

Gene-regulatory Kolmogorov-Arnold networks

Project outline

Gene regulatory networks (GRNs) try to model the regulatory relationship between genes as a directed graph $D = (V, E)$, where the nodes V correspond to the genes and the interactions E model the regulatory relationships between genes the $v \in V$. State-of-the-art models take as input a gene expression matrix $\mathbf{X} \in \mathbb{R}^{|G| \times N}$ for genes G and N samples. Furthermore, current models for GRN inference assume that for a given target gene $t \in V$ all its regulators $R = \{r \mid r \in V \wedge (r, t) \in E\}$ are independent. However, this disregards higher-order effects between two different regulators $r_1, r_2 \in R, r_1 \neq r_2$ of t , which are known to exist.

Kolmogorov-Arnold networks (KANs) [Liu et al., 2024] have been recently proposed as an alternative to classical Multi-Layer-Perceptrons (MLPs). KANs modify MLPs by replacing the fixed activation functions on the neurons with learnable activation functions. Furthermore, KANs have been shown to be able to learn mathematical functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ given the input and output values. This suits perfectly the problem of GRN inference because here we want to learn the regulatory relationship $f : \mathbb{R}^{|R|} \rightarrow \mathbb{R}$ between the regulatory genes R and a target gene v .

Project goal

The steps of the project are:

1. Get familiar with Kolmogorov-Arnold networks (KAN tutorial).
2. Build and train a KAN for a regulatory relationship given training, validation and testing data.
3. Evaluate the trained model (pytorch evaluation metrics, qualitative analysis of the network).

Requirements:

- Python and PyTorch knowledge
- Familiarity with using UNIX environments and usage of the command line

References

Z. Liu, Y. Wang, S. Vaidya, F. Ruelle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark. Kan: Kolmogorov-arnold networks, 2024. URL <https://arxiv.org/abs/2404.19756>.