The Implicit Bias for Adaptive Optimization Algorithms on Homogeneous Neural Networks

Bohan Wang, Qi Meng, Wei Chen, Tie-Yan Liu

Microsoft Research Asia

{v-bohanwang, meq, wche, tie-yan.liu}@microsoft.com

Contents

- **Background**
  - Implicit bias
  - Adaptive Optimizers
  - Related work
- Main Results
- Numerical Experiments

Implicit Bias

- Deep neural networks usually generalize well despite most of their local minima generalize poorly.

- Implicit bias is one plausible explanation, the intuition of which is optimization algorithms implicitly regularize the training process and find the minimum which generalizes well.

Implicit Bias

- There are different interpretations for implicit bias:
  - (Indirectly) the escaping rate from saddle point, flat minima
  - (Directly) the convergent point in L^2 regression
  - (Directly) the convergent direction in logistic regression (This paper)

- It is a standard practice to study the form of convergent direction in logistic regression for homogeneous neural networks.

Adaptive Optimizers

- Adaptive optimizers are a series of gradient-based optimizers which utilize the historical gradient information to adjust the learning rate component-wisely.

- The general update rule:

$$w(t+1) - w(t) = -h(t) \odot \nabla\mathcal{L}(w(t))$$

  - $h(t)$ is the conditioner
  - $\nabla\mathcal{L}$ is the gradient empirical loss
  - $\odot$ is the component wise multiplication (Hadamard product)

Adaptive Optimizers

- They have been shown (empirically) to achieve faster convergent rate than vanilla GD/SGD, but (sometimes) worse generalization performance

- The implicit bias for adaptive optimizers?

Microsoft

Related Work:

- The implicit bias of gradient descent (GD) has been well studied.
  - Lyu & Li (2019) shows that for logistic regression task, GD on homogeneous neural networks drives the parameter towards the direction of some KKT point of the corresponding L^2 max-margin problem:

$$\min \ \|w\|^2 \ \ s.t. \ y_\mathrm{i}\Phi(w, x_i) \geq 1 \quad \forall \, i \in [N]$$

  - Ali et al. (2020) shows that for linear L^2 regression using SGLD with SGD noise covariance, the parameter at time t is close to the ridge regression estimate with tuning parameter $\frac{1}{t}$.

Related Work

- (Qian & Qian, 2019) proves the convergent direction of AdaGrad on **linear** logistic regression.

- There is little theoretical analysis on the generalization performance of adaptive optimizers, especially in the **non-linear** logistic case or from the viewpoint of implicit bias.

Contents

- Background
- **Main Results**
  - Problem setups
  - Main results
  - Discussions
  - Proof sketch
- Numerical Experiments

Problem Setups

- Let $\{(x_1, y_1), \cdots, (x_N, y_N)\}$ be the sample set. Let $\Phi(w, x)$ be the output(prediction) of neural network Φ with parameter w and data x.

- We use $w(t)$ as the parameter at time t.

- We use Clarke's sub-gradient $\bar{\partial}$.

- We focus on logistic regression with loss $\ell = \ell_{exp}$ and $\ell = \ell_{log}$. Given sample $\{(x_i, y_i)\}_{i=1}^N$, the empirical loss for parameter w is defined as $\mathcal{L}(w) = \sum_{i=1}^N \ell(y_i \Phi(w, x_i))$.

Adaptive Optimizers (discrete form)

- The discrete form of the optimizers:

$$w(t+1) - w(t) = -h(t) \odot \bar{\partial}\mathcal{L}(w(t))$$

For AdaGrad, $h(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_{\boldsymbol{p}} + \sum_{\tau=0}^{t} \bar{\partial}\mathcal{L}\big(w(\tau)\big)^2}$.

For RMSProp, $h(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_{\boldsymbol{p}} + \sum_{\tau=0}^{t} (1-b) e^{-(1-b)(t-\tau)} \bar{\partial}\mathcal{L}\big(w(\tau)\big)^2}$.

For Adam (w/m), $h(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_{\boldsymbol{p}} + \frac{\sum_{\tau=0}^{t} (1-b) e^{-(1-b)(t-\tau)} \bar{\partial}\mathcal{L}\big(w(\tau)\big)^2}{1-b^t}}$.

For any optimizer, $h_\infty = \lim_{t \to \infty} h(t)$.

ε is a constant added to avoid the conditioner being zero.

Adaptive Optimizers (continuous form)

- The continuous form of the optimizers:

$$\frac{dw(t)}{dt} = -h(t) \odot \bar{\partial}\mathcal{L}(w(t))$$

For AdaGrad, $h(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_{\boldsymbol{p}} + \int_0^t \bar{\partial}\mathcal{L}\big(w(\tau)\big)^2 d\tau}.$

For RMSProp, $h(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_{\boldsymbol{p}} + \int_0^t (1-b)e^{-(1-b)(t-\tau)}\bar{\partial}\mathcal{L}\big(w(\tau)\big)^2 d\tau}.$

For Adam (w/m), $h(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_{\boldsymbol{p}} + \frac{\int_0^t (1-b)e^{-(1-b)(t-\tau)}\bar{\partial}\mathcal{L}\big(w(\tau)\big)^2 d\tau}{1-b^t}}.$

For any optimizer, $h_\infty = \lim_{t\to\infty} h(t).$

Assumptions

- We need several mild assumptions:
  - For continuous case:
    - The neural network is locally Lipschitz with respect to parameter
    - The neural network is homogenous
    - There exists a time when NN achieves correct classification
  - For discrete case, two additional assumption are needed:
    - The neural network is M smooth with respect to the parameter
    - The learning rate is upper bounded and lower bounded.

Main Theorem

Theorem: Under the assumptions, (1) for AdaGrad(continuous /discrete), any limit point ($t \to \infty$) of parameter direction $w_t / \|w_t\|_2$ is a KKT point of the following optimization problem:

$$\min \left\| h_\infty^{-1/2} \odot w \right\|^2 \quad s.t.\ y_i \Phi(w, x_i) \geq 1 \qquad \forall\, i \in [N];$$

(2) for RMSProp and Adam without momentum(continuous/discrete), the direction is a KKT point of

$$\min \|w\|^2 \ \ s.t.\ \mathrm{y}_i \Phi(w, x_i) \geq 1 \qquad \forall\, i \in [N].$$

Discussions

- Our results shows RMSProp, Adam (w/m) and GD share similar generalization property in terms of margin, while AdaGrad has worse performance and sensitive to initialization

- The exponential weighted design in the conditioner and ε accelerate the training process before convergence, and still lead to the max-margin solution.

Extensions

- A simple modification of the proof can lead to the results of multi-class classification, where only the constraints $y_i\Phi(w, x_i) \geq 1$ are changed into $\left(\Phi(w, x_i)\right)_{y_i} - \left(\Phi(w, x_i)\right)_j \geq 1$.

- While there is not necessarily only one limit point, the definability condition (used in (Ji & Telgarsky, 2020)) can ensure this.

Proof Sketch

Adaptive Gradient Flow (AGF)

$$\frac{dv(t)}{dt} = -\beta(t) \odot \bar{\partial}\mathcal{L}(v(t)) \text{ with}$$

- $\lim_{t\to\infty} \beta(t) = \mathbf{1}_p$
- $\frac{d \log(\beta(t))}{dt}$ is Lebesgue Integrable

Proof Sketch

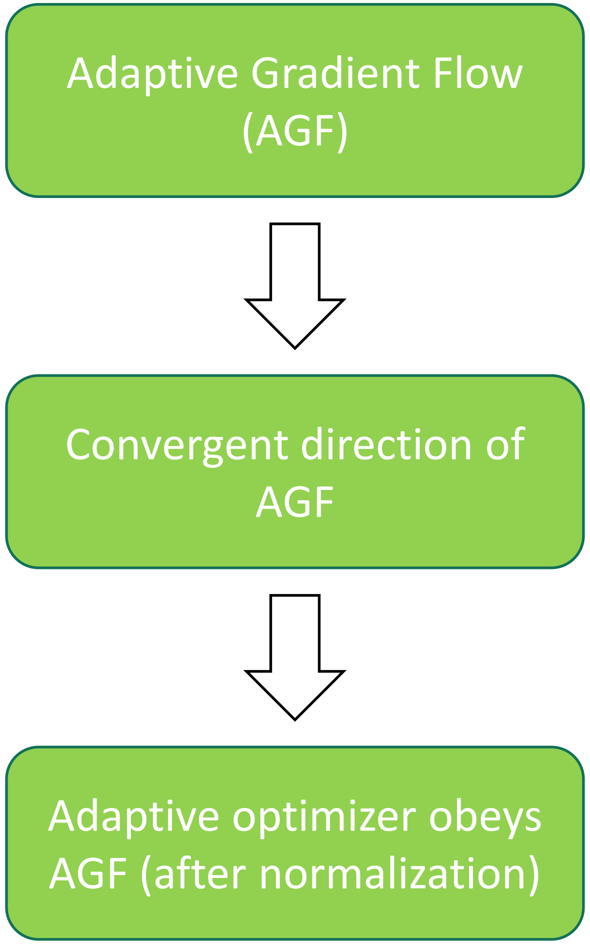
Adaptive Gradient Flow (AGF)

↓

Convergent direction of AGF

- Define surrogate margin as $\tilde{\gamma}(t) = \frac{\ell^{-1}(\mathcal{L}(v(t)))}{\left\|\beta(t)^{-\frac{1}{2}} \odot v(t)\right\|^{L}}$;
- Prove surrogate margin is lower bounded;
- Use surrogate margin to bound derivatives and prove loss converges to zero;
- Prove for every limit point of parameter direction $\bar{v}^*$, there exists a sequence $v(t_i)$ converges to $\bar{v}^*$, with $v(t_i)$ satisfies $(\varepsilon_i, \delta_i)$ approximately KKT condition, $\lim_{i\to\infty} \varepsilon_i = 0$, and $\lim_{i\to\infty} \delta_i = 0$;
- By Mangasarian-Fromovitz constraint qualification, $\bar{v}^*$ is then a KKT point.

Proof Sketch



- For AdaGrad, $\mathrm{h}_\infty \equiv \lim_{t\to\infty} \frac{1}{\sqrt{1+m(t)}}$ exists and is non-zero, while for RMSProp and Adam (w/m), $h_\infty = \frac{1}{\sqrt{\varepsilon}}\mathbf{1}_p$. Therefore, $v(t) \equiv h_\infty^{-1/2} \odot w(t)$ is well defined.

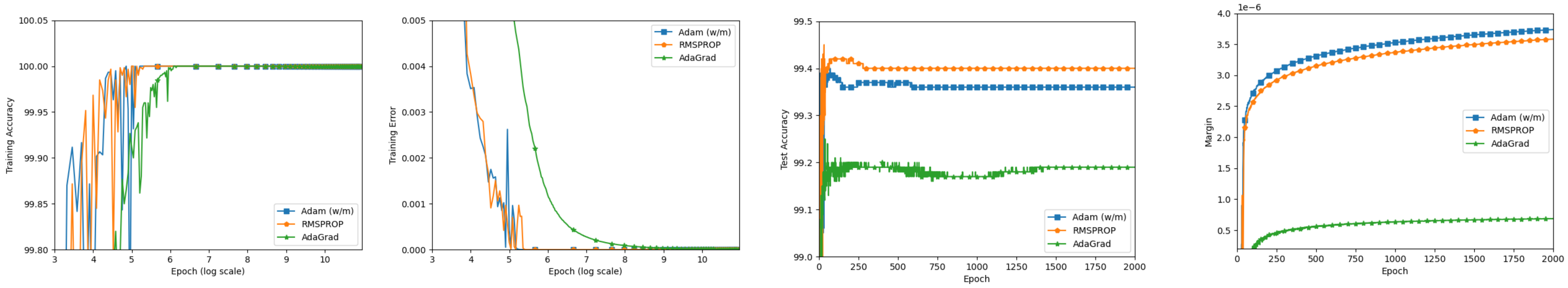
- $v(t)$ obeys adaptive gradient flow by a key observation that $\int_0^\infty \bar{\partial}\mathcal{L}\left(w(t)\right)^2\ dt < \infty$.

Contents

- Background
- Theoretical Results
- **Numerical Experiments**
  - Observation of Margin
  - Directions of h_∞
  - Effect of ε on the generalization performance

Observation of Margin

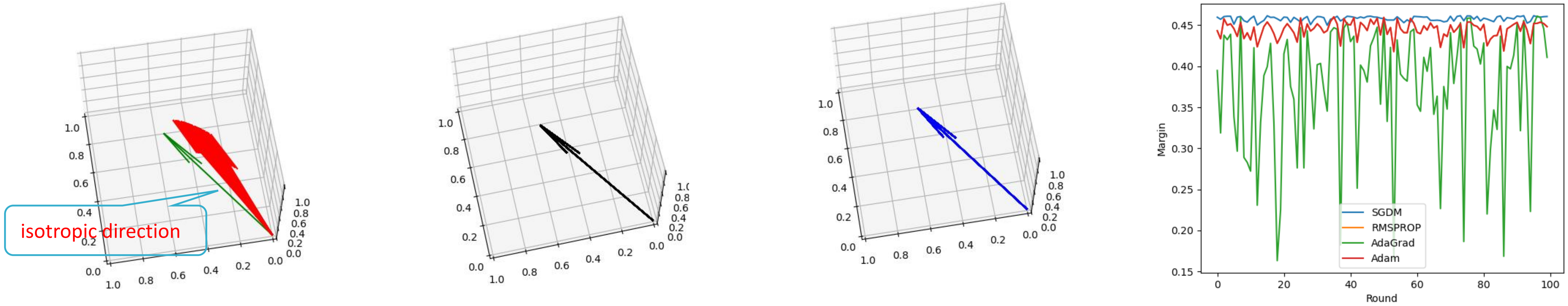
We run the experiment on MNIST dataset with a four-layer CNN.



- Margin and test accuracy of RMSProp and Adam (w/m) are significantly larger than those of AdaGrad.

Directions of h_∞

We run the experiment of a linear separable dataset with dimension 2 and parameter dimension 3.



(a) $h_\infty^{-1/2}$ in AdaGrad

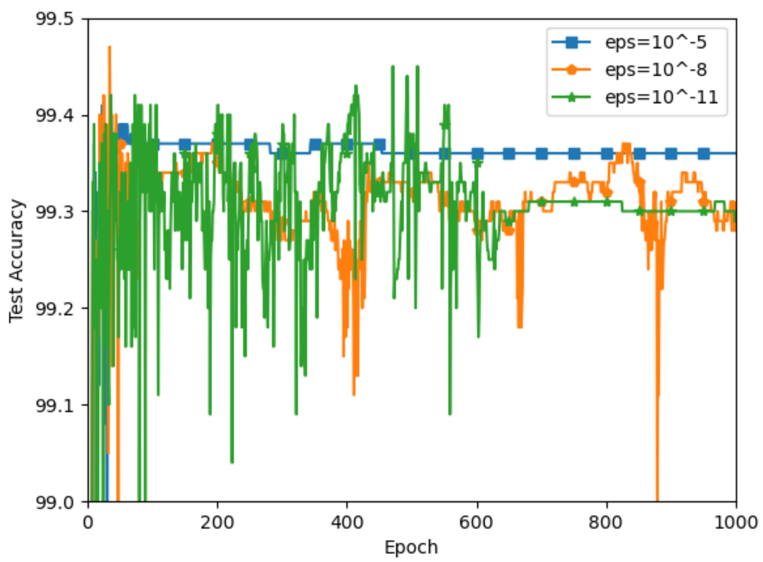
(b) $h_\infty^{-1/2}$ in RMSProp

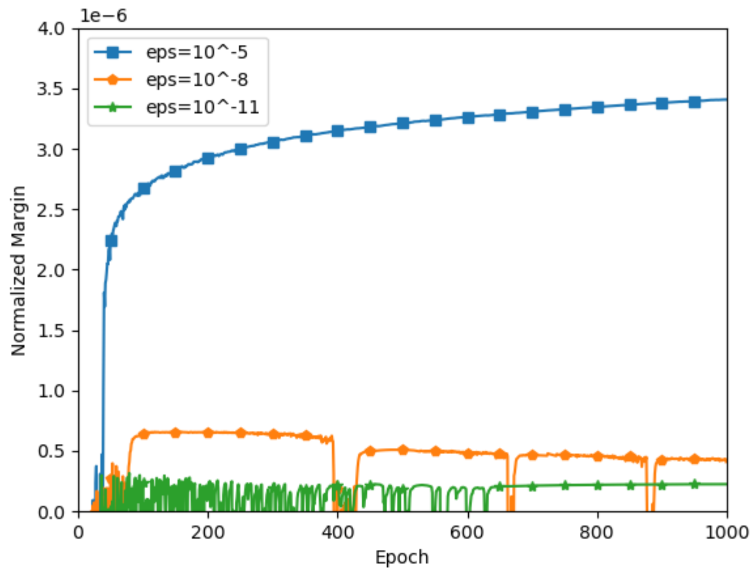
(c) $h_\infty^{-1/2}$ in Adam (w/m)

- The directions of h_∞ of RMSProp and Adam(w/m) are isotropic, while that of AdaGrad is not and varies with initialization.

Effect of ε

We run the experiment on MNIST dataset with a four-layer CNN.





- Larger ε leads to larger test accuracy and larger margin.

Thank you!

For any question, please feel free to drop a mail at

v-bohanwang@microsoft.com.