Tamkang University
淡江大學

Tamkang University

Social Computing and Big Data Analytics
社群運算與大數據分析
Deep Learning with Google TensorFlow
(Google TensorFlow 深度學習)

1052SCBDA10
MIS MBA (M2226) (8606)
Wed, 8,9, (15:10-17:00) (L206)

Min-Yuh Day
戴敏育
Assistant Professor
專任助理教授
Dept. of Information Management, Tamkang University
淡江大學 資訊管理學系
http://mail. tku.edu.tw/myday/

2017-05-03

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2017/02/15	Course Orientation for Social Computing and Big Data Analytics (社群運算與大數據分析課程介紹)
2	2017/02/22	Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data (資料科學與大數據分析：探索、分析、視覺化與呈現資料)
3	2017/03/01	Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem (大數據基礎：MapReduce典範、Hadoop與Spark生態系統)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
4	2017/03/08	Big Data Processing Platforms with SMACK: Spark, Mesos, Akka, Cassandra and Kafka (大數據處理平台SMACK：Spark, Mesos, Akka, Cassandra, Kafka)
5	2017/03/15	Big Data Analytics with Numpy in Python (Python Numpy 大數據分析)
6	2017/03/22	Finance Big Data Analytics with Pandas in Python (Python Pandas 財務大數據分析)
7	2017/03/29	Text Mining Techniques and Natural Language Processing (文字探勘分析技術與自然語言處理)
8	2017/04/05	Off-campus study (教學行政觀摩日)

課程大綱 (Syllabus)

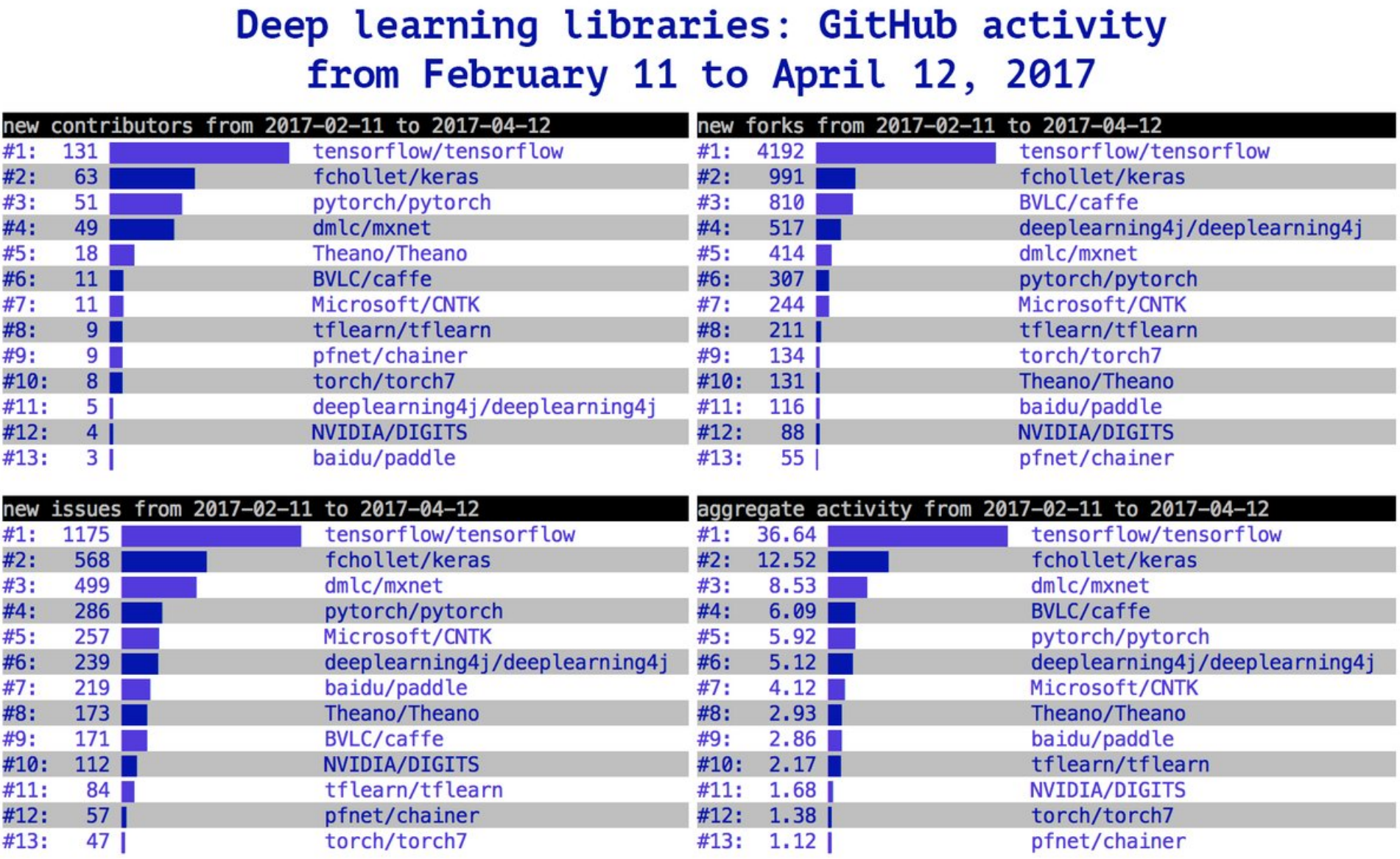
週次 (Week)	日期 (Date)	內容 (Subject/Topics)
9	2017/04/12	Social Media Marketing Analytics (社群媒體行銷分析)
10	2017/04/19	期中報告 (Midterm Project Report)
11	2017/04/26	Deep Learning with Theano and Keras in Python (Python Theano 和 Keras 深度學習)
12	2017/05/03	Deep Learning with Google TensorFlow (Google TensorFlow 深度學習)
13	2017/05/10	Sentiment Analysis on Social Media with Deep Learning (深度學習社群媒體情感分析)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
14	2017/05/17	Social Network Analysis (社會網絡分析)
15	2017/05/24	Measurements of Social Network (社會網絡量測)
16	2017/05/31	Tools of Social Network Analysis (社會網絡分析工具)
17	2017/06/07	Final Project Presentation I (期末報告 I)
18	2017/06/14	Final Project Presentation II (期末報告 II)

Deep Learning with Google TensorFlow

Deep Learning Libraries: Tensorflow and Keras

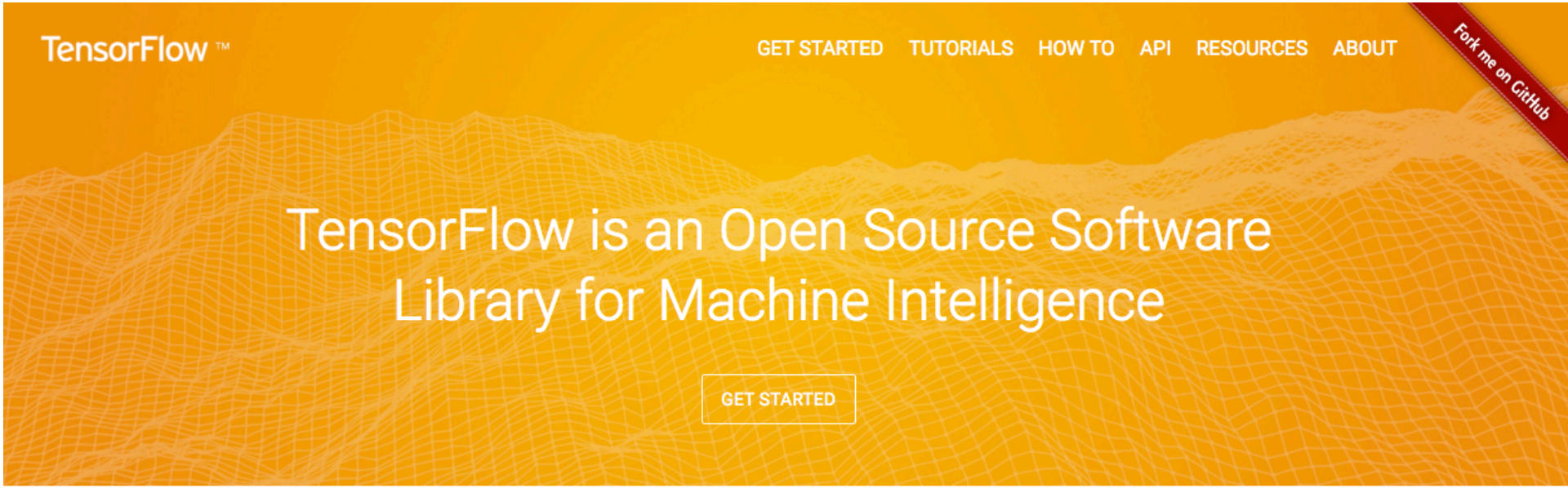


Source: https://twitter.com/fchollet/status/852194634470223873

TensorFlow

Source: https://github.com/tensorflow/tensorflow

Google TensorFlow



About TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

https://www.tensorflow.org/

TensorFlow

TensorFlow

is an

Open Source

Software Library

for

Machine Intelligence

https://www.tensorflow.org/

TensorFlow

numerical computation using data flow graphs

https://www.tensorflow.org/

TensorFlow

Tensor

- **3**
  - # a rank 0 tensor; this is a **scalar** with shape []
- **[1. ,2., 3.]**
  - # a rank 1 tensor; this is a **vector** with shape [3]
- **[[1., 2., 3.], [4., 5., 6.]]**
  - # a rank 2 tensor; a **matrix** with shape [2, 3]
- **[[[1., 2., 3.]], [[7., 8., 9.]]]**
  - # a rank 3 **tensor** with shape [2, 1, 3]

https://www.tensorflow.org/

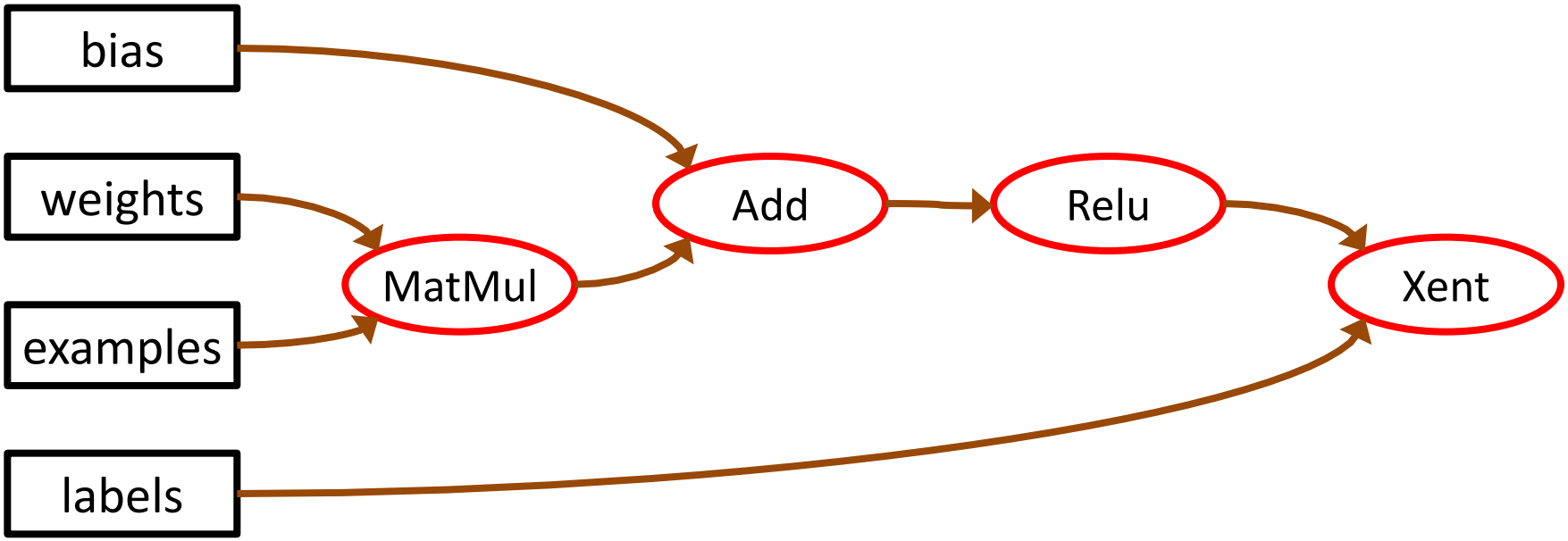
TensorFlow

Nodes:
mathematical operations

edges:
multidimensional data arrays (tensors)
communicated between nodes

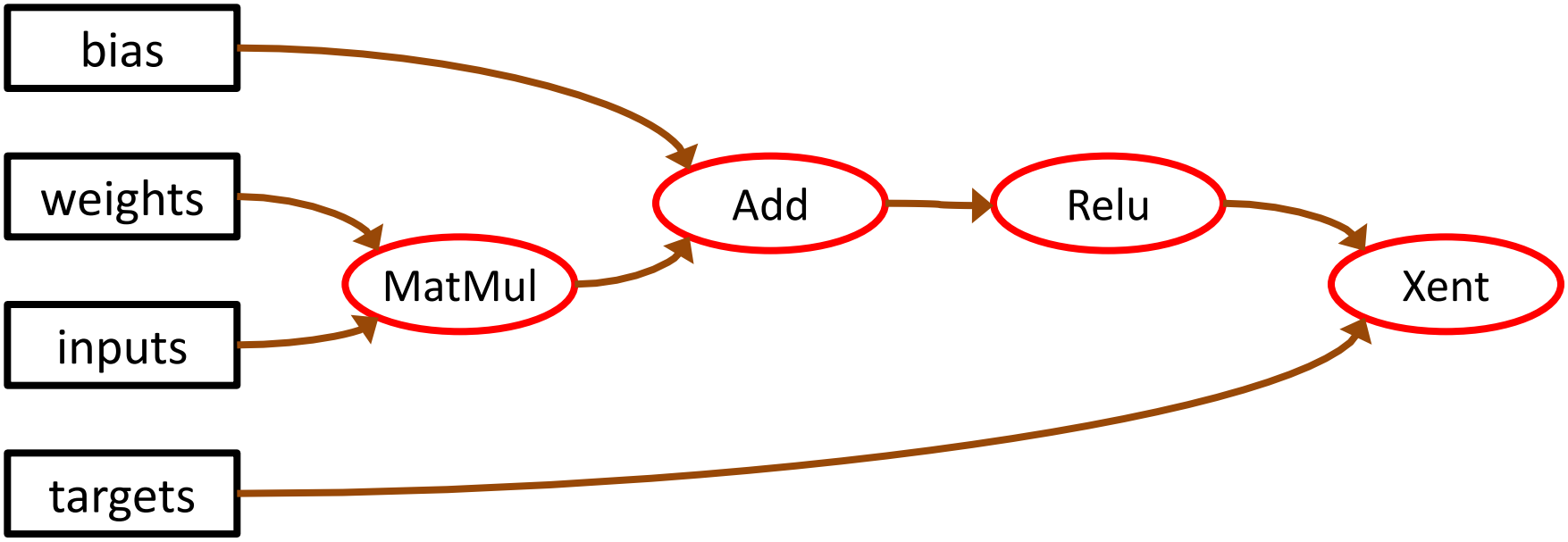
Computation is a Dataflow Graph

Graph of Nodes,
also called Operations or ops.



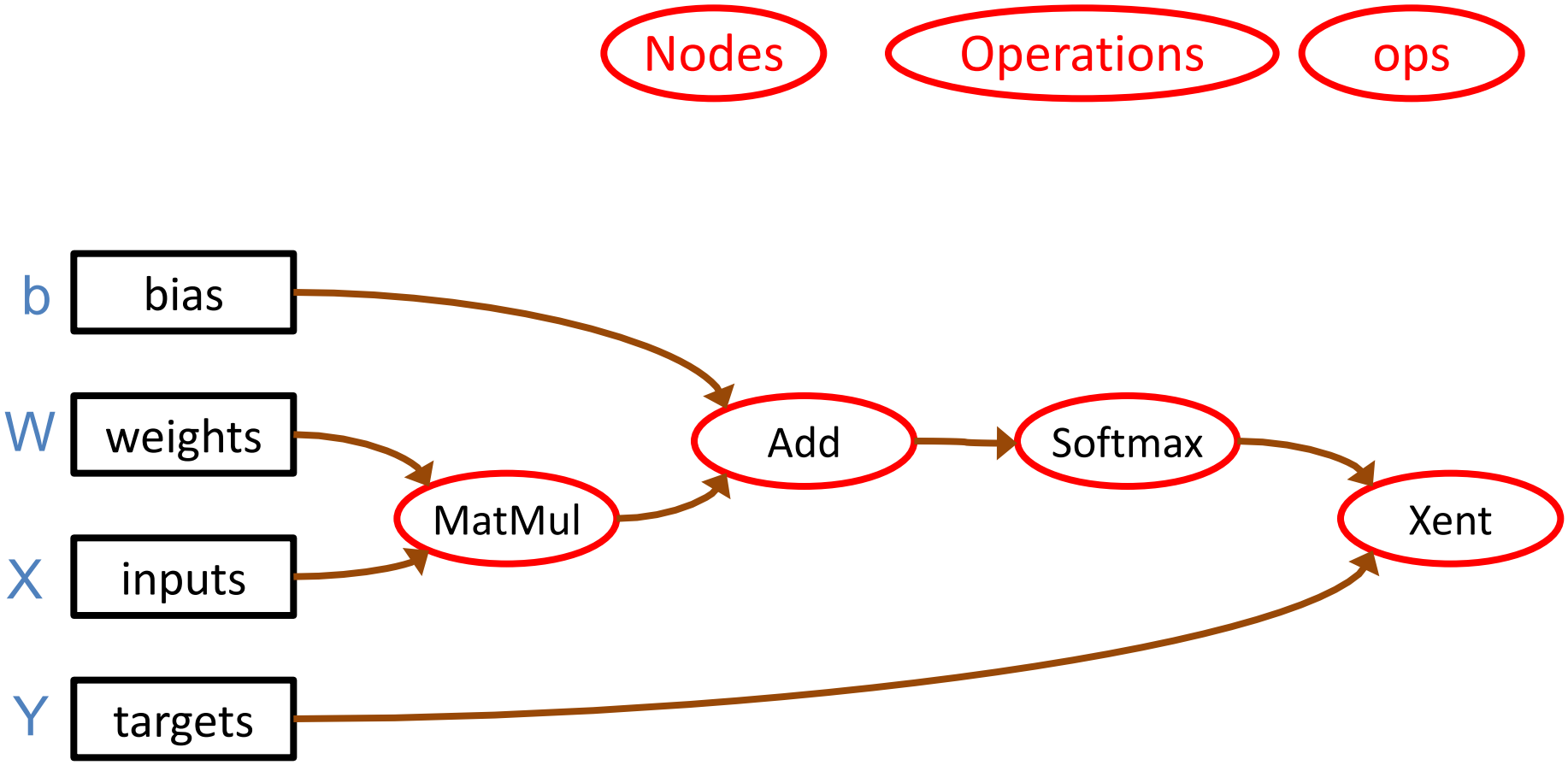
Computation is a Dataflow Graph

Edges are N-dimensional arrays: **Tensors**



Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Logistic Regression as Dataflow Graph

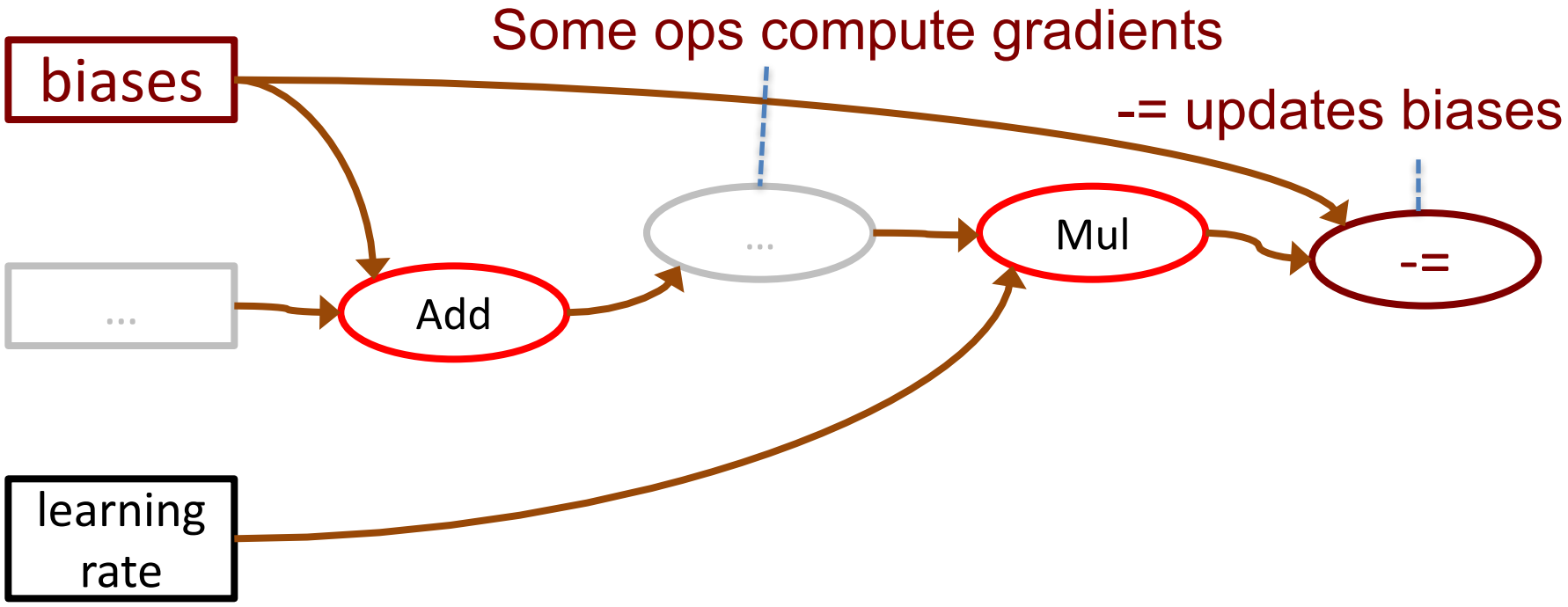


Edges are N-dimensional arrays: **Tensors**

Computation is a Dataflow Graph

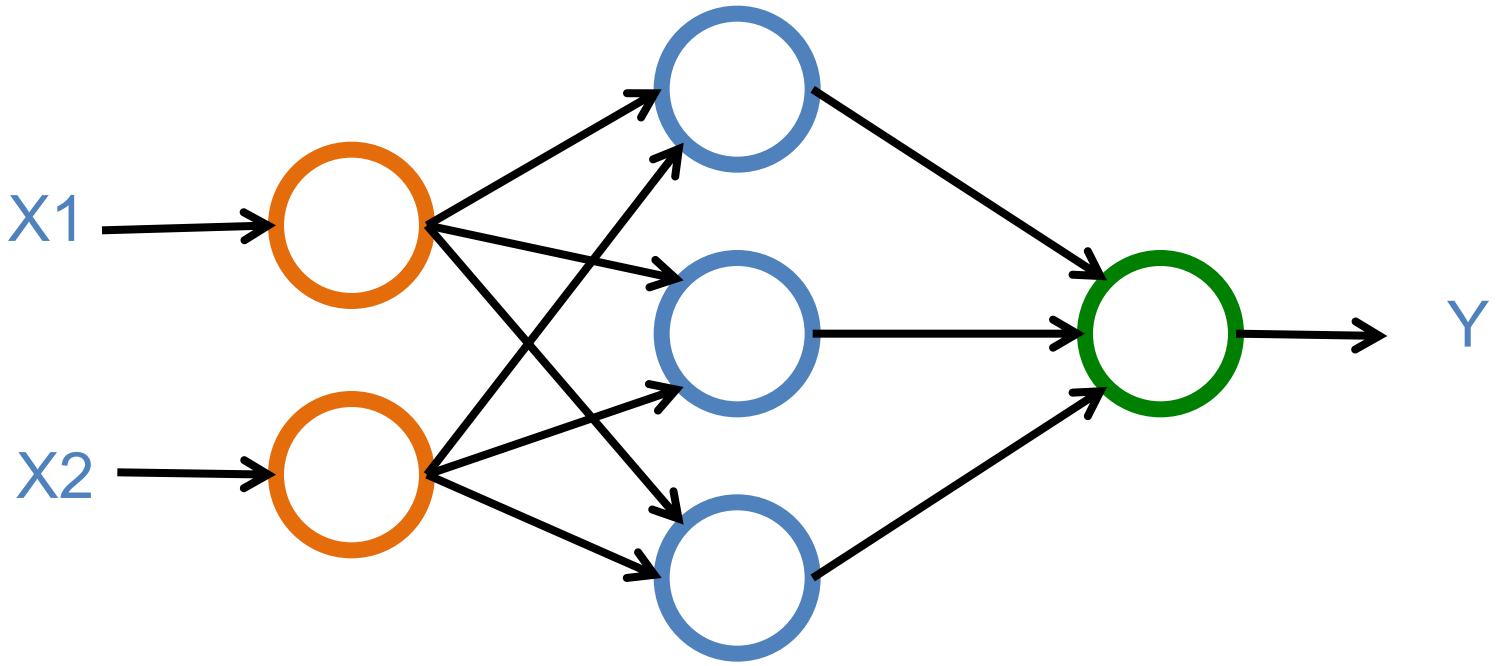
with state

‘Biases’ is a variable



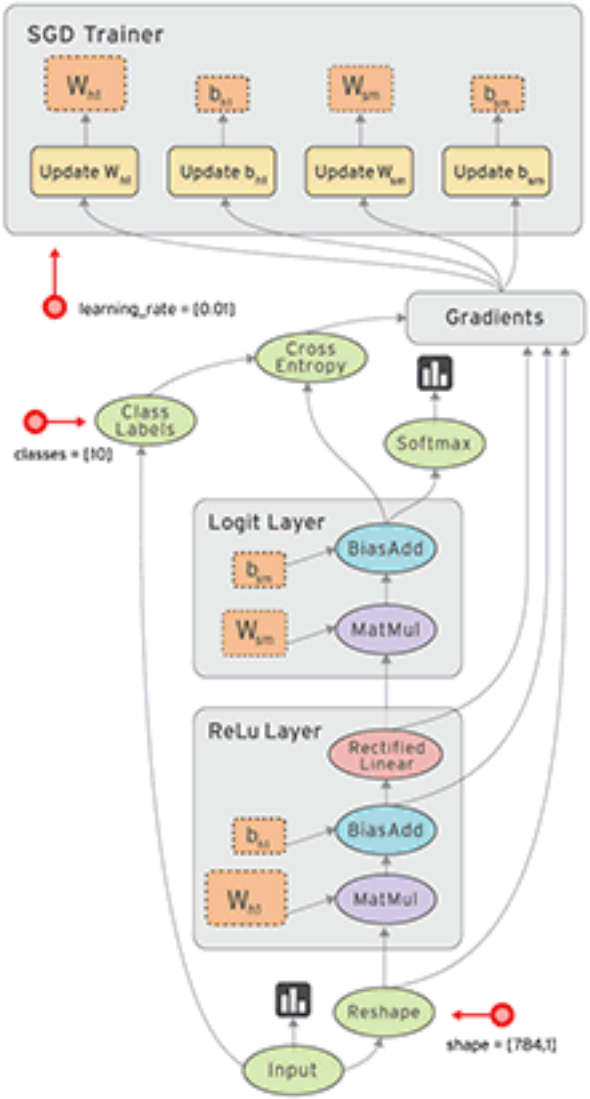
Neural Networks

Input Layer (X) Hidden Layer (H) **Output Layer (Y)**



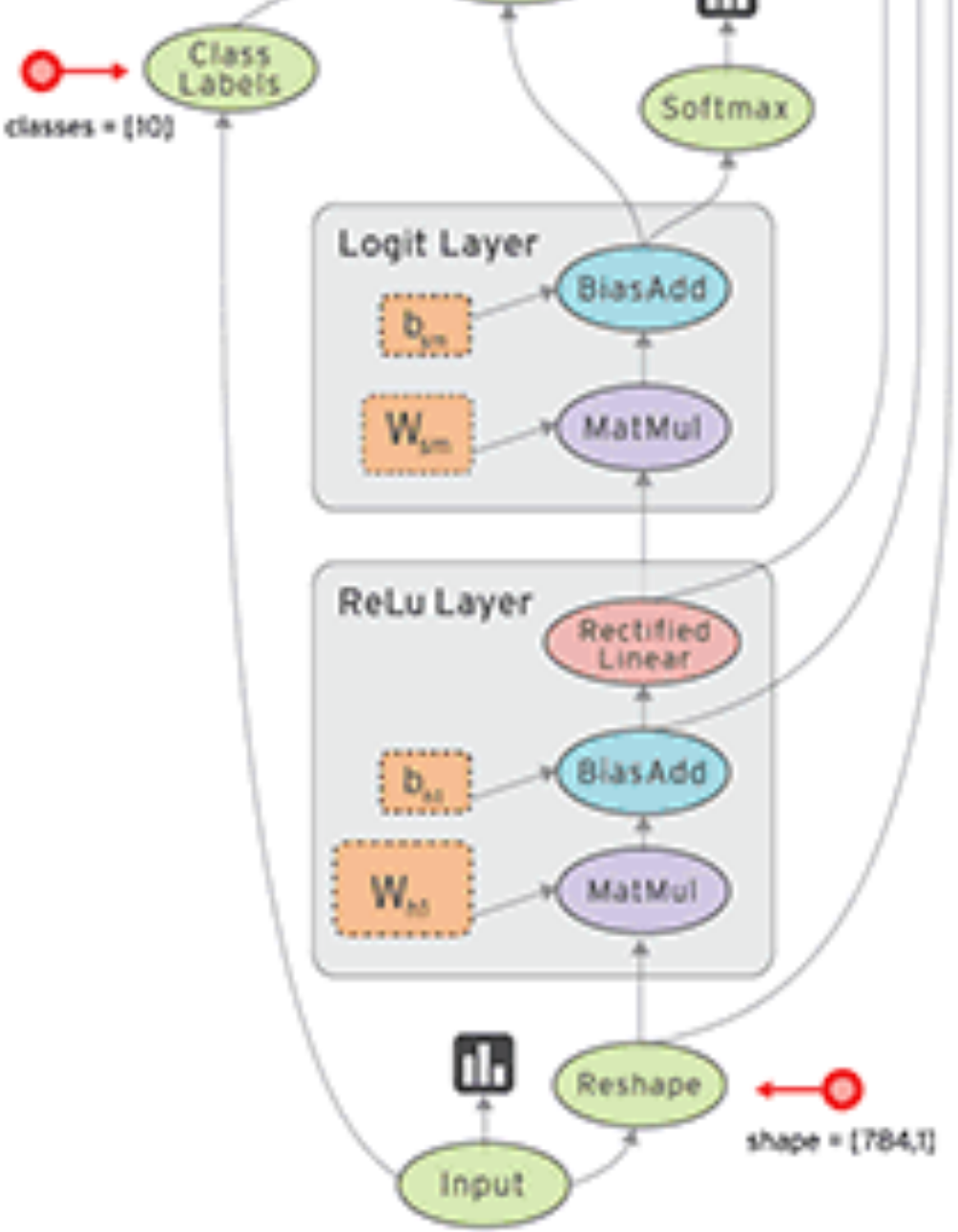
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Data Flow Graph



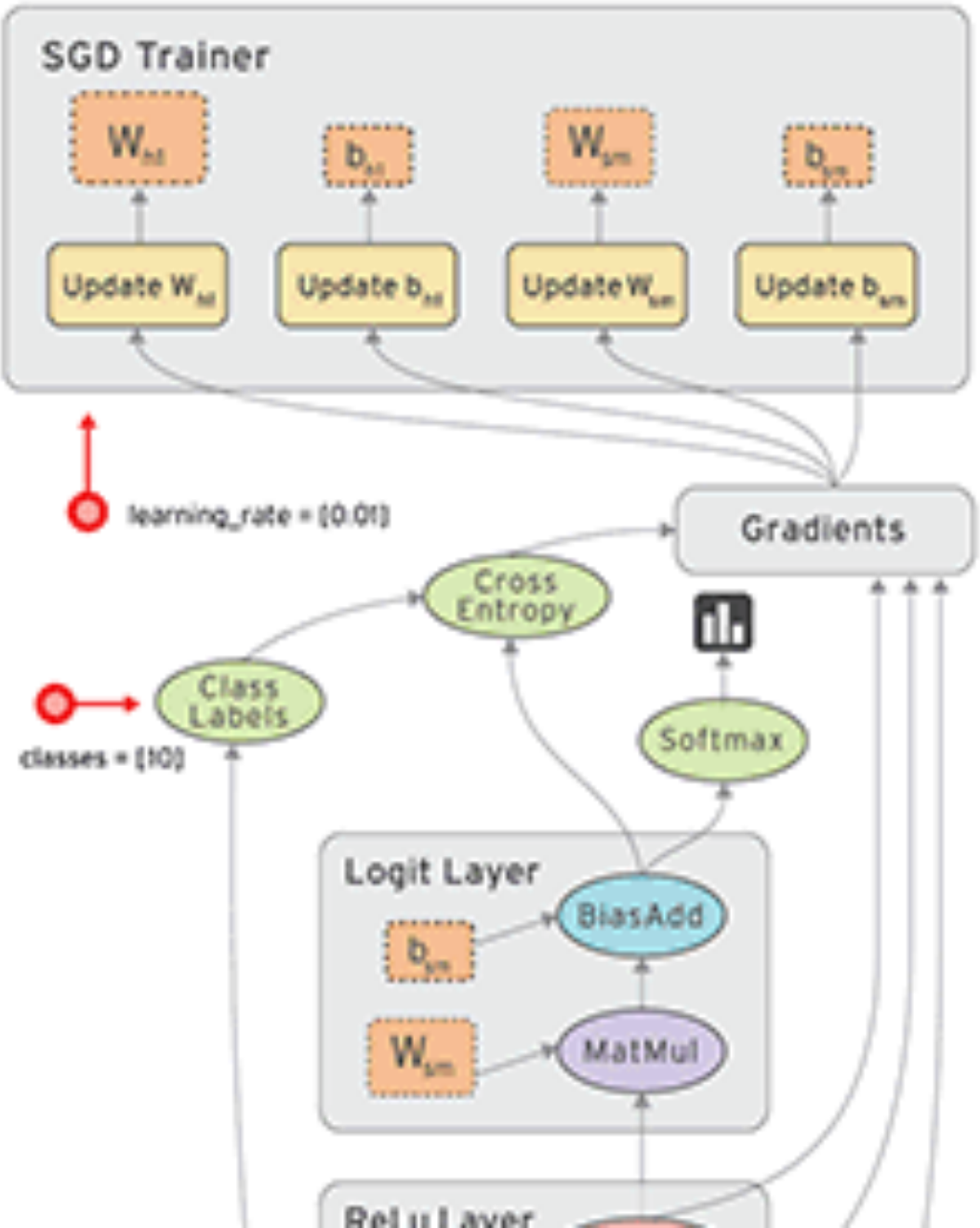
Source: https://www.tensorflow.org/

Data Flow Graph

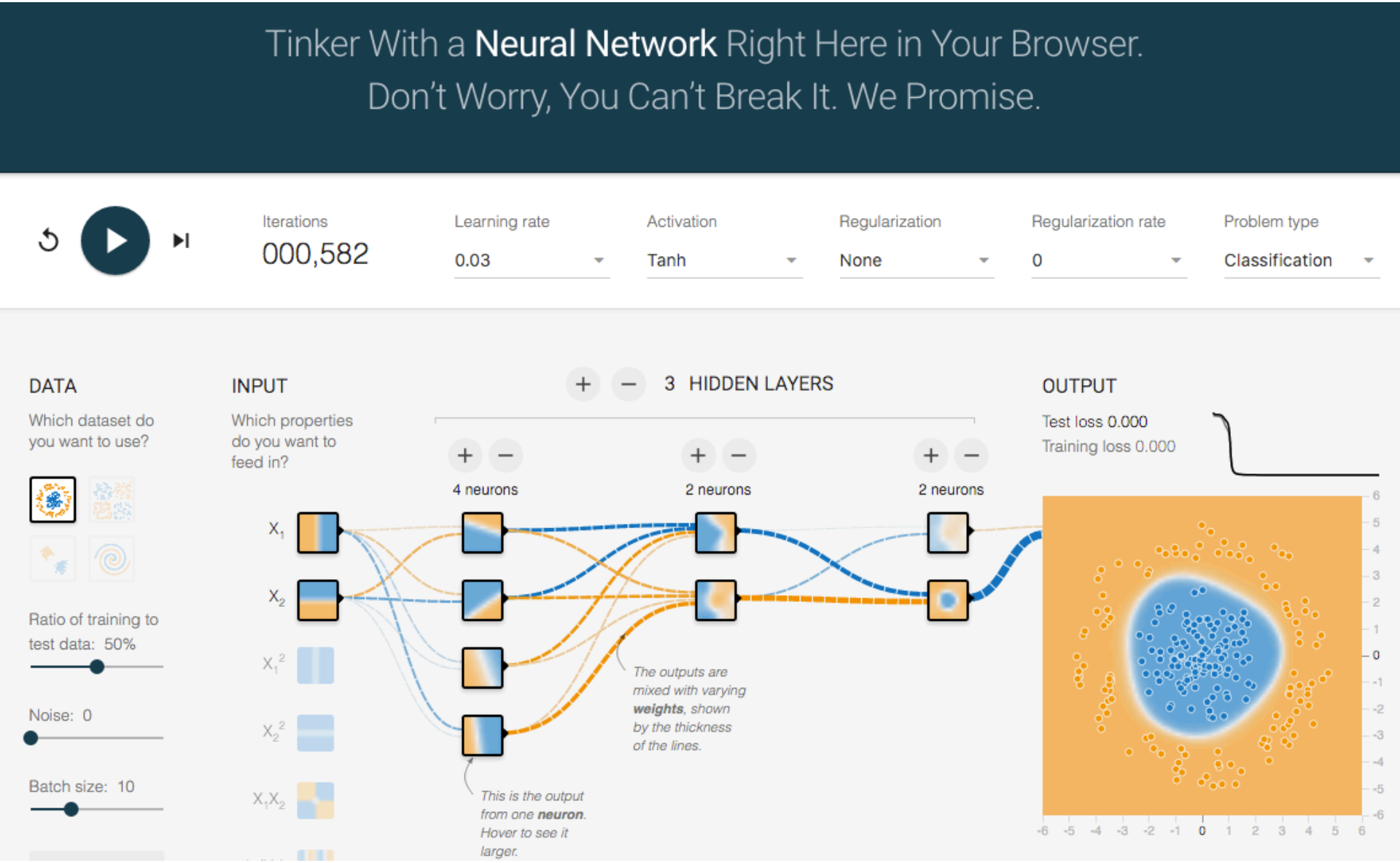


Source: https://www.tensorflow.org/

Data Flow Graph

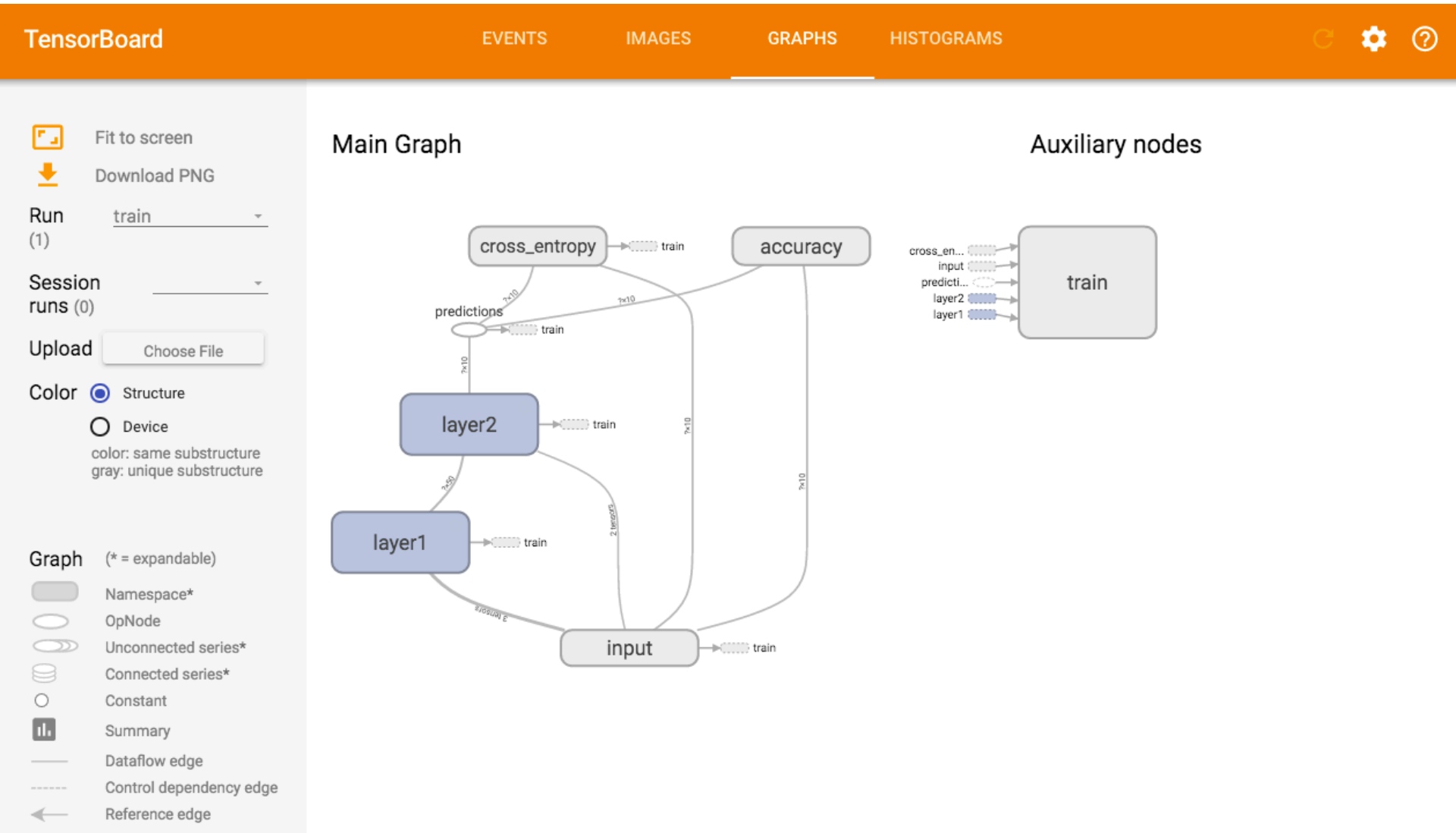


TensorFlow Playground



http://playground.tensorflow.org/

TensorBoard



https://www.tensorflow.org/tensorboard/index.html#graphs

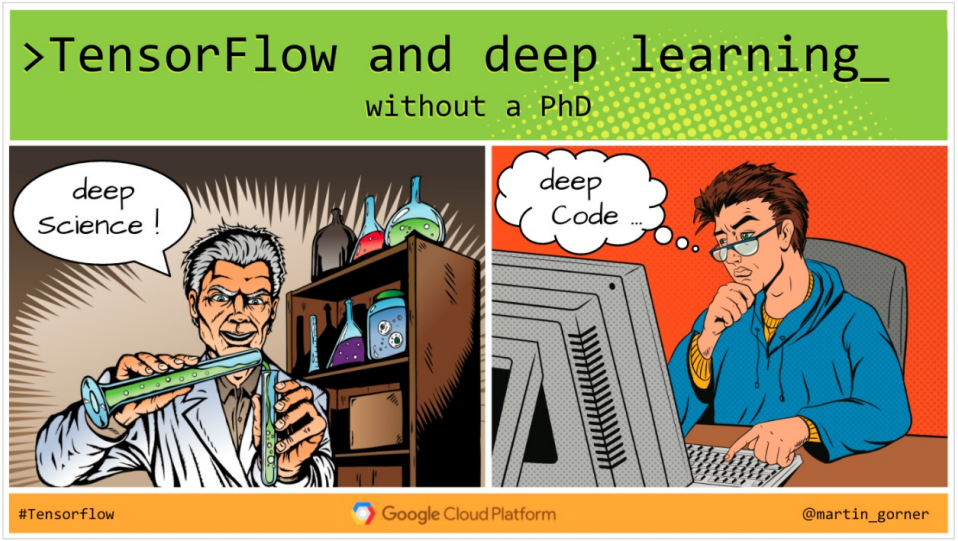
TensorFlow and Deep Learning

TensorFlow and Deep Learning

Did you find a mistake? Please file a bug.

TensorFlow and deep learning, without a PhD — 149 min remaining

1. Overview



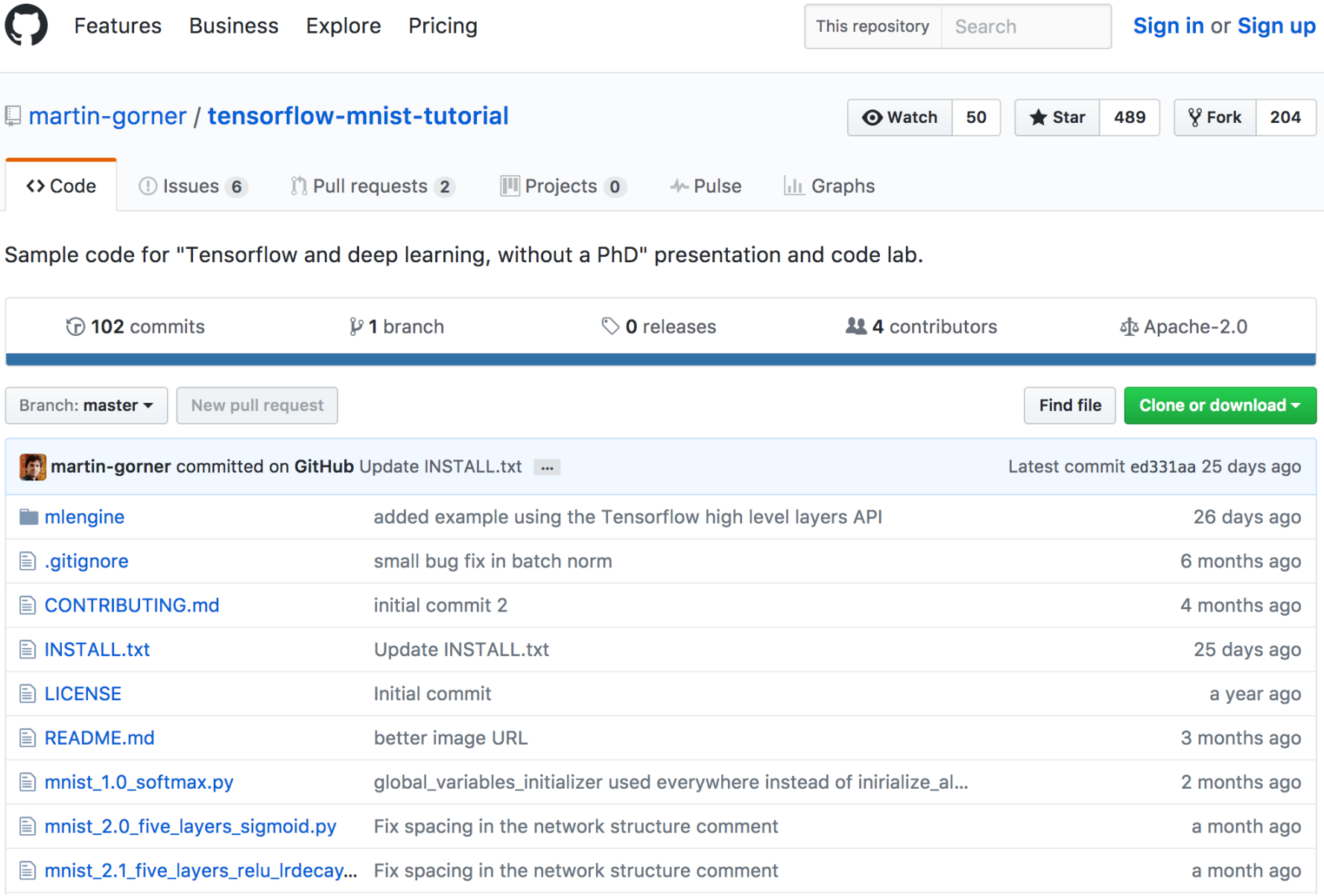
In this codelab, you will learn how to build and train a neural network that recognises handwritten digits. Along the way, as you enhance your neural network to achieve 99% accuracy, you will also discover the tools of the trade that deep learning professionals use to train their models efficiently.

This codelab uses the MNIST dataset, a collection of 60,000 labeled digits that has kept generations of PhDs busy for almost two decades. You will solve the problem with less than 100 lines of Python / TensorFlow code.

What you'll learn

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0

TensorFlow MNIST Tutorial



https://github.com/martin-gorner/tensorflow-mnist-tutorial/

TensorFlow and Deep Learning

- What is a neural network and how to train it
- How to build a basic 1-layer neural network using TensorFlow
- How to add more layers
- Training tips and tricks: overfitting, dropout, learning rate decay ...
- How to troubleshoot deep neural networks
- How to build convolutional networks

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0

TensorFlow

```
conda info --envs
```

```
conda --version
python --version
```

```
conda list
```

```
conda create -n tensorflow python=3.5
```

```
source activate tensorflow
```

```
activate tensorflow
```

Source: http://deeplearning.net/software/theano/install.html#anaconda

```
pip install Theano
```

```
conda install pydot
```

```
sudo pip install keras
```

```
pip install keras
```

```
pip install tensorflow
```

```
pip install ipython[all]
```

Source: http://deeplearning.net/software/theano/install.html#anaconda

pip install tensorflow

```
bash-3.2$ pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-1.1.0-cp36-cp36m-macosx_10_11_x86_64.whl (31.3MB)
    100% |████████████████████████████████| 31.3MB 23kB/s
Requirement already satisfied: wheel>=0.26 in ./anaconda/lib/python3.6/site-packages (from tensorflow)
Requirement already satisfied: six>=1.10.0 in ./anaconda/lib/python3.6/site-packages (from tensorflow)
Collecting protobuf>=3.2.0 (from tensorflow)
  Downloading protobuf-3.2.0-py2.py3-none-any.whl (360kB)
    100% |████████████████████████████████| 368kB 453kB/s
Requirement already satisfied: werkzeug>=0.11.10 in ./anaconda/lib/python3.6/site-packages (from tensorflow)
Requirement already satisfied: numpy>=1.11.0 in ./anaconda/lib/python3.6/site-packages (from tensorflow)
Requirement already satisfied: setuptools in ./anaconda/lib/python3.6/site-packages/setuptools-27.2.0-py3.6.
egg (from protobuf>=3.2.0->tensorflow)
Installing collected packages: protobuf, tensorflow
Successfully installed protobuf-3.2.0 tensorflow-1.1.0
bash-3.2$
```

TensorFlow MNIST Tutorial

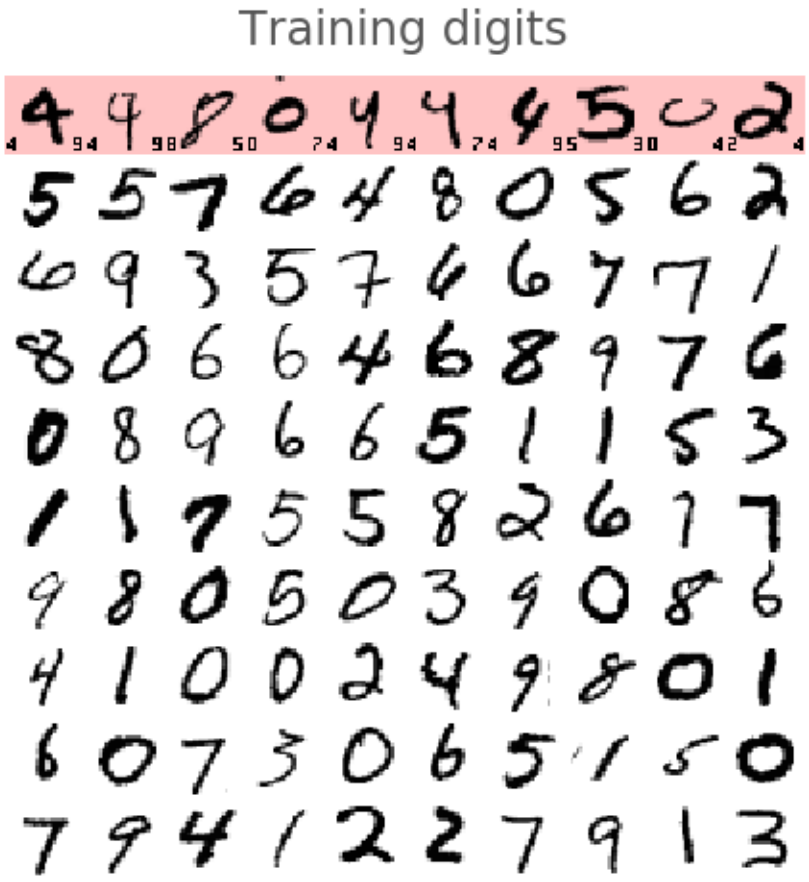
```
git clone https://github.com/martin-gorner/tensorflow-mnist-tutorial.git
```

```
cd tensorflow-mnist-tutorial
```

```
python3 mnist_1.0_softmax.py
```

Source: https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/INSTALL.txt

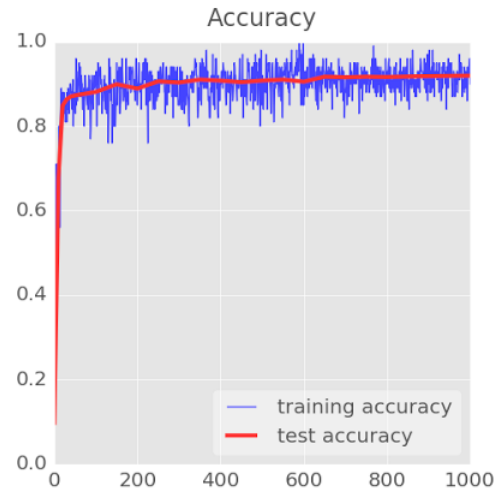
MNIST dataset: 60,000 labeled digits

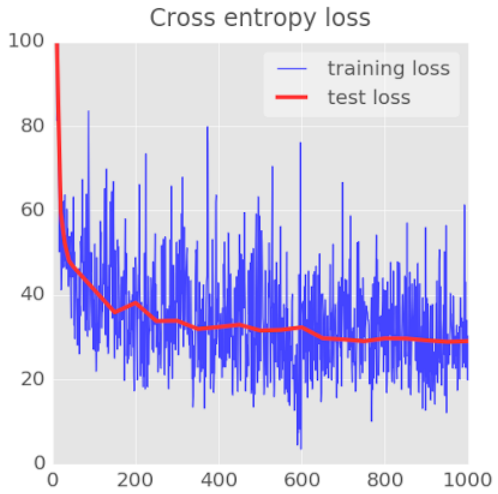


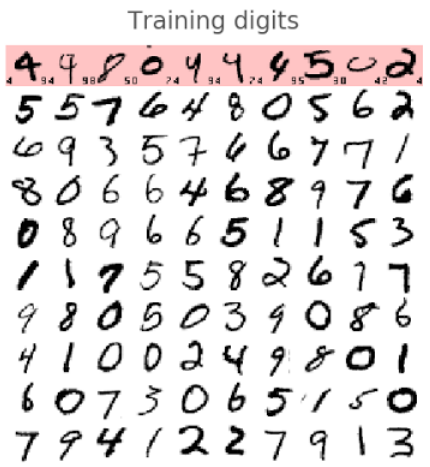
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0

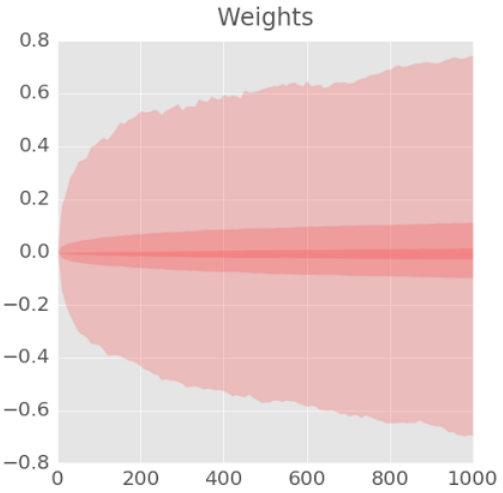
```
cd tensorflow-mnist-tutorial
```

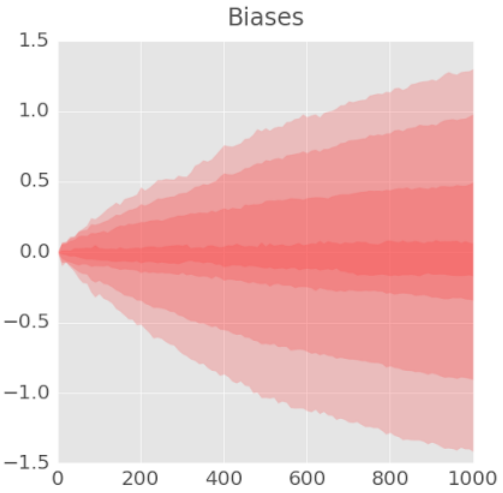
```
python3 mnist_1.0_softmax.py
```

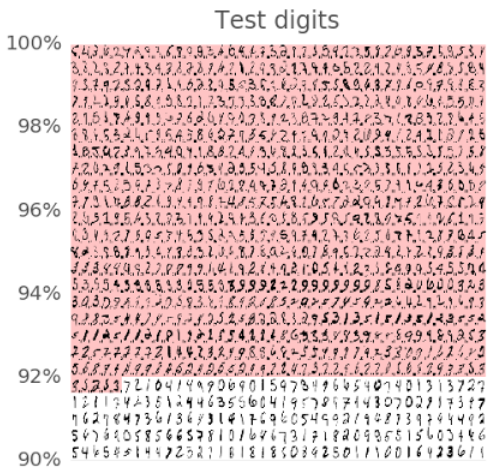












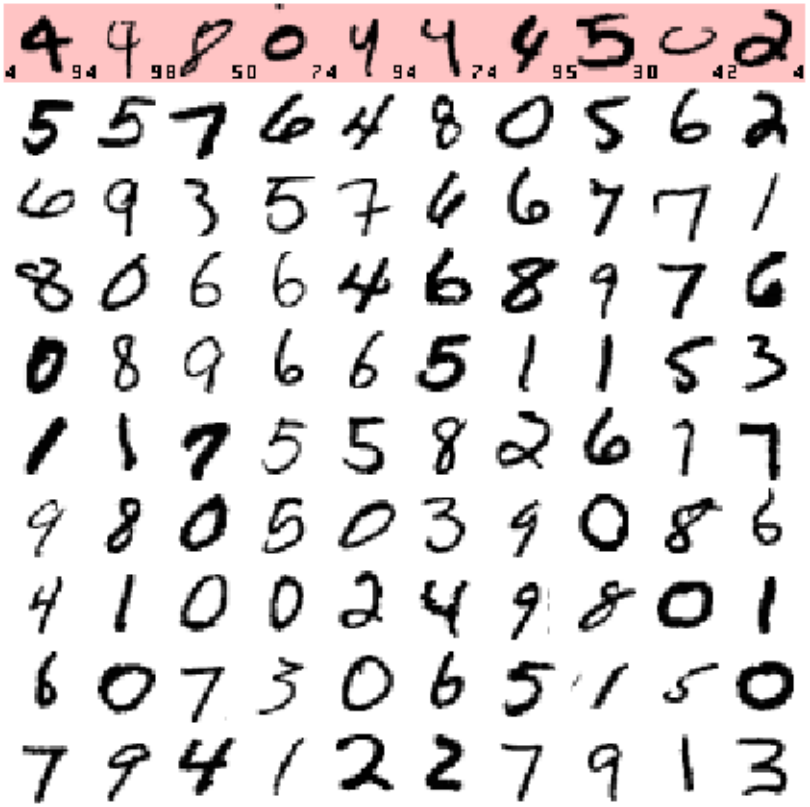
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#1

Train a Neural Network

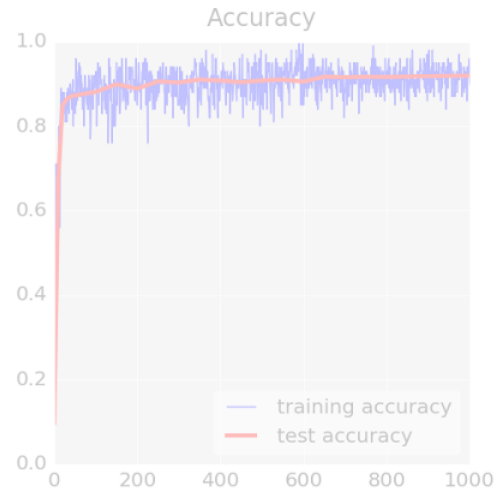
Training digits
updates to weights and biases =>
better recognition (loop)

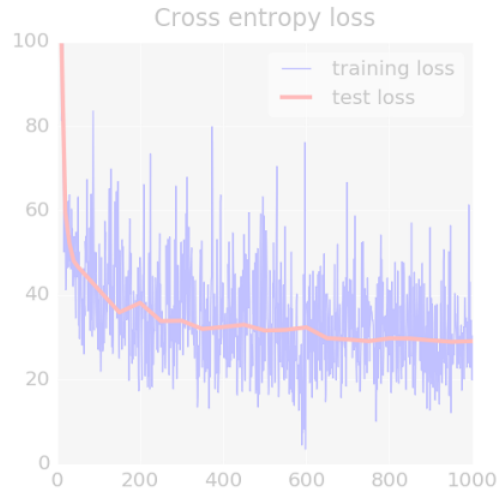
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2

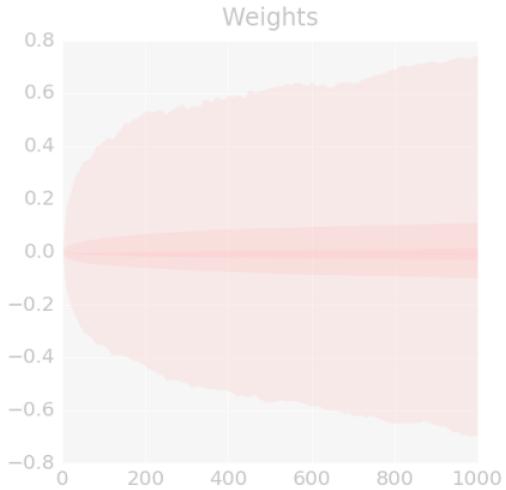
Training digits

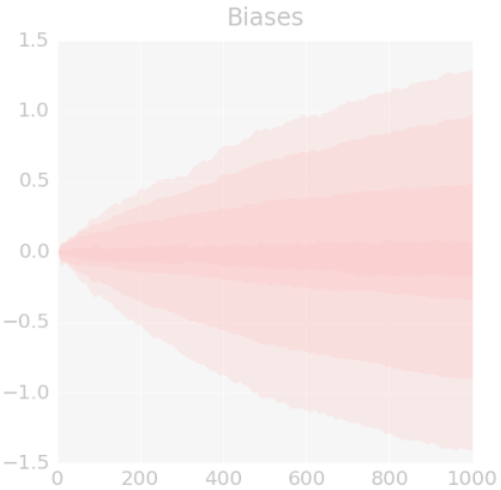


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2

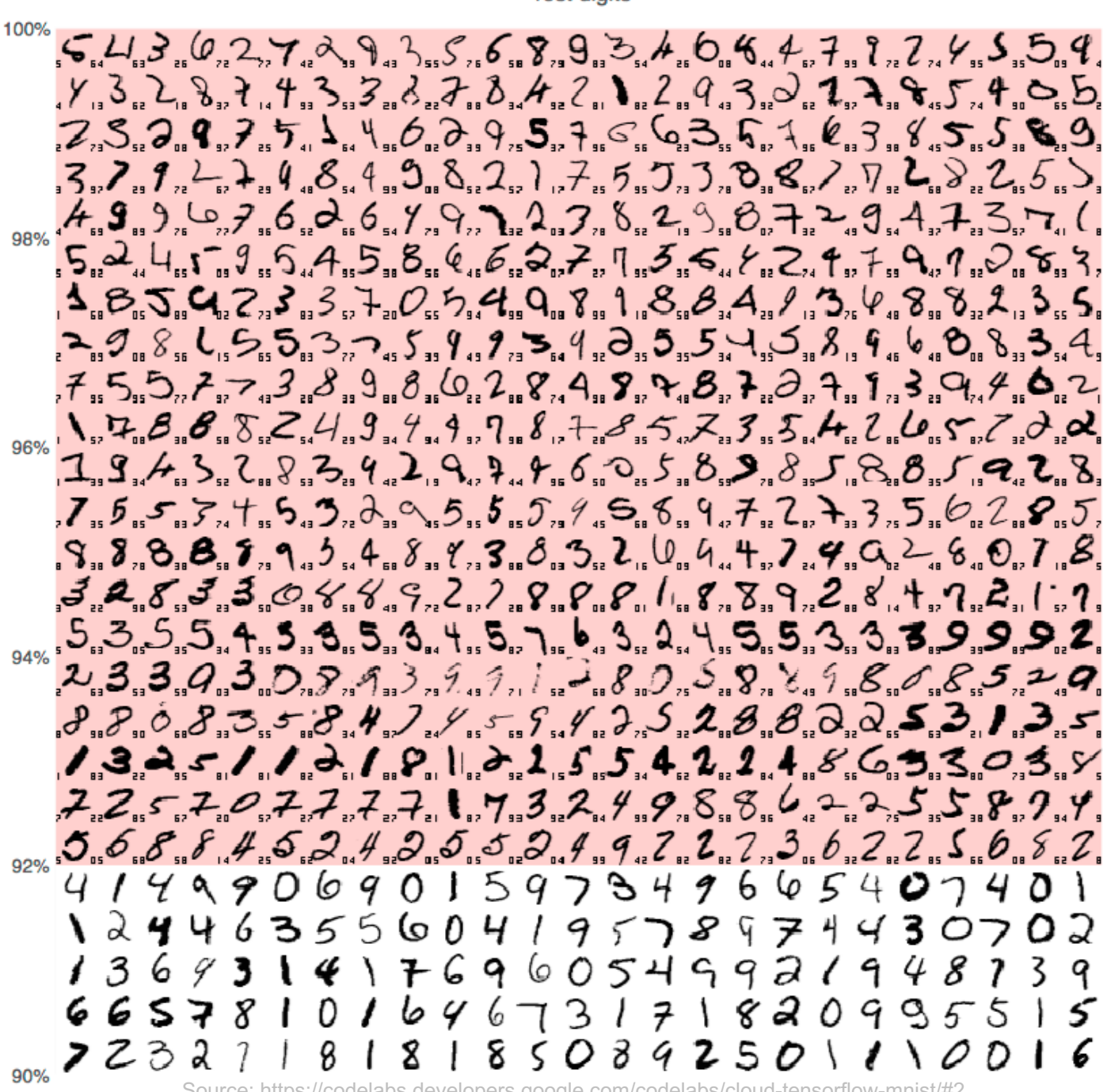




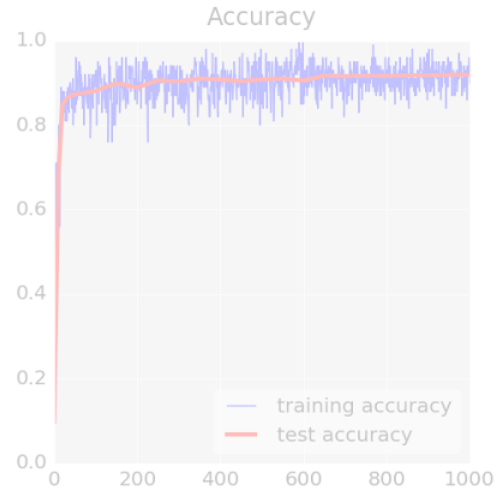


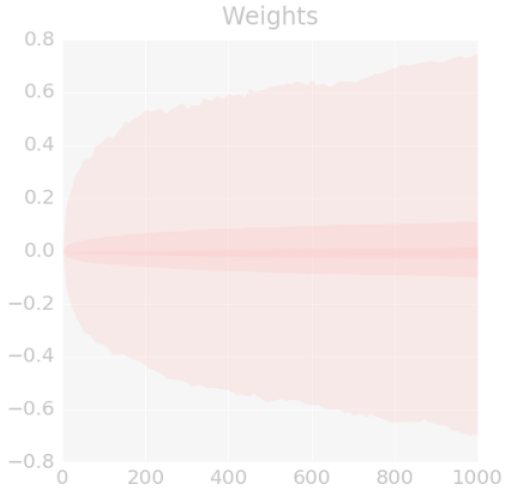


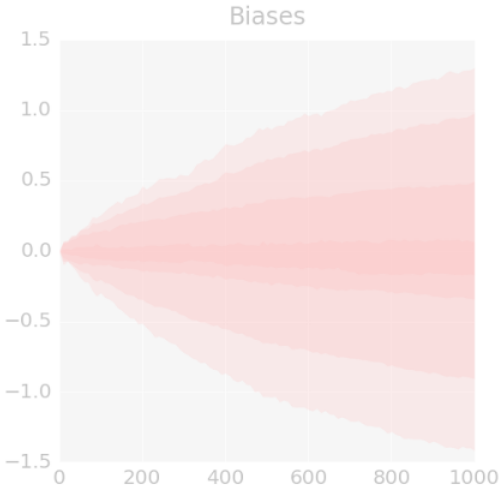
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2



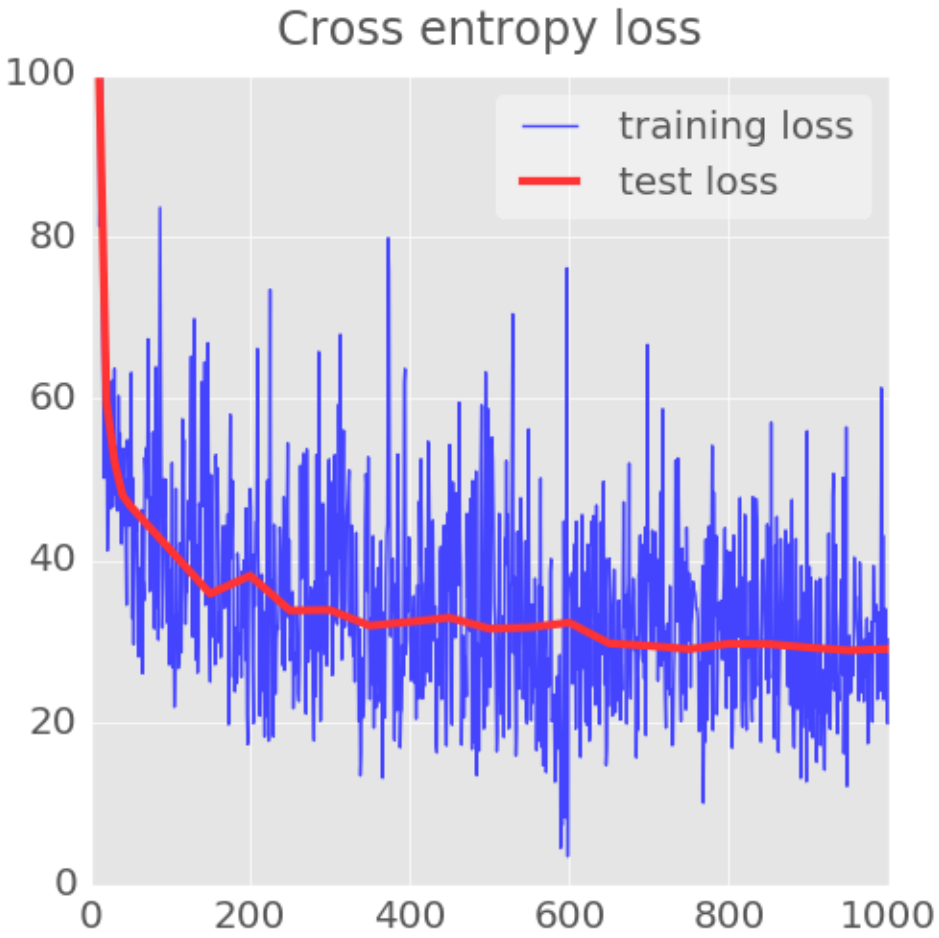
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2



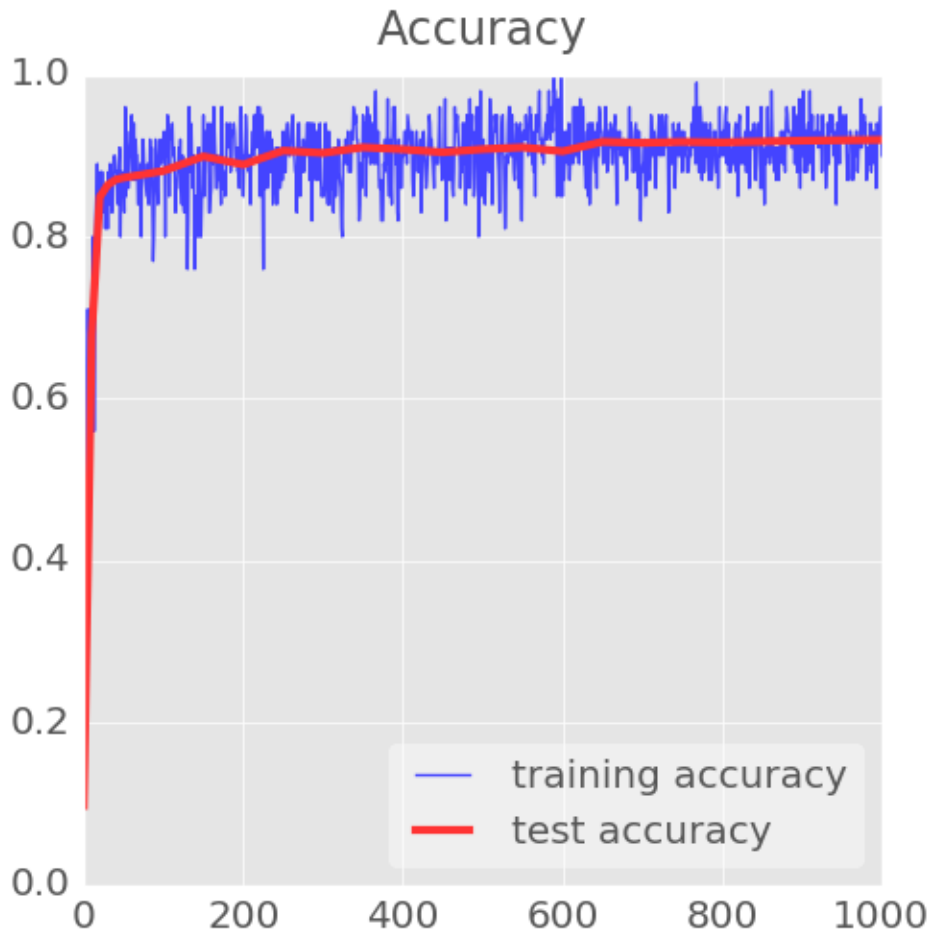




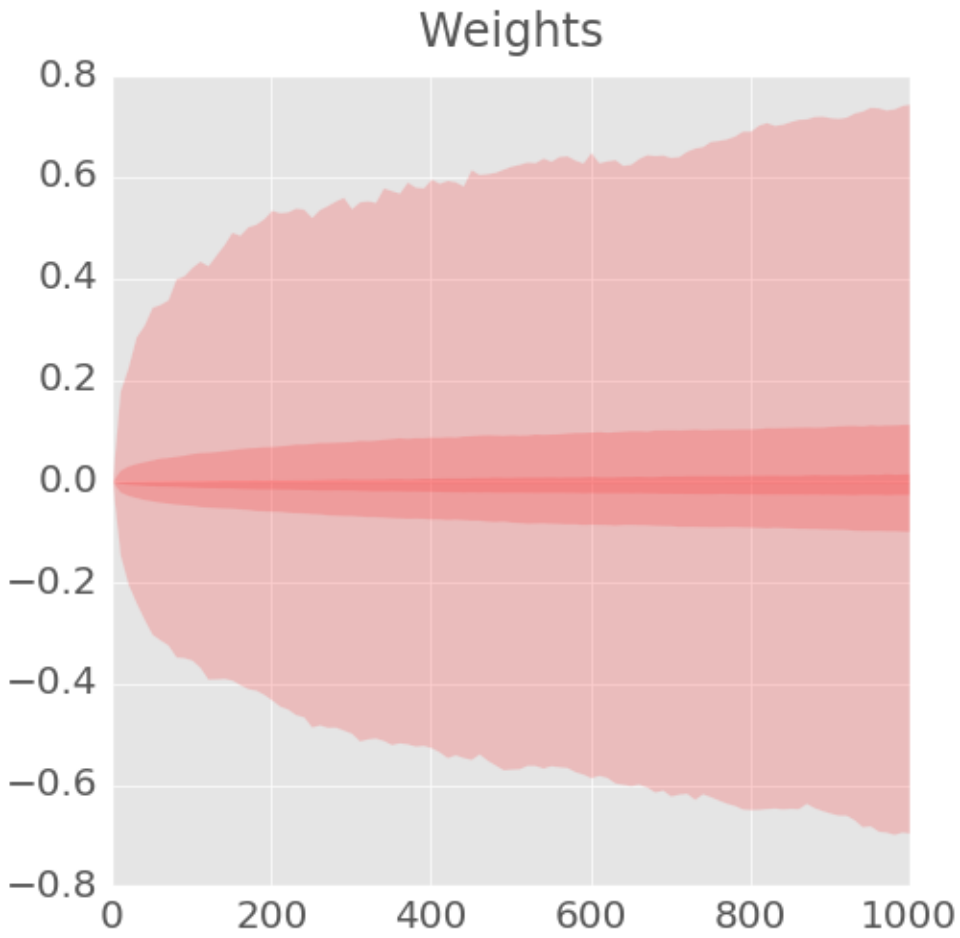
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2



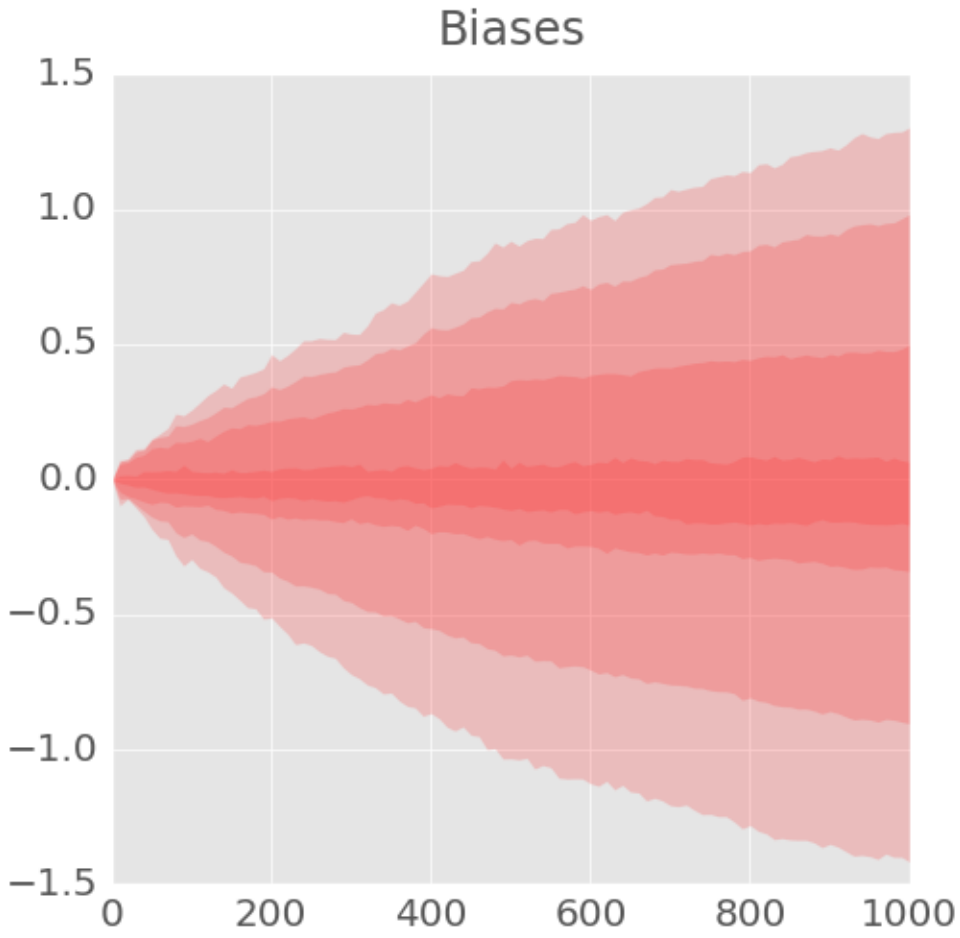
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2



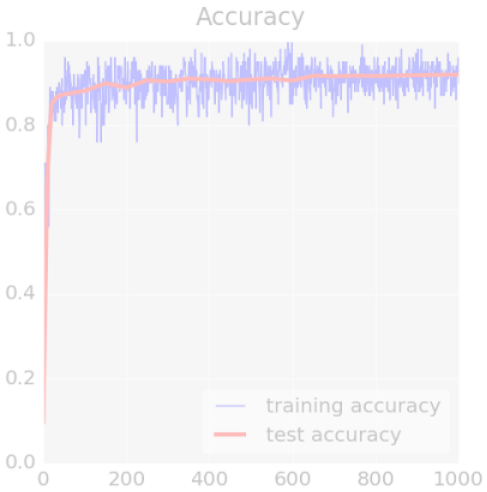
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2

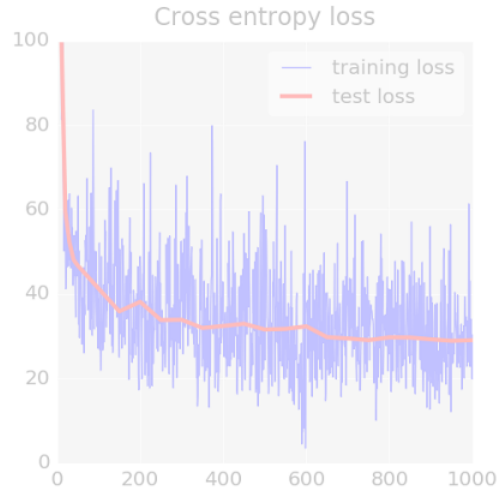


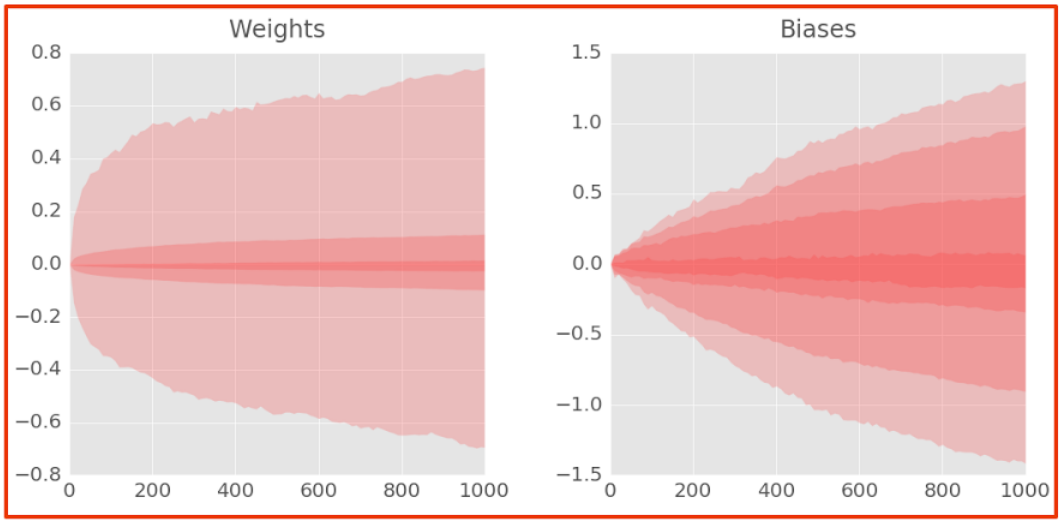
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2

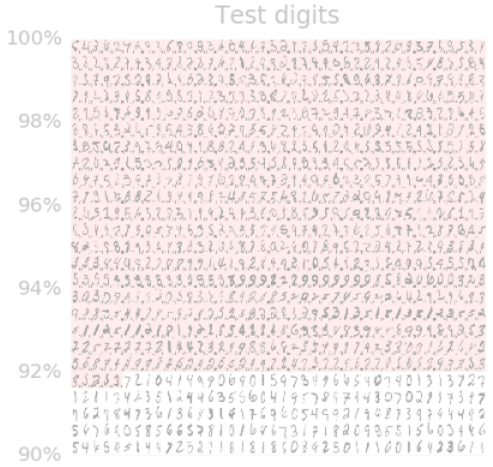


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2







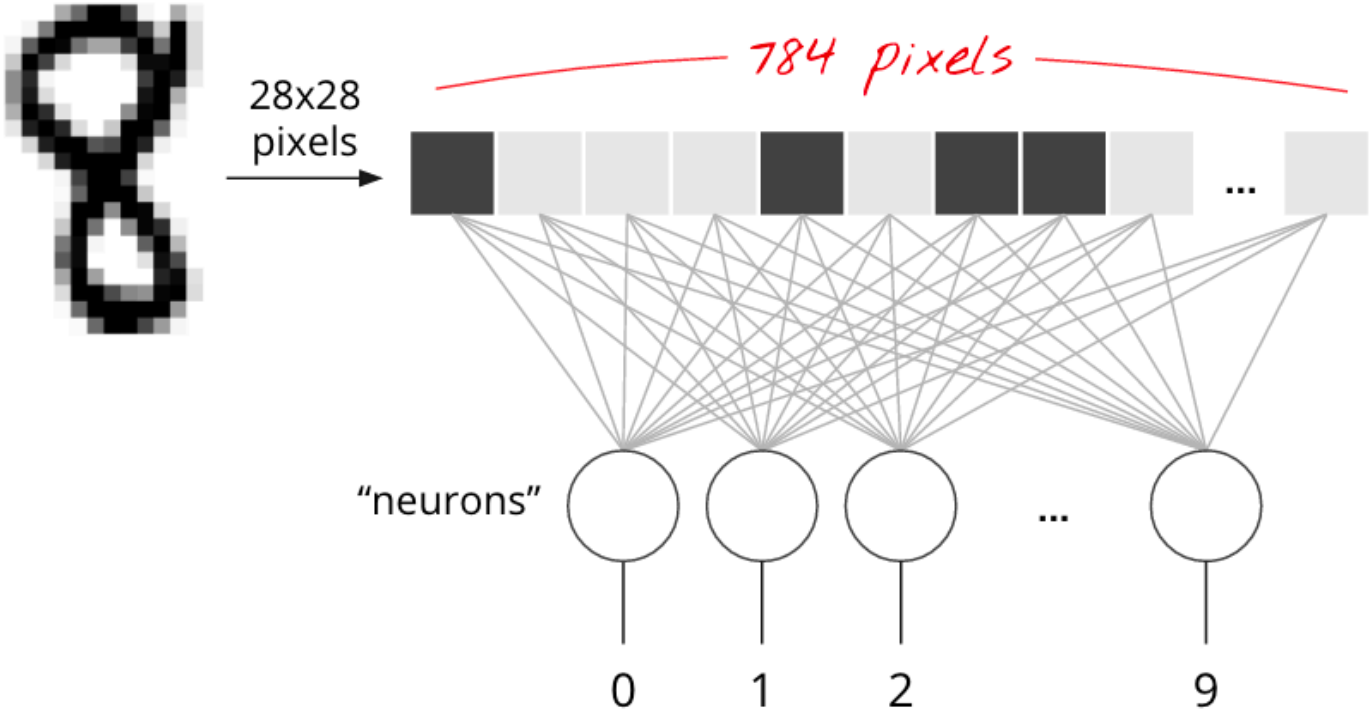


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#2

Cookbook

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Very Simple Model: Softmax Classification

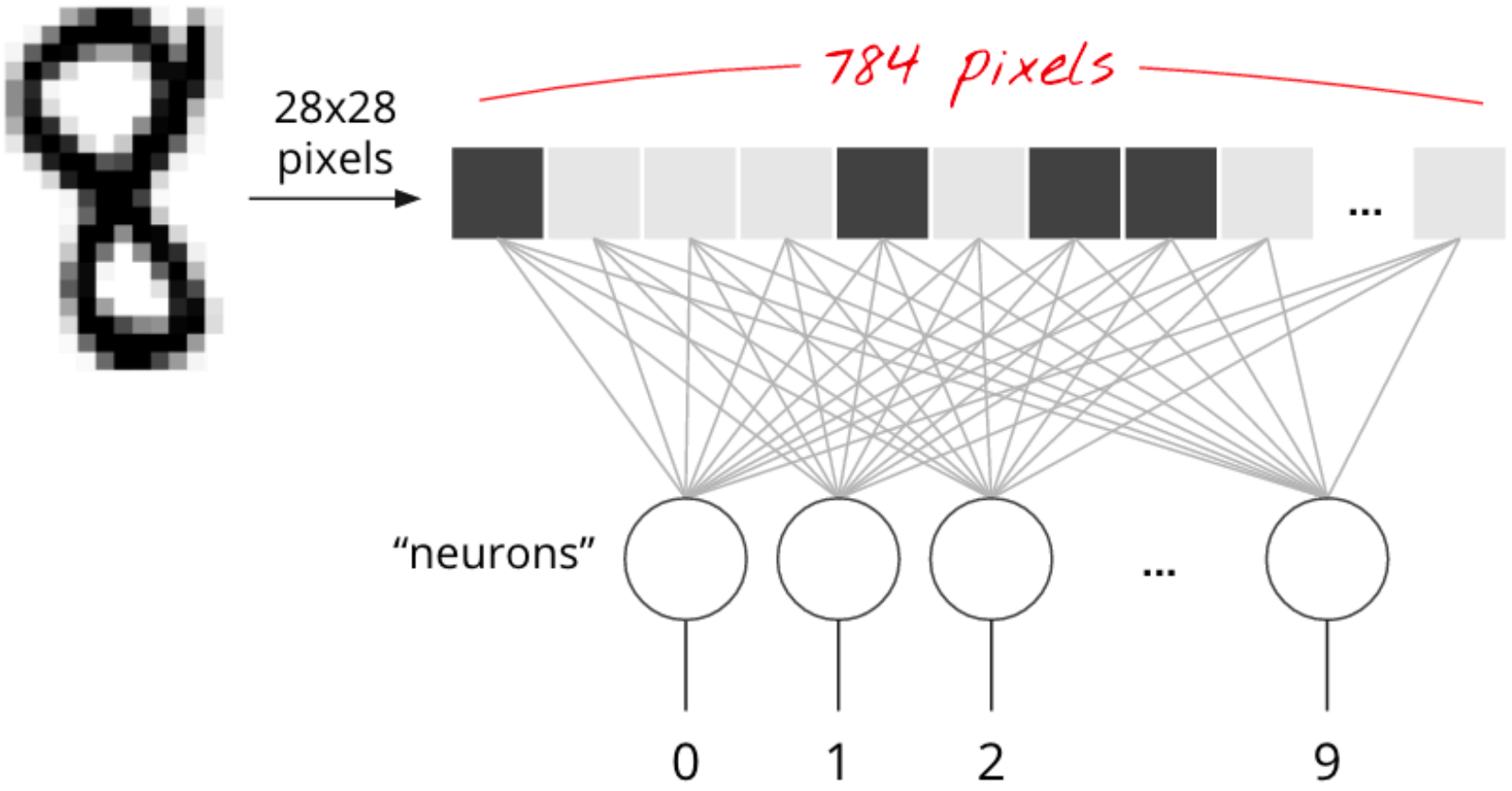


weighted sum of all pixels + bias

$$softmax(L_n) = \frac{e^{L_n}}{||e^L||}$$

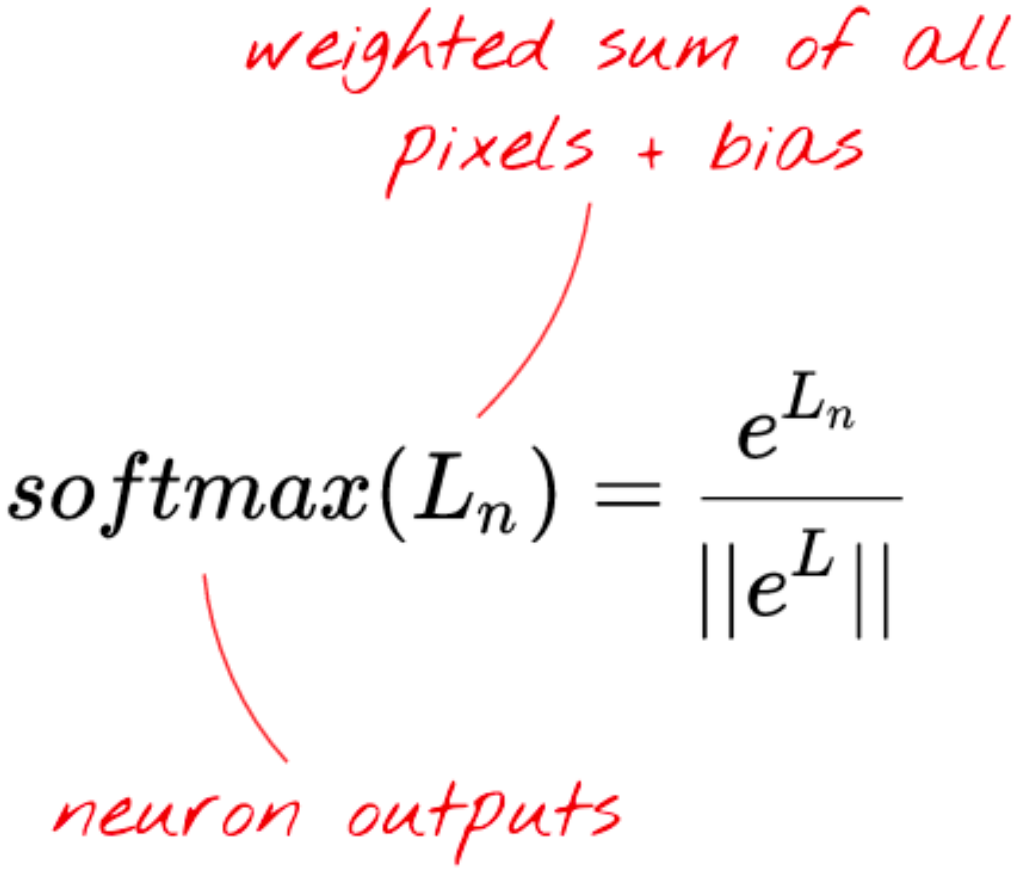
neuron outputs

Very Simple Model: Softmax Classification



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Very Simple Model: Softmax Classification



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

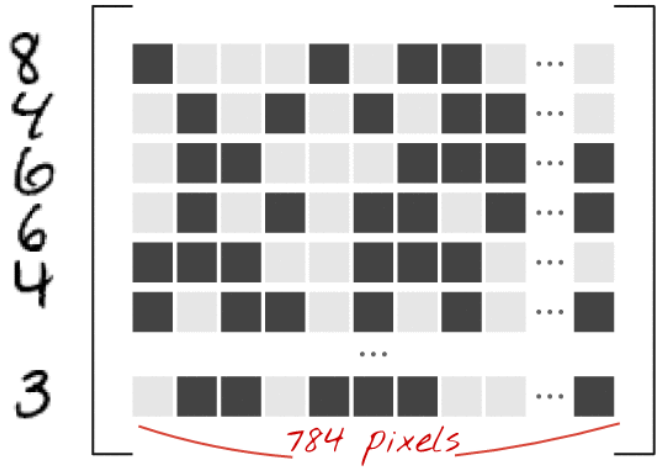
In Matrix notation, 100 images at a time

$$\begin{bmatrix} W_{0,0} & W_{0,1} & W_{0,2} & W_{0,3} & \cdots & W_{0,9} \\ W_{1,0} & W_{1,1} & W_{1,2} & W_{1,3} & \cdots & W_{1,9} \\ W_{2,0} & W_{2,1} & W_{2,2} & W_{2,3} & \cdots & W_{2,9} \\ W_{3,0} & W_{3,1} & W_{3,2} & W_{3,3} & \cdots & W_{3,9} \\ W_{4,0} & W_{4,1} & W_{4,2} & W_{4,3} & \cdots & W_{4,9} \\ W_{5,0} & W_{5,1} & W_{5,2} & W_{5,3} & \cdots & W_{5,9} \\ W_{6,0} & W_{6,1} & W_{6,2} & W_{6,3} & \cdots & W_{6,9} \\ W_{7,0} & W_{7,1} & W_{7,2} & W_{7,3} & \cdots & W_{7,9} \\ W_{8,0} & W_{8,1} & W_{8,2} & W_{8,3} & \cdots & W_{8,9} \\ & & \cdots & & & \\ W_{783,0} & W_{783,1} & W_{783,2} & & \cdots & W_{783,9} \end{bmatrix}$$

10 columns

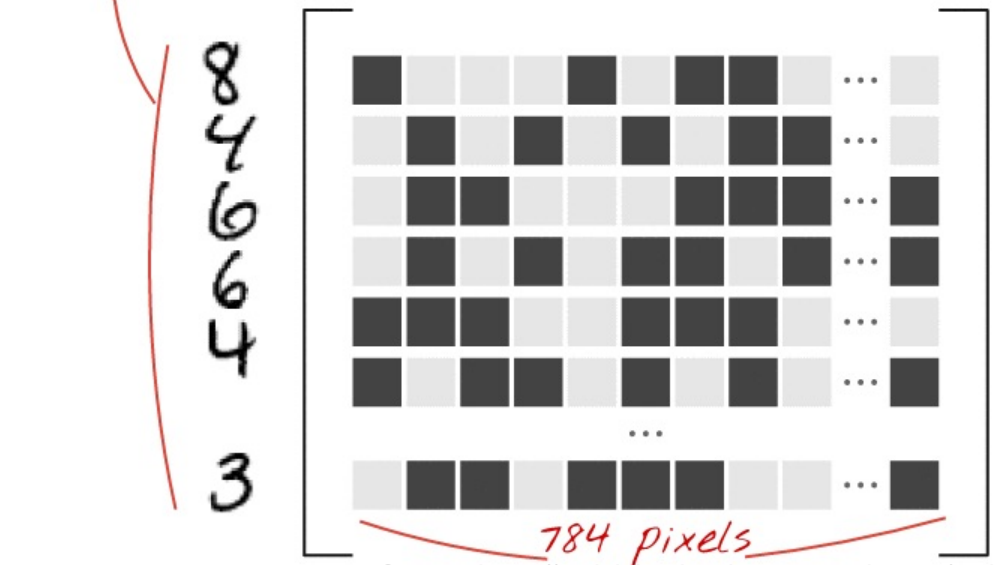
784 lines

X: 100 images, one per line, flattened



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

X: 100 images, one per line, flattened

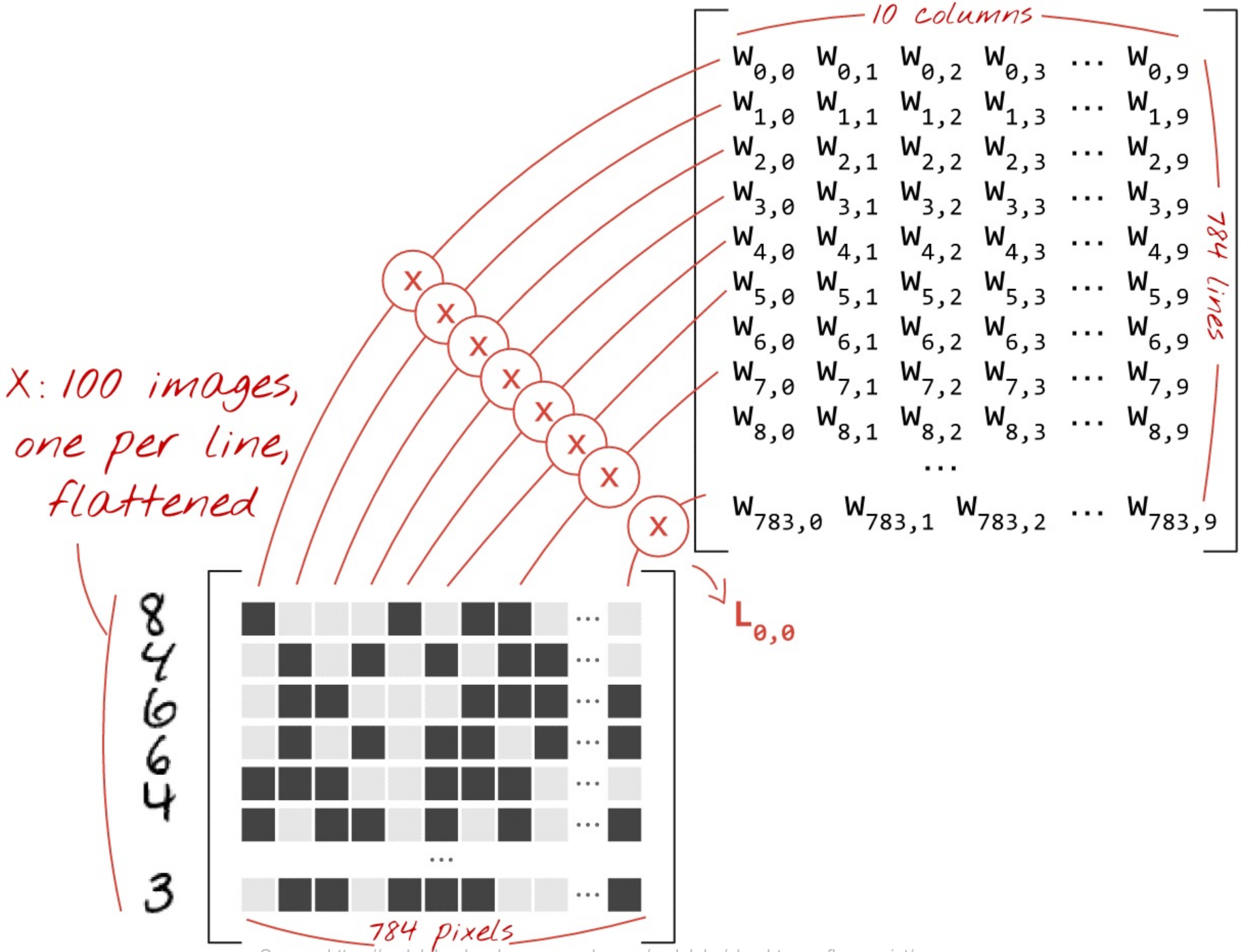


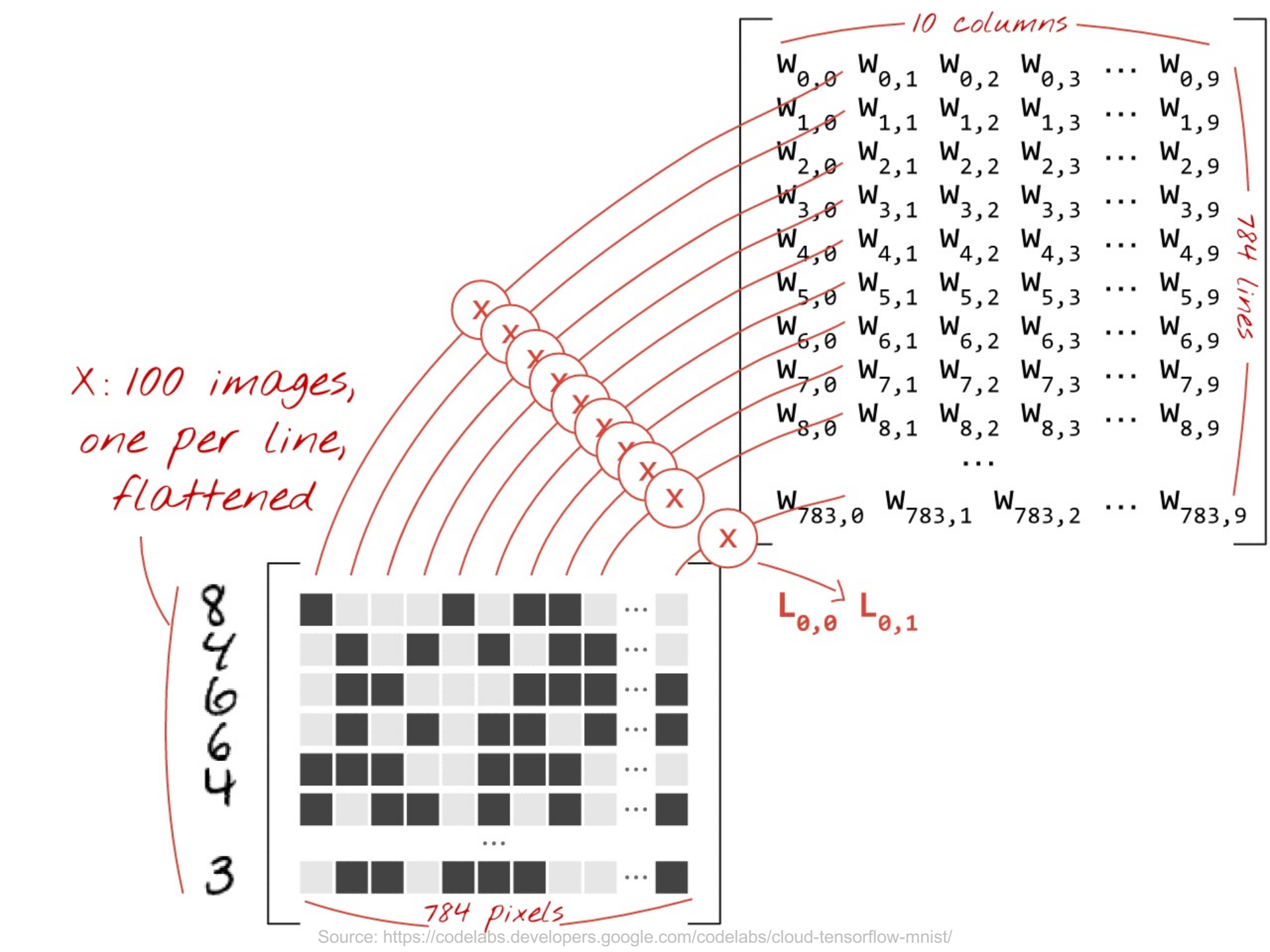
$$\begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & w_{0,3} & \dots & w_{0,9} \\ w_{1,0} & w_{1,1} & w_{1,2} & w_{1,3} & \dots & w_{1,9} \\ w_{2,0} & w_{2,1} & w_{2,2} & w_{2,3} & \dots & w_{2,9} \\ w_{3,0} & w_{3,1} & w_{3,2} & w_{3,3} & \dots & w_{3,9} \\ w_{4,0} & w_{4,1} & w_{4,2} & w_{4,3} & \dots & w_{4,9} \\ w_{5,0} & w_{5,1} & w_{5,2} & w_{5,3} & \dots & w_{5,9} \\ w_{6,0} & w_{6,1} & w_{6,2} & w_{6,3} & \dots & w_{6,9} \\ w_{7,0} & w_{7,1} & w_{7,2} & w_{7,3} & \dots & w_{7,9} \\ w_{8,0} & w_{8,1} & w_{8,2} & w_{8,3} & \dots & w_{8,9} \\ & & \dots & & & \\ w_{783,0} & w_{783,1} & w_{783,2} & & \dots & w_{783,9} \end{bmatrix}$$

10 columns

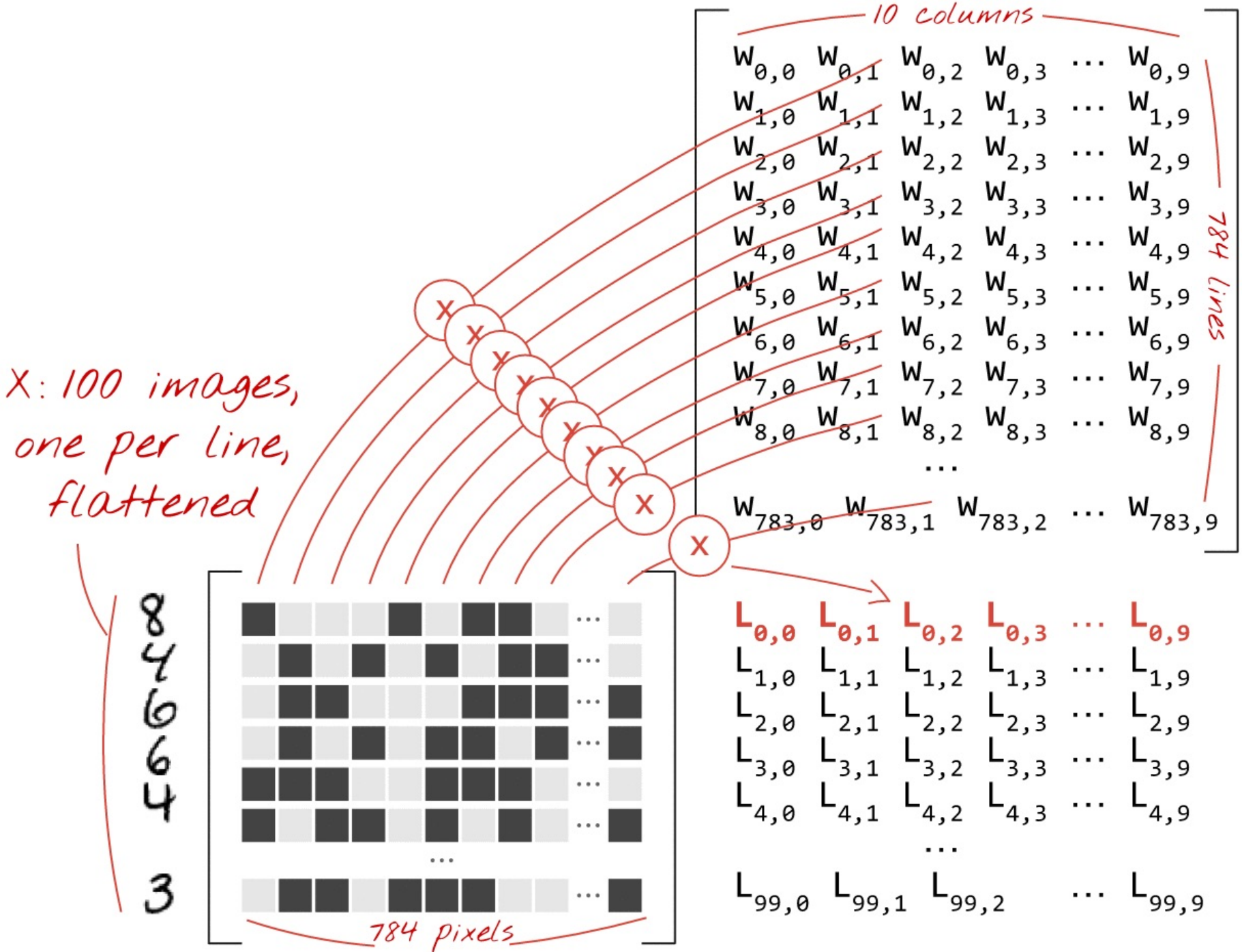
784 lines

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/


10 columns
784 lines
X: 100 images, one per line, flattened
784 pixels
$L_{0,0}$
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/



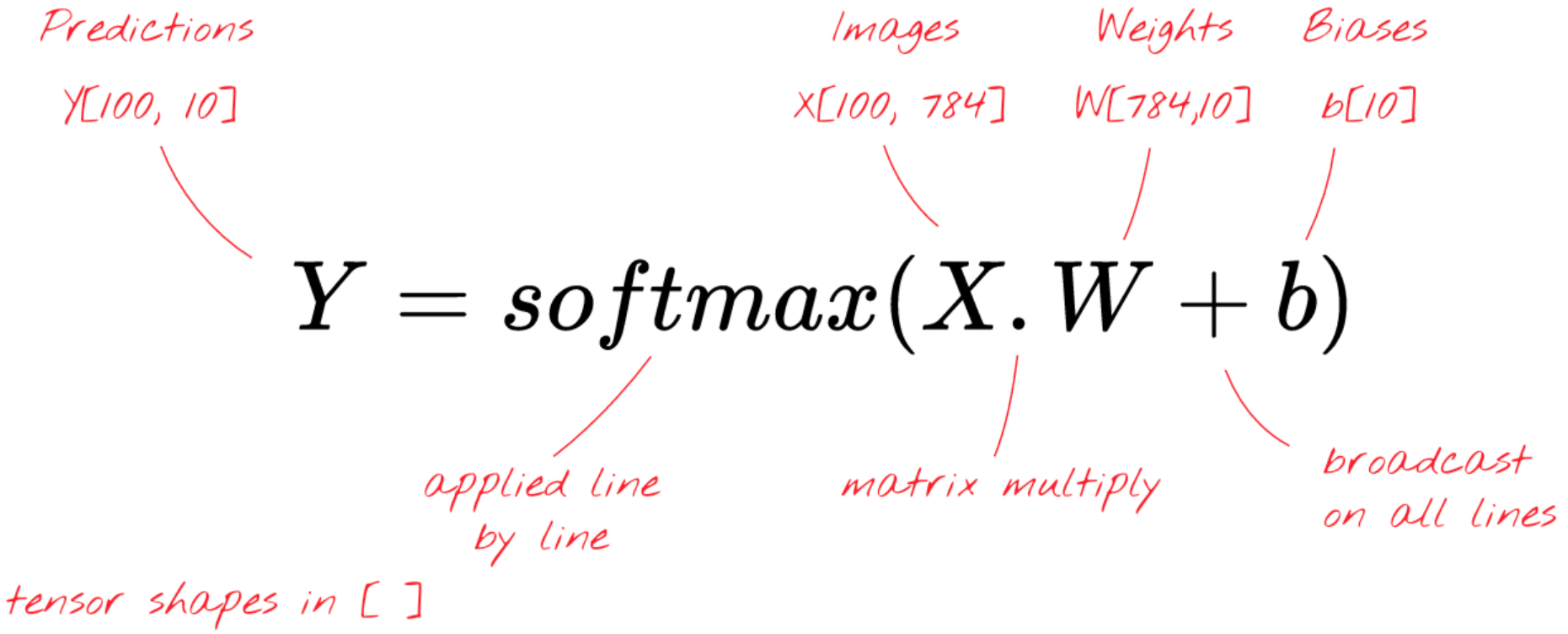
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/


10 columns
784 lines
X: 100 images, one per line, flattened
784 pixels
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

What are "weights" and "biases" ?
How is the "cross-entropy" computed ?
How exactly does the training algorithm work ?

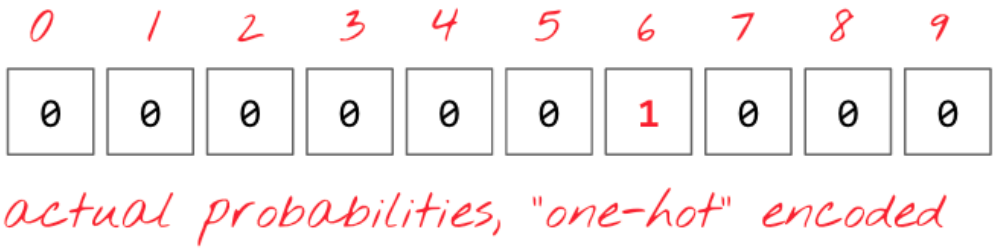
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Y = f(X)

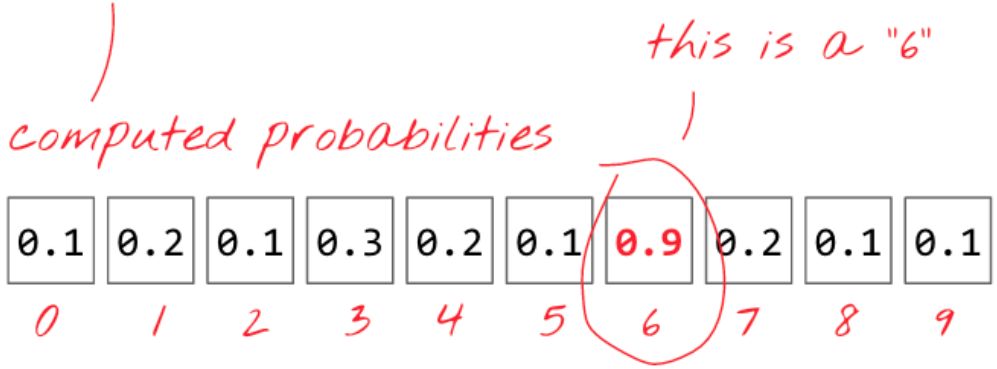


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

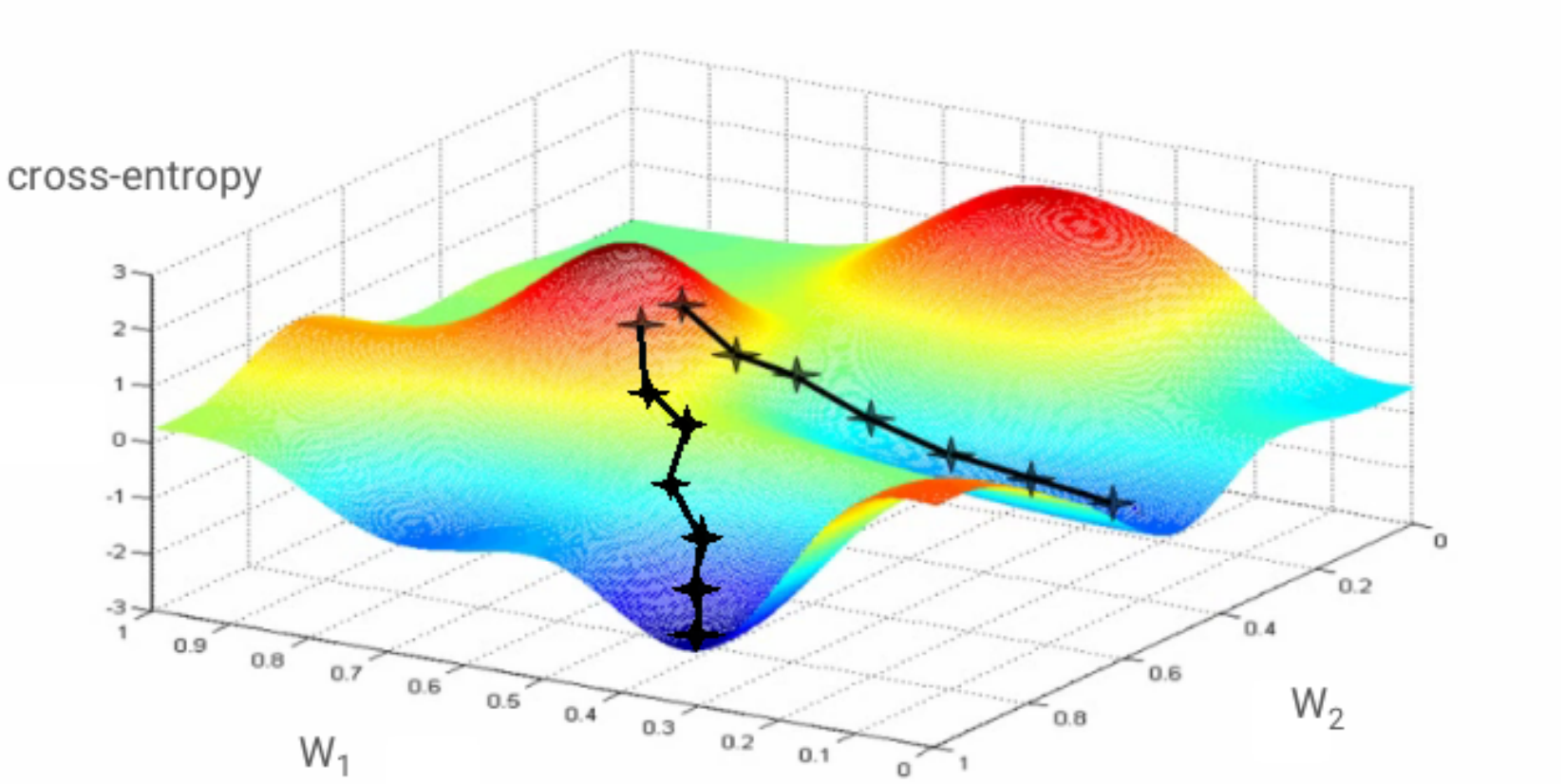
Cross Entropy



Cross entropy: $-\sum Y_i' . log(Y_i)$



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Training Loop

Training digits and labels

=> loss function

=> gradient (partial derivatives)

=> steepest descent

=> update weights and biases

=> repeat with next mini-batch of training images and labels

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

"mini-batches":
100 images and labels

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

```
import tensorflow as tf
```

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#5

mnist_1.0_softmax.py

```
import tensorflow as tf
X = tf.placeholder(tf.float32, [None, 28, 28, 1])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))

init = tf.initialize_all_variables()
```

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#5

mnist_1.0_softmax.py

```
# model
Y = tf.nn.softmax(tf.matmul(tf.reshape(X, [-1, 784]), W) + b)
# placeholder for correct labels
Y_ = tf.placeholder(tf.float32, [None, 10])

# loss function
cross_entropy = -tf.reduce_sum(Y_ * tf.log(Y))
# % of correct answers found in batch
is_correct = tf.equal(tf.argmax(Y,1), tf.argmax(Y_,1))
accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))
```

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#5

mnist_1.0_softmax.py

```
sess = tf.Session()
sess.run(init)

for i in range(1000):
    # load batch of images and correct answers
    batch_X, batch_Y = mnist.train.next_batch(100)
    train_data={X: batch_X, Y_: batch_Y}

    # train
    sess.run(train_step, feed_dict=train_data)
```

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#5

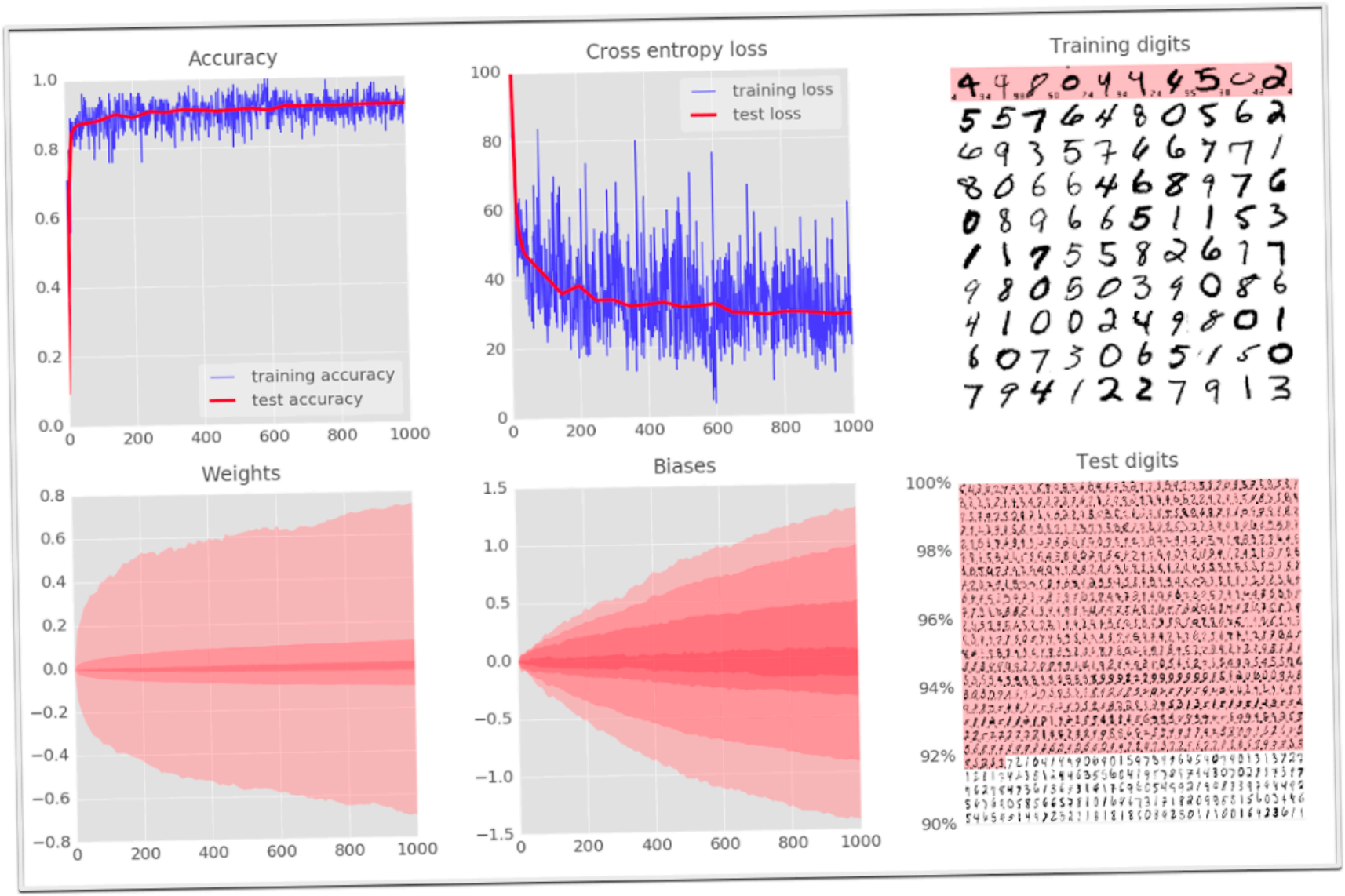
mnist_1.0_softmax.py

```
# success ?
a,c = sess.run([accuracy, cross_entropy],
feed_dict=train_data)


# success on test data ?
test_data={X: mnist.test.images, Y_: mnist.test.labels}
a,c = sess.run([accuracy, cross_entropy], feed=test_data)
```

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#5

mnist_1.0_softmax.py

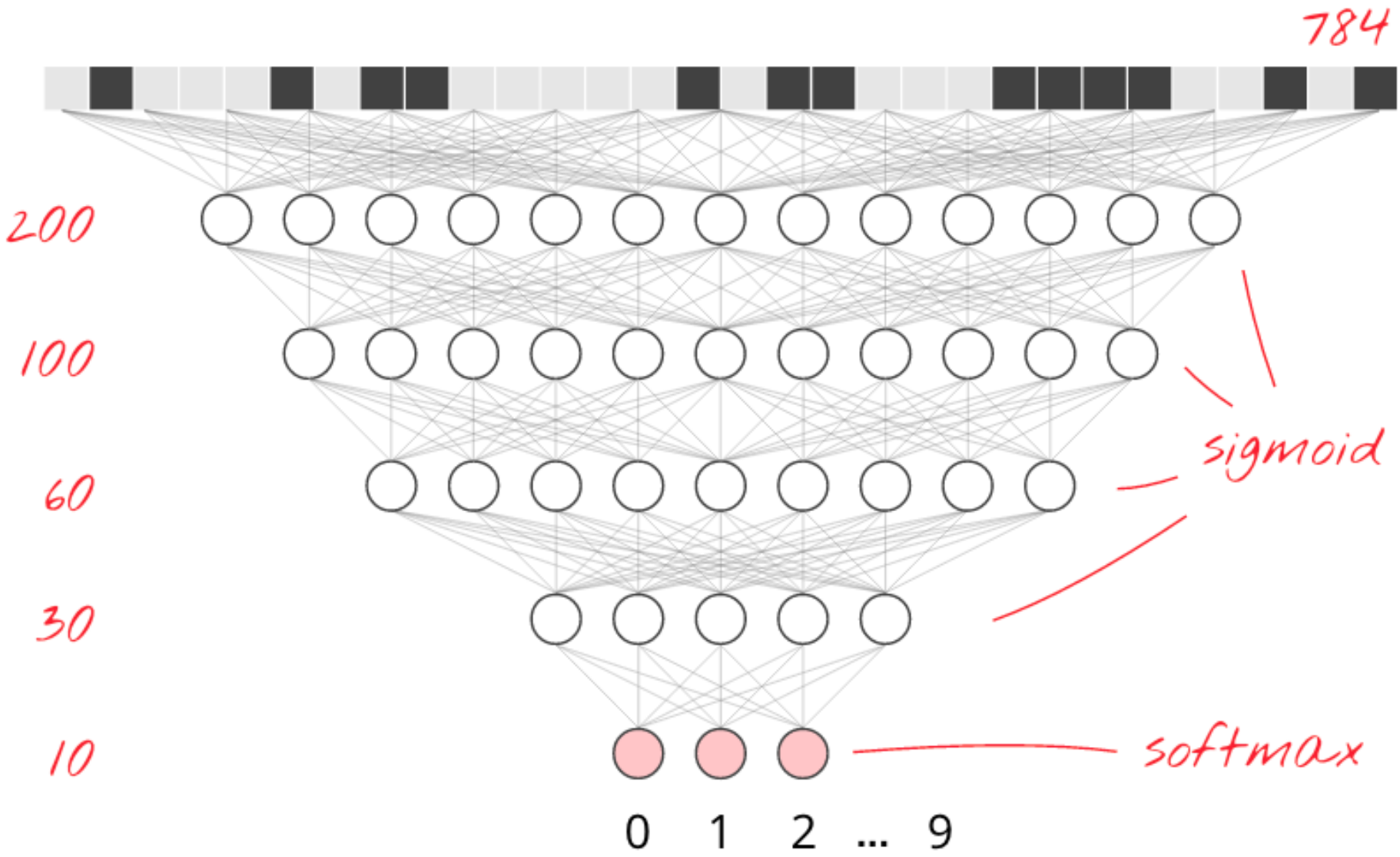


92%

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#5

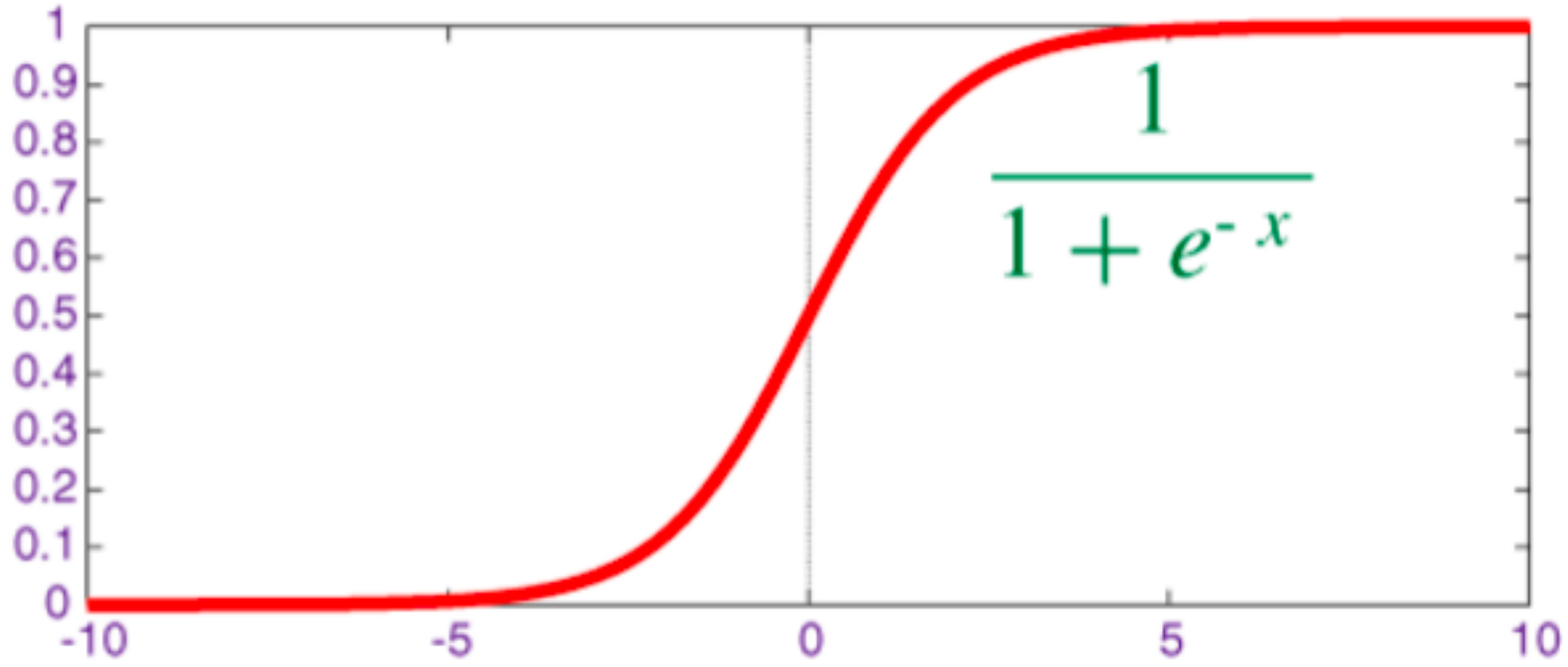
Deep Learning

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

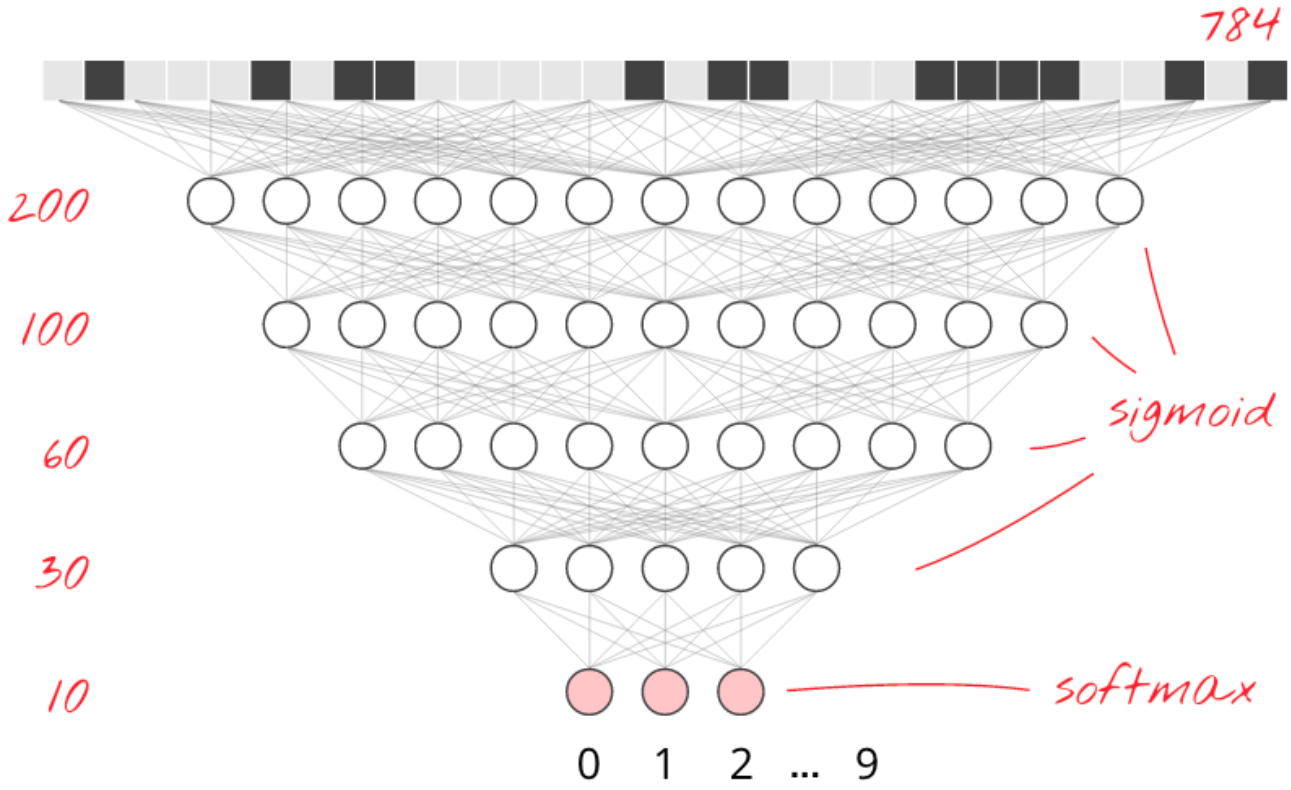


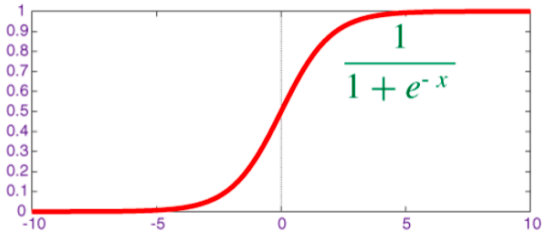
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Sigmoid

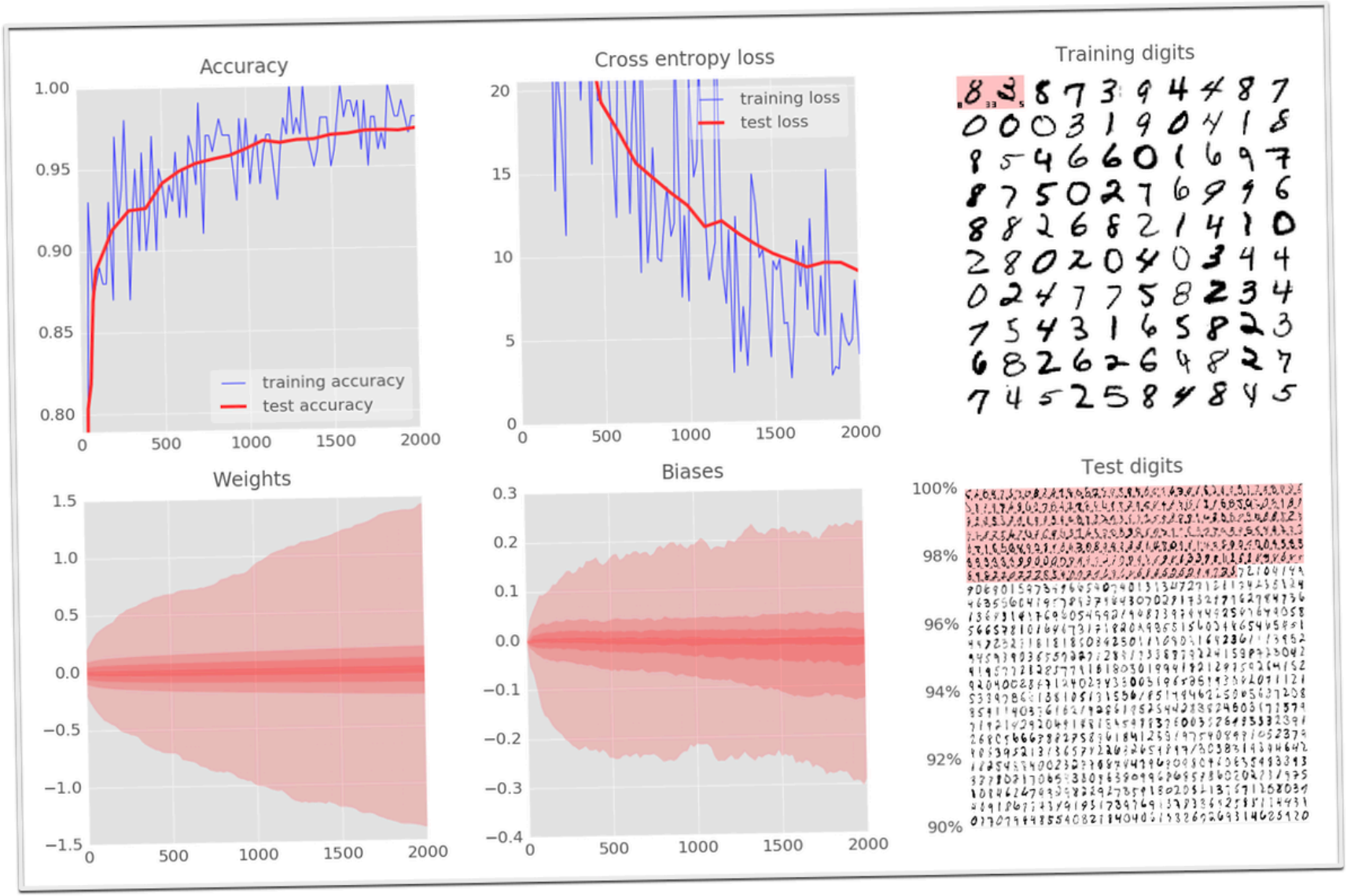


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/





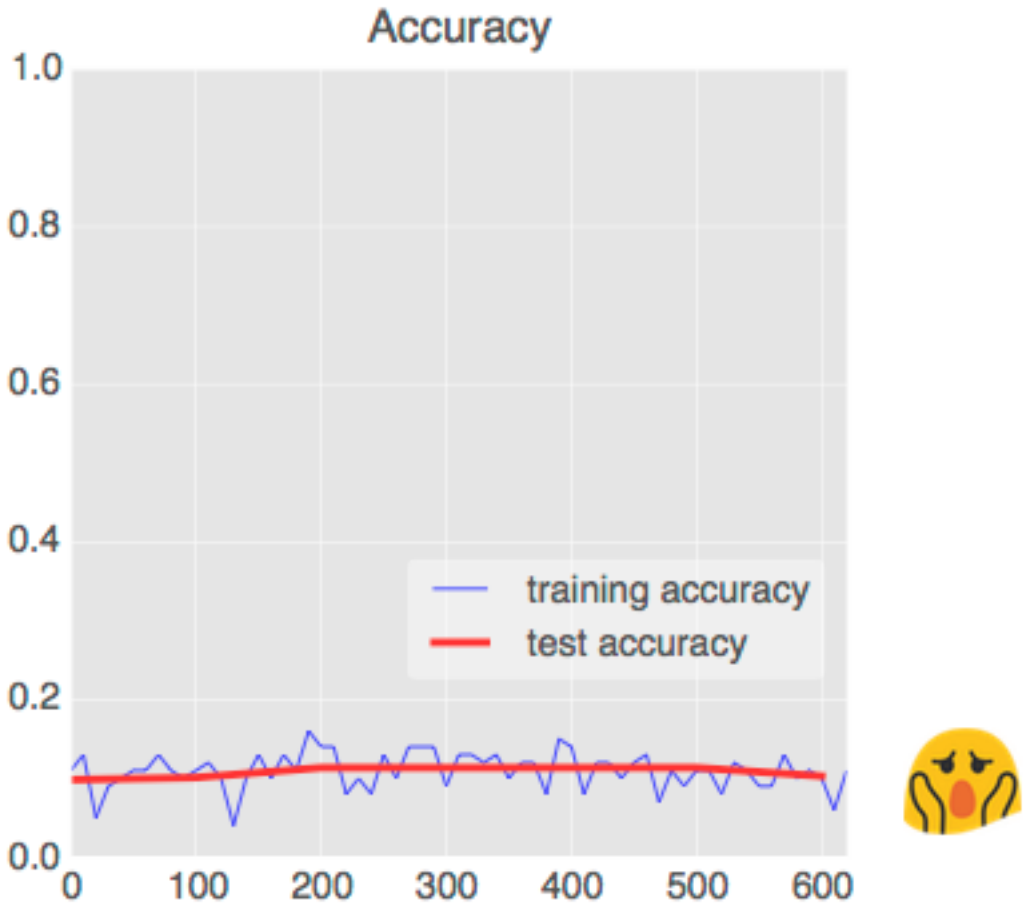
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

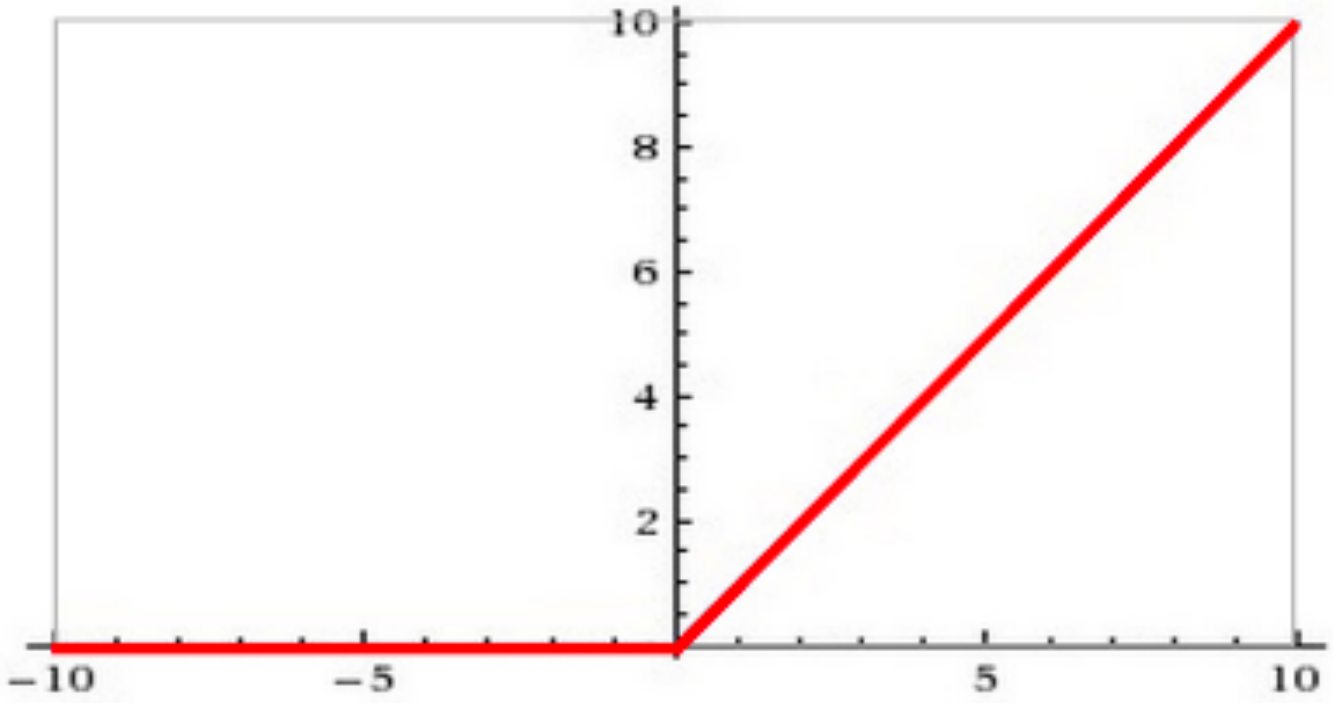
ReLU

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

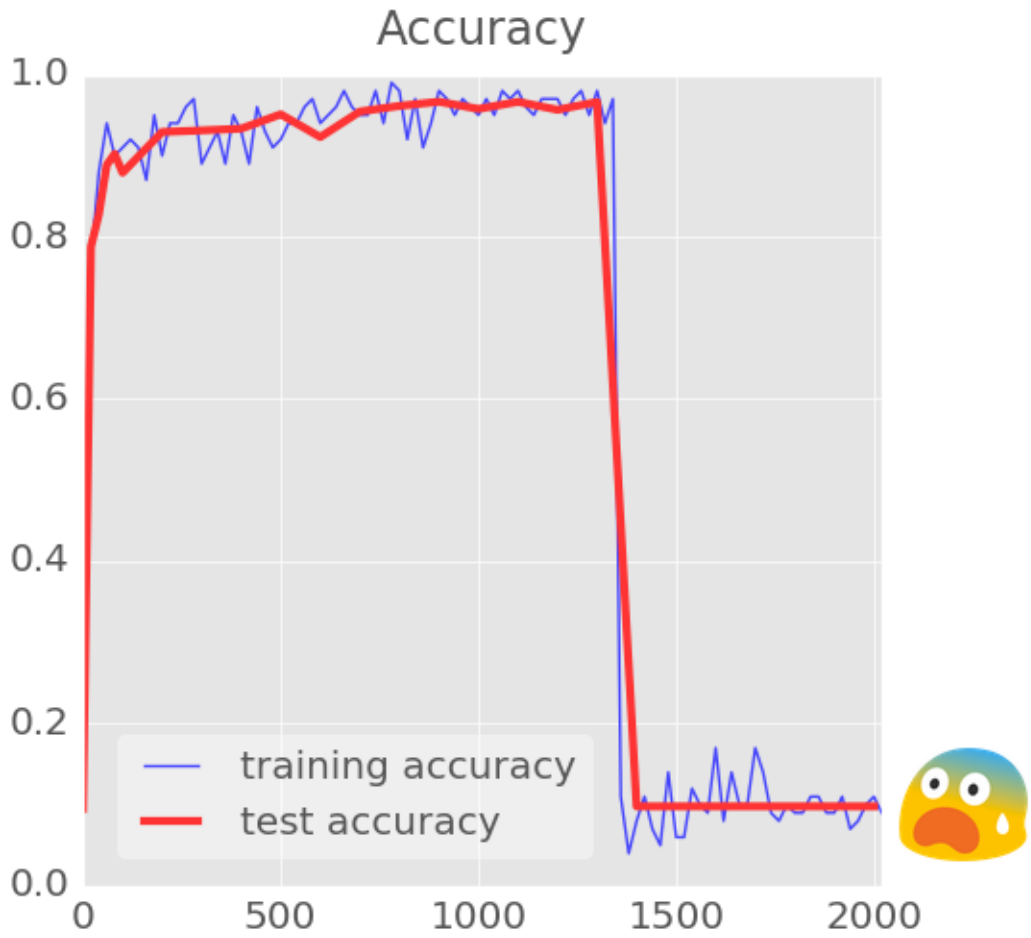


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

ReLU



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

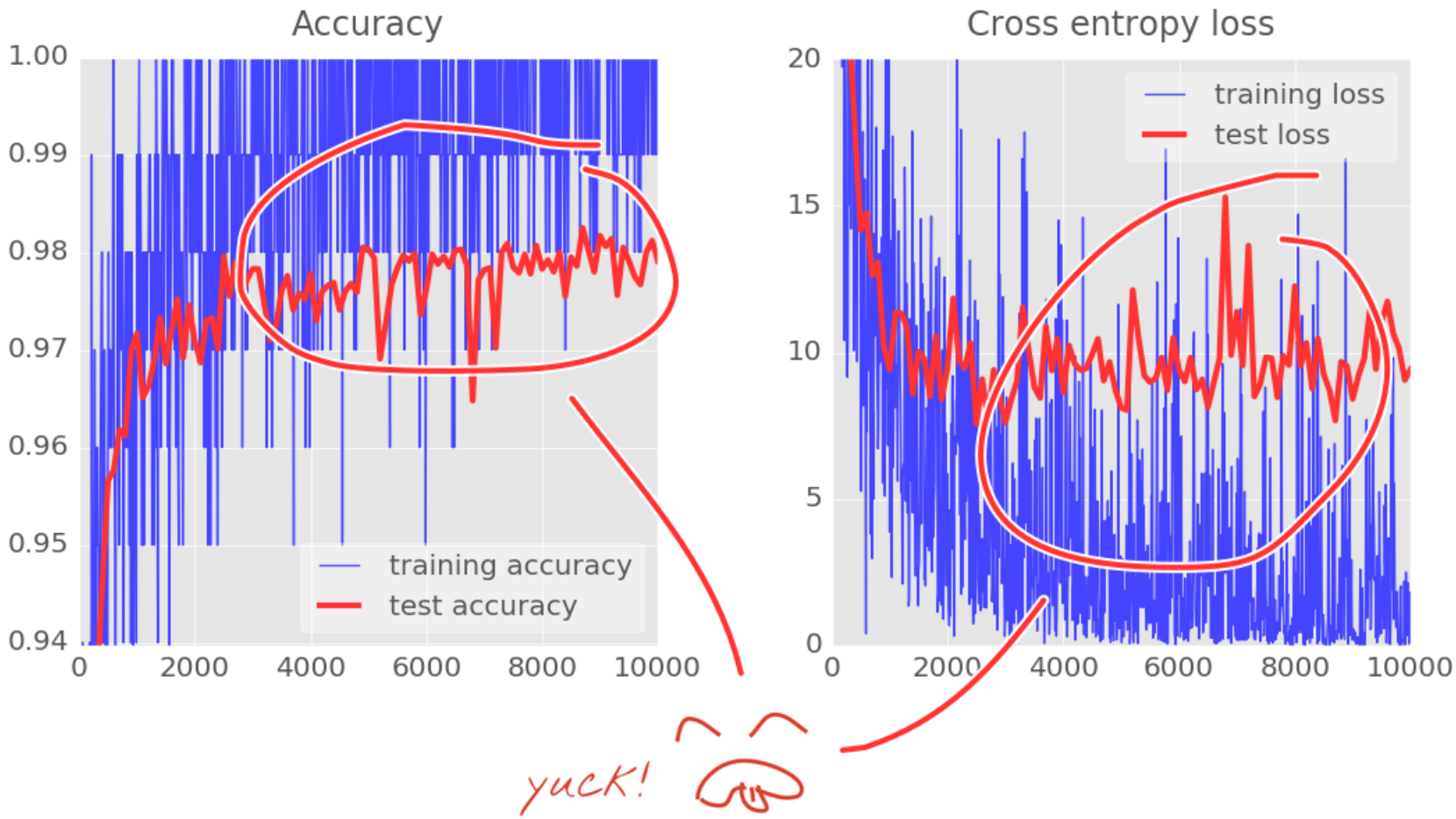


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Learning Rate

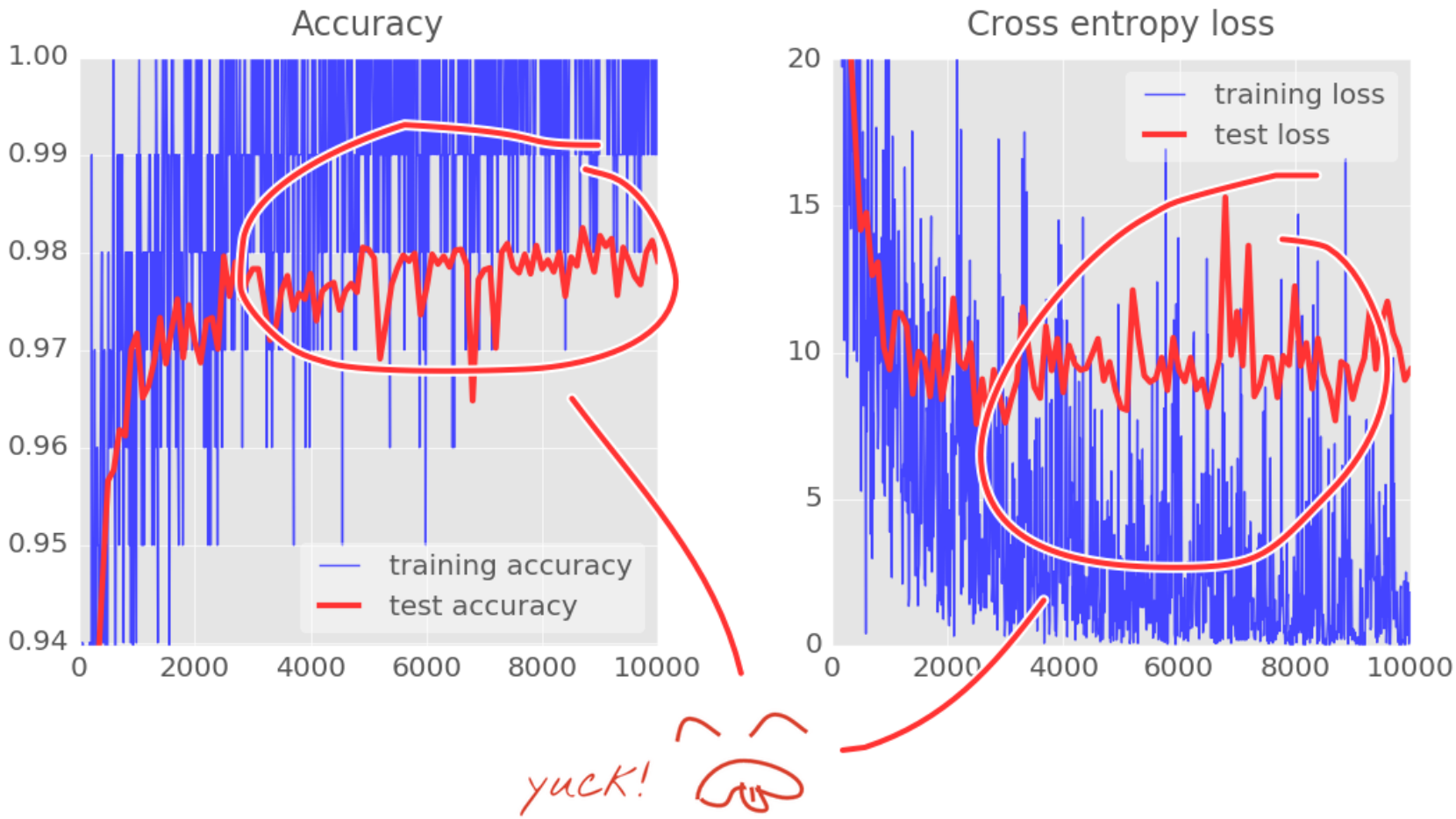
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

LR = 0.003

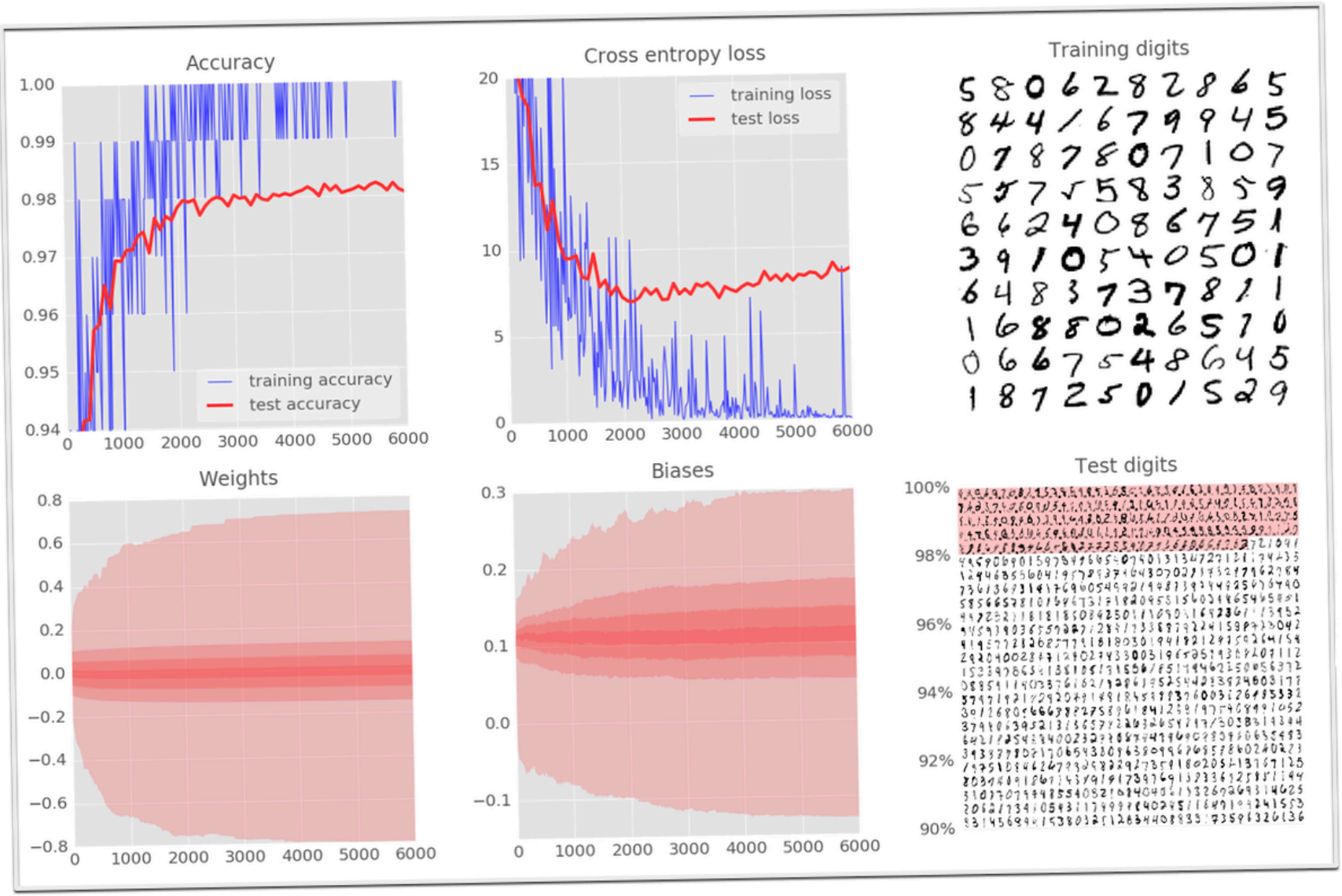


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

LR = 0.003

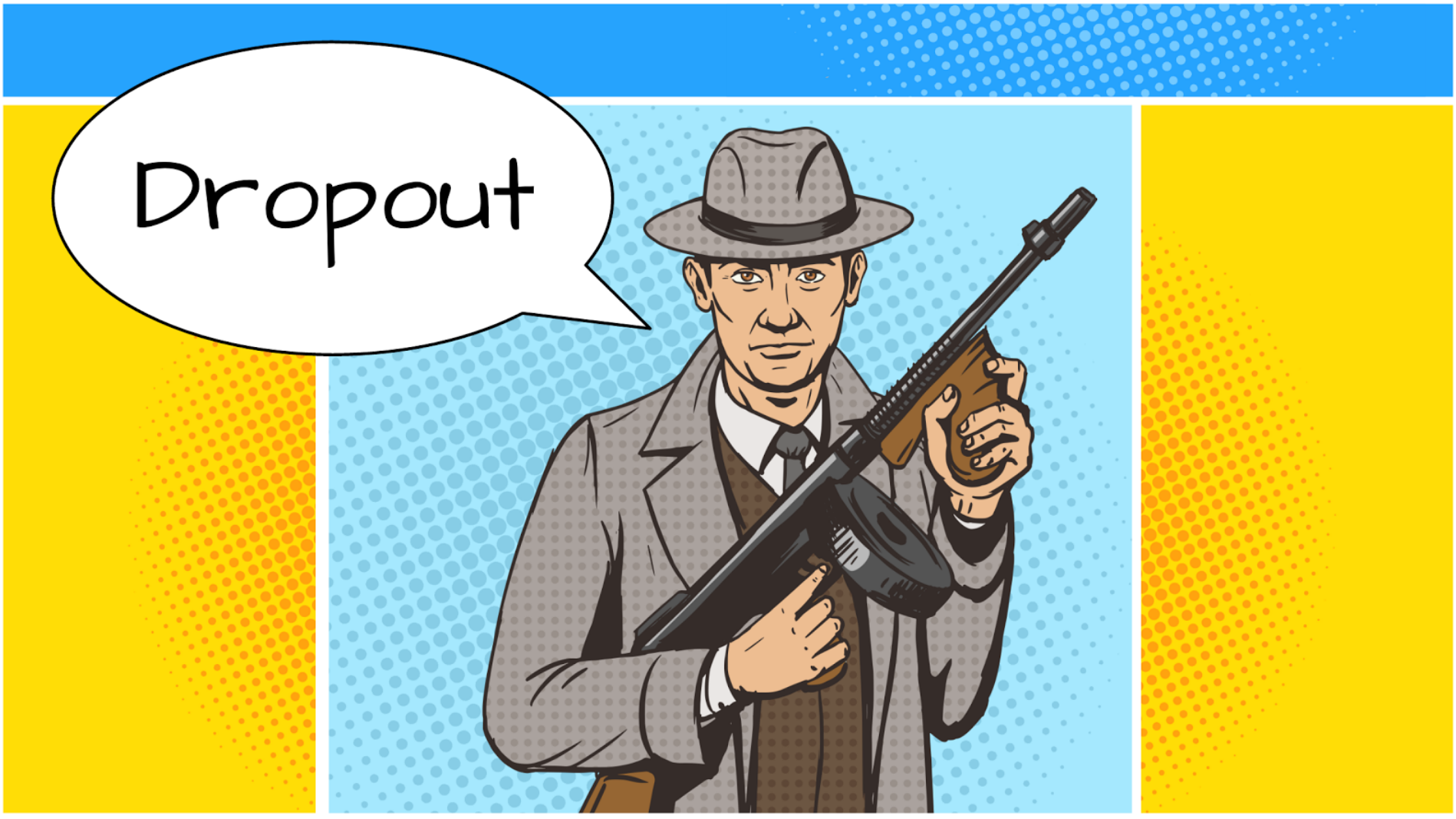


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

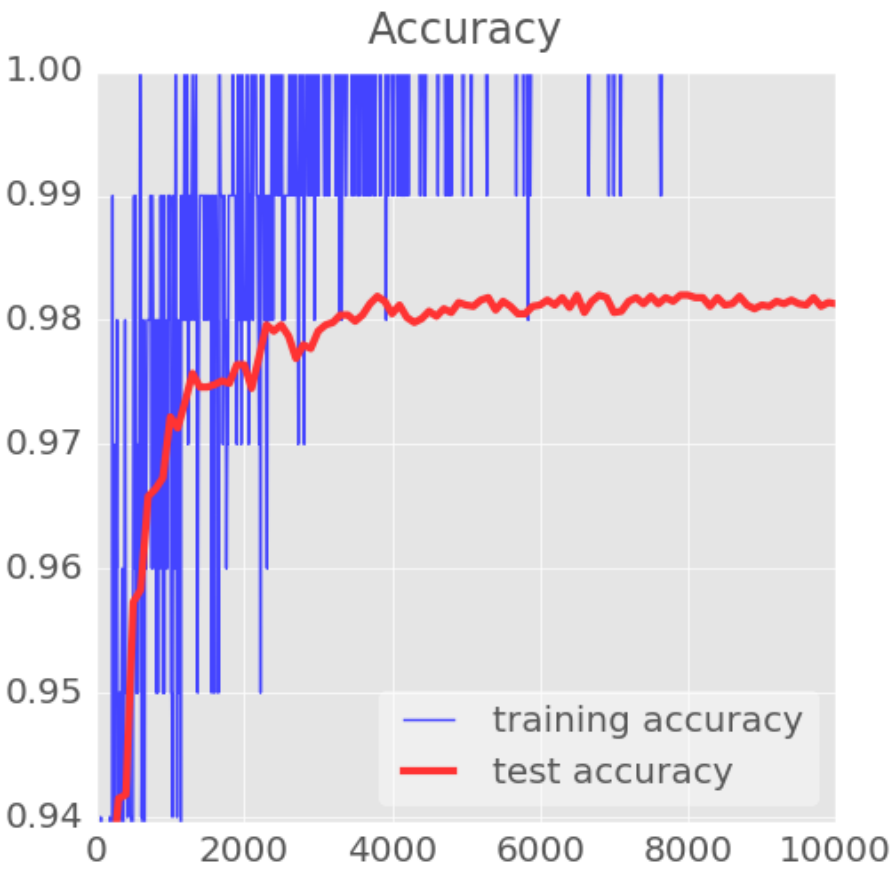


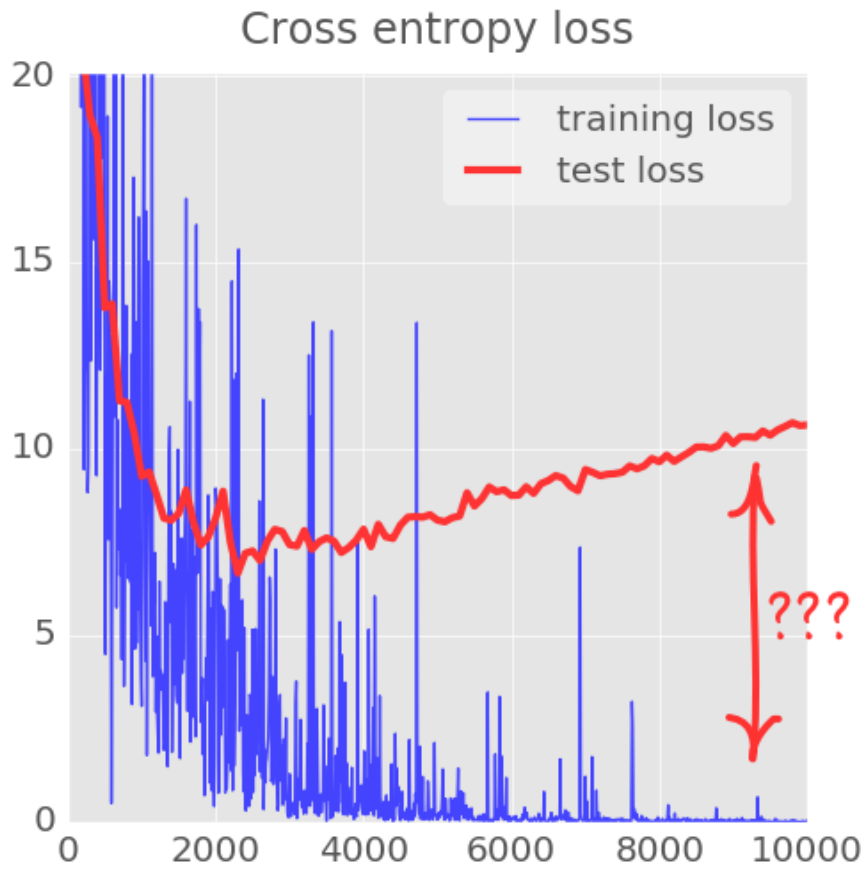
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Dropout



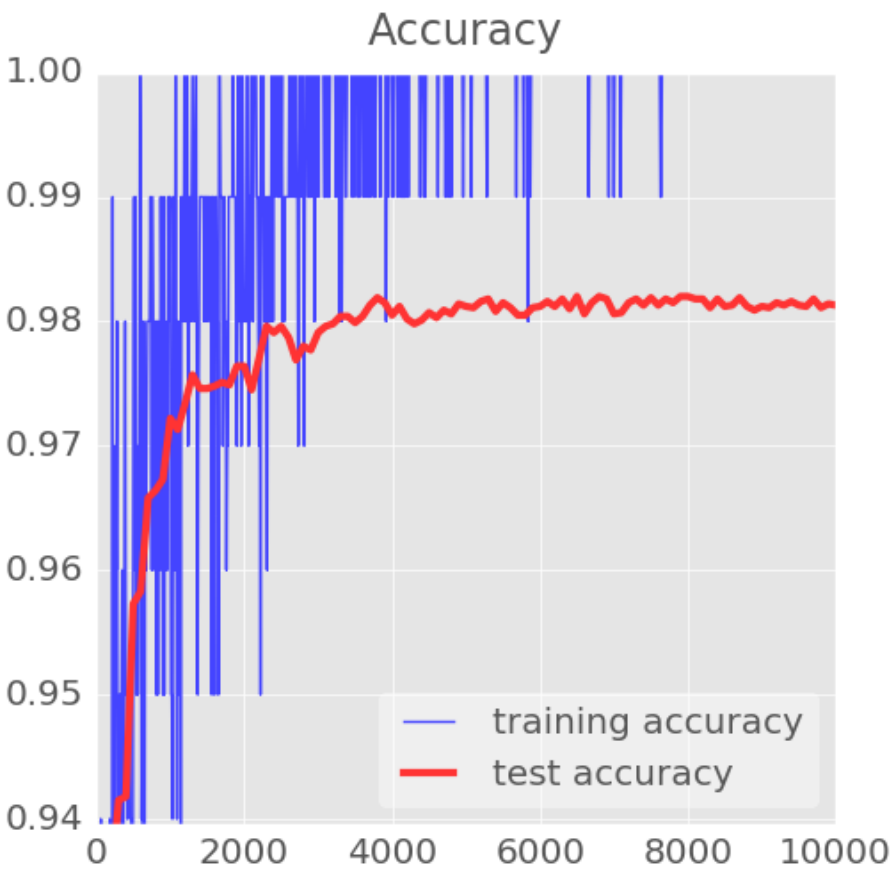
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

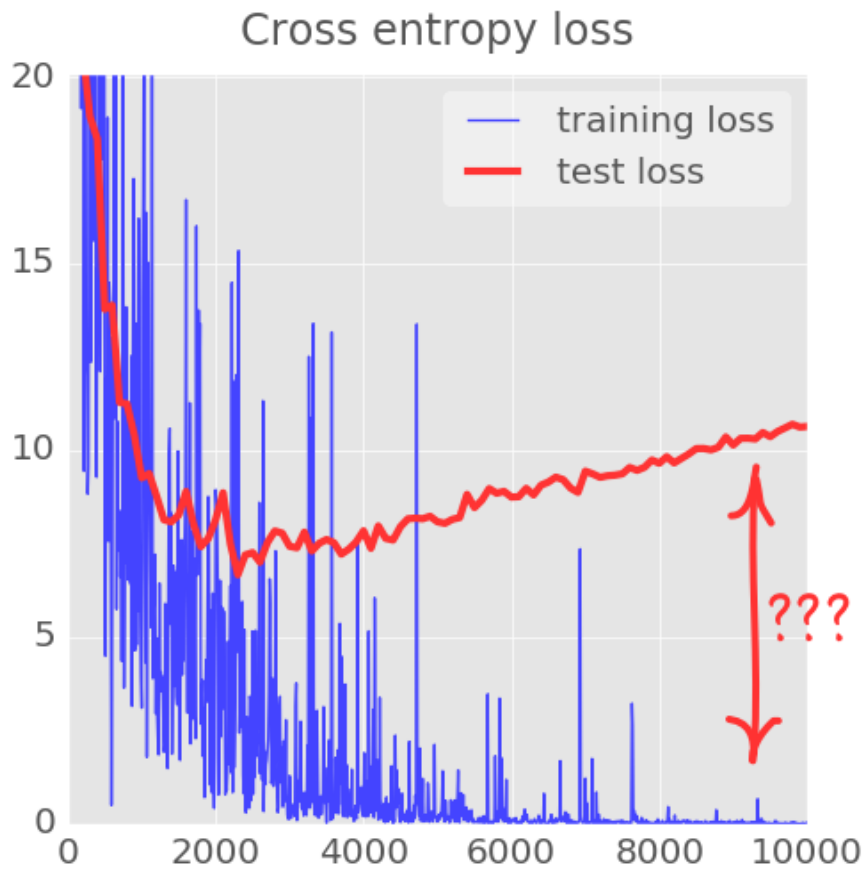




Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

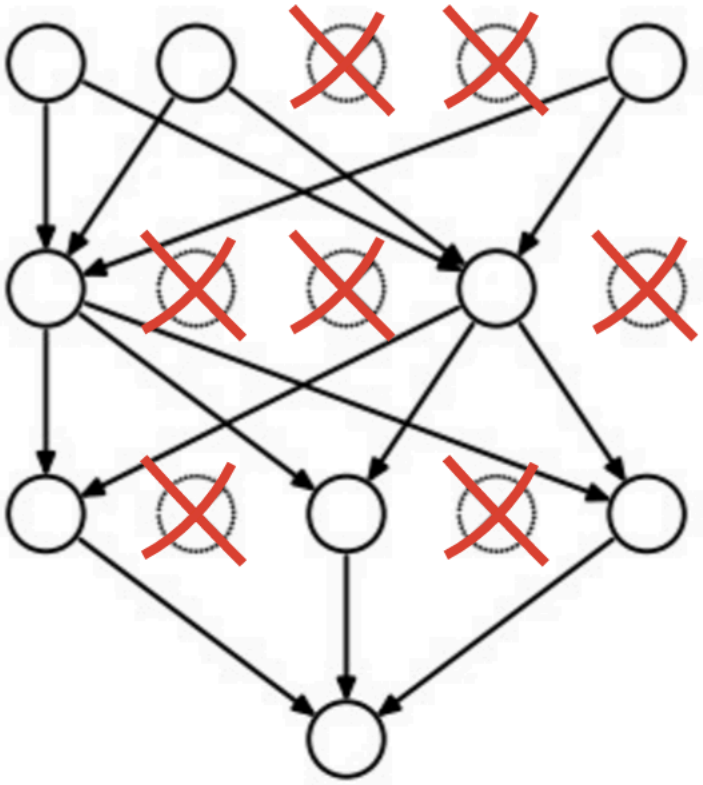
Overfitting



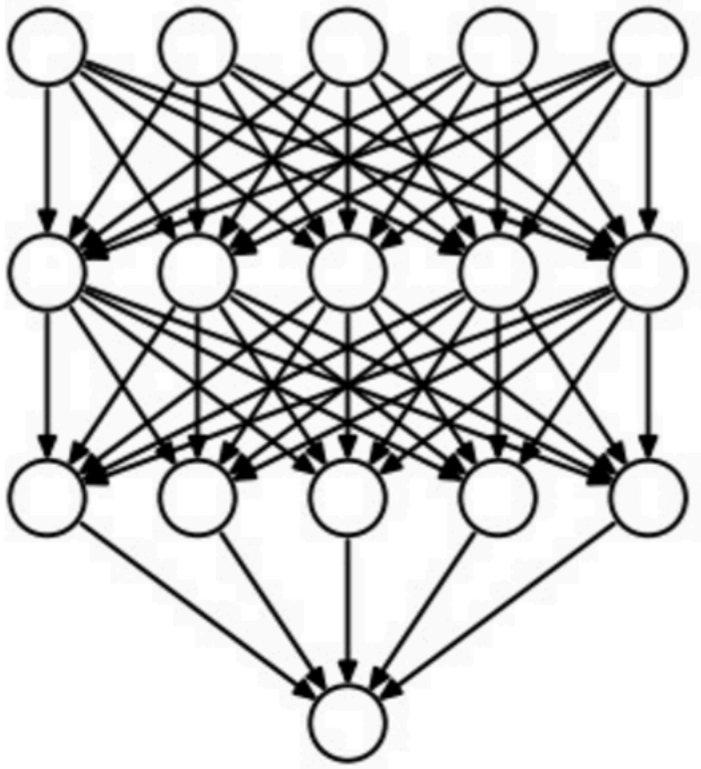


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Dropout



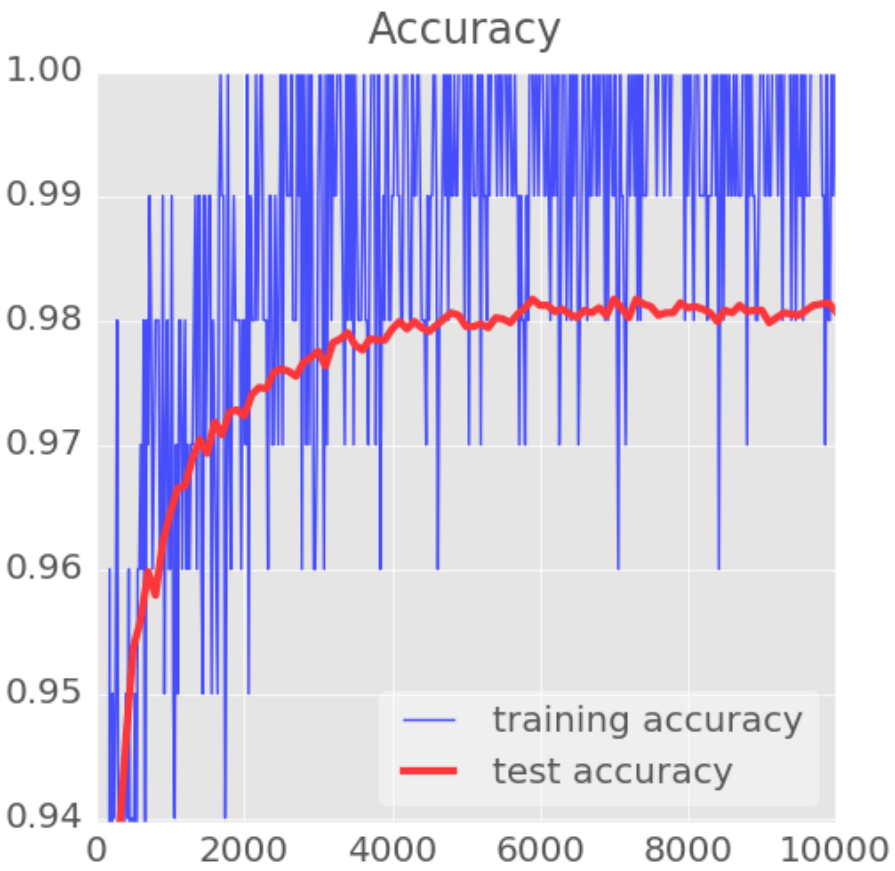
Training
pkeep = 0.75

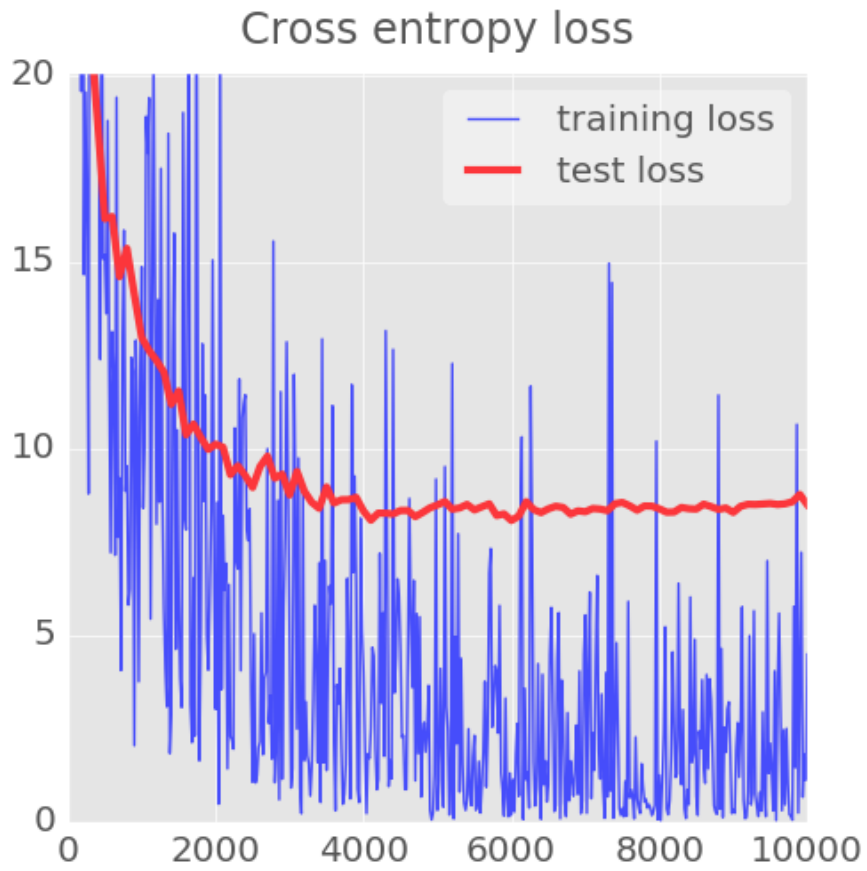


Test
pkeep = 1.0

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

TensorFlow MNIST Tutorial

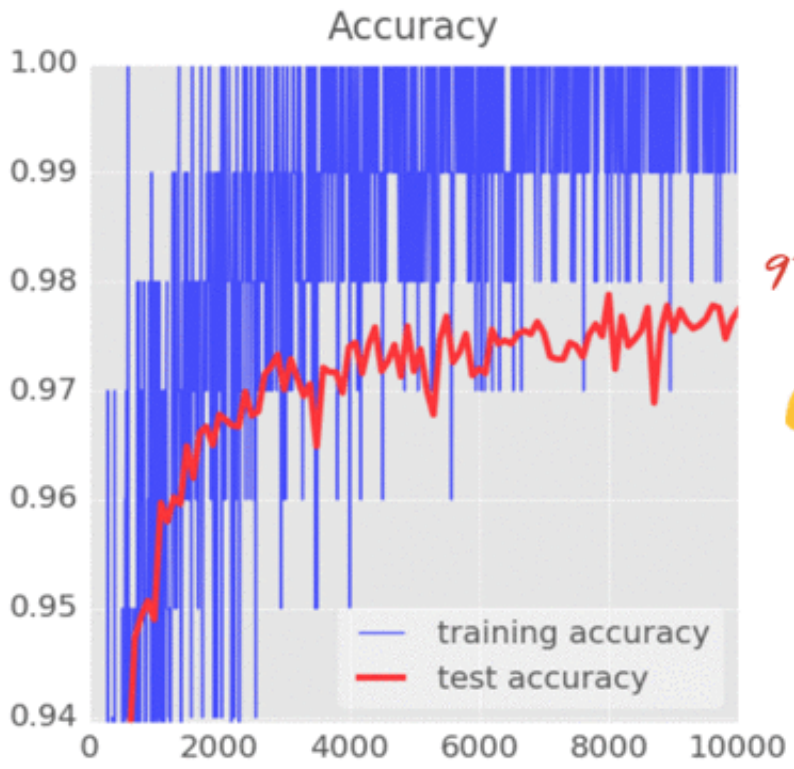




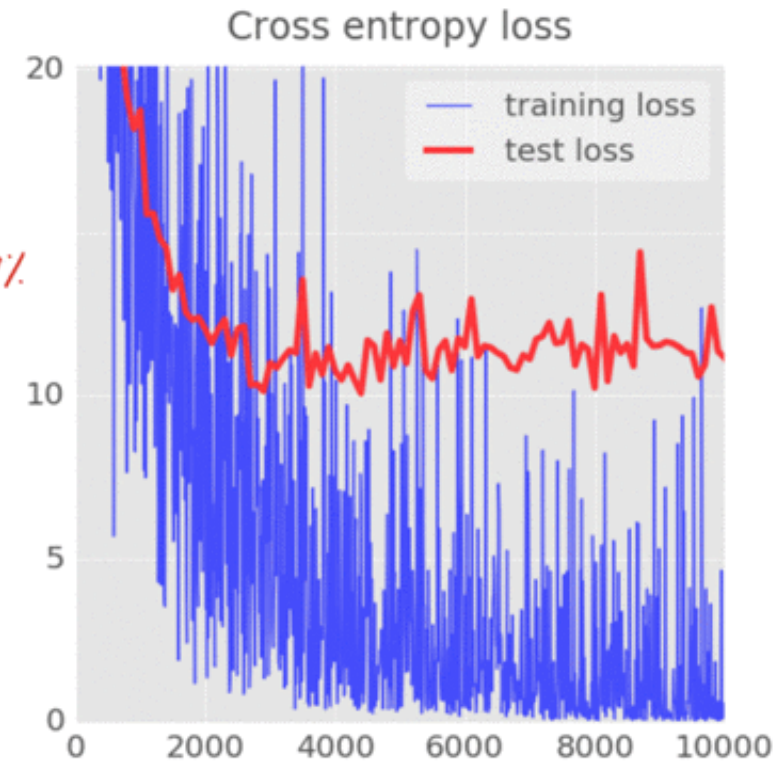
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

TensorFlow MNIST Tutorial

5 layers
sigmoid

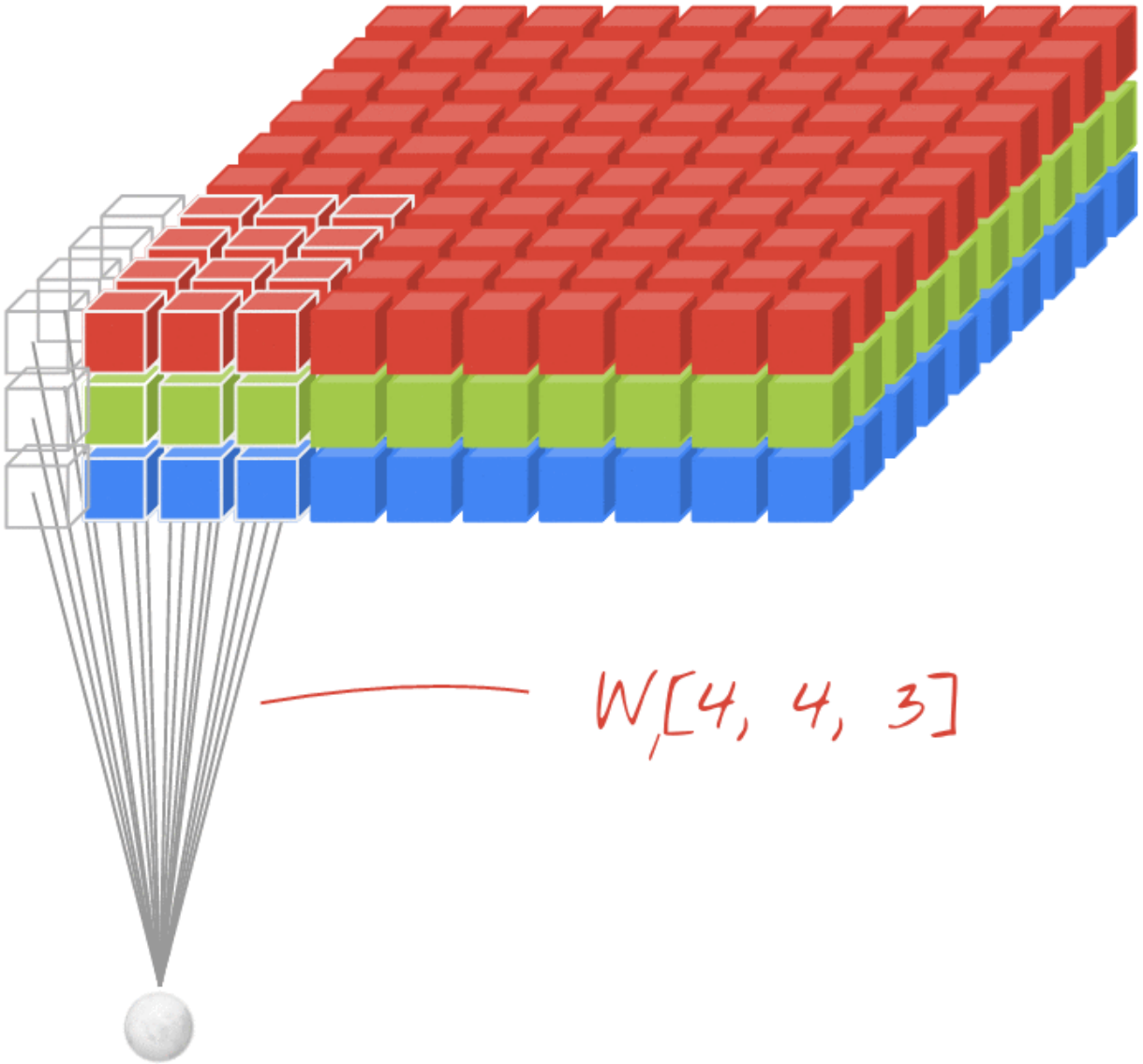


97.9%

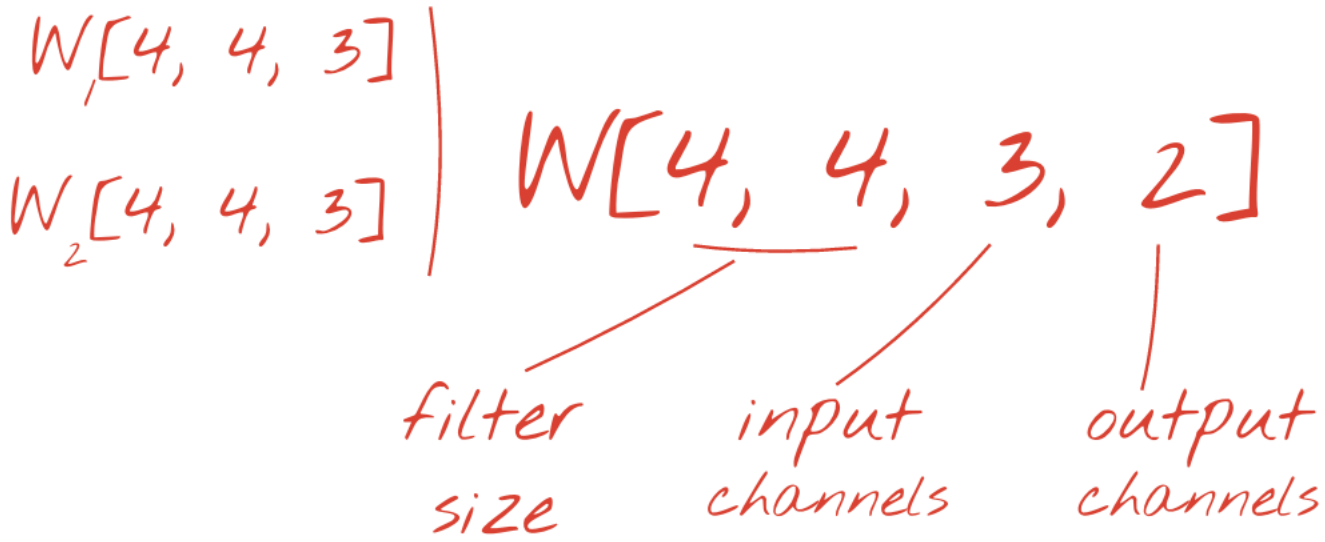


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

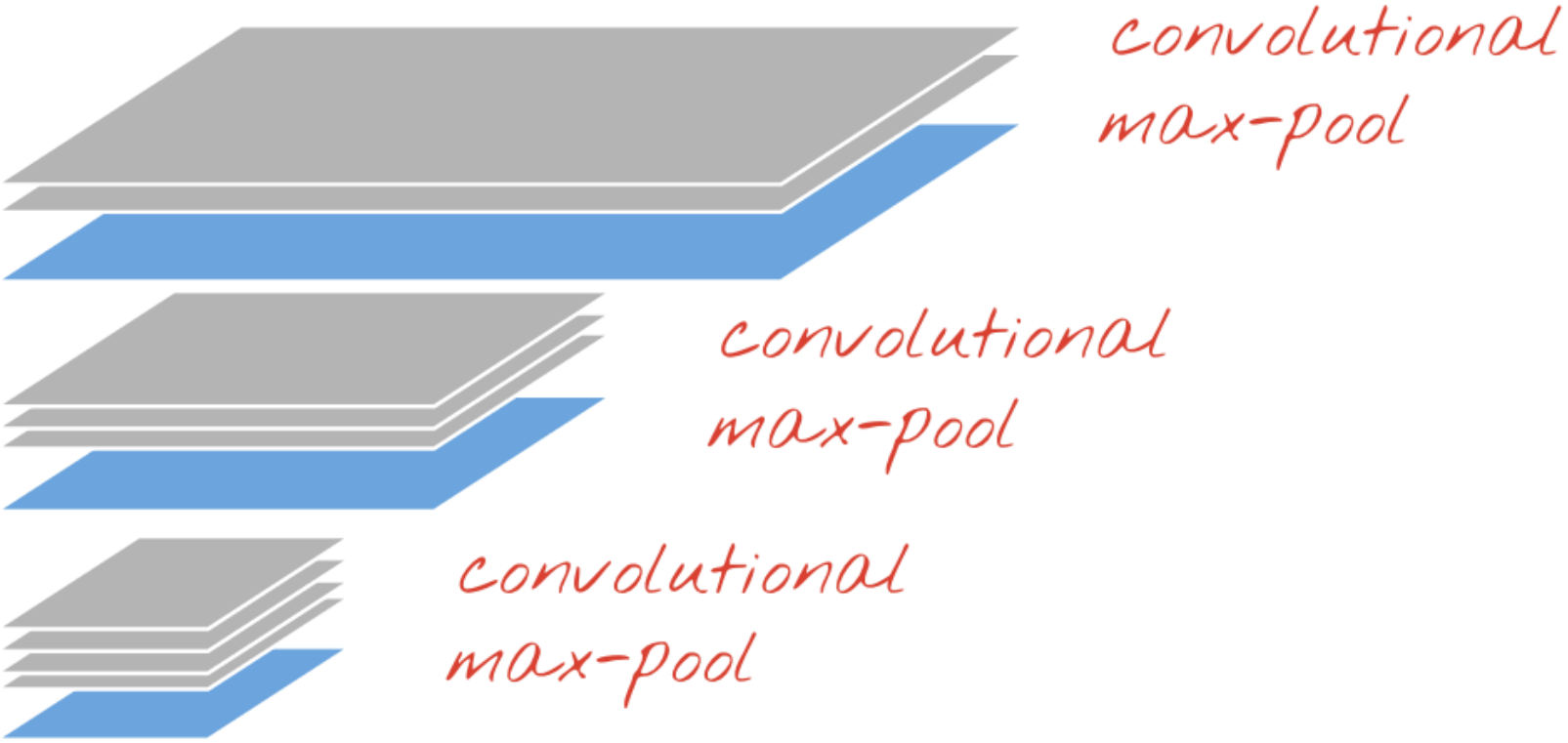


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/



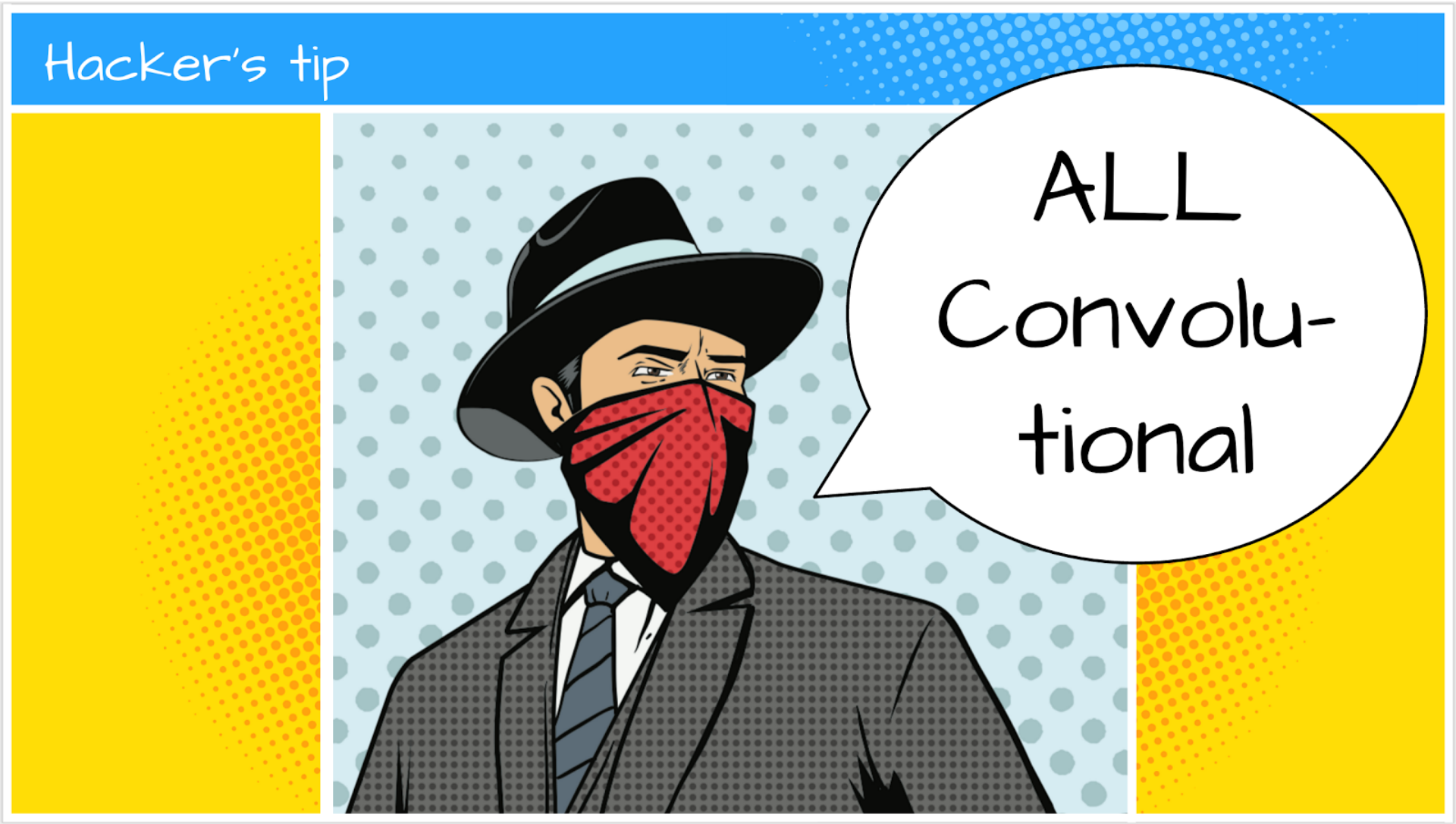
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

Convolutional Max-Pool



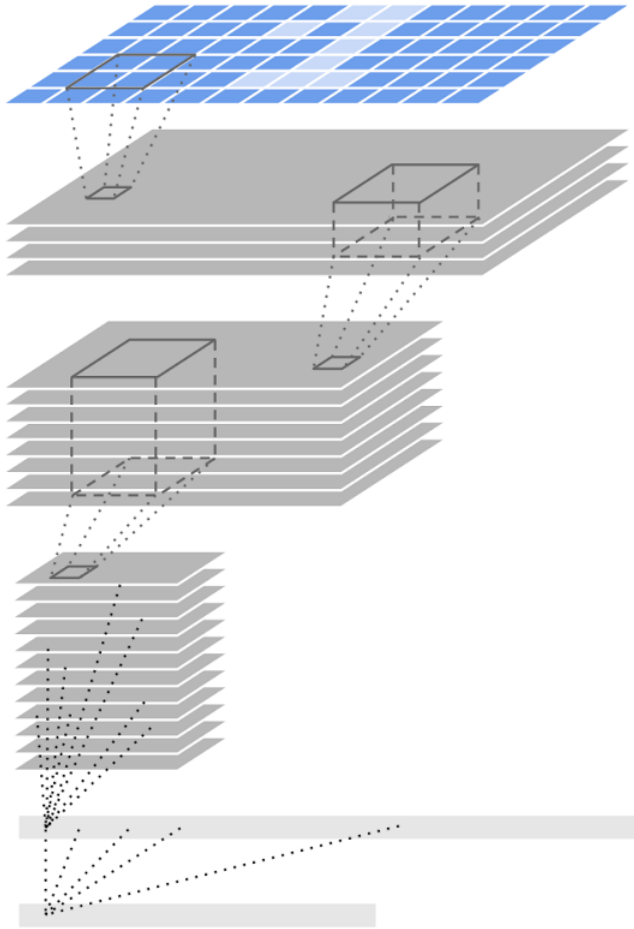
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

All Convolutional



Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

+ biases on all layers



convolutional layer, 4 channels
W1[5, 5, 1, 4] stride 1

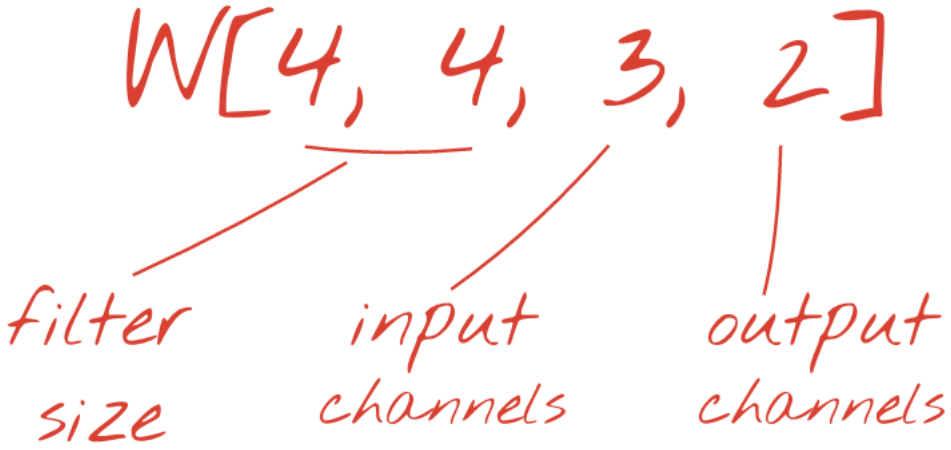
convolutional layer, 8 channels
W2[4, 4, 4, 8] stride 2

convolutional layer, 12 channels
W3[4, 4, 8, 12] stride 2

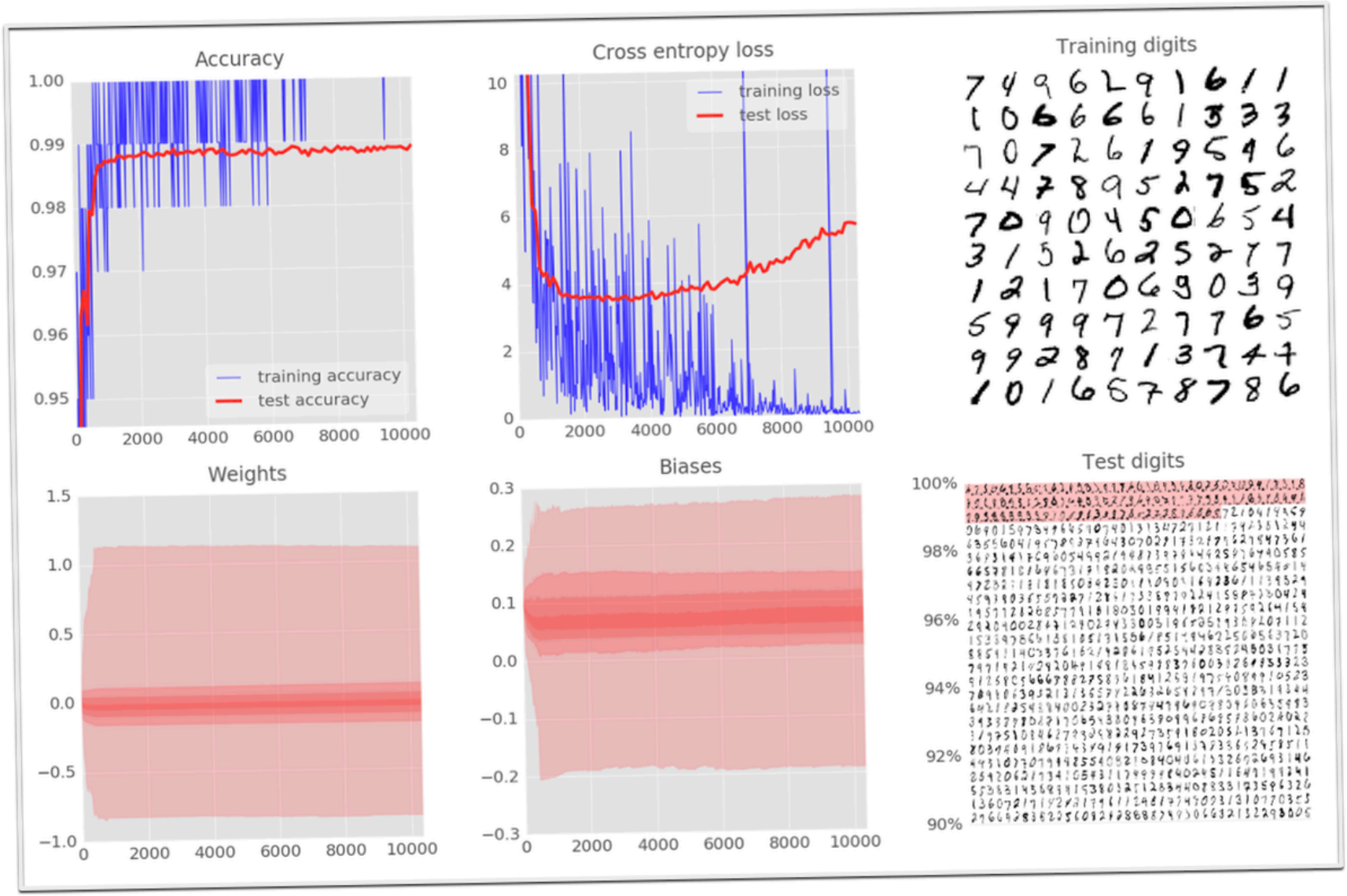
fully connected layer
W4[7x7x12, 200]

softmax readout layer
W5[200, 10]

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

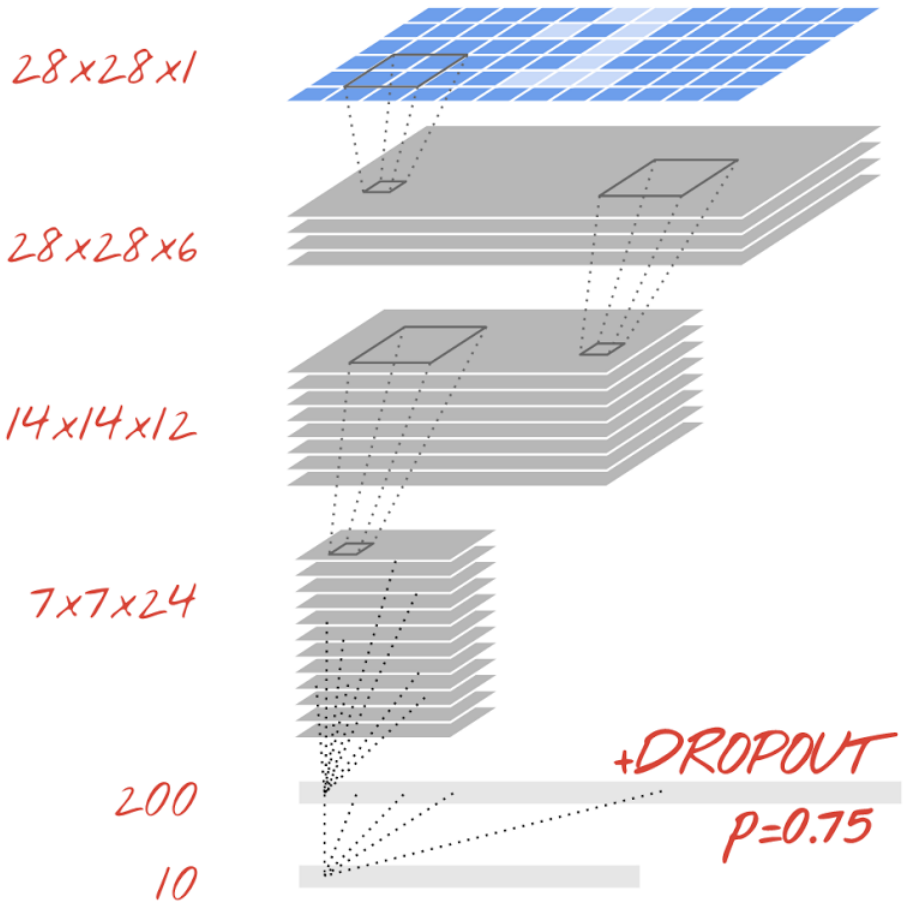


Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/


Accuracy
1.00
0.99
0.98
0.97
0.96
0.95
training accuracy
test accuracy
0
2000
4000
6000
8000
10000
Cross entropy loss
10
8
6
4
2
0
training loss
test loss
Training digits
Weights
1.5
1.0
0.5
0.0
−0.5
−1.0
Biases
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Test digits
100%
98%
96%
94%
92%
90%

98.9%

+ biases on all layers



convolutional layer, **6 channels**
W1[**6**, **6**, 1, 6] stride 1

convolutional layer, **12 channels**
W2[**5**, **5**, 6, 12] stride 2

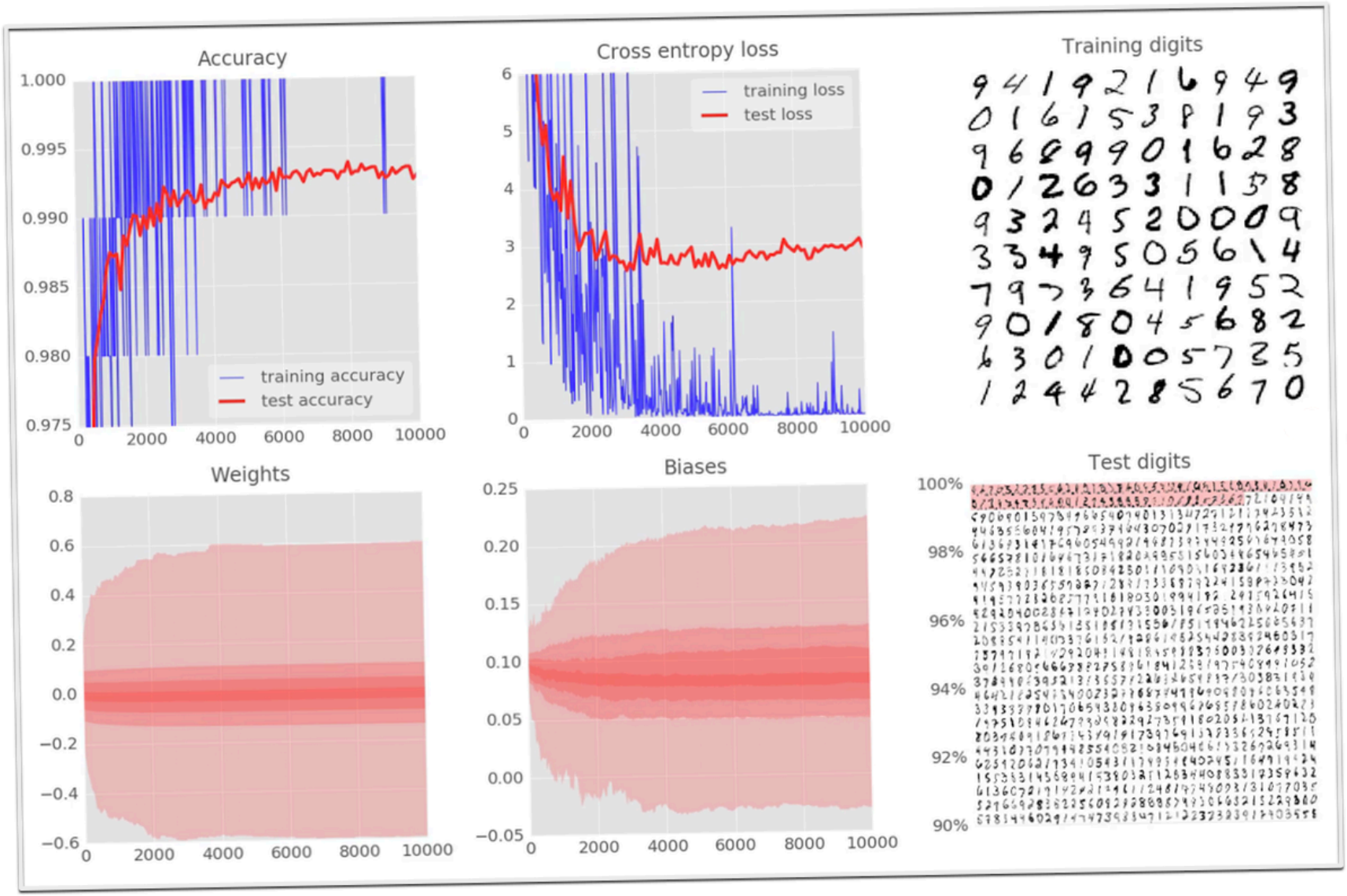
convolutional layer, **24 channels**
W3[**4**, **4**, 12, 24] stride 2

fully connected layer
W4[7x7x24, 200]

softmax readout layer
W5[200, 10]

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

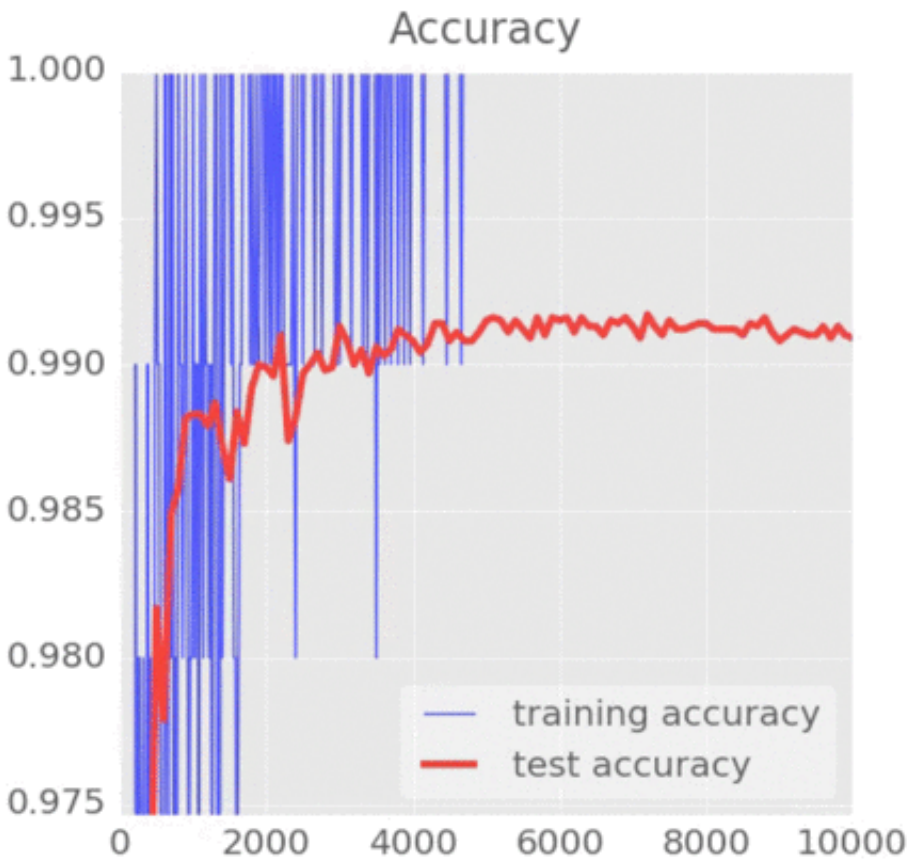
TensorFlow MNIST Tutorial

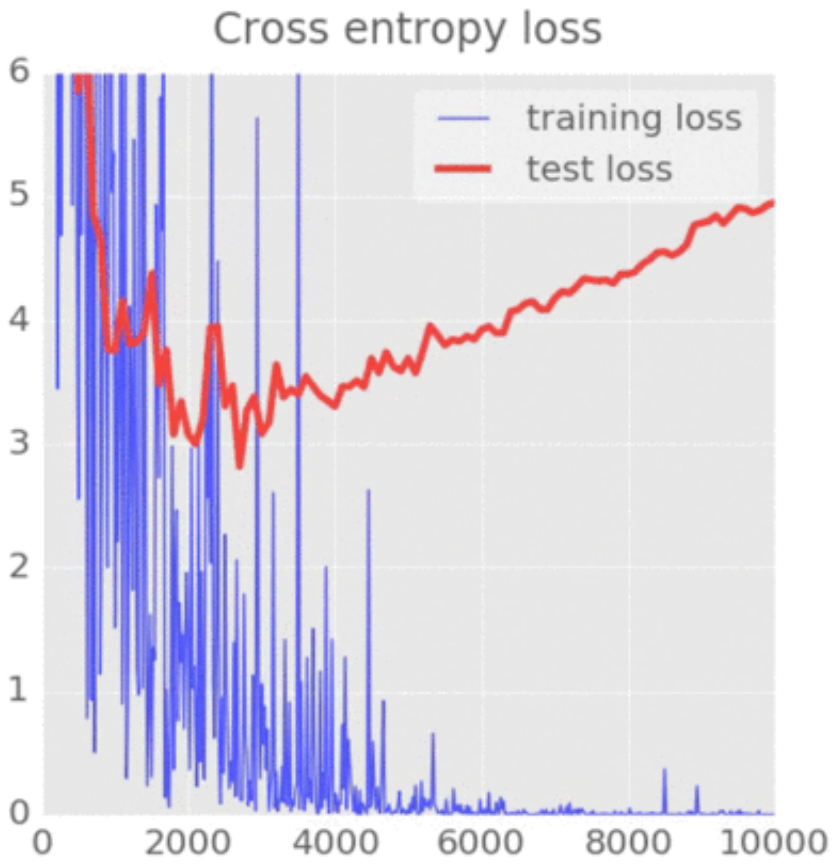


99.3%

Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

TensorFlow MNIST Tutorial

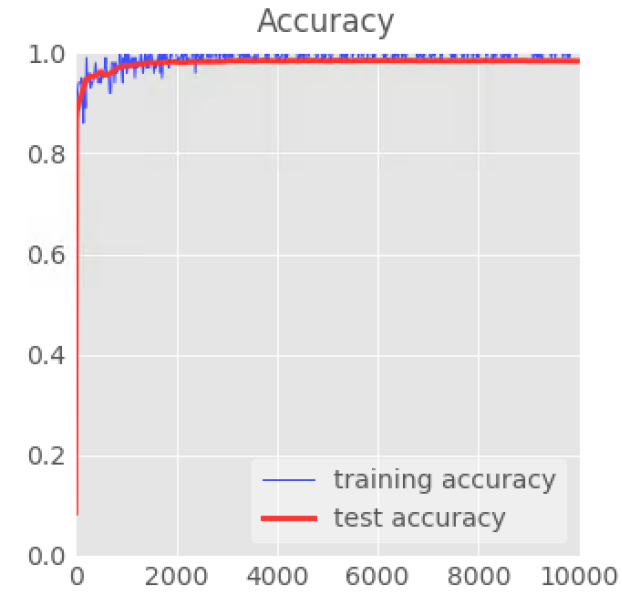


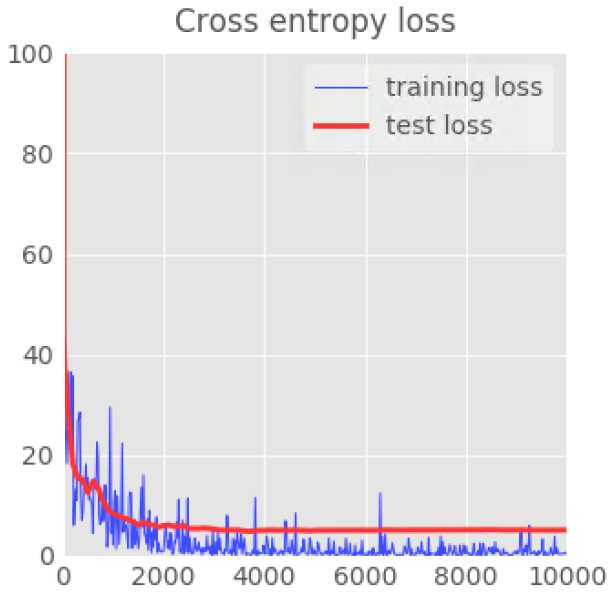


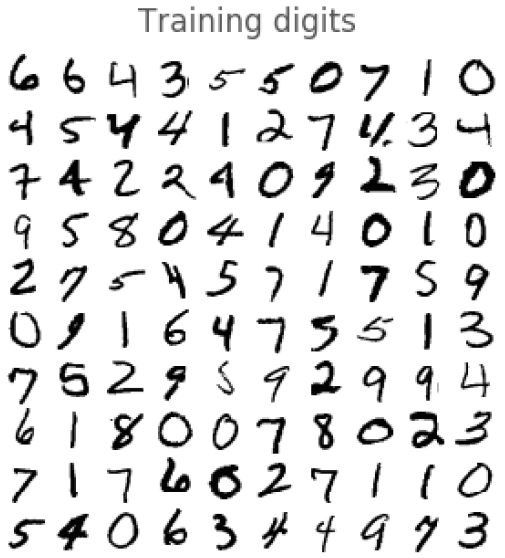
larger convolutional network

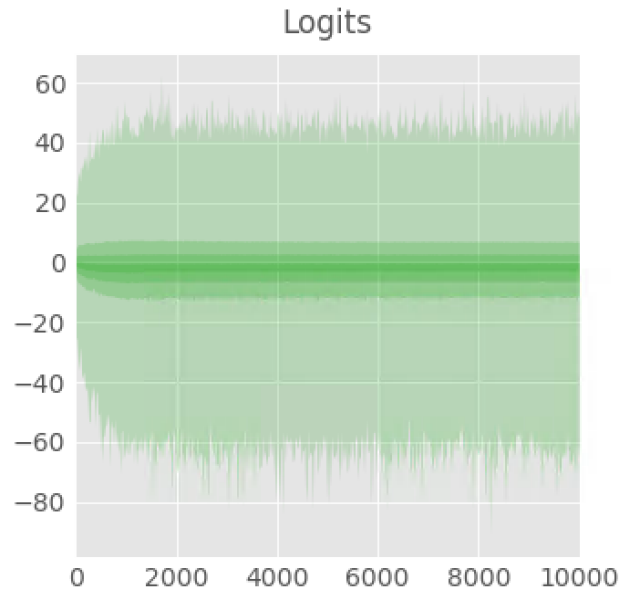
Source: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/

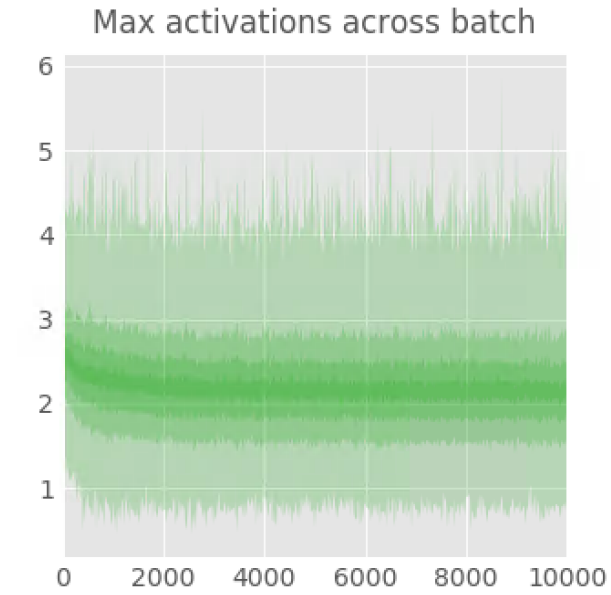
TensorFlow MNIST Tutorial

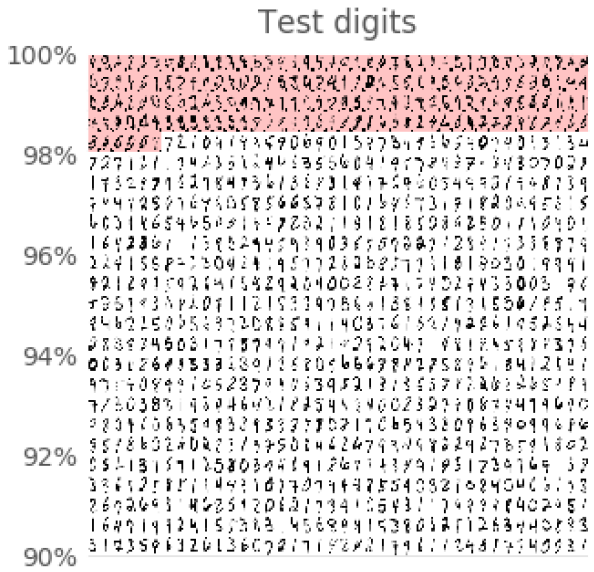






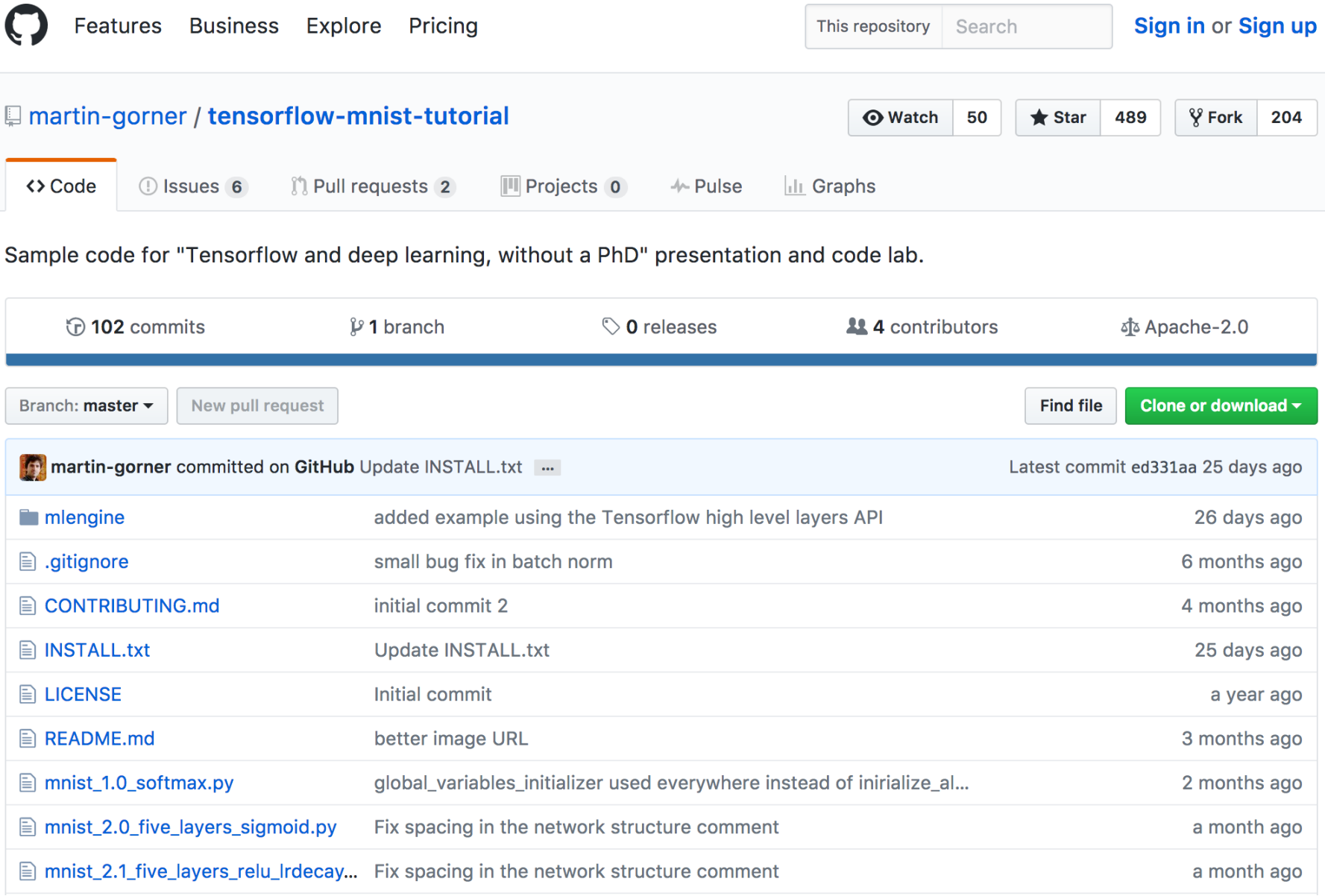






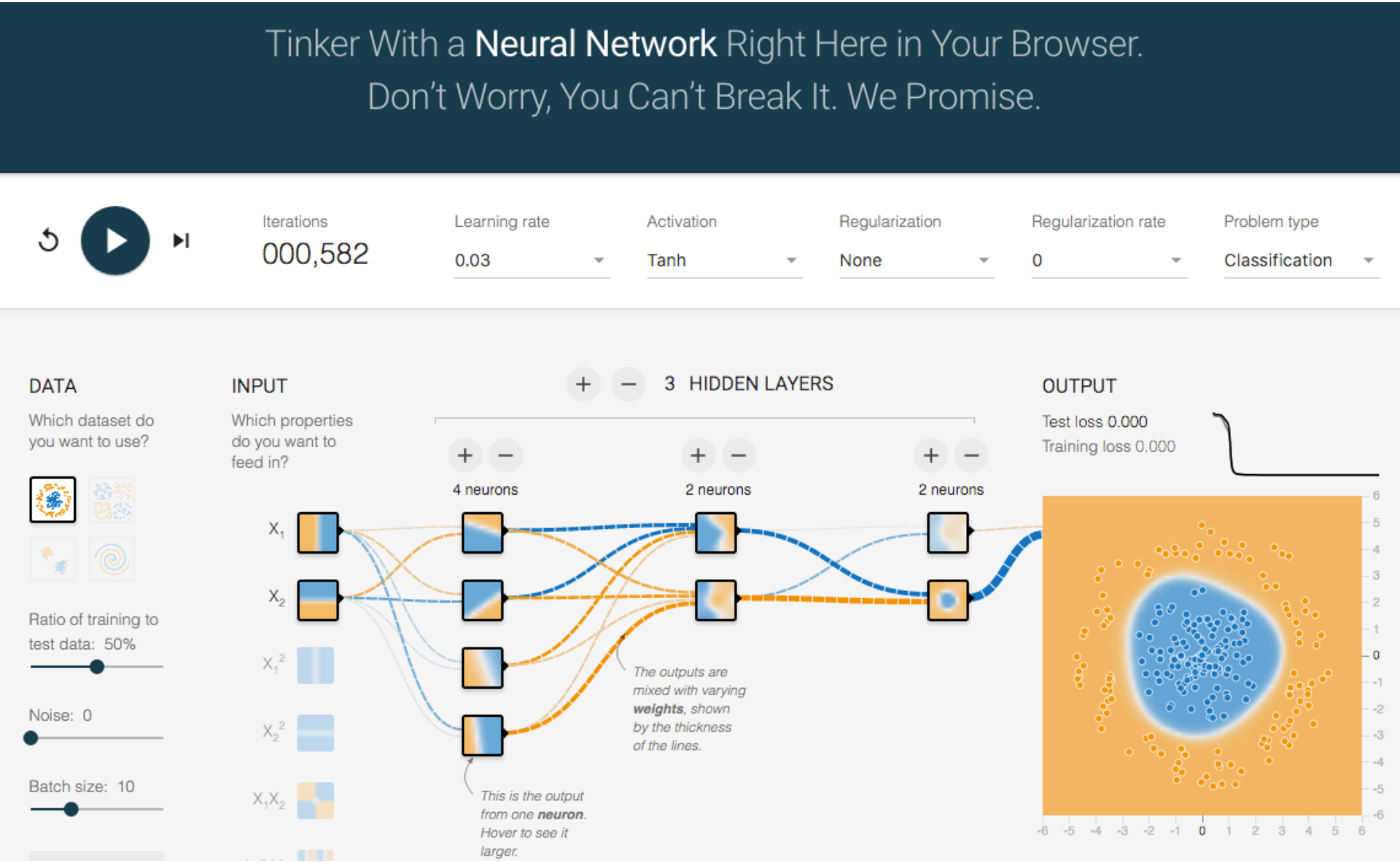
Source: https://github.com/martin-gorner/tensorflow-mnist-tutorial/

TensorFlow MNIST Tutorial



https://github.com/martin-gorner/tensorflow-mnist-tutorial/

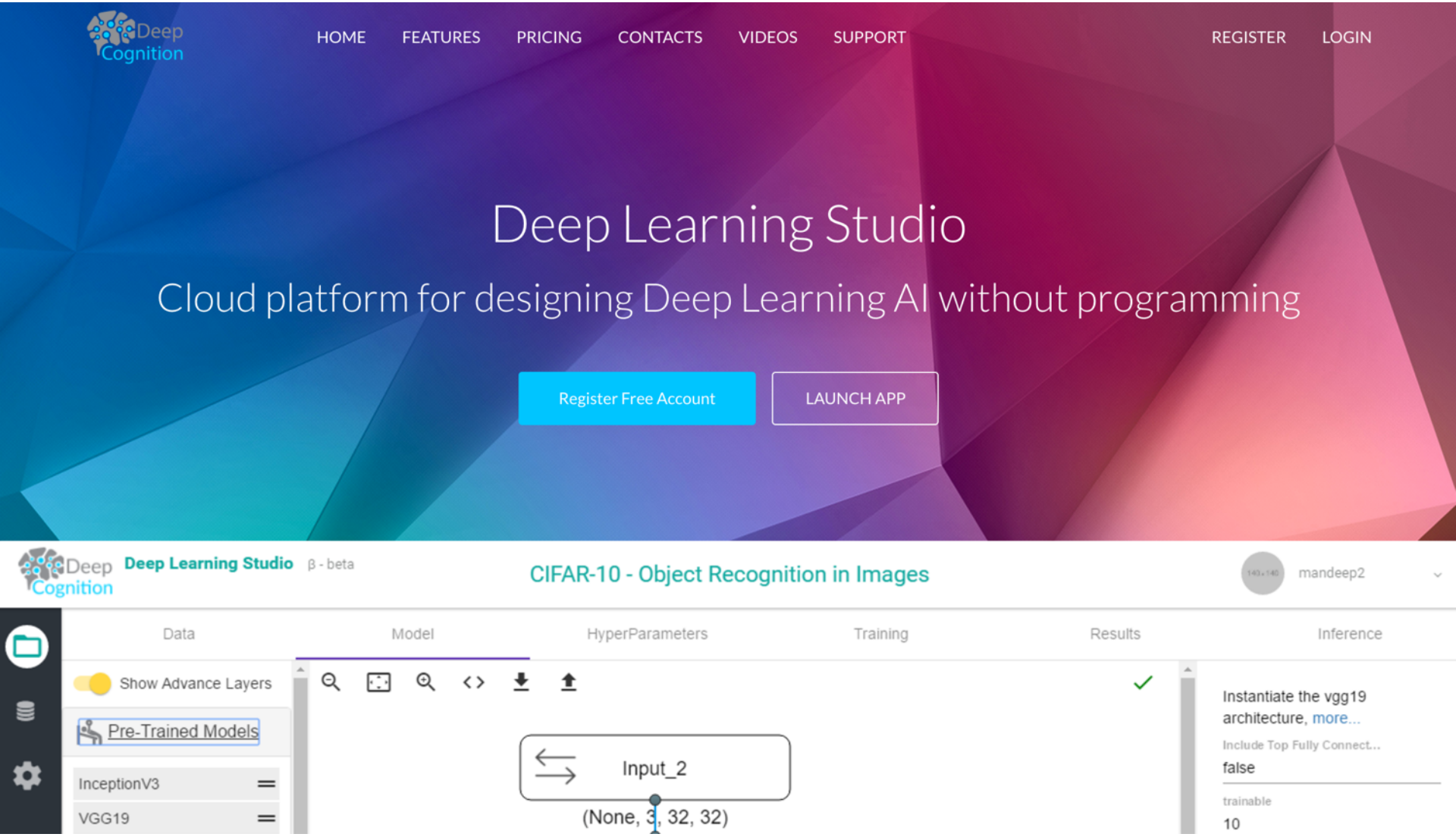
TensorFlow Playground



http://playground.tensorflow.org/

Deep Learning Studio

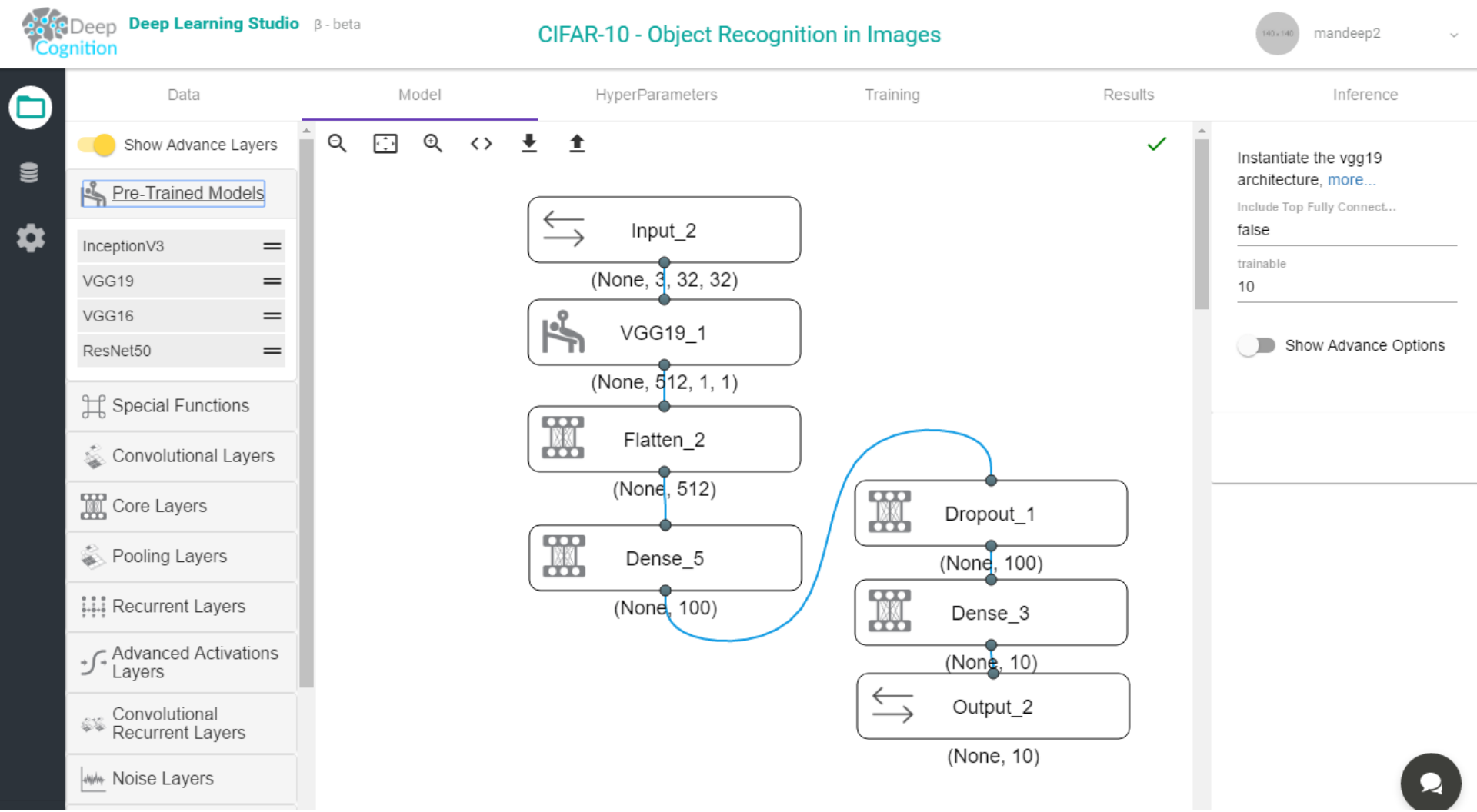
Cloud platform for designing Deep Learning AI without programming



http://deepcognition.ai/

Deep Learning Studio

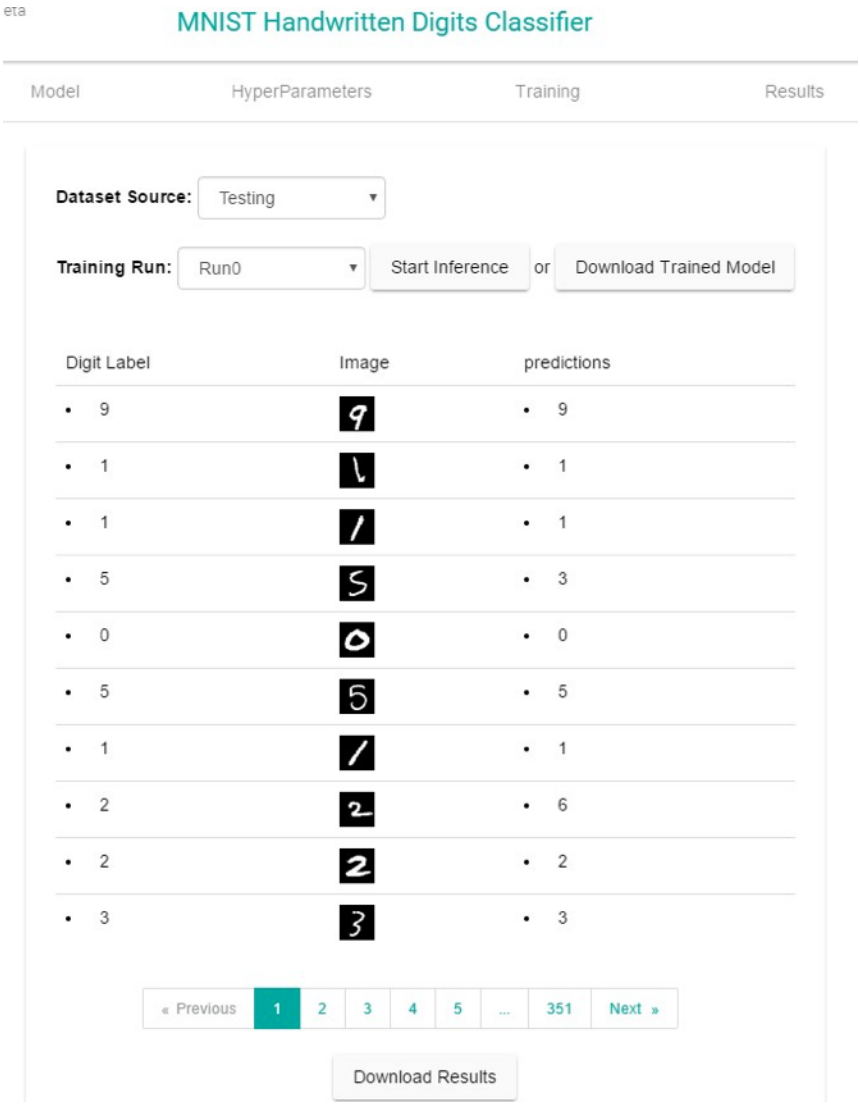
Cloud platform for designing Deep Learning AI without programming



http://deepcognition.ai/

Deep Learning Studio

Cloud platform for designing Deep Learning AI without programming



http://deepcognition.ai/

References

- Sunila Gollapudi (2016), Practical Machine Learning, Packt Publishing
- Sebastian Raschka (2015), Python Machine Learning, Packt Publishing
- Martin Gorner (2017), TensorFlow and Deep Learning without a PhD, Part 1 (Google Cloud Next '17), https://www.youtube.com/watch?v=u4alGiomYP4
- Martin Gorner (2017), TensorFlow and Deep Learning without a PhD, Part 2 (Google Cloud Next '17), https://www.youtube.com/watch?v=fTUwdXUFfI8
- Martin Gorner (2017), TensorFlow and Deep Learning without a PhD, https://goo.gl/pHeXe7, https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist
- Deep Learning Basics: Neural Networks Demystified, https://www.youtube.com/playlist?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU
- Deep Learning SIMPLIFIED, https://www.youtube.com/playlist?list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu
- TensorFlow: https://www.tensorflow.org/
- Theano: http://deeplearning.net/software/theano/
- Keras: http://keras.io/
- Deep Learning Studio: Cloud platform for designing Deep Learning AI without programming, http://deepcognition.ai/
- http://p.migdal.pl/2017/04/30/teaching-deep-learning.html
- https://github.com/leriomaggio/deep-learning-keras-tensorflow