Supplementary for: [Re] AdaBelief Optimizer, Adapting Stepsizes by the Belief in Observed Gradients

Anonymous Author(s)
Affiliation
Address
email

1 Experiments on language modeling

1.1 Penn Treebank dataset

We ran experiments using LSTM [5] models on Penn Treebank dataset [8] and plot train perplexities (Fig. 1) and test perplexities (Fig. 2) for 3 independent runs.

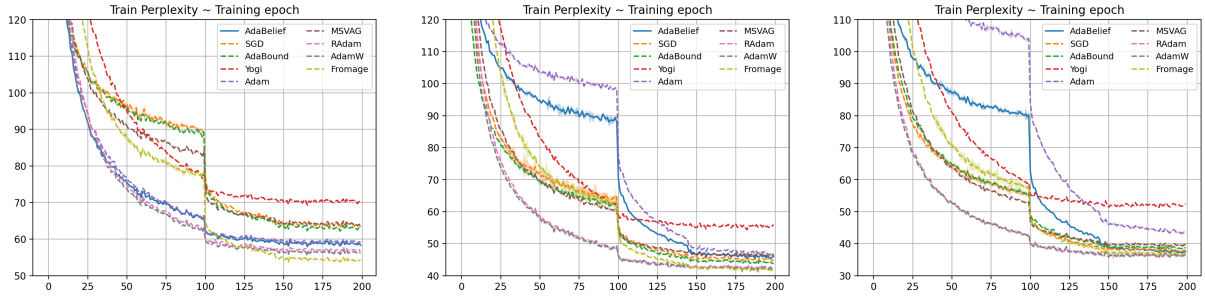


Figure 1: Left to right: Train perplexity ($[\mu \pm \sigma]$) on Penn Treebank for 1,2,3-layer LSTM

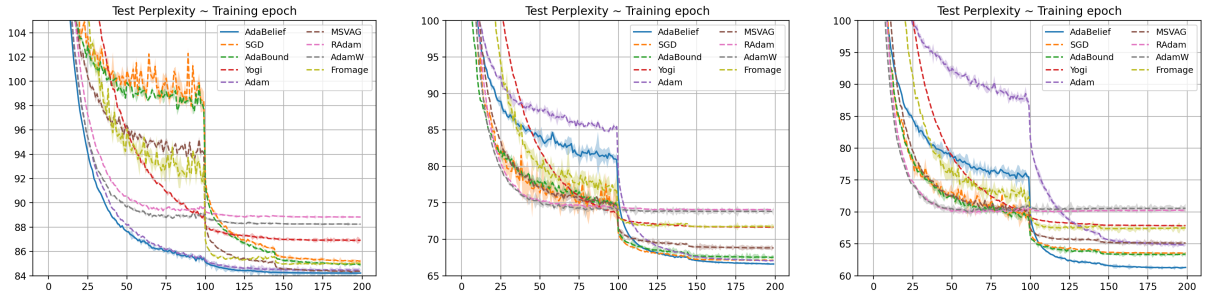


Figure 2: Left to right: Test perplexity ($[\mu \pm \sigma]$) on Penn Treebank for 1,2,3-layer LSTM

1.2 WikiText-2 dataset

We perform experiments on WikiText-2 dataset [9] using LSTM models with Adam [7] and AdaBelief [15] as optimizers. Train perplexities (Fig. 3) and test perplexities (Fig. 4) are reported for 3 independent runs.

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

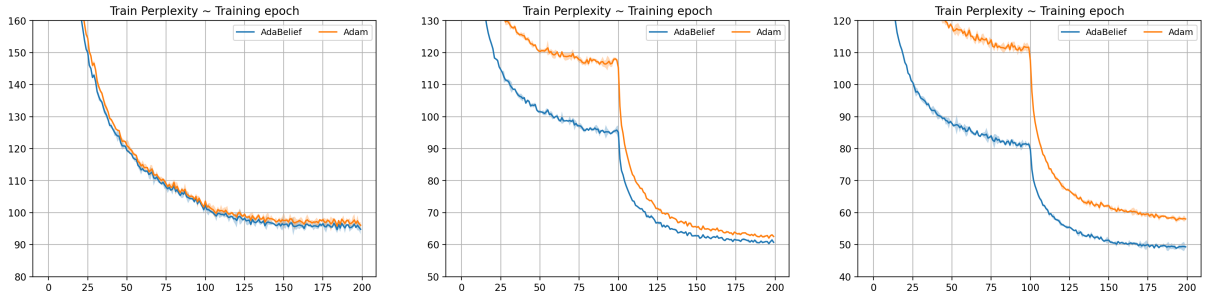


Figure 3: Left to right: Train perplexity ($[\mu \pm \sigma]$) on WikiText-2 for 1,2,3-layer LSTM

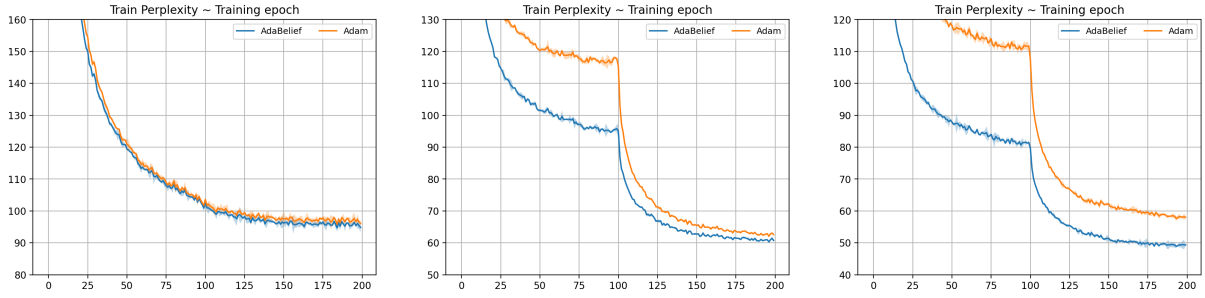


Figure 4: Left to right: Test perplexity ($[\mu \pm \sigma]$) on WikiText-2 for 1,2,3-layer LSTM

2 Experiments on image classification

2.1 Cifar10 and Cifar100

We ran experiments on Cifar10 and Cifar100 on VGG11 [14], ResNet34 [4], DenseNet [6] architectures. We report train accuracies (Fig. 5) and test accuracies (Fig. 6) for 3 independent runs.

3 Experiments on generative modeling

3.1 WGAN

We run experiments on Cifar10 dataset using WGAN [1] for the task of generative modelling. We present a collage of fake images output by WGAN for each optimizer (Fig. 7).

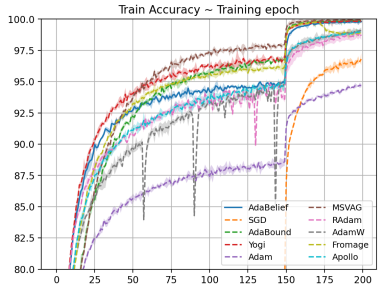
3.2 WGAN-GP

We run experiments on Cifar10 dataset using WGAN-GP [3]. We present a collage of fake images output by WGAN for each optimizer (Fig. 8). Fig. **??** shows the images obtained from training Padam [10] using different partials.

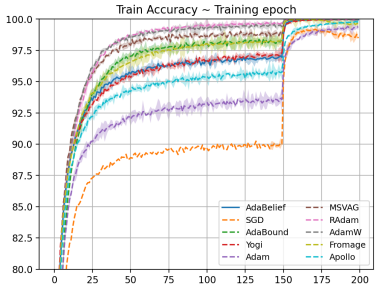
4 Experiments on Reinforcement Learning

4.1 Space Invaders

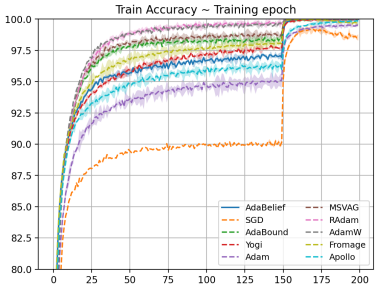
We train an agent to learn to play Space Invaders (Atari Game) using DQN [12] architecture with Adam [7] and AdaBelief [15] as optimizers. Fig. 10 shows the Q value and Fig. 11 plots the reward function against training steps.



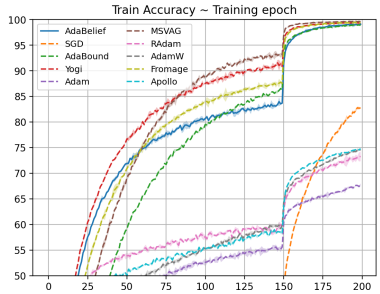
(a) VGG11 on Cifar10



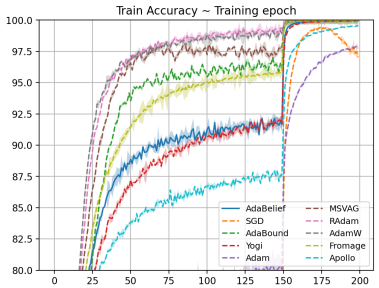
(b) Resnet34 on Cifar10



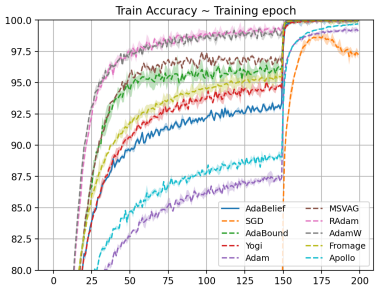
(c) Densenet121 on Cifar10



(d) VGG11 on Cifar100

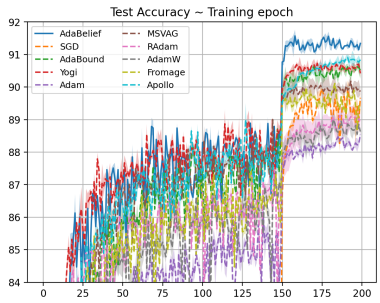


(e) Resnet34 on Cifar100

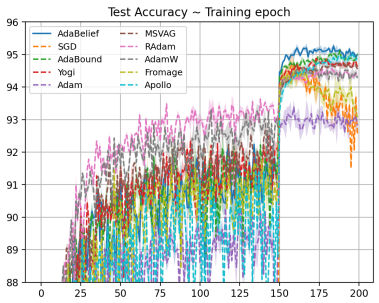


(f) Densenet121 on Cifar100

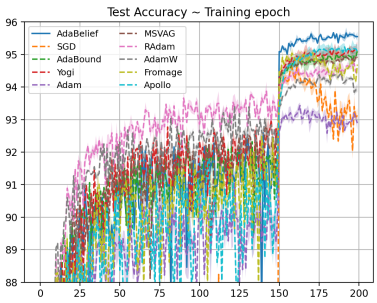
Figure 5: Train accuracy ($[\mu \pm \sigma]$) on Cifar 10 and Cifar 100.



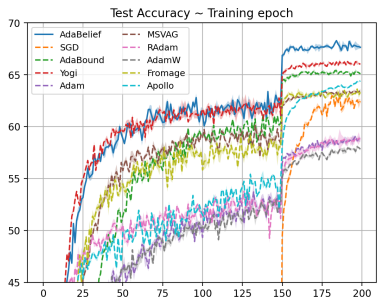
(a) VGG11 on Cifar10



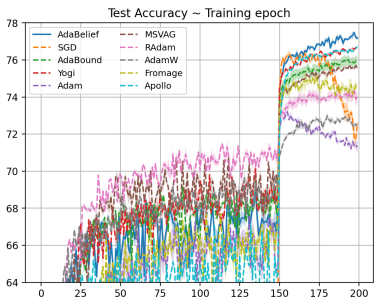
(b) Resnet34 on Cifar10



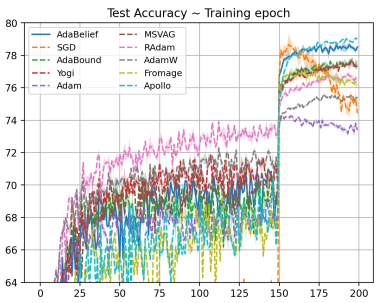
(c) Densenet121 on Cifar10



(d) VGG11 on Cifar100



(e) Resnet34 on Cifar100



(f) Densenet121 on Cifar100

Figure 6: Test accuracy ($[\mu \pm \sigma]$) on Cifar 10 and Cifar 100.

(a) AdaBelief (b) AdaBound (c) Adam

(d) MSVAG (e) Fromage (f) RAdam

(g) RMSProp (h) Yogi (i) SGD

Figure 7: Fake samples from WGAN trained with different optimizers

(a) AdaBelief (b) AdaBound (c) Adam

(d) MSVAG (e) Fromage (f) RAdam

(g) RMSProp (h) Yogi (i) SGD

Figure 8: Fake samples from WGAN-GP trained with different optimizers

(a) Real (b) $p = 0.125$ (c) $p = 0.2$

(d) $p = 0.0625$ (e) $p = 0.25$ (f) $p = 0.4$

Figure 9: Fake samples from Padam WGAN-GP trained with different partials

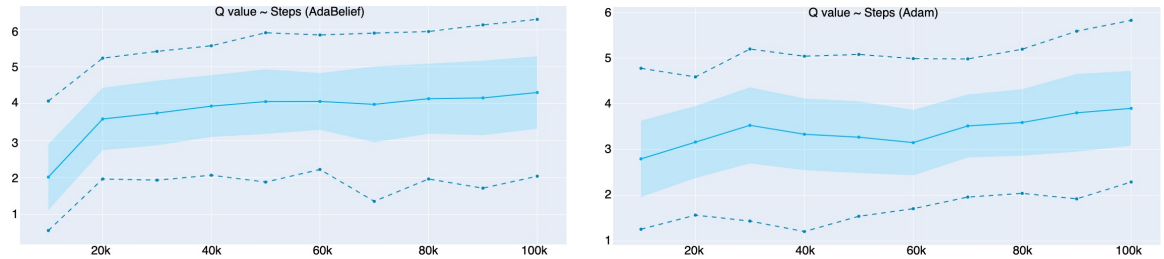


Figure 10: Q value on RL toy experiment using different optimizer

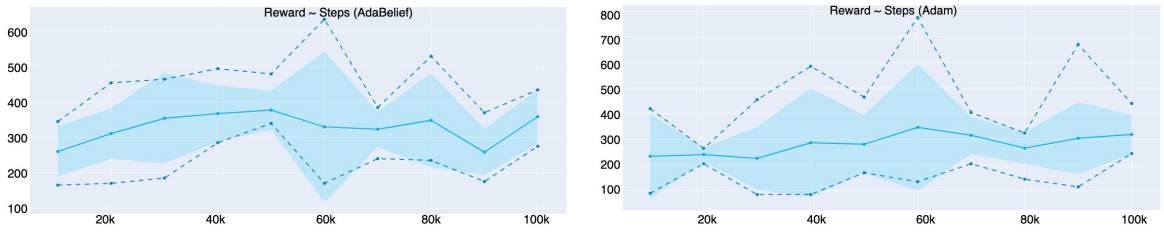
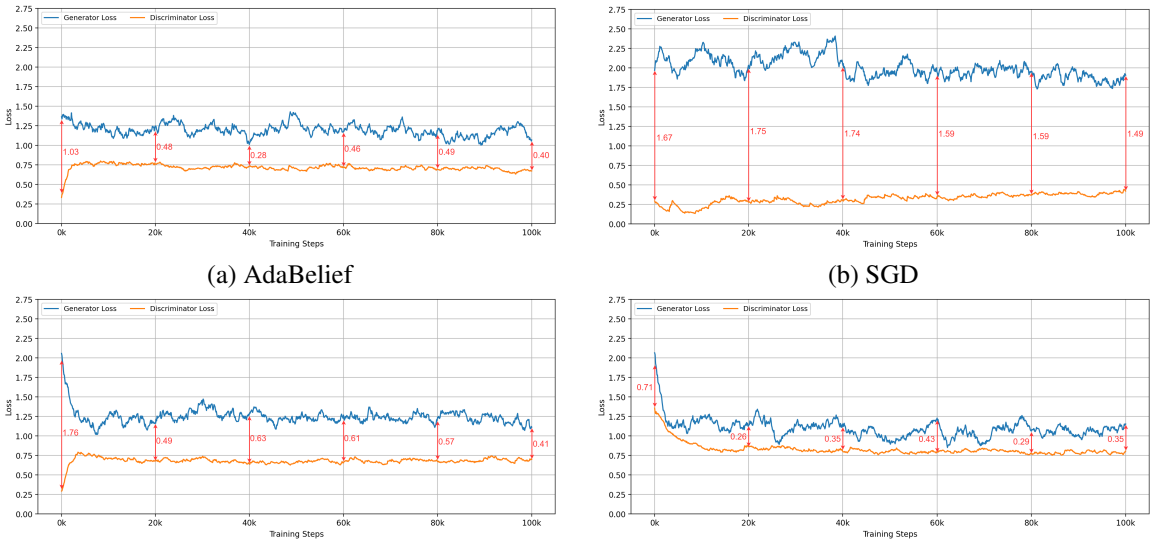


Figure 11: Reward function on RL toy experiment using different optimizer

5 Stability Analysis

5.1 SN-GAN

To analyse the stability of GANs we measure the gap between generator and discriminator losses at different stages of training in SN-GAN [11] on Cifar10 dataset. We do this exercise for AdaBelief [15], SGD [13], Adam [7], RMSProp [2]. Figure 12 highlights the difference in red. A higher gap is attributed to unstable training and a small gap means that the training is stable. From this we can see that the order of stability from most to least follows as: RMSProp, AdaBelief, Adam, SGD.



(a) AdaBelief (b) SGD

(c) Adam (d) RMSProp

Figure 12: SN-GAN Generator Discriminator loss after smoothing the curves with $\beta = 0.95$

6 Convergence Analysis

6.1 Cifar10, Cifar100, LSTM

To understand convergence abilites of different optimizers we make use of definition <x> from the main paper. Table 1 shows the convergence epoch for the different optimizer for experiments performed on Cifar10, Cifar100 using VGG11, ResNet34, DenseNet as backbones and on PTB dataset trained using LSTMs.

Optimizer	**CIFAR-10**			**CIFAR-100**			**LSTM**		
	VGG11	**ResNet34**	**DenseNet121**	**VGG11**	**ResNet34**	**DenseNet121**	**1 layer**	**2 layer**	**3 layer**
Adam	164	161	163	181	160	161	117	160	166
AdaBelief	159	165	168	162	181	172	118	137	154
RAdam	163	176	162	180	169	180	110	106	107
AdamW	160	163	165	174	173	178	115	106	105
Yogi	161	173	166	164	175	174	119	123	119
MSVAG	159	179	163	176	170	166	130	125	119
Fromage	164	182	163	161	175	165	115	117	117
AdaBound	169	182	164	165	168	179	156	129	127
SGD	167	FTC	162	FTC	166	FTC	157	151	123
Apollo	177	FTC	174	186	172	179	-	-	-

Table 1: Epoch of convergence (out of 200) for each optimizer for different experiments. FTC denotes *failed to converge*. AdaBelief converges at epochs similar to other optimizers from Adaptive gradient family.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[2] Alex Graves. Generating sequences with recurrent neural networks, 2014.

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

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

[8] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

[9] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016.

[10] Harshal Mittal, Kartikey Pandey, and Yash Kant. Iclr reproducibility challenge report (padam : Closing the generalization gap of adaptive gradient methods in training deep neural networks), 2019.

[11] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[13] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[15] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C. Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. *Advances in Neural Information Processing Systems*, 33:18795–18806, 2020.