Neural Network Classification

Karen Mazidi

Classification on the PimaIndiansDiabetes2 data set in package mlbench.

Load data, divide into train and test

```
library(mlbench)
data("PimaIndiansDiabetes2")
df <- PimaIndiansDiabetes2[complete.cases(PimaIndiansDiabetes2[]),]
set.seed(1234)
i <- sample(1:nrow(df), 0.75*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Logistic regression

Performing logistic regression as a baseline. Accuracy on the test set is .765.

```
glm1 <- glm(diabetes~., data=train, family=binomial)
probs <- predict(glm1, newdata=test, type="response")
pred1 <- ifelse(probs>0.5, "pos", "neg")
mean(pred1==test$diabetes)
```

```
## [1] 0.7653061
```

Preprocessing

We need to scale the predictors.

```
scaled <- scale(df[,-9])
df_scaled <- data.frame(cbind(scaled, df$diabetes))
colnames(df_scaled) <- colnames(df)
df_scaled$diabetes <- factor(df_scaled$diabetes)
train_scaled <- df_scaled[i,]
test_scaled <- df_scaled[-i,]
```

More preprocessing

The neuralnet() function does not handle factors. We have to replace the target factor with a one-hot encoding of diabetes.

```
class_column <- train_scaled$diabetes
n <- length(class_column)
x <- matrix(0, n, length(levels(class_column)))
x[(1L:n) + n * (unclass(class_column) - 1L)] <- 1
dimnames(x) <- list(names(class_column), levels(class_column))
train_scaled <- cbind(train_scaled[,1:8], x)
names(train_scaled) <- c(names(train_scaled)[1:8], "neg", "pos")
```

Create the formula

```
n <- names(train)
f <- as.formula(paste("neg + pos ~", paste(n[!n %in% "diabetes"], collapse = " + ")))
```

Build the neural network

```
library(neuralnet)
set.seed(1234)
nnet1 <- neuralnet(f, data=train_scaled, hidden=c(5,2), threshold=0.1, act.fct = "logistic", linear.outp
```

```
## hidden: 5, 2    thresh: 0.1    rep: 1/1    steps:     269    error: 19.33646 time: 0.31 secs
```

```
plot(nnet1)
```

Evaluate on test set

The mean accuracy was 0.776, one percent higher than logistic regression.

```
nnet.results <- compute(nnet1, test_scaled[,1:8])
pred <- max.col(nnet.results$net.result)
mean(test_scaled$diabetes==pred)
```

```
## [1] 0.7755102041
```