Comparison between Adam and AdamW in simple tasks

Sunhwi Kim [1]

Abstract

To improve a generalization performance of adaptive gradient methods, one optimizer is proposed with the weight decay decoupled, escaping from L_2 regularization, called as AdamW. To verify the effect of this decoupling, this report compares two optimizers, Adam and AdamW in simple environments which consist of a supervised learning and an unsupervised learning with small size model and dataset, without a learning rate scheduler. Because the simplification makes difference from the AdamW paper experiment settings, two optimizers are compared on new situations, post-overfit and regularization in a loss function, which is not L_2 regularization. These environment variation makes small but noticeable changes on behaviors of two optimizers. The code for comparison is on https://github.com/rlatjsgnl308/AI51101Project

1. Introduction

Adaptive gradient methods, such as Adagrad(Duchi et al., 2011), RMSprop(Tieleman, 2012), and Adam(Kingma & Ba, 2014), are widely chosen for training a deep learning model. Instead of same learning rate for each feature axis, adaptive gradient methods adjust each learning rate based on obtained function information. The enumerated optimizers make a diagonal pre-conditioning matrix which estimates the Hessian of an objective function from the calculated gradients, based on the fact that large curvature tends to make large magnitude of gradients. By dividing a square root of the estimated diagonal elements, the learning rate for each axis is calculated, which gives a similar effect with making a function level set more circular.

Especially, Adam takes the momentum method to remove noise from the stochasticity and differential curvature. The momentum method uses information in the history of gradients in the way that encourages directions with stable gradients but smooths out directions which make large fluctuation. Because this method gives faster convergence than the naive gradient descent on a training process, adaptive gradient optimizers are prevalent in a model train.

The vanilla gradient descent method, such as SGD (stochastic gradient descent), doesn't handle differential curvature in the objective function, making a convergence process noisy. But, several examples show that SGD, or with momentum, reduces generalization error more than adaptive gradient methods, even if the training loss is higher than adaptive gradient methods(Wilson et al., 2017). Among various hypotheses for poor generalization of the adaptive gradient methods, Ilya Loshchilov et al.(Loshchilov & Hutter, 2017) focus on the regularization method adopted for better generalization. They noticed that L_2 regularization is not effective for the adaptive gradient method. Based on this observation, they imposed the weight decay explicitly on an optimization process, and obtained less test loss.

This report validates the performance of AdamW with the similar experiments in AdamW paper on simpler model and dataset pairs, comparing with Adam. Two tasks are conducted, which consists of both a supervised learning and an unsupervised learning. Each model performs image classification and image generation with Adam and AdamW under various learning rate and weight decay settings. Two optimizers are analyzed in regards to the test statistics, and for image generation, qualitative performance comparison is also involved.

The main results are listed below.

- To exploit the benefit of the decoupled weight decay, there should be no overfit during training, because AdamW can make generalization worse after overfit than Adam.
- Adam has the strong dependency between learning rate and weight decay rate in the hyperparameter set which brings good generalization, regardless of the overfit situation.
- In VAE where training loss itself contains regularization for diversity, both optimizers don't show difference at wide area of weight decay rate.
- In spite of hyperparameter independence, AdamW

[1]School of Electrical Engineering, Ulsan National Institute of Science and Technology, Ulsan, Korea, shkim308@unist.ac.kr.

This reports borrows the format from *Proceedings of the 41st International Conference on Machine Learning*

shows higher sensitivity on learning rate change itself than Adam.

2. Related Works

2.1. Weight decay and L_2 regularization

To prevent a deep learning model from overfitting, several regularization methods are proposed. The weight decay(Hanson & Pratt, 1988) restricts a model complexity by decaying some weights to zero based on the equation 1.

$$\theta_{t+1} = (1-\lambda)\theta_t - \alpha\nabla f_t(\theta_t) \tag{1}$$

The solution, equation 2, of the equation 1 shows that each weight is exponentially decreasing, if the factor α is less than 1 and the gradient sequence get stabilized.

$$\theta_t = -(1-\lambda)\Sigma_{t=1}^{n}\alpha^{n-t}\nabla f_t(\theta_t) + \alpha^n\theta_0 \tag{2}$$

The effect of weight decay is easily obtained by L_2 regularization, which adds the squared norm of a parameter vector to the objective function as in equation 3.

$$f_t^{reg}(\theta_t) = f_t(\theta_t) + \frac{\lambda'}{2}||\theta_t||_2^2 \tag{3}$$

With the gradient descent update rule, L_2 regularization becomes same with weight decay, shown in equation 4, if $\lambda' = \lambda/\alpha$.

$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha(\nabla f_t(\theta_t) - \lambda'\theta_t) \\ &= (1-\alpha\lambda')\theta_t - \alpha\nabla f_t(\theta_t)\end{aligned} \tag{4}$$

Therefore, many optimizers implement the weight decay through L_2 regularization.

2.2. Adam with weight decay

Adaptive gradient methods, including Adam, also adopt L_2 regularization for generalization, but their test performance were worse than SGD in spited of better, or on-par, training loss(Wilson et al., 2017). There is an opinion that L_2 regularization is not effective weight decay implementation for Adam with respect to generalization. The algorithm 1 describes Adam, where all operations are element-wise. As shown in the algorithm, this process has two problem. The first one is that momentum and pre-conditioner include the regularization term, which hinders estimation of the objective function shape. The second is that the weight with the large pre-conditioning value is not regularized enough(Loshchilov & Hutter, 2017).

2.3. AdamW

To resolve above problems, AdamW(Loshchilov & Hutter, 2017) decouples weight decay with gradient descent process,

Algorithm 1 Adam with L_2 regularization

Input: learning rate α, weight decay rate λ, Adam parameters β_1, β_2, ϵ
Initialize: timestep t $\leftarrow$ 0, model parameter θ_0, first momentum vector m_0, second momentum vector v_0, schedule multiplier η_0
repeat
  t $\leftarrow$ t+1
  $\nabla f_t(\theta_{t-1}) \leftarrow$ SelectBatch(θ_{t-1})
  $g_t \leftarrow \nabla f_t(\theta_{t-1}) + \lambda\theta_{t-1}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)g_t$
  $v_t \leftarrow \beta_1 v_{t-1} + (1-\beta_1)g_t^2$
  $\hat{m}_t = m_t/(1-\beta_1^t)$
  $\hat{v}_t = v_t/(1-\beta_1^t)$
  $\eta_t \leftarrow$ SetScheduleMultiplier(t)
  $\theta_t \leftarrow \theta_{t-1} - \eta_t\alpha\hat{m}_t/\sqrt{\hat{v}_t+\epsilon}$
until $StoppingCriterion$ is met
Return: optimized parameters θ_t

as shown in the algorithm 2. The only difference is that weight decay becomes explicit, removing L_2 regularization. It makes the weight decay work as defined, which means all weight are regularized with same rate, and only information from the objective function is given for the momentum and pre-conditioning matrix. In addition, it doesn't make any coupling between decay rate and learning rate, $\lambda' = \lambda/\alpha$, like L_2 regularization.

3. Experiments

3.1. Model and dataset

3.1.1. VGG AND CIFAR-10

The first experiment is implemented on image classification, one of representative supervised learning tasks. A simple model, VGG net(Simonyan & Zisserman, 2014), is trained to classifiy images in a high accuracy. Among VGG net, the VGG11 with batch normalization is loaded from pytorch[1] without the pre-trained weight. The training dataset is CIFAR-10[2], which contains 32x32 colored images with 10 labels.

3.1.2. VAE AND FASHION MNIST

Image generation is selected as the second experiment for verification on unsupervised learning. A VAE(Variational AutoEncoder)(Kingma & Welling, 2013) is implemented with dense layers and trained to generate Fashion MNIST[3] dataset, which contains 28x28 gray image with 10 labels.

[1] https://pytorch.org/hub/pytorch_vision_vgg/
[2] Details about CIFAR-10:https://www.cs.toronto.edu/ kriz/cifar.html
[3] https://github.com/zalandoresearch/fashion-mnist

Algorithm 2 AdamW

Input: learning rate α, weight decay rate λ, Adam parameters β_1, β_2, ϵ
Initialize: timestep t $\leftarrow$ 0, model parameter θ_0, first momentum vector m_0, second momentum vector v_0, schedule multiplier η_0
repeat
t $\leftarrow$ t+1
$\nabla f_t(\theta_{t-1}) \leftarrow$ SelectBatch(θ_{t-1})
$g_t \leftarrow \nabla f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
$v_t \leftarrow \beta_1 v_{t-1} + (1-\beta_1) g_t^2$
$\hat{m}_t = m_t / (1-\beta_1^t)$
$\hat{v}_t = v_t / (1-\beta_1^t)$
$\eta_t \leftarrow$ SetScheduleMultiplier(t)
$\theta_t \leftarrow \theta_{t-1} - \eta_t (\alpha \hat{m}_t / \sqrt{\hat{v}_t + \epsilon} + \lambda \theta_{t-1})$
until $StoppingCriterion$ is met
Return: optimized parameters θ_t

Layer	Encoder	Decoder
First	784x500	4x500
Second	500x4	500x784

Table 1. Layers of the dense VAE. 784 is from the square of the number of pixels in one side of images. For nonlinearity, only ReLU is adopted. The latent variable is set to 4 dimension vector.

The detailed information of the model is given in the table 1. To estimate mean and log-variance for the latent space, dense layers with the same dimension is utilized.

3.2. Test setting and hyperparameters

There are two kinds of test for each task, grid test and training test. In the grid test, the test performance of a model is evaluated on a grid of learning rate and weight decay rate pairs. To make a grid, learning rate and weight decay rate are set to (0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001, 0.00005, 0.00002, 0.00001) and (10, 5, 2, 1, 0.5, 0.2, 0.1, 0.05, 0.02), respectively. The weight decay rate is multiplied with 0.001. In the training test, the test statistics during training is compared with two optimizers under different weight decay rate. The learning rate is fixed to 0.0001, and the weight decay rate setting is same with the grid test.

For other hyperparmeters of optimizers, the default settings are used, which is served by pytorch. For both tasks, the number of epochs is set to 200 for the training test and 100 for the grid test. The same batch size, 128, is utilized, and the results are averaged on multiple trials with two seeds, 7 and 77. To observe the effect of only weight decay decoupling, there is no learning rate scheduling.

3.3. Results

3.3.1. VGG AND CIFAR-10

At the training test, AdamW shows lower training loss (high training accuracy) than Adam, with inferior generalization performance, which is totally contrary to what the AdamW paper(Loshchilov & Hutter, 2017) mentioned.

There are two possible reasons for this unexpected outcome. First, the number of epochs is not enough for AdamW to shows its effectiveness, because the AdamW paper run much larger epochs for training (200 vs 1800). However, in the figure 1, the severe overfit is observed for both Adam and AdamW. With AdamW, the VGG model works well for training dataset with loss near 0 and accuracy near 1, but test performance doesn't follow the train phase tendency. It means that this test tried already the excessive number of epoch for generalization.

In spited of the decreased model size, the overfit is caused by less data augmentation, random cropping with a fixed size and only horizontal flipping, than the AdamW paper. The AdamW paper 'applied the regular data augmentation procedure', which is inferred to standard translation and flipping because the experimental setting followed the shake-shake regularization(Gastaldi, 2017). The effect of random cropping is somewhat similar to image translation, especially if the translation only remains the corner, but definitely this model is blind to the feature from the vertical flipping.

The second opinion is that the overfit makes the situation different. Important is that the AdamW paper didn't show results after overfit with test statistics non-increasing trend. The training behavior of the VGG model represents that AdamW becomes worse than Adam under the overfit situation. This is supported by the test loss and accuracy at the time the overfit is starting (test loss get increased) in the figure 1. Before the minimum test loss is obtained, AdamW get lower loss than Adam, and slightly higher test accuracy generally, which is matched with the consequence of the AdamW paper. However, after the minimum test loss point, AdamW increases test loss more steeply than Adam, and shows worse generalization performance with most of weight decay rate.

The figure 1 indicates that the overfit get relived as an weight decay rate increases with Adam. It means that different from AdamW, Adam has a robust generalization performance with a high weight decay rate under the overfit situation. The possible reason is the decoupled weight decay. Weight decay method can prevent the overfit, but not block it totally. After overfit occurs, AdamW still obtains the exact empirical loss function information, such as a pre-conditioning matrix, which makes the overfit worse. Although the weight decay process after the descent, it just recursively injects overfitted weights. However, Adam has some noise in the gradient

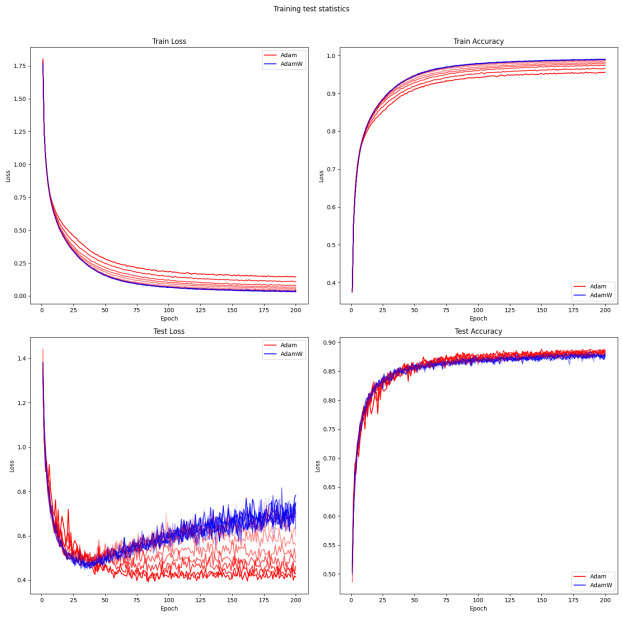


Figure 1. This figure shows the whole training process. The first row gives training loss and accuracy, and the second row gives test loss and accuracy. It indicates that there was the overfit during the training procedure. A low weight decay rate is visualized with high transparency.

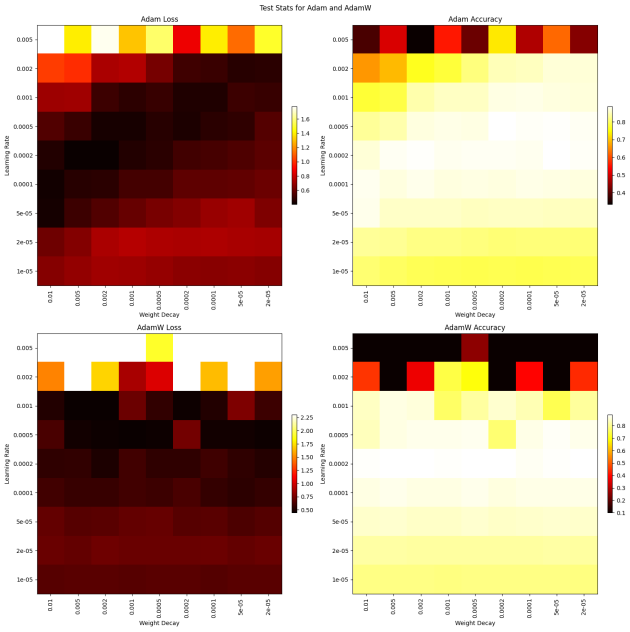


Figure 2. This figure shows the results of the grid test on VGG with Adam(the first row) and AdamW(the second row). The first column shows test loss, and the second column shows test accuracy. With respect to generalization performance, Adam shows clear dependency between learning rate and weight decay rate, which AdamW doesn't have.

and estimated pre-conditioning matrix due to the imperfect implementation of weight decay. With the unintentional noise from L_2 regularization, the overfit is not that severe with Adam as the weight decay rate increases. Therefore, to utilize the advantage from AdamW, it is important to care about overfit, such as with large size data set, or with an appropriate number of epochs.

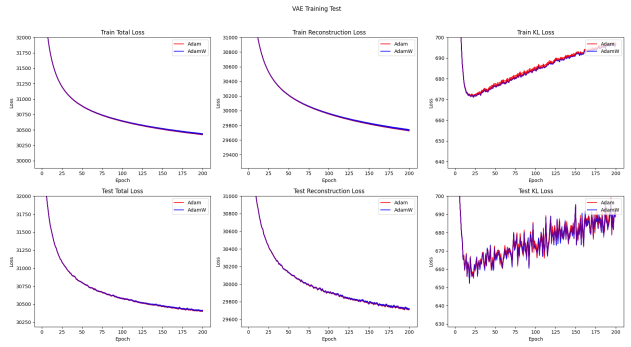
Although the overfit make the trend flipped, Adam still shows the linear correlation between learning rate and weight decay rate. The figure 2 shows the same tendency with the AdamW paper whose x-axis, weight decay rate, is reversed, even without any learning rate scheduler. In the hyperparameter area where Adam shows good generalization, the learning rate depends linearly on the weight decay rate, but AdamW has no dependency between two hyperparameters. Instead of independence on hyperparameters, the generalization performance of AdamW is heavily dependent on the learning rate itself. Different from Adam, AdamW fails to train at the certain range of learning rate. The narrower range of effective learning rates is observed also in the AdamW paper(Figure 1) when there is no learning rate schedule.

3.3.2. VAE AND FASHION MNIST

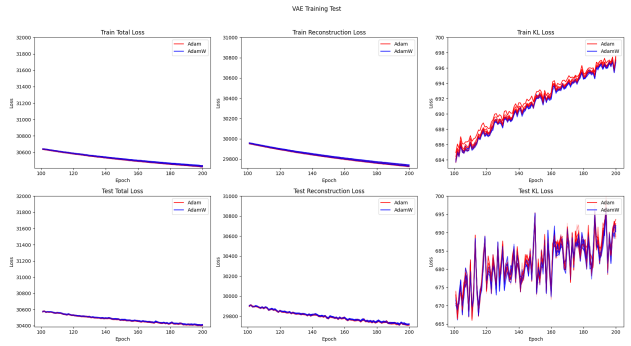
VAE has different side with VGG. VAE is one of the unsupervised learning and its train loss itself contains regularization term, the KL divergence loss. With this regularization term, two optimizers didn't show the overfit for all weight decay rate. Although the KL divergence loss is increased as the number of epochs increases, it doesn't mean overfit, but that the model doesn't ignore its latent space, guaranteeing sample diversity. Although AdamW shows higher train reconstruction loss, which contains possibility for better generalization, it is compensated with less KL divergence, shown in the figure 3.

In addition, all difference, made by weight decay rate change, is not significant. It is interesting that Adam becomes almost insensitive to weight decay rate, given the linear correlation which Adam shows in the VGG case. Following two tendencies, both optimizers give the similar test loss performance and sampled images has no difference qualitatively, for all weight decay rate. Qualitative results are shown in the figure 4. The possible reason is that the weight decay rate, adopted for generalization, has less meaning than the VGG case, because VAE already learns the diversity from KL loss.

Insensitivity on weight decay rate is also supported by the figure 5 which is the result of the grid test. The weight decay rate change makes some perturbation on the same learning rate, but it doesn't build any main stream. For KL loss, Adam shows more fluctuation for the weight decay



(a)



(b)

Figure 3. (a) shows the whole training process. The first row shows the train total loss (reconstruction loss + KL divergence loss), reconstruction loss and KL divergence loss, and the second row shows the same statistics with the test dataset. (b) shows the training process at the last 100 epochs to amplify the small difference between two optimizers. The weight decay rate doesn't have a large effect on the results. A low weight decay rate is visualized with high transparency.

rate than AdamW, but the learning rate change leads the trend on performance change. The sampled images in the figure 6 also show no visible difference for all weight decay rate. Only variation in the learning rate makes distinguishable changes in each image. AdamW shows more changes such as brightness. This pattern is also shown in the figure 5 which indicates more reconstruction loss variation in AdamW.

4. Conclusion & Discussion

This report experiments the effect of weight decay decoupling with simplified but diverse environment which the AdamW paper doesn't try. The results obtained are slightly apart from what the AdamW paper presented. AdamW is weak, when the overfit already occurs, showing fast performance deterioration. This result supports that Adam has dependency between learning rate and weight decay rate, but also shows that performance of AdamW is sensitive to

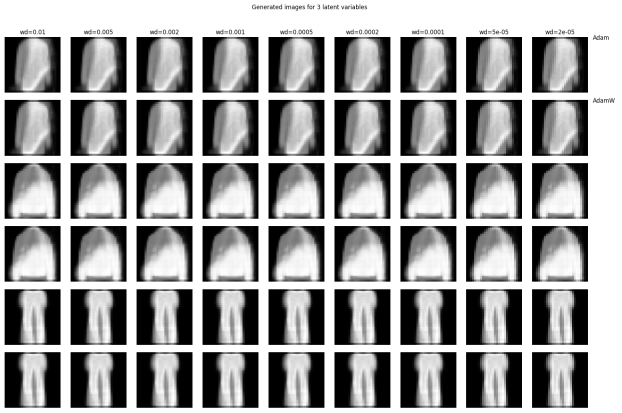


Figure 4. 3 sampled images with Adam (the top row) and AdamW (the bottom row). Each column represents different weight decay rate. Similar to the statistical results, the weight decay rate doesn't have a large effect on the results. A low weight decay rate is positioned at the left.

learning rate change. In the training with regularization for diversity, dependency on weight decay rate, which Adam innately has, gets insignificant.

Although it is hard for one optimizer to be the best for all deep learning field, these results make a question about the performance of AdamW on other training method, such as adversarial training, semi-supervised learning, and reinforcement learning, especially with an explicit regularization(Kumar et al., 2021).

There are some limitations for this experiment. These results are from only two experiments with the small number of seeds. To support this observation confidently, more trials with similar experimental settings are necessary. In addition, the train loss of VAE model is mathematically driven with variational lower bound, different from other regularizers. Therefore, other kinds of regularizers for generalization, not relevant to direct weight decaying, should be conducted to testify that it is possible for Adam to get less correlation between learning rate and weight decay with other generalization support.

References

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Gastaldi, X. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.

Hanson, S. and Pratt, L. Comparing biases for minimal

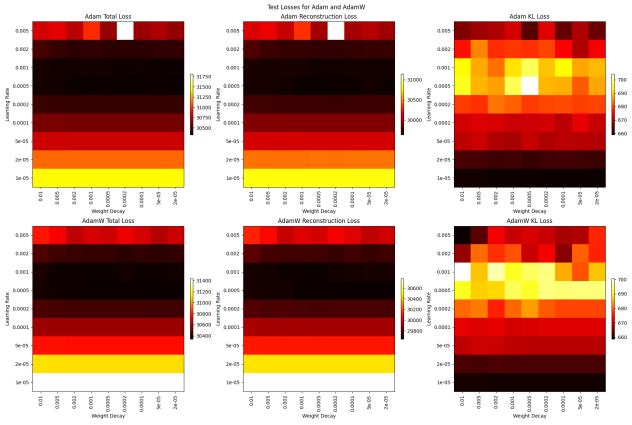


Figure 5. The loss function tendency on learning rate and weight decay rate change with Adam (the first row) and AdamW (the second row). From the right, each grid shows total loss, reconstruction loss, and KL loss. AdamW shows slightly more invariant performance than Adam with respect to the weight decay rate change. However, Adam shows less total and reconstruction loss change on learning rate change than AdamW.

network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

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

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

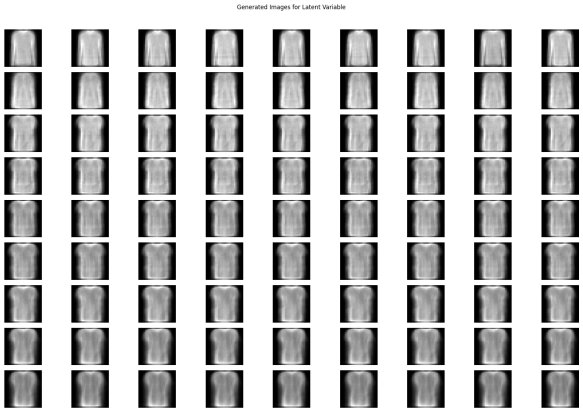
Kumar, A., Agarwal, R., Ma, T., Courville, A., Tucker, G., and Levine, S. Dr3: Value-based deep reinforcement learning requires explicit regularization. *arXiv preprint arXiv:2112.04716*, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

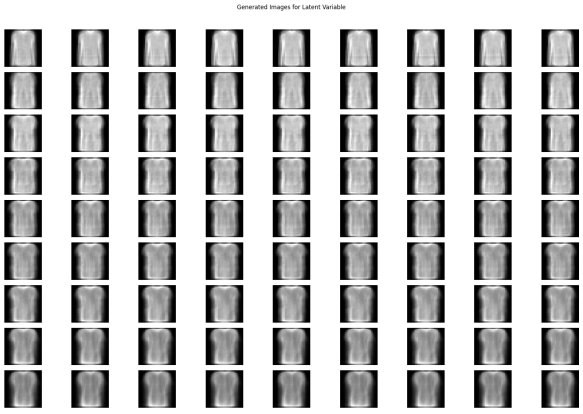
Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tieleman, T. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26, 2012.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.



(a)



(b)

Figure 6. (a) shows sampled images from models trained with Adam which adopts different learning rate and weight decay rate, and (b) is results from AdamW. All images are generated from one same latent variable. As in the training test, all generated samples shows similar qualitative performance, but the learning rate changes fine details in each image, different from weight decay rate.