Neural Network solutions to Witsenhausen problem

Jiaojiao Fan
GTID:903565753
Email: jiaojiaofan@gatech.edu

***Abstract*—In this report, several neural networks with different structures are implemented to solve Witsenhausen problem. Other improving strategies include optimizers, initializations and forced function fixing. Finally, the result are compared with former people and a better result is obtained. Also, the shortcoming of the neural network also shows in this project. The neural network may be stuck into a near local minima.**

1. Introduction

In this report, we proposed several solutions to the well-known and still unsolved Witsenhausen counterexample. [1] There have been some meaningful tries to detect the global minima of the min problem, such as Lee [2]and M. Barglietto [3] Some of their manipulations are also refered in this project. Other than that, thanks to the development of the neural networks, many other meaningful attempts are also taken such as input convex neural network (ICNN) structure [4] Different results would be listed to show the effect.

2. The Witsenhausen Counterexample

The Witsenhausen counterexample has been outstanding for more than 50 years. It is formulated by Hans Witsenhausen in 1968. [1] It is a counterexample to a natural conjecture that in a system with linear dynamics, Gaussian disturbance, and quadratic cost,affine control laws are optimal to minimize the cost. However, Witsenhausen courterexample, shown in figure below, has nonlinear control laws

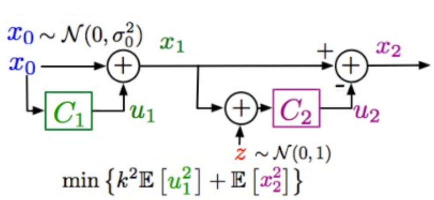

Figure 1. Witsenhausen couterexample

that outperform all linear laws.

$$f(x) = \gamma_1(x) + x \qquad g(x) = \gamma_2(x) \tag{1}$$

As a result, $f(x_0) = x_1$ and $g(x_1 + z) = u_2$. Out goal is to minimize the quadratic cost:$k^2\,\mathbb{E}[U_1^2] + \mathbb{E}[X_2^2]$, which can also be written as

$$\begin{aligned}\min J(f,g) :=& k^2\,\mathbb{E}[f(X_0) - X_0]^2 + \\ & \mathbb{E}[(f(X_0) - g(f(X_0) + N))^2]\end{aligned} \tag{2}$$

In equation (2), there is a parameter k^2, which in fact determines the cost gap between the linear controller and nonlinear controller [3]. If k^2 is smaller, the gap is bigger. For better comparison with the results got by previous researchers, k^2 is set as 0.04 in this report.

In addition, it is already known that $f(x)$ must have some strict property to be optimal [5] :

- *Any optimal controller f is a strictly increasing unbounded piecewise real analytic function with a real analytic inverse*

This means f has to be smooth enough. But interestingly, the neural network(NN) optimized result is exactly opposite from this property. The sharper the f becomes (opposite to smooth), the smaller the cost is.

3. Basic Neural Network Setup

The whold process could be generally separated into 4 parts:
1) Initialization setup for the f net and g net
2) Train the NNs using Gaussian distribution data. In this report, all data keeps the consistency: $x_0 \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 5$, and $N \sim \mathcal{N}(0, 1)$.
3) Fix f net and continue to train g net.

3.1. Neural Network Architecture

Basically, two NNs are taken to represent f and g seperately using Pytorch structure. [1] For f net and g net, all layers are linear layers and activated by CELU [6] function, i.e.

$$CELU(x) = max(0, x) + min(0, \alpha * (exp(x/\alpha) - 1)) \tag{3}$$

is used since it makes the activation function continuously differentiable and improves the performance in initialization setup process. It is worth noting that CELU is convex and monotone increasing activation function. For f net structure,

1. The code of this project could be found at: https://github.com/sbyebss/Witsenhausen

we tried ICNN as f net's integral function:F net, i.e. f NN works as the derivative function of the function represented by F NN. We also have tried ResNet [7] since it performs much better than ordinary linear layer NNs.

3.2. optimizer

For updating parameters of two NNs, we first use Stochastic Gradient Descent (SGD) and then used ADAM. In comparison, SGD performs much slower and becomes not stable while entering plateau of loss decreasing. ADAM increases the stability and speed a lot. So we focus on the better optimizer compared to ADAM later. There were a lot of variation of ADAM during the past several years. We mainly care whether the optimizer could lead the NN to the global optimizer.

Acknowledgment

Thanks to my advisor: Dr. Yongxin Chen and the other PhD studens: Rahul Singh and Qinsheng Zhang for their kind help. They give me a lot of ideas about how to improve the performance during the process of project.

References

[1] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM Journal on Control*, vol. 6, no. 1, pp. 131–147, 1968.

[2] J. T. Lee, E. Lau, and Y.-C. Ho, "The witsenhausen counterexample: A hierarchical search approach for nonconvex optimization problems," *IEEE Transactions on Automatic Control*, vol. 46, no. 3, pp. 382–397, 2001.

[3] M. Baglietto, T. Parisini, and R. Zoppoli, "Numerical solutions to the witsenhausen counterexample by approximating networks," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1471–1477, 2001.

[4] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 146–155.

[5] Y. Wu and S. Verdú, "Witsenhausen's counterexample: A view from optimal transport theory," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 5732–5737.

[6] J. T. Barron, "Continuously differentiable exponential linear units," *arXiv preprint arXiv:1704.07483*, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.