NYU TANDON SCHOOL OF ENGINEERING

Advanced Project Updating

Haoyu Fang

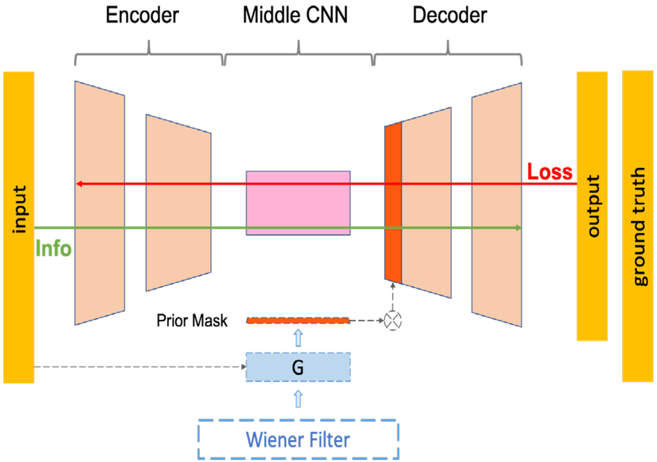
Outline

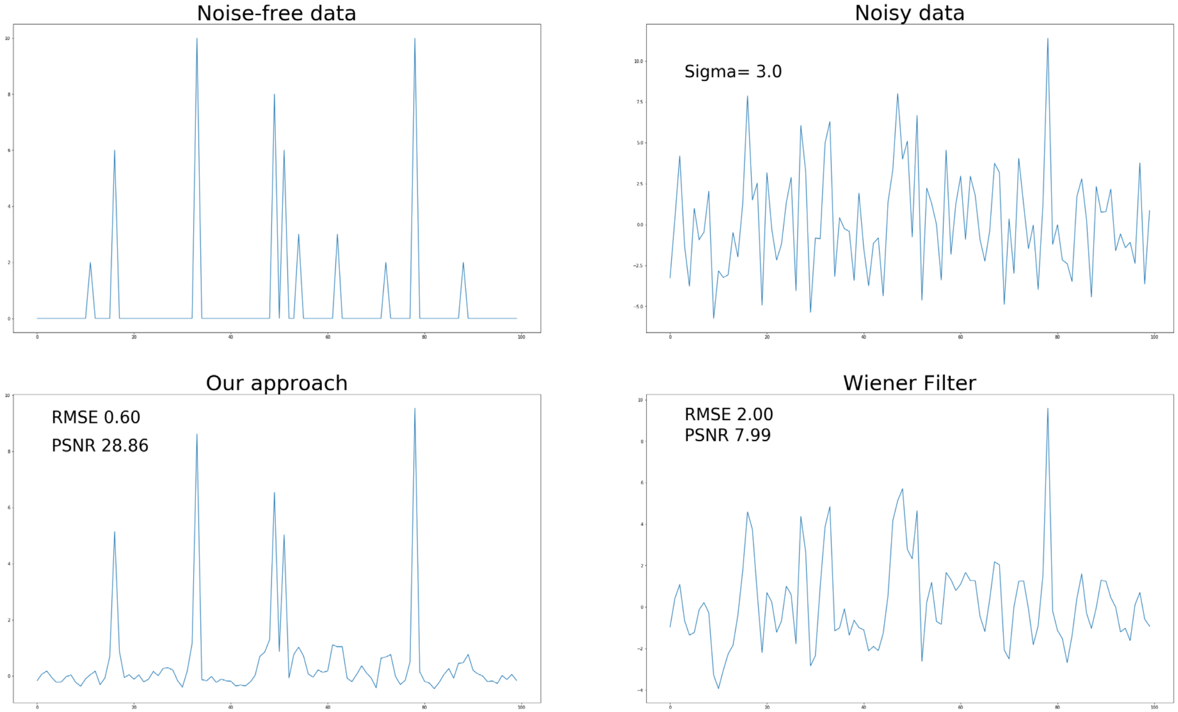
1. **Previous progress**
   1) **An interpretable neural networks with prior guidance**
   2) **Experimental results on signal denoising tasks**
2. Current implementation and experiments
   1) DNN-based denoising algorithm overview
   2) Third-party-library-free implementation (mainly based on Numpy)
   3) Experiments about batch normalization
3. Analysis and inspiration
   1) A signal-processing interpretation of DNN-based deep denoising algorithms
   2) Batch normalization layer with sparse regularization
4. Future work

Previous progress

Results of signal denoising on a synthesized data

The proposed interpretable neural networks with prior guidance





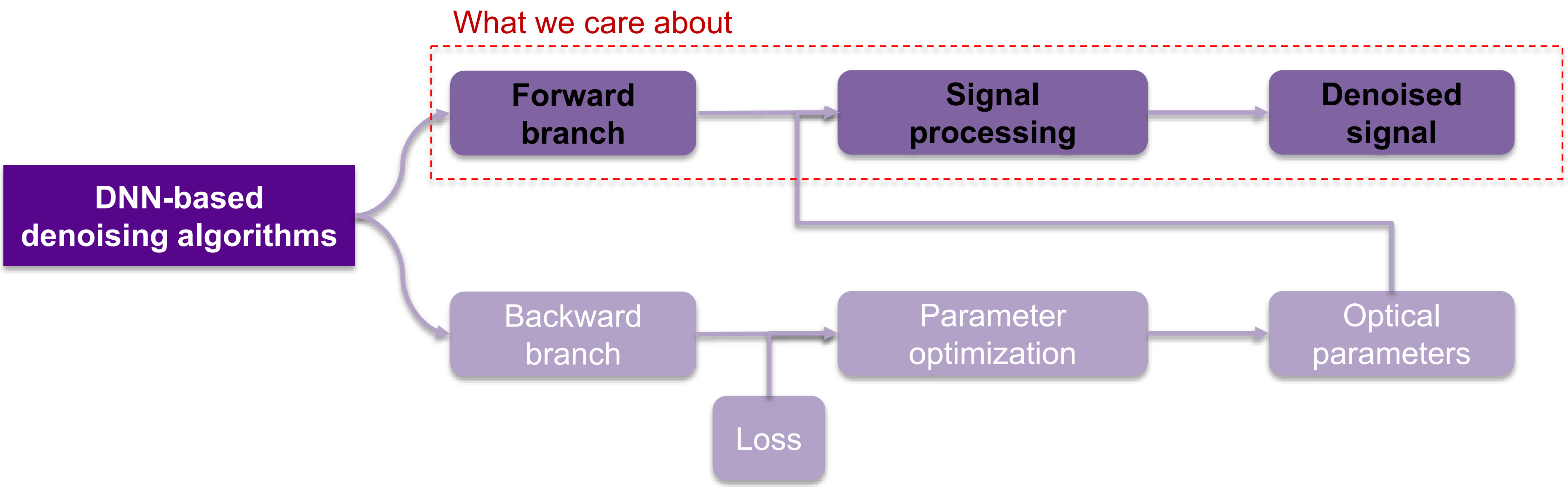
NYU TANDON SCHOOL OF ENGINEERING

Outline

1. Previous progress
    1) An interpretable neural networks with prior guidance
    2) Experimental results on signal denoising tasks
2. **Current implementation and experiments**
    1) **DNN-based denoising algorithm overview**
    2) **Third-party-library-free implementation (mainly based on Numpy)**
    3) **Experiments about batch normalization**
3. Analysis and inspiration
    1) A signal-processing interpretation of DNN-based deep denoising algorithms
    2) Batch normalization layer with sparse regularization
4. Future work

NYU TANDON SCHOOL OF ENGINEERING

DNN-based denoising algorithms



Third-party-library-free implementation

Improved model (weights and bias):

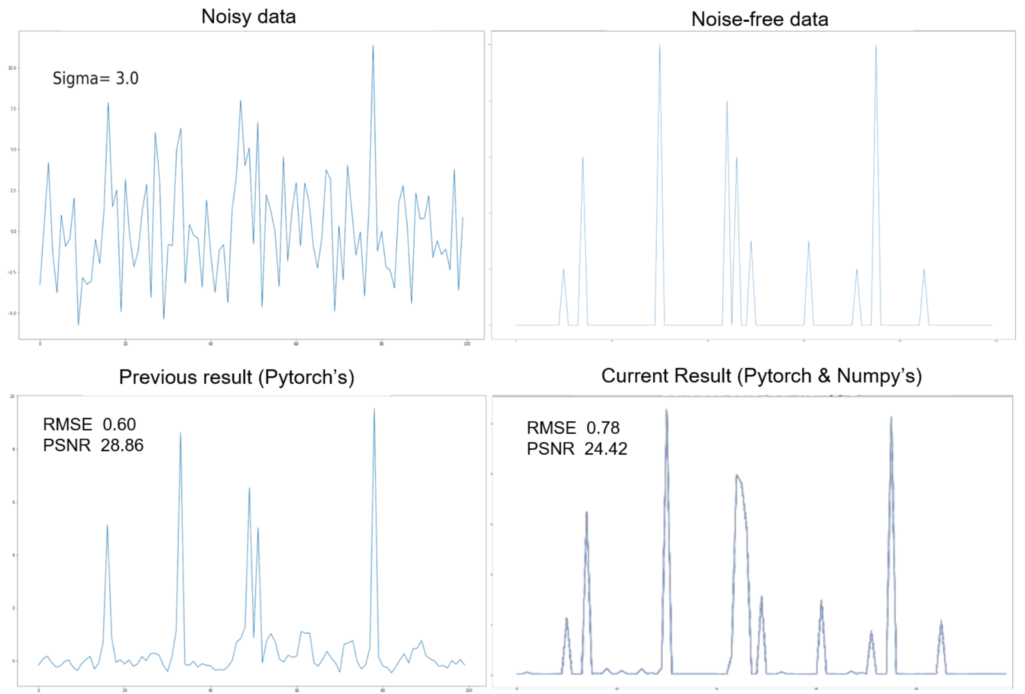
- ❑ Learning rate scheduler
- ❑ Gradient scaling
- ❑ Avoiding local optimums

Results:

- ❑ Stable training
- ❑ High average accuracy
- ❑ Removing most fluctuations
- ❑ Worsen in some cases

Causes:

- ❑ Specific cases follow local optimums' distribution
- ❑ Small training batch size
- ❑ Slight differences in input samples



Third-party-library-free implementation

Implemented components
(details are shown in the report):

- ❑ Convolutional layer

$$out(C_{out_i}, j) = bias(C_{out_i}) + \Sigma_{k=0}^{C_{in}-1} weights(C_{out_i}, k) \odot input,$$

- ❑ Batch normalization layer

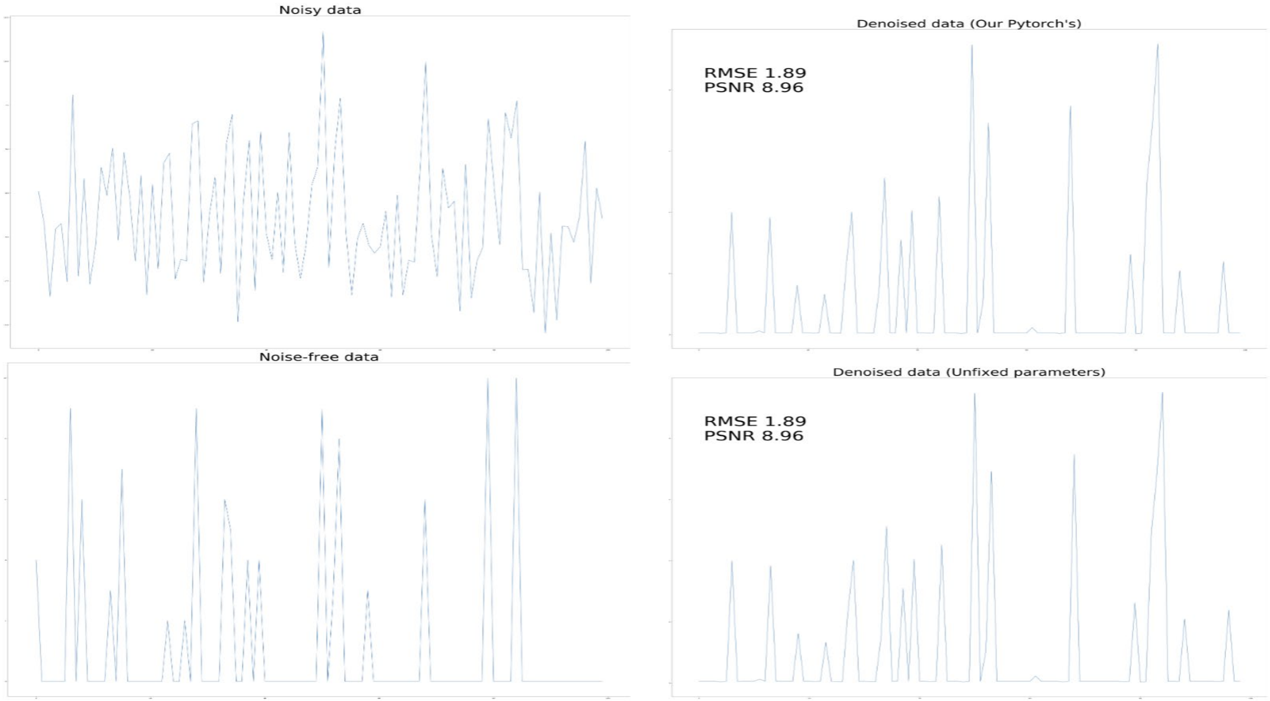
$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta,$$

- ❑ Max pooling / Average pooling layer
- ❑ Wiener filter and prior learning network

$$y = \begin{cases} \frac{\sigma^2}{\sigma_x^2} m_x + (1 - \frac{\sigma^2}{\sigma_x^2})x = \frac{(\sigma_x^2 - \sigma^2)x}{\sigma_x^2} + \frac{\sigma^2 m_x}{\sigma_x^2} \\ m_x \end{cases},$$

Third-party-library-free implementation

Compared with third-party-library-based implementation (Pytorch)



Pytorch- and Numpy-based are identical

Observations on influence of Batch normalization

Comparison between networks without vs with BN layer

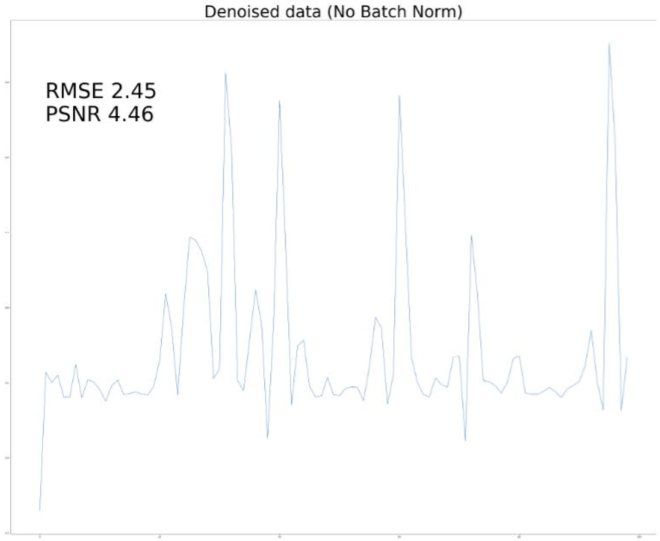


Figure 3: No batch normalization layer (network trained after 200 epochs).

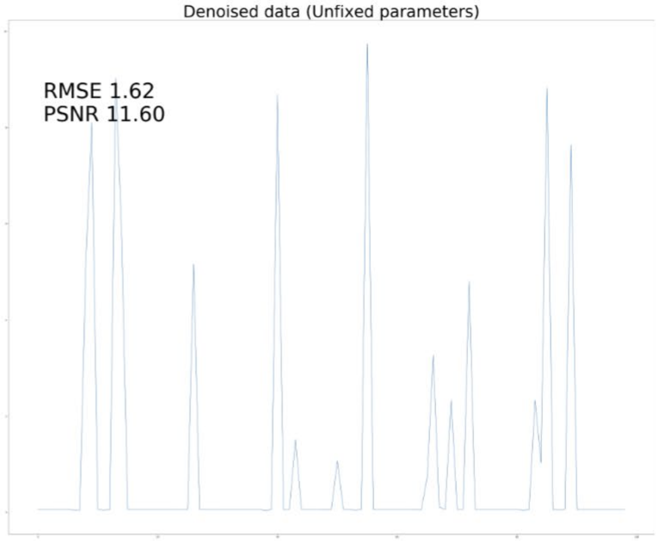


Figure 4: Batch normalization layer with unfixed means and SDs (network trained after 200 epochs).

Batch normalization shows great influence to the denoising results no matter which mode is chosen, even if there is only one sample in each mini-batch.

Observations on influence of Batch normalization

The purpose of batch normalization:

- ❑ Concentrate all latent features to a local mean by recentering and rescaling the feature representations (optimize probability distribution)
- ❑ Filtering the latent feature, control the outliers and reduce their aspects (signal processing view)
- ❑ Batch normalization layer has two modes: fixed-parameter mode and unfixed-parameter mode

- Fixed parameters mode

$$x = \gamma \frac{x - \mathrm{Mean}_{global}}{\sqrt{Var_{global} + \varepsilon}} + \beta$$

- Unfixed parameters mode

$$x = \gamma \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}} + \beta$$

E(x) and Var(x) belongs to current batch.

Observations on influence of Batch normalization

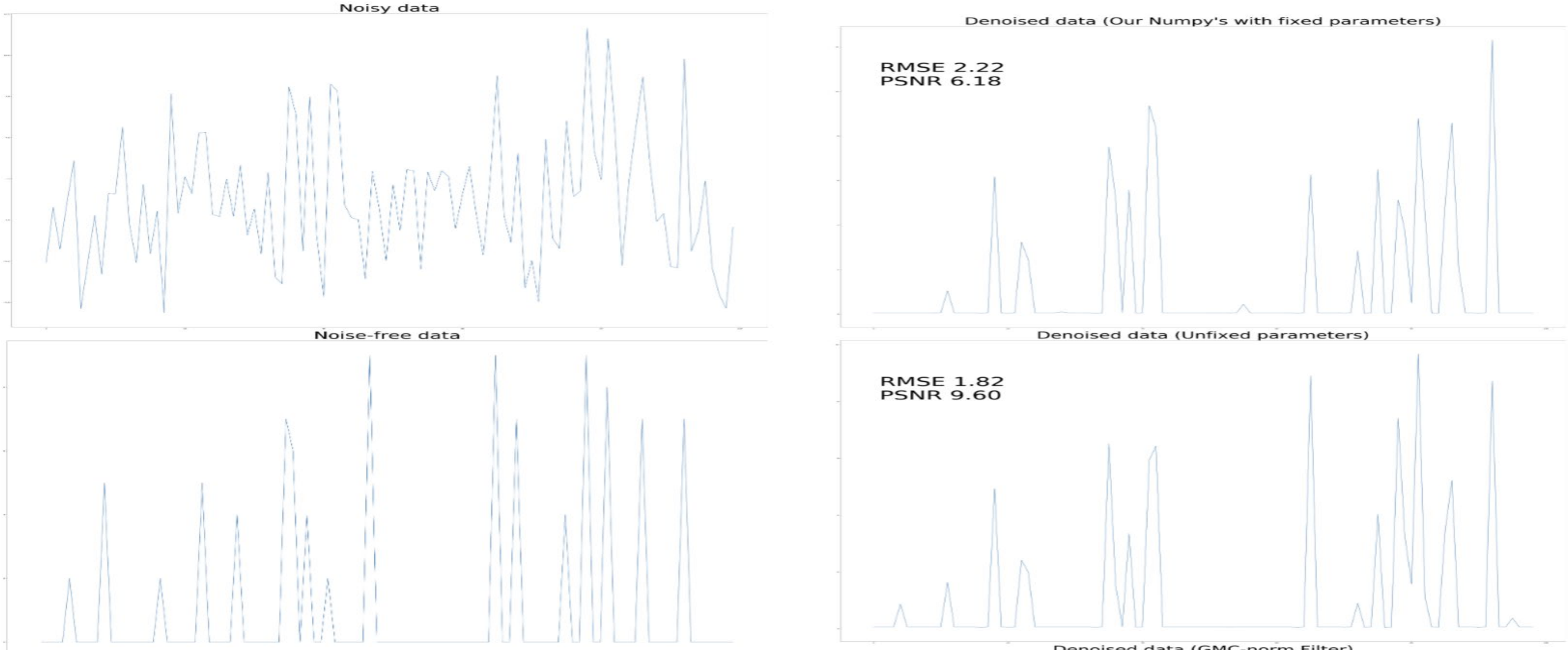
Comparison between BN layer with fixed vs unfixed mode

Experiment No.	fixed-parameter	unfixed-parameter	wiener filter	L1-norm filter	GMC-norm filter [15]
PSNR					
1	9.57	9.93	4.85	4.59	1.71
2	9.65	9.77	4.96	4.46	1.71
3	9.56	9.88	4.75	4.56	1.58
Average	9.59	9.86	4.85	4.54	1.66
RMSE					
1	1.84	1.80	2.40	2.44	2.88
2	1.83	1.82	2.39	2.45	2.88
3	1.84	1.80	2.41	2.44	2.90
Average	1.84	1.81	2.40	2.44	2.89

The unfixed mode has higher average accuracy than fixed mode and show superior performance in most cases.

Observations on influence of Batch normalization

Comparison between BN layer with fixed vs unfixed mode



Outline

1. Previous progress
   1) An interpretable neural networks with prior guidance
   2) Experimental results on signal denoising tasks
2. Current implementation and experiments
   1) DNN-based denoising algorithm overview
   2) Third-party-library-free implementation (mainly based on Numpy)
   3) Experiments about batch normalization
3. **Analysis and inspiration**
   1) **A signal-processing interpretation of DNN-based deep denoising algorithms**
   2) **Batch normalization layer with sparse regularization**
4. Future work

A signal-processing interpretation of DNN-based deep denoising algorithms

A standard convolutional layer

$$\begin{cases} x = x_i * k \\ x = \gamma \dfrac{x - Mean}{\sqrt{Var + \varepsilon}} + \beta \\ x_{i+1} = relu(x) = |x| \end{cases}$$

Signal-processing interpretation

$$x_{i+1} = \text{argmin}(x)\text{: Loss}(label_i - x * k) + \left|\gamma \frac{x - Mean}{\sqrt{Var + \varepsilon}} + \beta - x\right|$$

$$Training: K^*, \Gamma^*, B^* = argmin_{K,\Gamma,B}\{\mathscr{L}(Y - \Omega(X \odot K)) + [\Gamma|| \frac{X - E(X)}{\sqrt{Var(X)}} + B - X||\},$$

$$Testing: X^* = argmin_X\{\Omega(X \odot K) + [\Gamma|| \frac{X - E(X)}{\sqrt{Var(X)}} + B - X||]\},$$

Batch normalization layer with sparse regularization

Knowing that the batch normalization layer filter the input feature and sparse feature input helps the network converge fast and stable;

Features show big difference among layers;

Is it possible to sparsify latent features while concentrate their distribution?

$$Training : K^*, \Gamma^*, B^*, \Lambda^* = argmin_{K,\Gamma,B}\{\mathscr{L}(Y - \Omega(X \odot K)) + [\Gamma||\frac{X - E(X)}{\sqrt{Var(X)}} - X + B|| + \Lambda||X||^p]\},$$

$$Testing : X^* = argmin_X\{\Omega(X \odot K) + [\Gamma||\frac{X - E(X)}{\sqrt{Var(X)}} + B - X|| + \Lambda||X||^p]\},$$

Theoretically, adding a proper regularization as above term in the energy function can further constrain the sparsity of output features. In practice, to replace the batch normalization layer with an adaptive filter with sparse regularization can achieve this goal.

Outline

1. Previous progress
   1) An interpretable neural networks with prior guidance
   2) Experimental results on signal denoising tasks
2. Current implementation and experiments
   1) DNN-based denoising algorithm overview
   2) Third-party-library-free implementation (mainly based on Numpy)
   3) Experiments about batch normalization
3. Analysis and inspiration
   1) A signal-processing interpretation of DNN-based deep denoising algorithms
   2) Batch normalization layer with sparse regularization
4. **Future work**

Future work

- Explore a general form for unsupervised DNN-based denoising algorithms which approximate practical denoising applications and process signals without labels' information;

- Try different adaptive filters with various sparse regularizations to take the place of traditional batch normalization layers and check if the performance get improved.

NYU TANDON SCHOOL OF ENGINEERING

Reference

[1] Mart ́ın Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, SanjayGhemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In12th{USENIX}symposium on operating systems design and implementation ({OSDI}16), pages 265–283,2016.
[2] Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudo-random number generator.SIAMJournal on computing, 15(2):364–383, 1986.
[3] Shang-Hua Gao, Qi Han, Duo Li, Pai Peng, Ming-Ming Cheng, and Pai Peng. Representative batch normal-ization with feature calibration. InIEEE Conf. Comput. Vis. Pattern Recog, 2021.
[4] ZAHRA Habibi, HADI Zayyani, and MOHAMMAD Shams Esfand Abadi. A robust subband adaptive filteralgorithm for sparse and block-sparse systems identification.Journal of Systems Engineering and Electronics,32(2):487–497, 2021.
[5] Andr ́as Horv ́ath and Jalal Al-Afandi. Filtered batch normalization. In2020 25th International Conferenceon Pattern Recognition (ICPR), pages 6778–6785. IEEE, 2021.
[6] Hidenori Ide and Takio Kurita. Improvement of learning for cnn with relu activation by sparse regularization.In2017 International Joint Conference on Neural Networks (IJCNN), pages 2684–2691. IEEE, 2017.

Reference

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducinginternal covariate shift. InInternational conference on machine learning, pages 448–456. PMLR, 2015.
[8] Yann A LeCun, L´eon Bottou, Genevieve B Orr, and Klaus-Robert M¨uller. Efficient backprop. InNeuralnetworks: Tricks of the trade, pages 9–48. Springer, 2012.
[9] Jae S Lim. Two-dimensional signal and image processing.Englewood Cliffs, 1990.
[10] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts.arXiv preprintarXiv:1608.03983, 2016.
[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization.arXiv preprint arXiv:1711.05101,2017.
[12] Yijie Niu and Jiyou Fei. A sparsity-assisted fault diagnosis method based on nonconvex sparse regularization.IEEE Access, 9:59027–59037, 2021.
[13] Ivan Selesnick.Sparse regularization via convex analysis.IEEE Transactions on Signal Processing,65(17):4481–4494, 2017.
[14] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficientnumerical computation.Computing in science & engineering, 13(2):22–30, 2011.

Thank you!!