WCP: Worst-Case Perturbations for Semi-Supervised Deep Learning

Liheng Zhang [1], Guo-Jun Qi [1,2,*]
[1]Laboratory for MAchine Perception and LEarning (MAPLE)
http://maple-lab.net/
[2]Futurewei Technologies
https://github.com/maple-research-lab/WCP

Abstract

In this paper, we present a novel regularization mechanism for training deep networks by minimizing the Worse-Case Perturbation *(WCP). It is based on the idea that a robust model is least likely to be affected by small perturbations, such that its output decisions should be as stable as possible on both labeled and unlabeled examples. We will consider two forms of WCP regularizations – additive and DropConnect perturbations, which impose additive noises on network weights, and make structural changes by dropping the network connections, respectively. We will show that the worse cases of both perturbations can be derived by solving respective optimization problems with spectral methods. The WCP can be minimized on both labeled and unlabeled data so that networks can be trained in a semi-supervised fashion. This leads to a novel paradigm of semi-supervised classifiers by stabilizing the predicted outputs in presence of the worse-case perturbations imposed on the network weights and structures. We conduct experiments to demonstrate the proposed method outperforms many state-of-the-art models in literature. The source code will be released after the paper is accepted for publication.*

1. Introduction

When training a predictive model f_θ with parameters θ, the idea behind many "denoising" approaches in literature [30, 9] is to train a robust model that would not change its predictions abruptly in presence of model noises. In this paper, we will show that the idea can be further elaborated to train a regularized model by minimizing the change of model predictions in the *worst* case when a given magnitude of perturbations is presented.

For example, for a sigmoid classifier, it would be intuitive to learn a preferred linear boundary that has the largest margin to separate datapoints of different classes. This large margin principle is well connected with the idea of minimizing the impact of worst-case model perturbation, since a maximum-margin classifier [2, 3, 21, 18] is *least* likely to change its predictions when it is maximally perturbed. We refer the readers to the deferred example shown in Figure 1 that will be discussed in Section 4.1 in the context of additive perturbation. Although it is a simple example, it reveals the intrinsic relation between the classic large margin principle and the worst-case perturbation regularization, while the latter can be applied to a general nonlinear model and unsupervised data.

Formally, in this paper, we present a novel paradigm of regularized deep networks by minimizing the impact of *Worse-Case Perturbations* (WCP) to train robust models. We will present two forms of WCP mechanisms – the Additive Perturbation with additive noises on model weights, and the DropConnect Perturbation by making structural changes by dropping network connections. We will show how to tractably derive the worse-case perturbations that maximally change the network predictions, and integrate them to regularize the training of the network weights and structures. We will apply the proposed WCP regular-

[*] Corresponding author: G.-J. Qi. Email: guojunq@gmail.com

izer to explore both labeled and unlabeled data in a semi-supervised fashion, yielding the classifiers that have stable predictions against worse perturbations.

The remainder of the paper is organized as follows. We discuss the relation with the existing works in Section 2. In Section 3, we present the proposed formulation of the WCP regularization, followed by the additive perturbation and the DropConnect perturbation respectively in Section 4 and Section 5. We show how to integrate both perturbations to train a robust network in Section 6. We demonstrate the superior performances of the WCP-regularized model through experiments in Section 7, and conclude the paper in Section 8.

2. Related Works

There exist several works in literature that regularize the model training by *randomly* corrupting networks and/or data. Among them is the seminal dropout regularization that randomly removes neurons when training networks [8]. Along this line, the DropConnect [31] is a natural extension by randomly dropping neural connections during the training. Essentially, the removal of neurons and their connections can be seen as forming an ensemble of network architectures in the training process, yielding a robust network by "averaging" over the resultant ensemble. In parallel, some other models seek to improve the consistency of predicted outputs over the perturbed data to the train semi-supervised models. These include the GAN-based methods that train a multi-class discriminator to distinguish fake samples from the real classes [16], or learn localized generators [23] to investigate the output consistency along the manifold.

This idea has been further extended to train semi-supervised classifiers [36, 14, 1, 33, 26, 27, 25, 20, 28, 6, 17, 19] by temporally fusing networks through the training process. For example, [9] propose to predict target values of unlabeled examples by taking an exponential moving average of the predictions from stochastic networks trained with random dropouts and data augmentations over recent iterations. [29] further extend the idea of the temporal ensembling to maximize the consistency of predictions between mean teachers and the running student networks on both labeled and unlabeled data. [34] combined multiple transformations to explore the self-supervised training of semi-supervised classifiers in an ensemble in an auto-encoding transformation fashion [35, 22, 15, 32] .

Instead of generating an ensemble of *randomly* corrupted networks, the WCP aims to find and enhance the most vulnerable part of a network by making the weights and connections most resilient against the worst-case perturbations. In contrast to the WCP that directly imposes perturbations on network weights and structures, [10] explore the vulnerability of a network through virtual adversarial examples that would maximally alter the network predictions. The WCP is orthogonal to the approach that uses the virtual adversarial examples to train a robust classifier [10]. In contrast, the WCP seeks to reveal and enhance the most vulnerable part of the model in terms of both additive and dropconnect noises.

As aforementioned, the WCP is also well connected with the idea of training robust models with the large margin principle [2, 3, 4, 18] that has gained success before the deep learning era. Thus, in this paper, we also aim to close the research gap by bringing the principle of classic regularization methods to train more competitive deep networks.

3. The Formulation

In this section, we present the proposed Worst-Case Perturbations (WCP) for regularizing deep networks, and show its application to semi-supervised learning by exploring unlabeled data to train networks.

Formally, consider a deep network $f_\theta(\mathbf{x})$ that takes $\mathbf{x}$ sampled from a data distribution $\mathcal{D}$ as input and has $\theta \in \Theta$ as its weights. Suppose there is a perturbation function g applied to the model weights (additive perturbation) and structures (dropconnect perturbation), resulting in a perturbed version of the model $f_{g(\theta)}(\mathbf{x})$.

We have some constraints $\mathcal{G}$ on a perturbation function such that the it would not arbitrarily perturb the model. For example, for an additive perturbation, we can restrict the largest norm on the additive noise added to the model; for a dropconnect perturbation, the maximum number of dropped connections can be set. We will discuss these constraints later in detail.

Then the worse-case perturbations subject to the constraints can be solved by

$$\Omega_\theta = \max_{g \in \mathcal{G}} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}} \ell\left(f_\theta(\mathbf{x}), f_{g(\theta)}(\mathbf{x})\right) \tag{1}$$

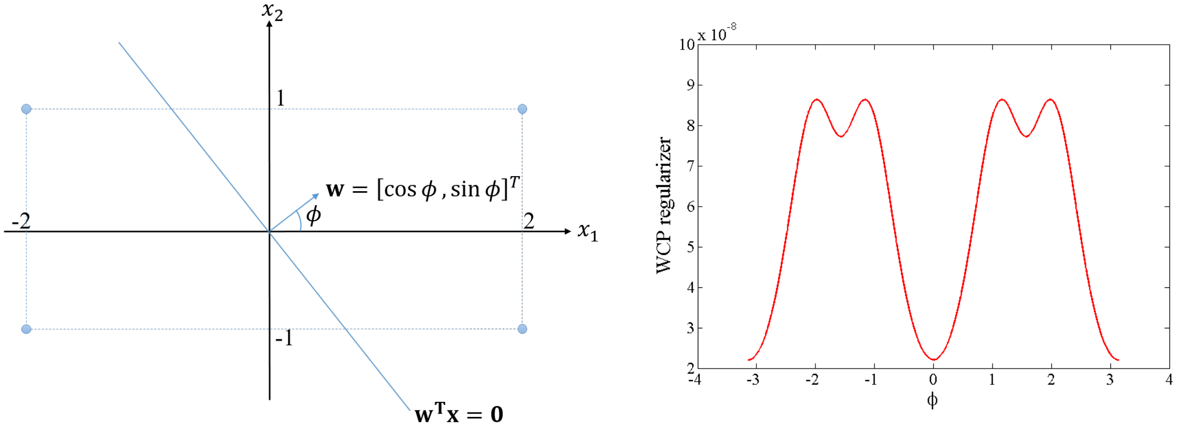

Figure 1: (a) A toy example of a sigmoid unit for four datapoints; (b) the relation between ϕ and the corresponding WCP value. From the "teeth" curve, the minimum WCP regularizer occurs at $\phi = 0$ (similary at $\phi = \pm\pi$) with the corresponding boundary $x_1 = 0$. Two local minima of the WCP regularizer occur at $\pm\frac{\pi}{2}$, corresponding to $x_2 = 0$. The result is obtained with $\epsilon = 10^{-3}$.

where $\ell(\cdot,\cdot)$ is a loss function measuring the difference between the outputs of the original and perturbed functions (e.g., squared ℓ_2 loss and Kullback-Leibler divergence), and we use the expected change of the network outputs f_θ over the data distribution $\mathcal{D}$ to quantify how much the model has been perturbed.

Here we assume the loss function ℓ has the following properties:

- $\ell(y,z) = 0$ when $y = z$;
- $\ell(y,z) \geq 0$, i.e., its minimal value is zero;
- $\ell(y,z)$ is at least twice differentiable.

We wish to minimize the impact of worse-case perturbations as a way to regularize the training of deep networks. In other words, this encourages the model to avoid putting its decision boundary through the dense areas of datapoints such that the perturbations are least likely to incur a large change to the outputs of f_θ.

Thus, the worse-case perturbation Ω_θ can serve as a regularizer on the model f_θ when it is trained by minimizing the conventional training errors $\mathcal{E}_\theta(\mathbf{x},\mathbf{y})$ (e.g., cross-entropy loss) on training examples $(\mathbf{x},\mathbf{y}) \sim \mathcal{T}$. Therefore, we have the following objective to train the deep network,

$$\min_\theta \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{T}} \mathcal{E}_\theta(\mathbf{x},\mathbf{y}) + \gamma\Omega_\theta$$

where γ is a balancing coefficient trading off between the training errors and the worse-case perturbation regularizer.

For Ω_θ, it can involve both labeled and unlabeled data, and thus it could explore unlabeled examples in a semi-supervised fashion to train the network.

In the next two sections, we will discuss two forms of perturbations for the WCP model.

4. Additive Perturbation

In this section, we will discuss the first form of perturbation – the additive perturbation.

It imposes an additive noise on the model parameters θ, that is $g(\theta) = \theta + \delta$, along with a constraint on the norm of the noise $\mathcal{G} = \{\delta | \|\delta\| \leq \epsilon\}$. In this case, the WCP regularizer (1) becomes

$$\Omega_\theta^{\text{add}} = \max_{\|\delta\|\leq\epsilon} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right).$$

Taking the Taylor expansion of $\ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right)$ at $\delta = 0$, we have an approximate

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right) \approx \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \frac{1}{2}\delta^T S_\theta \delta \qquad (2)$$

where

$$S_\theta \triangleq \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \nabla^2 \ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right)|_{\delta=0}$$

is the second-order Hessian matrix at $\delta = 0$.

We use the following two facts in the above expansion.

- $\ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right)$ becomes 0 at $\delta = 0$, since $f_\theta(\mathbf{x})$ and $f_{\theta+\delta}(\mathbf{x})$ are equal for $\delta = 0$;
- $\ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right)$ attains its minimal value of 0 at $\delta = 0$, and thus the first-order term vanishes as the gradient becomes zero at this stationary point.

Then, the WCP regularizer can be solved by

$$\max_{\|\delta\|\leq\epsilon} \delta^T S_\theta \delta,$$

where the optimal δ^* attains at $\epsilon\mathbf{u}_\theta$ with $\mathbf{u}_\theta$ being the singular vector corresponding to the largest singular value of S_θ. By plugging δ^* into Ω_θ, we have the following regularizer of additive perturbation

$$\Omega_\theta^{\text{add}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \ell\left(f_\theta(\mathbf{x}), f_{\theta+\epsilon\mathbf{u}_\theta}(\mathbf{x})\right) \tag{3}$$

It is worth noting that the singular vector $\mathbf{u}_\theta$ can be computed efficiently by power iteration and the finite difference method [5]. In practice, we found even a single-step power iteration is enough in our experiments. This boils down to approximate $\mathbf{u}_\theta$ by evaluating the gradient of $\ell\left(f_\theta(\mathbf{x}), f_{\theta+\delta}(\mathbf{x})\right)$ near $\delta = 0$. This could significantly reduce the computational cost compared with naively solving a Singular Value Decomposition problem.

4.1. A Sigmoid Example

Here, we use a toy example to show the insight into how the WCP regularizes the training of deep networks.

Consider a sigmoid unit

$$f_\mathbf{w}(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})} \in [0,1],$$

which is the most basic building blocks in neural networks, with an input vector $\mathbf{x} = [x_1, x_2]^T$. In Figure 1(a), we consider four samples on the 2-D input space, and focus on a family of unit-norm parameters $\mathbf{w} \triangleq [\cos\phi, \sin\phi]^T$ with ϕ as the angle between $\mathbf{w}$ and the x_1-axis. It is not hard to see that the boundary $f_\mathbf{w}(\mathbf{x}) = \frac{1}{2}$ is given by $\mathbf{w}^T\mathbf{x} = 0$.

Without any data labels, it is intuitive to see that the most preferred $f_\mathbf{w}$ is given by $\phi = 0$, i.e., the boundary $x_1 = 0$, as it has the largest margin to separate datapoints. In other words, this boundary resides in a lowest-density area far apart from any datapoints.

This intuitive result exactly coincides with the one derived by minimizing the WCP regularizer (3) with a l_2 distance for the loss function ℓ. To show it, we plot the relation between the angle ϕ and the corresponding value of the WCP regularizer in Figure 1(b). The result shows the minimum WCP occurs when $\phi = 0$, which is consistent with our intuition.

We also observe there are two local minima of the WCP regularizer at $\pm\frac{\pi}{2}$, corresponding to the boundary $x_2 = 0$. This is not surprising as they have a locally large margin separating datapoints.

This example reveals an interesting relation between the minimizing the WCP with the additive perturbation and the large margin principle in the context of a sigmoid unit with linear boundary.

5. DropConnect Perturbation

The second perturbation under consideration is the DropConnect perturbation, which would change the network structure by dropping its connections. Specifically, for every parameter θ_i in θ, we define an indicator variable α_i in the vector α denoting if the corresponding connection should be dropped from the network by setting the weight to zero: $\alpha_i = 1$ denotes a dropped connection while $\alpha_i = 0$ indicates an intact one.

In this way, the perturbation function can be written as

$$g(\theta) = (1-\alpha)\circ\theta$$

with element-wise product $\circ$, and the constraint on α is

$$\mathcal{G}_\alpha = \{\alpha | \alpha \in \{0,1\}^N, \|\alpha\|_0 = \lfloor\sigma N\rfloor\},$$

where $\|\cdot\|_0$ is the ℓ_0 norm, N is the number of network weights in θ and $\sigma \in [0,1]$ is a preset dropconnect ratio, i.e., the portion of weight connections to be dropped.

By applying Taylor expansion again, we have

$$\begin{aligned}\alpha^* &= \mathop{\arg\max}_{\alpha\in\mathcal{G}_\alpha} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \ell\left(f_\theta(\mathbf{x}), f_{(1-\alpha)\circ\theta}(\mathbf{x})\right)\\ &\approx \mathop{\arg\max}_{\alpha\in\mathcal{G}_\alpha} \frac{1}{2}\alpha^T Q \alpha\end{aligned} \tag{4}$$

where

$$Q = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \nabla^2 \ell\left(f_\theta(\mathbf{x}), f_{(1-\alpha)\circ\theta}(\mathbf{x})\right)|_{\alpha=0}$$

is the Hessian matrix of the loss ℓ at $\alpha = 0$, which is a $N \times N$ semi-positive definite matrix.

It is obvious that (4) is a typical Binary Quadratic Programming (BQP) problem, which is NP-hard but admits an approximate solution to α^*. For example, it can be solved by spectral method or by converting into a semidefinite programming problem. Here, we choose an alternative spectral subgradient method [12]. While (4) contains a constraint on the number of nonzero elements in α, we will show how to solve a constrained BQP by the spectral subgradient method.

Once such α^* is solved, we can obtain the following WCP regularizer for the DropConnect perturbation,

$$\Omega_\theta^{\text{drop}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \ell(f_\theta(\mathbf{x}), f_{(1-\alpha^*)\circ\theta}(\mathbf{x}))$$

By the convention, we apply the dropconnect perturbation layer-wise in a deep network instead of applying it to an entire network as a whole, i.e., the set of dropped weight connections at various layers are sought individually. This can make the dropconnect WCP more computationally efficient as well as prevent too many connections from being dropped at few layers.

5.1. Spectral Gradient for Constrained BQP

In this section, we will present an approximate solution to the BQP problem in (4) with a linear constraint. First, let us define $\beta_i = 2\alpha_i - 1$ for $i = 1, \cdots, N$, which converts $\{0, 1\}$-constraint on α_i into $\{\pm 1\}$-constraint on β_i.

Putting all β_i together, we define an augmented $(N + 1)$-dim vector $\bar{\beta} = [\beta_i]_{i=1}^{N+1}$ by introducing an additional variable $\beta_{N+1} = 1$. Then the constraint $\mathcal{G}_\alpha$ on α becomes

$$\mathcal{G}_{\bar{\beta}} = \{\bar{\beta} | \bar{\beta} \in \{\pm 1\}^{N+1}, e^T\bar{\beta} = c\},$$ [1]

where $c = 2\lfloor\sigma N\rfloor - N + 1$, and $e \in \mathbb{R}^{N+1}$ is an all-one vector. Then the BQP can be reformulated in terms of $\bar{\beta}$, where the binary constraint on $\bar{\beta}_i$ can be rewritten as a quadratic constraint $\bar{\beta}_i^2 = 1$.

To solve the constrained BQP, we can introduce a Lagrange multiplier μ_i for each binary constraint $\bar{\beta}_i^2 = 1$, and μ_0 for the linear constraint $e^T\bar{\beta} = c$.

Then, the dual problem for the BQP can be written as

$$\min_{\mu,\mu_0} h(\mu, \mu_0) \tag{5}$$

with

$$\begin{aligned} &h(\mu, \mu_0) \\ &= \max_{\|\bar{\beta}\|^2=N+1} \bar{\beta}^T[L + \text{diag}(\mu)]\bar{\beta} - e^T\mu - c\mu_0 \\ &= (N + 1)\lambda_{\max}(L + \text{diag}(\mu)) - e^T\mu - c\mu_0 \end{aligned} \tag{6}$$

where

$$L = \left(\begin{array}{cc} Q & Qe + \frac{1}{2}\mu_0 e \\ e^TQ + \frac{1}{2}\mu_0 e^T & 0 \end{array} \right) \in \mathbb{R}^{N+1\times N+1},$$

and $\lambda_{\max}$ and $\mathbf{u}_{\max}$ denote the largest eigenvalue of $L + \text{diag}(\mu)$ and its corresponding eigenvector of unit norm. Like in the additive perturbation, $\mathbf{u}_{\max}$ can be efficiently approximated by using a single-step power iteration without naively solving the eigenvalue decomposition problem.

In Eq. (6), the maximum $\bar{\beta}^*$ is attained at

$$\bar{\beta}^* = \sqrt{N+1}\mathbf{u}_{\max} \tag{7}$$

The dual problem (5) can be solved by the gradient descent method over iterations. It is not hard [13] to show its gradient wrt μ and μ_0 is

$$\nabla_\mu h = (N + 1)\mathbf{u}_{\max}^2 - e$$

and

$$\frac{\partial h}{\partial \mu_0} = \frac{1}{2}(N + 1)\mathbf{u}_{\max}^T \left(\begin{array}{cc} 0 & e \\ e & 0 \end{array} \right) \mathbf{u}_{\max} - c$$

where $\mathbf{u}_{\max}^2$ denotes an element-wise square of $\mathbf{u}_{\max}$.

During training the WCP model with the dropconnect perturbation, over each mini-batch, we compute the above gradient to make an one-step update of the Lagrange multipliers μ and μ_0 along the descending direction, before the maximum $\bar{\beta}^*$ is taken with the updated multipliers. Finally, note that both $\pm\bar{\beta}^*$ are optimal for (6) and we should choose the one closer to $\beta_{N+1} = 1$ as required.

[1] Indeed, we instead impose an equivalent quadratic constraint $e^T\bar{\beta}\beta_{N+1} = c$ since $\beta_{N+1} = 1$.

Table 1: Error rate on CIFAR-10 over ten runs with different number of labeled examples. All methods use the same 13-layer architecture.

	1000 labels	2000 labels	4000 labels
GAN [24]			18.63± 2.32
Π model [9]			12.36± 0.31
Temporal Ensembling [9]			12.16± 0.31
VAT [10]			11.36
VAT+EntMin [10]			10.55
Supervised-only [29]	46.43±1.21	33.94±0.73	20.66±0.57
Π model [29]	27.36±1.20	18.02±0.60	13.20±0.27
Mean Teacher [29]	21.55±1.48	15.73±0.31	12.31±0.28
The proposed WCP	**17.62±1.52**	**11.93±0.39**	**9.72±0.31**

Table 2: Error rate on SVHN over ten runs with different number of labeled examples. All methods use the same 13-layer architecture.

	250 labels	500 labels	1000 labels
GAN [24]		18.44±4.8	8.11± 11.3
Π model [9]		6.65±0.53	4.82± 0.17
Temporal Ensembling [9]		5.12±0.13	4.42± 0.16
VAT [10]			5.42
VAT+EntMin [10]			3.86
Supervised-only [29]	27.77±3.18	16.88±1.30	12.32±0.95
Π model [29]	9.69±0.92	6.83±0.66	4.95±0.26
Mean Teacher [29]	4.35±0.50	4.18±0.27	3.95±0.19
The proposed WCP	**4.29±0.10**	**3.75±0.11**	**3.58±0.186**

6. Integrating Additive and DropConnect Perturbations

Additive and DropConnect perturbations could be integrated to train a semi-supervised classifier jointly. Consider a model with network weights θ. After an optimal additive perturbation δ^* and a dropconnect perturbation α^* are sought, the perturbed model weights $g(\theta)$ become $(1-\alpha^*)\circ(\theta+\delta^*)$.

Then the WCP regularizer integrating both worst-case perturbations can be written as

$$\Omega_\theta = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \ell\left(f_\theta(\mathbf{x}), f_{(1-\alpha^*)\circ(\theta+\delta^*)}(\mathbf{x})\right)$$

over both labeled and unlabeled data. This could be combined with the conventional classification loss (e.g., cross-entropy loss) to train a semi-supervised model. In each minibatch, both perturbations δ^* and α^* are updated iteratively to find the most vulnerable part of model weights and improve their robustness by minimizing the resultant regularizer.

In experiments, we found the best performances can be obtained by jointly imposing additive perturbations on the parameters of an entire model, while adding dropconnect perturbations only to some layers separately. We argue that the dropconnect perturbations could not be jointly applied to all the layers of a network because they could block the network connectivity by over-dropping the connections of some layers.

7. Experiments

In this section, we will conduct experiments to evaluate the performance of the semi-supervised classifiers based on the proposed WCP regularizer on both

Table 3: Ablation study of the impact of different model components. The error rate is reported on the test set of CIFAR-10 with $4,000$ labels.

Additive Perturbation	✓	✓	✓
DropConnect Perturbation		✓	✓
Entropy Minimization (EntMin)			✓
Error rate	10.15	9.85	**9.51**

Table 5: Error rate of the WCP with different dropconnect ratios on CIFAR-10 with $4,000$ labels, with the other hyperparameters fixed.

Dropconnect ratio	0.1	0.2	0.3	0.4	0.5	0.7
Error rate	9.81	**9.51**	9.66	9.78	9.92	10.26

Table 4: Error rate of worst-case dropconnect perturbations on different layers of each convolutional block on CIFAR-10 with $4,000$ labels.

DropConnect	Error rate
1st layers	9.77
2nd layers	**9.51**
3rd layers	10.08

CIFAR-10 and SVHN datasets.

7.1. Architecture and Implementation Details

For the sake of fair comparison, we adopt the same 13-layer architecture that has been used in the existing state-of-the-art models [9, 29, 10]. It consists of three blocks, and each block has three convolutional layers, followed by a 2×2 maxpooling and a dropout layer. The output feature map is globally averaged to a 128-dimensional vector after the third block, and a fully-connected layer follows to map the resultant vector to ten output classes with a softmax operation.

The additive perturbation is added to the network from the input layer of samples through the whole network with a magnitude ϵ of 8.0 and 3.5 on CIFAR-10 [7] and SVHN [11] datasets, respectively. The dropconnect perturbation is applied to the second layer of each convolutional block with a dropconnect ratio of 0.2 on both datasets. The cross-entropy loss and the WCP regularizer is combined with a fixed balancing coefficient $\gamma = 1.0$. The Kullback-Leibler divergence is adopted as the loss function ℓ in both perturbations. To ensure a fair comparison with the state-of-the-art virtual adversarial training model [10], Entropy Minimization (EntMin) is also adopted. Adam optimizer is used to train the network with an initial learning rate of 0.001 and $\beta_1 = 0.9$. The network is trained for a total of $1,000$ $(5,00)$ epochs on CIFAR-10 (SVHN). After the first 800 (400) epochs, the learning rate is scheduled to linearly decay to zero while β_1 being fixed to 0.5 on CIFAR-10 (SVHN). The hyperparameters are chosen based on the performance on a validation set with 20% labeled examples from the training set. Then the network is retrained with the selected hyperparameters on the whole training set, and the performance is reported on a separate test set.

We adopt the standard way to augment input images in literature [9, 29, 10]. They include both horizontal flips and random translations on CIFAR-10 images, with only random translations on the digits of the SVHN dataset.

7.2. Results

Table 1 and 2 compare the error rates of different methods on CIFAR-10 and SVHN dataset, respectively. Both the mean and deviation of the error rates are reported over ten runs with varying numbers of labeled examples. The comparisons show that the proposed WCP model outperforms the existing state-of-the-art semi-supervised models, including Mean Teacher [29], Virtual Adversarial Training [10], Temporal Ensembling [9], and Π model [29]. The results were achieved by integrating both additive and dropconnect perturbations. The following ablation study will analyze the effect of individual perturbations.

7.3. Ablation Study and Analysis

We conduct an ablation study of individual perturbations to evaluate their impacts on the performance. Table 3 reports the results on CIFAR-10 with $4,000$ labels. We evaluate on the impact of additive perturbation, and dropconnect perturbation, and entropy minimization on the model performance. The results show that all of them contribute to the reduction in the error rates. We also note that even if the entropy minimization were removed, the WCP would still outperform the compared algorithms including VAT and Mean Teacher. With the entropy minimization added, the WCP also outperforms the best performing VAT+EntMin that uses the entropy minimization as well.

Moreover, we evaluate the impact of where to impose the dropconnect perturbation in each convolutional block on the performance of the WCP model. Table 4 compares the error rates when the dropconnect perturbation is applied to different layers of each block. It shows that the smallest error rate is achieved when the dropconnect perturbation is added to the middle layer of each block.

Finally, Table 5 shows the results when different ratios are used for the dropconnect perturbation. The smallest error rate is achieved at $\sigma = 0.2$. Although the error rate changes slightly with varying ratios, the results show that the model performance is quite stable without too large fluctuation.

8. Conclusion

In this paper, we present two forms of model perturbations to train a robust classifier in a semi-supervised fashion. It assumes that a robust model should make stable predictions even if its weights and structures are worst perturbed to a certain degree of magnitude. To this end, the additive and dropconnect perturbations are developed. Given a magnitude of additive noise and dropconnect ratio, the worst-case perturbations are derived and applied to the model. Then the network is trained by minimizing the change of model predictions subject to these perturbations. Experiments demonstrate the proposed WCP-regularized classifier outperforms the state-of-the-art semi-supervised methods on both CIFAR-10 and SVHN datasets.

Acknowledgement

The work was done while L. Zhang visited the Seattle Cloud Lab of Futurewei Technologies. G.-J. Qi conceived the idea and prepared the manuscript, while L. Zhang performed the experiments.

References

[1] John Blitzer and Xiaojin Zhu. Semi-supervised learning for natural language processing. In *Tutorial Abstracts of ACL-08: HLT*, 2008. 2

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 1, 2

[3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 1, 2

[4] Marzieh Edraki and Guo-Jun Qi. Generalized loss-sensitive adversarial learning with manifold margins. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018. 2

[5] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2012. 4

[6] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019. 2

[7] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 7

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[9] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 1, 2, 6, 7

[10] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2, 6, 7

[11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 7

[12] Carl Olsson, Anders Eriksson, and Fredrik Kahl. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. 2007. 5

[13] Kaare Brandt Petersen et al. The matrix cookbook. 5

[14] Nitin Namdeo Pise and Parag Kulkarni. A survey of semi-supervised learning methods. In *2008 International Conference on Computational Intelligence and Security*, volume 2, pages 30–34. IEEE, 2008. 2

[15] Guo-Jun Qi. Learning generalized transformation equivariant representations via autoencoding transformations. *arXiv preprint arXiv:1906.08628*, 2019. 2

[16] Guo-Jun Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision*, pages 1–23, 2019. 2

[17] Guo-Jun Qi, Charu C Aggarwal, and Thomas Huang. Link prediction across networks by biased cross-network sampling. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 793–804. IEEE, 2013. 2

[18] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Image classification with kernelized spatial-context. *IEEE Transactions on Multimedia*, 12(4):278–287, 2010. 1, 2

[19] Guo-Jun Qi, Wei Liu, Charu Aggarwal, and Thomas Huang. Joint intermodal and intramodal label transfers for extremely rare or unseen classes. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1360–1373, 2016. 2

[20] Guo-Jun Qi, Yan Song, Xian-Sheng Hua, Hong-Jiang Zhang, and Li-Rong Dai. Video annotation by active learning and cluster tuning. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 114–114. IEEE, 2006. 2

[21] Guo-Jun Qi, Qi Tian, and Thomas Huang. Locality-sensitive support vector machine by exploring local correlation and global regularization. In *CVPR 2011*, pages 841–848. IEEE, 2011. 1

[22] Guo-Jun Qi, Liheng Zhang, Chang Wen Chen, and Qi Tian. Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8130–8139, 2019. 2

[23] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1517–1525, 2018. 2

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 6

[25] Yan Song, Guo-Jun Qi, Xian-Sheng Hua, Li-Rong Dai, and Ren-Hua Wang. Video annotation by active learning and semi-supervised ensembling. In *2006 IEEE International Conference on Multimedia and Expo*, pages 933–936. IEEE, 2006. 2

[26] Jinhui Tang, Xian-Sheng Hua, Guo-Jun Qi, and Xiuqing Wu. Typicality ranking via semi-supervised multiple-instance learning. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 297–300, 2007. 2

[27] Jinhui Tang, Haojie Li, Guo-Jun Qi, and Tat-Seng Chua. Integrated graph-based semi-supervised multiple/single instance learning framework for image annotation. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 631–634, 2008. 2

[28] Jinhui Tang, Guo-Jun Qi, Meng Wang, and Xian-Sheng Hua. Video semantic analysis based on

structure-sensitive anisotropic manifold ranking. *Signal Processing*, 89(12):2313–2323, 2009. 2

[29] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 2, 6, 7

[30] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 1

[31] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013. 2

[32] Jiayu Wang, Wengang Zhou, Guo-Jun Qi, Zhongqian Fu, Qi Tian, and Houqiang Li. Transformation gan for unsupervised image synthesis and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[33] Meng Wang, Xian-Sheng Hua, Tao Mei, Richang Hong, Guojun Qi, Yan Song, and Li-Rong Dai. Semi-supervised kernel density estimation for video annotation. *Computer Vision and Image Understanding*, 113(3):384–396, 2009. 2

[34] Xiao Wang, Daisuke Kihara, Jiebo Luo, and Guo-Jun Qi. Enaet: Self-trained ensemble autoencoding transformations for semi-supervised learning. *arXiv preprint arXiv:1911.09265*, 2019. 2

[35] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2555, 2019. 2

[36] Xiaojin Zhu. Semi-supervised learning tutorial. In *International Conference on Machine Learning (ICML)*, pages 1–135, 2007. 2