Dear readers,

Eq.(10) can have a different version of the Kronecker product.

According to Theorem 2.2 in [1], the following should hold:

$$\frac{\partial vec\left(aA(x)\right)}{\partial x}=\left(I\otimes a\right)\frac{\partial vec\left(A(x)\right)}{\partial x} \tag{1}$$

where a is a vector, A is a matrix depending on a vector x. In this case, the order of terms in the Kronecker product of Eq.(10) should be reversed.

We use eigenvalue-decomposition to reduce the computational burden when solving the large trust-region optimization problem via the python function 'def TRS_solver_Eigen' in the file 'Uav_mhe_SL_Hessian_trust_region_neural.py'. However, strictly following the definition in Theorem 2.2 can cause complex eigenvalues. Therefore, we opted to use the definition in Eq.(4) of [2] instead.

Our algorithm can still work with definition (1) by taking only the real parts of eigenvalues, although this requires re-tuning the hyperparameters of the trust-region method (TRM). Below are two examples of mean loss obtained this way, demonstrating performance comparable to that in our paper.

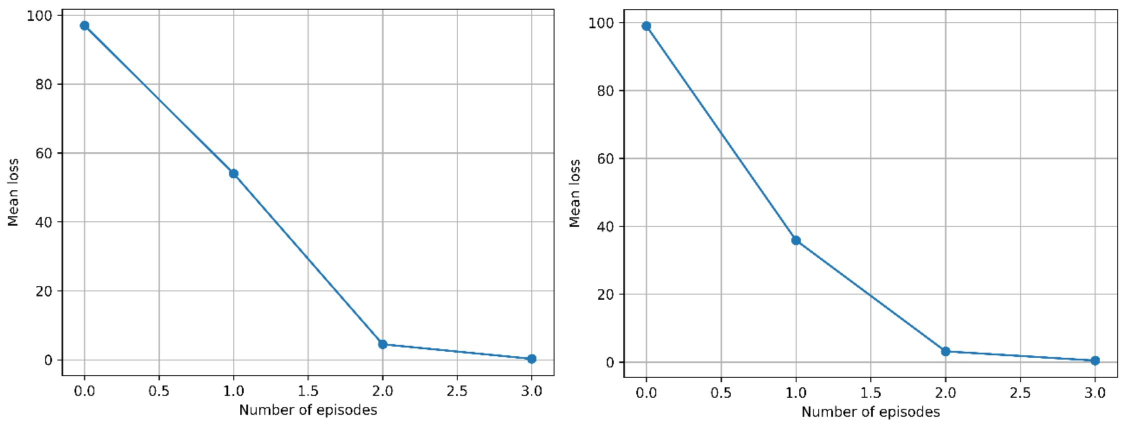

Fig.1 The training time and ultimate loss of these two examples are (281 s, 0.25) and (282 s, 0.45), respectively, both of which are comparable to the performance in our paper.

This is an interesting phenomenon worth further investigation.

[1]. Magnus, Jan R., and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.

[2]. Dyro, Robert, Edward Schmerling, Nikos Arechiga, and Marco Pavone. "Second-Order Sensitivity Analysis for Bilevel Optimization." In *International Conference on Artificial Intelligence and Statistics*, pp. 9166-9181. PMLR, 2022.

王秉亨

WANG Bingheng

1-Nov-2024