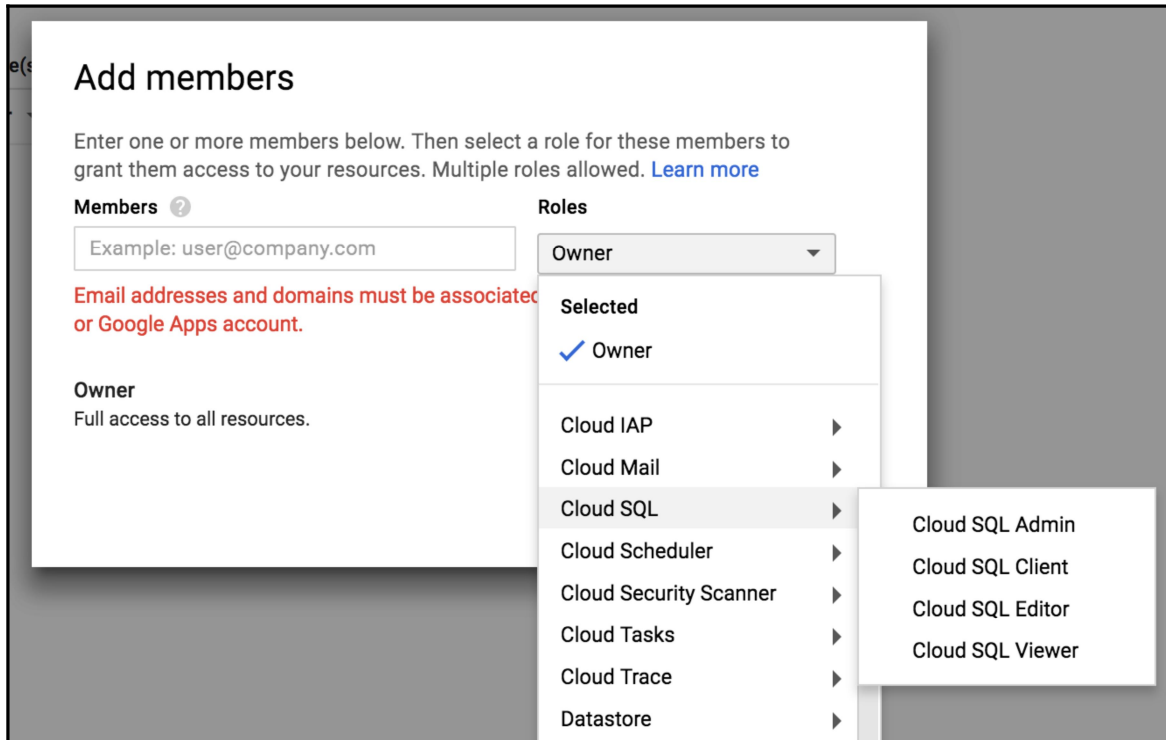


Chapter 1: Introducing the Google Cloud Platform



Chapter 2: Google Compute Engine

May 15th, 2013

- Google Compute Engine is available for open signups! We're excited to announce that Google Compute Engine is now available for open signups and anyone can sign up for the service. For signup instructions, see the [signup page](#).

Compute Engine VM instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. You can choose from micro-VMs to large instances running Debian, Windows, or other standard images. Create your first VM instance, import it by CloudEndure migration service or try the quickstart to build a sample app.

Create

or

Import

or

Take the quickstart

Machine type [Basic view](#)

Cores

1 vCPU 1 - 64

Memory

3.75 GB 1 - 6.5

Extend memory [?](#)

CPU platform [?](#)

Automatic

GPUs

The number of GPU dies is linked to the number of CPU cores and memory selected for this instance. For this machine type, you can select no fewer than 1 GPU die.

[Learn more](#)

Number of GPUs None

GPU type NVIDIA Tesla K80

i Machines with GPUs can't be preemptible nor migrate on host maintenance

Equivalent [REST](#) or [command line](#)

<input type="checkbox"/> Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input checked="" type="checkbox"/> sparrow	us-east1-d		10.142.0.2	35.185.34.184	SSH <input type="button" value="v"/> <input type="button" value="⋮"/>

```

packt-gcp x +
Welcome to Cloud Shell! Type "help" to get started.
alexis_perrier@packt-gcp:~$

```

```

alexis_perrier@packt-gcp:~$ gcloud compute instances list
NAME          ZONE          MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP  STATUS
sparrow       us-east1-d    fl-micro      false         10.142.0.2   35.196.158.218  RUNNING

```

Components			
Status	Name	ID	Size
Update Available	BigQuery Command Line Tool	bq	< 1 MiB
Update Available	Cloud SDK Core Libraries	core	7.2 MiB
Update Available	Cloud Storage Command Line Tool	gsutil	3.0 MiB
Update Available	gcloud app Python Extensions	app-engine-python	6.2 MiB
Not Installed	App Engine Go Extensions	app-engine-go	97.7 MiB
Not Installed	Cloud Bigtable Command Line Tool	cbt	4.0 MiB
Not Installed	Cloud Bigtable Emulator	bigtable	3.5 MiB
Not Installed	Cloud Datalab Command Line Tool	datalab	< 1 MiB
Not Installed	Cloud Datastore Emulator	cloud-datastore-emulator	15.4 MiB
Not Installed	Cloud Datastore Emulator (Legacy)	gcd-emulator	38.1 MiB
Not Installed	Cloud Pub/Sub Emulator	pubsub-emulator	33.2 MiB
Not Installed	Emulator Reverse Proxy	emulator-reverse-proxy	14.5 MiB
Not Installed	Google Container Local Builder	container-builder-local	3.7 MiB
Not Installed	Google Container Registry's Docker credential helper	docker-credential-gcr	2.2 MiB
Not Installed	gcloud Alpha Commands	alpha	< 1 MiB
Not Installed	gcloud Beta Commands	beta	< 1 MiB
Not Installed	gcloud app Java Extensions	app-engine-java	116.9 MiB
Not Installed	gcloud app PHP Extensions	app-engine-php	21.9 MiB
Not Installed	kubectl	kubectl	15.9 MiB

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

Allow HTTP traffic

Allow HTTPS traffic

Management **Disks** Networking SSH Keys

Deletion rule

Delete boot disk when instance is deleted

Network interfaces ?

Network interface X

Network
default

Subnetwork
default

Internal IP
10.132.0.2

Internal IP type
Ephemeral

External IP


- None
- Ephemeral
- Create IP address


Done Cancel


NETWORKING


- VPC network > VPC networks
- Network services > External IP addresses
- Interconnect > Firewall rules
- > Routes
- > VPC network peering
- > Shared VPC


STACKDRIVER


 VPC network


 VPC networks

 External IP addresses

 Firewall rules

 Routes

 VPC network peering

 Shared VPC

← Reserve a static address

Name [?]

Description (Optional)

IP version
 IPv4
 IPv6

Type
 Regional
 Global (to be used with Global forwarding rules [Learn more](#))

Region [?]

Attached to [?]

Firewalls


Allow HTTP traffic

Allow HTTPS traffic


Machine type

[Basic view](#)

Cores

 1 vCPU 1 - 64

Memory

 3.75 GB 1 - 6.5

Extend memory [?](#)

CPU platform [?](#)

Automatic [▼](#)

GPUs

The number of GPU dies is linked to the number of CPU cores and memory selected for this instance. For this machine type, you can select no fewer than 1 GPU die.
[Learn more](#)

Number of GPUs [▼](#) 1 **GPU type** [▼](#) NVIDIA Tesla K80

i Machines with GPUs can't be preemptible nor migrate on host maintenance

[^](#) [Less](#)

[Choosing a machine type](#) [↗](#)

Chapter 3: Google Cloud Storage

The screenshot shows the Google Cloud Platform Storage browser interface. At the top, there's a navigation bar with the Google Cloud Platform logo, the project name 'MainProject', and various utility icons. Below the navigation bar, the page title is 'Browser'. There are buttons for 'CREATE BUCKET', 'REFRESH', 'DELETE', and 'SHOW INFO PANEL'. A search bar with the placeholder 'Filter by prefix...' and a 'Columns' dropdown menu are also present. The main content area is titled 'Buckets' and displays a table of storage buckets.

<input type="checkbox"/>	Name	Default storage class [?]	Location	Lifecycle [?]	Labels [?]	Requester pays [?]	
<input type="checkbox"/>	cnn-hwr	Multi-Regional	US	None		● Off	⋮
<input type="checkbox"/>	dataprep-staging-40b171b3-998c-4f92-ba43-e905ba2b96d2	Multi-Regional	US	None		● Off	⋮
<input type="checkbox"/>	progetto-1-191608.appspot.com	Multi-Regional	US	None		● Off	⋮

The screenshot shows the 'Create a bucket' form in the Google Cloud Platform Storage browser. The page title is 'Create a bucket'. The form includes a 'Name' field with a placeholder and a note: 'Must be unique across Cloud Storage. If you're serving website content, enter the website domain as the name.' Below the name field is a 'Default storage class' section with radio buttons for 'Multi-Regional' (selected), 'Regional', 'Nearline', and 'Coldline'. There is a 'Location' dropdown menu currently set to 'United States'. At the bottom, there is a summary table of costs and operations.

Storage cost	Retrieval cost	Class A operations [?]	Class B operations [?]
\$0.026 per GB-month	Free	\$0.005 per 1,000 ops	\$0.0004 per 1,000 ops

At the bottom left, there is a link to 'Show advanced settings'.

Google Cloud Platform MainProject

Browser UPLOAD FILES UPLOAD FOLDER CREATE FOLDER

Filter by prefix...

Buckets / cnn-hwr

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified
<input type="checkbox"/>	app.yaml	119 B	application/octet-stream	Multi-Regional	3/14/18, 9:56 AM
<input type="checkbox"/>	cart.py	393 B	application/octet-stream	Multi-Regional	3/14/18, 8:51 AM
<input type="checkbox"/>	cnn_hwr.py	5.73 KB	application/octet-stream	Multi-Regional	2/19/18, 9:26 AM
<input type="checkbox"/>	optest.py	23 B	application/octet-stream	Multi-Regional	3/14/18, 9:29 AM

Google Cloud Platform MainProject

Browser CREATE BUCKET REFRESH DELETE

Filter by prefix... Columns

Buckets

<input type="checkbox"/>	Name	Default storage class	Location	Lifecycle
<input type="checkbox"/>	cnn-hwr	Multi-Regional	US	None
<input type="checkbox"/>	dataprep-staging-40b171b3-998c-4f92-ba43-e905ba2b96d2	Multi-Regional	US	None
<input type="checkbox"/>	progetto-1-191608.appspot.com	Multi-Regional	US	None
<input type="checkbox"/>	staging.progetto-1-191608.appspot.com	Multi-Regional	US	Enabled

Google Cloud Platform MainProject

SQL Create a MySQL Second Generation instance

Instance ID
Choice is permanent. Use lowercase letters, numbers, and hyphens. Start with a letter.

Root password
Set a password for the root user. [Learn more](#)

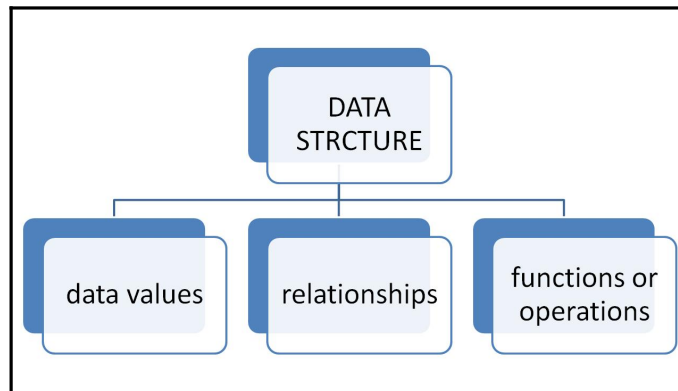
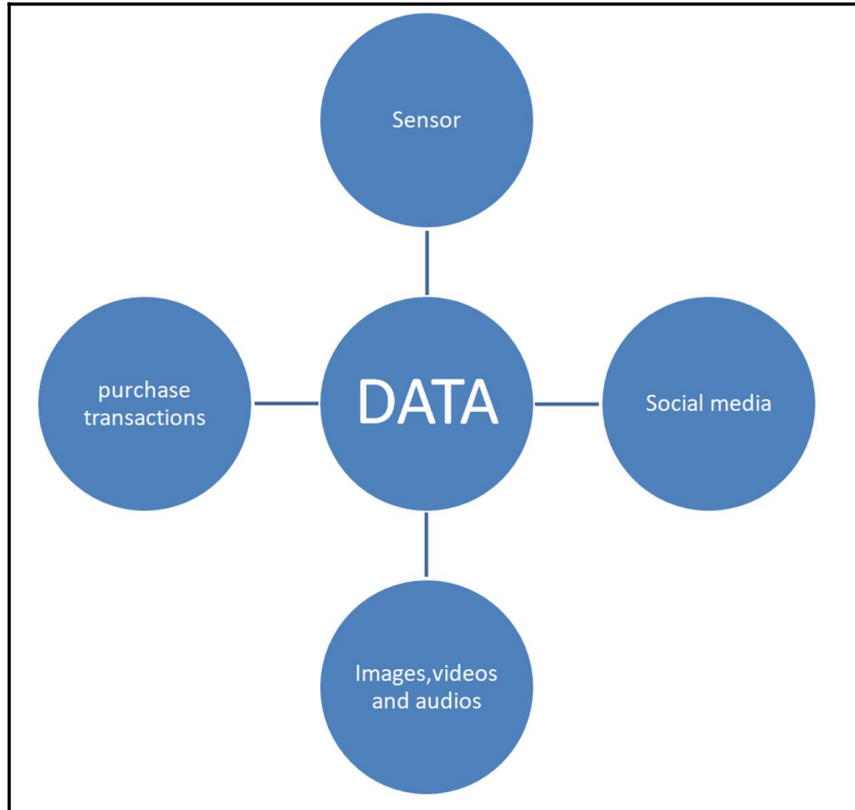
No password

Location ?
For better performance, keep your data close to the services that need it.

Region Choice is permanent	Zone Can be changed at any time
<input type="text" value="us-central1"/>	<input type="text" value="Any"/>

⌵ Show configuration options

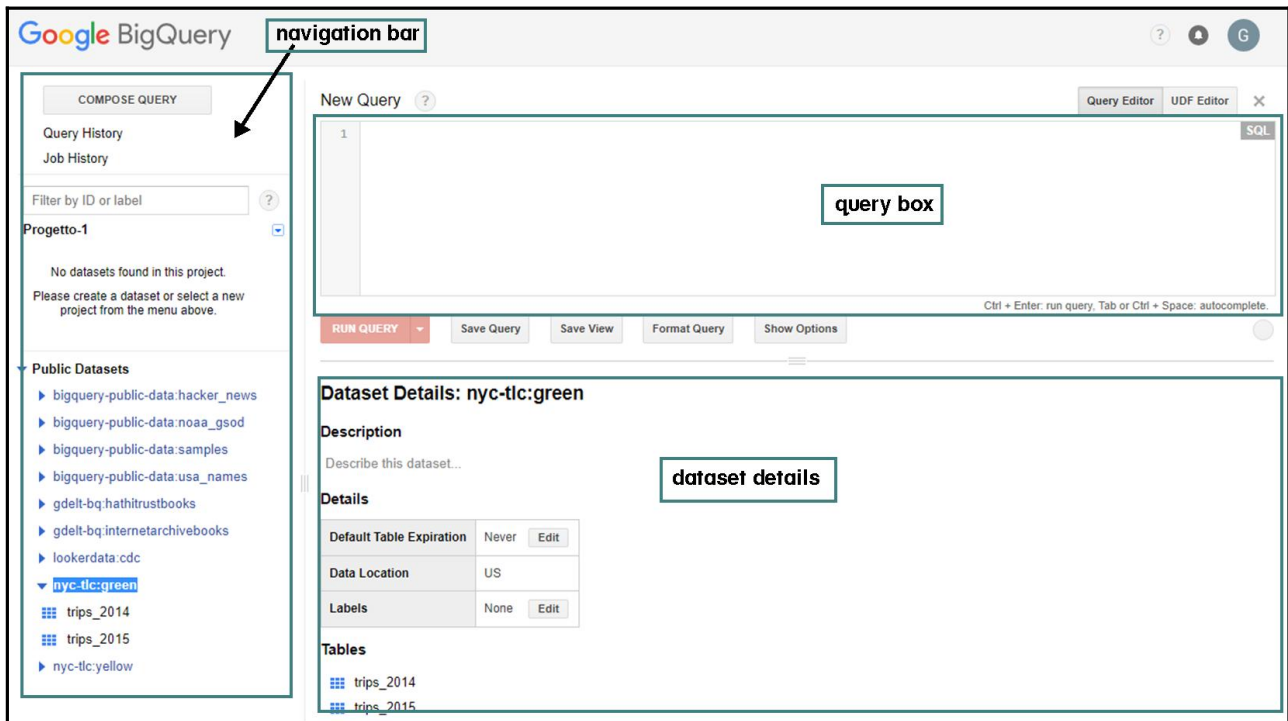
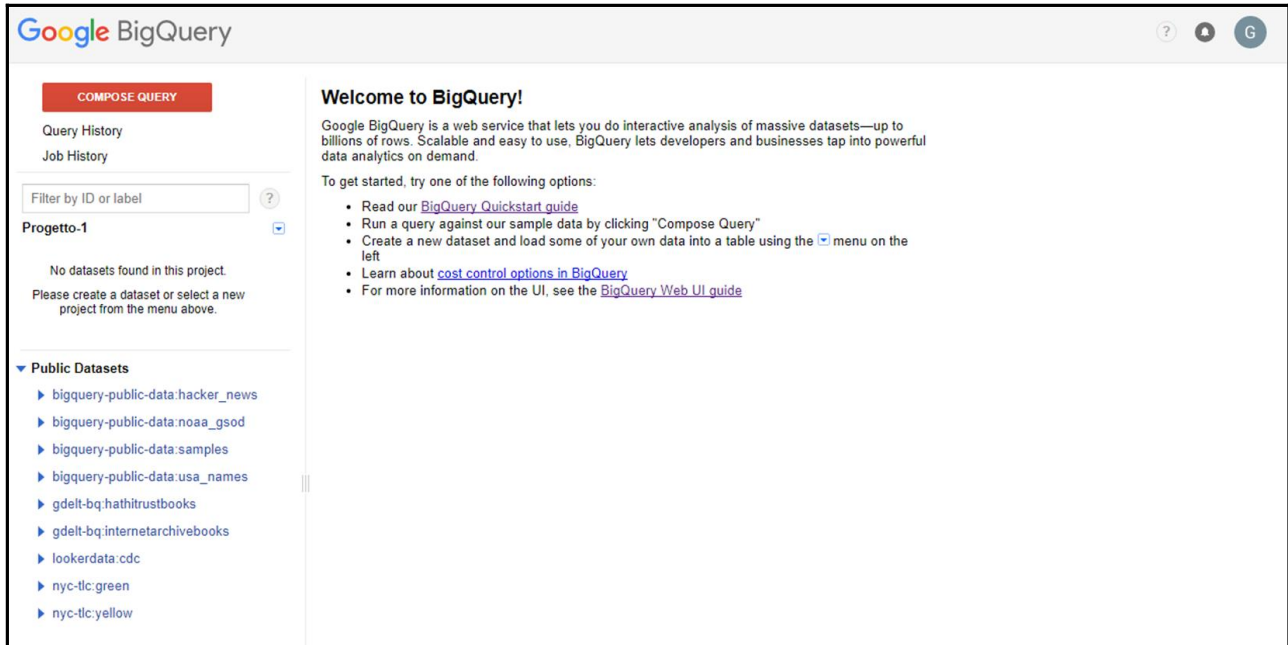
Chapter 4: Querying Your Data with BigQuery



PRIMARY KEY

ID	Surname	Name	Gender	City
1	Red	John	M	Boston
2	Black	Asia	F	Chicago
3	Blue	Mark	M	San Francisco

	A	B	C	D	E
1	N	Museum	City	Visitors2016	Visitors2015
2	1	Colosseo e Foro Romano	ROMA	6408852	6551046
3	2	Scavi di Pompei	POMPEI	3283740	2934010
4	3	Galleria degli Uffizi	FIRENZE	2010631	1971758
5	4	Galleria dell'Accademia di Firenze	FIRENZE	1461185	1415397
6	5	Castel Sant'Angelo	ROMA	1234443	1047326
7	6	Venaria Reale	VENARIA R.	1012033	580786
8	7	Museo Egizio di Torino	TORINO	881463	863535
9	8	Circuito Museale Boboli ...	FIRENZE	852095	772934
10	9	Reggia di Caserta	CASERTA	683070	497197
11	10	Galleria Borghese	ROMA	527937	506442
12					



Google BigQuery

COMPOSE QUERY

Query History
Job History

Filter by ID or label

Progetto-1

No datasets found in this project.
Please create a dataset or select a new project from the menu above.

Public Datasets

- bigquery-public-data:hacker_news
- bigquery-public-data: NOAA_gsod
- bigquery-public-data:samples
- bigquery-public-data:usa_names
- gdelt-bq:hathitrustbooks
- gdelt-bq:internetarchivebooks
- lookerdata:cdc
- nyc-tlc:green
 - trips_2014
 - trips_2015
- nyc-tlc:yellow
 - trips

New Query

```

1 #standardSQL
2 SELECT
3   TIMESTAMP_TRUNC(pickup_datetime,
4     MONTH) month,
5   COUNT(*) trips
6 FROM
7   `bigquery-public-data.new_york.tlc_yellow_trips_2015`
8 GROUP BY
9   1
10 ORDER BY
11  1
  
```

Query Editor UDF Editor

SQL

Ctrl + Enter: run query. Tab or Ctrl + Space: autocomplete.

Query complete (4.5s elapsed, 1.09 GB processed)

RUN QUERY Save Query Save View Format Query Show Options

Download as CSV Download as JSON Save as Table Save to Google Sheets

Results Details

Row	month	trips
1	2015-01-01 00:00:00.000 UTC	12748986
2	2015-02-01 00:00:00.000 UTC	12450521
3	2015-03-01 00:00:00.000 UTC	13351609
4	2015-04-01 00:00:00.000 UTC	13071789
5	2015-05-01 00:00:00.000 UTC	13158262
6	2015-06-01 00:00:00.000 UTC	12324935
7	2015-07-01 00:00:00.000 UTC	11562783
8	2015-08-01 00:00:00.000 UTC	11130304
9	2015-09-01 00:00:00.000 UTC	11225063
10	2015-10-01 00:00:00.000 UTC	12315488
11	2015-11-01 00:00:00.000 UTC	11312676
12	2015-12-01 00:00:00.000 UTC	11460573

Table JSON

Untitled Report

File View Page Help

Toolbar

To select an existing data

Add a data source

A data source provides data for charts. Select an existing data source or click CREATE NEW DATA SOURCE.

OKAY, GOT IT

Select Data Source

- [Sample] World Population Data 2...
- [Sample] Google Analytics Data
- [Sample] Firebase Analytics Data (...)
- [Sample] Firebase Analytics Data (...)
- [Sample] Firebase Analytics Data (...)
- [Sample] AdWords Data
- [Sample] YouTube Data
- [Sample] Rio Olympics Data
- [Sample] Search Console Data (Sit...
- [Sample] Search Console Data (UR...
- [Sample] Firebase Analytics Data ...

CREATE NEW DATA SOURCE

To create new data source

You are about to add a data source to this report

[Sample] World Population Data 2005 - 2014



Note that **Report Editors** can create charts using the new data source(s), and can add dimensions and metrics not currently included in the report.

CANCEL

ADD TO REPORT

Google Data Studio beta Home

Start a new report

ALL TEMPLATES

blank report

Acme Marketing
Google Analytics

Search Console Report
Search Console

AdWords Overview
Google Adwords

ALL OWNED BY ME SHARED WITH ME TRASH Search

REPORTS

DATA SOURCES

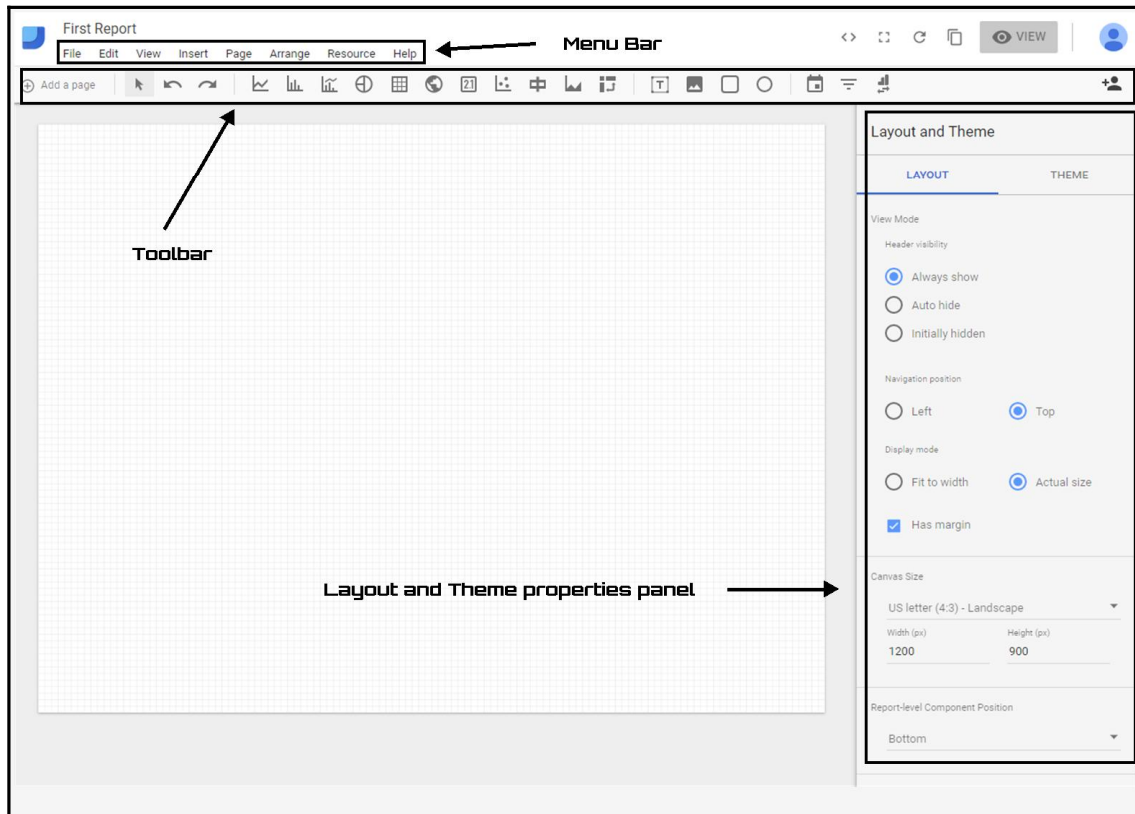
New Features!

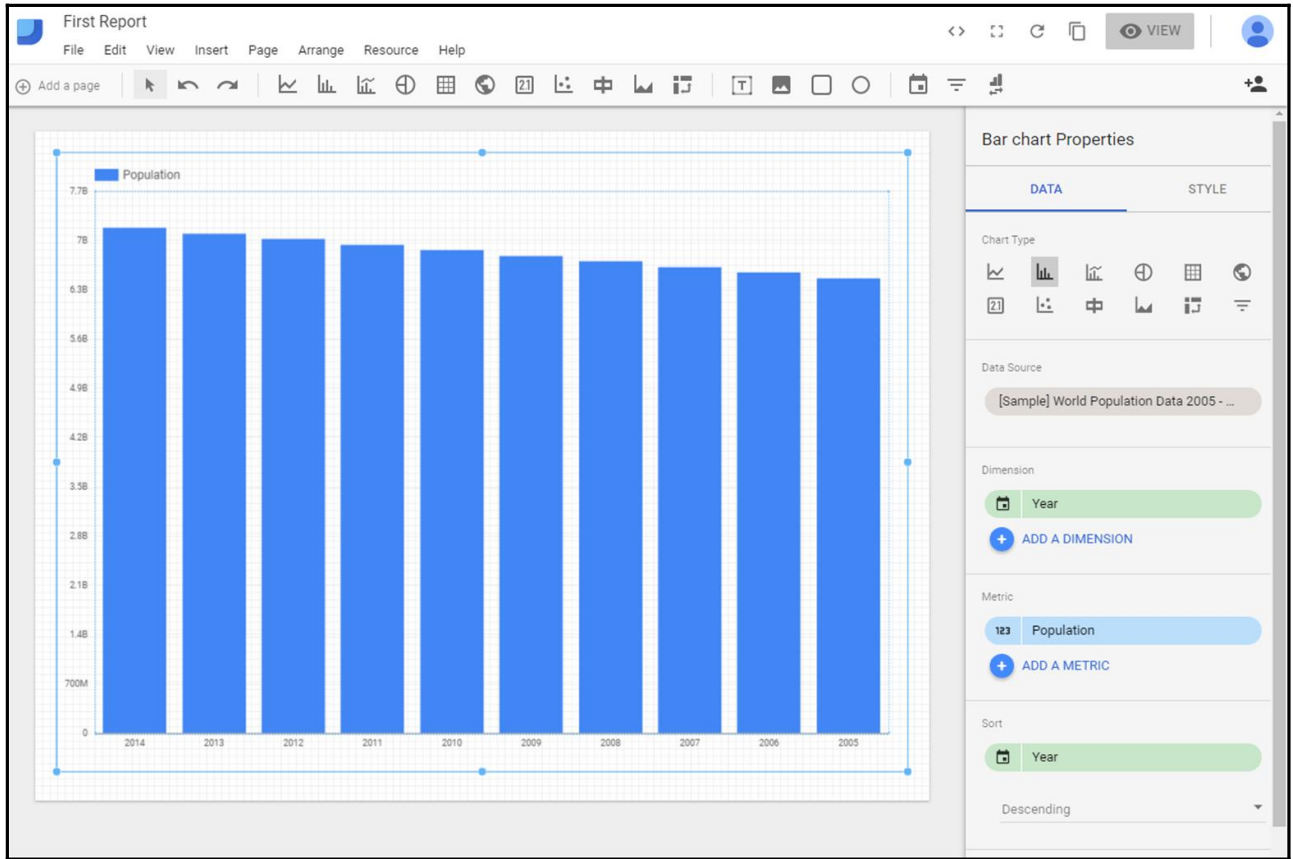
Video tutorials

User settings

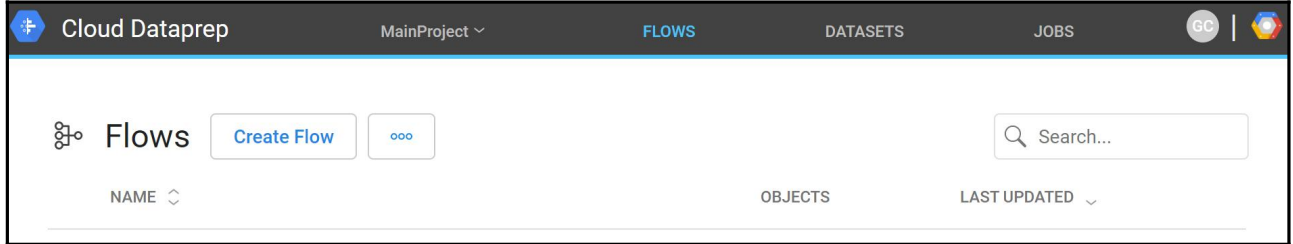
Today	Owner	Last opened by me
Welcome to Data Studio! (Start here)	Google Data Studio	9:07 AM
[MAKE A COPY] Data Studio Template Seer Int...	Seer Interactive	Jan 31, 2018
Stack Overflow Questions	Data Studio Master Account	Jan 10, 2018

blank report





Chapter 5: Transforming Your Data



The 'Create Flow' dialog box is shown, containing the following fields and buttons:

- Flow Name:** Data Wrangling
- Flow Description:** Cleaning and preparing the data
- Buttons:** Cancel and Create

CleaningData CleaningData Flow • Full Data Run Job GC

Grid Columns Find column Filters

name	gender	#	age	#	right1	#	wrong
9 Categories	2 Categories	10 - 32	5 - 98	2 - 95			
Emma	F	24	80	20			
Olivia	F	32	75	25			
Noah	M	15	60	40			
Ava	F	18	45	55			
Mason	M	21	54	46			
Isabella	F	28	19	85			
Lucas	M	30	13	87			
Elijah	M	10	98	2			
Mia	F	22	5	95			

column's data quality bar

column's data histogram

CleaningData CleaningData Flow • Full Data Run Job GC

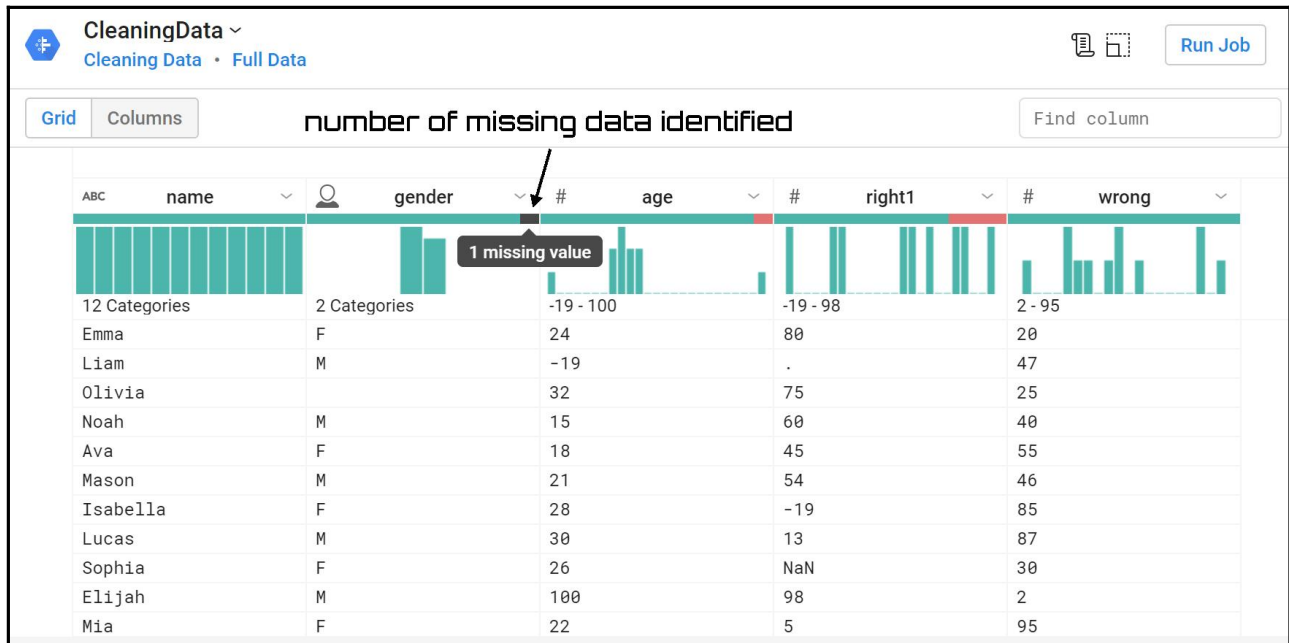
Grid Columns Find column Filters

gender column menu

name	gender	#	age	#	right1	#	wrong
9 Categories	2 Categories	10 - 15 1 11.11%	2 - 95				
Emma	F	80	20				
Olivia	F	75	25				
Noah	M	60	40				
Ava	F	45	55				
Mason	M	54	46				
Isabella	F	19	85				
Lucas	M	13	87				
Elijah	M	98	2				
Mia	F	5	95				

- Rename
- Change type >
- Move >
- Edit column >
- Column Details
- Show related steps
- Find >
- Filter >
- Clean >
- Formula >
- Aggregate >

5 Columns 9 Rows 3 Data Types



suggestions panel

suggestions preview

gender	any_gender
F	F
M	F
M	F
F	F
M	F
F	F
M	F
F	F
M	F
M	F

Affects 1 column, all rows Creates 1 column

Suggestions

- Delete columns
 - gender
- Rename
 - Rename gender to 'newColumnName'
- Aggregate and group data
 - COUNT(), ANY(gender)
- Create a new column
 - ANY(gender)

Source	to be dropped	Preview
	gender	gender
2 Categories		
F		F
M		M
		NA
M		M
F		F
M		M
F		F
M		M
F		F
M		M
F		F

missing value is replaced with NA value

< Recipe Edit Step ×

Columns required

gender ×

Column

Formula required

ifmissing(\$col, 'NA')

Group by

Column

Order by ?

Edit formula

applied formula

Source	to be dropped	Preview
#	age	#
-19 - 100		
24		24
-19		NA
32		32
15		15
18		18
21		21
28		28
30		30
26		26
100		100
22		22

negative value is replaced with NA value

< Recipe Add Step ×

Transformation

Apply formula

Sets the values of one or more columns to the result of a formula [set]

Columns ? required

age ×

Column

Formula required

IF(\$col<0, 'NA', \$col)

Transform Builder

ABC	name	gender	#	age	#	right1	#	wrong
	12 Categories	2 Categories	-19 - 100	-19 - 98	3 mismatched values	2 - 95		
	Emma	F	24	80		20		
	Liam	M	-19	.	←	47		
	Olivia	NA	32	75		25		
	Noah	M	15	60		40		
	Ava	F	18	45		55		
	Mason	M	21	54		46		
	Isabella	F	28	-19		85		
	Lucas	M	30	13		87		
	Sophia	F	26	NaN	←	30		
	Elijah	M	100	98		2		
	Mia	F	22	5		95		
	Oliver	M	NA	NaN	←	21		

CleaningData
Run Job

Grid Columns
Find column
Filters

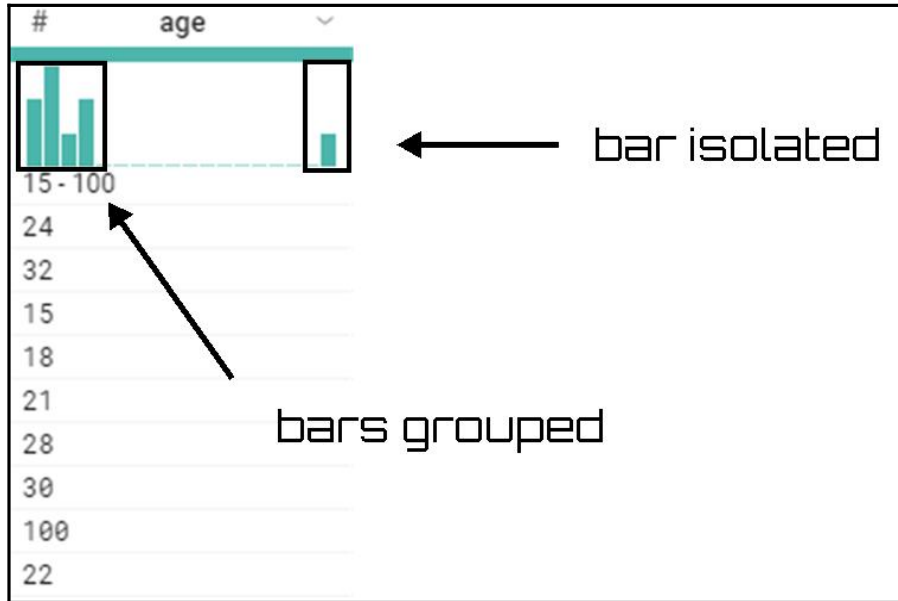
ABC	name	gender	#	age	#	right1	#	wrong
	9 Categories	2 Categories	15 - 100	5 - 98		2 - 95		
	Emma	F	24	80		20		
	Olivia	F	32	75		25		
	Noah	M	15	60		40		
	Ava	F	18	45		55		
	Mason	M	21	54		46		
	Isabella	F	28	19		85		
	Lucas	M	30	13		87		
	Elijah	M	100	98		2		
	Mia	F	22	5		95		

New Step Recipe

- 1 Set gender to IFMISSING(\$col, 'NA')
- 2 Set age to IF(\$col < 0, 'NA', \$col)
- 3 Delete rows where ISMISMATCHED(right1, ['Integer'])
- 4 Set gender to IFMISMATCHED(\$col, ['Gender'], 'F')
- 5 Set right1 to IF(\$col == -19, 19, \$col)

preview of the dataset with changes made

↑ Actions planned in the Recipe panel



Grid Columns Find column Filters

age

Overview Patterns

SUMMARY

Valid	9	100.0%
Unique	9	100.0%
Outliers	1	11.1%
Mismatched	0	0.0%
Missing	0	0.0%

STATISTICS

Minimum	15.00
Lower Quartile	21.00
Median	24.00
Upper Quartile	30.00
Maximum	100.00

TOP VALUES

100	1
15	1
18	1
21	1
22	1
24	1
28	1
30	1
32	1

MISMATCHED VALUES

None

OUTLIERS

100	1
-----	---

5 Columns 9 Rows 3 Data Types

Source	to be dropped	Preview	
#	age	#	age
15 - 100		15 - 32	
24		24	
32		32	
15		15	
18		18	
21		21	
28		28	
30		30	
100		10	
22		22	

< Recipe Add Step X

Transformation

Apply formula

Sets the values of one or more columns to the result of a formula [set]

Columns required

age

Column

Formula required

if(\$col == 100, '10', \$col)

100 is replaced with 10

CleaningData Flow > CleaningData Job 1 of 1
GC

View Dependencies
Export Results

100% Valid

0% Mismatched

0% Missing

5 Columns

9 Rows

Job Status: Complete

Job ID: 99002

Environment: Google Dataflow

Dataset: CleaningData

Launch Time: Today at 12:30 ...

Finish Time: Today at 12:35 ...

Duration: 5 minutes

Execution: Manual

CSV < 1kB

ABC	name	gender	#	age	#	right1	#	wrong	
Valid	9	Valid	9	Valid	9	Valid	9	Valid	9
Mismatched	0	Mismatched	0	Mismatched	0	Mismatched	0	Mismatched	0
Empty	0	Empty	0	Empty	0	Empty	0	Empty	0
Top 9 values		Top 2 values							
Noah	1	F	5						
Mia	1	M	4						
Mason	1								
Isabella	1								
Ava	1								
Elijah	1								
Emma	1								
Lucas	1								
Olivia	1								
				Minimum	18	Minimum	5	Minimum	2
				Lower quartile	18	Lower quartile	18	Lower quartile	24
				Median	22	Median	50	Median	43
				Upper quartile	25	Upper quartile	64	Upper quartile	63
				Maximum	32	Maximum	98	Maximum	95

ABC	column	#	Ozone	#	Solar_R	##	Wind	#	Temp	#	Month	#	Day
	153 Categories	1 - 168		7 - 334		2 - 21		56 - 97		5 - 9		1 - 31	
	"1"	41		190		7.4		67		5		1	
	"2"	36		118		8		72		5		2	
	"3"	12		149		12.6		74		5		3	
	"4"	18		313		11.5		62		5		4	
	"5"	NA		NA		14.3		56		5		5	
	"6"	28		NA		14.9		66		5		6	
	"7"	23		299		8.6		65		5		7	
	"8"	19		99		13.8		59		5		8	
	"9"	8		19		20.1		61		5		9	
	"10"	NA		194		8.6		69		5		10	
	"11"	7		NA		6.9		74		5		11	
	"12"	16		256		9.7		69		5		12	
	"13"	11		290		9.2		66		5		13	
	"14"	14		274		10.9		68		5		14	
	"15"	18		65		13.2		58		5		15	
	"16"	14		334		11.5		64		5		16	
	"17"	34		307		12		66		5		17	

Grid Columns
Find column
Filters

##	Ozone	##	Solar_R	##	Wind	##	Temp	#	Month
0.00 - 1.00	0.00 - 1.00	0.00 - 1.00	0.00 - 1.00	0.00 - 1.00	0.00 - 1.00	0.00 - 1.00	0.00 - 1.00	5 - 9	
0.23952095808383234	0.5596330275229358	0.2771739130434783	0.25					5	
0.20958083832335328	0.3394495412044037	0.3097826086956522	0.375					5	
0.0658682634730539	0.43425076452599307	0.5597826086956522	0.425					5	
0.10179640718562874	0.9357798165137615	0.5	0.125					5	
0.1317365269461078	0.8929663688562691	0.3423913043478261	0.2					5	
0.10778443113772455	0.28134556574923547	0.625	0.05					5	
0.041916167664670656	0.03669724770642202	0.9673913043478262	0.1					5	
0.08982035928143713	0.7614678899082569	0.40217391304347827	0.3					5	
0.059880239520958084	0.8654434250764526	0.375	0.225					5	
0.07784431137724551	0.8165137614678899	0.4673913043478262	0.275					5	
0.10179640718562874	0.17737003058103976	0.592391304347826	0.025					5	
0.07784431137724551	1	0.5	0.175					5	
0.19760479041916168	0.9174311926605505	0.5271739130434783	0.225					5	
0.029940119760479042	0.21712538226299694	0.875	0					5	
0.17365269461077845	0.963302752293578	0.5	0.275					5	
0.059880239520958084	0.11314984709400122	0.40217391304347827	0.125					5	

New Step Recipe

- Delete rows where ISMISMATCHED(Ozone, ['Integer'])
- Delete rows where ISMISMATCHED(Solar_R, ['Integer'])
- Set Ozone to (Ozone - MIN(Ozone)) / (MAX(Ozone) - MIN(Ozone))
- Set Solar_R to (Solar_R - MIN(Solar_R)) / (MAX(Solar_R) - MIN(Solar_R))
- Set Wind to (Wind - MIN(Wind)) / (MAX(Wind) - MIN(Wind))
- Set Temp to (Temp - MIN(Temp)) / (MAX(Temp) - MIN(Temp))

7 Columns 111 Rows 3 Data Types

Grid Columns
Find column
Filters

##	Ozone	##	Solar_R	##	Wind	##	Temp
-1.2 - 3.8		-2.0 - 1.6		-2.2 - 3.0		-2.2 - 2.0	
-0.03317961116176968		0.057286242845388184		-0.7170778435439346		-1.1376469142641918	
-0.18411964554522905		-0.736182826167326		-0.547665341309007		-0.6106068162536531	
-0.9086318105858342		-0.3945503103424074		0.7511638424921047		-0.3997907778494375	
-0.7275037693256828		1.4127959024087748		0.44057425506140413		-1.6646870122747306	
-0.5765637349422235		1.258510250100747		-0.37825283907407947		-1.3484629534684074	
-0.697315762448991		-0.9455784971567922		1.0899888469619603		-1.980911071081054	
-1.0293838380926017		-1.8272027960598078		2.8688201204287007		-1.7700950318768385	
-0.7878797830790666		0.7846328894403761		-0.06766325164337894		-0.9268308758599763	
-0.9388198174625261		1.1593266164741578		-0.20884033683915196		-1.2430549338662997	
-0.8482557968324503		0.9830001566935547		0.27116175282647664		-1.0322388946620842	
-0.7275037693256828		-1.320264224190574		0.9205763447270322		-2.086319090683162	
-0.8482557968324503		1.6442243008708166		0.44057425506140413		-1.453870973070515	
-0.24449565929861283		1.3466734799910487		0.5817513402571771		-1.2430549338662997	
-1.0097598518459854		-1.176990975618834		2.3888108307630713		-2.1917271102852696	
-0.36524768680538033		1.511979536035364		0.44057425506140413		-1.0322388946620842	
-0.9388198174625261		-1.55169270262526155		-0.06766325164337894		-1.6646870122747306	

7 Columns 111 Rows 3 Data Types

New Step Recipe

- Delete rows where ISMISMATCHED(Ozone, ['Integer'])
- Delete rows where ISMISMATCHED(Solar_R, ['Integer'])
- Set Ozone to (Ozone - AVERAGE(Ozone)) / STDEV(Ozone)
- Set Solar_R to (Solar_R - AVERAGE(Solar_R)) / STDEV(Solar_R)
- Set Wind to (Wind - AVERAGE(Wind)) / STDEV(Wind)
- Set Temp to (Temp - AVERAGE(Temp)) / STDEV(Temp)

Grid Columns
Find column
Filters

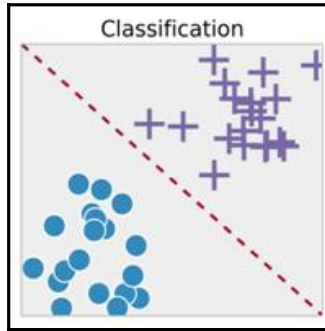
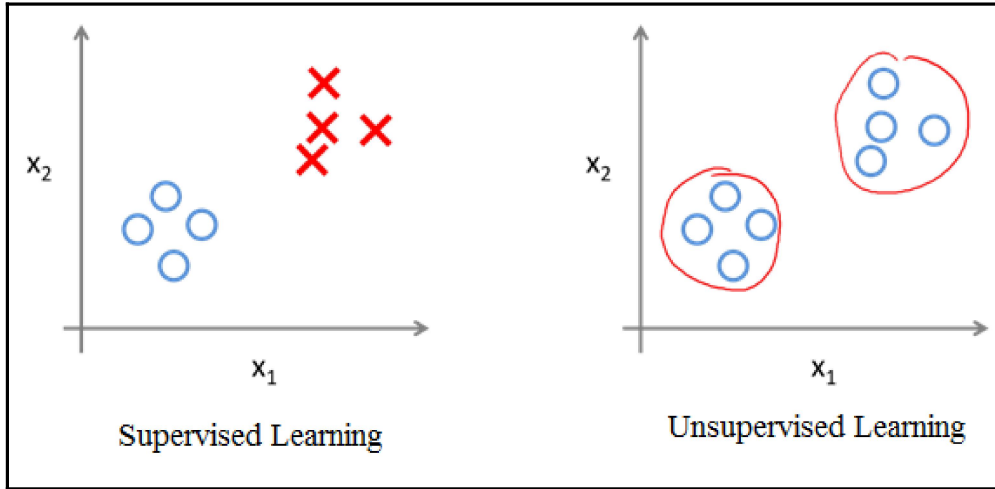
Ozone

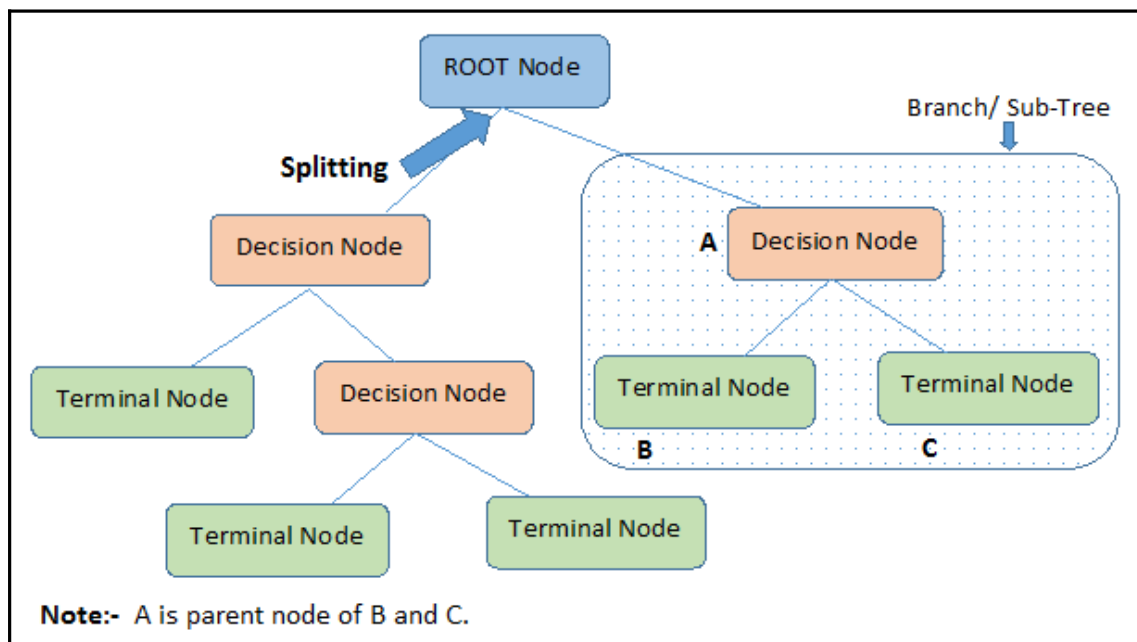
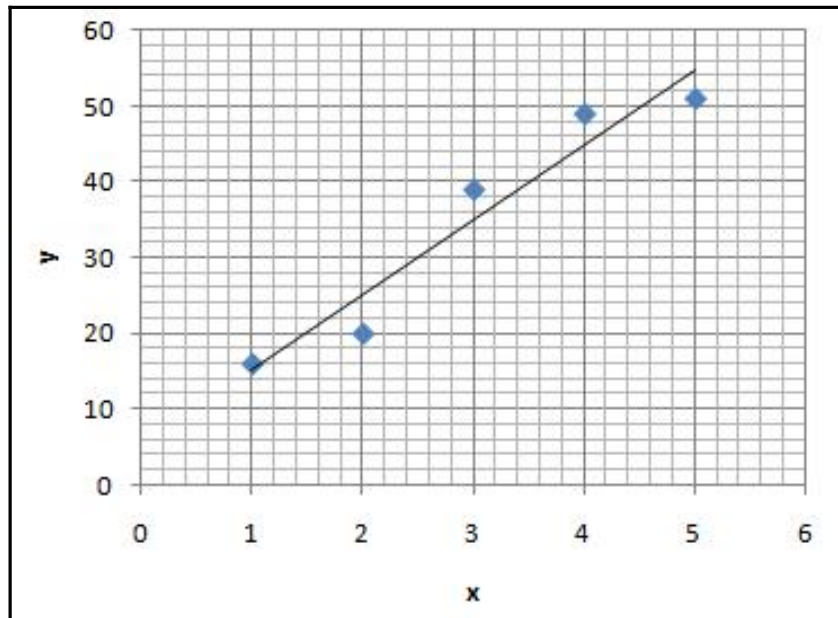
Overview Patterns

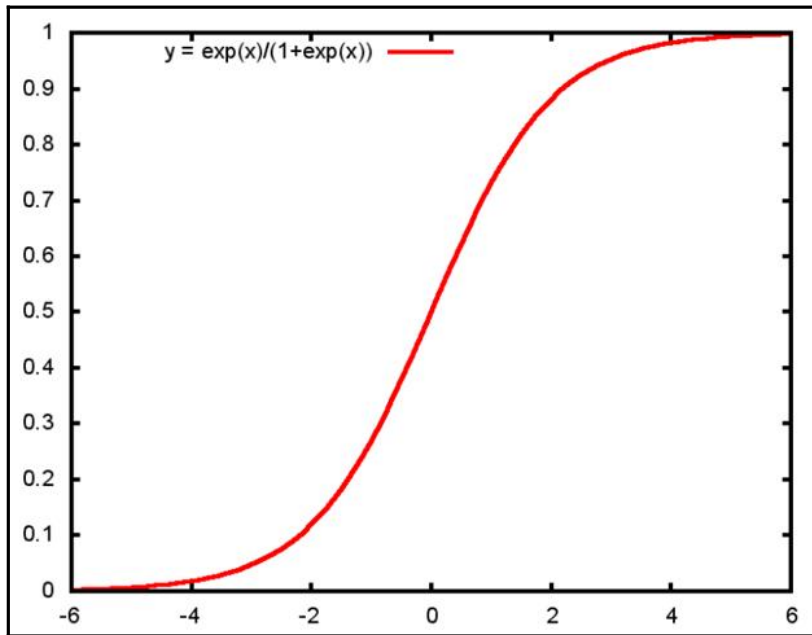
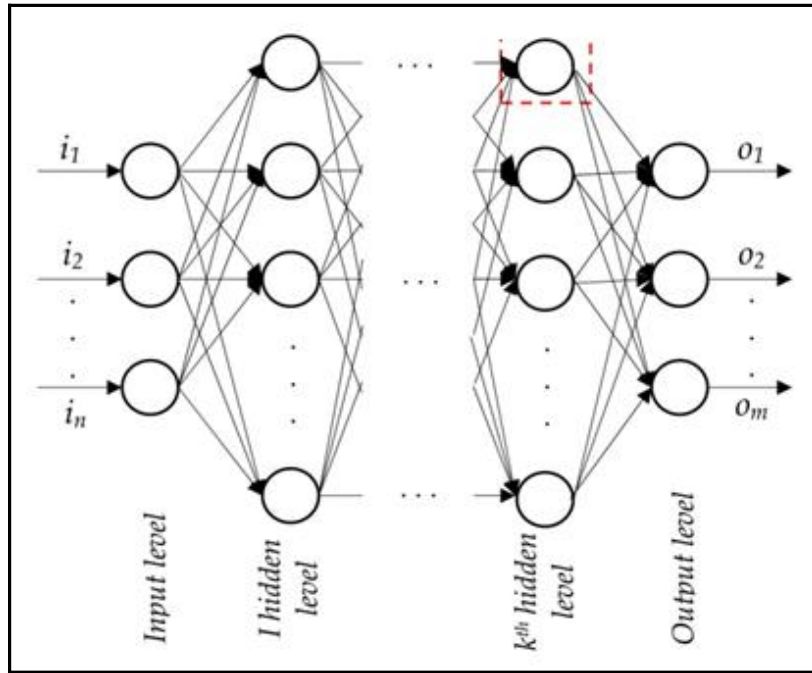
SUMMARY		TOP VALUES		MISMATCHED VALUES	
Valid	111 100.0%	-0.5765637349422235	6	None	
Unique	66 59.5%	-0.8784438037091423	4		
Outliers	1 0.9%	-0.8482557968324503	4		
Mismatched	0 0.0%	-0.7878797830790666	4		
Missing	0 0.0%	-0.7275037693256828	4		
		-0.6671277555722991	4		
		-0.6369397486956072	4		
		-0.9991958312159098	3		
		-0.9388198174625261	3		
		-0.3048716730519966	3		
		0.05738440946830596	3		
		-1.0595718449692935	2		
		-0.9086318105858342	2		
		-0.5463757280655316	2		

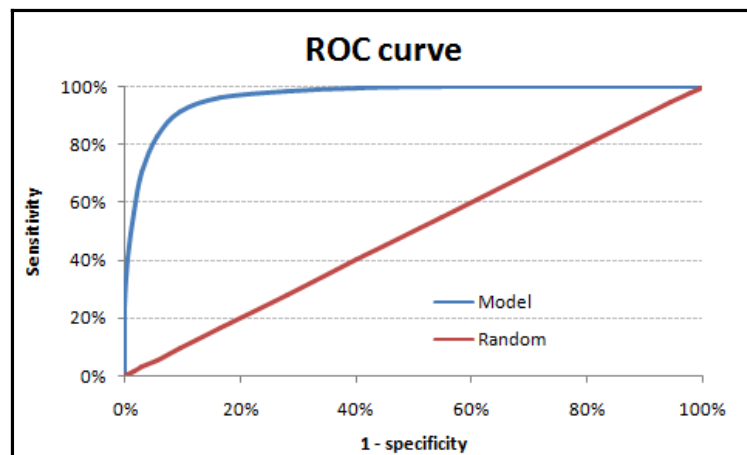
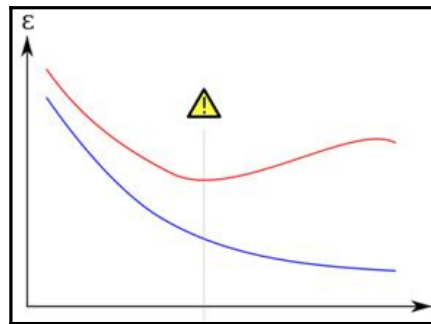
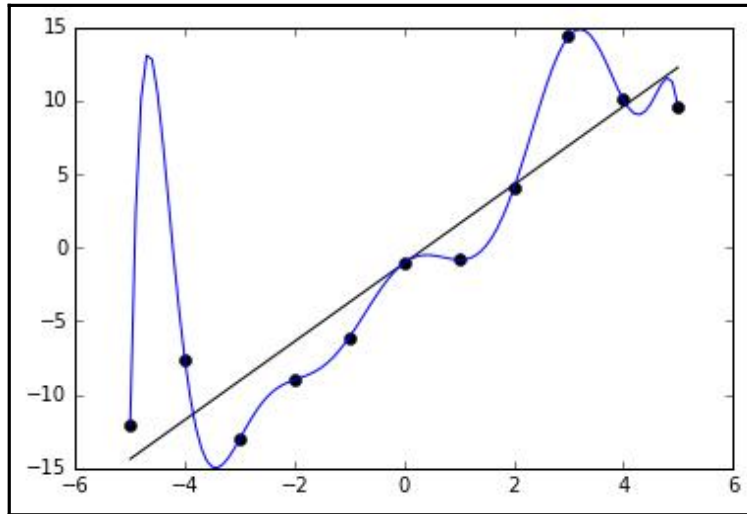
STATISTICS		OUTLIERS	
Minimum	-1.24	3.800697262178099	1
Lower Quartile	-0.73		
Median	-0.34		
Upper Quartile	0.60		
Maximum	3.80		
Average	-0.00		
Standard Deviation	1.00		

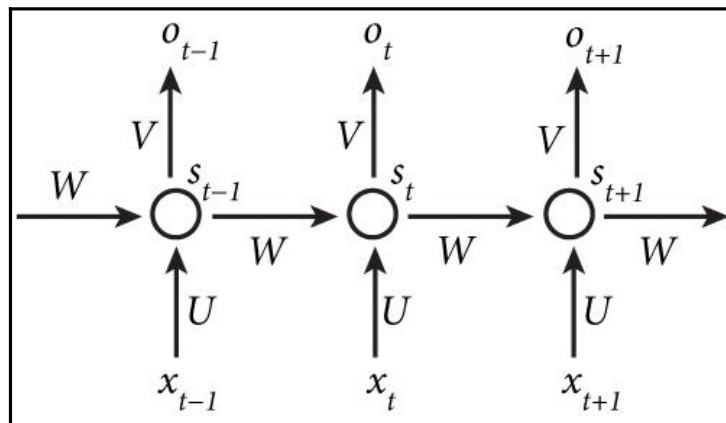
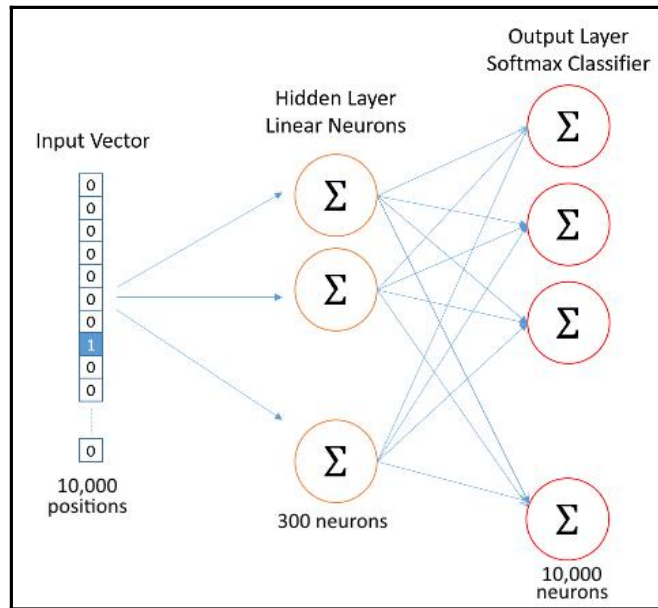
Chapter 6: Essential Machine Learning

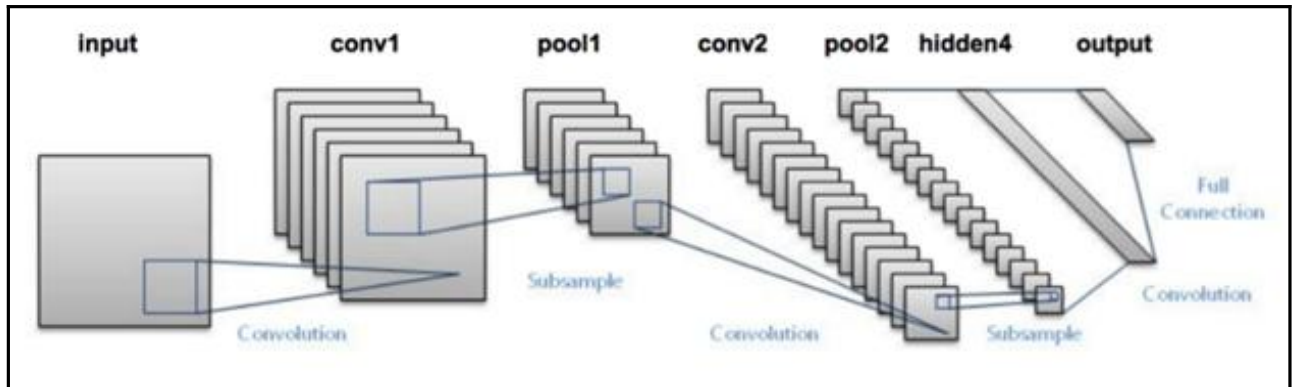
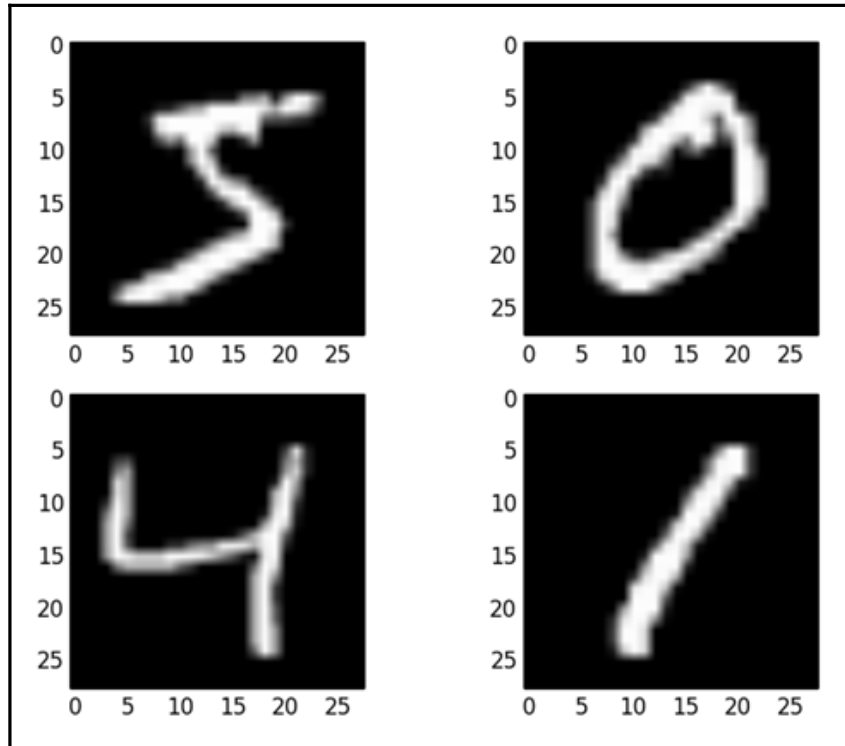




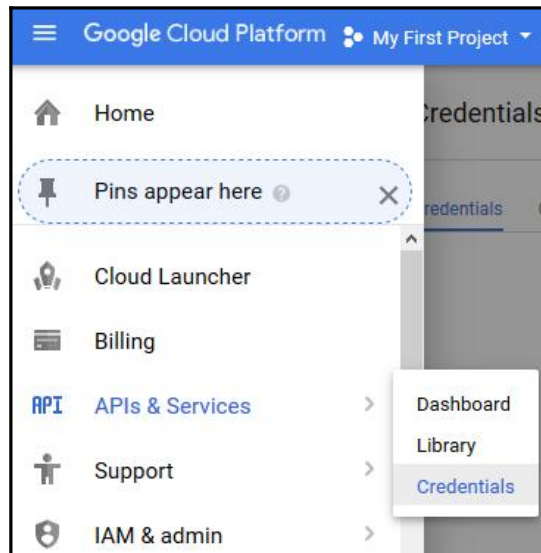
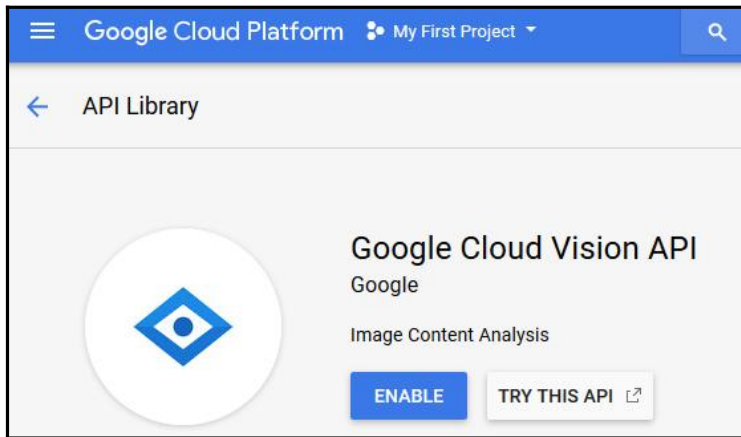


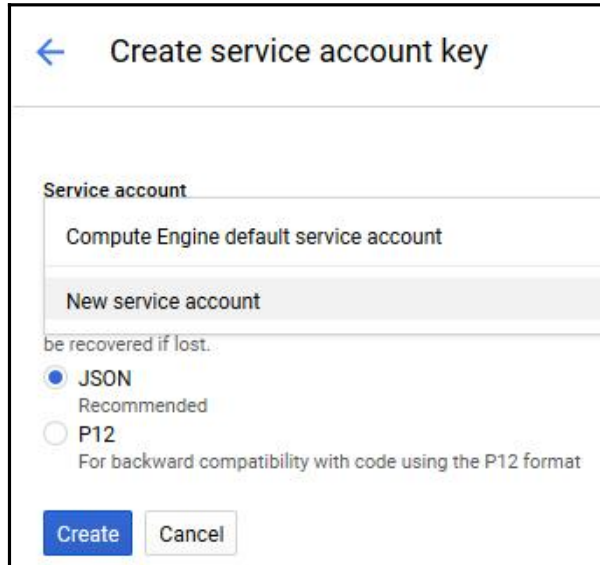
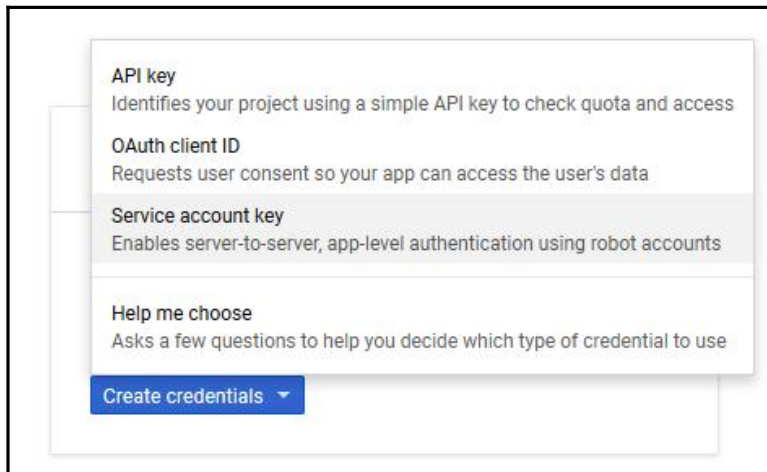


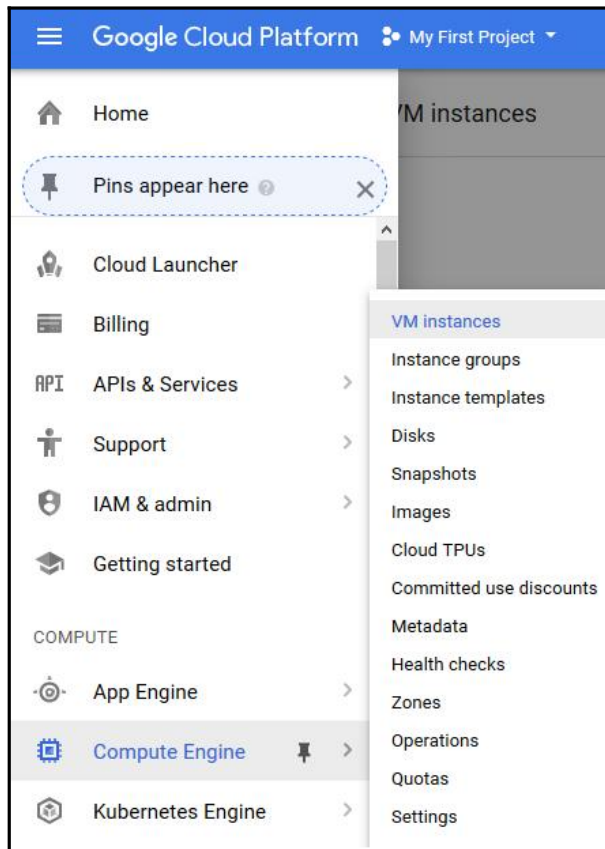
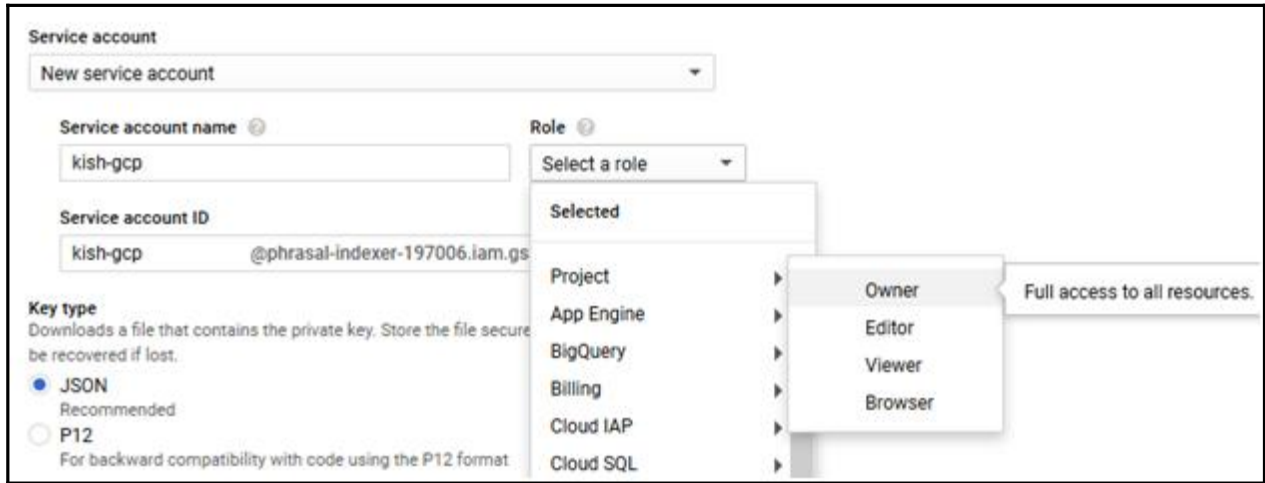




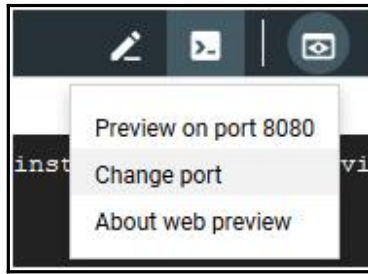
Chapter 7: Google Machine Learning APIs





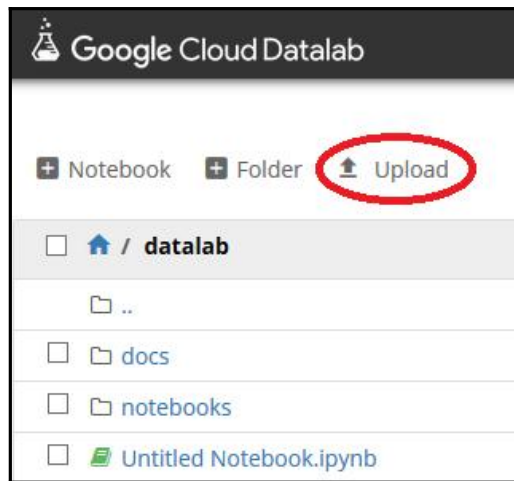


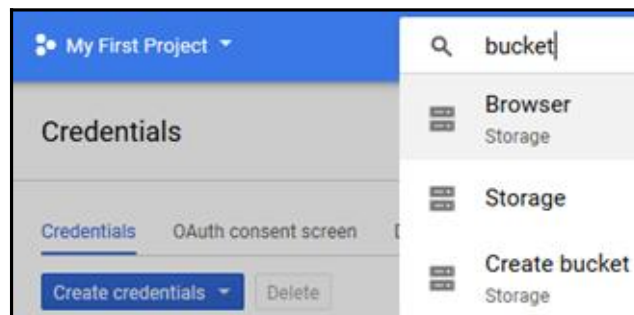
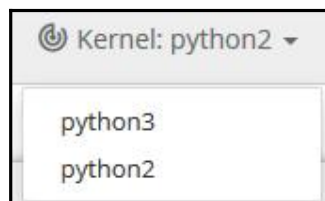
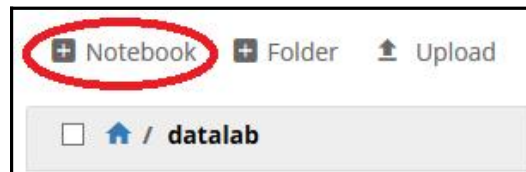
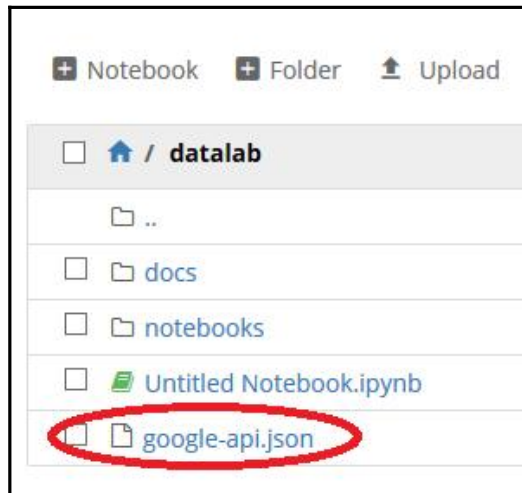
```
Cloud Shell
phrasal-indexer-197006 x +
Welcome to Cloud Shell! Type "help" to get started.
akishore_gcp@phrasal-indexer-197006:~$ datalab create --no-create-repository mlgcp
```



Change Preview Port

Port Number:





← Create a bucket

Name ⓘ
Must be unique across Cloud Storage. If you're [serving website content](#), enter the website domain as the name.

Default storage class ⓘ
[Compare storage classes](#)

Multi-Regional
 Regional
 Nearline
 Coldline

Location

United States ▼

Storage cost	Retrieval cost	Class A operations ⓘ	Class B operations ⓘ
\$0.026 per GB-month	Free	\$0.005 per 1,000 ops	\$0.0004 per 1,000 ops

⌵ [Specify labels](#)

Browser

```
%bash
gsutil cp gs://kish-bucket/11.jpg /content/datalab/

Copying gs://kish-bucket/11.jpg...
/ [1 files][ 3.4 MiB/ 3.4 MiB]
Operation completed over 1 objects/3.4 MiB.
```

Fields	
face_annotations[]	<p>FaceAnnotation</p> <p>If present, face detection has completed successfully.</p>
landmark_annotations[]	<p>EntityAnnotation</p> <p>If present, landmark detection has completed successfully.</p>
logo_annotations[]	<p>EntityAnnotation</p> <p>If present, logo detection has completed successfully.</p>
label_annotations[]	<p>EntityAnnotation</p> <p>If present, label detection has completed successfully.</p>
text_annotations[]	<p>EntityAnnotation</p> <p>If present, text (OCR) detection has completed successfully.</p>

full_text_annotation	<p>TextAnnotation</p> <p>If present, text (OCR) detection or document (OCR) text detection has completed successfully. This annotation provides the structural hierarchy for the OCR detected text.</p>
safe_search_annotation	<p>SafeSearchAnnotation</p> <p>If present, safe-search annotation has completed successfully.</p>
image_properties_annotation	<p>ImageProperties</p> <p>If present, image properties were extracted successfully.</p>
crop_hints_annotation	<p>CropHintsAnnotation</p> <p>If present, crop hints have completed successfully.</p>
web_detection	<p>WebDetection</p> <p>If present, web detection has completed successfully.</p>
error	<p>Status</p> <p>If set, represents the error message for the operation. Note that filled-in image annotations are guaranteed to be correct, even when <code>error</code> is set.</p>

```
1 response
  face_annotations {
    bounding_poly {
      vertices {
        x: 1427
        y: 279
      }
      vertices {
        x: 2176
        y: 279
      }
      vertices {
        x: 2176
        y: 1149
      }
      vertices {
        x: 1427
        y: 1149
      }
    }
    fd_bounding_poly {
      vertices {
        x: 1494
        y: 461
      }
      vertices {
        x: 2098
        y: 461
      }
      vertices {
        x: 2098
```

```
response.face_annotations[0].bounding_poly.vertices
```

```
[x: 1427  
y: 279  
 , x: 2176  
y: 279  
 , x: 2176  
y: 1149  
 , x: 1427  
y: 1149  
]
```



response_label

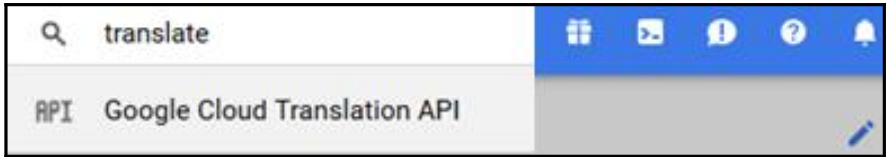
```
label_annotations {  
  mid: "/m/09g5pq"  
  description: "people"  
  score: 0.9678634405136108  
  topicality: 0.9678634405136108  
}  
label_annotations {  
  mid: "/m/0ytgt"  
  description: "child"  
  score: 0.954465389251709  
  topicality: 0.954465389251709  
}  
label_annotations {  
  mid: "/m/01k74n"  
  description: "facial expression"  
  score: 0.9380199909210205  
  topicality: 0.9380199909210205  
}  
label_annotations {  
  mid: "/m/06z04"  
  description: "skin"  
  score: 0.9294329881668091  
  topicality: 0.9294329881668091  
}
```

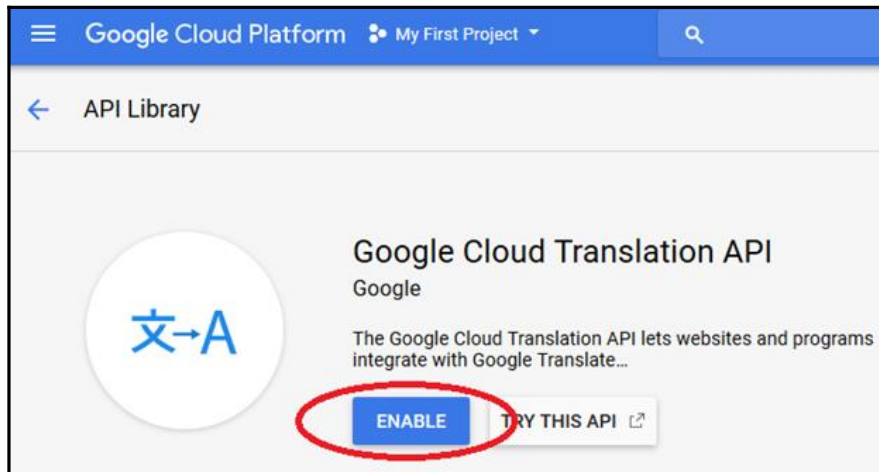
response

```
text_annotations {  
  locale: "en"  
  description: "LITTLE\nMISS\nMUCH\  bounding_poly {  
    vertices {  
      x: 1374  
      y: 1425  
    }  
    vertices {  
      x: 2151  
      y: 1425  
    }  
    vertices {  
      x: 2151  
      y: 2303  
    }  
    vertices {  
      x: 1374  
      y: 2303  
    }  
  }  
}
```

```
response
logo_annotations {
  mid: "/m/0d07ph"
  description: "Wikipedia"
  score: 0.5256200432777405
  bounding_poly {
    vertices {
      x: 31
      y: 28
    }
    vertices {
      x: 113
      y: 28
    }
    vertices {
      x: 113
      y: 124
    }
    vertices {
      x: 31
      y: 124
    }
  }
}
```

```
response
landmark_annotations {
  mid: "/m/0l8cb"
  description: "Taj Mahal"
  score: 0.940950870513916
  bounding_poly {
    vertices {
      x: 6
      y: 9
    }
    vertices {
      x: 227
      y: 9
    }
    vertices {
      x: 227
      y: 155
    }
    vertices {
      x: 6
      y: 155
    }
  }
  locations {
    lat_lng {
      latitude: 27.174698469698683
      longitude: 78.042073
    }
  }
}
```





```
Welcome to Cloud Shell! Type "help" to get started.  
akishore_gcp@phrasal-indexer-197006:~$ datalab connect mlgcp
```

```
client.get_languages()

[{'language': u'af', u'name': u'Afrikaans'},
 {'language': u'sq', u'name': u'Albanian'},
 {'language': u'am', u'name': u'Amharic'},
 {'language': u'ar', u'name': u'Arabic'},
 {'language': u'hy', u'name': u'Armenian'},
 {'language': u'az', u'name': u'Azerbaijani'},
 {'language': u'eu', u'name': u'Basque'},
 {'language': u'be', u'name': u'Belarusian'},
 {'language': u'bn', u'name': u'Bengali'},
 {'language': u'bs', u'name': u'Bosnian'},
 {'language': u'bg', u'name': u'Bulgarian'},
 {'language': u'ca', u'name': u'Catalan'},
 {'language': u'ceb', u'name': u'Cebuano'},
 {'language': u'ny', u'name': u'Chichewa'},
 {'language': u'zh', u'name': u'Chinese (Simplified)'},
 {'language': u'zh-TW', u'name': u'Chinese (Traditional)'},
 {'language': u'co', u'name': u'Corsican'},
 {'language': u'hr', u'name': u'Croatian'},
 {'language': u'cs', u'name': u'Czech'},
 {'language': u'da', u'name': u'Danish'},
 {'language': u'nl', u'name': u'Dutch'},
 {'language': u'en', u'name': u'English'},
 {'language': u'eo', u'name': u'Esperanto'},
 {'language': u'et', u'name': u'Estonian'},
 {'language': u'tl', u'name': u'Filipino'},
 {'language': u'fi', u'name': u'Finnish'},
 {'language': u'fr', u'name': u'French'},
 {'language': u'fy', u'name': u'Frisian'},
```

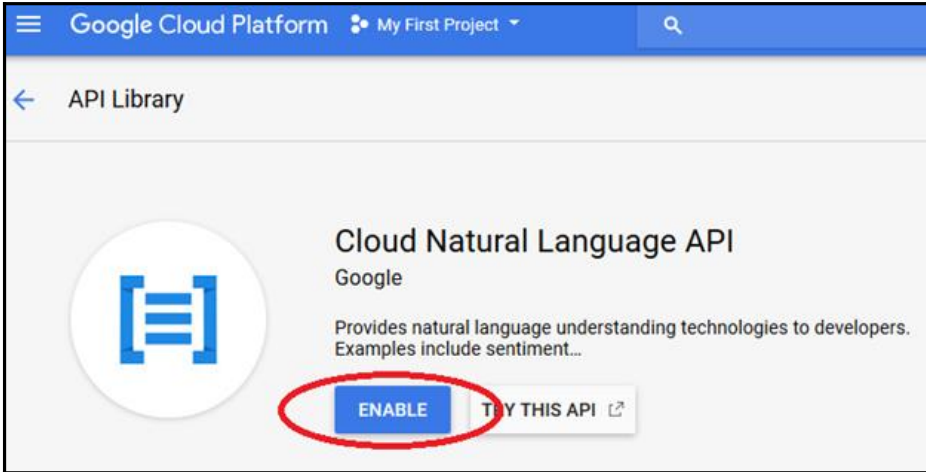
```
client.translate('koszula')

{'detectedSourceLanguage': u'pl',
 'input': 'koszula',
 'translatedText': u'shirt'}
```

```
client.translate(['Me llamo Kishore', 'My name is Kishore'], target_language='de')

[{'detectedSourceLanguage': u'es',
 'input': 'Me llamo Kishore',
 'translatedText': u'Mein Name ist Kishore'},
 {'detectedSourceLanguage': u'en',
 'input': 'My name is Kishore',
 'translatedText': u'Mein Name ist Kishore'}]
```

Q natural language|
API Cloud Natural Language API



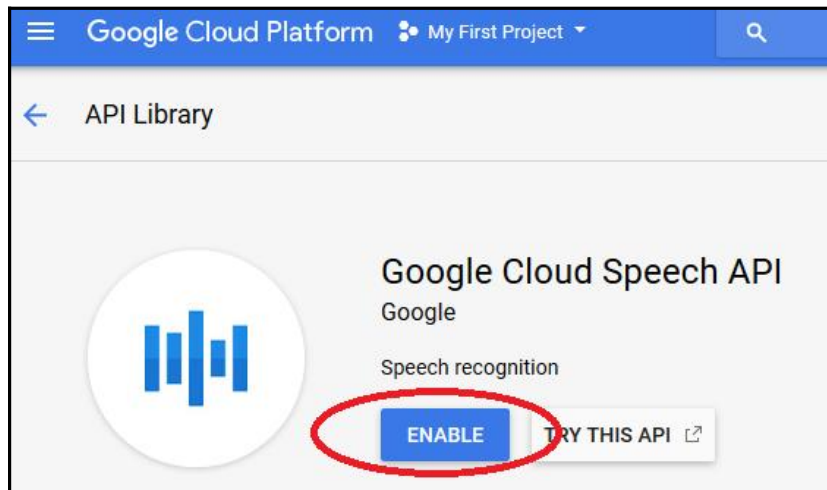
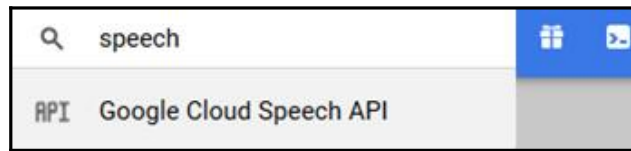
```
document = language.types.Document(  
    content=html_content,  
    language='es',  
    type='HTML',  
)
```

```
document = language.types.Document(  
    gcs_content_uri='gs://my-text-bucket/sentiment-me.txt'  
    type=language.enums.HTML,  
)
```

```
name: Michelangelo Caravaggio  
name: Italian  
name: The Calling of Saint Matthew
```



```
NOUN: Michelangelo
NOUN: Caravaggio
PUNCT: ,
ADJ: Italian
NOUN: painter
PUNCT: ,
VERB: is
VERB: known
ADP: for
PUNCT: "
DET: The
NOUN: Calling
ADP: of
NOUN: Saint
NOUN: Matthew
PUNCT: "
PUNCT: .
```



```
alternatives {
  transcript: "how are you"
  confidence: 0.897314190865
}
```

🔍 video |

RPI Cloud Video Intelligence API

```
videointelligence.enums.Feature.  
videointelligence.enums.Feature.EXPLICIT_CONTENT_DETECTION  
videointelligence.enums.Feature.FACE_DETECTION  
videointelligence.enums.Feature.FEATURE_UNSPECIFIED  
videointelligence.enums.Feature.LABEL_DETECTION  
videointelligence.enums.Feature.SHOT_CHANGE_DETECTION
```

```
Video label description: playground  
  Label category description: city  
  Segment 0: 0.0s to 15.6s  
  Confidence: 1.0
```

```
Video label description: playground slide  
  Segment 0: 0.0s to 15.6s  
  Confidence: 0.790848553181
```

```
Video label description: outdoor play equipment  
  Segment 0: 0.0s to 15.6s  
  Confidence: 0.932455003262
```

```
Video label description: leisure  
  Segment 0: 0.0s to 15.6s  
  Confidence: 0.470728754997
```

```
Shot label description: fun
  Segment 0: 13.033333s to 14.0s
  Confidence: 0.58486521244
  Segment 1: 14.033333s to 15.6s
  Confidence: 0.686410725117

Shot label description: playground
  Label category description: city
  Segment 0: 0.0s to 13.0s
  Confidence: 1.0
  Segment 1: 13.033333s to 14.0s
  Confidence: 0.959590852261
  Segment 2: 14.033333s to 15.6s
  Confidence: 0.966488361359

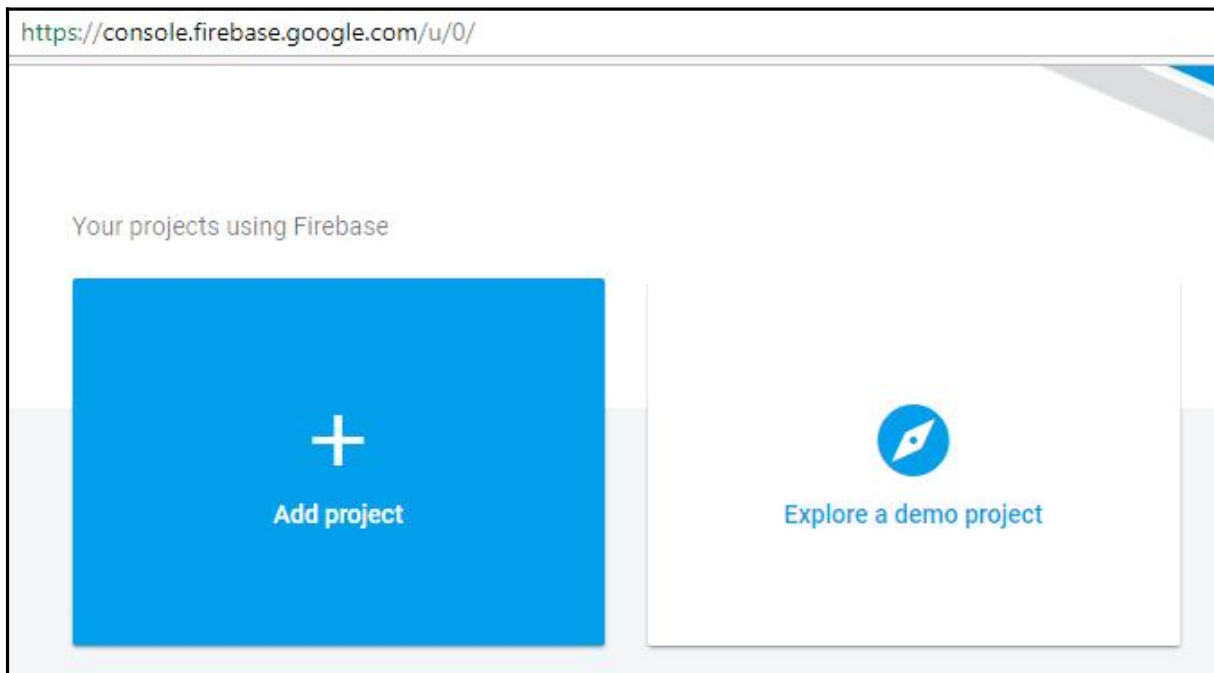
Shot label description: playground slide
  Segment 0: 0.0s to 13.0s
  Confidence: 0.784813344479
  Segment 1: 13.033333s to 14.0s
  Confidence: 0.475744605064
  Segment 2: 14.033333s to 15.6s
  Confidence: 0.530465841293
```

```
Frame label description: play
  Label category description: person
  First frame time offset: 0.951819s
  First frame confidence: 0.886474370956

Frame label description: swing
  First frame time offset: 0.951819s
  First frame confidence: 0.453329116106

Frame label description: seesaw
  Label category description: playground
  First frame time offset: 0.951819s
  First frame confidence: 0.526799023151
```

Chapter 8: Creating ML Applications with Firebase



Add a project

Project name

Project ID [?]

Country/region [?]

By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at any time. [Learn more](#)

By proceeding and clicking the button below, you agree that you are using Firebase services in your app and agree to the applicable [terms](#).

CANCEL **CREATE PROJECT**

```
Microsoft Windows [User]
(c) 2013 Microsoft Corporation

C:\Users\Admin>E:
E:\>mkdir firebase
E:\>cd firebase
E:\firebase>
```

```
Command Prompt

E:\firebase>npm install -g firebase-tools
C:\Users\Admin\AppData\Roaming\npm\firebase -> C:\Users\Admin\AppData\Roaming\node_modules\firebase-tools\bin\firebase
+ firebase-tools@3.18.4
added 1 package, removed 3 packages and updated 5 packages in 58.822s

E:\firebase>
```

```
Command Prompt - firebase init
E:\firebase>firebase init

  #####  #####  #####  #####  #####  #####  #####  #####  #####  #####
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##

You're about to initialize a Firebase project in this directory:

?                                     <Y/n>
```

```
Command Prompt - firebase init
E:\firebase>firebase init

  #####  #####  #####  #####  #####  #####  #####  #####  #####  #####
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
  ##      ##      ##      ##      ##      ##      ##      ##      ##      ##

You're about to initialize a Firebase project in this directory:

Before we get started, keep in mind:
  * You are initializing in an existing Firebase project directory
?                                     Yes
?
< > Database: Deploy Firebase Realtime Database Rules
< > Firestore: Deploy rules and create indexes for Firestore
< * > Functions: Configure and deploy Cloud Functions
> < * > Hosting: Configure and deploy Firebase Hosting sites
< > Storage: Deploy Cloud Storage security rules
```

```
=== Functions Setup
A          directory will be created in your project with a Node.js
package pre-configured. Functions can be deployed with .

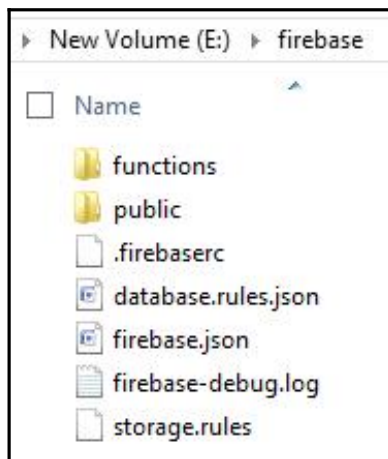
?                                               <Use arrow keys
>
> JavaScript
  TypeScript
```

```
Command Prompt - firebase init

=== Project Setup

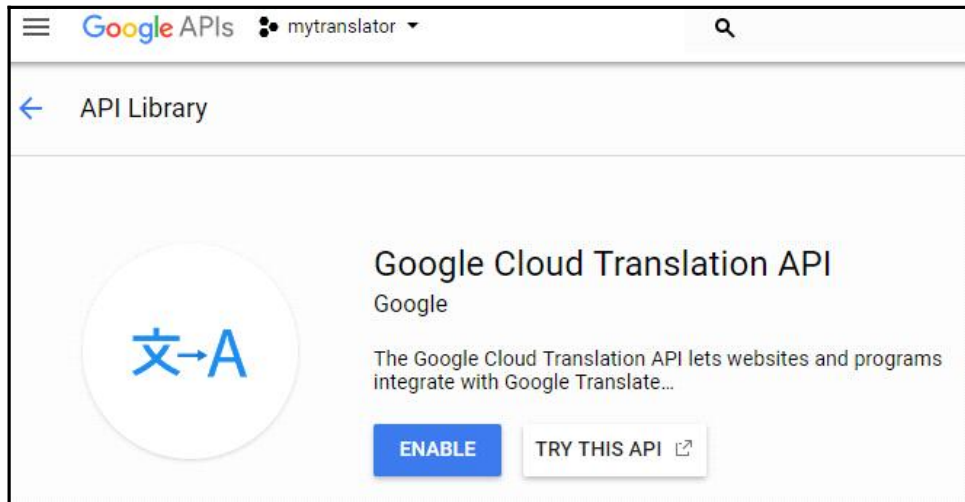
First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running
but for now we'll just set up a default project.

?
[don't setup a default project]
> mytranslator <mytranslator-c656d>
[create a new project]
```



```
?
i Writing configuration info to
i Writing project information to
+ Firebase initialization complete!
E:\firebase>
```

```
E:\firebase>cd functions
E:\firebase\functions>npm install @google-cloud/translate
+ @google-cloud/translate@1.1.0
added 12 packages in 6.67s
```



```
E:\firebase>firebase deploy --only functions
```

```
=== Deploying to 'mytranslator-c656d' ...
```

```
i deploying
```

```
i functions: ensuring necessary APIs are enabled...
```

```
+ functions: all necessary APIs are enabled
```

```
i functions: preparing directory for uploading...
```

```
i functions: packaged <40.97 KB> for uploading
```

```
+ functions: folder uploaded successfully
```

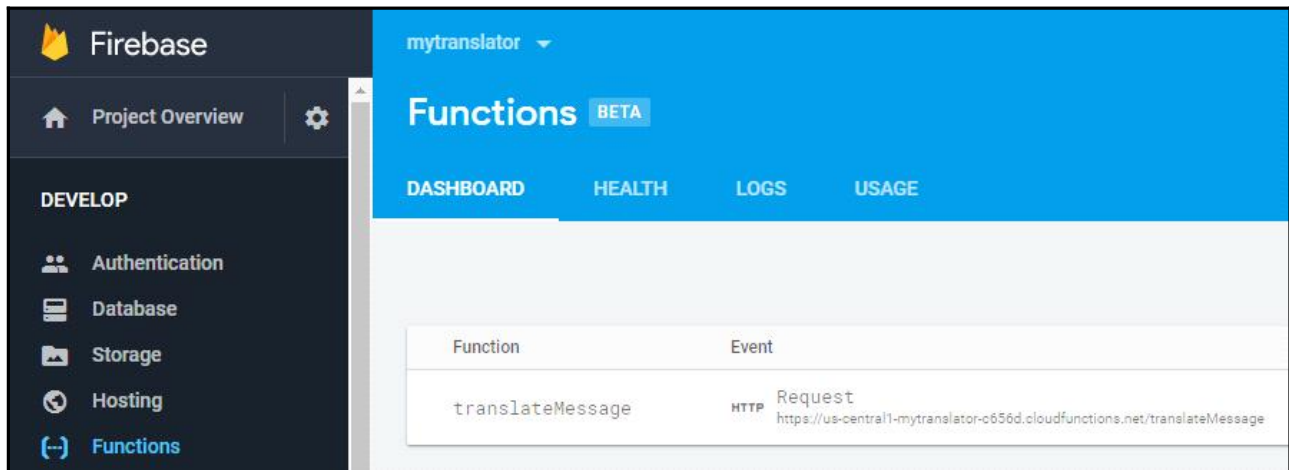
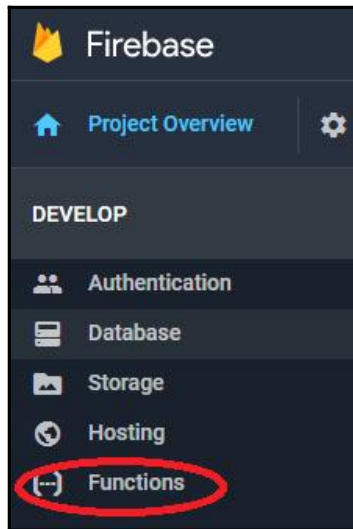
```
i functions: updating function ...
```

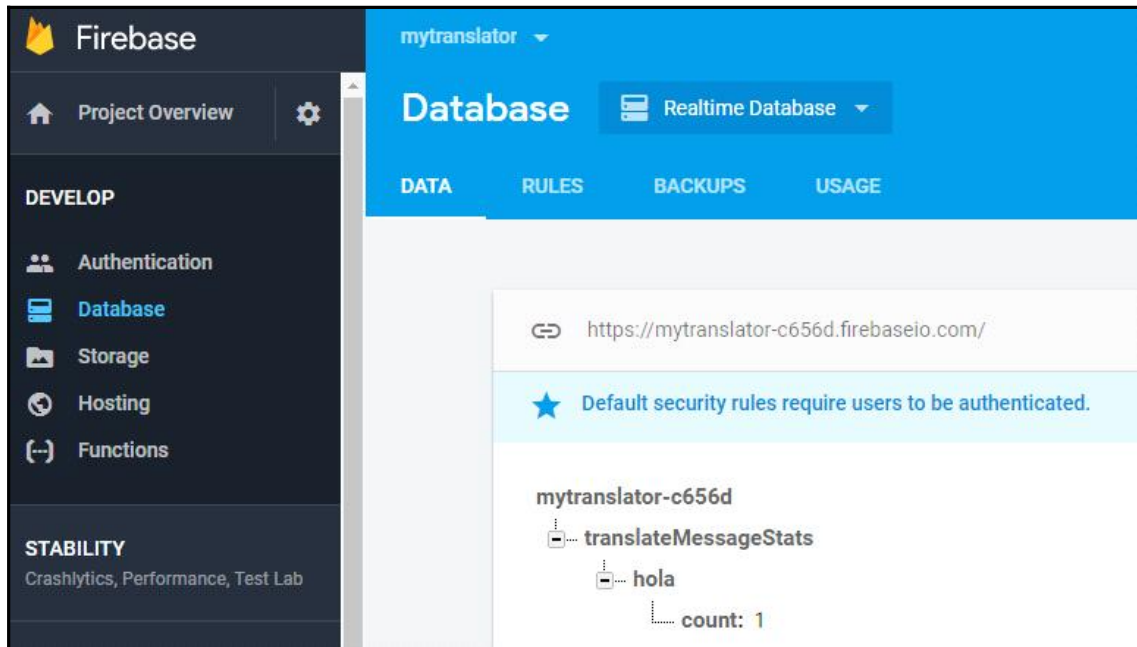
```
+ functions[translateMessage]: Successful update operation.
```

```
+
```

```
https://console.firebase.google.com/project/mytranslator-c656d/
```

```
overview
```



```

E:\firebase>firebase deploy

=== Deploying to 'mytranslator-c656d' ...

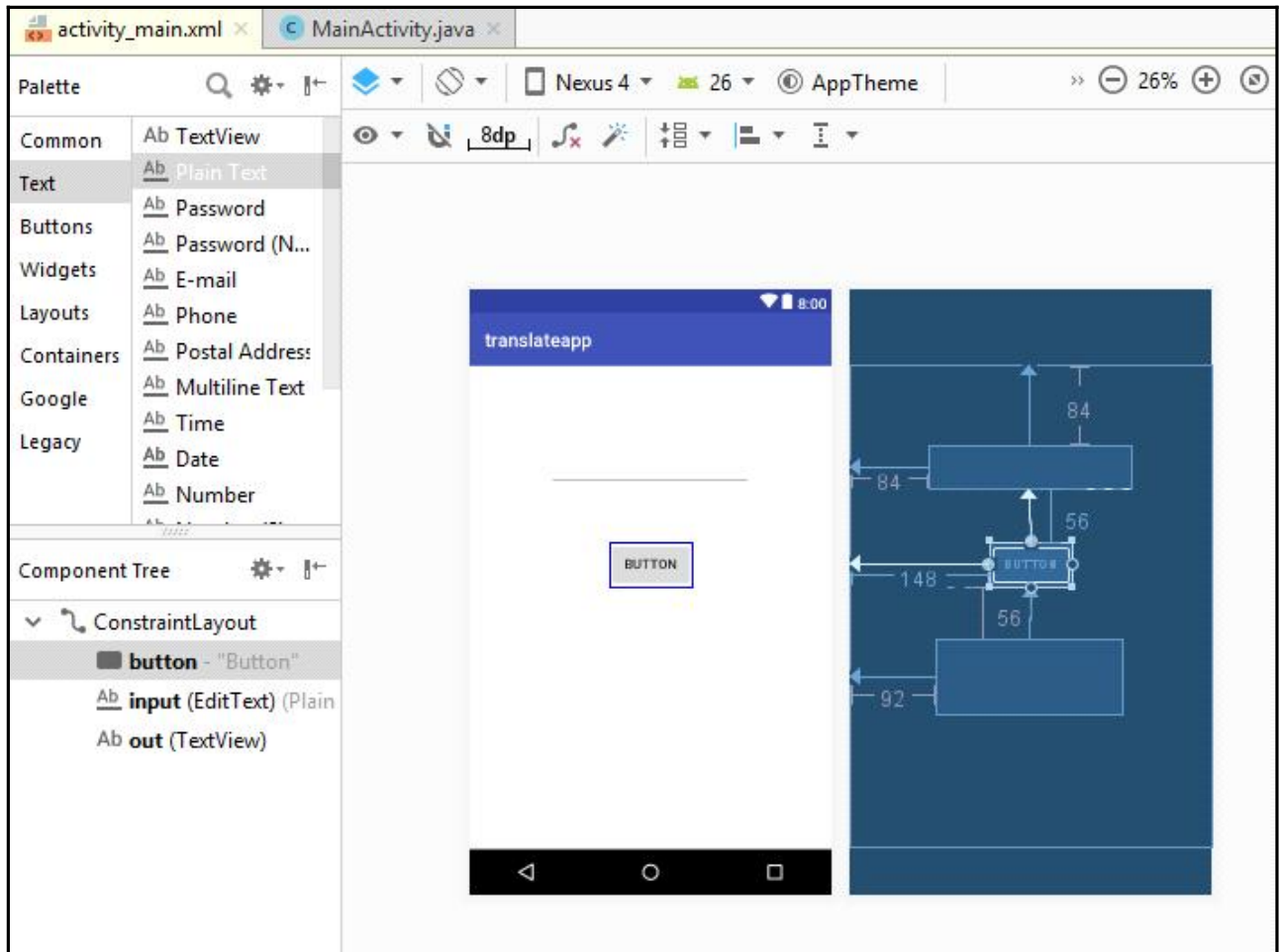
i deploying
i functions: ensuring necessary APIs are enabled...
+ functions: all necessary APIs are enabled
i functions: preparing          directory for uploading...
i functions: packaged          <40.94 KB> for uploading
+ functions: folder uploaded successfully
i hosting: preparing          directory for upload...
+ hosting: 1 files uploaded successfully
i functions: updating function          ...
+ functions[translateMessage]: Successful update operation.

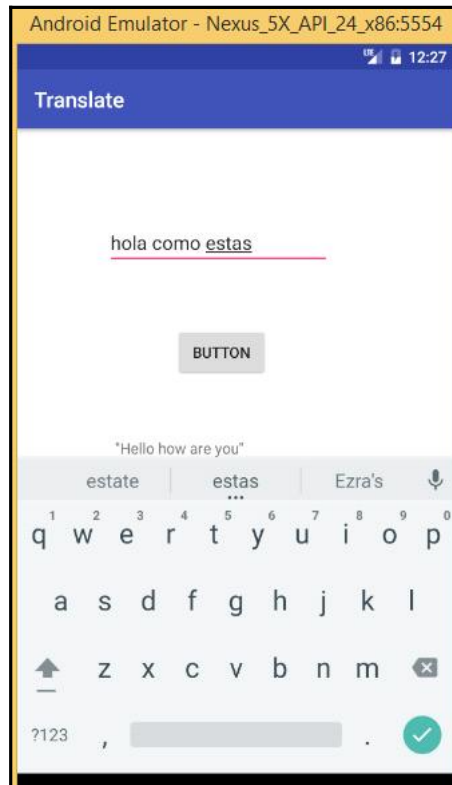
+

https://console.firebase.google.com/project/mytranslator-c656d/
overview https://mytranslator-c656d.firebaseio.com

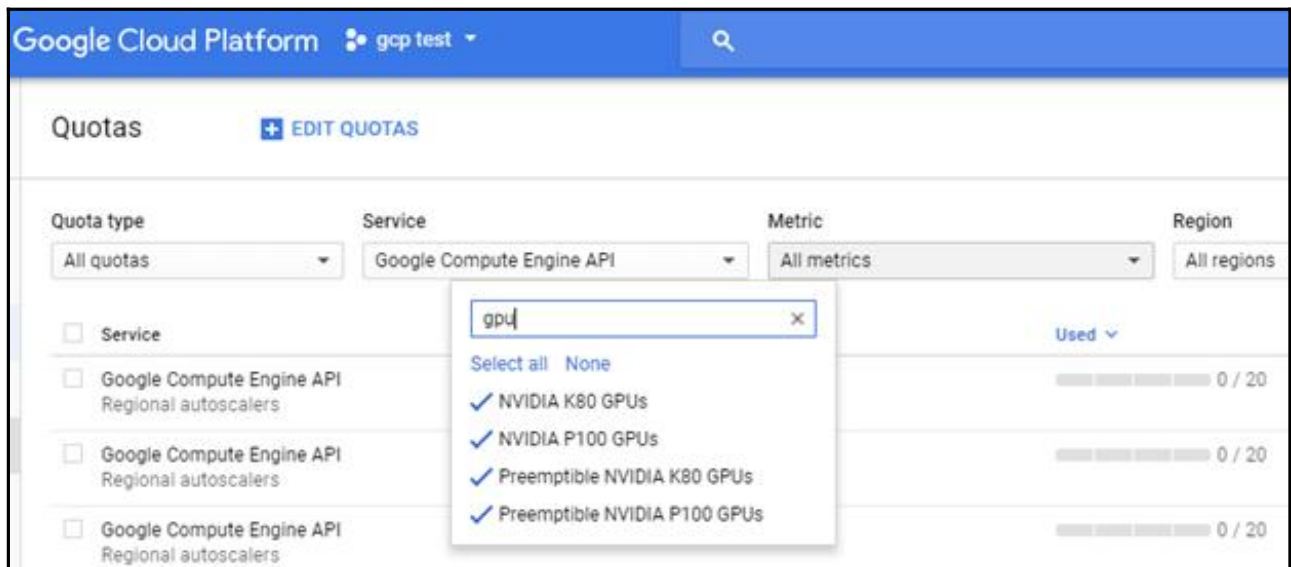
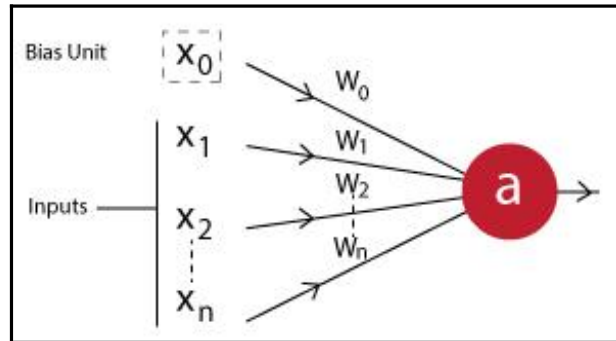
```

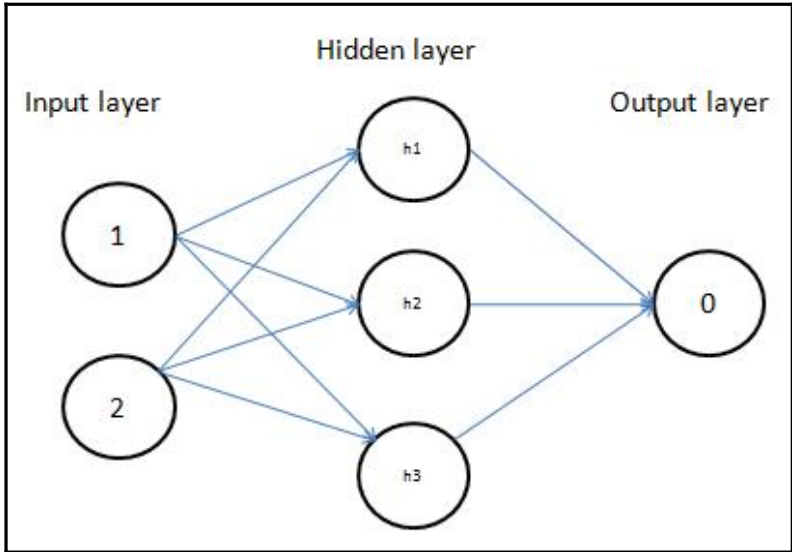
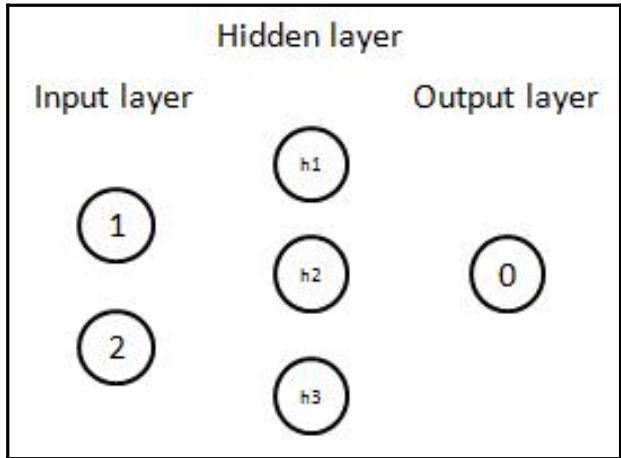
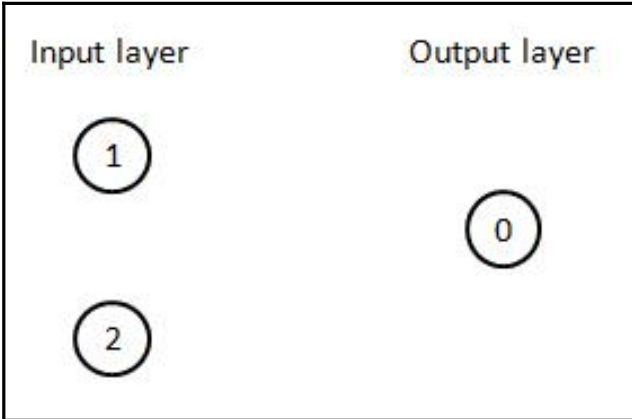


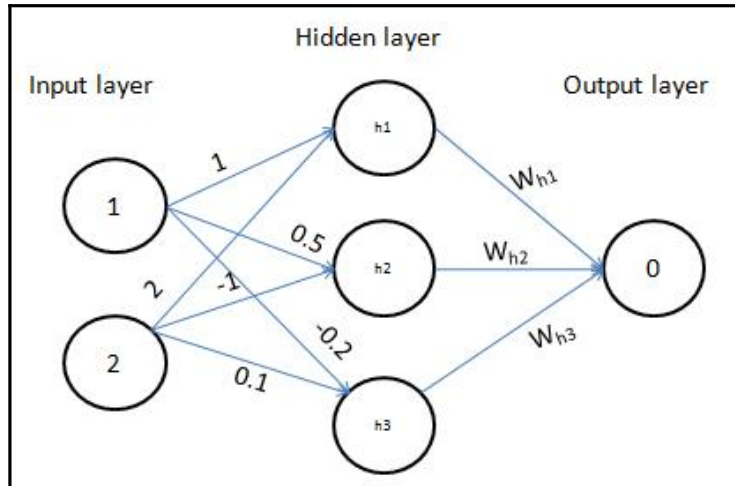
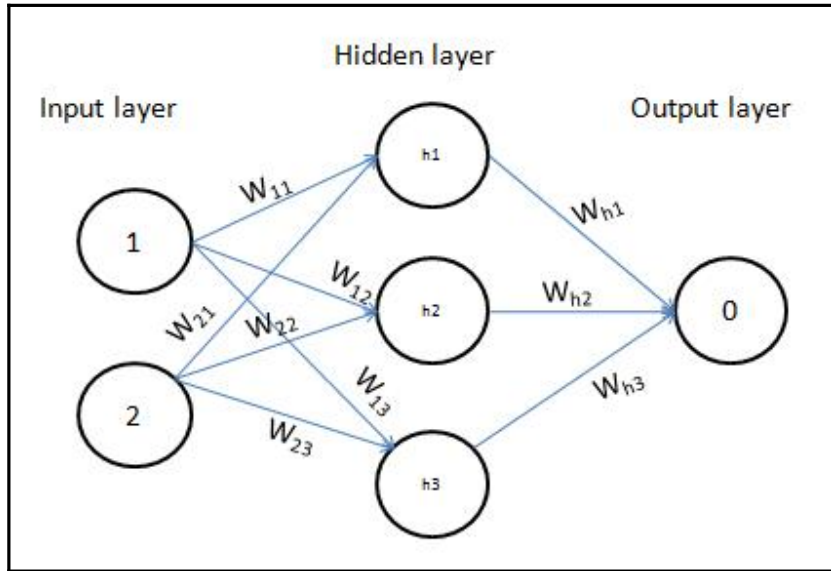


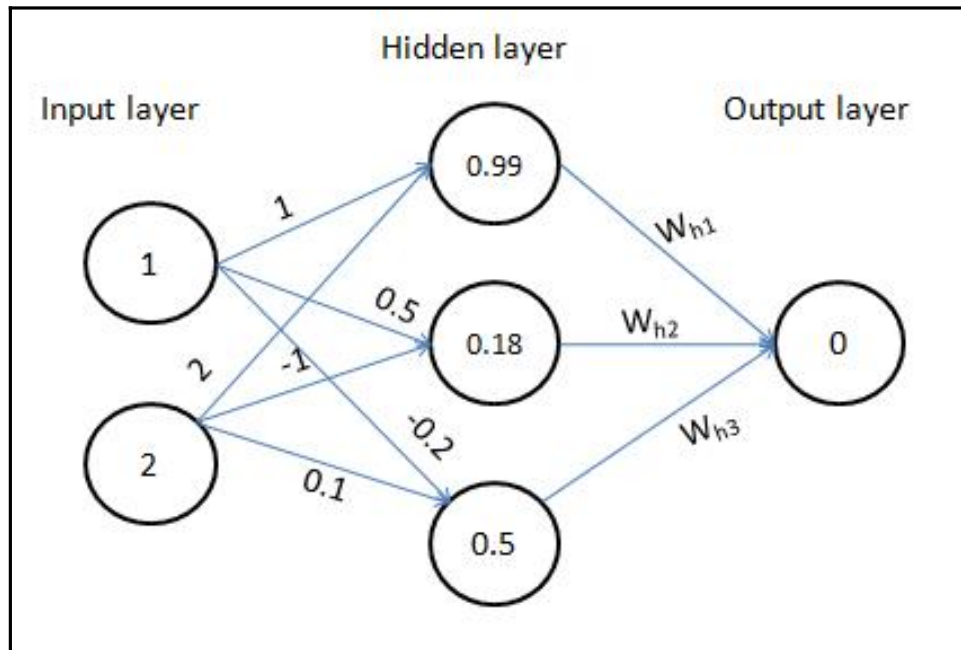
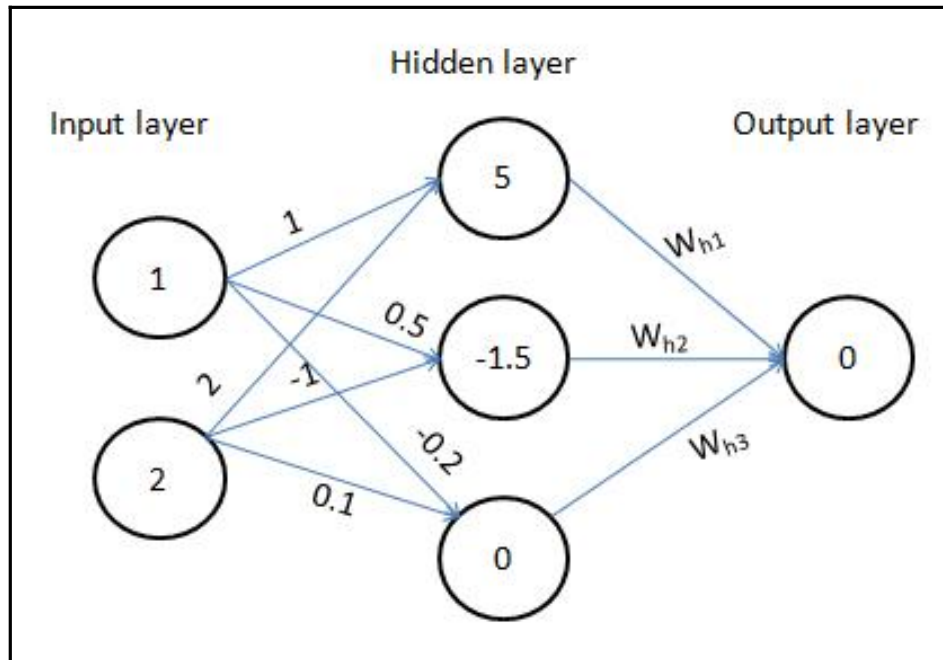


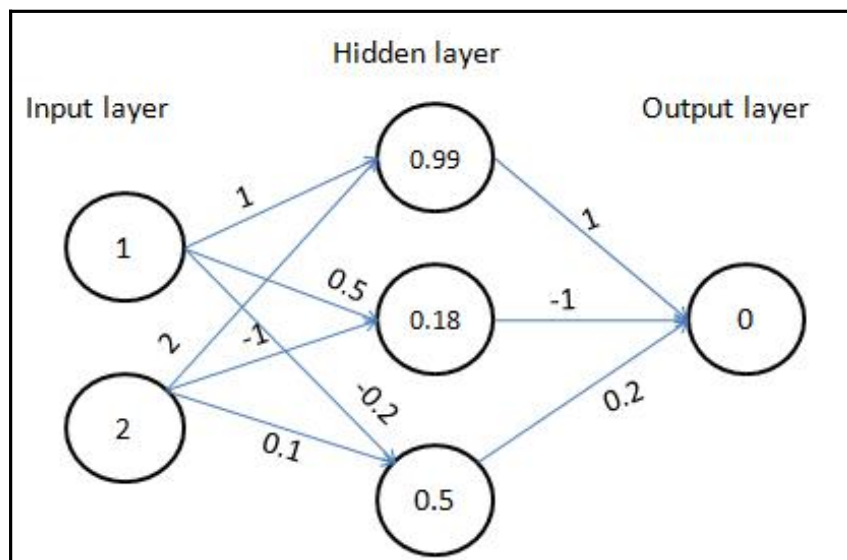
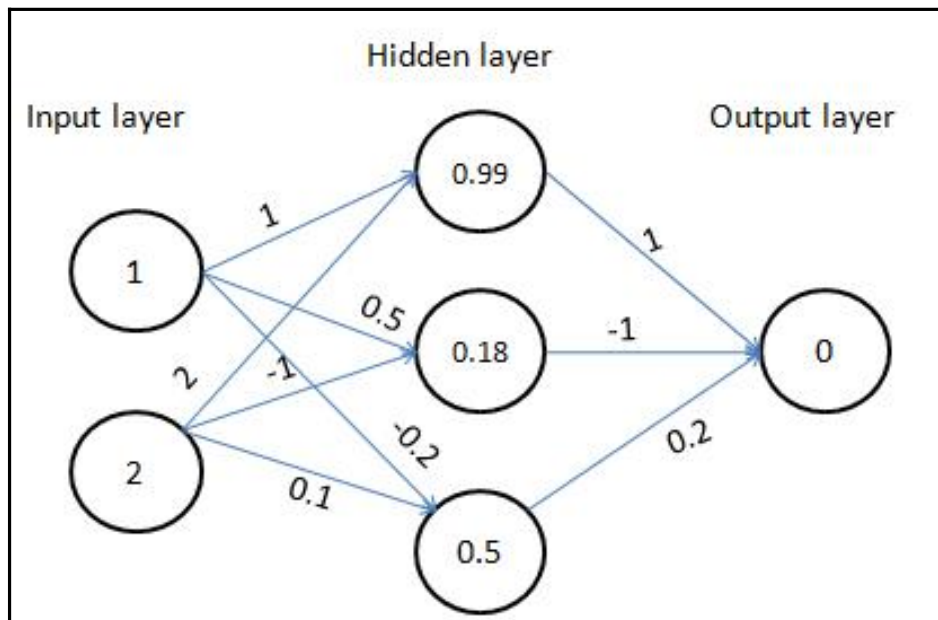
Chapter 9: Neural Networks with TensorFlow and Keras











```
from keras.models import Sequential
from keras.layers import Dense
```

```
model.summary()
```

```
-----  
Layer (type)                Output Shape                Param #  
-----  
dense_1 (Dense)             (None, 3)                   9  
-----  
dense_2 (Dense)             (None, 1)                   4  
-----  
Total params: 13  
Trainable params: 13  
Non-trainable params: 0  
-----
```

```
model.compile(loss='mean_squared_error', optimizer='sgd')
```

```
model.fit(x,y,epochs=10)
```

```
Epoch 1/10  
2/2 [=====] - 0s 2ms/step - loss: 0.9903  
Epoch 2/10  
2/2 [=====] - 0s 741us/step - loss: 0.9413  
Epoch 3/10  
2/2 [=====] - 0s 711us/step - loss: 0.8959  
Epoch 4/10  
2/2 [=====] - 0s 686us/step - loss: 0.8539  
Epoch 5/10  
2/2 [=====] - 0s 729us/step - loss: 0.8151  
Epoch 6/10  
2/2 [=====] - 0s 624us/step - loss: 0.7792  
Epoch 7/10  
2/2 [=====] - 0s 657us/step - loss: 0.7460  
Epoch 8/10  
2/2 [=====] - 0s 676us/step - loss: 0.7153  
Epoch 9/10  
2/2 [=====] - 0s 606us/step - loss: 0.6869  
Epoch 10/10  
2/2 [=====] - 0s 612us/step - loss: 0.6606
```

```
model.get_weights()
```

```
[array([[ -0.9857289 ,  0.5377088 , -0.2417026 ],
        [ -0.3587317 , -0.82248205,  0.98158664]], dtype=float32),
 array([ 0.00195088,  0.01157819, -0.00272675], dtype=float32),
 array([[ 0.8347641 ],
        [ 0.6618012 ],
        [-0.40098628]], dtype=float32),
 array([0.1461481], dtype=float32)]
```

```
new_input = np.array([[2,5]])
model.predict(new_input)

array([[ -0.2005076]], dtype=float32)
```

```
model = Sequential()
model.add(Dense(3,input_dim=2,activation='sigmoid'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='sgd')
model.fit(x,y,epochs=10)

Epoch 1/10
2/2 [=====] - 0s 56ms/step - loss: 0.6903
Epoch 2/10
2/2 [=====] - 0s 636us/step - loss: 0.6560
Epoch 3/10
2/2 [=====] - 0s 528us/step - loss: 0.6242
Epoch 4/10
2/2 [=====] - 0s 473us/step - loss: 0.5948
Epoch 5/10
2/2 [=====] - 0s 478us/step - loss: 0.5674
Epoch 6/10
2/2 [=====] - 0s 478us/step - loss: 0.5421
Epoch 7/10
2/2 [=====] - 0s 443us/step - loss: 0.5186
Epoch 8/10
2/2 [=====] - 0s 430us/step - loss: 0.4969
Epoch 9/10
2/2 [=====] - 0s 440us/step - loss: 0.4768
Epoch 10/10
2/2 [=====] - 0s 438us/step - loss: 0.4581

<keras.callbacks.History at 0x7f7d091da208>
```

```
from numpy.random import seed
seed(1)
from tensorflow import set_random_seed
set_random_seed(2)

model = Sequential()
model.add(Dense(3,input_dim=2,activation='sigmoid'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='sgd')
model.fit(x,y,epochs=10)
```

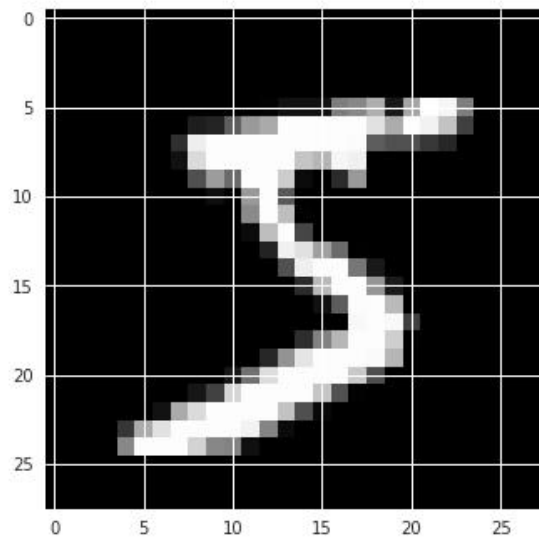
```
from keras.datasets import mnist
```

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

```
import matplotlib.pyplot as plt  
plt.imshow(X_train[0], cmap=plt.get_cmap('gray'))
```

```
<matplotlib.image.AxesImage at 0x7f2190cd4630>
```

```
/usr/local/envs/py3env/lib/python3.5/site-packages/  
not found. Falling back to DejaVu Sans  
(prop.get_family(), self.defaultFamily[fonttext]))
```



```
print(X_train.shape)  
print(y_train.shape)  
print(X_test.shape)  
print(y_test.shape)
```

```
(60000, 28, 28)  
(60000,)  
(10000, 28, 28)  
(10000,)
```

```
from keras.utils import np_utils
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
```

```
1 model = Sequential()
2 model.add(Dense(1000,input_dim=784,activation='sigmoid'))
3 model.add(Dense(10,activation='softmax'))
4 model.compile(loss='categorical_crossentropy', optimizer='sgd',metrics=['accuracy'])
5 model.fit(X_train,y_train,epochs=10,batch_size=1024,validation_data=(X_test,y_test),verbose=2)
```

Train on 60000 samples, validate on 10000 samples

```
Epoch 1/10
- 1s - loss: 1.3256 - acc: 0.6351 - val_loss: 0.8083 - val_acc: 0.8165
Epoch 2/10
- 1s - loss: 0.6900 - acc: 0.8394 - val_loss: 0.5774 - val_acc: 0.8670
Epoch 3/10
- 1s - loss: 0.5355 - acc: 0.8728 - val_loss: 0.4837 - val_acc: 0.8817
Epoch 4/10
- 1s - loss: 0.4600 - acc: 0.8888 - val_loss: 0.4320 - val_acc: 0.8909
Epoch 5/10
- 1s - loss: 0.4134 - acc: 0.8989 - val_loss: 0.3982 - val_acc: 0.8984
Epoch 6/10
- 1s - loss: 0.3803 - acc: 0.9068 - val_loss: 0.3745 - val_acc: 0.9004
Epoch 7/10
- 1s - loss: 0.3552 - acc: 0.9122 - val_loss: 0.3562 - val_acc: 0.9054
Epoch 8/10
- 1s - loss: 0.3352 - acc: 0.9170 - val_loss: 0.3417 - val_acc: 0.9094
Epoch 9/10
- 1s - loss: 0.3185 - acc: 0.9209 - val_loss: 0.3301 - val_acc: 0.9112
Epoch 10/10
- 1s - loss: 0.3044 - acc: 0.9244 - val_loss: 0.3194 - val_acc: 0.9144
```

```
1 model = Sequential()
2 model.add(Dense(1000,input_dim=784,activation='relu'))
3 model.add(Dense(10,activation='softmax'))
4 model.compile(loss='categorical_crossentropy', optimizer='sgd',metrics=['accuracy'])
5 model.fit(X_train,y_train,epochs=10,batch_size=1024,validation_data=(X_test,y_test),verbose=2)

Train on 60000 samples, validate on 10000 samples
Epoch 1/10
- 1s - loss: 11.6053 - acc: 0.2778 - val_loss: 11.3874 - val_acc: 0.2932
Epoch 2/10
- 1s - loss: 11.3660 - acc: 0.2941 - val_loss: 11.3854 - val_acc: 0.2930
Epoch 3/10
- 1s - loss: 11.3262 - acc: 0.2968 - val_loss: 11.3722 - val_acc: 0.2934
Epoch 4/10
- 1s - loss: 11.3217 - acc: 0.2972 - val_loss: 11.3691 - val_acc: 0.2939
Epoch 5/10
- 1s - loss: 11.3165 - acc: 0.2975 - val_loss: 11.3485 - val_acc: 0.2953
Epoch 6/10
- 1s - loss: 11.3046 - acc: 0.2983 - val_loss: 11.3528 - val_acc: 0.2946
Epoch 7/10
- 1s - loss: 11.2998 - acc: 0.2987 - val_loss: 11.3139 - val_acc: 0.2977
Epoch 8/10
- 1s - loss: 11.2944 - acc: 0.2988 - val_loss: 11.3160 - val_acc: 0.2973
Epoch 9/10
- 1s - loss: 11.2871 - acc: 0.2994 - val_loss: 11.3224 - val_acc: 0.2973
Epoch 10/10
- 1s - loss: 11.2879 - acc: 0.2994 - val_loss: 11.3152 - val_acc: 0.2975
```

```
X_train=X_train/255
X_test=X_test/255
```

```

1 model = Sequential()
2 model.add(Dense(1000,input_dim=784,activation='relu'))
3 model.add(Dense(10,activation='softmax'))
4 model.compile(loss='categorical_crossentropy', optimizer='sgd',metrics=['accuracy'])
5 model.fit(X_train,y_train,epochs=10,batch_size=1024,validation_data=(X_test,y_test),verbose=2)

```

```

Train on 60000 samples, validate on 10000 samples
Epoch 1/10
- 1s - loss: 2.0114 - acc: 0.4501 - val_loss: 1.7018 - val_acc: 0.6994
Epoch 2/10
- 1s - loss: 1.5020 - acc: 0.7495 - val_loss: 1.2889 - val_acc: 0.7926
Epoch 3/10
- 1s - loss: 1.1731 - acc: 0.8024 - val_loss: 1.0284 - val_acc: 0.8251
Epoch 4/10
- 1s - loss: 0.9652 - acc: 0.8251 - val_loss: 0.8637 - val_acc: 0.8429
Epoch 5/10
- 1s - loss: 0.8302 - acc: 0.8391 - val_loss: 0.7540 - val_acc: 0.8536
Epoch 6/10
- 1s - loss: 0.7379 - acc: 0.8489 - val_loss: 0.6773 - val_acc: 0.8621
Epoch 7/10
- 1s - loss: 0.6716 - acc: 0.8562 - val_loss: 0.6210 - val_acc: 0.8677
Epoch 8/10
- 1s - loss: 0.6218 - acc: 0.8623 - val_loss: 0.5780 - val_acc: 0.8735
Epoch 9/10
- 1s - loss: 0.5830 - acc: 0.8673 - val_loss: 0.5441 - val_acc: 0.8763
Epoch 10/10
- 1s - loss: 0.5521 - acc: 0.8711 - val_loss: 0.5167 - val_acc: 0.8811

```

```

1 model = Sequential()
2 model.add(Dense(2000,input_dim=784,activation='relu'))
3 model.add(Dense(10,activation='softmax'))
4 model.compile(loss='categorical_crossentropy', optimizer='sgd',metrics=['accuracy'])
5 model.fit(X_train,y_train,epochs=300,batch_size=1024,validation_data=(X_test,y_test),verbose=2)

```

```

1 model = Sequential()
2 model.add(Dense(2000,input_dim=784,activation='relu'))
3 model.add(Dense(500,activation='relu'))
4 model.add(Dense(10,activation='softmax'))
5 model.compile(loss='categorical_crossentropy', optimizer='sgd',metrics=['accuracy'])
6 model.fit(X_train,y_train,epochs=300,batch_size=1024,validation_data=(X_test,y_test),verbose=2)

```

```

1 model = Sequential()
2 model.add(Dense(2000,input_dim=784,activation='relu'))
3 model.add(Dense(500,activation='relu'))
4 model.add(Dense(10,activation='softmax'))
5 model.compile(loss='categorical_crossentropy', optimizer='sgd',metrics=['accuracy'])
6 model.fit(X_train,y_train,epochs=300,batch_size=16,validation_data=(X_test,y_test),verbose=2)

```

```
import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
%matplotlib inline
```

```
from tensorflow.examples.tutorials.mnist import input_data
data = input_data.read_data_sets('data/MNIST/', one_hot=True)
```

```
Extracting data/MNIST/train-images-idx3-ubyte.gz
Extracting data/MNIST/train-labels-idx1-ubyte.gz
Extracting data/MNIST/t10k-images-idx3-ubyte.gz
Extracting data/MNIST/t10k-labels-idx1-ubyte.gz
```

```
data.train.images.shape
(55000, 784)

data.train.labels.shape
(55000, 10)
```

```
data.train.cls = np.argmax(data.train.labels, axis=1)
data.test.cls = np.argmax(data.test.labels, axis=1)
```

```
data.train.cls[0]
7

data.train.labels[0]
array([0., 0., 0., 0., 0., 0., 0., 1., 0., 0.]
```

```
train_input_fn = tf.estimator.inputs.numpy_input_fn(
    x={"x2": np.array(data.train.images)},
    y=np.array(data.train.cls),
    num_epochs=None,
    batch_size=1024,
    shuffle=True)
```

```
train_input_fn()
```

```
{{'x2': <tf.Tensor 'random_shuffle_queue_DequeueMany:1' shape=(1024, 784) dtype=float32>},  
<tf.Tensor 'random_shuffle_queue_DequeueMany:2' shape=(1024,) dtype=int64>}
```

```
test_input_fn = tf.estimator.inputs.numpy_input_fn(  
    x={"x2": np.array(data.test.images)},  
    y=np.array(data.test.cls),  
    num_epochs=1,  
    shuffle=False)
```

```
feature_x1= tf.feature_column.numeric_column("x2", shape=(784))
```

```
num_hidden_units = [512, 256, 128]
```

```
model.train(input_fn=train_input_fn, steps=2000)
```

```
INFO:tensorflow:Create CheckpointSaverHook.  
INFO:tensorflow:Saving checkpoints for 1 into /tmp/tmpiIpGrH/model.ckpt.  
INFO:tensorflow:loss = 2377.0044, step = 1  
INFO:tensorflow:global_step/sec: 5.41405  
INFO:tensorflow:loss = 130.9731, step = 101 (18.476 sec)  
INFO:tensorflow:global_step/sec: 5.57075  
INFO:tensorflow:loss = 40.6942, step = 201 (17.950 sec)  
INFO:tensorflow:global_step/sec: 5.78901  
INFO:tensorflow:loss = 24.525383, step = 301 (17.274 sec)  
INFO:tensorflow:global_step/sec: 5.77324  
INFO:tensorflow:loss = 27.47683, step = 401 (17.321 sec)  
INFO:tensorflow:global_step/sec: 5.73862  
INFO:tensorflow:loss = 10.29439, step = 501 (17.426 sec)  
INFO:tensorflow:global_step/sec: 5.60209  
INFO:tensorflow:loss = 8.66, step = 601 (17.852 sec)  
INFO:tensorflow:global_step/sec: 5.49976  
INFO:tensorflow:loss = 4.0309644, step = 701 (18.182 sec)  
INFO:tensorflow:global_step/sec: 5.69763  
INFO:tensorflow:loss = 3.0329823, step = 801 (17.551 sec)
```

```
result = model.evaluate(input_fn=test_input_fn)

INFO:tensorflow:Starting evaluation at 2018-03-17
INFO:tensorflow:Restoring parameters from /tmp/tm
INFO:tensorflow:Finished evaluation at 2018-03-17
INFO:tensorflow:Saving dict for global step 2000:
```

```
result
```

```
{'accuracy': 0.972,
 'average_loss': 0.13137925,
 'global_step': 2000,
 'loss': 16.630285}
```

```
def model_fn(features, labels, mode, params):
    x = features["x2"]
    net = tf.layers.dense(inputs=x, name='h1', units=512, activation=tf.nn.relu)
    net2 = tf.layers.dense(inputs=net, name='h2', units=256, activation=tf.nn.relu)
    net3 = tf.layers.dense(inputs=net2, name='h3', units=128, activation=tf.nn.relu)
    net4 = tf.layers.dense(inputs=net3, name='softmax', units=10, activation=tf.nn.softmax)

    y_pred_cls = tf.argmax(net4, axis=1)
    if mode == tf.estimator.ModeKeys.PREDICT:
        spec = tf.estimator.EstimatorSpec(mode=mode, predictions=y_pred_cls)
    else:
        cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=labels, logits = net4)
        loss = tf.reduce_mean(cross_entropy)
        optimizer = tf.train.AdamOptimizer(learning_rate=params["learning_rate"])
        train_op = optimizer.minimize(loss=loss, global_step=tf.train.get_global_step())
        metrics = {"accuracy": tf.metrics.accuracy(labels, y_pred_cls)}
        spec = tf.estimator.EstimatorSpec(mode=mode, loss=loss, train_op=train_op, eval_metric_ops=metrics)
    return spec
```

```
def model_fn(features, labels, mode, params):
```

```
x = features["x2"]
net = tf.layers.dense(inputs=x, name='h1', units=512, activation=tf.nn.relu)
net2 = tf.layers.dense(inputs=net, name='h2', units=256, activation=tf.nn.relu)
net3 = tf.layers.dense(inputs=net2, name='h3', units=128, activation=tf.nn.relu)
net4 = tf.layers.dense(inputs=net3, name='softmax', units=10, activation=tf.nn.softmax)
```

```
y_pred_cls = tf.argmax(net4, axis=1)
```

```
if mode == tf.estimator.ModeKeys.PREDICT:
    spec = tf.estimator.EstimatorSpec(mode=mode, predictions=y_pred_cls)
```

```
else:
    cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=labels, logits = net4)
    loss = tf.reduce_mean(cross_entropy)
    optimizer = tf.train.AdamOptimizer(learning_rate=params["learning_rate"])
    train_op = optimizer.minimize(loss=loss, global_step=tf.train.get_global_step())
    metrics = {"accuracy": tf.metrics.accuracy(labels, y_pred_cls)}
    spec = tf.estimator.EstimatorSpec(mode=mode, loss=loss, train_op=train_op, eval_metric_ops=metrics)
```

```
params = {"learning_rate": 1e-4}
```

```
model = tf.estimator.Estimator(model_fn=model_fn,
                               params=params)
```

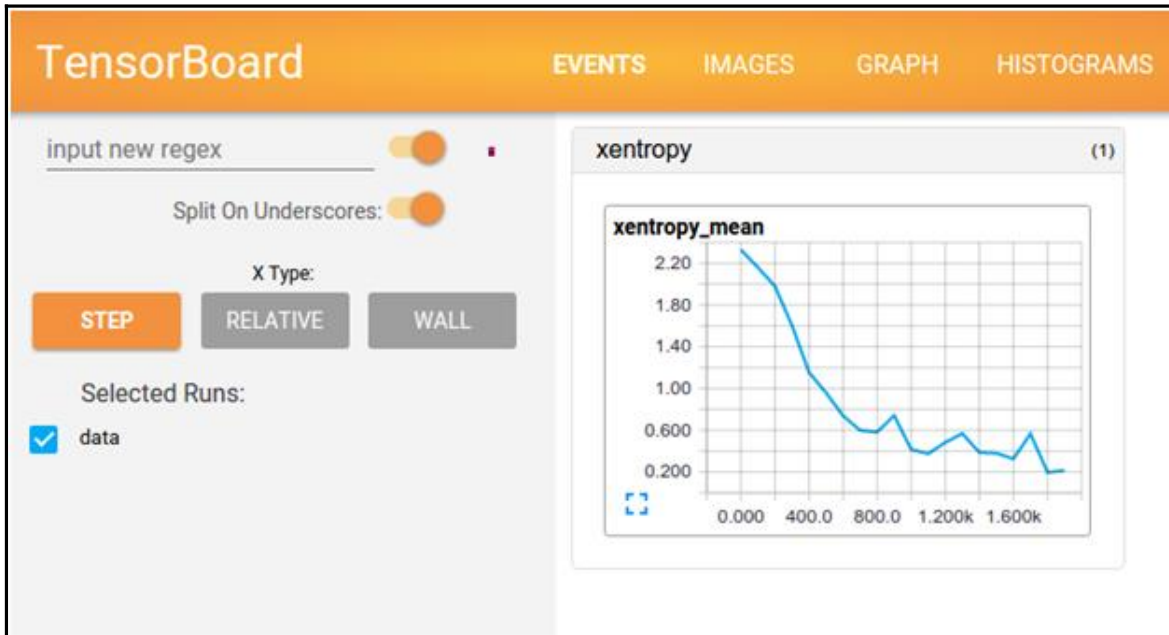
```
result = model.evaluate(input_fn=test_input_fn)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2018-03-17-10:34:50
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from /tmp/tmpthfo2z2/model.ckpt-2000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2018-03-17-10:34:51
INFO:tensorflow:Saving dict for global step 2000: accuracy = 0.9682, global_step = 2000, loss = 1.4945148
```

```
result
```

```
{'accuracy': 0.9682, 'global_step': 2000, 'loss': 1.4945148}
```

Chapter 10: Evaluating Results with TensorBoard



```
# Import relevant packages
from google.datalab.ml import TensorBoard as tb
# Start the tensorboard
tb.start('./logs/1519829425.6576147')
```

TensorBoard was started successfully with pid 4067. Click [here](#) to access it.

4067

```

@staticmethod
def start(logdir):
    """Start a TensorBoard instance.

    Args:
        logdir: the logdir to run TensorBoard on.
    Raises:
        Exception if the instance cannot be started.
    """
    if logdir.startswith('gs://'):
        # Check user does have access. TensorBoard will start successfully regardless
        # the user has read permissions or not so we check permissions here to
        # give user alerts if needed.
        datalab.storage._api.Api.verify_permitted_to_read(logdir)

    port = datalab.utils.pick_unused_port()
    args = ['tensorboard', '--logdir=' + logdir, '--port=' + str(port)]
    p = subprocess.Popen(args)
    retry = 5
    while (retry > 0):
        if datalab.utils.is_http_running_on(port):
            url = '/_proxy/%d/' % port
            html = '<p>TensorBoard was started successfully with pid %d. ' % p.pid
            html += 'Click <a href="%s" target="_blank">here</a> to access it.</p>' % url
            IPython.display.display_html(html, raw=True)
            return p.pid
        time.sleep(1)
        retry -= 1

    raise Exception('Cannot start TensorBoard.')

```

```

from keras.callbacks import TensorBoard
from time import time
model = Sequential()
model.add(Dense(num_pixels, input_dim=num_pixels, kernel_initializer='normal', activation='relu', name='first_layer'))
model.add(Dense(1000, activation='relu', name='hidden_layer'))
model.add(Dense(num_classes, kernel_initializer='normal', activation='softmax', name='output_layer'))
tensorboard = TensorBoard(log_dir="logs/tensor_new6")

# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=5, validation_data=(X_test, y_test), batch_size=1024, verbose=2, callbacks=[tensorboard])

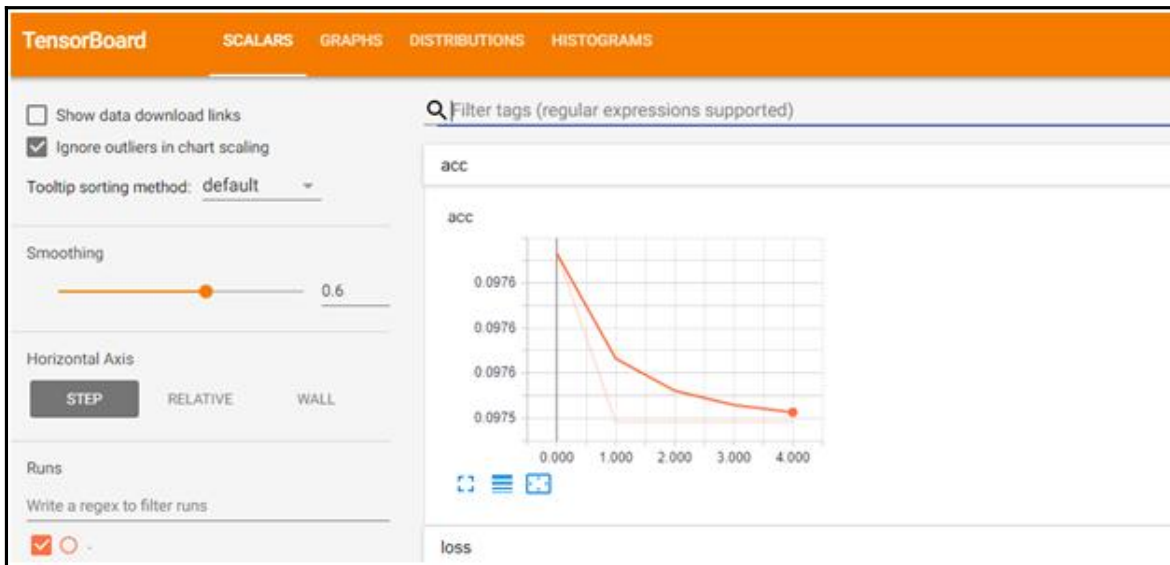
Train on 60000 samples, validate on 10000 samples
Epoch 1/5
- 31s - loss: 13.0498 - acc: 0.1898 - val_loss: 13.0217 - val_acc: 0.1921
Epoch 2/5
- 31s - loss: 13.1082 - acc: 0.1867 - val_loss: 13.1524 - val_acc: 0.1840
Epoch 3/5
- 32s - loss: 13.0979 - acc: 0.1873 - val_loss: 13.0121 - val_acc: 0.1927
Epoch 4/5
- 32s - loss: 13.0369 - acc: 0.1912 - val_loss: 13.0379 - val_acc: 0.1911
Epoch 5/5
- 31s - loss: 13.0398 - acc: 0.1910 - val_loss: 13.0170 - val_acc: 0.1924

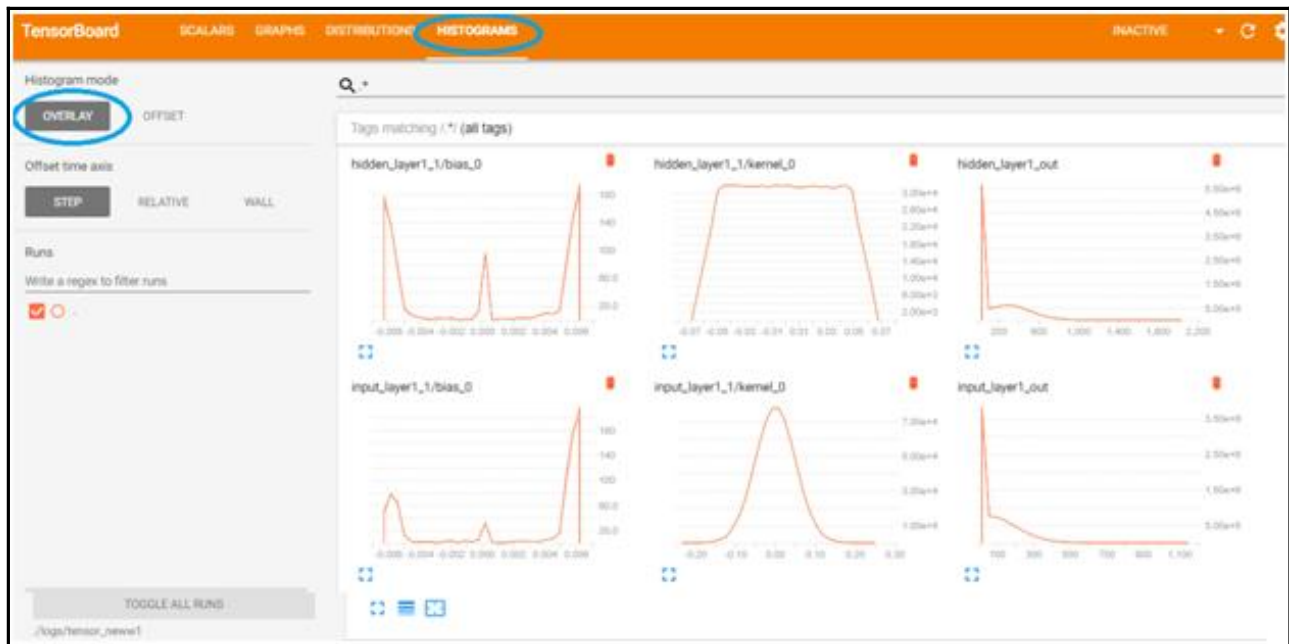
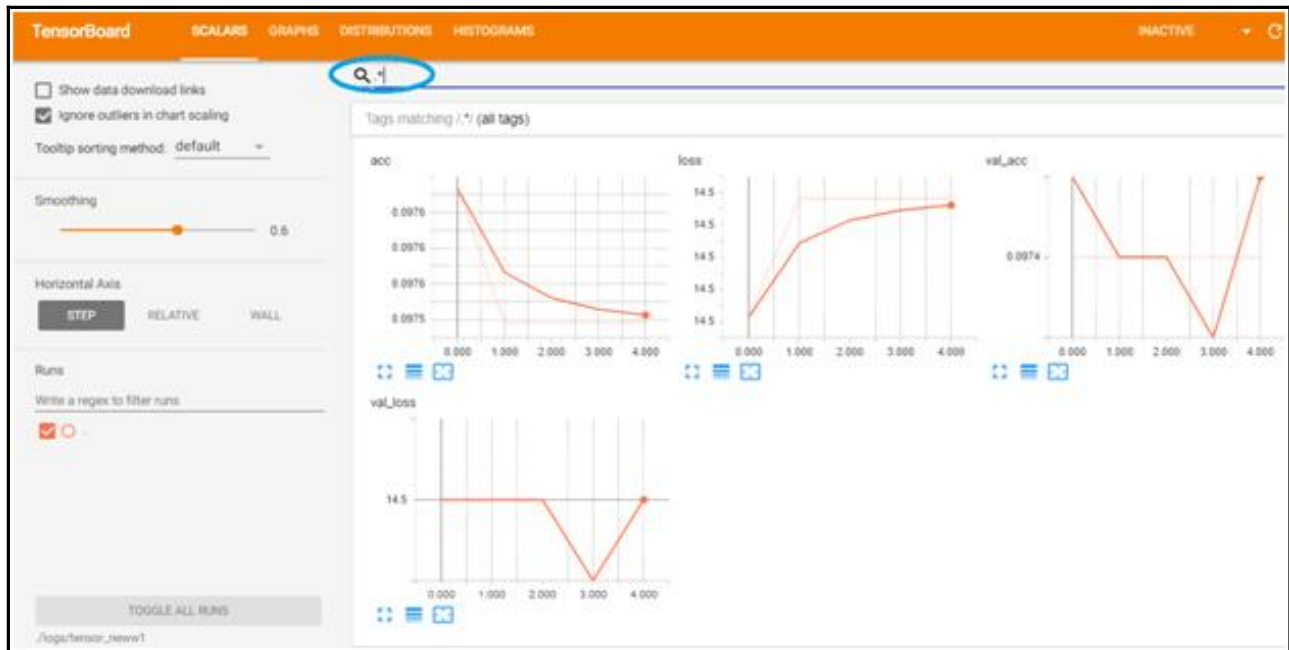
```

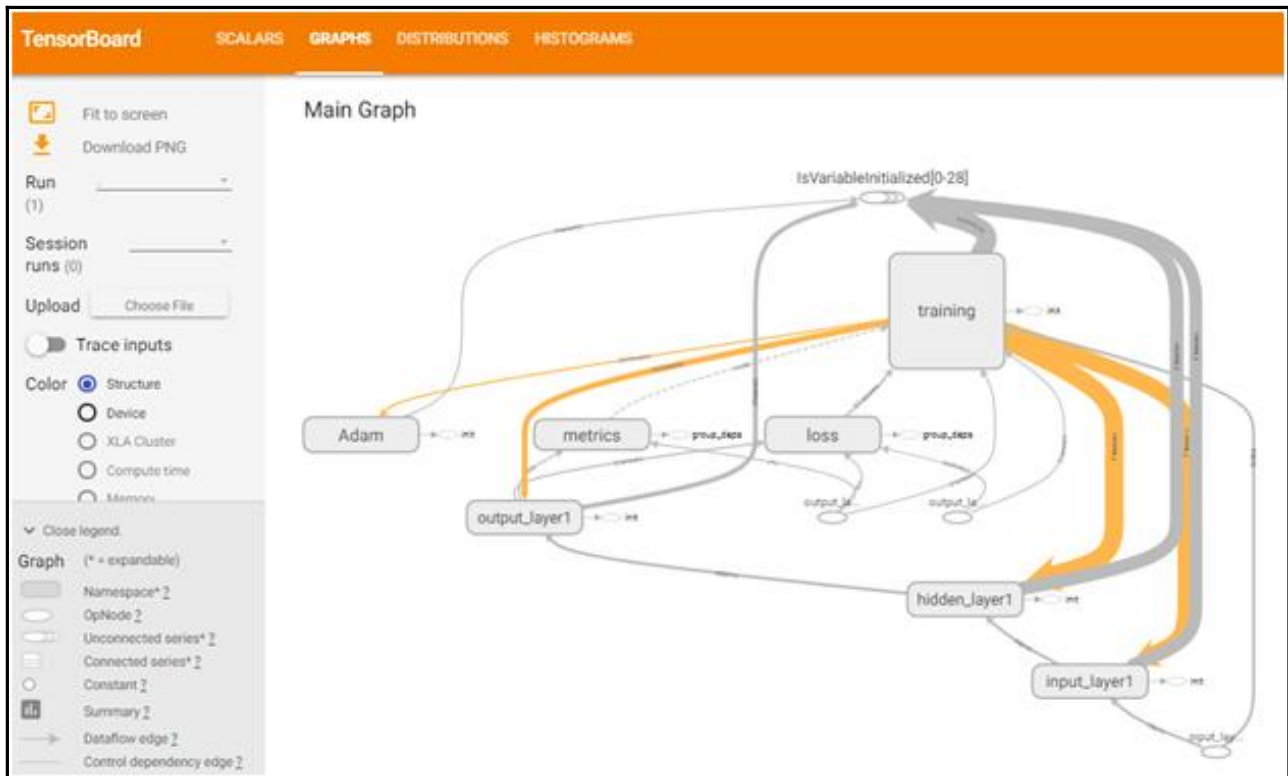
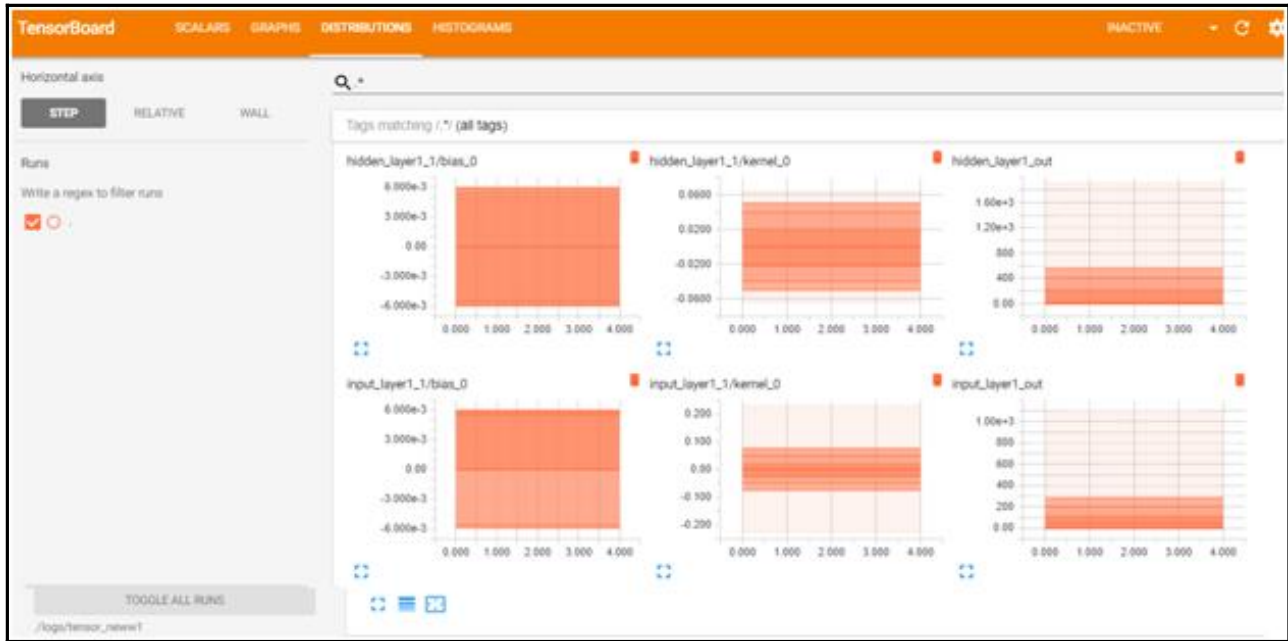
```
# Import relevant packages
from google.datalab.ml import TensorBoard as tb
# Start the tensorboard
tb.start('./logs/tensor_new6')
```

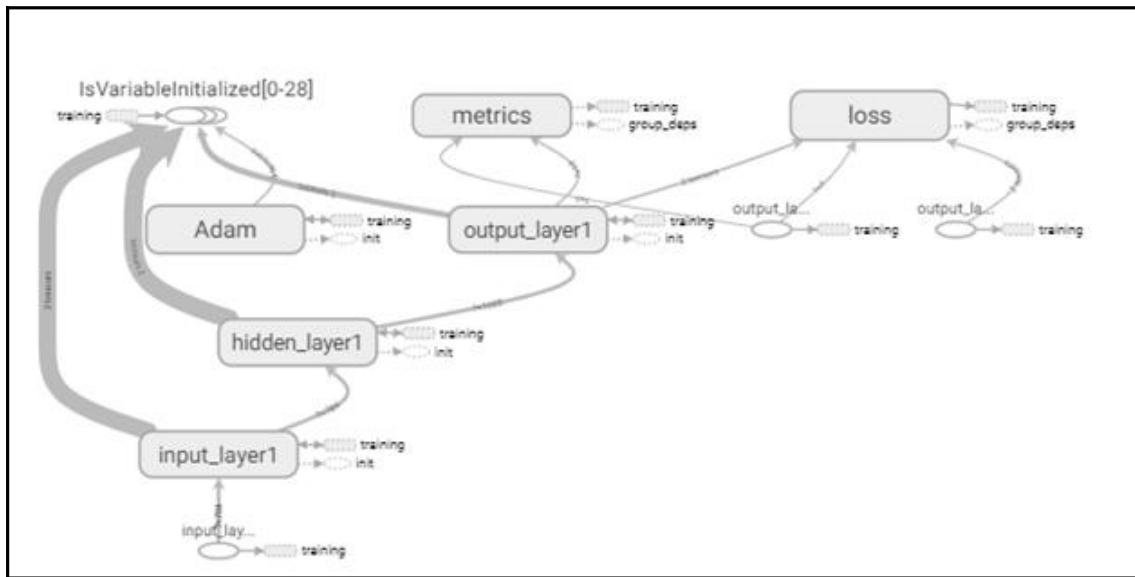
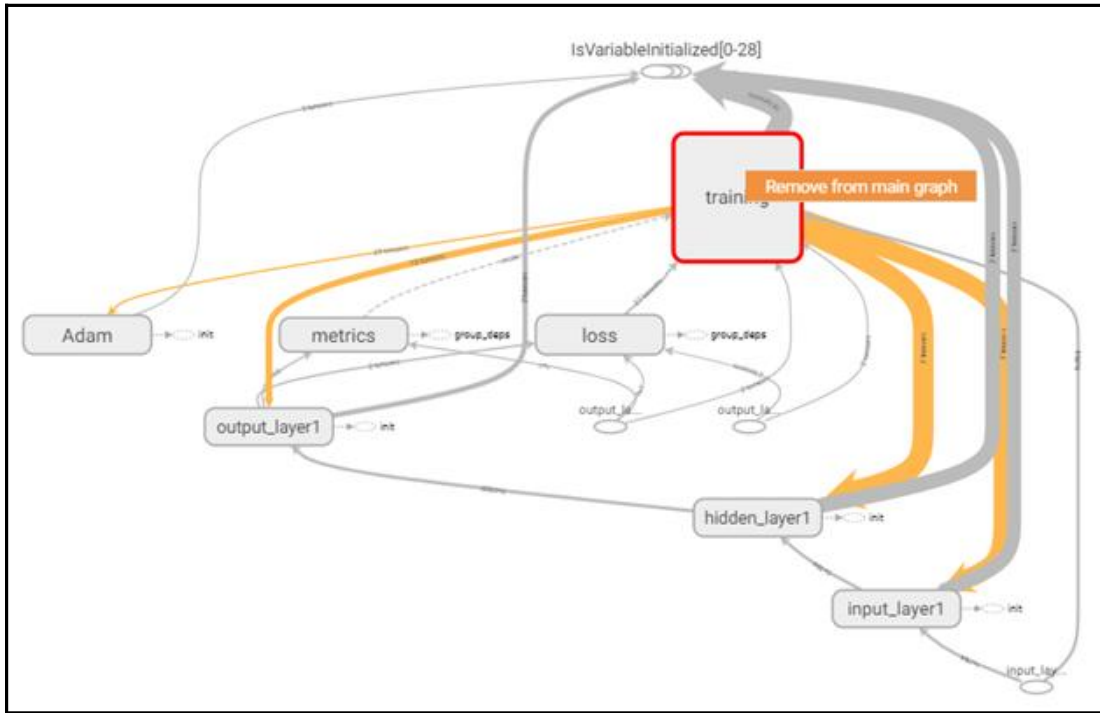
TensorBoard was started successfully with pid 7617. Click [here](#) to access it.

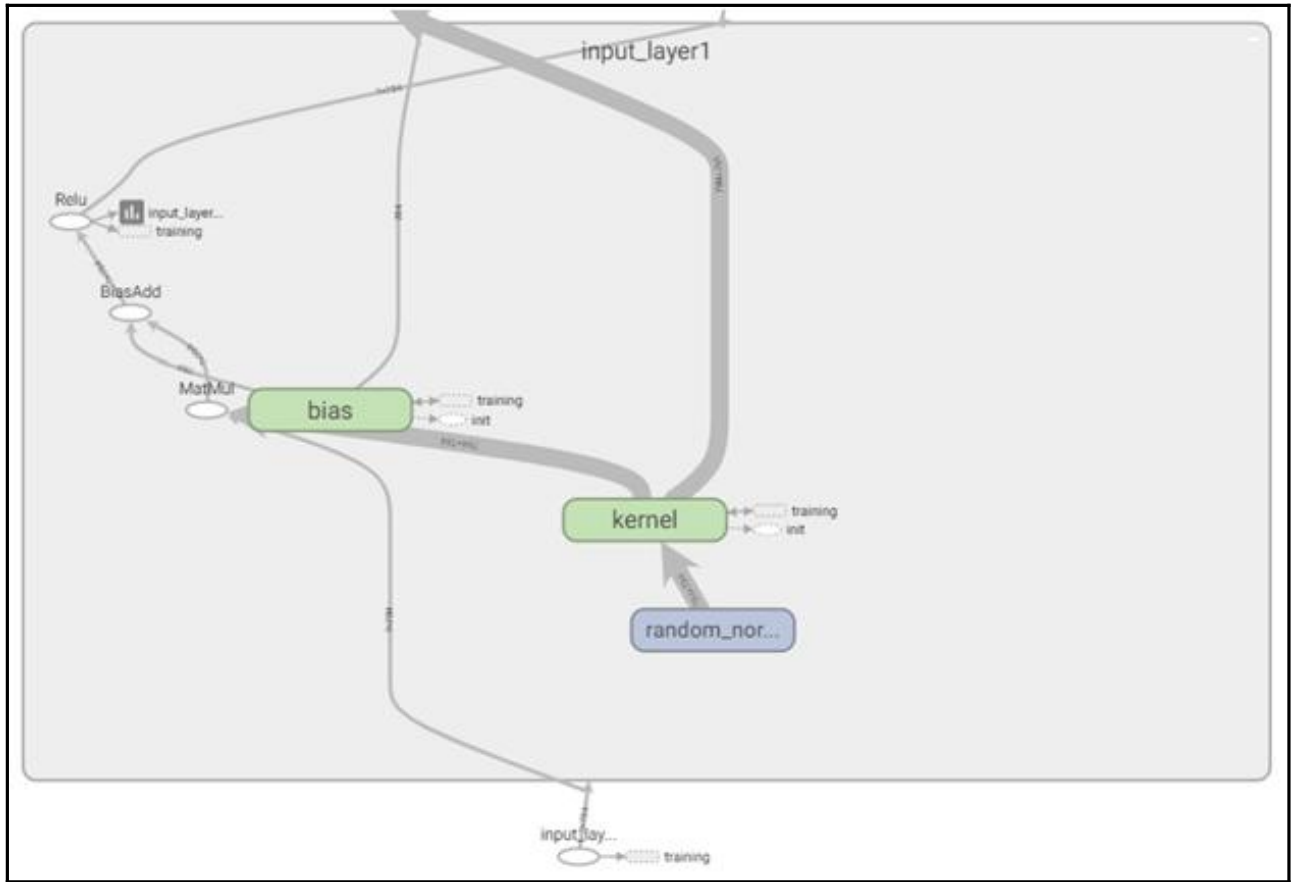
7617

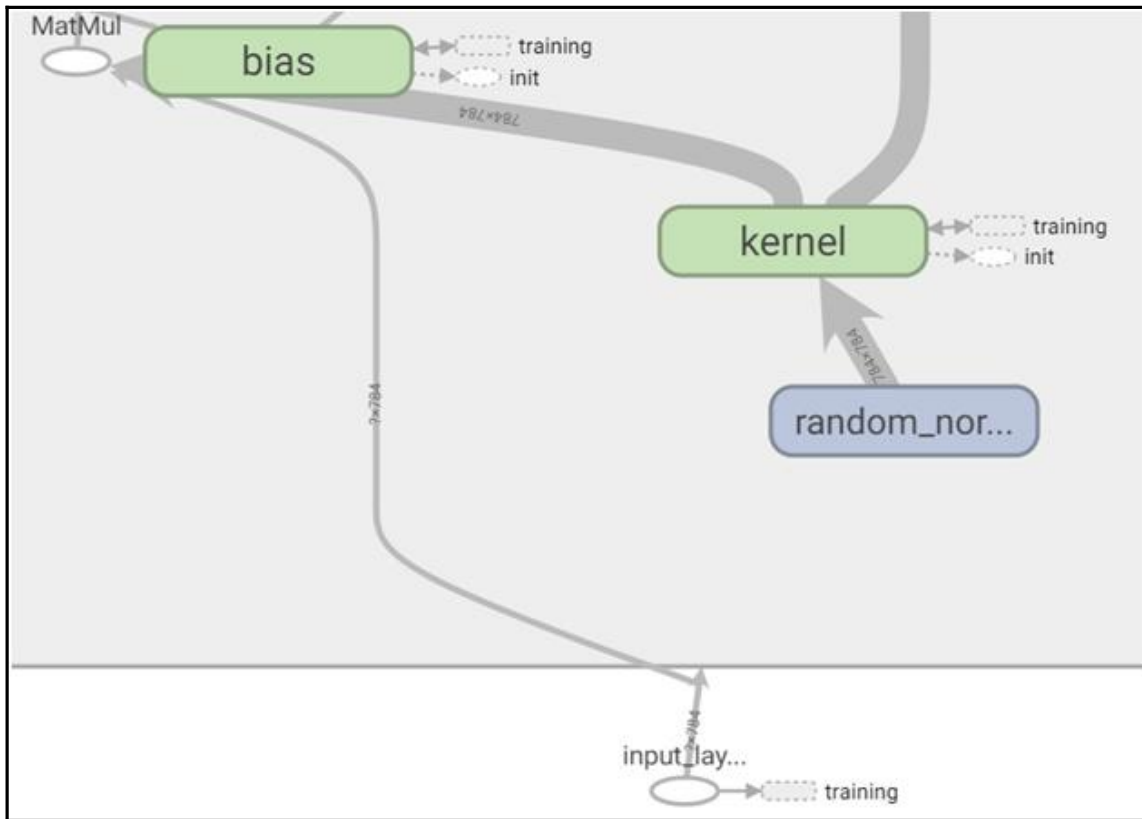












Run (1)

Session runs (0)

Upload

Trace inputs

Color Structure

Device

XLA Cluster

Compute time

Memory

The graph visualization shows a network of nodes. A node labeled 'hidden_layer1' is highlighted with a red border. Below it, a node labeled 'input_layer1' is highlighted with an orange border. Other nodes are highlighted with yellow or hatched patterns. The nodes are connected by lines representing edges in the graph.

```

from keras import backend as K
K.clear_session()

from keras.callbacks import TensorBoard
from time import time
model = Sequential()
model.add(Dense(num_pixels, input_dim=num_pixels, kernel_initializer='normal', activation='sigmoid',name='input_layer1'))
model.add(Dense(1000,activation='relu',name='hidden_layer1'))
model.add(Dense(num_classes, kernel_initializer='normal', activation='softmax',name='output_layer1'))
tensorboard = TensorBoard(log_dir="logs/tensor_neww3",histogram_freq=1,batch_size=10000)

# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```

```

model.fit(X_train, y_train, epochs=5, validation_data=(X_test,y_test) ,batch_size=1024, verbose=2, callbacks=[tensorboard])

```





```
model = tf.estimator.DNNClassifier(feature_columns=feature_columns,  
                                   hidden_units=num_hidden_units,  
                                   activation_fn=tf.nn.relu,  
                                   num_classes=num_classes,  
                                   model_dir='/content/datalab/logs/')
```

```
from google.datalab.ml import TensorBoard as tb
```

```
tb.start('/content/datalab/logs/')
```

TensorBoard was started successfully with pid 1133. Click [here](#) to access it.

```
1133
```

```

def model_fn(features, labels, mode, params):
    x = features["x2"]
    net = tf.layers.dense(inputs=x, name='h1',units=512, activation=tf.nn.relu)
    net2 = tf.layers.dense(inputs=net, name='h2',units=256, activation=tf.nn.relu)
    net3 = tf.layers.dense(inputs=net2, name='h3',units=128, activation=tf.nn.relu)
    net4 = tf.layers.dense(inputs=net3, name='softmax',units=10,activation=tf.nn.softmax)

    y_pred_cls = tf.argmax(net4, axis=1)

    if mode == tf.estimator.ModeKeys.PREDICT:
        spec = tf.estimator.EstimatorSpec(mode=mode,predictions=y_pred_cls)
    else:
        cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=labels,logits = net4)
        loss = tf.reduce_mean(cross_entropy)
        optimizer = tf.train.ProximalAdagradOptimizer(learning_rate=params["learning_rate"],l1_regularization_strength=0.001)
        train_op = optimizer.minimize(loss=loss, global_step=tf.train.get_global_step())
        accuracy = tf.metrics.accuracy(labels, y_pred_cls)
        metrics = {'accuracy': accuracy}
        tf.summary.scalar('train_accuracy', accuracy[1])

        tf.summary.histogram("hidden1",net)
        tf.summary.histogram("hidden2",net2)
        tf.summary.histogram("hidden3",net3)

        spec = tf.estimator.EstimatorSpec(mode=mode,loss=loss,train_op=train_op,eval_metric_ops=metrics)

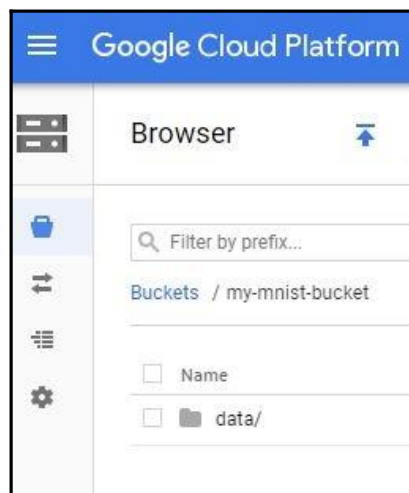
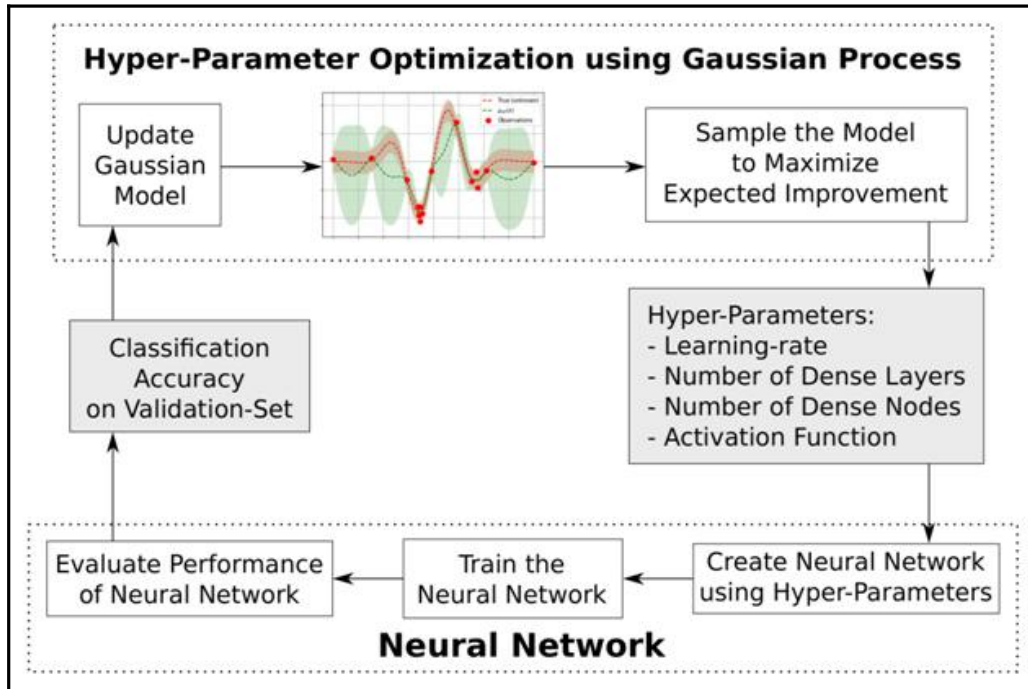
```

```

params = {"learning_rate": 0.1}
model = tf.estimator.Estimator(model_fn=model_fn,
                               params=params,model_dir='/content/datalab/docs/log10/')

```

Chapter 11: Optimizing the Model through Hyperparameter Tuning



```
Training output      {
                    "completedTrialCount": "10",
                    "trials": [
                      {
                        "trialId": "4",
                        "hyperparameters": {
                          "num-steps": "7658",
                          "lr": "0.014984704974612019"
                        },
                        "finalMetric": {
                          "trainingStep": "7658",
                          "objectiveValue": 0.984000027179718
                        }
                      },
                      {
                        "trialId": "9",
                        "hyperparameters": {
                          "lr": "0.012684523913596773",
                          "num-steps": "9994"
                        },
                        "finalMetric": {
                          "trainingStep": "9994",
                          "objectiveValue": 0.9832000136375427
                        }
                      }
                    ],
                    }
                    }
```

Chapter 12: Preventing Overfitting with Regularization

```
import numpy as np

np.random.seed(42)
x1=np.random.random_sample((2000,5))
y=[[1,0]]
y1=np.repeat(y,2000,axis=0)

np.random.seed(0)
x2=np.random.random_sample((2000,5))*-1
y=[[0,1]]
y2=np.repeat(y,2000,axis=0)

x_new=np.append(x1,x2,axis=0)

y_new=np.append(y1,y2,axis=0)
```

```
np.random.seed(5)
x1_badsamples = np.random.random_sample((200,5))
x2_badsamples = np.random.random_sample((200,5))*-1

y1_badsamples = np.repeat([[0,1]],200,axis=0)
y2_badsamples = np.repeat([[1,0]],200,axis=0)

x_badsamples = np.append(x1_badsamples,x2_badsamples,axis=0)
y_badsamples = np.append(y1_badsamples,y2_badsamples,axis=0)
```

```
final_x=np.append(x_new,x_badsamples,axis=0)
final_y=np.append(y_new,y_badsamples,axis=0)
```

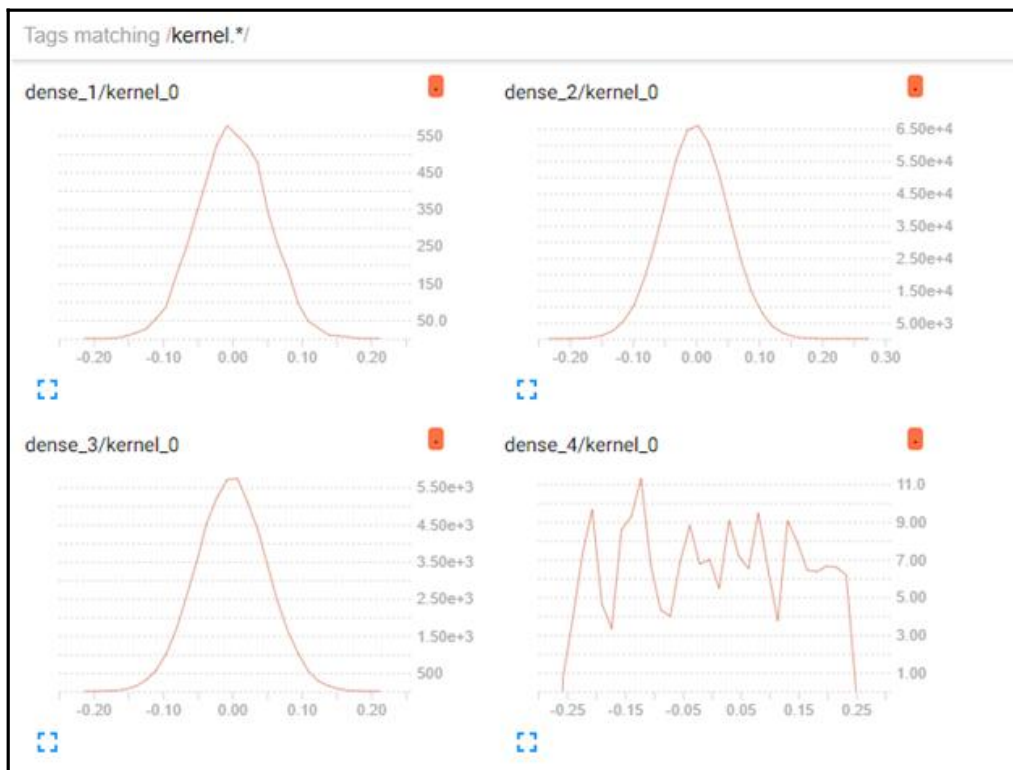
```

np.random.seed(10)
x1_test = np.random.random_sample((200,5))
x2_test = np.random.random_sample((200,5))*-1

y1_test = np.repeat([[1,0]],200,axis=0)
y2_test = np.repeat([[0,1]],200,axis=0)

test_x=np.append(x1_test,x2_test,axis=0)
test_y=np.append(y1_test,y2_test,axis=0)

```



$$\text{Cost} = \underbrace{\sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2}_{\text{Loss function}} + \lambda \underbrace{\sum_{j=0}^M W_j^2}_{\text{Regularization Term}}$$

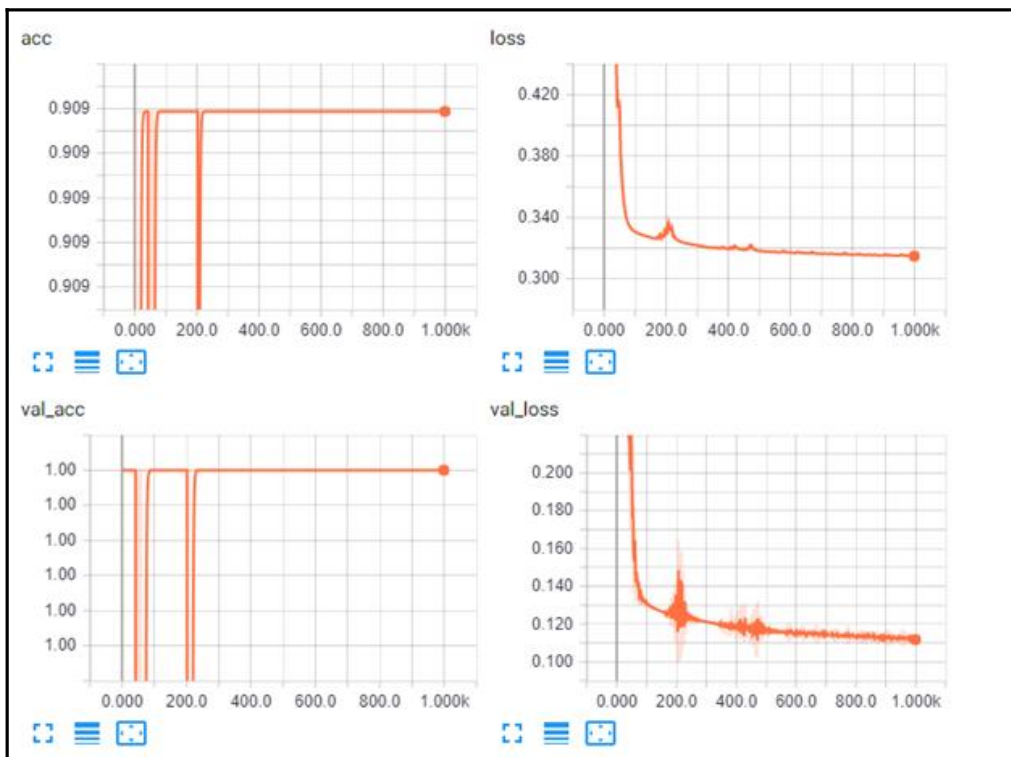
```

from keras.regularizers import l2
from keras import backend as K
K.clear_session()
model = Sequential()
model.add(Dense(1000, input_dim=5, kernel_initializer='normal', kernel_regularizer=l2(0.01), activation='relu'))
model.add(Dense(500, kernel_initializer='normal', kernel_regularizer=l2(0.01), activation='relu'))
model.add(Dense(100, kernel_initializer='normal', kernel_regularizer=l2(0.01), activation='relu'))

model.add(Dense(2, activation='softmax'))
tensorboard = TensorBoard(log_dir = "/content/datalab/logs/tensor_17", histogram_freq=1000, batch_size=5000)

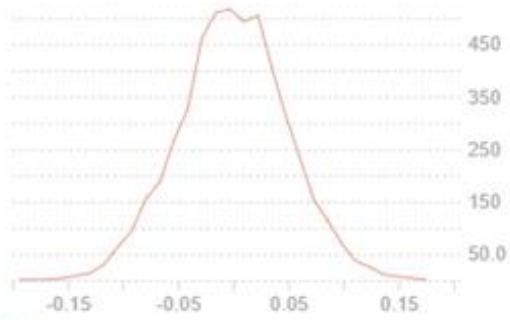
adam = Adam(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])

```

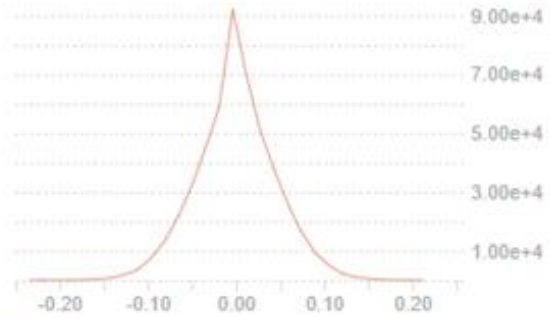


Tags matching /kernel*/

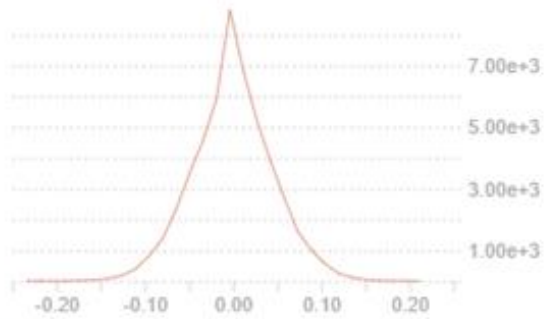
dense_1/kernel_0



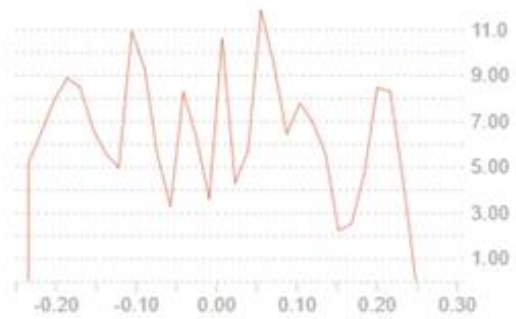
dense_2/kernel_0



dense_3/kernel_0

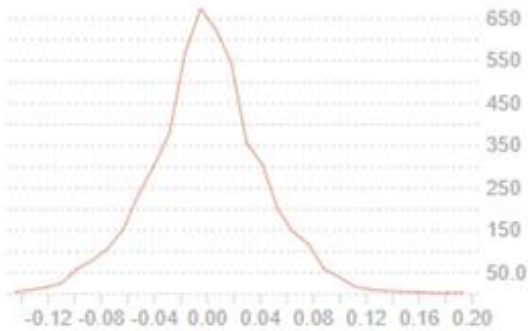


dense_4/kernel_0

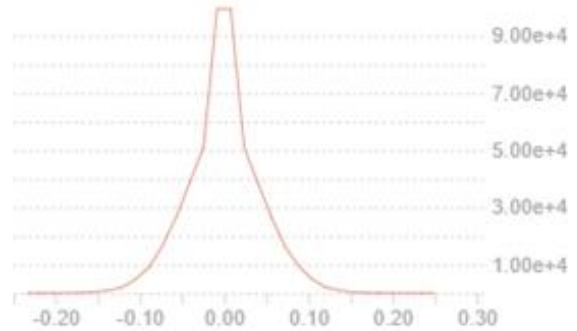


Tags matching /kernel*/

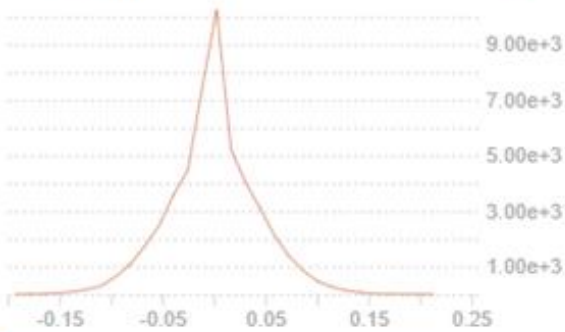
dense_1/kernel_0



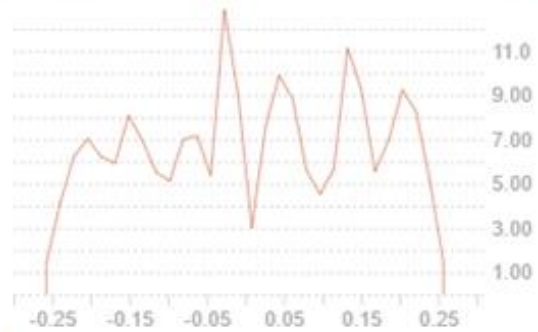
dense_2/kernel_0



dense_3/kernel_0



dense_4/kernel_0

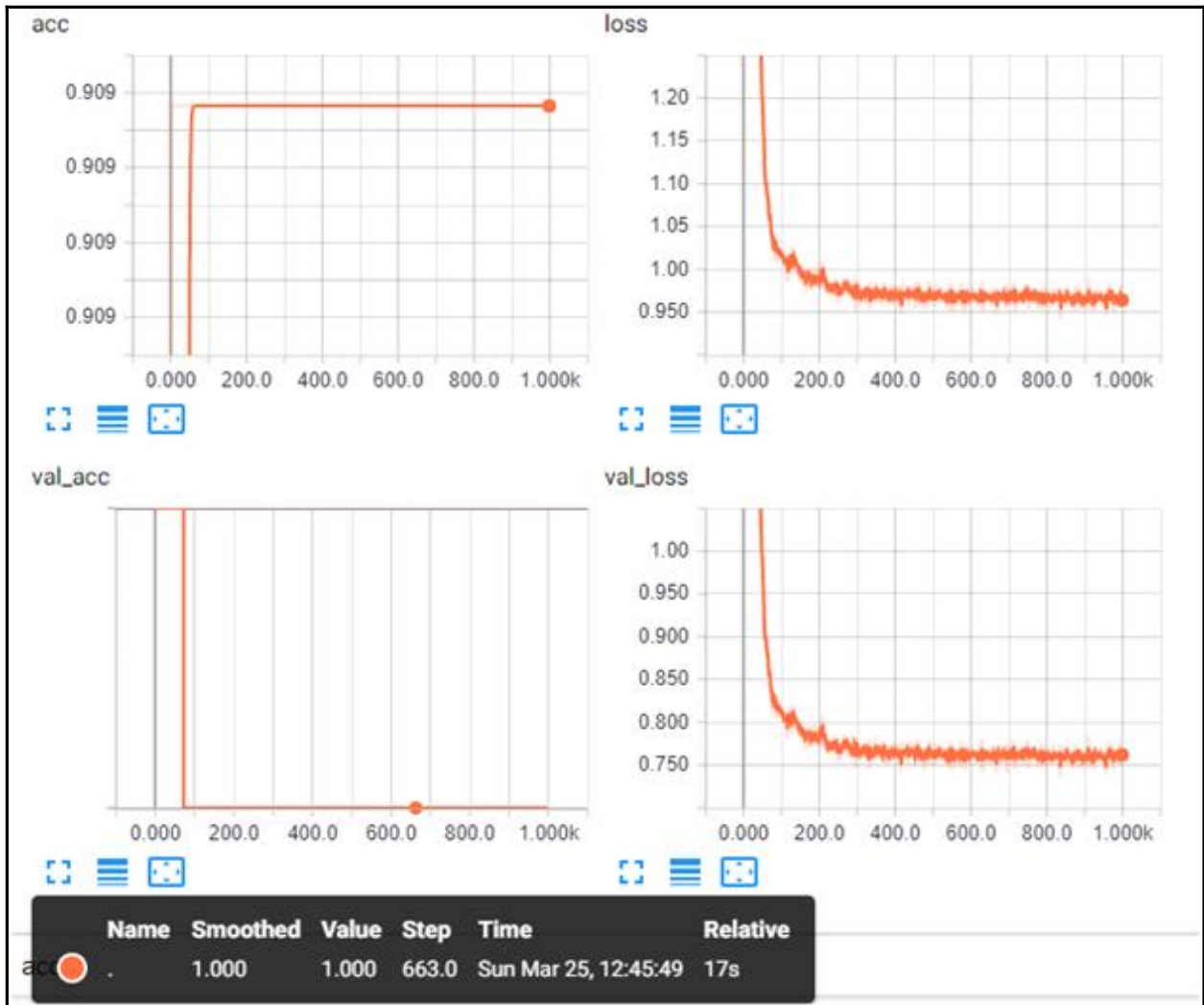


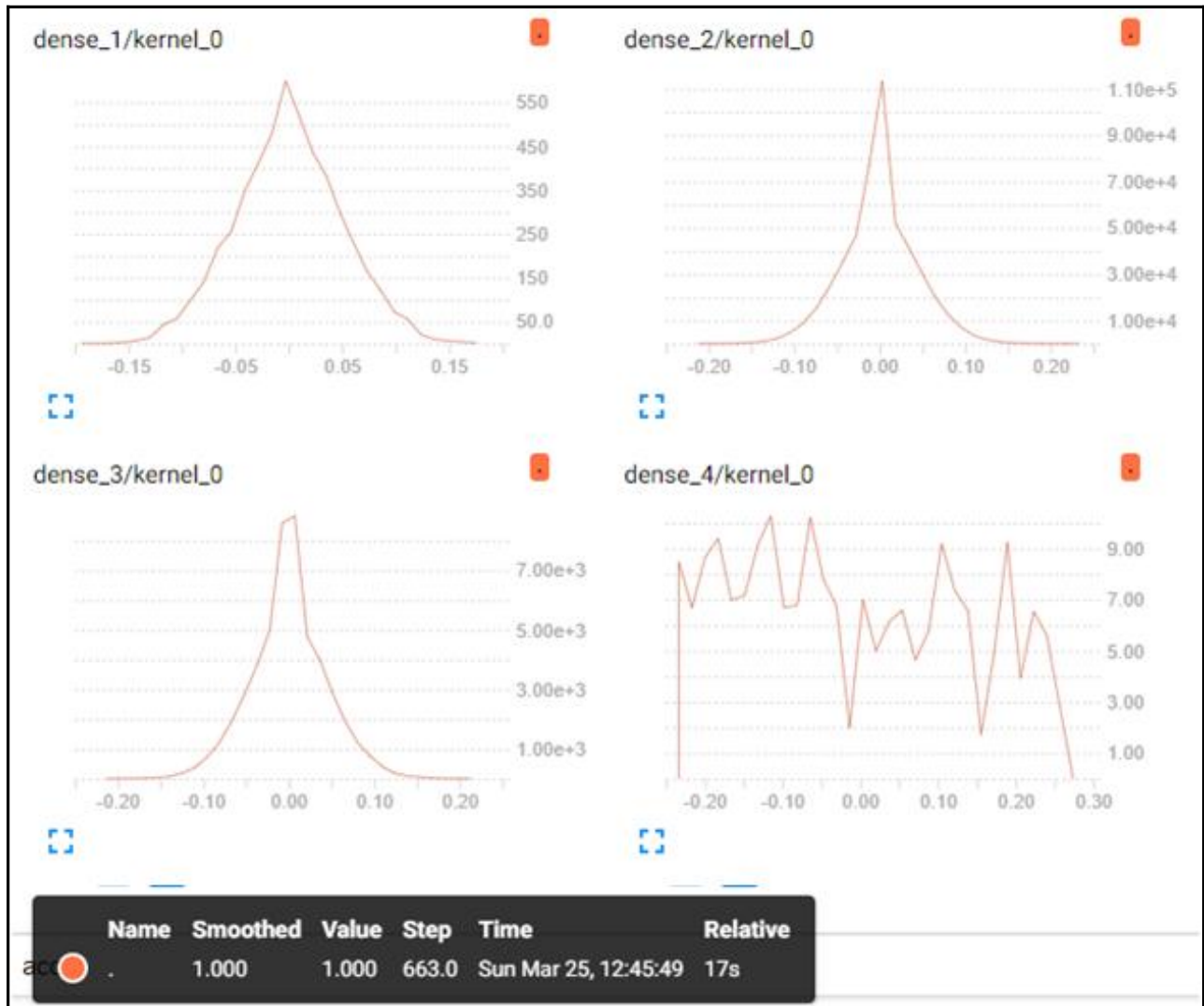
```
from keras.regularizers import l1
from keras import backend as K
K.clear_session()
model = Sequential()
model.add(Dense(1000, input_dim=5, kernel_initializer='normal', kernel_regularizer=l1(0.001), activation='relu'))
model.add(Dense(500, kernel_initializer='normal', kernel_regularizer=l1(0.001), activation='relu'))
model.add(Dense(100, kernel_initializer='normal', kernel_regularizer=l1(0.001), activation='relu'))

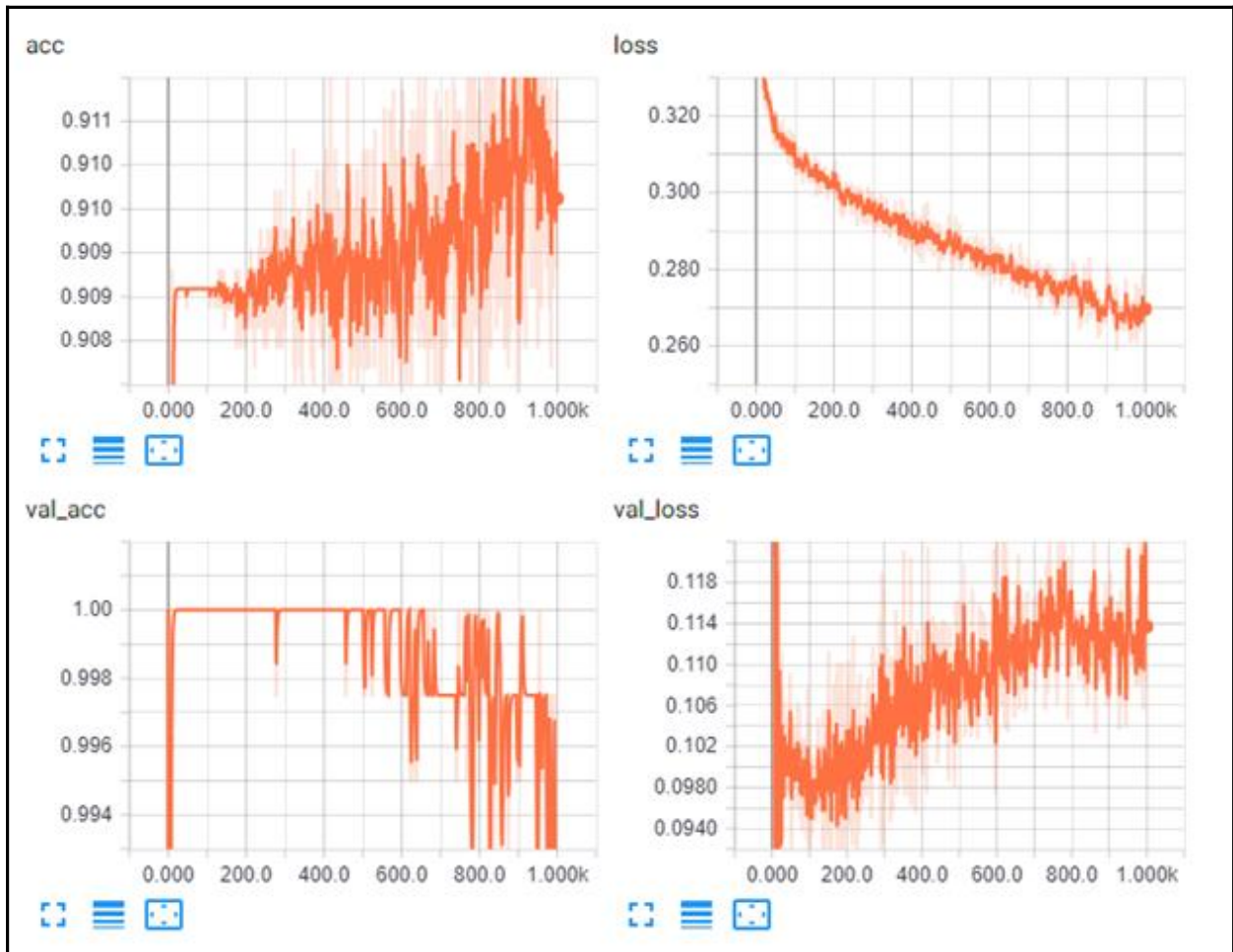
model.add(Dense(2, activation='softmax'))
tensorboard = TensorBoard(log_dir = "/content/datalab/logs/tensor_22", histogram_freq=1000, batch_size=5000)

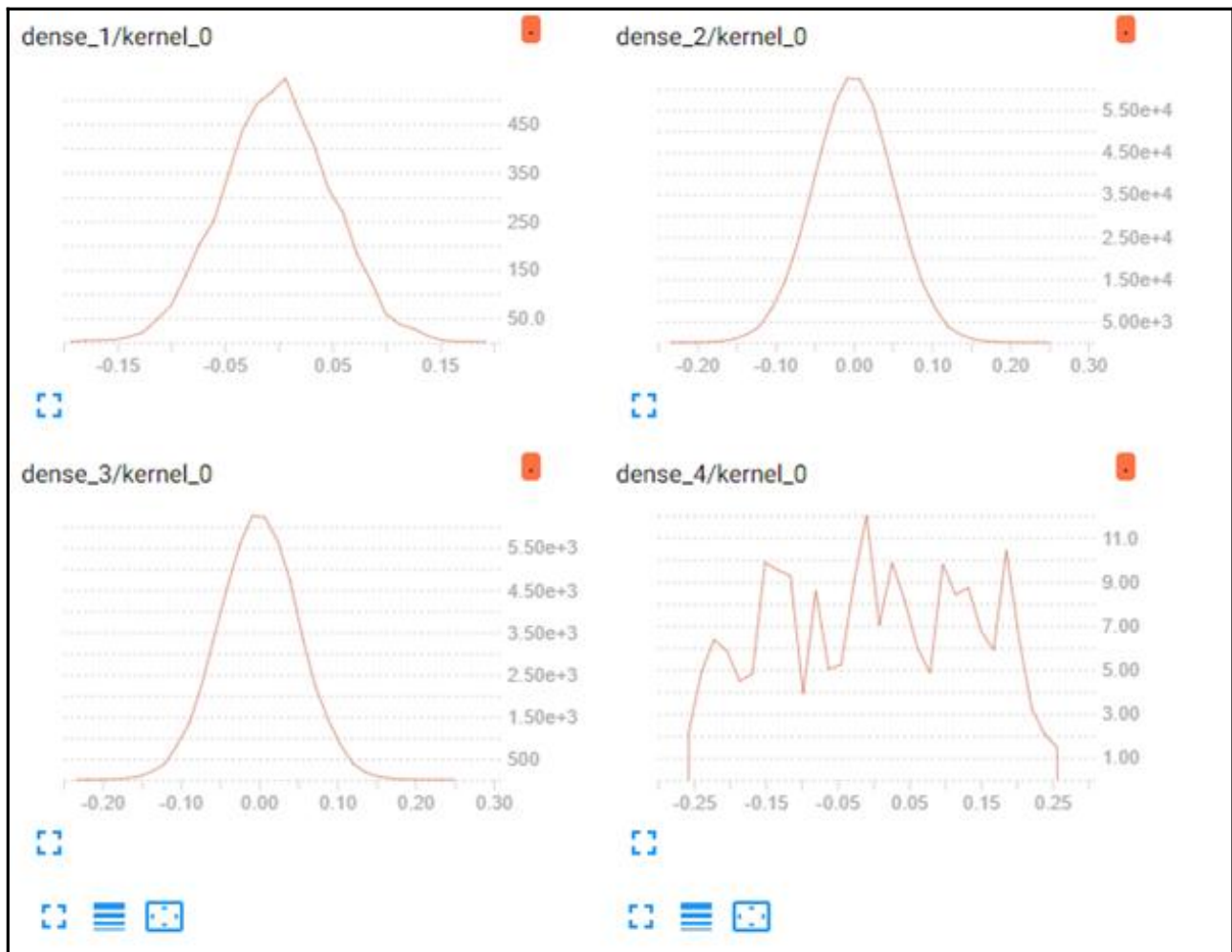
adam = Adam(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
```

```
model.fit(final_x, final_y, validation_data=(test_x, test_y), epochs=1000, batch_size=5000, verbose=2, callbacks=[tensorboard])
```

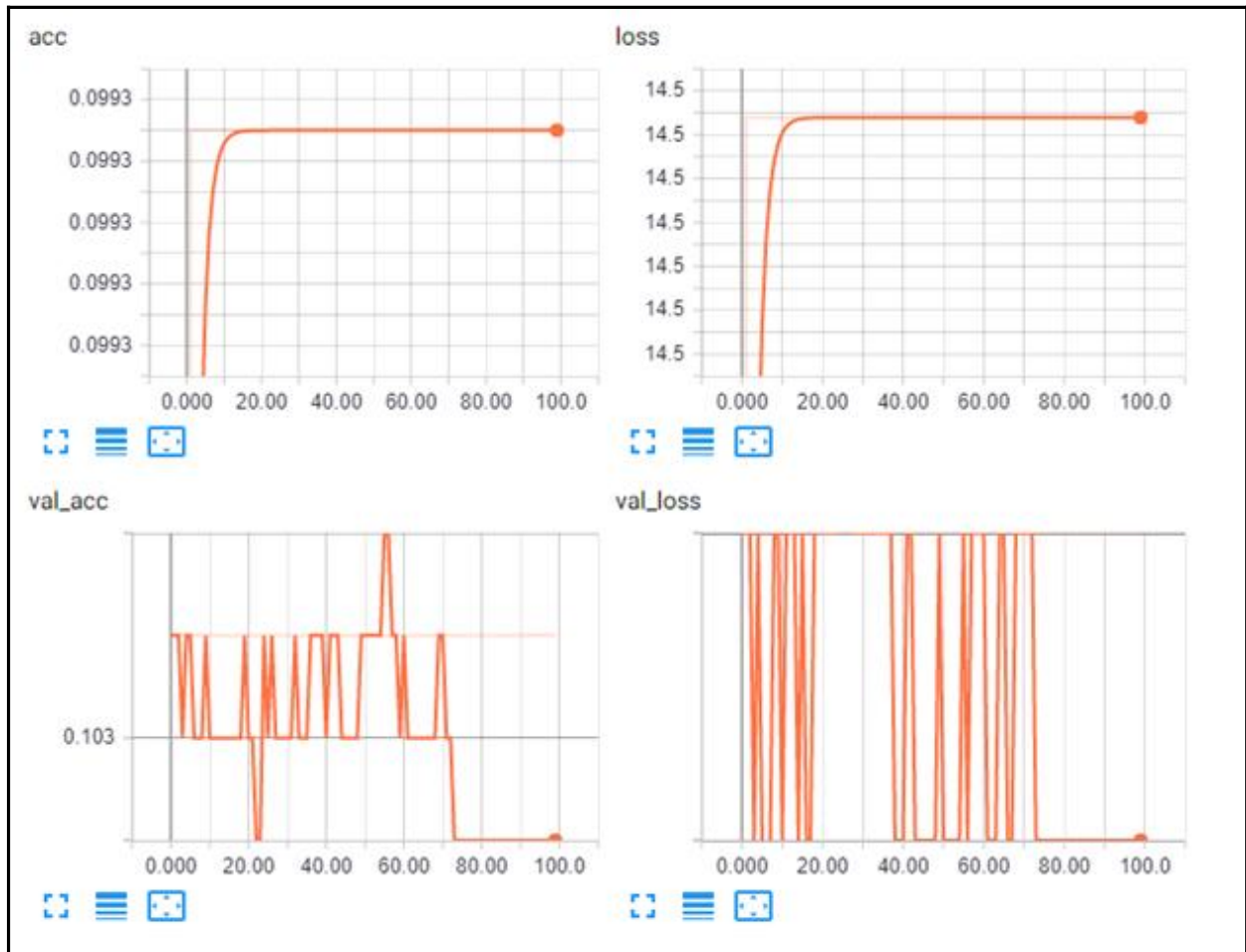


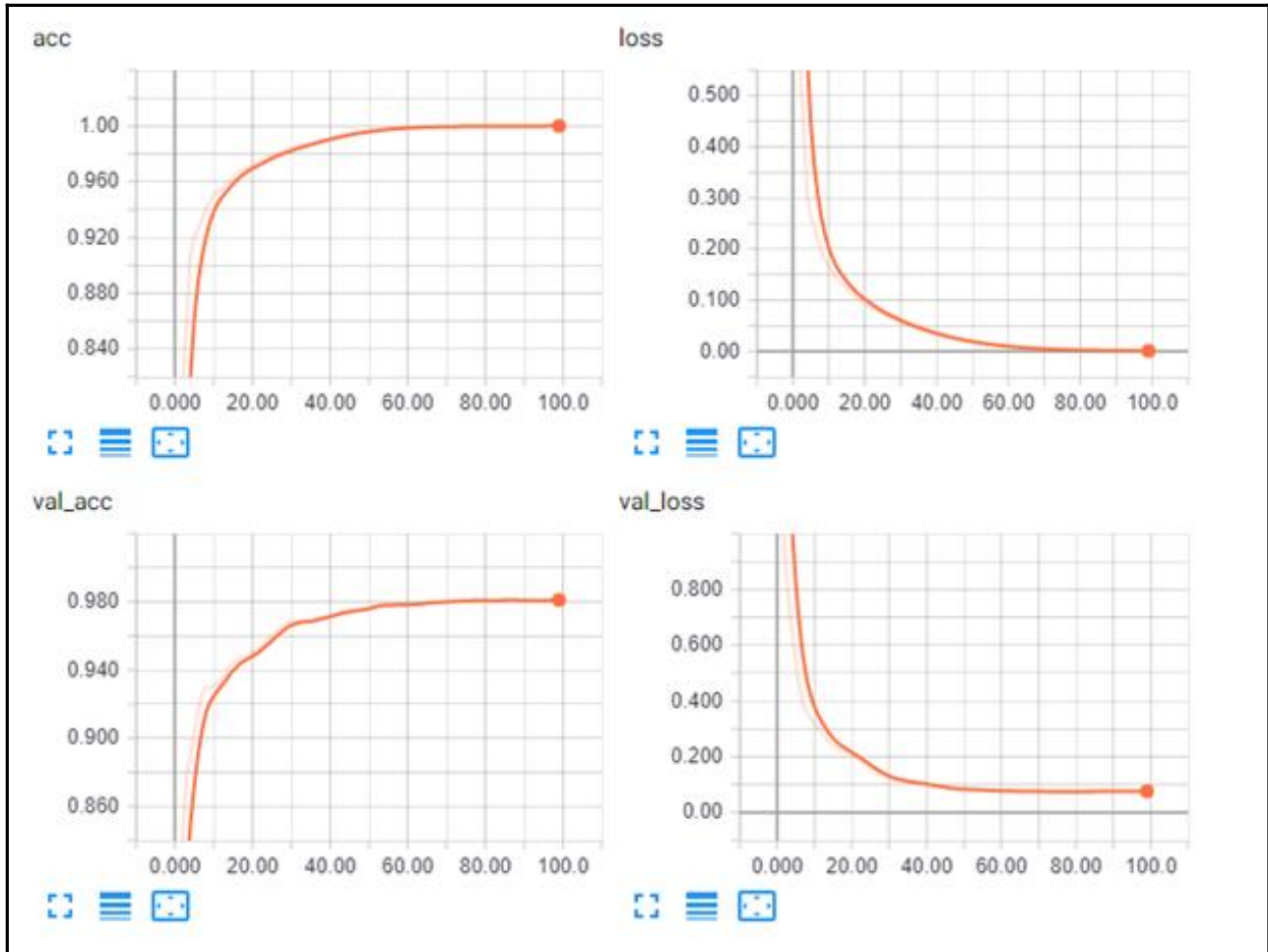


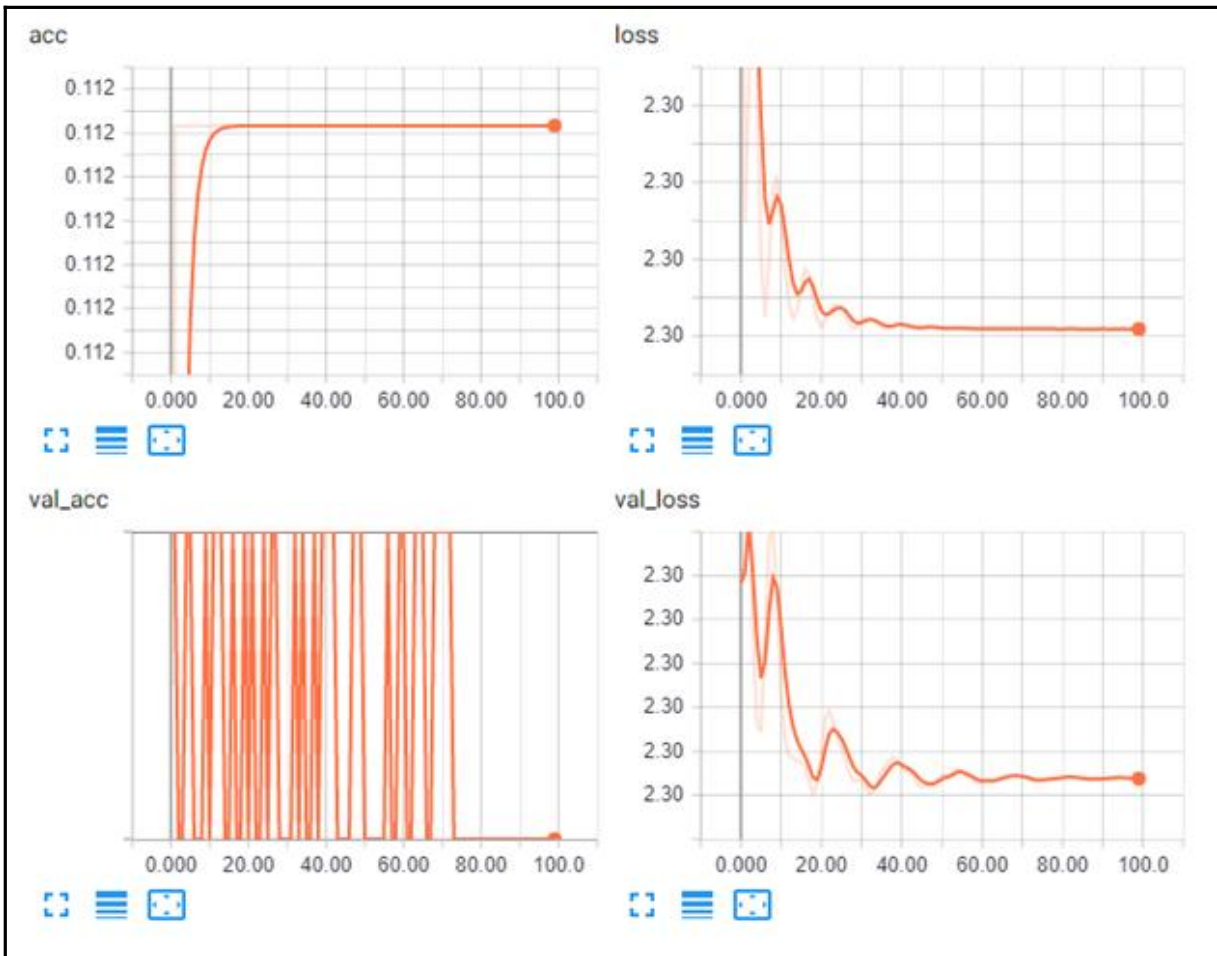




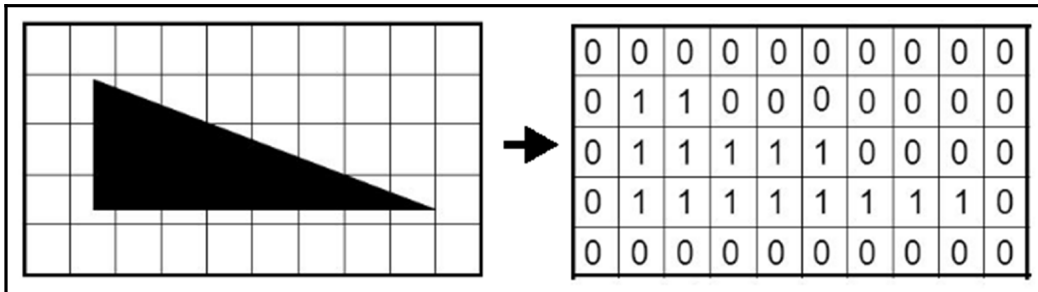
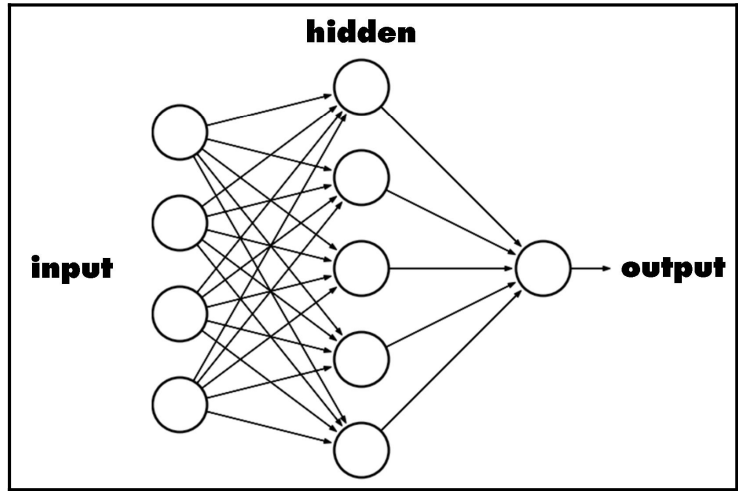
```
# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]
```

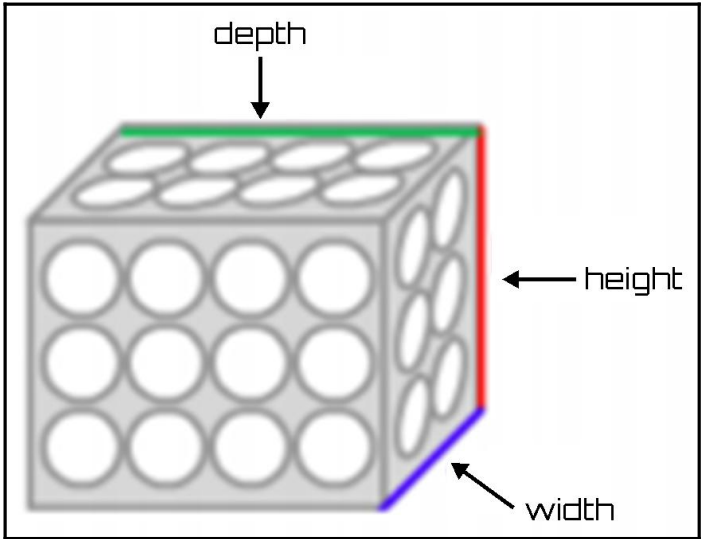
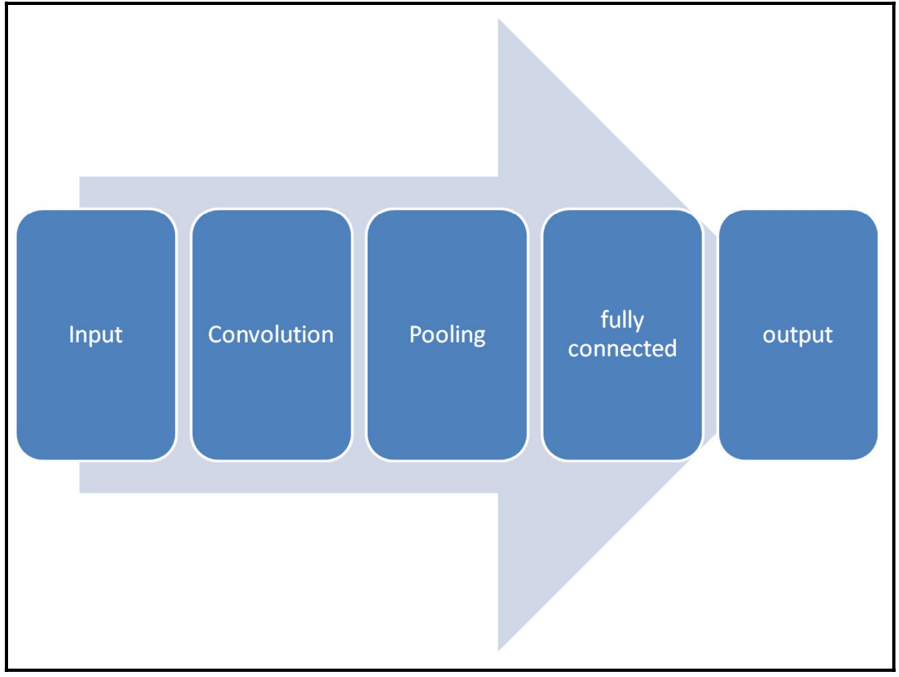


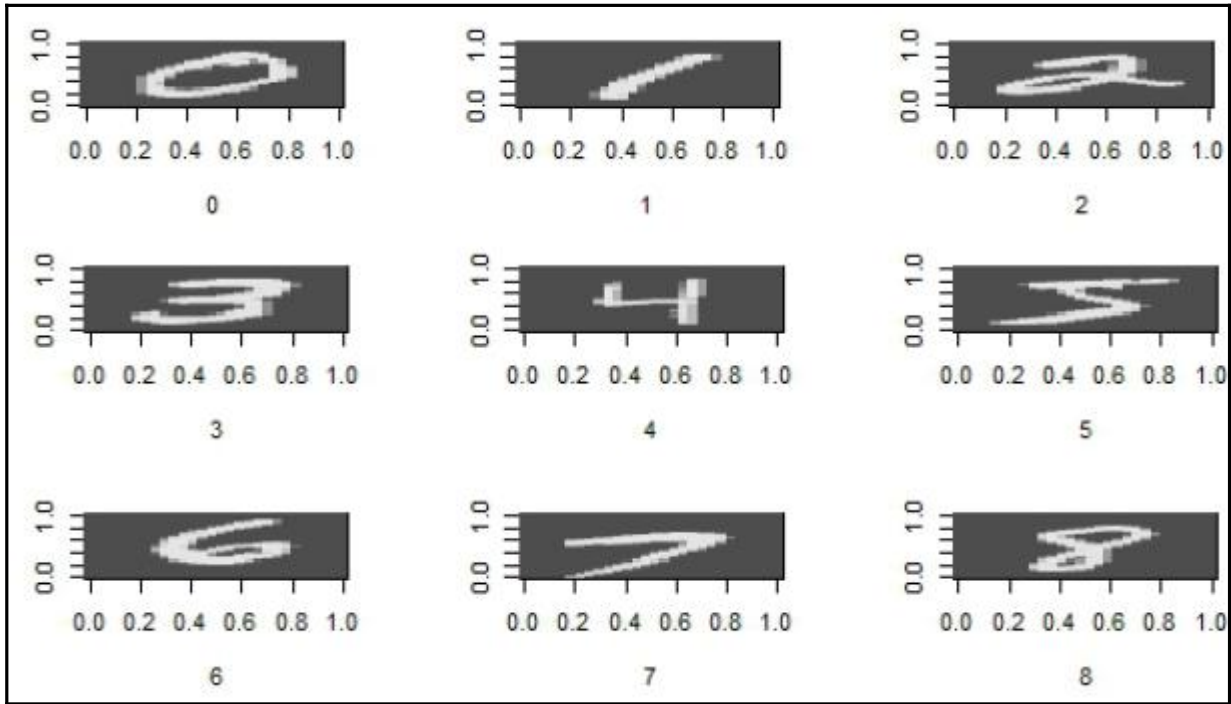




Chapter 13: Beyond Feedforward Networks – CNN and RNN







Google Cloud Platform MainProject

DASHBOARD ACTIVITY

Project info

- Project name: MainProject
- Project ID: progetto-1-191608
- Project number: 864125566776

Resources

- Cloud Storage: 2 buckets

RPI APIs

Requests (requests/sec)

Google Cloud Platform status

All services normal

Billing

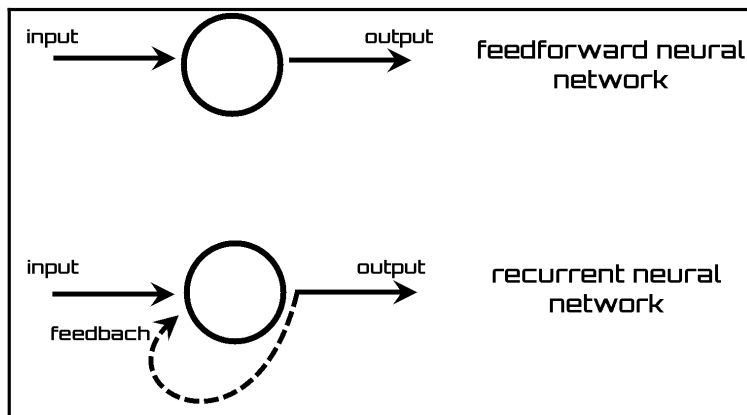
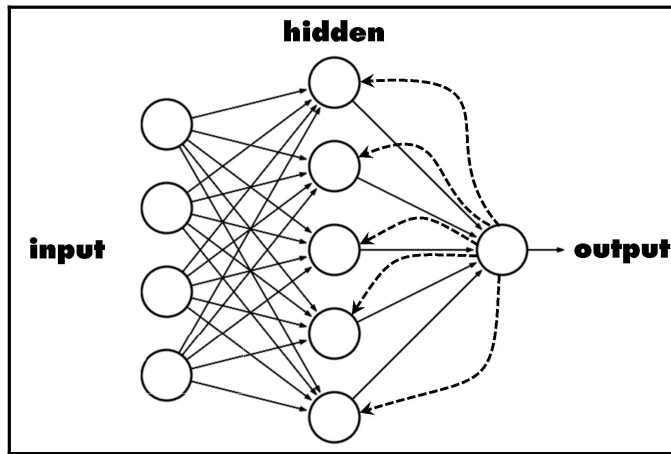
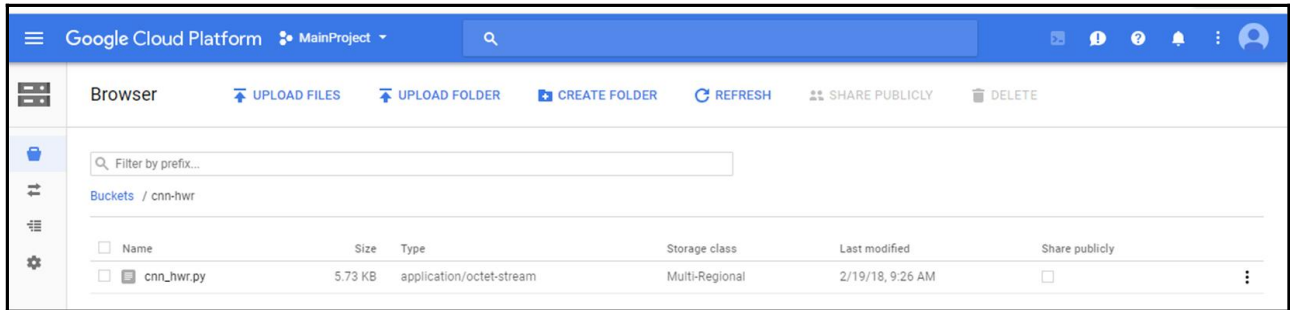
Estimated charges: EUR €0.00

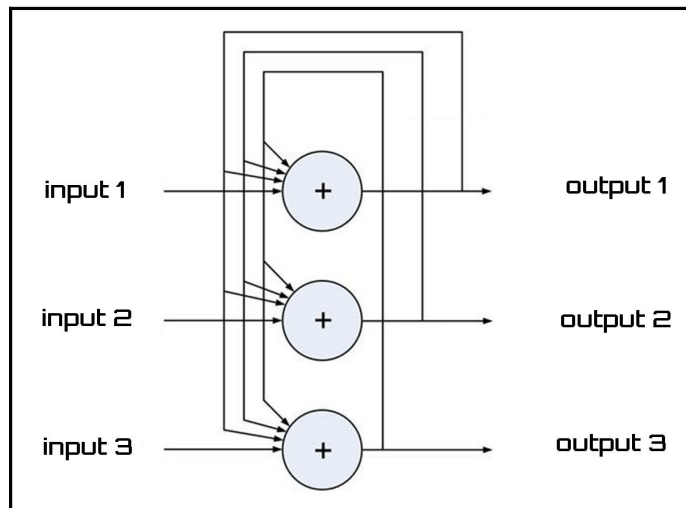
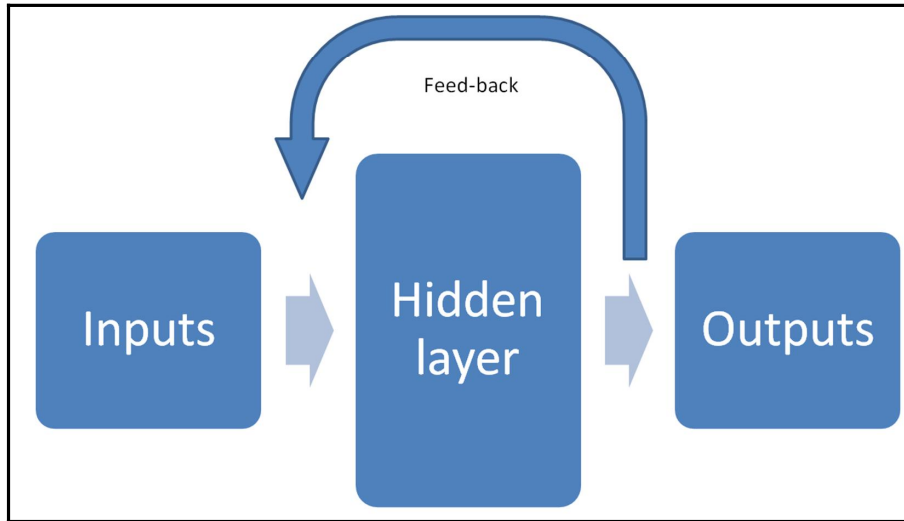
Error Reporting

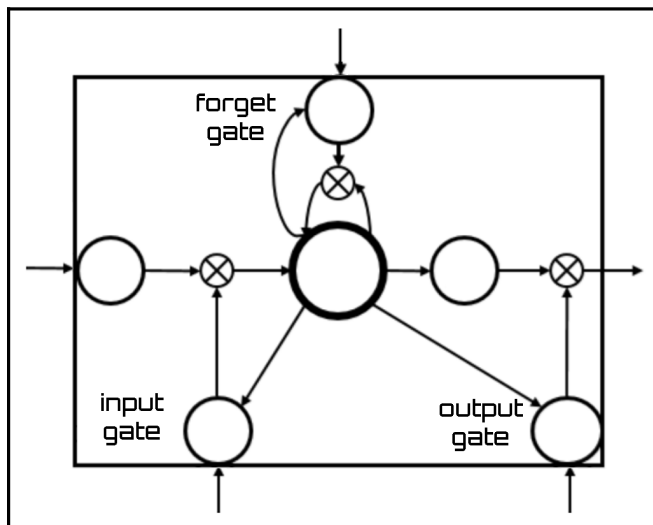
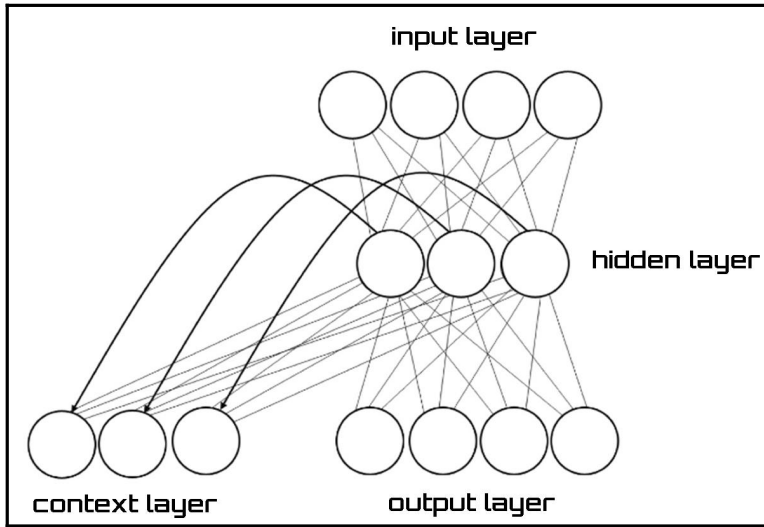
Activate Google Cloud Shell

```

progetto-1-191608 x
Welcome to Cloud Shell! Type "help" to get started.
giuseppe_ciaburro@progetto-1-191608:~$
  
```







Google Cloud Platform MainProject

DASHBOARD ACTIVITY

Project info

Project name: MainProject

Project ID: progetto-1-191608

Project number: 864125566776

Resources

Cloud Storage: 2 buckets

API APIs

Requests (requests/sec)

Google Cloud Platform status

All services normal

Billing

Estimated charges: EUR €0.00

```
Welcome to Cloud Shell! Type "help" to get started.  
giuseppe_ciaburro@progetto-1-191608:~$
```

Browser

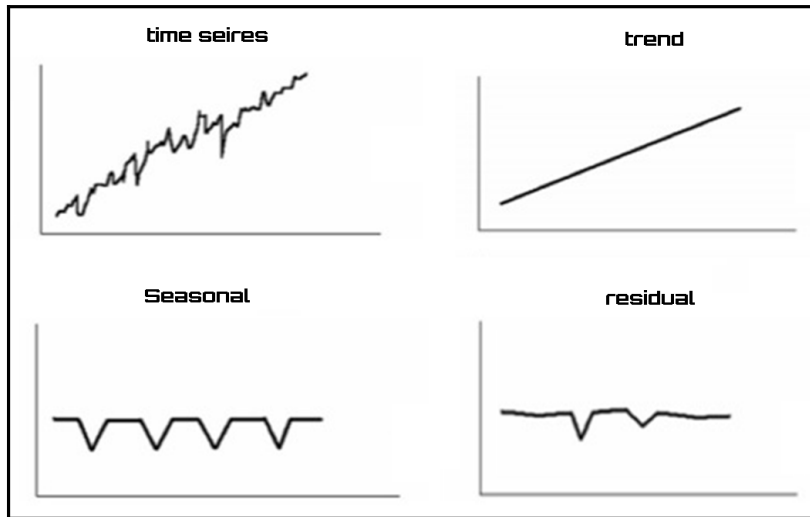
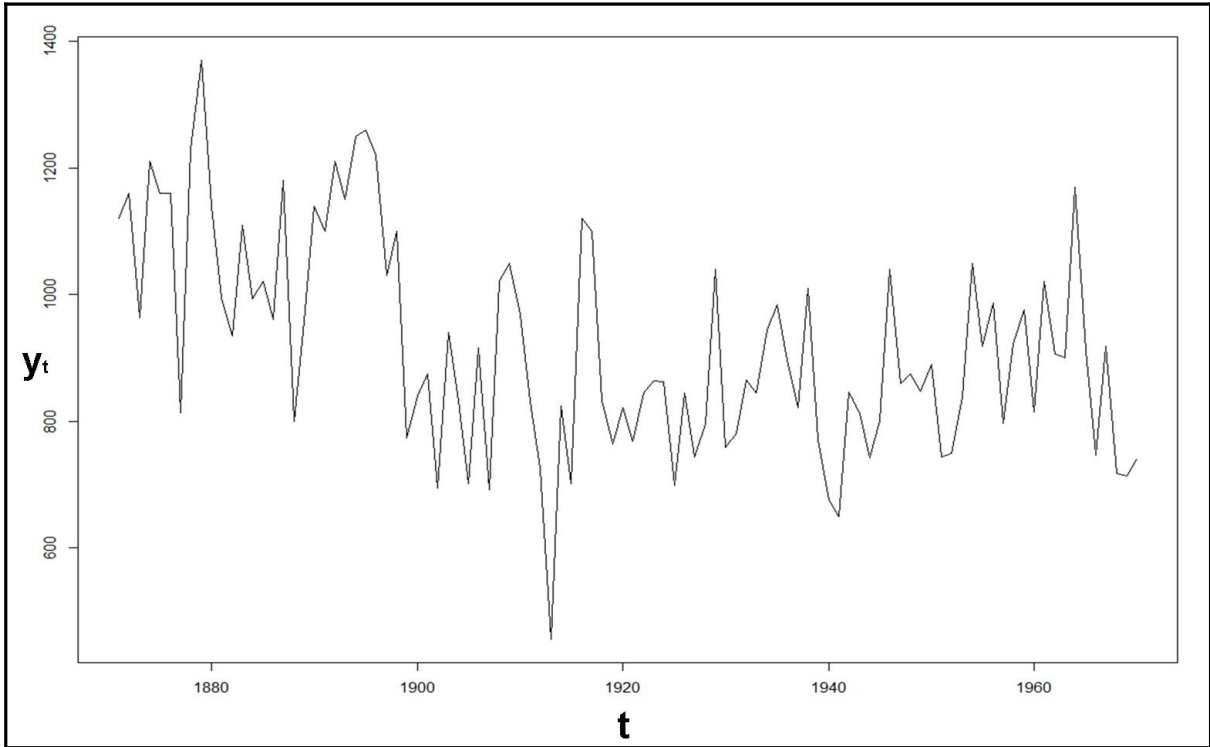
UPLOAD FILES | UPLOAD FOLDER | CREATE FOLDER | REFRESH | SHARE PUBLICLY | DELETE

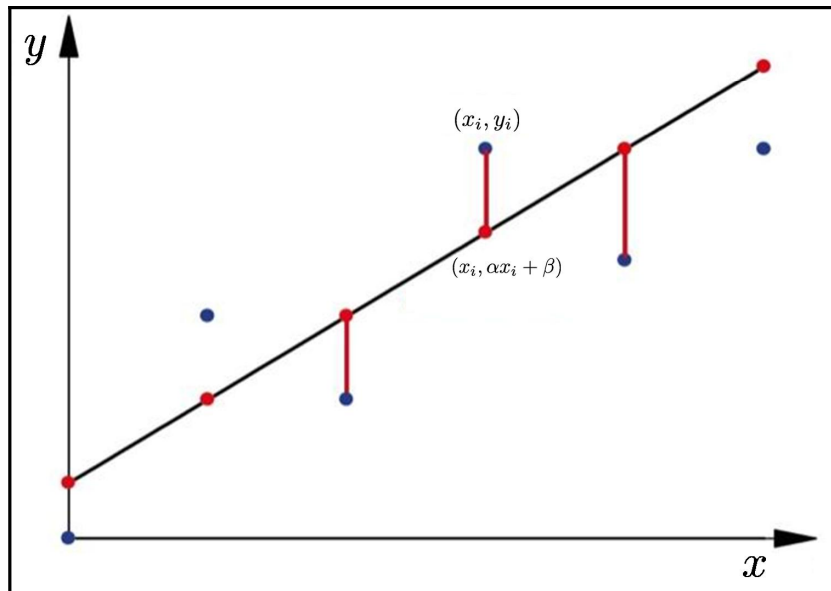
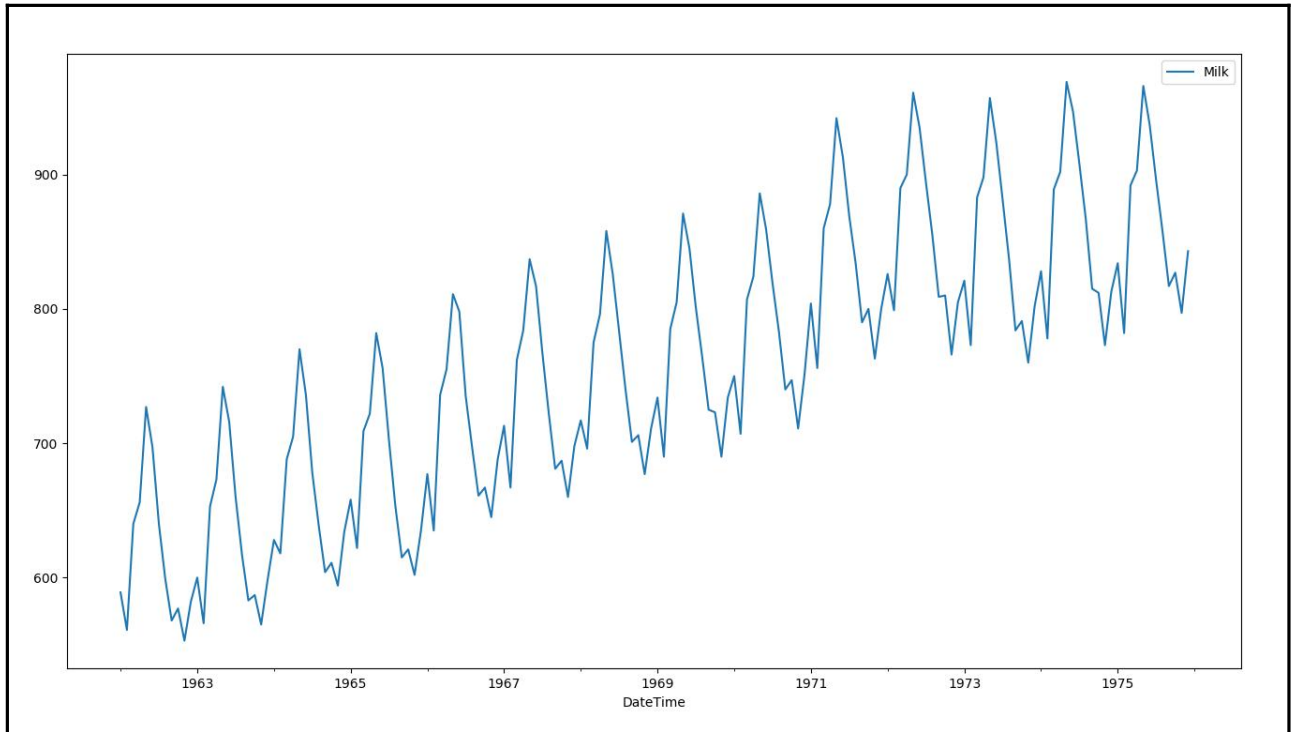
Filter by prefix...

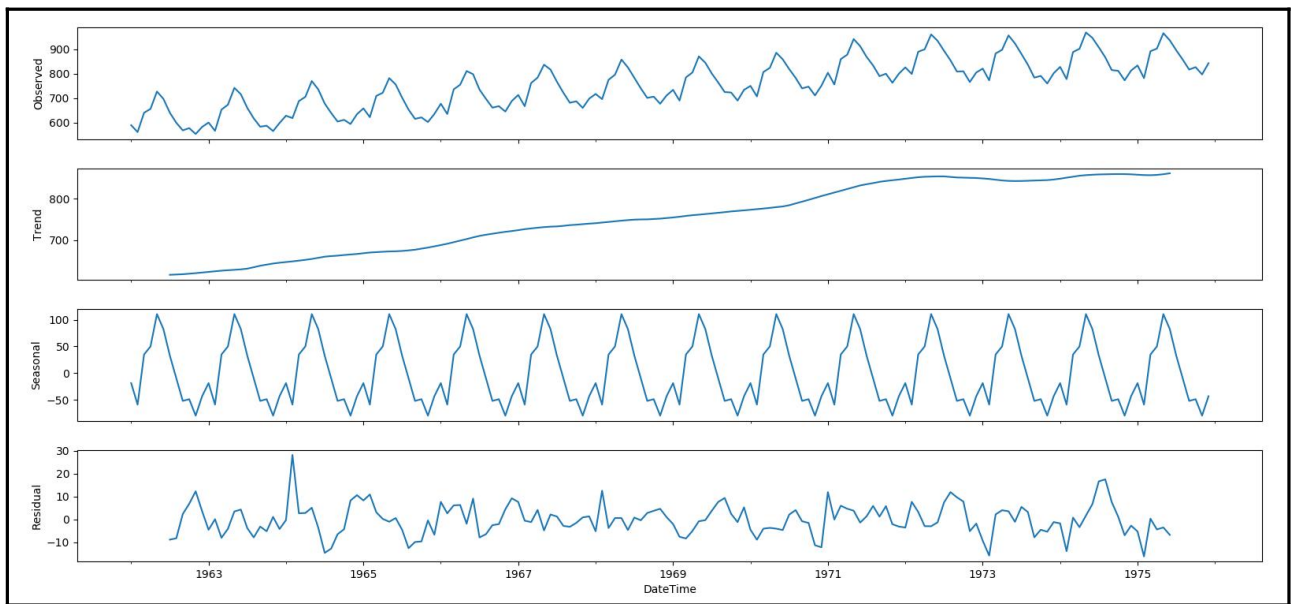
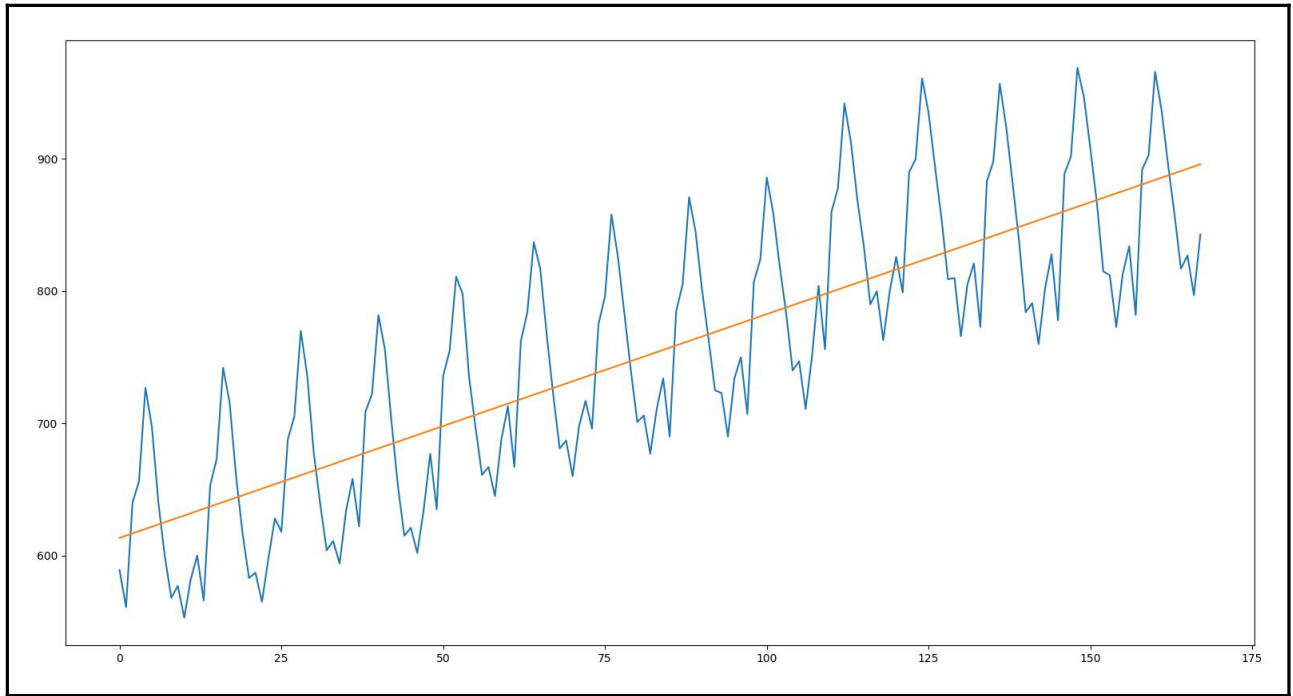
Buckets / rnn_hwr

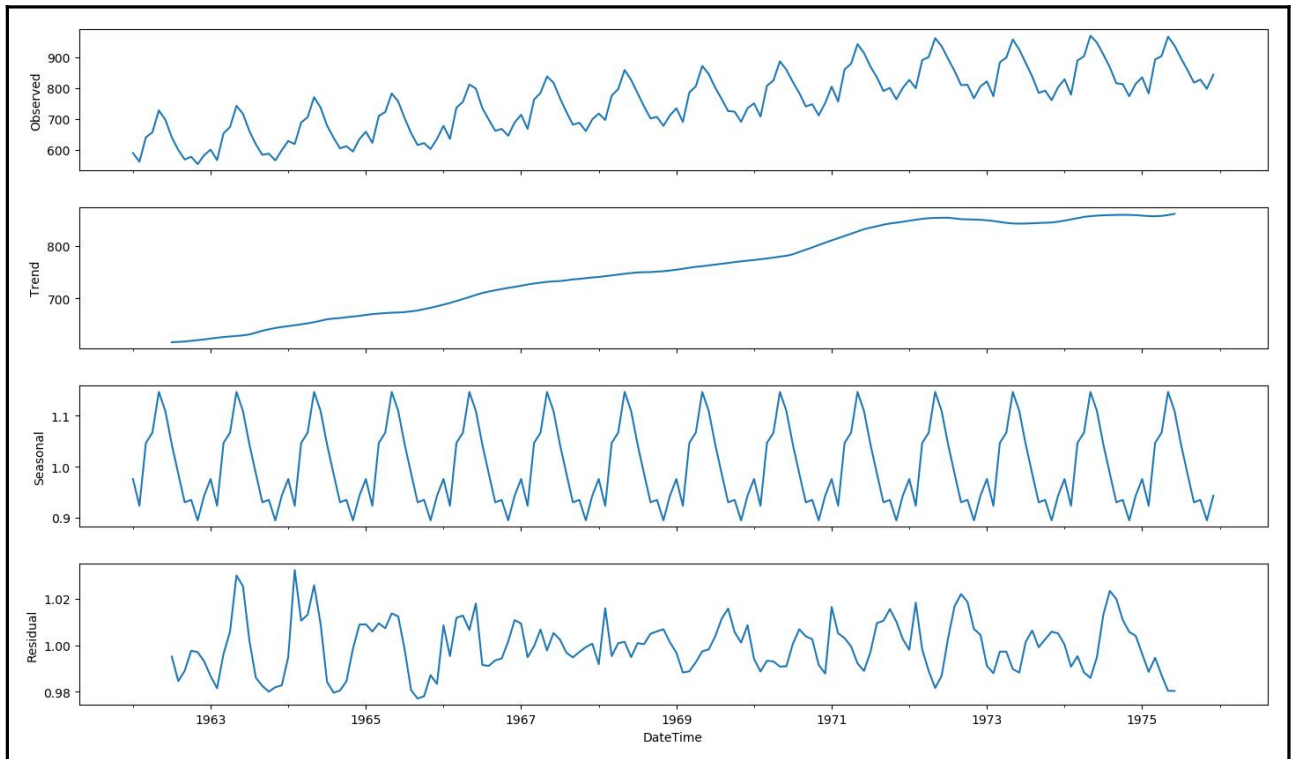
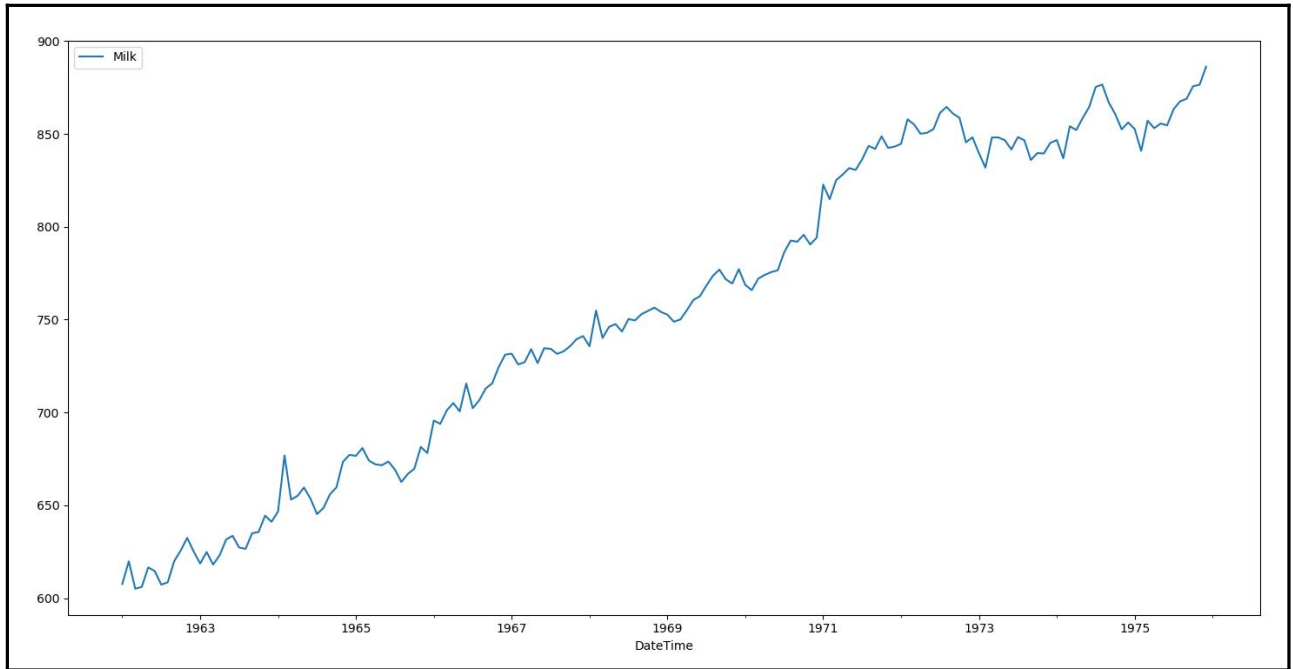
Name	Size	Type	Storage class	Last modified	Share publicly
rnn_hwr.py	5.73 KB	application/octet-stream	Multi-Regional	2/19/18, 9:26 AM	<input type="checkbox"/>

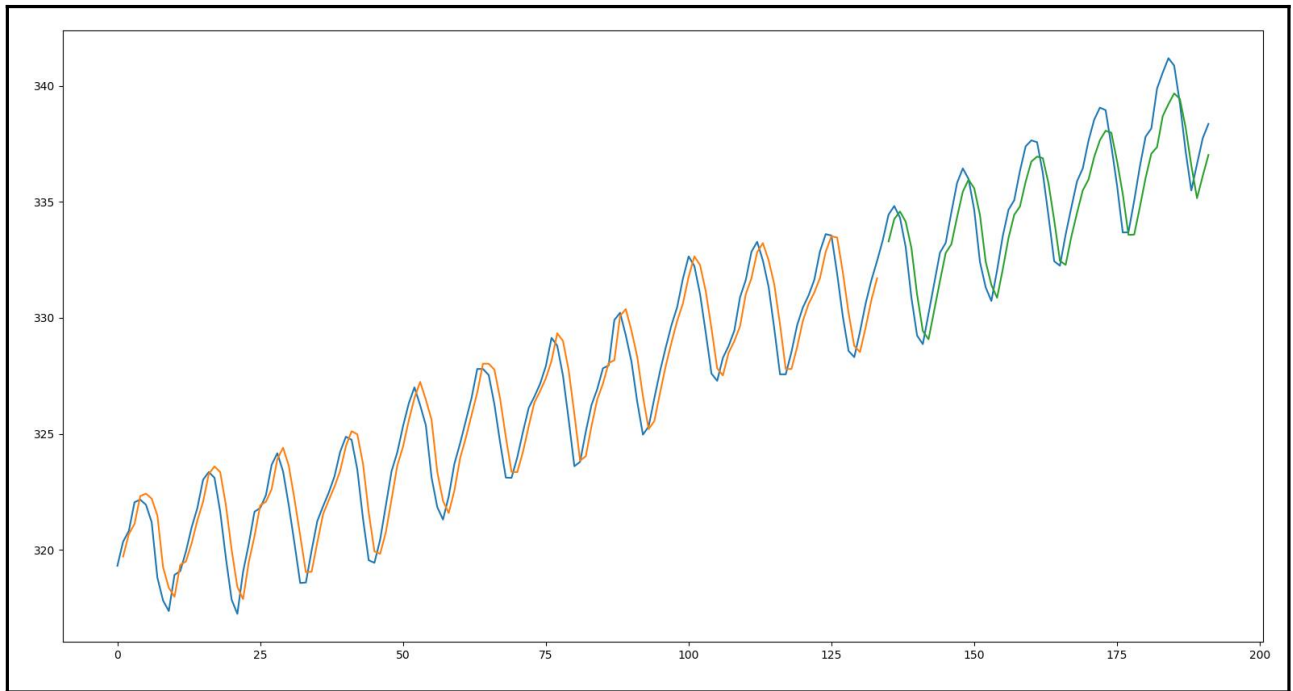
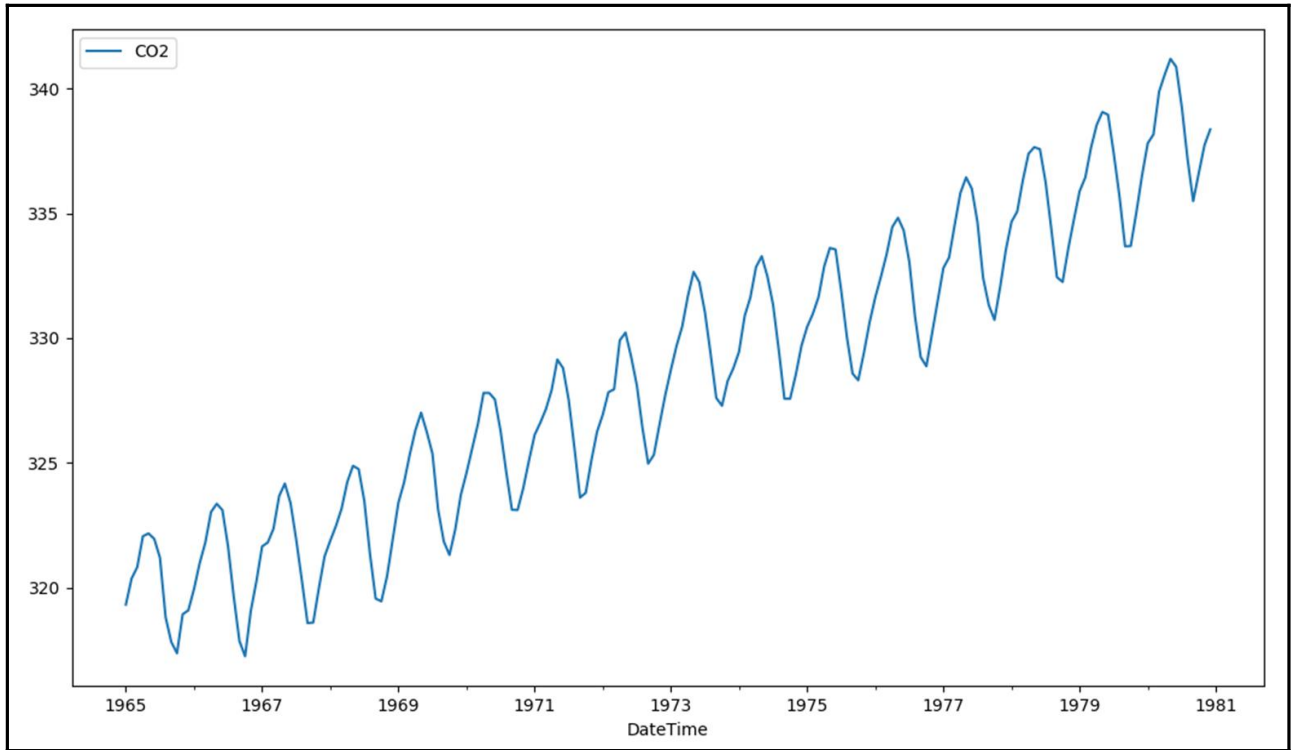
Chapter 14: Time Series with LSTMs



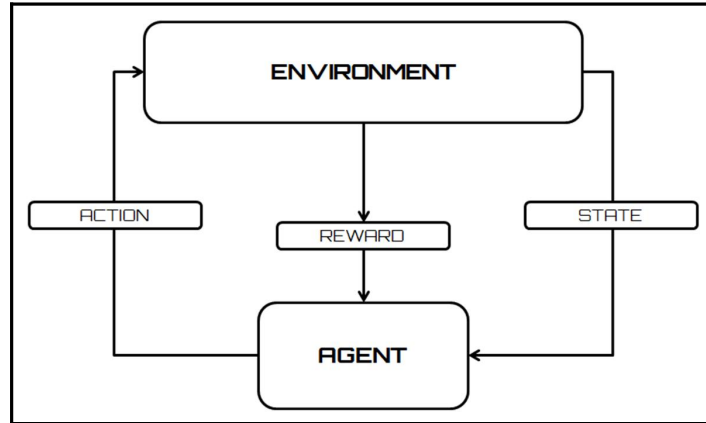









Chapter 15: Reinforcement Learning



Q table

	ACTION					
STATE	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	0	0	0	0	0
	0	0	0	0	0	0


Environments Documentation




Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.


[View documentation >](#)
[View on GitHub >](#)



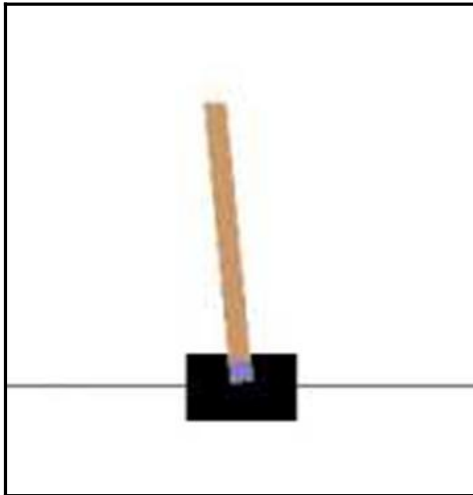
RandomAgent on Pendulum-v0



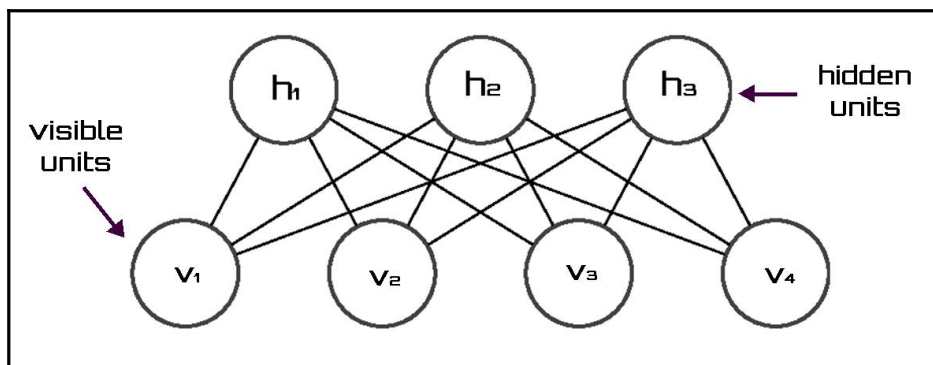
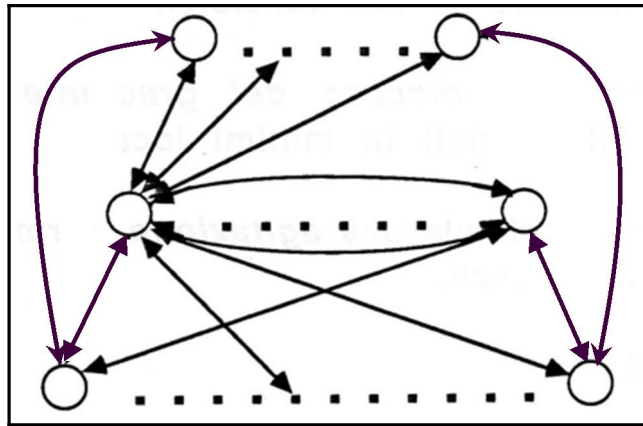
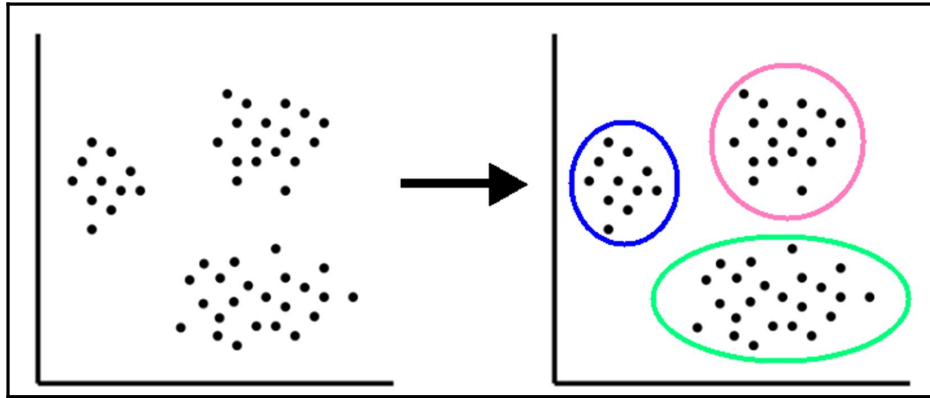
RandomAgent on SpaceInvaders-v0

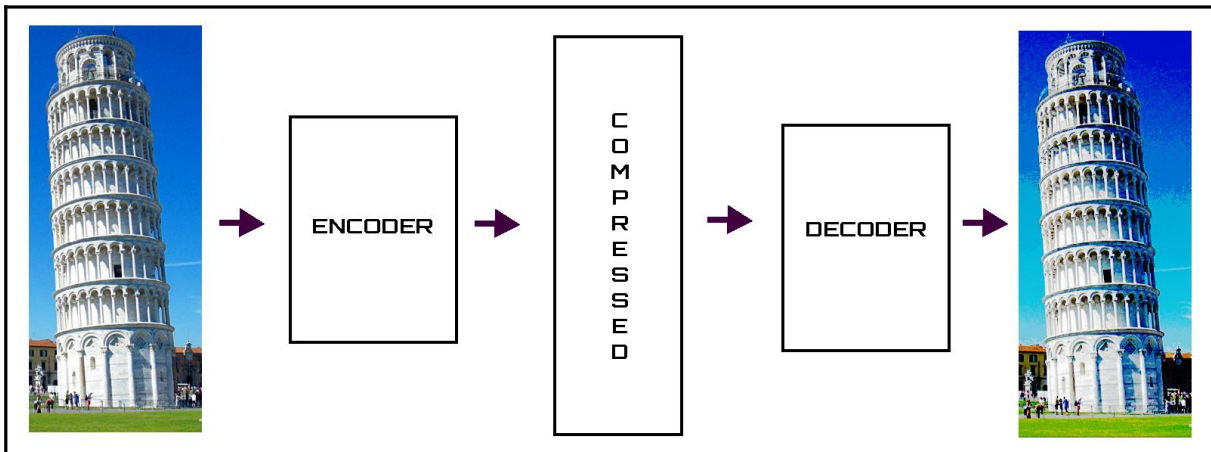
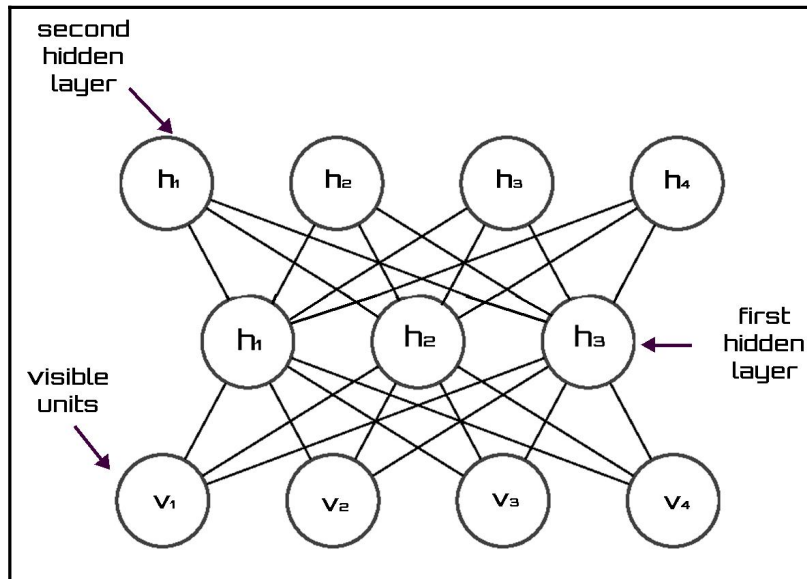


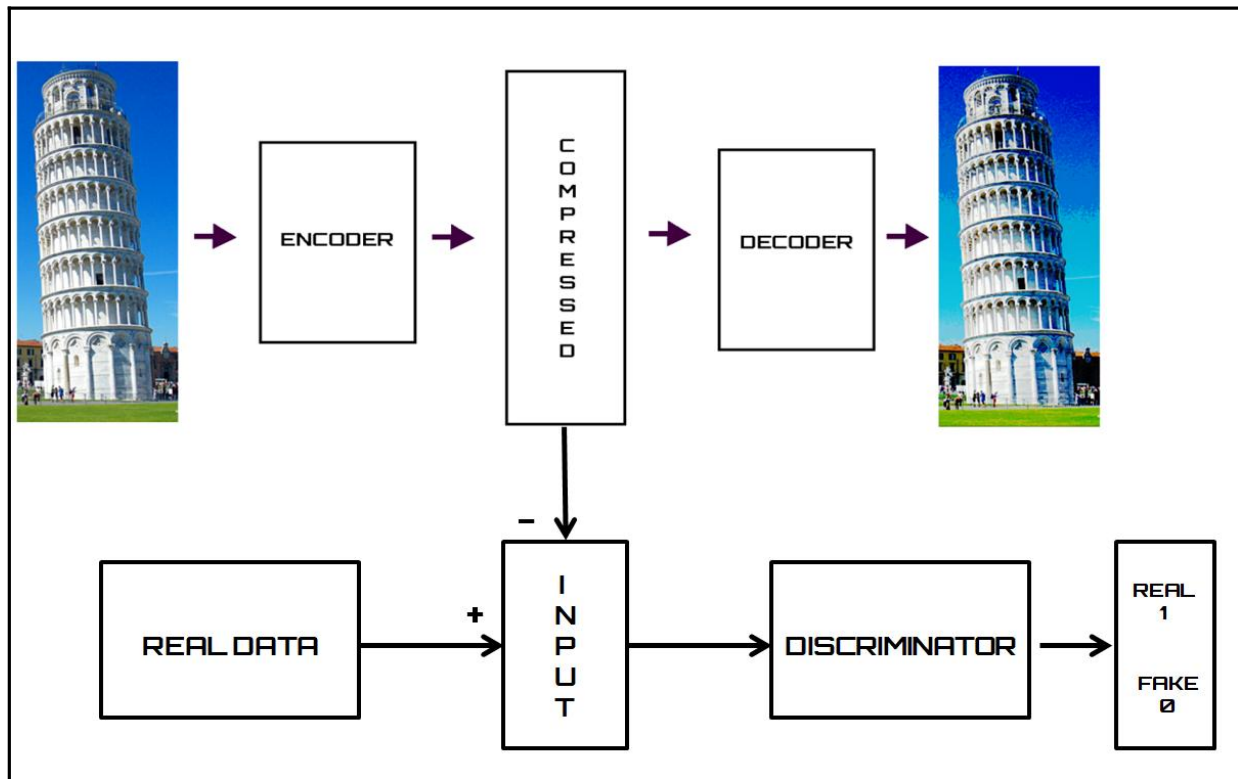
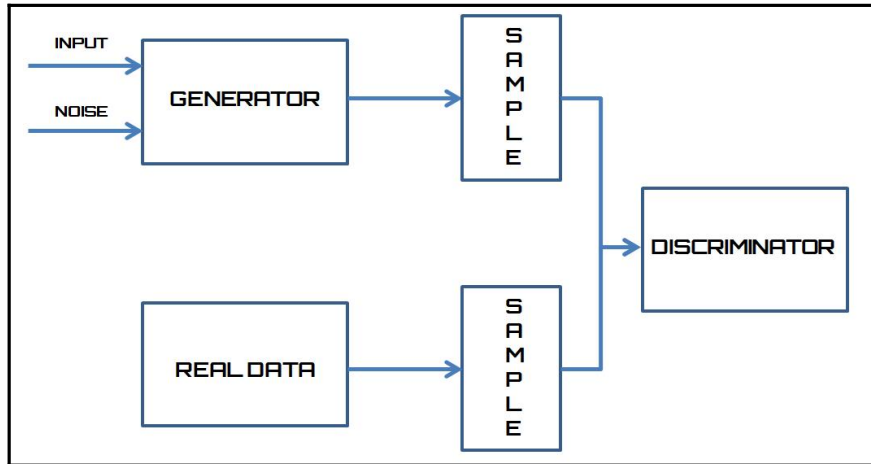
Open source interface to reinforcement learning tasks.
The `gym` library provides an easy-to-use suite of reinforcement



Chapter 16: Generative Neural Networks

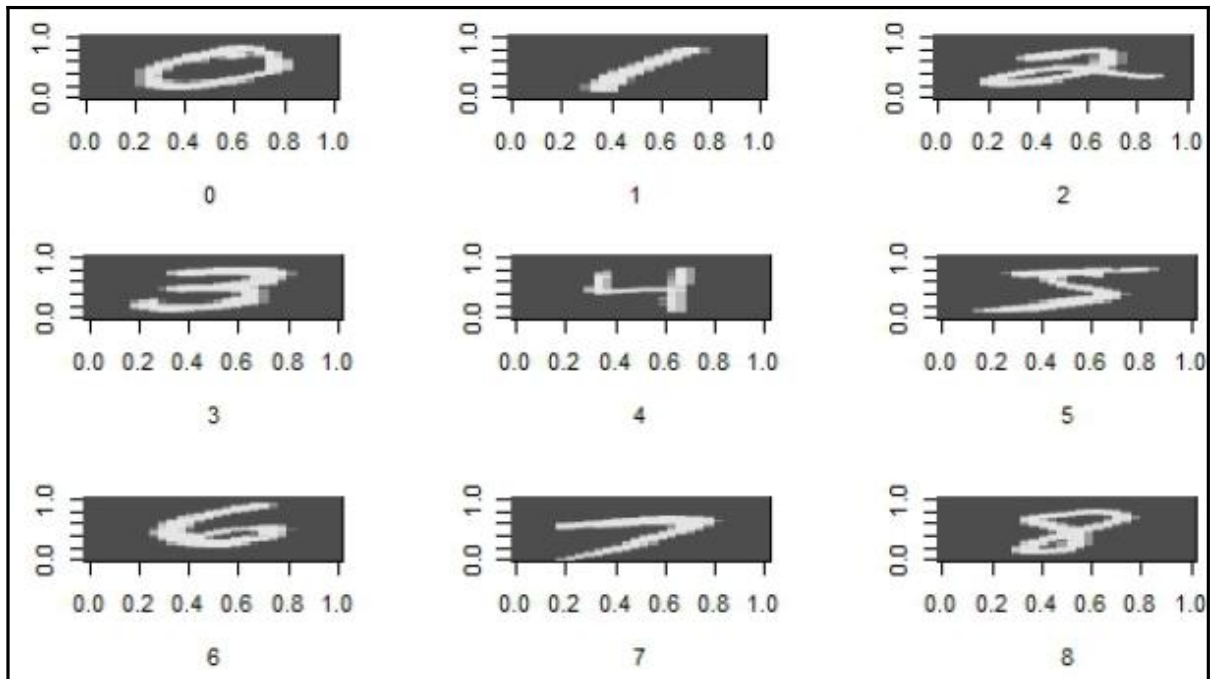






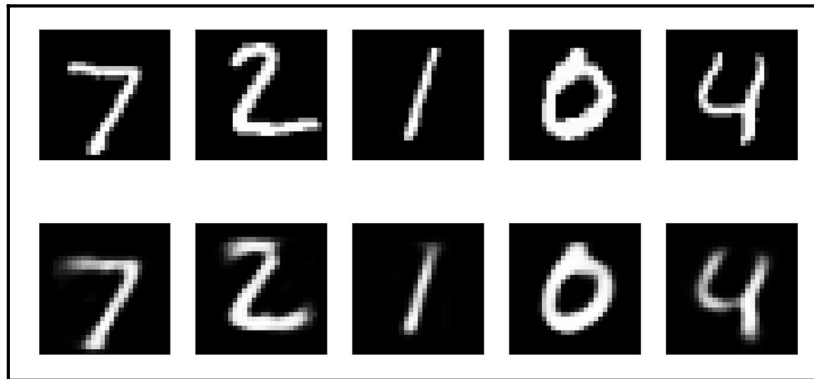
The RBM model:

```
Predict: [1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 0 1 0 1 0 1 1 1 1  
1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0  
1 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0  
0 1 1]  
Real: [1 0 1 0 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 0  
1 0 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 0  
1 0 0 0 1 0 1 0 1 0 1 1 1 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0  
1 1 1]
```

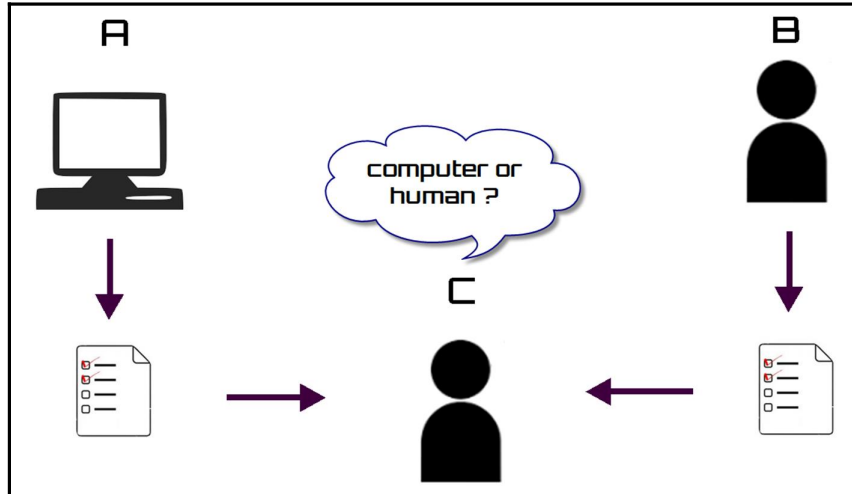



```
Prompt dei comandi
Epoch 88/100
60000/60000 [=====] - 6s 100us/step - loss: 0.0979 - val_loss: 0.0963
Epoch 89/100
60000/60000 [=====] - 6s 98us/step - loss: 0.0979 - val_loss: 0.0962
Epoch 90/100
60000/60000 [=====] - 6s 101us/step - loss: 0.0978 - val_loss: 0.0962
Epoch 91/100
60000/60000 [=====] - 6s 102us/step - loss: 0.0977 - val_loss: 0.0961
Epoch 92/100
60000/60000 [=====] - 6s 103us/step - loss: 0.0976 - val_loss: 0.0960
Epoch 93/100
60000/60000 [=====] - 6s 108us/step - loss: 0.0976 - val_loss: 0.0960
Epoch 94/100
60000/60000 [=====] - 7s 118us/step - loss: 0.0975 - val_loss: 0.0959
Epoch 95/100
60000/60000 [=====] - 6s 106us/step - loss: 0.0974 - val_loss: 0.0959
Epoch 96/100
60000/60000 [=====] - 7s 113us/step - loss: 0.0974 - val_loss: 0.0958
Epoch 97/100
60000/60000 [=====] - 7s 121us/step - loss: 0.0973 - val_loss: 0.0957
Epoch 98/100
60000/60000 [=====] - 8s 141us/step - loss: 0.0973 - val_loss: 0.0957
Epoch 99/100
60000/60000 [=====] - 8s 125us/step - loss: 0.0972 - val_loss: 0.0956
Epoch 100/100
60000/60000 [=====] - 7s 120us/step - loss: 0.0971 - val_loss: 0.0956

C:\pythonscript\GEN>
```



Chapter 17: Chatbots



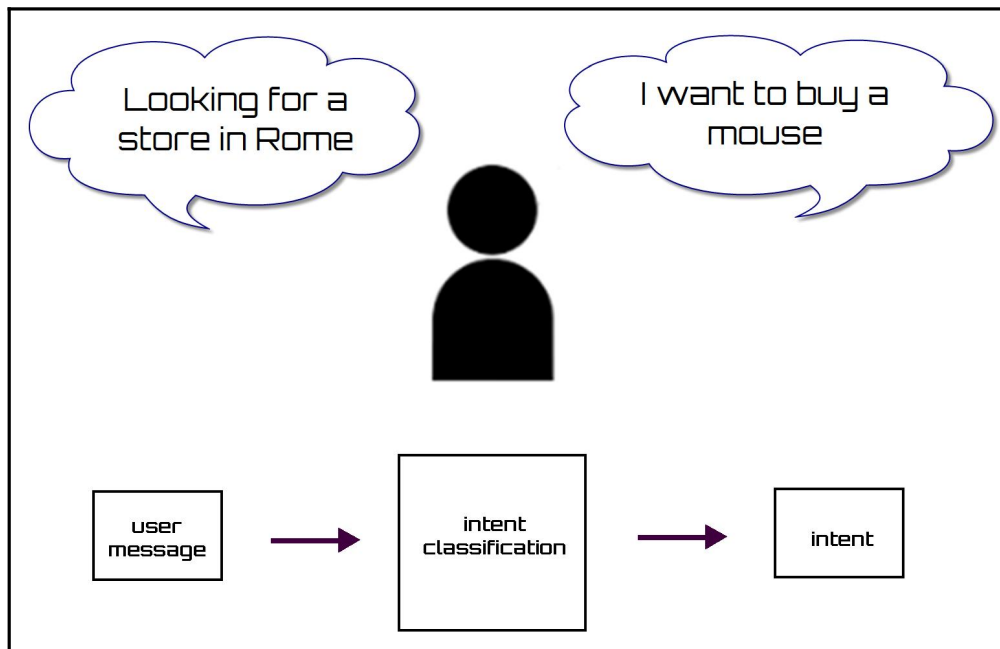
```
=====
EEEEEEEE L      IIIIII ZZZZZZZ      AAA
E         L      I      Z      A   A
E         L      I      Z      A   A
EEEEEE   L      I      Z      A   A
E         L      I      Z      AAAAAA
E         L      I      Z      A   A
EEEEEEEE LLLLLLL IIIIII ZZZZZZ      A   A
=====
ELIZA > (PLEASE TYPE IN ALL CAPS) WHAT'S YOUR NAME DEAR ?
AMIT
ELIZA > HI, I'M ELIZA. WHAT DO YOU WANT TO TALK ABOUT ?
AMIT > CAN I TALK ABOUT MY PROBLEM ?
ELIZA > SURE... !
AMIT > DUE TO ON GOING EXAMS IN COLLEGE, I HAVE A LOT OF STRESS.
ELIZA > PLEASE RELAX A LITTLE. SLEEP WELL.
AMIT > THANKS FOR YOUR ADVICE.
ELIZA > NO MENTION.
AMIT > BYE
ELIZA > BYE AND KEEP IN TOUCH...
=====
```

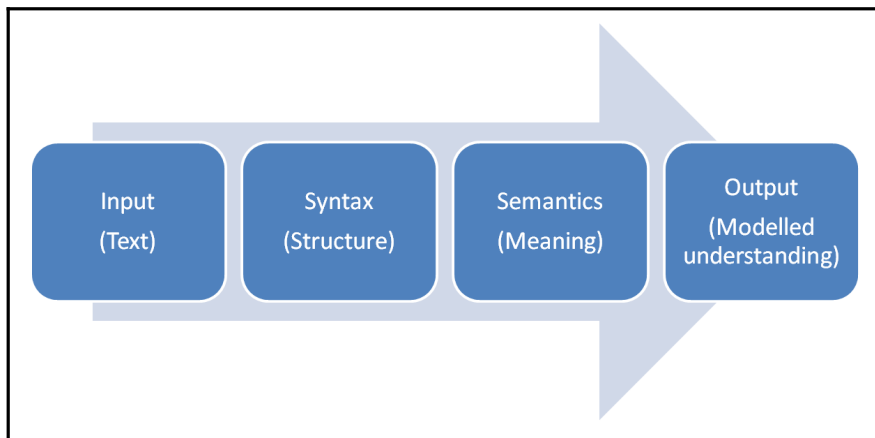
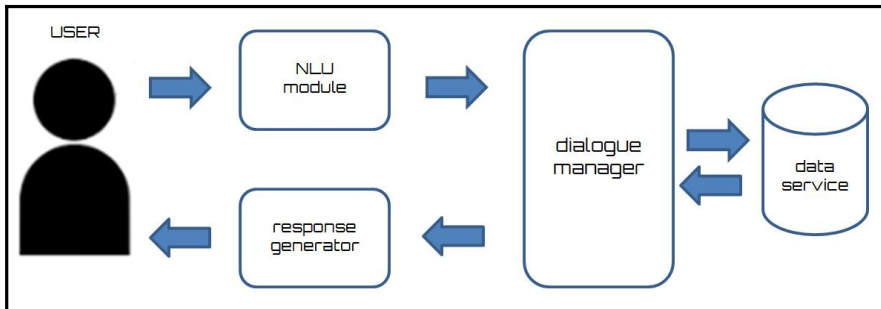
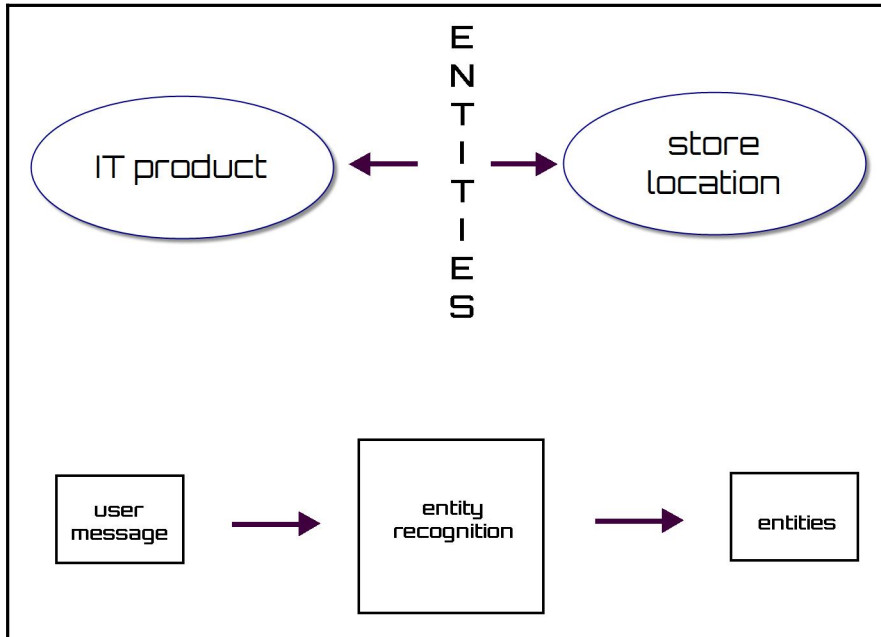
```
Sound Blaster                D R S B A I T S O                version 2.20
(c) Copyright Creative Labs, Inc. 1992, all rights reserved


Please enter your name ...Dave
HELLO DAVE, MY NAME IS DOCTOR SBAITSO.


I AM HERE TO HELP YOU.
SAY WHATEVER IS IN YOUR MIND FREELY,
OUR CONVERSATION WILL BE KEPT IN STRICT CONFIDENCE.
MEMORY CONTENTS WILL BE WIPED OFF AFTER YOU LEAVE,

SO, TELL ME ABOUT YOUR PROBLEMS.
>
```

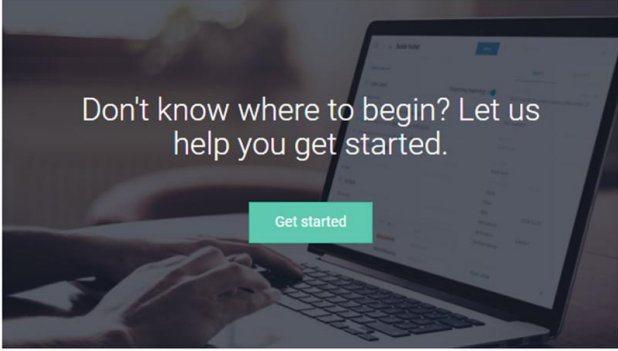





Dialogflow


Welcome to Dialogflow!

- + Create Agent
- > Docs
- > Forum
- ? Support
- Account
- Logout




Don't know where to begin? Let us help you get started.


Get started

Now it's time to create your first agent.

CREATE AGENT



Please, create at least one agent to access the test console


Dialogflow

Agent name
CREATE

- + Create Agent
- > Docs
- > Forum
- ? Support
- Account
- Logout

DEFAULT LANGUAGE ?

English – en ▼

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+2:00) Europe/Kaliningrad ▼

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Create a new Google project ▼

Enables Cloud functions, Actions on Google and permissions management.

API VERSION

Dialogflow V2 API [beta]

Use [Dialogflow V2 API](#) as default for the agent. Your webhook will receive [V2 format requests](#) and should return [V2 format responses](#).

Dialogflow

Intent name
SAVE
⋮

Try it now

RomeWeather ⚙️

en
+

Intents
+

Entities
+

Training [beta]
+

Integrations
+

Analytics [new]
+

Fulfillment
+

Prebuilt Agents
+

Small Talk
+

> Docs

> Forum

Contexts ?
▼

Events ?
▼

Training phrases ?
▲

Train the intent with what your users will say

Provide examples of how users will express their intent in natural language. Adding numerous phrases with different variations and parameters will improve the accuracy of intent matching. [Learn more](#)

[ADD TRAINING PHRASES](#)

Action and parameters ?
▲

Extract the action and parameters

ⓘ Please use test console above to try a sentence.

▶ See how it works in [Google Assistant](#).

Training phrases ?

Search training
🔍
▲

” Add user expression

” What's the weather like in Rome

PARAMETER NAME	ENTITY	RESOLVED VALUE	
geo-city	@sys.geo-city	Rome	✕


” Weather forecast in Rome today

PARAMETER NAME	ENTITY	RESOLVED VALUE	
geo-city	@sys.geo-city	Rome	✕
date	@sys.date	today	✕

AgentDomains

USER SAYS COPY CURL

How's the weather in Rome today

 DEFAULT RESPONSE PLAY

The weather forecast for 2018-04-06 in Rome is not available

INTENT

[WheatherLocation](#)

ACTION

Not available

PARAMETER	VALUE
geo-city	Rome
date	2018-04-06
