Building Recurrent Networks by Unfolding Iterative Thresholding for Sequential Sparse Recovery

Scott Wisdom[1] Thomas Powers[1] James Pitton[1,2] Les Atlas[1]
[1]Department of Electrical Engineering, University of Washington, Seattle, WA, USA
[2]Applied Physics Laboratory, University of Washington, Seattle, WA, US

Problem statement

- While effective, conventional deep network architectures are designed by trial-and-error and are thus difficult to interpret and improve

Contribution

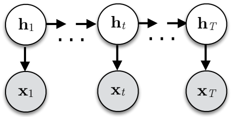
- We construct a principled deep recurrent neural network, the SISTA-RNN, from an existing sequential sparse recovery model
- The SISTA-RNN has distinct advantages:
  - Trains faster
  - Achieves better performance than conventional deep networks
  - Has interpretable weights

Method

- Deep unfolding constructs deep networks from inference algorithms for statistical models

Statistical model → **Inference algorithm** → **Unfolded deep network**

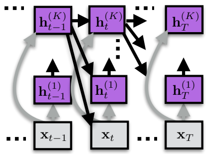
Sequential sparse recovery



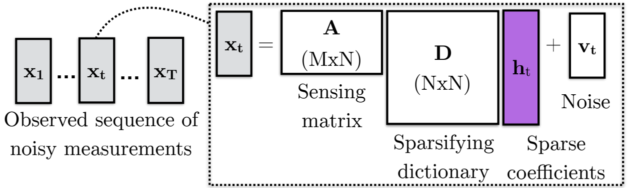
Sequential iterative soft-thresholding algorithm (SISTA)

for t=1:T
  for k=1:K
    $\hat{\mathbf{h}}_t^{(k)} = \sigma_b(\mathbf{x}_t, \hat{\mathbf{h}}_t^{(k-1)})$

SISTA-RNN



Statistical model: sequential sparse coding



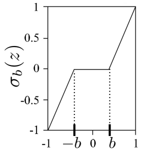
- Inference of $\mathbf{h}$ minimizes the negative log-likelihood:

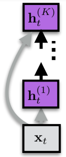
$$\underset{\mathbf{h}_{1:T}}{\text{minimize}} \sum_{t=1}^{T} \Big(\underbrace{\frac{1}{2}\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2 + \lambda_1\|\mathbf{h}_t\|_1}_{\text{LASSO}} + \underbrace{\frac{\lambda_2}{2}\|\mathbf{D}\mathbf{h}_t - \mathbf{F}\mathbf{D}\mathbf{h}_{t-1}\|_2^2}_{\text{Temporal model}} \Big)$$

Inference algorithm: iterative soft-thresholding

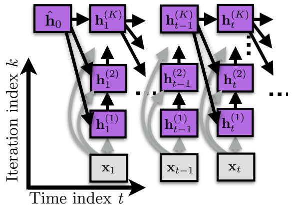
- Iterative soft-thresholding algorithm (ISTA) solves the LASSO

ISTA:
for $k = 1 : K$,
$$h^{(k)} = \sigma_b\left\{z(x, h^{(k-1)})\right\}$$
$$\sigma_b(z) = \max\{|z| + b\}\,\mathrm{sign}\{z\}$$

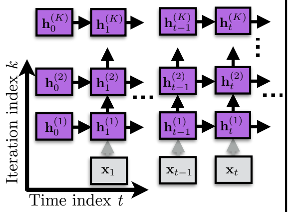




Unfolded deep network: sequential ISTA RNN (SISTA-RNN)



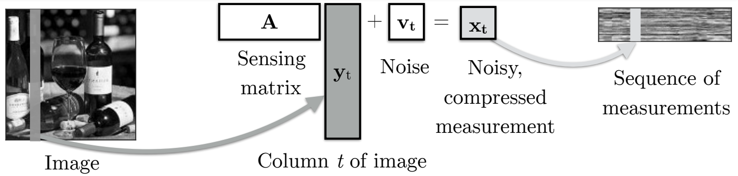
SISTA-RNN



Conventional generic RNN stack

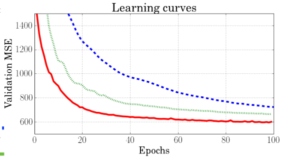
Experiment: column-wise compressive sensing of images

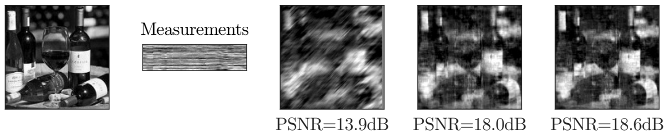
- Goal: recover Caltech-256 images from noisy compressed measurements



- Results: SISTA-RNN trains fastest and achieves best performance

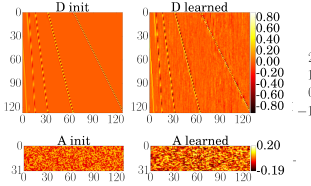
	Algorithm	Oracle?	# iter. K	# tr. I	MSE	PSNR (dB)
Baselines	SISTA	No	3	None	4740	12.1
	SISTA to convergence	No	$\leq$ 1825	None	3530	13.4
	SSpaRSA to convergence	No	$\leq$ 420	None	3520	13.4
	SISTA	Yes	3	None	4160	13.3
	SISTA to convergence	Yes	$\leq$ 694	None	2400	15.0
	SSpaRSA to convergence	Yes	$\leq$ 225	None	2440	15.0
	ℓ_1-homotopy	Yes	$\leq$ 314	None	1490	17.1
	Generic RNN, rand. init.	No	3	24885	720	20.7
Proposed	Trained SISTA-RNN, rand. init.	No	3	24485	637	21.2
	Trained SISTA-RNN, SISTA init.	No	3	24485	**541**	**22.2**

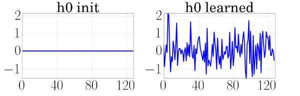




- The learned SISTA-RNN weights are interpretable

λ_1 init :0.5
λ_1 learned :3.07
λ_2 init :1.0
λ_2 learned : -0.04
α init :1.0
α learned :2.02





This work funded by ONR contract N00014-12-G-0078, delivery order 0013

Code available at github.com/stwisdom/sista-rnn