Using Distributionally Robust Optimization to sharpen Mixed Integer Programming Formulations For Trained Neural Networks

Will Ma

Decision, Risk, and Operations Division
Graduate School of Business, Columbia University

joint work with **Ross Anderson** (Google), **Joey Huchette** (Rice), **Christian Tjandraatmadja** (Google), **Juan Pablo Vielma** (MIT)

The Basic Problem

Given a neural network NN which maps a region $D \subseteq \mathbb{R}^n$ to $\mathbb{R}$, compute

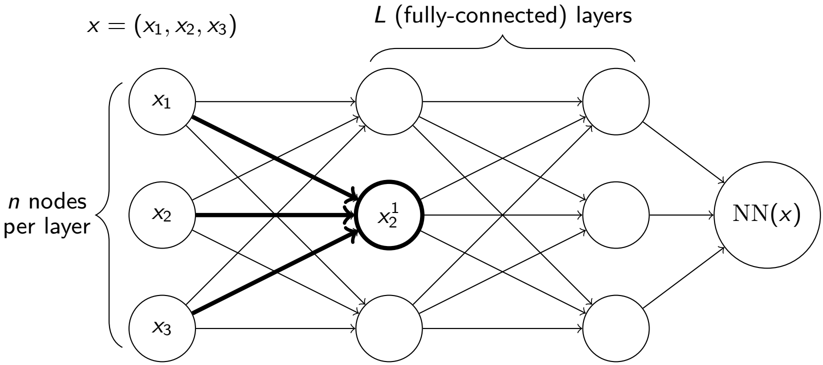
$$\max_{x \in D} \mathrm{NN}(x).$$

Outline of Talk

$$\max_{x \in D} \mathrm{NN}(x)$$

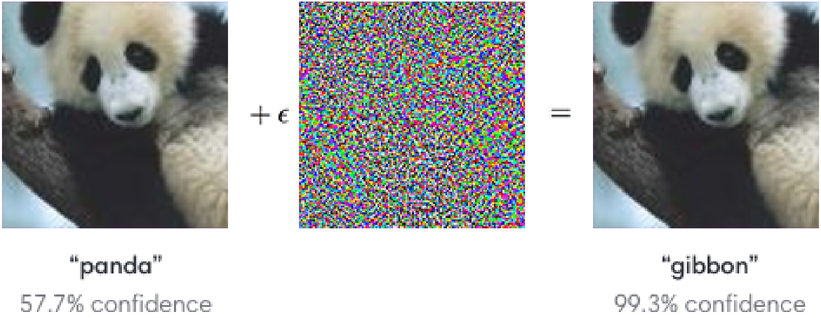
1. The Optimization Problem for a **Trained** Neural Network
   - Mixed Integer Programming (MIP) Formulations
   - Using Distributionally Robust Optimization (DRO) with Marginals
2. The Statistical Problem when the NN needs to be learned

What is an (artificial feedforward) Neural Network?



- at i'th neuron in ℓ'th layer, $x_i^\ell = \sigma(w^{\ell,i} \cdot x^{\ell-1} + b^{\ell,i})$, where $x^{\ell-1}$ is vector of variables from previous layer and σ is non-linear *activation function*
- often, $\sigma = \max\{\cdot, 0\}$ (ReLU)
- we will more generally consider $x_i^\ell = \max_{k=1,\dots,d}(w^{\ell,i,k} \cdot x^{\ell-1} + b^{\ell,i,k})$

Motivation for the Problem: NN Verification



- NN can classify images, but is vulnerable to adversarial examples (Goodfellow/Shlens/Szegedy '15)
- Want to verify NN is insensitive under small perturbations, e.g. prove

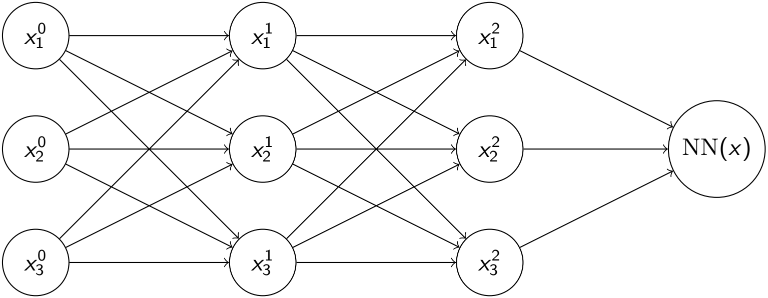
$$\max_{x:\|x-(\text{reference panda})\|_\infty \le \varepsilon} \mathrm{NN}_{\text{gibbon}}(x) - \mathrm{NN}_{\text{panda}}(x) \le 0$$

Basic Problem

Given a fixed $\mathrm{NN} : D \to \mathbb{R}$, compute $\max_{x \in D} \mathrm{NN}(x)$.

- $\mathrm{NN}(x)$ is generally non-convex
  - but composed of piecewise-linear functions
- verifiable optimality matters (cannot use first-order methods)
- evaluation is easy, but search region D is a large/continuous
  - e.g. box domain (∞-norm ball)

MIP to the rescue

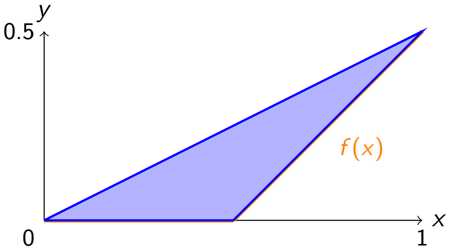


- recall that $x_i^\ell = \max_{k=1,\dots,d}(w^{\ell,i,k} \cdot x^{\ell-1} + b^{\ell,i,k})$
- add integer vector $z^{\ell,i} \in \{0,1\}^d$ at each neuron which equals the basis vector $\mathbf{e}^k$ for a piece k taking the maximum
- at each neuron ℓ, i, use **linear** constraints on the x_i^ℓ, $z_k^{\ell,i}$, and $x^{\ell-1}$ variables
- then solve using Branch and Bound

Desirable Properties of MIP Formulations

- single neuron defined by $f(x) = \max_{k=1,\ldots,d}(w^k \cdot x + b^k)$
- $x \in D$ is vector for previous layer, $y \in \mathbb{R}$ is value for neuron, z is vector of added integer variables
- valid: when $z \in \{0,1\}^d$, (x,y)-region described is function itself
- sharp: when $z \in [0,1]^d$, (x,y)-region described is convex hull
- ideal: (x,y,z)-region described is convex hull in extended space

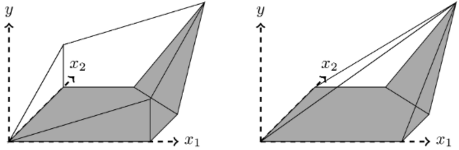
$$\boxed{\begin{array}{c} f(x) = \max\{x - 0.5, 0\} \\ \text{over } D = [0,1] \end{array}}$$



Not all MIP formulations are the same!

	Big-M	Our Formulation	Disjunctive ("Balas")
Tightness of LP-relaxation for Single Neuron	valid, not sharp	valid, sharp, **ideal for $d = 2$**	valid, **ideal**
# of Continuous Variables	$\mathbf{\Theta(Ln)}$	$\mathbf{\Theta(Ln)}$	$\Theta(Ln^2 d)$
Speed of BnB in Practice		**fastest**	

- Our formulation is tailored to Neural Nets (specifically, the max of d affine functions)
- Goal: add constraints (not variables) to Big-M formulation until sharp



Assume $D = [0,1]^n$. Want to describe points (x, y) in

$$\mathrm{conv}\left(\{(x, f(x)) : x \in [0,1]^n\}\right)$$

Step 1: at a fixed x, upper bound on y is

$$\sup_{X \text{ random vector over } [0,1]^n \text{ with } \mathbb{E}[X]=x} \mathbb{E}[f(X)]$$

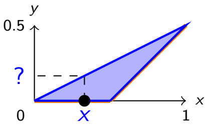
Step 2: by convexity/Jensen, can restrict X to $\{0,1\}^n$

$$= \sup_{X \text{ random vector over } \{0,1\}^n \text{ with } \mathbb{E}[X_i]=x_i \ \forall i} \mathbb{E}[f(X)]$$

Step 3: reinterpret as DRO with marginals

$$= \sup_{\theta \in \Gamma(\mathrm{Ber}(x_1),\ldots,\mathrm{Ber}(x_n))} \mathbb{E}_{X\sim\theta}[f(X)]$$

$f(x) = \max\{x - 0.5, 0\}$
over $D = [0,1]$



- $x = \frac{1}{3} \cdot 1 + \frac{2}{3} \cdot 0$
- upper bound on y is $\frac{1}{3} f(1) + \frac{2}{3} f(0) = \frac{1}{6}$

$$y \leq \sup_{\theta \in \Gamma(\mathrm{Ber}(x_1),\ldots,\mathrm{Ber}(x_n))} \mathbb{E}_{X\sim\theta}[\max_{k=1,\ldots,d}(w^k \cdot x + b^k)]$$

$$\iff y \leq \sup_{\nu \in \mathcal{D}(\{w^1,\ldots,w^d\})} \left(\sum_{i=1}^{n} \sup_{\theta_i \in \Gamma(\nu_i,\mathrm{Ber}(x_i))} \mathbb{E}_{(W_i,V_i)\sim\theta_i}[W_i V_i] + \sum_{k=1}^{d} b^k \nu(w^k) \right)$$

sup-sup duality from Distributionally Robust Optimization with marginals

- can switch sup and max and assume inner problem takes worst case
- Meilijson/Nadas '79, Natarajan/Song/Teo '09 [continuous marginals], Chen/M./Natarajan/Simchi-Levi/Yan '18 **[arbitrary marginals]**

Let $z_k = \nu(w^k)$, the probability that k'th piece is maximum

- after LP duality and a bit of massaging, get

$$y \leq \sum_{i=1}^{n} \min_{K=1,\ldots,d} \left(w_i^K x_i + \sum_{k=1}^{d} \max\{w_i^k - w_i^K, 0\} z_k \right) + \sum_{k=1}^{d} b^k z_k$$

- equivalent to exponential family of **linear** constraints

Our Formulation for a Single Neuron with $D = \prod_{i=1}^{n}[L_i, U_i]$, arbitrary b^k

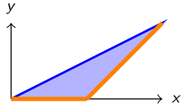
$$y \leq \sum_{i=1}^{n}\left(w_i^{K(i)}x_i + \sum_{k=1}^{d}\max\{(w_i^k - w_i^{K(i)})U_i, (w_i^{K(i)} - w_i^k)L_i\}z_k\right) + \sum_{k=1}^{d} b^k z_k$$

$\forall K : [n] \to [d] \longleftarrow$ exponential family, but separation oracle

$y \geq w^k \cdot x + b^k \qquad \forall k = 1, \dots, d \longleftarrow$ no lower-bound constraints added, by convexity

$x \in [L_1, U_1] \times \cdots \times [L_n, U_n]$

$z \in \Delta^d \cap \{0,1\}^d$



- valid, sharp, and no redundant constraints
- when $d = 2$, ideal
- the $d = 2$ result holds even if D is a product of simplices (useful for one-hot encoding binary features)

Some Remarks on our Sharpness/Idealness Results

Our results do not imply:

- an integral LP relaxation for whole network (unless only one neuron)

Our results do suggest:

- branching heuristics will be faster in practice
  [corroborated by our experiments; see also Vielma '15, Huchette '18]

Our results do prove:

- **minimal** polyhedral description of convex hull of ReLU (or max of $d = 2$ affine functions), in extended space with z-variables
- exponentially many facets; tractable separation procedure

But what about original space without z-variables?

Minimal Polyhedral Description in Original Space

We also establish the minimal polyhedral description of

$$\mathrm{conv}\Big(\big\{(x, \max\{w \cdot x + b, 0\}) : x \in D\big\}\Big)$$

(or max of 2 affine functions) with inequalities in x and y variables.

Exponentially many facets; tractable separation procedure

Key proof technique:

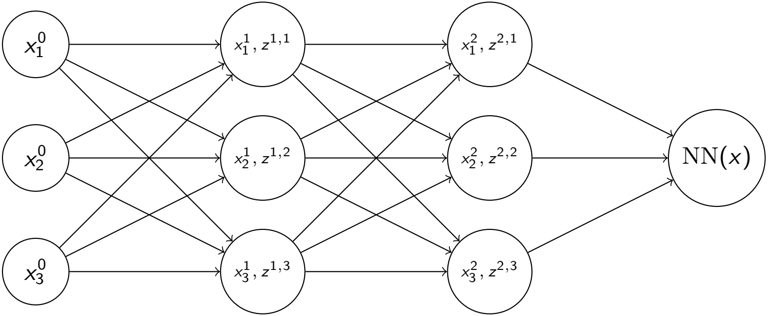
- when $d = 2$, function $\max_{k=1,\dots,d}(w^k \cdot x + b^k)$ is supermodular (modulo appropriate sign flips)
- therefore, extremal distribution in DRO with Marginals problem

$$\sup_{\theta \in \Gamma(\mathrm{Ber}(x_1),\dots,\mathrm{Ber}(x_n))} \mathbb{E}_{V\sim\theta}\left[\max_{k=1,\dots,d}(w^k \cdot V + b^k)\right]$$

is comonotonic coupling (perfect positive correlation)

Not the case when $d > 2$, which makes problem significantly more challenging!

Bound Propagation in the Formulations



- formulations at layer ℓ depend on domain $D^{\ell-1}$ for layer $\ell - 1$
- can write formulations for relaxation $\prod_i [L_i^{\ell-1}, U_i^{\ell-1}] \supseteq D^{\ell-1}$
- valid bounds L_i^ℓ, U_i^ℓ, with $\ell = 1, \ldots, L$, can be efficiently propagated through network, and require formulations of Projected Polyhedron

The Optimization Problem

Given a fixed $\mathrm{NN} : D \to \mathbb{R}$, compute $\max_{x \in D} \mathrm{NN}(x)$.

The Statistical Problem

Given data points $(x_j, y_j)_{j=1,\ldots,J}$, drawn IID from a distribution over $D \times \mathbb{R}$, find a point $x \in D$ which maximizes $\mathbb{E}[y|x]$.

A Solution Method:

1. Train a neural network NN which accurately predicts y given x, using the data points $(x_j, y_j)_{j=1,\ldots,J}$.
2. Find the point $x \in D$ which maximizes $\mathrm{NN}(x)$.

A Zoo of Applications

- $\max_x \mathrm{NN}_{\mathsf{panda}}(x)$
- x is the price vector for several complement/substitute products; y is the demand for a particular product
- x is a DNA sequence; y is how well it binds to a particular protein

Objections?

A Zoo of Issues

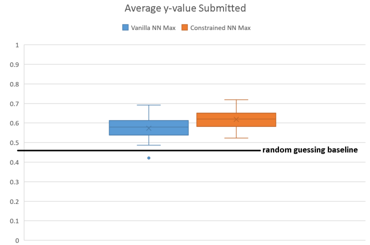
- endogeneity; prediction does not imply causation
- even with no confounding, low prediction error does not imply uniformly good approximation across D
- optimizer's curse: usually end up at x where $\mathrm{NN}(x)$ is astronomically high yet $\mathbb{E}[y|x]$ is worse than that of a "safe" point
- uninterpretable model; cannot incorporate any structure into $\mathrm{NN}(x)$

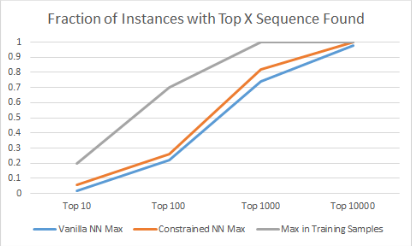
But, Some Unique Benefits

- NN solution method allows for *automated non-linear extrapolation*
- cannot be achieved by linear regression, or sticking to training points x_j with high value of y_j

Experiments on DNA Data

- $x \in D = \{G, C, A, T\}^8$; 65536 possible sequences
- assume y **deterministically** equals $f(x) \in [0, 1]$
- 1% of sequences are randomly chosen to be training data
- train fully-connected 4x100 ReLU NN; average squared loss < 0.01
- submit 15 best sequences **not in training data** according to NN
- repeat experiment (with new sequences as training data)—50 instances
- test Vanilla NN Max and Constrained NN Max (suggested by Bastani '19)





A Zoo of Possibilities

Summary

1. MIP formulations for $\max_{x \in D} \mathrm{NN}(x)$ problem
   - "**Strong Mixed-Integer Programming Formulations for Trained Neural Networks**" joint with Ross Anderson, Joey Huchette, Christian Tjandraatmadja, Juan Pablo Vielma
2. Using duality result from DRO with Marginals
   - "**Distributionally Robust Linear and Discrete Optimization with Marginals**" joint with Louis Chen, Karthik Natarajan, David Simchi-Levi, Zhenzhen Yan
3. Minimal polyhderal descriptions under no added integer variables, with application to bound propagation [in preparation]
4. Statistical Problem [WIP]

Thanks!

My contact: `wm2428@gsb.columbia.edu`