

```
> library("neuralnet")
# Generate 50 random numbers uniformly distributed between 0 and 100 And store them as a data frame
> traininginput <- as.data.frame(runif(50, min=0, max=100))
> trainingoutput <- sqrt(traininginput)
```

Column bind the data into one variable

```
> trainingdata <- cbind(traininginput,trainingoutput)
> colnames(trainingdata) <- c("Input","Output")
train
> net.sqrt <- neuralnet(Output~Input,trainingdata, hidden=10, threshold=0.01)
view
> print(net.sqrt)
```

Test the neural network on some training data

```
testdata <- as.data.frame((1:10)^2)
net.results <- compute(net.sqrt, testdata) #Run them through the neural network
Tipi
> ls(net.results)
```

Sonuçlar

```
> print(net.results$net.result)
```

Sonuçların daha güzel gösterimi

```
> cleanoutput <- cbind(testdata,sqrt(testdata),
+                       as.data.frame(net.results$net.result))
> colnames(cleanoutput) <- c("Input","Expected Output","Neural Net Output")
> print(cleanoutput)
```

```
> data("iris")
> iris.dataset <- iris
> View(iris.dataset)
> iris.dataset$setosa <- iris.dataset$Species=="setosa"
> iris.dataset$virginica = iris.dataset$Species == "virginica"
> iris.dataset$versicolor = iris.dataset$Species == "versicolor"
> View(iris.dataset)
> train <- sample(x = nrow(iris.dataset), size = nrow(iris)*0.5)
> train
> iris.train <- iris.dataset[train,]
> iris.valid <- iris.dataset[-train,]
> nn <- neuralnet(setosa+versicolor+virginica ~ Sepal.Length + Sepal.Width,
data=iris.train, hidden=3,
+               rep = 2, err.fct = "ce", linear.output = F, lifesign = "minimal", stepmax = 1000000)
> plot(nn, rep="best")

> comp <- compute(nn, iris.valid[-3:-8])
> pred.weights <- comp$net.result
> idx <- apply(pred.weights, 1, which.max)
> pred <- c('setosa', 'versicolor', 'virginica')[idx]
> table(pred, iris.valid$Species
+ )
```