

# Implementation of a Neural Network in R

**Potential Scenario:** Students scores in NRES 710 Class

Student with marks less than 50% is considered FAIL (0) and above 50% PASS (1)

Assignments/Labs /100	Final Project /100	Final Grade
20	90	PASS
10	20	FAIL
30	40	FAIL
20	50	FAIL
80	50	PASS
30	80	PASS

# Implementation of a Neural Network in R

# install package

```
install.packages("neuralnet")
```

```
ASL=c(20,10,30,20,80,30) # Assignments/Labs
```

```
FNP=c(90,20,40,50,50,80) # Final Project
```

```
GRD=c(1,0,0,0,1,1) # Final Grade
```

# Combine multiple columns or features into a single set of data

```
df=data.frame(ASL,FNP,GRD)
```

Import the neuralnet library and create NN classifier model by passing argument set of label and features, dataset, number of neurons in hidden layers, and error calculation.

```
require(neuralnet) # load library
```

```
# fit neural network
```

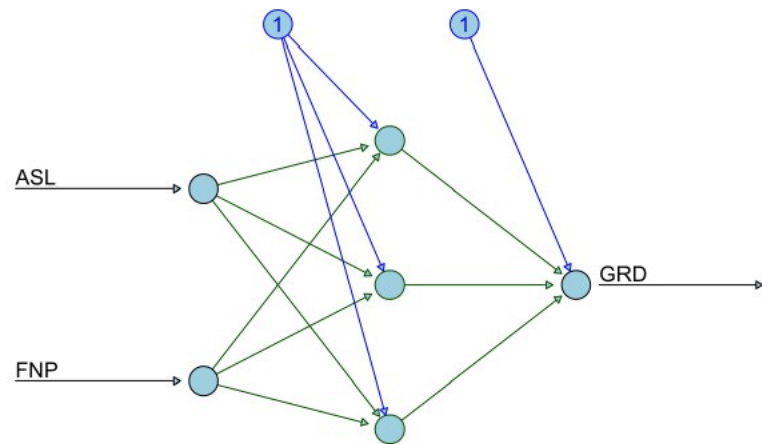
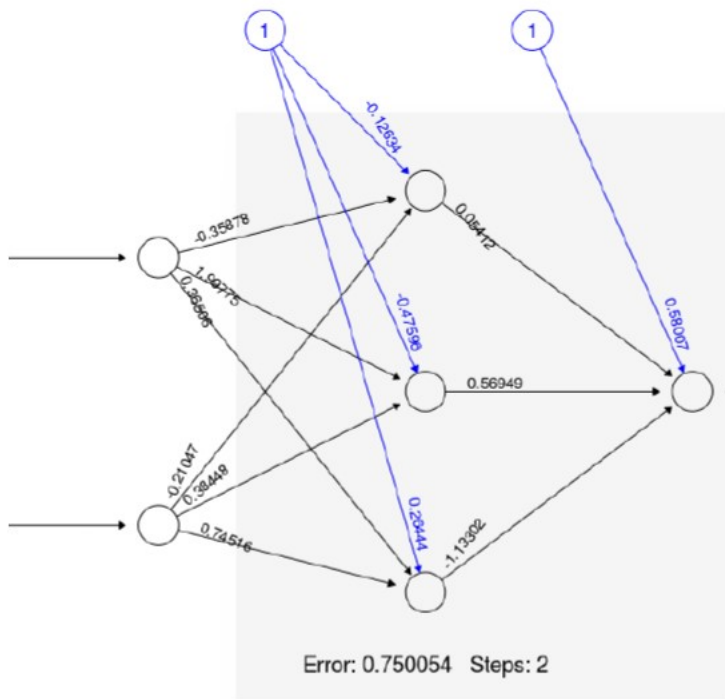
```
nn=neuralnet(GRD~ASL+FNP,data=df, hidden=3,act.fct = "logistic", linear.output = FALSE)
```

Python Users:

```
pip install neuralnetwork
```

# Plotting Neural Network

```
plot(nn,col.hidden = 'darkgreen', col.hidden.synapse = 'darkgreen', show.weights = F, information = F, fill = 'lightblue')
```



# Create Test Dataset and Predict

- `ASL=c(30,40,85)`
- `FNP=c(85,50,40)`
- `test=data.frame(ASL,FNP) # predict the results for the test set`
- **Predict the probability score for the test data using the `compute` function.**
- `Predict=compute(nn,test)`
- `Predict$net.result`
- Converting probabilities into binary classes setting threshold level 0.5
- `prob <- Predict$net.result`
- `pred <- ifelse(prob>0.5, 1, 0) pred`
- **Report Results**