AdaGrad and Adam with gradient difference estimator

Lianjia Liu [1] Jaykumar Bhagiya [1] Akansh Maurya [1]

[1]Universität des Saarlandes

Abstract

Gradient descent optimizers such as AdaGrad and Adam are pivotal in machine learning. However, these methods sometimes face challenges like slow convergence and overshooting due to their reliance on gradient norms for adjusting step sizes. This project explores an innovative approach by incorporating gradient differences into the step size calculations of AdaGrad and Adam, aiming to assess the impact on convergence and stability across various machine learning tasks.

Methods

AdaGrad[1] with Gradient Differences

- Standard AdaGrad accumulates the sum of squares of past gradients to adjust learning rates.
- Modified AdaGrad with gradient differences accumulates the sum of squares of differences between successive gradients.

$$G_i^{(t)} = \|g_i^{(t)}\|_2^2 \qquad \theta_i^{(t+1)} = \theta_i^{(t)} - \frac{\eta}{\sqrt{G_i^{(t)}} + \epsilon} \cdot g_i^{(t)} \quad \text{(AdaGrad)}$$

$$\tilde{G}_i^{(t)} = \|\tilde{g}_i^{(t)}\|_2^2 \qquad \text{where} \quad \tilde{g}_i^{(t)} = g_i^{(t)} - g_i^{(t-1)}$$

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \frac{\eta}{\sqrt{\tilde{G}_i^{(t)}} + \epsilon} \cdot \tilde{g}_i^{(t)} \quad \text{(AdaGrad with Difference)} \qquad (1)$$

Adam[3] with Gradient Differences

- Standard Adam uses the first moment (mean) and second moment (uncentered variance) of the gradients to adapt the learning rate for each parameter.
- Modified Adam with gradient differences accumulates the sum of squares of differences between successive gradients.

$$\tilde{m}_i^{(t)} = \beta_1 \tilde{m}_i^{(t-1)} + (1-\beta_1) g_i^{(t)}$$

$$\tilde{v}_i^{(t)} = \beta_2 \tilde{v}_i^{(t-1)} + (1-\beta_2) \|\tilde{g}_i^{(t)}\|_2^2 \qquad \text{where} \quad \tilde{g}_i^{(t)} = g_i^{(t)} - g_i^{(t-1)}$$

$$\hat{m}_i^{(t)} = \frac{\tilde{m}_i^{(t)}}{1-\beta_1^t} \qquad \hat{\tilde{v}}_i^{(t)} = \frac{\tilde{v}_i^{(t)}}{1-\beta_2^t}$$

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \frac{\eta}{\sqrt{\hat{\tilde{v}}_i^{(t)}} + \epsilon} \cdot \hat{m}_i^{(t)} \quad \text{(Adam with Difference)} \qquad (2)$$

Experiment 0 on Logistic Regression: 3 Binary Classification Datasets

This experiments were designed to evaluate the impact of feature shifting on Logistic Regression model on three distinct binary datasets (diabetes, ionosphere, and creditcard).

$$P(Y=1|X) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\ldots+\theta_p x_p)}}$$

Dataset	# of data	# of features
Diabetes	768	8
Credit Card	284807	29
Ionosphere	351	34

Table 1. Dataset Details

The results of our experiments showed that:

1. **Feature shifting** Had a mixed impact on performance, improving the results for some optimizers on some datasets and hindering them on others.
2. **AdaGrad and AdaGradWithDiff** Were more sensitive to feature shifting than Adam and AdamWithDiff.
3. **Adam and AdamWithDiff** were more robust to different datasets and data preprocessing techniques.

Experiment 0 on Logistic Regression: Continued..

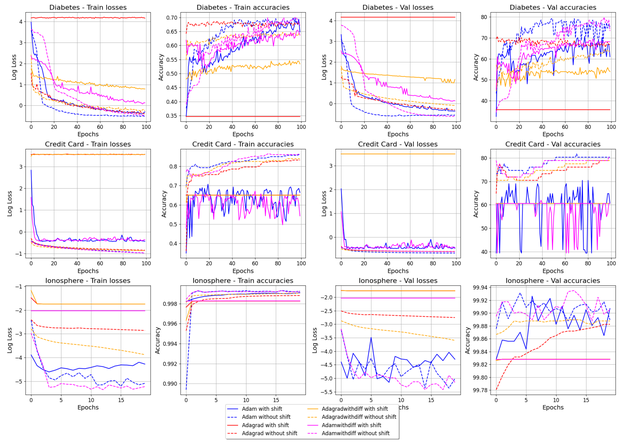


Figure 1. Comparison of losses and accuracies for different optimizers and data shifts on three datasets.

Experiment 1 on Deep Learning: CIFAR10[4] Image Classification

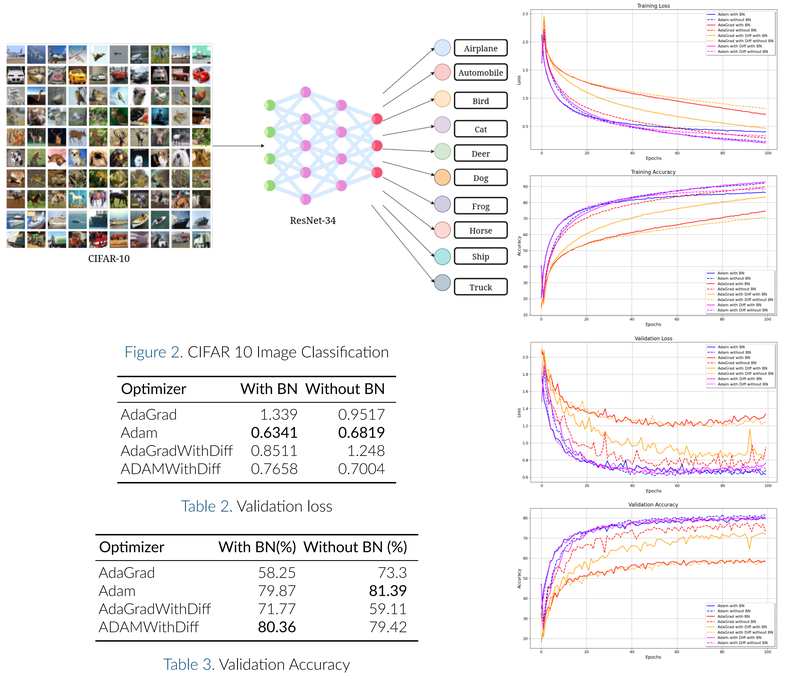


Figure 2. CIFAR 10 Image Classification

Optimizer	With BN	Without BN
AdaGrad	1.339	0.9517
Adam	**0.6341**	**0.6819**
AdaGradWithDiff	0.8511	1.248
ADAMWithDiff	0.7658	0.7004

Table 2. Validation loss

Optimizer	With BN(%)	Without BN (%)
AdaGrad	58.25	73.3
Adam	79.87	**81.39**
AdaGradWithDiff	71.77	59.11
ADAMWithDiff	**80.36**	79.42

Table 3. Validation Accuracy

Experiment 2 on Deep Learning: Image Style Transfer [2]

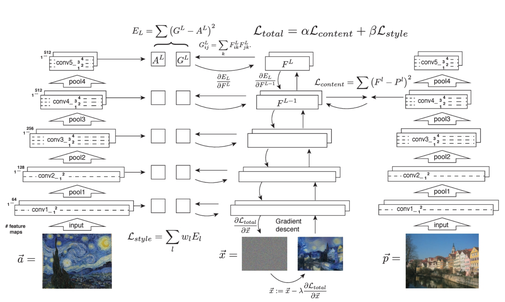
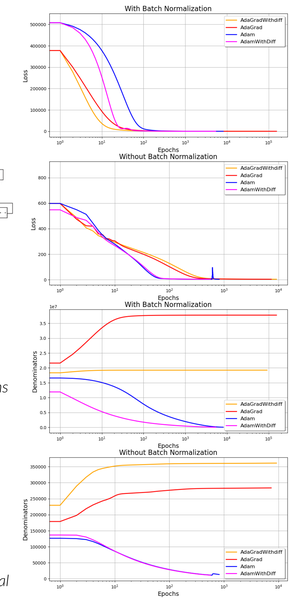


Figure 3. Epochs to converge to the given threshold. **The scales of epochs vary between models which is mainly due to the different settings of loss weights. The number of epochs is not necessarily the same with different parameter settings.*

Models	Content_Weight	Style_Weight	Loss_Threshold
BN	1e2	1e11	4
w/o. BN	1	1e6	3.2

Table 4. Hyperparameters Settings. **Large weights for model with batch normalization is for achieving similar styled-transferred results during empirical experiments.*



Discussion and Future Work

Discussion:

- **Performance comparisons:** Adam and AdamWithDiff perform similarly across all experiments. While Adagrad and AdaGradWithDiff behave differently in various tasks.
- **Robustness:** AdaGrad and AdaGradDiff are more sensitive to the data given their different performance when applying feature shifting and batch normalization. Adam and AdamWithDiff are both more robust to those experiments.

Future Work:

- **Understand more about optimizers' behavior:** Tracking the denominator values in more tasks to get insights of how these optimizers influence the gradient descend process.

References

[1] John Duchi, Elad Hazan, and Yoram Singer.
Adaptive subgradient methods for online learning and stochastic optimization.
Journal of Machine Learning Research, 12(61):2121–2159, 2011.

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge.
Image style transfer using convolutional neural networks.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[3] Diederik P. Kingma and Jimmy Ba.
Adam: A method for stochastic optimization, 2017.

[4] Alex Krizhevsky and Geoffrey Hinton.
Learning multiple layers of features from tiny images.
Technical report, University of Toronto, 2009.