
# LocalGLMnet: An Interpretable Deep Learning Architecture

EAA e-Conference on
Data Science & Data Ethics

12 May 2022

*Mario V. Wüthrich*

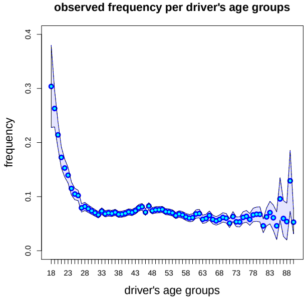
*RiskLab, ETH Zürich*

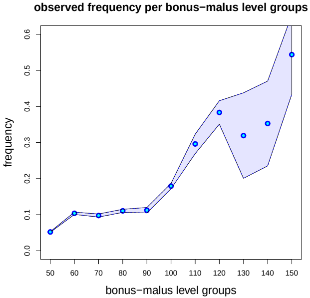
# Outline of presentation

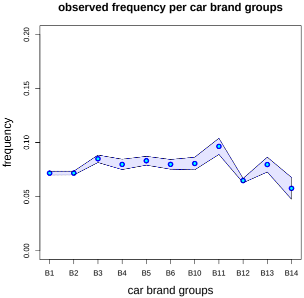
- Regression problem
- Generalized linear models (GLMs)
- Neural network regression models
- LocalGLMnet architecture
- Example
- Outlook: regularization

# Regression modeling: car insurance example

```
'data.frame':   678007 obs. of  10 variables:
$ IDpol      : num  1 3 5 10 11 13 15 17 18 21 ...
$ 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","B2","B3",..: 9 9 9 9 9 9 9 9 9 9 ...
$ Region     : Factor w/ 22 levels "R11","R21","R22",..: 18 18 3 15 15 8 8 20 20 12 ...
$ ClaimNb    : num  0 0 0 0 0 0 0 0 0 0 ...
```



# Regression modeling: car insurance example

```
'data.frame':   678007 obs. of  10 variables:
$ IDpol      : num  1 3 5 10 11 13 15 17 18 21 ...
$ 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","B2","B3",..: 9 9 9 9 9 9 9 9 9 9 ...
$ Region     : Factor w/ 22 levels "R11","R21","R22",..: 18 18 3 15 15 8 8 20 20 12 ...
$ ClaimNb    : num  0 0 0 0 0 0 0 0 0 0 ...
```

**Goal.**

- Find a suitable regression function that describes the systematic effects as a function of the available covariates $\boldsymbol{x} \in \mathbb{R}^q$.
- This gives us pure risk premium

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

where $\boldsymbol{x}$ are the covariates (explanatory variables) describing claim $Y$.

## Generalized linear models (GLMs)

- GLM: Choose strictly monotone link function $g$ and assume

$$\boldsymbol{x} = (x_1, \ldots, x_q) \;\mapsto\; g\left(\mu^{\mathrm{GLM}}(\boldsymbol{x})\right) \;=\; \beta_0 + \sum_{j=1}^{q} \beta_j x_j,$$

  for regression parameter $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_q) \in \mathbb{R}^{q+1}$.
- Regression parameter $\boldsymbol{\beta}$ is estimated with MLE.
- Examples: Gaussian, Poisson, Gamma and Inverse Gaussian GLMs.
- GLMs are linear in covariate $\boldsymbol{x}$ after applying link $g$, i.e., explainable.
- Often a linear function does not fit the data: requires covariate engineering.
- 50 years of GLMs: Nelder–Wedderburn (1972).

# From GLMs to neural networks

- GLM: Choose strictly monotone link function $g$ and assume

$$\boldsymbol{x} \mapsto g\left(\mu^{\mathrm{GLM}}(\boldsymbol{x})\right) = \beta_0 + \sum_{j=1}^{q} \beta_j x_j.$$

- (Neural) network: Set for regression function

$$\boldsymbol{x} \mapsto g\left(\mu^{\mathrm{net}}(\boldsymbol{x})\right) = \beta_0 + \sum_{j=1}^{q_d} \beta_j z_j^{(d:1)}(\boldsymbol{x}),$$

where $\boldsymbol{x} \mapsto \boldsymbol{z}^{(d:1)}(\boldsymbol{x}) \in \mathbb{R}^{q_d}$ is a network of depth $d$.

▶ Network learns a new representation $\boldsymbol{z} = \boldsymbol{z}^{(d:1)}(\boldsymbol{x})$ of covariate $\boldsymbol{x}$.

- Network: Set for regression function

$$\boldsymbol{x} \mapsto g\left(\mu^{\text{net}}(\boldsymbol{x})\right) = \beta_0 + \sum_{j=1}^{q_d} \beta_j z_j^{(d:1)}(\boldsymbol{x}),$$

where $\boldsymbol{x} \mapsto \boldsymbol{z}^{(d:1)}(\boldsymbol{x}) \in \mathbb{R}^{q_d}$ is a network of depth $d$.

▶ Network learns a new representation $\boldsymbol{z} = \boldsymbol{z}^{(d:1)}(\boldsymbol{x})$ of covariate $\boldsymbol{x}$.

- **Pros.**
  - A well-trained network often outperforms a GLM (universal approximation).
  - Networks can process any kind of information $\boldsymbol{x}$.
- **Drawbacks.**
  - Network solution is often not interpetable and explainable.
  - No simple way of variable selection.

# LocalGLMnet: definition

- GLM: $$\boldsymbol{x} \mapsto g\left(\mu^{\mathrm{GLM}}(\boldsymbol{x})\right) = \beta_0 + \sum_{j=1}^{q} \beta_j x_j.$$

- **Idea.** Let a network learn regression attentions $\boldsymbol{\beta} = \boldsymbol{\beta}(\boldsymbol{x})$.

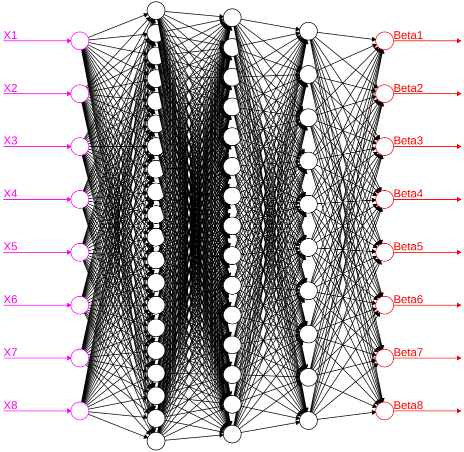
- Choose a network of depth $d$

$$\boldsymbol{z}^{(d:1)} : \mathbb{R}^q \to \mathbb{R}^q, \qquad \boldsymbol{x} \mapsto \boldsymbol{\beta}(\boldsymbol{x}) = \boldsymbol{z}^{(d:1)}(\boldsymbol{x}).$$

- LocalGLMnet: Set for regression function

$$\boldsymbol{x} \mapsto g\left(\mu(\boldsymbol{x})\right) = \beta_0 + \sum_{j=1}^{q} \beta_j(\boldsymbol{x})\, x_j.$$

# LocalGLMnet for regression attentions $\boldsymbol{\beta}(\boldsymbol{x})$



# LocalGLMnet: interpretation

- LocalGLMnet:

$$\boldsymbol{x} \mapsto g(\mu(\boldsymbol{x})) = \beta_0 + \sum_{j=1}^{q} \beta_j(\boldsymbol{x})\, x_j.$$

- If $\beta_j(\boldsymbol{x}) \equiv 0$: drop term $x_j$.
- If $\beta_j(\boldsymbol{x}) \equiv \beta_j \ (\neq 0)$: we have a GLM term in $x_j$.
- If $\beta_j(\boldsymbol{x}) = \beta_j(x_j)$: no interactions of term $x_j$ with $x_{j'}$, $j' \neq j$.
- Interactions: study gradient

$$\nabla \beta_j(\boldsymbol{x}) = \left( \frac{\partial}{\partial x_1} \beta_j(\boldsymbol{x}), \ldots, \frac{\partial}{\partial x_q} \beta_j(\boldsymbol{x}) \right) \in \mathbb{R}^q.$$

- LocalGLMnets have the universal approximation property.

# LocalGLMnet: identifiability

- LocalGLMnet:

$$\boldsymbol{x} \mapsto g(\mu(\boldsymbol{x})) = \beta_0 + \sum_{j=1}^{q} \beta_j(\boldsymbol{x})\, x_j.$$

- We do not have identifiability as we may still receive

$$\beta_j(\boldsymbol{x}) x_j = x_{j'},$$

  by learning a regression attention $\beta_j(\boldsymbol{x}) = x_{j'}/x_j$.

- We did not encounter this difficulty in gradient descent fitting, because the regression function seems rather pre-determined by the linear terms $x_j$ and using a GLM initialization for the gradient descent fitting algorithm.

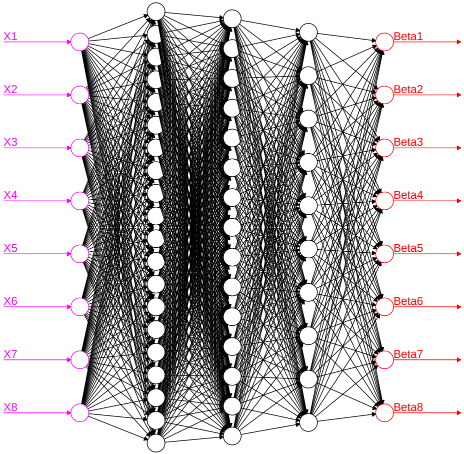
# Synthetic example

- Choose regression function for $\boldsymbol{x} = (x_1, \ldots, x_8)$

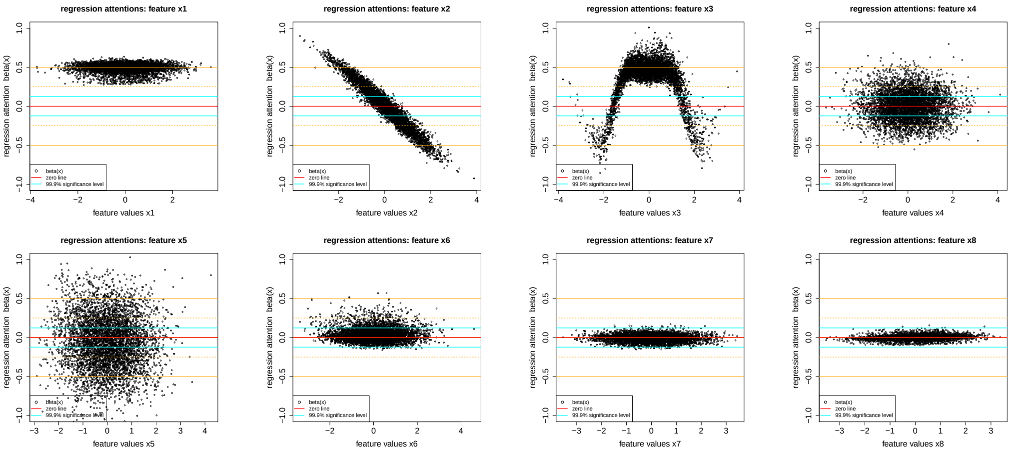
$$\mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3|\sin(2x_3) + \frac{1}{2}x_4x_5 + \frac{1}{8}x_5^2x_6.$$

- Note that $x_7$ and $x_8$ do not enter the regression function.
- Simulate $\boldsymbol{x}$ and Gaussian observations $Y$ with means $\mu(\boldsymbol{x})$ and unit variance.
- Fit a LocalGLMnet to the attention weights $\boldsymbol{\beta}(\boldsymbol{x}) = \boldsymbol{z}^{(d:1)}(\boldsymbol{x})$, of depth $d = 4$ with $(20, 15, 10, 8)$ hidden neurons, see next slide.
- Fitting is done with stochastic gradient descent, and using early stopping.

# LocalGLMnet for regression attentions $\beta(x)$



# Estimated regression attentions $\widehat{\beta}(\boldsymbol{x})$




$$\mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3|\sin(2x_3) + \frac{1}{2}x_4x_5 + \frac{1}{8}x_5^2x_6.$$

# Empirical Wald test

- Variables $x_7$ and $x_8$ do not enter the (true) regression function.
- This should imply $\widehat{\beta}_j(\boldsymbol{x}) \approx 0$ for $j = 7, 8$.
- We have empirical means and standard deviations

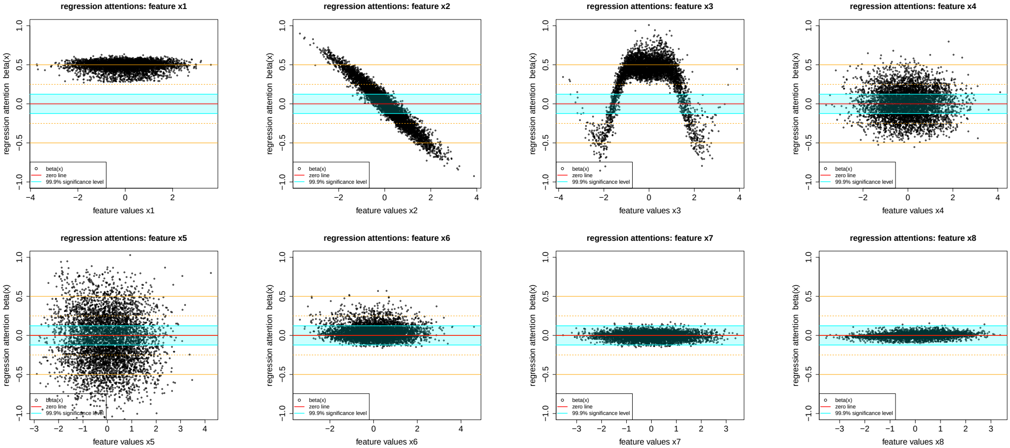
$$\bar{\beta}_7 = -0.0068, \ \bar{\beta}_8 = -0.0010 \approx 0 \qquad \text{and} \qquad \widehat{s}_7 = 0.0461, \ \widehat{s}_8 = 0.0290.$$

- Choose significance level $\alpha \in (0, 1)$ and consider

$$I_\alpha = \left[\Phi^{-1}(\alpha/2) \cdot \widehat{s}_{7/8}, \ \Phi^{-1}(1 - \alpha/2) \cdot \widehat{s}_{7/8}\right].$$

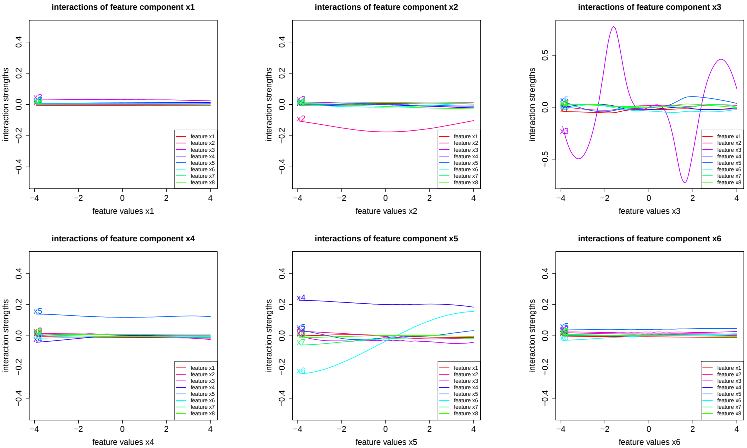
- Perform empirical Wald test for null hypothesis $H_0$: $\beta_j(\boldsymbol{x}) = 0$.

# Estimated $\widehat{\beta}(x)$ and variable selection



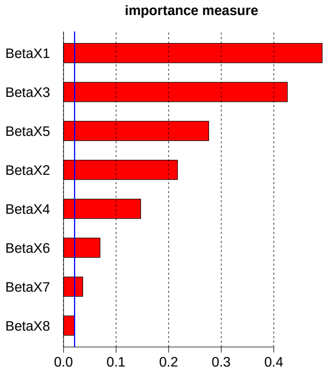

$$\mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3|\sin(2x_3) + \frac{1}{2}x_4x_5 + \frac{1}{8}x_5^2x_6.$$

# Gradients $\nabla \widehat{\beta}_j(\boldsymbol{x})$ for interactions




$$\mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3|\sin(2x_3) + \frac{1}{2}x_4x_5 + \frac{1}{8}x_5^2x_6.$$

# Importance measure




Define importance measure

$$\mathrm{VI}_j = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{\beta}_j(\boldsymbol{x}_i)\right|.$$

# Discussions

- LocalGLMnet provides an explainable regression model.
- LocalGLMnet allows for variable selection.
- LocalGLMnet allows for a natural importance measure.
- LocalGLMnet allows for the study of interactions.

- All considerations have been based on continuous covariates.
  - Categorical covariates are more difficult $\Rightarrow$ use regularization.
  - LocalGLMnet needs a bias regularization step to receive unbiasedness.
- Including too many random components leads to more over-fitting potential.
- If predictive power is insufficient: fit network on selected covariates.

# Group LASSO regularization

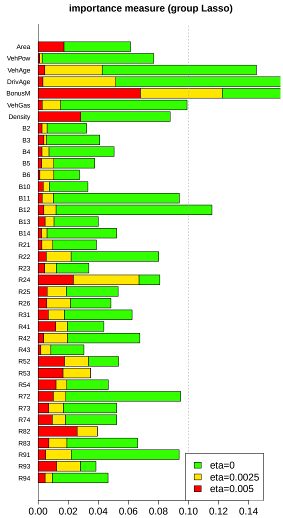
Assume covariates $\boldsymbol{x}$ have a natural group structure $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$. Consider for fitting the network parameter $\boldsymbol{\theta}$ a penalized loss

$$\underset{\boldsymbol{\theta}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n} L\left(Y_i, \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right) + \sum_{k=1}^{K} \eta_k \|\boldsymbol{\beta}_k(\boldsymbol{x}_i)\|_2,$$

with regularization parameters $\eta_k \geq 0$.

Shrinks unimportant weights $\beta_j(\boldsymbol{x})$ to $0$.

Figure shows initial car insurance example:
no regularization (green), medium regularization (yellow), strong regularization (red).



# Balance property

- Typically, gradient descent fitted networks do not fulfill the balance property

$$\sum_{i=1}^{n} \widehat{\mu}(\boldsymbol{x}_i) = \sum_{i=1}^{n} g^{-1}\left(\widehat{\beta}_0 + \sum_{j=1}^{q} \widehat{\beta}_j(\boldsymbol{x}_i) x_{i,j}\right) \neq \sum_{i=1}^{n} Y_i.$$

- This implies that insurance prices are biased.
  - Use bias correction according to Denuit-Charpentier-Trufin (2021) or
  - an additional GLM step with canonical link, see Wüthrich (2020),

$$\boldsymbol{x}_i \mapsto g(\mu(\boldsymbol{x}_i)) = \alpha_0 + \sum_{j=1}^{q} \alpha_j \, \widehat{\beta}_j(\boldsymbol{x}_i) x_{i,j},$$

for regression parameter $(\alpha_0, \ldots, \alpha_q)$ and (frozen) covariates $z_{i,j} = \widehat{\beta}_j(\boldsymbol{x}_i) x_{i,j}$.


# Thank you very much for your attention

## Contact


*Mario V. Wüthrich*

*RiskLab, ETH Zürich*

*+41 44 632 3390*

*mario.wuethrich@math.ethz.ch*


EAA e-Conference on
Data Science & Data Ethics

12 May 2022

# References


- Denuit, Charpentier, Trufin (2021). Autocalibration and Tweedie-dominance for insurance pricing with machine learning. *Insurance: Mathematics & Economics* **101**, 485-497.
- Richman, Wüthrich (2021). LocalGLMnet: interpretable deep learning for tabular data. *SSRN Manuscript*, ID 3892015.
- Richman, Wüthrich (2021). LASSO regularization within the LocalGLMnet architecture. *SSRN Manuscript*, ID 3927187.
- Wüthrich (2020). Bias regularization in neural network models for general insurance pricing. *European Actuarial Journal* **10**, 179-202.
- Wüthrich, Merz (2021). Statistical foundations of actuarial learning and its applications. *SSRN Manuscript*. Manuscript ID 3822407.