A VARIANCE PRINCIPLE EXPLAINS WHY DROPOUT FINDS FLATTER MINIMA

Anonymous authors
Paper under double-blind review

ABSTRACT

Although dropout has achieved great success in deep learning, little is known about how it helps the training find a good generalization solution in the high-dimensional parameter space. In this work, we show that the training with dropout finds the neural network with a flatter minimum compared with standard gradient descent training. We further study the underlying mechanism of why dropout finds flatter minima through experiments. We propose a *Variance Principle* that the variance of a noise is larger at the sharper direction of the loss landscape. Existing works show that SGD satisfies the variance principle, which leads the training to flatter minima. Our work show that the noise induced by the dropout also satisfies the variance principle that explains why dropout finds flatter minima. In general, our work points out that the variance principle is an important similarity between dropout and SGD that lead the training to find flatter minima and obtain good generalization.

1 INTRODUCTION

Dropout is used with gradient-descent-based algorithms for training DNNs (Hinton et al., 2012; Srivastava et al., 2014). During training, the output of each neuron is multiplied with a random variable with probability p as one and $1-p$ as zero. Note that p is called dropout rate, and every time for computing concerned quantity, the variable is randomly sampled at each feedforward operation. Dropout has been an indispensable trick in the training of deep neural networks (DNNs), however, with very little understanding.

Similar to SGD, training with dropout is equivalent to that with some specific noise. To understand what kind of noise benefits the generalization of training, we proposes a variance principle of a noise, that is,

Variance Principle: the variance of a noise is larger at the sharper direction of the loss landscape.

If a noise satisfies the variance principle, it can help the training select flatter minima and leads the training to better generalization. As shown in Zhu et al. (2018); Feng & Tu (2021), the noise in SGD satisfies the variance principle and SGD can find flatter minima and obtain better generalization (Keskar et al., 2016; Neyshabur et al., 2017).

In this work, we study the characteristic of minima learned with dropout. We show that compared with the standard gradient descent (GD), the GD with dropout selects flatter minima. As suggested by many existing works (Keskar et al., 2016; Neyshabur et al., 2017; Zhu et al., 2018), flatter minima are more likely to have better generalization and stability. We then put efforts to show that the noise induced by the dropout satisfies the variance principle, which explains why dropout finds flatter minima.

To examine the variance principle, we explore the relation between the flatness of the loss landscape and the noise structure induced by dropout at minima through three methods and obtain a consistent result that the noise is larger at the sharper direction of the loss landscape to help the training select flatter minima.

First, we examine the inverse variance-flatness relation, similar to Feng & Tu (2021). We define the flatness of a minimum at one direction by the length of the largest interval in the considered direction, which covers the minimum and no point in the interval has loss larger than twice of the loss of the

minimum, denoted by $F_{\boldsymbol{p}}$ for the direction $\boldsymbol{p}$. We then consider two definitions of the noise structure, i.e., random trajectory covariance Σ_t and random gradient covariance Σ_g. For the random trajectory covariance, we train the network to an "exploration phase" (Shwartz-Ziv & Tishby, 2017), where the loss decreases with a very slow speed and then sample parameter sets $\{\boldsymbol{\theta}_i\}_{i=1}^N$ from N consecutive training steps to compute the covariance, where $\boldsymbol{\theta}_i$ is the network parameter set at step i. For the gradient covariance, we train the network until the loss is very small and then freeze the training. We sample N gradients $\{\boldsymbol{g}_i\}_{i=1}^N$ with different dropout variables to compute the covariance. In each sample, the dropout rate is fixed. For both random trajectory covariance and random gradient covariance, we perform principle component analysis (PCA) and obtain similar results. We find that at the direction of larger variance (larger eigen-value), the loss landscape of the minimum is sharper, i.e., inverse variance-flatness relation.

Second, we study the relation between the Hessian and the noise structure induced by dropout. The eigenvalues of the Hessian of the loss at a minimum are also often used to indicate the flatness. The landscape at an eigen-direction is claimed sharper if the corresponding eigen-value is larger. For each eigen-direction $\boldsymbol{v}_j$, we project the parameter trajectory $\{\boldsymbol{\theta}_i\}_{i=1}^N$ or gradients $\{\boldsymbol{g}_i\}_{i=1}^N$ to the direction of $\boldsymbol{v}_j$ and compute the variance. We find that the noise variance is larger at the direction with a larger eigen-value.

Third, we show that the Hessian matrix aligns well with the random gradient covariance of gradients $\{\boldsymbol{g}_i\}_{i=1}^N$, i.e., their eigen directions of large eigen-values are close, similar to Zhu et al. (2018).

These empirical works show that the noise structure induced by the dropout tends to have larger variance in order to escape the sharper direction, i.e., variance principle, thus, leading to flatter minima. These characteristics of dropout are very similar to SGD (Keskar et al., 2016; Zhu et al., 2018; Feng & Tu, 2021). The similarity between dropout and SGD suggests that modelling their similarity may be a key to understanding how and what stochasticity benefits the training.

2 Related works

Dropout was proposed as a simple way to prevent neural networks from overfitting, and thus improving the generalization of the network, to a certain extent (Hinton et al., 2012; Srivastava et al., 2014). Many researches have empirically shown that SGD can improve the generalization performance of neural networks through finding a flatter solution (Li et al., 2017; Jastrzebski et al., 2017; 2018). This work utilizes the current understanding of SGD to study dropout and shows that much similarity is shared between SGD and dropout.

The flatness of the solution is an important aspect of understanding the generalization of neural networks (Keskar et al., 2016; Neyshabur et al., 2017; Zhu et al., 2018). A number of works suggested that the learning rate and batch size determine the flatness of the solutions (Jastrzebski et al., 2017; 2018; Wu et al., 2018). Li et al. (2017) propose a visualization method of the loss landscape at 1-d cross-section to visualize the flatness.

Feng & Tu (2021) investigate the connection between SGD learning dynamics and the loss landscape through the principle component analysis (PCA), and show that SGD dynamics follow a low-dimensional drift-diffusion motion in the weight space. Through characterizing the loss landscape by its flatness in each PCA direction around the solution found by SGD, they also reveal a robust inverse relation between the weight variance and the landscape flatness in PCA directions, thus finding that SGD serves as a landscape dependent annealing algorithm to search for flat minima.

Zhu et al. (2018) study a general form of gradient based optimization dynamics with unbiased noise to analyze the behavior of SGD on escaping from minima and its regularization effects. They also introduce an indicator to characterize the efficiency of escaping from minima through measuring the alignment of noise covariance and the curvature of loss function and thus revealing the anisotropic noise of SGD.

Table 1: Three types of experiments explain why dropout finds flat minima. $\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S))$ is the variance of the network parameters or gradients projected in the characteristic direction of the Hessian matrix.

	Dropout covariance Σ	
	Trajectory variance Σ_t	gradient variance Σ_g
Interval flatness $F_{\boldsymbol{v}}$	$\lambda(\Sigma)$ vs. $F_{\boldsymbol{v}}$, Fig. 2	
Hessian flatness $\lambda(H)$	$\{\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S)), \lambda_i(H)\}$, Fig. 3	
	$\setminus$	Alignment: $Tr(H\Sigma_g)$, Fig. 4.

3 PRELIMINARY

3.1 DROPOUT

Consider an L-layer neural network $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. With dropout (Srivastava et al., 2014), the feedforward operation in a network $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is

$$f_{\boldsymbol{\theta}}^{[0]}(\boldsymbol{x}) = \boldsymbol{x}, \tag{1}$$

$$r_j \sim \mathrm{Bernoulli}(p), \tag{2}$$

$$f_{\boldsymbol{\theta}}^{[l]}(\boldsymbol{x}) = \boldsymbol{r}^{[l]} \circ \sigma \circ (\boldsymbol{W}^{[l-1]} f_{\boldsymbol{\theta}}^{[l-1]}(\boldsymbol{x}) + \boldsymbol{b}^{[l-1]}) \quad 1 \leq l \leq L-1, \tag{3}$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{x}) = \boldsymbol{W}^{[L-1]} f_{\boldsymbol{\theta}}^{[L-1]}(\boldsymbol{x}) + \boldsymbol{b}^{[L-1]}, \tag{4}$$

where p is the dropout rate, $\boldsymbol{W}^{[l]} \in \mathbb{R}^{m_{l+1} \times m_l}$, $\boldsymbol{b}^{[l]} = \mathbb{R}^{m_{l+1}}$, $m_0 = d_{\mathrm{in}} = d$, $m_L = d_{\mathrm{o}}$, σ is a scalar function and "$\circ$" means entry-wise operation. We denote the set of parameters by

$$\boldsymbol{\theta} = (\boldsymbol{W}^{[0]}, \boldsymbol{W}^{[1]}, \ldots, \boldsymbol{W}^{[L-1]}, \boldsymbol{b}^{[0]}, \boldsymbol{b}^{[1]}, \ldots, \boldsymbol{b}^{[L-1]}),$$

3.2 INTERVAL FLATNESS

We use the definition of flatness in Feng & Tu (2021). For convenience, we call it *interval flatness* Around a specific solution $\boldsymbol{\theta}_0^*$, we compute the loss function profile $L_{\boldsymbol{v}}$ along the direction $\boldsymbol{v}$:

$$L_{\boldsymbol{v}}(\delta\theta) \equiv L(\boldsymbol{\theta}_0^* + \delta\theta \boldsymbol{v}).$$

The interval flatness $F_{\boldsymbol{v}}$ is defined as the width of the region within which $L_{\boldsymbol{v}}(\delta\theta) \leq 2L_{\boldsymbol{v}}(0)$. We determine $F_{\boldsymbol{v}}$ by finding two closest points $\boldsymbol{\theta}_{\boldsymbol{v}}^l < 0$ and $\boldsymbol{\theta}_{\boldsymbol{v}}^r > 0$ on each side of the minimum that satisfy $L_{\boldsymbol{v}}(\boldsymbol{\theta}_{\boldsymbol{v}}^l) = L_{\boldsymbol{v}}(\boldsymbol{\theta}_{\boldsymbol{v}}^r) = 2L_{\boldsymbol{v}}(0)$. The interval flatness is defined as:

$$F_{\boldsymbol{v}} \equiv \boldsymbol{\theta}_{\boldsymbol{v}}^r - \boldsymbol{\theta}_{\boldsymbol{v}}^l. \tag{5}$$

A larger value of $F_{\boldsymbol{v}}$ means a flatter landscape in the direction $\boldsymbol{v}$.

3.3 RANDOMNESS INDUCED BY DROPOUT

3.3.1 RANDOM TRAJECTORY DATA

The training process usually goes through two phases, one is fast convergence phase, the other is exploration phase, in which the loss decay very slowly. This can be understood by frequency principle (Xu et al., 2019; 2020), which states that DNNs fast learn low-frequency components but slowly learn high-frequency ones.

We collect parameter sets $S_{para} = \{\boldsymbol{\theta}_i\}_{i=1}^N$ from N consecutive training steps in the exploration phase, where $\boldsymbol{\theta}_i$ is the network parameter set at step i.

3.3.2 RANDOM GRADIENT DATA

We train the network until the loss is very small and then freeze the training. We sample N gradients of the loss function w.r.t. the parameters with different dropout variables, i.e., $S_{grad} = \{\boldsymbol{g}_i\}_{i=1}^N$. In each sample, the dropout rate is fixed.

3.4 INVERSE VARIANCE-FLATNESS RELATION

We study the inverse variance flatness relation for both random trajectory data and random gradient data. For convenience, we denote data as S and its covariance as Σ.

3.4.1 INTERVAL FLATNESS VS. VARIANCE

Following Feng & Tu (2021), we perform principle component analysis (PCA) for Σ. Denote $\lambda_i(\Sigma)$ as the i-th eigenvalue of Σ, and denote its corresponding eigen-vector as $\boldsymbol{v}_i(\Sigma)$. We then compute the flatness of the loss landscape in the direction $\boldsymbol{v}_i(\Sigma)$, and denote it as $F_{\boldsymbol{v}_i(\Sigma)}$. Finally, we display the scatter plot of $\{\lambda_i(\Sigma), F_{\boldsymbol{v}_i(\Sigma)}\}$.

3.4.2 PROJECTED VARIANCE VS. HESSIAN FLATNESS

The method in Feng & Tu (2021), requires many training steps to obtain the covariance. To obtain the variance induced by the dropout at a fixed position $\boldsymbol{\theta}$, we propose another way to characterize the inverse variance-flatness relation. We use the the eigen-value $\lambda_i(H)$ of the Hessian H of the loss landscape at $\boldsymbol{\theta}$ to denote the sharpness. The variance induced by dropout at the corresponding eigen-direction $\boldsymbol{v}_i(H)$ is computed by the following procedure. For each eigen-direction $\boldsymbol{v}_i$ of Hessian H, we project the sampled parameters or the gradients of sampled parameter S to direction $\boldsymbol{v}_i$ by inner product, denoted by $\mathrm{Proj}_{\boldsymbol{v}_i}(S)$. Then, the variance at direction $\boldsymbol{v}_i(H)$ is the variance of dataset $\mathrm{Proj}_{\boldsymbol{v}_i}(S)$, denoted by $\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S))$. Finally, we display the scatter plot of $\{\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S)), \lambda_i(H)\}$.

3.5 ALIGNMENT BETWEEN HESSIAN AND GRADIENT COVARIANCE

We use the method in Zhu et al. (2018) to quantify the alignment between the noise structure and the curvature of loss surface. For each training step i, we calculate the alignment parameter T_i:

$$T_i = \mathrm{Tr}(H_i \Sigma_i),$$

where Σ_i is the ith-step covariance matrix generated by dropout layers and H_i is the ith-step Hessian matrix of network parameters.

4 EXPERIMENTAL SETUP

To understand the effects of dropout, we trained a number of networks with different structures. We consider two types of neural networks: 1) Fully-connected neural networks (FNNs) that are optimized for performance on MNIST. For FNN, all parameters are initialized by Xavier initialization(Glorot & Bengio, 2010). 2) Convolutional neural networks (CNNs) that are optimized for performance on CIFAR-10. For CNN, all parameters are initialized by He initialization (He et al., 2015). The loss of all our experiments is cross entropy loss.

For Fig. 1 (a), we use the FNN with size $784 - 1024 - 1024 - 10$. We add dropout layers behind the first and the second layers with dropout rate of 0.8 and 0.5, respectively. We train the network using default Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.0001.

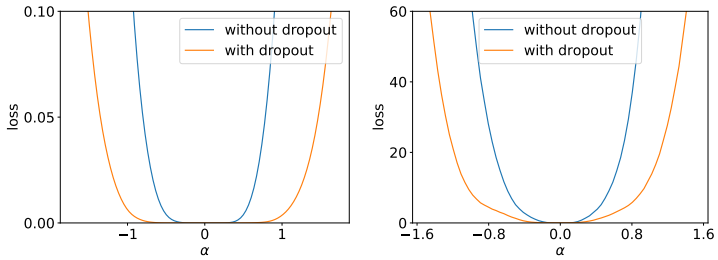
For Fig. 1 (b), we use vgg-9 (Simonyan & Zisserman, 2014) to compare the loss landscape flatness w/o dropout layers. Models are trained using GD with Nesterov momentum, training-size 128 for 300 epochs. The learning rate was initialized at 0.1, and divided by a factor of 10 at epochs 150, 225 and 275. We only use 2048 examples for training to compromise with the computational burden.

For Fig. 2, 3, 4, we use the FNN with size $784 - 50 - 50 - 10$. We train the network using GD with the first 10,000 training data as the training set. We add a dropout layer behind the second layer. The dropout rate and learning rate are specified and unchanged in each experiment.

5 DROPOUT FINDS FLATTER MINIMA

Dropout is almost ubiquitous in training deep networks. It is interesting and important to understand what makes dropout improve the generalization of training neural networks. Inspired by the study of SGD (Keskar et al., 2016), we explore the flatness of the minima found by dropout.

The loss landscape of DNNs is highly non-convex and complicate (Skorokhodov & Burtsev, 2019), but with certain characteristics, such as embedding principle (Zhang et al., 2021) shows that the loss landscape of any network "contains" all critical points of all narrower networks, in the sense that, any critical point of any narrower network can be embedded to a critical point of the target network preserving its output function. It is impractical to visualize the loss landscape in the high-dimensional space. To compare two minima, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, in a loss landscape, one simple way (Keskar et al., 2016) is to visualize 1-d cross-section of the interpolation between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. However, empirical studies (Li et al., 2017) show that this simple way could be misleading when $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ have difference of orders of magnitude. Li et al. (2017) visualizes loss functions using filter-wise normalized directions to remove the scaling effect. We adopt this method (Li et al., 2017) in this work as follows. To obtain a direction for a network with parameters $\boldsymbol{\theta}$, we begin by producing a random Gaussian direction vector $\boldsymbol{d}$ with dimensions compatible with $\boldsymbol{\theta}$. Then, we normalize each filter in $\boldsymbol{d}$ to have the same norm of the corresponding filter in $\boldsymbol{\theta}$. In other words, we make the replacement $\boldsymbol{d}_{i,j} \leftarrow \frac{\boldsymbol{d}_{i,j}}{\|\boldsymbol{d}_{i,j}\|} \|\boldsymbol{\theta}_{i,j}\|$, where $\boldsymbol{d}_{i,j}$ represents the jth filter (not the jth weight, for FNN, one layer is one filter.) of the ith layer of $\boldsymbol{d}$, and $\|\cdot\|$ denotes the Frobenius norm. We use $f(\alpha) = L(\boldsymbol{\theta} + \alpha \boldsymbol{d})$ to characterize the loss landscape around the minima obtained with dropout layers $\boldsymbol{\theta}^*_{\boldsymbol{d}}$ and without dropout layer $\boldsymbol{\theta}^*$. For both network structures shown in Fig. 1, dropout can improve the generalization of the network and find a flatter minimum.



(a) flatness of FNN (b) flatness of vgg-9

Figure 1: The 1D visualization of solutions of different network structures obtained with or without dropout layers. (a) The FNN is trained on MNIST dataset. For experiment with dropout layers, we add dropout layer after the first and the second layers, the dropout rates of the two dropout layers are 0.8 and 0.5, respectively. The test accuracy for model with dropout layers is 98.7% while 98.1% for model without dropout layers. (b) The vgg-9 network is trained on CIFAR-10 dataset using the first 2048 examples as training dataset. For experiment with dropout layers, we add dropout layers after the pooling layers, the dropout rates of dropout layers are 0.8. The test accuracy for model with dropout layers is 60.6% while 59.2% for model without dropout layers.

6 Inverse Variance-Flatness Relation

Similar to SGD, the effect of dropout can be equivalent to imposing a specific noise on the gradient. A random noise, such as isotropic noise, can help the training escape local minima, but can not robustly improve generalization (An, 1996; Zhu et al., 2018). The noise induced by the dropout should have certain properties that can lead the training to good minima.

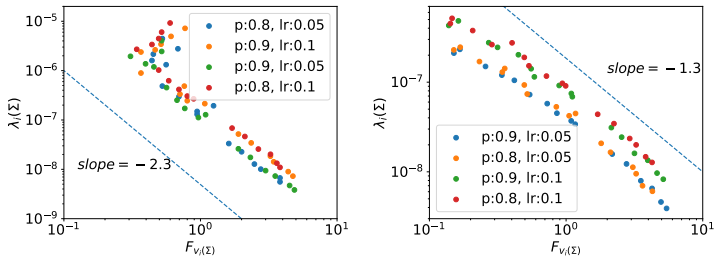
In this section, we show that the noise induced by the dropout satisfies the variance principle, that is, the noise variance is larger along the sharper direction of the loss landscape at a minimum. The landscape-dependent structure helps the training escape sharp minima. We utilize three methods to examine the variance principle for dropout, as summarized in Table 1.

6.1 Interval Flatness vs. Variance

We use the principle component analysis (PCA) to study the weight variations when the accuracy is nearly 100%. The FNNs are trained on MNIST with the first 10000 examples as the training set for computational efficiency. The networks are trained with full batch for different learning rates and

dropout rates under the same random seed (that is, with the same initialization parameters). After the loss is small enough, we sample the parameters or gradients of parameters N times ($N = 3000$ in this experiment) and use the method introduced in Sec. 3.3 to construct covariance matrix Σ by the weights S_{para} or gradients S_{grad} between two hidden layers. The PCA was done for the covariance matrix Σ. We then compute the interval flatness of the loss function landscape at eigen-directions, i.e., $\{F_{\boldsymbol{v}_i(\Sigma)}\}_{i=1}^N$. Note that the PCA spectrum $\{\lambda_i(\Sigma)\}_{i=1}^N$ indicate the variance of weights S_{para} or gradients S_{grad} at corresponding eigen-directions.

As shown in Fig. 2, for different learning rates and dropout rates, there is an inverse relationship between the interval flatness of the loss function landscape $\{F_{\boldsymbol{v}_i(\Sigma)}\}_{i=1}^N$ and the dropout variance, i.e., the PCA spectrum $\{\lambda_i(\Sigma)\}_{i=1}^N$. We can approximately see a power-law relationship between $\{F_{\boldsymbol{v}_i(\Sigma)}\}_{i=1}^N$ and $\{\lambda_i(\Sigma)\}_{i=1}^N$ for different dropout rates and learning rates.



(a) datasets S sampled from parameters

(b) datasets S sampled from gradients of parameters

Figure 2: The inverse relation between the variance $\{\lambda_i(\Sigma)\}_{i=1}^N$ and the flatness $\{F_{\boldsymbol{v}_i(\Sigma)}\}_{i=1}^N$ for different choices of dropout rate p and learning rate lr. The FNN is trained on MNIST dataset using the first 10000 examples as training dataset. The PCA is done for different datasets S sampled from parameters for (a) and sampled from gradients of parameters for (b). The dash lines give the approximate slope of the scatter.

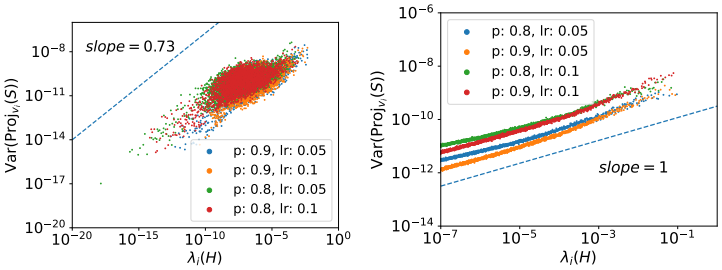
6.2 Projected variance vs. Hessian flatness

The eigenvalues of the Hessian of the loss at a minimum are also often used to indicate the flatness. A large eigenvalue correspond to a sharper direction. In this section, we study the relationship between eigen-values of the Hessian H of parameters $\boldsymbol{\theta}$ at the end point of training and the variances of dropout at corresponding eigen-directions. As mentioned in the Preliminary section, we sample the parameters or gradients of parameters 1000 times, that is $N = 1000$. For each eigen-direction $\boldsymbol{v}_i$ of Hessian H, we project the sampled parameters or the gradients of sampled parameter to direction $\boldsymbol{v}_i$ by inner product, denoted by $\mathrm{Proj}_{\boldsymbol{v}_i}(S)$. Then, we compute the variance of the projected data, i.e., $\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S))$.

As shown in Fig. 3, we find that there is also a power-law relationship between $\{\lambda_i(H)\}_{i=1}^D$ and $\{\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S))\}_{i=1}^D$ for different dropout rates and learning rates, no matter if S is sampled from parameters or gradients of parameters.

6.3 Alignment between Hessian and gradient covariance

Zhu et al. (2018) show that the alignment indicator $\mathrm{Tr}(H\Sigma)$ plays an crucial role for stochastic processes escaping from minima, where H is the Hessian and Σ is the noise covariance. In this subsection, we study the alignment between the Hessian and the random gradient covariance at each training step. Note that the training is performed by GD without dropout. At step i, we sample the gradients of parameters $\{\boldsymbol{g}_i^j\}_{j=1}^N$ by tentatively adding a dropout layer between the hidden layers. For each step i, we the compute $\mathrm{Tr}(H_i\Sigma_i)$, where H_i is the Hessian of the loss at the parameter set at step i and Σ_i is the covariance of $\{\boldsymbol{g}_i^j\}_{j=1}^N$.



(a) datasets S sampled from parameters

(b) datasets S sampled from gradients of parameters

Figure 3: The relation between the variance $\{\mathrm{Var}(\mathrm{Proj}_{\boldsymbol{v}_i}(S))\}_{i=1}^{D}$ and the eigenvalue $\{\lambda_i(H)\}_{i=1}^{D}$ for different choices of dropout rate p and learning rate lr. The FNN is trained on MNIST dataset using the first 10000 examples as training dataset. The projection is done for different datasets S sampled from parameters for (a) and sampled from gradients of parameters for (b). The dash lines give the approximate slope of the scatter.

In order to show the anisotropic structure, we construct the isotropic noise for comparison, i.e., $\bar{\Sigma}_i = \frac{\mathrm{Tr}\,\Sigma_i}{D} I$ of the covariance matrix Σ_i, where D is the number of parameters. In our experiments, $D = 2500$. As shown in Fig.4 , in the whole training process under different learning rates and dropout rates, $\mathrm{Tr}(H_i\Sigma_i)$ is much larger than $\mathrm{Tr}(H_i\bar{\Sigma}_i)$, indicating the anisotropic structure of dropout noise and its high alignment with the Hessian matrix.

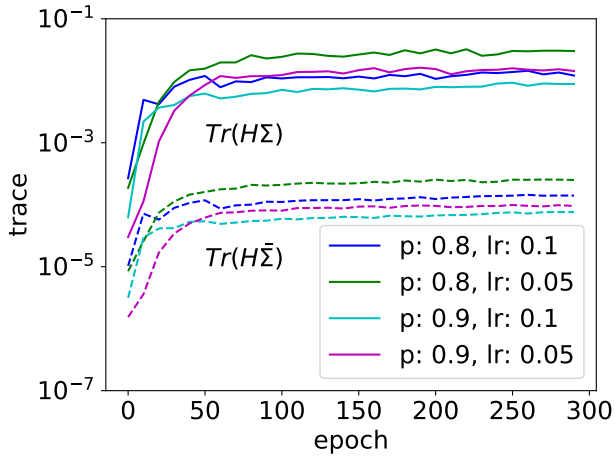


Figure 4: Comparison between $\mathrm{Tr}(H_i\Sigma_i)$ and $\mathrm{Tr}(H_i\bar{\Sigma}_i)$ in each training epoch i for different choices of dropout rate p and learning rate lr. The FNN is trained on MNIST dataset using the first 10000 examples as training dataset. The solid and the dotted lines represent the value of $\mathrm{Tr}(H_i\Sigma_i)$ and $\mathrm{Tr}(H_i\bar{\Sigma}_i)$, respectively.

7 Conclusion and Discussion

In this work, we propose a variance principle that noise is larger at the sharper direction of the loss landscape. If a noise satisfies the variance principle, it helps the training select flatter minima and leads the training to good generalization. We empirical show that dropout finds flat minima during the training and examine that the dropout satisfies the variance principle to explain why dropout finds flat minima by the following three perspectives: interval flatness vs. variance, projected variance vs. Hessian flatness and alignment between Hessian and gradient covariance.

The dropout and the SGD are common in sharing the variance principle. Based on this, there could be a general theory framework that can model both dropout and SGD to understand why they have better generalization. The research on understanding the good generalization of dropout and SGD is far from complete. As a starting point, the variance principle shows a promising and reasonable direction for understanding the stochastic training of neural networks.

REFERENCES

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

Yu Feng and Yuhai Tu. The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima. Proceedings of the National Academy of Sciences, 118(9), 2021.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pp. 1026–1034, 2015.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. arXiv preprint arXiv:1711.04623, 2017.

Stanislaw Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of dnn loss and the sgd step length. arXiv preprint arXiv:1807.05031, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR (Poster), 2015.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. arXiv preprint arXiv:1712.09913, 2017.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. arXiv preprint arXiv:1706.08947, 2017.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017.

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

Ivan Skorokhodov and Mikhail Burtsev. Loss landscape sightseeing with multi-point optimization. arXiv preprint arXiv:1910.03867, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1):1929–1958, 2014.

Lei Wu, Chao Ma, et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. Advances in Neural Information Processing Systems, 31:8279–8288, 2018.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In International Conference on Neural Information Processing, pp. 264–274. Springer, 2019.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. Communications in Computational Physics, 28(5):1746–1767, 2020.

Yaoyu Zhang, Zhongwang Zhang, Tao Luo, and Zhi-Qin John Xu. Embedding principle of loss landscape of deep neural networks. arXiv preprint arXiv:2105.14573, 2021.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. arXiv preprint arXiv:1803.00195, 2018.