Summer Research: Adversarial Machine Learning

under the guidance of Professor Daniel Bienstock

Yu Liu[1] Zi Zhuang[2]

[1]MSOR Student, yl4342

[2]MSOR Student, zz2693

October 2020

Content

- Adversarial examples in machine learning
- How to generate adversarial examples
- How to defend the attack
- Our work
- Introduce to the dataset and model
- The constraints
- Experiments and results
- Conclusion
- Future Direction
- Reference

Adversarial examples in machine learning

Definition

Adversarial examples are imperceptibly perturbed natural inputs that induce erroneous predictions in classifiers.

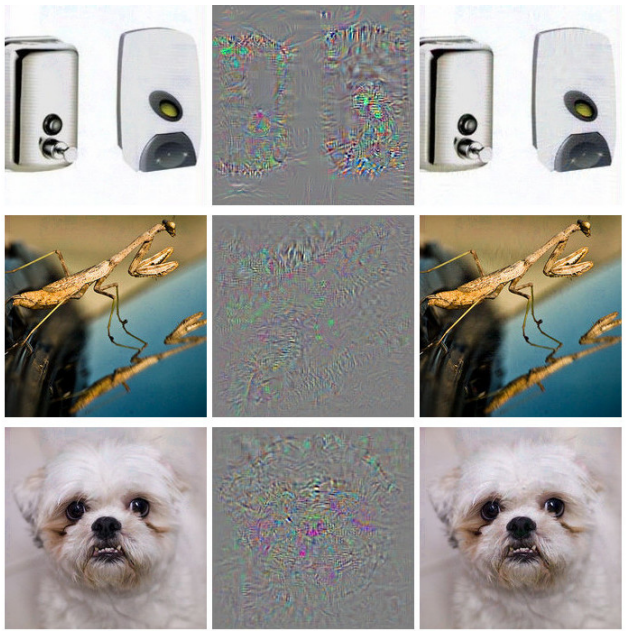


Figure: Adversarial examples[1]

[1] Adversarial examples for AlexNet by Szegedy et. al (2013). All images in the left column are correctly classified. The middle column shows the (magnified) error added to the images to produce the images in the right column all categorized (incorrectly) as "Ostrich".

How to generate adversarial examples

L-BFGS

Consider $f : [0,1]^m \to \{1...k\}$ is a classifier, the loss function is $loss_f : [0,1]^m \times \{1...k\} \to \mathbb{R}^+$. For a given input $x \in [0,1]^m$, and a wrong label l:

- Minimize $||r||_2$ s.t. $f(x+r) = l$, $x + r \in [0,1]^m$
- L-BFGS: Minimize $c||r||_2 + loss_f(x+r, l)$, s.t. $x + r \in [0,1]^m$

FGSM(Fast Gradient Sign Method)

Let x be the input, y be the corresponding label, and $J(x; y)$ be the cost function.

- The perturbation:$\eta = \epsilon sign(\nabla_x J(x, y))$

PGD

- $x^{t+1} = Proj_{x+S}(x^t + \epsilon \nabla_x J(x, y))$

How to defend the attack

- Training with the mixture of adversarial and clean examples
- Training with an adversarial objection function

$$\tilde{J}(x, y) = \alpha J(x, y) + (1 - \alpha) J(x + \epsilon sign(\nabla_x J(x, y)))$$

- Training with weight sparsity(add an l_1-norm regularization term)

Our work

Our goal

Training adversarial-robust neural network related to image is fully studied, we want to do the similar thing in the field of finance, especially around the High-Frequency Trading(HFT) data.

Method

- We choose to use PGD to generate adversarial examples and train a robust network on the mixture of adversarial and clean examples.
- Specially, We use a two-step alternative method, that is to alternate training net and generate adversarial examples on the new net we get.

Introduce to the dataset and model

Raw data

- LOB data of Ford Company provided by NYSE are downloaded from WRDS.
- Training set is 09/02/2019 – 09/27/2019. Validation and test are the following four weeks.
- Only contain the first hour since opening, i.e. 09:30 – 10:30.

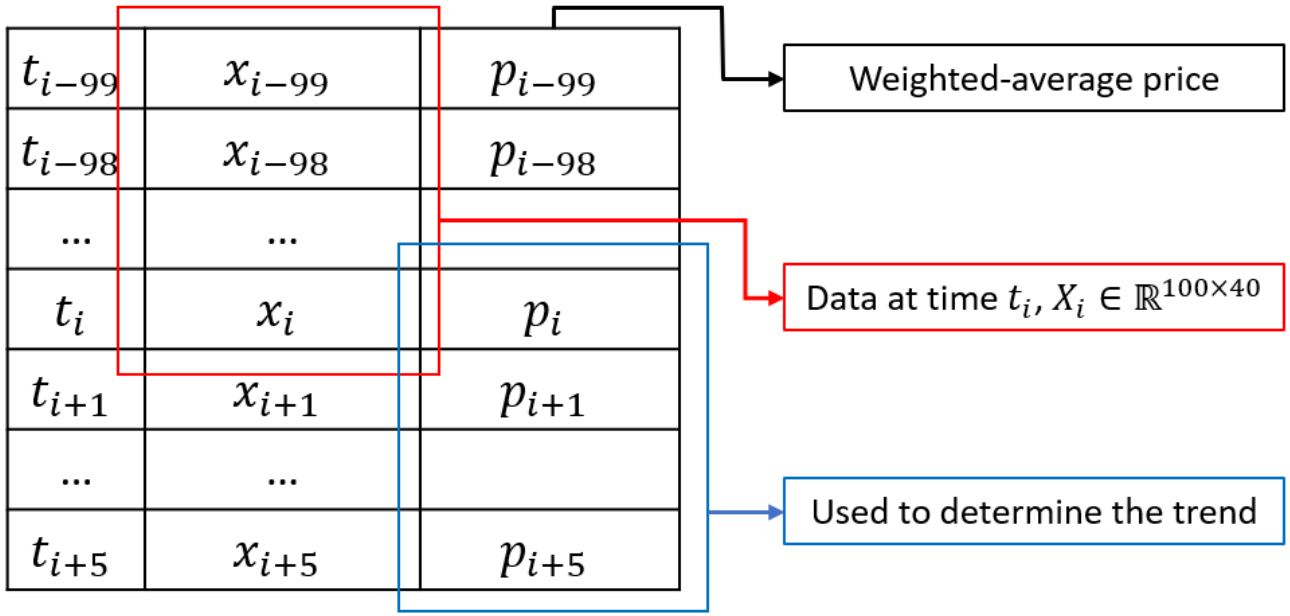
Preprocess

- Normalize - based on past week's mean and deviation.
- Sort - ask price in descending order and bid price in ascending order.
- Reformat - 40 features, i.e. $X_i = \{P^i_{ask}, S^i_{ask}, P^i_{bid}, S^i_{bid}\}_{i=1,\ldots,10}$

Introduce to the dataset and model

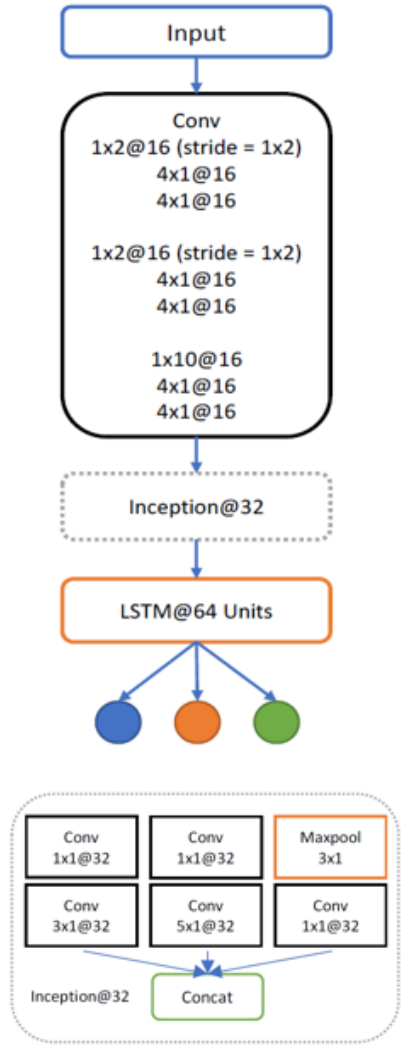
Goal of the Model

- Given the 100 consecutive data, we want to predict whether the weighted-average price is rising, falling or unchanged in the next 5 data.
- We use a specific threshold to determine this, which is chosen to average the number of data with each label.



Introduce to the dataset and model

- Convolution layer with filter size(1*2) and stride(1*2), which in fact is micro-price.
- Filters size (4*1) capture interactions over time step.
- Inception are considered as different moving average in technical analysis.
- No pooling except in Inception: causing under-fitting in time-series data.
- LSTM could capture the temporal behavior.



The constraints

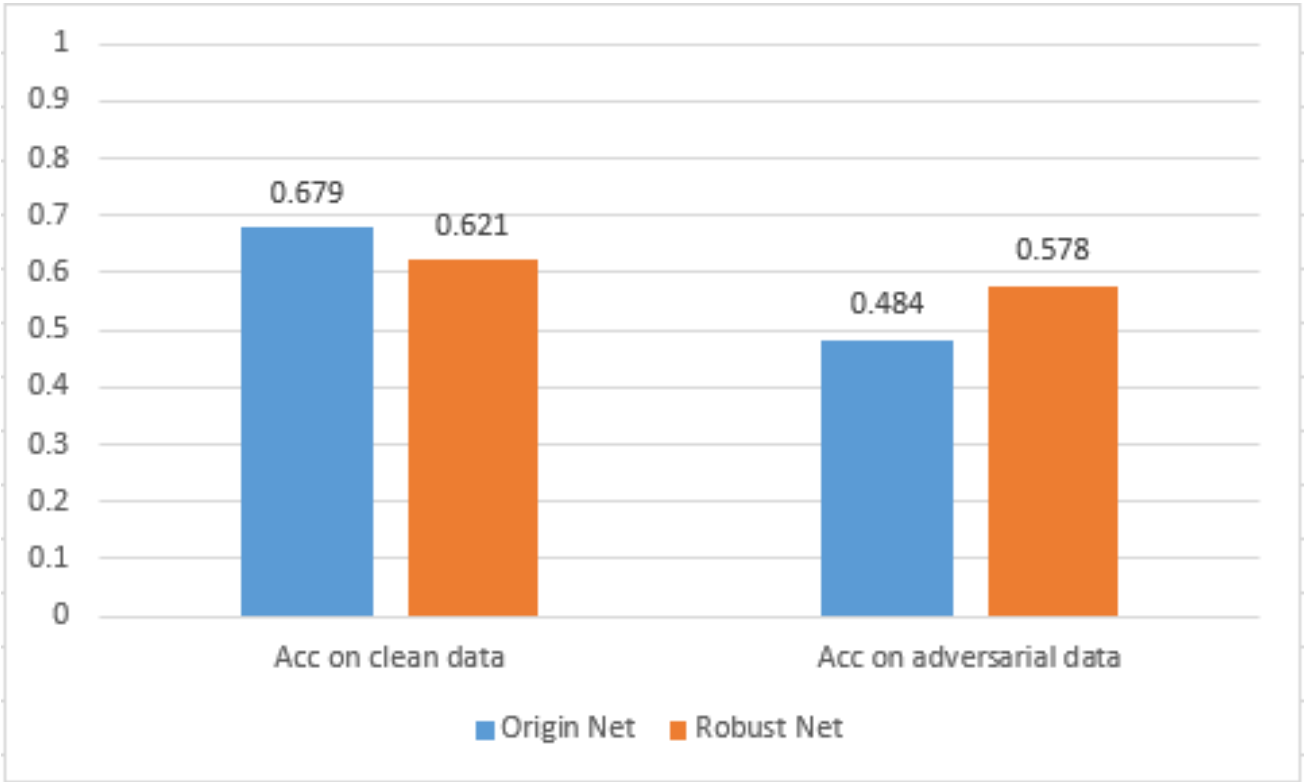
- Attack raw size claimed by traders but not price.
- Not L-2 or L-infinity used in image attacker.
- Capital required: Within some budget ($10,000), aim to fool algorithms by perturbing the bid or ask size.
- Relative size: the number of shares of the adversary's propagated perturbation as a percentage of total size on the book.

Experiments and results

Experiment

We use the clean data to train the origin net, and use the net to get the adversarial data. Then we use two-step method to train the robust net.[a]

[a] Code is here:https://github.com/YuLiuCU/summer-research



Conclusion

- We show by the experiment that adversarial examples will do huge influence on the accuracy of a neural network. They will lower the accuracy to under 50%, which means maybe we will not get a positive profit in expectation.
- By using two-step training method, some accuracy is sacrificed in exchange for stability. We think it is acceptable.

Future Direction

- We only consider the trend of the price, but not the P&L of the operator. We are going to find some trading strategies and do adversarial attack on it, then see what happened to the profit.
- The attack maybe from someone so we can compare the budget to conduct the attack and the loss of the one been attacked.

Reference

Micah Goldblum, Avi Schwarzschild, Naftali Cohen, Tucker Balch, Ankit B Patel, and Tom Goldstein.
Adversarial attacks on machine learning systems for high-frequency trading.
arXiv preprint arXiv:2002.09565, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy.
Explaining and harnessing adversarial examples.
arXiv preprint arXiv:1412.6572, 2014.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry.
Adversarial examples are not bugs, they are features.
In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu.
Towards deep learning models resistant to adversarial attacks.
arXiv preprint arXiv:1706.06083, 2017.

Adamantios Ntakaris, Martin Magris, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis.
Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods.
Journal of Forecasting, 37(8):852–866, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.
Intriguing properties of neural networks.
arXiv preprint arXiv:1312.6199, 2013.

Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry.
Training for faster adversarial robustness verification via inducing relu stability.
arXiv preprint arXiv:1809.03008, 2018.

Zihao Zhang, Stefan Zohren, and Stephen Roberts.
Deeplob: Deep convolutional neural networks for limit order books.
IEEE Transactions on Signal Processing, 67(11):3001–3012, 2019.