Learning Rate Investigation

Yu Yang & Liwei Huang

University of Minnesota

December 9, 2019

Overview

Introduction

- Learning rate is a crucial hyper-parameter in deep neural network training, choosing a proper learning rate can
    - reduce training time.
    - boost the model performance.
- In our project, we:
    - propose four decaying sequences and three dynamic updating strategies.
    - investigate the effect of learning rate decaying schemes on convergence time and model performance under two datasets.

Datasets

MNIST

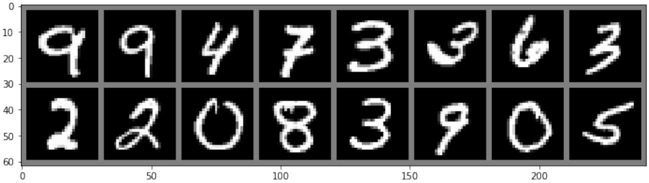


Figure 1: MNIST Example Images

CIFAR-10

Figure 2: CIFAR-10 Example Images

Model and Loss

Convolutional Neural Networks are used for the multi-class classification.

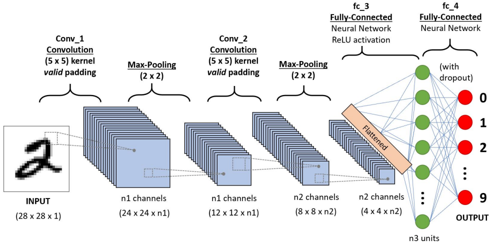


Figure 3: CNN model

Cross-Entropy Loss

$$l(x, class) = -\log\left(\frac{\exp\left(x[class]\right)}{\sum_j \exp\left(x[j]\right)}\right) = -x[class] + \log\left(\sum_j \exp\left(x[j]\right)\right)$$

Decaying Sequences

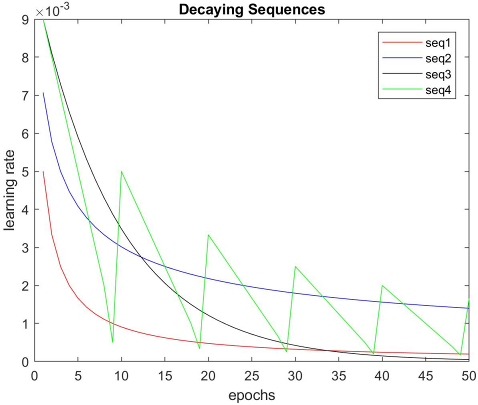


Figure 4: seq1: $r_n = \frac{r_0}{n}$; seq2: $r_n = \frac{r_0}{\sqrt{n}}$; seq3: $r_n = 0.9^n r_0$; seq4: $r_n = q_n r_0$, where $q_n = (1 - \frac{((n-1) \mod 10)}{10})\frac{1}{[\frac{n}{10}]+1}$

Updating Strategies

We propose three dynamic learning rate updating strategies:

- By Epoch: update after every epoch
- By Cutoff: update when the change in loss is smaller than a cutoff
- By Oscillate: update when the loss increases

Algorithm 1 Update by epoch

1: Set initial learning rate: $lr = r_0$
2: Run the model and update parameters with lr
3: Obtain a sequence of learning rate $R = \{r_1, r_2, \cdots, r_n\}$
4: **for** $t \leftarrow 1, 2, \cdots, n$: **do**
5: $\quad lr \leftarrow R.pop(0)$
6: $\quad$ run the model and update parameters with lr

Updating Strategies (Cont.)

Algorithm 2 Update by cutoff

for $t \leftarrow 1, 2, \cdots, n$: **do**
    if $|valid\ loss_t - valid\ loss_{t-1}| < cutoff$ **then**
        $lr \leftarrow R.pop(0)$
        $cutoff = cutoff \times 0.2$
    run the model and update parameters with lr

Algorithm 3 Update by oscillate

for $t \leftarrow 1, 2, \cdots, n$: **do**
    if $valid\ loss_t - valid\ loss_{t-1} > 0$ **then**
        $lr \leftarrow R.pop(0)$
    run the model with learning rate lr.

Experiment Configuration

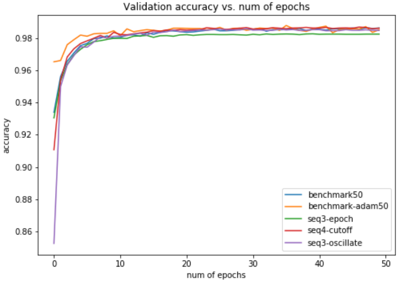
- Settings
  - 12 combinations of the 4 sequences and 3 strategies using SGD optimizer with $r_0 = 0.01$
  - SGD benchmark with fixed learning rate 0.001
  - Adam benchmark with fixed learning rate 0.001
- Metrics
  - Split the datasets into training set and validation set, train the model on the training set, evaluate on the validation set.
  - Use the number of epochs at convergence point as the metric of convergence time.
  - Use the final accuracy as the metric of model performance.
- Comparisons
  - For each strategy, compare four sequences, and pick out the optimal combination.
  - Compare the optimal settings with the two benchmarks.

MNIST: Results & Analyses

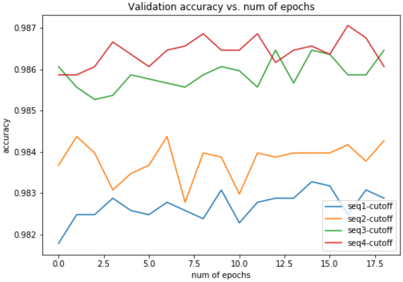
Table 1: MNIST Comparison

Method		Accuracy	Convergence
Benchmark	*SGD*	0.98264	34
	Adam	0.98574	30
By Epoch	*seq1*	0.97608	48
	seq2	0.98230	38
	seq3	0.98263	**19**
	seq4	0.98527	40
By Cutoff	*seq1*	0.98290	32
	seq2	0.98405	28
	seq3	0.98614	29
	seq4	**0.98656**	22
By Oscillate	*seq1*	0.98372	25
	seq2	0.98544	27
	seq3	0.98514	24
	seq4	0.98586	30

MNIST: Results & Analyses (Cont.)



(a) Comparison of best sequences chosen from different criteria



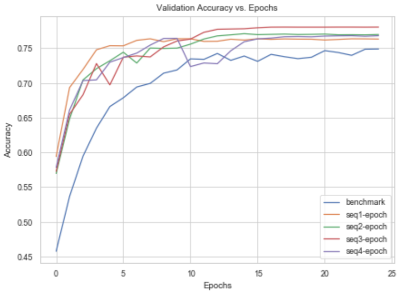
(b) A closer look at final accuracy under "By cutoff"

Figure 5: MNIST illustration

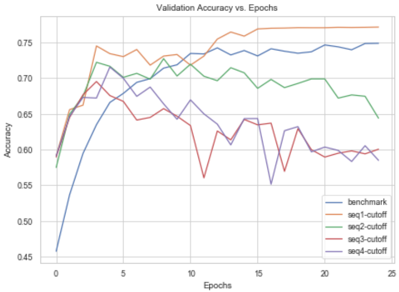
CIFAR-10: Results & Analyses

Table 2: CIFAR Comparison

Method		Accuracy	Convergence
Benchmark	*SGD*	0.7487	25
	Adam	0.7309	17
By Epoch	*seq1*	0.763	**8**
	seq2	0.7696	12
	seq3	**0.7803**	11
	seq4	0.7679	14
By Cutoff	*seq1*	0.7714	15
	seq2	0.6438	25
	seq3	0.6004	25
	seq4	0.5848	25
By Oscillate	*seq1*	0.7677	**8**
	seq2	0.773	11
	seq3	0.7692	15
	seq4	0.7673	12



(a) A closer look at pattern of different sequences under "By epoch"



(b) A closer look at pattern of different sequences under "By cutoff"

Figure 6: CIFAR illustration

Key Results Recap[1]

- MNIST: "seq3 By Epoch" converges the fastest, "seq4 By Cutoff" achieves the highest accuracy.
- CIFAR-10: "seq1 By Oscillate" converges the fastest, "seq3 By Epoch" achieves the highest accuracy.
- It is worth the effort to design a learning rate decaying scheme.
- The results for MNIST and CIFAR-10 differ a lot.
    - A proper decaying scheme can make a great improvement in CIFAR-10, but little change in MNIST.
    - MNIST and CIFAR-10 have different data and model complexity.
- Although seq3 is a convergent sequence, its performance is pretty good.

[1] Check https://github.com/yuyangstatistics/lr_decaying for code and more graphical results.

Conclusion

- Conclusion
  - Proposed four decaying sequences and three updating strategies.
  - Performed experiments and analysis on MNIST and CIFAR-10 datasets under 14 different settings.
  - Described our findings and analyses on the experimental results and provided some guidelines for practitioners.
- Future work
  - Study the effect of learning rate in a more generic framework and on more datasets.

References & Useful Resources

- Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to documentrecognition.Proceedings of the IEEE, 86(11):2278–2324, 1998.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report,Citeseer, 2009.

- Project code github page: `https://github.com/yuyangstatistics/lr_decaying`