Deep AUC Maximization (DAM)

Tianbao (TB) Yang

Department of Computer Science
The University of Iowa

Before I start ...

This talk include some results from the following Papers:

1. *Non-Convex Min-Max Optimization: Provable Algorithms and Applications in Machine Learning.* Optimization Methods and Software, 2020 (2018).
2. *Stochastic AUC Maximization with Deep Neural Networks.* ICLR'20.
3. *Communication-Efficient Distributed Stochastic AUC Maximization with Deep Neural Networks.* ICML'20.
4. *Optimal Epoch Stochastic Gradient Descent Ascent Methods for Min-Max Optimization.* NeurIPS'20.
5. *Federated Deep AUC Maximization for Heterogeneous Data with a Constant Communication Complexity.* ICML'21.
6. *Fast Objective and Duality Gap Convergence for Non-convex Strongly-concave Min-max Problems.* arXiv, 2020.
7. *Robust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification.* arXiv, 2020.
8. *Stochastic Optimization of Areas Under Precision-Recall Curves with Provable Convergence.* arXiv, 2021.

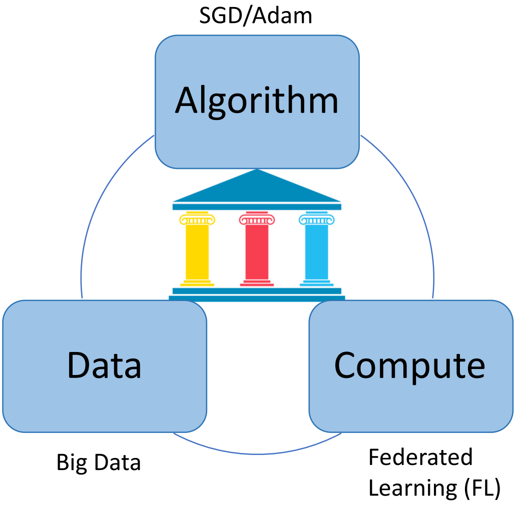
Outline

The AI **Revolution**

Deep Learning

- AI beats human on Image Recognition (2015)
- AlphaGo beats human champion (2017)
- AI beats radiologists on interpreting X-ray images (2019)
- AlphaFold solves Protein Folding (2020)
- ...

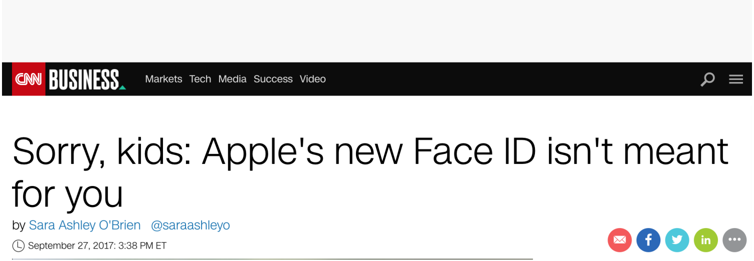
Three Pillars of Deep Learning



Accelerating AI **Democratization**

Many Challenges to be Addressed

- Data Issues (e.g., imbalanced data, small data)
- Model Issues (e.g., fairness, interpretability)
- ...



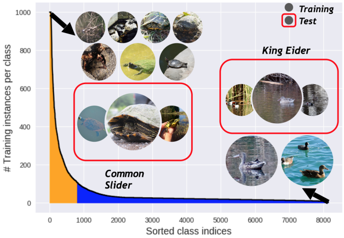
Data Imbalance

is very common in real world

- Rare Disease Identification
- Terrorist Identification
- Credit Card Fraud Detection
- ...

would cause

- dramatic performance drop
- unfairness, ethical issues



picture courtesy: Jamal et al. 2020.

DL with Imbalanced Data Faces New Challenges

Performance Metrics of Imbalanced Data

- Accuracy
  - not suitable for imbalanced data

- Area under the Curve (AUC)
  - area under ROC curve (AUROC)
  - area under Precision-Recall curve (AUPRC)
  - widely used for evaluating the performance

How to Optimize AUC for Deep Learning?

Performance Metrics of Imbalanced Data

- Accuracy
  - not suitable for imbalanced data

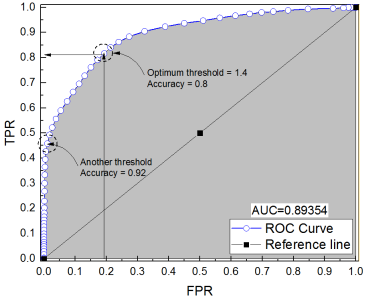
- Area under the Curve (AUC)
  - area under ROC curve (AUROC)
  - area under Precision-Recall curve (AUPRC)
  - widely used for evaluating the performance

How to Optimize AUC for Deep Learning?

Outline

AUROC

Area under ROC Curve



AUC Max. is more Difficult Accuracy Max.

Example 1		Example 2		Example 3	
Prediction	Ground Truth	Prediction	Ground Truth	Prediction	Ground Truth
0.9	1	0.9	1	0.9	1
0.8	1	**0.41**(↓)	1	**0.41**(↓)	1
0.7	1	0.7	1	**0.40**(↓)	1
0.6	0	0.6	0	**0.49**(↓)	0
0.6	0	**0.49**(↓)	0	**0.48**(↓)	0
0.47	0	0.47	0	0.47	0
0.47	0	0.47	0	0.47	0
⋮	⋮	⋮	⋮	⋮	⋮
0.1	0	0.1	0	0.1	0
Acc=0.92		Acc=0.92 (—)		Acc=0.92 (—)	
AUC=1.00		AUC=**0.89** (↓)		AUC=**0.78** (↓)	

AUC Surrogate Loss

$$\text{True-AUC}(h) = \Pr(h(\mathbf{x}) \geq h(\mathbf{x}')|y = 1, y' = -1)$$

- h: prediction model (e.g., deep neural network)
- $\mathbf{x}, \mathbf{x}'$ random data

$$\text{True-AUC}(h) = \mathrm{E}[\mathbb{I}(h(\mathbf{x}) - h(\mathbf{x}') \geq 0)|y = 1, y' = -1]$$

$$\min_h \text{AUC-Surrogate}(h) = \mathrm{E}[\ell(h(\mathbf{x}) - h(\mathbf{x}'))|y = 1, y' = -1]$$

$$\min_h \text{AUC-Surrogate}(h) = \frac{1}{n_+}\frac{1}{n_-}\sum_{\mathbf{x}_i \in \mathcal{D}_+}\sum_{\mathbf{x}_j \in \mathcal{D}_-} \ell(h(\mathbf{x}_i) - h(\mathbf{x}_j))$$

AUC Surrogate Loss

$$\text{True-AUC}(h) = \Pr(h(\mathbf{x}) \geq h(\mathbf{x}')|y = 1, y' = -1)$$

- h: prediction model (e.g., deep neural network)
- $\mathbf{x}, \mathbf{x}'$ random data

Surrogate loss function

$$\text{True-AUC}(h) = \mathrm{E}[\mathbb{I}(h(\mathbf{x}) - h(\mathbf{x}') \geq 0)|y = 1, y' = -1]$$

$$\min_h \text{AUC-Surrogate}(h) = \mathrm{E}[\ell(h(\mathbf{x}) - h(\mathbf{x}'))|y = 1, y' = -1]$$

$$\min_h \text{AUC-Surrogate}(h) = \frac{1}{n_+}\frac{1}{n_-}\sum_{\mathbf{x}_i \in \mathcal{D}_+}\sum_{\mathbf{x}_j \in \mathcal{D}_-} \ell(h(\mathbf{x}_i) - h(\mathbf{x}_j))$$

Challenge of Optimizing a Pairwise Surrogate Loss

$$\min_h \text{AUC-Surrogate}(h) = \frac{1}{n_+}\frac{1}{n_-}\sum_{\mathbf{x}_i \in \mathcal{D}_+}\sum_{\mathbf{x}_j \in \mathcal{D}_-} \ell(h(\mathbf{x}_i) - h(\mathbf{x}_j))$$

Issues of Traditional methods:

- High costs: n samples: $O(n^2)$
- Focus on Linear Models (e.g. SVM)
- Not suitable for online learning: data coming sequentially
- Not suitable for distributed optimization: data in different machines

Our Earlier Work: Zhao, Jin, Hoi, Yang (ICML 2011)

- first online AUC maximization
- large buffer, high computation, cannot scale up
- no convergence rate

Square loss Mitigates the Optimization Challenge

Square loss is an exception:

square loss

$$\min_{\mathbf{w}\in\mathbb{R}^d} A(\mathbf{w}) \triangleq \mathrm{E}_{\mathbf{z},\mathbf{z}'}[(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}') - 1)^2 | y = 1, y' = -1] \quad (1)$$

Min-max Reformulation (Ying et al. 2016):

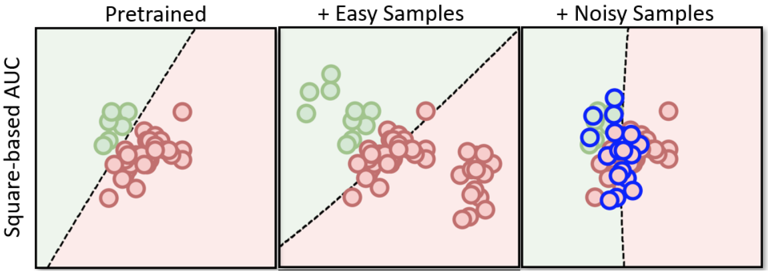
$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d \\ (a,b)\in\mathbb{R}^2}} \max_{\alpha\in\mathbb{R}} f(\mathbf{w}, a, b, \alpha) = \mathrm{E}_{\mathbf{z}}[F(\mathbf{w}, a, b, \alpha, \mathbf{z})], \quad (2)$$

- $\mathbf{z} = (\mathbf{x}, y)$
- Ying et al. (2016): focuses on linear model

Is Square loss Good for AUC Maximization?

No Really!

- Adverse Effect on Easy Data
- Sensitive to Noisy Data
- Explanation: Consider SGD update



Our Margin-based Surrogate Loss

Decomposition of Square loss:

$$\begin{aligned} A(\mathbf{w}) =& \mathrm{E}[(h_{\mathbf{w}}(\mathbf{x}) - a(\mathbf{w}))^2 | y = 1] + \mathrm{E}[(h_{\mathbf{w}}(\mathbf{x}') - b(\mathbf{w}))^2 | y' = 1] \\ &+ (1 + b(\mathbf{w}) - a(\mathbf{w}))^2 \end{aligned}$$

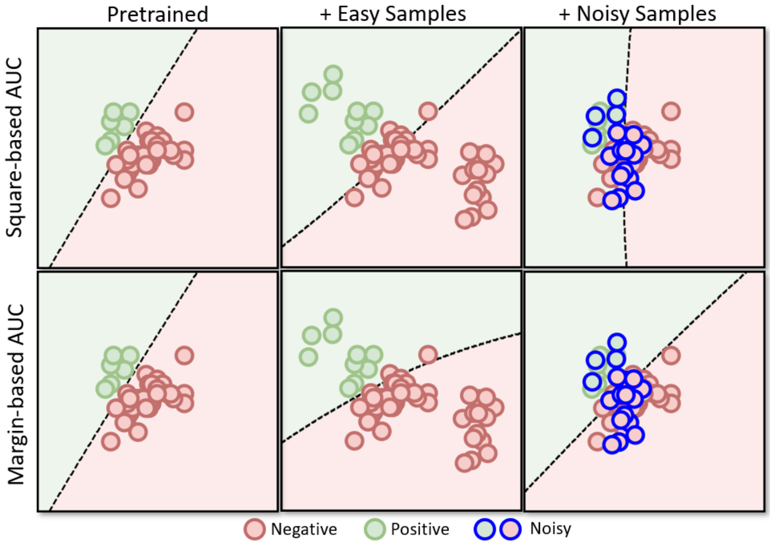
$a(\mathbf{w})$ ($b(\mathbf{w})$): average score of positive data (negative data)

Margin-based Loss: (Yuan et al. 2020)

$$\begin{aligned} A_1(\mathbf{w}) =& \mathrm{E}[(h_{\mathbf{w}}(\mathbf{x}) - a(\mathbf{w}))^2 | y = 1] + \mathrm{E}[(h_{\mathbf{w}}(\mathbf{x}') - b(\mathbf{w}))^2 | y' = 1] \\ &+ (m + b(\mathbf{w}) - a(\mathbf{w}))_+^2 \end{aligned}$$

where $[s]_+ = \max(0, s)$, m is a margin parameter.

Margin-based Surrogate Loss is more robust



Min-max formulation of Margin-based Surrogate Loss

$$
\begin{aligned}
A_1(\mathbf{w}) =& \mathrm{E}[(h_{\mathbf{w}}(\mathbf{x}) - a(\mathbf{w}))^2 | y = 1] + \mathrm{E}[(h_{\mathbf{w}}(\mathbf{x}') - b(\mathbf{w}))^2 | y' = 1] \\
& + (m + b(\mathbf{w}) - a(\mathbf{w}))_+^2
\end{aligned}
$$

Using convex conjugate: $[s]_+^2 = \max_{\alpha \geq 0} 2\alpha s - \alpha^2$

$$
\min_{\mathbf{w}, a, b} \max_{\alpha \in [0, \infty)} \mathrm{E}_{\mathbf{z}}[F(\mathbf{w}, a, b, \alpha; \mathbf{z})]
$$

- $\mathbf{z} = (\mathbf{x}, y)$
- **Non-Convex Strongly-Concave Min-Max Problem** for DL

Stochastic Gradient Descent Ascent (SGDA) Method

Consider

$$\min_{\mathbf{w}\in W}\max_{\alpha\in\Omega} f(\mathbf{w},\alpha) = \mathrm{E}_{\mathbf{z}}[f(\mathbf{w},\alpha,\mathbf{z})]$$

SGDA

$$\mathbf{w}_{t+1} = \prod_W[\mathbf{w}_t - \eta_t\nabla_w f(\mathbf{w}_t,\alpha_t,\mathbf{z}_t)], \quad \alpha_{t+1} = \prod_\Omega[\alpha_t + \eta_t\nabla_\alpha f(\mathbf{w}_t,\alpha_t,\mathbf{z}_t)]$$

Issues of Previous Works

- Analysis focuses on convex-concave
- Polynomially decreasing step size not practical

Our Stochastic Algorithms

Algorithm 1 A Stagewise Framework

from Proximal Point Method

1: **for** $s = 1, 2, ..., S$ **do**
2: Let $f_s(\mathbf{w}, \alpha) = f(\mathbf{w}, \alpha) + \gamma \|\mathbf{w} - \mathbf{w}^{(s)}\|^2$
3: $(\mathbf{w}^{(s+1)}, \alpha^{(s+1)}) = \mathcal{A}(f_s, \mathbf{w}^{(s)}, \alpha^{(s)}, \eta_s, T_s)$
4: decrease step size η_{s+1} and increase T_{s+1} accordingly
5: **end for**
6: Return $(\bar{\mathbf{w}}^{(S+1)}, \bar{\alpha}^{(S+1)})$

- Inexact Proximal Point Method
- Added Quadratic term: theoretical and practical benefits
- Easy to Implement (fixed number of iterations for each sub-problem)
- $\mathcal{A}$ could be any suitable algorithms (SGDA, AdaGrad, MirrorProx, ...)

Summary of Our Theoretical Results

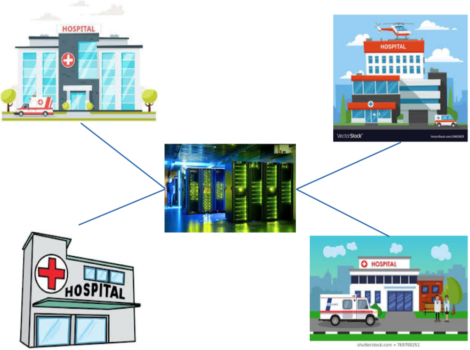
Table: Blue are our results. Red indicate optimal results. SC: strongly concave, PL: Polyak-Łojasiewicz condition. OGDA: optimistic gradient descent ascent.

Work	Conditions	Batch Size	$\mathcal{A}$	Sample Complexity
Rafique et al.'18	Concave	O(1)	SGDA	$O(\frac{1}{\epsilon^6})$
Rafique et al.'18	SC	O(1)	SGDA	$O(\frac{1}{\epsilon^4} + \frac{n}{\epsilon^2})$
Yan et al.'20	SC	$O(1)$	SGDA	$O(\frac{1}{\epsilon^4})$
Liu et al.'20	SC, PL	$O(1)$	SGDA AdaGrad	$O(\frac{1}{\mu^2\epsilon})$
Guo et al.'20	SC, PL	$O(1)$	OGDA STORM	$O(\frac{1}{\mu\epsilon})$
Lin et al.'19	Concave	$O(1)$	SGDA	$O(1/\epsilon^8)$
Lin et al.'19	SC	$O(1/\epsilon^2)$	SGDA	$O(1/\epsilon^4)$

where ϵ is the accuracy level

Federated Deep AUC Maximization

- Data at one site is biased
- Data might not be shared
- Federated Learning



Federated Deep AUC Maximization

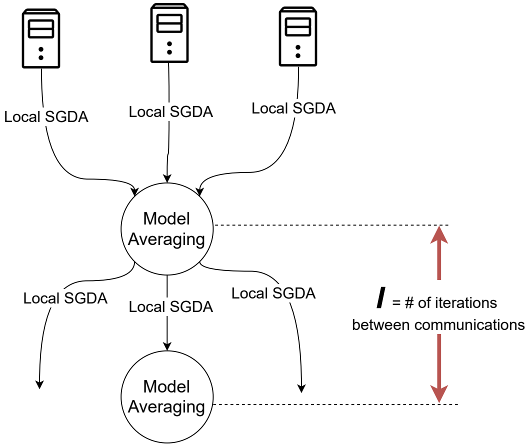
Guo et al. 2020a (ICML): First work on federated Non-convex Concave min-max learning

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d \\ (a,b)\in\mathbb{R}^2}} \max_{\alpha\in\mathbb{R}} f(\mathbf{w}, a, b, \alpha) = \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{w}, a, b, \alpha),$$

- Federated Learning: communication complexity is critical
- $f_k(\mathbf{w}, a, b, \alpha) = \mathrm{E}_{\mathbf{z}^k}[F_k(\mathbf{w}, a, b, \alpha; \mathbf{z}^k)]$, $\mathbf{z}^k = (\mathbf{x}^k, y^k) \sim \mathbb{P}_k$
- K: total number of machines

Federated Deep AUC Maximization

$\mathcal{A}$ implemented by local updates: Communication Perioadically



Complexity Result

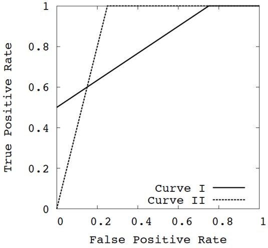
under PL condition with Heterogeneous Data

Alg.	Setting	Iteration Compl.	Comm. Compl.
Naive Parallel	Distributed	$O(1/(K\mu^2\epsilon))$	$O(1/(K\mu^2\epsilon))$
Guo et al. (ICML'20)	Distributed	$O(1/(K\mu^2\epsilon))$	$O(1/(\mu^{3/2}\epsilon^{1/2}))$
Yuan et al. (ICML'21)	Distributed	$O(1/(K\mu^2\epsilon))$	$O(1/\mu)$

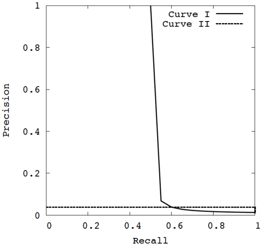
Outline

Motivation

Maximizing AUROC does not maximize AUPRC



(a) Comparing AUC-ROC for two algorithms



(b) Comparing AUC-PR for two algorithms

(picture courtesy: Davis&Goadrich, ICML'04)

Highly Imbalanced Data

In the real-world

MIT AICURES Challenge (earlier leaderboard): 2.2% Positive Examples

Rank	Model	Author	Submissions	Test PRC-AUC
1	MolecularG	AIDrug@PA	7	0.725
2	-	AGL Team	20	0.702
3	MoleculeKit	DIVE@TAMU	7	0.677
4	GB	BI	6	0.67
5	Chemprop ++	AICures@MIT	4	0.662
6	-	Mingjun Liu	3	0.657
7	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.651
8	RF + fingerprint	Cyrus Maher@Vir Bio	1	0.649
9	Graph Self-supervised Learning	SJTU_NRC_Mila	3	0.622
10	-	Congjie He	10	0.611

Rank	Model	Author	Submissions	Test ROC-AUC
1	MoleculeKit	DIVE@TAMU	7	0.928
2	Chemprop ++	AICures@MIT	4	0.877
3	-	Gianluca Bontempi	7	0.848
4	-	Apoorv Umang	1	0.84
5	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.837
6	-	Kexin Huang	1	0.824
7	Chemprop	Rajat Gupta	7	0.818
8	MLP	IITM	7	0.807
9	Graph Self-supervised Learning	SJTU_NRC_Mila	3	0.8
10	-	Congjie He	10	0.8

AUPRC Maximization is even more Challenging

Mathematically Complex

$$\mathrm{AUPRC} = \int_{-\infty}^{\infty} \Pr(Y = 1 | f(X) \geq c) d \Pr(f(X) \leq c | Y = 1),$$

We need a simpler surrogate

Average Precision (AP)

$$\mathrm{AP} = \frac{1}{n_+} \sum_{i=1}^{n} \mathbb{I}(y_i = 1) \frac{\sum_{s=1}^{n} \mathbb{I}(y_s = 1)\mathbb{I}(h_{\mathbf{w}}(\mathbf{x}_s) \geq h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{s=1}^{n} \mathbb{I}(h_{\mathbf{w}}(\mathbf{x}_s) \geq h_{\mathbf{w}}(\mathbf{x}_i))},$$

- Training Data: $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$
- $h_{\mathbf{w}}(\mathbf{x})$: prediction network

Challenges of Optimizing AP

- Indicator function $\mathbb{I}(h_{\mathbf{w}}(\mathbf{x}_s) \geq h_{\mathbf{w}}(\mathbf{x}_i))$
- AP is non-decomposable over Individual data and over Pairs of data

Previous Works in IR and CV:

- Designing different differentiable surrogate losses
- Methods for Computing Gradient
- Simply use Mini-batch Data to Compute AP and its Gradient
- Hence, No Convergence Guarantee for Stochastic Optimization
- Very Sensitive to Batch Size (Cakir et al., 2019; Qin et al., 2008; Rolinek et al., 2020)

Our work: First Stochastic Algorithm with Convergence Guarantee

Stochastic Optimization of AP (SOAP)

differentiable surrogate

$$\min_{\mathbf{w}} P(\mathbf{w}) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{D}_+} \frac{-\sum_{s=1}^{n} \mathbb{I}(y_s = 1)\ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i)}{\sum_{s=1}^{n} \ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i)}.$$

A finite-sum of two-level stochastic dependent compositional functions:

$$P(\mathbf{w}) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{D}_+} \boxed{f(g_i(\mathbf{w}))}$$

- $g_i(\mathbf{w}) = [\sum_{s=1}^{n} \mathbb{I}(y_s = 1)\ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i), \sum_{s=1}^{n} \ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i)]$
- $f(g) = \frac{-g_1}{g_2}$

Stochastic Optimization of AP (SOAP)

Estimating Gradients by Mini-batch Samples

$$\nabla_{\mathbf{w}} P(\mathbf{w}) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{D}_+} \nabla_{\mathbf{w}} g_i(\mathbf{w})^\top \left(\frac{-1}{[g_i(\mathbf{w})]_2}, \frac{[g_i(\mathbf{w})]_1}{([g_i(\mathbf{w})]_2)^2} \right)^\top.$$

Key ideas inspired from (Wang et al. 2017a, b):

- $\mathbf{u}^1_{\mathbf{x}_i} = (1-\gamma)\mathbf{u}^1_{\mathbf{x}_i} + \gamma[\tilde{g}_i(\mathbf{w}_t)]_1$
- $\mathbf{u}^2_{\mathbf{x}_i} = \max((1-\gamma)\mathbf{u}^2_{\mathbf{x}_i} + \gamma[\tilde{g}_i(\mathbf{w}_t)]_2, u_0)$

Our Gradient Estimator:

$$\nabla_{\mathbf{w}} \widehat{P}(\mathbf{w}) = \frac{1}{B_+} \sum_{\mathbf{x}_i \in \mathcal{B}_+} \nabla_{\mathbf{w}} \tilde{g}_i(\mathbf{w})^\top \left(\frac{-1}{\mathbf{u}^2_{\mathbf{x}_i}}, \frac{\mathbf{u}^1_{\mathbf{x}_i}}{(\mathbf{u}^2_{\mathbf{x}_i})^2} \right)^\top.$$

Novelty: Only Update **u** for sampled positive data

Convergence of SOAP

Novel Analysis of SGD-style Update, Momentum-Style, AMSGRAD-style, Adam-Style Update

$$\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla P(\mathbf{w}_t)\|^2\right] \leq \epsilon^2$$

- SGD-style: $T = O(1/\epsilon^5)$
- Momentum-style, AMSGRAD-style, Adam-style: $T = O(1/\epsilon^4)$ (Wang et al. 2021)

Outline

CheXpert Competition: Classifying X-ray Images

The 1st Place

Stanford ML Group (Andrew Ng)
150+ teams worldwide

Leaderboard

Will your model perform as well as radiologists in detecting different pathologies in chest X-rays?

Rank	Date	Model	AUC	Num Rads Below Curve
1	Aug 31, 2020	DeepAUC-v1 *ensemble*	0.930	2.8

CheXpert Competition: Classifying X-ray Images

Data Set

- 224,316 chest X-rays images of 65,240 patients
- 14 common chest radiographic observations
- Only 5 selected diseases for evaluation
  - Atelectasis, Cardiomegaly, Consolidation, Edema, Pleural Effusion

Results: 2%+ AUC improvement of DAM over standard DL

Model	AUROC	NRBC	Rank
Stanford Baseline (Irvin et al, AAAI'19)	0.9065	1.8	85
Hierarchical Learning (Pham et al. 2020)	0.9299	2.6	2
Ours (Yuan et al, 2020)	**0.9305**	**2.8**	**1**

CheXpert Competition: Classifying X-ray Images

Data Set

- 224,316 chest X-rays images of 65,240 patients
- 14 common chest radiographic observations
- Only 5 selected diseases for evaluation
  - Atelectasis, Cardiomegaly, Consolidation, Edema, Pleural Effusion

Results: 2%+ AUC improvement of DAM over standard DL

Model	AUROC	NRBC	Rank
Stanford Baseline (Irvin et al, AAAI'19)	0.9065	1.8	85
Hierarchical Learning (Pham et al. 2020)	0.9299	2.6	2
Ours (Yuan et al, 2020)	**0.9305**	**2.8**	**1**

Kaggle Melanoma Classification Competition

Kaggle Competition

- May 27, 2020 - August 10, 2020
- 33,126 training images, with only 584 malignant melanoma samples

>2% AUC improvement of DAM over standard DL

Results in AUROC:

- Top 1% rank (ranked 33 out of 3314 teams)
- Ensemble: our (0.9438, 10 models) vs winner (0.9490, 18 models)
- Single Model: our (0.9423) vs winner (0.9167)
- Post-competition: DAM + standard DL gives 0.9503.

Kaggle Melanoma Classification Competition

Kaggle Competition

- May 27, 2020 - August 10, 2020
- 33,126 training images, with only 584 malignant melanoma samples

$>$2% AUC improvement of DAM over standard DL

Results in AUROC:

- Top 1% rank (ranked 33 out of 3314 teams)
- Ensemble: our (0.9438, 10 models) vs winner (0.9490, 18 models)
- Single Model: our (0.9423) vs winner (0.9167)
- Post-competition: DAM + standard DL gives 0.9503.

MIT AICures Challenge: 1st Place

Drug Discovery for Fighting Secondary Effects of Covid by predicting antibacterial properties of moleculars

Rank	Model	Author	Submissions	10-fold CV ROC-AUC	10-fold CV PRC-AUC	Test ROC-AUC	Test PRC-AUC
1		DIVE@TAMU	11			0.957	0.729
2	MolecularG	AIDrug@PA	9			0.7	0.725
3		AGL Team	20			0.675	0.702
4		phucdoitoan@Fujitsu	14	0.898 +/- 0.113	0.508 +/- 0.253	0.867	0.694
5	GB	BI	6			0.698	0.67
6	Chemprop ++	AICures@MIT	4			0.877	0.662
7		Mingjun Liu	3			0.72	0.657
8	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.905 +/- 0.133	0.494 +/- 0.333	0.837	0.651

With DAM, $>$ 5% AUPRC improvement and $>$2% AUROC improvement

- Collaboration with TAMU (Dr. Shuiwang Ji's group)
- The Original Result (without using DAM): AUPRC: 0.677

MIT AICures Challenge: 1st Place

Drug Discovery for Fighting Secondary Effects of Covid by predicting antibacterial properties of moleculars

Rank	Model	Author	Submissions	10-fold CV ROC-AUC	10-fold CV PRC-AUC	Test ROC-AUC	Test PRC-AUC
1		DIVE@TAMU	11			0.957	0.729
2	MolecularG	AIDrug@PA	9			0.7	0.725
3		AGL Team	20			0.675	0.702
4		phucdoitoan@Fujitsu	14	0.898 +/- 0.113	0.508 +/- 0.253	0.867	0.694
5	GB	BI	6			0.698	0.67
6	Chemprop ++	AICures@MIT	4			0.877	0.662
7		Mingjun Liu	3			0.72	0.657
8	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.905 +/- 0.133	0.494 +/- 0.333	0.837	0.651

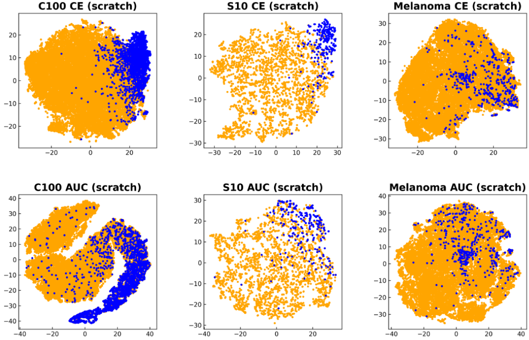
With DAM, > 5% AUPRC improvement and >2% AUROC improvement

- Collaboration with TAMU (Dr. Shuiwang Ji's group)
- The Original Result (without using DAM): AUPRC: 0.677

Outline

Is DAM just Complex Non-Convex Optimization?

No: Feature Learning is Important



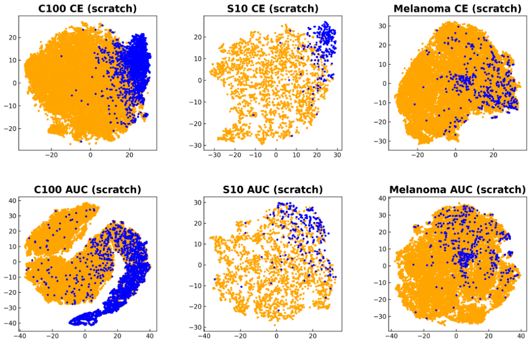
Optimizing AUC from Scratch does not work well (why?)

Our current solution: Two Stage

- Stage I: Standard DL learn the feature network
- Stage II: DAM (re-initialize classifier layer and learn all layers)

Is DAM just Complex Non-Convex Optimization?

No: Feature Learning is Important



Optimizing AUC from Scratch does not work well (why?)

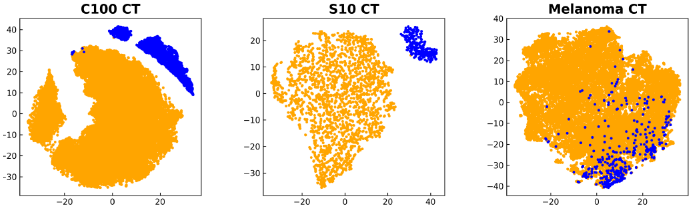
Our current solution: Two Stage

- Stage I: Standard DL learn the feature network
- Stage II: DAM (re-initialize classifier layer and learn all layers)

How to conduct DAM end-to-end?

Is there a better solution: Compositional Training

$$\max_{\mathbf{w}} \mathrm{AUC}(\boxed{\mathbf{w} - \alpha \nabla L_{\mathrm{CE}}(\mathbf{w})})$$



Very Effective Not Only for feature learning but also AUC Maximization (available soon)

Conclusions

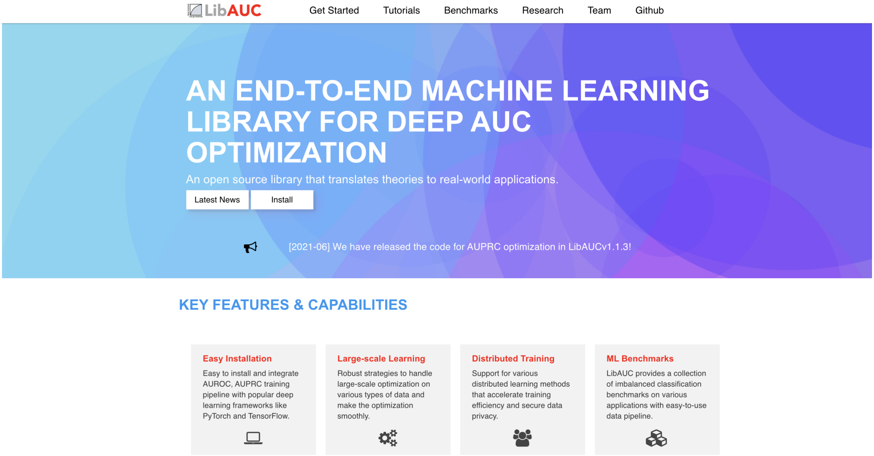
Our Achivements

- A new learning paradigm for DL with imbalanced data
- Provable and Practical Stochastic Algorithms
- For AUROC and AUPRC Maximization
- Communication Efficient FL Algorithms for AUROC
- The 1st Place at Stanford CheXpert Competition
- The 1st Place at MIT AICures Challenge

Open Problems:

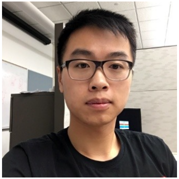
- End-to-End Learning for DAM (theory and practice)
- How do loss functions affect DL?

LibAUC: www.libauc.org



Acknowledgements: Students

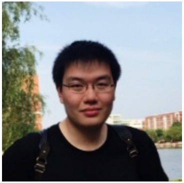
Current and Former PhD Students and Postdoc:



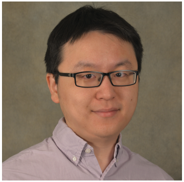
Zhuoning Yuan

Zhishuai Guo

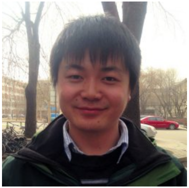
Qi Qi



Mingrui Liu
(AP, GMU)



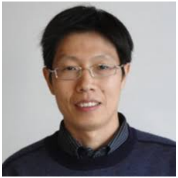
Yi Xu
(Alibaba)



Yan Yan
(AP, WSU)

Acknowledgements: Collaborators

Qihang Lin
(UIowa)



Yiming Ying
(UAlbany)

Shuiwang Ji
(TAMU)

Milan Sonka
(UIowa)

Nitesh Chawla
(ND)

Hassan Rafique
(UIndy)

Acknowledgements: Funding Support

Funding Support from NSF Career Award

References

Too long; Please refer to our papers

THANK YOU!

QUESTIONS?

Collaborations are more than Welcome!