LISTA: Concerns and Extensions

Zachary Levine

Signal Sampling Final Project Presentation

February 5, 2023

Paper

Learning Fast Approximations of Sparse Coding, by Karol Gregor and Yann LeCun [2].

The good

- Strong numerical evidence

The good

- Strong numerical evidence
- The presented algorithms converge in less time and iterations than the standard ISTA and CoD algorithms

Theoretical concerns: Straw man fallacy

- Why use $\tanh(x)$?

Theoretical concerns: Straw man fallacy

- Why use $\tanh(x)$?
- Interaction terms?

Practical concerns

- Dictionary learning dependence

Practical concerns

- Dictionary learning dependence
- Not actually "learning" ISTA

Practical concerns

- Dictionary learning dependence
- Not actually "learning" ISTA
- What's the point?

Practical concerns

- Dictionary learning dependence
- Not actually "learning" ISTA
- What's the point?
- Sensitivity analysis

Practical concerns

- Dictionary learning dependence
- Not actually "learning" ISTA
- What's the point?
- Sensitivity analysis
- Why learn a dictionary if the one we feed to the ISTA to generate the ideal codes to train against is fixed?

Extensions: Motivation

We want a model that:

Extensions: Motivation

We want a model that:

- Learns ISTA implicitly from the data without having to run ISTA along side training the encoder to give as training data.

Extensions: Motivation

We want a model that:

- Learns ISTA implicitly from the data without having to run ISTA along side training the encoder to give as training data.
- Has no dependence on learning W_d before we even train the neural network.

What to use?

Extensions: Motivation

We want a model that:

- Learns ISTA implicitly from the data without having to run ISTA along side training the encoder to give as training data.
- Has no dependence on learning W_d before we even train the neural network.

What to use? an autoencoder

Autoencoder

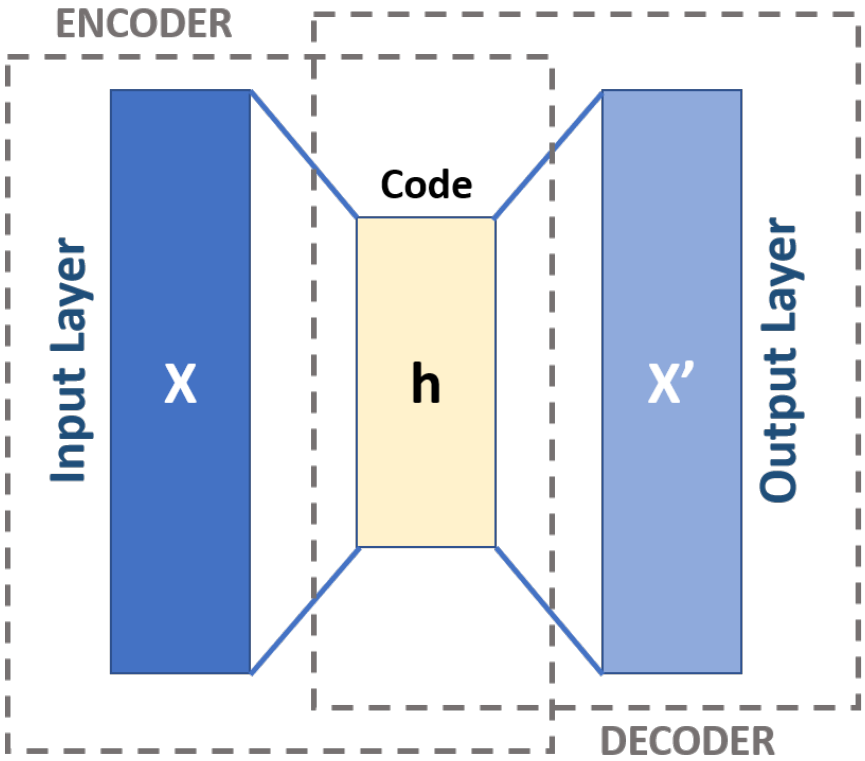

Figure: The basic structure of an autoencoder [3].

Data

Figure: Four sample images from the Berkley Segmentation Dataset

Steps

Our experiment is composed of the following steps.

Steps

Our experiment is composed of the following steps.

1. Implement proximal gradient descent for W_d

Steps

Our experiment is composed of the following steps.

1. Implement proximal gradient descent for W_d
2. Implement FISTA [1]

Steps

Our experiment is composed of the following steps.

1. Implement proximal gradient descent for W_d
2. Implement FISTA [1]
3. Implement LISTA [2]

Steps

Our experiment is composed of the following steps.

1. Implement proximal gradient descent for W_d
2. Implement FISTA [1]
3. Implement LISTA [2]
4. Implement Unsupervised Autoencoder based LISTA [4]

Steps

Our experiment is composed of the following steps.

1. Implement proximal gradient descent for W_d
2. Implement FISTA [1]
3. Implement LISTA [2]
4. Implement Unsupervised Autoencoder based LISTA [4]
5. Compare our results to baseline

Training the dictionary, FISTA

- We found that training on a random sample of 10 patches from our set yielded superior results as opposed to all patches from one image for which the training set was much larger.
- This is likely because a random sample of patches may better represent the distribution of all patches than patches from a single image.
- We trained our dictionary and then ran FISTA for several images by:

Training the dictionary, FISTA

- We found that training on a random sample of 10 patches from our set yielded superior results as opposed to all patches from one image for which the training set was much larger.
- This is likely because a random sample of patches may better represent the distribution of all patches than patches from a single image.
- We trained our dictionary and then ran FISTA for several images by:
  1. cutting them into patches

Training the dictionary, FISTA

- We found that training on a random sample of 10 patches from our set yielded superior results as opposed to all patches from one image for which the training set was much larger.
- This is likely because a random sample of patches may better represent the distribution of all patches than patches from a single image.
- We trained our dictionary and then ran FISTA for several images by:
  1. cutting them into patches
  2. finding the sparse codes for each patch

Training the dictionary, FISTA

- We found that training on a random sample of 10 patches from our set yielded superior results as opposed to all patches from one image for which the training set was much larger.
- This is likely because a random sample of patches may better represent the distribution of all patches than patches from a single image.
- We trained our dictionary and then ran FISTA for several images by:
  1. cutting them into patches
  2. finding the sparse codes for each patch
  3. multiplying the sparse codes by W_d

Training the dictionary, FISTA

- We found that training on a random sample of 10 patches from our set yielded superior results as opposed to all patches from one image for which the training set was much larger.
- This is likely because a random sample of patches may better represent the distribution of all patches than patches from a single image.
- We trained our dictionary and then ran FISTA for several images by:
  1. cutting them into patches
  2. finding the sparse codes for each patch
  3. multiplying the sparse codes by W_d
  4. reconstructing the result from its constituent patches

Training LISTA

As a second comparator, we implemented the LISTA model as described by Yan and LeCun in Pytorch with ten unfolded iterations. We trained for 10 epochs on random patches of 10 images with an Adam Optimizer and a learning rate of 0.001.

Unsupervised LISTA: First attempts

- We tried to train an autoencoder based on the identical architecture presented by Gregor and LeCun, with a simple multiplication by the learned W_d as the final layer.
- We set the loss to be the standard MSE loss between the original image and the reconstruction, instead of against the FISTA sample.

Unsupervised LISTA: First attempts

- We tried to train an autoencoder based on the identical architecture presented by Gregor and LeCun, with a simple multiplication by the learned W_d as the final layer.
- We set the loss to be the standard MSE loss between the original image and the reconstruction, instead of against the FISTA sample.
- This approach did not work at all.
- While the model could learn codes for patches and reach a minimum MSE of 0.001, the reconstructed images looked nothing like the original images.

Unsupervised LISTA: First attempts

- We tried to train an autoencoder based on the identical architecture presented by Gregor and LeCun, with a simple multiplication by the learned W_d as the final layer.
- We set the loss to be the standard MSE loss between the original image and the reconstruction, instead of against the FISTA sample.
- This approach did not work at all.
- While the model could learn codes for patches and reach a minimum MSE of 0.001, the reconstructed images looked nothing like the original images.
- In fact, the reconstructed images all were composed of horizontal lines with zero likeness to the original image.

Unsupervised LISTA: First attempts

- We tried to train an autoencoder based on the identical architecture presented by Gregor and LeCun, with a simple multiplication by the learned W_d as the final layer.
- We set the loss to be the standard MSE loss between the original image and the reconstruction, instead of against the FISTA sample.
- This approach did not work at all.
- While the model could learn codes for patches and reach a minimum MSE of 0.001, the reconstructed images looked nothing like the original images.
- In fact, the reconstructed images all were composed of horizontal lines with zero likeness to the original image.

Unsupervised LISTA: Key Takeaways

Our results suggest that more complex and perhaps domain-specific architecture is required to learn ISTA without training data.

Relevant Literature

In Learned Convolutional Sparse Coding, Hillel Sreter and Raja Giryes propose a novel convolutional recurrent sparse autoencoder, a convolutional extension of LISTA with a linear convolutional decoder [4].

$$Z(k+1) = h_\theta\left(W_e X + S Z(k)\right) \quad Z(0) = 0 \longrightarrow \mathbf{z}_{k+1} = S_\theta\left(\mathbf{z}_k + \mathbf{w_e} * \left(\mathbf{x} - \mathbf{w_d} * \mathbf{z}_k\right)\right)$$

Architecture

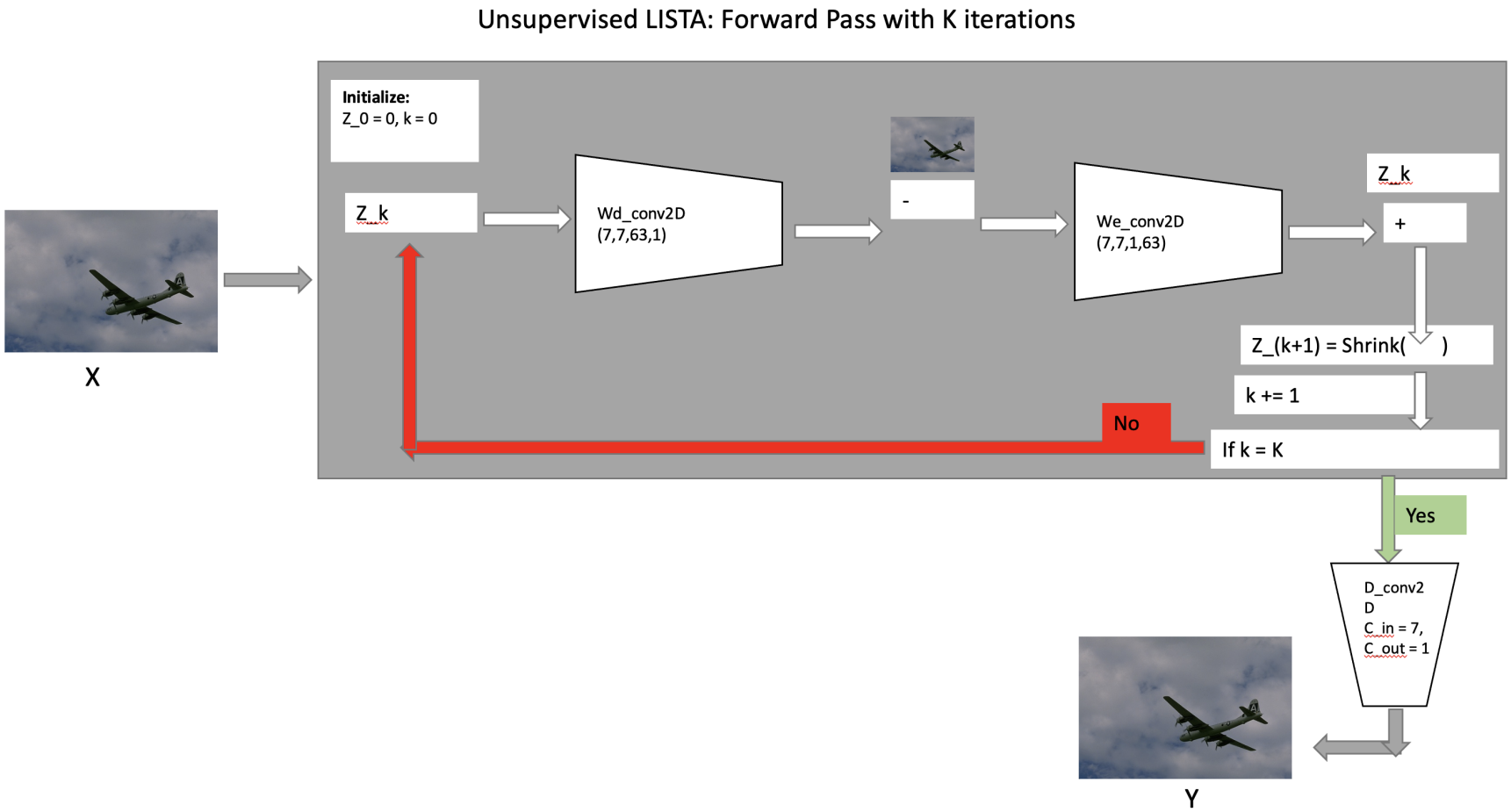

Figure: Architecture of the unsupervised convolutional autoencoder model [4].

Architecture

- We trained this network on the same Berkley Segmentation dataset, for each epoch grabbing a random image

Architecture

- We trained this network on the same Berkley Segmentation dataset, for each epoch grabbing a random image
- We used a simple MSE loss and an SGD optimizer with learning rate of 0.001, for 10 epochs with 10 iterations of unfolded convolutional ISTA performed in each forward pass.

Architecture

- We trained this network on the same Berkley Segmentation dataset, for each epoch grabbing a random image
- We used a simple MSE loss and an SGD optimizer with learning rate of 0.001, for 10 epochs with 10 iterations of unfolded convolutional ISTA performed in each forward pass.
- Training this way takes a total of 1 minute and 20 seconds.

Architecture

- We trained this network on the same Berkley Segmentation dataset, for each epoch grabbing a random image
- We used a simple MSE loss and an SGD optimizer with learning rate of 0.001, for 10 epochs with 10 iterations of unfolded convolutional ISTA performed in each forward pass.
- Training this way takes a total of 1 minute and 20 seconds.
- This network does train, but we wondered how well, and how our results compared to standard LISTA.

Comparing Dictionaries

We can see that the filters from two dictionaries seem to be similar in structure.

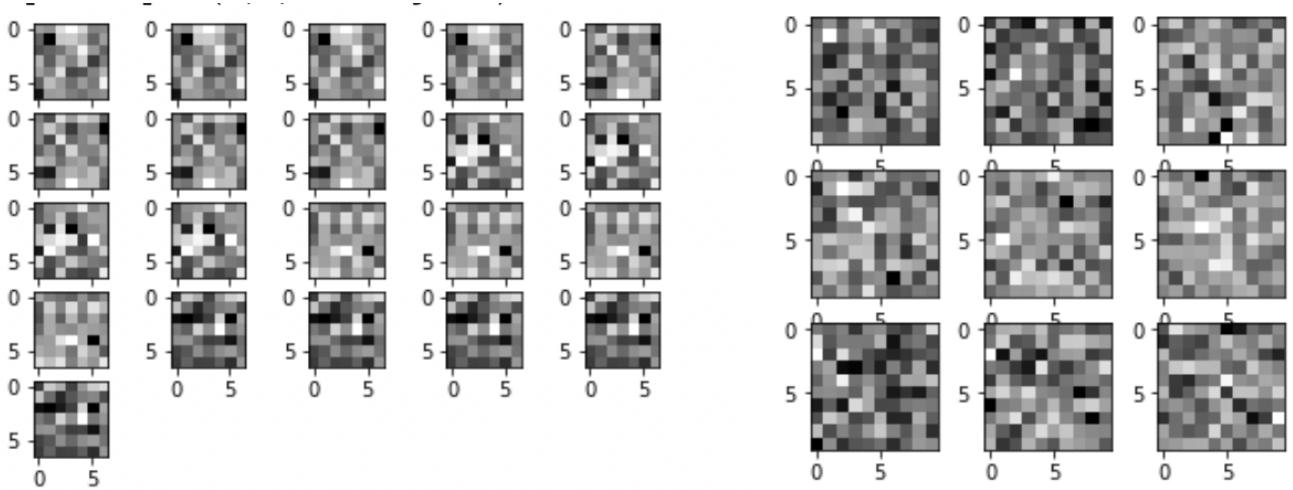

Figure: The first 25 filters from the learned dictionary used as the weight matrix in convolution W_d (left). The first 9 filters from the learned dictionary W_d from the Berkley Segmentation Dataset based on 10 epochs, sparsity parameter $\alpha = 0.00001$, and a learning rate of 0.01 (right).

Comparison Metrics

We compare the autoencoder to LISTA in three metrics:

1. Visual reconstruction similarity
2. Reconstruction error per epoch
3. Sparsity of solutions obtained

Comparison: Visual Similarity

We compare the autoencoder to LISTA in three metrics:

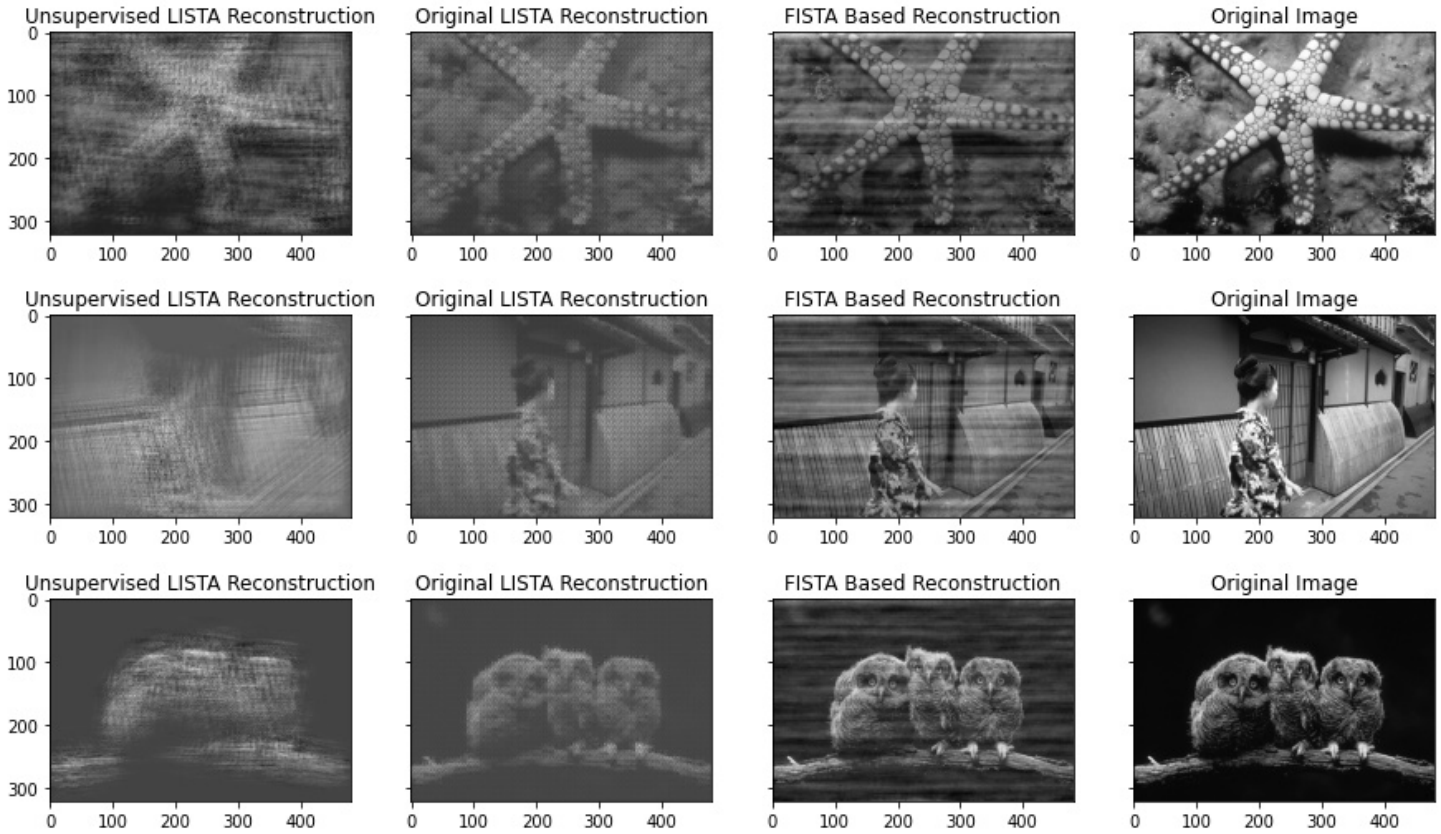

Figure: Comparing reconstructions from the model to LISTA with identical parameters (10 epochs, and 10 unfolded ISTA iterations), as well as standard FISTA and the original images, on images held out from the

Comparison: Visual Similarity Pt. 2

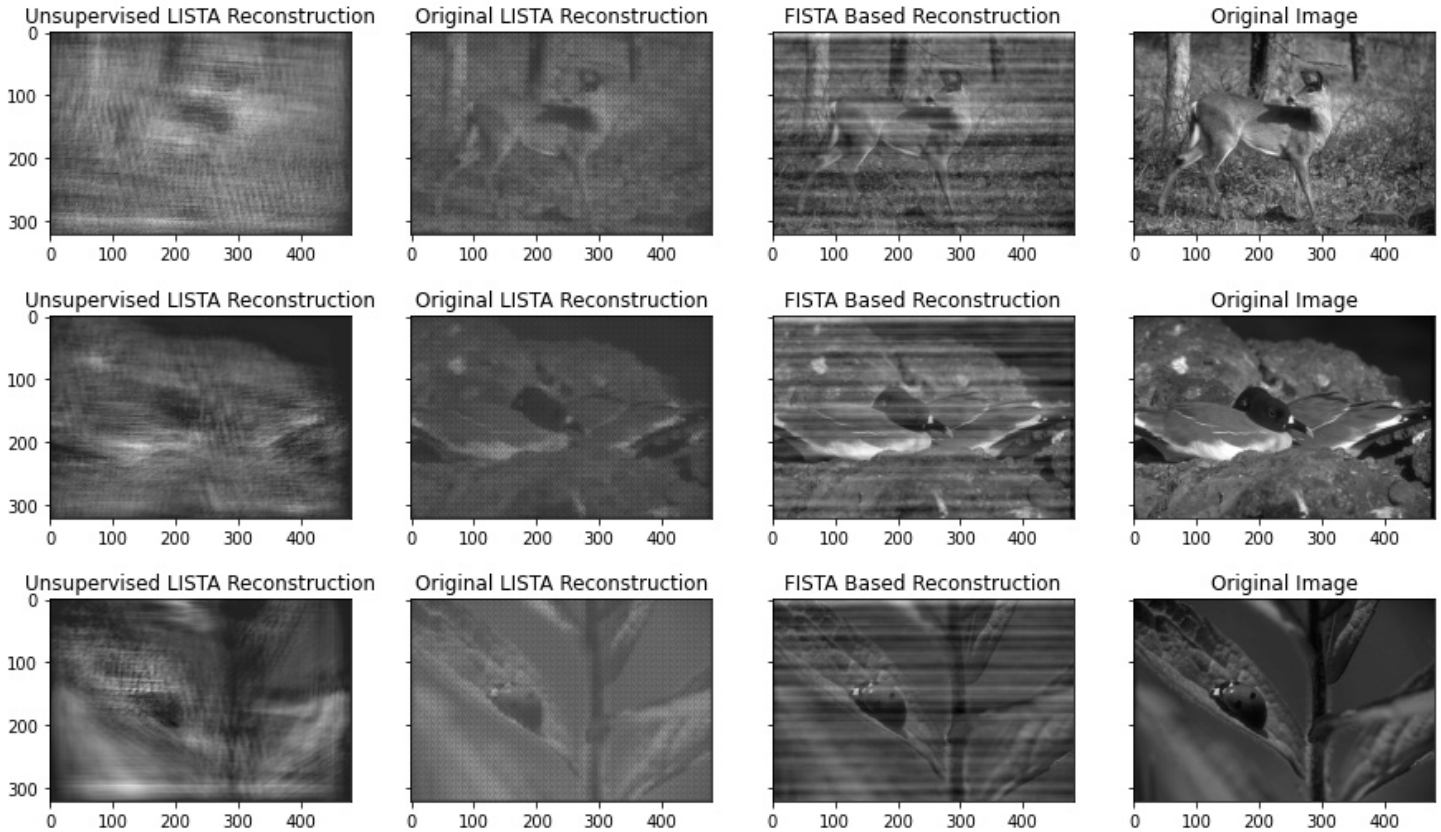

Figure: Comparing reconstructions from the model to LISTA with identical parameters (10 epochs, and 10 unfolded ISTA iterations), as well as standard FISTA and the original images, on images held out from the training set

Visual Similarity: Key Takeaways

- First, as seen in the above results, we can see that our reconstruction does recover some of the original image, but the original LISTA and FISTA reconstructions are closer to the original image than given epochs, learning rate, and iteration numbers.
- Perhaps more training or different hyper parameters are required.
- Still, the fact that the model was able to learn anything at all is interesting, because there was was no supervisory signal telling the network to do LISTA.

MSE Comparison

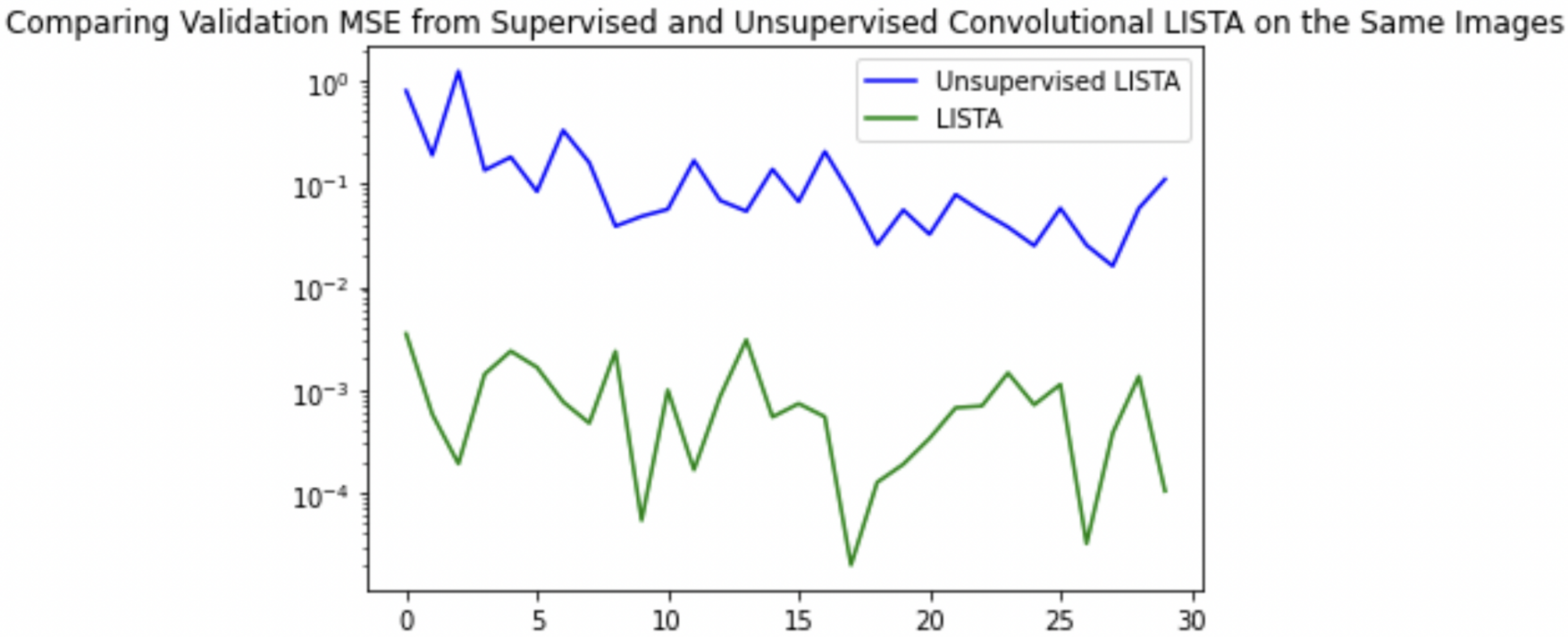

Figure: Comparing validation loss between standard LISTA and convolutional unsupervised LISTA on the same images with identical training parameters over 30 epochs. The y axis of the figure is shown on a log scale.

Sparsity Comparison

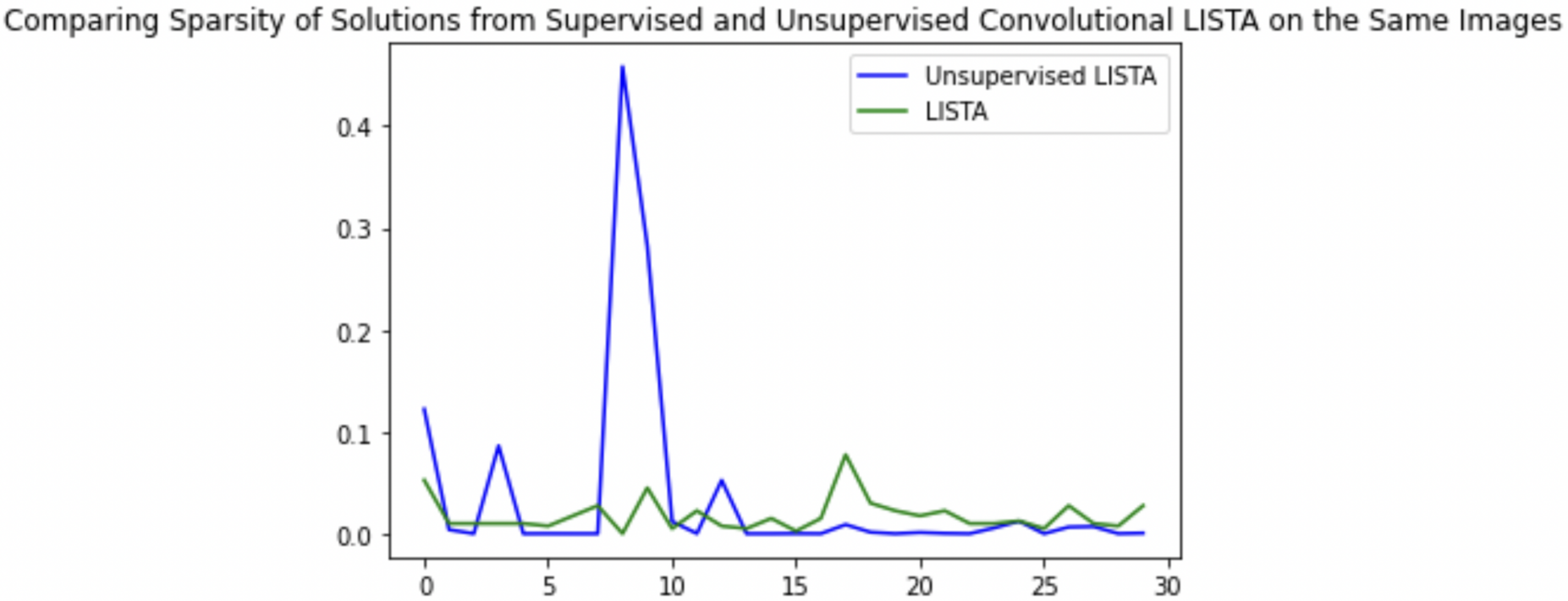

Figure: Comparing sparsity between standard LISTA and convolutional unsupervised LISTA on the same images with identical training parameters over 30 epochs.

Conclusions

1. It was argued that while there is value in LISTA in terms of its numerical dominance over ISTA and other non learned methods, the main theoretical grounding presented by the authors is shaky.

Conclusions

1. It was argued that while there is value in LISTA in terms of its numerical dominance over ISTA and other non learned methods, the main theoretical grounding presented by the authors is shaky.
2. Inspired by more practical concerns, we considered an extension of LISTA that is unsupervised based on convolutional autoencoders, and saw that it underperformed compared to standard LISTA and FISTA on the same number of iterations in terms both visual reconstruction similarity, accuracy, but matched LISTA on sparsity.

Conclusions

1. It was argued that while there is value in LISTA in terms of its numerical dominance over ISTA and other non learned methods, the main theoretical grounding presented by the authors is shaky.
2. Inspired by more practical concerns, we considered an extension of LISTA that is unsupervised based on convolutional autoencoders, and saw that it underperformed compared to standard LISTA and FISTA on the same number of iterations in terms both visual reconstruction similarity, accuracy, but matched LISTA on sparsity.
3. Further investigation is required to confirm our results and what they mean.

Sources Cited


Amir Beck and Marc Teboulle.
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM Journal on Imaging Sciences, 2(1):183?202, Jan 2009.


Karol Gregor and Yann LeCun.
Learning fast approximations of sparse coding.
In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML?10, page 399?406, Madison, WI, USA, Jun 2010. Omnipress.


Michael Massi.
Papers With Code- Autoencoder.
2021.


Hillel Sreter and Raja Giryes.
Learned convolutional sparse coding.