AN EXPONENTIAL LEARNING RATE SCHEDULE FOR BATCH NORMALIZED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Intriguing empirical evidence exists that deep learning can work well with exotic schedules for varying the learning rate. This paper suggests that the phenomenon may be due to Batch Normalization or BN (Ioffe & Szegedy, 2015), which is ubiquitous and provides benefits in optimization and generalization across all standard architectures. The following new results are shown about BN with weight decay and momentum (in other words, the typical use case which was not considered in earlier theoretical analyses of stand-alone BN (Ioffe & Szegedy, 2015; Santurkar et al., 2018; Arora et al., 2018))

• Training can be done using SGD with momentum and an exponentially increasing learning rate schedule, i.e., learning rate increases by some \((1 + \alpha)\) factor in every epoch for some \(\alpha > 0\). (Precise statement in the paper.)
To the best of our knowledge this is the first time such a rate schedule has been successfully used, let alone for highly successful architectures. As expected, such training rapidly blows up network weights, but the net stays well-behaved due to normalization.

• Mathematical explanation of the success of the above rate schedule: a rigorous proof that it is equivalent to the standard setting of BN + SGD + Standard Rate Tuning + Weight Decay + Momentum. This equivalence holds for other normalization layers as well, Group Normalization (Wu & He, 2018), Layer Normalization (Ba et al., 2016), Instance Norm (Ulyanov et al., 2016), etc.

• A worked-out toy example illustrating the above linkage of hyperparameters. Using either weight decay or BN alone reaches global minimum, but convergence fails when both are used.

1 INTRODUCTION

Batch Normalization (BN) offers significant benefits in optimization and generalization across architectures, and has become ubiquitous. Usually best performance is attained by adding weight decay and momentum in addition to BN.

Usually weight decay is thought to improve generalization by controlling the norm of the parameters. However, it is fallacious to try to separately think of optimization and generalization because we are dealing with a nonconvex objective with multiple optima. Even slight changes to the training surely lead to a different trajectory in the loss landscape, potentially ending up at a different solution! One needs trajectory analysis to have a hope of reasoning about the effects of such changes.

In the presence of BN and other normalization schemes, including GroupNorm, LayerNorm, and InstanceNorm, the optimization objective is scale invariant to the parameters, which means rescaling parameters would not change the prediction. The current paper introduces new modes of analysis for such settings. This rigorous analysis yields the surprising conclusion that the original learning rate (LR) schedule and weight decay can be folded into a new exponential schedule for learning rate: in each iteration multiplying it by \((1 + \alpha)\) for some \(\alpha > 0\) that depends upon the momentum and weight decay rate.

\(^1\)Often the parameters that compute the output do not have BN and are thus not scale-invariant. In our experiments we found that fixing the output layer randomly doesn’t harm the performance of the network. So the trainable parameters satisfy scale-invariance
Theorem 1.1 (Main, Informal). SGD on a scale-invariant objective with initial learning rate $\eta$, weight decay factor $\lambda$, and momentum $\gamma$ is equivalent to SGD where at iteration $t$, the learning rate $\tilde{\eta}_t$ in the new exponential learning rate schedule is defined as $\tilde{\eta}_t = \alpha^{2t-1} \eta$, where $\alpha$ is a non-zero root of equation

$$x^2 - (1 + \gamma - \lambda \eta) x + \gamma = 0.$$ 

Specifically, when momentum $\gamma = 0$, the above schedule can be simplified as $\tilde{\eta}_t = (1 - \lambda \eta)^{2t-1} \eta$ and $\tilde{\gamma} = 0$.

The above theorem requires that the product of learning rate and weight decay factor, $\lambda \eta$, is small compared to $1 - \gamma$, which is almost always satisfied in practice. The rigorous and most general version of above theorem is Theorem 2.5, which deals with adaptive learning rate schedule, momentum and weight decay.

Such an exponential increase in learning rate seems absurd at first sight and to the best of our knowledge, no deep learning success has been reported using such an idea before. It does highlight the above-mentioned viewpoint that in deep learning, optimization and regularization are not easily separated. Of course, the exponent trumps the effect of initial lr very fast, which serves as another explanation of the standard wisdom that initial lr is unimportant when training with BN.

Note that it is customary in BN to switch to a lower learning rate upon reaching a plateau in the validation loss. According to the analysis in the above theorem, this corresponds to an exponential growth with a smaller exponent, except for a transient effect when a correction term is needed for the two processes to be equivalent (see discussion around Theorem 2.4).

Thus the final training algorithm is roughly as follows: Start from a convenient learning rate like 0.1, and grow it at an exponential rate with a suitable exponent. When validation loss plateaus, switch to an exponential growth of lr with a lower exponent. Repeat the procedure until the training loss saturates.

In Section 3, we demonstrate on a toy example how weight decay and normalization are inseparably involved in the optimization process. With either weight decay or normalization alone, SGD is guaranteed to achieve zero training error. But with both turned on, SGD fails to converge to global minimum.

In Section 4 of experiments, we verify our theoretical findings on CNNs and ResNets. We also construct better Exponential learning rate schedules by incorporating the Cosine learning rate schedule, which opens the possibility of even more general theory of rate schedule tuning towards better performance.

1.1 RELATED WORK

There have been other theoretical analyses of training models with scale-invariance. (Cho & Lee, 2017) proposed to run Riemmanian gradient descent on Grassmann manifold $G(1, n)$ since the weight matrix is scaling invariant to the loss function. (Hoffer et al., 2018) observed that the effective stepsize is proportional to $\eta \lVert w_t \rVert^2$. (Arora et al., 2019) show the gradient is always perpendicular to the current parameter vector which has the effect that norm of each scale invariant parameter group increases monotonically, which has an auto-tuning effect. (Wu et al., 2018) proposes a new adaptive learning rate schedule motivated by scale-invariance property of Weight Normalization.

Previous work for understanding Batch Normalization. (Santurkar et al., 2018) suggested that the success of BN has does not derive from reduction in Internal COvariate Shift, but by making landscape smoother. (Kohler et al., 2018) shows 2-layer linear nets with BN could achieve exponential convergence rate, but their analysis is for a variant of GD with an inner optimization loop rather than GD itself. (Bjorck et al., 2018) observe that the higher learning rates enabled by BN empirically improves generalization. (Arora et al., 2019) prove that with certain mild assumption, (S)GD with BN finds approximate first order stationary point with any fixed learning rate. None of the above analyses incorporated weight decay, but (Zhang et al., 2019) argued qualitatively that weight decay makes parameters have smaller norms, and thus larger effective learning rate. None of the above analyses deals with momentum.
1.2 Preliminaries and Notations

For batch $B = \{x_i\}_{i=1}^{B}$, network parameter $\theta$, we use $f_{\theta}$ to denote the network and use $L_t(f_{\theta}) = L(f_{\theta}, B_t)$ to denote the loss function at iteration $t$. When there’s no ambiguity, we also use $L_t(\theta)$ for convenience.

We say a loss function $L(\theta)$ is scale invariant to its parameter $\theta$ is for any $c \in \mathbb{R}^+$, $L(\theta) = L(c \theta)$, in practice, the source of scale invariance is usually different types of normalization layers, including Batch Normalization [Ioffe & Szegedy, 2015], Group Normalization [Wu & He, 2018], Layer Normalization [Ba et al., 2016], Instance Norm [Ulyanov et al., 2016], etc.

Implementations of SGD with Momentum/Nesterov comes with subtle variations in literature. We adopt the variant from [Sutskever et al., 2013], also the default in PyTorch [Paszke et al., 2017]. $L2$ regularization (a.k.a. Weight Decay) is another common trick used in deep learning. Combining them together, we get the one of the mostly used optimization algorithms below.

**Definition 1.2.** [SGD with Momentum and Weight Decay][cite] At iteration $t$, with randomly sampled batch $B_t$, update the parameters $\theta_t$ and momentum $v_t$ as following:

$$\begin{align*}
\theta_t &= \theta_{t-1} - \eta_t v_t \\
v_t &= \gamma v_{t-1} + \nabla_{\theta} \left( L_t(\theta_{t-1}) + \frac{\lambda_{t-1}}{2} \| \theta_{t-1} \|^2 \right),
\end{align*}$$

where $\eta_t$ is the learning rate at epoch $t$, $\gamma$ is the momentum coefficient, and $\lambda$ is the factor of weight decay. Usually, $v_t$ is initialized to be 0.

For ease of analysis, we will use the following equivalent of Definition 1.2.

$$\frac{\theta_t - \theta_{t-1}}{\eta_{t-1}} = \gamma \frac{\theta_{t-1} - \theta_{t-2}}{\eta_{t-2}} - \nabla_{\theta} \left( L(\theta_{t-1}) + \frac{\lambda_{t-1}}{2} \| \theta_{t-1} \|^2 \right),$$

where $\eta_{t-1}$ and $\theta_{t-1}$ must be chosen in a way such that $v_0 = \frac{\theta_0 - \theta_{-1}}{\eta_{-1}}$ is satisfied, e.g. $\theta_{-1} = \theta_0$ and $\eta_{-1}$ could be arbitrary.

2 Deriving Exponential Learning Rate Schedule

As warmup in Section 2.1 we show how to interpret Fixed LR + Fixed WD + Fixed Momentum as an equivalent Exponential Learning Rate + Fixed Momentum. However, usually in deep learning fixed LR is insufficient to reach full training accuracy and instead one needs a few phases where LR is reduced by some factor between phases. Section 2.2 shows how to interpret such a multiphase LR schedule + WD as a certain multiphase exponential LR schedule.

In principle all results can be derived from the Main Theorem 2.5 but that is harder to understand. Hence the simpler Theorems 2.1 and Theorem 2.4 are given separately. There are also other possible variants where momentum can also be changed (as discussed later in experiments section) but for simplicity momentum is left unchanged.

2.1 Replacing WD by Exponential Learning Rate: Case of Constant LR

In this subsection, we use notation of Section 1.2 and assume $\eta$ (LR), $\gamma$ (Momentum) and $\lambda$ (WD) are fixed. The following lemma shows how to replace WD with an exponential LR schedule.

**Theorem 2.1.** The following two sequences of parameters, $\{\theta_t\}_{t=0}^{\infty}$ and $\{\tilde{\theta}_t\}_{t=0}^{\infty}$, define the same sequence of network functions, i.e. $f_{\theta_t} = f_{\tilde{\theta}_t}$, $\forall t \in \mathbb{N}$, given $\tilde{\theta}_0 = \theta_0$, $\tilde{\theta}_{-1} = \theta_{-1} \alpha$.

1. $\theta_t - \theta_{t-1} = \gamma (\theta_{t-1} - \theta_{t-2}) - \eta \nabla_{\theta} (L(\theta_{t-1}) + \frac{\lambda_{t-1}}{2} \| \theta_{t-1} \|^2)$
2. $\tilde{\theta}_t - \tilde{\theta}_{t-1} = \gamma (\tilde{\theta}_{t-1} - \tilde{\theta}_{t-2}) - \alpha^{-2t-1} \eta \nabla_{\theta} L(\tilde{\theta}_{t-1})$
Under review as a conference paper at ICLR 2020

Figure 1: Taking PreResNet32 with published hyperparameters and replacing WD during first phase (Fixed LR) by exponential LR according to Theorem 2.1 to the schedule \( \tilde{\eta}_t = 0.1 \times 1.481^{t} \), momentum 0.9. Plot on right shows weight norm \( \|w_t\|^2 \) of the first convolutional layer in the second residual block grows exponentially, satisfying \( \frac{\|w_t\|^2}{\eta_t} = \text{constant} \). Reason being that according to the proof it is essentially the norm square of the weights when trained with Fixed LR + WD + Momentum, and published hyperparameters kept this norm roughly constant during training.

where \( \alpha \) is a positive root of equation

\[
x^2 - (1 + \gamma - \lambda \eta) x + \gamma = 0, \quad (4)
\]

which is always smaller than 1. When \( \lambda = 0 \), then \( \alpha = \gamma \) is the unique non-zero solution.

Remark 2.2. We implicitly assume \( \lambda \) and \( \gamma \) are small enough such that \( \alpha_t \) are always positive, which is always true in practice, and so is \( P_t \). (In case of constant LR, we need the quadratic equation to have at least one positive root.) The reason behind this requirement is that our definition of "scale invariance" only holds for all positive scalar \( c \), since in neural networks, multiplying activations by \(-1\) before a normalization layer usually yields completely different outputs. Using very high rate of weight decay can flip the sign of the weight.

2.2 Replacing WD by Exponential LR: Case of Multiple LR Phases

Usual practice in deep learning shows that reaching full training accuracy requires reducing the learning rate a few times.

Definition 2.3. Step Decay is the (standard) learning rate schedule, where training has \( K \) phases, where phase \( I \) starts at iteration \( T_I \), and all iterations in phase \( I \) use a fixed learning rate of \( \eta_I \).

Translating WD into the learning rate here leads to the following result. We give an informal version here, and the exact version is Theorem 2.7 later in Section 2.2. The version below is correct up to a correction term in the learning rate schedule. The correction is nontrivial only in the first few iterations of the phase. It is also nontrivial at the end of training when learning rate in the original schedule falls a lot, like \( 0.001 \). These effects are also explored in the experiments, where we empirically find on CIFAR10 that ignoring the correction term does not change performance much.

Theorem 2.4 (Tapered-Exponential LR Schedule). (Informal) If WD is turned off, the following sequence of learning rates \( \{\tilde{\eta}_t\} \) is almost equivalent throughout phase \( I \), \( (\tilde{\eta}_0 = \eta_1) \)

\[
\tilde{\eta}_{t+1} = \begin{cases} 
\tilde{\eta}_t \times \left( \frac{1 + \gamma - \lambda \eta_I + \sqrt{(1 + \gamma - \lambda \eta_I)^2 - 4 \gamma}}{2} \right)^{-2} & \text{if } t \neq T_I \text{ for some } I \\
\tilde{\eta}_t \times \frac{\eta_I}{\eta_{t-1}} \times \left( \frac{1 + \gamma - \lambda \eta_I + \sqrt{(1 + \gamma - \lambda \eta_I)^2 - 4 \gamma}}{2} \right)^{-2} & \text{if } t = T_I \text{ for some } I 
\end{cases} \quad (5)
\]

2.3 Proof Sketch

In this subsection, we will first give the most general statement of the equivalence.
Define the same sequence of network functions, i.e. \( f_\theta = f_{\tilde{\theta}} \), \( \forall t \in \mathbb{N} \), given the initial conditions, \( \theta_0 = P_0 \theta_0, \tilde{\theta}_{-1} = P_{-1} \theta_{-1} \).

1. \( \frac{\theta_t - \theta_{t-1}}{\eta_{t-1}} = \gamma \frac{\theta_{t-1} - \theta_{t-2}}{\eta_{t-2}} - \nabla \theta \left( (L(\theta_{t-1}) + \frac{\lambda_{t-1}}{2} \|\theta_{t-1}\|^2) \right), \) for \( t = 1, 2, \ldots ; \)

2. \( \frac{\tilde{\theta}_t - \tilde{\theta}_{t-1}}{\eta_{t-1}} = \gamma \frac{\tilde{\theta}_{t-1} - \tilde{\theta}_{t-2}}{\eta_{t-2}} - \nabla \theta (\tilde{\theta}_{t-1}), \) for \( t = 1, 2, \ldots , \)

where \( \eta_t = P_t P_{t+1} \eta_t, P_t = \prod_{i=-1}^{t} \alpha_i^{-1}, \forall t \geq -1 \) and \( \alpha_t \) recursively defined as

\[
\alpha_t = -\eta_{t-1} \gamma_{t-1} + 1 + \eta_{t-1} \gamma (1 - \alpha_{t-1}^{-1}), \forall t \geq 1.
\]

needs to be always positive. Here \( \alpha_0, \alpha_{-1} \) are free parameters. Different choice of \( \alpha_0, \alpha_{-1} \) would lead to different trajectory for \( \{\tilde{\theta}_t\} \), but the equality that \( \tilde{\theta}_t = P_t \theta_t \) is always satisfied. If the initial condition is given via \( v_0 \), then it’s also free to choose \( \lambda_{-1}, \theta_{-1} \), as long as \( \frac{\theta_0 - \theta_{-1}}{\lambda_{-1} v_0} = v_0 \).

Now we are ready to give the exact version of Theorem 2.4.

**Definition 2.6** (Tapered-Exponential LR Schedule, Full version). Given a Step Decay LR schedule with \( \{T_i\}_{i=1}^{K}, \{\eta_i\}_{i=1}^{K} \), the corresponding Tapered-Exponential schedule is the following \( (\alpha_0 = \alpha_{-1} = 1) \):

\[
\alpha_t = \begin{cases} 
-\eta_{t-1} \lambda + 1 + \gamma (1 - \alpha_{t-1}^{-1}), \forall T_{I-1} \leq t \leq T_I - 1; \\
-\eta_{t} \lambda + 1 + \frac{\alpha_{t-1}^{-1}}{\eta_{t-1}} \gamma (1 - \alpha_{t-1}^{-1}), \forall t = T_I; 
\end{cases}
\]

\[
P_t = \prod_{i=1}^{t} \alpha_i^{-1}; \quad \tilde{\eta}_t = P_t P_{t+1} \eta_t.
\]

**Theorem 2.7** (Rigorous Theorem 2.4). The schedule of Theorem 2.4 is the same as that of Definition 2.6 throughout phase 1, \( (\eta_0 = \eta_1) \), in the sense that

\[
\frac{\eta_{t-1}}{\eta_t} - \frac{\tilde{\eta}_{t-1}}{\tilde{\eta}_t} < \frac{3}{2} \left( \frac{\gamma}{(z_2^2)^2} \right)^{t-T_I} = \frac{3}{2} \left( \frac{z_2}{z_1} \right)^{t-T_I}, \quad \forall T_I + 1 \leq t \leq T_{I+1} - 1.
\]

where \( z_1 \) is the larger root of \( x^2 - (1 + \gamma - \lambda_1) x + \gamma = 0 \). In Appendix A, we show that \( z_1 \geq 1 - \frac{n}{1-\gamma} \). When \( \lambda_1 \) is small compared to \( 1 - \gamma \), which is usually the case in practice, one could approximate \( z_1 \) by 1.

Before sketching the proof, we restate a simple but key lemma from (Arora et al., 2019).
Lemma 2.8 (Scale Invariance). If for any \( c \in \mathbb{R}^+ \), \( L(\theta) = L(c\theta) \), then

1. \( \langle \nabla_\theta L, \theta \rangle = 0 \);
2. \( \nabla_\theta L|_{\theta=\theta_0} = c\nabla_\theta L|_{\theta=c\theta_0} \), for any \( c > 0 \)

Proof Sketch of Theorem 2.5. This core of this proof relies on the property of scale invariance property of normalization layers, which allows us to have access to the gradients of \( \tilde{\theta}_{t-1} \) from its scaled version, \( \tilde{\theta}_{t-1} \).

The proof is based on induction — assuming \( \tilde{\theta}_{t-1} = P_t\theta_{t-1}, \tilde{\theta}_{t-2} = P_{t-2}\theta_{t-2} \), and using Lemma 2.8 we could replace \( \nabla_\theta L(\tilde{\theta}_{t-1}) \) by \( \frac{\nabla_\theta L(\tilde{\theta}_{t-1})}{P_{t-1}} \), and thus have all three basis for \( \theta_t \) in hand. The goal is now reduced to pick a suitable \( \eta_t \) such that the coefficients in update rule 2 are the same of those in Update rule 1, under global rescaling. This rescaling will be called \( P_t \). In the full proof we show we can always find such \( \eta_t \) given proper initial condition.

3 Example illustrating interplay of WD and BN

The paper so far has shown that effects of different hyperparameters in training are not easily separated, since their combined effect on the trajectory is complicated. We give a simple example to illustrate this, where convergence is guaranteed if we use either BatchNorm or weight decay in isolation, but convergence fails if both are used. (Momentum is turned off for clarity of presentation. See Appendix A).

Setting: Suppose we are fine-tuning the last linear layer of the network, where the input of the last layer is assumed to follow a standard Gaussian distribution \( \mathcal{N}(0, I_m) \), where \( m \) is the input dimension of last layer. We also assume this is a binary classification task with logistic loss, \( l(u, y) = \ln(1 + \exp(-uy)) \), where label \( y \in \{-1, 1\} \) and \( u \in \mathbb{R} \) is the output of the neural net. For simplicity we assume the the input of the last layer are already separable, and w.l.o.g. we assume the label is equal to the sign of the first coordinate of \( x \in \mathbb{R}^m \), namely \( \text{sign}(x_1) \). Thus the training loss and training error are simply

\[
L(w) = \mathbb{E}_{x \sim \mathcal{N}(0, I_m), y = \text{sign}(x_1)} \left[ \ln(1 + \exp(-x^\top wy)) \right],
\]

\[
\Pr_{x \sim \mathcal{N}(0, I_m), y = \text{sign}(x_1)} [x^\top wy \leq 0] = \arccos \frac{w_1}{\|w\|}
\]

Case 1: WD alone: Since both the above function and L2 regularization are convex \( w \), vanilla SGD with suitably small learning rate could get arbitrarily close to the global minimum for this regularized objective, which has 100% training accuracy.

Case 2: BN alone: Add a BN layer after the linear layer (here we use global batch statistics for simplicity), and fix scalar and bias term to 1 and 0. The objective becomes

\[
L_{BN}(w) = \mathbb{E}_{x \sim \mathcal{N}(0, I_m), y = \text{sign}(x_1)} \left[ \ln(1 + \exp(-x^\top wy)) \right].
\]

The following lower bound holds for the norm of the stochastic gradient, (see Appendix A) where \( c_w(x) \in R \) is a random variable with constant distribution (independent of \( w \)).

\[
\| \nabla_w L_{BN}(w, x) \| \geq \frac{c_w(x)}{\|w\|}, \forall w \in \mathbb{R}^m,
\]

By Pythagorean Theorem, \( \|w_{t+1}\|^2 = \|w_t\|^2 + \eta^2 \| \nabla_w L_{BN}(w, x) \|^2 \geq \|w_t\|^2 + \eta^2 \frac{c_w^2}{\|w_t\|} \). As a result, for any fixed learning rate, \( \|w_{t+1}\|^4 \geq \|w_t\|^4 + 2\eta^2 c_w^2 \) grows linearly with high probability. Following the analysis of (Arora et al., 2019), this is like reducing the effective learning rate, and when \( \|w_t\| \) is large enough, the effective learning rate is small enough, and thus SGD can find the local minimum, which is the unique global minimum.
statement: At least once in every \( \eta \) decay (Tapered-Exponential LR schedule contains two parts when entering a new phase I: an instant LR original schedule reduces lr a lot. The experiments explore the effect of this correction term. The translation to exponential lr schedule is exact except for correction term which happens when optimal, then the angle between \( w \) to \( 1 \) to \( 1 \) [Nonconvergence] Starting from iteration any \( T_0 \), with probability arbitrarily close to \( 1 - \delta \) over the randomness of samples, the training error will be larger than \( \varepsilon \) at least once for the following consecutive \( \frac{1}{2(\eta \lambda - 2\pi)} \ln \frac{32\|w_{T_0}\|_2^2 \varepsilon}{\eta \sqrt{m}} + 6 \ln \frac{1}{\delta} \) iterations.

Sketch. (See full proof in Appendix A.) The high level idea of this proof is that if the test error is huge exponential growing. Thus a natural question arises: Can we simplify T-EXP LR schedule by fixing the parameters in the last fully connected layer for scale invariance of the objective.

4 EXPERIMENTS

The translation to exponential lr schedule is exact except for correction term which happens when original schedule reduces lr a lot. The experiments explore the effect of this correction term. The Tapered-Exponential LR schedule contains two parts when entering a new phase I: an instant LR decay (\( \frac{m}{\eta I} \)) and an adjustment of the growth factor. The first part is relative small compared to the huge exponential growing. Thus a natural question arises: Can we simplify T-EXP LR schedule by dropping the part of instant LR decay?

Also, previously we have only verified our equivalence theorem in Step Decay LR schedules. But it’s not sure how would the Exponential LR schedule behave on more rapid time-varying LR schedules such as Cosine LR schedule.

Settings: The initial learning rate is 0.1 and the momentum is 0.9 in all settings. We fix all the scalar and bias of BN, because otherwise they together with the following conv layer grow exponentially, sometimes exceeding the range of float32 when trained with large growth rate for a long time. We fix the parameters in the last fully connected layer for scale invariance of the objective.

4.1 THE BENEFIT OF INSTANT LR DECAY

We tried the following LR schedule (we call it T-EXP v2). Interestingly, up to fluctuations of growth factor when entering a new phase, this schedule is equivalent to a constant LR schedule, but with the weight decay coefficient reduced correspondingly at the beginning of each phase.

\[
\tilde{\eta}_{t+1} = \tilde{\eta}_t \times \left( \frac{1 + \gamma - \lambda \eta t + \sqrt{(1 + \gamma - \lambda \eta t)^2 - 4\gamma}}{2} \right)^{-2}, \quad \forall t \leq T_{t+1} \tag{7}
\]
Figure 3: Instant LR decay is crucial when $\frac{\eta_t}{\eta_{t-1}} - 1$ is very small. When $\frac{\eta_t}{\eta_{t-1}} - 1$ is divided by 100, it would take T-EXP hundreds of epochs to reach its equilibrium. As a result, T-EXP achieves better test accuracy than T-EXP in shorter time. As a comparison, when $\frac{\eta_t}{\eta_{t-1}} - 1$ is divided by 10, it only takes 70 epochs to return to equilibrium. It’s even faster without growth rate decay.

Figure 4: Both Cosine and Step Decay schedule behaves almost the same as their exponential counterpart, as predicted by our equivalence theorem. The (exponential) Cosine LR schedule achieves better test accuracy, with a entirely different trajectory.

4.2 Better Exponential LR Schedule with Cosine LR

We apply the T-EXP LR schedule (Theorem 2.4) on the Cosine LR schedule by (Loshchilov & Hutter, 2016), where the learning rate changes every epoch, and thus correction terms cannot be ignored. Learning rate at epoch $t \leq T$ is defined as:

$$\eta_t = \eta_0 \frac{1 + \cos\left(\frac{t}{T} \pi\right)}{2}.$$  \hspace{1cm} (8)

Our experiments show this hybrid schedule with Cosine LR performs better on CIFAR10, but this finding needs to be verified on other datasets.

5 Conclusions

The paper shows rigorously how BN allows a host of very exotic learning rate schedules in deep learning, and verifies these effects in experiments. The lr increases exponentially in almost every iteration during training. The exponential increase derives from use of weight decay.

This also is a substantial improvement over earlier theoretical analyses of BN, since it accounts for weight decay and momentum, which are always combined in practice.

Our tantalising experiments with a hybrid of exponential and cosine rates suggest that more surprises may lie out there. Our theoretical analysis of interrelatedness of hyperparameters could also lead to faster hyperparameter search.
REFERENCES


A Omitted Proofs

A.1 Omitted Proof in Section 2

Some Facts about Equation 4: Suppose $z_1, z_2 (z_1 \geq z_2)$ are the two real roots of the following equation, we have

$$x^2 - (1 + \gamma - \lambda \eta)x + \gamma = 0$$

1. $0 \leq z_2 \leq z_1 \leq 1$;
2. Let $t = \frac{\lambda \eta}{1 - \gamma}$, we have $1 - z_1 \leq \frac{t}{1 + t}$;
3. if we view $z_1$ as a function of

Proof. Let $f(x) = x^2 - (1 + \gamma - \lambda \eta)x + \gamma$, we have $f(1) = f(\gamma) = \lambda \eta \geq 0$. Note the minimum of $f$ is taken at $x = \frac{1 + \gamma - \lambda \eta}{2} \in [0, 1]$, the both roots of $f(x) = 0$ must lie between 0 and 1, if exists.

$$1 - z_1 = \frac{1 - \gamma + \lambda \eta + \sqrt{(1 - \gamma)^2 - 2(1 + \gamma) \lambda \eta + \lambda^2 \eta^2}}{2}$$

$$= (1 - \gamma) \frac{1 + t - \sqrt{1 - \frac{1 + \gamma}{1 - \gamma} t + t^2}}{2}$$

$$= (1 - \gamma) \frac{2t + 2 \frac{1 + \gamma}{1 - \gamma} t}{2(1 + t + \sqrt{1 - \frac{1 + \gamma}{1 - \gamma} t + t^2})}$$

$$\leq (1 - \gamma) \frac{\frac{4}{1 - \gamma} t}{4(1 + t)}$$

$$= \frac{t}{(1 + t)}$$

Proof of Theorem 2.5. We will prove by induction. By assumption $S(t) : P_t \theta_t = \tilde{\theta}_t$ for $t = -1, 0$. Now we will show that $S(t) \implies S(t+1), \forall t \geq 0$. 


To conclude that $P_t\theta_t = \tilde{\theta}_t$, it suffices to show that the coefficients before $\tilde{\theta}_{t-1}$ is the same to that in (2). In other words, we need to show
\[
-1 + \alpha_t^{-1}(1 - \eta_t - \lambda_{t-1}) = \frac{\gamma(1 - \alpha_t^{-1})}{\tilde{\eta}_{t-2}},
\]
which is equivalent to the definition of $\alpha_t$, (equation 6).

Proof of Theorem 2.4. Assuming $z_f^I$ and $z_f^J(z_f^I > z_f^J)$ are the roots of Equation 4 with $\eta = \eta_I$ we can rewrite the recursion in Theorem 2.5 as the following:
\[
\alpha_t = -\eta_I \lambda + 1 + \gamma(1 - \alpha_t^{-1}) = -(z_f^I + z_f^J) + z_f^I z_f^J \alpha_t^{-1}.
\]
In other words, we have
\[
\alpha_t - z_f^I = \frac{z_f^J}{\alpha_t^{-1}}(\alpha_t^{-1} - z_f^I), t \geq 1,
\]
which means that if we enter each phase with $\alpha_{t_I} \geq z_f^I$, we have $\alpha_t \geq z_f^I$ for the whole phase. Thus we conclude that $\alpha_t$ will be larger than $z_f^I$ for phase 1. It’s not hard to show that since $\eta_I > \eta_J, \forall I < J, z_f^I > z_f^J$ and $z_f^I < z_f^J$. Now we will prove $\alpha_t$ is always larger than $z_f^I$. If $\alpha_t \geq z_f^I$, then we have $\alpha_t \geq z_f^I \geq z_f^J$. In each phase, due to the above inequality, $|\alpha_t - z_f^I|$ is decreasing, which guarantees that if $\alpha_t > z_f^I$ when entering the phase, then $\alpha_t > z_f^I$ for every iteration in this phase.

Note that the when entering a new phase, since $\eta_I < \eta_{t-1}$, $\alpha_{t_I}$ will be larger than it would be without changing phase, which is already larger than $z_f^I$. Thus for the whole process, $\alpha_t$ will be larger than $z_f^I$.

Thus for $\alpha_{t_I-1} \in [z_f^I, \infty)$, $\alpha_t - z_f^I = \frac{z_f^J}{\alpha_t^{-1}}(\alpha_t^{-1} - z_f^I) \leq \frac{z_f^J}{z_f^I}(\alpha_{t_I-1} - z_f^I) = \frac{\gamma}{z_f^I} (\alpha_{t_I-1} - z_f^I)$, which means $\alpha_t$ geometrically converges to its stable fixed point $z_f$. With small $\lambda\eta_I$, one could approximate $z_f^I$ by 1 and thus $\frac{\gamma}{z_f^I}$ by $\gamma$.

Note that $\frac{\eta_{t-1}}{\eta_t} = \alpha_t \alpha_{t+1}$ and $\frac{\eta_{t-1}}{\eta_t} = (z_f^I)^2$. Since that $0.5 \leq z_f^I < \alpha_t \leq 1$, $0.5 \leq z_f^I \leq 1$, we have $|\alpha_{T_I} - z_f^I| \leq 0.5$, and thus $|\alpha_t - z_f^I| \leq \frac{1}{2} (\frac{\gamma}{z_f^I})^{t - T_I}, \forall T_I \leq t \leq T_{I+1} - 1$. Thus we have
\[
\left|\frac{\tilde{\eta}_{t-1}}{\tilde{\eta}_t} - \frac{\tilde{\eta}_{t-1}}{\tilde{\eta}_t}\right| = \alpha_t \alpha_{t+1} - (z_f^I)^2 \leq 3|\alpha_t - z_f^I| \leq \frac{3}{2} \left(\frac{\gamma}{(z_f^I)^2}\right)^{t - T_I}.
\]

\[\square\]
A.2 Proof for Theorem B.1

Proof. Let’s use $R_t, D_t, C_t$ to denote $\|\theta_t\|^2, \|\theta_{t+1} - \theta_t\|^2, \theta_t^T(\theta_{t+1} - \theta_t)$ respectively.

The only property we will use about loss is $\nabla_{\theta_t} L = 0$.

Expanding the square of $\|\theta_{t+1}\|^2 = \|(\theta_{t+1} - \theta_t) + \theta_t\|^2$, we have

$$S(t) : R_{t+1} - R_t = D_t + 2C_t, \forall t.$$  

We also have

$$\frac{C_t}{\eta_t} = \frac{\theta_t^T\theta_{t+1} - \theta_t^T \theta_t}{\eta_t} = \theta_t^T (\gamma \theta_t - \theta_{t-1}) - \lambda_t \theta_t = \frac{\gamma}{\eta_{t-1}}(D_t + C_{t-1}) - \lambda_t R_t,$$

namely,

$$P(t) : \frac{C_t}{\eta_t} - \frac{\gamma C_{t-1}}{\eta_{t-1}} = \frac{\gamma}{\eta_{t-1}} C_{t-1} - \lambda_t R_t.$$  

Simplify $\frac{S(t)}{\eta_t} - \frac{\gamma S(t-1)}{\eta_{t-1}} + P(t)$, we have

$$\frac{R_{t+1} - R_t}{\eta_t} - \gamma \frac{R_t - R_{t-1}}{\eta_{t-1}} = \frac{D_t}{\eta_t} + \frac{\gamma D_{t-1}}{\eta_{t-1}} - 2\lambda_t R_t. \quad (11)$$  

When $\lambda_t = 0$, we have

$$\frac{R_{t+1} - R_t}{\eta_t} = \gamma^{t+1} \frac{R_0 - R_{-1}}{\eta_{t-1}} + \sum_{i=0}^{t} \gamma^{t-i} \left( \frac{D_i}{\eta_i} + \frac{\gamma D_{i-1}}{\eta_{i-1}} \right) \geq \gamma^{t+1} \frac{R_0 - R_{-1}}{\eta_{t-1}}. $$

Further if $\eta_t = \eta$ is a constant, we have

$$R_{t+1} = \sum_{i=0}^{t} \frac{1 - \gamma^{t-i+1}}{1 - \gamma} (D_i + \gamma D_{i-1}) - \gamma \frac{1 - \gamma^{t+1}}{1 - \gamma} (R_0 - R_{-1}),$$

which covers the result without momentum in (Arora et al., 2019) as a special case:

$$R_{t+1} = \sum_{i=0}^{t} D_i, \quad \square$$

Proof of Theorem B.2. Take average of Equation 11 over $t$, when the limits $R_\infty = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} R_t$, $D_\infty = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \|w_t - w_{t+1}\|^2$ exists, we have

$$\frac{1 + \gamma}{\eta} D_\infty = 2\lambda R_\infty,$$

which is

$$\sqrt{\frac{D_\infty}{R_\infty}} = \sqrt{\frac{2\eta \lambda}{1 + \gamma}}. \quad \square$$
A.3 OMITTED PROOFS IN SECTION 3

We will need the following lemma when lower bounding the norm of the stochastic gradient.

**Lemma A.1** (Concentration of Chi-Square). Suppose $X_1, \ldots, X_k \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$, then

$$\Pr \left[ \sum_{i=1}^{k} X_i^2 < k \beta \right] \leq \left( \beta e^{1-\beta} \right)^{\frac{k}{2}}.$$  \hfill (12)

**Proof.** This Chernoff-bound based proof is a special case of Dasgupta & Gupta (2003).

$$\Pr \left[ \sum_{i=1}^{k} X_i^2 < k \beta \right] \leq \left( \beta e^{1-\beta} \right)^{\frac{k}{2}} = \Pr \left[ \exp \left( kt \beta - t \sum_{i=1}^{k} X_i^2 \right) \geq 1 \right]$$

$$\leq \mathbb{E} \left[ \exp \left( kt \beta - t \sum_{i=1}^{k} X_i^2 \right) \right] \quad \text{(Markov Inequality)} \hfill (13)$$

The last equality uses the fact that $\mathbb{E} \left[ t X_i^2 \right] = \frac{1}{\sqrt{2t}}$ for $t < \frac{1}{2}$. The proof is completed by taking $t = \frac{1-\beta}{2\beta}$. \hfill \Box

**Proof. Step 1:** We will use $\hat{u}$ to denote $\frac{u}{\|u\|}$ and $\angle u w$ to denote $\arccos(\hat{u}^T \hat{w})$. Note that training error $\leq \varepsilon$ is equivalent to $\angle e_1 w_t < \varepsilon$. Let $T_1 = \frac{1}{2(\eta \lambda - 2\varepsilon^2)} \ln \frac{32\|w_T_0\|^2 \varepsilon}{\eta \sqrt{m}}$, and $T_2 = 6 \ln \frac{1}{\delta}$. Thus if we assume the training error is smaller than $\varepsilon$ from iteration $T_0$ to $T_0 + T_1 + T_2$, then by spherical triangle inequality, $\angle w_t w_t' \leq \angle e_1 w_t' + \angle e_1 w_t = 2\varepsilon$, for $T_0 \leq t, t' \leq T_0 + T_1 + T_2$.

Now let’s define $w_t' = (1 - \eta \lambda) w_t$ and for any vector $w$, and we have the following two relationships:

1. $\|w_t'\| = (1 - \eta \lambda)\|w\|$.  
2. $\|w_{t+1}\| \leq \frac{\|w_t\|}{\cos 2\varepsilon}$.

The second property is because by Lemma 2.8, $(w_{t+1} - w_t') \perp w_t'$ and by assumption of small error, $\angle w_{t+1} w_t' \leq 2\varepsilon$.

Therefore

$$\frac{\|w_{T_1 + T_0}\|^2}{\|w_{T_1}\|^2} \leq \left( \frac{1 - \eta \lambda}{\cos 2\varepsilon} \right)^{2T_1} \leq \left( \frac{1 - \eta \lambda}{1 - 2\varepsilon^2} \right)^{2T_1} \leq e^{-2T_1(\eta \lambda - 2\varepsilon^2)} = \frac{\eta \sqrt{m}}{32\|w_{T_0}\|^2 \varepsilon}.$$  \hfill (14)

In other word, $\|w_{T_0 + T_1}\|^2 \leq \frac{\eta \sqrt{m}}{32\varepsilon}$. Since $\|w_{T_0 + t}\|$ is monotone decreasing, $\|w_{T_0 + t}\|^2 \leq \frac{\eta \sqrt{m}}{32\varepsilon}$ holds for any $t = T_1, \ldots, T_1 + T_2$.

**Step 2:** We show that the norm of the stochastic gradient is lower bounded with constant probability. Note that $x_t^T \frac{u_t}{\|u_t\|}$ and $\Pi_{u_t} x_t$ are independent gaussian r.v., where $x_t^T \frac{u_t}{\|u_t\|} \sim \mathcal{N}(0, 1)$ and $\|\Pi_{u_t} x_t\|^2 \sim \chi^2(m - 1)$.

$$\Pr \left[ \left| x_t^T \frac{u_t}{\|u_t\|} \right| < 1 \right] \geq \frac{1}{2}, \hfill (15)$$

and by Lemma A.1,
Theorem B.2. For SGD with constant LR $\eta$, weight decay $\lambda$ and momentum $\gamma$, when when the limits $R_{\infty} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \|w_t\|^2$, $D_{\infty} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \|w_{t+1} - w_t\|^2$ exists, we have

$$\sqrt{\frac{D_{\infty}}{R_{\infty}}} = \sqrt{\frac{2\eta\lambda}{1 + \gamma}}.$$
Figure 5: PreResNet32 trained with SGD with 0.9 momentum, 0.0005 WD and 0.1 LR. Here we consider 2 schedules: the orange one divides LR by 10 and multiplies WD by 10 at epoch 80 and 120. The blue divides LR by 10 at epoch 80. Both setting suggest that the instant decay of LR could allow network to stabilize shortly, but in the long run, the average angle between weights in consecutive epochs converges to a value independent of LR, only depending on $LR \times WD$, as predicted by Theorem B.2.