METHODS WITH PRECONDITIONING WITH WEIGHT DECAY REGULARIZATION

A PREPRINT

Kreinin M.
Department of Data Science
Moscow Institute of Physics and Technology
Moscow, Russia
kreinin.mv@phystech.edu

Beznosikov A.
Department of Data Science
Moscow Institute of Physics and Technology
Moscow, Russia
beznosikov.an@phystech.edu

May 2, 2023

ABSTRACT

This paper investigates the convergence behavior of optimization methods with preconditioning that utilize weight decay regularization, specifically focusing on popular variants such as AdamW and OASIS. We explore different alternatives to these method, with the goal of investigating their convergence speed and accuracy. Also we conduct experiments on benchmark datasets and models in order to compare them on practice. Overall, our study provides insights into the design of regularization techniques methods with preconditioning.

Keywords Adam · AdamW · OASIS · Regularization · Weight Decay · Optimization

1 Introduction

In machine learning we consider unconstrained optimization problem

$$\min_{w \in \mathbb{R}^d} f(w). \tag{1}$$

Problems of the form (1) cover a plethora of applications, including empirical risk minimization [Chapelle et al., 2000], deep learning [LeCun et al., 2015], and supervised learning [Cunningham et al., 2008] tasks such as regularized least squares [Rifkin and Lippert, 2007] or logistic regression [Shalev-Shwartz and Ben-David, 2014].

The classic base method for solving the optimization problem is gradient descent, but this minimization problem can be difficult to solve, particularly when the number of training samples, or problem dimension, is large. In such cases, evaluating the full gradient on every iteration in the context of gradient descent becomes prohibitively expensive, especially considering that gradient descent often requires numerous iterations to converge. In modern machine learning especially large problems represent the greatest interest. For such cases stochastic gradient descent [Robbins and Monro, 1951] became popular solution. Despite its simplicity, it proved itself to be an efficient and effective optimization method. For a long time first-ordered methods were most popular approach of solving optimization problems.

Other way of solving the problem are methods with adaptive gradient [Hazan et al., 2007]. These methods posses several superiority over first-ordered methods. Firstly, they have bigger potential of distributed solving, because first ordered methods spend majority of time on ”communication”. Secondly, they are less sensitive to the choice of hyperparameters up to the point that hyperparameters can be set equal to one. Lastly, this methods often simply show faster convergence on modern large optimization problems, especially this methods became applicable in neural networks solving [Kingma and Ba, 2014]. Nowadays it is known that preconditioning methods often outperform other methods on modern large optimization problems [Goldberg et al., 2011, Kingma and Ba, 2014, Zhang et al., 2018, Yao et al., 2021].

Preconditioning methods refer to techniques that involve scaling the gradient of a problem by a specific matrix D_t, which enables the gradient to take into account the geometry of the problem. In the traditional case $D_t = (\nabla^2 f(w))^{-1}$, which corresponds Newton's method, however hessian is difficult to calculate and even more difficult to reverse, because of that some heuristics are used to replace the reversed hessian [Dennis and Moré, 1977]. In OASIS [Jahani et al., 2021] or AdaHessian [Yao et al., 2021] hessian is assumed to have diagonal dominance. In Adam [Kingma and Ba, 2014] gradient is simply normalized. This heuristics were proved to be effective. Generally, the step of preconditioned algorithms can be expressed as

$$w_t = w_{t-1} - \eta \cdot D_t^{-1} g_t, \tag{2}$$

where η is a learning rate, D_t is a preconditioning matrix, obtained through different heuristics, and g_t is an unbiased stochastic gradient, i.e. $\mathbb{E}[g_t] = \nabla f(w)$.

We do not prescribe the exact method for obtaining g_t. In large-scale problems, stochastic gradient descent is a more efficient approach since computing the full gradient is prohibitively expensive. While computing the Hessian is considerably more costly, so there exist various techniques for computing the preconditioning matrix D_t, and we delegate the specific choice of the preconditioning approach to the authors of individual methods.

Despite the various advantages offered by preconditioning methods, they are prone to overfitting. As a result, preconditioning methods in practice are almost always utilized with various methods to combat overfitting, with regularization being the primary such method employed. Regularization is a powerful technique in machine learning that aims to prevent overfitting by adding additional constraints to the model. It has been widely applied to various machine learning problems, including image classification [Zhu et al., 2017], speech recognition [Zhou et al., 2017], and natural language processing [Wu et al., 2022], and has shown its effectiveness in improving the generalization capability of neural networks [Girosi et al., 1995].

In methods with preconditioning appears to be several ways to include regularization. We can include regularizer r in g_t calculation so it will be taken into consideration while calculating D_t. This method is equal to considering optimization problem

$$\min_{w \in \mathbb{R}^d} f(w) + r(w). \tag{3}$$

Or we can include regularizer only on last step, decreasing norm of w [Loshchilov and Hutter, 2017]. This way of regularization is called weight decay and surprisingly turns out to be more efficient in practical problems. There are few other ways of considering regularizer which will be discussed further in the paper.

2 Notation

In preconditioned methods, there exist several techniques for incorporating regularization into the optimization process. In this study, we consider three different approaches, which are illustrated in Algorithm 1 using different colors. In general, these methods can be characterized by the stage in which the regularization term is incorporated into the optimization process.

Algorithm 1 Different ways of regularization

Require: η, f
  while w not converged **do**
    $t = t + 1$
    $g_t \leftarrow$ stochastic gradient
    $g_t \leftarrow g_t + \nabla r(w_t)$ standart regularization
    $D_t \leftarrow$ preconditioning matrix
    $w_t \leftarrow w_{t-1} - \eta \cdot D_t^{-1}\left(g_t + \nabla r(w_t)\right) - \eta \cdot \nabla r(w_t)$ hessian weight decay, weight decay
  end while

To be more specific, the first regularization technique illustrated in blue involves simply adding the regularization term to the objective function. This regularizer is included in the pseudo-gradient and factored into the calculation of D_t, and can be viewed as solving the following optimization problem:

$$\min_{w \in \mathbb{R}^d} F(w) = f(w) + r(w),$$

where $f(w)$ is the objective function and $r(w)$ is the regularization term. In essence, this approach involves applying the basic preconditioning method to the function F.

The second regularization technique, shown in orange, is a novel approach. Here, the regularization term is added before applying D_t, without affecting its computation.

The last regularization approach we consider is known as weight decay, illustrated by the color red in the general scheme. This method only incorporates the regularizer during the algorithmic step, avoiding interference of regularization with the preconditioning stage.

Overall, it is important to carefully consider the impact of regularization when designing optimization algorithms, and we hope that our investigation of this techniques will prove useful to researchers in the field.

In our paper we mainly focus on weight decay regularization. Though this technique is simpler than the others we consider, it can be quite effective in many cases.

3 Weight decay regularization

The most common regularization technique is ℓ_2 regularization, defined as $r(w_t) = \frac{\lambda}{2}||w_t||^2$, which yields $\nabla r(w_t) = \lambda \cdot w_t$. In this case, the weight decay regularization achieves its name. It only serves to reduce the norm of the weight vector, through the subtraction of $\eta \cdot w_t$ in the final step.

However in our theoretical evaluations we use regularizer in general form. Thus, algorithmic step that we consider is

$$w_{t+1} = w_t - \eta \cdot D_t^{-1} \nabla f(w_t) - \eta \cdot \nabla r(w_t), \tag{4}$$

where D_t is a preconditioning matrix, η is a learning rate, f is a target function and r is a regularizer.

3.1 Convergence speed of preconditioning methods

We set ourselves a goal to estimate a convergence speed of methods with preconditioning with weight decay regularization. Although step of methods with weight decay seems simple, it can be viewed in a rather unexpected way. We can put D_t^{-1} out of brackets which gives

$$w_{t+1} = w_t - \eta \cdot D_t^{-1}(\nabla f(w_t) + D_t \nabla r(w_t)). \tag{5}$$

That suggests the need to introduce a variable $\tilde{r}$ such that $\nabla \tilde{r}(w_t) = D_t \nabla r(w_t)$ and new target function $\tilde{F} = f + \tilde{r}$. New regularizer $\tilde{r}$ becomes adaptive, because both r and D_t depends on weight vector w_t.

The convergence speed is typically measured in terms of the number of iterations required to reach a certain level of error. To obtain estimates on the number of iterations required to converge to a given error, we must impose certain assumptions on the function.

Throughout this work we assume that each f : $\mathbb{R}^d \to \mathbb{R}$ is convex, twice differentiable and L-smooth. Additionally we imply a PL-condition to make another evaluation concerning speed of convergence. This is formalized in the following assumptions.

Assumption 1 (Convexity). *The function f is convex, i.e.* $\forall y, x \in \mathbb{R}^d$

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

Assumption 2 (PL–condition). *There exists* $\mu > 0$*, such that* $||\nabla f(w)|| \geq 2\mu(f(w) - f^*)$*,* $\forall w \in \mathbb{R}^d$

Assumption 3 (L-l-smoothness). *The gradients of F are L-Lipschitz continuous* $\forall w \in \mathbb{R}^d$*, i.e. there exists a constant* $L > 0$ *such that* $\forall x, y \in \mathbb{R}^d$*,*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}||x - y||^2$$

The gradient of r is l-Lipschitz continuous $\forall w \in \mathbb{R}^d$*, i.e. there exists a constant* $l > 0$ *such that* $\forall x, y \in \mathbb{R}^d$*,*

$$r(x) \leq r(y) + \langle \nabla r(y), x - y \rangle + \frac{l}{2}||x - y||^2$$

Also we must introduce restrictions on preconditioner D_t

$$\alpha I \preccurlyeq D_t \preccurlyeq \Gamma I \Leftrightarrow \frac{I}{\alpha} \preccurlyeq D_t^{-1} \preccurlyeq \frac{I}{\Gamma} \tag{6}$$

Using introduced assumptions we prooved convergence of methods with preconditioning with weight decay regularization in general form. Our results are framed in Theorem 1 and Theorem 2

Theorem 1. *Suppose the Assumption 1, 2 hold, let $\varepsilon > 0$ and let the step-size satisfy*

$$\eta < \frac{2\alpha}{L + l \cdot \alpha}$$

Then, the number of iterations performed by AdamW algorithm, starting from an initial point $w_0 \in \mathbb{R}^d$ with $\Delta_0 = \tilde{F}(w_0) - \tilde{F}^*$, required to obtain and ε-approximate solution of the convex problem (1) *can be bounded by*

$$T = \mathcal{O}\left(\frac{2\Delta_0 \Gamma \alpha}{(2\alpha - \tilde{L}\eta)\eta\varepsilon}\right)$$

Proof. Let's write 1-Convexity for step t and $t + 1$:

$$f(w_{t+1}) \leq f(w_t) + \langle \nabla f(w_t), w_{t+1} - w_t \rangle + \frac{L}{2} ||w_{t+1} - w_t||^2,$$

Okay, by definition for our algorithm we have:

$$w_{t+1} - w_t = -\eta D_t^{-1} \nabla f(w_t) - \eta \nabla r(w_t),$$

from the previous expression , we express the gradient of the function

$$\nabla f(w_t) = \frac{1}{\eta} D^t (w_t - w_{t+1}) - D^t \nabla r(w_t),$$

replace $\nabla f(w_t)$ and by definition of matrix D_t, $I \leq \frac{D_t}{\alpha}$

$$f(w_{t+1}) \leq f(w_t) + \langle \frac{1}{\eta} D_t (w_t - w_{t+1}) - D_t \nabla r(w_t), w_{t+1} - w_t \rangle + \frac{L}{2\alpha} ||w_{t+1} - w_t||^2_{D_t},$$

now let's bring it together

$$f(w_{t+1}) \leq f(w_t) + \left(\frac{L}{2\alpha} - \frac{1}{\eta}\right) ||w_{t+1} - w_t||^2_{D_t} - \langle D_t \nabla r(w_t), w_{t+1} - w_t \rangle,$$

define new regularization function $\tilde{r} : \nabla \tilde{r} = D_t \nabla r(w_t)$.

then rewrite step using the variable and assumption 3-L-l-smoothness

$$\tilde{r}(w_{t+1}) \leq \tilde{r}(w_t) + \langle \tilde{r}(w_t), w_{t+1} - w_t \rangle + \frac{l}{2} (w_{t+1} - w_t)^T D_t (w_{t+1} - w_t),$$

let's replace the old regularization function with a new one

$$f(w_{t+1}) \leq f(w_t) + \left(\frac{L}{2\alpha} - \frac{1}{\eta}\right) ||w_{t+1} - w_t||^2_{D_t} + \tilde{r}(w_t) - \tilde{r}(w_{t+1}) + \frac{l}{2} ||w_{t+1} - w_t||^2_{D_t},$$

now let's define a new loss function $\tilde{F}(w) = f(w) + \tilde{r}(w)$, $F(w) = f(w) + r(w)$, ($\tilde{L} = L + l\alpha$), we get:

let's rewrite our inequality in new notation

$$\tilde{F}(w_{t+1}) \leq \tilde{F}(w_t) + \left(\frac{\tilde{L}}{2\alpha} - \frac{1}{\eta}\right) ||w_{t+1} - w_t||^2_{D_t},$$

now we select the step in such a way that $\frac{\tilde{L}}{2\alpha} - \frac{1}{\eta} < 0$, $\eta < \frac{2\alpha}{\tilde{L}}$

$$\left(\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha}\right) ||w_{t+1} - w_t||^2_{D_t} \leq \tilde{F}(w_t) - \tilde{F}(w_{t+1}),$$

let's sum up our inequalities and evaluate the left part from below

$$\frac{\eta^2(T+1)}{\Gamma}\left(\frac{1}{\eta}-\frac{\tilde{L}}{2\alpha}\right)\cdot\min_{k=0,T}||\nabla f(w_t)+\nabla\tilde{r}(w_t)||^2\leq\frac{\eta^2}{\Gamma}\left(\frac{1}{\eta}-\frac{\tilde{L}}{2\alpha}\right)\cdot\sum_{t=0}^{T}||\nabla f(w_t)+\nabla\tilde{r}(w_t)||^2\leq\tilde{F}(w_0)-\tilde{F}(w_*),$$

moving everything to the right we get the following estimate

$$\min_{t=0,T}||\nabla f(w_t)+\nabla\tilde{r}(w_t)||^2\leq\frac{(\tilde{F}(w_0)-\tilde{F}(w_*))\Gamma}{(\frac{1}{\eta}-\frac{\tilde{L}}{2\alpha})\eta^2(T+1)}=\varepsilon$$

$$T+1\geq\frac{\Delta_0\Gamma}{(\frac{1}{\eta}-\frac{\tilde{L}}{2\alpha})\eta^2\varepsilon}$$

we get an estimate for the number of steps required for a given accuracy

$$T=\mathcal{O}\left(\frac{2\Delta_0\Gamma\alpha}{(2\alpha-\tilde{L}\eta)\eta\varepsilon}\right)$$

□

Theorem 2. *Suppose the Assumption 1, 2, 3 hold, let $\varepsilon>0$ and let the step-size satisfy*

$$\eta\leq\frac{2\alpha}{\tilde{L}}$$

Then, the number of iterations performed by AdamW algorithm, starting from an initial point $w_0\in\mathbb{R}^d$ with $\Delta_0=\tilde{F}(w_0)-\tilde{F}^*$, required to obtain and ε-approximate solution of the convex problem (1) *can be bounded by*

$$T=\mathcal{O}\left(\frac{\ln\frac{\Delta_0}{\epsilon}}{2\mu\eta^2(\frac{1}{\eta}-\frac{\tilde{L}}{2\alpha})}\right)$$

Proof. The proof of this theorem will be similar to the previous one, the main difference is that we impose another assumoption 2-PL–condition on the original function Assume

$$\nabla\tilde{F}=\nabla f+\nabla\tilde{r}$$

$$L+||D_t||l=\tilde{L}$$

rewrite step in terms of new function

$$w_{t+1}-w_t=-\eta D_t^{-1}\nabla r(w_t)-\eta\nabla r(w_t)=-\eta D_t^{-1}(\nabla f+\nabla\tilde{r})(w_t)=-\eta D_t^{-1}\nabla\tilde{F}(w_t),$$

Then we write $\tilde{L}$-smoothness for $\tilde{F}$

$$\tilde{F}(w_{t+1})-\tilde{F}(w_t)\leq\langle\nabla\tilde{F}(w_t),w_{t+1}-w_t\rangle+\frac{\tilde{L}}{2}||w_{t+1}-w_t||^2,$$

then combine it together and use constraints on the matrix $\alpha\cdot I\preccurlyeq D_t\preccurlyeq\Gamma\cdot I$

$$\tilde{F}(w_{t+1})-\tilde{F}(w_t)\leq-\langle\frac{1}{\eta}D_t(w_{t+1}-w_t),w_{t+1}-w_t\rangle+\frac{\tilde{L}}{2}||w_{t+1}-w_t||^2=(\frac{\tilde{L}}{2\alpha}-\frac{1}{\eta})||w_{t+1}-w_t||^2_{D_t}=$$

$$=\quad(\frac{\tilde{L}}{2\alpha}-\frac{1}{\eta})||w_{t+1}-w_t||^2_{D_t}=(\frac{\tilde{L}}{2\alpha}-\frac{1}{\eta})||-\eta D_t^{-1}\nabla\tilde{F}(w_t)||^2_{D_t}\leq(\frac{\tilde{L}}{2\alpha}-\frac{1}{\eta})\eta^2||\nabla\tilde{F}(w_t)||^2_{D_t^{-1}},$$

Then we use PL-condition 2-PL–condition for the function $\tilde{F}$:

$$||\nabla\tilde{F}(w_t)||^2_{D_t^{-1}}\geq2\mu(\tilde{F}(w_t)-\tilde{F}^*),$$

subtract the exact solution from both parts and apply PL-condition

$$\tilde{F}(w_t) - F^* \geq \tilde{F}(w_{t+1}) - \tilde{F}^* + (\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha})\eta^2 2\mu(\tilde{F}(w_t) - \tilde{F}^*) = \left(1 + 2\mu\eta^2(\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha})\right)(\tilde{F}(w_{t+1}) - \tilde{F}^*),$$

we apply the expression for each step, $\Delta_0 = \tilde{F}(w_0) - \tilde{F}(w_*)$

$$\varepsilon \geq \Delta_0 \left(1 + 2\mu\eta^2(\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha})\right)^{-T} \geq (\tilde{F}(w_T) - \tilde{F}^*),$$

we get the necessary number of steps to get together with the error ε

$$T = \frac{\ln \frac{\Delta_0}{\varepsilon}}{\ln(1 + 2\mu\eta^2(\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha}))} \approx \frac{\ln \frac{\Delta_0}{\varepsilon}}{2\mu\eta^2(\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha})}$$

and finally we get:

$$T = \mathcal{O}\left(\frac{\ln \frac{\Delta_0}{\varepsilon}}{2\mu\eta^2(\frac{1}{\eta} - \frac{\tilde{L}}{2\alpha})}\right)$$

□

3.2 Convergence trajectory of preconditioning methods

In the previous subsection we have proved convergence of preconditioned methods, however we have pointed out above that methods with weight decay does not converge to the initial optimized function $F = f + r$, but rather to a new function $\tilde{F} = f + \tilde{r}$.

We will consider two algorithms OASIS and Adam, and its variations. Their main difference is in the calculation of the pseudo hessian. In Adam, the Hessian is a diagonal matrix consisting of squares of derivatives, in OASIS we have a stochastic Hessian, which is calculated through a random variable from the Randemacher distribution.

We framed three methods of regularization for Adam and OASIS in Algorithm 2 and Algorithm 3 respectively. Results of our computational experiments are framed in Figure 1 and Figure 2. In can be seen that methods with weight decay converges only by special criterion $\nabla f + D_t \nabla r$.

Algorithm 2 OASIS

Require: $w_0, \eta_0, D_0, \theta_0 = +\infty$

$w_1 = w_0 - \eta \hat{D_0}^{-1} \nabla f(w_0)$

for $k = 1, 2, ...$ **do**

$g_k = \nabla f(w_k) + \nabla r(w_{t-1})$ OASISL2

$D_k = \beta D_{k-1} + (1 - \beta_2) \cdot diag\left(z_k \odot \nabla^2 \left(f(w_k) + r(w_k)\right) z_k\right)$ OASISWH

$(\hat{D_k})_{ii} = max\{|D_k|_{i,i}; \alpha\}, \forall i = \overline{1, d}$

$\eta_k = min\{\sqrt{1 + \theta_{k-1}} \cdot \eta_{k-1}; \frac{||w_k - w_{k-1}||_{\hat{D_k}}}{2||\nabla f(w_k) - \nabla f(w_{k-1})||^*_{\hat{D_k}}}\}$

$w_{k+1} = w_k - \eta_k g_k D_k^{-1} - \eta \nabla r(w_{t-1})$ OASISW

$\theta_k = \frac{\eta_k}{\eta_{k-1}}$

end for

Algorithm 3 Adam

Require: $\eta, \beta_1, \beta_2, \epsilon, f, r$
while θ not converged **do**
$t = t + 1$
$g_t = \nabla f(w_{t-1}) + \nabla r(w_{t-1})$ AdamL2
$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$
$\hat{m}_t = \frac{m_t}{1-\beta_1^t} + \nabla r(w_{t-1})$ AdamWH
$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
$w_t = w_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{v_t}+\epsilon} - \eta \nabla r(w_{t-1})$ AdamW
end while

3.3 Experiments

We set up experiments on the dataset mushrooms, logistic regression was chosen as the model, the optimizers were AdamL2, AdamW, AdamWH, OASISL2, OASISW, OASISWH, SGD. Various learning rates and weight decays were considered.

We observe the effect that by the new criterion the AdamW, AdamWH, OASISW, OASISWH methods converge much better than by the old one, and they also converge better than AdamL2, OASISL2.

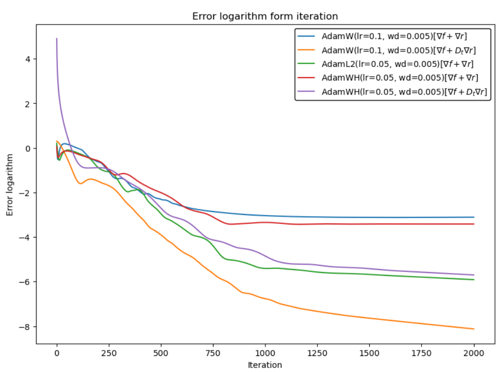

Figure 1: Adam on dataset mushrooms

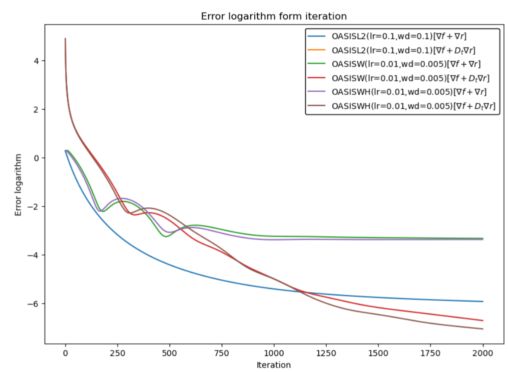

Figure 2: OASIS on dataset mushrooms

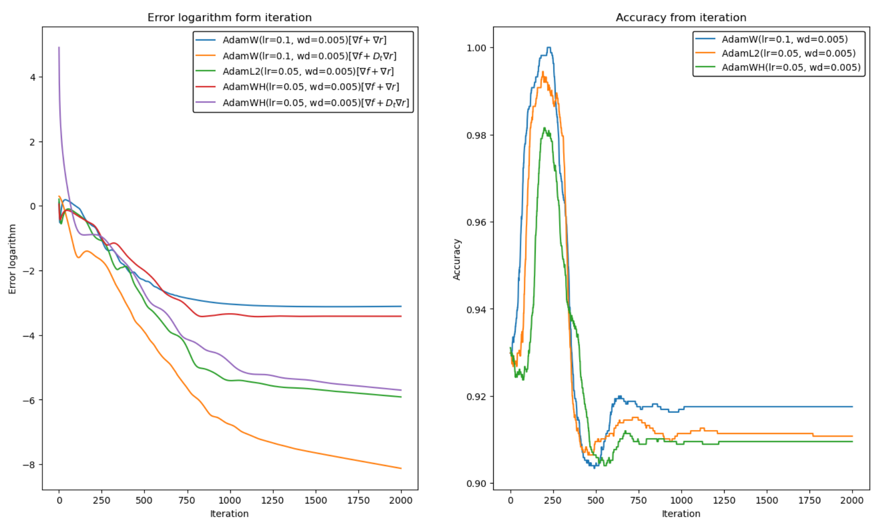

Figure 3: Adam algrorithms on dataset: mushrooms

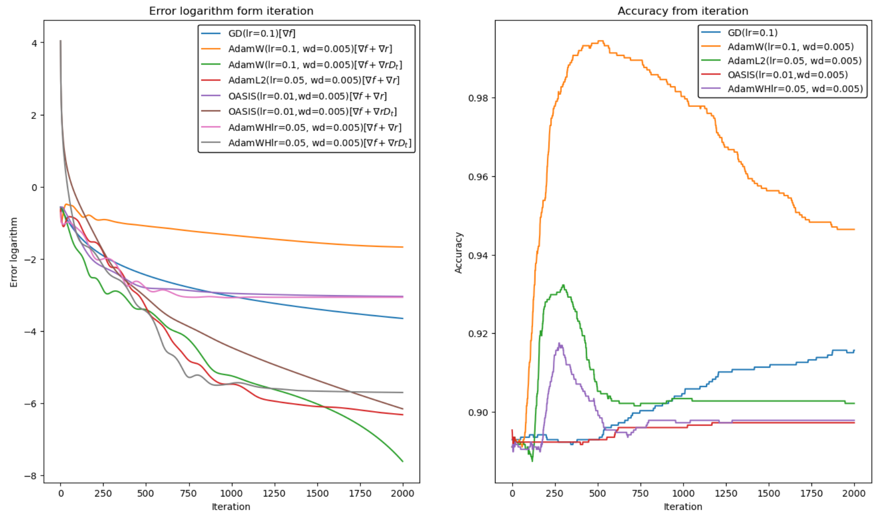

Figure 4: Compare different optimization algorithms on dataset: mushrooms

As we can see, the AdamW method shows the best result compared to other optimization algorithms, it is also clear that the new AdamW criterion converges much faster than the old one, and thus it has a better generalization ability compared to other methods.

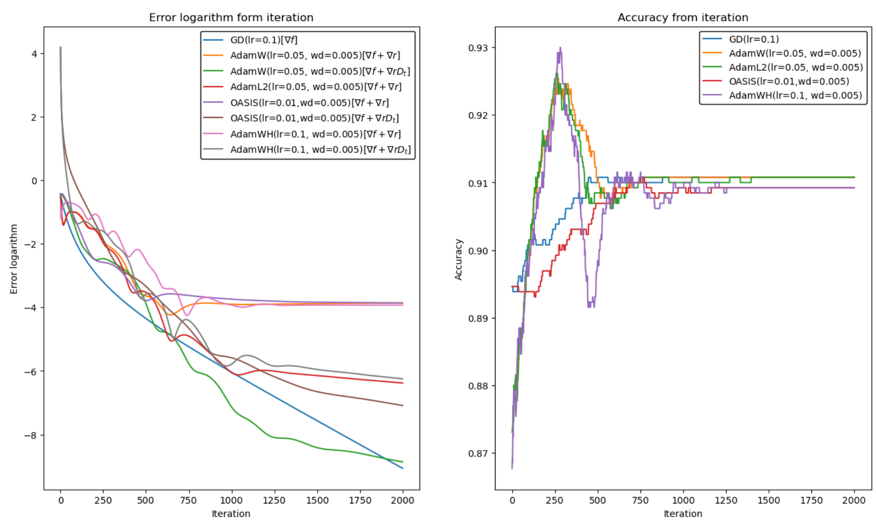

Figure 5: Compare different optimization algorithms on dataset: mushrooms

Here we see a similar picture as in the last example.

3.3.1 Mushrooms

Here are all the experiments that were performed on the mushrooms dataset. Learning rate and weight decay are signed for each method on the picture, as well as the criterion used to calculate the error.

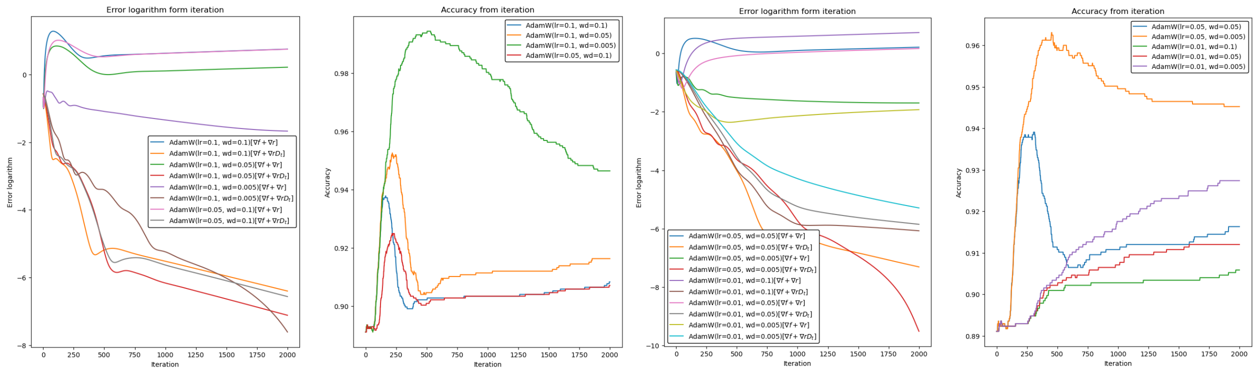

Figure 6: AdamW on dataset mushrooms

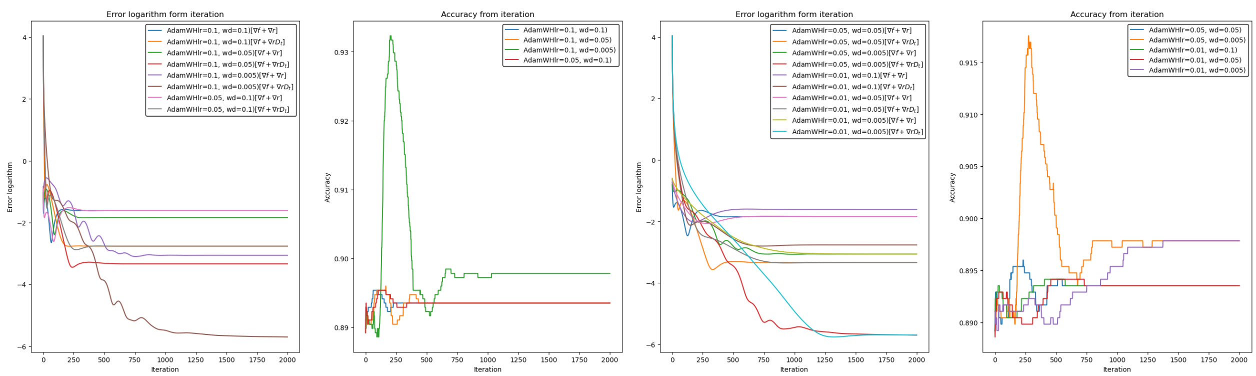

Figure 7: AdamWH on dataset mushrooms

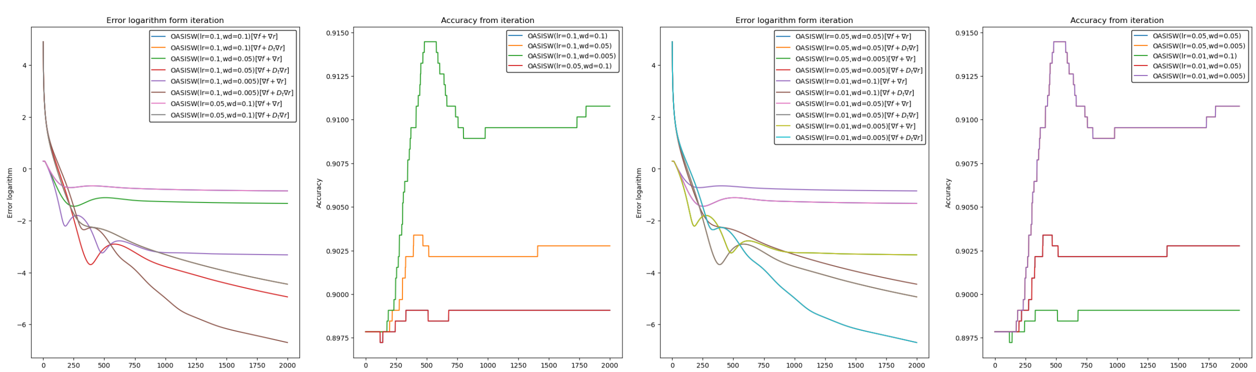

Figure 8: OASISW on dataset mushrooms

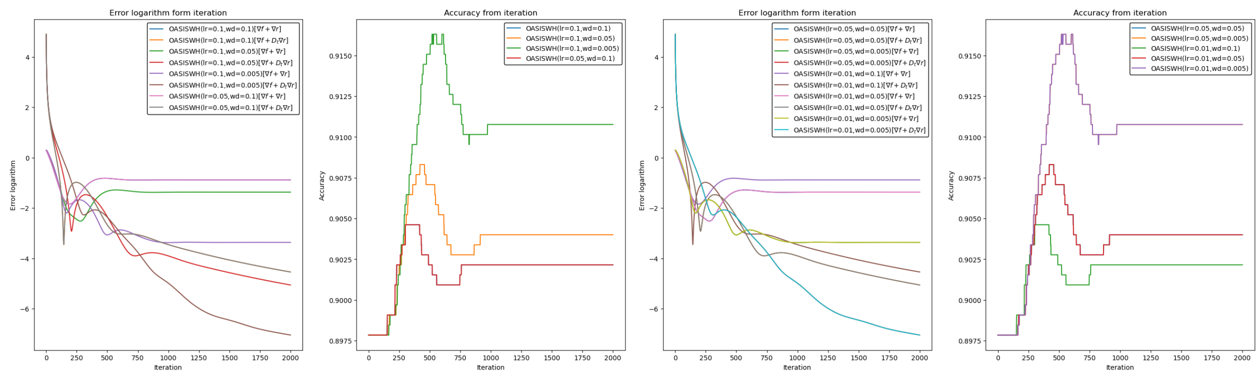

Figure 9: OASISWH on dataset mushrooms

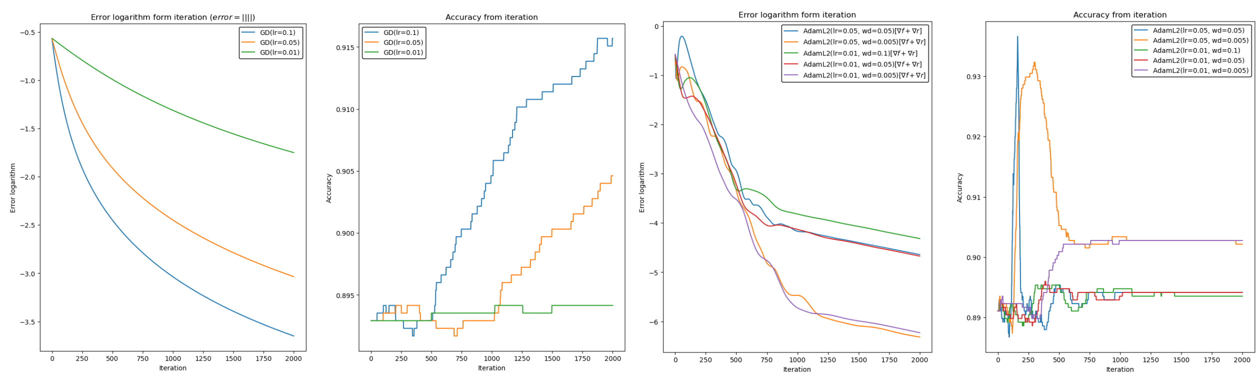

Figure 10: Gradient descent and AdamL2 on dataset mushrooms

3.3.2 Wine

Here are all the experiments that were performed on the wine dataset. Learning rate and weight decay are signed for each method on the picture, as well as the criterion used to calculate the error.

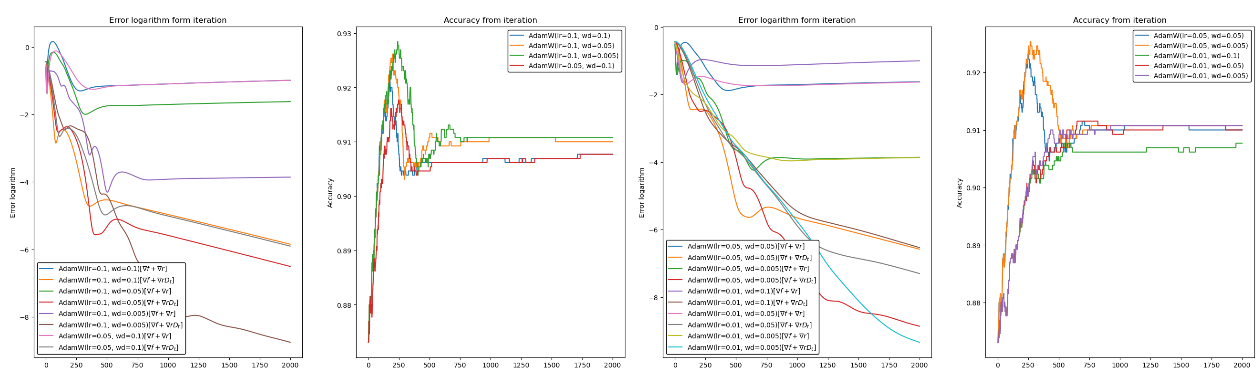

Figure 11: AdamW on dataset wine

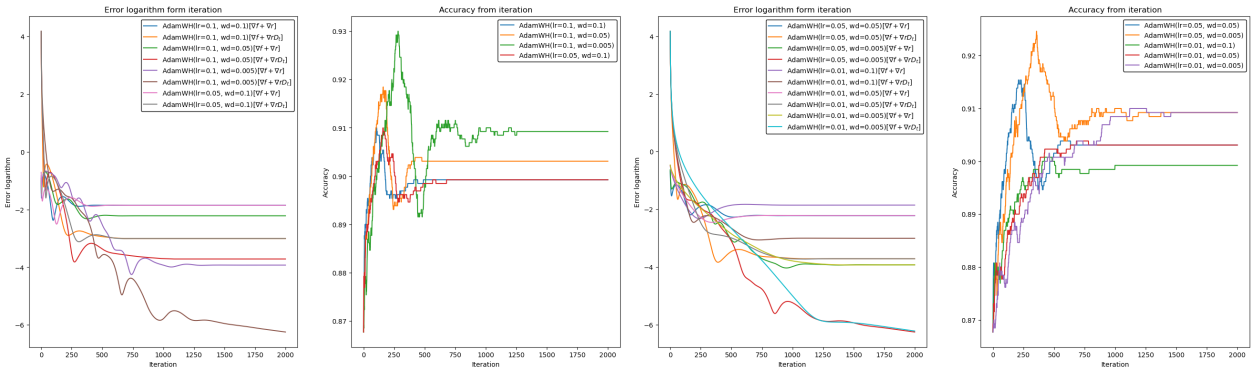

Figure 12: AdamWH on dataset wine

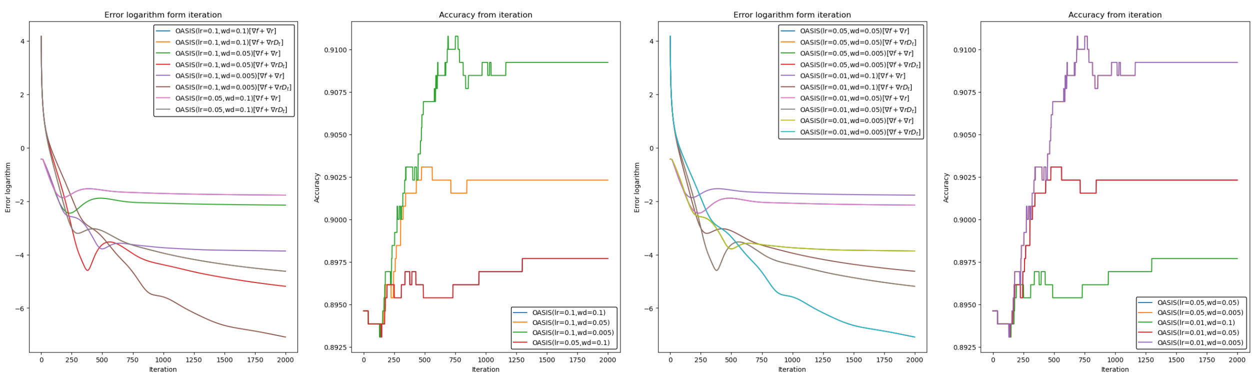

Figure 13: OASISW on dataset wine

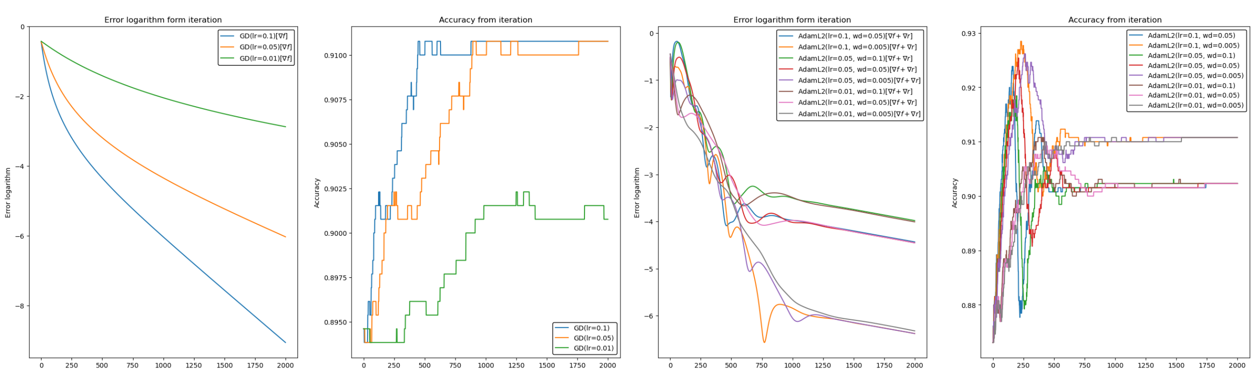

Figure 14: Gradient descent and AdamL2 on dataset wine

And to finalize our results let's show the results of training ResNet18 on 4 different optimizators with CosineAnnealingLr on dataset CIFAR10.

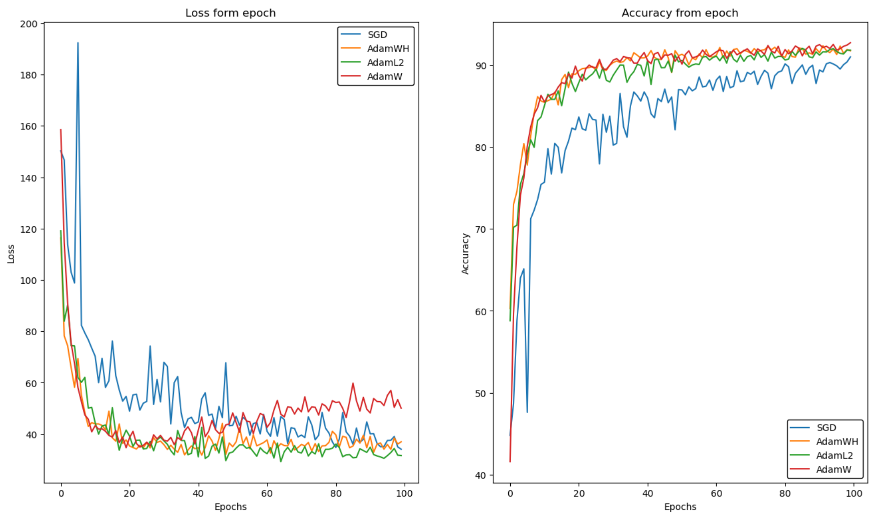

Figure 15: Compare different optimization algorithms on dataset: CIFAR10

AdamW shows the worst results for the loss function, but wins all other methods in accuracy on the test sample, exactly this generalizing ability of the AdamW method was mentioned earlier.

3.4 Conclusion:

This paper investigated preconditioning methods with weight decay regularization. Two theorems were established for these methods, each under different assumptions regarding the target loss function. The theorems demonstrated that these methods converge to a different criterion that is significantly superior to the original one. Consequently, these methods exhibit reduced overfitting and greater generalization ability, leading to excellent performance when training neural networks. Empirical results were also presented to support these findings.

References

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. *Advances in neural information processing systems*, 13, 2000.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49, 2008.

Ryan M Rifkin and Ross A Lippert. Notes on regularized least squares. 2007.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

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

Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. *Advances in Neural Information Processing Systems*, 20, 2007.

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

Andrew Goldberg, Xiaojin Zhu, Alex Furger, and Jun-Ming Xu. Oasis: Online active semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 362–367, 2011.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10665–10673, 2021.

John E Dennis, Jr and Jorge J Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

Majid Jahani, Sergey Rusakov, Zheng Shi, Peter Richtárik, Michael W Mahoney, and Martin Takáč. Doubly adaptive scaled algorithm for machine learning using second-order information. *arXiv preprint arXiv:2109.05198*, 2021.

Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5513–5522, 2017.

Yingbo Zhou, Caiming Xiong, and Richard Socher. Improved regularization techniques for end-to-end speech recognition. *arXiv preprint arXiv:1712.07108*, 2017.

Tingting Wu, Xiao Ding, Minji Tang, Hao Zhang, Bing Qin, and Ting Liu. Stgn: an implicit regularization method for learning with noisy labels in natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7587–7598, 2022.

Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural computation*, 7(2):219–269, 1995.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.