electronics

MDPI

Article

Learning-Rate Annealing Methods for Deep Neural Networks

Kensuke Nakamura [1], Bilel Derbel [2], Kyoung-Jae Won [3] and Byung-Woo Hong [1,*]

[1] Computer Science Department, Chung-Ang University, Seoul 06974, Korea; kensuke@image.cau.ac.kr
[2] Computer Science Department, University of Lille, 59655 Lille, France; bilel.derbel@univ-lille.fr
[3] Biotech Research and Innovation Centre (BRIC), University of Copenhagen, Nørregade 10, 1165 Copenhagen, Denmark; kyoung.won@bric.ku.dk
* Correspondence: hong@cau.ac.kr

Abstract: Deep neural networks (DNNs) have achieved great success in the last decades. DNN is optimized using the stochastic gradient descent (SGD) with learning rate annealing that overtakes the adaptive methods in many tasks. However, there is no common choice regarding the scheduled-annealing for SGD. This paper aims to present empirical analysis of learning rate annealing based on the experimental results using the major data-sets on the image classification that is one of the key applications of the DNNs. Our experiment involves recent deep neural network models in combination with a variety of learning rate annealing methods. We also propose an annealing combining the sigmoid function with warmup that is shown to overtake both the adaptive methods and the other existing schedules in accuracy in most cases with DNNs.

Keywords: learning rate annealing; stochastic gradient descent; image classification

check for updates

Citation: Nakamura, K.; Derbel, B.; Won, K.-J.; Hong, B.-W. Learning-Rate Annealing Methods for Deep Neural Networks. *Electronics* **2021**, *10*, 2029. https://doi.org/10.3390/electronics10162029

Academic Editor: Tomasz Trzcinski

Received: 13 July 2021
Accepted: 19 August 2021
Published: 22 August 2021

1. Introduction

Deep learning is a machine learning paradigm based on deep neural networks (DNNs) that have achieved great success in various fields including segmentation, recognition, and many others. However, the training of DNN is a difficult problem, since it is a global optimization problem to find the parameters updated by the stochastic gradient descent [1–5] (SGD) and its variants.

The learning rate annealing significantly affects the accuracy of the trained model. In the gradient descent methods, the loss gradient is computed using the current model with the training set, and then each model parameter is updated by the loss gradient multiplied by the learning rate. In order to escape the local minima and saddle points and converge to the global optima, the learning rate will start a large value and then shrink to zero. It is ideal that the learning rate for each model parameter is determined automatically based on the convergence of the parameter. To this aim, gradient-based adaptive learning rate methods, for example, RMSprop and Adam, were developed. The adaptive method provides a quick convergence of algorithms in general. Unfortunately, the test accuracy of networks trained using the adaptive method is usually inferior to SGD with scheduled annealing. The scheduled annealing enables us to directly control the stochastic noise that helps the algorithm to escape local minima and saddle points and to converge the global optimal solution. Therefore, the hand-crafted schedule is an essential approach in practice.

However, there is no common choice regarding the scheduled-annealing for the optimization of deep neural networks. On the one hand, the classical annealing, for example, exponential function and staircase, that were designed for shallow networks, may not be suitable for DNNs. On the other hand, the recent warmup strategy designed for DNNs is heuristic and does not provide a smooth decay of step-size. Consequently, researchers in application fields should take time to test a number of annealing methods. More importantly, SGD has been performed using different annealing methods in different papers related to the DNN optimization. These facts motivate us to rethink the annealing strategy of SGD.

This paper aims to present a comparative analysis of learning rate annealing methods based on the experimental results using the major data-sets on the image classification that is one of the key applications of the DNNs. Our experiment involves both shallow and recent deep models in combination with a variety of learning rate annealing methods. We also propose an annealing that combines the sigmoid function with warmup in order to leverage the decay and warmup strategies with a smooth learning rate curve. The proposed annealing is shown to outperform the adaptive methods and the other state-of-the-art schedules, in most cases with DNNs, in our experiments.

We summarize the background in Section 2. We study the related works and propose our method in Section 3. We then present and discuss our empirical results that compare annealing methods in Section 4 and conclude in Section 5.

2. Backgrounds

We overview the general background of deep networks and its optimization before we study related works on the learning rate annealing in the next section.

Progress and application of DNNs: Deep neural networks have made significant progress in a variety of applications for understanding visual scenes [6–8], sound information [9–12], physical motions [13–16], graph-based data representation [17] , and other decision processes [18–21]. Their optimization algorithms related to our work are reviewed in the following.

Variance reduction: The variance of stochastic gradients is detrimental to SGD, motivating variance reduction techniques [22–28] that aim to reduce the variance incurred due to their stochastic process of estimation, and improve the convergence rate mainly for convex optimization, while some are extended to non-convex problems [29–31]. One of the most practical algorithms for better convergence rates includes momentum [32], modified momentum for accelerated gradient [33], and stochastic estimation of accelerated gradient descent [34]. These algorithms are more focused on the efficiency in convergence than the generalization of models for accuracy. We focus on the baseline SGD with learning rate annealing.

Energy landscape: The understanding of energy surface geometry is significant in deep optimization of highly complex non-convex problems. It is preferred to drive a solution toward a plateau in order to yield better generalization [35–37]. Entropy-SGD [36] is an optimization algorithm biased toward such a wide flat local minimum. In our approach, we do not attempt to explicitly measure geometric properties of the loss landscape with extra computational cost, but instead implicitly consider the variance determined by the learning rate annealing.

Batch size selection: There is a trade-off between the computational efficiency and the stability of gradient estimation leading to the selection of their compromise with, generally, a constant, while the learning rate is scheduled to decrease for convergence. The generalization effect of stochastic gradient methods has been analyzed with constant batch size [38,39]. On the other hand, increasing the batch size per iteration with a fixed learning rate has been proposed in [40], where the equivalence of increasing the batch size to learning rate decay is demonstrated. A variety of varying batch size algorithms have been proposed by variance of gradients [41–44]. However, the batch size is usually fixed in practice, since increasing the batch size results in a huge computational cost.

3. Learning-Rate Annealing Methods

3.1. Preliminary

Let us start with a review of the gradient descent method that considers a minimization problem of an objective function $F\colon \mathbb{R}^m \to \mathbb{R}$ in a supervised learning framework: $w^* = \arg\min_w F(w)$, where F is associated with parameters $w = (w_1, w_2, \cdots, w_m)$ in the finite-sum form: $F(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(h_w(x_i), y_i) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$, where $h_w\colon X \to Y$ is a prediction function defined with the associated model parameters w from a data space X

to a label space Y, and $f_i(w) := \ell(h_w(x_i), y_i)$ is a differentiable loss function defined by the discrepancy between the prediction $h_w(x_i)$ and the true label y_i. The objective is to find optimal parameters w^* by minimizing the empirical loss incurred on a given set of training data $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$.

The optimization of supervised learning applications that often require a large number of training data mainly uses the stochastic gradient descent (SGD) that updates solution w^t at each iteration t based on the gradient:

$$w^{t+1} = w^t - \eta^t(\nabla F(w^t) + \xi^t), \tag{1}$$

where $\eta^t \in \mathbb{R}$ is a learning rate and ξ^t is an independent noise process with zero mean. The computation of gradients for the entire training data is computationally expensive and often intractable, so that the stochastic gradient is computed using batch β^t at each iteration t:

$$w^{t+1} = w^t - \eta^t \left(\frac{1}{B} \sum_{i \in \beta^t} \nabla f_i(w^t) \right), \tag{2}$$

where β^t is a subset of the index set $[n] = \{1, 2, \cdots, n\}$ for the training data and $B := |\beta^t|$ is the batch size that is usually fixed during the training process. Thus, the annealing of learning rate (η^t) determines both the step-size and the stochastic noise in the model update.

3.2. Related Works

3.2.1. Adaptive Learning Rate

In the gradient-based optimization, it is desirable to determine the step-size automatically based on the loss gradient that reflects the convergence of each of the unknown parameters. To this aim, the parameter-wise adaptive learning rate scheduling has been developed, such as AdaGrad [45], AdaDelta [46,47], RMSprop [48], and Adam [49], that provide a quick convergence of the algorithm in practice. Recent works of the adaptive method include the combination of Adam with SGD [50], automatic selection of learning rate methods [51], and efficient loss-based method [52]. However, the adaptive method is usually inferior to SGD in accuracy for unknown data in supervised learning, such as the image classification with conventional shallow models [53].

In practice, SGD with scheduled annealing shows better results than the adaptive methods due to the benefits of a generalization and training advantage. Therefore, the hand-crafted schedule is still an essential approach for optimization problems.

3.2.2. Learning-Rate Decay

A schedule defines how things will change over time. In general, learning rate scheduling specifies a certain learning rate for each epoch and batch. There are two types of methods for scheduling global learning rates: the decay, and the cyclical one. The most preferred method is the learning rate annealing that is scheduled to gradually decay the learning rate during the training process. A relatively large step-size is preferred at the initial stages of training in order to obtain a better generalization effect [54]. The shrinkage of the learning rate reduces the stochastic noise. This avoids the oscillation near the optimal point and helps the algorithm to converge.

The popular decay methods of the learning rate are the step (staircase) decay [40] and the exponential decay [55]. The staircase method drops the learning rate in several step intervals and achieves a pleasing result in practice. The exponential decay [55] attenuates the learning rate sequentially for each step and provides a smooth curve. The top row of Figure 1 shows the learning rate schedules, including these methods.

The other one is the cyclical method [56], in which a learning rate period that consists of an upper and a lower bound is repeated during epochs. The observation behind the cyclic method is that increasing the learning rate in the optimization process may have a negative effect, but can result in a better generalization of the trained model [56]. The learning rate period can be a single decay, for example, the exponential decay and step decay method [57], and a triangle function.

3.2.3. Learning-Rate Warmup

The learning rate warmup, for example [58], is a recent approach that uses a relatively small step size at the beginning of the training. The learning rate is increased linearly or non-linearly to a specific value in the first few epochs, and then shrinks to zero. The observations behind the warmup are that: the model parameters are initialized using a random distribution, and thus, the initial model is far from the ideal one; thus, an overly large learning rate causes numerical instability; and training a initial model carefully in the first few epochs may enable us to apply a larger learning rate in the middle stage of the training, resulting in a better regularization [59]. The bottom row of Figure 1 provides the learning rate schedules by the conventional annealing methods with warmup. Among them, the trapezoid [60] is a drastic approach that is designed to train the model using the upper-bound step-size as much as possible.

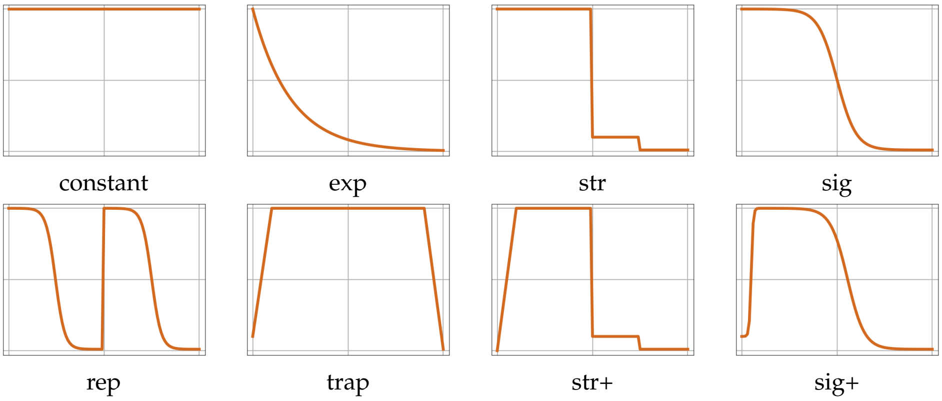

Figure 1. (y-axis) Learning rate over (x-axis) epochs by (**top row**) conventional annealing methods: constant learning rate $\eta = 0.1$, exponential function (exp) [55], staircase (str) [40] with $\eta = 0.1, 0.01, 0.001$, and sigmoid function (sig), and (**bottom row**) warmup methods: twice-repeated sigmoids (rep), trapezoid with 10%-epoch warmup (trap) [60], staircase with the warmup (str+), and our sigmoid with the warmup (sig+).

3.3. Proposed Sigmoid Decay with Warmup

We consider a simple variant of exponential decay, that is, sigmoid decay. We decay the learning rate using a sigmoid function during the training as follows:

$$\eta^t = \eta_{\text{low}} + (\eta_{\text{up}} - \eta_{\text{low}}) \frac{1}{1 + e^{\kappa(2t-1)}} \tag{3}$$

where η^t is the learning rate at step t (scaled in [0, 1] for numerical convenience), and η_{up}, and η_{low}, respectively define the upper and lower bounds of desired learning rates. Note that these parameters are shared by the conventional schedules. κ is a coefficient that can adjust the slope of the learning rate curve, and we use $\kappa = 1/5$.

Moreover, we propose a sigmoid decay with the warmup schedule, which is known as a good heuristic for the training. The right-bottom section of Figure 1 draws the curve by the proposed Sigmoid Decay with Warmup (sig+) that aims to leverage both the decay and warmup while providing a smooth curve of the learning rate.

The proposed annealing (sig+) is designed to leverage both a smooth learning rate curve with a warmup strategy. Concretely, different from the conventional decay methods (exp, str, sig) shown in the top row of Figure 1, our method employs the warmup strategy and that enables us to use a large learning rate with deep neural network models that can be fragile at the initial stage. In contrast to the cyclic method and the existing warm-up methods (rep, trap, st+) shown at the bottom of Figure 1, our sig+ provides a smooth learning rate curve that yields a desirable shrinkage of the stochastic noise in the optimization process.

4. Experimental Results and Discussion

We now experiment the learning rate schedules shown in Figure 1 and the adaptive methods: RMSprop [48], and Adam [49] using both conventional shallow networks and deep neural networks (DNNs) based on major benchmarks shown in Table 1 on the image classification that is one of the important tasks of neural computing.

Table 1. Data-sets on the image classification used in our experiments.

Data-Set	Content	#Class	Pixel Size	Channel	#Training Data	#Test Data
MNIST [61]	handwritten digits	10	28 × 28	gray	60,000	10,000
Fashion-MNIST [62]	fashion items	10	28 × 28	gray	60,000	10,000
SVHN [63]	digits in street	10	32 × 32	color	73,257	26,032
CIFAR-10 [64]	natural photo	10	32 × 32	color	60,000	10,000
CIFAR-100 [64]	natural photo	100	32 × 32	color	60,000	10,000

4.1. Experimental Set-Up

We employ fully connected (fc) networks with two hidden layers (NN-2) and with three hidden layers (NN-3), two convolution layers with two fc layers (LeNet-4), and VGG-9 [65] as the conventional shallow networks. For these shallow networks, we use MNIST [61] and Fashion-MNIST [62] data-sets shown in Table 1. We also conduct experiments using deep neural networks: VGG-19 [65], ResNet-18, ResNet-50 [66,67], DenseConv [68], and GoogLeNet [69]. We employ these models since the convolution kernel in VGG and the skip-connection in ResNet are the two fundamental architectures of recent deep networks; DenseConv is one of the successive deep networks; and GoogLeNet shows a superior performance in many practical tasks as we demonstrate. For these deep networks, we use SVHN [63] and CIFAR [64] benchmarks summarized in Table 1. We employ the batch size of $B = 128$, the momentum of 0.9, the weight-decay of 0.0005, and the epoch size of 100 as practical conditions.

Regarding the hyper-parameters of annealing, we have experimented SGD with three constant learning rates $\eta = 0.1$, $\eta = 0.01$, and $\eta = 0.001$, and observed that $\eta = 0.1$ is too large; $\eta = 0.001$ is too small; and $\eta = 0.01$ tends to be good for the tested networks. Therefore, we set the upper and lower bounds of learning rate curves as $\eta_{\text{up}} = 0.1$ and $\eta_{\text{low}} = 0.001$. The learning rate scale of RMSprop and Adam is set to 0.001 for the shallow models and 0.0001 for the deep models. We assign the warmup step to 10% of a total epoch, and the initial learning rate of the warmup is set to 0.01.

We performed each condition 10 times individually. We drew the average curve of test accuracy over epochs for qualitative evaluation, and also present the average and the maximum of test accuracy for quantitative evaluation within the individual trials.

4.2. Effect of Annealing Methods for Shallow Networks

Figures 2 and 3 show accuracy curves for MNIST and Fashion-MNIST by SGD using learning rate decay and warmup, respectively. The constant $\eta = 0.1$ oscillates over the entire epochs, indicating that the learning rate was too large. Adaptive methods (rms and adam) have converged faster than the others, but fell into a local minima with lower accuracy compared to the decay annealing methods. In contrast to the exponential decay, the sigmoid decay (sig) keeps the learning rate high during the early stages of training, and its accuracy curve rose slowly. However, the sigmoid decay drew a better curve than the exponential decay in the latter half of the training phase due the generalization effect of the larger step-size in the early phase. The accuracy curve of the sigmoid decay (sig) is further compared with the cyclic method (rep) and the warmup variants (trap, str+, and sig+) in Figure 3. The accuracy curves varied along with the designed learning rate curves. The proposed annealing (sig+) follows the original sigmoid but converges to a slightly better solution in most cases using the shallow models.

The accuracy of the shallow networks with MNIST and Fashion-MNIST data-sets are summarized in Table 2. The step decay with warmup and the exponential decay have achieved superior performance in the majority of networks for, respectively, MNIST and Fashion-MNIST. It is difficult to observe the effects of the warmup strategy except for LeNet-4. The adaptive methods, that is, RMSprop and Adam, are fast, but their accuracy is lower than SGD with the constant $\eta = 0.01$. The decaying methods generally showed better accuracy than the constant learning rate $\eta = 0.01$.

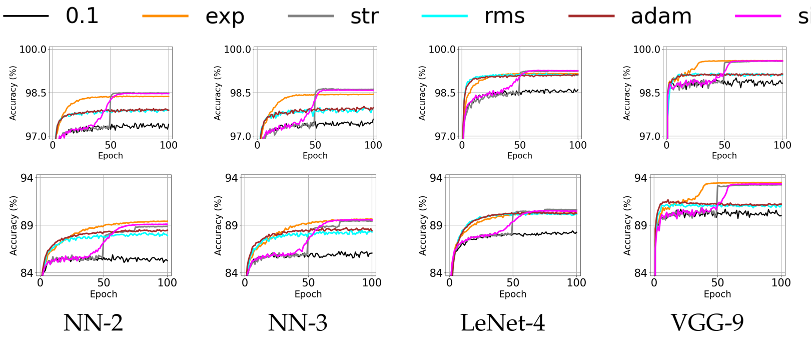

Figure 2. Shallow model with learning rate decay. Accuracy curve over epochs for MNIST (**top row**), and Fashion-MNIST (**bottom row**) by SGD with different learning rate annealings: constant learning rate $\eta = 0.1$, exponential (exp), staircase (str), RMSprop (rms), Adam (adam), and sigmoid (sig): The per-epoch average of validation accuracy over 10 trials is shown in the y-axis where the range of accuracy is fixed for each data-set. Epoch $(0, 1, 2, \ldots, 100)$ is shown in the x-axis.

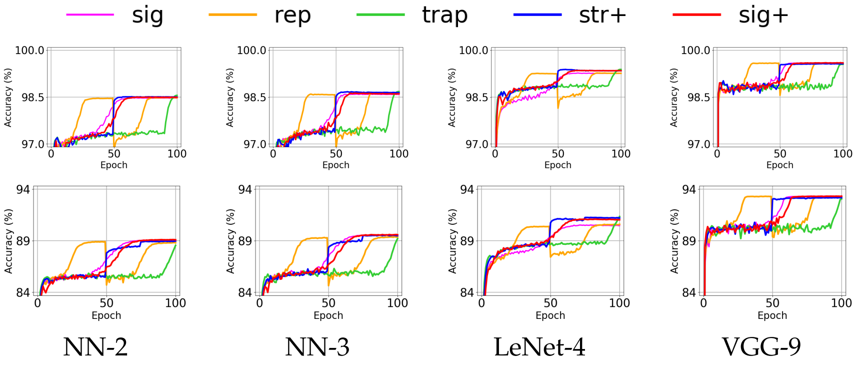

Figure 3. Shallow model with learning rate warmup. Accuracy curve over epochs for MNIST (**top row**), and Fashion-MNIST (**bottom row**) by SGD with different learning rate annealings: sigmoid function without warmup (sig), twice-repeated sigmoids (rep), trapezoid with 10%-epoch warmup (trap), staircase with the warmup (str+), and our sigmoid with the warmup (sig+): The per-epoch average of validation accuracy over 10 trials is shown in the y-axis where the range of accuracy is fixed for each data-set. Epoch $(0, 1, 2, \ldots, 100)$ is shown in the x-axis.

Table 2. Validation accuracy by SGD with different learning rate annealings: constant learning rate $\eta = 0.1, 0.01, 0.001$, exponential function (exp), staircase (str), sigmoid function (sig), RMSprop (rms), Adam (adam), twice-repeated sigmoids (rep), trapezoid with 10%-epoch warmup (trap), staircase with the warmup (str+), and our sigmoid with the warmup (sig+). The average of the last 10%-epoch accuracy and the maximum accuracy over 10 trials are calculated.

(1) Average (Upper Part) and Maximum (Lower Part) of Validation Accuracy for MNIST												
ave	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
NN-2	97.87	98.26	97.90	98.44	98.55	98.52	98.20	98.19	98.53	98.57	**98.58**	98.55
NN-3	97.97	98.27	97.97	98.51	98.69	98.65	98.28	98.25	98.69	98.70	**98.72**	98.67
LeNet-4	98.97	99.23	99.05	99.22	99.30	99.31	99.30	99.27	99.33	99.40	**99.42**	99.40
VGG-9	99.32	99.62	99.33	**99.64**	99.62	99.62	99.38	99.37	99.63	99.56	99.60	99.62
max	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
NN-2	98.06	98.38	98.01	98.59	98.65	98.65	98.33	98.33	98.63	98.67	**98.68**	98.65
NN-3	98.18	98.39	98.09	98.65	98.81	98.79	98.39	98.41	98.80	**98.84**	98.83	98.73
LeNet-4	99.20	99.39	99.18	99.40	99.45	99.48	99.41	99.37	99.48	99.52	**99.53**	99.51
VGG-9	99.43	99.66	99.42	**99.71**	99.66	99.69	99.50	99.43	99.69	99.62	99.65	99.68
(2) Average and Maximum of Validation Accuracy for Fashion-MNIST												
ave	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
NN-2	86.93	89.19	86.89	**89.54**	88.97	89.23	88.87	89.00	89.03	88.54	89.07	89.23
NN-3	87.28	89.42	87.67	**89.80**	89.58	89.71	89.16	89.22	89.49	89.20	89.66	89.72
LeNet-4	89.16	90.68	89.33	90.45	90.77	90.65	90.79	90.76	90.68	91.34	**91.39**	91.22
VGG-9	91.83	93.14	91.79	**93.56**	93.33	93.38	91.97	92.07	93.42	93.06	93.27	93.36
max	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
NN-2	87.25	89.55	87.08	**89.72**	89.28	89.48	89.21	89.40	89.43	88.84	89.36	89.47
NN-3	87.52	89.83	87.90	90.02	89.88	**90.11**	89.49	89.52	89.78	89.42	89.93	90.05
LeNet-4	89.78	91.08	89.86	91.16	91.40	91.36	91.30	91.08	91.46	91.69	**91.72**	91.56
VGG-9	92.10	93.65	92.33	**93.82**	93.55	93.67	92.36	92.46	93.80	93.37	93.64	93.61

4.3. Effect of Annealing Methods for Deep Neural Networks

We provide comparative analysis of the annealing methods with deep neural networks based on SVHN, CIFAR-10, and CIFAR-100 data-sets. Regarding the hyper-parameters of annealing methods, we use the same learning rate values as the shallow networks except for that the initial learning rate of the adaptive methods was tuned to 1/10 of the previous experiment. Figure 4 demonstrates that, just as the shallow networks, the adaptive methods (rms and adam) have converged faster than the others, but to a local minima with low accuracy. Figure 5 shows that the accuracy curve of DNNs also follows the learning rate curve as the shallow models. Among the tested annealing methods, the proposed annealing (sig+) successfully drew the best curves in the last half-epochs.

Table 3 summarizes the test accuracy using DNNs and provides the following observations and intuitions: the employment of a large learning rate in the first and middle stages of training process (e.g., str and sig) results in better accuracy than the exponential one (exp); the smoothing curve (sig) that avoids drastic change of the step-size has led to better accuracy than the non-smooth step function (str); the warmup strategy has further improved DNNs than the original one (i.e., str and str+); and the proposed annealing using sigmoid and warmup together provides the best performance in most cases with the deep networks. Therefore, we conclude that the slope and smoothness of the learning rate curve have a significant influence on the training process, and the proposed method has successfully improved the accuracy of DNNs with the employment of warmups independent of the DNN architecture and data-sets.

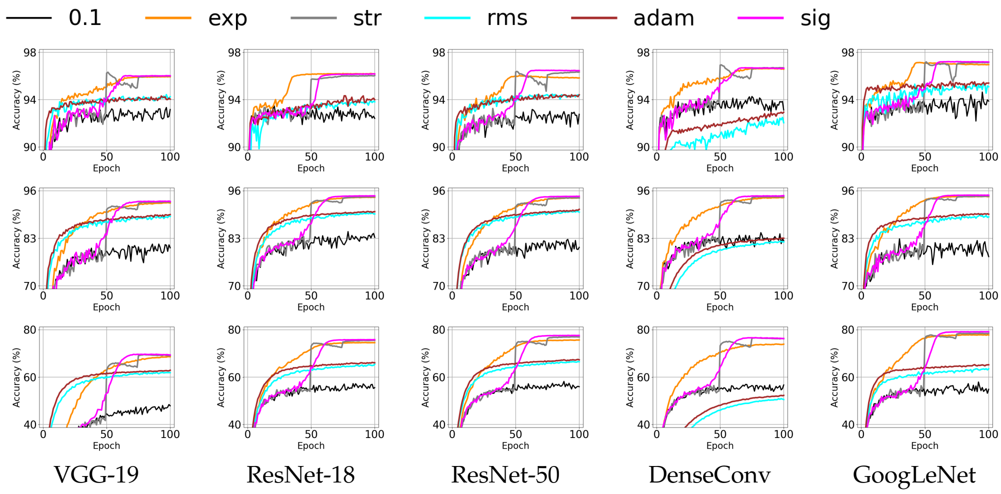

Figure 4. DNNs with learning rate decay. Accuracy curve over epochs for SVHN (**top row**), CIFAR-10 (**middle row**), and CIFAR-100 (**bottom row**) by SGD with different learning rate annealings: constant learning rate $\eta = 0.1$, exponential (exp), staircase (str), RMSprop (rms), Adam (adam), and sigmoid (sig): Per-epoch average of validation accuracy over 10 trials is shown in the y-axis where the range of accuracy is fixed for each data-set. Epoch $(0, 1, 2, \ldots, 100)$ is shown in the x-axis.

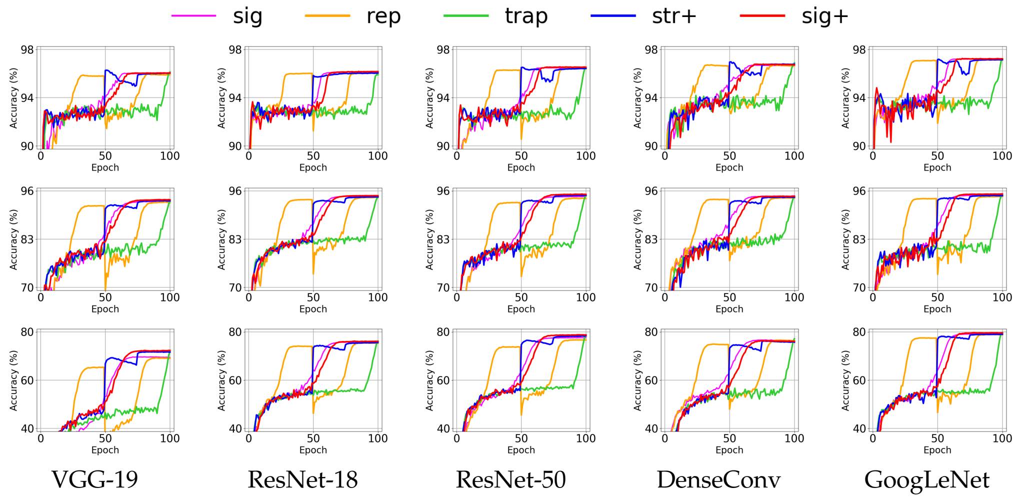

Figure 5. DNNs with learning rate warmup. Accuracy curve over epochs for SVHN (**top row**), CIFAR-10 (**middle row**), and CIFAR-100 (**bottom row**) by SGD with different learning rate annealings: sigmoid function without warmup (sig), twice-repeated sigmoids (rep), trapezoid with 10%-epoch warmup (trap), staircase with the warmup (str+), and our sigmoid with the warmup (sig+): Per-epoch average of validation accuracy over 10 trials is shown in the y-axis where the range of accuracy is fixed for each data-set. Epoch (0, 1, 2, . . . , 100) is shown in the x-axis.

Table 3. Validation accuracy by SGD with different learning rate annealings: constant learning rate $\eta = 0.1, 0.01, 0.001$, exponential function (exp), staircase (str), sigmoid function (sig), RMSprop (rms), Adam (adam), twice-repeated sigmoids (rep), trapezoid with 10%-epoch warmup (trap), staircase with the warmup (str+), and our sigmoid with the warmup (sig+). The average of the last 10%-epoch accuracy and the maximum accuracy over 10 trials are calculated.

(1) Average (Upper Part) and Maximum (Lower Part) of Validation Accuracy for SVHN												
ave	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
VGG-19	87.01	94.98	94.65	95.94	**96.33**	96.06	94.68	94.61	96.02	96.11	96.31	96.08
ResNet-18	94.31	96.04	94.19	**96.23**	96.06	96.20	94.33	95.05	96.10	95.90	96.08	96.20
ResNet-50	94.14	94.98	93.77	96.10	96.34	96.51	94.91	94.84	96.43	96.41	96.56	**96.57**
DenseConv	95.29	95.36	90.80	96.68	96.97	96.76	92.82	93.22	96.84	96.89	**97.01**	96.82
GoogLeNet	95.43	95.88	95.91	97.17	97.22	97.25	95.80	95.93	97.18	97.14	97.22	**97.27**
max	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
VGG-19	94.67	95.14	94.83	96.07	**96.49**	96.25	94.78	94.74	96.12	96.17	96.38	96.23
ResNet-18	94.49	96.29	94.56	96.33	96.19	96.34	94.54	95.19	96.28	96.05	96.22	**96.36**
ResNet-50	94.68	95.24	94.03	96.38	96.59	96.65	95.04	95.08	96.68	96.56	**96.72**	96.65
DenseConv	95.61	95.80	91.06	96.78	97.08	96.90	93.03	93.48	96.92	97.00	**97.12**	96.91
GoogLeNet	95.72	96.30	96.10	97.29	97.36	97.39	95.97	96.14	97.26	97.27	97.36	**97.40**
(2) Average and Maximum of Validation Accuracy for CIFAR-10												
ave	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
VGG-19	84.37	90.56	89.72	92.80	93.04	93.26	89.67	89.89	92.98	93.15	93.46	**93.72**
ResNet-18	86.73	91.59	90.15	94.32	94.37	94.69	90.73	90.91	94.30	94.35	94.48	**94.81**
ResNet-50	85.29	91.81	90.05	94.13	94.17	94.61	91.24	91.28	94.10	94.64	94.87	**95.20**
DenseConv	86.79	88.91	75.85	94.24	94.46	**94.72**	82.97	83.64	94.22	94.35	94.41	94.65
GoogLeNet	86.30	91.64	89.61	94.54	94.57	94.93	90.46	90.59	94.57	94.85	94.83	**95.23**
max	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
VGG-19	85.80	91.07	90.02	92.99	93.46	93.43	89.92	90.07	93.13	93.35	93.68	**93.87**
ResNet-18	87.52	91.81	90.34	94.57	94.50	94.98	91.25	91.36	94.43	94.62	94.77	**95.09**
ResNet-50	87.01	92.09	90.36	94.77	94.40	95.16	91.82	91.73	94.44	94.80	95.04	**95.33**
DenseConv	87.57	89.43	76.54	94.57	94.74	**94.93**	83.60	84.25	94.40	94.64	94.81	94.92
GoogLeNet	87.31	91.95	89.80	94.91	95.04	**95.43**	90.94	90.97	94.94	95.19	94.96	95.41
(3) Average and Maximum of Validation Accuracy for CIFAR-100												
ave	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
VGG-19	51.04	65.09	64.05	68.75	69.65	69.92	63.10	63.49	69.31	71.88	71.86	**72.43**
ResNet-18	60.00	68.02	65.35	74.67	75.65	75.99	66.36	66.90	75.55	76.14	75.66	**76.17**
ResNet-50	60.36	69.67	66.06	75.68	77.05	77.68	67.77	68.11	76.86	78.69	78.42	**78.92**
DenseConv	60.02	62.71	36.21	74.10	76.63	76.83	51.36	52.69	76.65	**77.15**	76.23	76.55
GoogLeNet	60.45	69.05	63.98	78.04	78.69	79.35	65.49	66.37	78.97	**79.74**	79.15	79.73
max	0.1	0.01	0.001	exp	str	sig	rms	adam	rep	trap	str+	sig+
VGG-19	52.60	66.02	64.65	69.10	70.50	70.33	64.37	63.97	70.21	72.52	72.33	**72.70**
ResNet-18	60.79	68.62	65.86	74.83	76.11	**76.69**	67.10	67.59	75.79	76.66	75.82	76.55
ResNet-50	62.00	70.15	67.09	76.32	77.58	78.19	68.71	68.62	77.81	79.18	78.86	**79.38**
DenseConv	60.92	63.48	37.06	74.78	76.84	77.20	52.69	53.54	76.89	**77.55**	76.62	76.99
GoogLeNet	61.66	69.50	64.46	78.38	79.20	79.97	66.96	67.86	79.58	**80.18**	79.70	80.10

5. Conclusions

We have studied learning rate annealing strategies that impact the trained networks, and applied the annealing methods to the shallow networks and the deep networks using the major data-sets. We have performed a comparative analysis of learning rate schedules and adaptive methods, and observed that applying the schedule to SGD has better results in test accuracy than the currently preferred schedules and adaptive methods. Additionally, our results showed that the warmup improves the model accuracy of deep models. Concretely, we have performed that our sigmoid decay with warmup as a learning rate policy leads to superior performance for deep neural networks.

The contribution of the proposed annealing is not limited to image classification. It will be directly applicable to other supervised learning tasks. Moreover, it will be useful for studying the characteristics of generative adversarial networks, since the proposed annealing enables us to control the learning rate while providing a pleasing result, leading to a better understanding of the adversarial losses.

Author Contributions: Conceptualization, B.D., K.-J.W. and B.-W.H.; methodology, B.-W.H.; software, K.N.; validation, B.D., K.-J.W. and B.-W.H.; formal analysis, B.-W.H.; investigation, K.N., B.D. and K.-J.W.; resources, K.N.; data curation, K.N.; writing—original draft preparation, K.N.; writing—review and editing, B.D., K.-J.W. and B.-W.H.; visualization, K.N.; supervision, B.D., K.-J.W. and B.-W.H.; project administration, B.-W.H.; funding acquisition, B.D., K.-J.W. and B.-W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea: NRF-2017R1A2B4 006023 and NRF-2019K1A3A1A77074958, and by the Korea government (MSIT): IITP-2021-0-01341, Artificial Intelligence Graduate School, Chung-Ang University and IITP-2021-0-01574, High-Potential Individuals Global Training Program.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]
2. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Cogn. Model.* **1988**, *5*, 1. [CrossRef]
3. Zhang, T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 116.
4. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 177–186.
5. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Rev.* **2018**, *60*, 223–311. [CrossRef]
6. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 3431–3440.
7. Jin, K.H.; McCann, M.T.; Froustey, E.; Unser, M. Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Trans. Image Process.* **2017**, *26*, 4509–4522. [CrossRef] [PubMed]
8. Cai, J.; Gu, S.; Zhang, L. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Trans. Image Process.* **2018**, *27*, 2049–2062. [CrossRef]
9. Aytar, Y.; Vondrick, C.; Torralba, A. Soundnet: Learning sound representations from unlabeled video. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 892–900.
10. Li, J.; Dai, W.P.; Metze, F.; Qu, S.; Das, S. A comparison of Deep Learning methods for environmental sound detection. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 126–130.
11. Mesaros, A.; Heittola, T.; Benetos, E.; Foster, P.; Lagrange, M.; Virtanen, T.; Plumbley, M.D. Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 379–393. [CrossRef]
12. Van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. *Adv. Neural Inf. Process. Syst.* **2013**, 2643–2651.
13. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; Brox, T. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 2758–2766.
14. Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.
15. Finn, C.; Levine, S. Deep visual foresight for planning robot motion. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2786–2793.

16. Shahroudy, A.; Ng, T.T.; Gong, Y.; Wang, G. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1045–1058. [CrossRef]
17. Ullah, I.; Manzo, M.; Shah, M.; Madden, M. Graph Convolutional Networks: Analysis, improvements and results. *arXiv* **2019**, arXiv:1912.09592.
18. Owens, A.; Efros, A.A. Audio-visual scene analysis with self-supervised multisensory features. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 631–648.
19. Ahmad, K.; Conci, N. How Deep Features Have Improved Event Recognition in Multimedia: A Survey. In *ACM Transactions on Multimedia Computing Communications and Applications*; ACM: New York, NY, USA, 2019.
20. Wang, H.; Wang, N.; Yeung, D.Y. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1235–1244.
21. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & deep learning for recommender systems. In Proceedings of the First Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10.
22. Roux, N.L.; Schmidt, M.; Bach, F. A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 2, pp. 2663–2671.
23. Johnson, R.; Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 315–323.
24. Chatterji, N.S.; Flammarion, N.; Ma, Y.A.; Bartlett, P.L.; Jordan, M.I. On the Theory of Variance Reduction for Stochastic Gradient Monte Carlo. In Proceedings of the 3rd International Conference on Machine Learning (ICML-2018), Stockholm, Sweden, 11–13 July 2018; pp. 764–773.
25. Zhong, W.; Kwok, J. Fast stochastic alternating direction method of multipliers. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 46–54.
26. Shen, Z.; Qian, H.; Zhou, T.; Mu, T. Adaptive Variance Reducing for Stochastic Gradient Descent. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-2016), New York, NY, USA, 9 July 2016; pp. 1990–1996.
27. Zou, D.; Xu, P.; Gu, Q. Stochastic Variance-Reduced Hamilton Monte Carlo Methods. In Proceedings of the 35th International Conference on Machine Learning, Stockholm Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; PMLR: Stockholm Sweden, 2018; Volume 80, pp. 6028–6037.
28. Zhou, K.; Shang, F.; Cheng, J. A Simple Stochastic Variance Reduced Algorithm with Fast Convergence Rates. In Proceedings of the 35th International Conference on Machine Learning, Stockholm Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; PMLR: Stockholm Sweden, 2018; Volume 80, pp. 5980–5989.
29. Allen-Zhu, Z.; Hazan, E. Variance reduction for faster non-convex optimization. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 699–707.
30. Huo, Z.; Huang, H. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
31. Liu, S.; Kailkhura, B.; Chen, P.Y.; Ting, P.; Chang, S.; Amini, L. Zeroth-order stochastic variance reduction for nonconvex optimization. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 3731–3741.
32. Sutton, R. Two problems with back propagation and other steepest descent learning procedures for networks. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Amherst, MA, USA, 15–17 August 1986; pp. 823–832.
33. Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence O $(1/k^2)$. *Doklady AN USSR* **1983**, *269*, 543–547.
34. Kidambi, R.; Netrapalli, P.; Jain, P.; Kakade, S. On the Insufficiency of Existing Momentum Schemes for Stochastic Optimization. In Proceedings of the 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 11–16 February 2018; pp. 1–9. [CrossRef]
35. Hochreiter, S.; Schmidhuber, J. Flat minima. *Neural Comput.* **1997**, *9*, 1–42. [CrossRef] [PubMed]
36. Chaudhari, P.; Choromanska, A.; Soatto, S.; LeCun, Y.; Baldassi, C.; Borgs, C.; Chayes, J.; Sagun, L.; Zecchina, R. Entropy-SGD: Biasing Gradient Descent Into Wide Valleys. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
37. Dinh, L.; Pascanu, R.; Bengio, S.; Bengio, Y. Sharp minima can generalize for deep nets. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 7–9 August 2017; Volume 70, pp. 1019–1028.
38. Hardt, M.; Recht, B.; Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 3rd International Conference on Machine Learning*; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 1225–1234.
39. He, F.; Liu, T.; Tao, D. Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1141–1150.
40. Smith, S.L.; Kindermans, P.J.; Ying, C.; Le, Q.V. Don't decay the learning rate, increase the batch size. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
41. Balles, L.; Romero, J.; Hennig, P. Coupling adaptive batch sizes with learning rates. *arXiv* **2016**, arXiv:1612.05086

42. Levy, K. Online to Offline Conversions, Universality and Adaptive Minibatch Sizes. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1613–1622.
43. De, S.; Yadav, A.; Jacobs, D.; Goldstein, T. Automated inference with adaptive batches. *Artif. Intell. Stat.* **2017**, *54*, 1504–1513.
44. Liu, X.; Hsieh, C.J. Fast Variance Reduction Method with Stochastic Batch Size. In Proceedings of the Thirty-Fifth International Conference on Machine Learning (ICML-2018), Stockholm, Sweden, 10–15 July 2018.
45. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
46. Schaul, T.; Zhang, S.; LeCun, Y. No more pesky learning rates. In Proceedings of the International Conference on Machine Learning (ICML-2013), Atlanta, GA, USA, 16–21 June 2013; pp. 343–351.
47. Zeiler, M.D. ADADELTA: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
48. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Landro, N.; Gallo, I.; La Grassa, R. Combining Optimization Methods Using an Adaptive Meta Optimizer. *Algorithms* **2021**, *14*, 186. [CrossRef]
51. Carvalho, P.; Lourencco, N.; Machado, P. Evolving Learning Rate Optimizers for Deep Neural Networks. *arXiv* **2021**, arXiv:2103.12623.
52. Pouyanfar, S.; Chen, S.C. T-LRA: Trend-based learning rate annealing for deep neural networks. In Proceedings of the 2017 IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, CA, USA, 19–21 April 2017; pp. 50–57.
53. Wilson, A.C.; Roelofs, R.; Stern, M.; Srebro, N.; Recht, B. The marginal value of adaptive gradient methods in machine learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4148–4158.
54. Li, Y.; Wei, C.; Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv* **2019**, arXiv:1907.04595.
55. George, A.P.; Powell, W.B. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Mach. Learn.* **2006**, *65*, 167–198. [CrossRef]
56. Smith, L.N. Cyclical learning rates for training neural networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472.
57. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. In Proceedings of the International Conference on Learning Representations (ICLR-2017), Toulon, France, 24–26 April 2017.
58. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv* **2017**, arXiv:1706.02677.
59. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 558–567.
60. Xing, C.; Arpit, D.; Tsirigotis, C.; Bengio, Y. A walk with sgd. *arXiv* **2018**, arXiv:1802.08770.
61. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
62. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
63. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading Digits in Natural Images with Unsupervised Feature Learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, Granada, Spain, 12–17 December 2011.
64. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Citeseer: Princeton, NJ, USA, 2009.
65. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
66. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
67. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
68. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
69. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.