Robust Implicit Neural Networks via Contraction Theory

Non-Euclidean Monotone Operator Networks (NE-MON)

Saber Jafarpour*, Alexander Davydov*, Anton Proskurnikov, and Francesco Bullo

Decision and Control Laboratory
Georgia Institute of Technology

https://github.com/davydovalexander/Non-Euclidean_Mon_Op_Net

November 29, 2021

Acknowledgment

Alexander Davydov
UCSB

Anton Proskurnikov
Politecnico di Torino, Italy.

Francesco Bullo
UCSB

SJ* and A. Davydov* and A Proskurnikov and F. Bullo. *Robust Implicit Networks via Non-Euclidean Contractions*. NeurIPS, https://openreview.net/forum?id=SwfsoPuGYku, 2021.

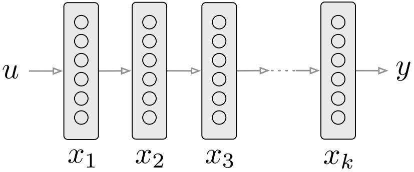
A. Davydov and SJ and F. Bullo. *Non-Euclidean Contraction Theory for Robust Nonlinear Stability*. arXiv: https://arxiv.org/abs/2103.12263, May 2021.

SJ and A. Davydov and F. Bullo. *Non-Euclidean Contraction Theory for Monotone and Positive Systems*. arXiv: http://arxiv.org/abs/2106.03194, May 2021.

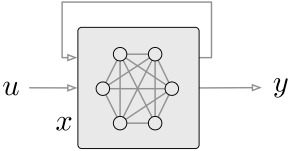
Implicit Neural Networks (INNs)

Definitions and motivations

- Explicit hidden layers are replaced by a single implicit layer



Feedforward neural network

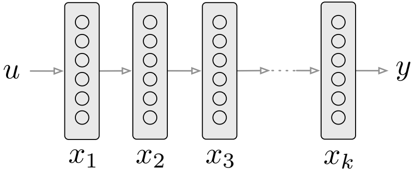


Implicit neural network

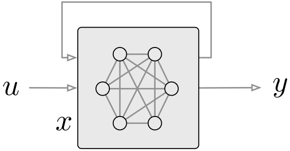
Implicit Neural Networks (INNs)

Definitions and motivations

- Explicit hidden layers are replaced by a single implicit layer



Feedforward neural network



Implicit neural network

- traditional neural networks:

$$x^{i+1} = \Phi(A_i x^i + B_i u + b_i)$$

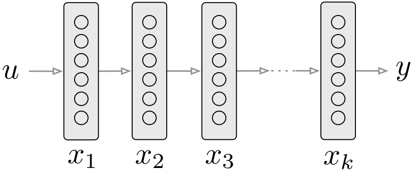
$$y = Cx^k + c$$

- $\Phi((y_1, \ldots, y_n)) = (\Phi_1(y_1), \ldots, \Phi_n(y_n))^\top$ is a diagonal activation function.

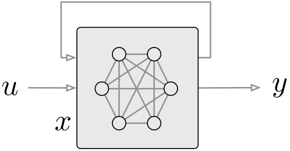
Implicit Neural Networks (INNs)

Definitions and motivations

- Explicit hidden layers are replaced by a single implicit layer



Feedforward neural network



Implicit neural network

- traditional neural networks:

$$x^{i+1} = \Phi(A_i x^i + B_i u + b_i)$$
$$y = Cx^k + c$$

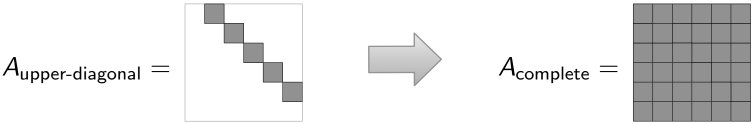
- implicit neural networks:

$$x = \Phi(Ax + Bu + b)$$
$$y = Cx + c$$

- $\Phi((y_1, \dots, y_n)) = (\Phi_1(y_1), \dots, \Phi_n(y_n))^\top$ is a diagonal activation function.

Motivation #1: Generalizing FF to fully-connected synaptic matrices

$x^{i+1} = \Phi(A_i x^i + B_i u + b_i) \quad \Longleftrightarrow \quad x = \Phi(Ax + Bu + b)$, where A has upper diagonal structure.

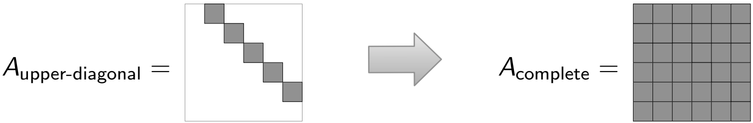


$A_{\text{upper-diagonal}} =$

$A_{\text{complete}} =$

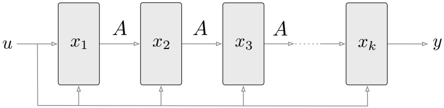
Motivation #1: Generalizing FF to fully-connected synaptic matrices

$x^{i+1} = \Phi(A_i x^i + B_i u + b_i) \quad \Longleftrightarrow \quad x = \Phi(Ax + Bu + b)$, where A has upper diagonal structure.



$$A_{\text{upper-diagonal}} = \qquad \qquad A_{\text{complete}} =$$

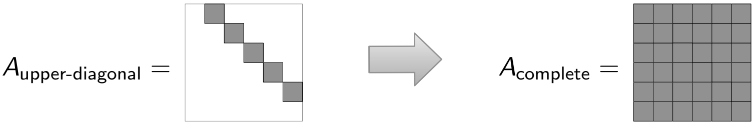
Motivation #2: Weight-tied infinite-depth NN → fixed-point of INN



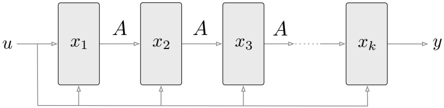
$$x^{i+1} = \Phi(Ax^i + B_i u + b_i) \implies \lim_{i\to\infty} x^i = x^* \text{ solution to the INN}$$

Motivation #1: Generalizing FF to fully-connected synaptic matrices

$x^{i+1} = \Phi(A_i x^i + B_i u + b_i) \iff x = \Phi(Ax + Bu + b)$, where A has upper diagonal structure.



Motivation #2: Weight-tied infinite-depth NN → fixed-point of INN



$$x^{i+1} = \Phi(Ax^i + B_i u + b_i) \implies \lim_{i\to\infty} x^i = x^* \text{ solution to the INN}$$

Motivation #3: Neural ODE model (large time) → fixed-point of INN

$\dot{x} = -x + \Phi(Ax + Bu + b) \implies \lim_{t\to\infty} x(t) = x^*$ solution to INN

Implicit Neural Networks (INNs)

Training implicit network

- Training INNs:
  1. loss function $\mathcal{L}$
  2. training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^N$
  3. **training optimization problem**

$$\min_{A,B,C} \quad \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i + c)$$

$$x_i = \Phi(Ax_i + B\widehat{u}_i + b)$$

Implicit Neural Networks (INNs)

Training implicit network

- Training INNs:
  1. loss function $\mathcal{L}$
  2. training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^N$
  3. **training optimization problem**

$$\min_{A,B,C} \quad \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i + c)$$

$$x_i = \Phi(Ax_i + B\widehat{u}_i + b)$$

- Efficient back-propagation through implicit differentiation
- Stochastic gradient descent: at each step solve $x = \Phi(Ax + Bu + b)$.

Implicit Neural Networks (INNs)

Training implicit network

- Training INNs:
    1. loss function $\mathcal{L}$
    2. training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^N$
    3. **training optimization problem**

$$\min_{A,B,C} \quad \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i + c)$$
$$x_i = \Phi(Ax_i + B\widehat{u}_i + b)$$

- Efficient back-propagation through implicit differentiation
- Stochastic gradient descent: at each step solve $x = \Phi(Ax + Bu + b)$.

Challenge #1: well-posedness of fixed-point equation

computing solution of of fixed-point equation

Robustness of INNs

Adverserial examples

- **Adversarial examples:** a small change in input causes a big change in output?

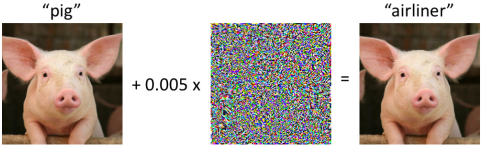


Image credit: MIT CSAIL

Robustness of INNs

Adverserial examples

- **Adversarial examples:** a small change in input causes a big change in output?

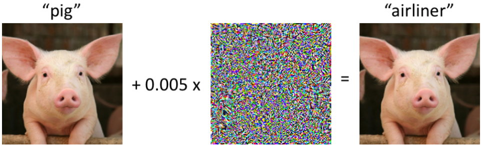


Image credit: MIT CSAIL

- Robustness measures: input-to-output Lipschitz constant

  C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. 2014

Robustness of INNs

Adverserial examples

- **Adversarial examples:** a small change in input causes a big change in output?

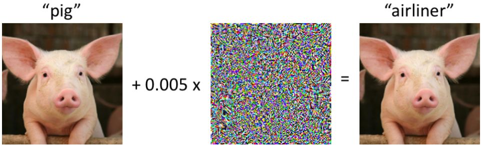


Image credit: MIT CSAIL

- Robustness measures: input-to-output Lipschitz constant

  C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. 2014

  1. ℓ_2**-norm Lipschitz constant**: not informative in many scenarios
  2. ℓ_∞**-norm Lipschitz constant**: large-scale input wt wide-spread perturbations

Robustness of INNs

Adverserial examples

- **Adversarial examples:** a small change in input causes a big change in output?

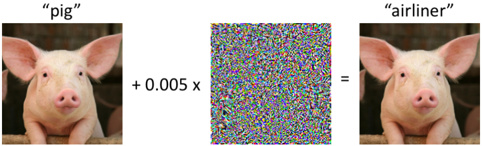


Image credit: MIT CSAIL

- Robustness measures: input-to-output Lipschitz constant

  C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. 2014

  1. ℓ_2**-norm Lipschitz constant**: not informative in many scenarios
  2. ℓ_∞**-norm Lipschitz constant**: large-scale input wt wide-spread perturbations

Challenge #2: computing robustness margins

Challenge #3: implementing robustness in training

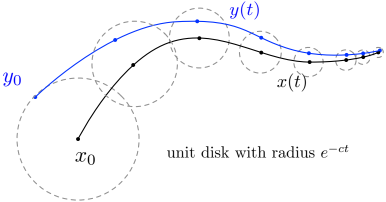
Recent literature on implicit NNs

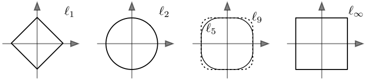
1. S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. 2019
2. L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Y. Tsai. Implicit deep learning. 2019
3. E. Winston and J. Z. Kolter. Monotone operator equilibrium networks. 2020. URL `https://arxiv.org/abs/2006.08591`
4. M. Revay, R. Wang, and I. R. Manchester. Lipschitz bounded equilibrium networks. 2020. URL `https://arxiv.org/abs/2010.01732`
5. A. Kag, Z. Zhang, and V. Saligrama. RNNs incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients? In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HylpqA4FwS`
6. K. Kawaguchi. On the theory of implicit deep learning: Global convergence with implicit layers. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=p-NZIuwqhI4`
7. S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin. Fixed point networks: Implicit depth models with Jacobian-free backprop, 2021. URL `https://arxiv.org/abs/2103.12803`. ArXiv e-print

Contraction theory

Definitions

$\dot{x} = \mathsf{G}(x)$ is contractive if its flow is a contraction map

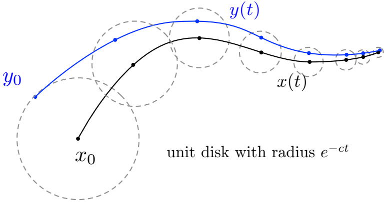


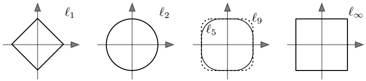


Contraction theory

Definitions

$\dot{x} = G(x)$ is contractive if its flow is a contraction map



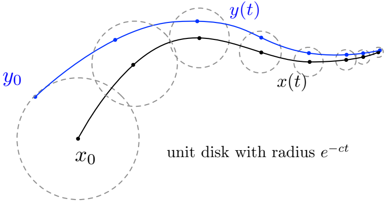


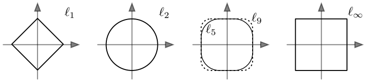
1. initial conditions are forgotten
2. unique globally exponential stable equilibrium
3. input-to-state robustness
4. **accurate numerical integration and fixed-point computation**

Contraction theory

Definitions

$\dot{x} = \mathsf{G}(x)$ is contractive if its flow is a contraction map





1. initial conditions are forgotten
2. unique globally exponential stable equilibrium
3. input-to-state robustness
4. **accurate numerical integration and fixed-point computation**

A vector field $\mathsf{G} : \mathbb{R}^n \to \mathbb{R}^n$ is contracting with respect to the norm $\|\cdot\|$ iff

$$\mu(D_x \mathsf{G}(x)) \leq -c, \qquad \text{for all } x$$

Contraction theory

Matrix measures

The **matrix measure** of $A \in \mathbb{R}^{n\times n}$ wrt to $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}.$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,

Contraction theory

Matrix measures

The **matrix measure** of $A \in \mathbb{R}^{n\times n}$ wrt to $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}.$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,

$$\mu_2(A) = \tfrac{1}{2}\lambda_{\max}(A + A^\top)$$

$$\mu_1(A) = \max_j \left(a_{jj} + \sum\nolimits_{i\neq j} |a_{ij}|\right) \qquad \mu_\infty(A) = \max_i \left(a_{ii} + \sum\nolimits_{j\neq i} |a_{ij}|\right)$$

Contraction theory

Matrix measures

The **matrix measure** of $A \in \mathbb{R}^{n\times n}$ wrt to $\|\cdot\|$:

$$\mu_{\|\cdot\|}(A) := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}.$$

- Directional derivative of norm $\|\cdot\|$ in direction of A,

$$\mu_2(A) = \tfrac{1}{2}\lambda_{\max}(A + A^\top)$$

$$\mu_1(A) = \max_j \Big(a_{jj} + \sum\nolimits_{i\neq j} |a_{ij}|\Big) \qquad \mu_\infty(A) = \max_i \Big(a_{ii} + \sum\nolimits_{j\neq i} |a_{ij}|\Big)$$

Basic properties:

subadditivity: $\mu(A+B) \leq \mu(A) + \mu(B),$

convexity: $\mu(\theta A + (1-\theta)B) \leq \theta\mu(A) + (1-\theta)\mu(B), \quad \forall\theta \in [0,1]$

norm/spectrum: $\mathrm{Re}(\lambda) \leq \mu(A) \leq \|A\|, \quad \forall\lambda \in \mathrm{spec}(A)$

Contraction theory

Non-Euclidean contractions

$\ell_2 -$ **contraction** **LMI**

$$\mu_2(D_x\mathsf{G}(x)) \leq -c \quad \Longleftrightarrow \quad D_x\mathsf{G}(x) + D_x\mathsf{G}(x)^\top \preceq -cI$$

- Monotone Operator Theory

  E. K. Ryu and S. Boyd. Primer on monotone operator methods. *Applied Computational Mathematics*, 15(1):3–43, 2016

Contraction theory

Non-Euclidean contractions

$\ell_2 -$ **contraction** | **LMI**

$$\mu_2(D_x\mathsf{G}(x)) \leq -c \quad \iff \quad D_x\mathsf{G}(x) + D_x\mathsf{G}(x)^\top \preceq -cI$$

- Monotone Operator Theory

  E. K. Ryu and S. Boyd. Primer on monotone operator methods. *Applied Computational Mathematics*, 15(1):3–43, 2016

$\ell_\infty -$ **contraction** | **Diagonal Dominance**

$$\mu_\infty(D_x\mathsf{G}(x)) \leq -c, \quad \iff \quad (D_x\mathsf{G}(x))_{ii} + \sum_{j\neq i} |(D_x\mathsf{G}(x))_{ij}| \leq -c, \quad \forall i$$

- Non-Euclidean Monotone Operator Theory

Solvability of fixed-point equations

A contraction-based framework

Problem statement

For a fixed-point equation

$$x = \mathsf{F}(x, u) \qquad \text{(for implicit neural networks } \mathsf{F}(x, u) = \Phi(Ax + Bu + b)\text{)}$$

1. when do we have a unique solution?
2. how to efficiently compute it?

Solvability of fixed-point equations

A contraction-based framework

Problem statement

For a fixed-point equation

$$x = \mathsf{F}(x, u) \qquad \text{(for implicit neural networks } \mathsf{F}(x, u) = \Phi(Ax + Bu + b))$$

1. when do we have a unique solution?
2. how to efficiently compute it?

Infinite layer interpretation: convergence of the Picard iterations

$$x^{k+1} = \mathsf{F}(x^k, u)$$

Banach Fixed-point Theorem: $\|D_x \mathsf{F}(x, u)\| < 1$.

Solvability of fixed-point equations

A contraction-based framework

Key insight

$$\begin{array}{ccc} \text{Fixed-point of} & \Longleftrightarrow & \text{Equilibrium point of} \\ x = \mathsf{F}(x, u) & & \dot{x} = -x + \mathsf{F}(x, u) \end{array}$$

- **Contraction theory**: existence and uniqueness of equilibrium point

$$\mu(D_x\mathsf{F}(x, u)) < 1.$$

- $\mu(D_x\mathsf{F}(x, u)) < 1$ is less conservative than $\|D_x\mathsf{F}(x, u)\| < 1$.

Solvability of fixed-point equations

A contraction-based framework

Key insight

Fixed-point of $x = \mathsf{F}(x, u)$ $\Longleftrightarrow$ Equilibrium point of $\dot{x} = -x + \mathsf{F}(x, u)$

- **Contraction theory**: existence and uniqueness of equilibrium point

$$\mu(D_x\mathsf{F}(x, u)) < 1.$$

- $\mu(D_x\mathsf{F}(x, u)) < 1$ is less conservative than $\|D_x\mathsf{F}(x, u)\| < 1$.

Theorem: Fixed-point via matrix measure condition

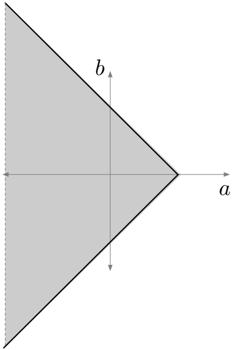
If $\mu(D_x\mathsf{F}(x, u)) < 1$ then

1. F has a unique fixed-point x_u^*.
2. $x^{k+1} = (1 - \alpha)x^k + \alpha\mathsf{F}(x^k, u)$ converges to x_u^*, for $0 < \alpha \leq \alpha^*$.

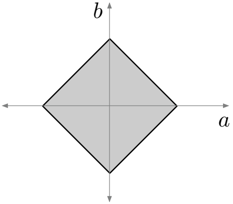
Solvability of fixed-point equations

Example

- $\mathsf{F}(x) = Mx \in \mathbb{R}^2$ with $M = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \in \mathbb{R}^{2\times 2}$.



$\mathrm{osL}_\infty \mathsf{F} = \mu_\infty(M) < 1$



$\mathrm{Lip}_\infty \mathsf{F} = \|M\|_\infty < 1$

Well-posedness of INNs

Computing fixed-points

$$x = \Phi(Ax + Bu + b)$$

Theorem: Fixed-points of INNs

If $\mu_\infty(A) < 1$, then

1. there exists a unique fixed-point,
2. for $\alpha \in]0, (1 - \min_i (a_{ii})_-))^{-1}]$, the average map is a contraction map:
$$N_\alpha(x) := (1 - \alpha)x + \alpha\Phi(Ax + Bu + b)$$
3. minimal contraction factor is
$$\mathrm{Lip}(N_{\alpha^*}) = 1 - \frac{1 - \mu_\infty(A)_+}{1 - \min_i (a_{ii})_-}$$

Well-posedness of INNs

Computing fixed-points

$$x = \Phi(Ax + Bu + b)$$

Theorem: Fixed-points of INNs

If $\mu_\infty(A) < 1$, then

1. there exists a unique fixed-point,
2. for $\alpha \in]0, (1 - \min_i (a_{ii})_-))^{-1}]$, the average map is a contraction map:
$$N_\alpha(x) := (1 - \alpha)x + \alpha\Phi(Ax + Bu + b)$$
3. minimal contraction factor is
$$\mathrm{Lip}(N_{\alpha^*}) = 1 - \frac{1 - \mu_\infty(A)_+}{1 - \min_i (a_{ii})_-}$$

Interpretation: The iteration $x^{k+1} = N_\alpha(x^k)$ is Euler discretization of

$$\dot{x} = -x + \Phi(Ax + Bu + b)$$

Robustness of fixed-point equations

Input-to-state Lipschitz bounds

Problem statement

How does the fixed-point of $x = \mathsf{F}(x, u)$ change with u?

Robustness of fixed-point equations

Input-to-state Lipschitz bounds

Problem statement

How does the fixed-point of $x = \mathsf{F}(x, u)$ change with u?

Theorem: Input-to-state Lipschitz bounds

x_u^* is a fixed-point of $x = \mathsf{F}(x, u)$ and $\mu(D_x\mathsf{F}) < 1$, then

$$\|x_u^* - x_v^*\| \leq \frac{\|D_u\mathsf{F}\|}{1 - \mu(D_x\mathsf{F})}\|u - v\|$$

Robustness of INNs

Computing the Lipschitz bounds

$$\begin{aligned} x &= \Phi(Ax + Bu + b), \\ y &= Cx + c \end{aligned}$$

- How to compute Lipschitz bounds in INNs?

$$u \underbrace{\mapsto}_{\mathrm{Lip}_{u \to x^*}} x^* \underbrace{\mapsto}_{\mathrm{Lip}_{x^* \to y}} y$$

$$\mathrm{Lip}_{u \to y} = \mathrm{Lip}_{u \to x^*} \mathrm{Lip}_{x^* \to y}$$

Robustness of INNs

Computing the Lipschitz bounds

$$\begin{aligned} x &= \Phi(Ax + Bu + b), \\ y &= Cx + c \end{aligned}$$

- How to compute Lipschitz bounds in INNs?

$$u \underbrace{\mapsto}_{\mathrm{Lip}_{u\to x^*}} x^* \underbrace{\mapsto}_{\mathrm{Lip}_{x^*\to y}} y$$

$$\mathrm{Lip}_{u\to y} = \mathrm{Lip}_{u\to x^*}\mathrm{Lip}_{x^*\to y}$$

Theorem: Input-to-output Lipschitz constant

if $\mu_\infty(A) < 1$ then

$$\mathsf{Lip}_{u\to y} = \frac{\|B\|_\infty\|C\|_\infty}{1-\mu_\infty(A)_+}.$$

Training INNs

Well-posedness condition + promoting robustness

How to train well-posed and robust INNs?

1. Loss function $\mathcal{L}$
2. Training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^N$

$$\min_{A,B,C} \quad \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i + c) + \quad \lambda \quad \mathrm{Lip}_{u \to y}$$

$$x_i = \Phi(Ax_i + B\widehat{u}_i + b)$$

$$\mu_\infty(A) \leq \gamma,$$

- $\gamma < 1$ is a hyperparameter
- $\lambda \geq 0$ is a regularization parameter.

Training INNs

Well-posedness condition + promoting robustness

How to train well-posed and robust INNs?

1. Loss function $\mathcal{L}$
2. Training data $(\widehat{u}_i, \widehat{y}_i)_{i=1}^{N}$

$$\min_{A,B,C} \quad \sum_{i=1}^{N} \mathcal{L}(\widehat{y}_i, Cx_i + c) + \quad \lambda \quad \mathrm{Lip}_{u \to y}$$

$$x_i = \Phi(Ax_i + B\widehat{u}_i + b)$$

$$\mu_\infty(A) \leq \gamma,$$

- $\gamma < 1$ is a hyperparameter
- $\lambda \geq 0$ is a regularization parameter.

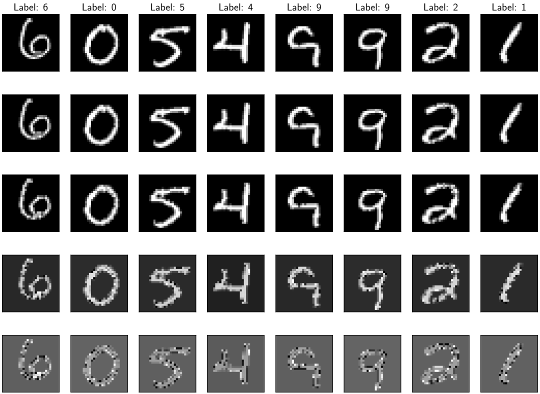
Theorem: Parametrization of ℓ_∞-measure constraint

$$\mu_\infty(A) \leq \gamma \quad \Longleftrightarrow \quad \exists T \text{ s.t. } A = T + |T|\mathbb{1}_n + \gamma I_n.$$

Numerical Experiments

Robustness of INNs

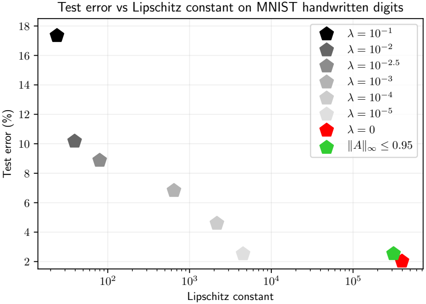
- MNIST handwritten digit dataset
- implicit neural network order: $n = 100$
- Loss function: cross entropy
- perturbation: inversion attack

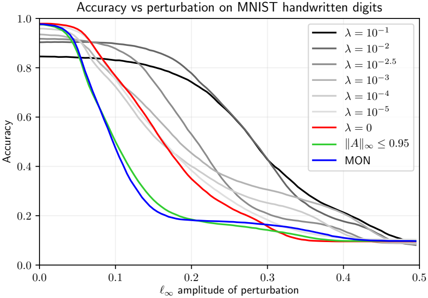


Numerical Experiments

Robustness of INNs

- Tradeoff between **accuracy** and **robustness**

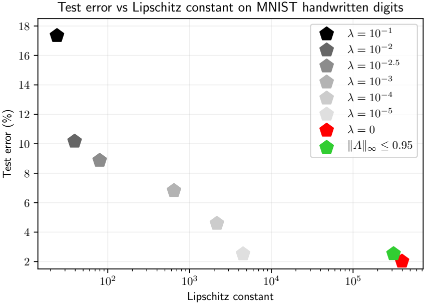


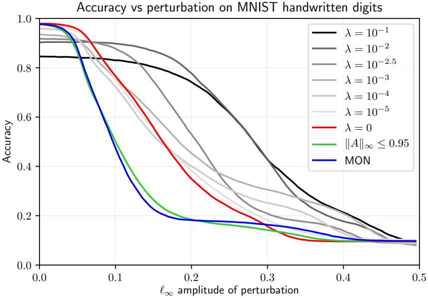


Numerical Experiments

Robustness of INNs

- Tradeoff between **accuracy** and **robustness**


Test error vs Lipschitz constant on MNIST handwritten digits

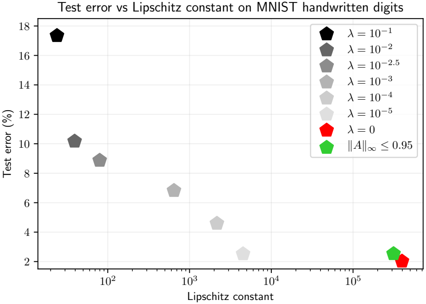

Accuracy vs perturbation on MNIST handwritten digits

- Pareto-optimal curve

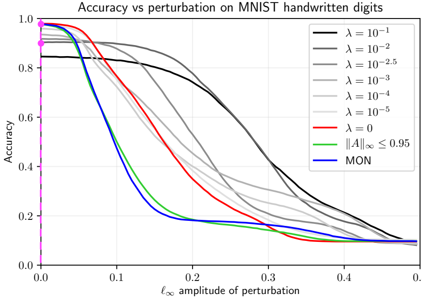
Numerical Experiments

Robustness of INNs

- Tradeoff between **accuracy** and **robustness**



Test error vs Lipschitz constant on MNIST handwritten digits



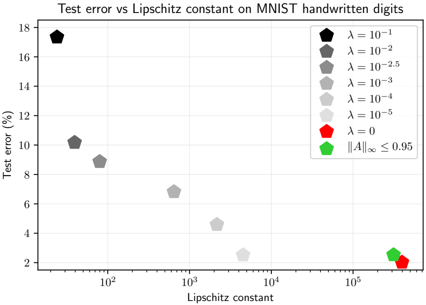
Accuracy vs perturbation on MNIST handwritten digits

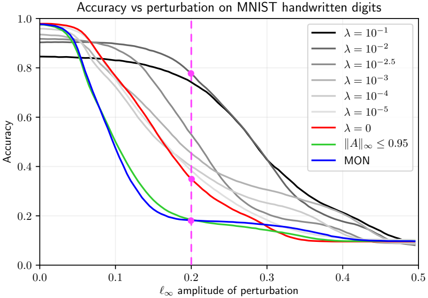
- Pareto-optimal curve
- Clean performance vs. robustness

Numerical Experiments

Robustness of INNs

- Tradeoff between **accuracy** and **robustness**


Test error vs Lipschitz constant on MNIST handwritten digits


Accuracy vs perturbation on MNIST handwritten digits

- Pareto-optimal curve
- Clean performance vs. robustness

Conclusions

- Non-Euclidean contraction theory using matrix measures
- Existence, uniqueness, and computing fixed-points of INNs
- Robustness margins of INNs using input-to-output Lipschitz constants
- Improve robustness in training using Lipschitz bounds