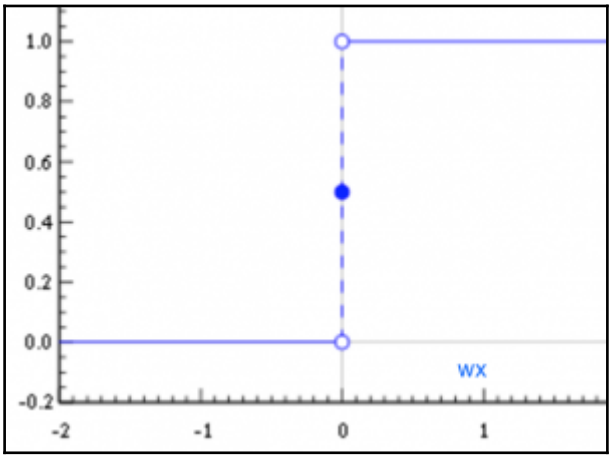
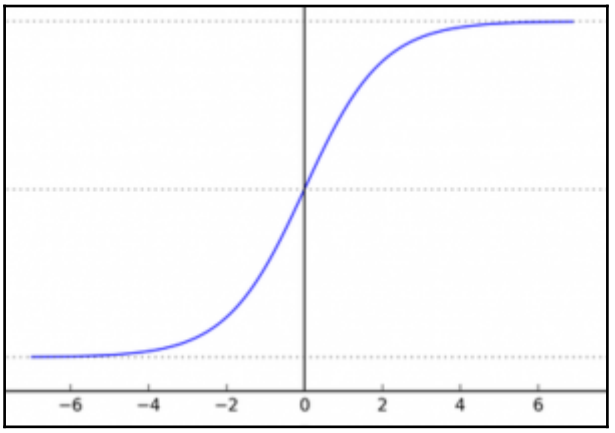
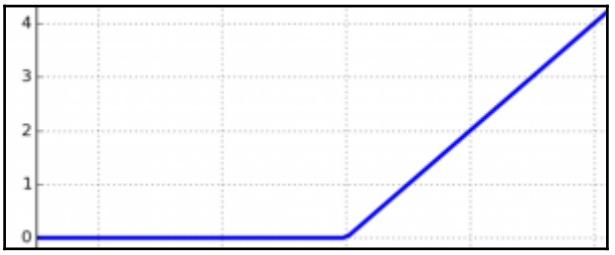
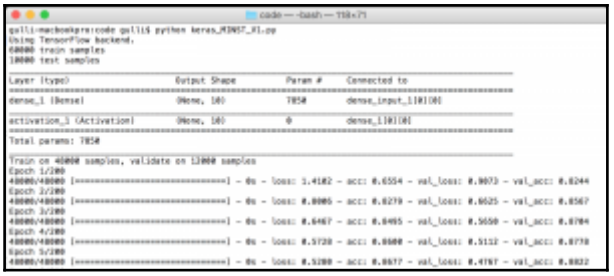
Chapter 1: Neural Networks Foundations

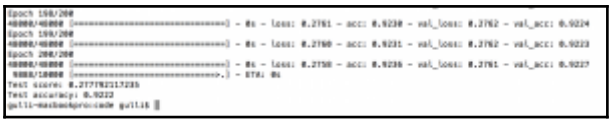




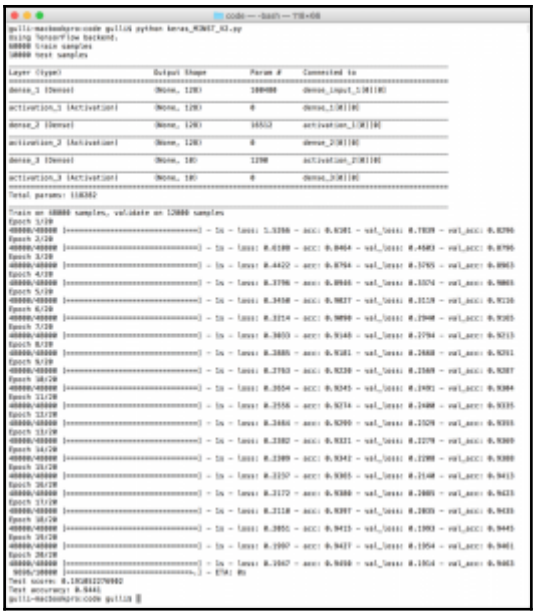




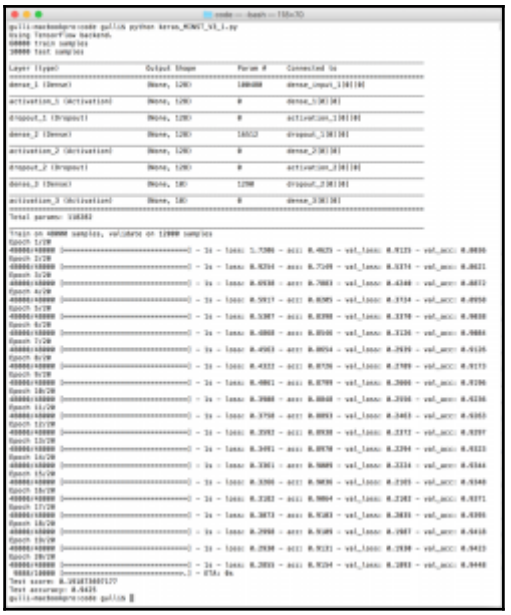
```
guili-macbookpro:code guili$ python keras_MINIST_V1.py
Using TensorFlow backend.
40000 train samples
10000 test samples
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                7850
_________________________________________________________________
activation_1 (Activation)    (None, 10)                0
=================================================================
Total params: 7850
_________________________________________________________________
Train on 40000 samples, validate on 12000 samples
Epoch 1/200
40000/40000 [==============================] - 0s - loss: 1.4182 - acc: 0.6554 - val_loss: 0.9073 - val_acc: 0.8244
Epoch 2/200
40000/40000 [==============================] - 0s - loss: 0.8005 - acc: 0.8279 - val_loss: 0.6525 - val_acc: 0.8567
Epoch 3/200
40000/40000 [==============================] - 0s - loss: 0.6467 - acc: 0.8495 - val_loss: 0.5650 - val_acc: 0.8704
Epoch 4/200
40000/40000 [==============================] - 0s - loss: 0.5720 - acc: 0.8600 - val_loss: 0.5132 - val_acc: 0.8778
Epoch 5/200
40000/40000 [==============================] - 0s - loss: 0.5280 - acc: 0.8677 - val_loss: 0.4787 - val_acc: 0.8832
```



```
Epoch 199/200
40000/40000 [==============================] - 0s - loss: 0.2761 - acc: 0.9230 - val_loss: 0.2762 - val_acc: 0.9224
Epoch 199/200
40000/40000 [==============================] - 0s - loss: 0.2760 - acc: 0.9231 - val_loss: 0.2762 - val_acc: 0.9223
Epoch 200/200
40000/40000 [==============================] - 0s - loss: 0.2758 - acc: 0.9236 - val_loss: 0.2761 - val_acc: 0.9227
 9888/10000 [============================>.] - ETA: 0s
Test score: 0.27717821172351
Test accuracy: 0.9223
guili-macbookpro:code guili$
```

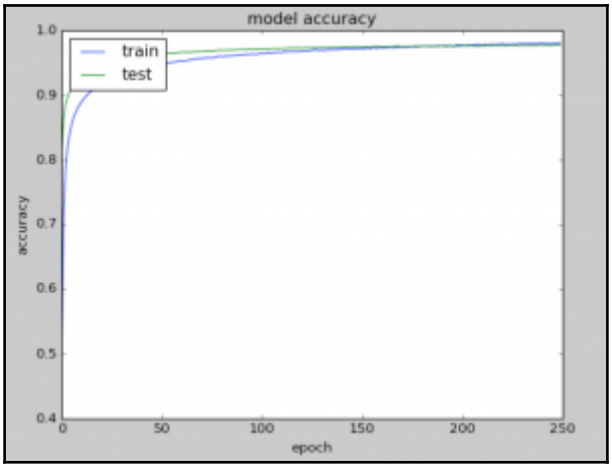


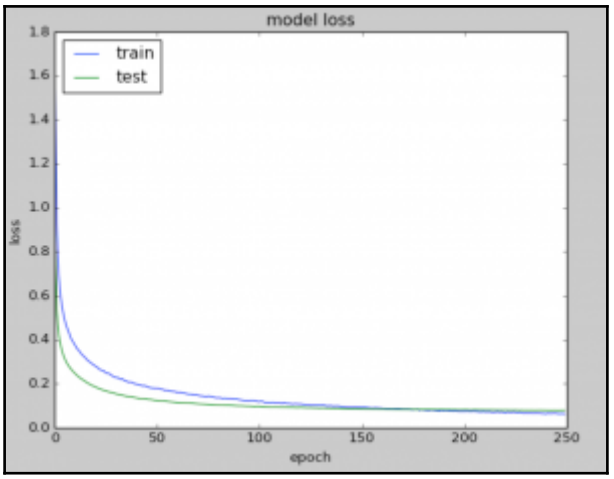
```
guili-macbookpro:code guili$ python keras_MINIST_V2.py
Using TensorFlow backend.
60000 train samples
10000 test samples
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               100480
_________________________________________________________________
activation_1 (Activation)    (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               16512
_________________________________________________________________
activation_2 (Activation)    (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290
_________________________________________________________________
activation_3 (Activation)    (None, 10)                0
=================================================================
Total params: 118,282
_________________________________________________________________
Train on 48000 samples, validate on 12000 samples
Epoch 1/20
48000/48000 [==============================] - 1s - loss: 1.3286 - acc: 0.6301 - val_loss: 0.7839 - val_acc: 0.8296
Epoch 2/20
48000/48000 [==============================] - 1s - loss: 0.6180 - acc: 0.8464 - val_loss: 0.4683 - val_acc: 0.8795
Epoch 3/20
48000/48000 [==============================] - 1s - loss: 0.4422 - acc: 0.8764 - val_loss: 0.3765 - val_acc: 0.8963
Epoch 4/20
48000/48000 [==============================] - 1s - loss: 0.3756 - acc: 0.8946 - val_loss: 0.3374 - val_acc: 0.9061
Epoch 5/20
48000/48000 [==============================] - 1s - loss: 0.3458 - acc: 0.9027 - val_loss: 0.3128 - val_acc: 0.9116
Epoch 6/20
48000/48000 [==============================] - 1s - loss: 0.3214 - acc: 0.9090 - val_loss: 0.2948 - val_acc: 0.9165
Epoch 7/20
48000/48000 [==============================] - 1s - loss: 0.3053 - acc: 0.9140 - val_loss: 0.2794 - val_acc: 0.9213
Epoch 8/20
48000/48000 [==============================] - 1s - loss: 0.2885 - acc: 0.9181 - val_loss: 0.2668 - val_acc: 0.9251
Epoch 9/20
48000/48000 [==============================] - 1s - loss: 0.2763 - acc: 0.9220 - val_loss: 0.2569 - val_acc: 0.9287
Epoch 10/20
48000/48000 [==============================] - 1s - loss: 0.2654 - acc: 0.9245 - val_loss: 0.2491 - val_acc: 0.9304
Epoch 11/20
48000/48000 [==============================] - 1s - loss: 0.2556 - acc: 0.9274 - val_loss: 0.2408 - val_acc: 0.9335
Epoch 12/20
48000/48000 [==============================] - 1s - loss: 0.2466 - acc: 0.9290 - val_loss: 0.2329 - val_acc: 0.9355
Epoch 13/20
48000/48000 [==============================] - 1s - loss: 0.2382 - acc: 0.9321 - val_loss: 0.2279 - val_acc: 0.9369
Epoch 14/20
48000/48000 [==============================] - 1s - loss: 0.2309 - acc: 0.9342 - val_loss: 0.2208 - val_acc: 0.9380
Epoch 15/20
48000/48000 [==============================] - 1s - loss: 0.2237 - acc: 0.9365 - val_loss: 0.2148 - val_acc: 0.9413
Epoch 16/20
48000/48000 [==============================] - 1s - loss: 0.2172 - acc: 0.9380 - val_loss: 0.2085 - val_acc: 0.9433
Epoch 17/20
48000/48000 [==============================] - 1s - loss: 0.2110 - acc: 0.9397 - val_loss: 0.2035 - val_acc: 0.9435
Epoch 18/20
48000/48000 [==============================] - 1s - loss: 0.2055 - acc: 0.9415 - val_loss: 0.1993 - val_acc: 0.9445
Epoch 19/20
48000/48000 [==============================] - 1s - loss: 0.1997 - acc: 0.9427 - val_loss: 0.1954 - val_acc: 0.9461
Epoch 20/20
48000/48000 [==============================] - 1s - loss: 0.1947 - acc: 0.9450 - val_loss: 0.1914 - val_acc: 0.9463
 9536/10000 [===========================>..] - ETA: 0s
Test score: 0.19148127979
Test accuracy: 0.9441
guili-macbookpro:code guili$
```

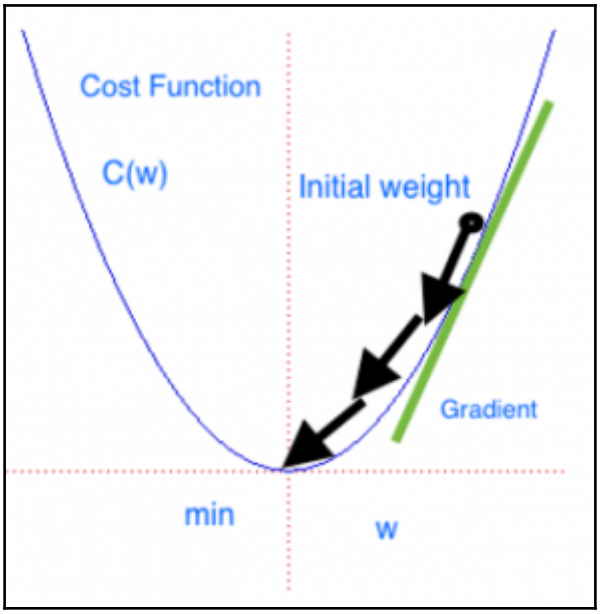


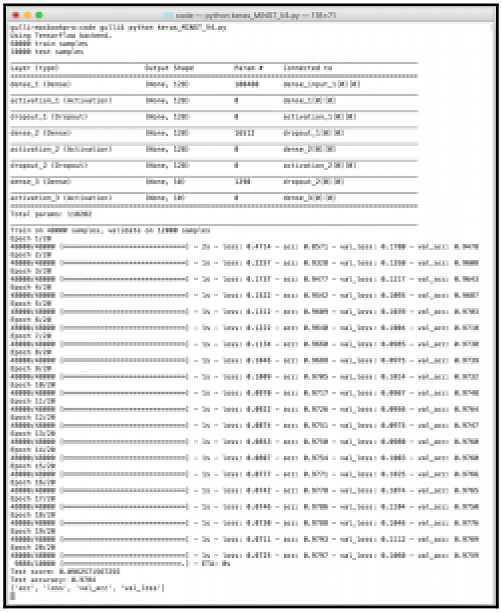
```
guili-macbookpro:code guili$ python keras_MNIST_v1_1.py
Using TensorFlow backend.
60000 train samples
10000 test samples
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               100480
_________________________________________________________________
activation_1 (Activation)    (None, 128)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               16512
_________________________________________________________________
activation_2 (Activation)    (None, 128)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290
_________________________________________________________________
activation_3 (Activation)    (None, 10)                0
=================================================================
Total params: 118,282
_________________________________________________________________
Train on 48000 samples, validate on 12000 samples
Epoch 1/20
48000/48000 [==============================] - 1s - loss: 1.7386 - acc: 0.4623 - val_loss: 0.9123 - val_acc: 0.8036
Epoch 2/20
48000/48000 [==============================] - 1s - loss: 0.9294 - acc: 0.7249 - val_loss: 0.5374 - val_acc: 0.8621
Epoch 3/20
48000/48000 [==============================] - 1s - loss: 0.6938 - acc: 0.7883 - val_loss: 0.4348 - val_acc: 0.8873
Epoch 4/20
48000/48000 [==============================] - 1s - loss: 0.5917 - acc: 0.8205 - val_loss: 0.3754 - val_acc: 0.8950
Epoch 5/20
48000/48000 [==============================] - 1s - loss: 0.5307 - acc: 0.8398 - val_loss: 0.3378 - val_acc: 0.9038
Epoch 6/20
48000/48000 [==============================] - 1s - loss: 0.4868 - acc: 0.8548 - val_loss: 0.3126 - val_acc: 0.9084
Epoch 7/20
48000/48000 [==============================] - 1s - loss: 0.4563 - acc: 0.8654 - val_loss: 0.2939 - val_acc: 0.9126
Epoch 8/20
48000/48000 [==============================] - 1s - loss: 0.4333 - acc: 0.8726 - val_loss: 0.2789 - val_acc: 0.9173
Epoch 9/20
48000/48000 [==============================] - 1s - loss: 0.4061 - acc: 0.8799 - val_loss: 0.2666 - val_acc: 0.9196
Epoch 10/20
48000/48000 [==============================] - 1s - loss: 0.3900 - acc: 0.8848 - val_loss: 0.2556 - val_acc: 0.9236
Epoch 11/20
48000/48000 [==============================] - 1s - loss: 0.3750 - acc: 0.8893 - val_loss: 0.2463 - val_acc: 0.9263
Epoch 12/20
48000/48000 [==============================] - 1s - loss: 0.3593 - acc: 0.8938 - val_loss: 0.2371 - val_acc: 0.9297
Epoch 13/20
48000/48000 [==============================] - 1s - loss: 0.3491 - acc: 0.8978 - val_loss: 0.2294 - val_acc: 0.9323
Epoch 14/20
48000/48000 [==============================] - 1s - loss: 0.3361 - acc: 0.9005 - val_loss: 0.2234 - val_acc: 0.9344
Epoch 15/20
48000/48000 [==============================] - 1s - loss: 0.3266 - acc: 0.9036 - val_loss: 0.2165 - val_acc: 0.9360
Epoch 16/20
48000/48000 [==============================] - 1s - loss: 0.3183 - acc: 0.9064 - val_loss: 0.2102 - val_acc: 0.9371
Epoch 17/20
48000/48000 [==============================] - 1s - loss: 0.3073 - acc: 0.9103 - val_loss: 0.2031 - val_acc: 0.9391
Epoch 18/20
48000/48000 [==============================] - 1s - loss: 0.2998 - acc: 0.9105 - val_loss: 0.1987 - val_acc: 0.9410
Epoch 19/20
48000/48000 [==============================] - 1s - loss: 0.2930 - acc: 0.9131 - val_loss: 0.1930 - val_acc: 0.9423
Epoch 20/20
48000/48000 [==============================] - 1s - loss: 0.2853 - acc: 0.9154 - val_loss: 0.1893 - val_acc: 0.9440
 9888/10000 [============================>.] - ETA: 0s
Test score: 0.18187860127
Test accuracy: 0.9426
guili-macbookpro:code guili$
```

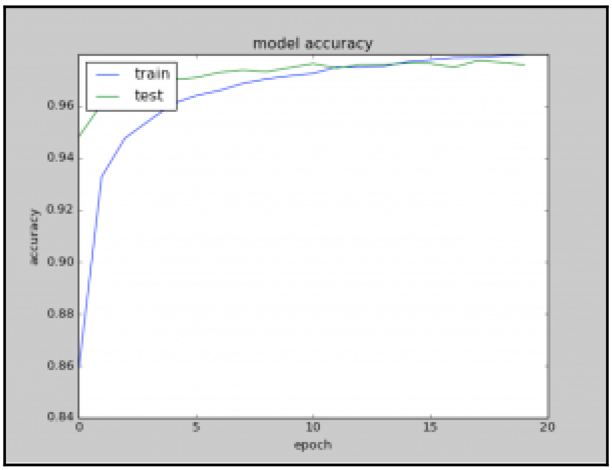
```
Epoch 248/250
48000/48000 [==============================] - 1s - loss: 0.0630 - acc: 0.9804 - val_loss: 0.0785 - val_acc: 0.9768
Epoch 249/250
48000/48000 [==============================] - 1s - loss: 0.0634 - acc: 0.9799 - val_loss: 0.0789 - val_acc: 0.9775
Epoch 250/250
48000/48000 [==============================] - 1s - loss: 0.0636 - acc: 0.9818 - val_loss: 0.0787 - val_acc: 0.9773
 9856/10000 [============================>.] - ETA: 0s
Test score: 0.0718828012328
Test accuracy: 0.9777
guili-macbookpro:code guili$
```

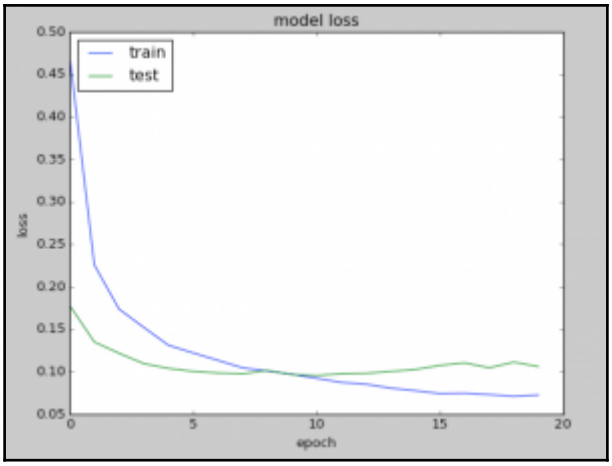


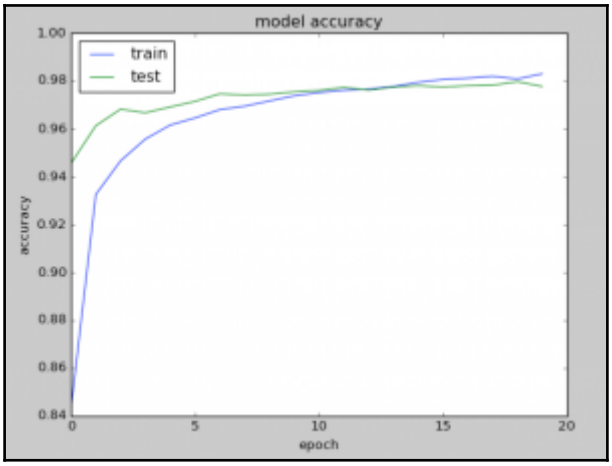

model loss
train
test
loss
epoch
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250

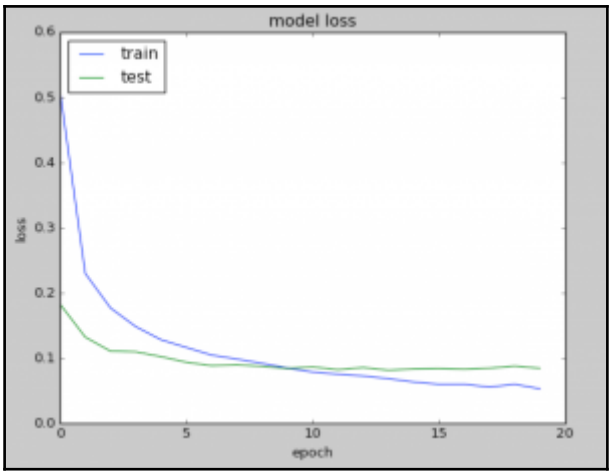

Cost Function
C(w)
Initial weight
Gradient
min
w

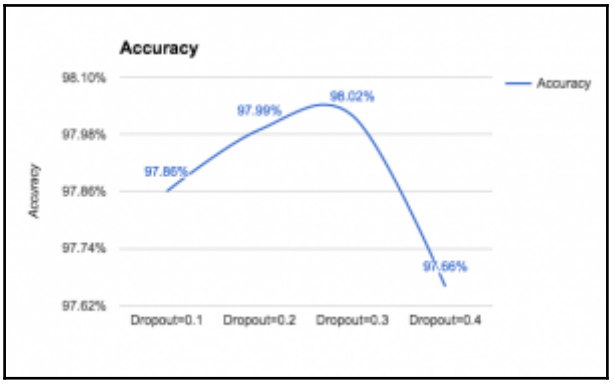



model accuracy
train
test
accuracy
epoch
0.84
0.86
0.88
0.90
0.92
0.94
0.96
0
5
10
15
20


model loss
train
test
loss
epoch
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
5
10
15
20

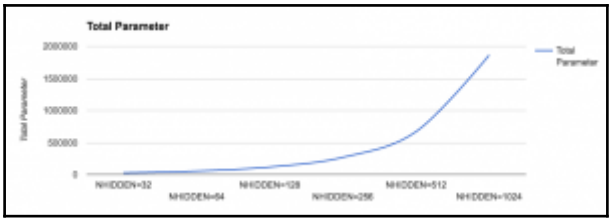

model accuracy
train
test
accuracy
epoch
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0
5
10
15
20


model loss
train
test
loss
epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
5
10
15
20

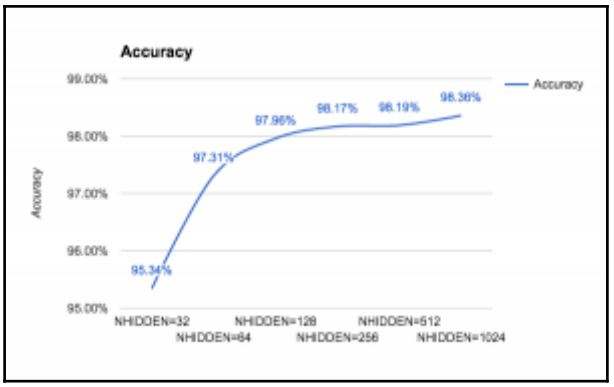

Accuracy
Accuracy
98.10%
97.98%
97.86%
97.74%
97.62%
97.86%
97.99%
98.02%
97.66%
Dropout=0.1
Dropout=0.2
Dropout=0.3
Dropout=0.4

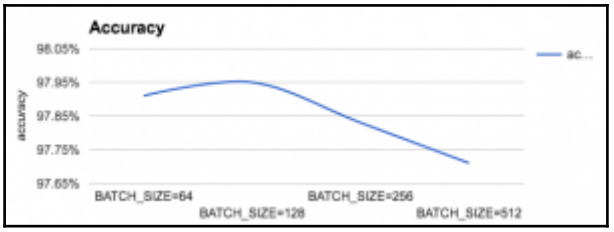

Accuracy
Accuracy
100.00%
98.00%
96.00%
94.00%
92.00%
92.22%
94.41%
97.70%
97.84%
97.93%
97.91%
MNIST_V1
MNIST_2HID
MNIST_DRP
MNIST_RMS
MNIST_ADM
MNIST_EPC

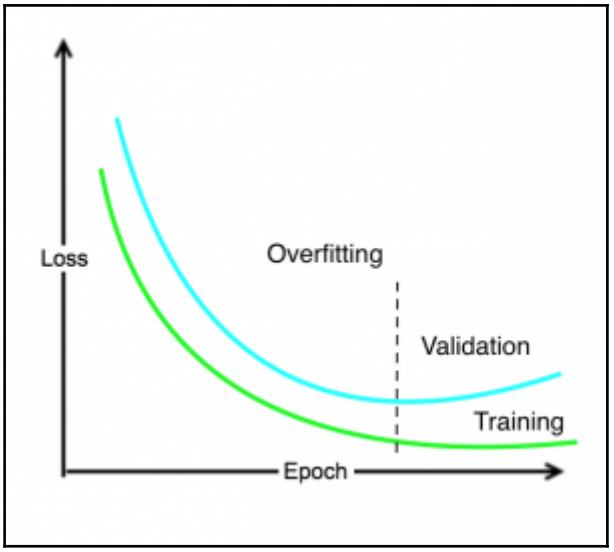

Accuracy
120.00%
90.00%
60.00%
30.00%
0.00%
Accuracy
10.28%
96.26%
97.93%
96.48%
MINST_LR=0.1
MINST_LR=0.01
MINST_LR=0.001
MINST_LR=0.0001
Accuracy

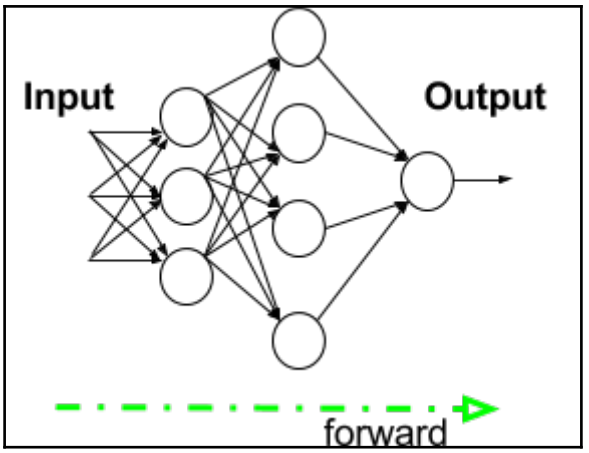

Total Parameter
Total Parameter
NHIDDEN=32
NHIDDEN=64
NHIDDEN=128
NHIDDEN=256
NHIDDEN=512
NHIDDEN=1024

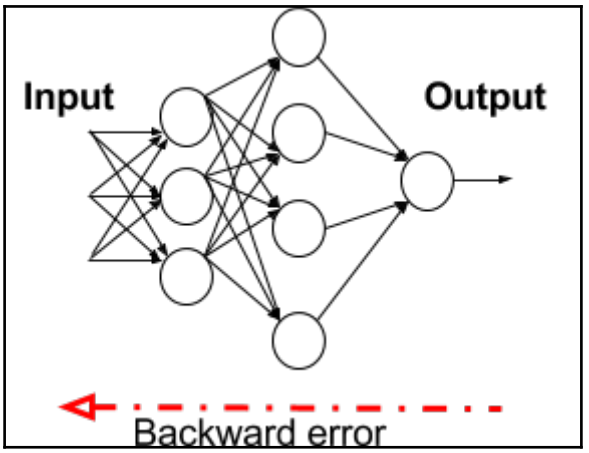

ExecutionTimePerIteration
16
12
8
4
0
ExecutionTimePerIteration
1
1.2
1.5
2.5
4.5
13
NHIDDEN=32
NHIDDEN=64
NHIDDEN=128
NHIDDEN=256
NHIDDEN=512
NHIDDEN=1024
Executi...


Accuracy
99.00%
98.00%
97.00%
96.00%
95.00%
Accuracy
95.34%
97.31%
97.96%
98.17%
98.19%
98.38%
NHIDDEN=32
NHIDDEN=64
NHIDDEN=128
NHIDDEN=256
NHIDDEN=512
NHIDDEN=1024
Accuracy

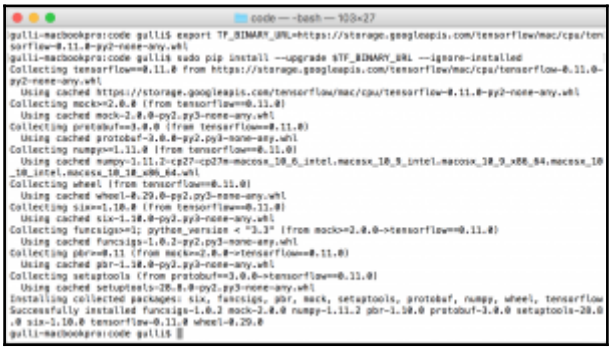

Accuracy
98.05%
97.95%
97.85%
97.75%
97.65%
accuracy
BATCH_SIZE=64
BATCH_SIZE=128
BATCH_SIZE=256
BATCH_SIZE=512
ac...

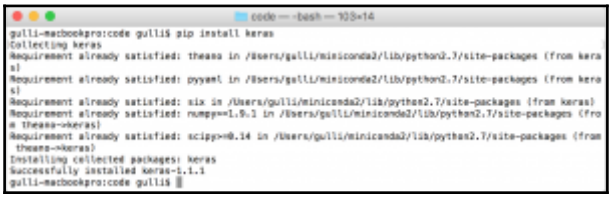

Loss
Overfitting
Validation
Training
Epoch

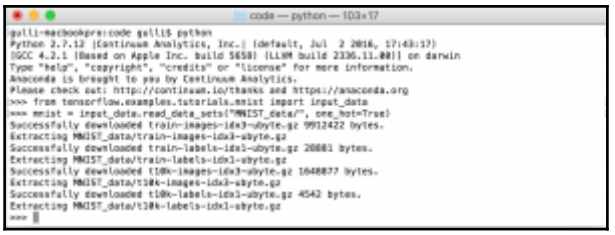

Input
Output
forward


Input
Output
Backward error

Chapter 2: Keras Installation and API





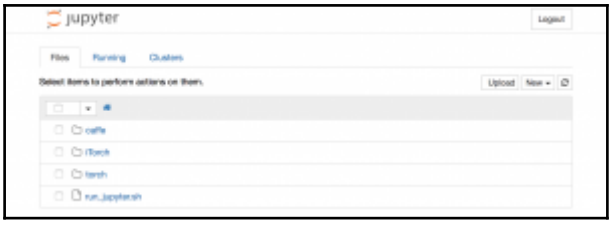



```
gulli-macbookpro:dl-docker gulli$ git clone https://github.com/saiprashanths/dl-docker
.git
Cloning into 'dl-docker'...
remote: Counting objects: 89, done.
remote: Total 89 (delta 0), reused 0 (delta 0), pack-reused 89
Unpacking objects: 100% (89/89), done.
gulli-macbookpro:dl-docker gulli$
```

```
gulli-macbookpro:dl-docker gulli$ cd dl-docker/
gulli-macbookpro:dl-docker gulli$ docker build -t floydhub/dl-docker:cpu -f Dockerfile
.cpu .
Sending build context to Docker daemon 284.2 kB
Step 1 : FROM ubuntu:14.04
 ---> 3f755ca42730
Step 2 : MAINTAINER Sai Soundararaj <saip@outlook.com>
 ---> Using cache
 ---> af82b42b4e1c
Step 3 : ARG TH_VERSION=rel-0.8.2
 ---> Using cache
 ---> c8d03ba70cff
Step 4 : ARG TENSORFLOW_VERSION=0.8.0
 ---> Using cache
 ---> de0ed51e5732
Step 5 : ARG TENSORFLOW_ARCH=cpu
 ---> Using cache
 ---> 270d4bfbccaa
Step 6 : ARG KERAS_VERSION=1.0.3
 ---> Using cache
 ---> 61219a95474f
Step 7 : ARG LASAGNE_VERSION=v0.1
 ---> Using cache
 ---> 585e125f1e76
Step 8 : ARG TORCH_VERSION=latest
 ---> Using cache
 ---> fa5c4246c2ec
Step 9 : ARG CAFFE_VERSION=master
 ---> Using cache
 ---> 989ad8491f04
Step 10 : RUN apt-get update && apt-get install -y              bc              build-
```

```
gulli-macbookpro:dl-docker gulli$ docker run -it -p 8888:8888 -p 6006:6006 floydhub/dl
-docker:cpu bash
root@788e0d54bfc0:~# ls
caffe  iTorch  run_jupyter.sh  torch
root@788e0d54bfc0:~#
```

```
root@780e0d54bfc0:~# sh run_jupyter.sh
[I 10:51:17.489 NotebookApp] Copying /root/.ipython/kernels -> /root/.local/share/jupy
ter/kernels
[I 10:51:17.498 NotebookApp] Writing notebook server cookie secret to /root/.local/sha
re/jupyter/runtime/notebook_cookie_secret
[W 10:51:17.520 NotebookApp] WARNING: The notebook server is listening on all IP addre
sses and not using encryption. This is not recommended.
[I 10:51:17.536 NotebookApp] Serving notebooks from local directory: /root
[I 10:51:17.536 NotebookApp] 0 active kernels
[I 10:51:17.537 NotebookApp] The Jupyter Notebook is running at: http://[all ip addres
ses on your system]:8888/?token=503b59dc969d43f588638e3bd153dd1525837ff46d7b1eb9
[I 10:51:17.537 NotebookApp] Use Control-C to stop this server and shut down all kerne
ls (twice to skip confirmation).
[C 10:51:17.539 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=503b59dc969d43f588638e3bd153dd1525837ff46d7b1eb9
[I 10:51:32.547 NotebookApp] 302 GET / (172.17.0.1) 0.60ms
[I 10:51:32.553 NotebookApp] 302 GET /tree? (172.17.0.1) 0.86ms
[I 10:51:40.287 NotebookApp] 302 GET /?token=503b59dc969d43f588638e3bd153dd1525837ff46
d7b1eb9 (172.17.0.1) 0.96ms
```

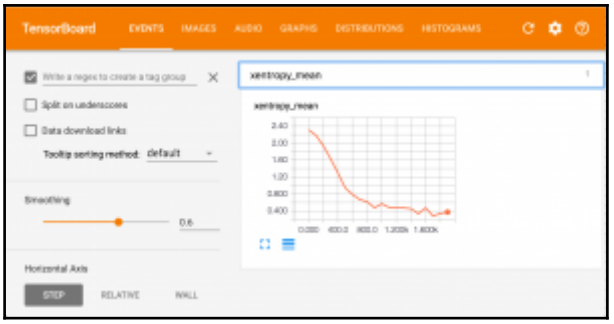


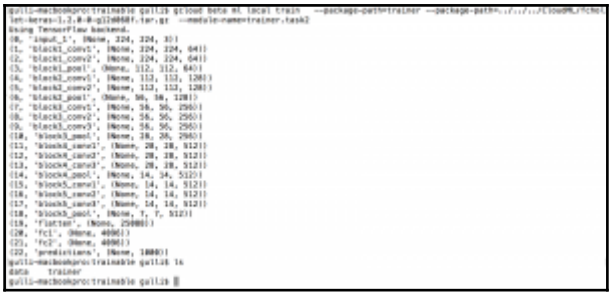

```
root@7b599d8dcaeb:~# tensorboard --logdir .
```

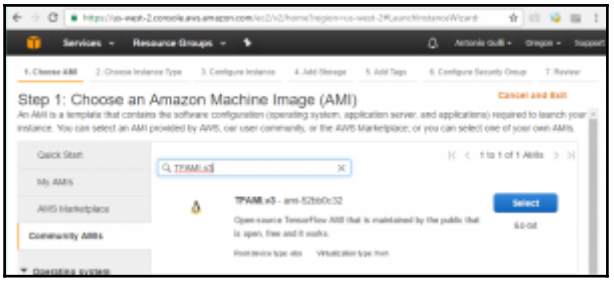

```
gulli-macbookpro:google-cloud-sdk gulli$ git clone https://github.com/GoogleCloudPlatform/cloudml-samples/
Cloning into 'cloudml-samples'...
remote: Counting objects: 118, done.
remote: Total 118 (delta 0), reused 0 (delta 0), pack-reused 118
Receiving objects: 100% (118/118), 84.40 KiB | 0 bytes/s, done.
Resolving deltas: 100% (48/48), done.
gulli-macbookpro:google-cloud-sdk gulli$
```

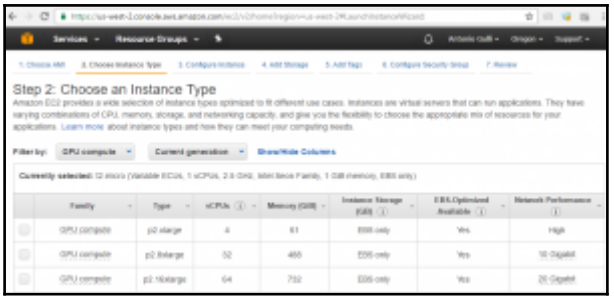
```
gulli-macbookpro:code@book gulli$ cd cloudml-samples/mnist/trainable/
gulli-macbookpro:trainable gulli$ ls
trainer
gulli-macbookpro:trainable gulli$ gcloud beta ml local train  --package-path=trainer  --module-name=trainer.task
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
Extracting /var/folders/dx/s5b4dlj92sz_sls8tf35mjr00cm8l/T/tmpcWRkfj/train-images-idx3-ubyte.gz
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
Extracting /var/folders/dx/s5b4dlj92sz_sls8tf35mjr00cm8l/T/tmpcWRkfj/train-labels-idx1-ubyte.gz
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting /var/folders/dx/s5b4dlj92sz_sls8tf35mjr00cm8l/T/tmpcWRkfj/t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting /var/folders/dx/s5b4dlj92sz_sls8tf35mjr00cm8l/T/tmpcWRkfj/t10k-labels-idx1-ubyte.gz
Step 0: loss = 2.32 (0.010 sec)
Step 100: loss = 2.19 (0.002 sec)
Step 200: loss = 1.96 (0.002 sec)
Step 300: loss = 1.64 (0.002 sec)
Step 400: loss = 1.30 (0.002 sec)
Step 500: loss = 0.95 (0.002 sec)
Step 600: loss = 0.80 (0.002 sec)
Step 700: loss = 0.67 (0.002 sec)
Step 800: loss = 0.62 (0.002 sec)
Step 900: loss = 0.48 (0.002 sec)
Training Data Eval:
  Num examples: 55000  Num correct: 47295  Precision @ 1: 0.8599
Validation Data Eval:
  Num examples: 5000  Num correct: 4347  Precision @ 1: 0.8694
Test Data Eval:
  Num examples: 10000  Num correct: 8649  Precision @ 1: 0.8649
Step 1000: loss = 0.58 (0.010 sec)
Step 1100: loss = 0.49 (0.115 sec)
Step 1200: loss = 0.49 (0.002 sec)
Step 1300: loss = 0.48 (0.002 sec)
Step 1400: loss = 0.48 (0.002 sec)
Step 1500: loss = 0.36 (0.002 sec)
Step 1600: loss = 0.49 (0.002 sec)
Step 1700: loss = 0.29 (0.002 sec)
Step 1800: loss = 0.35 (0.002 sec)
Step 1900: loss = 0.39 (0.002 sec)
Training Data Eval:
  Num examples: 55000  Num correct: 49363  Precision @ 1: 0.8975
Validation Data Eval:
  Num examples: 5000  Num correct: 4519  Precision @ 1: 0.9038
Test Data Eval:
  Num examples: 10000  Num correct: 9000  Precision @ 1: 0.9000
gulli-macbookpro:trainable gulli$
```

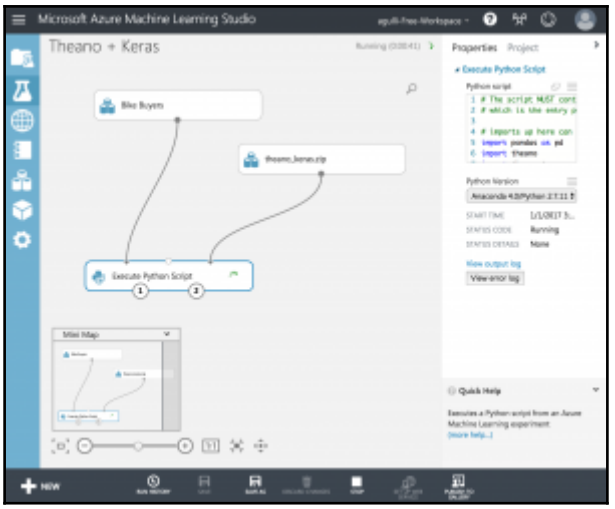
```
gulli-macbookpro:trainable gulli$ tensorboard --logdir=data/ --port=8888
Starting TensorBoard 29 on port 8888
```

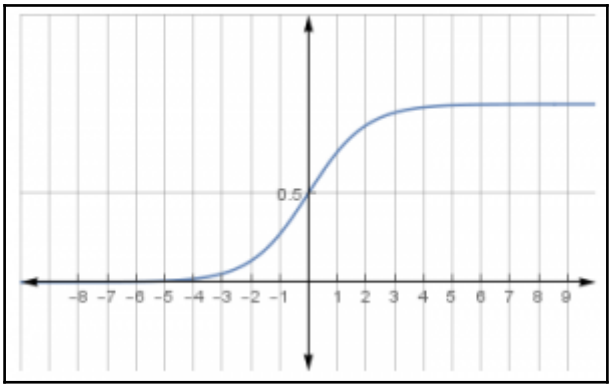

TensorBoard
EVENTS
IMAGES
AUDIO
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
Write a regex to create a tag group
Split on underscores
Data download links
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
xentropy_mean
xentropy_mean

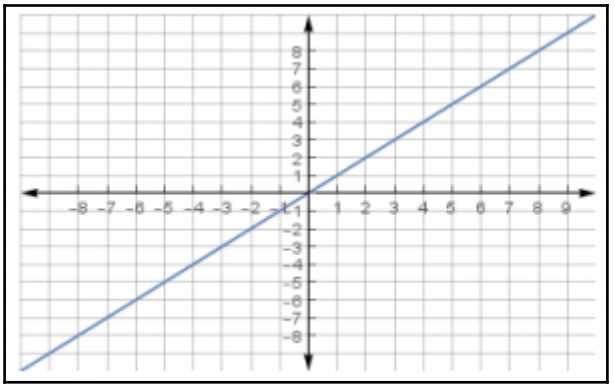

(0, 'input_1', (None, 224, 224, 3))
(1, 'block1_conv1', (None, 224, 224, 64))
(2, 'block1_conv2', (None, 224, 224, 64))
(3, 'block1_pool', (None, 112, 112, 64))
(4, 'block2_conv1', (None, 112, 112, 128))
(5, 'block2_conv2', (None, 112, 112, 128))
(6, 'block2_pool', (None, 56, 56, 128))
(7, 'block3_conv1', (None, 56, 56, 256))
(8, 'block3_conv2', (None, 56, 56, 256))
(9, 'block3_conv3', (None, 56, 56, 256))
(10, 'block3_pool', (None, 28, 28, 256))
(11, 'block4_conv1', (None, 28, 28, 512))
(12, 'block4_conv2', (None, 28, 28, 512))
(13, 'block4_conv3', (None, 28, 28, 512))
(14, 'block4_pool', (None, 14, 14, 512))
(15, 'block5_conv1', (None, 14, 14, 512))
(16, 'block5_conv2', (None, 14, 14, 512))
(17, 'block5_conv3', (None, 14, 14, 512))
(18, 'block5_pool', (None, 7, 7, 512))
(19, 'flatten', (None, 25088))
(20, 'fc1', (None, 4096))
(21, 'fc2', (None, 4096))
(22, 'predictions', (None, 1000))
data trainer

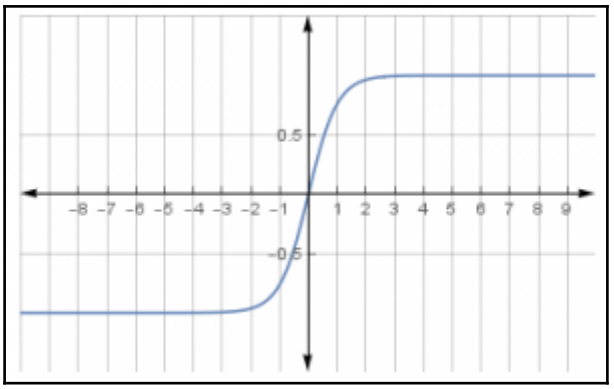

Step 1: Choose an Amazon Machine Image (AMI)
Quick Start
My AMIs
AWS Marketplace
Community AMIs
Select

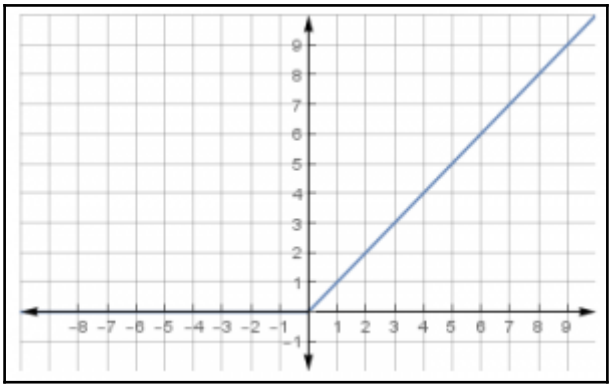

Step 2: Choose an Instance Type
Family
Type
vCPUs
Memory (GiB)
GPU compute
p2.xlarge
p2.8xlarge
p2.16xlarge

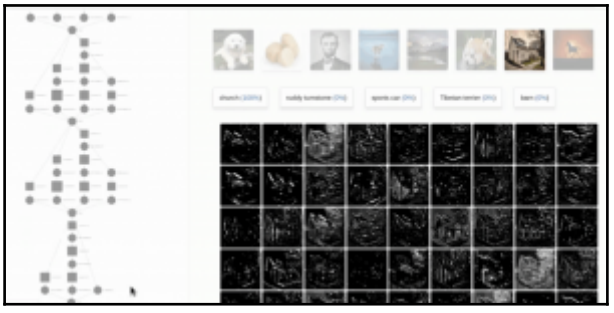

Microsoft Azure Machine Learning Studio
Theano + Keras
Bike Buyers
theano_keras.zip
Execute Python Script
Mini Map
Properties
Project
Execute Python Script
Python script
Python Version
Quick Help
NEW


0.5
-8 -7 -6 -5 -4 -3 -2 -1 1 2 3 4 5 6 7 8 9

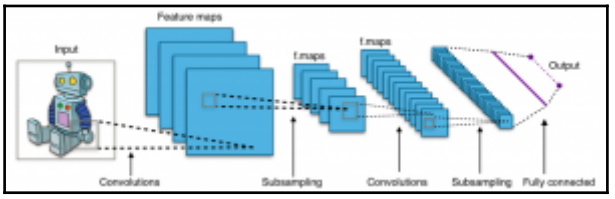

8 7 6 5 4 3 2 1
-8 -7 -6 -5 -4 -3 -2 -1 1 2 3 4 5 6 7 8 9
-1 -2 -3 -4 -5 -6 -7 -8

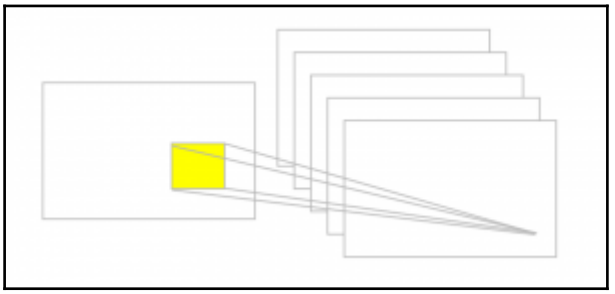

0.5
-0.5
-8 -7 -6 -5 -4 -3 -2 -1 1 2 3 4 5 6 7 8 9

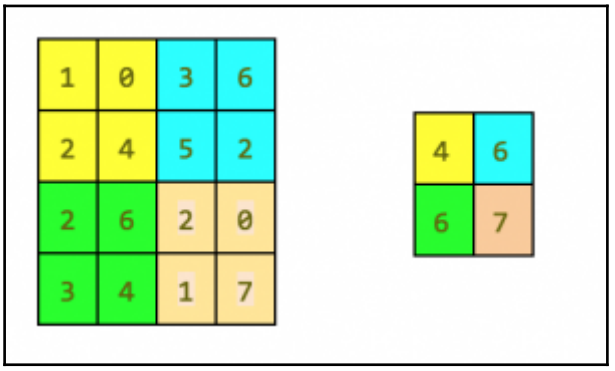

9
8
7
6
5
4
3
2
1
-1
-8 -7 -6 -5 -4 -3 -2 -1 1 2 3 4 5 6 7 8 9

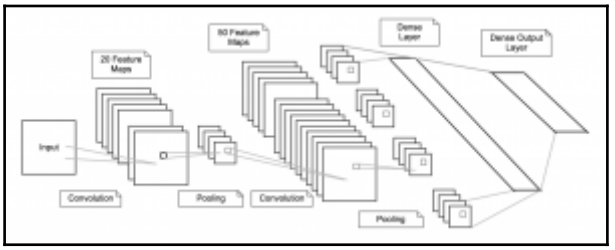


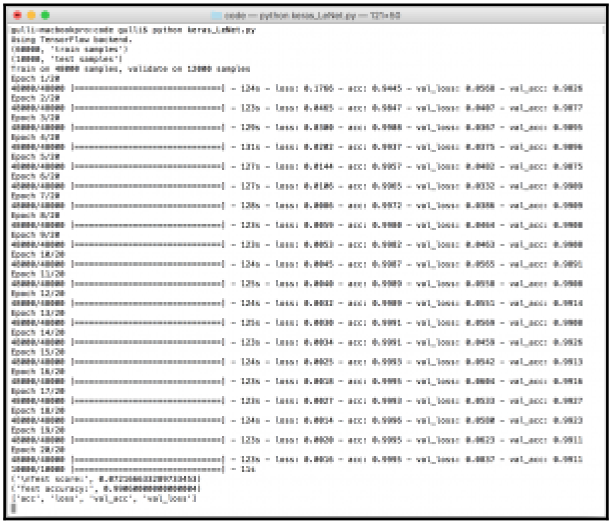
Chapter 3: Deep Learning with ConvNets

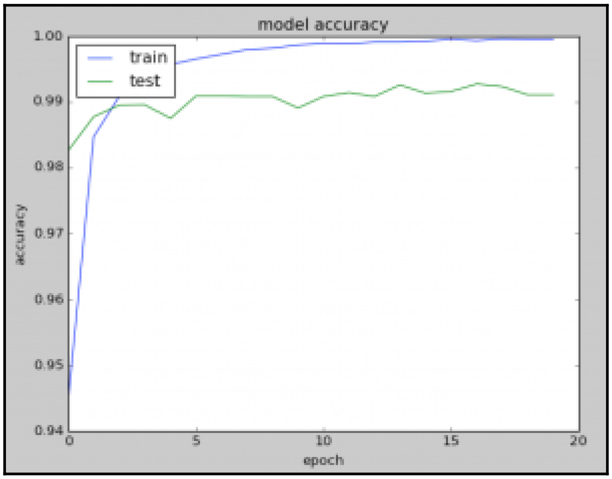


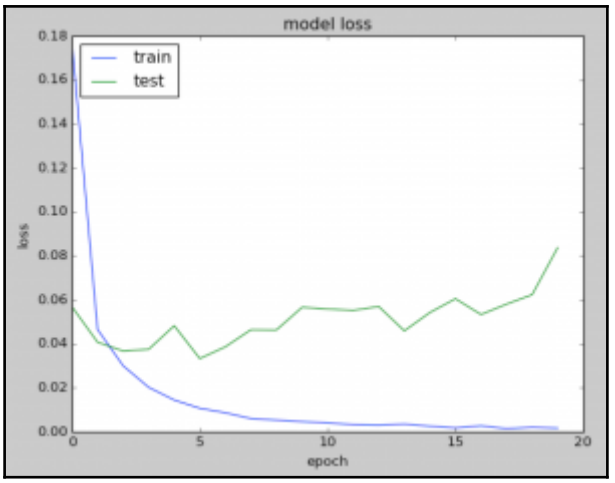


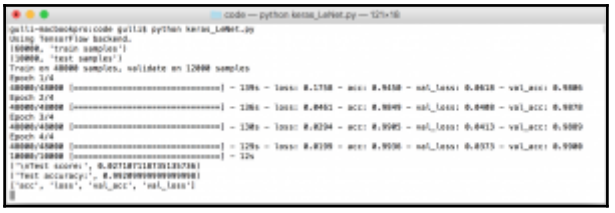


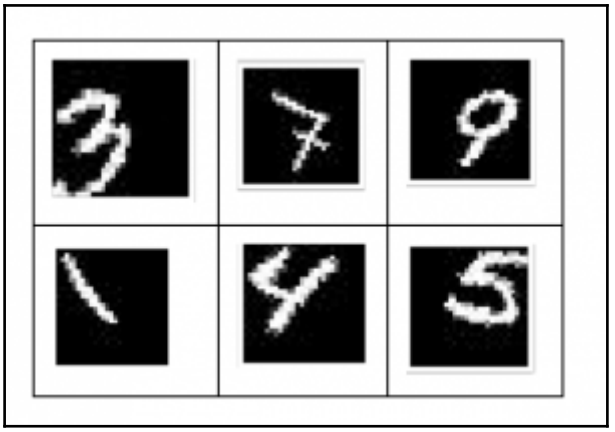


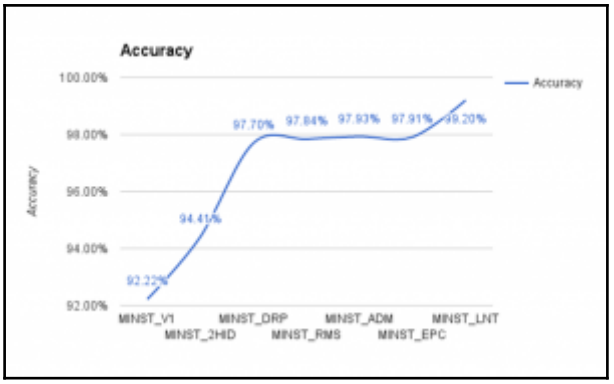

Using TensorFlow backend.


model accuracy
train
test
accuracy
epoch
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0
5
10
15
20


model loss
train
test
loss
epoch
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20

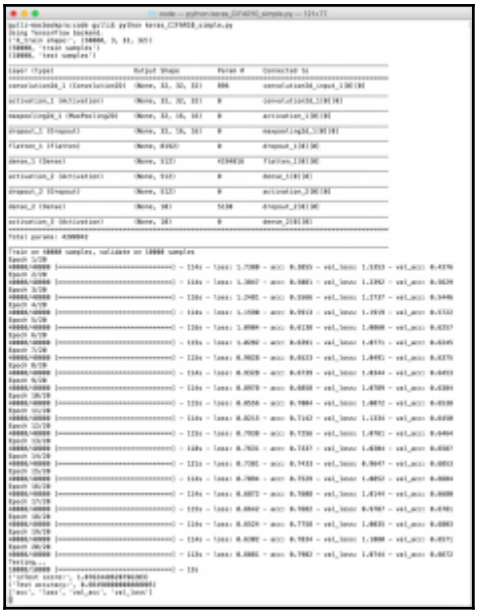

guili-macbookpro:code guili$ python keras_LeNet.py
Using TensorFlow backend.
('60000', 'train samples')
('10000', 'test samples')
Train on 48000 samples, validate on 12000 samples
Epoch 1/4
48000/48000 [==============================] - 139s - loss: 0.1758 - acc: 0.9458 - val_loss: 0.0618 - val_acc: 0.9806
Epoch 2/4
48000/48000 [==============================] - 136s - loss: 0.0601 - acc: 0.9849 - val_loss: 0.0480 - val_acc: 0.9870
Epoch 3/4
48000/48000 [==============================] - 136s - loss: 0.0394 - acc: 0.9905 - val_loss: 0.0413 - val_acc: 0.9889
Epoch 4/4
48000/48000 [==============================] - 129s - loss: 0.0299 - acc: 0.9936 - val_loss: 0.0375 - val_acc: 0.9900
10000/10000 [==============================] - 32s
('\nTest score:', 0.027187511873913574)
('Test accuracy:', 0.99209999999999998)
['acc', 'loss', 'val_acc', 'val_loss']

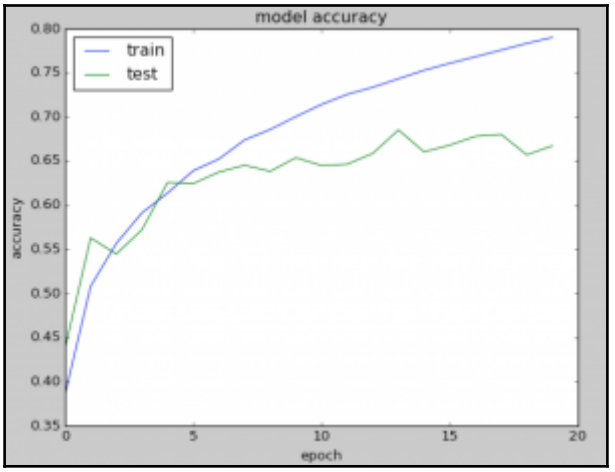


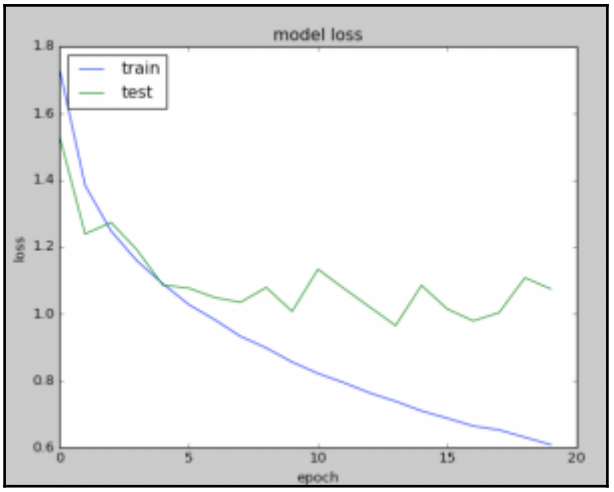

Accuracy
100.00%
98.00%
96.00%
94.00%
92.00%
Accuracy
Accuracy
92.22%
94.41%
97.70%
97.84%
97.93%
97.91%
98.20%
MINST_V1
MINST_2HID
MINST_DRP
MINST_RMS
MINST_ADM
MINST_EPC
MINST_LNT


accuracy_deep and accuracy
100.00%
75.00%
50.00%
25.00%
0.00%
accuracy_deep
accuracy
92.32%
90.00%
85.56%
81.76%
78.85%
78.14%
72.44%
68.23%
48.26%
MINST_DEP=5000
MINST_DEP=3000
MINST_DEP=1500
MINST_DEP=600
MINST_DEP=300


airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

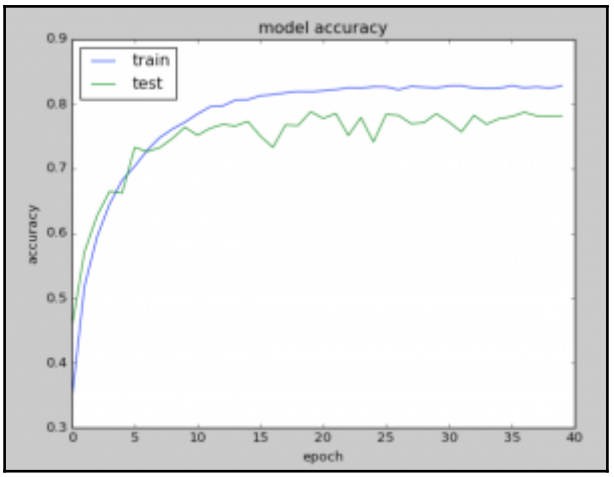


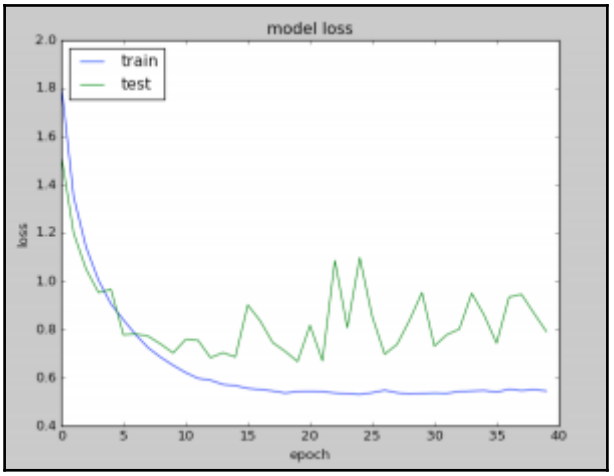

model accuracy
train
test
accuracy
epoch
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0
5
10
15
20



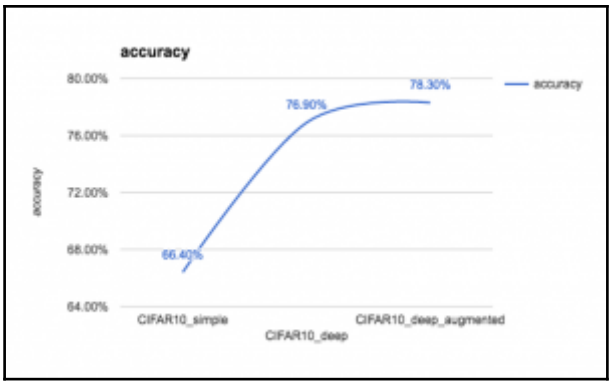

```
guili-macbookpro:code guili$ python keras_CIFAR10_V1.py
Using TensorFlow backend.
('X_train shape:', (50000, 3, 32, 32))
(50000, 'train samples')
(10000, 'test samples')
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
convolution2d_1 (Convolution2D)  (None, 32, 32, 32)    896         convolution2d_input_1[0][0]
____________________________________________________________________________________________________
activation_1 (Activation)        (None, 32, 32, 32)    0           convolution2d_1[0][0]
____________________________________________________________________________________________________
convolution2d_2 (Convolution2D)  (None, 32, 32, 32)    9248        activation_1[0][0]
____________________________________________________________________________________________________
activation_2 (Activation)        (None, 32, 32, 32)    0           convolution2d_2[0][0]
____________________________________________________________________________________________________
maxpooling2d_1 (MaxPooling2D)    (None, 32, 16, 16)    0           activation_2[0][0]
____________________________________________________________________________________________________
dropout_1 (Dropout)              (None, 32, 16, 16)    0           maxpooling2d_1[0][0]
____________________________________________________________________________________________________
convolution2d_3 (Convolution2D)  (None, 64, 16, 16)    18496       dropout_1[0][0]
____________________________________________________________________________________________________
activation_3 (Activation)        (None, 64, 16, 16)    0           convolution2d_3[0][0]
____________________________________________________________________________________________________
convolution2d_4 (Convolution2D)  (None, 64, 14, 14)    36928       activation_3[0][0]
____________________________________________________________________________________________________
activation_4 (Activation)        (None, 64, 14, 14)    0           convolution2d_4[0][0]
____________________________________________________________________________________________________
maxpooling2d_2 (MaxPooling2D)    (None, 64, 7, 7)      0           activation_4[0][0]
____________________________________________________________________________________________________
dropout_2 (Dropout)              (None, 64, 7, 7)      0           maxpooling2d_2[0][0]
____________________________________________________________________________________________________
flatten_1 (Flatten)              (None, 3136)          0           dropout_2[0][0]
____________________________________________________________________________________________________
dense_1 (Dense)                  (None, 512)           1606144     flatten_1[0][0]
____________________________________________________________________________________________________
activation_5 (Activation)        (None, 512)           0           dense_1[0][0]
____________________________________________________________________________________________________
dropout_3 (Dropout)              (None, 512)           0           activation_5[0][0]
____________________________________________________________________________________________________
dense_2 (Dense)                  (None, 10)            5130        dropout_3[0][0]
____________________________________________________________________________________________________
activation_6 (Activation)        (None, 10)            0           dense_2[0][0]
====================================================================================================
Total params: 1676842
____________________________________________________________________________________________________

Train on 40000 samples, validate on 10000 samples
Epoch 1/50
40000/40000 [==============================] - 43s - loss: 1.8179 - acc: 0.3443 - val_loss: 1.5258 - val_acc: 0.4551
Epoch 2/50
40000/40000 [==============================] - 382s - loss: 1.3586 - acc: 0.5182 - val_loss: 1.1998 - val_acc: 0.5714
```

```
Epoch 39/50
40000/40000 [==============================] - 348s - loss: 0.5497 - acc: 0.8246 - val_loss: 0.8689 - val_acc: 0.7011
Epoch 40/50
40000/40000 [==============================] - 348s - loss: 0.5447 - acc: 0.8280 - val_loss: 0.7950 - val_acc: 0.7256
Testing...
10000/10000 [==============================] - 41s
['\nTest score:', 0.7993653458678618]
['Test accuracy:', 0.7682909999999998]
['acc', 'loss', 'val_acc', 'val_loss']
```


model accuracy
train
test
accuracy
epoch
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
30
35
40

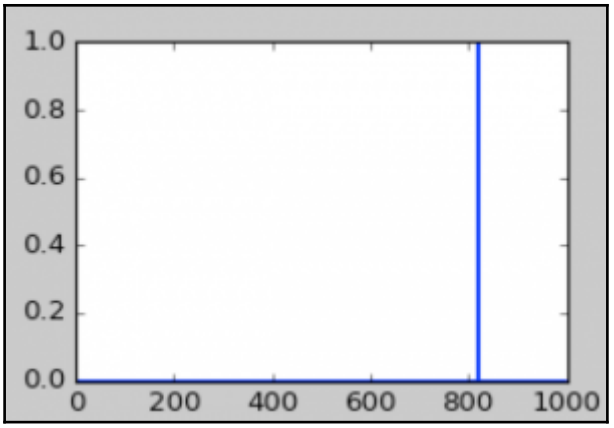

model loss
train
test
loss
epoch
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
5
10
15
20
25
30
35
40

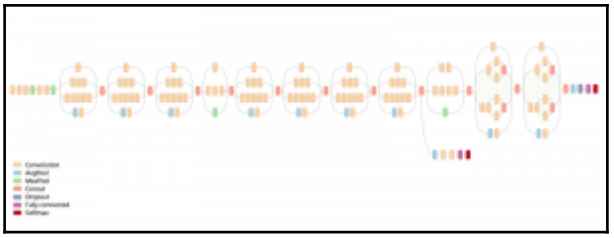
```
Epoch 46/50
50000/50000 [==============================] - 405s - loss: 0.6280 - acc: 0.7297
Epoch 47/50
50000/50000 [==============================] - 424s - loss: 0.6349 - acc: 0.7303
Epoch 48/50
50000/50000 [==============================] - 408s - loss: 0.6319 - acc: 0.7295
Epoch 49/50
50000/50000 [==============================] - 401s - loss: 0.6306 - acc: 0.7301
Epoch 50/50
50000/50000 [==============================] - 398s - loss: 0.6284 - acc: 0.7357
Testing...
10000/10000 [==============================] - 42s
('\nTest score:', 0.73130152046845461)
('Test accuracy:', 0.78369999999999995)
['acc', 'loss']
```



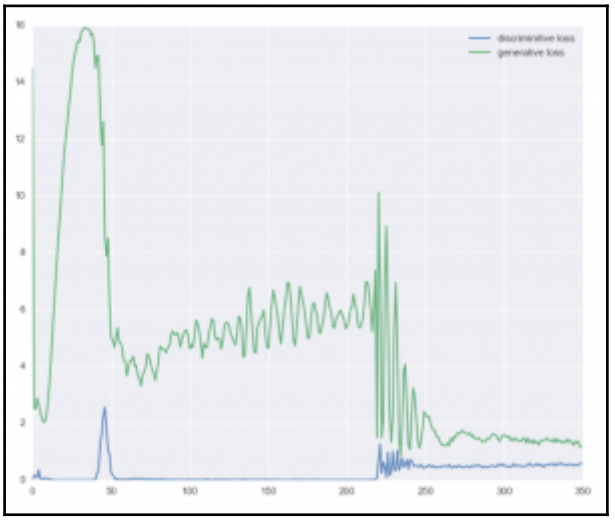
```
gulli-macbookpro:code gulli$ python keras_EvaluateCIFAR10.py
Using TensorFlow backend.
2/2 [==============================] - 0s
[3 5]
gulli-macbookpro:code gulli$
```


code — -bash — 108×5
set properly.
gulli-macbookpro:code gulli$ python keras_VGG16.py
Using TensorFlow backend.
285
gulli-macbookpro:code gulli$

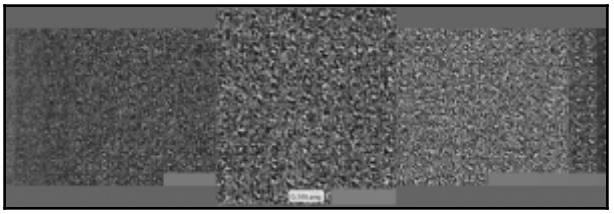

1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000

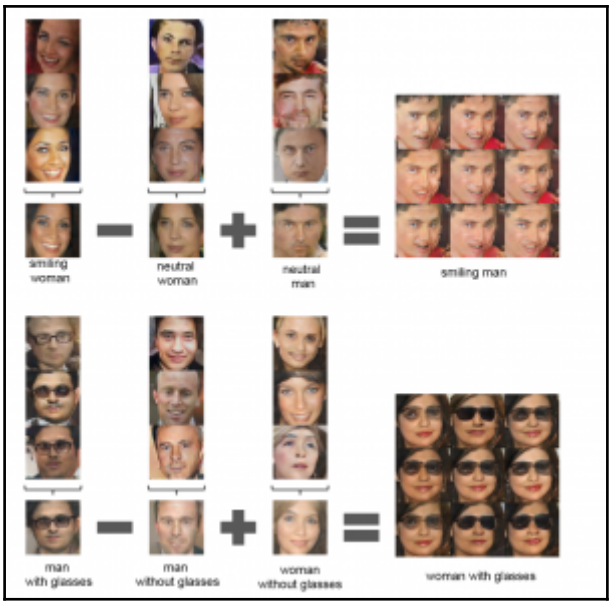


Chapter 4: Generative Adversarial Networks and WaveNet




This flower is pink, white, and yellow in color, and has petals that are striped
Stage-I
Stage-II





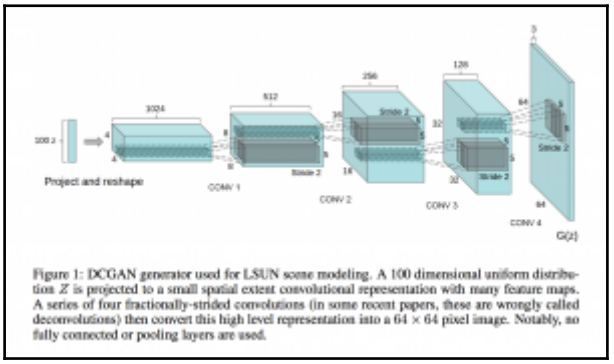
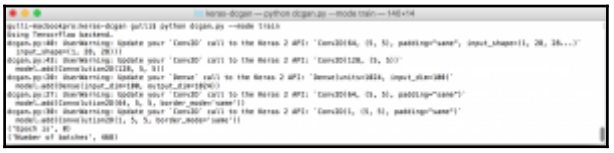
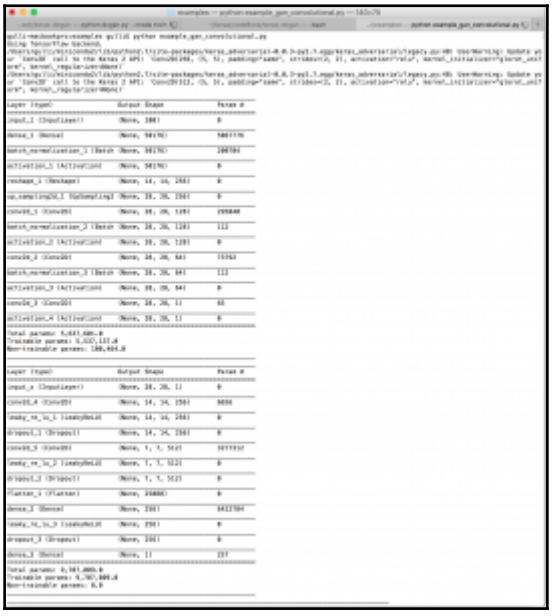
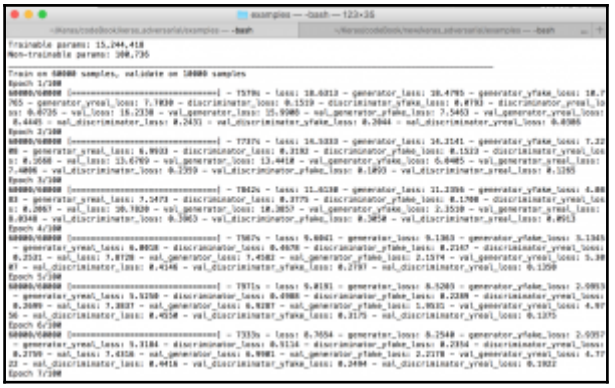
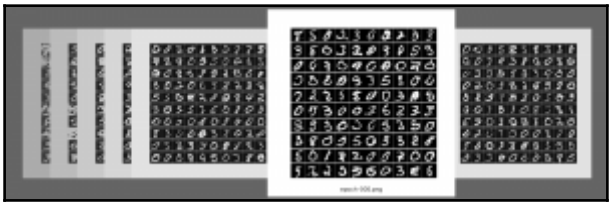


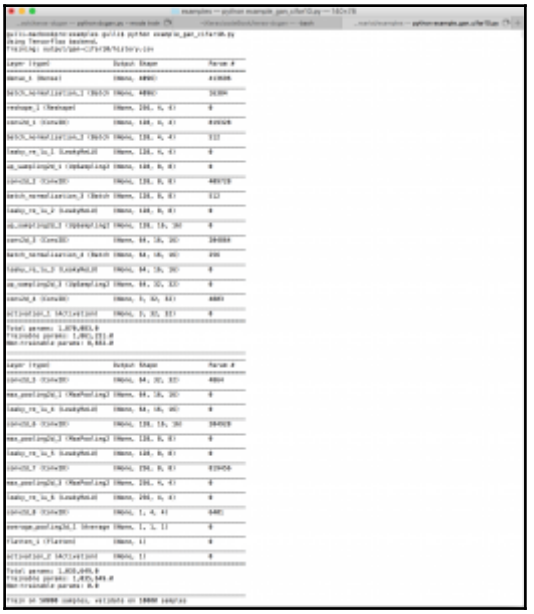
Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

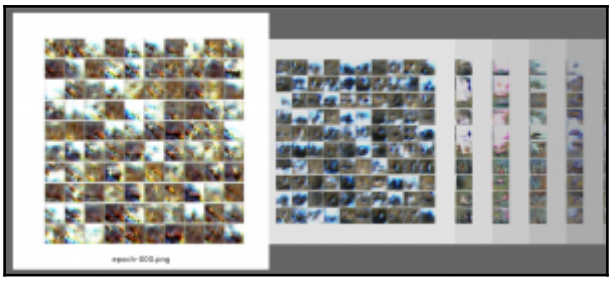




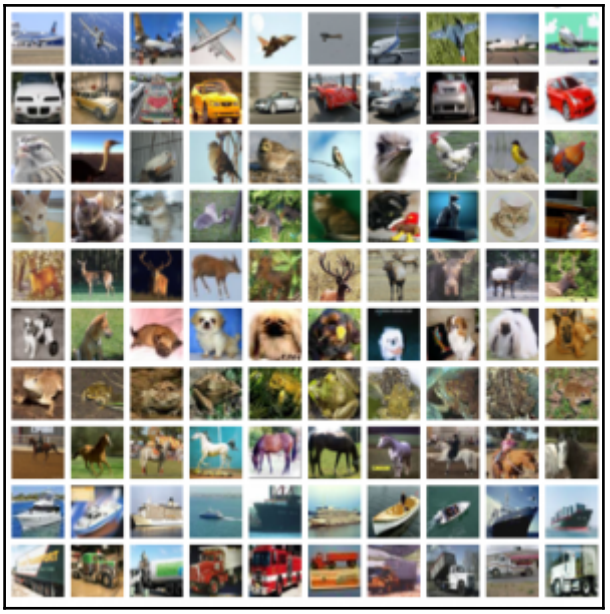

examples — -bash — 123×35
Trainable params: 15,244,418
Non-trainable params: 300,736
Train on 60000 samples, validate on 10000 samples
Epoch 1/100
Epoch 2/100
Epoch 3/100
Epoch 4/100
Epoch 5/100
Epoch 6/100
Epoch 7/100



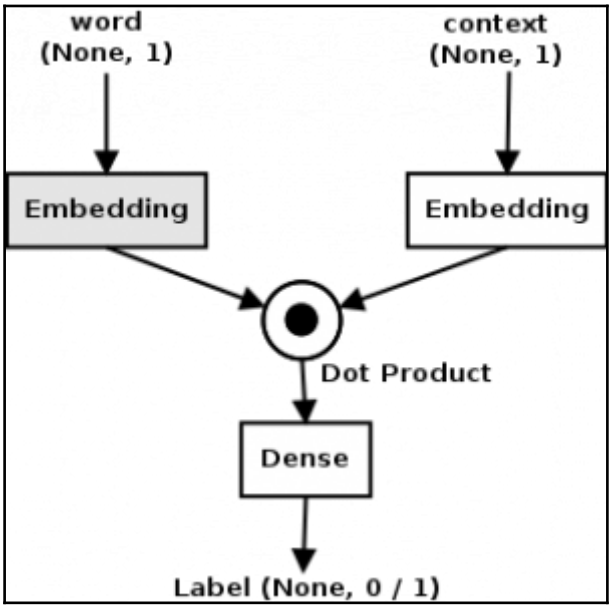

examples — python example_gan_cifar10.py — 142×78
Using TensorFlow backend.
Total params: 1,879,681.0
Trainable params: 1,901,151.0
Non-trainable params: 6,561.0
Total params: 1,858,945.0
Trainable params: 1,851,521.0
Non-trainable params: 0.0

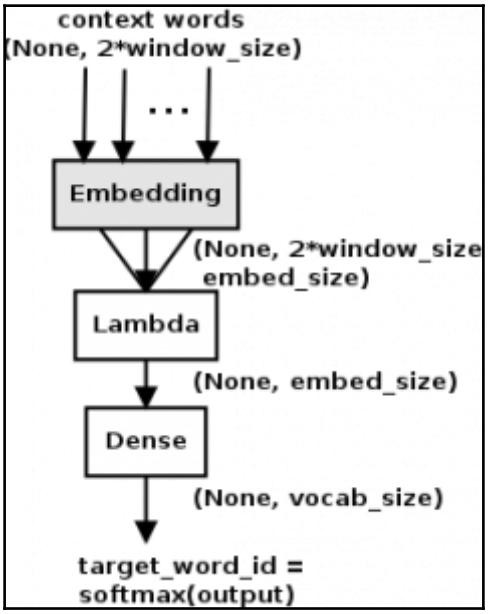


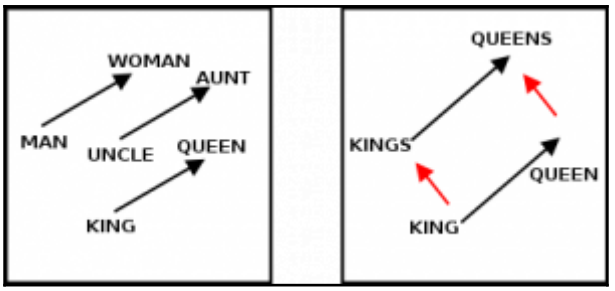
epoch-097.png

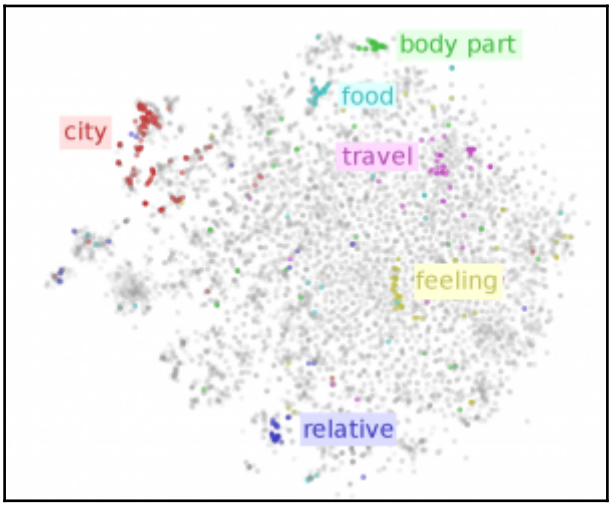


Chapter 5: Word Embeddings

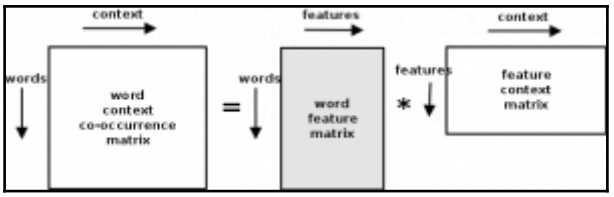









```
2017-01-30 16:16:27,796 : INFO : PROGRESS: at 76.44% examples, 691859 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:28,801 : INFO : PROGRESS: at 77.74% examples, 693040 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:29,807 : INFO : PROGRESS: at 79.00% examples, 693746 words/s, in_qsize 2, out_qsize 0
2017-01-30 16:16:30,815 : INFO : PROGRESS: at 79.99% examples, 692107 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:31,819 : INFO : PROGRESS: at 80.03% examples, 682583 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:32,842 : INFO : PROGRESS: at 81.15% examples, 682090 words/s, in_qsize 1, out_qsize 0
2017-01-30 16:16:33,869 : INFO : PROGRESS: at 82.46% examples, 683117 words/s, in_qsize 0, out_qsize 1
2017-01-30 16:16:34,873 : INFO : PROGRESS: at 83.77% examples, 684403 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:35,882 : INFO : PROGRESS: at 85.02% examples, 685224 words/s, in_qsize 5, out_qsize 0
2017-01-30 16:16:36,884 : INFO : PROGRESS: at 86.36% examples, 686831 words/s, in_qsize 0, out_qsize 1
2017-01-30 16:16:37,925 : INFO : PROGRESS: at 87.51% examples, 686556 words/s, in_qsize 2, out_qsize 0
2017-01-30 16:16:38,925 : INFO : PROGRESS: at 88.57% examples, 685873 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:39,933 : INFO : PROGRESS: at 89.84% examples, 686756 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:40,936 : INFO : PROGRESS: at 91.17% examples, 688126 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:41,939 : INFO : PROGRESS: at 92.43% examples, 688894 words/s, in_qsize 0, out_qsize 1
2017-01-30 16:16:42,946 : INFO : PROGRESS: at 93.69% examples, 689612 words/s, in_qsize 1, out_qsize 0
2017-01-30 16:16:43,960 : INFO : PROGRESS: at 94.97% examples, 690484 words/s, in_qsize 1, out_qsize 0
2017-01-30 16:16:44,978 : INFO : PROGRESS: at 96.30% examples, 691348 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:45,982 : INFO : PROGRESS: at 97.58% examples, 692158 words/s, in_qsize 0, out_qsize 0
2017-01-30 16:16:46,980 : INFO : PROGRESS: at 98.83% examples, 692731 words/s, in_qsize 2, out_qsize 0
2017-01-30 16:16:48,092 : INFO : PROGRESS: at 99.92% examples, 691317 words/s, in_qsize 4, out_qsize 1
2017-01-30 16:16:48,124 : INFO : worker thread finished; awaiting finish of 2 more threads
2017-01-30 16:16:48,125 : INFO : worker thread finished; awaiting finish of 1 more threads
2017-01-30 16:16:48,128 : INFO : worker thread finished; awaiting finish of 0 more threads
2017-01-30 16:16:48,129 : INFO : training on 85026040 raw words (59645573 effective words) took 86.2s, 691572
effective words/s
2017-01-30 16:16:48,129 : INFO : precomputing L2-norms of word weight vectors
```

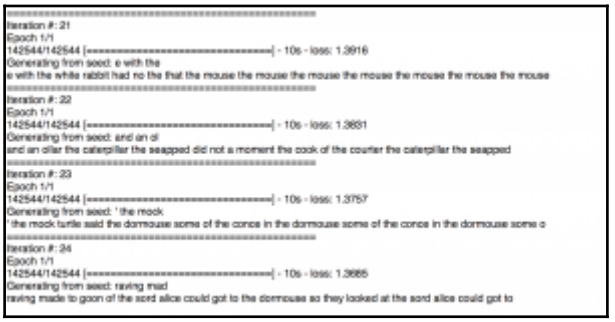


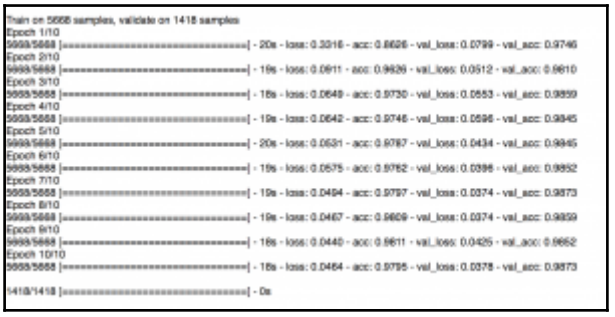

```
Epoch 9/20
4960/4960 [==============================] - 3s - loss: 0.0337 - acc: 0.9855 - val_loss: 0.0263 - val_acc: 0.9882
Epoch 10/20
4960/4960 [==============================] - 3s - loss: 0.0369 - acc: 0.9843 - val_loss: 0.0277 - val_acc: 0.9878
Epoch 11/20
4960/4960 [==============================] - 3s - loss: 0.0331 - acc: 0.9881 - val_loss: 0.0303 - val_acc: 0.9878
Epoch 12/20
4960/4960 [==============================] - 3s - loss: 0.0289 - acc: 0.9879 - val_loss: 0.0291 - val_acc: 0.9882
Epoch 13/20
4960/4960 [==============================] - 3s - loss: 0.0261 - acc: 0.9901 - val_loss: 0.0305 - val_acc: 0.9878
Epoch 14/20
4960/4960 [==============================] - 3s - loss: 0.0261 - acc: 0.9895 - val_loss: 0.0310 - val_acc: 0.9859
Epoch 15/20
4960/4960 [==============================] - 3s - loss: 0.0355 - acc: 0.9857 - val_loss: 0.0307 - val_acc: 0.9873
Epoch 16/20
4960/4960 [==============================] - 3s - loss: 0.0247 - acc: 0.9893 - val_loss: 0.0263 - val_acc: 0.9888
Epoch 17/20
4960/4960 [==============================] - 3s - loss: 0.0249 - acc: 0.9891 - val_loss: 0.0329 - val_acc: 0.9854
Epoch 18/20
4960/4960 [==============================] - 3s - loss: 0.0299 - acc: 0.9895 - val_loss: 0.0265 - val_acc: 0.9882
Epoch 19/20
4960/4960 [==============================] - 3s - loss: 0.0282 - acc: 0.9887 - val_loss: 0.0287 - val_acc: 0.9882
Epoch 20/20
4960/4960 [==============================] - 3s - loss: 0.0401 - acc: 0.9839 - val_loss: 0.0311 - val_acc: 0.9878

2126/2126 [==============================] - 0s
Test score: 0.031, accuracy: 0.986
```

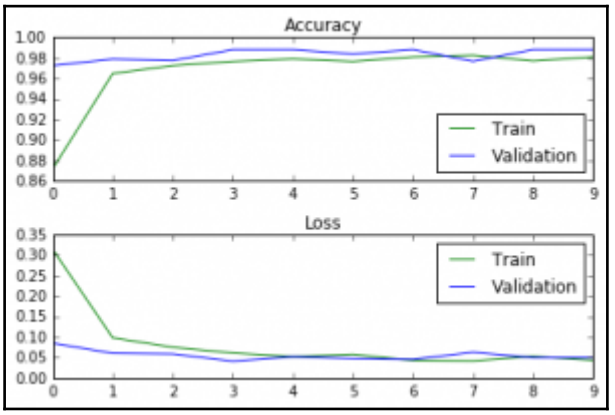
```
[(4960, 40), (2126, 40), (4960, 2), (2126, 2)]
Train on 4960 samples, validate on 2126 samples
Epoch 1/10
4960/4960 [==============================] - 7s - loss: 0.1748 - acc: 0.9240 - val_loss: 0.0360 - val_acc: 0.9840
Epoch 2/10
4960/4960 [==============================] - 7s - loss: 0.0859 - acc: 0.9649 - val_loss: 0.0431 - val_acc: 0.9845
Epoch 3/10
4960/4960 [==============================] - 7s - loss: 0.0585 - acc: 0.9754 - val_loss: 0.0528 - val_acc: 0.9779
Epoch 4/10
4960/4960 [==============================] - 8s - loss: 0.0565 - acc: 0.9798 - val_loss: 0.0386 - val_acc: 0.9873
Epoch 5/10
4960/4960 [==============================] - 8s - loss: 0.0792 - acc: 0.9683 - val_loss: 0.0233 - val_acc: 0.9892
Epoch 6/10
4960/4960 [==============================] - 8s - loss: 0.0618 - acc: 0.9746 - val_loss: 0.0247 - val_acc: 0.9911
Epoch 7/10
4960/4960 [==============================] - 7s - loss: 0.0569 - acc: 0.9752 - val_loss: 0.0266 - val_acc: 0.9906
Epoch 8/10
4960/4960 [==============================] - 8s - loss: 0.0419 - acc: 0.9829 - val_loss: 0.0211 - val_acc: 0.9920
Epoch 9/10
4960/4960 [==============================] - 7s - loss: 0.0371 - acc: 0.9849 - val_loss: 0.0206 - val_acc: 0.9920
Epoch 10/10
4960/4960 [==============================] - 9s - loss: 0.0422 - acc: 0.9815 - val_loss: 0.0266 - val_acc: 0.9906
2126/2126 [==============================] - 1s
Test score: 0.027, accuracy: 0.991
```

```
[(4960, 100), (2126, 100), (4960, 2), (2126, 2)]
Train on 4960 samples, validate on 2126 samples
Epoch 1/10
4960/4960 [==============================] - 0s - loss: 1.9577 - acc: 0.5667 - val_loss: 0.4448 - val_acc: 0.8556
Epoch 2/10
4960/4960 [==============================] - 0s - loss: 0.5245 - acc: 0.7842 - val_loss: 0.3167 - val_acc: 0.9078
Epoch 3/10
4960/4960 [==============================] - 0s - loss: 0.3025 - acc: 0.9002 - val_loss: 0.2456 - val_acc: 0.9473
Epoch 4/10
4960/4960 [==============================] - 0s - loss: 0.2338 - acc: 0.9270 - val_loss: 0.2068 - val_acc: 0.9398
Epoch 5/10
4960/4960 [==============================] - 0s - loss: 0.1802 - acc: 0.9520 - val_loss: 0.1720 - val_acc: 0.9581
Epoch 6/10
4960/4960 [==============================] - 0s - loss: 0.1561 - acc: 0.9552 - val_loss: 0.1561 - val_acc: 0.9610
Epoch 7/10
4960/4960 [==============================] - 0s - loss: 0.1396 - acc: 0.9631 - val_loss: 0.1535 - val_acc: 0.9577
Epoch 8/10
4960/4960 [==============================] - 0s - loss: 0.1216 - acc: 0.9645 - val_loss: 0.1338 - val_acc: 0.9628
Epoch 9/10
4960/4960 [==============================] - 0s - loss: 0.1152 - acc: 0.9641 - val_loss: 0.1273 - val_acc: 0.9643
Epoch 10/10
4960/4960 [==============================] - 0s - loss: 0.1044 - acc: 0.9706 - val_loss: 0.1257 - val_acc: 0.9647

1888/2126 [=========================>....] - ETA: 0s
Test score: 0.126, accuracy: 0.965
```

Chapter 6: Recurrent Neural Network — RNN



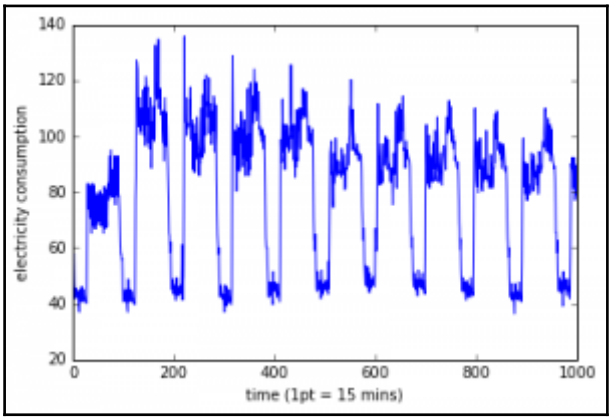




```
Test score: 0.038, accuracy: 0.987

#pred label sentence
1     1     i like th mission impossible one ...
1     1     we 're gon na like watch mission impossible or hoot . (
1     1     the people who are worth it know how much i love the da vinci code .
0     0     ok brokeback mountain is such a horrible movie .
1     1     brokeback mountain is the most amazing / beautiful / romantic /
            Heartbreaking movie I have ever or will ever see in my life
```

```
Train on 3131 samples, validate on 783 samples
Epoch 1/1
3131/3131 [==============================] - 81s - loss: 0.3013 - acc: 0.8263 - val_loss: 0.2934 - val_acc: 0.9159

783/783 [==============================] - 3s
Test score: 0.293, accuracy: 0.916
```

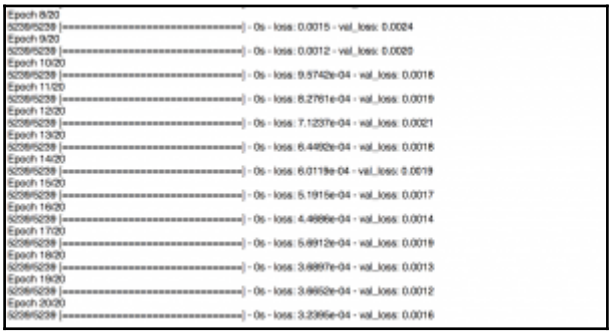
```
Train on 3131 samples, validate on 783 samples
Epoch 1/1
3131/3131 [==============================] - 269s - loss: 0.2889 - acc: 0.8226 - val_loss: 0.2788 - val_acc: 0.9036

783/783 [==============================] - 12s
Test score: 0.279, accuracy: 0.904
```

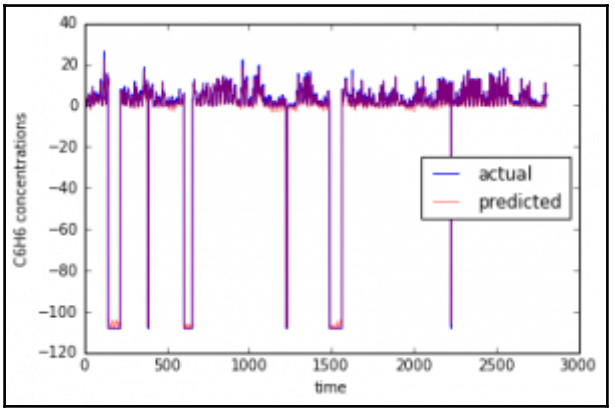


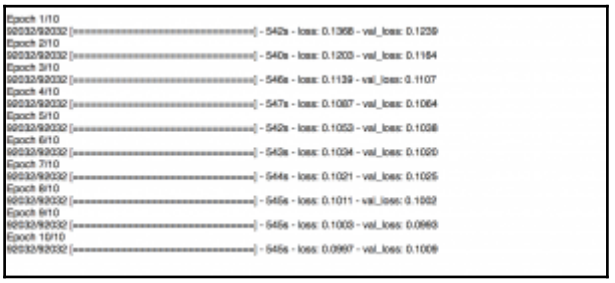

```
(98179, 20, 1) (42077, 20, 1) (98179, 1) (42077, 1)
Train on 98179 samples, validate on 42077 samples
Epoch 1/5
98179/98179 [==============================] - 41s - loss: 0.0066 - mean_squared_error: 0.0066 - val_loss: 0.0040 -
val_mean_squared_error: 0.0040
Epoch 2/5
98179/98179 [==============================] - 41s - loss: 0.0045 - mean_squared_error: 0.0045 - val_loss: 0.0039 -
val_mean_squared_error: 0.0039
Epoch 3/5
98179/98179 [==============================] - 43s - loss: 0.0041 - mean_squared_error: 0.0041 - val_loss: 0.0038 -
val_mean_squared_error: 0.0038
Epoch 4/5
98179/98179 [==============================] - 44s - loss: 0.0039 - mean_squared_error: 0.0039 - val_loss: 0.0040 -
val_mean_squared_error: 0.0040
Epoch 5/5
98179/98179 [==============================] - 44s - loss: 0.0038 - mean_squared_error: 0.0038 - val_loss: 0.0038 -
val_mean_squared_error: 0.0038

42077/42077 [==============================] - 2s

MSE: 0.004, RMSE: 0.062
```

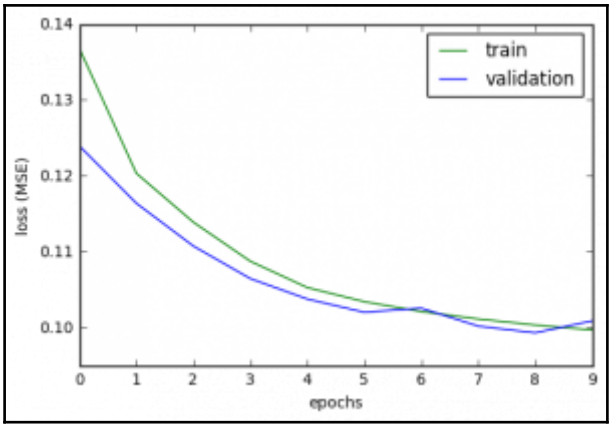
```
Train on 98112 samples, validate on 42048 samples
Epoch 1/1
98112/98112 [==============================] - 37s - loss: 0.0056 - mean_squared_error: 0.0056 - val_loss: 0.0038 -
val_mean_squared_error: 0.0038
Epoch 2/5
Train on 98112 samples, validate on 42048 samples
Epoch 1/1
98112/98112 [==============================] - 36s - loss: 0.0044 - mean_squared_error: 0.0044 - val_loss: 0.0037 -
val_mean_squared_error: 0.0037
Epoch 3/5
Train on 98112 samples, validate on 42048 samples
Epoch 1/1
98112/98112 [==============================] - 38s - loss: 0.0043 - mean_squared_error: 0.0043 - val_loss: 0.0038 -
val_mean_squared_error: 0.0038
Epoch 4/5
Train on 98112 samples, validate on 42048 samples
Epoch 1/1
98112/98112 [==============================] - 37s - loss: 0.0042 - mean_squared_error: 0.0042 - val_loss: 0.0038 -
val_mean_squared_error: 0.0038
Epoch 5/5
Train on 98112 samples, validate on 42048 samples
Epoch 1/1
98112/98112 [==============================] - 37s - loss: 0.0040 - mean_squared_error: 0.0040 - val_loss: 0.0035 -
val_mean_squared_error: 0.0035
41952/42048 [============================>.] - ETA: 0s

MSE: 0.003, RMSE: 0.059
```

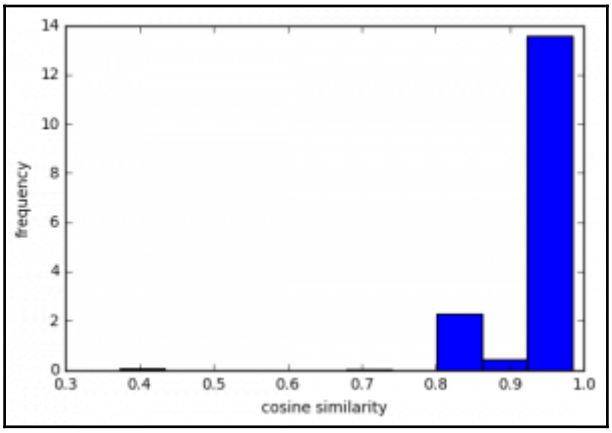
Chapter 7: Additional Deep Learning Models

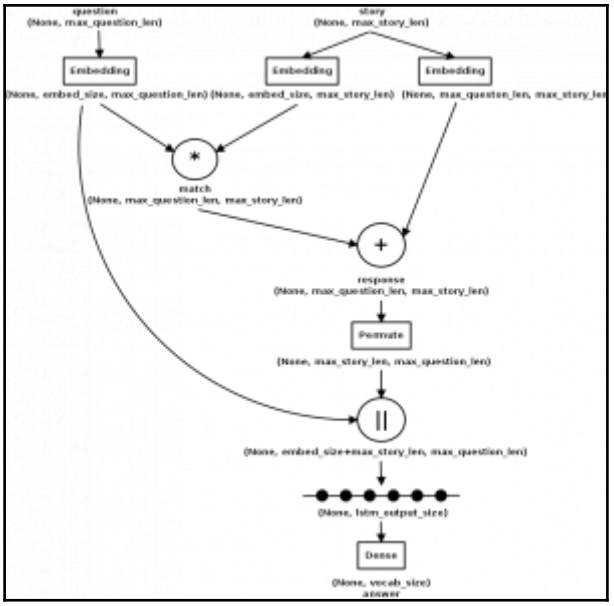




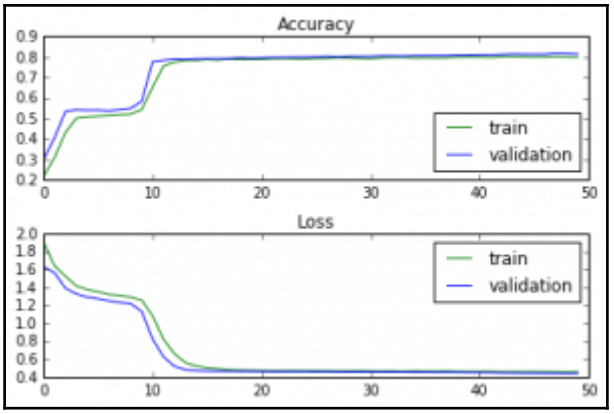


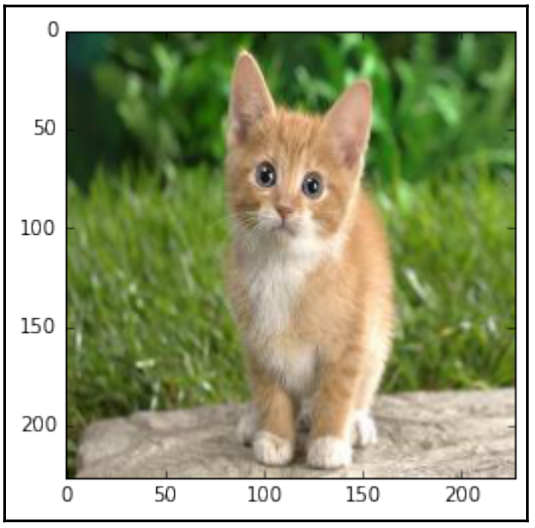

train
validation
loss (MSE)
epochs
0.14
0.13
0.12
0.11
0.10
0
1
2
3
4
5
6
7
8
9

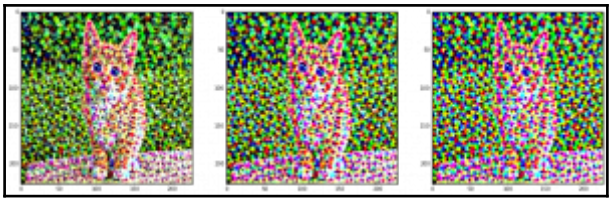

frequency
cosine similarity
14
12
10
8
6
4
2
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

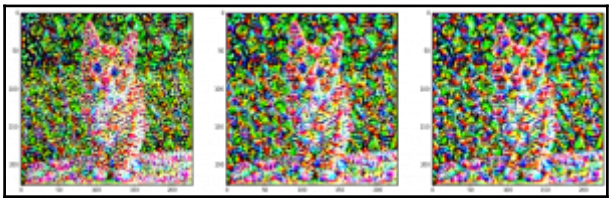


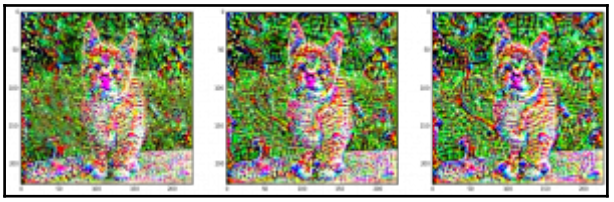

```
Epoch 38/50
10000/10000 [==============================] - 5s - loss: 0.4536 - acc: 0.7952 - val_loss: 0.4489 - val_acc: 0.8071
Epoch 39/50
10000/10000 [==============================] - 5s - loss: 0.4503 - acc: 0.7993 - val_loss: 0.4489 - val_acc: 0.8083
Epoch 40/50
10000/10000 [==============================] - 5s - loss: 0.4490 - acc: 0.8003 - val_loss: 0.4475 - val_acc: 0.8086
Epoch 41/50
10000/10000 [==============================] - 5s - loss: 0.4492 - acc: 0.7997 - val_loss: 0.4472 - val_acc: 0.8099
Epoch 42/50
10000/10000 [==============================] - 5s - loss: 0.4511 - acc: 0.7966 - val_loss: 0.4489 - val_acc: 0.8069
Epoch 43/50
10000/10000 [==============================] - 5s - loss: 0.4577 - acc: 0.8025 - val_loss: 0.4437 - val_acc: 0.8114
Epoch 44/50
10000/10000 [==============================] - 5s - loss: 0.4576 - acc: 0.8023 - val_loss: 0.4431 - val_acc: 0.8136
Epoch 45/50
10000/10000 [==============================] - 5s - loss: 0.4575 - acc: 0.8013 - val_loss: 0.4422 - val_acc: 0.8127
Epoch 46/50
10000/10000 [==============================] - 5s - loss: 0.4587 - acc: 0.7998 - val_loss: 0.4420 - val_acc: 0.8127
Epoch 47/50
10000/10000 [==============================] - 5s - loss: 0.4574 - acc: 0.8005 - val_loss: 0.4412 - val_acc: 0.8126
Epoch 48/50
10000/10000 [==============================] - 5s - loss: 0.4559 - acc: 0.8023 - val_loss: 0.4408 - val_acc: 0.8168
Epoch 49/50
10000/10000 [==============================] - 5s - loss: 0.4550 - acc: 0.8003 - val_loss: 0.4395 - val_acc: 0.8154
Epoch 50/50
10000/10000 [==============================] - 5s - loss: 0.4577 - acc: 0.7985 - val_loss: 0.4407 - val_acc: 0.8139
```

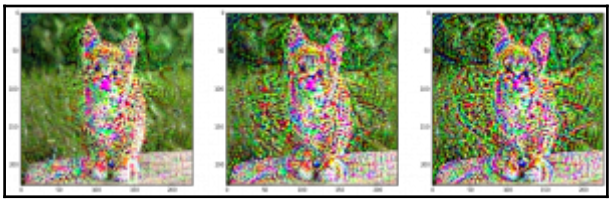

Accuracy
train
validation
Loss
train
validation

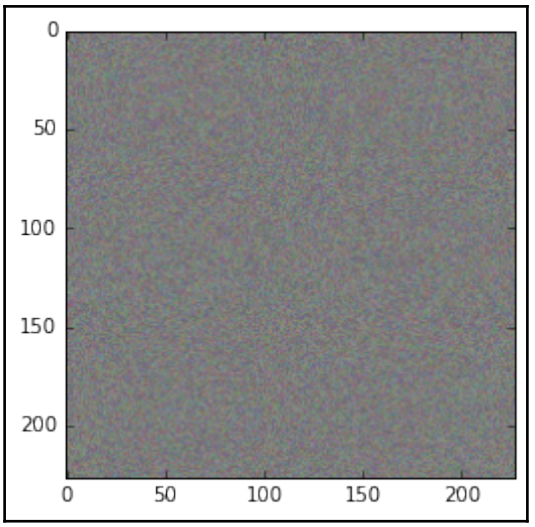


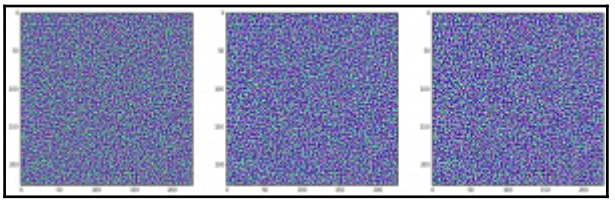


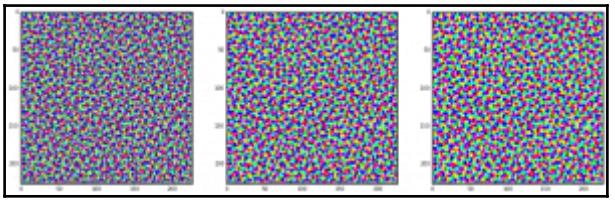


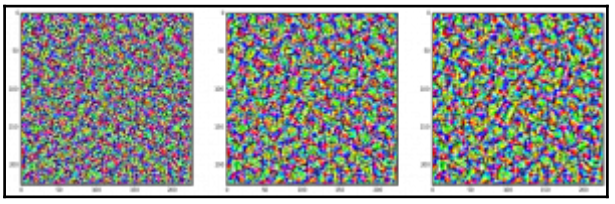


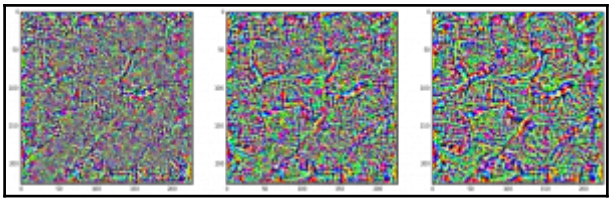


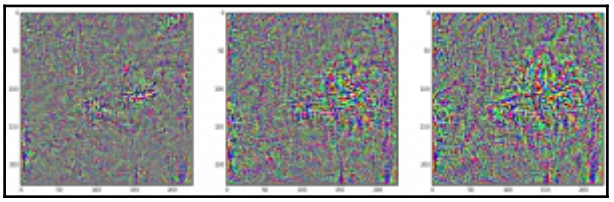

0
50
100
150
200
0
50
100
150
200

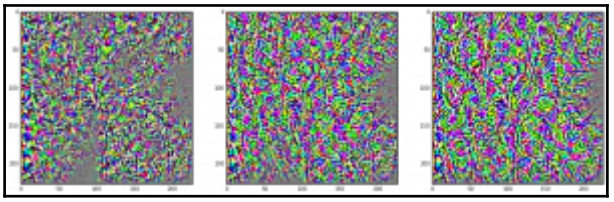


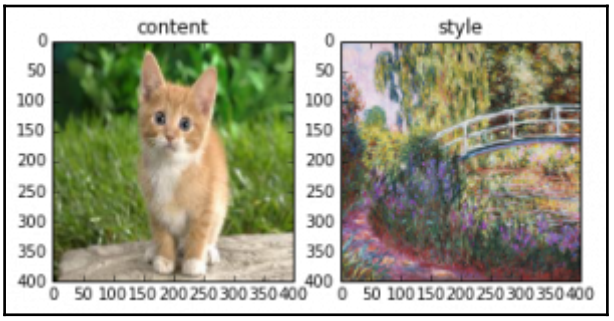


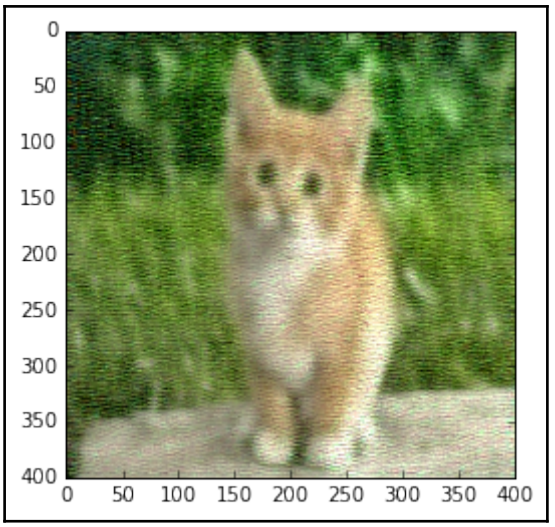


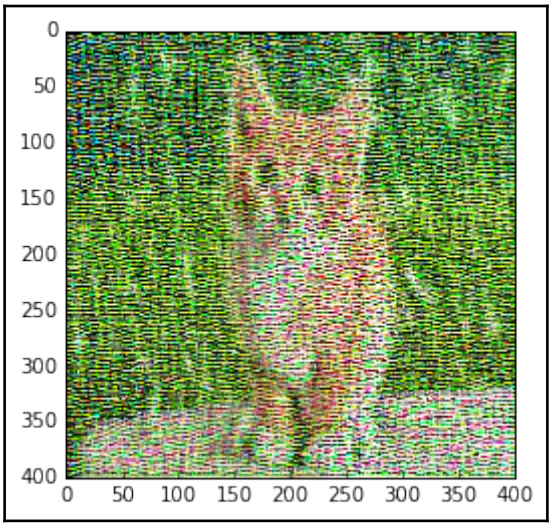






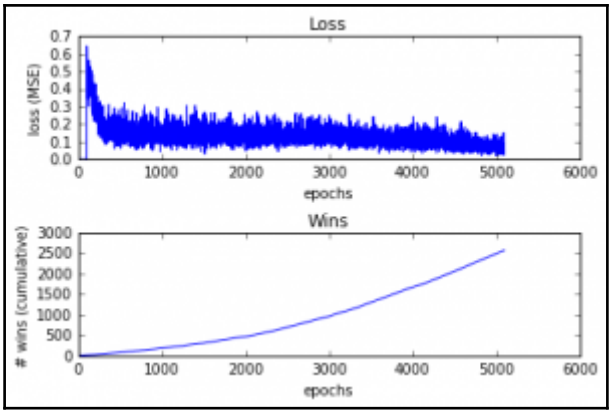

content
style
0
50
100
150
200
250
300
350
400


0
50
100
150
200
250
300
350
400
0 50 100 150 200 250 300 350 400


0
50
100
150
200
250
300
350
400
0 50 100 150 200 250 300 350 400

Chapter 8: AI Game Playing

```
Epoch 5075/5100 | Loss 0.02603 | Win Count 2548
Epoch 5076/5100 | Loss 0.06248 | Win Count 2549
Epoch 5077/5100 | Loss 0.09836 | Win Count 2550
Epoch 5078/5100 | Loss 0.05955 | Win Count 2551
Epoch 5079/5100 | Loss 0.07357 | Win Count 2552
Epoch 5080/5100 | Loss 0.05425 | Win Count 2553
Epoch 5081/5100 | Loss 0.05961 | Win Count 2553
Epoch 5082/5100 | Loss 0.05737 | Win Count 2553
Epoch 5083/5100 | Loss 0.06699 | Win Count 2554
Epoch 5084/5100 | Loss 0.04265 | Win Count 2555
Epoch 5085/5100 | Loss 0.06579 | Win Count 2556
Epoch 5086/5100 | Loss 0.06825 | Win Count 2557
Epoch 5087/5100 | Loss 0.09329 | Win Count 2557
Epoch 5088/5100 | Loss 0.06124 | Win Count 2558
Epoch 5089/5100 | Loss 0.15128 | Win Count 2559
Epoch 5090/5100 | Loss 0.03769 | Win Count 2560
Epoch 5091/5100 | Loss 0.06348 | Win Count 2560
Epoch 5092/5100 | Loss 0.03817 | Win Count 2561
Epoch 5093/5100 | Loss 0.05225 | Win Count 2562
Epoch 5094/5100 | Loss 0.04986 | Win Count 2563
Epoch 5095/5100 | Loss 0.06316 | Win Count 2564
Epoch 5096/5100 | Loss 0.07558 | Win Count 2564
Epoch 5097/5100 | Loss 0.04027 | Win Count 2565
Epoch 5098/5100 | Loss 0.03801 | Win Count 2566
Epoch 5099/5100 | Loss 0.02446 | Win Count 2567
Epoch 5100/5100 | Loss 0.04321 | Win Count 2568
```


Loss
loss (MSE)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1000
2000
3000
4000
5000
6000
epochs
Wins
wins (cumulative)
3000
2500
2000
1500
1000
500
0
0
1000
2000
3000
4000
5000
6000
epochs

Appendix: Conclusion