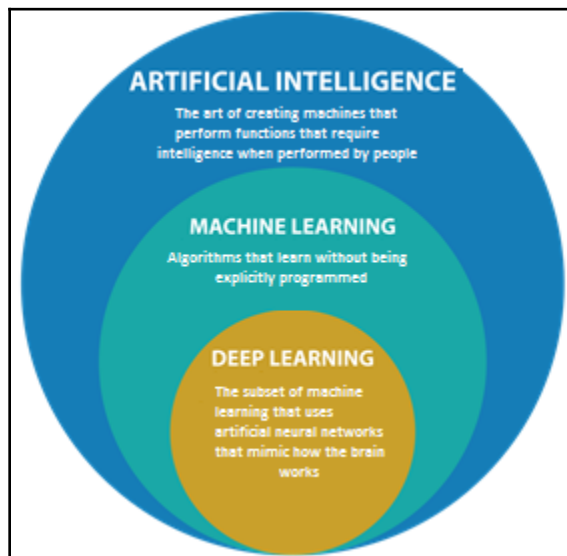
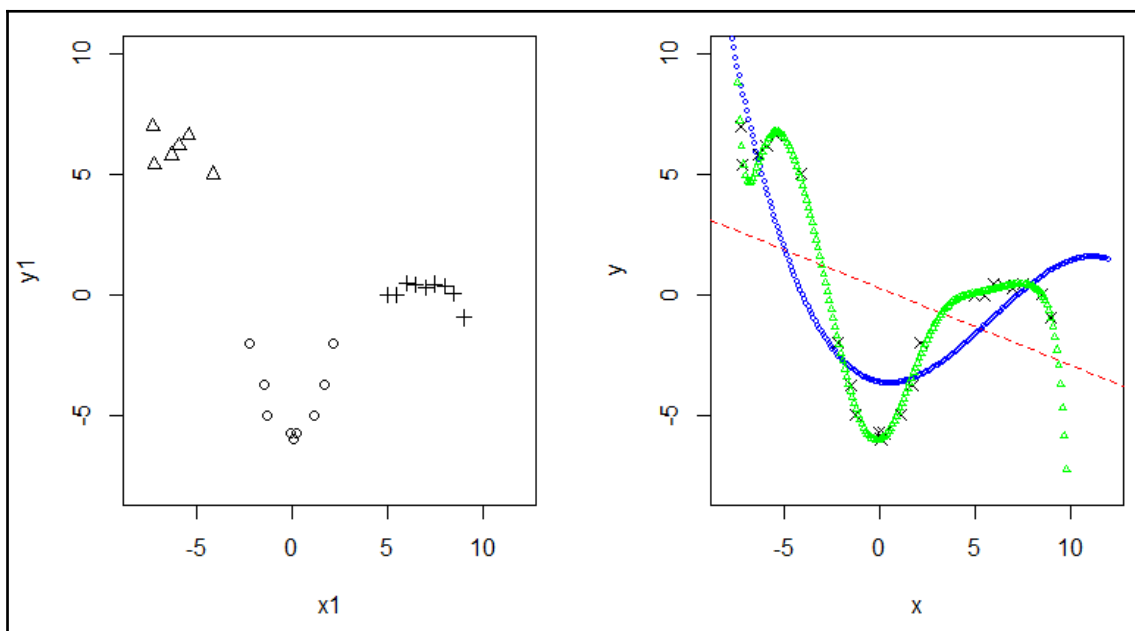
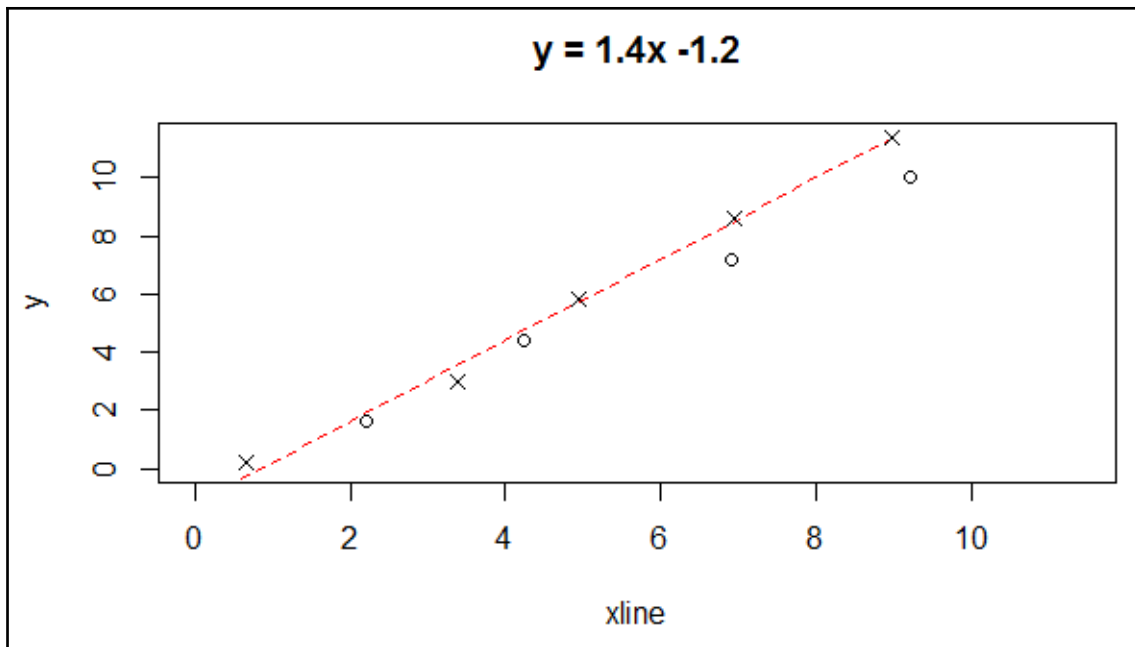


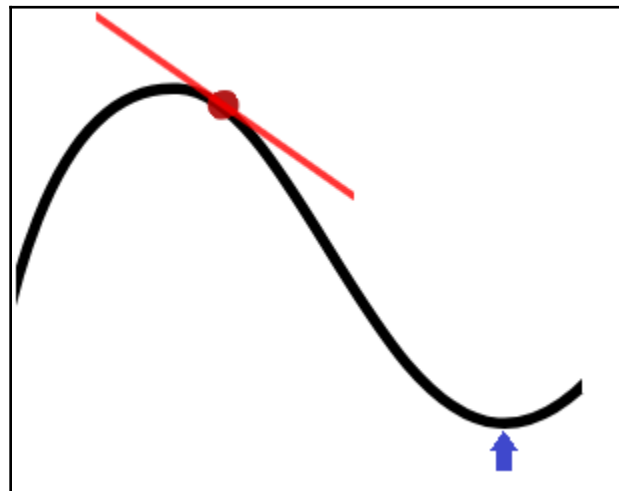
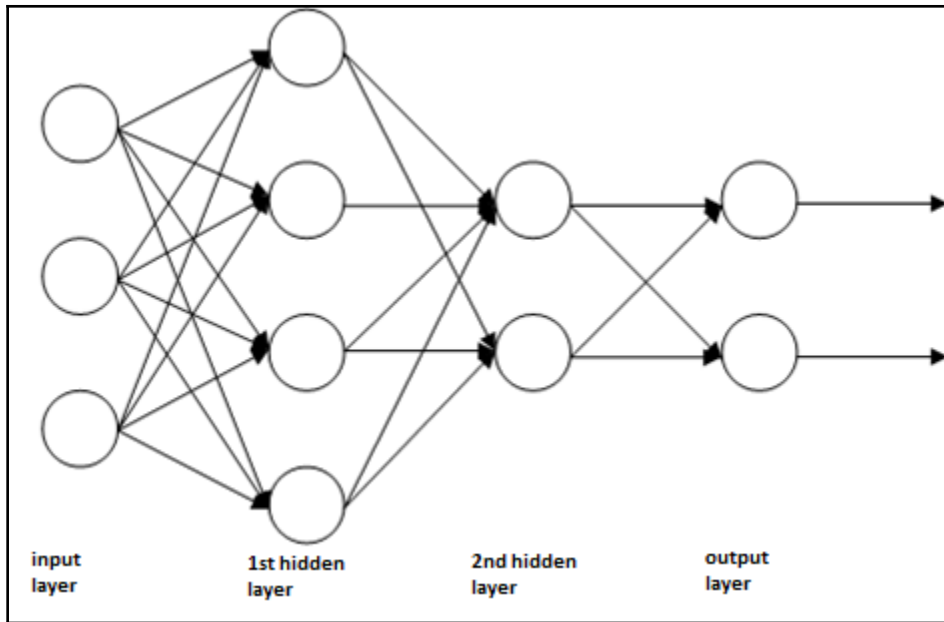
# 1

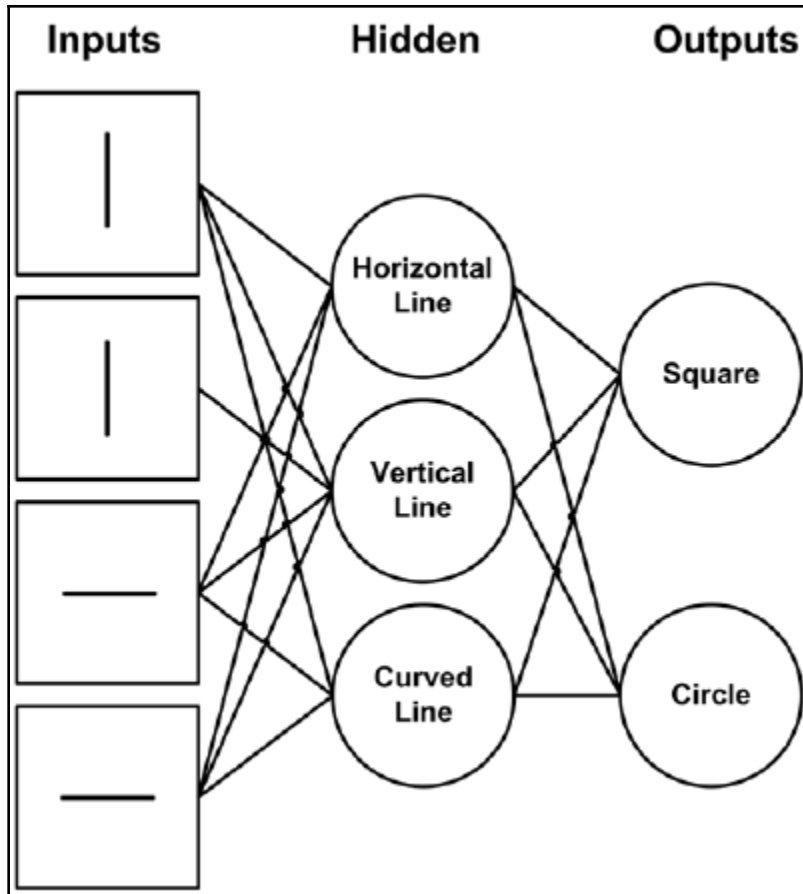
## Graphic Bundle

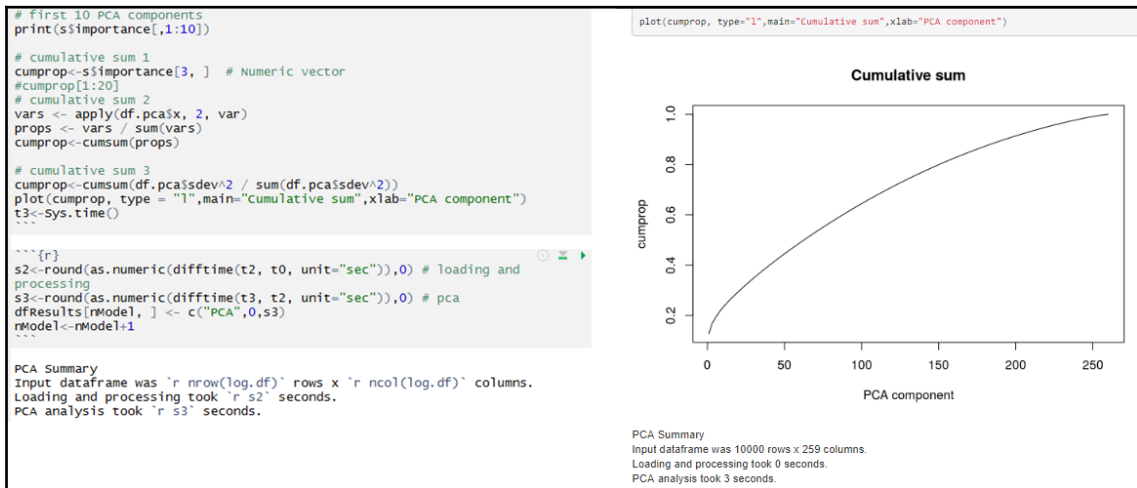
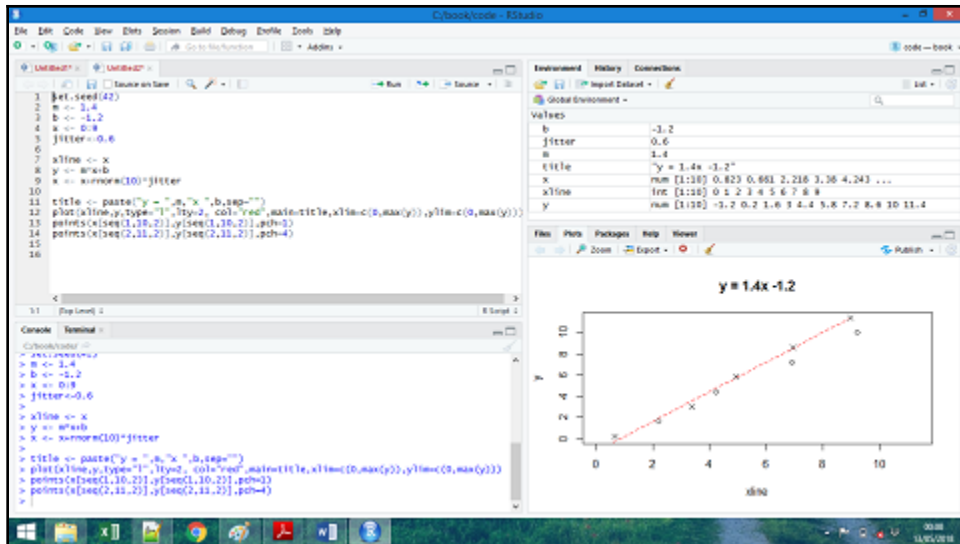
### Chapter 1: Getting Started with Deep Learning











The screenshot shows a web browser window displaying an R Shiny application. The browser address bar shows the URL `http://127.0.0.1:5593`. The application title is "R Deep Learning".

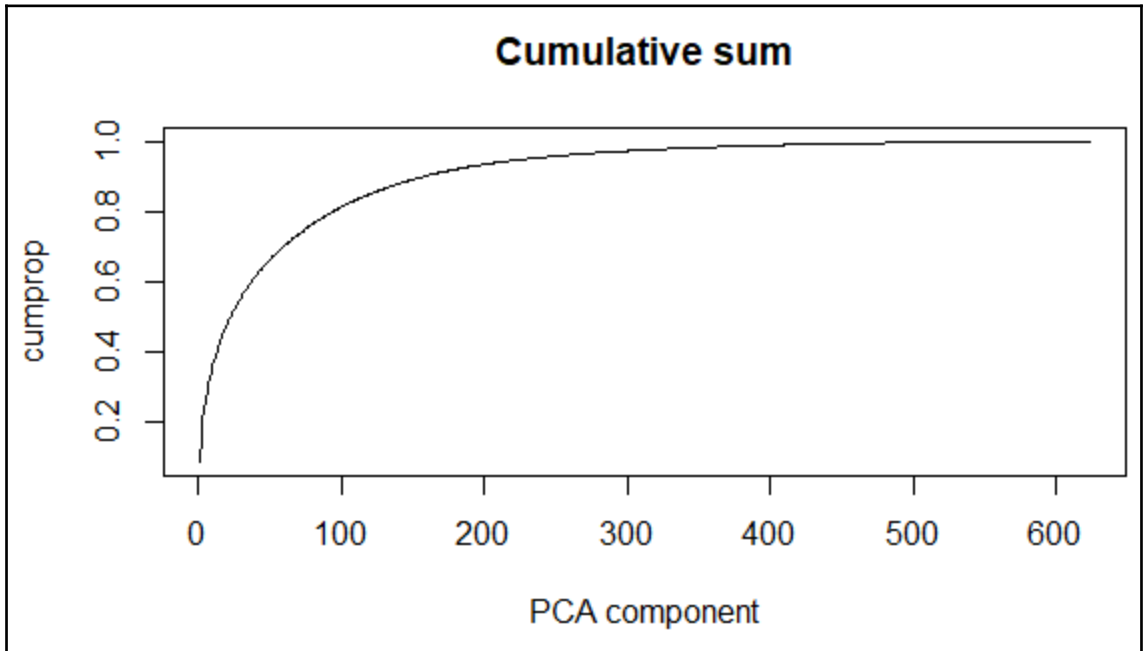
The main interface is titled "Convolutional Layers" and contains the following elements:

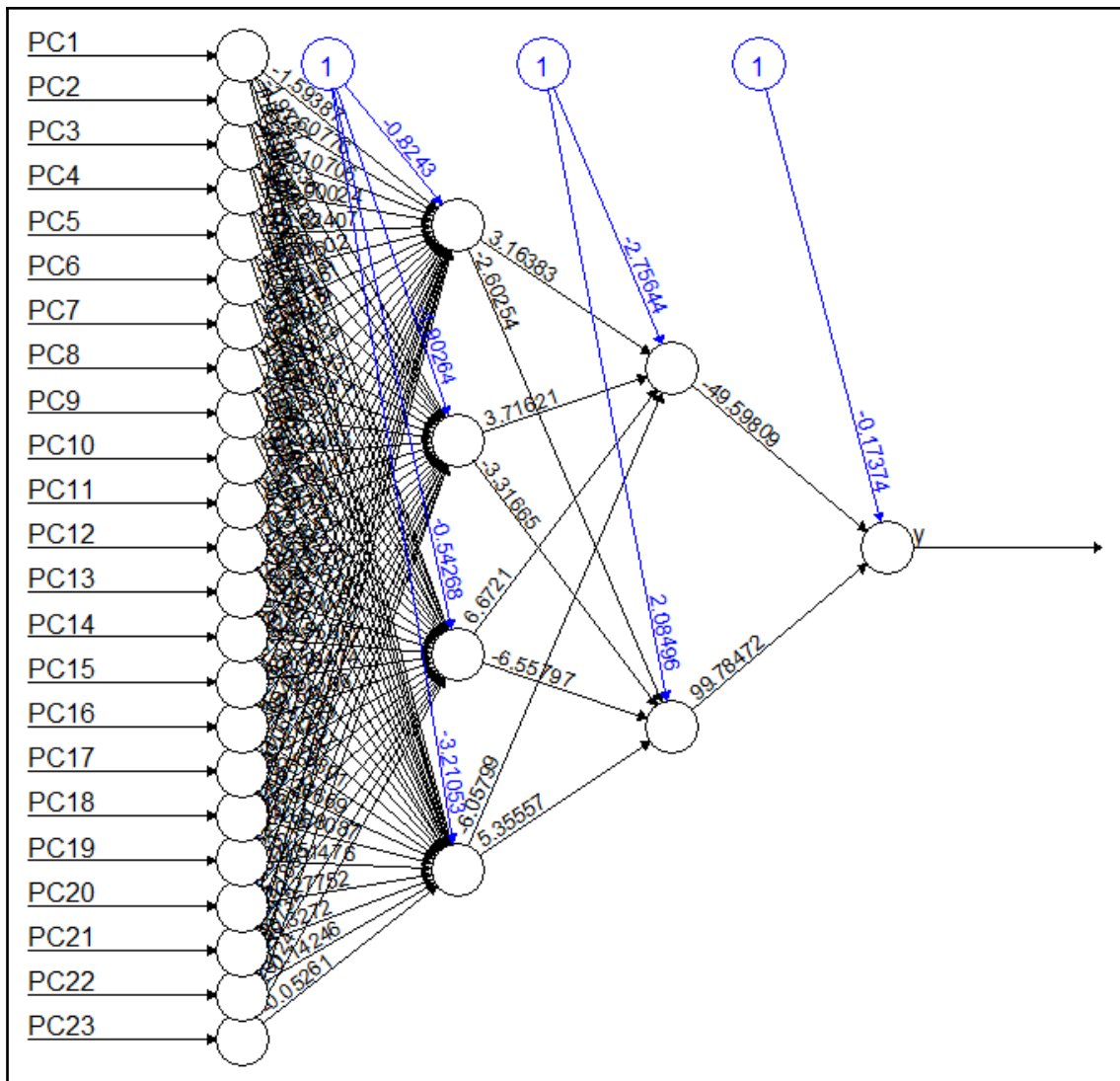
- Select an image:** A slider control with a value of 4 and a range from 1 to 100.
- Select convolutional layer:** Radio buttons for "Horizontal Line", "Vertical Line" (selected), "Diagonal1", and "Diagonal2".
- Conv Layer:** A table with three rows, each containing the value 100.

Below the controls are two image panels:

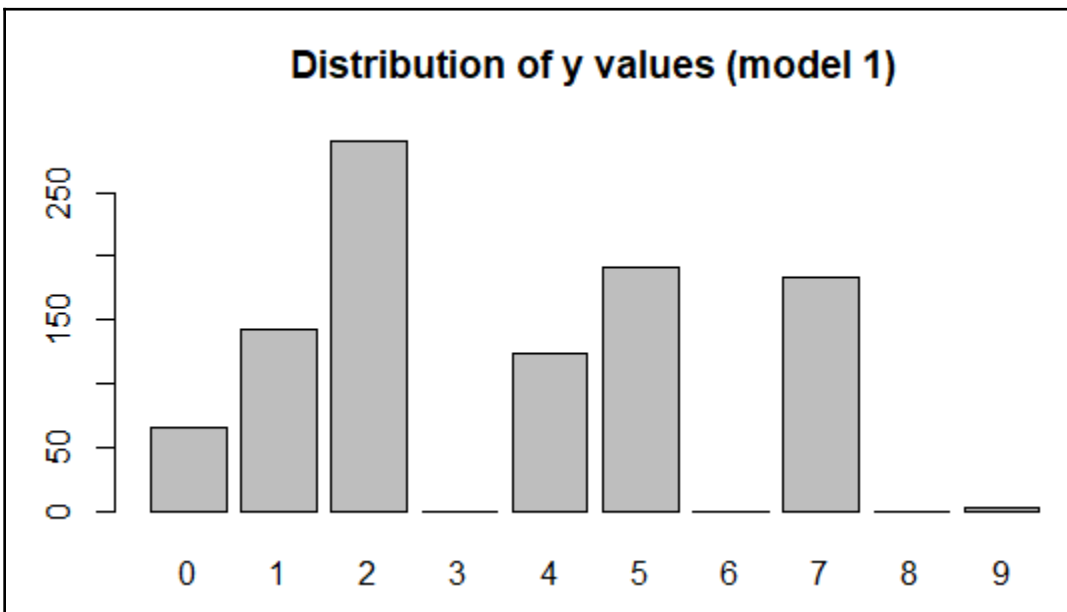
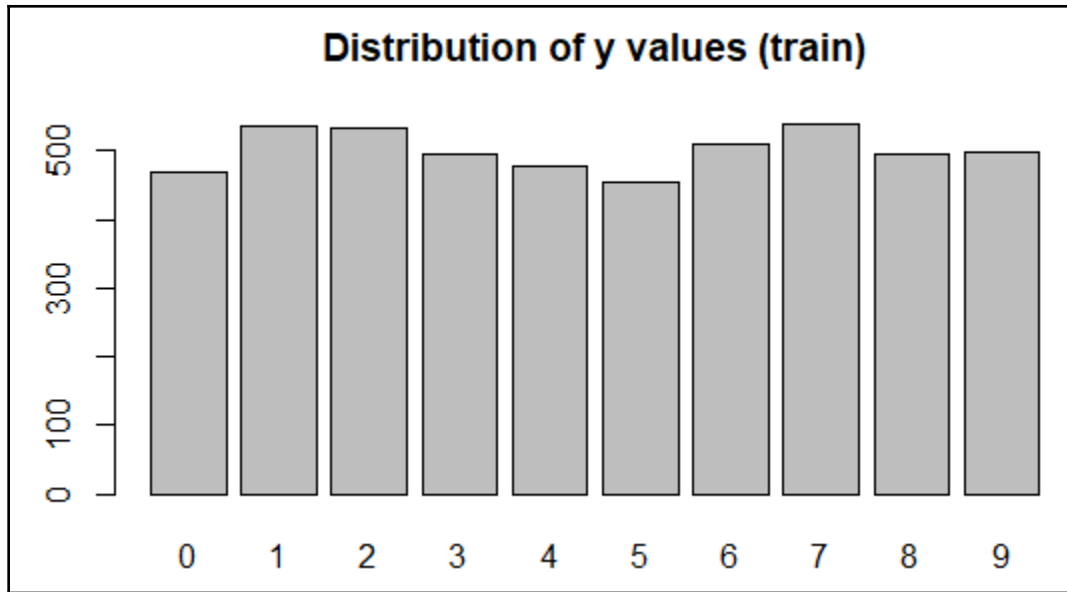
- Original image:** Displays a handwritten digit '4' on a black background. The text "index: 4, label = 4" is centered above the image.
- Image after Convolutional applied:** Displays the same handwritten digit '4' on a black background, but with a blurred or smoothed appearance. The text "index: 4, label = 4" is centered above the image.

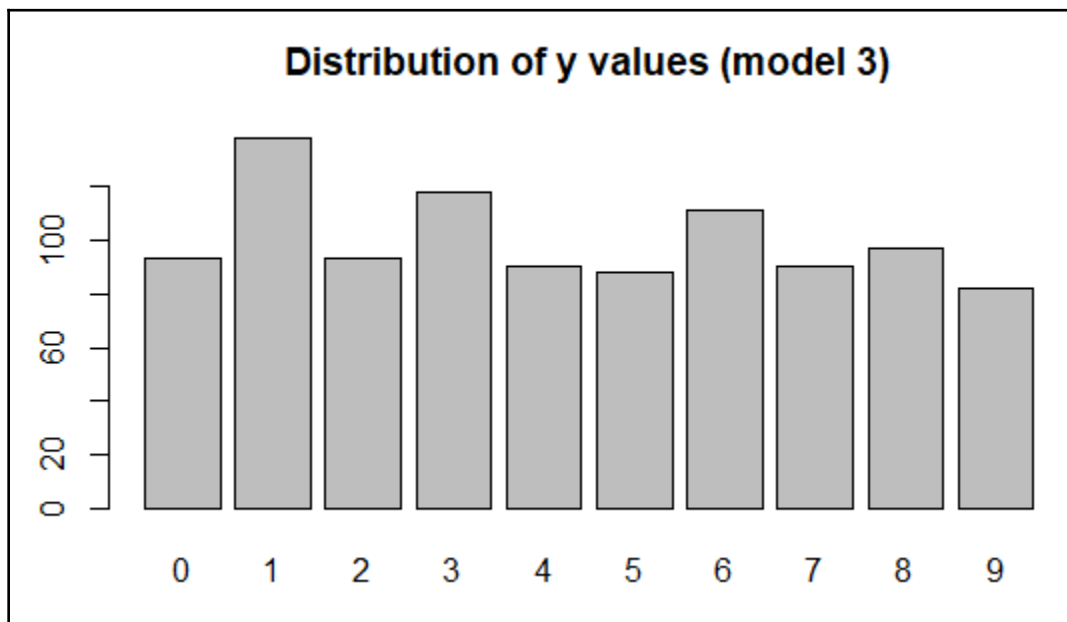
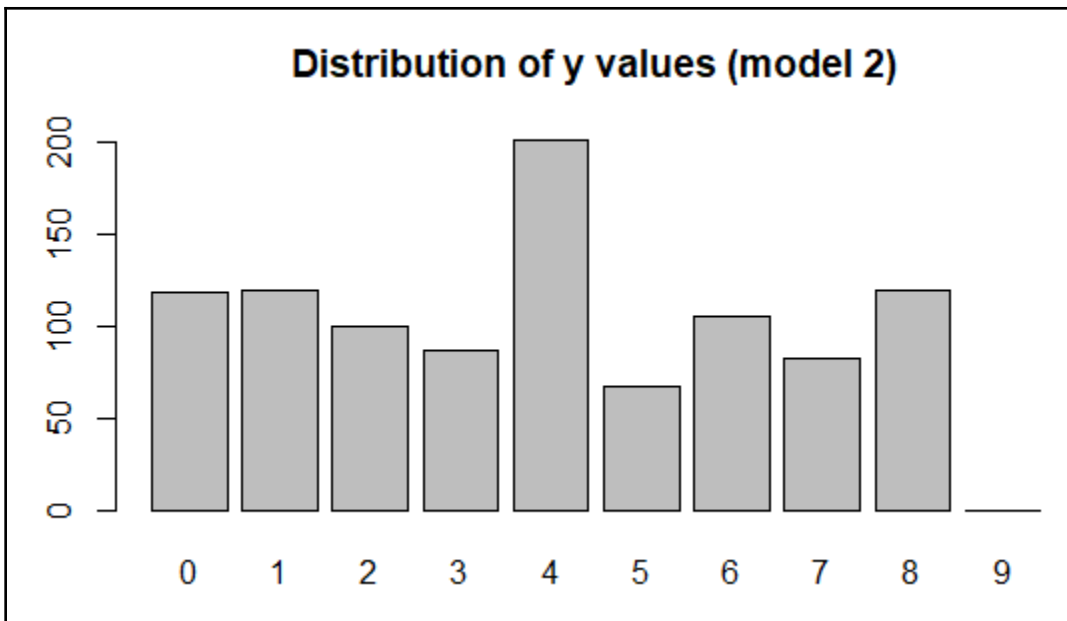
## Chapter 2: Training a Prediction Model

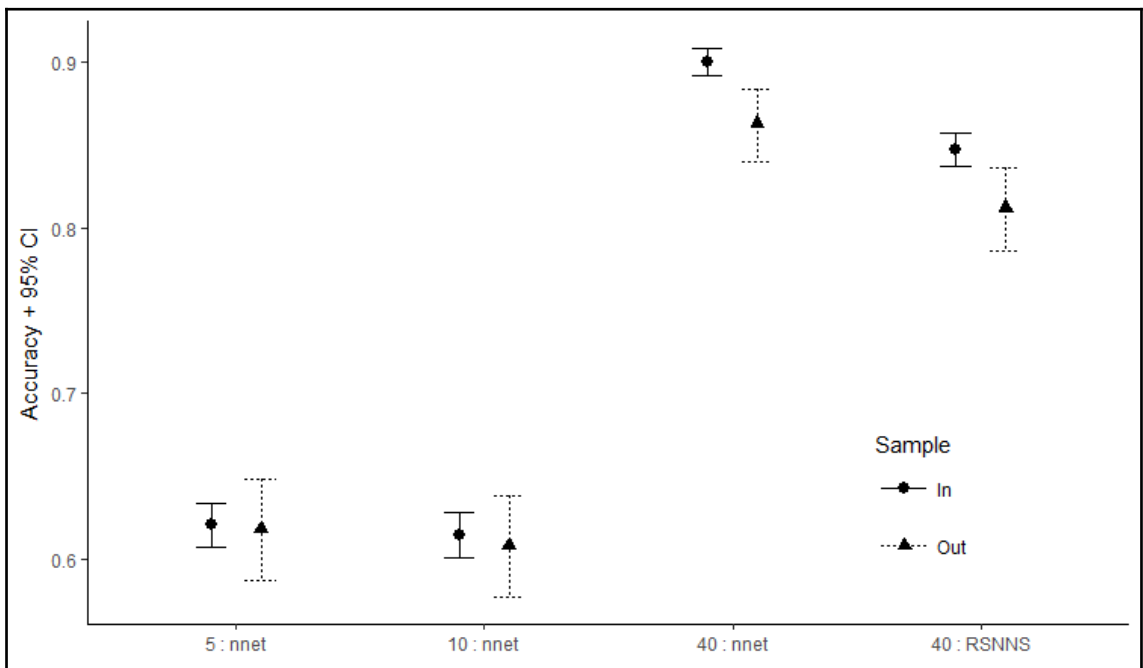
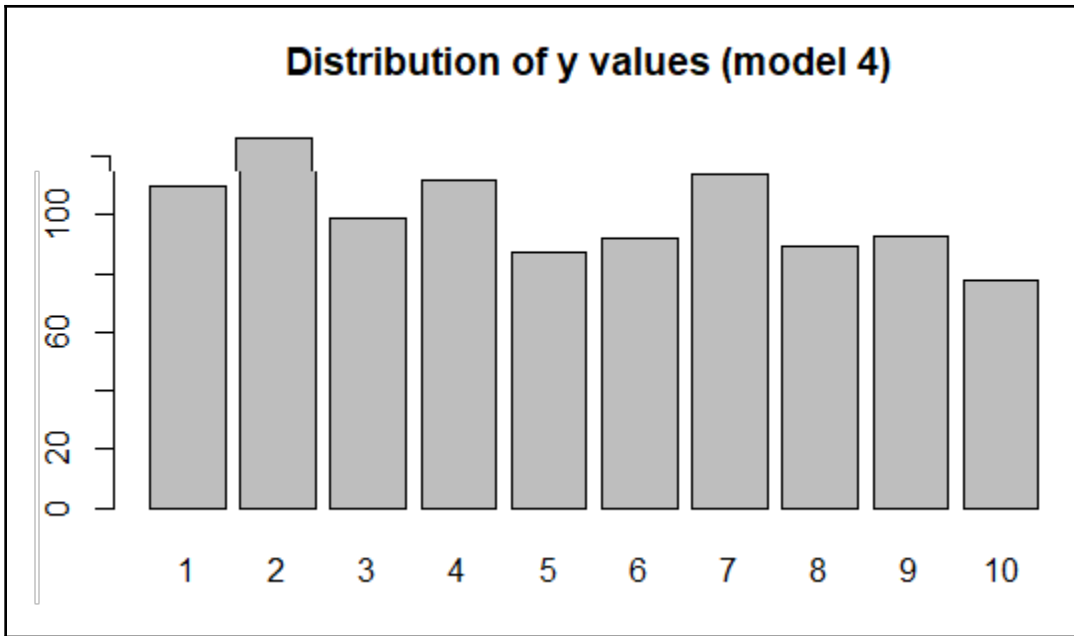


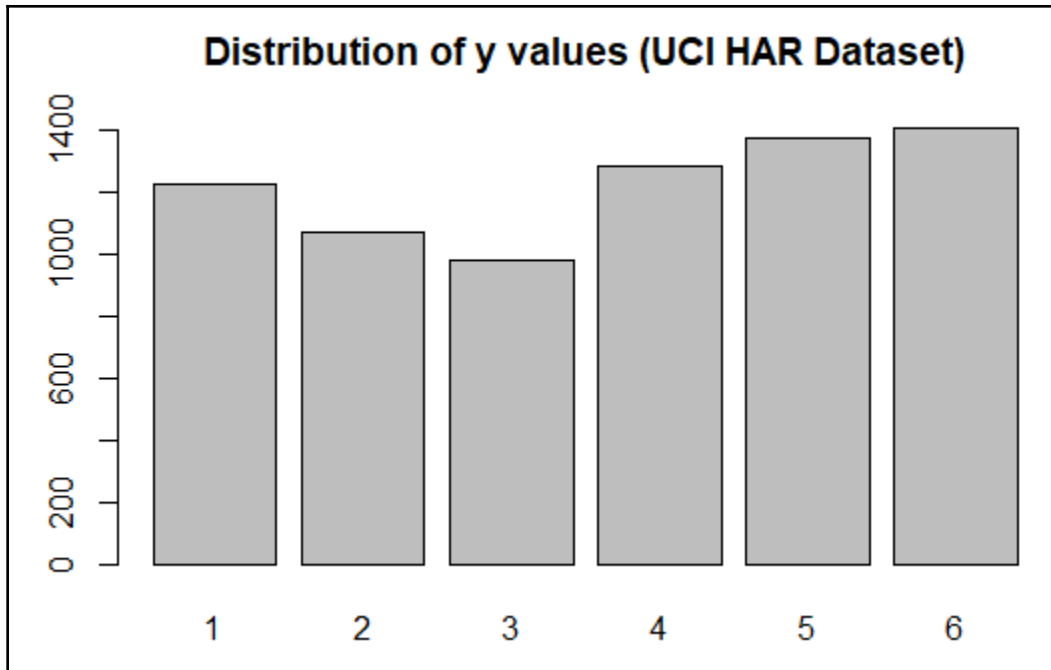




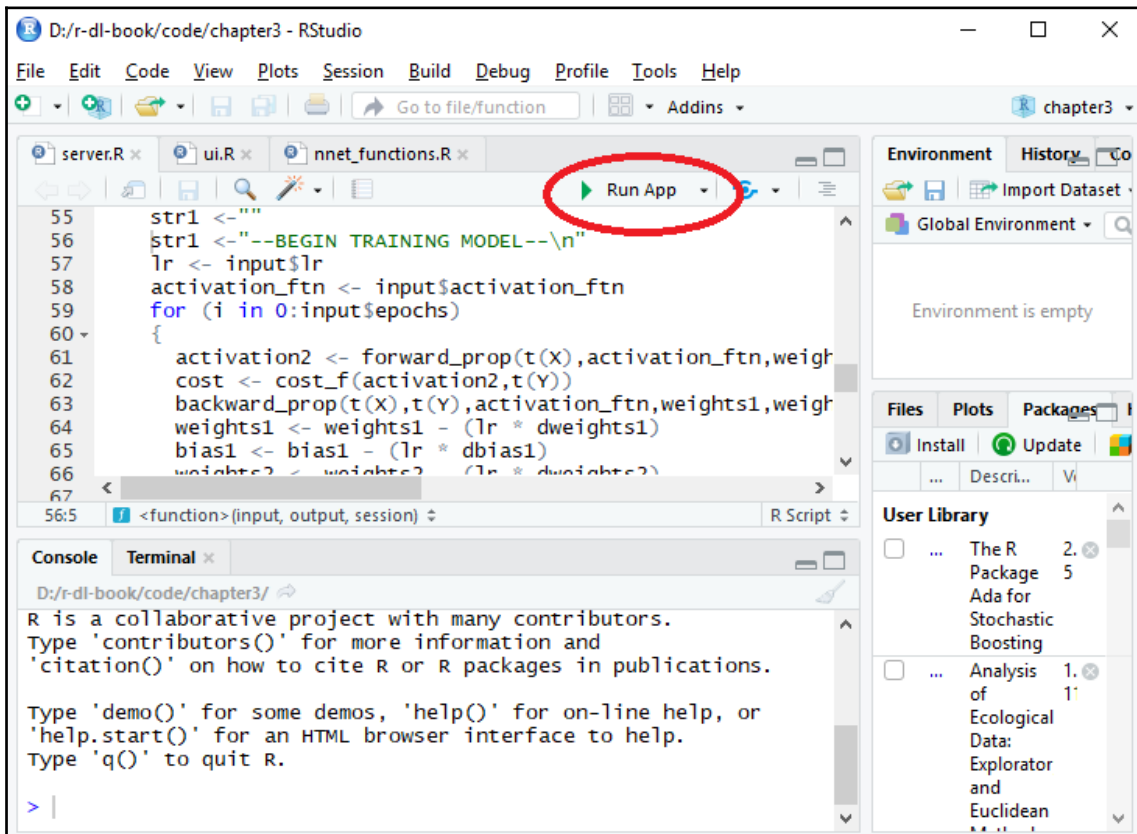


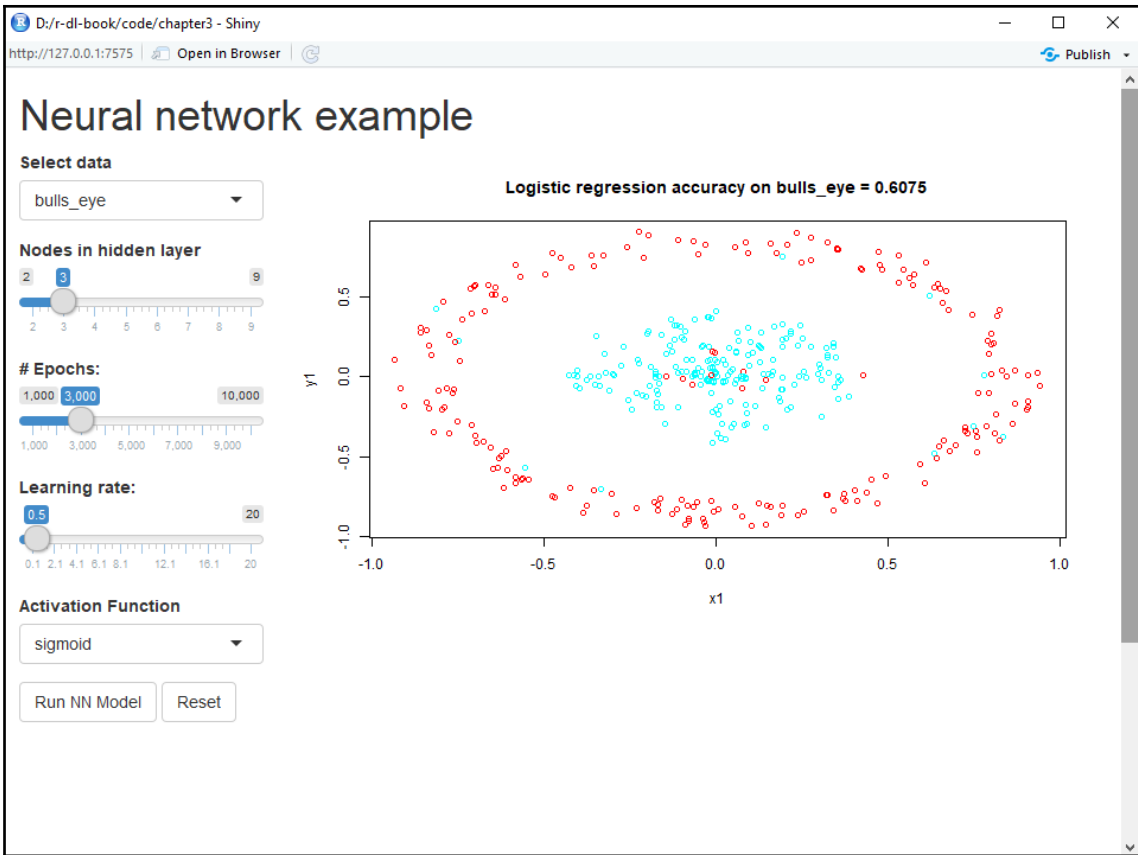


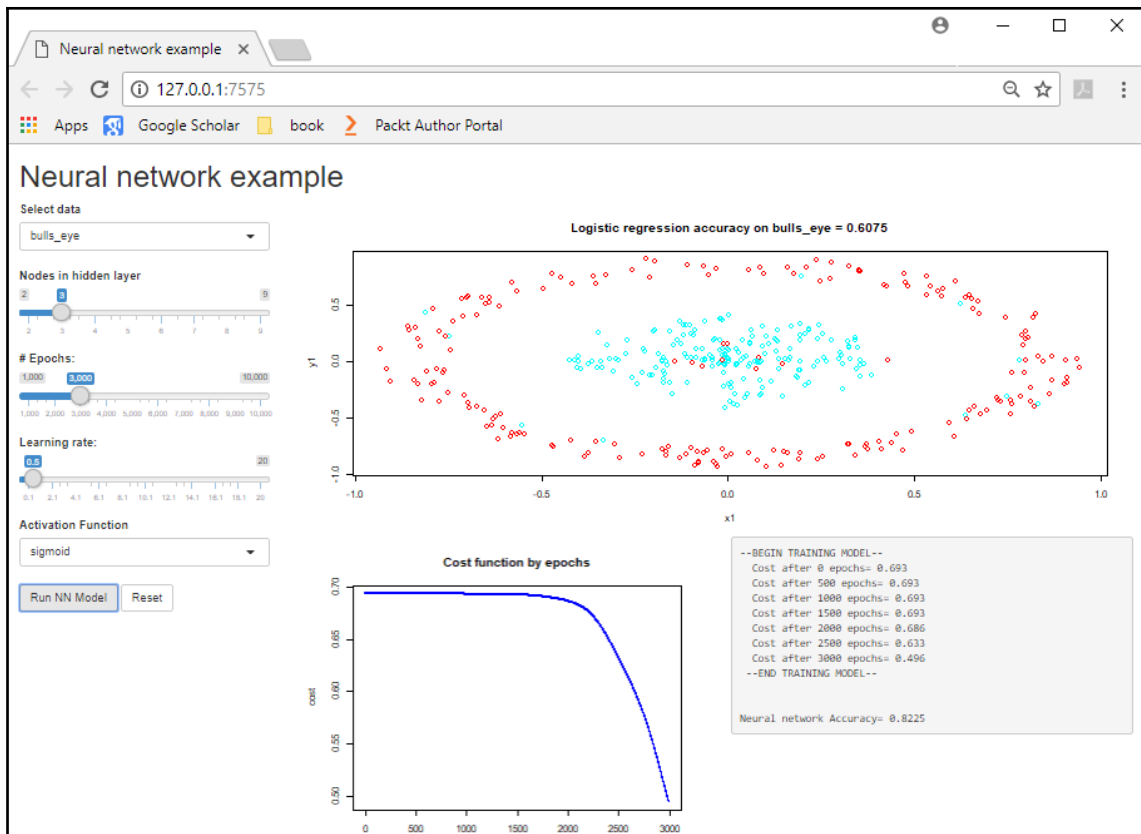


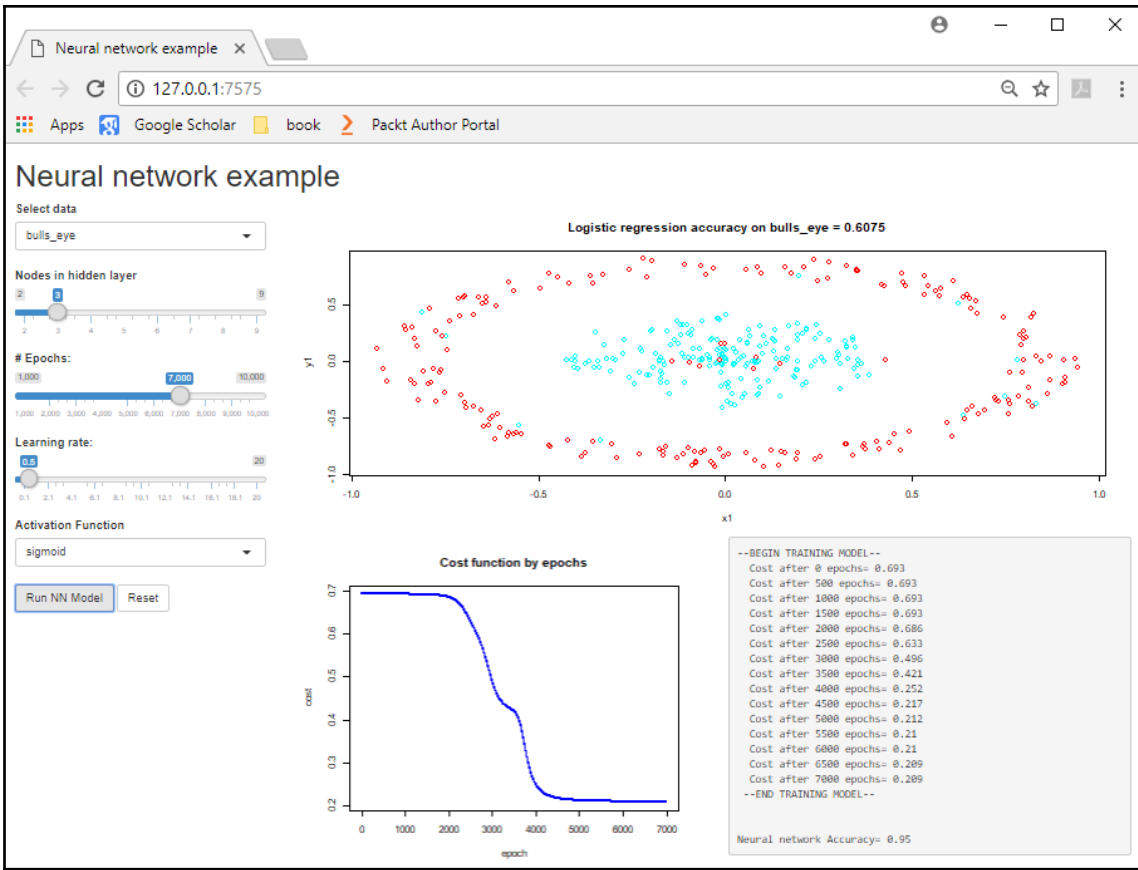


## Chapter 3: Deep Learning Fundamentals

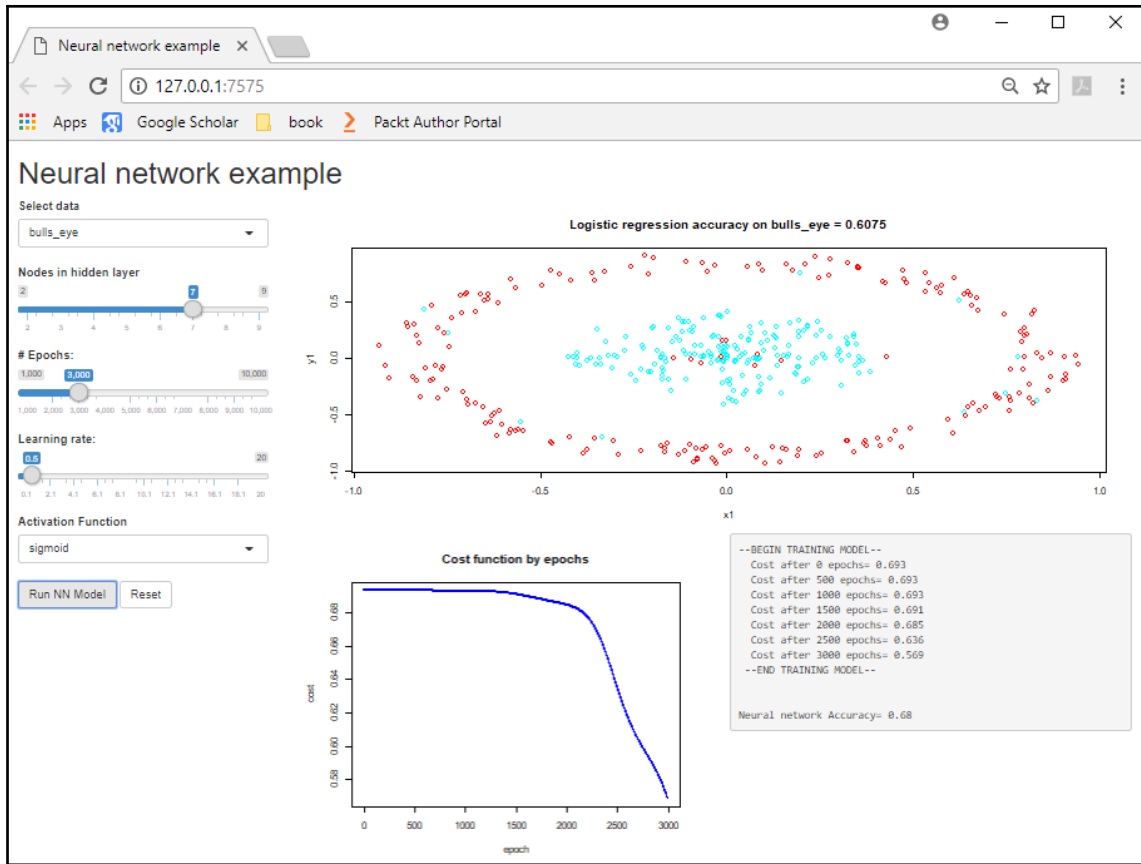


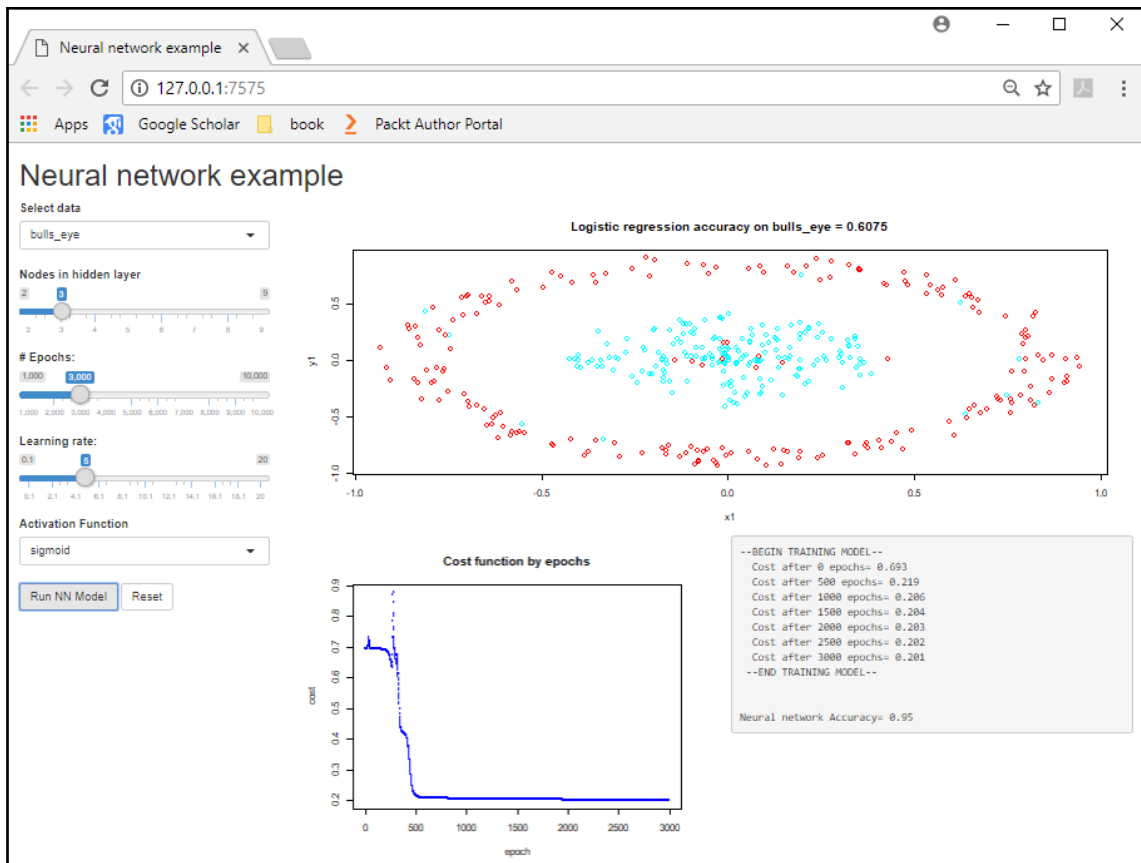


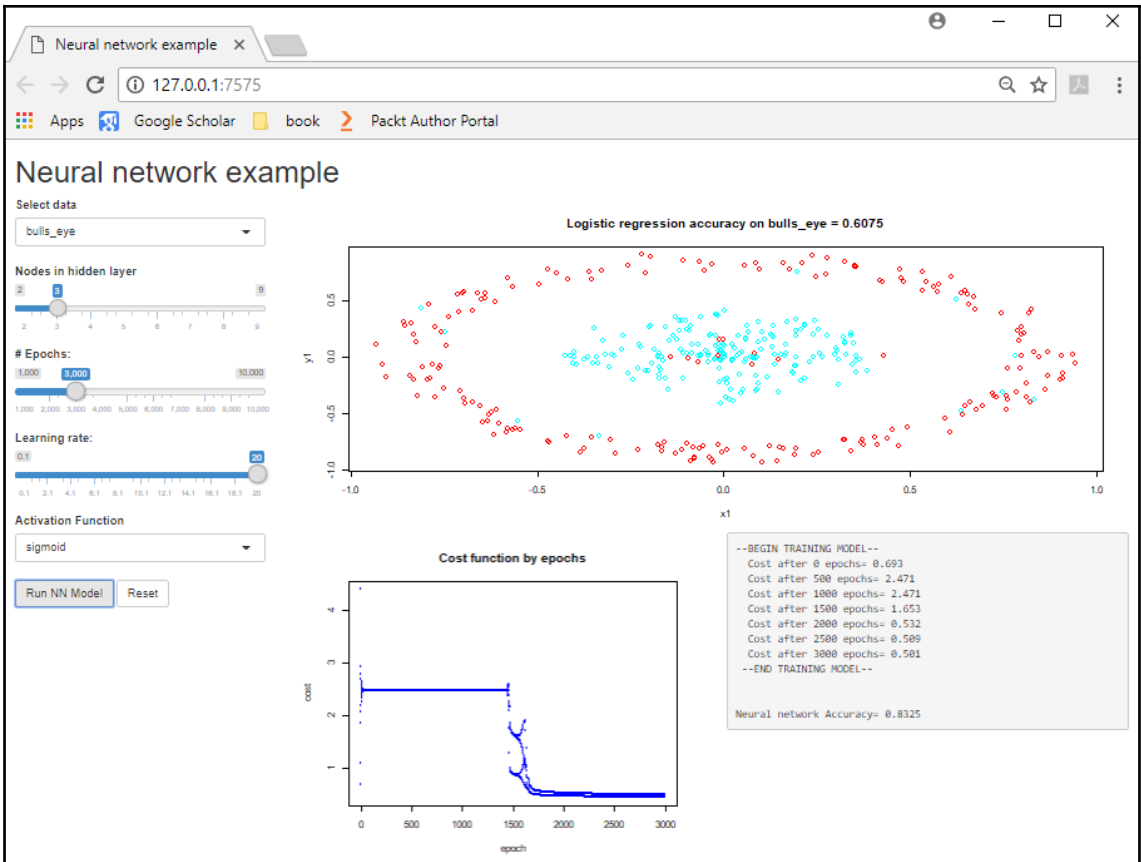


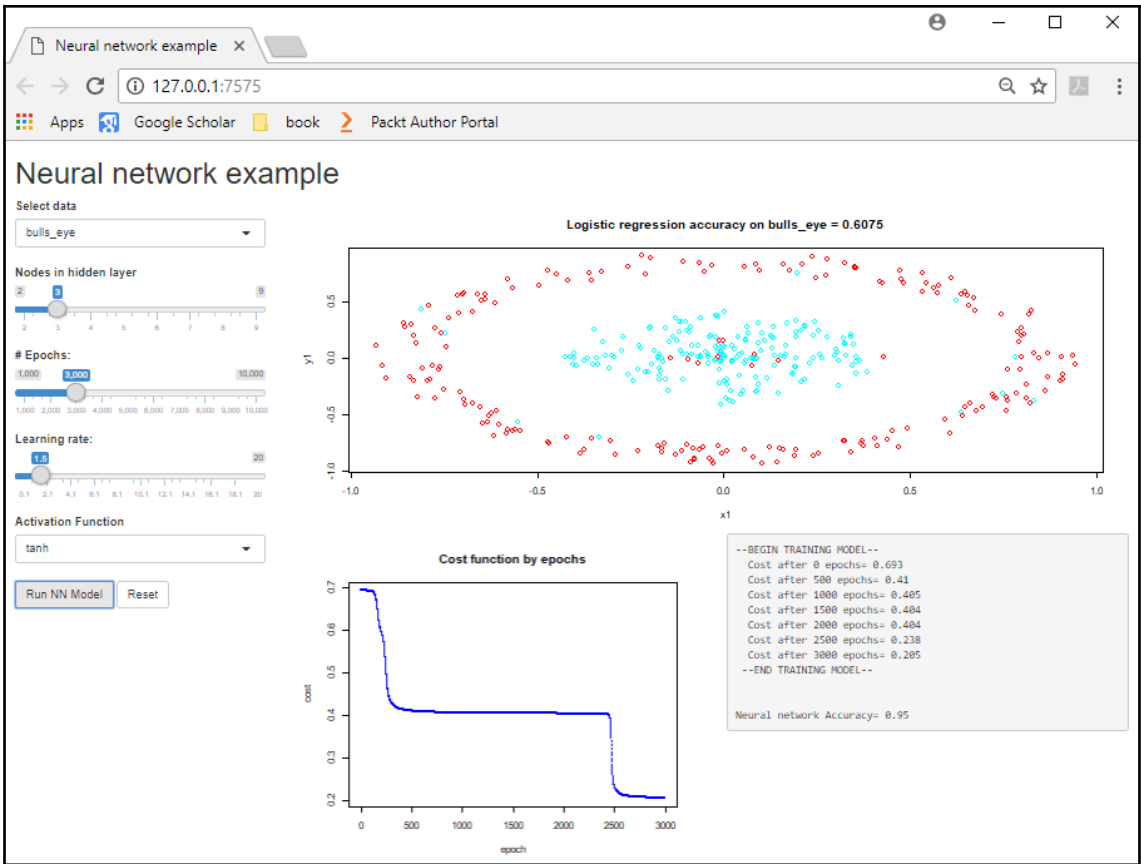


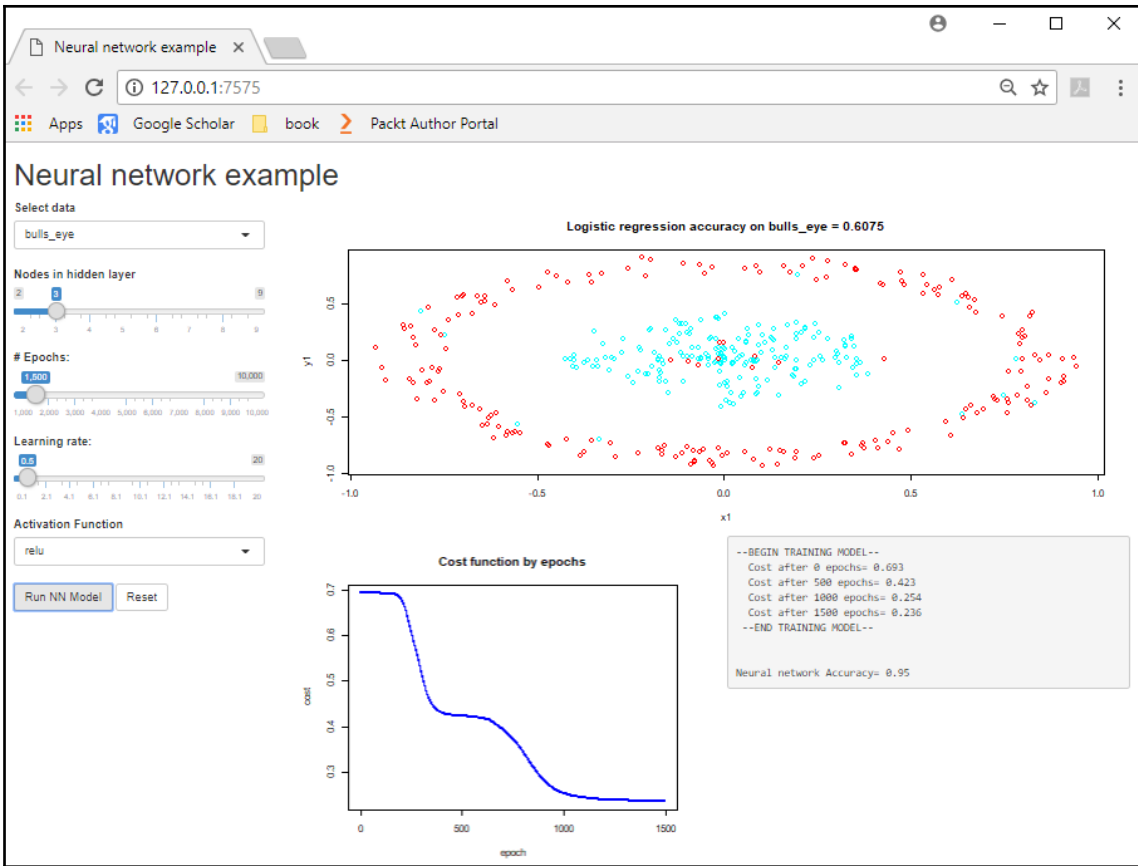


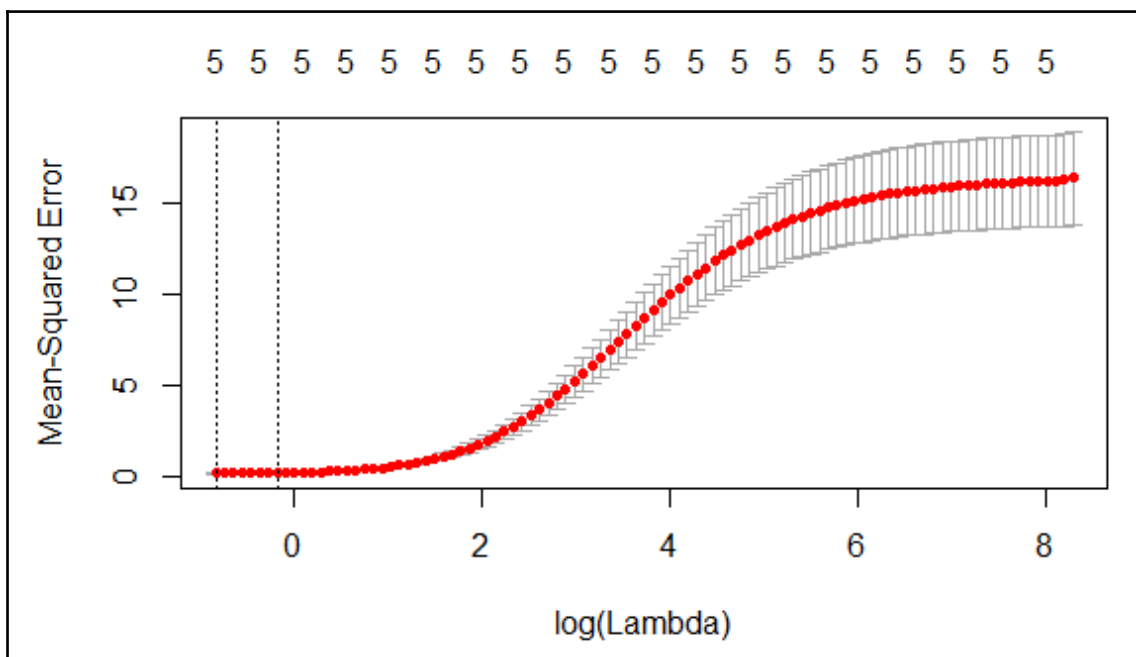
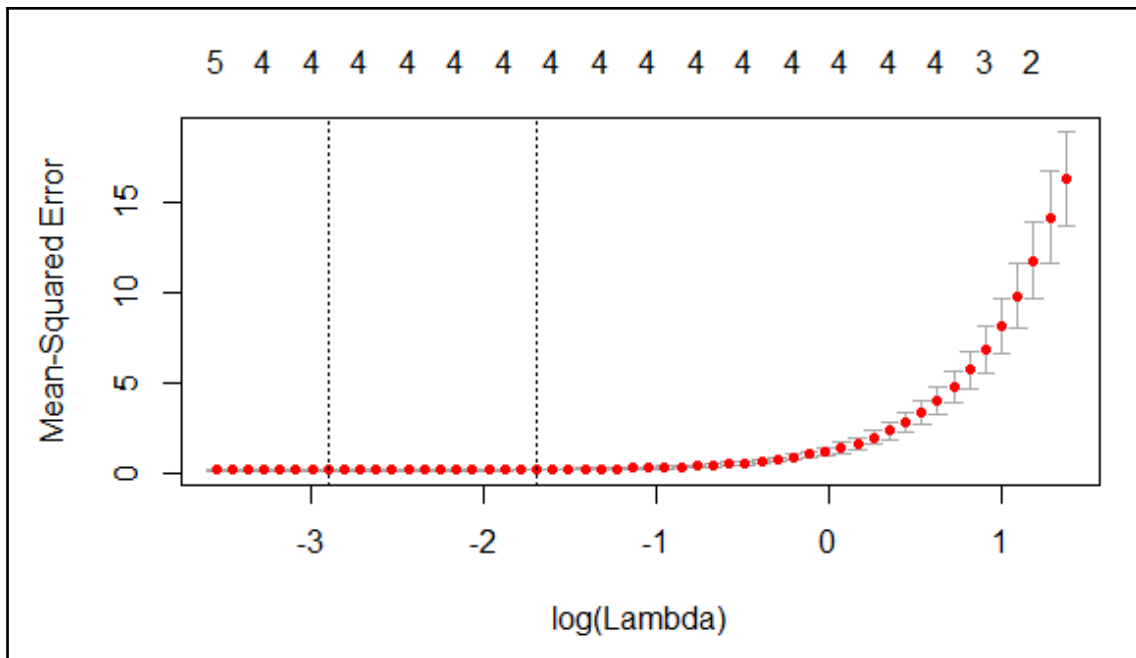


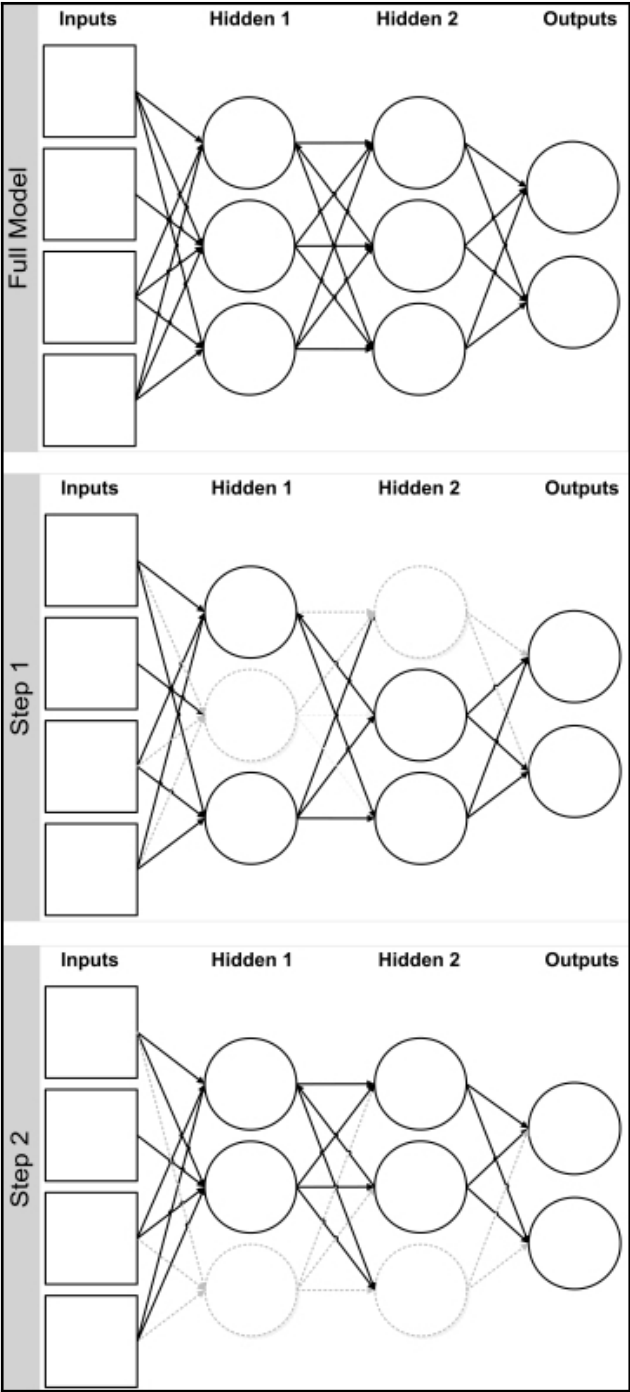




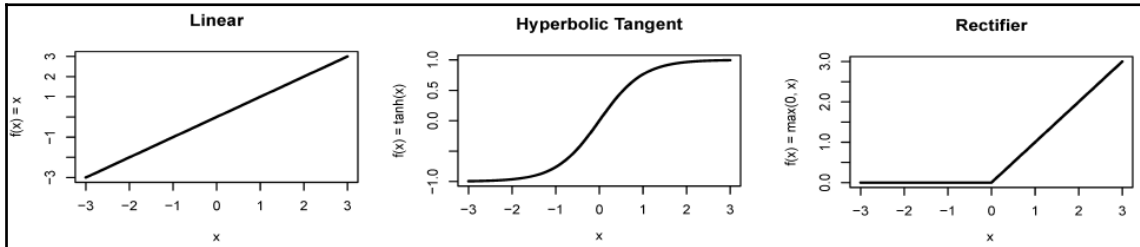
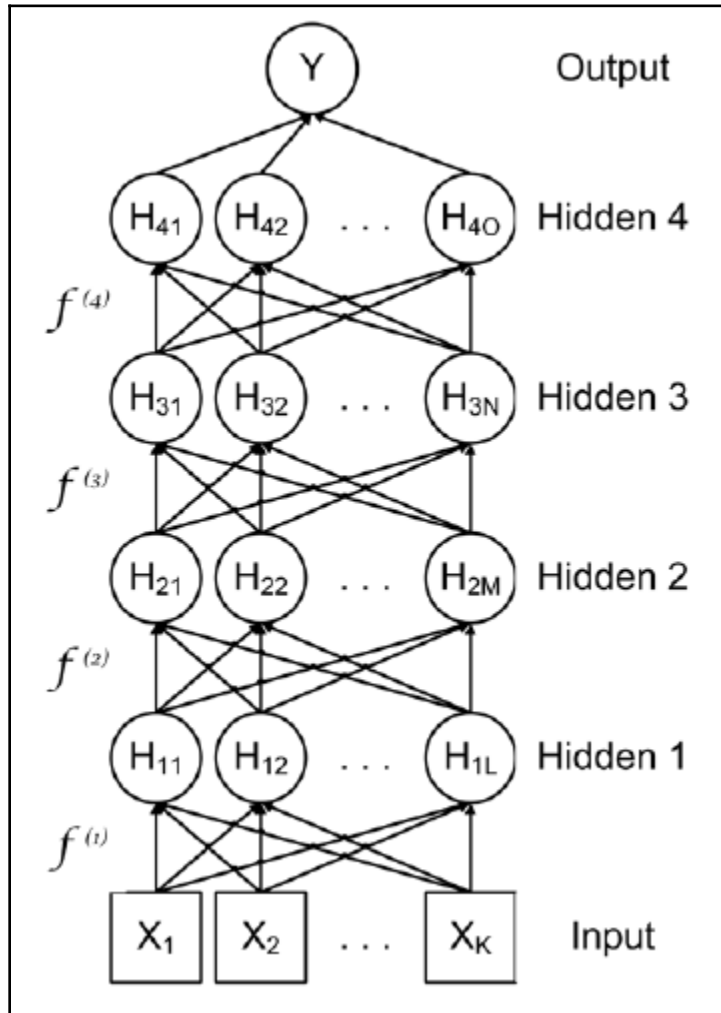




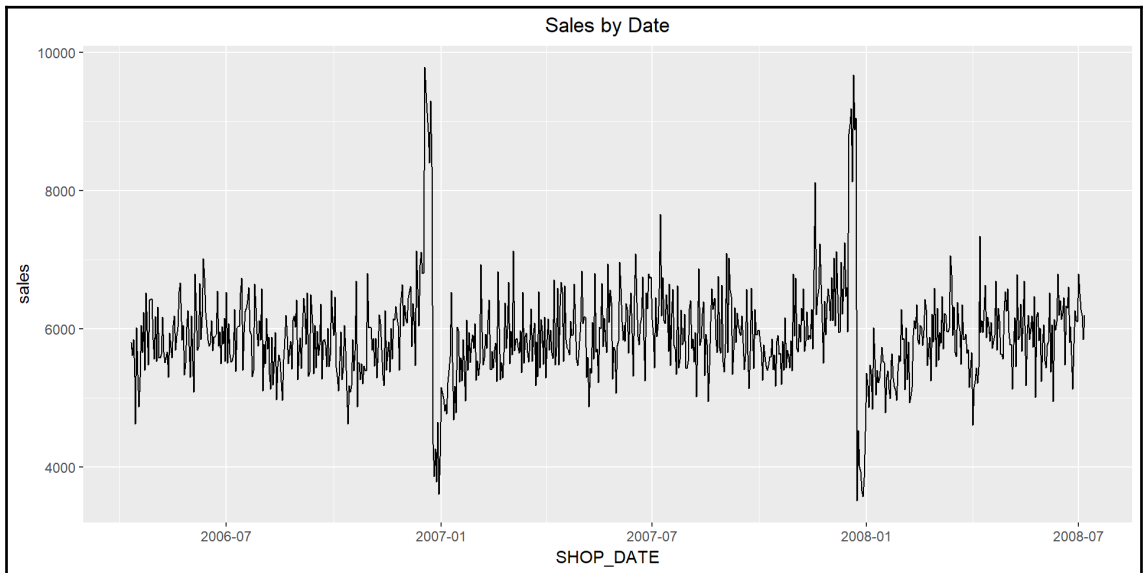
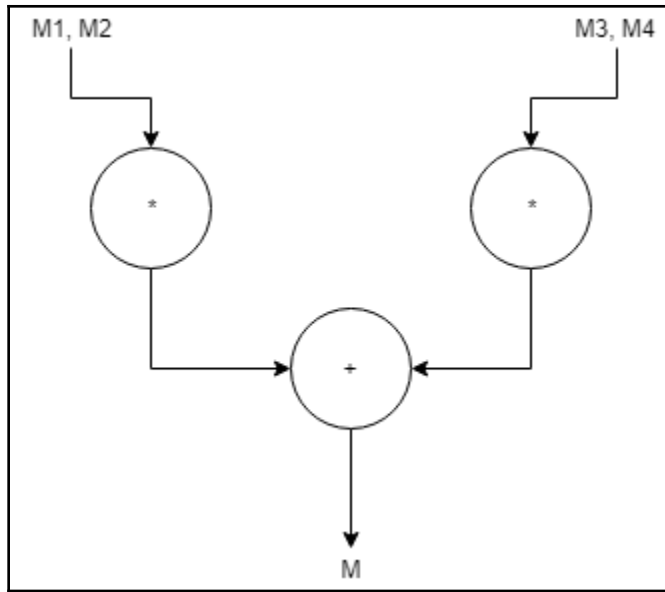


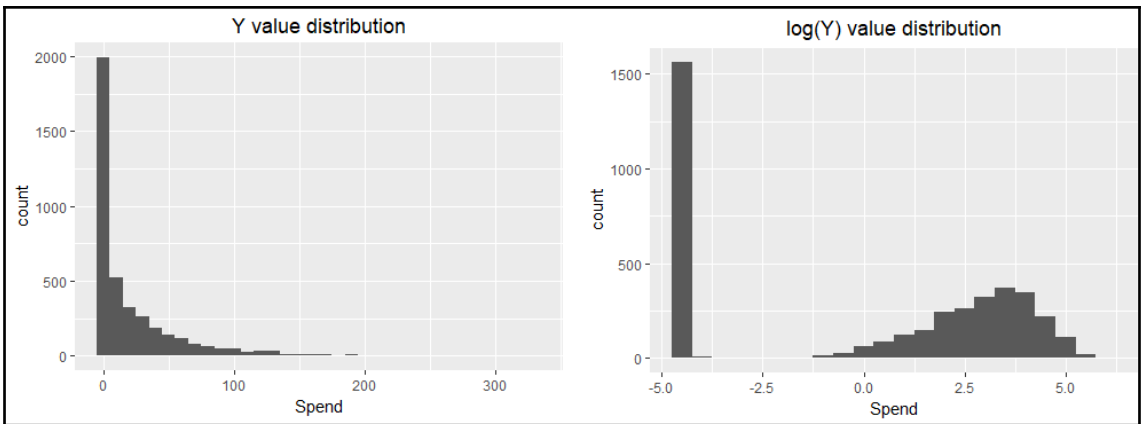
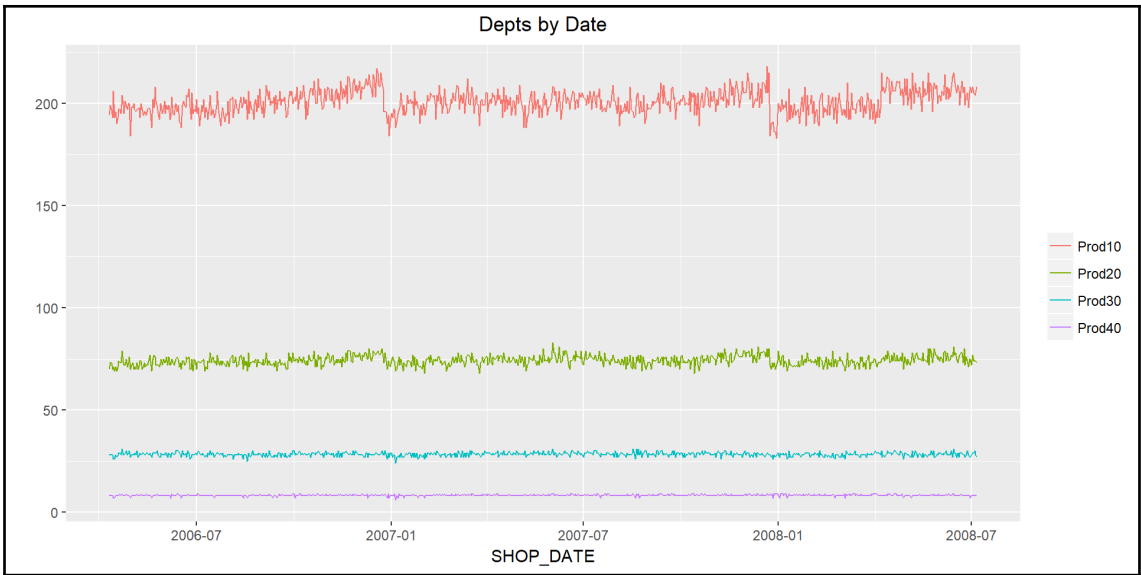


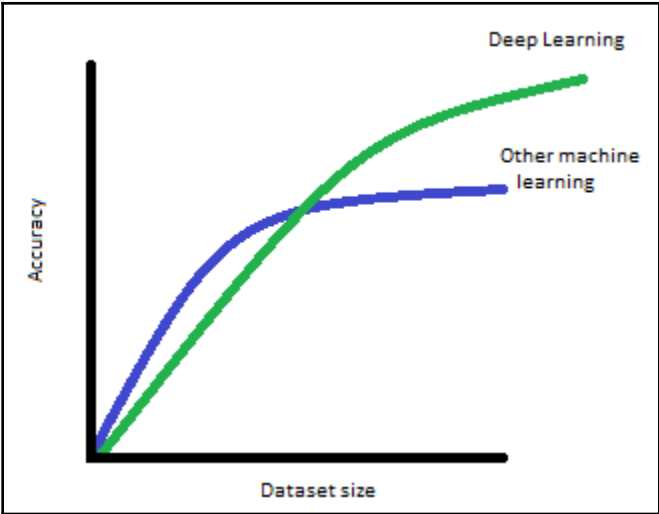
# Chapter 4: Training Deep Prediction Models



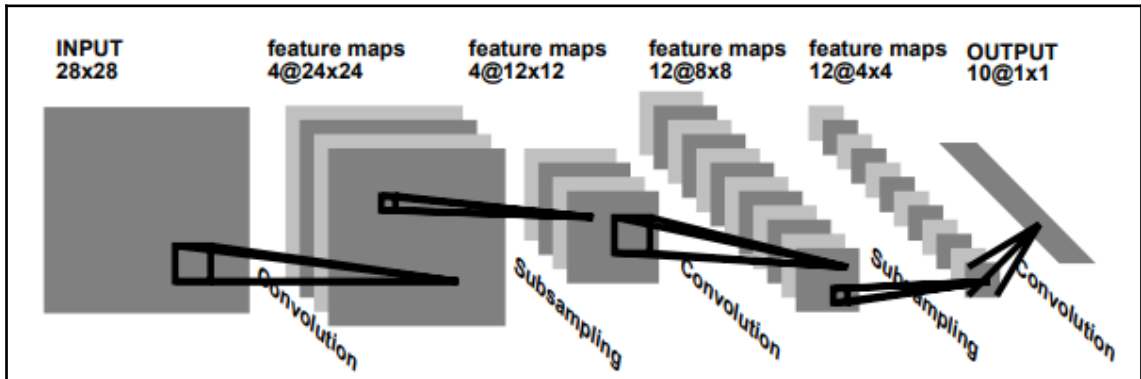


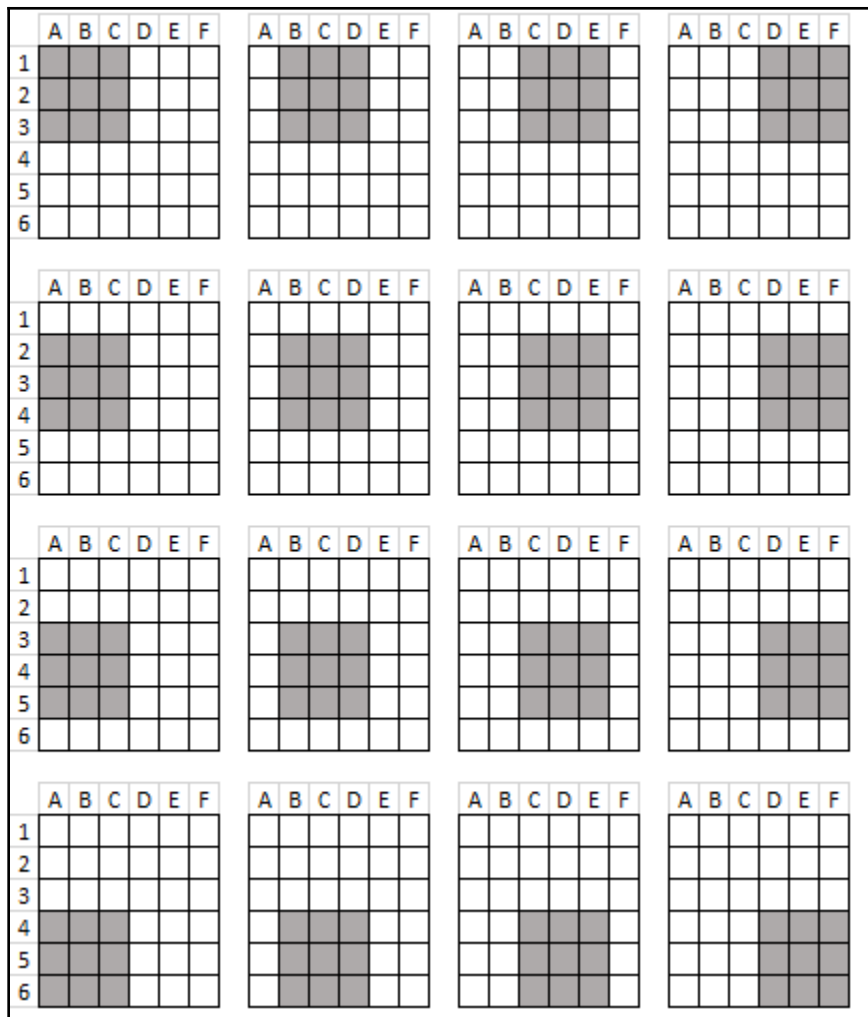


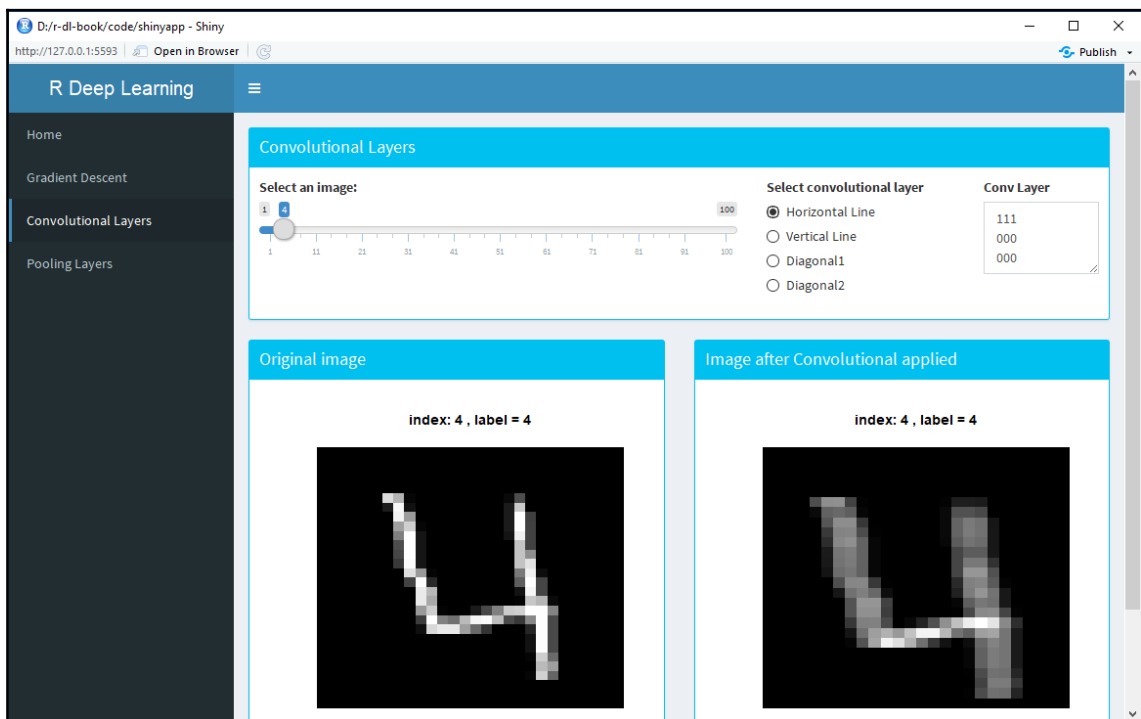
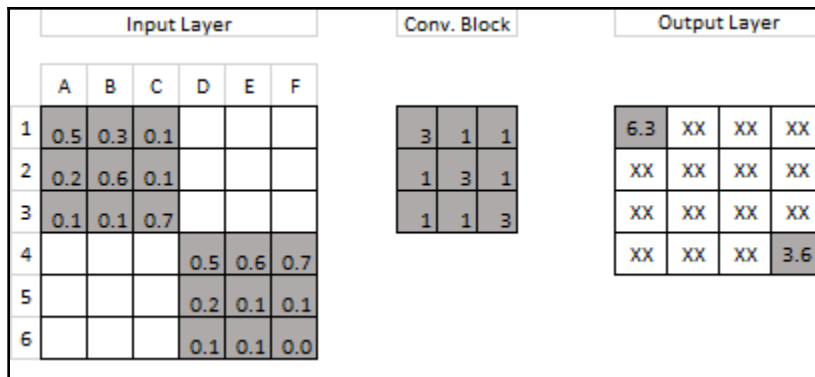


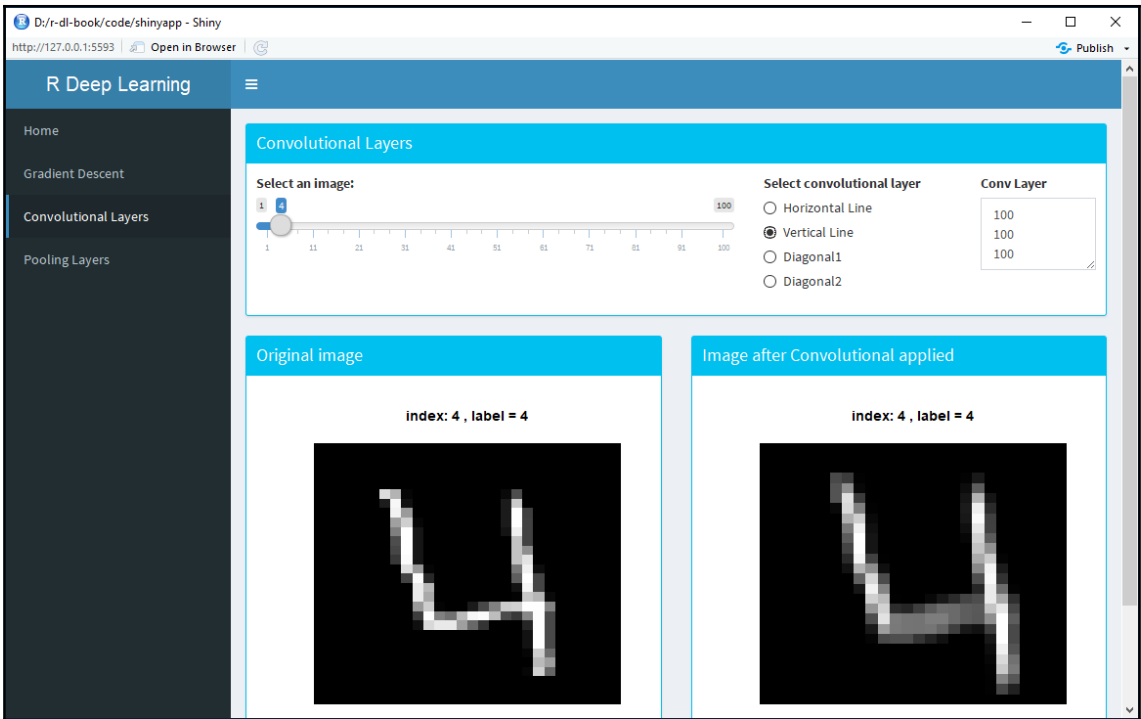


# Chapter 5: Image Classification using Convolutional Neural Networks

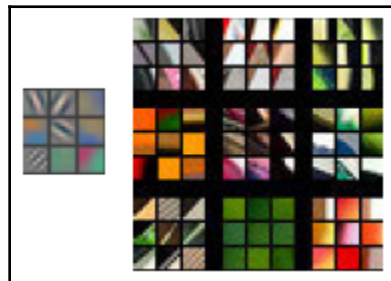


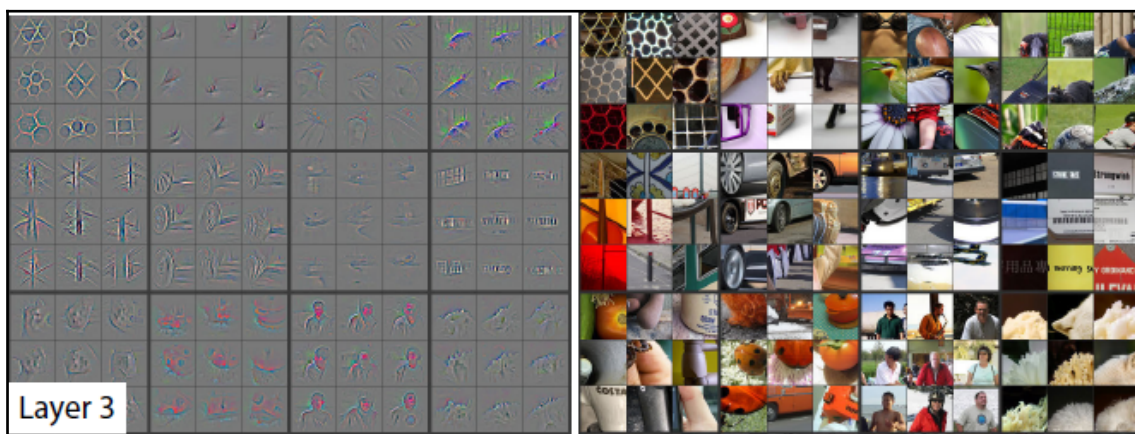
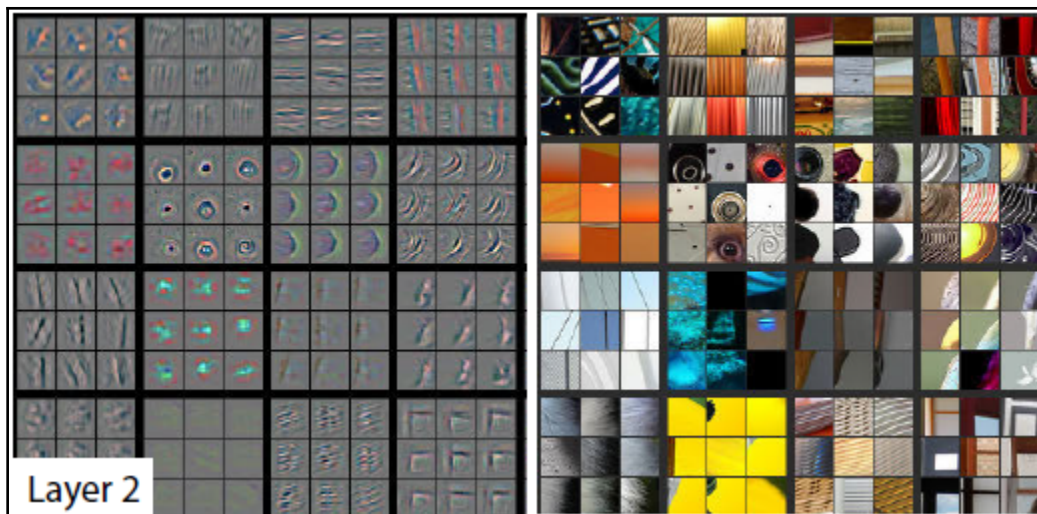






	A	B	C	D	E	F	
	0	0	0	0	0	0	0
1	0						0
2	0						0
3	0						0
4	0						0
5	0						0
6	0						0
	0	0	0	0	0	0	0



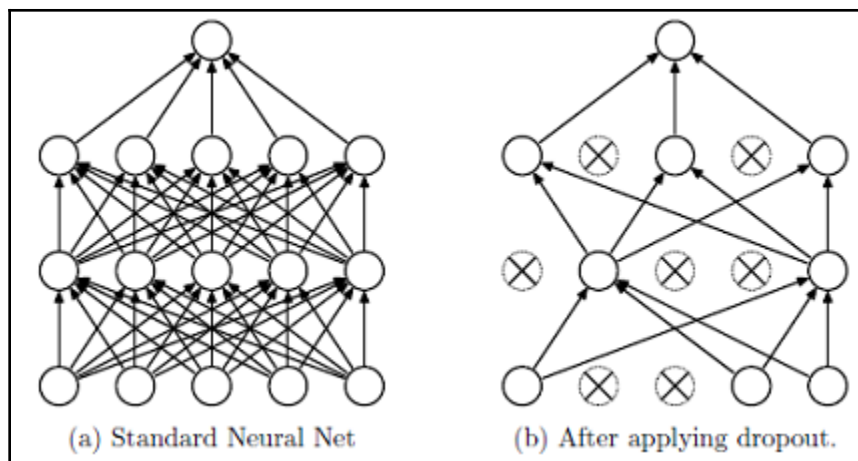


	A	B	C	D	E	F
1	7	0	8	6	2	1
2	6	6	0	8	4	2
3	2	2	1	2	3	6
4	5	0	2	2	2	5
5	4	2	2	1	9	7
6	7	2	0	5	2	4

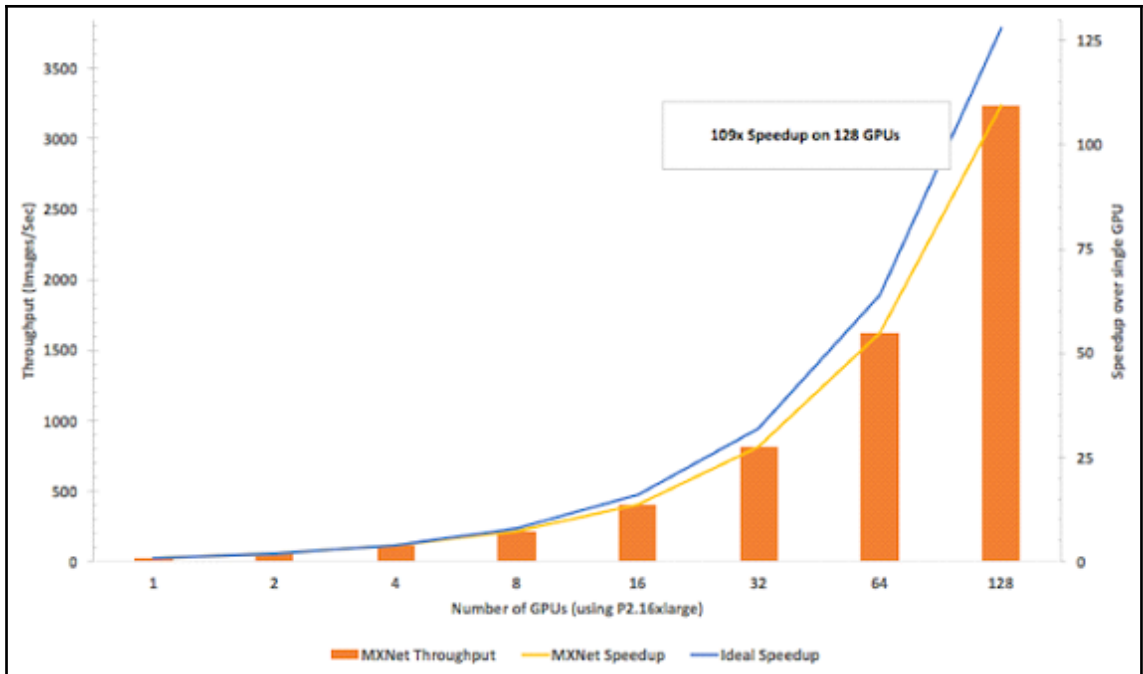
  

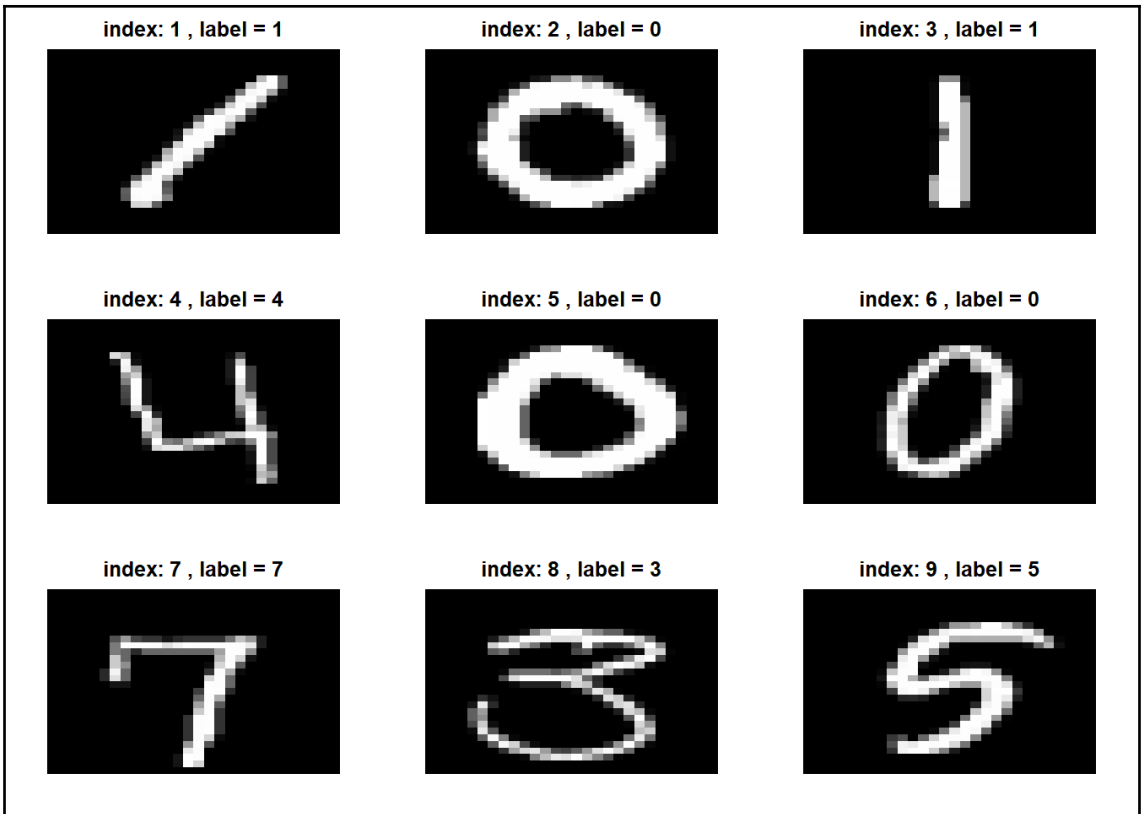
7	8	4
5	2	6
7	5	9

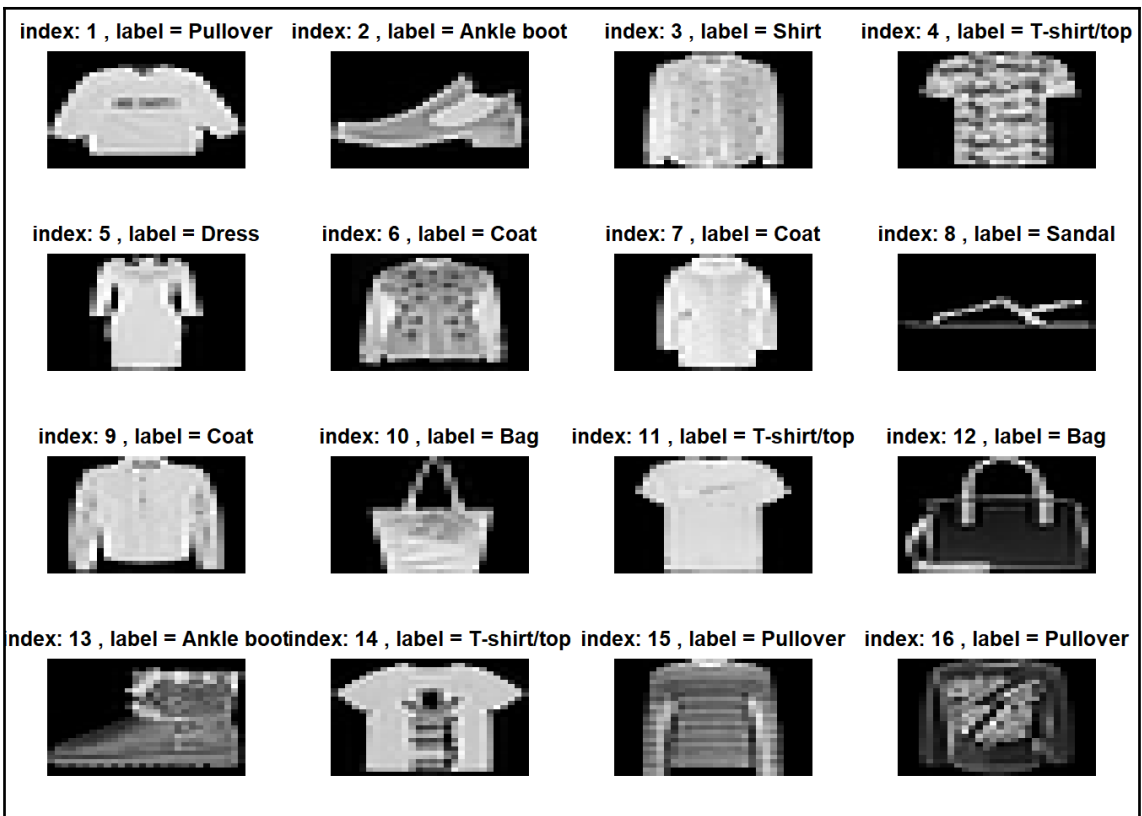


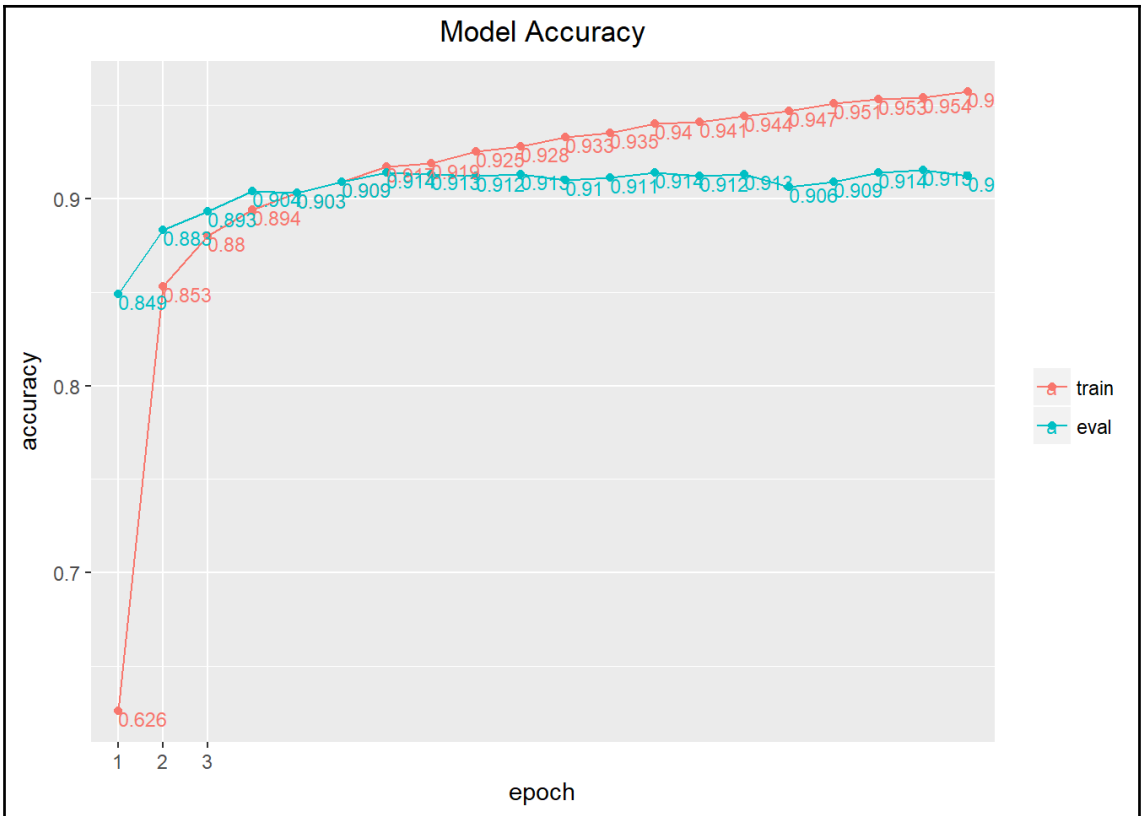


Category	Output from Dense Layer (x)	$e^x$	Output Probability
0	-1.3	0.27	0.00
1	5.2	181.27	0.00
2	8.3	4,023.87	0.00
3	11.2	73,130.44	0.00
4	10.1	24,343.01	0.00
5	17.2	29,502,925.92	0.78
6	15.8	7,275,331.96	0.19
7	5.2	181.27	0.00
8	3.1	22.20	0.00
9	13.5	729,416.37	0.02
		37,609,556.58	

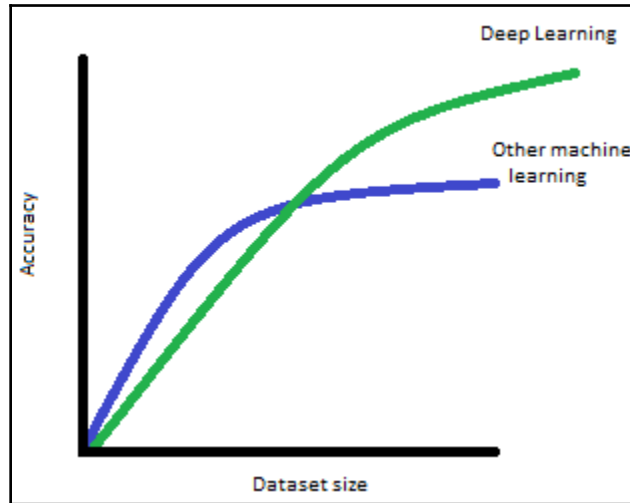
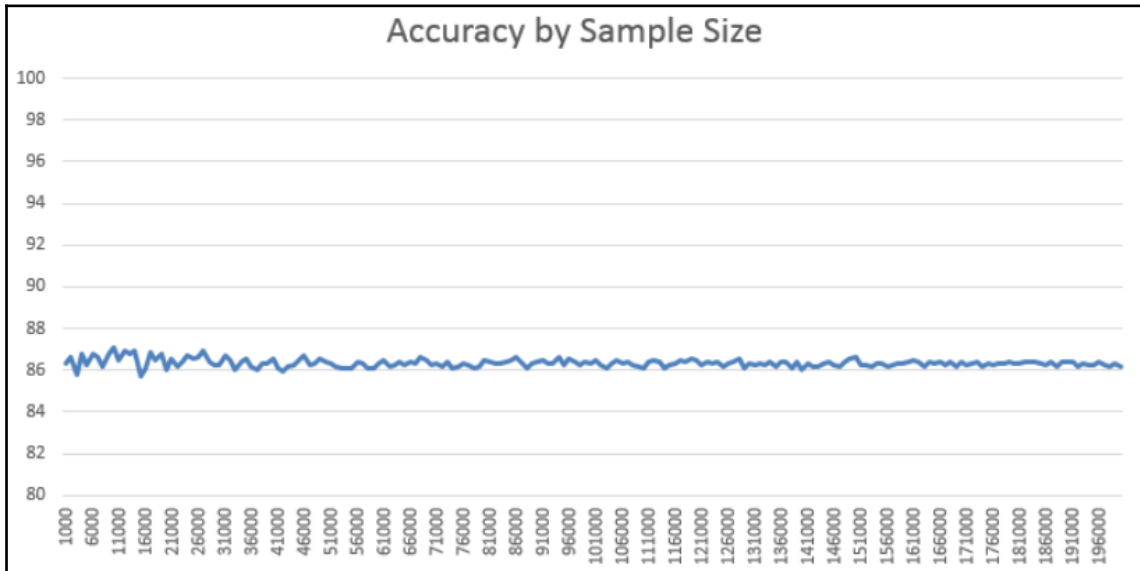


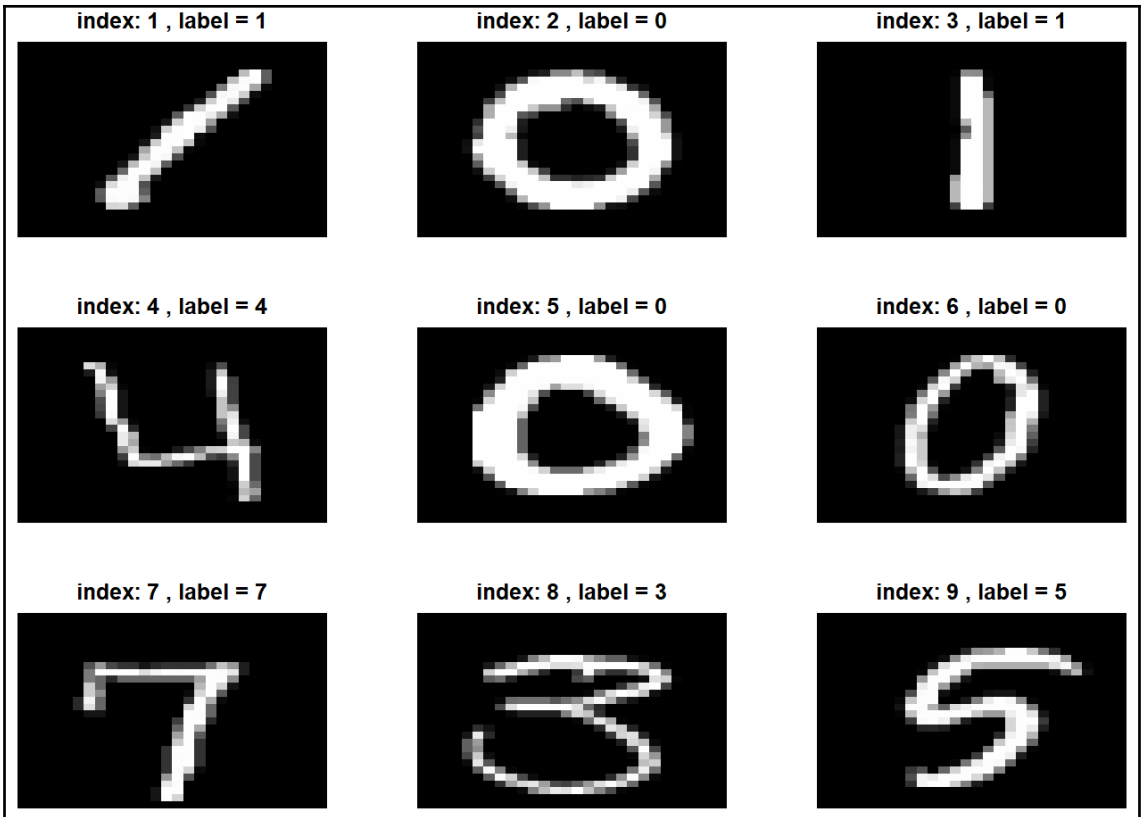


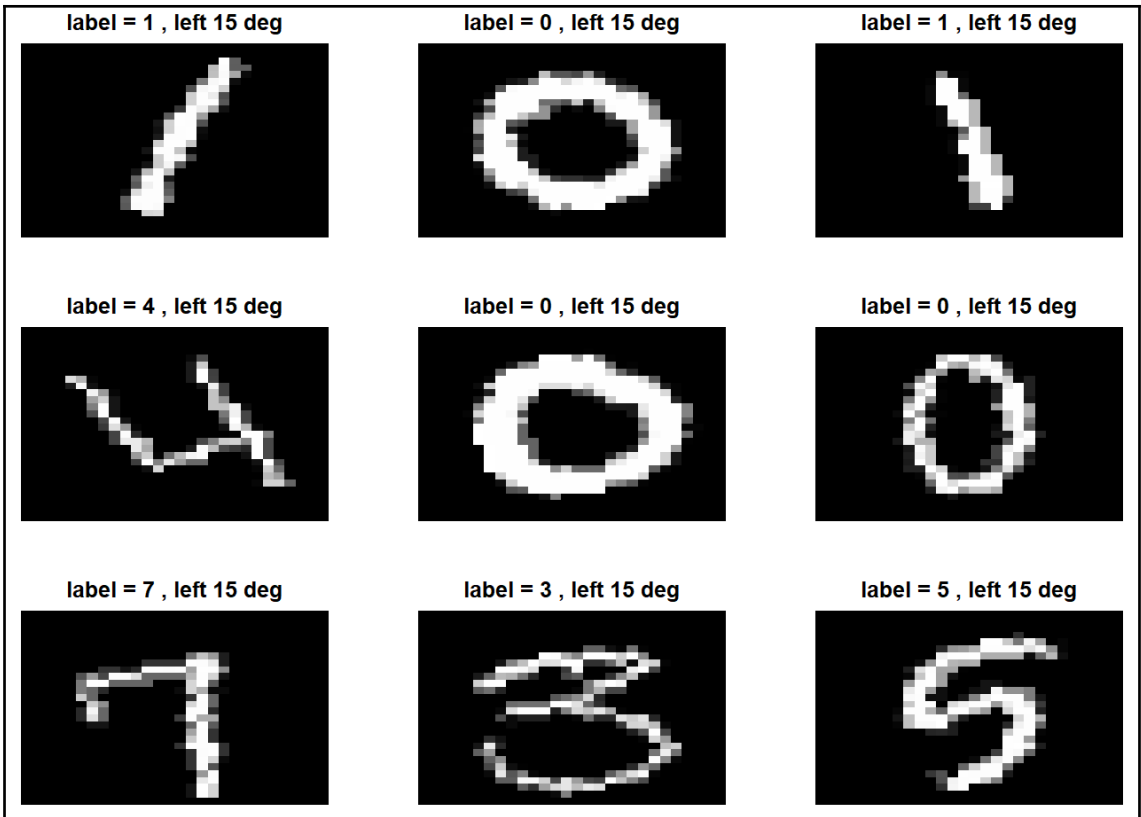




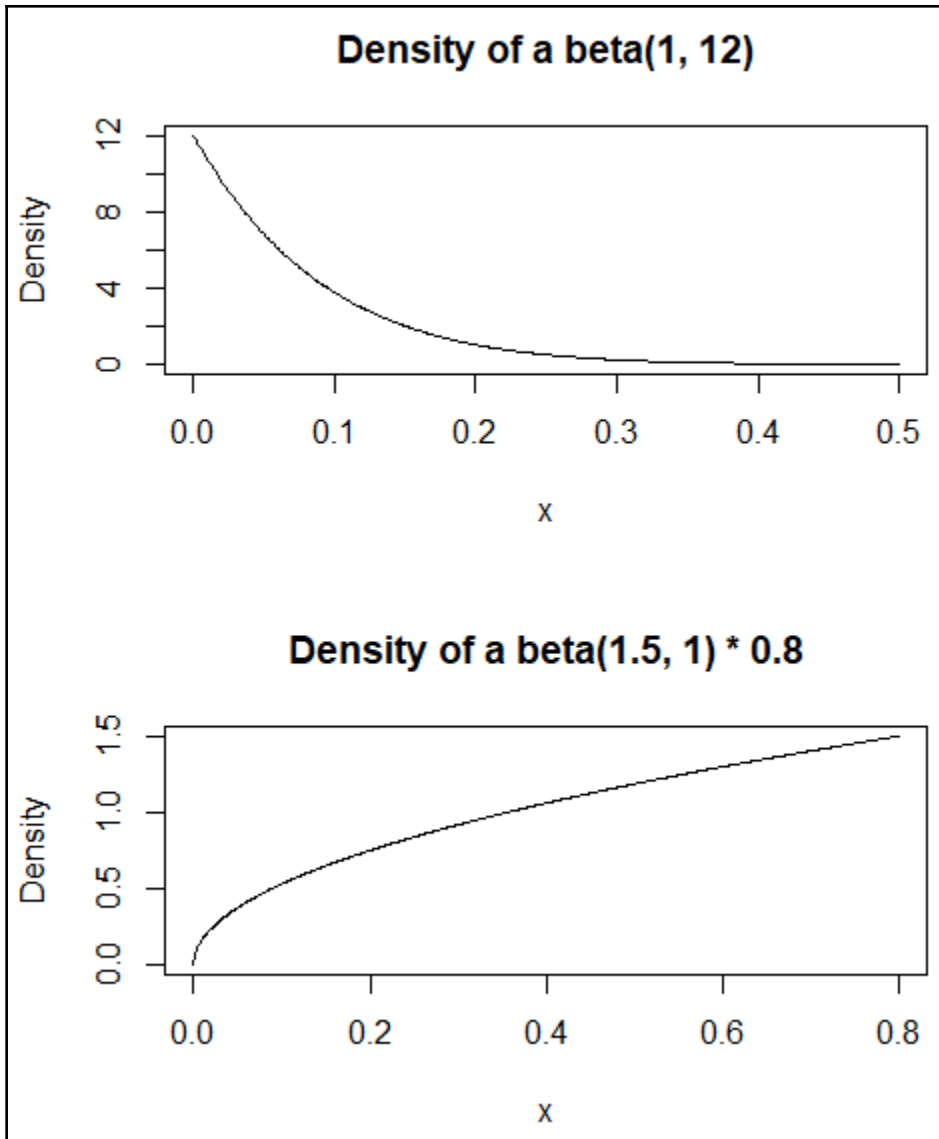
# Chapter 6: Tuning and Optimizing Models.

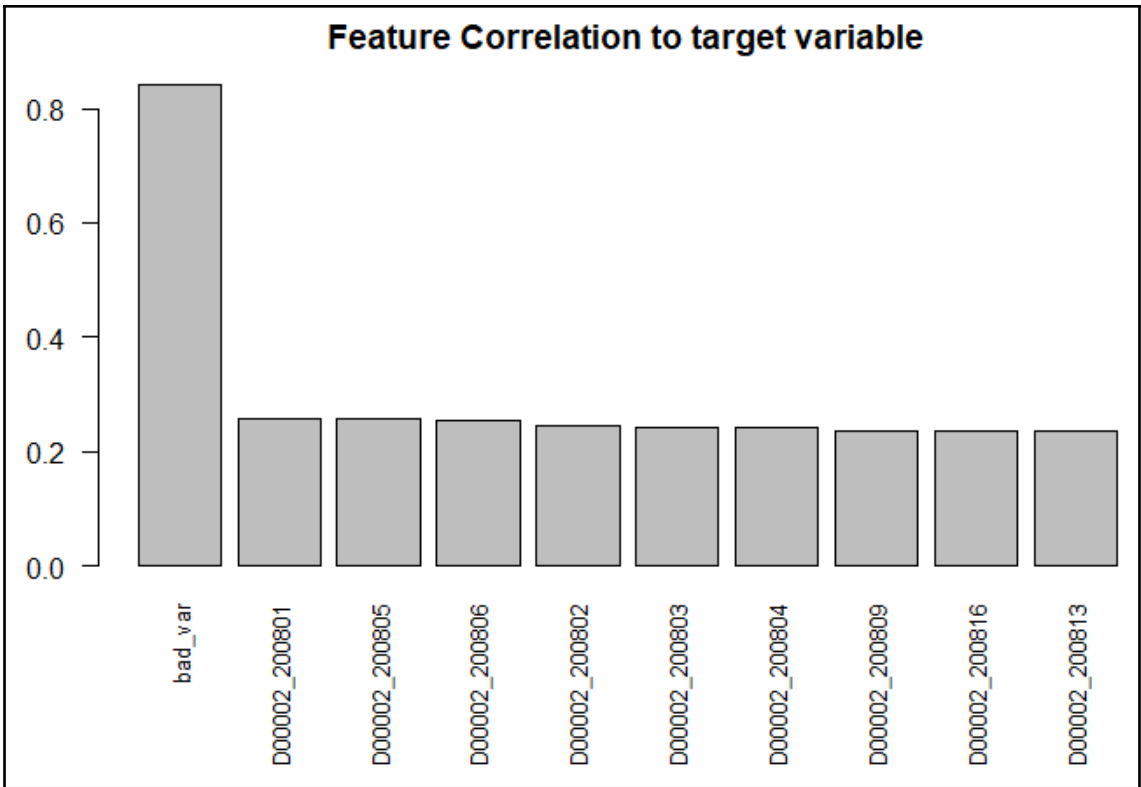


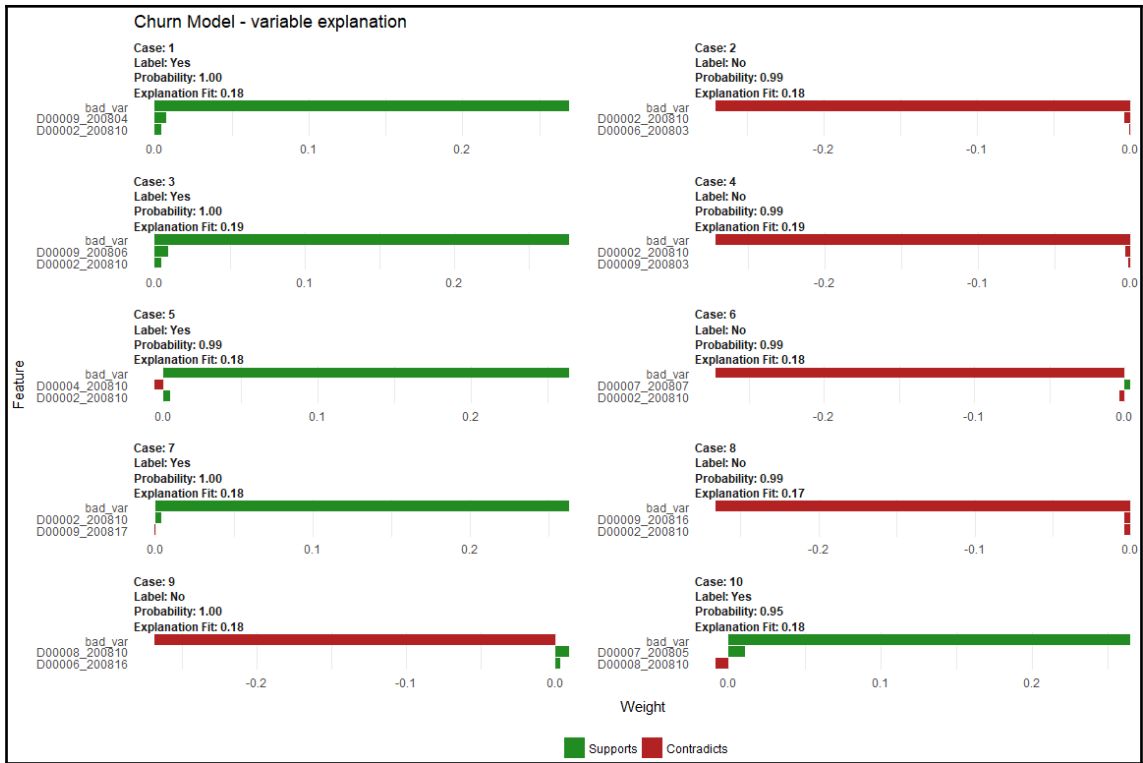


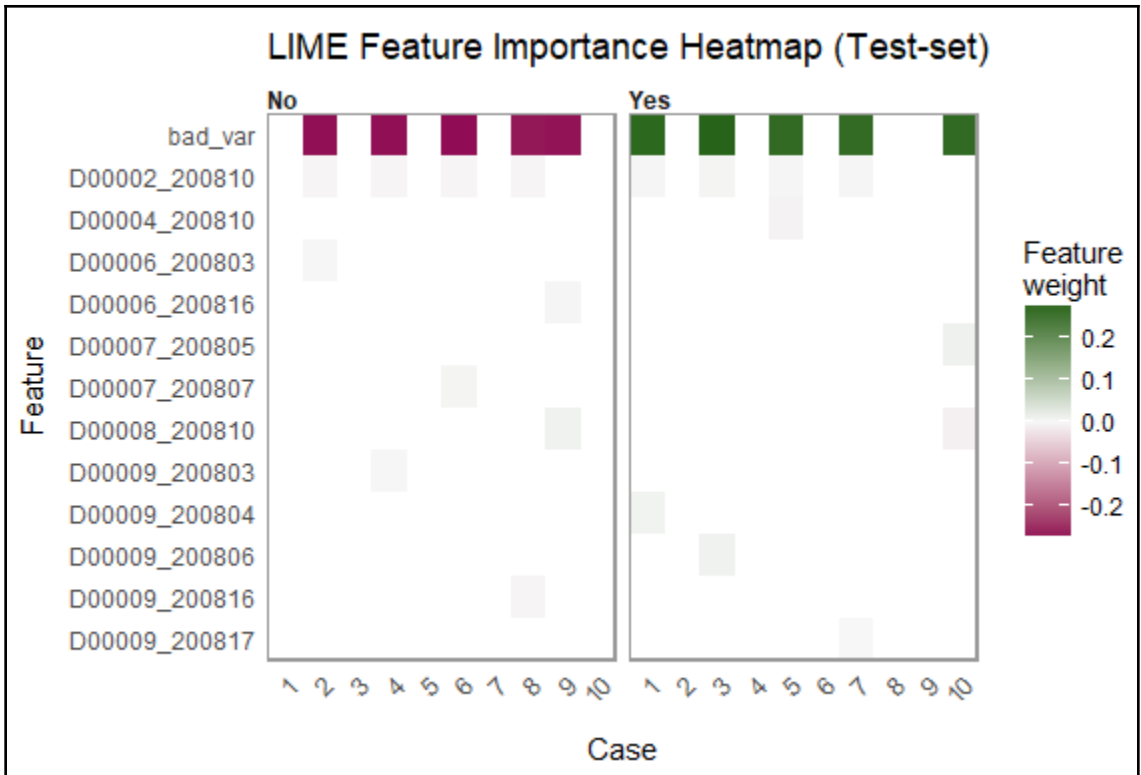


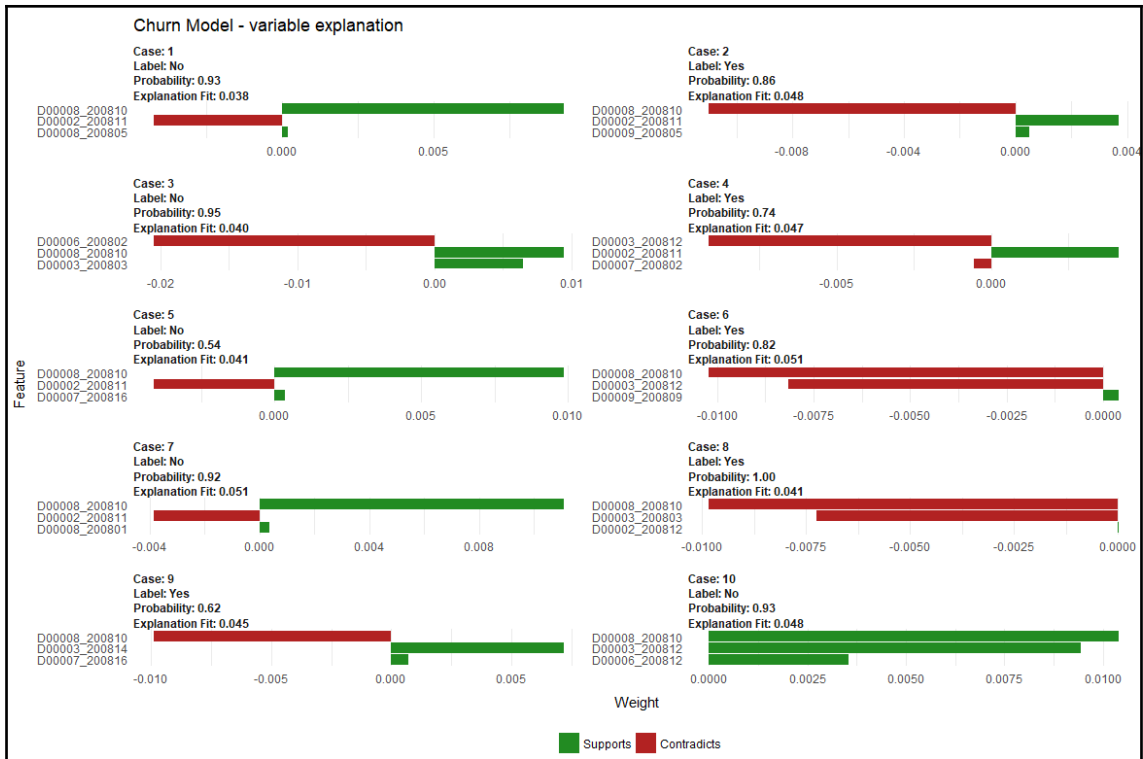


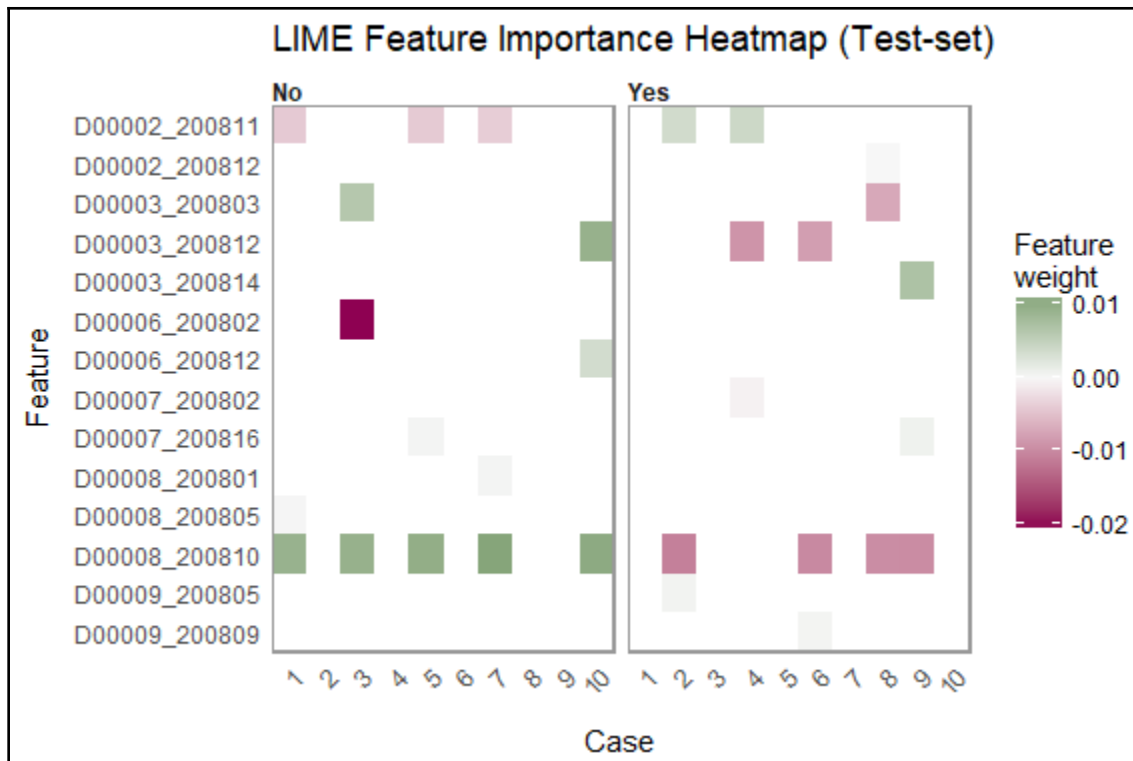




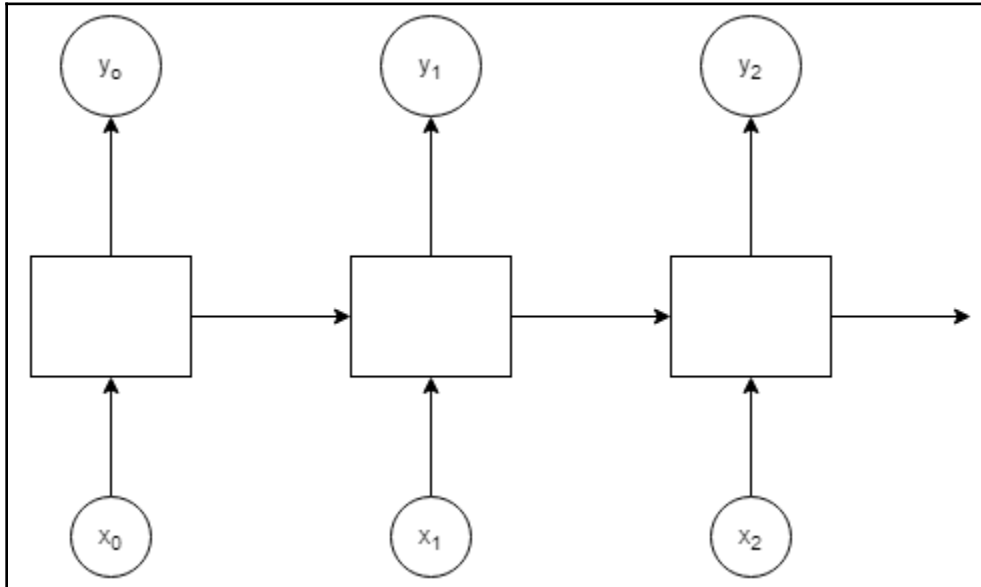




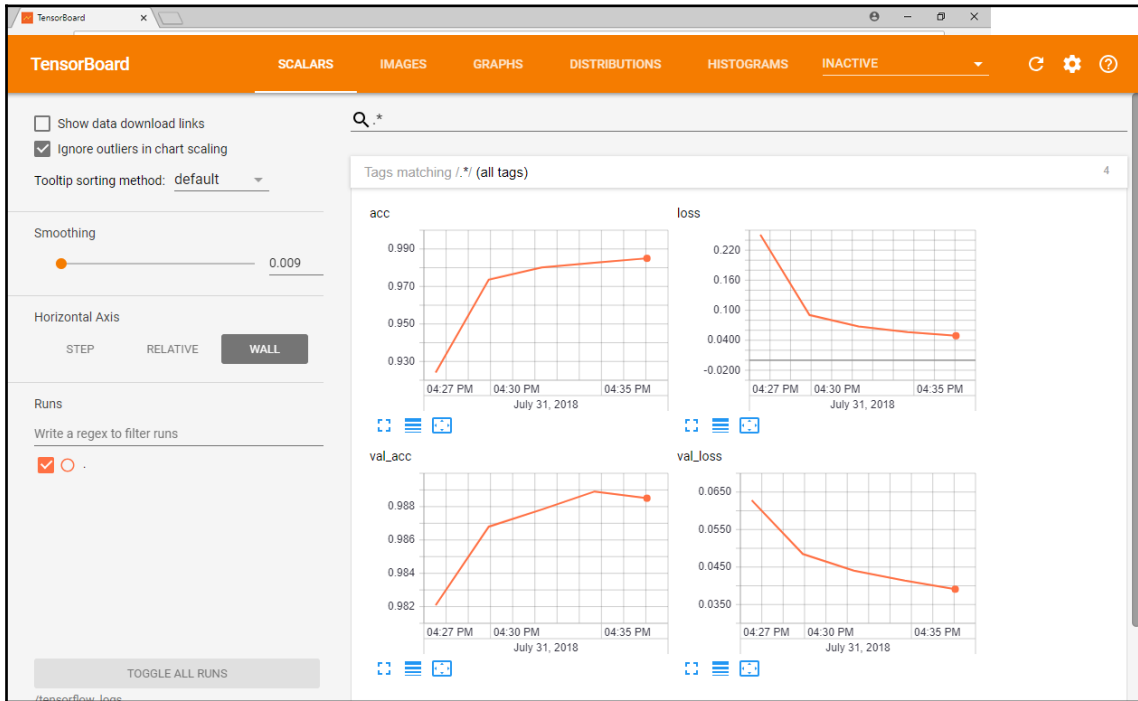




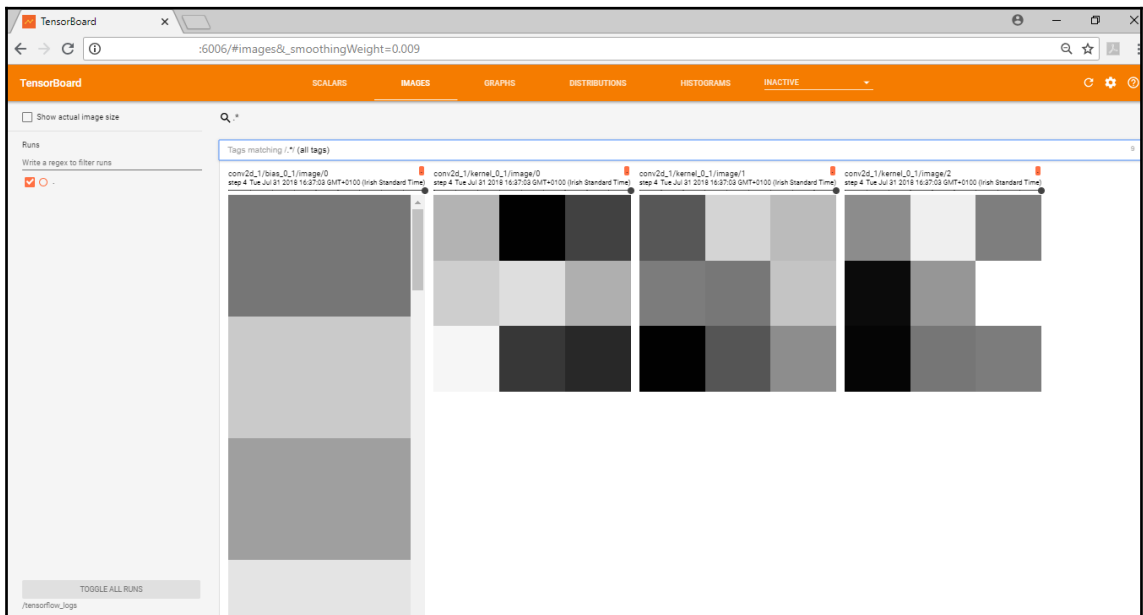
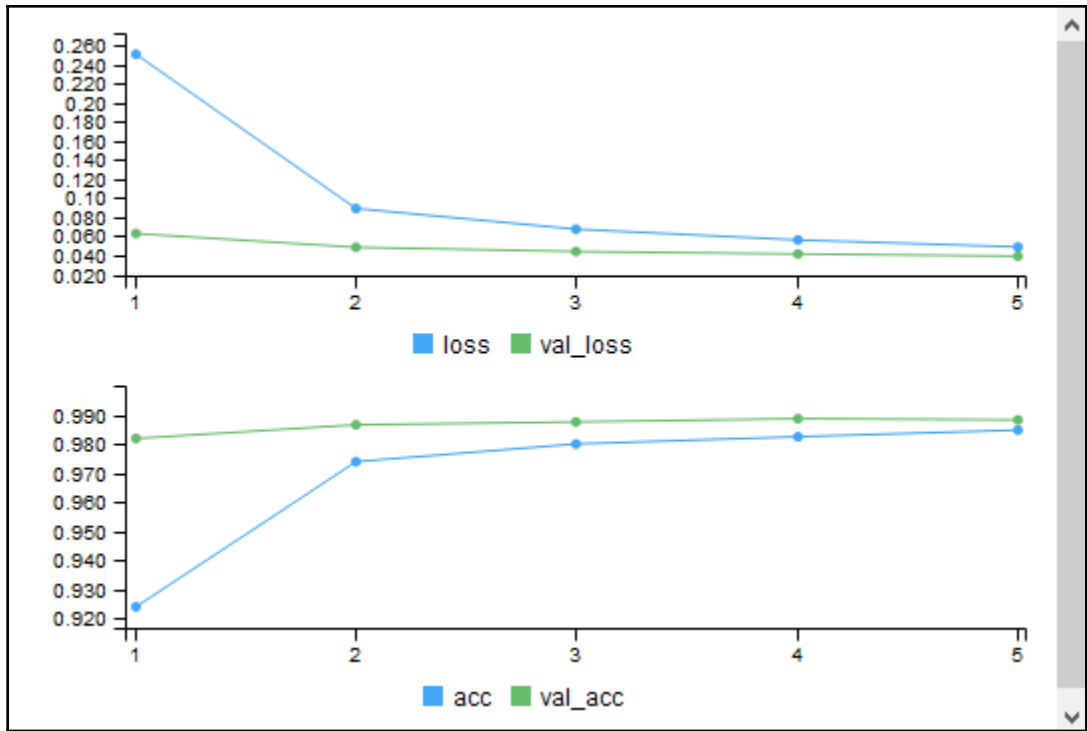
# Chapter 7: Natural Language Processing using Recurrent Neural Networks

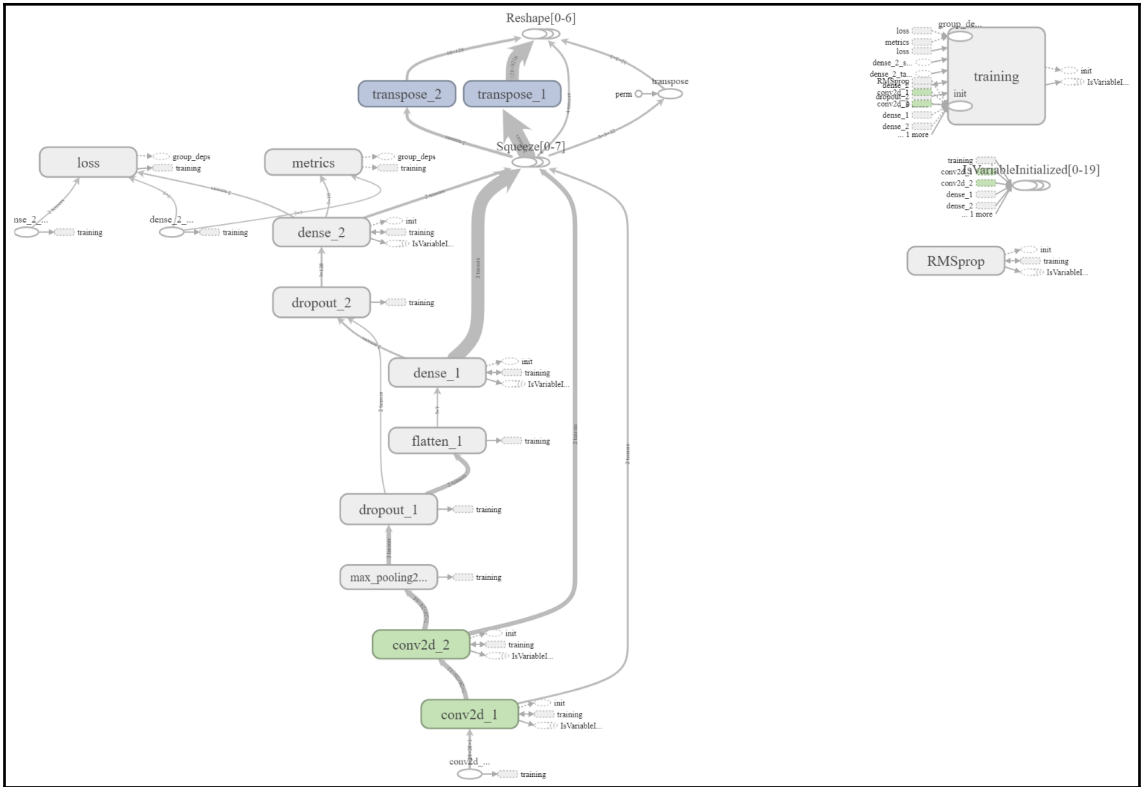


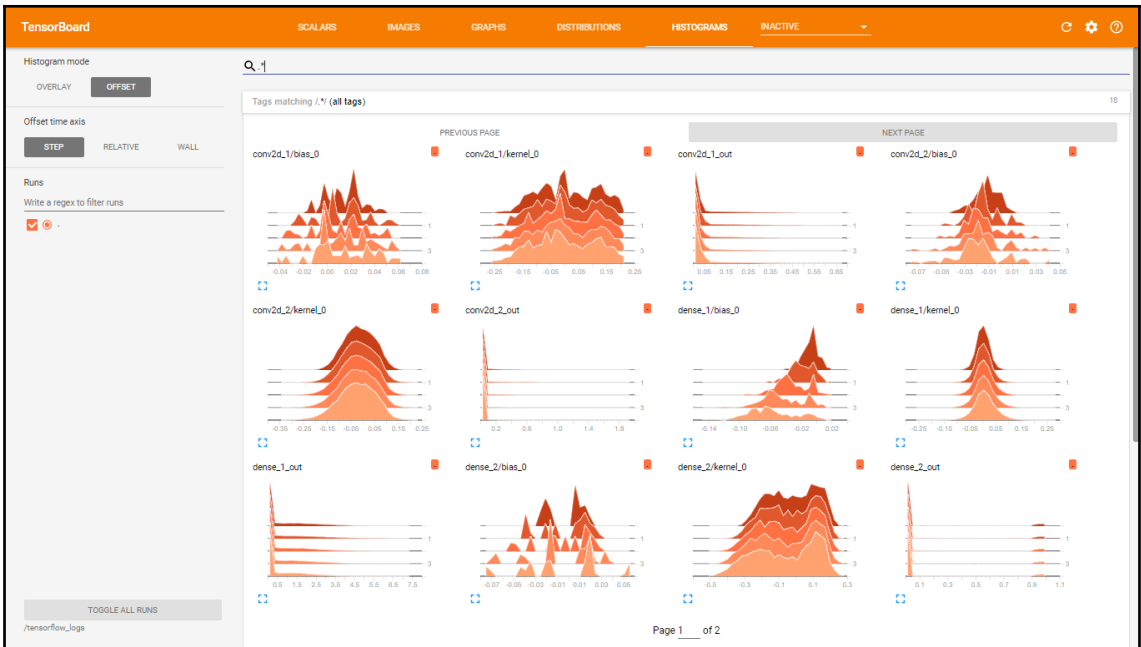
# Chapter 8: Deep Learning models using TensorFlow in R

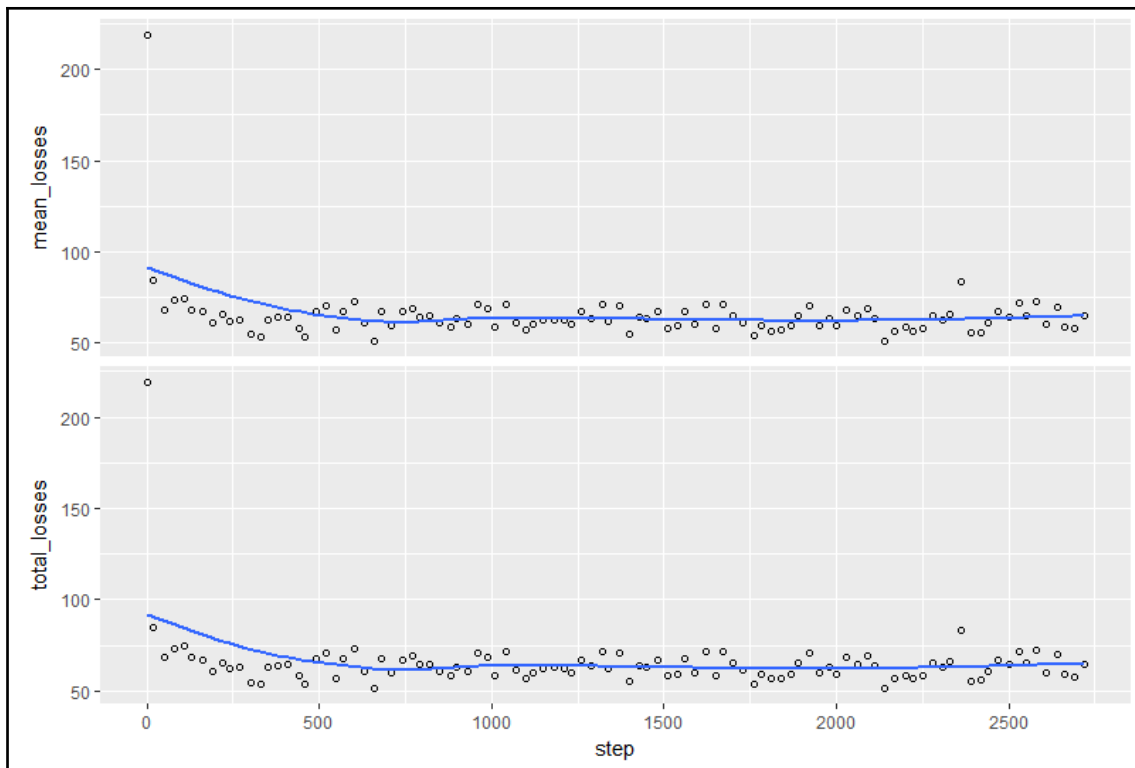




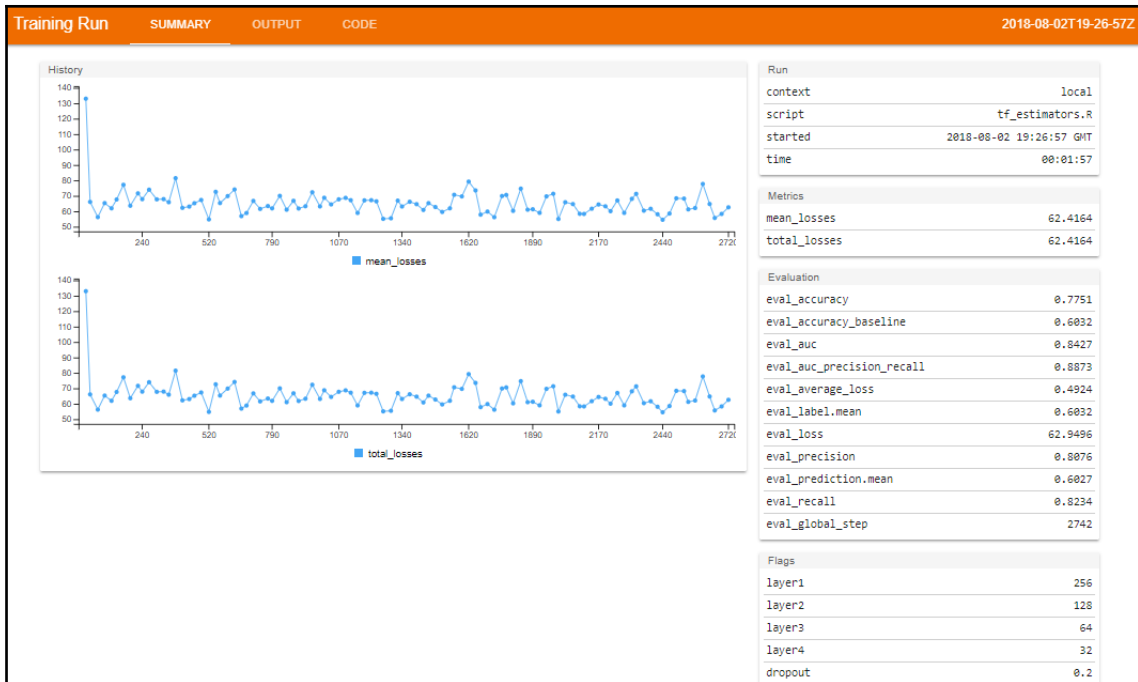






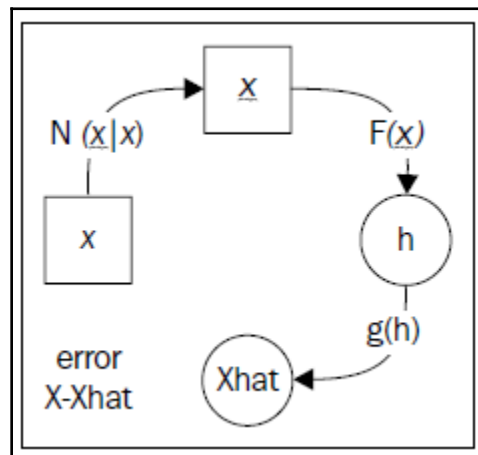
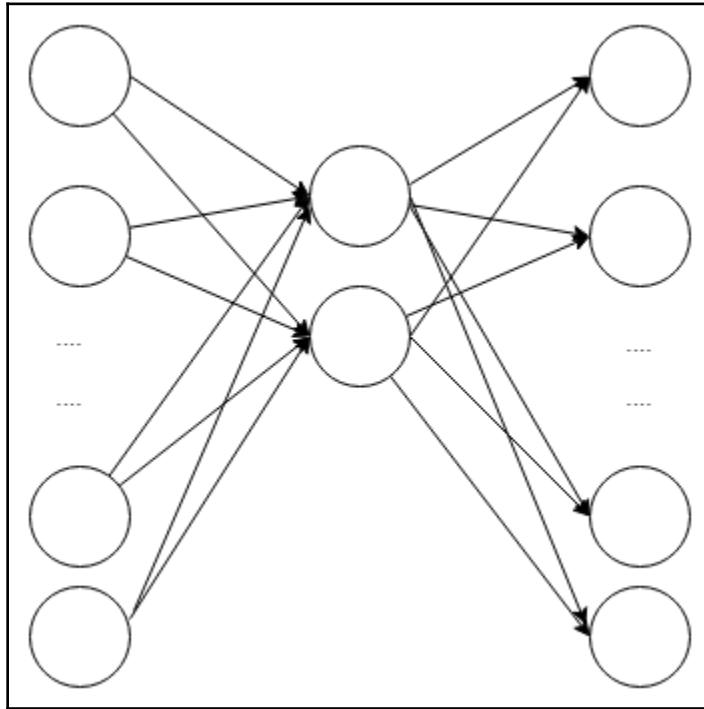






Compare Runs		2018-08-02T19-50-17Z 2018-08-02T19-52-04Z	
Run		Run	
context	local	context	local
script	tf_estimators.R	script	tf_estimators.R
started	2018-08-02 19:50:17 GMT	started	2018-08-02 19:52:04 GMT
time	00:01:45	time	00:01:34
Metrics		Metrics	
mean_losses	58.6853	mean_losses	68.7100
total_losses	58.6853	total_losses	68.7100
Evaluation		Evaluation	
eval_accuracy	0.7746	eval_accuracy	0.7724
eval_accuracy_baseline	0.6032	eval_accuracy_baseline	0.6032
eval_auc	0.8431	eval_auc	0.8425
eval_auc_precision_recall	0.8874	eval_auc_precision_recall	0.8873
eval_average_loss	0.4896	eval_average_loss	0.4844
eval_label.mean	0.6032	eval_label.mean	0.6032
eval_loss	62.5818	eval_loss	61.9193
eval_precision	0.8053	eval_precision	0.822
eval_prediction.mean	0.59	eval_prediction.mean	0.5918
eval_recall	0.8259	eval_recall	0.7948
eval_global_step	2742	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: Identifying Anomalous Data





```

1 library(keras)
2 library(corrplot)
3
4 options(width = 70, digits = 2)
5 options(scipen=999)
6
7
8 dataDirectory <- "../data"
9 if (!file.exists(paste(dataDirectory, "/train.csv", sep="")))
10 {
11   link <- "https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/R/data/mnist/"
12   if (!file.exists(paste(dataDirectory, "/mnist_csv.zip", sep="")))
13     download.file(link, destfile = paste(dataDirectory, "/mnist_csv.zip", sep=""))
14   unzip(paste(dataDirectory, "/mnist_csv.zip", sep=""), exdir = dataDirectory)
15   if (file.exists(paste(dataDirectory, "/test.csv", sep="")))
16     file.remove(paste(dataDirectory, "/test.csv", sep=""))
17 }
18
19 data <- read.csv("../data/train.csv", header=TRUE)
20
21 set.seed(42)
22 sample = sample(nrow(data), 0.5*nrow(data))
23 test <- setdiff(seq_len(nrow(data)), sample)
24 train.x <- data[sample, -1]
25 <

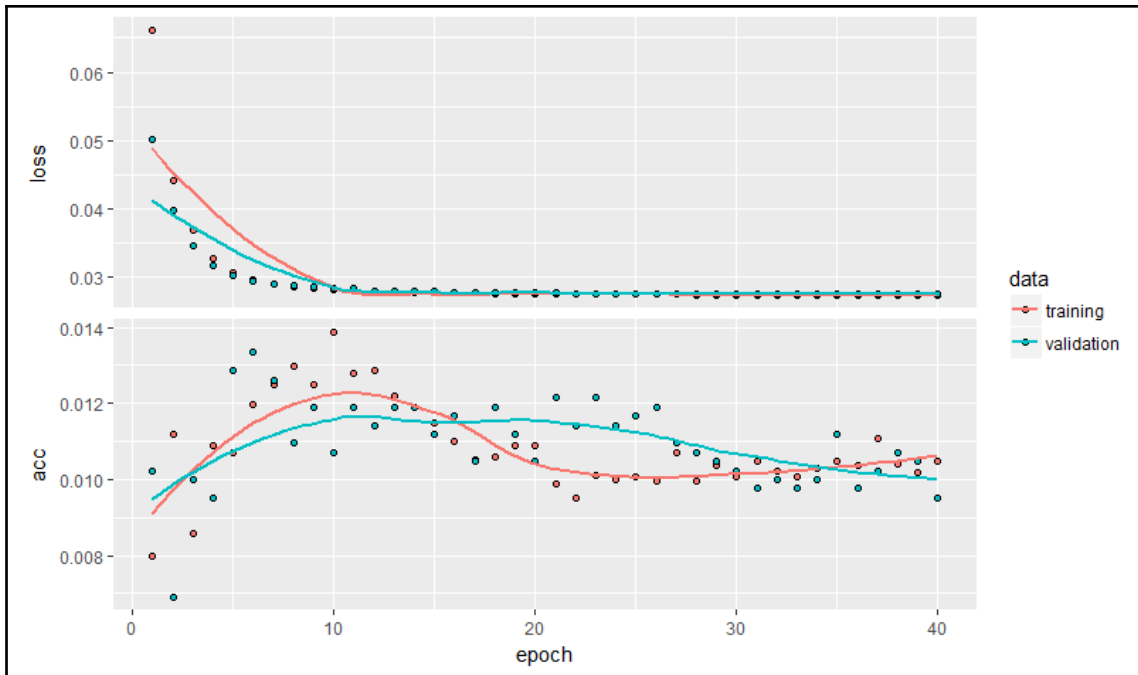
```

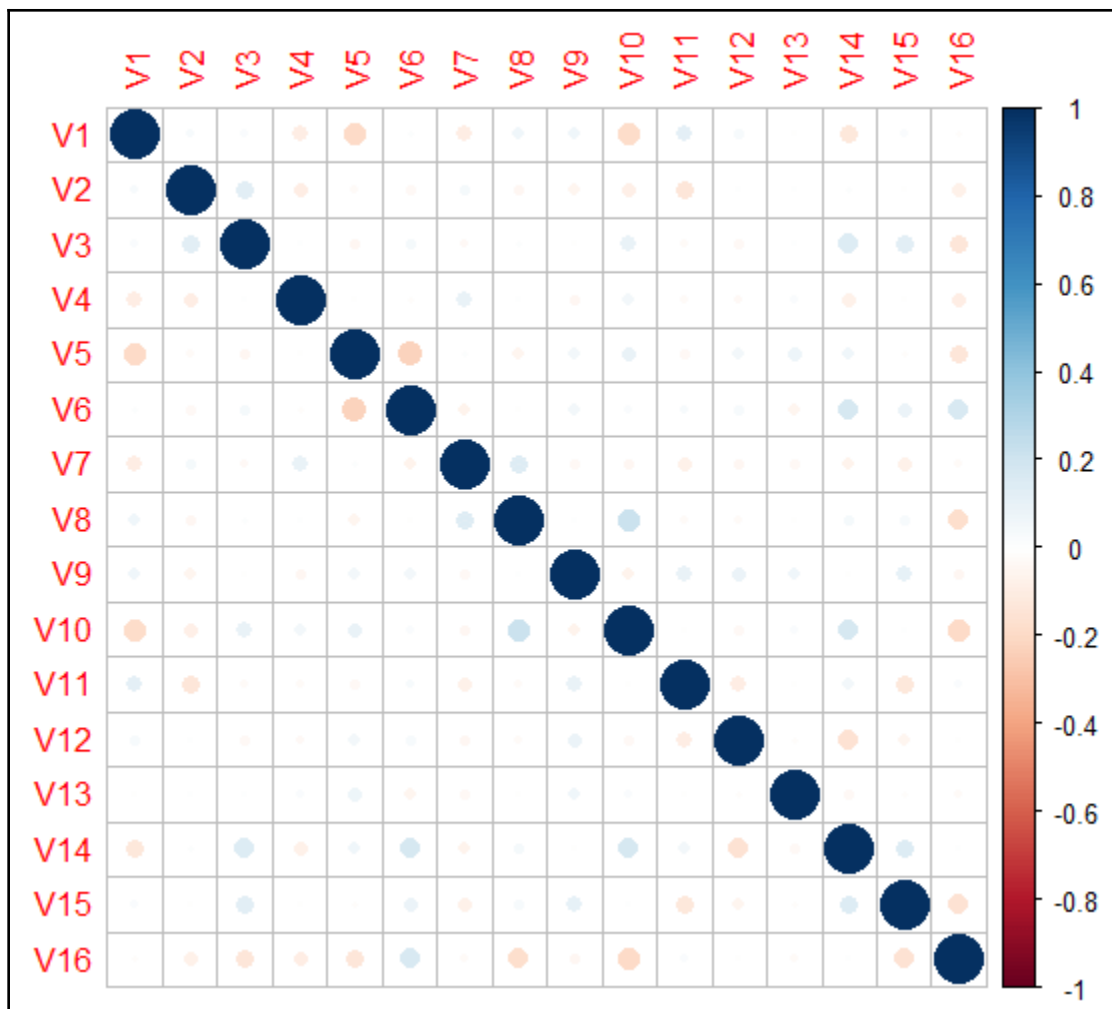
Console Terminal

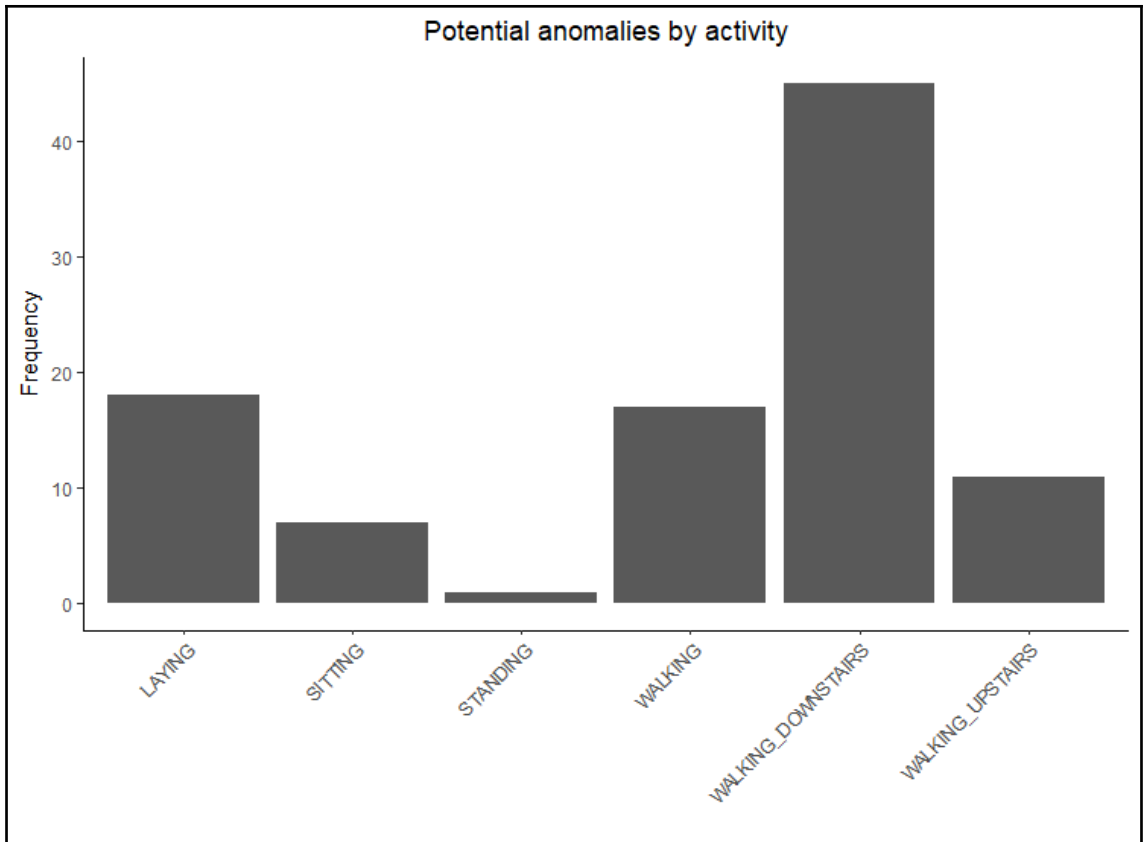
```

D:/r-di-book/code/chapter8/
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

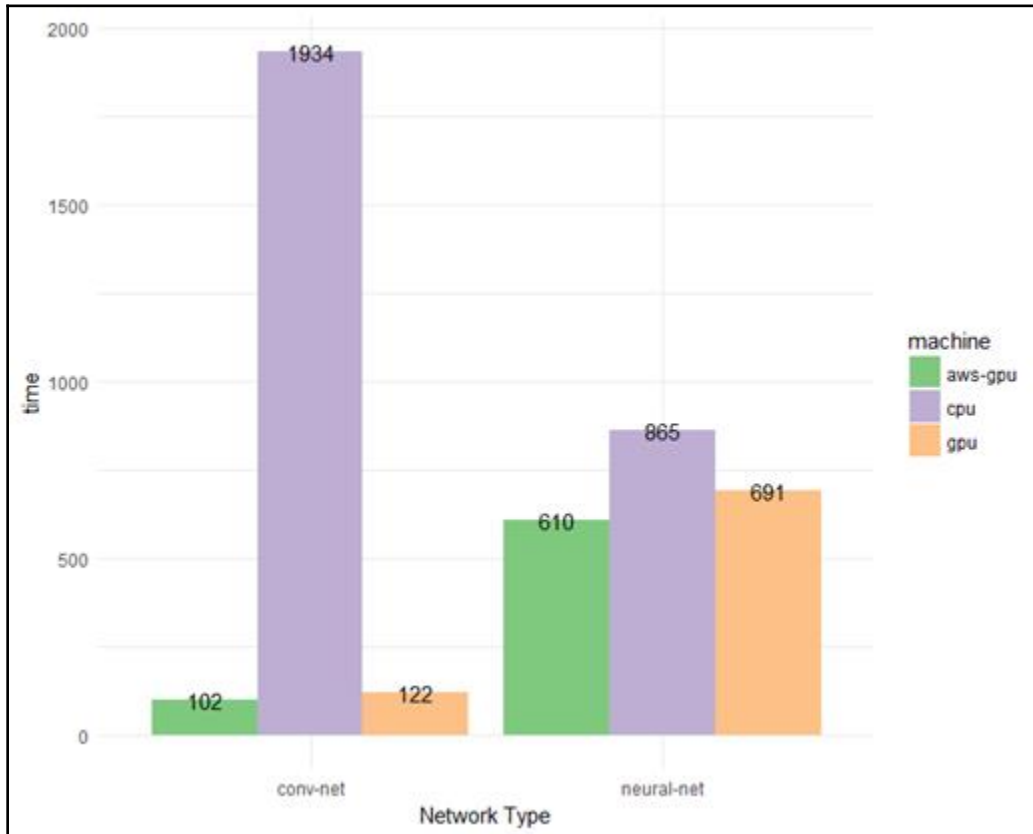
```

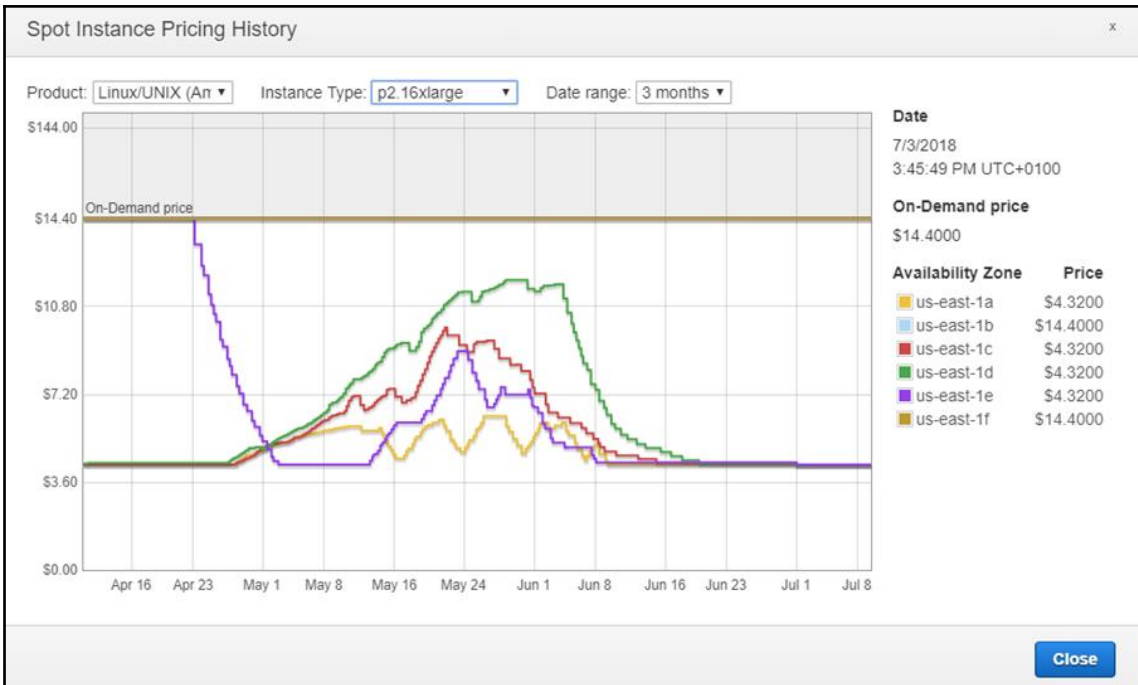
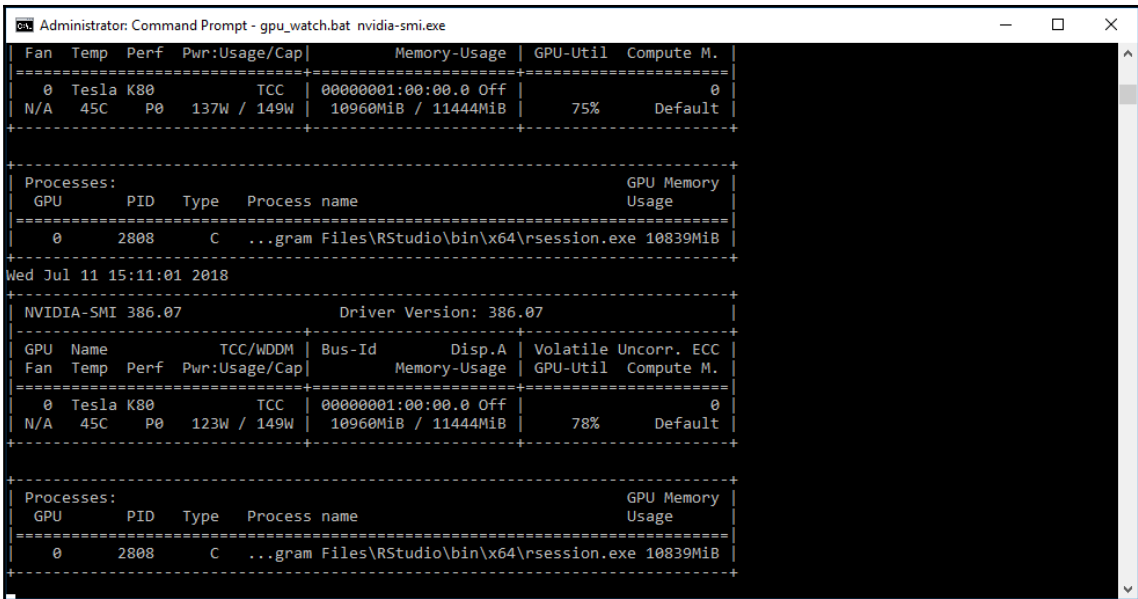


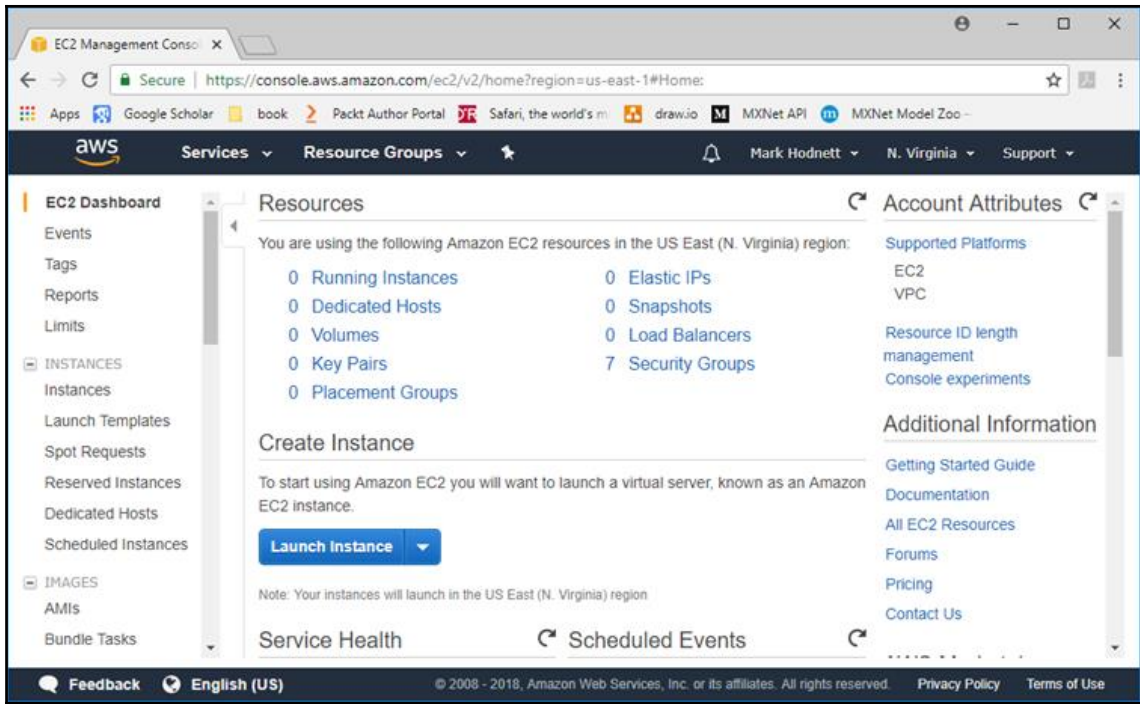




## Chapter 10: Running Deep Learning Models in the Cloud







The screenshot shows the AWS Management Console interface for the 'Launch Instance Wizard'. The browser address bar indicates the URL: `https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:`. The navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information for 'Mark Hodnett' in 'N. Virginia'. The wizard progress bar shows seven steps: 1. Choose AMI (active), 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

### Step 1: Choose an Amazon Machine Image (AMI)

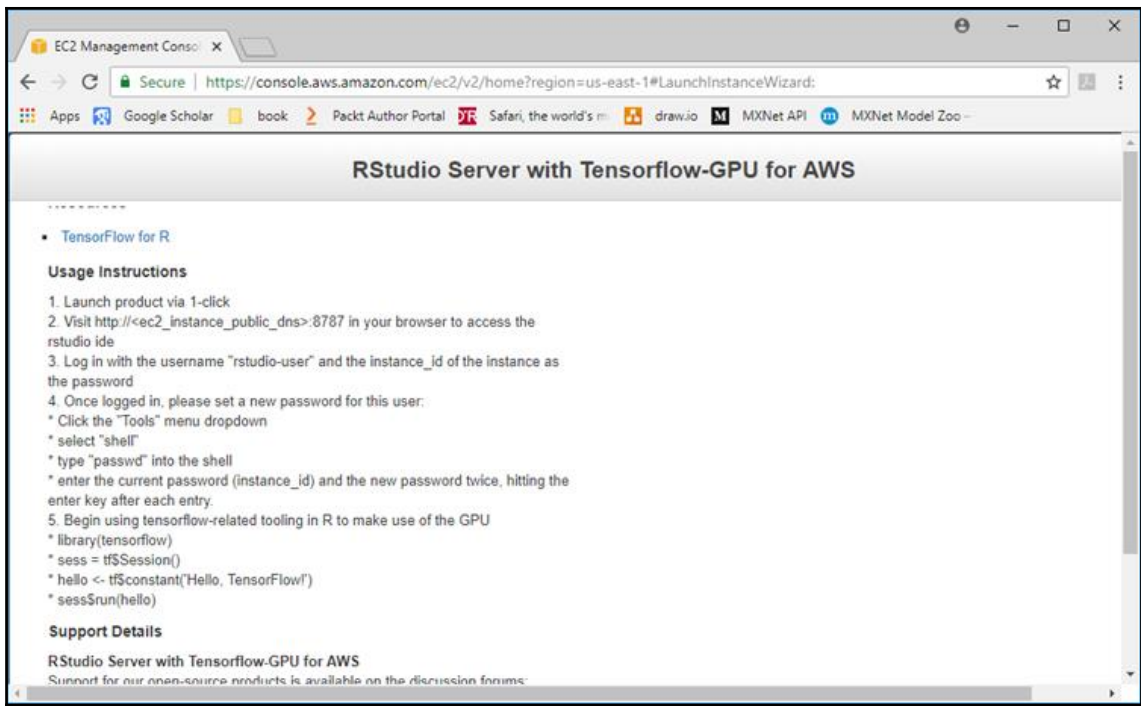
[Cancel and Exit](#)

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.

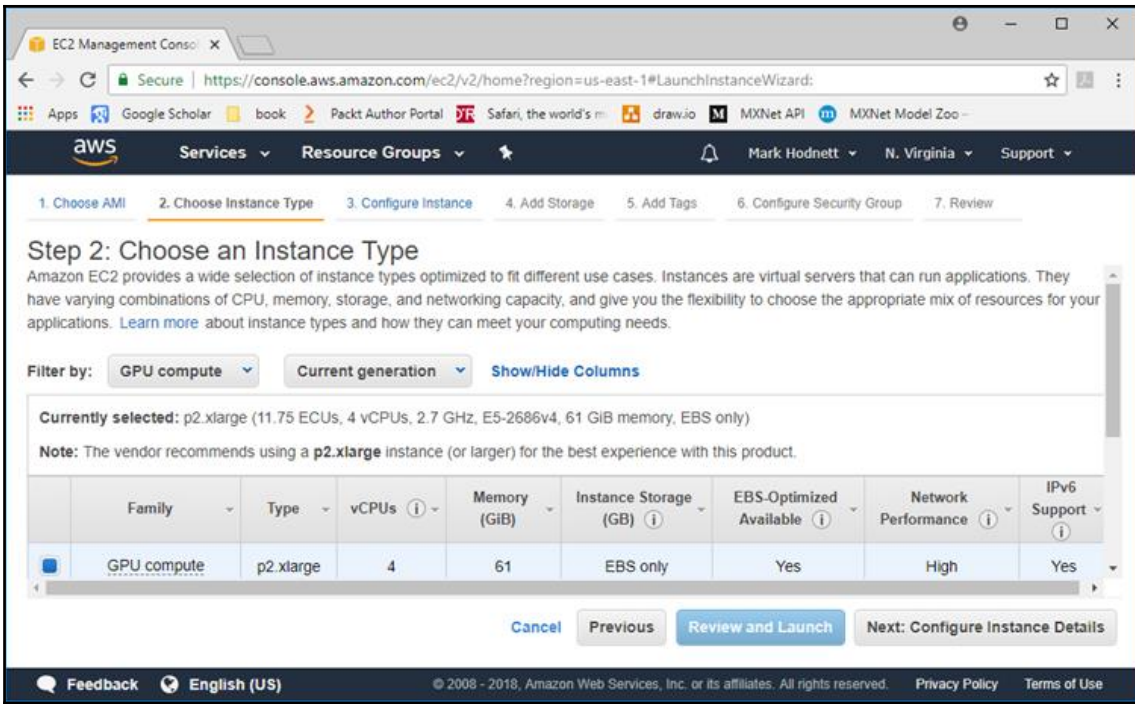
Search results for 'rstudio' are shown, displaying the 'RStudio Server Pro for AWS' AMI. The AMI details include:

- Rating: ★★★★★ (0) | 1.1.423.aws.1 | [Previous versions](#) | Sold by RStudio
- Price: **\$2.65/hr or \$14,000/yr (40% savings) for software + AWS usage fees**
- OS: Linux/Unix, Ubuntu V16.04 | 64-bit Amazon Machine Image (AMI) | Updated: 3/12/18
- Description: 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 ...

The footer of the console includes 'Feedback', 'English (US)', and copyright information: '© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.







EC2 Management Console

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

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

## 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

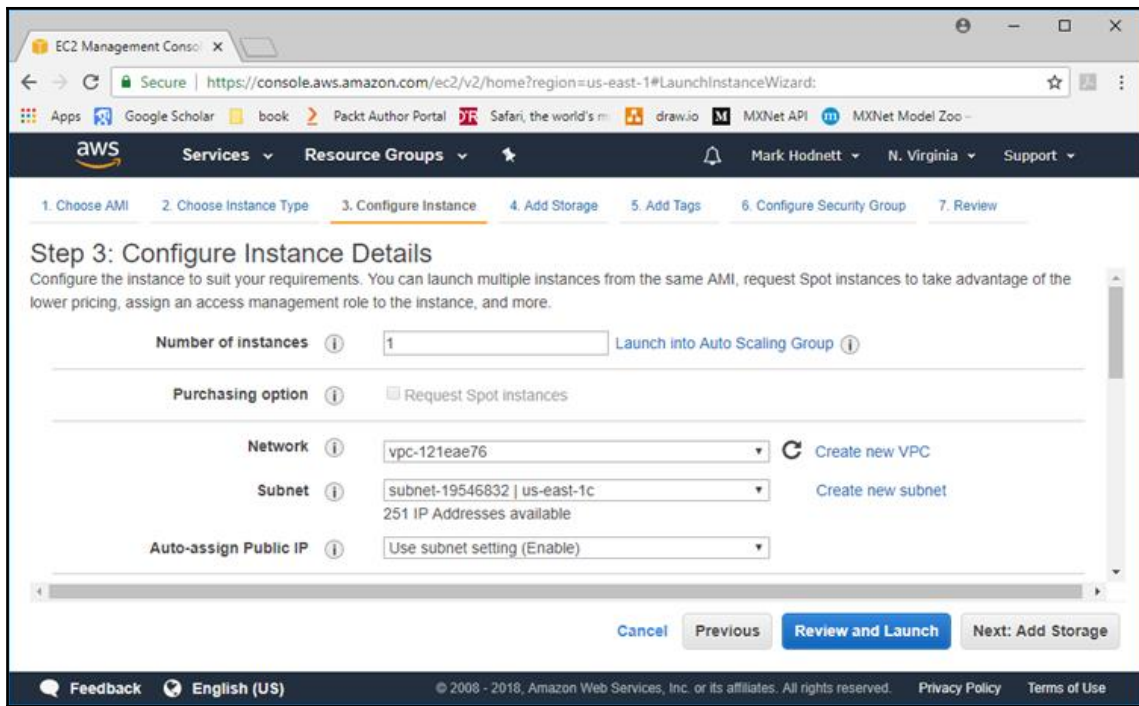
Currently selected: p2.xlarge (11.75 ECUs, 4 vCPUs, 2.7 GHz, E5-2686v4, 61 GiB memory, EBS only)

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

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



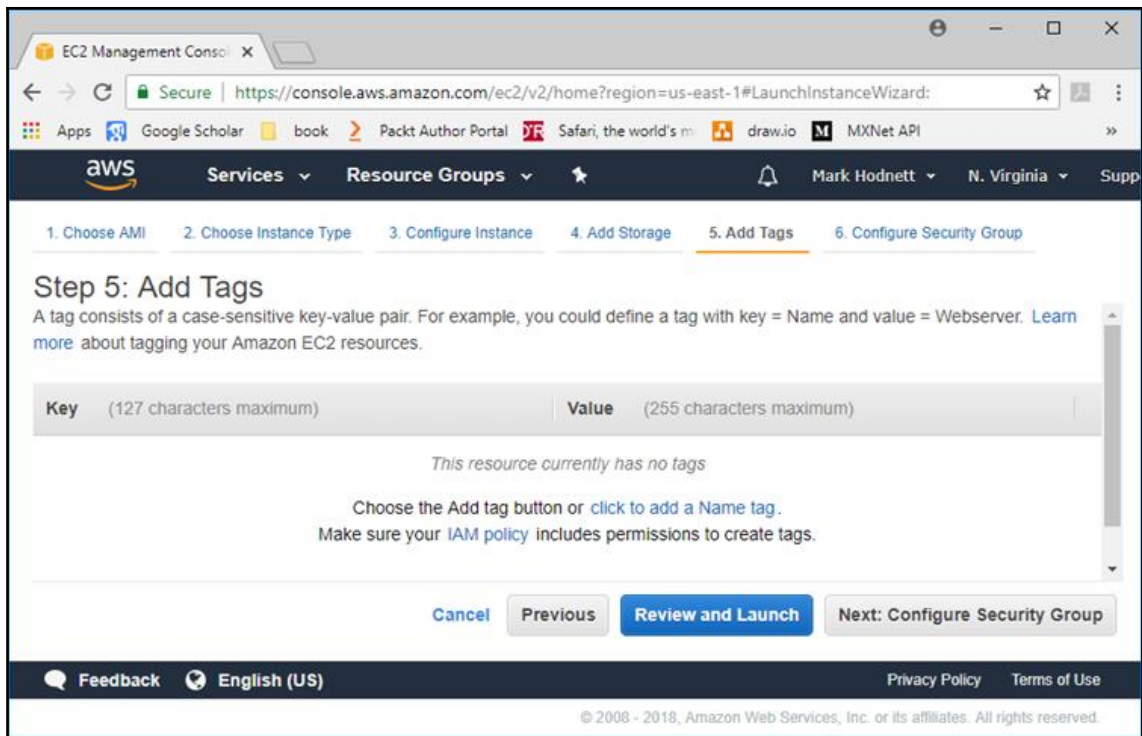
The screenshot shows the AWS Management Console interface for configuring an EC2 instance. The current step is 'Step 4: Add Storage'. The page includes a progress bar at the top with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (highlighted), 5. Add Tags, 6. Configure Security Group, and 7. Review.

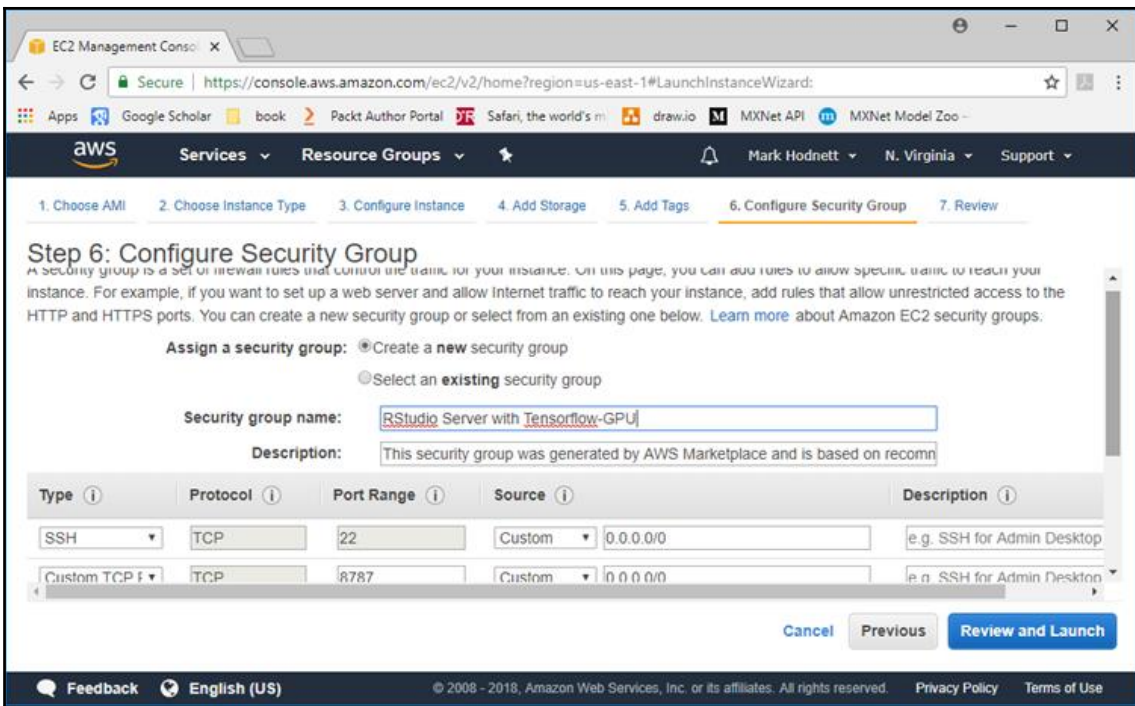
Below the progress bar, the text reads: 'Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.'

The main configuration area contains a table with the following columns: Volume Type, Device, Snapshot, Size (GiB), Volume Type, IOPS, Throughput (MB/s), Delete on Termination, and Encrypted. The table shows one row for the 'Root' volume with the following values: /dev/sda1, snap-0f4a870e33c971469, 60, Magnetic, N/A, N/A, a checked box for 'Delete on Termination', and 'Not Encrypted'.

Below the table is an 'Add New Volume' button. At the bottom of the configuration area, there are four buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Tags'.

The footer of the console includes 'Feedback', 'English (US)', and copyright information: '© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.





EC2 Management Console

Secure | <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard>

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### 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

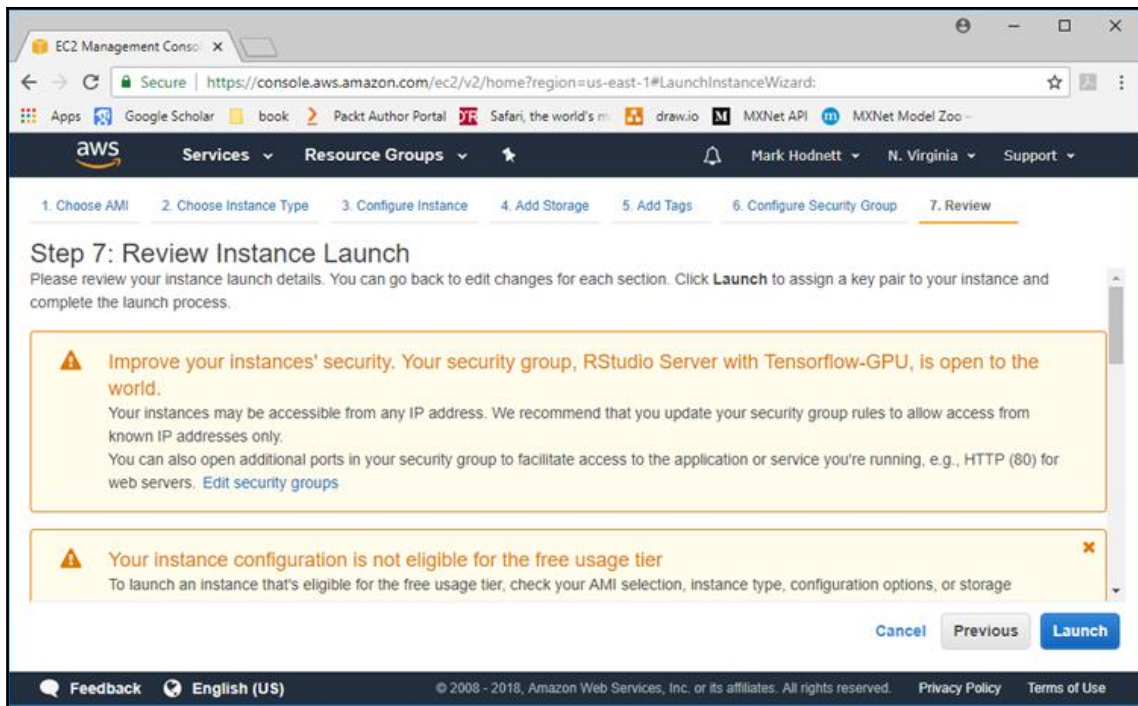
Security group name:

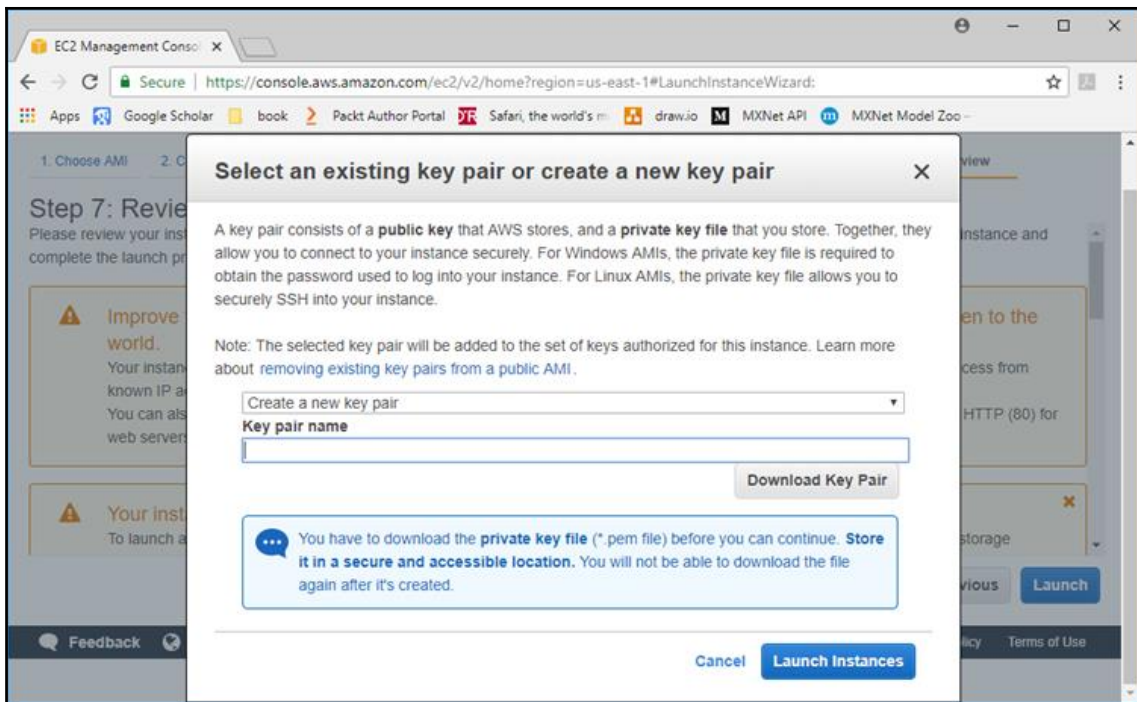
Description:

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

Cancel Previous **Review and Launch**

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use







The screenshot shows the AWS Management Console for the EC2 service. The browser address bar indicates the URL: <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home>. The page is titled "Resources" and shows the following counts for 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	

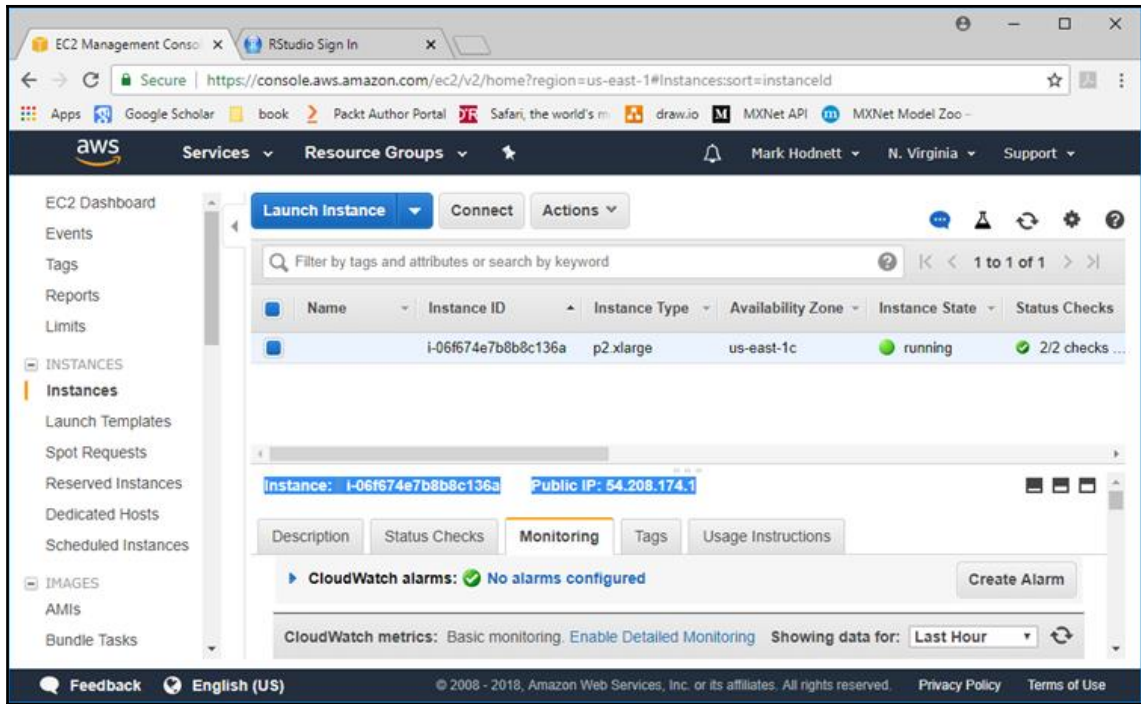
Below the resource counts, there is a "Create Instance" section with the text: "To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance." A blue "Launch Instance" button is visible. A note states: "Note: Your instances will launch in the US East (N. Virginia) region".

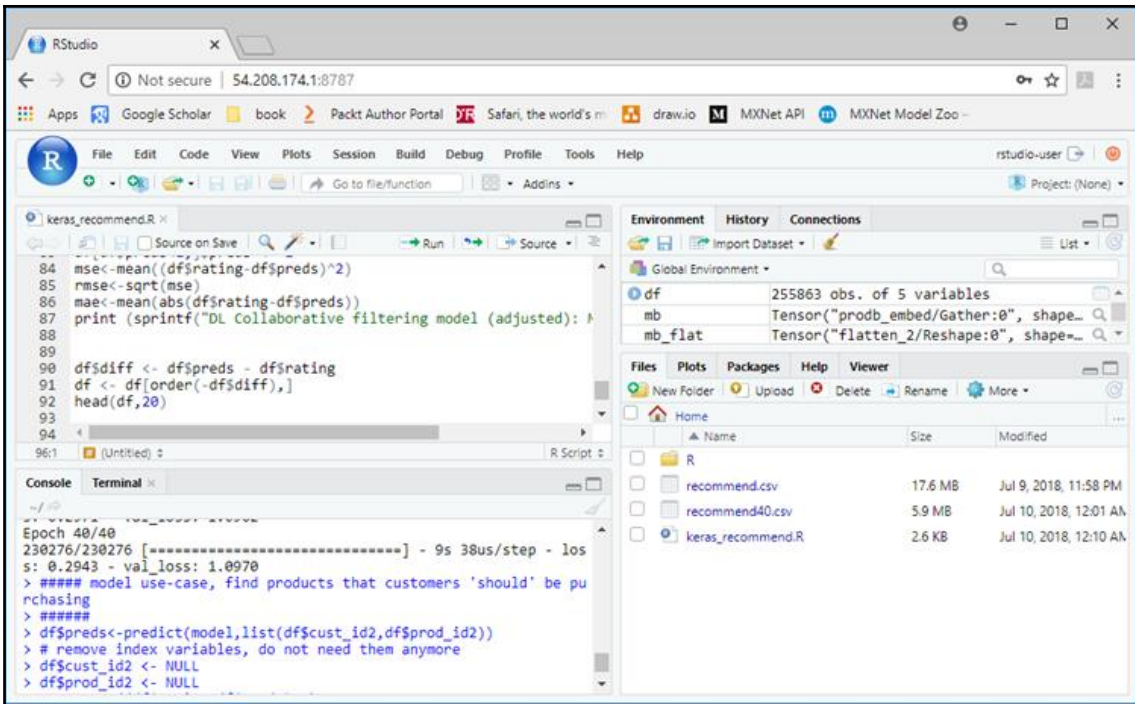
The left sidebar contains navigation links for "EC2 Dashboard", "Events", "Tags", "Reports", "Limits", "INSTANCES", "Instances", "Launch Templates", "Spot Requests", "Reserved Instances", "Dedicated Hosts", "Scheduled Instances", "IMAGES", "AMIs", "Bundle Tasks", and "ELASTIC BLOCK".

The right sidebar contains "Account Attributes" (Supported Platforms: EC2, VPC; Resource ID length management; Console experiments) and "Additional Information" (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us).

At the bottom of the page, there is a footer with "Feedback", "English (US)", "© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.", "Privacy Policy", and "Terms of Use".







The screenshot shows the RStudio interface with a script editor and a console. The script editor contains the following R code:

```
84 mse<-mean((df$rating-df$preds)^2)
85 rmse<-sqrt(mse)
86 mae<-mean(abs(df$rating-df$preds))
87 print(sprintf("DL Collaborative filtering model (adjusted): %f", mae))
88
89
90 df$difff <- df$preds - df$rating
91 df <- df[order(-df$difff),]
92 head(df,20)
93
94
```

The console output shows the results of the script execution:

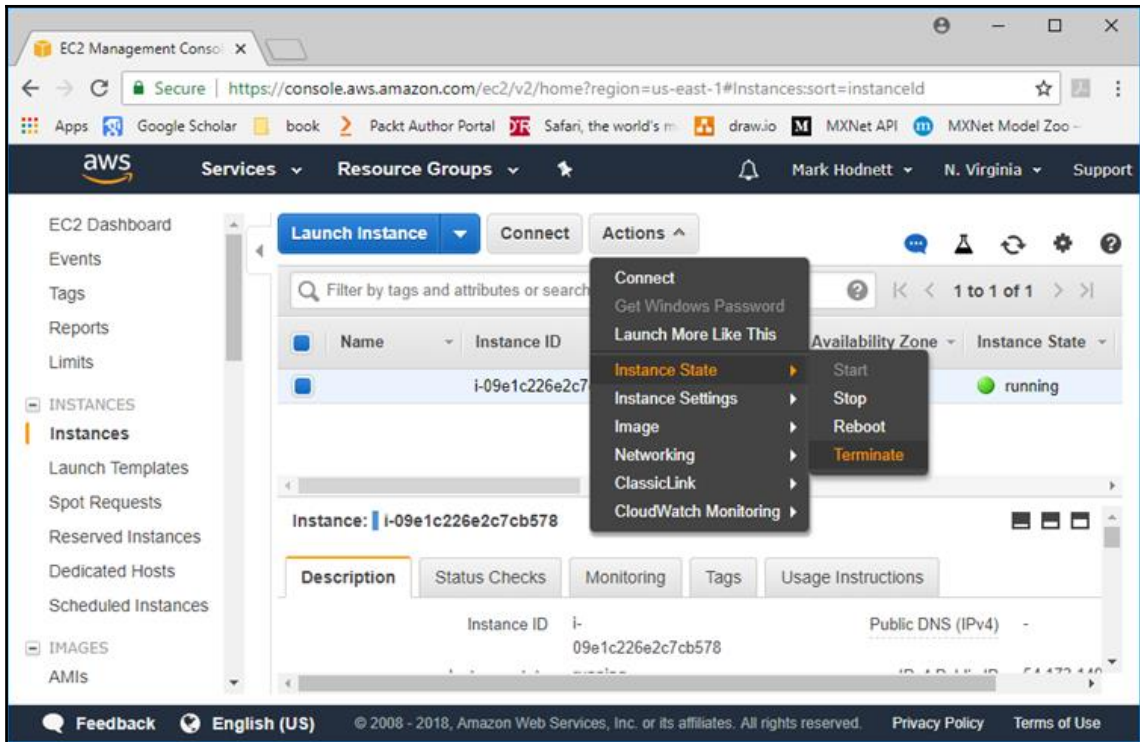
```
Epoch 40/40
230276/230276 [=====] - 9s 38us/step - loss: 0.2943 - val_loss: 1.0970
> ##### model use-case, find products that customers 'should' be purchasing
> #####
> df$preds<-predict(model,list(df$cust_id2,df$prod_id2))
> # remove index variables, do not need them anymore
> df$cust_id2 <- NULL
> df$prod_id2 <- NULL
```

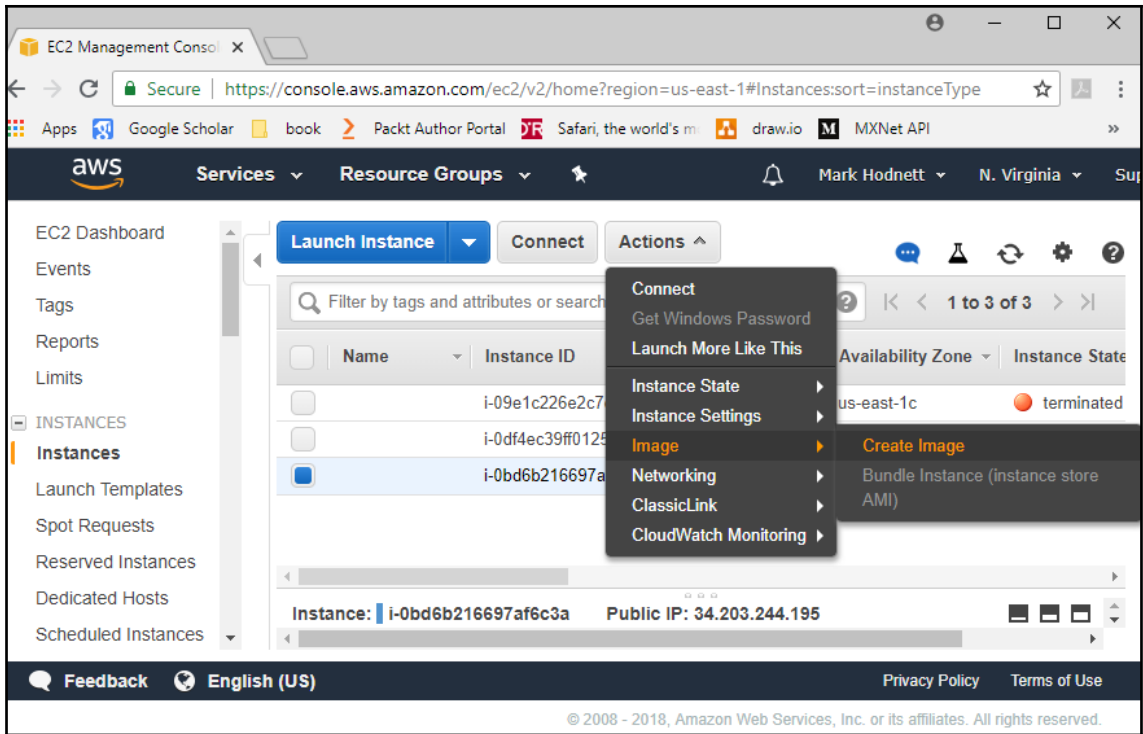
The Environment pane shows the following objects:

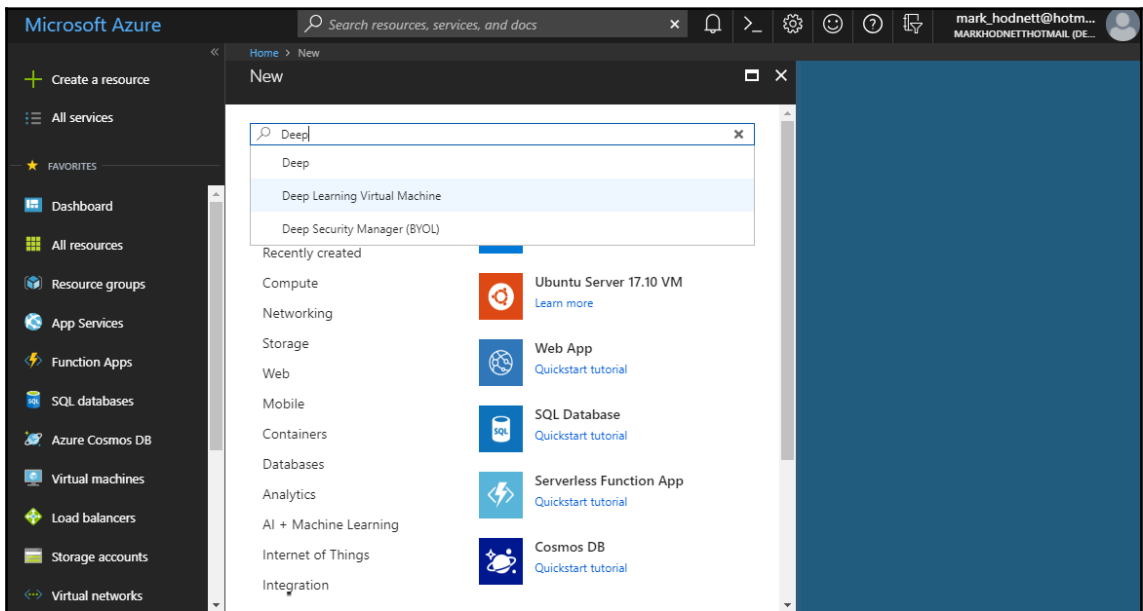
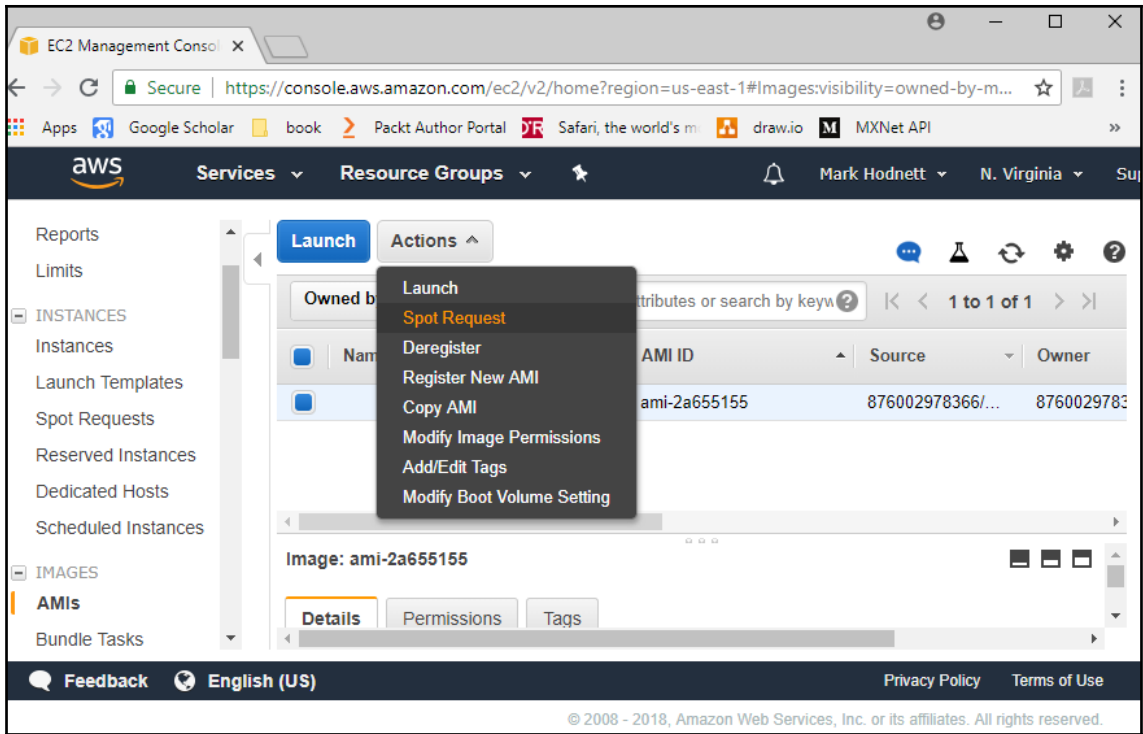
Object	Value
df	255863 obs. of 5 variables
mb	Tensor("prod_embed/Gather:0", shape=...)
mb_flat	Tensor("flatten_2/Reshape:0", shape=...)

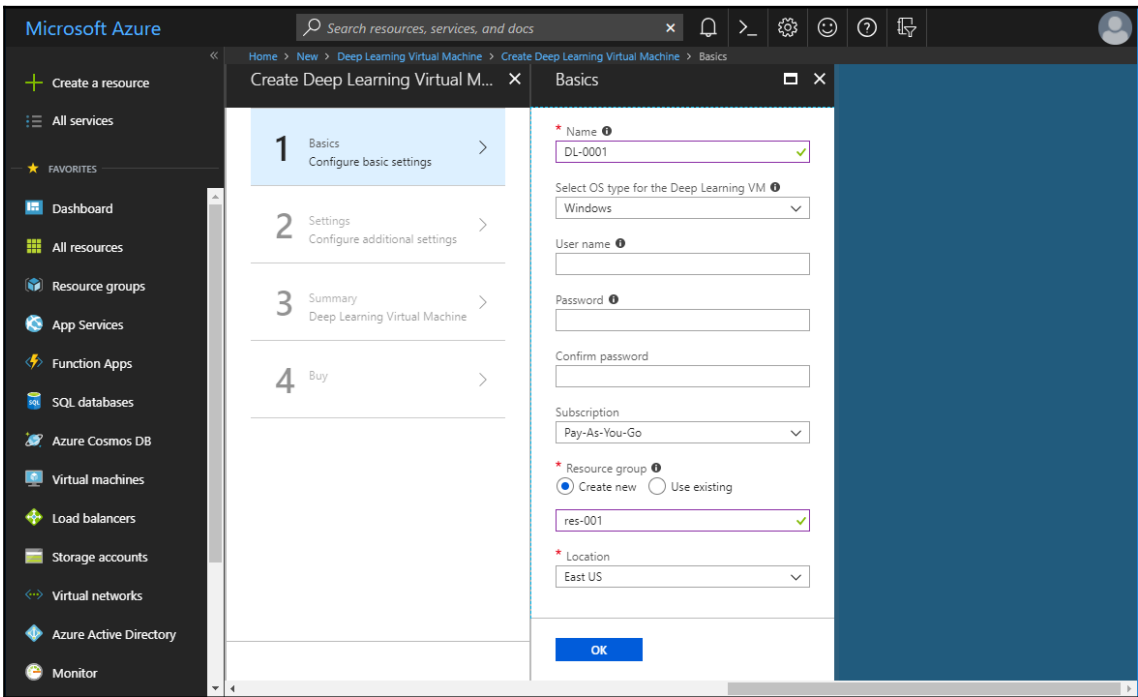
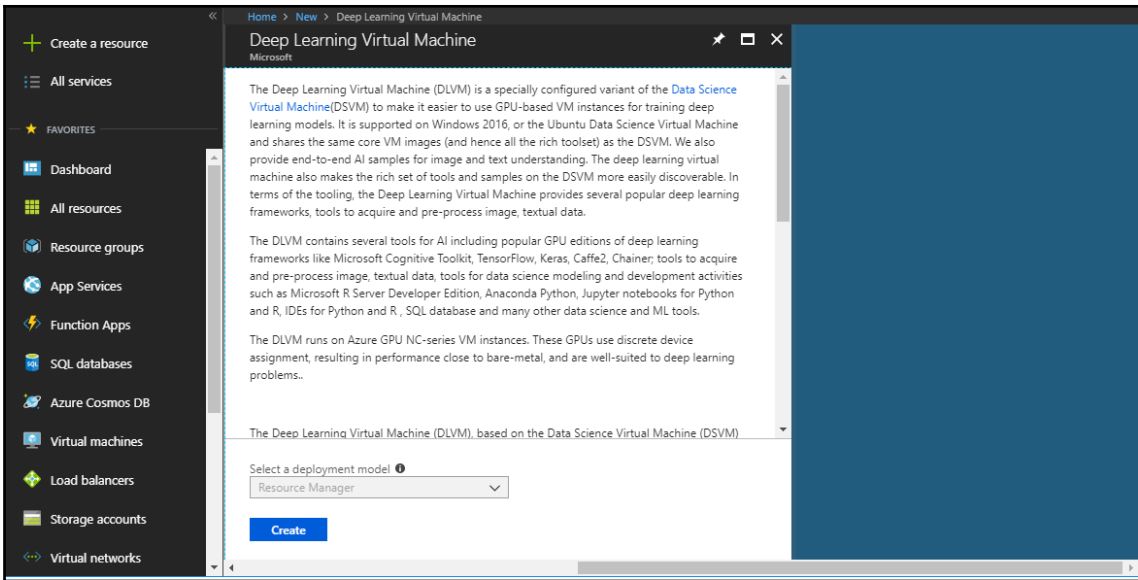
The Files pane shows the following files:

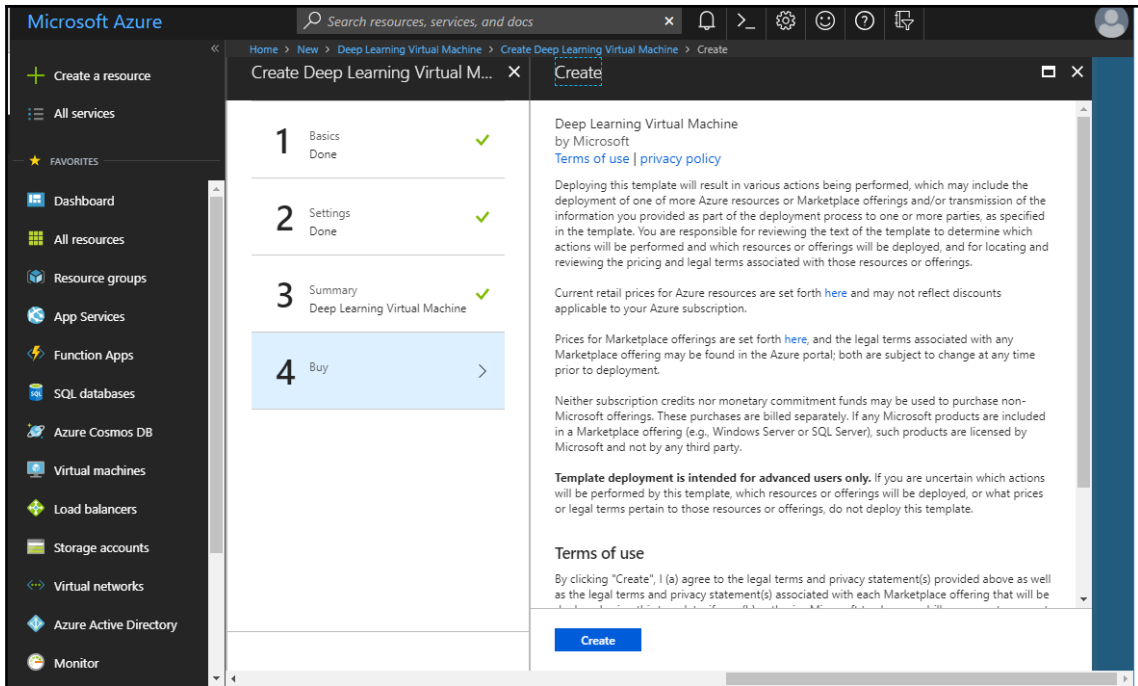
Name	Size	Modified
R		
recommend.csv	17.6 MB	Jul 9, 2018, 11:58 PM
recommend40.csv	5.9 MB	Jul 10, 2018, 12:01 AM
keras_recommend.R	2.6 KB	Jul 10, 2018, 12:10 AM

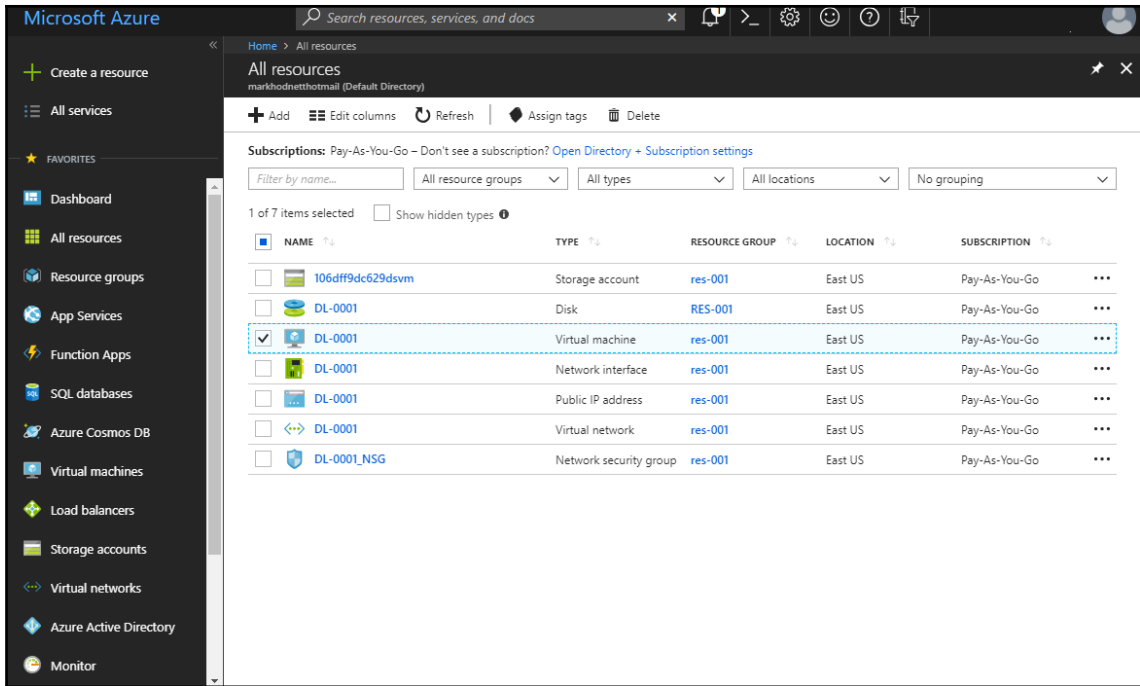




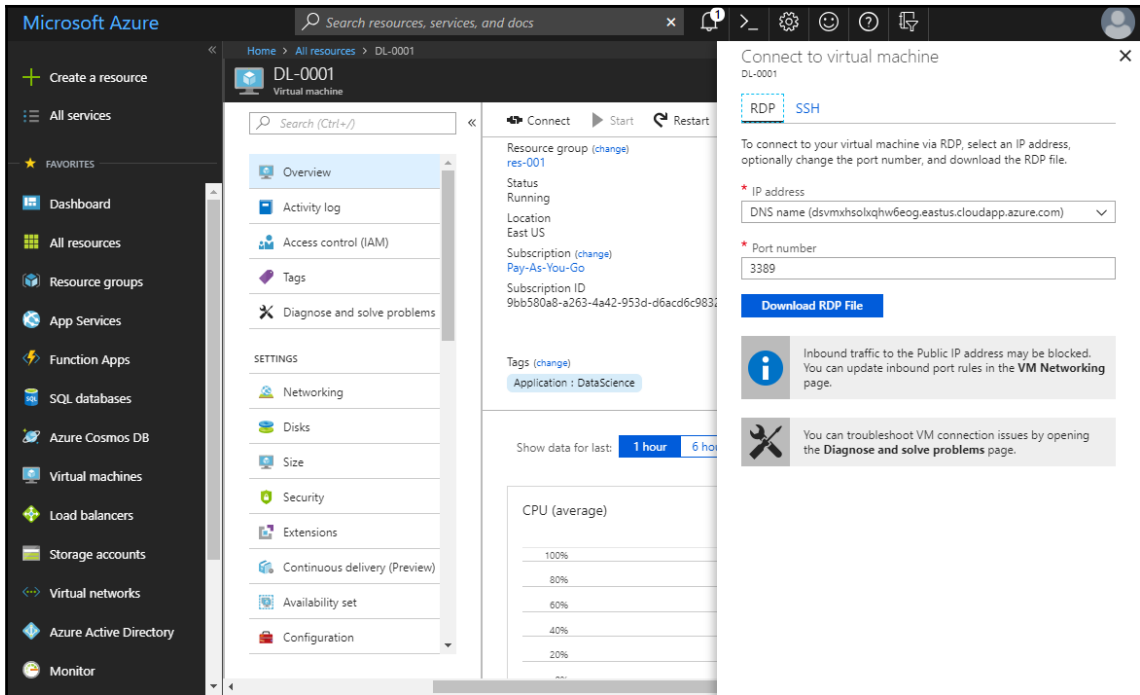


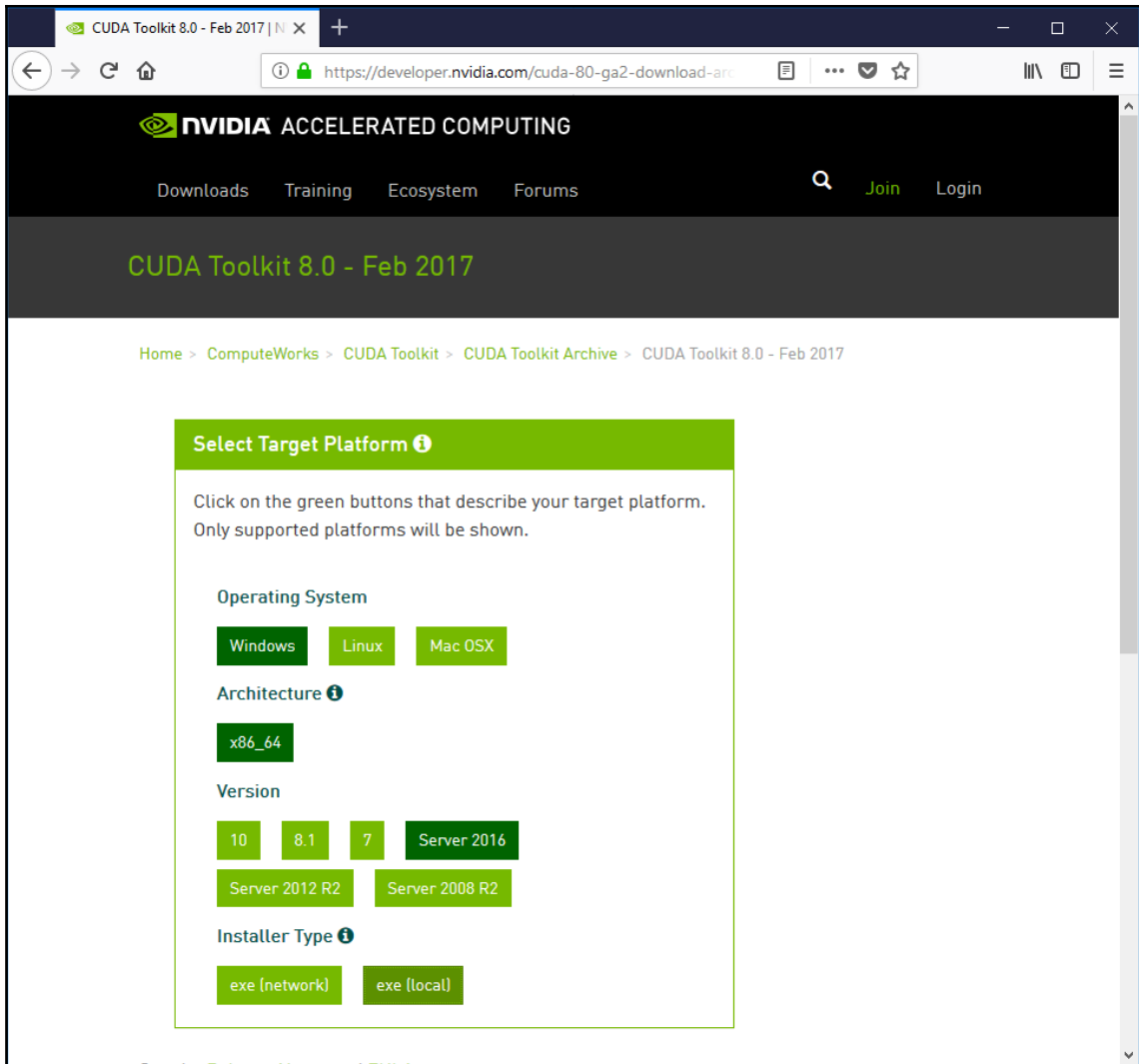












The screenshot shows the RStudio interface. The script editor contains the following R code:

```

1 devtools::install_github("rstudio/cloudml")
2 library(cloudml)
3 gcloud_init()
4
5
6 # cloudml::train("keras_recommend.R")

```

The console window shows the output of the `gcloud` command:

```

$ "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 gcloud.

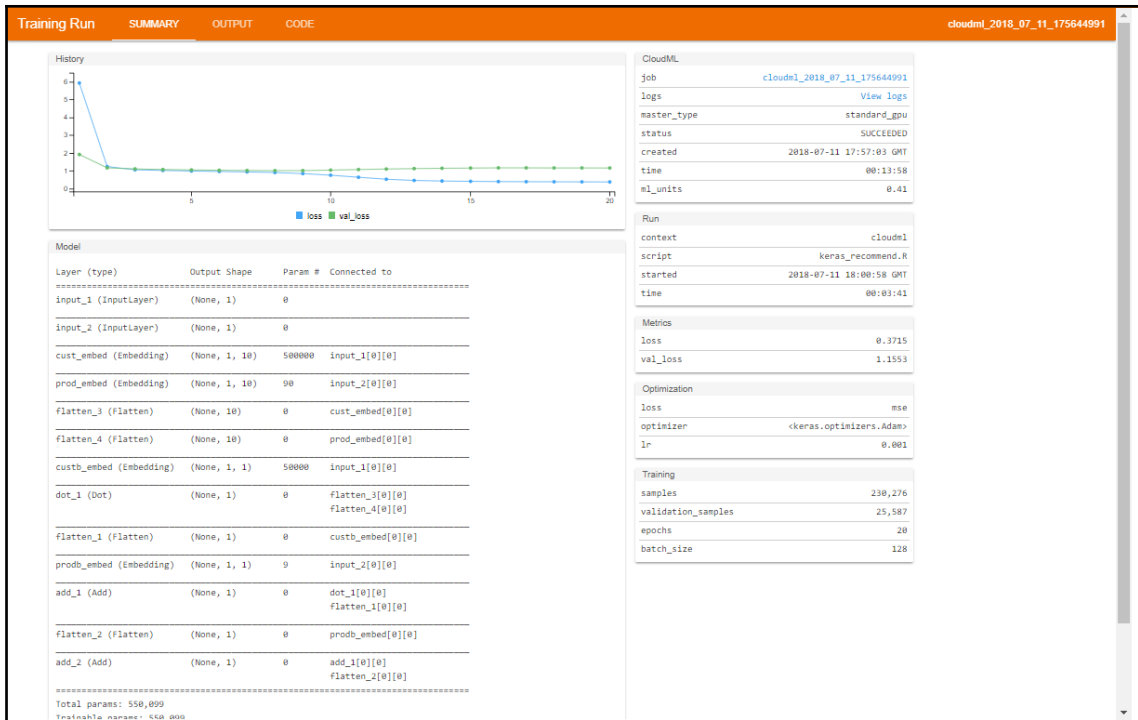
settings from your current configuration [default] are:
core:
  account:
  disable_usage_reporting: 'False'
  project: iconic-range-209

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

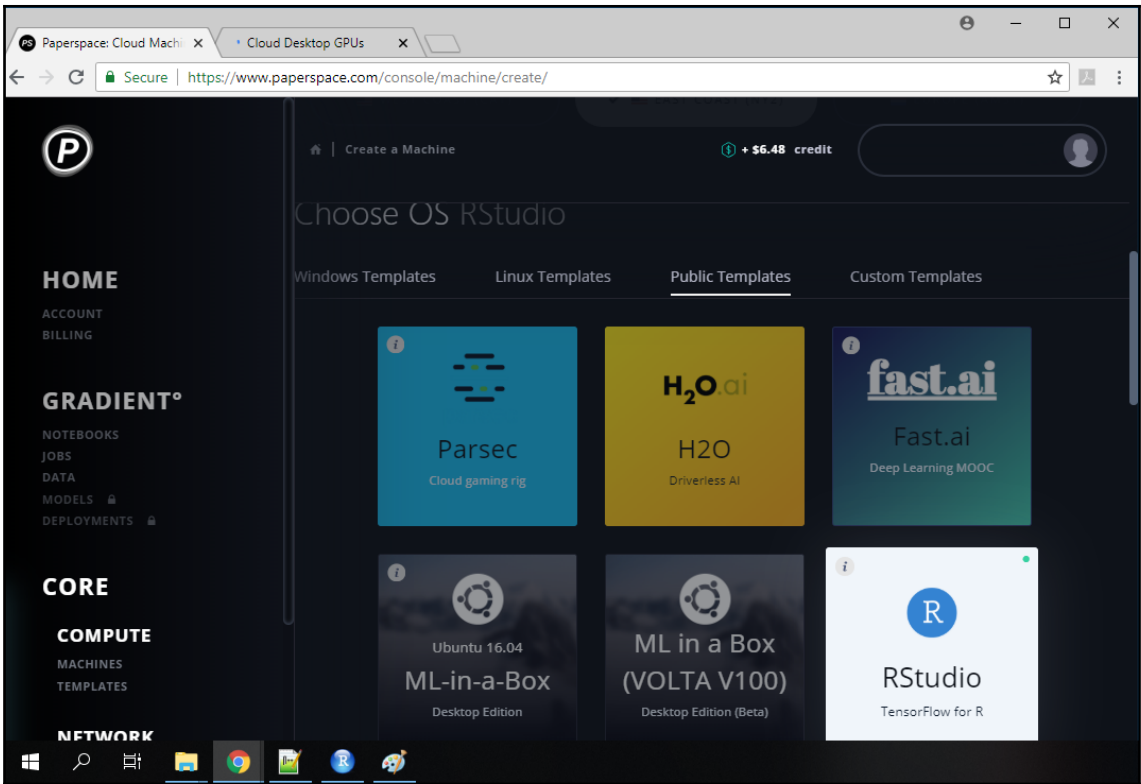
```

The screenshot shows the Google Cloud Platform ML Engine Jobs page. The page displays a list of training jobs with the following columns: Job ID, Type, Creation time, Elapsed time, and Logs.

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



```
Training Run SUMMARY OUTPUT CODE cloudml_2018_07_11_175644991
87 > mae <- mean(abs(df$rating - df$preds))
88
89 > print(sprintf("DL Collaborative filtering model: MSE=%1.3f, RMSE=%1.3f, MAE=%1.3f",
90 +   mse, rmse, mae))
91 [1] "DL Collaborative filtering model: MSE=0.163, RMSE=0.403, MAE=0.281"
92
93 > df <- df[order(-df$preds), ]
94
95 > head(df)
96   prod_id  cust_id rating  preds
97 193512 D00005 CUST0000991836  5 5.842071
98  54820 D00002 CUST0000485110  5 5.836084
99 17735  D00001 CUST0000299527  5 5.806091
100 37823 D00001 CUST0000448940  5 5.796087
101 97862 D00003 CUST0000264653  5 5.785829
102 61905 D00002 CUST0000124725  5 5.783827
103
104 > df[df$preds > 5, ]$preds <- 5
105
106 > df[df$preds < 1, ]$preds <- 1
107
108 > mse <- mean((df$rating - df$preds)^2)
109
110 > rmse <- sqrt(mse)
111
112 > mae <- mean(abs(df$rating - df$preds))
113
114 > print(sprintf("DL Collaborative filtering model (adjusted): MSE=%1.3f, RMSE=%1.3f, MAE=%1.3f",
115 +   mse, rmse, mae))
116 [1] "DL Collaborative filtering model (adjusted): MSE=0.150, RMSE=0.387, MAE=0.242"
117
118 > df$diff <- df$preds - df$rating
119
120 > df <- df[order(-df$diff), ]
121
122 > head(df, 20)
123   prod_id  cust_id rating  preds  diff
```





# Chapter 11: The Next Level in Deep Learning

