviation* in Definition 2. A smaller *feature deviation* means a smaller feature discrepancy between the features extracted from training set and true data, and represents a better generalizability of the feature extractor.

Definition 2 *Under the conditions of Definition 1, for each one-dimensional feature extractor $\phi(\cdot)$, feature deviation is defined as $Dev(P, \mathbf{X}, \phi)$.*

$$\begin{aligned} Dev(P, \mathbf{X}, \phi) &= |E\phi(P) - E\phi(\mathbf{X})| \\ E\phi(\mathbf{X}) &= \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i) \\ E\phi(P) &= \mathbb{E}_{\mathbf{x}\sim P}\phi(\mathbf{x}) \end{aligned} \tag{3}$$

for the M-dimensional feature extractor $\Phi(\cdot)$, feature deviation can be defined as:

$$Dev(P, \mathbf{X}, \Phi) = \sqrt{\sum_{j=1}^{M} Dev(P, \mathbf{X}, \phi_j)^2} \tag{4}$$

Variance-based confidence bound

Built on previous statistical learning theories (Maurer and Pontil 2009; Namkoong and Duchi 2017; Tolstikhin and Seldin 2013), we give the relationship between the our proposed *empirical feature variance* and *feature deviation*.

Corollary 1 *Under the conditions of Definition 1 and 2, let $\mathcal{F}$ be a finite class of functions $\phi : \mathcal{X} \to [0, 1]$, for each $\delta \in (0, 1)$, with probability at least $1 - \delta$, $\forall \phi \in \mathcal{F}$:*

$$\begin{aligned} Dev(P, \mathbf{X}, \phi) \leq & \sqrt{\frac{4(N-1)Var\left[\phi(\mathbf{X})\right]\ln(3|\mathcal{F}|/\delta)}{N^2}} \\ & + \frac{5\ln(3|\mathcal{F}|/\delta) + N}{2N} \end{aligned} \tag{5}$$

where $|\mathcal{F}|$ is the cardinality of $\mathcal{F}$.

For a more concise and simple formula expression, without loss of generality, we choose $[0, 1]$ as the feature space and it can be easily extended to a more general feature space. Observed form Corollary 1, *feature deviation* can be upper-bounded by *empirical feature variance*. We then extend Corollary1 to multi-dimensional extracted features.

Corollary 2 *Under the conditions of Corollary 1 for each $\delta \in (0, 1)$, with probability at least $1 - M\delta$, $\forall \phi \in \mathcal{F}$*

$$\begin{aligned} Dev(P, \mathbf{X}, \Phi) &\leq G\left(Var\left[\Phi(\mathbf{X})\right], \delta\right) \\ G(t, \delta) &= \sqrt{\alpha(\delta)^2 t + M\beta(\delta)^2 + 2\alpha(\delta)\beta(\delta)\sqrt{Mt}} \end{aligned} \tag{6}$$

where $\alpha(\delta)$, $\beta(\delta)$ are irrelevant to $Var\left[\Phi(\mathbf{X})\right]$ and related to δ.

Corollary 2 demonstrates that *feature deviation* could be restrained by *feature variance* with high confidence using a limited number of samples. Since it is a common practice to minimize upper bounds, we propose a penalty of feature variance to constrain the *feature deviation* and obtain more generalized features.

Discussion

To further study the effect of our proposed empirical variance on expected risk, we take a simple binary classification problem as an example. Assuming the inputs of N samples are $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$, the corresponding labels are $Y = (y_1, \cdots, y_N)$, and (x, y) is subject to a joint distribution Q. The feature extractor is defined as $\phi(\cdot)$, so the features for each sample are described as: $\phi(\mathbf{X}) = (\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_N))$. We compute sigmoid loss for each sample as follows:

$$\begin{aligned} z &= l(\phi(\mathbf{x}), y) \\ &= -y\log(\frac{1}{1+e^{-\phi(\mathbf{x})}}) - (1-y)\log(\frac{1}{1+e^{\phi(\mathbf{x})}}) \end{aligned} \tag{7}$$

Based on Equation 7, we can deduce that:

$$\begin{aligned} |z_i - z_j| &\leq |\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)| + 1 \\ \forall i, j &\in \{1, ..., N\} \end{aligned} \tag{8}$$

We define the expected risk in Equation 9 and empirical risk in Equation 10 respectively.

$$El(\phi, Q) = \mathbb{E}_{(\mathbf{x},y)\sim Q} l(\phi(\mathbf{x}), y) \tag{9}$$

$$El(\phi(\mathbf{X}), Y) = \frac{1}{N}\sum_{i=1}^{N} l(\phi(\mathbf{x}_i), y_i) \tag{10}$$

Combine Equation 8 and Corollary 1, for each $\delta \in (0, 1)$, with probability at least $1 - \delta$, $\forall \phi \in \mathcal{F}$, we can give following results:

$$\begin{aligned} |El(\phi, Q) - El(\phi(X), Y)| \leq & \frac{7\ln(3|\mathcal{F}|/\delta) + 3N}{2N} \\ & + \sqrt{\frac{8(N-1)Var\left[\phi(\mathbf{X})\right]\ln(3|\mathcal{F}|/\delta)}{N^2}} \end{aligned} \tag{11}$$

In Equation 11, we demonstrate that feature variance could constrain the gap between expected risk and empirical risk, since ERM is used to minimize the empirical risk, thus a penalty on empirical feature variance could help minimize the expected error and provide a better generalization result.

FVR algorithm

Consider the multi-class classification problem over C classes. Assume $\mathbf{w}$ represents the parameters of the classifier, $\Phi(\cdot)$ represents the feature extractor, we adopt the widely used cross entropy loss as empirical risk.

$$CE(\mathbf{w}\Phi(\mathbf{X}), \mathbf{y}) = -\sum_{i=1}^{N} log\frac{e^{\mathbf{w}_{y_i}^T\Phi(\mathbf{x}_i)}}{\sum_{j=1}^{C} e^{\mathbf{w}_j^T\Phi(\mathbf{x}_i)}} \tag{12}$$

We propose to add a penalty of empirical feature variance to the original classification loss. The formulation is given in Equation 13:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_S + \gamma\mathcal{L}_{FVR} \\ &= CE(\mathbf{w}\Phi(\mathbf{X}), \mathbf{y}) + \gamma Var\left[\Phi(\mathbf{X})\right] \end{aligned} \tag{13}$$

where γ is a regularization factor.

Feature variance regularization can be understood intuitively. Training data are only a small part sampled from the

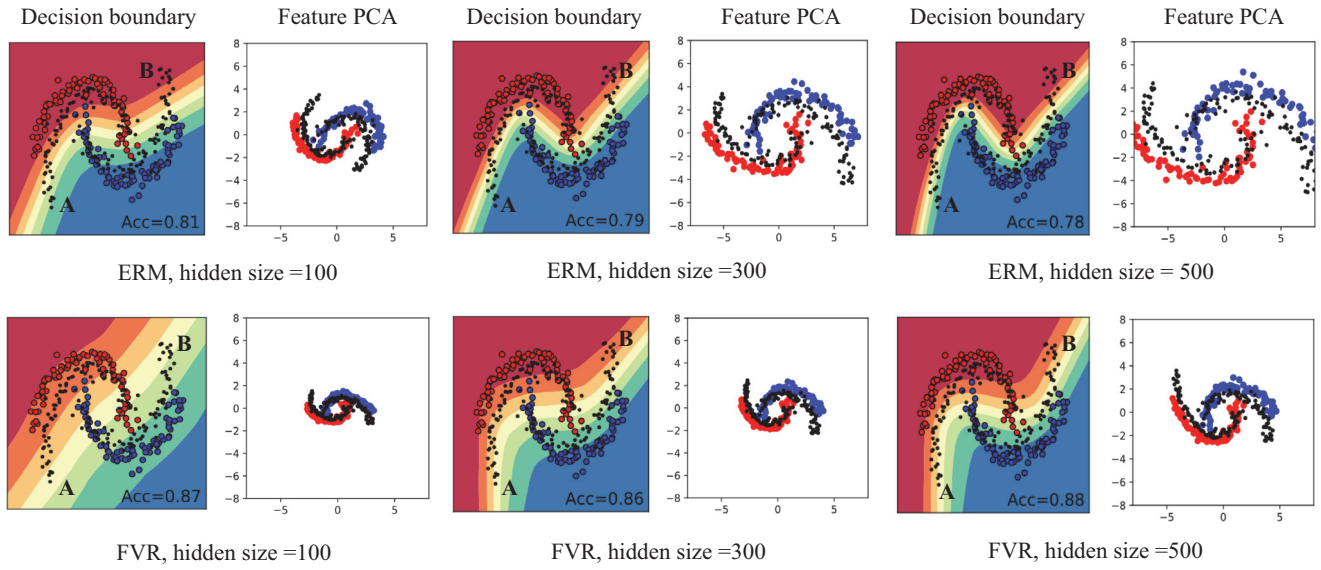


Figure 1: The interleaving moons 2D problem under ERM training and FVR training. Training samples are represented by red dots (label 0) and blue dots (label 1), test samples are represented by black dots. The test accuracy is annotated at the right corner of each decision boundary graph. See text for detailed discussion.[Best viewed in color]

true data-generating distribution, it is hard for feature extractors learned from this subset to represent new samples well, especially when empirical risk minimization encourages models to be as confident as possible of the predictions only on training samples. Since ERM promotes to generate relatively separated features to get discriminative results, bringing in confusion into feature extraction by feature variance regularization can be seen as a modifying power and prevent the feature extractors from over-representing training samples.

Also, our theoretical explanation proves that the *feature deviation* can be upper-bounded by *empirical feature variance*, which means a constraint on the feature variance works for generating similar feature distribution for the training set and unobservable true data. Also expected error could be minimized through FVR. Therefore FVR helps to improve the generalization performance of models.

Experiments

This section presents our experimental results. Firstly, experiments on a toy example are performed to study the effect of our method on decision boundary and feature representation. Then we compare FVR to other regularization methods on CIFAR100 and study the performance on ImageNet. And experiments on fine-grained visual classification and semantic segmentation are followed. We use Pytorch framework (Paszke et al. 2017) to implement our experiments on GeForce GTX 1080 GPUs.

Toy example

In this part, a toy example on interleaving moons 2D problem is presented, and the results are in Figure 1. Training data contains 150 samples in total, consisting of an upper moon (red dots) and a lower moon (blue dots). To verify the generalizability of the model, we generate test samples (black dots) by rotating each sample in training data by $30°$. We study the effect of FVR on decision boundaries and feature representations by comparing it to ERM training. The two-layer perceptron is chosen as the network architecture, with different hidden layer sizes (100, 300, 500).

(1) Effect on decision boundary. Some observations can be gained from the column "Decision boundary" in Figure 1.

- For the same training procedure, a larger hidden size leads to a more compact decision boundary with large curvature. In the case of ERM training, the test accuracy annotated in the lower right corner of the map decreases, which usually indicates poor generalization of the model. While FVR training prevents the accuracy from dropping.
- For the same hidden size, FVR training provides a smoother and sparser decision boundary than ERM training, with a higher test accuracy indicating better generalization ability. ERM training misclassifies some edge points around A and B in test samples, while FVR reduces the misclassification probability.

(2) Effect on feature representation. We perform a 2-dimensional PCA transformation on the features of the hidden layer and conduct a visualization.

- For the same training procedure, a larger hidden size leads to a more dispersed feature distribution and larger feature variance, and a larger deviation between training features and test features.
- For the same hidden size, ERM training results in a larger discrepancy between training features and test features compared to FVR training, and some test samples fall in

the space without any training samples. However, FVR constrains feature variance and allows the feature extractor to adapt test samples better, which means that the classifier obtained from the training set can better classify the test set. This is consistent with our theoretical analysis that FVR could reduce the feature deviation and provide more generalized results.

Experiments on CIFAR100

In our following experiments, we evaluate the efficacy of our approach on CIFAR100 (Krizhevsky and Hinton 2009). CIFAR100 consists of training images of size 50K and test images of size 10K, where samples are 32 x 32 color images from 100 categories. All experiments are carried out with an SGD optimizer with a momentum of 0.9, a batch size of 128 and a total epoch number of 150. The initial learning rate is set to 0.1 which is subsequently decayed by 0.1 at epoch 60, 90, 130, respectively. We choose ResNet18 (He et al. 2016) and MobileNetV2 (Sandler et al. 2018) as our network architectures.

Comparison to other regularization methods We now compare FVR to other regularization techniques on CIFAR100 and the results are shown in Table 1. We divide experimental results into three groups:

- ERM: The optimizer is empirical risk minimization without any regularizers.
- Single regularizer: We compare our proposed method to L2 and L1 weight decay, dropout (Srivastava et al. 2014), virtual adversarial training (Miyato et al. 2018) and label smooth (Szegedy et al. 2016). Different from the concerns on model weights (L1/L2 weight decay), network architecture (dropout), network input (VAT) and prediction probability (label smooth), FVR is applied directly on features and aims to constrain feature variance to get generalized feature representations.

  For our proposed method, we apply FVR to more representative feature layers (the penultimate feature and the output feature) rather than shallow layers. We study the effect of one-scale constraint (applying FVR only on the penultimate layer or the output layer) and two-scale constraint (applying FVR both on the penultimate layer and the output layer) respectively. As can be observed from the results, our method achieves the largest accuracy improvement compared to other regularizers. And two-scale constraints of FVR brings about larger performance gain.
- Combinations of different regularizers: Since different regularization methods have different principles, a combination of them may give better results. The combination of L2 weight decay and FVR creates the largest performance improvement over ERM.

Without a special explanation, the "baseline" in our experiments below refers to models trained with ERM and L2 weight decay, because it is a widely-used regularization technique. For convenience, in later experiments, we only choose the penultimate layer to conduct feature variance regularization, and multiple constraints could be potentially used to get better results.

Table 1: Test accuracy compared to other regularization methods on CIFAR-100

Method	ResNet18	MobileNetV2
ERM	72.94	73.05
L2	74.76	74.26
L1	73.06	73.08
dropout	74.18	73.30
VAT	71.39	71.97
Label smooth	74.70	74.12
FVR(penultimate)	**75.20**	**74.44**
FVR(output)	74.96	73.68
FVR(2-scale)	**76.22**	**74.98**
L2+dropout	75.87	74.55
L2+VAT	74.36	74.10
L2+Label smooth	76.40	74.96
L2+FVR(penultimate)	**77.07**	**75.52**
L2+FVR(output)	76.44	75.38
L2+FVR(2-scale)	**77.56**	**76.33**

Table 2: Test accuracy compared to baselines on ImageNet under different data size on ResNet50

Training Data	Method	Top1	Top5
All data	baseline	76.23	92.97
	FVR	**76.80**	**93.24**
30% of all data	baseline	66.91	87.16
	FVR	**67.66**	**87.75**
10% of all data	baseline	50.68	74.71
	FVR	**52.39**	**76.46**

Experiments on ImageNet

As a large scale dataset, ImageNet (Deng et al. 2009) has 1.2M training images and 50K test images from 1000 categories. With nearly 1000 samples in each category, we suffer from a less serious over-fitting problem when training ImageNet than small datasets. We make use of ImageNet to study the performance of FVR under the different size of training data on ResNet50 (He et al. 2016). For implement details, the initial learning rate is 0.1 and divided by 10 at 30, 60, 90 and 110 epoch, respectively. The models are trained for 120 epochs with a mini-batch size as 256. The weight decay is set to 0.0001. As can be seen from Table 2, with all training data provided, the improvement of top1 and top5 are 0.57% and 0.27%, respectively. When the training data size is reduced to 30% and 10% of the original size with the test dataset unchanged, which means more serious over-fitting problem, FVR achieves larger performance gain over ERM, certifying effects of preventing over-fitting.

Fine-Grained classification

Fine-grained Visual Categorization(FGVC) aims at identifying sub-categories of the same super-category and has been a challenging task because of the high inter-class similarity and the data shortage. The mismatch between classification difficulty and available data size leads to generalization problems. Our experiments are conducted on three representative FGVC datasets, *i.e*. CUB-200-2011 (Wah et al. 2011), Stanford Cars (Yang et al. 2015) and FGVC Aircraft (Maji

Table 3: Test accuracy compared to previous works on CUB-200-2011, Stanford Cars, FGVC Aircraft

Methods	*CUB-200-2011*			*Stanford Cars*			*FGVC Aircraft*		
	VGG16	ResNet50	Inception	VGG16	ResNet50	Inception	VGG16	ResNet50	Inception
baseline	79.1	85.4	83.1	87.0	91.7	91.2	85.1	88.1	88.4
B-CNN	84.1	-	-	**91.3**	-	-	84.1	-	-
CBP	84.0	-	-	-	-	-	-	-	-
LRBP	84.2	-	-	90.9	-	-	87.3	-	-
ST-CNN	-	-	84.1	-	-	-	-	-	-
FCAN	-	84.3	-	-	91.5	-	-	-	-
FVR(Ours)	**84.6**	**87.1**	**85.2**	91.1	**93.8**	**92.1**	**88.1**	**90.7**	**89.3**

et al. 2013) on three backbones, *i.e*. VGG16 (Simonyan and Zisserman 2014), ResNet50 (He et al. 2016), Inception (Ioffe and Szegedy 2015). CUB-200-2011 are birds images from 200 classes officially split into 5,994 training and 5,794 test images. Stanford Cars are car images from 196 classes officially split into 8,144 training and 8,041 test images. FGVC-Aircraft is aircraft images from 100 classes officially split into 6,667 training and 3,333 test images. We train networks with a batch size of 64, weight decay of 1e-4 and the total epoch number of 120. The initial learning rate is set to 0.01, which is subsequently decayed by 0.1 at epoch 40, 70, 100, respectively.

The results are shown in Table 3. FVR outperforms baselines by a large margin, such as 5.5% for CUB-200-2011 on VGG-16. We also compare FVR to some previous works specifically designed for FGVC, *i.e*. B-CNN (Lin, Roy-Chowdhury, and Maji 2015), CBP (Gao et al. 2016), LRBP (Kong and Fowlkes 2017), ST-CNN (Jaderberg et al. 2015) FCAN (Liu et al. 2016). To be fair, we compare to previous fine-grained methods under the same architectures and input size(448×448) as our method. Our approach even rivals many specially-designed previous methods for the fine-grained datasets.

Semantic segmentation

We evaluate FVR on semantic segmentation task, and adopt ENet (Paszke et al. 2016) as the base model, Cityscapes (Cordts et al. 2016) as benchmark dataset. Cityscapes dataset focuses on urban visual scene understanding and consists of 2,975 training, 500 validation and 1525 testing images with fine-grained annotations. The task is to segment an image into 19 classes belonging to 7 categories (*e.g*. person and rider belong to the same category human). All images are in a resolution of 1024 x 2048. We obtain our performance on validation and testing images using Cityscapes online sever. In the training process, we use poly learning rate policy with base learning rate 0.01, set weight decay to 0.0001, and set training batch size to 12.

Since segmentation can be regarded as pixel-wise classification, we compute feature variance using Equation (2) by regarding each pixel as a sample. According to Table 4. FVR gives an mIOU gain by 1.8% in Cityscapes validation set and 1.0% in Cityscapes test set.

Table 4: mIOU/% on Cityscapes Validation set and test set on ENet

Method	Validation	Test
baseline	61.1	60.9
FVR	**62.9**	**61.9**

Ablation studies

Some ablation studies are carried out in this section. We first explore the transfer ability of FVR on cross-domain visual classification tasks. Then we visualize the feature maps and note that FVR reduces the attention on background noises. We also demonstrate the robustness to label noise and present training curves.

Cross-domain visual classification

We validate our proposed method in an unsupervised adaptation task among SVHN (Netzer et al. 2011), MNIST (LeCun et al. 1998), USPS (Hull 1994) which are composed of 10 classes of digits. We consider two cross-domain pairs: SVHN→MNIST and USPS→MNIST, and use the same ResNet18 architecture in CIFAR100 experiments.

Results of our experiments are provided in Table 5. Our proposed method consistently improve over the baseline models in source-only setting, with an average margin of 7.7%. And FVR even has an improvement when domain adaption method DDC (Tzeng et al. 2014) is used in the source + target setting. It proves that not only does FVR improve the generalizability of features, but also promotes the transfer ability.

Table 5: Results on digital recognition datasets for unsupervised domain adaption based on ResNet-18. "S" represents for source domain only, "S+T" represents for source domain + target domain.

Methods	SVHN→MNIST	USPS→MNIST	Avg
S	77.2	69.3	73.3
S + FVR	**81.4**	**80.6**	**81.0**
S + T	93.4	82.5	87.9
S + T + FVR	**94.2**	**85.2**	**89.7**

To further study the effect of FVR on transfer ability, we consider a special cross-domain pair: CUB-200-

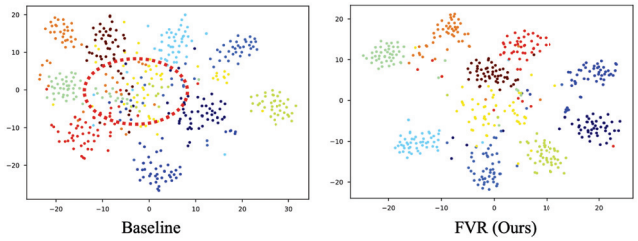


Figure 2: Visualization of features extracted on target domain by source models, in CUB-200-2011→ImageNet Birds experiment pair. Source and Target domain have different fine-grained labels while coming from the same super-category. Each color corresponds to a category. [Best viewed in color]

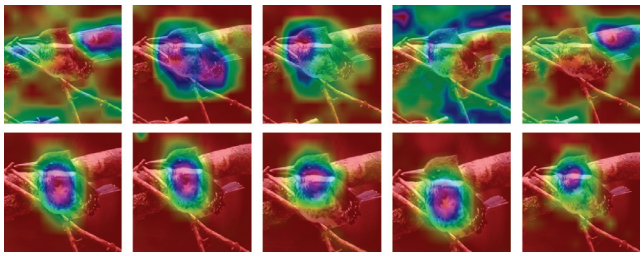


Figure 3: Row1: Feature maps of baseline. Row2: Feature maps of FVR model. Each column represents each feature channel. All evaluations are conducted on CUB-200-2011 validation set. Drawn after normalization. [Best viewed in color]

2011→ImageNet Birds, and the target domain consists of 10 species of birds in ImageNet which have no overlap with bird species in CUB-200-2011. We compare the transfer ability of the source model trained with and without FVR, and visualize feature distributions by t-SNE (Maaten and Hinton 2008). As shown in Figure 2, with FVR training, the source model extracts more discriminative features on the target domain, even if categories are not shared between domains. This demonstrates that FVR promotes the extraction of information about super-category, *i.e.* birds, therefore enhance transfer between different fine-grained categories.

Feature map visualization and analysis

To observe the behaviors of FVR on feature extraction, we also visualize the middle feature map. To be specific, we visualize ResNet50 conv4_3 feature maps on CUB-200-2011 validation set using algorithm in (Selvaraju et al. 2017). We observe feature maps belonging to feature dimensions with the top five average activation values. Figure 3 shows under the variance constraint on each dimension, the corresponding feature maps become less responsive to the environment.

Label noise

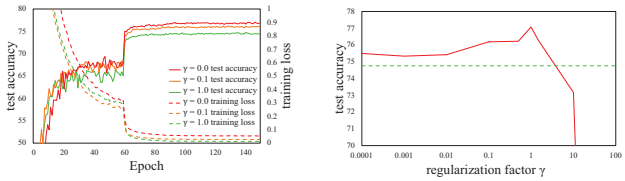
In this experiment, we introduce noise to labels of CIFAR100 by randomly permuting a fraction of labels in training data. As shown in Table 6, we observe that FVR allows the network to be more robust to label noise compared to baseline and label smooth. The regularization on feature variance does not need to involve in label information, therefore provides more robust performance under label noise.

Table 6: Test accuracy of FVR on CIFAR100 with label noise using ResNet18 compared to label smooth

Label Noise	Baseline	Label smooth	FVR
20%	66.11	67.92	**69.23**
40%	62.96	63.30	**65.00**
60%	52.64	52.26	**54.73**

Curves and the choice of hyperparameter

For CIFAR100 experiments, we draw training curves in Figure 4(a). The higher test accuracy is achieved when training with feature variance regularization, while training loss converges to a higher level, since FVR prevents the network from over-fitting to the training samples. We also observe a stable performance of FVR to the choice of regularization factor γ (Figure 4(b)).



(a) (b)

Figure 4: (a) FVR training gives higher test accuracy despite higher training loss. (b) The robustness of FVR to the hyper-parameter γ.

Conclusion

Feature variance regularization is a simple and efficient technique. We come up with this idea in an intuitive way to prevent the models from learning sample-based features. And our theoretical analysis proves that our method could restrain the deviation of features on the training set and true data and minimize the expected error. Our experiments show FVR achieves promising performances in multiple tasks. Thus it could be a useful tool in the training process of neural networks.

Acknowledgments

This work was supported by National Key Research and Development Program of China (No. 2018YFB0105005), National Natural Science Foundation of China (No. 61832007, 61622403, 61621091), Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua Xilinx AI Research Fund, and the authors gratefully acknowledge the support from TOYOTA.

References

Audibert, J.-Y.; Munos, R.; and Szepesvári, C. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410(19):1876–1902.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

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.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17(1):2096–2030.

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T. 2016. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 317–326.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

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.

Hoeffding, W. 1994. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*. Springer. 409–426.

Hull, J. J. 1994. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence* 16(5):550–554.

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

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *Advances in neural information processing systems*, 2017–2025.

Kong, S., and Fowlkes, C. 2017. Low-rank bilinear pooling for fine-grained classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 365–374.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2015. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, 1449–1457.

Liu, X.; Xia, T.; Wang, J.; Yang, Y.; Zhou, F.; and Lin, Y. 2016. Fully convolutional attention networks for fine-grained recognition. *arXiv preprint arXiv:1603.06765*.

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

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Maurer, A., and Pontil, M. 2009. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*.

Miyato, T.; Maeda, S.-i.; Ishii, S.; and Koyama, M. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*.

Namkoong, H., and Duchi, J. C. 2017. Variance-based regularization with convex objectives. In *Advances in Neural Information Processing Systems*, 2971–2980.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Ng, A. Y. 2004. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, 78. ACM.

Paszke, A.; Chaurasia, A.; Kim, S.; and Culurciello, E. 2016. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

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.

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.

Tolstikhin, I. O., and Seldin, Y. 2013. Pac-bayes-empirical-bernstein inequality. In *Advances in Neural Information Processing Systems*, 109–117.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Yang, L.; Luo, P.; Change Loy, C.; and Tang, X. 2015. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3973–3981.