

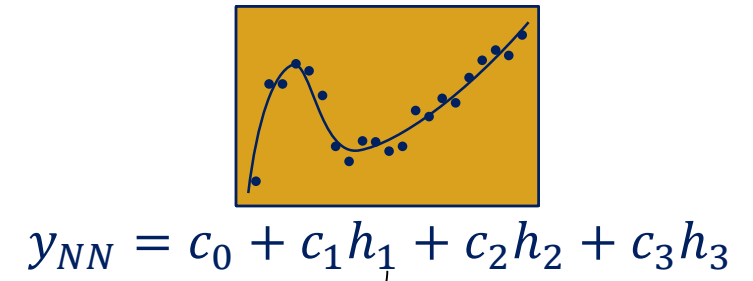
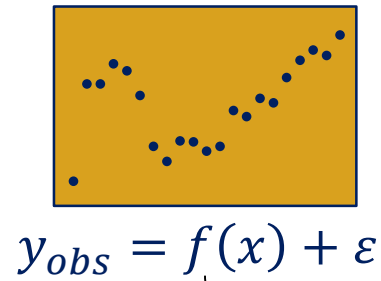
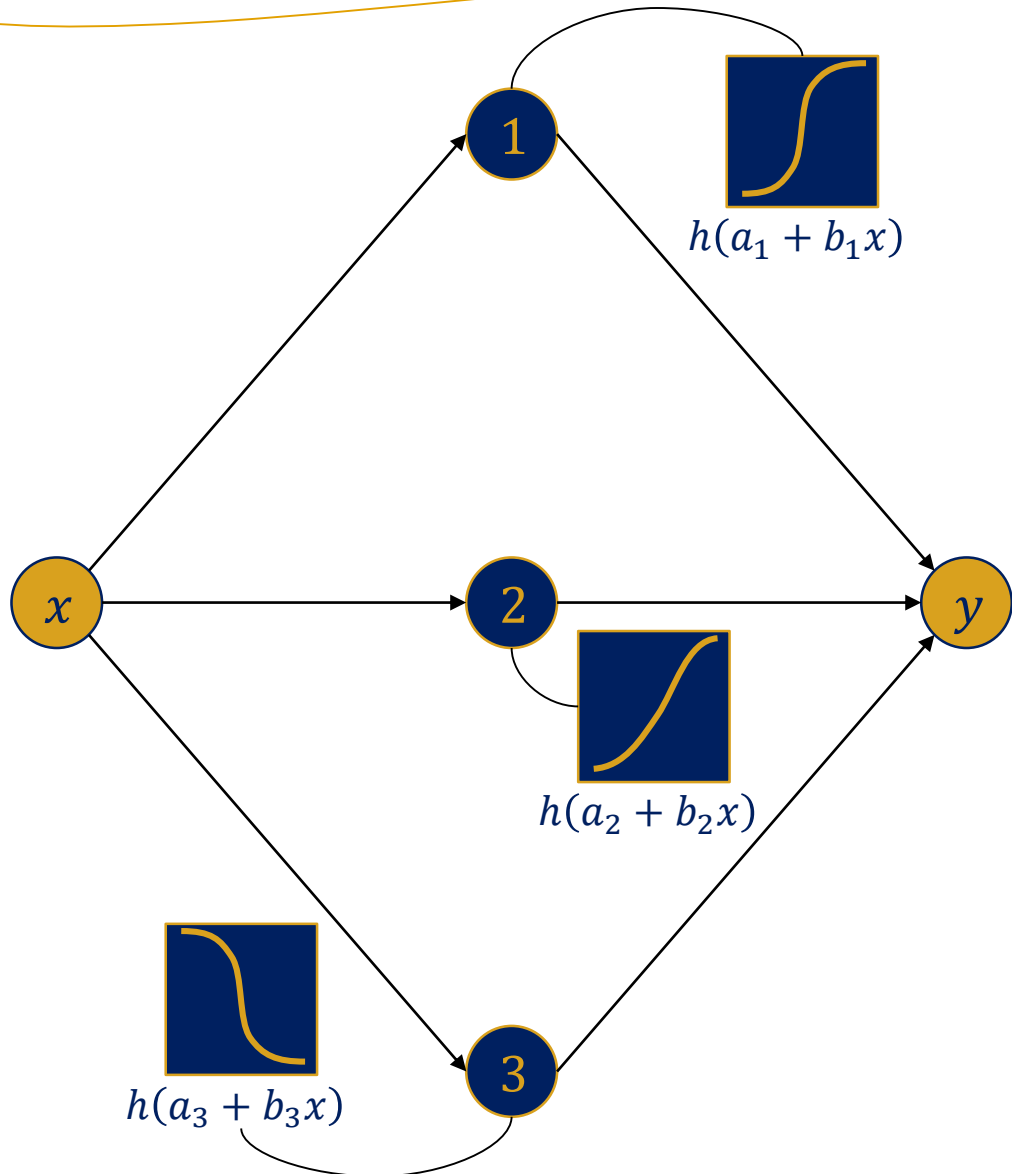


**Neural Networks and Cellular Automata  
in  
Actuarial and Financial Engineering**

Frank Cuypers

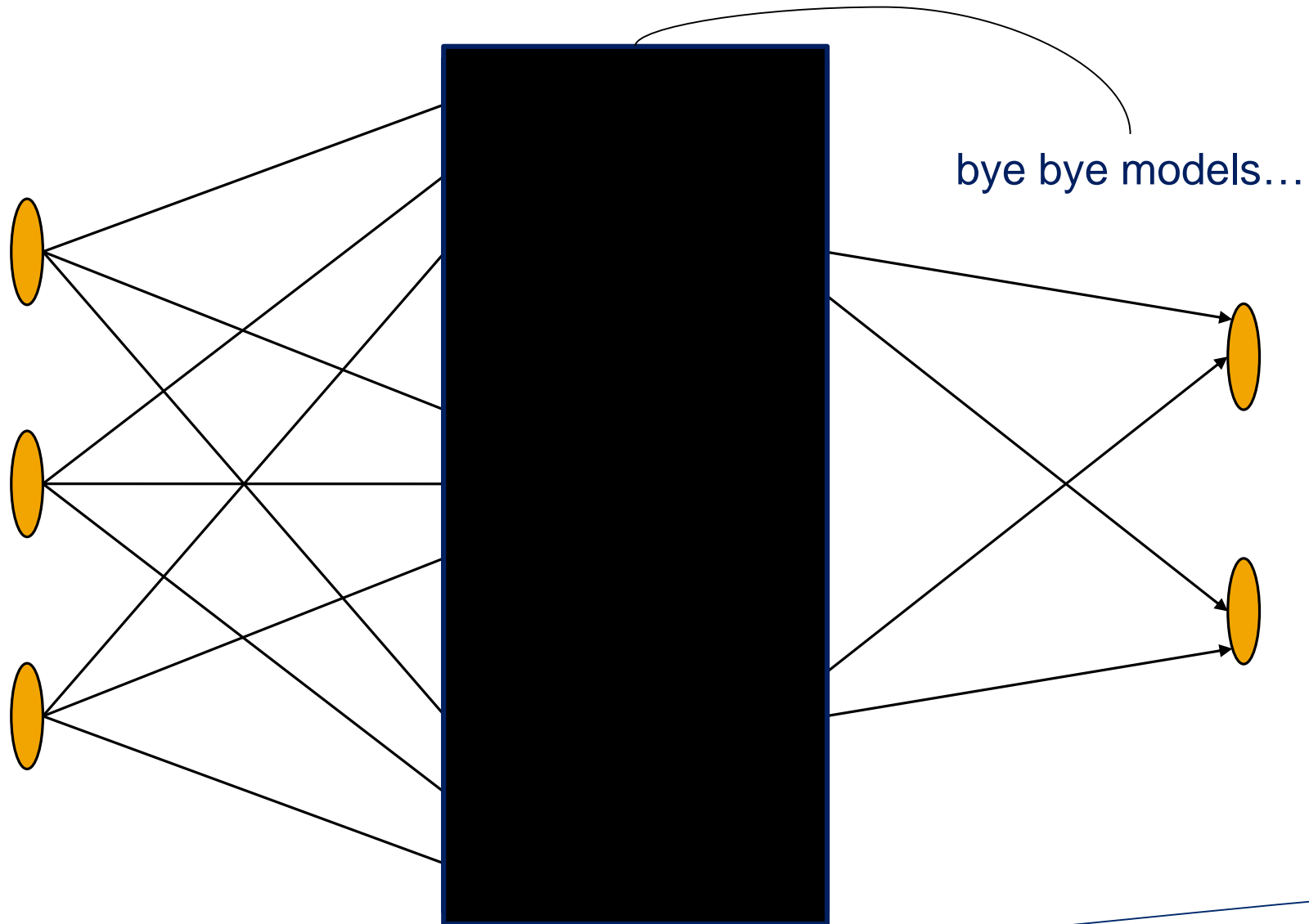
- Neural Networks
- Cellular Automata

# Neural Networks in a Nutshell

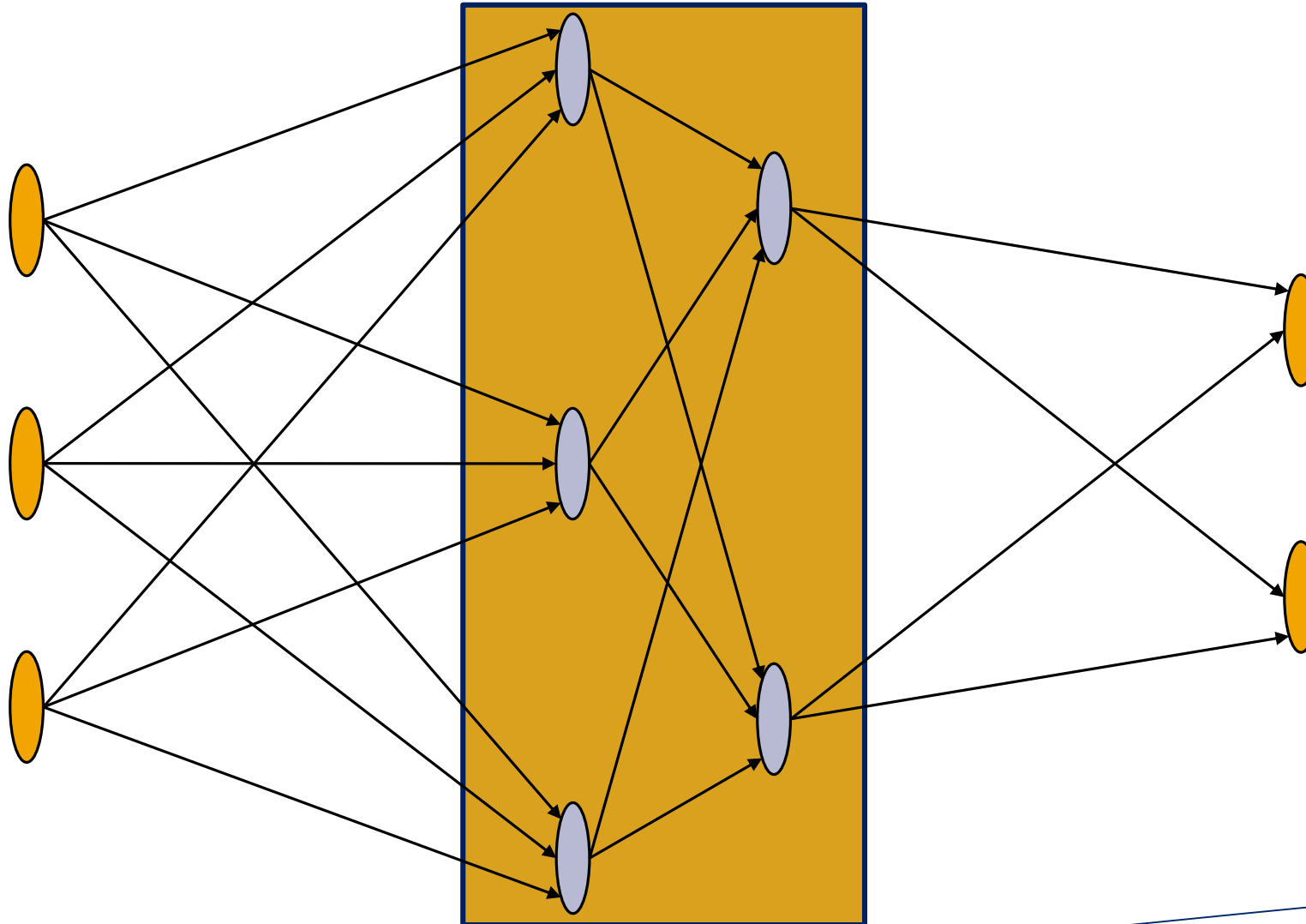


Training: minimize  $\sum (y_{obs} - y_{NN})^2$

# Neural Networks

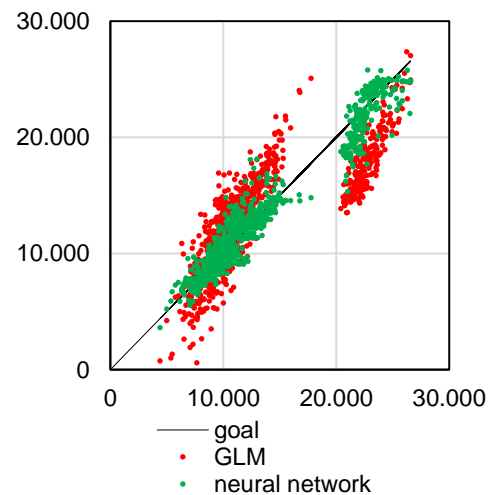
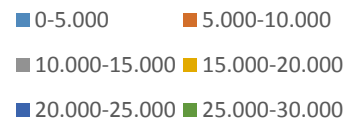
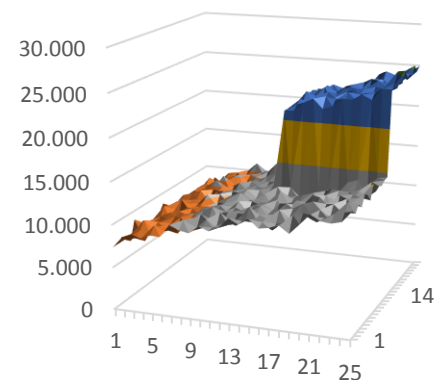


# Neural Networks – 2 Hidden Layers



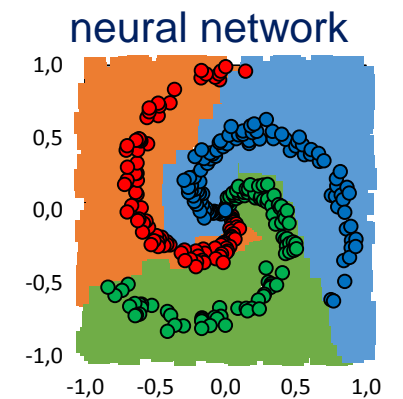
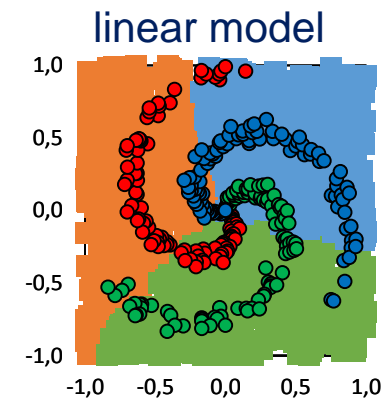
## Regression

- Output = e.g. size of claim
- Each output neuron gives a number, which can take any value

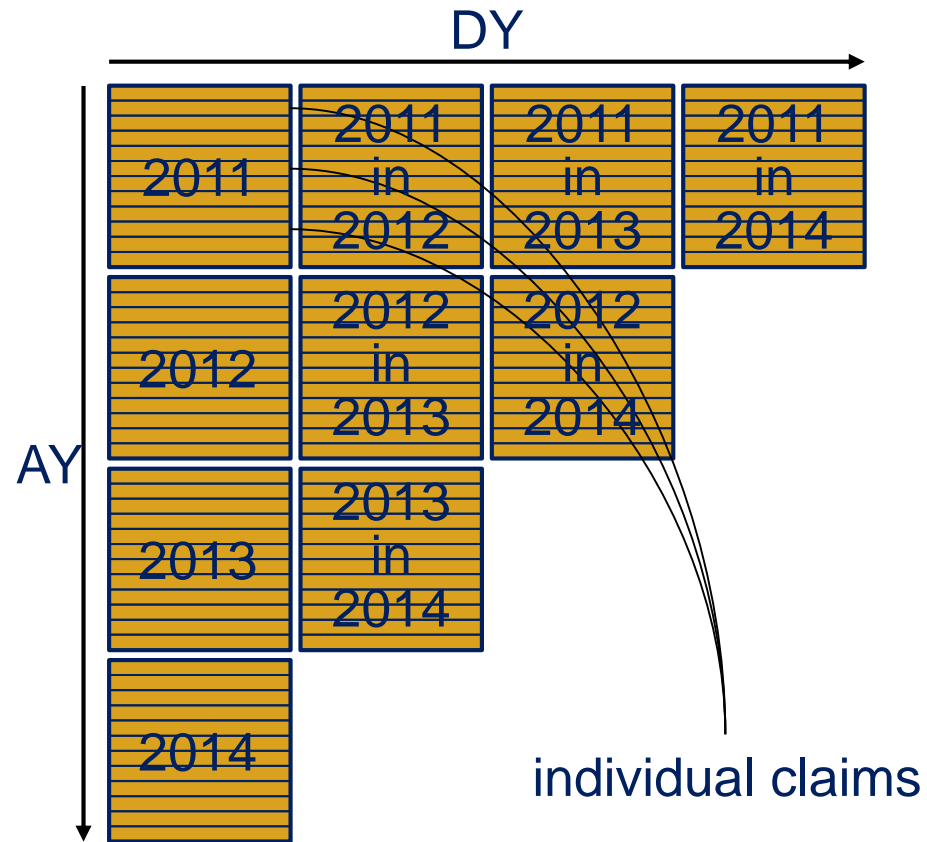


## Classification

- Output = e.g. type of claim
- Each output neuron gives a probability, which all add up to 100%

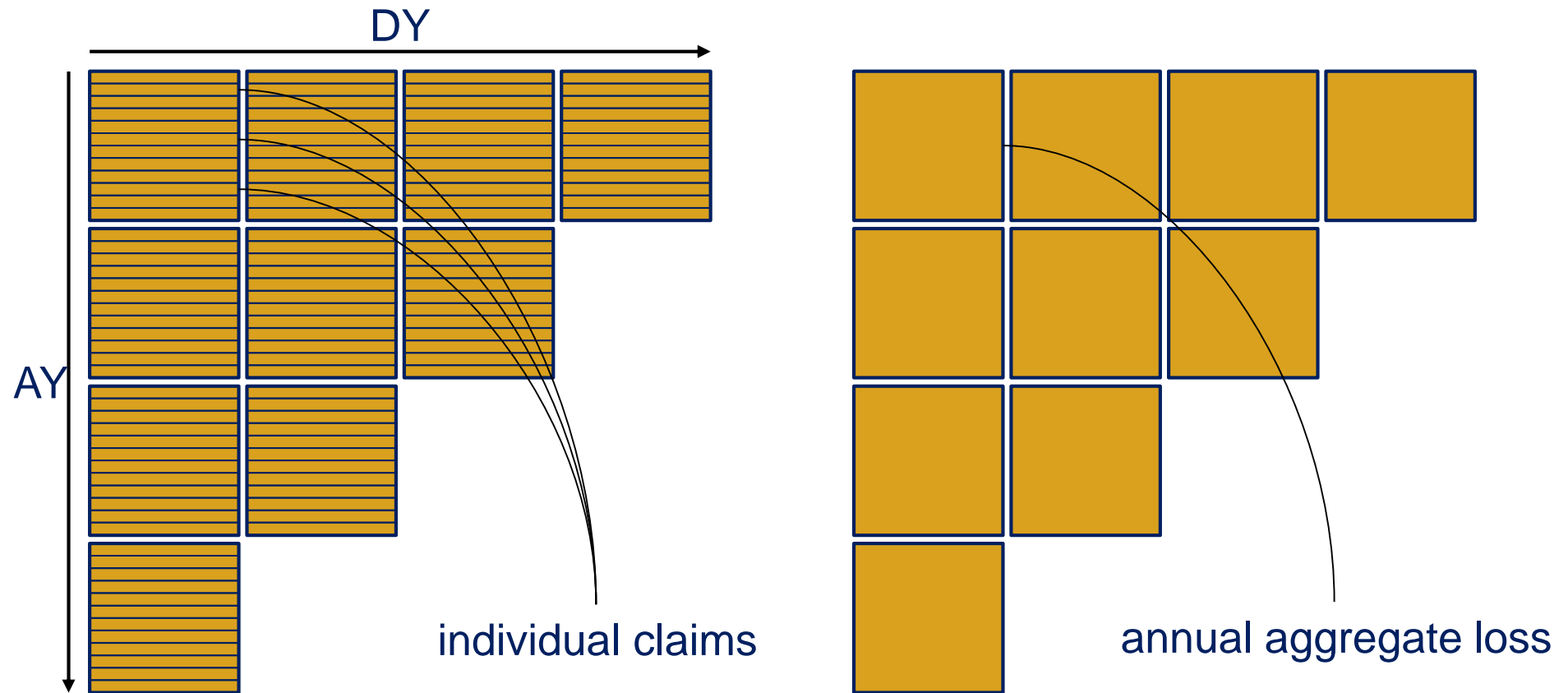


# Loss Development



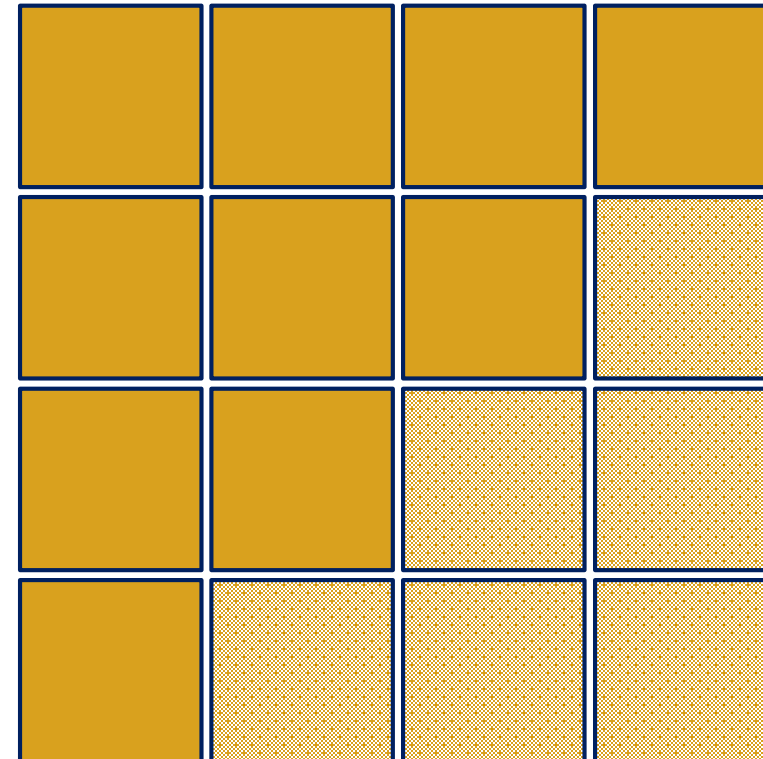
# Loss Development

- Aggregate all claims of a given AY into a single aggregate loss ☹️



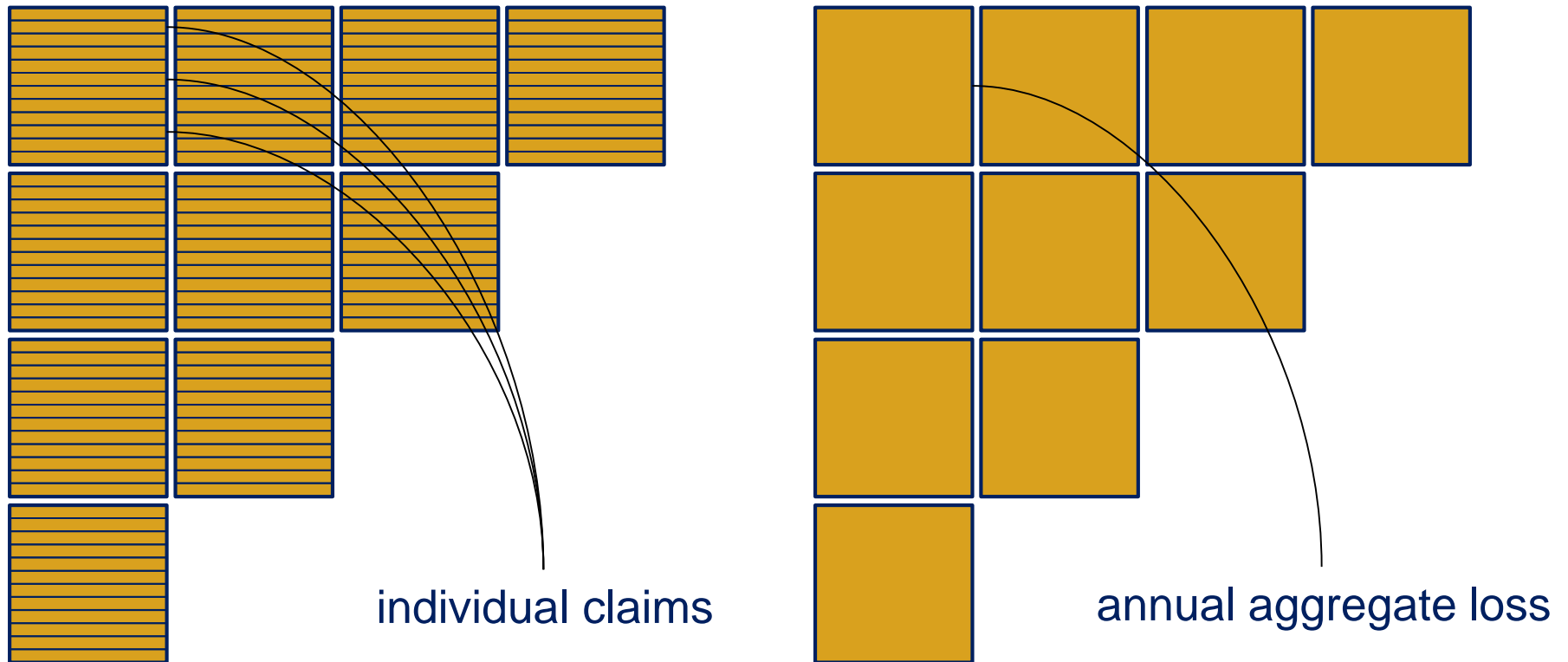
# Traditional Loss Development

- Aggregate all claims of a given AY into a single aggregate loss ☹️
- Develop with
  - Chain Ladder
  - Born-Ferg
  - Cape Cod
  - ...
- Assume
  - Homogenous portfolio
  - Independent AY



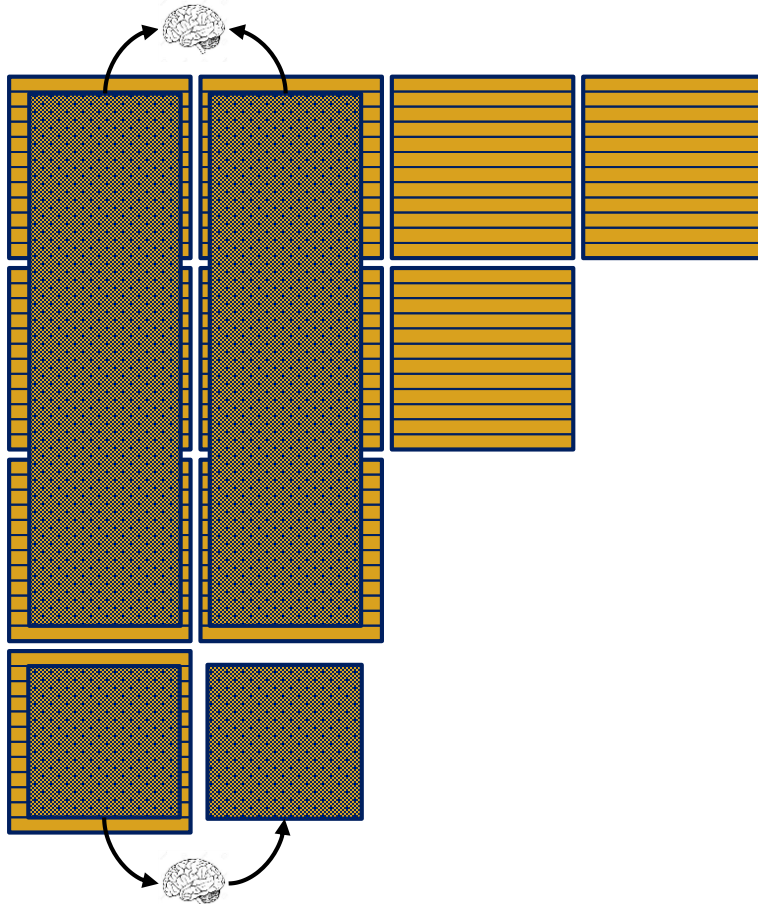
# Individual Claims Development

- ~~Aggregate all claims of a given AY into a single aggregate loss ☹️~~
- Use individual claims information 😊



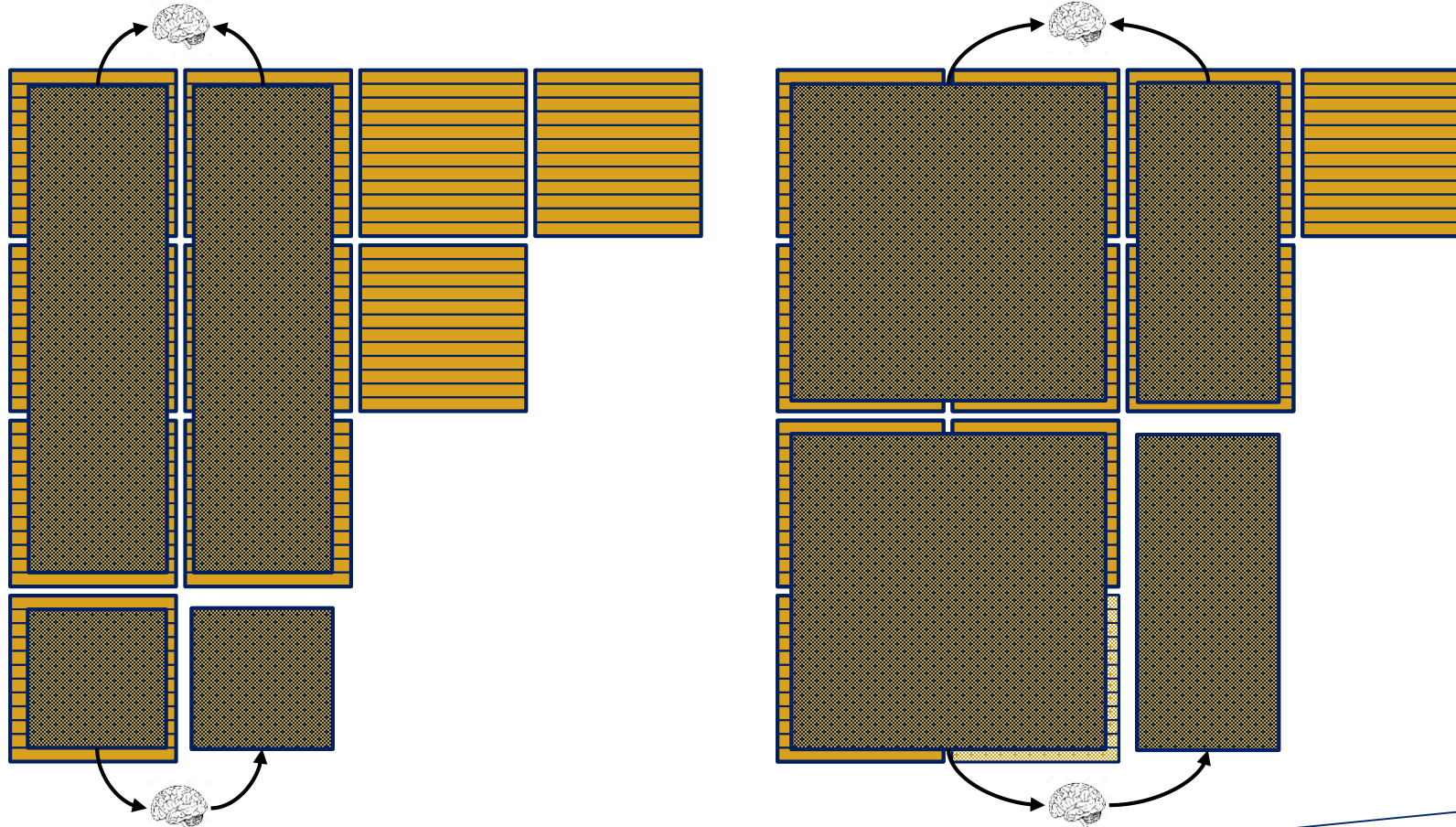
# Individual Claims Development

- ~~Aggregate all claims of a given AY into a single aggregate loss ☹️~~
- Use individual claims information 😊 cascading DY neural network



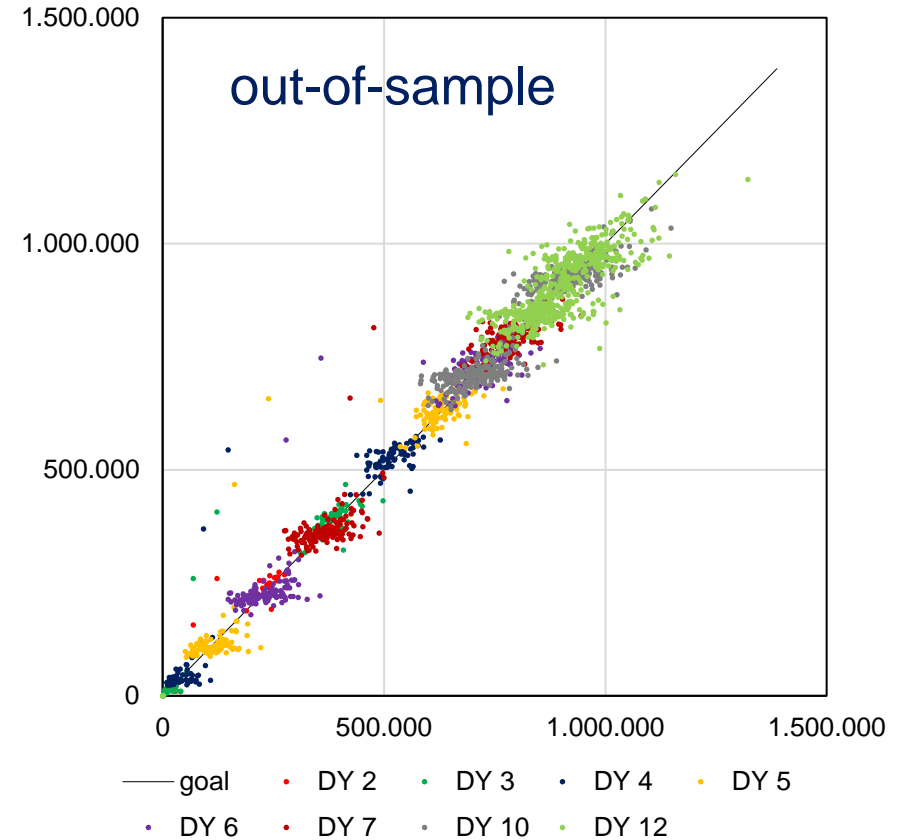
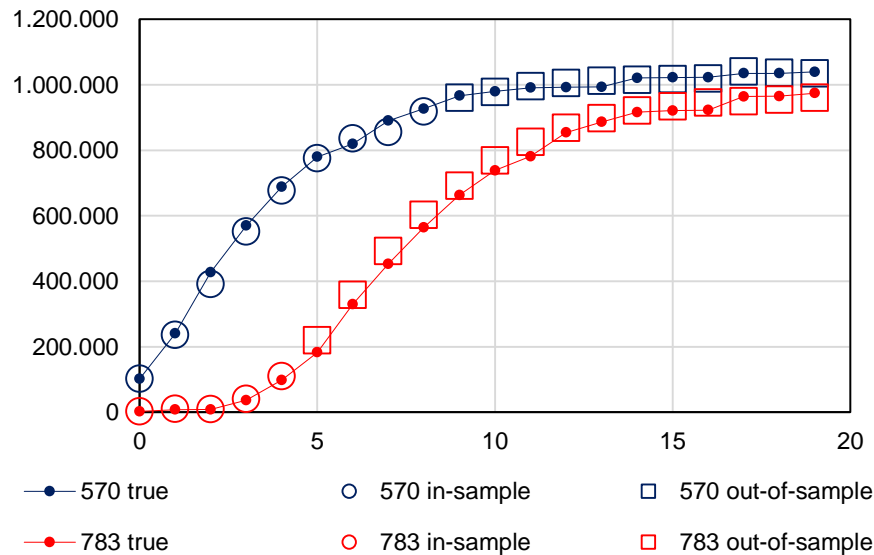
# Individual Claims Development

- ~~Aggregate all claims of a given AY into a single aggregate loss ☹️~~
- Use individual claims information 😊 cascading DY neural network



# Simple Test

- 2 types of claims
- Cascading DY
- 1 hidden layer
- 8 neurons
- Pairs only



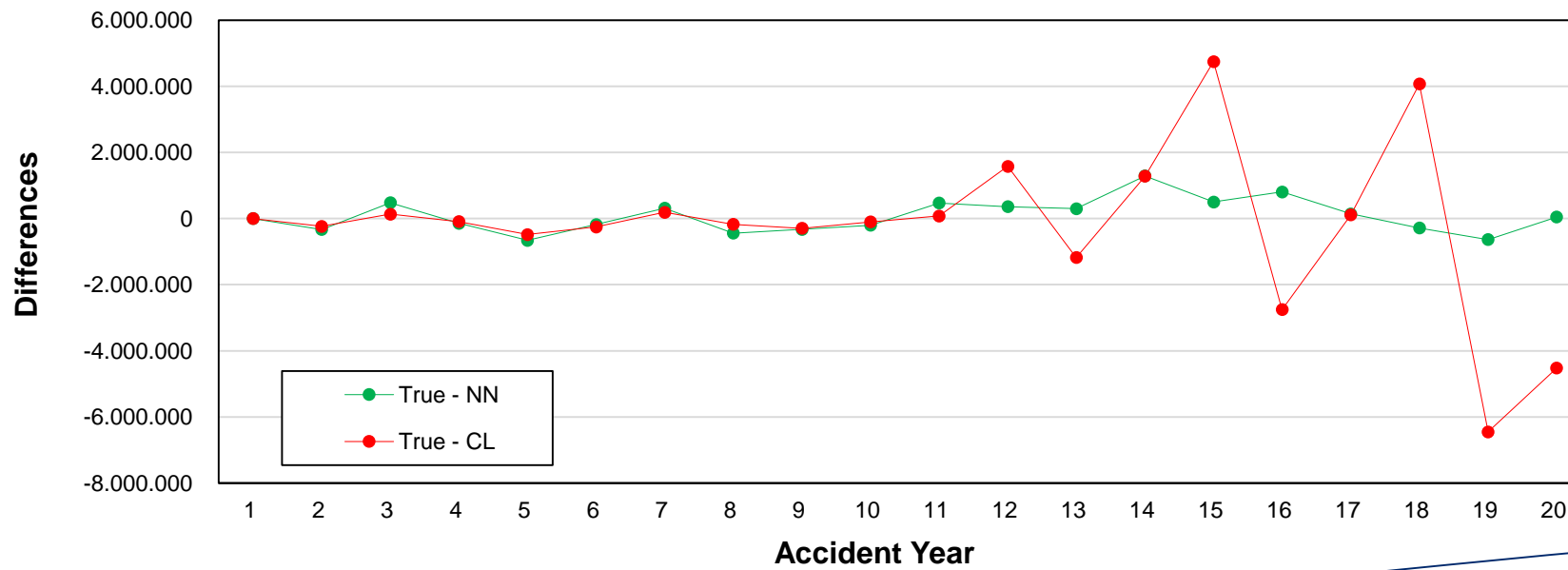
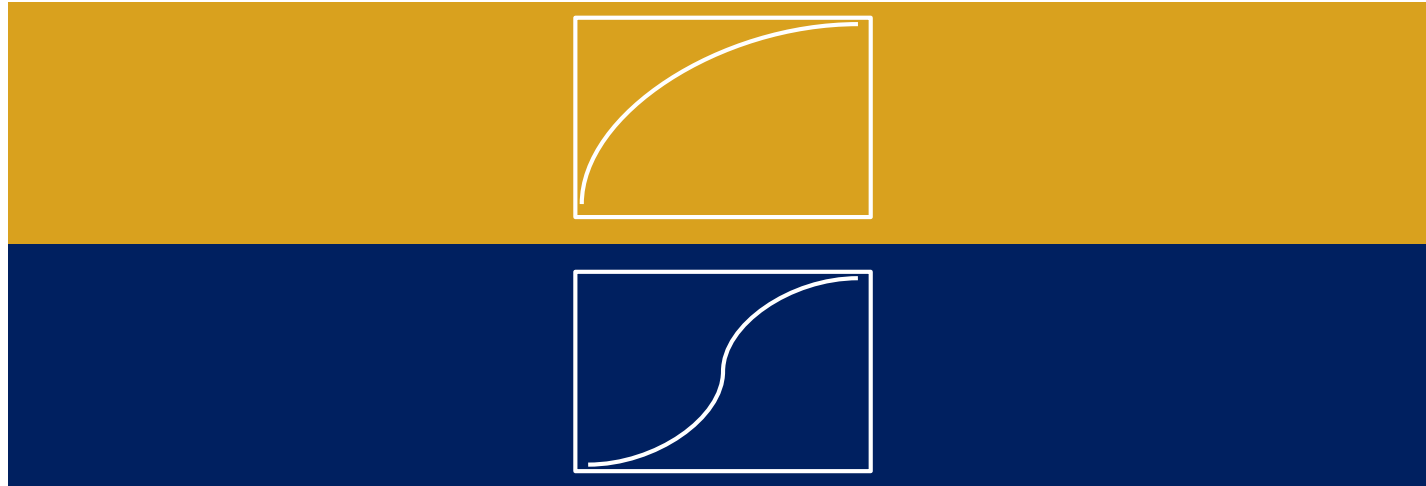
## Aggregate Loss Development

- Develop with
  - Chain Ladder
  - Born-Ferg
  - ...
- ☹️ Aggregates all claims of a given AY into a single aggregate loss
- ☹️ Works either on paid or incurred losses
- 💣 Assumes
  - Homogeneous portfolio
  - Independent AY

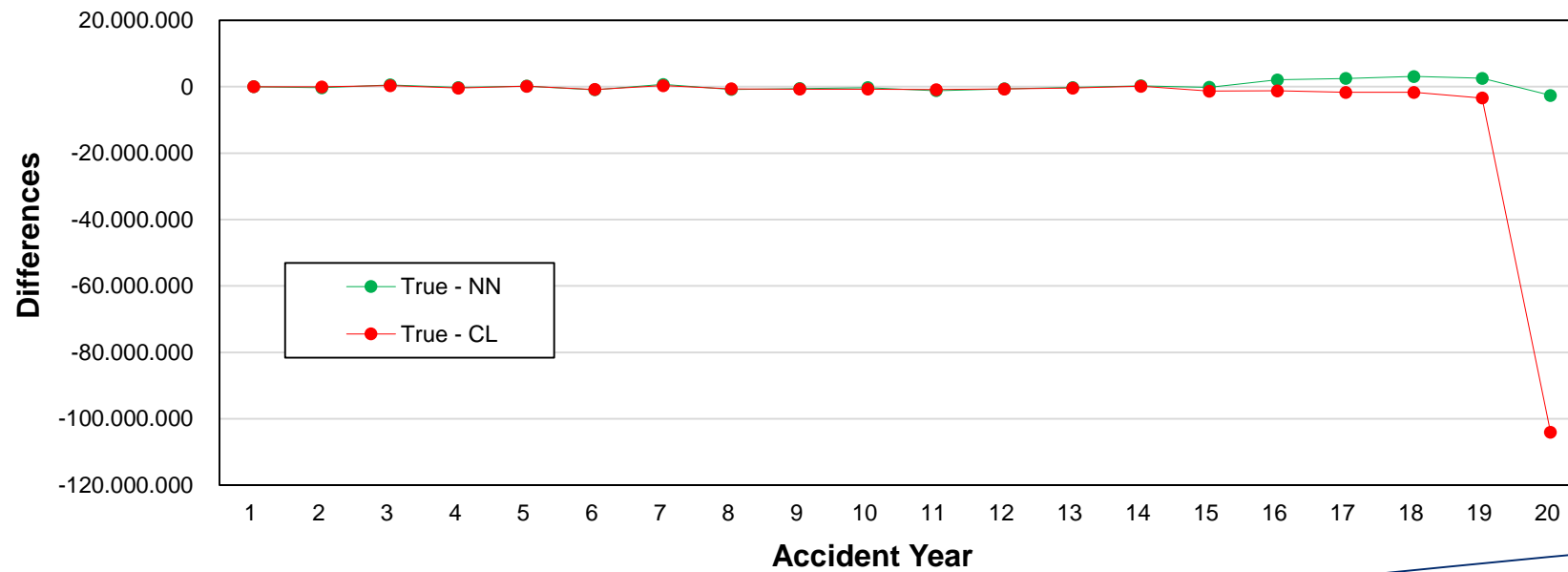
## Individual Claims Development

- Develop with
  - DY or AY cascades
  - Convolutional networks
  - ...
- 😊 Considers all individual claims' features, including non monetary inputs
- 😊 Considers simultaneously payments and reserves
- 😊 Works with
  - Heterogeneous portfolios
  - Dependent AY

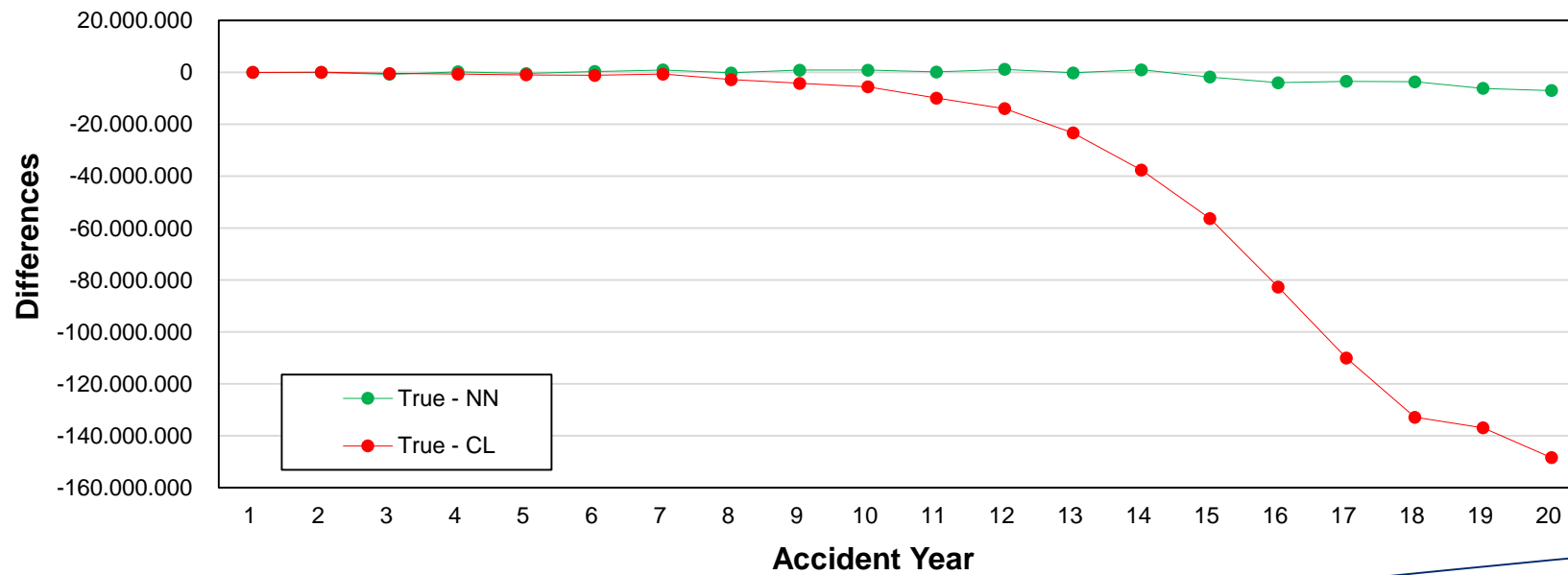
# DY Cascade vs Chain Ladder



# DY Cascade vs Chain Ladder



# DY Cascade vs Chain Ladder



# Outline



- Neural Networks
- Cellular Automata

# Cellular Automaton

Nearest neighbour 2 states XOR automaton:

$$Z_i(t) = Z_{i-1}(t-1) \oplus Z_{i+1}(t-1)$$

$\oplus$	-1	1
-1	-1	1
1	1	-1

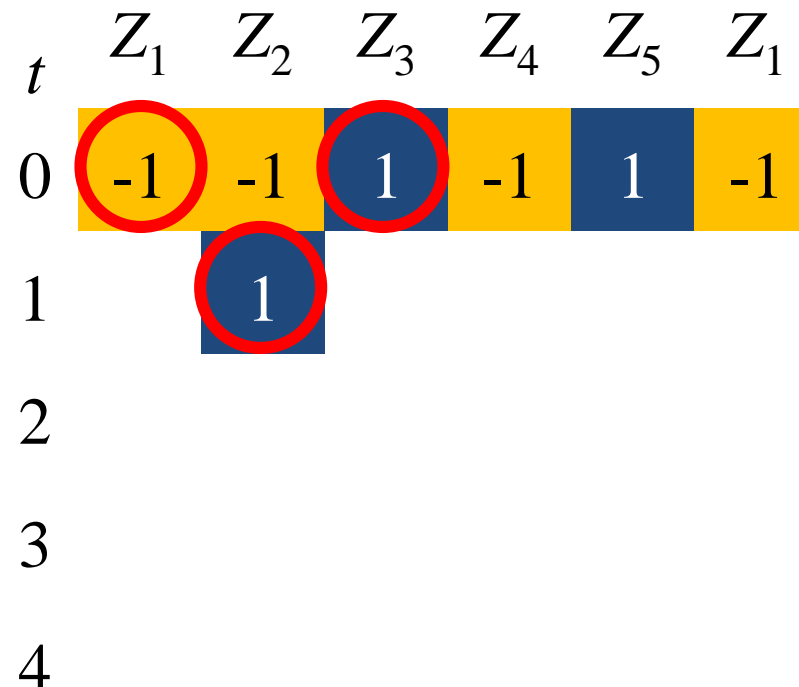
$t$	$Z_1$	$Z_2$	$Z_3$	$Z_4$	$Z_5$	$Z_1$
0	-1	-1	1	-1	1	-1
1						
2						
3						
4						

# Cellular Automaton

Nearest neighbour 2 states XOR automaton:

$$Z_i(t) = Z_{i-1}(t-1) \oplus Z_{i+1}(t-1)$$

$\oplus$	-1	1
-1	-1	1
1	1	-1



# Cellular Automaton

Nearest neighbour 2 states XOR automaton:

$$Z_i(t) = Z_{i-1}(t-1) \oplus Z_{i+1}(t-1)$$

$\oplus$	-1	1
-1	-1	1
1	1	-1

$t$	$Z_1$	$Z_2$	$Z_3$	$Z_4$	$Z_5$	$Z_1$
0	-1	-1	1	-1	1	-1
1	-1	1	-1	-1	-1	-1
2						
3						
4						

# Cellular Automaton

Nearest neighbour 2 states XOR automaton:

$$Z_i(t) = Z_{i-1}(t-1) \oplus Z_{i+1}(t-1)$$

$\oplus$	-1	1
-1	-1	1
1	1	-1

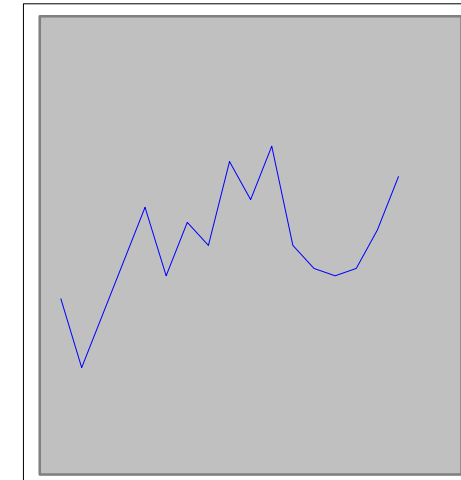
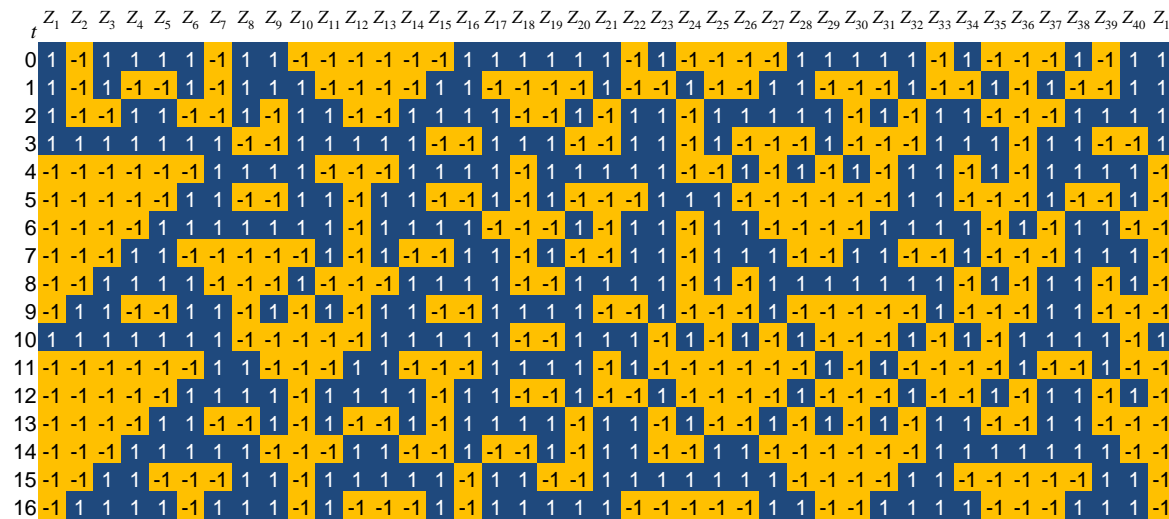
$t$	$Z_1$	$Z_2$	$Z_3$	$Z_4$	$Z_5$	$Z_1$
0	-1	-1	1	-1	1	-1
1	-1	1	-1	-1	-1	-1
2	1	-1	1	-1	-1	1
3	1	-1	-1	1	1	1
4	1	1	1	1	-1	1

Nearest neighbour 2 states XOR automaton:

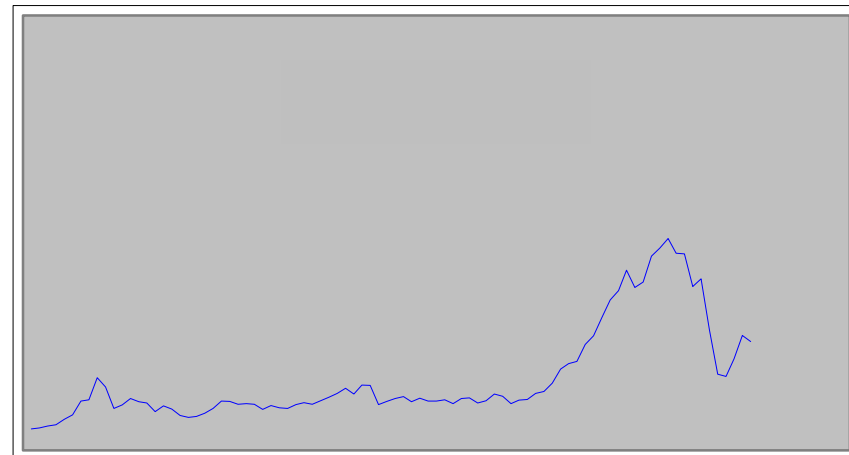
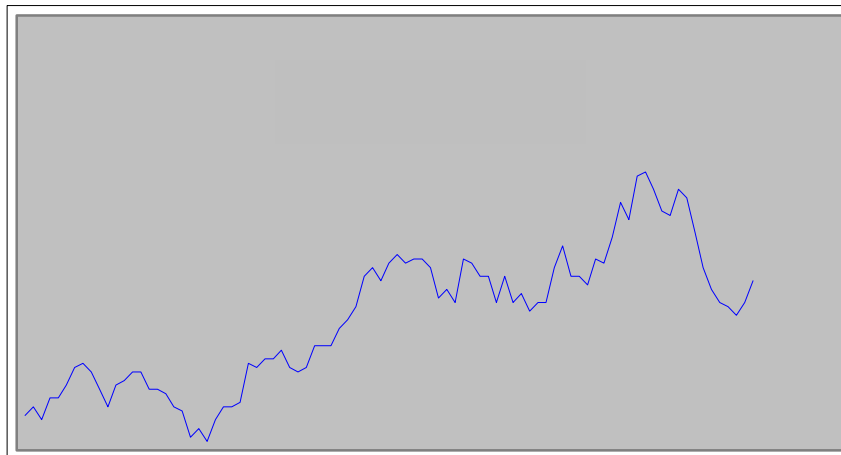
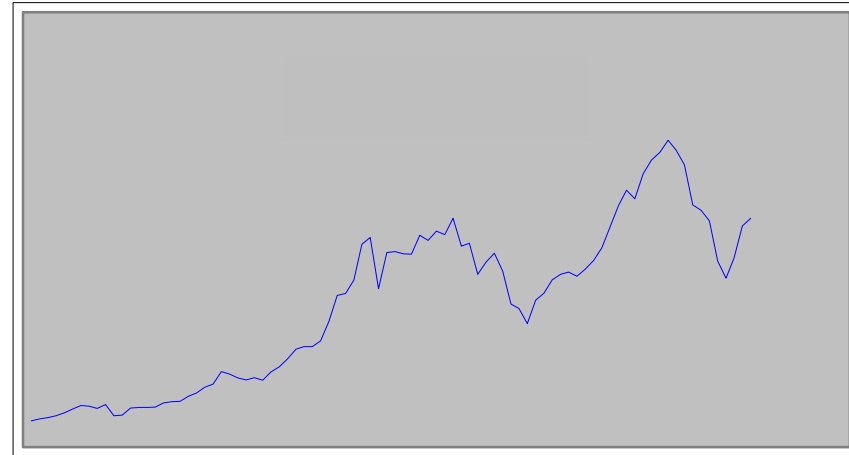
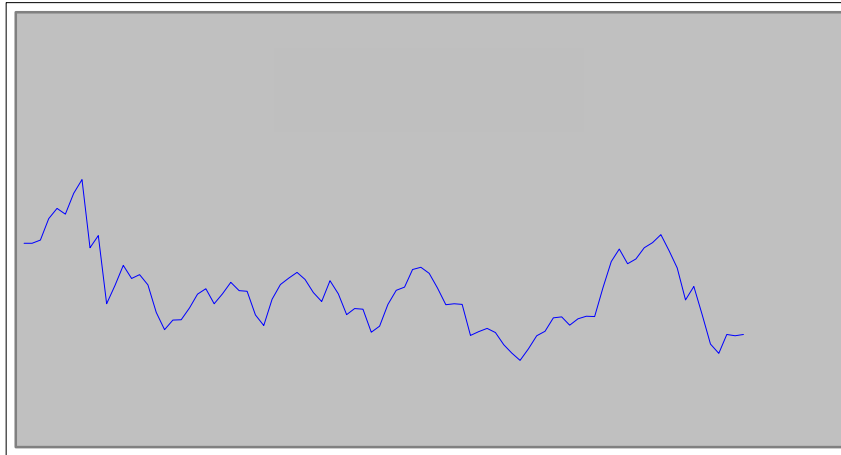
$$Z_i(t) = Z_{i-1}(t-1) \oplus Z_{i+1}(t-1)$$

$$I(t) = I(t-1) + \sum_{i=1}^N Z_i(t)$$

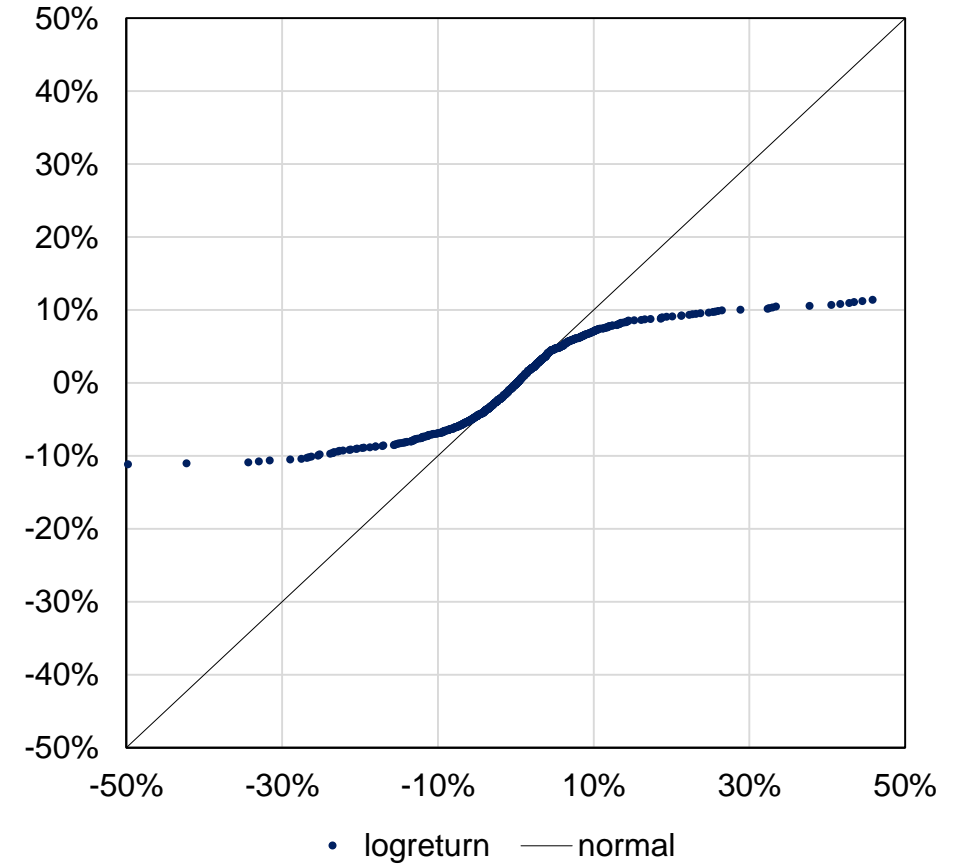
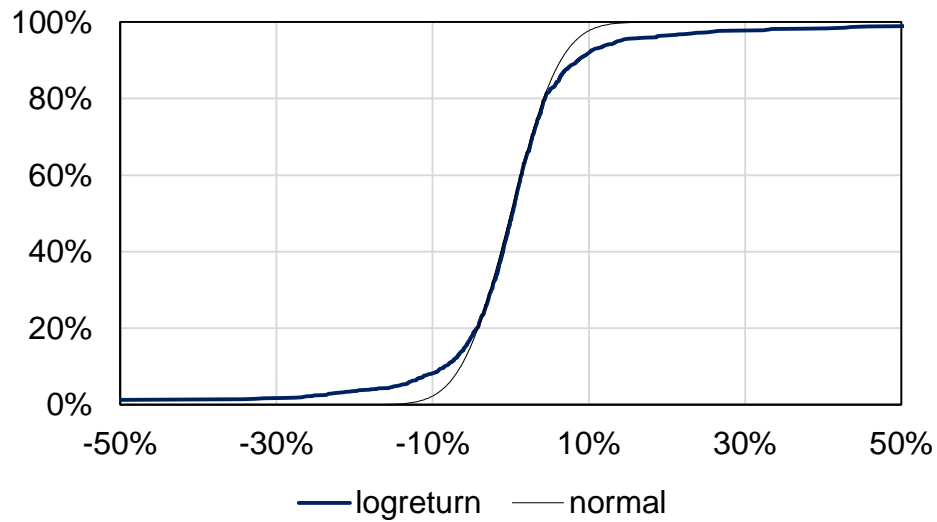
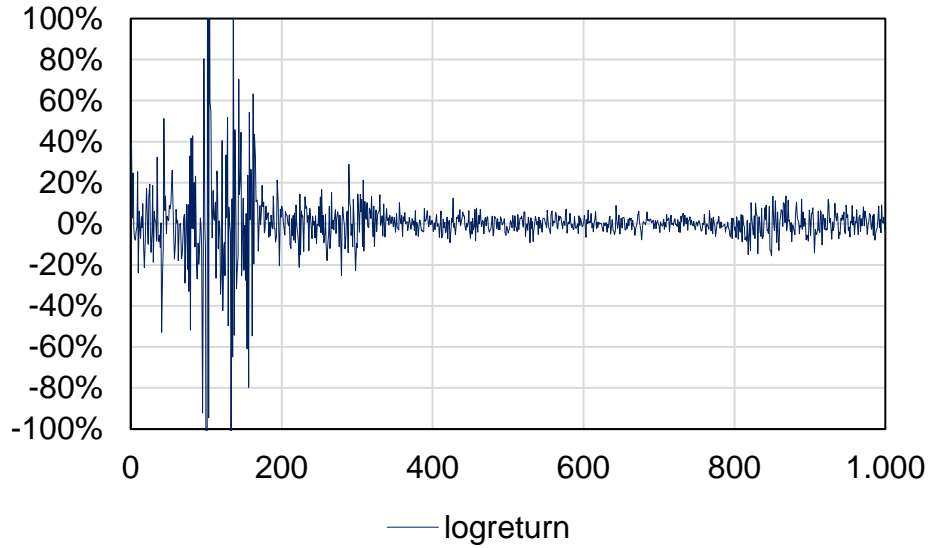
$\oplus$	-1	1
-1	-1	1
1	1	-1



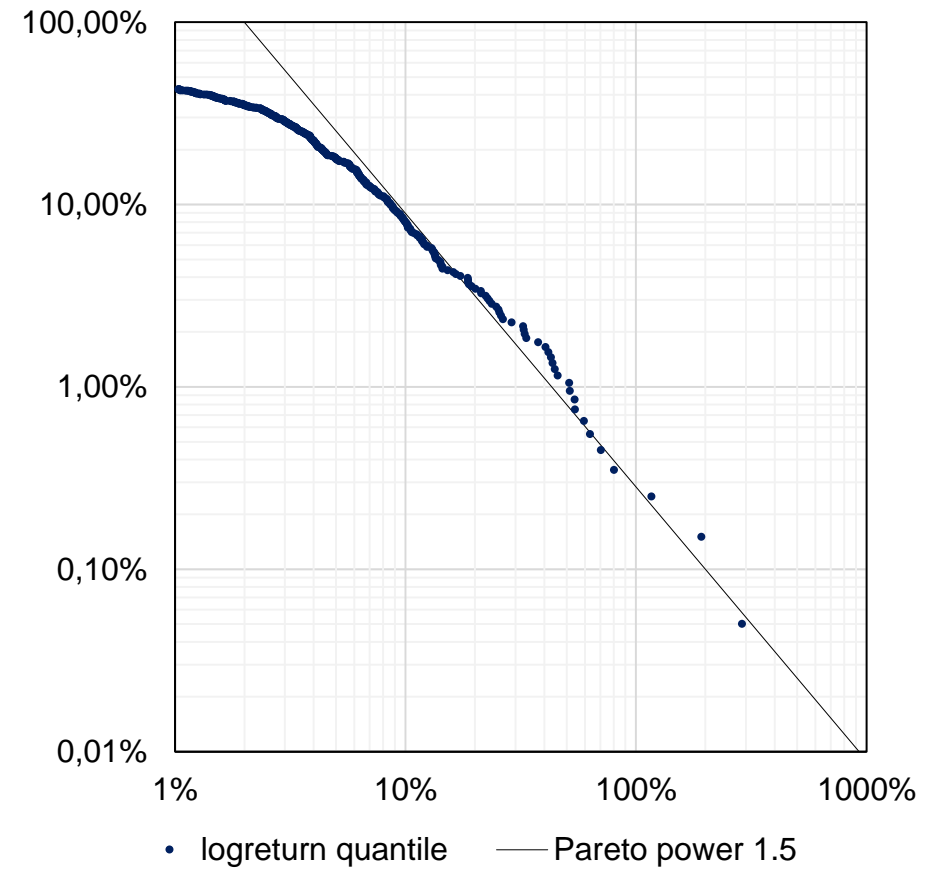
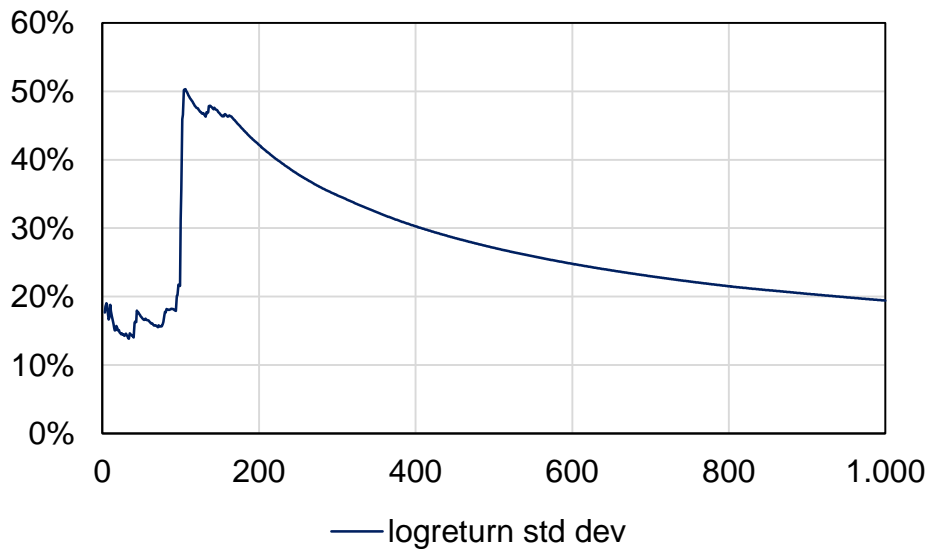
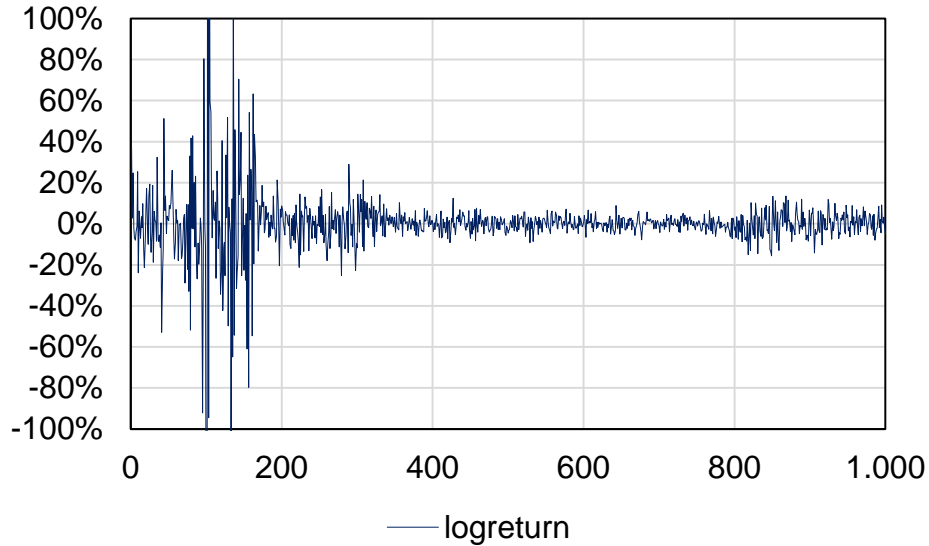
# Equity Indices 1988 - 2010



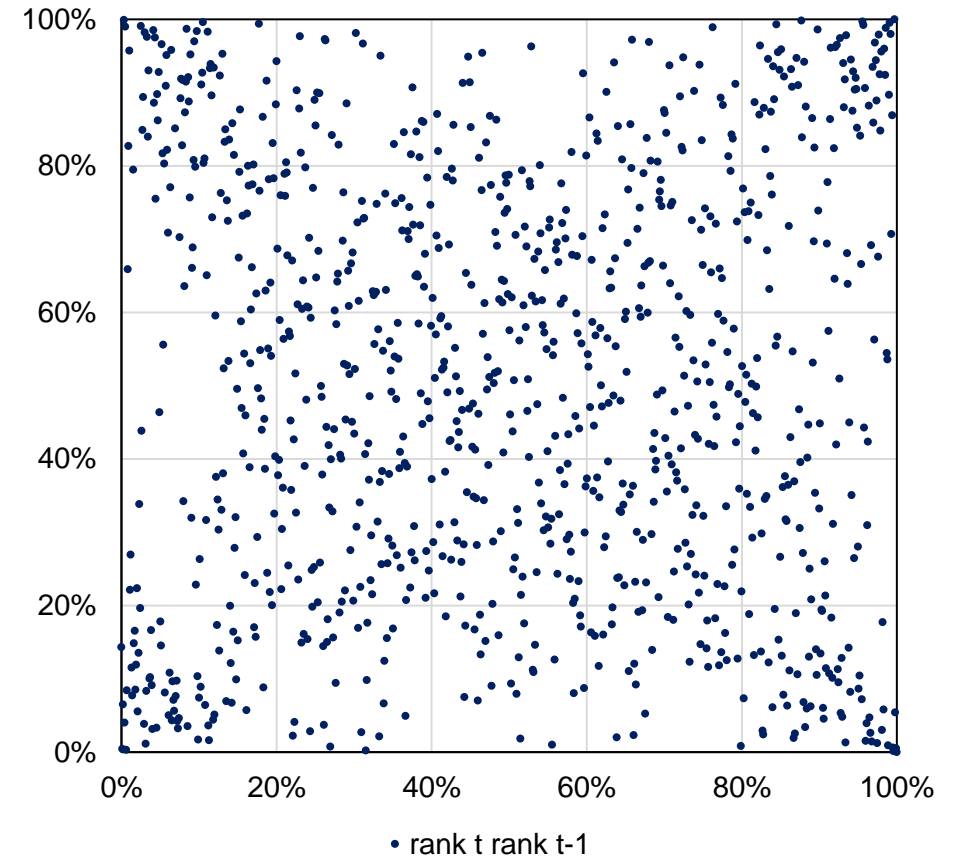
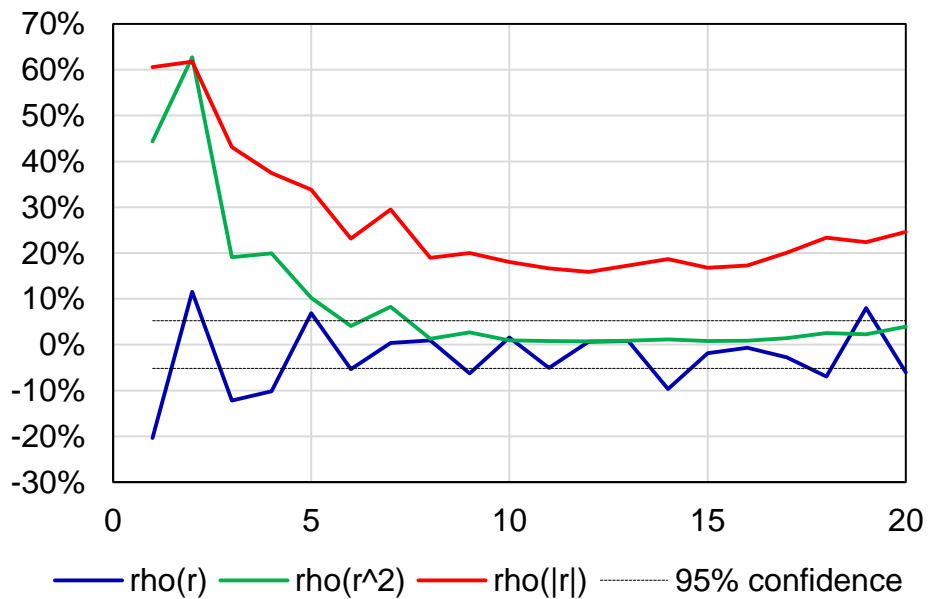
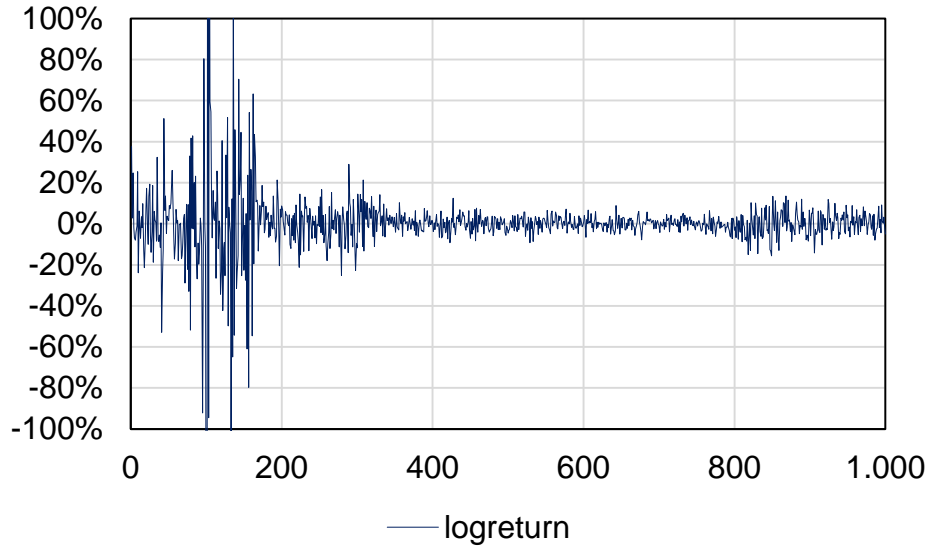
# Logreturns: Distribution



# Logreturns: Volatility & Tail



# Logreturns: Autocorrelation



- Neural Networks
- Cellular Automata

## Statutory reserving

Different models depending on

- data availability / quality
- line of business / market
- processes / products
- ...
- actuarial judgment

→ 1<sup>st</sup> moment of a distribution

 standard reserving model

## Solvency II

Different models depending on

- data availability / quality
- line of business / market
- processes / products
- ...
- actuarial judgment

→ n<sup>th</sup> moment of a distribution

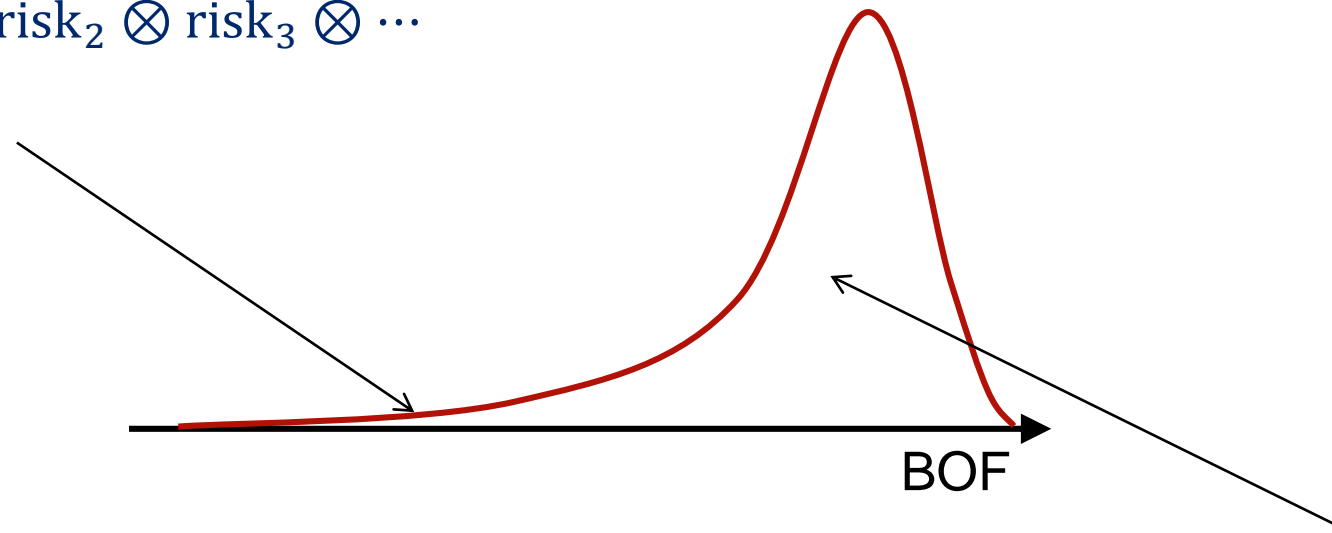
 standard solvency formula

## Internal models

- Numerical aggregation of realistic distributions

$$SCR \leftarrow \text{risk}_1 \otimes \text{risk}_2 \otimes \text{risk}_3 \otimes \dots$$

- Probe the true tail



## Standard Solvency II formula

- Analytic linear approximation

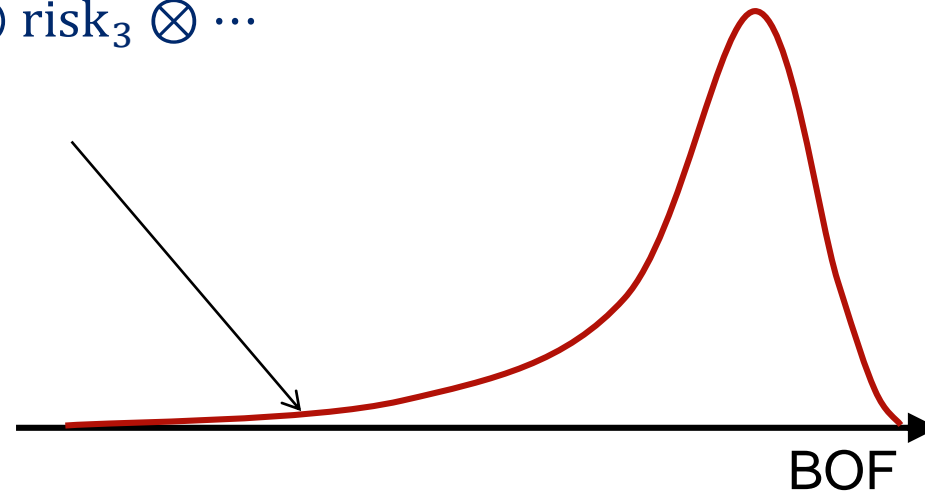
$$SCR \leftarrow \sigma^2 = \sum \rho_{ij} \sigma_i \sigma_j$$

- Probe the tail with 2<sup>nd</sup> moments

# ... et Carthago delenda est!

## Internal model

- Numerical aggregation of realistic distributions  
 $SCR \leftarrow risk_1 \otimes risk_2 \otimes risk_3 \otimes \dots$
- Probe the true tail



*Cut along dotted line*



## Lecturer's Coordinates



Frank Cuypers

+41 (41) 725 32 94

[frank.cuypers@prs-zug.com](mailto:frank.cuypers@prs-zug.com)