Neural Network Classification Practice

Installing Packages

```
if(!requireNamespace("tidyverse"))install.packages('tidyverse')
```

```
## Loading required namespace: tidyverse
```

```
if(!requireNamespace("caret"))install.packages('caret')
```

```
## Loading required namespace: caret
```

```
if(!requireNamespace("neuralnet"))install.packages('neuralnet')
```

```
## Loading required namespace: neuralnet
```

```
if(!requireNamespace("keras"))install.packages('keras')
```

```
## Loading required namespace: keras
```

```
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'
##
## The following object is masked from 'package:dplyr':
##
##     compute
```

```
library(keras)
```

Reading Data

```
data <- read.csv('banknote.csv')
data <- na.omit(data)
cat("There are", nrow(data), "observations left.")
```

```
## There are 1372 observations left.
```

```
str(data)
```

```
## 'data.frame':    1372 obs. of  5 variables:
##  $ variance: num  3.622 4.546 3.866 3.457 0.329 ...
##  $ skewness: num  8.67 8.17 -2.64 9.52 -4.46 ...
##  $ curtosis: num  -2.81 -2.46 1.92 -4.01 4.57 ...
##  $ entropy : num  -0.447 -1.462 0.106 -3.594 -0.989 ...
##  $ class   : int  1 1 1 1 1 1 1 1 1 1 ...
```

Split train and test data

```
set.seed(123)
training.samples <- data$class %>% createDataPartition(p=0.75,list=FALSE)
train.data <- data[training.samples,]
test.data <- data[-training.samples,]
nrow(train.data)
```

```
## [1] 1029
```

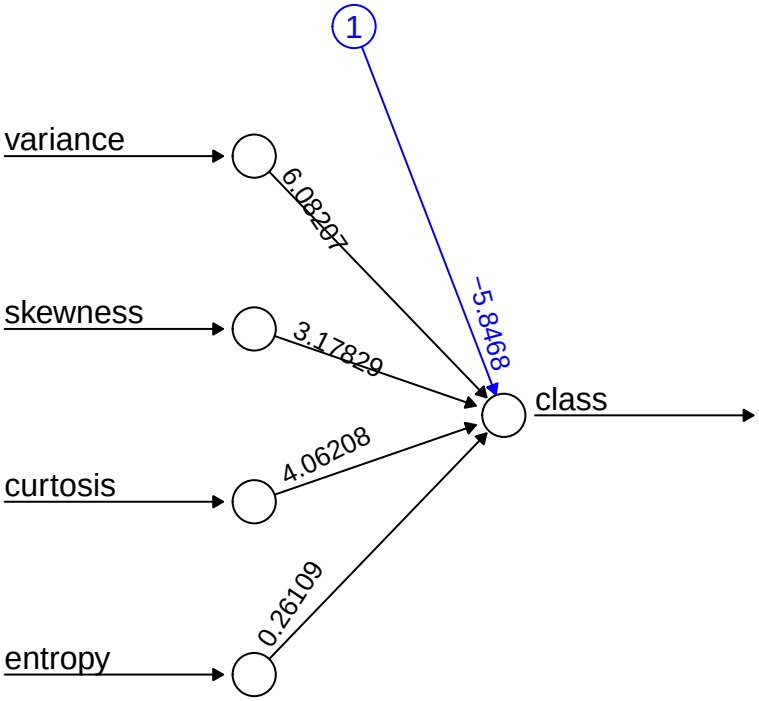
```
nrow(test.data)
```

```
## [1] 343
```

Perception model 1

(i) No hidden layer
(ii) The default loss function of "SSE"
(iii) the default activation function of "logistic"

```
set.seed(123)
model <- neuralnet(class~., data=train.data, hidden = 0, err.fct = "sse", linear.output = FALSE)
plot(model, rep = "best")
```



Error: 3.148348 Steps: 3014

Confusion matrix - Perception model 1

```
probabilities <- model %>% predict(test.data) %>% as.vector()
predicted.class <- ifelse(probabilities>0.5,1,0)
confusionMatrix(factor(predicted.class),factor(test.data$class), positive = "1")
```

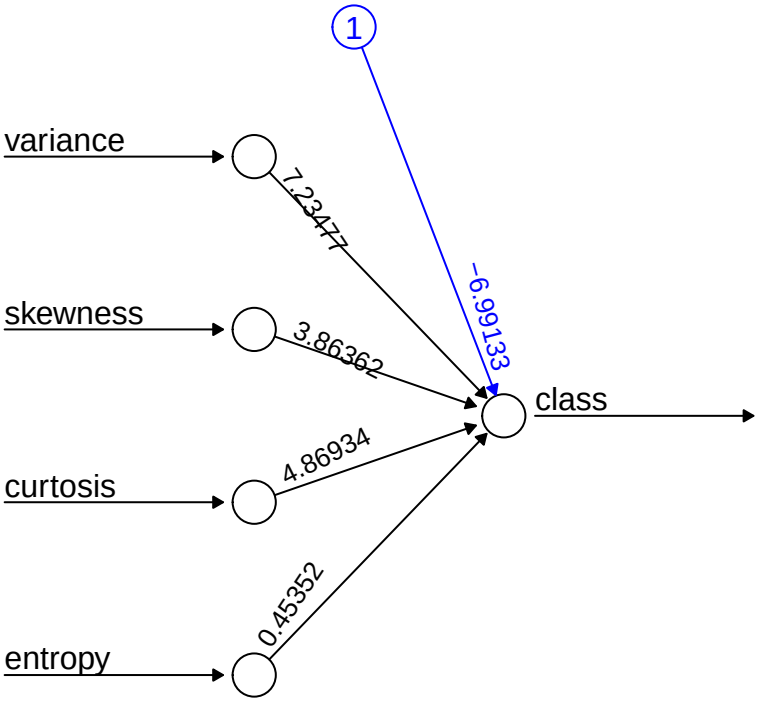
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 160   1
##          1   1 181
##
##                Accuracy : 0.9942
##                  95% CI : (0.9791, 0.9993)
##     No Information Rate : 0.5306
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9883
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9945
##             Specificity : 0.9938
##          Pos Pred Value : 0.9945
##          Neg Pred Value : 0.9938
##              Prevalence : 0.5306
##          Detection Rate : 0.5277
```

```
##    Detection Prevalence : 0.5306
##       Balanced Accuracy : 0.9941
##
##        'Positive' Class : 1
##
```

Perception model 2

(i) No hidden layer
(ii) The loss function of "ce" (namely, cross-entropy)
(iii) the default logistic function of "logistic"

```
set.seed(123)
model <- neuralnet(class~., data=train.data, hidden = 0, err.fct = "ce", linear.output = FALSE)
plot(model, rep = "best")
```



Confusion matrix - Perception model 2

```
probabilities <- model %>% predict(test.data) %>% as.vector()
predicted.class <- ifelse(probabilities>0.5,1,0)
confusionMatrix(factor(predicted.class),factor(test.data$class),positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 160   3
##          1   1 179
```

```
##
##                Accuracy : 0.9883
##                  95% CI : (0.9704, 0.9968)
##     No Information Rate : 0.5306
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9766
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9835
##             Specificity : 0.9938
##          Pos Pred Value : 0.9944
##          Neg Pred Value : 0.9816
##              Prevalence : 0.5306
##          Detection Rate : 0.5219
##    Detection Prevalence : 0.5248
##       Balanced Accuracy : 0.9887
##
##        'Positive' Class : 1
##
```

Logistic regression model

```
model <- glm(class~., family = binomial, data = train.data)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model
```

```
##
## Call:  glm(formula = class ~ ., family = binomial, data = train.data)
##
## Coefficients:
## (Intercept)     variance     skewness     curtosis      entropy
##     -7.0385       7.2886       3.8913       4.9051       0.4592
##
## Degrees of Freedom: 1028 Total (i.e. Null);  1024 Residual
## Null Deviance:       1410
## Residual Deviance: 39.38     AIC: 49.38
```

CE loss function model better ensembles logistic regression model

Confusion matrix - Logistic regression model

```
probabilities <- model %>% predict(test.data) %>% as.vector()
predicted.class <- ifelse(probabilities>0.5,1,0)
confusionMatrix(factor(predicted.class),factor(test.data$class),positive="1")
```

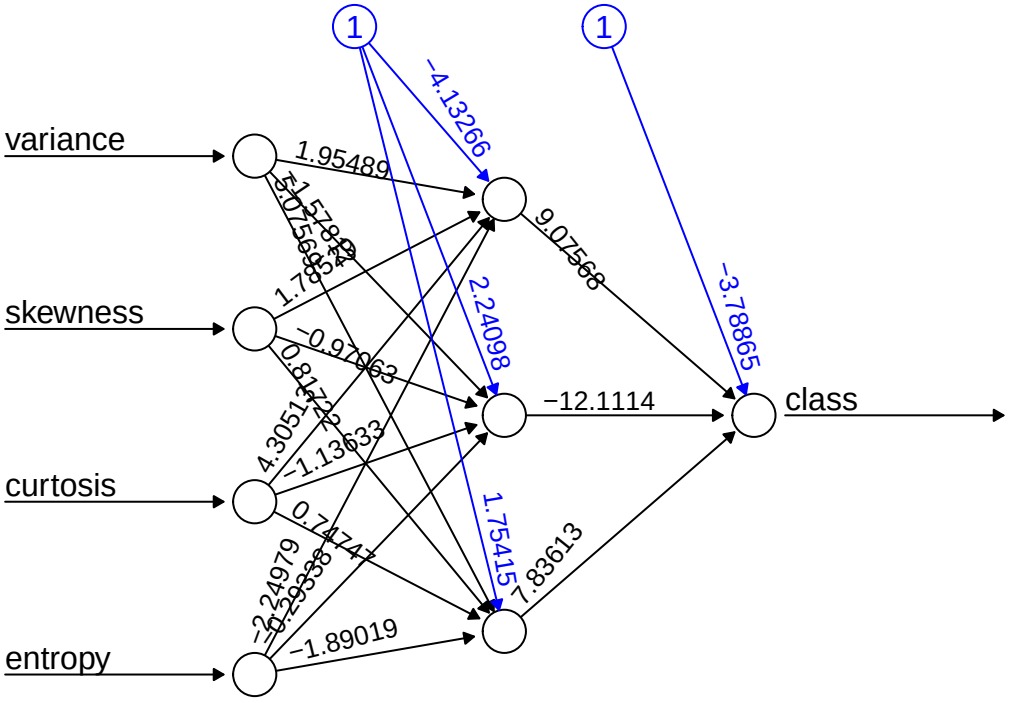
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 160   3
```

```
##          1   1 179
##
##                Accuracy : 0.9883
##                  95% CI : (0.9704, 0.9968)
##     No Information Rate : 0.5306
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9766
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9835
##             Specificity : 0.9938
##          Pos Pred Value : 0.9944
##          Neg Pred Value : 0.9816
##              Prevalence : 0.5306
##          Detection Rate : 0.5219
##    Detection Prevalence : 0.5248
##       Balanced Accuracy : 0.9887
##
##        'Positive' Class : 1
##
```

Perception model 3

(i): one hidden layer with 3 neurons (ii): the default loss function of "sse" (iii): the default activation function of "logistic"

```
set.seed(123)
model <- neuralnet(class~., data=train.data, hidden=3, err.fct="sse",linear.output=FALSE)
plot(model, rep = "best")
```



Error: 0.005289 Steps: 160

Confusion matrix - Perception model 3

```
probabilities <- model %>% predict(test.data) %>% as.vector()
predicted.class <- ifelse(probabilities>0.5,1,0)
confusionMatrix(factor(predicted.class),factor(test.data$class),positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 161   0
##          1   0 182
##
##                Accuracy : 1
##                  95% CI : (0.9893, 1)
##     No Information Rate : 0.5306
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5306
##          Detection Rate : 0.5306
```

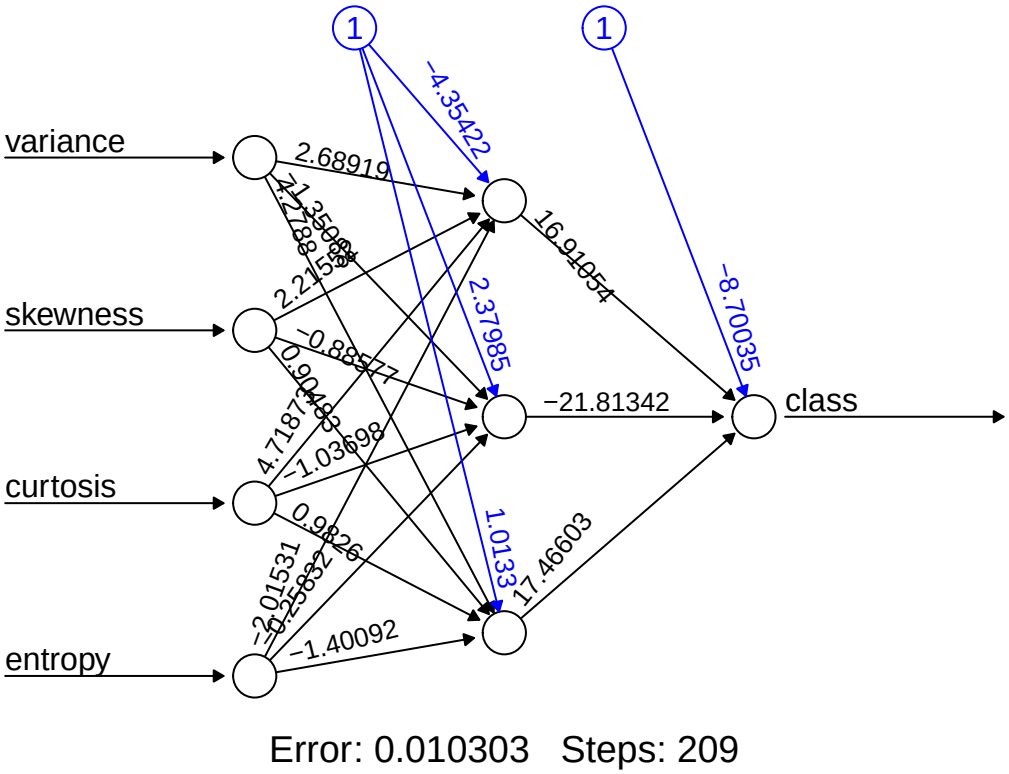
```
##    Detection Prevalence : 0.5306
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 1
##
```

The prediction with hidden layer is better than no hidden layer.

Perception model 4

(i): one hidden layer with 3 neurons (ii): the loss function of "ce" (iii): the default activation function of "logistic"

```
set.seed(123)
model <- neuralnet(class~., data=train.data, hidden=3, err.fct="ce",linear.output=FALSE)
plot(model, rep = "best")
```



Confusion matrix - Perception model 4

```
probabilities <- model %>% predict(test.data) %>% as.vector()
predicted.class <- ifelse(probabilities>0.5,1,0)
confusionMatrix(factor(predicted.class),factor(test.data$class),positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 161   0
##          1   0 182
```

```
##
##                Accuracy : 1
##                  95% CI : (0.9893, 1)
##     No Information Rate : 0.5306
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5306
##          Detection Rate : 0.5306
##    Detection Prevalence : 0.5306
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 1
##
```

The prediction with hidden layer is better than no hidden layer.