Learning to Trade III
Deep Hedging with Impact
Deep Bellman Hedging
Open Topics

TU Munich
Visiting Prof. Hans Buehler
Prof. Blanka Horvath

TU München 2022, Revision 1.1

http://deep-hedging.com

Recap: Vanilla Deep Hedging

AI for Derivatives Trading

Recap

Notation

- Everything is in discrete time → markets not complete.
- We call s_t the state of the market. It represents all known information.
- That means that any observable random variable R_t can be written as $R(s_t)$.
- At any point t we observe the model prices $H_t^i = H^i\,(s_t)$ of n hedging instruments.
- Any cashflows etc are aggregated. That means if t is beyond the expiry of the instrument, then H_t^i represents the sum of all its cashflows
- The (sum of the) model prices of our existing portfolio is $Z_t \equiv Z(s_t)$.
- The market is observed under the statistical measure P.

Recap

Trading cost

- Trading $a \in \mathrm{R}^n$ units of H at t will cost $c_t(a)$ in excess of the model price H_t.
- The function c is normalized to $c_t(0) = 0$, non-negative, and convex.
- Convexity excludes fixed trading cost.
- Trading is limited to where $c < \infty$.

Vanilla Deep Hedging

- Assume T is such that all instruments in our portfolio Z and any relevant hedging instruments are expired.
- Valuation of Z_T and H_T is trivial, and *model-independent*: it is the sum of all cashflows until T.
- We will not solve for an **policy** a which is a function of the state, i.e. at any time t the hedging action is $a_t \equiv a(s_t)$.
- Total **gains** are

$$G^a := Z_T - Z_t + \sum_{r=t}^{T-1} a_r \cdot (H_T - H_r) - c_r(a_r)$$

Vanilla Deep Hedging

- We call a **monotone**, **concave**, and **cash-invariant** functional U_t which is normalized to $U_t(0) = 0$ a conditional **monetary utility.**
  - Then $-U_t(X)$ is a normalized conditional **convex risk measure**.
- Let u be C^1, monotone, and concave and normalized to $u(0) = 0, u'(0) = 1$. Its *Optimised Certainty Equivalent (OCE)*

$$U_t(X) \coloneqq \sup_{y_t}: E_t[u(X + y_t) - y_t]$$

is a monetary utility.

Vanilla Deep Hedging

- Let U be a **monetary utility**. Then the **Vanilla Deep Hedging** problem [1] is given as

$$\max_{a=a_t,\dots,a_{T-1}} : U_t\left(Z_T - Z_t + \sum_{r=t}^{T-1} a_r \cdot (H_T - H_r) - c_r(a_r) \right)$$

- Natural machine learning approach: write a as **neural network**.
- Called *periodic policy search* in reinforcement learning [2]

---

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf
[2 Reinforcement Learning, Sutton and Barto, 2018

3.1 Deep Hedging with Impact

Deep Hedging for Optimal Order Scheduling
Joint work with Richard Gramblicka (JPM and ETH)

Optimal Order Scheduling

- We now use our framework for the case where $t = 0$ represent the market open, T is the close, and where τ_r are **intraday** hedging decision points. We formally set $\tau_0 := 0, \tau_T := 1$ as we will operate in business time.
- Our task is to “delta”-hedge a given portfolio Z.
- Trading will incur market impact

Optimal Order Scheduling

Market Impact

- Market impact means that
  - if we buy an asset, we move the price up (we pay more)
  - if we sell, we move the price down (we earn less)
  - impact decays over time.
- Trading too fast in short periods creates more impact than trading slower over longer periods.
  - If we trade over a period then we have to strike a balance between *certainty of execution* and *implementation risk* (price may move against us)
- Several good papers on the topic, most notably [1] and [2].

---

[1] Optimal Liquidation, Almgren, R. and Chriss, N. (2000) https://papers.ssrn.com/sol3/papers.cfm?abstract_id=53501
[2] Dynamical Models of Market Impact and Algorithms for Order Execution, Gatheral and Schied https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2034178

Optimal Order Scheduling

- Gatheral-Schied model for market impact [1] (usually not in the context of derivatives)
- Let $\bar{S}$ be the asset dynamics if we do not trade.
  Then the continuous time stock price dynamics given a trading policy a are given as

$$S_t := \bar{S}_t + \int_0^t K'^{(t-s)} h(a_s) ds \approx \bar{S}_t e^{\int_0^t K'^{(t-s)} \frac{1}{S_0} h(a_s) ds}$$

- Here h is the impact function and K' is a decay kernel.
- Note that h is of order S_0.
- Impact cannot be arbitrary to avoid "roundtrip" statistical arbitrage [1]
  - If we want to buy we first *sell* fast, and then buy back slow.
  - Be aware of appropriate market abuse regulation.

[1] Dynamical Models of Market Impact and Algorithms for Order Execution, Gatheral and Schied https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2034178

Optimal Order Scheduling

- Gatheral-Schied model for market impact [1]:

$$S_t := \bar{S}_t + \int_0^t K'(t-s)h(a_s)ds$$

- A few examples:
  - $h(x) = cx$ and $K'(\tau) = 1$ for "permanent" impact.
  - $h(x) = S_0 \operatorname{sign}(x)\, c|x|^\delta$ and $K'(\tau) = \tau^{-\gamma}$ for $\gamma + \delta \geq 1$ c.f. [1].
    Empirical values are $\delta \approx 0.5$ and $\gamma \approx 0.5$ ("square root rule"), c.f. [1], [2]
- If $K'(\tau) = e^{-\rho\tau}$ then h must be linear, c.f. [1]
- Asymptotically exponential kernel in [3]: $h(x) = S_0 \operatorname{sign}(x)\, c|x|^\delta$ and $K'(\tau) := \delta\, \rho \frac{e^{-\rho\tau}}{(1-e^{-\rho\tau})^{1-\delta}}$

—
[1] Dynamical Models of Market Impact and Algorithms for Order Execution, Gatheral and Schied https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2034178
[2] Presentation http://faculty.baruch.cuny.edu/jgatheral/Buzios2009.pdf
[3] Exponential Resilience and Decay of Market Impact, Schied, Gatheral, Slynko https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1650937

Optimal Order Scheduling

- Gatheral-Schied model for market impact [1] – discrete version: a_r now represents the number of shares to buy in $[\tau_r, \tau_{r+1})$:

$$S_r \coloneqq \bar{S}_r + \sum_{e=0}^{r-1} h(a_e) \int_{\tau_e}^{\tau_r} K'(\tau_r - s)ds \equiv \bar{S}_r + \underbrace{\sum_{e=0}^{r-1} h(a_r)K(e,r)}_{=:I_r}$$

- Given a historic intraday time series of spot prices $\bar{S}$ this is easy to compute.
- Most importantly, we can pre-compute $K(e,r)$ outside our training loop

[1] Dynamical Models of Market Impact and Algorithms for Order Execution, Gatheral and Schied https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2034178

Optimal Order Scheduling

- Total cost of trading VWAP over $[\tau_r, \tau_{r+1})$ is $a_r P_r$ where P_r is given as the average price

$$P_r := \frac{1}{\tau_{r+1} - \tau_r} \int_{\tau_r}^{\tau_{r+1}} S_t \, dt = \bar{P}_r + \frac{1}{\tau_{r+1} - \tau_r} \int_{\tau_r}^{\tau_{r+1}} \int_0^t K'(t-s) h(a_s) ds \, dt$$

  - $\bar{P}$ denotes the historic average price from real data.
  - The expression on the right can also be pre-computed in closed form for a given h, K'.
- To simplify calculations we might simply set

$$P_r \approx \bar{P}_r + \frac{I_{r+1} - I_r}{2}$$

Deep Hedging with Impact

- Total **gains** of trading a policy a:

$$G^a := Z_T(S_T) - Z_0 + \sum_{t=0}^{T-1} a_t(S_T - P_t) - \varsigma |a_t| P_t$$

  where ς represents and additional half spread cost.

- We wrote $Z_T(S_T)$ to stress that our derivative model value is computed / interpolated using the impacted spot price.
- Fits perfectly into our Deep Hedging framework – this time with *market impact* – by solving

$$\max_{a_0,\dots,a_{T-1}} : U(\,G^a\,)$$

Deep Hedging with Impact

- Practical Implementation of our program:

$$\max_{a_0,\dots,a_{T-1}} : U(\, G^a \,) \qquad G^a := Z_T(S_T) - Z_0 + \sum_{t=0}^{T-1} a_t(S_T - P_t) - \varsigma |a_t| P_t$$

- Use actual historic intraday market data and portfolio fair values.
- Interpolate $Z_T(S_T)$ using classic greeks or similar.
- Might want to add delta limits and other risk limits.
- Express everything relative to expected volume time.
- Features for the a network include the typical electronic scheduling features such as volume prediction models, order book imbalance, market direction predictors, c.f. [1]

---

[1] Algorithmic Trading and Quantitative Strategies, Hardy and Nehren https://www.routledge.com/Algorithmic-Trading-and-Quantitative-Strategies/Velu-Hardy-Nehren/p/book/9781498737166

Deep Hedging with Impact

- Properties assuming hedging instrument is just stock:

$$\max_{a_0,\dots,a_{T-1}} : U(\,G^a\,) \quad G^a := Z_T(S_T) - Z_0 + \sum_{t=0}^{T-1} a_t(S_T - P_t) - \varsigma|a_t|\,P_t$$

- Will attempt an optimal delta-hedge for the portfolio vs market impact.
- High risk aversion will lead to full delta-hedge, while low risk aversion will lead to maximizing returns → ensure absence of statistical arbitrage.
- If Z is a stock position, and if risk aversion is high → classic optimal liquidation problem.

Deep Hedging with Impact

- Experiments with only impact show
  - Recover classic analytic results on delta-hedging options with proportional and fixed cost
- *Ensure you adhere to applicable regulation.*
  - If Gatheral's statistical arbitrage condition [1] is violated, then the model will attempt to make money buy first trading fast in the opposite direction
  - If the model is able to push the stock price, it might be driven to do that in order to change the value of the derivative
    Example:
    - Stock is 100$
    - Hedging a barrier which will pay us 1bn$ if the stock reaches 100.025$
    - Model is incentivized to buy the stock fast in order to generate sizeable impact which increases the probability to trigger the barrier.

[1] Dynamical Models of Market Impact and Algorithms for Order Execution, Gatheral and Schied https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2034178

Deep Hedging with Impact

- Good
  - First model for hedging derivatives under impact
  - Nice example of Deep Hedging with state-dependent market dynamics
  - Impact modelling shows theoretical bounds shown by Gatheral-Schied
  - Shown to converge in known scenarios

- But
  - As written, problems needs to be solved whenever portfolio changes (like Deep Hedging) → potentially too slow for practical use.
  - However, model can be pre-trained on a wide number of past portfolios e.g. using historic positions.
  - Requires parsimonious representation of portfolio risk, e.g. via interpolation.

3.2 Deep Bellman Hedging

Universal AI for Deep Hedging
Joint work with Phillip Murray (JPM, Imperial) and Ben Wood (JPM)

Deep Bellman Hedging

- Deep Hedging needs to be solved for every initial portfolio Z and initial market state.
  - A kind of American Monte Carlo scheme
  - Most AI models aim to train once (possibly taking much longer) → once the model is trained it can be used for any initial portfolio and market state
  - Commonly referred to as "dynamic programming" or "Bellman" approach [1]

[1] Deep Bellman Hedging, Buehler et al https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Deep Bellman Hedging

We now rewrite this notation as follows:

- Define $\delta_r := \sum_{u=t}^{r} a_r$ as the position in H in r and set
- $DZ_{r:T} := Z_T - Z_r + \delta_r(H_T - H_r)$ represents the future value of our portfolio plus any existing position in our hedges as they are also “in our portfolio”
  - $DZ_{t:T} = DZ_{t:r} + DZ_{r:T}$ with clear past and future separation.
  - If r is past all cashflows of Z then $DZ_{r:T} = 0$.
- $a \star H_{r:T} := \sum_{u=r}^{T} a_r(H_T - H_r)$ represents future hedges.
  - $a \star H_{t:T} = a \star H_{t:r} + a \star H_{r:T}$ with past and future separation. Here the future contains only new hedges. Old hedges are captured in Z.
- $C_{r:T}(a) := \sum_{u=r}^{T} c_r(a_r)$ are cost

Deep Bellman Hedging

- This gives

$$U_t^* = \max_{a=a_t,\dots,a_{T-1}} : U_t\big(DZ_{t:T} + a \star H_{t:T} - C_{t:T}(a)\big)$$

- Replace for the moment U with the expectation operator E.

$$E_t^* = \max_{a=a_t,\dots,a_{T-1}} : E_t\big(DZ_{t:T} + a \star H_{t:T} - C_{t:T}(a)\big)$$

- For any intermediate r we get

$$\max_a : E_t \Big(E_r\big(DZ_{r:T} + a \star H_{r:T} - C_{r:T}(a)\big) + DZ_{t:r} + a \star H_{t:r} - C_{t:r}(a) \Big)$$

  - Clear split between past and future.
  - If r is past the last cashflows of Z then, $DZ_{r:T} = 0$.
  - That means our approach really learns the difference between classic model price and risk-adjusted expected cashflows.

Deep Bellman Hedging

- Monotonicity means that

$$E_t^* = \max_{a_t,\ldots,a_r} : E_t\left(\max_{a_r,\ldots,a_T} E_r\big(DZ_{r:T} + a \star H_{r:T} - C_{r:T}(a)\big) + DZ_{t:r} + a \star H_{t:r} - C_{t:r}(a)\right)$$

- Which yields for $r > t$

$$E_t^* = \max_{a_t,\ldots,a_r} : E_t\big(E_r^* + DZ_{t:r} + a \star H_{t:r} - C_{t:r}(a)\big)$$

  - Boundary condition $E_T^* := 0$.
  - The same calculation is true if $E_t(X) = U_t(X) := -\frac{1}{\lambda}\log E_t[\exp(-\lambda x)]$ since the entropy is time-consistent. It and the expectation are the only law-invariant time consistent monetary utilities [1]

[1] Representation results for law invariant time consistent functions, Kupper et al
https://www.mat.univie.ac.at/~schachermayer/preprnts/prpr0138.pdf

Dynamic Programming in RL

- We use small variable names for instances, and capital letters for random variables. Then

$$V^*(\delta_t, s_t) = \max_a \colon\ U_t[\,\beta(s_t)\, V^*(\,\delta_t + a_t, S_{t+1}\,) + R(a, s_t)]$$

$$R(a, s_t) \coloneqq DZ_{t:t+1} + \delta_t DH_{t:t+1} + a \star H_{t:t+1} - C_{t:t+1}(a)$$
$$R(a, s_t) = dZ_t + (a + \delta_t) \cdot dH_t - c_t(a)$$

- Links past with future value function.
- Boundary condition $V^*(\delta, s_{T+1}) = 0$.
- $\beta(s_t) \leq 1$ is a discount factor (more on this later)

Dynamic Programming in RL

- Our representation

$$V^*(\delta_t, s_t) = \max_a : U_t[\,\beta(s_t)\,V^*(\,\delta_t + a_t, S_{t+1}\,) + R(a, s_t)]$$

  with boundary condition $V^*(\delta, s_{T+1}) = 0$ still implicitly depends on a fixed portfolio Z.

- We managed to make it dynamic in our hedging instruments but only by assuming they are the same across all states.

Dynamic Programming in RL

<u>We now make a mental leap:</u>

- Let M be a set of possible portfolios, e.g. all L^1 payoffs on our path space.
- If $a \in R^n$ and $\bar{z}, \bar{h}^1, \dots, \bar{h}^n \in M$ then $\bar{z} + a \cdot \bar{h} \in M$.
  - For $\bar{p} \in M$ we define the return of its model values as $dM_t(\bar{p})$ i.e. if $\bar{p}$ represents our portfolio Z then $dM_t(\bar{p}) = dZ_t$
  - If $\bar{p}$ is expired, then $dM_t(\bar{p}) = 0$.
- Define then then functional equation:

$$V^*(\bar{p}_t, s_t) = \max_a : \; U_t\left[\beta(s_t)\, V^*\left(\bar{p}_t + a_t \cdot \bar{h}, S_{t+1} \right) + R(a, \bar{p}_t, s_t)\right]$$

$$\begin{aligned} &R(a, \bar{p}_t, s_t) \\ &:= dM_t(\bar{p}_t) + a \cdot dM_t\left(\bar{h}\right) - c_t(a) \\ &= dM_t\left(\bar{p}_t + a \cdot \bar{h}\right) - c_t(a) \end{aligned}$$

Dynamic Programming in RL

- We also allow for different hedging instruments per time step.
  In summary we obtain

$$V^*(\bar{p}_t, s_t)$$
$$= \max_a : U_t\left[\beta(s_t)\, V^*\left(\bar{p}_{t+1} + a_t \cdot \bar{h}^t, S_{t+1}\right) + dM_t\left(\bar{p}_t + a \cdot \bar{h}^t\right) - c_t(a)\right]$$

- Past value
- Future value
- Immediate returns of new portfolio,
- Transaction cost
- Boundary condition $V^*(\bar{p}, s_{T+1}) := M_{T+1}(\bar{p})$.

Dynamic Programming in RL

- In dynamic programming, we start with a fixed point equation such as:

$$V^*(\bar{p}_t, s_t) = TV^*(\bar{p}_t, s_t)$$

with "Bellman operator" T given as

$$TV(\bar{p}_t, s_t) \coloneqq \max_a: \ U_t[\beta(s_t)V(\bar{p}^a_{t+1}, S_{t+1}) + dM_t(\bar{p}^a_{t+1}) - c_t(a)]$$
$$\bar{p}^a_{t+1} \coloneqq \bar{p}_t + a \cdot \bar{h}^t$$

- We no longer require that U is time consistent as we now solve for a local problem.
- We now *solve* for V^* … think of it being a neural network
  - Problems with boundary conditions i.e. $V^*(\bar{p}, s_{T+1}) \coloneqq M_{T+1}(\bar{p})$ …works.
  - Do we need a boundary condition?

Deep Bellman Hedging

- Classic approach to solve a fixed point equation $V^*(\bar{p}_t, s_t) = TV^*(\bar{p}_t, s_t)$ for an operator

$$TV(\bar{p}_t, s_t) \coloneqq \max_{a} : U_t[\beta(s_t)\, V(\bar{p}^a_{t+1}, S_{t+1}) + dM_t(\bar{p}^a_{t+1}) - c_t(a)]$$
$$\bar{p}^a_{t+1} \coloneqq \bar{p}_t + a \cdot \bar{h}^t$$

  - No boundary condition.
  - Start with arbitrary initial $V^0 \coloneqq 0$. Let $V^{n+1} \coloneqq TV^n$.
  - If $\beta < 1$ and $T0 < \infty$ then this converges to a finite optimal solution for any monetary utility U. More natural if cashflows are not discounted. Proof in [1].

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Practical Implementation

- Practical issue even in the boundary case:

$$TV(\bar{p}_t, s_t) := \max_a: \; U_t[\beta(s_t)\, V(\bar{p}^a_{t+1}, S_{t+1}\,) + dM_t(\bar{p}^a_{t+1}) - c_t(a)]$$
$$\bar{p}^a_{t+1} := \bar{p}_t + a \cdot \bar{h}^t$$

- How do we represent elements of the set M of "all payoffs" ?
  - Idea in [1]: use signature representation … pretty heavy. Might still work.
  - Represent vanilla options as a grid – AI Flow Trader @ JP Morgan. https://www.risk.net/awards/7928696/equity-derivatives-house-of-the-year-jp-morgan

[1] US patent https://uspto.report/patent/app/20210082049

Practical Implementation

- We are given historic data $s_0, \dots s_m$ for times $\tau_0, \dots, \tau_m$.
- At each historic date τ_t we had booked k_t instruments. We also add our hedging instruments.
- For each instrument we have computed FV, greeks, scenarios and other additive risk metrics. Call those **feature vectors** $x_t^{t,i} \in R^F$ for τ_t and $x_{t+1}^{t,i}$ when computed at the next time step *for the same instruments*. If an instrument is expired its feature vector is zero. The joint matrix is p_r^t for $r = t, t+1$.
- We also assume that we have historically collected cashflows $m_t = (m_t^1, \dots, m_t^{k_t})$ between τ_t and τ_{t+1} for these instruments.
- We denote by M_r^t the model value in r of the instruments booked in t, here taking into account only future cashflows (the standard in FV calculations). That means

$$dM(x_t) \coloneqq M_{t+1}^t - M_t^t + m_t$$

- The feature vectors for our hedging instruments are denoted by h_r^t.

Practical Implementation

- Solve for V^* which satisfies for all pairs (w, t), see also [1]

$$V^*(w \cdot x_t^t, s_t) = \max_a: \; U_t \left[\begin{array}{c} \beta(s_t)\, V^*(w \cdot x_{t+1}^t + a \cdot h_t^t, S_{t+1}) \\ + dM(w \cdot x_t^t) + a \cdot dM(h_t^t) - c_t(a) \end{array} \right]$$

- Starting portfolio
- New terminal portfolio
- Returns of starting portfolio
- Returns of new hedge
- Transaction cost

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Practical Implementation

- Numerical solution via Actor-Critic: let

$$TV(w, s_t) := \max_a : \; U_t \left[\begin{array}{c} \beta(s_t)\, V(w \cdot x_{t+1}^t + a \cdot h_t^t, S_{t+1}) \\ +dM(w \cdot x_t^t + a \cdot h_t^t) - c_t(a) \end{array} \right]$$

- Chose random weights W. Let $V^0 := 0$.
- **Actor:** given V^{n-1} maximize above for a function (neural network) a^n of the feature vector $f^{(w)} := (w \cdot x_t^t, s_t)$.
- For $U = E$ this looks as follows:

$$\max_a : \beta(s_t) \sum_{w,t} \frac{1}{|W| m} \left[\begin{array}{c} V^{n-1}\left(w \cdot x_{t+1}^t + a^n(f^{(w)}) \cdot h_t^t, S_{t+1}\right) \\ +dM\left(w \cdot x_t^t + a^n(f^{(w)}) \cdot h_t^t\right) - c_t\left(a^n(f^{(w)})\right) \end{array} \right]$$

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Practical Implementation

- For OCE monetary utilities U with utility function u we also solve for a network y^n

$$\max_{a,y}: \sum_{w,t} \frac{1}{|W|m} u\left[(*) + y(f^{(w)})\right] - y\left(f^{(w)}\right)$$

$$\begin{aligned}(*) = \beta(s_t)\, V^{n-1}\left(w \cdot x_{t+1}^t + a^n(f^{(w)}) \cdot h_t^t, S_{t+1}\right) \\ +dM\left(w \cdot x_t^t + a^n(f^{(w)}) \cdot h_t^t\right) \\ -c_t\left(a^n\left(f^{(w)}\right)\right)\end{aligned}$$

- Note this also yields optimal samples $TV^{n-1}(w, s_t)$.

Practical Implementation

- **Critic:** given samples of TV^{n-1} solve for neural network V^n to satisfy

$$V^n(w, s_t) \equiv TV^{n-1}(w, s_t)$$

- We minimize quadratic distance to find a neural network V^n which solves

$$\min_V: \sum_{w,t} \frac{1}{|W|m} \left(V(w, s_t) - TV^{n-1}(w, s_t) \right)^2$$

- This interpolation program may also be solved by simpler, classic methods such as kernel interpolators.

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Practical Deep Bellman Hedging

[1] is the first of its kind

- Full Bellman RL with portfolio and market as state
- Continuous state and action space
- Derivatives as hedging instruments

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Practical Deep Bellman Hedging

Experimental framework which still requires lots of work

- Numerical issues:
    - How long to train in each actor/critic step? AC literature uses *one* step… does not seem to work.
    - Search for optimal action over network V^{n-1} inefficient as not concave.
- Parametrization
    - Using the proposed parametrization: actual number of data points not huge. Can we expand universe with “market simulation”?
      In the current case, the task would be to create a portfolio simulator which generates pairs of both instrument and market data.
    - Other parametrizations: ….?
- Robustness vs. model estimation error; intra/extrapolation.

- Refer to forthcoming paper on arxiv for technical details

3.* Open topics

Market Simulator for Fixed Income

Existing work has focused on equities market

- Build a market simulator for fixed income markets
- Cover treasuries, futures, swap rates to start with
  - Assess performance on hedging off-the-run swaps
- Expand to swaptions and assess performance hedging Bermudans. (option exercise rights in our favour are easily incorporated)

Steering Wheels

Incorporating Trader Views

- As it stands there is no easy mechanism for a trader to alter the behaviour of Deep Hedging.
- Classic method of shifting market data requires heavy re-training of market simulator, and subsequently of Deep Hedging
- Is there a more explicit way to express opinions of future market regimes … and if so, how do we measure the quality of these opinions as it pertains hedging performance.

Interpolation and Extrapolation

Knowing when to Stop (1)

- Deep Hedging is via its market simulator trained with a range of input data.
- When used in production, the new state might be outside the range of experience our historic data.
- Find robust, easily implemented methods to identify such cases and either warn the user, or fall back to a robust default strategy.
- General topic in machine learning

Modelling Incoming Trade Flow

Predictive Model for Client Demand

- As it stands Deep Hedging was modelled assuming the given portfolio does not change.
- In real life, clients will request prices for trades. Deep Hedging provides marginal pricing *but* it does not help to anticipate probable client demand.
- Challenge is to find a parametrization of “client demand” which can be efficiently modelled and integrated into Deep Hedging.

Explainability

Understanding why Deep Hedging makes certain decisions

- General topic in Machine Learning
- If the model provides an *unexpected* action. How do we know whether it is “correct” and understand “why” the model is recommending it.

Please ask questions