Прогноз временных рядов с помощью ансамбля нейронных сетей в R: пакет nnfor

Введение

Пакет nnfor для R упрощает прогнозирование временных рядов с помощью многослойных персептронов (MLP) и Extreme Learning Machines (ELM). В настоящее время он использует пакет нейронной сети для R, который предоставляет все механизмы для обучения MLP. Обучение ELM реализовано непосредствнно в пакете nnfor .

Известно, что большие сети, как правило, очень медленно обучаются. nnfor отличается от существующих реализаций нейронных сетей для R тем, что предоставляет код для автоматического проектирования сетей с разумной производительностью прогнозирования, но также обеспечивает всесторонний контроль для опытного пользователя. Автоматическая спецификация разработана с учетом соображений экономии. Это увеличивает надежность получаемых сетей, но также помогает сократить время обучения.

Прогнозирование с помощью MLP

С пакетом nnfor можно производить экстраполяционный (одномерный) прогноз или также включать независимые переменные.

Одномерное прогнозирование

Основная функция mlp(), и в ее простейшей форме нужно только ввести временной ряд для моделирования ВР. Здесь GAZP.win нормализованный ВР цен акций Газпрома с 1 января 2019 года по 1 декабря 2020 года.

```
library(nnfor)
fit1 <- mlp(GAZP.win)
print(fit1)
```

MLP fit with 5 hidden nodes and 20 repetitions. Series modelled in differences: D1. Univariate lags: (1) Forecast combined using the median operator. MSE: 9.3646.

Выходные данные показывают, что полученная сеть имеет 5 скрытых узлов, она была обучена 20 раз, а различные прогнозы были объединены с использованием медианного оператора. mlp(). Функция автоматически генерирует ансамбли сетей, обучение которых начинается с различными случайными начальными весами. Анализ осуществленный рядом экспертов показал,что в качестве оператора комбинации медиана работает

очень хорошо. Она обеспечивает лучшую производительность, но требует несколько больших ансамблей, а среднее арифметическое, вероятно, лучше всего избегать!

Кроме того авторам пакета установлено, что обучение сети на загрузочных сериях или использование нескольких случайных инициализаций обучения приводит к аналогичной производительности, и поэтому для прогнозирования временных рядов можно избежать этапа начальной загрузки, что значительно упрощает процесс. Эти результаты встроены в настройки по умолчанию nnfor .

Можно получить визуальную сводку, используя plot(fit1) функцию:

Серые входные узлы представляют собой авторегрессию (в нашем лучае это одна авторегрессия), а пурпурные (их у нас нет) - детерминированные входные данные (например, сезонность). Если бы были включены какие-либо другие регрессоры, они были бы показаны голубым цветом.

Функция mlp() принимает несколько аргументов , чтобы точно настроить результирующую сеть. Аргумент hd определяет фиксированное число скрытых узлов. Если это одно число, то нейроны расположены в одном скрытом узле. Если это вектор, то они расположены в несколько слоев. Обычные нейронные сети, прогнозирующие отдельные временные ряды, не выигрывают от нескольких скрытых слоев. Проблема прогнозирования обычно не такая сложная!

```
library(nnfor)
fit2 <- mlp(GAZP.win,hd = c(5,2) )
print(fit2)
plot(fit2)
```

MLP fit with (5,2) hidden nodes and 20 repetitions. Series modelled in differences: D1. Univariate lags: (1) Forecast combined using the median operator. MSE: 9.3689.

Из сводки видно, что„ введя дополниттельный скрытый слой, мы даже немного проиграли в точности модели.

Аргумент reps определяет, сколько тренировочных повторений используется. Если вы хотите обучить единственную сеть, вы можете использовать ее reps=1, хотя есть неопровержимые доказательства того, что это бесполезно. По умолчанию reps=20 это компромисс между скоростью тренировки и производительностью, но чем больше повторений вы

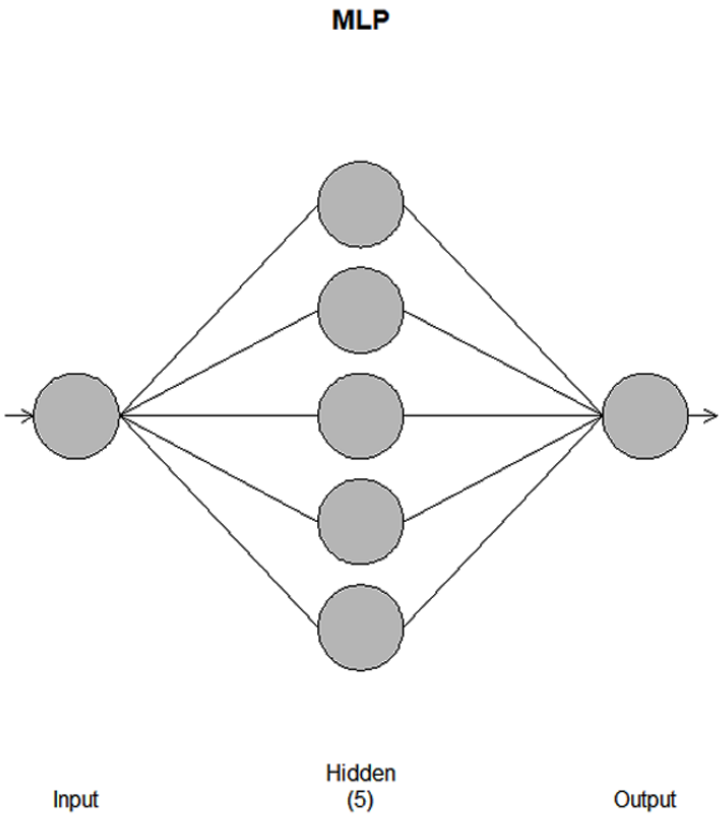

Рис. 1: Структура нейронной сети, построеной nnfor

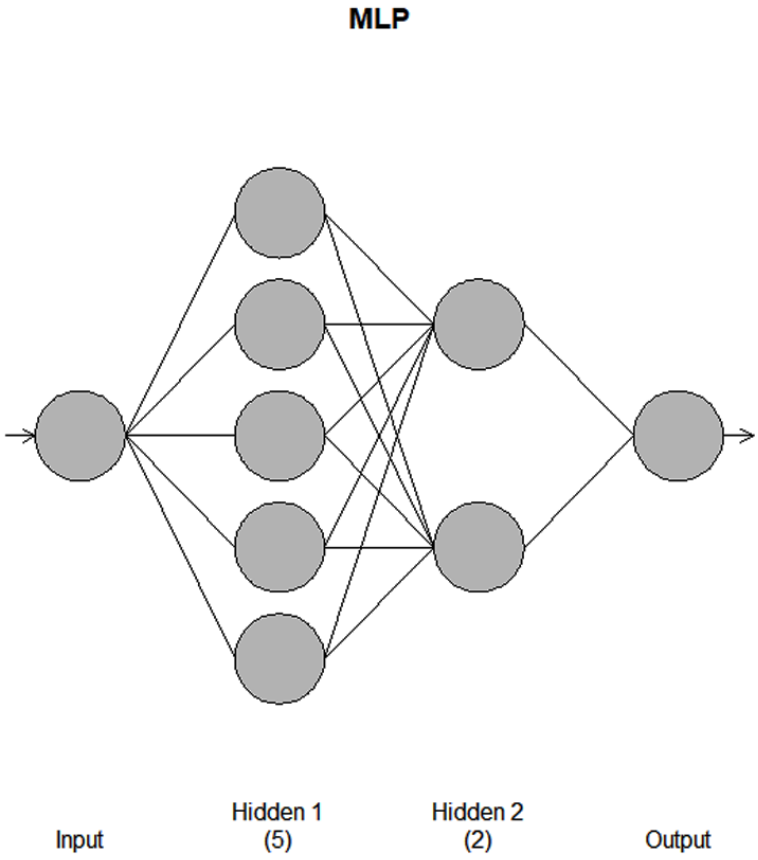

Рис. 2: Структура нейронной сети, построеной nnfor при hd = c(5,2)

можете себе позволить, тем лучше. Они помогают не только в производительности модели, но и в стабильности результатов при переобучении сети.

Как различные учебные повторы объединены управляется аргументом comb, который принимает параметры median, mean и mode. Среднее значение и медиана очевидны. Режим рассчитывается с использованием максимума оценки плотности ядра прогнозов для каждого периода времени.

Аргумент lags позволяет выбрать авторегрессионные задержки, учитываемые сетью. Если это не предусмотрено, тогда сеть использует лаг 1. Это предлагаемые лаги, которые могут не оставаться в окончательных сетях. Вы можете принудительно использовать это с помощью аргумента keep или полностью отключить автоматический выбор ввода с помощью аргумента sel.lag=FALSE.

```
library(nnfor)
fit2 <- mlp(GAZP.win,hd = c(5,2), ,reps=25, lags=1:24)
plot(fit2)
```

MLP fit with (5,2) hidden nodes and 25 repetitions. Series modelled in differences: D1. Univariate lags: (2,13,14,24) Forecast combined using the median operator. MSE: 9.0079.

```
library(nnfor)
fit2 <- mlp(GAZP.win,hd = c(5,2), ,reps=25, lags=1:24, sel.lag=FALSE)
plot(fit2)
```

MLP fit with (5,2) hidden nodes and 25 repetitions. Series modelled in differences: D1. Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,

16,17,18,19,20,21,22,23,24) Forecast combined using the median operator. MSE: 4.5847.

Нейронные сети не очень хороши в моделировании трендов. Поэтому полезно удалить тренд из временного ряда перед его моделированием. Это обрабатывается аргументом difforder. Если difforder=0 никакое различие не проводится. Для diff=1 выполняются разности уровней. Аналогично, если difforder=12 выполняется разность 12-го порядка. Если временной ряд является сезонным с сезонным периодом 12, тогда это будут сезонные различия. Вы можете использовать оба difforder=c(1,12)

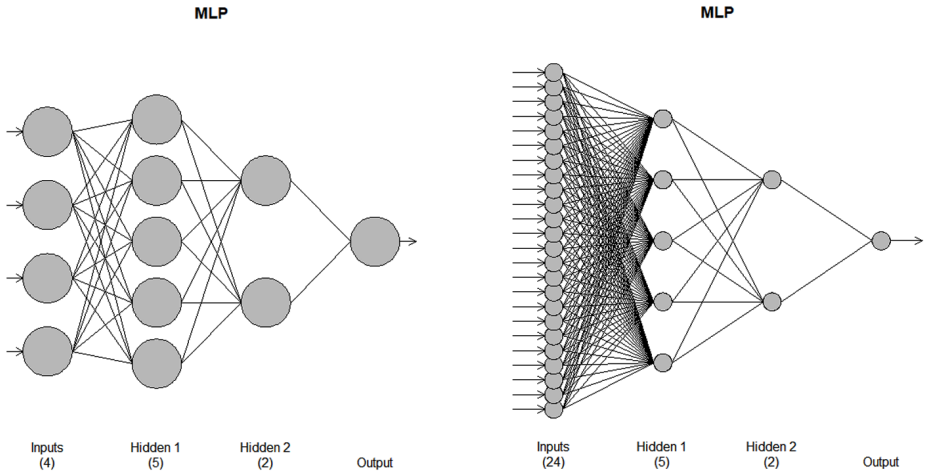

Рис. 3: Структура нейронной сети, построеной nnfor при sel.lag=TRUE и sel.lag=FALSE)

или любой другой набор разностных ордеров. Если difforder=NULL тогда код решает автоматически. Если есть тренд, используются первые разности. Сериал также тестируется на сезонность. Если да, то используется тест Кановы-Хансена, чтобы определить, является ли он детерминированным или стохастическим. Если последнее, то также добавляются сезонные различия.

Детерминированную сезонность лучше моделировать с использованием сезонных фиктивных переменных. Это можно контролировать с помощью логического аргумента allow.det.season. Детерминированная сезонность может быть либо набором двоичных фиктивных переменных, либо парой синус-косинусов (аргумент det.type). Если сезонный период больше 12, то для экономии рекомендуется тригонометрическое представление. Логический аргумент outplot дает график соответствия сети BP.

```
library(nnfor)
fit2 <- mlp(GAZP.win,hd = c(5,2), reps=25, lags=1:24, outplot=TRUE)
```

Количество скрытых узлов может быть либо предварительно установлено с помощью аргумента, hd либо указано автоматически, как опре-

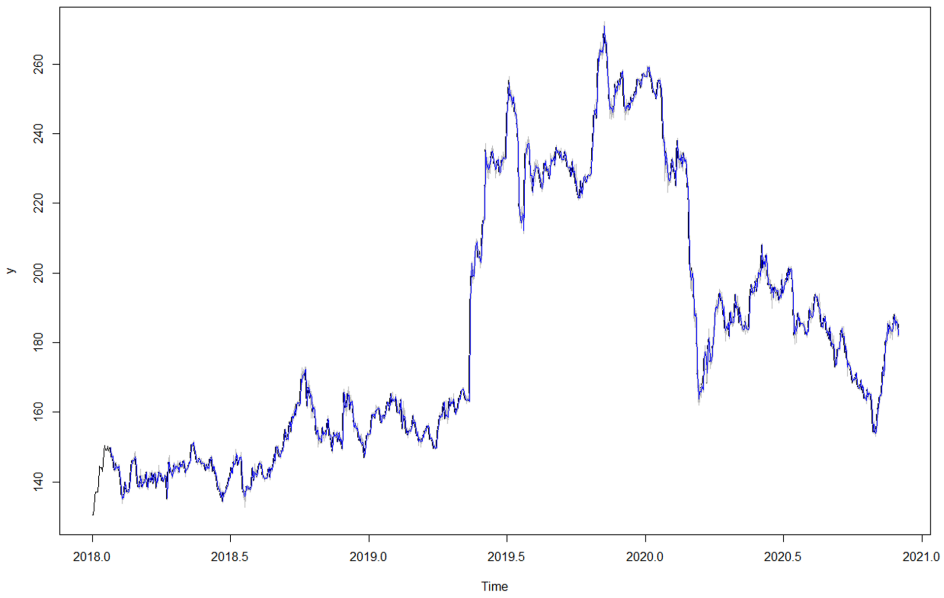

Рис. 4: График соотвествия сети BP цен акций Газпрома 2018-01-01, 2020-12-01

делено с помощью аргумента hd.auto.type.

По умолчанию это hd.auto.type="set"и использует ввод, предоставленный в hd (по умолчанию 5).

Вы можете установить это значение hd.auto.type="valid"для тестирования с использованием выборки проверки (20 % временного ряда) или hd.auto.type="cv"для использования 5-кратной перекрестной проверки. Количество скрытых узлов для оценки устанавливается аргументом hd.max.

```
library(nnfor)
fit2 <- mlp(GAZP.win, hd.auto.type="valid", hd.max=8)
print(fit2)
```

MLP fit with 5 hidden nodes and 20 repetitions. Series modelled in differences: D1. Univariate lags: (1) Forecast combined using the median operator. MSE: 9.3619.

Учитывая, что обучение сетей может занять много времени, вы можете повторно использовать уже указанную / обученную сеть. В следующем примере мы повторно fit2 используем новый временной ряд.

```
library(nnfor)
fit2 <- mlp(X, model=fit2 )
```

Это сохраняет как спецификацию, так и обучение из fit2. Если вы хотите использовать только спецификацию, но переобучить сеть, то используйте аргумент retrain=TRUE.

Для составления прогнозов используется функция forecast(), для которой требуется обученный сетевой объект и горизонт прогноза h.

```
frc <- forecast(fit2,h=12)
print(frc$mean)
```

Time Series:
Start = c(2020, 336)
End = c(2020, 360)
Frequency = 365
t+1 t+2 t+3 t+4 t+5 t+6 t+7 t+8 t+9
185.02 185.36 185.37 185.45 185.49 185.54 185.58 185.63 185.67
t+10 t+11 t+12 t+13 t+14 t+15 t+16 t+17 t+18

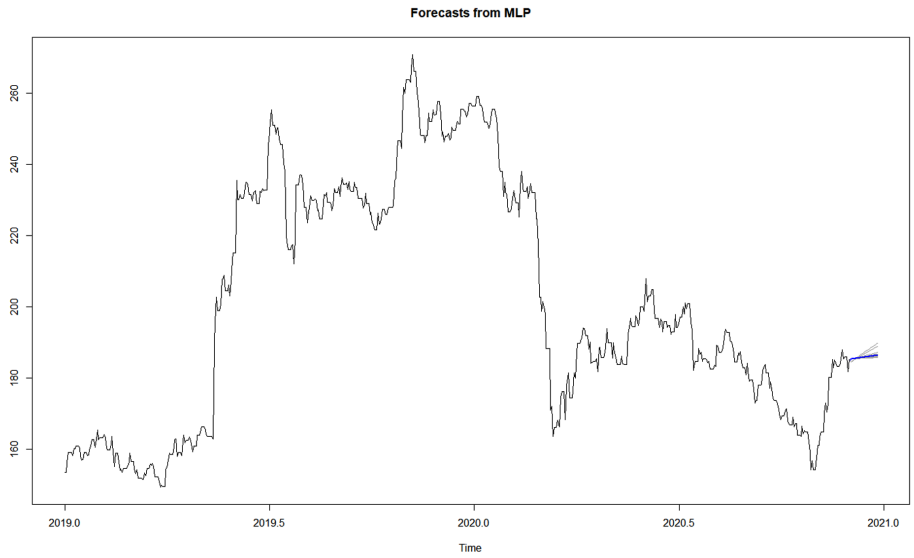

Рис. 5: График BP цен акций Газпрома 2019-01-01, 2020-12-01 и прогноз на 25 дней

185.72 185.76 185.81 185.85 185.90 185.94 185.99 186.03 186.08
t+19 t+20 t+21 t+22 t+23 t+24 t+25
186.12 186.17 186.21 186.26 186.30 186.35 186.40

```
plot(frc)
```

На графике прогнозов серым цветом представлены прогнозы всех участников ансамбля. Результат forecast() является классом forecast. Для доступа к точечным прогнозам используется frc$mean, а frc$all.mean содержит прогнозы отдельных членов ансамбля.

Использование регрессоров

В mlp() функции есть три аргумента , которые позволяют использовать объясняющие переменные: xreg, xreg.lags и xreg.keep. Первый используется для ввода дополнительных регрессоров. Они должны быть организованы в виде массива и быть равной длине временного ряда в выборке плюс горизонт прогнозирования, хотя могут быть и более длинными. Предположим, что мы хотим использовать в качестве регрессора для прогнозирования временного ряда акций Газпрома с 2018-01-01 по

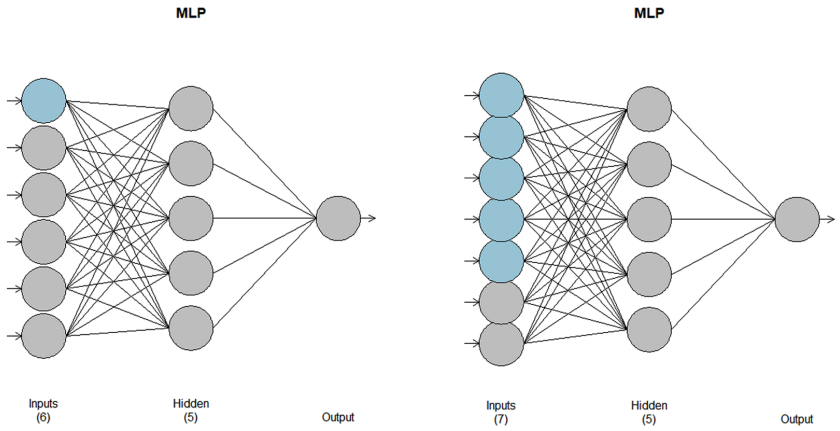

Рис. 6: Структура сети с регрессором (отмечен голубым цветом) при xreg.lags=list(0) /list(1:5)

2020-12-01 BP цен нефти Brent с 2018-01-01 по 2020-12-10. Сначала мы создаем входные данные. Обязательно, чтобы регрессор имел class "matrix array"После создания данных моделируем серию.

```
Brent_z <-subset(Brend_xx3,t >= "2018-01-01")
z <-Brent_z[,2]
z <- cbind(z)
fit2 <- mlp(GAZP.win,reps=25,lags =1:22,xreg=z,
xreg.lags=list(0),xreg.keep=list(TRUE),outplot=TRUE)
print(fit2)
plot(fit2)
```

MLP fit with 5 hidden nodes and 25 repetitions. Series modelled in differences: D1. Univariate lags: (2,4,7,9,14) 1 regressor included. - Regressor 1 lags: (0) Forecast combined using the median operator. MSE: 6.3952.

В модели используется один регрессор, но можно включить любое количество регрессоров.

Для составления прогнозов используем forecast() функцию, но теперь используем xreg ввод. Способ выполнить эту работу - ввести регрессоры, начиная с того же наблюдения, которое использовалось во время обучения сети, расширенных по мере необходимости для покрытия горизонта

прогноза. Не нужно удалять неиспользуемые регрессоры. Об этом позаботится сеть.

```
frc.reg <- forecast(fit2,xreg=z, h=7)
```

Time Series:
Start = c(2020, 336)
End = c(2020, 342)
Frequency = 365
t+1 t+2 t+3 t+4 t+5 t+6
185.1302 184.5792 184.2239 184.6864 184.7219 184.7174
t+7
184.7047

В приложении преведена полная сводка параметров функции mlp ().

Прогнозирование с помощью ELM

Чтобы использовать Extreme Learning Machines (EML), ye;yj ghbvtybnm функцию elm(). Многие входные данные идентичны mlp(). По умолчанию ELM начинаются с очень большого скрытого слоя (100 узлов), который при необходимости удаляется.

На графике сети есть несколько черных и несколько серых линий. Последние обрезаются. Установлено 20 сетей (контролируемых аргументом reps). Каждая сеть может иметь разные конечные соединения. Можно проверить их, используя plot(fit3,1), где второй аргумент определяет, какую сеть строить.

```
fit3 <- elm(GAZP.win,lags =1:21,sel.lag=FALSE)
print(fit3)
plot(fit3)
```

ELM fit with 100 hidden nodes and 20 repetitions. Series modelled in differences: D1. Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21) Forecast combined using the median operator. Output weight estimation using: lasso. MSE: 9.6171.

```
par(mfrow=c(2,2))
for (i in 1:4){plot(fit3,i)}
par(mfrow=c(1,1))
```

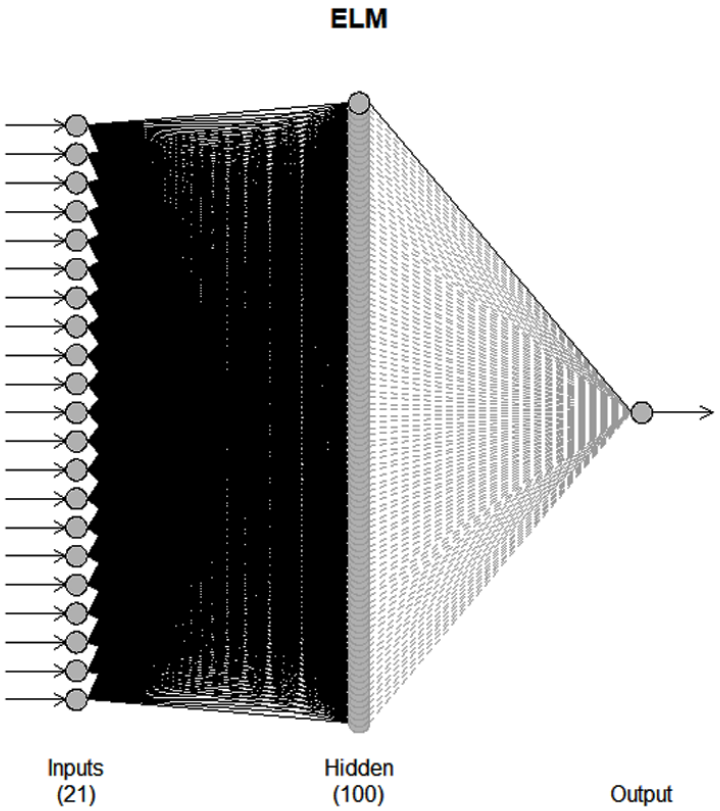

Рис. 7: Структура сети- результат работы функции elm

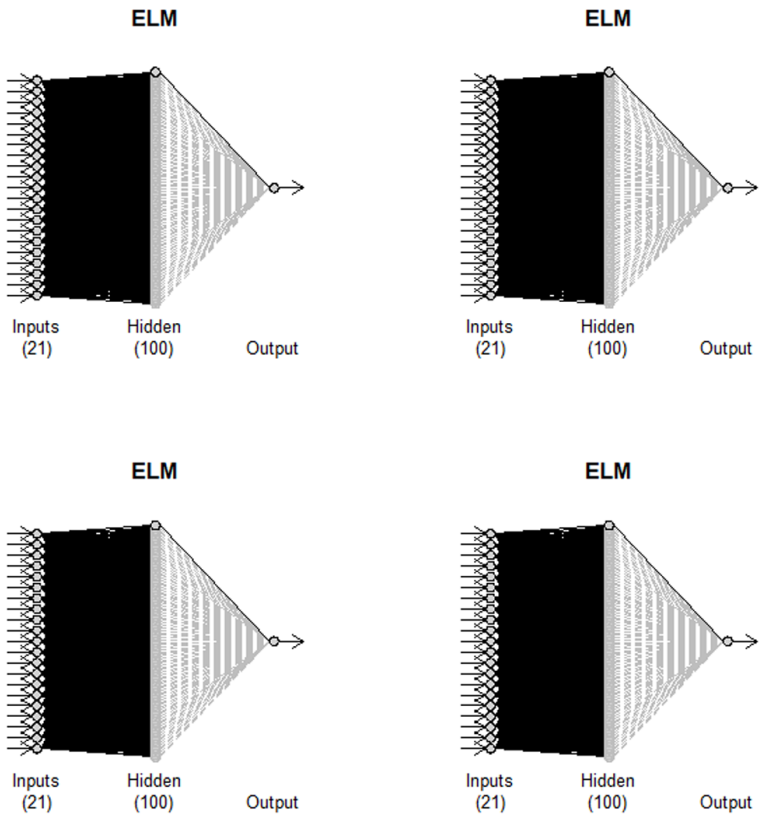

Рис. 8: Варианты соединений сетей

Как выполняется обрезка, контролируется аргументом. type По умолчанию используется регрессия LASSO (type = «lasso»). В качестве альтернативы можно использовать «гребешок» для регрессии гребня, «шаг» для пошаговой OLS и «lm» для получения решения OLS без отсечения.

Другое отличие от mlp() функции - barebone аргумент. Когда это FALSE, тогда eml() строятся на основе пакета нейронной сети . Если установлено значение, TRUE используется другая внутренняя реализация, которая помогает ускорить вычисления при значительном количестве входов.

Для прогноза используйте forecast() функцию так же, как и раньше.

```
frc <- forecast(fit3,h=25)
print(frc$mean)
```

Time Series:
Start = c(2020, 336)
End = c(2020, 360)
Frequency = 365
t+1 t+2 t+3 t+4 t+5 t+6 t+7 t+8
185.3796 185.4192 185.4588 185.4983 185.5379 185.5775 185.6171 185.6567
t+9 t+10 t+11 t+12 t+13 t+14 t+15 t+16
185.6963 185.7359 185.7755 185.8150 185.8546 185.8942 185.9338 185.9734
t+17 t+18 t+19 t+20 t+21 t+22 t+23 t+24
186.0130 186.0526 186.0922 186.1317 186.1713 186.2109 186.2505 186.2901
t+25
186.3297

По ссылке можно загрузить архив, который содержит файлы скриптов и данных для осуществления прогноза цен акций Газпрома со 2 декабря 2020 года на 25 дней с использованием следующей mlp() функции

```
fit1 <- mlp(GAZP.win,hd = c(8,4,2),reps=55,lags=1:15,outplot=TRUE)
```

Этом же архиве находятся графические результаты работы этой модели

Приложение функция mlp()

Функция mlp многоуровневый перцептрон для прогнозирования временных рядов

Описание

```
mlp(y, m = frequency(y), hd = NULL, reps = 20, comb = c("median",

"mean", "mode"), lags = NULL, keep = NULL, difforder = NULL,

outplot = c(FALSE, TRUE), sel.lag = c(TRUE, FALSE),

allow.det.season = c(TRUE, FALSE), det.type = c("auto", "bin",

"trg"), xreg = NULL, xreg.lags = NULL, xreg.keep = NULL,

hd.auto.type = c("set", "valid", "cv", "elm"), hd.max = NULL,

model = NULL, retrain = c(FALSE, TRUE), ...)
```

АРГУМЕНТЫ

y Входной временной ряд. Может быть объектом ts или msts

m Частота временного ряда. По умолчанию забирается из y

hd Количество скрытых узлов. Это может быть вектор, где каждое число представляет количество скрытых узлов другого скрытого слоя.

reps Количество сетей для обучения, результат - ансамблевый прогноз

comb Комбинированный оператор для прогнозов, когда количество повторений > 1. Может быть «median», «режимом» (на основе оценки KDE) и «mean».

lags Лаги y для использования в качестве входных данных. Если ничего не указано, используется 1: частота (y). Используйте 0, чтобы не допускать одномерных задержек.

keep Логический вектор, заставляющий лаги оставаться в модели, если sel.lag == TRUE. Если NULL, то keep = rep (FALSE, length (lags)).

difforder Вектор, включая разностные запаздывания. Например, c (1,12) применит первую и сезонную (12) разницы. Для неразличения используйте 0. Для автоматического выбора используйте NULL.

outplot Предоставьте график соответствия модели. Может иметь значение TRUE или FALSE.

sel.lag Автоматически выбирать лаги. Может иметь значение TRUE или FALSE. allow.det.season Разрешить моделирование сезонности с помощью детерминированных фиктивных переменных.

det.type Тип используемых манекенов детерминированной сезонности. Это может быть «bin» для двоичной пары синус-косинус. С «auto», если используется только одна сезонность и периодичность до 12, то используется «bin», в противном случае - «trg».
xreg Экзогенные регрессоры. Каждый столбец представляет собой отдельный регрессор, и размер выборки должен быть не меньше целевого набора в выборке, но может быть больше
xreg.lags Это список, содержащий лагы для каждой экзогенной переменной. Каждый список представляет собой числовой вектор, содержащий лаги. Если xreg имеет 3 столбца, то список xreg.lags должен содержать три элемента. Если NULL, то он автоматически указывается
xreg.keep Список логических векторов, заставляющих лагы xreg оставаться в модели, если sel.lag == TRUE. Если NULL, то все внешние лаги могут быть удалены. Синтаксис для нескольких xreg такой же, как для xreg.lags.
hd.auto.type Используется только если hd == NULL. "set" фиксирует hd = 5. «Действительный» использует 20% проверочный набор (случайным образом), выбранный для поиска наилучшего количества скрытых узлов "cv"использует пятикратную перекрестную проверку. «elm» использует ELM для оценки количества скрытых узлов (экспериментально)
hd.max Если для hd.auto.type установлено значение «valid» или «cv», то этот аргумент может использоваться для установки максимального количества скрытых узлов для оценки, в противном случае максимальное значение устанавливается автоматически
model Ранее обученный объект mlp. Если это предусмотрено, то та же модель
аппроксимируется по y без повторной оценки каких-либо параметров модели.
retrain Если предусмотрена предыдущая модель, переобучать сеть или нет.
... Дополнительные входы для функции нейронной сети.
Мой сайт
Архив скриптов, данных и графиков
ВЕРСТКА LATEX 08.12.2020