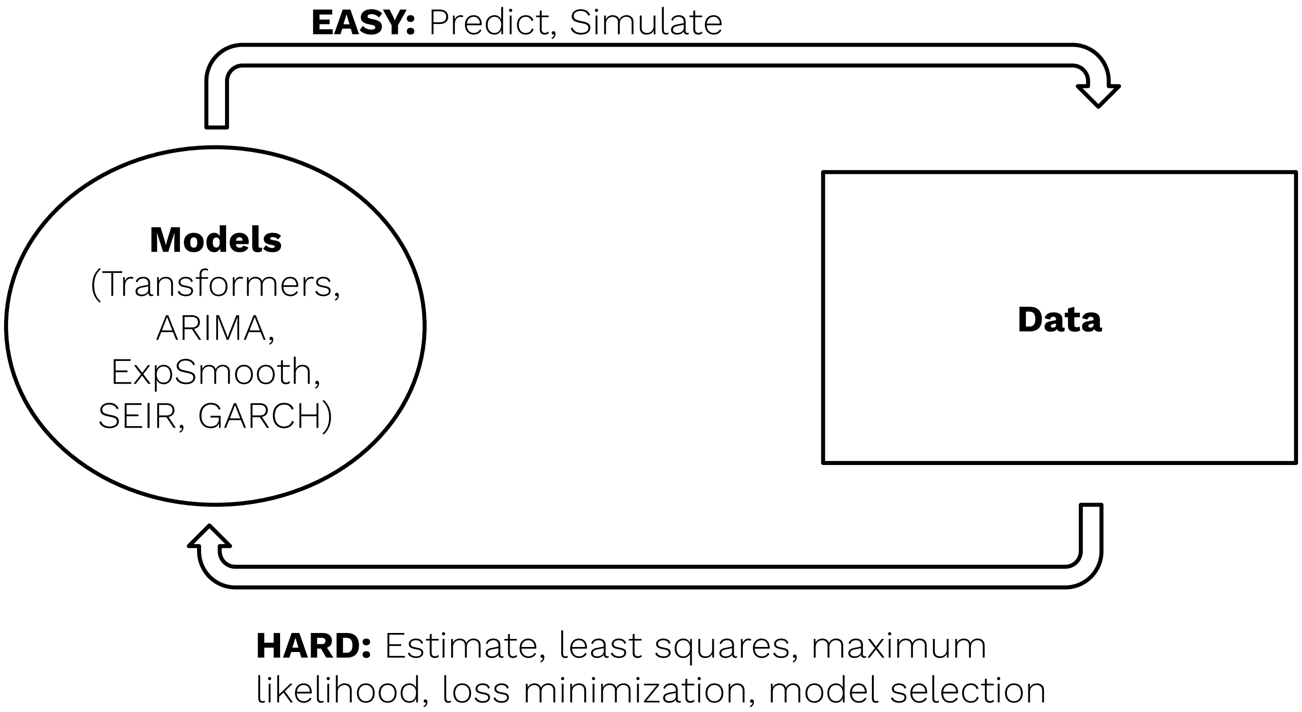
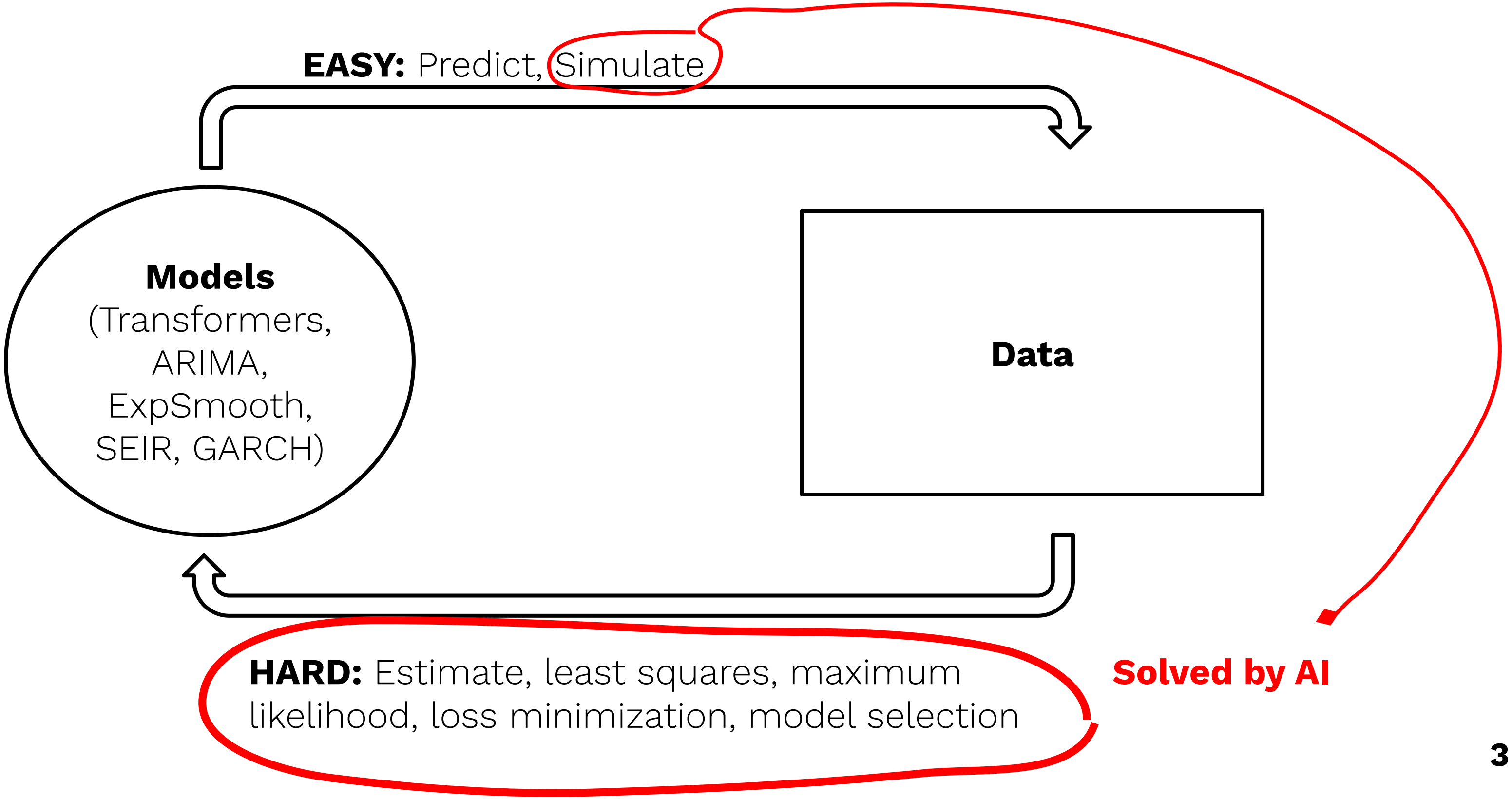
Towards **Artificial Intelligence** in Time Series **Forecasting**

Pablo Montero Manso
Marcel Schartz

THE UNIVERSITY OF SYDNEY


EASY: Predict, Simulate
Models
(Transformers,
ARIMA,
ExpSmooth,
SEIR, GARCH)
Data
HARD: Estimate, least squares, maximum
likelihood, loss minimization, model selection


EASY: Predict, Simulate
Models
(Transformers,
ARIMA,
ExpSmooth,
SEIR, GARCH)
Data
HARD: Estimate, least squares, maximum
likelihood, loss minimization, model selection
Solved by AI

“

AI algorithms outperform Statistical models *even when all assumptions of the Statistical models are met.*

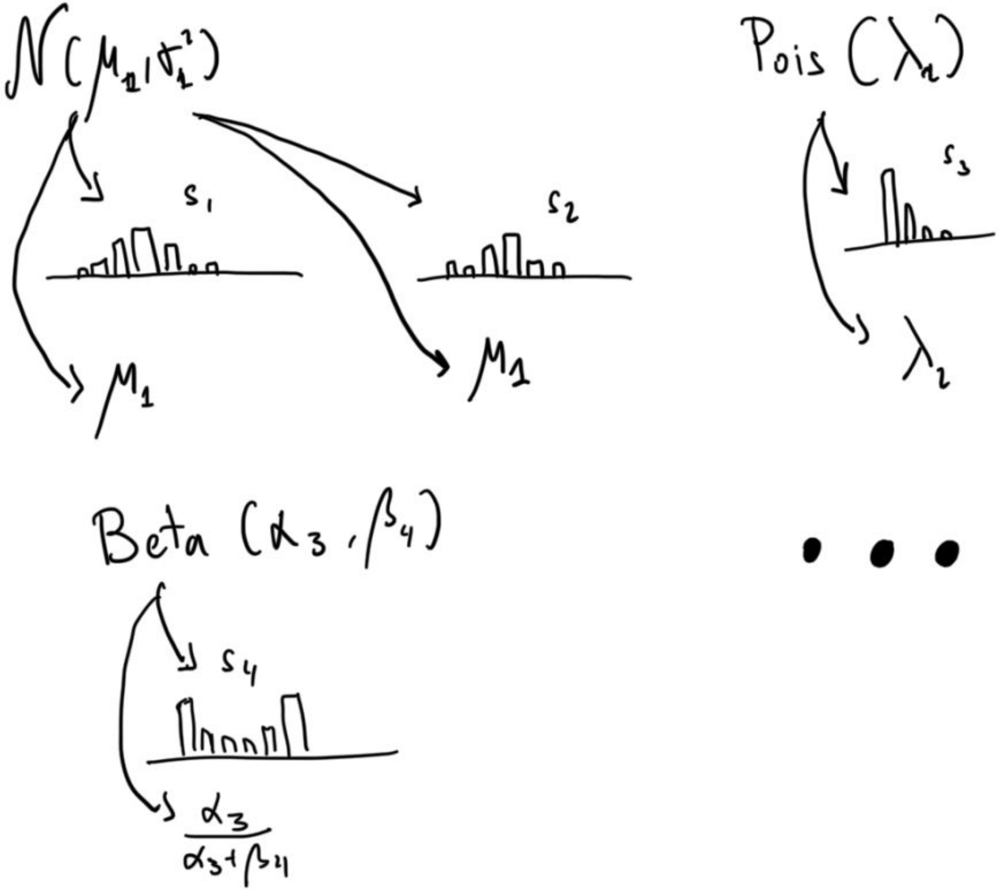
Example: Sample Mean

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

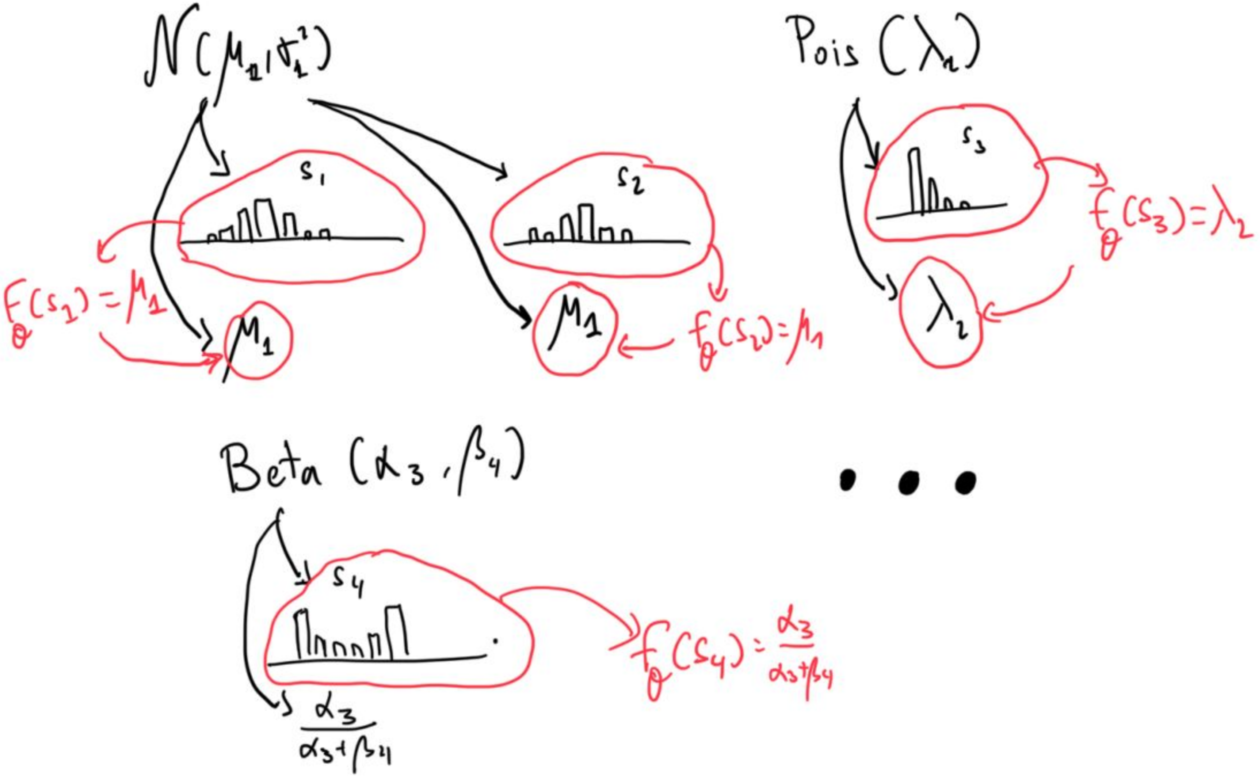
- The **sample mean** is an algorithm predicting the *population mean* $\longrightarrow$ $\frac{1}{N}\sum_{i=1}^{N} X_i$
- There are **better** algorithms, in **squared error** sense $\longrightarrow$ $\left(\mu - \frac{1}{N}\sum_{i=1}^{N} X_i\right)^2$
- These algorithms are **difficult** to design $\longrightarrow$ $\frac{0.999}{N}\sum_{i=1}^{N} X_i$

How can we use AI to create a better algorithm than the sample mean?

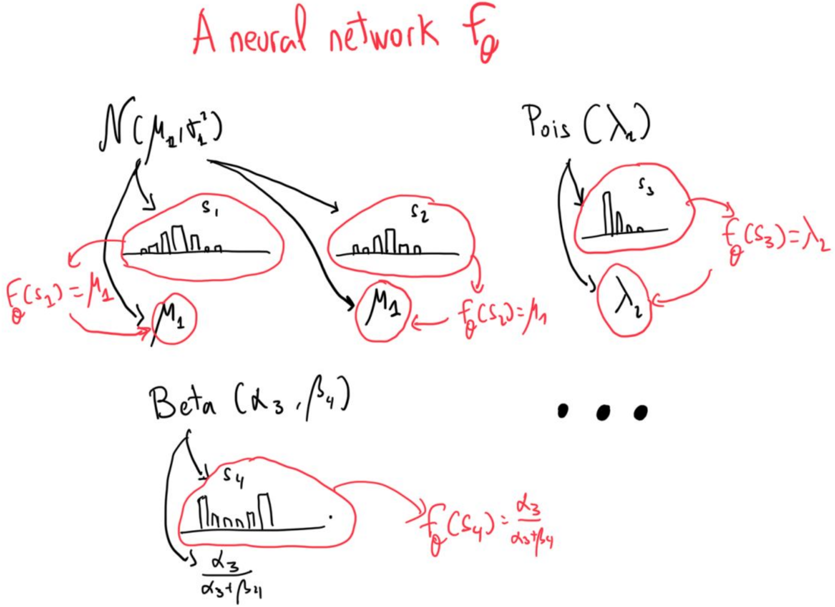
1. **Simulate** every possible scenario of ‘predicting the mean’: **(Dataset, True Mean) pairs**
2. **Train** an **AI model** to learn to **map from** the **Datasets to** the **True Means**


$\mathcal{N}(\mu_1, \sigma_1^2)$
s_1
s_2
μ_1
μ_1
$\text{Pois}(\lambda_2)$
s_3
λ_2
$\text{Beta}(\alpha_3, \beta_4)$
s_4
$\frac{\alpha_3}{\alpha_3+\beta_4}$
...

A neural network f_θ



Analogy with self-play in AlphaGo Zero



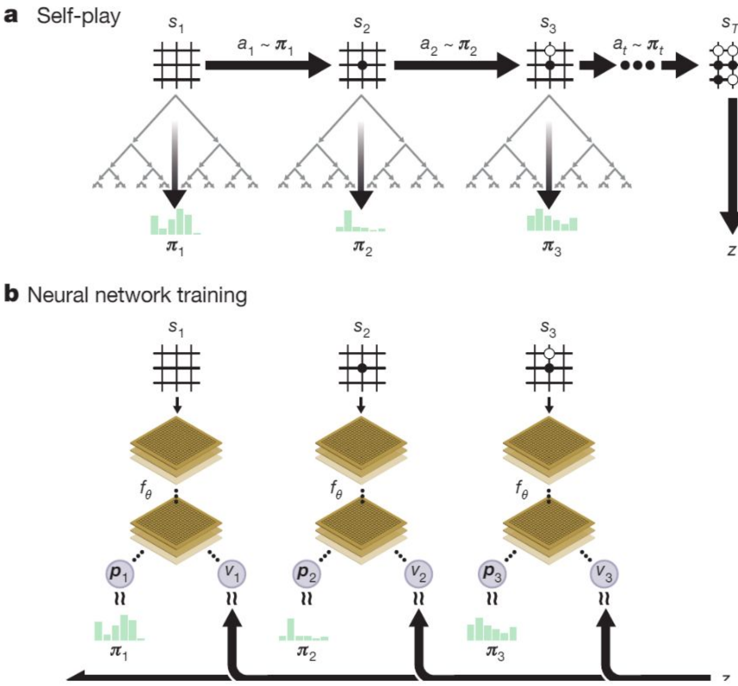


Figure 1 | Self-play reinforcement learning in AlphaGo Zero. a, The

Generate a diverse set of time series, then train an AI to recognize them.

This AI will outperform almost everything else, almost always

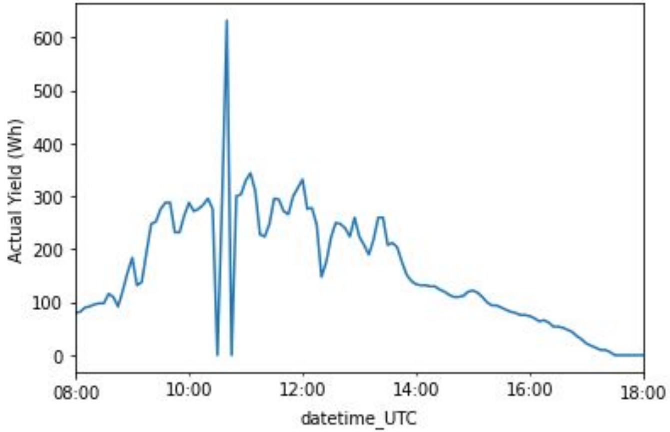
Why this idea is so powerful

1. **Outperforms** classic algorithms under their ideal conditions.
2. **Trivial to extend** to new conditions, for which the classic algorithm does not exist and is very hard to create.
   - ☐ Outliers
   - ☐ Structural Breaks
   - ☐ Noise distributions
   - ☐ Cost Functions

Outliers

- Faulty sensors, one-off events
- **No** good **algorithm**
- We apply **crude heuristics**

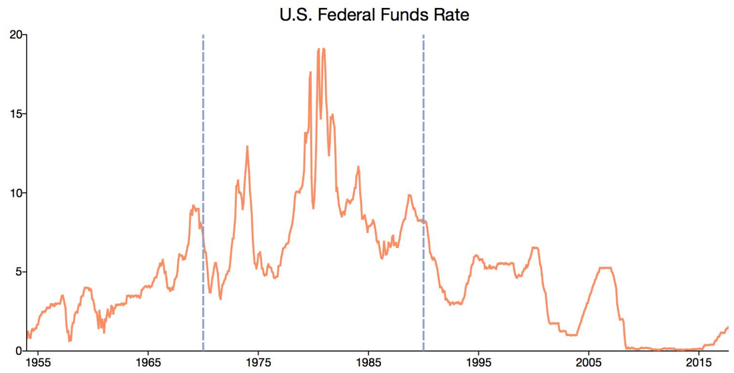
- **However,** suggesting a good 'true model' is very easy
- **Example:** Force values to 0 with some prob.



https://stackoverflow.com/questions/62473007/modify-outliers-caused-by-sensor-failures-in-timeseries-data

Structural Breaks

- Policy change, Metric change, Disrupting new product in the market
- Concept is valid, algorithms are crude (*Hyp. Tests in 2023?*)
- However, trivial to pose some good underlying models



https://www.aptech.com/structural-breaks/

GPU Radeon RX580 Price and Crypto in 2017

https://www.quora.com/Why-are-GPUs-so-expensive-2

$$\begin{cases} y_{t+1} = \phi_1 y_t + \phi_2 y_{t-1} & t \leq T_1 \\ y_{t+1} = \alpha_1 y_t + \alpha_2 y_{t-1} & t > T_1 \end{cases}$$

Current Algorithm A:

1. Find the break points via hyp test
2. Fit a model to each

Current Algorithm B:

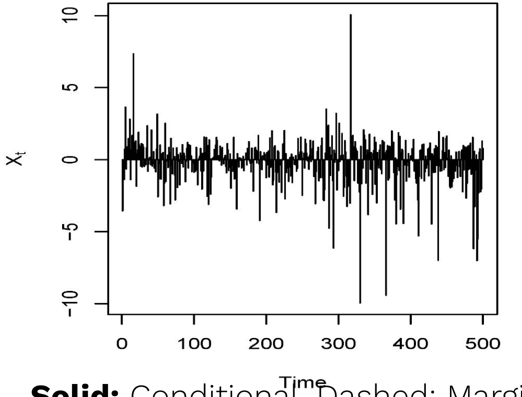
1. Suppose R break points
2. Find the optimal partitioning given a model via Dynamic Programming

AI:

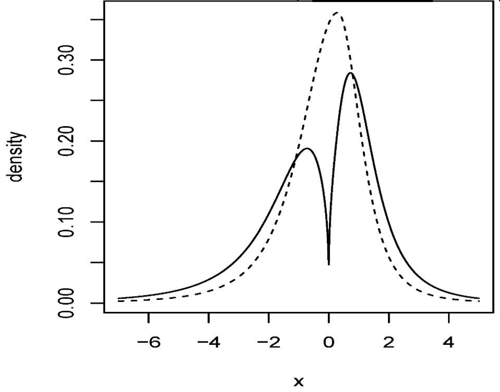
1. Generate a bunch of potential scenarios combining number of breaks and model families
2. Train AI on those scenarios

Exotic Distributions

- Mixture of distributions in the conditional errors
- *What is a good algorithm for that?*

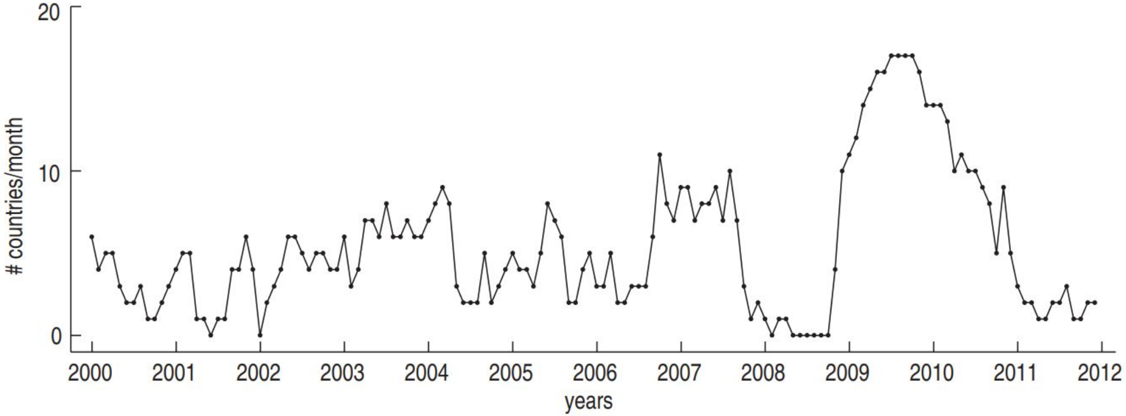


Solid: Conditional, Dashed: Marginal



J. McNeil, A. (2021). Modelling volatile time series with v-transforms and copulas. *Risks*, *9*(1), 14.

Count of countries with less than 2% inflation



Scotto, M. G., Weiss, C. H., & Gouveia, S. (2015). Thinning-based models in the analysis of integer-valued time series: a review. *Statistical Modelling*, *15*(6), 590-618.

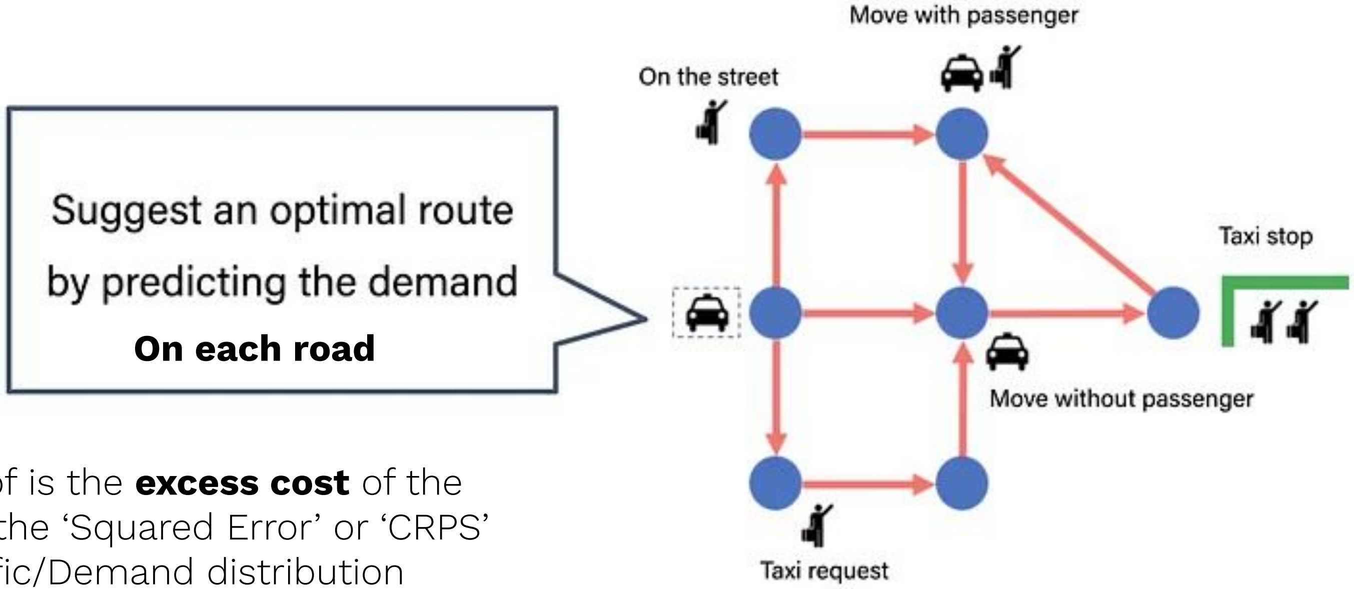
- Some months **Poisson**, others, **Negative Binomial**, throw in some **Zero-Inflation**...
- Solved by INAR(**1**) or Generalized INAR(**1**) process, Thinning Operators...
- Hard to fit, easy to generate

Realistic Objective: Predict-Then-Optimize

- **Forecasts are not used in a vacuum**, they are part of a decision-making process
- A good forecast in the ‘Squared Error’ sense might be suboptimal
- Difficult

Propagate Information of Cost to Train Model

Predict Traffic, Demand -> Simulate -> Create Route, Measure the Cost



The error of is the **excess cost** of the route, not the 'Squared Error' or 'CRPS' of the Traffic/Demand distribution

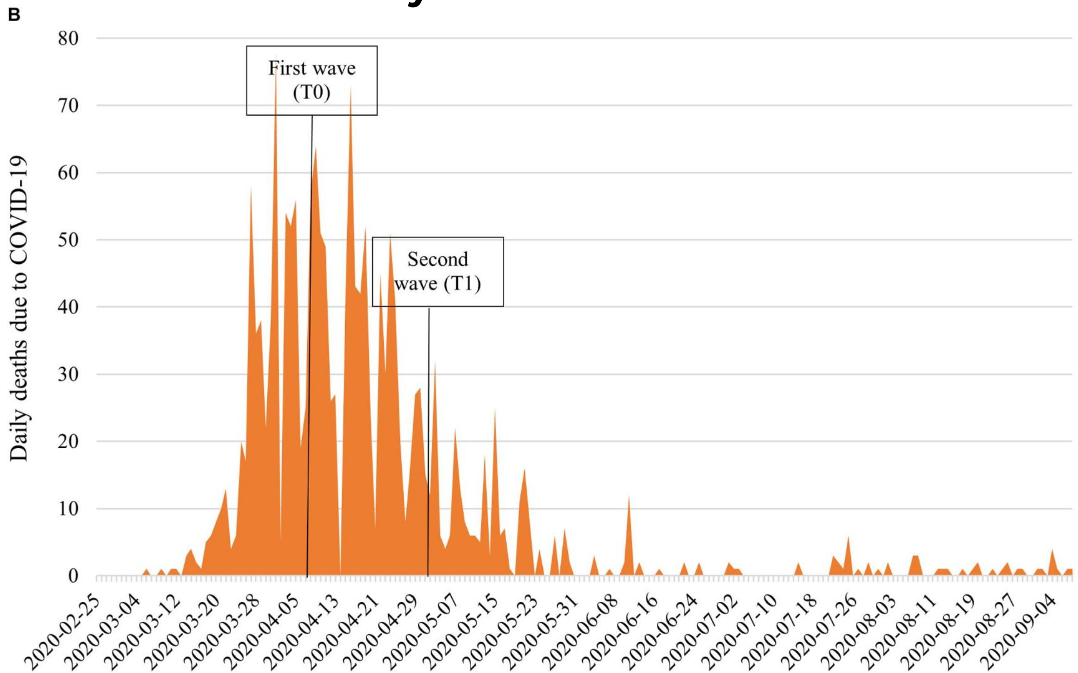
https://medium.com/optuna/parallel-hyperparameter-tuning-with-optuna-and-kubeflow-pipelines-4ef05ce614ae

All together: COVID-19 Forecasting

- Epidemiological curve + weekly effects + lockdown + vaccination + measurement changes (PCR) + variants
- Use it in a **Optimal Hospital Resource allocation** (Beds, Staff, Respirators,...)

Garcia-Vicuña, D., Esparza, L., & Mallor, F. (2022). Hospital preparedness during epidemics using simulation: the case of COVID-19. *Central European journal of operations research*, 1-37.

Daily Covid Deaths Switzerland



Time Series is different from other AI-dominated fields: **The Old and The New**

- Computer Vision, Natural Language Processing are tasks with no foundational scientific principles
- Physics, Signal Processing, Epidemiology, Economics have convincing ‘true’ models
- Even Computer Vision exploits known theoretical properties, such as rotation invariance (a flipped picture of a cat is still a cat). NLP that learns to do Math.

Proposed Model and Experiments

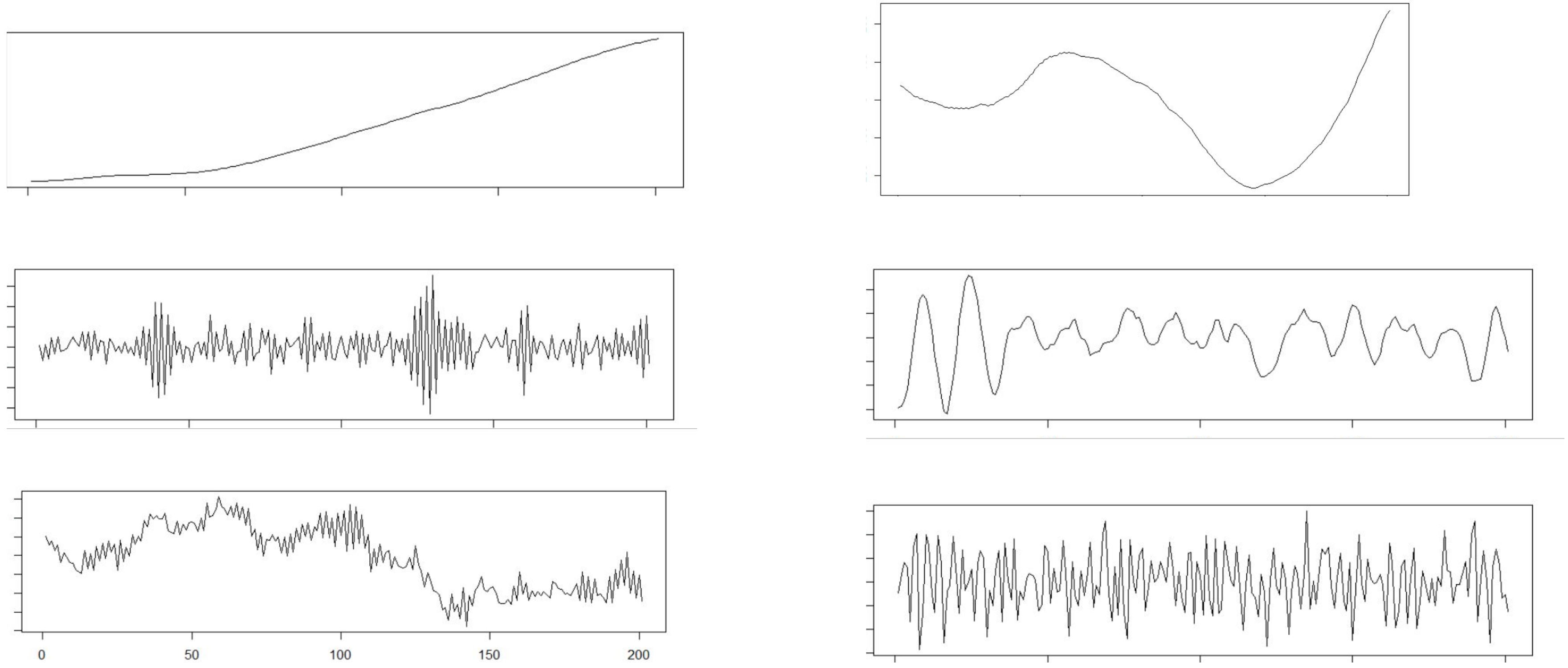
Experiments

1. **Simulate** time series under some **known conditions (scenarios).** As diverse as possible
2. **Train an AI** model **on these simulations**
3. Test the AI against the reference methods for those known conditions
4. Test performance of the AI in **real data**

Time Series Processes

- **AR:** The linear autoregressive family
  - Contains many realistic patterns: periodic, trends, etc.
  - Includes models such as Exponential smoothing
- **GARCH:** Used in finance, **very difficult** to estimate
- Both AR and GARCH have good reference algorithms

Diversity of Linear Processes



Reference Algorithms for AR processes

- Maximum Likelihood
- Yule-Walker
- Burg
- Ordinary Least Squares (not equal to MLE in AR proc.)

Reference GARCH algorithm

GARCH(p, q) model specification [edit]

The lag length p of a GARCH(p, q) process is established in three steps:

1. Estimate the best fitting AR(q) model

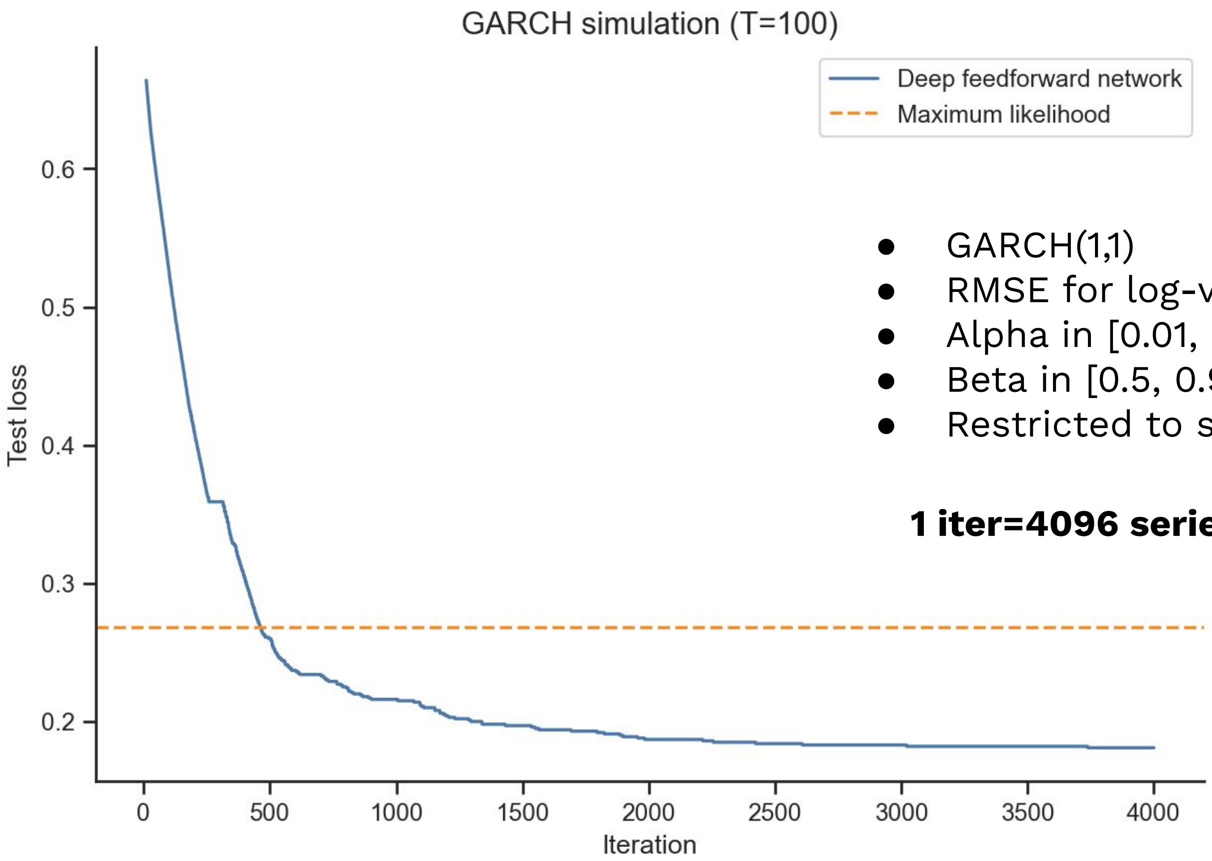
$$y_t = a_0 + a_1 y_{t-1} + \cdots + a_q y_{t-q} + \epsilon_t = a_0 + \sum_{i=1}^{q} a_i y_{t-i} + \epsilon_t .$$

2. Compute and plot the autocorrelations of ϵ^2 by

$$\rho = \frac{\sum_{t=i+1}^{T} (\hat{\epsilon}_t^2 - \hat{\sigma}_t^2)(\hat{\epsilon}_{t-1}^2 - \hat{\sigma}_{t-1}^2)}{\sum_{t=1}^{T} (\hat{\epsilon}_t^2 - \hat{\sigma}_t^2)^2}$$

3. The asymptotic, that is for large samples, standard deviation of $\rho(i)$ is $1/\sqrt{T}$. Individual values that are larger than this indicate GARCH errors. To estimate the total number of lags, use the Ljung–Box test until the value of these are less than, say, 10% significant. The Ljung–Box Q-statistic follows χ^2 distribution with n degrees of freedom if the squared residuals ϵ_t^2 are uncorrelated. It is recommended to consider up to T/4 values of n. The null hypothesis states that there are no ARCH or GARCH errors. Rejecting the null thus means that such errors exist in the conditional variance.

RESULTS on Volatility forecasting (GARCH)



- GARCH(1,1)
- RMSE for log-volatility
- Alpha in [0.01, 0.5]
- Beta in [0.5, 0.99]
- Restricted to stationary

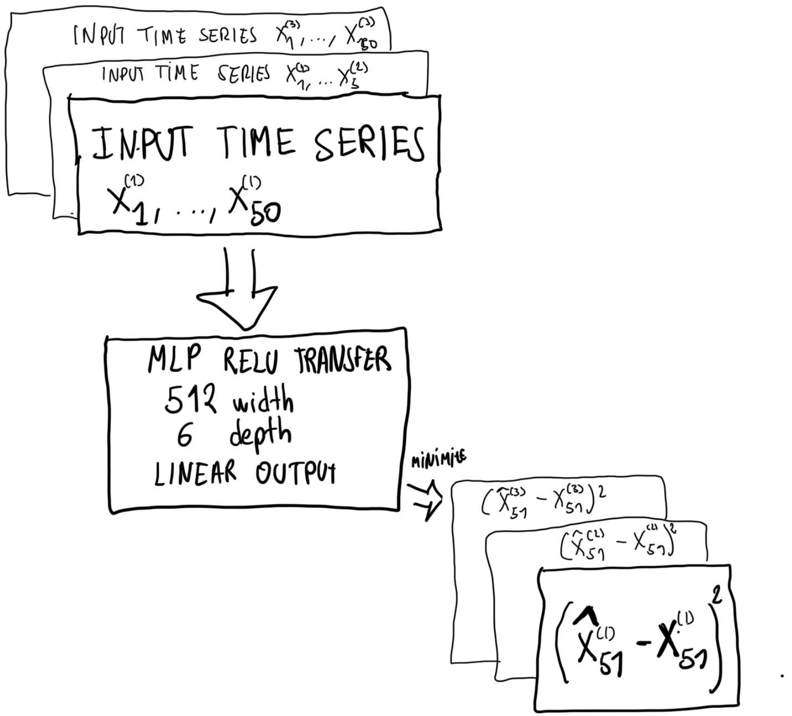
1 iter=4096 series

AR Simulation Scenario

Generate Time series of AR processes

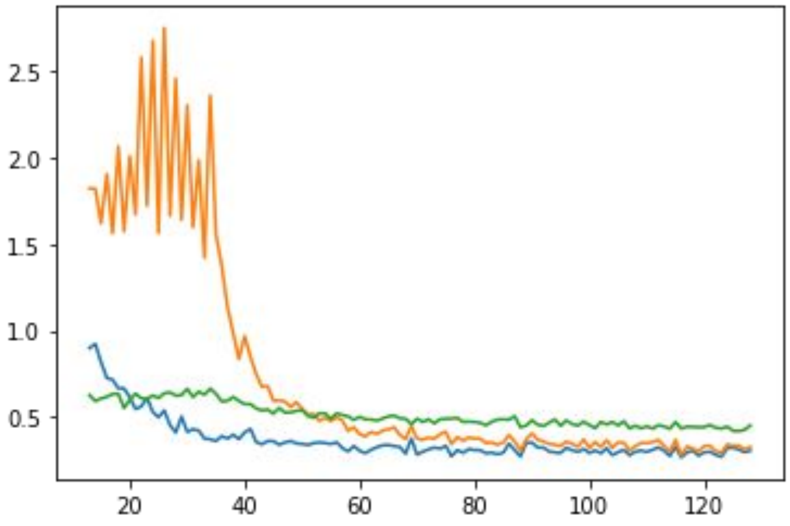
- All lengths from 1 to 125
- All Orders from 1 to 15
- All possible coefficients by sampling from the PACF from -1 to 1

AI: Neural Network Architecture



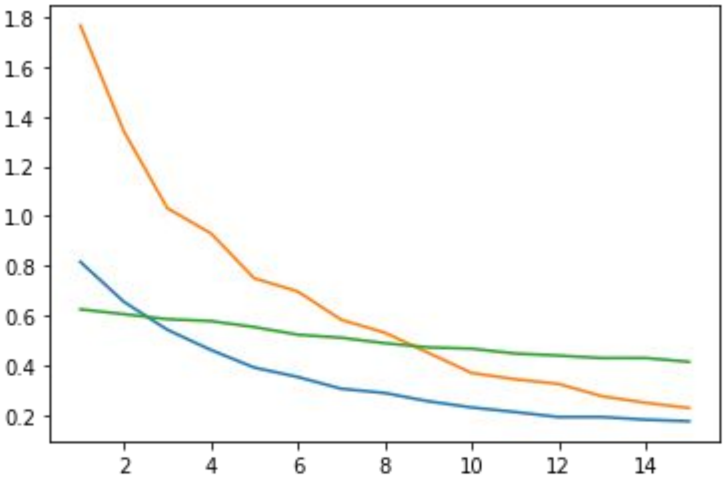
Results at
~ 0.8 Trillion Samples

Squared error vs Length of the Series

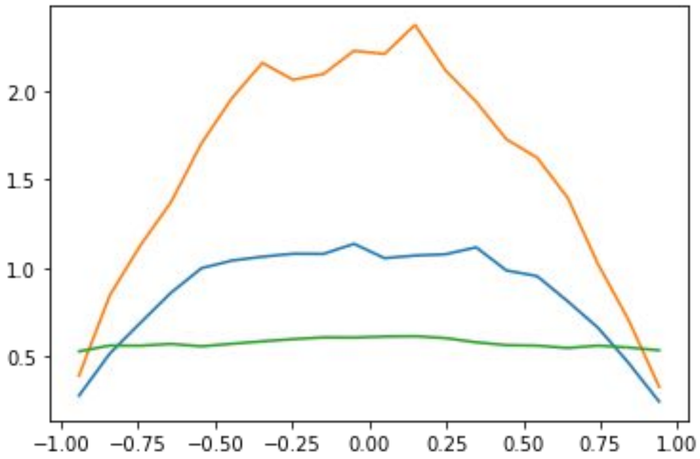


- AR by Maximum Likelihood
- AI trained on AR
- Average Rank (0.5 means equal performance, >0.5 means AI better)

Squared Error vs Order of the AR



Squared Error vs Coefficient in AR(1)

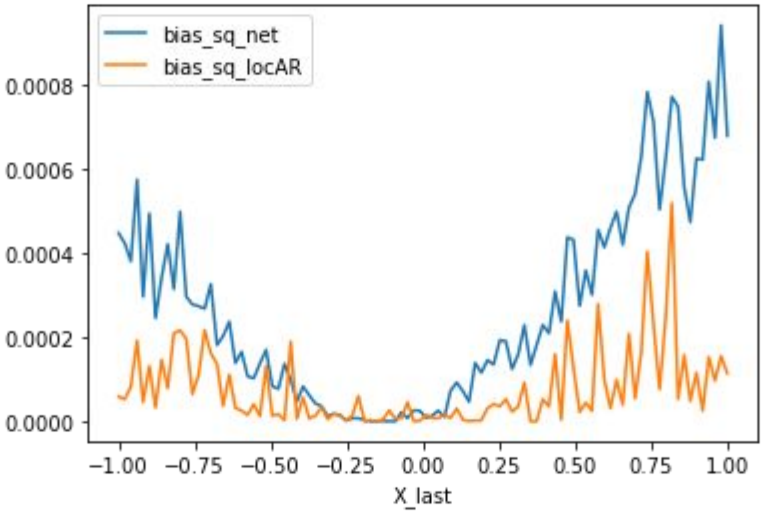


Insight: More Signal -> Less Benefit of AI

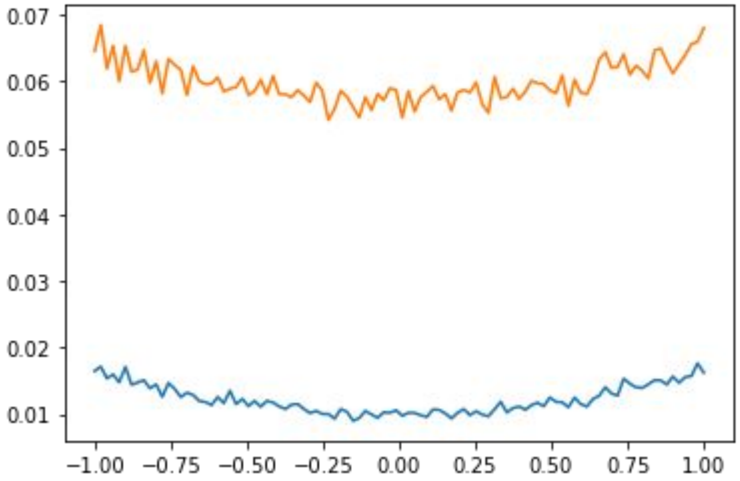
- **AR by Maximum Likelihood**
- **AI trained on AR**
- **Average Rank (0.5 means equal performance, > 0.5 means AI better)**

Bias-Variance Decomposition of Squared Error:
The AI is learning a Biased Estimator

Bias



Variance



- **AR by Maximum Likelihood**
- **AI trained on AR**

This experiment takes a LONG TIME to run!

Real Data

- M4 Yearly and Quarterly
- AI model trained exclusively on simulations of AR **outperforms** Max.Lik.
- **Outperforms** ARIMA if we just difference all time series
- Adding the M4 training to the dataset, it outperforms the M4 winning model

Conclusions

Summary

- A general methodology that can be used to generate **prediction and estimation algorithms** that will outperform most hand-crafted models even under their ideal scenarios
- Combine **existing knowledge** of Time Series with compute

Promising Future

- There will always be some properties of your problem that you know how to simulate, but not how to estimate
- **Multivariate Time Series**
- **Long Forecast Horizons (unlimited!)**
- Pre-Trained Models: Versions of this idea will be made public soon