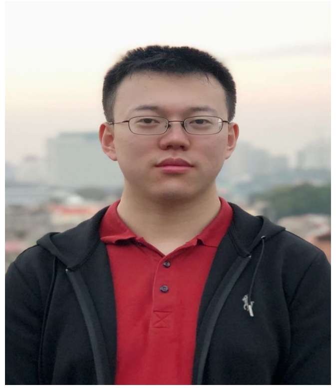
Learning via non-convex min-max games

Meisam Razaviyayn

University of Southern California

Maher Nouiehed

USC → AUB



Tianjian Haung

USC

Maziar Sanjabi

USC→ EA

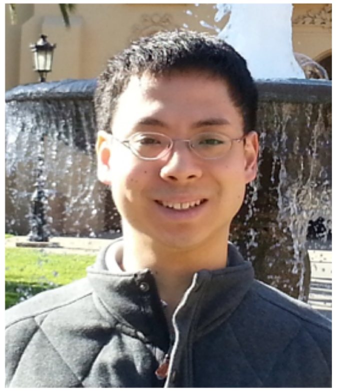
Sina Baharlouei

USC

Jimmy Ba

University of Toronto

Vector Institute



Jason Lee

Princeton

Jong-Shi Pang

USC

Non-convex min-max games/optimizations

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Non-convex min-max games/optimizations

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Convex sets

Non-convex min-max games/optimizations

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\ f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Convex sets

Gradient Lipschitz
non-convex

Non-convex min-max games/optimizations

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\ f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Convex sets

Gradient Lipschitz
non-convex

Stochastic setting

$$f(\boldsymbol{\theta},\boldsymbol{\alpha}) + p(\boldsymbol{\theta}) - q(\boldsymbol{\alpha})$$

Non-convex min-max games/optimizations

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\ f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Convex sets

Gradient Lipschitz non-convex

Stochastic setting

$$f(\boldsymbol{\theta},\boldsymbol{\alpha}) + p(\boldsymbol{\theta}) - q(\boldsymbol{\alpha})$$

- Why is this problem important? *Recent* Applications?
- Why is it challenging?

Application 1: Min-max problems and robustness

- Design a system with a robust performance against changes in certain parameters

Application 1: Min-max problems and robustness

- Design a system with a robust performance against changes in certain parameters

- Design for nominal value: $$\min_{\boldsymbol{\theta} \in \Theta} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha}_0)$$

- Robust design: $$\min_{\boldsymbol{\theta} \in \Theta} \quad \max_{\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\| \leq \delta} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Application 1: Min-max problems and robustness

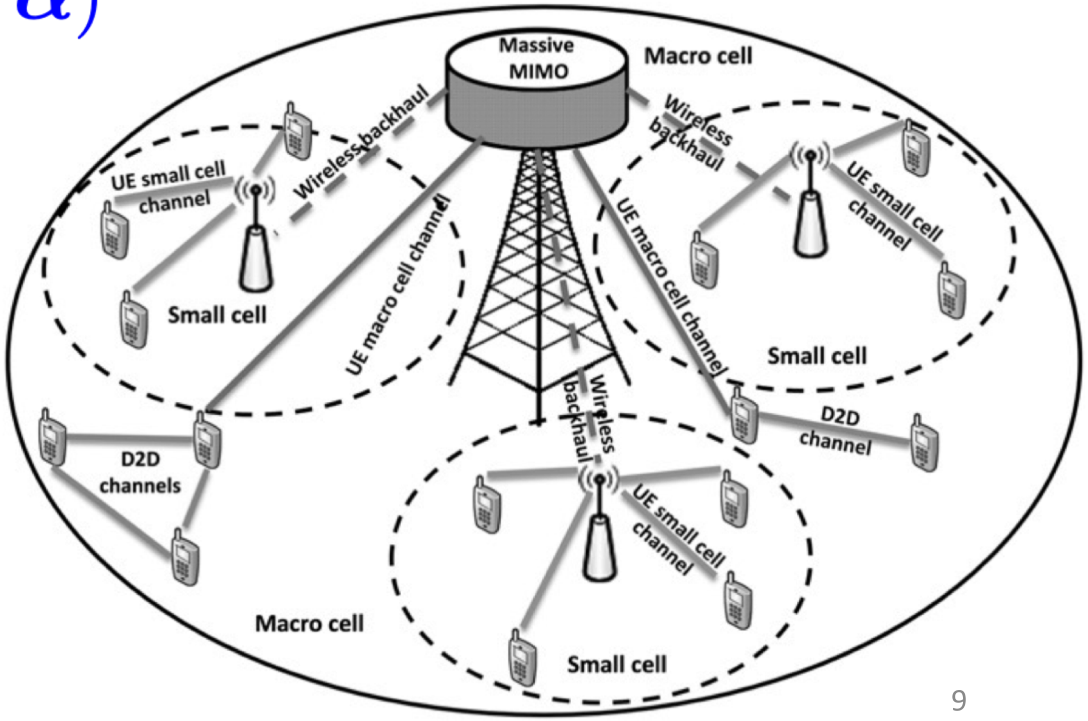
- Design a system with a robust performance against changes in certain parameters

- Design for nominal value: $$\min_{\boldsymbol{\theta}\in\Theta} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha}_0)$$

- Robust design: $$\min_{\boldsymbol{\theta}\in\Theta} \quad \max_{\|\boldsymbol{\alpha}-\boldsymbol{\alpha}_0\|\leq\delta} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

- Massive MIMO application

$$\min_{\mathbf{w}} \quad \max_{\mathbf{H}\in\mathcal{H}} \quad \ell(\mathbf{w},\mathbf{H})$$



Application 1: Min-max problems and robustness

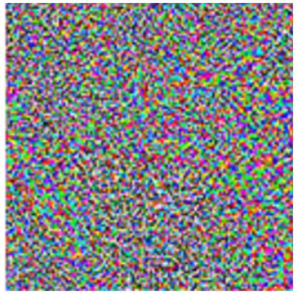
- Adversarial attacks to neural networks

Application 1: Min-max problems and robustness

➢ Adversarial attacks to neural networks

"panda"
57.7% confidence

$+\,0.007\,\times$



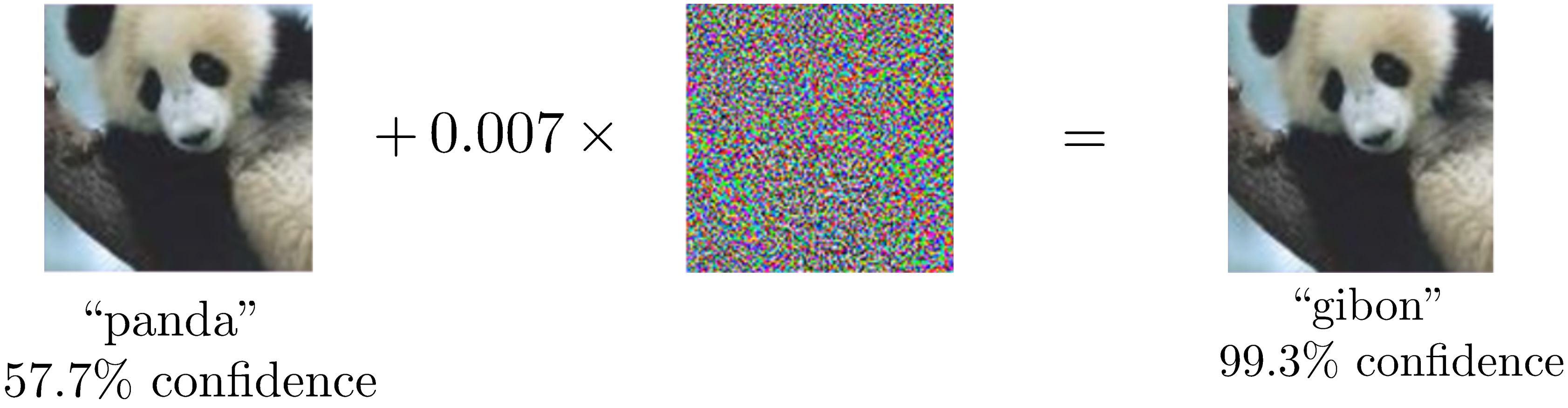
$=$

"gibon"
99.3% confidence

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Application 1: Min-max problems and robustness

- Adversarial attacks to neural networks

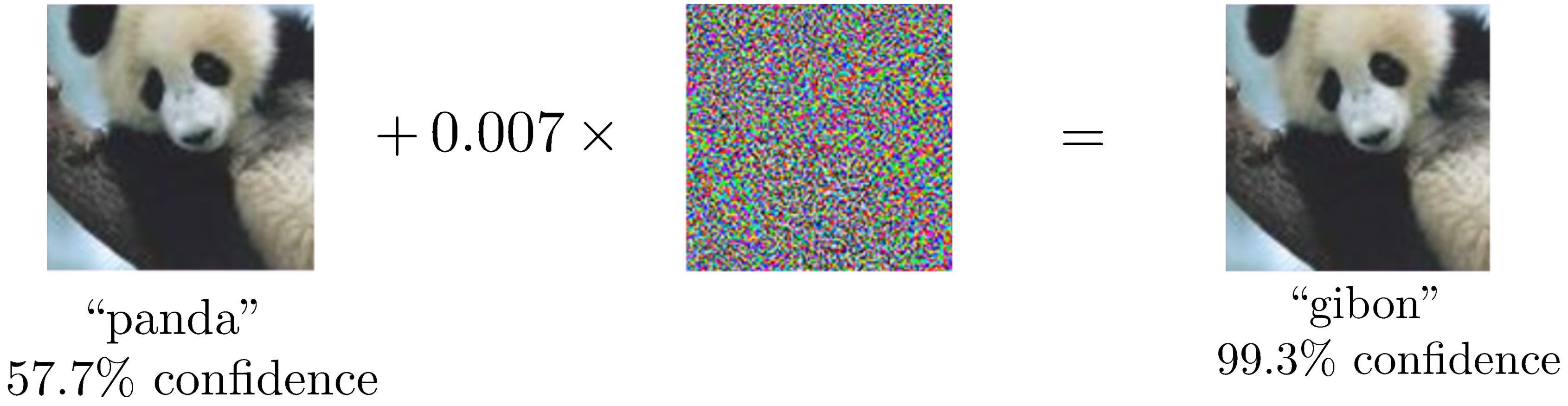


Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Carlini, Nicholas, and David Wagner. "Towards evaluating the robustness of neural networks." *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

Application 1: Min-max problems and robustness

- Adversarial attacks to neural networks



$$\min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{\theta}, \boldsymbol{\alpha}_0) \longrightarrow \min_{\boldsymbol{\theta} \in \Theta} \max_{\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\| \leq \delta} f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

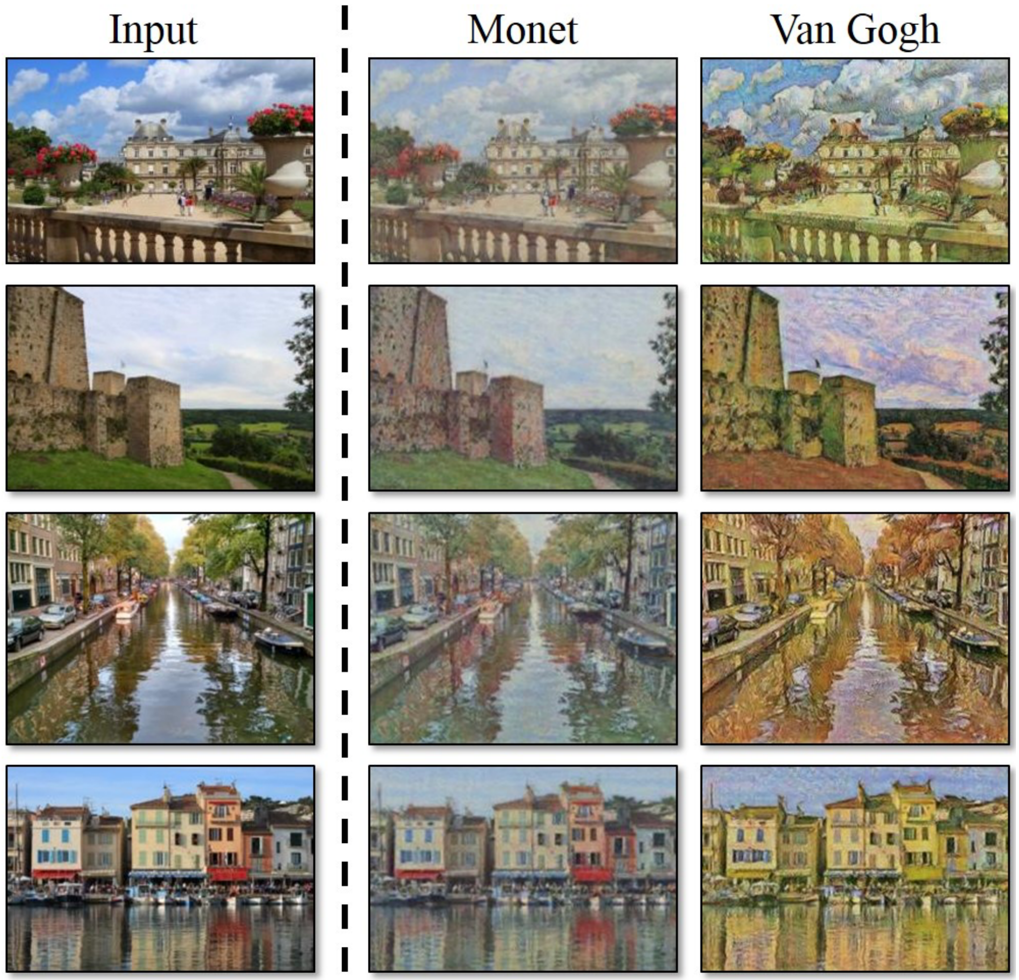
Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).
Carlini, Nicholas, and David Wagner. "Towards evaluating the robustness of neural networks." *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

Application 2: Min-max and GANs

Goal: Generate samples that look like real samples $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim \mathbb{P}_x$

Application 2: Min-max and GANs

Goal: Generate samples that look like real samples $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim \mathbb{P}_x$



https://junyanz.github.io/CycleGAN/

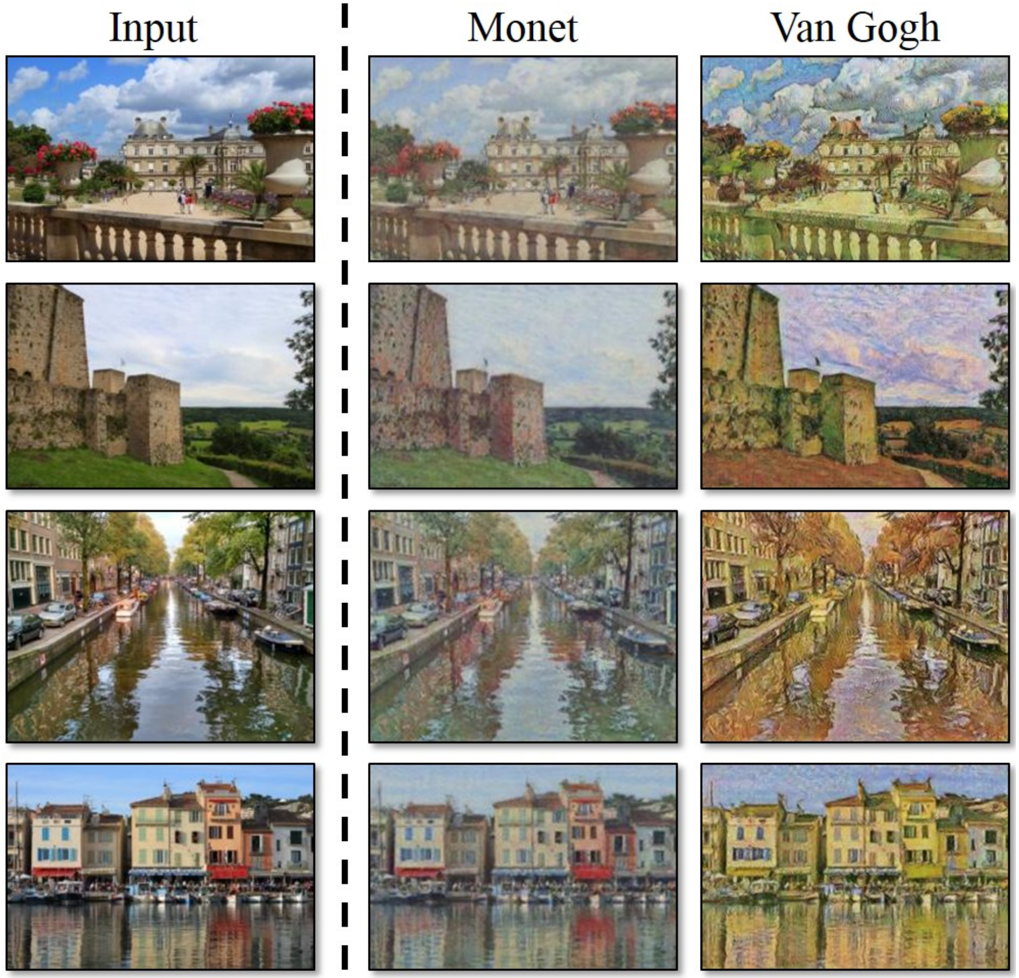
Application 2: Min-max and GANs

Goal: Generate samples that look like real samples $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim \mathbb{P}_x$

$$\mathbf{z} \longrightarrow \boxed{G(\cdot)} \longrightarrow G(\mathbf{z})$$

Neural Network

We need $G(\mathbf{z})$ to have the same distribution as $\mathbb{P}_x$



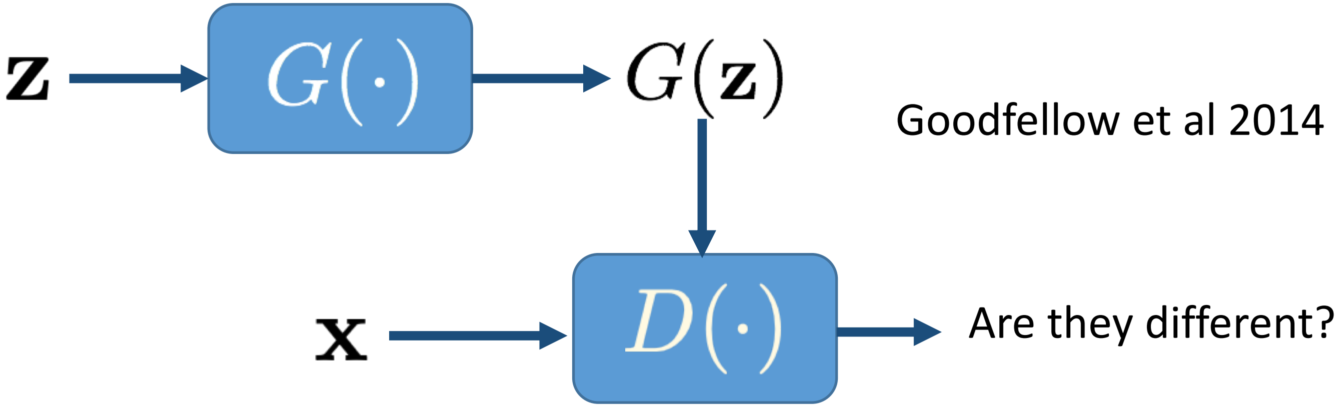
https://junyanz.github.io/CycleGAN/

Application 2: Min-max and GANs

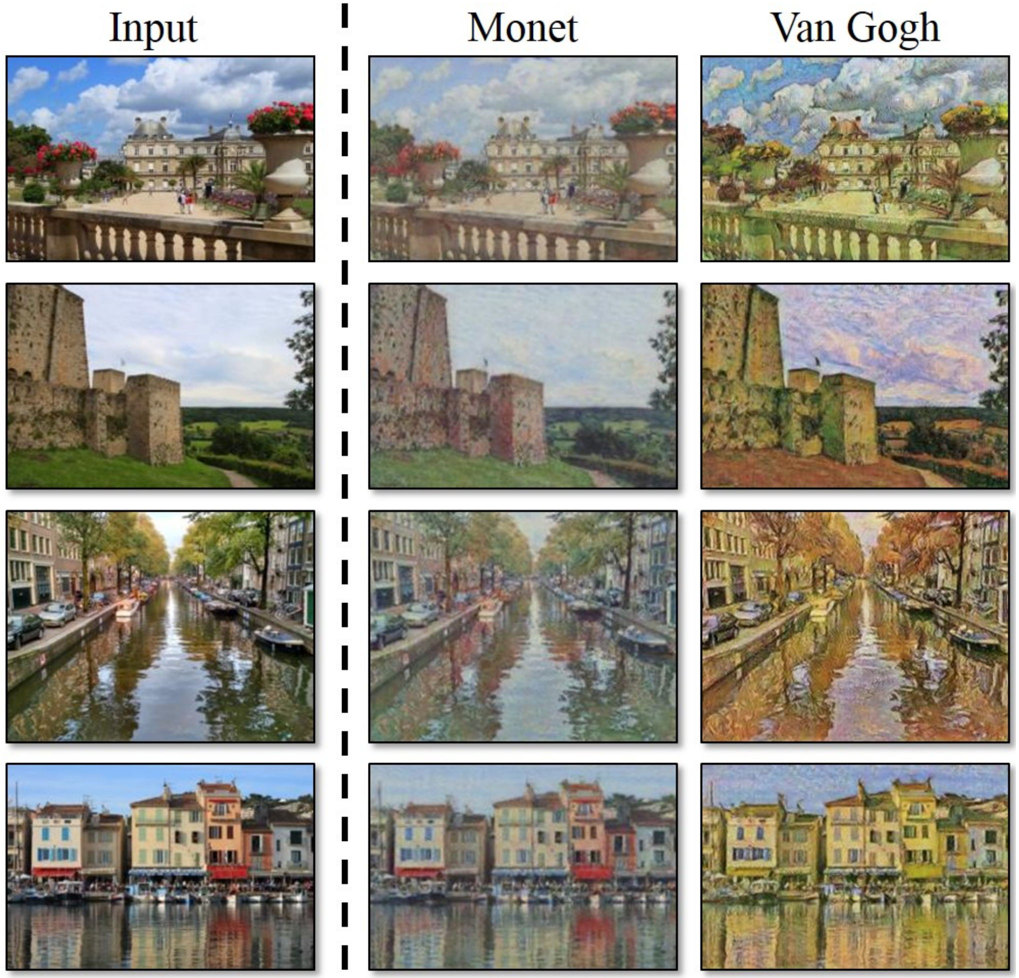
Goal: Generate samples that look like real samples $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim \mathbb{P}_x$

Neural Network

We need $G(\mathbf{z})$ to have the same distribution as $\mathbb{P}_x$



Goodfellow et al 2014



https://junyanz.github.io/CycleGAN/

Application 2: Min-max and GANs

Goal: Generate samples that look like real samples $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim \mathbb{P}_x$

$$\mathbf{z} \longrightarrow \boxed{G(\cdot)} \longrightarrow G(\mathbf{z})$$

Neural Network

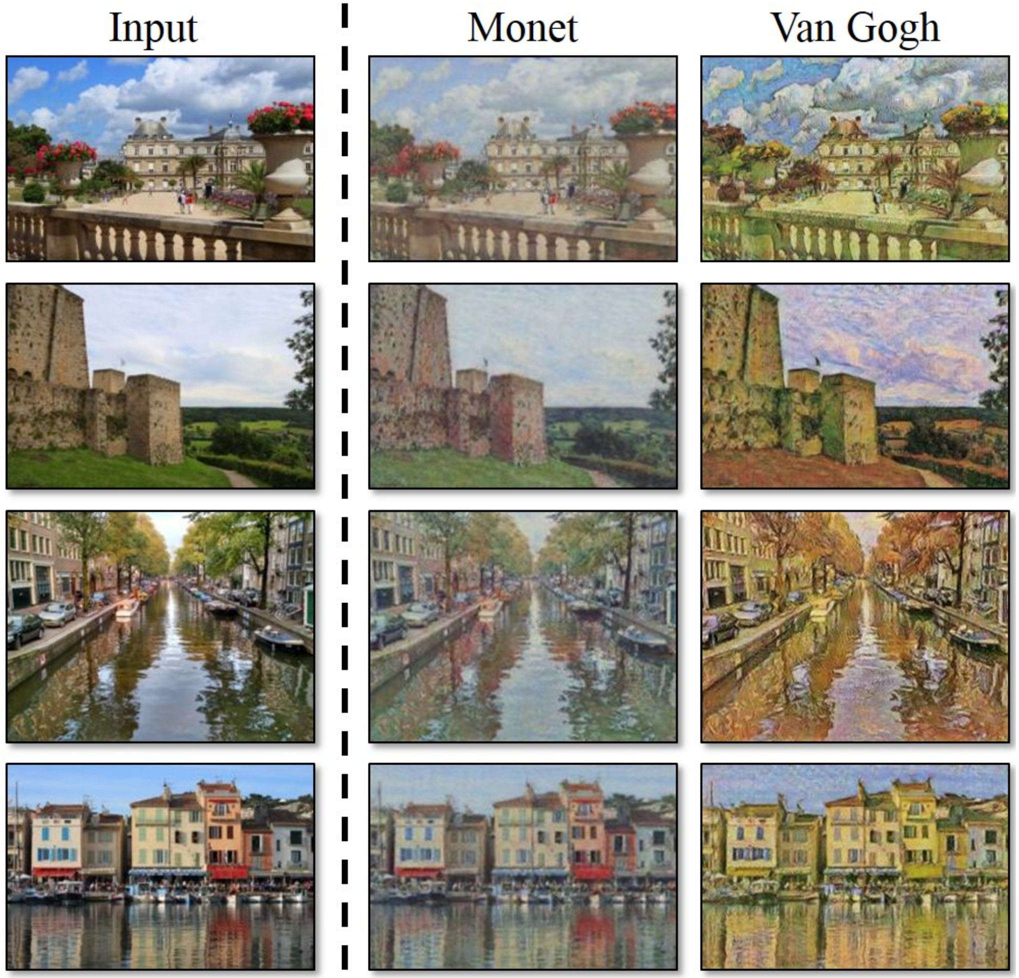
We need $G(\mathbf{z})$ to have the same distribution as $\mathbb{P}_x$

$$\begin{tikzcd} \mathbf{z} \arrow[r] & \boxed{G(\cdot)} \arrow[r] & G(\mathbf{z}) \arrow[d] & \\ & \mathbf{x} \arrow[r] & \boxed{D(\cdot)} \arrow[r] & \text{Are they different?} \end{tikzcd}$$

Goodfellow et al 2014

Are they different?

➢ The two neural networks are playing a zero-sum game



https://junyanz.github.io/CycleGAN/

Application 2: Min-max and GANs

$$\mathbf{z} \longrightarrow \boxed{G(\cdot)} \longrightarrow G(\mathbf{z})$$

- MMD GANs

$$\min_{G} \max_{D} \left\| \mathbb{E}[D(G(\mathbf{z}))] - \mathbb{E}[D(\mathbf{x})] \right\|$$

- Jensen-Shannon GANs:

$$\min_{G} \max_{D \in \mathbb{D}} \mathbb{E}_{\mathbf{x}} \left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z}} \log\left(1 - D(G(\mathbf{z}))\right)$$

$$\mathbb{D} = \text{ set of all functions with range } (0,1)$$

- Wasserstein GANs:

$$\begin{aligned} \min_{G} \max_{\gamma} \quad & \mathbb{E}_{\mathbf{x}} \left[\gamma(\mathbf{x})\right] - \mathbb{E}_{\mathbf{z}} \left[\gamma(G(\mathbf{z}))\right] \\ \text{s.t.} \quad & \gamma(\mathbf{x}) - \gamma(\mathbf{y}) \le \|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \end{aligned}$$

All are non-convex min-max problems!

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha}) \qquad\qquad \min_{\boldsymbol{\beta}\in\mathcal{B}} \quad h(\boldsymbol{\beta})$$

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

$$\min_{\boldsymbol{\beta}\in\mathcal{B}} \quad h(\boldsymbol{\beta})$$

- Apply (projected) gradient descent

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

$$\min_{\boldsymbol{\beta}\in\mathcal{B}} \quad h(\boldsymbol{\beta})$$

- Apply (projected) gradient descent:
  - Objective function improves over iterates
  - It is not exhaustive search

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

$$\min_{\boldsymbol{\beta} \in \mathcal{B}} \quad h(\boldsymbol{\beta})$$

- Apply (projected) gradient descent:
  - Objective function improves over iterates
  - It is not exhaustive search
  - Convergence to certain stationarity concepts
  - Iteration complexity lower- and upper- bounds

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

- What should we do? Gradient descent/ascent?

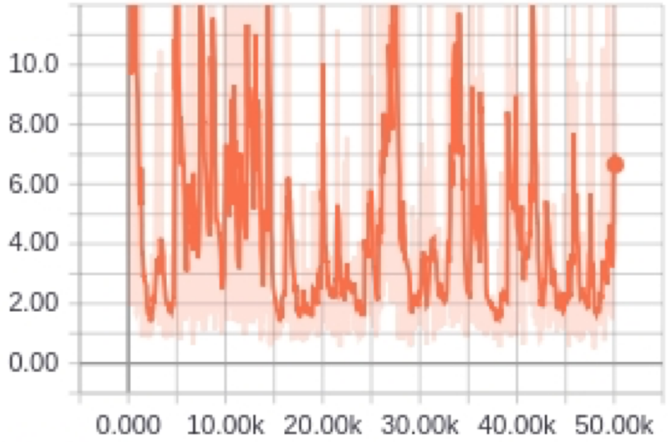
$$\min_{\boldsymbol{\beta}\in\mathcal{B}}\quad h(\boldsymbol{\beta})$$

- Apply (projected) gradient descent:
  - Objective function improves over iterates
  - It is not exhaustive search
  - Convergence to certain stationarity concepts
  - Iteration complexity lower- and upper- bounds

Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- What should we do? Gradient descent/ascent?



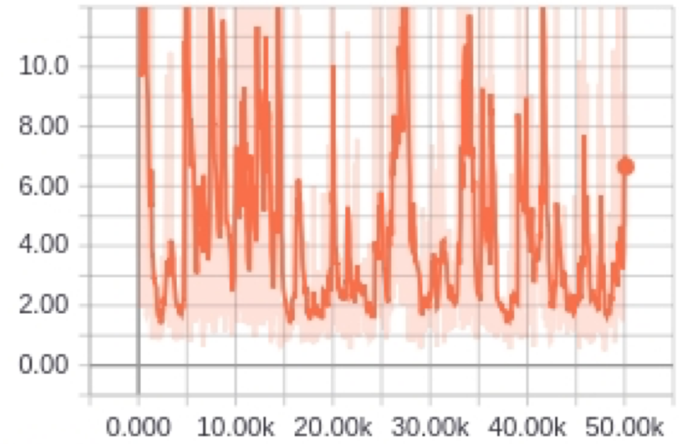
$$\min_{\boldsymbol{\beta} \in \mathcal{B}} \quad h(\boldsymbol{\beta})$$

- Apply (projected) gradient descent:
  - Objective function improves over iterates
  - It is not exhaustive search
  - Convergence to certain stationarity concepts
  - Iteration complexity lower- and upper- bounds

Why are non-convex min-max problems challenging?

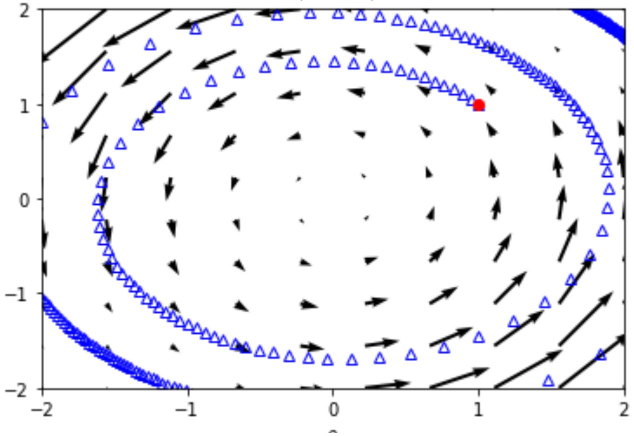
$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- What should we do? Gradient descent/ascent?



Iterates trajectory

for $f(\theta, \alpha) = \theta\alpha$



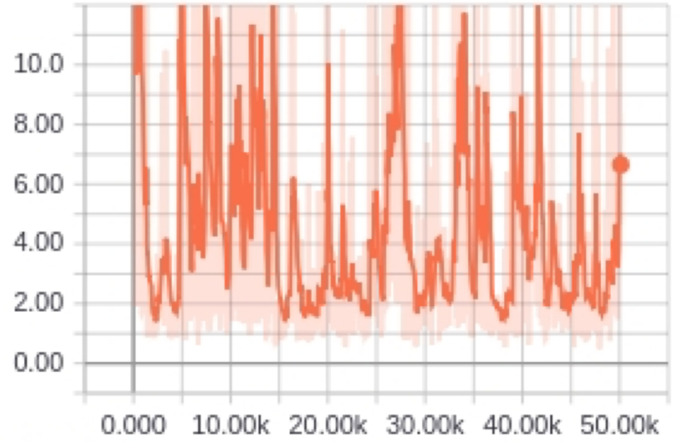
$$\min_{\boldsymbol{\beta}\in\mathcal{B}} \quad h(\boldsymbol{\beta})$$

- Apply (projected) gradient descent:
  - Objective function improves over iterates
  - It is not exhaustive search
  - Convergence to certain stationarity concepts
  - Iteration complexity lower- and upper- bounds

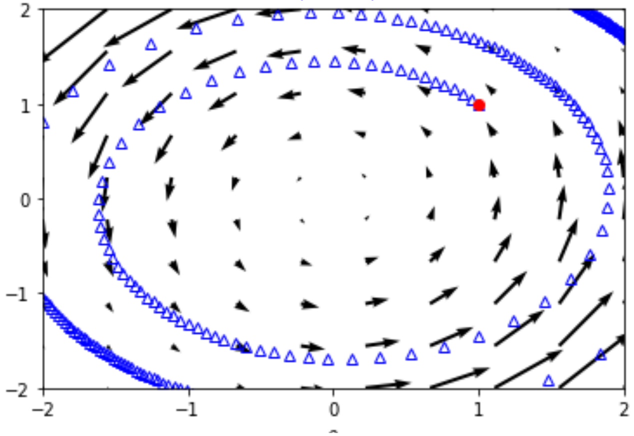
Why are non-convex min-max problems challenging?

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- ➢ What should we do? Gradient descent/ascent?



Iterates trajectory for $f(\theta, \alpha) = \theta\alpha$



- ➢ Even more: what should we compute?

$$\min_{\boldsymbol{\beta} \in \mathcal{B}} \quad h(\boldsymbol{\beta})$$

- ➢ Apply (projected) gradient descent:
  - ➢ Objective function improves over iterates
  - ➢ It is not exhaustive search
  - ➢ Convergence to certain stationarity concepts
  - ➢ Iteration complexity lower- and upper- bounds

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \geq 0, \ \ \forall \boldsymbol{\theta} \in \Theta \qquad \langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \rangle \leq 0, \ \ \forall \boldsymbol{\alpha} \in \mathcal{A}$$

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \geq 0, \ \ \forall \boldsymbol{\theta} \in \Theta \qquad \langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \rangle \leq 0, \ \ \forall \boldsymbol{\alpha} \in \mathcal{A}$$

- Existence?

Defining a stationary concept

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \geq 0, \;\; \forall \boldsymbol{\theta} \in \Theta \qquad \langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \rangle \leq 0, \;\; \forall \boldsymbol{\alpha} \in \mathcal{A}$$

- Existence?

Theorem [Pang-Razaviyayn 2016]: Suppose that the constraint sets are non-empty, compact, and convex. Moreover, assume that $f(\cdot)$ is continuously differentiable (+/- convex). Then, the first-order NE exists.

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \geq 0, \ \ \forall \boldsymbol{\theta} \in \Theta \qquad \langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \rangle \leq 0, \ \ \forall \boldsymbol{\alpha} \in \mathcal{A}$$

- Existence?

> **Theorem [Pang-Razaviyayn 2016]:** Suppose that the constraint sets are non-empty, compact, and convex. Moreover, assume that $f(\cdot)$ is continuously differentiable (+/- convex). Then, the first-order NE exists.

- Can we compute it?

Defining a stationary concept

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*,\boldsymbol{\alpha}^*), \boldsymbol{\theta}-\boldsymbol{\theta}^*\rangle \geq 0,\ \ \forall \boldsymbol{\theta}\in\Theta \qquad \langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*,\boldsymbol{\alpha}^*), \boldsymbol{\alpha}-\boldsymbol{\alpha}^*\rangle \leq 0,\ \ \forall \boldsymbol{\alpha}\in\mathcal{A}$$

- Existence?

> **Theorem [Pang-Razaviyayn 2016]:** Suppose that the constraint sets are non-empty, compact, and convex. Moreover, assume that $f(\cdot)$ is continuously differentiable (+/- convex). Then, the first-order NE exists.

- Can we compute it? ϵ –First-order NE:

Defining a stationary concept

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Game perspective → First-order Nash equilibrium:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \geq 0, \;\; \forall \boldsymbol{\theta} \in \Theta \qquad \langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \rangle \leq 0, \;\; \forall \boldsymbol{\alpha} \in \mathcal{A}$$

- Existence?

> **Theorem [Pang-Razaviyayn 2016]:** Suppose that the constraint sets are non-empty, compact, and convex. Moreover, assume that $f(\cdot)$ is continuously differentiable (+/- convex). Then, the first-order NE exists.

- Can we compute it? ϵ −First-order NE:

$$\langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle \geq -\epsilon, \;\; \forall \boldsymbol{\theta} \in \Theta \quad \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \leq 1$$

$$\langle \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*), \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \rangle \leq \epsilon, \;\; \forall \boldsymbol{\alpha} \in \mathcal{A} \quad \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\| \leq 1$$

Computation of $\epsilon -$ first-order Nash equilibrium

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Computation of $\epsilon -$ first-order Nash equilibrium

$$\min_{\boldsymbol{\theta} \in \Theta} \boxed{\max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

Computation of ϵ − first-order Nash equilibrium

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\boxed{g(\boldsymbol{\theta})}$$

Computation of $\epsilon -$ first-order Nash equilibrium

$$\min_{\boldsymbol{\theta} \in \Theta} \boxed{\max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta} \in \Theta} \boxed{g(\boldsymbol{\theta})}$$

Computation of $\epsilon -$ first-order Nash equilibrium

- Apply gradient descent to $g(\cdot)$

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^{t} - \gamma \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{t})\right]_{+}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

Computation of $\epsilon -$ first-order Nash equilibrium

- Apply gradient descent to $g(\cdot)$

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^t)\right]_+$$

- Is it differentiable?

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

Computation of ϵ − first-order Nash equilibrium

- Apply gradient descent to $g(\cdot)$

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^{t} - \gamma \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{t})\right]_{+}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Is it differentiable?
  - When $f(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$

Computation of $\epsilon -$ first-order Nash equilibrium

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Apply gradient descent to $g(\cdot)$

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^{t} - \gamma \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{t})\right]_{+}$$

- Is it differentiable?
  - When $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$

Danskin's Theorem: $\nabla_{\theta} g(\boldsymbol{\theta}_0) = \nabla_{\theta} f(\boldsymbol{\theta}_0, \boldsymbol{\alpha}_0) \quad \text{where} \quad \boldsymbol{\alpha}_0 = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta}_0, \boldsymbol{\alpha})$

Computation of ϵ – first-order Nash equilibrium

- Apply gradient descent to $g(\cdot)$

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^t)\right]_+$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

$$\downarrow$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Is it differentiable?
  - When $f(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$

Danskin's Theorem: $\nabla_\theta g(\boldsymbol{\theta}_0) = \nabla_\theta f(\boldsymbol{\theta}_0, \boldsymbol{\alpha}_0) \quad \text{where} \quad \boldsymbol{\alpha}_0 = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta}_0, \boldsymbol{\alpha})$

Algorithm: for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t, \boldsymbol{\alpha})$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+$$

Iteration complexity

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha} \in \mathcal{A}} \ f(\boldsymbol{\theta}^t, \boldsymbol{\alpha})$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+$$

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \ f(\boldsymbol{\theta}^t, \boldsymbol{\alpha})$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+$$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2} \log \epsilon^{-1})$ gradient evaluations for computing ϵ −first-order NE.

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

$$\begin{aligned}&\text{for } t = 1, 2, \ldots \text{ do}\\ &\quad \boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \; f(\boldsymbol{\theta}^t, \boldsymbol{\alpha})\\ &\quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+\end{aligned} \longrightarrow \text{Need } \mathcal{O}(\epsilon^{-2}) \text{ iterations on } \boldsymbol{\theta}$$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2} \log \epsilon^{-1})$ gradient evaluations for computing ϵ –first-order NE.

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

$$\begin{aligned}
&\text{for } t = 1,2,\ldots \text{ do}\\
&\quad \boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t,\boldsymbol{\alpha})\\
&\quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+
\end{aligned}$$

Apply K steps of projected gradient ascent on $\boldsymbol{\alpha}$ $\quad K \approx \mathcal{O}(\log(\epsilon^{-1}))$

Need $\mathcal{O}(\epsilon^{-2})$ iterations on $\boldsymbol{\theta}$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2}\log\epsilon^{-1})$ gradient evaluations for computing ϵ –first-order NE.

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

$$\text{for } t = 1,2,\ldots \text{ do}$$

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}}\ f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}) \longrightarrow$$

Apply K steps of projected gradient ascent on $\boldsymbol{\alpha}$ $K \approx \mathcal{O}(\log(\epsilon^{-1}))$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+ \longrightarrow \text{Need } \mathcal{O}(\epsilon^{-2}) \text{ iterations on } \boldsymbol{\theta}$$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2}\log\epsilon^{-1})$ gradient evaluations for computing ϵ –first-order NE.

- Optimal rate up to logarithmic factors

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

$$\boxed{\begin{array}{l} \text{for } t = 1, 2, \ldots \text{ do} \\ \quad \boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \; f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}) \\ \quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+ \end{array}}$$

Apply K steps of projected gradient ascent on $\boldsymbol{\alpha}$ $\quad K \approx \mathcal{O}(\log(\epsilon^{-1}))$

Need $\mathcal{O}(\epsilon^{-2})$ iterations on $\boldsymbol{\theta}$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2}\log\epsilon^{-1})$ gradient evaluations for computing ϵ –first-order NE.

- Optimal rate up to logarithmic factors
- Can be obtained under Polyak-Łojasiewicz (PL) condition
  - Requires establishing Danskin's-type result under PL assumption

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t, \boldsymbol{\alpha})$$

$$\boldsymbol{\theta}^{t+1} = [\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})]_+$$

Apply K steps of projected gradient ascent on $\boldsymbol{\alpha}$ $\quad K \approx \mathcal{O}(\log(\epsilon^{-1}))$

Need $\mathcal{O}(\epsilon^{-2})$ iterations on $\boldsymbol{\theta}$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2} \log \epsilon^{-1})$ gradient evaluations for computing ϵ –first-order NE.

- Optimal rate up to logarithmic factors
- Can be obtained under Polyak-Łojasiewicz (PL) condition
  - Requires establishing Danskin's-type result under PL assumption

Strongly convex composite with affine ✓

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

$$\begin{aligned}
&\text{for } t = 1, 2, \ldots \text{ do} \\
&\quad \boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}) \\
&\quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+
\end{aligned}$$

Apply K steps of projected gradient ascent on $\boldsymbol{\alpha}$ $\quad K \approx \mathcal{O}(\log(\epsilon^{-1}))$

Need $\mathcal{O}(\epsilon^{-2})$ iterations on $\boldsymbol{\theta}$

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is strongly concave in $\boldsymbol{\alpha}$. Then, the algorithm requires $O(\epsilon^{-2}\log\epsilon^{-1})$ gradient evaluations for computing ϵ −first-order NE.

- Optimal rate up to logarithmic factors
- Can be obtained under Polyak-Łojasiewicz (PL) condition
  - Requires establishing Danskin's-type result under PL assumption

Strongly convex composite with affine ✓

Extend further?

Non-convex-concave scenario

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$ (but not strongly concave)

Non-convex-concave scenario

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$ (but not strongly concave)
- $g(\cdot)$ is no longer differentiable

Non-convex-concave scenario

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$ (but not strongly concave)
- $g(\cdot)$ is no longer differentiable
- Smoothify $g(\cdot)$

Non-convex-concave scenario

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Assume $f(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$ (but not strongly concave)

- $g(\cdot)$ is no longer differentiable

- Smoothify $g(\cdot)$ $$g_\lambda(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta}, \boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

Non-convex-concave scenario

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$ (but not strongly concave)
- $g(\cdot)$ is no longer differentiable
- Smoothify $g(\cdot)$ $\quad g_\lambda(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta},\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$
- **Algorithm:**

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} g_\lambda(\boldsymbol{\theta}^t)\right]_+$$

Non-convex-concave scenario

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})}$$

$$\min_{\boldsymbol{\theta}\in\Theta} \boxed{g(\boldsymbol{\theta})}$$

- Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$ (but not strongly concave)
- $g(\cdot)$ is no longer differentiable
- Smoothify $g(\cdot)$ $\quad g_\lambda(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{\alpha}\in\mathcal{A}} f(\boldsymbol{\theta},\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$
- **Algorithm:**

$$\boldsymbol{\theta}^{t+1} \approx \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} g_\lambda(\boldsymbol{\theta}^t)\right]_+ \xrightarrow{\text{Danskin's Theorem}}$$

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \; f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$$

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Algorithm:

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+$$

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

Algorithm:

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+$$

for $t = 1, 2, \ldots$ do

$\quad \boldsymbol{\alpha}_0 = \boldsymbol{\alpha}^t$

$\quad$ for $\tau = 1, 2, \ldots, K$ do

$\quad\quad \boldsymbol{\alpha}_{\tau+1} = \left[\boldsymbol{\alpha}_\tau + \gamma\nabla_{\boldsymbol{\alpha}} f_\lambda(\boldsymbol{\theta}^t, \boldsymbol{\alpha}_\tau)\right]_+$

$\quad$ end for

$\quad \boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}_K$

$\quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^{t+1})\right]_+$

end for

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Algorithm:

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$$

for $t = 1, 2, \ldots$ do

$\quad \boldsymbol{\alpha}_0 = \boldsymbol{\alpha}^t$

$\quad$ for $\tau = 1, 2, \ldots, K$ do

$\quad\quad \boldsymbol{\alpha}_{\tau+1} = \left[\boldsymbol{\alpha}_\tau + \gamma\nabla_{\boldsymbol{\alpha}} f_\lambda(\boldsymbol{\theta}^t,\boldsymbol{\alpha}_\tau)\right]_+$

$\quad$ end for

$\quad \boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}_K$

$\quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$

end for

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$. Then, the above algorithm requires $O(\epsilon^{-3.5}\log\epsilon^{-1})$ gradient evaluations for computing ϵ –first-order NE.

$O(\epsilon^{-3.5})$ **vs** $O(\epsilon^{-4})$ without adding a regularizer/acceleration

Algorithm	Iteration Complexity
[Lu et al 2018]	$O(\epsilon^{-4})$
[Lin et al 2019]	$O(\epsilon^{-4})$

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Algorithm:

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$$

for $t = 1, 2, \ldots$ do

$\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}^t$

for $\tau = 1, 2, \ldots, K$ do

$\boldsymbol{\alpha}_{\tau+1} = \left[\boldsymbol{\alpha}_\tau + \gamma\nabla_{\boldsymbol{\alpha}} f_\lambda(\boldsymbol{\theta}^t,\boldsymbol{\alpha}_\tau)\right]_+$

end for

$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}_K$

$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$

end for

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$. Then, the above algorithm requires $O(\epsilon^{-3.5}\log\epsilon^{-1})$ gradient evaluations for computing ϵ −first-order NE.

$O(\epsilon^{-3.5})$ **vs** $O(\epsilon^{-4})$ without adding a regularizer/acceleration

[Thekumparampil et al 2019]

Algorithm	Iteration Complexity
[Lu et al 2018]	$O(\epsilon^{-4})$
[Lin et al 2019]	$O(\epsilon^{-4})$

Iteration complexity

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad f(\boldsymbol{\theta},\boldsymbol{\alpha})$$

Algorithm:

for $t = 1, 2, \ldots$ do

$$\boldsymbol{\alpha}^{t+1} \approx \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}}\ f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}) - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|^2$$

$$\boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$$

for $t = 1, 2, \ldots$ do

$\quad \boldsymbol{\alpha}_0 = \boldsymbol{\alpha}^t$

$\quad$ for $\tau = 1, 2, \ldots, K$ do

$\quad\quad \boldsymbol{\alpha}_{\tau+1} = \left[\boldsymbol{\alpha}_\tau + \gamma\nabla_{\boldsymbol{\alpha}} f_\lambda(\boldsymbol{\theta}^t,\boldsymbol{\alpha}_\tau)\right]_+$

$\quad$ end for

$\quad \boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}_K$

$\quad \boldsymbol{\theta}^{t+1} = \left[\boldsymbol{\theta}^t - \gamma\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^t,\boldsymbol{\alpha}^{t+1})\right]_+$

end for

Theorem [Nouiehed, Huang, Sanjabi, Lee, Razaviyayn 2018]: Assume $f(\boldsymbol{\theta},\boldsymbol{\alpha})$ is concave in $\boldsymbol{\alpha}$. Then, the above algorithm requires $O(\epsilon^{-3.5}\log\epsilon^{-1})$ gradient evaluations for computing ϵ −first-order NE.

$O(\epsilon^{-3.5})$ **vs** $O(\epsilon^{-4})$ without adding a regularizer/acceleration

[Thekumparampil et al 2019]

Algorithm	Iteration Complexity
[Lu et al 2018]	$O(\epsilon^{-4})$
[Lin et al 2019]	$O(\epsilon^{-4})$

Are these results useful in practice?

Training robust neural networks

"panda"
57.7% confidence

$+ 0.007 \times$

$=$

"gibon"
99.3% confidence

Training robust neural networks

“panda”
57.7% confidence

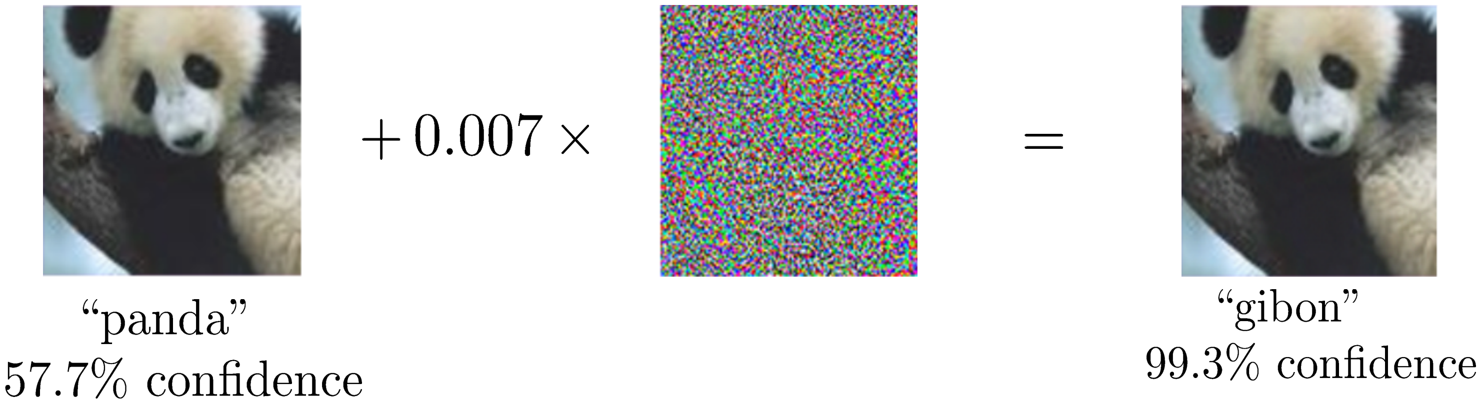
$+0.007 \times$

$=$

“gibon”
99.3% confidence

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)$$

Training robust neural networks



$$\boxed{\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)} \longrightarrow \boxed{\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \le \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})}$$

Training robust neural networks

$+\,0.007\times$

$=$

"panda"
57.7% confidence

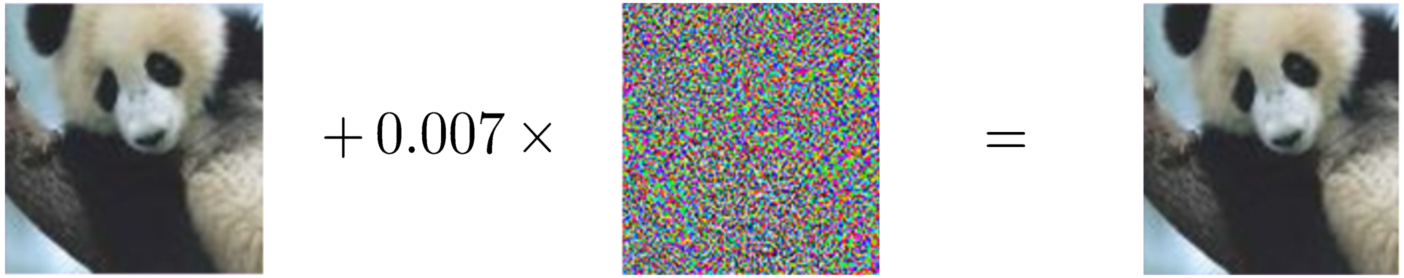
"gibon"
99.3% confidence

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i) \quad \longrightarrow \quad \min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \le \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

[Madry et al. 2017]: **Repeat**:

- Apply multi-steps of gradient ascent on $\boldsymbol{\delta}$ (reinitialize multiple times and pick the best)
- Perform one step of gradient descent on **w**

Training robust neural networks



"panda"
57.7% confidence

"gibon"
99.3% confidence

$$\boxed{\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)} \longrightarrow \boxed{\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \le \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})}$$

[Madry et al. 2017]: **Repeat**:

- Apply multi-steps of gradient ascent on $\boldsymbol{\delta}$ (reinitialize multiple times and pick the best)
- Perform one step of gradient descent on **w**

- No theoretical convergence guarantee, not scalable, and requires heavy tuning to work

Training robust neural networks

$+\,0.007\times$

$=$

"panda"
57.7% confidence

"gibon"
99.3% confidence

$$\boxed{\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)} \longrightarrow \boxed{\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\|\le\epsilon} \ell(\mathbf{w}, \mathbf{x}_i+\boldsymbol{\delta})}$$

[Madry et al. 2017]: **Repeat**:

- Apply multi-steps of gradient ascent on $\boldsymbol{\delta}$ (reinitialize multiple times and pick the best)
- Perform one step of gradient descent on **w**

- No theoretical convergence guarantee, not scalable, and requires heavy tuning to work
- Can we apply our theory and algorithm?

Training robust neural networks

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \leq \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

Training robust neural networks

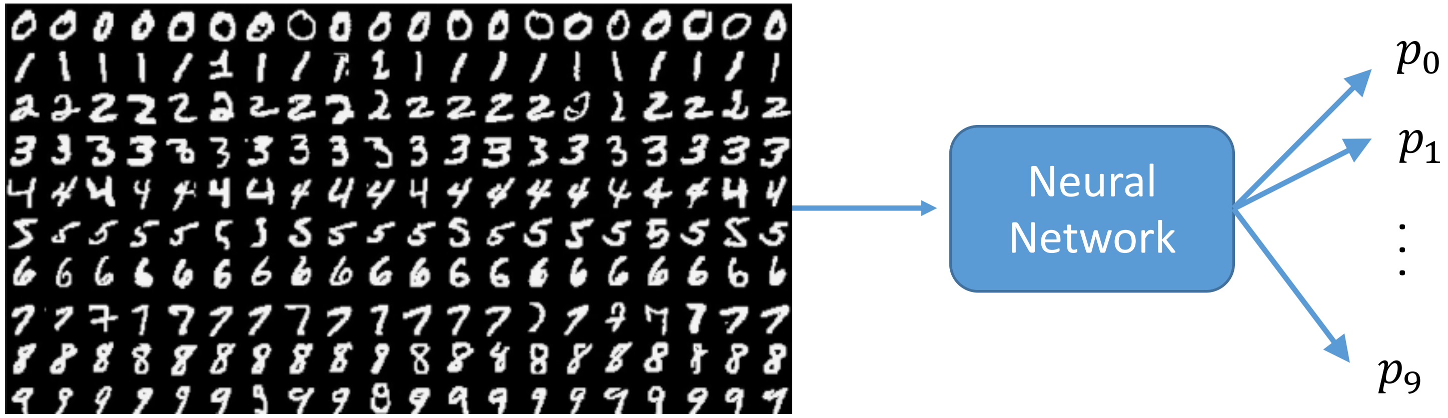
$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \le \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

- Idea: approximate the maximization with a concave function

Training robust neural networks

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \leq \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

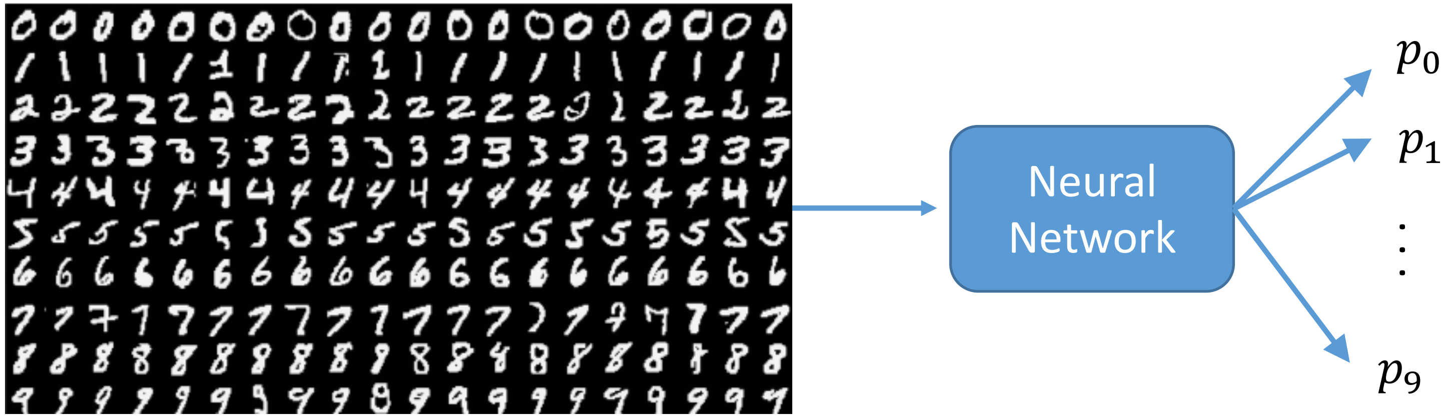
- Idea: approximate the maximization with a concave function



Training robust neural networks

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \le \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

- Idea: approximate the maximization with a concave function

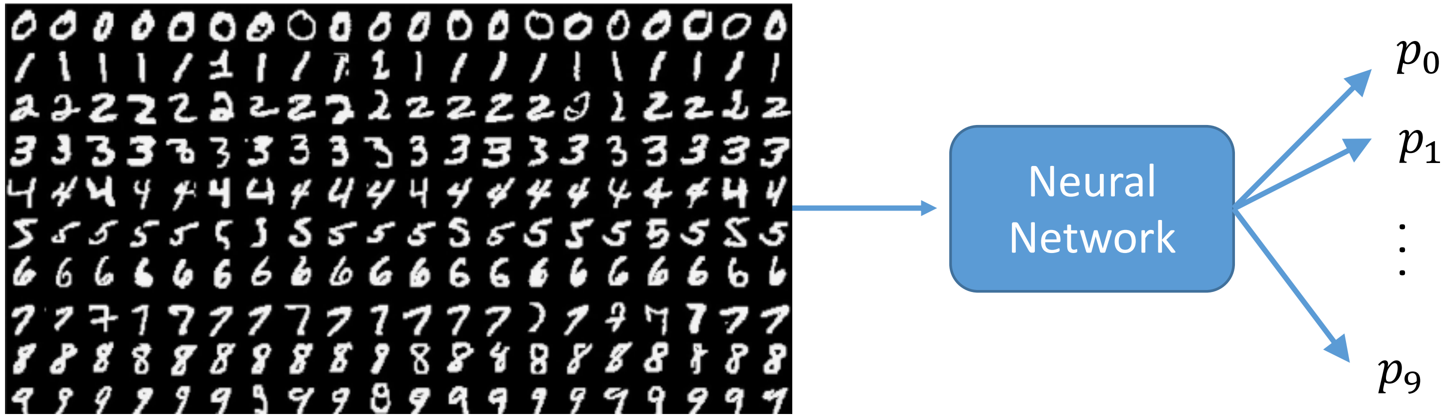


$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max \left\{ \ell\big(\mathbf{w}, \mathbf{x}_i + d_0(\mathbf{w}, \mathbf{x}_i)\big), \ldots, \ell\big(\mathbf{w}, \mathbf{x}_i + d_9(\mathbf{w}, \mathbf{x}_i)\big) \right\}$$

Training robust neural networks

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\| \leq \epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

- Idea: approximate the maximization with a concave function



$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max \left\{ \ell\left(\mathbf{w}, \mathbf{x}_i + d_0(\mathbf{w}, \mathbf{x}_i)\right), \ldots, \ell\left(\mathbf{w}, \mathbf{x}_i + d_9(\mathbf{w}, \mathbf{x}_i)\right) \right\}$$

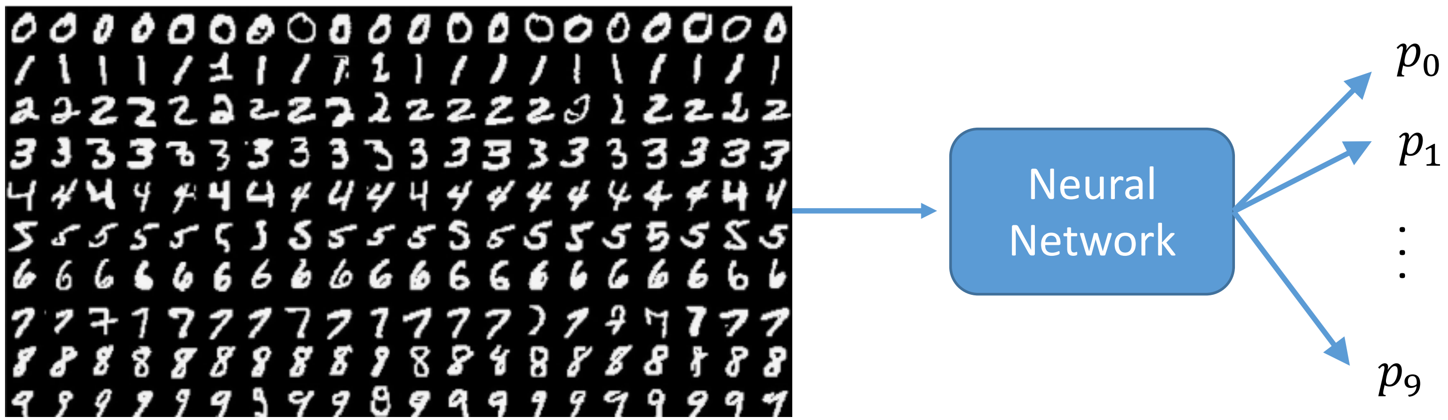
$\nabla_{\mathbf{x}} p_0(\mathbf{w}, \mathbf{x}_i) - \nabla_{\mathbf{x}} p_c(\mathbf{w}, \mathbf{x}_i)$

$\nabla_{\mathbf{x}} p_9(\mathbf{w}, \mathbf{x}_i) - \nabla_{\mathbf{x}} p_c(\mathbf{w}, \mathbf{x}_i)$

Training robust neural networks

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\|\leq\epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

➢ Idea: approximate the maximization with a concave function



$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max \left\{ \ell\big(\mathbf{w}, \mathbf{x}_i + d_0(\mathbf{w}, \mathbf{x}_i)\big), \ldots, \ell\big(\mathbf{w}, \mathbf{x}_i + d_9(\mathbf{w}, \mathbf{x}_i)\big) \right\}$$

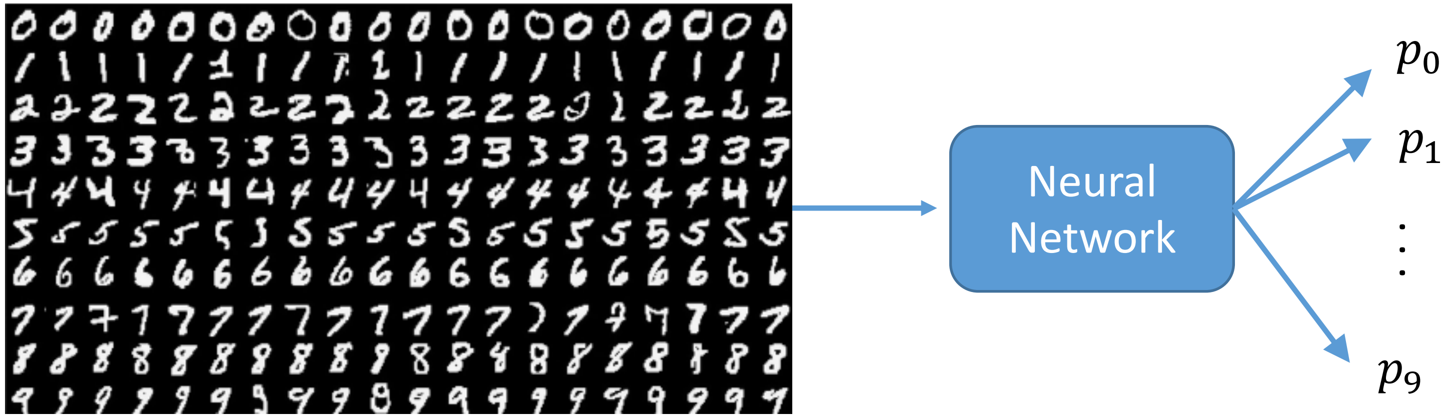
$$\nabla_{\mathbf{x}} p_0(\mathbf{w}, \mathbf{x}_i) - \nabla_{\mathbf{x}} p_c(\mathbf{w}, \mathbf{x}_i) \qquad \nabla_{\mathbf{x}} p_9(\mathbf{w}, \mathbf{x}_i) - \nabla_{\mathbf{x}} p_c(\mathbf{w}, \mathbf{x}_i)$$

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left[\max_{\mathbf{t}\in\mathcal{P}} \sum_{k=0}^{9} t_k \ell\big(\mathbf{w}, \mathbf{x}_i + d_k(\mathbf{w}, \mathbf{x}_i)\big) \right]$$

Training robust neural networks

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max_{\|\boldsymbol{\delta}\|\le\epsilon} \ell(\mathbf{w}, \mathbf{x}_i + \boldsymbol{\delta})$$

- Idea: approximate the maximization with a concave function



$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max \left\{ \ell\big(\mathbf{w}, \mathbf{x}_i + d_0(\mathbf{w}, \mathbf{x}_i)\big), \ldots, \ell\big(\mathbf{w}, \mathbf{x}_i + d_9(\mathbf{w}, \mathbf{x}_i)\big) \right\}$$

$$\nabla_{\mathbf{x}} p_0(\mathbf{w}, \mathbf{x}_i) - \nabla_{\mathbf{x}} p_c(\mathbf{w}, \mathbf{x}_i) \qquad \nabla_{\mathbf{x}} p_9(\mathbf{w}, \mathbf{x}_i) - \nabla_{\mathbf{x}} p_c(\mathbf{w}, \mathbf{x}_i)$$

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left[\max_{\mathbf{t}\in\mathcal{P}} \sum_{k=0}^{9} t_k \ell\big(\mathbf{w}, \mathbf{x}_i + d_k(\mathbf{w}, \mathbf{x}_i)\big) \right]$$

Non-convex in ***w***, but concave in ***t***

Numerical results

[1] Madry et al. "Towards deep learning models resistant to adversarial attacks." *ICLR 2017*

[2] Zhang et al. “Theoretically principled trade-o between robustness and accuracy” *ICML 2019*.

Numerical results

[1] Madry et al. "Towards deep learning models resistant to adversarial attacks." *ICLR 2017*

[2] Zhang et al. “Theoretically principled trade-o between robustness and accuracy” *ICML 2019*.

→ No theoretical convergence guarantee

Numerical results

[1] Madry et al. "Towards deep learning models resistant to adversarial attacks." *ICLR 2017*

[2] Zhang et al. "Theoretically principled trade-o between robustness and accuracy" *ICML 2019*.

→ No theoretical convergence guarantee

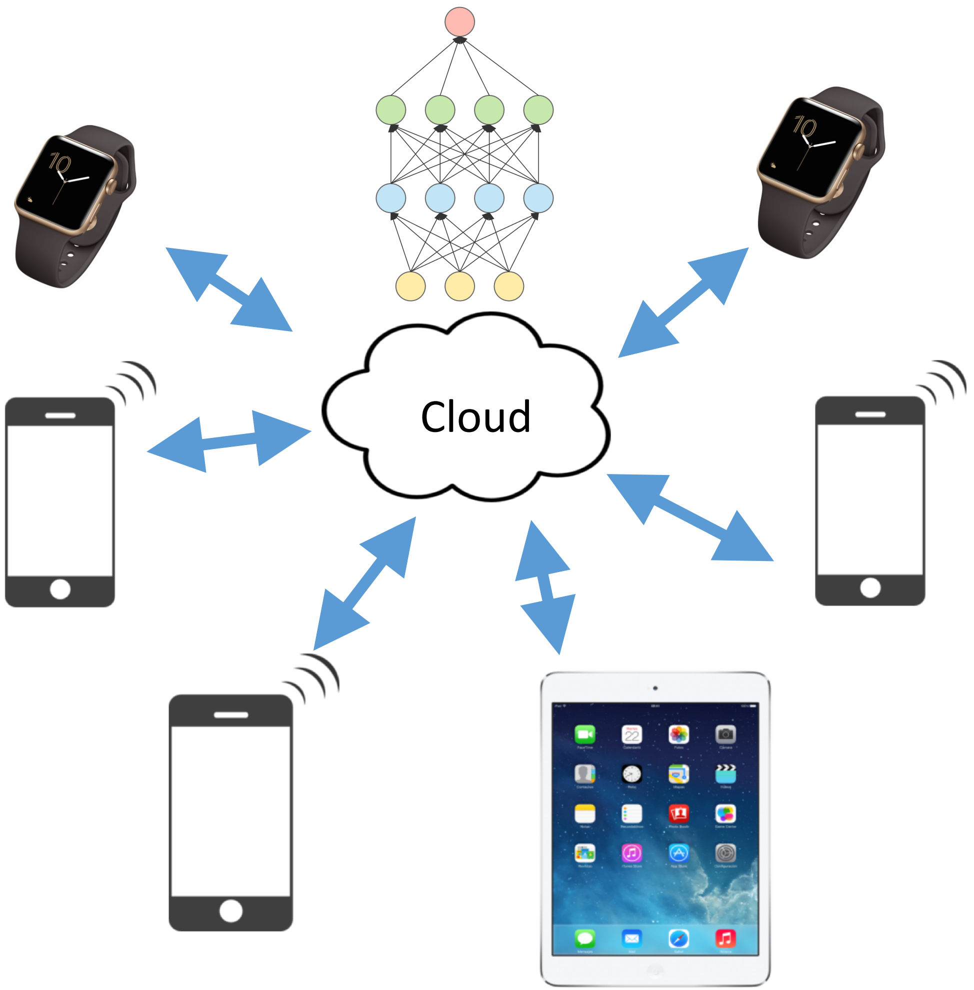
	Regular Performance	Performance under FGSM attack			Performance under PGD attack		
---	----	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$
[1]	**98.58%**	96.09%	94.82%	89.84%	94.64%	91.41%	78.67%
[2]	97.21%	96.19%	96.17%	96.14%	95.01%	94.36%	94.11%
Proposed	98.20%	**97.04%**	**96.66%**	**96.23%**	**96.00%**	**95.17%**	**94.22%**

FGSM attack: Goodfellow, Shlens, and Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572* (2014).

PGD attack: Kurakin, Goodfellow, and Bengio, "Adversarial Machine Learning" at Scale, ICLR 2016.

Min-max and fairness among users in learning

➢ Designing a machine learning model that works for everyone

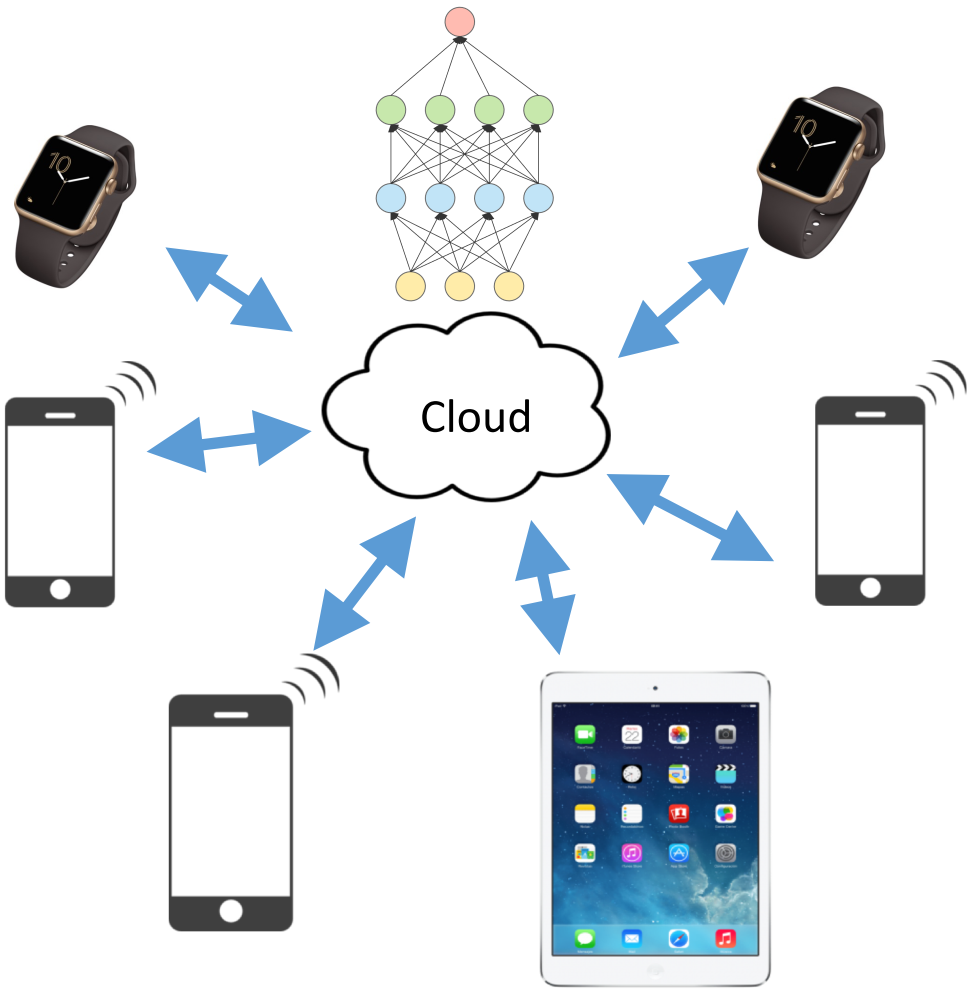


Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

Min-max and fairness among users in learning

- Designing a machine learning model that works for everyone

$$\min_{\mathbf{w}} \; \max\{\ell_1(\mathbf{w}), \ldots, \ell_k(\mathbf{w})\}$$



Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

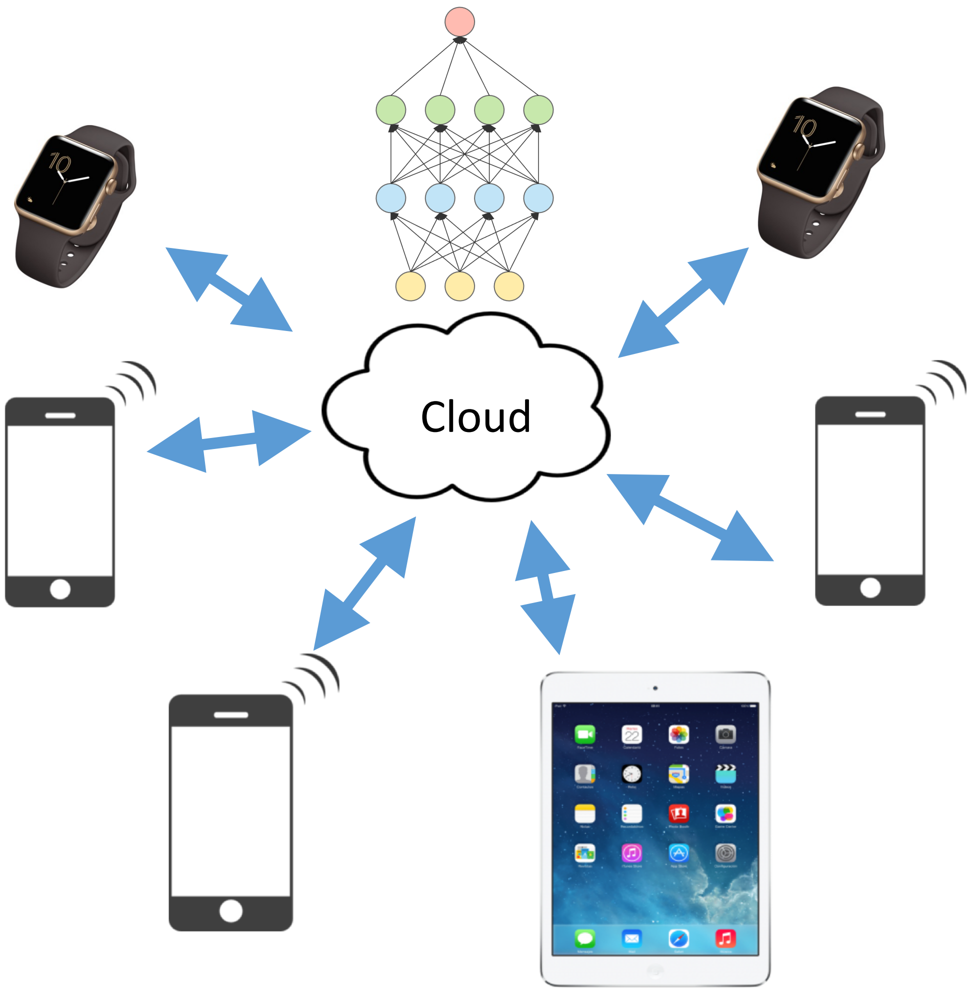
Min-max and fairness among users in learning

- Designing a machine learning model that works for everyone

$$\min_{\mathbf{w}} \quad \max\{\ell_1(\mathbf{w}), \ldots, \ell_k(\mathbf{w})\}$$

$$\downarrow$$

$$\min_{\mathbf{w}} \quad \max_{\mathbf{t} \in \mathcal{P}} \sum_{i=1}^{k} t_i \ell_i(\mathbf{w})$$

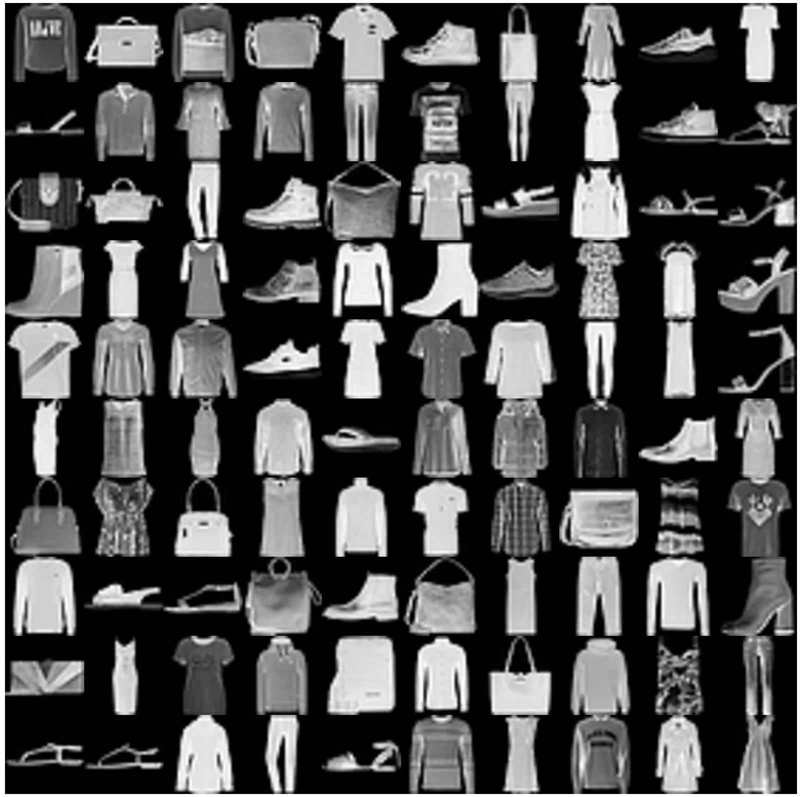


Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

Numerical results

➢ Fair performance among different categories of data

$$\min_{\mathbf{w}} \ \max\{\ell_1(\mathbf{w}), \ell_2(\mathbf{w}), \ell_3(\mathbf{w})\}$$



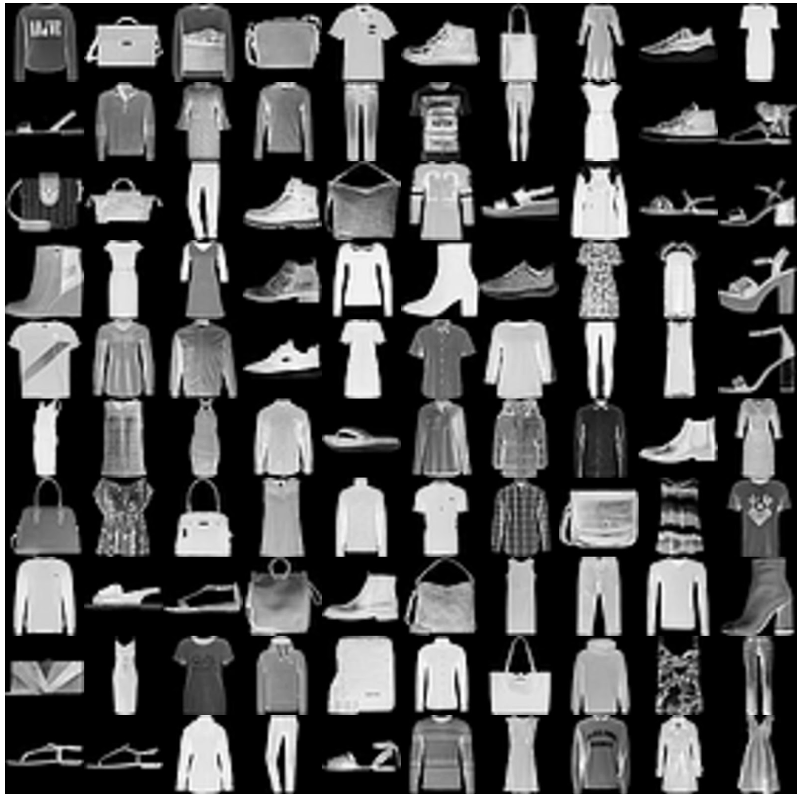
Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

Numerical results

- Fair performance among different categories of data

$$\min_{\mathbf{w}} \quad \max\{\ell_1(\mathbf{w}), \ell_2(\mathbf{w}), \ell_3(\mathbf{w})\}$$

$$\min_{\mathbf{w}} \quad \max_{\mathbf{t}\in\mathcal{P}} \quad \sum_{i=1}^{3} t_i \ell_i(\mathbf{w})$$



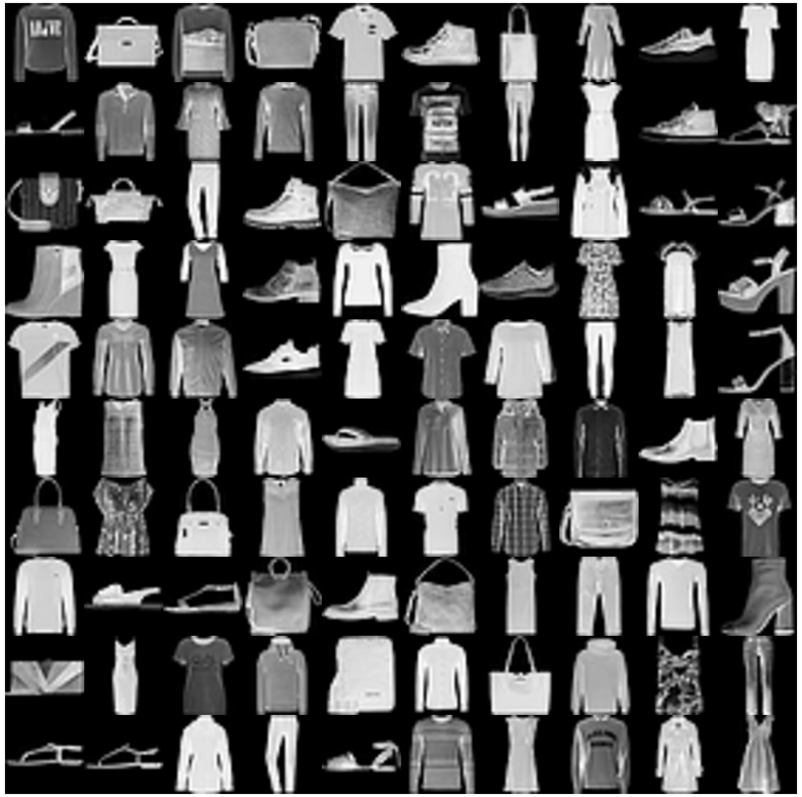
Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

Numerical results

- Fair performance among different categories of data

$$\min_{\mathbf{w}} \quad \max\{\ell_1(\mathbf{w}), \ell_2(\mathbf{w}), \ell_3(\mathbf{w})\}$$

$$\min_{\mathbf{w}} \quad \max_{\mathbf{t} \in \mathcal{P}} \quad \sum_{i=1}^{3} t_i \ell_i(\mathbf{w})$$



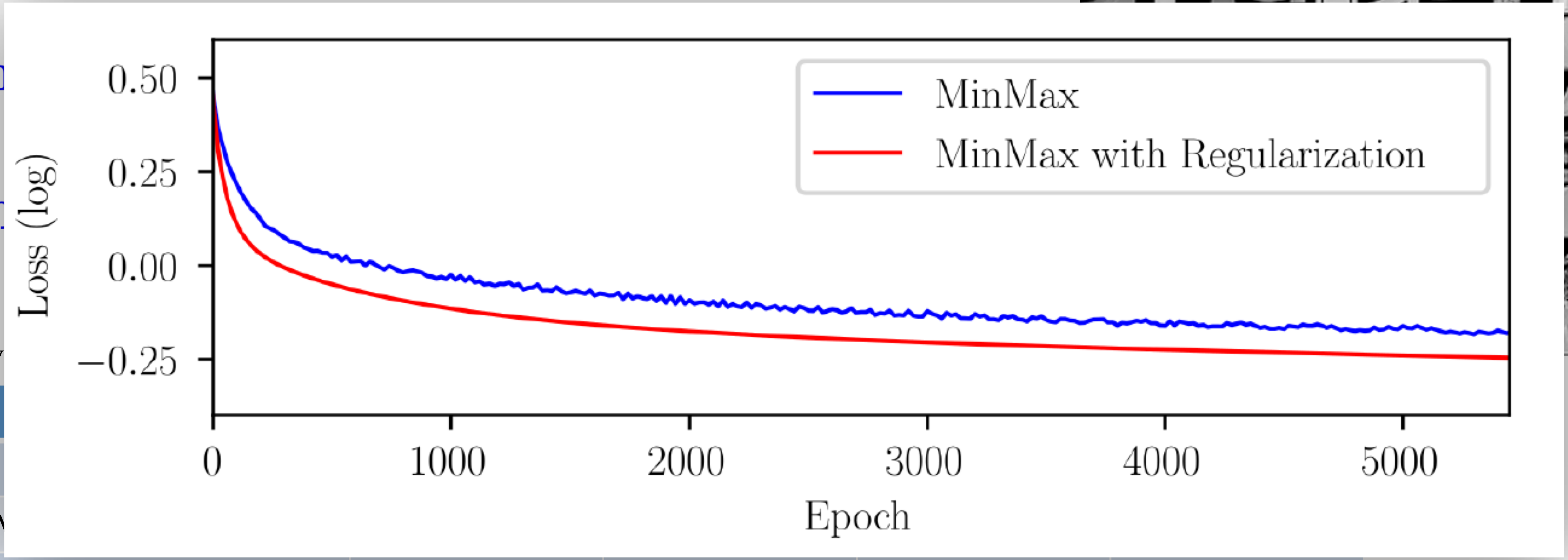
Average performance over 100 training:

	T-shirt/Top	Coat	Shirt
Normal Training	84.1 ±1.8%	86.4 ±2.1%	**70.6 ±3.7%**
Min-max no regularizer	75.4 ±1.5%	**71.6 ±3.0%**	73.3 ±1.9%
Min-max with regularizer	76.3 ±1.4%	**73.9 ±2.8%**	74.8 ±1.6%

- Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn, "Solving a class of non-convex min-max games using iterative first order methods," arXiv:1902.08297, *accepted in NeurIPS 2019*.
- Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

Numerical results

- Fair performance among different categories of data



Min-max with regularizer	76.3 ±1.4%	**73.9 ±2.8%**	74.8 ±1.6%	**73.4 ±2.4%**

Mohri et al. "Agnostic federated learning." *arXiv:1902.00146* (2019).

Min-max and fairness in machine learning

- Discriminatory behaviors in human decisions and machine learning models:
  - [Bickel et al., 1975]: Sex bias in graduate admissions in Berkeley
  - [Datta et al. 2015]: Google's online advertising showed high-income jobs ads to men more than to women.
  - [Sweeney 2013]: ads for arrest records shows up on searches for distinctively black names.
  - Amazon's recruitment engine has bias against women*

* https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Min-max and fairness in machine learning

- Discriminatory behaviors in human decisions and machine learning models:
  - [Bickel et al., 1975]: Sex bias in graduate admissions in Berkeley
  - [Datta et al. 2015]: Google's online advertising showed high-income jobs ads to men more than to women.
  - [Sweeney 2013]: ads for arrest records shows up on searches for distinctively black names.
  - Amazon's recruitment engine has bias against women*
- Different reasons such as old data human bias

* https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Min-max and fairness in machine learning

- Discriminatory behaviors in human decisions and machine learning models:
  - [Bickel et al., 1975]: Sex bias in graduate admissions in Berkeley
  - [Datta et al. 2015]: Google's online advertising showed high-income jobs ads to men more than to women.
  - [Sweeney 2013]: ads for arrest records shows up on searches for distinctively black names.
  - Amazon's recruitment engine has bias against women*

- Different reasons such as old data human bias

- Regulated domains: employment, housing, education, …

* https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Min-max and fairness in machine learning

- Discriminatory behaviors in human decisions and machine learning models:
  - [Bickel et al., 1975]: Sex bias in graduate admissions in Berkeley
  - [Datta et al. 2015]: Google's online advertising showed high-income jobs ads to men more than to women.
  - [Sweeney 2013]: ads for arrest records shows up on searches for distinctively black names.
  - Amazon's recruitment engine has bias against women*

- Different reasons such as old data human bias

- Regulated domains: employment, housing, education, …

- Designing *discrimination-free* machine learning models

* https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Min-max and fairness in machine learning

- Discriminatory behaviors in human decisions and machine learning models:
  - [Bickel et al., 1975]: Sex bias in graduate admissions in Berkeley
  - [Datta et al. 2015]: Google's online advertising showed high-income jobs ads to men more than to women.
  - [Sweeney 2013]: ads for arrest records shows up on searches for distinctively black names.
  - Amazon's recruitment engine has bias against women*
- Different reasons such as old data human bias
- Regulated domains: employment, housing, education, …
- Designing *discrimination-free* machine learning models

* https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Min-max and fairness in machine learning

- Discriminatory behaviors in human decisions and machine learning models:
  - [Bickel et al., 1975]: Sex bias in graduate admissions in Berkeley
  - [Datta et al. 2015]: Google's online advertising showed high-income jobs ads to men more than to women.
  - [Sweeney 2013]: ads for arrest records shows up on searches for distinctively black names.
  - Amazon's recruitment engine has bias against women*
- Different reasons such as old data human bias
- Regulated domains: employment, housing, education, …
- Designing *discrimination-free* machine learning models
- **Goals**:
  - Make $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\hat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

Protected feature

$$(\mathbf{x}, \mathbf{s}) \longrightarrow \boxed{\text{Prediction}} \longrightarrow \hat{\mathbf{y}} \approx \mathbf{y}$$

* https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Min-max and fairness in machine learning

- **Goals**:
  - Make $\hat{y}$ and s independent
  - Keep $\hat{y}$ close to y

Protected feature

$$(\mathbf{x}, \mathbf{s}) \longrightarrow \boxed{\text{Prediction}} \longrightarrow \hat{\mathbf{y}} \approx \mathbf{y}$$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

Min-max and fairness in machine learning

- **Goals**:
  - Make $\widehat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\widehat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

Protected feature

$$(\mathbf{x}, \mathbf{s}) \longrightarrow \boxed{\text{Prediction}} \longrightarrow \hat{\mathbf{y}} \approx \mathbf{y}$$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

Keep classification error small

Min-max and fairness in machine learning

- **Goals**:
  - Make $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\hat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

Protected feature

$$(\mathbf{x}, \mathbf{s}) \longrightarrow \boxed{\text{Prediction}} \longrightarrow \hat{\mathbf{y}} \approx \mathbf{y}$$

$$\min_{\boldsymbol{\theta}} \quad \underbrace{\mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))]}_{\text{Keep classification error small}} + \underbrace{\lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})}_{\text{Imposing fairness}}$$

Min-max and fairness in machine learning

- **Goals**:
  - Make $\hat{y}$ and s independent
  - Keep $\hat{y}$ close to y

Protected feature

$$(\mathbf{x}, \mathbf{s}) \longrightarrow \boxed{\text{Prediction}} \longrightarrow \hat{\mathbf{y}} \approx \mathbf{y}$$

$$\min_{\boldsymbol{\theta}} \quad \underbrace{\mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))]}_{\text{Keep classification error small}} + \underbrace{\lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})}_{\text{Imposing fairness}}$$

Different correlation measures: Mutual information [*Kamishima et al. 2011*], false positive/negative rates [*Bechavod & Ligett 2017*], equalized odds [*Donini et al. 2018*], Pearson correlation coefficient [*Zaffar et al. 2015, 2017*], Hilbert Schmidt independence criterion [*Pérez-Suay et al. 2017*]

Min-max and fairness in machine learning

- **Goals**:
  - Make $\hat{y}$ and s independent
  - Keep $\hat{y}$ close to y

Protected feature

$$(\mathbf{x}, \mathbf{s}) \longrightarrow \boxed{\text{Prediction}} \longrightarrow \hat{\mathbf{y}} \approx \mathbf{y}$$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

Keep classification error small

Imposing fairness

Different correlation measures: Mutual information [*Kamishima et al. 2011*], false positive/negative rates [*Bechavod & Ligett 2017*], equalized odds [*Donini et al. 2018*], Pearson correlation coefficient [*Zaffar et al. 2015, 2017*], Hilbert Schmidt independence criterion [*Pérez-Suay et al. 2017*]

- Either do not have convergence guarantees or cannot guarantee statistical independence

Rényi Fair Inference

- **Goals**:
  - Make $\widehat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\widehat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda\rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

Rényi Fair Inference

- **Goals**:
  - Make $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\hat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

- **Use Rényi (maximal) correlation**

$$\rho(A, B) = \sup_{f,g} \quad \mathbb{E}[f(A)g(B)]$$

$$\text{s.t.} \quad \mathbb{E}[f(A)] = \mathbb{E}[g(B)] = 0, \quad \mathbb{E}[f^2(A)] = \mathbb{E}[g^2(B)] = 1$$

Rényi Fair Inference

- **Goals**:
  - Make $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\hat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

- **Use Rényi (maximal) correlation**

$$\rho(A,B) = \sup_{f,g} \quad \mathbb{E}[f(A)g(B)]$$

$$\text{s.t.} \quad \mathbb{E}[f(A)] = \mathbb{E}[g(B)] = 0, \;\; \mathbb{E}[f^2(A)] = \mathbb{E}[g^2(B)] = 1$$

- **Rényi Fair Inference [*Bahrlouei et al 2019*]**

$$\min_{\boldsymbol{\theta}} \max_{f,g} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \mathbb{E}[f(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))g(\mathbf{s})]$$

$$\text{s.t.} \quad \mathbb{E}[f(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] = \mathbb{E}[g(\mathbf{s})] = 0, \;\; \mathbb{E}[f^2(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] = \mathbb{E}[g^2(\mathbf{s})] = 1$$

Rényi Fair Inference

- **Goals**:
  - Make $\hat{\boldsymbol{y}}$ and $\boldsymbol{s}$ independent
  - Keep $\hat{\boldsymbol{y}}$ close to $\boldsymbol{y}$

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

- **Use Rényi (maximal) correlation**

$$\rho(A, B) = \sup_{f,g} \quad \mathbb{E}[f(A)g(B)]$$
$$\text{s.t.} \quad \mathbb{E}[f(A)] = \mathbb{E}[g(B)] = 0, \ \ \mathbb{E}[f^2(A)] = \mathbb{E}[g^2(B)] = 1$$

- **Rényi Fair Inference [*Bahrlouei et al 2019*]**

$$\min_{\boldsymbol{\theta}} \max_{f,g} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \mathbb{E}[f(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))g(\mathbf{s})]$$
$$\text{s.t.} \quad \mathbb{E}[f(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] = \mathbb{E}[g(\mathbf{s})] = 0, \ \ \mathbb{E}[f^2(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] = \mathbb{E}[g^2(\mathbf{s})] = 1$$

- Can be solved for discrete random variables

Rényi Fair Inference

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

Theorem [Witsenhausen 1975]: In the discrete case, Rényi correlation is the second largest singular value of the matrix $Q = [q_{ij}]$ where $q_{ij} = \frac{P(s_i, y_j)}{\sqrt{P(s_i)\, P(y_j)}}$

$$\rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})^2 = \max_{\mathbf{v} \perp \mathbf{v}_1, \|\mathbf{v}\| \leq 1} \mathbf{v}^T \mathbf{Q} \mathbf{Q}^T \mathbf{v}$$

Rényi Fair Inference

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}[\ell(\mathbf{y}, \hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}))] + \lambda \rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})$$

Theorem [Witsenhausen 1975]: In the discrete case, Rényi correlation is the second largest singular value of the matrix $Q = [q_{ij}]$ where $q_{ij} = \frac{P(s_i, y_j)}{\sqrt{P(s_i)\, P(y_j)}}$

$$\rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})^2 = \max_{\mathbf{v} \perp \mathbf{v}_1, \|\mathbf{v}\| \leq 1} \mathbf{v}^T \mathbf{Q} \mathbf{Q}^T \mathbf{v}$$

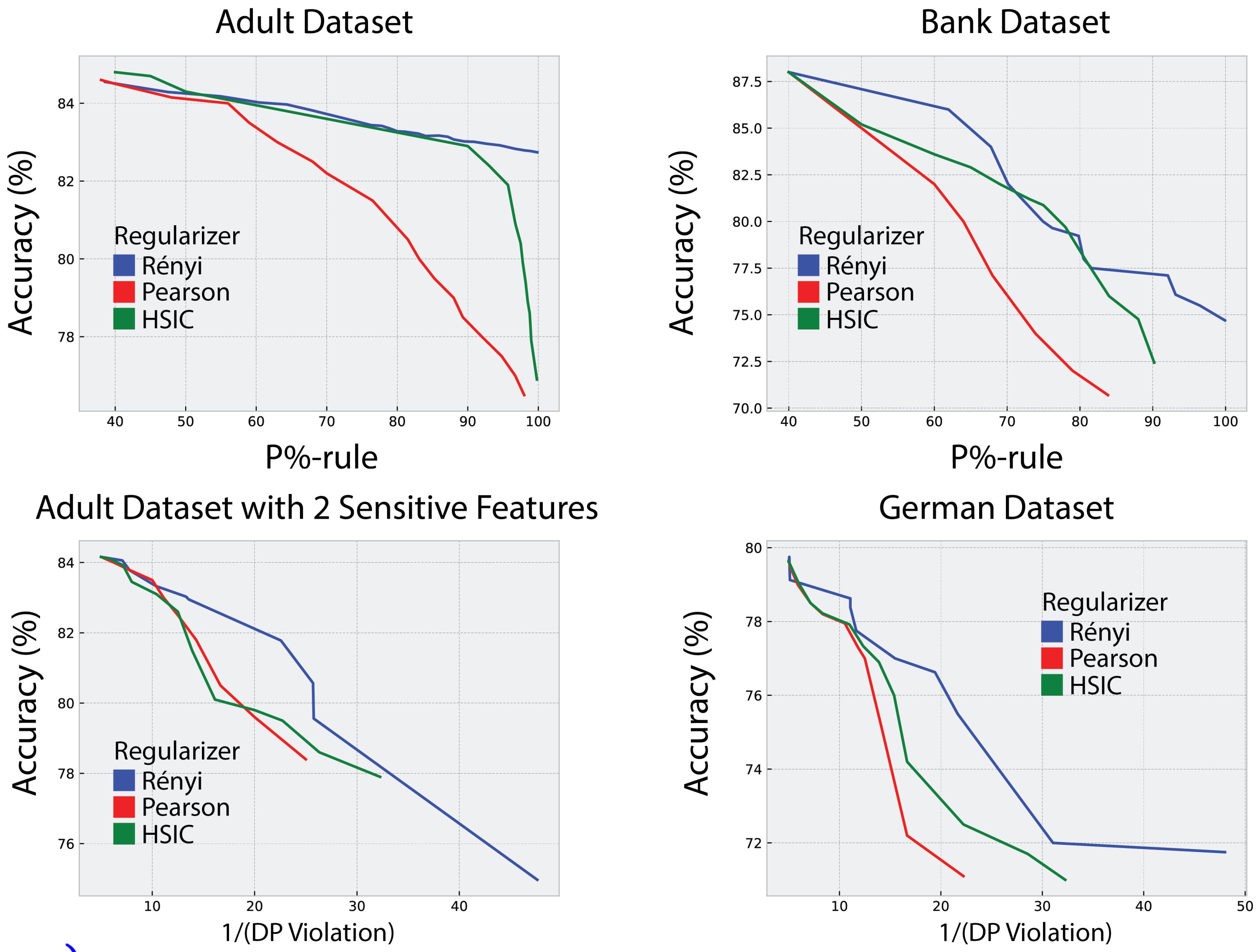
Theorem [Baharlouei, Nouiehed, Razaviyayn 2019]: When **s** is binary, we have

$$\rho(\hat{\mathbf{y}}_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{s})^2 = 1 - \frac{\min_{\mathbf{w}} \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{y}}_{\boldsymbol{\theta}} - \mathbf{s}]}{\mathbb{P}(\mathbf{s} = 1)\mathbb{P}(\mathbf{s} = 0)}$$

- PL case, can be solved effiiently

Numerical Experiments

- Pearson correlation coefficient
  - [*Zaffar et al. 2015, 2017*]
- Hilbert Schmidt Independence Criterion
  - [*Pérez-Suay et al. 2017*]
- Rényi Fair Inference
  - [*Baharlouei et al. 2019*]

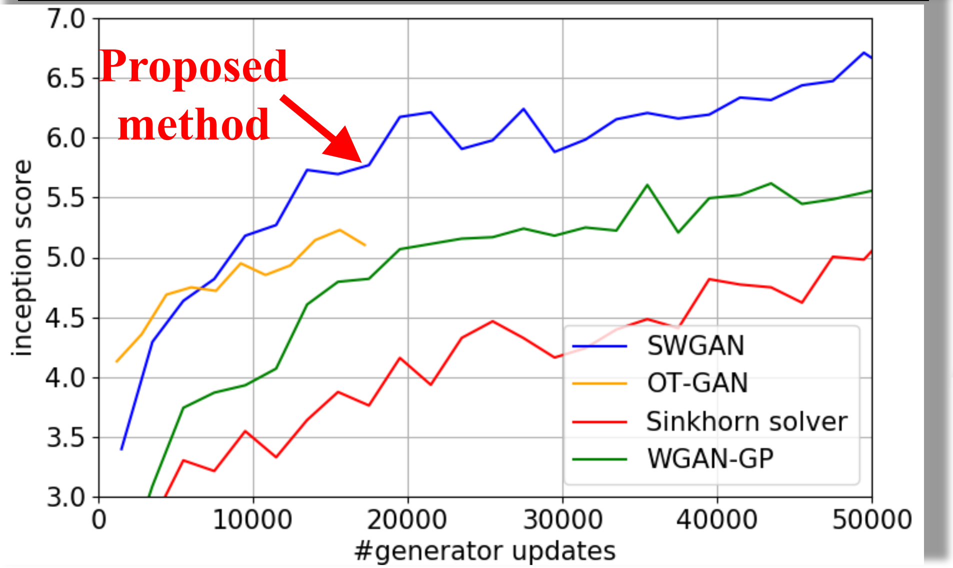


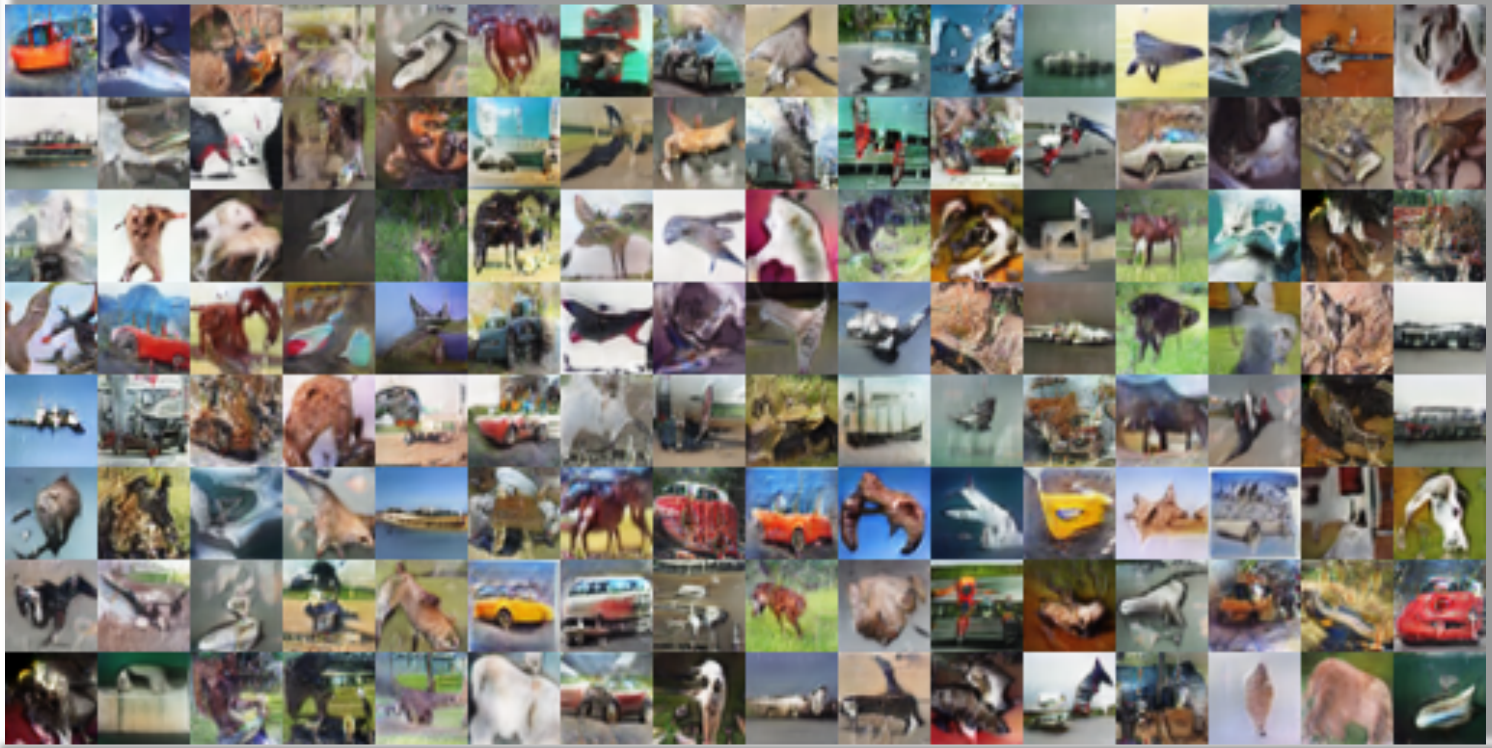
$$p\% = \min\left\{\frac{\mathbb{P}(\hat{Y}=1|S=1)}{\mathbb{P}(\hat{Y}=1|S=0)}, \frac{\mathbb{P}(\hat{Y}=1|S=0)}{\mathbb{P}(\hat{Y}=1|S=1)}\right\}$$

$$\text{DP Violation} = \max_{a,b} |\mathbb{P}(\hat{Y}=1|S=a) - \mathbb{P}(\hat{Y}=1|S=b)|$$

Extension to stochastic setting and applications in training GANs

- Sanjabi, Ba, Razaviyayn, Lee. “On the convergence and robustness of training GANs with regularized optimal transport,” *Neurips* 2018





Summary

- Non-convex min-max problems are challenging

- Special cases could be solved *efficiently*

- These problems appear in many applications
  - Robust learning
  - GANs
  - Fair learning
  - and many more…

Future work

- This is just a first step

- How far we can go into the non-convex world? Providing upper- lower- bounds?

Future work

- This is just a first step

- How far we can go into the non-convex world? Providing upper- lower- bounds?

- How far we can go beyond first-order stationarity/Nash equilibrium concept?

$$\min_{-1\leq\theta\leq1} \max_{-2\leq\alpha\leq2} -\theta^2 + \alpha^2 + 4\theta\alpha$$

A long history

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad f(\boldsymbol{\theta}, \boldsymbol{\alpha})$$

- Using monotone operator:
  - [Sibony'70], [Korpelevich'76], [Nemirovski'04], [Martinet'70], [Rockafellar'76], [Di-Sun'99], [Juditsky-Nemirovsky'16], …

- Weak Monotonicity
  - [Davis-Grimmer'17, Davis-Drusvyatskiy'18, Zhang-He'18, Lin et al'18], …

- More general VI's
  - [Facchinei-Pang'03], [Monteiro-Svaiter'10], [Nesterov'07], [Dong-Lan'14], …

- Stochastic VI's
  - [Juditsky-Nemirovski-Tauvel '11], [Koshal-Nedic-Shanbag'13], [Rosasco-Villa-Vũ'14], [Balamurugan-Bach'16], …

- Bilinear convex-concave
  - [Arrow-Hurwicz-Uzawa'58, Zhu-Chan'08], [Chambolle-Pock'11&16], [Chen-Lan-Ouyang'14], [Dong-Lan'14, Chambolle et al'17], [Wang-Xiao'17], …

- Convex-Concave saddle points
  - [Tseng'08], [He and Monterio'17], [Hamedani-Jalilzadeh-Aybat-Shanbhag'18], …

Other recent results in non-convex min-max regimes

- [Lu, Tsaknakis, and Hong 2019]
- [Gidel, Hemmat, Pezeshki, Huang, Lepriol, Lacoste-Julien, and Mitligkas 2018]
- [Gidel, Jebara, and Lacoste-Julien 2018]
- [Lu, Tsaknakis, Hong, Chen 2019]
- [Hameani, Jalilzadeh, Aybat, Shanbhag 2018]
- [Rafique, Liu, Lin, and Yang 2018]
- [Sinha, Namkoong, and Duchi 2018]
- [Thekumparampil, Jain, Netrapalli, and Oh 2019]
- [Jin, Netrapalli, and Jordan 2019]
- [Lin, Jin, Jordan 2019]
- [Letcher, Balduzzi, Racaniere, Martens, Foerster, Tuyls, and Graepel 2019]
- [Lin, Liu, Rafique, Yang 2018]
- [Mescheder, Geiger, and Nowozin 2018]
- [Mokhtari, Ozdaglar, Pattathil 2019]
- [Daskalakis, Ilyas, Syrgkanis, and Zeng 2018]
- [Daskalakis and Panageas 2018]
- [Daskalakis and Panageas 2019]

- And many other recent works…

References

- Jong-Shi Pang and Meisam Razaviyayn, “A unified distributed algorithm for non-cooperative games,” book chapter in *Big Data over Networks, 2016*.
- Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D. Lee. “On the convergence and robustness of training GANs with regularized optimal transport,” *NeurIPS* 2018.
- Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn, “Solving a class of non-convex min-max games using iterative first order methods,” arXiv:1902.08297, *NeurIPS 2019*.
- Sina Baharlouei, Maher Nouiehed, and Meisam Razaviyayn. "Rènyi Fair Inference,” *Submitted to ICLR 2019, arXiv 1906.12005*.

- Codes are available at: **Optimization for Data-Driven Science (ODDS)** lab, GitHub account
  - https://github.com/optimization-for-data-driven-science

Questions