EE5111 Final Project:

Massive MIMO Channel Estimation with an Untrained Deep Neural Network

Eren Balevi, Akash Doshi, Jeffrey G. Andrews

Final Report by

Nithin Varma K, Hemanth Ram G K , Prabash Reddy M, Tanzir Silar

June 6, 2021

Introduction & Motivation

- Wireless Channel estimation for a communication link has always been quite challenging, due to its probabilistic and time varying nature.
- In MULTI-ANTENNA systems, like **Massive MIMO cellular networks**, this is especially true for obtaining accurate channel state information (CSI) is a central activity both for precoding the spatial streams before transmission and for coherently combining the received signals from each antenna.
- This difficulty is **fundamentally due to pilot contamination** – which is the interference of pilot symbols utilized by the users in neighboring cells – **and noise**.
- A low complexity and scalable (in terms of the number of antennas) channel estimator is very desirable for massive MIMO and **current solutions have non-trivial drawbacks**.
- The MMSE estimator is not preferred because of this complexity and large number of samples in proportion to number of antennas are required for channel correlation matrix estimation
- On the other hand, despite its low complexity, the LS estimator achieves significantly less accurate channel estimation than minimum mean square error (MMSE) estimation.

Introduction & Motivation

- The introduction of techniques from deep learning points to a potential remedy, since these techniques have been recently used for other challenging communication theory problems like **"An introduction to deep learning for the physical layer," "Power of deep learning for channel estimation and signal detection in OFDM systems" ,etc**. but without closed-form solutions.
- However, these are based on supervised deep neural networks (DNNs), which are fairly complex and typically require a large number of parameters to be trained with large datasets .Thus, they do not seem suitable for online channel estimation in wireless systems, where channels change quite rapidly.
- Promisingly, a recent special DNN design called a deep image prior ,**does not require training**, and thus avoids the need for a training dataset. This design is very appealing for **online DNN based channel estimation**, in which training a DNN may also not be tolerable in terms of latency.
- **The goal of this paper is to design an algorithm to estimate the channel using the ideas from deep image prior with reasonably good accuracy and complexity.**

System Model

- We consider a cellular network that has base stations with a large number of antennas and single antenna users. Specifically, base stations have M antennas and serve U users such that $U \ll M$.
- We assume that OFDM symbols with N_f subcarriers are transmitted in a time division duplex (TDD) frame structure.
- For the target base station the received signal in the frequency domain Y can be expressed as

$$\mathrm{Y} = \sum_{u \in S_u} \sqrt{\rho_u} \mathrm{H_u} \otimes \mathrm{x_u^H} + \sum_{v \in S_v} \sqrt{\rho_v} \mathrm{H_v} \otimes \mathrm{x_v^H} + \mathrm{Z}, \tag{1}$$

S_u is set of users connected to base station, ρ_u is the transmit power
H_u is channel between target base station and u_{th} user, x_u is the pilot sequence used for channel estimation.
S_v is the set of users that are connected to the interferer cells and H_v is the channel between the target base station and the interferer users
Z denotes the Gaussian Noise Matrix, with i.i.d. elements which have zero mean and variance σ^2.

Channel Estimate using LS and MMSE

The u^{th} user signal in the base station is

$$\mathrm{Y_u} = \sqrt{\rho_u} N_p \mathrm{H_u} + \sum_{v \in S_v} \sqrt{\rho_v} \tilde{\mathrm{H}}_\mathrm{v} + \mathrm{Z_u}, \tag{2}$$

$\underline{Y}_u$ is columnised version of Y_u

$$\underline{\hat{\mathrm{H}}}_\mathrm{u} = \mathrm{A_u} \underline{\mathrm{Y}}_\mathrm{u}, \tag{3}$$

where

$$\mathrm{A_u} = \begin{cases} \dfrac{1}{\sqrt{\rho_u} N_p} \mathrm{I}_{\mathrm{MN_f}} & \text{LS} \\ \sqrt{\rho_u} \mathrm{R}_{\underline{\mathrm{H}}_\mathrm{u}} (\Gamma_v + \rho_u N_p \mathrm{R}_{\underline{\mathrm{H}}_\mathrm{u}} + \sigma^2 \mathrm{I}_{\mathrm{MN_f}})^{-1} & \text{MMSE} \end{cases} \tag{4}$$

in which $\mathrm{R}_{\underline{\mathrm{H}}_\mathrm{u}}$ is the correlation matrix of the channel and

$$\Gamma_v = \sum_{v \in S_v} \rho_v \mathrm{R}_{\underline{\tilde{\mathrm{H}}}_\mathrm{v}}. \tag{5}$$

Deep Image Prior

- Conventional NN and SOTA CNNs used for restoration, classification and generation tasks require a large dataset of inputs and labels to train.
- Problem: They require a lot of data and time to learn the weights and parameters.
- The only information required to solve the problem is contained in the single degraded input image and the structure of the network used for reconstruction.
- Advantage of DIP: Data is not required and takes lesser time to run.

Deep Image Prior

- x_0 is the original noisy image which we have to restore.
- Input for the model Z and the weights of the model θ are randomly initialized.
- Produce output f(z) such that it is close to x_0

$$\Theta^* = \arg\min_{\Theta} ||f_\theta(z) - x_0||_2^2. \tag{10}$$

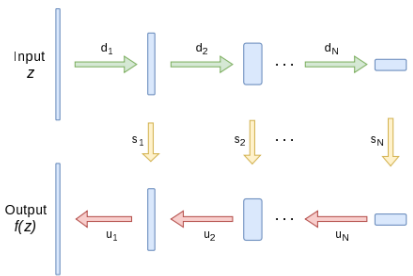


Figure: Model used in DIP (Ignore S1,S2,..,Sn-1 skip connections).

Deep Image Prior

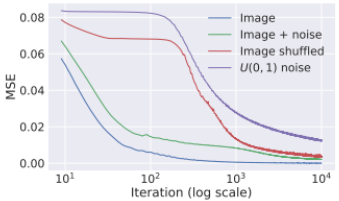


Figure: Learning curves for the reconstruction task using: a natural image, the same plus i.i.d. noise, the same randomly scrambled, and white noise. Naturally-looking images result in much faster convergence.

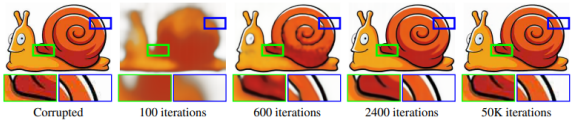


Figure: Blind restoration of a JPEG-compressed image.

Deep Decoder

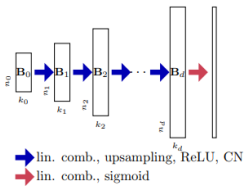


Figure: Deep Decoder Architecture

- Based on the same idea as Deep Image Prior ie. No need of data nor training.
- Uses a different architecture compared to DIP model. DD has an increasing size model rather than the hourglass model of DIP.

Deep Decoder Vs Deep Image Prior

DD uses only 1*1 convolutions unlike the usual CNN in DIP.Hence, DD uses way lesser parameters compared to DIP.

Results in lesser chance of overfitting to noise unlike DIP which fits after a certain number of iterations and needs early stopping to avoid that.

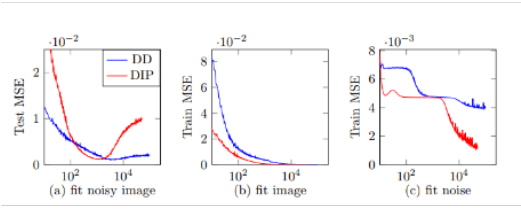


Figure: Denoising with the Deep Decoder and Deep Image Prior

Deep Channel Estimator Model

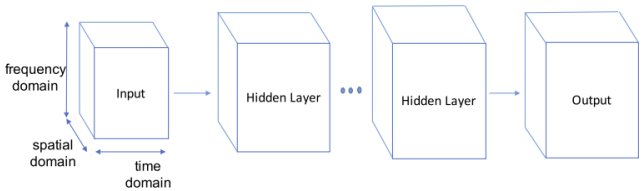


Figure: DCEM Architecture

Deep Channel Estimator Model

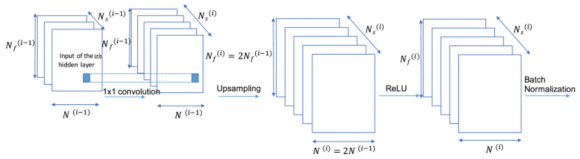


Figure: The structure of the i^{th} hidden layer, whose input dimension is $N_s^{(i-1)} \times N_f^{(i-1)} \times N^{(i-1)}$ and output dimension is $N_s^{(i)} \times N_f^{(i)} \times N^{(i)}$.

For $i = 0, 1, \ldots, l-2$

$$f_{\theta_i} = \text{BatchNorm}(\text{ReLU}(\text{Upsampler}(\theta_i \circledast Z_i)) \tag{7}$$

For the last hidden layer:

$$f_{\theta_{l-1}} = \text{BatchNorm}(\text{ReLU}(\theta_{l-1} \circledast Z_{l-1}), \tag{8}$$

The output of the Deep Channel Estimator:

$$\hat{Y}_T = f_{\theta_l}(f_{\theta_{l-1}}(\cdots f_{\theta_0}(Z_0))), \tag{6}$$

Objective Function :

$$\Theta^* = \arg\min_{\Theta} ||Y_T - \hat{Y}_T||_2^2. \tag{10}$$

Simulation Overview

- The performance metric we use is the normalized mean square error.

$$\text{NMSE} = \mathbb{E}\left[\frac{||\underline{\mathbf{H}}_u - \hat{\underline{\mathbf{H}}}_u||_2^2}{||\underline{\mathbf{H}}_u||_2^2}\right], \tag{11}$$

- Using MATLAB LTE toolbox to model MIMO channels and PyTorch to model the deep channel estimator.
- Stages:
  1. Single Antenna OFDM Communication
  2. Single-Cell Massive MIMO
  3. Multi-Cell Massive MIMO
- Comparing performance of LS, MMSE and DCE with different values of k (number of channels in the hidden layers).

Experimental Details

- The received signal or the output of the deep channel estimator is chosen to be a M $\times$ 2 $\times$ 64 $\times$ 64 matrix, where 2 represents the real and imaginary part, the first 64 represents the number of sub carriers, and the next 64 is the number of OFDM symbols and M represents the number of antennas.
- In order to optimize the number of channels per layer or the value of k, we take a single channel realization disturbed by the least noise (i.e the highest SNR in our range) and observe the convergence of its NMSE with the number of epochs by performing Adam optimization with a learning rate of 0.01.(Early Stopping)
- We find the number of epochs at which the lowest NMSE is achieved for a given k, and proceed to denoise the received signal for the aforementioned range of SNRs for the calculated number of epochs. This approach is often referred to as early stopping.

Variation of the MSE of the DD's output with respect to true H and noisy H with the number of epochs

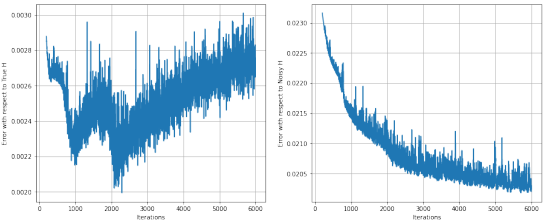


Figure: SNR = 15dB

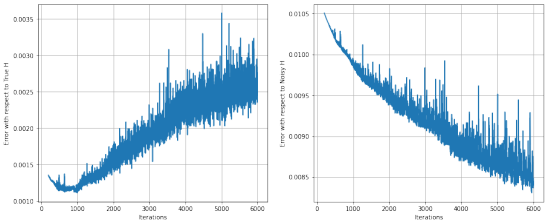
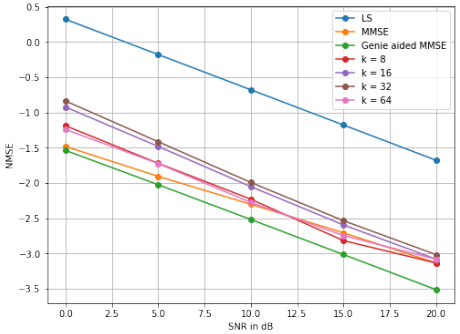


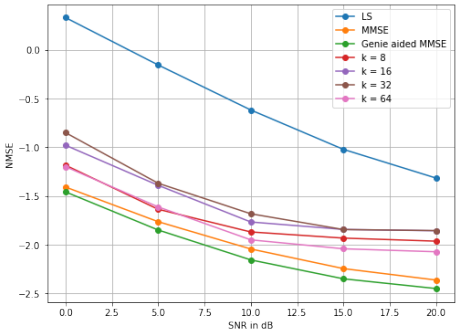
Figure: SNR = 20dB

- As we keep going through the iterations the the model fits the output with the noisy received OFDM grid and the loss function keeps decreasing.
- But the MSE w.r.t true OFDM grid first decreases and then increasing because of the overfitting of the noise.
- So we stop our iterations after an optimum epoch, which is often referred to as early stopping.

Single Antenna OFDM Communication



(a) Without Interence



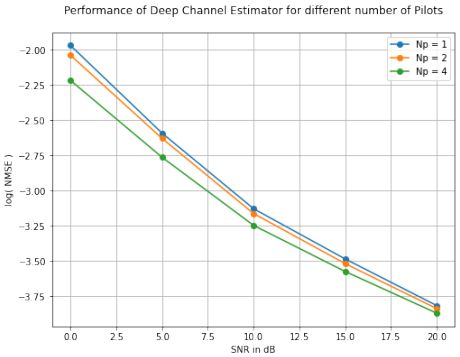
(b) With co-channel inteference (SIR = 6dB)

Figure: NMSE of the proposed estimator for different k and M = 1 with respect to SNR in comparison to LS and MMSE estimators.

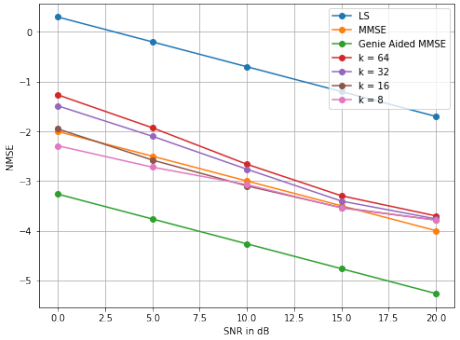
Single Antenna OFDM Communication

- The NMSE is lowest for k = 8 and k=64, gets progressively higher for k = 16 and k = 32. However, there is very little to tell apart the different architectures at SNR of 20 dB.
- Our channel estimator for k = 8 clearly outperforms LS and approaches the "MMSE" performance.
- The success of the deep decoder can be attributed to exploiting correlations in the OFDM grid.
- With 10% co-channel interference of SIR 6dB, we a see a similar trend in that k=8 and 64 perform better than other models.
- It might be beneficial to stop the iterations even earlier when we have interference but we do not have access to the noiseless received signal, we cannot ascertain when to stop

Single-Cell Massive MIMO



(a) NMSE of proposed estimator for different amount of pilots N_p



(b) NMSE of porposed estimator for different k and $M = 64$ with respect to SNR in comparison to LS and MMSE estimators.

Single-Cell Massive MIMO

- The deep channel estimator is mainly intended for multiple antennas and we choose M=64.
- Here,the spatial domain is used to stack up the real and imaginary domain, because this axis is more appropriate for uncorrelated samples in the architecture.
- Increasing the number of pilots from Np = 1 to Np = 4 benefits the NMSE, but no benefit is obtained beyond that. This is because the LTE-EPA model which has very high temporal correlation, and consequently needs very few pilots in the time domain to represent the channel accurately.
- At larger noise levels (or lower SNR), smaller values of k perform much better. However at higher SNR, due to early stopping, all the architectures tend to the same NMSE, with the higher k ones performing slightly better.

Multi-Cell Massive MIMO

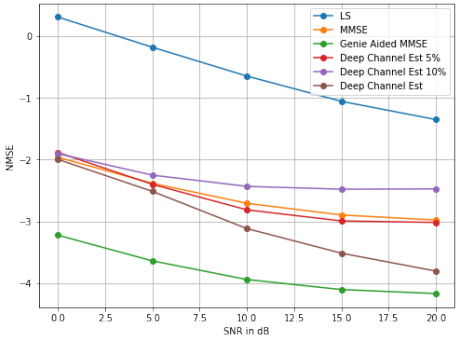


Figure: Random pilot contaminations at SIR = 6dB. Reference curves are in the presence of 10% interference at SIR = 6dB.

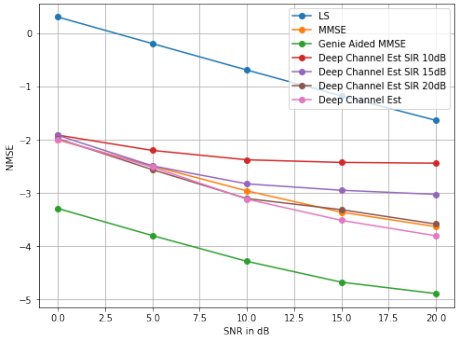


Figure: Contiguous pilot contamination. Reference curves are at SIR = 10dB.

Multi-Cell Massive MIMO

- The flattening out of the NMSE curve with increased interference is due to not patching the signal in the areas corrupted by interference beyond a certain limit.
- To further quantify the pilot contamination performance of our estimator, we verify its robustness for a different power allocation method. Accordingly, pilots are not only randomly but also contiguously distributed over the resource elements. To be more precise, 2 blocks of 8×8 squares (corresponding to 3% of the overall time-frequency grid) are chosen randomly, in which interference at SIRs of 10, 15 and 20dB is injected.
- Although the deep channel estimator in this case can tolerate lower powers of interference than the previous case, its performance, as illustrated, is still better than LS estimator for all SNRs and very close to MMSE estimator for the SIRs that are greater than 15 dB.

Complexity Analysis

- The number of multiply-accumulate operations (MACs) for the DCE can be easily seen to be directly proportional to the number of channels, k, which for a multi-user setting is simply twice the number of receive antennas 2*M. So, the **complexity of the deep channel estimator is O(M)**.
- On the other hand, the **MMSE estimator requires** $(4M^3\text{-}M)/3$ **complex multiplications and M complex divisions** [Eq. 3].
- So, there is a drastic decrease in complexity brought out by the Deep channel estimator for very large number of antennas.
- There exists an inverse relationship between the number of parameters to be trained and the required epochs (result has in fact been proven for the case of supervised learning of a single hidden layer neural network).[1]

[1] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks," Mar. 2019, arXiv:1903.11680. [Online]. Available: https://arxiv.org/abs/1903.11680

Complexity Analysis

- The model complexity is determined by the computation capability (latency) and the tolerance allowed.
- The number of multiply-accumulate operations (MACs) for the DCE can be found to be of the form M*f(k)+g(k). So, as far as M is concerned, the complexity of the deep channel estimator is O(M).
- On the other hand, the MMSE estimator requires $(4M^3 - M)/3$ complex multiplications and M complex divisions [Eq. 3].
- So, there is a drastic decrease in complexity brought out by the Deep channel estimator for a very large number of antennas.

Conclusion

- This deep channel estimator exhibits superior performance compared to LS and MMSE estimators that have no inherent way of dealing with pilot contamination
- Promisingly, this low-complexity estimator outperforms the more complex MMSE estimator, in which the channel correlation matrices are estimated from the samples, and even approaches the "Genie-Aided MMSE" where the channel statistics are perfectly known for free.
- The salient features of the proposed estimator are as follows. **The number of parameters scale at a rate less than the square root of number of antennas**, which yields hundreds or thousands of weights as opposed to millions of parameters in conventional DNNs.
- Future work in this area could be to study the deep channel estimator for high mobility channels. Similarly, observing the performance of the deep channel estimator for mmWave channels seems intriguing.

Thank You

Theoretical Analysis

- The denoising capability of the proposed LS-type deep channel estimator determines how close it can approach the MMSE estimation performance.
- Theorem 1: The proposed LS-type deep channel estimator achieves the MMSE estimator performance as the product of the number of base station antennas M , number of subcarriers N_f and coherence time interval N goes to infinity, assuming there is no pilot contamination. That is,

$$\lim_{MN_fN\to\infty} \frac{\epsilon_{\mathrm{dce}}}{\epsilon_{\mathrm{mmse}}} = \zeta, \tag{11}$$

  where ϵ_{dce} and ϵ_{mmse} are the channel estimation errors for the proposed deep channel estimator and conventional MMSE estimator, respectively, and ζ is a scalar and depends on the number of the DNN parameters.
- we prove that our architecture can filter all the noise for asymptotically high- dimensional signals, e.g., for massive MIMO-OFDM, and can achieve the MMSE estimator performance.

Step - I

$$\min_{\Theta} \frac{||\mathbf{n} - \mathbf{n}_{\text{fit}}(\Theta)||^2}{||\mathbf{n}||^2} \geq 1 - c \left(\frac{\left(\prod_{i=0}^{l-1} N_s^{(i)}\right)^{\frac{2}{l}} \log\left(\prod_{i=0}^{l-1} N_f^{(i)} N^{(i)}\right)}{N_s^{(l)} N_f^{(l)} N^{(l)}} \right) \tag{19}$$

with the probability $\{1 - 2(N_f^{(0)} N^{(0)})^{-\left(\prod_{i=0}^{l} N_s^{(i)}\right)^{\frac{2}{l+1}}}\}^2$

$$\lim_{M \to \infty} \left\{ 1 - 2(N_f^{(0)} N^{(0)})^{-\left(\prod_{i=0}^{l} N_s^{(i)}\right)^{\frac{2}{l+1}}} \right\} = 1 \tag{20}$$

$$n_{\text{supp}} = \inf \left\{ \frac{||\mathbf{n} - \mathbf{n}_{\text{fit}}(\Theta^*)||^2}{||\mathbf{n}||^2} \right\} = 1 - c \left(\frac{\left(\prod_{i=0}^{l-1} N_s^{(i)}\right)^{\frac{2}{l}} \log\left(\prod_{i=0}^{l-1} N_f^{(i)} N^{(i)}\right)}{N_s^{(l)} N_f^{(l)} N^{(l)}} \right) \tag{21}$$

[2] R. Heckel and P. Hand, "Deepdecoder: Concise image repre-sentations from untrained non-convolutional networks," Feb. 2019,arXiv:1810.03982. [Online]. Available: https://arxiv.org/abs/1810.03982

Step - II

$$\log\left(\prod_{i=0}^{l-1} N_f^{(i)} N^{(i)}\right) = \log\left(\frac{(N_f^{(l-1)} N^{(l-1)})^l}{\prod_{i=0}^{l-1} 4^i}\right) \tag{22}$$

$$\log\left(\frac{(N_f^{(l-1)} N^{(l-1)})^l}{\prod_{i=0}^{l-1} 4^i}\right) \ll N_f^{(l)} N^{(l)} \tag{23}$$

Step - II

From the DIP paper[3]

$$\min_{\Theta} E(Y_T, \hat{Y}_T) \tag{24}$$

$$\min_{\Theta} E(Y_T, G(\Theta)) + R(G(\Theta)) \tag{25}$$

$$R(G(\Theta)) = ||\Theta||_F^2 + \langle A, \Theta^T \Theta \rangle \tag{26}$$

$$(N_s^{(l)})^2 = (\theta_l \circledast N_s^{(l-1)})^2 \tag{27}$$

$$N_s^{(l-1)} = \sqrt{2r} \tag{28}$$

where r is number of independently received samples which is rank of the channel.[4]

[3] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," inProc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018,pp. 9446–9454.

[4] S. Du and J. D. Lee, "On the power of over-parametrization in neural networks with quadratic activation," Jun. 2018,arXiv:1803.01206.[Online]. Available: https://arxiv.org/abs/1803.0120

Step - III

$$\epsilon_{\mathrm{mmse}} = \mathrm{tr}(\mathsf{C}), \tag{29}$$

$$\mathsf{C} = \mathsf{R} - \mathsf{R}\left(\mathsf{R} + \frac{1}{\mathrm{SNR}}\mathsf{I}\right)^{-1}\mathsf{R}. \tag{30}$$

$$\epsilon_{\mathrm{mmse}} = \sum_{m=1}^{\mathrm{rank}(\mathsf{R})}\left(\lambda_m - \frac{\lambda_m^2}{\lambda_m + 1/\mathrm{SNR}}\right). \tag{31}$$

$$\lim_{MN_f N\to\infty} \epsilon_{\mathrm{mmse}} = 0. \tag{32}$$

$$\epsilon_{\mathrm{ls}} = \frac{\mathrm{rank}(\mathsf{R})}{\mathrm{SNR}}. \tag{33}$$

Step - III

Replacing n with $n_f it$ in the SNR of the LS estimate we get equation 34.

$$\epsilon_{\mathrm{dce}} = \frac{\mathrm{rank(R)}}{\mathrm{SNR}} \times \left(\left(1 + \mathrm{rank(R)c} \times \frac{l \log(N_f N) - \log(4^{0.5l(l-1)})}{M N_f N} \right)^{\frac{1}{2}} - 1 \right)^2. \tag{34}$$

$$\lim_{M N_f N \to \infty} \epsilon_{\mathrm{dce}} = 0, \tag{35}$$

Excerpts from Important References

For a k hidden node shallow net-work with quadratic activation and n training data points, we can show that as long as $k \geq \sqrt{2n}$, over-parametrization enables local search algorithms to find a **globally** optimal solution for general smooth and convex loss functions[5]

[5] S. Du and J. D. Lee, "On the power of over-parametrization in neural networks with quadratic activation," Jun. 2018,arXiv:1803.01206.[Online]. Available: https://arxiv.org/abs/1803.01206

Excerpts from Important References

Proposition 1. Consider a deep decoder with one layer and arbitrary upsampling and input matrices. That is, let $B_1 \in \mathbb{R}^{n_1 \times k}$ and $U_1 \in \mathbb{R}^{n \times n_1}$. Let $\eta \in \mathbb{R}^n$ be zero-mean Gaussian noise with covriance matrix I. Assume that $k^2 \log(n_1)/n \leq 1/32$. Then, with probability at least $1 - 2{n_1}^{-k^2}$,

$$\min_{\mathrm{C}} ||\mathrm{G}(\mathrm{C}) - \eta||_2^2 \geq ||\eta||_2^2 \left(1 - 20\frac{k^2 \log(n_1)}{n}\right).$$

The proposition asserts that the deep decoder can only fit a small portion of the noise energy,precisely a proportion determined by its number of parameters relative to the output dimension, n.Our simulations and preliminary analytic results suggest that this statement extends to multiple layers in that the lower bound becomes $\left(1 - c\frac{k^2 \log(\prod_{i=1}^{d} n_i)}{n}\right)$, where c is a numerical constant.