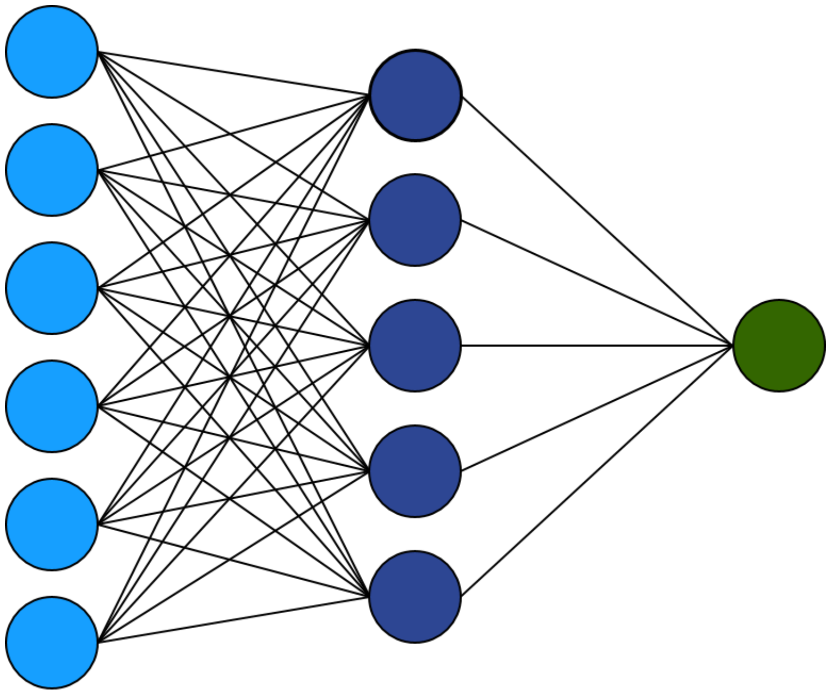
Feedforward Neural Networks as Statistical Models

Andrew McInerney, University of Limerick

11 Oct 2022

Feedforward Neural Networks



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

Statistical Perspctive

Feedforward neural networks are non-linear regression models.

$$y_i = g(x_i) + \varepsilon_i, \; \varepsilon_i \sim N(0, \sigma^2)$$

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

Uncertainty Quantification

Then, as $n \to \infty$

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

Estimate Σ using

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

However, inverting $I_o(\theta)$ be problematic.

Redundancy

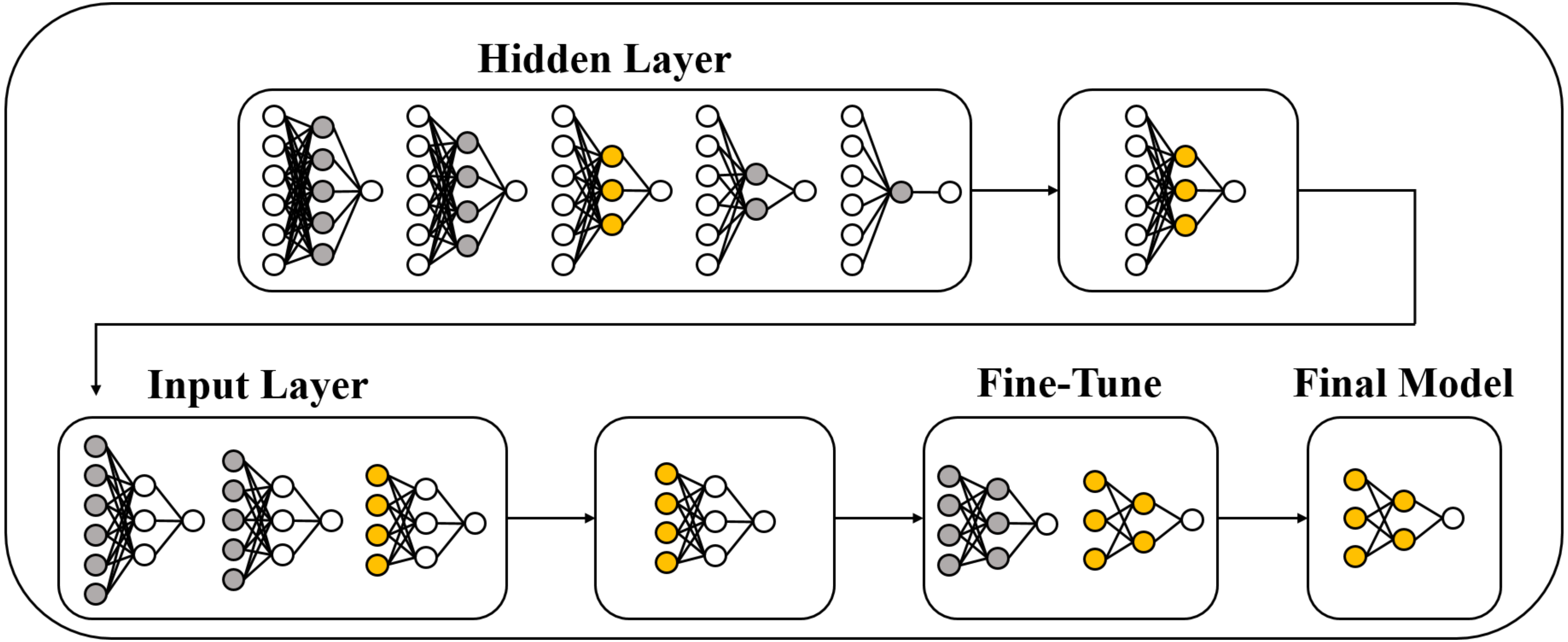
- Redundant hidden nodes can lead to issues of unidentifiability for some of the parameters (Fukumizu 1996)

- Redundant hidden nodes $\implies$ Singular information matrix.

- Model selection is required.

Model Selection

Select number of hidden nodes and input nodes.



Hypothesis Testing

Wald test:

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

$$H_0 : \omega_j = 0$$

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

Likelihood ratio test:

$$2(\ell_1 - \ell_0) \sim \chi_q^2$$

Covariate-Effect Plots

Propose covariate-effect plots of the following form:

$$\hat{\beta}_j(x) = \frac{1}{n}\sum_{i=1}^{n}\left[g(x+\sigma_j, X \setminus x_{ij}) - g(x, X \setminus x_{ij})\right]$$

And their associated uncertainty:

$$\hat{\beta}_j(x) \sim N[\beta_j(x), \nabla_\theta^T \beta_j(x)\ \Sigma\ \nabla_\theta \beta_j(x)]$$

R Implementation

selectnn

Data Application

Boston Housing Data

506 communities in Boston, MA (James et al., 2022)

Response:

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

12 Explanatory Variables:

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

Boston Housing: Model Selection

Model Selection:

```r
library(selectnn)
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
## [ ... ]
```

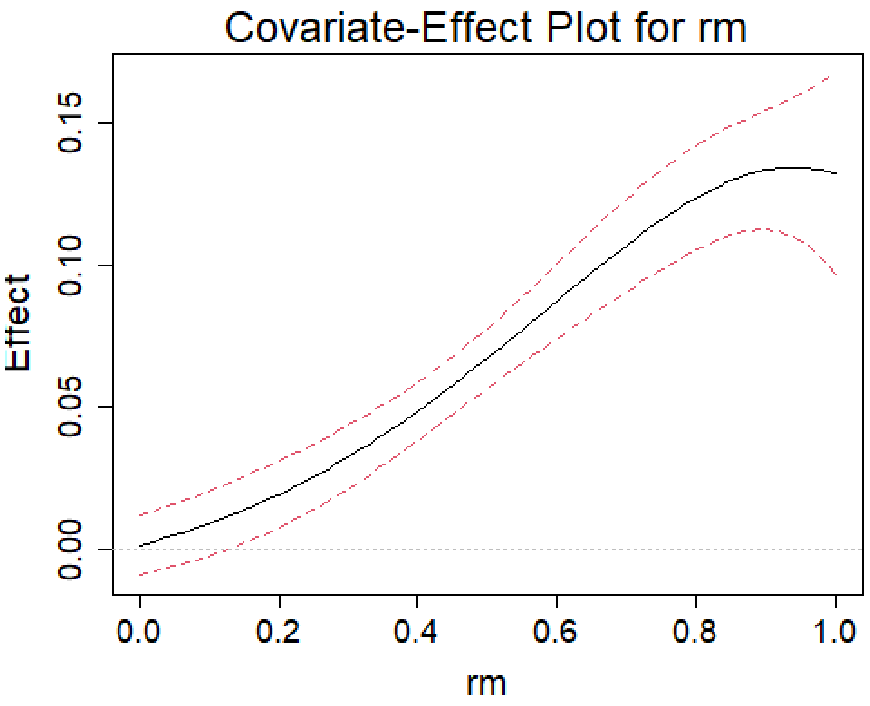
Boston Housing: Model Summary

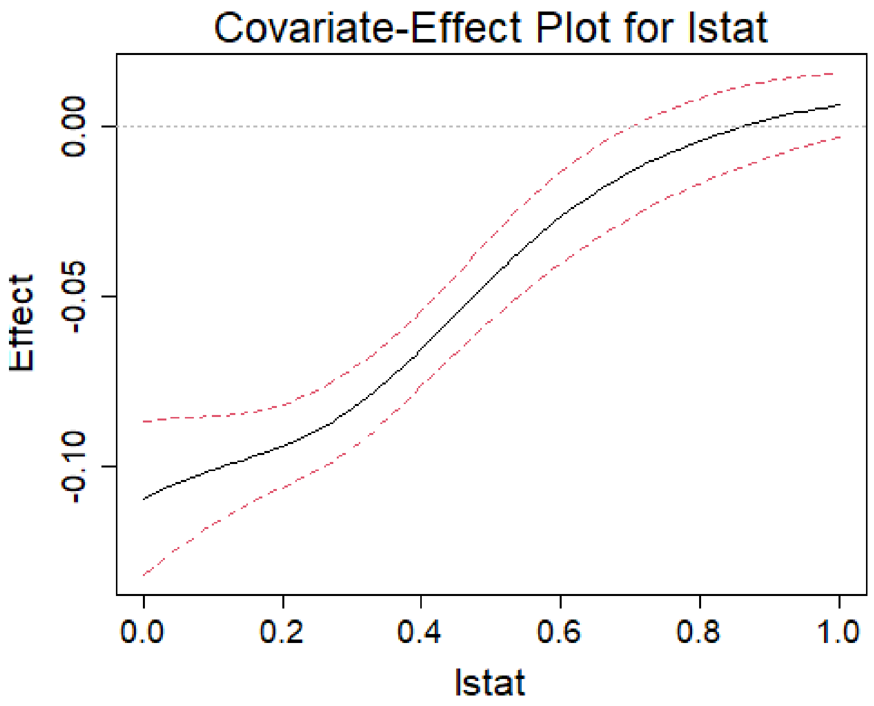
```r
library(statnnet)
stnn <- statnnet(nn)
summary(stnn)
```

```
## [ ... ]
## Coefficients:
##            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
##
## Weights:
## [ ... ]
```

Boston Housing: Plots

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





Summary & References

Summary

References

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

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2022). ISLR2: Introduction to Statistical Learning, Second Edition. R package version 1.3-1.