Yes, we CANN!

Mario V. Wüthrich
RiskLab, ETH Zurich

Florence, May 21-24, 2019

Insurance data science related activities

- **Yes, we CANN!** Editorial of ASTIN Bulletin 49/1, 2019

- **Actuarial Data Science** Initiative of the Swiss Association of Actuaries SAV
  - ⋆ Case study: French motor third-party liability claims, SSRN 2018
  - ⋆ Insights from inside neural networks, SSRN 2018
  - ⋆ Nesting classical actuarial models into neural networks, SSRN 2019

  `www.actuarialdatascience.org`

- **Insurance Data Science Conference**, June 14, 2019, ETH Zurich
  `www.insurancedatascience.org`

Yes, we CANN![1]

[1]In the introduction to my presentation at the Waterloo conference, Prof. Sheldon Lin (University of Toronto) suggested to rename this title to "Let's make Actuarial Science great again".

The modeling cycle

(1) data collection, data cleaning and data pre-processing (80% of the total time)

(2) selection of model class (model vs. algorithmic culture, Breiman (2001))

(3) choice of objective function

(4) 'solving' a (non-convex) optimization problem

(5) model validation (or prediction accuracy)

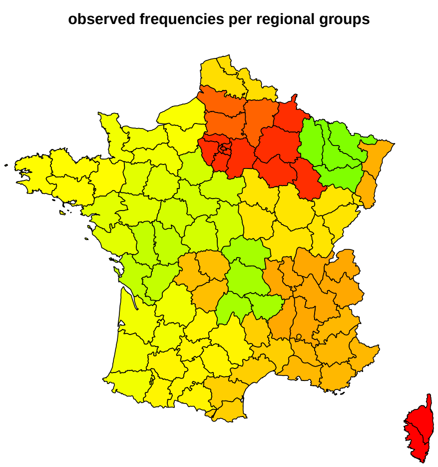
(6) possibly go back to (1)

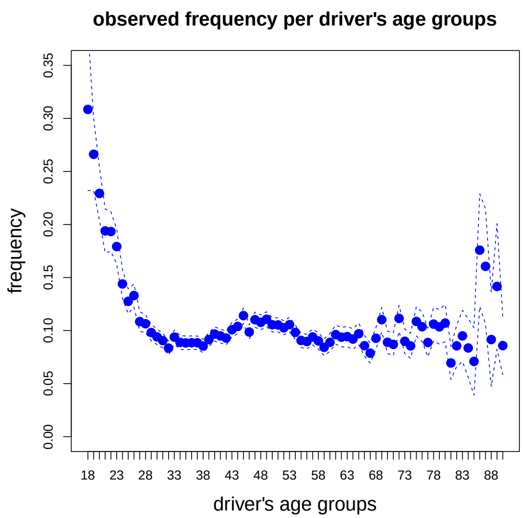
▷ 'solving' involves:

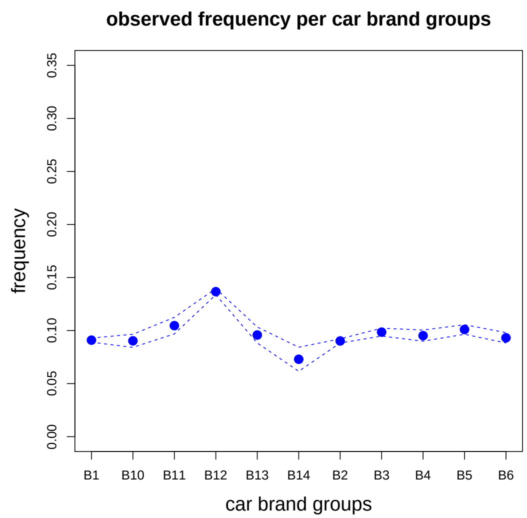
⋆ choice of algorithm
⋆ choice of step size, stopping criterion
⋆ choice of seed (starting value)

Data collection: car insurance frequency example

```
> str(freMTPL2freq)        #source R package CASdatasets
'data.frame':    678013 obs. of  12 variables:
 $ IDpol      : num  1 3 5 10 11 13 15 17 18 21 ...
 $ ClaimNb    : num 1 1 1 1 1 1 1 1 1 1 ...
 $ Exposure   : num  0.1 0.77 0.75 0.09 0.84 0.52 0.45 0.27 0.71 0.15 ...
 $ Area       : Factor w/ 6 levels "A","B","C","D",..: 4 4 2 2 2 5 5 3 3 2 ...
 $ VehPower   : int  5 5 6 7 7 6 6 7 7 7 ...
 $ VehAge     : int  0 0 2 0 0 2 2 0 0 0 ...
 $ DrivAge    : int  55 55 52 46 46 38 38 33 33 41 ...
 $ BonusMalus: int  50 50 50 50 50 50 50 68 68 50 ...
 $ VehBrand   : Factor w/ 11 levels "B1","B10","B11",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ VehGas     : Factor w/ 2 levels "Diesel","Regular": 2 2 1 1 1 2 2 1 1 1 ...
 $ Density    : int  1217 1217 54 76 76 3003 3003 137 137 60 ...
 $ Region     : Factor w/ 22 levels "R11","R21","R22",..: 18 18 3 15 15 8 8 20 20 12 ...
```







Generalized linear models (GLMs)

- Determine from data $\mathcal{D} = \{(Y_1, \boldsymbol{x}_1), \ldots, (Y_n, \boldsymbol{x}_n)\}$ an unknown regression function

$$\boldsymbol{x} \;\mapsto\; \mu(\boldsymbol{x}) = \mathbb{E}[Y].$$

- Selection of model class: Poisson GLM with canonical (log-)link:

$$\boldsymbol{x} \;\mapsto\; \mu_{\boldsymbol{\beta}}^{\mathrm{GLM}}(\boldsymbol{x}) \;=\; \exp\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle \;=\; \exp\left\{ \sum_j \beta_j x_j \right\}.$$

- Estimate regression parameter $\boldsymbol{\beta}$ with maximum likelihood $\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}$ by minimizing the corresponding deviance loss (objective function)

$$\boldsymbol{\beta} \;\mapsto\; \mathcal{L}_{\mathcal{D}}(\boldsymbol{\beta}).$$

Example: car insurance Poisson frequencies

```
> str(freMTPL2freq)        #source R package CASdatasets
'data.frame':    678013 obs. of  12 variables:
 $ IDpol      : num  1 3 5 10 11 13 15 17 18 21 ...
 $ ClaimNb    : num 1 1 1 1 1 1 1 1 1 1 ...
 $ Exposure   : num  0.1 0.77 0.75 0.09 0.84 0.52 0.45 0.27 0.71 0.15 ...
 $ Area       : Factor w/ 6 levels "A","B","C","D",..: 4 4 2 2 2 5 5 3 3 2 ...
 $ VehPower   : int  5 5 6 7 7 6 6 7 7 7 ...
 $ VehAge     : int  0 0 2 0 0 2 2 0 0 0 ...
 $ DrivAge    : int  55 55 52 46 46 38 38 33 33 41 ...
 $ BonusMalus: int  50 50 50 50 50 50 50 68 68 50 ...
 $ VehBrand   : Factor w/ 11 levels "B1","B10","B11",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ VehGas     : Factor w/ 2 levels "Diesel","Regular": 2 2 1 1 1 2 2 1 1 1 ...
 $ Density    : int  1217 1217 54 76 76 3003 3003 137 137 60 ...
 $ Region     : Factor w/ 22 levels "R11","R21","R22",..: 18 18 3 15 15 8 8 20 20 12 ...
```

	# param.	in-sample loss (in 10^{-2})	out-of-sample loss (in 10^{-2})
homogeneous ($\mu \equiv \text{const.}$)	1	32.935	33.861
Model GLM (Poisson)	48	31.257	32.149

Note for low frequency examples of, say, 5%: we have in the true model $\mathcal{L}_{\mathcal{D}} \approx 30.3 \cdot 10^{-2}$.

From GLMs to neural networks

- Example of a GLM (with log-link $\Rightarrow$ exponential output activation):

$$\boldsymbol{x} \;\mapsto\; \mu_{\boldsymbol{\beta}}^{\mathrm{GLM}}(\boldsymbol{x}) \;=\; \exp\langle \boldsymbol{\beta}, \boldsymbol{x}\rangle.$$

- Choose network of depth $d \in \mathbb{N}$ with network parameter $\theta = (\theta_{1:d}, \theta_{d+1})$:

$$\boldsymbol{x} \;\mapsto\; \mu_{\theta}^{\mathrm{NN}}(\boldsymbol{x}) \;=\; \exp\langle \theta_{d+1}, \boldsymbol{z}\rangle,$$

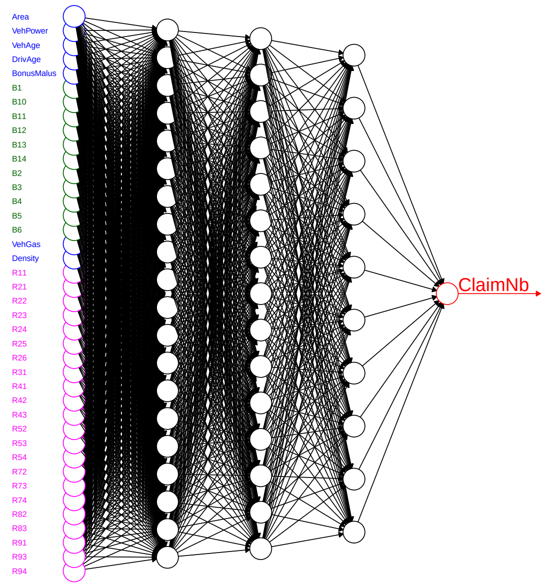
with neural network function (covariate pre-processing $\boldsymbol{x} \mapsto \boldsymbol{z}$)

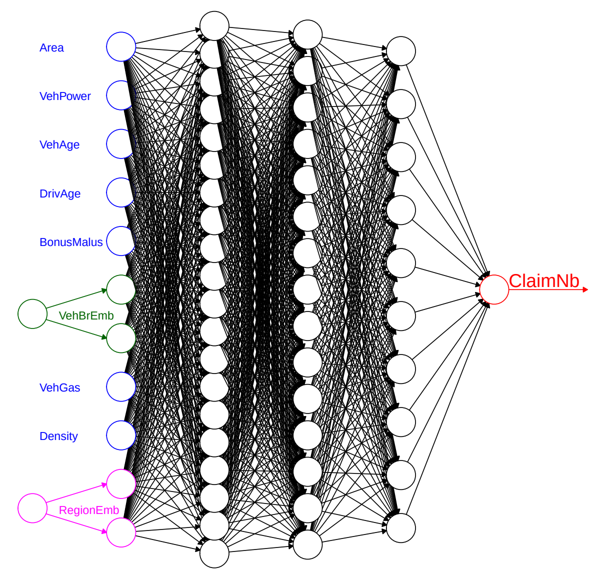
$$\boldsymbol{x} \;\mapsto\; \boldsymbol{z} \;=\; \boldsymbol{z}_{\theta_{1:d}}(\boldsymbol{x}) \;=\; \left(\boldsymbol{z}^{(d)} \circ \cdots \circ \boldsymbol{z}^{(1)}\right)(\boldsymbol{x}).$$

Neural network with embeddings

- Network of depth $d \in \mathbb{N}$ with network parameter θ

$$\boldsymbol{x} \mapsto \mu_\theta^{\rm NN}(\boldsymbol{x}) = \exp\langle \theta_{d+1}, \boldsymbol{z} \rangle = \exp\left\langle \theta_{d+1}, \left(\boldsymbol{z}^{(d)} \circ \cdots \circ \boldsymbol{z}^{(1)}\right)(\boldsymbol{x}) \right\rangle.$$



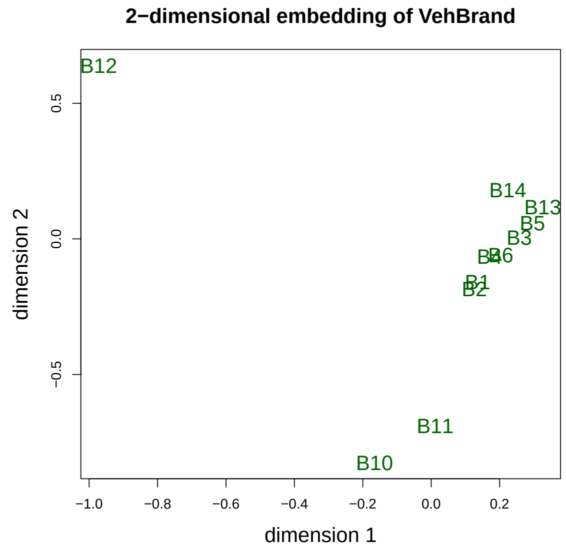


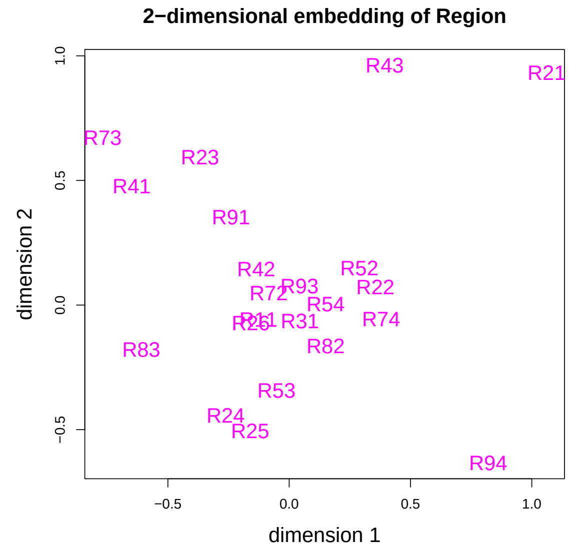
- Gradient descent method (GDM) provides $\widehat{\theta}$ w.r.t. deviance loss $\theta \mapsto \mathcal{L}_{\mathcal{D}}(\theta)$.

- Exercise early stopping of GDM because MLE over-fits (in-sample).

NN example: car insurance frequencies

	# param.	in-sample loss (in 10^{-2})	out-of-sample loss (in 10^{-2})
homogeneous ($\mu \equiv \mathrm{const.}$)	1	32.935	33.861
Model GLM (Poisson)	48	31.257	32.149
NN (2-dim. embeddings)	792	30.165	31.453

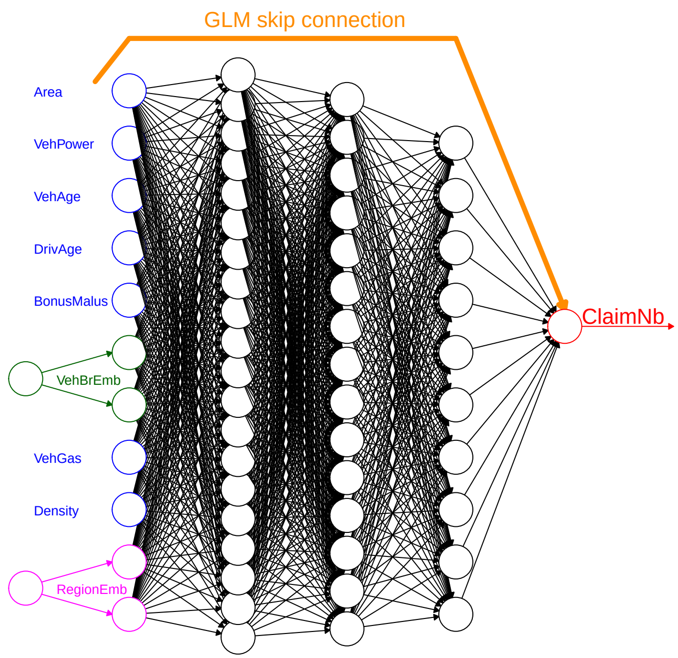




Combined Actuarial Neural Network: part I

- Choose regression function with parameter $(\boldsymbol{\beta}, \theta)$

$$\boldsymbol{x} \;\mapsto\; \mu^{\text{CANN}}_{(\boldsymbol{\beta},\theta)}(\boldsymbol{x}) = \exp\left\{\langle \boldsymbol{\beta}, \boldsymbol{x}\rangle + \left\langle \theta_{d+1}, \boldsymbol{z}^{(d)} \circ \cdots \circ \boldsymbol{z}^{(1)}(\boldsymbol{x}) \right\rangle\right\}.$$

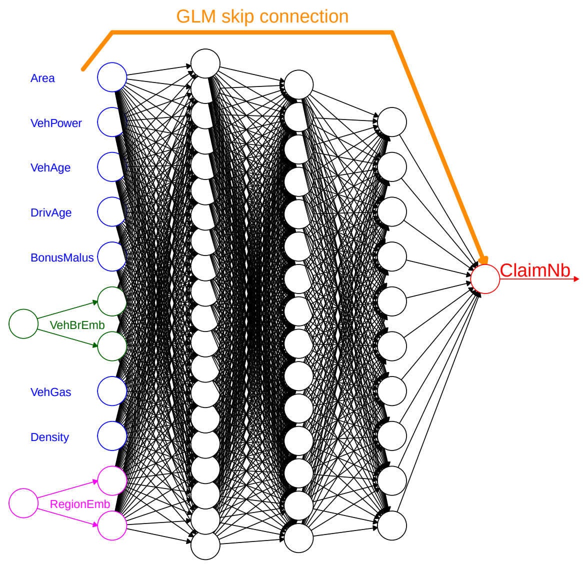


- GDM provides $(\widehat{\boldsymbol{\beta}}, \widehat{\theta})$ w.r.t. deviance loss $(\boldsymbol{\beta}, \theta) \mapsto \mathcal{L}_{\mathcal{D}}(\boldsymbol{\beta}, \theta)$.

Combined Actuarial Neural Network: part II

- Choose regression function with parameter $(\boldsymbol{\beta}, \theta)$

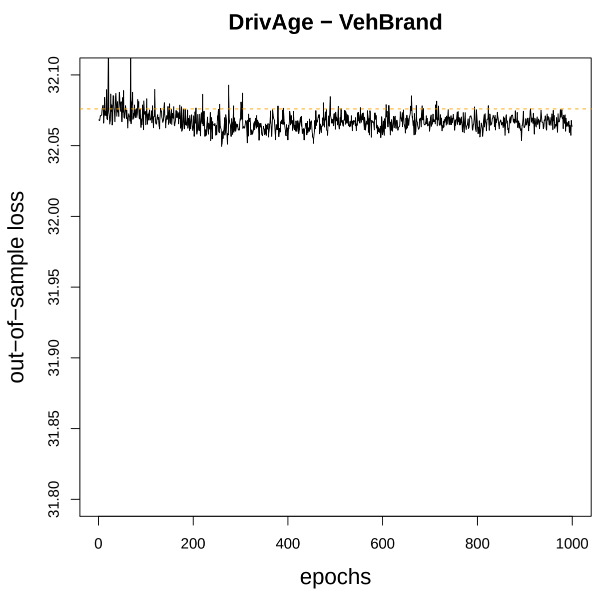
$$\mu^{\text{CANN}}_{(\boldsymbol{\beta},\theta)}(\boldsymbol{x}) = \exp\left\{\langle \boldsymbol{\beta}, \boldsymbol{x}\rangle + \left\langle \theta_{d+1}, \boldsymbol{z}^{(d)} \circ \cdots \circ \boldsymbol{z}^{(1)}(\boldsymbol{x})\right\rangle\right\}.$$

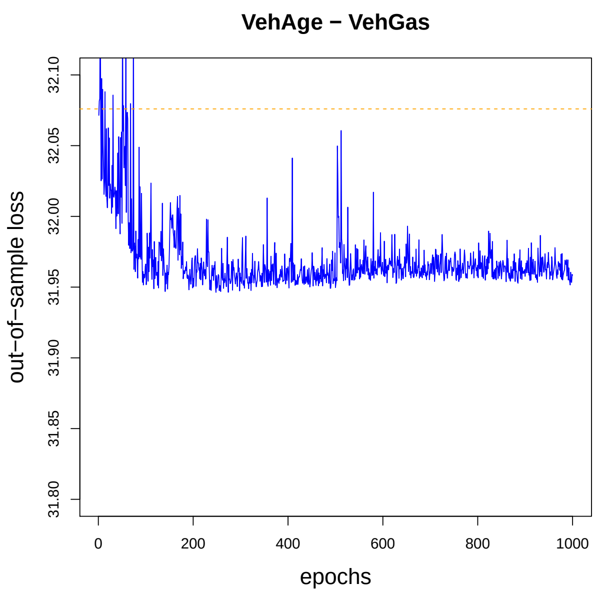


- GDM provides $(\widehat{\boldsymbol{\beta}}, \widehat{\theta})$ w.r.t. deviance loss $(\boldsymbol{\beta}, \theta) \mapsto \mathcal{L}_{\mathcal{D}}(\boldsymbol{\beta}, \theta)$.

- Initialize gradient descent algorithm with $\widehat{\boldsymbol{\beta}}^{\text{MLE}}$ and $\theta_{d+1} = 0$!

Combined Actuarial Neural Network





Possible GDM results of the CANN approach.

CANN example: car insurance frequencies

	# param.	in-sample loss (in 10^{-2})	out-of-sample loss (in 10^{-2})
homogeneous ($\mu \equiv \text{const.}$)	1	32.935	33.861
Model GLM (Poisson)	48	31.257	32.149
CANN (2-dim. embeddings)	792 (+48)	30.476	31.566
NN (2-dim. embeddings)	792	30.165	31.453

Variants of CANN

- Freeze $\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}$ (use as offset) and only train network parameter θ

$$\mu^{\mathrm{CANN}}_{(\boldsymbol{\beta},\theta)}(\boldsymbol{x}) = \exp\left\{\langle\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}},\boldsymbol{x}\rangle + \left\langle \theta_{d+1}, \boldsymbol{z}^{(d)}\circ\cdots\circ\boldsymbol{z}^{(1)}(\boldsymbol{x})\right\rangle\right\}.$$

- Introduce trainable credibility weight α for the offset

$$\mu^{\mathrm{CANN}}_{(\boldsymbol{\beta},\theta)}(\boldsymbol{x}) = \exp\left\{\alpha\langle\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}},\boldsymbol{x}\rangle + (1-\alpha)\left\langle \theta_{d+1}, \boldsymbol{z}^{(d)}\circ\cdots\circ\boldsymbol{z}^{(1)}(\boldsymbol{x})\right\rangle\right\}.$$

- Find missing interactions in (x_l, x_k) in addition to the offset

$$\mu^{\mathrm{CANN}}_{(\boldsymbol{\beta},\theta)}(\boldsymbol{x}) = \exp\left\{\langle\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}},\boldsymbol{x}\rangle + \left\langle \theta_{d+1}, \boldsymbol{z}^{(d)}\circ\cdots\circ\boldsymbol{z}^{(1)}(x_l, x_k)\right\rangle\right\}.$$

- Learn across different portfolios.

Issue in almost all neural network models

- Neural network calibrations do not have the balance property (unbiasedness on portfolio level).

- GLM calibrations do have the balance property (critical point of the deviance loss for full rank design matrix under the canonical link function).

- Neural network calibration needs regularization for correction of this deficiency.

- Neural networks provide multiple equally good models (non-uniqueness of a best model). What are the properties of typical good models?

Summary and outlook

- CANN allows us to identify missing structure in GLMs (more) explicitly.
- CANN allows us to learn across different portfolios.
- CANN needs care in terms of the balance property.

Thanks to my co-authors:

Christoph Buser (smile.direct versicherungen)
Andrea Ferrario (Mobiliar Lab for Analytics)
Andrea Gabrielli (RiskLab, ETH Zurich)
Michael Merz (University of Hamburg)
Alexander Noll (PartnerRe)
Ronald Richman (AIG)
Robert Salzmann (Signal Iduna)
Jürg Schelldorfer (Swiss Re)