Reinforcement Learning for Hyperparameter Optimization of Machine Learning Models

Ammar Hussain, Mavi Ruiz-Blondet, David Stanley

General problem: Our original proposal was to address the issue of hyperparameter optimization by using reinforcement learning (RL). We planned to use the stock market as a test case for this application. Our search for papers returned a few different applications of reinforcement learning, besides the original application presented in the proposal. Some of these papers use RL to perform stock market prediction, either using a value or a policy search. In these models the hyperparameters are chosen manually. The second application was the use of RL for finding hyperparameters or network architectures. It has been very interesting to see the different definitions of state, reward and action can have a very straightforward or a much more abstracted application.

Reinforcement learning applied directly to the stock market: For the papers In this section, the states generally reflect states from the stock market (such as high, low) and it could include states from other sources as well, such as the weather. The actions can be mapped to [-1, 0, 1] which represent sell, hold and buy. The reward directly maps to the profit/loss after each transaction. *Xiao-Yang et al.* main contributions were: 1) Modularize the environment, the RL agents and the applications they interact with, such that novel RL agents, or new applications (such as the latest NFT rage) or new features (such as sentiment analysis from twitter feeds) can easily be added to the model. 2) creates Open-AI gym environments that can take in both historical data and live trading APIs, so that trained models can easily be deployed. *Zhouran et al.* implemented a Deep Deterministic Policy Gradient algorithm, which is a form of deep RL, where the deep part allows for modelling the complex environment and the RL part handles sequential decision making process, which is optimal for time-series data. The authors prefer to use direct policy search over value based RL such as Q-learning due to the large size of action-state spaces. The states correspond to prices of stocks in the market, amount of holdings and cash balance. It also adds replay buffers to reduce the correlation between the time-series data. *Yu Deng et al.* also implemented deep RL for stock market prediction, however, they used the deep network as a way to model the complex environment of stock fluctuation. The states are then features extracted by this deep network which go into the RL for the final decision, the reward of that output is then used to train the RL and the deep network. As the input layer to the deep network, the authors chose a fuzzy layer to deal with the high uncertainty and noisiness of the data. The main challenge of this architecture is the training, as it has many layers and they are recurrent due to the RL architecture. To address the recurrency issue, the authors use backpropagation through time (BPTT). This flattens the network, creating lots of layers. To avoid diminishing gradients, the authors use task-aware BPTT, where they attach virtual links from the final reward to each layer in the network, in addition to the original gradient.

Reinforcement learning applied to hyperparameter optimization: In this section, we discuss 3 papers that use RL in different ways to improve the performance of existing ML algorithms by automatically optimizing their hyperparameters. The work of Zoph and Le (2016) used RL to train a recurrent neural network, called the controller network, to create different neural network architectures with variable number of hidden nodes and layers. The controller ‘walks’ through the different nodes on the network, and chooses an aggregation and an activation function. It can also choose to add nodes and layers to the network. The reward of the network is how well the ‘child’ network classifies a test set. This means that every step requires training and testing a network, so the authors set up a training scheme that replicates the controllers in different servers and trains ~50 child networks at once.

Jomaa et al (2019) applies reinforcement learning to the problem of hyperparameter optimization. It formulates hyperparameter optimization as a Markov decision process, with explicit definitions for the state, action, reward function and transition function. Furthermore, the authors aim to allow the reinforcement learning algorithm to generalize to novel, unseen datasets. To this end, the authors evaluate the performance of their algorithm across 50 datasets taken from the UCI machine learning repository.

Unlike Zoph and Le (2016), which has a controller traverse the network, the RL agent in Jomaa et al estimates a new set of network parameters at each timestep. More specifically, their MDP is formulated in the following manner. First, the reward function is considered to be cross validation loss of the model, trained with parameters λ. Next, the actions that the agent takes is to choose a new set of model parameters, λ. Therefore, the space of actions is equivalent to the space of possible hyperparameter configurations. The state is defined as the history of all hyperparameter configurations explored and their corresponding responses, since initialization up to the present time step. Additionally, the state includes a set of “metadata” features, which serve to provide information about the statistical properties of the current dataset. These include measurements like the skewness, max, and minimum of various properties of the dataset. We note that the fact that the state includes the entire history of previous states implies that the state grows at each time step. It also allows the algorithm to skirt the memoryless property of the MDP, although the authors do not provide details for this decision. The underlying model the authors attempt to optimize is an LSTM model, which they apply to the aforementioned 50 UCI datasets. The environment used is the OpenAI framework, and their code is available on GitHub.

Lastly, Dong et al (2018) focuses on applying reinforcement learning to optimizing the hyperparameters of object tracking algorithms. These algorithms track the locations of objects in video data streams. The authors follow a similar approach to that used by Jomaa et al (2019). They propose a means for not only optimizing hyperparameters for an object tracking algorithm, but also for optimizing the hyperparameters of each specific video sequence. This is in contrast to other approaches, that pick a single set of hyperparameters and hold them fixed for all inputs. For example, the Siam-py algorithm, which they use as the basis for their optimization, originally had its 5 hyperparameters optimized by a grid search on a validation dataset. More generically speaking, this is akin to having an algorithm that adapts its hyperparameters according to the dataset it is processing. For their reinforcement learning approach, the reward is defined as the tracking accuracy and the goal is to maximize the expected cumulative tracking accuracies. They make use of a neural network to represent the agent. Unlike the paper by Jomaa et al., their definitions of state space and loss are specific to the problem of video object tracking.

Related papers: Two additional papers related to stock market prediction were added to increase features or data for our task. The first one, written by *Araci,* introduces an open source natural language processing (NLP) tool that has been specifically trained to understand valence (positive or negative) on financial texts. This tool could be used as an additional input, besides the simple swing of stock price, to our environment variables. The second article by *Roozen and Lelli* introduces a dataset of earning calls transcripts from 10 popular stocks in NASDAQ from 2016 to 2020. This dataset could feed the NLP classifier developed by Araci, which in turn could become one of our best predictors for stock market swing, given that the research area in sentiment analysis in relation to stock market prediction is a very active area of research, which would suggest it is a meaningful path to follow.

Future work: Predicting the stock market has been a classical exercise for Machine Learning researchers and students. There have been several articles that apply direct policy search RL combined with deep neural networks to make the best predictions. On our main topic of interest, which is using RL to train hyperparameters, we found a huge variability between papers on how to conceptualize states, actions and rewards in order to achieve a model that could find the right hyperparameters for a child model. Given that the deep direct RL has been successfully used in stock market predictions multiple times, but that it has many parameters to optimize, combining that with a novel conceptualization of states, actions and rewards to train a model that could optimize the deep DRL has the potential to allow this powerful tool to become a turn key model, not only for stock market prediction, but possibly for time series data more generally.

References

Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." *arXiv preprint arXiv:1908.10063* (2019).

Deng, Yue, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. "Deep direct reinforcement learning for financial signal representation and trading." *IEEE transactions on neural networks and learning systems* 28, no. 3 (2016): 653-664.

Dong, Xingping, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, and Fatih Porikli. "Hyperparameter optimization for tracking with continuous deep q-learning." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 518-527. 2018. https://ieeexplore.ieee.org/document/8578159

Jomaa, Hadi S., Josif Grabocka, and Lars Schmidt-Thieme. "Hyp-rl: Hyperparameter optimization by reinforcement learning." arXiv preprint arXiv:1906.11527 (2019). https://arxiv.org/pdf/1906.11527.pdf

Liu, Xiao-Yang, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." *arXiv preprint arXiv:2011.09607* (2020).

Roozen, Dexter, and Francesco Lelli. "Stock Values and Earnings Call Transcripts: a Dataset Suitable for Sentiment Analysis." (2021).

Xiong, Zhuoran, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. "Practical deep reinforcement learning approach for stock trading." *arXiv preprint arXiv:1811.07522* (2018).

Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." arXiv preprint arXiv:1611.01578 (2016). https://arxiv.org/abs/1611.01578

Slack Upload