

Assignment 6.1: Module 6 Exercises

Zachariah Freitas

2022-12-03

ADS 506 Module 6 Exercises: Chapter 9

This assignment is due on Day 7 of the learning week. The assignment for this module is a mixture of programming and written work. Complete this entire assignment in R Markdown. You will need to include the question and number that you are answering within your submitted assignment. **Once completed, you will knit your deliverable to a Word/PDF file.**

Chapter 9: Neural Networks (Page 201): #2 & 3

Forecasting Australian Wine Sales: Figure 6.26 shows time plots of monthly sales of six types of Australian wines (red, rose, sweet white, dry white, sparkling, and fortified) for 1980-1994. Data available in AustralianWines.csv. The units are thousands of liters. You are hired to obtain short-term forecasts (2-3 months ahead) for each of the six series, and this task will be repeated every month.

```
library(fpp2)
library(zoo)
library(readr)
library(dplyr)
library(caret) # For the confusion Matrix

set.seed(506)

AustralianWines <- read_csv("data/AustralianWines.csv",
                           col_types = cols(Month = col_date(format = "%b-%y")))
```

2. Use neural networks to forecast fortified wine sales, as follows:

- Partition the data using the period until December 1993 as the training period.
- Run a neural network using R's `nnetar` with 11 non-seasonal lags (i.e., $p = 11$). Leave all other arguments at their default.

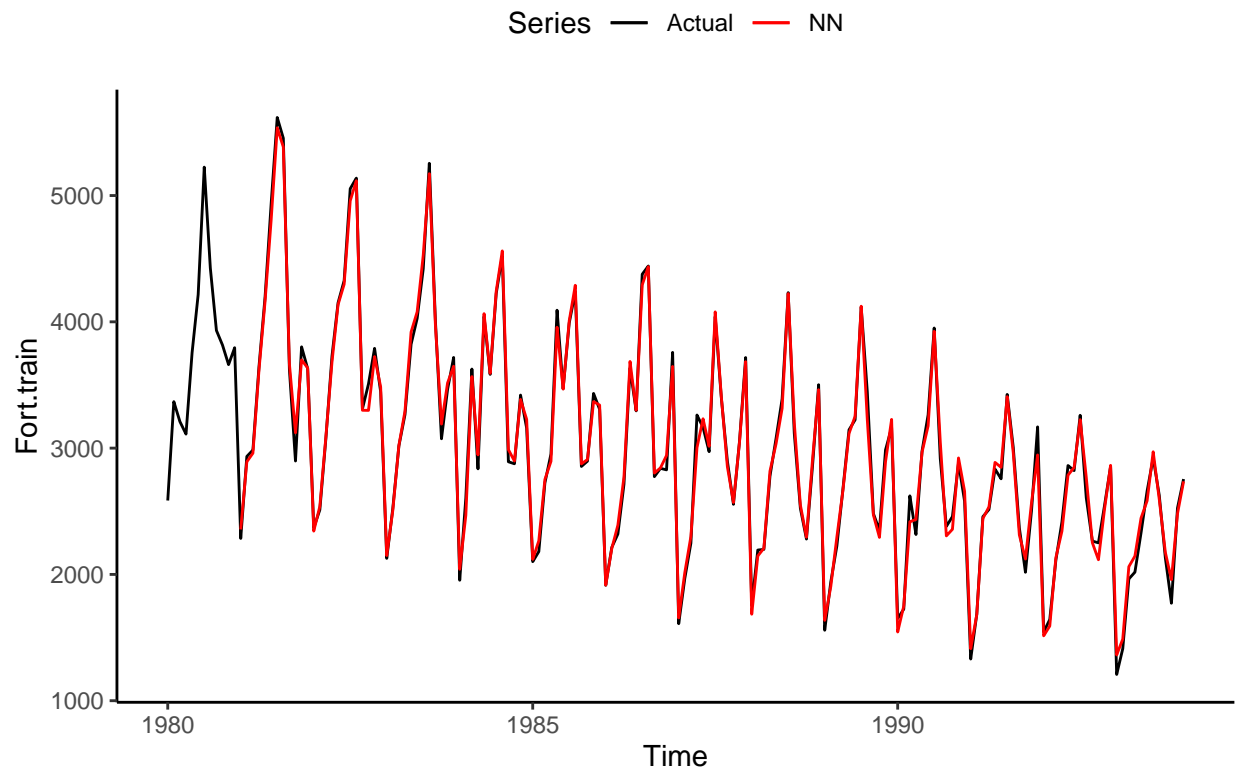
```
# Create Time Series Object
Fortified <- ts(AustralianWines$Fortified, start = c(1980, 1), frequency = 12)
# Training and test partitions
Fort.train <- window(Fortified, end = c(1993, 12))
Fort.test <- window(Fortified, start = c(1994, 1), end = c(1994, 12))

Fortified.nnetar <- nnetar(Fort.train, p = 11)
summary(Fortified.nnetar)
```

| ## | Length | Class | Mode |
|--------------|--------|--------------|-----------|
| ## x | 168 | ts | numeric |
| ## m | 1 | -none- | numeric |
| ## p | 1 | -none- | numeric |
| ## P | 1 | -none- | numeric |
| ## scalex | 2 | -none- | list |
| ## size | 1 | -none- | numeric |
| ## subset | 168 | -none- | numeric |
| ## model | 20 | nnetarmodels | list |
| ## nnetargs | 0 | -none- | list |
| ## fitted | 168 | ts | numeric |
| ## residuals | 168 | ts | numeric |
| ## lags | 12 | -none- | numeric |
| ## series | 1 | -none- | character |
| ## method | 1 | -none- | character |
| ## call | 3 | -none- | call |

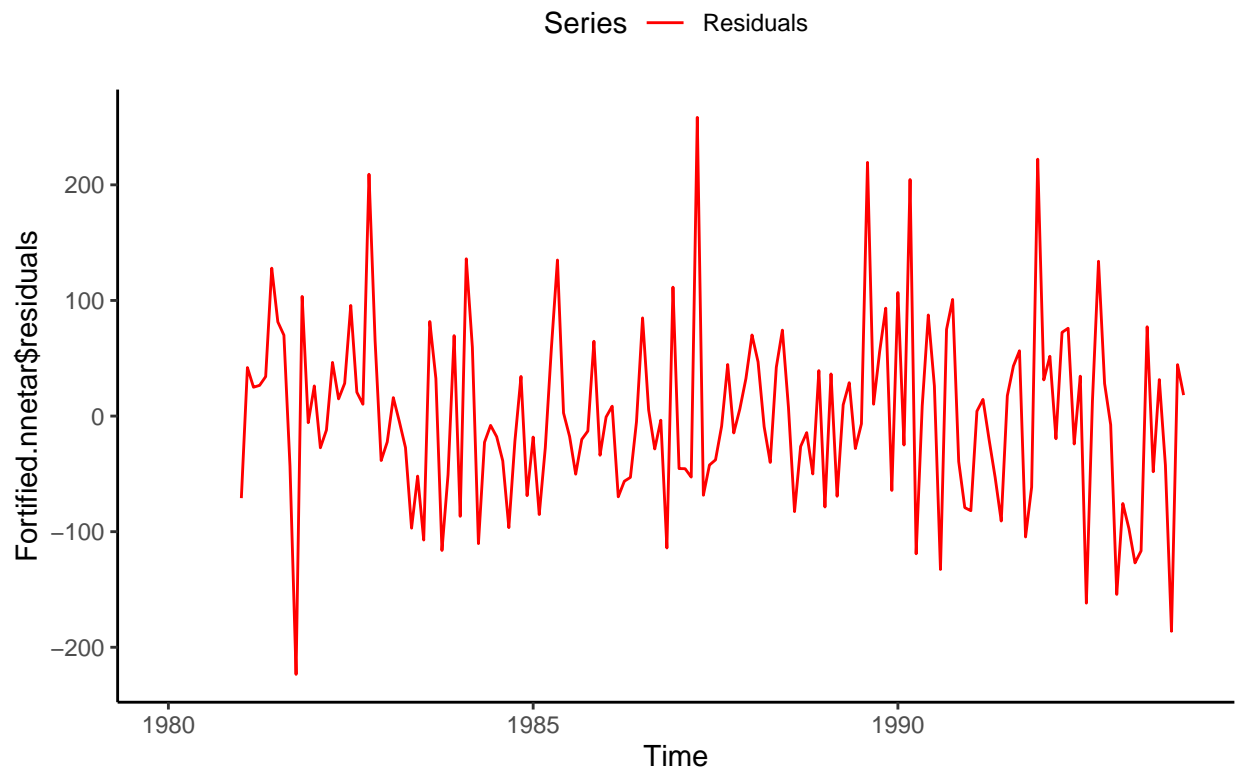
- a. Create a time plot for the actual and forecasted series over the training period. Create also a time plot of the forecast errors for the training period. Interpret what you see in the plots.

```
autoplot(Fort.train, series = "Actual") +  
  autolayer(Fortified.nnetar$fitted, series = "NN") +  
  guides(color = guide_legend(title = "Series")) +  
  scale_color_manual(values = c(Actual = "black", NN = "red")) +  
  theme_classic() +  
  # coord_cartesian(xlim = c(15, 18)) +  
  theme(legend.position = "top")
```



Create a time plot of the forecast errors for the training period. Interpret what you see in the plots.

```
autoplot(Fortified.nnetar$residuals, series = "Residuals") +  
  guides(color = guide_legend(title = "Series")) +  
  scale_color_manual(values = c(Residuals = "red")) +  
  theme_classic() +  
  # coord_cartesian(xlim = c(15, 18)) +  
  theme(legend.position = "top")
```



Answer: I see that the model does a pretty good job at predicting the downward demand trend, but tends to under predict peaks and tends to over predict bottoms. I would say that this model may be over-fitted.

- b. Use the neural network to forecast sales for each month in the validation period (January 1994 to December 1994).

```
Fortified.nnetar.pred <- forecast(Fortified.nnetar, h = 12)
Fortified.nnetar.pred
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1994 1381.386 1310.258 1902.544 2152.620 2172.317 2342.142 2546.089 2654.521
##           Sep      Oct      Nov      Dec
## 1994 2180.248 1817.884 2334.757 2776.608
```

```
accuracy(Fortified.nnetar.pred, Fort.test)
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.2410033  77.4552  59.00195 -0.2523812  2.19140  0.2120417
## Test set     118.8856332 294.1473 268.47137  4.2004839 12.19547  0.9648344
##           ACF1 Theil's U
## Training set -0.01151312      NA
## Test set     -0.26442258  0.6508992
```

Answer: The RMSE of train vs test suggests that this model is over-fitted.

3. Compare your neural network to an exponential smoothing model used to forecast fortified wine sales.
 - a. Use R's ets function to automatically select and fit an exponential smoothing model to the training period until December 1993. Which model did ets fit?

```
Fortified.ets <- ets(Fort.train, model="ZZZ")
summary(Fortified.ets)
```

```
## ETS(M,A,M)
##
## Call:
## ets(y = Fort.train, model = "ZZZ")
##
## Smoothing parameters:
##   alpha = 0.0243
##   beta  = 0.0011
##   gamma = 1e-04
##
## Initial states:
##   l = 4049.4027
##   b = -12.1017
##   s = 1.1369 1.0373 0.8857 0.9522 1.2762 1.3697
##       1.1443 1.1121 0.9335 0.8626 0.7041 0.5855
##
## sigma: 0.091
##
##      AIC      AICc      BIC
## 2753.520 2757.600 2806.628
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 15.49844 285.2321 222.3198 -0.04620971 7.098462 0.7989745
##              ACF1
## Training set 0.06838359
```

Answer: ets function fitted an M,A,M model. Multiplicative, Additive, and Multiplicative. The parameter estimates are $\hat{\alpha} = 0.0243$, $\hat{\beta} = 0.0011$, and $\hat{\gamma} = 0.0243$.

b. Use this exponential smoothing model to forecast sales for each month in 1994.

```
Fortified.ets.pred <- forecast(Fortified.ets, h = 12)
```

```
accuracy(Fortified.ets.pred, Fort.test)
```

```
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 15.49844 285.2321 222.3198 -0.04620971  7.098462 0.7989745
## Test set    98.19161 318.9037 242.6523  3.22878698 10.237376 0.8720456
##              ACF1 Theil's U
## Training set  0.06838359      NA
## Test set     -0.04043727 0.6850152
```

Answer: See code above.

- c. How does the neural network compare to the exponential smoothing model in terms of predictive performance in the training period? In the validation period?

```
print("Neural Network Model")
```

```
## [1] "Neural Network Model"
```

```
accuracy(Fortified.nnetar.pred, Fort.test)
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.2410033  77.4552  59.00195 -0.2523812  2.19140  0.2120417
## Test set     118.8856332 294.1473 268.47137  4.2004839 12.19547  0.9648344
##           ACF1 Theil's U
## Training set -0.01151312      NA
## Test set     -0.26442258 0.6508992
```

```
print("Exponential Smoothing Model")
```

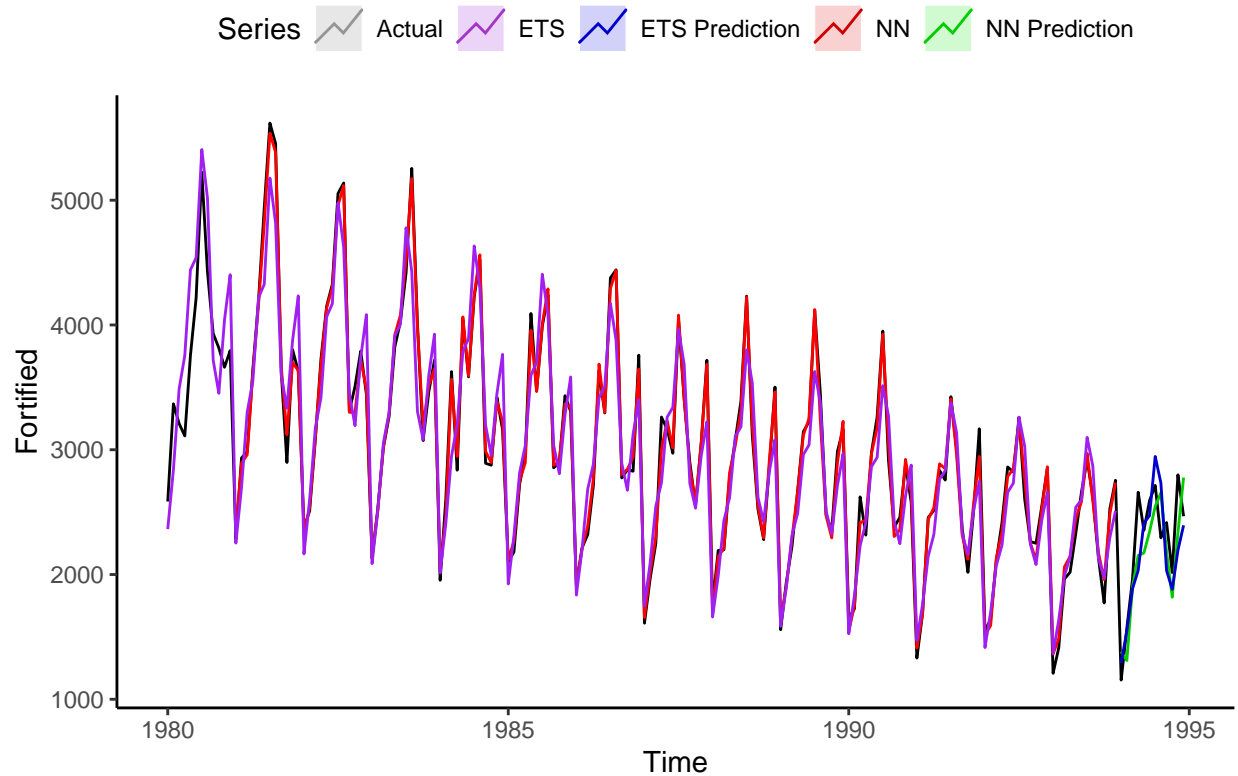
```
## [1] "Exponential Smoothing Model"
```

```
accuracy(Fortified.ets.pred, Fort.test)
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 15.49844 285.2321 222.3198 -0.04620971  7.098462  0.7989745
## Test set     98.19161 318.9037 242.6523  3.22878698 10.237376  0.8720456
##           ACF1 Theil's U
## Training set  0.06838359      NA
## Test set     -0.04043727 0.6850152
```

```
autoplot(Fortified, series = "Actual") +
  autolayer(Fortified.nnetar$fitted, series = "NN") +
  autolayer(Fortified.ets$fitted, series = "ETS") +
  autolayer(Fortified.nnetar.pred, series = "NN Prediction") +
  autolayer(Fortified.ets.pred, series = "ETS Prediction", PI = F) +
  guides(color = guide_legend(title = "Series")) +
  scale_color_manual(values = c(Actual = "black", NN = "red",
                                `NN Prediction` = "green",
                                ETS = "purple",
                                `ETS Prediction` = "blue")) +

  theme_classic() +
  # coord_cartesian(xlim = c(15, 18)) +
  theme(legend.position = "top")
```

Answer: Comparing the predictive performance in the training period, the neural network model vastly out performed the exponential smoothing model with RMSE of 74 vs 222 respectively. Almost a 74% improvement over the ETS model.

Comparing the predictive performance in the validation period, the neural network model out performed the exponential smoothing model with RMSE of 252 vs 318 respectively. Almost a 21% improvement over the ETS model.