Webinar

Deep Neural Net Applications on Trading and Risks

Welcome

We will begin promptly at 11 AM ET.

If you are unable to hear the speakers, please let us know in the chat box. You may enter your questions in the Q&A, we will address them at the end of the presentation. You can find a copy of the slide deck and recording of this webinar: www.fdpinstitute.org/webinars

FDP INSTITUTE®

Financial Data Professional Institute

FDP Institute provides world class training and education to financial professionals to meet the accelerating needs of digital transformation in the industry.

Introductions

Hossein Kazemi, PhD, CFA
Senior Advisor,
CAIA Association &
FDP Institute

Gary Wong, PhD
Business & AI/ML
Applications Development
Artemis AG

Today's Topic:

Deep Neural Net Applications on Trading & Risks

Deep Neural Net Applications in Capital Markets

FDP Institute Webinar

Gary Wong PhD   Artemis AG

gary.wong@ArtemisAG.co.uk – for general AI/ML/new tech applications
gary.wong@riskfuel.com – for DNN applications

Bio

Gary Wong, PhD gary.wong@ArtemisAG.co.uk +44 7940 914065

- **AI (Artificial Intelligence) and ML (Machine Learning) application expert**
- **Technology Consultant working with investment banks and funds**
- **Background : Head Trader, Managing Director and Head of Structured Trading for a large global bank**

Provide strategic and project advice/support on implementation of practical applications of AI, ML and other new technologies in trading, quant modelling and risk management in major global banks and fintech firms, including

- ***Deep Neural Net*** *– volatility analysis and 1M+ times computation speedup on XVA*
- ***Natural Language Processing*** *– nowcasting for trading and risk*
- ***Graph Computing*** *– monitor and resolve data issues and to untangle black-box processes*
- ***Graph Analytics*** *– rapid identification of contagion and secondary impact*
- ***NoCode/LowCode*** *– streamlining operations and simplifying software improvements*

Deep Neural Net (DNN)

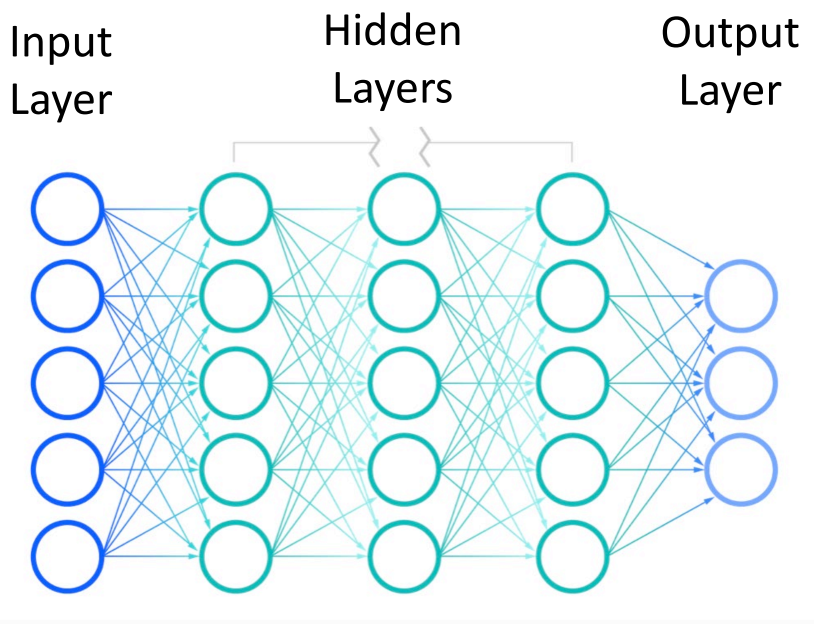
Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Deep Neural Networks (DNNs) are comprised of a node layers,

- Input layer
- one or more hidden layers
- Output layer

Each node (or neuron) connects to another and has an associated numerical **weight** and **threshold**. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Deep Neural Net (DNN)

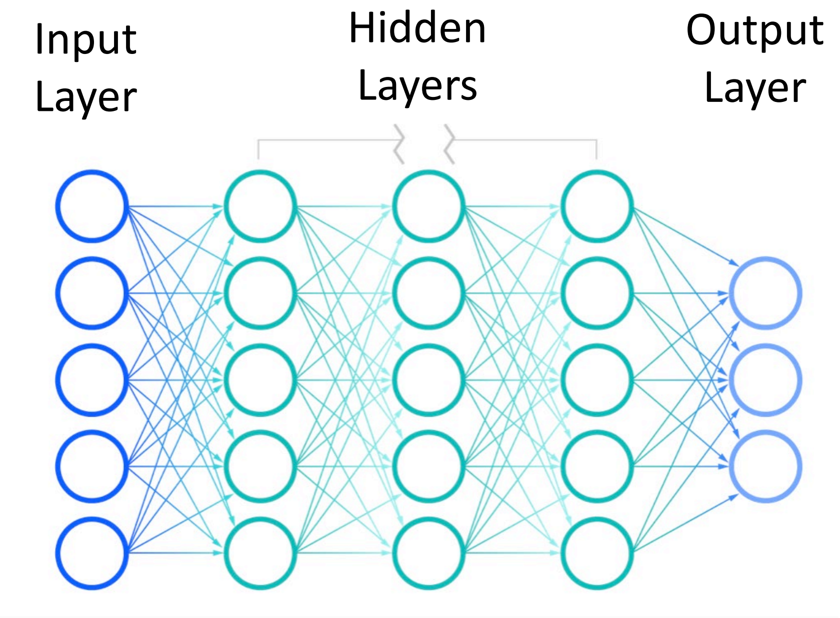


Deep Neural Net (DNN)

Neural nets are **a means of doing machine learning, in which a computer learns to perform some task by analysing training examples**. Usually, the examples have been hand-labelled in advance.

An object recognition system, for instance, might be fed thousands of labelled images of cars, houses, coffee cups, and so on, and it would find visual patterns in the images that consistently correlate with particular labels – by keep adjusting the **weights** and **thresholds**

Deep Neural Net (DNN)



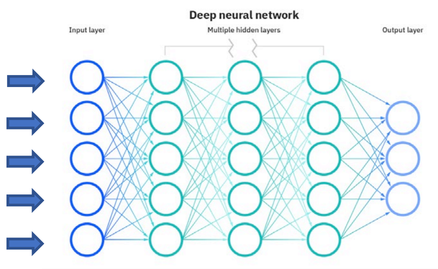
Neural Net Universality Theorem

http://neuralnetworksanddeeplearning.com/chap4.html

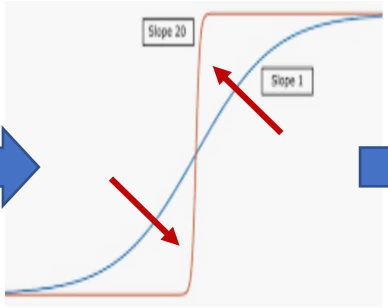
No matter what the function*, there is guaranteed to be a neural network so that for every possible input, x, the value f(x) (or some close approximation) is output from the network**

By adjusting the parameters like weights and bias, one can generate output of step functions

INPUTS

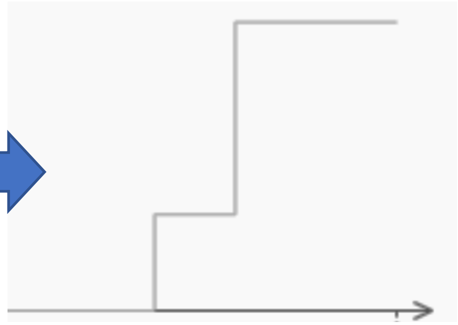


Adjust Parameters

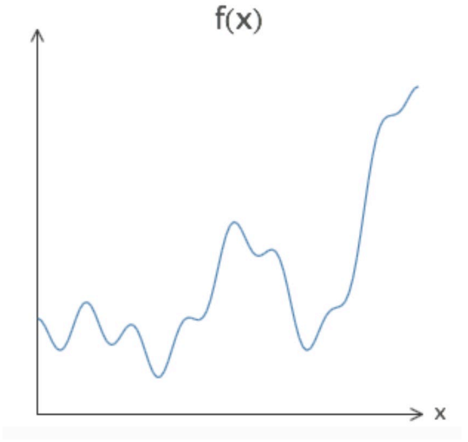


Sigmoid into Step

Adjust Parameters



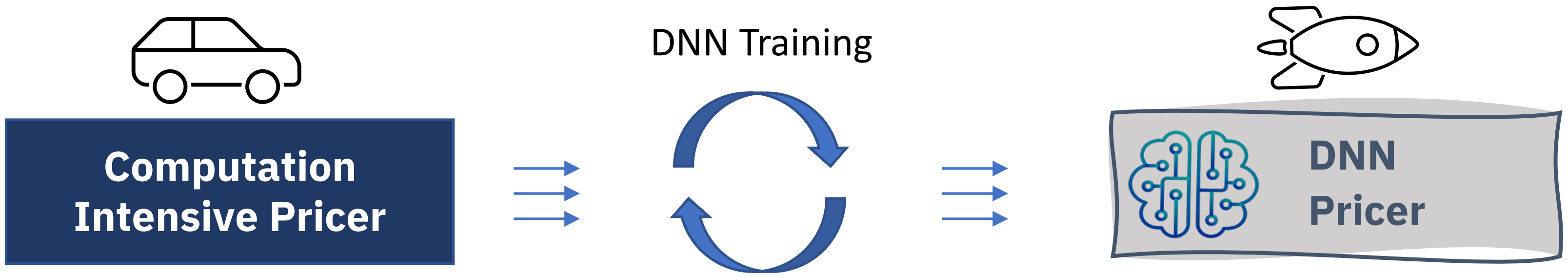
Superposition of multiple steps



* Continuous function ** Approximation – not exact – but can be arbitrarily close

Deep Neural Net (DNN) Use Case - Riskfuel

DNN Replication – huge speed up on complex models



- Train a Deep Neural Net to replicate **your existing computation-intensive model** for production run
- DNN is just a bunch of nodes with weights – so run VERY FAST calculations
- Quants can now concentrate on developing sophisticated trading and risk models – and run the DNN version in production without being limited by run time constraints

CAPITAL MARKETS: FRUSTRATIONS

“It is *SO DIFFICULT* to modify the infrastructure and adopt new tools …”

MODELS SOPHISTICATION VS RUN-TIME & COSTS

NO REAL-TIME RISK MANAGEMENT

COMPUTE IS WAY TOO EXPENSIVE

BIG OPERATIONAL RISK WITH OVERNIGHT BATCH

DEMANDING REGULATORY REQUIREMENTS (FRTB IS LOOMING)

Deep Neural Net (DNN) Use Cases in Capital Markets

DNN Replication for Speed and Computation Costs

- Complex non-analytical calculations (such as Monte Carlo simulations)
- Long-dated structures
- Real-time Pricing and Risks for Trading

Large-scale, repeated calculations – Risks and Regulatory Calculations

- Valuation Adjustments XVA – CVA, FVA, KVA… (requiring nested MC simulations)
- Fundamental Review of Trading Book (FRTB) – lots of sensitivities with multiple horizons
- Back-testing / Scenarios Analysis
- Capital Calculations

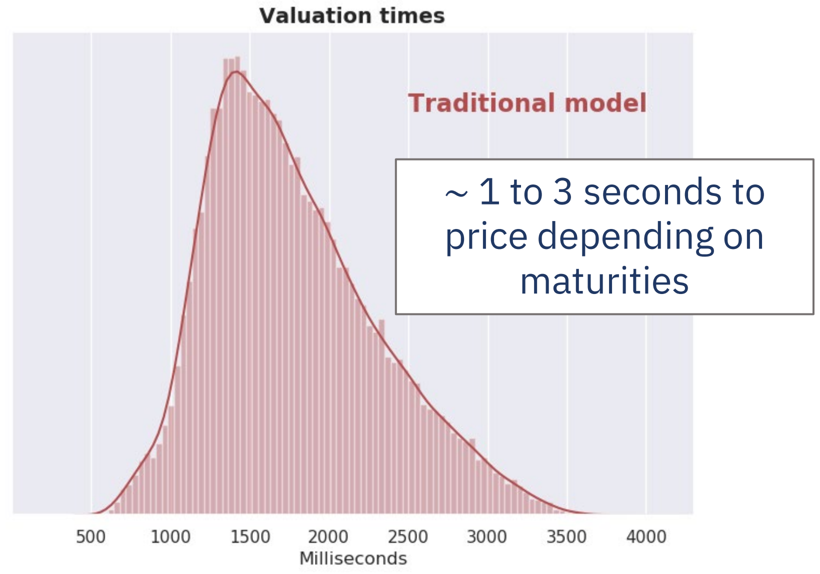
Reduce Operational Risks of large-scale calculations

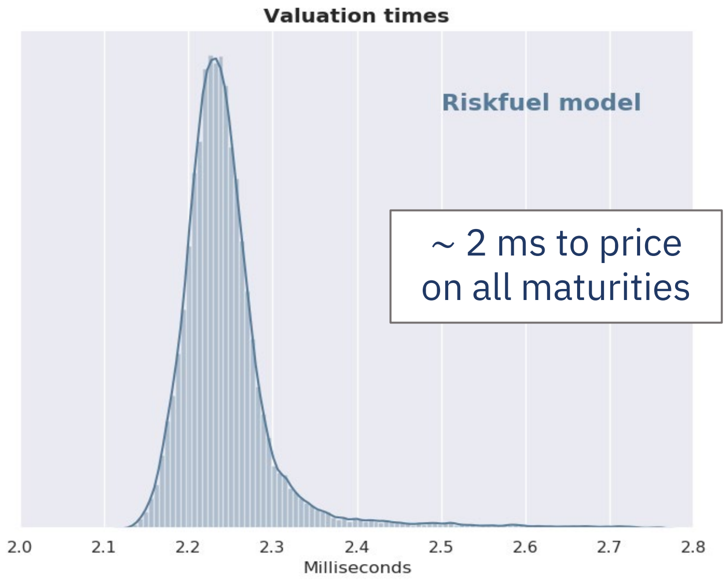
- Short End-of-Day and Overnight Run = Early Errors Detection = Quick Remediation = Low Ops Risk

Works with Microsoft Azure

"… the combination of a **Riskfuel-accelerated version of the foreign exchange barrier option model** and with an Azure ND40rs_v2 Virtual Machine showed a **20M+ times performance improvement over the traditional model**."

https://azure.microsoft.com/blog/azure-gpus-with-riskfuels-technology-offer-20-million-times-faster-valuation-of-derivatives



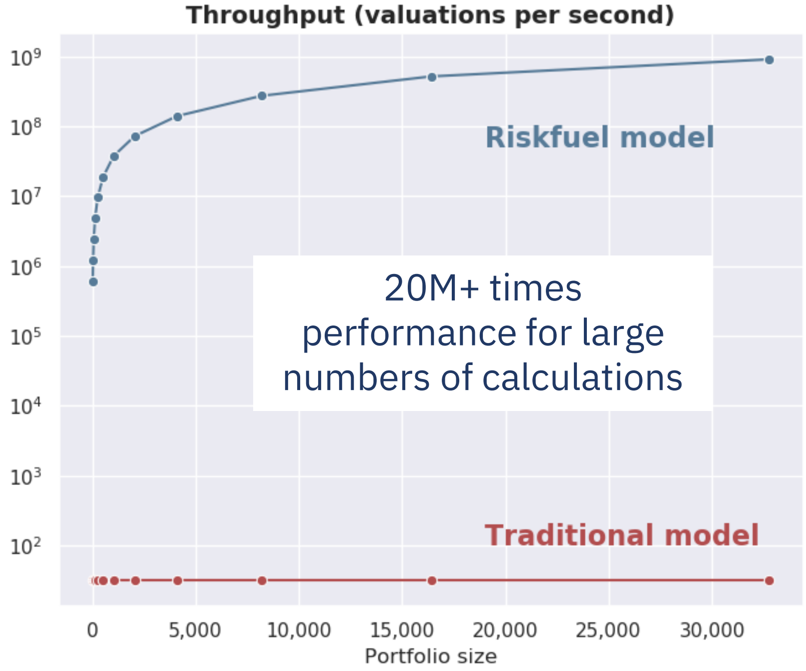


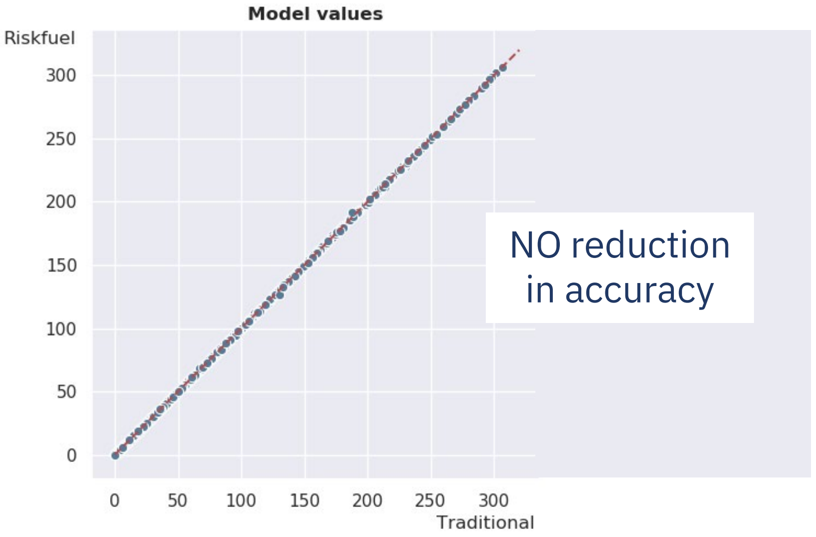
(… but are your results accurate? Ans: yes)

20M+ times performance improvement

“It is critical to point out here that the speedup resulting from the **Riskfuel** model does not sacrifice accuracy. In addition to being extremely fast, the Riskfuel model effectively **matches the results generated by the traditional model**.”

https://azure.microsoft.com/blog/azure-gpus-with-riskfuels-technology-offer-20-million-times-faster-valuation-of-derivatives



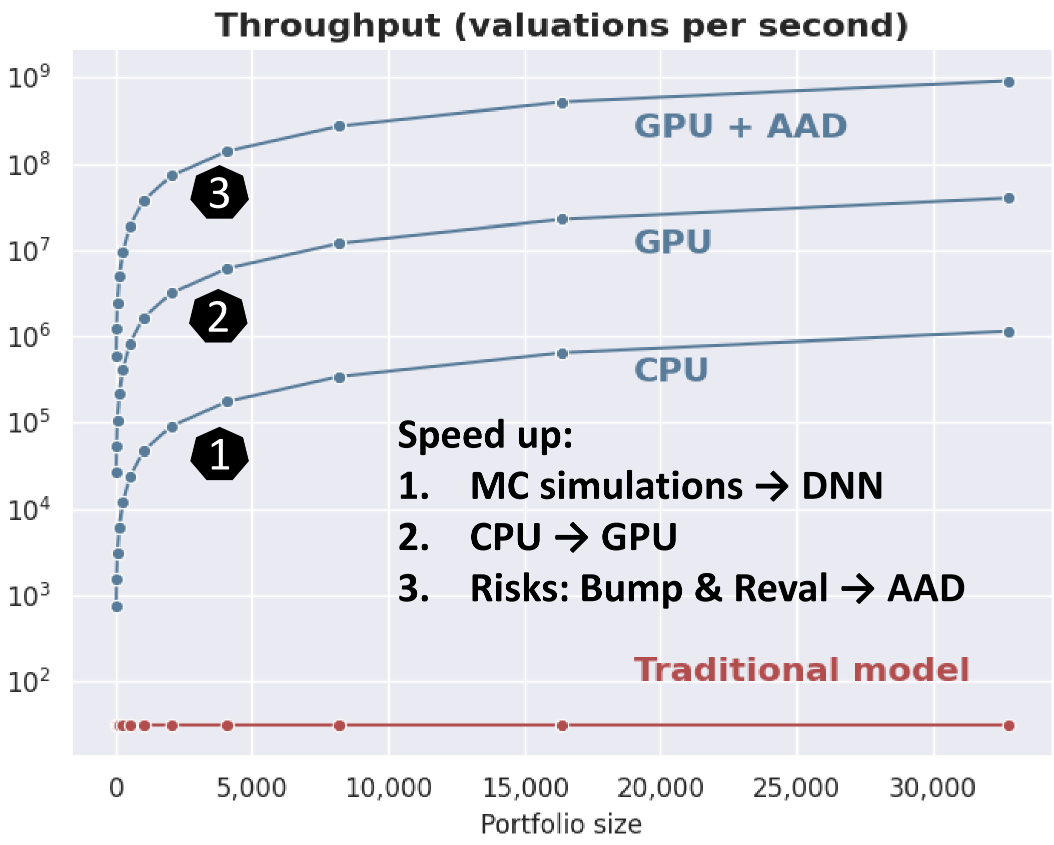


Risk.net Technology Award 2021 – Scotiabank XVA Platform

Microsoft/Nvidia/**Riskfuel**/Scotiabank XVA Platform

Example: FX Barrier Option

- 81 input dimensions
- Full FX volatility surface (5 x 12)
- 2 IR curves (domestic and foreign currency)
- Trade specific details (barrier levels, barrier start dates, time to maturity, etc)
- Large domain of approximation suitable for XVA



*https://azure.microsoft.com/en-us/blog/azure-gpus-with-riskfuels-technology-offer-20-million-times-faster-valuation-of-derivatives/

➢ 1,000,000+ times faster changes ***EVERYTHING***

Common OTC Trades	Valuation (V)	Risks Calculations (R)		500 Trades (V + R)
Simple (e.g. American Options)	~ 1 sec	~ 50+ calc	~ 50 sec	***~ 6 mins ****
Medium (e.g. Bermudan Swaptions)	~ 2 sec	~ 70 – 100+ calc	~ 170 sec	***~ 20 mins ****
Complex (MC simulations)	~ 5 – 15 sec	~ 100+ calc	~ 1,000 sec	***~ 2 hours ****
Riskfuel (parallel run in GPU)	25mm calc / sec	900mm calc / sec (AAD)	< 1 sec	**< 1 sec**

✓ *Real-time / Fast intraday risks as markets fly around*

✓ *Remove compute bottlenecks to vastly simplified batch runs*

- Benchmark: 1 CPU Server = 72 virtual CPUs (Standard_F72s_v2 with 36 core)

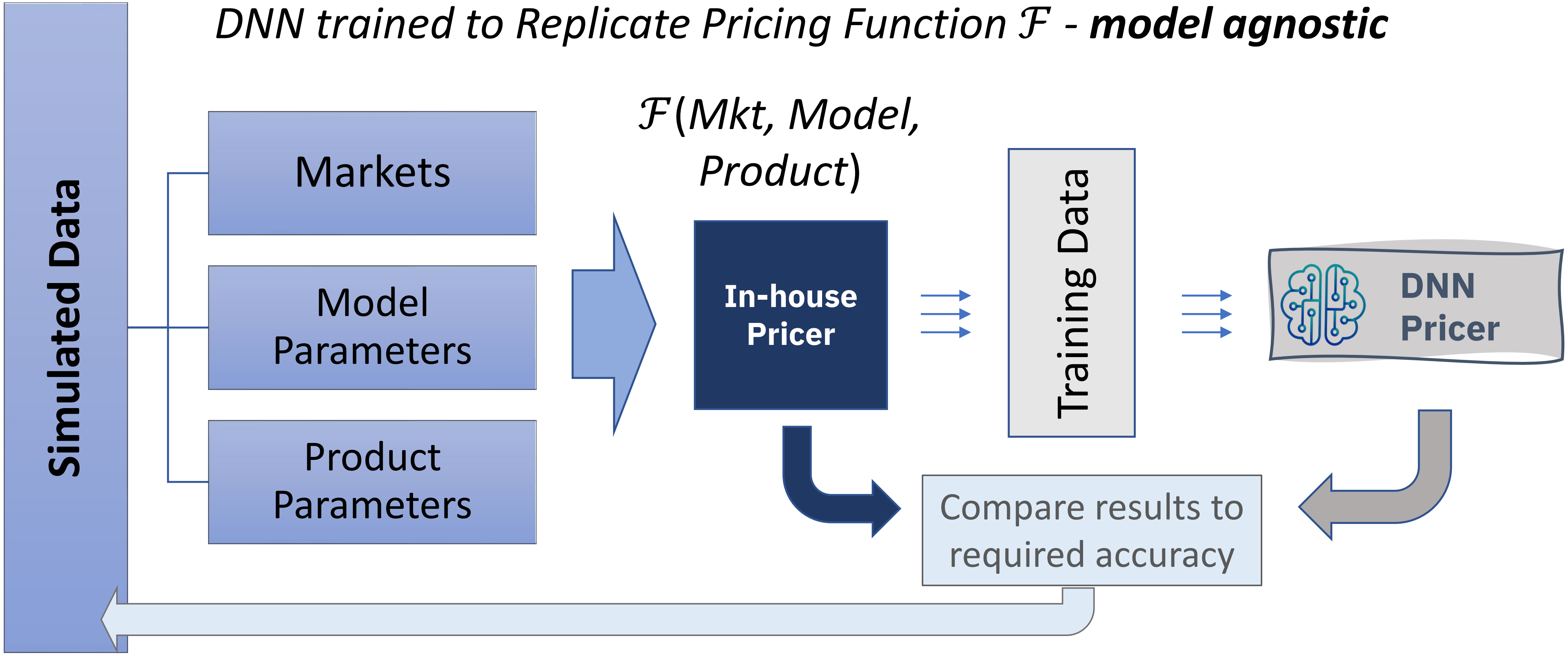
Flexibility to Deploy DNN vs Complex Pricer

- In-house Pricing Library are generally sophisticated and complex
  … and <u>a lot of works</u> to migrate or to upgrade

- The 'twinned' DNN version, however, are *<u>Simple</u>* and *<u>Standardised</u>*

Main Challenges – manage high dimensionality

DNN trained to Replicate Pricing Function $\mathcal{F}$ - ***model agnostic***

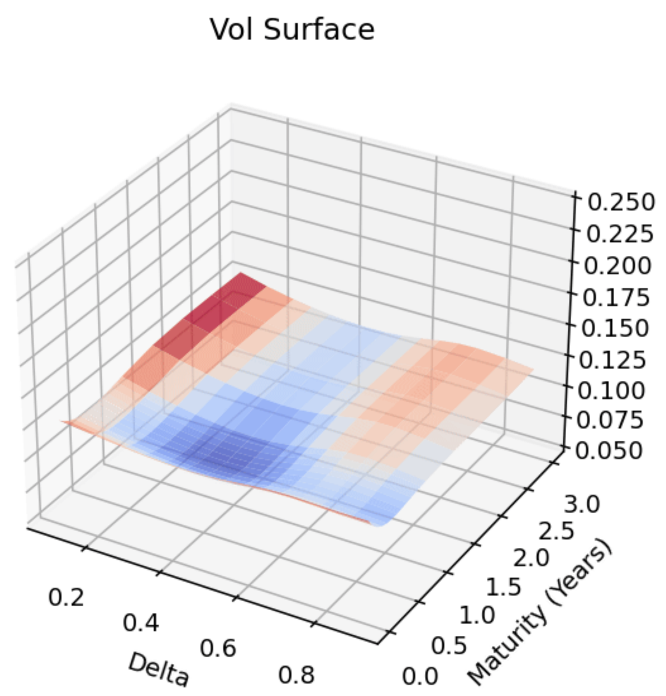


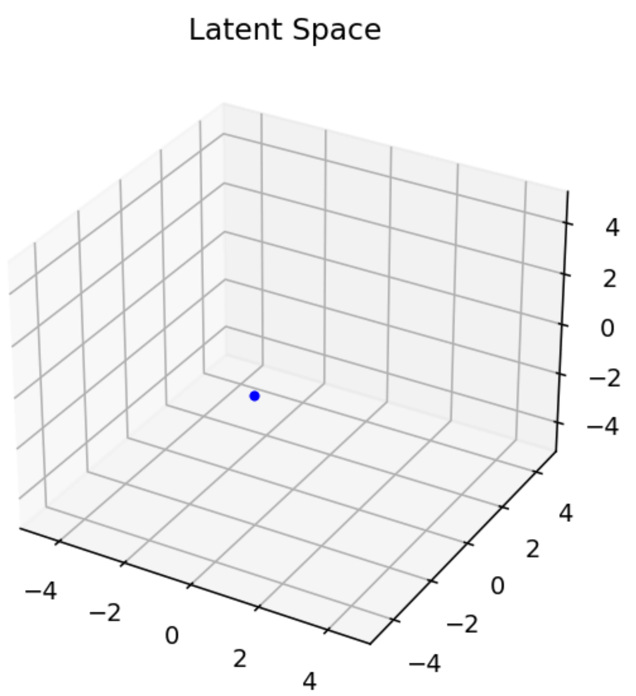
Examples:

- FX Barriers around 80 dimensions
- Bermudan Swaption around 160 dimensions
- Autocallable around 400 dimensions
- In between – Callable CMS spread, Range Accrual…

Variational autoencoders:

deep learning on volatility

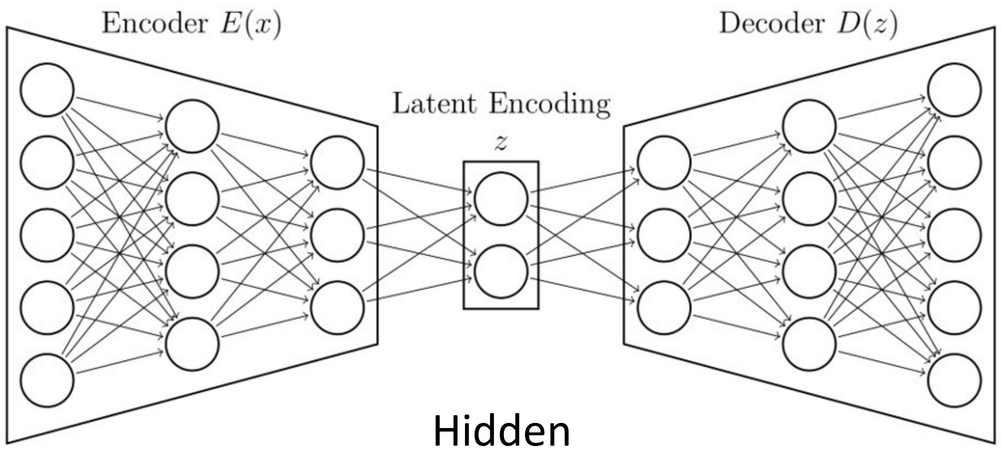




Autoencoders (AE)

An **autoencoder** is a neural network architecture capable of ***discovering structure*** within data in order to develop a ***compressed representation*** of the original input data.

Specifically, we'll design a neural network architecture such that we *impose a bottleneck in the network which forces a* ***compressed*** *knowledge representation of the original input*



Input Layer

Hidden Layers - **bottleneck**

Output Layer

- A bottleneck constrains the amount of information that can traverse the full network, forcing a learned compression of the input data
- Without the presence of an information bottleneck, our network could easily learn to simply memorize the input values by passing these values along through the network

Variational Autoencoders (VAE)

An autoencoder learns to compress and reconstruct data from an input to an output, while minimizing the difference between the original and reconstructed data

VAEs are ***generative models*** that learn to model the ***underlying distribution of the input data*** – therefore can be used to generate new data points that are similar to the original data.

Specifically VAEs learn a ***smooth and continuous latent space representation*** that can be used to generate new data points by sampling from the learned probability distribution, at the cost of slightly lower reconstruction quality.

AEs use a deterministic mapping from the input to the latent space (where their encoded vectors lie) that may not be continuous or allow easy interpolation.
It recovers the original data faithfully, but generally cannot generate new data points

Variational Autoencoders (VAE) – Concept

x is the input data that has a distribution **P(x);**
z is the latent variable which we want to learn about

$$P(x) = \int p(x|z)\,p(z)dz$$

p(x|z) is the likelihood of the data given the latent variable;
p(z) is the prior distribution over the latent variable
The likelihood P(x) tells us how to compute the distribution over the observed data **x** given hidden (latent) variable **z**.

*Or, flip it the other way round, given an observed data example **x**, we want to understand what possible values of the latent variable **z** were responsible for it*

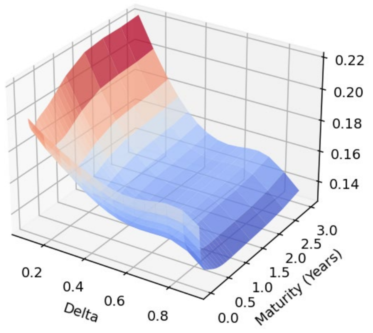
The posterior distribution $P(z|x) = \dfrac{p(x|z)p(z)}{p(x)}$

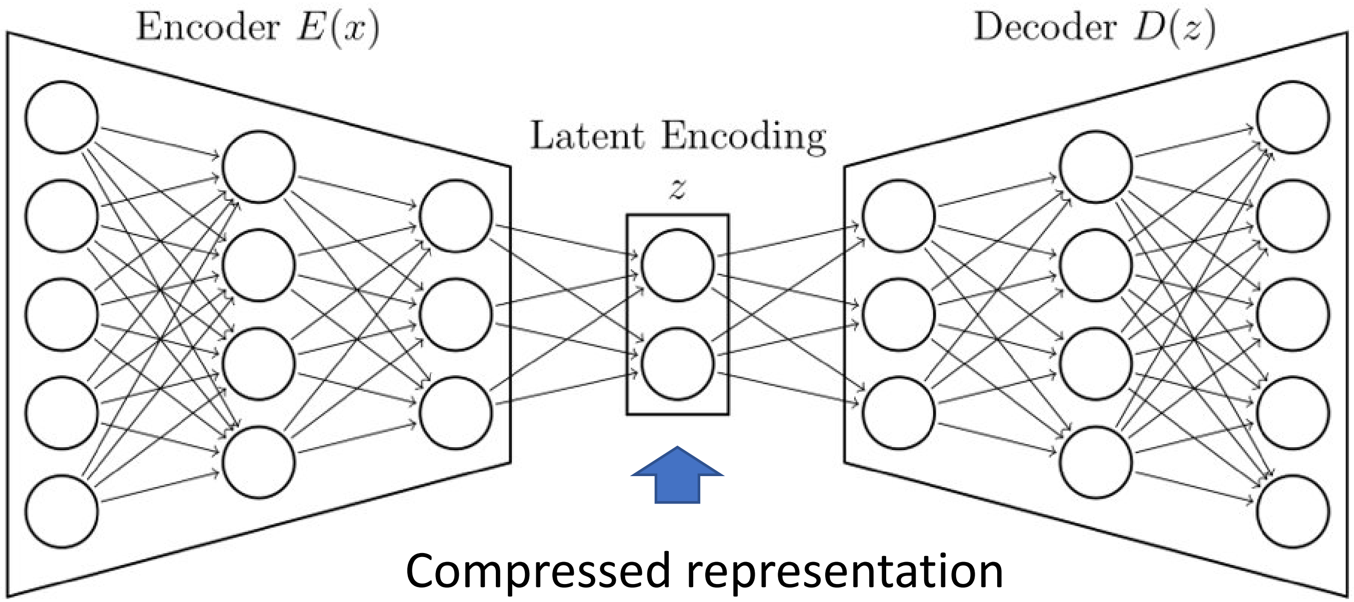
*A good explanation on the maths and the optimization of VAE can be found in Borealis AI:
https://www.borealisai.com/research-blogs/tutorial-5-variational-auto-encoders/

Variational Autoencoders (VAE)

Learn to *analyze* and *reproduce* IMAGES – in this case Volatility Surfaces

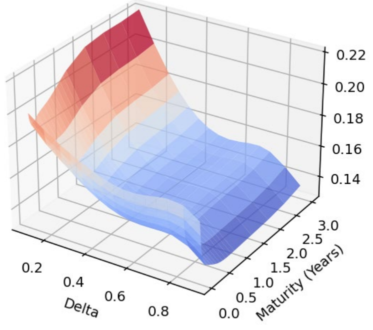
Input image





Compressed representation
of the original image

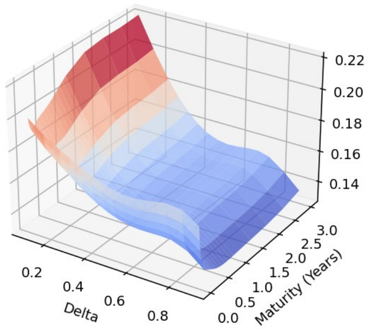
Reproduced image

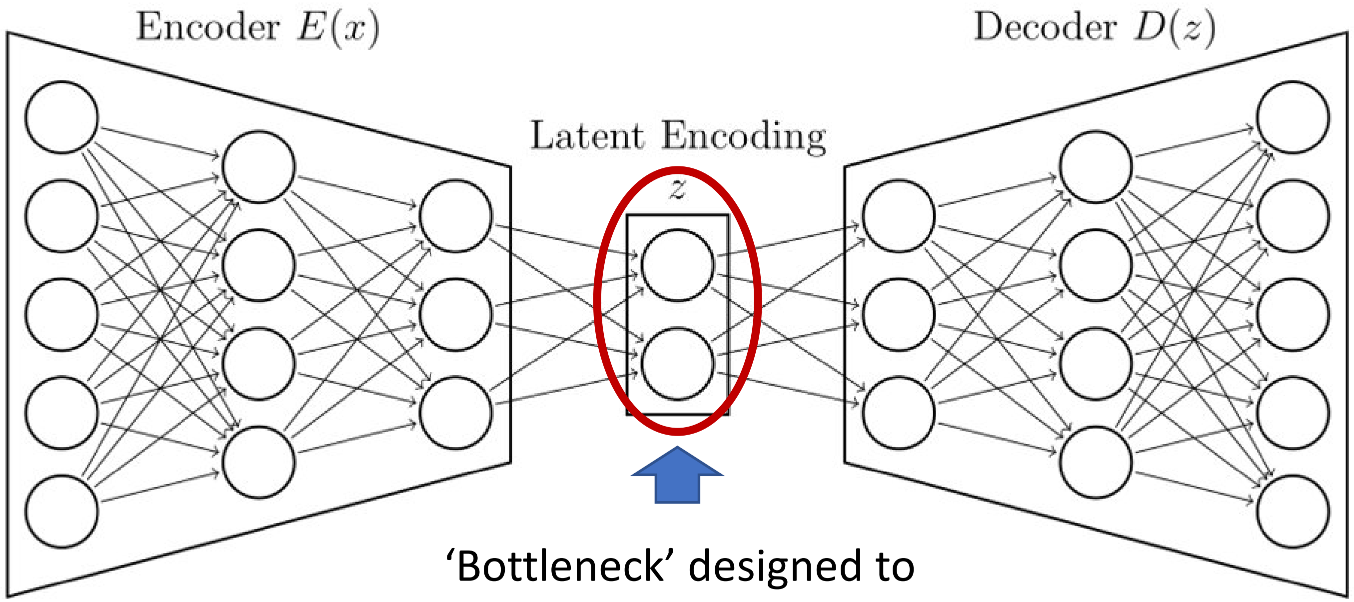


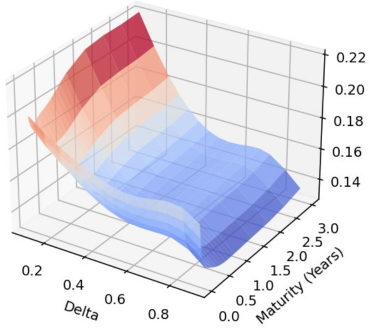
3 Components : 1. Encoder + 2. Latent Space + 3. Decoder

Variational Autoencoders (VAE)

Decompose the internal structure of input data into a few major factors







‘Bottleneck’ designed to decompose data into a few major factors

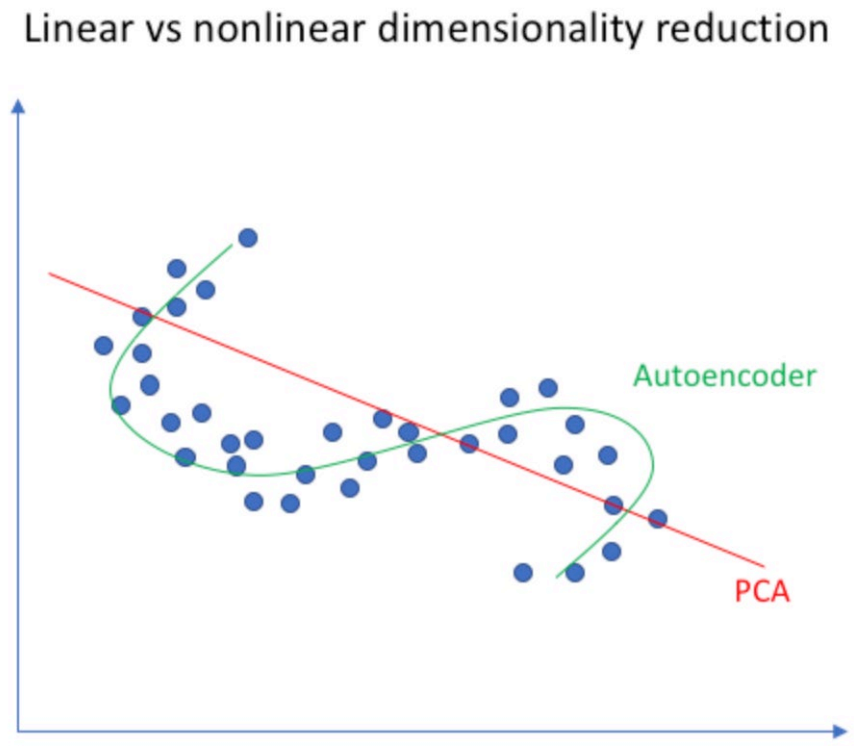
1. Encoder + 2. Latent Space + 3. Decoder

Non-Linear Factors vs Linear Factors

Because neural networks are capable of learning nonlinear relationships, this can be thought of as a more powerful (nonlinear) generalization of PCA (Principal Component Analysis).

PCA attempts to discover a lower dimensional linear relationship, which describes the original data in a reduced number of 'linear factors'.

Autoencoders are capable of learning nonlinear relationship, which describes the original data in a reduced number of 'factors' – which can be non-linear, thus do not have the limitation of PCA, and have more powerful explanatory power.



Variational Autoencoders (VAE) on Implied Vol Surfaces

Variational autoencoders: A hands-off approach to volatility. The Journal of Financial Data Science, 4(2):125–138, 2022. *Maxime Bergeron, Nicholas Fung, John Hull, Zissis Poulos, and Andreas Veneris.*

3 KEY TAKEAWAYS:

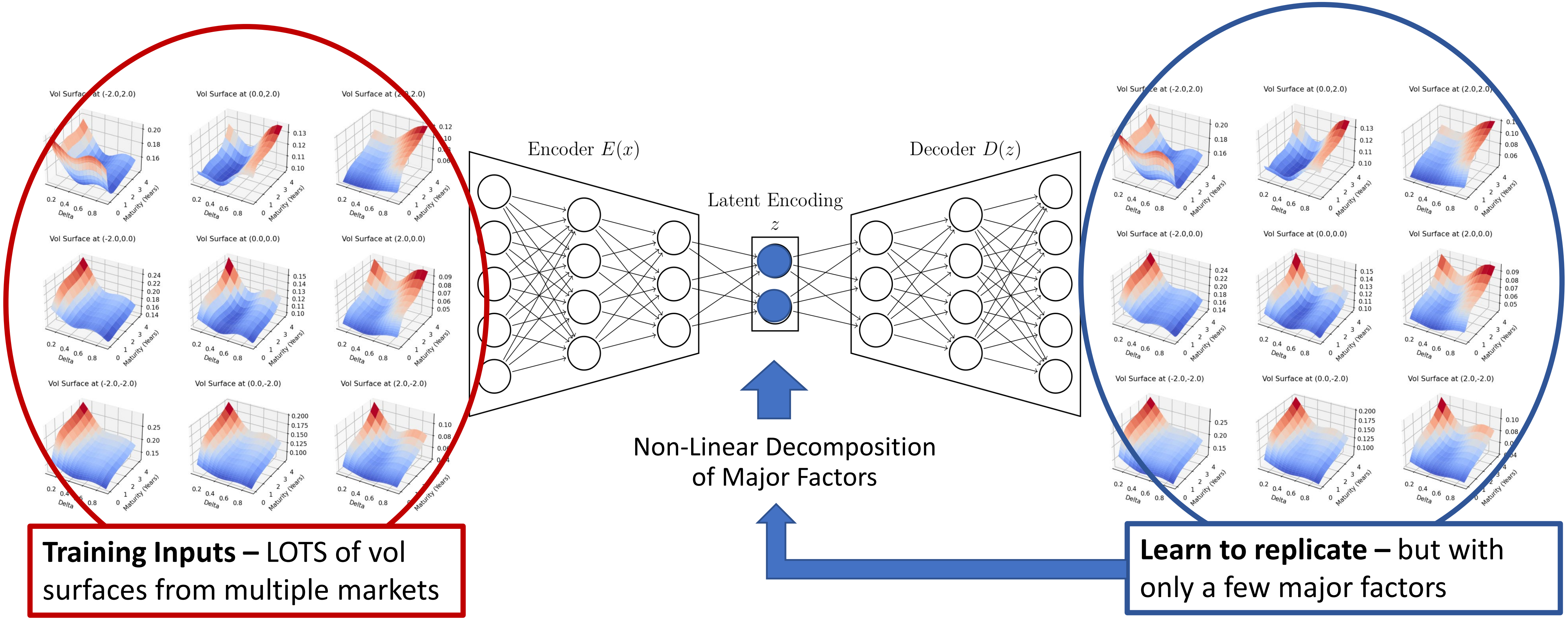
1. Show how synthetic yet realistic volatility surfaces for an asset can be generated using variational autoencoders trained on multiple assets at once.

2. Illustrate how variational autoencoders can be used to construct a complete volatility surface when only a small number of points are available - without making assumptions about the process driving the underlying asset or the shape of the surface.

3. Empirically demonstrate the approach using foreign exchange data.

Deep Learning on Vol Surfaces

- Training Data from historical data – arbitrage-free surfaces
- Cross Learning - train with multiple markets – more data points + transfer learning

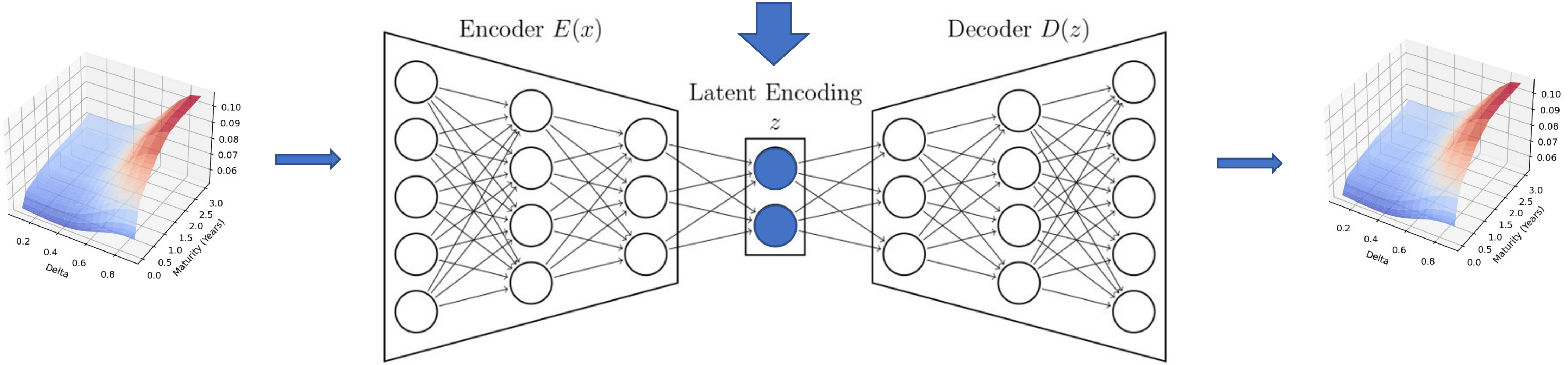


VAEs are a form of manifold learning

PCA & AE → Learn **shape of** volatility surface (compression)

VAE → Learn **shape of the space of** volatility surfaces

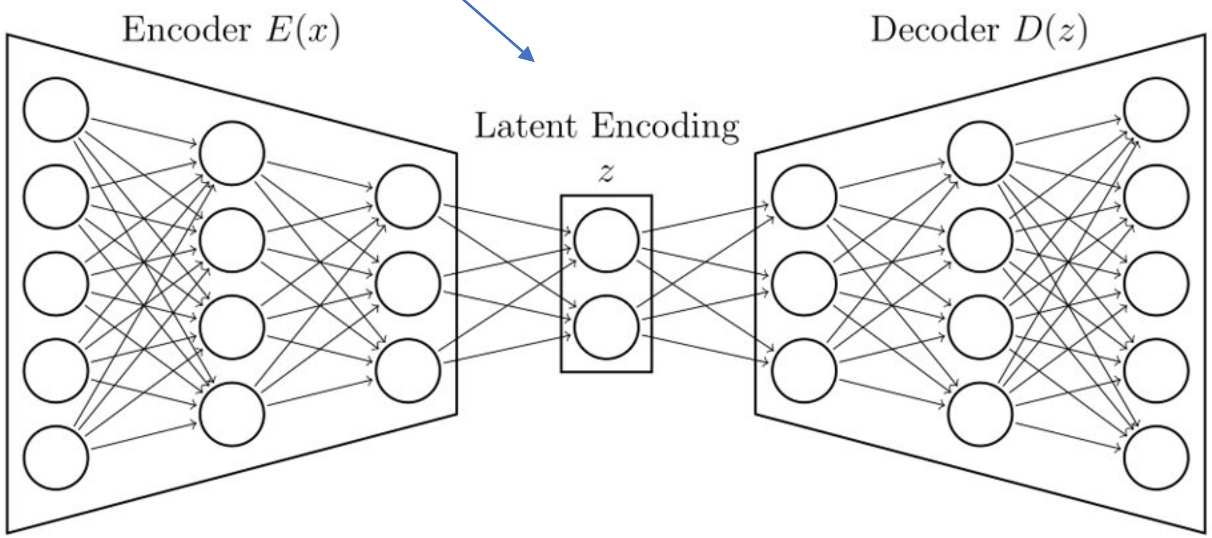
VAE latent variables are trained to encode a distribution *not just a compression*!



Therefore… an intelligent way (educated guess) to INTERPOLATE and EXTRAPOLATE from sparse data – beyond the historical patterns from its training

Case study: FX market data - **Riskfuel**

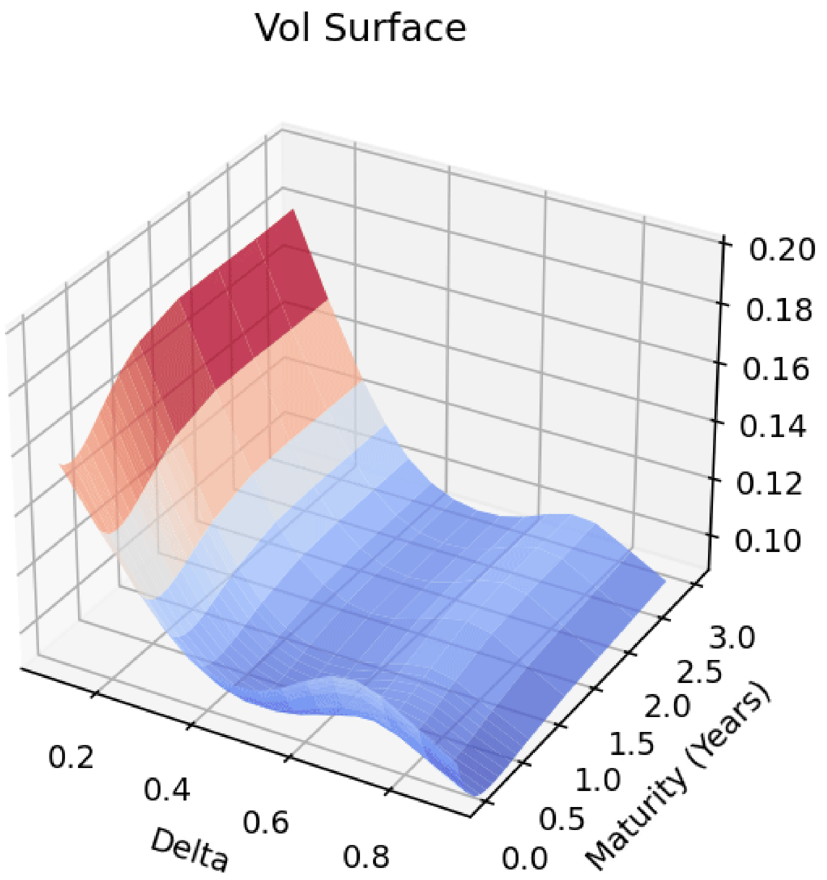
- OTC **data from 2012-2020** for five currency pairs
- Each surface = 40-point grid of 8 maturities x 5 deltas
- Data split chronologically: **set aside March – December 2020 for validation**
- Train VAE's with 2, 3 and 4 dimensional latent encodings

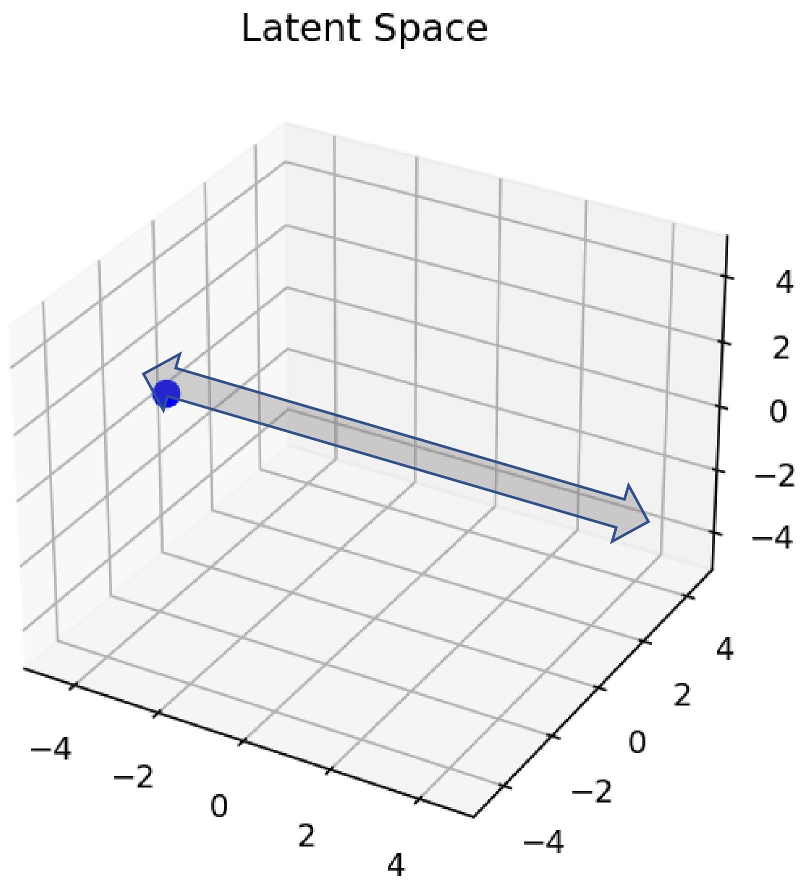


Latent Factor: Skew and Wings

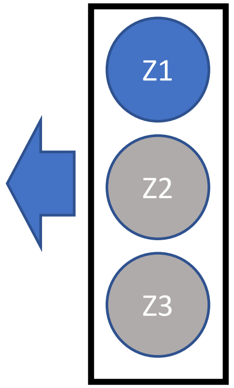
Riskfuel

Unsupervised learning…   identify ‘principal’ factors





3 latent variables

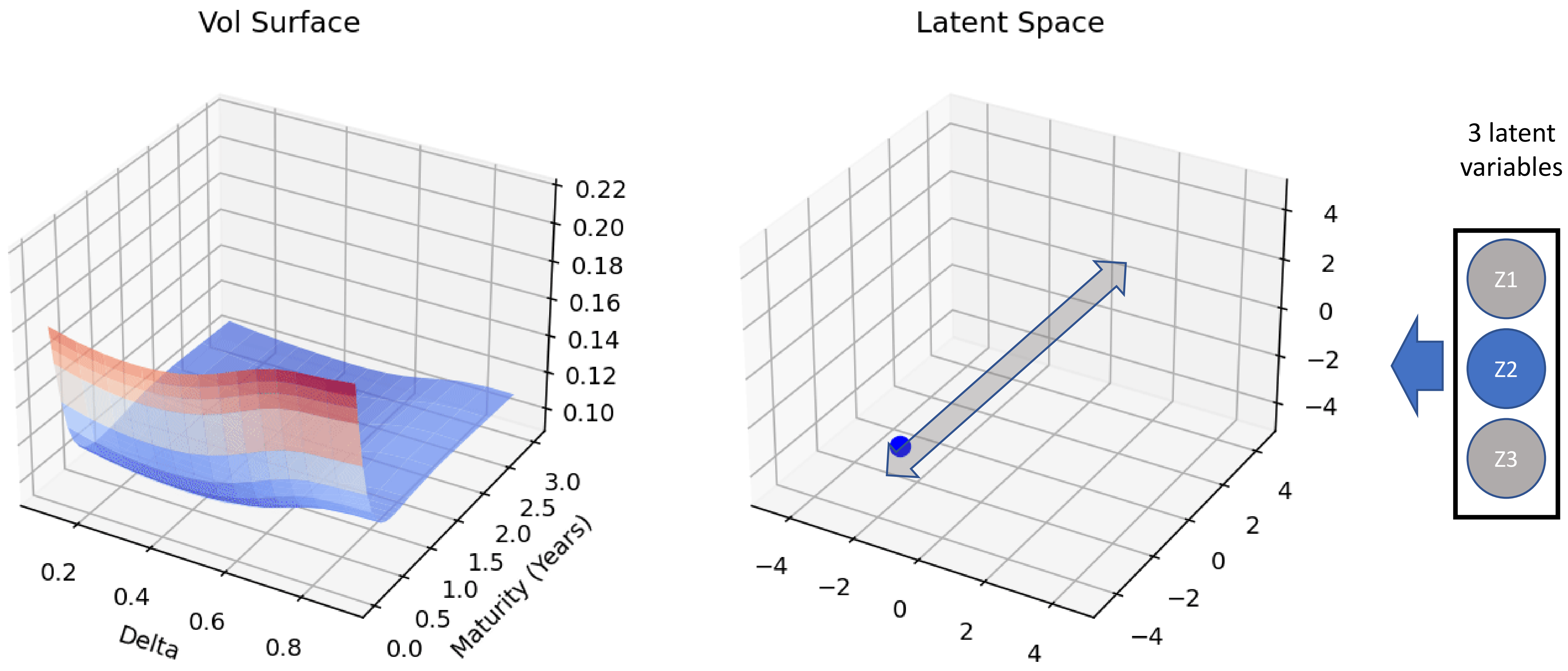


… while avoiding arbitrage conditions

Latent Factor: Term Structure

Riskfuel

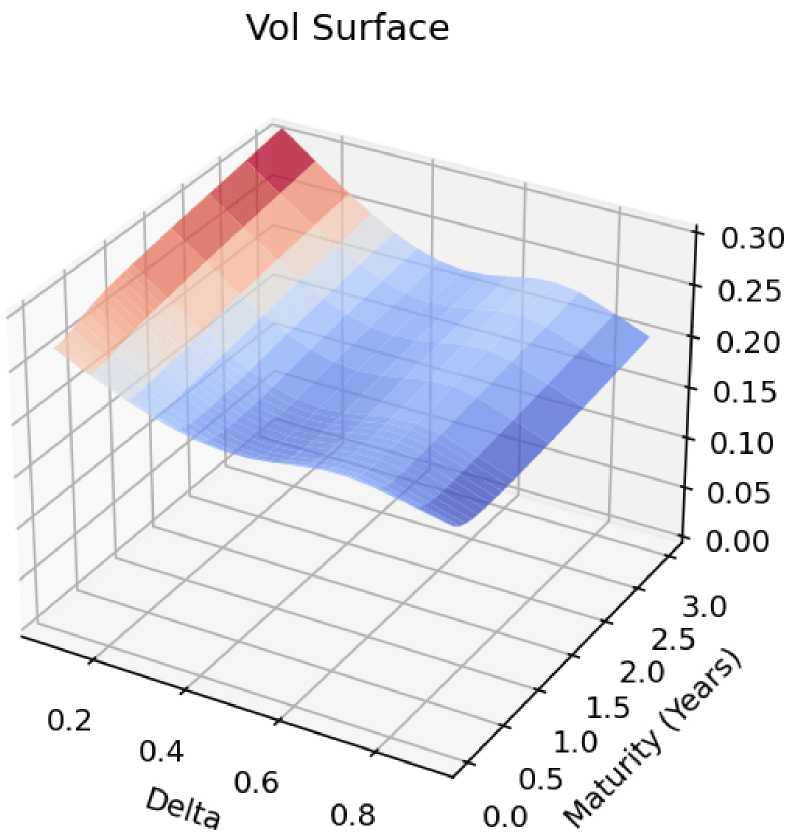
Unsupervised learning… identify 'principal' factors

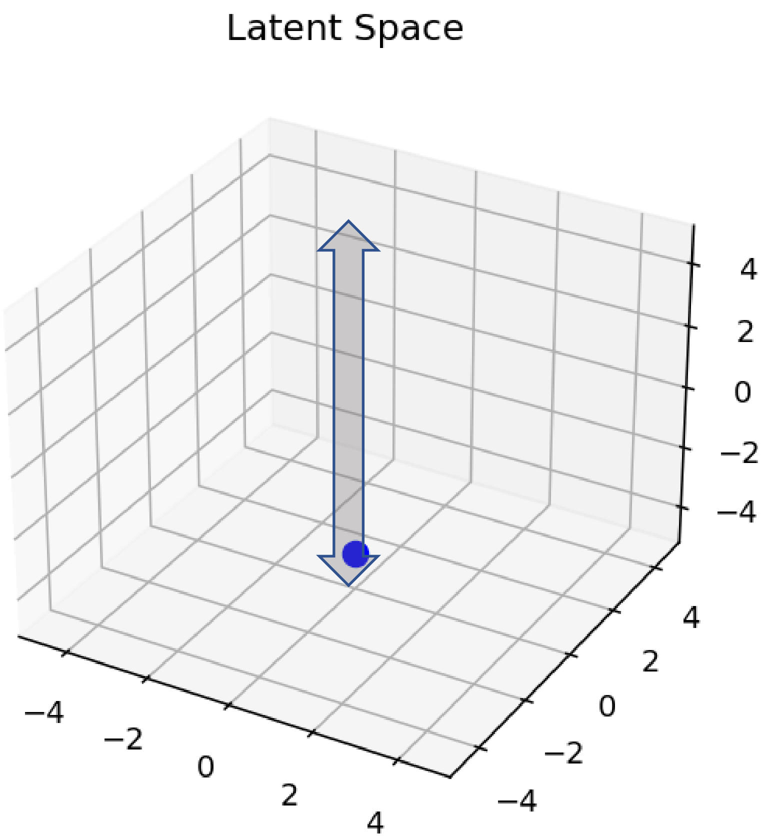


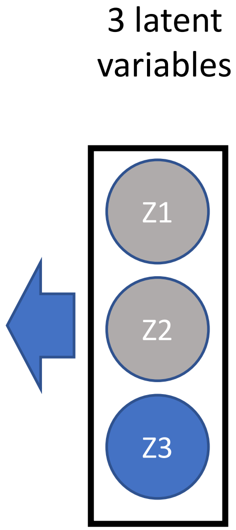
… while avoiding arbitrage conditions

Latent Factor: Volatility level

Unsupervised learning… identify 'principal' factors







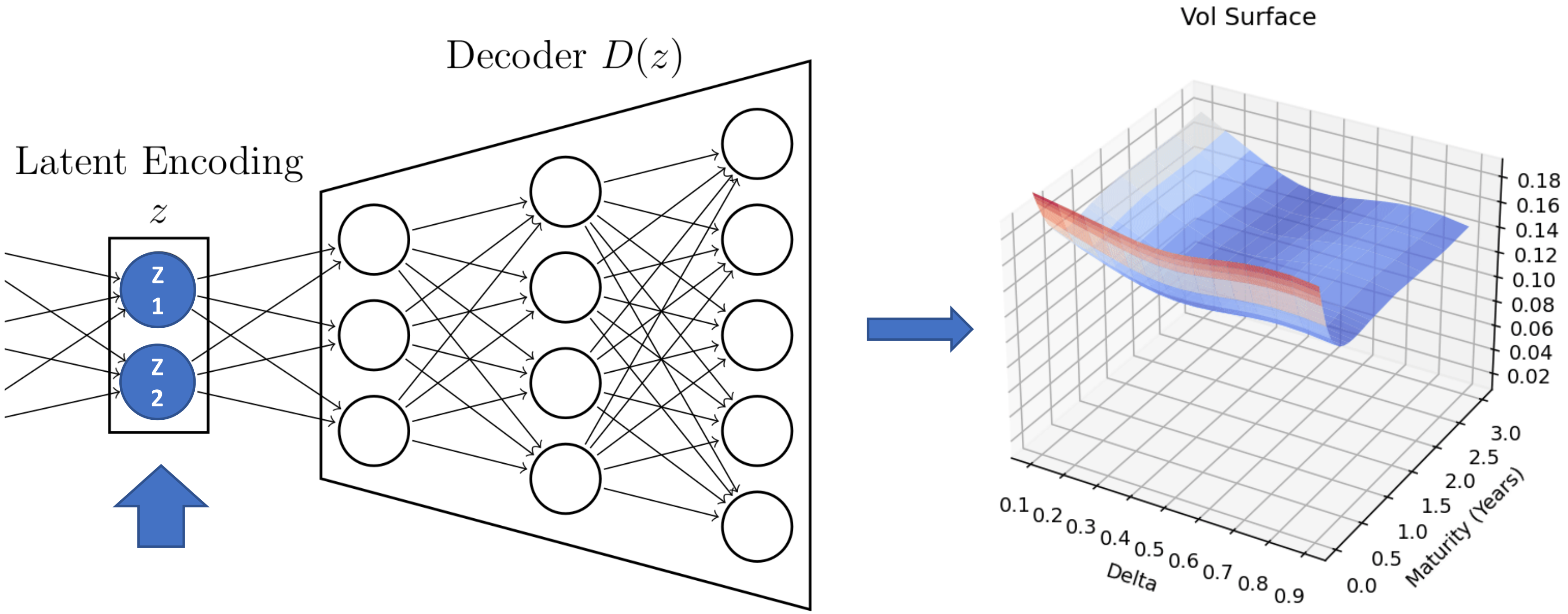
… while avoiding arbitrage conditions

Applications

Modify latent factors for synthetic surfaces

The geometry of *the space of* volatility surfaces controlled by the latent factors
without making assumptions about the process driving the underlying asset or the shape of the surface

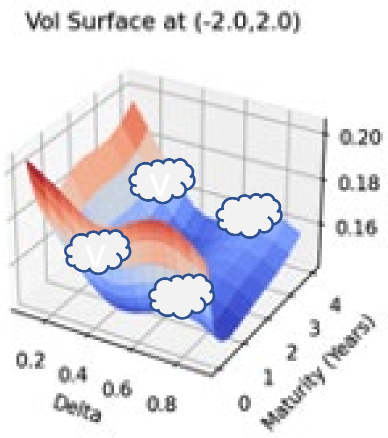
- Realistic stress testing
- Training data for fast AI models
- Generate vol surfaces not observed before

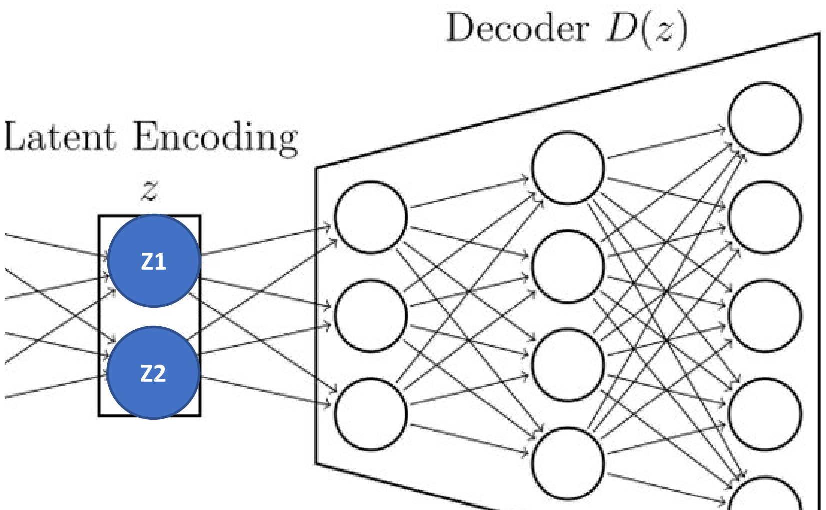


Interpolate Missing Points in Vol Surfaces

- Interpolating arbitrage-free Implied Vol surfaces from sparse data
- Find latent encodings that generate best fit surfaces – without strong model assumptions

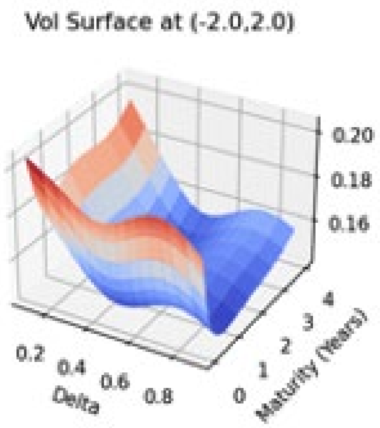
Incomplete Vol surface





Adjust Latent Factors
for best fit

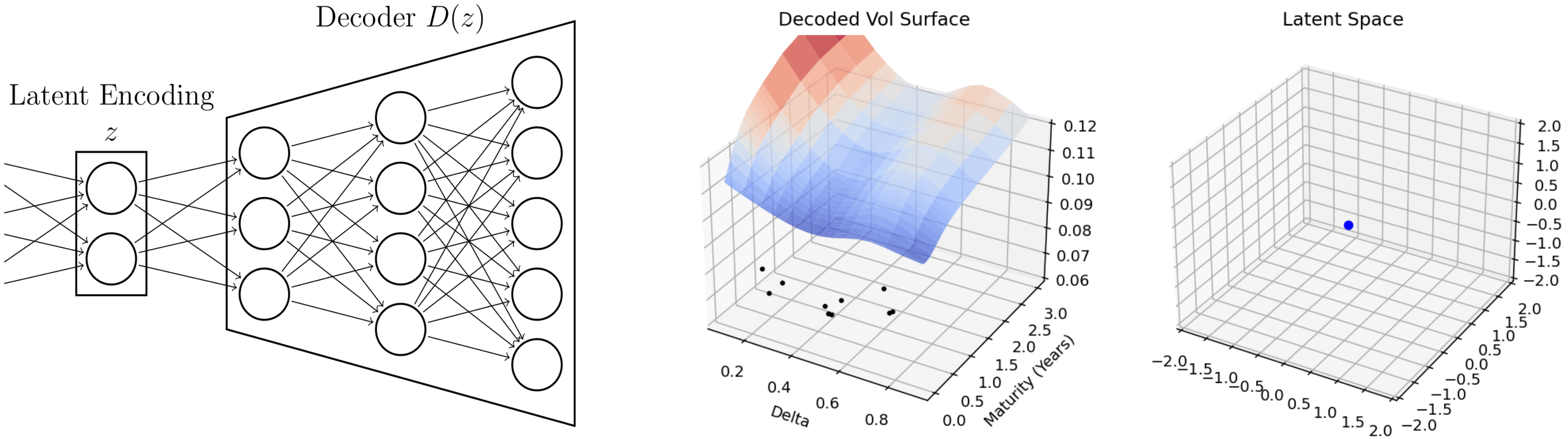
Completed Vol surface



Extrapolate Vol Surfaces from Sparse Data

- Observe subset of points on a surface (*tight deltas, short maturities*)
- Find latent encodings that generate best fit surface at these locations – without strong model assumptions

Liquid
Exchange-Traded
Short-dated
Options

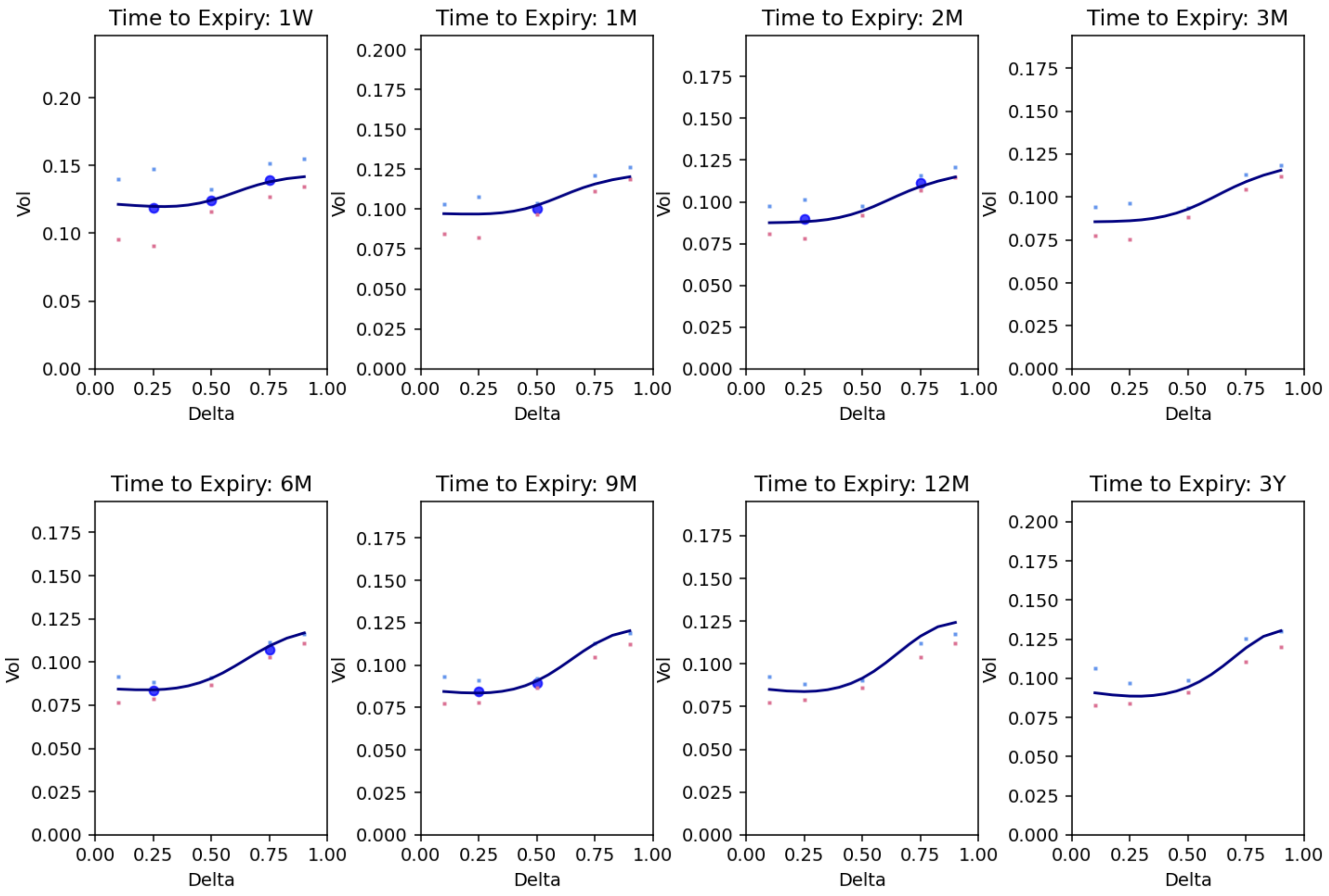


Riskfuel

Extrapolate = Interpolate and Complete

- Calibrate to market data with sparse and short-dated data points
- Recognize potential patterns to fit longer-dated vol surface

Different time slides of a vol surface



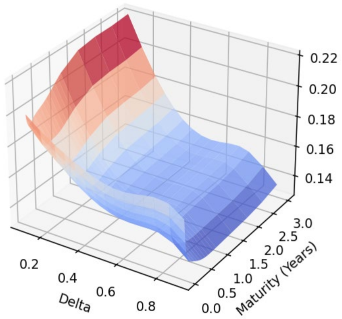
Riskfuel

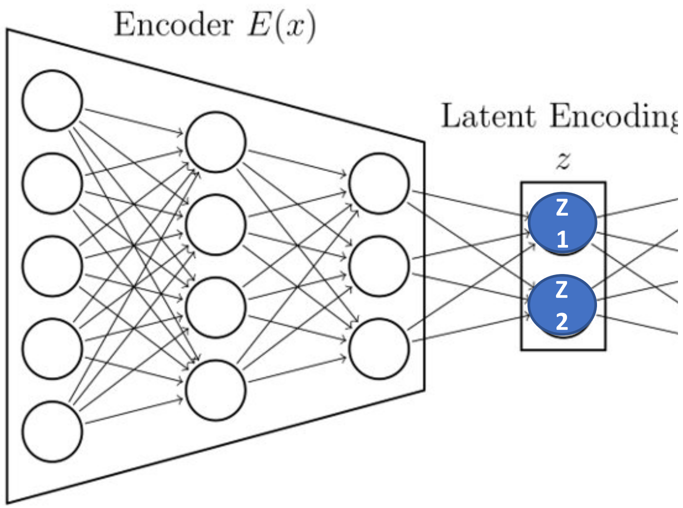
Look for Trade Opportunities or Outliers

Feed surfaces through encoder half of the **previously trained** VAE

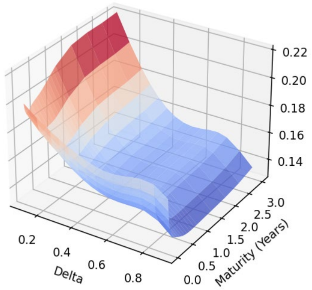
Great for detecting outliers and trade opportunities

INPUT Vol Surface





OUTPUT Vol Surface



Compare (INPUT Vol Surface , OUTPUT Vol Surface) ;

IF ≥ Tolerance – i.e. have NOT seen the shape before – THEN 

Applications

As VAE learns the shape of the space of the volatility surfaces:

- ➢ Generate realistic synthetic vol surfaces for scenario analytics
  - ✓ Including stress scenarios
- ➢ Complete whole vol surface from sparse data points
  - ✓ Illiquid markets
  - ✓ Extend exchange-traded options to longer-dated vol surfaces
- ➢ Detect outliers and trading opportunities real-time as market moves

What’s next?

How does each latent factor relate / react to other market observables?

Example: Vol Surface movement vs Spot

Some Applications:

- Find the corrected 'Adjusted Delta' in real-time at the desk, or when running risk scenarios
- Big move scenarios – market moves by say 5%, 10%, 20% etc
- Adjust pre-open vol surfaces using latest spot level
- Cross-Market trades – for example, adjust pre-US open vol surfaces from Asian close level
- P&L Attribution – by scenarios (not just sensitivities) giving traders and risk managers better insight in positions and exposure

Discussions and Questions

"We always overestimate the change that will occur in the next two years and underestimate the change that will occur in the next ten.

Don't let yourself be lulled into inaction"

- Bill Gates

Contact:

gary.wong@ArtemisAG.co.uk
– for general AI/ML/new technology applications

gary.wong@riskfuel.com
– for DNN applications

FDP Webinar

Please join us for our upcoming webinar:

Register Here: https://bit.ly/42WxJ2d

Thank You

FDP INSTITUTE®
by CAIA

Contact Us:

fdpinstitute.org

info@fdpinstitute.org

@FDPInstitute

linkedin.com/company/FDP Institute