When Deep Learning Meets Nontrivial Constraints

MMLS2023

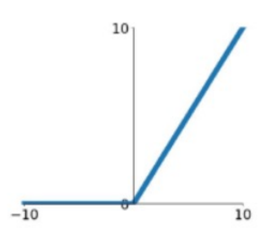
Buyun Liang, Ryan D Devera, Hengyue Liang, Wenjie Zhang, Tim Mitchell, Ju Sun

University of Minnesota

1. Motivating examples & methods

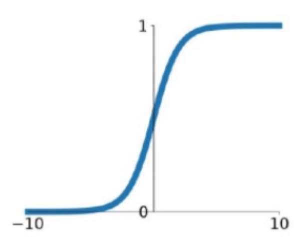
(Constrained deep learning: CDL)

1.1 Embedding constraints into DL models



ReLU
(Rectified Linear Unit)

Nonnegativity



Sigmoid

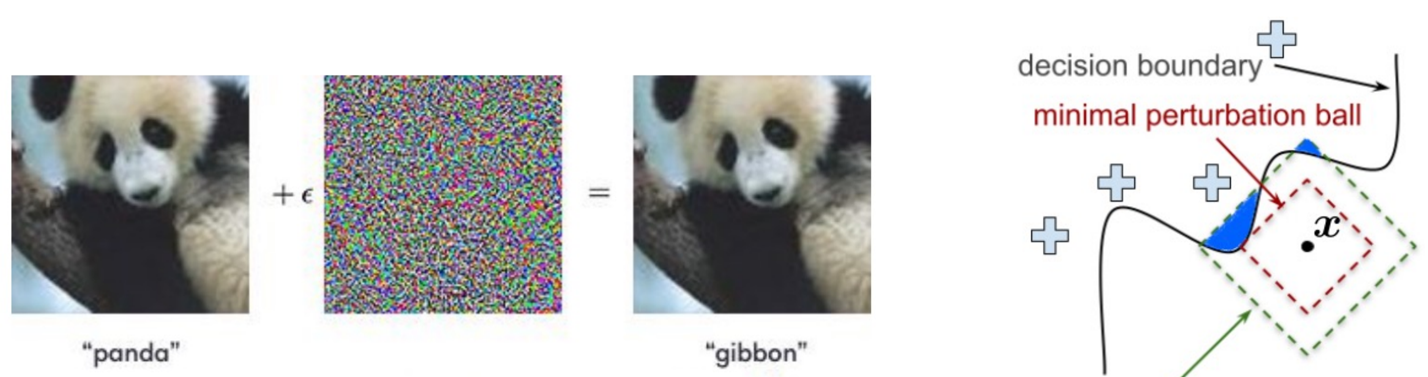
[0, 1]

$$z \mapsto \left[\frac{e^{z_1}}{\sum_j e^{z_j}}, \ldots, \frac{e^{z_p}}{\sum_j e^{z_j}}\right]^{\intercal}$$

Softmax

Nonnegativity and summed to 1

1.2 Robustness evaluation



$\boldsymbol{x}$ $\quad \boldsymbol{\delta} \quad$ $\boldsymbol{x}'$

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right) \quad \text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \le \epsilon, \quad \boldsymbol{x}' \in [0,1]^n$$

Maximum adversarial loss

$$\min_{\boldsymbol{x}'} d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \quad \text{s.t. } \max_{i \neq y} f_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}') \ge f_{\boldsymbol{\theta}}^{y}(\boldsymbol{x}'), \quad \boldsymbol{x}' \in [0,1]^n$$

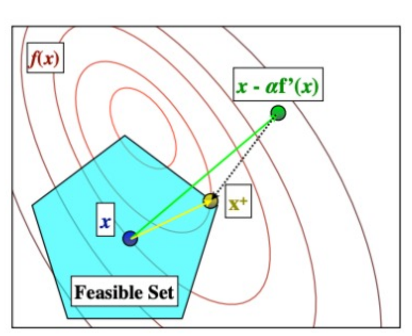
Minimum distortion radius

- **Projected gradient descent**

$$\min_{\mathbf{x} \in \mathcal{Q}} f(\mathbf{x})$$

$$\mathbf{x}_{k+1} = P_{\mathcal{Q}}\Big(\mathbf{x}_k - \underbrace{\alpha_k}_{\text{Step size}} \nabla f(\mathbf{x}_k)\Big)$$

$$P_{\mathcal{Q}}(\mathbf{x}_0) = \arg\min_{\mathbf{x} \in \mathcal{Q}} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2 \quad \text{Projection operator}$$



Key hyperparameters:
(1) step size
(2) iteration number

Problem: tricky to set **iteration number** & **step size**
i.e., tricky to decide where to stop

- **Penalty method**

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2 \quad \text{Perceptual distance}$$

$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\,\widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x})\,]$$

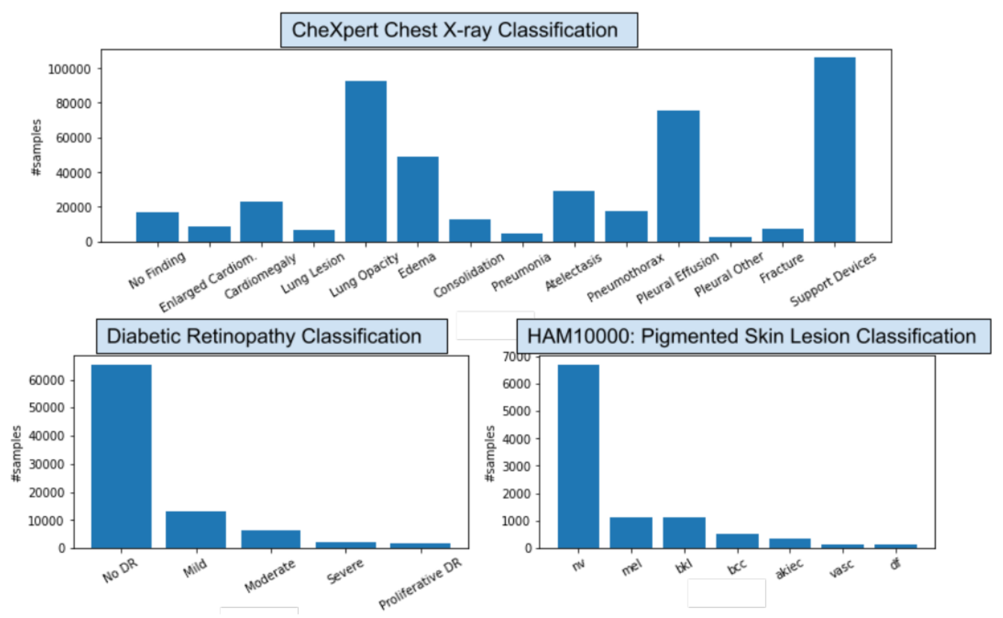
Projection onto the constraint is complicated

$$\max_{\widetilde{\mathbf{x}}} \quad \mathcal{L}(f(\widetilde{\mathbf{x}}), y) - \lambda \max\left(0, \|\phi(\widetilde{\mathbf{x}}) - \phi(\mathbf{x})\|_2 - \epsilon\right)$$

Solve it for each fixed λ and then increase λ

Problem: large **constraint violation** or **suboptimal** solution

1.3 Imbalanced learning



Class imbalance in healthcare datasets

	Predicted POS	Predicted NEG
POS	70	30
NEG	1000	9000

Accuracy:	9070/10100 =	0.898
True Positive Rate (Sensitivity, Recall):		0.7
True Negative Rate (Specificity):		0.9
Balanced Accuracy:	(0.7 + 0.9)/2 =	0.80
Precision (POS):	70/1070 =	0.065
F1 Score:	2*0.065*0.7/(0.065 + 0.7) =	0.119

Reliable evaluation in imbalanced learning: Precision needed

Typical learning objective $\quad \min_{f \in \mathcal{H}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{D}_{\boldsymbol{x},\boldsymbol{y}}} \mathbb{1}\{\boldsymbol{y} \neq f(\boldsymbol{x})\}$ $\quad$ Accuracy maximization

$$\begin{aligned}
&\textbf{fix precision, optimize recall (FPOR):} && \max_{\boldsymbol{\theta},t}\ \mathrm{recall}(f_{\boldsymbol{\theta}}, t) \quad \text{s.t. } \mathrm{precision}(f_{\boldsymbol{\theta}}, t) \ge \alpha, \\
&\textbf{fix recall, optimize precision (FROP):} && \max_{\boldsymbol{\theta},t}\ \mathrm{precision}_t \quad \text{s.t. } \mathrm{recall}(f_{\boldsymbol{\theta}}, t) \ge \alpha, \\
&\textbf{optimize } F_\beta \textbf{ score (OFBS):} && \max_{\boldsymbol{\theta},t}\ F_\beta(f_{\boldsymbol{\theta}}, t), \\
&\textbf{optimize AP (OAP):} && \max_{\boldsymbol{\theta}}\ \mathrm{AP}(f_{\boldsymbol{\theta}}).
\end{aligned}$$

- **Lagrangian method**

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \quad \text{s.t. } g(\boldsymbol{x}) \le \mathbf{0}$$

$$\Updownarrow$$

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda} \ge \mathbf{0}} f(\boldsymbol{x}) + \boldsymbol{\lambda}^{\intercal} g(\boldsymbol{x})$$

Idea: alternating minimize x and maximize $\boldsymbol{\lambda}$ via gradient descent

Problem: infeasible solution; slow convergence

1.4 Other problems

- **Augmented Lagrangian methods for PINNs: infeasible solution**
- **First-order solver for PINNs: low quality solution**

2. No good solvers for CDL yet

Solvers or modeling languages	Nonconvex	Nonsmooth	Differentiable manifold constraints	General smooth constraint	Specific constrained ML problem	General CDL
PyTorch, Tensorflow, JAX, MXNet	✓	✓	✗	✗	✗	✗
CVX, AMPL, YALMIP, SDPT3, Cplex, Gurobi*, SDPT3, TFOCS	✗	✓	✗	✗	✗	✗
(Py)manopt, Geomstats, McTorch, Geoopt	✓	✓	✓	✗	✗	✗
KNITRO, IPOPT, GENO, ensmallen, TFCO, Cooper	✓	✓	✓	✓	✗	✗
Scikit-learn, MLib, Weka	✓	✓	✗	✗	✓	✗

3. GRANSO & PyGRANSO

- **Principled answers to issues in CDL methods**

  Stationarity & **feasibility** check: KKT condition
  Line search methods
  Gradient-sampling-based idea for nonsmoothness

- **A principled solver: GRANSO**

GRANSO $\quad$ Nonconvex, nonsmooth, constrained

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}), \quad \text{s.t. } c_i(\boldsymbol{x}) \le 0,\ \forall\, i \in \mathcal{I}; \quad c_i(\boldsymbol{x}) = 0,\ \forall\, i \in \mathcal{E}.$$

Penalty sequential quadratic programming

$$\min_{d \in \mathbb{R}^n,\ s \in \mathbb{R}^p} \quad \mu\left(f(x_k) + \nabla f(x_k)^{\intercal} d\right) + e^{\intercal} s + \frac{1}{2} d^{\intercal} H_k d$$

$$\text{s.t.} \quad c(x_k) + \nabla c(x_k)^{\intercal} d \le s, \quad s \ge 0,$$

Keep advantages:
Principled stopping criterion and line search, to obtain a solution with certificate (stationarity & feasibility check)
Quasi-newton style method for fast convergence, i.e., reasonable speed and high-precision solution

Problem:
Lack of Auto-Differentiation
Lack of GPU Support
No native support of tensor variables
⇒ impossible to do deep learning with GRANSO

- **NCVX PyGRANSO: first general-purpose solver for CDL**

Advantages:
Auto-Differentiation; GPU Support; support of tensor variables

Constrained folding:
Reduce # of constraints: reduce the cost of QP in the SQP

$$h_j(\boldsymbol{x}) = 0 \Longleftrightarrow |h_j(\boldsymbol{x})| \le 0 \qquad c_i(\boldsymbol{x}) \le 0 \Longleftrightarrow \max\{c_i(\boldsymbol{x}), 0\} \le 0$$

Equality into non-negative inequality $\qquad$ inequality into non-negative inequality

$$\mathcal{F}(|h_1(\boldsymbol{x})|, \cdots, |h_i(\boldsymbol{x})|, \max\{c_1(\boldsymbol{x}), 0\}, \cdots, \max\{c_j(\boldsymbol{x}), 0\}) \le 0,$$

All non-negative inequalities into one

See ncvx.org for detailed examples for CDL!

Ref: [1] Liang, B., Mitchell, T., & Sun, J. (2022). NCVX: A general-purpose optimization solver for constrained machine and deep learning. In OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop). [2] Liang, H., Liang, B., Peng, L., Cui, Y., Mitchell, T., & Sun, J. (2023). Optimization and Optimizers for Adversarial Robustness. arXiv preprint arXiv:2303.13401..
[3] Liang, H., Liang, B., Cui, Y., Mitchell, T., & Sun, J. (2022). Optimization for robustness evaluation beyond ℓp metrics. In OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop). [4] Peng, L., Travadi, Y., Cui, Y., & Sun, J. (2022). Direct Metric Optimization for Imbalanced Classification. In preparation for the Journal of Machine Learning Research.