Advanced Project Report
Neural Networks with Prior Guidance for Signal Denoising and Speech Enhancement

Haoyu Fang

***Abstract*—Neural networks have achieved superior performance in various important tasks. However, lack of the interpretability makes the neural networks still a black box for most of the users. The black-box nature becomes the Achille's heel of neural network for full-exploitation. In past decades, studies on interpretable deep network become a popular research area. Many researches, especially on vision tasks, put forward effective network architectures to provide the model representatives with a comprehensive meaning. However, many signals/data are different from images which have clear semantic information, thus ambiguity within data makes networks performing on these data forms more difficult to understand. Triggered by these challenges, I focus on signal denoising tasks and propose an end-to-end-trained network guided by task-based knowledge that transforms input data into an integrated and ordered representative in high-dimensional feature space. Experiments on a synthesized sparse signal dataset and a human speech dataset verify that prior-guided structure can be deployed on deep network and the network forces the learned features to locate in a feature map in an order manner. Finally, a deep network with the proposed structure succeed in recognising different audio sounds and remove noise.**

I. Introduction

Neural networks have achieved superior performance in various important tasks (i.e. semantic segmentation, super-resolution image recovery and speech enhancement etc.). However, the black-box nature of neural networks brings crucial challenges for researchers to understand the learning process and makes it expensive to design and fine-tune a network even the tasks are very clear. The interpretability becomes an Achille's heel of neural network, and consequentially reduces the opportunity for its full-exploitation.

In recent year, studies on the interpretability of neural networks make many breakthroughs, especially on vision-based tasks. Representatively, [7] puts forward a probabilistic generative model for recognition tasks,in which a convolutional filter who detects object contours will activate its neighbor and can quickly determine object contours (shown in Figure 1(b)). In this network, an active filter gets more likely to activate its near neighbors and keeps its far neighbors salient. Another work [25] attempts to design a interpretable convolutional layer (shown in Figure 1(a)), filters in which are only sensitive to fix object categories. Zhang et. al realises this design by generating learnable masks that allow message passing for several filters but block others. However, these developments of interpretable neural network design are strongly related to the nature of images. The semantic information in a image is relatively clear and direct for pattern recognition. For instance, color and location of a pixel in the image is fixed and strongly associated with object-part pattern (e.g. object category, texture and structure etc.) so that this pixel can hardly be recognized as two patterns. This assumption brings new challenge to aforementioned techniques and limits their direct development on other forms of data.

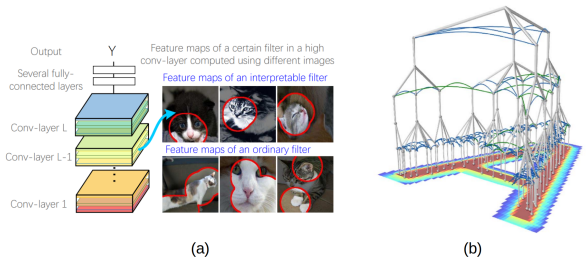

Fig. 1: (a) Interpretable convolutional filters that are sensitive to specific object categories[25]. (b) Recursive cortical network [7] that links filter's receptive field with a Conditional Random Field to realise location-aware contour detection.

Different from image, many signals have ambiguous semantic information such as 3D point cloud, 3D surface and 1D audio records. For example, one sample of an audio signal are likely to be a combination of signals with different patterns (e.g. music and noise), no matter in what domain we observe the signal. Though neural networks can also performs well on these data, the inherent ambiguity within data make the network more difficult to predict. Another factor that makes neural networks performing on audio data lack of interpretability is its highly-sparse and disorder feature representatives. Conventional model-driven denoising approaches [1, 2, 4] attempt to transform input signals into specific feature space (e.g. frequency, wavelet etc.), where signals with special/target pattern are separated from other signals and can be easily thresholded. However, deep networks [15, 12] also transform original signals into high-dimension feature space, but useful knowledge of one pattern is distributed in the whole feature maps and even overlap with other patterns' feature sometimes. This trait makes it difficult to threshold features of target pattern in a human-comprehensible manner. These challenges trigger me to explore a novel deep network structures that

- transform input data into an integrated and ordered representative in deep feature domain;

- consist of object-aware convolutional filters activated by some specific signal pattern (e.g. white noise) but remain inactive when processing signal with other patterns.

II. Related Works

A. Neural Networks

Thanks to accessibility to powerful computational resource and increasing sea of data, deep learning technology confronts its revolution in the past decades. The structures of deep neural networks have rapidly changed and enhanced themselves to predict more accurate and efficiently. Multi-Layer Perceptrons (**MLPs** [3]), one of useful architectures, was popular in the early stage of neural network research. Inspired by human feedforward and feedback nerve system, this architecture attempts to learn and memorized knowledge via forward and backward propagation progresses. The basic unit in this architecture is a component called neuron that consists of a linear and nonlinear operations. A group of neurons construct layers, which are recursively linked withed each other (shown in Figure 2). In a MLP network, each neuron is fully connected to all neurons in the next layer so that these layers are named Fully-connected layers. Between two sequential layers, the forward relationship for one neuron x_j^l in l-th layer and another neuron x_i^{l+1} in $l+1$-th layer is expressed as:

$$x_i^{l+1} = \sigma\{\Sigma_j(W_{ij}^{l+1}x_j^l + b_i^{l+1})\}, \tag{1}$$

where W^{l+1} and b^{l+1} are weights and biases, respectively and $\sigma(\cdot)$ refers to a nonlinear activation functions. In recent researches, Rectified Linear Units (**ReLU** $= max\{x, 0\}$ [14]) become a popular activation function widely used in deep neural networks. All weights (W's) and biases (b's) are trainable knowledge that can be learned from datasets via Back-propagation [11] (shown in following equation).

$$\begin{aligned} z_i^{l+1} &= \Sigma_j(W_{ij}^{l+1}x_j^l + b_i^{l+1}), \\ \frac{\partial c}{\partial W_{ij}^{l+1}} &= \frac{\partial c}{\partial z_i^{l+1}}\frac{\partial z_i^{l+1}}{\partial W_{ij}^{l+1}}, \\ \frac{\partial c}{\partial b_i^{l+1}} &= \frac{\partial c}{\partial z_i^{l+1}}\frac{\partial z_i^{l+1}}{\partial b_i^{l+1}}, \end{aligned}$$

where c denotes the cost and $\partial c/\partial z_i^{l+1}$ is called local gradient.

However, since fully-connected layers result in a rapid increase in parameters of a network, MLPs consume a great amount of computational resource and make themselves difficult to trained deeper. To overcome this limitation, [6] put forward a new neural network framework in which neurons are only connected to local neighbors and weights are shared across different spatial locations. To better learn features from a local neighbor, convolution takes the place of linear operation in previous MLPs' neurons by Equation 2:

$$x^{l+1} = \sigma(k^{l+1} \odot x^l + b^l), \tag{2}$$

where $\odot$ is convolution, x^l and x^{l+1} are all neurons in l-th and $l+1$-th layer respectively. k^{l+1} is the kernel of the

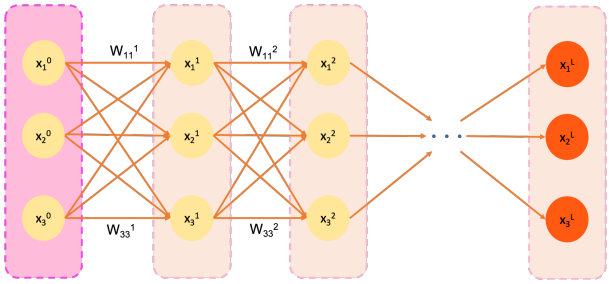

Fig. 2: Illustration of MLP architecture.

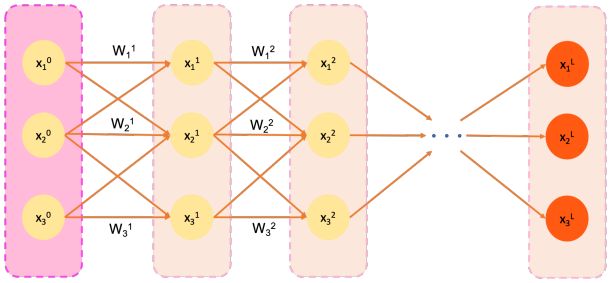

Fig. 3: Illustration of CNN architecture.

$l+1$-th layer. Because of the convolution within each layer, this kind of architecture is called Convolutional Neural Networks (**CNNs**).Due to the limited connection among neurons, CNNs significantly decrease the amount of parameters and make deep network easier to implement and outperform many conventional methods consequentially. Nowadays, CNNs are quickly developed and widely used in various applications including sequence processing and vision tasks. In this project, I use CNN-based frameworks as base models to develop the proposed Prior-guided Interpretable Neural Networks.

B. Interpretable Deep Networks

The black-box nature of end-to-end learning strategy makes CNN difficult for people to understand the logic of CNN predictions and make it more challenging for users to control its learning process. In recent years, an increasing number of researchers and scholars start to notice that high model interpretability demonstrates huge significance in development of deep networks in both theory and practice and make the networks training controllable. To further explore the hidden logic in neural network learning, a huge amount of efforts have been made in roughly six aspects [24]: 1) visualization of deep feature representation in latent network layers;2) diagnosis of networks' representations; 3) disentanglement of mixed feature representation encoded by a variety of filters in a deep network; 4) building explainable models; 5) semantic-level middle-to-end learning via human-computer interaction; 6) evaluation metrics for network interpretability.

Visualization of filters and feature representation of neural networks is a direct and fundamental way to investigate how latent layers learn information. [23, 13, 5] attempt to invert CNN filters and learned feature into images. Some researches [26, 27, 10, 22] apply passive virtualization and

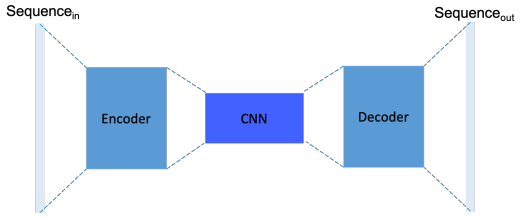

Fig. 4: Illustration of Encoder-decoder pipeline. This figure provides a general framework but the specific network design varies among different tasks. Detailed information are found in Figure 5.

retrieve target latent values from CNNs' outputs. Based on the virtual feedback from CNNs, some studies turn to apply statistical approaches to analyze CNN features. [17] proposed LIME method that compare gradient-based visualization and extracted image regions, in order to interpret network representations. More than observe or analyze CNN representations, some researchers develop specific structures [7, 25] to activate or inactivate filters within CNNs to learn more meaningful and comprehensible representations (i.e. spatial structures of objects in images, local details of point cloud and surface etc.). Nowadays. interpretability of neural network attracts more attention and become an essential key to comprehend and control the learning process in deep neural networks.

III. Method

A. Base Model Design

CNN-based Encoder-decoder architecture is capable of handling size/length-variable inputs and generating element-wise (i.e. pixel-wise, voxel-wise, etc) outputs. Network based denoising/enhancement is a task that requires element-wise classification. Therefore, I apply networks following this architecture as base model.

Figure 4 illustrates the basic structure of encode-decoder networks. The network consist of three components. The encoder network, fed with data that might have various lengths, learns a fix-shape latent feature via converting the information of original data into high-dimension representatives. Decoder network is the counterpart which deciphers deep feature by filtering unessential information and reforms semantic prediction in terms of given tasks. The last component is a convolution network that performs on the high-dimension representatives generated by the encoder. This convolution network attempts to deepen the feature map and make it more sparse in deep feature domain which assists the decoder to keep features of importance.

The detailed network design is illustrated in Figure 5. The shallow base model with fewest parameters are trained to recover noisy signals with single noise amplitude. The other two base models are trained to denoise signals with multi noise amplitudes and human speeches in noisy environments respectively. The shallow base model is trained and tested on SSS dataset and demonstrate its capability to remove white noise with single noise amplitude. Figure 6 illustrates the performance of the base model when handle noise with each noise amplitude.

Model Configuration				
Networks	Encoder	Middle CNN	Decoder	Fully Connection
Base Model	Conv(2,1,0) 1-3 Maxpool(2) Conv(2,1,0) 3-5 Maxpool(2)	Conv(2,1,0) 5-8 Conv(2,1,0) 8-12	Conv 12-5 Upsample(2) Conv 5-3 No ReLU Upsample(2)	FC 3-10 SoftMax
Base Model*	Conv(2,1,0) 1-3 Maxpool(2) Conv(2,1,0) 3-5 Maxpool(2)	Conv(2,1,0) 5-8 Conv(2,1,0) 8-12 Conv(2,1,0) 12-16	Conv(2,1,0) 16-12 Conv(2,1,0) 12-8 Conv(2,1,0) 8-5 No ReLU Upsample(2)	FC 5-10 SoftMax
Deep Base Model	Conv(2,1,0) 1-3 Conv(2,1,0) 3-64 Conv(1,1,0) 64-64 Maxpool(2) Conv(2,1,0) 64-128 Conv(1,1,0) 128-128 Maxpool(2) Conv(2,1,0) 128-256 Conv(2,1,0) 256-512 Maxpool(2)	Conv(1,1,0) 512-512 Conv(2,1,0) 512-1024 Conv(1,1,0) 1024-1024	Conv(2,1,0) 1024-512 Conv(2,1,0) 512-512 Upsample(2) Conv(2,1,0) 512-256 Conv(1,1,0) 256-256 Upsample(2) Conv(2,1,0) 256-128 Conv(1,1,0) 128-128 No ReLU Upsample(2)	FC 128-128 FC 128-256 SoftMax
Note	Conv(2,1,0) denotes a convolution layer with 2-size kernel, 1 step and 0 padding Conv 1-3 denotes the convolution layer turn a 1-channel feature into a 3-channel feature All convolution layers are followed by Batch-Normalization and ReLU. 'No ReLU' means this convolution layer are not followed by a ReLU unit.			
	Maxpool(2) denotes a maxpooling layer with 2-size kernel			
	Upsample(2) denotes a upsampling layer with 2-size kernel			
	FC 3-10 denotes the a 3-channel feature map is turned to be a 10-channel feature after the fully connection layer			

Fig. 5: Model Configuration, layer in the red box will be processed by the prior mask.

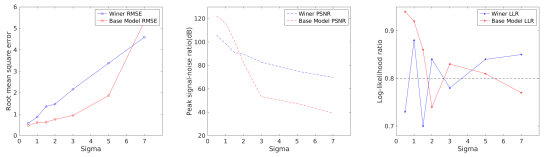

Fig. 6: Base model removes the white noise with various single noise amplitude.

B. Prior Guided Network

Task Prior

In a classic signal degradation system, the degraded signal can be describe: $s = s^* \odot g + n$, where s^* is the original input, s is the degraded signal, g is the system function, n is the noise and $\odot$ denotes convolution. To restore the original signal, many model-driven approaches [4, 19] provide good solution. While restoring signal, many of these approaches also predict the system function g, which is a effective prior knowledge to guide neural network.

Networks with Prior Mask

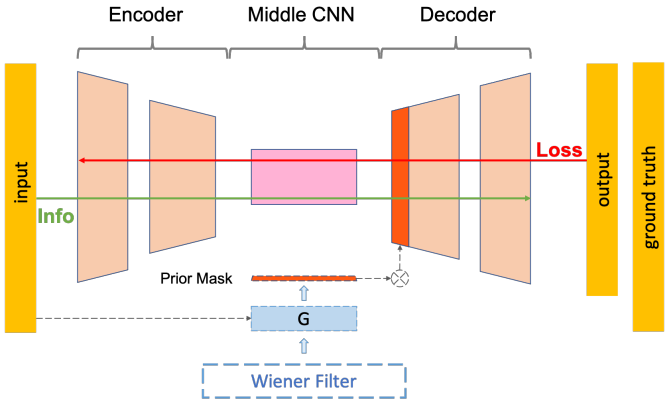

Fig. 7: The pipeline of proposed interpretable neural network with prior guidance.

Figure 7 demonstrates the module added on the base model. While feeding data into the network, a Wiener filter predicts the possible degradation system function G. Then a prior mask T, generated by uniformly up-sampling/down-sampling the inverse system function G^{-1}, was added to all convolutional filters in the first convolutional layer (layers in red area) of the decoder networks. In this way, the forward-propagation in these filter can be described as:

$$z = \delta\{x \odot [k(1 + \beta t_k)] + b\}, \tag{3}$$

where $\delta()$ is the activate function (RuLU in this paper), x is the feature output from middle CNN's final ReLU operation. b is the bias (in this layer, all filters' bias are set as zero for convenient observation.) β is a weight constant and t_k is a segment of T that added on filter k. While the back-propagation in the last layer of middle CNN is:

$$We^l = e^{l+1} \odot ROT[k^{l+1}(1 + \beta t_k)]\delta^{'}(z^l) \tag{4}$$

where l is current layer (final convolution layer of middle CNN), $ROT(\cdot)$ is a 1-D flipping function. Equation 4 demonstrates the prior mask t_k is passed backward to guide the learning processing in Encoder and middle CNN.

C. Loss Function Design

I deploy two kinds of loss to train the network. Huber loss [18] is a effective loss function widely used in denoising tasks. Huber function can be written as:

$$Loss_{huber} = \begin{cases} \frac{1}{2}x^2, & |y - \hat{y}| < 1 \\ |y - \hat{y}| - \frac{1}{2}, & |y - \hat{y}| \geq 1 \end{cases} \tag{5}$$

where $y - \hat{y}$ is the difference between network's prediction y and the ground truth $\hat{y}$. To constrain the influence of prior mask to the training process, an entropy-based loss (shown in Equation 6) is introduce into the network in back-propagation.

$$Loss_{entropy} = H(T|X) + \Sigma_x p(T, x)H(T|X = x), \tag{6}$$

where T denotes the prior mask and X is the feature map generated by the final convolution layer after the ReLU operation. The first and the second part in Equation 6 are a low inter-category entropy and low spatial entropy [7] and shown in following equations respectively.

$$H(T|X) = -\Sigma_x p(x)\Sigma p(T|x)log[p(T|x)], \tag{7}$$

$$\Sigma_x p(T, x)H(T|X = x) = \Sigma_x p(T, x)\Sigma_u \frac{p(T_u|x)}{p(T|x)} log(\frac{p(T_u|x)}{p(T|x)}). \tag{8}$$

In Equation 8, the term $H(T|X = x)$ encourage a low conditional entropy of spatial distribution of x's activates. In another words, the well-trained filter should only be activated by a single region u of the feature map x, instead of being repetitively triggered.

As a conclusion, the final loss in the network is defined as:

$$Loss = Loss_{huber} + \alpha Loss_{entropy}, \tag{9}$$

where α is a pre-defined constant.

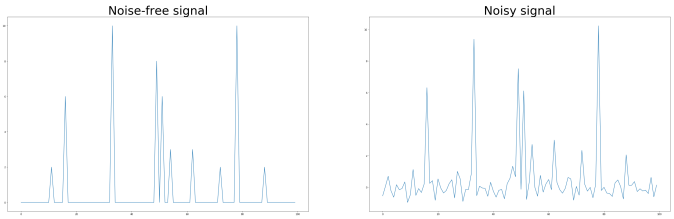

Fig. 8: Example of SSS dataset with $\sigma = 0.5$.

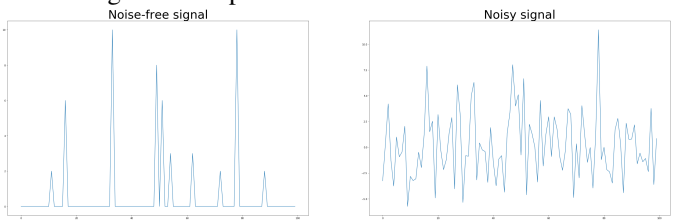

Fig. 9: Example of SSS dataset with $\sigma = 3.0$.

IV. Experiments and Results

A. Dataset

Synthesized-Sparse-Signal (SSS) Dataset

I synthesize a variety of spares signals and construct the **SSS Dataset**. I generate 7000 sequences with a length of 100 samples, 20% of which are randomly given non-zero values from 0 to 10. The 7000 sequences are noise-free signals used as ground truth. I add white noise to each sequence by following formula $signal_{noise} = signal + \sigma * noise$, where variance of the white noise is 1, constant $\sigma = k, k \in \{0.5, 1, 1.5, 2, 3, 5, 7\}$. For each σ value, I generate 1000 noisy data in order to make the dataset highly balanced. Figure 8 and Figure 9 illustrates signals with different noise amplitudes. I randomly split the dataset into 70% trainset and 30% testset.

Human Speech Dataset

The Human-Speech dataset (an example shown in Figure 10) is constructed by two sources: speech data is collected by

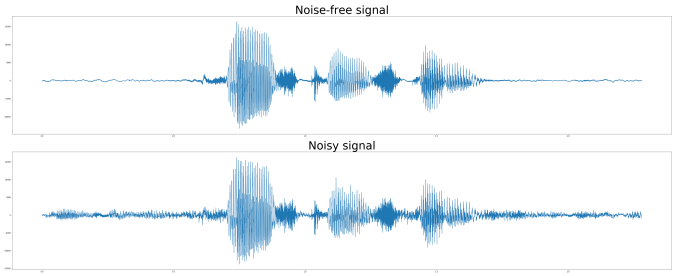

Fig. 10: Example of Human Speech Dataset (Time domain).

the Voice Bank corpus [21] and environmental sounds are provided by the Diverse Environments Multi-channel Acoustic Noise Database (DEMAND) [20]. Voice Bank Corpus collects 30 native English speakers from different parts of the world reading out around 400 sentences. The speech is obtained from two genders (male/female) of speakers. The background data is obtained from DEMAND. DEMAND[20] is a collection of 16-channel recordings sampled with 48kHz of acoustic noise recorded in 13 different environmental conditions [20].

To construct an artificial speech-with-noise dataset (shown in Figure 11), I dwonsample all records to 16kHz and mix the speech and noise at various signal-to-noise ratios (SNRs). For training dataset, noise and speech are added together with one of the 8 noise types at one of the following four SNRs: 0, 5, 10 and 15dB. I constructed the training dataset with 11,572 training samples from 28 speakers under 32 different noise conditions. Test samples are also synthetically mixed at one of the following four different SNRs: 2.5, 7.5, 12.5 and 17.5dB with one of the 5 noise types (types are different from training samples), resulting in 20 noise conditions for 2 speakers. Therefore, the testset features 824 samples from unseen speakers and noise conditions.

B. Evaluation Metrics

To give a fair evaluation of the proposed method, I introduce three metrics: Root mean square error (**RMSE**) and Peak signal-to-noise ratio (**PSNR**) and Log-likelihood ratio (**LLR**) that are commonly used in audio denoising and speech enhancement tasks. Since both datasets provide noise-free signals as ground truth, objective metrics like RMSE and PSNR demonstrate the difference between denoised/enhanced signals and ground truth. large PSNR but small RMSE refer to better model performance. While LLR metric reflects the correlation between input signal and output signal. Instead of supervising the model to provide accurate prediction, LLR metrics get the goal to reduce unnecessary distortion within denoising/deconvolution process. The purpose of LLR metrics are of importance in speech enhancement tasks which pursues a natural output. The mathematical implementation of the evaluation metrics are introduced in the rest of the section.

RMSE

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{j=1}^{n}(y_j - \hat{y}_j)^2}, \tag{10}$$

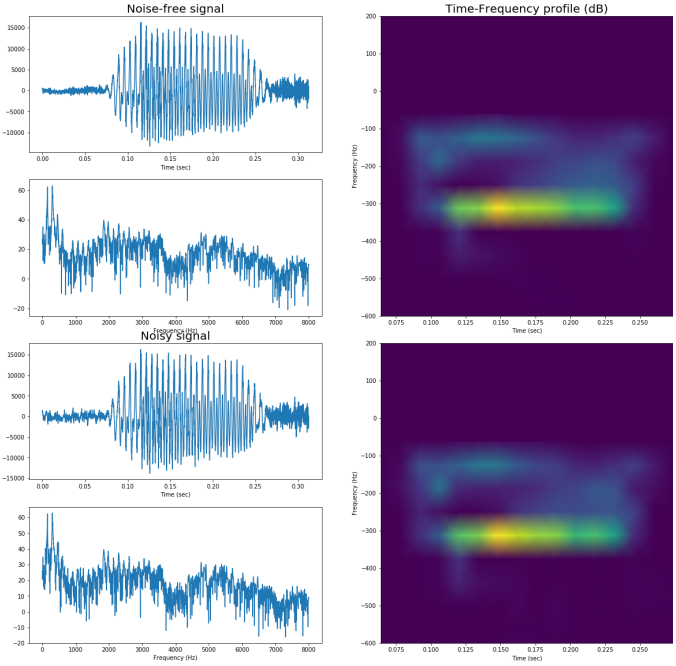

Fig. 11: Illustration of Human Speech Dataset.

where n is the length of the sequence, y is the predicted output and $\hat{y}$ is the ground truth.

PSNR

$$PSNR = \frac{1}{N} log_{10}\{\frac{MAX^2}{\Sigma_{j=1}^{n}(y_j - \hat{y}_j)^2}\}, \tag{11}$$

where MAX is the maximum possible sample value of the signal (MAX is 10 and $2^8 - 1$ in SSS Dataset and Human Speech dataset respectively). n is the length of the sequence, y is the predicted output and $\hat{y}$ is the ground truth.

LLR

$$d_{LLR}(\vec{a}_p, \vec{a}_i) = log\{\frac{\vec{a}_p \mathbf{R_i} \vec{a_p}^T}{\vec{a}_i \mathbf{R_i} \vec{a_i}^T}\} \tag{12}$$

where $\vec{a}_i$ is the Linear Predictive Coding (LPC) vector [9] of the original signal frame (an input signal), $\vec{a}_p$ is the LPC vector of the enhanced signal frame, $\mathbf{R_i}$ is the auto-correlation matrix of the input signal. Only the smallest 95% of the frame LLR values were used to compute the average LLR value [8]. The LLR values are supposed to be limited in the range of $(0, 1)$ to further reduce the number of outliers.

C. Basic Experimental Setup

In general, I implemented our models using the PyTorch [16] framework, which is an open-sourced deep learning platform that provides strong GPU support for computation efficiency. I use SGD optimizer with a momentum of 0.9 and an initial learning rate of 0.0001. I use an early stopping mechanism to monitor training, if the performance on test set

does not improve for 10 consecutive epochs then I decay the learning rate by half. My model takes 100 epoches and 500 epoches before convergence on SSS Dataset and DEMAND respectively using an NVIDIA RTX 1650 GPU. The hyper-parameters α and β are set as 0.5 and 0.3.

D. Results

In Figure 6, the shallow base model demonstrates its capability to remove white noise when the noise amplitude is homogeneous. However, I observe that the base model has difficulties when handle noise with multiple amplitudes. When the difference between two noise patterns are small, the network processes the noises as they are identical, the performance of the base model (shown in Figure 14 orange box) is slightly inferior to its own performance when there is noise with only one amplitude pattern. The orange boxes in Figure 15 and Figure 16 illustrate the base model fail to recognize the 'normal' samples and classify most of samples as noise when noises' pattern involves large difference such as $\sigma = 0.5$ and $\sigma = 3$. The unsuccessful denoising results in high RMSE and low PSNR, while LLR value approximates 0.8 (ratio non-zero sample in noise-free data). I also slightly modified the base model into base model* by adding a few layers on the original network, but the base model* remain low performance, thus I can conclude that this fatal failure is not associated with complexity of the network. Compared with base model, the model-driven approaches and the proposed prior guided network demonstrates a stable performance. Figure 17 and Figure 18 list several denosing and speech enhancement results, which demonstrate the prior-guided structure can be deployed to deep neural network and make the network classify various audio signals.

Another observation is the proposed network pushes learned features to gradually locate in order. During the training process, I repeatably recorded kernels' weights in the first convolutional layer of the decoder network, which is masked by the prior knowledge. The weights are randomly initialized. After several epochs, though kernels in both models start to be sparse, but model with prior-guidance gradually concentrate its active kernels. The other evidence is more direct. After both models get well-trained, I fed signals with different noise amplitude to both models and observed the output features of masked layer, only the proposed model provides a ordered feature map which helps me to understand the learning process.

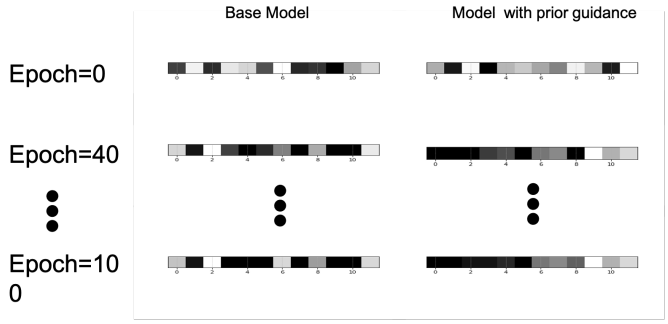

Fig. 12: During the training progress, the weights of kernels on masked layer gradually become concentrated while weights of kernels in base model still randomly vary.

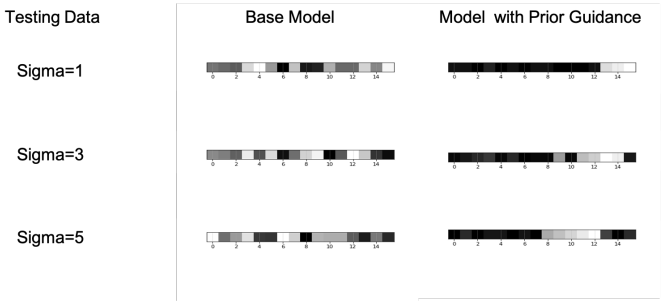

Fig. 13: After models are well trained, feed models with signals with different noise pattern and observe the output of the first convolutional layer in decoder, I find feature of the proposed model has clear order but feature of base model does not.

V. Conclusion

This project focus on improving the interpretability of neural network. Applying model-driven approaches to guide the network can give a training clues for network to shape its feature space. During the project, I concentrate on the signal denosing tasks and propose a end-to-end-trained network guided by Wiener Filter's knowledge and push the network to transforms input data into ordered representatives in high-dimensional feature space. Experiments on a synthesized sparse signal dataset and a human speech dataset verify that the network forces the learned features locate in a feature map in an order manner and the network can recognise different audio sounds and remove noise.

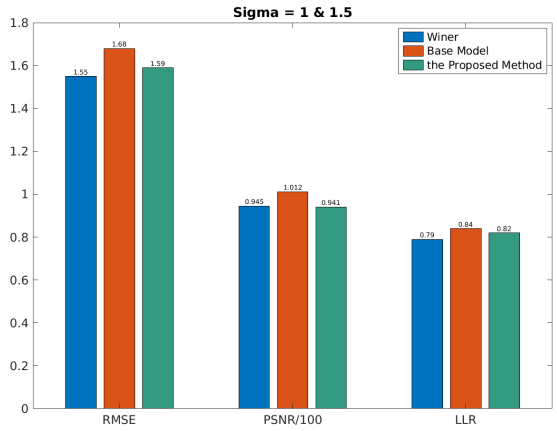

Fig. 14: Denosing results when models are trained and tested on noise data whose σ equals to 1 and 1.5.

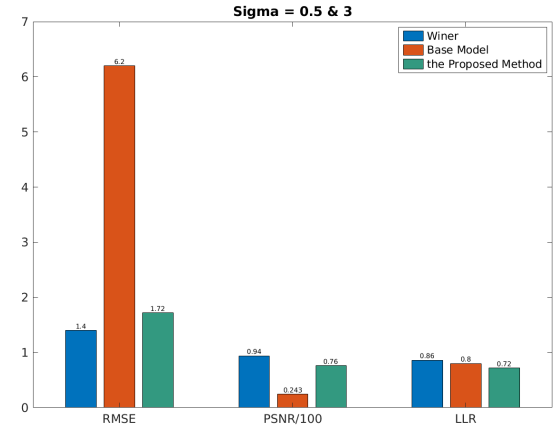

Fig. 15: Denosing results when models are trained and tested on noise data whose σ equals to 0.5 and 3.

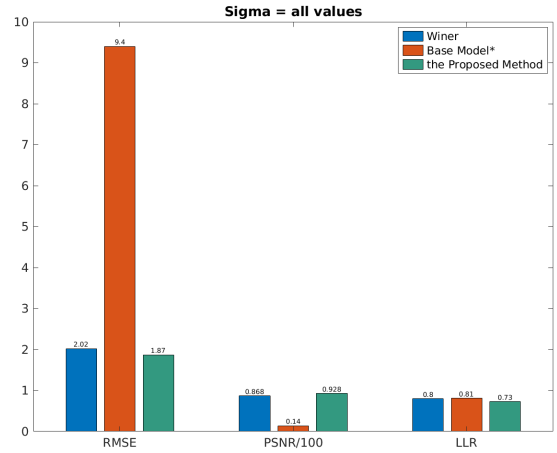

Fig. 16: Denosing results when models are trained and tested on the whole dataset.

References

[1] Mikhled Alfaouri and Khaled Daqrouq. Ecg signal denoising by wavelet transform thresholding. *American Journal of applied sciences*, 5(3):276–281, 2008.

[2] Manuel Blanco-Velasco, Binwei Weng, and Kenneth E Barner. Ecg signal denoising and baseline wander correction based on the empirical mode decomposition. *Computers in biology and medicine*, 38(1):1–13, 2008.

[3] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.

[4] Jingdong Chen, Jacob Benesty, Yiteng Huang, and Simon Doclo. New insights into the noise reduction wiener filter. *IEEE Transactions on audio, speech, and language processing*, 14(4):1218–1234, 2006.

[5] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016.

[6] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[7] Dileep George, Wolfgang Lehrach, Ken Kansky, Miguel Lázaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou, Zhaoshi Meng, Yi Liu, Huayan Wang, et al. A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science*, 358(6368), 2017.

[8] John HL Hansen and Bryan L Pellom. An effective quality evaluation protocol for speech enhancement algorithms. In *Fifth international conference on spoken language processing*, 1998.

[9] Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2007.

[10] Soheil Kolouri, Charles E Martin, and Heiko Hoffmann. Explaining distributed neural activations via unsupervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.

[11] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[12] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018.

[13] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[14] Vinod Nair and Geoffrey E Hinton. Rectified linear units

Fig. 17: Some examples of denoising results on SSS Dataset.

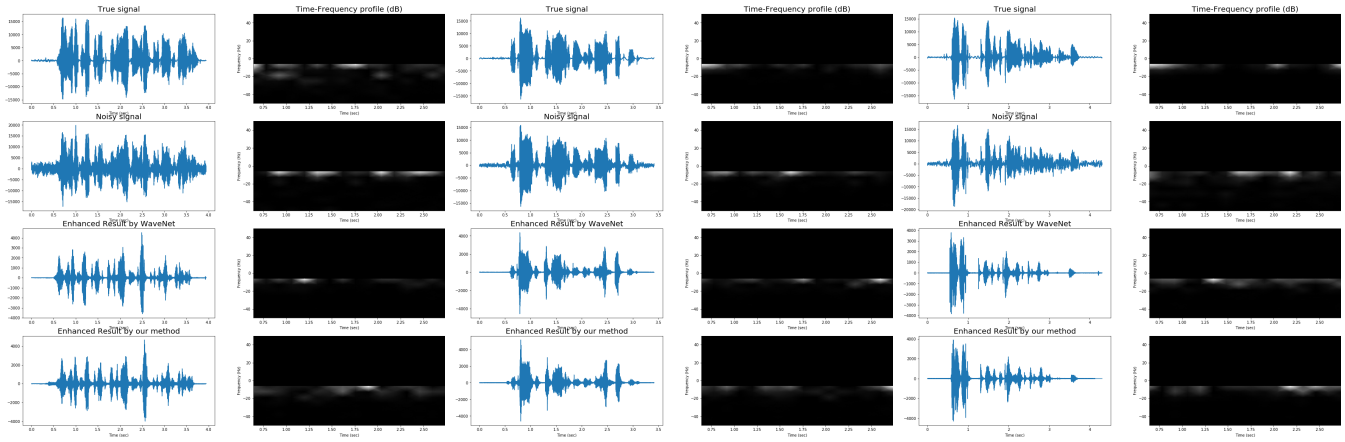

Fig. 18: Human speech enhancement results.

improve restricted boltzmann machines. In *ICML*, 2010.

[15] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[18] Sylvain Sardy, Paul Tseng, and Andrew Bruce. Robust wavelet denoising. *IEEE Transactions on Signal Processing*, 49(6):1146–1152, 2001.

[19] Ivan Selesnick, Alessandro Lanza, Serena Morigi, and Fiorella Sgallari. Non-convex total variation regularization for convex denoising of signals. *Journal of Mathematical Imaging and Vision*, pages 1–17, 2020.

[20] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings. *The Journal of the Acoustical Society of America*, 133(5):3591–3591, 2013.

[21] Christophe Veaux, Junichi Yamagishi, and Simon King. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–4. IEEE, 2013.

[22] Achmadnoer Sukma Wicaksana and Cynthia CS Liem. Human-explainable features for job candidate screening prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1664–1669. IEEE, 2017.

[23] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[24] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[25] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.

[26] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.