The Faculty of Computer Science
Bachelor’s Programme “Data Science and Business Analytics”

The work is performed by
4th year DSBA student
Aksenov Nikita

Supervisor
Denis Bogutskiy
HSE Lab “Artificial Intelligence in mathematical finance”

Transfer Learning for Optimal Market Making in Synthetical Markets

Moscow

Content

Introduction & Research goal

Research Goal

The research studies **how Transfer Learning affects training time and performance** of Reinforcement Learning models in context of optimal Market Making

Relevance

Usage of **RL algorithms in Market Making** is a new way of constructing Market Making algorithms. However, the approach requires a lot of data for a complicated Neural Network to converge and to show decent results. We believe that this problem can be solved by applying transfer learning to the Reinforcement learning AI model.

Problem statement

The problem statement is to compare the convergence speed of a RL model and a model with transfer learning, applied to solving the Market Making task on a synthetic market.

Tasks

1. Study properties of the simulated data
2. Generate features to increase classification accuracy
3. Train and test classification models
4. Build environment for RL model training
5. Train RL models in several configurations (with and without transfer learning)
6. Assess the performance of RL models with different configurations and training time

Agent Based Simulator

Agent Based Simulator

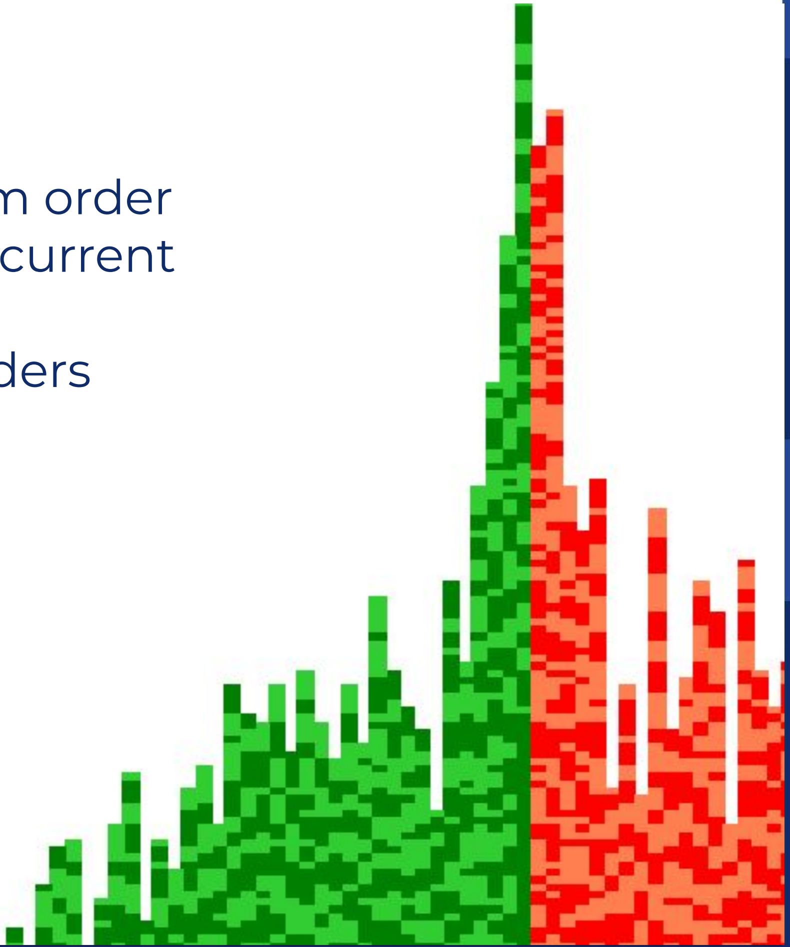
Agents:
- Random
- Chartist
- Fundamentalist
- Market Maker

Assets:
- Stock

Exchange

Iterative cycle:

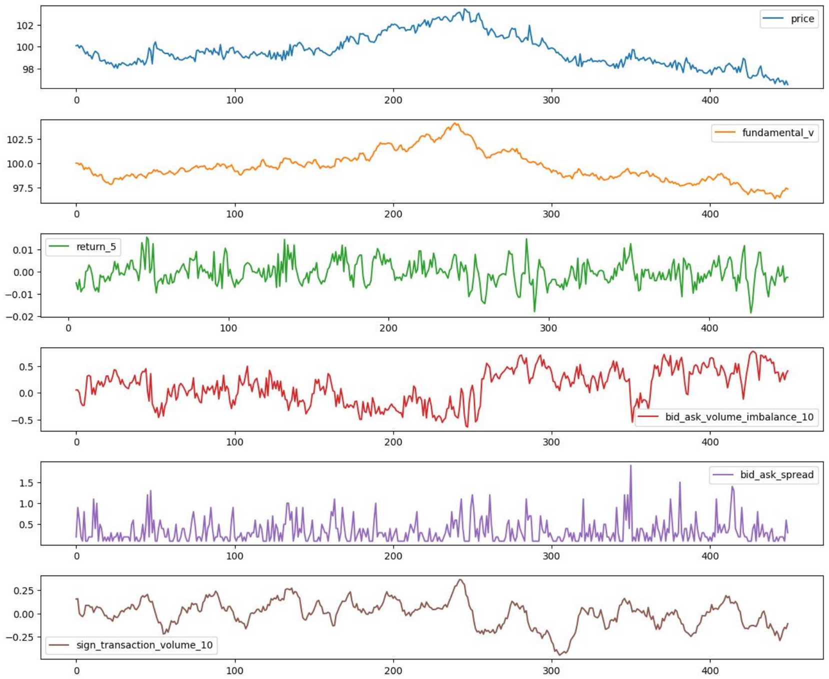
1. Agents are called in random order
   a. Called Agent observes current market information
   b. Called Agent places orders
   c. Order book is cleaned
2. Dividends are updated
3. The cycle repeats



Simulator setup settings:

Asset price - 200, risk free rate - 5%, dividend yield - 5%.

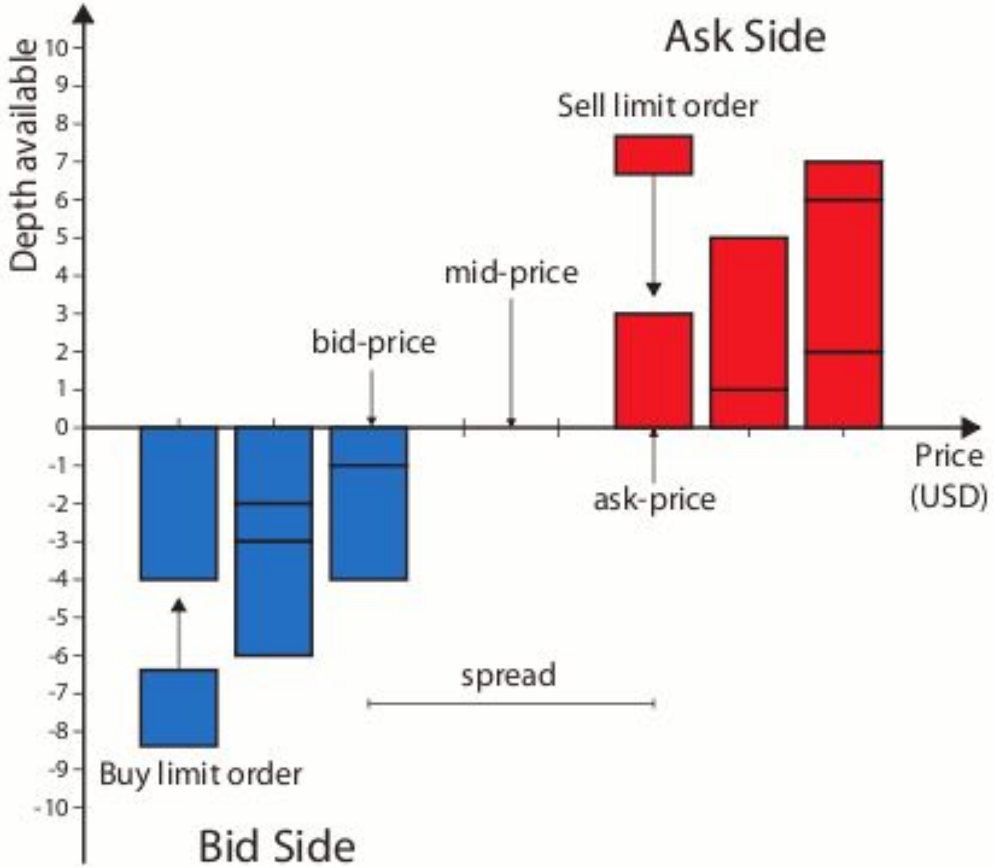
Chartists, Fundamentalists, Random traders - 40 each. 1 Market Maker if present.



Feature extraction

Feature generation

- Bid-ask volume imbalance
- Signed transaction volume
- Previous dividends
- Returns
- Fundamental value



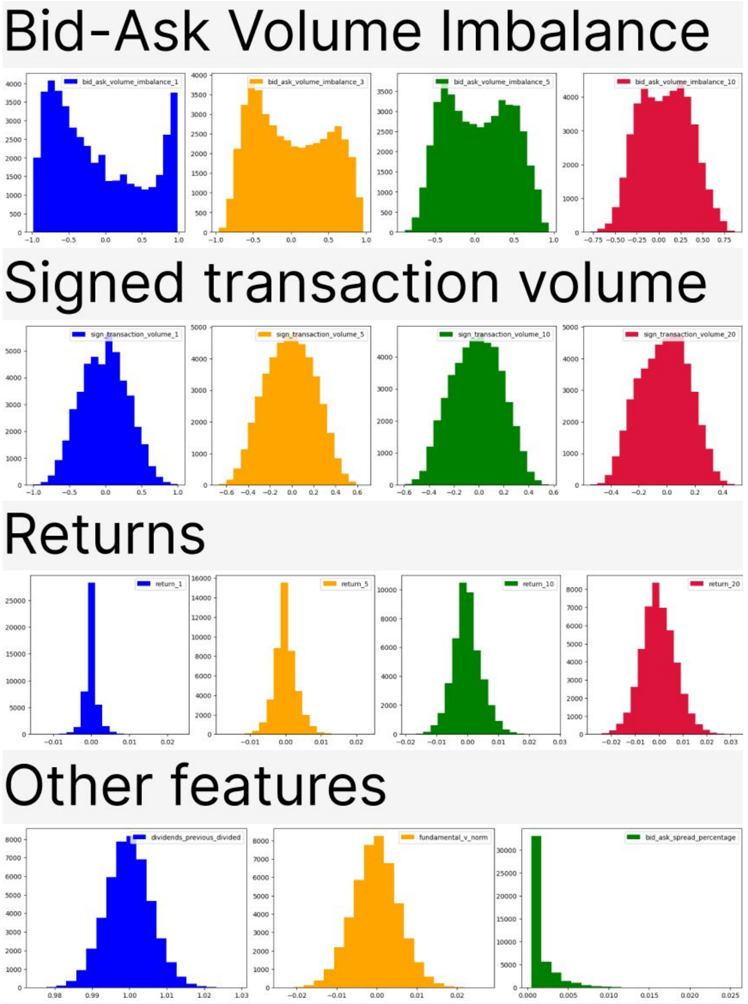
Feature generation

Lag or depth dependent features:

- Bid-Ask Volume Imbalance (1, 3, 5, 10 - depth)
- Signed transaction volume (1, 5, 10, 20 - lags)
- Returns (1, 5, 10, 20 - lags)

Other features:

- Last dividends difference from the risk free returns
- Fundamental price
- Bid-Ask spread



Classification accuracy

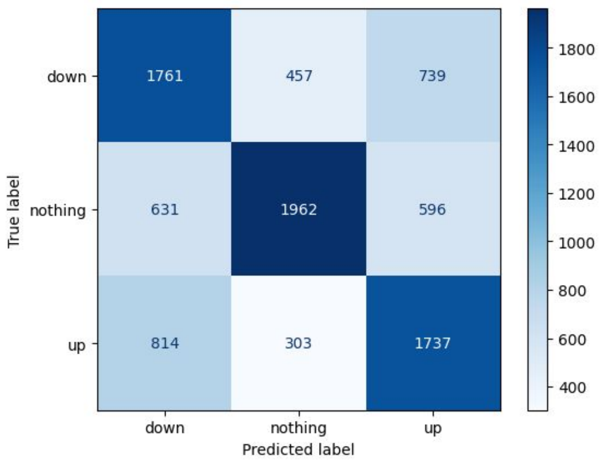
Log regression / XGBoost

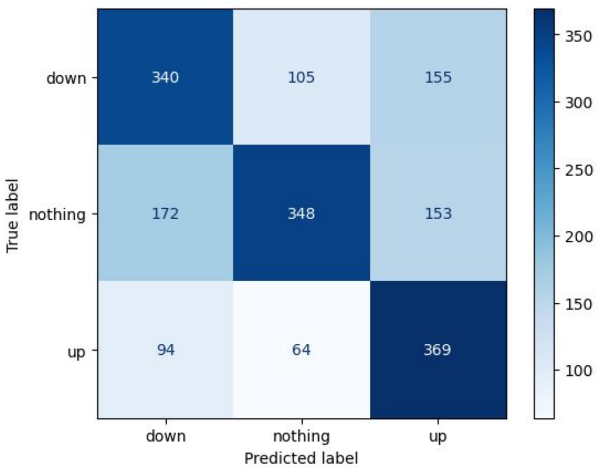
Classification Accuracy

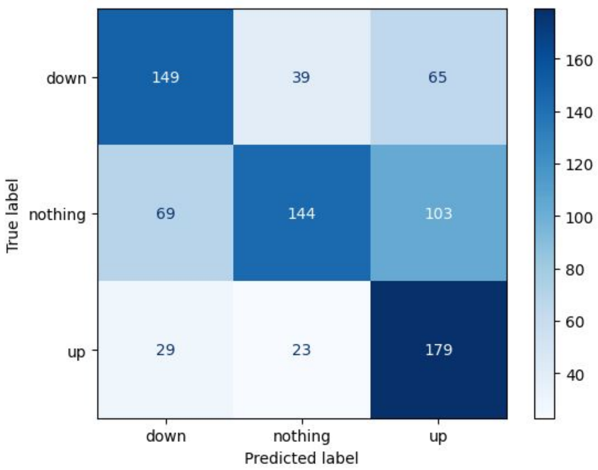
	Lag 1 (~40 000 observations)		Lag 5 (~8000 observations)		Lag 10 (~4000 observations)	
	Train	Test	Train	Test	Train	Test
XGBoost	0.63	0.61	0.67	0.59	0.73	0.59
Log regression	0.56	0.57	0.52	0.52	0.53	0.50

Confusion matrices of XGBoost

Lag 1 Test (~8000 observations)	Lag 5 Test (~1600 observations)	Lag 10 Test (~800 observations)







NN models

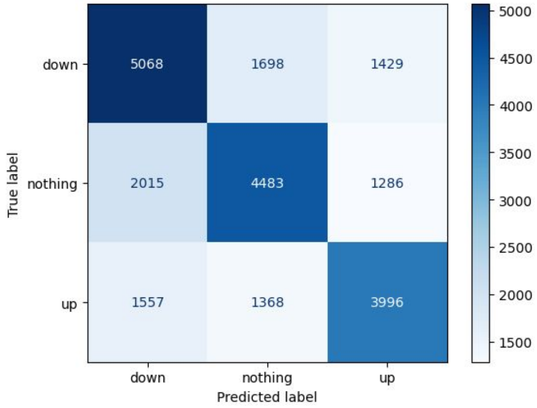
Classification Accuracy

Model	Dataset	Lag 1 (test)	Lag 5 (test)
Multilayer perceptron	Presence of Market Maker	**0.6**	**0.62**
Multilayer perceptron	Absence of Market Maker	**0.6**	**0.54**
LSTM	Presence of Market Maker	0.33	0.55
LSTM	Absence of Market Maker	0.33	0.5

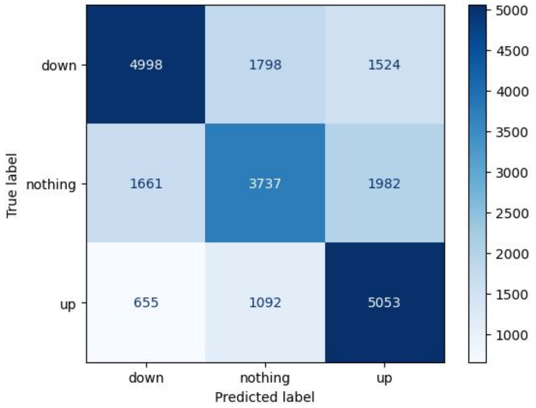
Confusion matrices of MLP

Dataset with Market Maker

Lag 1

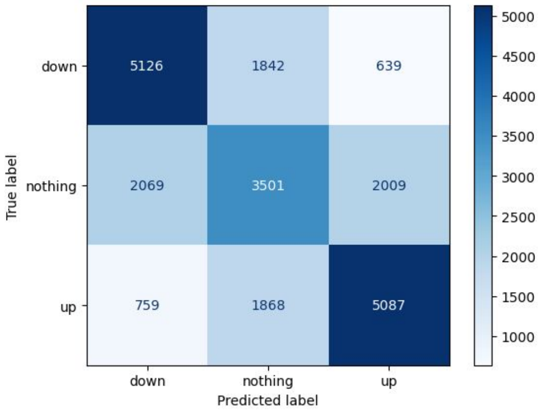


Lag 5

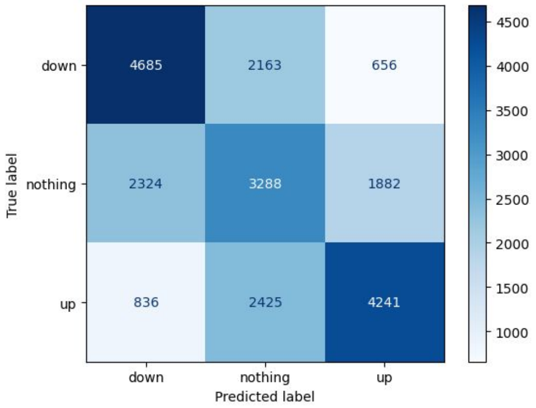


Dataset without Market Maker

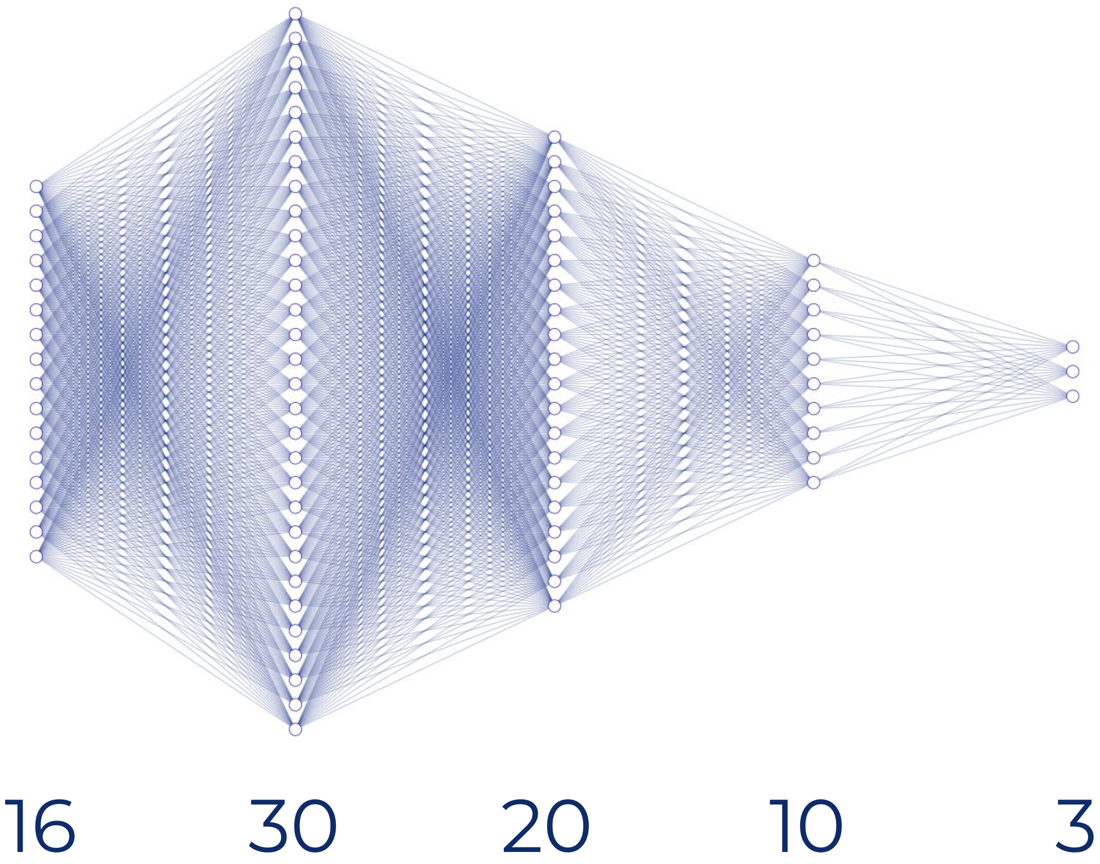
Lag 1



Lag 5



Classification MLP NN architecture



Reinforcement Learning models

Reinforcement Learning setup

State space

16 Features + Inventory

All features are normalized to be in range of [-1; 1]

Action Space

Limit orders (Q, P)

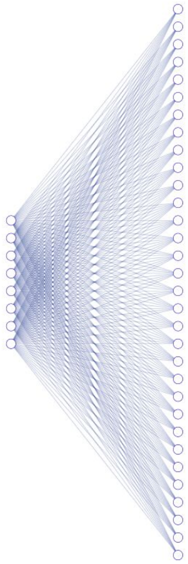
Q - quantity in order (0, 10,.. 90)
P - deviation from the best price [-5; 10]

Reward function

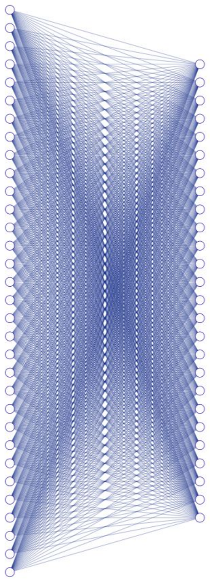
$$Reward = (MIN(Q\_executed_{bid}, Q\_executed_{ask}) * (P\_executed_{ask} - P\_executed_{bid}) / a) - |Assets| / Inventory\_Restriction$$

PPO - Proximal Policy Optimization (Default configuration)

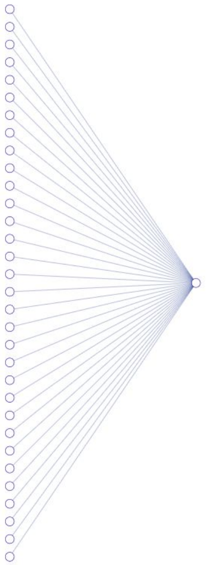
Feature extraction network



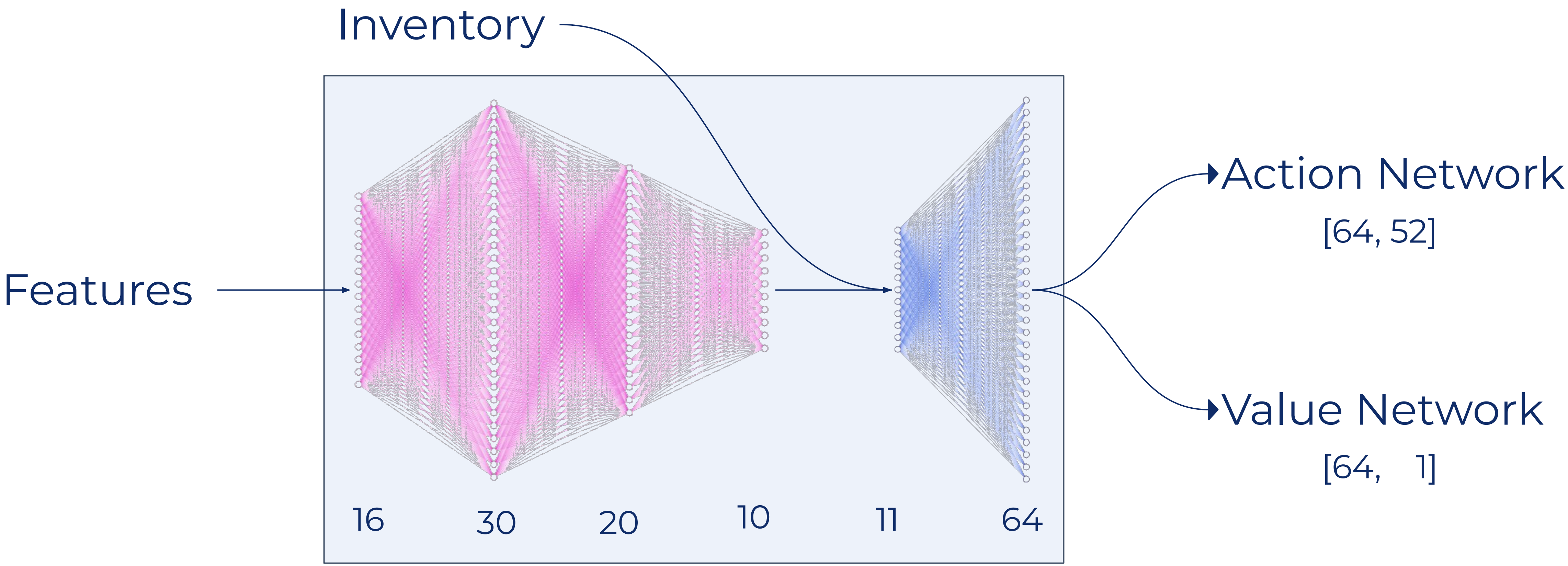
Action Network



Value Network



PPO - Transfer Learning architecture



Feature extraction network

PPO - Proximal Policy Optimization

Default PPO

Extractor Network:
[17, 64]

Action Network:
[64, 52]

Value Network:
[64, 1]

PPO with Transfer Learning (Lag 1)

Extractor Network:
Classification MLP (Lag 1)
[11, 64]

Action Network:
[64, 52]

Value Network:
[64, 1]

PPO with Transfer Learning (Lag 5)

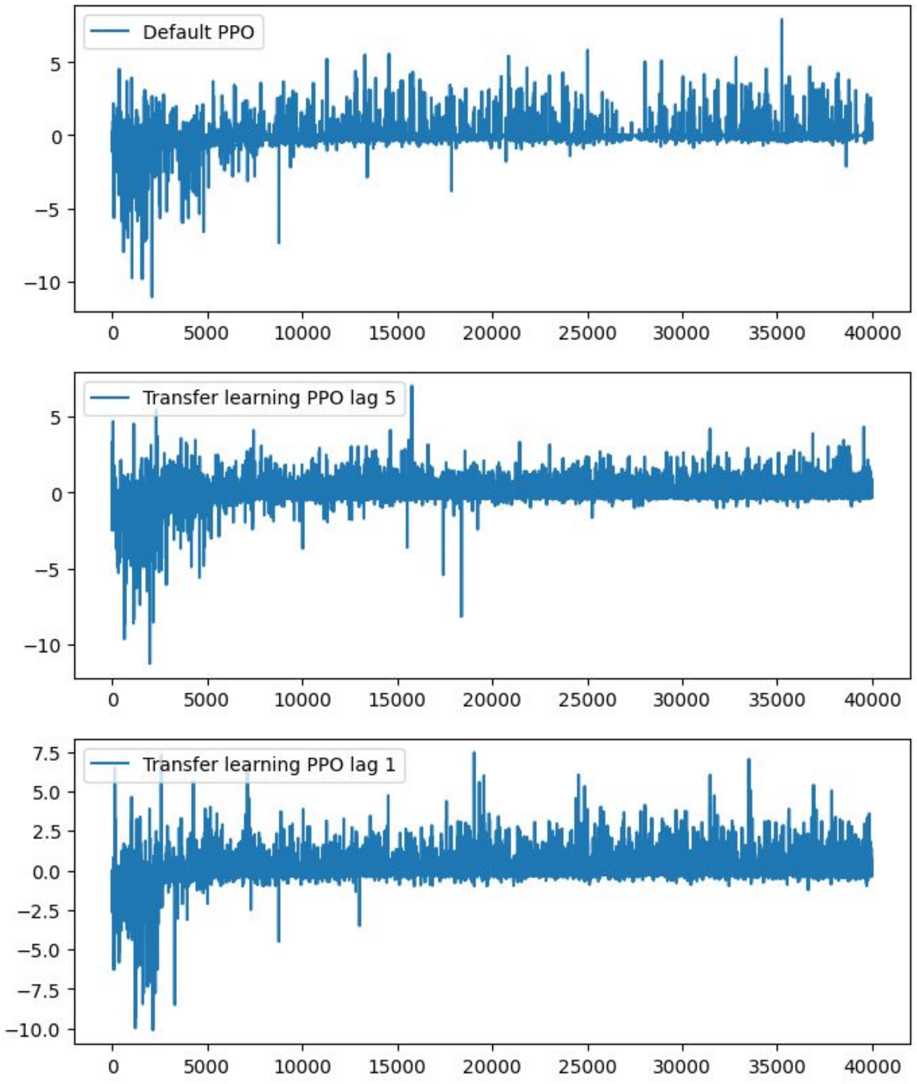
Extractor Network:
Classification MLP (Lag 5)
[11, 64]

Action Network:
[64, 52]

Value Network:
[64, 1]

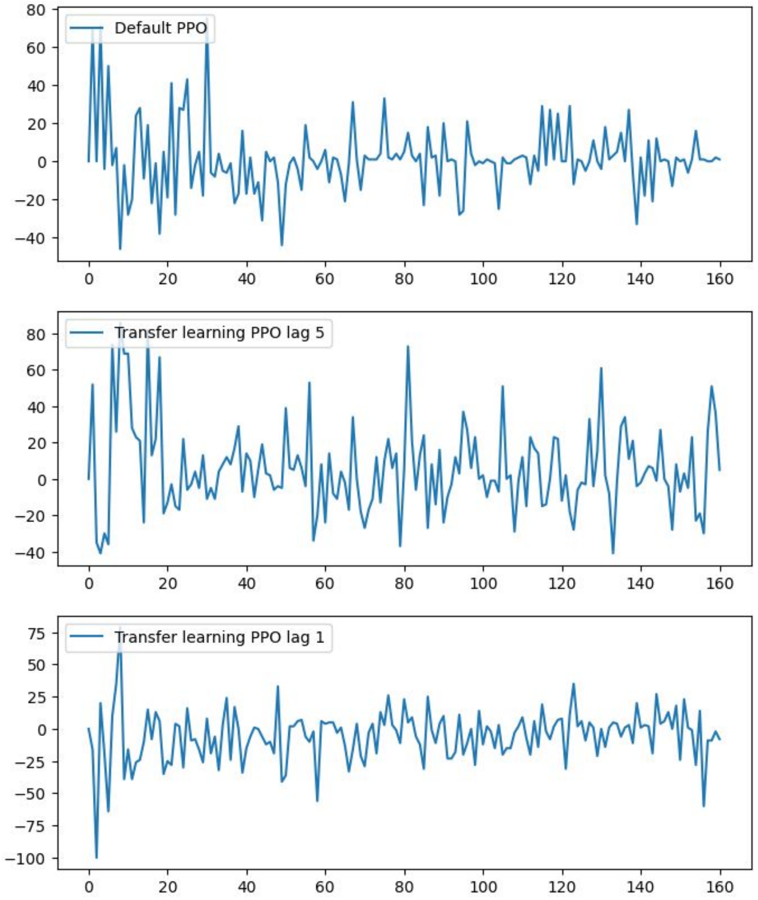
Results

Performance comparison by loss

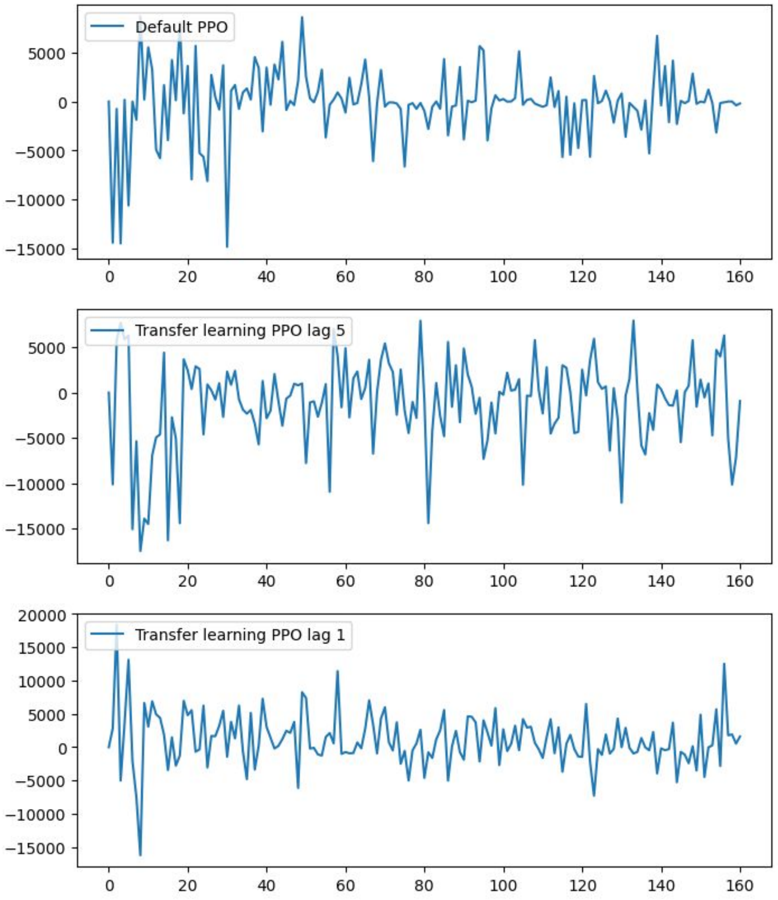


Performance comparison by assets and cash

Assets



Cash



Test setup

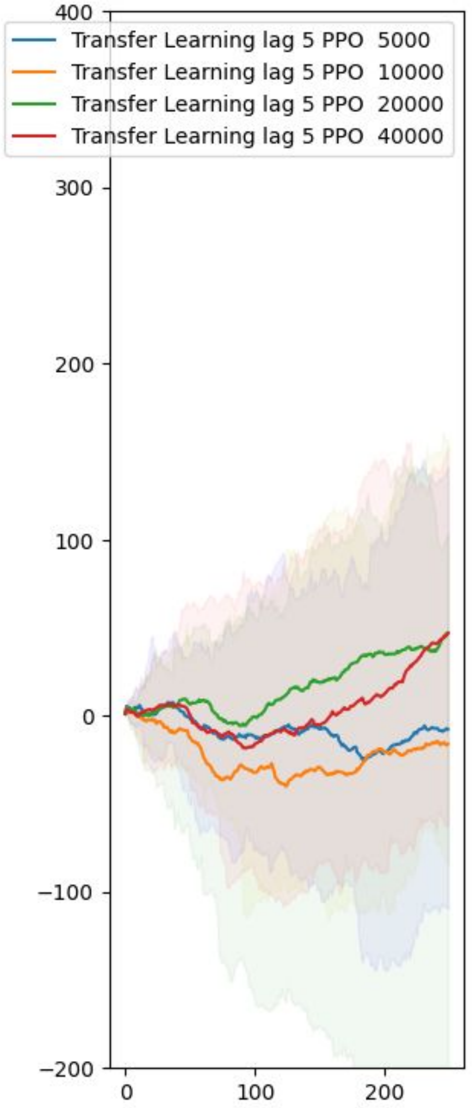
The models were trained for 40000 iterations.

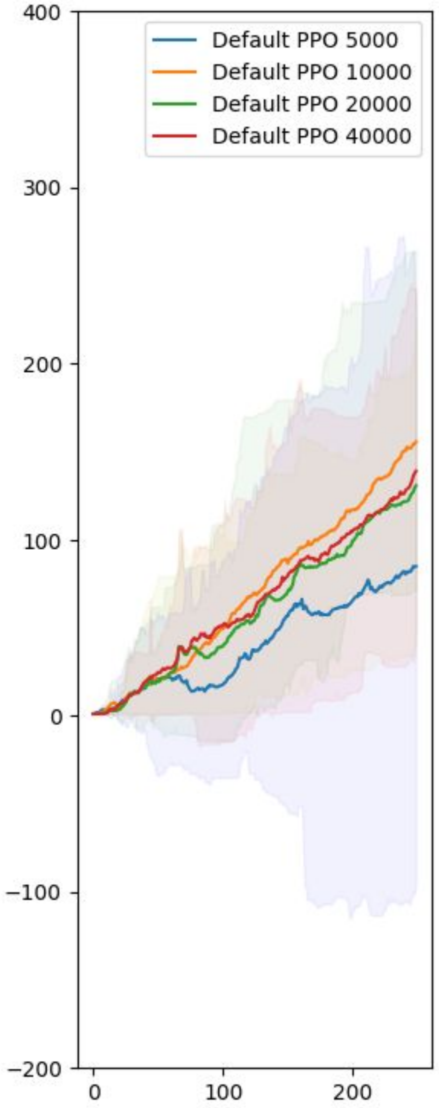
Model weights are saved and tested after 5000, 10000, 20000 and 40000 training iterations.

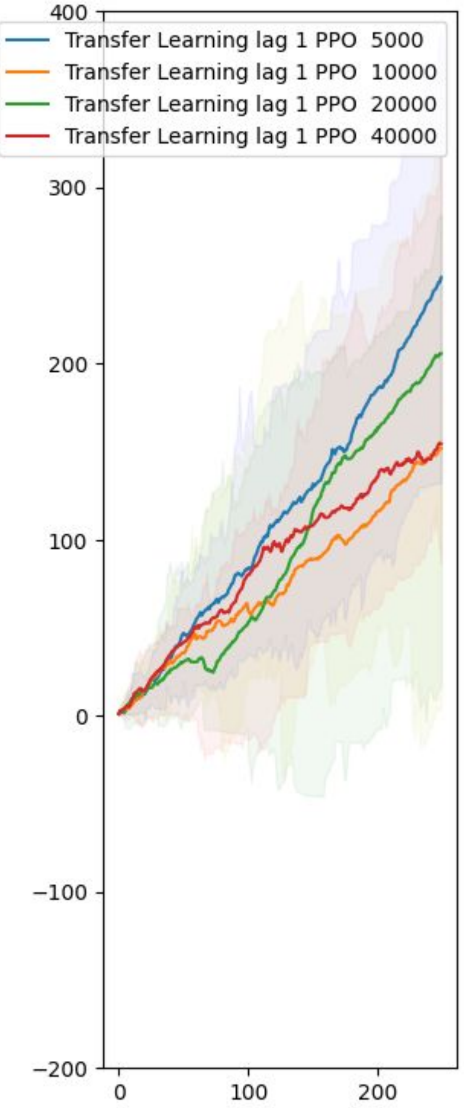
Each model is tested in 20 independent simulations, 250 iterations each.

Comparison is made by agent equity - agent cash + current value of the position

Performance comparison by equity







Performance comparison by equity

Model \ Training iterations	5000	10000	20000	40000
Default PPO	84.8 (134.2)	155.5 (82)	138.9 (108.6)	82 (91)
Transfer Learning lag 5	-7.6 (132.8)	-16 (145.9)	47 (106.6)	46.9 (102.6)
Transfer Learning lag 1	**248.8 (103)**	151.7 (118.8)	205.7 (105)	154.3 (121.1)

Conclusion

Research conclusion

In the research we studied effectiveness of Transfer Learning approach used for training Reinforcement Learning agent for solving optimal Market Making problem.

It is shown that for Agent Based market simulators, applying Transfer Learning resulted into the best performing model, which required the least amount of training.

Compared to Default PPO model, usage of **Transfer Learning** that utilizes MLP classification of 1 step price movement model resulted into **60% better performance** with **50% less training iterations**

Thanks for attention

HSE UNIVERSITY • EST 1992 •