IMPLICIT DEEP LEARNING

Laurent El Ghaoui

CDAR Risk Seminar
UC Berkeley
March 10, 2020

Berkeley Artificial Intelligence Laboratory (BAIR)
EECS and IEOR Departments, UC Berkeley

COLLABORATORS

Joint work with:

- Armin Askari, Fangda Gu, Bert Travacca, Alicia Tsai (UC Berkeley);
- Mert Pilanci (Stanford);
- Emmanuel Vallod, Stefano Proto (`www.sumup.ai`).

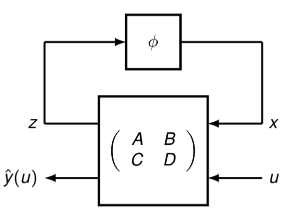
Sponsors:

NSF

BAIR BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

SUMUP ANALYTICS

IMPLICIT PREDICTION RULE



Equilibrium equation:

$$x = \phi(Ax + Bu)$$

Prediction:

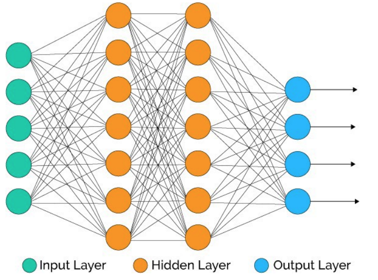
$$\hat{y}(u) = Cx + Du$$

- Input $u \in \mathbb{R}^p$, predicted output $\hat{y}(u) \in \mathbb{R}^q$, hidden "state" vector $x \in \mathbb{R}^n$.
- Model parameter matrix:

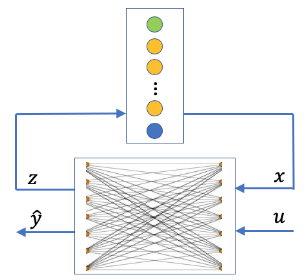
$$M = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right).$$

- Activation: vector map $\phi : \mathbb{R}^n \to \mathbb{R}^n$, *e.g.* the ReLU: $\phi(\cdot) = \max(\cdot, 0)$ (acting componentwise on vectors).

DEEP NEURAL NETS AS IMPLICIT MODELS

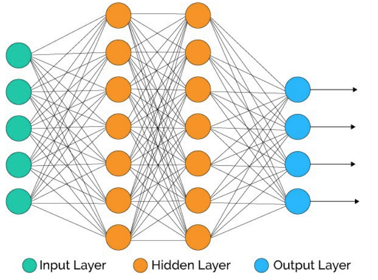


A neural network.

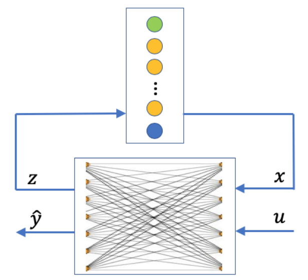


An implicit model.

DEEP NEURAL NETS AS IMPLICIT MODELS



A neural network.



An implicit model.

Implicit models are more general: they allow loops in the network graph.

EXAMPLE

Fully connected, feedforward neural network:

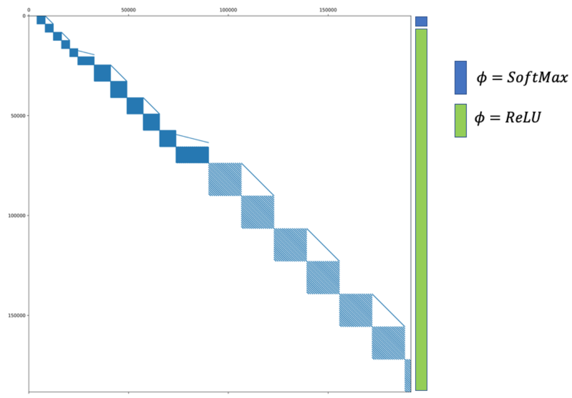
$$\hat{y}(u) = W_L x_L, \quad x_{l+1} = \phi_l(W_l x_l), \quad l = 1, \ldots, L-1, \quad x_0 = u.$$

Implicit model:

$$\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) = \left(\begin{array}{cccc|c} 0 & W_{L-1} & \ldots & 0 & 0 \\ & 0 & \ddots & \vdots & \vdots \\ & & \ddots & W_1 & 0 \\ & & & 0 & W_0 \\ \hline W_L & 0 & \ldots & 0 & 0 \end{array}\right), \quad x = \begin{pmatrix} x_L \\ \vdots \\ x_1 \end{pmatrix}, \quad \phi(z) = \begin{pmatrix} \phi_L(z_L) \\ \vdots \\ \phi_1(z_1) \end{pmatrix}.$$

The equilibrium equation $x = \phi(Ax + Bu)$ is easily solved via backward substitution (forward pass).

EXAMPLE: RESNET20



The A matrix for ResNet20.

- 20-layer network, implicit model of order $n \sim 180000$.
- Convolutional layers have blocks with Toeplitz structure.
- Residual connections appear as lines.

NEURAL NETWORKS AS IMPLICIT MODELS

Framework covers most neural network architectures:

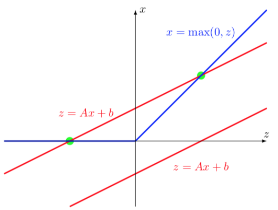
- Neural nets have strictly upper triangular matrix A.
- Equilibrium equation solved by substitution, *i.e.* “forward pass”.
- State vector x contains all the hidden features.
- Activation ϕ can be different for each component or blocks of x.
- Covers CNNs, RNNs, recurrent neural networks, (Bi-)LSTM, attention, transformers, etc.

WELL-POSEDNESS

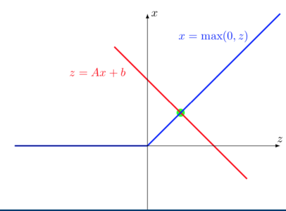
The matrix $A \in \mathbb{R}^{n \times n}$ is said to be well-posed for ϕ if, for every $b \in \mathbb{R}^n$, a solution $x \in \mathbb{R}^n$ to the equation

$$x = \phi(Ax + b),$$

exists, and it is unique.



Equation has two or no solutions, depending on **sign**(b).



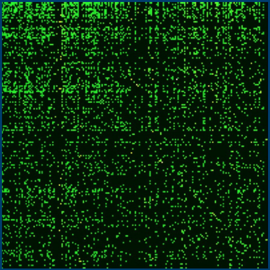
Solution is unique for every b.

PERRON-FROBENIUS THEORY

A square matrix P with non-negative entries admits a real eigenvalue λ with a non-negative eigenvector $v \neq 0$:

$$Pv = \lambda v.$$

The value λ dominates all the other eigenvalues.



A web link matrix.

Google's Page rank search engine relies on computing the Perron-Frobenius eigenvector of the web link matrix.

PF SUFFICIENT CONDITION FOR WELL-POSEDNESS

Fact: Assume that ϕ is componentwise non-expansive (*e.g.*, $\phi = \text{ReLU}$):

$$\forall\, u, v \in \mathbb{R}^n \;:\; |\phi(u) - \phi(v)| \leq |u - v|.$$

Then the matrix A is well-posed for ϕ if the non-negative matrix $|A|$ satisfies

$$\lambda_{pf}(|A|) < 1,$$

in which case the solution can be found via the fixed-point iterations:

$$x(t+1) = \phi(Ax(t) + b), \quad t = 0, 1, 2, \ldots$$

Covers neural networks: since then $|A|$ is strictly upper triangular, thus $\lambda_{pf}(|A|) = 0$.

NORM CONDITION

More conservative condition: $\|A\|_\infty < 1$, where

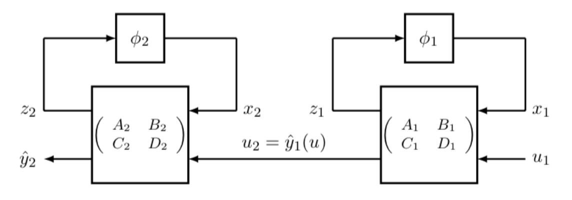
$$\lambda_{\mathrm{PF}}(|A|) \leq \|A\|_\infty := \max_i \sum_j |A_{ij}|.$$

Under previous PF conditions for well-posedness:

- we can always rescale the model so that $\|A\|_\infty < 1$, without altering the prediction rule;
- scaling related to PF eigenvector of $|A|$.

Hence during training we may simply use norm condition.

COMPOSING IMPLICIT MODELS

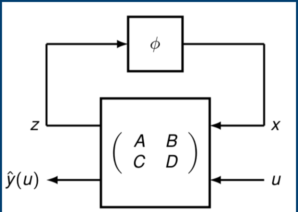


A cascade connection.

Class of implicit models closed under the following connections:

- Cascade
- Parallel and sum
- Multiplicative
- Feedback

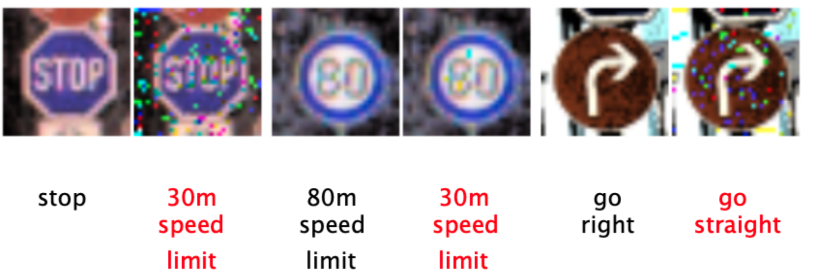
Goal: analyze the impact of input perturbations on the state and outputs.



Motivations:

- Diagnose a given (implicit) model.
- Generate adversarial attacks.
- Defense: modify the training problem so as to improve robustness properties.

WHY DOES IT MATTER?



Changing a few carefully chosen pixels in a test image can cause a classifier to mis-categorize the image (Kwiatkowska *et al.*, 2019).

Input is unknown-but-bounded: $u \in \mathcal{U}$, with

$$\mathcal{U} := \left\{ u^0 + \delta \in \mathbb{R}^p \ : \ |\delta| \leq \sigma_u \right\},$$

- $u^0 \in \mathbb{R}^n$ is a “nominal” input;
- $\sigma_u \in \mathbb{R}^n_+$ is a measure of componentwise uncertainty around it.

Assume (sufficient condition for) well-posedness:

- ϕ componentwise non-expansive;
- $\lambda_{\mathrm{PF}}(|A|) < 1$.

Nominal prediction:

$$x^0 = \phi(Ax^0 + Bu^0), \quad \hat{y}(u^0) = Cx^0 + Du^0.$$

COMPONENT-WISE BOUNDS ON THE STATE AND OUTPUT

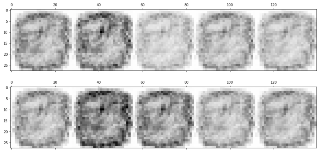
Fact: If $\lambda_{\rm PF}(|A|) < 1$, then $I - |A|$ is invertible, and

$$|\hat{y}(u) - \hat{y}(u^0)| \leq S|u - u^0|,$$

where

$$S := |C|(I - |A|)^{-1}|B| + |D|$$

is a "sensitivity matrix" of the implicit model.



Sensitivity matrix of a classification network with 10 outputs (each image is a row).

GENERATE A SPARSE ATTACK ON A TARGETED OUTPUT

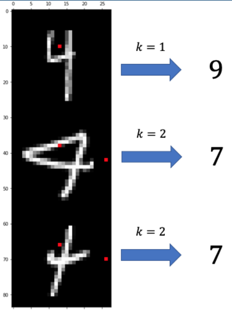
Attack method:

- select the output to attack based on the rows (class) of sensitivity matrix;
- select top k entries in chosen row;
- randomly alter corresponding pixels.

GENERATE A SPARSE ATTACK ON A TARGETED OUTPUT

Attack method:

- select the output to attack based on the rows (class) of sensitivity matrix;
- select top k entries in chosen row;
- randomly alter corresponding pixels.



Changing $k = 1$ (top) $k = 2$ (mid, bot) pixels, images are wrongly classified, and accuracy decreases from 99% to 74%.

GENERATE A SPARSE BOUNDED ATTACK ON A TARGETED OUTPUT

Target a specific output with sparse attacks:

$$\mathcal{U} := \left\{ u^0 + \delta \in \mathbb{R}^p \ : \ |\delta| \le \sigma_u, \ \ \textbf{Card}(\delta) \le k \right\},$$

With $k \le n$. Solve a linear program, with c related to chosen target:

$$\max_{x,\, u} \ c^\top x \ : \quad \begin{array}{l} x \ge Ax + Bu, \ \ x \ge 0, \ \ |x - x^0| \le \sigma_x, \ \ |u - u^0| \le \sigma_u \\ \|\,\textbf{diag}(\sigma_u)^{-1}(u - u^0)\|_1 \le k. \end{array}$$

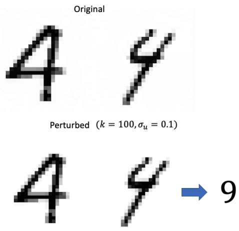
GENERATE A SPARSE BOUNDED ATTACK ON A TARGETED OUTPUT

Target a specific output with sparse attacks:

$$\mathcal{U} := \left\{ u^0 + \delta \in \mathbb{R}^p \ : \ |\delta| \le \sigma_u, \ \ \mathbf{Card}(\delta) \le k \right\},$$

With $k \le n$. Solve a linear program, with c related to chosen target:

$$\max_{x,\,u} \ c^\top x \ : \quad x \ge Ax + Bu, \ \ x \ge 0, \ \ |x - x^0| \le \sigma_x, \ \ |u - u^0| \le \sigma_u$$
$$\|\mathbf{diag}(\sigma_u)^{-1}(u - u^0)\|_1 \le k.$$



Changing $k = 100$ pixels by a tiny amount ($\sigma_u = 0.1$), target images are wrongly classified by a network with 99% nominal accuracy.

TRAINING PROBLEM

- Inputs: $U = [u_1, \ldots, u_m]$, with m data points $u_i \in \mathbb{R}^p$, $i \in [m]$.
- Outputs: $Y = [y_1, \ldots, y_m]$, with m responses $y_i \in \mathbb{R}^q$, $i \in [m]$.
- Loss function:

$$\mathcal{L}(Y, \hat{Y}) = \sum_{i=1}^{m} \mathcal{L}(y_i, \hat{y}_i)$$

with *e.g.* $\mathcal{L}$ a cross-entropy loss (assuming $y \geq 0$, $\mathbf{1}^T y = 1$)

$$\mathcal{L}(y, \hat{y}) = \log(\sum_{j=1}^{q} e^{\hat{y}(j)}) - y^\top \hat{y}.$$

Predictions: with $X = [x_1, \ldots, x_m] \in \mathbb{R}^{n \times m}$ the matrix of hidden feature vectors, and ϕ acting columnwise,

$$\hat{Y} = CX + DU, \quad X = \phi(AX + BU).$$

TRAINING PROBLEM

$$\min_{X,A,B,C,D} \quad \mathcal{L}(Y, \hat{Y}) + \pi(A, B, C, D)$$
$$\text{s.t.} \quad \hat{Y} = CX + DU, \quad X = \phi(AX + BU), \quad \|A\|_\infty \leq \kappa.$$

- Constraint on A with $\kappa < 1$ ensures well-posedness.
- $\pi(\cdot)$ is a (convex) penalty, *e.g.* one that encourages robustness:

$$\pi(A, B, C, D) \propto \frac{1}{2} \frac{\|B\|_\infty^2 + \|C\|_\infty^2}{1 - \|A\|_\infty} + \|D\|_\infty.$$

- May also incorporate penalties to encourage sparsity, low-rank, etc., *e.g.*:

$$\sum_{i \in [p]} \|Be_i\|_\infty$$

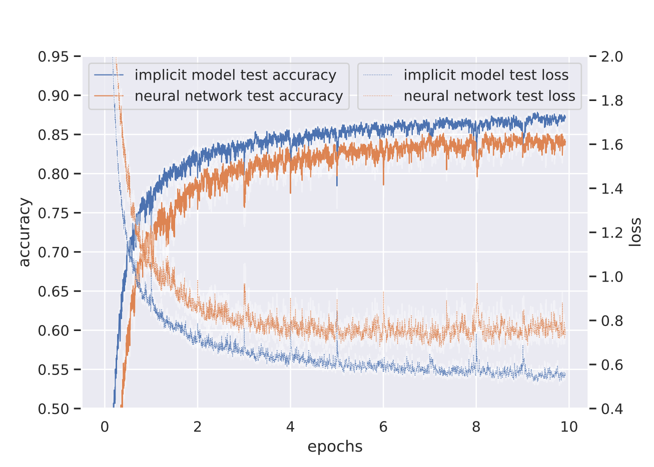
encourages entire columns of B to be zero, for feature selection.

PROJECTED (SUB) GRADIENT

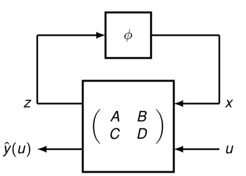
SGD can be adapted to the problem:

- Differentiating through the equilibrium equation is possible.
- Need to deal with the constraint of well-posedness via projection.
- Projection on constraint $\|A\|_\infty \leq \kappa$ can be done extremely fast using (vectorized) bisection, solving for each row of A in parallel.
- Can extend to Frank-Wolfe methods, which are suited to seeking sparse models.

EXAMPLE: TRAFFIC SIGN DATA SET

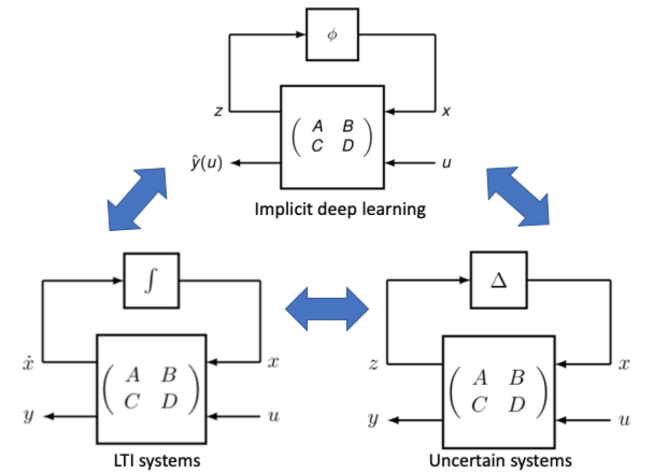


TAKE-AWAYS



- Implicit models are more general than standard neural networks.
- Well-posedness is a key property that can be enforced via norm or eigenvalue conditions.
- Models can be composed together in modular fashion.
- The notationally very simple framework allows for rigorous analyses for robustness, model compression, architecture optimization, etc.
- The corresponding training problem is amenable to SGD methods.

TOWARDS A GENERAL THEORY?



THANK YOU!