

# MMSE Based Channel Estimator

Zhao-Jie, Luo

*janny00kevin@gmail.com*

Advisor: Professor Carrson C. Fung

National Yang Ming Chiao Tung University

Mar. 14, 2024

- ① Carrson C. Fung, and Dmytro Ivakhnenkov, "Model-Driven Neural Network Based MIMO Channel Estimator"
- ② M. Eisen, C. Zhang, L.F.O. Chamon, D.D. Lee and A. Ribeiro, "Learning optimal power allocations in wireless systems," *IEEE Trans. on Signal Processing*, vol. 67(10), pp. 2775-2790, May 2019.
- ③ OpenAI Spinning Up introduction to RL Part 3: Intro to Policy Optimization

# Problem Formulation

Our objective is to minimize the expected mean square error:

$$\min_{\hat{\mathbf{h}}} \mathbb{E}_{\mathbf{y}, \mathbf{h}} \left[ \left\| \mathbf{h} - \hat{\mathbf{h}} \right\|_2^2 \right]$$

and it can be written in epigraph form as:

$$\begin{aligned} & \min_{t, \mathbf{h}} t \\ & s.t. \mathbb{E}_{\mathbf{y}, \mathbf{h}} \left[ \left\| \mathbf{h} - \hat{\mathbf{h}} \right\|_2^2 \right] \leq t \end{aligned}$$

# Primal-Dual Optimization Method (1)

We use parameterize channel estimator so that  $\hat{\mathbf{h}} = \phi(\mathbf{y}; \boldsymbol{\theta})$ , with  $\boldsymbol{\theta}$  denoting the parameters of the neural network.

Then the Lagrangian function of (15) can be written as

$$\begin{aligned}\mathcal{L}(\hat{\mathbf{h}}, t, \lambda) &= t + \lambda \left( \mathbb{E}_{\mathbf{y}, \mathbf{h}} \left[ \|\mathbf{h} - \hat{\mathbf{h}}\|_2^2 \right] - t \right) \\ &= t + \lambda \left( \mathbb{E}_{\mathbf{y}, \mathbf{h}} \left[ \|\mathbf{h} - \phi(\mathbf{y}; \boldsymbol{\theta})\|_2^2 \right] - t \right)\end{aligned}$$

# Primal-Dual Optimization Method (2)

It is uncertain whether or not the duality gap equals zero.

However, the stationary point of  $\mathcal{L}(\hat{\mathbf{h}}, t, \lambda)$  can be found via the KKT conditions by solving for the primal and dual variables alternately using gradient descent and ascent, respectively:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_{\boldsymbol{\theta},k} \lambda_k \nabla_{\boldsymbol{\theta}_k} \mathbb{E} \left[ \|\mathbf{h} - \phi(\mathbf{y}; \boldsymbol{\theta}_k)\|_2^2 \right]$$

$$t_{k+1} = t_k - \alpha_{t,k} (1 - \lambda_k)$$

$$\lambda_{k+1} = \left[ \lambda_k + \alpha_{\lambda,k} \left( \mathbb{E} \left[ \|\mathbf{h} - \phi(\mathbf{y}; \boldsymbol{\theta}_{k+1})\|_2^2 \right] - t_{k+1} \right) \right]_+$$

# Policy Gradient (1)

We have the policy gradient theorem:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\tau}[G(\tau)] = \mathbb{E}_{\tau} \left[ \sum_{t=0}^{T-1} G(\tau) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t) \right]$$

At each time step,  $t = 1, \dots, T - 1$ :

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\tau}[G(\tau)] = \mathbb{E}_{\tau} [G(\tau) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)]$$

And we can estimate the policy gradient with sample mean:

$$\widehat{\nabla}_{\boldsymbol{\theta}} \mathbb{E}_{\tau}[G(\tau)] = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} G(\tau) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)$$

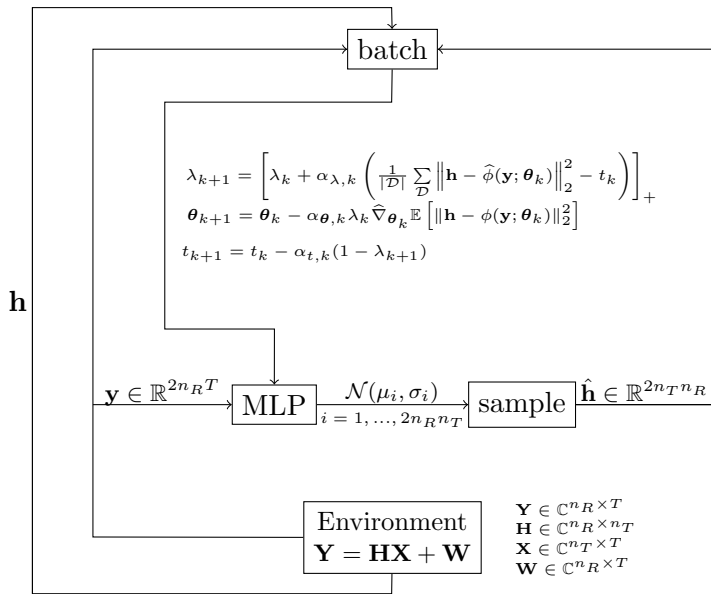
# Policy Gradient (2)

Our goal is to minimize the mean square error, by substituting  $\mathbb{E}_\tau[G(\tau)]$ ,  $\pi_\theta(A_t|S_t)$  with  $\mathbb{E}_{\mathbf{y}, \mathbf{h}} \left[ \|\mathbf{h} - \phi(\mathbf{y}; \theta)\|_2^2 \right]$ , and  $\pi_\theta(\hat{\mathbf{h}}|\mathbf{y})$ . Thus, we obtain the estimated policy gradient for our problem:

$$\widehat{\nabla}_\theta \mathbb{E}_{\mathbf{y}, \mathbf{h}} \left[ \|\mathbf{h} - \phi(\mathbf{y}; \theta)\|_2^2 \right] = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \left\| \mathbf{h} - \hat{\phi}(\mathbf{y}; \theta) \right\|_2^2 \nabla_\theta \log \pi_\theta \left( \hat{\mathbf{h}} | \mathbf{y} \right),$$

where  $\hat{\phi}(\mathbf{y}; \theta) = \hat{\mathbf{h}}$  is the sampled output of the policy.

# Experiment Diagram





# Simulation Result

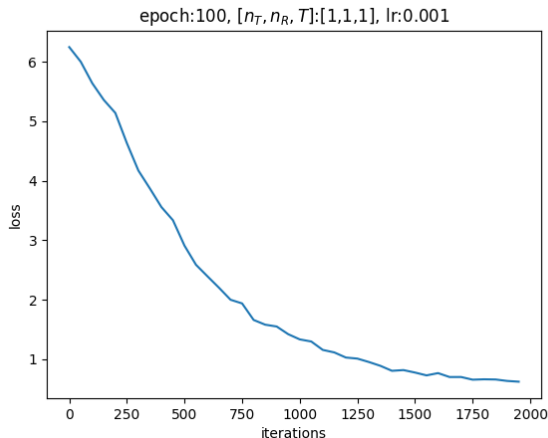


Figure: hidden layer sizes:  $[64,32]$ ,  $\mu_H = 5, \sigma_H = 0.2, \mu_W = 0, \sigma_W = 0.1$

# Simulation Result

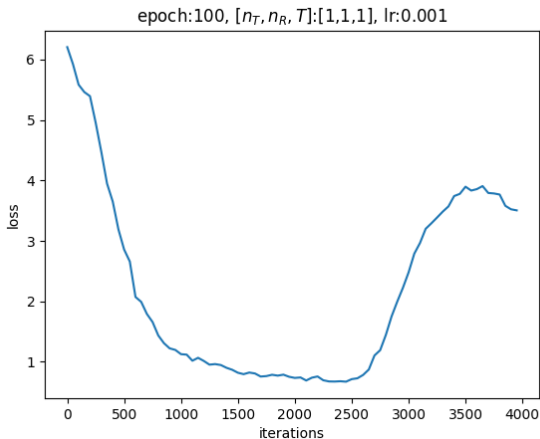


Figure: increase the iterations from 2000 to 4000