Part 2 MLP

Shaaf Ahmad W17906977

2022-04-29

## Discussion of the various methods used for defining the input vector

Time series is a dataset in which one or more variables are measured over time. We will be doing a One-step forecast using the Autoregression (AR) model. AR is a linear model which predicts a variables future value using its past (lag) values.

However there are other models which can be used in time series forecasting.

#### Moving Average (MA)

Moving average models use lagged values of forecast errors. The moving average model has use multiple steps back. MA(3) model has an order of 3 and uses 3 time step back.

#### Autoregressive moving average (ARMA)

ARMA models combine both Autoregression and Moving average into one model. ARMA uses both values of past forecast errors and past lag values. This model is often known as ARMA(p,q) model where p is p lagged values and q is the residuals up to q lags.

#### Autoregressive integrated moving average (ARIMA)

ARIMA(p,d,q)

#### Seasonal autoregressive integrated moving-average (SARIMA)

library(readxl)  
library(neuralnet)  
library(grid)  
library(MASS)  
library("remotes")  
library(MLmetrics)

##   
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':  
##   
## Recall

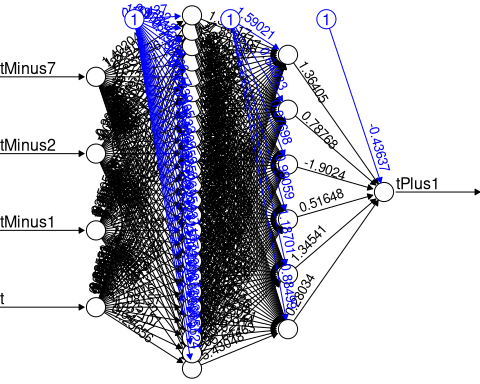
UoW\_energy<-read\_excel("UoW\_load.xlsx", sheet = "Sheet1")  
twoInput<-read\_excel("UoW\_load.xlsx", sheet = "Sheet2")  
threeInput<-read\_excel("UoW\_load.xlsx", sheet = "Sheet3")  
fourInput<-read\_excel("UoW\_load.xlsx", sheet = "Sheet4")  
  
fourInput2<-read\_excel("UoW\_load.xlsx", sheet = "Sheet5")

normalize <- function(x) {  
 return((x - min(x)) / (max(x) - min(x)))  
}  
  
  
  
threeInput\_norm<-as.data.frame(lapply(threeInput,normalize))  
  
twoInput\_norm<-as.data.frame(lapply(twoInput,normalize))  
  
fourInput\_norm<-as.data.frame(lapply(fourInput,normalize))  
  
fourInput2\_norm<-as.data.frame(lapply(fourInput2,normalize))

threeInput\_train<-threeInput\_norm[1:427,]  
threeInput\_test<-threeInput\_norm[428:497,]  
  
twoInput\_train<-twoInput\_norm[1:428,]  
twoInput\_test<-twoInput\_norm[429:498,]  
  
  
fourInput\_train<-fourInput\_norm[1:426,]  
fourInput\_test<-fourInput\_norm[427:496,]  
  
fourInput2\_train<-fourInput2\_norm[1:422,]  
fourInput2\_test<-fourInput2\_norm[423:492,]

### 4 input2 h1 24 h2 6

set.seed(12345)  
fourInput2Model.h1.24.h2.6.TRUE.nn<-neuralnet(tPlus1~tMinus7+tMinus2+tMinus1+t,data = fourInput2\_train,hidden = c(24,6),act.fct = "logistic",linear.output = TRUE)  
  
plot(fourInput2Model.h1.24.h2.6.TRUE.nn, rep = "best")



fourInput2Model.h1.24.h2.6.TRUE.nn\_Result<-compute(fourInput2Model.h1.24.h2.6.TRUE.nn,fourInput2\_test[1:4])  
  
  
predicted\_tomorrow<-fourInput2Model.h1.24.h2.6.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput2\_test$tPlus1)

## [,1]  
## [1,] 0.8733008

fourInput2\_train<-fourInput2\_norm[1:422,]  
fourInput2\_test<-fourInput2\_norm[423:492,]  
  
fourInput2\_train\_original<-as.data.frame(fourInput2[1:422,"tPlus1"])  
  
  
fourInput2\_test\_original<-as.data.frame(fourInput2[423:492,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput2\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput2\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput2\_train\_original)

## tPlus1  
## 1 116.0  
## 2 114.2  
## 3 113.4  
## 4 120.4  
## 5 84.8  
## 6 66.8

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 423 100.92438  
## 424 100.93163  
## 425 96.13211  
## 426 64.57237  
## 427 113.83882  
## 428 111.27003  
## 429 113.99119  
## 430 115.42845  
## 431 110.27030  
## 432 103.93752  
## 433 64.16877  
## 434 122.72588  
## 435 137.26390  
## 436 125.91301  
## 437 115.91984  
## 438 110.59192  
## 439 112.50681  
## 440 66.69581  
## 441 123.28300  
## 442 131.17609  
## 443 129.08493  
## 444 107.20832  
## 445 114.18972  
## 446 103.89696  
## 447 82.68535  
## 448 121.45342  
## 449 133.44543  
## 450 136.14795  
## 451 116.40142  
## 452 126.26396  
## 453 110.19704  
## 454 67.48895  
## 455 119.67309  
## 456 141.33556  
## 457 134.26174  
## 458 121.24340  
## 459 120.98519  
## 460 115.34431  
## 461 68.56041  
## 462 124.12409  
## 463 98.75572  
## 464 121.10599  
## 465 101.17239  
## 466 99.39194  
## 467 90.00363  
## 468 68.62255  
## 469 95.86246  
## 470 111.14203  
## 471 110.87653  
## 472 92.46929  
## 473 89.34287  
## 474 78.41452  
## 475 65.71949  
## 476 105.28795  
## 477 110.53851  
## 478 118.78981  
## 479 95.89865  
## 480 93.07443  
## 481 94.29146  
## 482 65.79684  
## 483 117.66557  
## 484 100.60791  
## 485 93.08098  
## 486 108.19023  
## 487 98.30344  
## 488 89.15382  
## 489 64.83492  
## 490 106.38136  
## 491 115.80919  
## 492 100.53333

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 12.81393

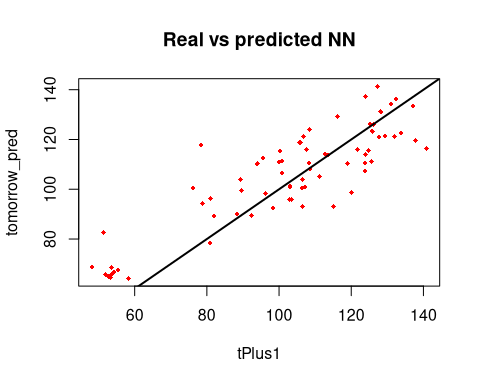
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 10.40969

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 0.1193085

plot(data.frame(fourInput2\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)

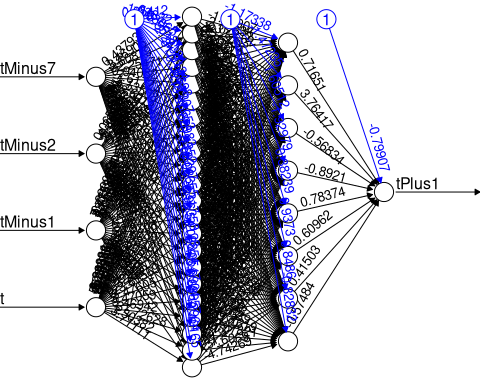


final\_result <- cbind(fourInput2\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 423 107.2 100.92438  
## 424 103.0 100.93163  
## 425 81.0 96.13211  
## 426 53.2 64.57237  
## 427 113.6 113.83882  
## 428 100.8 111.27003  
## 429 124.0 113.99119  
## 430 124.8 115.42845  
## 431 119.0 110.27030  
## 432 89.4 103.93752  
## 433 58.2 64.16877  
## 434 133.8 122.72588  
## 435 124.0 137.26390  
## 436 126.2 125.91301  
## 437 121.8 115.91984  
## 438 123.8 110.59192  
## 439 95.6 112.50681  
## 440 54.2 66.69581  
## 441 125.8 123.28300  
## 442 128.2 131.17609  
## 443 116.2 129.08493  
## 444 123.8 107.20832  
## 445 112.8 114.18972  
## 446 106.4 103.89696  
## 447 51.4 82.68535  
## 448 129.4 121.45342  
## 449 137.0 133.44543  
## 450 132.4 136.14795  
## 451 140.8 116.40142  
## 452 125.2 126.26396  
## 453 94.0 110.19704  
## 454 55.4 67.48895  
## 455 137.8 119.67309  
## 456 127.2 141.33556  
## 457 131.0 134.26174  
## 458 132.0 121.24340  
## 459 127.8 120.98519  
## 460 100.2 115.34431  
## 461 53.6 68.56041  
## 462 108.4 124.12409  
## 463 120.0 98.75572  
## 464 106.8 121.10599  
## 465 103.0 101.17239  
## 466 89.6 99.39194  
## 467 88.4 90.00363  
## 468 48.2 68.62255  
## 469 103.4 95.86246  
## 470 125.6 111.14203  
## 471 100.0 110.87653  
## 472 98.2 92.46929  
## 473 92.4 89.34287  
## 474 80.8 78.41452  
## 475 51.8 65.71949  
## 476 111.2 105.28795  
## 477 108.4 110.53851  
## 478 105.8 118.78981  
## 479 103.0 95.89865  
## 480 106.4 93.07443  
## 481 78.8 94.29146  
## 482 53.6 65.79684  
## 483 78.4 117.66557  
## 484 106.4 100.60791  
## 485 115.2 93.08098  
## 486 108.6 108.19023  
## 487 96.2 98.30344  
## 488 82.0 89.15382  
## 489 52.8 64.83492  
## 490 100.8 106.38136  
## 491 107.6 115.80919  
## 492 76.2 100.53333

### 4 input h1 22 h2 8

set.seed(12345)  
fourInput2Model.h1.22.h2.8.TRUE.nn<-neuralnet(tPlus1~tMinus7+tMinus2+tMinus1+t,data = fourInput2\_train,hidden = c(22,8),act.fct = "logistic",linear.output = TRUE)  
  
plot(fourInput2Model.h1.22.h2.8.TRUE.nn,rep = "best")



fourInput2Model.h1.22.h2.8.TRUE.nn\_Result<-compute(fourInput2Model.h1.22.h2.8.TRUE.nn,fourInput2\_test[1:4])  
  
  
predicted\_tomorrow<-fourInput2Model.h1.22.h2.8.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput2\_test$tPlus1)

## [,1]  
## [1,] 0.4898705

fourInput2\_train<-fourInput2\_norm[1:422,]  
fourInput2\_test<-fourInput2\_norm[423:492,]  
  
fourInput2\_train\_original<-as.data.frame(fourInput2[1:422,"tPlus1"])  
  
  
fourInput2\_test\_original<-as.data.frame(fourInput2[423:492,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput2\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput2\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput2\_train\_original)

## tPlus1  
## 1 116.0  
## 2 114.2  
## 3 113.4  
## 4 120.4  
## 5 84.8  
## 6 66.8

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 423 105.39483  
## 424 96.24024  
## 425 90.66070  
## 426 64.83783  
## 427 98.67070  
## 428 112.18752  
## 429 114.34123  
## 430 128.11673  
## 431 125.48985  
## 432 114.40134  
## 433 65.24495  
## 434 111.10533  
## 435 124.31176  
## 436 104.06357  
## 437 124.38019  
## 438 113.43138  
## 439 107.76499  
## 440 69.10305  
## 441 140.80964  
## 442 132.28629  
## 443 123.88073  
## 444 117.34571  
## 445 111.40447  
## 446 113.72119  
## 447 122.80017  
## 448 117.74621  
## 449 83.86426  
## 450 136.52843  
## 451 123.68469  
## 452 142.30171  
## 453 90.34236  
## 454 73.78724  
## 455 112.47570  
## 456 132.07010  
## 457 136.03759  
## 458 115.79248  
## 459 120.40635  
## 460 116.06143  
## 461 72.65905  
## 462 112.60498  
## 463 56.10750  
## 464 118.03032  
## 465 115.44935  
## 466 105.02823  
## 467 71.05379  
## 468 71.96277  
## 469 120.26011  
## 470 42.47882  
## 471 127.14718  
## 472 97.92313  
## 473 90.64609  
## 474 86.75070  
## 475 30.89854  
## 476 128.44074  
## 477 72.15477  
## 478 107.26298  
## 479 96.18826  
## 480 88.50205  
## 481 94.56977  
## 482 62.35297  
## 483 134.53077  
## 484 26.52160  
## 485 97.20026  
## 486 101.19819  
## 487 104.93911  
## 488 100.88891  
## 489 70.66672  
## 490 108.87871  
## 491 18.19959  
## 492 101.99379

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 26.60657

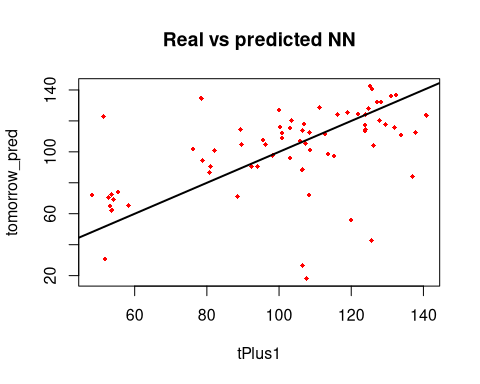
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 18.05455

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 0.198103

plot(data.frame(fourInput2\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)

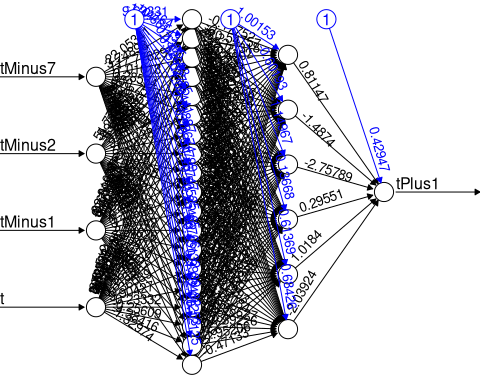


final\_result <- cbind(fourInput2\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 423 107.2 105.39483  
## 424 103.0 96.24024  
## 425 81.0 90.66070  
## 426 53.2 64.83783  
## 427 113.6 98.67070  
## 428 100.8 112.18752  
## 429 124.0 114.34123  
## 430 124.8 128.11673  
## 431 119.0 125.48985  
## 432 89.4 114.40134  
## 433 58.2 65.24495  
## 434 133.8 111.10533  
## 435 124.0 124.31176  
## 436 126.2 104.06357  
## 437 121.8 124.38019  
## 438 123.8 113.43138  
## 439 95.6 107.76499  
## 440 54.2 69.10305  
## 441 125.8 140.80964  
## 442 128.2 132.28629  
## 443 116.2 123.88073  
## 444 123.8 117.34571  
## 445 112.8 111.40447  
## 446 106.4 113.72119  
## 447 51.4 122.80017  
## 448 129.4 117.74621  
## 449 137.0 83.86426  
## 450 132.4 136.52843  
## 451 140.8 123.68469  
## 452 125.2 142.30171  
## 453 94.0 90.34236  
## 454 55.4 73.78724  
## 455 137.8 112.47570  
## 456 127.2 132.07010  
## 457 131.0 136.03759  
## 458 132.0 115.79248  
## 459 127.8 120.40635  
## 460 100.2 116.06143  
## 461 53.6 72.65905  
## 462 108.4 112.60498  
## 463 120.0 56.10750  
## 464 106.8 118.03032  
## 465 103.0 115.44935  
## 466 89.6 105.02823  
## 467 88.4 71.05379  
## 468 48.2 71.96277  
## 469 103.4 120.26011  
## 470 125.6 42.47882  
## 471 100.0 127.14718  
## 472 98.2 97.92313  
## 473 92.4 90.64609  
## 474 80.8 86.75070  
## 475 51.8 30.89854  
## 476 111.2 128.44074  
## 477 108.4 72.15477  
## 478 105.8 107.26298  
## 479 103.0 96.18826  
## 480 106.4 88.50205  
## 481 78.8 94.56977  
## 482 53.6 62.35297  
## 483 78.4 134.53077  
## 484 106.4 26.52160  
## 485 115.2 97.20026  
## 486 108.6 101.19819  
## 487 96.2 104.93911  
## 488 82.0 100.88891  
## 489 52.8 70.66672  
## 490 100.8 108.87871  
## 491 107.6 18.19959  
## 492 76.2 101.99379

### 4 input model 2 with hidden layer1 = 19 hidden layer 2 = 6

set.seed(12345)  
  
fourInput2Model.h1.19.h2.6.TRUE.nn<-neuralnet(tPlus1~tMinus7+tMinus2+tMinus1+t,data = fourInput2\_train,hidden = c(19,6),act.fct = "logistic",linear.output = TRUE)  
  
plot(fourInput2Model.h1.19.h2.6.TRUE.nn,rep = "best")



fourInput2Model.h1.19.h2.6.TRUE.nn\_Result<-compute(fourInput2Model.h1.19.h2.6.TRUE.nn,fourInput2\_test[1:4])  
  
  
predicted\_tomorrow<-fourInput2Model.h1.19.h2.6.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput2\_test$tPlus1)

## [,1]  
## [1,] 0.8578953

fourInput2\_train<-fourInput2\_norm[1:422,]  
fourInput2\_test<-fourInput2\_norm[423:492,]  
  
fourInput2\_train\_original<-as.data.frame(fourInput2[1:422,"tPlus1"])  
  
  
fourInput2\_test\_original<-as.data.frame(fourInput2[423:492,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput2\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput2\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput2\_train\_original)

## tPlus1  
## 1 116.0  
## 2 114.2  
## 3 113.4  
## 4 120.4  
## 5 84.8  
## 6 66.8

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 423 99.09466  
## 424 98.74984  
## 425 93.31315  
## 426 66.89030  
## 427 119.10995  
## 428 107.83279  
## 429 115.60944  
## 430 115.24156  
## 431 121.21650  
## 432 104.55675  
## 433 70.41765  
## 434 117.72496  
## 435 130.09555  
## 436 129.51331  
## 437 119.72890  
## 438 109.39372  
## 439 113.65017  
## 440 63.00965  
## 441 116.12781  
## 442 129.83798  
## 443 129.33117  
## 444 105.32295  
## 445 117.72222  
## 446 102.74548  
## 447 87.55774  
## 448 111.14574  
## 449 138.57845  
## 450 142.32527  
## 451 114.96117  
## 452 127.74142  
## 453 107.40782  
## 454 68.95635  
## 455 128.07238  
## 456 143.65013  
## 457 133.63682  
## 458 125.24684  
## 459 122.74978  
## 460 115.17490  
## 461 66.20363  
## 462 125.16394  
## 463 133.60943  
## 464 117.02162  
## 465 111.49994  
## 466 103.54621  
## 467 90.58268  
## 468 82.64291  
## 469 83.95320  
## 470 115.63559  
## 471 120.60228  
## 472 95.36512  
## 473 84.84493  
## 474 81.55872  
## 475 70.26218  
## 476 101.24563  
## 477 101.99210  
## 478 112.21657  
## 479 95.40345  
## 480 92.82681  
## 481 97.04224  
## 482 65.08663  
## 483 106.74197  
## 484 92.02536  
## 485 102.56271  
## 486 107.31773  
## 487 95.08801  
## 488 87.54651  
## 489 67.89171  
## 490 90.43669  
## 491 107.24327  
## 492 103.61468

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 13.48993

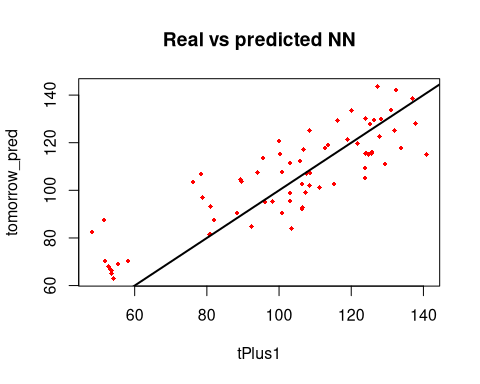
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 11.11006

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 0.1301574

plot(data.frame(fourInput2\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



final\_result <- cbind(fourInput2\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 423 107.2 99.09466  
## 424 103.0 98.74984  
## 425 81.0 93.31315  
## 426 53.2 66.89030  
## 427 113.6 119.10995  
## 428 100.8 107.83279  
## 429 124.0 115.60944  
## 430 124.8 115.24156  
## 431 119.0 121.21650  
## 432 89.4 104.55675  
## 433 58.2 70.41765  
## 434 133.8 117.72496  
## 435 124.0 130.09555  
## 436 126.2 129.51331  
## 437 121.8 119.72890  
## 438 123.8 109.39372  
## 439 95.6 113.65017  
## 440 54.2 63.00965  
## 441 125.8 116.12781  
## 442 128.2 129.83798  
## 443 116.2 129.33117  
## 444 123.8 105.32295  
## 445 112.8 117.72222  
## 446 106.4 102.74548  
## 447 51.4 87.55774  
## 448 129.4 111.14574  
## 449 137.0 138.57845  
## 450 132.4 142.32527  
## 451 140.8 114.96117  
## 452 125.2 127.74142  
## 453 94.0 107.40782  
## 454 55.4 68.95635  
## 455 137.8 128.07238  
## 456 127.2 143.65013  
## 457 131.0 133.63682  
## 458 132.0 125.24684  
## 459 127.8 122.74978  
## 460 100.2 115.17490  
## 461 53.6 66.20363  
## 462 108.4 125.16394  
## 463 120.0 133.60943  
## 464 106.8 117.02162  
## 465 103.0 111.49994  
## 466 89.6 103.54621  
## 467 88.4 90.58268  
## 468 48.2 82.64291  
## 469 103.4 83.95320  
## 470 125.6 115.63559  
## 471 100.0 120.60228  
## 472 98.2 95.36512  
## 473 92.4 84.84493  
## 474 80.8 81.55872  
## 475 51.8 70.26218  
## 476 111.2 101.24563  
## 477 108.4 101.99210  
## 478 105.8 112.21657  
## 479 103.0 95.40345  
## 480 106.4 92.82681  
## 481 78.8 97.04224  
## 482 53.6 65.08663  
## 483 78.4 106.74197  
## 484 106.4 92.02536  
## 485 115.2 102.56271  
## 486 108.6 107.31773  
## 487 96.2 95.08801  
## 488 82.0 87.54651  
## 489 52.8 67.89171  
## 490 100.8 90.43669  
## 491 107.6 107.24327  
## 492 76.2 103.61468

### 4 input model 2 with hidden layer1 = 13 hidden layer 2 = 3

set.seed(12345)  
fourInput2Model.h1.13.h2.3.TRUE.nn<-neuralnet(tPlus1~tMinus7+tMinus2+tMinus1+t,data = fourInput2\_train,hidden = c(13,3),act.fct = "logistic",linear.output = TRUE)  
  
plot(fourInput2Model.h1.13.h2.3.TRUE.nn)  
  
fourInput2Model.h1.13.h2.3.TRUE.nn\_Result<-compute(fourInput2Model.h1.13.h2.3.TRUE.nn,fourInput2\_test[1:4])  
  
  
predicted\_tomorrow<-fourInput2Model.h1.13.h2.3.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput2\_test$tPlus1)

## [,1]  
## [1,] 0.860829

fourInput2\_train<-fourInput2\_norm[1:422,]  
fourInput2\_test<-fourInput2\_norm[423:492,]  
  
fourInput2\_train\_original<-as.data.frame(fourInput2[1:422,"tPlus1"])  
  
  
fourInput2\_test\_original<-as.data.frame(fourInput2[423:492,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput2\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput2\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput2\_train\_original)

## tPlus1  
## 1 116.0  
## 2 114.2  
## 3 113.4  
## 4 120.4  
## 5 84.8  
## 6 66.8

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 423 98.00652  
## 424 100.33416  
## 425 92.63977  
## 426 67.71819  
## 427 101.09992  
## 428 116.92490  
## 429 109.63731  
## 430 116.08624  
## 431 110.87367  
## 432 105.67714  
## 433 65.48559  
## 434 124.15788  
## 435 131.89069  
## 436 127.05603  
## 437 119.11469  
## 438 111.60276  
## 439 113.44590  
## 440 68.23537  
## 441 123.54104  
## 442 132.89402  
## 443 128.22199  
## 444 107.21470  
## 445 118.88104  
## 446 104.98212  
## 447 79.36305  
## 448 128.27108  
## 449 138.78859  
## 450 134.85038  
## 451 116.56659  
## 452 126.19682  
## 453 110.91368  
## 454 67.16166  
## 455 124.92384  
## 456 140.59965  
## 457 132.56801  
## 458 125.64297  
## 459 121.37093  
## 460 115.00308  
## 461 67.91515  
## 462 124.88980  
## 463 136.61272  
## 464 120.17154  
## 465 102.51953  
## 466 105.76876  
## 467 91.42622  
## 468 80.55573  
## 469 92.56395  
## 470 115.03925  
## 471 122.50419  
## 472 90.06302  
## 473 83.73194  
## 474 78.53093  
## 475 66.37062  
## 476 111.48889  
## 477 115.86070  
## 478 114.87144  
## 479 91.87609  
## 480 89.06845  
## 481 93.67852  
## 482 66.70085  
## 483 119.08315  
## 484 108.78742  
## 485 104.55861  
## 486 106.93471  
## 487 97.10718  
## 488 82.67221  
## 489 63.82201  
## 490 103.13653  
## 491 96.17871  
## 492 108.99475

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 13.46795

MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 10.7206

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput2\_test\_original)))

## [1] 0.1258685

plot(data.frame(fourInput2\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)  
  
  
final\_result <- cbind(fourInput2\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 423 107.2 98.00652  
## 424 103.0 100.33416  
## 425 81.0 92.63977  
## 426 53.2 67.71819  
## 427 113.6 101.09992  
## 428 100.8 116.92490  
## 429 124.0 109.63731  
## 430 124.8 116.08624  
## 431 119.0 110.87367  
## 432 89.4 105.67714  
## 433 58.2 65.48559  
## 434 133.8 124.15788  
## 435 124.0 131.89069  
## 436 126.2 127.05603  
## 437 121.8 119.11469  
## 438 123.8 111.60276  
## 439 95.6 113.44590  
## 440 54.2 68.23537  
## 441 125.8 123.54104  
## 442 128.2 132.89402  
## 443 116.2 128.22199  
## 444 123.8 107.21470  
## 445 112.8 118.88104  
## 446 106.4 104.98212  
## 447 51.4 79.36305  
## 448 129.4 128.27108  
## 449 137.0 138.78859  
## 450 132.4 134.85038  
## 451 140.8 116.56659  
## 452 125.2 126.19682  
## 453 94.0 110.91368  
## 454 55.4 67.16166  
## 455 137.8 124.92384  
## 456 127.2 140.59965  
## 457 131.0 132.56801  
## 458 132.0 125.64297  
## 459 127.8 121.37093  
## 460 100.2 115.00308  
## 461 53.6 67.91515  
## 462 108.4 124.88980  
## 463 120.0 136.61272  
## 464 106.8 120.17154  
## 465 103.0 102.51953  
## 466 89.6 105.76876  
## 467 88.4 91.42622  
## 468 48.2 80.55573  
## 469 103.4 92.56395  
## 470 125.6 115.03925  
## 471 100.0 122.50419  
## 472 98.2 90.06302  
## 473 92.4 83.73194  
## 474 80.8 78.53093  
## 475 51.8 66.37062  
## 476 111.2 111.48889  
## 477 108.4 115.86070  
## 478 105.8 114.87144  
## 479 103.0 91.87609  
## 480 106.4 89.06845  
## 481 78.8 93.67852  
## 482 53.6 66.70085  
## 483 78.4 119.08315  
## 484 106.4 108.78742  
## 485 115.2 104.55861  
## 486 108.6 106.93471  
## 487 96.2 97.10718  
## 488 82.0 82.67221  
## 489 52.8 63.82201  
## 490 100.8 103.13653  
## 491 107.6 96.17871  
## 492 76.2 108.99475

## 4 inputs t-3 ,t-2 ,t-1 ,t

### 4 input h1 12

set.seed(12345)  
fourInputModel.h1.12.TRUE.nn<-neuralnet(tPlus1~tMinus3+tMinus2+tMinus1+t,  
 data = fourInput\_train,hidden = c(12),  
 act.fct = "logistic",linear.output = TRUE)  
plot(fourInputModel.h1.12.TRUE.nn)  
  
  
fourInputModel.h1.12.TRUE.nn\_Result<-compute(fourInputModel.h1.12.TRUE.nn,fourInput\_test[1:4])  
  
  
predicted\_tomorrow<-fourInputModel.h1.12.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput\_test$tPlus1)

## [,1]  
## [1,] 0.8654291

fourInput\_train<-fourInput\_norm[1:426,]  
fourInput\_test<-fourInput\_norm[427:496,]  
  
fourInput\_train\_original<-as.data.frame(fourInput[1:426,"tPlus1"])  
  
  
fourInput\_test\_original<-as.data.frame(fourInput[427:496,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput\_train\_original)

## tPlus1  
## 1 102.4  
## 2 87.2  
## 3 67.6  
## 4 116.2  
## 5 116.0  
## 6 114.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 427 109.28747  
## 428 91.21362  
## 429 86.40901  
## 430 65.70321  
## 431 101.14864  
## 432 120.19916  
## 433 114.58704  
## 434 122.68661  
## 435 108.76283  
## 436 108.02641  
## 437 66.94672  
## 438 125.77848  
## 439 132.93402  
## 440 132.04889  
## 441 126.95703  
## 442 103.10324  
## 443 109.97852  
## 444 70.95184  
## 445 126.26102  
## 446 133.54639  
## 447 129.48256  
## 448 120.31505  
## 449 112.31858  
## 450 92.46283  
## 451 90.07475  
## 452 119.52063  
## 453 139.41158  
## 454 136.89591  
## 455 135.26987  
## 456 128.22364  
## 457 104.26214  
## 458 66.63845  
## 459 134.47853  
## 460 138.32100  
## 461 138.31116  
## 462 132.85484  
## 463 115.16861  
## 464 112.34020  
## 465 72.28767  
## 466 133.52714  
## 467 131.02586  
## 468 118.03456  
## 469 108.97402  
## 470 90.42431  
## 471 71.59871  
## 472 77.44710  
## 473 88.17399  
## 474 122.58340  
## 475 119.46722  
## 476 106.88936  
## 477 93.22370  
## 478 76.15050  
## 479 68.82110  
## 480 92.03557  
## 481 119.12640  
## 482 116.60328  
## 483 110.85022  
## 484 87.30786  
## 485 93.28794  
## 486 63.76587  
## 487 99.32815  
## 488 132.06653  
## 489 102.43467  
## 490 113.69740  
## 491 103.00010  
## 492 79.07954  
## 493 68.26530  
## 494 98.57441  
## 495 113.62699  
## 496 108.63360

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 13.53395

MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 11.04124

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

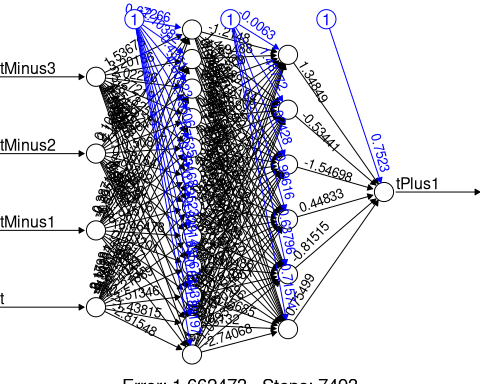
## [1] 0.1308255

plot(data.frame(fourInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)  
  
  
final\_result <- cbind(fourInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 427 107.2 109.28747  
## 428 103.0 91.21362  
## 429 81.0 86.40901  
## 430 53.2 65.70321  
## 431 113.6 101.14864  
## 432 100.8 120.19916  
## 433 124.0 114.58704  
## 434 124.8 122.68661  
## 435 119.0 108.76283  
## 436 89.4 108.02641  
## 437 58.2 66.94672  
## 438 133.8 125.77848  
## 439 124.0 132.93402  
## 440 126.2 132.04889  
## 441 121.8 126.95703  
## 442 123.8 103.10324  
## 443 95.6 109.97852  
## 444 54.2 70.95184  
## 445 125.8 126.26102  
## 446 128.2 133.54639  
## 447 116.2 129.48256  
## 448 123.8 120.31505  
## 449 112.8 112.31858  
## 450 106.4 92.46283  
## 451 51.4 90.07475  
## 452 129.4 119.52063  
## 453 137.0 139.41158  
## 454 132.4 136.89591  
## 455 140.8 135.26987  
## 456 125.2 128.22364  
## 457 94.0 104.26214  
## 458 55.4 66.63845  
## 459 137.8 134.47853  
## 460 127.2 138.32100  
## 461 131.0 138.31116  
## 462 132.0 132.85484  
## 463 127.8 115.16861  
## 464 100.2 112.34020  
## 465 53.6 72.28767  
## 466 108.4 133.52714  
## 467 120.0 131.02586  
## 468 106.8 118.03456  
## 469 103.0 108.97402  
## 470 89.6 90.42431  
## 471 88.4 71.59871  
## 472 48.2 77.44710  
## 473 103.4 88.17399  
## 474 125.6 122.58340  
## 475 100.0 119.46722  
## 476 98.2 106.88936  
## 477 92.4 93.22370  
## 478 80.8 76.15050  
## 479 51.8 68.82110  
## 480 111.2 92.03557  
## 481 108.4 119.12640  
## 482 105.8 116.60328  
## 483 103.0 110.85022  
## 484 106.4 87.30786  
## 485 78.8 93.28794  
## 486 53.6 63.76587  
## 487 78.4 99.32815  
## 488 106.4 132.06653  
## 489 115.2 102.43467  
## 490 108.6 113.69740  
## 491 96.2 103.00010  
## 492 82.0 79.07954  
## 493 52.8 68.26530  
## 494 100.8 98.57441  
## 495 107.6 113.62699  
## 496 76.2 108.63360

’### 4 input h1 12 h2 6

set.seed(12345)  
fourInputModel.h1.12.h2.6.TRUE.nn<-neuralnet(tPlus1~tMinus3+tMinus2+tMinus1+t,  
 data = fourInput\_train,hidden = c(12,6),  
 act.fct = "logistic",linear.output = TRUE)  
plot(fourInputModel.h1.12.h2.6.TRUE.nn, rep="best")



fourInputModel.h1.12.h2.6.TRUE.nn\_Result<-compute(fourInputModel.h1.12.h2.6.TRUE.nn,fourInput\_test[1:4])  
  
  
predicted\_tomorrow<-fourInputModel.h1.12.h2.6.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput\_test$tPlus1)

## [,1]  
## [1,] 0.83887

fourInput\_train<-fourInput\_norm[1:426,]  
fourInput\_test<-fourInput\_norm[427:496,]  
  
fourInput\_train\_original<-as.data.frame(fourInput[1:426,"tPlus1"])  
  
  
fourInput\_test\_original<-as.data.frame(fourInput[427:496,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput\_train\_original)

## tPlus1  
## 1 102.4  
## 2 87.2  
## 3 67.6  
## 4 116.2  
## 5 116.0  
## 6 114.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 427 110.70692  
## 428 91.57261  
## 429 87.26998  
## 430 66.91469  
## 431 102.68952  
## 432 110.45344  
## 433 107.60761  
## 434 118.00774  
## 435 111.63228  
## 436 109.08161  
## 437 63.84904  
## 438 125.64671  
## 439 127.66070  
## 440 124.12594  
## 441 126.09688  
## 442 100.43013  
## 443 107.94416  
## 444 66.57254  
## 445 122.81687  
## 446 131.58748  
## 447 125.84490  
## 448 118.95621  
## 449 112.85685  
## 450 91.33416  
## 451 97.47647  
## 452 107.90531  
## 453 135.66808  
## 454 132.28897  
## 455 130.05024  
## 456 128.20496  
## 457 102.32867  
## 458 77.15684  
## 459 133.44996  
## 460 132.68985  
## 461 126.52731  
## 462 129.86538  
## 463 109.03699  
## 464 109.92169  
## 465 61.18673  
## 466 128.36109  
## 467 115.54232  
## 468 116.28499  
## 469 111.88854  
## 470 95.89033  
## 471 66.44041  
## 472 70.62651  
## 473 85.85831  
## 474 108.60488  
## 475 118.04051  
## 476 110.69462  
## 477 102.28250  
## 478 69.75885  
## 479 63.71431  
## 480 93.46939  
## 481 106.04511  
## 482 111.15772  
## 483 107.44814  
## 484 87.95455  
## 485 91.90721  
## 486 64.93708  
## 487 104.28617  
## 488 80.47443  
## 489 97.74420  
## 490 112.22173  
## 491 107.30694  
## 492 81.68823  
## 493 61.74497  
## 494 95.02235  
## 495 108.46356  
## 496 107.34318

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 13.8611

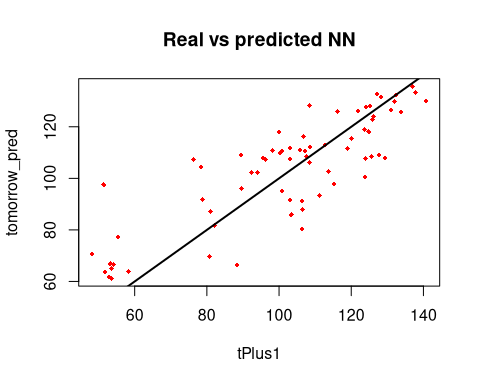
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 11.05985

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 0.1288353

plot(data.frame(fourInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



final\_result <- cbind(fourInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 427 107.2 110.70692  
## 428 103.0 91.57261  
## 429 81.0 87.26998  
## 430 53.2 66.91469  
## 431 113.6 102.68952  
## 432 100.8 110.45344  
## 433 124.0 107.60761  
## 434 124.8 118.00774  
## 435 119.0 111.63228  
## 436 89.4 109.08161  
## 437 58.2 63.84904  
## 438 133.8 125.64671  
## 439 124.0 127.66070  
## 440 126.2 124.12594  
## 441 121.8 126.09688  
## 442 123.8 100.43013  
## 443 95.6 107.94416  
## 444 54.2 66.57254  
## 445 125.8 122.81687  
## 446 128.2 131.58748  
## 447 116.2 125.84490  
## 448 123.8 118.95621  
## 449 112.8 112.85685  
## 450 106.4 91.33416  
## 451 51.4 97.47647  
## 452 129.4 107.90531  
## 453 137.0 135.66808  
## 454 132.4 132.28897  
## 455 140.8 130.05024  
## 456 125.2 128.20496  
## 457 94.0 102.32867  
## 458 55.4 77.15684  
## 459 137.8 133.44996  
## 460 127.2 132.68985  
## 461 131.0 126.52731  
## 462 132.0 129.86538  
## 463 127.8 109.03699  
## 464 100.2 109.92169  
## 465 53.6 61.18673  
## 466 108.4 128.36109  
## 467 120.0 115.54232  
## 468 106.8 116.28499  
## 469 103.0 111.88854  
## 470 89.6 95.89033  
## 471 88.4 66.44041  
## 472 48.2 70.62651  
## 473 103.4 85.85831  
## 474 125.6 108.60488  
## 475 100.0 118.04051  
## 476 98.2 110.69462  
## 477 92.4 102.28250  
## 478 80.8 69.75885  
## 479 51.8 63.71431  
## 480 111.2 93.46939  
## 481 108.4 106.04511  
## 482 105.8 111.15772  
## 483 103.0 107.44814  
## 484 106.4 87.95455  
## 485 78.8 91.90721  
## 486 53.6 64.93708  
## 487 78.4 104.28617  
## 488 106.4 80.47443  
## 489 115.2 97.74420  
## 490 108.6 112.22173  
## 491 96.2 107.30694  
## 492 82.0 81.68823  
## 493 52.8 61.74497  
## 494 100.8 95.02235  
## 495 107.6 108.46356  
## 496 76.2 107.34318

### 4 input h1 18 h2 4

set.seed(12345)  
fourInputModel.h1.18.h2.4.TRUE.nn<-neuralnet(tPlus1~tMinus3+tMinus2+tMinus1+t,  
 data = fourInput\_train,hidden = c(18,4),  
 act.fct = "logistic",linear.output = TRUE)  
  
plot(fourInputModel.h1.18.h2.4.TRUE.nn)  
  
  
  
  
  
fourInputModel.h1.18.h2.4.TRUE.nn\_Result<-compute(fourInputModel.h1.18.h2.4.TRUE.nn,fourInput\_test[1:4])  
  
  
predicted\_tomorrow<-fourInputModel.h1.18.h2.4.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput\_test$tPlus1)

## [,1]  
## [1,] 0.8268304

fourInput\_train<-fourInput\_norm[1:426,]  
fourInput\_test<-fourInput\_norm[427:496,]  
  
fourInput\_train\_original<-as.data.frame(fourInput[1:426,"tPlus1"])  
  
  
fourInput\_test\_original<-as.data.frame(fourInput[427:496,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput\_train\_original)

## tPlus1  
## 1 102.4  
## 2 87.2  
## 3 67.6  
## 4 116.2  
## 5 116.0  
## 6 114.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 427 115.41387  
## 428 93.35983  
## 429 88.21911  
## 430 65.11493  
## 431 103.57940  
## 432 118.96015  
## 433 110.53292  
## 434 120.89058  
## 435 108.86539  
## 436 106.98582  
## 437 65.26756  
## 438 123.02868  
## 439 133.08911  
## 440 125.50454  
## 441 133.65202  
## 442 101.57406  
## 443 108.19842  
## 444 70.47613  
## 445 122.67309  
## 446 135.23398  
## 447 125.84114  
## 448 126.53596  
## 449 112.74267  
## 450 92.31002  
## 451 93.96114  
## 452 124.33926  
## 453 146.18032  
## 454 133.56988  
## 455 132.13146  
## 456 124.69183  
## 457 102.14447  
## 458 75.57281  
## 459 129.48905  
## 460 140.43251  
## 461 127.93967  
## 462 138.68699  
## 463 111.17348  
## 464 109.35985  
## 465 71.31981  
## 466 127.19939  
## 467 173.15407  
## 468 115.27421  
## 469 113.23750  
## 470 93.98272  
## 471 70.46465  
## 472 77.68812  
## 473 98.44016  
## 474 121.75525  
## 475 119.87778  
## 476 113.57375  
## 477 80.76820  
## 478 75.76763  
## 479 66.78505  
## 480 94.40314  
## 481 116.75791  
## 482 113.45463  
## 483 115.44462  
## 484 89.27848  
## 485 94.96351  
## 486 62.90710  
## 487 106.23399  
## 488 134.58293  
## 489 100.45455  
## 490 114.53357  
## 491 101.17478  
## 492 80.42555  
## 493 65.60819  
## 494 98.08057  
## 495 112.96321  
## 496 108.31016

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 15.31159

MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 12.1437

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

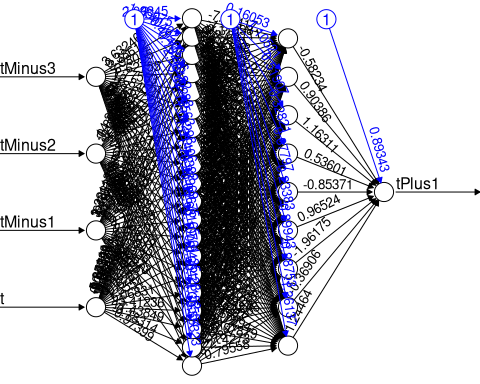
## [1] 0.1410865

plot(data.frame(fourInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)  
  
  
final\_result <- cbind(fourInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 427 107.2 115.41387  
## 428 103.0 93.35983  
## 429 81.0 88.21911  
## 430 53.2 65.11493  
## 431 113.6 103.57940  
## 432 100.8 118.96015  
## 433 124.0 110.53292  
## 434 124.8 120.89058  
## 435 119.0 108.86539  
## 436 89.4 106.98582  
## 437 58.2 65.26756  
## 438 133.8 123.02868  
## 439 124.0 133.08911  
## 440 126.2 125.50454  
## 441 121.8 133.65202  
## 442 123.8 101.57406  
## 443 95.6 108.19842  
## 444 54.2 70.47613  
## 445 125.8 122.67309  
## 446 128.2 135.23398  
## 447 116.2 125.84114  
## 448 123.8 126.53596  
## 449 112.8 112.74267  
## 450 106.4 92.31002  
## 451 51.4 93.96114  
## 452 129.4 124.33926  
## 453 137.0 146.18032  
## 454 132.4 133.56988  
## 455 140.8 132.13146  
## 456 125.2 124.69183  
## 457 94.0 102.14447  
## 458 55.4 75.57281  
## 459 137.8 129.48905  
## 460 127.2 140.43251  
## 461 131.0 127.93967  
## 462 132.0 138.68699  
## 463 127.8 111.17348  
## 464 100.2 109.35985  
## 465 53.6 71.31981  
## 466 108.4 127.19939  
## 467 120.0 173.15407  
## 468 106.8 115.27421  
## 469 103.0 113.23750  
## 470 89.6 93.98272  
## 471 88.4 70.46465  
## 472 48.2 77.68812  
## 473 103.4 98.44016  
## 474 125.6 121.75525  
## 475 100.0 119.87778  
## 476 98.2 113.57375  
## 477 92.4 80.76820  
## 478 80.8 75.76763  
## 479 51.8 66.78505  
## 480 111.2 94.40314  
## 481 108.4 116.75791  
## 482 105.8 113.45463  
## 483 103.0 115.44462  
## 484 106.4 89.27848  
## 485 78.8 94.96351  
## 486 53.6 62.90710  
## 487 78.4 106.23399  
## 488 106.4 134.58293  
## 489 115.2 100.45455  
## 490 108.6 114.53357  
## 491 96.2 101.17478  
## 492 82.0 80.42555  
## 493 52.8 65.60819  
## 494 100.8 98.08057  
## 495 107.6 112.96321  
## 496 76.2 108.31016

### 4 input h1 20 h2 9

set.seed(12345)  
fourInputModel.h1.20.h2.9.TRUE.nn<-neuralnet(tPlus1~tMinus3+tMinus2+tMinus1+t,  
 data = fourInput\_train,hidden = c(20,9),  
 act.fct = "logistic",linear.output = TRUE)  
  
plot(fourInputModel.h1.20.h2.9.TRUE.nn, rep = "best")



fourInputModel.h1.20.h2.9.TRUE.nn\_Result<-compute(fourInputModel.h1.20.h2.9.TRUE.nn,fourInput\_test[1:4])  
  
  
predicted\_tomorrow<-fourInputModel.h1.20.h2.9.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,fourInput\_test$tPlus1)

## [,1]  
## [1,] 0.8270169

fourInput\_train<-fourInput\_norm[1:426,]  
fourInput\_test<-fourInput\_norm[427:496,]  
  
fourInput\_train\_original<-as.data.frame(fourInput[1:426,"tPlus1"])  
  
  
fourInput\_test\_original<-as.data.frame(fourInput[427:496,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(fourInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(fourInput\_train\_original)  
#tomorrow\_max  
  
  
head(fourInput\_train\_original)

## tPlus1  
## 1 102.4  
## 2 87.2  
## 3 67.6  
## 4 116.2  
## 5 116.0  
## 6 114.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 427 112.38601  
## 428 93.04354  
## 429 87.87836  
## 430 65.07969  
## 431 104.40162  
## 432 113.84713  
## 433 101.10768  
## 434 114.94482  
## 435 106.40685  
## 436 110.71020  
## 437 68.58837  
## 438 124.17220  
## 439 127.26352  
## 440 128.35465  
## 441 129.09860  
## 442 101.72657  
## 443 108.52525  
## 444 65.99960  
## 445 122.68385  
## 446 133.39028  
## 447 131.57047  
## 448 116.99503  
## 449 111.54690  
## 450 92.85489  
## 451 93.78215  
## 452 115.11240  
## 453 121.97393  
## 454 133.33047  
## 455 126.89569  
## 456 128.44917  
## 457 104.15415  
## 458 75.77308  
## 459 135.56350  
## 460 132.33473  
## 461 137.31683  
## 462 134.11745  
## 463 109.77183  
## 464 111.10130  
## 465 68.49806  
## 466 125.25786  
## 467 128.57362  
## 468 104.80453  
## 469 108.70721  
## 470 91.87080  
## 471 64.20870  
## 472 69.18129  
## 473 73.22007  
## 474 113.73542  
## 475 109.35641  
## 476 100.31148  
## 477 88.81573  
## 478 68.88074  
## 479 64.15062  
## 480 83.73842  
## 481 105.73436  
## 482 107.12429  
## 483 109.76282  
## 484 88.74548  
## 485 94.37786  
## 486 65.15036  
## 487 106.93987  
## 488 82.89845  
## 489 105.91693  
## 490 126.78670  
## 491 104.07762  
## 492 76.45185  
## 493 66.77084  
## 494 89.48380  
## 495 113.77829  
## 496 100.70742

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 14.27877

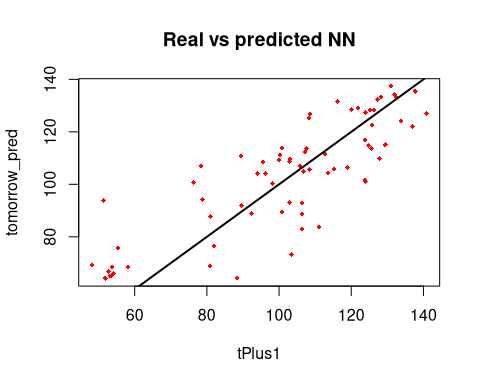
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 11.73427

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(fourInput\_test\_original)))

## [1] 0.1353839

plot(data.frame(fourInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



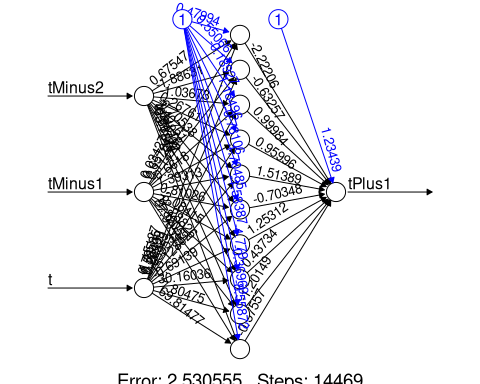
final\_result <- cbind(fourInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 427 107.2 112.38601  
## 428 103.0 93.04354  
## 429 81.0 87.87836  
## 430 53.2 65.07969  
## 431 113.6 104.40162  
## 432 100.8 113.84713  
## 433 124.0 101.10768  
## 434 124.8 114.94482  
## 435 119.0 106.40685  
## 436 89.4 110.71020  
## 437 58.2 68.58837  
## 438 133.8 124.17220  
## 439 124.0 127.26352  
## 440 126.2 128.35465  
## 441 121.8 129.09860  
## 442 123.8 101.72657  
## 443 95.6 108.52525  
## 444 54.2 65.99960  
## 445 125.8 122.68385  
## 446 128.2 133.39028  
## 447 116.2 131.57047  
## 448 123.8 116.99503  
## 449 112.8 111.54690  
## 450 106.4 92.85489  
## 451 51.4 93.78215  
## 452 129.4 115.11240  
## 453 137.0 121.97393  
## 454 132.4 133.33047  
## 455 140.8 126.89569  
## 456 125.2 128.44917  
## 457 94.0 104.15415  
## 458 55.4 75.77308  
## 459 137.8 135.56350  
## 460 127.2 132.33473  
## 461 131.0 137.31683  
## 462 132.0 134.11745  
## 463 127.8 109.77183  
## 464 100.2 111.10130  
## 465 53.6 68.49806  
## 466 108.4 125.25786  
## 467 120.0 128.57362  
## 468 106.8 104.80453  
## 469 103.0 108.70721  
## 470 89.6 91.87080  
## 471 88.4 64.20870  
## 472 48.2 69.18129  
## 473 103.4 73.22007  
## 474 125.6 113.73542  
## 475 100.0 109.35641  
## 476 98.2 100.31148  
## 477 92.4 88.81573  
## 478 80.8 68.88074  
## 479 51.8 64.15062  
## 480 111.2 83.73842  
## 481 108.4 105.73436  
## 482 105.8 107.12429  
## 483 103.0 109.76282  
## 484 106.4 88.74548  
## 485 78.8 94.37786  
## 486 53.6 65.15036  
## 487 78.4 106.93987  
## 488 106.4 82.89845  
## 489 115.2 105.91693  
## 490 108.6 126.78670  
## 491 96.2 104.07762  
## 492 82.0 76.45185  
## 493 52.8 66.77084  
## 494 100.8 89.48380  
## 495 107.6 113.77829  
## 496 76.2 100.70742

## 3 input model t-2, t-1, t

### 2 input h1 10

set.seed(12345)  
threeInputModel.h1.10.TRUE.nn<-neuralnet(tPlus1~tMinus2+tMinus1+t,  
 data = threeInput\_train,hidden = c(10),  
 act.fct = "logistic",linear.output = TRUE)  
  
plot(threeInputModel.h1.10.TRUE.nn, rep= "best")



threeInputModel.h1.10.TRUE.nn\_Result<-compute(threeInputModel.h1.10.TRUE.nn,threeInput\_test[1:3])  
  
  
predicted\_tomorrow<-threeInputModel.h1.10.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,threeInput\_test$tPlus1)

## [,1]  
## [1,] 0.8005061

threeInput\_train<-threeInput\_norm[1:427,]  
threeInput\_test<-threeInput\_norm[428:497,]  
  
  
threeInput\_train\_original<-as.data.frame(threeInput[1:427,"tPlus1"])  
  
  
threeInput\_test\_original<-as.data.frame(threeInput[428:497,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(threeInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(threeInput\_train\_original)  
#tomorrow\_max  
  
  
head(threeInput\_train\_original)

## tPlus1  
## 1 109.0  
## 2 102.4  
## 3 87.2  
## 4 67.6  
## 5 116.2  
## 6 116.0

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 428 99.58419  
## 429 99.29377  
## 430 94.26516  
## 431 63.26021  
## 432 105.42245  
## 433 116.40639  
## 434 110.34088  
## 435 118.63714  
## 436 111.81026  
## 437 108.77897  
## 438 65.04129  
## 439 118.65096  
## 440 132.45671  
## 441 126.56322  
## 442 118.91152  
## 443 111.17199  
## 444 115.01894  
## 445 69.54444  
## 446 123.17207  
## 447 134.10015  
## 448 129.04014  
## 449 104.37596  
## 450 118.08640  
## 451 102.17162  
## 452 93.52822  
## 453 115.94882  
## 454 147.53440  
## 455 138.18540  
## 456 118.25427  
## 457 128.82049  
## 458 111.51520  
## 459 69.59385  
## 460 124.22854  
## 461 140.08096  
## 462 131.37178  
## 463 123.63254  
## 464 118.98344  
## 465 116.34850  
## 466 71.44988  
## 467 126.58537  
## 468 176.89322  
## 469 118.40670  
## 470 97.24998  
## 471 90.40987  
## 472 71.51656  
## 473 74.84639  
## 474 97.20345  
## 475 162.44050  
## 476 123.48539  
## 477 90.60911  
## 478 81.07126  
## 479 81.88898  
## 480 65.15025  
## 481 97.62394  
## 482 115.48003  
## 483 112.94401  
## 484 97.54737  
## 485 94.70315  
## 486 100.38072  
## 487 63.07737  
## 488 108.50970  
## 489 136.05806  
## 490 103.01508  
## 491 107.13335  
## 492 100.21638  
## 493 77.63520  
## 494 64.35359  
## 495 100.11177  
## 496 109.04757  
## 497 108.87780

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 16.25837

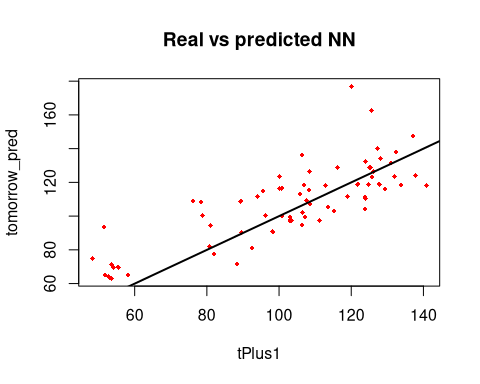
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 12.64046

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 0.1427849

plot(data.frame(threeInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)

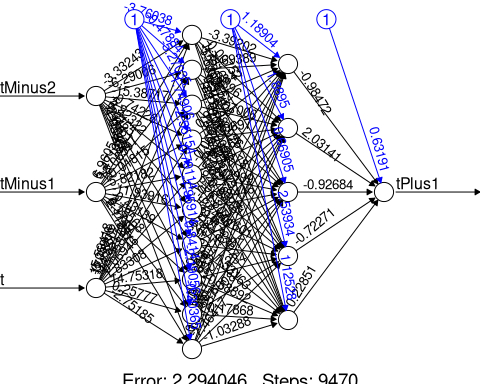


final\_result <- cbind(threeInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 428 107.2 99.58419  
## 429 103.0 99.29377  
## 430 81.0 94.26516  
## 431 53.2 63.26021  
## 432 113.6 105.42245  
## 433 100.8 116.40639  
## 434 124.0 110.34088  
## 435 124.8 118.63714  
## 436 119.0 111.81026  
## 437 89.4 108.77897  
## 438 58.2 65.04129  
## 439 133.8 118.65096  
## 440 124.0 132.45671  
## 441 126.2 126.56322  
## 442 121.8 118.91152  
## 443 123.8 111.17199  
## 444 95.6 115.01894  
## 445 54.2 69.54444  
## 446 125.8 123.17207  
## 447 128.2 134.10015  
## 448 116.2 129.04014  
## 449 123.8 104.37596  
## 450 112.8 118.08640  
## 451 106.4 102.17162  
## 452 51.4 93.52822  
## 453 129.4 115.94882  
## 454 137.0 147.53440  
## 455 132.4 138.18540  
## 456 140.8 118.25427  
## 457 125.2 128.82049  
## 458 94.0 111.51520  
## 459 55.4 69.59385  
## 460 137.8 124.22854  
## 461 127.2 140.08096  
## 462 131.0 131.37178  
## 463 132.0 123.63254  
## 464 127.8 118.98344  
## 465 100.2 116.34850  
## 466 53.6 71.44988  
## 467 108.4 126.58537  
## 468 120.0 176.89322  
## 469 106.8 118.40670  
## 470 103.0 97.24998  
## 471 89.6 90.40987  
## 472 88.4 71.51656  
## 473 48.2 74.84639  
## 474 103.4 97.20345  
## 475 125.6 162.44050  
## 476 100.0 123.48539  
## 477 98.2 90.60911  
## 478 92.4 81.07126  
## 479 80.8 81.88898  
## 480 51.8 65.15025  
## 481 111.2 97.62394  
## 482 108.4 115.48003  
## 483 105.8 112.94401  
## 484 103.0 97.54737  
## 485 106.4 94.70315  
## 486 78.8 100.38072  
## 487 53.6 63.07737  
## 488 78.4 108.50970  
## 489 106.4 136.05806  
## 490 115.2 103.01508  
## 491 108.6 107.13335  
## 492 96.2 100.21638  
## 493 82.0 77.63520  
## 494 52.8 64.35359  
## 495 100.8 100.11177  
## 496 107.6 109.04757  
## 497 76.2 108.87780

### h1 10 h2 5

set.seed(12345)  
threeInputModel.h1.10.h2.5.TRUE.nn<-neuralnet(tPlus1~tMinus2+tMinus1+t,  
 data = threeInput\_train,hidden = c(10,5),  
 act.fct = "logistic",linear.output = TRUE)  
plot(threeInputModel.h1.10.h2.5.TRUE.nn, rep = "best")



threeInputModel.h1.10.h2.5.TRUE.nn\_Result<-compute(threeInputModel.h1.10.h2.5.TRUE.nn,threeInput\_test[1:3])  
  
  
predicted\_tomorrow<-threeInputModel.h1.10.h2.5.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,threeInput\_test$tPlus1)

## [,1]  
## [1,] 0.792176

threeInput\_train<-threeInput\_norm[1:427,]  
threeInput\_test<-threeInput\_norm[428:497,]  
  
  
threeInput\_train\_original<-as.data.frame(threeInput[1:427,"tPlus1"])  
  
  
threeInput\_test\_original<-as.data.frame(threeInput[428:497,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(threeInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(threeInput\_train\_original)  
#tomorrow\_max  
  
  
head(threeInput\_train\_original)

## tPlus1  
## 1 109.0  
## 2 102.4  
## 3 87.2  
## 4 67.6  
## 5 116.2  
## 6 116.0

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 428 99.18077  
## 429 98.28410  
## 430 94.11057  
## 431 61.86398  
## 432 108.55584  
## 433 115.78830  
## 434 110.38910  
## 435 113.99071  
## 436 111.62277  
## 437 108.38860  
## 438 63.51736  
## 439 121.04131  
## 440 130.55348  
## 441 126.42593  
## 442 117.94681  
## 443 110.82313  
## 444 113.84163  
## 445 66.22916  
## 446 118.38618  
## 447 129.58981  
## 448 127.03427  
## 449 104.60852  
## 450 115.80953  
## 451 101.86946  
## 452 96.61073  
## 453 112.72286  
## 454 138.21947  
## 455 132.48866  
## 456 119.71844  
## 457 129.12671  
## 458 112.73243  
## 459 75.24940  
## 460 121.15406  
## 461 133.88401  
## 462 128.47334  
## 463 122.81549  
## 464 119.44034  
## 465 116.84398  
## 466 68.84720  
## 467 118.74281  
## 468 160.05906  
## 469 116.26582  
## 470 97.11413  
## 471 94.43389  
## 472 75.00240  
## 473 79.21938  
## 474 103.72290  
## 475 152.39928  
## 476 115.57280  
## 477 90.59619  
## 478 89.25702  
## 479 84.63052  
## 480 67.37584  
## 481 100.33471  
## 482 114.35517  
## 483 113.16297  
## 484 96.81303  
## 485 94.35452  
## 486 97.56757  
## 487 59.09769  
## 488 113.82879  
## 489 165.15034  
## 490 105.78513  
## 491 105.61527  
## 492 99.05765  
## 493 81.62498  
## 494 68.71155  
## 495 102.61774  
## 496 104.33979  
## 497 107.40699

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 16.2137

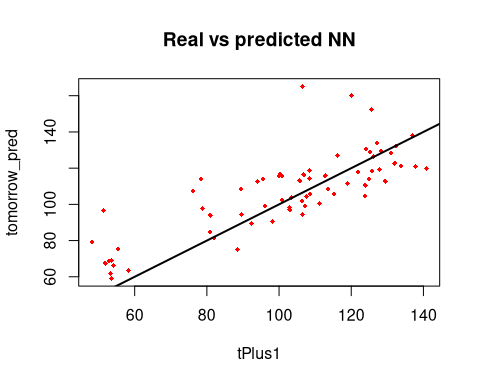
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 12.03248

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 0.1388687

plot(data.frame(threeInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)

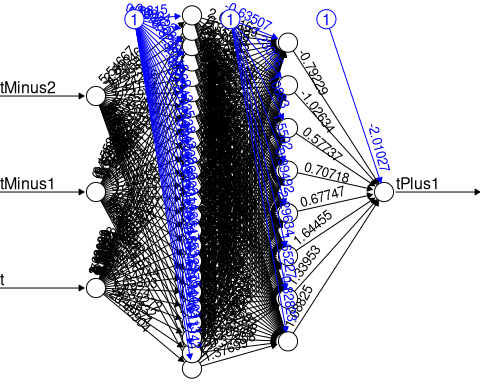


final\_result <- cbind(threeInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 428 107.2 99.18077  
## 429 103.0 98.28410  
## 430 81.0 94.11057  
## 431 53.2 61.86398  
## 432 113.6 108.55584  
## 433 100.8 115.78830  
## 434 124.0 110.38910  
## 435 124.8 113.99071  
## 436 119.0 111.62277  
## 437 89.4 108.38860  
## 438 58.2 63.51736  
## 439 133.8 121.04131  
## 440 124.0 130.55348  
## 441 126.2 126.42593  
## 442 121.8 117.94681  
## 443 123.8 110.82313  
## 444 95.6 113.84163  
## 445 54.2 66.22916  
## 446 125.8 118.38618  
## 447 128.2 129.58981  
## 448 116.2 127.03427  
## 449 123.8 104.60852  
## 450 112.8 115.80953  
## 451 106.4 101.86946  
## 452 51.4 96.61073  
## 453 129.4 112.72286  
## 454 137.0 138.21947  
## 455 132.4 132.48866  
## 456 140.8 119.71844  
## 457 125.2 129.12671  
## 458 94.0 112.73243  
## 459 55.4 75.24940  
## 460 137.8 121.15406  
## 461 127.2 133.88401  
## 462 131.0 128.47334  
## 463 132.0 122.81549  
## 464 127.8 119.44034  
## 465 100.2 116.84398  
## 466 53.6 68.84720  
## 467 108.4 118.74281  
## 468 120.0 160.05906  
## 469 106.8 116.26582  
## 470 103.0 97.11413  
## 471 89.6 94.43389  
## 472 88.4 75.00240  
## 473 48.2 79.21938  
## 474 103.4 103.72290  
## 475 125.6 152.39928  
## 476 100.0 115.57280  
## 477 98.2 90.59619  
## 478 92.4 89.25702  
## 479 80.8 84.63052  
## 480 51.8 67.37584  
## 481 111.2 100.33471  
## 482 108.4 114.35517  
## 483 105.8 113.16297  
## 484 103.0 96.81303  
## 485 106.4 94.35452  
## 486 78.8 97.56757  
## 487 53.6 59.09769  
## 488 78.4 113.82879  
## 489 106.4 165.15034  
## 490 115.2 105.78513  
## 491 108.6 105.61527  
## 492 96.2 99.05765  
## 493 82.0 81.62498  
## 494 52.8 68.71155  
## 495 100.8 102.61774  
## 496 107.6 104.33979  
## 497 76.2 107.40699

### 3input h1 24 h2 8

set.seed(12345)  
  
threeInputModel.h1.24.h2.8.TRUE.nn<-neuralnet(tPlus1~tMinus2+tMinus1+t,  
 data = threeInput\_train,hidden = c(24,8),  
 act.fct = "logistic",linear.output = TRUE)  
  
plot(threeInputModel.h1.24.h2.8.TRUE.nn, rep = "best")



threeInputModel.h1.24.h2.8.TRUE.nn\_Result<-compute(threeInputModel.h1.24.h2.8.TRUE.nn,threeInput\_test[1:3])  
  
  
predicted\_tomorrow<-threeInputModel.h1.24.h2.8.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,threeInput\_test$tPlus1)

## [,1]  
## [1,] 0.7643925

threeInput\_train<-threeInput\_norm[1:427,]  
threeInput\_test<-threeInput\_norm[428:497,]  
  
  
threeInput\_train\_original<-as.data.frame(threeInput[1:427,"tPlus1"])  
  
  
threeInput\_test\_original<-as.data.frame(threeInput[428:497,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(threeInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(threeInput\_train\_original)  
#tomorrow\_max  
  
  
head(threeInput\_train\_original)

## tPlus1  
## 1 109.0  
## 2 102.4  
## 3 87.2  
## 4 67.6  
## 5 116.2  
## 6 116.0

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 428 98.96335  
## 429 98.68424  
## 430 93.95552  
## 431 63.38885  
## 432 110.26899  
## 433 116.49625  
## 434 111.99178  
## 435 112.38150  
## 436 111.83557  
## 437 108.38671  
## 438 64.99164  
## 439 112.11756  
## 440 132.52475  
## 441 128.68155  
## 442 118.31049  
## 443 110.79739  
## 444 113.60249  
## 445 66.25791  
## 446 116.17700  
## 447 130.61096  
## 448 124.27333  
## 449 104.39754  
## 450 116.25129  
## 451 101.91096  
## 452 94.84304  
## 453 116.60927  
## 454 138.98796  
## 455 131.46438  
## 456 119.51413  
## 457 126.41269  
## 458 114.47114  
## 459 70.97358  
## 460 116.16040  
## 461 138.21473  
## 462 133.36581  
## 463 123.06740  
## 464 118.08371  
## 465 116.86173  
## 466 69.73728  
## 467 126.91015  
## 468 159.44185  
## 469 115.57853  
## 470 96.81234  
## 471 93.46303  
## 472 74.02250  
## 473 81.66844  
## 474 110.58062  
## 475 154.96793  
## 476 118.93999  
## 477 90.43021  
## 478 64.32694  
## 479 84.12905  
## 480 69.80563  
## 481 101.01836  
## 482 115.08846  
## 483 107.31232  
## 484 96.99495  
## 485 94.56259  
## 486 98.97954  
## 487 60.44808  
## 488 116.71703  
## 489 169.88193  
## 490 100.58105  
## 491 115.83588  
## 492 99.06369  
## 493 79.62869  
## 494 62.75588  
## 495 102.82489  
## 496 103.58798  
## 497 105.21352

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 17.30199

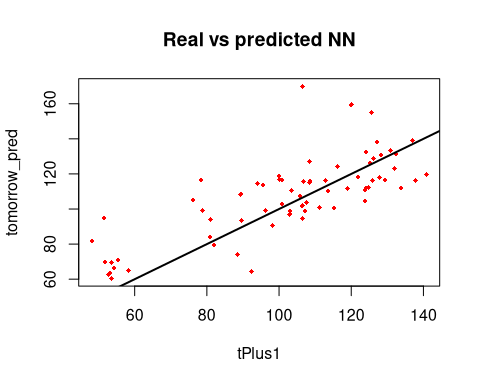
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 13.10563

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 0.1488524

plot(data.frame(threeInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



final\_result <- cbind(threeInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 428 107.2 98.96335  
## 429 103.0 98.68424  
## 430 81.0 93.95552  
## 431 53.2 63.38885  
## 432 113.6 110.26899  
## 433 100.8 116.49625  
## 434 124.0 111.99178  
## 435 124.8 112.38150  
## 436 119.0 111.83557  
## 437 89.4 108.38671  
## 438 58.2 64.99164  
## 439 133.8 112.11756  
## 440 124.0 132.52475  
## 441 126.2 128.68155  
## 442 121.8 118.31049  
## 443 123.8 110.79739  
## 444 95.6 113.60249  
## 445 54.2 66.25791  
## 446 125.8 116.17700  
## 447 128.2 130.61096  
## 448 116.2 124.27333  
## 449 123.8 104.39754  
## 450 112.8 116.25129  
## 451 106.4 101.91096  
## 452 51.4 94.84304  
## 453 129.4 116.60927  
## 454 137.0 138.98796  
## 455 132.4 131.46438  
## 456 140.8 119.51413  
## 457 125.2 126.41269  
## 458 94.0 114.47114  
## 459 55.4 70.97358  
## 460 137.8 116.16040  
## 461 127.2 138.21473  
## 462 131.0 133.36581  
## 463 132.0 123.06740  
## 464 127.8 118.08371  
## 465 100.2 116.86173  
## 466 53.6 69.73728  
## 467 108.4 126.91015  
## 468 120.0 159.44185  
## 469 106.8 115.57853  
## 470 103.0 96.81234  
## 471 89.6 93.46303  
## 472 88.4 74.02250  
## 473 48.2 81.66844  
## 474 103.4 110.58062  
## 475 125.6 154.96793  
## 476 100.0 118.93999  
## 477 98.2 90.43021  
## 478 92.4 64.32694  
## 479 80.8 84.12905  
## 480 51.8 69.80563  
## 481 111.2 101.01836  
## 482 108.4 115.08846  
## 483 105.8 107.31232  
## 484 103.0 96.99495  
## 485 106.4 94.56259  
## 486 78.8 98.97954  
## 487 53.6 60.44808  
## 488 78.4 116.71703  
## 489 106.4 169.88193  
## 490 115.2 100.58105  
## 491 108.6 115.83588  
## 492 96.2 99.06369  
## 493 82.0 79.62869  
## 494 52.8 62.75588  
## 495 100.8 102.82489  
## 496 107.6 103.58798  
## 497 76.2 105.21352

### 3 input h1 23 h2 3

set.seed(12345)  
threeInputModel.h1.23.h2.3.TRUE.nn<-neuralnet(tPlus1~tMinus2+tMinus1+t,  
 data = threeInput\_train,hidden = c(12,4),  
 act.fct = "logistic",linear.output = TRUE)  
  
plot(threeInputModel.h1.23.h2.3.TRUE.nn)  
  
  
threeInputModel.h1.23.h2.3.TRUE.nn\_Result<-compute(threeInputModel.h1.23.h2.3.TRUE.nn,threeInput\_test[1:3])  
  
  
predicted\_tomorrow<-threeInputModel.h1.23.h2.3.TRUE.nn\_Result$net.result  
  
cor(predicted\_tomorrow,threeInput\_test$tPlus1)

## [,1]  
## [1,] 0.8274499

threeInput\_train<-threeInput\_norm[1:427,]  
threeInput\_test<-threeInput\_norm[428:497,]  
  
  
threeInput\_train\_original<-as.data.frame(threeInput[1:427,"tPlus1"])  
  
  
threeInput\_test\_original<-as.data.frame(threeInput[428:497,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(threeInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(threeInput\_train\_original)  
#tomorrow\_max  
  
  
head(threeInput\_train\_original)

## tPlus1  
## 1 109.0  
## 2 102.4  
## 3 87.2  
## 4 67.6  
## 5 116.2  
## 6 116.0

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 428 99.47278  
## 429 98.47734  
## 430 93.19302  
## 431 62.11616  
## 432 98.20010  
## 433 120.52915  
## 434 108.69555  
## 435 116.82425  
## 436 112.16485  
## 437 108.54875  
## 438 63.47589  
## 439 123.15205  
## 440 134.01458  
## 441 124.95512  
## 442 121.38441  
## 443 109.87939  
## 444 114.20502  
## 445 68.17857  
## 446 122.80593  
## 447 132.01803  
## 448 126.64241  
## 449 103.54020  
## 450 119.37832  
## 451 102.64180  
## 452 96.77953  
## 453 114.32561  
## 454 139.75721  
## 455 133.13524  
## 456 117.25497  
## 457 125.26587  
## 458 114.39688  
## 459 69.99998  
## 460 126.21422  
## 461 139.67750  
## 462 128.25392  
## 463 125.22027  
## 464 114.45486  
## 465 115.84556  
## 466 70.53998  
## 467 123.63889  
## 468 157.51434  
## 469 117.51196  
## 470 98.22022  
## 471 95.50873  
## 472 73.85360  
## 473 82.16898  
## 474 107.59445  
## 475 147.14736  
## 476 118.21006  
## 477 99.33138  
## 478 87.54562  
## 479 83.29668  
## 480 71.55776  
## 481 100.33859  
## 482 119.35890  
## 483 111.89082  
## 484 96.58146  
## 485 93.60483  
## 486 97.98553  
## 487 66.88342  
## 488 105.21618  
## 489 117.59405  
## 490 101.34052  
## 491 111.41869  
## 492 99.52737  
## 493 78.53056  
## 494 65.57754  
## 495 101.45105  
## 496 109.68835  
## 497 109.38288

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 14.895

MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

## [1] 11.806

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(threeInput\_test\_original)))

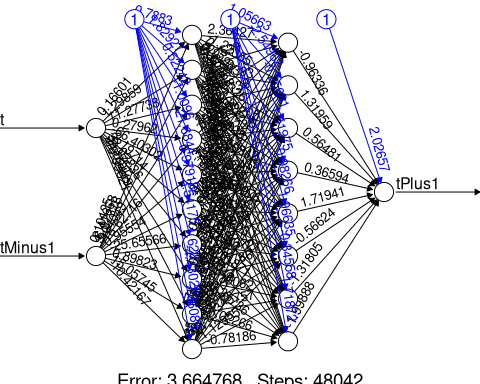
## [1] 0.1381211

plot(data.frame(threeInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)  
  
  
final\_result <- cbind(threeInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 428 107.2 99.47278  
## 429 103.0 98.47734  
## 430 81.0 93.19302  
## 431 53.2 62.11616  
## 432 113.6 98.20010  
## 433 100.8 120.52915  
## 434 124.0 108.69555  
## 435 124.8 116.82425  
## 436 119.0 112.16485  
## 437 89.4 108.54875  
## 438 58.2 63.47589  
## 439 133.8 123.15205  
## 440 124.0 134.01458  
## 441 126.2 124.95512  
## 442 121.8 121.38441  
## 443 123.8 109.87939  
## 444 95.6 114.20502  
## 445 54.2 68.17857  
## 446 125.8 122.80593  
## 447 128.2 132.01803  
## 448 116.2 126.64241  
## 449 123.8 103.54020  
## 450 112.8 119.37832  
## 451 106.4 102.64180  
## 452 51.4 96.77953  
## 453 129.4 114.32561  
## 454 137.0 139.75721  
## 455 132.4 133.13524  
## 456 140.8 117.25497  
## 457 125.2 125.26587  
## 458 94.0 114.39688  
## 459 55.4 69.99998  
## 460 137.8 126.21422  
## 461 127.2 139.67750  
## 462 131.0 128.25392  
## 463 132.0 125.22027  
## 464 127.8 114.45486  
## 465 100.2 115.84556  
## 466 53.6 70.53998  
## 467 108.4 123.63889  
## 468 120.0 157.51434  
## 469 106.8 117.51196  
## 470 103.0 98.22022  
## 471 89.6 95.50873  
## 472 88.4 73.85360  
## 473 48.2 82.16898  
## 474 103.4 107.59445  
## 475 125.6 147.14736  
## 476 100.0 118.21006  
## 477 98.2 99.33138  
## 478 92.4 87.54562  
## 479 80.8 83.29668  
## 480 51.8 71.55776  
## 481 111.2 100.33859  
## 482 108.4 119.35890  
## 483 105.8 111.89082  
## 484 103.0 96.58146  
## 485 106.4 93.60483  
## 486 78.8 97.98553  
## 487 53.6 66.88342  
## 488 78.4 105.21618  
## 489 106.4 117.59405  
## 490 115.2 101.34052  
## 491 108.6 111.41869  
## 492 96.2 99.52737  
## 493 82.0 78.53056  
## 494 52.8 65.57754  
## 495 100.8 101.45105  
## 496 107.6 109.68835  
## 497 76.2 109.38288

## 2 input t and t-1

set.seed(12345)  
twoInput\_model.h1.10.h2.8.nn<-neuralnet(tPlus1~t+tMinus1,data = twoInput\_train, hidden = c(10,8),act.fct = "logistic",linear.output = TRUE)  
  
plot(twoInput\_model.h1.10.h2.8.nn, rep = "best")



twoInput\_model.h1.10.h2.8.nn\_Result<-compute(twoInput\_model.h1.10.h2.8.nn,twoInput\_test[1:2])  
  
  
predicted\_tomorrow<-twoInput\_model.h1.10.h2.8.nn\_Result$net.result  
  
cor(predicted\_tomorrow,twoInput\_test$tPlus1)

## [,1]  
## [1,] 0.1761059

twoInput\_train<-twoInput\_norm[1:428,]  
twoInput\_test<-twoInput\_norm[429:498,]  
  
  
twoInput\_train\_original<-as.data.frame(twoInput[1:428,"tPlus1"])  
  
  
twoInput\_test\_original<-as.data.frame(twoInput[429:498,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(twoInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(twoInput\_train\_original)  
#tomorrow\_max  
  
  
head(twoInput\_train\_original)

## tPlus1  
## 1 114.8  
## 2 109.0  
## 3 102.4  
## 4 87.2  
## 5 67.6  
## 6 116.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 429 105.07508  
## 430 101.86644  
## 431 100.97215  
## 432 101.32955  
## 433 82.78957  
## 434 127.65491  
## 435 106.48714  
## 436 75.36306  
## 437 115.11881  
## 438 115.23672  
## 439 112.98346  
## 440 95.20900  
## 441 107.61771  
## 442 123.19028  
## 443 115.31634  
## 444 116.58034  
## 445 113.24593  
## 446 116.10812  
## 447 91.55484  
## 448 111.97387  
## 449 117.04623  
## 450 118.07925  
## 451 108.75523  
## 452 114.30773  
## 453 105.33203  
## 454 99.44577  
## 455 110.67617  
## 456 121.59430  
## 457 126.75749  
## 458 124.79490  
## 459 129.53307  
## 460 117.65280  
## 461 93.02343  
## 462 105.24057  
## 463 126.91213  
## 464 118.63631  
## 465 121.80290  
## 466 121.98687  
## 467 119.03098  
## 468 99.52555  
## 469 125.58044  
## 470 101.20736  
## 471 111.33684  
## 472 100.64276  
## 473 99.37978  
## 474 86.39936  
## 475 116.90022  
## 476 105.74265  
## 477 80.33023  
## 478 117.04787  
## 479 95.36566  
## 480 94.60417  
## 481 92.46209  
## 482 80.12863  
## 483 122.30526  
## 484 103.94230  
## 485 101.76672  
## 486 99.81703  
## 487 97.21555  
## 488 104.70572  
## 489 76.56803  
## 490 112.31261  
## 491 65.21742  
## 492 99.73975  
## 493 107.20382  
## 494 102.93093  
## 495 95.20040  
## 496 86.11744  
## 497 109.50402  
## 498 95.04452

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 26.88411

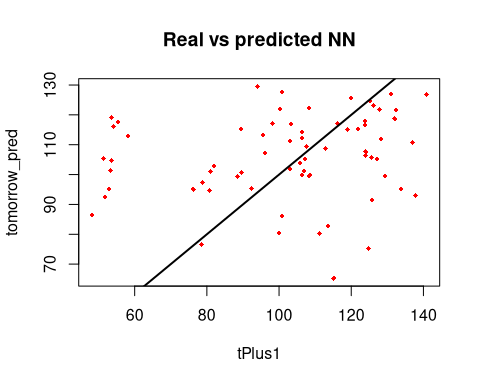
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 20.42569

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 0.2581478

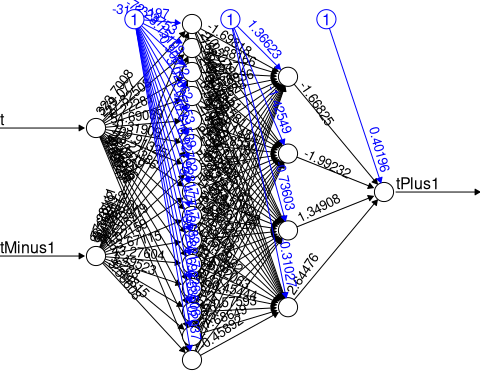
plot(data.frame(twoInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



final\_result <- cbind(twoInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 429 107.2 105.07508  
## 430 103.0 101.86644  
## 431 81.0 100.97215  
## 432 53.2 101.32955  
## 433 113.6 82.78957  
## 434 100.8 127.65491  
## 435 124.0 106.48714  
## 436 124.8 75.36306  
## 437 119.0 115.11881  
## 438 89.4 115.23672  
## 439 58.2 112.98346  
## 440 133.8 95.20900  
## 441 124.0 107.61771  
## 442 126.2 123.19028  
## 443 121.8 115.31634  
## 444 123.8 116.58034  
## 445 95.6 113.24593  
## 446 54.2 116.10812  
## 447 125.8 91.55484  
## 448 128.2 111.97387  
## 449 116.2 117.04623  
## 450 123.8 118.07925  
## 451 112.8 108.75523  
## 452 106.4 114.30773  
## 453 51.4 105.33203  
## 454 129.4 99.44577  
## 455 137.0 110.67617  
## 456 132.4 121.59430  
## 457 140.8 126.75749  
## 458 125.2 124.79490  
## 459 94.0 129.53307  
## 460 55.4 117.65280  
## 461 137.8 93.02343  
## 462 127.2 105.24057  
## 463 131.0 126.91213  
## 464 132.0 118.63631  
## 465 127.8 121.80290  
## 466 100.2 121.98687  
## 467 53.6 119.03098  
## 468 108.4 99.52555  
## 469 120.0 125.58044  
## 470 106.8 101.20736  
## 471 103.0 111.33684  
## 472 89.6 100.64276  
## 473 88.4 99.37978  
## 474 48.2 86.39936  
## 475 103.4 116.90022  
## 476 125.6 105.74265  
## 477 100.0 80.33023  
## 478 98.2 117.04787  
## 479 92.4 95.36566  
## 480 80.8 94.60417  
## 481 51.8 92.46209  
## 482 111.2 80.12863  
## 483 108.4 122.30526  
## 484 105.8 103.94230  
## 485 103.0 101.76672  
## 486 106.4 99.81703  
## 487 78.8 97.21555  
## 488 53.6 104.70572  
## 489 78.4 76.56803  
## 490 106.4 112.31261  
## 491 115.2 65.21742  
## 492 108.6 99.73975  
## 493 96.2 107.20382  
## 494 82.0 102.93093  
## 495 52.8 95.20040  
## 496 100.8 86.11744  
## 497 107.6 109.50402  
## 498 76.2 95.04452

set.seed(12345)  
twoInput\_model.h1.15.h2.4.nn<-neuralnet(tPlus1~t+tMinus1,data = twoInput\_train, hidden = c(15,4),act.fct = "logistic",linear.output = TRUE)  
  
plot(twoInput\_model.h1.15.h2.4.nn, rep = "best")



twoInput\_model.h1.15.h2.4.nn\_Result<-compute(twoInput\_model.h1.15.h2.4.nn,twoInput\_test[1:2])  
  
  
predicted\_tomorrow<-twoInput\_model.h1.15.h2.4.nn\_Result$net.result  
  
cor(predicted\_tomorrow,twoInput\_test$tPlus1)

## [,1]  
## [1,] 0.2835334

twoInput\_train<-twoInput\_norm[1:428,]  
twoInput\_test<-twoInput\_norm[429:498,]  
  
  
twoInput\_train\_original<-as.data.frame(twoInput[1:428,"tPlus1"])  
  
  
twoInput\_test\_original<-as.data.frame(twoInput[429:498,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(twoInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(twoInput\_train\_original)  
#tomorrow\_max  
  
  
head(twoInput\_train\_original)

## tPlus1  
## 1 114.8  
## 2 109.0  
## 3 102.4  
## 4 87.2  
## 5 67.6  
## 6 116.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 429 105.72370  
## 430 102.34485  
## 431 101.19042  
## 432 99.94561  
## 433 117.21049  
## 434 107.51283  
## 435 107.02497  
## 436 74.90570  
## 437 115.00287  
## 438 115.48231  
## 439 113.56790  
## 440 89.31934  
## 441 117.48899  
## 442 122.83362  
## 443 115.09357  
## 444 116.62134  
## 445 113.23671  
## 446 116.95797  
## 447 100.51308  
## 448 147.68710  
## 449 116.65637  
## 450 118.31207  
## 451 108.89672  
## 452 114.86894  
## 453 105.96567  
## 454 125.89774  
## 455 138.01977  
## 456 120.69659  
## 457 126.17143  
## 458 123.92284  
## 459 128.79328  
## 460 118.52144  
## 461 97.26915  
## 462 117.56174  
## 463 126.32951  
## 464 118.04404  
## 465 121.18435  
## 466 121.55854  
## 467 119.76443  
## 468 113.35808  
## 469 137.24644  
## 470 102.06627  
## 471 112.11538  
## 472 100.83569  
## 473 98.22298  
## 474 85.81149  
## 475 28.19108  
## 476 123.35303  
## 477 79.25103  
## 478 117.84807  
## 479 94.76664  
## 480 93.37248  
## 481 91.81139  
## 482 121.49900  
## 483 134.43988  
## 484 104.54442  
## 485 102.18649  
## 486 99.94830  
## 487 97.49280  
## 488 103.64591  
## 489 82.02807  
## 490 113.20105  
## 491 63.53334  
## 492 100.54638  
## 493 107.91888  
## 494 102.84304  
## 495 93.83890  
## 496 122.36818  
## 497 126.80247  
## 498 95.50976

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 27.33963

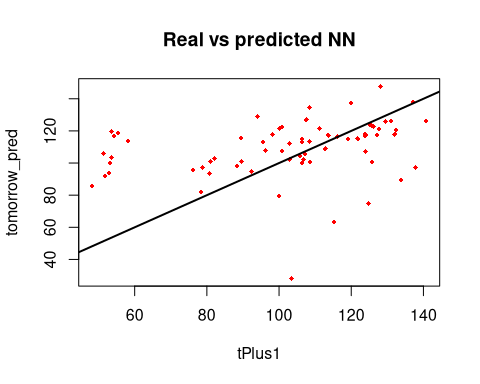
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 19.65625

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 0.2531819

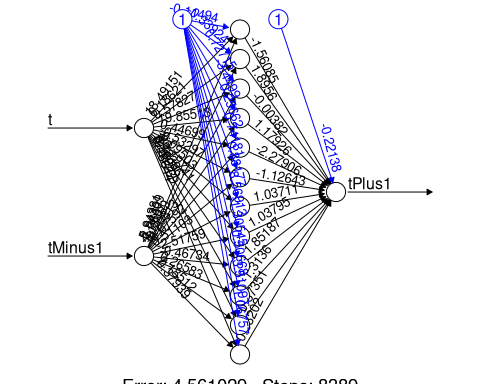
plot(data.frame(twoInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



final\_result <- cbind(twoInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 429 107.2 105.72370  
## 430 103.0 102.34485  
## 431 81.0 101.19042  
## 432 53.2 99.94561  
## 433 113.6 117.21049  
## 434 100.8 107.51283  
## 435 124.0 107.02497  
## 436 124.8 74.90570  
## 437 119.0 115.00287  
## 438 89.4 115.48231  
## 439 58.2 113.56790  
## 440 133.8 89.31934  
## 441 124.0 117.48899  
## 442 126.2 122.83362  
## 443 121.8 115.09357  
## 444 123.8 116.62134  
## 445 95.6 113.23671  
## 446 54.2 116.95797  
## 447 125.8 100.51308  
## 448 128.2 147.68710  
## 449 116.2 116.65637  
## 450 123.8 118.31207  
## 451 112.8 108.89672  
## 452 106.4 114.86894  
## 453 51.4 105.96567  
## 454 129.4 125.89774  
## 455 137.0 138.01977  
## 456 132.4 120.69659  
## 457 140.8 126.17143  
## 458 125.2 123.92284  
## 459 94.0 128.79328  
## 460 55.4 118.52144  
## 461 137.8 97.26915  
## 462 127.2 117.56174  
## 463 131.0 126.32951  
## 464 132.0 118.04404  
## 465 127.8 121.18435  
## 466 100.2 121.55854  
## 467 53.6 119.76443  
## 468 108.4 113.35808  
## 469 120.0 137.24644  
## 470 106.8 102.06627  
## 471 103.0 112.11538  
## 472 89.6 100.83569  
## 473 88.4 98.22298  
## 474 48.2 85.81149  
## 475 103.4 28.19108  
## 476 125.6 123.35303  
## 477 100.0 79.25103  
## 478 98.2 117.84807  
## 479 92.4 94.76664  
## 480 80.8 93.37248  
## 481 51.8 91.81139  
## 482 111.2 121.49900  
## 483 108.4 134.43988  
## 484 105.8 104.54442  
## 485 103.0 102.18649  
## 486 106.4 99.94830  
## 487 78.8 97.49280  
## 488 53.6 103.64591  
## 489 78.4 82.02807  
## 490 106.4 113.20105  
## 491 115.2 63.53334  
## 492 108.6 100.54638  
## 493 96.2 107.91888  
## 494 82.0 102.84304  
## 495 52.8 93.83890  
## 496 100.8 122.36818  
## 497 107.6 126.80247  
## 498 76.2 95.50976

set.seed(12345)  
twoInput\_model.h1.12.nn<-neuralnet(tPlus1~t+tMinus1,data = twoInput\_train, hidden = c(12),act.fct = "logistic",linear.output = TRUE)  
  
plot(twoInput\_model.h1.12.nn, rep = "best")



twoInput\_model.h1.12.nn\_Result<-compute(twoInput\_model.h1.12.nn,twoInput\_test[1:2])  
  
  
predicted\_tomorrow<-twoInput\_model.h1.12.nn\_Result$net.result  
  
cor(predicted\_tomorrow,twoInput\_test$tPlus1)

## [,1]  
## [1,] 0.2468435

twoInput\_train<-twoInput\_norm[1:428,]  
twoInput\_test<-twoInput\_norm[429:498,]  
  
  
twoInput\_train\_original<-as.data.frame(twoInput[1:428,"tPlus1"])  
  
  
twoInput\_test\_original<-as.data.frame(twoInput[429:498,"tPlus1"])  
#twoInput\_test\_original  
  
tomorrow\_min<-min(twoInput\_train\_original)  
#tomorrow\_min  
  
tomorrow\_max<-max(twoInput\_train\_original)  
#tomorrow\_max  
  
  
head(twoInput\_train\_original)

## tPlus1  
## 1 114.8  
## 2 109.0  
## 3 102.4  
## 4 87.2  
## 5 67.6  
## 6 116.2

unnormalize <- function(x, min, max) {   
 return( (max - min)\*x + min )  
}  
  
  
tomorrow\_pred <- unnormalize(predicted\_tomorrow, tomorrow\_min, tomorrow\_max)  
tomorrow\_pred

## [,1]  
## 429 106.72789  
## 430 102.89243  
## 431 102.72229  
## 432 104.28054  
## 433 91.89736  
## 434 123.89741  
## 435 109.64048  
## 436 84.38400  
## 437 115.16716  
## 438 117.54116  
## 439 116.59899  
## 440 97.44913  
## 441 144.38331  
## 442 126.95137  
## 443 114.78426  
## 444 118.52706  
## 445 112.80402  
## 446 120.47096  
## 447 102.74668  
## 448 127.36333  
## 449 116.43844  
## 450 121.97754  
## 451 106.05622  
## 452 117.73825  
## 453 107.52521  
## 454 109.39327  
## 455 117.30432  
## 456 118.52090  
## 457 129.16043  
## 458 121.37397  
## 459 134.63796  
## 460 122.04064  
## 461 101.59131  
## 462 132.13927  
## 463 131.00645  
## 464 117.41091  
## 465 121.90813  
## 466 124.09170  
## 467 123.94425  
## 468 105.50874  
## 469 124.63297  
## 470 97.80182  
## 471 114.85638  
## 472 102.30401  
## 473 102.37122  
## 474 86.33722  
## 475 98.43084  
## 476 112.46505  
## 477 87.68173  
## 478 121.72595  
## 479 96.44169  
## 480 96.54909  
## 481 93.94407  
## 482 92.56995  
## 483 119.92782  
## 484 105.30505  
## 485 103.12038  
## 486 101.24282  
## 487 96.76881  
## 488 107.28726  
## 489 92.63886  
## 490 108.34842  
## 491 62.64595  
## 492 97.31305  
## 493 109.45403  
## 494 105.92222  
## 495 97.91141  
## 496 92.36550  
## 497 121.46762  
## 498 93.38770

RMSE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 26.7221

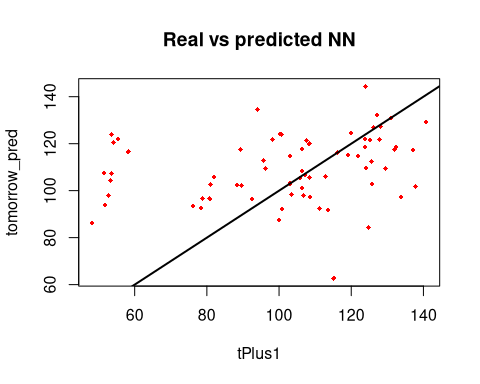
MAE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 19.53432

MAPE(as.numeric(unlist(tomorrow\_pred)),as.numeric(unlist(twoInput\_test\_original)))

## [1] 0.256825

plot(data.frame(twoInput\_test\_original, tomorrow\_pred) ,col='red',main='Real vs predicted NN',pch=18,cex=0.7)  
abline(0,1,lwd=2)



final\_result <- cbind(twoInput\_test\_original, tomorrow\_pred)  
final\_result

## tPlus1 tomorrow\_pred  
## 429 107.2 106.72789  
## 430 103.0 102.89243  
## 431 81.0 102.72229  
## 432 53.2 104.28054  
## 433 113.6 91.89736  
## 434 100.8 123.89741  
## 435 124.0 109.64048  
## 436 124.8 84.38400  
## 437 119.0 115.16716  
## 438 89.4 117.54116  
## 439 58.2 116.59899  
## 440 133.8 97.44913  
## 441 124.0 144.38331  
## 442 126.2 126.95137  
## 443 121.8 114.78426  
## 444 123.8 118.52706  
## 445 95.6 112.80402  
## 446 54.2 120.47096  
## 447 125.8 102.74668  
## 448 128.2 127.36333  
## 449 116.2 116.43844  
## 450 123.8 121.97754  
## 451 112.8 106.05622  
## 452 106.4 117.73825  
## 453 51.4 107.52521  
## 454 129.4 109.39327  
## 455 137.0 117.30432  
## 456 132.4 118.52090  
## 457 140.8 129.16043  
## 458 125.2 121.37397  
## 459 94.0 134.63796  
## 460 55.4 122.04064  
## 461 137.8 101.59131  
## 462 127.2 132.13927  
## 463 131.0 131.00645  
## 464 132.0 117.41091  
## 465 127.8 121.90813  
## 466 100.2 124.09170  
## 467 53.6 123.94425  
## 468 108.4 105.50874  
## 469 120.0 124.63297  
## 470 106.8 97.80182  
## 471 103.0 114.85638  
## 472 89.6 102.30401  
## 473 88.4 102.37122  
## 474 48.2 86.33722  
## 475 103.4 98.43084  
## 476 125.6 112.46505  
## 477 100.0 87.68173  
## 478 98.2 121.72595  
## 479 92.4 96.44169  
## 480 80.8 96.54909  
## 481 51.8 93.94407  
## 482 111.2 92.56995  
## 483 108.4 119.92782  
## 484 105.8 105.30505  
## 485 103.0 103.12038  
## 486 106.4 101.24282  
## 487 78.8 96.76881  
## 488 53.6 107.28726  
## 489 78.4 92.63886  
## 490 106.4 108.34842  
## 491 115.2 62.64595  
## 492 108.6 97.31305  
## 493 96.2 109.45403  
## 494 82.0 105.92222  
## 495 52.8 97.91141  
## 496 100.8 92.36550  
## 497 107.6 121.46762  
## 498 76.2 93.38770