AI in Actuarial Science

Ronald Richman FIA FASSA CERA CPCU

Old Mutual Insure

12 July 2022

Remembering Wim Els

"I can't imagine ASSA without Wim. I've worked with him closely since my days as ASABA President, as a Council member, as Principal of the Academy and now as President. The last year was particularly difficult but he helped to guide us with his wisdom and humour. He was a genuine transformation champion, and the South African actuarial community will miss him very much."
Lusani Mulaudzi – President: ASSA

"The loss of Wim has been a devastating blow for the ASSA staff including myself. We were planning a surprise function to celebrate his 25th anniversary at ASSA next week Tuesday. Wim has been a font of wisdom and experience in the office, combining humour with a steely determination to serve the Society, to accelerate transformation and to ensure we serve the public interest. To me, Wim has been not only a hardworking and dedicated colleague but also a mentor and friend. We send our condolences to his wife, Dalene, and his daughters."
Mike McDougall - Chief Executive: ASSA

AI in AS

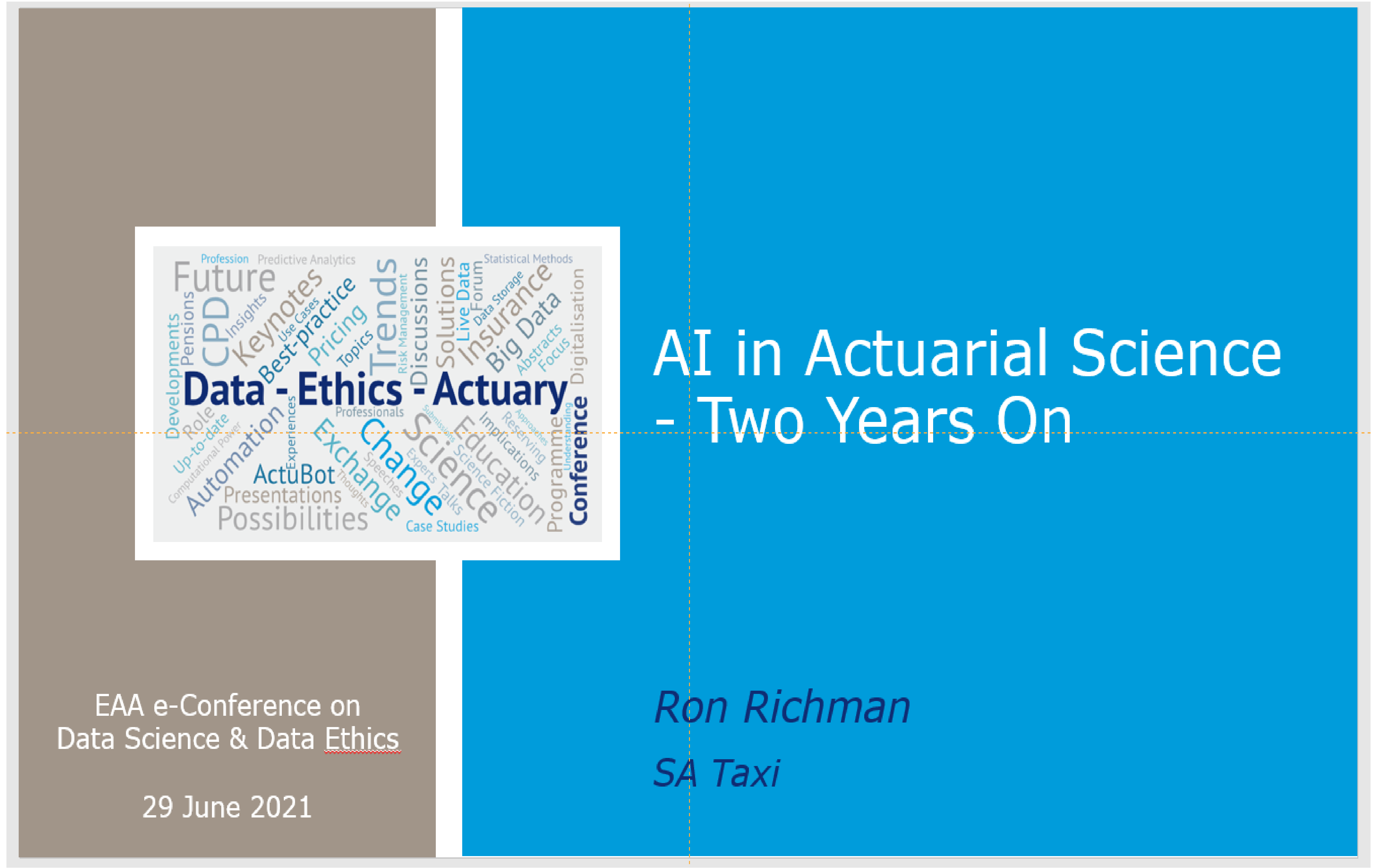
Annals of Actuarial Science (2020), 1–23
doi:10.1017/S1748499520000238

REVIEW

AI in actuarial science – a review of recent advances – part 1

Ronald Richman

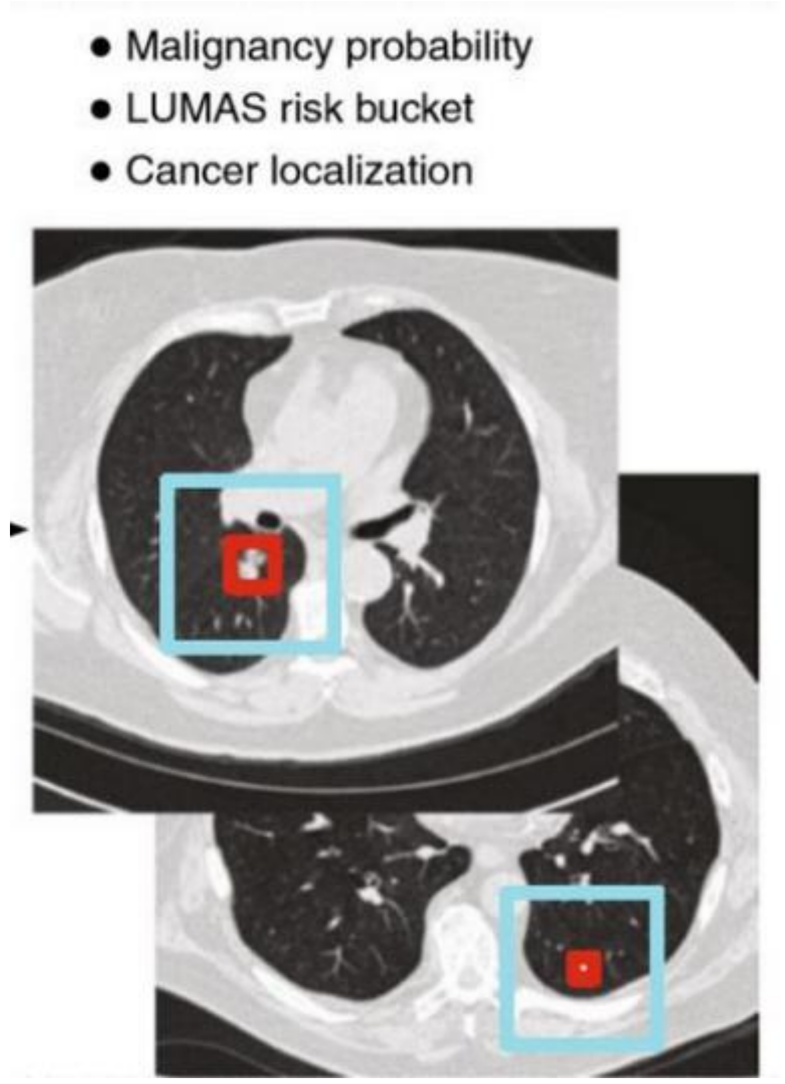
Institute and Faculty of Actuaries

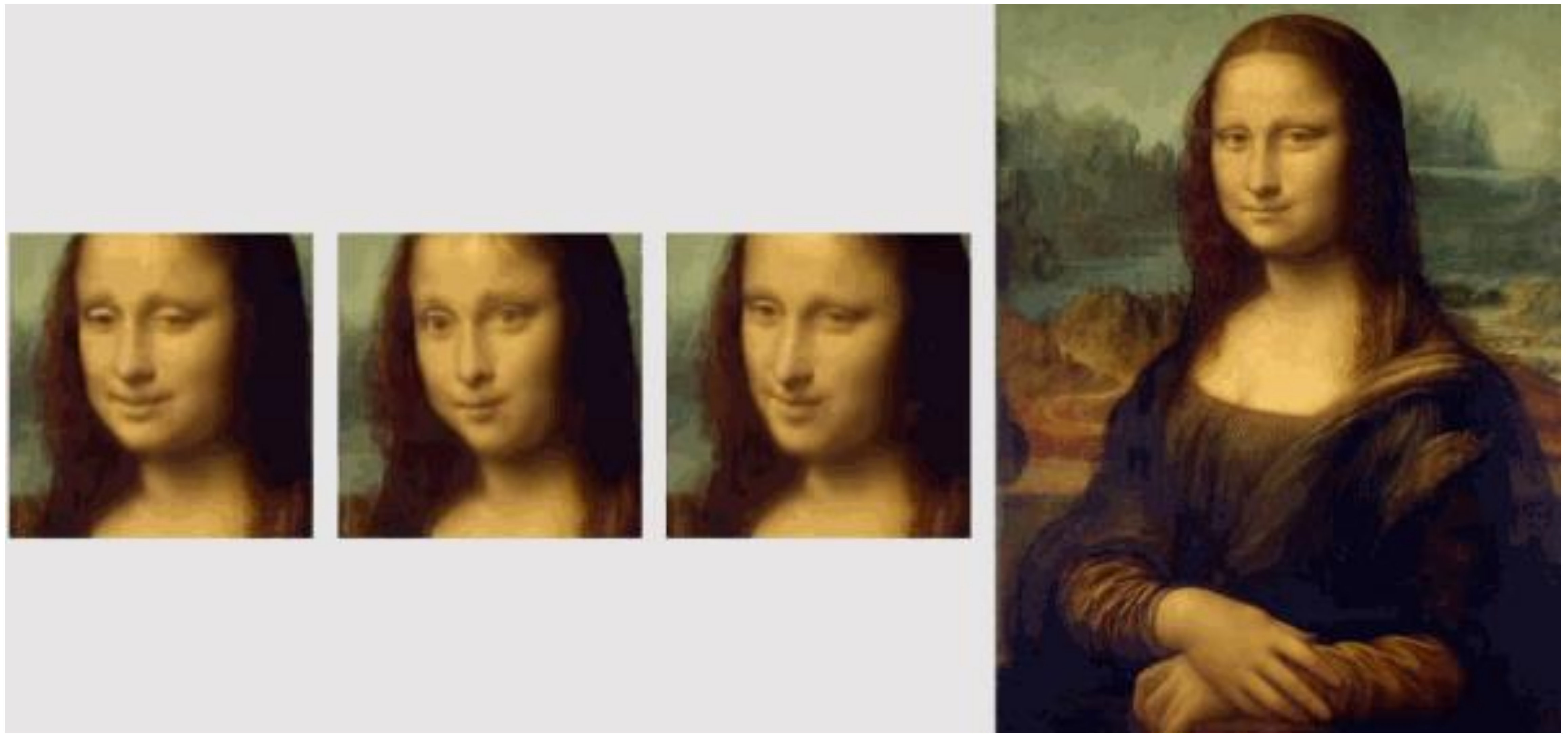


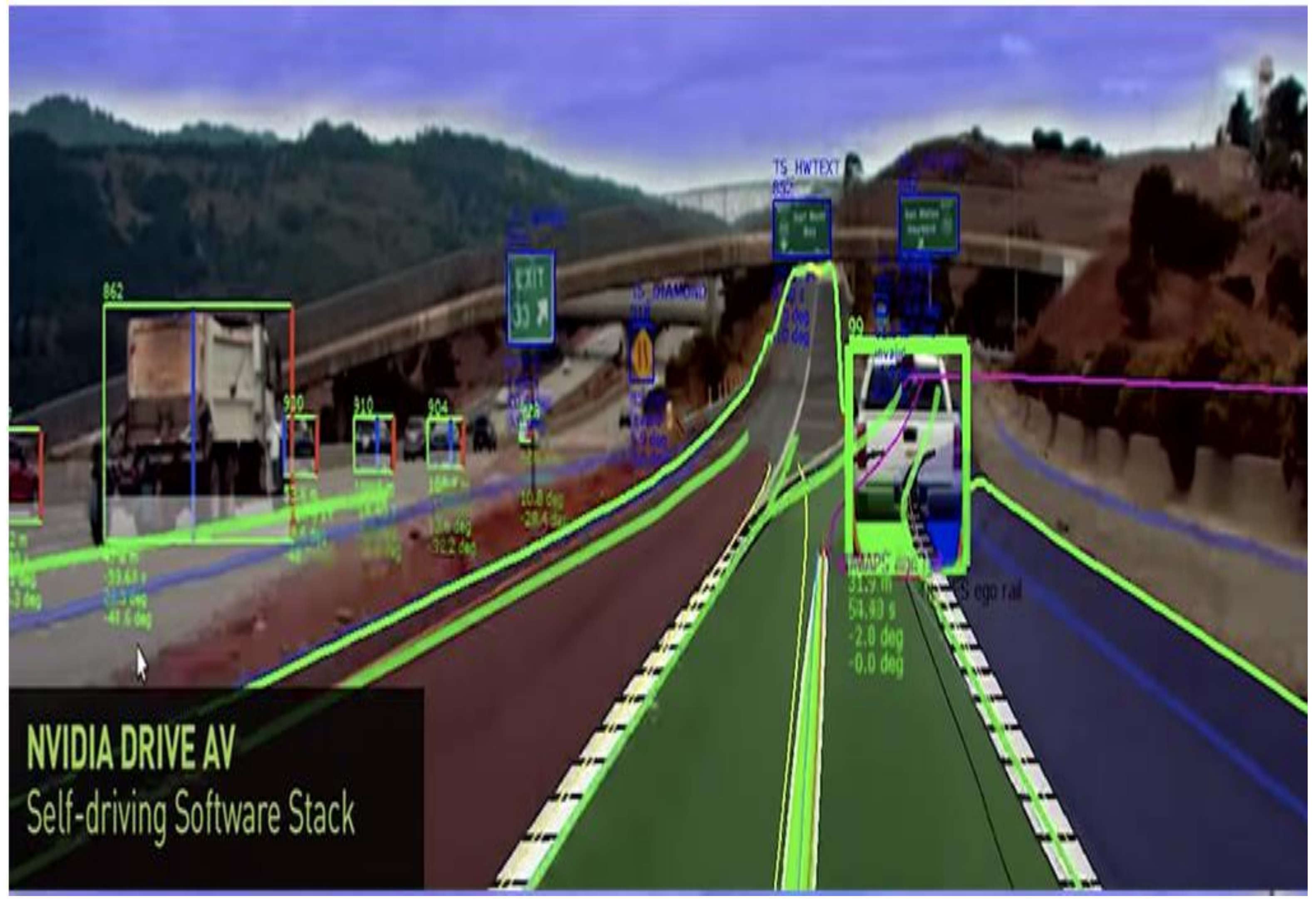
Goals of the talk

- **What machine learning implies for actuarial science**
- **Understand the problems solved by deep learning**
- **Discuss the tools of the trade**
- **Discuss recent successes of deep learning in actuarial science**
- **Discuss emerging challenges and solutions**

Deep Learning in the Wild







An exciting part of the world of finance is insurance

I think we all know that the insurance industry is exciting. I see it everywhere - the airlines, the cars, most all the businesses in the world. The insurance industry can really drive the economic innovation.

But one area of insurance that I really want to see develop more is financial advice. It might be a private sector service but insurance companies are not really there anymore. In general we are not allowed to talk to clients about financial solutions - we need to find a new solution. It would be fun to see what a private sector insurance can deliver.

- Man from www.thispersondoesnotexist.com/
- Mona Lisa from Samsung AI team
- Text from https://talktotransformer.com/
- Self- driving from NVIDIA blog
- Cancer detection from Nature Medicine

Actuarial Data Science

- **Traditionally, actuaries responsible for statistical and financial management of insurers**
  Today, actuaries, data scientists, machine learning engineers and others work alongside each other

- **Actuaries focused on specialized areas such as pricing/reserving**
  Many applications of ML/DL within insurance but outside of traditional areas

- **Actuarial science merges statistics, finance, demography and risk management**
  Currently evolving to include ML/DL

- **According to Data Science working group of the SAA:**
  Actuary of the fifth kind - job description is expanded further to include statistical and computer-science
  Actuarial data science - subset of mathematics/statistics, computer science and actuarial knowledge

- **Focus of talk: ML/DL within Actuarial Data Science – applications of machine learning and deep learning to traditional problems dealt with by actuaries**

Definitions and Diagram from Data Science working group of the Swiss Association of Actuaries (SAA)

Agenda

- **Introduction**
- **<u>Machine Learning</u>**
- **Deep Learning**
- **Tools of the Trade**
- **Selected Applications**
- **Challenges**
- **Conclusion**

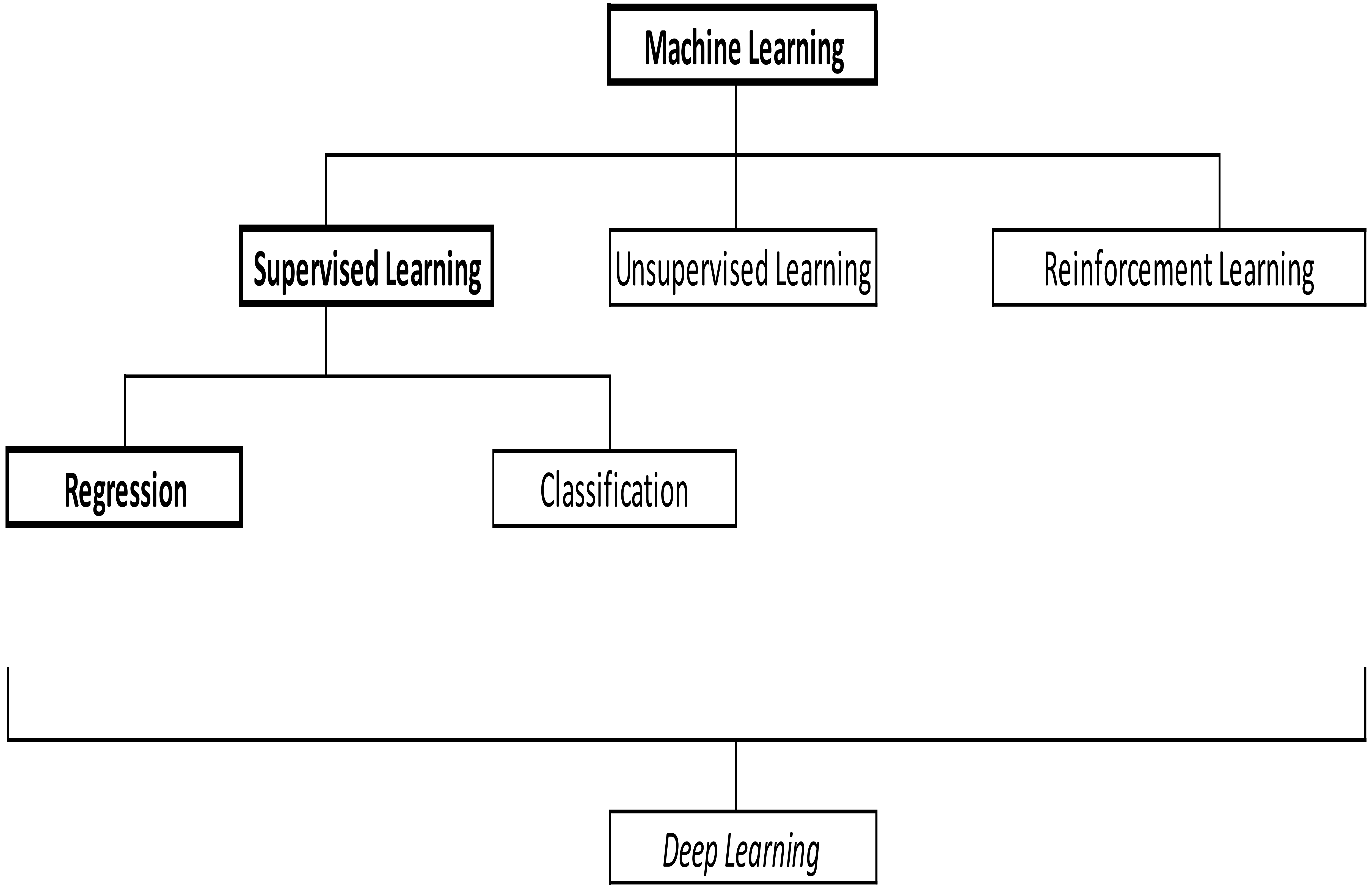
Machine Learning

- **Machine Learning "the study of algorithms that allow computer programs to automatically improve through experience" (Mitchell 1997)**

- **Machine learning is an approach to the field of Artificial Intelligence**
  Systems trained to recognize patterns within data to acquire knowledge (Goodfellow, Bengio and Courville 2016).

- **Earlier attempts to build AI systems = hard code knowledge into knowledge bases … but doesn't work for highly complex tasks e.g. image recognition, scene understanding and inferring semantic concepts (Bengio 2009)**

- **ML Paradigm – feed data to the machine and let it figure out what is important from the data!**
  Deep Learning represents a specific approach to ML.



Supervised Learning

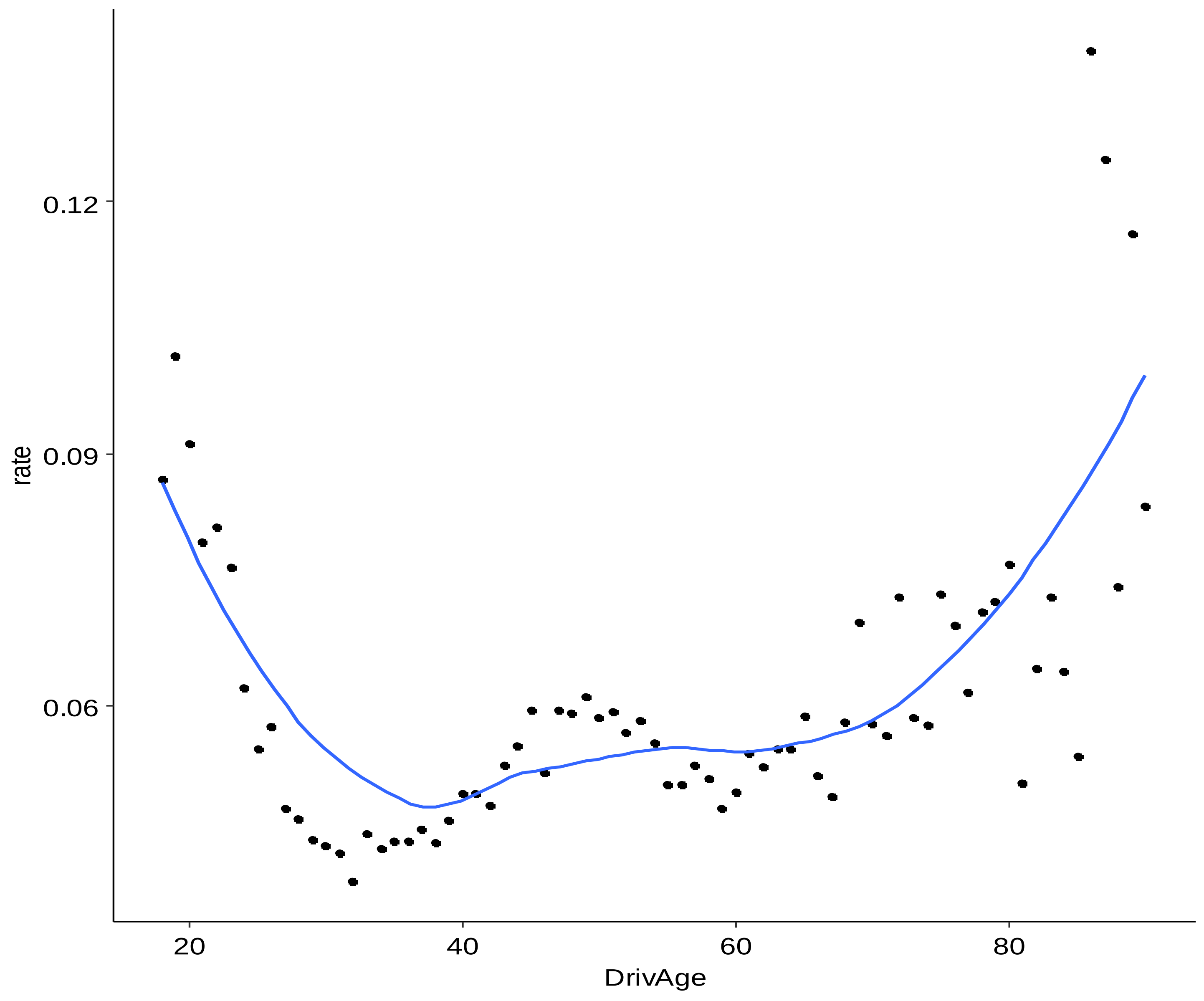
- **Supervised learning = application of machine learning to datasets that contain features and outputs with the goal of predicting the outputs from the features (Friedman, Hastie and Tibshirani 2009).**

- **Feature engineering - Suppose we realize that Claims depends on Age^2 => enlarge feature space by adding Age^2 to data. Other options – add interactions/basis functions e.g. splines**

y (*outputs*)

X (*features*)

```
> freMTPL2freq
          IDpol ClaimNb    Exposure Area VehPower VehAge DrivAge BonusMalus VehBrand  VehGas Density Region DrivAge_2
     1:       1       1 0.100000000    D        5      0      55         50      B12 Regular    1217    R82      3025
     2:       3       1 0.770000000    D        5      0      55         50      B12 Regular    1217    R82      3025
     3:       5       1 0.750000000    B        6      2      52         50      B12  Diesel      54    R22      2704
     4:      10       1 0.090000000    B        7      0      46         50      B12  Diesel      76    R72      2116
     5:      11       1 0.840000000    B        7      0      46         50      B12  Diesel      76    R72      2116
    ---
678009: 6114326       0 0.002739726    E        4      0      54         50      B12 Regular    3317    R93      2916
678010: 6114327       0 0.002739726    E        4      0      41         95      B12 Regular    9850    R11      1681
678011: 6114328       0 0.002739726    D        6      2      45         50      B12  Diesel    1323    R82      2025
678012: 6114329       0 0.002739726    B        4      0      60         50      B12 Regular      95    R26      3600
678013: 6114330       0 0.002739726    B        7      6      29         54      B12  Diesel      65    R72       841
>
```



Goal: Explaining or Predicting?

- **Which of the following are an ML technique?**
  Linear regression and friends (GLM/GLMM)
  Generalized Additive model (GAM)
  Exponential Smoothing
  Chain-Ladder and Bornhuetter-Ferguson

- **It depends on the goal:**
  Are we building a causal understanding of the world (inferences from unbiased coefficients)?
  Or do we want to make predictions (bias-variance trade-off)?

- **Distinction between tasks of predicting and explaining, see Shmueli (2010). Focus on predictive performance leads to:**

  Building algorithms to predict responses instead of specifying a stochastic data generating model (Breiman 2001)…
  … favouring models with good predictive performance at expense of interpretability.
  Accepting bias in model coefficients if this is expected to reduce the overall prediction error.
  Quantifying predictive error (i.e. out-of-sample error)

- **ML relies on a different approach to building, parameterizing and testing statistical models, based on statistical learning theory, and focuses on predictive accuracy.**

Actuaries solve “umbrella problems”

- **Actuarial work involves making predictions, which are either used directly (General Insurance) or indirectly (Life)**
- **Consider two different problems – <u>taking an umbrella</u> compared to <u>investing in a rain-dance during a drought</u>**

	Umbrella problem	**Rain-dance problem**
Problem	Should I take an umbrella?	Should I invest in a rain-dance?
Task	Prediction - Will it rain?	Causal inference - Will the rain-dance cause it to rain?
Tool	Machine Learning	Unbiased regression – or post-selection inference

- **Actuaries most often focus on “umbrella problems” (Kleinberg, Ludwig, Mullainathan et al. 2015)**
- **Decisions of actuaries based on models generally do not need to be based on causal understanding (unless required by a regulator or practising standards) but interpretability helps the acceptance of models**
- **Can lead to new theoretical insights – see Golden, Brockett, Ai et al. (2016) on credit scores**

Recipe for Actuarial Data Science

- **Actuarial problems are often supervised regressions =>**
- **If an actuarial problem can be expressed as a regression, then machine and deep learning can be applied.**
- **Obvious areas of application:**

  P&C pricing
  IBNR reserving
  Experience analysis
  Mortality modelling
  Lite valuation models

- **But don't forget about unsupervised learning either!**

Agenda

- **Introduction**
- **Machine Learning**
- **<u>Deep Learning</u>**
- **Tools of the Trade**
- **Selected Applications**
- **Challenges**
- **Conclusion**

Actuarial Modelling

- **Actuarial modelling tasks vary between:**

Human input

Empirically/data driven
NL pricing
Approximation of nested Monte Carlo
Portfolio specific mortality

Feature engineering

Model Driven
IBNR reserving (Chain-Ladder)
Life experience analysis (AvE)
Capital modelling (Log-normal/Clayton copula)
Mortality forecasting (Lee-Carter)

Model Specification

- **Feature engineering = data driven approach to enlarging a feature space using human ingenuity and expert domain knowledge**
  Apply various techniques to the raw input data – PCA/splines
  Enlarge features with other related data (economic/demographic)

- **Model specification = model driven approach where define structure and form of model (often statistical) and then find the data that can be used to fit it**

Issues with Traditional Approach

- **In many domains, including actuarial science, traditional approach to designing machine learning systems relies on human input for feature engineering or model specification.**

- **Three arguments against traditional approach:**

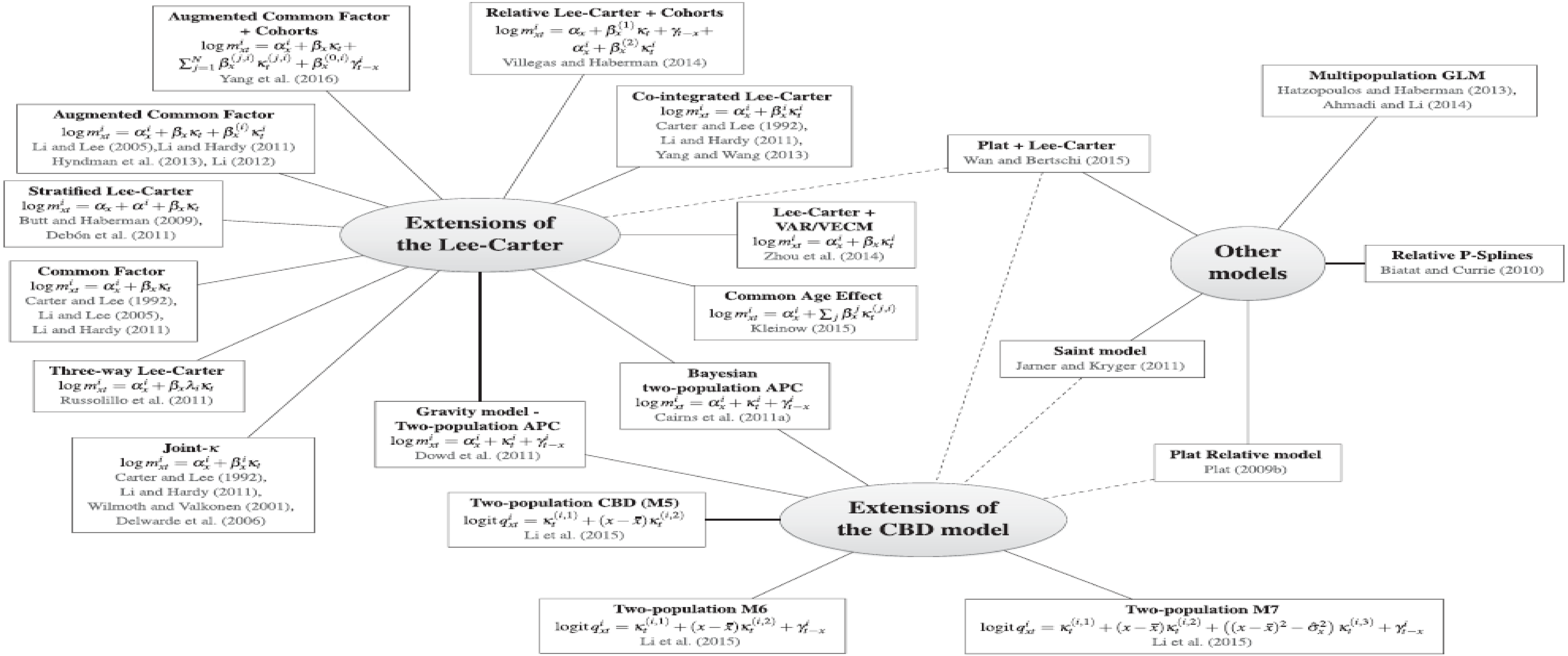
  Complexity – which are the relevant features to extract/what is the correct model specification? Difficult with very high dimensional, unstructured data such as images or text. (Bengio 2009; Goodfellow, Bengio and Courville 2016)

  Expert knowledge – requires suitable prior knowledge, which can take decades to build (and might not be transferable to a new domain) (LeCun, Bengio and Hinton 2015)

  Effort – designing features is time consuming/tedious => limits scope and applicability (Bengio, Courville and Vincent 2013; Goodfellow, Bengio and Courville 2016)

- **Within actuarial modelling, complexity is not only due to unstructured data. Many difficult problems of model specification arise when performing actuarial tasks at a large scale:**
  Multi-LoB IBNR reserving
  Mortality forecasting for multiple populations

Complexity: Multi-population Mortality Modelling



- Diagram excerpted from Villegas, Haberman, Kaishev et al. (2017)

Representation Learning

- **Representation Learning = ML technique where algorithms automatically design features that are optimal (in some sense) for a particular task**

- **Traditional examples are PCA (unsupervised) and PLS (supervised):**

  PCA produces features that summarize directions of greatest variance in feature matrix

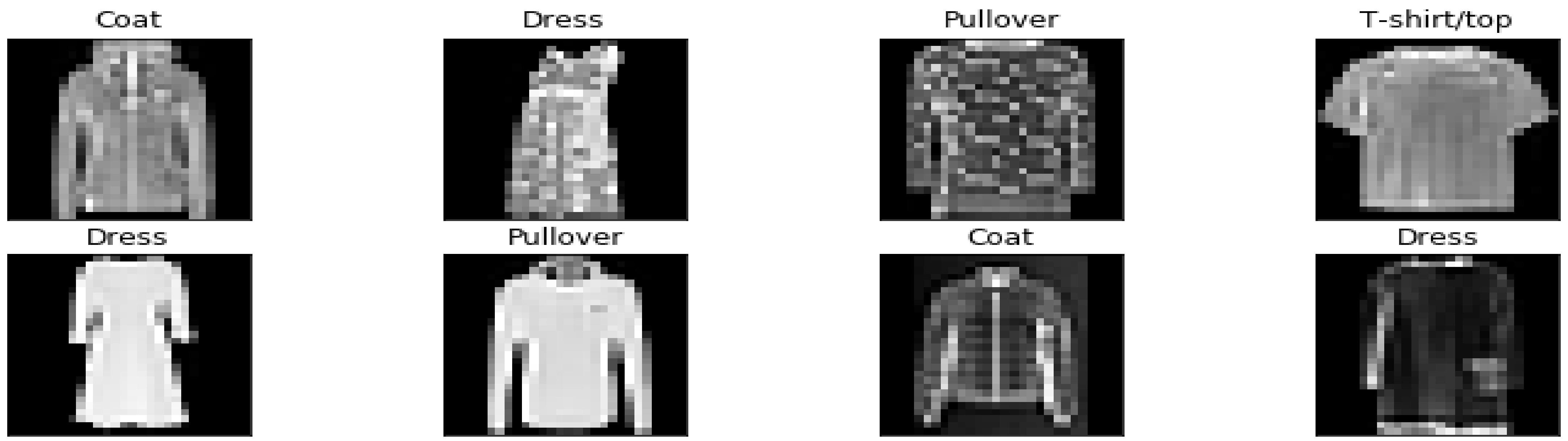
  PLS produces features that maximize covariance with response variable (Stone and Brooks 1990)

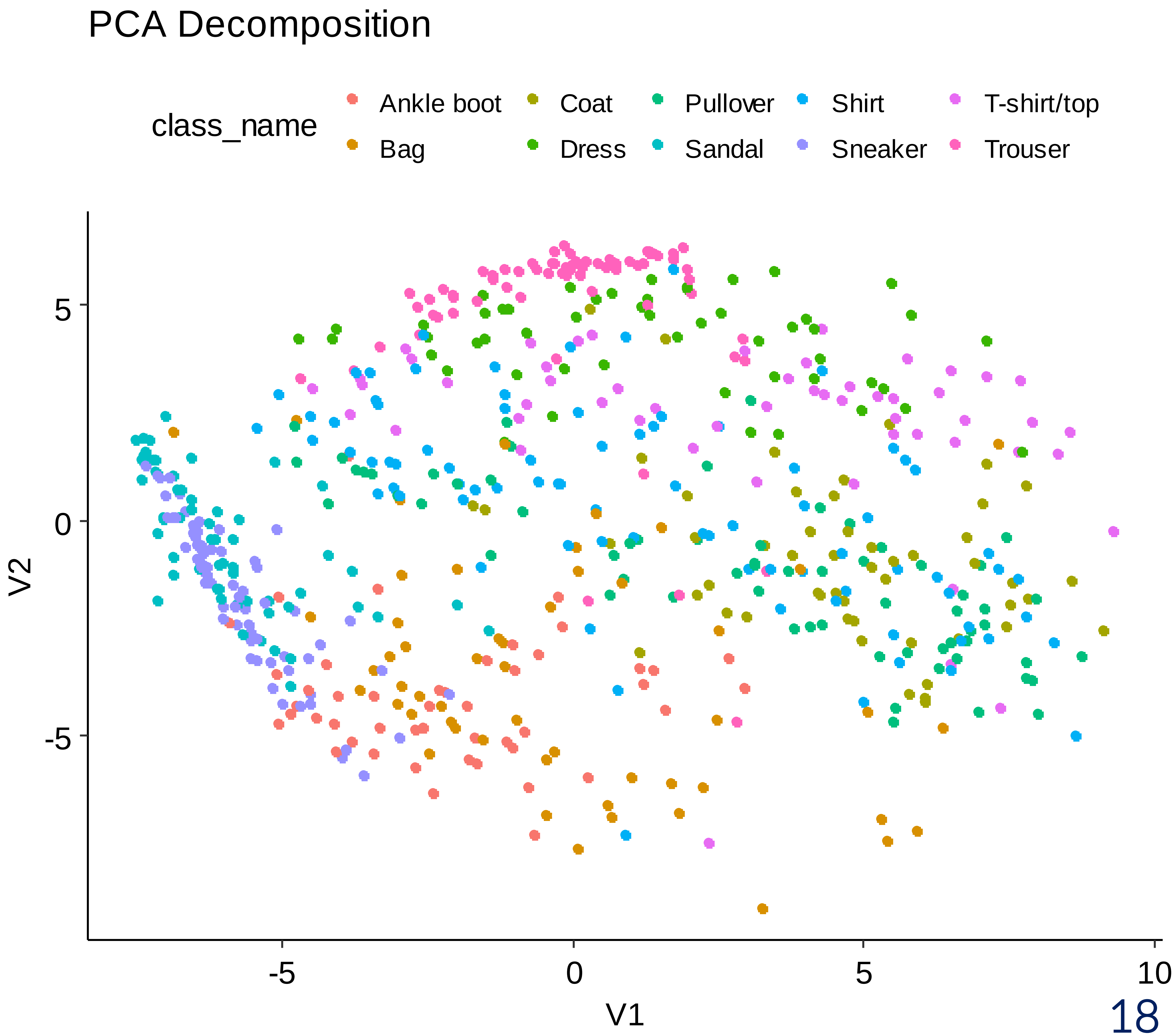
- **Feature space then comprised of learned features which can be fed into ML/DL model**

- **BUT: Simple/naive RL approaches often fail when applied to high dimensional data**

Example: Fashion-MNIST (1)

- **Inspired by Hinton and Salakhutdinov (2006)**
- **Fashion-MNIST –70 000 images from Zolando of common items of clothing**
- **Grayscale images of 28x28 pixels**
- **Classify the type of clothing**
- **Applied PCA directly to the images - results do not show much differentiation between classes**





Deep Learning

- **Deep Learning = representation learning technique that automatically constructs hierarchies of complex features to represent abstract concepts**

  Features in lower layers composed of simpler features constructed at higher layers => complex concepts can be represented automatically

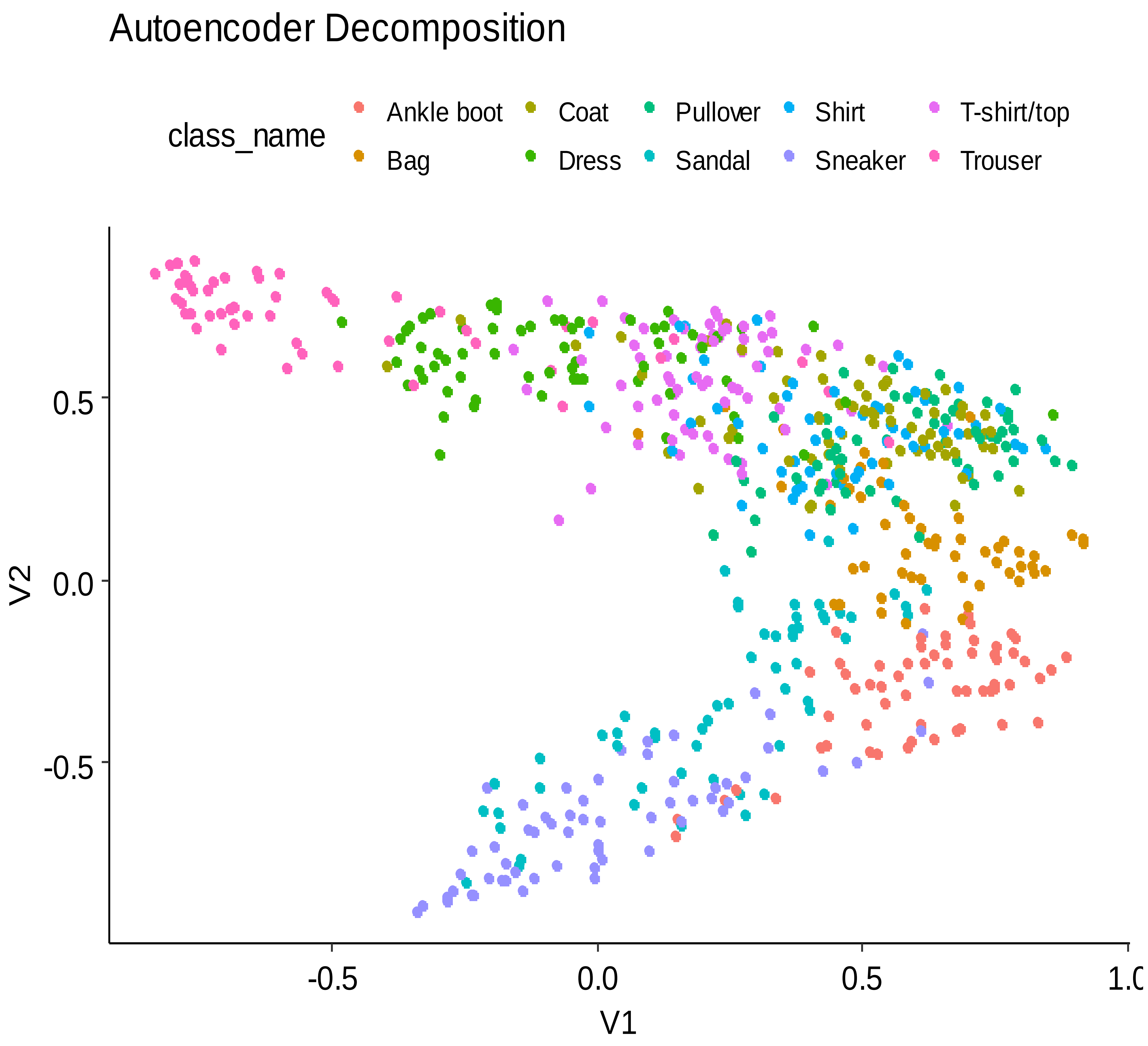
- **Typical example of deep learning is feed-forward neural networks, which are multi-layered machine learning models, where each layer learns a new representation of the features.**

- **The principle: Provide raw data to the network and let it figure out what and how to learn.**

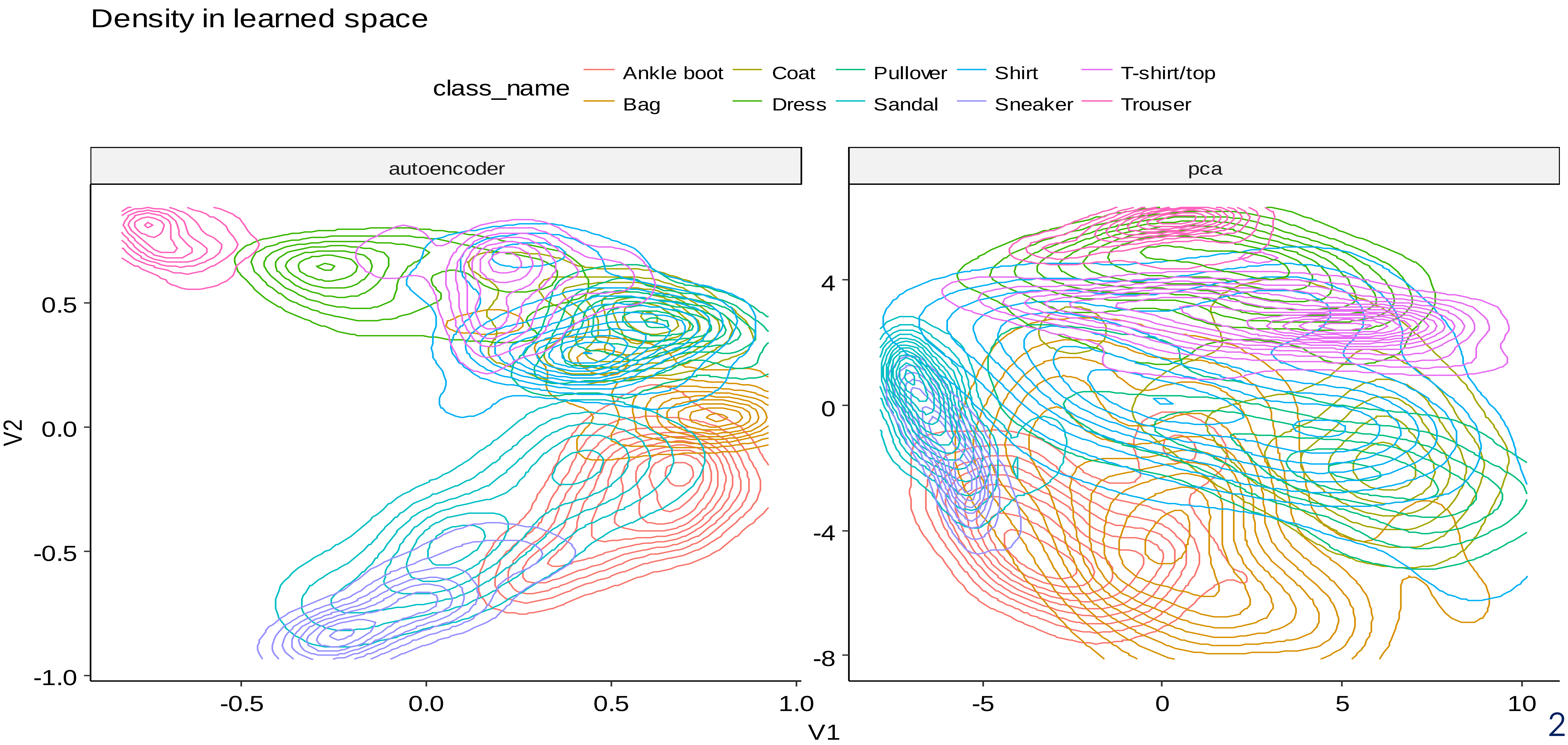
- **Desiderata for AI by Bengio (2009): “Ability to learn with little human input the low-level, intermediate, and high-level abstractions that would be useful to represent the kind of complex functions needed for AI tasks.”**

Example: Fashion-MNIST (2)

- **Applied a deep autoencoder to the same data (trained in unsupervised manner)**
  - Type of non-linear PCA
- **Differences between some classes much more clearly emphasized**
- **Deep representation of data automatically captures meaningful differences between the images without (much) human input**
- **Automated feature/model specification**
- **Aside – feature captured in unsupervised learning might be useful for supervised learning too.**
- **Goodfellow, Bengio and Courville (2016) : "basic idea is features useful for the unsupervised task also be useful for the supervised learning task"**



Fashion-MNIST – Density Plot



Deep Learning for Actuarial Modelling

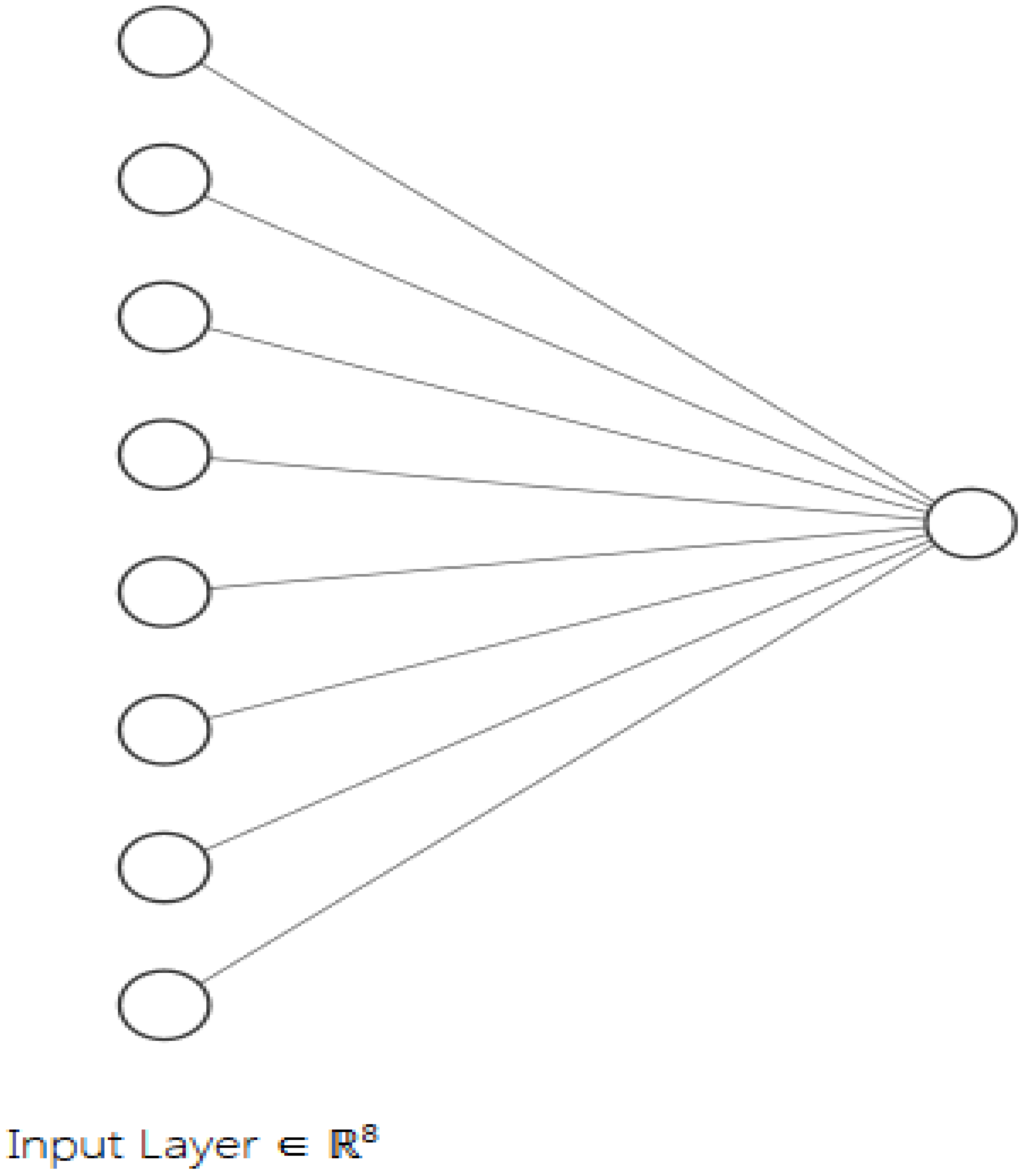
- **Actuarial tasks vary between Empirically/data driven and Model Driven**
- **Both approaches traditionally rely on manual specification of features or models**
- **Deep learning offers an empirical solution to both types of modelling task – feed data into a suitably deep neural network => learn an optimal representation of input data for task**
- **Exchange of model specification for a new task => architecture specification**
- **Opportunity – improve best estimate modelling**
- **Deep learning comes at a (potential) cost – relying on a learned representation means less understanding of models, to some extent**

Agenda

- **Introduction**
- **Machine Learning**
- **Deep Learning**
- **<u>Tools of the Trade</u>**
- **Selected Applications**
- **Challenges**
- **Conclusion**

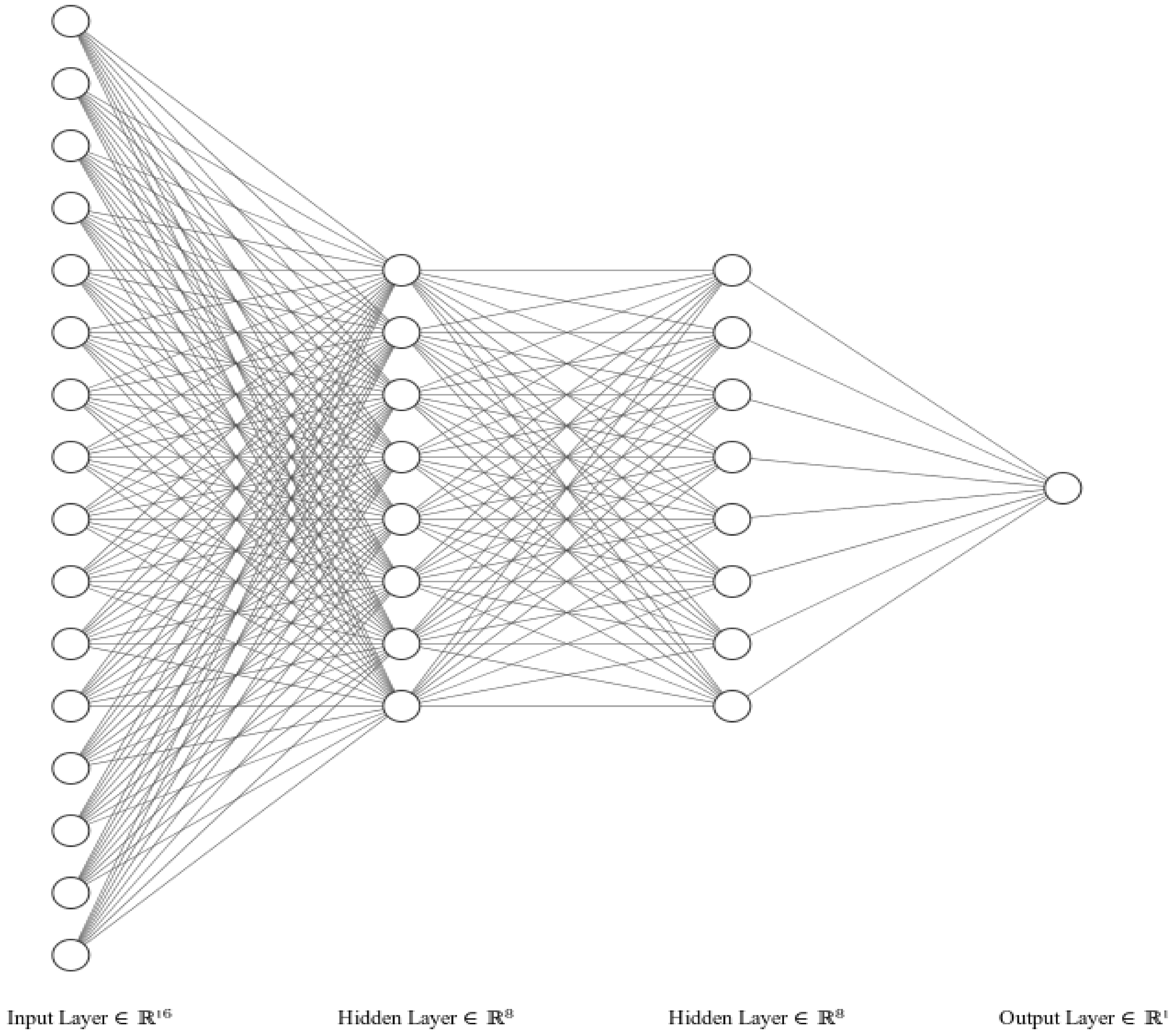
Single Layer NN = Linear Regression

- **Single layer neural network**
  Circles = variables
  Lines = connections between inputs and outputs
- **Input layer holds the variables that are input to the network…**
- **… multiplied by weights (coefficients) to get to result**
- **Single layer neural network is a GLM!**



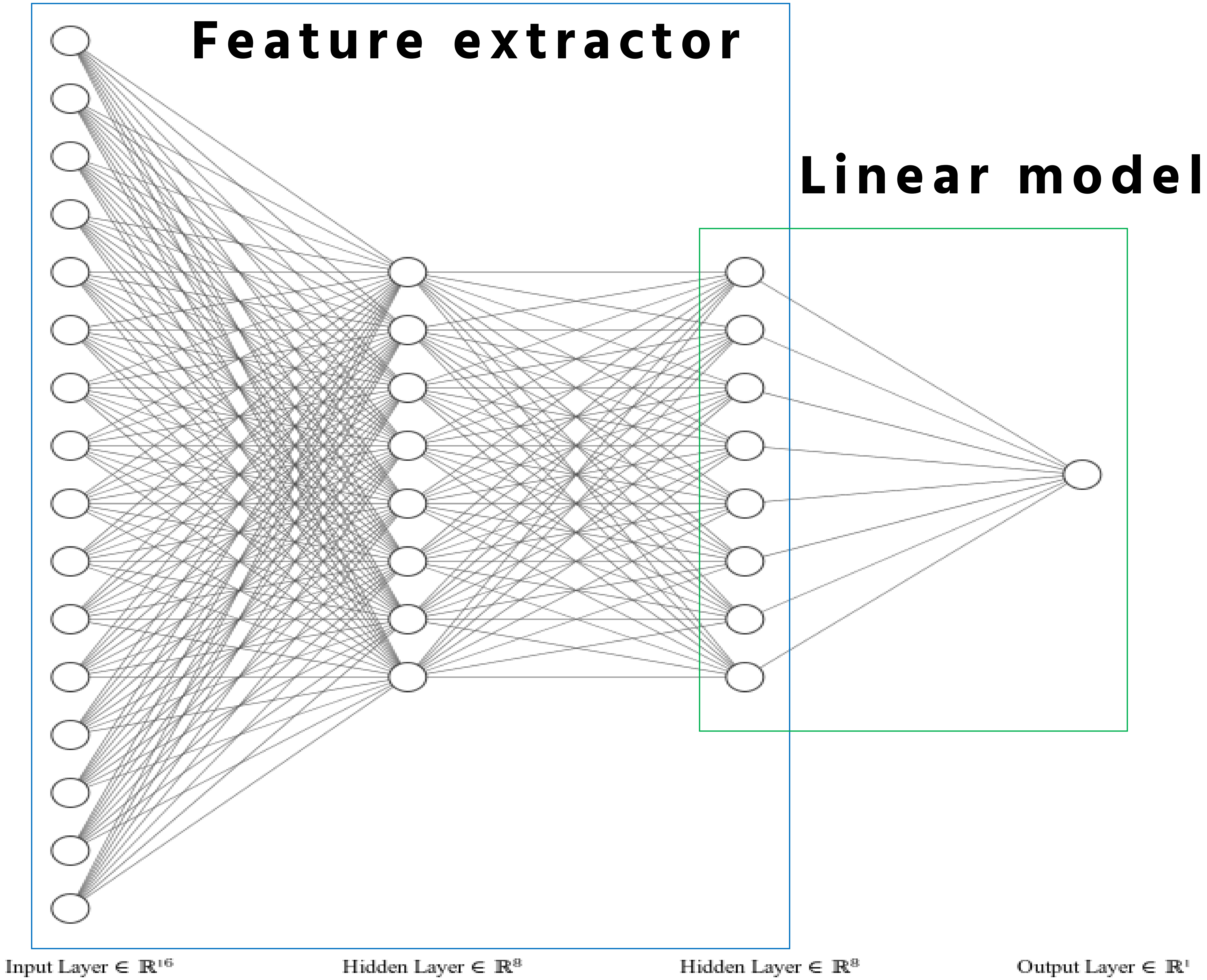
Deep Feedforward Net

- **Deep = multiple layers**
- **Feedforward = data travels from left to right**
- **Fully connected network (FCN) = all neurons in layer connected to all neurons in previous layer**
- **More complicated representations of input data learned in hidden layers - subsequent layers represent regressions on the variables in hidden layers**



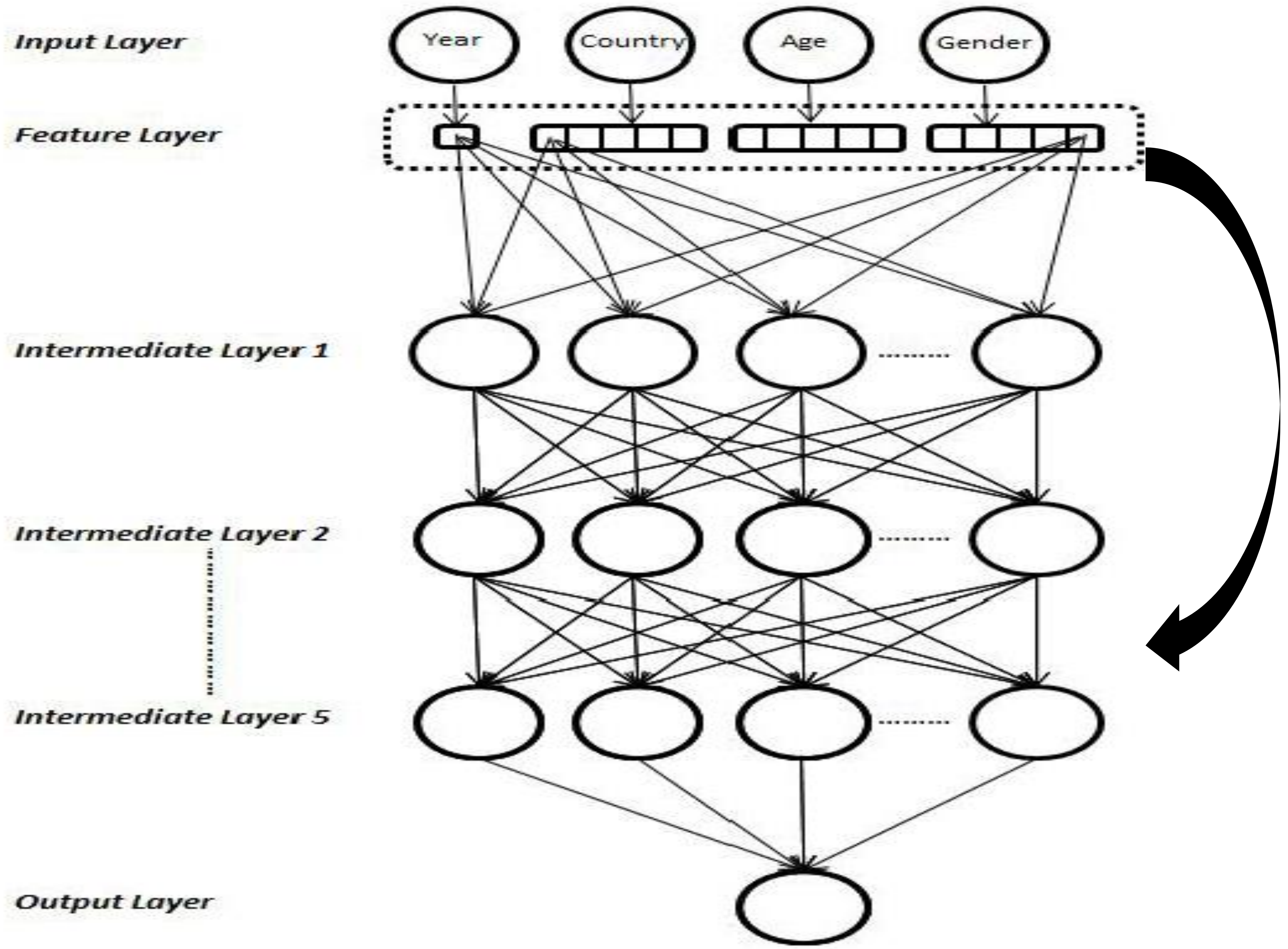
FCN generalizes GLM

- Intermediate layers = representation learning, guided by supervised objective.
- Last layer = (generalized) linear model, where input variables = new representation of data
- No need to use GLM – strip off last layer and use learned features in, for example, XGBoost
- Or mix with traditional method of fitting GLM



Example – Lee-Carter Neural Net

- **Multi-population mortality forecasting model (Richman and Wüthrich 2018)**
- **Supervised regression on HMD data (inputs = Year, Country, Age; outputs = mx)**
- **5 layer deep FCN**
- **Generalizes the LC model**



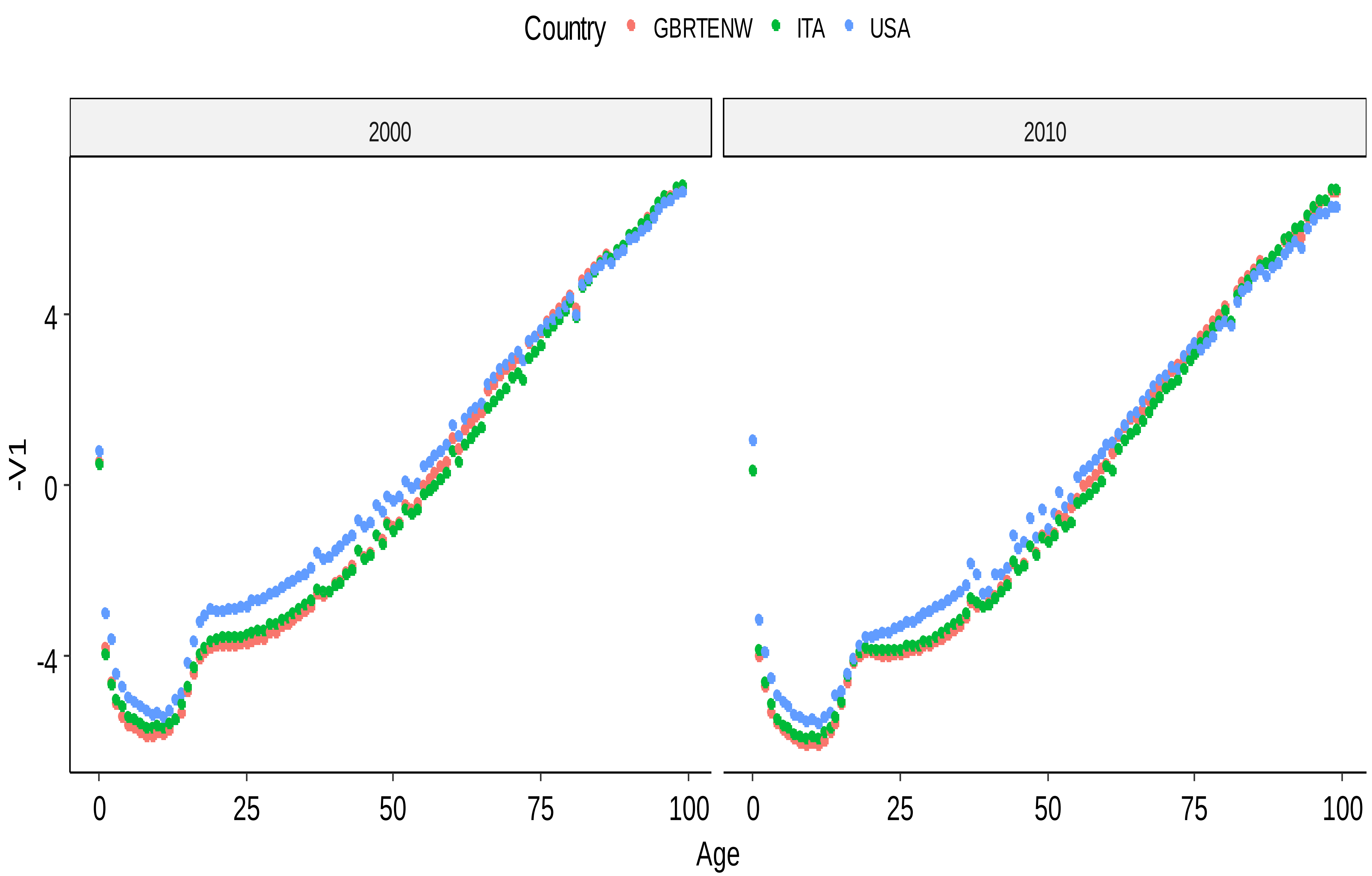
Features in last layer of network

- **Representation = output of last layer (128 dimensions) with dimension reduced using PCA**

- **Can be interpreted as relativities of mortality rates estimated for each period**

- **Output shifted and scaled to produce final results**

- **Generalization of Brass Logit Transform where base table specified using NN (Brass 1964)**



$y_x = a + b * z_x$, where:

y_x = *logit of mortality at age x*
a,b = regression coefficients
z_x = *logit of reference mortality*

Specialized Architectures

- **Most modern examples of DL achieving state of the art results on tasks rely on using specialized architectures i.e. not simple fully connected networks**

- **Key principle - Use architecture that expresses useful priors (inductive bias) about the data => Achievement of major performance gains**

  Embedding layers – categorical data (or real values structured as categorical data)

  Autoencoder – unsupervised learning

  Convolutional neural network – data with spatial/temporal dimension e.g. images and time series

  Recurrent neural network – data with temporal structure

  Skip connections – makes training neural networks easier

- **Section ends with example of fine tuning a specialized architecture for a new task**

(Some) Actuarial Applications of DL

	Pricing	Reserving	Telematics	Mortality Forecasting	Quantitative Risk Management
Feed-forward Nets	• Ferrario, Noll and Wüthrich (2018) • Noll, Salzmann and Wüthrich (2018) • Wüthrich and Buser (2018)	• Castellani, Fiore, Marino et al. (2018) • Doyle and Groendyke (2018) • Gabrielli and Wüthrich (2018) • Hejazi and Jackson (2016, 2017) • Wüthrich (2018) • Zarkadoulas (2017)	• Gao and Wüthrich (2017) • Gao, Meng and Wüthrich (2018) • Gao, Wüthrich and Yang (2018)		• Castellani, Fiore, Marino et al. (2018) • Hejazi and Jackson (2016, 2017)
Convolutional Neural Nets			• Gao and Wüthrich (2019)		
Recurrent Neural Nets		• Kuo (2018a, 2018b)		• Nigri, Levantesi, Marino et al. (2019)	
Embedding Layers	• Richman (2018) • Schelldorfer and Wüthrich (2019) • Wüthrich and Merz (2019)	• Gabrielli, Richman and Wüthrich (2018) • Gabrielli (2019)		• Richman and Wüthrich (2018)	
Autoencoders			• Richman (2018)	• Hainaut (2018) • Richman (2018)	

Embedding Layer – Categorical Data

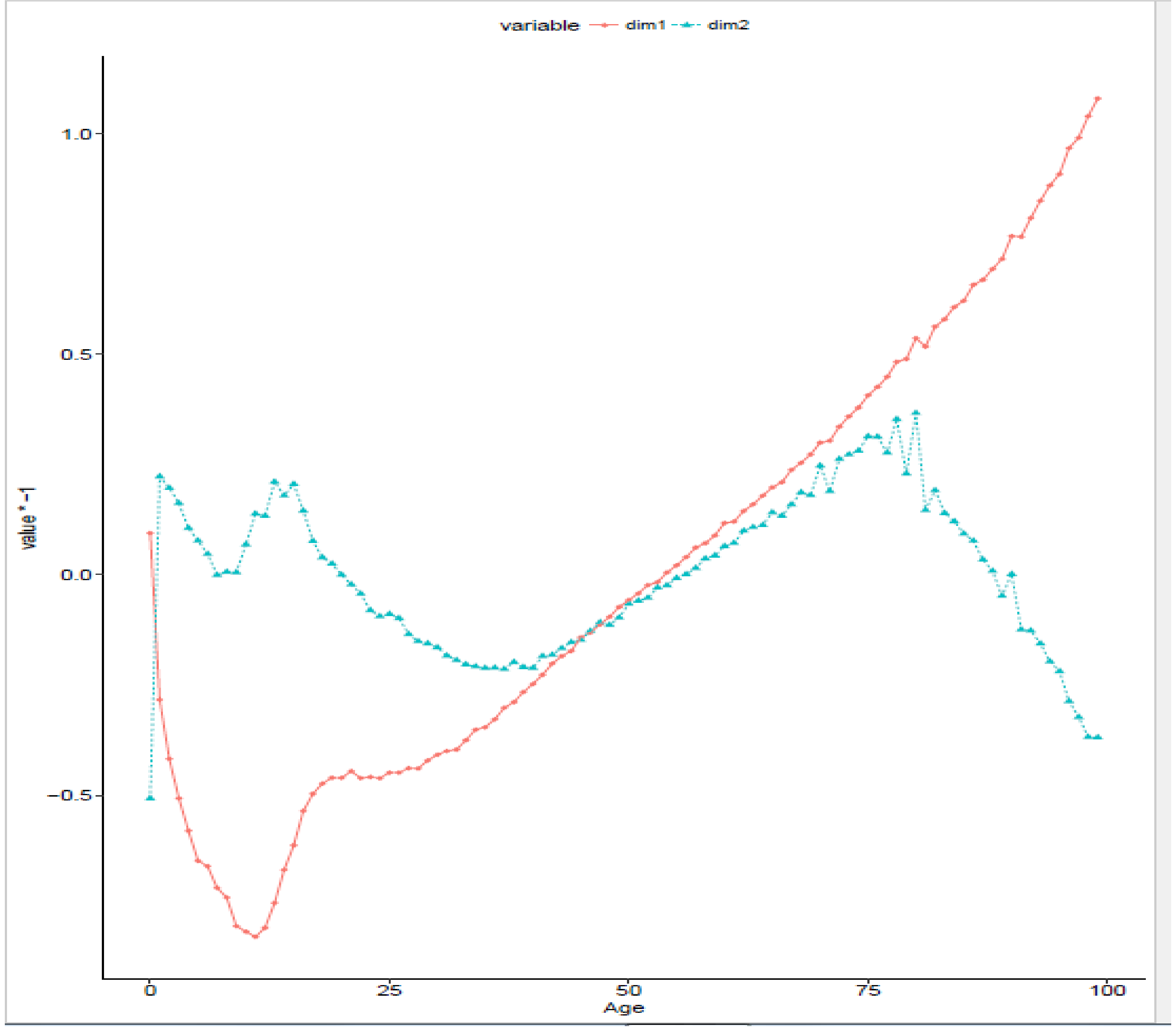
- **One hot encoding expresses the prior that categories are orthogonal => similar observations not categorized into groups**
- **Traditional actuarial solution – credibility**
- **Embedding layer prior – related categories should cluster together:**
  - Learns dense vector transformation of sparse input vectors and clusters similar categories together
  - Can pre-calibrate to MLE of GLM models, leading to CANN proposal of Wüthrich and Merz (2019)

	Actuary	Accountant	Quant	Statistician	Economist	Underwriter
Actuary	1	0	0	0	0	0
Accountant	0	1	0	0	0	0
Quant	0	0	1	0	0	0
Statistician	0	0	0	1	0	0
Economist	0	0	0	0	1	0
Underwriter	0	0	0	0	0	1

	Finance	Math	Stastistics	Liabilities
Actuary	0.5	0.25	0.5	0.5
Accountant	0.5	0	0	0
Quant	0.75	0.25	0.25	0
Statistician	0	0.5	0.85	0
Economist	0.5	0.25	0.5	0
Underwriter	0	0.1	0.05	0.75

Learned embeddings

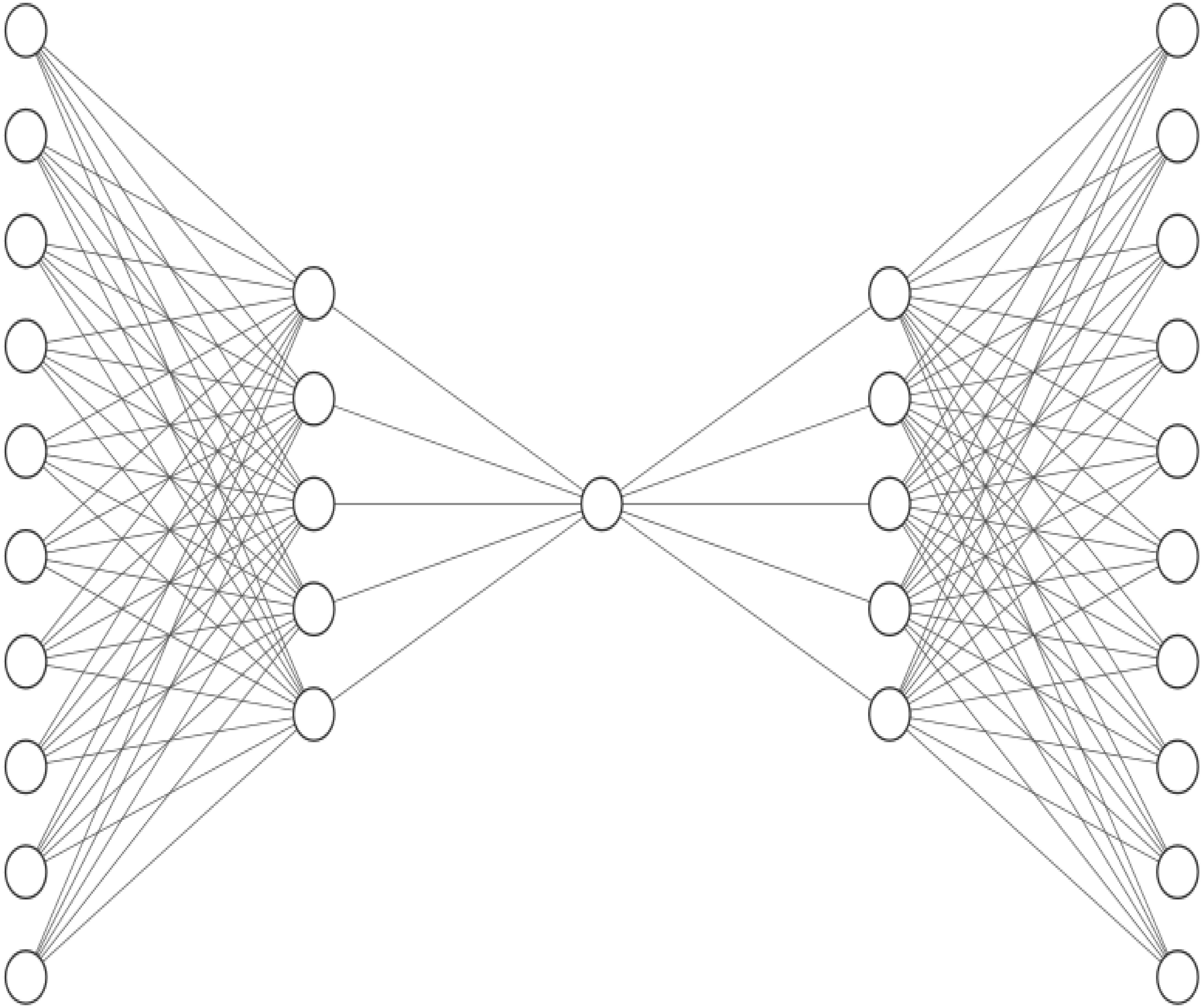
- **Age embeddings extracted from LC NN model**
- **Five dimensions reduced using PCA**
- **Age relativities of mortality rates**
- **In deeper layers of network, combined with other inputs to produce representations specific to:**
    - Country
    - Gender
    - Time
- **First dimension of PCA is shape of lifetable**
- **Second dimension is shape of child, young and older adult mortality relative to middle age and oldest age mortality**



Autoencoder – Unsupervised Learning

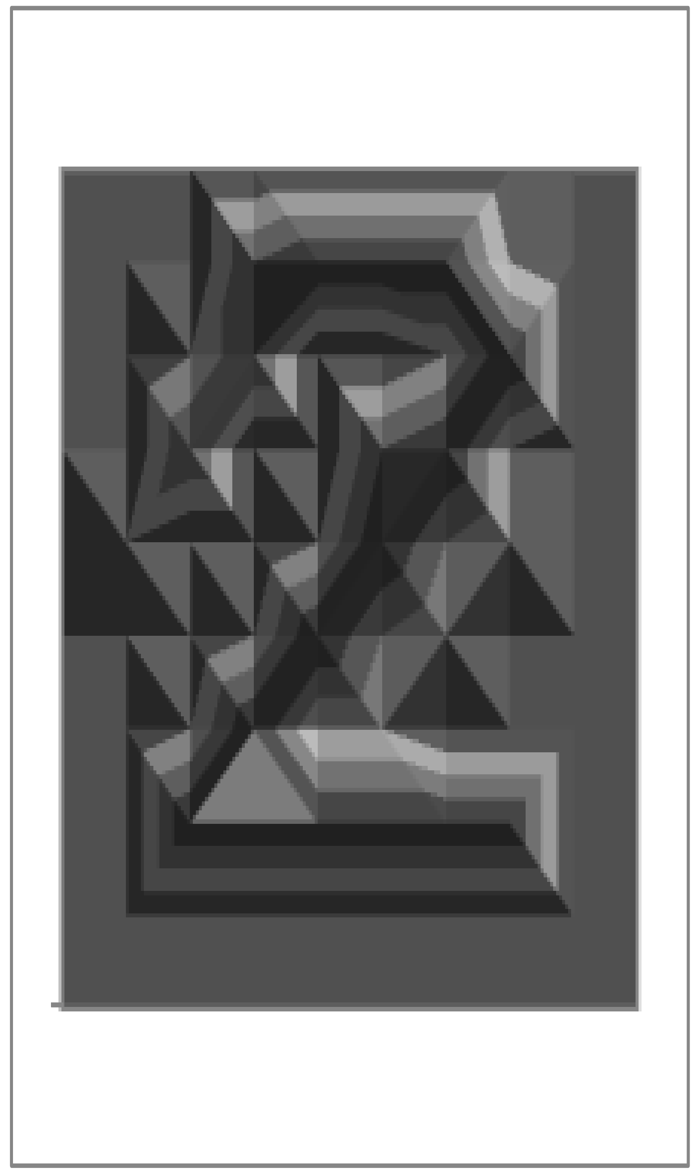
- **Autoencoder = network is trained to produce output equal to the input**
  - Vector input and output
  - Bottleneck in middle restricts dimension of encoded data…
  - … in this example, to 1, but can be to multiple dimensions
  - Performs a type of non-linear PCA

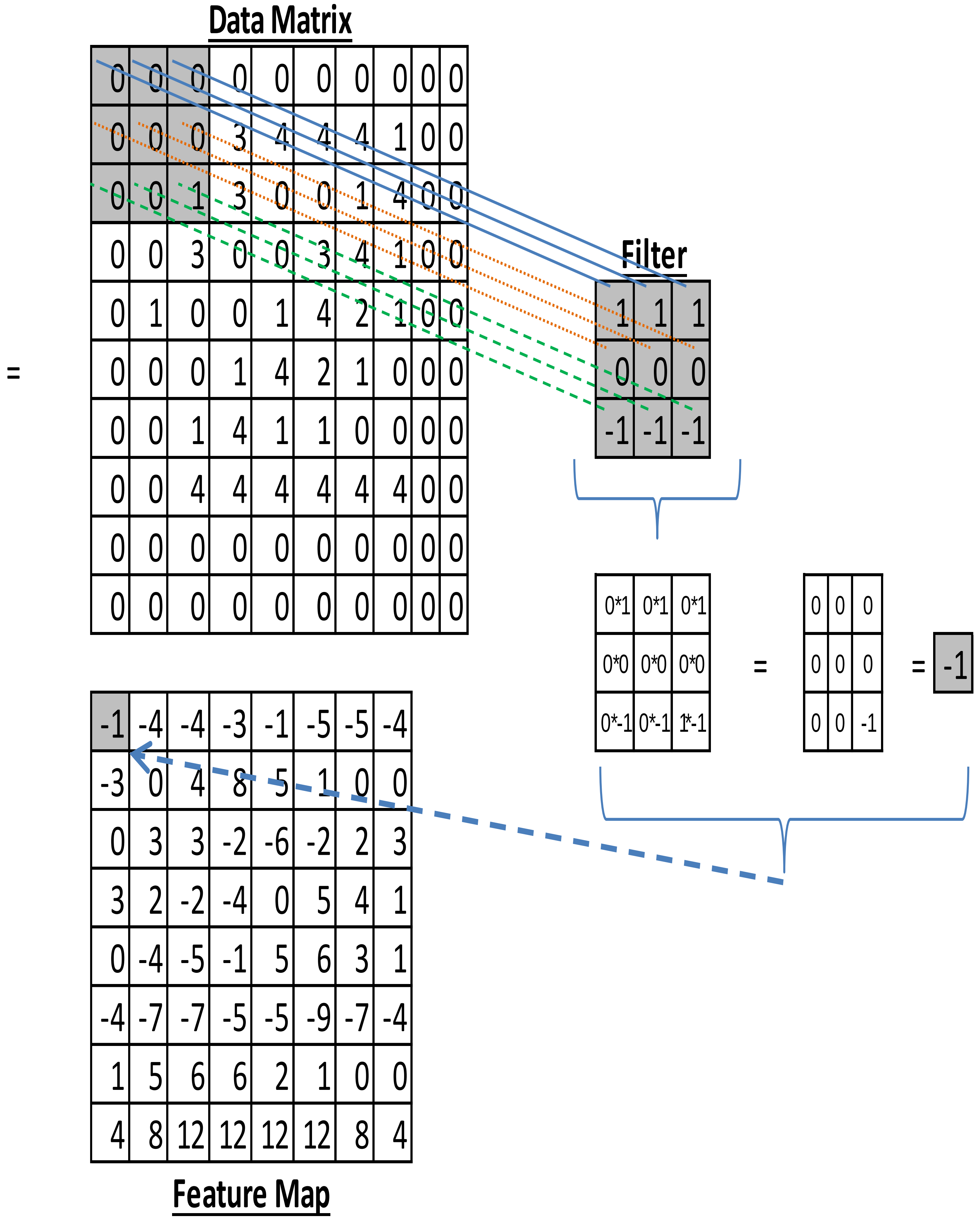
- **Bottleneck layer expresses prior that data can be summarized in only a few dimensions**



Convolutional NN - Images

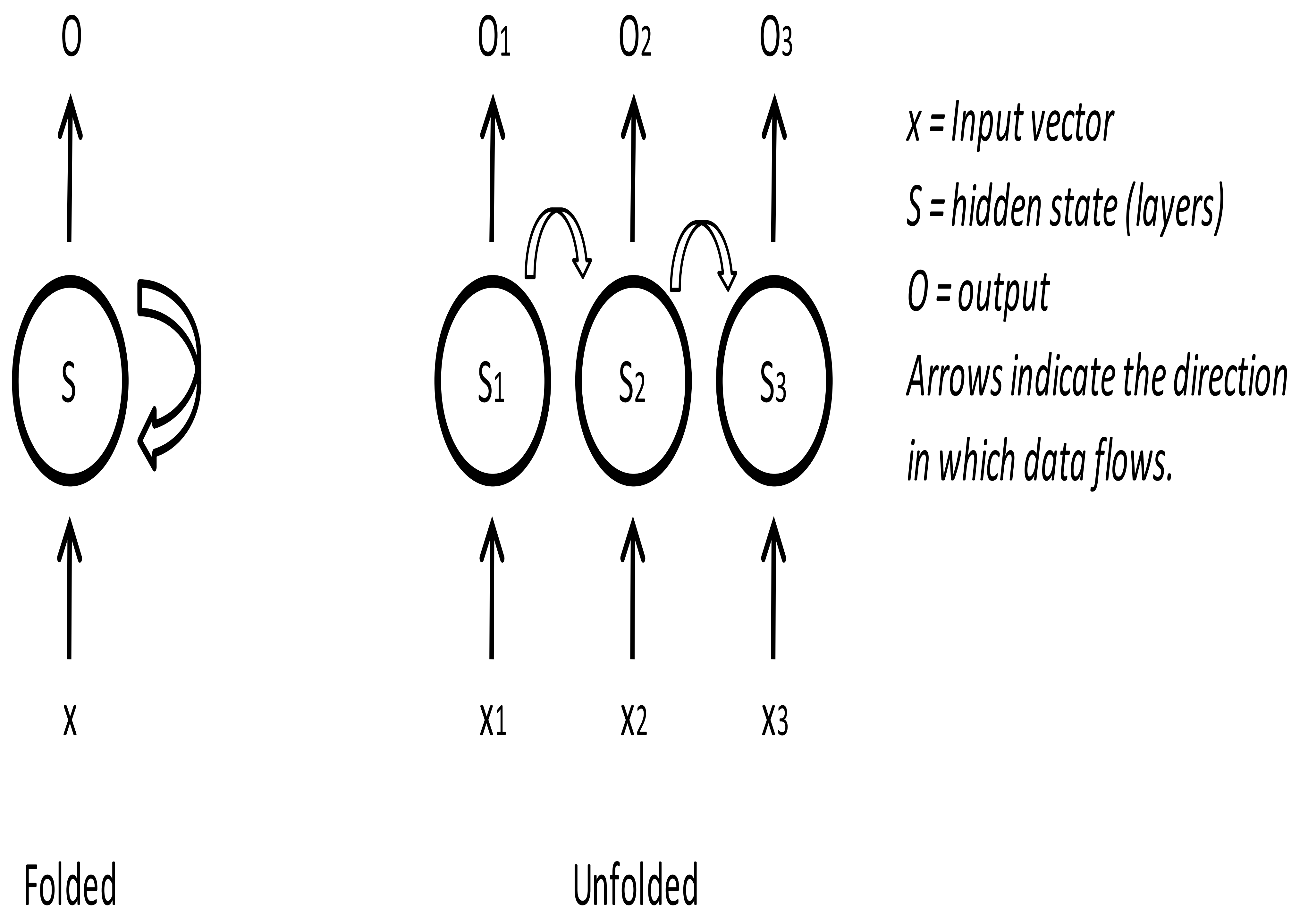
- **Prior - features in images are position invariant i.e. can recognize at any position within an image**
  Also applies to audio/speech and text/time series data
- **Convolutional network is locally connected and shares weights => expresses prior of position invariance**
  Far fewer parameters than FCN
- **Each neuron (i.e. feature map) in network derived by applying filter to input data**
  Weights of filter learned when fitting network
  Multiple filters can be applied





Recurrent NN – Temporal data

- **Data with temporal structure implies that previous observations should influence the current observation**
- **Recurrent network maintains state of hidden neurons over time**
  - Past representation useful for current prediction i.e. network has a 'memory'
- **Several implementations of the recurrent concept which control how network remembers and forgets state**



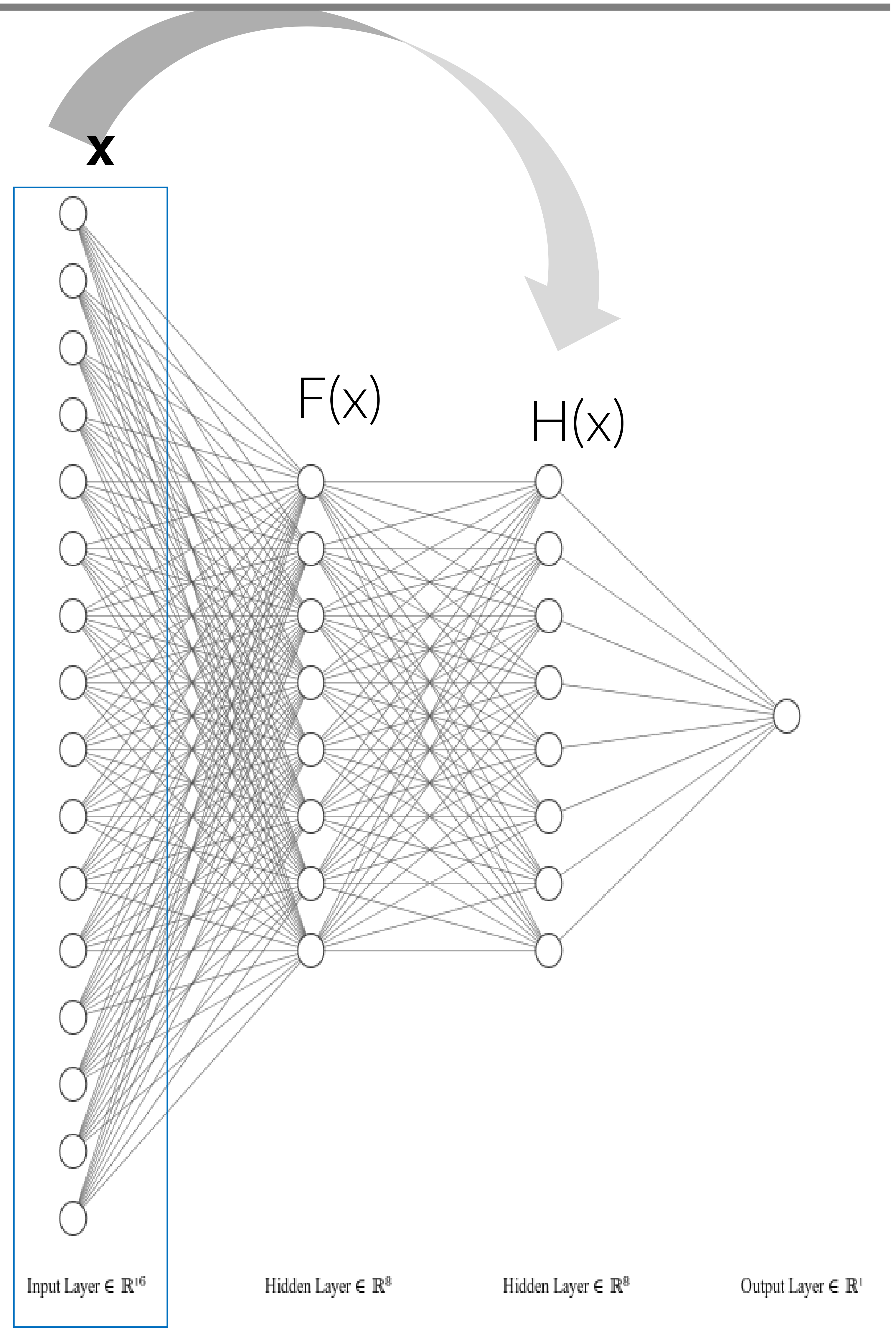
Skip Connections

- **Extra connections between disconnected layers of the NN**

- **NN then only needs to learn a “residual”:**
  H(x) := x + F(x)

- **Widely used in computer vision but also useful on tabular data**
- **Makes networks easier to optimize**
  Veit, Wilber and Belongie (2016) show that resulting NN functions as an ensemble (can delete layers)
  Greff, Srivastava and Schmidhuber (2016) extend this view by showing that layers learn refined estimates of input representations

- **Allows for combination of simple models together with “neural boosting”**
  Leads to the CANN proposal (Wüthrich and Merz 2019)



Transfer Learning

- **Machine learning problem where model trained on source domain/task reused for target domain/task (Pan and Yang 2009)**

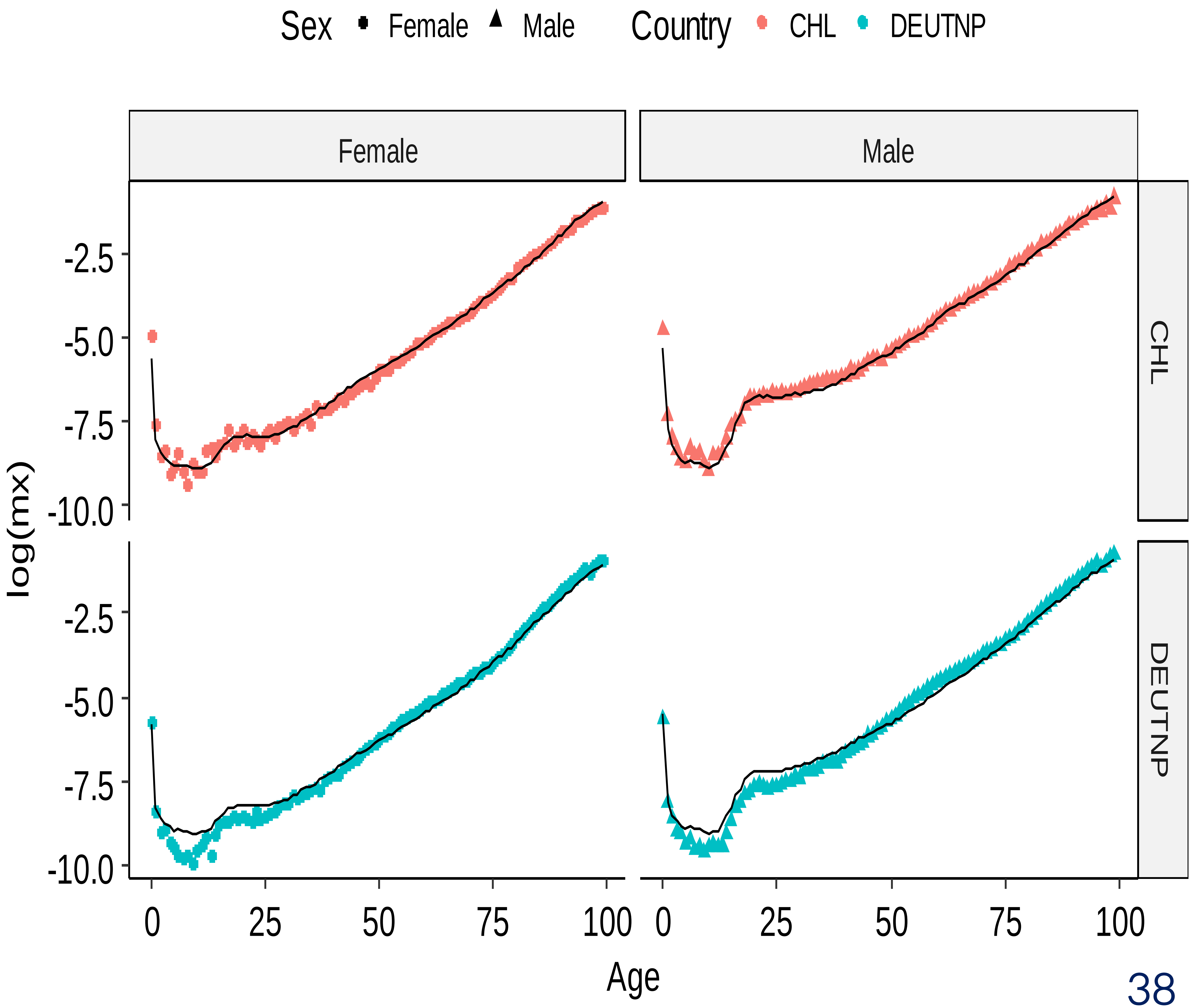
- **Formal definition - Given source/target domain D_S/D_T and source/target task D_S/D_T, improve a predictive function in D_T using D_S/T_S where $D_S \neq D_T$ or $T_S \neq T_T$**

- **According to (Bengio 2012), DL ideal for transfer learning:**
  "it focuses on learning representations and in particular 'abstract' representations, representations that ideally disentangle the factors of variation present in the input."

- **Often useful when target domain does not contain enough data to train a full DL model => use pretrained model as a feature extractor**
  Computer vision – pretrained classification model
  Natural langauge – pretrained language model
  Model is then fine-tuned to adapt it to target domain/task
  See the fast.ai Python library for excellent implementations of transfer learning algorithms

Example: TL in the LC NN model

- **Model relies on disentangled representations for (Country, Sex, Age, Time), implying that:**
  - Can fine tune only the Country representation for new data (i.e $\boldsymbol{D_S} \neq \boldsymbol{D_T}$ but $\boldsymbol{T_S} = \boldsymbol{T_T}$)
- **Used data for Germany/Chile in 1999 to train a new Country embedding i.e. no temporal variation seen by model and projections made for 2015/2008**
- **Results are impressive for adult mortality**



Agenda

- **Introduction**
- **Machine Learning**
- **Deep Learning**
- **Tools of the Trade**
- **Selected Applications**
- **Challenges**
- **Conclusion**

Selected Applications

- **Following examples chosen to showcase ability of deep learning to solve the issues with the traditional actuarial (or ML) approaches.**

- **In most of these instances, deep learning solution outperforms the traditional actuarial or machine learning approach**

- **Complexity – which are the relevant features to extract/what is the correct model specification?**

  Multi-population mortality forecasting
  Multi LoB IBNR reserving
  Non-life pricing

- **Expert knowledge – requires suitable prior knowledge, which can take decades to build**

  Analysis of telematics data

- **Effort – designing relevant features is time consuming/tedious => limits scope and applicability**

  Lite valuation models

Multi-population mortality forecasting

- **Availability of multiple high quality series of mortality rates, but how to translate into better forecasts?**
- **Multi-population models (Kleinow 2015; Li and Lee 2005)**
  - Many competing model specifications, without much theory to guide model selection
  - Relatively disappointing performance of two models (CAE and ACF)
- **Richman and Wüthrich (2018) – deep neural net with embedding layers**
- **Outperforms both single and multiple populations models**

	Model	Average MSE	Median MSE	Best Performance
1	LC_SVD	5.50	2.48	19
2	ACF_SVD_region	3.46	2.50	36
3	ACF_SVD_country	7.30	4.77	9
4	ACF_BP	6.12	3.00	12

	Model	Average MSE	Median MSE	Best Performance
1	LC_SVD	5.50	2.48	33
2	CAE_SVD	4.76	2.35	13
3	CAE2_SVD	12.01	1.79	14
4	CAE2_BP	5.59	3.46	16

	Model	Average MSE	Median MSE	Best Performance
1	LC_SVD	5.50	2.48	7
2	LC_ACF_region	3.46	2.50	10
3	ACF_BP	6.12	3.00	4
4	CAE_BP	5.59	3.46	4
5	DEEP	2.68	1.38	51

Multi LoB IBNR reserving (1)

- **Even using triangles, most reserving exercises are more data rich than assumed by traditional (widely applied) methods (CL/BF/CC):**
  - Incurred/Paid/Outstanding
  - Amounts/Cost per Claim/Claim Counts
  - Multiple LoBs
  - Multiple Companies

- **Traditional solutions:**
  - Munich Chain Ladder (Quarg and Mack 2004) reconciles Incurred and Paid triangles (for single LoB) by adding a correction term to the Chain Ladder formula based on regression
  - Credibility Chain Ladder (Gisler and Wüthrich 2008) derives LDFs for sub-portfolios of a main LoB using credibility theory
  - Double Chain Ladder (Miranda, Nielsen and Verrall 2013) relates incurred claim count triangles to payment triangles

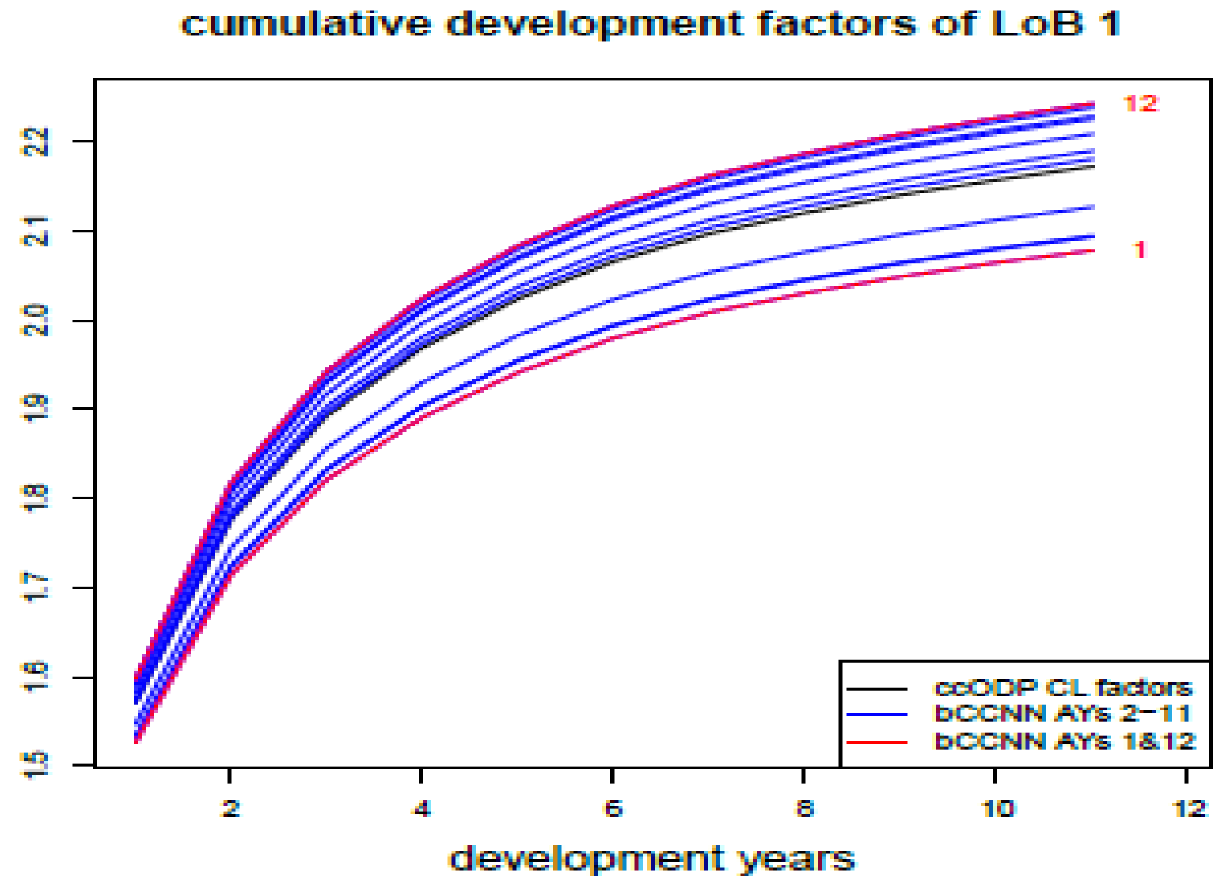
- **Would assume that multi-LoB methods have better predictive performance compared univariate methods, but no study (yet) comparing predictive performance of multi-LoB methods (Meyers (2015) compares several univariate reserving models)**

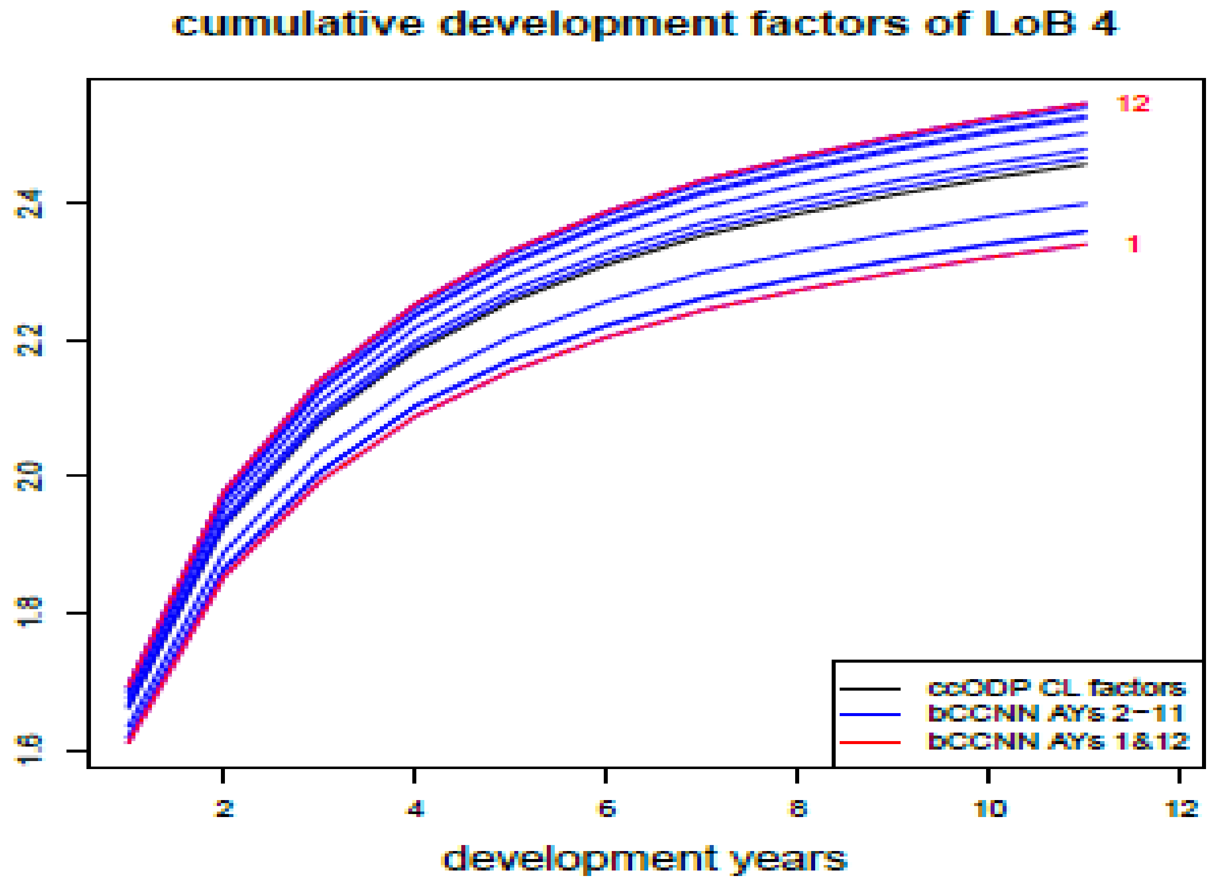
- **General statistical solution for leveraging multiple data sources not proposed**

Multi LoB IBNR reserving (2)

- **Recent work embedding the ODP CL model into a deep neural network (multi-LoB solution)**
- **6 Paid triangles generated using the simulation machine of Gabrielli and Wüthrich (2018)**
  - Know true reserves
  - Relatively small data (12*12*6=478 data points)
- **Gabrielli, Richman and Wüthrich (2018) use classical ODP model plus neural boosting on 6 triangles simultaneously**
  - Dramatically reduced bias compared to ODP model
  - Model learns smooth development factors adjusting for accident year effects
- **Gabrielli (2019) extends model to include both paid and count data**
  - Further reduction in bias versus the previous model

	LoB 1	LoB 2	LoB 3	LoB 4	LoB 5	LoB 6
true reserves R_m^{true}	39,689	37,037	16,878	71,630	72,548	31,117
CL reserves R_m^{CL}	38,569	35,460	15,692	67,574	70,166	29,409
bCCNN reserves R_m^{LoB} (LoBs individually)	39,233	35,899	15,815	70,219	70,936	30,671
bCCNN reserves R_m^{+} (multiple LoBs)	40,271	37,027	16,400	70,563	73,314	30,730

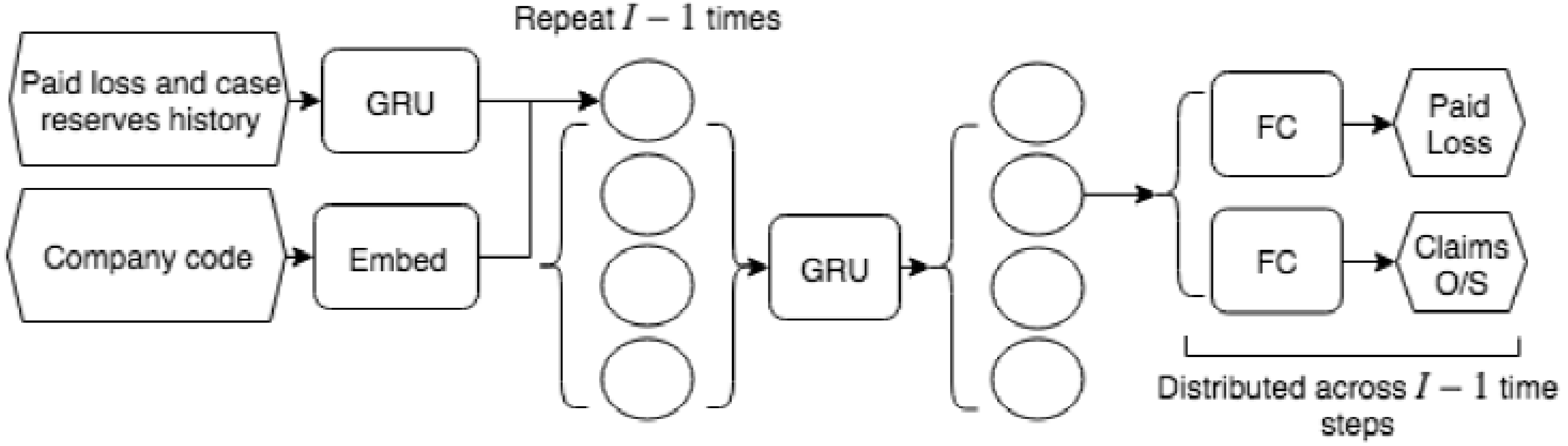




		LoB 1	LoB 2	LoB 3	LoB 4	LoB 5	LoB 6
(i)	true claims reserves R_m^{true}	39'689	37'037	16'878	71'630	72'548	31'117
(ii)	CL reserves R_m^{CL}	38'569	35'460	15'692	67'574	70'166	29'409
(iii)	single NNDODP reserves R_m^{ind}	39'407	36'283	16'123	70'547	71'873	31'092
(iv)	multiple NNDODP reserves R_m^{joint}	40'403	37'172	16'434	70'727	73'513	30'770

Multi LoB IBNR reserving (3)

- **DeepTriangle model of Kuo (2018) takes different approach; models claims paid and outstanding for NAIC Schedule P data for 6 LoBs and multiple companies**
- **For each accident year, development is seen as a time series => model with Recurrent Neural Network**
- **Predictions of RNN combined with company specific embedding layers to produce forecasts**
- **Compares results to models in Meyers (2015) and an AutoML model; DeepTriangle model shows impressive performance on all lines**
- **Lastly, <u>granular reserving</u> for claim type/property damaged/region/age etc difficult with normal chain-ladder approach as too much data to derive LDFs judgementally; see solution in Wüthrich (2018).**



Line of Business	Mack	ODP	CIT	LIT	ML	DT
MAPE						
Commercial Auto	0.060	0.217	0.052	0.052	0.068	**0.043**
Other Liability	0.134	0.223	0.165	0.152	0.142	**0.109**
Private Passenger Auto	0.038	0.039	0.038	0.040	0.036	**0.025**
Workers' Compensation	0.053	0.105	0.054	0.054	0.067	**0.046**
RMSPE						
Commercial Auto	0.080	0.822	0.076	0.074	0.096	**0.057**
Other Liability	0.202	0.477	0.220	0.209	0.181	**0.150**
Private Passenger Auto	0.061	0.063	0.057	0.060	0.059	**0.039**
Workers' Compensation	0.079	0.368	0.080	0.080	0.099	**0.067**

Table and Diagram from Kuo (2018)

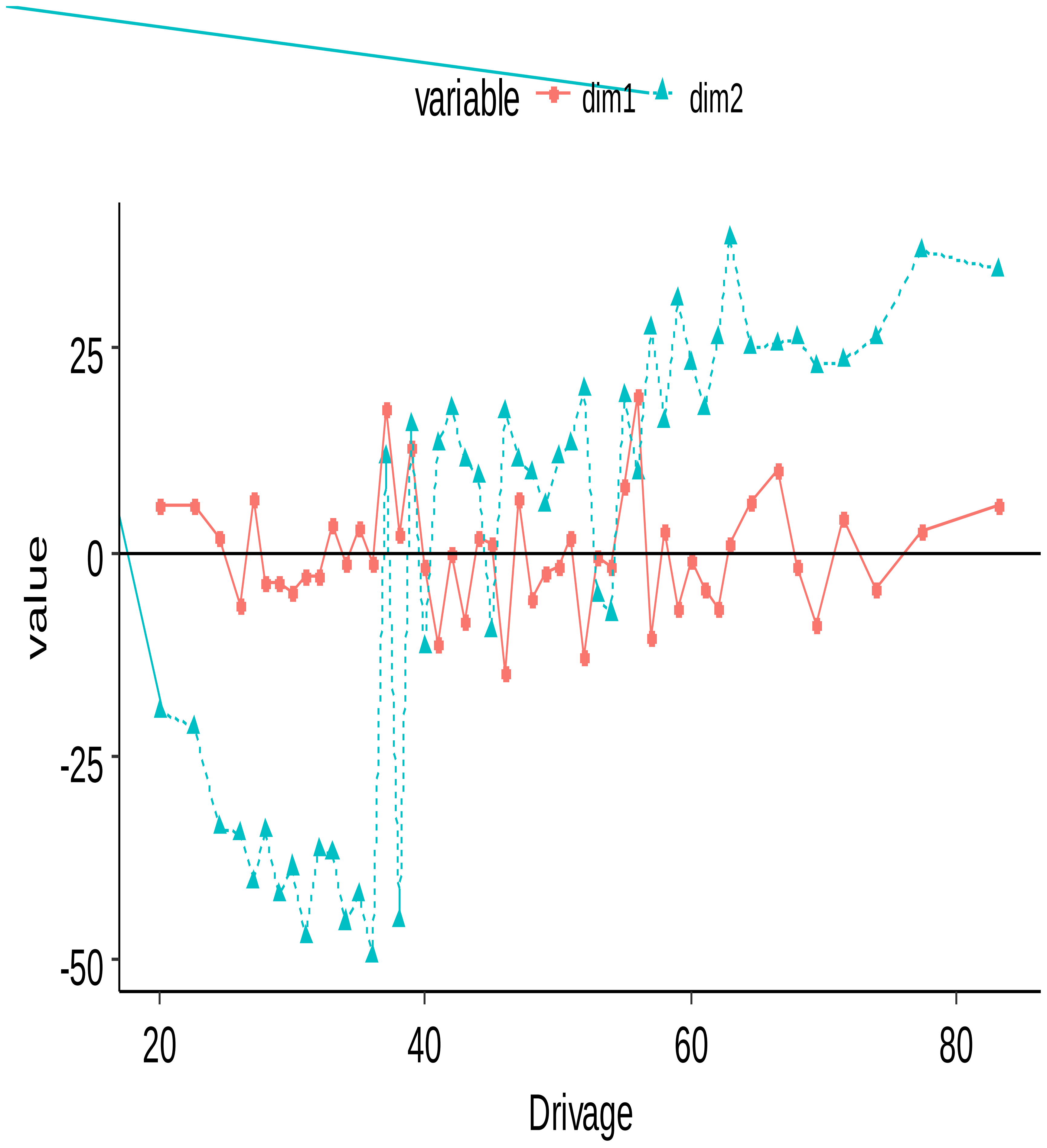
Non-life pricing (1)

- **Non-life Pricing (tabular data fit with GLMs) seems like obvious application of ML/DL**

- **Noll, Salzmann and Wüthrich (2018) is tutorial paper (with code) in which apply GLMs, regression trees, boosting and (shallow) neural networks to French TPL dataset to model frequency**
  - ML approaches outperform GLM
  - Boosted tree performs about as well as neural network…
  - ….mainly because ML approaches capture some interactions automatically
  - In own analysis, found that surprisingly, off the shelf approaches do not perform particularly well on frequency models.
  - These include XGBoost and 'vanilla' deep networks

Non-life pricing (2)

- **Deep neural network applied to raw data (i.e. no feature engineering) did not perform well**
- **Embedding layers provide significant gain in performance over GLM and other NN architectures**
    - Beats performance of best non-deep model in Noll, Salzmann and Wüthrich (2018) (OOS Loss = 0.3141 using boosting)
- **Layers learn a (multi-dimensional) schedule of relativities at each age (shown after applying t-SNE)**
- **Transfer learning – use the embeddings learned on one partition of the data, for another unseen partition of data**
    - Boosts performance of GLM



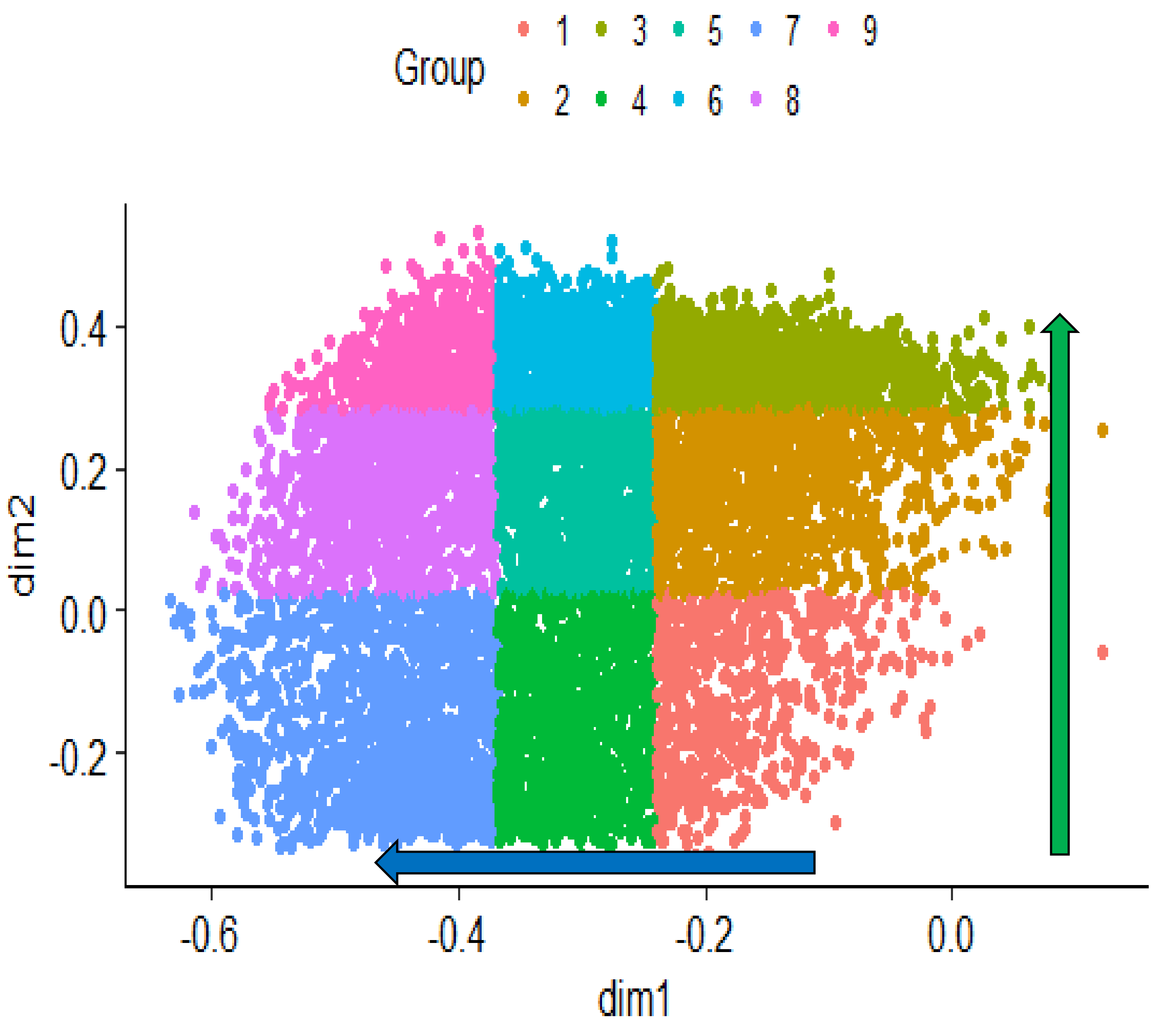
Model	OutOfSample
GLM	0.3217
GLM_Keras	0.3217
NN_shallow	0.3150
NN_no_FE	0.3258
NN_embed	0.3068
GLM_embed	0.3194
NN_learned_embed	0.2925

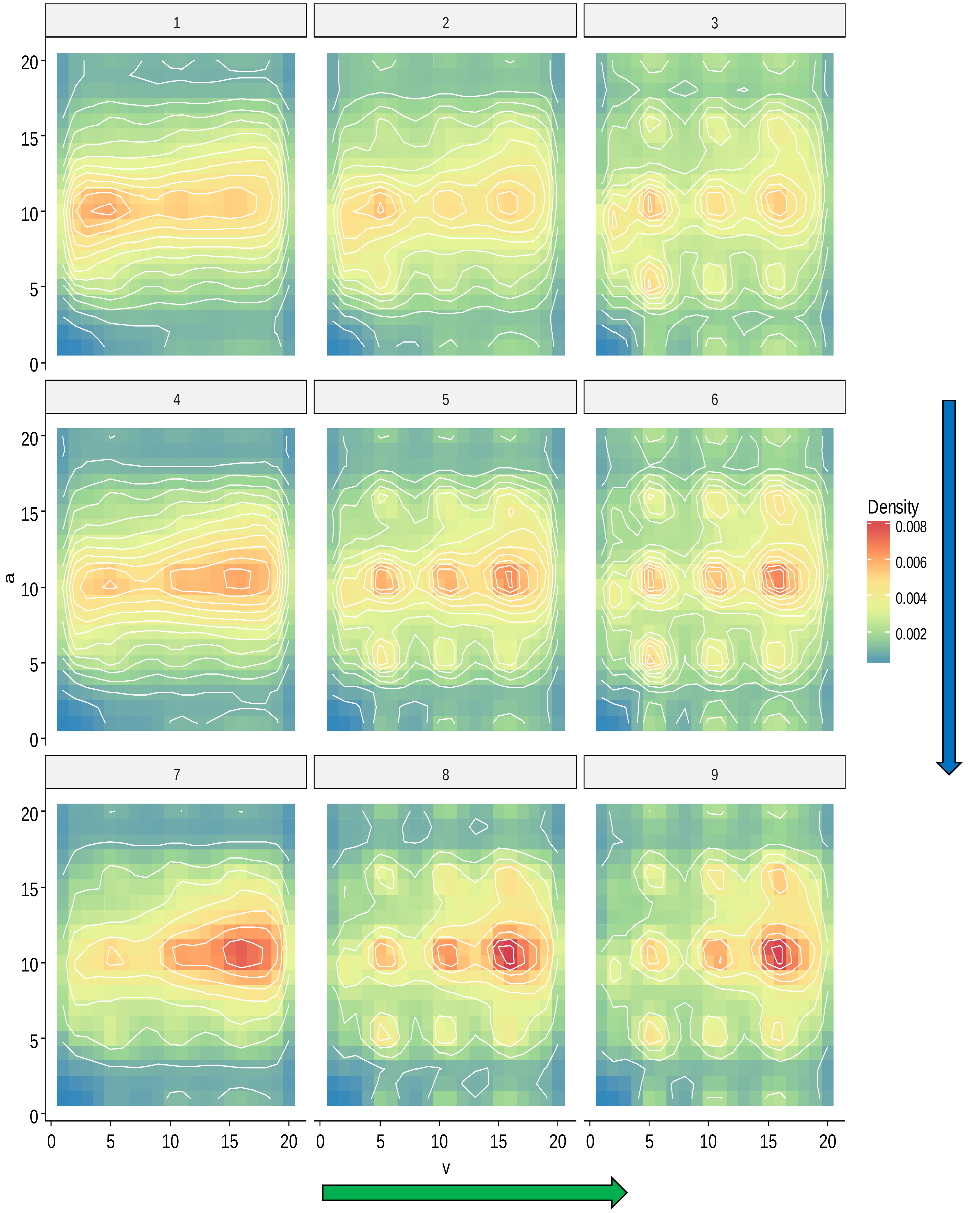
Telematics data (1)

- **Telematics produces high dimensional data (position, velocity, acceleration, road type, time of day) at high frequencies – new type of data for actuarial science!**
  - To develop "standard" models/approaches for incorporating into actuarial work might take many years => rely on deep learning
- **Mot immediately obvious how to incorporate into pricing - most approaches look to summarize telematics data streams before analysis with deep learning**
- **From outside actuarial literature, feature matrices containing summary statistics of trips analysed using RNNs plus embedding layers such as Dong, Li, Yao et al. (2016), Dong, Yuan, Yang et al. (2017) and Wijnands, Thompson, Aschwanden et al. (2018)**
- **For pricing (within actuarial literature) series of papers by Wüthrich (2017), Gao and Wüthrich (2017) and Gao, Meng and Wüthrich (2018) discuss analysis of velocity and acceleration information from telematics data feed**
- **Focus on v-a density heatmaps which capture velocity and acceleration profile of driver but these are also high dimensional**
- **Wüthrich (2017) and Gao and Wüthrich (2017) apply unsupervised learning methods (K-means, PCA and shallow auto-encoders) to summarize v-a heat-maps - Stunning result = continuous features are highly predictive**
  - Unsupervised learning applied to high dimensional data produces useful features for supervised learning

Telematics data (2)

- **Analysis using deep convolutional autoencoder with 2 dimensions.**
- **Within these dimensions (left hand plot):**
  - Right to left = amount of density in high speed bucket
  - Lower to higher = “discreteness” of the density
- **Another application is to identify drivers for UBI at correct rate (and use resulting features for pricing). See Gao and Wüthrich (2019) who apply CNNs to identify drivers based on velocity/acceleration/angle**
  - 75% accuracy on 180s of data





Lite Valuation Models (1)

- **Major challenge in valuation of Life business with embedded options/guarantees or with-profits is run time of (nested) stochastic models**
- **In general, for Variable Annuity business, guarantees are priced and hedged using Monte Carlo simulations**
- **Under Solvency II, Life business with nested options/guarantees must be valued using nested Monte Carlo to derive the Solvency Capital Requirements (SCR)**
  - Outer loop - MC simulations to derive risk factors at t+1 under the real world measure
  - Inner loops - MC simulations to derive valuation given risk factors at t+1 under risk neutral measure
- **Running full MC valuation is time consuming; common solutions are:**
  - high performance computing
  - replicating portfolios
  - Least Squares Monte Carlo (LSMC), where regression fit to results of inner loop conditional on outer loop
  - "lite" valuation models, see work by Gan and Lin (2015)

Lite Valuation Models (2)

- **Recent work using neural networks to enhance this process**
- **Hejazi and Jackson (2016, 2017) provide novel approach based on matching prototype contracts**
- **For VA valuation and hedging, Doyle and Groendyke (2018) build a lite valuation model using a shallow neural network that takes key market and contract data and outputs contract value and hedging parameters.**

  Achieve highly accurate results versus full MC approach.

- **For modelling with-profits contracts in SII, Nigri, Levantesi, Marino et al. (2019) replace inner loop basis function regression of LSMC with SVM and a deep neural network, and compare results with full nested MC.**

  Find that DL beats the basis function regression and SVM, producing highly accurate evaluations of the SCR.

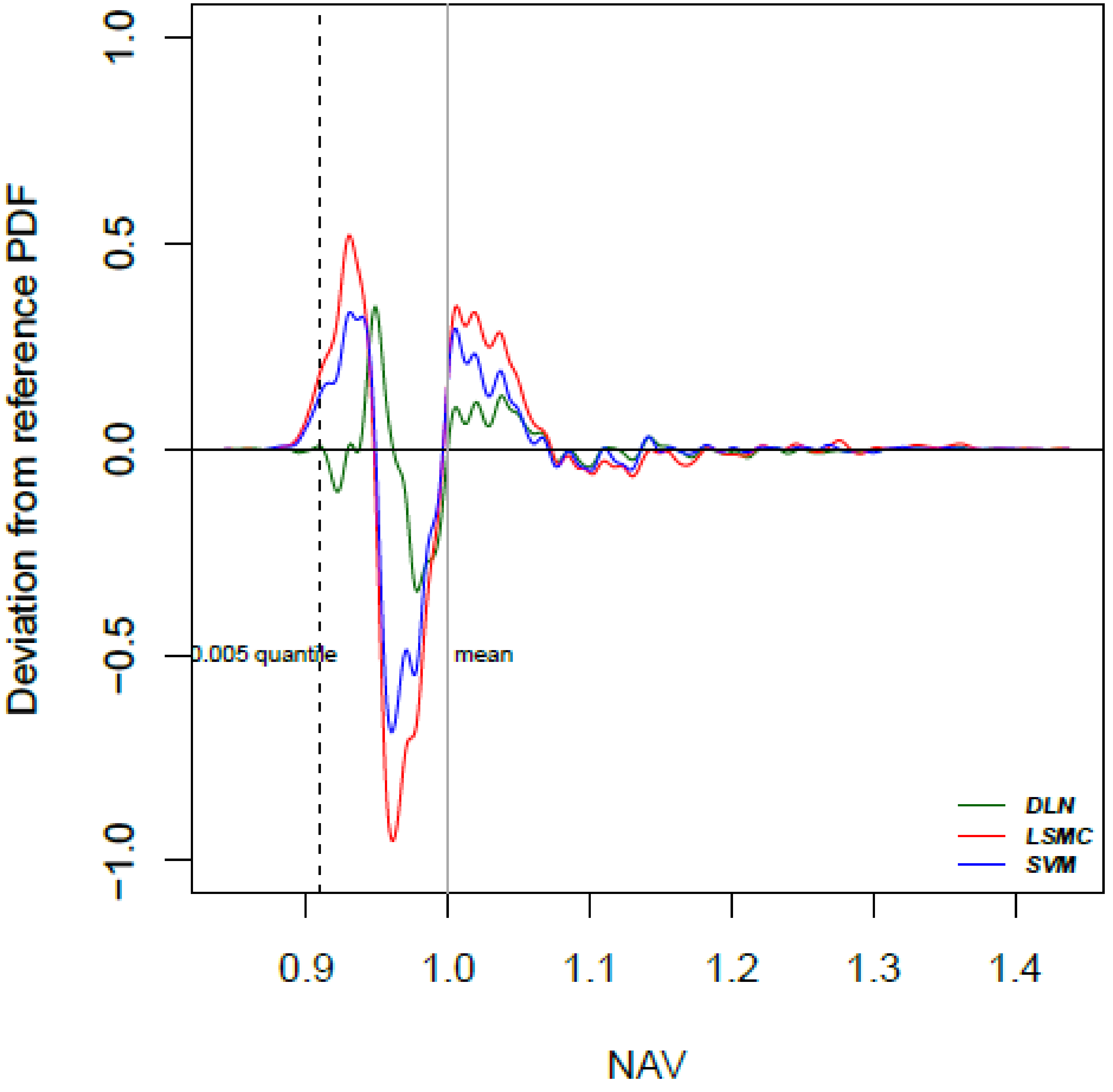


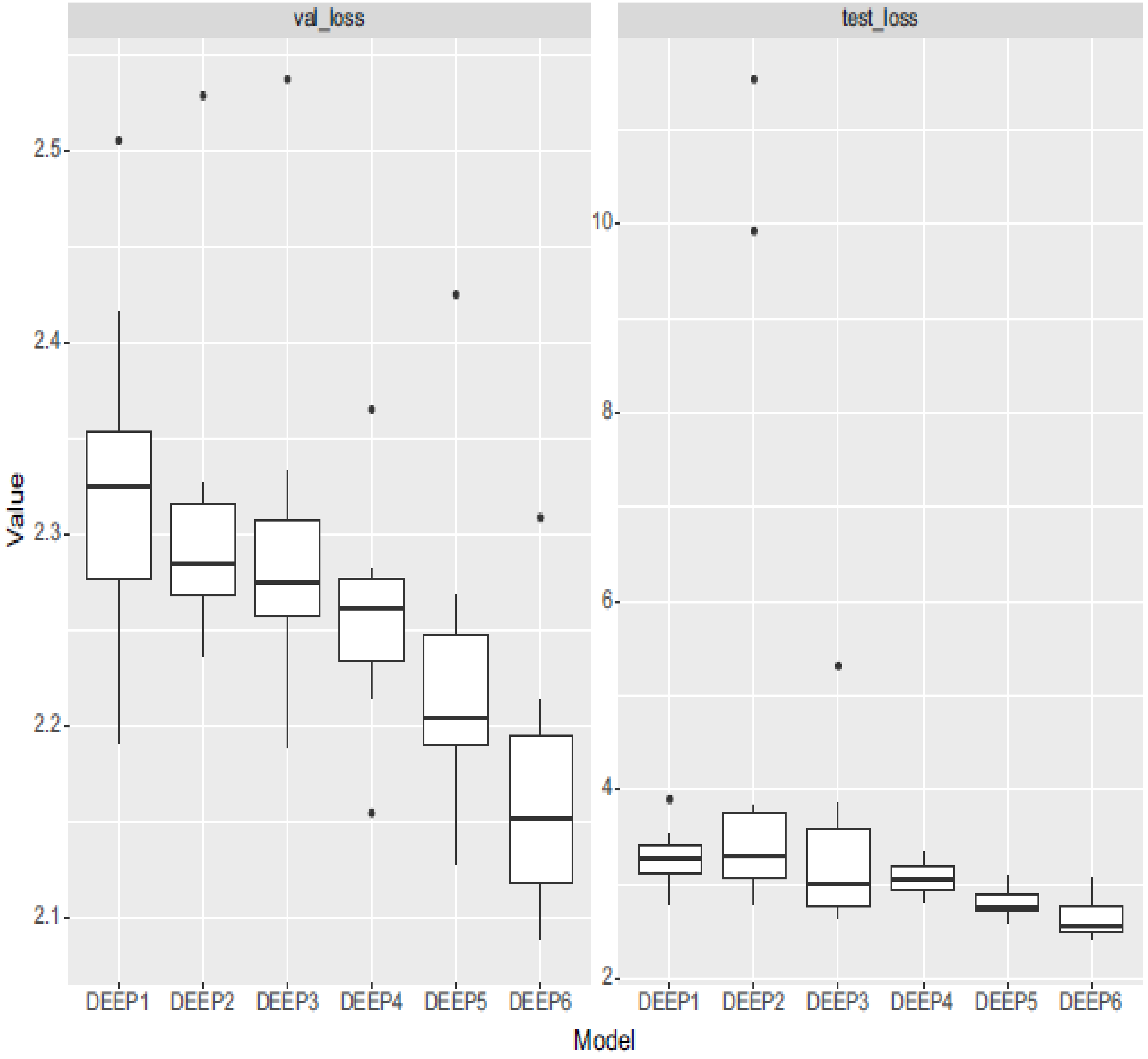
Diagram from Nigri, Levantesi, Marino et al. (2019

Agenda

- **Introduction**
- **Machine Learning**
- **Deep Learning**
- **Tools of the Trade**
- **Selected Applications**
- **Challenges**
- **Conclusion**

Stability of results

- **The training of neural networks contains some randomness due to:**
  - Random initialization of parameters
  - Dropout
  - Shuffling of data
- **Leads to validation and test set results that can exhibit variability. Not a "new" problem; see Guo and Berkhahn (2016).**
- **Problem worse on small datasets (where other ML techniques are stable) and autoencoders**
- **Example – validation and test set results of 6 DL models run 10 times on LC NN model applied to full HMD dataset.**
- **Solutions - Average models over several runs or at several points in the training (see Gabrielli (2019))**
- **Results of network might not match portfolio average due to early stopping. See Wüthrich (2019) for analysis and solutions,**



Interpretability

- **A common complaint is that neural networks are "black boxes" i.e. in some way, it is not possible to understand how the network has derived its results from the input.**

- **Taken to an extreme, some views are that neural networks might not be suitable for the insurance industry.**

- **We should differentiate between explaining a phenomenon versus interpreting a model**
  - Explaining = causal understanding built via modelling; not necessarily achievable using models built for prediction (since model parameters are biased)
  - Interpretability = understanding why a model makes a prediction.

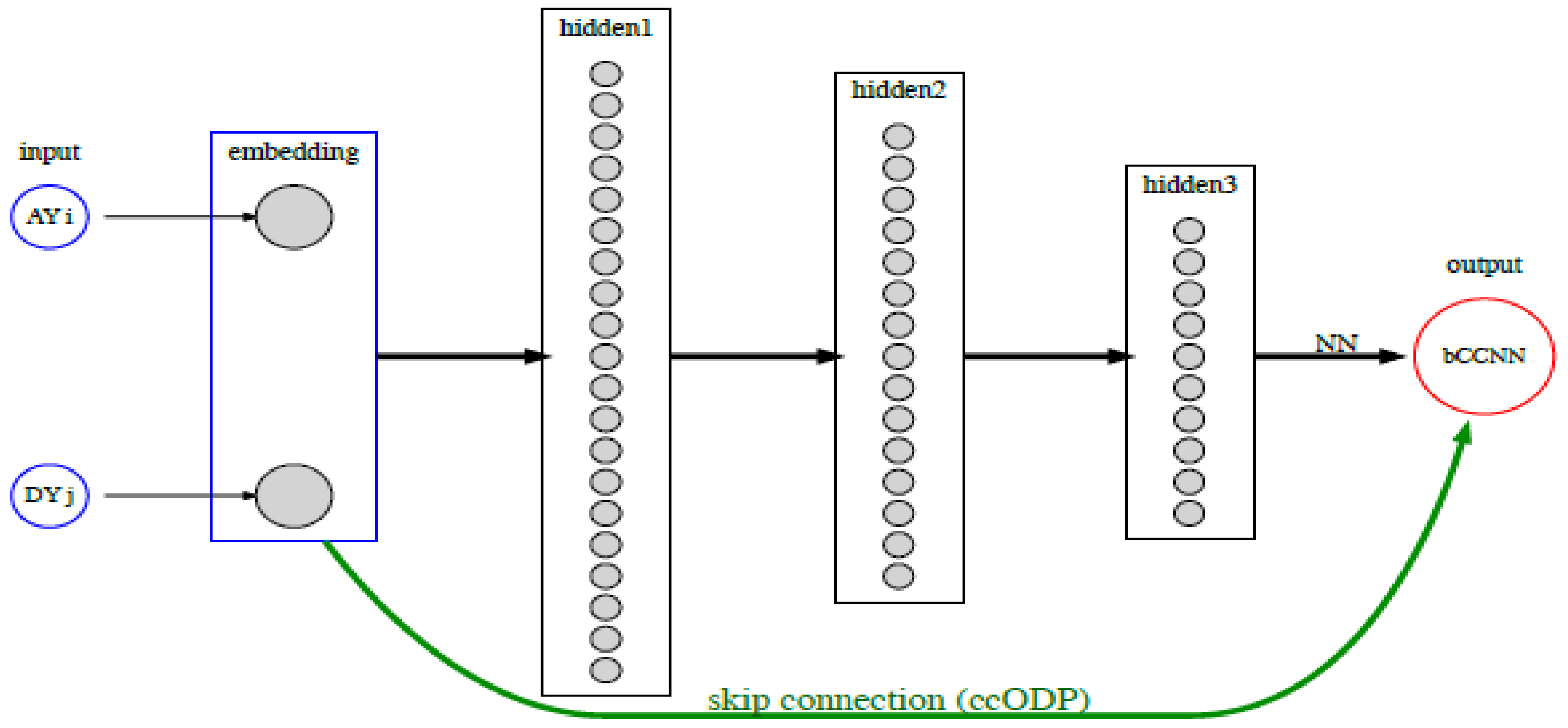
- **General purpose machine learning interpretability techniques such as LIME (Ribeiro, Singh and Guestrin 2016) and ANCHOR (Ribeiro, Singh and Guestrin 2018) allow for the interpretation of neural networks**

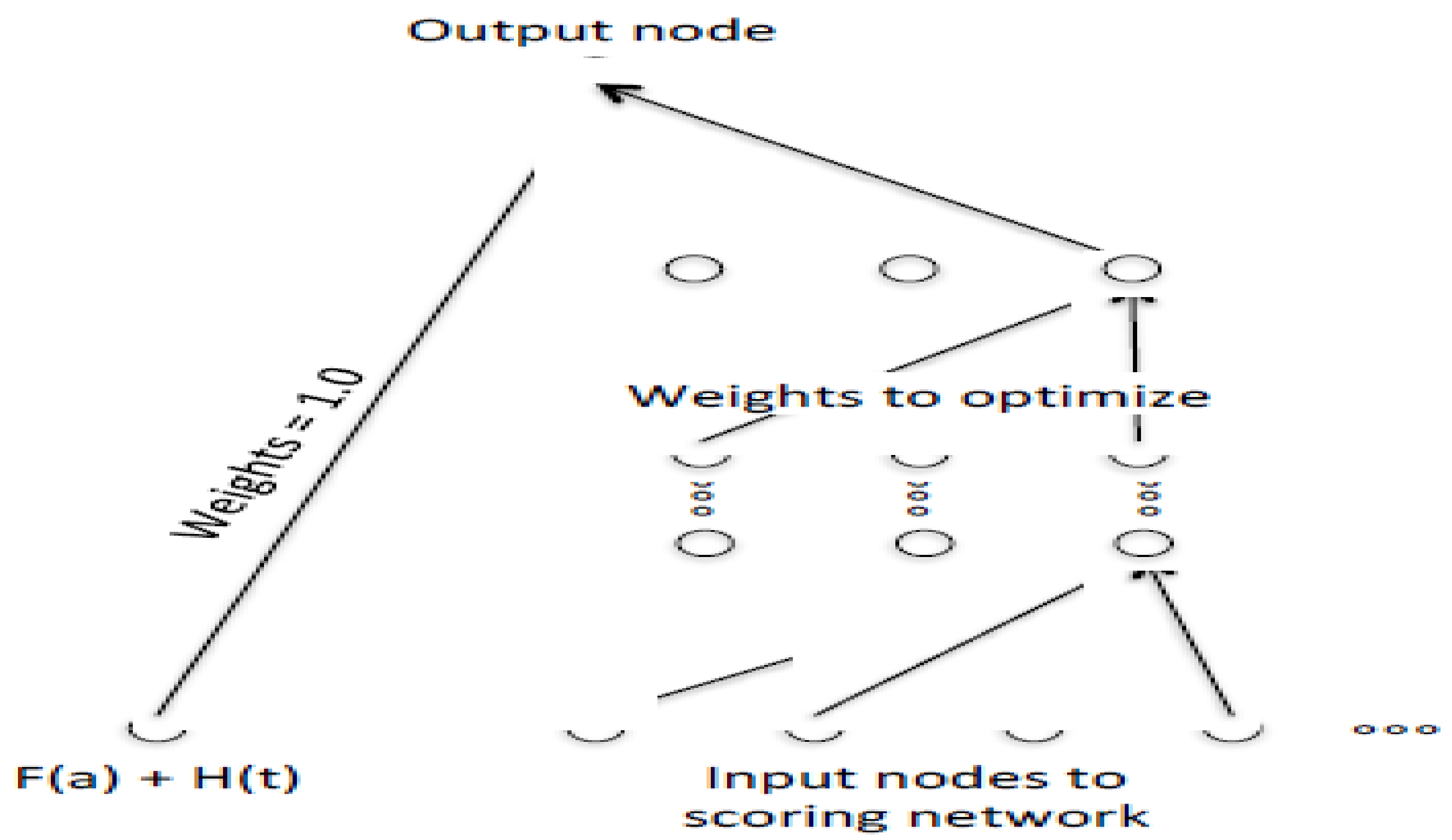
- **To what extent are neural networks black boxes?**
  - Can inspect learned representations at each stage of the model, leading to an understanding of what representation/model has been specified
  - Many visualization techniques developed, especially for convolutional neural networks

- **Can neural networks be designed for interpretability?**

Combined Actuarial Neural Net (CANN)

- **Combine a traditional actuarial model together with a neural net (Wüthrich and Merz 2018). Implemented so far for pricing (Schelldorfer and Wüthrich 2019) and reserving (Gabrielli 2019; Gabrielli, Richman and Wuthrich 2018)**
  - Traditional model (calibrated with MLE) directly connected with output of network using skip connection
  - Model output then enhanced by model structure learned by neural net to explain residuals
  - Easy to interpret (and fast to calibrate)
- **Can use the CANN model to highlight major differences from predictions of traditional model i.e. isolate the network output.**
  - Can be used as model diagnostic (Schelldorfer and Wüthrich 2019)
  - Shifts the interpretability problem
- **See Breeden and Leonova (2019) who use a similar proposal to incorporate prior economic information into a credit model**
  - Age and Economic effects via skip connection; Cohort effects via neural networks





Bottom diagram from Breeden and Leonova (2019)

Uncertainty bounds

- **Ability to quantify extent of uncertainty in predictions is key to many actuarial tasks; however, focus of deep learning literature is on best estimate**

- **Several approaches proposed:**
  - Use of dropout as an approximation of model uncertainty (Gal 2016; Kendall and Gal 2017)
  - Quantile regression to derive prediction bounds (Smyl 2018)
  - Use neural networks for GAMLSS regression

- **Not immediately obvious how to reconcile to traditional actuarial framework (often relies on bootstrapping)**
  - Seemingly, framework of Kendall and Gal (2017) for computer vision correlates with traditional actuarial understanding (model and parameter risk = epistemic uncertainty; process risk = aleatoric uncertainty)

- **Gabrielli, Richman and Wüthrich (2018) apply bootstrap to the multi-LoB ODP NN model – found that decreased bias but increased RMSEP versus separate ODP models**

- **More research needed**

Agenda

- **Introduction**
- **Machine Learning**
- **Deep Learning**
- **Tools of the Trade**
- **Selected Applications**
- **Challenges**
- **<u>Conclusion</u>**

Conclusion

- **Deep learning can enhance the predictive power of models built by actuaries**
- **Emphasis on predictive performance and potential gains of moving from traditional actuarial and statistical methods to machine and deep learning approaches.**
- **Measurement framework utilized within machine learning – focus on testing predictive performance => focus on measurable improvements in predictive performance led to refinements and enhancements of deep learning architectures**
- **Learned representations from deep neural networks often have readily interpretable meaning**
- **Very useful for high-frequency and high-dimensional data**
- **Application of deep learning techniques to actuarial problems is rapidly emerging field within actuarial science => appears reasonable to predict more advances in the near-term.**
- **Deep learning is not a panacea for all modelling issues - applied to the wrong domain, deep learning will not produce better or more useful results than other techniques.**

Acknowledgements

- Mario Wüthrich
- Nicolai von Rummell
- Data Science working group of the SAA

Appendix - Other Techniques

- **Dropout (Srivastava, Hinton, Krizhevsky et al. 2014)**
  used to regularize NNs, can be combined with L1 or L2 regularizers
- **Batchnorm (Ioffe and Szegedy 2015)**
  technique used to make NNs easier to optimize and also provides a regularization effect
- **Attention (Bahdanau, Cho and Bengio 2014)**
  allows networks to choose most relevant parts of a representation
- **Generative Adversarial Models (GANs) (Goodfellow, Pouget-Abadie, Mirza et al. 2014)**
  Game between two NNs, whereby a generator network produces output that tries to trick a discriminator network.
  Useful for generative modelling, but other interesting applications such as BiGAN (Donahue, Krähenbühl and Darrell 2016)
  Variational autoencoders (VAEs) (Kingma and Welling 2013)
  Autoencoder with distributional assumptions made on codes Neural
- **Network Architecture Search (NNAS)**
  Techniques used to design NNs automatically
- **Pruning**
  New technique that takes a trained NN and tries to reduce redundancy (number of layers/parameters) while maintaining performance
  Part of Tensorflow 2 API

References

- **See** https://gist.github.com/RonRichman/655cca0dd79afcd20b33d3131c537414