LocalGLMnet: interpretable deep learning

Mario V. Wüthrich
RiskLab, ETH Zurich

Joint work with Ronald Richman (Old Mutual Insure)

October 13, 2021

Online Joint Section Colloquium of the IAA

Regression modeling: a car insurance example

```
'data.frame':   678007 obs. of  13 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 ...
$ 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 ...
$ ClaimTotal: num  0 0 0 0 0 0 0 0 0 0 ...
$ ClaimNb   : num  0 0 0 0 0 0 0 0 0 0 ...
```

Goal. Find suitable regression function $\mu(\cdot)$ such that

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

where $\boldsymbol{x} \in \mathbb{R}^q$ are the covariates (explanatory variables) describing claim $Y \sim F_{\boldsymbol{x}}$.

GLM and neural networks

- GLM: Choose strictly monotone link function g and assume

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

  That is, we have a linear function in covariate $\boldsymbol{x}$ after applying link g.

- Neural network: Set for regression function

$$\boldsymbol{x} \mapsto g\left(\mu^{\mathrm{NN}}(\boldsymbol{x})\right) = \left\langle \boldsymbol{\beta}, \boldsymbol{z}^{(d:1)}(\boldsymbol{x}) \right\rangle,$$

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

- The neural network learns a new representation $\boldsymbol{z}^{(d:1)}(\boldsymbol{x})$ of the covariate $\boldsymbol{x}$. Typically, a well trained neural network outperforms a GLM.

- **Drawback.** The network solution is often not interpetable and explainable.

LocalGLMnet: definition

- GLM:

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

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

- Choose a neural 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

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

LocalGLMnet: interpretations

- LocalGLMnet:

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

  - ⋆ 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

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

  - ⋆ 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.

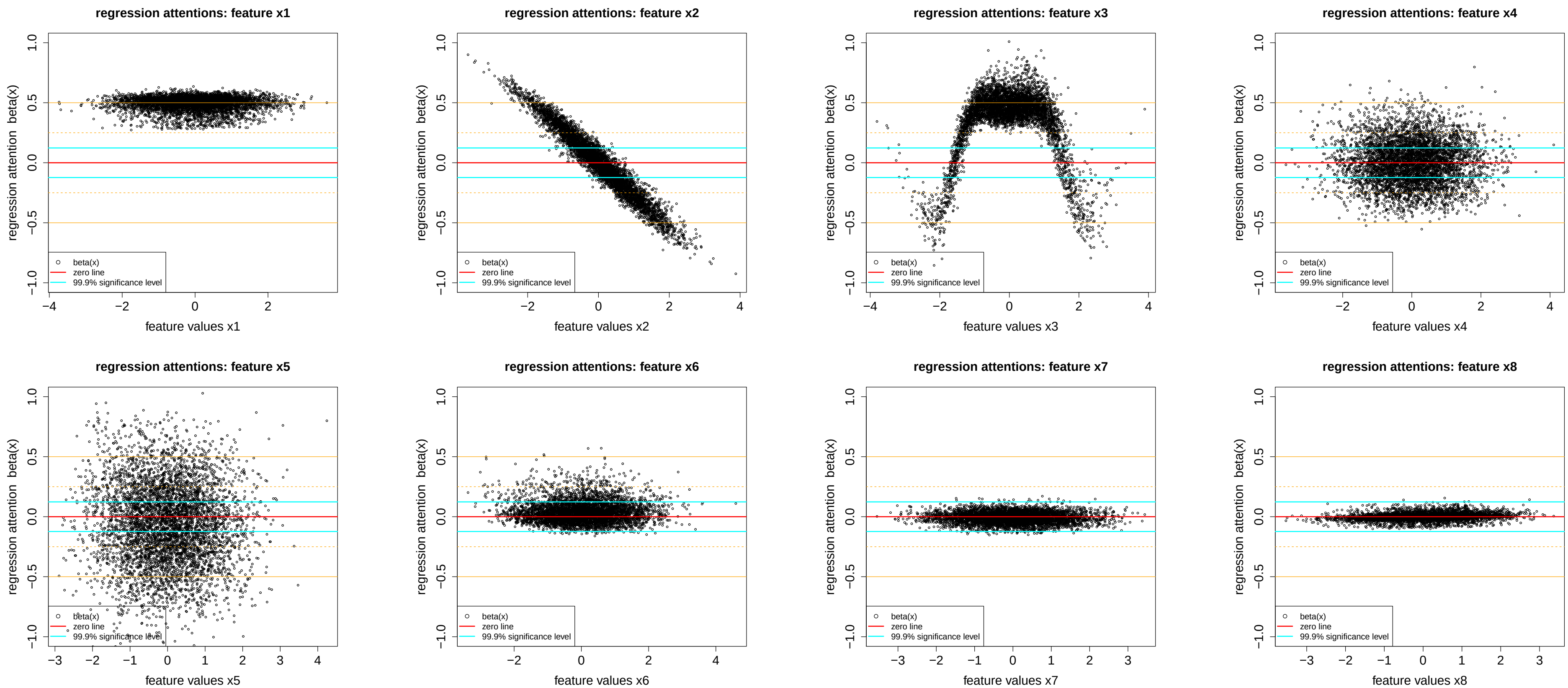
Synthetic example

- Choose regression function

$$\boldsymbol{x} = (x_1, \ldots, x_8)^\top \in \mathbb{R}^8 \mapsto \mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3| \sin(2x_3) + \frac{1}{2}x_4 x_5 + \frac{1}{8}x_5^2 x_6.$$

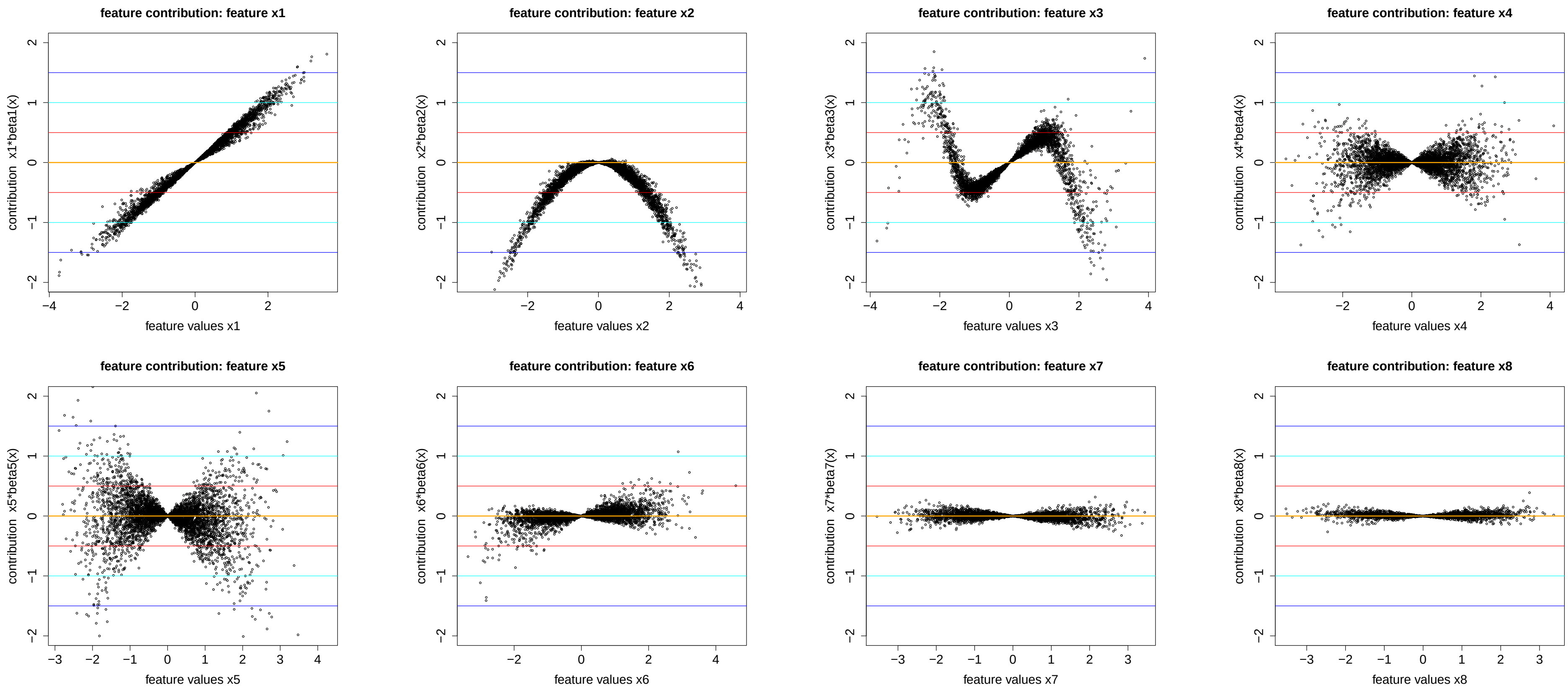
- Note that x_7 and x_8 do not enter the regression function.
- We simulate Gaussian observations Y with means $\mu(\boldsymbol{x})$ and unit variance.
- We fit a LocalGLMnet for $\boldsymbol{\beta}(\boldsymbol{x}) = \boldsymbol{z}^{(d:1)}(\boldsymbol{x})$ of depth $d = 4$ with $(20, 15, 10, 8)$ hidden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

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



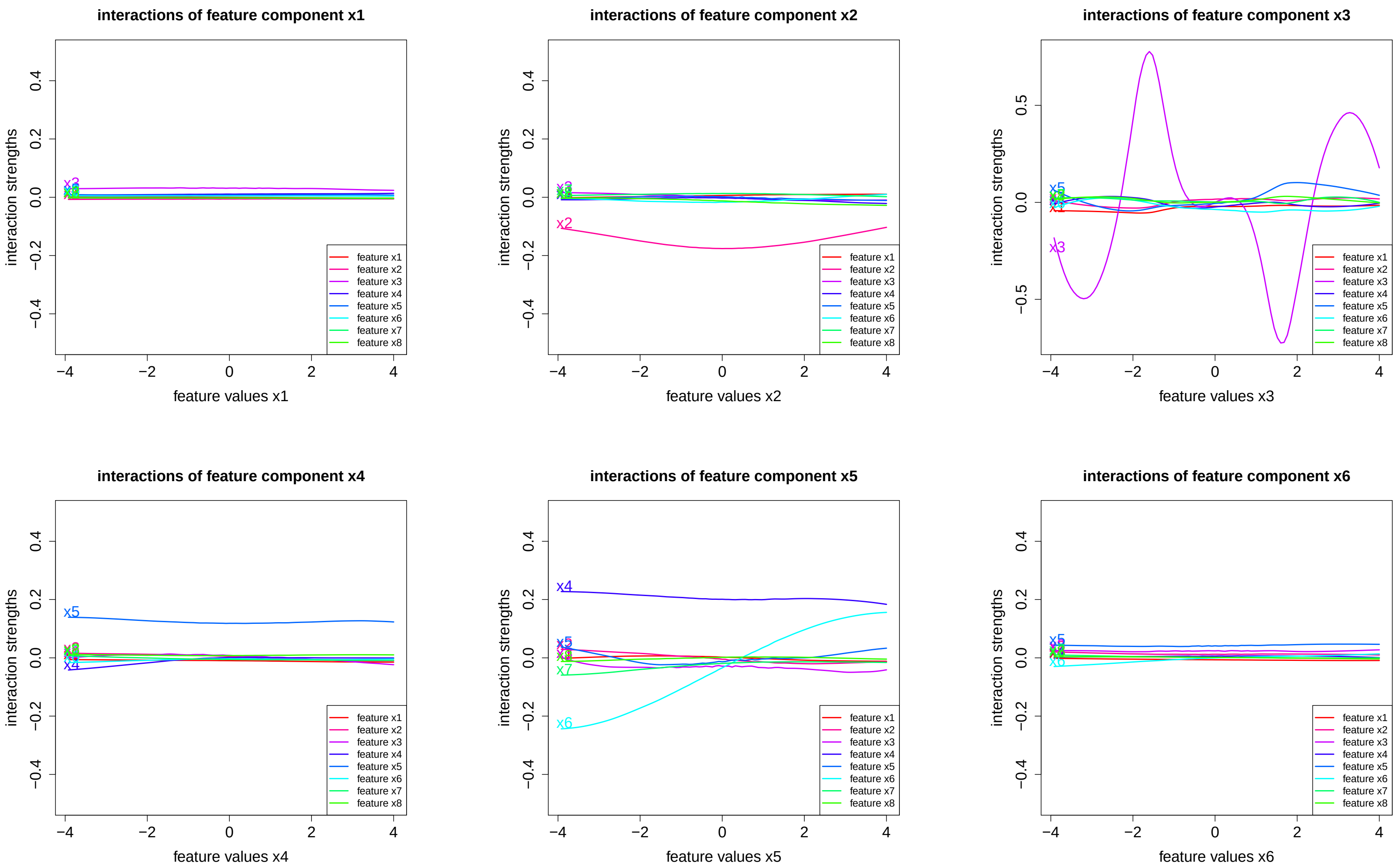
$$\mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3|\sin(2x_3) + \frac{1}{2}x_4x_5 + \frac{1}{8}x_5^2x_6.$$

Estimated terms $\widehat{\beta}_j(\boldsymbol{x})x_j$



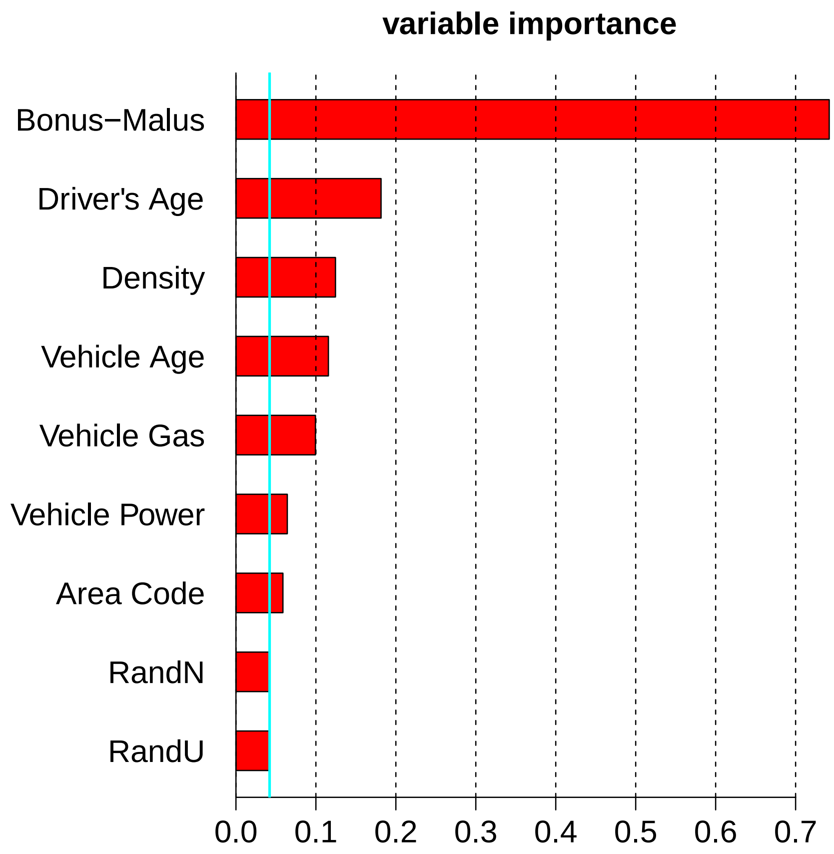
$$\mu(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{4}x_2^2 + \frac{1}{2}|x_3|\sin(2x_3) + \frac{1}{2}x_4x_5 + \frac{1}{8}x_5^2x_6.$$

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



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

Variable importance: car example



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

Take Aways

- A neural network as an extension of a GLM (representation learning).
- Neural networks suffer the deficiency of not being interpretable.
- We have introduced the LocalGLMnet that estimates the GLM parameters through a neural network.
- This leads to interpretable results that allow for the study of variable importance.
- Variable selection can be done in LocalGLMnets.
- Interactions can be studied in LocalGLMnets.

- LocalGLMnet: interpretable deep learning for tabular data.
  SSRN Manuscript 3892015 (2021).

  `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3892015`