# Artificial Neural Variability for Deep Learning: On Overfitting, Noise Memorization, and Catastrophic Forgetting

**Zeke Xie**
*xie@ms.k.u-tokyo.ac.jp*
*University of Tokyo, Bunkyo-ku, Tokyo 113-0333, Japan and RIKEN Center for AIP, Chuo-ku, Tokyo 103-0027, Japan*

**Fengxiang He**
*fengxiang.he@sydney.edu.au*
**Shaopeng Fu**
*shfu7008@sydney.edu.au*
*University of Sydney, Level 1, Chippendale NSW 2008, Australia*

**Issei Sato**
*sato@k.u-tokyo.ac.jp*
*University of Tokyo, Bunkyo-ku, Tokyo 113-0333, Japan, and RIKEN Center for AIP, Chuo-ku, Tokyo 103-0027, Japan*

**Dacheng Tao**
*Dacheng.Tao@uts.edu.au*
*University of Sydney, Level 1, Chippendale NSW 2008, Australi*

**Masashi Sugiyama**
*sugi@k.u-tokyo.ac.jp*
*RIKEN Center for AIP, Chuo-ku, Tokyo 103-0027, Japan, and University of Tokyo, Bunkyo-ku, Tokyo 113-0333, Japan*

**Deep learning is often criticized by two serious issues that rarely exist in natural nervous systems: overfitting and catastrophic forgetting. It can even memorize randomly labeled data, which has little knowledge behind the instance-label pairs. When a deep network continually learns over time by accommodating new tasks, it usually quickly overwrites the knowledge learned from previous tasks. Referred to as the *neural variability*, it is well known in neuroscience that human brain reactions exhibit substantial variability even in response to the same stimulus. This mechanism balances accuracy and plasticity/flexibility in the motor learning of natural nervous systems. Thus, it motivates us to design a similar mechanism, named *artificial neural variability* (ANV), that helps artificial neural networks learn some advantages from "natural" neural**

**networks. We rigorously prove that ANV plays as an implicit regularizer of the mutual information between the training data and the learned model. This result theoretically guarantees ANV a strictly improved generalizability, robustness to label noise, and robustness to catastrophic forgetting. We then devise a** *neural variable risk minimization* **(NVRM) framework and** *neural variable optimizers* **to achieve ANV for conventional network architectures in practice. The empirical studies demonstrate that NVRM can effectively relieve overfitting, label noise memorization, and catastrophic forgetting at negligible costs.**

## 1 Introduction

Inspired by natural neural networks, artificial neural networks have achieved comparable performance with humans in a variety of application domains (LeCun, Bengio, & Hinton, 2015; Witten, Frank, Hall, & Pal, 2016; Silver et al., 2016; He, Zhang, Ren, & Sun, 2016; Litjens et al., 2017). Deep neural networks are usually highly overparameterized (Keskar, Mudigere, Nocedal, Smelyanskiy, & Tang, 2017; Dinh, Pascanu, Bengio, & Bengio, 2017; Arpit et al., 2017; Kawaguchi, Huang, & Kaelbling, 2019); the number of weights is usually way larger than the sample size. The extreme overparameterization gives deep neural network excellent approximation (Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe, & White, 1989; Hornik, 1993) and optimization (Allen-Zhu, Li, & Song, 2019; Arora, Cohen, & Hazan, 2018; Li & Liang, 2018; Allen-Zhu, Li, & Liang, 2019) abilities, as well as a prohibitively large hypothesis capacity. This phenomenon makes almost all capacity-based generalization bounds vacuous. Besides, former empirical results demonstrate that deep neural networks almost surely achieve zero training error even when the training data are randomly labeled (Zhang, Bengio, Hardt, Recht, & Vinyals, 2017). This memorization of noise suggests that deep learning is good at overfitting.

Deep learning performs poorly at learning multiple tasks from dynamic data distributions (Parisi, Kemker, Part, Kanan, & Wermter, 2019). The functionality of artificial neural networks is sensitive to weight perturbations. Thus, continually learning new tasks can quickly overwrite the knowledge learned through previous tasks, which is called catastrophic forgetting (McCloskey & Cohen, 1989; Goodfellow, Mirza, Xiao, Courville, & Bengio, 2013). Neuroscience has motivated a few algorithms for overcoming catastrophic forgetting and variations in data distributions (Kirkpatrick et al., 2017; Zenke, Poole, & Ganguli, 2017; Chen, Mai, Xiao, & Zhang, 2019).

Natural neural networks have much better generalizability and robustness. Can we learn from human brains again for more innovations in deep learning? An extensive body of work in neuroscience suggests that

neural variability is essential for learning and proper development of human brains, which refers to the mechanism that human brain reactions exhibit substantial variability even in response to the same stimulus (Churchland, Byron, Ryu, Santhanam, & Shenoy, 2006; Churchland et al., 2010; Dinstein, Heeger, & Behrmann, 2015). Neural variability acts as a central role in motor learning, which helps balance the need for accuracy and the need for plasticity and flexibility (Fetters, 2010). The ever-changing environment requires performers to constantly adapt to both external (e.g., slippery surface) and internal (e.g., injured muscle) perturbations. It is also suggested that adult motor control systems can perform better by generating neural variability actively in order to leave room for adaptive plasticity and flexibility (Tumer & Brainard, 2007). An appropriate degree of neural variability is necessary to studies of early development (Hedden & Gabrieli, 2004; Ölveczky, Otchy, Goldberg, Aronov, & Fee, 2011). A study on Parkinsons disease suggests that the learning ability to for new movements and adaptablity to perturbations is dramatically reduced when neural variability is low (Mongeon, Blanchet, & Messier, 2013).

Inspired by the neuroscience knowledge, this letter formulates artificial neural variability theory for deep learning. We mathematically prove that ANV plays the role of an implicit regularizer of the mutual information between learned model weights and the training data. A beautiful coincidence in neuroscience is that neural variability in the rate of response to a steady stimulus also penalizes the information carried by nerve impulses (spikes) (Stein, Gossen, & Jones, 2005; Houghton, 2019). Our theoretical analysis guarantees that ANV can strictly relieve overfitting, label noise memorization, and catastrophic forgetting.

We further propose a neural variable risk minimization (NVRM) framework, which is an efficient training method to achieve ANV for artificial neural networks. In the NVRM framework, we introduce weight perturbations during inference to simulate the neural variability of human brains to relieve overfitting and catastrophic forgetting. The empirical mean of the loss in the presence of weight perturbations is referred to as *neural variable risk* (NVR). Similar to neural variability, replacing the conventional empirical risk minimization (ERM) by NVRM would balance the accuracy-plasticity trade-off in deep learning.

The rest of this letter is organized as follows. In section 2, we propose the neural variability theory and mathematically validate that ANV relieves overfitting, label noise memorization, and catastrophic forgetting. In section 3, we propose the NVRM framework and neural variable optimizers, which can achieve ANV efficiently in practice. In section 4, we conduct extensive experiments to validate the theoretical advantages of NVRM. In particular, training neural networks via neural variable optimizers can easily achieve remarkable robustness to label noise and weight perturbation. In section 5, we conclude our main contribution.

## 2 Neural Variability Theory

In this section, we formally introduce artificial neural variability into deep learning. We denote a model with the weights $\theta$ as $\mathcal{M}(\theta)$ and the training data set as $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ drawn from the data distribution $\mathcal{S}$. We define the empirical risk over the training data set $S$ as $\hat{L}(\theta) = L(\theta, S) = \frac{1}{m} \sum_{i=1}^{m} L(\theta, (x^{(i)}, y^{(i)}))$, and the population risk over the data distribution $\mathcal{S}$ as $L(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{S}}[L(\theta, (x, y))]$. We formally define $(b, \delta)$-neural variability ($(b, \delta)$-NV) as definition 1.

**Definition 1** (*Neural Variability/Regional Flatness*). *Suppose $L(\theta, S)$ is the loss function for the model $\mathcal{M}(\theta)$ on the data set $S$, $\hat{\theta}$ obeys a gaussian distribution centered at $\theta$ as $\hat{\theta} \sim \mathcal{N}(\theta, b^2 I)$, and*

$$|\mathbb{E}_{\hat{\theta}\sim\mathcal{N}(\theta, b^2 I)}[L(\hat{\theta}, S)] - L(\theta, S)| \leq \delta,$$

*where $|\cdot|$ denotes the absolute value, and both $\delta$ and $b$ are positive. Then the model $\mathcal{M}(\theta)$ is said to achieve $(b, \delta)$-neural variability at $\theta$ on the data set $S$. It can also be said the model achieves $(b, \delta)$-regional flatness at $\theta$ on the data set $S$.*

The definition has a similar form to $(\mathcal{C}_\epsilon, A)$-sharpness defined by Keskar et al. (2017). A model $\mathcal{M}(\theta)$ with $(b, \delta)$-neural variability can work almost equally well when its weights are randomly perturbed as $\hat{\theta} \sim \mathcal{N}(\theta, b^2 I)$. This definition mimics the neuroscience mechanism that human brains can work well or even better by actively generating perturbations (Tumer & Brainard, 2007). The definition of $(b, \delta)$-neural variability is also a measure of robustness to weight perturbations and a measure of weight uncertainty for Bayesian neural networks.

**2.1 Generalization.** In this section, we formulate the information theoretical foundation of $(b, \delta)$-neural variability by using the PAC-Bayesian framework (McAllester, 1999a, 1999b). The PAC-Bayesian framework provides guarantees on the expected risk of a randomized predictor (hypothesis) that depends on the training data set. The hypothesis is drawn from a distribution $Q$ and sometimes referred to as a posterior. We then denote the expected risk with respect to the distribution $Q$ as $L(Q)$ and the empirical risk with respect to the distribution $Q$ as $\hat{L}(Q)$. Suppose $P$ is the prior distribution over the weight space $\Theta$.

**Lemma 1** (The PAC-Bayesian Generalization Bound (McAllester, 1999b)). *For any real $\Delta \in (0, 1)$, with probability at least $1 - \Delta$, over the draw of the training data set $S$, the expected risk for all distributions $Q$ satisfies*

$$L(Q) \leq \hat{L}(Q) + 4\sqrt{\frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\left(\frac{2m}{\Delta}\right)\right]},$$

*where $\mathrm{KL}(Q\|P)$ denotes the Kullback "Leibler divergence from $P$ to $Q$.*

The PAC-Bayesian generalization bound closely depends on the prior $P$ over the model weights. We make a mild assumption 1.

**Assumption 1.** *The prior over model weights is gaussian, $P = \mathcal{N}(0, \sigma^2 I)$.*

Assumption 1 justified as it can be interpreted as weight decay, which is widely used in related papers (Graves, 2011; Neyshabur, Bhojanapalli, McAllester, & Srebro, 2017; He, Liu, & Tao, 2019). We note that $\sigma^2$ is very large in practice, as $\sigma^2$ is equal to the inverse weight decay strength.

We consider a distribution $Q_{nv}$ over model weights of the form $\theta + \epsilon$, where $\theta$ is drawn from the distribution $Q$ and $\epsilon \sim \mathcal{N}(0, b^2 I)$ is a random variable. Following the theoretical analysis, particularly equation 7 of Neyshabur et al. (2017), we formulate theorem 1.

**Theorem 1** (The Generalization Advantage of ANV). *Suppose the model $\mathcal{M}(\theta^\star)$ achieves $(b, \delta)$-neural variability at $\theta^\star$, and assumption 1 holds. Then for any real $\Delta \in (0, 1)$, with probability at least $1 - \Delta$, over the draw of the training data set $S$, the expected risk for all distributions $Q_{nv}$ satisfies*

$$L(Q_{nv}) \leq \hat{L}(\theta^\star) + 4\sqrt{\frac{1}{m}\left[\mathrm{KL}(Q_{nv}\|P) + \ln\left(\frac{2m}{\Delta}\right)\right]} + \delta,$$

*where $N$ is the number of model weights and $\mathrm{KL}(Q_{nv}\|P) = \sum_{i=1}^{N}\left[\log\left(\frac{\sigma}{b}\right) + \frac{b^2 + \theta_i^{\star 2}}{2\sigma^2} - \frac{1}{2}\right]$.*

We leave all proofs for appendix A. We note that $\mathrm{KL}(Q_{nv}\|P)$ as the function of $b$ decreases with $b$ for $b \in (0, \sigma)$, and reaches the global minimum at $b = \sigma$. As $\sigma$ is much larger than 1 and $b$ in practice, the PAC-Bayesian bound monotonically decreases with $b$ given $\delta$. The bound is tighter than the bound in lemma 1 when the model has strong ANV, which means $b$ is large given a small $\delta$.

It is known that the information in the model weights relates to overfitting (Hinton & Van Camp, 1993) and flat minima (Hochreiter & Schmidhuber, 1997). Achille, Paolini, and Soatto (2019) argued that the information in the weights controls the PAC-Bayesian bound. We show that the generalization bound in theorem 1 positively correlates with the mutual information of learned model weights and training data. Given two random variables $\theta$ and $S$, their Shannon mutual information is defined as $I(\theta; S) = \mathbb{E}_{S \sim \mathcal{S}}[\mathrm{KL}(p(\theta|S)\|p(\theta))]$, which is the expected Kullback-Leibler divergence from the prior distribution $p(\theta)$ of $\theta$ to the distribution $p(\theta|S)$ after an observation of $S$ (Cover & Thomas, 2012). In the case of theorem 1, we have

$$\mathbb{E}_{S \sim \mathcal{S}}[\mathrm{KL}(Q_{nv}\|P)] = I(\theta; S), \tag{2.1}$$

where $\theta \sim Q_{nv}$. It indicates that penalizing the mutual information of the learned model weights and training data, $I(\theta; S)$, is equivalent to decreasing the expected $\text{KL}(Q_{nv}\|P)$, which may improve generalization. As $S \to \theta \to \theta + \epsilon$ is a Markov process, we have the data processing mutual information inequality $I(\theta + \epsilon; S) < I(\theta; S)$. It indicates that ANV regularizes the mutual information between the learned model weights and training data. This theoretical evidence is quite close to the neuroscience mechanism of penalizing the information carried by nerve impulses (Stein et al., 2005).

Different from the PAC-Bayesian approach, another theoretical framework for the generalization bound based on mutual information was proposed by Xu and Raginsky (2017). Following these authors, we formulate an alternative mutual-information-based generalization bound in appendix B.

**2.2 Robustness to Label Noise.** Noisy labels can remarkably damage the generalization of deep networks, because deep networks can completely memorize corrupted label information (Zhang et al., 2017). Memorizing noisy labels is one of the most serious overfitting issues in deep learning. We will show that ANV relieves deep networks from memorizing noisy labels by penalizing the mutual information of the model weights $\theta$ and the labels $y$ conditioned on the inputs $x$.

In section 4 of Achille and Soatto (2018), the expected cross-entropy loss can be decomposed into several terms to describe it. If the data distribution $\mathcal{S}$ is fixed, the expected cross-entropy loss for the training performance can be decomposed into three terms:

$$\mathcal{H}_f(y|x, \theta) = \mathbb{E}_S \mathbb{E}_{\theta \sim Q(\theta|S)} \left[ \sum_{i=1}^{m} -\log f(y^{(i)}|x^{(i)}, \theta) \right] \tag{2.2}$$

$$= \mathcal{H}(y|x) + \mathbb{E}_{x, \theta \sim Q(\theta|S)} \text{KL}[p(y|x)\| f(y|x, \theta)] - I(\theta; y|x), \tag{2.3}$$

where $f$ denotes the model's map from an input $x$ to a class distribution and $\mathcal{H}(\cdot)$ denotes the entropy $\mathbb{E}[-\log(\cdot)]$. The meaning of each term has been interpreted by Achille and Soatto (2018) in detail. The first term relates to the intrinsic error that we would commit in predicting the labels even if we knew the underlying data distribution. The second term relates to the efficiency of the model and the class of functions $f$ with respect to which the loss is optimized. Here we focus on the last term: the label memorization can be given by the mutual information between the model weights and the labels conditioned on inputs, namely, $I(\theta; y|x)$. In the traditional paradigm of deep learning, minimizing $-I(\theta; y|x)$ is expected. Thus, deep learning easily overfits noisy labels. Noisy labels as outliers of the data distribution imply a positive value of $I(\theta; y|x)$, which requires more information to be memorized. We need to reduce $I(\theta; y|x)$ effectively to prevent deep networks

from overfitting noisy labels. With this approach, theorem 2.1 of Harutyunyan, Reing, Steeg, and Galstyan (2020) also supported that memorization of noisy labels is prevented by decreasing $I(\theta; y|x)$.

Suppose a model $\mathcal{M}(\theta)$ achieves $(b, \delta)$-NV, $\delta$ is small, and we inject weight noise $\epsilon$ to this model. We have a Markov process as $y|x \rightarrow \theta \rightarrow \hat{\theta}$, where we denote $\theta + \epsilon$ as $\hat{\theta}$. Based on equation 2.2, we have

$$\mathcal{H}_f(y|x, \hat{\theta}) = \mathcal{H}(y|x) + \mathbb{E}_{x,\theta \sim Q_{nv}(\theta|S)} \text{KL}[p(y|x)\|f(y|x, \theta)] - I(\hat{\theta}; y|x), \quad (2.4)$$

According to definition 1, the model $\mathcal{M}(\hat{\theta})$ may achieve nearly equal training performance to $\mathcal{M}(\theta)$ given small $\delta$. At the same time, obviously, $I(\hat{\theta}; y|x)$ is smaller than $I(\theta; y|x)$ as $\epsilon$ penalizes the mutual information. This suggests that increasing $b$ given $\delta$ for a $(b, \delta)$-neural variable model can penalize the memorization of noisy labels by regularizing the mutual information of learned model weights and training data.

**2.3 Robustness to Catastrophic Forgetting.** The ability to continually learn over time by accommodating new tasks while retaining previously learned tasks is referred to as continual or lifelong learning (Parisi et al., 2019). However, the main issue of continual learning is that artificial neural networks are prone to catastrophic forgetting. In natural neural systems, neural variability leaves room for the excellent plasticity and continual learning ability (Tumer & Brainard, 2007). It is natural to conjecture that ANV can help relieve catastrophic forgetting and enhance continual learning.

We take regularization-based continual learning (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi, Babiloni, Elhoseiny, Rohrbach, & Tuytelaars, 2018; Doan, Bennani, Mazoure, Rabusseau, & Alquier, 2020) as an example. The basic idea of this learning is to strongly regularize weights most relevant to previous tasks. Usually the regularization is strong enough to fix learning near the solution learned from previous tasks. In a way, the model tends to learn in the overlapping region of optimal solutions for multiple tasks. The intuition behind ANV is clear: if a model is more robust to weight perturbation, it will have a wider optimal region shared by multiple tasks.

Suppose a model $\mathcal{M}(\theta)$ continually learns tasks A and B, where the learned solutions are, respectively, $\theta_A^\star \sim Q_A$ and $\theta_B^\star \sim Q_B$. The distribution $Q_B$ describes model weights of the form $\theta_A^\star + \epsilon$, where $\epsilon$ is a random variable. For any real $\Delta \in (0, 1)$, with probability at least $1 - \Delta$, over the draw of the training data set $S$ for task A, the expected risk for all distributions $Q_B$ satisfies

$$L(Q_B) \leq \hat{L}(Q_B) + 4\sqrt{\frac{1}{m}\left[\text{KL}(Q_B\|P) + \ln\left(\frac{2m}{\Delta}\right)\right]}. \quad (2.5)$$

For $\theta_B^\star = \theta_A^\star + \epsilon$, as $S \to \theta_A^\star \to \theta_B^\star$ is a Markov process, and the weight perturbation $\epsilon$ is learned from the training data set $S_B$ (of task B) only, the data processing mutual information inequality $I(\theta_B^\star; S) < I(\theta_A^\star; S)$ still holds. We recall the mutual information analysis above and have $I(\theta_B^\star; S) = \mathbb{E}_{S \sim S}[\mathrm{KL}(Q_B \| P)]$. Thus, we have

$$L(Q_B) - \hat{L}(Q_B) = [L(Q_B) - \hat{L}(Q_A)] - [\hat{L}(Q_B) - \hat{L}(Q_A)] \qquad (2.6)$$

$$< 4\sqrt{\frac{1}{m}\left[\mathrm{KL}(Q_B \| P) + \ln\left(\frac{2m}{\Delta}\right)\right]}. \qquad (2.7)$$

Considering the expectation with respect to the data distribution, we obtain

$$E_S[(L(Q_B) - \hat{L}(Q_A) - \delta_{AB})^2] < \frac{16}{m}\left[I(\theta_B^\star; S) + \ln\left(\frac{2m}{\Delta}\right)\right], \qquad (2.8)$$

where $\delta_{AB} = \hat{L}(Q_B) - \hat{L}(Q_A)$ and $I(\theta_B^\star; S) < I(\theta_A^\star; S)$. So the population risk $L(Q_B)$ (for task A) can be well bounded by the empirical risk increasing $\delta_{AB}$ and the mutual information $I(\theta_A^\star; S)$. Here $\delta_{AB}$, the empirical risk increasing due to weight perturbation, is a kind of measure of robustness to weight perturbation. It suggests that increasing robustness to weight perturbation can help relieve catastrophic forgetting.

## 3 Neural Variable Risk Minimization

In this section, we aim to learn empirical minimizers with ANV, given a conventional network architecture. Our method is to minimize the empirical risk in a certain region rather than the empirical loss at a single point,

$$L_{\mathrm{NV}}(\theta, S) = \mathbb{E}_{\hat{\theta} \sim \mathcal{N}(\theta, b^2 I)}[L(\hat{\theta}, S)], \qquad (3.1)$$

where $b$ is the variability hyperparameter. We call the risk the neural variable risk (NVR), and call optimizing the NVR neural variable risk minimization (NVRM). The model $\mathcal{M}(\theta^\star)$ learned by NVRM can naturally achieve $(b, \delta)$-neural variability, where $\delta = |L_{\mathrm{NV}}(\theta^\star, S) - L(\theta^\star, S)|$. In this letter, we usually let $\epsilon$ obey a gaussian distribution, because gaussian noise is the noise type that penalizes information most effectively given a certain variance. But it is easy to generalize our framework to other noise types, such as Laplace noise and uniform noise. The noise type can be regarded as a hyperparameter.

The next question is, How to perform NVRM? Unfortunately, NVR is intractable in practice. But it is possible to approximately estimate the NVR and its gradient by sampling $\hat{L}_{\mathrm{NV}}(\theta, S) = L(\theta + \epsilon, S)$, where $\epsilon \sim \mathcal{N}(0, b^2 I)$. This unbiased estimation method is also used in variational inference (Graves, 2011).

We propose a class of novel optimization algorithms to employ NVRM in practice. We can write NVRM update as

$$\theta_t = \theta_{t-1} - \eta \frac{\partial L(\theta_{t-1} + \epsilon_{t-1}, (x, y))}{\partial \theta}. \tag{3.2}$$

NVRM is mimicking the neural variability of human brains in response to the same stimulus. NVRM exhibits the variability of predictions and back-propagation even in response to the same inputs. This update cannot be implemented inside an optimizer. But if we introduce $\hat{\theta}_t = \theta_t + \epsilon_t$ into the NVRM update, then we obtain a novel updating rule:

$$\hat{\theta}_t = \hat{\theta}_{t-1} - \eta \frac{\partial L(\hat{\theta}_{t-1}, (x, y))}{\partial \hat{\theta}} + \epsilon_t - \epsilon_{t-1}. \tag{3.3}$$

We can combine this updating rule with popular optimizers, such as SGD (Bottou, 1998; Sutskever, Martens, Dahl, & Hinton, 2013); then we easily get a class of novel optimization algorithms, such as NVRM-SGD. We call this class of optimization algorithms *neural variable optimizers*. The pseudocode of NVRM-SGD is displayed in algorithm 1. Similarly, we can also easily obtain NVRM-Adam by adding the four colored lines of algorithm 1 into adaptive momentum estimation (Adam) (Kingma & Ba, 2014). The source code is available at https://github.com/zeke-xie/artificial-neural-variability-for-deep-learning. We note that it is necessary to apply the denoising step before we evaluate the model learned by NVRM on test data sets, as $\hat{\theta}_t = \theta_t + \epsilon_t$. We also call such weight perturbations virtual perturbations, which need to be applied before inference and removed after backpropagation. We can easily empower neural networks with ANV by importing a neural variable optimizer to train them.

**3.1 A Deep Learning Dynamical Perspective.** We note that it is possible to theoretically analyze NVRM from a deep learning dynamical perspective. NVRM actually introduces Hessian-dependent gradient noise into learning dynamics instead of injected white gaussian noise in conventional noise injection methods, as the second-order Taylor approximation $\nabla L(\theta + \epsilon) \approx \nabla L(\theta) + \nabla^2 L(\theta)\epsilon$ holds for small weight perturbation. Zhu, Wu, Yu, Wu, and Ma (2019) argued that anisotropic gradient noise is often beneficial for escaping sharp minima. Xie, Sato, and Sugiyama (2021) and Xie, Wang, Zhang, Sato, and Sugiyama (2020) further quantitatively proved that Hessian-dependent gradient noise is exponentially helpful for learning flat minima. Again, flat minima (Hochreiter & Schmidhuber, 1997) are closely related to overfitting and the information in the model weights. This can mathematically explain the advantage of NVRM from a different perspective. We leave the diffusion-based approach as future work.

---

**Algorithm 1:** NVRM-SGD.

---

**Input:** Training data $S$, the variability hyperparameter $b$, the number of iterations $T$,

learning rate $\eta$, initialized weights $\theta_0$

**Output:** The model weights $\theta$

$\epsilon_0 = 0$;

**repeat**

    **for** $(x, y) \in S$ **do**

        $\epsilon_t \sim \mathcal{N}(0, b^2 I)$;

        $g_t = \frac{\partial L(\theta_{t-1}, (x,y))}{\partial \theta}$;

        $\theta_t = \theta_{t-1} - \eta g_t$;

        $\theta_t = \theta_t + \epsilon_t - \epsilon_{t-1}$;

    **end**

**until** *stopping criterion is not met*;

$\theta_T = \theta_T - \epsilon_T$

---

**3.2 Related Work.** One related line of research is injecting weight noise into deep networks during training (An, 1996; Neelakantan et al., 2015; Zhou, Liu, Li, Lin, Zhou, & Zhao, 2019). For example, perturbed stochastic gradient descent (PSGD) is SGD with a conventional weight noise injection method, which is displayed in algorithm 2. Another famous example is stochastic gradient langevin dynamics (SGLD) (Welling & Teh, 2011), which differs from PSGD only in the magnitude of injected gaussian noise. However, this conventional line does not remove the injected weight noise after each iteration, which makes it essentially different from our method. In section 4, we empirically verify that the denoising step is significantly helpful for preventing overfitting.

Variational inference (VI) for Bayesian neural networks (Graves, 2011; Blundell, Cornebise, Kavukcuoglu, & Wierstra, 2015; Khan et al., 2018) aims at estimating the posterior distribution of model weights given training data. VI requires expensive costs to update the posterior distribution (model uncertainty) during training. This line believes estimating the exact posterior is important but ignores the importance of enhancing model certainty. In contrast, our method is the first to actively encourage model

uncertainty for multiple benefits by choosing the variability hyperparameter $b$. ANV may be regarded as applying a neuroscience-inspired hyperprior over model uncertainty. Inspired by recent work on Bayesian neural networks, we conjecture that the NVRM framework could help improve adversarial robustness (Carbone et al., 2020) and fix overconfidence problems (Kristiadi, Hein, & Hennig, 2020).

Another related line of research is randomized smoothing (Duchi, Bartlett, & Wainwright, 2012; Nesterov & Spokoiny, 2017). Wen et al. (2018) applied the idea of randomized smoothing in training of deep networks and proposed the so-called SmoothOut method to optimize a weight-perturbed loss. This is also what the proposed NVRM does. We note that the original SmoothOut is actually a different implementation of NVRM with uniform noise, which both belong to Randomized Smoothing. However, this line of research (Duchi et al., 2012; Wen et al., 2018) focused only on improving performance on clean data sets by escaping from sharp minima. To the best of our knowledge, our work is the first along this line to theoretically and empirically analyze label noise memorization and catastrophic forgetting.

In summary, our letter further made two important contributions beyond the existing related work: (1) we discovered that NVRM can play a very important role in regularizing mutual information, which helps relieve label noise memorization and catastrophic forgetting, and (2) we implemented the weight-perturbed gradient estimation as a simple and effective optimization framework. NVRM as an optimizer is more elegant and easier to use than the existing methods like SmoothOut, which need to update the weights to calculate a perturbed loss before each back propagation.

## 4 Empirical Analysis

We conducted systematic comparison experiments to evaluate the proposed NVRM framework. To secure a fair comparison, every experimental setting was repeatedly trialed 10 times while all irrelevant variables were strictly controlled. We evaluated NVRM by the mean performance and the standard deviations over 10 trials. Implementation details are in appendix C.

**4.1 Robustness to Weight Perturbation.** For ResNet-34 (He et al., 2016) trained on clean data, CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), we perturbed the weights by isotropic gaussian noise of different noise scales to evaluate the test accuracy to weight perturbation. Figure 1 demonstrates that the models trained by NVRM are significantly more robust to weight perturbation and have lower expected minima sharpness defined by Neyshabur et al. (2017). This empirically verifies that the conventional neural networks trained via NVRM indeed learn strong ANV.
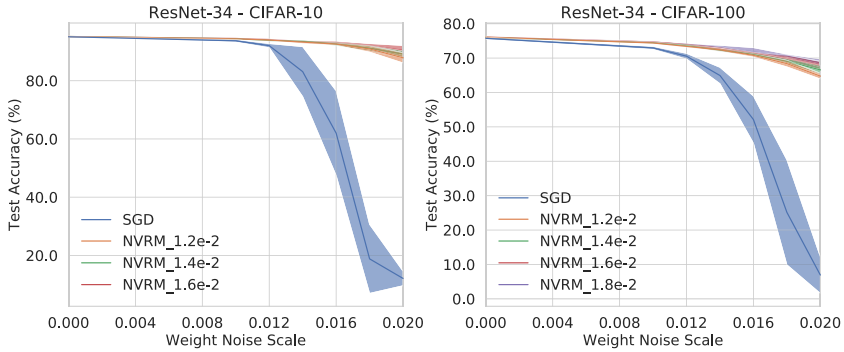
Figure 1: Curves of test accuracy to weight noise scale. The NVRM-trained network can almost retain reasonably well performance, while the SGD-trained network has nearly lost all learned knowledge due to relatively large weight noise.
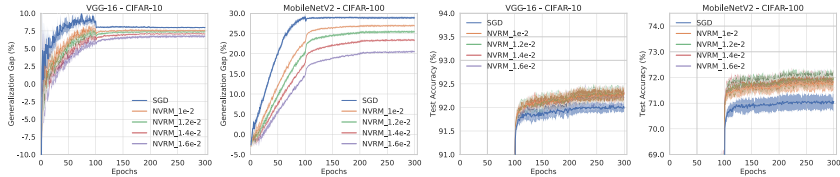


Figure 2: NVRM with various variability scales $b$ can consistently improve generalization. Left two panels: Curves of generalization gap. Right two panels: Curves of test accuracy. We train VGG-16 on CIFAR-10 and MobileNetV2 on CIFAR-100. More results of VGG-16 on CIFAR-100 and MobileNetV2 on CIFAR-10 are in appendix E.

**4.2 Improved Generalization.** Model: VGG-16 (Simonyan & Zisserman, 2015) and MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). Data set: CIFAR-10 and CIFAR-100. We evaluated the test accuracy and the generalization gap, which is defined as the difference between the training accuracy and the test accuracy. The results in Figure 2 clearly demonstrate that NVRM can significantly narrow the generalization gap while slightly improving the test accuracy, which was also supported by Nesterov and Spokoiny (2017) and Wen et al. (2018).

**4.3 Robustness to Noisy Labels.** Model: ResNet-34. Data set: CIFAR-10 and CIFAR-100. We inserted two classes of label noise into the data sets: the uniform flip label noise (symmetric label noise) and the pair-wise flip label noise (asymmetric label noise). Figure 3 demonstrates that SGD seriously overfits noisy labels; meanwhile, NVRM can avoid memorizing noisy labels
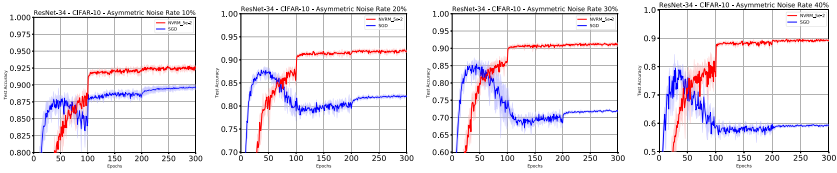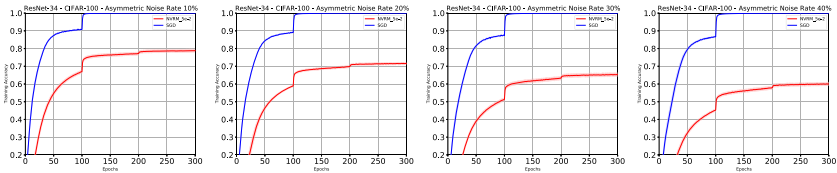
(a) Test accuracy to epochs of ResNet-34 on CIFAR-10 with noisy labels.
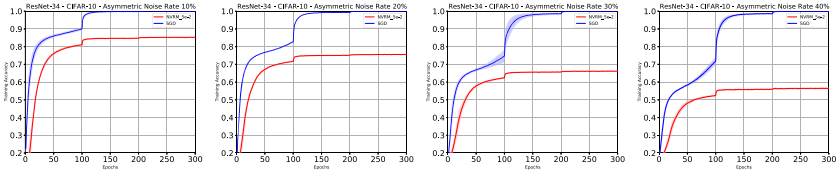


(b) Training accuracy to epochs of ResNet-34 on CIFAR-10 with noisy labels.
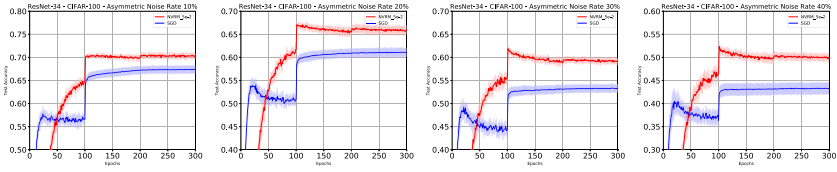


(c) Test accuracy to epochs of ResNet-34 on CIFAR-100 with noisy labels.



(d) Training accuracy to epochs of ResNet-34 on CIFAR-100 with noisy labels.

Figure 3: Curves of test accuracy to epochs of ResNet-34. NVRM with default $b = 0.05$ can significantly relieve memorizing noisy labels. Particularly, NVMR stops learning when the training error is close to the label noise rate. SGD almost memorizes all noisy labels, while NVRM almost only learns clean labels. The four columns are for asymmetric label noise rate 10%, 20%, 30%, and 40%, respectively.

effectively. The experimental results of symmetric label noise in appendix E also support our theory.

**4.4 Robustness to Catastrophic Forgetting.** Model: Three-layer fully-connected network (FCN). Data set: Permuted MNIST (LeCun, 1998). Continual learning setting: FCN continually learns five tasks, and we made different random pixels permutation for each task. We evaluated the
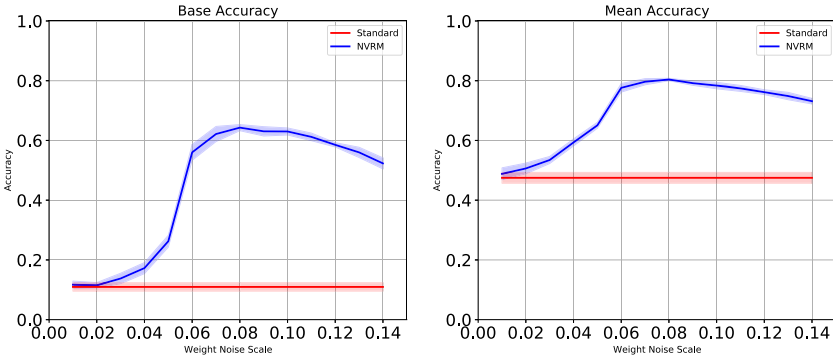
Figure 4: NVRM prevents catastrophic forgetting effectively with various variability scales $b$ (weight noise scales). Left: The accuracy of the first task after continually learning five tasks. Right: The mean accuracy of all five tasks after continually learning five tasks.
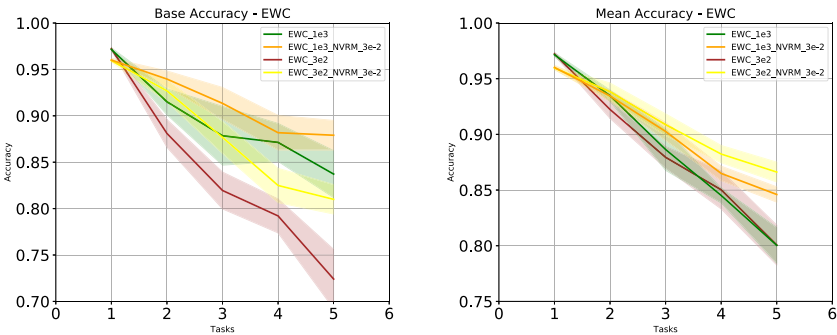


Figure 5: Curves of test accuracy to the number of tasks in continually learning Permuted MNIST. Left: The accuracy of the base task with EWC. Right: The mean accuracy of all learned tasks with EWC. The importance hyperparameter of EWC is set to 300 and 1000. NVRM enhances EWC effectively.

accuracy of the base task (the first task) and the mean accuracy of all learned tasks after each task. Figure 4 shows that NVRM forgets the knowledge learned from the previous task much more slowly than standard empirical risk minimization. The empirical results demonstrate that the models learned under NVRM framework are significantly more robust to catastrophic forgetting. In Figure 5, we also verified that NVRM can enhance a popular neuroscience-inspired continual learning method, elastic weight consolidation (EWC; Kirkpatrick et al., 2017). We present the incremental class learning task in appendix C.
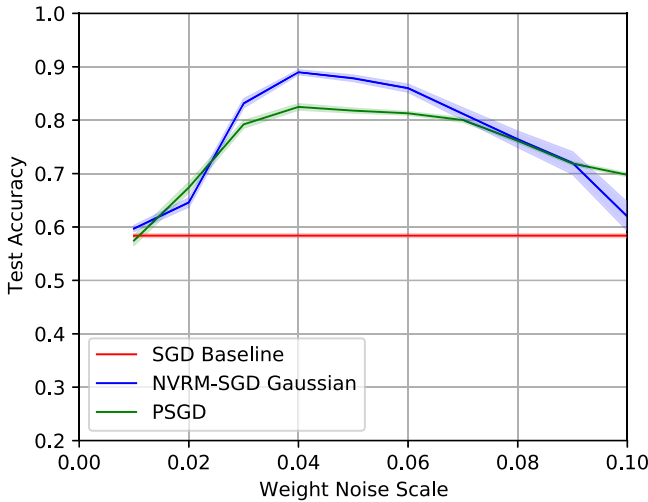
Figure 6: The denoising step is helpful for preventing overfitting. Data set: CIAF-10 with 40% label noise. While we are the first to report that PSGD may prevent memorizing noisy labels much better than SGD, NVRM can still outperform PSGD significantly by nearly seven points.

**4.5 Is the Denoising Step Really Helpful?** We empirically compared the NVRM approach with PSGD, which uses a conventional noise injection method, on label noise memorization. We display the test errors of training ResNet34 on CIFAR-10 with 40% asymmetric label noise under various variability scales/weight noise scales in Figure 6. The results demonstrate that, surprisingly, PSGD may prevent memorizing noisy labels much better than SGD, which has not been reported by existing work yet. However, NVRM can still outperform PSGD significantly for learning with noisy labels. Thus, the denoising step in NVRM is not theoretically reasonable but also empirically powerful.

**4.6 Choices of Noise Types.** We do not have to let $\epsilon$ be gaussian. We consider the noise type as a hyperparameter, and empirically compare three common noise types—gaussian noise, Laplace noise, and uniform noise—on CIAF-10 with noisy labels—because the tasks of learning with noisy labels can well reflect the ability to prevent overfitting. We display the test errors of training ResNet34 on CIFAR-10 under various variability scales $b$ with 40% asymmetric label noise in Figure 7. The result demonstrates that with a wide range of the variability hyperparameter $b$, NVRM with three noise types can achieve remarkable improvements over the baseline SGD. This is not surprising because any of this noise may theoretically regularize the mutual information. Note that NVRM uniform is identical to the
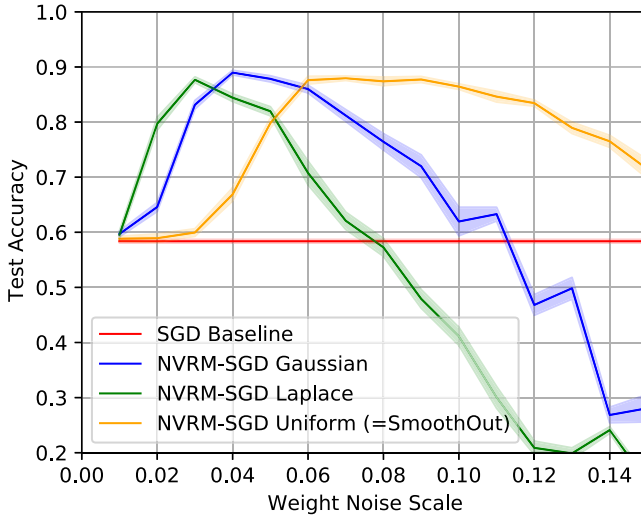
Figure 7: Choices of noise types for NVRM. Data set: CIAF-10 with 40% label noise. With a proper variability scale (weight noise scale) *b*, NVRM gaussian outperforms NVRM Uniform and NVRM Laplace by nearly one point, while NVRM Uniform is more robust to the variability scale.

original smoothout, which applies uniform smoothing to SGD. The original paper of smoothout (Wen et al., 2018) argued that uniform noise may be slightly better than gaussian noise on clean data sets. However, existing work (Duchi et al., 2012; Wen et al., 2018) did not discover the ability of randomized smoothing or smoothout to learn with noisy label. Figure 7 suggests that the conclusion of noise types is richer than Wen et al. (2018) expected, and there is no "free lunch." We discovered that the optimal test performance of NVRM gaussian is better than NVRM uniform and NVRM Laplace by nearly one point, while NVRM Uniform is more robust to the variability scale. It indicates that gaussian noise and uniform noise have different advantages.

## 5 Conclusion

A well-known term in neuroscience, neural variability, suggests that the human brain response to the same stimulus exhibits substantial variability and significantly contributes to balancing the accuracy and plasticity/flexibility in motor learning in natural neural networks. Inspired by this mechanism, this letter introduced ANV for balancing the accuracy and plasticity/flexibility in artificial neural networks. We proved that ANV acts as an implicit regularizer to control the mutual information between

the training data and the learned model, which further secures preventing the learned model from overfitting and catastrophic forgetting. These two abilities are theoretically related to the robustness to weight perturbations. The proposed NVRM framework is an efficient approach to achieving ANV for artificial neural networks. The empirical results demonstrate that our method can (1) enhance the robustness to weight perturbation, (2) improve generalizability, (3) relieve the memorization of noisy labels, and (4) mitigate catastrophic forgetting. Particularly, NVRM, an optimization approach, may handle memorization of noisy labels well at negligible computational and coding costs. One line code of importing a neural variable optimizer is all you need to achieve ANV for your models.

## Appendix A: Proofs

### A.1 Proof of Theorem 1.

**Proof.** We consider a distribution $Q_{nv}$ over predictors with weights of the form $\theta + \epsilon$, where $\theta$ is drawn from the distribution $Q$ and $\epsilon \sim \mathcal{N}(0, b^2 I)$ is a random variable indicating weight perturbation. Suppose the model $\mathcal{M}(\theta)$ achieves $(b, \delta(\theta))$-neural variability, which follows the notation of the expected sharpness used by Neyshabur et al. (2017). We start our theoretical analysis based on equation 7 of Neyshabur et al. (2017). We can bound the expected risk over the distribution $Q_{nv}$ as

$$L(Q_{nv}) \leq \hat{L}(Q) + [\hat{L}(Q_{NV}) - \hat{L}(Q)] + 4\sqrt{\frac{1}{m}\left[KL(Q_{nv}\|P) + \ln\left(\frac{2m}{\Delta}\right)\right]},$$

(A.1)

$$= \hat{L}(Q) + 4\sqrt{\frac{1}{m}\left[KL(Q_{nv}\|P) + \ln\left(\frac{2m}{\Delta}\right)\right]} + \mathbb{E}_{\theta \sim Q}[\delta(\theta)]. \quad (A.2)$$

We emphasize that this bound holds for any distribution $Q$ (any method of choosing $\theta$ dependent on the training data set) and any prior $P$. We use a very special distribution $Q:Pr(\theta = \theta^\star) = 1$. Thus we can bound the expected risk over the distribution $Q_{nv}$ as

$$L(Q_{nv}) \leq \hat{L}(\theta^\star) + 4\sqrt{\frac{1}{m}\left[KL(\theta^\star + \epsilon\|P) + \ln\left(\frac{2m}{\Delta}\right)\right]} + \delta(\theta^\star). \quad (A.3)$$

The Kullback-Leibler divergence of the two gaussians can be written as

$$KL(\theta^\star + \epsilon\|P) = \sum_{i=1}^{N}\left[\log\left(\frac{\sigma}{b}\right) + \frac{b^2 + \theta_i^{\star 2}}{2\sigma^2} - \frac{1}{2}\right], \quad (A.4)$$

where $N$ is the number of the model weights. Finally, we have

$$L(Q_{\text{nv}}) \leq \hat{L}(\theta^\star) + 4 \sqrt{\frac{1}{m} \left[ \sum_{i=1}^{N} \left[ \log\left(\frac{\sigma}{b}\right) + \frac{b^2 + \theta_i^{\star 2}}{2\sigma^2} - \frac{1}{2} \right] + \ln\left(\frac{2m}{\Delta}\right) \right]} + \delta.$$

(A.5)

$\square$

## Appendix B: The Mutual-Information Generalization Bound

We formulate a mutual information theoretical foundation of $(b, \delta)$-neural variability, which is more related to the neuroscience mechanism of penalizing the information carried by nerve impulses (Stein et al., 2005).

It is known that the information in the model weights relates to overfitting (Hinton & Van Camp, 1993) and flat minima (Hochreiter & Schmidhuber, 1997). According to lemma 2, if the mutual information of the parameters and data decreases, the upper bound of the expected generalization gap will also decrease.

**Lemma 2** (Xu & Raginsky, 2017). *Suppose $L(\theta, (x, y))$ is the loss function of the model $\mathcal{M}(\theta)$, such that $L(\theta, (x, y))$ is $\sigma$-sub-gaussian random variable for each $\theta$. Let the training data set $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ and the test sample $\bar{S} = (\bar{x}, \bar{y})$ be sampled from the data distribution $\mathcal{S}$ independently, and $\theta$ be the model weights learned from the algorithm $\mathcal{A}(\theta|S)$. Then the expected generalization gap meets the following property:*

$$\mathbb{E}\left[L(\theta, (\bar{x}, \bar{y})) - L(\theta, S)\right] \leq \sqrt{\frac{2\sigma^2}{m} I(\theta; S)},$$

*where $I(\theta; S)$ denotes the mutual information between the parameters $\theta$ and the training data set $S$.*

**Theorem 2.** *Suppose the conditions of lemma 2 hold, and the model $\mathcal{M}(\theta)$ achieves $(b, \delta)$-NV on the training data set $S$. Then the expected generalization gap of the model $\mathcal{M}(\theta)$ satisfies*

$$\mathbb{E}\left[L(\theta + \epsilon, (\bar{x}, \bar{y})) - L(\theta, S)\right] \leq \sqrt{\frac{2\sigma^2}{m} I(\theta + \epsilon; S)} + \delta,$$

*where $\epsilon \sim \mathcal{N}(0, b^2 I)$ is gaussian noise, and $\delta$ only depends on the training loss landscape.*

**Proof.** Given the model $\mathcal{M}(\theta)$, we can easily obtain a new model $\mathcal{M}(\theta + \epsilon)$ close to $\mathcal{M}(\theta)$ by injecting a gaussian noise $\epsilon \sim \mathcal{N}(0, b^2 I)$. By lemma 2, we have the expected generalization gap of $\mathcal{M}(\theta + \epsilon)$ meets

$$\mathbb{E}\left[L(\theta + \epsilon, (\bar{x}, \bar{y})) - \frac{1}{m}\sum_{i=1}^{m} L(\theta + \epsilon, (x^{(i)}, y^{(i)}))\right] \leq \sqrt{\frac{2\sigma^2}{m} I(\theta + \epsilon; S)}. \quad \text{(B.1)}$$

Based on the definition of $(b, \delta)$-NV, we have

$$\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} L(\theta + \epsilon, (x^{(i)}, y^{(i)})) - \frac{1}{m}\sum_{i=1}^{m} L(\theta, (x^{(i)}, y^{(i)}))\right] \leq \delta. \quad \text{(B.2)}$$

Thus, we obtain

$$\mathbb{E}\left[L(\theta + \epsilon, (\bar{x}, \bar{y})) - \frac{1}{m}\sum_{i=1}^{m} L(\theta, (x^{(i)}, y^{(i)}))\right] \leq \sqrt{\frac{2\sigma^2}{m} I(\theta + \epsilon; S)} + \delta. \quad \text{(B.3)}$$

$\square$

Obviously, the bound monotonically decreases with the variability scale $b$ given $\delta$. The bound is tighter than the bound in lemma 2 when the model has good ANV, which means $b$ is large given a small $\delta$. A large $b$ can even penalize the mutual information to nearly zero. Therefore, strong ANV brings a tighter generalization bound.

## Appendix C: Implementation Details

We introduce the details of each experiment in this section. In experiment 1, we evaluated NVRM's robustness to weight perturbation. In experiment 2, we evaluated the generalizability of NVRM. In experiment 3, we evaluated NVRM's robustness to noisy labels. In experiment 4, we evaluated NVRM's robustness to catastrophic forgetting. In experiment 5, we studied the usefulness of the de-noising step. In experiment 6, we studied choices of noise types.

Our experiment is conducted on a computing cluster with GPUs of NVIDIATesla V100 16 GB and CPUs of IntelXeon Gold 6140 CPU @ 2.30 GHz.

### C.1 Robustness, Generalization, and Label Noise.

*C.1.1 General Settings.* Experiments are conducted based on three popular deep learning networks VGG-16 (Simonyan & Zisserman, 2015), MobileNetV2 (Sandler et al., 2018), and ResNet-34 (He et al., 2016). The detailed architectures are presented in Table 1. Similarly, all data sets involved in our experiments are generated based on two standard benchmark data sets, CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009).[1]

---

[1]https://www.cs.toronto.edu/kriz/cifar.html.

Table 1: The Detailed Architectures of Models Used in the Experiments.

| VGG-16 | MobileNetV2 | ResNet-34 |
|---|---|---|
| conv3-64 × 2 | fc-32 | conv3-64 |
| maxpool | $\begin{bmatrix} \text{conv1-}k \\ \text{conv3-}k \\ \text{conv1-16} \end{bmatrix} \times 1$ | $\begin{bmatrix} \text{conv3-64} \\ \text{conv3-64} \end{bmatrix} \times 3$ |
| conv3-128 × 2 | $\begin{bmatrix} \text{conv1-6k} \\ \text{conv3-6k} \\ \text{conv1-24} \end{bmatrix} \times 2$ | $\begin{bmatrix} \text{conv3-128} \\ \text{conv3-128} \end{bmatrix} \times 4$ |
| maxpool | $\begin{pmatrix} \text{conv1-6k} \\ \text{conv3-6k} \\ \text{conv1-32} \end{pmatrix} \times 3$ | $\begin{bmatrix} \text{conv3-256} \\ \text{conv3-256} \end{bmatrix} \times 6$ |
| conv3-256 ×3 | $\begin{pmatrix} \text{conv1-6k} \\ \text{conv3-6k} \\ \text{conv1-64} \end{pmatrix} \times 4$ | $\begin{bmatrix} \text{conv3-512} \\ \text{conv3-512} \end{bmatrix} \times 3$ |
| maxpool | $\begin{bmatrix} \text{conv1-6k} \\ \text{conv3-6k} \\ \text{conv1-96} \end{bmatrix} \times 3$ | avgpool |
| conv3-512 × 3 | $\begin{pmatrix} \text{conv1-6k} \\ \text{conv3-6k} \\ \text{conv1-160} \end{pmatrix} \times 3$ | |
| maxpool | $\begin{bmatrix} \text{conv1-6k} \\ \text{conv3-6k} \\ \text{conv1-320} \end{bmatrix} \times 1$ | |
| conv3-512 × 3 | fc-1280 | |
| maxpool | | |
| fc-512 × 2 | | |
| | fc-10 or fc-100 | |

Notes: "conv $x$ - $c$" represents a convolution layer with kernel size $x \times x$ and $c$ output channels, and "fc - $c$" represents a fully connected layer with $c$ output channels. In the architecture of MobileNetV2, [·] represents a bottleneck, and (·) is simply a combination of three convolution layers but can halve both the width and height of the input of the block. The $k$ in [·] or (·) denotes the number of channels of the input of the corresponding block. In the architecture of ResNet-34, [·] represents a basic block.

We follow the official version to split training sets and test sets in our experiments. For preprocessing and data augmentation, we performed per pixel mean subtraction, horizontal random flip, and 32 × 32 random crops after padding with four pixels on each side. The batch size is set as 128, and the weight decay factor is set as 0.0001. We selected the optimal learning rate from {0.0001, 0.001, 0.01, 0.1, 1, 10} and used 0.1 for SGD/NVRM-SGD. Note that we used the common $L_2$ regularization as weight decay, which

is widely used in most cases, while Loshchilov and Hutter (2018) and Xie, Sato, and Sugiyama (2020) suggested that decoupled weight decay or stable weight decay is better in adaptive gradient methods. We employ SGD and NVRM-SGD to train models unless we specify otherwise. For the learning rate schedule, we initialized the learning rate as 0.1 and divided it by 10 after every 100 epochs. All models are trained for 300 epochs. The momentum factor is set as 0 for VGG-16 and MobileNetV2 in experiment 1, and 0.9 for ResNet-34 in experiments 2 and 3.

*C.1.2 Robustness to Weight Perturbation.* For experiment 1, we injected isotropic gaussian noise of different variances to all the model weights and then evaluated the changes of the test accuracy. Six noise scales {0.01, 0.012, 0.014, 0.016, 0.018, 0.02} are involved in our experiments.

*C.1.3 Learning with Noisy Labels.* For experiment 3, we also generate a group of data sets with label noise. The symmetric label noise is generated by flipping every label to other labels with uniform flip rates {20%, 40%, 60%, 80%}. The asymmetric label noise by flipping label $i$ to label $i + 1$ (except that label 9 is flipped to label 0) with pair-wise flip rates {10%, 20%, 30%, 40%}. We employed the code of Han et al. (2018) for generating noisy labels for CIFAR-10 and CIFAR-100.

*C.1.4 The Usefulness of the Denoising Step and Choices of Noise Types.* The hyperparameter settings of experiments 5 and 6 follow experiment 3, which are performed on learning with noisy labels. In experiment 6, we let the weight noise $\epsilon$, respectively, obey $\mathcal{N}(0, b^2)$, $Laplace(0, b)$, $Uniform(-b, b)$ for NVRM gaussian, NVRM Laplace, and NVRM uniform.

## C.2 Catastrophic Forgetting.

*C.2.1 Permuted MNIST.* For experiment 4, we used fully connected network (FCN), which has two hidden layers and 1024 ReLUs per hidden layer. As continual learning tasks usually employ adaptive optimizers, we compared Adam with NVRM-Adam on the popular benchmark task, Permuted MNIST. In Permuted MNIST, we have five continual tasks. For each task, we generated a fixed, random permutation by which the input pixels of all images would be shuffled. Each task was thus of equal difficulty to the original MNIST problem, though a different solution would be required for each. We evaluated the accuracy of the base task (the first task) and the mean accuracy of all learned tasks after each task.

In the EWC experiment, we try to validate if NVRM can improve EWC. We validated the performance improvements under two different importance hyperparameters $\lambda \in \{30, 1000\}$. In experiment 4, the batch size is set as 256, and the weight decay factor is set as 0.0001. As continual learning methods usually prefer adaptive optimizers, we employed Adam and NVRM-Adam as the optimizers. For the learning rate schedule, we fixed

the learning rate as 0.001 and applied no learning rate decay. We set the variability scale $b = 0.03$ in NVRM-Adam. All models are trained for one epoch per task, as one-epoch training has ensured good test performance on newly learned tasks.

*C.2.2 Split MNIST.* For experiment 4, we also supplied the experiment on split MNIST, another classical continual learning task. It is called incremental class learning. We train the models on the samples with a specific subset of labels for five continual tasks. We followed the usual setting (Zenke et al., 2017): $y \in \{0, 1\}$, $y \in \{2, 3\}$, $y \in \{4, 5\}$, $y \in \{6, 7\}$, and $y \in \{8, 9\}$ for five tasks, respectively. In each task, the model may learn only two new digits and may forget previously learned digits.

The model is the same as the model architecture for Permuted MNIST, except that we used the five-header output layers for five tasks, respectively. When we trained the models for one task, the headers for other tasks were frozen. The batch size was set as 256 and the weight decay factor as 0. Again, we employed Adam and NVRM-Adam as the optimizers, and new optimizers are used for each continual task. For the learning rate schedule, we fixed the learning rate as 0.001 and applied no learning rate decay. We also let the variability scale $b = 0.03$ in NVRM-Adam, unless we otherwise specify it.

## Appendix D: Additional Algorithm

---

**Algorithm 2:** Perturbed SGD.

**Input:** Training data $S$, the noise scale hyperparameter $b$, the number of iterations $T$,

learning rate $\eta$, initialized weights $\theta_0$

**Output:** The model weights $\theta$

**repeat**

    **for** $(x, y) \in S$ **do**

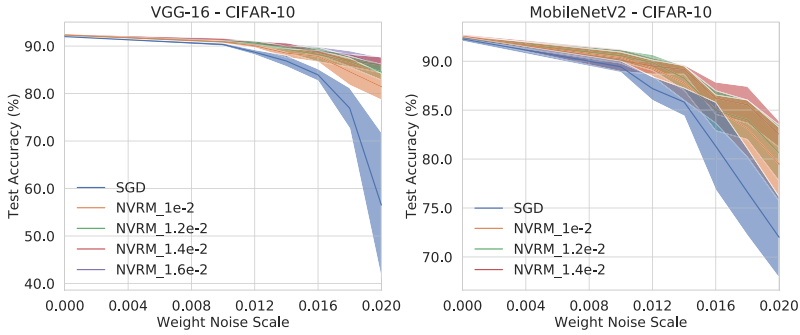        $\epsilon_t \sim \mathcal{N}(0, b^2 I)$;

        $g_t = \frac{\partial L(\theta_{t-1}, (x,y))}{\partial \theta}$;
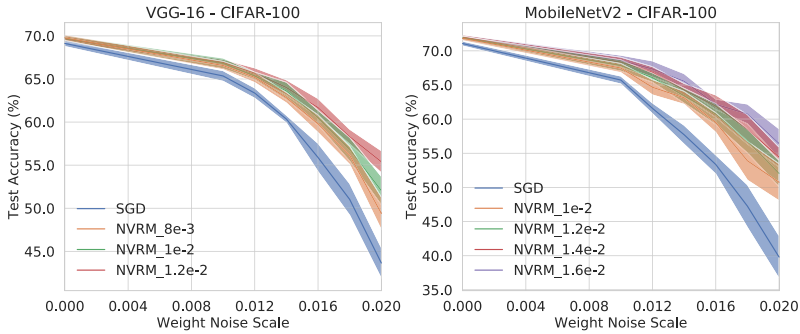
        $\theta_t = \theta_{t-1} - \eta(g_t + \epsilon_t)$;

    **end**

**until** *stopping criterion is not met*;

(a) Test accuracy to weight noise scale on CIFAR-10.



(b) Test accuracy to weight noise scale on CIFAR-100.

Figure 8: Curves of test accuracy to weight noise scale. Top: CIFAR-10; Bottom: CIFAR-100. The three columns are VGG-16 and MobileNetV2, respectively.

## Appendix E: Supplementary Experimental Results

**E.1 Robustness to Weight Perturbation.** See Figure 8. The empirical results demonstrate that NVRM can also make VGG-16 and MobileNetV2 more robust to weight perturbations. We also report that, obviously, the architecture of ResNet is much more optimal for achieving strong neural variability than VGG and MobileNet. We leave the network architecture study as future work.

**E.2 Improved Generalization.** See Figure 9.

**E.3 Robustness to Noisy Labels.** See Figure 10. SGD almost memorizes all corrupted labels. The results demonstrate that NVRM can also significantly improve the robustness to symmetric label noise.
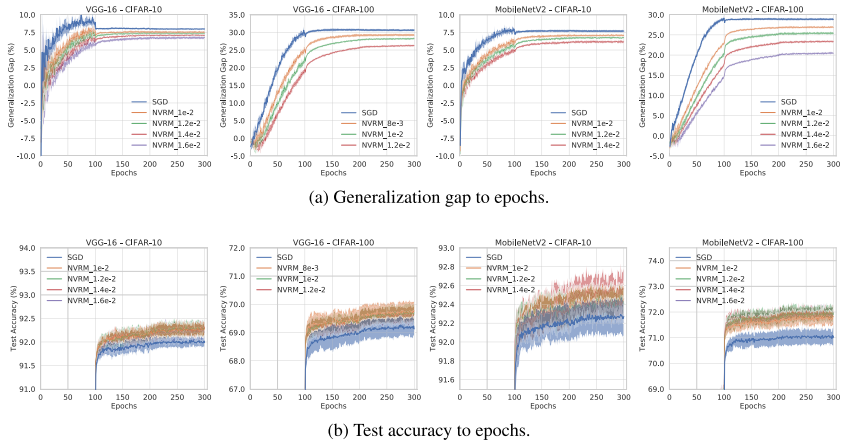
(a) Generalization gap to epochs.



(b) Test accuracy to epochs.

Figure 9: Curves of generalization gap and test accuracy to epochs. NVRM with various variability scales $b$ can consistently improve generalization. The four columns are, respectively, for (1) VGG-16 on CIFAR-10, (2) VGG-16 on CIFAR-100, (3) MobileNetV2 on CIFAR-10, and (4) MobileNetV2 on CIFAR-100.



(a) Test accuracy to epochs of ResNet-34 on CIFAR-10.



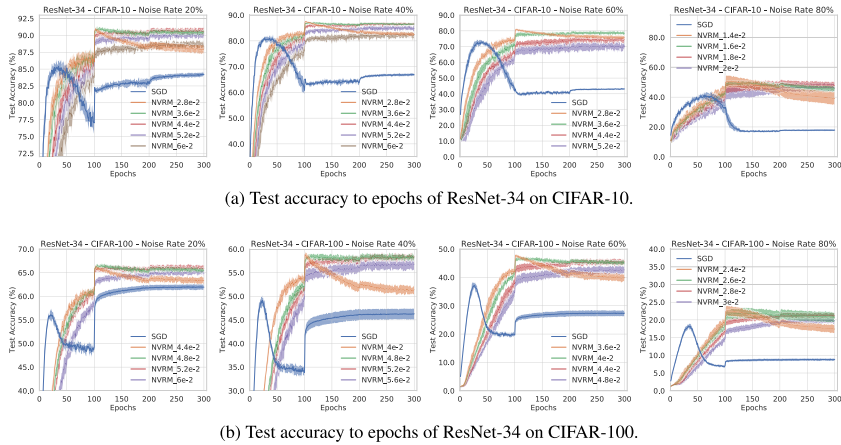(b) Test accuracy to epochs of ResNet-34 on CIFAR-100.

Figure 10: Curves of test accuracy to epochs of ResNet-34. The first row is on CIFAR-10 and the second on CIFAR-100. The four columns are, respectively, for label noise rate 20%, 40%, 60%, and 80%. NVRM with various variability scales $b$ can consistently relieve memorizing noisy labels.

**E.4 Robustness to Catastrophic Forgetting.** See Figures 11 and 12. NVRM also enhances robustness to catastrophic forgetting in the setting of both incremental class learning. The variability hyperparameter $b$ of NVRM
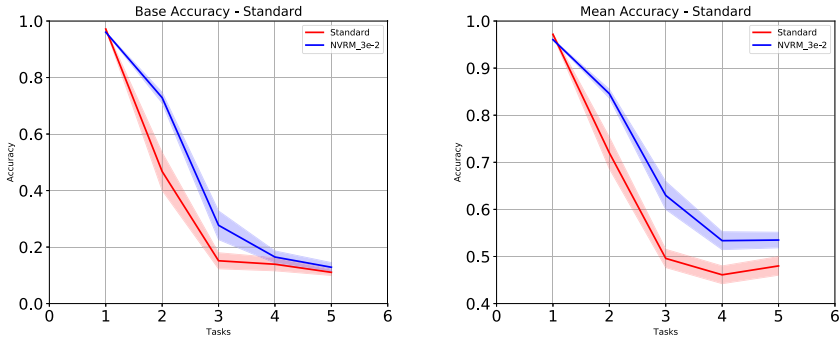
Figure 11: Curves of test accuracy to the number of tasks in continually learning Permuted MNIST. The two panels are, respectively, the accuracy of the base task and the mean accuracy of all learned tasks.
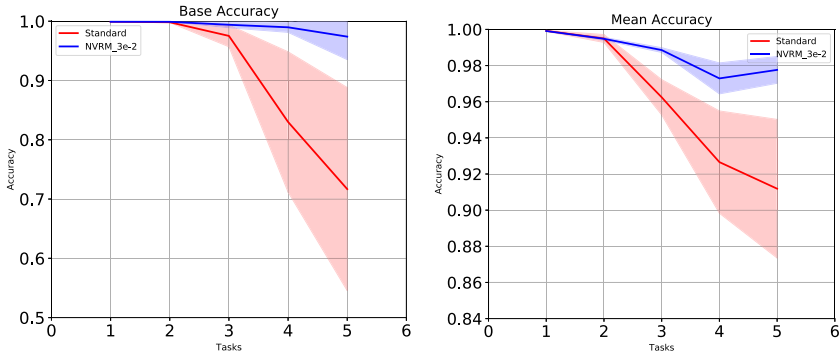


Figure 12: Curves of test accuracy to the number of tasks in continually learning split MNIST. The two panels are, respectively, the accuracy of the base task and the mean accuracy of all learned tasks.

is defaulted to be 0.03. The NVRM curves consistently outperform the counterpart curves.

**E.5 SGD with Large Gradient Noise Cannot Relieve Noise Memorization.** It is well-known that increasing the ratio of the learning rate and the batch size $\frac{\eta}{B}$ may enhance the scale of gradient noise in SGD and help find flatter minima (Jastrzębski et al., 2017; He et al., 2019). However, our theoretical analysis suggests that as stochastic gradient noise carries the information about training data, there is no theoretical guarantee that large stochastic gradient noise can work as well as NVRM. We empirically studied SGD with large stochastic gradient noise in Figure 13 and found that SGD with various learning rates finally still memorizes noisy labels, while
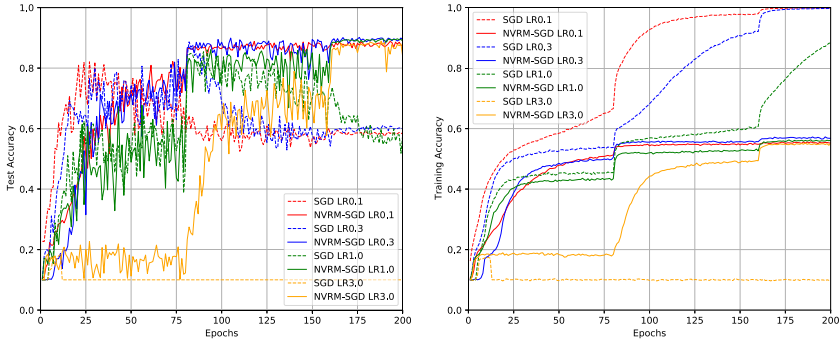
Figure 13: Curves of test accuracy to learning rates. Data set: CIAF-10 with 40% label noise. While SGD with larger stochastic gradient noise memorizes noisy labels more slowly, it still memorizes nearly all noisy labels in the final phase of training. In contrast, NVRM-SGD with various learning rates can consistently relieve overfitting noisy labels.

NVRM-SGD with various learning rates can consistently relieve overfitting noisy labels. Note that in Figure 13, we initialized the learning rates as {0.1, 0.3, 1, 3}, respectively, and divided the learning rate by 10 after every 60 epochs.

## Acknowledgments

## References

Achille, A., Paolini, G., & Soatto, S. (2019). *Where is the information in a deep neuralnetwork?* arXiv:1905.12213.

Achille, A., & Soatto, S. (2018). Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, *19*(1), 1947–1980.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., & Tuytelaars, T. (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision* (pp. 139–154). Berlin: Springer.

Allen-Zhu, Z., Li, Y., & Liang, Y. (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*, *32* (pp. 6155–6166). Red Hook, NY: Curran.

Allen-Zhu, Z., Li, Y., & Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *Proceedings of the International Conference on Machine Learning* (pp. 242–252).

An, G. (1996). The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation*, *8*(3), 643–674.

Arora, S., Cohen, N., & Hazan, E. (2018). On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning*.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., . . . Lacoste-Julien, S. (2017). A closer look at memorization in deep networks. In *Proceedings of the International Conference on Machine Learning* (pp. 233–242).

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 (pp. 1613–1622).

Bottou, L. (1998). Online learning and stochastic approximations. *On-Line Learning in Neural Networks*, *17*(9), 142.

Carbone, G., Wicker, M., Laurenti, L., Patane, A., Bortolussi, L., & Sanguinetti, G. (2020). *Robustness of Bayesian neural networks to gradient-based attacks*. arXiv:2002.04359.

Chen, Y., Mai, Y., Xiao, J., & Zhang, L. (2019). Improving the antinoise ability of DNNs via a bio-inspired noise adaptive activation function rand softplus. *Neural Computation*, *31*(6), 1215–1233.

Churchland, M. M., Byron, M. Y., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, . . . Shenoy, K. V. (2010). Stimulus onset quenches neural variability: A widespread cortical phenomenon. *Nature Neuroscience*, *13*(3), 369.

Churchland, M. M., Byron, M. Y., Ryu, S. I., Santhanam, G., & Shenoy, K. V. (2006). Neural variability in premotor cortex provides a signature of motor preparation. *Journal of Neuroscience*, *26*(14), 3697–3712.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. Hoboken, NJ: Wiley.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Dinh, L., Pascanu, R., Bengio, S., & Bengio, Y. (2017). Sharp minima can generalize for deep nets. In *Proceedings of the International Conference on Machine Learning* (pp. 1019–1028). PMLR.

Dinstein, I., Heeger, D. J., & Behrmann, M. (2015). Neural variability: Friend or foe? *Trends in Cognitive Sciences*, *19*(6), 322–328.

Doan, T., Bennani, M., Mazoure, B., Rabusseau, G., & Alquier, P. (2020). *A theoretical analysis of catastrophic forgetting through the NTK overlap matrix*. arXiv:2010.04003.

Duchi, J. C., Bartlett, P. L., & Wainwright, M. J. (2012). Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, *22*(2), 674–701.

Fetters, L. (2010). Perspective on variability in the development of human action. *Physical Therapy*, *90*(12), 1860–1867.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*(3), 183–192.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). *An empirical investigation of catastrophic forgetting in gradient-based neural networks*. arXiv:1312.6211.

Graves, A. (2011). Practical variational inference for neural networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *24* (pp. 2348–2356). Red Hook, NY: Curran.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., . . . Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*, *31* (pp. 8527–8537). Red Hook, NY: Curran.

Harutyunyan, H., Reing, K., Steeg, G. V., & Galstyan, A. (2020). *Improving generalization by controlling label-noise information in neural network weights*. arXiv:2002.07933.

He, F., Liu, T., & Tao, D. (2019). Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E, Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*, *32* (pp. 1141–1150). Red Hook, NY: Curran.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE.

Hedden, T., & Gabrieli, J. D. (2004). Insights into the ageing mind: A view from cognitive neuroscience. *Nature Reviews Neuroscience*, *5*(2), 87–96.

Hinton, G. E., & Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory* (pp. 5–13). New York: Association for Computing Machinery.

Hochreiter, S., & Schmidhuber, J. (1997). Flat minima. *Neural Computation*, *9*(1), 1–42.

Hornik, K. (1993). Some new results on neural network approximation. *Neural Networks*, *6*(8), 1069–1072.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Houghton, C. (2019). Calculating the mutual information between two spike trains. *Neural Computation*, *31*(2), 330–343.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., & Storkey, A. (2017). *Three factors influencing minima in SGD.* arXiv:1711.04623.

Kawaguchi, K., Huang, J., & Kaelbling, L. P. (2019). Effect of depth and width on local minima in deep learning. *Neural Computation*, *31*(7), 1462–1498.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the International Conference on Learning Representations*.

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., & Srivastava, A. (2018). Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In *Proceedings of the International Conference on Machine Learning* (pp. 2611–2620).

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization.* arXiv:1412.6980.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., . . . Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526.

Kristiadi, A., Hein, M., & Hennig, P. (2020). Being Bayesian, even just a bit, fixes overconfidence in RElU networks. In *Proceedings of the International Conference on Machine Learning* (pp. 5436–5446).

Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images*. Citeseer.

LeCun, Y. (1998). *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

Li, Y., & Liang, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing system*, *31* (pp. 8157–8166). Red Hook, NY: Curran.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., . . . Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60–88.

Loshchilov, I., & Hutter, F. (2018). Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*.

McAllester, D. A. (1999a). Pac-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory* (pp. 164–170). New York: ACM.

McAllester, D. A. (1999b). Some Pac-Bayesian theorems. *Machine Learning*, *37*(3), 355–363.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, *24*, 109–165.

Mongeon, D., Blanchet, P., & Messier, J. (2013). Impact of Parkinson's disease and dopaminergic medication on adaptation to explicit and implicit visuomotor perturbations. *Brain and Cognition*, *81*(2), 271–282.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., & Martens, J. (2015). *Adding gradient noise improves learning for very deep networks.* arXiv:1511.06807.

Nesterov, Y., & Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, *17*(2), 527–566.

Neyshabur, B., Bhojanapalli, S., McAllester, D., & Srebro, N. (2017). Exploring generalization in deep learning. In I. Guyon, Y. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, *30* (pp. 5947–5956). Red Hook, NY: Curran

Ölveczky, B. P., Otchy, T. M., Goldberg, J. H., Aronov, D., & Fee, M. S. (2011). Changes in the neural control of a complex motor sequence during learning. *Journal of Neurophysiology*, *106*(1), 386–397.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, *113*, 54–71.

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

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*.

Stein, R. B., Gossen, E. R., & Jones, K. E. (2005). Neuronal variability: Noise or part of the signal? *Nature Reviews Neuroscience*, *6*(5), 389–397.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the International Conference on Machine Learning*.

Tumer, E. C., & Brainard, M. S. (2007). Performance variability enables adaptive plasticity of "crystallized" adult birdsong. *Nature*, *450*(7173), 1240–1244.

Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 681–688).

Wen, W., Wang, Y., Yan, F., Xu, C., Wu, C., Chen, Y., & Li, H. (2018). *Smoothout: Smoothing out sharp minima to improve generalization in deep learning*. arXiv:1805.07898.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques*. San Mateo, CA: Morgan Kaufmann.

Xie, Z., Sato, I., & Sugiyama, M. (2020). *Stable weight decay regularization*. arXiv:2011.11152.

Xie, Z., Sato, I., & Sugiyama, M. (2021). A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *Proceedings of the International Conference on Learning Representations*.

Xie, Z., Wang, X., Zhang, H., Sato, I., and Sugiyama, M. (2020). *Adai: Separating the effects of adaptive learning rate and momentum inertia.* arXiv:2006.15815.

Xu, A., & Raginsky, M. (2017). Information-theoretic analysis of generalization capability of learning algorithms. In I. Guyon, Y. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, *30* (pp. 2524–2533). Red Hook, NY: Curran.

Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, *70*, 3987–3995.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). *Understanding deep learning requires rethinking generalization.* In *Proceedings of the International Conference on Machine Learning*.

Zhou, M., Liu, T., Li, Y., Lin, D., Zhou, E., & Zhao, T. (2019). Toward understanding the importance of noise in training neural networks. In *Proceedings of the International Conference on Machine Learning*.

Zhu, Z., Wu, J., Yu, B., Wu, L., & Ma, J. (2019). The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *Proceedings of the International Conference on Machine Learning* (pp. 7654–7663).