Coding Projects for Deep Learning in Asset Pricing

Henry Schellhorn

November 21, 2022

1 Introduction

All coding projects are about additions to/reformulation of the "Pelger machine" (see references [1] and [2]) in Matlab. There are 4 possible coding projects[1]:

1. Extension of GAN loop and verification of convergence
2. Integration of LSTM with GAN
3. Adding a positivity constraint for the SDF
4. Reformulation of GAN as a linear program.

The original goal of the Pelger machine is to minimize (over $\boldsymbol{\theta}_\omega$) and maximize (over $\boldsymbol{\theta}_g$) the mispricing error $S(\boldsymbol{\theta}_\omega, \boldsymbol{\theta}_g)$:

$$\min_{\boldsymbol{\theta}_\omega} \max_{\boldsymbol{\theta}_g} S(\boldsymbol{\theta}_\omega, \boldsymbol{\theta}_g)$$

1.1 Existing Code

I already coded both feedforward networks (SDF network and the Conditional Moment Network,) but did not code the state RNN, nor the moment RNN: see program GAN20211107Homework10, reference [3].

For pedagogical purposes, I took out some instructions from step 5 [2]of GAN20211107Homework10.m.. As part of homework assignment 8 (due November 10) all students in the class will need to write the formulae for step 5 and will have to code it as part of homework assignment 10 (due December 8). They will see that a straightforward implementation of step 5 does not satisfy the requirements: indeed, the cost function does not increase as it should as a function. This inspired projects 1 and 4.

[1]Unlike stated in the syllabus, I did not have the time to develop a project on "integration of moment conditions with GAN"

[2]Step 5 is the stochastic gradient algorithm for the conditional moment network.

2 Description of the Projects

Project 1: extension of GAN loop and verification of convergence

Students will analyze why a naive implementation of step 5 of the algorithm does not satisfy the requirements: in other terms, for fixed $\boldsymbol{\theta}_\omega$ step 5 should result in increasing $S(\boldsymbol{\theta}_\omega, \boldsymbol{\theta}_g)$ by modifying the parameter vector $\boldsymbol{\theta}_g$. They will implement different possible solutions, such as modifying the learning rate, or changing the initial condition.

Project 2: integration of LSTM with GAN

Students will code the state RNN and integrate it in GAN20211107Homework10.m. For simplicity, we will choose as recurrent neural network the AR(1) model, described in chapter 17 of my notes.

We have two pieces of information $x_{t,1}$ and $x_{t,2,i}$, say market return and weather for firm i ($i = 1, .., N$). We suppose for both of them an AR(1) model

$$\begin{aligned}
x_{t+1} &= c_1 + \phi_1 x_t + s_1 \varepsilon_t \qquad t > 1 \\
x_{t+1,2,i} &= c_2 + \phi_2 x_{t,2,i} + s_2 \varepsilon_{t,2,i} \qquad t > 1 \\
x_1 &= \varepsilon_1 \\
x_{1,2,i} &= \varepsilon_{1,2,i}
\end{aligned}$$

Note: we could also model all the second parameters (s_2,c_2 and ϕ_2) to be firm-specific, as well as introducing correlations. I just wanted to limit the number of parameters. Note that $\{\varepsilon_t, \varepsilon_{t,2,i}\}$ are assumed IID. Taking $N = 5$, The parameter vector is

$$\boldsymbol{\theta}_m = \begin{bmatrix} s_1 \\ s_2 \\ s_{2,2} \\ s_{2,3} \\ s_{2,4} \\ s_{2,5} \\ \phi_1 \\ \phi_2 \\ c_1 \\ c_2 \end{bmatrix}$$

The current minimization algorithm is a network that implements:

$$\omega(x, \theta_\omega) = \sigma(\mathbf{w}_\omega^{(3)} \boldsymbol{\sigma}(\mathbf{W}_\omega^{(2)} \mathbf{x} + \mathbf{b}_\omega^{(2)}) + b_\omega^{(3)})$$

Adding the RNN, you should make it implement

$$\omega(x, \theta_\omega, \boldsymbol{\theta}_m) = \sigma(\mathbf{w}_\omega^{(3)} \boldsymbol{\sigma}(\mathbf{W}_\omega^{(2)} \boldsymbol{\varepsilon} + \mathbf{b}_\omega^{(2)}) + b_\omega^{(3)})$$

where

$$\boldsymbol{\varepsilon}_t = \begin{bmatrix} \varepsilon_{t,1} \\ \varepsilon_{t,2} \end{bmatrix} = \begin{bmatrix} \frac{x_{t+1,1}-\phi_1 x_{t,1}-c_1}{s_1} \\ \frac{x_{t+1,2}-\phi_2 x_{t,2}-c_2}{s_2} \end{bmatrix}$$

For simplicity, students may want to integrate this code only to the steps 1 to 4 of the algorithm, and omit step 5.

Project 3: adding a positivity constraint to the SDF

We will add a regularizing term. The original goal of the Pelger machine is to minimize (over $\boldsymbol{\theta}_\omega$) and maximize (over $\boldsymbol{\theta}_g$) the mispricing error $S(\boldsymbol{\theta}_\omega, \boldsymbol{\theta}_g)$. To achieve positivity of the SDF m, we subtract a regularizing term $\frac{\lambda}{T}\sum\limits_{t=1}^{T} m(t;\boldsymbol{\theta}_\omega)$, with $\lambda > 0$ so that the goal is:

$$\min_{\boldsymbol{\theta}_\omega} \max_{\boldsymbol{\theta}_g} S(\boldsymbol{\theta}_\omega, \boldsymbol{\theta}_g) - \frac{\lambda}{T}\sum_{t=1}^{T} m(t;\boldsymbol{\theta}_\omega)$$

where we write:

$$m(t, \boldsymbol{\theta}_\omega) = 1 - \sum_{i=1}^{N} \omega(\mathbf{x}_{t,i}; \boldsymbol{\theta}_\omega^k) R_i^{e,t+1}$$

For simplicity, students may want to implement this code only to the steps 1 to 4 of the algorithm, and omit step 5.

Project 4: Reformulation of GAN as a linear program

It may be the case that changing the learning rate does not solve the problem of lack of convergence of the moment part of the deep neural network. Another route is to replace L_2 minimization by L_1 minimization, that is, replace $S(\boldsymbol{\theta}_\omega; \boldsymbol{\theta}_g)$ by an appropriate linear function, call it $S^{lin}(\boldsymbol{\theta}_\omega; \boldsymbol{\theta}_g)$. Then the students will inplement a linear program that $\min_{\boldsymbol{\theta}_\omega} S^{lin}(\boldsymbol{\theta}_\omega; \boldsymbol{\theta}_g)$ using the data in GAN20211107Homework10.m. Possible problems include unboundedness of the parameters, so that linear constraints might need to be added.

3 Schedule

We will meet every Thursday from November 3rd to December 1st at Burke 222. It is important to all meet together, so we will find a time to accommodate every team working on the project (only one representative per project is needed).

Date	Action	Deliverable
Oct. 27	Read chapters 17,18,19 (class notes)	Tell me which project you choose
Nov. 3	Meeting	
Nov. 10	Meeting	Code and run step 5 of the algorithm[3]
Nov 17	Meeting	
Dec 1	Meeting	Project code running
Dec. 8		Student presentation

Important: students who do not meet the November 10 deliverable will not be allowed to continue on the practical option, and will have to choose the theoretical option.

4 References

1. Chen, L., Pelger, M., and J. Zhu (2019). Deep Learning in Asset Pricing . https://arxiv.org/abs/1904.00745.

2. My class notes, Machine Learning for Asset Pricing, chapters 18 and 19.

3. Matlab program GAN20211107Homework10.m, on Canvas.

[3] i.e., do problem 4, homework 10 in advance. Of course, do not share your results with others.