IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 17, 2019, accepted August 26, 2019, date of publication August 29, 2019, date of current version September 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938200

Adaptive Regularization via Residual Smoothing in Deep Learning Optimization

JUNGHEE CHO[1], JUNSEOK KWON[2], (Member, IEEE), AND BYUNG-WOO HONG[2]
[1]Department of Mathematical Sciences, Seoul National University, Seoul 08826, South Korea
[2]Computer Science Department, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Byung-Woo Hong (hong@cau.ac.kr)

This work was supported in part by the National Research Foundation of Korea under Grant NRF-2017R1A2B4006023 and Grant NRF-2018R1A4A1059731.

ABSTRACT We present an adaptive regularization algorithm that can be effectively applied to the optimization problem in deep learning framework. Our regularization algorithm aims to take into account the fitness of data to the current state of model in the determination of regularity to achieve better generalization. The degree of regularization at each element in the target space of the neural network architecture is determined based on the residual at each optimization iteration in an adaptive way. Our adaptive regularization algorithm is designed to apply a diffusion process driven by the heat equation with spatially varying diffusivity depending on the probability density function following a certain distribution of residual. Our data-driven regularity is imposed by adaptively smoothing a simplified objective function in which the explicit regularization term is omitted in an alternating manner between the evaluation of residual and the determination of the degree of its regularity. The effectiveness of our algorithm is empirically demonstrated by the numerical experiments in the application of image classification problems, indicating that our algorithm outperforms other commonly used optimization algorithms in terms of generalization using popular deep learning models and benchmark datasets.

INDEX TERMS Adaptive regularization, deep learning optimization, residual smoothing.

I. INTRODUCTION

Deep neural networks have made a significant progress in a variety of applications at a number of domains such as image understanding [1], sound recognition [2], motion planning [3], and other decision support [4]. In particular, the successful application of convolutional neural networks (CNNs) [5] to the computer vision problems has driven advanced performance in a variety of applications such as recognition [6], segmentation [7], motion estimation [8] or reconstruction [9] due to their effective characteristic power and generalization capabilities, leading to large scale optimization problems where the numbers of both model parameters and training data are often huge. The optimization in the deep learning applications often involves the stochastic estimation of gradients using the stochastic gradient descent in order to improve the computational efficiency with a large number of training data. Albeit the choices of mini-batch size and learning rate are implicitly related to the generalization of the model [10], it is generally required to introduce an explicit regularization term in the objective function to avoid over-parameterization or over-fitting. The objective function mainly consists of a data fidelity term that measures a discrepancy between estimation and observation and a regularization term that imposes smoothness constraint in the solution space, and their relative significance is usually determined by a constant based on the ratio of variances between likelihood and prior distributions. However, the computation of those distributions is computationally intractable leading to the grid search approach in determining the control parameter between the data fidelity and the regularization. In addition, the choice of static control parameter implies that the underlying likelihood and prior probabilities follow single model distributions, which is often undesirable to represent complicated models.

In this work, we propose a simple, yet effective regularization scheme that is designed to impose adaptive regularity depending on both spatial and temporal domain of optimization. We consider residual that is indicative of fitness between data and the current state of model in the determination of

The associate editor coordinating the review of this article and approving it for publication was Nishant Unnikrishnan.

regularization in such a way that the adaptive application of regularization is achieved in both space and time for better generalization.. We develop an implicit regularization scheme based on a simplified objective function where the regularization term is omitted and a diffusion process is applied to the data fidelity term. The diffusivity of diffusion process driven by heat equation is determined based on a probability density function following a certain distribution of residual at each residual element in the course of optimization. In the application of our approach to the deep learning algorithm, we present a neural network architecture incorporating our adaptive regularization, which is efficiently implemented by an additional smoothing layer with a deterministic smoothing kernels. We present the effectiveness of our proposed algorithm for generalization of model in the application of image classification problems with popular network models and commonly used benchmark datasets while our algorithm can be naturally integrated with other architectures of networks such as autoencoder for image segmentation or motion estimation.

In the remainder of this paper, we relate our method to the prior works in Sec. II and present the conventional optimization algorithm in Sec. III followed by our proposed algorithm in Sec. IV. The implementation of our adaptive regularization algorithm in the deep neural network framework is provided in Sec. V and the results of numerical experiments are presented in Sec. VI and the conclusion follows in Sec. VII.

II. RELATED WORK

There have been a variety of regularization techniques in machine learning applications. One can categorize the techniques into two classes, namely, explicit regularization and implicit regularization. We provide a number of algorithms for the explicit regularization and the implicit regularization in Sec. II-A and Sec. II-B, respectively. Then, we discuss in more detail the closely related works to our algorithm in Sec. II-C where the smoothing technique is considered to impose regularity on the solution space.

A. EXPLICIT REGULARIZATION

1) WEIGHT DECAY

The objective function is assumed to include a regularization term that penalizes a perturbation of unknown parameters in terms of L_2^2 norm. The gradient descent of the regularization term yields the decay of weights in a recursively manner with a given rate parameter and a learning rate. It is considered as one of the most practical regularization algorithms due to its computational convenience, yet often blur the solution.

2) SPARSITY CONSTRAINT

Sparsity has emerged as a way to impose L_1 regularization to objective functions. The essential motivation of the sparsity assumption on the solution space stems from the modeling of the residual distribution with a sharp peak, which is known to be more realistic in most real-world problems. Sparsity constraints suppress undesirable perturbations while preserving discontinuities in order to avoid over-fitting.

3) ENTROPY MINIMIZATION

In the application of sparsity constraint to the probability distribution of solution, the entropy term in the objective function has been introduced in [11] where the entropy is to be minimized. Entropy minimization has been shown to improve exploration ability, thus can regularize the objective functions in reinforcement learning tasks [12].

In contrast to the above explicit regularization techniques, our algorithm bases on the objective function that omits the regularization term instead applying simple, yet effective diffusion process to the data fidelity term.

B. IMPLICIT REGULARIZATION

1) NOISE INJECTION

In the estimation of gradients using the stochastic gradient descent, stochastic noise is involved and its variance is related to the size of mini-batch. The injection of stochastic noise to the neural network can be used as a way to impose regularization to arrive at a better local minimum [13]. It is also shown that the variance of injected noise is related to the amount of imposed smoothness on the solution [14] where a tighter lower bounds of the objective function can be achieved by adding noises in a stochastic gradient descent iteration. In addition to the manipulation of noise, smoothing of ground truth label has been proposed to make the model less confident regarding its trained weights, thus improving generality [15] where the probability of each label is arbitrarily perturbed depending on a random distribution. Similarly, there has been a regularization algorithm that replaces one of the ground truth labels with an arbitrary label uniformly at random [16].

2) DROPOUT

One of the implicit implementations of sparsity constraints that suppress the value of weights to be zero is Dropout [17] that randomly eliminates units of the neural network with a uniform probability while training, thus prevents units from excessive co-adapting. There have been a variety of Dropout techniques including maxout network [18] that proposes a new activation function to leverage the dropout, stochastic pooling [19] that replaces the deterministic pooling with a stochastic procedure by randomly choosing activation from a multinomial distribution.

3) LEARNING RATE DECAY

The stochastic gradient descent often yields better training results with a learning rate annealing scheme that schedules a temporal series of learning rates in epoch where a decreasing scheduling is generally applied to improve convergence. Whereas, a decreasing annealing pattern has been repeated as a warm start to overcome undesirable sharp local minima in [20].

4) MODEL ENSEMBLE

There has been a regularization technique developed by combining differently trained neural networks and introducing regularization effects imposed by different network architectures [21]. The random ensemble of prediction functions is known to provide better training behavior [22] and the structural dropout, called Branchout [23], has been developed by randomly choosing a subset of branches in the convolutional neural networks.

5) BATCH NORMALIZATION

Batch normalization [24] has been proposed for resolving the internal covariant shift by normalizing layer inputs, in which the distribution of inputs of each layer changes during the training process. However, batch normalization is proven that it can also improve the regularization performance in training neural networks [25]. In addition, batch normalization enabled training with larger learning rates, which induces faster convergence and better generalization [26].

While the aforementioned implicit methods mainly impose global regularity on the solution space, our method uses the residual that is variable in energy space and optimization time, thus spatially and temporally varying regularization depending on the residual.

C. REGULARIZATION VIA ENERGY SMOOTHING

There is different perspective of imposing regularity that the geometric property of energy landscape is modified in such a way that undesirable insignificant local minima are eliminated by smoothing the energy [27] where the objective function is convolved with Gaussian kernels. The approximated solution to the specific evolutionary partial differential equation (PDE) leads to convex envelopes of the objective function, but the approximation is assumed to be a solution of the PDE with small perturbations, which is often not the case. A modified network has been proposed in [28] where the loss function is differentiable, smooth and computationally stable.

Unlike the conventional smoothing approaches for regularization in deep learning optimization, our algorithm considers diffusion of residual with spatially and temporally varying diffusivity leading to adaptive regularization that is more suited for complex models in a variety of deep learning applications.

III. PRELIMINARY

We consider a minimization problem in a supervised learning framework. Let $\chi = \{(x_i, y_i)\}_{i=1}^n$ be a set of training data where $x_i \in X \subset \mathbb{R}^N$ is the i-th input and $y_i \in Y \subset \mathbb{R}^M$ is its desired output. Let $h_w \colon X \rightarrow Y$ be a prediction function that is associated with its model parameters $w = (w_1, w_2, \cdots, w_m) \in \mathbb{R}^m$ where the dimension of the feature space is m. The objective of the supervised learning problem is to find optimal parameters w^* that are typically obtained by minimizing the empirical loss $\mathcal{L}(w)$ defined on the training data χ:

$$\mathcal{L}(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) + \lambda\, \gamma(w), \tag{1}$$

where we denote by $f_i(w)$ a data fidelity term for a pair of data (x_i, y_i) and by $\gamma(w)$ a regularization term, and $\lambda > 0$ is a control parameter for the balance between the two terms. The data fidelity $f_i(w)$ incurred by a set of parameters w with a sample (x_i, y_i) is designed to measure the discrepancy between the prediction $h_w(x_i)$ with input x_i and its desired output y_i. The regularization $\gamma(w)$ aims to impose smoothness condition on the prediction function $h_w(x_i)$, thus avoid overfitting of the model. The control parameter λ is determined based on the relation between the underlying distribution of data and the prior distribution of model.

We consider a first-order optimization algorithm to minimize the objective function that is assumed to be differentiable leading to the following gradient descent step at iteration t:

$$w^{t+1} := w^t - \eta^t \left(\frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w^t) + \lambda \nabla \gamma(w^t) \right), \tag{2}$$

where we denote by $\nabla f_i(w^t)$ gradient of f_i with respect to w at iteration t, and by η^t the learning rate. The computation of the above full gradient over the entire training data is often intractable due to a large number of data, which leads to the use of stochastic gradient that is computed using a subset uniformly selected at random from the training data. The iterative step of the stochastic gradient descent algorithm at iteration t reads:

$$w^{t+1} := w^t - \eta^t \left(\frac{1}{|\beta^t|} \sum_{i \in \beta^t} \nabla f_i(w^t) + \lambda \nabla \gamma(w^t) \right), \tag{3}$$

where β^t denotes a mini-batch that is the index set of a subset uniformly selected at random from the training data. The size of mini-batch $|\beta^t|$ is related to the variance of the gradient norms, and thus to the regularization of the model. The small size of mini-batch yields stochastic gradients with higher variance due to noise involved in the stochastic process leading to large regularization.

IV. REGULARIZATION VIA RESIDUAL SMOOTHING

The optimization of interest aims to minimize the objective function that consists of a data fidelity term, a regularization term, and a control parameter for their relative weight. The selection of control parameter is often critical to obtain a better solution and is determined by the ratio between the underlying distributions of the residual and the prior smoothness, both of which are mostly assumed to follow unimodal distributions. Thus, the control parameter is chosen to be constant, and it is generally required to apply a grid search over a range of parameters to choose optimal parameters. However, it is often ineffective to model the distribution of data fidelity and determine the ratio of its variance to the variance of prior distribution for a smooth solution based on a unimodal probability density function leading to a static control parameter for the trade-off between data fitting and smoothness. Thus, we propose an adaptive regularization scheme that considers residual in the determination of regularity at each point of the residual domain.

A. ADAPTIVE REGULARIZATION BASED ON RESIDUAL

The computation of empirical stochastic gradient involves the noise process following a certain distribution with zero mean, and its variance is inversely proportional to the size of mini-batch. In addition to the stability, the noise process is also related to the regularization, thus the size of mini-batch can be used in determining regularity in an implicit way. On the other hand, the control parameter λ in the objective function in (1) can be variable with a fixed mini-batch size for each sample (x_i, y_i), leading to the following modified objective function:

$$\tilde{\mathcal{L}}(w) = \frac{1}{n}\sum_{i=1}^{n} (f_i(w) + \lambda_i\, \gamma(w)), \tag{4}$$

where $\lambda_i \in \mathbb{R}$ denotes a weighting parameter for the regularization term and it is designed to be associated with each sample (x_i, y_i). We assume that the degree of regularity follows a distribution of the residual leading to the following data-driven regularity:

$$\lambda_i \propto 1 - \exp\left(-\frac{\|f_i(w)\|}{\nu}\right), \tag{5}$$

where ν is a parameter for the variance of the residual. The degree of regularity is designed to be proportional to the magnitude of residual for each sample. In addition to the adaptive application of regularity with respect to sample, we consider the temporal state of solution in the course of optimization leading to the update of model parameters based on the stochastic gradients incorporating data-driven regularity as follows:

$$w^{t+1} := w^t - \eta^t \frac{1}{|\beta^t|}\sum_{i\in\beta^t} \big(\nabla f_i(w^t) + \lambda_i^t \nabla\gamma(w^t)\big), \tag{6}$$

where λ_i^t is variable with respect to both optimization iteration t and sample index i. The intrinsic motivation of the temporally adaptive regularization stems from the limitation of the existing static scheme that imposes the same degree of regularity albeit the residual decays in the optimization steps. However, it is computationally expensive to construct the distribution from which the control parameter for regularization is determined while computing the gradients of both data fidelity and regularization terms. In addition to the computational efficiency, it is desired to consider the relative magnitude of residual in its spatial domain. Thus, we propose a simple, yet effective regularization scheme that is designed to impose adaptive regularity depending on both spatial and temporal domain of optimization, which is achieved by smoothing residual with spatially and temporally varying degree without explicit computation of gradient for the regularization term that is omitted in the objective function, as presented in the following section.

B. REGULARIZATION VIA DIFFUSION PROCESS THAT IS ADAPTIVE IN THE BOTH SPATIAL AND TEMPORAL DOMAINS

We propose a regularization algorithm which is developed based on smoothing the residual that measures the discrepancy between model and sample data without taking into account an explicit regularization term. We modify the objective function in Eq. (4) from which the regularization term is omitted and the original data fidelity term $f_i(w)$ is replaced with $g_i(w)$ as follows:

$$\bar{\mathcal{L}}(w) = \frac{1}{n}\sum_{i=1}^{n} g_i(w), \quad g_i(w) = \|u_i(w)\|_2^2, \tag{7}$$

where $g_i(w)$ is L_2^2 norm of the diffused residual $u_i(w)$ for each sample (x_i, y_i), and $u_i(w)$ is obtained by imitating the diffusion process using the heat equation as follows:

$$\begin{cases} \dfrac{\partial u_i(w;\tau)}{\partial \tau} = \kappa\, \Delta u_i(w;\tau), \\ u_i(w;0) = d_i(w), \end{cases} \tag{8}$$

where κ denotes a diffusion coefficient, Δ the Laplace operator, and τ an auxiliary variable for the diffusion time. When κ is constant, the solution of heat equation is given by the convolution of initial data with the Gaussian kernel and it implies that the value at every point of solution becomes gradually averaged from its surrounding data as the diffusion proceeds. The Neumann boundary condition is imposed and the initial condition $u_i(w; 0)$ is given by the residual defined by the magnitude of the discrepancy between the predication and the desired output as follows:

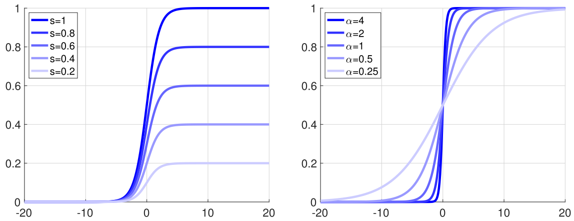
$$d_i(w) = |h_w(x_i) - y_i|, \tag{9}$$

where $d_i(w) \in \mathbb{R}^M$. In the diffusion equation, the coefficient κ is normally set to be constant, but we consider a diffusivity map $\kappa : \mathbb{R}^M \rightarrow \mathbb{R}^M$ that is employed to impose spatially varying regularity depending on the residual. The diffusivity map is designed to apply regularity following a distribution of residual based on the sigmoid function $S(x; s, \alpha)$ defined by:

$$S(x; s, \alpha) = \frac{s}{1 + \exp(-\alpha x)}, \tag{10}$$

where $s, \alpha \in \mathbb{R}^+$ are parameters that determine the vertical scale and the steepness of transition in function value, respectively.

The graphical illustration of the sigmoid function with varying parameters is presented in Fig. 1 where the functions with varying s and fixed $\alpha = 1$ are shown in (a), and the functions with varying α and fixed $s = 1$ are shown in (b). The update of parameters using the stochastic gradient



(a) varying s (fixed α) (b) varying α (fixed s)

FIGURE 1. Graphical illustration of sigmoid function with varying (a) scale parameter s with fixed α and (b) steepness parameter α with fixed s.

descent based on mini-batch β^t at each iteration t reads:

$$w^{t+1} := w^t - \eta^t \frac{1}{|\beta^t|} \sum_{i \in \beta^t} \nabla g_i(w^t), \tag{11}$$

where the computation of stochastic gradient $\nabla g_i(w^t)$ involves the diffusion $u_i(w^t)$ of residual $d_i(w^t)$. The diffusivity of the heat equation applied to the residual is determined based on a distribution formed by the sigmoid function and its associated parameters, scale s and steepness α, are chosen by global and local properties of residual in the neural network architecture as presented in the following section.

C. ANNEALING OF ADAPTIVE DIFFUSION

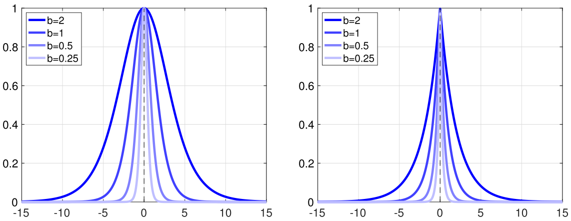
The proposed algorithm aims to impose adaptive regularization depending on the magnitude of residual by applying spatially varying diffusion to the residual. The diffusivity of the heat equation applied to the residual is determined based on the magnitude of residual following the sigmoid function as follows:

$$\kappa_i^t = S(d_i(w^t); s^t, \alpha) = \frac{s^t}{1 + \exp(-\alpha\, d_i(w^t))}, \tag{12}$$

where the diffusivity map $\kappa_i^t \in \mathbb{R}^M$ of the heat equation is determined by the sigmoid function of residual $d_i(w^t) \in \mathbb{R}^M$ at iteration t. We consider the temporal residual $d_i(w^t)$ given by the current state of solution w^t in determining the degree of temporal diffusion k_i^t for each sample (x_i, y_i). We also consider the scale parameter s^t that is variable in optimization time and present its annealing scheme in the following section.

1) GLOBAL ADAPTIVITY

The scale parameter $s^t \in \mathbb{R}^+$ of $S(d_t(w^t); s^t, \alpha)$ at time t in Eq. (12) determines the degree of diffusion that is applied to the entire domain of residual in an isotropic way with fixed steepness parameter $\alpha = 0$, thus it is global parameter that is dependent on time t. The motivation of introducing time-varying scale parameter is to consider the temporal decay of residual resulting in the decrease of diffusion that is equivalent to regularity. However, it is often necessary to allow larger stochastic noise in order to avoid undesirable sharp local minima in particular at the early stage of optimization. Thus, we propose to employ annealing schemes for the scale parameter s using the probability density functions of either the Logistic distribution $y_1(x; \mu, b)$ or the Laplace distribution $y_2(x; \mu, b)$ as defined by:



(a) Logistic distribution (b) Laplace distribution

FIGURE 2. Graphical illustration of the scaled probability density function associated with different distributions (a) Logistic distribution and (b) Laplace distribution with varying scale b for the annealing of the scale parameter s in sigmoid function. Each probability density function is scaled to have the maximum value 1.

$$y_1(x; \mu, b) = \frac{1}{4b} \operatorname{sech}^2\left(\frac{x-\mu}{2b}\right), \tag{13}$$

$$y_2(x; \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right), \tag{14}$$

where μ and b denote the mean and the scale, respectively. The graphical illustrations of the scaled probability density functions y_1 and y_2 with varying scale parameters b are presented in Fig. 2 where the maximum value of each probability density function is scaled to have the maximum value 1 and their associated distributions are (a) Logistic and (b) Laplace. The global difusivity map κ_i^t is then defined by the sigmoid function with s^t and fixed $\alpha = 0$ as defined by:

$$\kappa_i^t = S(d_i(w^t); s^t, \alpha), \quad \alpha = 0, \tag{15}$$

$$s^t = y(t; \mu, b), \tag{16}$$

where y can be either y_1 in Eq. (13) or y_2 in Eq. (14), and μ is chosen for the peak location and b is a scale parameter for the sharpness of the distribution centered at μ. The degree of regularization driven by the diffusion process based on the sigmoid function with the annealing for its scale parameter is gradually increasing up to the peak at the mean of the annealing distribution and decreasing afterwards arriving at the original objective function without diffusion.

2) LOCAL ADAPTIVITY

In the adaptive application of regularization in the domain of residual, we consider the relative magnitude of residuals so that different degree of regularization is applied to each residual element in its domain. The residual is initially normalized to have mean 0 and standard deviation 1 at each iteration in order to consider the relative significance among the residual elements. The diffusivity map with the local adaptive scheme is defined by:

$$\kappa_i^t = S(\tilde{d}_i^t; s, \alpha), \tag{17}$$

$$\tilde{d}_i^t = \frac{d_i(w^t) - \mu_i^t}{\sigma_i^t}, \tag{18}$$

where parameters $s, \alpha \in \mathbb{R}^+$ are chosen to be constant, and $\tilde{d}_i^t$ is the normalized residual of d_i^t with mean μ_i^t and standard deviation σ_i^t for each sample (x_i, y_i) at time t.

3) COMBINATION OF GLOBAL AND LOCAL ADAPTIVITY

Our final choice of the annealing scheme for adaptive regularization incorporates both global and local approaches considering the global decay of residual and its relative weight in the residual domain at each iteration, leading to the full adaptive scheme. The proposed diffusivity map for our algorithm integrates the global annealing of the scale parameter and the relative weight of residual leading to:

$$\kappa_i^t = S(\tilde{d}_i^t; s^t, \alpha), \tag{19}$$

$$\tilde{d}_i^t = \frac{d_i(w^t) - \mu_i^t}{\sigma_i^t}, \tag{20}$$

$$s^t = y(t; \mu, b), \tag{21}$$

where μ_i^t and σ_i^t denotes the mean and the standard deviation of temporal residual $d_i(w^t)$ for sample (x_i, y_i), respectively, and μ and b denotes the mean and the scale for the probability density function of the annealing distribution, respectively.

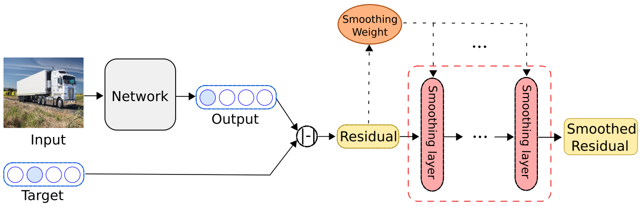


FIGURE 3. Schematic illustration of the network architecture incorporating our regularization algorithm for image classification problem. The target of the primary network is represented by an one-hot encoding and the residual is subsequently fed into a series of smoothing layers leading to the objective function based on the smoothed residual.

V. NETWORK ARCHITECTURE INCORPORATING ADAPTIVE REGULARITY

The neural network architecture with our proposed regularization algorithm is constructed by a primary network that yields an output of the prediction for the problem of interest and computes the associated residual that is subsequently fed into a series of smoothing layers leading to the objective function based on the smoothed residual. The schematic illustration of the network architecture is presented in Fig. 3 where the target of the primary network is represented by a one-hot encoding for the image classification problem. Our regularization algorithm applies a diffusion process to the residual depending on its magnitude using the heat equation based on the Laplace operator with spatially varying diffusivity, however the application of the Laplace operator is not suited for the residual domain, in which the spatial property among the neighboring elements is not locally related to the regularity of solution in the image classification problem while the Laplace operator is constrained to be applicable at the residual domain where the local affinity implies the regularity in the solution space, for example, autoencoder architectures. In the sequel, we employ an extended Laplace operator resulting in a global interpolation of all the elements in the residual domain to blur a one-hot encoding representation based on a fully connected layer with the following weights:

$$w_{jk} = \begin{cases} 1 - \kappa_j, & \text{if } k = j \\ \dfrac{\kappa_j}{M-1}, & \text{otherwise} \end{cases} \tag{22}$$

where $w_{jk} \in \mathbb{R}^{M \times M}$ denotes the filter element of a fully connected layer that connects from the k-th node of the residual to the j-th node of the successive diffused residual layer, M is the dimension of residual, and κ_j denotes the diffusivity value obtained by Eq. (19) for the j-th element of the residual. The number of smoothing layers is related to the diffusion time τ and the diffusivity κ in the heat equation in Eq. (8), and we set the number of smoothing layer to be one while the scale factor of the diffusivity varies for numerical stability and computational efficiency. The overall algorithm of our proposed method is presented in Algorithm 1.

VI. EXPERIMENTAL RESULTS

In the experiments, we empirically provide the quantitative evaluation of our algorithm by the comparative analysis using an image classification task. A detailed description of the experimental setup is presented in the following:

Datasets: We use four commonly used benchmark datasets including CIFAR-10, CIFAR-100 [29], Street View House Numbers (SVHN) [30], and Fashion-MNIST [31]. CIFAR-10 consists of 50K training and 10K testing images of the size 32 × 32 × 3 for 10 categories. CIFAR-100 is the same as CIFAR-10 except that it has 100 classes. We apply conventional image augmentation with padding, random cropping and flipping to CIFAR-10 and CIFAR-100 as pre-processing stesp. SVHN is a dataset of house numbers in the street images. It consists of 73257 training and 26032 testing images of the size 32 × 32 × 3 for 10 categories. Fashion-MNIST consists of 60K training and 10K testing images of the size 28 × 28 for 10 categories.

Neural Network Models: We consider neural network architectures ranging from shallow to deep models including ResNet20, ResNet56 [32] and DenseNet-BC with 100 layers ($k = 12$) [33].

Optimization and Hyperparameters: We use the stochastic gradient descent method and the objective function is the mean squared error of the residual that measures a difference between the prediction and the desired output. We use the following common hyperparameters across all the experiments; momentum is 0.9, mini-batch size is 128, number of epoch is 160 for CIFAR-10 and CIFAR-100, 100 for SVHN and 48 for Fashion-MNIST, learning rate is set to be 0.1 for the first 75 percent of epochs and 0.001 for the rest. For Fashion-MNIST dataset, we try several different values (1e−2, 1e−3, 1e−4, 1e−5) of weight decay for baseline experiments and the parameter 1e−3 which shows the best test accuracy among baselines is adopted for the proposed method. For other datasets, weight decay value 1e−4

Algorithm 1 Pseudocode for the Proposed Method

```
Initialize the weights $w$ of neural network $h_w$
Select annealing scheme $y$
Iteration $t = 0$
for $t = 0, 1, 2, \cdots$ do
    Sample a mini-batch $\beta^t$ from training dataset.
    Initialize loss $\mathcal{L}(w^t) = 0$
    for a $i$-th pair of data $(x_i, y_i)$ in $\beta^t$ do
        Get residual $d_i^t = |h_{w^t}(x_i) - y_i|$ and normalize it to get $\tilde{d}_i^t$.
        Get diffusivity $\kappa_i^t = S(\tilde{d}_i^t; s^t, \alpha)$ where $s^t = y(t; \mu, b)$
        Construct a fully connected layer $L$ with the weights $w_{jk}$ as in (22)
        Pass $d_i^t$ to the above fully connected layer $L$ and get the smoothed residual $r_i^t$
        $\mathcal{L}(w^t) \leftarrow \mathcal{L}(w^t) + \|r_i^t\|_2^2$
    end for
    $w^{t+1} \leftarrow w^t - \eta^t \frac{1}{|\beta^t|} \nabla \mathcal{L}(w^t)$ where $\eta^t$ is a learning rate at $t$
    $t \leftarrow t + 1$
end for
return Trained neural network $h_w$
```

TABLE 1. Validation accuracy based on ResNet20 using Fashion-MNIST dataset by SGD (left) with varying weight decay (wd) parameters from larger to smaller, our algorithm with global annealing scheme (middle) and the combination of global and local annealing scheme (right). The annealing of adaptive regularization parameter follows Laplace (left) and Logistic (right) distributions where the associated parameters are chosen by the grid search.

	SGD				Global		Global+Local	
wd acc	1e−2	1e−3	1e−4	1e−5	Laplace	Logistic	Laplace	Logistic
max	88.46	93.36	93.25	93.21	93.73	93.77	93.89	**93.91**
mean	88.04	93.30	93.19	93.16	93.65	93.64	**93.79**	93.67

is chosen for baseline and proposed method as recommended in [32], [33]. The unknown weights are initialized by the algorithm proposed in [34].

Quantitative Evaluation: We compute the learning curves that include training loss, training accuracy and validation accuracy. We perform 5 independent trials for each set of experiment and the maximum validation accuracy is taken across all the epochs and the average of the maximum is taken over 5 trials. We also compute the average validation accuracy over the last 10% of epochs and the average of the average is taken over 5 trials.

Computational Cost: The additional computational cost of our algorithm is $\mathcal{O}(M^2)$ for each training data where M is the dimension of residual. We use a single NVIDA GeForce GTX 1080 Ti GPU. When training CIFAR-10 dataset with ResNet56 architecture, the baseline (SGD) takes about one and half hours whereas our algorithm takes about two and half hours. For CIFAR-10 with DenseNet-BC, baseline takes about four hours whereas our algorithm takes about five hours. Although our algorithm is slower than SGD, it is affordable within a few hours.

A. ABLATION ANALYSIS ON THE ADAPTIVE REGULARIZATION PARAMETERS

We analyze the effect of global and its combination with local annealing schemes for the adaptive regularization based on ResNet20 using Fashion-MNIST dataset. We compare the performance of our algorithm to the baseline, stochastic gradient descent (SGD), to demonstrate that our algorithm outperforms SGD with grid search of regularization parameter. We apply SGD with varying weight decay values such as 1e−2, 1e−3, 1e−4, 1e−5 and the validation accuracy is presented in Table 1 where the results with our algorithm based on global adaptive annealing following Laplace and Logistic distributions are presented at the middle block, and the results based on the combination of global and local adaptive annealing following Laplace and Logistic distributions are presented at the right block. For the global adaptivity, the steepness parameter $\alpha = 0$ in Eq. (12) is used and the scale parameter b of distribution varies from 0.1 to 0.9 with step size 0.2 while the mean μ of distributions is set to be 75% point of epochs, the maximum of distributions is scaled to be 1. In the application of full adaptive schemes integrating global and local schemes, the same parameters as the global adaptive scheme are used except the steepness parameter $\alpha = 0.25, 0.5, 1, 2, 4$. We apply a grid search in the selection of parameters associated with our algorithms. It is shown that our algorithm outperforms SGD regardless of weight decay value associated with SGD, and the performance gain is achieved with the local adaptivity in addition to the global adaptivity.

B. EFFECT ON GENERALIZATION BASED ON PARTIAL TRAINING DATA

We empirically demonstrate the effect of our adaptive regularization algorithm based on ResNet20 using

TABLE 2. Validation accuracy based on ResNet20 using partial (1/2, 1/4 and 1/8) Fashion-MNIST dataset by SGD (left) with varying weight decay (wd) parameters from larger to smaller, our algorithm with the global+local adaptive scheme (right). The annealing of adaptive regularization parameter follows Laplace (left) and Logistic (right) distributions where the associated parameters are chosen by the grid search.

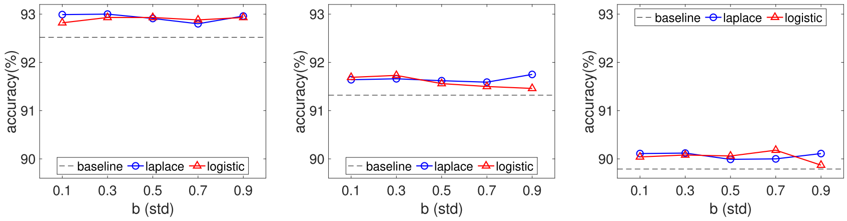
		SGD				Global+Local	
ratio	wd acc	1e−2	1e−3	1e−4	1e−5	Laplace	Logistic
1/2	max	88.84	92.52	91.76	91.92	**92.99**	92.93
	mean	88.55	92.45	91.73	91.87	**92.87**	92.80
1/4	max	88.46	91.32	90.55	90.58	**91.75**	91.73
	mean	88.27	91.25	90.52	90.53	**91.65**	91.58
1/8	max	88.14	89.79	89.00	88.62	90.12	**90.18**
	mean	88.02	89.72	88.94	88.55	90.01	**90.09**

Fashion-MNIST dataset. We select partial subset of training data uniformly at random for the training phase with varying ratio such as $1/2$, $1/4$ and $1/8$ in highlighting of the effectiveness of our algorithm in generalization. The validation accuracy of SGD is computed at a range of weight decay values, $1\mathrm{e}{-2}$, $1\mathrm{e}{-3}$, $1\mathrm{e}{-4}$, $1\mathrm{e}{-5}$, and its maximum and average are computed over 5 independent trials as shown at left block in Table 2. The maximum and average of validation accuracy obtained by our algorithm with fully adaptive regularization incorporating global and local schemes are presented at right block in Table 2 where Laplace and Logistic distributions are used for global annealing of adaptive regularization. The associated steepness parameter α with the sigmoid function for the local adaptive regularization is fixed as 1 whereas the associated scale parameter b with the distribution for the global adaptive regularization is selected by a grid search over a range of values from 0.1 to 0.9 with a step size 0.2 except for Laplace adaptivity scheme with $1/8$ partial data where grid search for b is done over a range from 0.9 to 1.7 with a step size 0.2.

The maximum validation accuracy obtained by our algorithm with different annealing distribution is presented in Figure 4 where the accuracy with Laplace and Logistic annealing distributions is shown in blue and red, respectively along with the baseline in black for each ratio of partial training set, (a) $1/2$, (b) $1/4$ and (c) $1/8$. It is shown that our algorithm outperforms the baseline across all the scale parameters for both annealing distributions, indicating that our algorithm achieves better generalization.

C. COMPARATIVE ANALYSIS WITH OTHER OPTIMIZATION ALGORITHMS

We compare our algorithm with previous smoothing method called label smoothing [15] and the commonly used optimization algorithms including Adam [35] and AdaGrad [36]. For label smoothing, we use the same hyperparameters as that of baseline and smoothing parameter $\epsilon = 0.1$. In our comparative analysis, we use deeper networks including ResNet56 and DenseNet-BC with 100 layers ($k = 12$) for the benchmark datasets that are CIFAR-10, CIFAR-100 and SVHN. The associated parameters with our algorithm are used by $b = 0.2, 0.5$ for Laplace, $b = 0.25, 0.5$ for Logistic distribution respectively and $\alpha = 1$. The maximum and mean validation accuracy are presented in Table 3 where

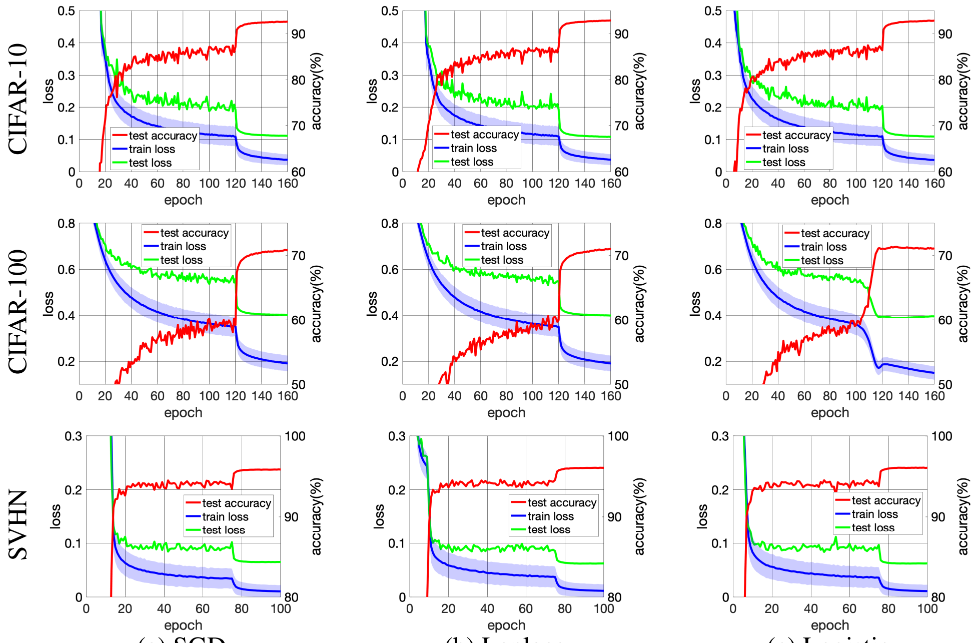


(a) 1/2 training data (b) 1/4 training data (c) 1/8 training data

FIGURE 4. Validation accuracy (y-axis) with varying scale parameter of distribution *b* (x-axis) associated with the annealing distribution using partial Fashion-MNIST dataset. The training is performed based on ResNet20 by SGD and our global+local adaptive regularization schemes. The partial ratios of training data used are (a) 1/2 (b) 1/4 and (c) 1/8.

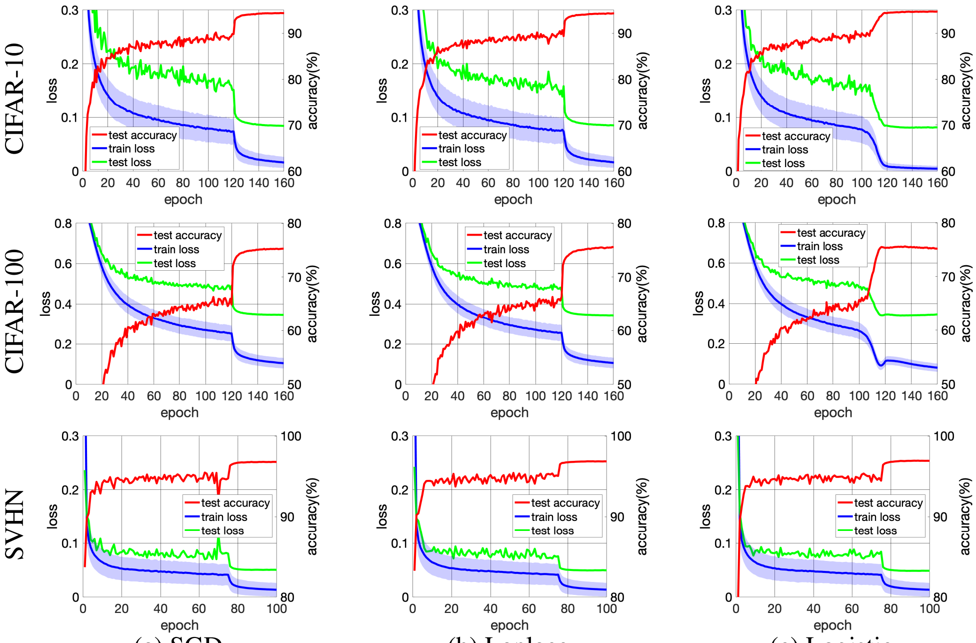
TABLE 3. Comparison of validation accuracy obtained by SGD, Adam, AdaGrad, Label smoothing (LS), our fully adaptive algorithm with Laplace and Logistic distributions from left to right. The training is performed based on the model ResNet56 (top block) and DenseNet-BC (bottom block). For each block of model, the dataset CIFAR-10 (top), CIFAR-100 (middle), SVHN (bottom) are used.

model	dataset	acc	SGD	Adam	AdaGrad	LS [15]	Laplace	Logistic
ResNet56	CIFAR-10	max	92.66	90.50	87.67	92.36	**92.93**	92.89
		mean	92.54	89.86	86.47	92.21	**92.78**	92.74
	CIFAR-100	max	70.94	68.42	66.55	71.09	71.06	**71.46**
		mean	70.71	68.21	66.41	70.86	70.89	**71.11**
	SVHN	max	95.88	95.82	94.70	95.99	96.09	**96.11**
		mean	95.83	95.37	94.36	95.95	96.05	**96.06**
DenseNet-BC	CIFAR-10	max	94.43	91.91	87.16	94.43	94.64	**94.76**
		mean	94.32	91.39	86.81	94.32	94.53	**94.66**
	CIFAR-100	max	75.41	67.37	58.66	75.30	75.67	**75.75**
		mean	75.22	66.75	58.44	75.12	**75.46**	75.31
	SVHN	max	96.82	96.11	95.10	96.86	96.89	**96.97**
		mean	96.78	95.67	94.78	96.83	96.85	**96.93**



(a) SGD (b) Laplace (c) Logistic

FIGURE 5. **Learning curves obtained based on ResNet56 model using CIFAR-10 (top), CIFAR-100 (middle), and SVHN (bottom) datasets. The validation accuracy, training loss, testing loss are presented in red, blue, green color, respectively. The learning performance of our regularization scheme based on (b) Laplace and (c) Logistic distributions is compared with (a) the SGD algorithm.**



(a) SGD (b) Laplace (c) Logistic

FIGURE 6. **Learning curves obtained based on DenseNet-BC ($k = 12, L = 100$) model using CIFAR-10 (top), CIFAR-100 (middle), and SVHN (bottom) datasets. The validation accuracy, training loss, testing loss are presented in red, blue, green color, respectively. The learning performance of our regularization scheme based on (b) Laplace and (c) Logistic distributions is compared with (a) the SGD algorithm.**

the results using ResNet56 (top block) and DenseNet-BC (bottom block) with SGD, Adam, AdaGrad, and our algorithms with Laplace and Logistic distributions are shown from left to right.

It is shown that our algorithm outperforms the other algorithms under comparison with their recommended parameters. Note that adaptive optimization methods such as Adam and AdaGrad often generalize worse than SGD for image classification task [37]. There is potential that our algorithm can be improved with wider range of grid search for the parameters α and b. The learning curves obtained by (a) SGD and our algorithms with (b) Laplace and (c) Logistic distributions using CIFAR-10 (top), CIFAR-100 (middle), SVHN (bottom) are presented in Figure 5 and Figure 6 where the learning curves with our algorithm indicate better generalization in terms of the validation accuracy. The learning rate is scheduled to be dropped at 75% of epochs from 0.1 to 0.001 and the global adaptive annealing reaches the peak of the associated distribution at 75% of epochs, which leads to an abrupt change in the learning curves.

VII. CONCLUSION AND DISCUSSION

In this paper, we have investigated the data-driven adaptive regularization by smoothing the residual of neural network for the image classification problem in an adaptive manner. The residual is defined by the discrepancy between the output of the neural network and the desired output. The regularization is imposed by diffusing the residual depending on the probability density function following either Laplace or Logistic distributions where the degree of regularization is proportional to the magnitude of each residual element. The combination of local and global annealing scheme that is designed to take into account residual in determining the degree of diffusion has been presented to spatially and temporally varying regularization. The effectiveness of the proposed algorithm has been demonstrated by the experimental results indicating the potential of our algorithm that can be easily integrated to a variety of problems in deep learning applications.

REFERENCES

[1] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2049–2062, Apr. 2018.

[2] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.

[3] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in RGB+D Videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1045–1058, May 2018.

[4] K. Ahmad and N. Conci, "How deep features have improved event recognition in multimedia: A survey," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 2, Jun. 2019, Art. no. 39.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, Feb. 2017, pp. 1–7.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[8] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. AAAI*, Feb. 2017, pp. 1–7.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[10] K. Levy, "Online to offline conversions, universality and adaptive minibatch sizes," in *Proc. NIPS*, 2017, pp. 1–10.

[11] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. NIPS*, 2004, pp. 1–8.

[12] J. Grau-Moya, F. Leibfried, and P. Vrancx, "Soft q-learning with mutual-information regularization," in *Proc. ICLR*, 2019, pp. 1–19.

[13] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. NIPS*, 2015, pp. 1–9.

[14] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Proc. NIPS*, 2017, pp. 1–10.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, Jun. 2016, pp. 2818–2826.

[16] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "Disturblabel: Regularizing CNN on the loss layer," in *Proc. CVPR*, Jun. 2016, pp. 4753–4762.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. ICML*, Feb. 2013, pp. 1–9.

[19] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," 2013, *arXiv:1301.3557*. [Online]. Available: https://arxiv.org/abs/1301.3557

[20] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. ICLR*, 2017, pp. 8–16.

[21] S. Singh, D. Hoiem, and D. Forsyth, "Swapout: Learning an ensemble of deep architectures," in *Proc. NIPS*, 2016, pp. 1–9.

[22] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, "Towards robust neural networks via random self-ensemble," in *Proc. ECCV*, Sep. 2018, pp. 369–385.

[23] B. Han, J. Sim, and H. Adam, "BranchOut: Regularization for online ensemble tracking with convolutional neural networks," in *Proc. CVPR*, Jul. 2017, pp. 521–530.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Mar. 2015, pp. 1–11.

[25] P. Luo, X. Wang, W. Shao, and Z. Peng, "Towards understanding regularization in batch normalization," in *Proc. ICLR*, 2019, pp. 1–23.

[26] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," in *Proc. NIPS*, 2018, pp. 7694–7705.

[27] H. Mobahi and J. W. Fisher, "On the link between Gaussian homotopy continuation and convex envelopes," in *Proc. CVPR Workshop*, 2015, pp. 43–56.

[28] C. Gulcehre, M. Moczulski, F. Visin, and Y. Bengio, "Mollifying networks," 2016, *arXiv:1608.04980*. [Online]. Available: https://arxiv.org/abs/1608.04980

[29] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009, vol. 1, no. 4.

[30] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS*, 2011.

[31] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: https://arxiv.org/abs/1708.07747

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, Jul. 2017, pp. 2261–2269.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imageNet classification," in *Proc. ICCV*, Dec. 2015, pp. 1026–1034.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[36] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[37] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4148–4158.

JUNGHEE CHO received the B.A. degree in mathematical sciences from Seoul National University, South Korea, in 2013, where he is currently pursuing the Ph.D. degree in mathematical sciences.

He is working in the field of image processing and deep learning. His research interests include image denoising, and optimization methods in deep learning and its application to computer vision.

JUNSEOK KWON received the B.S. degree, M.S. degree in object tracking (supervised by Prof. Kyoung Mu Lee), and the Ph.D. degree in electrical engineering and computer science from Seoul National University, South Korea, in 2006, 2008, and 2013, respectively.

He was a Postdoctoral Researcher with the Computer Vision Laboratory, ETH Zurich, from 2013 to 2014, under Prof. Luc Van Gool. He is currently an Assistant Professor with the School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea. He is working in the field of object tracking to capture the dynamics of cities. His research interests include visual tracking, visual surveillance, and Monte Carlo sampling method and its variants.

BYUNG-WOO HONG received the M.S. degree in computer vision from the Weizmann Institute of Science with Prof. Shimon Ullman and the Ph.D. degree in computer vision from the University of Oxford with Prof. Michael Brady, in 2001 and 2005, respectively.

He joined the Computer Science Department, Chung-Ang University, South Korea, as a faculty member, in 2008, after his postdoctoral research at the Computer Science Department, University of California Los Angeles, with Prof. Stefano Soatto. His research interests include image processing, computer vision, machine learning, and medical image analysis.