Adaptive Gradient Methods

Skoltech
Skolkovo Institute of Science and Technology

HSE UNIVERSITY

Student: Fakhriddin Tojiboev

Accelerated Gradient Methods vs Adaptive Gradient Methods

Accelerated Gradient Methods:
Pros:
- good generalization

Cons:
- converges slower

Adaptive Gradient Methods:
Pros:
- converges faster
- training stability

Cons:
- poor generalization

AdaBelief algorithm

Algorithm 1: Adam Optimizer

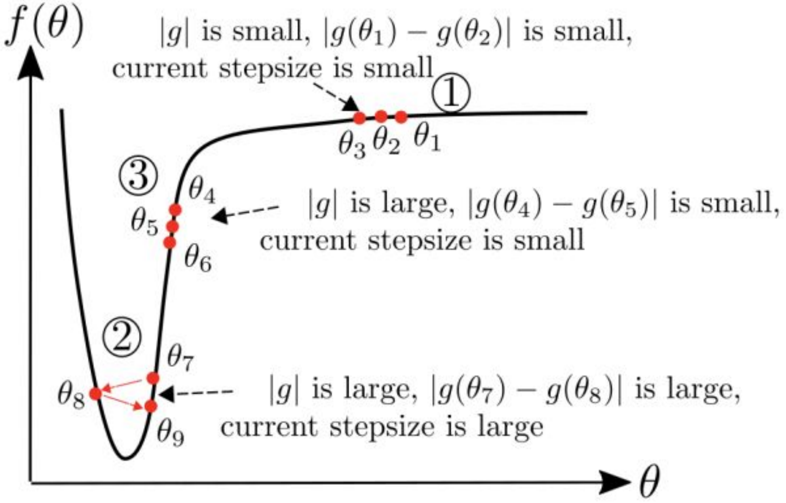
Initialize $\theta_0, m_0 \leftarrow 0$, $v_0 \leftarrow 0, t \leftarrow 0$
While θ_t not converged
$t \leftarrow t+1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
Bias Correction
$\widehat{m_t} \leftarrow \frac{m_t}{1-\beta_1^t}$, $\widehat{v_t} \leftarrow \frac{v_t}{1-\beta_2^t}$
Update
$\theta_t \leftarrow \prod_{\mathcal{F},\sqrt{\widehat{v_t}}} \left(\theta_{t-1} - \frac{\alpha \widehat{m_t}}{\sqrt{\widehat{v_t}}+\epsilon}\right)$

Algorithm 2: AdaBelief Optimizer

Initialize $\theta_0, m_0 \leftarrow 0$, $s_0 \leftarrow 0, t \leftarrow 0$
While θ_t not converged
$t \leftarrow t+1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t - m_t)^2$
Bias Correction
$\widehat{m_t} \leftarrow \frac{m_t}{1-\beta_1^t}$, $\widehat{s_t} \leftarrow \frac{s_t}{1-\beta_2^t}$
Update
$\theta_t \leftarrow \prod_{\mathcal{F},\sqrt{\widehat{s_t}}} \left(\theta_{t-1} - \frac{\alpha \widehat{m_t}}{\sqrt{\widehat{s_t}}+\epsilon}\right)$

AdaBelief algorithm

$$\Delta\theta_t^{SGD} = -\alpha m_t,\ \ \Delta\theta_t^{Adam} = -\alpha m_t/\sqrt{v_t},\ \ \Delta\theta_t^{AdaBelief} = -\alpha m_t/\sqrt{s_t}$$



Algorithm 2: AdaBelief Optimizer

Initialize θ_0, $m_0 \leftarrow 0$, $s_0 \leftarrow 0$, $t \leftarrow 0$
While θ_t not converged
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t - m_t)^2$
Bias Correction
$\widehat{m_t} \leftarrow \frac{m_t}{1-\beta_1^t}$, $\widehat{s_t} \leftarrow \frac{s_t}{1-\beta_2^t}$
Update
$\theta_t \leftarrow \prod_{\mathcal{F},\sqrt{\widehat{s_t}}} \left(\theta_{t-1} - \frac{\alpha \widehat{m_t}}{\sqrt{\widehat{s_t}}+\epsilon}\right)$

AdaBelief theoretical analysis

Theorem 2.2. *(Convergence for non-convex stochastic optimization) Under the assumptions:*

- *f is differentiable; $||\nabla f(x) - \nabla f(y)|| \leq L||x-y||$, $\forall x, y$; f is also lower bounded.*
- *The noisy gradient is unbiased, and has independent noise, i.e. $g_t = \nabla f(\theta_t) + \zeta_t, \mathbb{E}\zeta_t = 0, \zeta_t \perp \zeta_j$, $\forall t, j \in \mathbb{N}, t \neq j$.*
- *At step t, the algorithm can access a bounded noisy gradient, and the true gradient is also bounded. i.e. $||\nabla f(\theta_t)|| \leq H$, $||g_t|| \leq H$, $\forall t > 1$.*

Assume $\min_{j \in [d]} (s_1)_j \geq c > 0$, noise in gradient has bounded variance, $\mathrm{Var}(g_t) = \sigma_t^2 \leq \sigma^2$, $s_t \leq s_{t+1}, \forall t \in \mathbb{N}$, then the proposed algorithm satisfies:

$$\min_{t \in [T]} \mathbb{E}\Big|\Big|\nabla f(\theta_t)\Big|\Big|^2 \leq \frac{H}{\sqrt{T}\alpha}\Big[\frac{C_1 \alpha^2 (H^2 + \sigma^2)(1 + \log T)}{c} + C_2 \frac{d\alpha}{\sqrt{c}} + C_3 \frac{d\alpha^2}{c} + C_4\Big]$$

as in [27], C_1, C_2, C_3 are constants independent of d and T, and C_4 is a constant independent of T.

Corollary 2.2.1. *If $c > C_1 H$ and assumptions for Theorem 2.2 are satisfied, we have:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\Big[\alpha_t^2\Big|\Big|\nabla f(\theta_t)\Big|\Big|^2\Big] \leq \frac{1}{T}\frac{1}{\frac{1}{H} - \frac{C_1}{c}}\Big[\frac{C_1 \alpha^2 \sigma^2}{c}\Big(1 + \log T\Big) + C_2 \frac{d\alpha}{\sqrt{c}} + C_3 \frac{d\alpha^2}{c} + C_4\Big]$$

ACProp algorithm

Algorithm 1: AdaBelief

Initialize x_0, $m_0 \leftarrow 0$, $s_0 \leftarrow 0$, $t \leftarrow 0$
While x_t not converged

$$t \leftarrow t + 1$$
$$g_t \leftarrow \nabla_x f_t(x_{t-1})$$
$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$s_t \leftarrow \beta_2 s_{t-1} + (1 - \beta_2)(g_t - m_t)^2$$
$$x_t \leftarrow \prod_{\mathcal{F}, \sqrt{s_t}} \left(x_{t-1} - \frac{\alpha}{\sqrt{s_t + \epsilon}} m_t \right)$$

Algorithm 2: ACProp

Initialize x_0, $m_0 \leftarrow 0$, $s_0 \leftarrow 0$, $t \leftarrow 0$
While x_t not converged

$$t \leftarrow t + 1$$
$$g_t \leftarrow \nabla_x f_t(x_{t-1})$$
$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$x_t \leftarrow \prod_{\mathcal{F}, \sqrt{s_{t-1}}} \left(x_{t-1} - \frac{\alpha}{\sqrt{s_{t-1} + \epsilon}} g_t \right)$$
$$s_t \leftarrow \beta_2 s_{t-1} + (1 - \beta_2)(g_t - m_t)^2$$

ACProp theoretical analysis

Theorem 4.1 (convergence for stochastic non-convex case). *Under the following assumptions:*

- *f is continuously differentiable, f is lower-bounded by f^* and upper bounded by M_f. $\nabla f(x)$ is globally Lipschitz continuous with constant L:*

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y|| \tag{3}$$

- *For any iteration t, g_t is an unbiased estimator of $\nabla f(x_t)$ with variance bounded by σ^2. Assume norm of g_t is bounded by M_g.*

$$\mathbb{E}\big[g_t\big] = \nabla f(x_t) \quad \mathbb{E}\big[||g_t - \nabla f(x_t)||^2\big] \leq \sigma^2 \tag{4}$$

then for $\beta_1, \beta_2 \in [0, 1)$, with learning rate schedule as: $\alpha_t = \alpha_0 t^{-\eta}$, $\alpha_0 \leq \frac{C_l}{LC_u^2}$, $\eta \in [0.5, 1)$ for the sequence $\{x_t\}$ generated by ACProp, we have

$$\frac{1}{T}\sum_{t=1}^{T}\Big|\Big|\nabla f(x_t)\Big|\Big|^2 \leq \frac{2}{C_l}\Big[(M_f - f^*)\alpha_0 T^{\eta-1} + \frac{LC_u^2\sigma^2\alpha_0}{2(1-\eta)}T^{-\eta}\Big] \tag{5}$$

where C_l and C_u are scalars representing the lower and upper bound for A_t, e.g. $C_l I \preceq A_t \preceq C_u I$, where $A \preceq B$ represents $B - A$ is semi-positive-definite.

Experimental setup

Experiments were conducted on Google Colab. To run one experiment it took 5-6 hours in average.

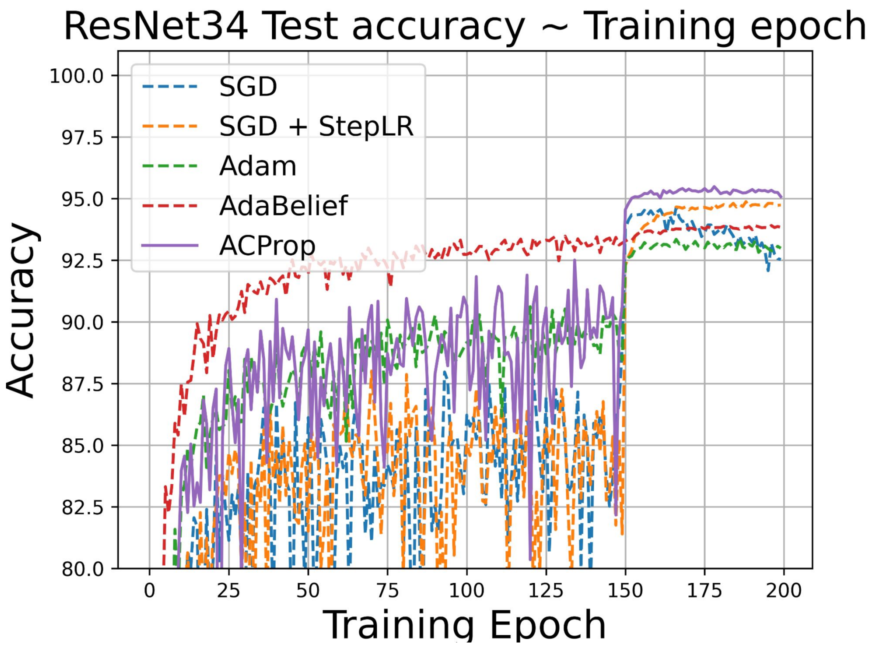
Dataset: CIFAR10

Models: ResNet18, ResNet34

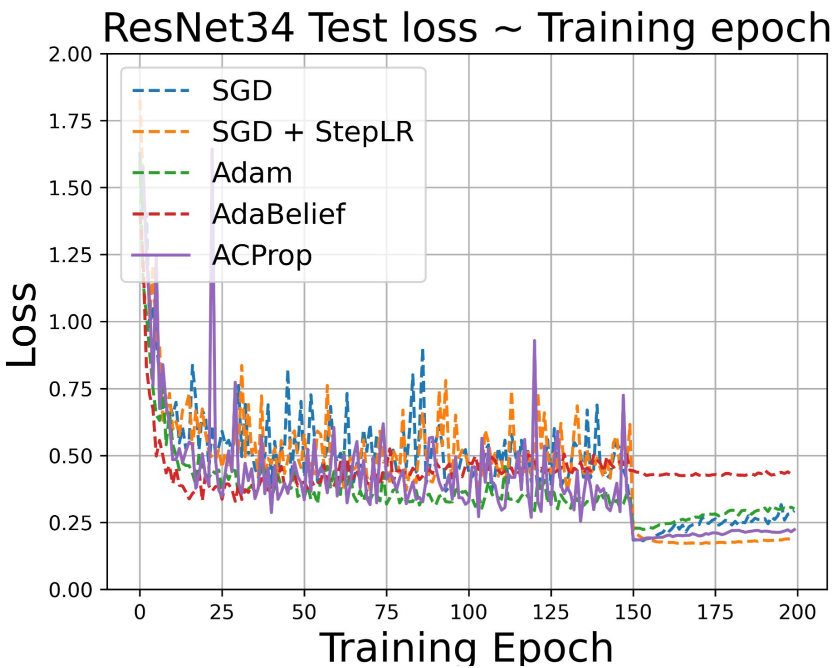
Optimizers:

- SGD lr=0.1 momentum=0.9 wdecay=0.0005
- Adam lr=0.001 betas=(0.9, 0.999) wdecay=0.0005 eps=1e-8
- AdaBelief lr=0.001 betas=(0.9, 0.999) eps=1e-8 wdecay=0.0005
- ACProp lr=0.001 betas=(0.9, 0.999) eps=1e-8 wdecay=0.0005

Results

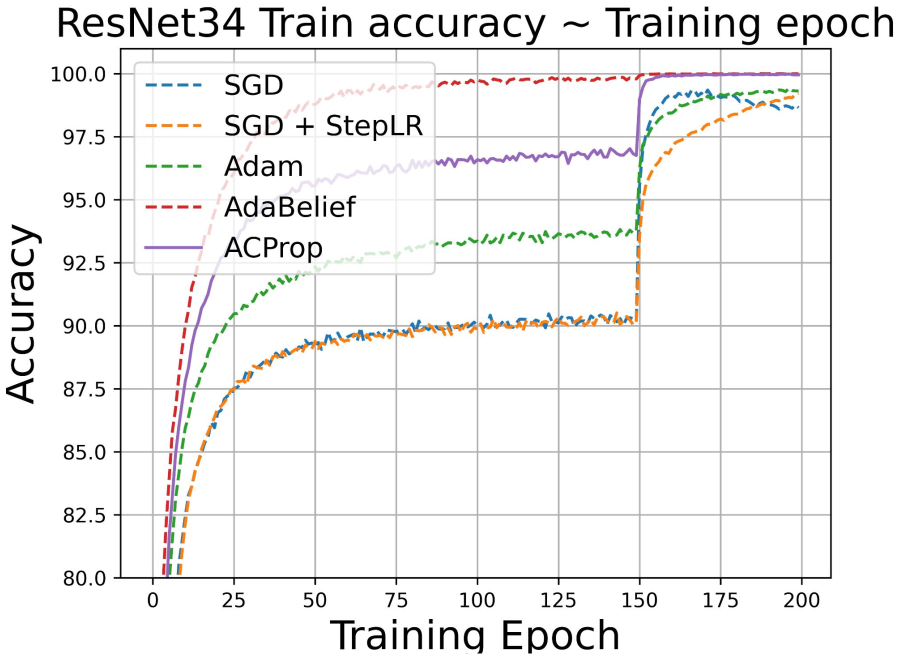


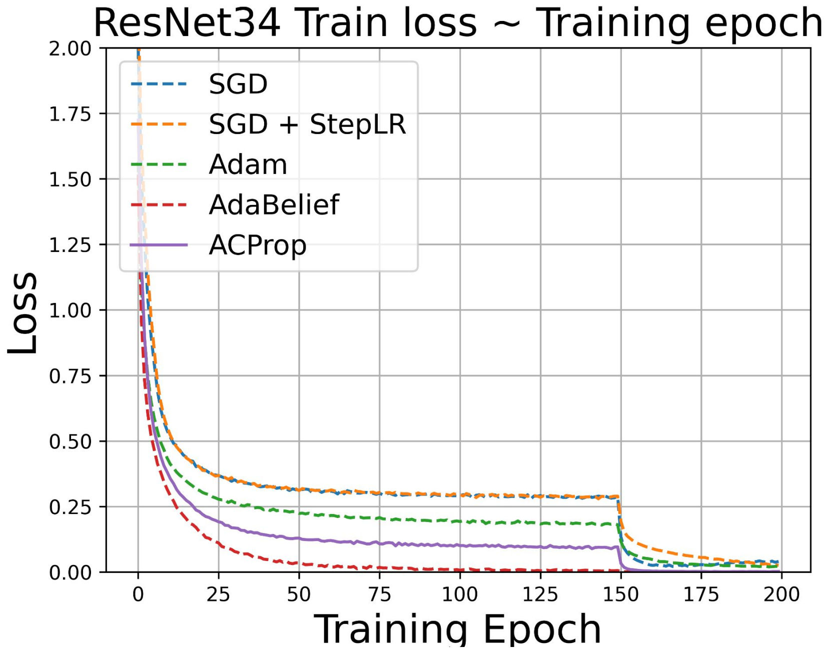
ResNet34 Test accuracy ~ Training epoch
Accuracy
Training Epoch
SGD
SGD + StepLR
Adam
AdaBelief
ACProp
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
0
25
50
75
100
125
150
175
200



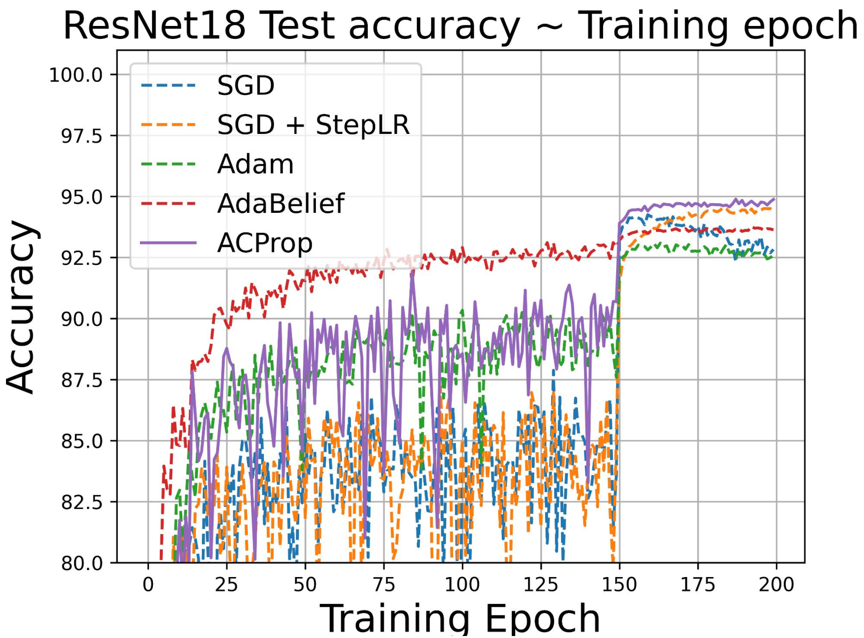
ResNet34 Test loss ~ Training epoch
Loss
Training Epoch
SGD
SGD + StepLR
Adam
AdaBelief
ACProp
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
125
150
175
200

Results

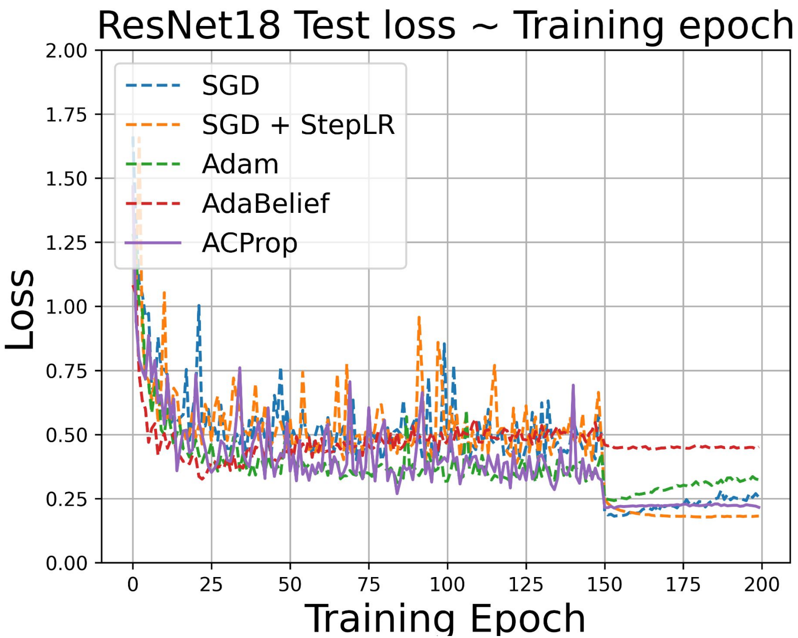

ResNet34 Train accuracy ~ Training epoch
Accuracy
Training Epoch
SGD
SGD + StepLR
Adam
AdaBelief
ACProp
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
0
25
50
75
100
125
150
175
200


ResNet34 Train loss ~ Training epoch
Loss
Training Epoch
SGD
SGD + StepLR
Adam
AdaBelief
ACProp
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
125
150
175
200

Results

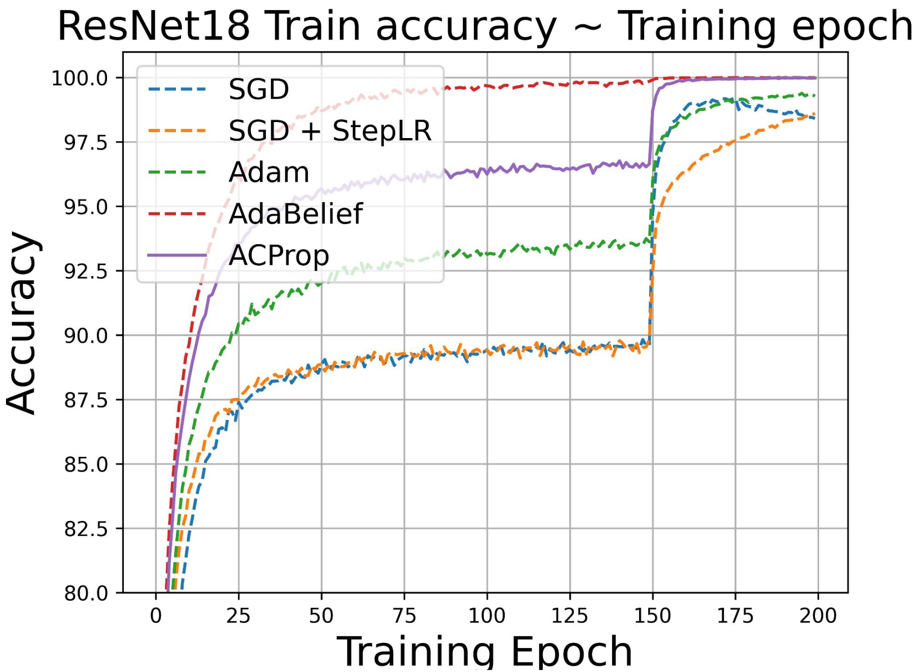


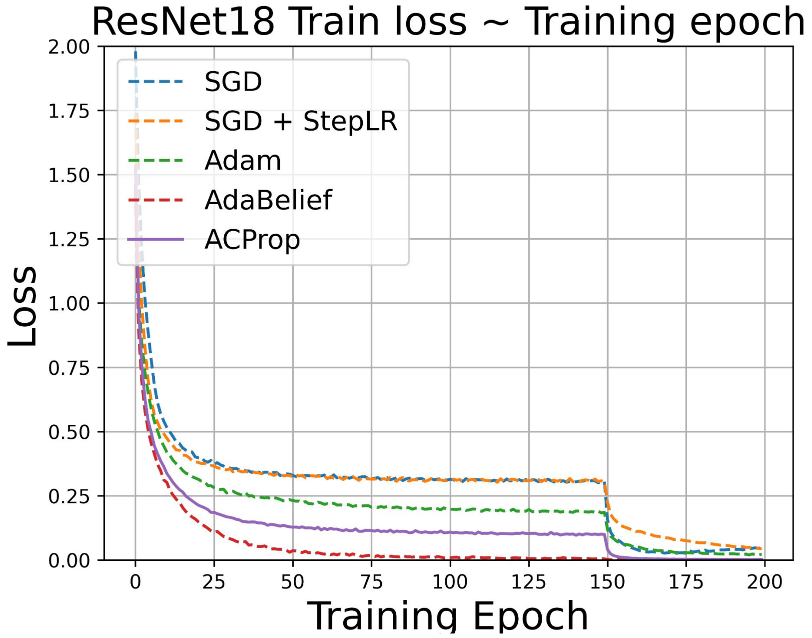
ResNet18 Test accuracy ~ Training epoch
Accuracy
Training Epoch
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
0
25
50
75
100
125
150
175
200
SGD
SGD + StepLR
Adam
AdaBelief
ACProp



ResNet18 Test loss ~ Training epoch
Loss
Training Epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
125
150
175
200
SGD
SGD + StepLR
Adam
AdaBelief
ACProp

Results


ResNet18 Train accuracy ~ Training epoch
Accuracy
Training Epoch
SGD
SGD + StepLR
Adam
AdaBelief
ACProp
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
0
25
50
75
100
125
150
175
200


ResNet18 Train loss ~ Training epoch
Loss
Training Epoch
SGD
SGD + StepLR
Adam
AdaBelief
ACProp
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
125
150
175
200

References

1. Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar Tatikonda, Nicha Dvornek, Xenophon Papademetris, James S. Duncan. “AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients”, NIPS 2020
2. Juntang Zhuang, Yifan Ding, Tommy Tang, Nicha Dvornek, Sekhar Tatikonda, James S. Duncan. “Momentum Centering and Asynchronous Update for Adaptive Gradient Methods”, NIPS 2021