OST
Eastern Switzerland
University of Applied Sciences

New Frontiers in Quantitative Risk Management

FinTech Colloquium

Thomas Krabichler

Autumn 2020 (Updated Version)

Centre for Banking & Finance

Purpose of the Talk

Objective

This presentation is supposed to provide you with

- **selected challenges** that arise in the financial industry,
- an introduction to how these challenges can be tackled by means of **machine learning** techniques.

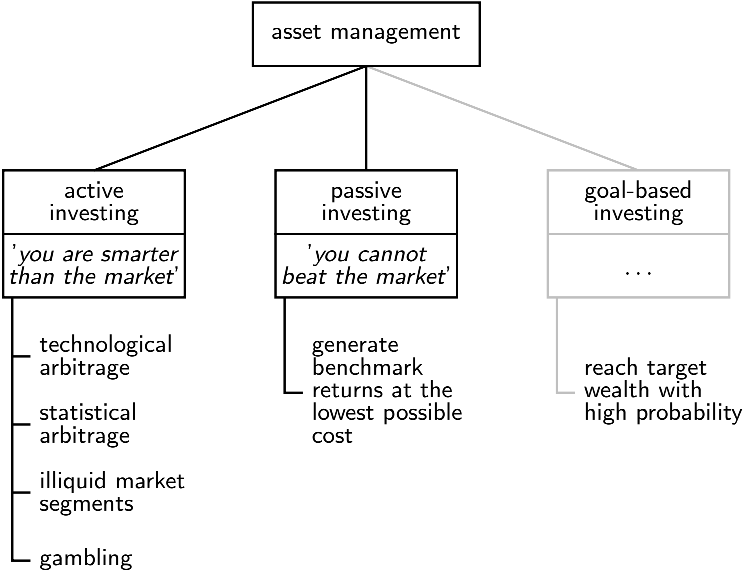
Disclaimer

- This introduction does **not** provide a **comprehensive** overview of how machine learning techniques are applied in the financial industry.
- The presented topics may grant an essential competitive advantage. However, please be aware of **inherent risks**.
- This talk does not disclose any profitable investment strategies.

Outline

1. Challenges
   - Asset Management
   - Pricing and (Over-)Hedging
2. Neural Networks
3. Machine Learning
   - Supervised Learning
   - Reinforcement Learning
4. Applications

Asset Management



Valuation and (Over-)Hedging

What is a fair price $P(0,T)$ of getting one monetary unit at time $T > 0$ as seen from $t = 0$?

- naive approach:
$$P(0,T) = 1$$
issues: **inflation risk**, **credit risk**, **liquidity risk**

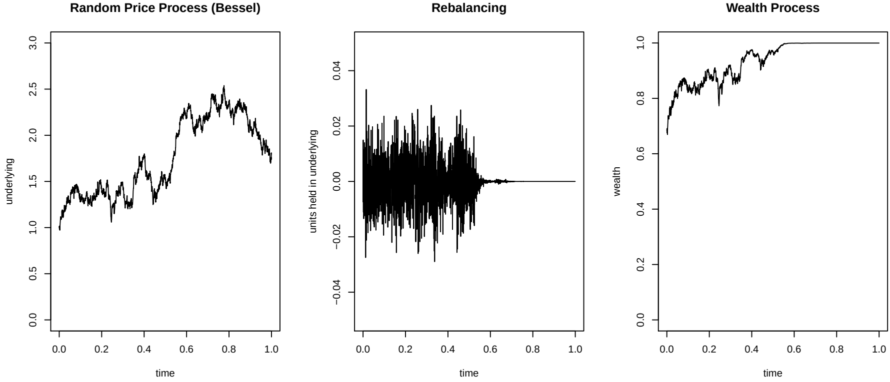
- static approach:
$$P(0,T) = \frac{1}{(1+r)^T}$$
for some interest rate r

Risk-Adjusted Valuation

$P(0,T)$ is the **minimal cost** to (super-)replicate the desired payoff.

Valuation and (Over-)Hedging

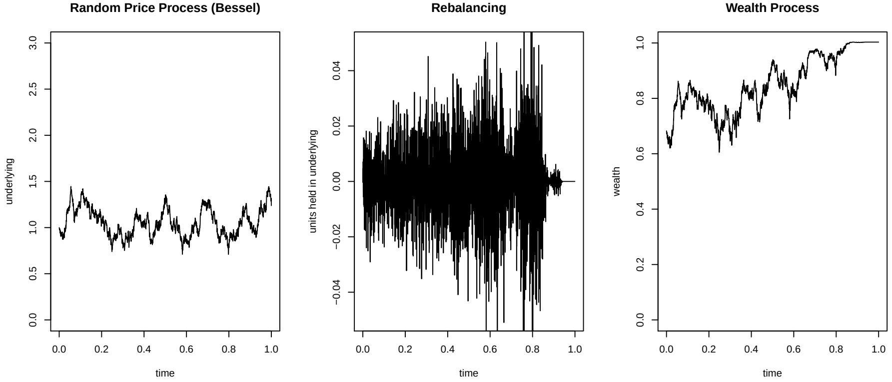
Monte-Carlo



- random price process $S_t = \sqrt{{W_{1,t}}^2 + {W_{2,t}}^2 + {W_{3,t}}^2}$
- (almost) frictionless (delta-)hedging results in minimal super-replication cost of 0.68

Valuation and (Over-)Hedging

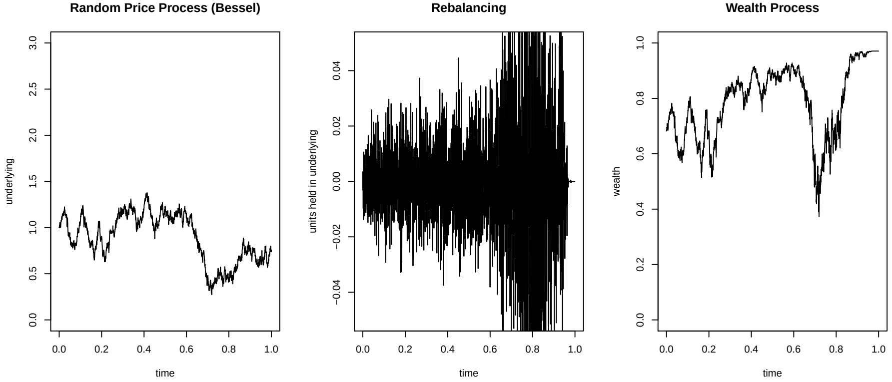
Monte-Carlo



- random price process $S_t = \sqrt{{W_{1,t}}^2 + {W_{2,t}}^2 + {W_{3,t}}^2}$
- (almost) frictionless (delta-)hedging results in minimal super-replication cost of 0.68

Valuation and (Over-)Hedging

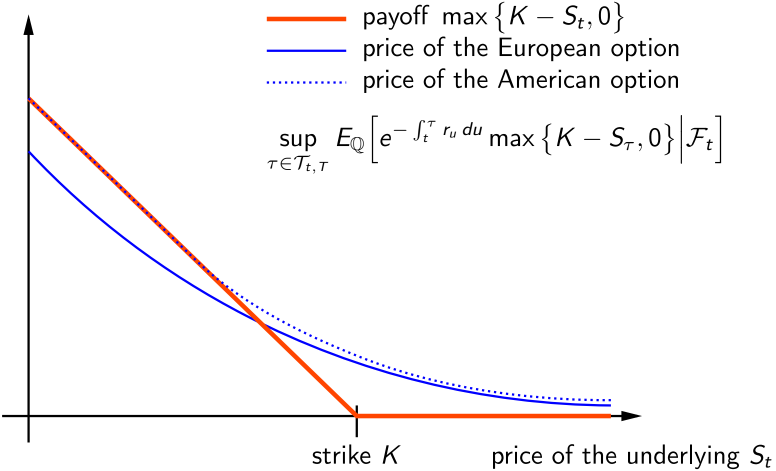
Monte-Carlo



- random price process $S_t = \sqrt{W_{1,t}{}^2 + W_{2,t}{}^2 + W_{3,t}{}^2}$
- (almost) frictionless (delta-)hedging results in minimal super-replication cost of 0.68

Valuation and (Over-)Hedging

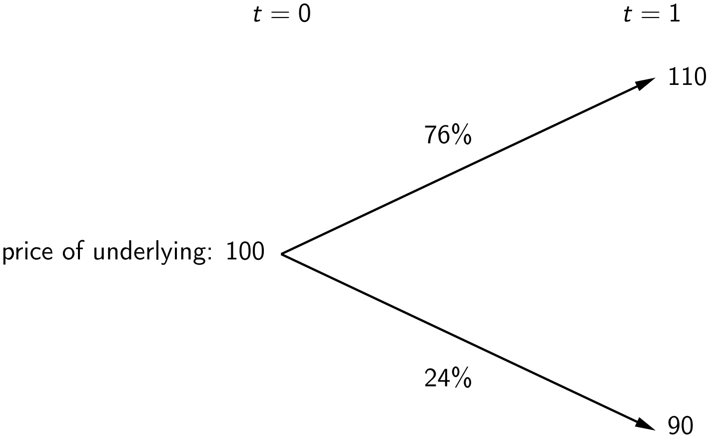
Option Pricing



Valuation and (Over-)Hedging

Dynamic Programming

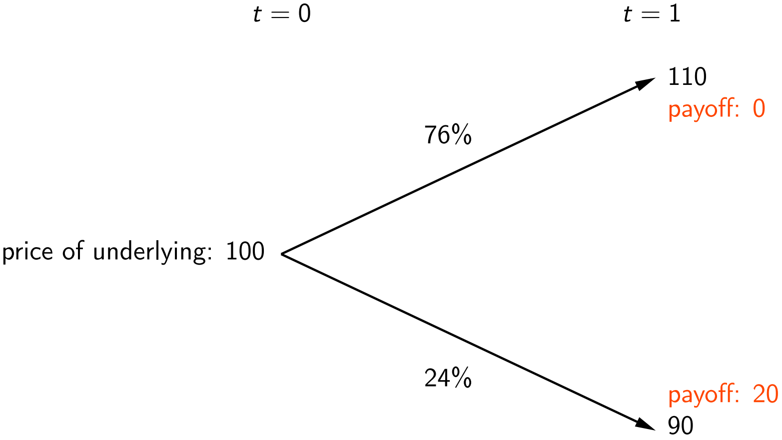
Discrete World: $K = 110$, $r = 5\%$



Valuation and (Over-)Hedging

Dynamic Programming

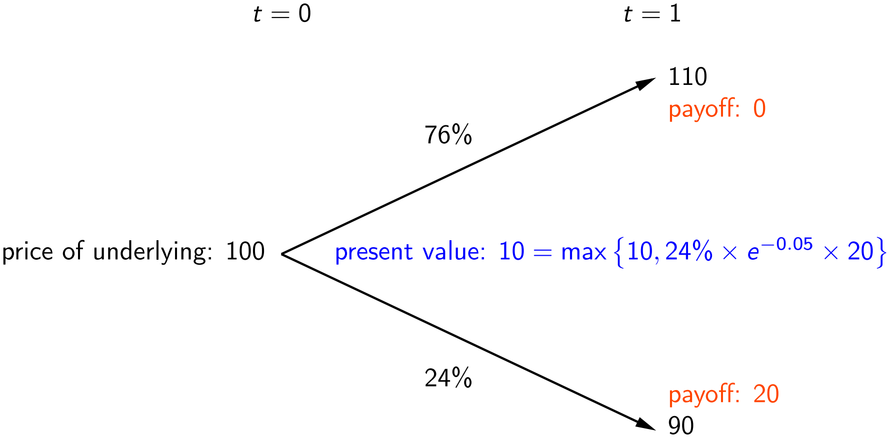
Discrete World: $K = 110$, $r = 5\%$



Valuation and (Over-)Hedging

Dynamic Programming

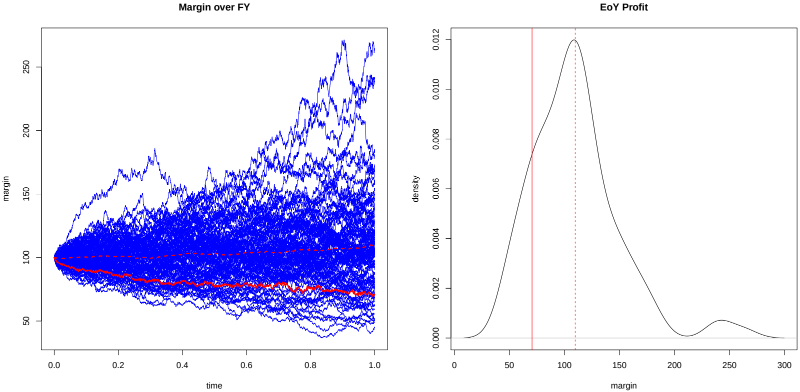
Discrete World: $K = 110$, $r = 5\%$



Valuation and (Over-)Hedging

Flaws of Classical Valuation Approaches

- Monte-Carlo-techniques or dynamic programming tend to be **computationally intensive**.
- The **level of sophistication** remains limited.



Valuation and (Over-)Hedging

The Curse of Dimension

No. of Underlyings	Discretisation of Space and Time	Runtime	Scale Unit
1	1 000	1	millisecond
2	1 000 000	1	second
3	1 000 000 000	17	minutes
4	10^{12}	12	days
5	10^{15}	32	years
6	10^{18}	317	centuries
⋮	⋮	⋮	⋮

- Longstaff-Schwartz (2001): 20 underlyings
- Becker-Cheridito-Jentzen (2018): 500 underlyings below 10 minutes with techniques inspired from **machine learning**

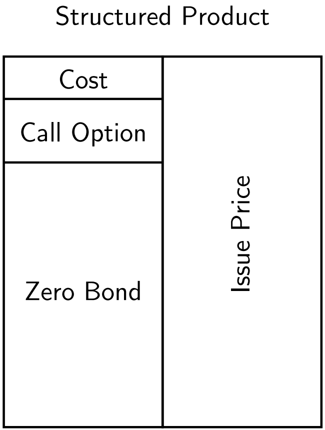
Valuation and (Over-)Hedging

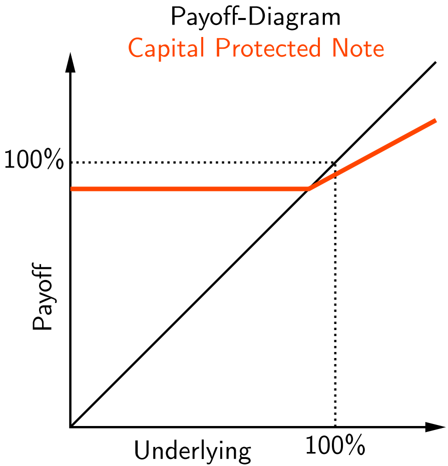
Investment in a Stock

source: Bloomberg

Valuation and (Over-)Hedging

Derivatives





Valuation and (Over-)Hedging

Capital Protected Note

Notional Amount	$NA =$ CHF 1 000
Issue Date	today ($t = 0$)
Maturity	$T = 1y$
Underlying	S&P500 index $(S_t)_{0 \leq t \leq T}$
Coupon	5%
Payoff	$NA \times$ (100% plus Contingent Payoff)
Issue Price	100%

Contingent Payoff (Down-and-Out Barrier Option): Provided that the underlying does not touch the knock-out barrier $94\% \times S_0$ during the lifetime of the contract (continuous observation), you will participate in the underlying's outperformance by getting

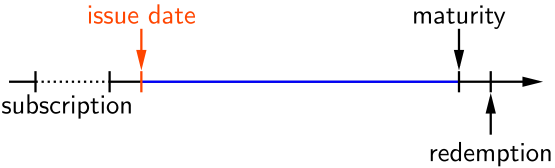
$$\max \left\{ S_T / S_0 - 105\%, 0 \right\}.$$

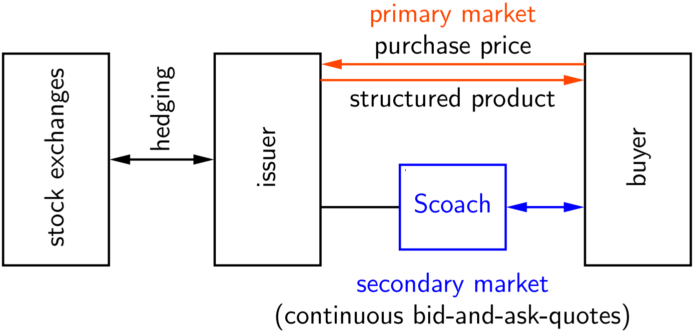
Valuation and (Over-)Hedging

- A **derivative** is a financial instrument whose price is derived from underlying market prices.
- Typical **underlyings** are commodities, currencies, equities, indices and rates.
- The **payoff-diagram** depicts the conversion of financial market scenarios into payoffs; see also the SVSP Swiss Derivative Map.
- According to Maringer et al., roughly 4% of the managed assets in Switzerland are invested in structured products.
- Reasons for their popularity:
  - They offer the possibility of **high returns** in every market situation.
  - They facilitate **bespoke** hedging and speculation.
  - The provide market access at **relatively low cost**.
- Create your own structured product: Credit Suisse my Solutions, Leonteq Constructor, UBS Equity Investor Marketplace, Vontobel Deritrade

Valuation and (Over-)Hedging

Lifecycle of Structured Products





Valuation and (Over-)Hedging

Hierarchy of financial assets from the accounting and pricing viewpoint (according to FASB 157):

- **Level 1**: Quotes are readily observable in the market.
- **Level 2**: Prices can be inferred through models and observable quantities.
- **Level 3**: Valuations involve complex models and subjective assumptions.

A professional and well-calibrated valuation platform must meet the following requirements:

- The model reprices level 1 products.
- The model features generally observed market phenomena.
- The model accounts for the significant risk drivers in a realistic manner.

Valuation and (Over-)Hedging

Risk-Adjusted Valuation

What is a fair price π_0 of getting $h(S)$ at time $T > 0$ as seen from $t = 0$, where $S = (S_t)_{0 \leq t \leq T}$ is a d-dimensional underlying risk factor and h some payoff function?

- Finding **realistic dynamics** is almost impossible due to the statistical uncertainty.
- The **(super-)replication strategy** is often not known explicitly.
- Trading off **complexity**, mathematical **tractability** and inherent **model risks** is very challenging.
- Analytically, it is very hard to deal with **transaction cost**.
- Maintaining and **automating** a suitable, efficient and well-calibrated valuation platform (e.g., stochastic local volatility models) for several thousand derivatives is tough.

The Game Has Changed

In 2017 a research group of DeepMind published the following results:

White	Black	Wins[3]	Draws	Losses
AlphaZero[1]	*Stockfish*	25	25	0
Stockfish[2]	*AlphaZero*	3	47	0

[1] AlphaZero is an algorithm that learns to play chess from scratch solely by **smart self-play**.

[2] Stockfish is a powerful open-source chess engine and TCEC world champion 2016.

[3] Outcome as seen from AlphaZero's perspective.

This result stimulates the imagination that quantitative methods for finance enter a new era.

The Game Has Changed

Paradigm

Regarding the presented challenges, what would a **clever**[1] financial agent with a lot of **experience**[2] and a decent **risk appetite**[3] do?

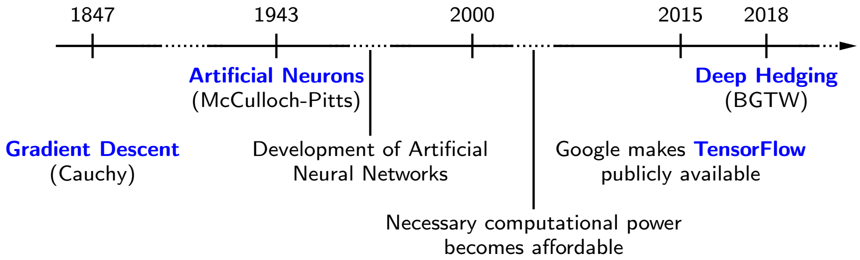
[1] The trained artificial agent has super-human skills in the specific task with respect to a given performance measure.

[2] The trained artificial agent has gained super-human experience in the considered task, e.g., a wealth of experience over 100 000 years acquired within as little as 30 minutes. Furthermore, the agent is unforgetful and demonstrates a consistent performance.

[3] The trained artificial agent can perfectly weigh up the benefits of a restructured portfolio and the costs to be borne. Its behaviour and performance can be validated almost instantaneously for arbitrary base and stress scenarios.

The Game Has Changed

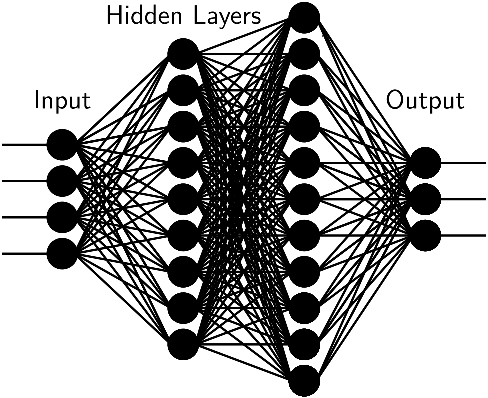
Selected Milestones



Outline

1. Challenges
   - Asset Management
   - Pricing and (Over-)Hedging
2. Neural Networks
3. Machine Learning
   - Supervised Learning
   - Reinforcement Learning
4. Applications

Neural Networks



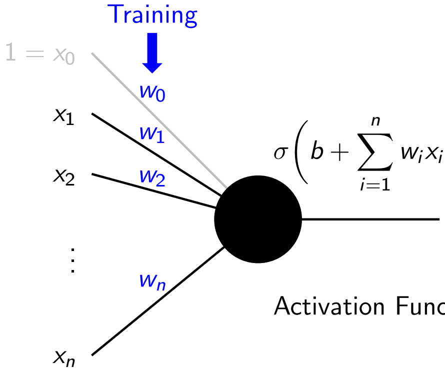
Machine Learning from the Mathematical Viewpoint

Simply put, it is the approximation of a high-dimensional non-linear function in terms of a (deep) neural network (DNN).

Neural Networks

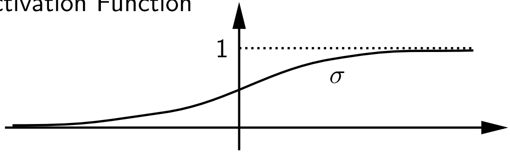
Perceptron

Input Weights Output



$$\sigma\left(b + \sum_{i=1}^{n} w_i x_i\right) = \sigma\left(\sum_{i=0}^{n} w_i x_i\right)$$

Activation Function



Neural Networks

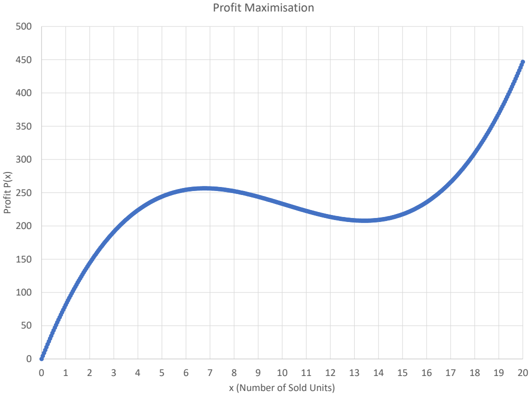
Mathematical Properties

- **Universal Approximation Theorems**: Provided that they are sufficiently large, neural networks can approximate complex functions arbitrarily close.
- Computing the derivative of the network output with respect to the weights is straightforward. Therefore, an incremental **learning process** becomes feasible.

Neural Networks

Business Problem

$$P(x) = \frac{1}{3}x^3 - 10.1x^2 + 91x$$

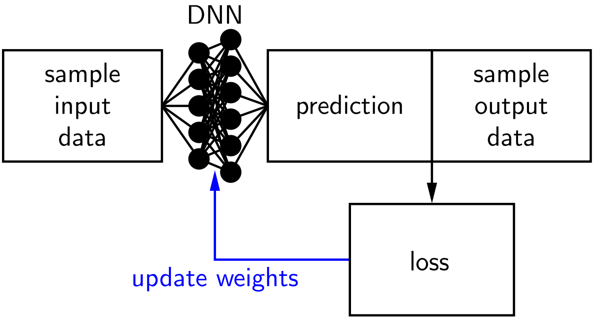


Outline

Machine Learning

Supervised Learning

Training: Minimise a Loss Function



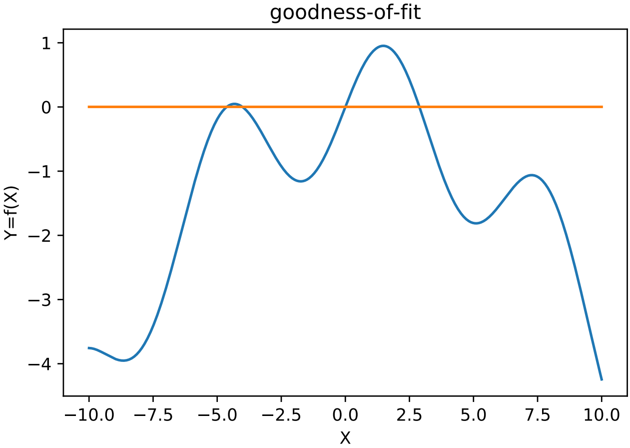
Validation: Check Accuracy of Prediction on Concealed Data

Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 0

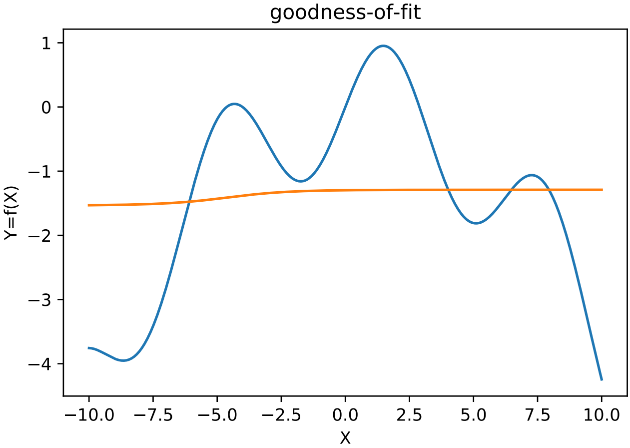


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 1 000

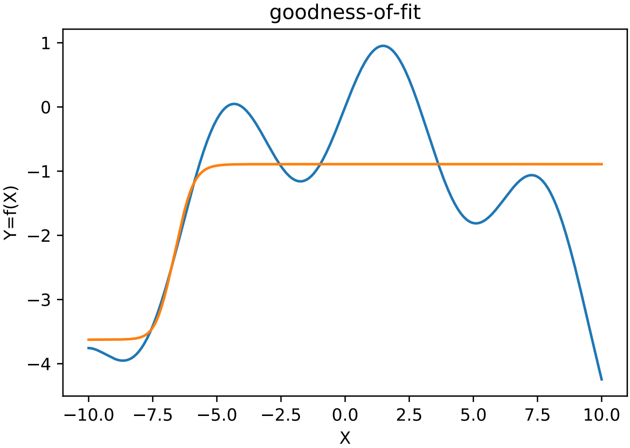


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 2 000

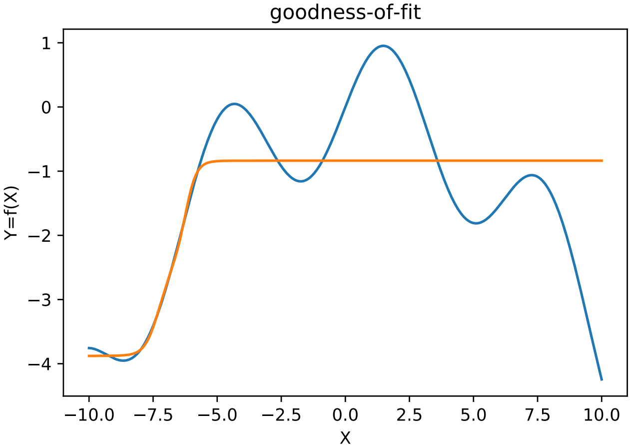


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 3 000

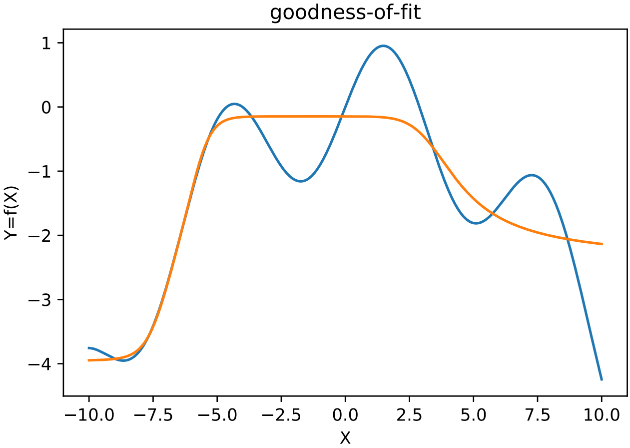


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 4 000

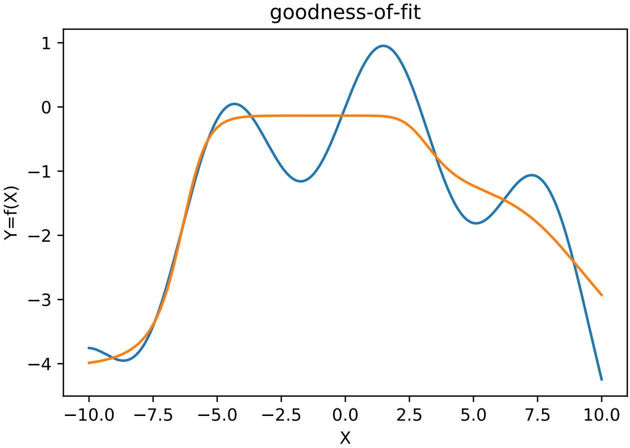


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 5 000

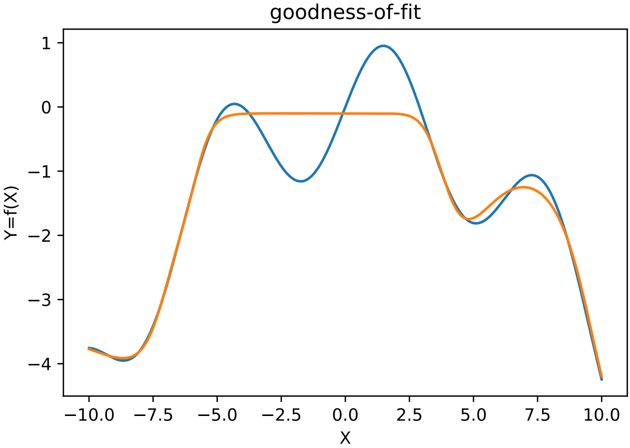


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 6 000

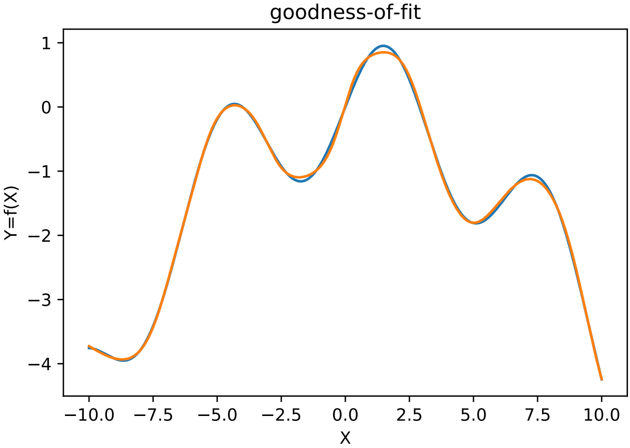


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 7 000

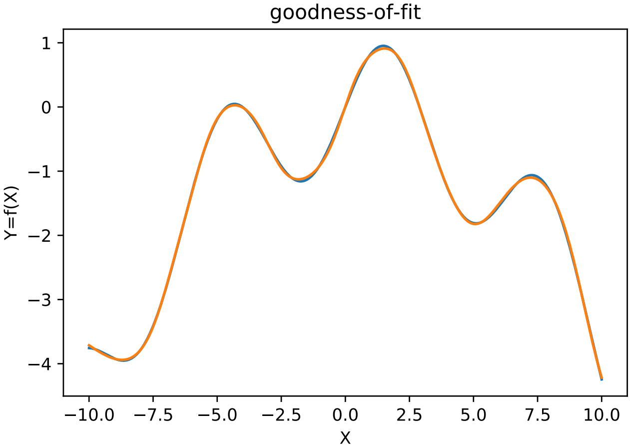


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 8 000

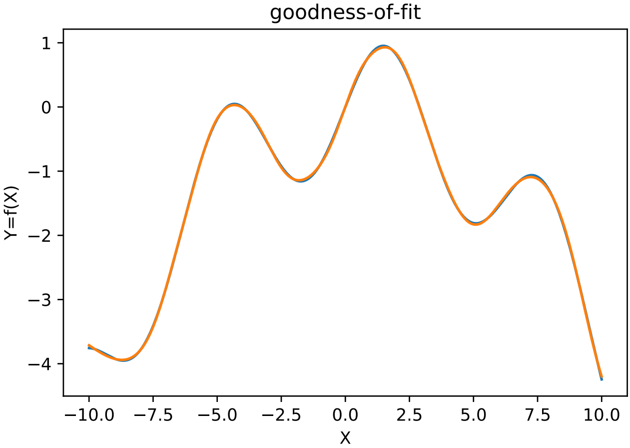


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 9 000

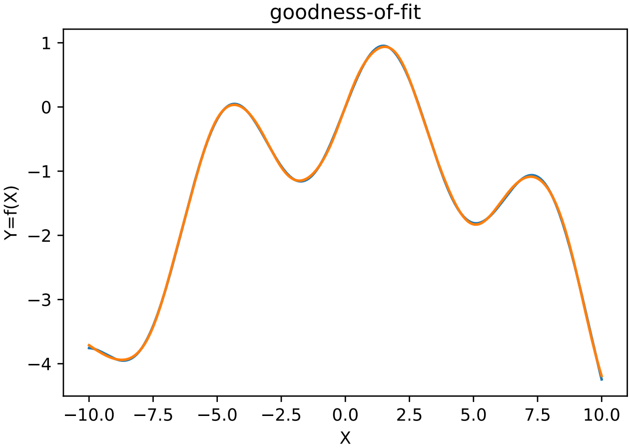


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 10 000

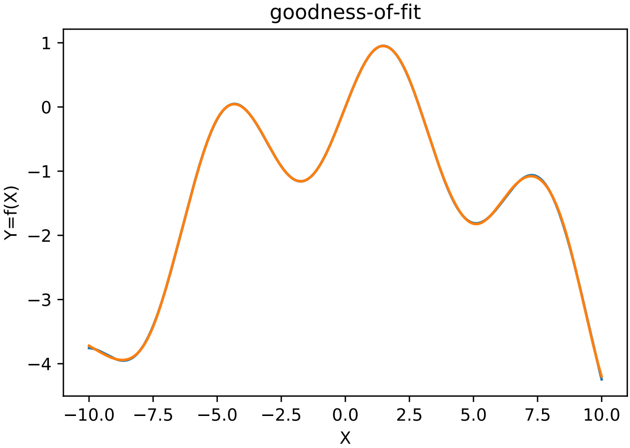


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 25 000

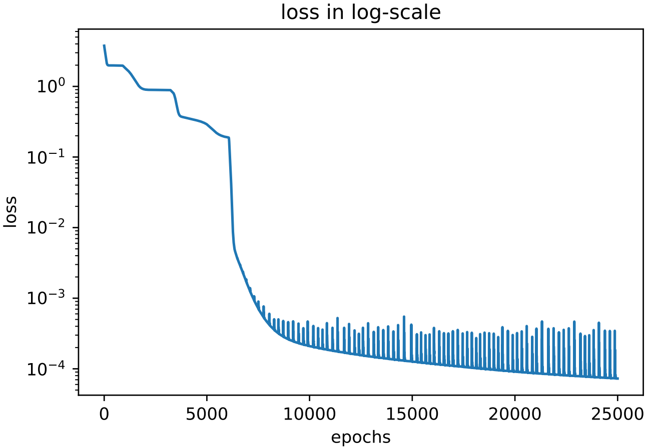


Machine Learning

Supervised Learning

Number of Nodes: 1–30–30–10–10–1

Number of Epochs: 25 000



Machine Learning

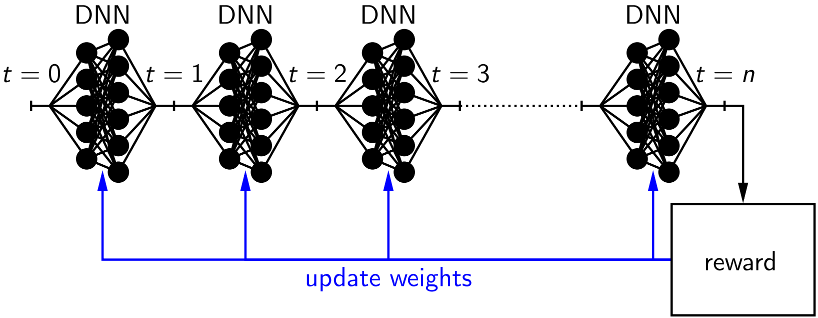
Observations

- The learning process evolves in small and random steps.
- The update of the weights results from the **backpropagation algorithm**. It can be seen as a very smart way of combining Monte-Carlo techniques and dynamic programming.
- Choosing suitable **hyperparameters** for the learning process might be tricky.
- Computing power is crucial.
- Neural networks can be evaluated efficiently by using pertinent software libraries, e.g., **TensorFlow**.
- Storing neural networks requires comparatively little storage space.

Machine Learning

Reinforcement Learning

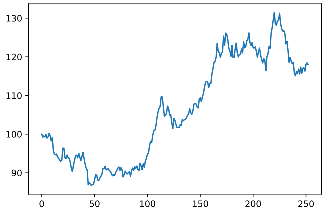
Training: Maximise a Reward Function

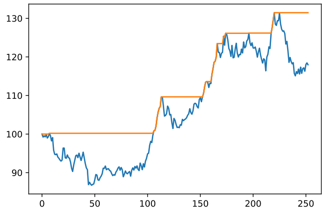


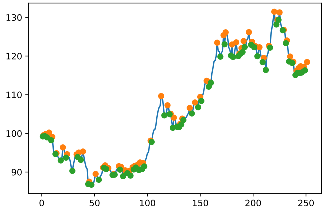
Validation: Check Performance of Decisions on New Scenarios

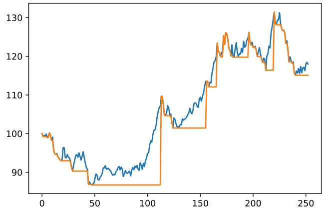
Reinforcement Learning

Scenarios, Features and States









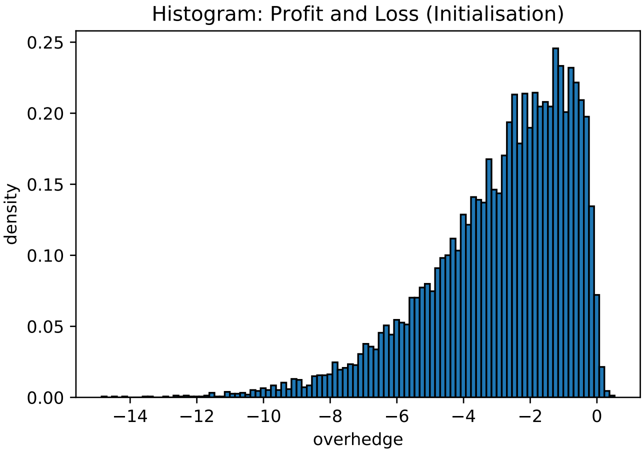
Machine Learning

Experiment on Deep Hedging

- Exposure: We issue a call option with payoff $\max\{S_T - K, 0\}$, strike $K = 100$ and maturity $T = 30d$.
- Market Environment:
  - bank account
  - underlying
- Rules:
  - Investment strategies must be **self-financing**.
  - Re-allocations are possible once a day and may involve proportional **transaction cost**.
- Objective: We aim to minimise the quadratic discrepancy between the due payoff and the value of the hedge.
- Training: 10 000 scenarios

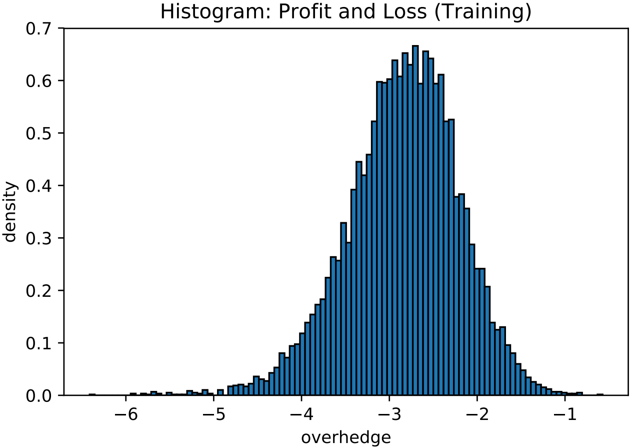
Machine Learning

Deep Hedging (without Transaction Cost)

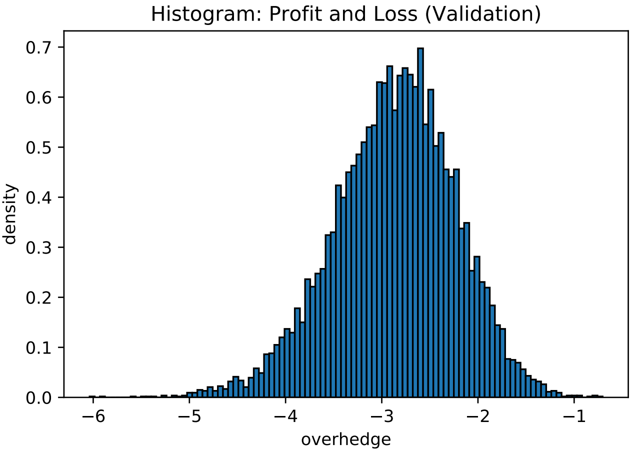


Machine Learning

Deep Hedging (without Transaction Cost)

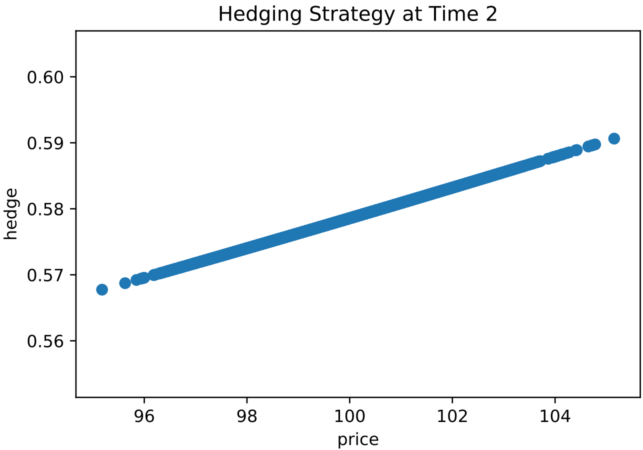


Deep Hedging (without Transaction Cost)



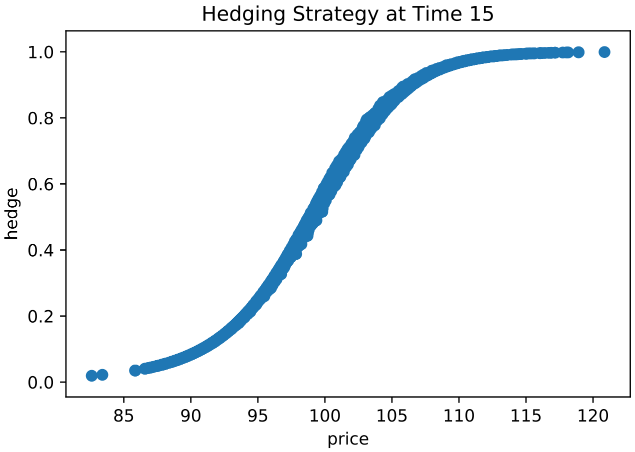
Machine Learning

Deep Hedging (without Transaction Cost)

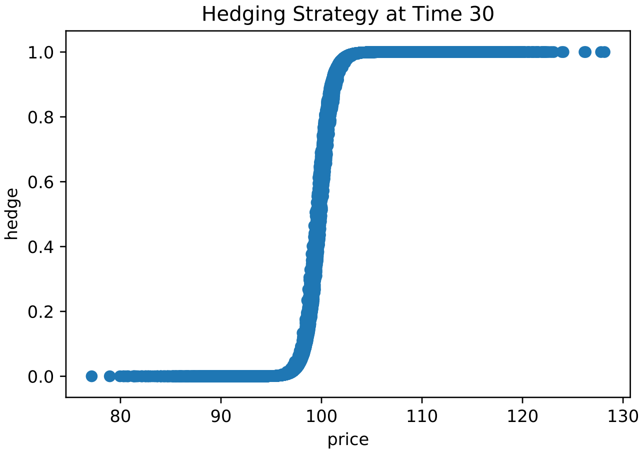


Machine Learning

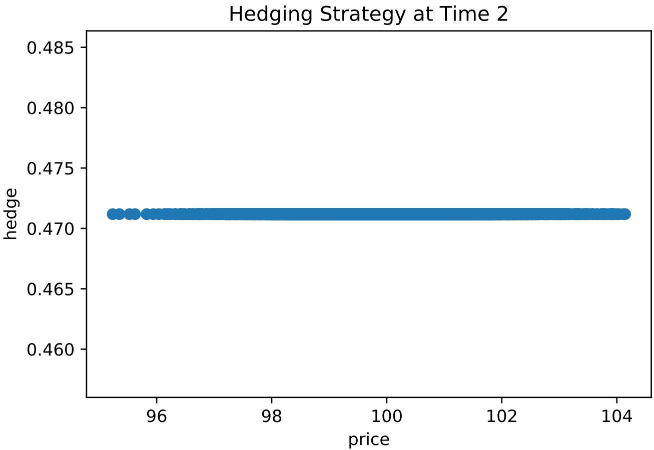
Deep Hedging (without Transaction Cost)



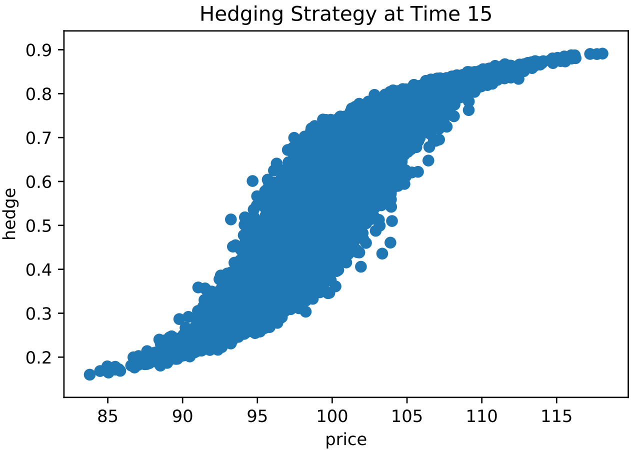
Deep Hedging (without Transaction Cost)



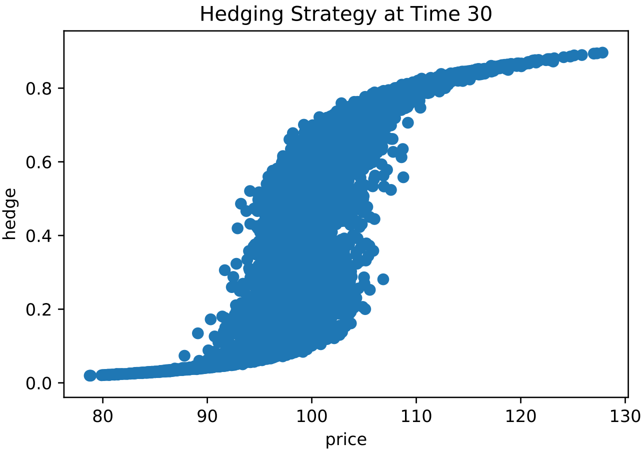
Deep Hedging (with Transaction Cost)



Deep Hedging (with Transaction Cost)



Deep Hedging (with Transaction Cost)



Machine Learning

Summary

Traditional Programming: The algorithm/recipe is specified line-by-line.

$$\text{data} + \text{program} \longrightarrow \text{output}$$

Supervised Learning: The instructors know and reveal the correct solution but not the approach (e.g., detection of counterfeit money).

$$\text{data} + \text{output} \longrightarrow \text{program}$$

Reinforcement Learning: The instructors do not know the «best» approach themselves; however, they can appraise the quality of a trial (e.g., quest for an optimal trading strategy).

$$\text{rules} + \text{scenarios} \longrightarrow \text{convincing strategy}$$

Machine Learning

Hypothesis

Techniques inspired from reinforcement learning pave the way for a new era in quantitative risk management from various viewpoints.

1. It is a **disruptive** technology; **high-dimensional** optimisation problems of this kind were not accessible until only recently.
2. It is a very **efficient** and **powerful** technology with
   - super fast requests-on-demand,
   - instantaneous validation (**model risk management**).
3. It is a very **flexible** technology. In a few lines of code, one easily accounts for
   - arbitrary path-dependent payoffs,
   - complex stochastic environments,
   - liquidity squeezes/transaction cost/price impacts,
   - regulatory constraints,
   - risk appetite.

Outline

Applications

Machine Learning in Finance

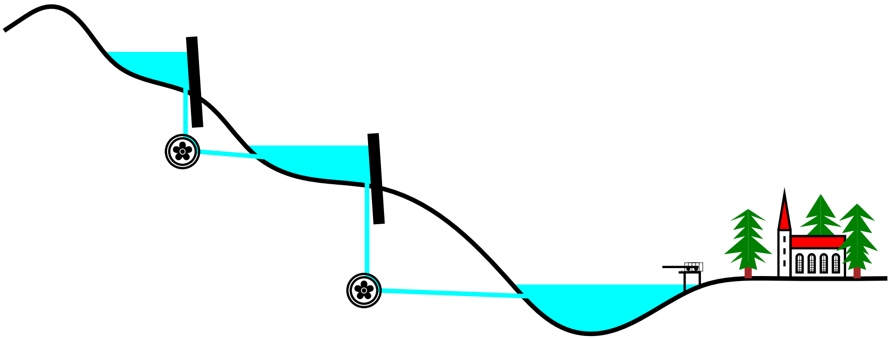
- Optimisation of business and **hedging** strategies
- Asset-liability-management, quantitative **risk management**
- **Valuation** of financial derivatives
- **Technology transformation** (automation, digitisation)
- **Forecasts** (e.g., client behaviour, credit migration and defaults, fraud detection, marketing)

  ⋮

The optimisation potential is (still) immense.

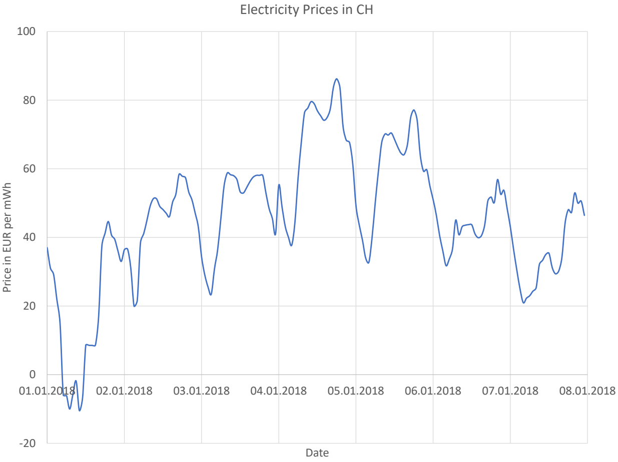
Applications

Hydro-electric Power Plant



Applications

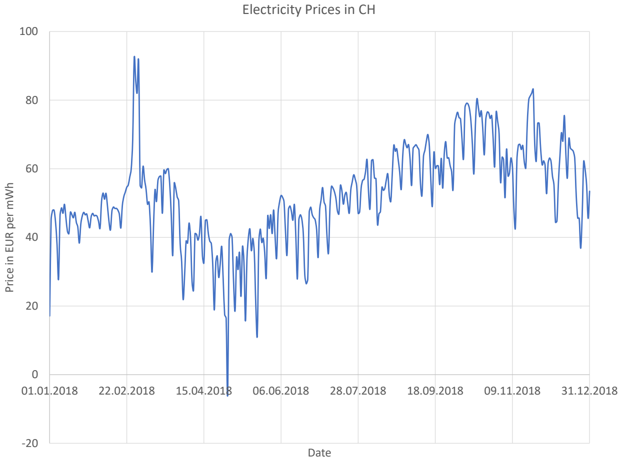
Hydro-electric Power Plant



source: Bloomberg

Applications

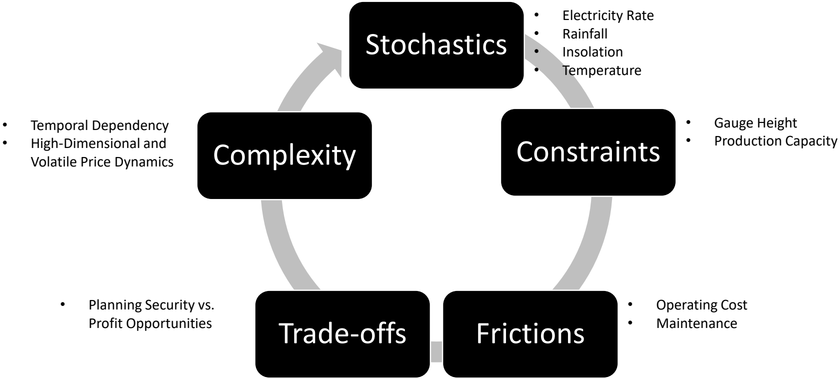
Hydro-electric Power Plant



source: Bloomberg

Applications

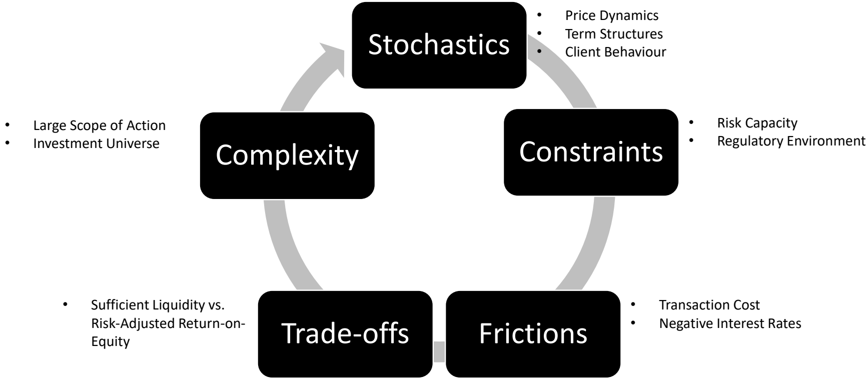
Hydro-electric Power Plant



Commodity: Energy

Applications

Asset-Liability-Management of a Financial Enterprise



Commodities: Credit, Liquidity, Money

Applications

Balance Sheet of a Financial Enterprise

assets	liabilities
investment portfolio	debts
	equity, share capital

Objective

Maximise the expected utility of the **return-on-equity** over different time instances while not exceeding a certain **draw-down** and while guaranteeing the **regulatory constraints** with a high probability.

Applications

Balance Sheet of a Retail Bank

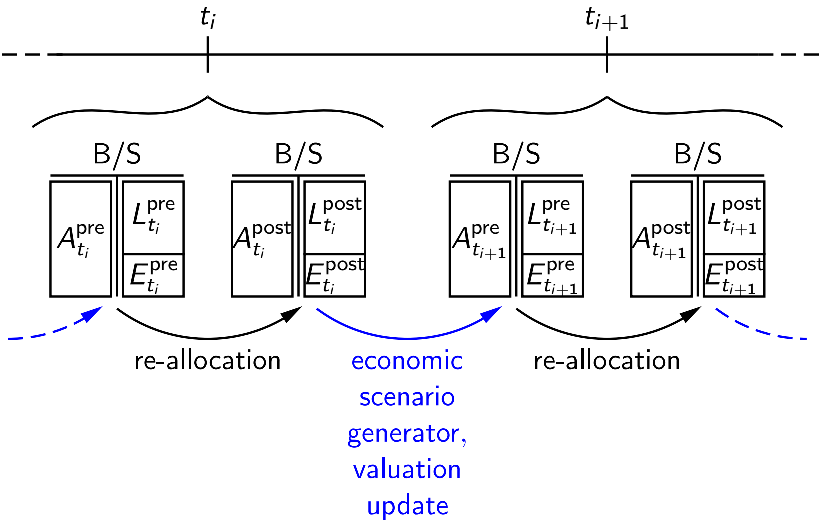
assets	**liabilities**
bonds (liquid, illiquid)	deposits (FT, NM)
credit (FT, NM)	interbank loans
RRR surplus	central bank reserve
cash	repos
equities swaps	equity, share capital

Objective

Maximise the expected utility of the **return-on-equity** over different time instances while not exceeding a certain **draw-down** and while guaranteeing the **regulatory constraints** with a high probability.

Applications

Balance Sheet Roll-Forward



Model Ingredients for Reinforcement Learning

- **economic scenario generator**
  - yield curves
  - credit migrations
  - stock prices
  - client behaviour

    ⋮
- parameterisation of the **states**
- **rule book**
  - constraints
  - eligible balance sheet restructuring
  - frictions
- **objective**

Applications

Deep Asset-Liability-Management

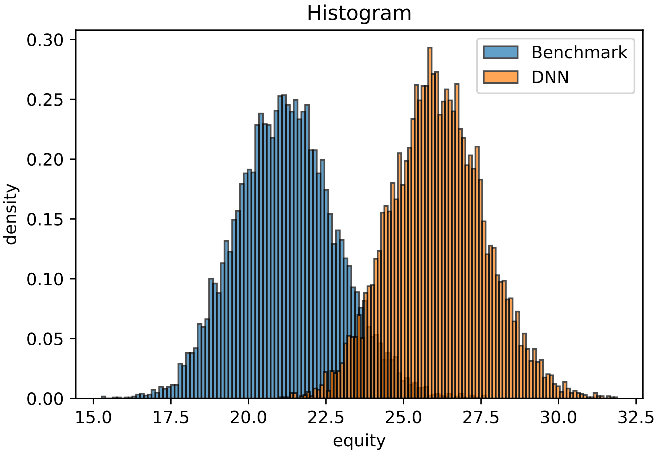
Simply put, one solves **high-dimensional** hedging problems with **constraints** in the presence of **frictions** by means of techniques inspired from reinforcement learning.

Applications

Similar Use Cases

- insurance company
  - pricing of contracts that accounts for insurance **risks**
  - optimised reinsurance programme
  - investment strategy that accounts for the necessary returns and **liquidity**
- production company
  - trading with **pricing impact**
  - optimal procurement under uncertainty and storage cost
- power production
  - optimised production under **uncertainty** and **constraints**
  - pricing and **hedging** in an **illiquid** environment

Applications



Applications

Further Research

- Reach a suitable level of **complexity**.
- Deal with **uncertainty** of model assumptions.
- Model choices and regulisations that promote **robust solutions**.
- Corroborate that sophisticated approach and additional complexity is **profitable**.

Concluding Remarks

The difficulty of a problem is always relative; certain problems are «difficult» for humans and «easy» for computers, and vice versa.

References

Becker, S., Cheridito, P., Jentzen, A.
Deep Optimal Stopping (2019).
Journal of Machine Learning Research. Vol. 20, No. 74, pp. 1–25.

Bühler, H., Gonon, L., Teichmann, J., Wood B.
Deep Hedging (2019).
Quantitative Finance. Vol. 19, No. 8, pp. 1271–1291.

Krabichler, T., Teichmann, J.
Deep Replication of a Runoff Portfolio (2020).
arXiv: 2009.05034.

Longstaff, F. A., Schwartz, E. S.
Valuing American Options by Simulation: A Simple Least-Squares Approach (2001).
The Review of Financial Studies.

Maringer, D., Pohl, W. and Vanini, P.
Structured Products: Performance, Costs, and Investments (2016).
Swiss Finance Institue White Paper.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., Hassabis, D.
A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play (2018).
Science. Vol. 362, No. 6419, pp. 1140–1144.

Contact

thomas.krabichler@ost.ch