Continuous Dropout

Xu Shen, Xinmei Tian, *Member, IEEE*, Tongliang Liu, Fang Xu, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Dropout has been proven to be an effective algorithm for training robust deep networks because of its ability to prevent overfitting by avoiding the co-adaptation of feature detectors. Current explanations of dropout include bagging, naive Bayes, regularization, and sex in evolution. According to the activation patterns of neurons in the human brain, when faced with different situations, the firing rates of neurons are random and continuous, not binary as current dropout does. Inspired by this phenomenon, we extend the traditional binary dropout to continuous dropout. On the one hand, continuous dropout is considerably closer to the activation characteristics of neurons in the human brain than traditional binary dropout. On the other hand, we demonstrate that continuous dropout has the property of avoiding the co-adaptation of feature detectors, which suggests that we can extract more independent feature detectors for model averaging in the test stage. We introduce the proposed continuous dropout to a feedforward neural network and comprehensively compare it with binary dropout, adaptive dropout, and DropConnect on Modified National Institute of Standards and Technology, Canadian Institute for Advanced Research-10, Street View House Numbers, NORB, and ImageNet large scale visual recognition competition-12. Thorough experiments demonstrate that our method performs better in preventing the co-adaptation of feature detectors and improves test performance.**

***Index Terms*— Co-adaptation, deep learning, dropout, overfitting, regularization.**

I. Introduction

DROPOUT is an efficient algorithm introduced by Hinton *et al*. [1] for training robust neural networks and has been applied to many vision tasks [2]–[4]. During the training stage, hidden units of the neural networks are randomly omitted at a rate of 50% [1], [5]. Thus, the presentation of each training sample can be viewed as providing updates of parameters for a randomly chosen subnetwork. The weights of this subnetwork are trained by backpropagation [6]. Weights are shared for the hidden units that are present among different subnetworks at each iteration. During the test stage, predictions are made by the entire network, which contains all the hidden units with their weights halved.

The motivation and intuition behind dropout is to prevent overfitting by avoiding co-adaptations of the feature detectors [1]. Deep network can achieve better representation than shallow networks, but overfitting is a serious problem when training a large feedforward neural network on a small training set [1], [7]. Randomly dropping the units from the neural network can greatly reduce this overfitting problem. Encouraged by the success of dropout, several related works have been presented, including fast dropout [8], adaptive dropout [9], and DropConnect [10]. To accelerate dropout training, Wang and Manning [8] suggested sampling the output from an approximated distribution rather than sampling binary mask variables for the inputs. Ba and Frey [9] proposed adaptively learning the dropout probability p from the inputs and weights of the network. Wan *et al*. [10] generalized dropout by randomly dropping the weights rather than the units.

To interpret the success of dropout, several explanations from both theoretical and biological perspectives have been proposed. Based on theoretical explanations, dropout is viewed as an extreme form of bagging [1], as a generalization of naive Bayes [1], or as adaptive regularization [11], [12], which is proven to be a very useful approach for neural network training [13]. From the biological perspective, Hinton *et al*. [1] explain that there is an intriguing similarity between dropout and the theory of the role of sex in evolution. However, no understanding from the perspective of the brain's neural network—the origin of deep neural networks—has been proposed. In fact, by analyzing the firing patterns of neural networks in the human brain [14]–[16], we find that there is a strong analogy between dropout and the firing pattern of brain neurons. That is, a small minority of strong synapses and neurons provide a substantial portion of the activity in all brain states and situations [14]. This phenomenon explains why we need to randomly delete hidden units from the network and train different subnetworks for different samples (situations). However, the remainder of the brain is not silent. The remaining neuronal activity in any given time window is supplied by very large numbers of weak synapses and cells. The amplitudes of oscillations of neurons obey a random continuous pattern [15], [16]. In other words, the division between "strong" and "weak" neurons is not absolute. They obey a continuous—rather than

Manuscript received March 29, 2016; revised January 13, 2017 and May 31, 2017; accepted August 31, 2017. Date of publication October 3, 2017; date of current version August 20, 2018. This work was supported in part by the 973 Project under Grant 2015CB351803, in part by the National Key Research and Development Program of China under Grant 2017YFB1002203, in part by NSFC under Grant 61572451, Grant 61390514, and Grant 61632019, in part by Youth Innovation Promotion Association CAS under Grant CX2100060016, in part by Fok Ying Tung Education Foundation under Grant WF2100060004, and in part by the Australian Research Council Projects under Grant FL-170100117, Grant DP-140102164, and Grant LP-150100671. *(Corresponding author: Xinmei Tian.)*

X. Shen and X. Tian are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, University of Science and Technology of China, Hefei 230027, China (e-mail: shenxu@mail.ustc.edu.cn; xinmei@ustc.edu.cn).

T. Liu and D. Tao are with the UBTech Sydney Artificial Intelligence Institute, School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Sydney, NSW 2008, Australia (e-mail: tongliang.liu@uts.edu.au; dacheng.tao@uts.edu.au).

F. Xu is with the CAS Key Laboratory of Brain Function and Disease, School of Life Sciences, University of Science and Technology of China, Hefei 230027, China (e-mail: xufan@mail.ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2750679

bimodal—distribution [15]. Consequently, we should assign a continuous random mask to each neuron in the dropout network for the divisions of "strong" and "weak" rather than use a binary mask to choose "activated" and "silent" neurons.

Inspired by this phenomenon, we propose a continuous dropout algorithm in this paper, i.e., the dropout variables are subject to a continuous distribution rather than the discrete (Bernoulli) distribution in [1]. Specifically, in our continuous dropout, the units in the network are randomly multiplied by continuous dropout masks sampled from $\mu \sim U(0, 1)$ or $g \sim \mathcal{N}(0.5, \sigma^2)$, termed uniform dropout or Gaussian dropout, respectively. Although multiplicative Gaussian noise has been mentioned in [17], no theoretical analysis or generalized continuous dropout form is presented. We investigate two specific continuous distributions, i.e., uniform and Gaussian, which are commonly used and are also similar to the process of neuron activation in the brain. We conduct extensive theoretical analyses, including both static and dynamic property analyses of our continuous dropout, and demonstrate that continuous dropout prevents the co-adaptation of feature detectors in deep neural networks. In the static analysis, we find that continuous dropout achieves a good balance between the diversity and independence of subnetworks. In the dynamic analysis, we find that continuous dropout training is equivalent to a regularization of covariance between weights, inputs, and hidden units, which successfully prevents the co-adaptation of feature detectors in deep neural networks.

We evaluate our continuous dropout through extensive experiments on several data sets, including Modified National Institute of Standards and Technology (MNIST), Canadian Institute for Advanced Research-10 (CIFAR-10), Street View House Numbers (SVHN), NORB, and Imagenet Large Scale Visual Recognition Competition-12 (ILSVRC-12). We compare it with Bernoulli dropout, adaptive dropout, and DropConnect. The experimental results demonstrate that our continuous dropout performs better in preventing the co-adaptation of feature detectors and improves test performance.

II. Continuous Dropout

Hinton *et al.* [1] interpret dropout from the biological perspective, i.e., it has an intriguing similarity to the theory of the role of sex in evolution [18]. Sexual reproduction involves taking half the genes of each parent and combining them to produce offspring. This corresponds to the result where dropout training works the best when $p = 0.5$; more extreme probabilities produce worse results [1]. The criteria for natural selection may not be individual fitness but rather the mixability of genes to combine [1]. The ability of genes to work well with another random set of genes makes them more robust. The mixability theory described in [19] is that sex breaks up sets of co-adapted genes, and this means that achieving a function using a large set of co-adapted genes is not nearly as robust as achieving the same function, perhaps less than optimally, in multiple alternative ways, each of which only uses a small number of co-adapted genes.

Following this train of thought, we can infer that randomly dropping units tends to produce more multiple alternative networks, which is able to achieve better performance. For example, when we use one hidden layer with n units for dropout training, i.e., the value of the dropout variable is randomly set to 0 or 1, 2^n alternative networks will be produced during training and will make up the entire network for testing. From this perspective, it is more reasonable to take the continuous dropout distribution into account because, for continuous dropout variables, a hidden layer with n units can produce an infinite number of multiple alternative networks, which are expected to work better than the Bernoulli dropout proposed in [1]. The experimental results in Section IV demonstrate the superiority of continuous dropout over Bernoulli dropout.

III. Co-Adaptation Regularization in Continuous Dropout

In this section, we derive the static and dynamic properties of our continuous dropout. Static properties refer to the properties of the network with a fixed set of weights, that is, given an input, how dropout affects the output of the network. Dynamic properties refer to the properties of updating of the weights for the network, i.e., how continuous dropout changes the learning process of the network [12]. Because Bernoulli dropout with $p = 0.5$ achieves the best performance in most situations [1], [20], we set $p = 0.5$ for Bernoulli dropout. For our continuous dropout, we apply $\mu \sim U(0, 1)$ and $g \sim \mathcal{N}(0.5, \sigma^2)$ for uniform dropout and Gaussian dropout to ensure that all three dropout algorithms have the same expected output (0.5).

A. Static Properties of Continuous Dropout

In this section, we focus on the static properties of continuous dropout, i.e., properties of dropout for a fixed set of weights. We start from the single layer of linear units, and then we extend it to multiple layers of linear and nonlinear units.

1) Continuous Dropout for a Single Layer of Linear Units: We consider a single fully connected (FC) linear layer with input $\mathbf{I} = [I_1, I_2, \ldots, I_n]^T$, weighting matrix $W = [w_{ij}]_{k\times n}$, and output $\mathbf{S} = [S_1, S_2, \ldots, S_k]^T$. The ith output $S_i = \sum_{j=1}^{n} w_{ij} I_j$. In Bernoulli dropout, each input unit I_j is kept with probability $p \sim$Bernoulli(0.5). The ith output and its expectation are

$$S_i^B = \sum_{j=1}^{n} w_{ij} I_j p_j \quad \text{and} \quad \mathbf{E}\left[S_i^B\right] = \frac{1}{2}\sum_{j=1}^{n} w_{ij} I_j.$$

In our uniform dropout, I_j is kept with probability $u \sim U(0, 1)$. The output becomes

$$S_i^U = \sum_{j=1}^{n} w_{ij} I_j u_j \quad \text{and} \quad \mathbf{E}\left[S_i^U\right] = \frac{1}{2}\sum_{j=1}^{n} w_{ij} I_j.$$

When Gaussian dropout is applied, I_j is kept with probability $g \sim \mathcal{N}(0.5, \sigma^2)$

$$S_i^G = \sum_{j=1}^{n} w_{ij} I_j g_j \quad \text{and} \quad \mathbf{E}\left[S_i^G\right] = \frac{1}{2}\sum_{j=1}^{n} w_{ij} I_j.$$

Therefore, the three dropout methods achieve the same expected output.

Because dropout is applied to the input units independently, the variance and covariance of the output units are

$$\begin{aligned}
\mathrm{Var}\big(S_i^U\big) &= \sum_{j=1}^{n} w_{ij}^2 I_j^2 \mathrm{Var}(u_j) = \sum_{j=1}^{n} w_{ij}^2 I_j^2 \frac{1}{12} \\
\mathrm{Cov}\big(S_i^U, S_l^U\big) &= \sum_{j=1}^{n} w_{ij} w_{lj} I_j^2 \frac{1}{12} \\
\mathrm{Var}\big(S_i^G\big) &= \sum_{j=1}^{n} w_{ij}^2 I_j^2 \mathrm{Var}(g_i) = \sum_{j=1}^{n} w_{ij}^2 I_j^2 \sigma^2 \\
\mathrm{Cov}\big(S_i^G, S_l^G\big) &= \sum_{j=1}^{n} w_{ij} w_{lj} I_j^2 \sigma^2 \\
\mathrm{Var}\big(S_i^B\big) &= \sum_{j=1}^{n} w_{ij}^2 I_j^2 p_j q_j = \sum_{j=1}^{n} w_{ij}^2 I_j^2 \frac{1}{4} \\
\mathrm{Cov}\big(S_i^B, S_l^B\big) &= \sum_{j=1}^{n} w_{ij} w_{lj} I_j^2 p_j q_j = \sum_{j=1}^{n} w_{ij} w_{lj} I_j^2 \frac{1}{4}.
\end{aligned}$$

The aim of dropout is to avoid the co-adaptation of feature detectors, reflected by the covariance between output units. Generally, networks with lower covariance between feature detectors tend to generate more independent subnetworks and therefore tend to work better during the test stage. Comparing the covariance of the output units of the three dropout algorithms, we can see that uniform dropout has a lower covariance than Bernoulli dropout. The covariance of Gaussian dropout is controlled by the parameter σ^2. Through extensive experiments, we find that Gaussian dropout with $\sigma^2 \in [(1/5), (1/3)]$ works the best among the three dropout algorithms. This phenomenon implies that there is a balance between the diversity of subnetworks (larger variance of the output of hidden units) and their independence (lower covariance between units in the same layer). Bernoulli dropout achieves the highest variance but its covariance is also the highest. In contrast, uniform dropout achieves the lowest covariance, but its variance is also the lowest. Gaussian dropout with a suitable σ^2 achieves the best balance between variance and covariance, ensuring a good generalization capability.

2) Continuous Dropout Approximation for Nonlinear Unit: For the nonlinear unit, we consider the case that the output of a single unit with total linear input S is given by the logistic sigmoidal function

$$O = \mathrm{sigmoid}(S) = \frac{1}{1 + ce^{-\lambda S}}. \tag{1}$$

For uniform dropout, $S = \sum_{i=1}^{n} w_i I_i u_i, u_i \sim U(0,1)$. We have $S = \sum_{i=1}^{n} U_i, U_i \sim U(0, w_i I_i)$, $\mathbf{E}[U_i] = (1/2) w_i I_i$, and $\mathrm{Var}(U_i) = (1/12) w_i^2 I_i^2$. Because $U_i \leq \max_i(w_i I_i)$, $s_n^2 = \sum_{i=1}^{n} \mathrm{Var}(U_i) \to \infty$. According to Corollary 2.7.1 of Lyapunov's central limit theorem [18], S tends to a normal distribution as $n \to \infty$. It yields that

$$\begin{aligned}
S &\sim \mathcal{N}\big(\mu_U, \sigma_U^2\big) \\
\mu_U &= \sum_{i=1}^{n} \mathbf{E}[U_i] = \sum_{i=1}^{n} \frac{1}{2} w_i I_i \\
\sigma_U^2 &= \sum_{i=1}^{n} \mathrm{Var}(U_i) = \sum_{i=1}^{n} \frac{1}{12} w_i^2 I_i^2.
\end{aligned} \tag{2}$$

For Gaussian dropout, $S = \sum_{i=1}^{n} w_i I_i g_i, g_i \sim \mathcal{N}(\mu, \sigma^2)$ and g_i i.i.d. We can easily infer that $S \sim \mathcal{N}(\mu_S, \sigma_S^2)$, where $\mu_S = \sum_{i=1}^{n} w_i I_i \mu$ and $\sigma_S^2 = \sum_{i=1}^{n} w_i^2 \ I_i^2 \sigma^2$.

Thus, for both uniform dropout and Gaussian dropout, S is subject to a normal distribution. In the following sections, we derive only the statistical property of Gaussian dropout because it is the same for uniform dropout.

The expected output is

$$\begin{aligned}
\mathbf{E}(O) &= \mathbf{E}[\mathrm{sigmoid}(S)] \\
&= \int_{-\infty}^{\infty} \mathrm{sigmoid}(x) \mathcal{N}\big(x | \mu_S, \sigma_S^2\big)\, dx \\
&\approx \mathrm{sigmoid}\left(\frac{\mu_S}{\sqrt{1 + \pi \sigma_S^2/8}}\right).
\end{aligned} \tag{3}$$

This means that for Gaussian dropout $g_i \sim \mathcal{N}(\mu, \sigma^2)$, we have the recursion

$$\mathbf{E}\big[S_i^h\big] = \sum_{l<h} \sum_{j} w_{ij}^{hl} \mathbf{E}\big[g_j^l\big] \mathbf{E}\big[O_j^l\big]$$

and

$$\mathbf{E}\big[O_i^h\big] \approx \mathrm{sigmoid}\left(\frac{\mathbf{E}\big[S_i^h\big]}{1 + \pi \mathrm{Var}\big(S_i^h\big)/8}\right) \tag{4}$$

while for Bernoulli dropout [12]

$$\begin{aligned}
\mathbf{E}\big[S_i^h\big] &= \sum_{l<h} \sum_{j} w_{ij}^{hl} \ \mathbf{E}\big[\delta_j^l\big] \mathbf{E}\big[O_j^l\big] \\
\mathbf{E}\big[O_i^h\big] &\approx \mathrm{sigmoid}\big(\mathbf{E}\big[S_i^h\big]\big).
\end{aligned} \tag{5}$$

In Bernoulli dropout, the expected output is only the propagation of deterministic variables among the entire network, whereas our continuous dropout has a regularization term of $(1 + \pi \mathrm{Var}(S_i^h)/8)^{1/2} = (1 + \pi(\sum_{i=1}^{n} w_i^2 I_i^2 \sigma^2)/8)^{1/2}$. Thus, continuous dropout can regularize complex weights and inputs during forward propagation.

B. Dynamic Properties of Continuous Dropout

In this section, we will investigate the dynamic properties of continuous dropout related to the training procedure and the update of the weights. We also start from the simple case of a single linear unit, and then we discuss the nonlinear case. As proven in the last section, in uniform dropout, the S tends to a normal distribution as $n \to \infty$. Therefore, we analyze the dynamic properties of Gaussian dropout only.

1) Continuous Dropout Gradient and Adaptive Regularization—Single Linear Unit: In the case of a single linear unit trained with dropout with an input I, an output $O = S$, and a target t, the error is typically quadratic of the form $E_D = (1/2)(t - O)^2$, where $O = S = \sum_i w_i p_i I_i$. In the linear case, the ensemble network is identical to the deterministic network obtained by scaling the connections using the dropout probabilities. For a single output O, the ensemble error of all possible subnetworks E_{ENS} is defined by

$$E_{\mathrm{ENS}} = \frac{1}{2}(t - O_{\mathrm{ENS}})^2 = \frac{1}{2}\left(t - \sum_{i=1}^{n} \mu w_i I_i\right)^2.$$

The gradients of the ensemble error can be computed by

$$\frac{\partial E_{\text{ENS}}}{\partial w_i} = -(t - O_{\text{ENS}})\mu I_i. \tag{6}$$

For Gaussian dropout, $E_D = (1/2)(t - \sum_{i=1}^{n} g_i w_i I_i)^2$. Here, $g \sim \mathcal{N}(\mu, \sigma^2)$ is the random variable with a Gaussian distribution. Hence, E_D is a random variable, while E_{ENS} is a deterministic function.

For dropout error, the learning gradients are of the form

$$\frac{\partial E_D}{\partial w_i} = \frac{\partial E_D}{\partial O}\frac{\partial O}{\partial w_i} = -(t - O)\frac{\partial O}{\partial w_i}$$

and therefore

$$\begin{aligned}\frac{\partial E_D}{\partial w_i} &= -(t - O_D)g_i I_i \\ &= -t g_i I_i + w_i g_i^2 I_i^2 + \sum_{j \neq i} w_j g_i g_j I_i I_j \end{aligned} \tag{7}$$

and

$$\begin{aligned}\mathbf{E}\left[\frac{\partial E_D}{\partial w_i}\right] &= -t\mu I_i + w_i(\mu^2 + \sigma^2)I_i^2 + \sum_{j \neq i} w_j \mu^2 I_i I_j \\ &= \frac{\partial E_{\text{ENS}}}{\partial w_i} + w_i I_i^2 \sigma^2. \end{aligned} \tag{8}$$

Remarkably, the relationship between the expectation of ensemble error and dropout error is

$$E_D = E_{\text{ENS}} + \frac{1}{2}\sum_{i=1}^{n} w_i^2 I_i^2 \sigma^2. \tag{9}$$

In Bernoulli dropout [12], this relationship is

$$E_D = E_{\text{ENS}} + \frac{1}{2}\sum_{i=1}^{n} w_i^2 I_i^2 \text{Var}(p_i). \tag{10}$$

Generally, the regularization term is weight decay based on the square of the weights, and it ensures that the weights do not become too large to overfit the training data. Bernoulli dropout extends this regularization term by incorporating the square of the input terms and the variance of the dropout variables; however, both the expected output and the weight of regularization term are determined by the dropout probability (p), i.e., there is no freedom for adjusting the model complexity to reduce overfitting. In contrast, in Gaussian dropout, we have an extra degree of freedom of σ^2 to achieve the balance between network output and model complexity.

2) Continuous Dropout Gradient and Adaptive Regularization—Single Sigmoidal Unit: In Gaussian dropout, for a single sigmoidal unit

$$O = \text{sigmoid}(S) = \frac{1}{1 + ce^{-\lambda S}}$$

where $S = \sum_i w_i g_i I_i$ and $S_{\text{ENS}} = \mathbf{E}[S] = \sum_i w_i \mu I_i$ with $\sigma_S^2 = \sum_i w_i^2 I_i^2 \sigma^2$ and $\mu_S = \sum_i w_i \mu I_i$. Commonly, we use relative entropy error

$$E_D = -(t \log O + (1 - t)\log(1 - O)). \tag{11}$$

By the chain rule $(\partial E_D/\partial w_i) = (\partial E_D/\partial O)(\partial O/\partial S)(\partial S/\partial w_i)$, we obtain

$$\frac{\partial E_D}{\partial w_i} = -\lambda(t - O)\frac{\partial S}{\partial w_i}.$$

For the ensemble network

$$\frac{\partial E_{\text{ENS}}}{\partial w_i} = -\lambda(t - O_{\text{ENS}})\frac{\partial S_{\text{ENS}}}{\partial w_i}.$$

We have

$$\begin{aligned} O_{\text{ENS}} &= \mathbf{E}[\text{sigmoid}(S)] \\ &= \int_{-\infty}^{\infty} \frac{e^S}{1 + e^S} e^{-\frac{S - \mu_S^2}{2\sigma_S^2}} ds \\ &\approx \text{sigmoid}\left(\frac{\mu_S}{\sqrt{1 + \pi\sigma_S^2/8}}\right). \end{aligned} \tag{12}$$

Therefore

$$\frac{\partial E_{\text{ENS}}}{\partial w_i} = -\lambda\left(t - \text{sigmoid}\left(\frac{\mu_S}{\sqrt{1 + \pi\sigma_S^2/8}}\right)\right)\mu I_i. \tag{13}$$

For the dropout network

$$\begin{aligned}\frac{\partial E_D}{\partial w_i} &= -\lambda(t - O)g_i I_i \\ &= -\lambda\left(t - \text{sigmoid}\left(\sum\nolimits_j w_j g_j I_j\right)\right) g_i I_i. \end{aligned} \tag{14}$$

Here, g_i are the random variables with Gaussian distributions; thus, $O_D = \text{sigmoid}(\sum_j w_j g_j I_j)$ and g_i are both random variables. It yields that

$$\mathbf{E}\left[\frac{\partial E_D}{\partial w_i}\right] = \mathbf{E}\left[-\lambda\left(t - \text{sigmoid}\left(\sum_j w_j g_j I_j | g_j = \mu\right)\right)\mu I_i\right] \tag{15}$$

where $O_D^{'} = \text{sigmoid}(\sum_j w_j g_j I_j | g_j = \mu)$, $\mathbf{E}[O_D^{'}] = \mu_S^{'} = \mu_S = \sum_i w_i \mu I_i$, and $\text{Var}(O_D^{'}) = \sum_{j \neq i} w_j^2 \ I_j^2 \sigma^2$

$$\mathbf{E}\left[O_D^{'}\right] \approx \text{sigmoid}\left(\frac{\mu_S^{'}}{\sqrt{1 + \pi\sigma_S^{'2}/8}}\right). \tag{16}$$

The gradient of the dropout is

$$\begin{aligned}\mathbf{E}\left[\frac{\partial E_D}{\partial w_i}\right] &\approx \frac{\partial E_{\text{ENS}}}{\partial w_i} + \lambda\mu I_i\left(\text{sigmoid}\left(\frac{\mu_S}{\sqrt{1 + \pi\sigma_S^{'}/8}}\right)\right. \\ &\qquad \left. - \text{sigmoid}\left(\frac{\mu_S}{\sqrt{1 + \pi\sigma_S^2/8}}\right)\right) \\ &\approx \frac{\partial E_{\text{ENS}}}{\partial w_i} + \lambda\mu I_i \text{sigmoid}^{'}\left(\frac{\mu_S}{\sqrt{1 + \pi\sigma_S^2/8}}\right) \\ &\quad \times \left(\frac{\sqrt{1 + \pi\sigma_S^2/8} - \sqrt{1 + \pi\sigma_S^{'2}/8}}{\sqrt{(1 + \pi\sigma_S^{'2}/8)(1 + \pi\sigma_S^2/8)}}\right) \end{aligned}$$

Fig. 1. Samples of benchmark data sets. MNIST and SVHN are digit classification tasks. NORB, CIFAR-10, and ImageNet 2012 are object recognition tasks. All of them are formulated as classification problems, which is commonly evaluated by classification accuracy (error).

$$
\begin{aligned}
&\approx \frac{\partial E_{\text{ENS}}}{\partial w_i} + \lambda \mu I_i \text{sigmoid}'\left(\frac{\mu_S}{\sqrt{1+\pi\sigma_S^2/8}}\right) \\
&\quad \times \left(\frac{\frac{\pi}{16}\mu_S w_i^2 I_i^2 \sigma^2}{\sqrt{\left(1+\pi\sigma_S'^2/8\right)\left(1+\pi\sigma_S^2/8\right)}}\right) \\
&\approx \frac{\partial E_{\text{ENS}}}{\partial w_i} + \lambda \mu I_i \text{sigmoid}'\left(\frac{\mu_S}{\sqrt{1+\pi\sigma_S^2/8}}\right) \\
&\quad \times \left(\frac{\frac{\pi}{16}\mu_S w_i^2 I_i^2 \sigma^2}{1+\pi\sigma_S^2/8}\right) \\
&= \frac{\partial E_{\text{ENS}}}{\partial w_i} + \lambda \mu I_i \text{sigmoid}'\left(\frac{\mu_S}{\sqrt{1+\pi\sigma_S^2/8}}\right) \\
&\quad \times \left(\frac{\frac{\pi}{16}\left(\sum_j w_j \mu I_j\right)\left(w_i^2 I_i^2 \sigma^2\right)}{1+\frac{\pi}{8}\sum_i w_i^2 I_i^2 \sigma^2}\right).
\end{aligned}
$$

For approximation

$$
\begin{aligned}
&E_D \\
&= E_{\text{ENS}} + \sum_{i=1}^{n} \lambda w_i \mu_i I_i \text{sigmoid}'\left(\frac{\mu_S}{\sqrt{1+\pi\sigma_S^2/8}}\right) \\
&\quad \times \left(\frac{\frac{\pi}{16}\left(\sum_j w_j \mu I_j\right)\left(w_i^2 I_i^2 \sigma^2\right)}{1+\frac{\pi}{8}\sum_i w_i^2 I_i^2 \sigma^2}\right) \\
&= E_{\text{ENS}} + \frac{1}{2}\lambda \text{sigmoid}'\left(\frac{\mu_S}{\sqrt{1+\pi\sigma_S^2/8}}\right) \\
&\quad \times \sum_{i=1}^{n}\sum_{j} w_i w_j \mu^2 I_i I_j \left(\frac{\pi}{8} w_i^2 I_i^2 \sigma^2\right) \Big/ \left(1+\frac{\pi}{8}\sum_i w_i^2 I_i^2 \sigma^2\right).
\end{aligned} \tag{17}
$$

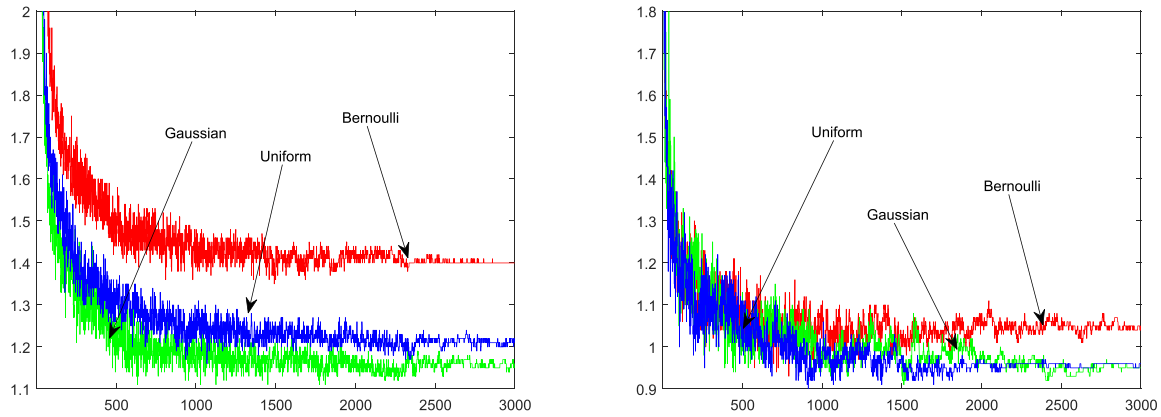
Note that for Bernoulli dropout [12]

$$
E_D = E_{\text{ENS}} + \frac{1}{2}\lambda \text{sigmoid}'(U) \sum_{i=1}^{n} w_i^2 I_i^2 \text{Var}(p_i). \tag{18}
$$

Bernoulli dropout provides only the magnitude of the regularization term, which is adaptively scaled by the square of the input terms, by the gain λ of the sigmoidal function, by the variance of the dropout variables, and by the instantaneous derivative of the sigmoidal function; however, this term tends to achieve only a simpler model and avoid overfitting. It has little help in avoiding the co-adaptation of units (feature detectors) in the same layer. In contrast, continuous dropout not only provides the regularization of squares of input units, weights, and dropout variance individually ($\sum_i w_i^2 \ I_i^2 \sigma^2$), but also regularizes the covariance between input units ($I_i I_j$) and weights ($w_i w_j$). In other words, in Gaussian dropout, the regularization term penalizes the covariance between weights, dropout variables, and input units; that is, it prevents the co-adaptation of feature detectors in the neural network. Therefore, through this co-adaptation regularization, Gaussian dropout can indeed avoid co-adaptation and overfitting.

IV. Experiments

We investigate the performance of our continuous dropout on MNIST [21], CIFAR-10 [22], SVHN [23], NORB [24], and ImageNet ILSVRC-2012 classification task [25]. Samples and brief description of these data sets are presented in Fig. 1. We compare continuous dropout with the original dropout proposed in [1] (Bernoulli dropout), adaptive dropout [9], and DropConnect [10]. Fast Dropout [8] is an approximation of Bernoulli dropout that accelerates the sampling process. Its performance is similar to that of Bernoulli dropout. For evaluation metric, the classification error, which is defined as the ratio of misclassified samples to all samples, is applied (0/1 loss). We use the publicly available THEANO library [26] to implement the feedforward neural networks that consist of



(a) (b)

Fig. 2. (a) and (b) *Testing errors versus epochs* of Bernoulli, uniform, and Gaussian dropouts on MNIST. No data augmentation is used in this experiment. By regularizing the covariance between neurons in the same layer, the capacity of the neural network is improved.

FC layers only, and the networks that consist of convolutional neural networks (CNNs) are implemented based on Caffe [27]. In all experiments, the dropout rate in Bernoulli dropout and DropConnect is set as 0.5 because this is the most commonly used configuration in dropout and performs the best. All the other parameters are selected based on performance on the validation set. To ensure that all three dropout algorithms achieve the same expected output, for uniform dropout, the variables u_i are subject to $U(0, 1)$. In Gaussian dropout, $g_i \sim \mathcal{N}(0.5,\ \sigma^2)$, and σ^2 is selected from $\{0.2,\ 0.3,\ 0.4\}$. For adaptive dropout, α is selected from $\{-1,\ 0,\ 1\}$ and β is selected from $\{-0.5, 0, 0.5\}$. To avoid divergence during propagation, we clip the Gaussian dropout variable to be in $[0, 1]$, yielding $g_i = 1$ if $g_i \geq 1$ and $g_i = 0$ if $g_i \leq 0$.

To verify whether the performance gain is statistically significant, we repeated all experiments N times for all methods and reported the mean error and standard derivation. Here, $N = 30$ for data sets MNIST, CIFAR-10, SVHN, and NORB, and $N = 10$ for data set ImageNet ILSVRC-2012 because of the high computational cost in this data set. In each of the N independent runs, we randomly initialized weights of the network and then applied different dropout algorithms to train this network. In other words, in the ith independent run ($i = 1, 2, \ldots, N$), all dropout algorithms share the same weights initialization. In another independent run, the network was randomly initialized again, i.e., the network had different initialized weights in the ith run and the jth run ($i \neq j$). In this way, we obtained N groups of results and then conducted paired t-test and paired Wilcoxon signed rank test between Gaussian dropout and all other baseline methods. Their p-values are reported.

A. Experiments on MNIST

We first verify the effectiveness of our continuous dropout on MNIST. The MNIST handwritten digit data set consists of 60 000 training images and 10 000 test images. Each image is 28×28 pixels in size. We randomly separate the 60 000 training images into two parts: 50 000 for training and 10 000 for validation. We replicate the results of dropout in [1] and use the same settings for uniform dropout and Gaussian dropout. These settings include a linear momentum schedule, a constant weight constraint, and an exponentially decaying learning rate. More details can be found in [1].

We train models with two FC layers using sigmoid or rectified linear unit (ReLU) activation functions ($784-800-800-10$). Table I shows the performance when image pixels are taken as the input and no data augmentation is utilized. From Table I, we can see that both uniform dropout and Gaussian dropout outperform Bernoulli dropout, adaptive dropout, and DropConnect on this data set, irrespective of whether sigmoid or ReLU is applied. Gaussian dropout achieves slightly better performance than uniform dropout. To further analyze the effects of continuous dropout, Fig. 2 shows the testing errors versus epochs of Bernoulli dropout, uniform dropout, and Gaussian dropout. We can see that continuous dropout achieves a considerably lower testing error than Bernoulli dropout, which demonstrates that continuous dropout has a better generalization capability.

1) Influence of Variance in Gaussian Dropout: In Section III, we find that we have an extra degree of freedom using σ^2 to achieve the balance between network output and model complexity. To investigate the influence of σ^2 on model performance in Gaussian Dropout, we train Gaussian Dropout models with $784 - 800 - 800 - 10$ neurons. Dropout masks are sampled from Gaussian distribution with mean 0.5 and variance in $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1.0\}$. Activation functions are set to be sigmoid or ReLU. Performances of Gaussian Dropout with different standard deviations are shown in Fig. 3. We can see that the best variance for Gaussian Dropout is $\{0.2,\ 0.3\}$. For normal distributions, the values less than two standard deviations from the mean account for 95.45% of the set. And for three standard deviations, that is 99.73%. Thus, almost all the values of $\mathcal{N}(0.5,\ 0.2)$ and $\mathcal{N}(0.5,\ 0.3)$ distribute in $[0,\ 1]$ (reasonable distribution for dropout mask variables). Most importantly, our Gaussian Dropout consistently outperforms Bernoulli Dropout for all sigma values, which demonstrate that the performance gain in Gaussian Dropout mainly comes from the distribution not the extra freedom of sigma.

TABLE I

PERFORMANCE COMPARISON ON MNIST (MEAN ERROR AND STANDARD DERIVATION). NO DATA AUGMENTATION IS USED. ARCHITECTURE: $784-800-800-10$. PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	Error (%) (p-value-T/p-value-W)	
	Sigmoid	ReLU
No dropout	1.58 $\pm$ 0.045 (6.5e-26/9.1e-7)	1.15 $\pm$ 0.036 (1.7e-19/9.1e-7)
Bernoulli dropout	1.35 $\pm$ 0.049 (3.1e-17/9.1e-7)	1.06 $\pm$ 0.037 (3.6e-17/9.1e-7)
Adaptive dropout	1.30 $\pm$ 0.072 (7.4e-11/1.1e-6)	1.02 $\pm$ 0.027 (1.9e-11/1.2e-6)
DropConnect	1.37 $\pm$ 0.058 (8.7e-19/9.1e-7)	1.01 $\pm$ 0.052 (8.5e-6/6.0e-5)
Uniform dropout	1.21 $\pm$ 0.046 (9.0e-7/1.2e-5)	0.96 $\pm$ 0.039 (0.031/0.027)
Gaussian dropout	**1.15** $\pm$ 0.035	**0.95** $\pm$ 0.028

TABLE II

PERFORMANCE COMPARISON ON MNIST WITH GAUSSIAN INITIALIZATION (MEAN ERROR AND STANDARD DERIVATION). NO DATA AUGMENTATION IS USED. PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	Architecture	Act Function	Error(%) (p-value-T/p-value-W)
No dropout	2CNN+1FC	ReLU	0.674 $\pm$ 0.047 (1.8e-15/9.1e-7)
Bernoulli dropout	2CNN+1FC	ReLU	0.551 $\pm$ 0.017 (3.3e-5/1.5e-4)
Adaptive dropout	2CNN+1FC	ReLU	0.591 $\pm$ 0.017 (8.6e-18/9.1e-7)
DropConnect	2CNN+1FC	ReLU	0.581 $\pm$ 0.012 (4.2e-18/9.1e-7)
Uniform dropout	2CNN+1FC	ReLU	0.549 $\pm$ 0.021 (2.9e-3/4.5e-3)
Gaussian dropout	2CNN+1FC	ReLU	**0.534** $\pm$ 0.006

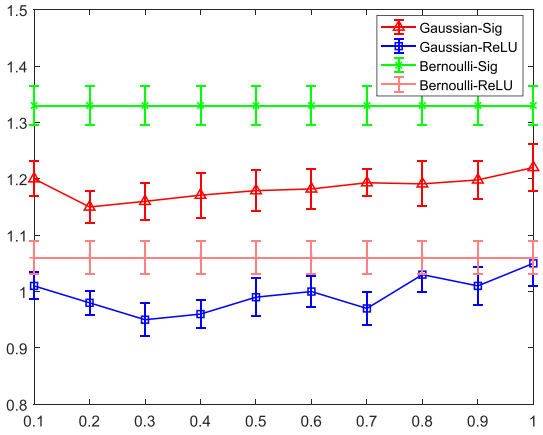


Fig. 3. Performance curve of Gaussian Dropout with respect to different variances. Our Gaussian Dropout consistently outperforms Bernoulli Dropout for all sigma values. It shows that the performance gain in Gaussian dropout mainly comes from the distribution not the extra freedom of σ.

2) Covariance of Hidden Units: In Section III, we demonstrate that continuous dropout can prevent the co-adaptation of feature detectors. To verify this property, we investigate the distribution of covariance between units in the same layer. We construct histograms of the variance of all pairs of units in the same layer in a trained $784-800-800-10$ MNIST model with ReLU. Fig. 4 shows the log of the number of pairs (N) whose covariance falls into different intervals. Histograms are obtained by taking all the 800×800 unit pairs in each layer and aggregating the results over 10 random input samples. For each sample, the dropout process is repeated 10 000 times to estimate the covariance. Fig. 4 shows that in continuous dropout, the distribution is more concentrated around 0, which indicates that continuous dropout performs better than Bernoulli dropout in preventing the co-adaptation of feature detectors. Furthermore, comparing Fig. 4(a) and (b), we can see that in "no dropout," the covariance in the second layer is much more concentrated around 0 than that in the first layer. After using continuous dropout, the covariance curve becomes more concentrated than "no dropout" in both layers. The reason why the effects of continuous dropout become less significant in a higher layer is that the room for improvement (reduce covariance) becomes smaller in a higher layer.

To further improve the classification results, we also apply a more powerful network, which consists of a two-layer CNN with $32-64$ feature maps and one fully connected layer with 150 ReLU units. All the dropout algorithms are applied on the FC layer. We use an initial learning rate of 0.01 and manually decay the learning rate by a multiplier (0.5 or 0.1) when the loss function of the validation error reaches a plateau. The input is also the original image pixels without cropping, rotation, or scaling. To verify whether the improvement of continuous dropout is benefited from a favored initialization, we initialize weights using both Gaussian distribution ($\mathcal{N}(0,\ 0.01)$) and uniform distribution proposed in [28]. The experimental results are summarized in Tables II and III. We can see that Gaussian dropout consistently performs the best among all dropout methods, no matter which initialization distribution is applied. Paired t-test and paired Wilcoxon signed rank test are conducted between Gaussian dropout and other methods. Tables II and III show that Gaussian dropout achieves statistically significant improvement over all baseline methods and the p-values are less than 0.05.

B. Experiments on CIFAR-10

The CIFAR-10 data set consists of 10 classes of 32×32 RGB images with 50 000 for training and 10 000 for testing.

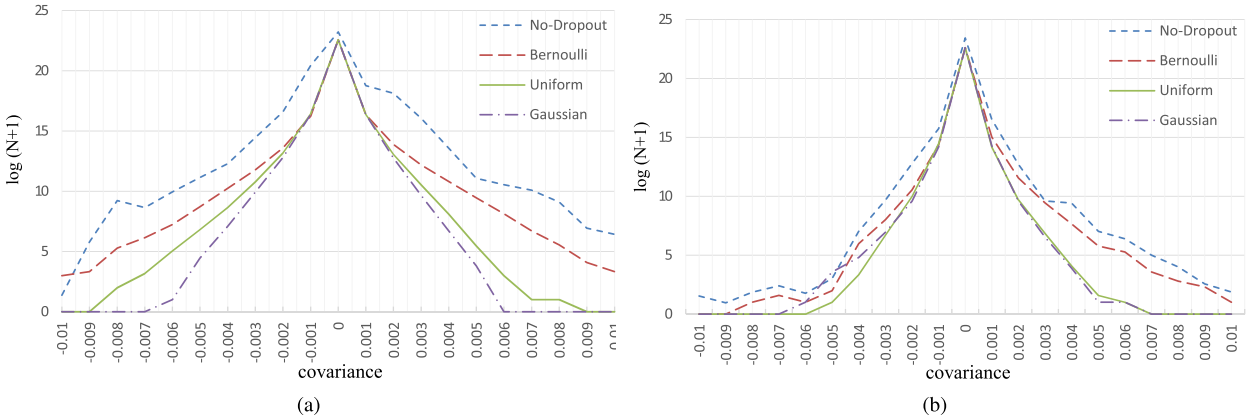


Fig. 4. Log histogram of covariance between pairs of units from the same layer. (a) Layer 1. (b) Layer 2. It shows that in continuous dropout, the distribution is more concentrated around 0, which indicates that continuous dropout performs better than Bernoulli dropout in preventing the co-adaptation of feature detectors (MNIST, 784 − 800 − 800 − 10, ReLU).

TABLE III

PERFORMANCE COMPARISON ON MNIST WITH UNIFORM INITIALIZATION (MEAN ERROR AND STANDARD DERIVATION). NO DATA AUGMENTATION IS USED. PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	Architecture	Act Function	Error(%) (p-value-T/p-value-W)
No dropout	2CNN+1FC	ReLU	0.670 ± 0.051 (3.0e-16/9.1e-7)
Bernoulli dropout	2CNN+1FC	ReLU	0.558 ± 0.018 (4.4e-8/6.5e-6)
Adaptive dropout	2CNN+1FC	ReLU	0.586 ± 0.016 (3.2e-13/1.0e-6)
DropConnect	2CNN+1FC	ReLU	0.579 ± 0.011 (4.6e-16/9.1e-7)
Uniform dropout	2CNN+1FC	ReLU	0.558 ± 0.018 (1.3e-11/1.4e-6)
Gaussian dropout	2CNN+1FC	ReLU	**0.521** ± 0.017

We preprocess the data by global contrast normalization and zero-phase component analysis whitening as in [29]. To produce comparable results with the state-of-the-art method, we apply all the dropout algorithms on the network-in-network model [30]. This network consists of seven convolutional layers and part of them are connected to pooling layers. Two dropout layers are applied to the pooling layers. To compare continuous dropout with adaptive dropout and DropConnect, we slightly change this model by omitting the two dropout layers between the CNNs and replace the last pooling layer by two FC layers with 128 and 10 units, respectively. Dropout is applied to the first FC layer. During training, we first initialize our model by the weights trained in [30], and then we fine-tune the model using different dropout methods. The learning rate is initialized by 0.01 and decayed by 10 every 3000 iterations, without any data augmentations.

The models are tested after 10 000 iterations, and the results are presented in Table IV. We can see that Gaussian dropout achieves the best performance among all dropout algorithms on this task again. Based on the results of paired t-test and paired Wilcoxon signed rank test, Gaussian dropout significantly outperforms all other methods (p-values are less than 0.05). To further investigate their performance on each class, confusion matrices are also reported, as shown in Fig. 5. We can see that Gaussian Dropout achieves the best performance on five classes among all six methods. Specifically, Gaussian Dropout achieves higher classification accuracy on 10, 8, 8, 8, and 7 classes than no dropout, Bernoulli dropout, Adaptive dropout, DropConnect, and uniform dropout, respectively.

TABLE IV

PERFORMANCE COMPARISON ON CIFAR-10 (MEAN ERROR AND STANDARD DERIVATION). PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	Error(%) (p-value-T/p-value-W)
No dropout	10.65 ± 0.114 (1.5e-14/9.1e-7)
Bernoulli dropout	10.55 ± 0.050 (5.5e-15/9.1e-7)
Adaptive dropout	10.46 ± 0.081 (1.6e-10/9.1e-7)
DropConnect	10.40 ± 0.178 (2.9e-7/7.8e-6)
Uniform dropout	10.47 ± 0.142 (5.2e-10/2.0e-6)
Gaussian dropout	**10.18** ± 0.129

C. Experiments on SVHN

The SVHN data set includes 604 388 training images (both training set and extra set) and 26 032 testing images [23]. Like MNIST, the goal is to classify the digit centered in each 32 × 32 image (0–9). The data set is augmented by: 1) randomly selecting a 28 × 28 region from the original image; 2) introducing 15% scaling and rotation variations; and 3) randomly flipping images during training. Following [10], we preprocess the images using local contrast normalization as in [31].

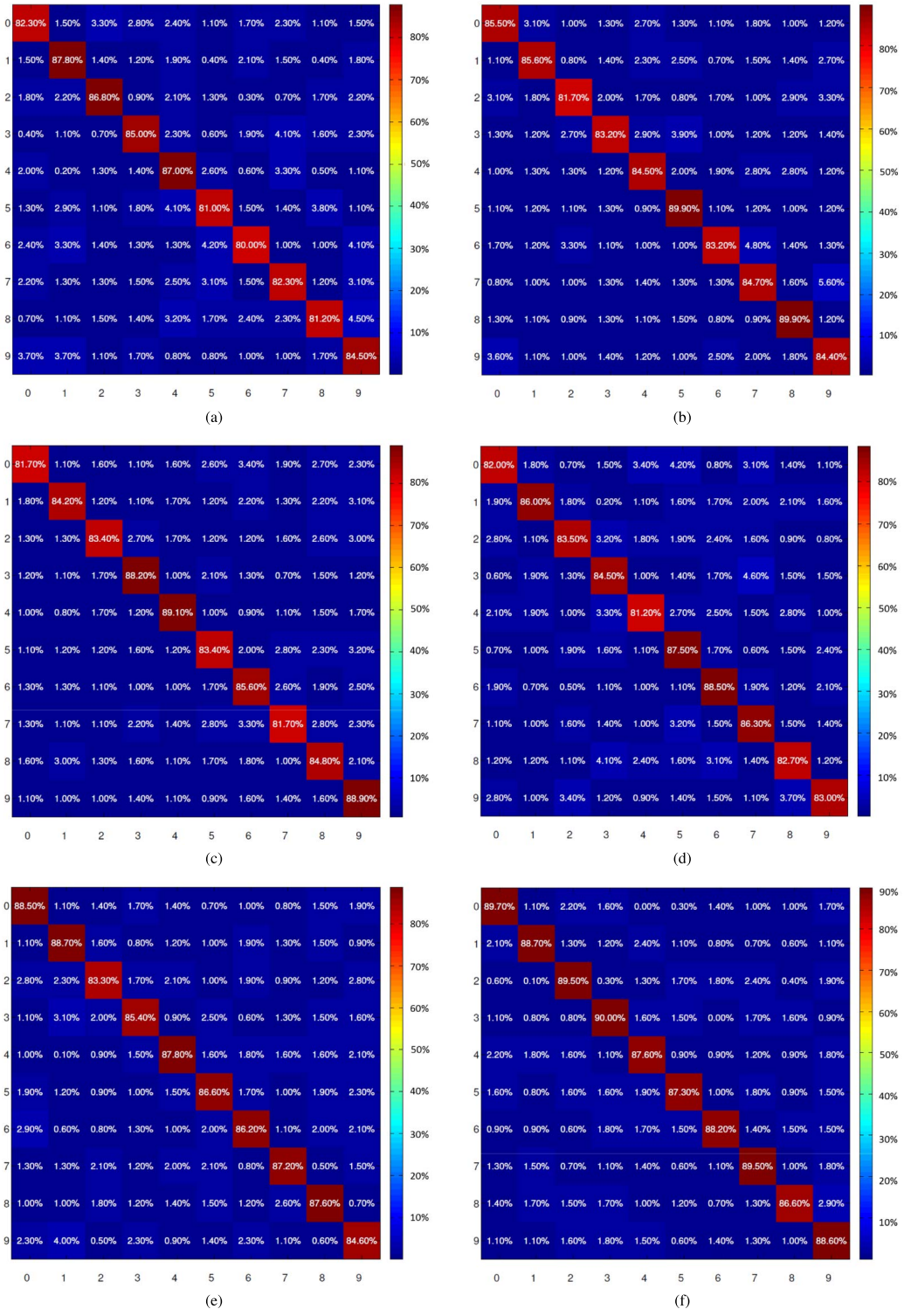


Fig. 5. Confusion matrices of all six methods. (a) No dropout. (b) Bernoulli dropout. (c) Adaptive dropout. (d) DropConnect. (e) Uniform dropout. (f) Gaussian dropout. We can see that Gaussian dropout achieves the best performance on five classes among all six methods. Specifically, Gaussian dropout achieves higher classification accuracy on 10, 8, 8, 8, and 7 classes than no dropout, Bernoulli dropout, adaptive dropout, DropConnect, and uniform dropout, respectively.

The model consists of two convolutional layers and two locally connected layers as described in [33] (layers-conv-local-11pct.cfg). An FC layer with 512 neurons and ReLU activations is added between the softmax layer and the final locally connected layer. We manually decrease the learning rate if the performance on validation set goes to plateaus [33]. In detail, we multiply the initial learning by 0.5 and then 0.1 repeatedly. Initial learning rate is set to 0.01. The bias

TABLE V

PERFORMANCE COMPARISON ON SVHN (MEAN ERROR AND STANDARD DERIVATION). PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	Error(%) (p-value-T/p-value-W)
No dropout	2.09 $\pm$ 0.002 (1.8e-35/9.1e-7)
Bernoulli dropout	2.00 $\pm$ 0.012 (4.2e-19/9.1e-7)
Adaptive dropout	2.13 $\pm$ 0.235 (3.5e-5/1.3e-4)
DropConnect	1.96 $\pm$ 0.007 (8.4e-16/9.1e-7)
Uniform dropout	**1.92** $\pm$ 0.012 (0.982/0.981)
Gaussian dropout	1.93 $\pm$ 0.012

TABLE VI

PERFORMANCE COMPARISON ON NORB (MEAN ERROR AND STANDARD DERIVATION). PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	Error(%) (p-value-T/p-value-W)
No dropout	3.55 $\pm$ 0.070 (1.4e-16/9.1e-7)
Bernoulli dropout	3.33 $\pm$ 0.128 (1.6e-5/9.9e-5)
Adaptive dropout	3.49 $\pm$ 0.204 (1.4e-8/2.0e-6)
DropConnect	3.53 $\pm$ 0.059 (1.2e-15/9.1e-7)
Uniform dropout	3.29 $\pm$ 0.197 (1.2e-3/2.4e-3)
Gaussian dropout	**3.15** $\pm$ 0.114

learning rate is set to be $2\times$ the learning rate for the weights. Additionally, weights are initialized with $\mathcal{N}(0,\ 0.1)$ random values for FC layers and $\mathcal{N}(0,\ 0.01)$ for convolutional layers. To further improve the performance, we train five independent networks with random permutations of the training sequence and different random seeds. We report the classification error of averaging the output probabilities from the five networks before making a prediction.

The experimental results on this data set are summarized in Table V. Comparing the mean classification errors, standard deviations, and p-values of paired t-test and paired Wilcoxon signed rank test, we can see that our proposed continuous dropout achieves better performance than no dropout, Bernoulli dropout, adaptive dropout, and DropConnect. The performance gain of Gaussian dropout is statistically significant (all p-values are less than 0.05). Uniform dropout and Gaussian dropout achieve similar performance on this data set. Besides, all dropout methods achieve stable performance on this data set with small standard deviation, except adaptive dropout that has a large standard deviation (0.235).

D. Experiments on NORB

In this experiment, we evaluate our models on the twofold NORB (jittered-cluttered) data set [24]. Each image is classified into one of the six classes, which appears on a random background. Images are downsampled from 108×108 to 48×48 as in [34]. We train on twofold of 29 160 images each and test on a total of 58 320 images. We use the same architecture as in SVHN. Data set is augmented by 15% rotation and scaling. No random crop or flip is applied. Models are trained with an initial learning rate of 0.01. Other training and testing settings are the same as in SVHN.

The experimental results are given in Table VI. From Table VI, we can see that Gaussian dropout significantly outperforms no dropout, adaptive dropout, DropConnect, and uniform dropout on this data set. Compared with the results on SVHN data set, all methods have a larger standard deviation on NORB data set. Experiments on these two data sets adopt the same network architecture and other experimental settings. The reason for higher standard deviation on NORB data set may be that we have much fewer training images on NORB. Therefore, the models trained on NORB are not as stable as that on SVHN.

E. Experiments on ILSVRC-2012

The ILSVRC-2012 data set was used for ILSVRC $2012-2014$ challenges. This data set includes images of 1000 classes, and is split into three sets: training (1.3M images), validation (50K images), and testing (100K images with held-out class labels). The classification performance is evaluated using two measures: the top-1 and top-5 error. The former is a multiclass classification error, and the latter is the main evaluation criteria used in ILSVRC, and is computed as the proportion of images such that the ground-truth category is outside the top-5 predicted categories.

We compare all the dropout algorithms by fine-tuning on the model with 16 layers proposed by Visual Geometry Group (VGG) team (configuration D) in [32]. The model consists of 13 convolution layers and three FC layers. All the filters used in the convolution layers are configured with 3×3 receptive field, and the numbers of channels are {64, 64, 128, 128, 256, 256, 256, 512, 512, 512, 512, 512, 512}, respectively. The convolution stride is fixed to 1 pixel; the spatial padding of convolution layer input is 1 pixel to preserve the spatial resolution of input. The convolutional layers are followed by three FC layers: the first two have 4096 channels each, and the third contains 1000 channels to perform 1000 way ILSVRC classification. All hidden layers are equipped with the rectification (ReLU [4]) and Bernoulli dropout is imposed on the first two FC layers. In our experiment, the two FC layers with Bernoulli dropout are replaced by adaptive dropout FC layers, DropConnect FC layers, and FC layers with uniform dropout and Gaussian dropout, respectively.

During training, weights are first initialized by the VGG_ILSVRC_16_layers model[1] in [32], and then fine-tuned by 100 000 iterations. The input to the ConvNet is fixed-sized 224×224 RGB images, which are zero-centered by a subtraction of [103.939, 116.779, 123.68] on BGR values. The batch size was set to 64, momentum to 0.9, and gradient clip to 35. The fine-tuning was regularized by weight

[1] http://www.robots.ox.ac.uk/~vgg/research/very_deep/

TABLE VII

PERFORMANCE COMPARISON ON IMAGENET ILSVRC-2012 (MEAN TOP-5/TOP-1 ERROR AND STANDARD DERIVATION). PAIRED t-TEST AND PAIRED WILCOXON SIGNED RANK TEST ARE CONDUCTED BETWEEN GAUSSIAN DROPOUT AND ALL OTHER BASELINE METHODS. THEIR p-VALUES ARE REPORTED: p-VALUE-T FOR t-TEST AND p-VALUE-W FOR WILCOXON SIGNED RANK TEST

Method	ConvNet config.	smallest image side		top-5 error(%) (p-value-T/p-value-W)	top-1 error(%) (p-value-T/p-value-W)
		train(S)	test(Q)		
Bernoulli Dropout[32]	VGG_ILSVRC_16_layers	256	256	8.86 ± 0.042 (9.5e-11/9.8e-4)	26.99 ± 0.065 (7.4e-11/9.8e-4)
Adaptive Dropout				8.41 ± 0.061 (1.1e-6/9.8e-4)	26.27 ± 0.046 (2.8e-8/9.8e-4)
DropConnect				8.56 ± 0.037 (2.3e-8/9.8e-4)	26.82 ± 0.050 (6.2e-11/9.8e-4)
Uniform Dropout				8.08 ± 0.048 (0.017/0.024)	25.91 ± 0.046 (0.005/0.014)
Gaussian Dropout				**7.99** ± 0.065	**25.79** ± 0.045

TABLE VIII

PERFORMANCE RANK OF DIFFERENT DROPOUT METHODS ON ALL FIVE DATA SETS

Method	MNIST	CIFAR-10	SVHN	NORB	ILSVRC-2012	Average rank
Bernoulli Dropout	3	5	4	3	5	4
Adaptive Dropout	5	3	5	4	3	4
DropConnect	4	2	3	5	4	3.6
Uniform Dropout	2	3	1	2	2	2
Gaussian Dropout	1	1	2	1	1	1.2

decay 5×10^{-4}. For adaptive dropout, alpha is set to 1 and beta is set to 0. Dropout ratio is 0.5 in DropConnect. In uniform dropout, mask is sampling from $U[0,\ 1]$, while the Gaussian dropout mask is sampled from $\mathcal{N}(0.5,\ 0.3^2)$. The learning rate was initially set to 10^{-4}, and then decreased by a factor of 10 after 50 000 iterations. Following [32], the smallest sides (denoted as S) of the training images are isotropically rescaled to 256.

During testing, the testing images are isotropically rescaled to a 256 smallest image side, denoted as Q. Then, the FC layers are first converted to convolutional layers (the first FC layer to a 7×7 convolutional layer and the last two FC layers to 1×1 convolutional layers). The resulting fully convolutional net is then applied to the whole (uncropped) image. The result is a class score map with the number of channels equal to number of classes. Then, the class score map is spatially averaged (sum-pooled). And the test set is also augmented by horizontal flipping of the images. Finally, the soft-max class posteriors of the original and flipped images are averaged to obtain final scores for the image as in [32].

Performances of all the dropout algorithms are shown in Table VII. Table VII shows that continuous dropout can improve the performance of conventional dropout algorithms even for very large scale data set. All the p-values are far less than 0.05, which indicates that Gaussian dropout achieves significantly performance gain over other methods on this data set.

To summarize the overall performance of different dropout methods, we rank all five dropout methods according to their performance on each of the five data sets, as shown in Table VIII. We can see that Gaussian dropout is ranked first on four data sets and ranked second on one data set.

V. CONCLUSION

In this paper, we have introduced a new explanation for the dropout algorithm from the perspective of the neural network properties in the human brain. The activation rate of neurons in neural networks for different situations is random and continuous. Inspired by this phenomenon, we extend the traditional binary dropout to continuous dropout. Thorough theoretical analyses and extensive experiments demonstrate that our continuous dropout has the advantage of reducing the co-adaptation while maintaining variance, and continuous dropout is equivalent to involving a regularizer that is able to prevent co-adaptation between feature detectors.

In the future, we plan to further explore continuous dropout from the following two aspects. First, although we have shown that continuous dropout penalizes the covariance between neurons, the corresponding regularization term is not explicitly defined. We will try to propose a more direct and interpretable way for the regularization term. Second, dropout is naturally viewed as a mixture of different models. From this perspective of view, we plan to derive an error bound for this way of mixture, leading to a more solid theoretical analysis of continuous dropout.

REFERENCES

[1] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (Jul. 2012). "Improving neural networks by preventing co-adaptation of feature detectors." [Online]. Available: https://arxiv.org/abs/1207.0580

[2] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[3] W. Hou, X. Gao, D. Tao, and X. Li, "Blind image quality assessment via deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1275–1286, Jun. 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[5] L. Shao, D. Wu, and X. Li, "Learning deep and wide: A spectral method for learning deep networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2303–2308, Dec. 2014.

[6] D. J. Spiegelhalter and S. L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," *Networks*, vol. 20, no. 5, pp. 579–605, 1990.

[7] L. Szymanski and B. McCane, "Deep networks are effective encoders of periodicity," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1816–1827, Oct. 2014.

[8] S. Wang and C. Manning, "Fast dropout training," in *Proc. ICML*, 2013, pp. 118–126.

[9] J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *Proc. NIPS*, 2013, pp. 3084–3092.

[10] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1058–1066.
[11] S. Wager, S. Wang, and P. S. Liang, "Dropout training as adaptive regularization," in *Proc. NIPS*, 2013, pp. 351–359.
[12] P. Baldi and P. Sadowski, "The dropout learning algorithm," *Artif. Intell.*, vol. 210, pp. 78–122, May 2014.
[13] J. Chorowski and J. M. Zurada, "Learning understandable neural networks with nonnegative weight constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 62–69, Jan. 2015.
[14] G. Buzsáki and K. Mizuseki, "The log-dynamic brain: How skewed distributions affect network operations," *Nature Rev. Neurosci.*, vol. 15, no. 4, pp. 264–278, 2014.
[15] P. Fatt and B. Katz, "Spontaneous subthreshold activity at motor nerve endings," *J. Physiol.*, vol. 117, no. 1, pp. 109–128, 1952.
[16] J. M. Bekkers, G. B. Richerson, and C. F. Stevens, "Origin of variability in quantal size in cultured hippocampal neurons and hippocampal slices," *Proc. Nat. Acad. Sci. USA*, vol. 87, no. 14, pp. 5359–5362, 1990.
[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] E. L. Lehmann, *Elements of Large-Sample Theory*. New York, NY, USA: Springer, 1999.
[19] A. Livnat, C. Papadimitriou, N. Pippenger, and M. Feldman, "Sex, mixability, and modularity," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 4, pp. 1452–1457, 2010.
[20] N. Srivastava, "Improving neural networks with dropout," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.
[21] 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.
[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009, p. 7, vol. 1.
[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.
[24] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proc. CVPR*, 2004, p. 104.
[25] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[26] O. Bergstra *et al.*, "Theano: A CPU and GPU math expression compiler," in *Proc. Python Sci. Comput. Conf.*, vol. 4. Austin, TX, USA, 2010, p. 3.
[27] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM MM*, 2014, pp. 675–678.
[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.
[29] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. (Feb. 2013). "Maxout networks." [Online]. Available: https://arxiv.org/abs/1302.4389
[30] M. Lin, Q. Chen, and S. Yan. (Dec. 2013). "Network in network." [Online]. Available: https://arxiv.org/abs/1312.4400
[31] M. Zerler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *Proc. ICLR*, 2013, pp. 1–9.
[32] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[33] A. Krizhevsky. (2012). *Cuda-Convnet*. [Online]. Available: http://code.google.com/p/cuda-convnet/
[34] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. CVPR*, Jun. 2012, pp. 3642–3649.

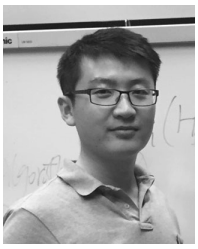
Xu Shen received the B.S. and Ph.D. degrees from the Department of Electronic Engineering and Information Science from, University of Science and Technology of China, Hefei, China, in 2012 and 2017, respectively.

His current research interests include multimedia, computer vision, and deep learning.

Xinmei Tian (M'13) received the B.E. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2005 and 2010, respectively.

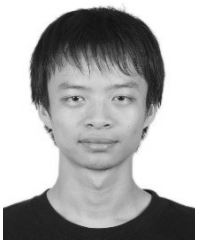
She is currently an Associate Professor with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, USTC. Her current research interests include multimedia information retrieval and machine learning.

Dr. Tian received the Excellent Doctoral Dissertation of the Chinese Academy of Sciences Award in 2012 and the Nomination of the National Excellent Doctoral Dissertation Award in 2013.



Tongliang Liu received the B.Eng. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree from the University of Technology Sydney, Ultimo, NSW, Australia.

He is currently a Lecturer with the School of Information Technologies, a Faculty Member of Engineering and Information Technologies, and a Core Member with the UBTECH Sydney AI Centre, The University of Sydney, Sydney, NSW, Australia. He has authored or co-authored over 30 research papers in journals including IEEE T-PAMI, T-NNLS, T-IP, ICML, and KDD. His current research interests include statistical learning theory, computer vision, and optimization.



Fang Xu received the B.S. degree and Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2009 and 2016, respectively.

He is currently a Post-Doctoral Fellow with USTC. His current research interests include cellular neuroscience and computational neuroscience.

Dacheng Tao (F'15) is currently a Professor of computer science with the School of Information Technologies, a Faculty Member of Engineering and Information Technologies, and the Inaugural Director with the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Sydney, NSW, Australia. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. He has authored or co-authored one monograph and over 500 papers in prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, CIKM, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM, and ACM SIGKDD. His current research interests include computer vision, data science, image processing, machine learning, and video surveillance.

Mr. Tao is a fellow of the OSA, IAPR, and SPIE. He was a recipient of the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM'07, the IEEE ICDM 2013 Best Student Paper Award, the 2014 ICDM 10-Year Highest-Impact Paper Award, the 2015 ACS Gold Disruptor Award, and the 2017 IEEE Signal Processing Society Best Paper Award. He received the 2015 Australian Scopus-Eureka Prize and the 2015 UTS Vice-Chancellor's Medal for Exceptional Research.