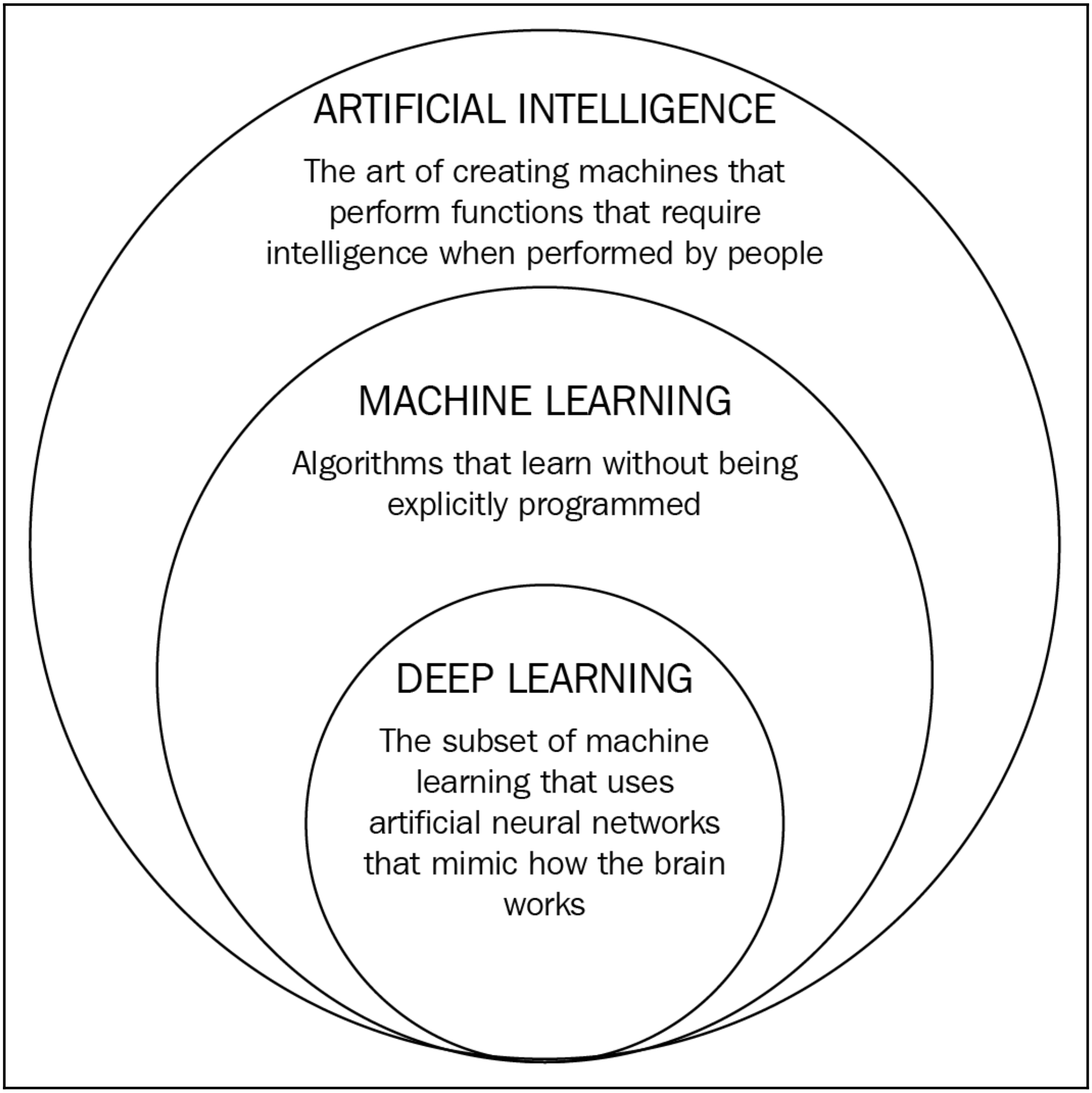
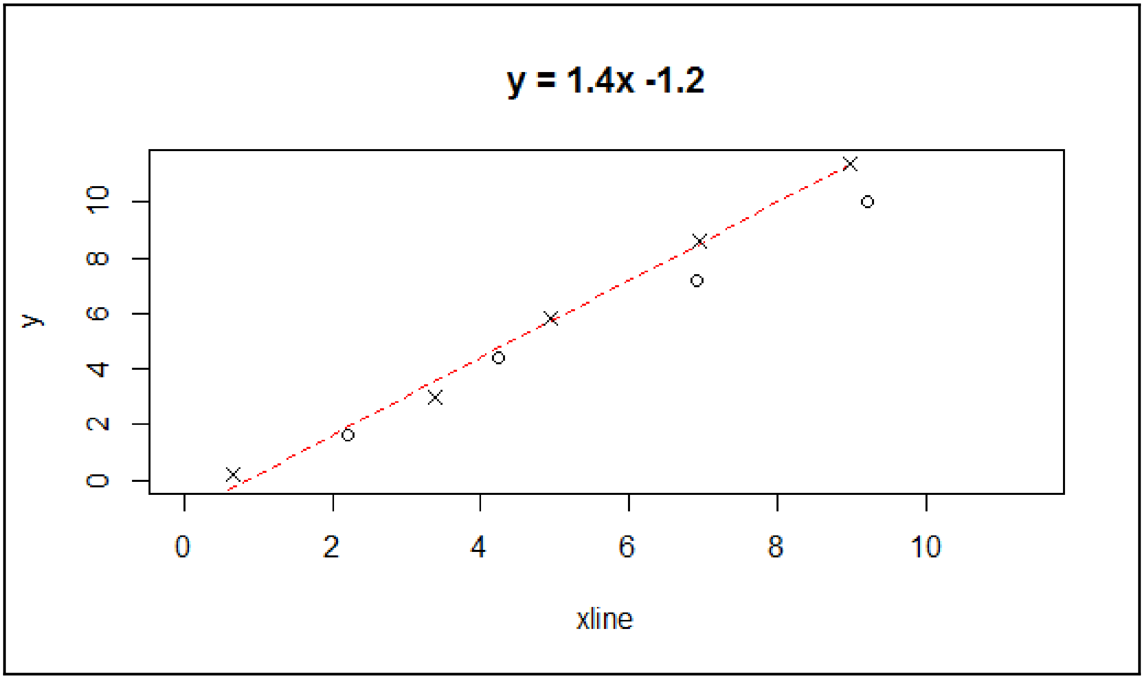
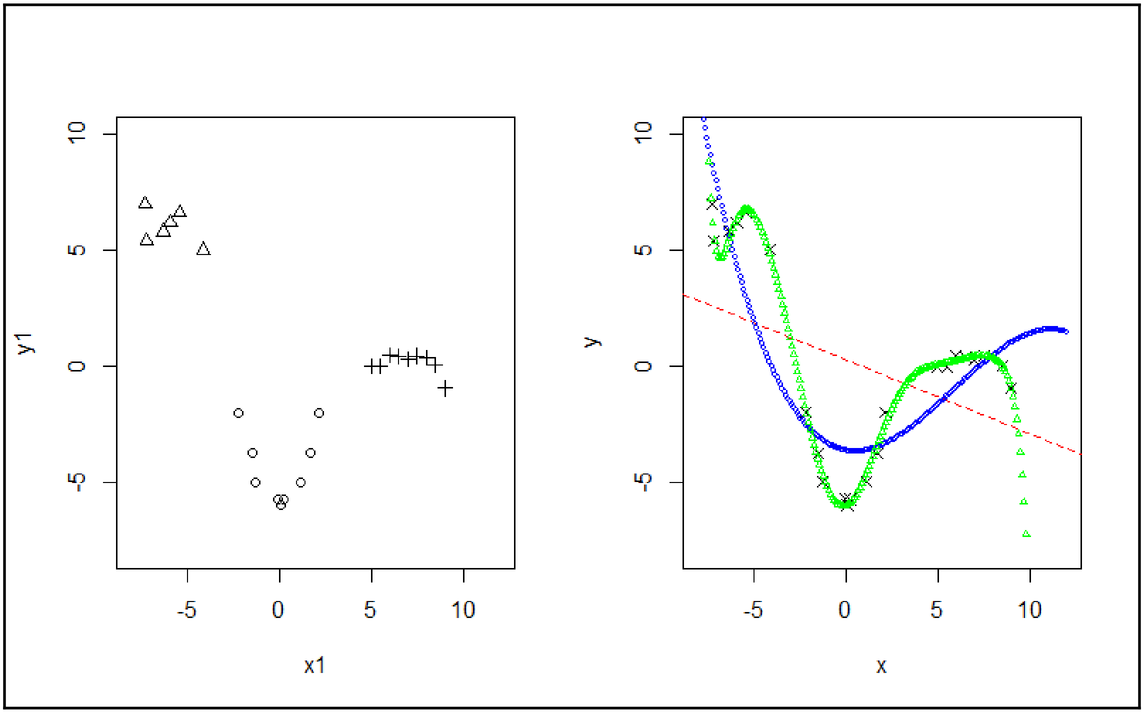
Table of Contents

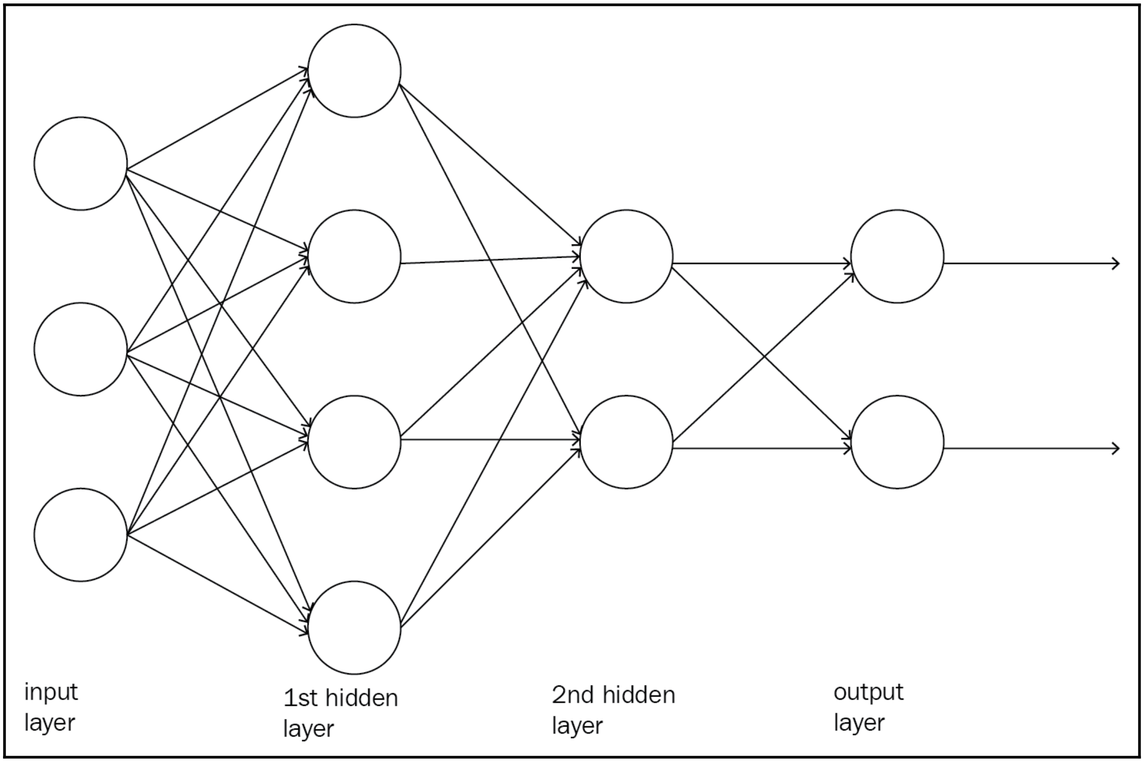
Graphics Bundle

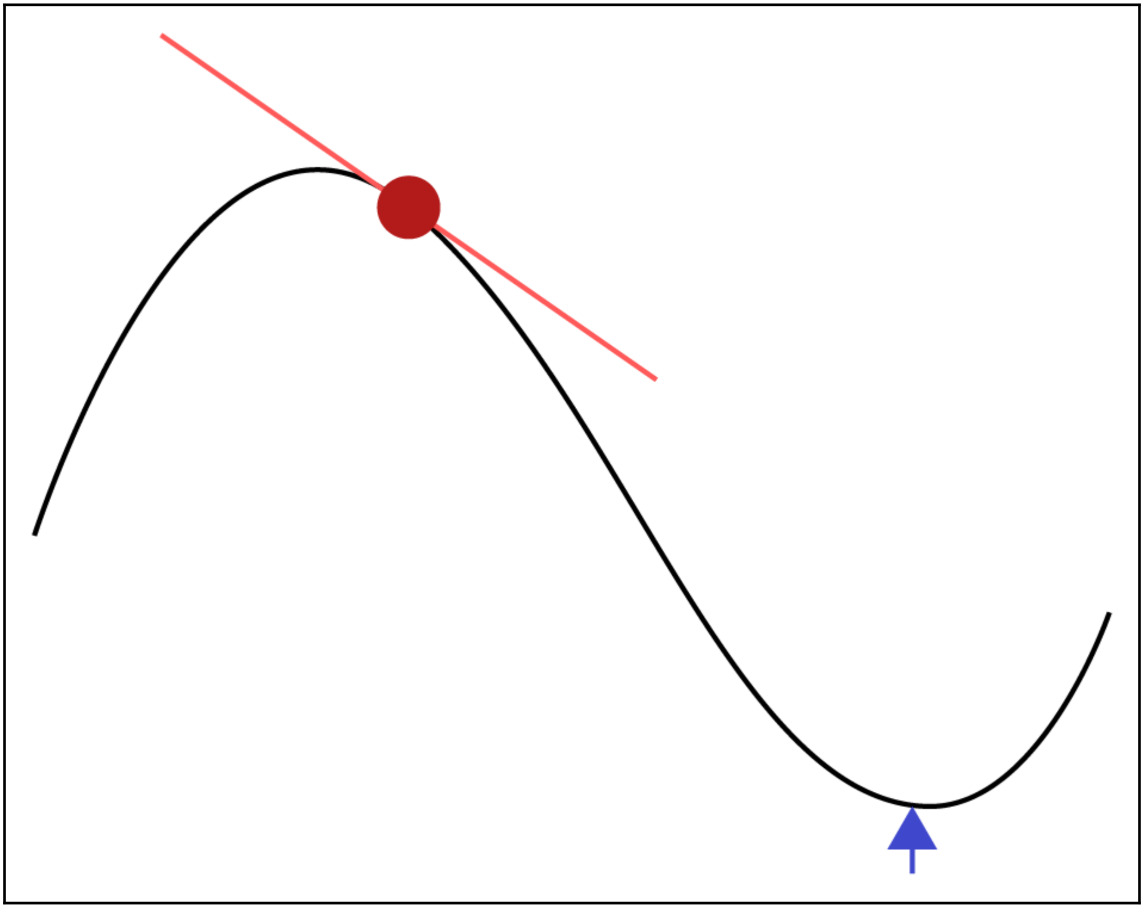
Chapter 1: Getting Started with Deep Learning

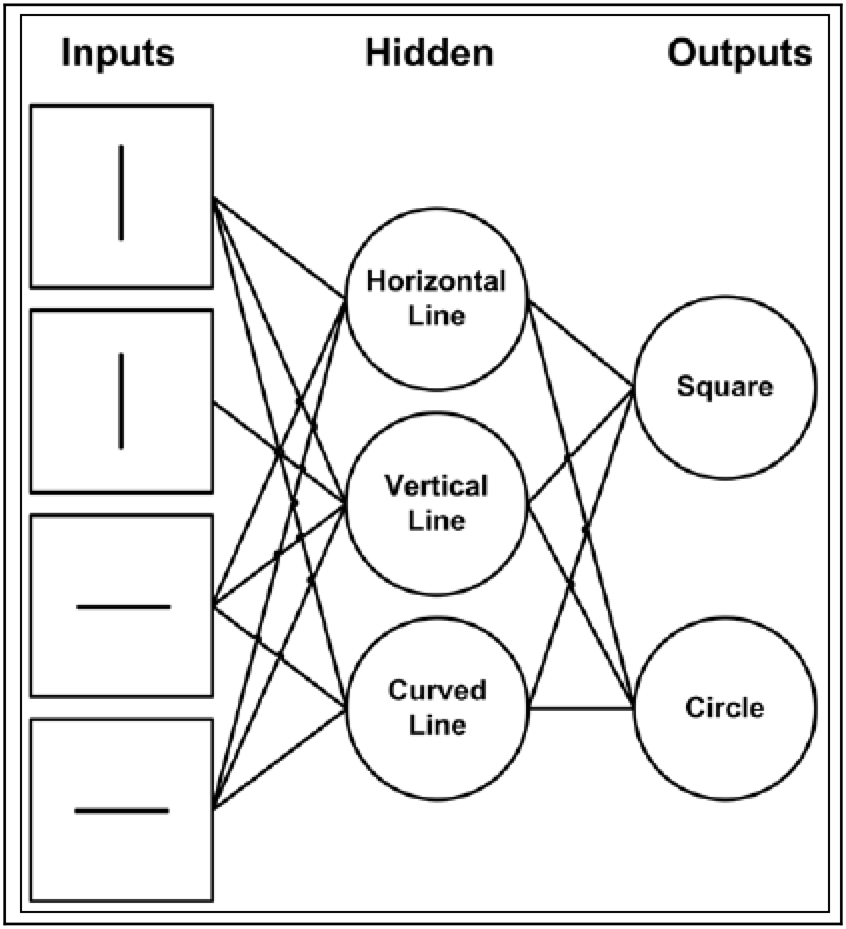

ARTIFICIAL INTELLIGENCE
The art of creating machines that perform functions that require intelligence when performed by people
MACHINE LEARNING
Algorithms that learn without being explicitly programmed
DEEP LEARNING
The subset of machine learning that uses artificial neural networks that mimic how the brain works

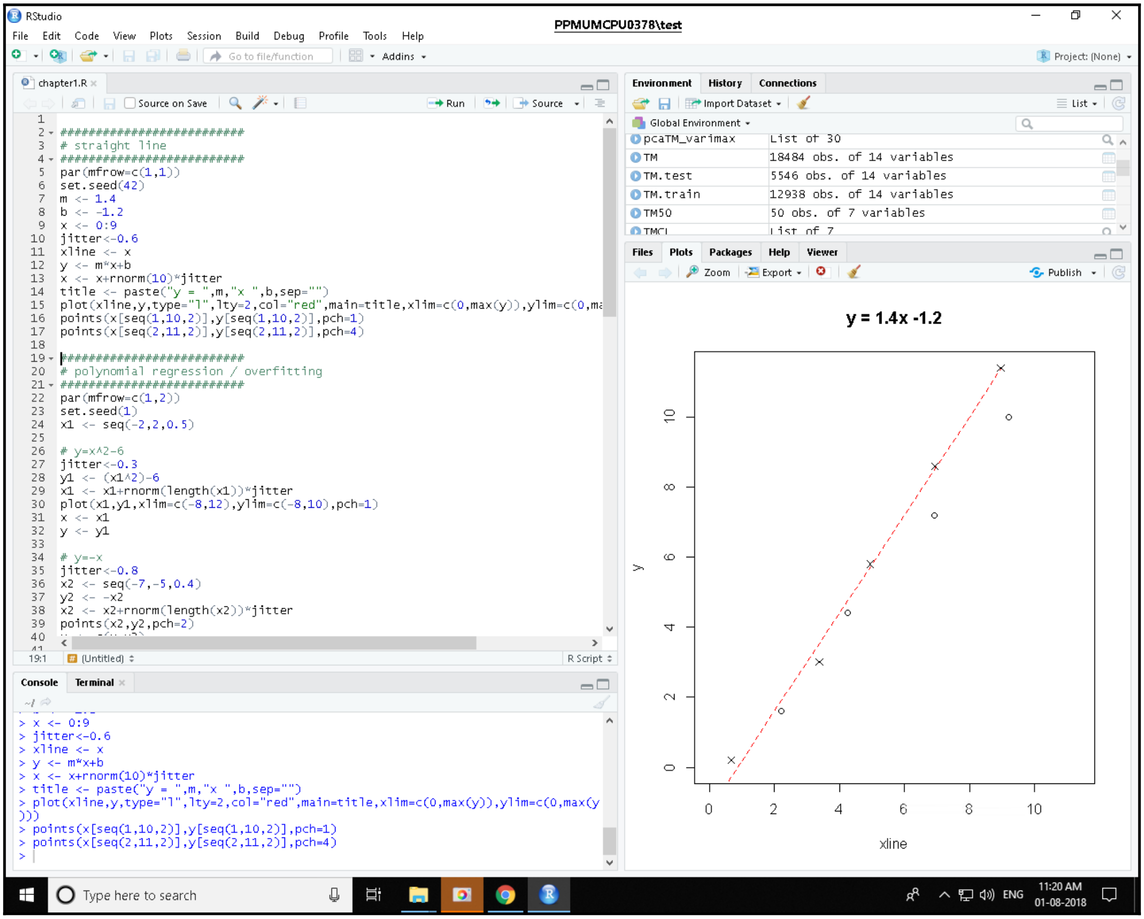

y = 1.4x -1.2
y
0
2
4
6
8
10
xline
0
2
4
6
8
10


y1
10
5
0
-5
-5
0
5
10
x1
y
10
5
0
-5
-5
0
5
10
x


input
layer
1st hidden
layer
2nd hidden
layer
output
layer



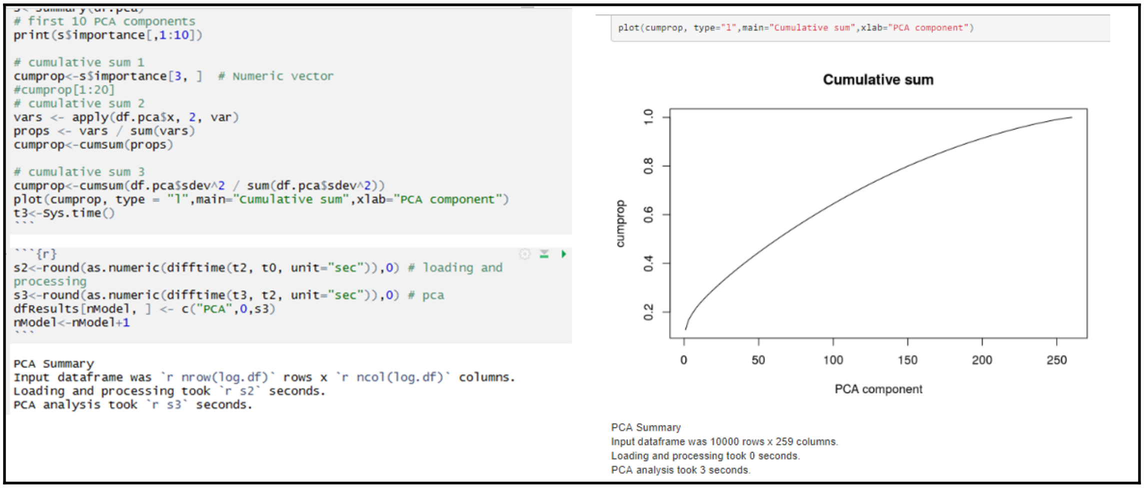

Inputs
Hidden
Outputs
Horizontal Line
Vertical Line
Curved Line
Square
Circle



```
##########################
# straight line
##########################
par(mfrow=c(1,1))
set.seed(42)
m <- 1.4
b <- -1.2
x <- 0:9
jitter<-0.6
xline <- x
y <- m*x+b
x <- x+rnorm(10)*jitter
title <- paste("y = ",m,"x ",b,sep="")
plot(xline,y,type="l",lty=2,col="red",main=title,xlim=c(0,max(y)),ylim=c(0,ma
points(x[seq(1,10,2)],y[seq(1,10,2)],pch=1)
points(x[seq(2,11,2)],y[seq(2,11,2)],pch=4)

##########################
# polynomial regression / overfitting
##########################
par(mfrow=c(1,2))
set.seed(1)
x1 <- seq(-2,2,0.5)

# y=x^2-6
jitter<-0.3
y1 <- (x1^2)-6
x1 <- x1+rnorm(length(x1))*jitter
plot(x1,y1,xlim=c(-8,12),ylim=c(-8,10),pch=1)
x <- x1
y <- y1

# y=-x
jitter<-0.8
x2 <- seq(-7,-5,0.4)
y2 <- -x2
x2 <- x2+rnorm(length(x2))*jitter
points(x2,y2,pch=2)
```

```
> x <- 0:9
> jitter<-0.6
> xline <- x
> y <- m*x+b
> x <- x+rnorm(10)*jitter
> title <- paste("y = ",m,"x ",b,sep="")
> plot(xline,y,type="l",lty=2,col="red",main=title,xlim=c(0,max(y)),ylim=c(0,max(y
)))
> points(x[seq(1,10,2)],y[seq(1,10,2)],pch=1)
> points(x[seq(2,11,2)],y[seq(2,11,2)],pch=4)
> 
```

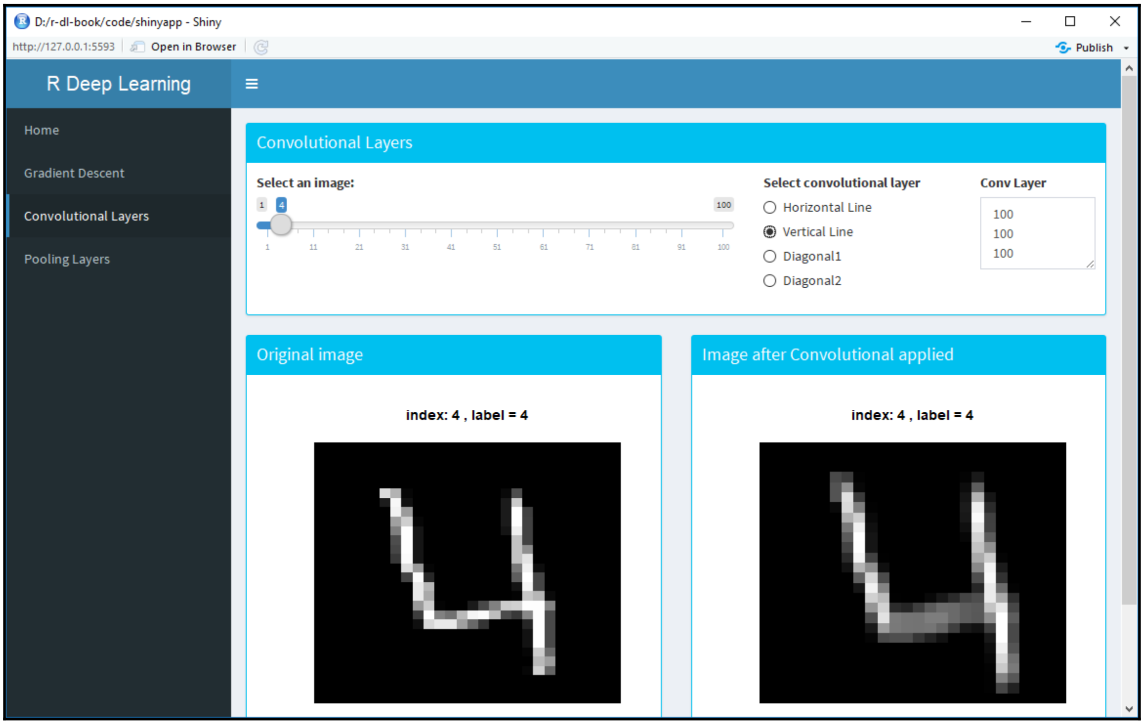
```
pcaTM_varimax  List of 30
TM             18484 obs. of 14 variables
TM.test        5546 obs. of 14 variables
TM.train       12938 obs. of 14 variables
TM50           50 obs. of 7 variables
TMCL           List of 7
```



~~~
s<-summary(df.pca)
# first 10 PCA components
print(s$importance[,1:10])

# cumulative sum 1
cumprop<-s$importance[3, ]  # Numeric vector
#cumprop[1:20]
# cumulative sum 2
vars <- apply(df.pca$x, 2, var)
props <- vars / sum(vars)
cumprop<-cumsum(props)

# cumulative sum 3
cumprop<-cumsum(df.pca$sdev^2 / sum(df.pca$sdev^2))
plot(cumprop, type = "l",main="Cumulative sum",xlab="PCA component")
t3<-Sys.time()
```

```{r}
s2<-round(as.numeric(difftime(t2, t0, unit="sec")),0) # loading and
processing
s3<-round(as.numeric(difftime(t3, t2, unit="sec")),0) # pca
dfResults[nModel, ] <- c("PCA",0,s3)
nModel<-nModel+1
```

PCA Summary
Input dataframe was `r nrow(log.df)` rows x `r ncol(log.df)` columns.
Loading and processing took `r s2` seconds.
PCA analysis took `r s3` seconds.
~~~

```
plot(cumprop, type="l",main="Cumulative sum",xlab="PCA component")
```

PCA Summary
Input dataframe was 10000 rows x 259 columns.
Loading and processing took 0 seconds.
PCA analysis took 3 seconds.



Select an image: 1 4 100

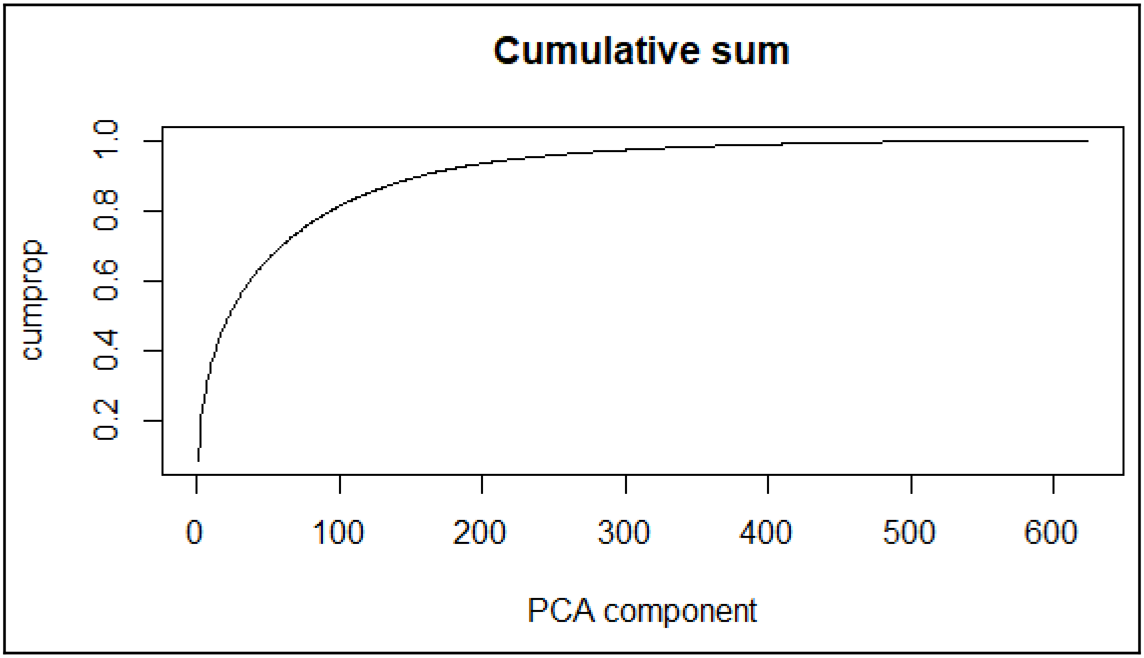
Select convolutional layer
○ Horizontal Line
● Vertical Line
○ Diagonal1
○ Diagonal2

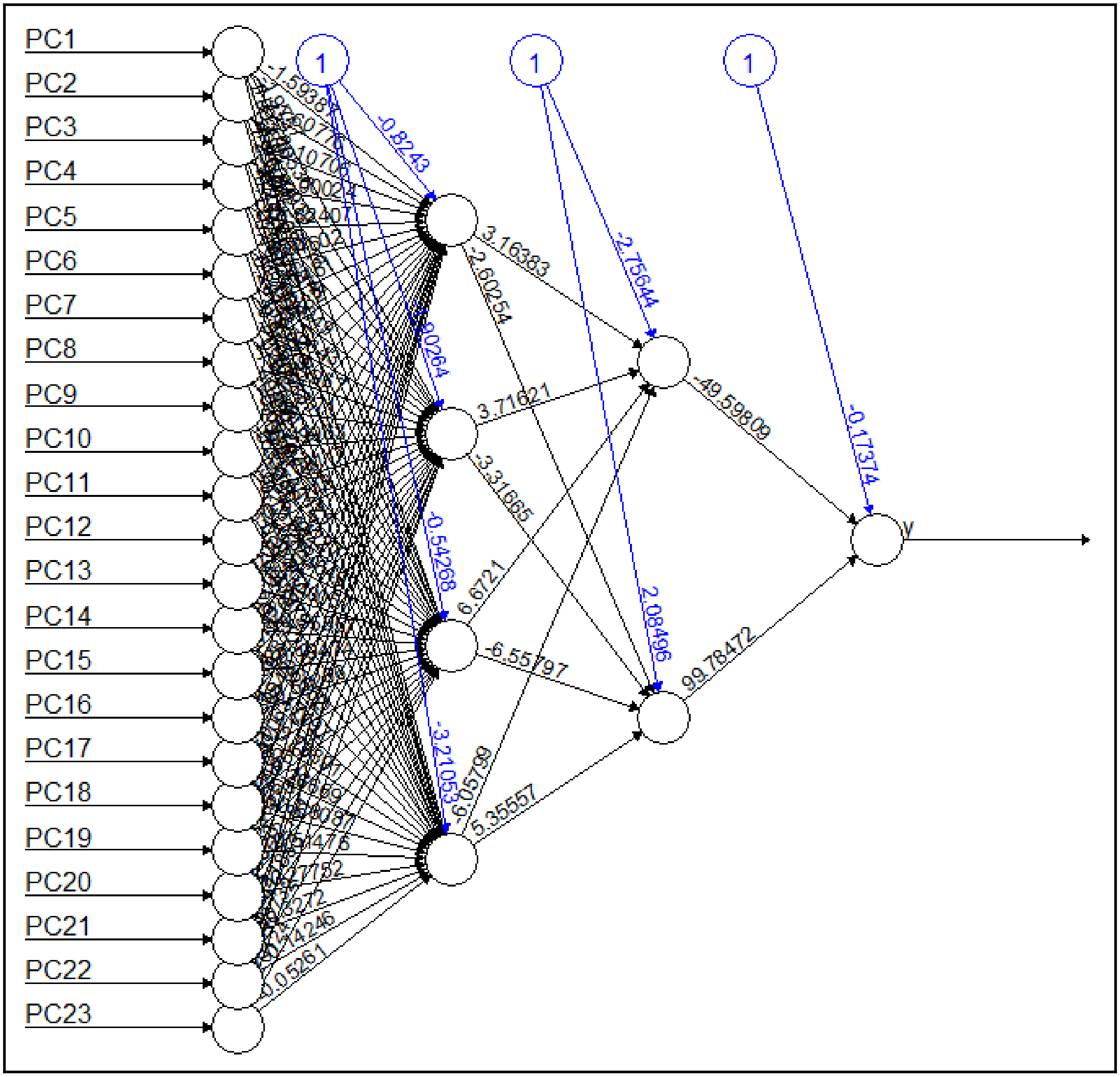
Conv Layer
100
100
100

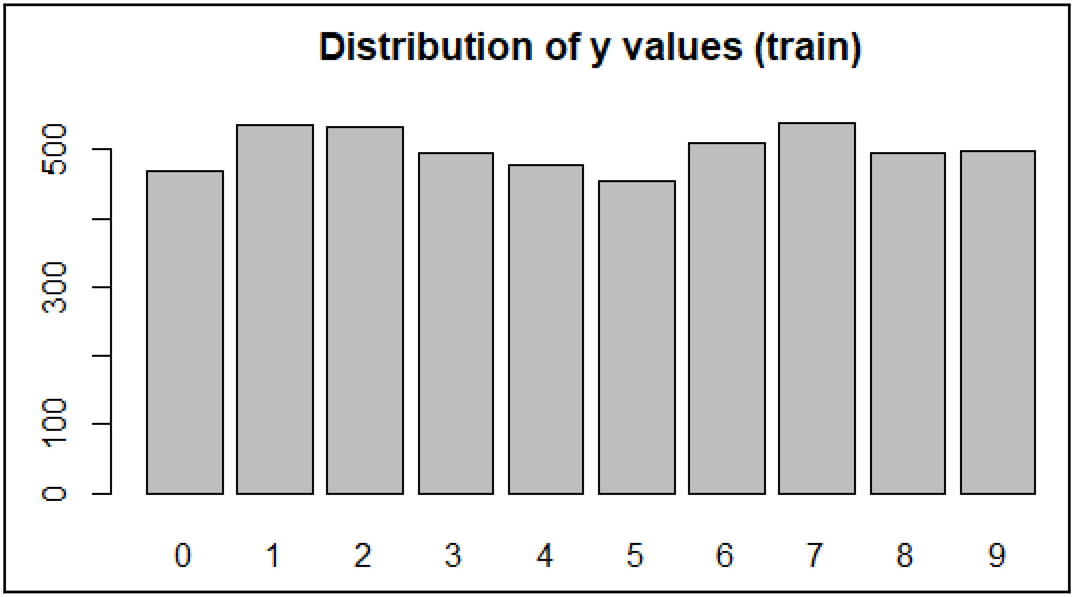
index: 4 , label = 4

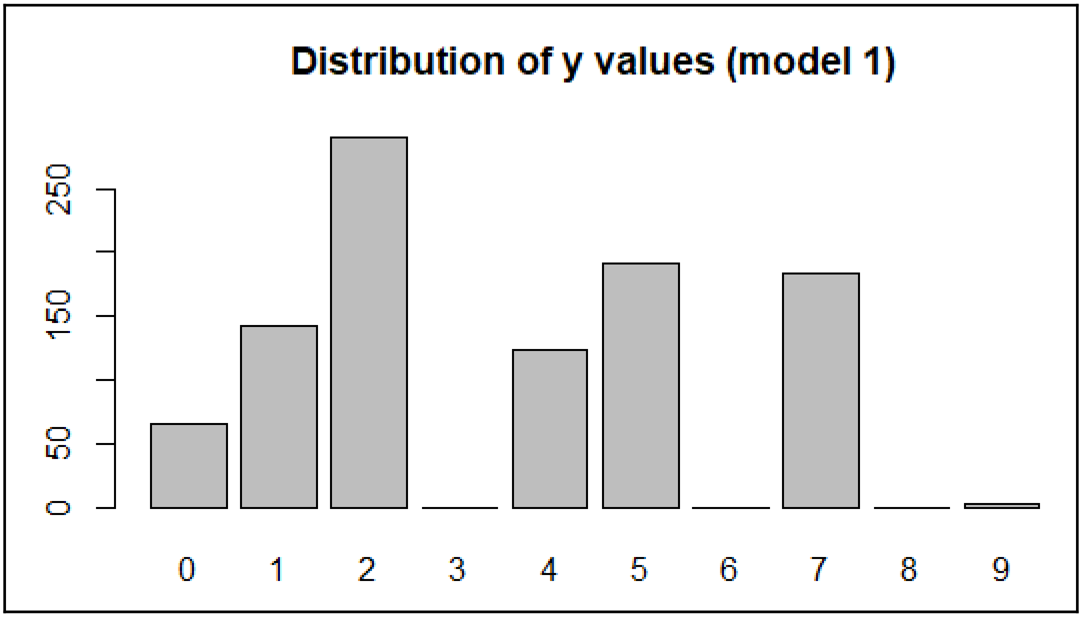
index: 4 , label = 4

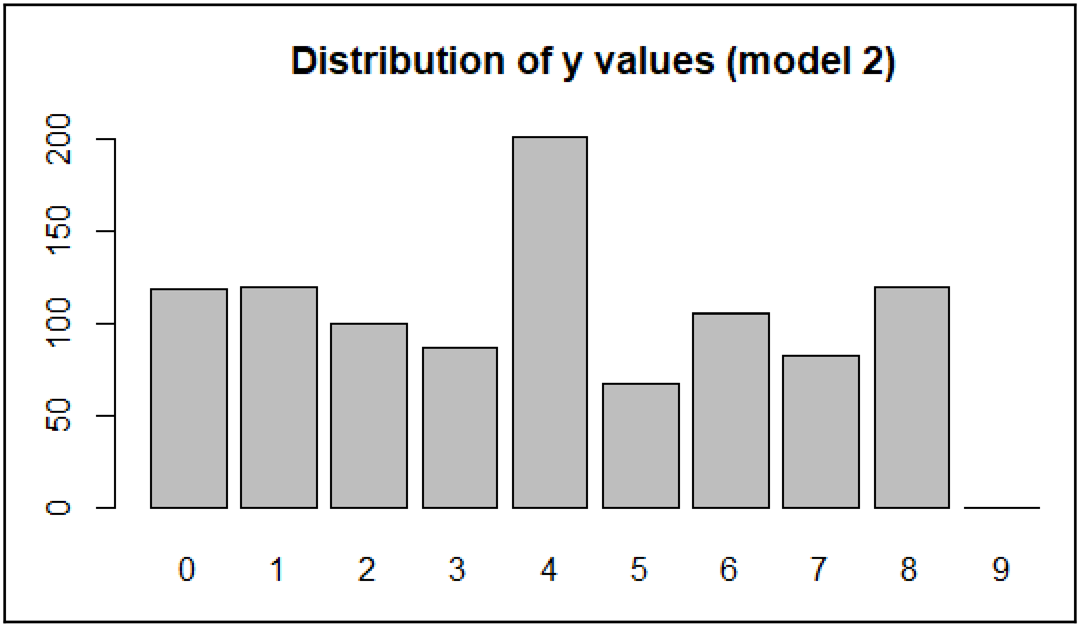
Chapter 2: Training a Prediction Model

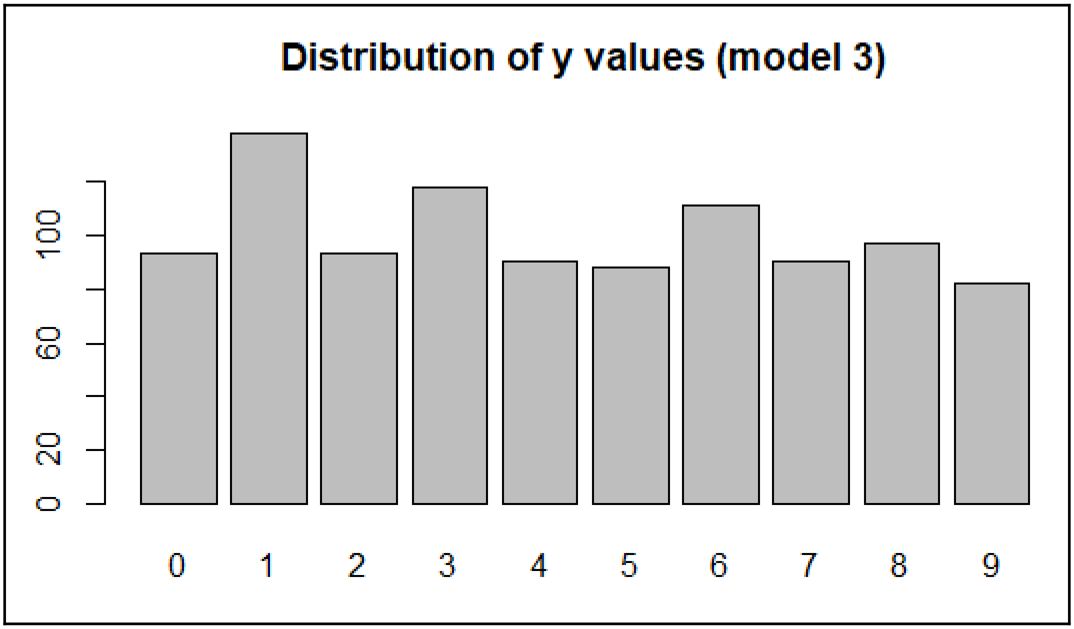


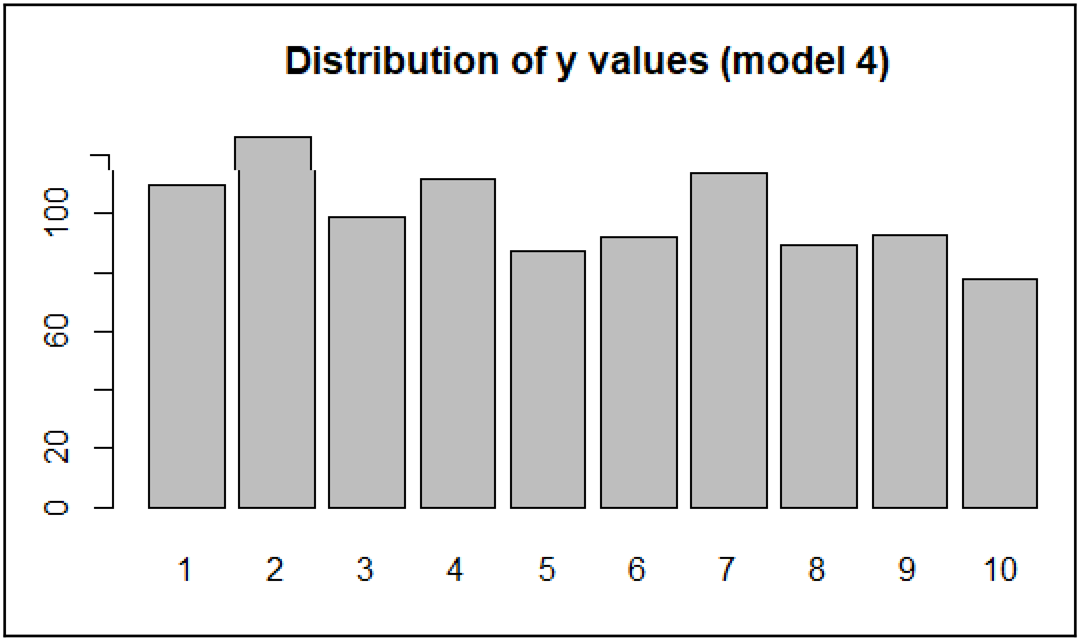

PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC20
PC21
PC22
PC23
1
1
1
-0.8243
3.16383
-2.60254
-2.75644
3.71621
-3.31665
-49.59809
-0.17374
-0.54268
6.6721
-6.55797
2.08496
99.78472
-3.21053
5.35557
y

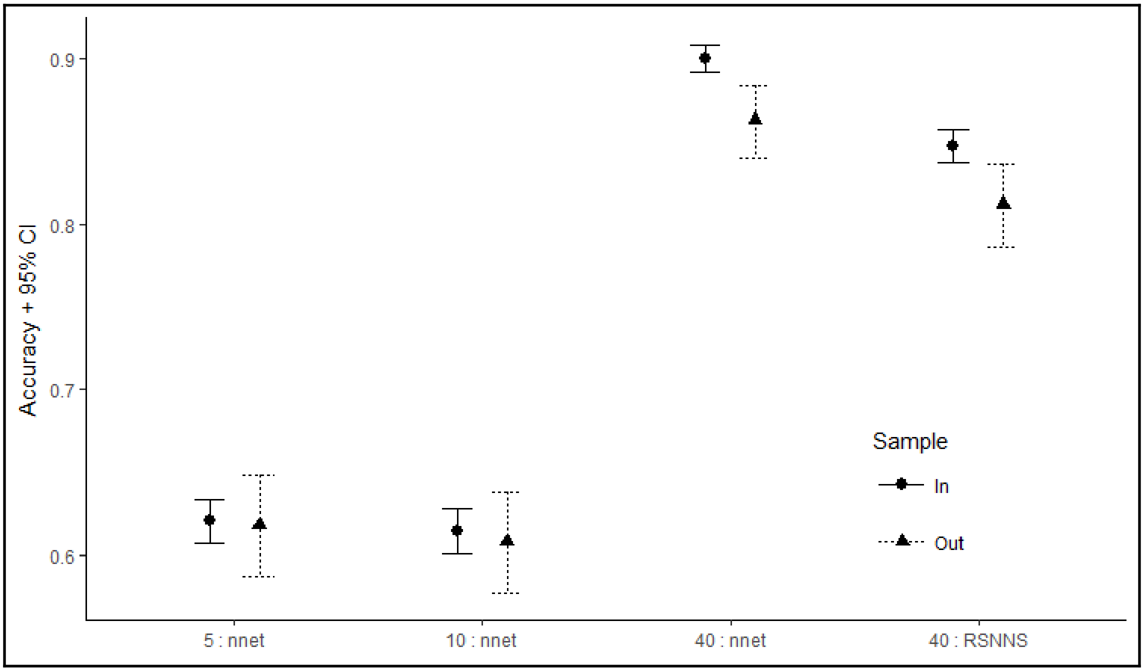

Distribution of y values (train)
0
100
300
500
0
1
2
3
4
5
6
7
8
9

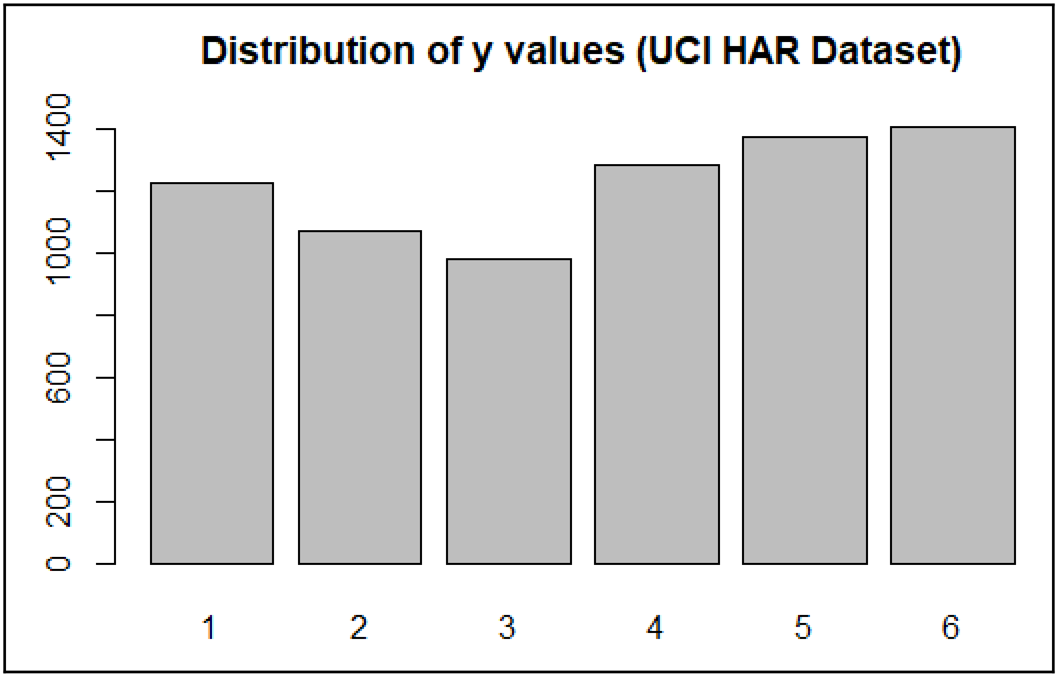

Distribution of y values (model 1)
0
50
150
250
0
1
2
3
4
5
6
7
8
9


Distribution of y values (model 2)
0
50
100
150
200
0
1
2
3
4
5
6
7
8
9

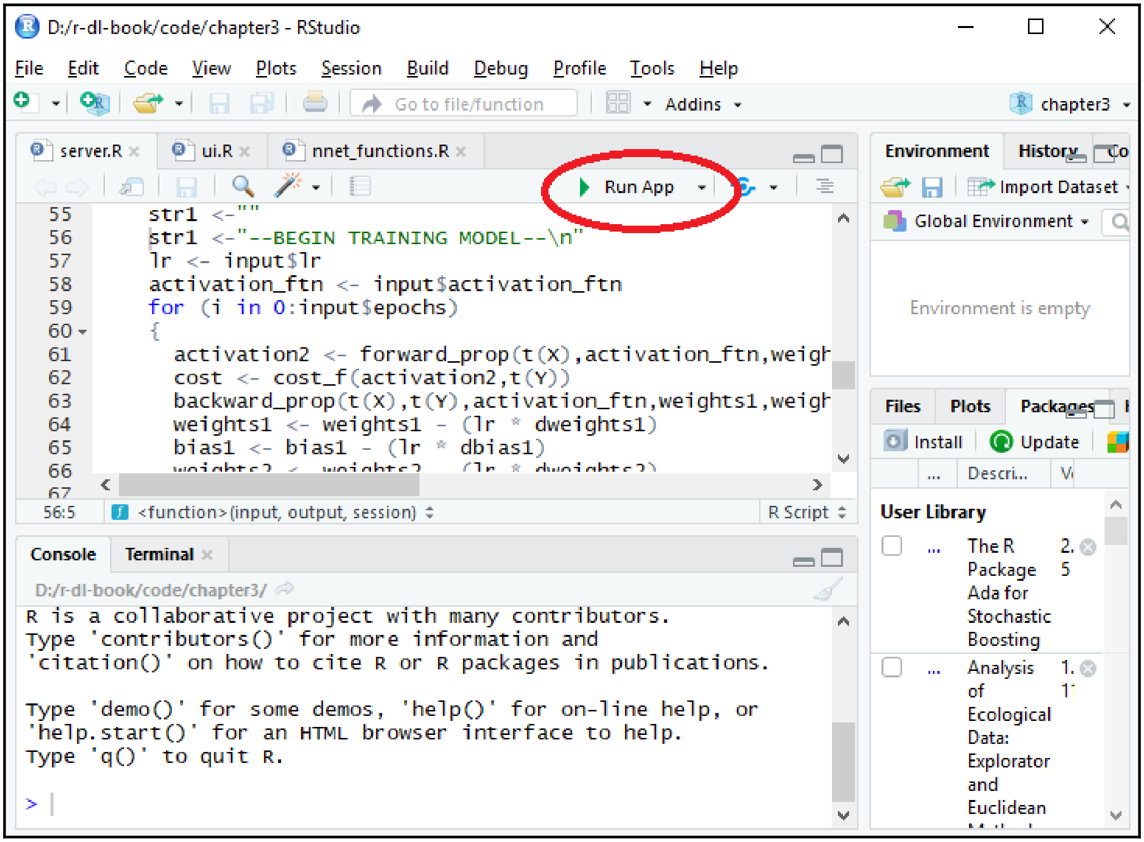

Distribution of y values (model 3)
0
20
60
100
0
1
2
3
4
5
6
7
8
9


Distribution of y values (model 4)
0
20
60
100
1
2
3
4
5
6
7
8
9
10


Accuracy + 95% CI
0.9
0.8
0.7
0.6
5 : nnet
10 : nnet
40 : nnet
40 : RSNNS
Sample
In
Out

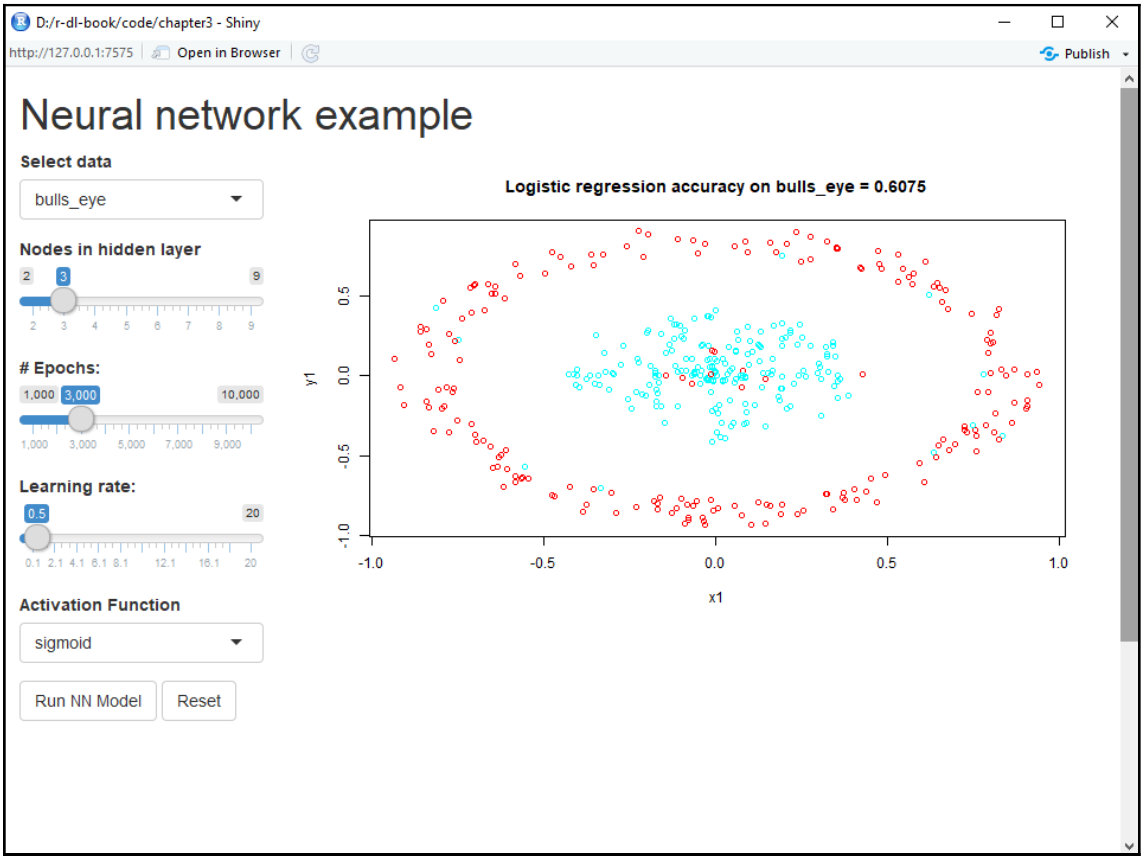

Distribution of y values (UCI HAR Dataset)
0
200
600
1000
1400
1
2
3
4
5
6

Chapter 3: Deep Learning Fundamentals


D:/r-dl-book/code/chapter3 - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
chapter3
server.R
ui.R
nnet_functions.R
Run App
56:5
<function>(input, output, session)
R Script
Environment
History
Import Dataset
Global Environment
Environment is empty
Files
Plots
Packages
Install
Update
Descri...
User Library
The R Package Ada for Stochastic Boosting
2.5
Analysis of Ecological Data: Explorator and Euclidean
Console
Terminal
D:/r-dl-book/code/chapter3/

```
    str1 <-""
    str1 <-"--BEGIN TRAINING MODEL--\n"
    lr <- input$lr
    activation_ftn <- input$activation_ftn
    for (i in 0:input$epochs)
    {
      activation2 <- forward_prop(t(X),activation_ftn,weigh
      cost <- cost_f(activation2,t(Y))
      backward_prop(t(X),t(Y),activation_ftn,weights1,weigh
      weights1 <- weights1 - (lr * dweights1)
      bias1 <- bias1 - (lr * dbias1)
      weights2 <- weights2 - (lr * dweights2)
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```



D:/r-dl-book/code/chapter3 - Shiny

http://127.0.0.1:7575

Neural network example

Select data

bulls_eye

Nodes in hidden layer

3

Epochs:

3,000

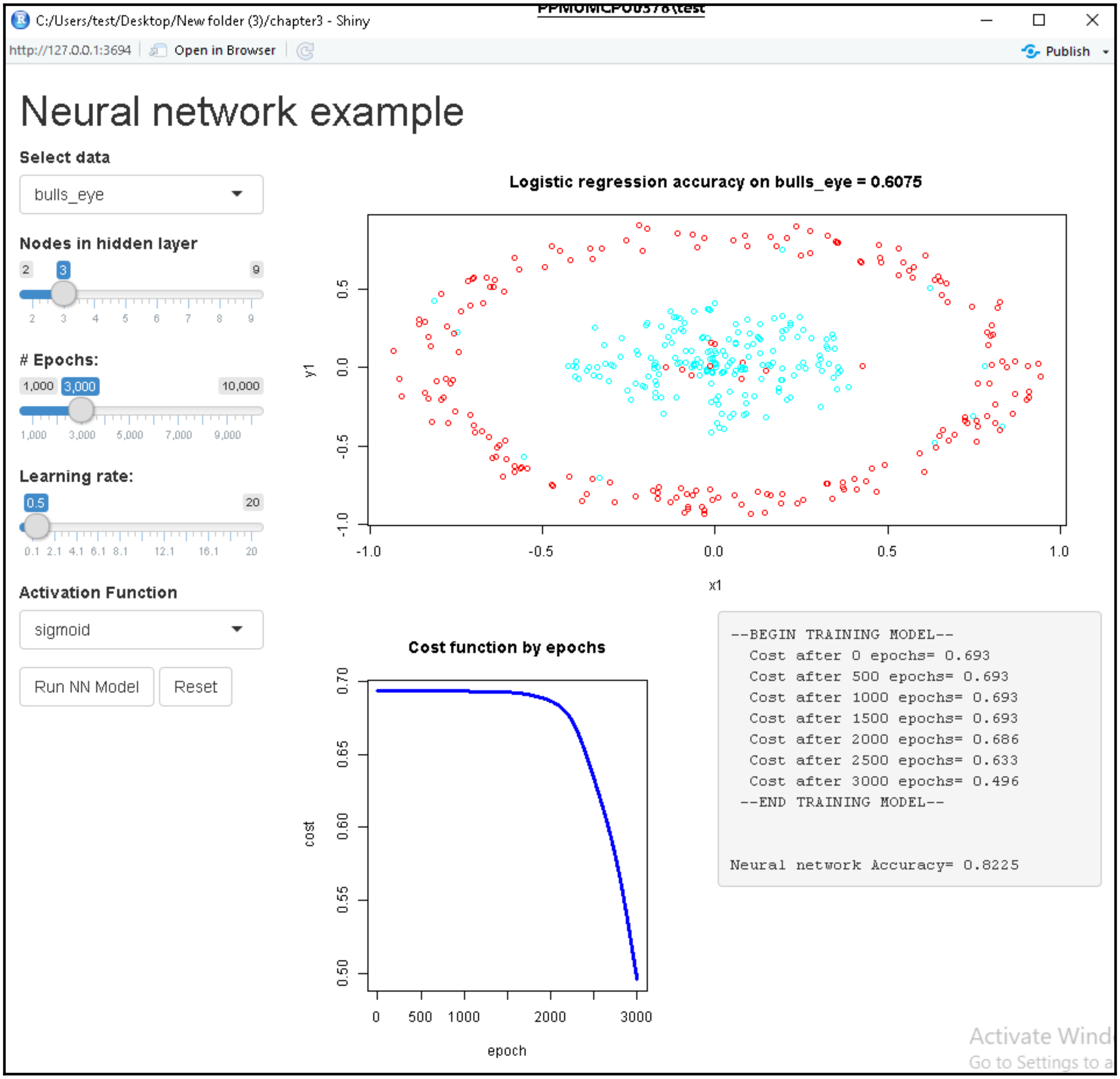
Learning rate:

0.5

Activation Function

sigmoid

Logistic regression accuracy on bulls_eye = 0.6075



Neural network example

Select data: bulls_eye

Nodes in hidden layer: 3 (2–9)

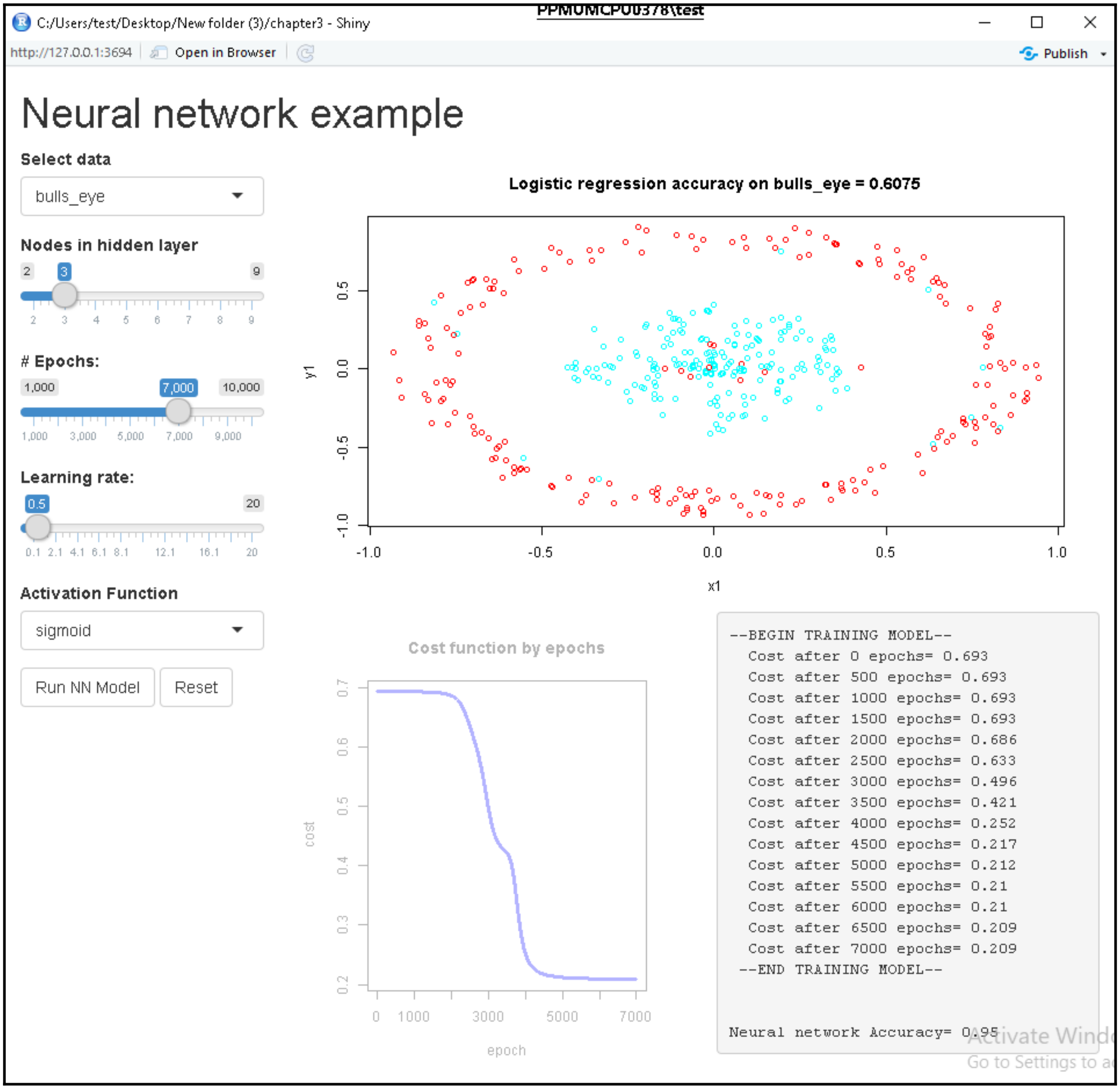
Epochs: 3,000 (1,000–10,000)

Learning rate: 0.5 (0.1–20)

Activation Function: sigmoid

Run NN Model | Reset

```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 0.693
  Cost after 1000 epochs= 0.693
  Cost after 1500 epochs= 0.693
  Cost after 2000 epochs= 0.686
  Cost after 2500 epochs= 0.633
  Cost after 3000 epochs= 0.496
 --END TRAINING MODEL--


Neural network Accuracy= 0.8225
```



Neural network example

Select data: bulls_eye

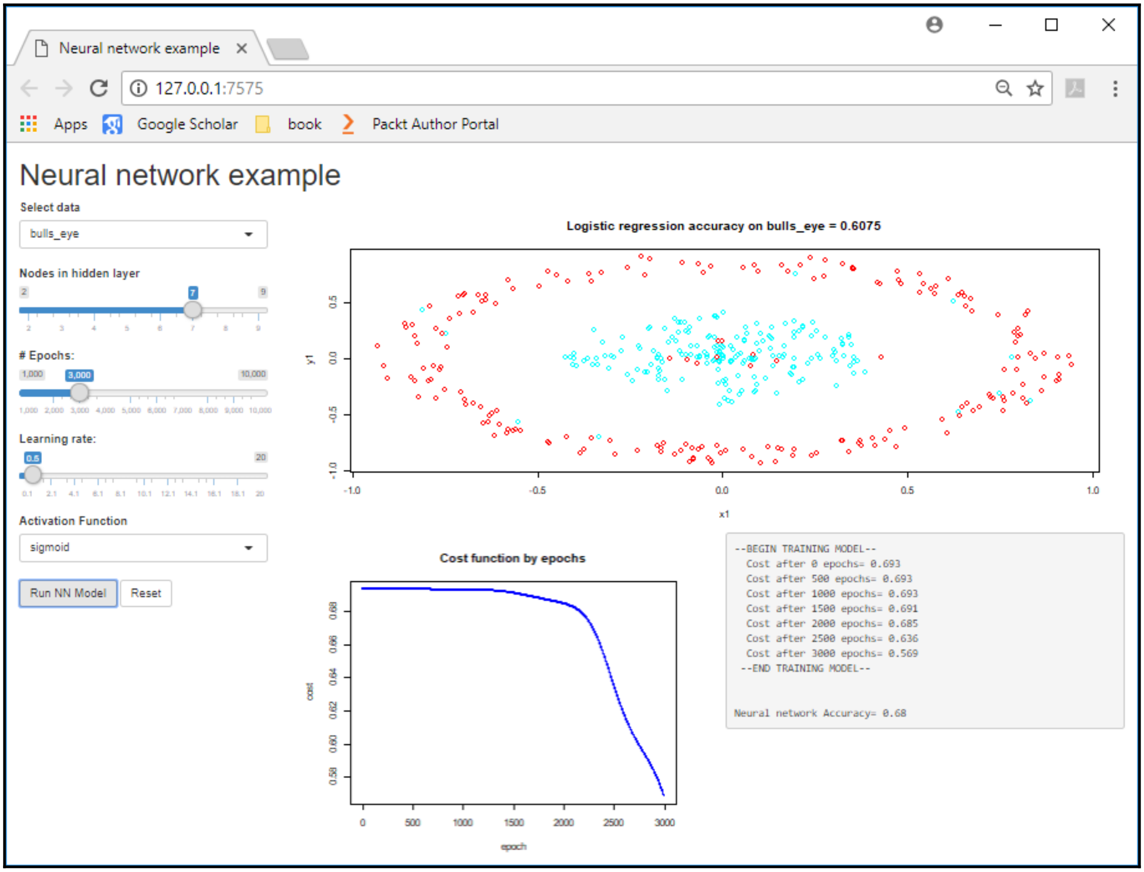
Nodes in hidden layer: 3 (2–9)

Epochs: 7,000 (1,000–10,000)

Learning rate: 0.5 (0.1–20)

Activation Function: sigmoid

```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 0.693
  Cost after 1000 epochs= 0.693
  Cost after 1500 epochs= 0.693
  Cost after 2000 epochs= 0.686
  Cost after 2500 epochs= 0.633
  Cost after 3000 epochs= 0.496
  Cost after 3500 epochs= 0.421
  Cost after 4000 epochs= 0.252
  Cost after 4500 epochs= 0.217
  Cost after 5000 epochs= 0.212
  Cost after 5500 epochs= 0.21
  Cost after 6000 epochs= 0.21
  Cost after 6500 epochs= 0.209
  Cost after 7000 epochs= 0.209
 --END TRAINING MODEL--


Neural network Accuracy= 0.95
```



Neural network example

Select data

bulls_eye

Nodes in hidden layer

7

Epochs:

3,000

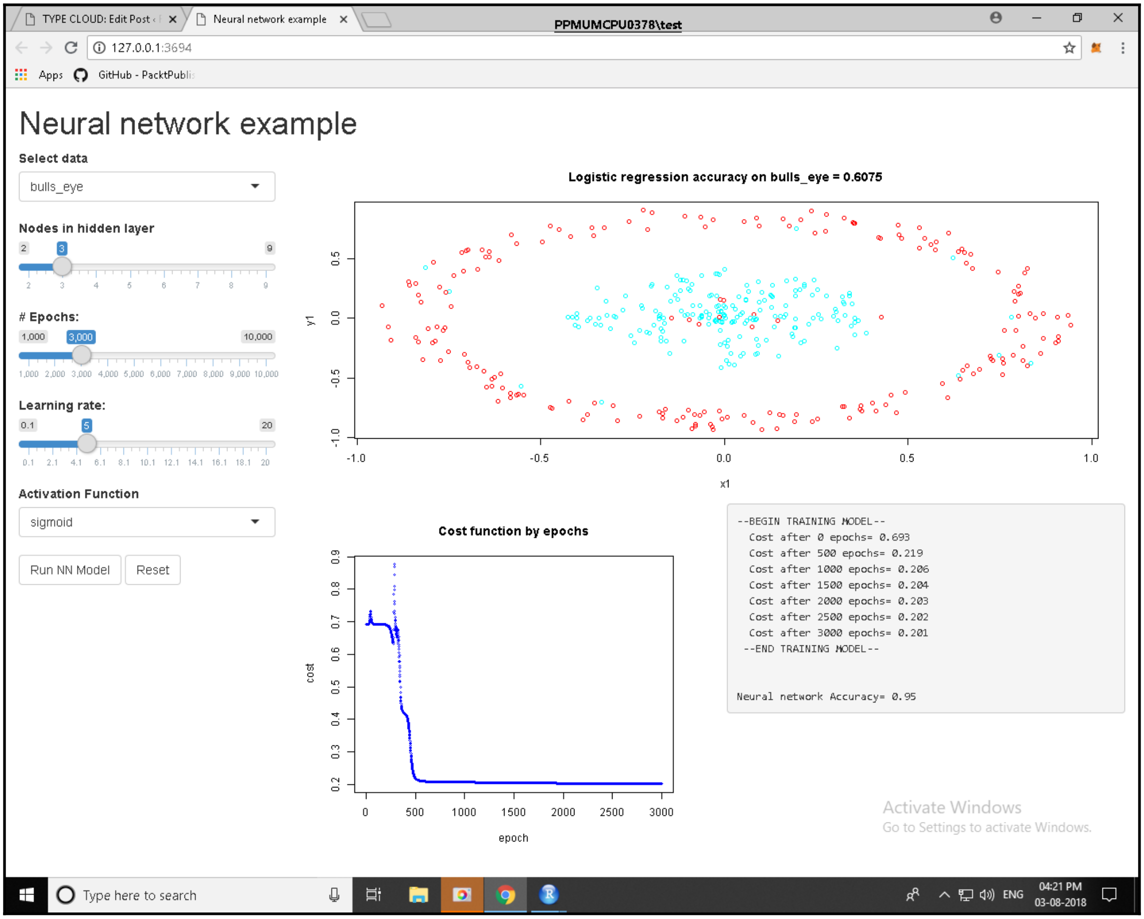
Learning rate:

0.5

Activation Function

sigmoid

```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 0.693
  Cost after 1000 epochs= 0.693
  Cost after 1500 epochs= 0.691
  Cost after 2000 epochs= 0.685
  Cost after 2500 epochs= 0.636
  Cost after 3000 epochs= 0.569
 --END TRAINING MODEL--


Neural network Accuracy= 0.68
```



Neural network example

Select data: bulls_eye

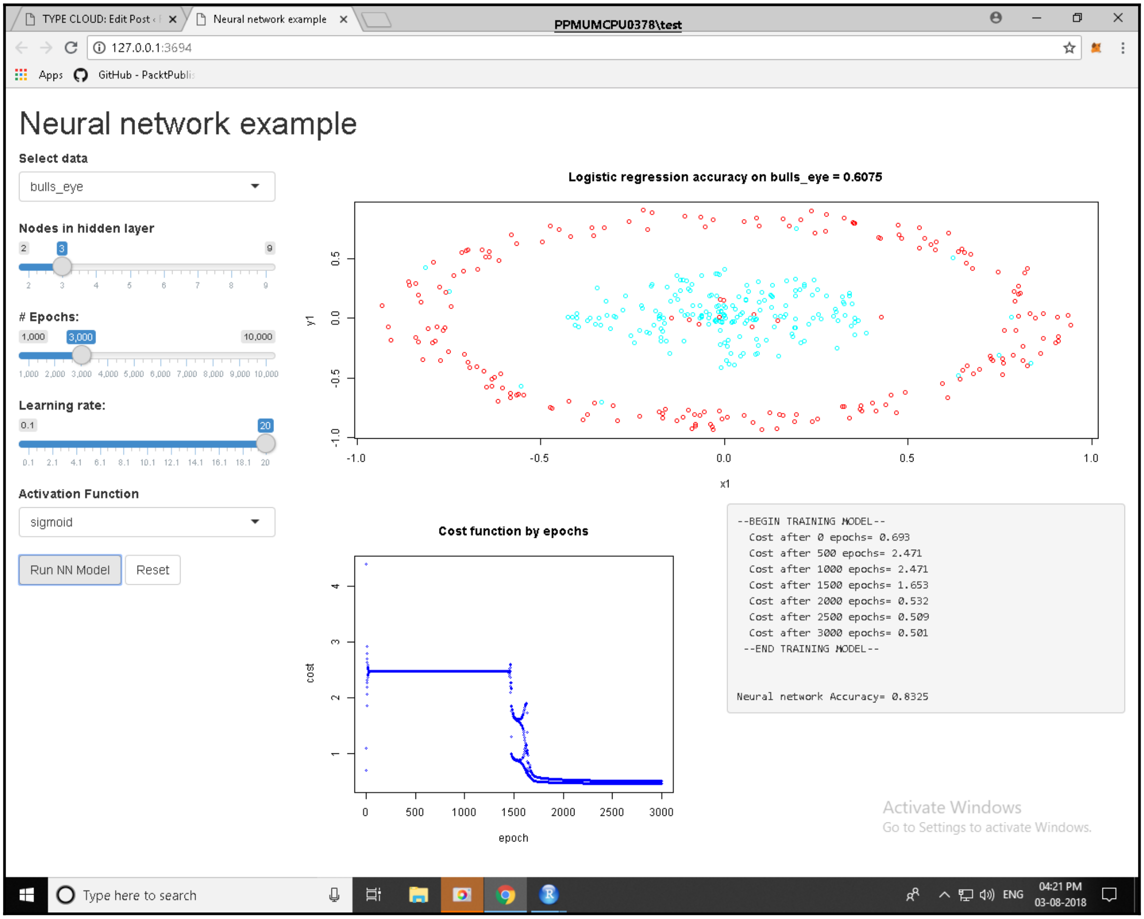
Nodes in hidden layer: 3

Epochs: 3,000

Learning rate: 5

Activation Function: sigmoid

```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 0.219
  Cost after 1000 epochs= 0.206
  Cost after 1500 epochs= 0.204
  Cost after 2000 epochs= 0.203
  Cost after 2500 epochs= 0.202
  Cost after 3000 epochs= 0.201
 --END TRAINING MODEL--


Neural network Accuracy= 0.95
```



Neural network example

Select data

bulls_eye

Nodes in hidden layer

3

Epochs:

3,000

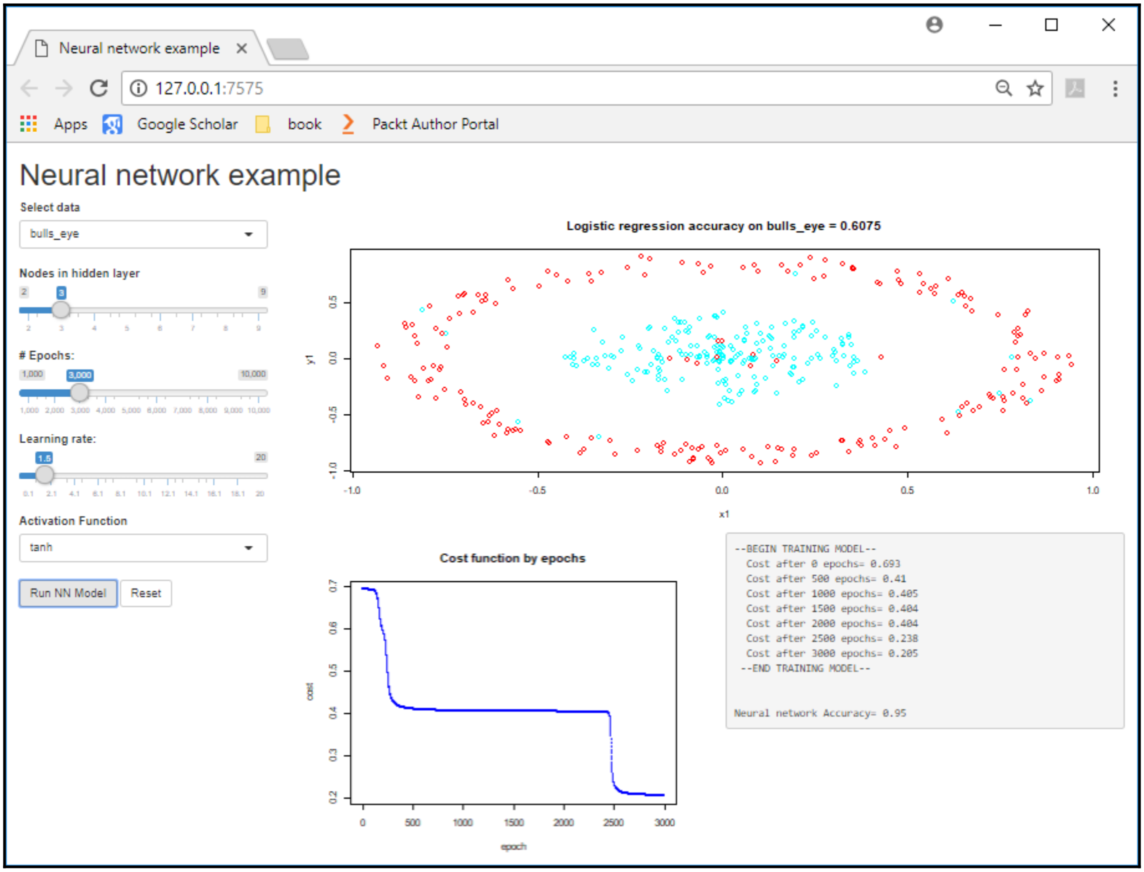
Learning rate:

20

Activation Function

sigmoid

```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 2.471
  Cost after 1000 epochs= 2.471
  Cost after 1500 epochs= 1.653
  Cost after 2000 epochs= 0.532
  Cost after 2500 epochs= 0.509
  Cost after 3000 epochs= 0.501
 --END TRAINING MODEL--


Neural network Accuracy= 0.8325
```



Neural network example

Select data: bulls_eye

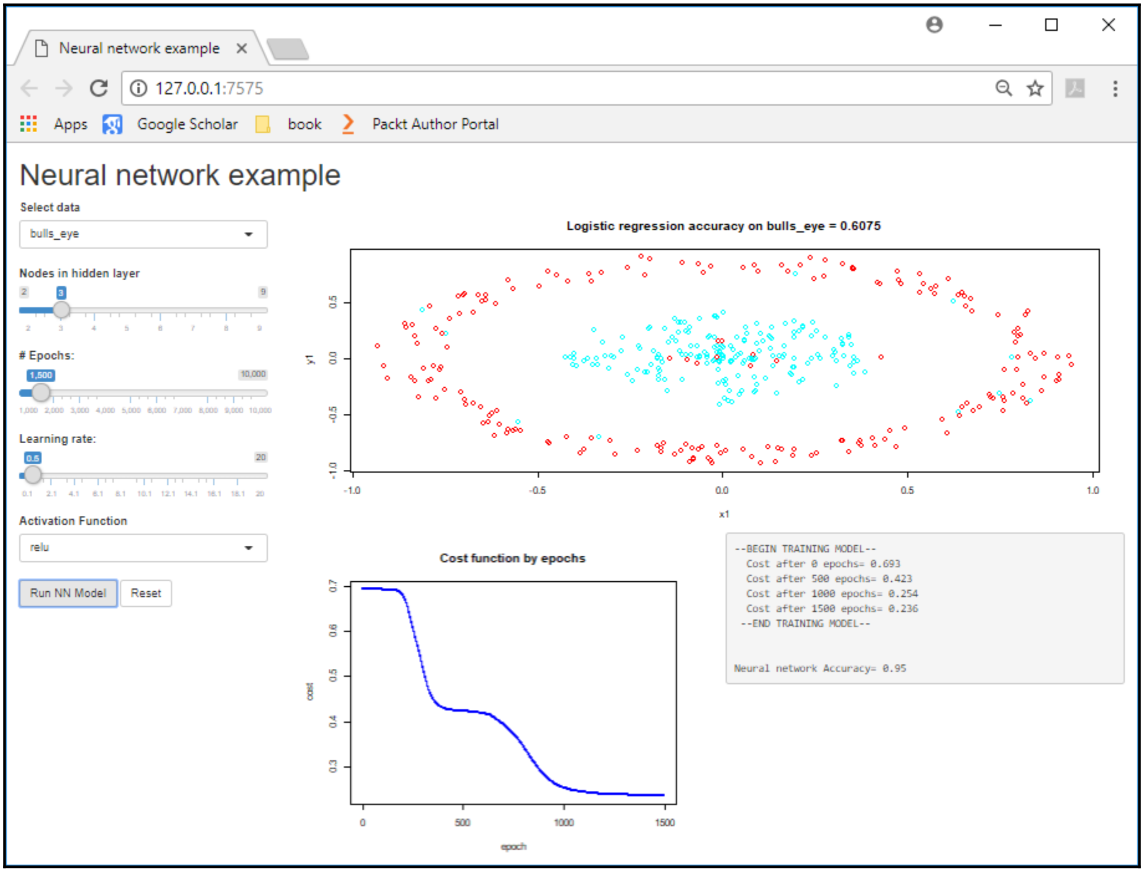
Nodes in hidden layer: 3

# Epochs: 3,000

Learning rate: 1.5

Activation Function: tanh

```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 0.41
  Cost after 1000 epochs= 0.405
  Cost after 1500 epochs= 0.404
  Cost after 2000 epochs= 0.404
  Cost after 2500 epochs= 0.238
  Cost after 3000 epochs= 0.205
 --END TRAINING MODEL--


Neural network Accuracy= 0.95
```



Neural network example

Select data

bulls_eye

Nodes in hidden layer

3

Epochs:

1,500

Learning rate:

0.5

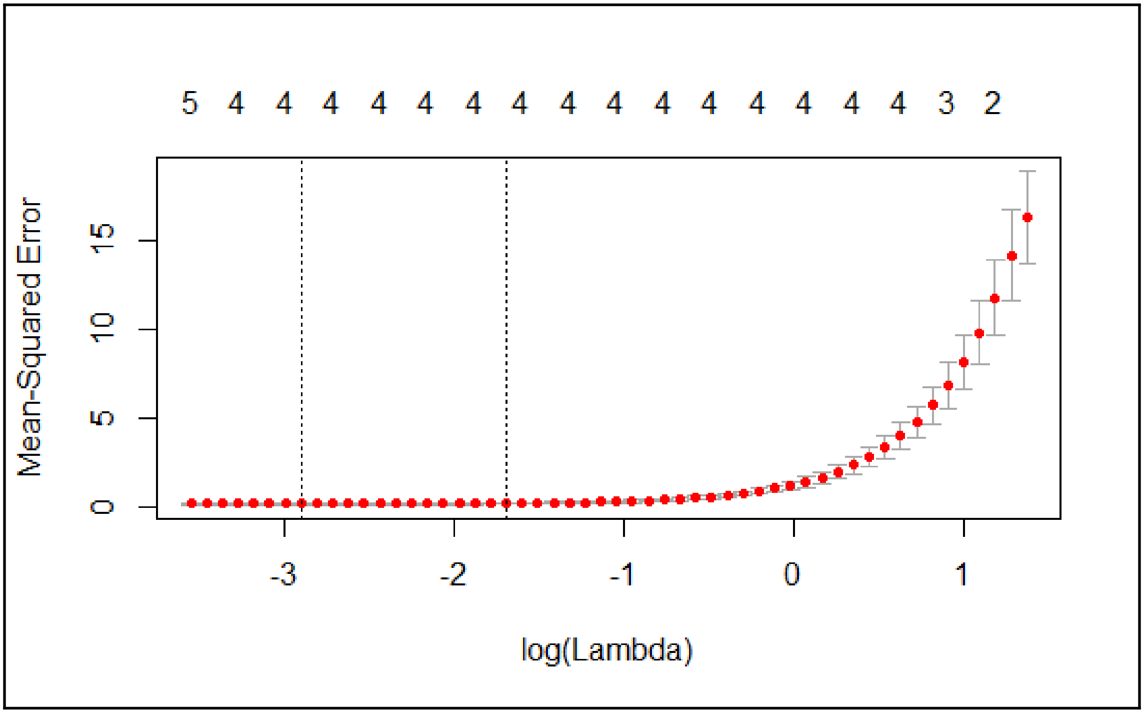
Activation Function

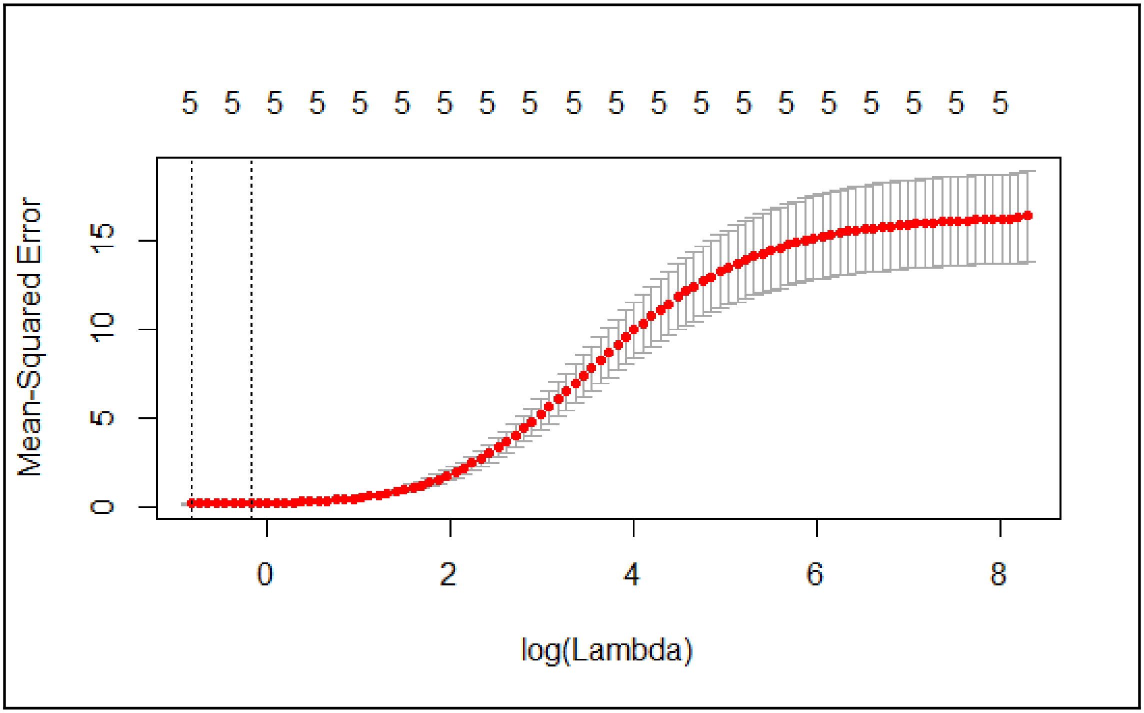
relu

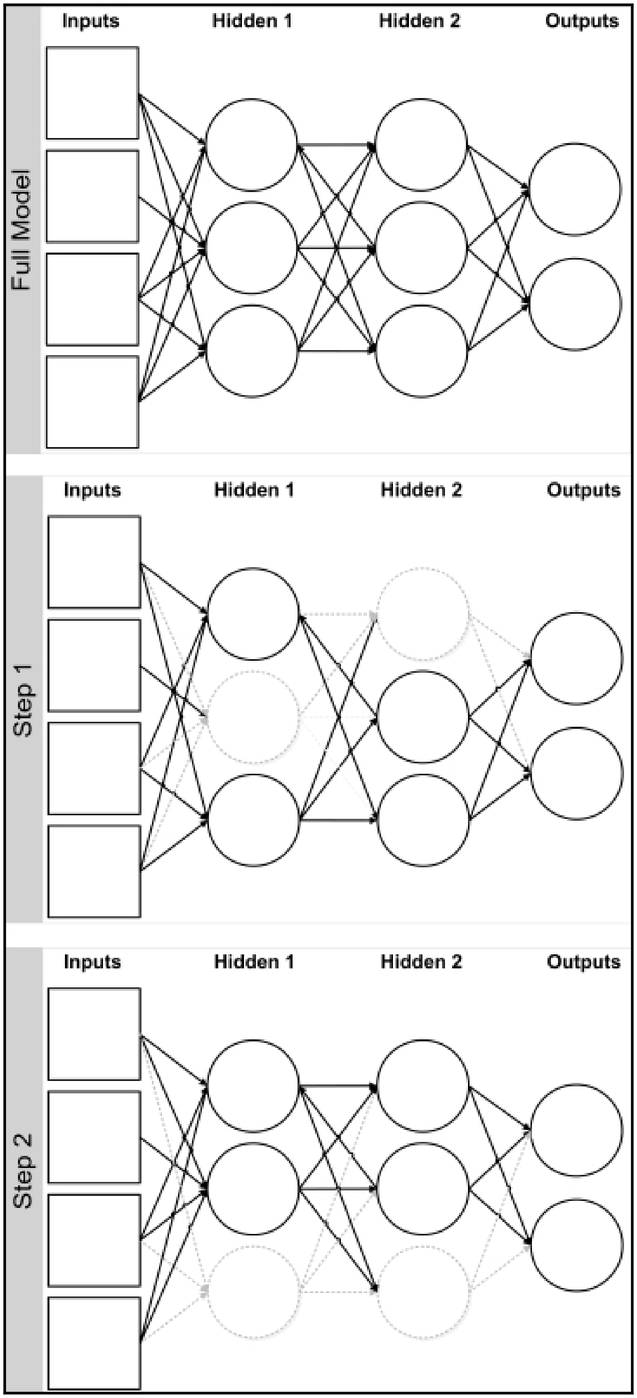
Logistic regression accuracy on bulls_eye = 0.6075

Cost function by epochs

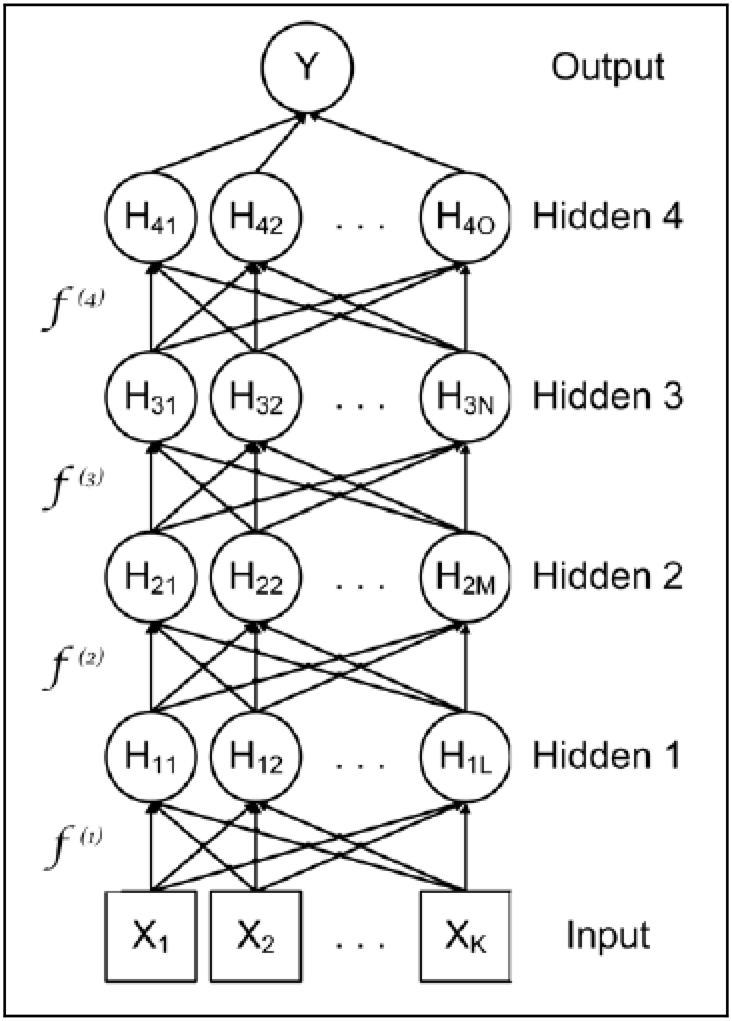
```
--BEGIN TRAINING MODEL--
  Cost after 0 epochs= 0.693
  Cost after 500 epochs= 0.423
  Cost after 1000 epochs= 0.254
  Cost after 1500 epochs= 0.236
 --END TRAINING MODEL--


Neural network Accuracy= 0.95
```

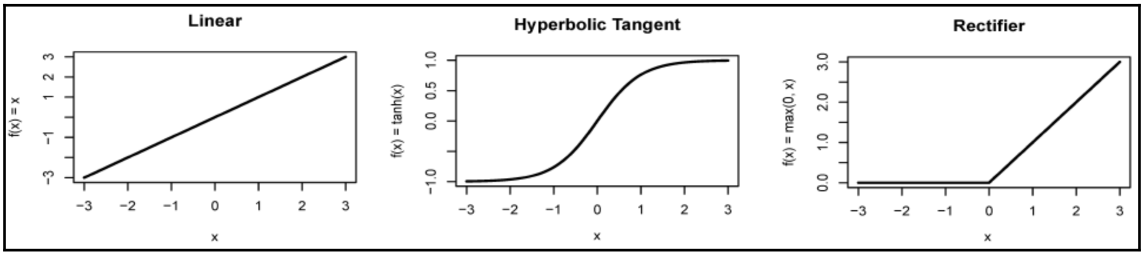

5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2
Mean-Squared Error
0
5
10
15
-3
-2
-1
0
1
log(Lambda)

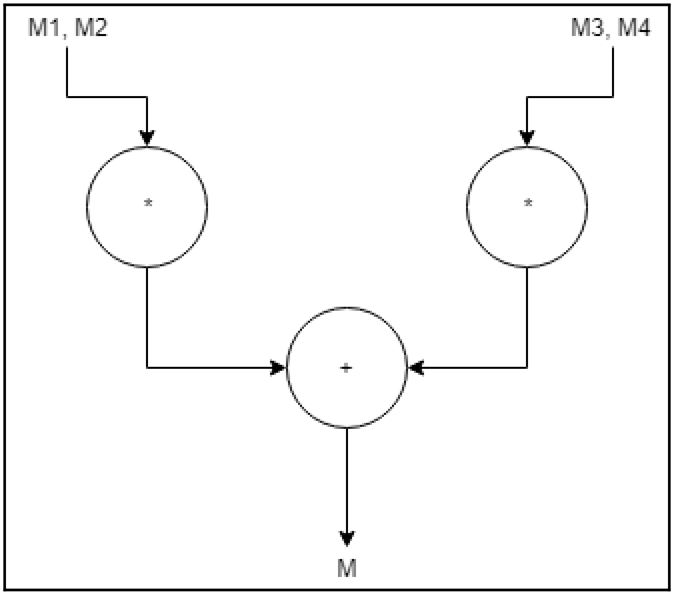

5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
Mean-Squared Error
0
5
10
15
0
2
4
6
8
log(Lambda)

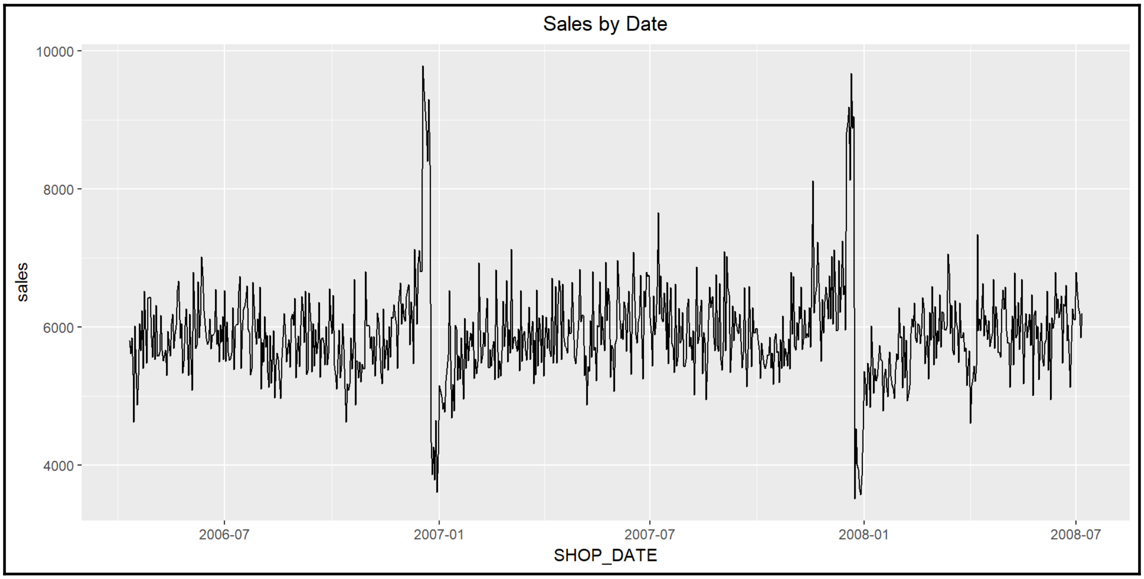

Full Model
Inputs
Hidden 1
Hidden 2
Outputs
Step 1
Inputs
Hidden 1
Hidden 2
Outputs
Step 2
Inputs
Hidden 1
Hidden 2
Outputs

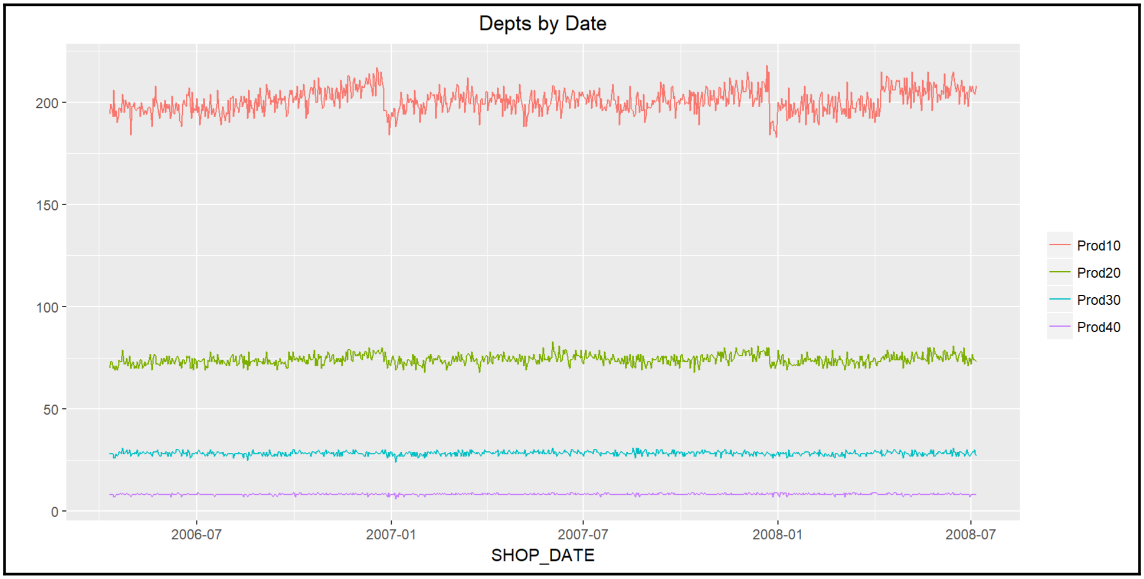
Chapter 4: Training Deep Prediction Models

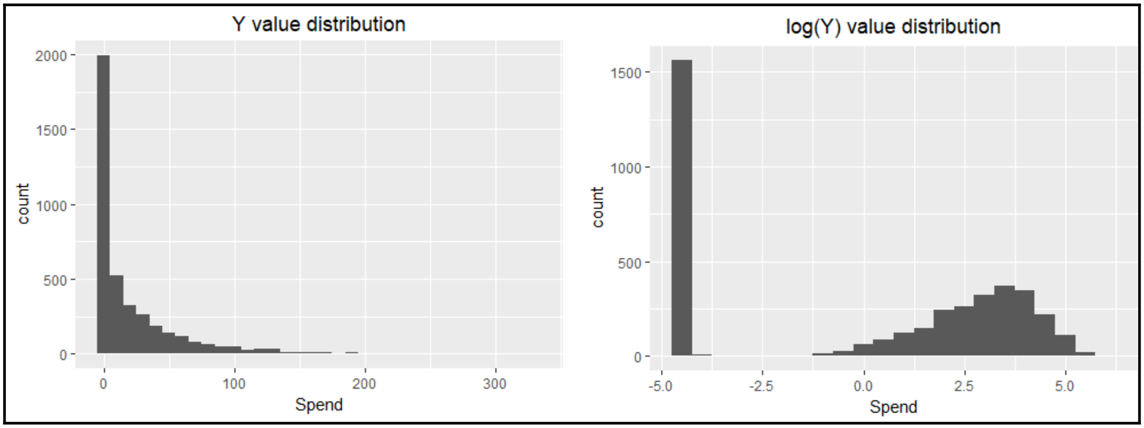


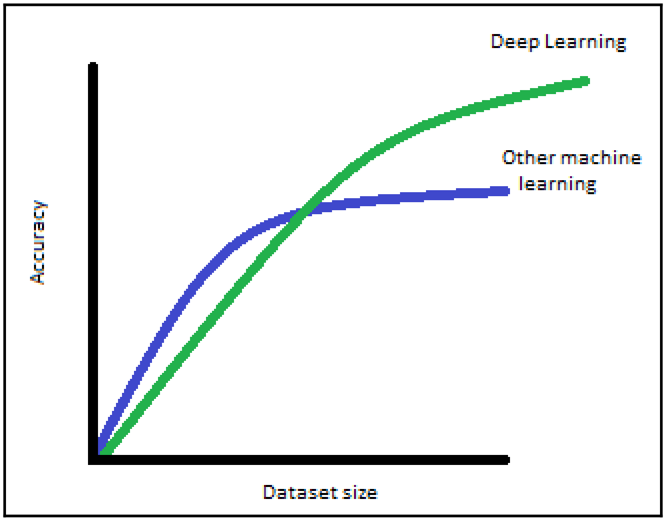



M1, M2
M3, M4
*
*
+
M

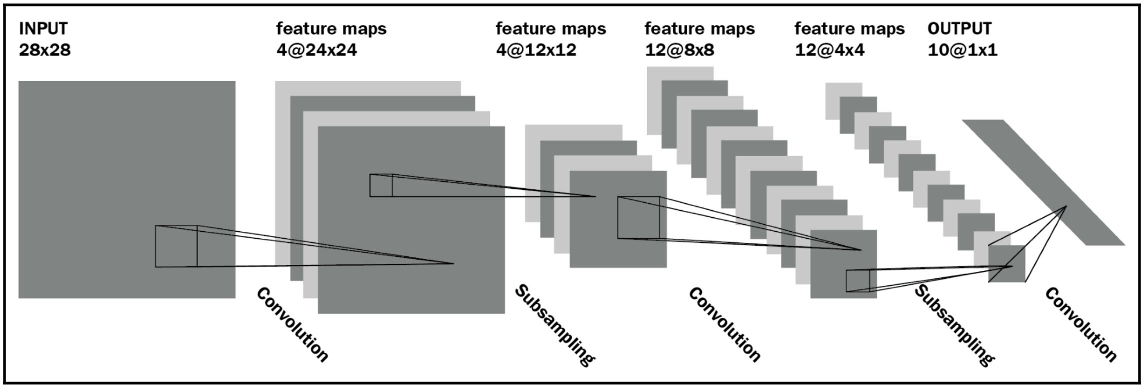

Sales by Date
10000
8000
6000
4000
sales
2006-07
2007-01
2007-07
2008-01
2008-07
SHOP_DATE

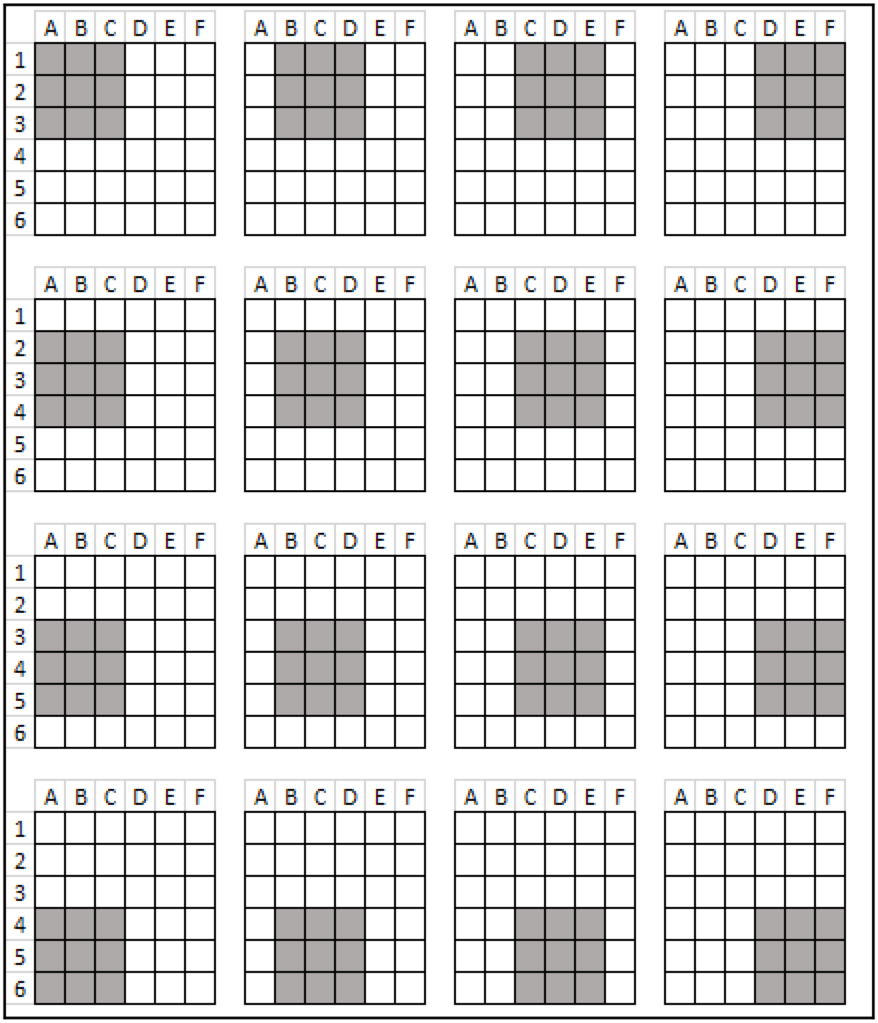

Depts by Date
200
150
100
50
0
2006-07
2007-01
2007-07
2008-01
2008-07
SHOP_DATE
Prod10
Prod20
Prod30
Prod40


Y value distribution
2000
1500
1000
500
0
count
0
100
200
300
Spend
log(Y) value distribution
1500
1000
500
0
count
-5.0
-2.5
0.0
2.5
5.0
Spend


Deep Learning
Other machine learning
Accuracy
Dataset size

Chapter 5: Image Classification Using Convolutional Neural Networks





	A	B	C	D	E	F
1	▨	▨	▨			
2	▨	▨	▨			
3	▨	▨	▨			
4						
5						
6						

	A	B	C	D	E	F
1		▨	▨	▨		
2		▨	▨	▨		
3		▨	▨	▨		
4						
5						
6						

	A	B	C	D	E	F
1			▨	▨	▨	
2			▨	▨	▨	
3			▨	▨	▨	
4						
5						
6						

	A	B	C	D	E	F
1				▨	▨	▨
2				▨	▨	▨
3				▨	▨	▨
4						
5						
6						

	A	B	C	D	E	F
1						
2	▨	▨	▨			
3	▨	▨	▨			
4	▨	▨	▨			
5						
6						

	A	B	C	D	E	F
1						
2		▨	▨	▨		
3		▨	▨	▨		
4		▨	▨	▨		
5						
6						

	A	B	C	D	E	F
1						
2			▨	▨	▨	
3			▨	▨	▨	
4			▨	▨	▨	
5						
6						

	A	B	C	D	E	F
1						
2				▨	▨	▨
3				▨	▨	▨
4				▨	▨	▨
5						
6						

	A	B	C	D	E	F
1						
2						
3	▨	▨	▨			
4	▨	▨	▨			
5	▨	▨	▨			
6						

	A	B	C	D	E	F
1						
2						
3		▨	▨	▨		
4		▨	▨	▨		
5		▨	▨	▨		
6						

	A	B	C	D	E	F
1						
2						
3			▨	▨	▨	
4			▨	▨	▨	
5			▨	▨	▨	
6						

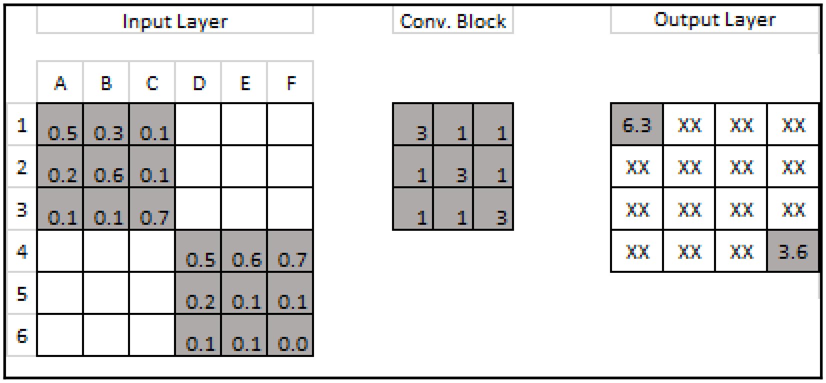
	A	B	C	D	E	F
1						
2						
3				▨	▨	▨
4				▨	▨	▨
5				▨	▨	▨
6						

	A	B	C	D	E	F
1						
2						
3						
4	▨	▨	▨			
5	▨	▨	▨			
6	▨	▨	▨			

	A	B	C	D	E	F
1						
2						
3						
4		▨	▨	▨		
5		▨	▨	▨		
6		▨	▨	▨		

	A	B	C	D	E	F
1						
2						
3						
4			▨	▨	▨	
5			▨	▨	▨	
6			▨	▨	▨	

	A	B	C	D	E	F
1						
2						
3						
4				▨	▨	▨
5				▨	▨	▨
6				▨	▨	▨



Input Layer

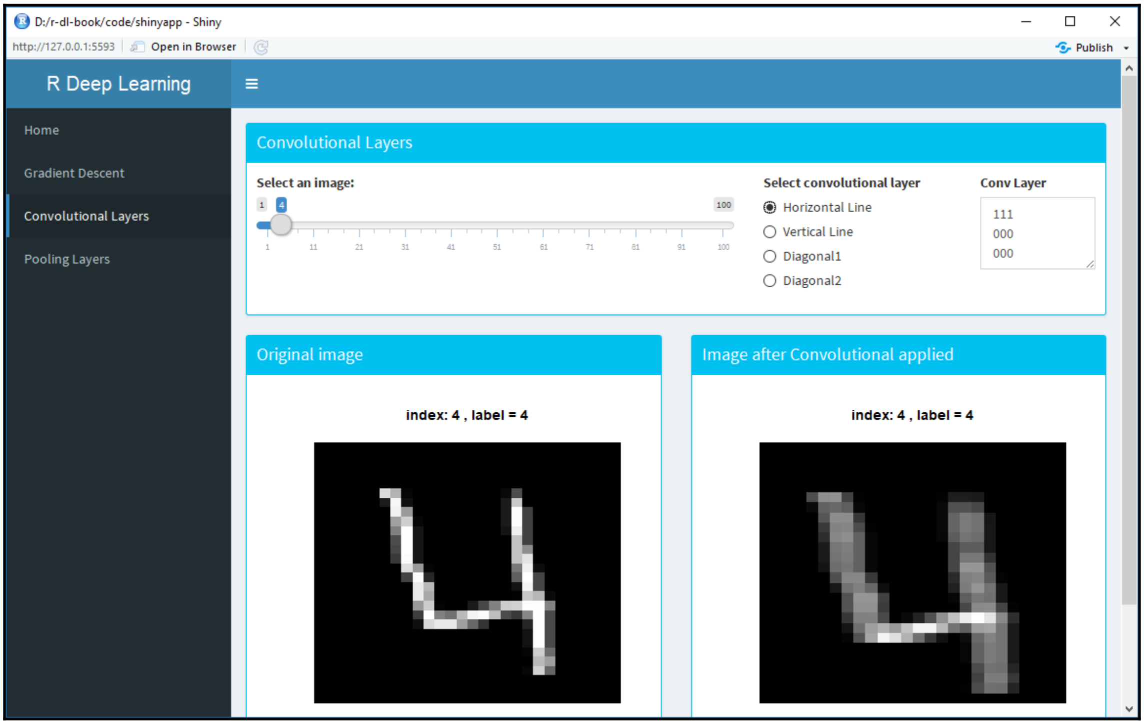
	A	B	C	D	E	F
1	0.5	0.3	0.1			
2	0.2	0.6	0.1			
3	0.1	0.1	0.7			
4				0.5	0.6	0.7
5				0.2	0.1	0.1
6				0.1	0.1	0.0

Conv. Block

3	1	1
1	3	1
1	1	3

Output Layer

6.3	XX	XX	XX
XX	XX	XX	XX
XX	XX	XX	XX
XX	XX	XX	3.6



Select an image: 4 (1–100)

Select convolutional layer

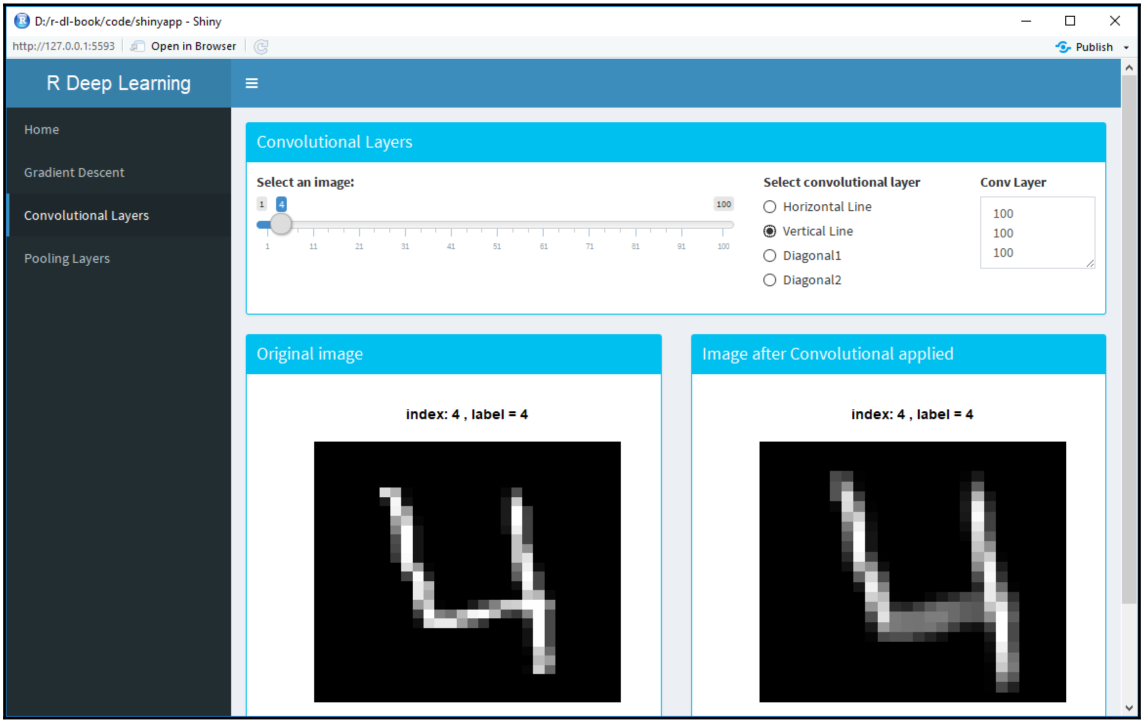
- ● Horizontal Line
- ○ Vertical Line
- ○ Diagonal1
- ○ Diagonal2

Conv Layer

```
111
000
000
```

index: 4 , label = 4

index: 4 , label = 4



Convolutional Layers

Select an image: 4

Select convolutional layer

○ Horizontal Line
● Vertical Line
○ Diagonal1
○ Diagonal2

Conv Layer

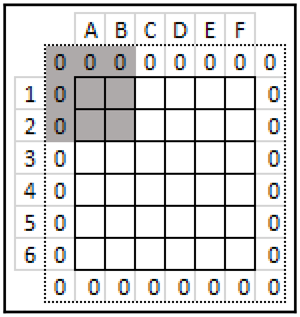
100
100
100

Original image

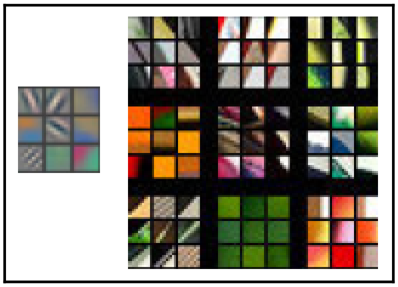
index: 4 , label = 4

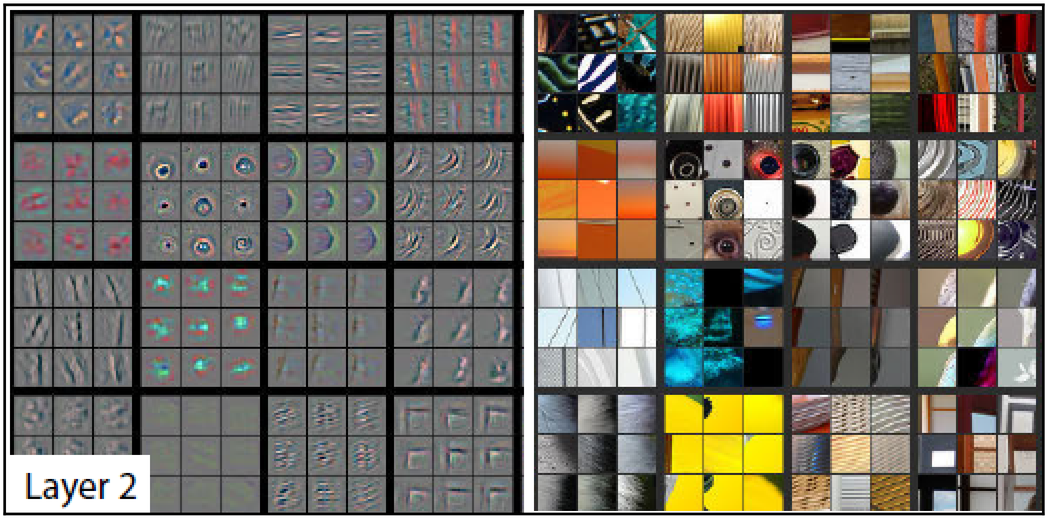
Image after Convolutional applied

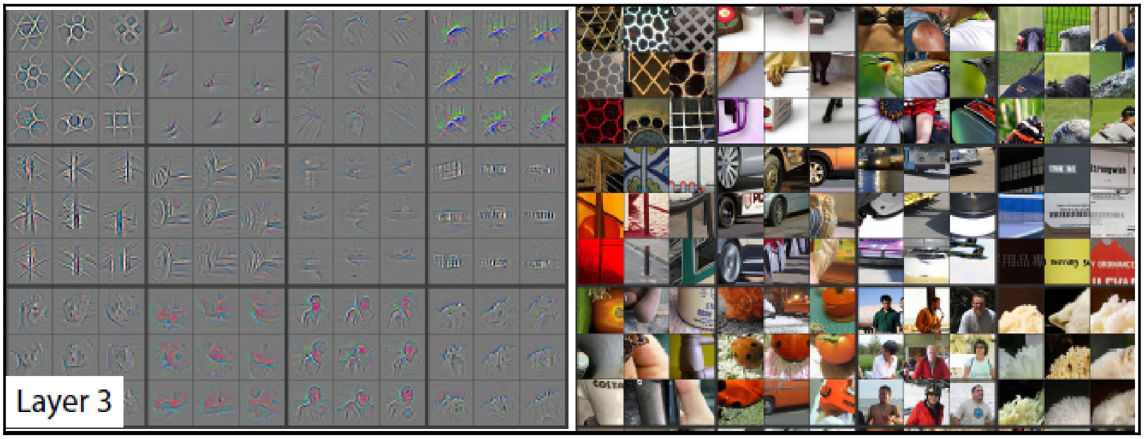
index: 4 , label = 4

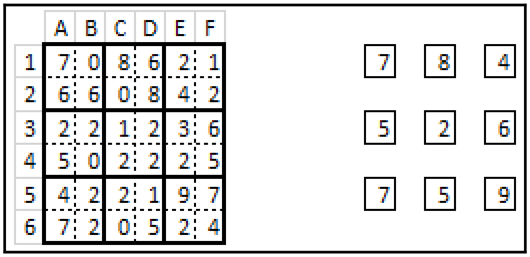


		A	B	C	D	E	F	
	0	0	0	0	0	0	0	0
1	0							0
2	0							0
3	0							0
4	0							0
5	0							0
6	0							0
	0	0	0	0	0	0	0	0



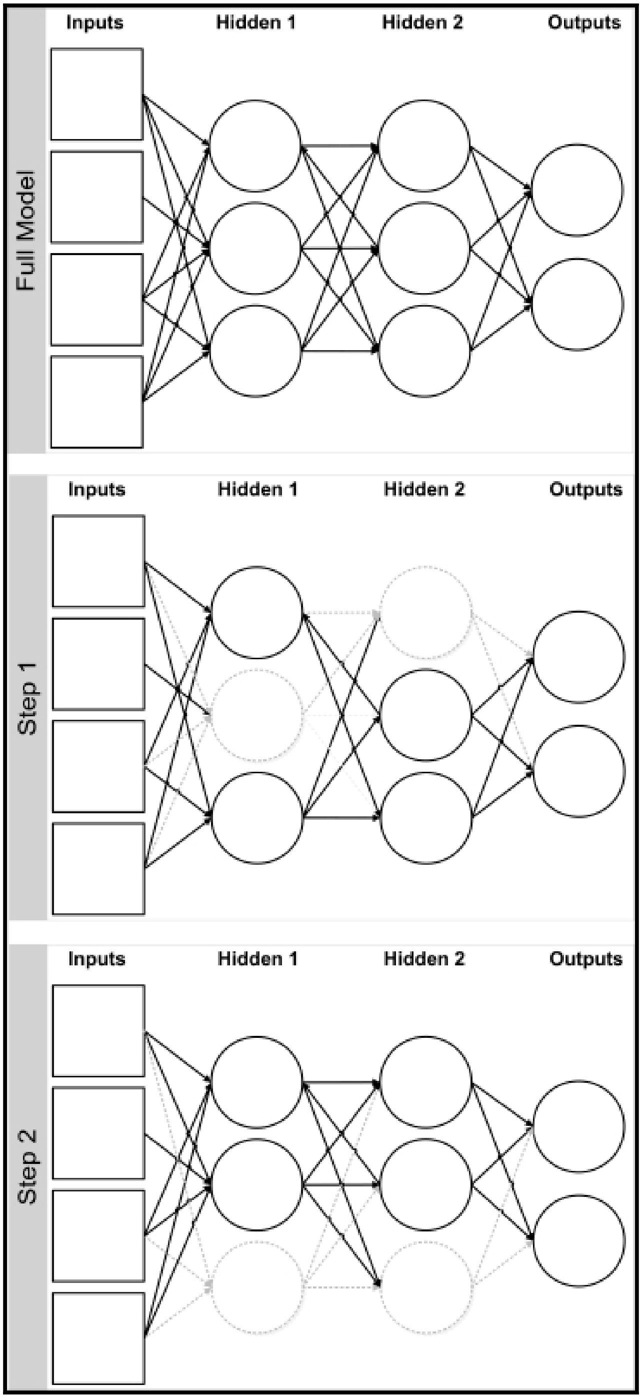


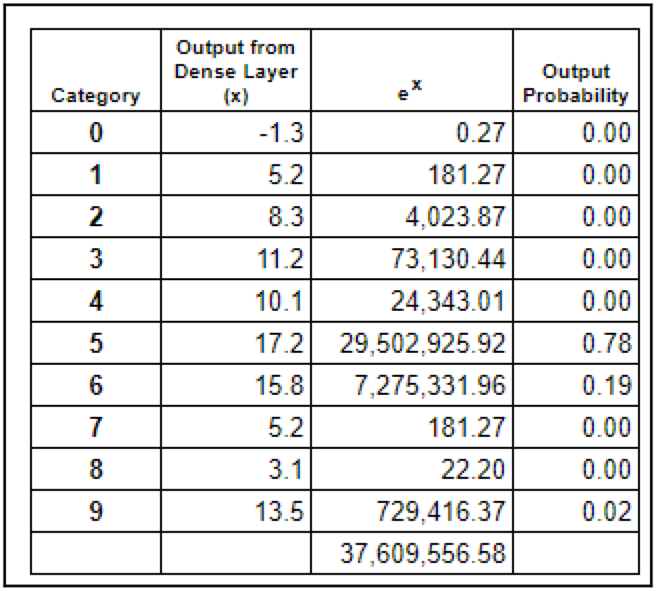




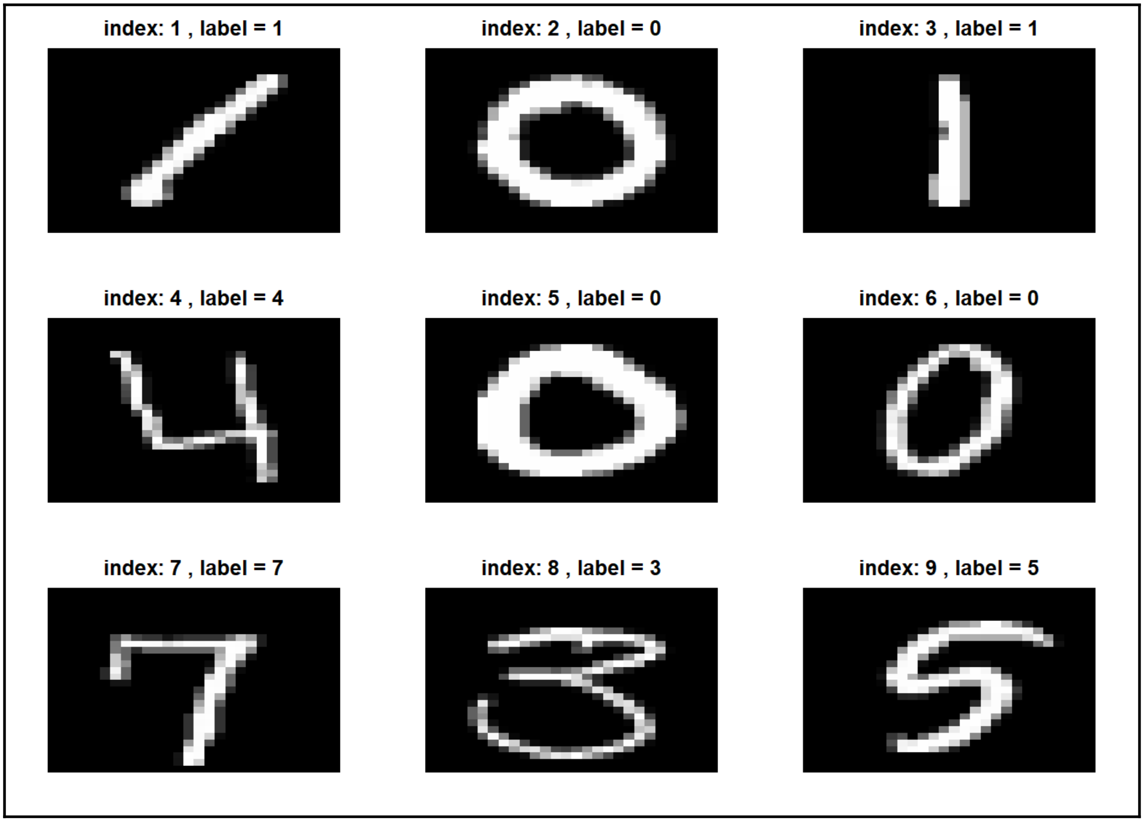
	A	B	C	D	E	F
1	7	0	8	6	2	1
2	6	6	0	8	4	2
3	2	2	1	2	3	6
4	5	0	2	2	2	5
5	4	2	2	1	9	7
6	7	2	0	5	2	4

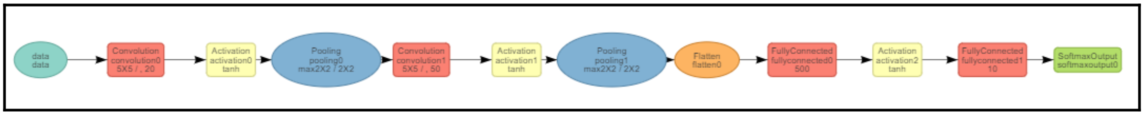
7	8	4
5	2	6
7	5	9

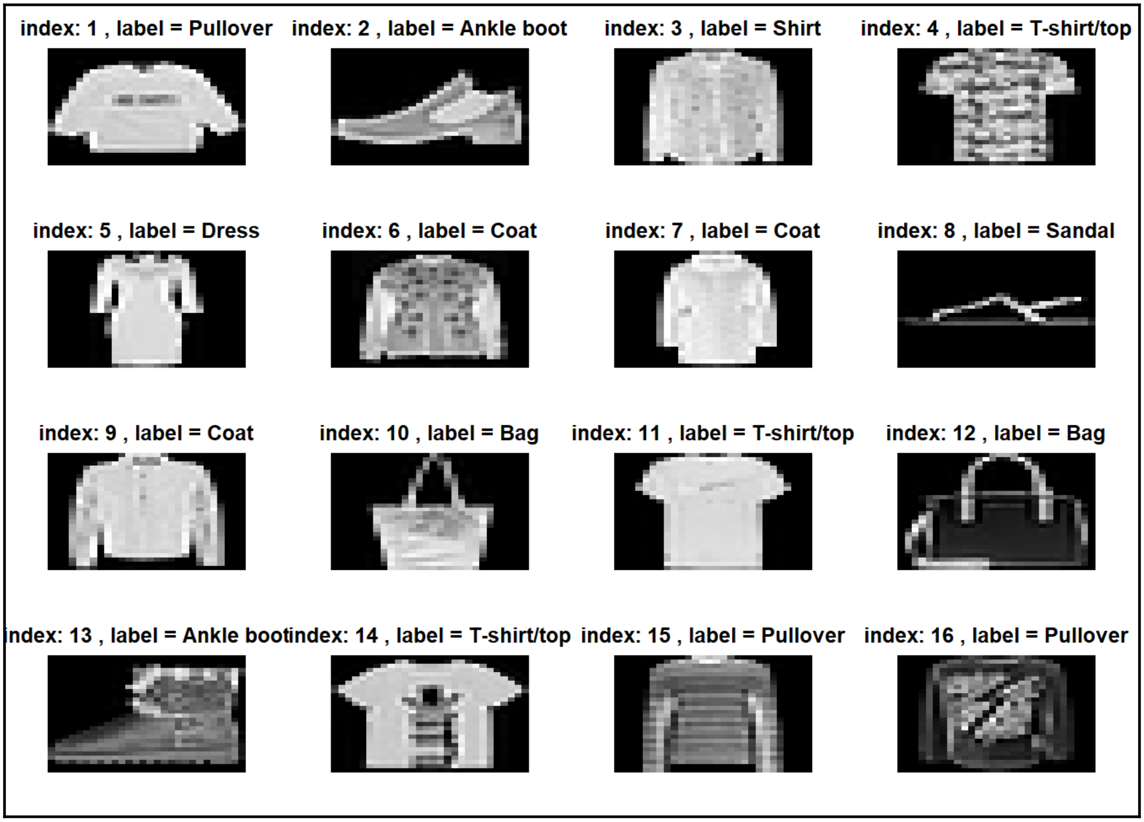

Full Model
Inputs
Hidden 1
Hidden 2
Outputs
Step 1
Inputs
Hidden 1
Hidden 2
Outputs
Step 2
Inputs
Hidden 1
Hidden 2
Outputs

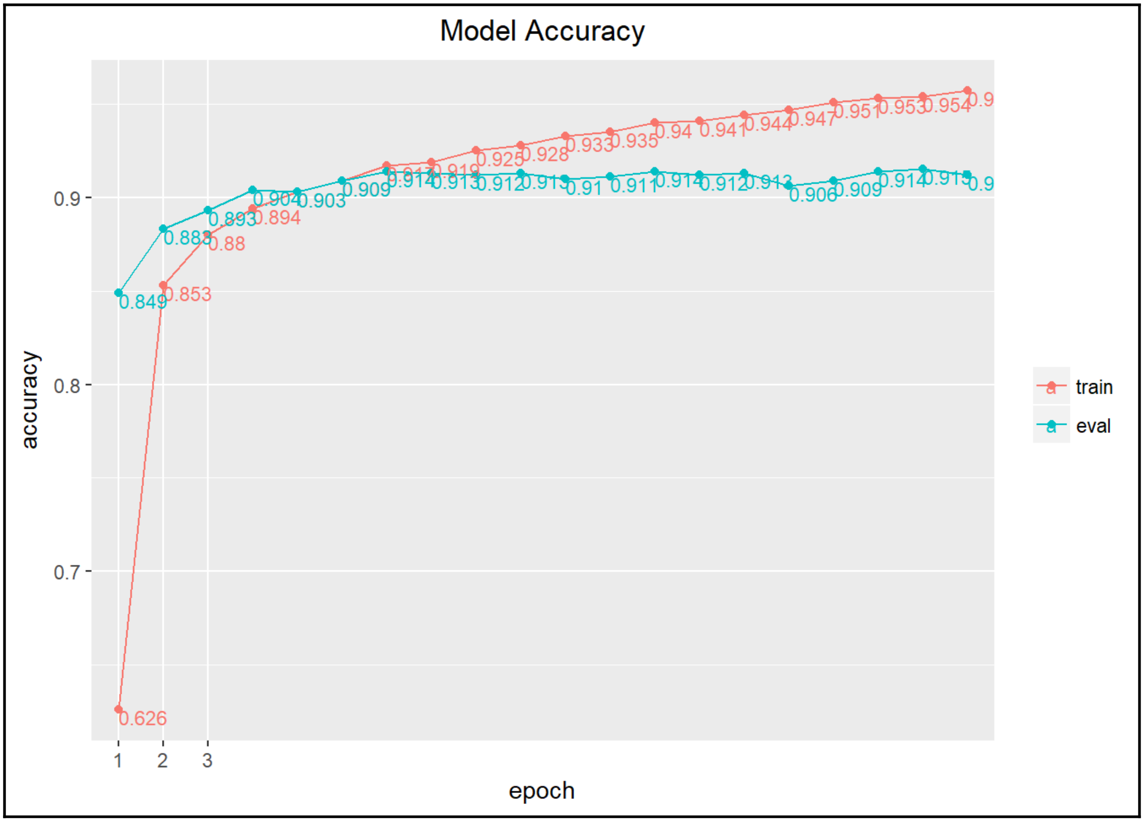


Category	Output from Dense Layer (x)	e^x	Output Probability
0	-1.3	0.27	0.00
1	5.2	181.27	0.00
2	8.3	4,023.87	0.00
3	11.2	73,130.44	0.00
4	10.1	24,343.01	0.00
5	17.2	29,502,925.92	0.78
6	15.8	7,275,331.96	0.19
7	5.2	181.27	0.00
8	3.1	22.20	0.00
9	13.5	729,416.37	0.02
		37,609,556.58	

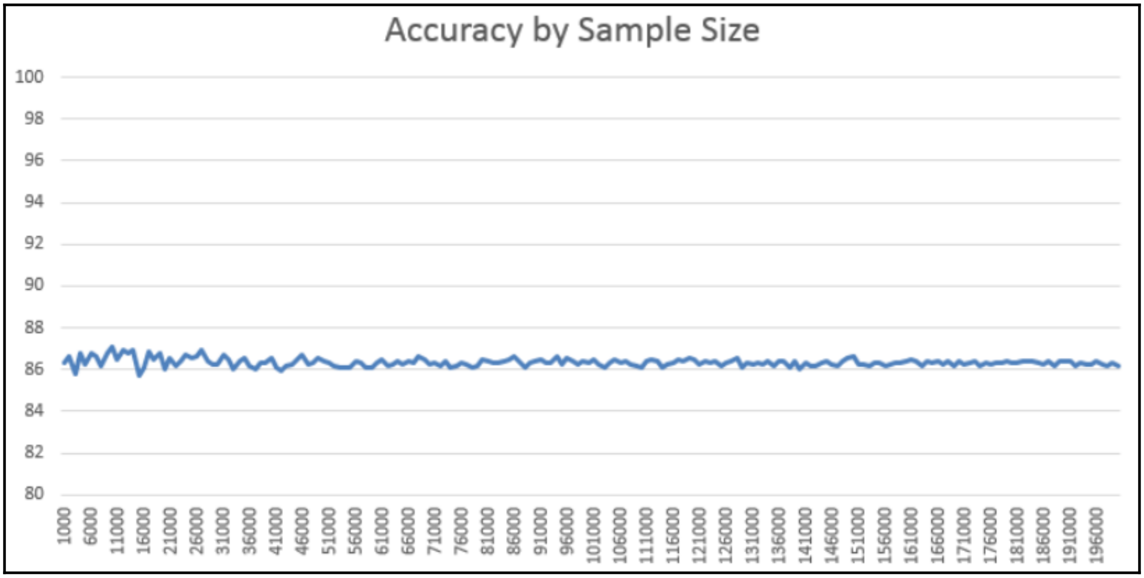

index: 1 , label = 1
index: 2 , label = 0
index: 3 , label = 1
index: 4 , label = 4
index: 5 , label = 0
index: 6 , label = 0
index: 7 , label = 7
index: 8 , label = 3
index: 9 , label = 5

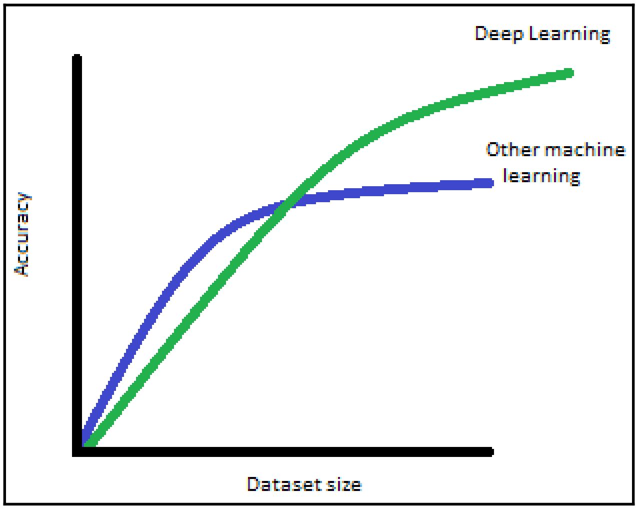

data
data
Convolution
convolution0
5X5 / , 20
Activation
activation0
tanh
Pooling
pooling0
max2X2 / 2X2
Convolution
convolution1
5X5 / , 50
Activation
activation1
tanh
Pooling
pooling1
max2X2 / 2X2
Flatten
flatten0
FullyConnected
fullyconnected0
500
Activation
activation2
tanh
FullyConnected
fullyconnected1
10
SoftmaxOutput
softmaxoutput0

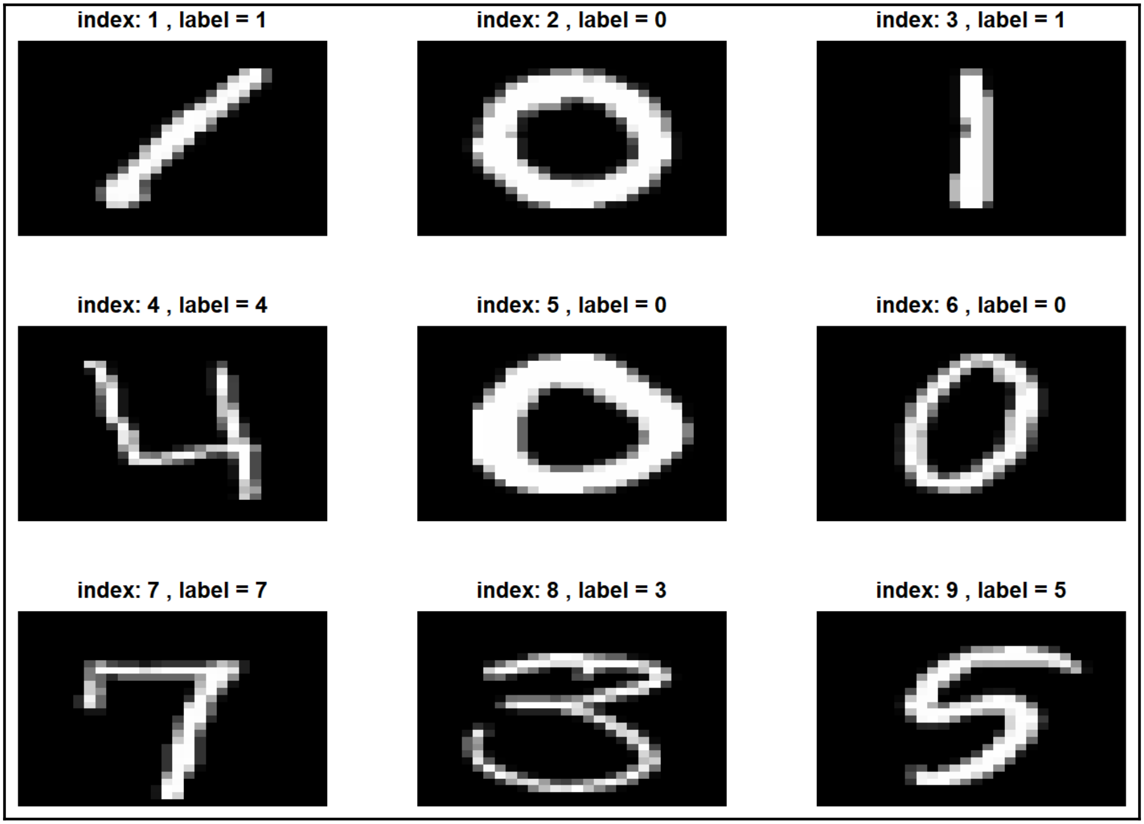

index: 1 , label = Pullover
index: 2 , label = Ankle boot
index: 3 , label = Shirt
index: 4 , label = T-shirt/top
index: 5 , label = Dress
index: 6 , label = Coat
index: 7 , label = Coat
index: 8 , label = Sandal
index: 9 , label = Coat
index: 10 , label = Bag
index: 11 , label = T-shirt/top
index: 12 , label = Bag
index: 13 , label = Ankle boot
index: 14 , label = T-shirt/top
index: 15 , label = Pullover
index: 16 , label = Pullover

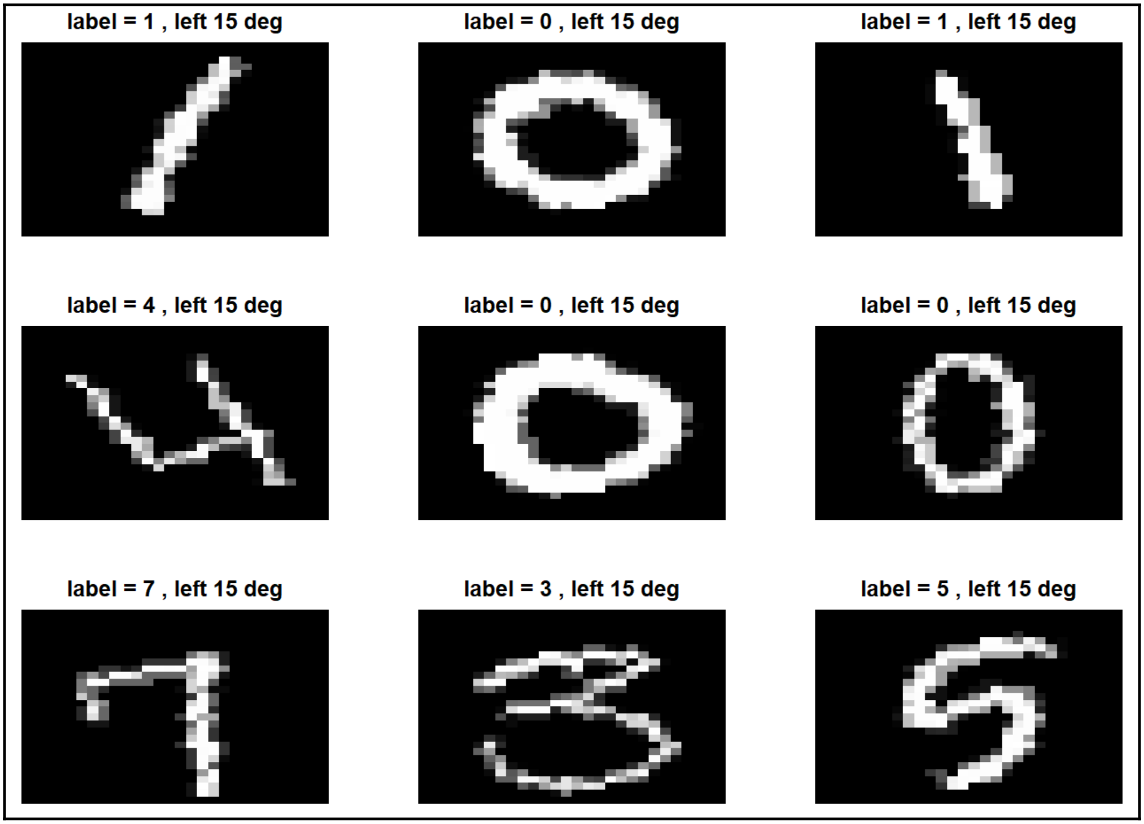

Model Accuracy
accuracy
0.9
0.8
0.7
epoch
1
2
3
train
eval
0.626
0.853
0.88
0.894
0.917
0.919
0.925
0.928
0.933
0.935
0.94
0.941
0.944
0.947
0.951
0.953
0.954
0.9
0.849
0.883
0.893
0.904
0.903
0.909
0.914
0.913
0.912
0.913
0.91
0.911
0.914
0.912
0.913
0.906
0.909
0.914
0.915
0.9

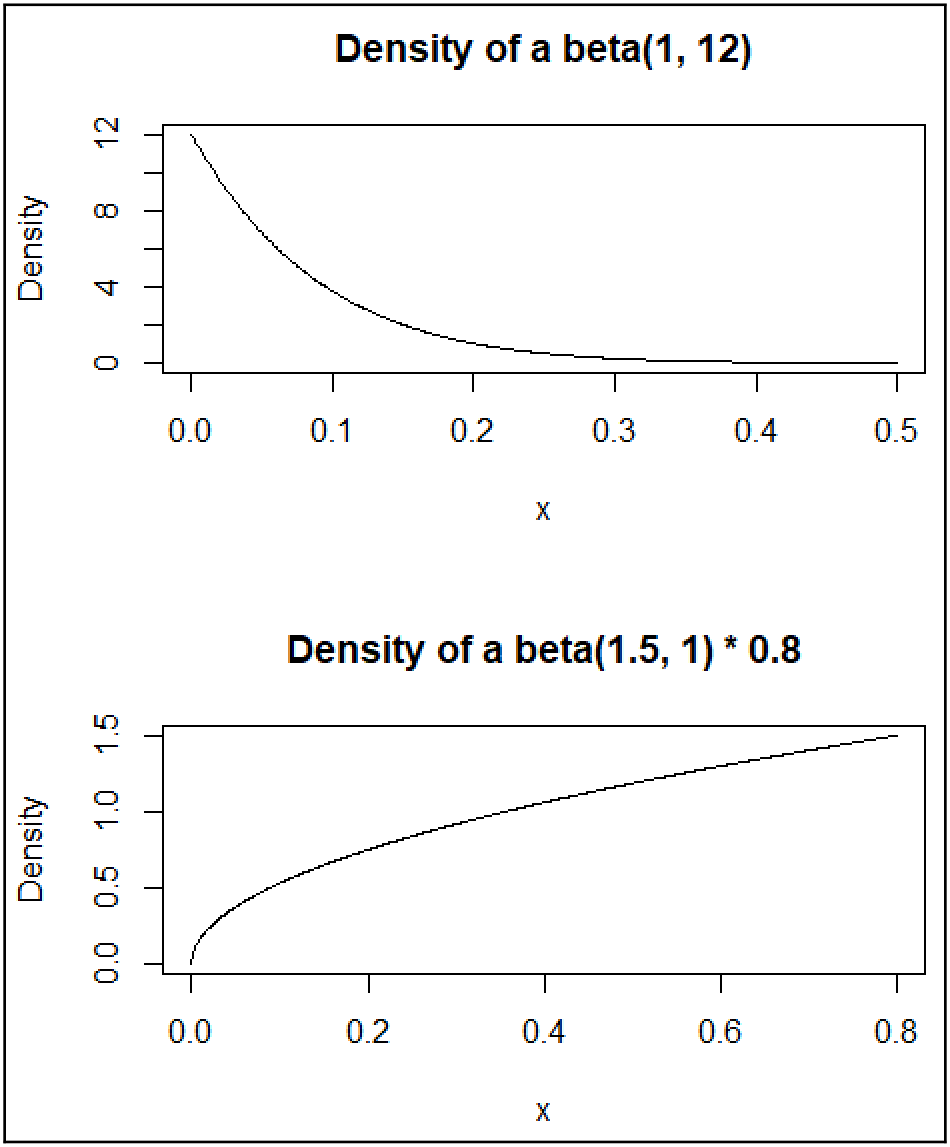
Chapter 6: Tuning and Optimizing Models

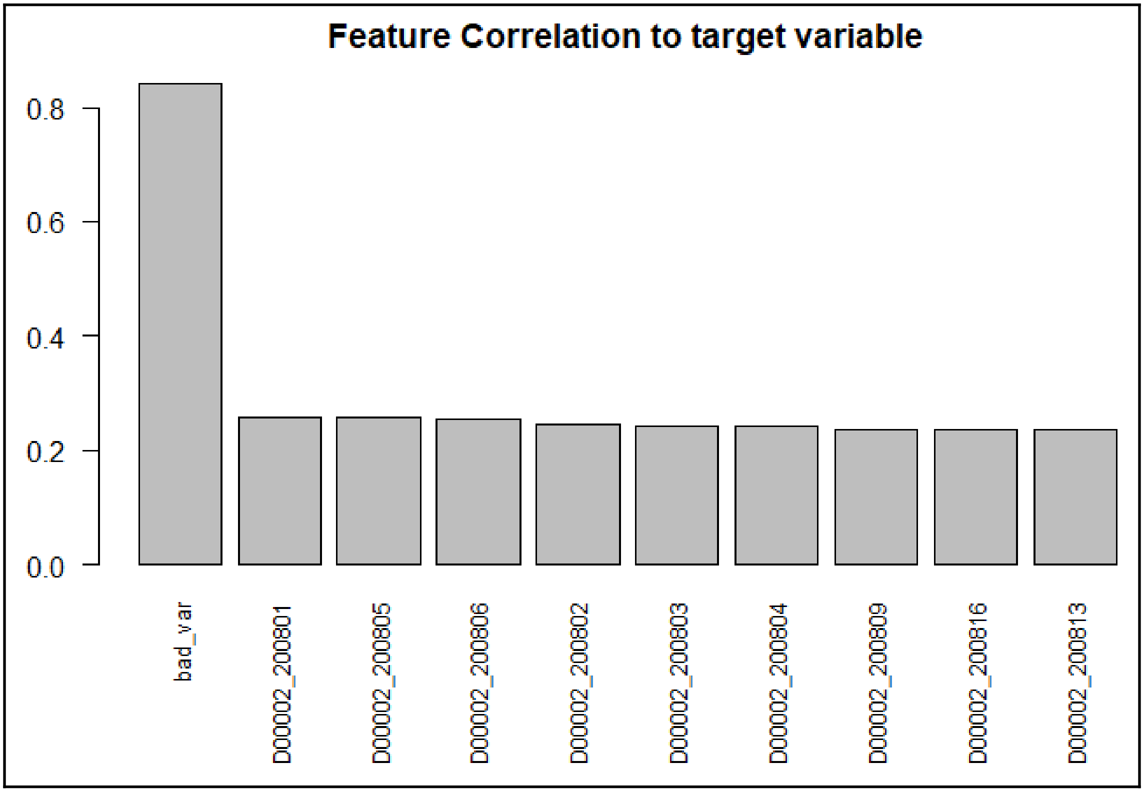


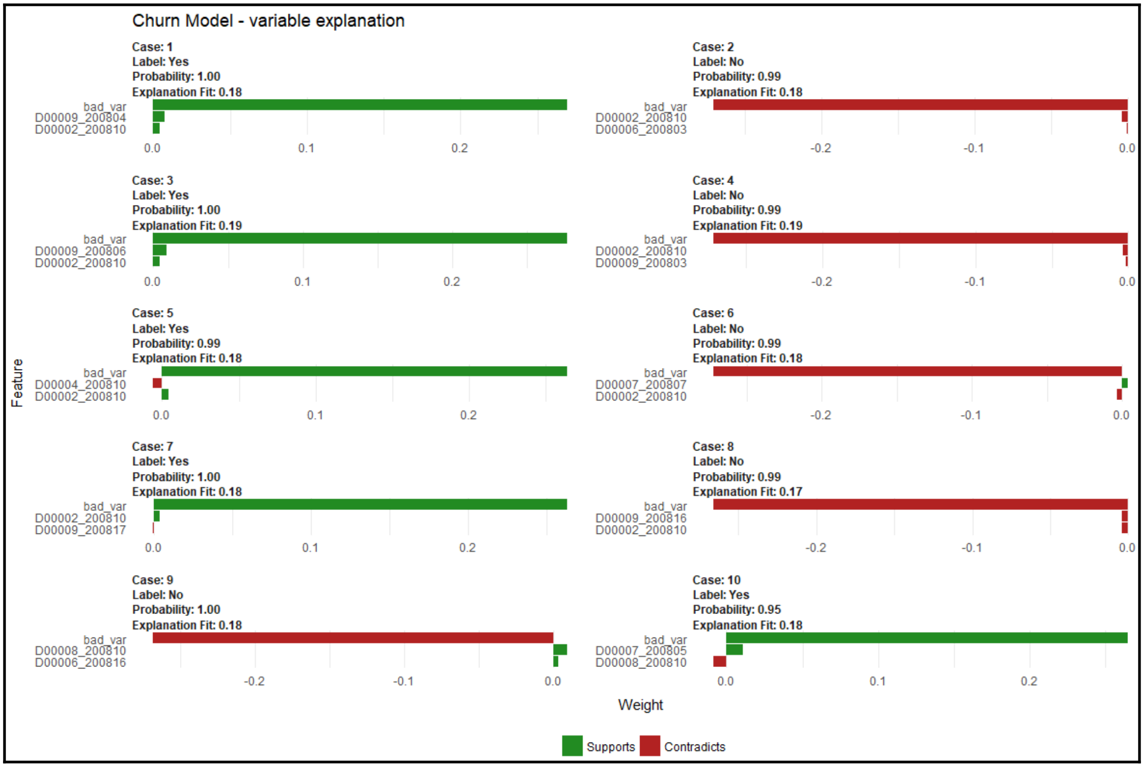


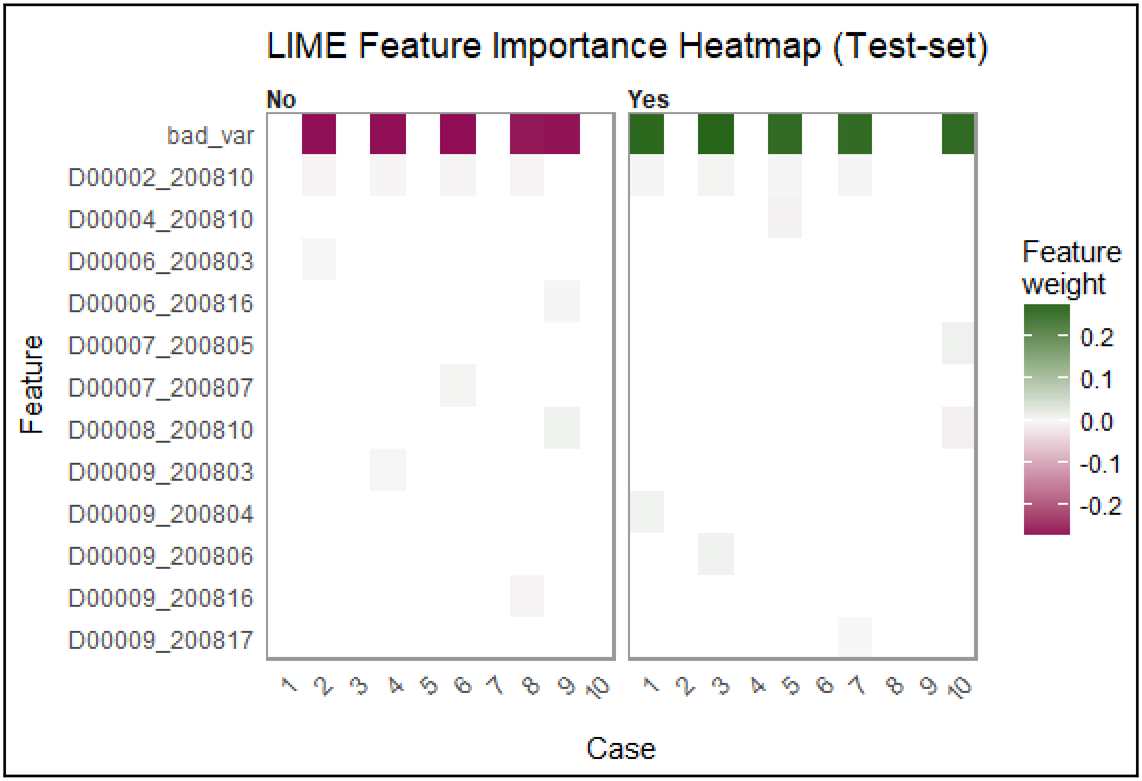

index: 1 , label = 1
index: 2 , label = 0
index: 3 , label = 1
index: 4 , label = 4
index: 5 , label = 0
index: 6 , label = 0
index: 7 , label = 7
index: 8 , label = 3
index: 9 , label = 5

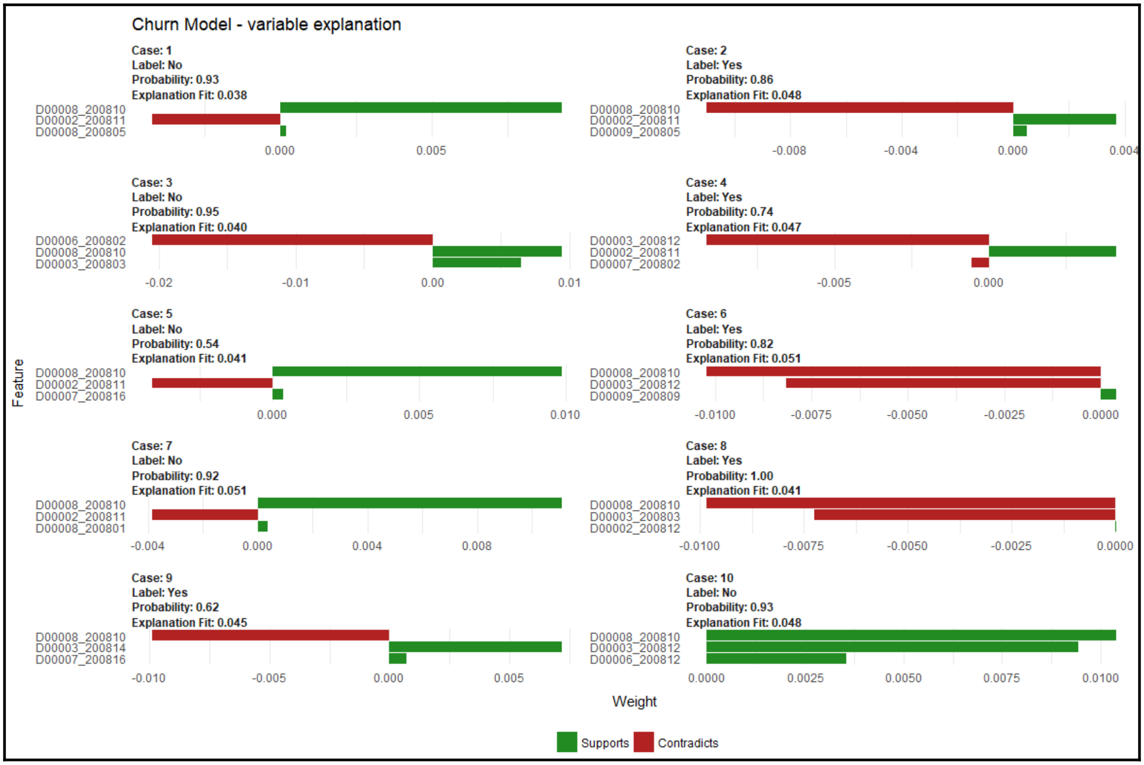

label = 1 , left 15 deg
label = 0 , left 15 deg
label = 1 , left 15 deg
label = 4 , left 15 deg
label = 0 , left 15 deg
label = 0 , left 15 deg
label = 7 , left 15 deg
label = 3 , left 15 deg
label = 5 , left 15 deg

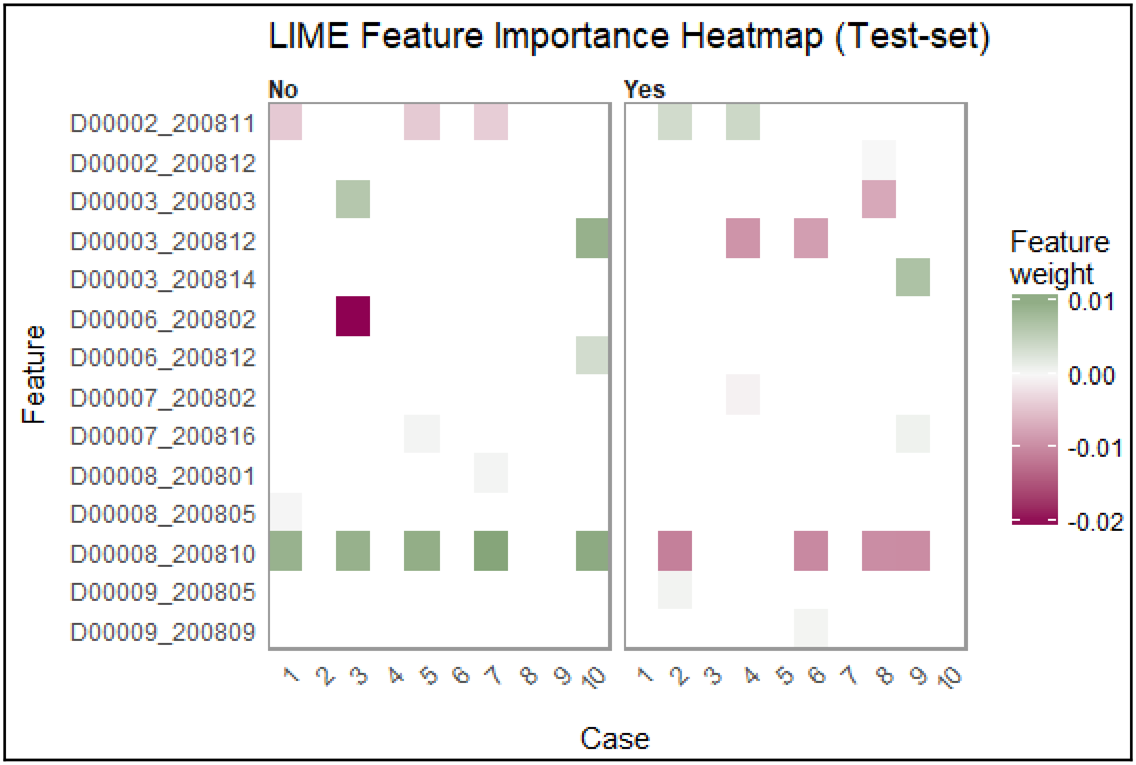

Density of a beta(1, 12)
Density
0
4
8
12
0.0
0.1
0.2
0.3
0.4
0.5
x
Density of a beta(1.5, 1) * 0.8
Density
0.0
0.5
1.0
1.5
0.0
0.2
0.4
0.6
0.8
x


Feature Correlation to target variable
0.8
0.6
0.4
0.2
0.0
bad_var
D00002_200801
D00002_200805
D00002_200806
D00002_200802
D00002_200803
D00002_200804
D00002_200809
D00002_200816
D00002_200813

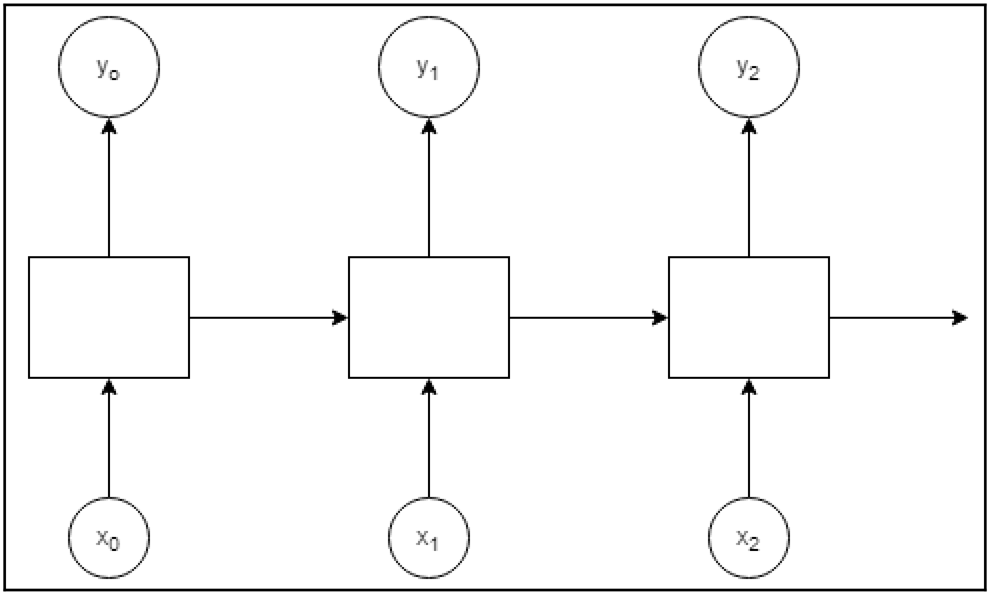

Churn Model - variable explanation
Case: 1
Label: Yes
Probability: 1.00
Explanation Fit: 0.18
bad_var
D00009_200804
D00002_200810
0.0
0.1
0.2
Case: 2
Label: No
Probability: 0.99
Explanation Fit: 0.18
bad_var
D00002_200810
D00006_200803
-0.2
-0.1
0.0
Case: 3
Label: Yes
Probability: 1.00
Explanation Fit: 0.19
bad_var
D00009_200806
D00002_200810
0.0
0.1
0.2
Case: 4
Label: No
Probability: 0.99
Explanation Fit: 0.19
bad_var
D00002_200810
D00009_200803
-0.2
-0.1
0.0
Case: 5
Label: Yes
Probability: 0.99
Explanation Fit: 0.18
bad_var
D00004_200810
D00002_200810
0.0
0.1
0.2
Case: 6
Label: No
Probability: 0.99
Explanation Fit: 0.18
bad_var
D00007_200807
D00002_200810
-0.2
-0.1
0.0
Case: 7
Label: Yes
Probability: 1.00
Explanation Fit: 0.18
bad_var
D00002_200810
D00009_200817
0.0
0.1
0.2
Case: 8
Label: No
Probability: 0.99
Explanation Fit: 0.17
bad_var
D00009_200816
D00002_200810
-0.2
-0.1
0.0
Case: 9
Label: No
Probability: 1.00
Explanation Fit: 0.18
bad_var
D00008_200810
D00006_200816
-0.2
-0.1
0.0
Case: 10
Label: Yes
Probability: 0.95
Explanation Fit: 0.18
bad_var
D00007_200805
D00008_200810
0.0
0.1
0.2
Feature
Weight
🟩 Supports
🟥 Contradicts


LIME Feature Importance Heatmap (Test-set)
No
Yes
bad_var
D00002_200810
D00004_200810
D00006_200803
D00006_200816
D00007_200805
D00007_200807
D00008_200810
D00009_200803
D00009_200804
D00009_200806
D00009_200816
D00009_200817
Feature
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Case
Feature weight
0.2
0.1
0.0
-0.1
-0.2

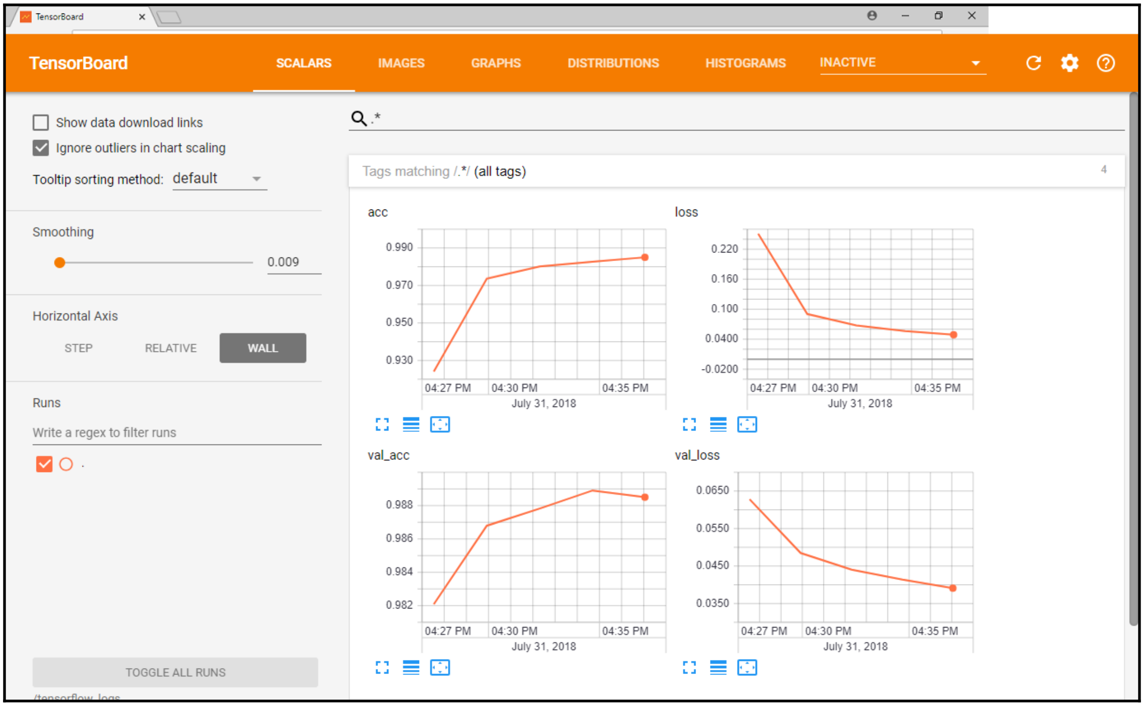

Churn Model - variable explanation
Case: 1
Label: No
Probability: 0.93
Explanation Fit: 0.038
D00008_200810
D00002_200811
D00008_200805
0.000
0.005
Case: 2
Label: Yes
Probability: 0.86
Explanation Fit: 0.048
D00008_200810
D00002_200811
D00009_200805
-0.008
-0.004
0.000
0.004
Case: 3
Label: No
Probability: 0.95
Explanation Fit: 0.040
D00006_200802
D00008_200810
D00003_200803
-0.02
-0.01
0.00
0.01
Case: 4
Label: Yes
Probability: 0.74
Explanation Fit: 0.047
D00003_200812
D00002_200811
D00007_200802
-0.005
0.000
Case: 5
Label: No
Probability: 0.54
Explanation Fit: 0.041
D00008_200810
D00002_200811
D00007_200816
0.000
0.005
0.010
Case: 6
Label: Yes
Probability: 0.82
Explanation Fit: 0.051
D00008_200810
D00003_200812
D00009_200809
-0.0100
-0.0075
-0.0050
-0.0025
0.0000
Case: 7
Label: No
Probability: 0.92
Explanation Fit: 0.051
D00008_200810
D00002_200811
D00008_200801
-0.004
0.000
0.004
0.008
Case: 8
Label: Yes
Probability: 1.00
Explanation Fit: 0.041
D00008_200810
D00003_200803
D00002_200812
-0.0100
-0.0075
-0.0050
-0.0025
0.0000
Case: 9
Label: Yes
Probability: 0.62
Explanation Fit: 0.045
D00008_200810
D00003_200814
D00007_200816
-0.010
-0.005
0.000
0.005
Case: 10
Label: No
Probability: 0.93
Explanation Fit: 0.048
D00008_200810
D00003_200812
D00006_200812
0.0000
0.0025
0.0050
0.0075
0.0100
Feature
Weight
🟩 Supports
🟥 Contradicts


LIME Feature Importance Heatmap (Test-set)
No
Yes
Feature
D00002_200811
D00002_200812
D00003_200803
D00003_200812
D00003_200814
D00006_200802
D00006_200812
D00007_200802
D00007_200816
D00008_200801
D00008_200805
D00008_200810
D00009_200805
D00009_200809
1
2
3
4
5
6
7
8
9
10
Case
Feature weight
0.01
0.00
-0.01
-0.02

Chapter 7: Natural Language Processing Using Deep Learning



Chapter 8: Deep Learning Models Using TensorFlow in R



□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.009

Horizontal Axis: STEP RELATIVE **WALL**

Runs

Write a regex to filter runs

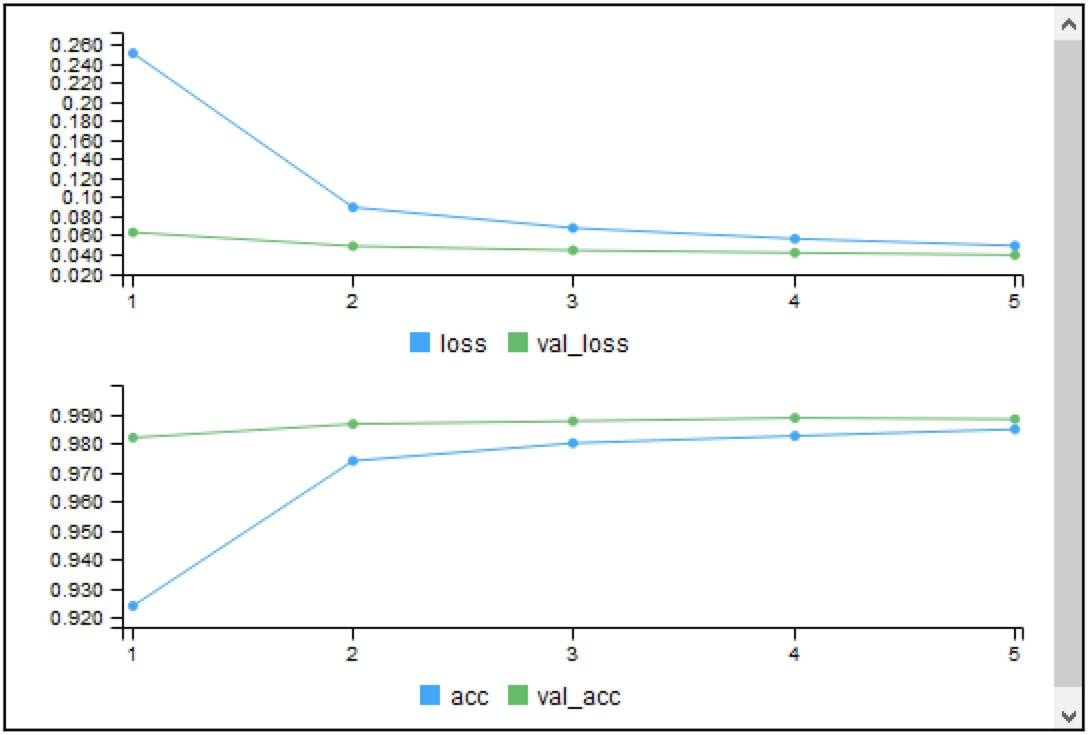
☑ ○ .

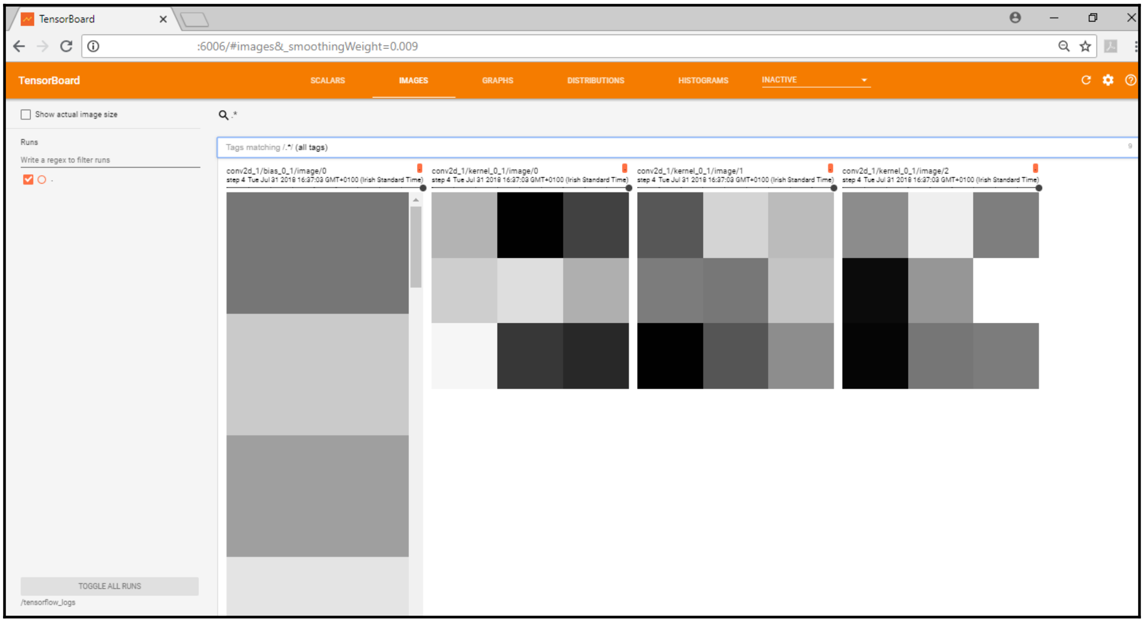
TOGGLE ALL RUNS

/tensorflow_logs

.*

Tags matching /.*/ (all tags) 4





:6006/#images&_smoothingWeight=0.009

.*

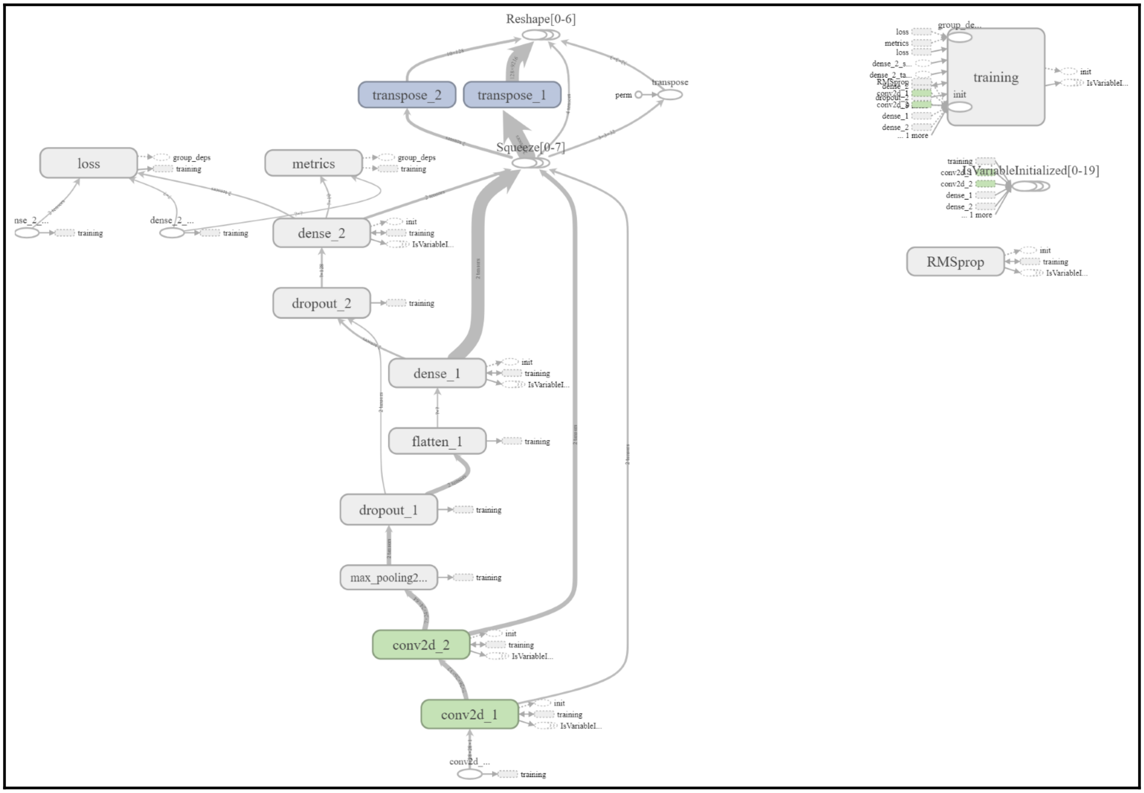
Tags matching /.*/ (all tags)

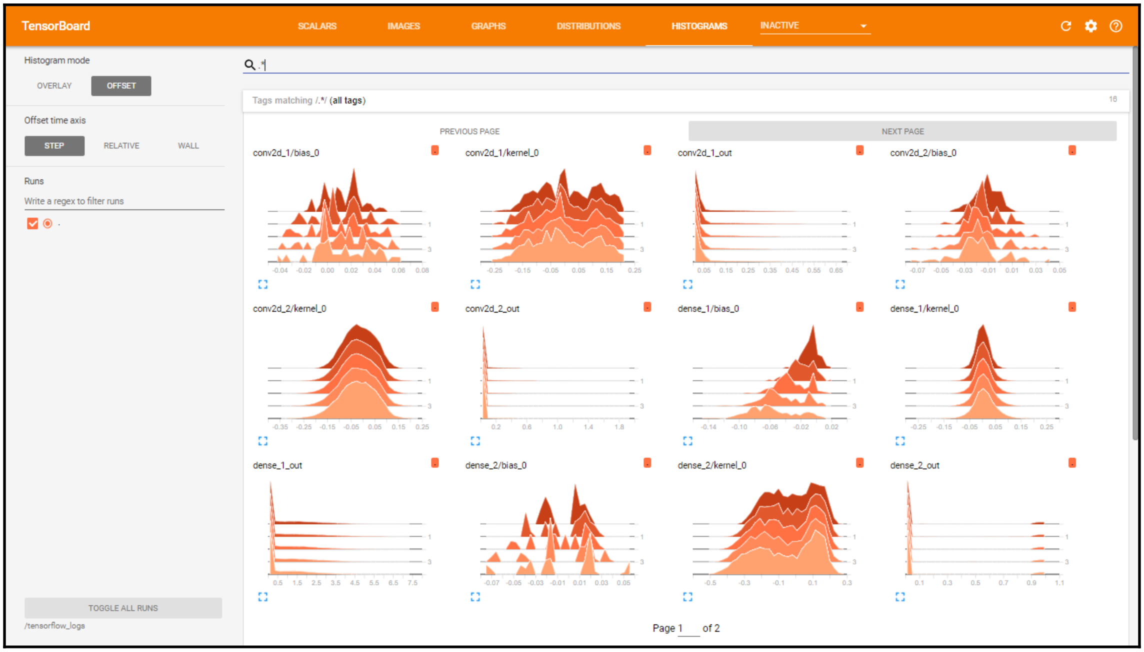
conv2d_1/bias_0_1/image/0
step 4 Tue Jul 31 2018 16:37:03 GMT+0100 (Irish Standard Time)

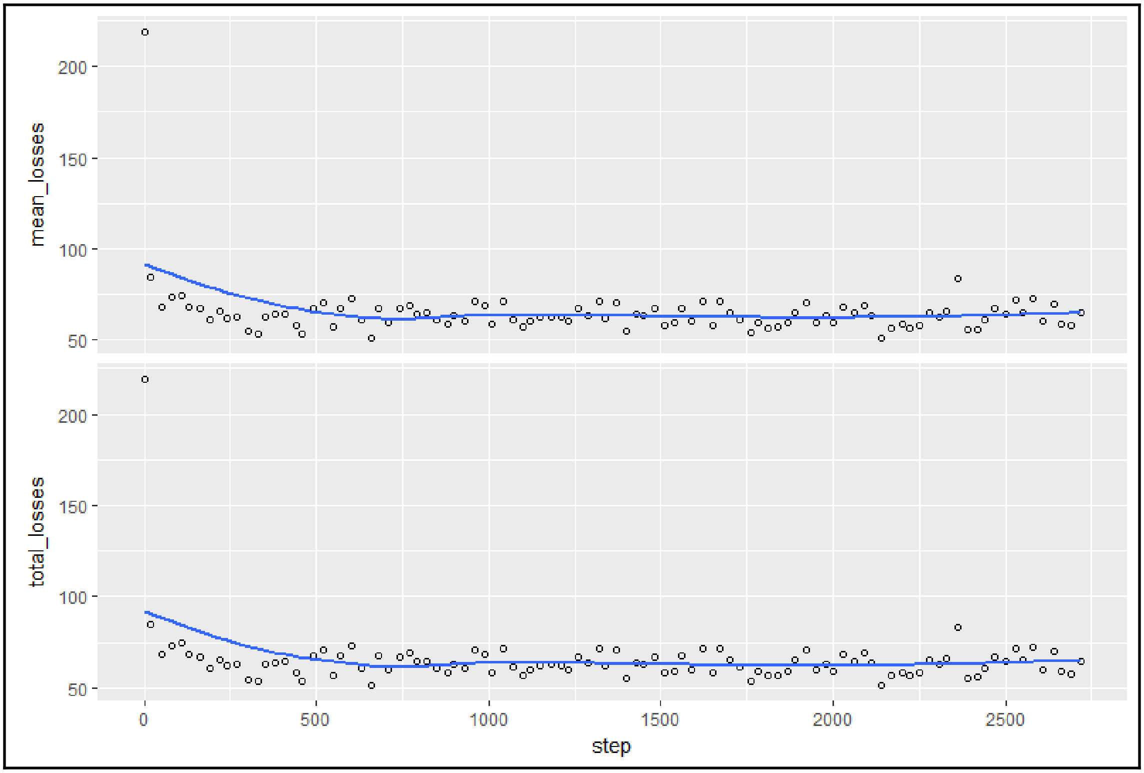
conv2d_1/kernel_0_1/image/0
step 4 Tue Jul 31 2018 16:37:03 GMT+0100 (Irish Standard Time)

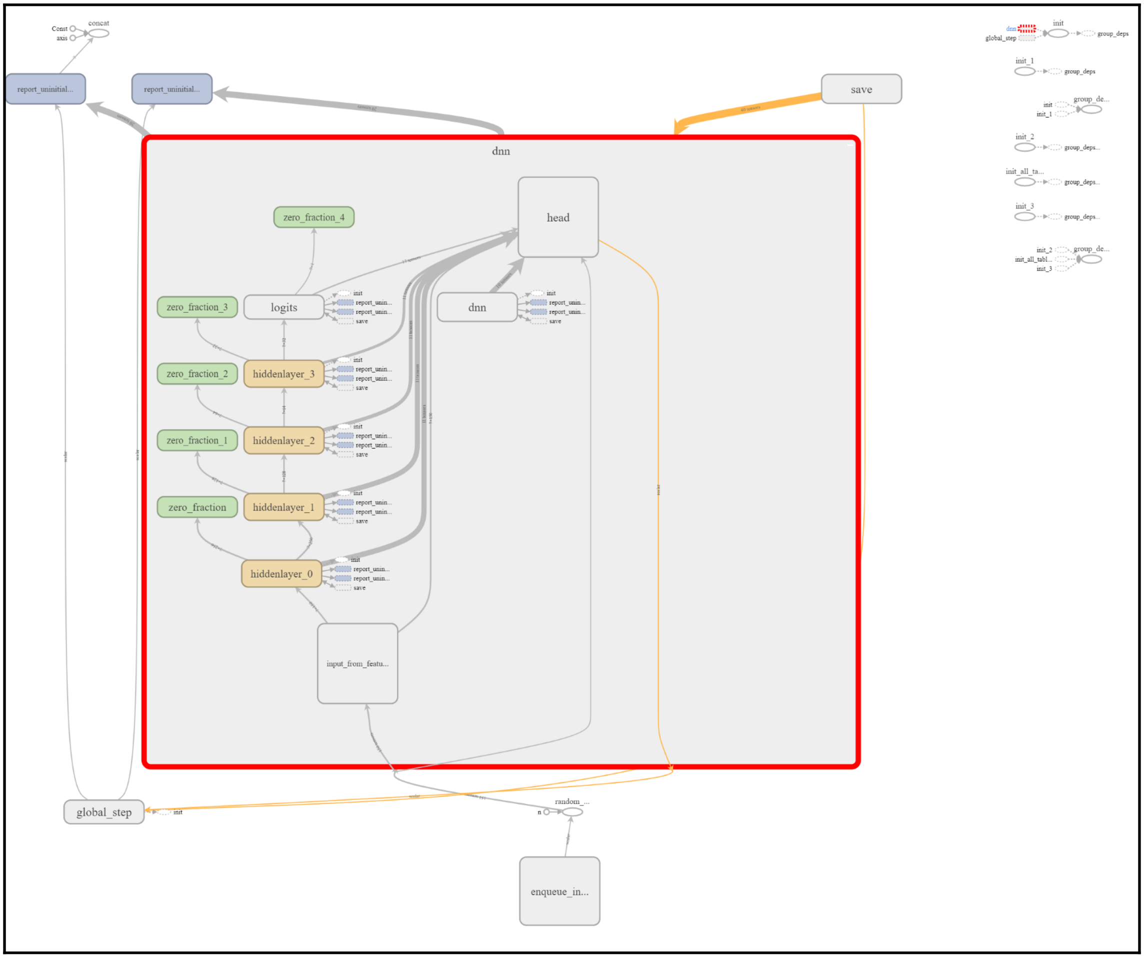
conv2d_1/kernel_0_1/image/1
step 4 Tue Jul 31 2018 16:37:03 GMT+0100 (Irish Standard Time)

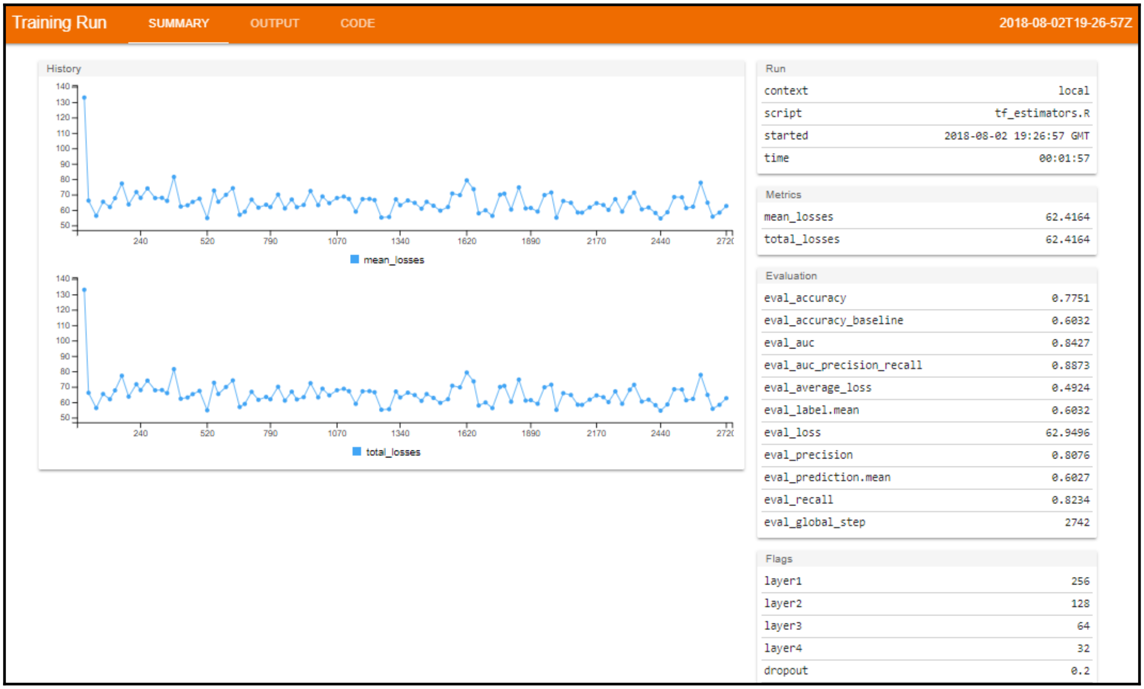
conv2d_1/kernel_0_1/image/2
step 4 Tue Jul 31 2018 16:37:03 GMT+0100 (Irish Standard Time)


Reshape[0-6]
transpose_2
transpose_1
transpose
perm
Squeeze[0-7]
loss
group_deps
training
metrics
group_deps
training
nse_2_...
training
dense_2_...
training
dense_2
init
training
IsVariableI...
dropout_2
training
dense_1
init
training
IsVariableI...
flatten_1
training
dropout_1
training
max_pooling2...
training
conv2d_2
init
training
IsVariableI...
conv2d_1
init
training
IsVariableI...
conv2d_...
training
group_de...
loss
metrics
loss
dense_2_s...
dense_2_ta...
RMSprop
dense_1
dropout_1
conv2d_1
conv2d_2
dense_1
dense_2
... 1 more
training
init
init
IsVariableI...
training
conv2d_1
conv2d_2
dense_1
dense_2
... 1 more
IsVariableInitialized[0-19]
RMSprop
init
training
IsVariableI...


TensorBoard
SCALARS
IMAGES
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
INACTIVE
Histogram mode
OVERLAY
OFFSET
Offset time axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
TOGGLE ALL RUNS
/tensorflow_logs
.*
Tags matching /.*/ (all tags)
16
PREVIOUS PAGE
NEXT PAGE
conv2d_1/bias_0
-0.04 -0.02 0.00 0.02 0.04 0.06 0.08
conv2d_1/kernel_0
-0.25 -0.15 -0.05 0.05 0.15 0.25
conv2d_1_out
0.05 0.15 0.25 0.35 0.45 0.55 0.65
conv2d_2/bias_0
-0.07 -0.05 -0.03 -0.01 0.01 0.03 0.05
conv2d_2/kernel_0
-0.35 -0.25 -0.15 -0.05 0.05 0.15 0.25
conv2d_2_out
0.2 0.6 1.0 1.4 1.8
dense_1/bias_0
-0.14 -0.10 -0.06 -0.02 0.02
dense_1/kernel_0
-0.25 -0.15 -0.05 0.05 0.15 0.25
dense_1_out
0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5
dense_2/bias_0
-0.07 -0.05 -0.03 -0.01 0.01 0.03 0.05
dense_2/kernel_0
-0.5 -0.3 -0.1 0.1 0.3
dense_2_out
0.1 0.3 0.5 0.7 0.9 1.1
1
3
Page 1 of 2


mean_losses
200
150
100
50
total_losses
200
150
100
50
0
500
1000
1500
2000
2500
step


Const
axis
concat
report_uninitial...
report_uninitial...
save
dnn
zero_fraction_4
head
zero_fraction_3
logits
dnn
init
report_unin...
report_unin...
save
zero_fraction_2
hiddenlayer_3
zero_fraction_1
hiddenlayer_2
zero_fraction
hiddenlayer_1
hiddenlayer_0
input_from_featu...
global_step
init
random_...
n
enqueue_in...
dnn
global_step
init
group_deps
init_1
group_deps
init
init_1
group_de...
init_2
group_deps...
init_all_ta...
group_deps...
init_3
group_deps...
init_2
init_all_tabl...
init_3
group_de...



Run

context	local
script	tf_estimators.R
started	2018-08-02 19:26:57 GMT
time	00:01:57

Metrics

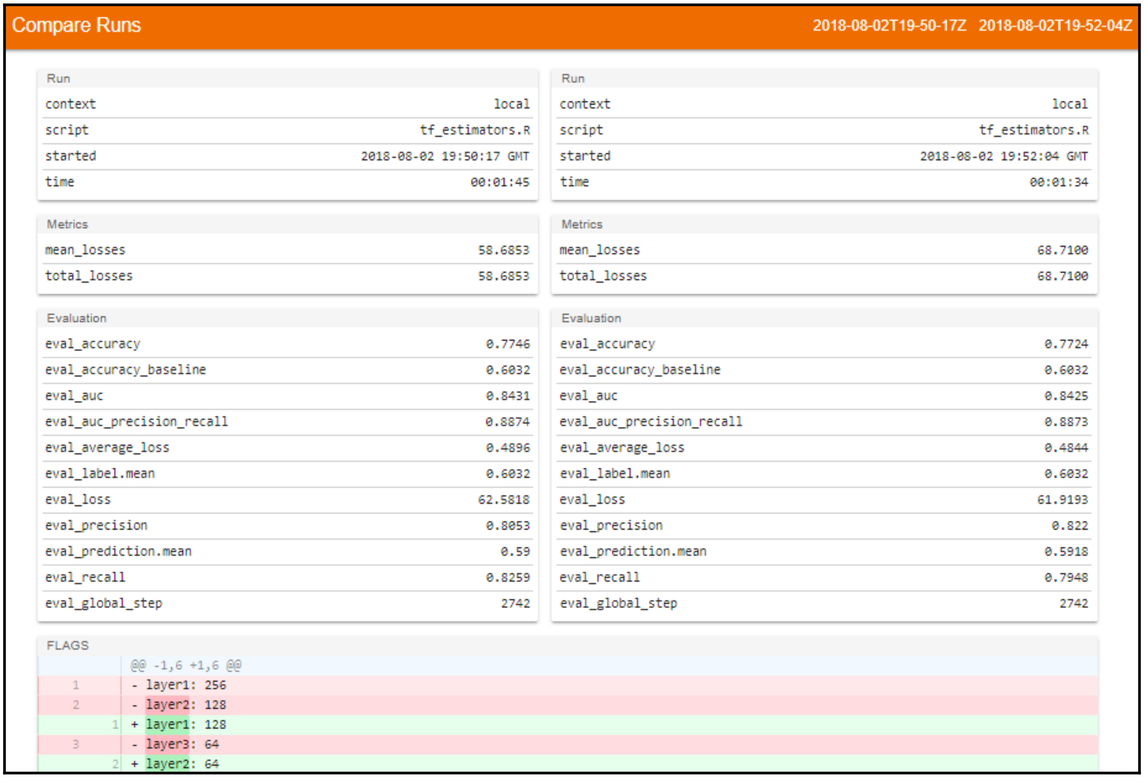
mean_losses	62.4164
total_losses	62.4164

Evaluation

eval_accuracy	0.7751
eval_accuracy_baseline	0.6032
eval_auc	0.8427
eval_auc_precision_recall	0.8873
eval_average_loss	0.4924
eval_label.mean	0.6032
eval_loss	62.9496
eval_precision	0.8076
eval_prediction.mean	0.6027
eval_recall	0.8234
eval_global_step	2742

Flags

layer1	256
layer2	128
layer3	64
layer4	32
dropout	0.2



Run	
context	local
script	tf_estimators.R
started	2018-08-02 19:50:17 GMT
time	00:01:45

Run	
context	local
script	tf_estimators.R
started	2018-08-02 19:52:04 GMT
time	00:01:34

Metrics	
mean_losses	58.6853
total_losses	58.6853

Metrics	
mean_losses	68.7100
total_losses	68.7100

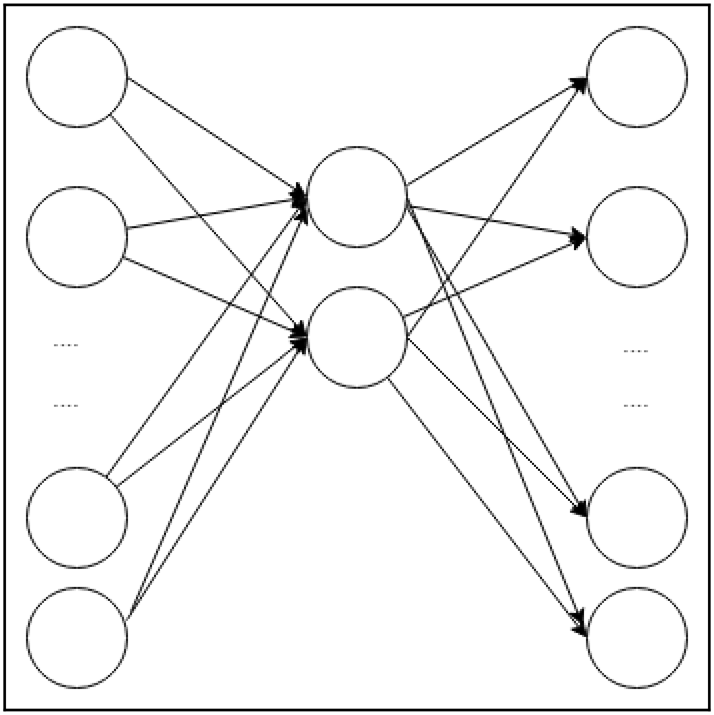
Evaluation	
eval_accuracy	0.7746
eval_accuracy_baseline	0.6032
eval_auc	0.8431
eval_auc_precision_recall	0.8874
eval_average_loss	0.4896
eval_label.mean	0.6032
eval_loss	62.5818
eval_precision	0.8053
eval_prediction.mean	0.59
eval_recall	0.8259
eval_global_step	2742

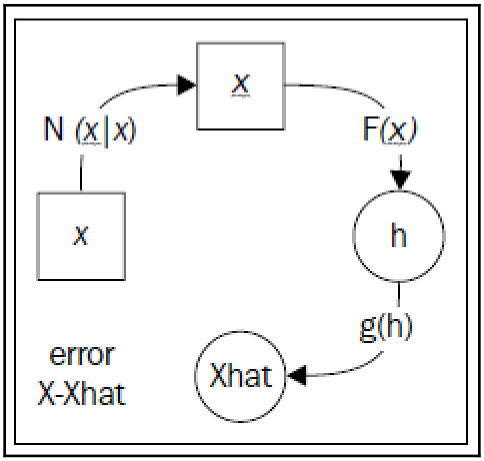
Evaluation	
eval_accuracy	0.7724
eval_accuracy_baseline	0.6032
eval_auc	0.8425
eval_auc_precision_recall	0.8873
eval_average_loss	0.4844
eval_label.mean	0.6032
eval_loss	61.9193
eval_precision	0.822
eval_prediction.mean	0.5918
eval_recall	0.7948
eval_global_step	2742

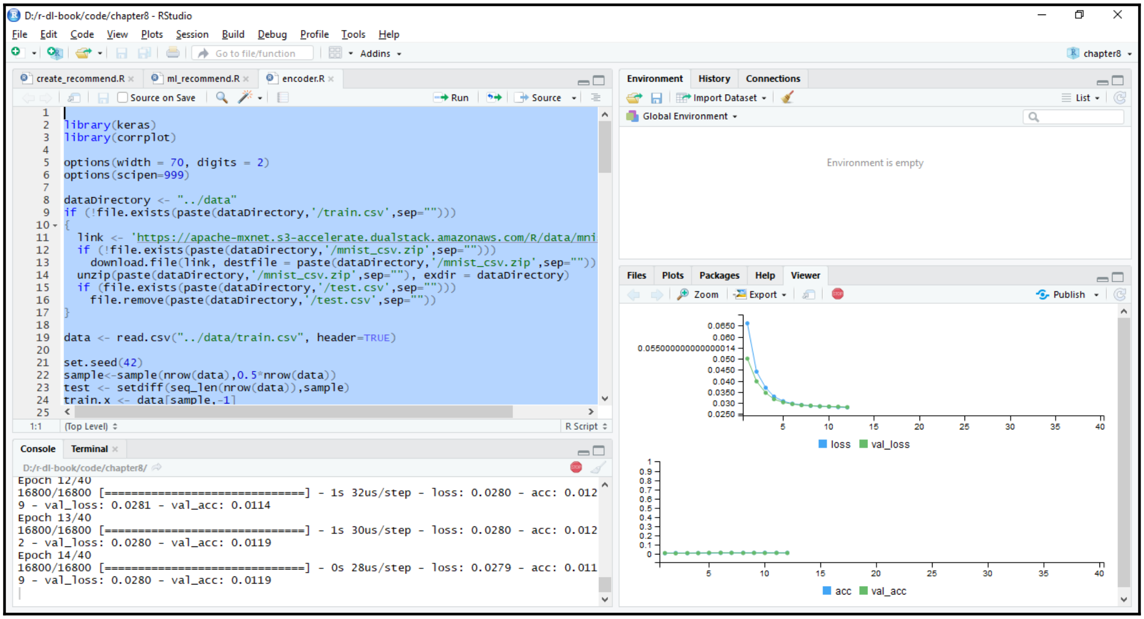
FLAGS

```
        @@ -1,6 +1,6 @@
1       - layer1: 256
2       - layer2: 128
    1   + layer1: 128
3       - layer3: 64
    2   + layer2: 64
```

Chapter 9: Anomaly Detection and Recommendation Systems

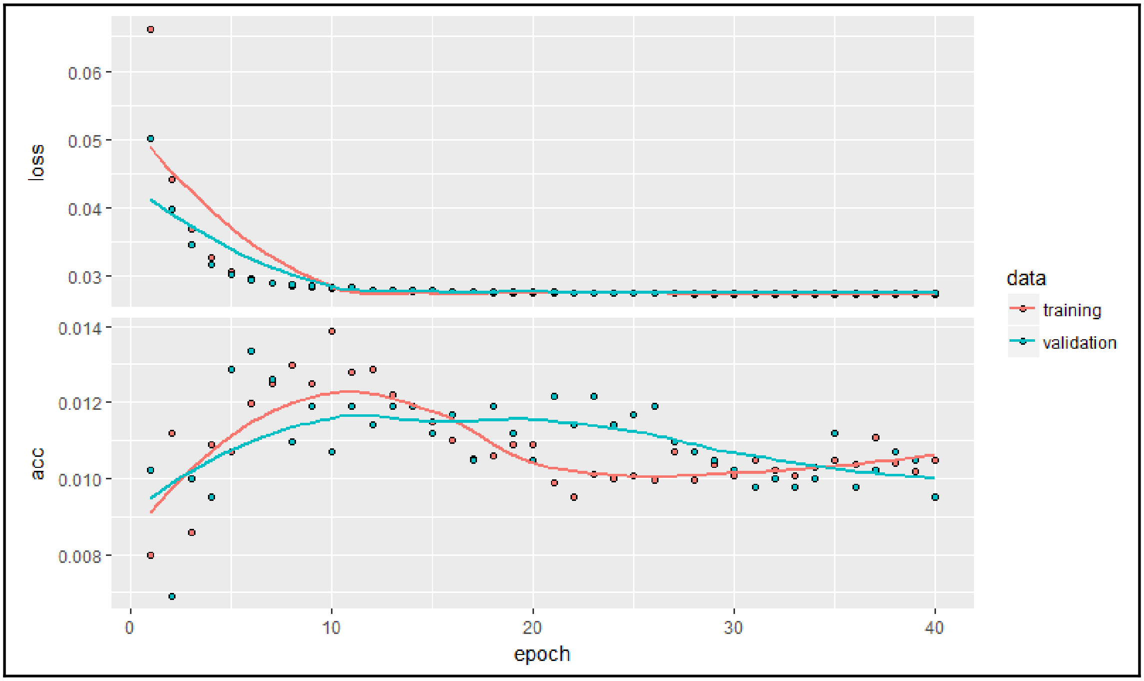


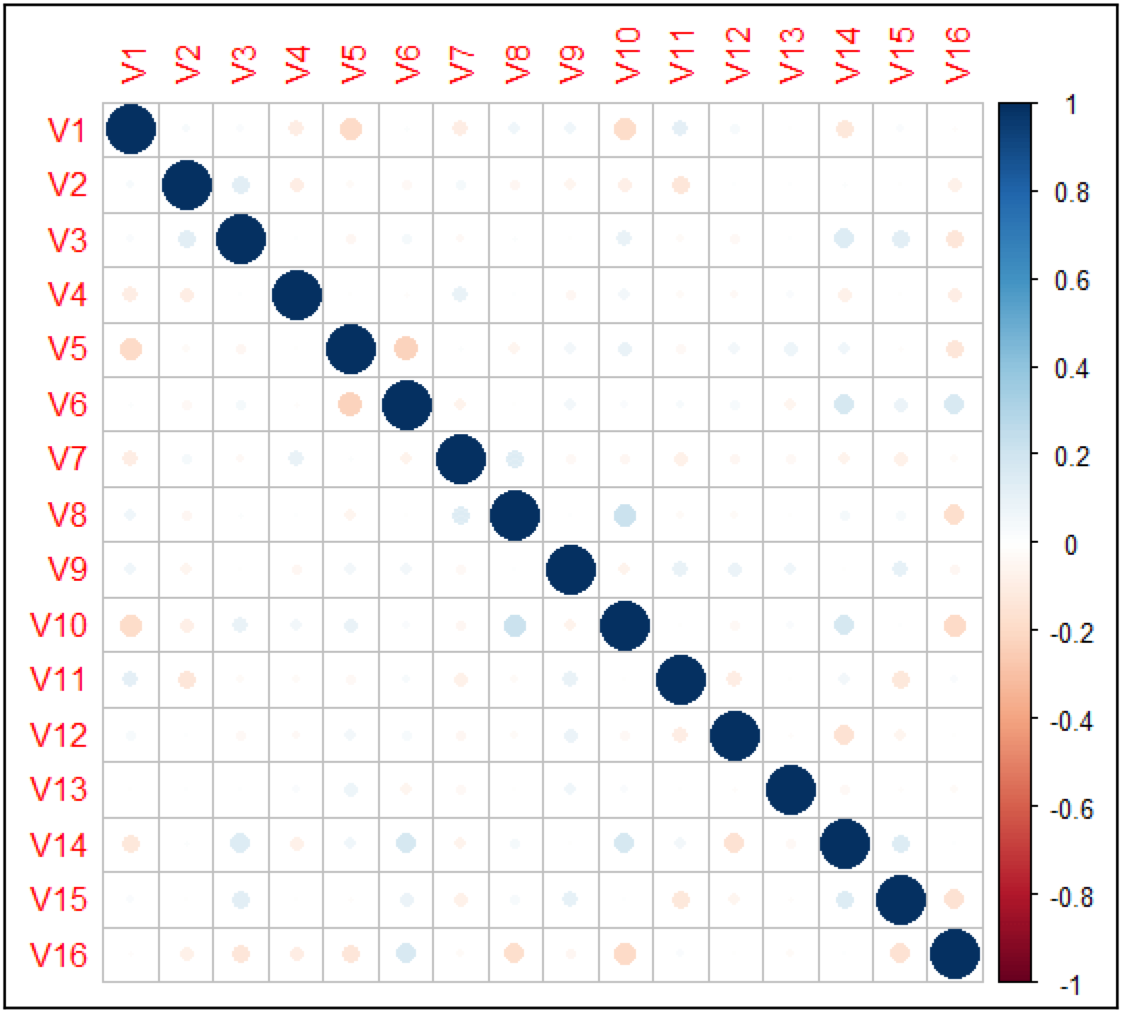


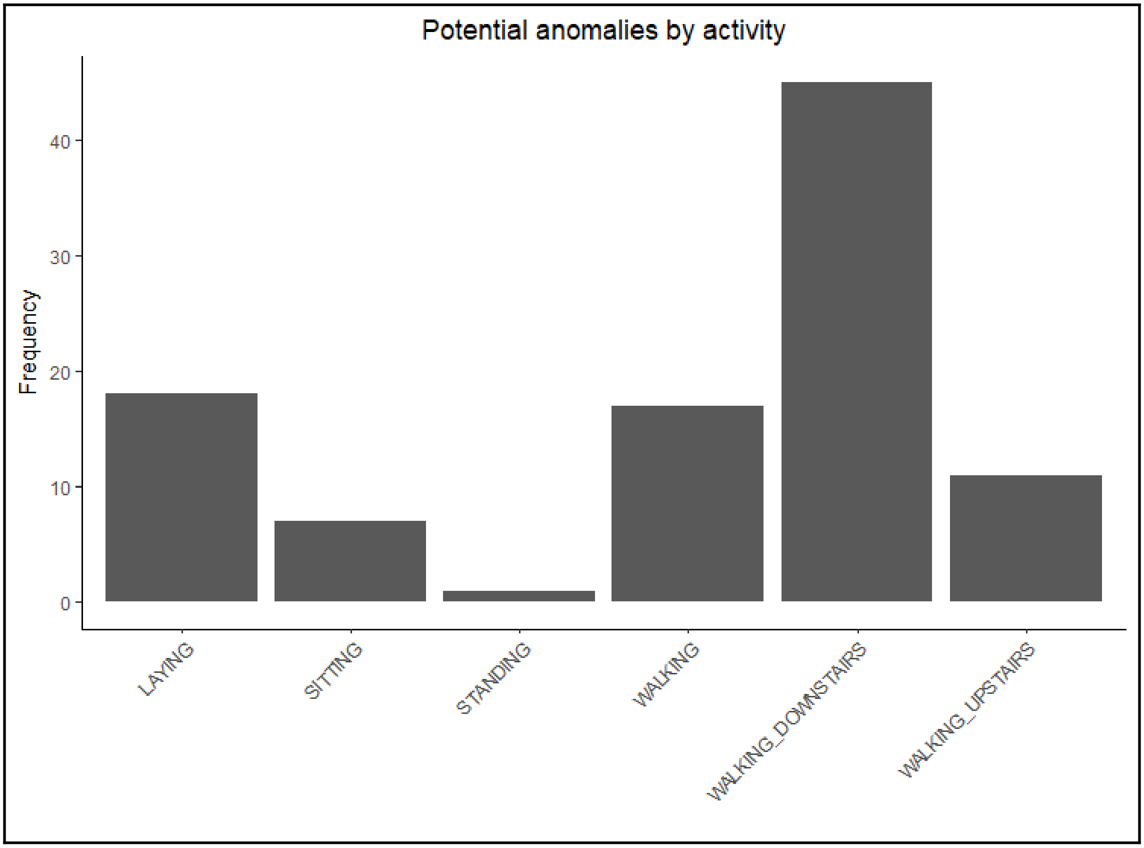


```

library(keras)
library(corrplot)

options(width = 70, digits = 2)
options(scipen=999)

dataDirectory <- "../data"
if (!file.exists(paste(dataDirectory,'/train.csv',sep="")))
{
  link <- 'https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/R/data/mni
  if (!file.exists(paste(dataDirectory,'/mnist_csv.zip',sep="")))
    download.file(link, destfile = paste(dataDirectory,'/mnist_csv.zip',sep=""))
  unzip(paste(dataDirectory,'/mnist_csv.zip',sep=""), exdir = dataDirectory)
  if (file.exists(paste(dataDirectory,'/test.csv',sep="")))
    file.remove(paste(dataDirectory,'/test.csv',sep=""))
}

data <- read.csv("../data/train.csv", header=TRUE)

set.seed(42)
sample<-sample(nrow(data),0.5*nrow(data))
test <- setdiff(seq_len(nrow(data)),sample)
train.x <- data[sample,-1]
```

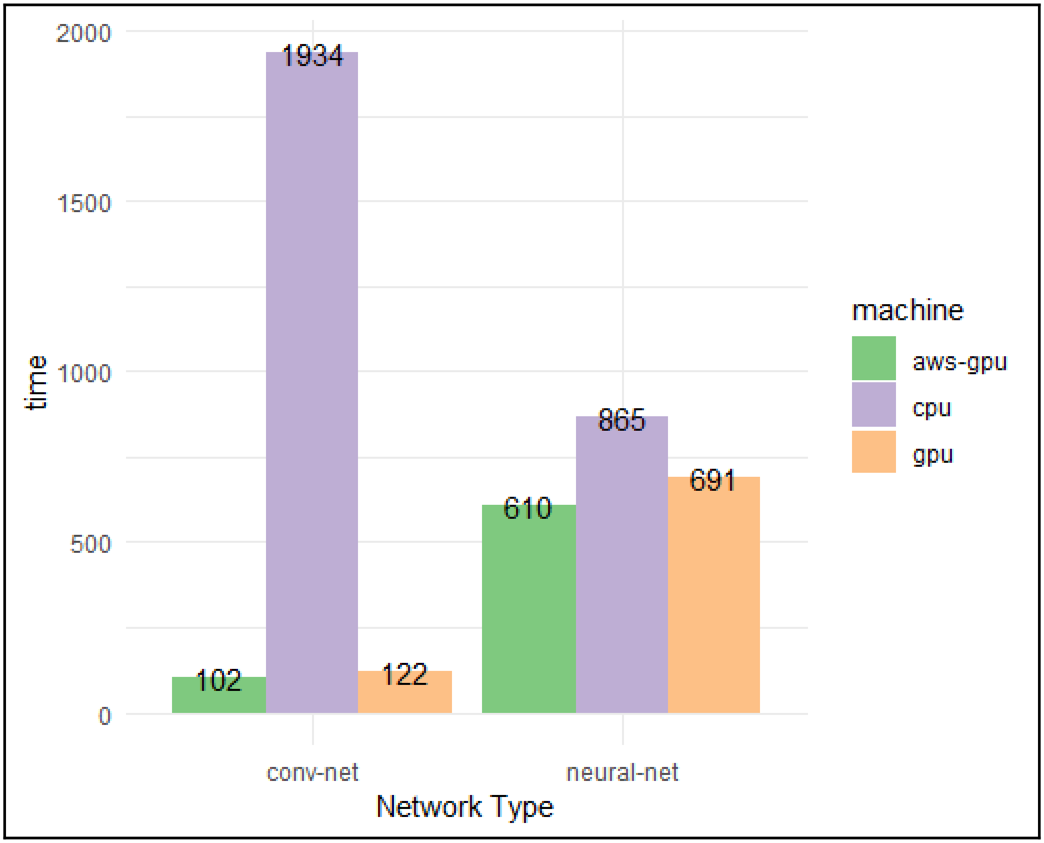
```
Epoch 12/40
16800/16800 [==============================] - 1s 32us/step - loss: 0.0280 - acc: 0.012
9 - val_loss: 0.0281 - val_acc: 0.0114
Epoch 13/40
16800/16800 [==============================] - 1s 30us/step - loss: 0.0280 - acc: 0.012
2 - val_loss: 0.0280 - val_acc: 0.0119
Epoch 14/40
16800/16800 [==============================] - 0s 28us/step - loss: 0.0279 - acc: 0.011
9 - val_loss: 0.0280 - val_acc: 0.0119
```

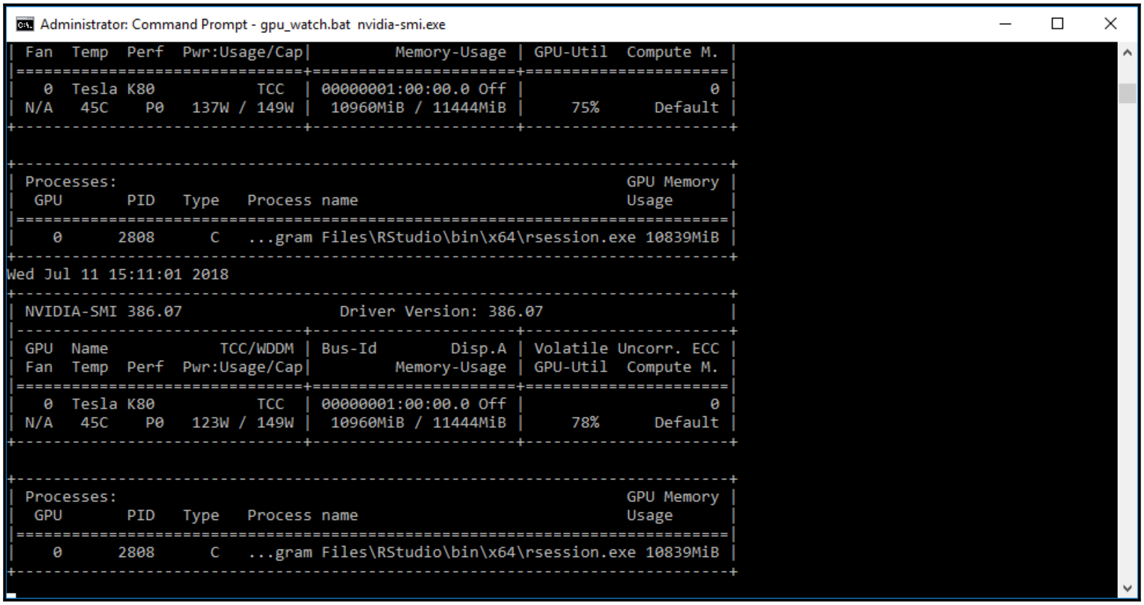



V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


Potential anomalies by activity
Frequency
0
10
20
30
40
LAYING
SITTING
STANDING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

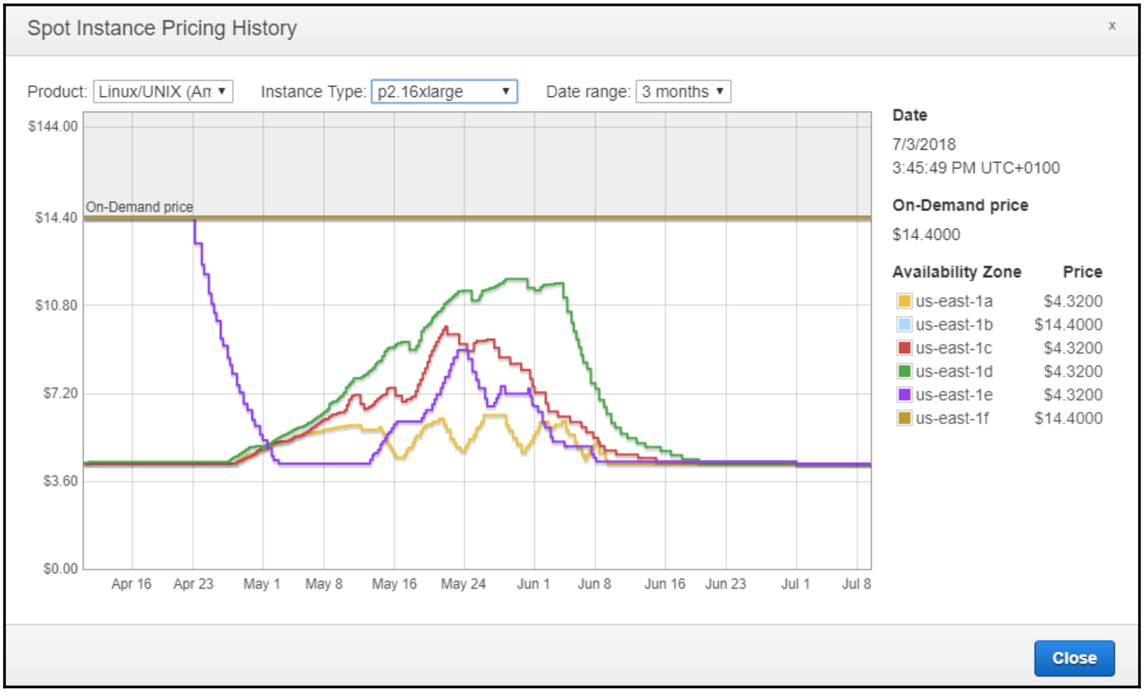
Chapter 10: Running Deep Learning Models in the Cloud





Administrator: Command Prompt - gpu_watch.bat nvidia-smi.exe

```
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           TCC  | 00000001:00:00.0 Off |                    0 |
| N/A   45C    P0   137W / 149W |  10960MiB / 11444MiB |     75%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      2808      C   ...gram Files\RStudio\bin\x64\rsession.exe 10839MiB |
+-----------------------------------------------------------------------------+
Wed Jul 11 15:11:01 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 386.07                 Driver Version: 386.07                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name            TCC/WDDM | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           TCC  | 00000001:00:00.0 Off |                    0 |
| N/A   45C    P0   123W / 149W |  10960MiB / 11444MiB |     78%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      2808      C   ...gram Files\RStudio\bin\x64\rsession.exe 10839MiB |
+-----------------------------------------------------------------------------+
```



Spot Instance Pricing History

Product: Linux/UNIX (An ▾   Instance Type: p2.16xlarge ▾   Date range: 3 months ▾

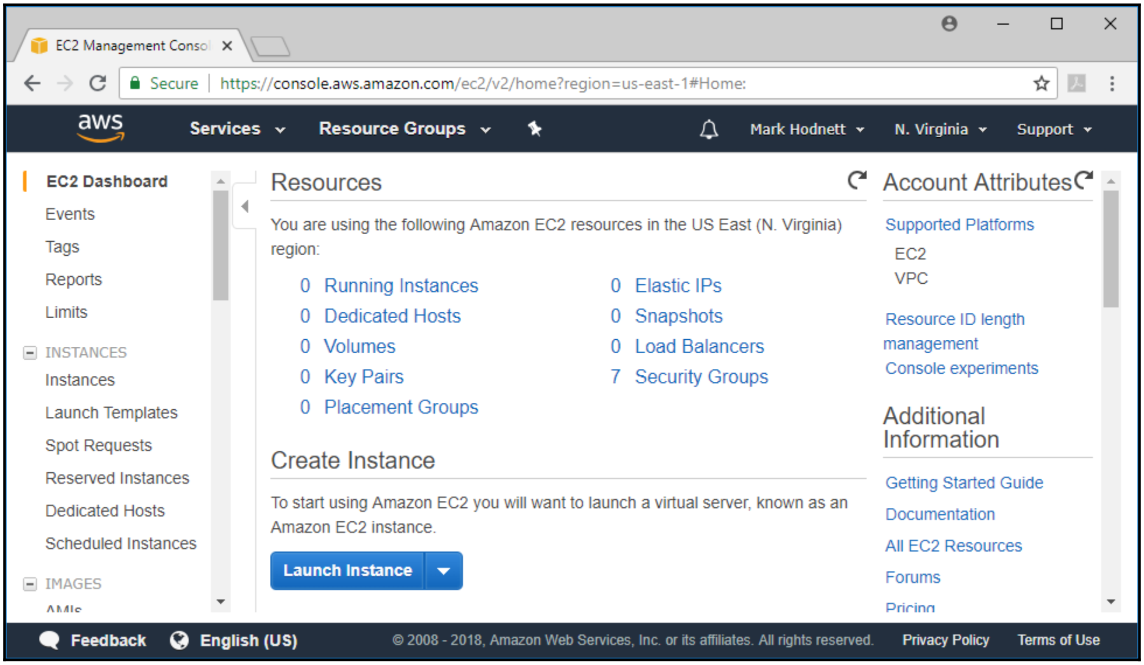
Date

7/3/2018
3:45:49 PM UTC+0100

On-Demand price

$14.4000

Availability Zone	Price
🟨 us-east-1a	$4.3200
🟦 us-east-1b	$14.4000
🟥 us-east-1c	$4.3200
🟩 us-east-1d	$4.3200
🟪 us-east-1e	$4.3200
🟫 us-east-1f	$14.4000



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0	Running Instances	0	Elastic IPs
0	Dedicated Hosts	0	Snapshots
0	Volumes	0	Load Balancers
0	Key Pairs	7	Security Groups
0	Placement Groups		

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Account Attributes

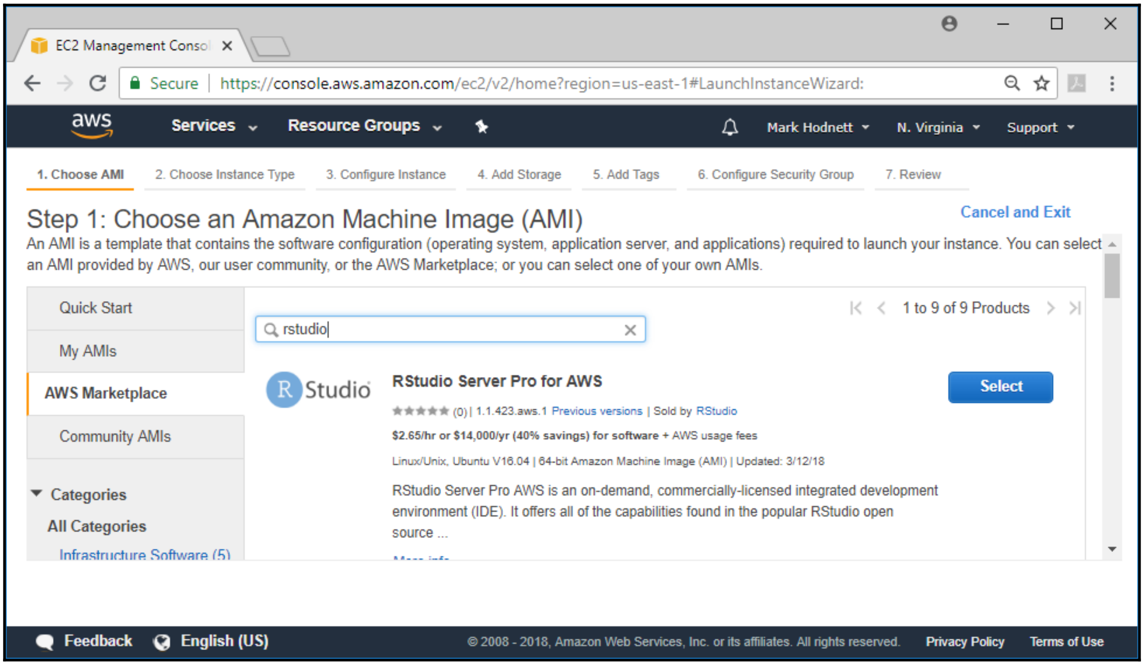
Supported Platforms
EC2
VPC

Resource ID length management
Console experiments

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

rstudio

RStudio Server Pro for AWS

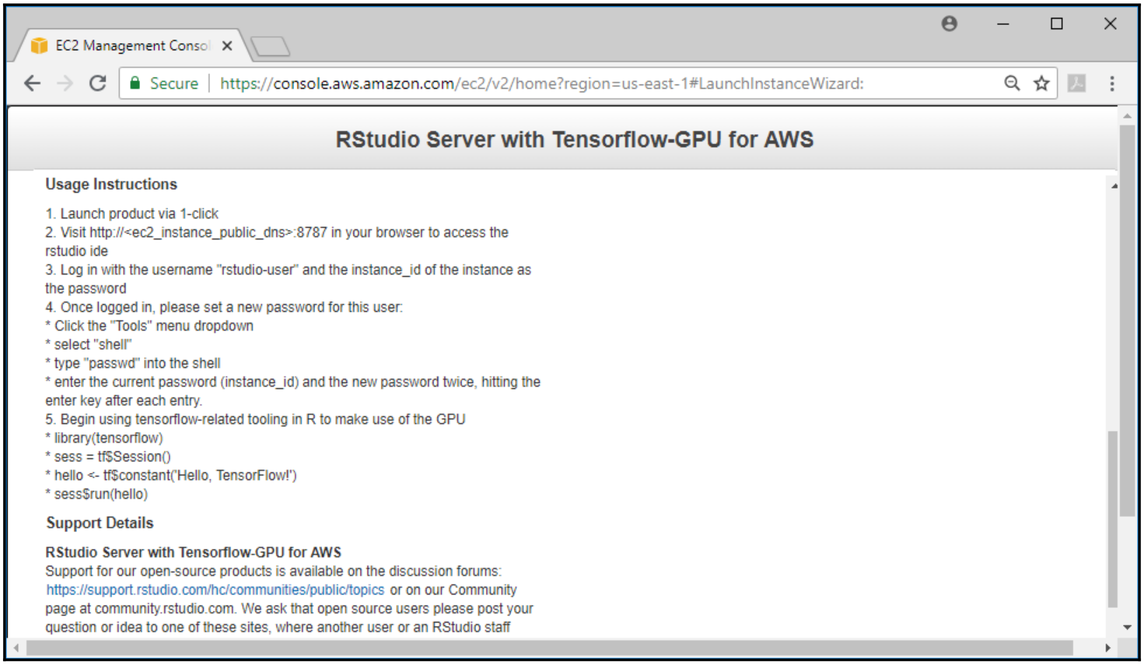
★★★★★ (0) | 1.1.423.aws.1 Previous versions | Sold by RStudio

$2.65/hr or $14,000/yr (40% savings) for software + AWS usage fees

Linux/Unix, Ubuntu V16.04 | 64-bit Amazon Machine Image (AMI) | Updated: 3/12/18

RStudio Server Pro AWS is an on-demand, commercially-licensed integrated development environment (IDE). It offers all of the capabilities found in the popular RStudio open source ...

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.


EC2 Management Consol
Secure
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

RStudio Server with Tensorflow-GPU for AWS

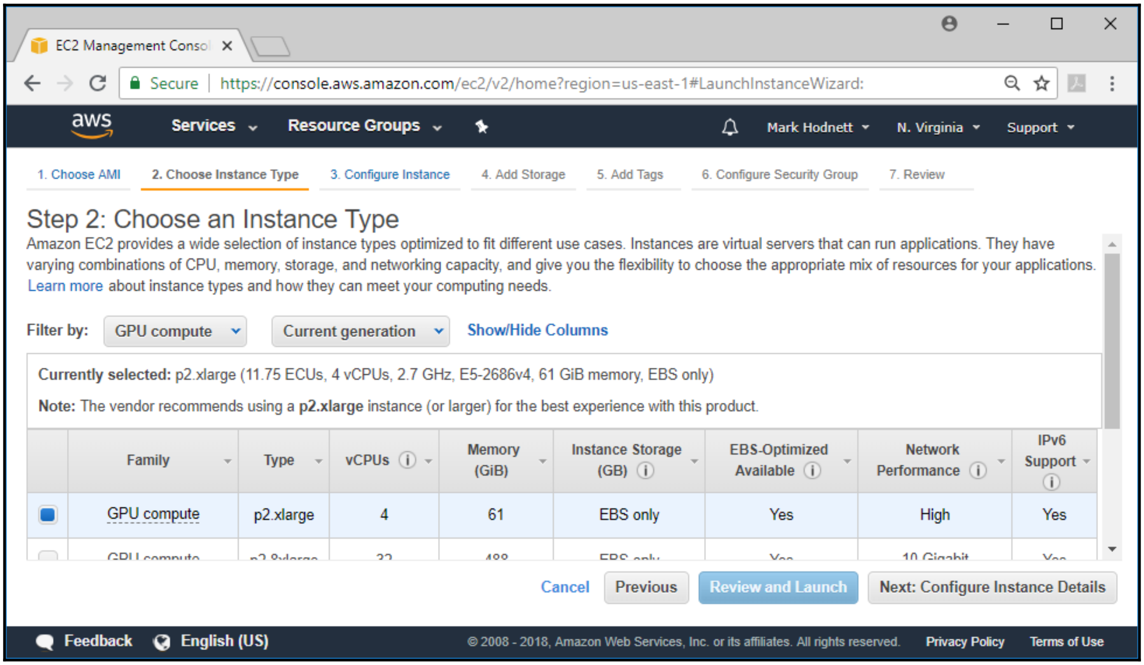
Usage Instructions

1. Launch product via 1-click
2. Visit http://<ec2_instance_public_dns>:8787 in your browser to access the rstudio ide
3. Log in with the username "rstudio-user" and the instance_id of the instance as the password
4. Once logged in, please set a new password for this user:
* Click the "Tools" menu dropdown
* select "shell"
* type "passwd" into the shell
* enter the current password (instance_id) and the new password twice, hitting the enter key after each entry.
5. Begin using tensorflow-related tooling in R to make use of the GPU
* library(tensorflow)
* sess = tf$Session()
* hello <- tf$constant('Hello, TensorFlow!')
* sess$run(hello)

Support Details

RStudio Server with Tensorflow-GPU for AWS

Support for our open-source products is available on the discussion forums: https://support.rstudio.com/hc/communities/public/topics or on our Community page at community.rstudio.com. We ask that open source users please post your question or idea to one of these sites, where another user or an RStudio staff



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

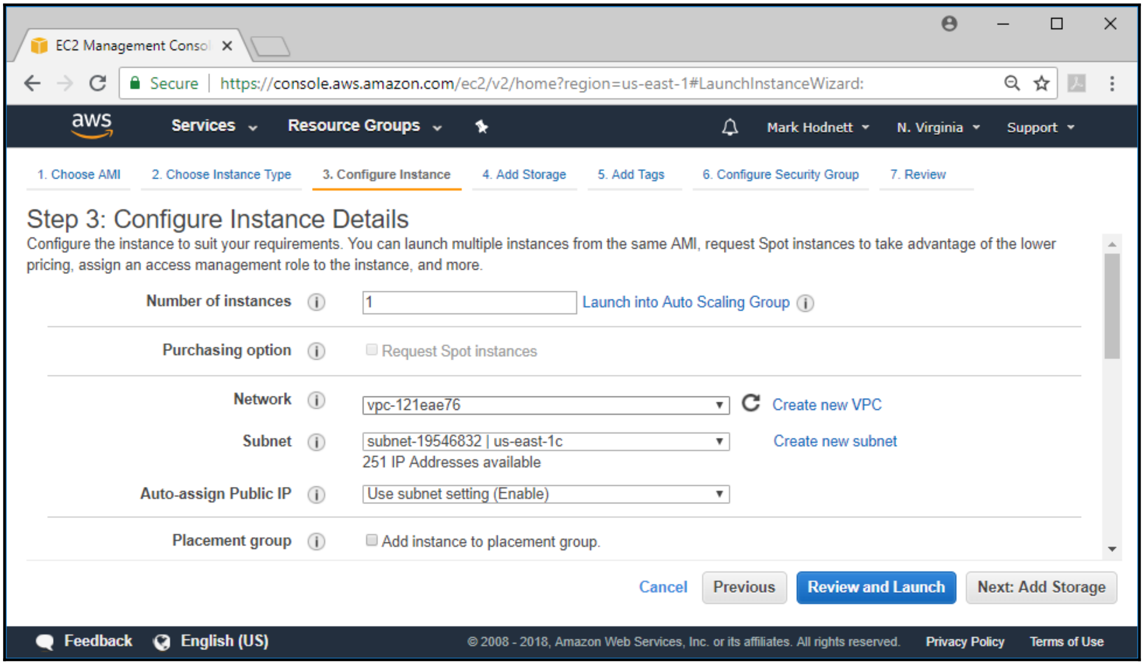
Filter by: GPU compute | Current generation | Show/Hide Columns

Currently selected: p2.xlarge (11.75 ECUs, 4 vCPUs, 2.7 GHz, E5-2686v4, 61 GiB memory, EBS only)

Note: The vendor recommends using a **p2.xlarge** instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☑	GPU compute	p2.xlarge	4	61	EBS only	Yes	High	Yes
□	GPU compute	p2.8xlarge	32	488	EBS only	Yes	10 Gigabit	Yes

Cancel
Previous
Review and Launch
Next: Configure Instance Details
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

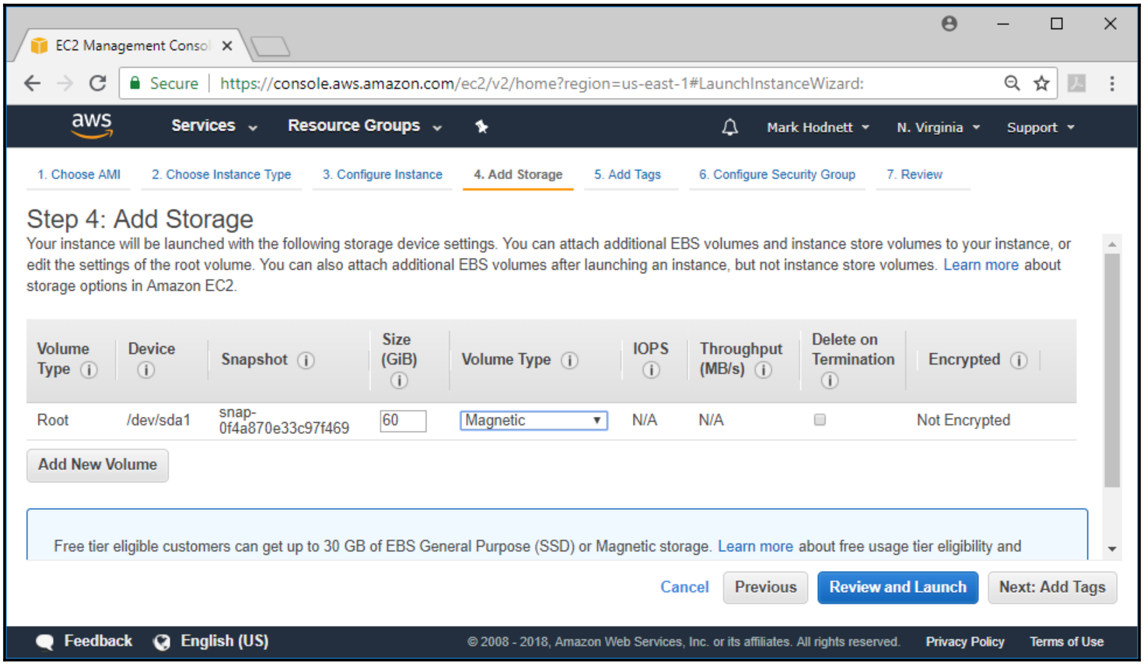
Purchasing option: □ Request Spot instances

Network: vpc-121eae76 — Create new VPC

Subnet: subnet-19546832 | us-east-1c — Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: □ Add instance to placement group.

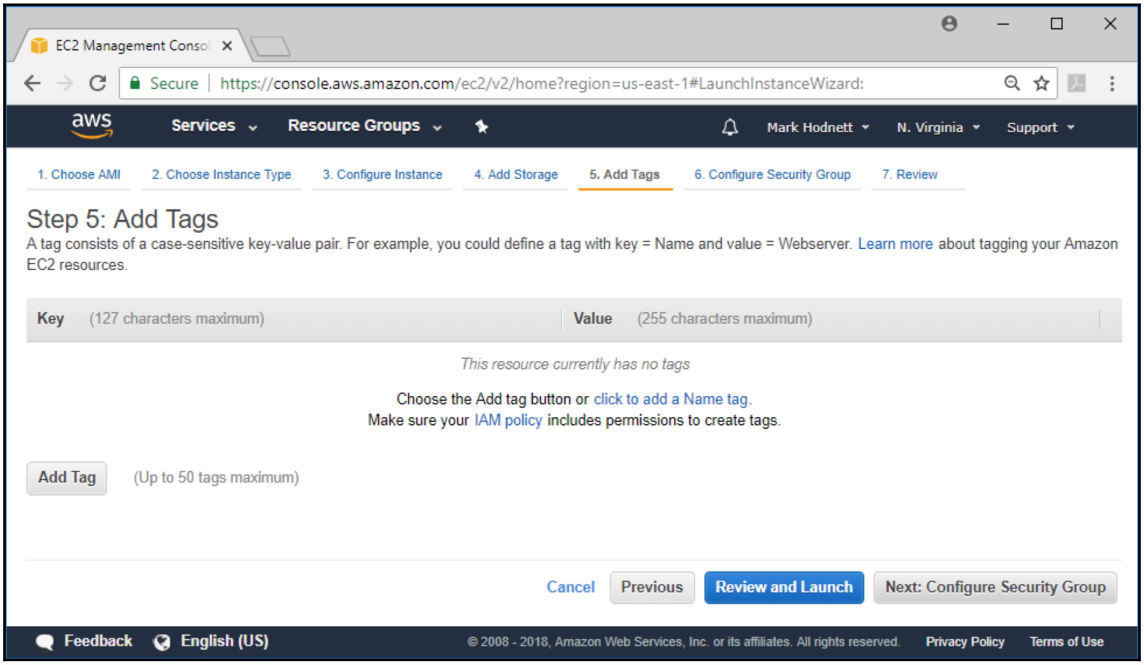


Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0f4a870e33c97f469	60	Magnetic	N/A	N/A	□	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and



Step 5: Add Tags

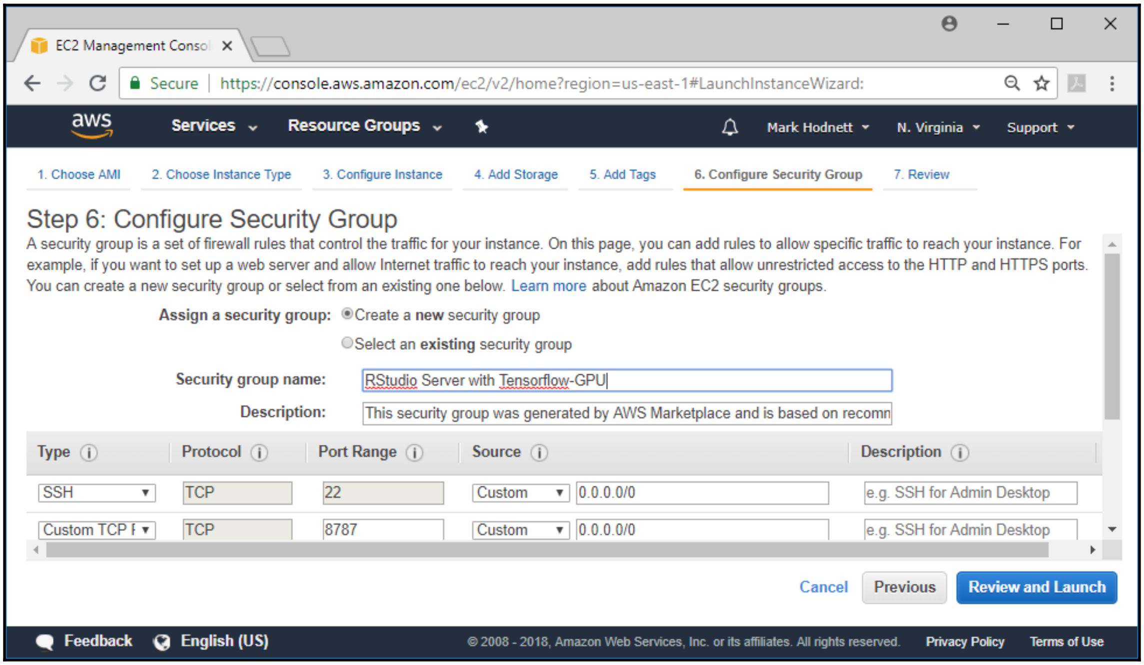
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

(Up to 50 tags maximum)



Step 6: Configure Security Group

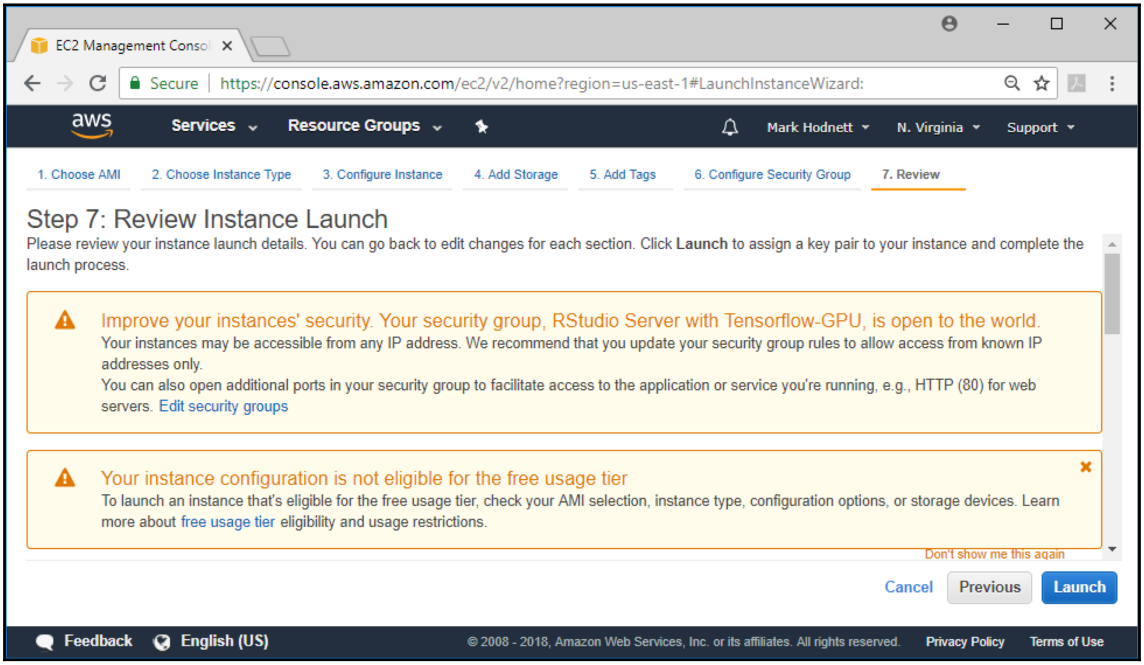
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: RStudio Server with Tensorflow-GPU

Description: This security group was generated by AWS Marketplace and is based on recomn

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8787	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

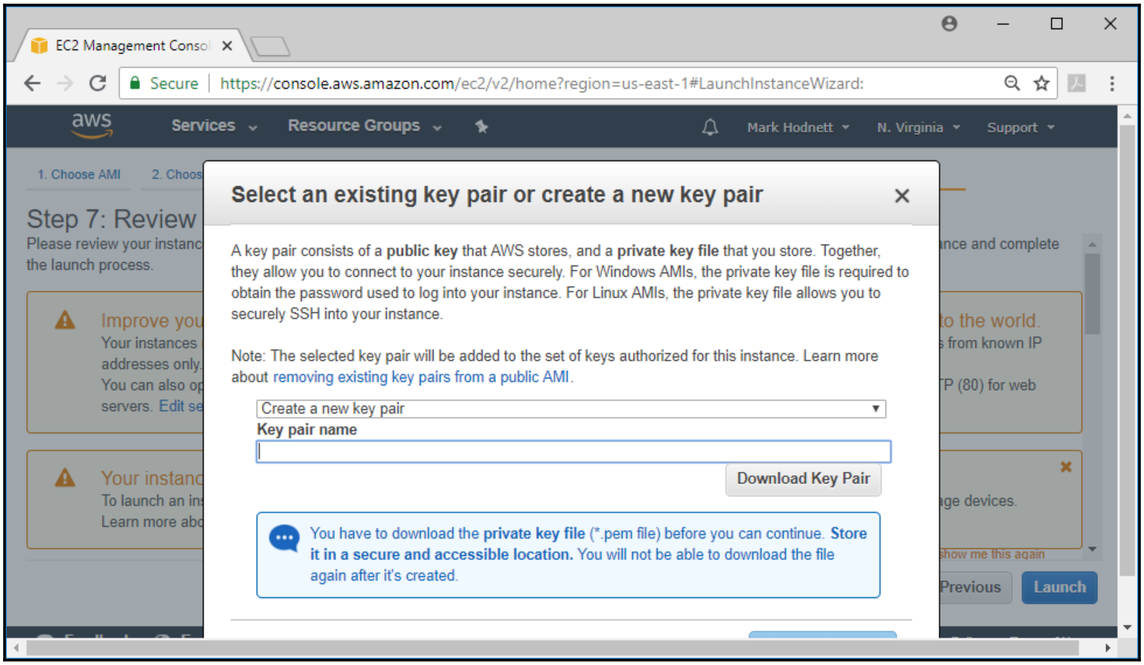


Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, RStudio Server with Tensorflow-GPU, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups

Your instance configuration is not eligible for the free usage tier
To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about free usage tier eligibility and usage restrictions.



Select an existing key pair or create a new key pair

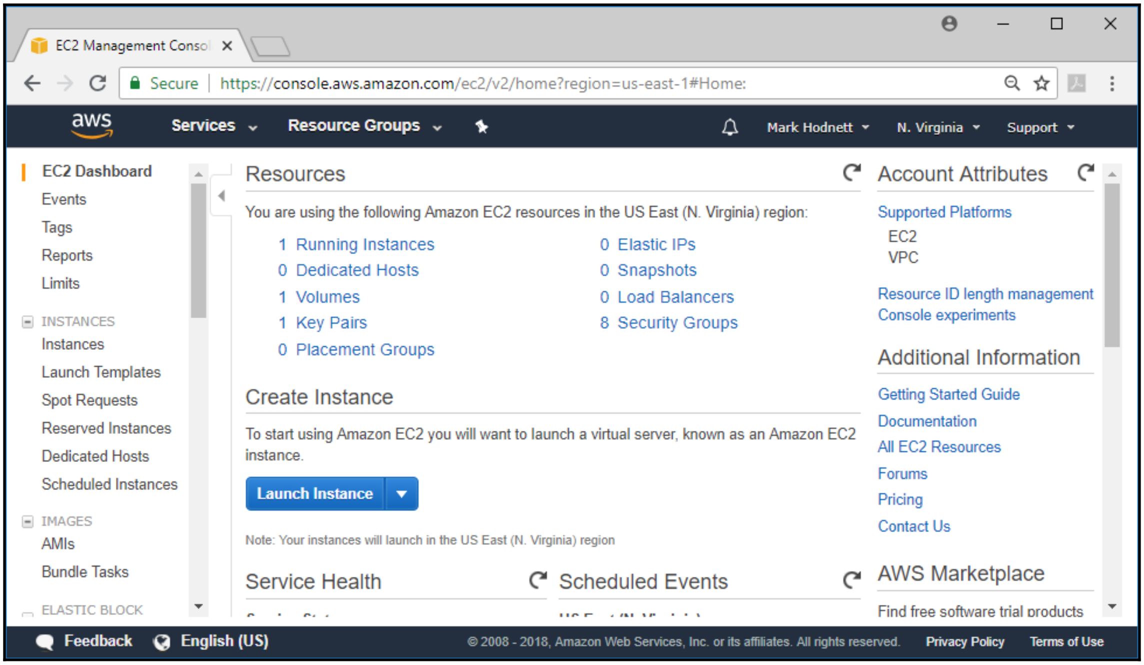
A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.



You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

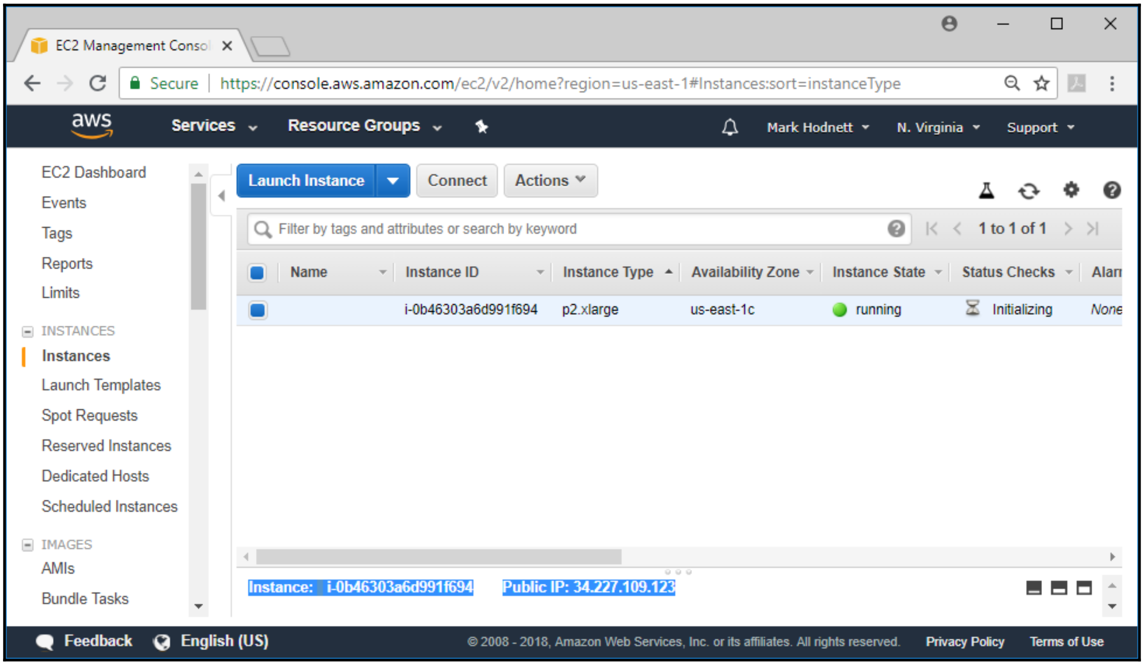
1 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
1 Volumes	0 Load Balancers
1 Key Pairs	8 Security Groups
0 Placement Groups	

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Find free software trial products

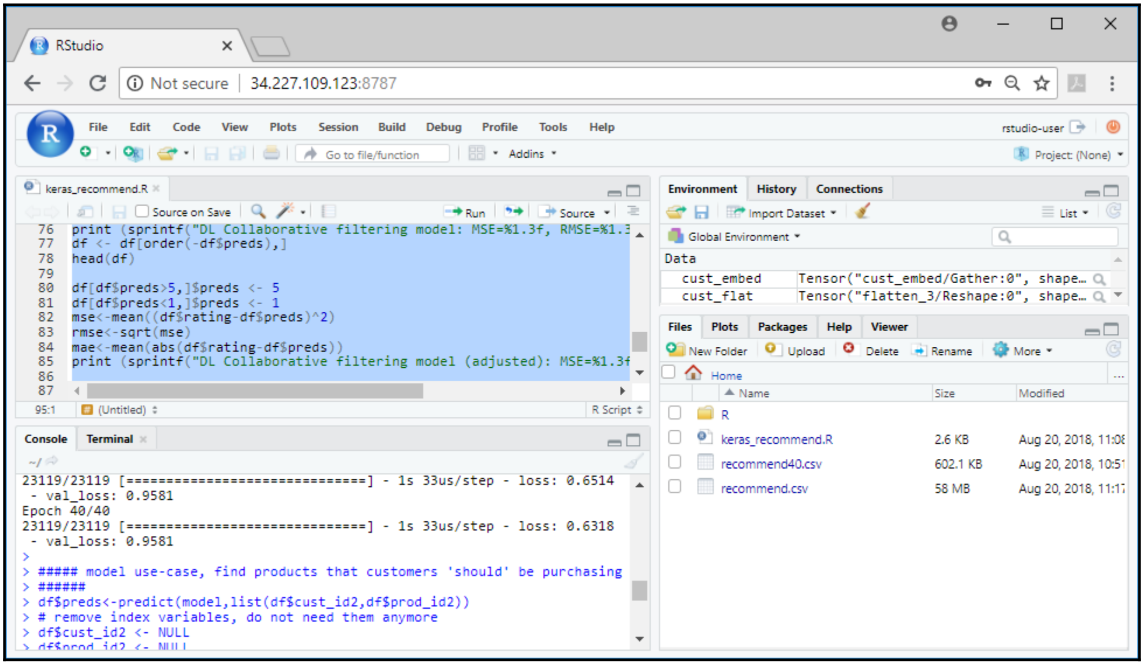
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alar
	i-0b46303a6d991f694	p2.xlarge	us-east-1c	running	Initializing	None

Instance: i-0b46303a6d991f694 Public IP: 34.227.109.123

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

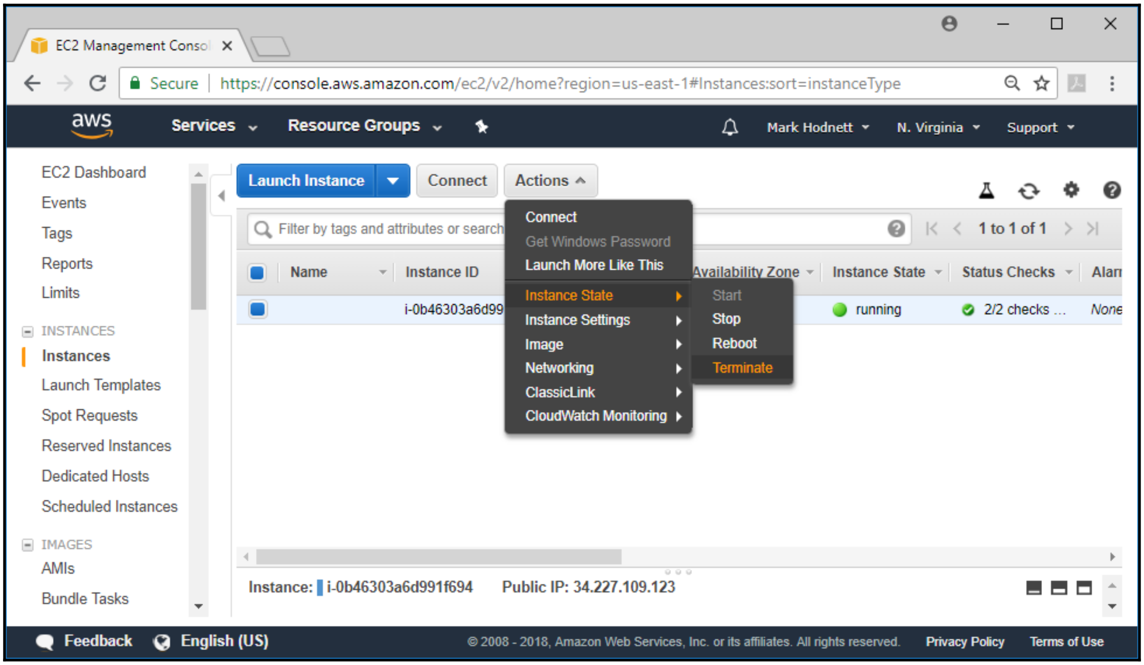

RStudio
Not secure | 34.227.109.123:8787
File Edit Code View Plots Session Build Debug Profile Tools Help
rstudio-user
Go to file/function
Addins
Project: (None)
keras_recommend.R
Source on Save
Run
Source
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Home
95:1
(Untitled)
R Script
Console Terminal
~/

```
print (sprintf("DL Collaborative filtering model: MSE=%1.3f, RMSE=%1.3
df <- df[order(-df$preds),]
head(df)

df[df$preds>5,]$preds <- 5
df[df$preds<1,]$preds <- 1
mse<-mean((df$rating-df$preds)^2)
rmse<-sqrt(mse)
mae<-mean(abs(df$rating-df$preds))
print (sprintf("DL Collaborative filtering model (adjusted): MSE=%1.3f
```

```
23119/23119 [==============================] - 1s 33us/step - loss: 0.6514
 - val_loss: 0.9581
Epoch 40/40
23119/23119 [==============================] - 1s 33us/step - loss: 0.6318
 - val_loss: 0.9581
>
> ##### model use-case, find products that customers 'should' be purchasing
> ######
> df$preds<-predict(model,list(df$cust_id2,df$prod_id2))
> # remove index variables, do not need them anymore
> df$cust_id2 <- NULL
> df$prod_id2 <- NULL
```

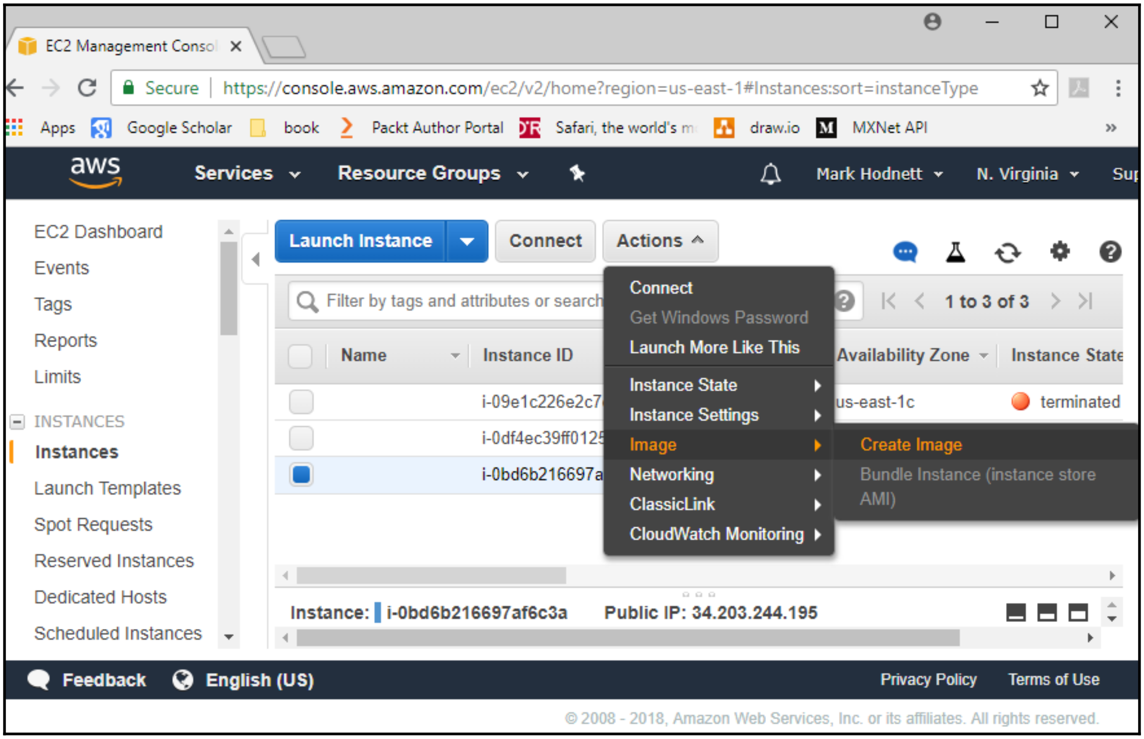
```
Data
cust_embed   Tensor("cust_embed/Gather:0", shape…
cust_flat    Tensor("flatten_3/Reshape:0", shape…
```

Name	Size	Modified
R		
keras_recommend.R	2.6 KB	Aug 20, 2018, 11:0
recommend40.csv	602.1 KB	Aug 20, 2018, 10:5
recommend.csv	58 MB	Aug 20, 2018, 11:1



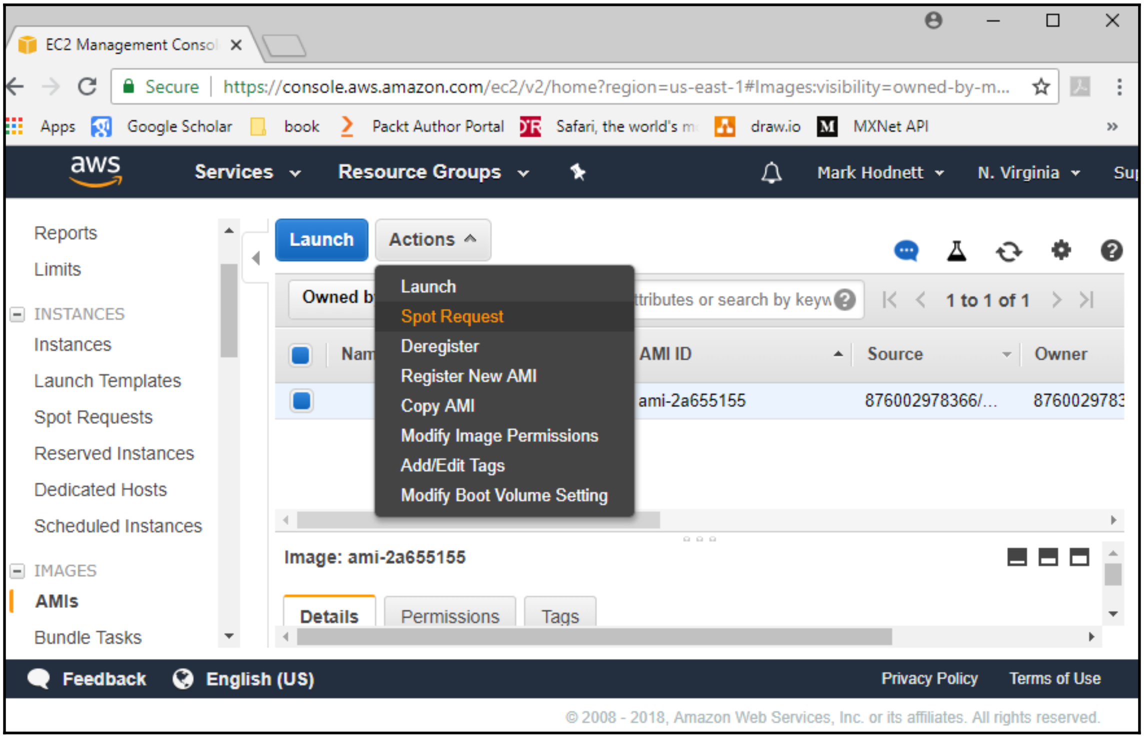
Name	Instance ID	Availability Zone	Instance State	Status Checks	Alarm
	i-0b46303a6d99		running	2/2 checks ...	None

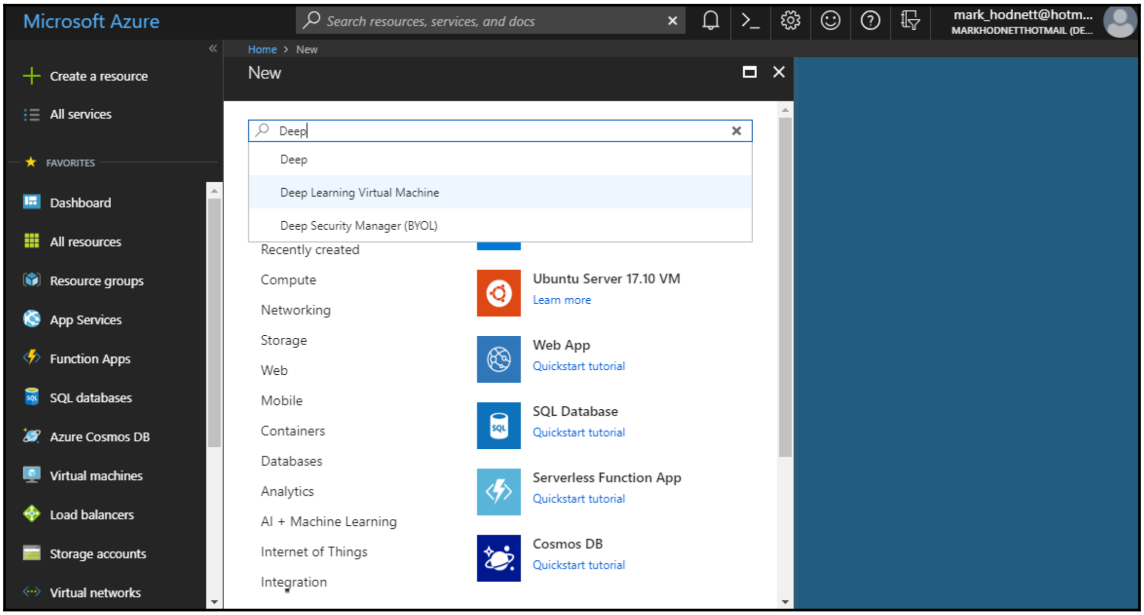
Instance: i-0b46303a6d991f694 Public IP: 34.227.109.123

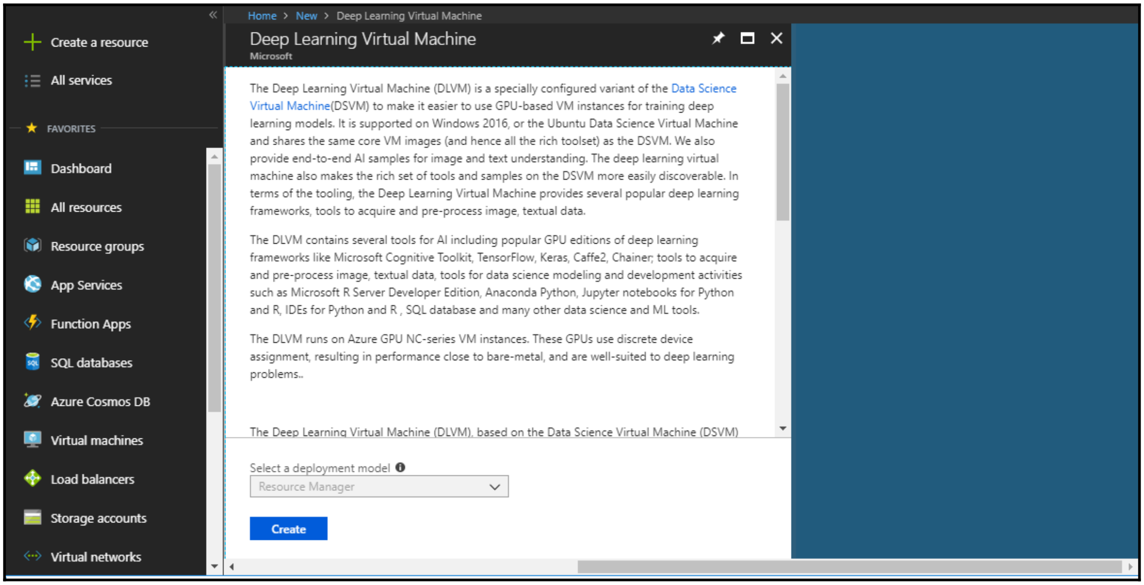


Instance ID	Availability Zone	Instance State
i-09e1c226e2c7	us-east-1c	terminated
i-0df4ec39ff0125		
i-0bd6b216697a		

Instance: i-0bd6b216697af6c3a Public IP: 34.203.244.195


EC2 Management Consol
Secure
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Images:visibility=owned-by-m...
Apps
Google Scholar
book
Packt Author Portal
Safari, the world's m
draw.io
MXNet API
aws
Services
Resource Groups
Mark Hodnett
N. Virginia
Su
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
IMAGES
AMIs
Bundle Tasks
Launch
Actions
Launch
Spot Request
Deregister
Register New AMI
Copy AMI
Modify Image Permissions
Add/Edit Tags
Modify Boot Volume Setting
Owned b
ttributes or search by keyw
1 to 1 of 1
Nam
AMI ID
Source
Owner
ami-2a655155
876002978366/...
876002978
Image: ami-2a655155
Details
Permissions
Tags
Feedback
English (US)
Privacy Policy
Terms of Use
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Microsoft Azure
Search resources, services, and docs
mark_hodnett@hotm...
MARKHODNETTHOTMAIL (DE...
Home > New
New
Create a resource
All services
FAVORITES
Dashboard
All resources
Resource groups
App Services
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Deep
Deep
Deep Learning Virtual Machine
Deep Security Manager (BYOL)
Recently created
Compute
Networking
Storage
Web
Mobile
Containers
Databases
Analytics
AI + Machine Learning
Internet of Things
Integration
Ubuntu Server 17.10 VM
Learn more
Web App
Quickstart tutorial
SQL Database
Quickstart tutorial
Serverless Function App
Quickstart tutorial
Cosmos DB
Quickstart tutorial



Deep Learning Virtual Machine
Microsoft

The Deep Learning Virtual Machine (DLVM) is a specially configured variant of the Data Science Virtual Machine(DSVM) to make it easier to use GPU-based VM instances for training deep learning models. It is supported on Windows 2016, or the Ubuntu Data Science Virtual Machine and shares the same core VM images (and hence all the rich toolset) as the DSVM. We also provide end-to-end AI samples for image and text understanding. The deep learning virtual machine also makes the rich set of tools and samples on the DSVM more easily discoverable. In terms of the tooling, the Deep Learning Virtual Machine provides several popular deep learning frameworks, tools to acquire and pre-process image, textual data.

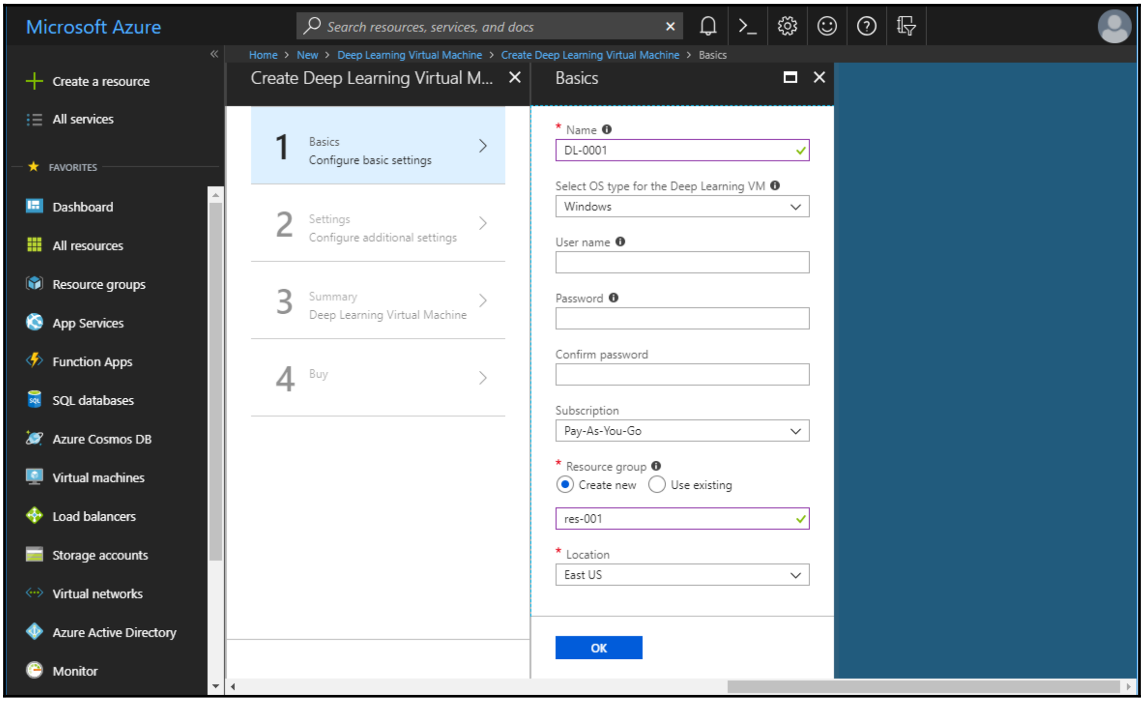
The DLVM contains several tools for AI including popular GPU editions of deep learning frameworks like Microsoft Cognitive Toolkit, TensorFlow, Keras, Caffe2, Chainer; tools to acquire and pre-process image, textual data, tools for data science modeling and development activities such as Microsoft R Server Developer Edition, Anaconda Python, Jupyter notebooks for Python and R, IDEs for Python and R , SQL database and many other data science and ML tools.

The DLVM runs on Azure GPU NC-series VM instances. These GPUs use discrete device assignment, resulting in performance close to bare-metal, and are well-suited to deep learning problems..

The Deep Learning Virtual Machine (DLVM), based on the Data Science Virtual Machine (DSVM)

Select a deployment model: Resource Manager

Create



Create Deep Learning Virtual M...

1 Basics
Configure basic settings

2 Settings
Configure additional settings

3 Summary
Deep Learning Virtual Machine

4 Buy

Basics

* Name: DL-0001

Select OS type for the Deep Learning VM: Windows

User name:

Password:

Confirm password:

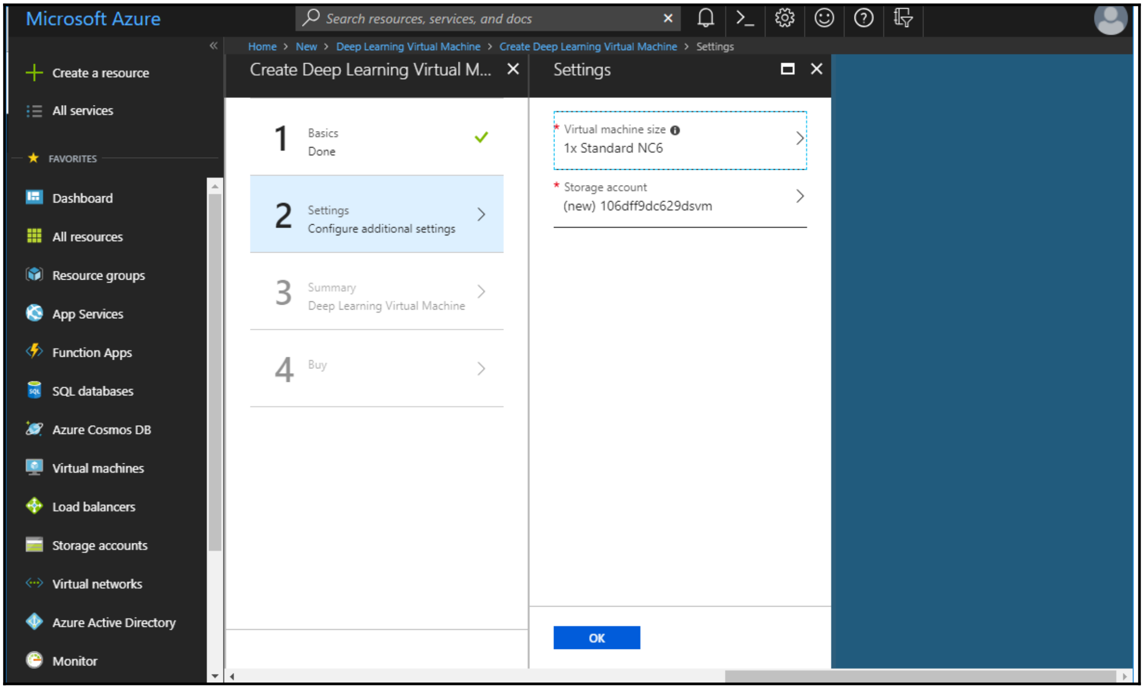
Subscription: Pay-As-You-Go

* Resource group: ● Create new ○ Use existing

res-001

* Location: East US

OK



1 Basics
Done ✓

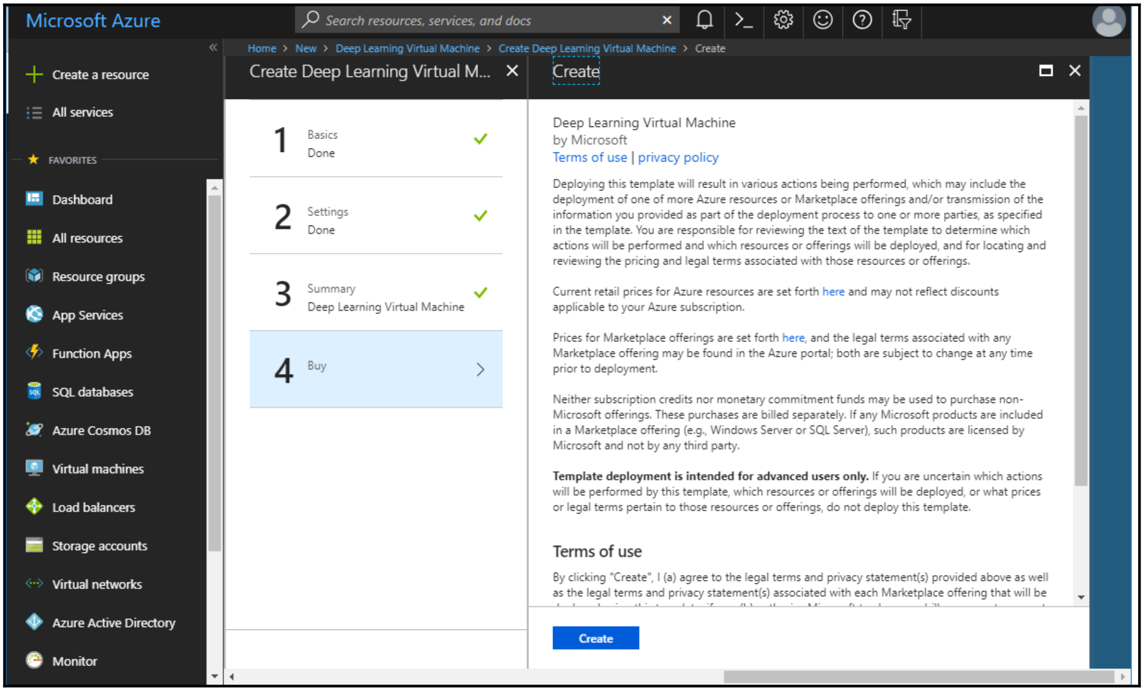
2 Settings
Configure additional settings

3 Summary
Deep Learning Virtual Machine

4 Buy

* Virtual machine size
1x Standard NC6

* Storage account
(new) 106dff9dc629dsvm



1 Basics Done ✓

2 Settings Done ✓

3 Summary Deep Learning Virtual Machine ✓

4 Buy

Deep Learning Virtual Machine
by Microsoft
Terms of use | privacy policy

Deploying this template will result in various actions being performed, which may include the deployment of one of more Azure resources or Marketplace offerings and/or transmission of the information you provided as part of the deployment process to one or more parties, as specified in the template. You are responsible for reviewing the text of the template to determine which actions will be performed and which resources or offerings will be deployed, and for locating and reviewing the pricing and legal terms associated with those resources or offerings.

Current retail prices for Azure resources are set forth here and may not reflect discounts applicable to your Azure subscription.

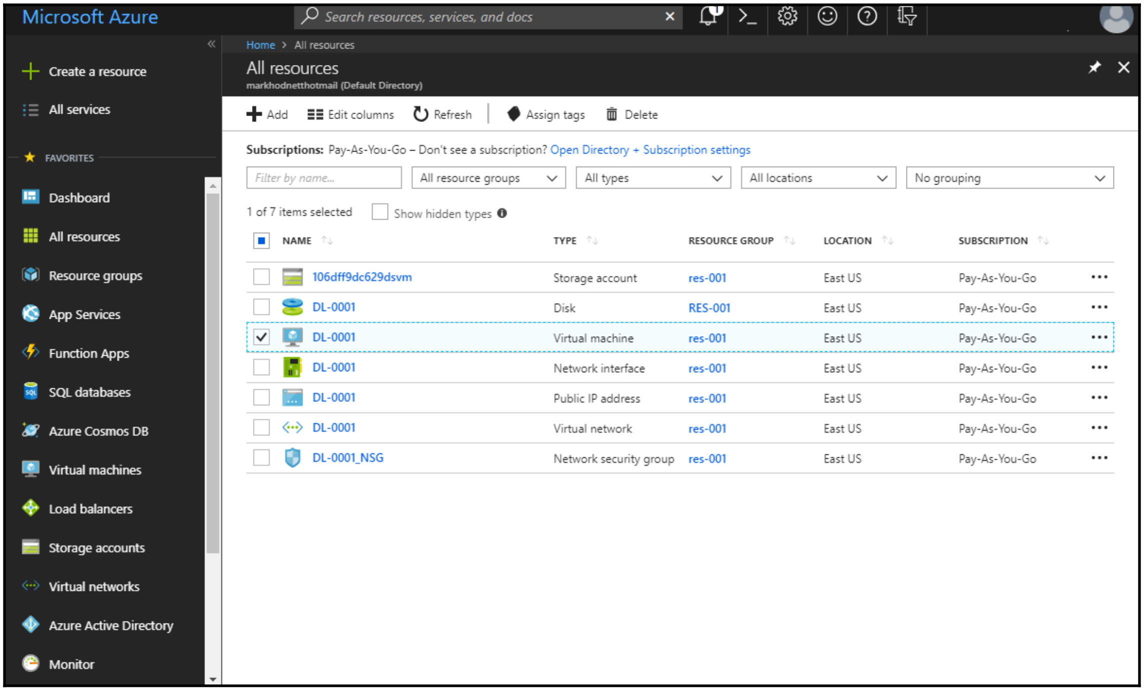
Prices for Marketplace offerings are set forth here, and the legal terms associated with any Marketplace offering may be found in the Azure portal; both are subject to change at any time prior to deployment.

Neither subscription credits nor monetary commitment funds may be used to purchase non-Microsoft offerings. These purchases are billed separately. If any Microsoft products are included in a Marketplace offering (e.g., Windows Server or SQL Server), such products are licensed by Microsoft and not by any third party.

Template deployment is intended for advanced users only. If you are uncertain which actions will be performed by this template, which resources or offerings will be deployed, or what prices or legal terms pertain to those resources or offerings, do not deploy this template.

Terms of use

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) provided above as well as the legal terms and privacy statement(s) associated with each Marketplace offering that will be

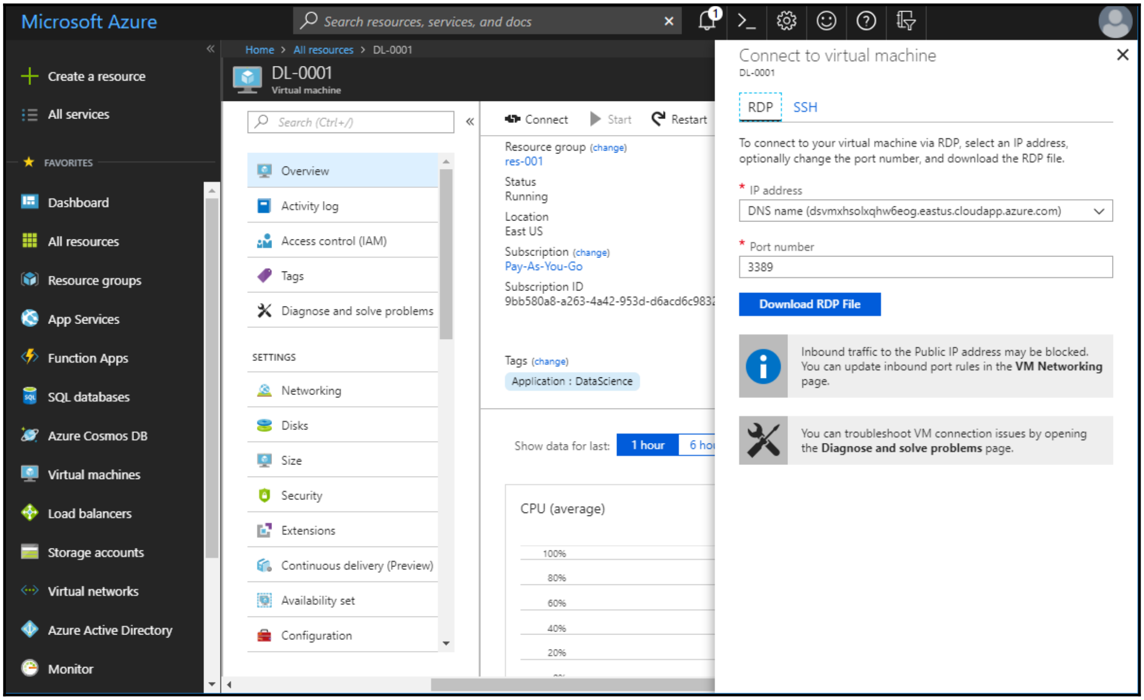


All resources
markhodnetthotmail (Default Directory)

Subscriptions: Pay-As-You-Go – Don't see a subscription? Open Directory + Subscription settings

1 of 7 items selected

	NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION
□	106dff9dc629dsvm	Storage account	res-001	East US	Pay-As-You-Go
□	DL-0001	Disk	RES-001	East US	Pay-As-You-Go
☑	DL-0001	Virtual machine	res-001	East US	Pay-As-You-Go
□	DL-0001	Network interface	res-001	East US	Pay-As-You-Go
□	DL-0001	Public IP address	res-001	East US	Pay-As-You-Go
□	DL-0001	Virtual network	res-001	East US	Pay-As-You-Go
□	DL-0001_NSG	Network security group	res-001	East US	Pay-As-You-Go



Resource group (change)
res-001

Status
Running

Location
East US

Subscription (change)
Pay-As-You-Go

Subscription ID
9bb580a8-a263-4a42-953d-d6acd6c983

Tags (change)
Application : DataScience

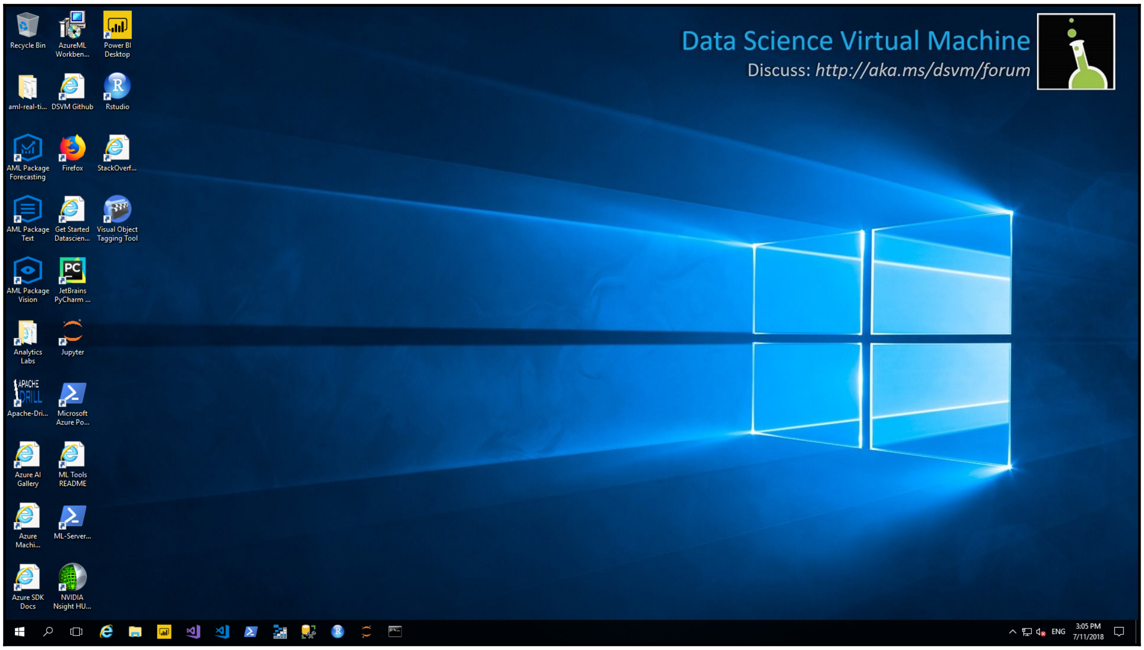
To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file.

* IP address
DNS name (dsvmxhsolxqhw6eog.eastus.cloudapp.azure.com)

* Port number
3389

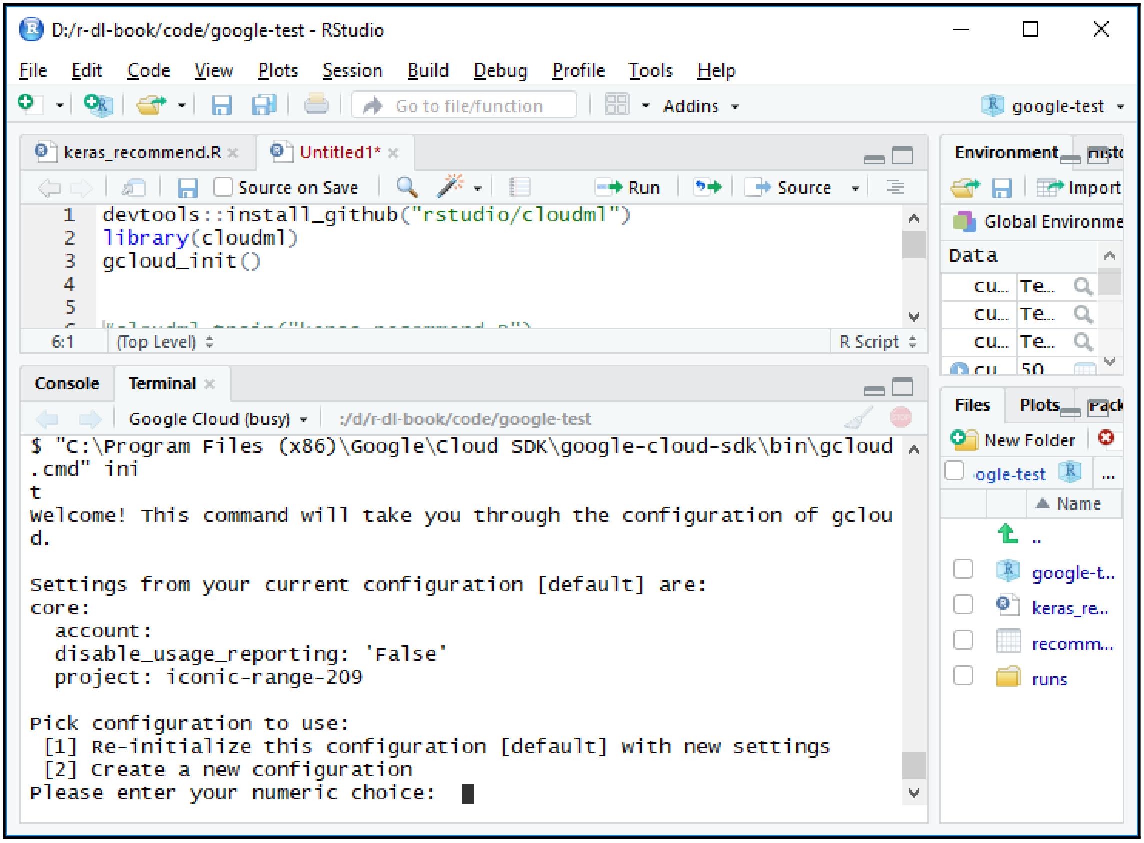
Inbound traffic to the Public IP address may be blocked. You can update inbound port rules in the **VM Networking** page.

You can troubleshoot VM connection issues by opening the **Diagnose and solve problems** page.



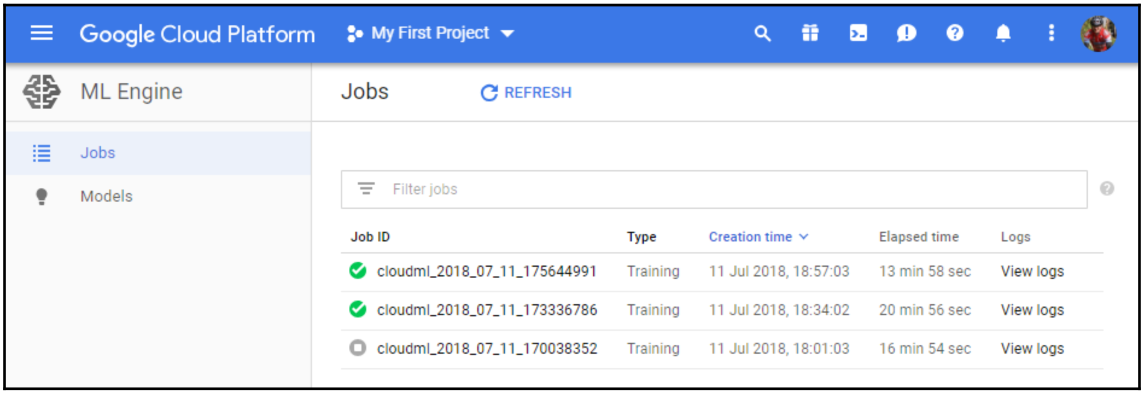
Data Science Virtual Machine

Discuss: *http://aka.ms/dsvm/forum*



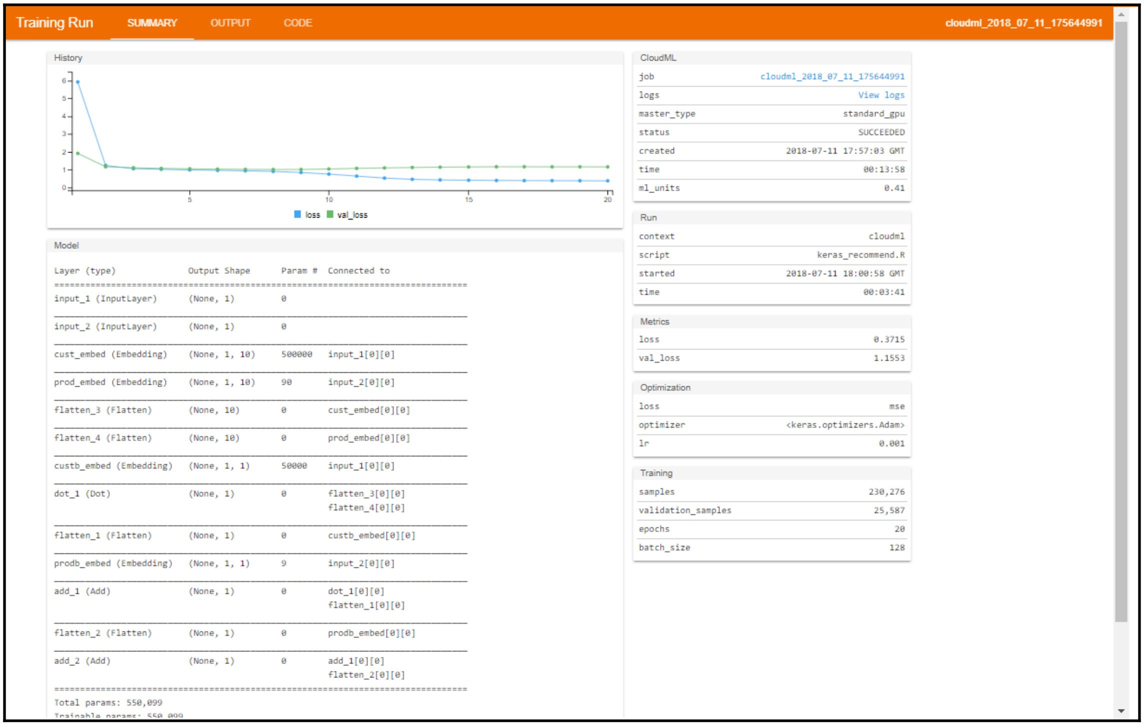
```
devtools::install_github("rstudio/cloudml")
library(cloudml)
gcloud_init()
```

```
$ "C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin\gcloud
.cmd" ini
t
Welcome! This command will take you through the configuration of gclou
d.

Settings from your current configuration [default] are:
core:
  account:
  disable_usage_reporting: 'False'
  project: iconic-range-209

Pick configuration to use:
 [1] Re-initialize this configuration [default] with new settings
 [2] Create a new configuration
Please enter your numeric choice:
```



Jobs

Job ID	Type	Creation time	Elapsed time	Logs
✓ cloudml_2018_07_11_175644991	Training	11 Jul 2018, 18:57:03	13 min 58 sec	View logs
✓ cloudml_2018_07_11_173336786	Training	11 Jul 2018, 18:34:02	20 min 56 sec	View logs
○ cloudml_2018_07_11_170038352	Training	11 Jul 2018, 18:01:03	16 min 54 sec	View logs



Model

```
Layer (type)              Output Shape      Param #   Connected to
================================================================================
input_1 (InputLayer)      (None, 1)         0
________________________________________________________________________________
input_2 (InputLayer)      (None, 1)         0
________________________________________________________________________________
cust_embed (Embedding)    (None, 1, 10)     500000    input_1[0][0]
________________________________________________________________________________
prod_embed (Embedding)    (None, 1, 10)     90        input_2[0][0]
________________________________________________________________________________
flatten_3 (Flatten)       (None, 10)        0         cust_embed[0][0]
________________________________________________________________________________
flatten_4 (Flatten)       (None, 10)        0         prod_embed[0][0]
________________________________________________________________________________
custb_embed (Embedding)   (None, 1, 1)      50000     input_1[0][0]
________________________________________________________________________________
dot_1 (Dot)               (None, 1)         0         flatten_3[0][0]
                                                      flatten_4[0][0]
________________________________________________________________________________
flatten_1 (Flatten)       (None, 1)         0         custb_embed[0][0]
________________________________________________________________________________
prodb_embed (Embedding)   (None, 1, 1)      9         input_2[0][0]
________________________________________________________________________________
add_1 (Add)               (None, 1)         0         dot_1[0][0]
                                                      flatten_1[0][0]
________________________________________________________________________________
flatten_2 (Flatten)       (None, 1)         0         prodb_embed[0][0]
________________________________________________________________________________
add_2 (Add)               (None, 1)         0         add_1[0][0]
                                                      flatten_2[0][0]
================================================================================
Total params: 550,099
Trainable params: 550,099
```

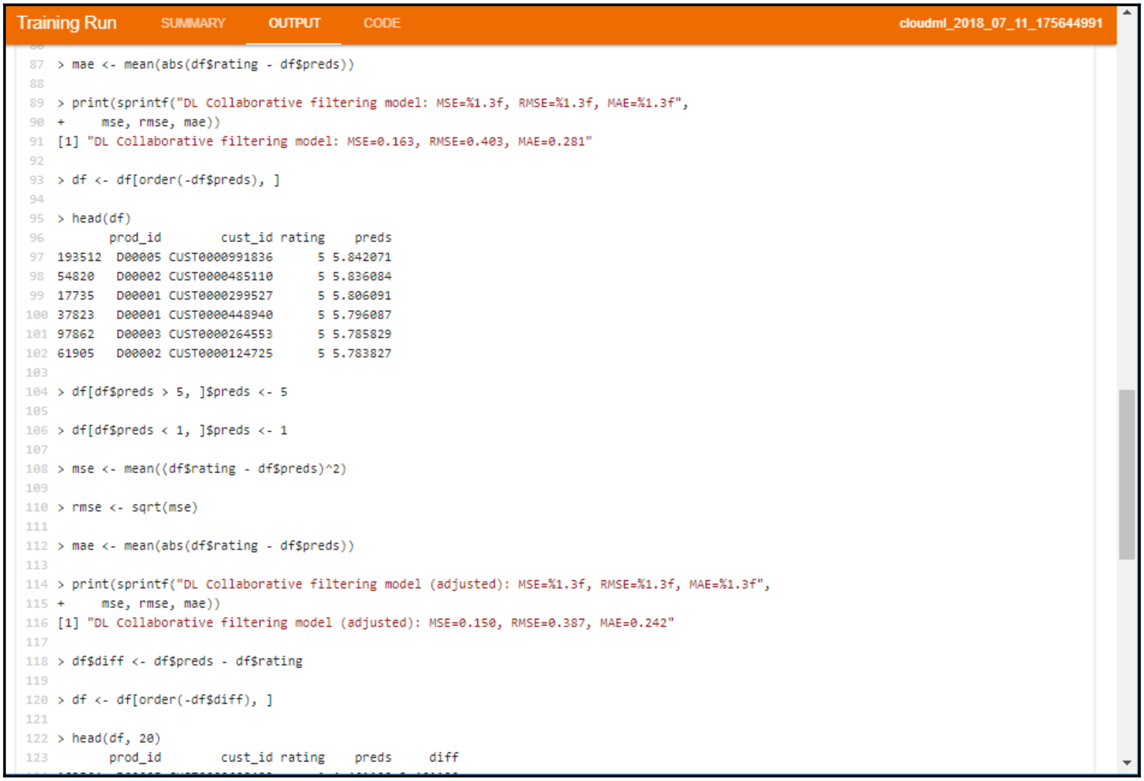
CloudML	
job	cloudml_2018_07_11_175644991
logs	View logs
master_type	standard_gpu
status	SUCCEEDED
created	2018-07-11 17:57:03 GMT
time	00:13:58
ml_units	0.41

Run	
context	cloudml
script	keras_recommend.R
started	2018-07-11 18:00:58 GMT
time	00:03:41

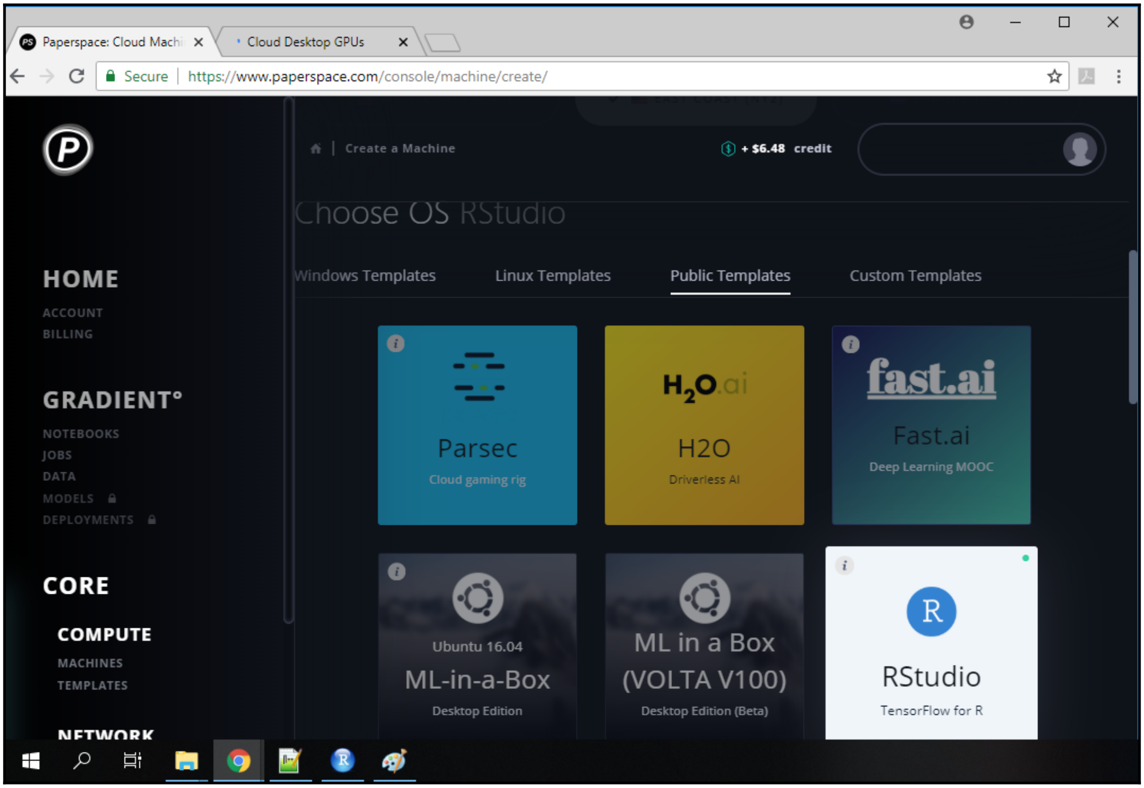
Metrics	
loss	0.3715
val_loss	1.1553

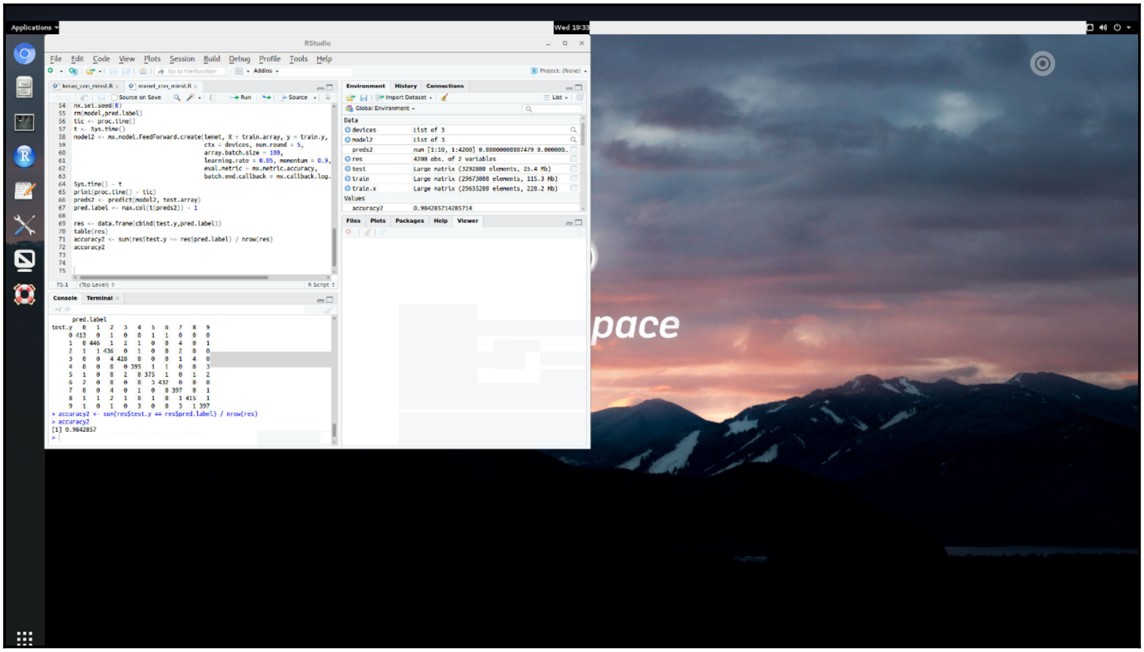
Optimization	
loss	mse
optimizer	<keras.optimizers.Adam>
lr	0.001

Training	
samples	230,276
validation_samples	25,587
epochs	20
batch_size	128



```
> mae <- mean(abs(df$rating - df$preds))

> print(sprintf("DL Collaborative filtering model: MSE=%1.3f, RMSE=%1.3f, MAE=%1.3f",
+     mse, rmse, mae))
[1] "DL Collaborative filtering model: MSE=0.163, RMSE=0.403, MAE=0.281"

> df <- df[order(-df$preds), ]

> head(df)
       prod_id        cust_id rating    preds
193512  D00005 CUST0000991836      5 5.842071
54820   D00002 CUST0000485110      5 5.836084
17735   D00001 CUST0000299527      5 5.806091
37823   D00001 CUST0000448940      5 5.796087
97862   D00003 CUST0000264553      5 5.785829
61905   D00002 CUST0000124725      5 5.783827

> df[df$preds > 5, ]$preds <- 5

> df[df$preds < 1, ]$preds <- 1

> mse <- mean((df$rating - df$preds)^2)

> rmse <- sqrt(mse)

> mae <- mean(abs(df$rating - df$preds))

> print(sprintf("DL Collaborative filtering model (adjusted): MSE=%1.3f, RMSE=%1.3f, MAE=%1.3f",
+     mse, rmse, mae))
[1] "DL Collaborative filtering model (adjusted): MSE=0.150, RMSE=0.387, MAE=0.242"

> df$diff <- df$preds - df$rating

> df <- df[order(-df$diff), ]

> head(df, 20)
       prod_id        cust_id rating    preds     diff
```


Paperspace: Cloud Machi
Cloud Desktop GPUs
Secure
https://www.paperspace.com/console/machine/create/
Create a Machine
+ $6.48 credit
Choose OS RStudio
HOME
ACCOUNT
BILLING
GRADIENT°
NOTEBOOKS
JOBS
DATA
MODELS
DEPLOYMENTS
CORE
COMPUTE
MACHINES
TEMPLATES
NETWORK
Windows Templates
Linux Templates
Public Templates
Custom Templates
Parsec
Cloud gaming rig
H2O.ai
H2O
Driverless AI
fast.ai
Fast.ai
Deep Learning MOOC
Ubuntu 16.04
ML-in-a-Box
Desktop Edition
ML in a Box
(VOLTA V100)
Desktop Edition (Beta)
R
RStudio
TensorFlow for R

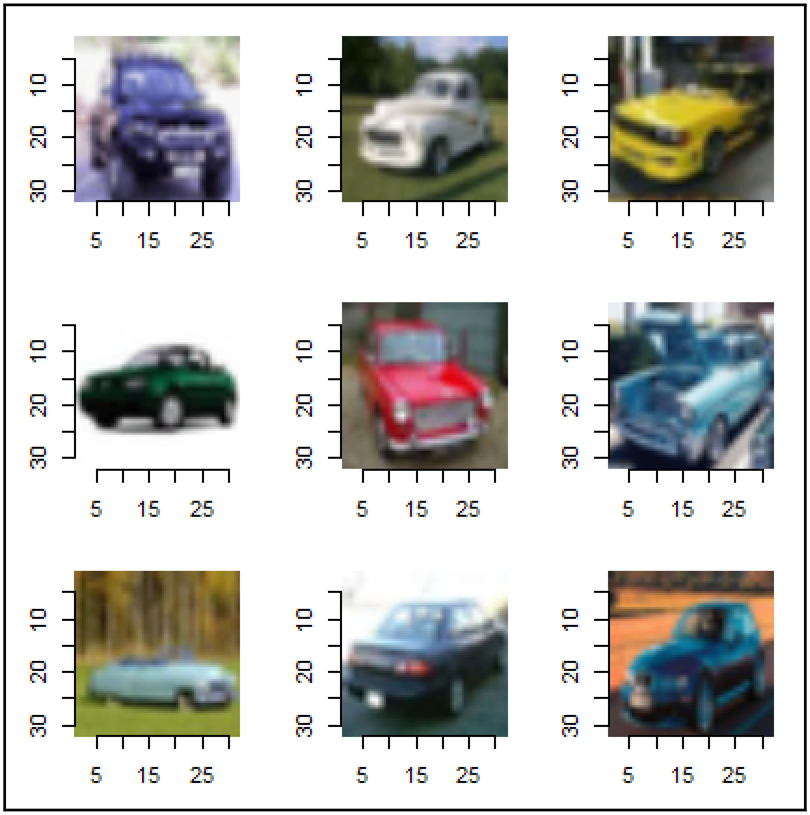


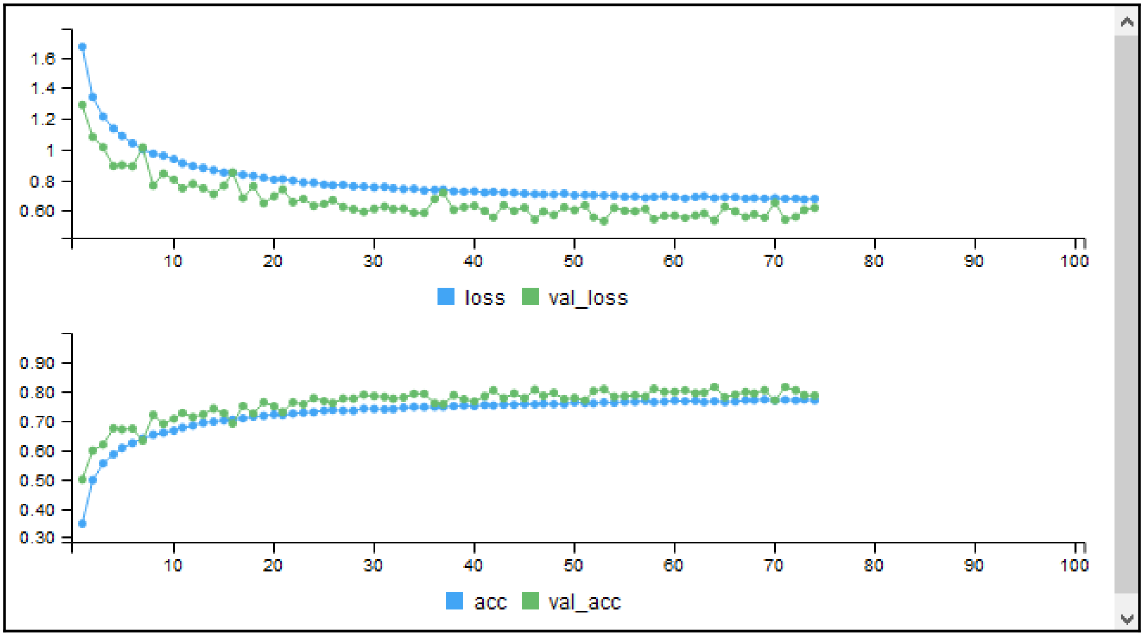
```
mx.set.seed(0)
rm(model,pred.label)
tic <- proc.time()
t <- Sys.time()
model2 <- mx.model.FeedForward.create(lenet, X = train.array, y = train.y,
                                       ctx = devices, num.round = 5,
                                       array.batch.size = 100,
                                       learning.rate = 0.05, momentum = 0.9,
                                       eval.metric = mx.metric.accuracy,
                                       batch.end.callback = mx.callback.log.
Sys.time() - t
print(proc.time() - tic)
preds2 <- predict(model2, test.array)
pred.label <- max.col(t(preds2)) - 1

res <- data.frame(cbind(test.y,pred.label))
table(res)
accuracy2 <- sum(res$test.y == res$pred.label) / nrow(res)
accuracy2
```

```
      pred.label
test.y   0   1   2   3   4   5   6   7   8   9
     0 413   0   1   0   0   1   1   0   0   0
     1   0 446   1   2   1   0   0   4   0   1
     2   1   1 436   0   1   0   0   2   0   0
     3   0   0   4 428   0   0   0   1   4   0
     4   0   0   0   0 395   1   1   0   0   3
     5   1   0   0   2   0 375   1   0   1   2
     6   2   0   0   0   0   3 432   0   0   0
     7   0   0   4   0   1   0   0 397   0   1
     8   1   1   2   1   0   1   0   1 415   1
     9   1   0   1   0   3   0   0   3   1 397
> accuracy2 <- sum(res$test.y == res$pred.label) / nrow(res)
> accuracy2
[1] 0.9842857
> 
```

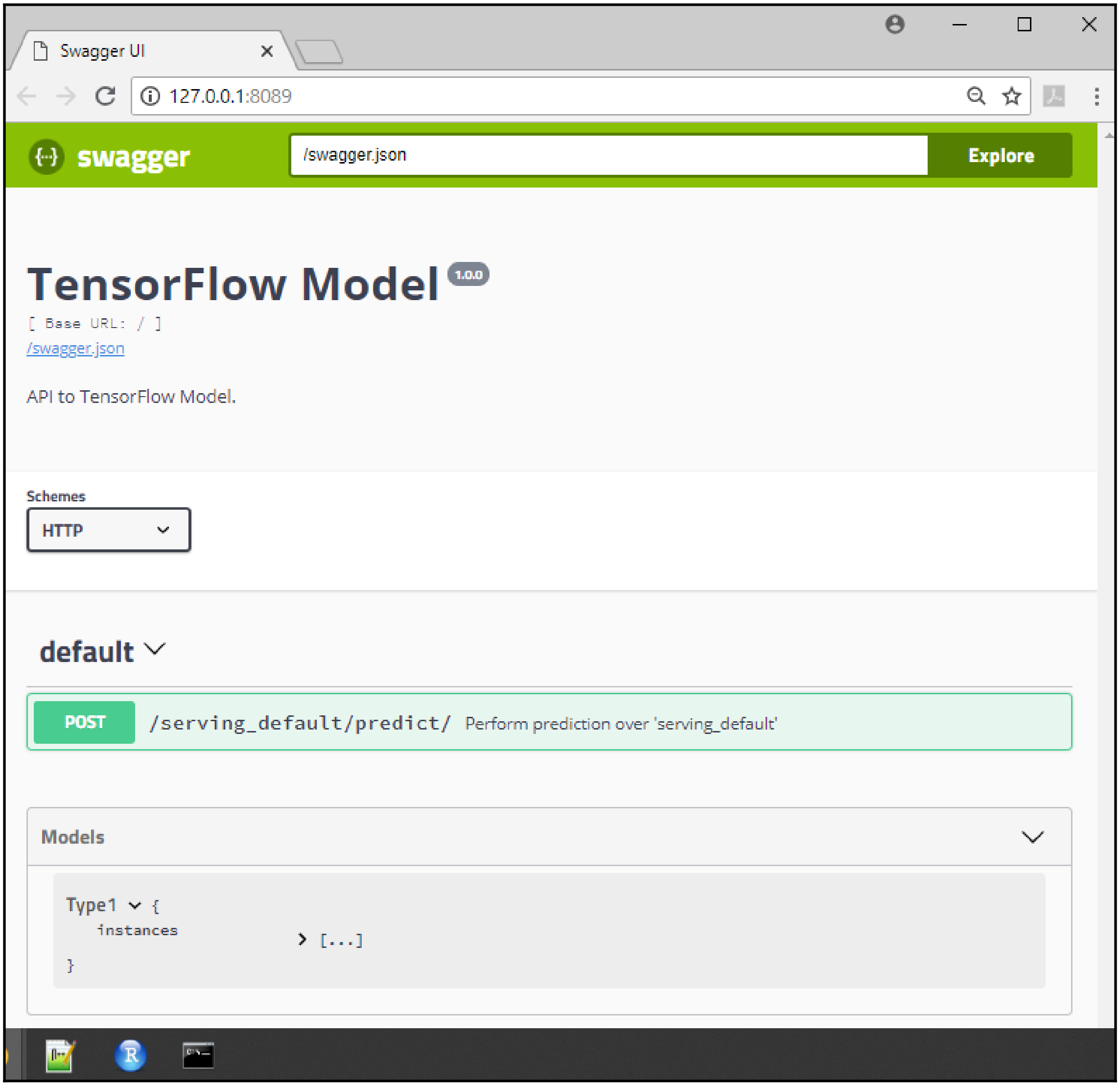
```
Data
devices      List of 3
model2       List of 3
preds2       num [1:10, 1:4200] 0.00000000007479 0.000003...
res          4200 obs. of 2 variables
test         Large matrix (3292800 elements, 25.4 Mb)
train        Large matrix (29673000 elements, 115.3 Mb)
train.x      Large matrix (29635200 elements, 228.2 Mb)
Values
accuracy2    0.984285714285714
```

Chapter 11: The Next Level in Deep Learning




1.6
1.4
1.2
1
0.8
0.60
10
20
30
40
50
60
70
80
90
100
🟦 loss 🟩 val_loss
0.90
0.80
0.70
0.60
0.50
0.40
0.30
10
20
30
40
50
60
70
80
90
100
🟦 acc 🟩 val_acc

FELT
feltbicycles.com
superlite



TensorFlow Model 1.0.0

[Base URL: /]

/swagger.json

API to TensorFlow Model.

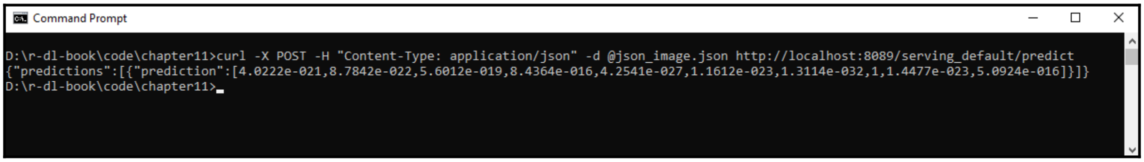
Schemes: HTTP

default

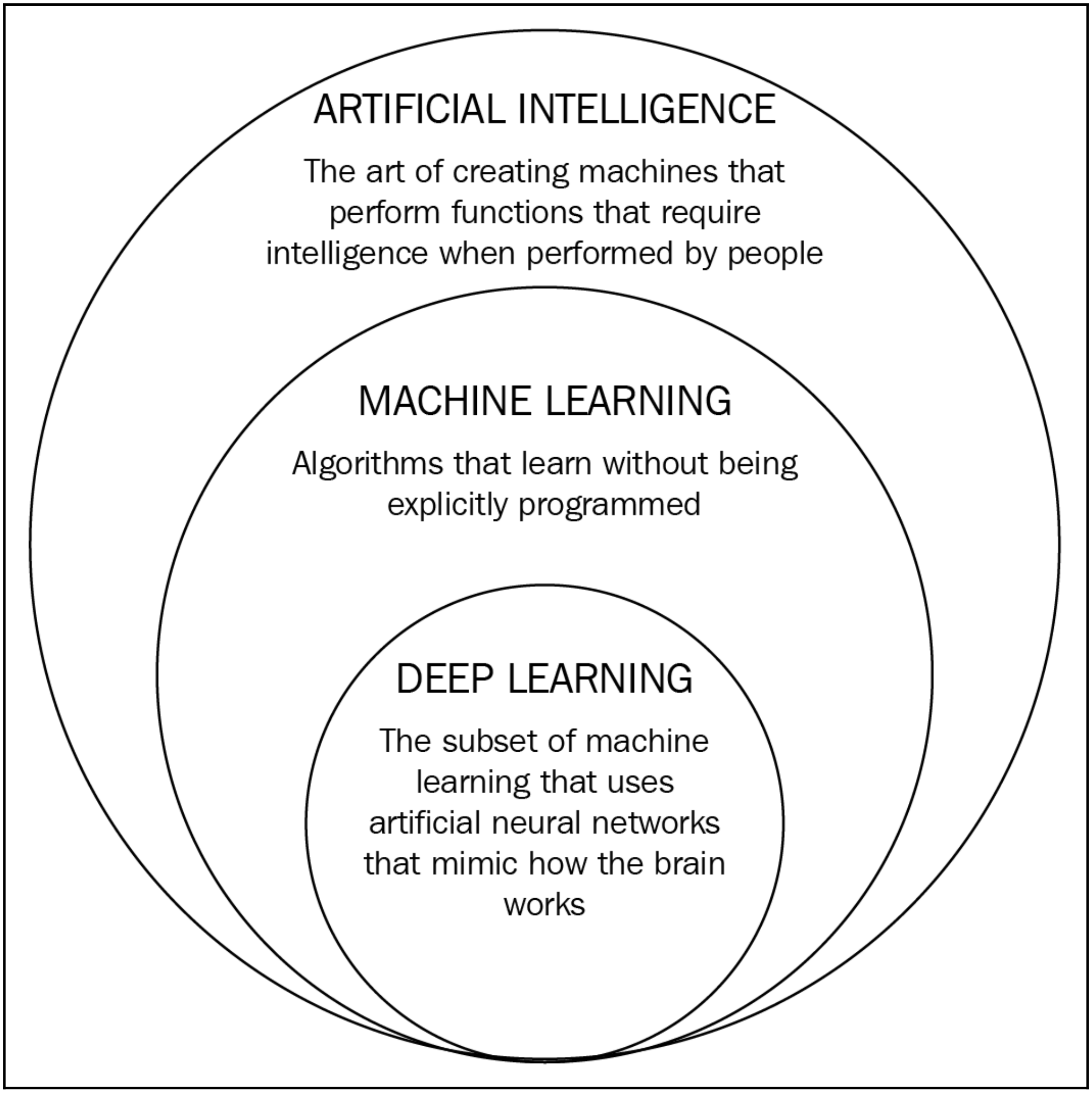
POST /serving_default/predict/ Perform prediction over 'serving_default'

Models

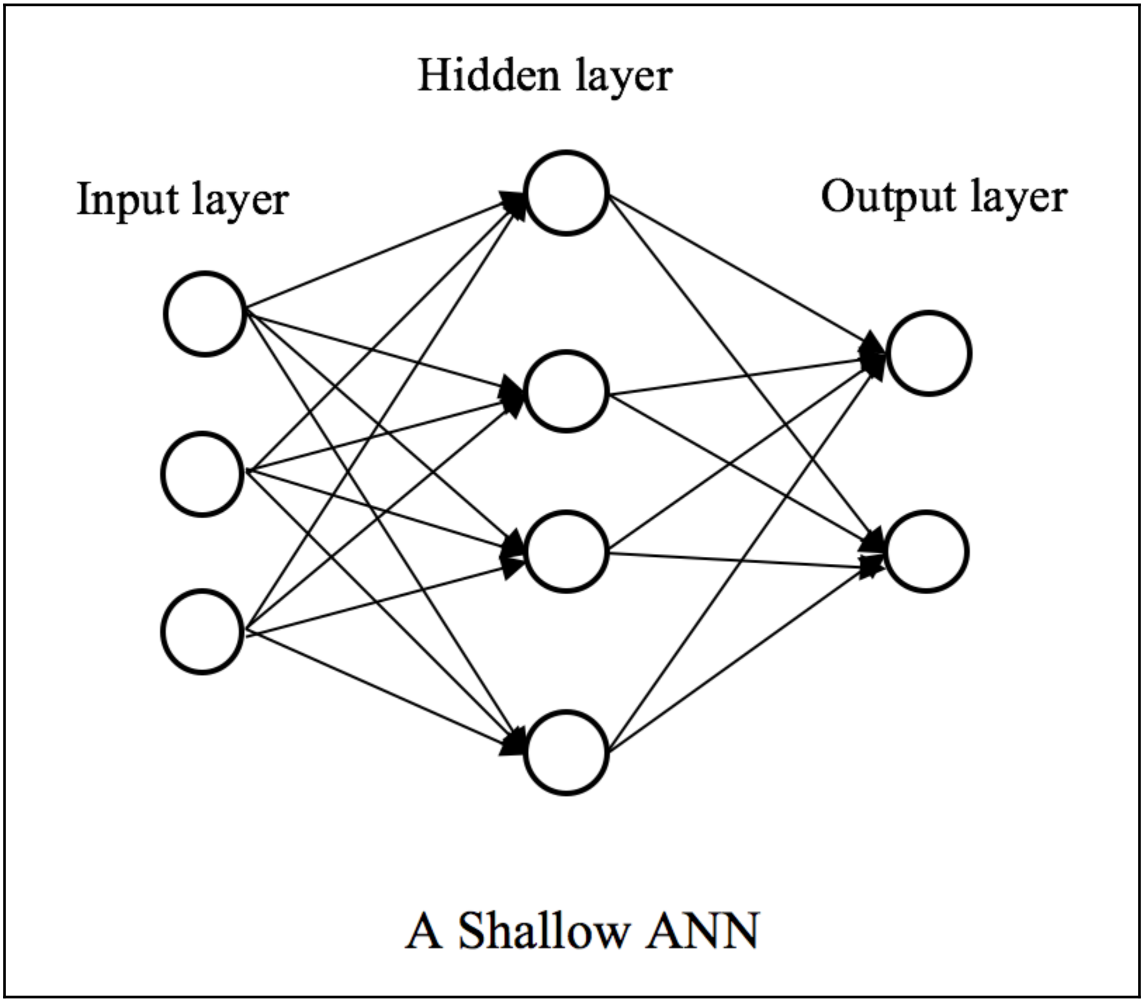
```
Type1 {
   instances        > [...]
}
```

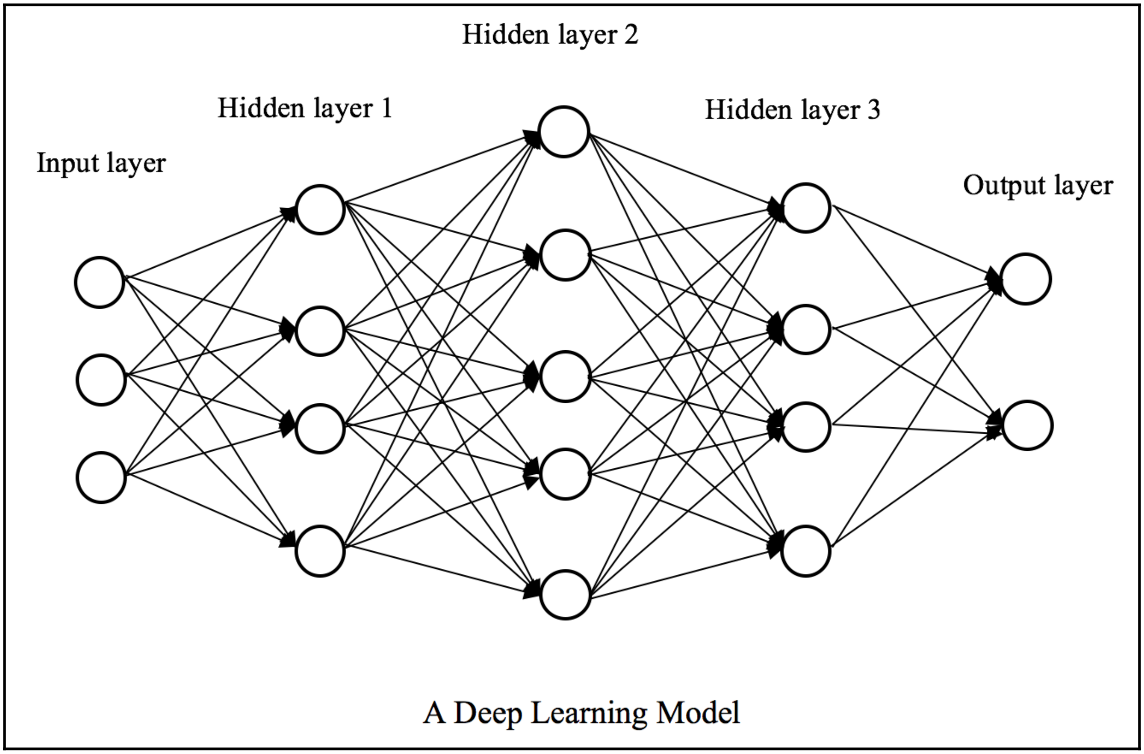


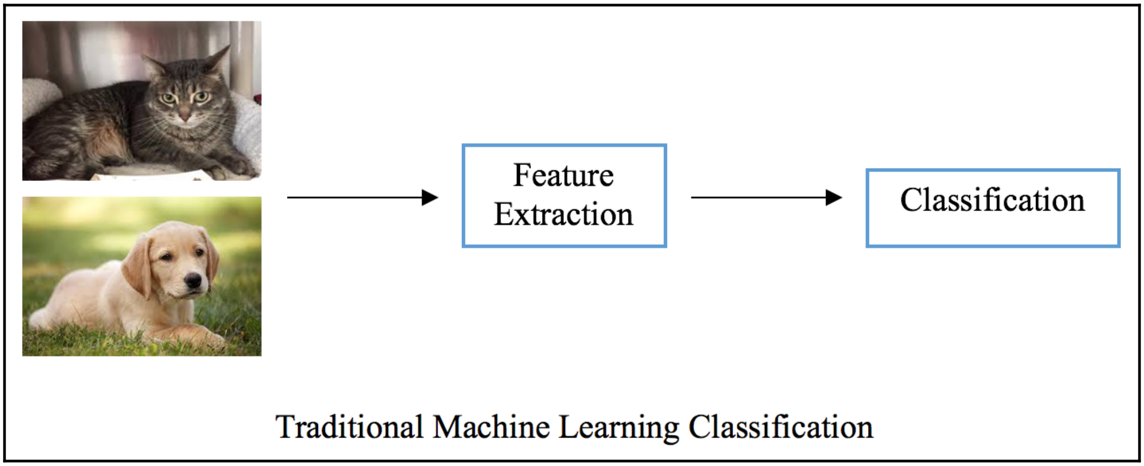
```
D:\r-dl-book\code\chapter11>curl -X POST -H "Content-Type: application/json" -d @json_image.json http://localhost:8089/serving_default/predict
{"predictions":[{"prediction":[4.0222e-021,8.7842e-022,5.6012e-019,8.4364e-016,4.2541e-027,1.1612e-023,1.3114e-032,1,1.4477e-023,5.0924e-016]}]}
D:\r-dl-book\code\chapter11>
```

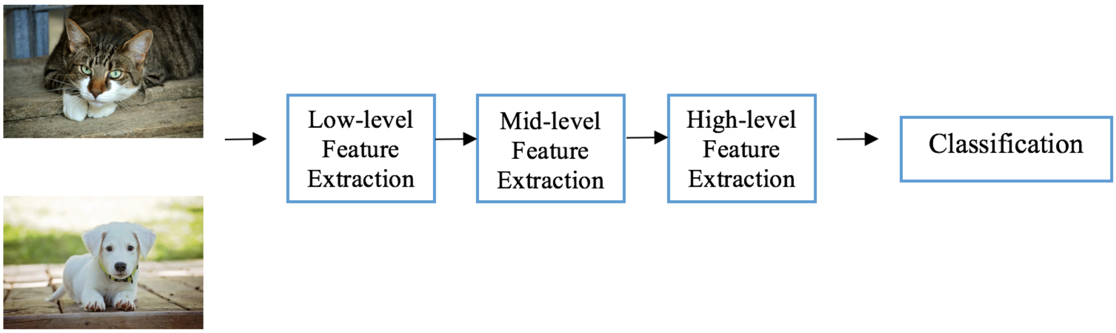

ARTIFICIAL INTELLIGENCE
The art of creating machines that perform functions that require intelligence when performed by people
MACHINE LEARNING
Algorithms that learn without being explicitly programmed
DEEP LEARNING
The subset of machine learning that uses artificial neural networks that mimic how the brain works

Chapter 12: Handwritten Digit Recognition using Convolutional Neural Networks

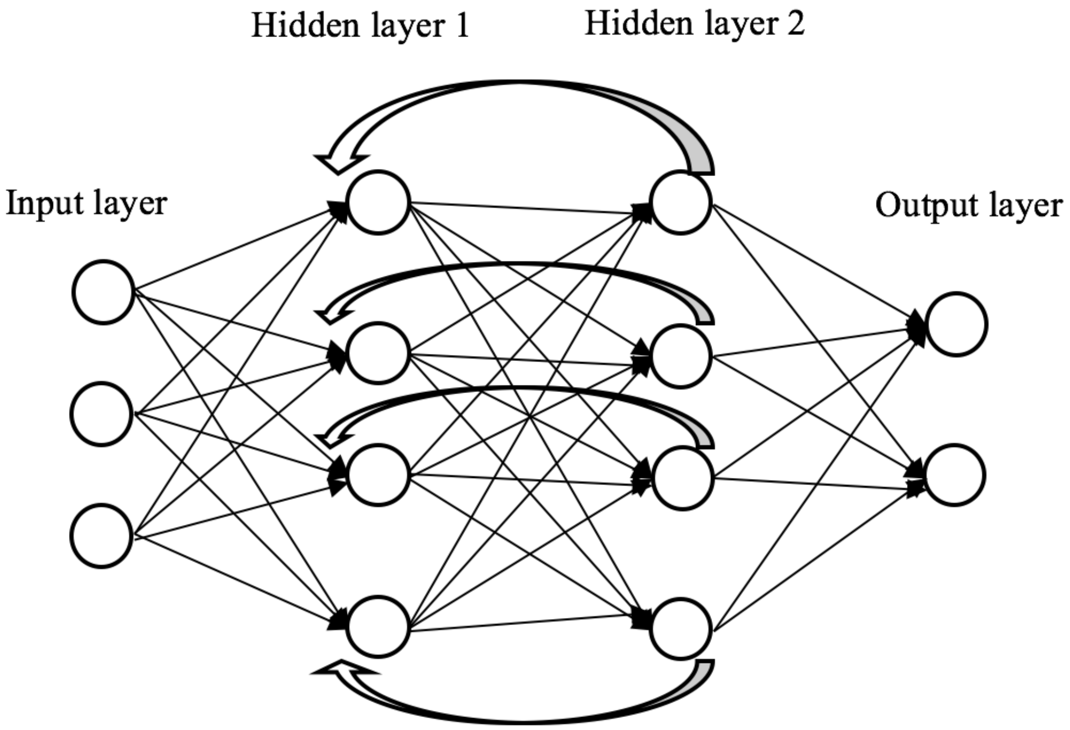



Hidden layer 2
Hidden layer 1
Hidden layer 3
Input layer
Output layer
A Deep Learning Model

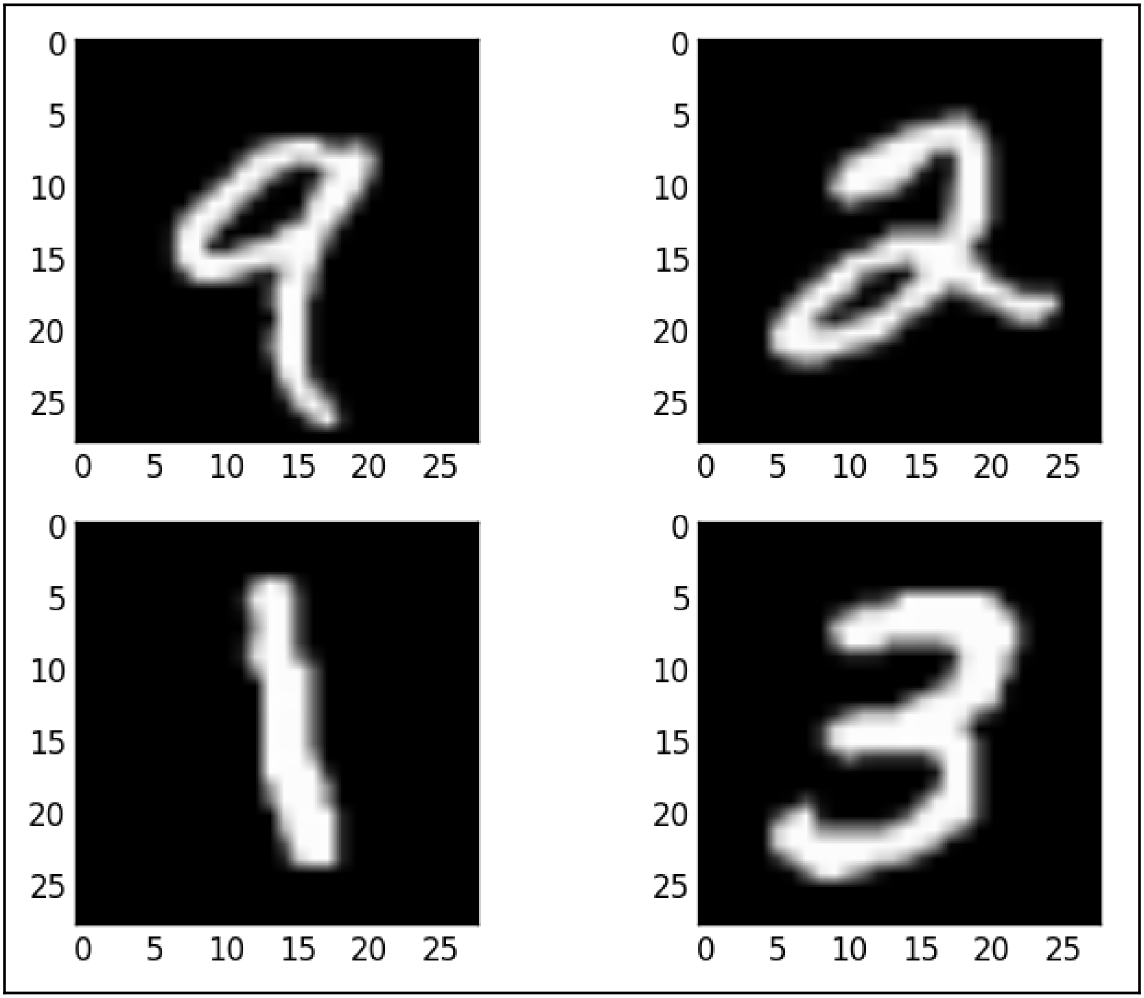

Feature Extraction
Classification
Traditional Machine Learning Classification

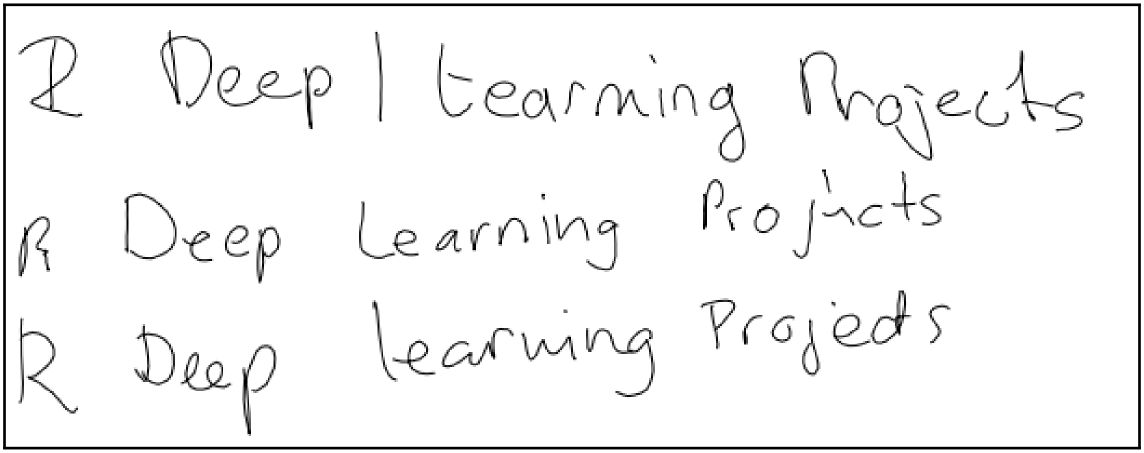

Low-level Feature Extraction
Mid-level Feature Extraction
High-level Feature Extraction
Classification

Deep Learning Based Classification



A Recurrent Neural Network

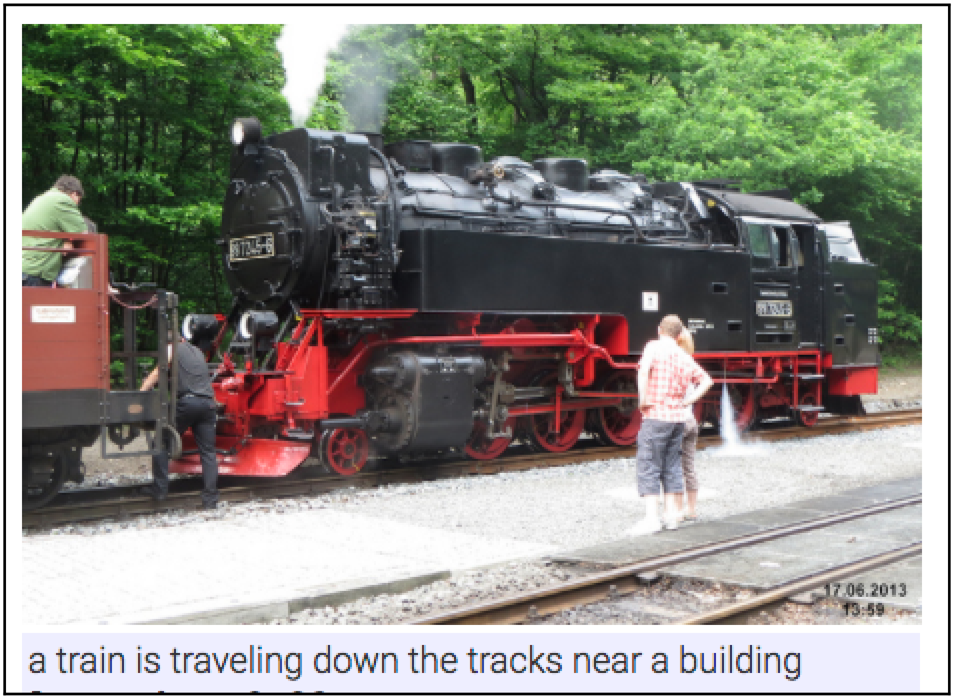

0
5
10
15
20
25
0
5
10
15
20
25



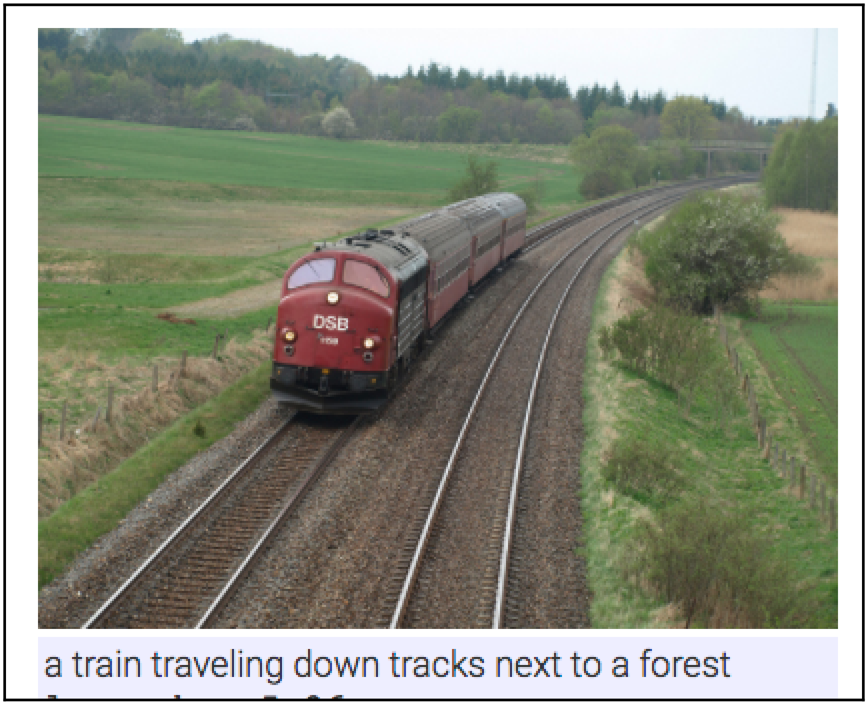
R Deep | Learning Projects

R Deep Learning Projects

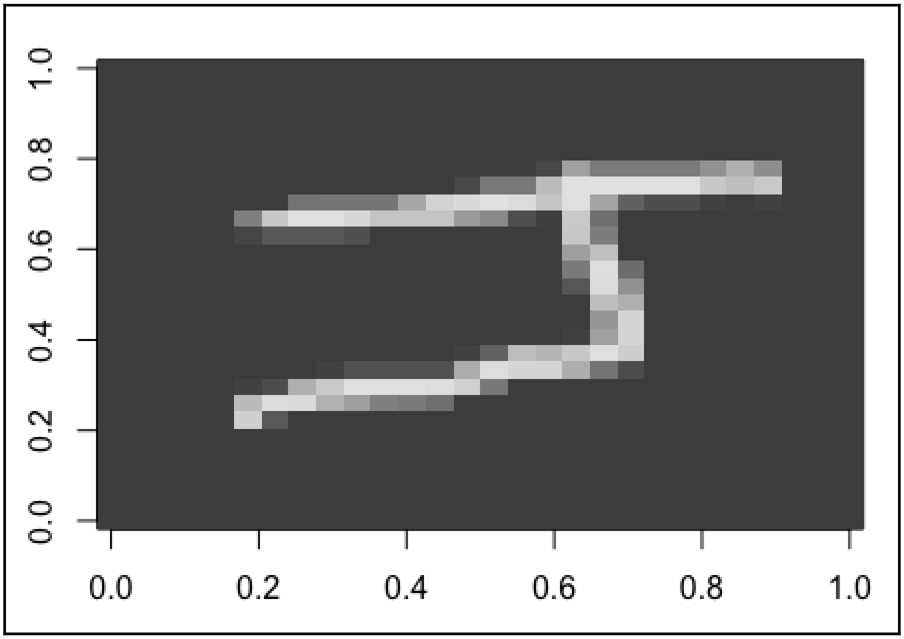
R Deep learning Projects

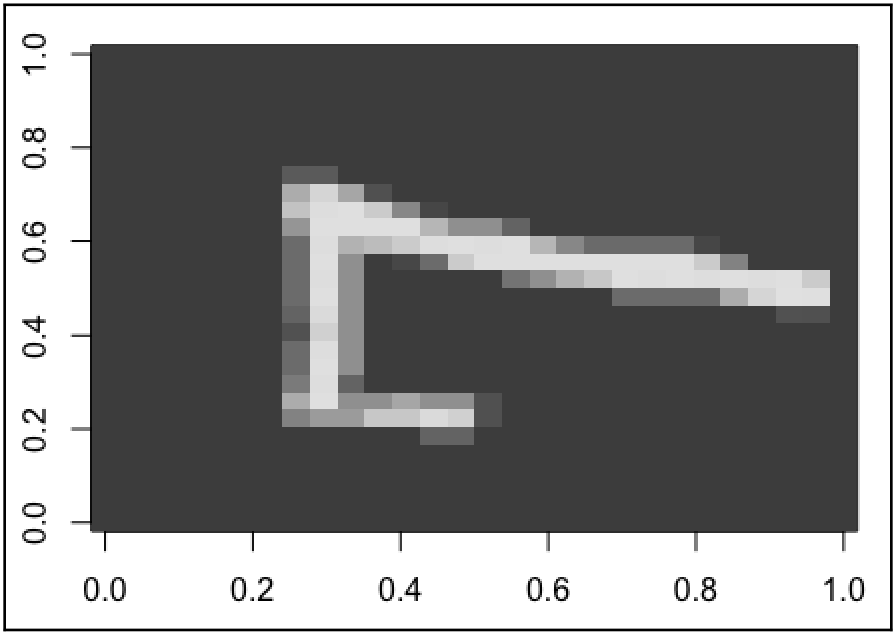


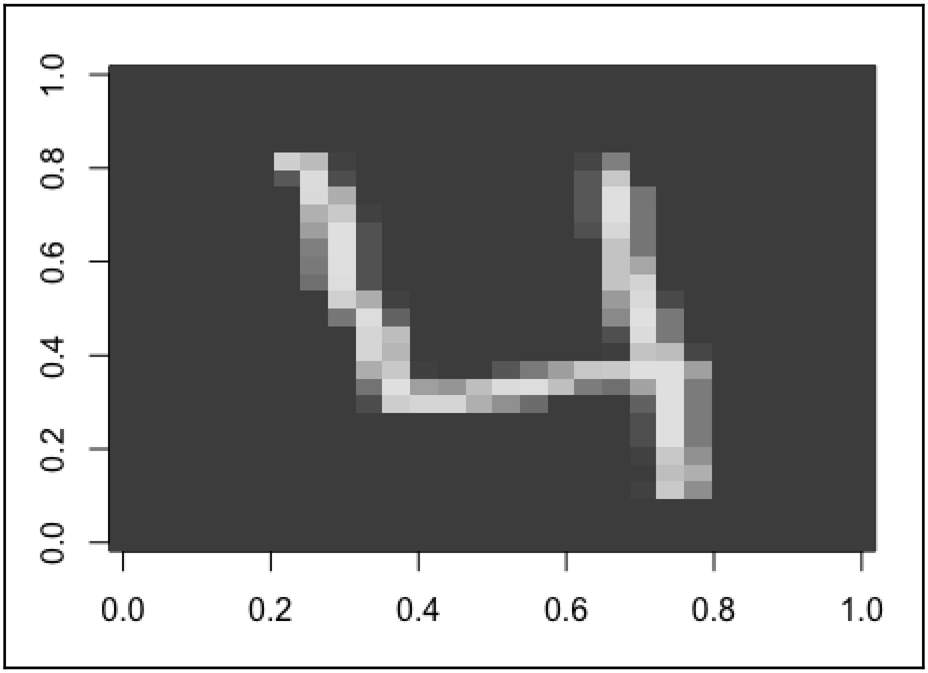
a train is traveling down the tracks near a building

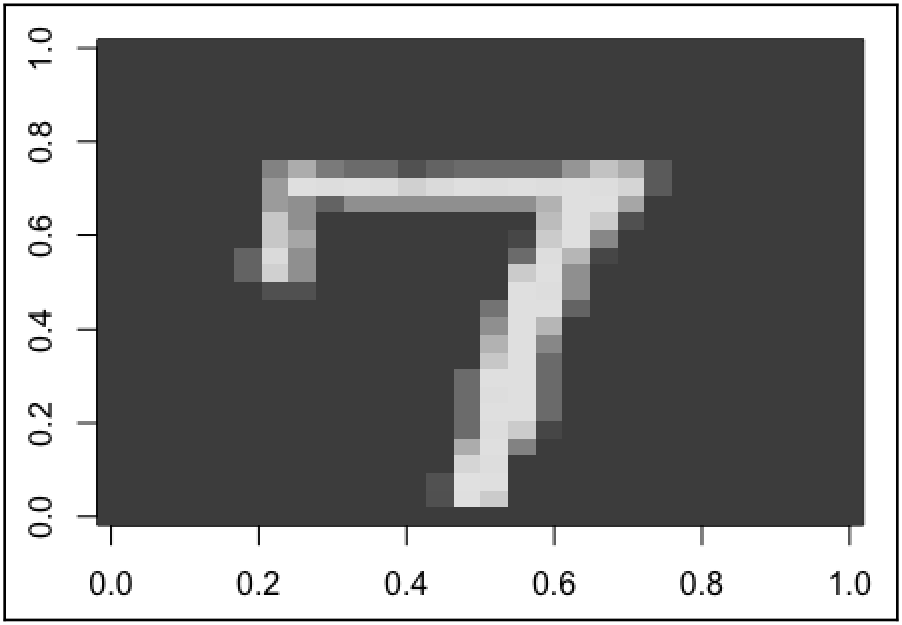


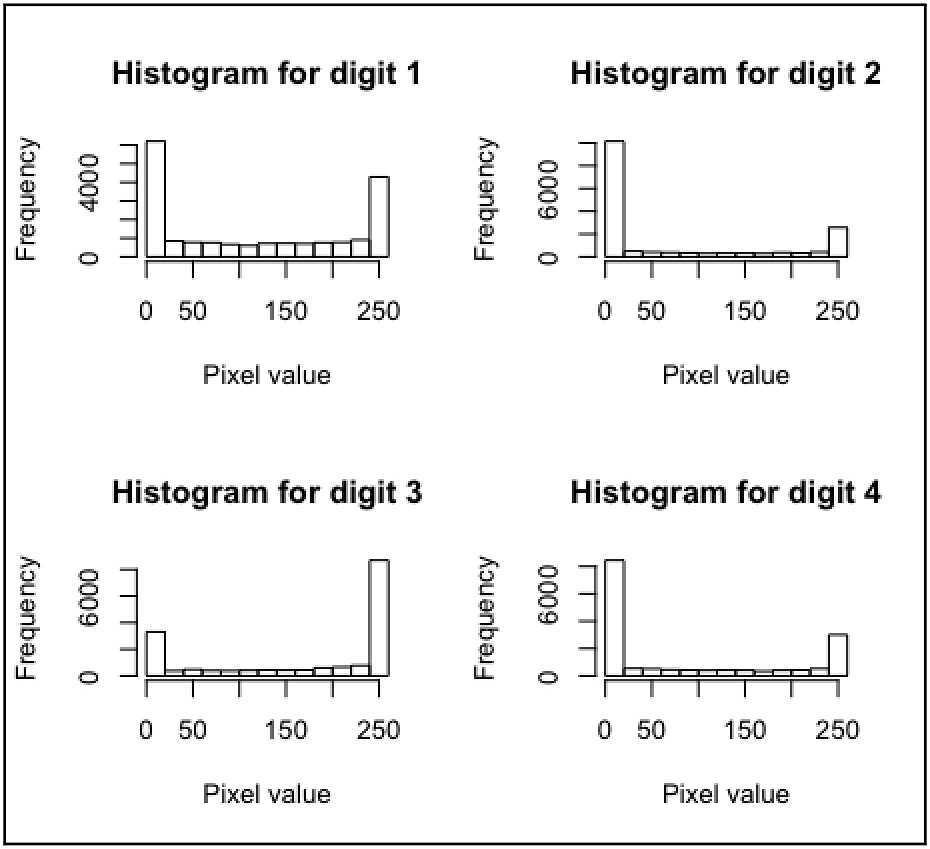
a train traveling down tracks next to a forest

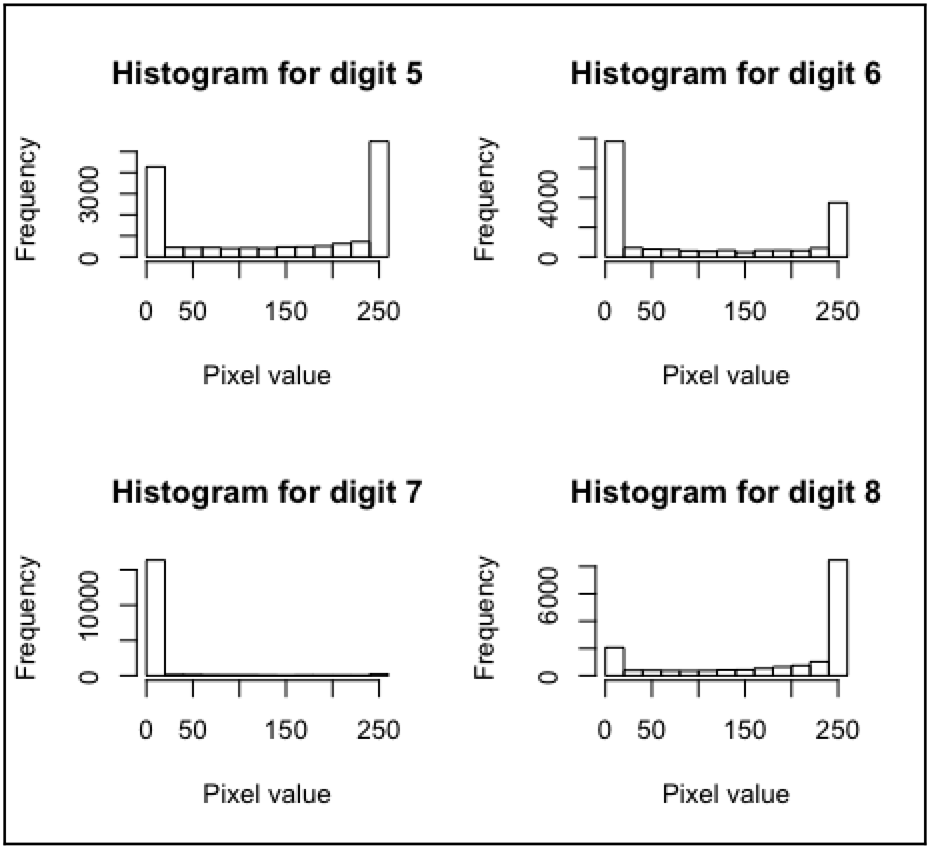


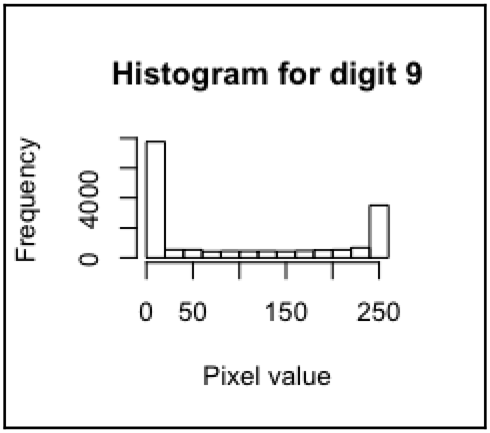

0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

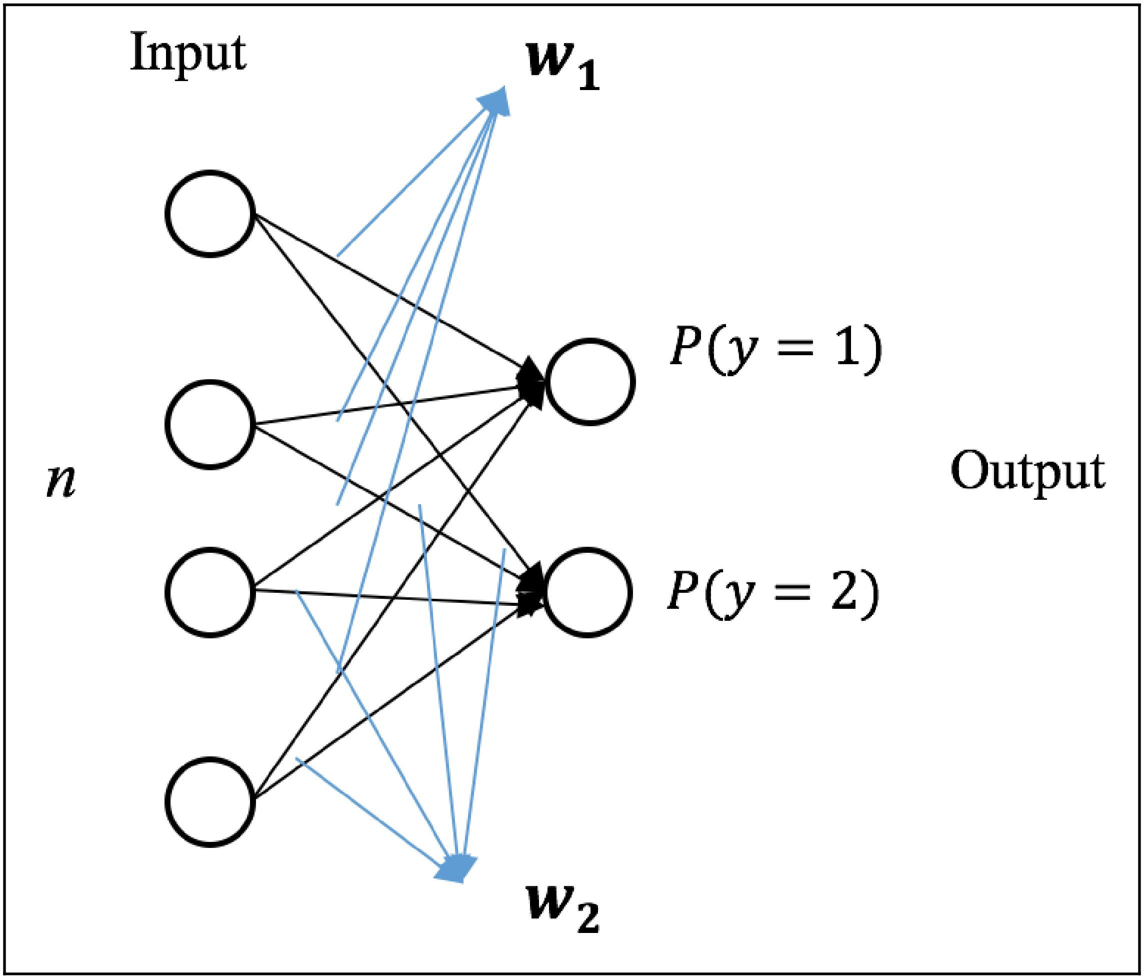

0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Histogram for digit 1
Frequency
0
4000
0
50
150
250
Pixel value
Histogram for digit 2
Frequency
0
6000
0
50
150
250
Pixel value
Histogram for digit 3
Frequency
0
6000
0
50
150
250
Pixel value
Histogram for digit 4
Frequency
0
6000
0
50
150
250
Pixel value


Histogram for digit 5
Frequency
0
3000
0
50
150
250
Pixel value
Histogram for digit 6
Frequency
0
4000
0
50
150
250
Pixel value
Histogram for digit 7
Frequency
0
10000
0
50
150
250
Pixel value
Histogram for digit 8
Frequency
0
6000
0
50
150
250
Pixel value


Histogram for digit 9
Frequency
0
4000
0
50
150
250
Pixel value



$$P(y = k|x, W) = softmax_k(Wx)$$

$$= \frac{exp(w_k x)}{\sum_{j=1}^{j=k} exp(w_j x)}$$

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(i)}, y^{(i)}) \ldots, (x^{(m)}, y^{(m)})$$

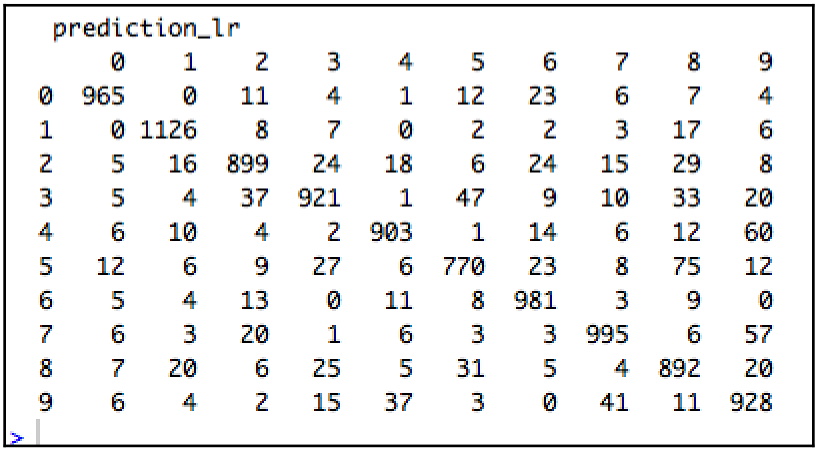
$$y \in 1, 2, \ldots, K$$

$$J(W) = -\left[\sum_{i=1}^{m} \sum_{k=1}^{k} 1\{y^{(i)} = k\} log \frac{exp(w_k x)}{\sum_{j=1}^{j=k} exp(w_j x)}\right]$$

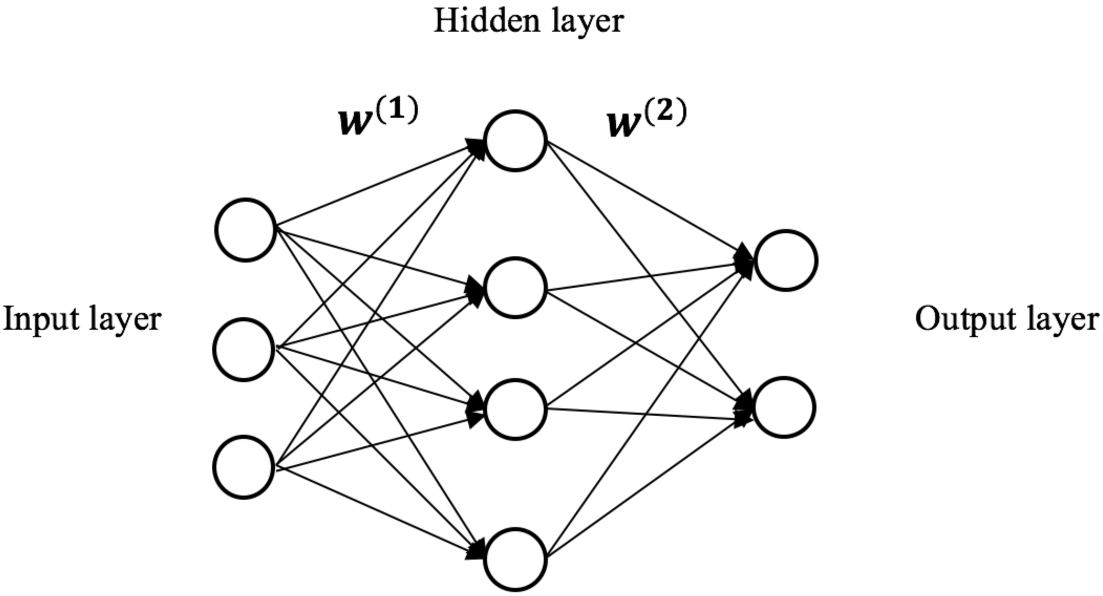
$$1\{y^{(i)} = k\} = \begin{cases} 1, & if\ \ y^{(i)} = k \\ 0, & otherwise \end{cases}$$

$$\triangle w_k = \frac{\partial}{\partial w_j} J(w) = -\sum_{i=1}^{m} [x^{(i)}(1\{y^{(i)} = k\} - P(y^{(i)} = k \mid x^{(i)}, W))]$$

$$y' = \underset{k}{\operatorname{argmax}} \frac{\exp\left(\boldsymbol{w}_k \boldsymbol{x}'\right)}{\sum_{j=1}^{j=K} \exp\left(\boldsymbol{w}_j \boldsymbol{x}'\right)} = \underset{k}{\operatorname{argmax}} \left(\boldsymbol{w}_k \boldsymbol{x}'\right)$$



```
   prediction_lr
       0    1    2    3    4    5    6    7    8    9
  0  965    0   11    4    1   12   23    6    7    4
  1    0 1126    8    7    0    2    2    3   17    6
  2    5   16  899   24   18    6   24   15   29    8
  3    5    4   37  921    1   47    9   10   33   20
  4    6   10    4    2  903    1   14    6   12   60
  5   12    6    9   27    6  770   23    8   75   12
  6    5    4   13    0   11    8  981    3    9    0
  7    6    3   20    1    6    3    3  995    6   57
  8    7   20    6   25    5   31    5    4  892   20
  9    6    4    2   15   37    3    0   41   11  928
>
```



$$a_h^{(2)} = f(z^{(2)}) = f(w_h^{(1)} x)$$

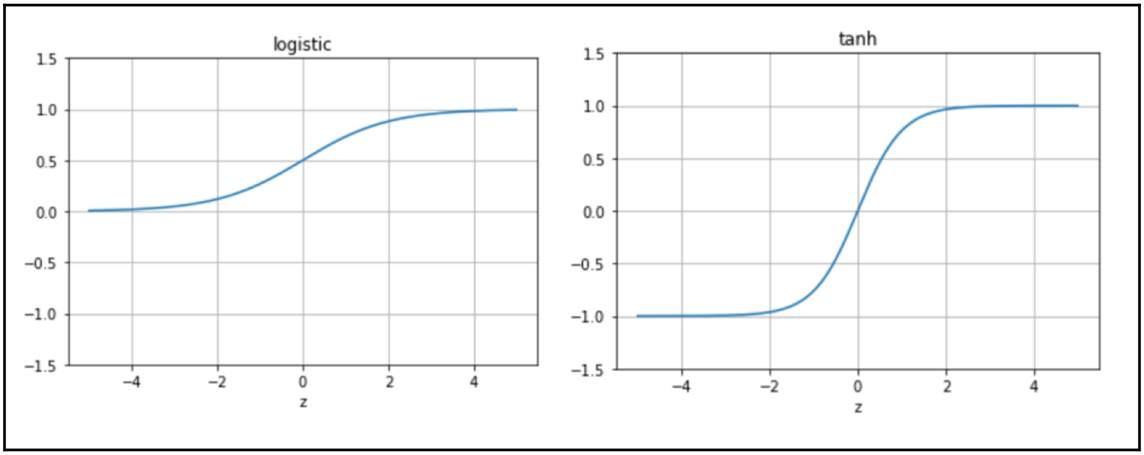
$$a_1^{(2)} = f(w_1^{(1)} x)$$

$$a_H^{(2)} = f(w_H^{(1)} x)$$

$$a_H^{(2)} = f(w_H^{(1)} x)$$

$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

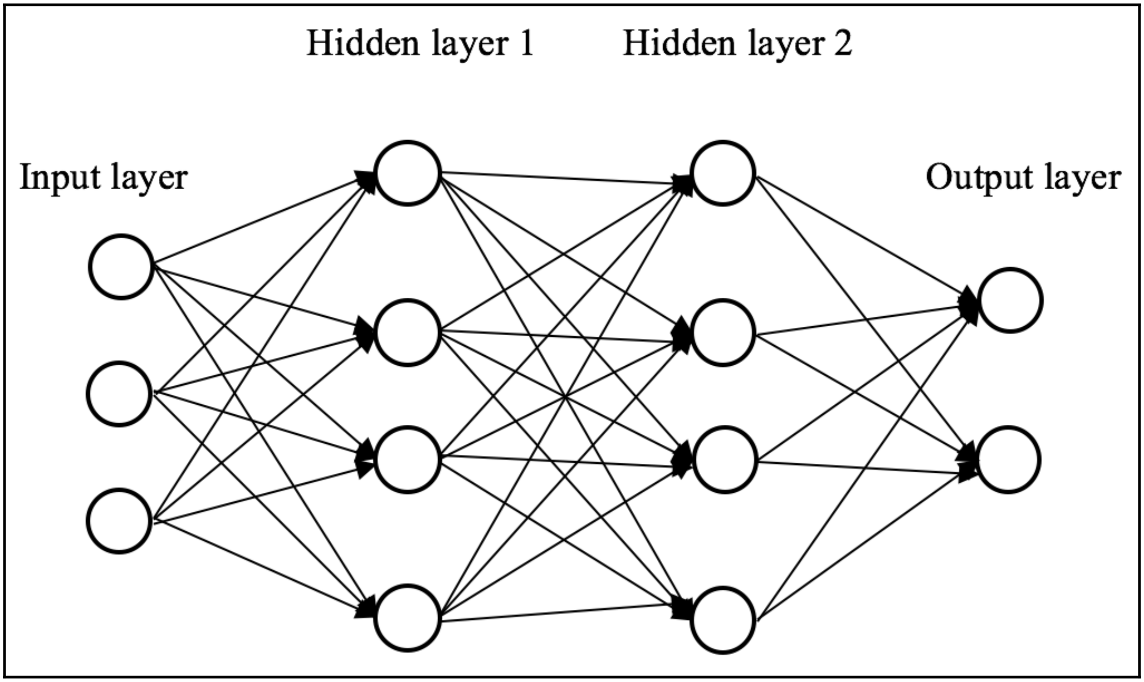
$$tanh(z) = \frac{e^{-z} - e^{-z}}{e^{z} + e^{-z}} = \frac{2}{1 + e^{-2z}} - 1$$



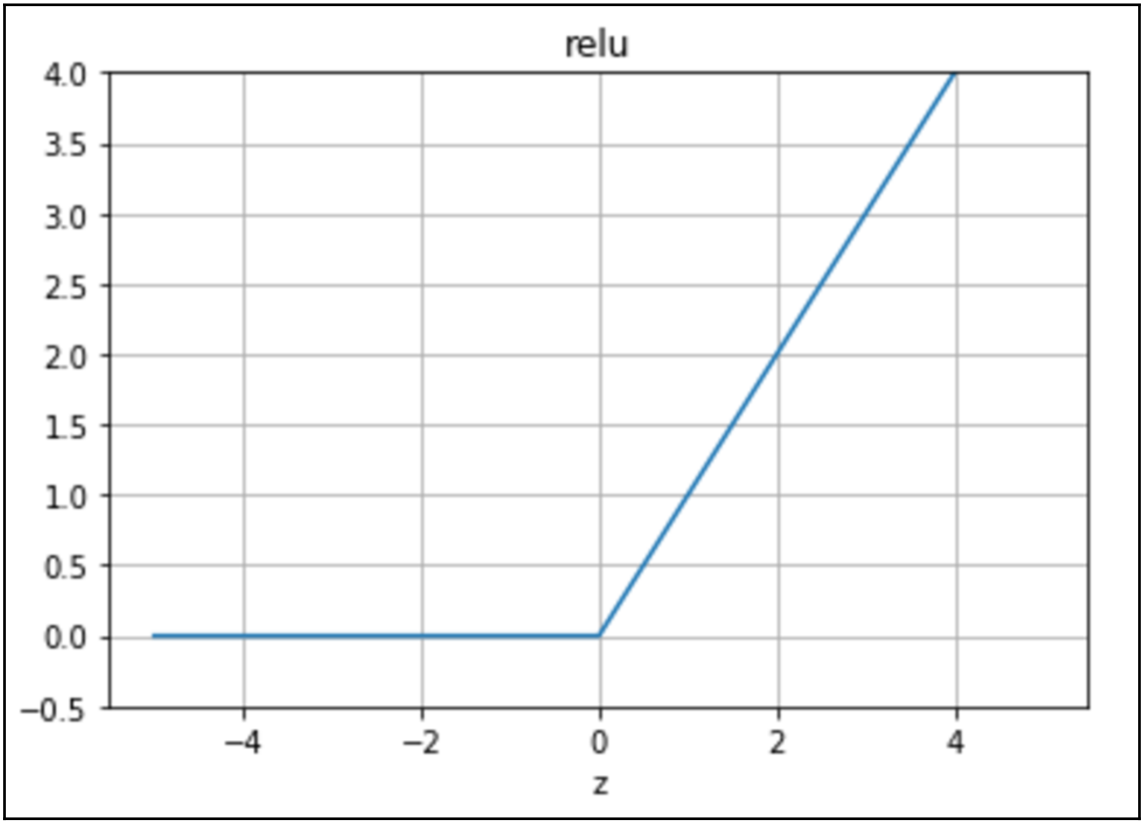
$$a_k^{(3)} = f(z^{(3)}) = softmax_k(W^{(2)}a^{(2)})$$

$$\triangle W^{(2)} = \frac{\partial J(W)}{\partial z_k^{(3)}} \frac{\partial z_k^{(3)}}{\partial W^{(2)}} = \delta^{(3)}(a^{(2)})^T$$

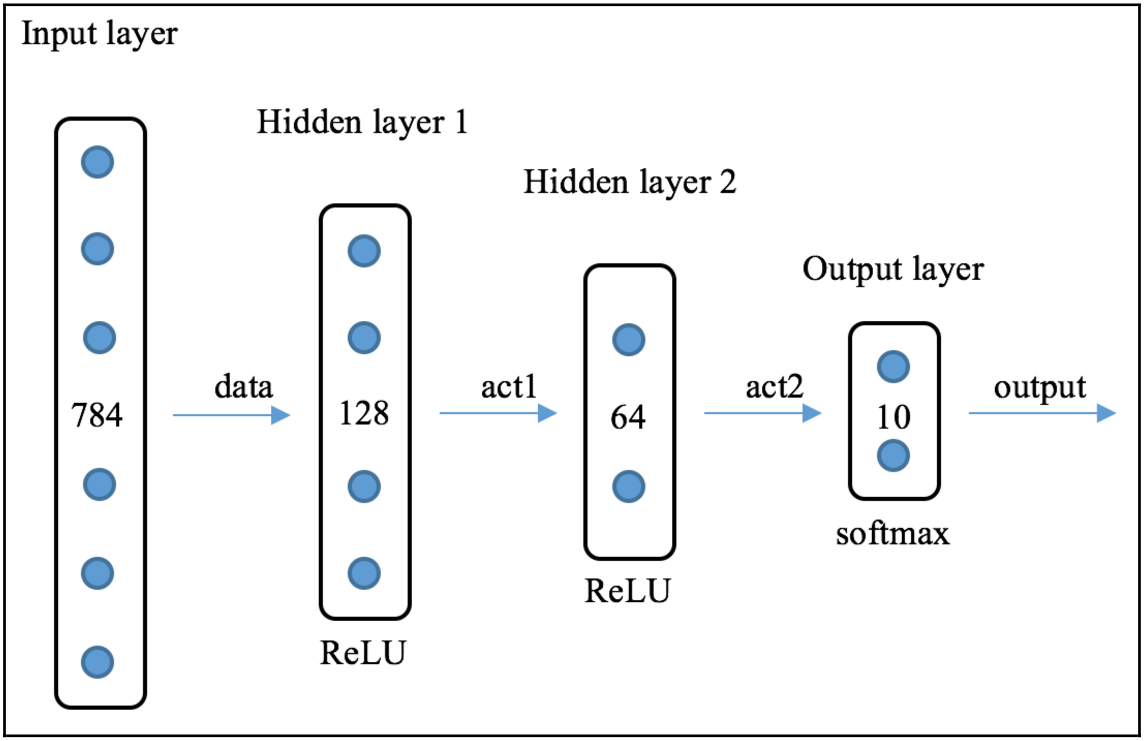
$$\triangle W^{(1)} = \frac{\partial J(W)}{\partial z_k^{(2)}} \frac{\partial z_k^{(2)}}{\partial W^{(1)}} = \delta^{(2)}(x)^T$$



$$relu(z) = z^{+} = max(0, z)$$

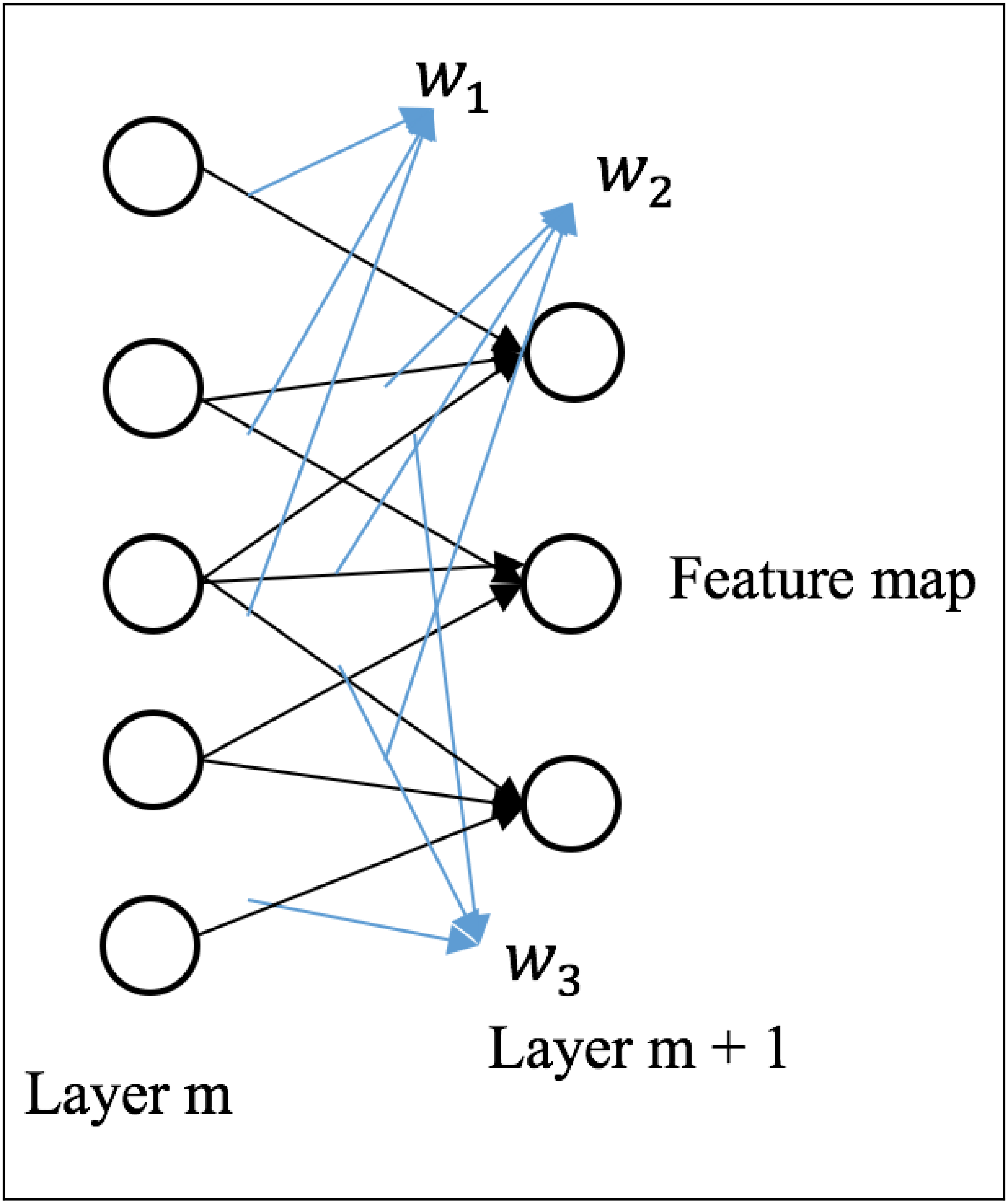


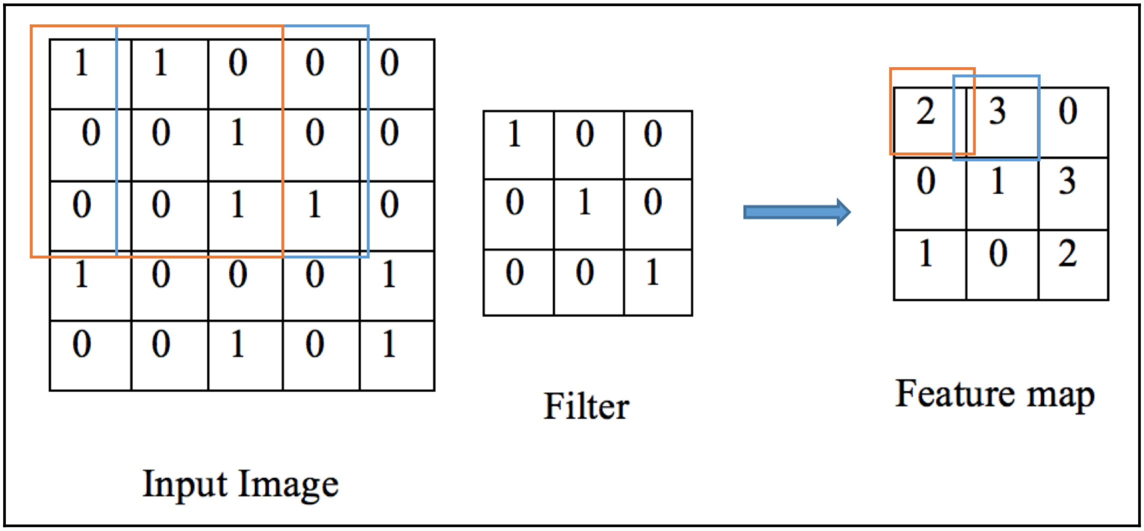
$$relu^{'}(z) = \begin{cases} 0, z < 0 \\ 1, z \geq 0 \end{cases}$$



$$\nu = \gamma\nu - \eta\triangle W$$

$$W = W + \nu$$


w_1
w_2
Feature map
w_3
Layer m + 1
Layer m



1	1	0	0	0
0	0	1	0	0
0	0	1	1	0
1	0	0	0	1
0	0	1	0	1

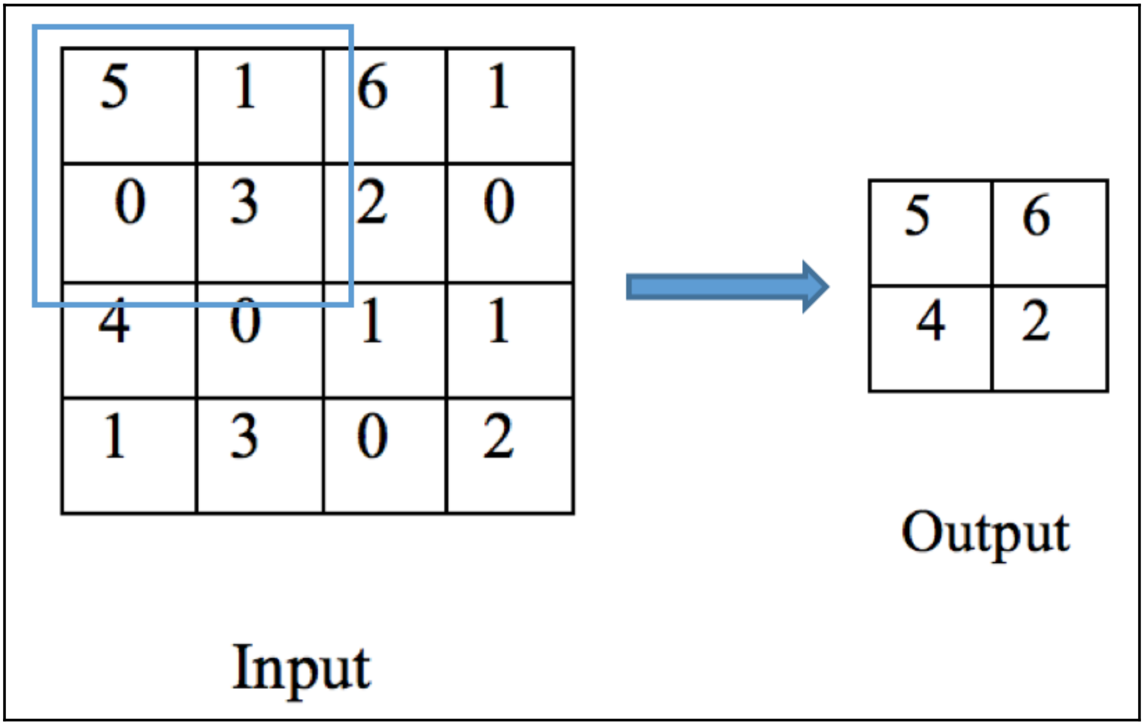
Input Image

1	0	0
0	1	0
0	0	1

Filter

2	3	0
0	1	3
1	0	2

Feature map

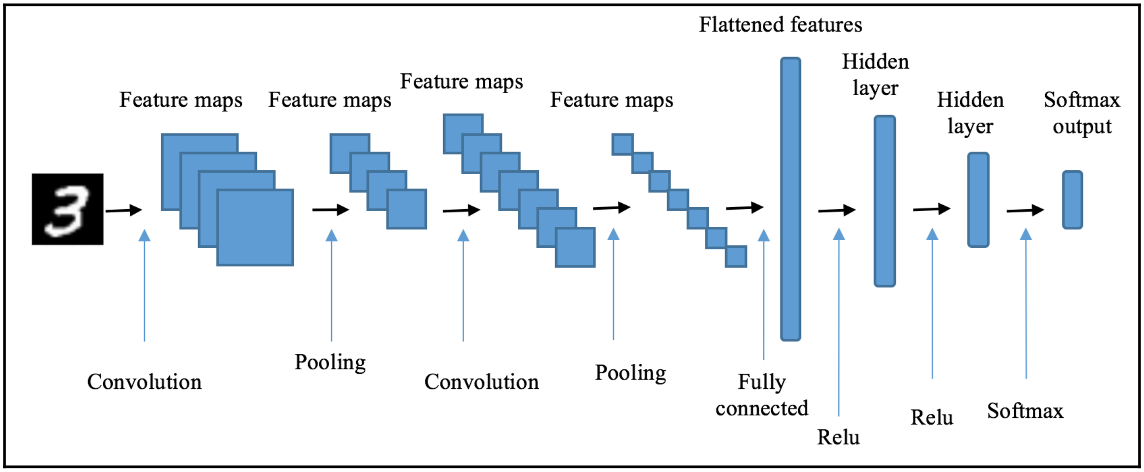


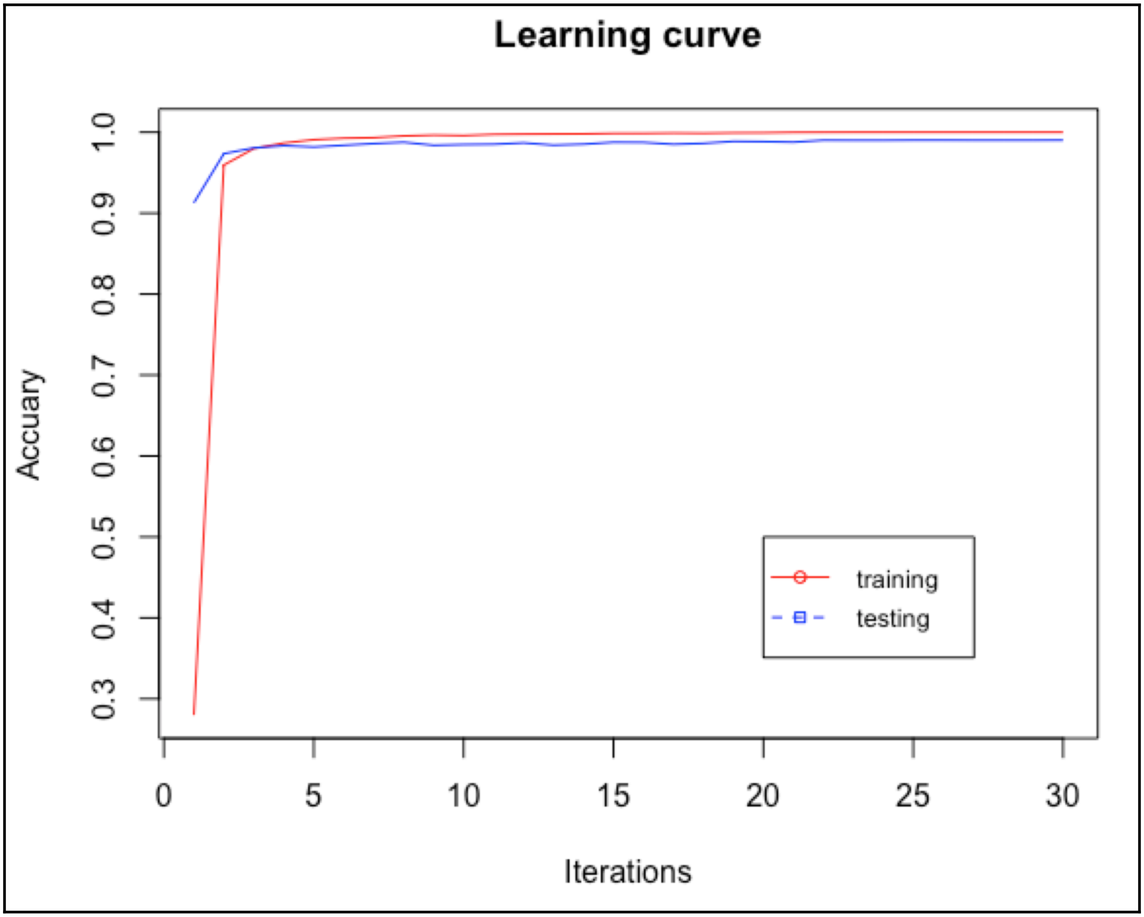
5	1	6	1
0	3	2	0
4	0	1	1
1	3	0	2

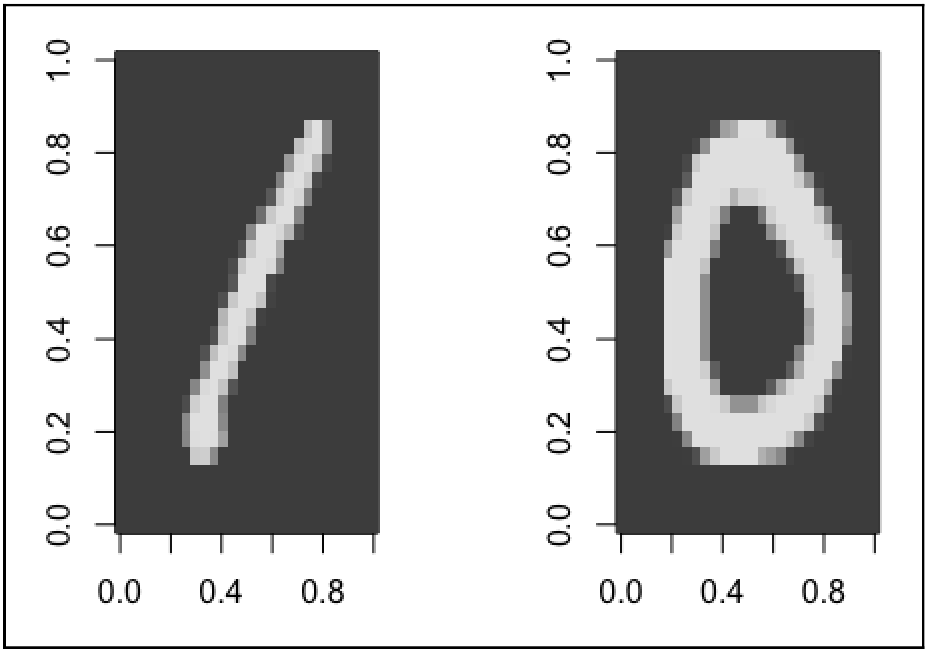
Input

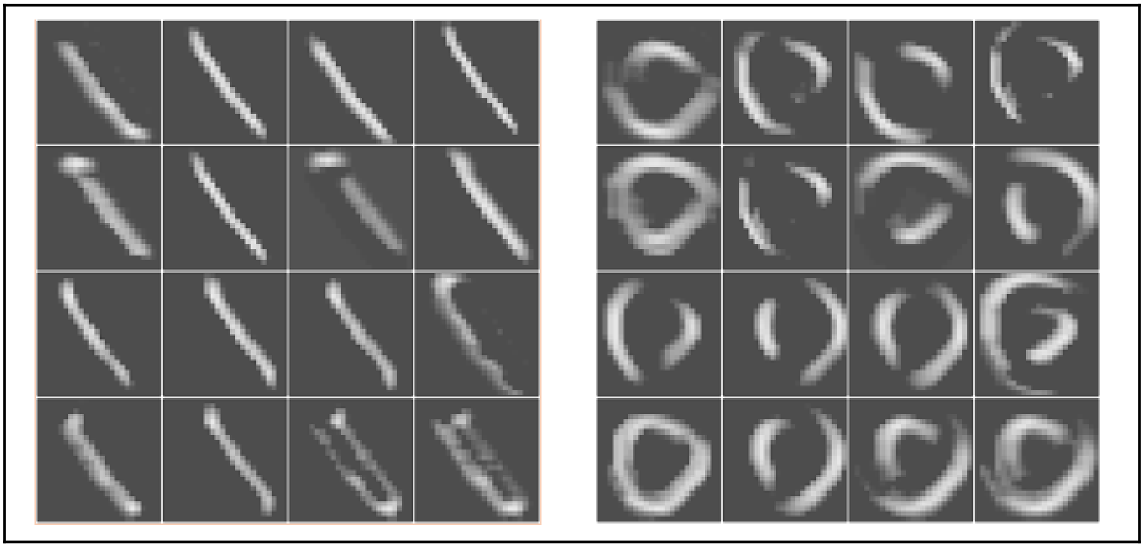
5	6
4	2

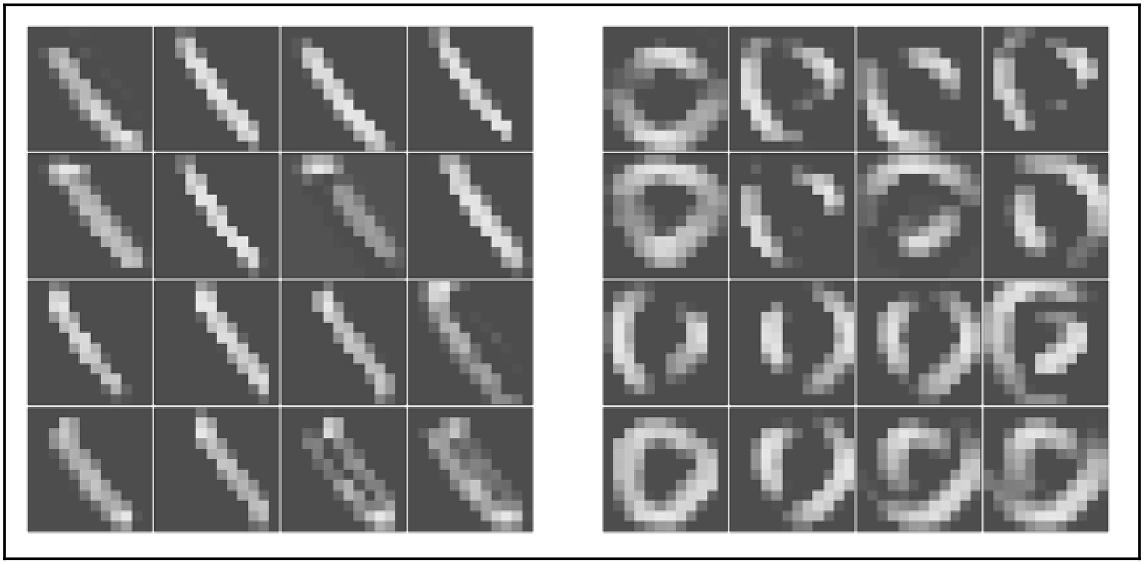
Output

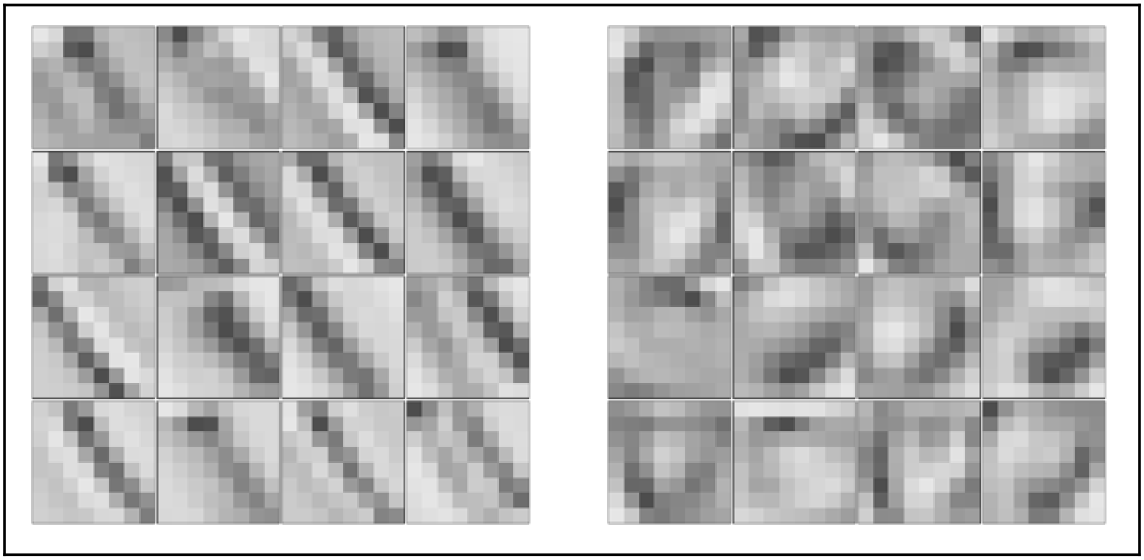

Flattened features
Feature maps
Feature maps
Feature maps
Feature maps
Hidden layer
Hidden layer
Softmax output
Convolution
Pooling
Convolution
Pooling
Fully connected
Relu
Relu
Softmax


Learning curve
Accuary
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0
5
10
15
20
25
30
Iterations
training
testing

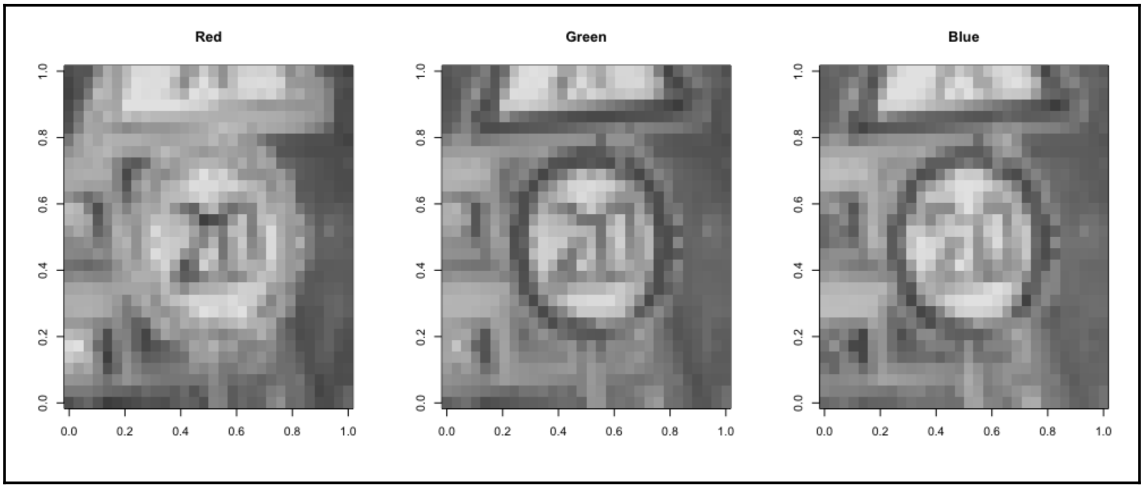

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.4
0.8
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.4
0.8

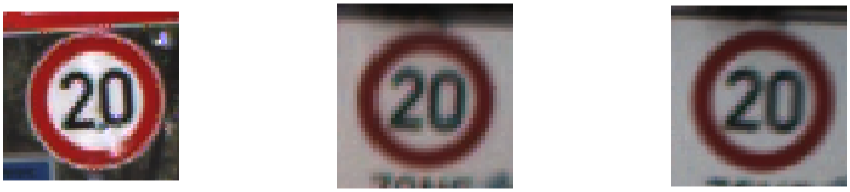




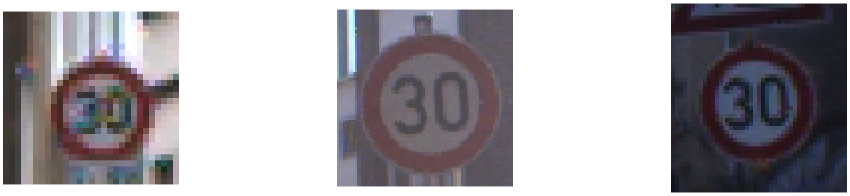


Chapter 13: Traffic Signs Recognition for Intelligent Vehicles

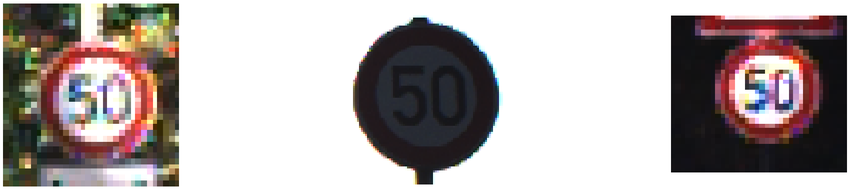




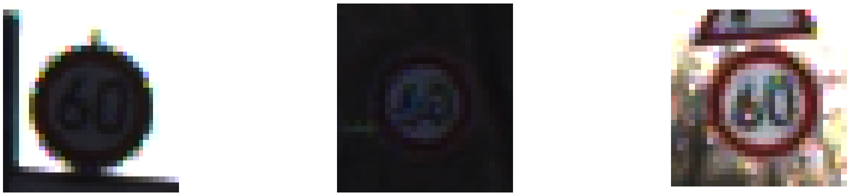
Class 0 : Speed limit (20km/h)



Class 1 : Speed limit (30km/h)

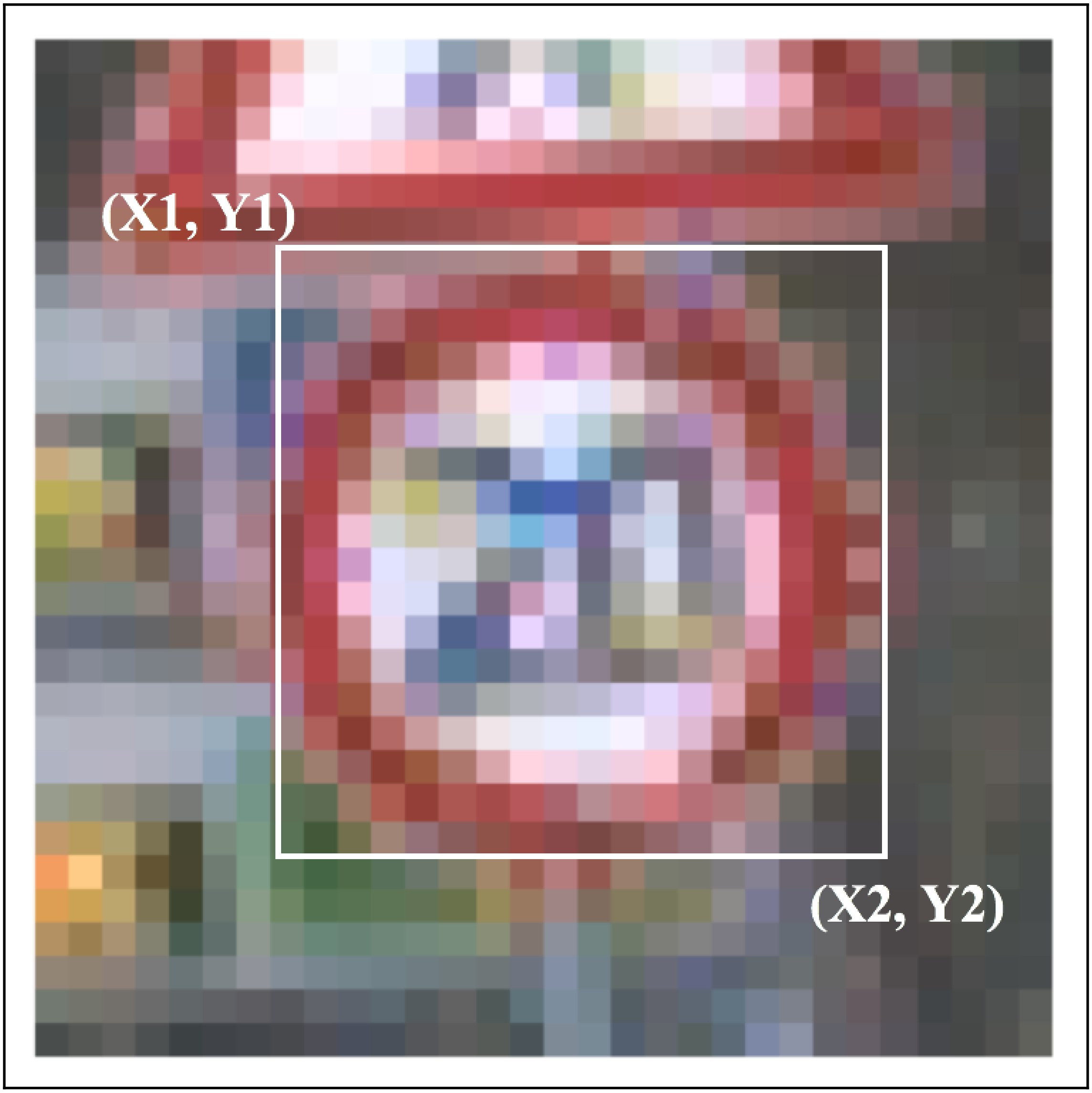


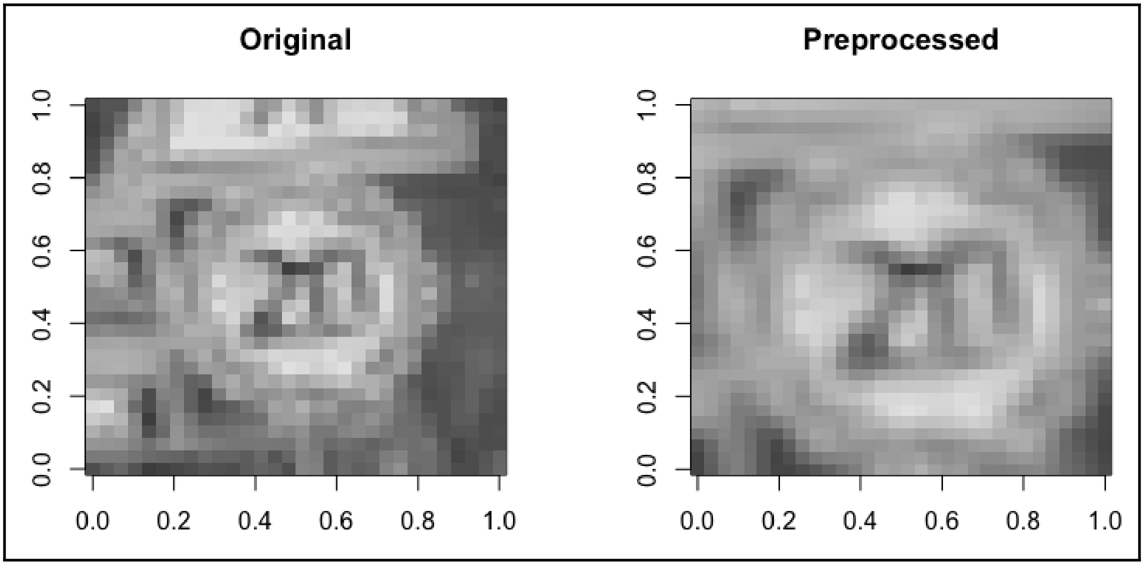
Class 2 : Speed limit (50km/h)

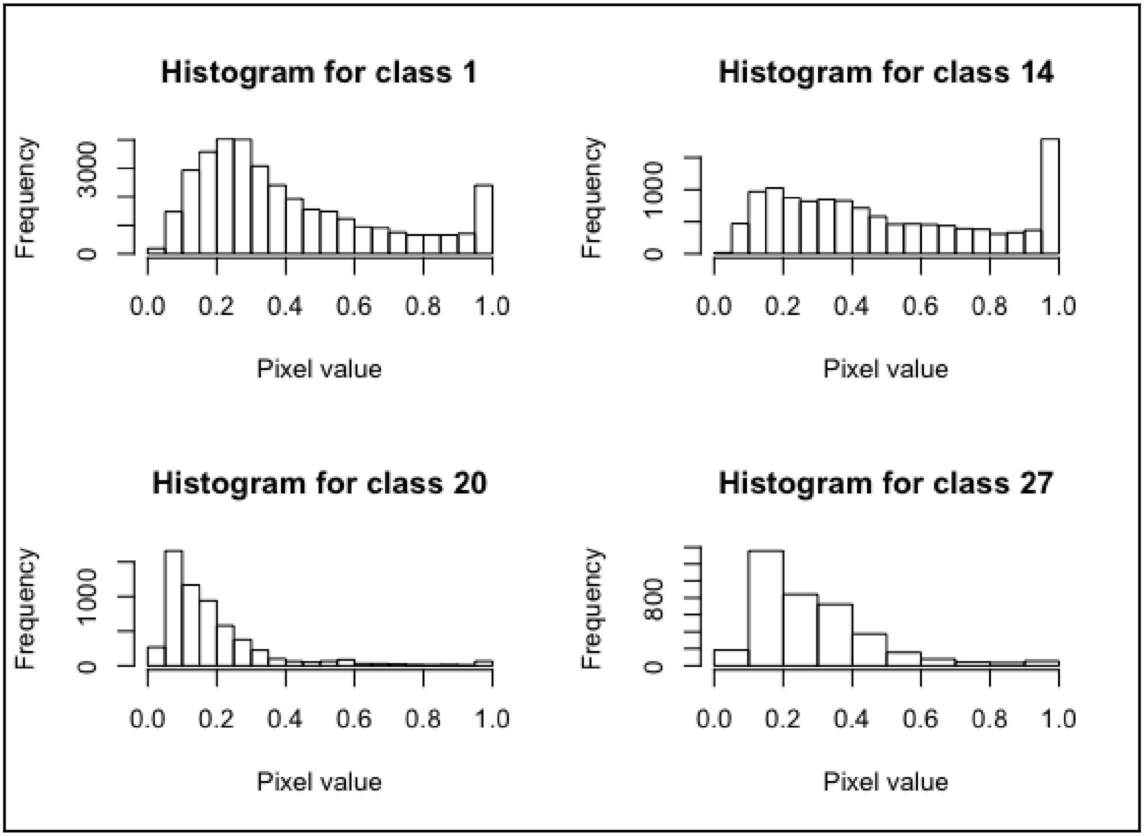


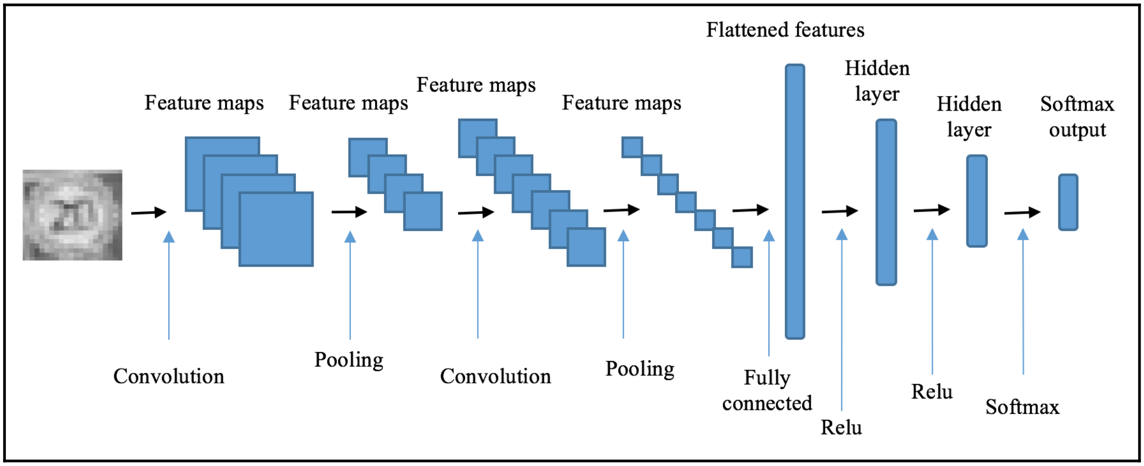
Class 3 : Speed limit (60km/h)

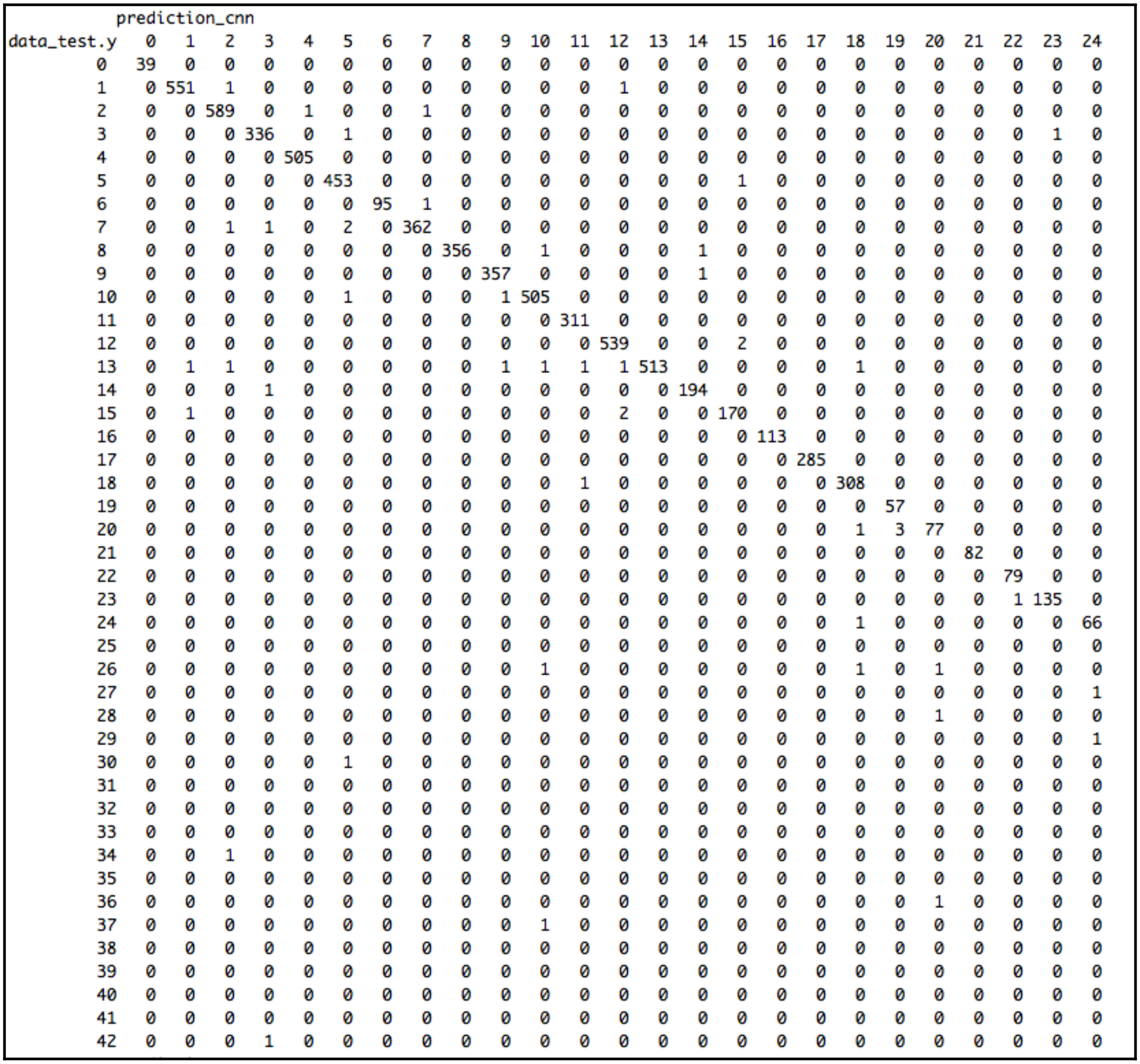
Class 4 : Speed limit (70km/h)


(X1, Y1)
(X2, Y2)

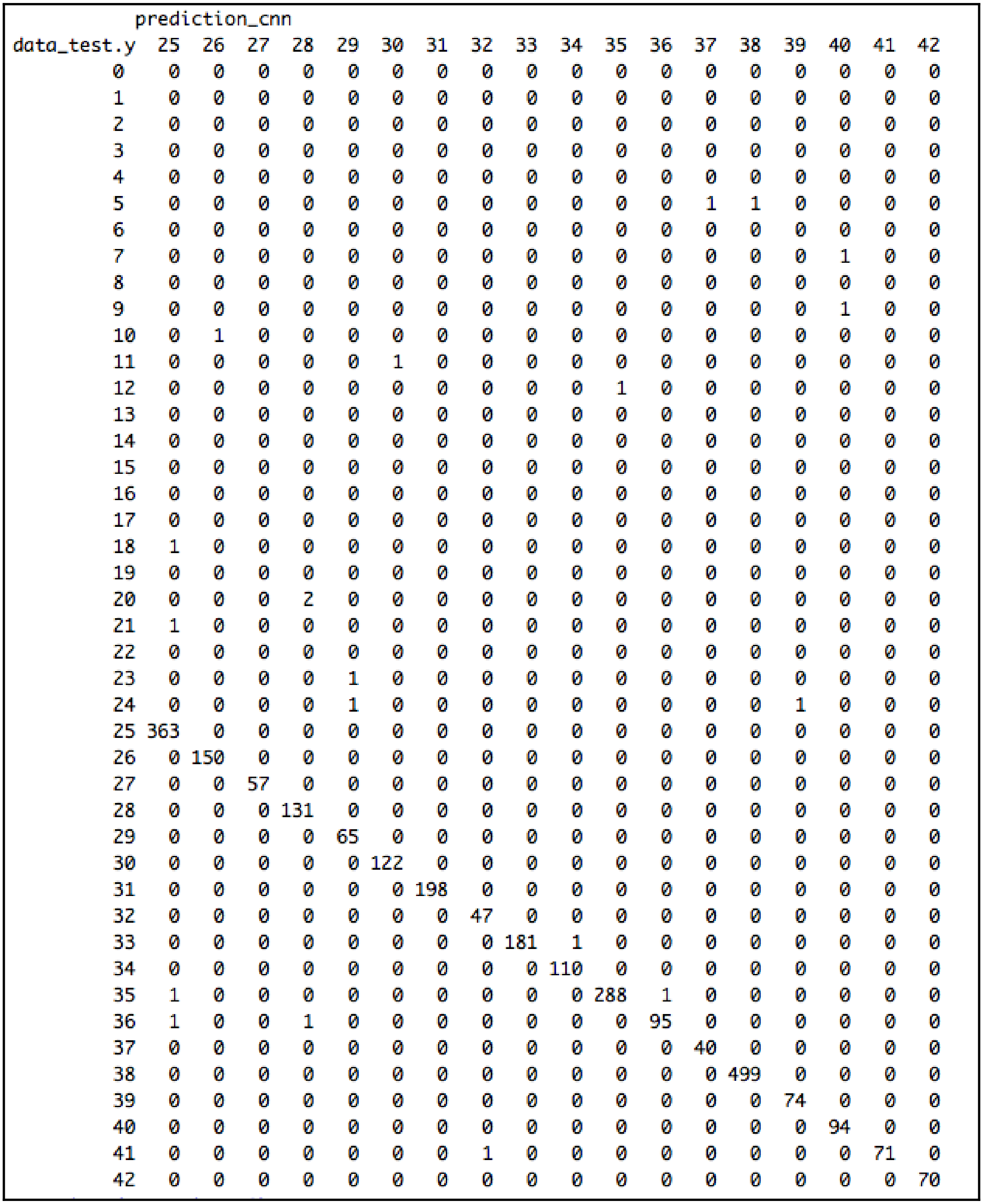

Original
Preprocessed
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Histogram for class 1
Frequency
0
3000
0.0
0.2
0.4
0.6
0.8
1.0
Pixel value
Histogram for class 14
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Pixel value
Histogram for class 20
Frequency
0
1000
0.0
0.2
0.4
0.6
0.8
1.0
Pixel value
Histogram for class 27
Frequency
0
800
0.0
0.2
0.4
0.6
0.8
1.0
Pixel value

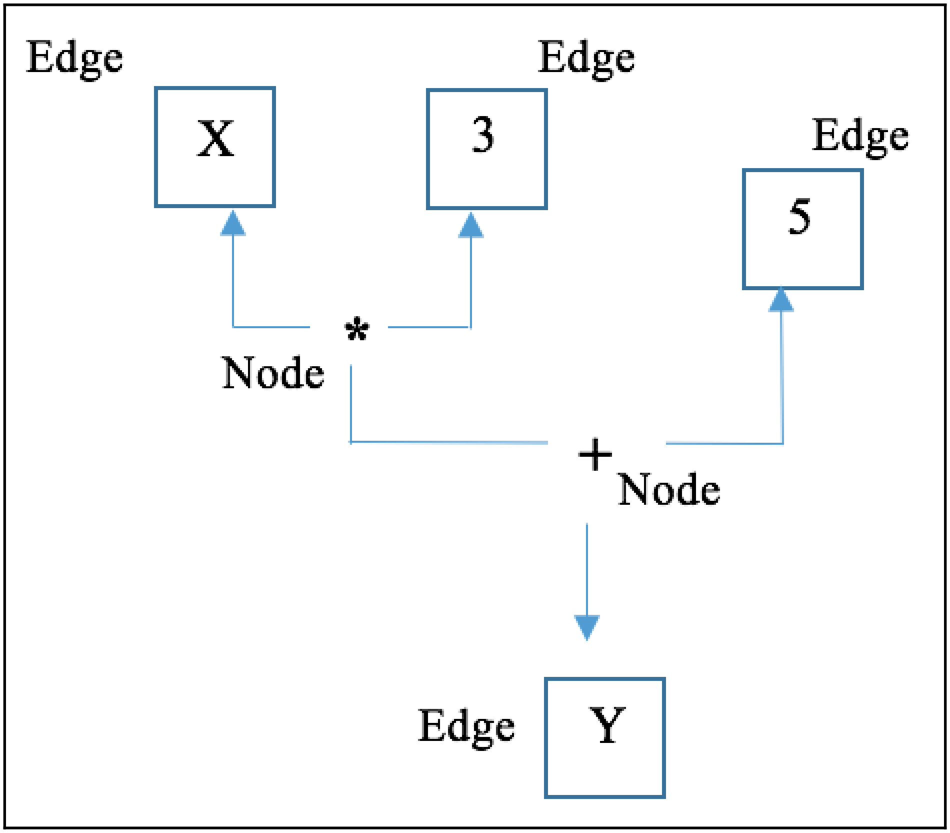

Flattened features
Feature maps
Feature maps
Feature maps
Feature maps
Hidden layer
Hidden layer
Softmax output
Convolution
Pooling
Convolution
Pooling
Fully connected
Relu
Relu
Softmax

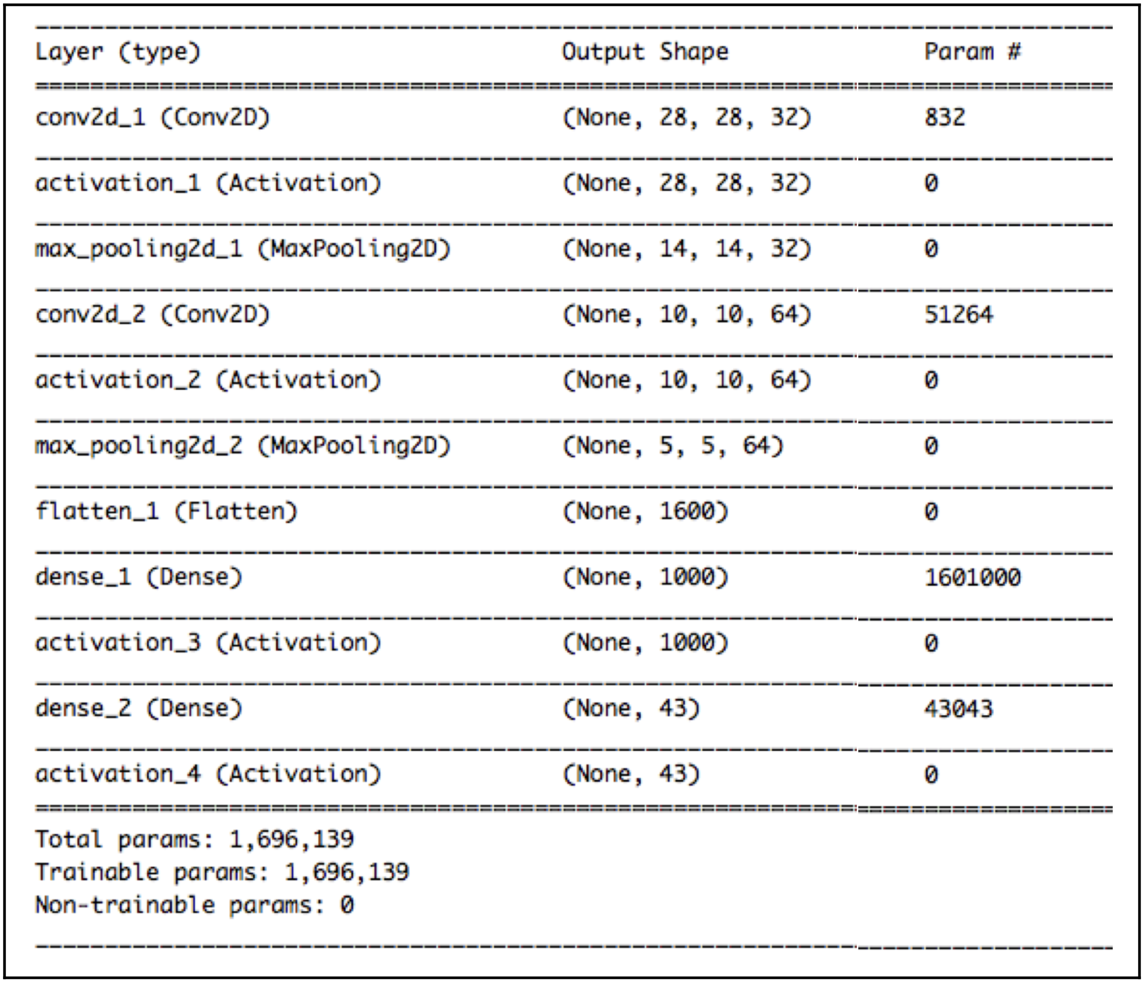


```
           prediction_cnn
data_test.y   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          0  39   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          1   0 551   1   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
          2   0   0 589   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          3   0   0   0 336   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
          4   0   0   0   0 505   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          5   0   0   0   0   0 453   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
          6   0   0   0   0   0   0  95   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          7   0   0   1   1   0   2   0 362   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          8   0   0   0   0   0   0   0   0 356   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0
          9   0   0   0   0   0   0   0   0   0 357   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
         10   0   0   0   0   0   1   0   0   0   1 505   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         11   0   0   0   0   0   0   0   0   0   0   0 311   0   0   0   0   0   0   0   0   0   0   0   0   0
         12   0   0   0   0   0   0   0   0   0   0   0   0 539   0   0   2   0   0   0   0   0   0   0   0   0
         13   0   1   1   0   0   0   0   0   0   1   1   1   1 513   0   0   0   0   1   0   0   0   0   0   0
         14   0   0   0   1   0   0   0   0   0   0   0   0   0   0 194   0   0   0   0   0   0   0   0   0   0
         15   0   1   0   0   0   0   0   0   0   0   0   0   2   0   0 170   0   0   0   0   0   0   0   0   0
         16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 113   0   0   0   0   0   0   0   0
         17   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 285   0   0   0   0   0   0   0
         18   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0 308   0   0   0   0   0   0
         19   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  57   0   0   0   0   0
         20   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   3  77   0   0   0   0
         21   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  82   0   0   0
         22   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  79   0   0
         23   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1 135   0
         24   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0  66
         25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         26   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   1   0   0   0   0
         27   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
         28   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
         29   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
         30   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         31   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         32   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         34   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         35   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         36   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
         37   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         38   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         39   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         40   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         41   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         42   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

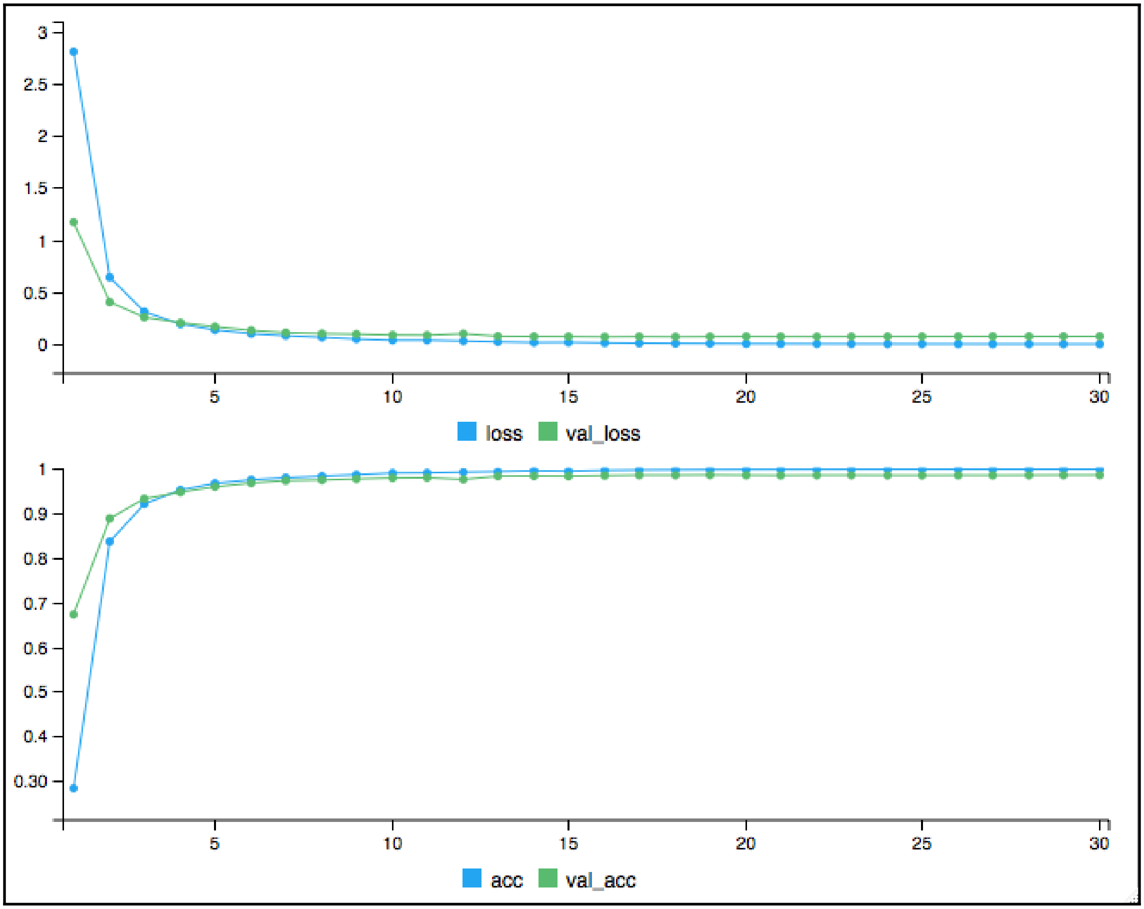


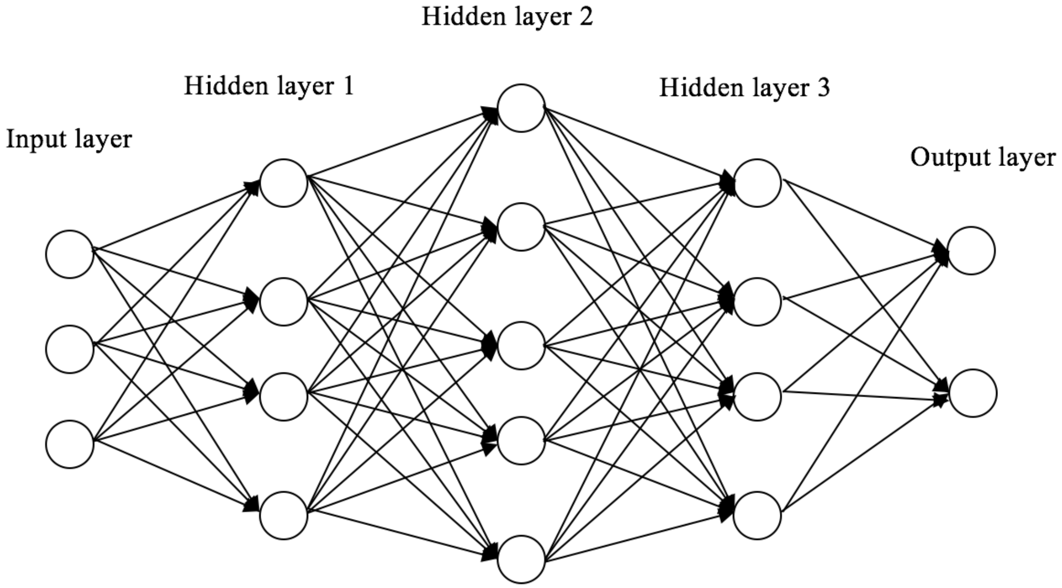
```
           prediction_cnn
data_test.y  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42
          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          5   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0
          6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
          8   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          9   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
         10   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         11   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
         12   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
         13   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         14   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         15   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         17   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         18   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         19   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         20   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         21   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         22   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         23   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
         24   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0
         25 363   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         26   0 150   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         27   0   0  57   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         28   0   0   0 131   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         29   0   0   0   0  65   0   0   0   0   0   0   0   0   0   0   0   0   0
         30   0   0   0   0   0 122   0   0   0   0   0   0   0   0   0   0   0   0
         31   0   0   0   0   0   0 198   0   0   0   0   0   0   0   0   0   0   0
         32   0   0   0   0   0   0   0  47   0   0   0   0   0   0   0   0   0   0
         33   0   0   0   0   0   0   0   0 181   1   0   0   0   0   0   0   0   0
         34   0   0   0   0   0   0   0   0   0 110   0   0   0   0   0   0   0   0
         35   1   0   0   0   0   0   0   0   0   0 288   1   0   0   0   0   0   0
         36   1   0   0   1   0   0   0   0   0   0   0  95   0   0   0   0   0   0
         37   0   0   0   0   0   0   0   0   0   0   0   0  40   0   0   0   0   0
         38   0   0   0   0   0   0   0   0   0   0   0   0   0 499   0   0   0   0
         39   0   0   0   0   0   0   0   0   0   0   0   0   0   0  74   0   0   0
         40   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  94   0   0
         41   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0  71   0
         42   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  70
```


Edge
Edge
Edge
X
3
5
*
Node
+
Node
Edge
Y

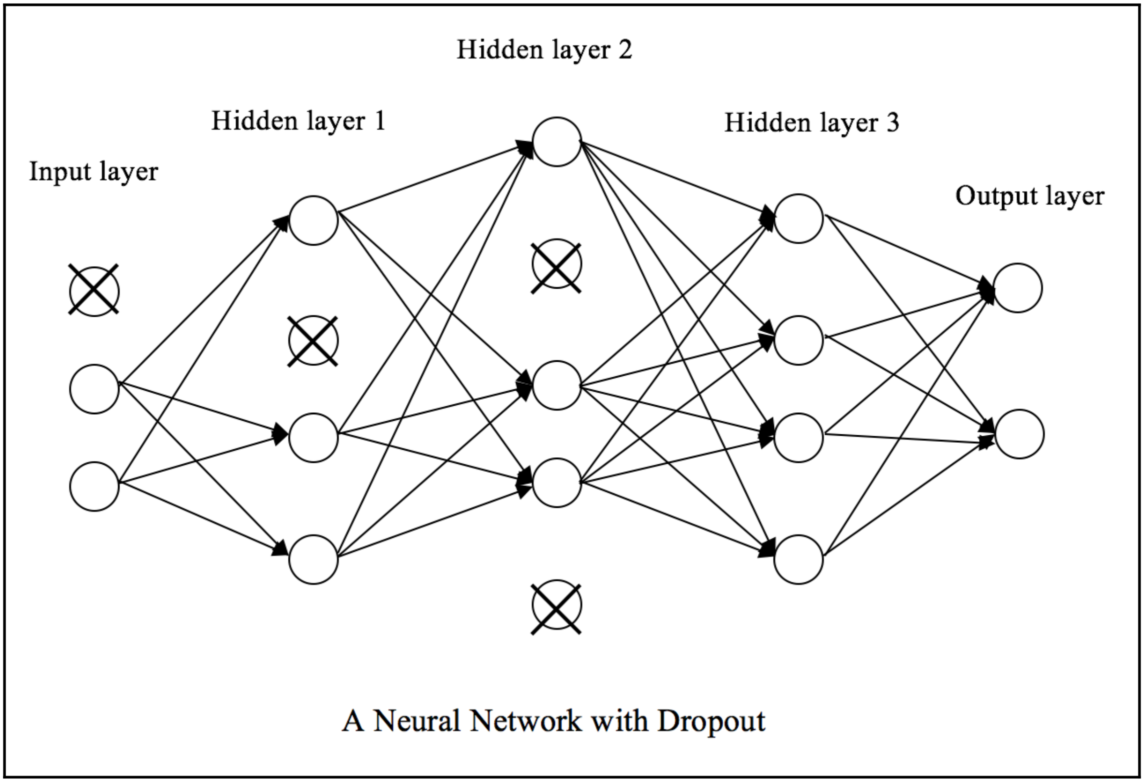


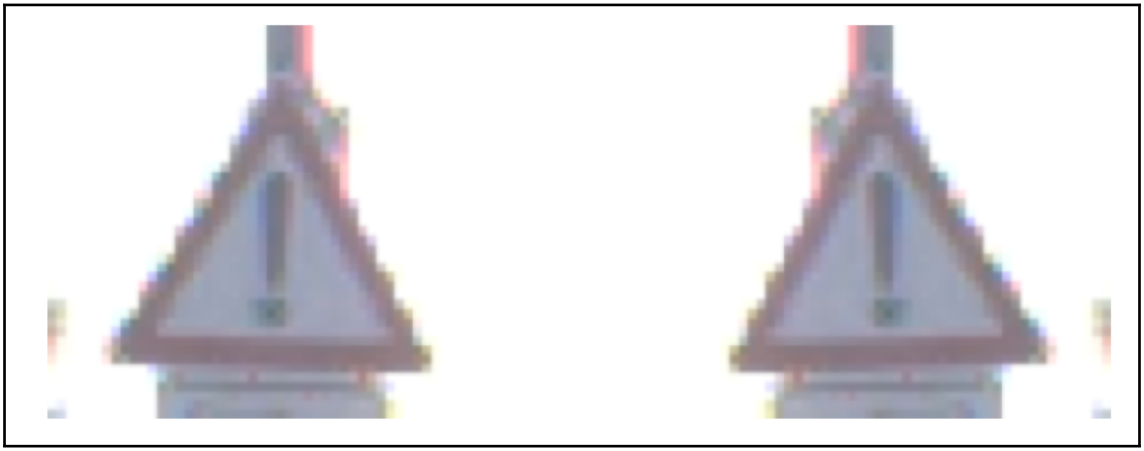
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 32)        832
_________________________________________________________________
activation_1 (Activation)    (None, 28, 28, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 14, 14, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 10, 10, 64)        51264
_________________________________________________________________
activation_2 (Activation)    (None, 10, 10, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2D) (None, 5, 5, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1000)              1601000
_________________________________________________________________
activation_3 (Activation)    (None, 1000)              0
_________________________________________________________________
dense_2 (Dense)              (None, 43)                43043
_________________________________________________________________
activation_4 (Activation)    (None, 43)                0
=================================================================
Total params: 1,696,139
Trainable params: 1,696,139
Non-trainable params: 0
_________________________________________________________________
```

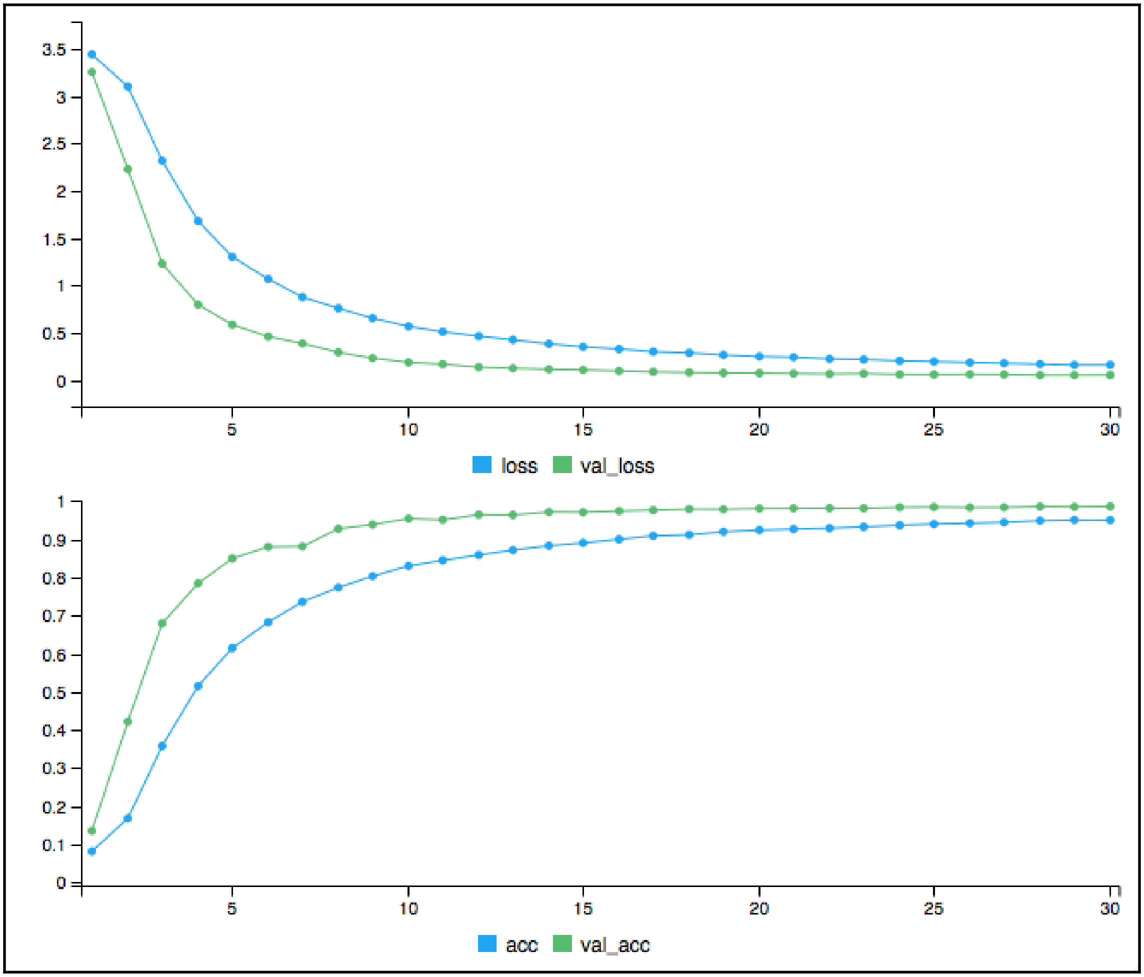

3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30
🟦 loss 🟩 val_loss
1
0.9
0.8
0.7
0.6
0.5
0.4
0.30
5
10
15
20
25
30
🟦 acc 🟩 val_acc



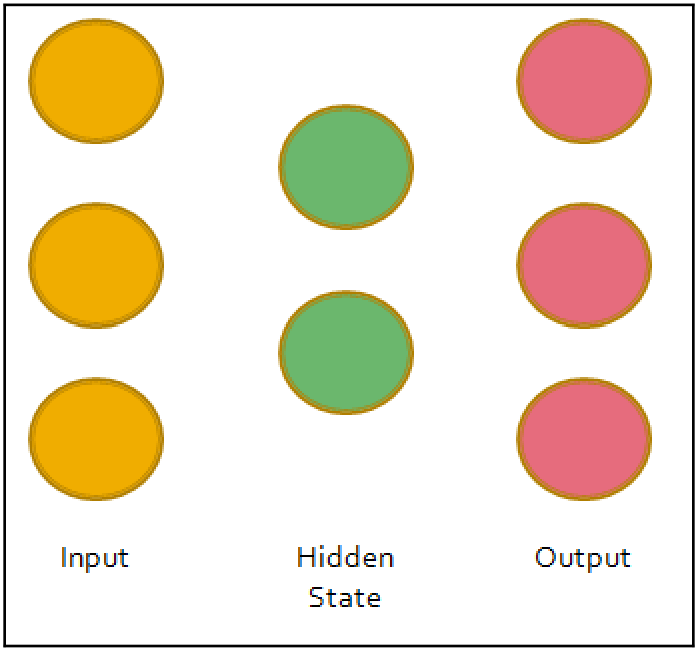
A Standard Neural Network

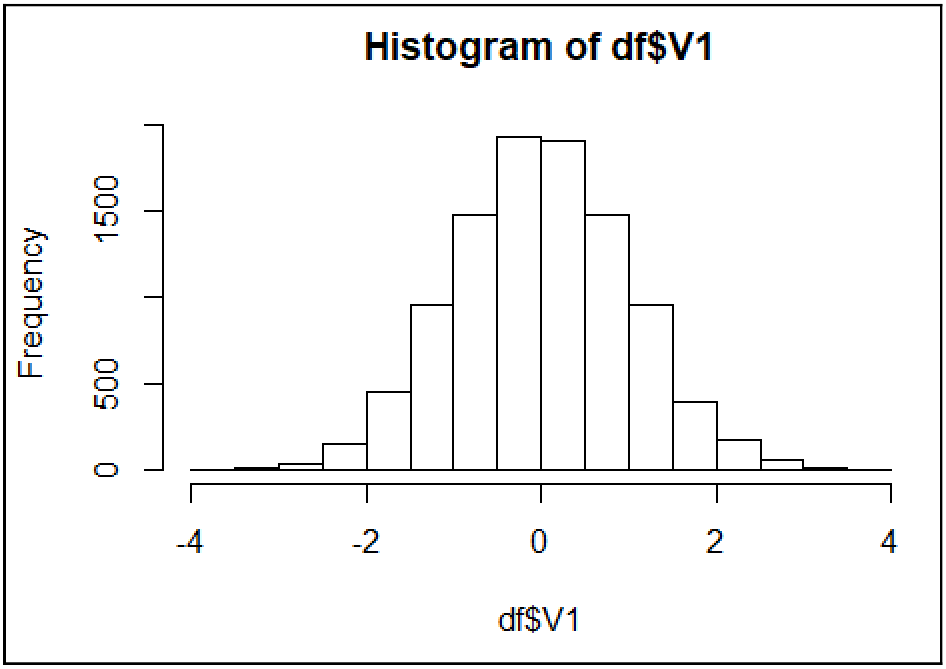

Hidden layer 2
Hidden layer 1
Hidden layer 3
Input layer
Output layer
A Neural Network with Dropout

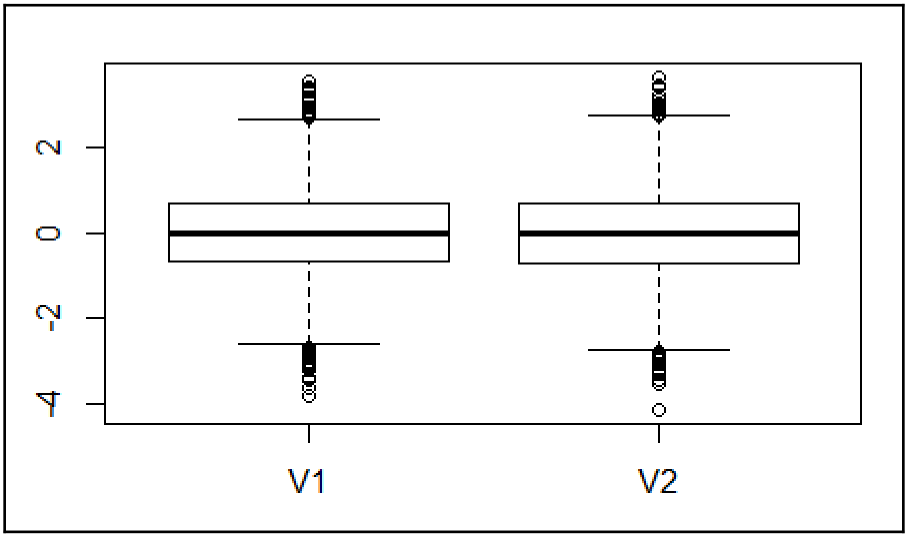


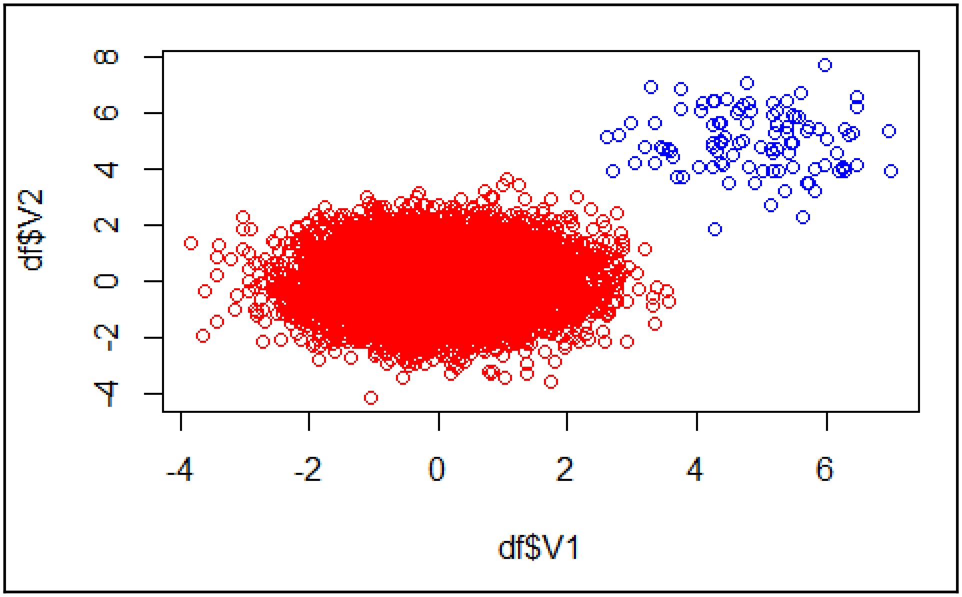

3.5
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30
🟦 loss 🟩 val_loss
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
🟦 acc 🟩 val_acc

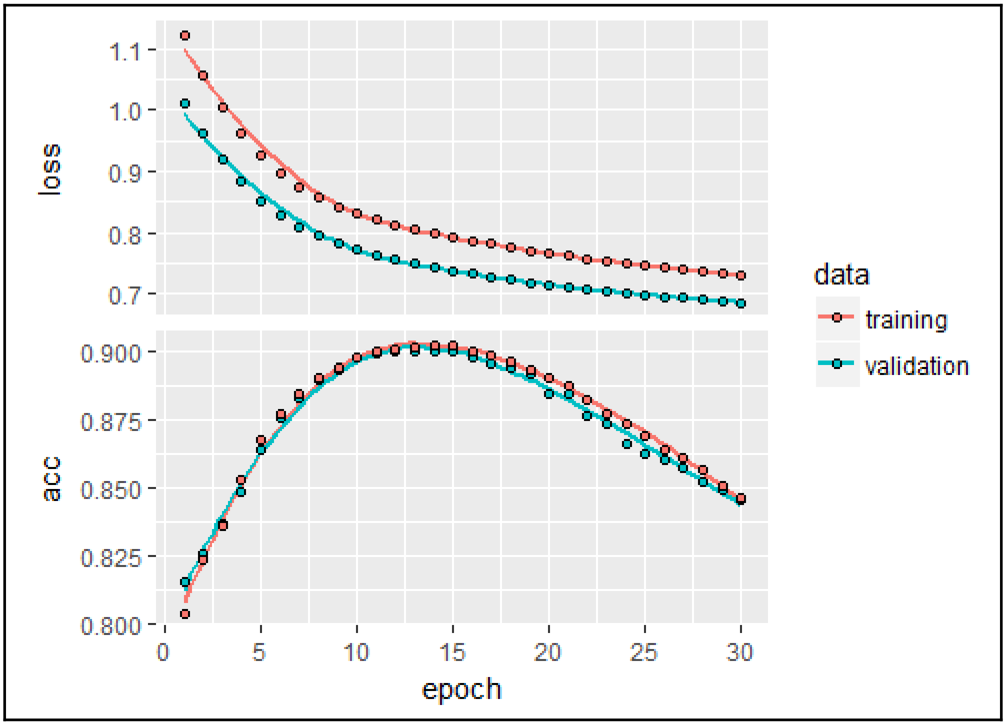
Chapter 14: Fraud Detection with Autoencoders

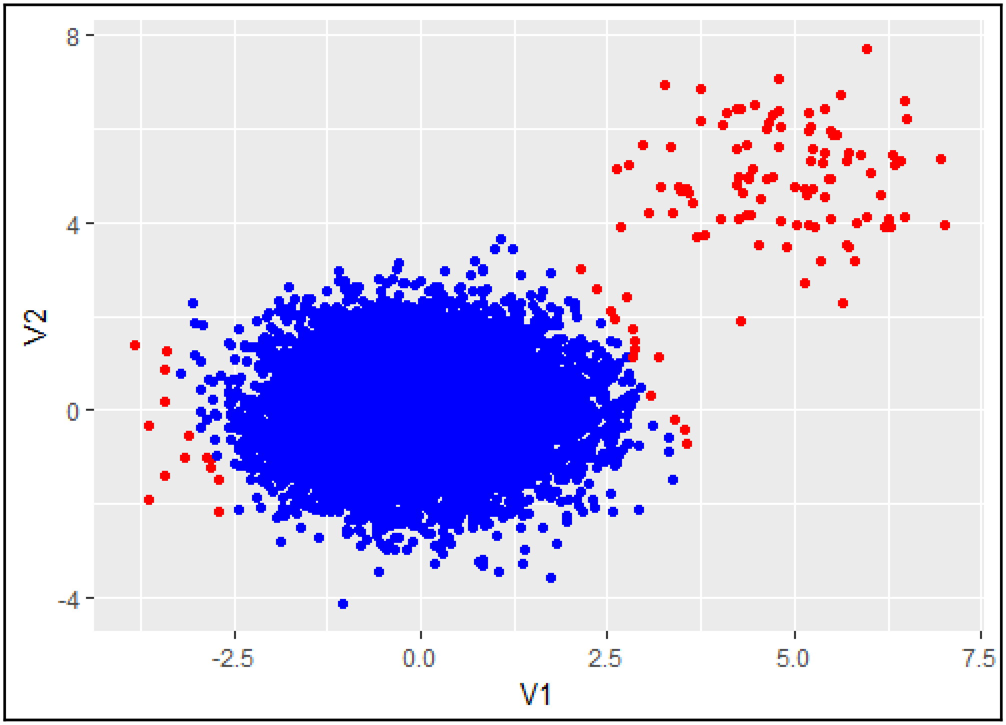


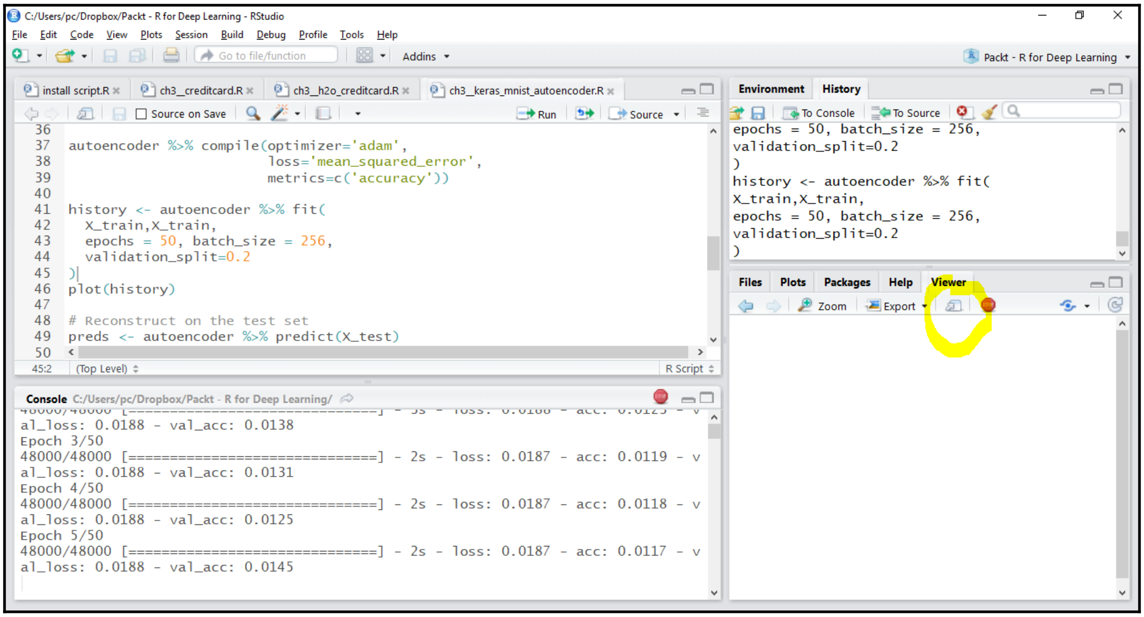

Histogram of df$V1
Frequency
0
500
1500
-4
-2
0
2
4
df$V1


2
0
-2
-4
V1
V2


df$V2
8
6
4
2
0
-2
-4
-4
-2
0
2
4
6
df$V1


loss
1.1
1.0
0.9
0.8
0.7
acc
0.900
0.875
0.850
0.825
0.800
0
5
10
15
20
25
30
epoch
data
training
validation

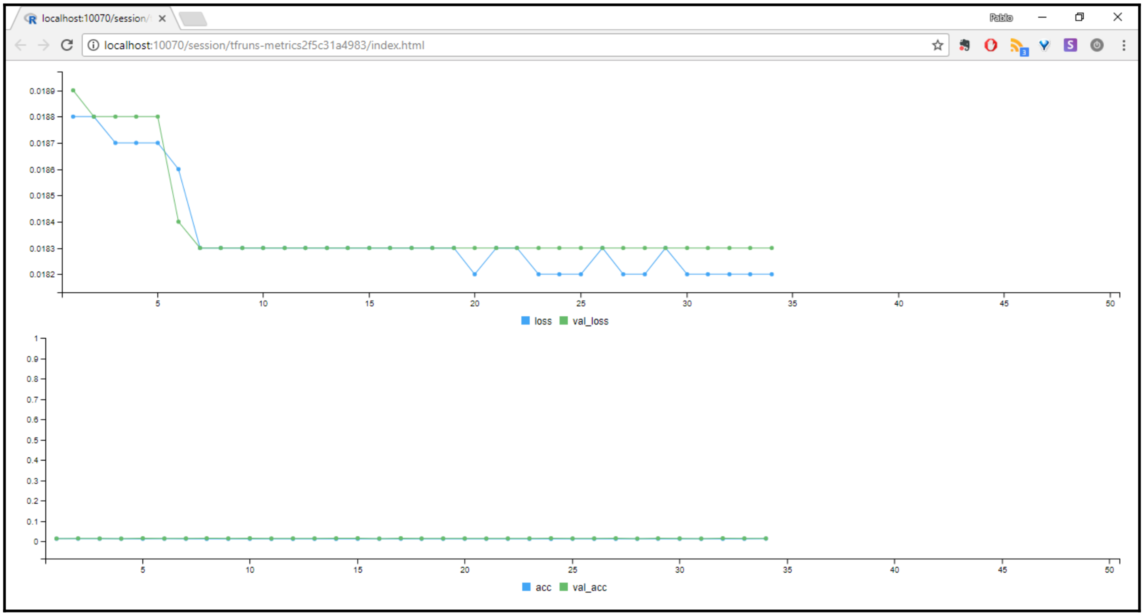


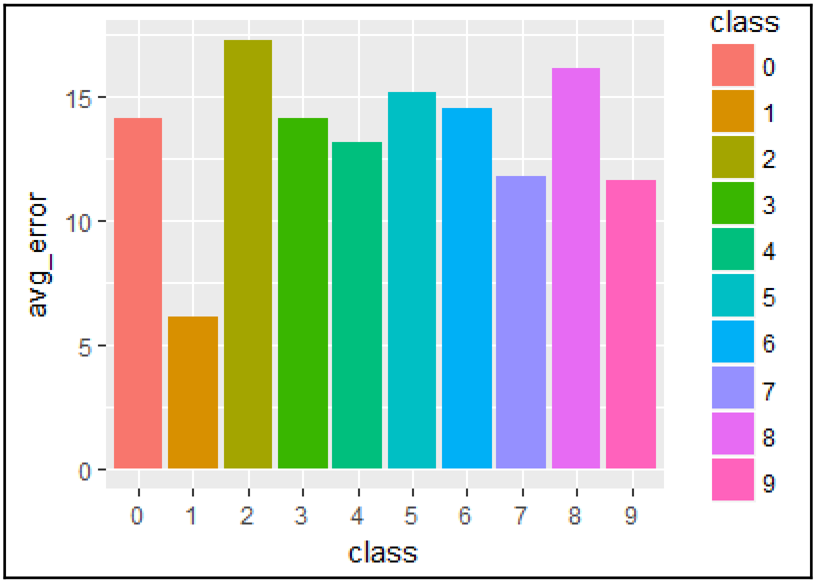

C:/Users/pc/Dropbox/Packt - R for Deep Learning - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Packt - R for Deep Learning
install script.R
ch3_creditcard.R
ch3_h2o_creditcard.R
ch3_keras_mnist_autoencoder.R
Source on Save
Run
Source
Environment
History
To Console
To Source
45:2
(Top Level)
R Script
Console C:/Users/pc/Dropbox/Packt - R for Deep Learning/
Files
Plots
Packages
Help
Viewer
Zoom
Export

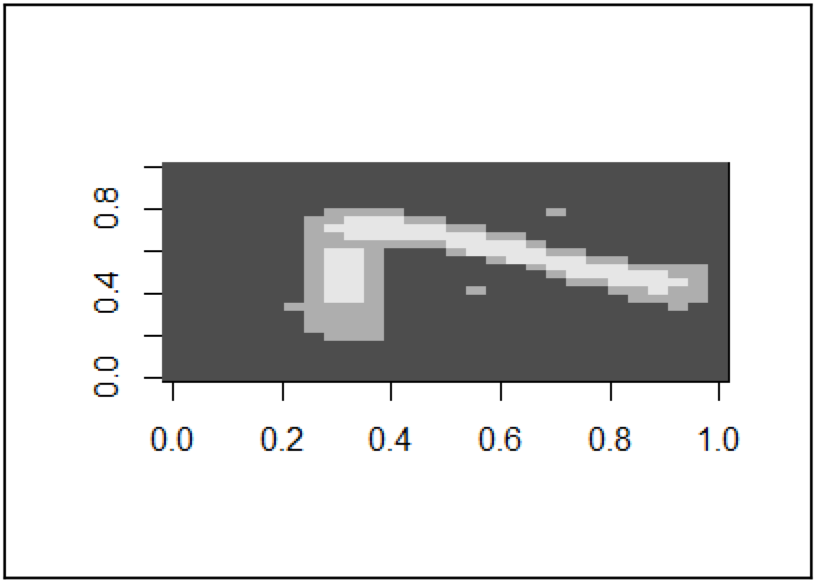
```
autoencoder %>% compile(optimizer='adam',
                        loss='mean_squared_error',
                        metrics=c('accuracy'))

history <- autoencoder %>% fit(
  X_train,X_train,
  epochs = 50, batch_size = 256,
  validation_split=0.2
)
plot(history)

# Reconstruct on the test set
preds <- autoencoder %>% predict(X_test)
```

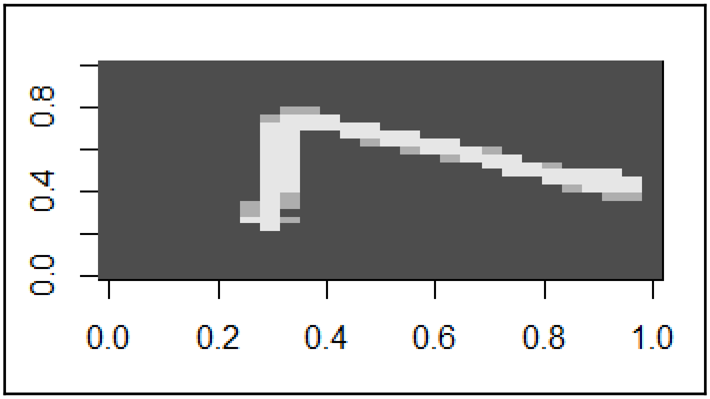
```
epochs = 50, batch_size = 256,
validation_split=0.2
)
history <- autoencoder %>% fit(
X_train,X_train,
epochs = 50, batch_size = 256,
validation_split=0.2
)
```

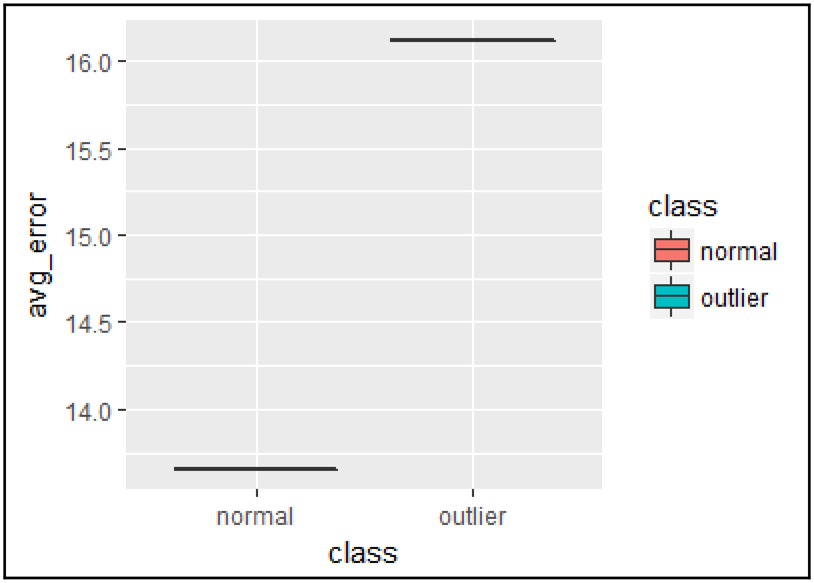
```
al_loss: 0.0188 - val_acc: 0.0138
Epoch 3/50
48000/48000 [==============================] - 2s - loss: 0.0187 - acc: 0.0119 - v
al_loss: 0.0188 - val_acc: 0.0131
Epoch 4/50
48000/48000 [==============================] - 2s - loss: 0.0187 - acc: 0.0118 - v
al_loss: 0.0188 - val_acc: 0.0125
Epoch 5/50
48000/48000 [==============================] - 2s - loss: 0.0187 - acc: 0.0117 - v
al_loss: 0.0188 - val_acc: 0.0145
```

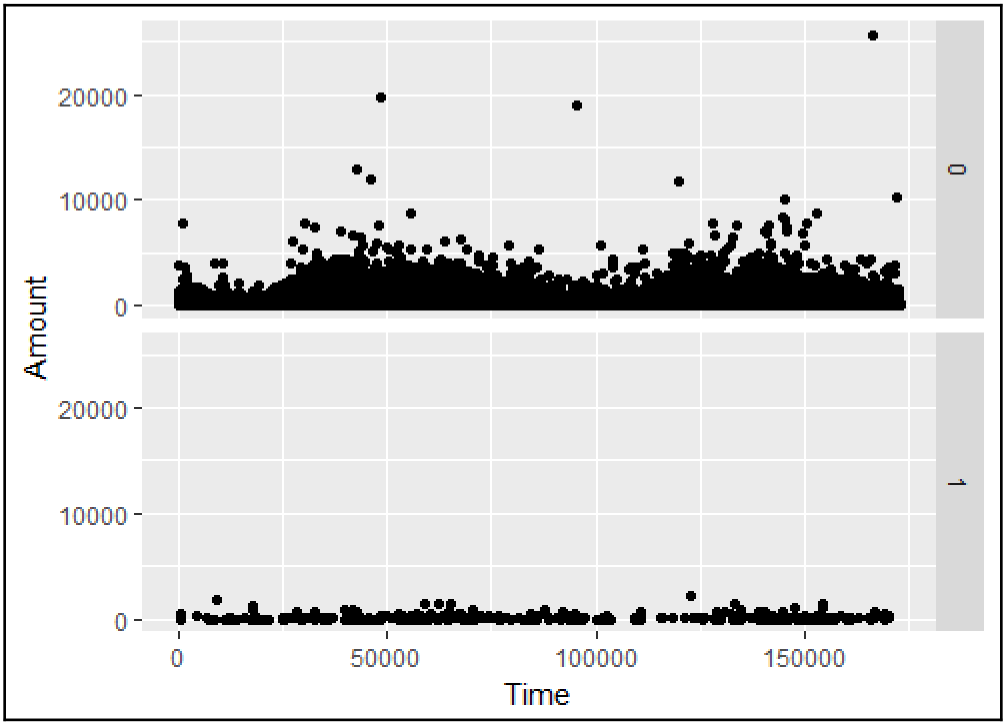

localhost:10070/session/
Pablo
localhost:10070/session/tfruns-metrics2f5c31a4983/index.html
0.0189
0.0188
0.0187
0.0186
0.0185
0.0184
0.0183
0.0182
5
10
15
20
25
30
35
40
45
50
loss
val_loss
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
35
40
45
50
acc
val_acc

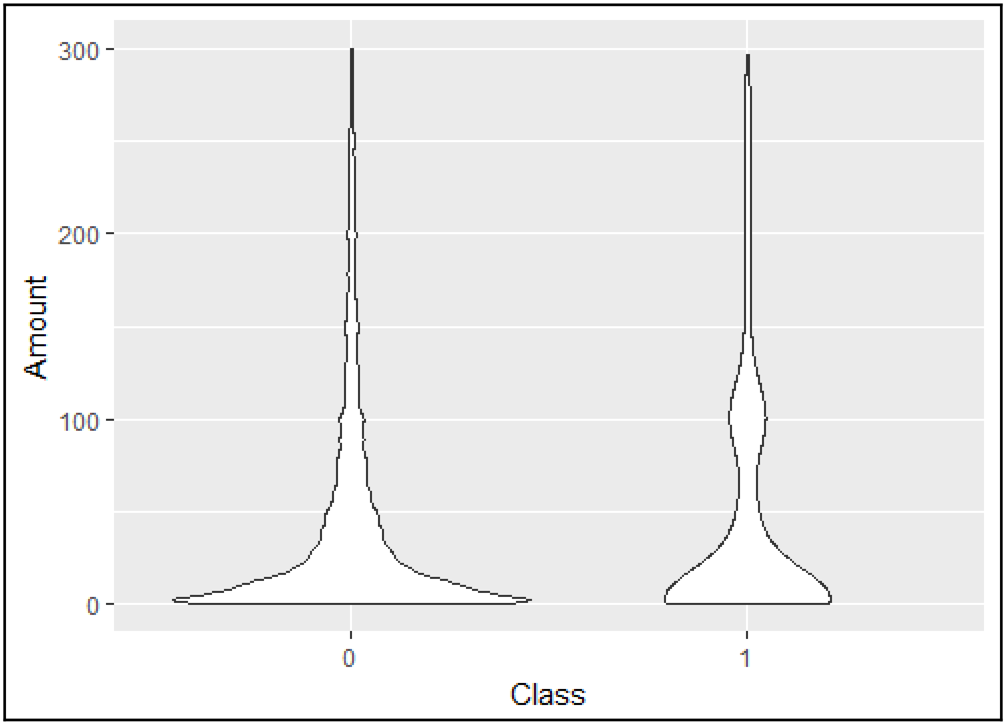

avg_error
15
10
5
0
0
1
2
3
4
5
6
7
8
9
class
class
0
1
2
3
4
5
6
7
8
9

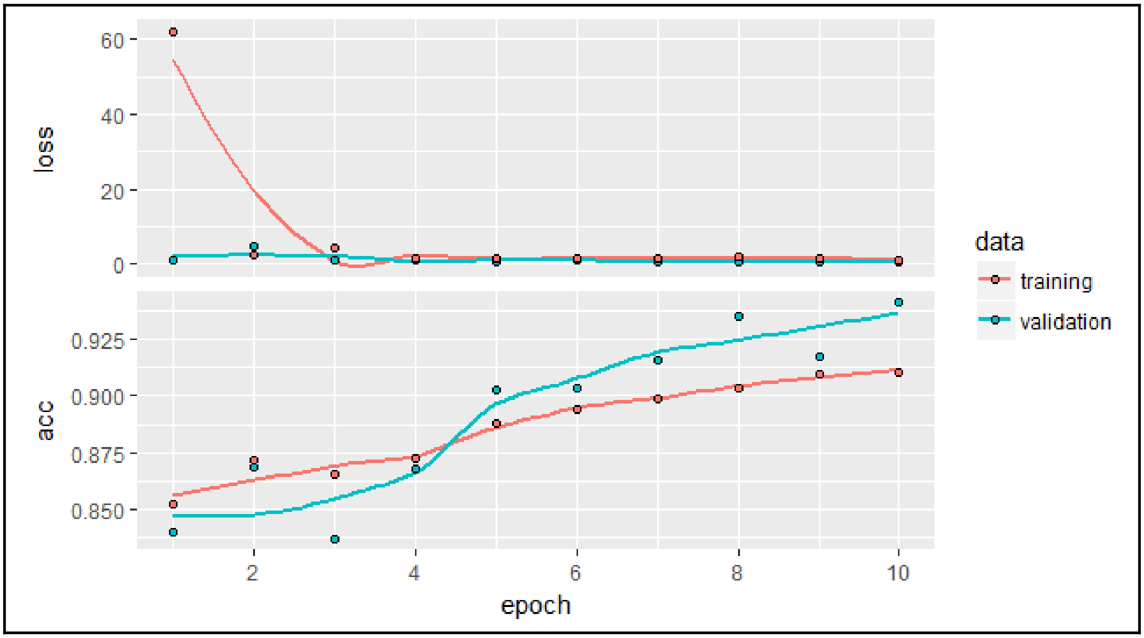

0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0

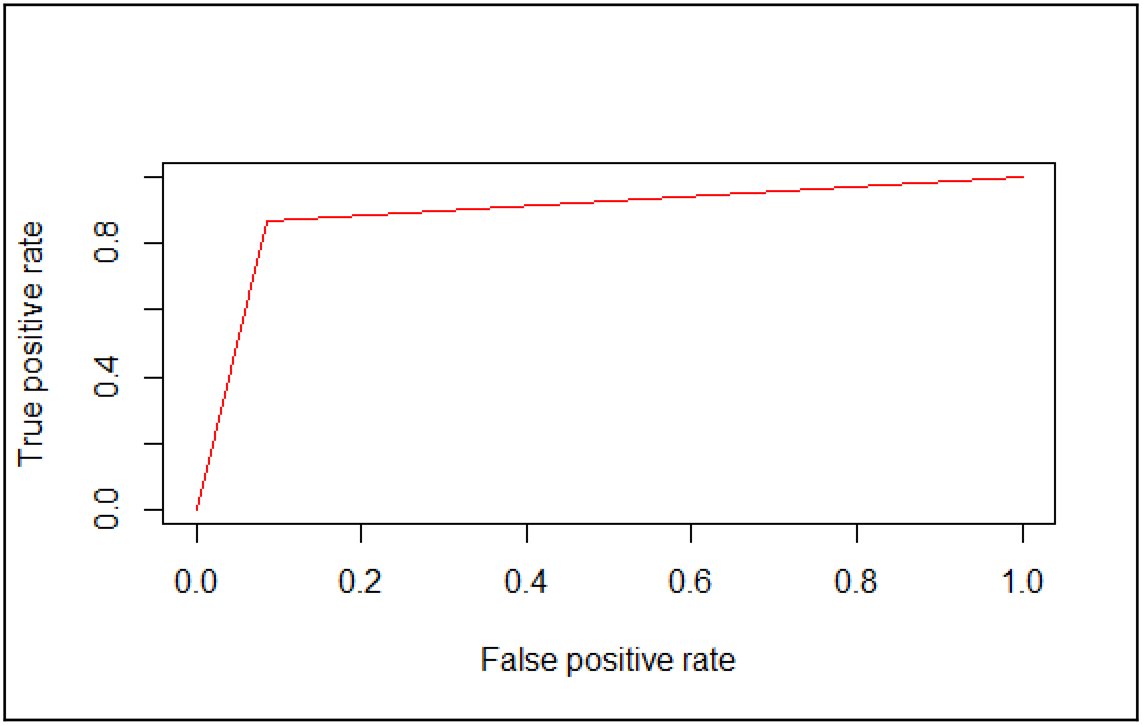

0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0

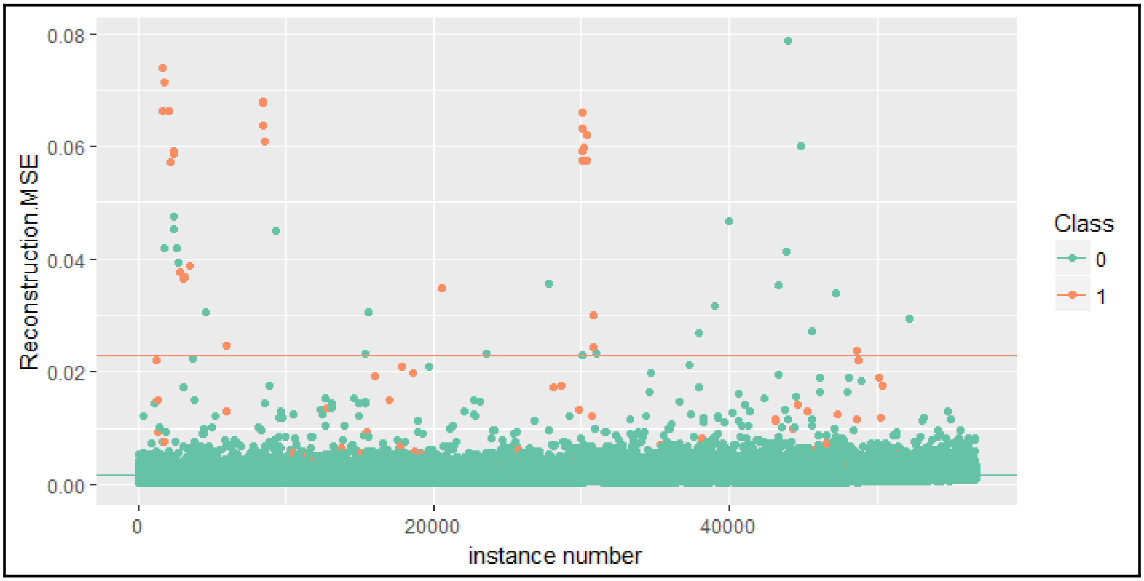

16.0
15.5
15.0
14.5
14.0
avg_error
normal
outlier
class
class
normal
outlier

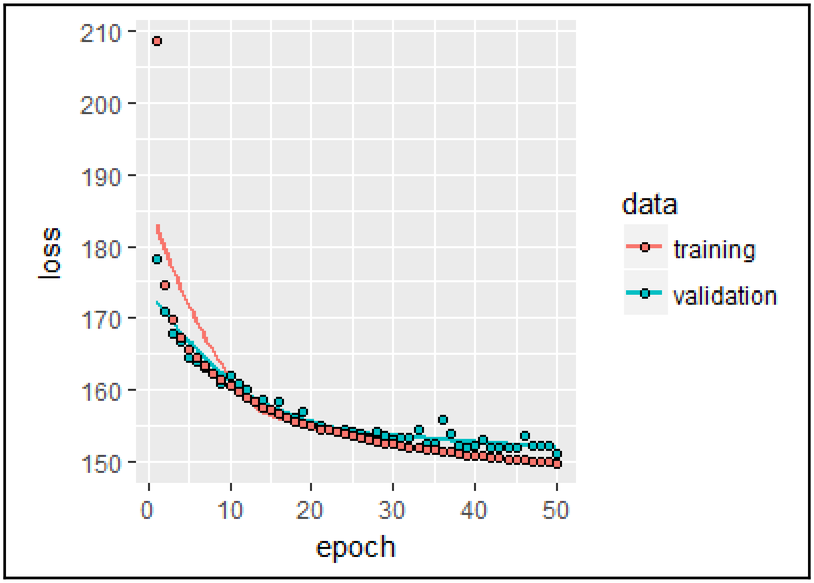

20000
10000
0
20000
10000
0
Amount
0
1
0
50000
100000
150000
Time

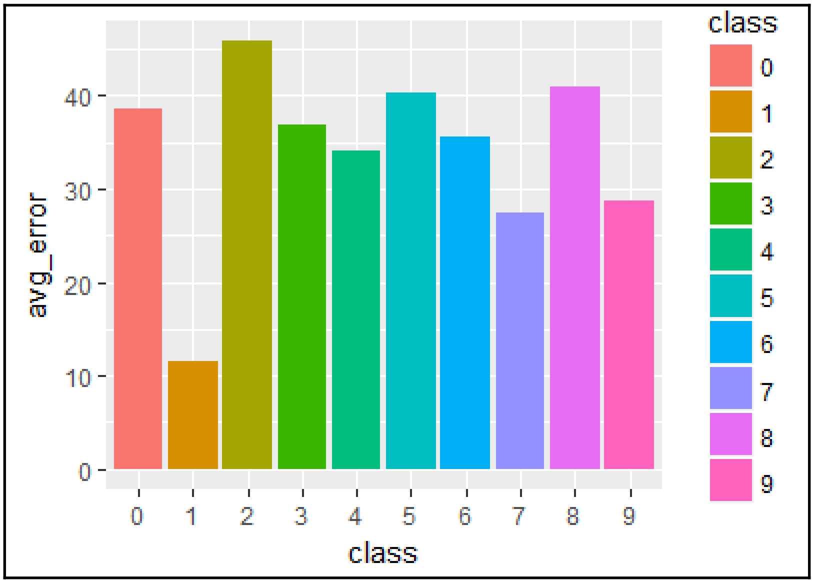

Amount
300
200
100
0
0
1
Class

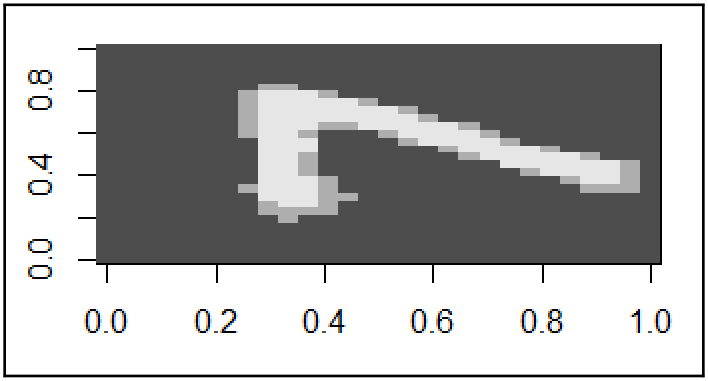

loss
60
40
20
0
acc
0.925
0.900
0.875
0.850
2
4
6
8
10
epoch
data
training
validation

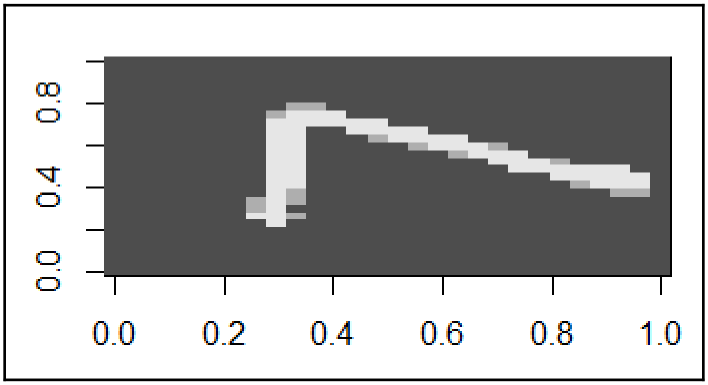

True positive rate
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

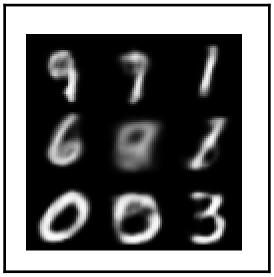

0.08
0.06
0.04
0.02
0.00
Reconstruction.MSE
0
20000
40000
instance number
Class
0
1

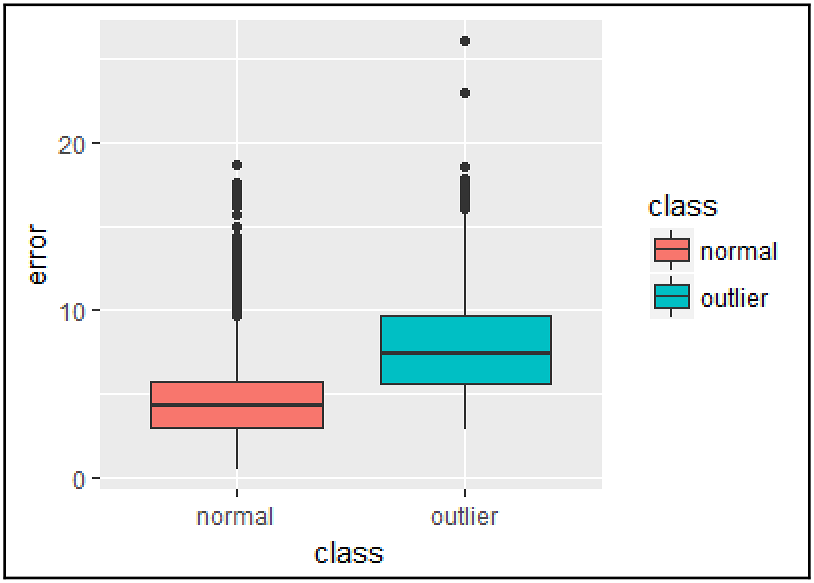

210
200
190
180
170
160
150
loss
0
10
20
30
40
50
epoch
data
training
validation

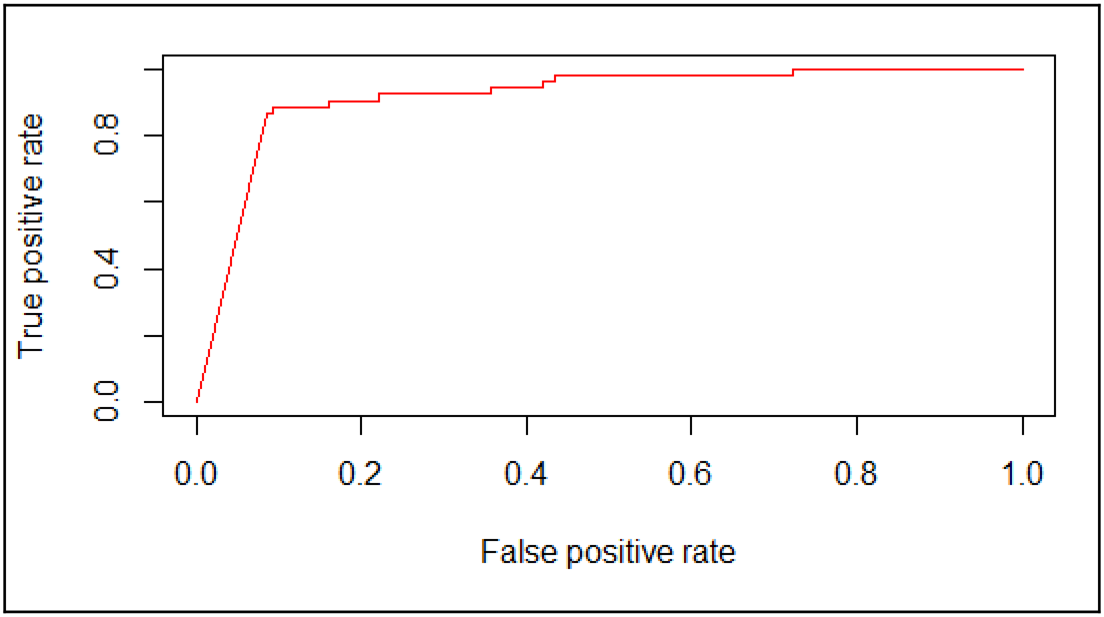

40
30
20
10
0
avg_error
0
1
2
3
4
5
6
7
8
9
class
class
0
1
2
3
4
5
6
7
8
9


0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0

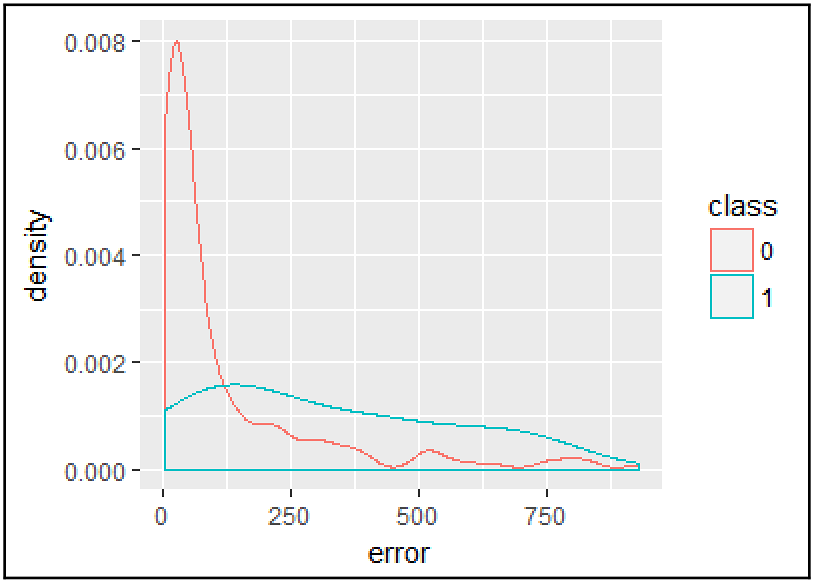

0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0

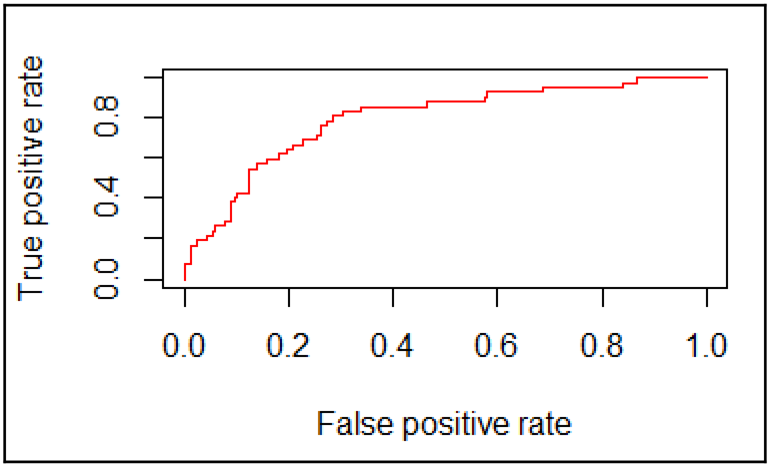






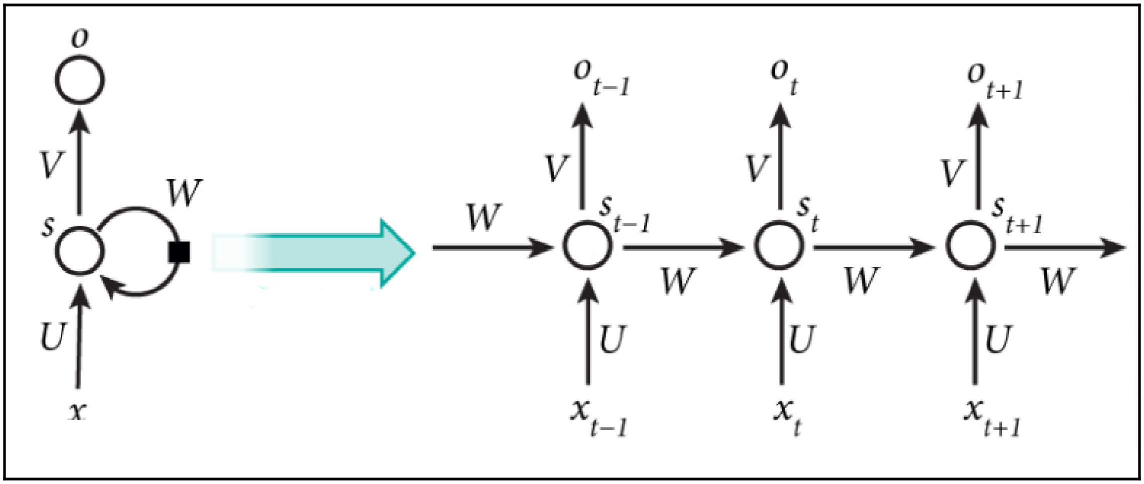
$$tfidf(w,d,D) = \frac{\text{frequency of w in d}}{\text{number of documents in D that have w in them}}$$


density
0.008
0.006
0.004
0.002
0.000
0
250
500
750
error
class
0
1


True positive rate
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

Chapter 15: Text Generation using Recurrent Neural Networks

$$s_k = \tanh(Ux_k + Ws_{k-1})$$
$$o_k = \mathrm{softmax}(Vs_k)$$



$$L(y, o) := -\frac{1}{N} \sum_{n \in N} y_n \log o_n$$

$$\frac{\partial L}{\partial U}, \frac{\partial L}{\partial V}, \frac{\partial L}{\partial W}$$

$$\frac{\partial L}{\partial W} := \frac{\partial L}{\partial o_t} \cdot \frac{\partial o_t}{\partial s_t} \cdot \frac{\partial s_t}{\partial W}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial o_t} \cdot \frac{\partial o_t}{\partial s_t} \cdot \frac{\partial s_t}{\partial s_{t-1}} \cdot \frac{\partial s_{t-1}}{\partial s_{t-2}} \cdot \ldots \cdot \frac{\partial s_1}{\partial W}$$

$$
\begin{aligned}
i &= \sigma(U^i x_t + W^i s_{t-1}) \\
f &= \sigma(U^f x_t + W^f s_{t-1}) \\
o &= \sigma(U^o x_t + W^o s_{t-1}) \\
g &= \tanh(U^g x_t + W^g s_{t-1}) \\
c_t &= c_{t-1} \cdot f + g \cdot i \\
s_t &= \tanh(c_t) \cdot o
\end{aligned}
$$

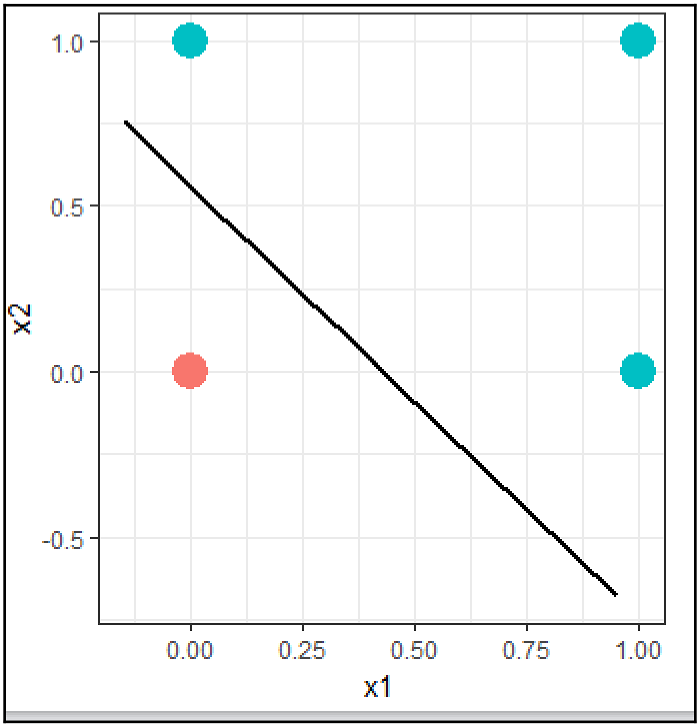
$$\sigma(x) := \frac{1}{1 + e^{-x}}$$

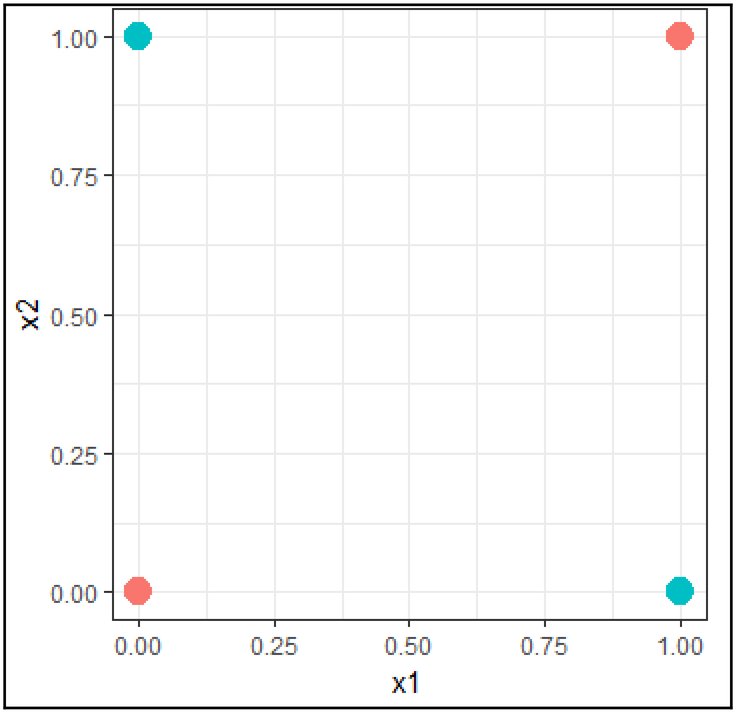
$$z = \sigma(U^z x_t + W^z s_{t-1})$$

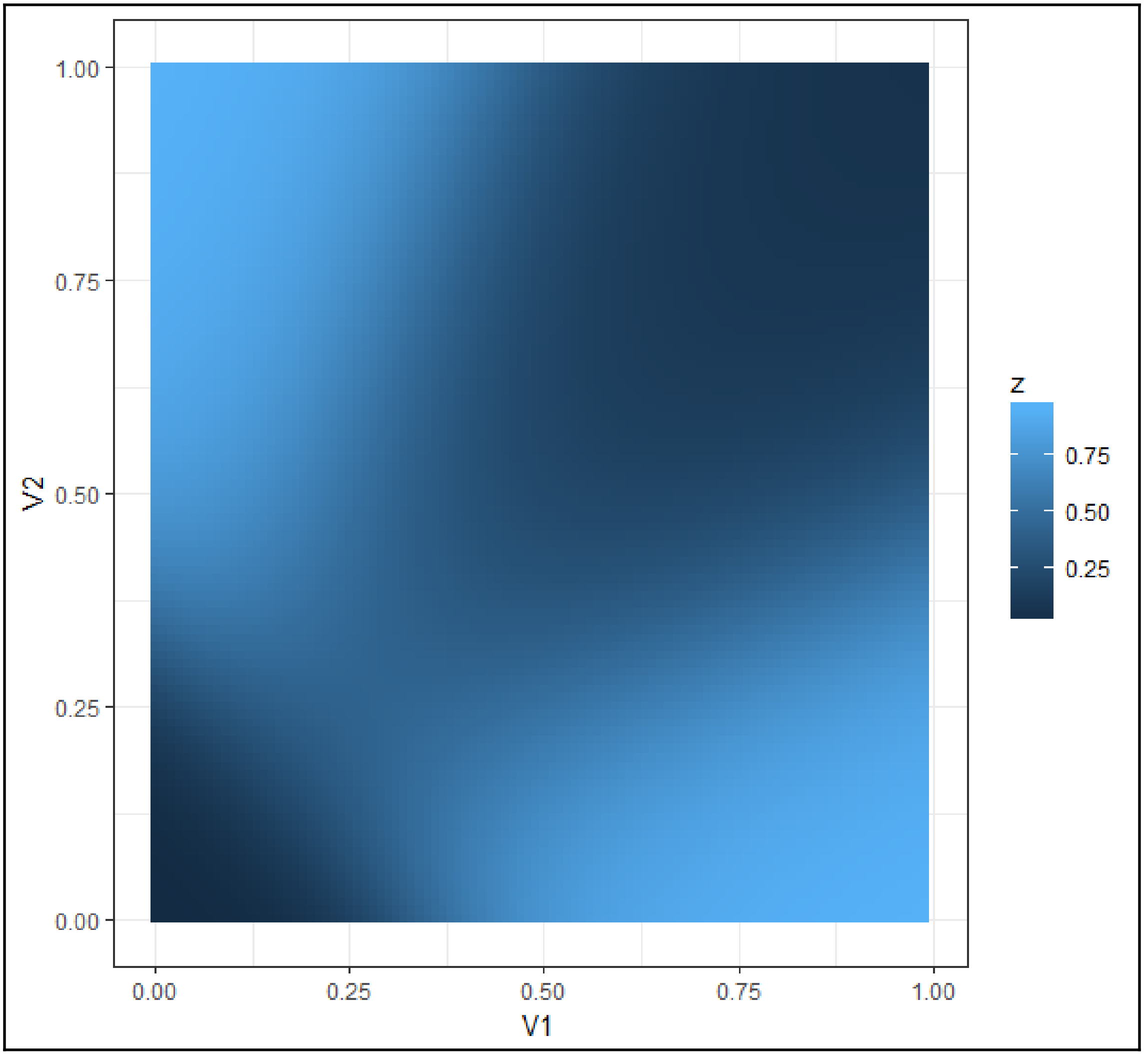
$$r = \sigma(U^r x_t + W^r s_{t-1})$$

$$h = \tanh(U^h x_t + W^h (s_{t-1} r))$$

$$s_t = (1 - z) \cdot h + z \cdot s_{t-1}$$

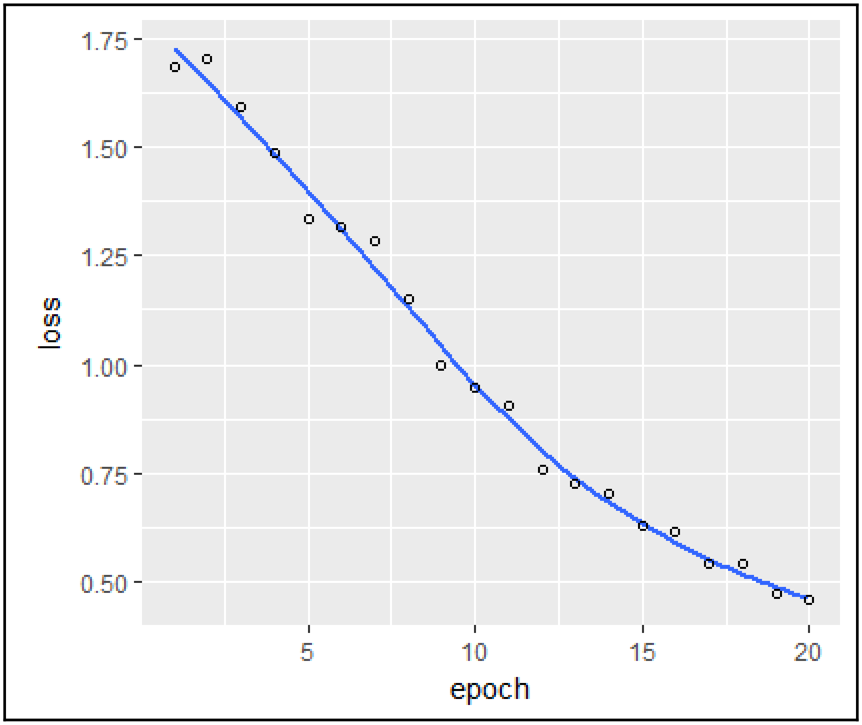



1.00
0.75
0.50
0.25
0.00
x2
0.00
0.25
0.50
0.75
1.00
x1

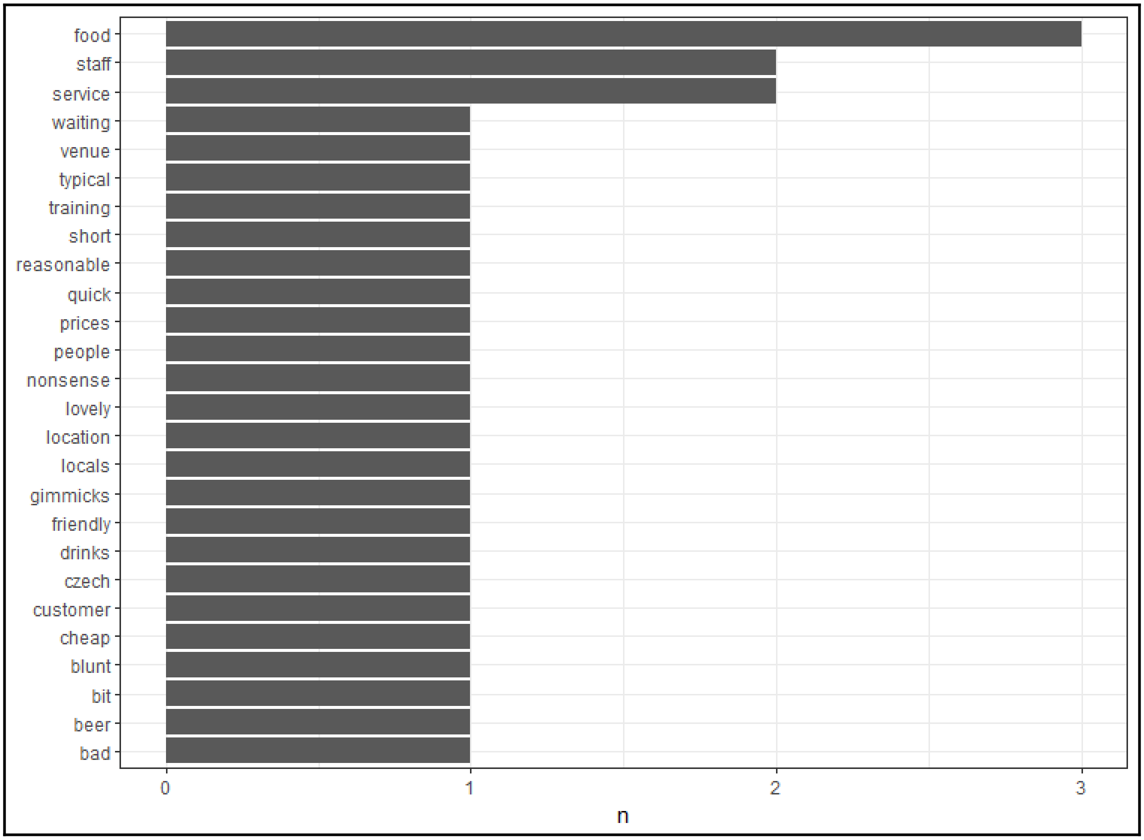

1.00
0.75
0.50
0.25
0.00
V2
0.00
0.25
0.50
0.75
1.00
V1
z
0.75
0.50
0.25

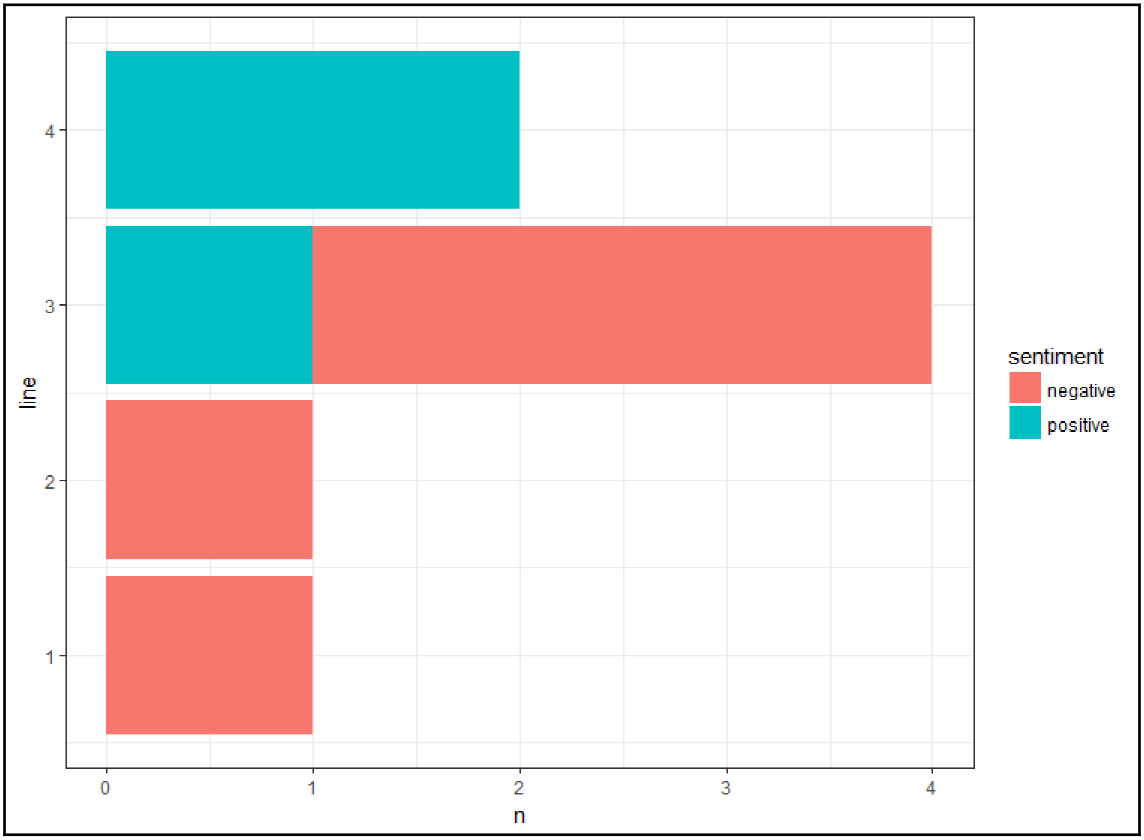
$$\mathbb{P}\left(\boldsymbol{c} \mid \boldsymbol{h}\right)$$

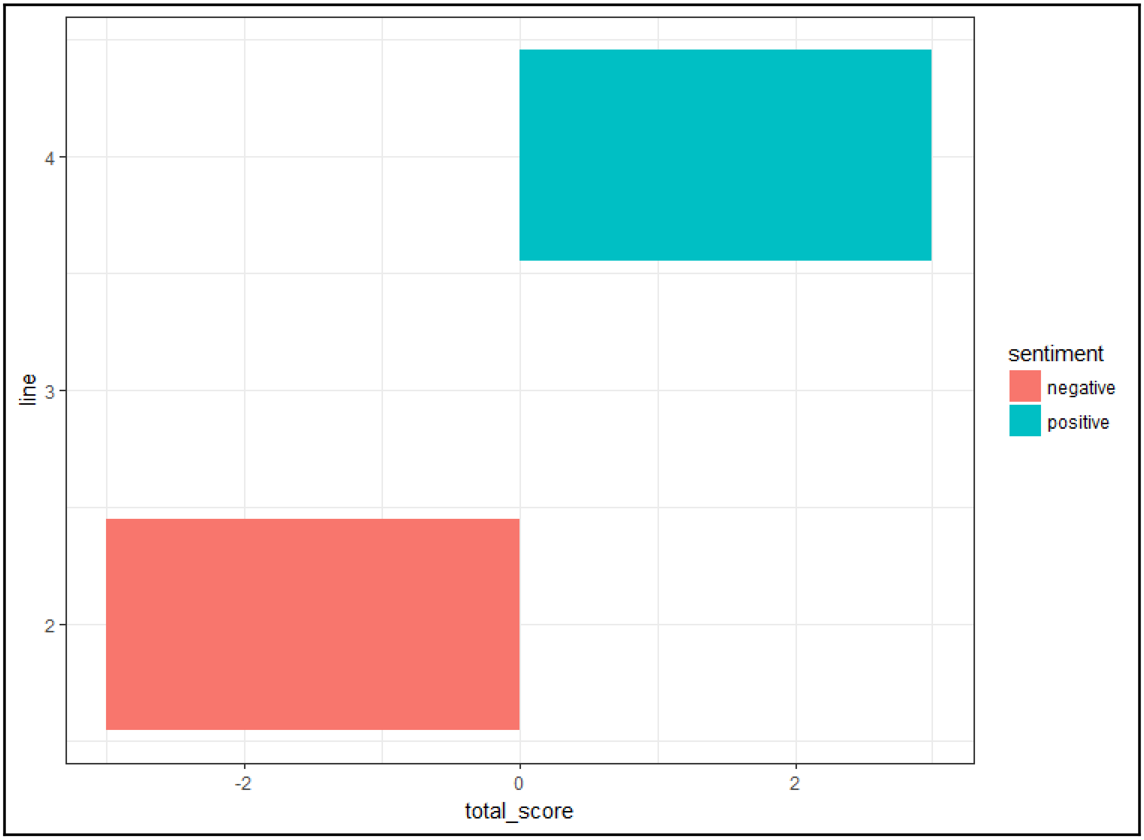
$$\mathbb{P}\left(\boldsymbol{c} \mid \boldsymbol{h}\right)$$

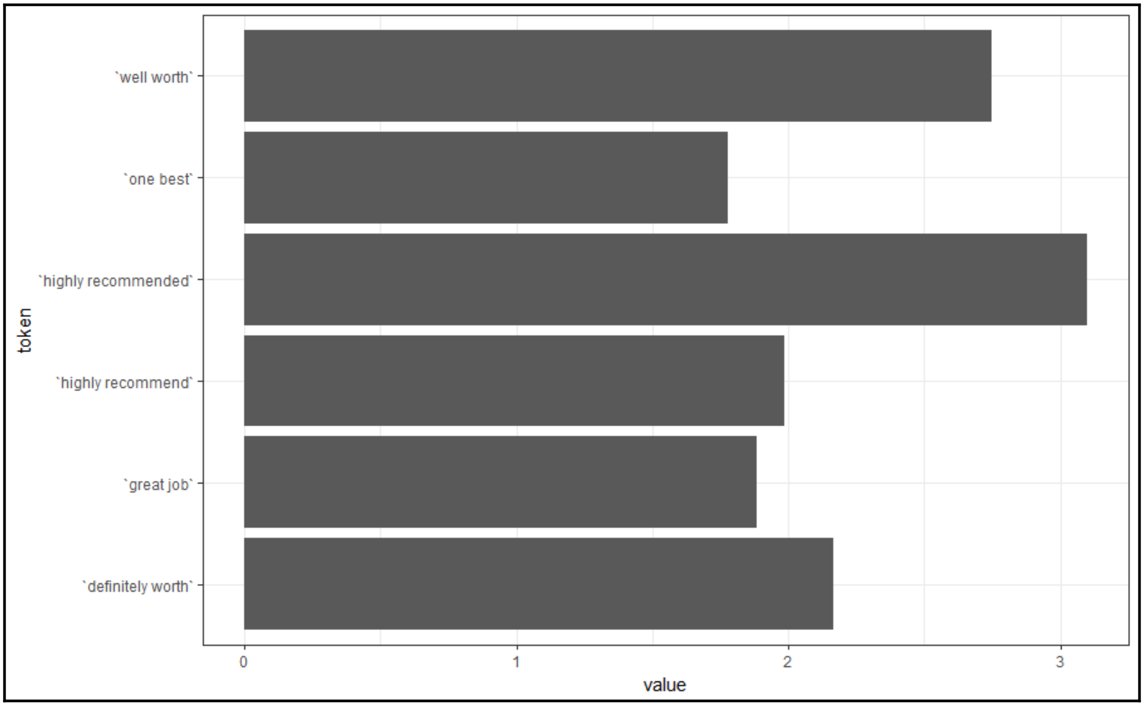

1.75
1.50
1.25
1.00
0.75
0.50
loss
5
10
15
20
epoch

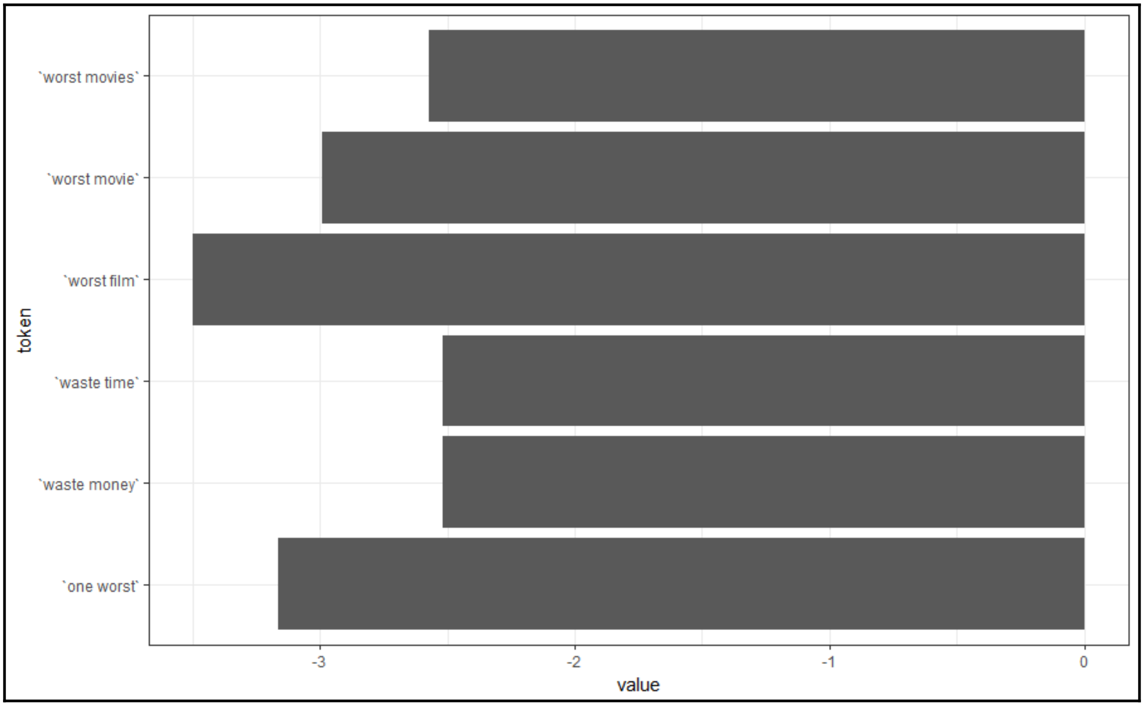
Chapter 16: Sentiment Analysis with Word Embedding

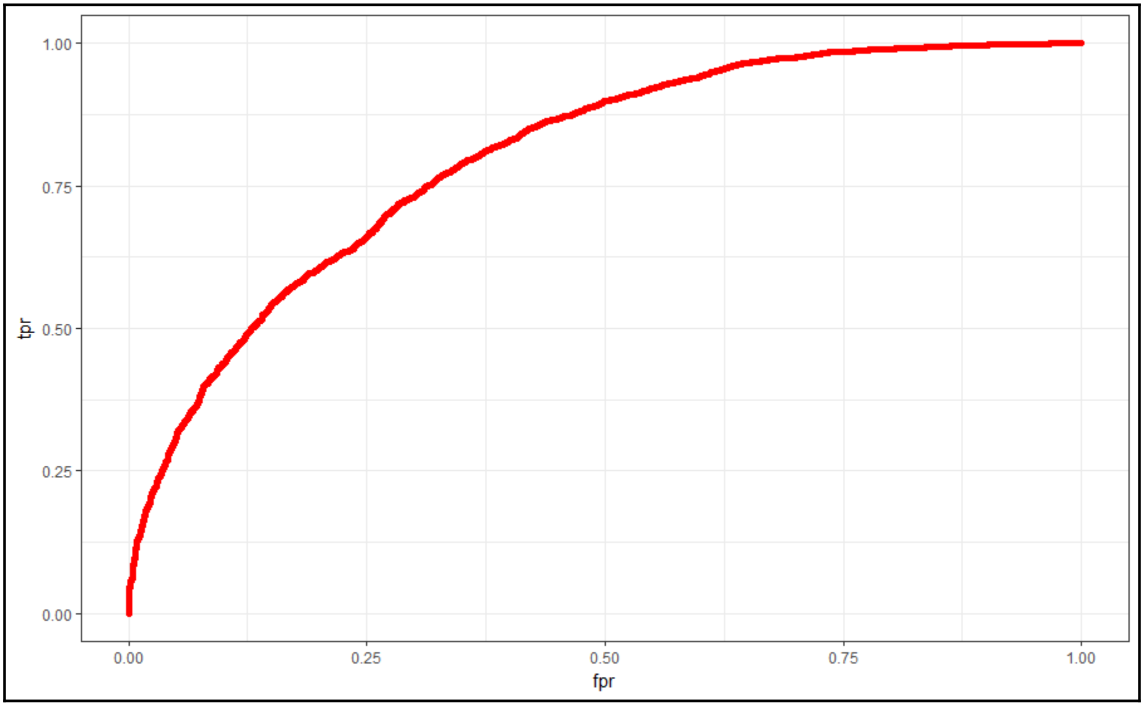



4
3
2
1
line
0
1
2
3
4
n
sentiment
negative
positive


4
3
2
line
-2
0
2
total_score
sentiment
negative
positive


`well worth`
`one best`
`highly recommended`
`highly recommend`
`great job`
`definitely worth`
token
0
1
2
3
value

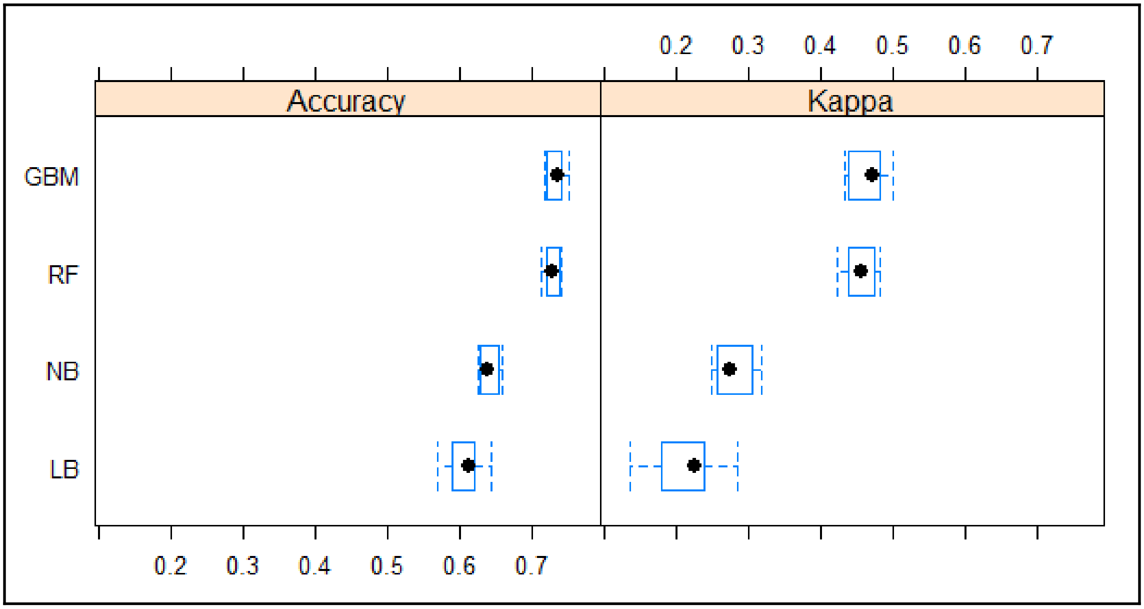

`worst movies`
`worst movie`
`worst film`
`waste time`
`waste money`
`one worst`
token
-3
-2
-1
0
value

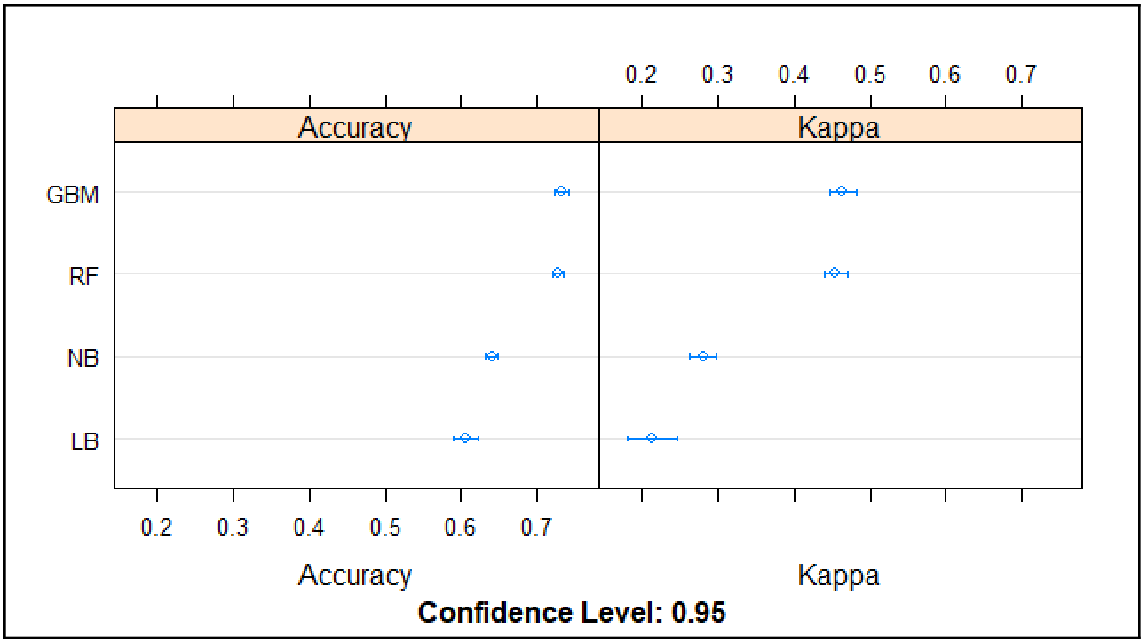


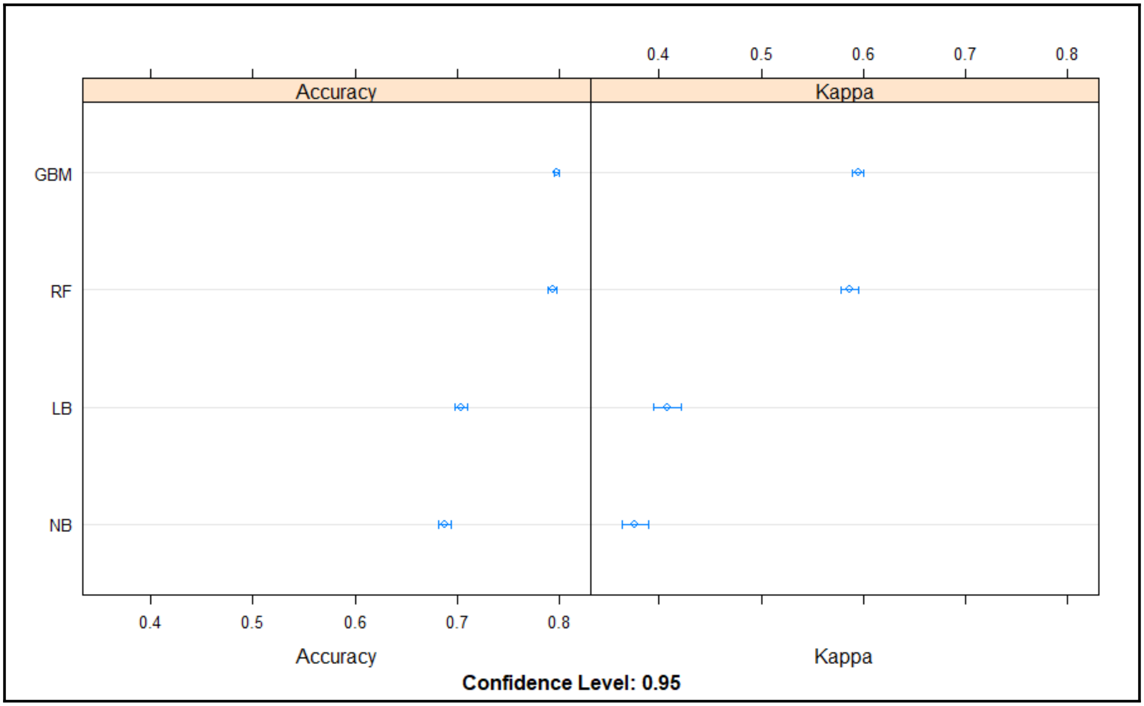
$$w_i^T w_j + b_i + b_j = log(X_{ij})$$

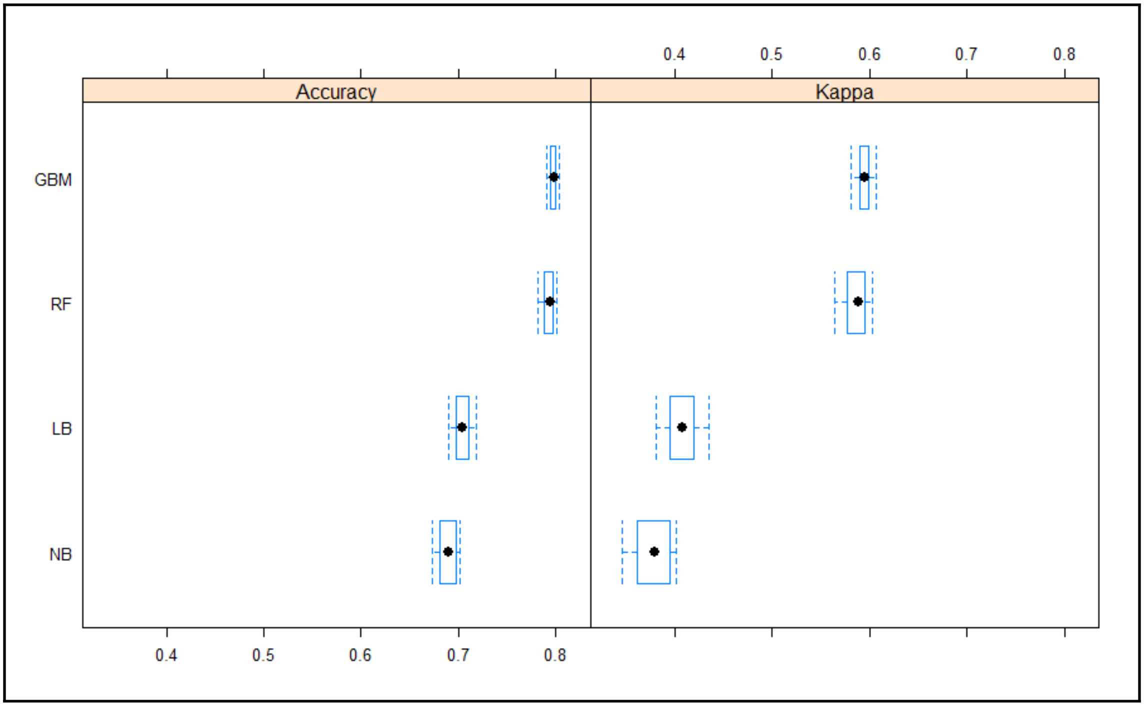
$$\sum_{i,j}^{V} f(X_{i,j}).\,(w_i^T w_j - b_i - b_j - log(Xij))^2$$

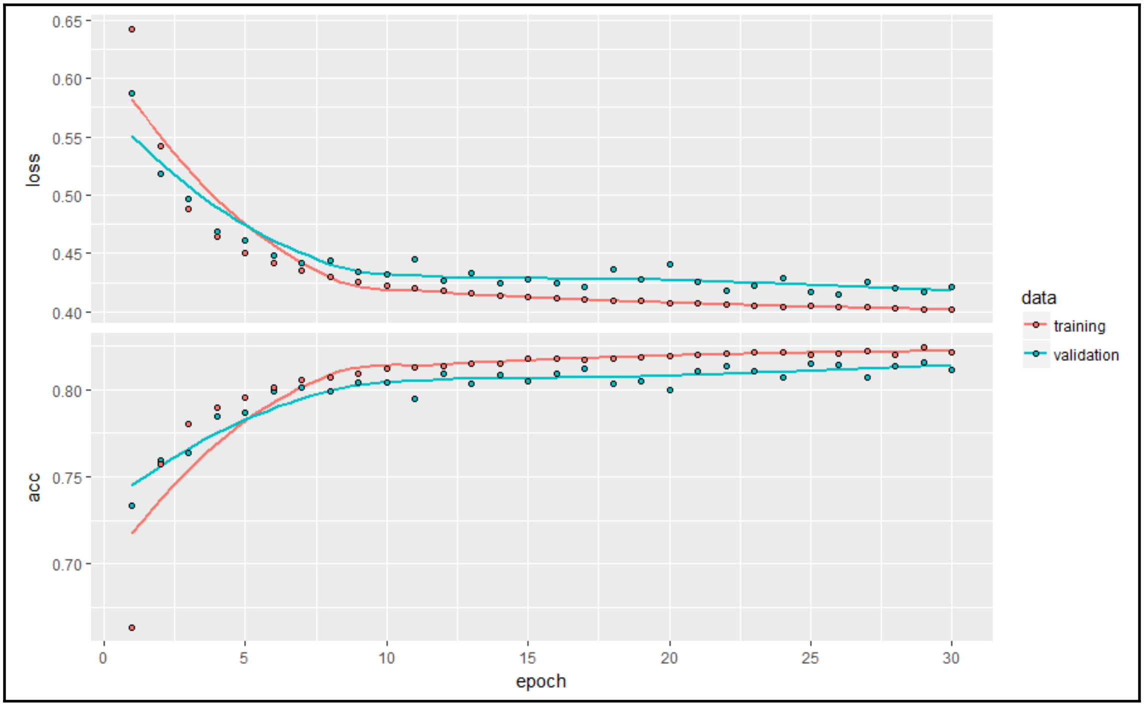
$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{X_{max}})^{\alpha} & , \quad X_{ij} < X_{max} \\ 1 & , \quad X_{ij} \geq X_{max} \end{cases}$$

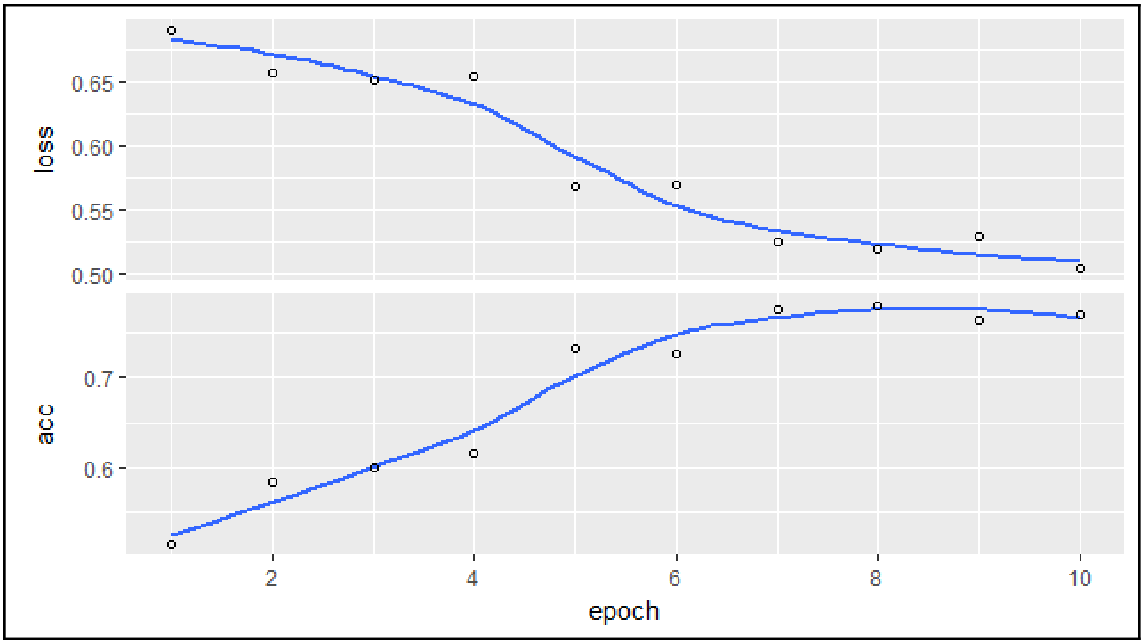

Accuracy
Kappa
GBM
RF
NB
LB
0.2
0.3
0.4
0.5
0.6
0.7

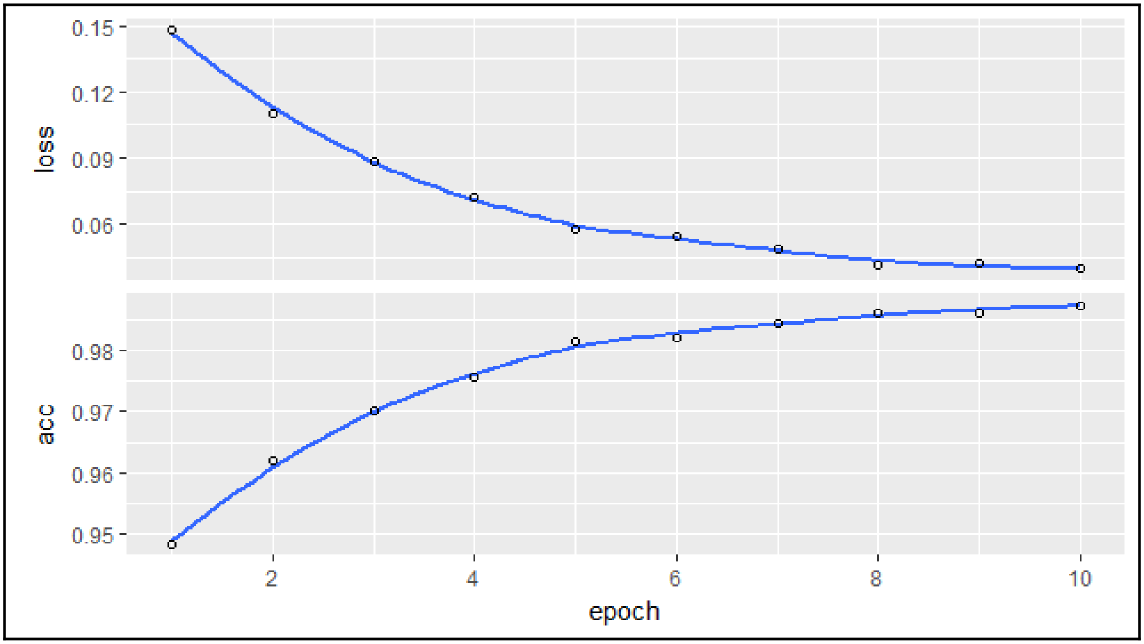

Accuracy
Kappa
GBM
RF
NB
LB
0.2
0.3
0.4
0.5
0.6
0.7
Confidence Level: 0.95

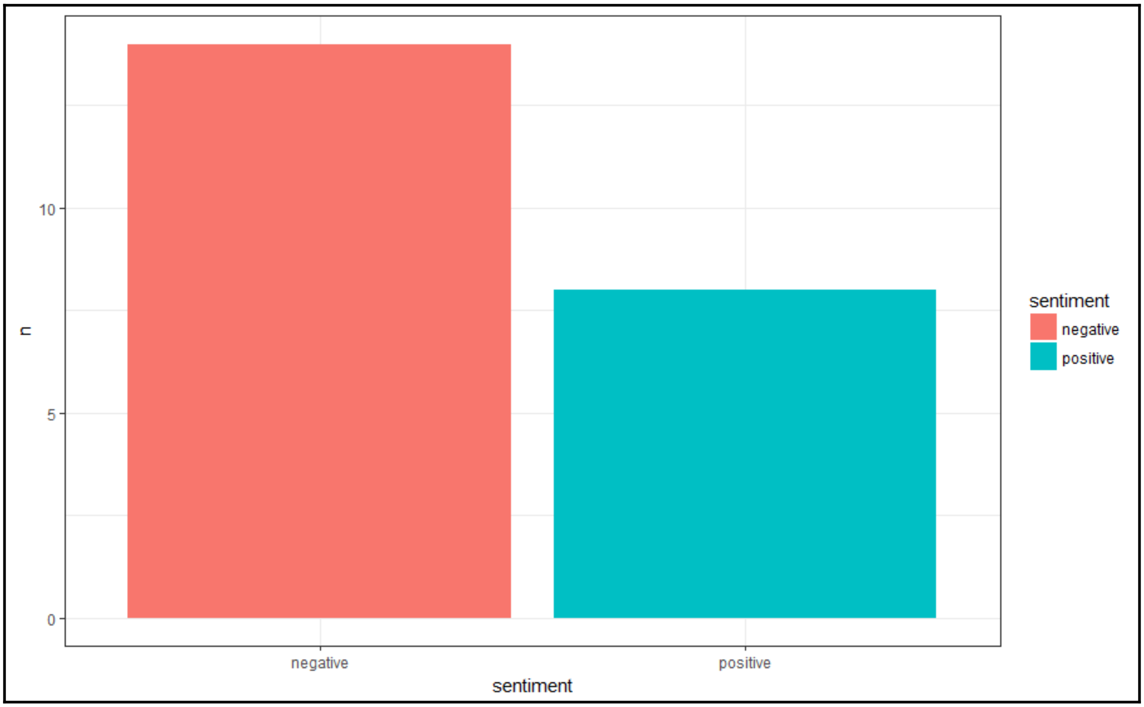

Accuracy
Kappa
GBM
RF
LB
NB
0.4
0.5
0.6
0.7
0.8
0.4
0.5
0.6
0.7
0.8
Accuracy
Kappa
Confidence Level: 0.95

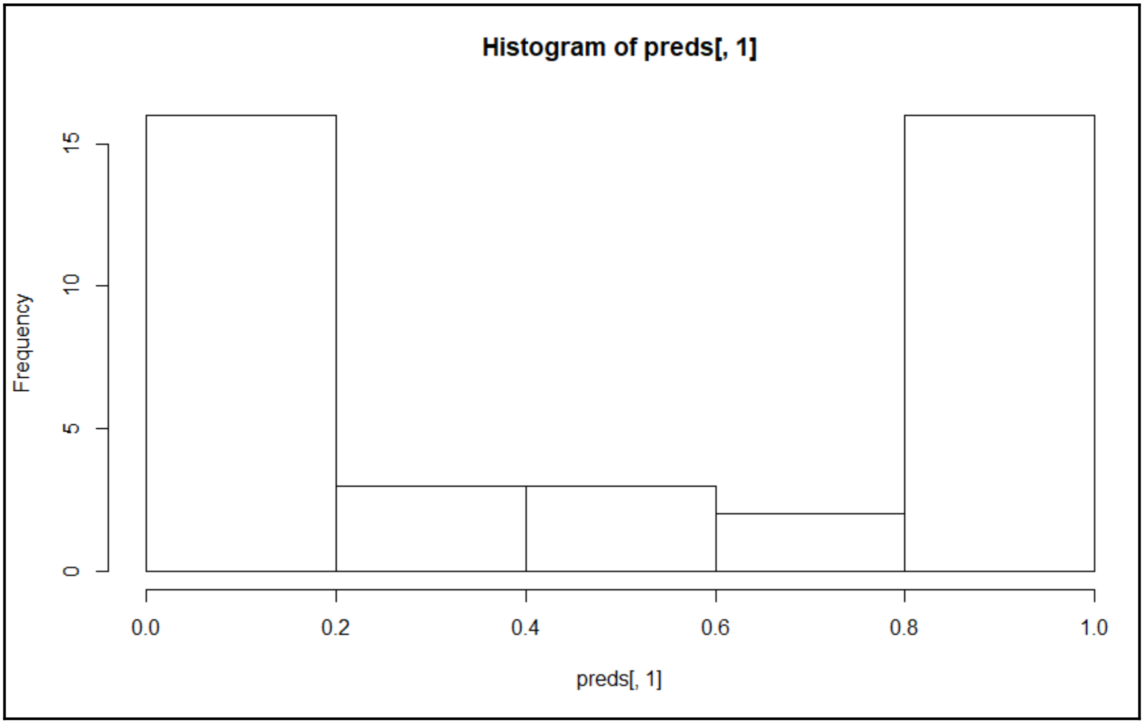

Accuracy
Kappa
GBM
RF
LB
NB
0.4
0.5
0.6
0.7
0.8
0.4
0.5
0.6
0.7
0.8


0.65
0.60
0.55
0.50
0.45
0.40
loss
0.80
0.75
0.70
acc
0
5
10
15
20
25
30
epoch
data
training
validation


0.65
0.60
0.55
0.50
loss
0.7
0.6
acc
2
4
6
8
10
epoch


0.15
0.12
0.09
0.06
loss
0.98
0.97
0.96
0.95
acc
2
4
6
8
10
epoch


10
5
0
n
negative
positive
sentiment
sentiment
negative
positive


Histogram of preds[, 1]
Frequency
0
5
10
15
0.0
0.2
0.4
0.6
0.8
1.0
preds[, 1]

Graphics Bundle Ends Here

Index