SFI Centre for Research Training in Foundations of Data Science

Centre for Research Training

Science Foundation Ireland SFI For what's next

A Statistical-Modelling Approach to Neural Networks

Andrew McInerney, Kevin Burke

University of Limerick

Pukyong National University, 27 March 2024

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Background

SFI Centre for Research Training in Foundations of Data Science

- Research: Neural networks from a statistical-modelling perspective

selectnn

interpretnn

Agenda

- Introduction
- Model Selection
  - Penalised Selection
  - Stepwise Selection
- Model Interpretation

Introduction

Background

Neural networks originated from attempts to model the human brain.

Early influential papers:

- McCulloch and Pitts (1943)
- Rosenblatt (1958)
- Rumelhart, Hinton and Williams (1986)

Background

Interest within the statistics community in the late 1980s and early 1990s.

Comprehensive reviews provided by White (1989), Ripley (1993), Cheng and Titterington (1994).

However, majority of research took place outside the field of statistics (Breiman, 2001; Hooker and Mentch, 2021).

Background

Renewed interest in merging statistical models and neural networks.

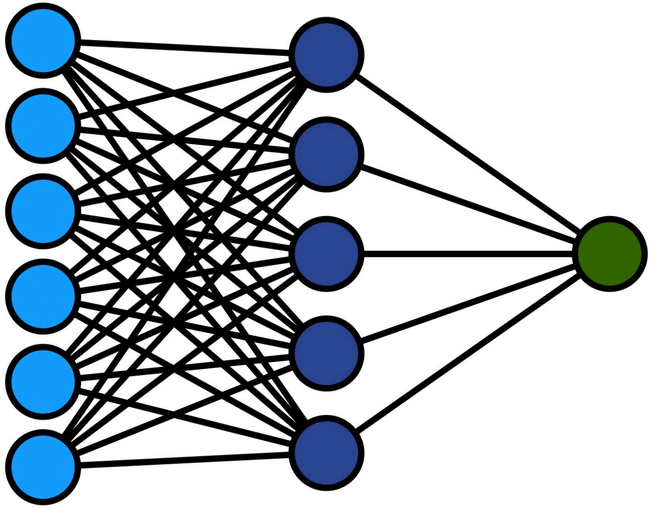
From a statistical viewpoint:

- Distributional regression (Rugamer et al., 2020, 2021).
- Mixed modelling (Tran et al., 2020).

From a machine-learning viewpoint:

- Neural Additive Models (Agarwal et al., 2020)

Feedforward Neural Networks



$$\mathrm{NN}(x_i) = \gamma_0 + \sum_{k=1}^{q} \gamma_k \, \phi\left(\sum_{j=0}^{p} \omega_{jk} x_{ji} \right)$$

Data Application

Insurance Data (Kaggle)

1,338 beneficiaries enrolled in an insurance plan

Response:

`charges`

6 Explanatory Variables:

`age, sex, bmi, children, smoker, region`

R Implementation

Many packages available to fit neural networks in R.

Some popular packages are:

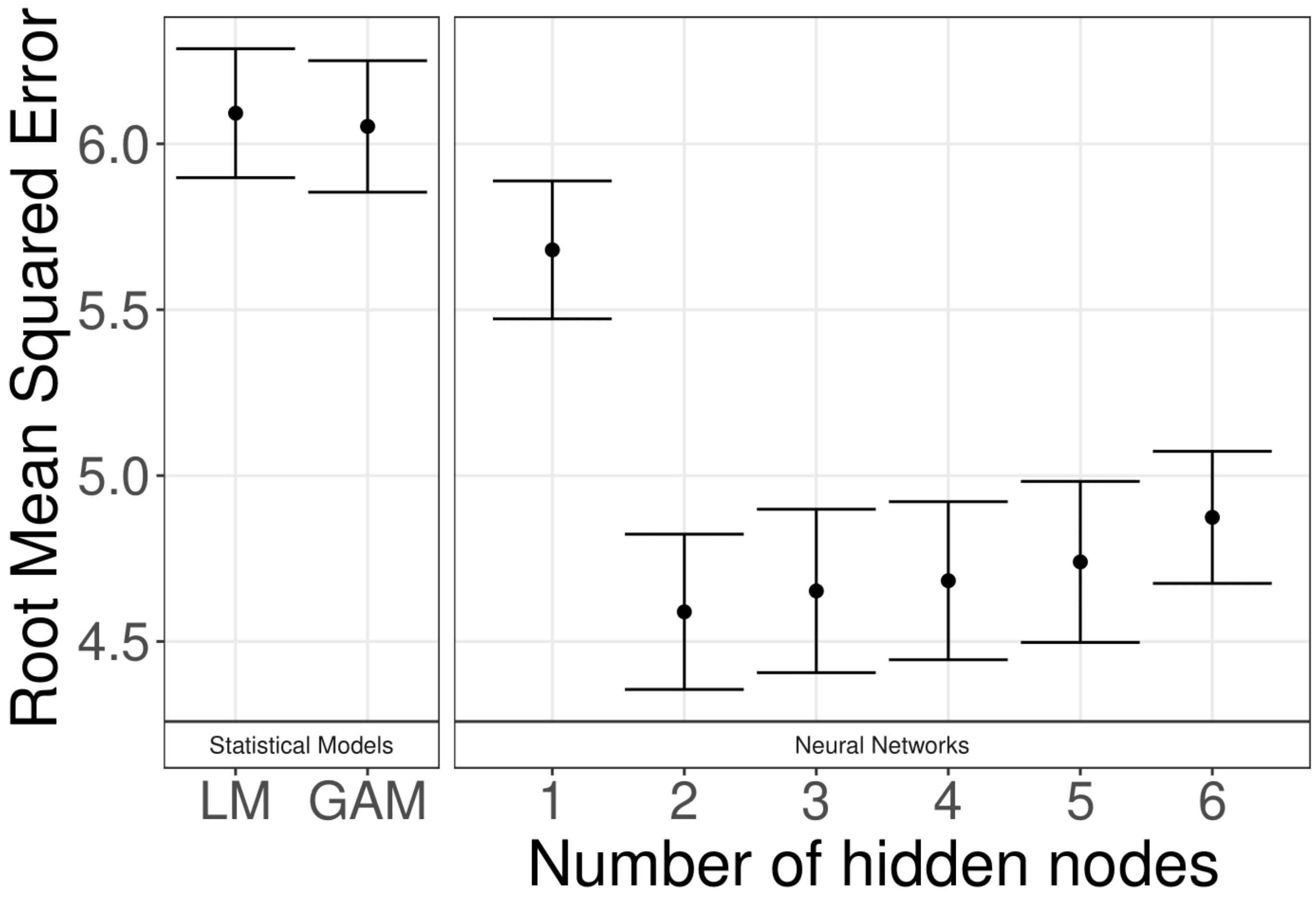
- `nnet`
- `neuralnet`
- `keras`
- `torch`

R Implementation: nnet

```
library(nnet)
nn <- nnet(charges ~ ., data = insurance, size = 2, maxit = 2
             linout = TRUE)
summary(nn)
```

```
## a 8-2-1 network with 21 weights
##  b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1 i6->h1 i7->h1 i8->
##   1.39  -0.43   0.08   0.03  -0.08  -3.16   0.07   0.11   0.
##  b->h2 i1->h2 i2->h2 i3->h2 i4->h2 i5->h2 i6->h2 i7->h2 i8->
##   6.31   0.04   0.13   2.19  -0.11  -6.19   0.15   0.12   0.
##  b->o h1->o h2->o
##  1.08 -4.82  2.45
## [...]
```

Motivation



Root Mean Squared Error
6.0
5.5
5.0
4.5
Statistical Models
Neural Networks
LM
GAM
1
2
3
4
5
6
Number of hidden nodes

Statistical Perspective

$$y_i = \mathrm{NN}(x_i) + \varepsilon_i,$$

where

$$\varepsilon_i \sim N(0, \sigma^2)$$

$$\ell(\theta, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathrm{NN}(x_i))^2$$

Model Selection

Penalised Selection

Smooth Information Criterion

Statistics and Computing (2023) 33:71
https://doi.org/10.1007/s11222-023-10204-8

ORIGINAL PAPER

Variable selection using a smooth information criterion for distributional regression models

Meadhbh O'Neill[1] · Kevin Burke[1]

Received: 7 March 2022 / Accepted: 3 January 2023 / Published online: 21 April 2023

Smooth Information Criterion

$$\mathrm{BIC} = -2\ell(\theta) + \log(n)\left[\sum_{j=1}^{p} |\beta_j|^0 + 1\right]$$

where

$$\ell(\theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2$$

Smooth Information Criterion

$$\mathrm{BIC} = -2\ell(\theta) + \log(n)\left[\sum_{j=1}^{p} |\beta_j|^0 + 1\right]$$

Introduce "smooth BIC":

$$\mathrm{SBIC} = -2\ell(\theta) + \log(n)\left[\sum_{j=1}^{p} \frac{\beta_j^2}{\beta_j^2 + \epsilon^2} + 1\right]$$

Extending to Neural Networks

$$\mathbb{E}(y) = \mathrm{NN}(X, \theta)$$

where

$$\mathrm{NN}(X, \theta) = \phi_o \left[\gamma_0 + \sum_{k=1}^{q} \gamma_k \phi_h \left(\sum_{j=0}^{p} \omega_{jk} x_j \right) \right]$$

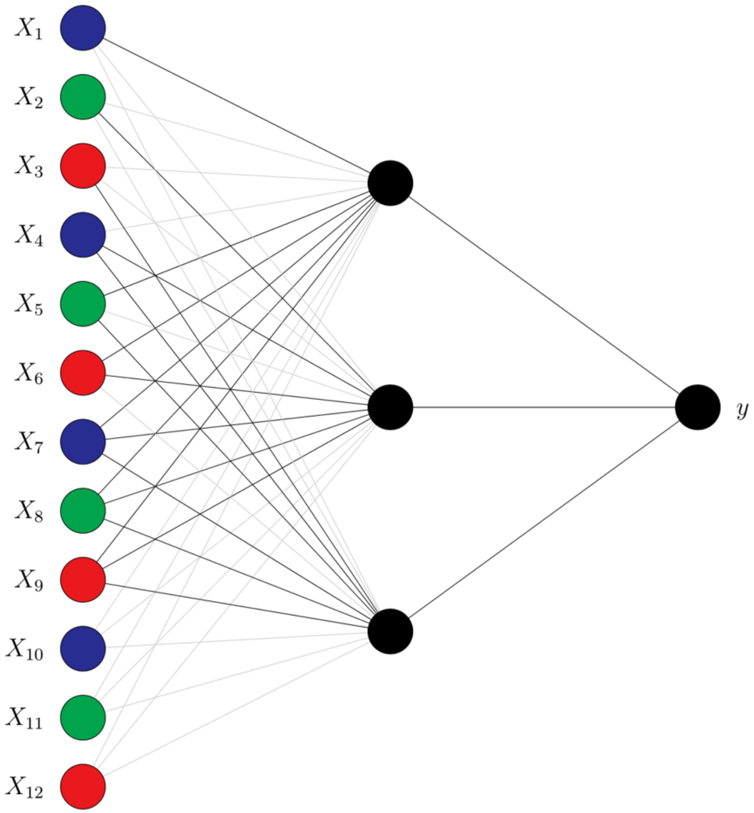
Extending to Neural Networks

$$\mathrm{SBIC} = -2\ell(\theta) + \log(n)\left[\sum_{jk}\frac{\omega_{jk}^2}{\omega_{jk}^2 + \epsilon^2} + \sum_{k}\frac{\gamma_k^2}{\gamma_k^2 + \epsilon^2} + q + 1\right]$$

where

$$\ell(\theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathrm{NN}(x_i))^2$$

Simulation Setup

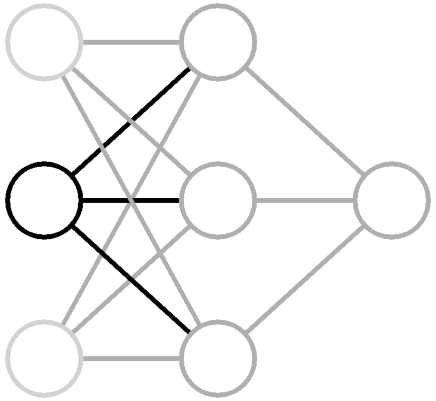


E = Exponential, B = Bernoulli, M = multivariate normal (correlated)

Results

n	Weights TPR	Weights TNR	Weights FDR	Input nodes TPR	Input nodes TNR	Input nodes FDR
100	0.95	0.68	0.21	0.99	0.37	0.17
250	1.00	0.96	0.03	1.00	0.90	0.03
500	1.00	0.98	0.02	1.00	0.93	0.02
1000	1.00	0.99	0.00	1.00	0.99	0.00

Extending to Group Sparsity



Single penalty:

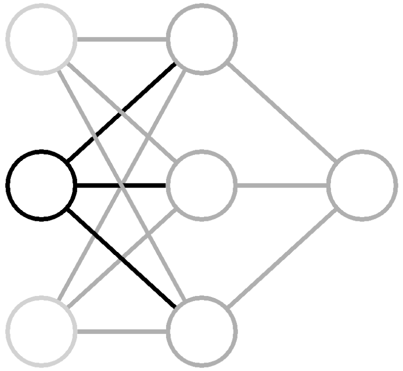
$$\frac{\omega_{jk}^2}{\omega_{jk}^2 + \epsilon^2}$$

Group penalty:

$$\mathrm{card}(\omega_j) \times \frac{||\omega_j||_2^2}{||\omega_j||_2^2 + \epsilon^2}$$

Group Sparsity

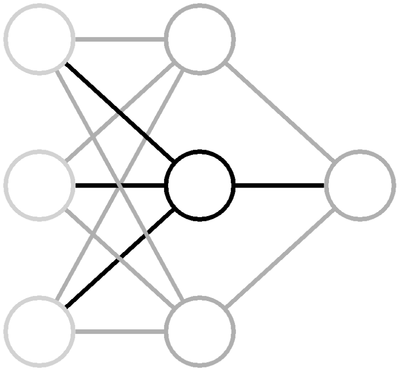
Input-neuron penalization



$$\text{IN-SBIC} = -2\ell(\theta) + \log(n)\left[q \times \sum_j \frac{||\omega_j||_2^2}{||\omega_j||_2^2 + \epsilon^2} + \sum_k \frac{\gamma_k^2}{\gamma_k^2 + \epsilon^2} + q + 1\right]$$

where $\omega_j = (\omega_{j1}, \omega_{j2}, \ldots, \omega_{jq})^T$

Group Sparsity

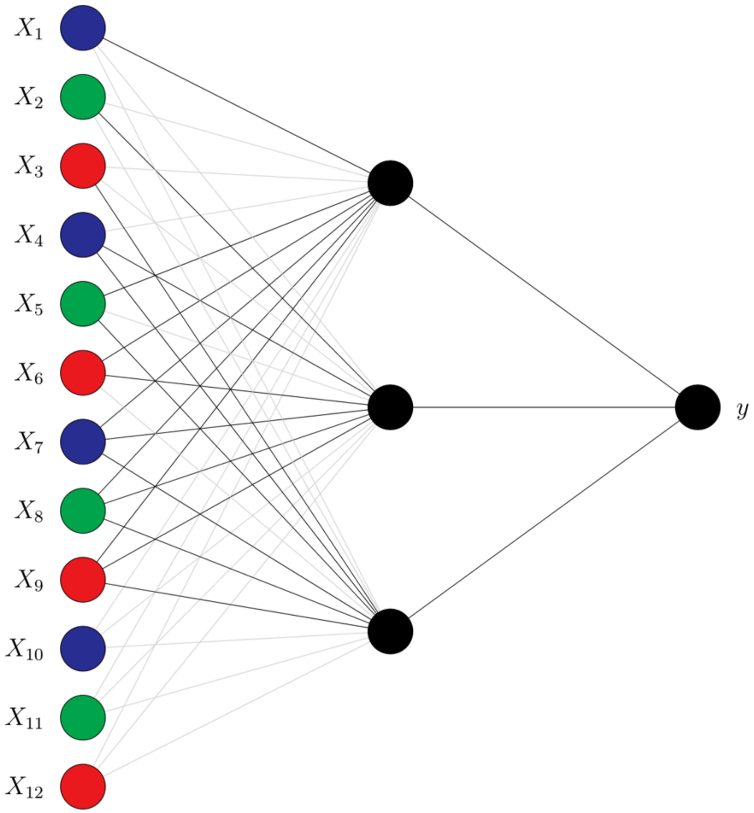


Hidden-neuron penalization

$$\text{HN-SBIC} = -2\ell(\theta) + \log(n)\left[(p+1) \times \sum_k \frac{||\theta^{(k)}||_2^2}{||\theta^{(k)}||_2^2 + \epsilon^2} + q + 1\right]$$

where $\theta^{(k)} = (\omega_{1k}, \omega_{2k}, \ldots, \omega_{pk}, \gamma_k)^T$

Simulation Setup

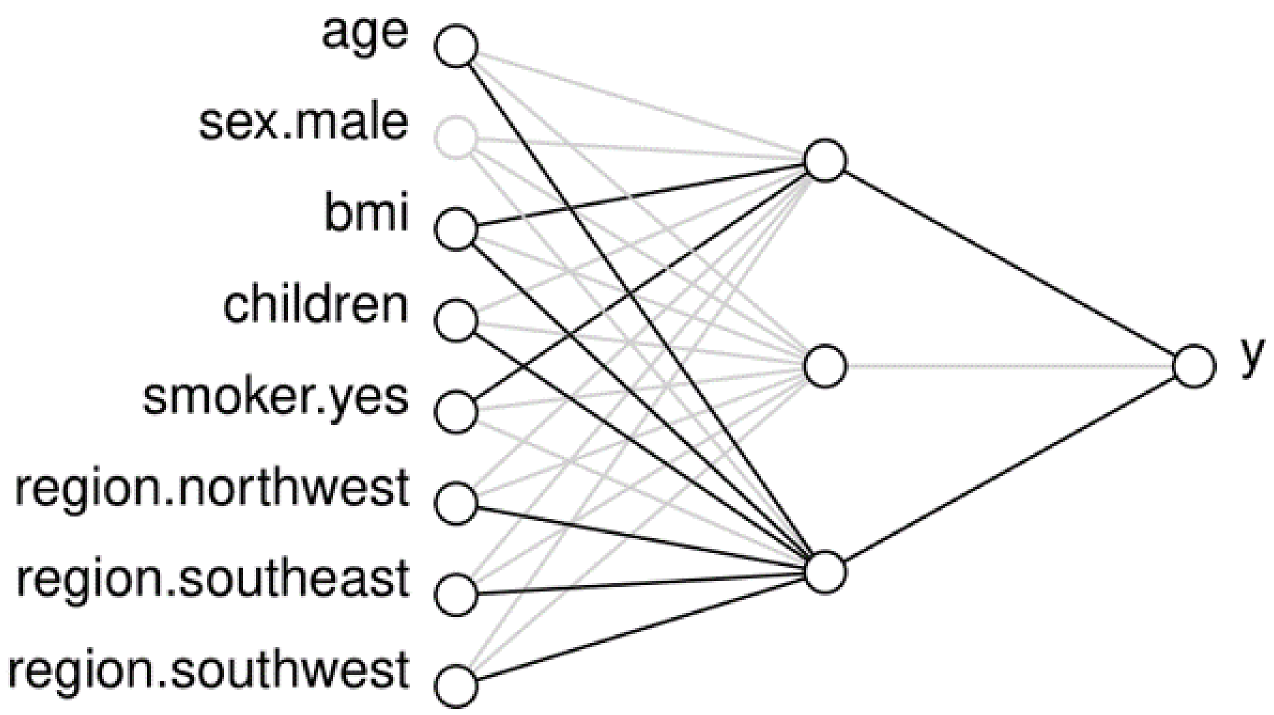


E = Exponential, B = Bernoulli, M = multivariate normal (correlated)

Results (IN-SBIC)

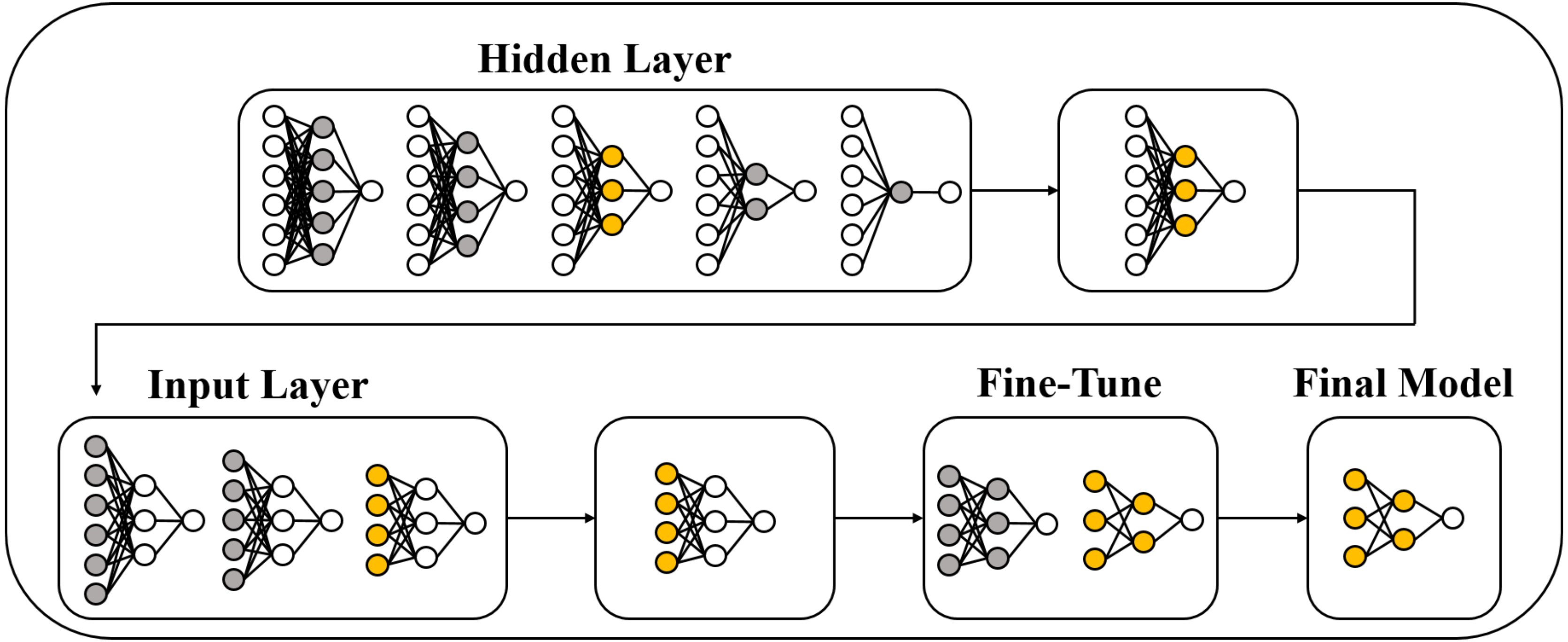
	Weights			Input nodes		
n	TPR	TNR	FDR	TPR	TNR	FDR
100	0.99	0.35	0.35	0.97	0.63	0.10
250	1.00	0.50	0.30	1.00	1.00	0.00
500	1.00	0.50	0.30	1.00	1.00	0.00
1000	1.00	0.50	0.30	1.00	1.00	0.00

Data Application - Results



Stepwise Selection

Model Selection



A Statistically-Based Approach to Feedforward Neural Network Model Selection (arXiv:2207.04248)

Insurance: Model Selection

```r
library(selectnn)
nn <- selectnn(charges ~ ., data = insurance, Q = 8,
               n_init = 5)
summary(nn)
```

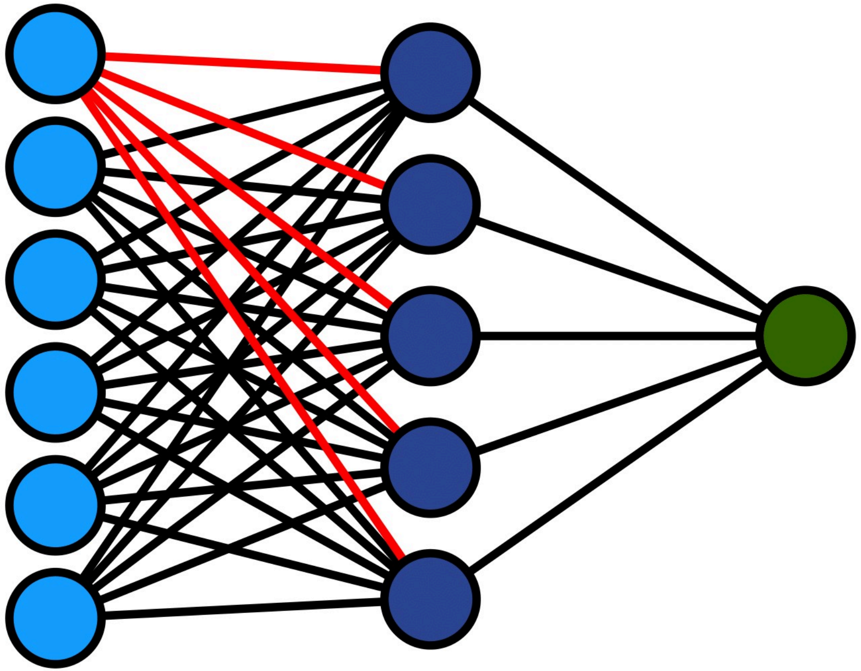
```
## [...]
## Number of input nodes: 4
## Number of hidden nodes: 2
##
## Value: 1218.738
##          Covariate Selected Delta.BIC
##         smoker.yes      Yes  2474.478
##                bmi      Yes   919.500
##                age      Yes   689.396
##           children      Yes    13.702
## [...]
```

Model Interpretation

Proposed Solution: interpretnn

```r
# install.packages("devtools")
library(devtools)
install_github("andrew-mcinerney/interpretnn")
```

Significance Testing



Wald test:

$$\omega_j = (\omega_{j1}, \omega_{j2}, \ldots, \omega_{jq})^T$$

$$H_0 : \omega_j = 0$$

$$(\hat{\omega}_j - \omega_j)^T \Sigma_{\hat{\omega}_j}^{-1} (\hat{\omega}_j - \omega_j) \xrightarrow{\mathcal{D}} \chi_q^2$$

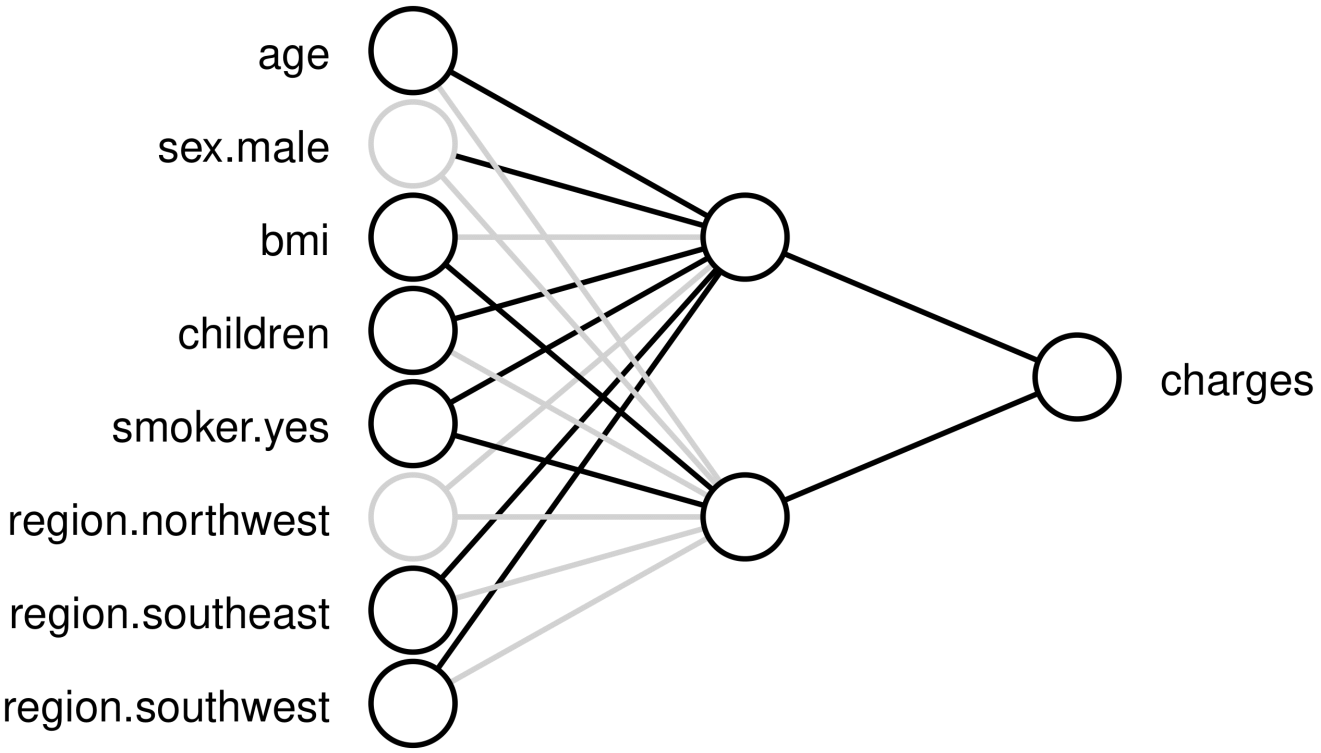
Insurance: Model Summary

```
intnn <- interpretnn(nn)
summary(intnn)
```

```
## Coefficients:
##                              Weights |      X^2     Pr(> X^
##               age    (-0.43***, 0.04) |  41.4363 1.01e-09 *
##          sex.male       (0.08*, 0.13) |   5.5055 6.38e-02 .
##               bmi     (0.03, 2.19***) | 105.6106 0.00e+00 *
##          children   (-0.08***, -0.11.) |  19.0146 7.43e-05 *
##        smoker.yes (-3.16***, -6.19***) | 250.6393 0.00e+00 *
##  region.northwest       (0.07., 0.15) |   2.8437 2.41e-01
##  region.southeast       (0.11*, 0.12) |   6.2560 4.38e-02 *
##  region.southwest      (0.15**, 0.14) |  10.8218 4.47e-03 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Insurance: Model Summary

```
plotnn(intnn)
```



Covariate-Effect Plots

$$\widehat{\overline{\mathrm{NN}}}_j(x) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{NN}(x_{i,1}, \ldots, x_{i,j-1}, x, x_{i,j+1}, \ldots)$$

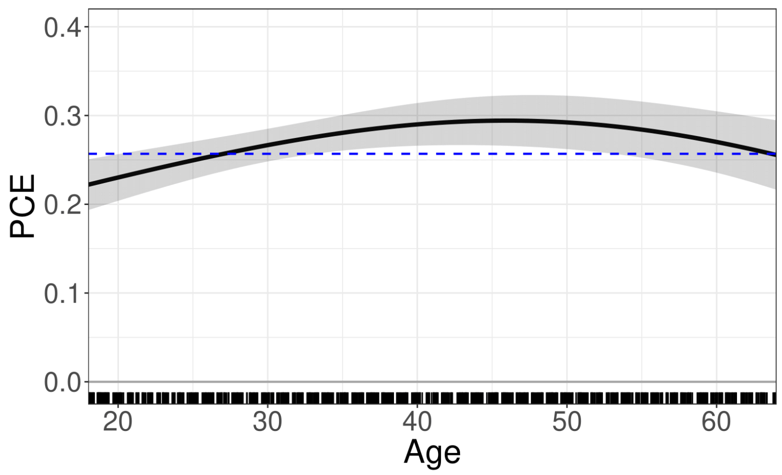
Covariate-effect plots of the following form:

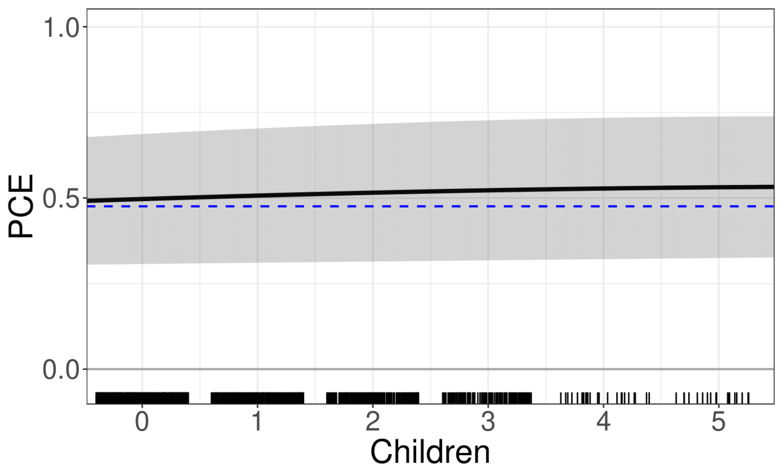
$$\hat{\beta}_j(x, d) = \widehat{\overline{\mathrm{NN}}}_j(x + d) - \widehat{\overline{\mathrm{NN}}}_j(x)$$

Usually set $d = \mathrm{SD}(x_j)$

Insurance: Covariate Effects

```
plot(intnn, conf_int = TRUE, which = c(1, 4))
```





Summary

- Treat neural networks as statistical models
- Perform penalised and stepwise model selection
- Use hypothesis tests and covariate-effect plots for interpretation

References

- McInerney, A., & Burke, K. (2022). A statistically-based approach to feedforward neural network model selection. *arXiv preprint arXiv:2207.04248*.
- McInerney, A., & Burke, K. (2023). Interpreting feedforward neural networks as statistical models. *arXiv preprint arXiv:2311.08139*.
- McInerney, A., & Burke, K. (2024). Combining a smooth information criterion with neural networks. *To appear on arXiv*.

R Packages

```
devtools::install_github(c("andrew-mcinerney/selectnn",
                           "andrew-mcinerney/interpretnn"))
```


andrew-mcinerney @amcinerney_ andrew.mcinerney@ul.ie