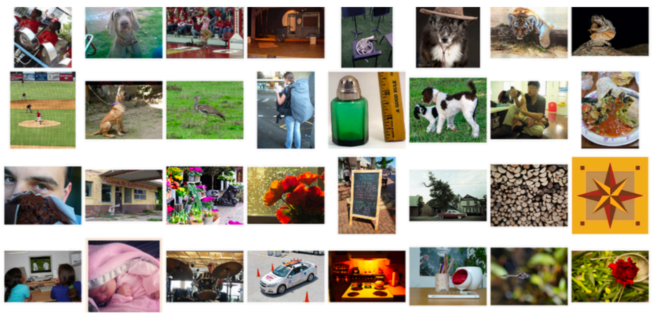
New Federated Learning Algorithms for Deep Learning with Unbounded Smooth Landscape

Mingrui Liu
Department of Computer Science
George Mason University
mingruil@gmu.edu

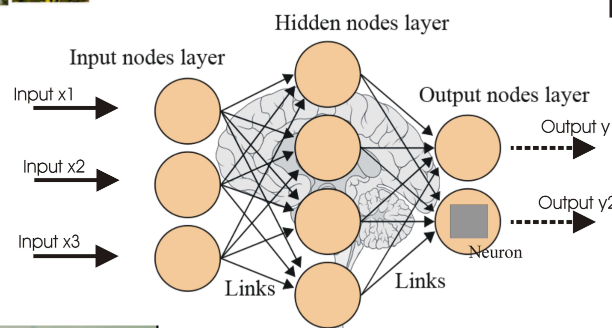
October 25, 2023 (Rensselaer Polytechnic Institute)

Empirical Success of Deep Learning

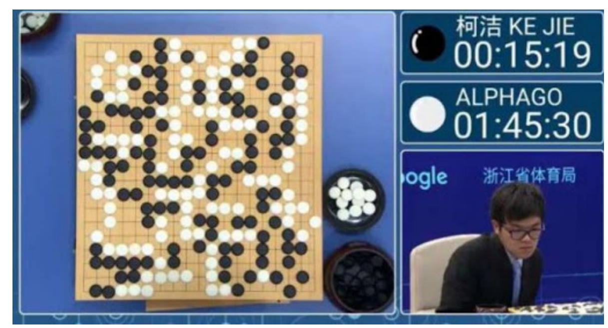


Computer Vision (Convolutional NN)

Natural Language Processing (Recurrent NN, Transformer)

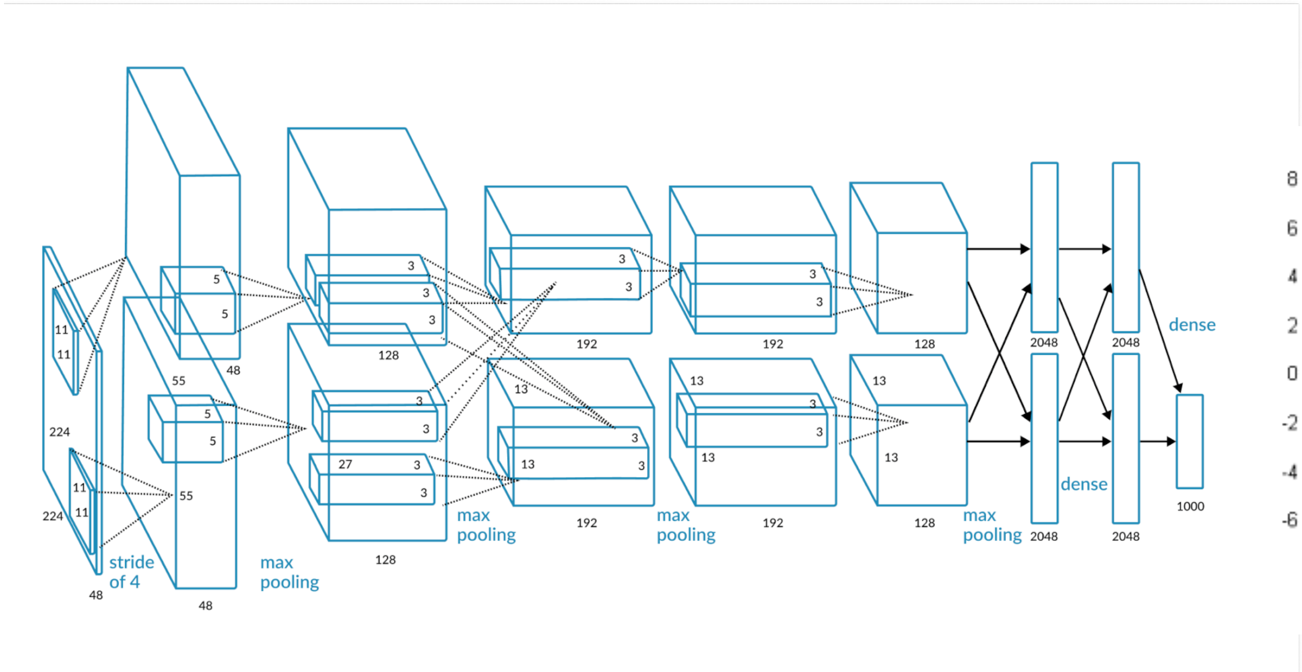


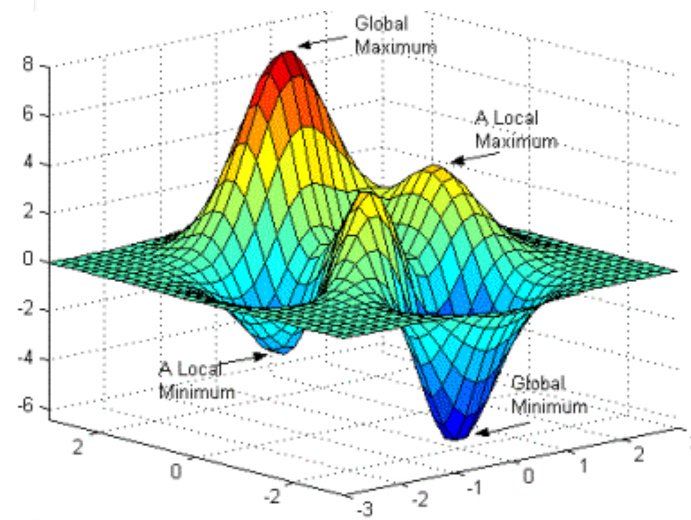
Generative Modeling (Generative Adversarial Networks)



Game (Reinforcement Learning, Policy NN)

Deep Neural Networks: Nonconvex Optimization





Alexnet: $\mathbf{x} \to f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}_L \circ \sigma\left(\ldots\sigma\left(\mathbf{w}_2 \circ \sigma(\mathbf{w}_1 \circ \mathbf{x})\right)\right)$

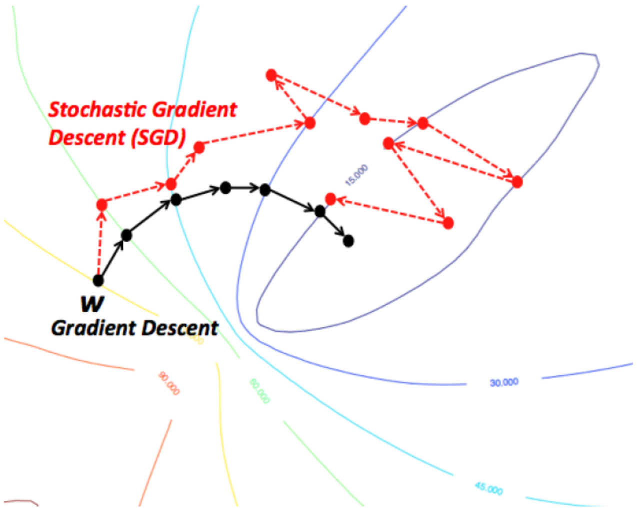
$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x},y}\left[\ell(f_{\mathbf{w}}(\mathbf{x}), y)\right]$$

The workhorse in Machine Learning

Stochastic Gradient Descent

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x},y}\left[\ell(f_{\mathbf{w}}(\mathbf{x}), y)\right]$$

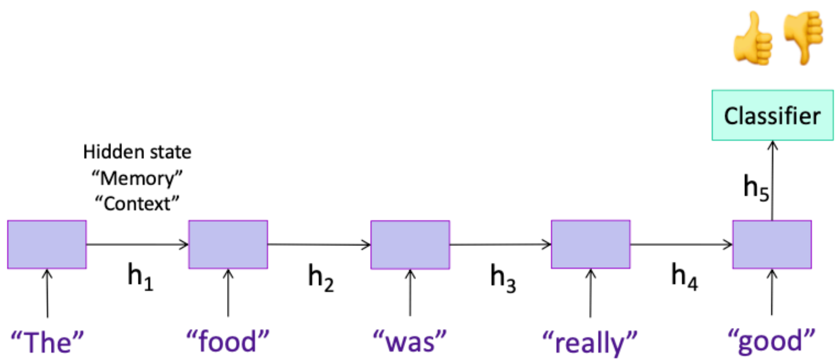
- Stochastic Gradient Descent (SGD) [Robbins-Monro'51]
  - Sample $(\mathbf{x}_t, y_t)$ uniformly
  - $\mathbf{w}_{t+1} = \mathbf{w}_t - \underbrace{\eta_t}_{\text{Learning rate}} \overbrace{\nabla \ell(\mathbf{w}_t, \mathbf{x}_t, y_t)}^{\text{Stochastic gradient}}$

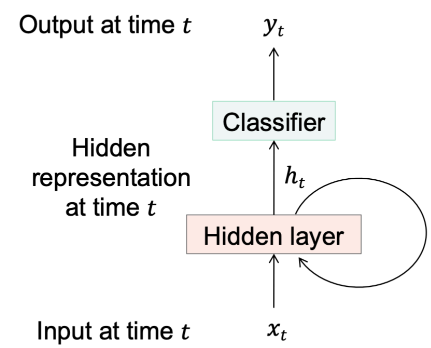


Assumptions in Optimization for Deep Learning

- Which assumption should we use for analyzing deep learning optimization such as SGD?
- We all like the "smoothness" assumption:
  - L-smooth function: $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$
  - In a smooth function,
    - Gradient goes to zero approaching to a local or global minimum, even if nonconvex
    - The function is upper bounded by a quadratic function
    - SGD can decrease the loss monotonically in expectation (a.k.a., descent lemma)

Gradient Explosion in Recurrent Neural Networks

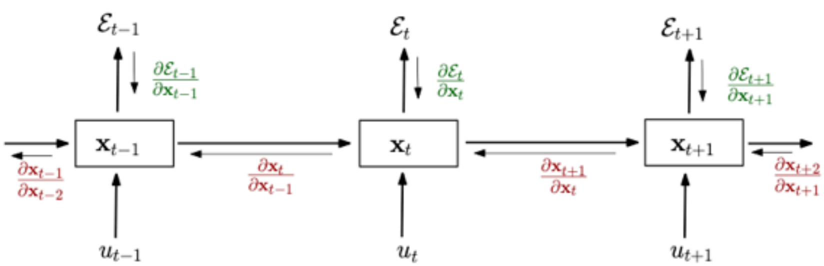




Recurrence:

$h_t = f_W(x_t, h_{t-1})$

new state | function of W | input at time t | old state



$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \geq i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \geq i > k} \mathbf{W}_{rec}^T diag(\sigma'(\mathbf{x}_{i-1}))$$

- Gradient will explode for long input if the recurrent matrix W has eigenvalue > 1
- The standard smoothness assumption does not hold

Unbounded Smoothness

- Smoothness is not satisfied in many cases
  - e.g., all univariate polynomials such as $x^4, \exp(x)$
- More importantly, [Zhang et al. ICLR'20] showed that deep neural networks have unbounded smoothness (e.g., gradient explosion)
- [Zhang et al. ICLR'20] introduced a weaker notion called "relaxed smoothness" or (L_0, L_1)-smoothness, and showed it holds for LSTMs
  - $\|\nabla^2 F(\mathbf{x})\| \leq L_0 + L_1 \|\nabla F(x)\|$

Transformers Satisfy Relaxed Smoothness

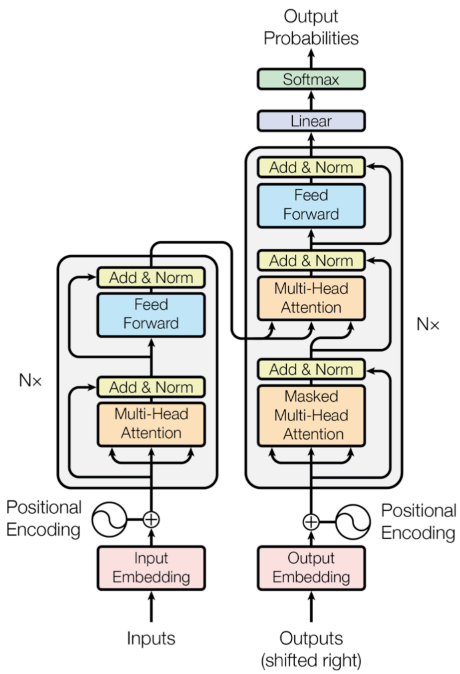
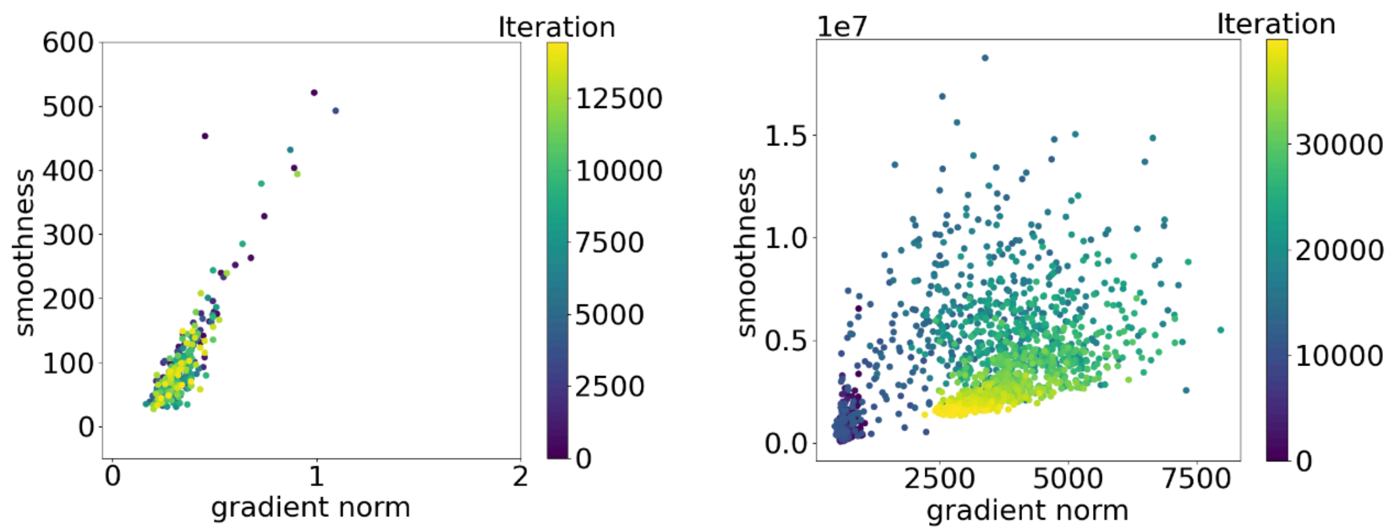


Figure 1: The Transformer - model architecture.



(a) Wikitext-2

(b) WMT'16 de-en

[Vaswani et al. NeurIPS'17]

We show that transformers satisfy relaxed smoothness

[Crawshaw-L.-Orabona-Zhang-Zhuang, NeurIPS'22]

SGD with Gradient Clipping under (L_0, L_1)-smoothness

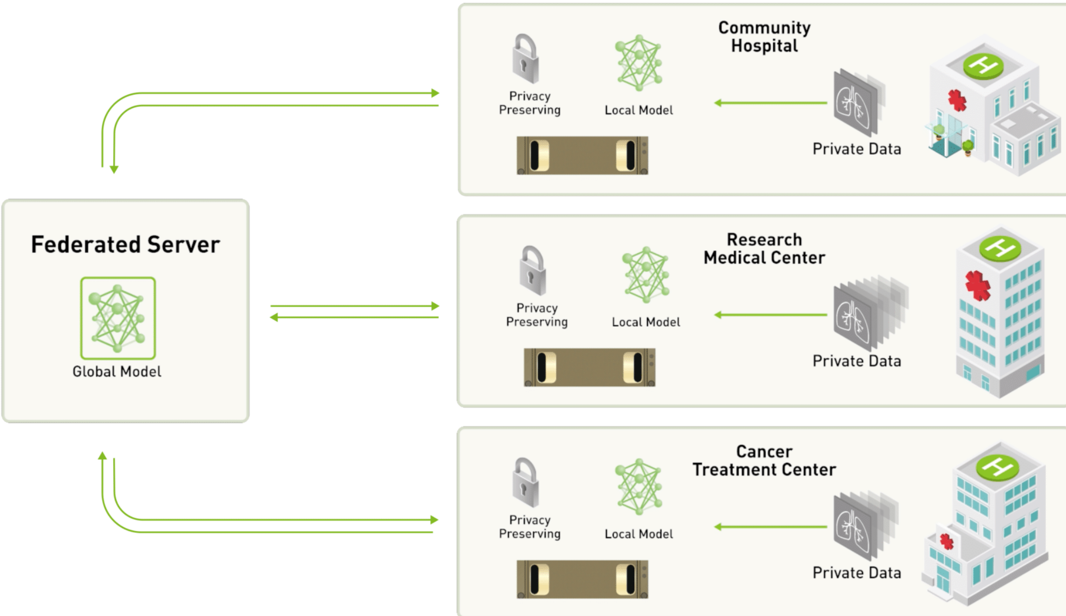
- Gradient clipping ensures SGD's convergence under (L_0, L_1)-smoothness [Zhang et al. ICLR'20]

Algorithm 1 Pseudo-code for norm clipping the gradients whenever they explode

$\hat{\mathbf{g}} \leftarrow \frac{\partial \mathcal{E}}{\partial \theta}$
if $\|\hat{\mathbf{g}}\| \geq \mathit{threshold}$ **then**
  $\hat{\mathbf{g}} \leftarrow \frac{\mathit{threshold}}{\|\hat{\mathbf{g}}\|} \hat{\mathbf{g}}$
end if

- Gradient clipping is necessary because relaxed smoothness can make the gradient exponentially large [Zhang et al. ICLR'20]
- But this algorithm is not scalable in large-scale federated deep learning

Motivation (Federated Learning)

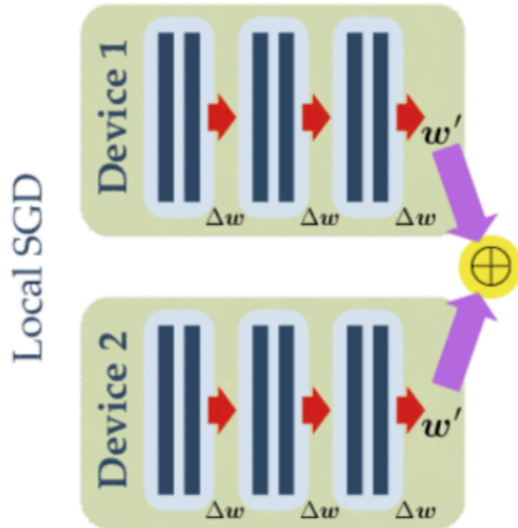


- Data is not shared
- Communication is expensive
- Data might not be i.i.d.
- Federated Learning (FL) [Mcmahan-Moore-Ramage-Hampson-Arcas, AISTATS'17]

How to design scalable algorithms in federated learning setting for relaxed smooth functions?

[L.-Zhuang-Lei-Liao, NeurIPS'22; Crawshaw-Bao-L., ICLR'23; Crawshaw-Bao-L., NeurIPS'23]

FedAvg (a.k.a., Local SGD)



Local SGD (FedAvg):
Multiple SGD updates on each
device before communication

Reduced Communication Cost: FedAvg is the default algorithm in FL, but only works for smooth problems

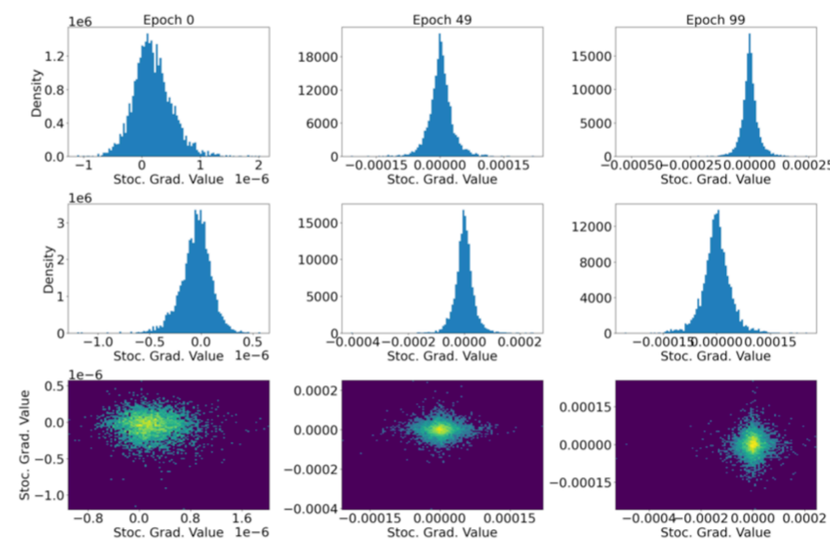
Q: How to design computation and communication-efficient algorithms for relaxed smooth problems such as RNN, LSTM, Transformers?

Communication-Efficient Federated Learning Algorithms for Relaxed Smooth Functions

Problem Setup (Homogeneous Data)

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\underbrace{\mathbf{x}}_{\text{Model Parameter}}) := \mathbb{E}_{\xi\sim\underbrace{\mathscr{D}}_{\text{Data Distribution}}}[F(\mathbf{x};\xi)]$$

- $f(\mathbf{x})$ is (L_0, L_1)-smooth:
  $\|\nabla^2 f(\mathbf{x})\| \le L_0 + L_1\|\nabla f(\mathbf{x})\|$ for any $\mathbf{x}\in\mathbb{R}^d$
- $f(\mathbf{x}_0) - \min_{\mathbf{x}} f(\mathbf{x}) \le \Delta$
- For all $\mathbf{x}\in\mathbb{R}^d$, $\mathbb{E}_{\xi\sim\mathscr{D}}\left[\nabla F(\mathbf{x};\xi)\right] = \nabla f(\mathbf{x})$,
  $\|\nabla F(\mathbf{x};\xi) - \nabla f(\mathbf{x})\| \le \sigma$ almost surely
- The stochastic gradient noise is unimodal and symmetric



Unimodal and Symmetric Noise in training LSTMs
[L.-Zhuang-Lei-Liao, NeurIPS 22]

Communication-Efficient Local Gradient Clipping

[L.-Zhuang-Lei-Liao, NeurIPS 22]

Algorithm 1 Communication Efficient Local Gradient Clipping (CELGC)

for $t = 0, \ldots, T$ **do**

  Each node i samples its stochastic gradient $\nabla F(\mathbf{x}_t^i; \xi_t^i)$, where $\xi_t^i \sim \mathcal{D}$.

  Each node i updates it local solution **in parallel**:

Local gradient clipping

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i - \min\left(\eta, \frac{\gamma}{\|\nabla F(\mathbf{x}_t^i; \xi_t^i)\|}\right) \nabla F(\mathbf{x}_t^i; \xi_t^i) \tag{2}$$

# of local steps

  if t is a multiple of I **then**

    Each worker resets the local solution as the averaged solution across nodes:

$$\mathbf{x}_t^i = \widehat{\mathbf{x}} := \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}_t^j \qquad \forall i \in \{1, \ldots, N\} \tag{3}$$

Periodically averages model every I steps

  end if

end for

Linear Speedup and Reduced Communication Complexity

- N: number of machines, σ: standard deviation in stochastic gradient
- Goal: finding ϵ-stationary point: an solution $\mathbf{x}$ such that $\|\nabla f(\mathbf{x})\| \leq \epsilon$

Theorem [L.-Zhuang-Lei-Liao, NeurIPS 22]

Choose $\gamma = O\left(\frac{N\epsilon^2}{\sigma L_0}\right), \eta = O\left(\frac{N\epsilon^2}{\sigma^2 L_0}\right), I = O\left(\frac{\sigma}{N\epsilon}\right)$.

Clipping Threshold

Learning Rate

# of local steps

To find ϵ-stationary point, the iteration complexity is $O\left(\frac{\Delta L_0 \sigma^2}{N\epsilon^4}\right)$, the

Reduced Communication Rounds

communication complexity is $O\left(\frac{\Delta L_0 \sigma}{\epsilon^3}\right)$

Linear Speedup

Analysis Roadmap

- At t-th iteration, define the indices of clients who perform clipping or not $J(t) = \{i \in [N] : \|\nabla F(\mathbf{x}_t^i; \xi_t^i)\| \geq \gamma/\eta\}, \bar{J}(t) = [N] \backslash J(t)$
- For either $i \in J(t)$ or $i \in \bar{J}(t)$, we show it decreases the loss function value sufficiently
- The local steps skip communication and introduce error, but the error can be controlled when the # of local steps is not extremely large
- Choose learning rate, clipping threshold, and # of local steps, we get linear speedup (because we are using N machines) and reduced communication rounds (due to the local steps)

Technical Challenges and Solutions

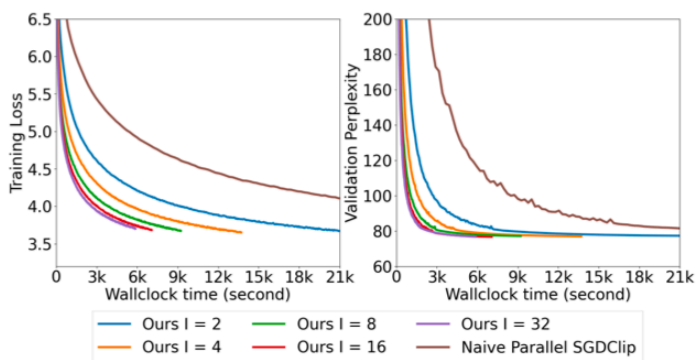
- The analysis roadmap looks so easy, but there are certain challenges:
  - Difficulty 1: The standard descent lemma in smooth case does not work
  - Solution: We introduce a new descent lemma in relaxed smooth setting and amenable to local steps
    - If the learning rate is small, the loss function monotonically decreases when synchronization occurs, even if the landscape is not smooth
    - Local steps do not hurt too much

Technical Challenges and Solutions

- Difficulty 2: The stochastic gradient for the non-clipping client is $\nabla F(\mathbf{x}_t^i; \xi_t^i)\mathbb{I}(\|F(\mathbf{x}_t^i; \xi_t^i)\| \leq \alpha)$, which may not follow the right direction due to the dependency between random variables
- Consider the following example (g: stochastic gradient):
  - $Pr(g = 2) = 0.2, Pr(g = -2) = 0.3, Pr(g = 3) = 0.5, \alpha = 2$
  - $\mathbb{E}\left[g \cdot \mathbb{I}(\ \ g\ \ \leq \alpha)\right] = -0.2$, but $\mathbb{E}\left[g\right] = 1.3$, different directions 😞
- Solution: the distributional assumption (unimodal and symmetric noise)
  - a new Lemma to show that the expectation of stochastic gradient in the non-clipping client aligns with the true gradient up to a constant factor: $\mathbb{E}\left[g\mathbb{I}(\|g\| \leq \alpha)\right] = \text{Pr}(\|g\| \leq \alpha)\Lambda\mathbb{E}[g], \Lambda = \text{diag}(c_1, \ldots, c_d), 0 < c_i \leq 1$ 😁

Experiments

- Train deep neural networks on 8 V100 GPUs
- Consider two tasks: language modeling and image classification
- Compare our algorithm with different I versus the naive parallel algorithm



Language Modeling on WikiText-2 on AWD-LSTM

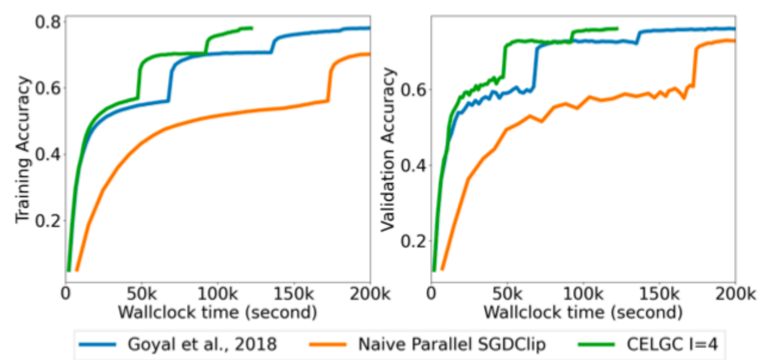
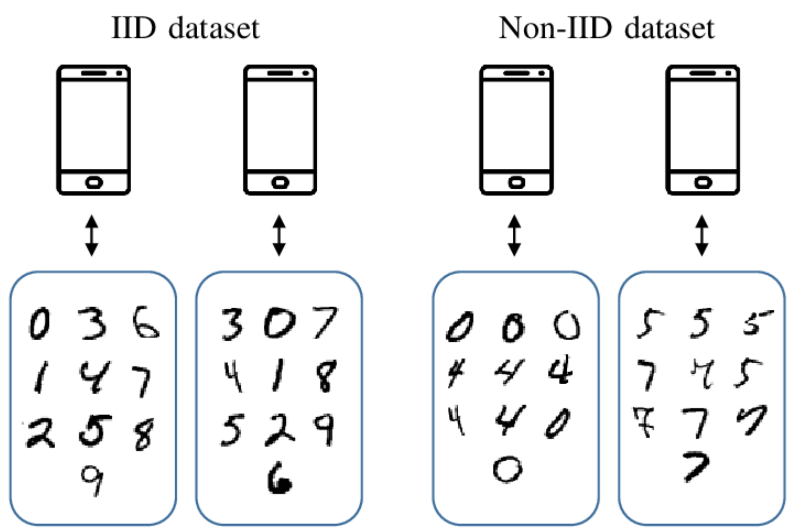


Image Classification on ImageNet with ResNet

Gradient clipping with local steps does not hurt the convergence, instead accelerates the training!

From i.i.d. Data to Non-i.i.d. Data



Does Local Gradient Clipping Work for Heterogeneous Data?

- The different client has different data distribution $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[F(\mathbf{x};\xi_i)]$
- The Local Gradient Clipping Algorithm does not work:
  - Consider the two clients case: $f_1(x) = \frac{1}{2}x^2 + a_1 x, \quad f_2(x) = \frac{1}{2}x^2 + a_2 x$
  - $a_1 = -\gamma - 1, \quad a_2 = \gamma + 2, \quad \gamma > 1,$ γ is the clipping threshold
  - Optimal solution is $x_* = -\frac{a_1 + a_2}{2} = -\frac{1}{2}$
  - Start from 0, run the local gradient clipping with learning rate 1 on each client for 1 iteration: the algorithm gets γ and $-\gamma$ on two clients respectively, the averaged model parameter becomes 0 again (the algorithm gets stuck!)

EPISODE (for Heterogeneous Data)

Algorithm 1: Episodic Gradient Clipping with Periodic Resampled Corrections (EPISODE)

Initialize $\boldsymbol{x}_0^i \leftarrow \boldsymbol{x}_0$, $\bar{\boldsymbol{x}}_0 \leftarrow \boldsymbol{x}_0$.
for $r = 0, 1, \ldots, R$ **do**
  for $i \in [N]$ **do**
    Sample $\nabla F_i(\bar{\boldsymbol{x}}_r; \widetilde{\xi}_r^i)$ where $\widetilde{\xi}_r^i \sim \mathcal{D}_i$, and update $\boldsymbol{G}_r^i \leftarrow \nabla F_i(\bar{\boldsymbol{x}}_r; \widetilde{\xi}_r^i)$.
  end for
  Update $\boldsymbol{G}_r = \frac{1}{N}\sum_{i=1}^N \boldsymbol{G}_r^i$.
  for $i \in [N]$ **do**
    for $t = t_r, \ldots, t_{r+1} - 1$ **do**
      Sample $\nabla F_i(\boldsymbol{x}_t^i; \xi_t^i)$, where $\xi_t^i \sim \mathcal{D}_i$, and compute $\boldsymbol{g}_t^i \leftarrow \nabla F_i(\boldsymbol{x}_t^i; \xi_t^i) - \boldsymbol{G}_r^i + \boldsymbol{G}_r$.
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i - \eta \boldsymbol{g}_t^i \mathbb{1}(\|\boldsymbol{G}_r\| \le \gamma/\eta) - \gamma \frac{\boldsymbol{g}_t^i}{\|\boldsymbol{g}_t^i\|} \mathbb{1}(\|\boldsymbol{G}_r\| \ge \gamma/\eta)$.
    end for
  end for
  Update $\bar{\boldsymbol{x}}_r \leftarrow \frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_{t_{r+1}}^i$.
end for

Indicator for episodic gradient clipping

Periodic Resampled Correction

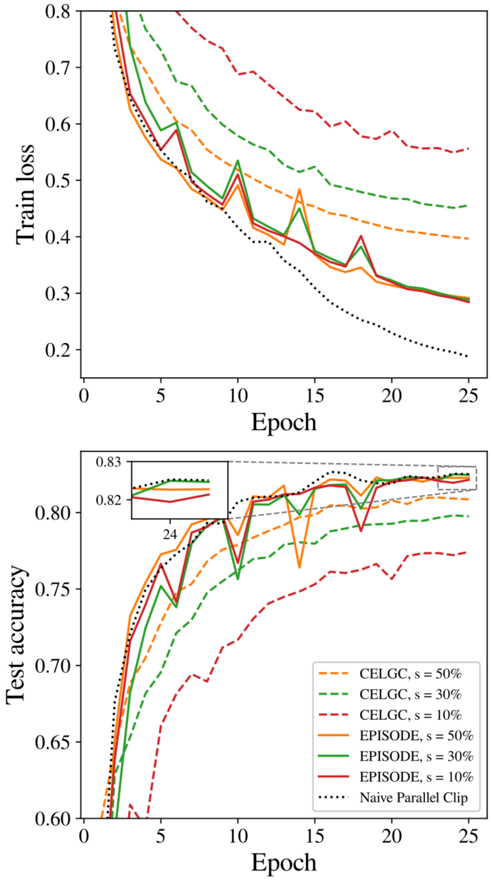
Data Heterogeneity

Theorem [Crawshaw-Bao-L.,ICLR 23] : EPISODE has iteration complexity $O\left(\frac{\Delta L_0 \sigma^2}{N\epsilon^4}\right)$, communication complexity is $O\left(\frac{\Delta L_0 + L_1(\kappa + \sigma)\sigma}{\epsilon^3}\right)$

Proof Technique Overview

- New localization Lemma:
  - In each communication round, the iterates of EPISODE stay in a bounded region almost surely, where the function is locally L-smooth
  - The radius of the bounded region does not depend on the data heterogeneity (κ), this is the key to show that the iteration complexity does not depend on κ
  - Each communication rounds the function value will decrease sufficiently

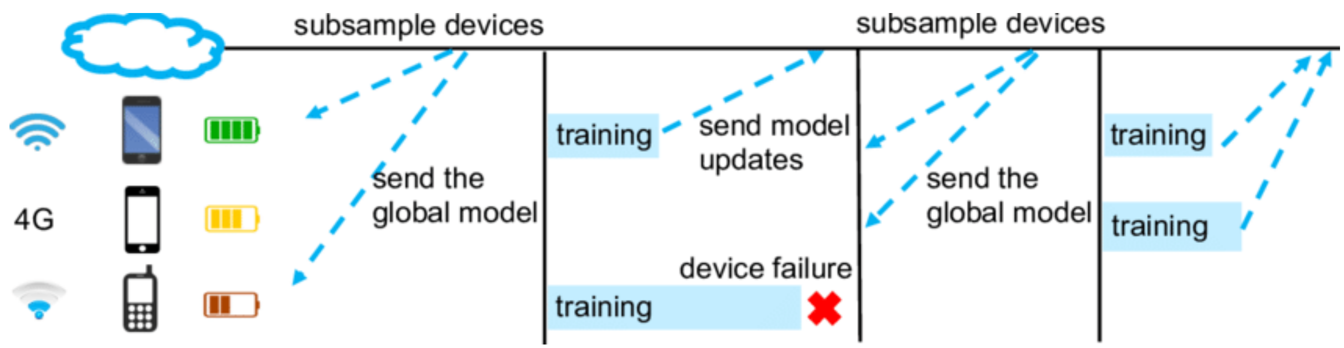
Experiments



(c) Effect of s (Epochs)

- Train a recurrent neural network on SNLI dataset (text classification) on eight GPUs
- Heterogeneous data: larger similarity (s) indicates smaller heterogeneity
- EPISODE does not suffer from high heterogeneity, while local gradient clipping (CELGC) suffers from data heterogeneity significantly

From Full Client Participation to Partial Client Participation



EPISODE++ Algorithm for Partial Client Participation

Algorithm 1 EPISODE++

Initialize $\bar{\boldsymbol{x}}_0$, $\boldsymbol{G}_0^i \leftarrow \nabla F_i(\bar{\boldsymbol{x}}_0, \tilde{\xi}_i)$, $\boldsymbol{G}_0 \leftarrow \frac{1}{N}\sum_{i=1}^N \boldsymbol{G}_0^i$
for $r = 0, 1, \dots, R-1$ **do**
  Sample $\mathcal{S}_r \subset [N]$ uniformly at random such that $|\mathcal{S}_r| = S$
  for $i \in \mathcal{S}_r$ **do**
    $\boldsymbol{x}_{r,0}^i \leftarrow \bar{\boldsymbol{x}}_r$
    for $k = 0, \dots, I-1$ **do**
      Sample $\nabla F_i(\boldsymbol{x}_{r,k}^i; \xi_{r,k}^i)$, where $\xi_{r,k}^i \sim \mathcal{D}_i$
      $\boldsymbol{g}_{r,k}^i \leftarrow \nabla F_i(\boldsymbol{x}_{r,k}^i; \xi_{r,k}^i) - \boldsymbol{G}_r^i + \boldsymbol{G}_r$
      $\boldsymbol{x}_{r,k+1}^i \leftarrow \boldsymbol{x}_{r,k}^i - \eta \boldsymbol{g}_{r,k}^i \mathbb{1}_{\|\boldsymbol{G}_r\| \le \gamma/\eta} - \gamma \frac{\boldsymbol{g}_{r,k}^i}{\|\boldsymbol{g}_{r,k}^i\|} \mathbb{1}_{\|\boldsymbol{G}_r\| \ge \gamma/\eta}$
    end for
    $\boldsymbol{G}_{r+1}^i \leftarrow \frac{1}{I}\sum_{k=0}^{I-1} \nabla F_i(\boldsymbol{x}_{r,k}^i; \xi_{r,k}^i)$
    $\Delta \boldsymbol{G}_r^i \leftarrow \boldsymbol{G}_{r+1}^i - \boldsymbol{G}_r^i$
  end for
  Update $\bar{\boldsymbol{x}}_{r+1} \leftarrow \frac{1}{S}\sum_{i \in \mathcal{S}_r} \boldsymbol{x}_{r,I}^i$
  Update $\boldsymbol{G}_{r+1} \leftarrow \boldsymbol{G}_r + \frac{1}{N}\sum_{i \in \mathcal{S}_r} \Delta \boldsymbol{G}_r^i$
  Denote $\boldsymbol{G}_{r+1}^i \leftarrow \boldsymbol{G}_r^i$ for all $i \notin \mathcal{S}_r$ $\boxed{1}$
end for

Data Heterogeneity

Theorem [Crawshaw-Bao-L.,NeurIPS 23] : EPISODE++ has iteration complexity

Number of subsampled clients

$O\left(\frac{\Delta L_0 \sigma^2}{S\epsilon^4}\right)$, communication complexity is $O\left(\frac{\Delta L_0 + L_1(\kappa+\sigma)\sigma}{\epsilon^3}\right)$

Provable Advantage over Clipped Minibatch SGD

- Baseline: Minibatch SGD (no local update, just local accumulation of batch size with one update before communication)
- It is proved by [Woodworth et al.'20] that minibatch SGD is always better than local SGD for heterogeneous data and full client participation
- In federated learning, we only have partial client participation
- We show that clipped minibatch SGD could be worse than EPISODE++ in the presence of partial client participation and unbounded smoothness

Hardness Results of Clipped Minibatch SGD

Algorithm 2 Clipped Minibatch SGD

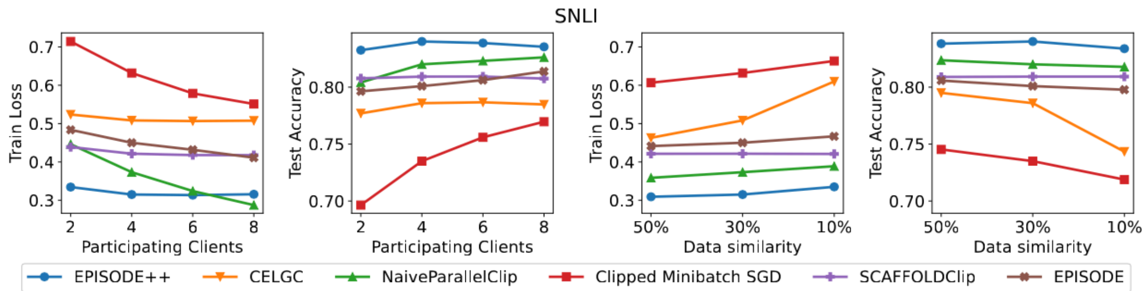
1: Initialize $\boldsymbol{x}_0$
2: **for** $r = 0, 1, \ldots, R-1$ **do**
3: Sample $\mathcal{S}_r \subset [N]$ uniformly at random such that $|\mathcal{S}_r| = S$
4: $\boldsymbol{g}_r = \frac{1}{SI} \sum_{i \in \mathcal{S}_r} \sum_{k=0}^{I-1} \nabla F_i(\boldsymbol{x}_r, \xi_{r,k}^i)$
5: Update $\boldsymbol{x}_{r+1} \leftarrow \boldsymbol{x}_r - \min\left(\eta, \frac{\gamma}{\|\boldsymbol{g}_r\|}\right) \boldsymbol{g}_r$
6: **end for**

Theorem [Crawshaw-Bao-L.,NeurIPS 23] : Fix $\epsilon > 0$, $L_0 > 0$, $L_1 > 0$, $M > \max(L_0/L_1, \epsilon)$, and $x_0 \in \mathbb{R}$. Pick any constant learning rate η and threshold γ based on the knowledge of above constants. There exists a function instance $\{f_i\}_{i=1}^N$ such as clipped minibatch SGD initialized at x_0 has communication complexity $\Omega\left(\frac{\Delta M L_1}{\epsilon^2}\right)$ with high probability, where M is the gradient upper bound (may be very large, e.g., exploding gradient)

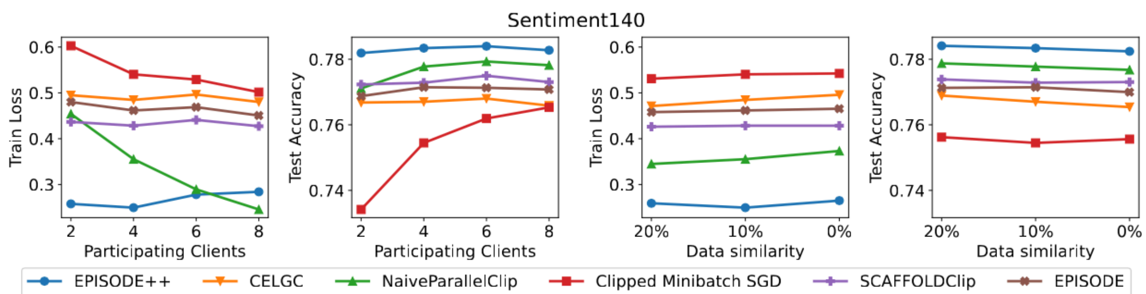
Proof Sketch of the Lower Bounds

- We analyze clipped minibatch SGD for three problem instances.
  - For linear objective function with high heterogeneity: if the clipping threshold is small (i.e., $\gamma/\eta \leq M$), then the clipped minibatch SGD will never converge with probability δ
  - For homogeneous exponential local objective, clipped minibatch SGD cannot converge if the learning rate is not sufficiently small (i.e., $\eta \geq 1/L_1 M$)
  - For a large clipping threshold (i.e., $\gamma/\eta \geq M$) and small learning rate (i.e., $\eta \leq 1/L_1 M$), the convergence rate of clipped minibatch SGD will depend on M for the third problem instance with homogeneous linear objectives

Experiments



(a) Training loss and testing accuracy for SNLI dataset.

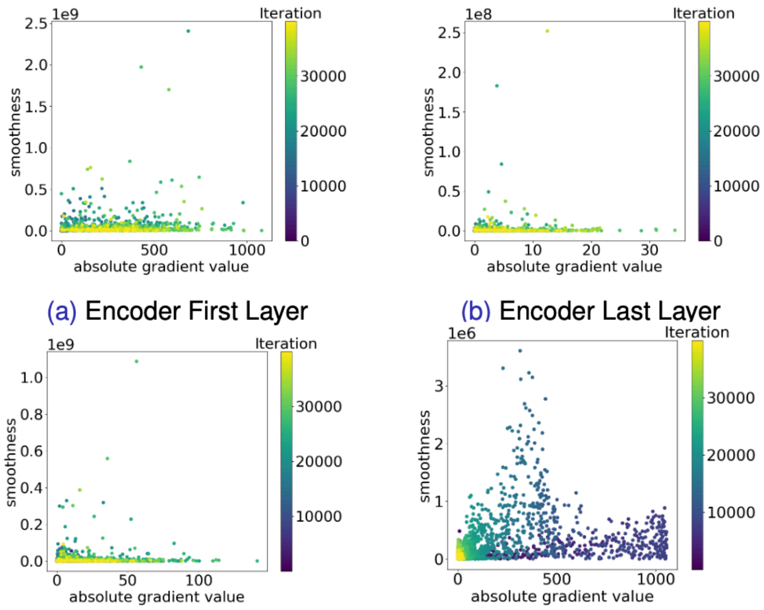


(b) Training loss and testing accuracy for Sentiment140 dataset.

Figure 1: Final training loss and testing accuracy for all algorithms, as participation ratio and data similarity varies. (a) and (b) show results for SNLI and Sentiment140, respectively.

An Adaptive Gradient Algorithm for Layer-Wise Relaxed Smooth Functions

Layer-wise Relaxed Smoothness in Transformer



(a) Encoder First Layer (b) Encoder Last Layer

(c) Decoder Second Layer (d) Decoder Last Layer

WMT'16 de-en

Relaxed smoothness parameters differ from layer to layer [Crawshaw-L.-Orabona-Zhang-Zhuang, NeurIPS'22]

Q: How to formally define layer-wise relaxed smoothness? Why people use Adam for training Transformers? Can we take advantage of this assumption to design better adaptive algorithms for training Transformers?

Coordinate-wise Relaxed Smoothness

- Let $\boldsymbol{L}_0 := [L_{0,1}, \ldots, L_{0,d}]^\top$ and $\boldsymbol{L}_1 := [L_{1,1}, \ldots, L_{1,d}]^\top$, A differentiable function $F(\mathbf{x})$ is $(\boldsymbol{L}_0, \boldsymbol{L}_1)$-smooth coordinate-wisely, if for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ such that $\|\mathbf{x} - \mathbf{y}\| \leq \frac{1}{\|\mathbf{L}_1\|_\infty}$, we have

$$\left| \frac{\partial F}{\partial x_j}(\mathbf{y}) - \frac{\partial F}{\partial x_j}(\mathbf{x}) \right| \leq \left(\frac{L_{0,j}}{\sqrt{d}} + L_{1,j} \left| \frac{\partial F}{\partial x_j}(\mathbf{x}) \right| \right) \|\mathbf{y} - \mathbf{x}\|_2, \ \forall j \in [d]$$

- When $L_{0,j} = L_0$ and $L_{1,j} = L_1$ for all $j \in [d]$, we recover the normal version of this assumption
- Can we analyze modern coordinate-wise algorithms with this assumption?
- Do we need gradient clipping?

Adam Algorithm (Coordinate-Wise Update)

Adam: A **method** for **stochastic optimization** [PDF] arxiv.org

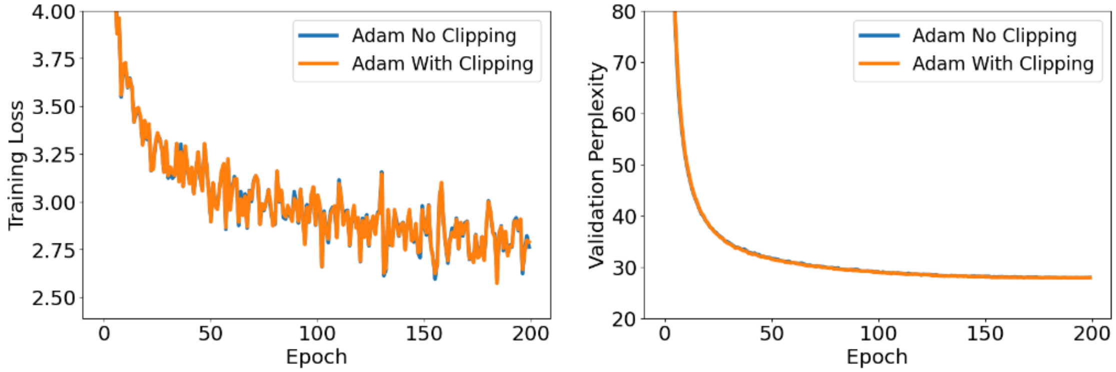
DP Kingma, J Ba - arXiv preprint arXiv:1412.6980, 2014 - arxiv.org

… to first-order **methods**. We propose **Adam**, a **method** for efficient **stochastic optimization** that only … The **method** computes individual adaptive learning rates for different parameters from …

☆ Save 99 Cite Cited by 123807 Related articles All 25 versions »

$m_0 \leftarrow 0$ (Initialize 1^{st} moment vector)
$v_0 \leftarrow 0$ (Initialize 2^{nd} moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
  $t \leftarrow t + 1$
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
  $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
  $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
  $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
end while

Gradient Clipping Might Be Implicit in Adam-type Algorithms



- Adam optimizer with and without gradient clipping
- Train a 16-layer GPT-2 transformer model to do language modeling (word level) in the Wikitext-103 dataset
  - Minibatch size is 256, learning rate warmup and cosine annealing
- Adam has almost a bounded update and clipping seems not necessary

A New Adam-type Algorithm (Generalized SignSGD)

Algorithm Generalized SignSGD
(All operations on vectors are element-wise)

Inputs: $\boldsymbol{x}_1$, β_1, β_2, η
$\boldsymbol{m}_0 = 0$, $\boldsymbol{v}_0 = 0$
for $t = 1, \cdots, T$ **do**
  Compute $\boldsymbol{g}_t$, an unbiased estimate of $\nabla F(\boldsymbol{x}_t)$
  $\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$
  $\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{m}_t^2$
  $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t}}$
end for

Difference with Adam: $\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$

Theoretical Convergence Guarantee (I)

Theorem [Crawshaw-L.-Orabona-Zhang-Zhuang, NeurIPS 22]

Run generalized SGD algorithm for T iterations, there exists setting for η, β_1, β_2 such that with high probability,

$$\min_{t \in [T]} \|\nabla F(\mathbf{x}_t)\|_1 \leq \tilde{O}\left(\frac{\|\sigma\|_1}{T^{1/4}} + \frac{1}{\sqrt{T}}\right) + \tilde{O}\left((\|\mathbf{M}\|_1 + \|\sigma\|_1)\exp(-T^{1/4})\right),$$

where $\mathbf{M}_j = \sup\left\{\left|\frac{\partial F}{\partial x_j}(\mathbf{x})\right| : F(\mathbf{x}) \leq F(\mathbf{x}_0)\right\} < +\infty$

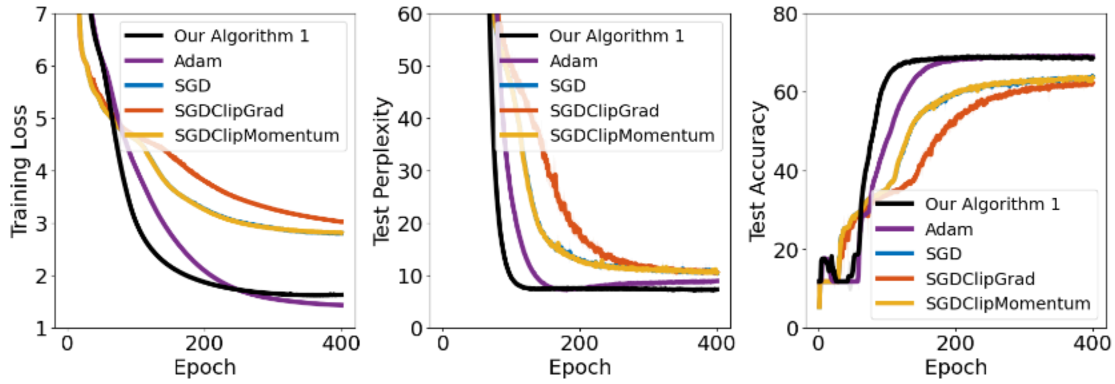
Theoretical Convergence Guarantee (II)

Theorem [Crawshaw-L.-Orabona-Zhang-Zhuang, NeurIPS 22]

Run generalized SGD with $\beta_2 = 0$ for T iterations, we have with high probability

$$\min_{t \in [T]} \|\nabla F(\mathbf{x}_t)\|_1 \leq \tilde{O}\left(\frac{\|\sigma\|_1}{T^{1/4}} + \frac{1}{\sqrt{T}} \right)$$

Transformer on Translation Task



- Train a 6-layer Transformer on WMT'16 German-English Translation Task
  - Mini-batch size is 256
  - Learning rate warm-up and decay
  - Training+testing with best hyperparameters repeated 5 times with different random seeds

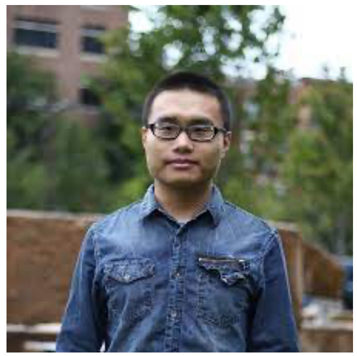
Summary

- Relaxed Smoothness condition in deep learning is widely-used
- Communication-efficient federated learning algorithm for relaxed smooth functions with homogeneous and heterogeneous data [L.-Zhuang-Lei-Liao, NeurIPS'22, Crawshaw-Bao-L., ICLR'23]
- New algorithms for partial client participation in federated learning for relaxed smooth functions and lower bounds [Crawshaw-Bao-L., NeurIPS'23]
- An Adam-type algorithm (generalized signSGD) for relaxed smooth functions which is competitive to best-tuned Adam [Crawshaw-L.-Zhang-Orabona-Zhuang, NeurIPS'22]

Acknowledgments

Michael Crawshaw
PhD Student
CS@George Mason

Yajie Bao
PhD Student
Statistics@SJTU



Zhenxun Zhuang
Research Scientist
Meta Platforms

Wei Zhang
Researcher
IBM T. J. Watson

Francesco Orabona
Associate Professor
ECE@ Boston University

Yunwen Lei
Assistant Professor
Univ of Hong Kong

Thank you for your attention!