MAdam: Modify Second Momentum Estimation in Adam

Yanting Miao Y43MIAO@UWATERLOO.CA
STUDENT ID: 20841196
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, CA

Abstract

I introduce a new hyperparameter in Adam and propose a new Adam variant, *MAdam*. MAdam is the first Adam variant adaptive estimate step size via learning rate, the direction of the current gradient and the direction of exponential moving average (EMA) of gradients. Further, MAdam is stable in training GAN and robust in different learning rates. Meanwhile, MAdam has a fast convergence speed as Adam and other variants in training deep convlutional neural networks. Additionally, I present a simple formula to fine-tune the new hyperparameter and give a clear theoretical analysis of effective step size of MAdam in different learning rate cases. Last, I provide multiple experiments to validate MAdam and other optimizers and the empirical results show that MAdam outperforms other optimization algorithm in training Generative Adversarial Networks (GAN) and achieve fast convergences and high accuracy rates in the image classification task. MAdam code and experiments code can be found at `https://github.com/andrew-miao/CS480-680-Project`.

Keywords: Deep Learning, Optimization Algorithm, GAN, CNN

1. Introduction

The first-order optimization algorithms have been highly successful in training modern deep neural networks. Typically, the Stochastic Gradient Descent algorithm (SGD) (Robbins and Monro, 1951), performs a central role in many machine learning stories due to its simplicity and effectiveness. To achieve fast convergence, Nesterov proposed an accelerated gradient scheme (NAG) (Nesterov, 1983) based on SGD and Ilya described SGD with momentum (Sutskever et al., 2013). These SGD based methods use a fixed learning rate for all parameters and attempt to find a "proper" direction to accelerate training. Recently, there is another type of accelerated scheme: adaptive learning rate methods, such as RMSProp (Hinton et al.) and Adam (Kingma and Ba, 2014). Unlike SGD variants, adaptive methods update learning rates for parameters during training.

The most dominant adaptive algorithm, Adam, provides faster convergence than SGD variants in the training phase. Hence, researchers use Adam as optimizer to train modern deep learning models, including Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2018). However, Adam is sensitive to learning rate and may converge to bad local optima, and thus, to solve these problems, many Adam variants are proposed in recent years. To improve robustness of Adam in extreme large and small learning rates, AdaBound (Luo et al., 2019) introduces a smooth transition from adaptive methods to SGD. To avoid Adam converge to a suspicious local optima, Liu et al. presented RAdam, which rectify variances of learning rate. Further, Zhuang et al. described AdaBelief to speed up Adam in low-variance case. Compare to Adam, although these variants achieve a similar or better accuracy rate in the image classification task, many optimizers are unstable to train Generative Adversarial Networks (GAN) (Goodfellow et al., 2014), and the performances of optimizers are much worse than Adam in training GAN task.

To enhance the stability, robustness, and convergence speed of Adam optimizer, I propose MAdam, a new Adam variant, which introduce the third hyperparameter in vanilla Adam. For simplicity, I use the same notation as Adam, g_t is the gradient at step t and its exponential moving average (EMA) is m_t. Before second momentum estimation, MAdam updates g_t as $\beta_3 g_t + (1 - \beta_3)(\frac{g_t - m_t}{\beta 1})$. Then, MAdam computes the EMA of updated g_t^2 as v_t. Typically, MAdam use the same update rule as Adam: computing $\frac{m_t}{\sqrt{v_t}}$. Intuitively, if the direction of g_t is close to its EMA m_t, MAdam takes larger step size than Adam, and MAdam will take a smaller step than Adam when g_t deviates from m_t, which means there is a uncertainty in the direction of gradient. Further, the new hyperparameter β_3 ameliorate the robustness of the optimizer and this hyperparameter is easy to fine-tuned and the more details about fine-tuning β_3 will show in the next section. My contributions can be summarized as:

- I propose MAdam by introducing a new hyperparameter in Adam, which provides faster convergence speed than Adam in theoretically.
- I present a very simple way to fine-tune this new hyperparameter and the stability is improved by the new hyperparameter.
- I validate the performance of MAdam with multiple experiments. Compare to Adam and other Adam variants, MAdam achieves a better quality of generated images in the training of GAN. In image classification tasks, MAdam achieves similar even better accuracy rates with Adam and other optimizers.

2. Methods

In this section, I will introduce MAdam algorithm. Meanwhile, to make comparison with Adam, this section also reviews Adam and points out the disadvantages of Adam and other variants. Consider convention and to make easy understanding, I will use similar notations as the Adam algorithm. The notations in MAdam are in the following.

- $f(\theta)$: Objective function with parameter vectors θ.
- β_1, β_2: Hyperparameters of exponential decay rates, $\beta_1, \beta_2 \in [0, 1)$, in the default setting, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
- α: Learning rate, in the default setting, $\alpha = 0.001$.
- ϵ: A small value to prevent overflow problem in practice, in the default setting, $\epsilon = 10^{-16}$
- β_3: The new hyperparameter controls the weight of deviation and depends on learning rate α, $\beta_3 \geq 0$, in the default setting, $\beta_3 = 0.9$.

- $\psi(\cdot)$: The update rule, in the default setting, $\psi(\cdot)$ utilizes the update rule in RAdam (Liu et al., 2019) (RAdam also computes $m_t/\sqrt{v_t}$), to avoid converge in a bad local optima in the early training stage.

Algorithm 1 *Adam*, all operations of vectors are element-wise

```
Initialize:
    $m_0 \leftarrow 0, v_0 \leftarrow 0, t \leftarrow 0$
while $\theta_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$
    $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$
    $\theta_t \leftarrow \theta_{t-1} - \alpha \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$
end while
return $\theta_t$
```

Algorithm 2 *MAdam*, all operations of vectors are element-wise

```
Initialize:
    $m_0 \leftarrow 0, v_0 \leftarrow 0, t \leftarrow 0$
while $\theta_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $\tilde{g_t} \leftarrow \beta_3 g_t + (1 - \beta_3)(g_t - m_t)/\beta_1$
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) \tilde{g_t}^2$
    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$
    $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$
    $\theta_t \leftarrow \psi(\theta_{t-1}, \alpha, \hat{m}_t, \hat{v}_t, \epsilon)$
end while
return $\theta_t$
```

2.1 Intuitive and Details of MAdam

Algorithm 1 and algorithm 2 shows the pseudo-code for Adam and MAdam respectively. The new hyperparameter and variable are marked in red. For simplicity, in this subsection, $\beta_3 < 1$ and ψ set as the update rule of Adam. MAdam is inspired by the RProp algorithm (Riedmiller and Braun, 1993). In Rprop algorithm, if the sign of the current gradient is the same as the sign of the last gradient, Rprop increases step size, otherwise, Rprop reduces step size. MAdam incorporates this idea, note that I update $\hat{g}_t$ before the second momentum estimation, and $\hat{g}_t = \beta_3 g_t + (1 - \beta_3)(g_t - m_t)/\beta_1$ can be rewritten as:

$$\begin{aligned} \hat{g}_t &= \beta_3 g_t + \frac{1 - \beta_3}{\beta_1}(g_t - \beta_1 m_{t-1} - (1 - \beta_1) g_t) \\ &= \beta_3 g_t + (1 - \beta_3)(g_t - m_{t-1}) \\ &= g_t - (1 - \beta_3) m_{t-1}. \end{aligned} \tag{1}$$

If the direction of g_t is close to m_{t-1}, similar to the current gradient has the same sign as the last gradient in Rprop, $\|\hat{g}_t\| = \|g_t - (1 - \beta_3) m_{t-1}\| < \|g_t\|$ will be a small value and increase step size, which is larger than Adam's step size. When the direction of g_t diverging with its EMA m_t, similar to the current gradient has a different sign from the sign of the last gradient, $\|\hat{g}_t\| = \|g_t - (1 - \beta_3) m_{t-1}\| > \|g_t\|$, MAdam reduce the step size, which will be smaller than the step size of Adam in this case. To derive more precise effective step size, I use the assumption from Kingma and Ba: $\left\|\hat{m}_t/\sqrt{\hat{v}_t}\right\| \approx \left\|\mathbb{E}[g]/\sqrt{\mathbb{E}[g^2]}\right\|$. Based this assumption, the effective step size of MAdam can be written as:

$$\begin{aligned}
\left\|\Delta_t^{MAdam}\right\| &\approx \alpha \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathbb{E}[\hat{g}^2]}} \right\| \\
&= \alpha \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathbb{E}[(g-(1-\beta_3)m)^2]}} \right\| \\
&\approx \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathbb{E}[(g-(1-\beta_3)E[g])^2]}} \right\| \because m \approx \mathbb{E}[g] \\
&= \alpha \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathrm{Var}(g)+\beta_3^2\mathbb{E}[g]^2}} \right\| \because \mathrm{Var}(g) = \mathbb{E}[g^2] - \mathbb{E}[g]^2.
\end{aligned} \tag{2}$$

Similarly, the effective step size of Adam can be written as,

$$\left\|\Delta_t^{Adam}\right\| = \alpha \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathrm{Var}(g)+\mathbb{E}[g]^2}} \right\| \leq \alpha. \tag{3}$$

Assuming gradients have low variance in parameter space, then I will derive the following result: $\frac{\left\|\Delta_t^{MAdam}\right\|}{\left\|\Delta_t^{Adam}\right\|} = \frac{1}{\beta_3}$. In the default setting, $\beta_3 = 0.9 < 1$, and thus, MAdam can be considered as accelerated Adam in low variance case. In the default setting, $\beta_3 = 0.9$, which means the upper bound Δ_t^{MAdam} is close to α. In this year, there is a new Adam variant, AdaBelief (Zhuang et al., 2020), which also computes $g_t - m_t$ in v_t. However, this algorithm doesn't calculate the bias correction of $g_t - m_t$, and AdaBelief computes the effective step size only based on variance of gradients. The effective step size of AdaBelief (different from original paper due to the bias correction) is

$$\left\|\Delta_t^{AdaBelief}\right\| = \frac{\alpha}{\beta_1} \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathrm{Var}(g)}} \right\|. \tag{4}$$

Although AdaBelief has a large effective step size, this algorithm is easy to overshoot. In my experiments, this algorithm is unstable in training GAN and fail to converge in large learning rate and default learning rate cases. Particularly, Adam and AdaBelief is two special cases of MAdam, when $\beta_3 = 1$, MAdam perform exactly same as Adam, and MAdam become AdaBelief when $\beta_3 = 0$.

2.2 The New Hyperparameter

In the previous subsection, I assume $\beta_3 < 1$. Meanwhile, I also point out Adam and AdaBelief are special cases of MAdam. In this subsection, I will present the fine-tune method of this hyperparameter, and discuss the case of $\beta_3 > 1$, which improve the stability of MAdam. For a new learning rate $\tilde{\alpha}$, I calculate the new hyperparameter $\tilde{\beta_3}$ via the following formula,

$$\tilde{\beta_3} = \begin{cases} \beta_3 & \text{if } \frac{1}{2} \leq \frac{\tilde{\alpha}}{\alpha} < 2, \\ \frac{2\tilde{\alpha}}{\alpha}\beta_3 & \text{otherwise}, \end{cases}$$

where β_3 and α are default values in MAdam. Considering a extreme large learning, the hyperparameter $\tilde{\beta_3}$ also will increase to an extreme large value. According to the equation 2, the effective step size of MAdam in extreme large learning case is

$$\left\|\Delta_t^{MAdam}\right\| \approx \tilde{\alpha} \left\| \frac{\mathbb{E}[g]}{\sqrt{\tilde{\beta_3}^2 \mathbb{E}[g]^2}} \right\| = \frac{\alpha}{2\beta_3} < \alpha \tag{5}$$

Hence, MAdam will not take a large step even in an extremely large learning rate case and holds an upper bound which is the same as Adam in the normal learning rates. Because of the existence of the upper bound in large learning cases, MAdam is more stable compare to other optimizers. For extreme small learning rates, $\tilde{\beta_3}$ is close to zero, the step size will be

$$\left\|\Delta_t^{MAdam}\right\| \approx \alpha \left\| \frac{\mathbb{E}[g]}{\sqrt{\mathrm{Var}(g)}} \right\|, \tag{6}$$

which is similar to AdaBelief. This result is reasonable, in small learning rates, the ideal optimizer should take a large step to explore the direction of descent.

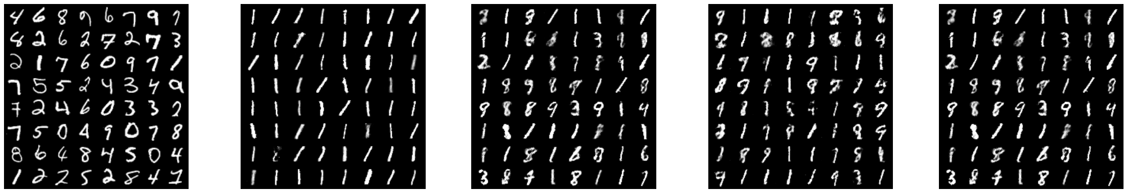
3. Experiments

To empirically evaluate MAdam, I designed multiple experiments to compare with four different optimizers, including Adam (Kingma and Ba, 2014), AdaBelief (Zhuang et al., 2020), AdaBound (Luo et al., 2019), and SGD (Robbins and Monro, 1951). Further, I investigated four different novel deep learning models, including GAN (Goodfellow et al., 2014), ResNet34 (He et al., 2016), ShuffleNet (Zhang et al., 2018), and MobileNet (Howard et al., 2017). By experiments in these convolutional neural networks, MAdam can be an efficient optimization algorithm in deep learning models.

I used the fine-tuned formula in section 2.2 to calculate different β_3 for different learning rates in training GAN, and other optimizers used the default hyperparameters in experiments. Due to the time constrain, I haven't test different learning rates in training image classification models, and thus, in image classification tasks, MAdam and all Adam variants optimizers use the default hyperparameters, the momentum set as 0.9 in SGD.

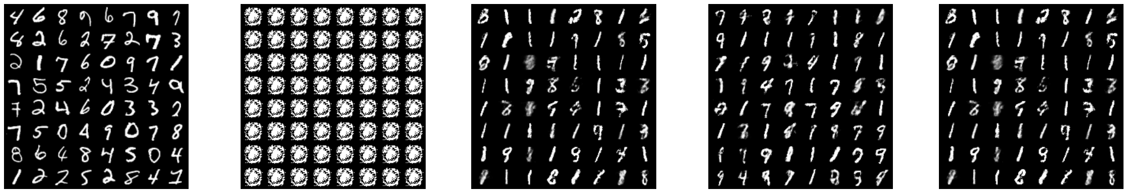
3.1 Experiment: GAN

In practice, recent Adam variants focused on image classification or language modeling tasks; however, there are not ample experimental validations for more complex neural networks, such as GAN. Hence, I experiment MAdam and other optimizers in GAN to validate the stability of different optimizers and alter learning rates to examine robustness of these optimizers. In this simple GAN, I utilized two multi-layer fully connected neural networks as generator and discriminator respectively in MNIST. The generator produces images from 200 dimension noise vectors and discriminator classifies real/fake images based on 784 dimension image vectors. Each optimizer uses minibatch size of 128. GAN is trained via each optimizer for 50 epochs and sampled the generated image from the last epoch.



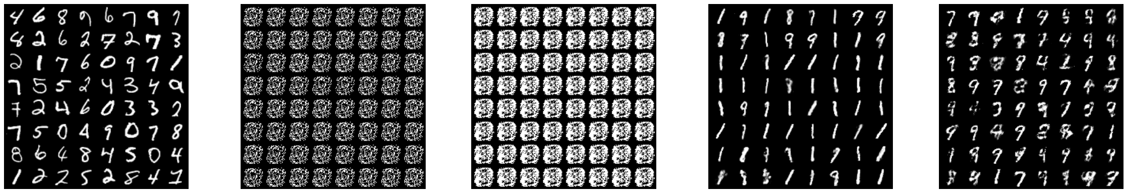
(a) real images (b) $\alpha = 0.003$ (c) $\alpha = 0.001$ (d) $\alpha = 0.0003$ (e) $\alpha = 0.0001$

Figure 1: Left to right: real images from MNIST, samples from GAN with different learning rates, all models are trained by MAdam. According to my fine-tune formula, $\beta_3 = \{5.4, 0.9, 0.6, 0.18\}$ in (b) to (e) respectively.



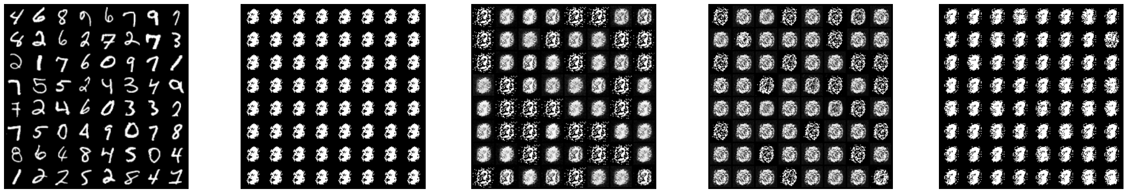
(a) real images (b) $\alpha = 0.003$ (c) $\alpha = 0.001$ (d) $\alpha = 0.0003$ (e) $\alpha = 0.0001$

Figure 2: Left to right: real images from MNIST, samples from GAN with different learning rates, all models are trained by Adam.



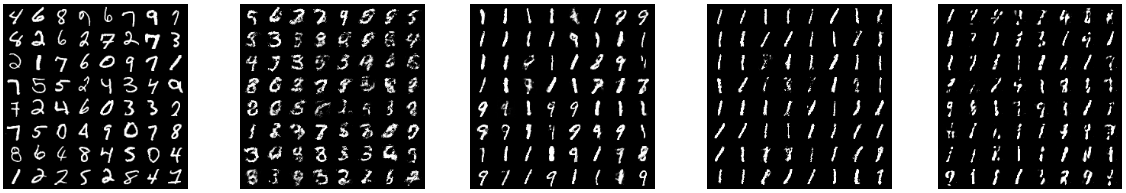
(a) real images (b) $\alpha = 0.003$ (c) $\alpha = 0.001$ (d) $\alpha = 0.0003$ (e) $\alpha = 0.0001$

Figure 3: Left to right: real images from MNIST, samples from GAN with different learning rates, all models are trained by AdaBelief.



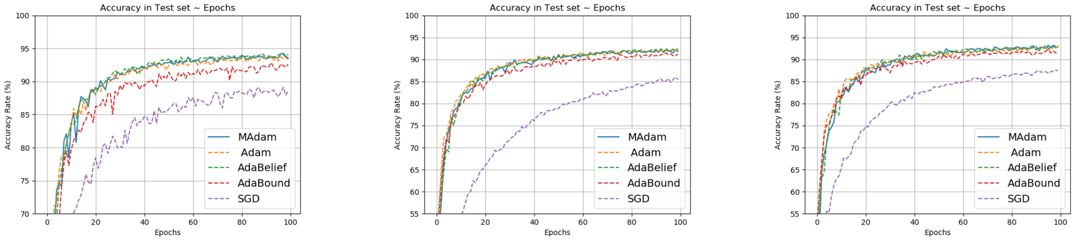
(a) real images (b) $\alpha = 0.003$ (c) $\alpha = 0.001$ (d) $\alpha = 0.0003$ (e) $\alpha = 0.0001$

Figure 4: Left to right: real images from MNIST, samples from GAN with different learning rates, all models are trained by AdaBound.



(a) real images (b) $\alpha = 0.003$ (c) $\alpha = 0.001$ (d) $\alpha = 0.0003$ (e) $\alpha = 0.0001$

Figure 5: Left to right: real images from MNIST, samples from GAN with different learning rates, all models are trained by SGD.



(a) ResNet34 (b) ShuffleNet (c) MobileNet

Figure 6: Left to right: Test accuracy of different models on CIFAR-10

The generated images are shown in Figure 1 to Figure 5. For the large learning rate case ($\alpha = 0.003$), MAdam outperforms Adam and its variants. It is clearly to observe that Adam and its variants fail to converge in the large learning rate case. Particularly, AdaBound diverges in complex neural networks whether the learning rate is large or small. For default and small learning rates cases, the performance of MAdam is consistent with my theoretical analysis in 2.1. MAdam provides less blur digits images in the small learning rates due to MAdam increases step size in this case. Further, there is an interesting point, researchers usually think SGD is not suitable in GAN given that SGD generates the mode collapse issue. This claim holds in my experiments when the learning rate is small. Nevertheless, SGD seems to escape from mode collapse trap and produce diverse digits, which is worth for future research. Meanwhile, MAdam and SGD are only two optimizers that can converge in different learning rates in GAN experiments, which indicates MAdam may have good generalization performance as SGD.

3.2 Experiment: Convolutional Neural Networks

To experiment the performance of optimizers in Convolutional Neural Networks (CNN): ResNet34, ShuffleNet, and MobileNet. I experiment these models on CIFAR-10 dataset to examine the effectiveness of MAdam in training deep CNN models for image classification tasks. For more precise, each optimizer train each model for 100 epochs in three times, and I choose the average accuracy rate of each epoch as the result. Figure 6 shows MAdam achieve fast convergences as Adam and its variants in different deep CNNs. Meanwhile, MAdam also reaches high accuracy rates in image classification as Adam and its variants do.

4. Conclusion and Future Works

I introduce a new hyperparameter in Adam and propose the MAdam optimization algorithm for deep learning models. MAdam can scale step size adaptively by the learning rate and the direction of the current gradient and its EMA. Further, I describe a simple fine-tune formula to calculate the new hyperparameter and present a theoretical analysis of the step size of MAdam.To my best knowledge, MAdam is the first Adam variant scale step size by different learning rates. Therefore, MAdam also is very robust for extreme learning rate cases. Meanwhile, I design multiple experiments to validate the stability, robustness, and fast convergence speed of MAdam on MNIST and CIFAR-10 dataset.

For future works, the robustness of optimizers hasn't tested in deep CNN models and researchers often set epochs as 200 in training deep neural networks in practice, and thus, it is worth observing the performance of MAdam and other optimizers in a large number of epochs. I haven't experimented with MAdam in Nature Language Processing tasks, such as machine translation and language modeling. Therefore, I need to examine the performance of MAdam in recurrent neural networks and other complex modern deep learning architectures: Transformer and DCGAN. Although MAdam can converge in a large learning rate during training GAN, it still faces the mode collapse issue. Hence, there is a research space for alleviating mode collapse in MAdam.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

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

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

Y. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591 vol.1, 1993.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1139–III–1147. JMLR.org, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33, 2020.