Deep Learning for Millimeter Wave Channel Estimation
Main Presentation

Chirag Ramesh

SPC Lab, Indian Institute of Science

September 08, 2018

Contents

Millimeter Wave Channel Model

- The model for a mmWave channel can be written as[1]:

$$\mathbf{H} = \sum_{l=1}^{N_p} \alpha_l \mathbf{a}_R(\theta_{R,l}, \phi_{R,l}) \mathbf{a}_T^H(\theta_{T,l}, \phi_{T,l}) = \overline{\mathbf{A}}_R \overline{\mathbf{H}}_b \overline{\mathbf{A}}_T^H$$

- N_p is the total number of paths
- α_l is the complex gain associated with the l-th path, $l \in \{1, 2, \dots N_p\}$
- $\mathbf{a}_R(\theta_R, \phi_R)$ and $\mathbf{a}_T(\theta_T, \phi_T)$ are the 3D receive and transmit array steering vectors respectively
- $\overline{\mathbf{H}}_b = diag(\alpha) \in \mathbb{C}^{N_p \times N_p}$ is the beamspace/virtual channel matrix
- $\overline{\mathbf{A}}_r \in \mathbb{C}^{N_r \times N_p}$ and $\overline{\mathbf{A}}_t \in \mathbb{C}^{N_t \times N_p}$ contain the array response vectors for the receiver and transmitter respectively.

[1] [1]: "An overview of signal processing techniques for millimeter wave MIMO systems," IEEE Journal of Selected Topics in Signal Processing, April 2016

- Assume that the AoAs and the AoDs come from a grid of size G ($\theta, \phi \in \{0, \frac{2\pi}{G}, \cdots, \frac{2\pi(G-1)}{G}\}$) with $G \gg N_p$.

- $\mathbf{A}_r \in \mathbb{C}^{N_r \times G}$ and $\mathbf{A}_t \in \mathbb{C}^{N_t \times G}$ contain the array response vectors corresponding to the angles in the grid for the receiver and transmitter respectively.

- $\mathbf{H}_b \in \mathbb{C}^{G \times G}$ is an N_p sparse matrix with N_p non-zero elements corresponding to the angles in the grid.

$$\mathbf{H} = \overline{\mathbf{A}}_R \overline{\mathbf{H}}_b \overline{\mathbf{A}}_T^H = \mathbf{A}_R \mathbf{H}_b \mathbf{A}_T^H$$

- For a large G, grid error is negligible.

- The received signal using Hybrid Analog/Digital Signal Processing can be written as follows:
$$\mathbf{Y} = \mathbf{W}^H\mathbf{HFX} + \mathbf{N}$$
- $\mathbf{F} = \mathbf{F}_{RF}\mathbf{F}_{BB} \in \mathbb{C}^{N_t \times N_s}$ is the transmit precoding matrix
- $\mathbf{W} = \mathbf{W}_{RF}\mathbf{W}_{BB} \in \mathbb{C}^{N_r \times N_s}$ is the received combining matrix
- Let the pilot matrix be $\mathbf{X} = \sqrt{P}\mathbf{I} \in \mathbb{C}^{N_s \times N_s}$
- The received signal can be written as $\mathbf{Y} = \sqrt{P}\mathbf{W}^H\mathbf{HF} + \mathbf{N}$
- After performing vectorization, the received signal is
$$\mathbf{y} = \sqrt{P}(\mathbf{F}^T \otimes \mathbf{W}^H)vec(\mathbf{H}) + \mathbf{n}$$

- Since $vec(\mathbf{H}) = ((\mathbf{A}_T^H)^T \otimes \mathbf{A}_R) vec(\mathbf{H}_b)$, the signal can be further simplified to

$$\mathbf{y} = \sqrt{P}(\mathbf{F}^T \otimes \mathbf{W}^H)((\mathbf{A}_T^H)^T \otimes \mathbf{A}_R) vec(\mathbf{H}_b) + \mathbf{n}$$

$$\Rightarrow \mathbf{y} = \sqrt{P}(\mathbf{F}^T (\mathbf{A}_T^H)^T \otimes \mathbf{W}^H \mathbf{A}_R) vec(\mathbf{H}_b) + \mathbf{n}$$

$$\Rightarrow \mathbf{y} = \mathbf{Q}\mathbf{h}_b + \mathbf{n}$$

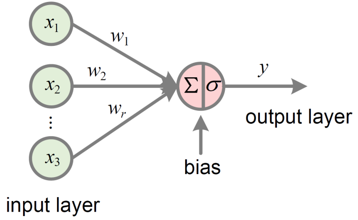
- Estimation of $\mathbf{h}_b$ is formulated as a sparse signal recovery problem:

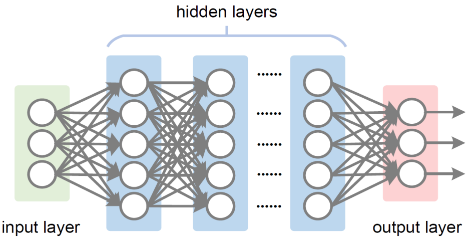
$$\min_{\mathbf{h}_b} \|\mathbf{h}_b\|_0$$

$$s.t. \quad \|\mathbf{y} - \mathbf{Q}\mathbf{h}_b\|_2 \leq \sigma$$

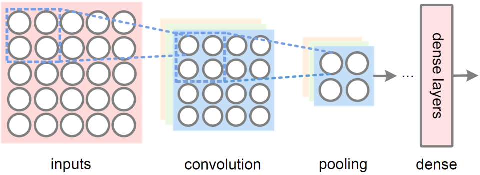
- Current literature uses Least Absolute Shrinkage and Selection Operator (LASSO), standard greedy recovery algorithms such as Orthogonal Matching Pursuit (OMP) and adaptive compressive sensing techniques.

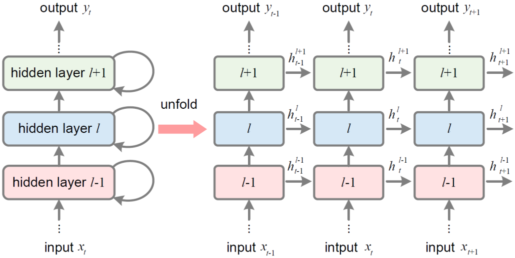
- A neuron is the basic unit of a neural network.
- A Neural Network (NN) is made by interconnecting these neurons in a layered architecture.
- The simplest NN, a Perceptron, has a step function as its activation function.
- For training this NN, a loss function is minimized.





- A fully connected feedforward NN has neurons connected to all the neurons in the next layer (unidirectional).
- A Convolutional Neural Network (CNN) has local and shared connections.
- Recurrent Neural Networks (RNN) are NNs with bidirectional connections.
- Deep Neural Networks (DNN) are NNs with a very large number of fully connected layers.





Why use Deep Learning?

Advantages

- Convenient optimization capability
- High processing capability
- Good computation speed
- Custom loss function
- Complex models can be easily learnt
- Uses history of observations

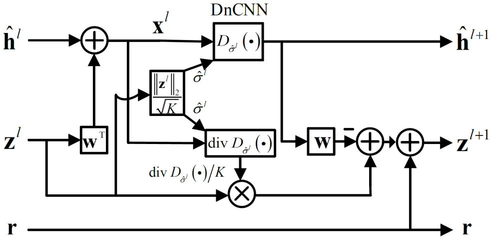
Limitations

- Converge to the local optimum of the non convex loss functions
- DNNs are subject to overfitting
- Hyperparameter tuning is empirical
- DNNs are data hungry
- No performance guarantees
- Difficult to interpret what is learnt

Channel estimation using LDAMP Neural Network

What is LDAMP?

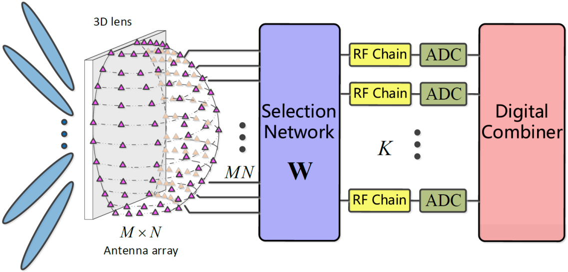
- LDAMP stands for Learned Denoising-based Approximate Message Passing[2].
- Combination of iterative sparse signal recovery with a Denoising Convolutional Neural Network (DnCNN).
- More obscure features learnt.
- Comparatively, it is faster and more accurate than other competing techniques.



[2][2]: "Deep Learning-based Channel Estimation for Beamspace mmWave Massive MIMO Systems," IEEE Wireless Communications Letters, May 2018

System Model

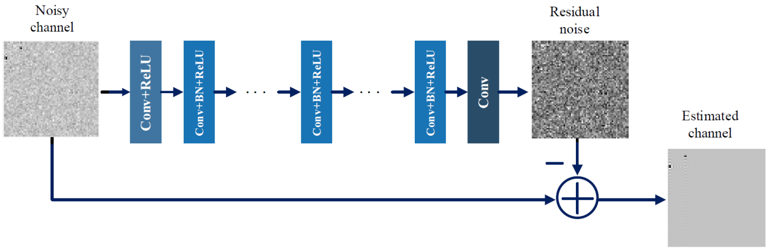
- Lens antenna array architecture used
- MxN antennas from the 3D lens are connected to K RF chains via the selection network W
- Here, K $\ll$ MN
- Received signal $\mathbf{y} \in \mathbb{R}^{MN \times 1}$
- Channel $\mathbf{h} \in \mathbb{R}^{MN \times 1}$
- Network $\mathbf{W} \in \mathbb{R}^{K \times MN}$



- The uplink received signal can be written as: $\mathbf{y} = \mathbf{h}s + \mathbf{n}$
- The received signal after the RF chains can be represented as: $\mathbf{r} = \mathbf{W}\mathbf{y} = (\mathbf{W}s)\mathbf{h} + \overline{\mathbf{n}}$

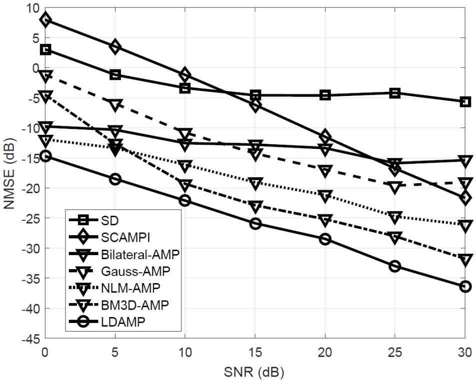
DnCNN structure

- Within the DnCNN, twenty convolutional layers are used.
- The first layer has 64 ($3 \times 3 \times 1$) filters with a Rectified Linear Unit (ReLU).
- Each of the next 18 layers have 64 ($3 \times 3 \times 64$) filters with batch normalization and then passed through a ReLU.
- The final convolutional layer uses one ($3 \times 3 \times 64$) filter to reconstruct the signal.
- The residual noise is learnt from the noisy channel and then subtracted from it.



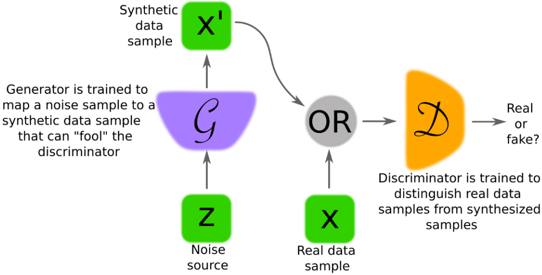
Results

- One user with a 4 path mmWave channel
- Implementation on MatCovNet
- L = 10 layers
- M = N = 64
- Training, Validation and Test sets contain 16640, 6400 and 10000 samples.
- Stochastic Gradient Descent used along with Adaptive Moment Estimation (ADAM) optimizer



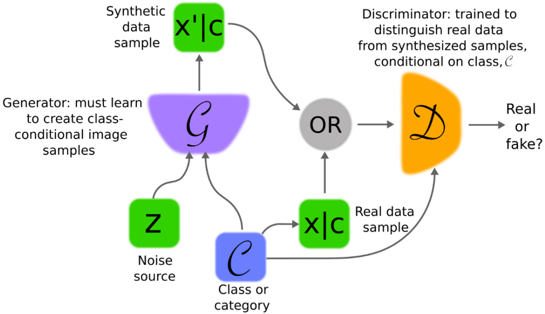
What is a GAN?

- Generative Adversarial Network is a game theoretic approach for synthetic data generation
- The generator $\mathcal{G} : \mathbb{R}^Z \to \mathbb{R}^M$ creates synthetic data
- The discriminator $\mathcal{D} : \mathbb{R}^M \to \{0, 1\}$ assesses the quality of the generated data
- Loss functions $J_{\mathcal{D}}(\theta_{\mathcal{G}}, \theta_{\mathcal{D}})$ and $J_{\mathcal{G}}(\theta_{\mathcal{G}}, \theta_{\mathcal{D}})$ are defined.



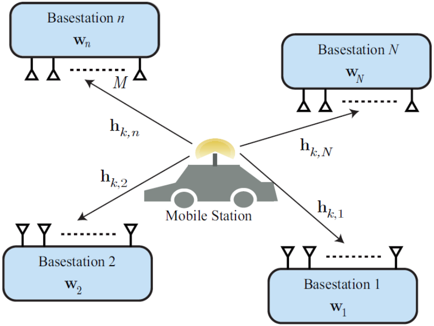
Why use a GAN?

- Highly effective for feature extraction & segmentation in image processing
- Conditional GANs represent multi-modal densities better than GANs
- Similar to maximizing the mutual information between the "latent code" and the output of the generator
- GANs implicitly learn the data generating distribution



System Model for UL Channel Estimation[3]

- 'N' BSs are connected to each other to share UL training signals.
- Each BS has 'M' antennas whereas the users have only one antenna each.
- Each BS has only one RF chain and applies analog-only combining.
- SIMO Wideband OFDM system with K subcarriers is considered.



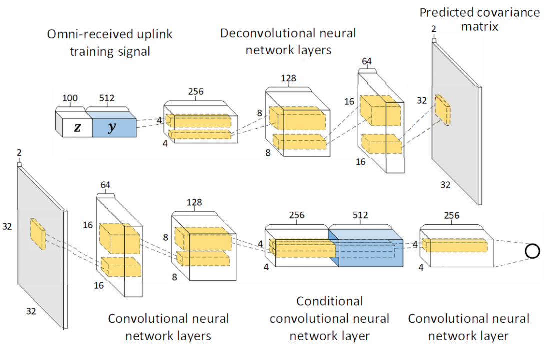
[3] [3]: "Generative Adversarial Estimation of Channel Covariance in Vehicular Millimeter Wave Systems," arXiv:1808.02208v1 [cs.IT]

- The received signal post combining on the k-th subcarrier for the n-th BS is written as
$$y_{k,n} = \mathbf{w}_n^T \mathbf{h}_{k,n} s_k + v_k$$
- $\mathbf{h}_{k,n} \in \mathbb{C}^{M\times 1}$ is the uplink channel vector between the user and the n-th BS on the k-th subcarrier
- $s_k \in \mathbb{C}^1$ is the pilot symbol on the k-th subcarrier
- $\mathbf{w}_n \in \mathbb{C}^{M\times 1}$ is the analog combiner at the n-th BS
- All the received signals at the BSs are collected together
$$\mathbf{y} = [y_{1,1}, \cdots, y_{K,1}, y_{1,2}, \cdots, y_{K,2}, \cdots, y_{K,N}]^T$$
- The channel covariance matrix is to be jointly estimated

Architecture

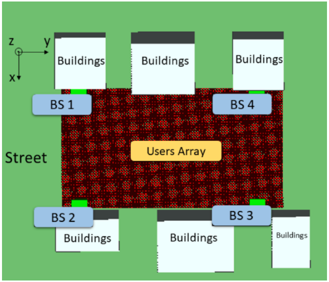
- $\mathcal{G} : \mathbb{R}^{Z} \, x \, \mathbb{R}^{NK} \rightarrow \mathbb{R}^{M^2}$
- $\mathcal{D} : \mathbb{R}^{M^2} \, x \, \mathbb{R}^{NK} \rightarrow \{0, 1\}$
- $\hat{\mathbf{R}} = \mathcal{G}(\mathbf{z}, \mathbf{y})$
- 200 epochs training

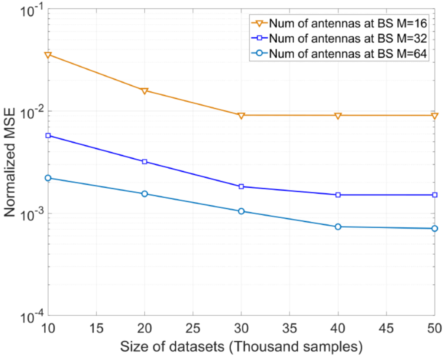
- $\mathcal{D}$ consists of stride-2 CNN with ReLU function.
- ADAM optimizer used with batch size of 256.
- Loss Function,
  $L(\mathcal{G}, \mathcal{D}) = \mathbb{E}[log(\mathcal{D}(\mathbf{R}, \mathbf{y}))] + \mathbb{E}[log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}, \mathbf{y})))]$



Results

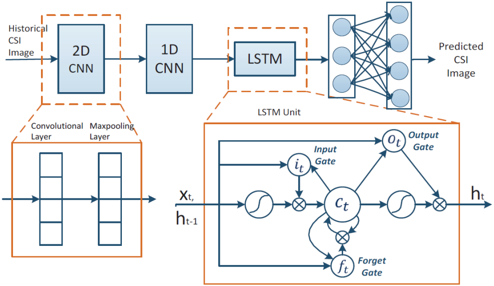
- Simulated using ray-tracer Wireless InSite and TensorFlow
- K = 64 subcarriers
- N = 4 BSs
- M = 32 Rx. Antennas on a ULA
- Upto 5 MPCs for each channel





What is LSTM?

- Long Short-Term Memory (LSTM) Neural Network is a NN with LSTM units.
- The LSTMs are trained to "forget" data after some amount of time.
- LSTMs handle time series data excellently.
- They are insensitive to time gap between images.
- Vanishing gradient problem is alleviated in LSTMs.



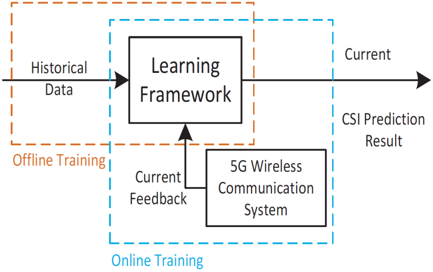
System Model [4]

- Flat block-fading MIMO system is considered.
- Received signal is $\mathbf{y}_i = \mathbf{H}\mathbf{p}_i + \mathbf{n}_i$
- $\mathbf{H} \in \mathbb{C}^{r \times t}$ is the channel matrix
- $\mathbf{p}_i \in \mathbb{C}^{t \times 1}$ is the i-th pilot, $i \in \{1, 2, \cdots, N\}$
- Channel data + Side information = Input of the network
- The side information features used are:
  - Frequency bands and the associated absorption values
  - Location measurement data
  - Weather, temperature and humidity

[4] [4]: "Channel State Information Prediction for 5G Wireless Communications: A Deep Learning Approach," in IEEE Transactions on Network Science and Engineering, June 2018

Procedure

- Input: CSI data + side information for each of the N channels
- LSTM works very well for sequential task learning, it is used to predict CSI.
- Output: Predicted CSI image
- l_2 loss function is used for training.
- A two step offline-online training is performed.

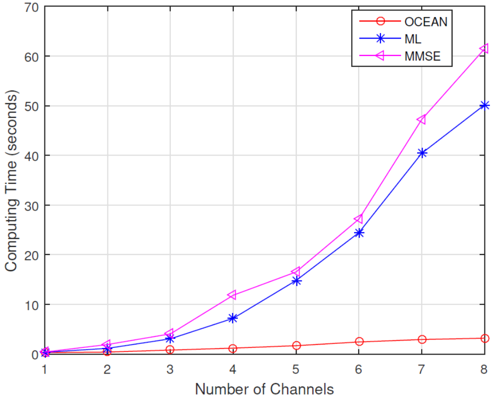


Data Collection

Data is collected for the following scenarios:

- Case 1: Outdoor free space scenario. Only LOS transmission.
- Case 2: Outdoor environment with few multi path components.
- Case 3: Indoor closed environment with a workroom.
- Case 4: Indoor closed environment in a corridor.

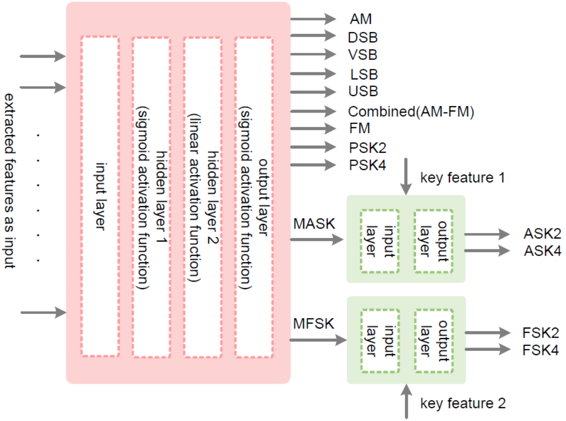
Online training happens once every five minutes.



Average ADR(%)	Case 1	Case 2	Case 3	Case 4
OCEAN	2.73	2.65	3.46	3.22
ANN	6.83	7.26	6.84	5.66

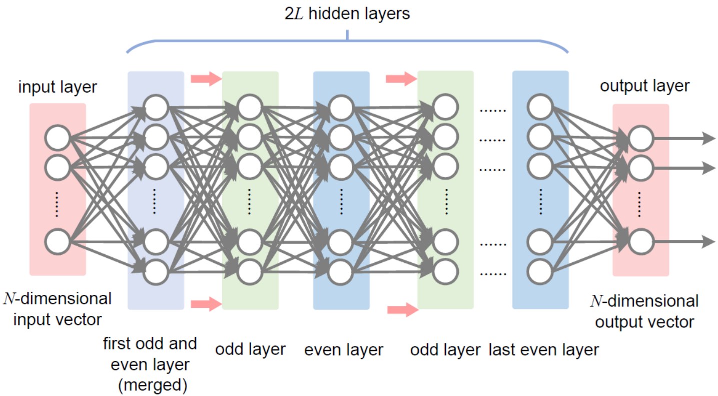
Existing Architectures for Wireless Systems with Deep Learning[5]

- Modulation scheme recognition is necessary.
- Previous techniques have used SVMs and ANNs for modulation recognition.
- A four layer NN shown can easily distinguish between modulation schemes.

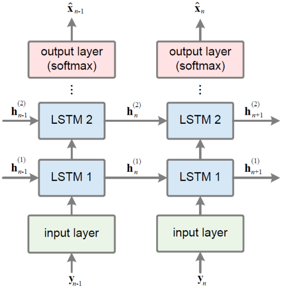


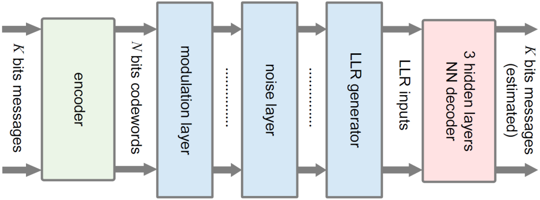
[5] [5]: "Deep Learning for Wireless Physical Layer: Opportunities and Challenges," arXiv:1710.05312 [cs.IT]

- Belief Propagation algorithm with L iterations can be unfolded into 2L layered fully connected DNN.
- The input is the N-dimensional LLR. The output is an N-bit decoded codeword.

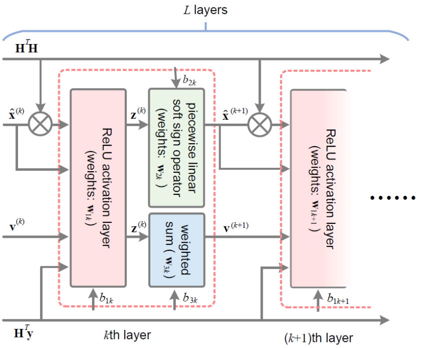


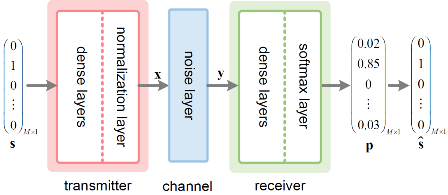
- LSTMs are used for data detection for a wideband channel.
- It performs similar to Maximum Likelihood Sequence Detection.
- An NN Decoder is used to decode codewords of length N with K information bits.
- It is shown to have worked for polar codes.



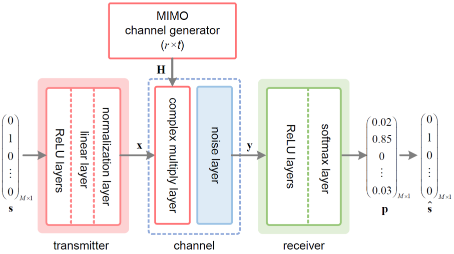


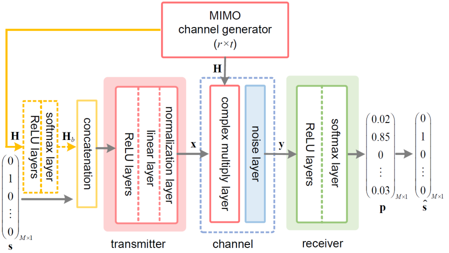
- DetNet architecture is used for data detection for a narrowband channel.
- Communication is considered as an end to end reconstruction problem and modelled as an Autoencoder.
- This autoencoder by default is trained to provide end to end performance, such as BER.





- Open loop, feedback and quantized feedback architectures for MIMO communications are shown.
- For a MU case, $J = \sum_{i=1}^{N} \alpha_i J_i$.
- J_i is the loss function for the i-th user.
- $\sum_i \alpha_i = 1$
- Each J_i is a cross entropy loss function.





Challenges

- For data detection, current architectures perform a straight forward unfolding of the iterations into different layers of the DNN. Knowledge of existing communication algorithms can enable creating novel specific architectures.
- Optimal input, output representations for deep learning systems are unknown.
- Performance analysis for such novel architectures can be done based on existing wireless literature.
- The exact functions learnt by the weights aren't clear yet. Understanding how these functions are built can help us understand how to build algorithms either with or without using DNNs.
- A combination of DNNs and existing schemes can be used so that each of them give their best features.

References I

1. R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 3, pp. 436–453, April 2016
2. H. He, C. Wen, S. Jin and G. Y. Li, "Deep Learning-based Channel Estimation for Beamspace mmWave Massive MIMO Systems," in IEEE Wireless Communications Letters, May 2018
3. "Generative Adversarial Estimation of Channel Covariance in Vehicular Millimeter Wave Systems," arXiv:1808.02208v1 [cs.IT]
4. C. Luo, J. Ji, Q. Wang, X. Chen and P. Li, "Channel State Information Prediction for 5G Wireless Communications: A Deep Learning Approach," in IEEE Transactions on Network Science and Engineering, June 2018
5. "Deep Learning for Wireless Physical Layer: Opportunities and Challenges," arXiv:1710.05312 [cs.IT]
6. "Generative Adversarial Networks: An Overview," arXiv:1710.07035 [cs.CV]

References II

7. A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu and D. Tujkovic, “Deep Learning Coordinated Beamforming for Highly-Mobile Millimeter Wave Systems,” in IEEE Access, vol. 6, pp. 37328-37348, 2018
8. Q. Xu, C. Zhang and L. Zhang, "Denoising Convolutional Neural Network," 2015 IEEE International Conference on Information and Automation, Lijiang, 2015, pp. 1184-1187
9. T. J. O’Shea and J. Hoydis. (2017) An introduction to deep learning for the physical layer. [Online]. Available: https://arxiv.org/abs/1702.00832,preprint
10. T. J. O’Shea, T. Erpek, and T. C. Clancy. (2017) Deep learning based MIMO communications. [Online]. Available: https://arxiv.org/abs/1707.07980,preprint
11. N. Farsad and A. Goldsmith. (2017) Detection algorithms for communication systems using deep learning. [Online]. Available: https://arxiv.org/abs/1705.08044,preprint
12. S. Park and R. W. Heath Jr, “Spatial channel covariance estimation for the hybrid MIMO architecture: A compressive sensing based approach,” arXiv preprint arXiv:1711.04207, 2017

Thank You!