# AMP-inspired Deep Networks

**Phil Schniter**

THE OHIO STATE UNIVERSITY Duke UNIVERSITY iiD

Collaborators: **Sundeep Rangan** (NYU), **Alyson Fletcher** (UCLA), **Mark Borgerding** (OSU)

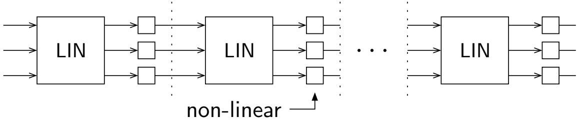
Supported in part by NSF grants IIP-1539960 and CCF-1527162.

Workshop on Information Theory and Applications (ITA) — Feb 2017

# Deep Neural Networks

Typical setup:

- Many layers, consisting of (affine) linear stages and scalar nonlinearities.




- Linear stages often constrained (e.g., small convolution kernels).
- Parameters learned by minimizing training error using backprop.

Open questions:

1. How should we interpret the learned parameters?
2. Can we speed up training?
3. Can we design a better network structure?

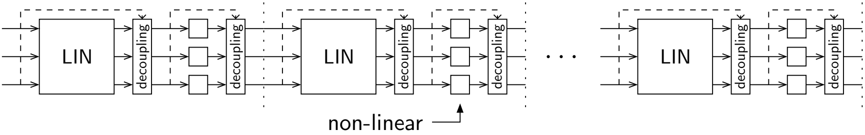
# Focus of this talk: Standard Linear Regression

- Consider recovering a vector $\boldsymbol{x}$ from noisy linear observations

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w},$$

  where $\boldsymbol{x}$ is drawn from an iid prior (e.g., sparse[1])

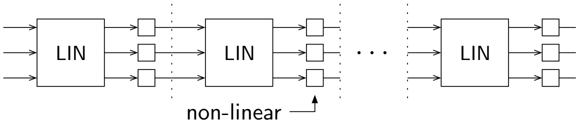
- For this application, we propose a deep network that is
  1) asymptotically optimal for a large class of $\boldsymbol{A}$,
  2) interpretable, and
  3) easy to train.




[1] Gregor/LeCun, Sprechmann/Bronstein/Sapiro, Kamilov/Mansour, Wang/Ling/Huang, Mousavi/Baraniuk, Borgerding/Schniter, etc.

# Understanding Deep Networks via Algorithms

- Many algorithms have been proposed for high-dimensional inference.
- Often, such algorithms are iterative, where each iteration consists of a linear operation followed by scalar nonlinearities.
- By "unfolding" such algorithms, we get deep networks.[2]




- Can such algorithms help us design/interpret/train deep nets?

[2]Gregor/LeCun, ICML 10.

# Algorithmic Approaches to Standard Linear Regression

- Recall goal: recovering/fitting $\boldsymbol{x}$ from noisy linear observations

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}.$$

- A popular approach is regularized loss minimization:

$$\arg\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda f(x),$$

  where, e.g., $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ for the lasso.

- Can also be interpreted as MAP estimation of $\boldsymbol{x}$ under priors

$$\boldsymbol{x} \sim \exp(-f(\boldsymbol{x})) \quad \& \quad \boldsymbol{w} \sim \mathcal{N}(0, \lambda \boldsymbol{I}).$$

- But often the goal is minimizing MSE or inferring marginal posteriors.

# High-dimensional MMSE Inference

- High dimensional MMSE inference is difficult in general.
- To simplify things, suppose that 1) $\boldsymbol{x}$ is iid
  2) $\boldsymbol{A}$ is large and random.
- The case of iid Gaussian $\boldsymbol{A}$ is well studied, but very restrictive.
- Instead, consider right-rotationally invariant $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}} \text{ with } \boldsymbol{V} \sim \text{Haar and indep of } \boldsymbol{x}.$$

- For this case, the replica prediction of the MMSE is[3]

$$\mathcal{E}(\gamma) = \operatorname{var}\{x|r\}, \quad r = x + \mathcal{N}(0, 1/\gamma), \quad \gamma = R_{\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}/\sigma^2}(-\mathcal{E}(\gamma))$$

[3] Tulino/Caire/Verdu/Shamai, IEEE-TIT, 2013.

# Achieving MMSE in standard linear regression

- Recently a "vector approximate message passing" (VAMP) algorithm has been proposed that iterates linear vector estimation with nonlinear scalar denoising.

- Under large right-rotationally invariant $\boldsymbol{A}$ and Lipschitz scalar denoisers, VAMP is rigorously characterized by a scalar state-evolution.[4]

- Under MMSE scalar denoising, VAMP's state-evolution fixed-points agree with the replica prediction!

- Operating regimes:
  **easy**: VAMP has a unique fixed point $\Rightarrow$ attains MMSE
  **hard**: VAMP has multiple fixed points $\Rightarrow$ suboptimal
  (but so are all other known polynomial-time methods)

[4] Rangan/Schniter/Fletcher, arXiv:1610.03082, 2016.

# VAMP for linear regression

initialize $\boldsymbol{r}_1, \gamma_1$

for $t = 0, 1, 2, \ldots$

$$\widehat{\boldsymbol{x}}_1 \leftarrow \left(\boldsymbol{A}^\top \boldsymbol{A}/\sigma^2 + \gamma_1 \boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}^\top \boldsymbol{y}/\sigma^2 + \gamma_1 \boldsymbol{r}_1\right) \qquad \text{LMMSE}$$

$$\alpha_1 \leftarrow \tfrac{\gamma_1}{N} \operatorname{Tr}\left[\left(\boldsymbol{A}^\top \boldsymbol{A}/\sigma^2 + \gamma_1 \boldsymbol{I}\right)^{-1}\right] \qquad \text{divergence}$$

$$\boldsymbol{r}_2 \leftarrow \tfrac{1}{1-\alpha_1}\left(\widehat{\boldsymbol{x}}_1 - \alpha_1 \boldsymbol{r}_1\right) \qquad \text{Onsager correction}$$

$$\gamma_2 \leftarrow \gamma_1 \tfrac{1-\alpha_1}{\alpha_1} \qquad \text{precision of } \boldsymbol{r}_2$$

---

$$\widehat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{g}(\boldsymbol{r}_2; \gamma_2) \qquad \text{(scalar) denoising}$$

$$\alpha_2 \leftarrow \tfrac{1}{N} \operatorname{Tr}\left[\tfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}}(\boldsymbol{r}_2; \gamma_2)\right] \qquad \text{divergence}$$

$$\boldsymbol{r}_1 \leftarrow \tfrac{1}{1-\alpha_2}\left(\widehat{\boldsymbol{x}}_2 - \alpha_2 \boldsymbol{r}_2\right) \qquad \text{Onsager correction}$$

$$\gamma_1 \leftarrow \gamma_2 \tfrac{1-\alpha_2}{\alpha_2} \qquad \text{precision of } \boldsymbol{r}_1$$

end

# MMSE-VAMP interpreted

initialize $\boldsymbol{r}_1, \gamma_1$

for $t = 0{,}1, 2, \ldots$

$\quad\widehat{\boldsymbol{x}}_1 \leftarrow$MMSE estimate of $\boldsymbol{x}$ under

$\qquad$pseudo-prior $\mathcal{N}(\boldsymbol{r}_1, \boldsymbol{I}/\gamma_1)$ & measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$

$\quad\boldsymbol{r}_2 \leftarrow$linear cancellation of $\boldsymbol{r}_1$ from $\widehat{\boldsymbol{x}}_1$

$\quad\widehat{\boldsymbol{x}}_2 \leftarrow$MMSE estimate of $\boldsymbol{x}$ under

$\qquad$prior $p(\boldsymbol{x})$ and pseudo-measurement $\boldsymbol{r}_2 = \boldsymbol{x} + \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}/\gamma_2)$

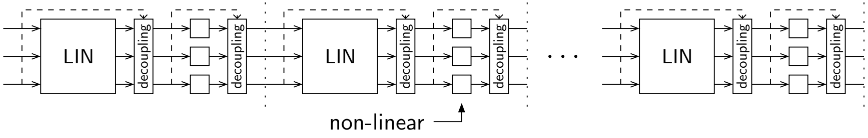
$\quad\boldsymbol{r}_1 \leftarrow$linear cancellation of $\boldsymbol{r}_2$ from $\widehat{\boldsymbol{x}}_2$

end

Linear cancellation "decouples" the iterations, so that global MMSE problem can be tackled by solving simpler local MMSE problems.

# Unfolding VAMP

Unfolding the VAMP algorithm gives the network[5]




Notice the two decoupling stages in each layer.

[5] Borgerding/Schniter'16

# Learning the network parameters

After unfolding an algorithm, one can use backpropagation (or similar) to "learn" the optimal network parameters.[6]

- Linear stage: $\widehat{\boldsymbol{x}}_1 = \boldsymbol{B}\boldsymbol{r}_1 + \boldsymbol{C}\boldsymbol{y}$
  $\rightarrow$ learn $(\boldsymbol{B}, \boldsymbol{C})$ for each layer.
- Nonlinear stage: $\widehat{x}_{2j} = g(r_{2j})\ \forall j$
  $\rightarrow$ learn a scalar function $g(\cdot)$ for each layer.
  e.g., spline, piecewise linear, etc.

[6]Gregor/LeCun, ICML 10.

# Result of learning

Suppose that the training data $\{\boldsymbol{y}^{(d)}, \boldsymbol{x}^{(d)}\}_{d=1}^{D}$ were constructed using
1) iid $x_j^{(d)} \sim p(x)$
2) $\boldsymbol{y}^{(d)} = \boldsymbol{A}\boldsymbol{x}^{(d)} + \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$
3) right-rotationally invariant $\boldsymbol{A}$.

Backpropagation recovers the parameter settings $(\boldsymbol{B}, \boldsymbol{C}, f)$ originally prescribed by the VAMP algorithm!

- The learned linear stages are MMSE under pseudo-prior $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{r}_1, \boldsymbol{I}/\gamma_1)$ & measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$
- The learned scalar nonlinearities are MMSE under prior $p(x_j)$ and pseudo-measurements $r_{2j} = x_j + \mathcal{N}(0, 1/\gamma_2)$

$\rightsquigarrow$ *This deep network is interpretable!*

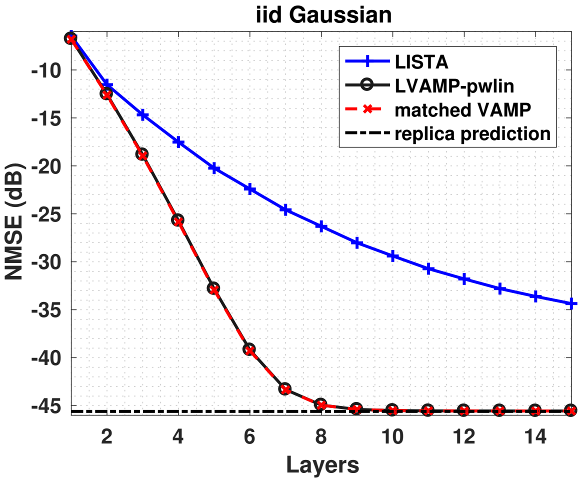
# Implications for training

Due to the decoupling stages...

- each linear stage should be locally MSE optimal, so
  - $\rightsquigarrow$ *the linear params* $(\boldsymbol{B}, \boldsymbol{C})$ *can be learned locally in each layer!*
- each non-linear stage should be locally MSE optimal, so
  - $\rightsquigarrow$ *the nonlinear function* $g(\cdot)$ *can be learned locally in each layer!*

*This deep network is easy to train!*

# Example with iid Gaussian $\boldsymbol{A}$



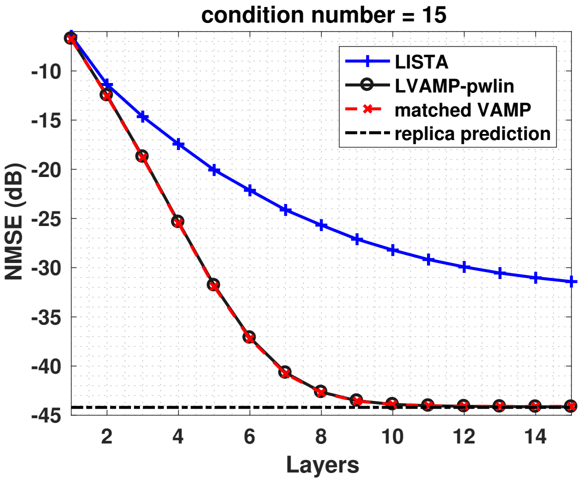

$n = 1024$
$m/n = 0.5$

$\boldsymbol{A} \sim$ iid $\mathcal{N}(0,1)$

$x \sim$ Bernoulli-Gaussian
$\Pr\{x \neq 0\} = 0.1$

SNR $= 40$ dB

# Example with non-iid Gaussian $\boldsymbol{A}$



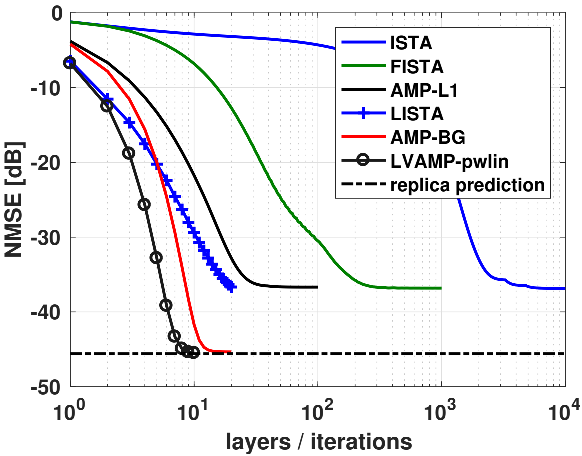

$n = 1024$
$m/n = 0.5$

$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\mathsf{T}$
$\boldsymbol{U}, \boldsymbol{V} \sim$ Haar
$s_n/s_{n-1} = \phi \ \forall n$

$x \sim$ Bernoulli-Gaussian
$\Pr\{x \neq 0\} = 0.1$

SNR $= 40$ dB

# Deep nets vs algorithms




$n = 1024$
$m/n = 0.5$

$\boldsymbol{A} \sim$ iid $\mathcal{N}(0,1)$

$x \sim$ Bernoulli-Gaussian
$\Pr\{x \neq 0\} = 0.1$

SNR $= 40$ dB

# Conclusions

- Our goal is to understand the design and interpretation of deep nets.
- For this talk, we restricted our focus to the problem of (e.g., sparse) linear regression.
- We proposed a deep net that is
  1) asymptotically MSE-optimal (for iid $\boldsymbol{x}$ and RRI $\boldsymbol{A}$)
  2) interpretable: ...LMMSE/decoupling/NL-MMSE/decoupling ...
  3) locally trainable.
- The proposed network is obtained by "unfolding" the VAMP algorithm and learning its parameters.
- In ongoing work, we are extending these ideas beyond linear regression.