Improved Generalization with Curvature Regularization

Xinyang Geng [1 2] **Lechao Xiao** [1 2] **Hossein Mobahi** [1] **Jeffrey Pennington** [1]

Abstract

Recent advances in high-performance computing and the abundance of large labeled datasets have enabled machine learning practitioners to successfully develop and deploy deep learning models with enormous numbers of parameters. Owing to the large degree of overparameterization, it is perhaps surprising that in practice such models often generalize well. Indeed, identifying which types of large models will generalize well and which will generalize poorly remains an important research direction in the theory of deep learning. One recent proposal is that parameter configurations corresponding to "sharp" minima may generalize worse than those that correspond to "wide" minima. In this paper, we propose a computationally-efficient method for approximating a sharpness measure based on the mean curvature of the loss landscape near a critical point. We devise a new form of regularization for deep learning models based on this sharpness measure. Our experiments on fully connected networks and convolutional networks show that such regularization can significantly improve generalization performance.

1. Introduction

In recent years, deep neural networks has been very successful in solving problems in various domains, including computer vision (Krizhevsky et al., 2012; He et al., 2016), natural language processing (Bahdanau et al., 2014; Luong et al., 2015), and reinforcement learning (Mnih et al., 2013; Silver et al., 2017). One of the main drivers of such success is the availability of large datasets. For example, the collection and labeling of ImageNet dataset (Russakovsky et al., 2015) have enabled researchers to train image recognition models on data representative of real world images, thus allowing the model to be applied directly, or to serve as the basis of transfer learning (Oquab et al., 2014).

To achieve good performance on large datasets, increasingly sophisticated models that require more computations have been developed (Szegedy et al., 2017). In order to accelerate the training on these large models and datasets, various techniques have been explored, including adaptive learning rate for stochastic gradient descent(SGD) such as ADAM (Kingma & Ba, 2014) and scaling up training with data parallelism such as (Goyal et al., 2017). These techniques allows the model to converge faster on the training data, thus enabling faster training.

While these acceleration techniques do provide speed advantages, it has recently been observed that they often reduce the generalization performance of the model. (Keskar et al., 2016) argue that there exist "sharp" minima which generalize worse than "wide" minima in the loss landscape, and large-batch gradient descent converges to such "sharp" minima. In (Wilson et al., 2017), the authors argue that ADAM has similar effect on some models. Many authors have given different definitions to the sharpness of minima and argued why "sharp" minima generalize worse. (Hochreiter & Schmidhuber, 1994) define "flat minima" to be the region of connected minima with high volume. (Keskar et al., 2016) define the sharpness of a minimum as the maximum value the loss function attains within some neighborhood of the minimum, and they argue that such definition is highly correlated to the large positive eigenvalues of the loss function Hessian at the minimum. In (Chaudhari et al., 2016), the authors incorporate the sharpness of the minima into the loss by defining the loss function as the "local entropy" of the original loss function. The authors argue that by optimizing the "local entropy" loss function, SGD will find minima with better smoothness properties and generalize better. In (Smith & Le, 2018), the authors show that by using Laplace approximation of the posterior distribution, one discovers that sharper minima with higher curvature generalize worse because they correspond to minima with low Bayesian evidence ratio. On the other hand, some authors have also challenged the argument that "sharp" minima generalize better. (Dinh et al., 2017) argue that by simply reparameterizing a neural network, one can change the sharpness

[1]Google Brain [2]Work done as part of the Google AI Residency program (g.co/airesidency). Correspondence to: Xinyang Geng <young.gengxy@gmail.com>, Lechao Xiao <xlc@google.com>.

Workshop on Modern Trends in Nonconvex Optimization for Machine Learning. The 35th International Conference on Machine Learning, Stockholm

measure under various definitions while keeping the generalization performance of the network the same.

In our work, we focus on the spectrum of the loss Hessian as a sharpness measure (Keskar et al., 2016). We show that:

- The spectrum of the loss Hessian can be approximated by the spectrum of the empirical Fisher information matrix, which is efficient to compute.
- For many network architectures, one can find better minima by incorporating the sharpness measure into the loss function as a form of regularization.

2. Problem Definition and Notation

Consider the following supervised learning problem. There is an unknown distribution $\mathcal{D}$ defined on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. The training set $\bar{\mathcal{D}}$ consists of $|\bar{\mathcal{D}}|$ instances $\{(x_i, y_i)\}_{i=1,\dots,|\bar{\mathcal{D}}|}$ drawn i.i.d. from $\mathcal{D}$. The training distribution $(X, Y) \sim \bar{\mathcal{D}}$ is defined as the uniform distribution on the training set, and we overload the symbol $\bar{\mathcal{D}}$ to also represent this distribution. Throughout the paper, we will use upper-case letters X, Y to represent random variables and lower-case letters x, y to represent actual data. We have a class of prediction functions $\{f_\theta\} : \mathcal{X} \to f(\mathcal{X})$ parametrized by $\theta \in \Theta$, where $f(\mathcal{X})$ is the output space of the model. The loss function $\ell : f(\mathcal{X}) \times \mathcal{Y} \to \mathbb{R}$ is some pre-fixed metric used to evaluate the performance of the prediction. The goal is to learn an optimal parameter θ that minimizes the following population risk,

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(f_\theta(X), Y)]. \tag{1}$$

For classification problems, one popular choice of model output space $f(\mathcal{X})$ is the conditional distribution on Y: $f_\theta(x) = p(Y|x;\theta)$, and the corresponding loss function is the negative log-likelihood loss,

$$\ell(f_\theta(x), y) = -\log p(y|x;\theta)\,. \tag{2}$$

Minimizing the negative log-likelihood corresponds to the maximum likelihood estimate of parameter θ.

Since one does not have access to the unknown distribution $\mathcal{D}$, one is aiming to find θ such that both the empirical risk

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[\ell(f_\theta(X), Y)] \\ &= \frac{1}{|\bar{\mathcal{D}}|}\sum_{(x,y)\in\bar{\mathcal{D}}} \ell(f_\theta(x), y)\end{aligned} \tag{3}$$

and the generalization gap

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(f_\theta(X), Y)] - \mathcal{L}(\theta) \tag{4}$$

are small. To evaluate the generalization gap, one usually approximates the true distribution $\mathcal{D}$ by a finite set of withholding data drawn i.i.d. from $\mathcal{D}$. In general, there can be (infinitely) many θ that minimize the loss Eq. 3. Finding an appropriate θ so that predictions made by f_θ can generalize to unseen distribution $\mathcal{D}$ is fundamental in machine learning.

3. Approximating Sharpness Measure with Empirical Fisher Information

In (Kearns, 1989; Keskar et al., 2016) the authors argue that the sharpness of a minimum can be characterized by the magnitude of the eigenvalues of the Hessian matrix of loss function $\nabla^2_w \mathcal{L}(\theta)$. If θ is a local minimum, we know that $\mathcal{L}$ is locally convex within some neighborhood of θ, and therefore $\nabla^2_\theta \mathcal{L}(\theta)$ is positive semi-definite, and all its eigenvalues are non-negative. Then a reasonable measure of the eigenvalue magnitude is the trace of the Hessian, as $\mathrm{tr}(\nabla^2_\theta \mathcal{L}(\theta)) = \sum_{i=1}^n \lambda_i$. For high-dimensional problems, instantiating the Hessian and computing its trace is computationally infeasible. For classification problems with negative log-likelihood loss $\mathcal{L}(\theta) = -\mathbb{E}_{X,Y\sim\bar{\mathcal{D}}}[\log p(Y|X;\theta)|\theta]$, we can nevertheless approximate the trace by leveraging a connection between the Hessian and Fisher information matrix at minima, under the assumption that the model is over-specified, i.e. that the minimum θ^* overfits the training set. In particular, recall that the Fisher information F on the training set $\bar{\mathcal{D}}$ is defined as (Martens, 2014),

$$F_\theta \triangleq -\mathbb{E}_{X\sim\bar{\mathcal{D}}, Y\sim p(Y|X;\theta)}[\nabla^2_\theta \log p(Y|X;\theta)]\,, \tag{5}$$

where Y is drawn from the model distribution $p(Y|x;\theta)$. Under suitable regularity conditions, an integration-by-parts identity gives F_θ the expected outer product of gradients,

$$\begin{aligned} F_\theta &= \mathbb{E}_{X\sim\bar{\mathcal{D}}, Y\sim p(Y|X;\theta)}[G(\theta)G(\theta)^\top]\,, \\ &\quad G(\theta) = \nabla_\theta \log p(Y|X;\theta) \end{aligned} \tag{6}$$

At a minimizer of an over-specified model, the model distribution is close to the true label distribution of $\bar{\mathcal{D}}$ and the Fisher approximates the Hessian,

$$F_{\theta^*} \approx -\mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[\nabla^2_\theta \log p(Y|X;\theta^*)] = \nabla^2_\theta \mathcal{L}(\theta^*)\,. \tag{7}$$

We can similarly approximate the model distribution with the training set distribution in eqn. (6) giving,

$$\bar{F}_{\theta^*} \triangleq \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[G(\theta^*)G(\theta^*)^\top]\,. \tag{8}$$

The matrix $\bar{F}$ is referred to as the "empirical Fisher" (Martens, 2014), and it approaches both the true Fisher and the Hessian as the model overfits the training set. Moreover, eqn. (8) may be interpreted as the covariance matrix of gradients of the loss function. Finally, we conclude that the trace of the Hessian at minimum θ^* can be approximated

Table 1. Test accuracy of various fully-connected networks trained on CIFAR-10

	No regularization	L_2 only	Curvature only	Curvature and L_2
3 layers of 512 units	64.97%	70.16%	69.98%	**71.65%**
3 layers of 4096 units	68.06%	71.36%	69.50%	**74.07%**

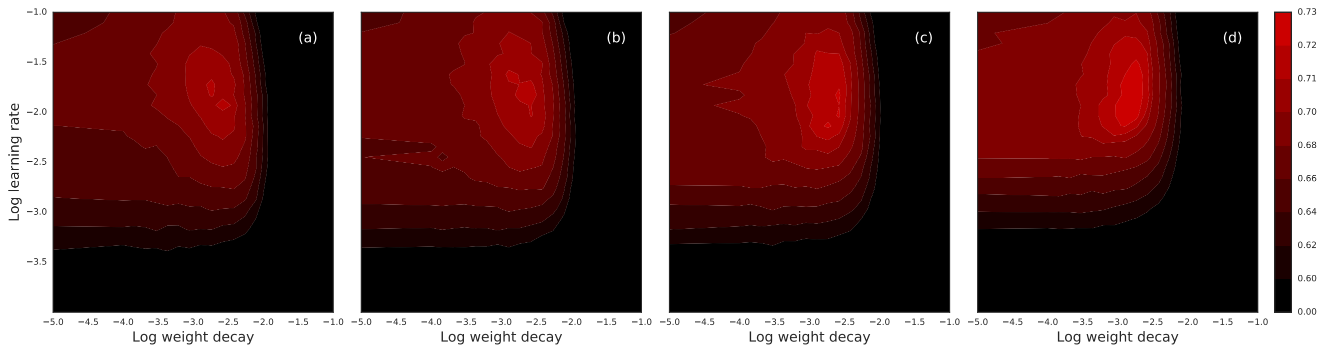

Figure 1. Test performance of a fully-connected three-hidden-layer width-4096 ReLU network trained on CIFAR-10 using SGD with batch size 256, plotted as a function of learning rate and weight decay, for curvature regularization coefficient (a) $\alpha = 0$, (b) $\alpha =$ 10,000, (c) $\alpha =$ 50,000, and (d) $\alpha =$ 200,000. Non-zero curvature regularization increases both the maximum generalization performance and the area of hyperparameter space corresponding to good performance.

by,

$$\begin{aligned}\operatorname{tr}(\nabla_\theta^2 \mathcal{L}(\theta^*)) &\approx \operatorname{tr}(\bar{F}_{\theta^*}) \\ &= \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[\operatorname{tr}(G(\theta^*)G(\theta^*)^\top)] \\ &= \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[G(\theta^*)^\top G(\theta^*)] \\ &= \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[\|\nabla_\theta \ell(f_{\theta^*}(X), Y)\|_2^2]\,.\end{aligned} \tag{9}$$

This quantity is merely the expected per-example squared L_2 norm of the gradient. It does not require the Hessian or Fisher to be explicitly materialized and can be computed efficiently.

4. Curvature Regularization

The expected gradient norm squared defined in eqn. (9) provides a computationally-efficient method for approximating the trace of the Hessian. A natural question to ask is whether we can incorporate this measure as part of the loss function to bias the gradient descent process into finding wider minima. The modified loss function we consider is,

$$\begin{aligned}\bar{\mathcal{L}}(\theta) &\triangleq \mathcal{L}(\theta) + \alpha \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[\|\nabla_\theta \ell(X,Y|\theta)\|_2^2] \\ &= \mathbb{E}_{(X,Y)\sim\bar{\mathcal{D}}}[\ell(f_{\theta^*}(X),Y) + \alpha\|\nabla_\theta \ell(f_\theta(X),Y)\|_2^2]\end{aligned} \tag{10}$$

where α is a positive hyperparameter controlling the strength of the regularization. Because the trace of the Hessian measures the mean curvature at a critical point, with some abuse of notation we refer to this type of regularization as "curvature" regularization.

While eqn. (10) defines the global loss function, in practice it will be optimized with stochastic gradient descent on mini-batches. Concretely, for each iteration a mini-batch $\mathcal{B} \subset \bar{\mathcal{D}}$ is sampled, and the loss function is computed on $\mathcal{B}$ as,

$$\begin{aligned}\bar{\mathcal{L}}_{\mathcal{B}}(\theta) &= \mathbb{E}_{(X,Y)\sim\mathcal{B}}[\ell(X,Y|\theta) + \alpha\|\nabla_\theta \ell(X,Y|\theta)\|_2^2] \\ &= \frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}} \ell(x,y|\theta) + \alpha\|\nabla_\theta \ell(x,y|\theta)\|_2^2\,.\end{aligned} \tag{11}$$

The computational overhead of this regularization method comes from the evaluation of $\|\nabla_\theta \ell(x,y|\theta)\|_2^2$, i.e. the squared gradient norm. Crucially, this gradient norm is computed *per example*; the average per-example gradient squared norm is not equal to the squared norm of the average gradient. A naïve implementation of per-example gradient computation would involve an explicit loop over the examples in a batch and would prevent efficient utilization of fast linear algebra and instruction-level parallelism. Nevertheless, the specific structure of back-propagation for many neural network architectures admits an efficient implementation that we now briefly describe.

Consider for simplicity a fully-connected feedforward architecture. Let $\{x_i; 1 \le i \le |\mathcal{B}|\}$ denote the post-activation units of a given layer and $y_i = Wx_i + b$ the pre-activation units of the following layer. Now suppose the loss for example i is ℓ_i, and $\bar{\mathcal{L}}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{1\le i\le|\mathcal{B}|}\ell_i$. Then the average gradient L_2 norm squared for the weights W is,

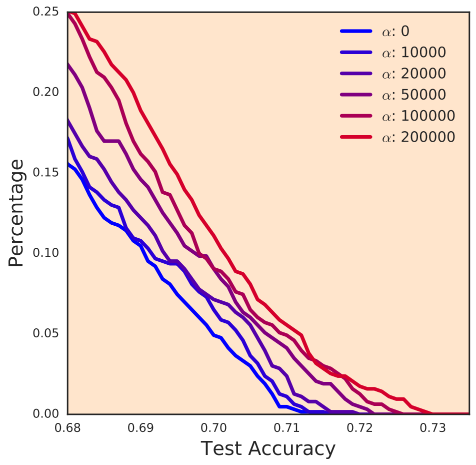

Figure 2. Percentage of models from Figure 1 yielding performance above various test accuracy thresholds for different strengths α of curvature regularization. Higher percentages indicate that a larger fraction of the weight-decay/learning rate hyperparameter plane corresponds good-performing models. Curvature regularization increases this fraction, suggesting that the regularizer may allow not only for better generalization performance but also for easier hyperparameter tuning.

$$\begin{aligned}\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left\|\frac{\partial \ell_i}{\partial W}\right\|_F^2 &= \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left\|\frac{\partial \ell_i}{\partial y_i}x_i^\top\right\|_F^2 \\ &= \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left\|\frac{\partial \ell_i}{\partial y_i}\right\|_2^2 \|x_i\|_2^2 \,,\end{aligned} \tag{12}$$

where we have defined the back-propagated error signals as $\delta_i = \partial \ell_i / \partial y_i$. Note that both δ and x are already computed during the course of the normal back-propagation algorithm, so evaluating eqn. (12) requires only minimal computational overhead. Gradients of this expression can also be readily computed using standard automatic differentiation procedures available in the major deep learning frameworks.

In practice, we find that adding this regularizer incurs a marginal computational cost of 15% in the fully-connected case. Generalizing this type of efficient implementation to other archictures such as convolutional networks is also possible. We leave to future work the analysis of more complicated configurations such as batch normalization and residual connections.

5. Experiments

We evaluate the performance of the curvature regularizer on CIFAR-10 and CIFAR-100 datasets with fully-connected networks and all-convolutional networks. The fully-connected network consists of three hidden layers of 4096 units, and the all-convolutional network is a nine-layer network with average pooling in the last layer. We use standard SGD with momentum to optimize these models. For the baseline models, we tune the learning rate and weight decay and report the best result. For the curvature regularized version, we tune an extra hyperparameter, namely the coefficient α in eqns. (10, 11). The result of an extensive search over hyperparameters is summarized in Table 1 for the fully-connected architecture and Table 2 for the convolutional case. The addition of the regularizer significantly improves the generalization performance. Furthermore, from the ablation study between the curvature and L_2 regularization, we can see that the curvature regularization functions orthogonally to L_2 regularization, which indicates that both should be used in practice.

Table 2. CNN performance on CIFAR-10 and CIFAR-100

Test accuracy	L_2 only	Curvature and L_2
CIFAR-10	92.83%	**93.54%**
CIFAR-100	70.68%	**73.39%**

Figure 1 summarizes the result of a large hyperparamter grid search for the fully-connected network in which we scanned over learning rate, weight decay, and curvature regularization. This analysis reveals that test performance can be quite sensitive to the learning rate and amount of weight decay, but that this sensitivity is reduced in the presence of curvature regularization. We quantify this effect by measuring the fraction of hyperparameter configurations yielding test accuracies greater than a given threshhold. Figure 2 shows that increasing the strength of curvature regularization indeed increases the volume of viable hyperparameter space and suggests that curvature regularization may allow not only for better generalization performance but also for easier hyperparameter tuning.

6. Discussion and Future Directions

In this note, we presented a computationally-efficient method for approximating the mean curvature of the loss surface at minima, and investigated the utility of this function as a regularizer in deep learning. Our experiments demonstrate that such “curvature” regularization can yield improved generalization performance and can enable more robust hyperparameter selection.

Efficient implementation of the regularizer is straightforward for fully-connected networks, but doing so within standard deep learning frameworks like TensorFlow can present some engineering challenges for more complicated architectures. Additionally, batch normalization presents a conceptual challenge since the loss function does not decompose into a sum over per-example losses. We believe the results presented here provide strong motivation to tackle these challenges in future work.

References

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Chaudhari, Pratik, Choromanska, Anna, Soatto, Stefano, LeCun, Yann, Baldassi, Carlo, Borgs, Christian, Chayes, Jennifer, Sagun, Levent, and Zecchina, Riccardo. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

Dinh, Laurent, Pascanu, Razvan, Bengio, Samy, and Bengio, Yoshua. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

Goyal, Priya, Dollár, Piotr, Girshick, Ross, Noordhuis, Pieter, Wesolowski, Lukasz, Kyrola, Aapo, Tulloch, Andrew, Jia, Yangqing, and He, Kaiming. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hochreiter, Sepp and Schmidhuber, Jürgen. Simplifying neural nets by discovering flat minima. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, pp. 529–536, Cambridge, MA, USA, 1994. MIT Press. URL http://dl.acm.org/citation.cfm?id=2998687.2998753.

Kearns, M. J. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.

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

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Luong, Thang, Pham, Hieu, and Manning, Christopher D. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

Martens, James. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Graves, Alex, Antonoglou, Ioannis, Wierstra, Daan, and Riedmiller, Martin. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

Oquab, Maxime, Bottou, Leon, Laptev, Ivan, and Sivic, Josef. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1717–1724. IEEE, 2014.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Smith, Sam and Le, Quoc V. A bayesian perspective on generalization and stochastic gradient descent. 2018. URL https://openreview.net/pdf?id=BJij4yg0Z.

Szegedy, Christian, Ioffe, Sergey, Vanhoucke, Vincent, and Alemi, Alexander A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, pp. 12, 2017.

Wilson, Ashia C, Roelofs, Rebecca, Stern, Mitchell, Srebro, Nati, and Recht, Benjamin. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4151–4161, 2017.