SFI Centre for Research Training in Foundations of Data Science

Centre for Research Training
Science Foundation Ireland For what's next

Feedforward Neural Networks as Statistical Models

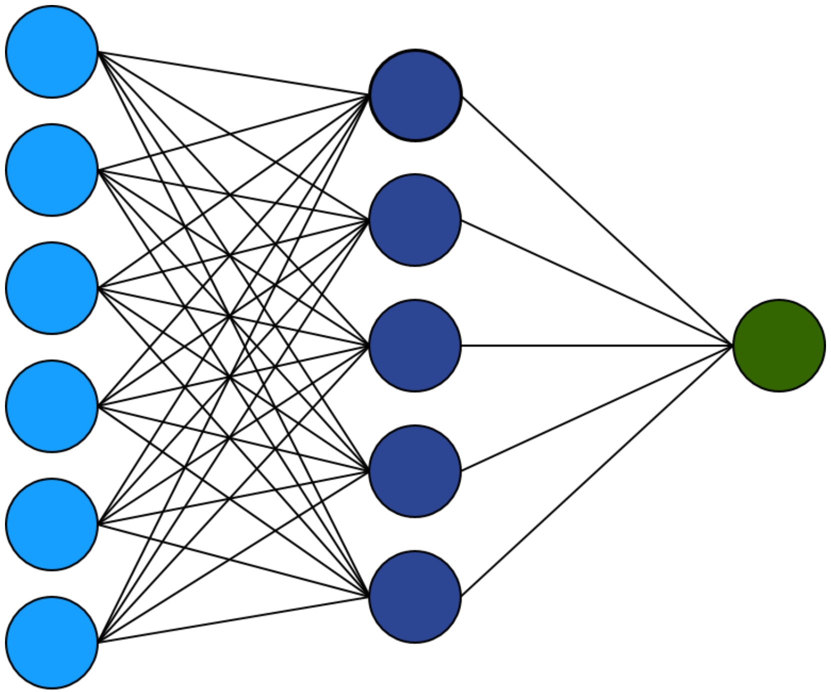
Andrew McInerney, Kevin Burke

University of Limerick

CMStatistics, 19 Dec 2022

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Feedforward Neural Networks



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

Data Application

Boston Housing Data (Kaggle)

506 communities in Boston, MA.

Response:

- `medv` (median value of owner-occupied homes)

12 Explanatory Variables:

- `rm` (average number of rooms per dwelling)
- `lstat` (proportion of population that are disadvantaged)

R Implementation

Many packages available to fit neural networks in R.

Some popular packages are:

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

R Implementation: nnet

```r
library(nnet)
nn <- nnet(medv ~ ., data = Boston, size = 8, maxit = 5000,
              linout = TRUE)
summary(nn)
```

```
## a 11-8-1 network with 105 weights
## options were - linear output units
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->
##    2.79    5.92    0.34    1.31    0.23   -1.31   -2.67    0.77   -0.22    1.
## i10->h1 i11->h1
##    1.20    1.26
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->
##   20.53    5.59    3.52   -0.64   12.64   -5.25   -4.12    0.24    2.64    0.
## i10->h2 i11->h2
##  -21.17    4.03
## [...]
```

Statistical Perspective

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

where

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

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

Uncertainty Quantification

Then, as $n \to \infty$

$$\hat{\theta} \sim N[\theta, \Sigma = \mathcal{I}(\theta)^{-1}]$$

Estimate Σ using

$$\hat{\Sigma} = I_o(\hat{\theta})^{-1}$$

However, inverting $I_o(\hat{\theta})$ can be problematic in neural networks.

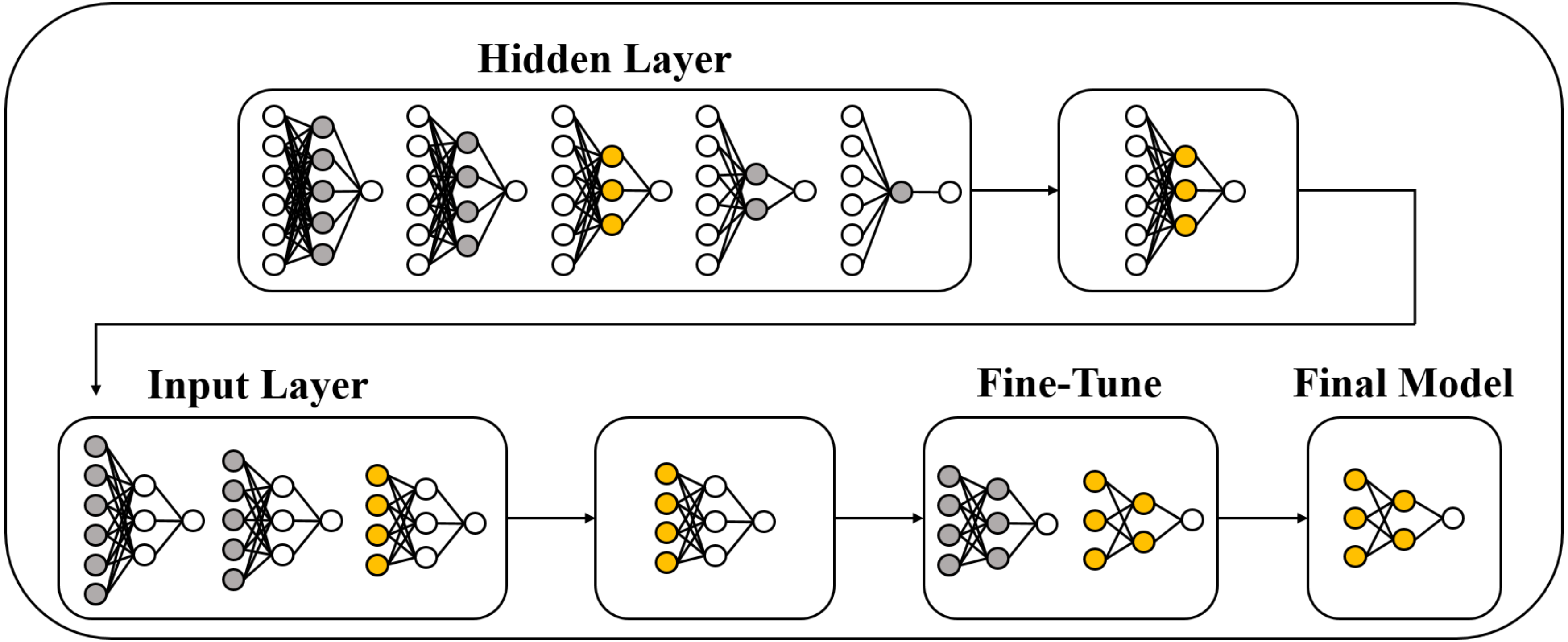
Redundancy

Redundant hidden nodes can lead to issues of unidentifiability for some of the parameters (Fukumizu 1996).

Redundant hidden nodes $\implies$ Singular information matrix.

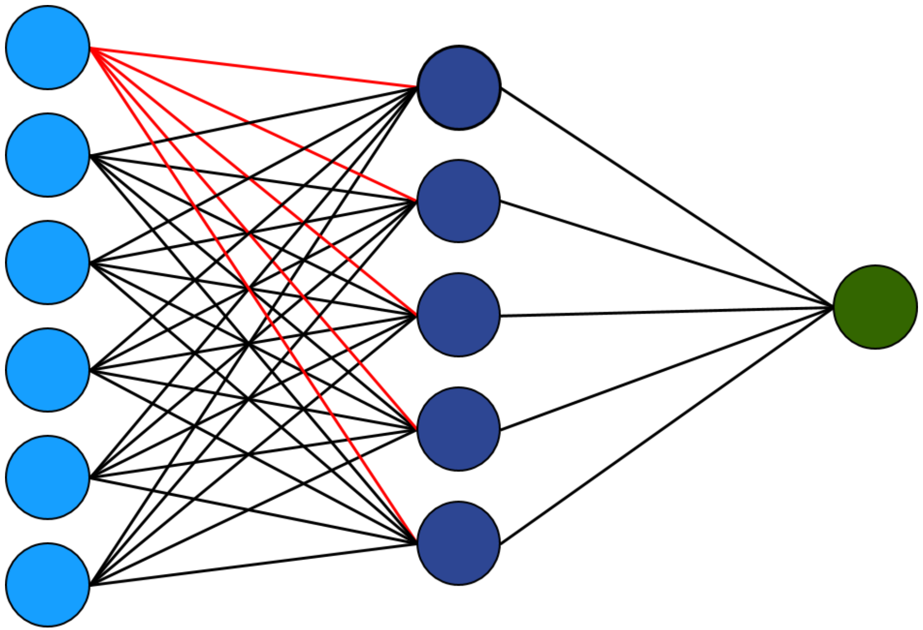
Model selection is required.

Model Selection



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

Hypothesis Testing



Wald test:

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

$$H_0 : \omega_j = 0$$

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

Covariate-Effect Plots

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

Propose covariate-effect plots of the following form:

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

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

Associated uncertainty via delta method.

R Implementation

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

Data Application (Revistied)

Boston Housing Data (Kaggle)

506 communities in Boston, MA.

Response:

- `medv` (median value of owner-occupied homes)

12 Explanatory Variables:

- `rm` (average number of rooms per dwelling)
- `lstat` (proportion of population that are disadvantaged)

Boston Housing: Model Selection

```r
library(statnn)
nn <- selectnn(medv ~ ., data = Boston, Q = 10,
               n_init = 10, maxit = 5000)
summary(nn)
```

```
## Call:
## selectnn.formula(formula = medv ~ ., data = Boston, Q = 10, n_init = 10,
##     maxit = 5000)
##
## Number of input nodes: 8
## Number of hidden nodes: 4
##
## Inputs:
##  Covariate Selected Delta.BIC
##         rm      Yes   236.907
##      lstat      Yes   168.023
## [...]
```

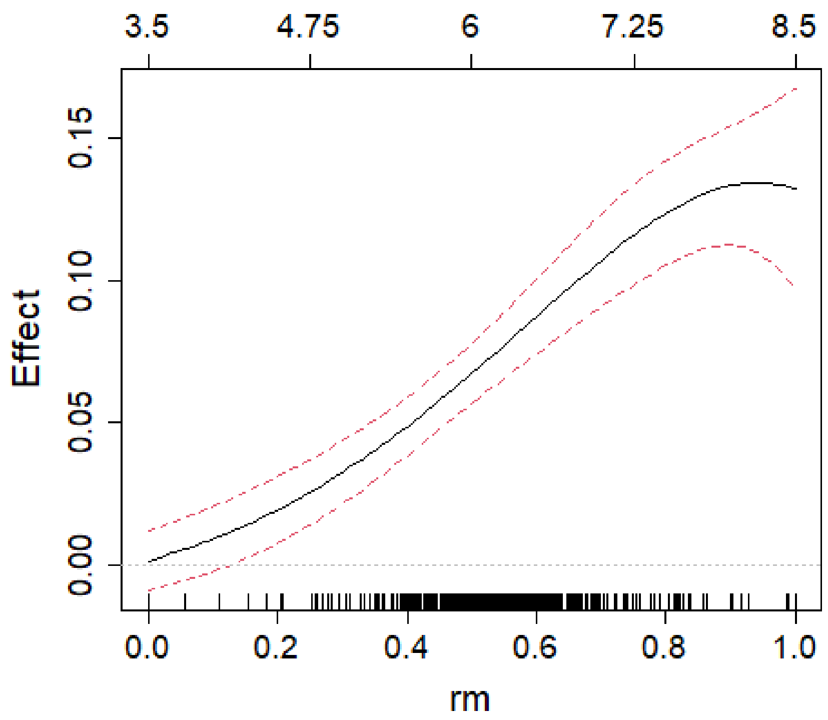
Boston Housing: Model Summary

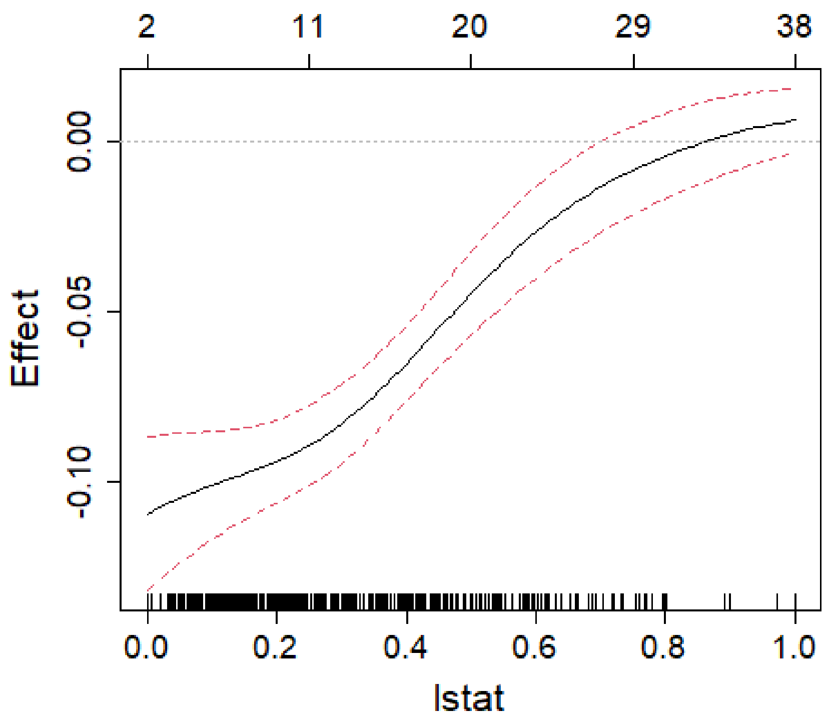
```
stnn <- statnn(nn)
summary(stnn)
```

```
## [...]
## Coefficients:
##                                              Wald
##            Estimate Std. Error |      X^2    Pr(> X^2)
##      crim -0.115769   0.019085 | 109.8369 0.00e+00 ***
##     indus -0.176500   0.018028 |  51.6302 1.65e-10 ***
##       nox -0.163091   0.020639 |  39.4919 5.51e-08 ***
##        rm  0.201211   0.017924 |  45.5051 3.12e-09 ***
##       dis  0.101701   0.022437 |  14.6031 5.60e-03 **
##       rad -0.099667   0.019687 | 107.3354 0.00e+00 ***
##   ptratio -0.192649   0.016672 |   7.8733 9.63e-02 .
##     lstat -0.263402   0.014443 |  50.2500 3.20e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [...]
```

Boston Housing: Covariate-Effect Plots

```
plot(stnn, conf_int = TRUE, method = "deltamethod", which = c(4, 8))
```





Summary

Feedforward neural networks are non-linear regression models.

Calculation of a likelihood function allows for uncertainty quantification.

Our R package extends existing neural network packages to allow for a more interpretable, statistically-based output.

References

Fukumizu, K. (1996). A regularity condition of the information matrix of a multilayer perceptron network. Neural Networks, 9(5):871–879.

McInerney, A. and Burke, K. (2022). A Statistically-Based Approach to Feedforward Neural Network Model Selection. arXiv preprint arXiv:2207.04248.

R Package

```
devtools::install_github("andrew-mcinerney/statnn")
```

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