아티스트를 위한 머신러닝 & 딥러닝

텐서플로를 활용한 딥러닝 #3

서울대학교 & V.DO / 김대식

art center nabi v.do

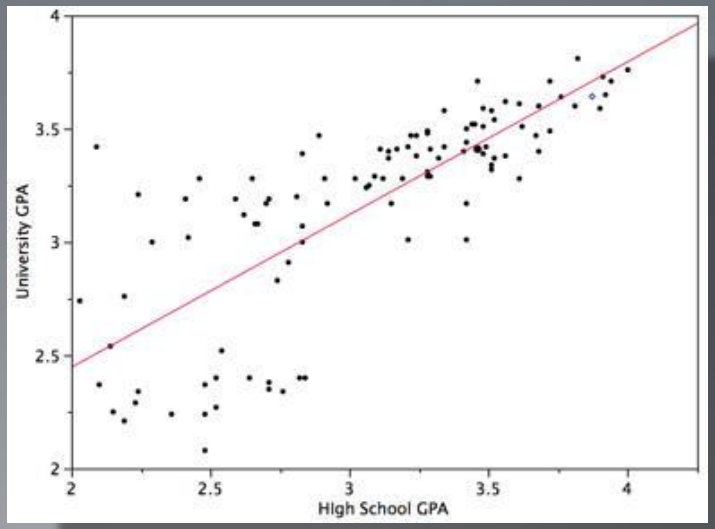
Recap

선형 회귀 분석(Linear Regression Analysis)

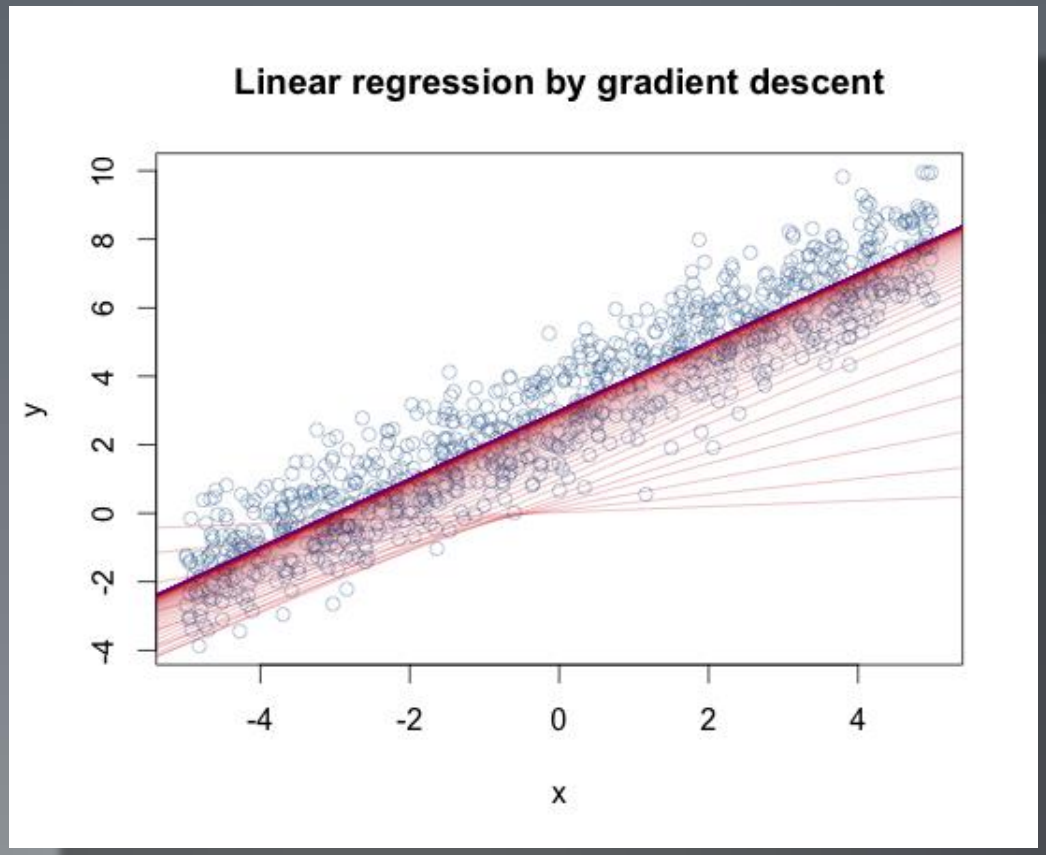
통계학에 가까운 개념

: X와 Y의 관계 분석, 관계에 대한 가설을 검정

→ 선형관계(Linear) 가정

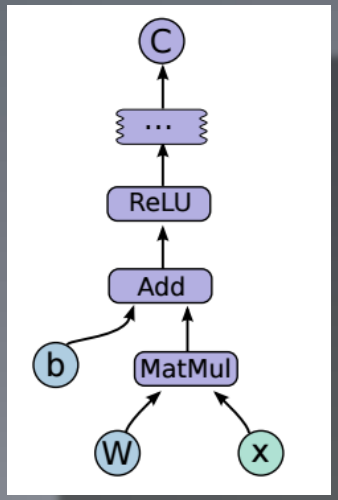


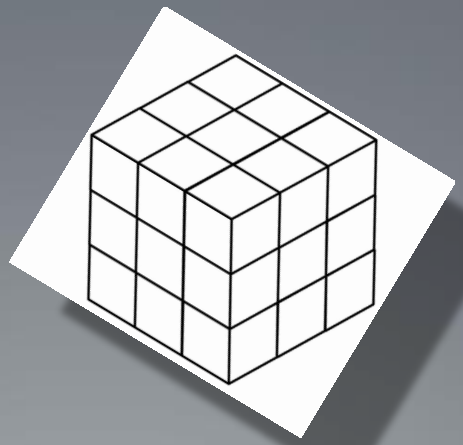
Gradient decent (경사 하강법)

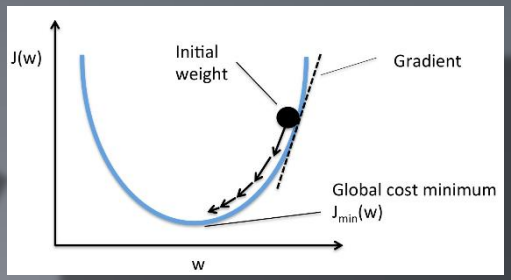


Tensorflow 기본

1.Graph를 그린다 → 2.데이터(Tensor)를 넣는다 → 3.Loss를 이용하여 학습







art center nabi v.do

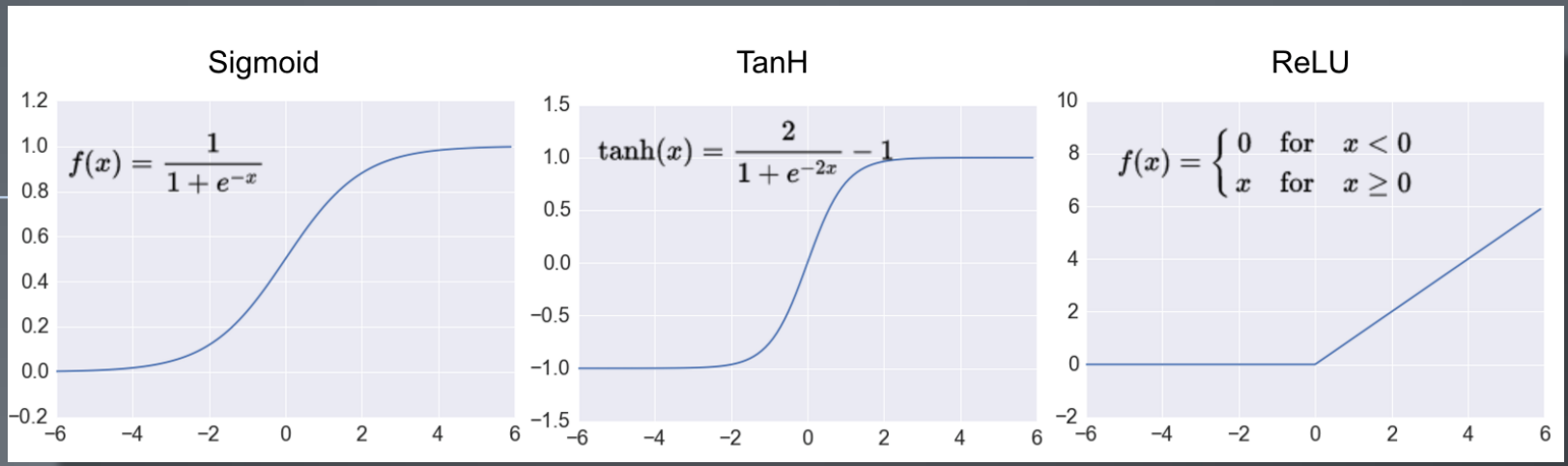
선형 회귀 ≒ 신경망

$$y = b + wx = wx$$

선형 회귀 + activation F
=신경망

$$y = \sigma(b + wx)$$

Activation 함수란?



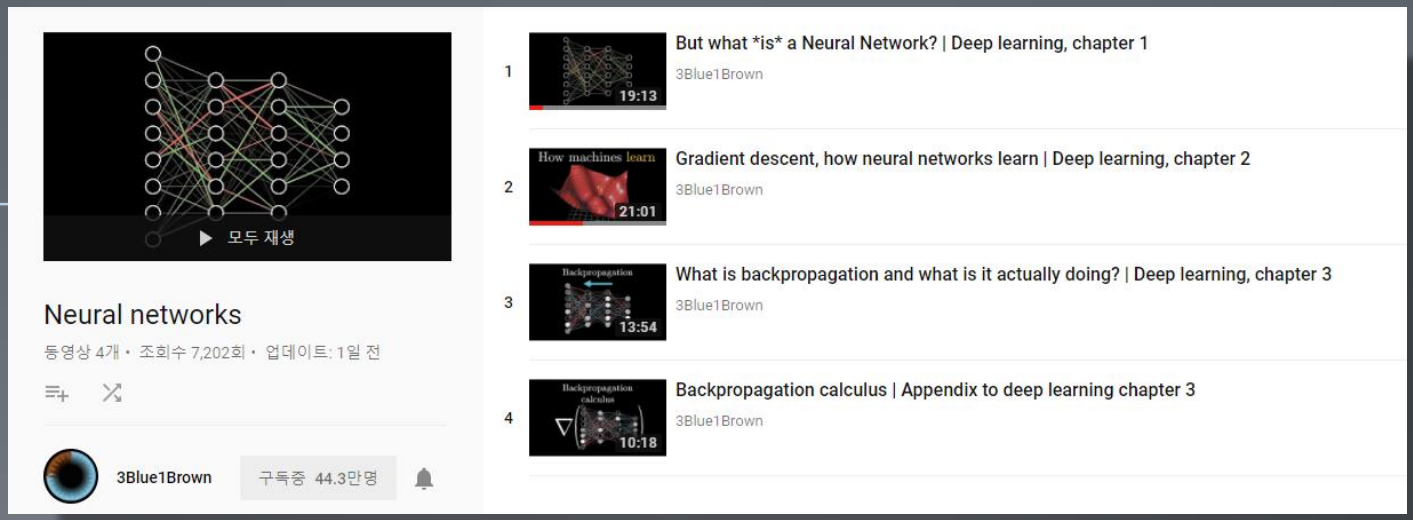
새로운 함수들이 쏟아짐

뉴럴 네트워크은 세상에 모든 함수를 표현해 낸다

*Approximation by superpositions of a sigmoidal function, by George Cybenko (1989). The result was very much in the air at the time, and several groups proved closely related results. Cybenko's paper contains a useful discussion of much of that work. Another important early paper is Multilayer feedforward networks are universal approximators, by Kurt Hornik, Maxwell Stinchcombe, and Halbert White (1989). This paper uses the Stone-Weierstrass theorem to arrive at similar results.

http://neuralnetworksanddeeplearning.com/chap4.html

What is a Neural network ?



https://youtu.be/aircAruvnKk

art center nabi v.do

다층 뉴럴네트워크
실습

art center nabi v.do

지도학습을 통한 hand written digit 분류

- MNIST datasets (다운로드 http://yann.lecun.com/exdb/mnist/)
- 오픈소스로 공개된 0 ~ 9 까지 사람 쓴 손 글씨 이미지 label된 image data
- Image size : 28 x 28 (784) pixels

[학습 데이터] - 60,000개
[테스트 데이터] - 10,000 개

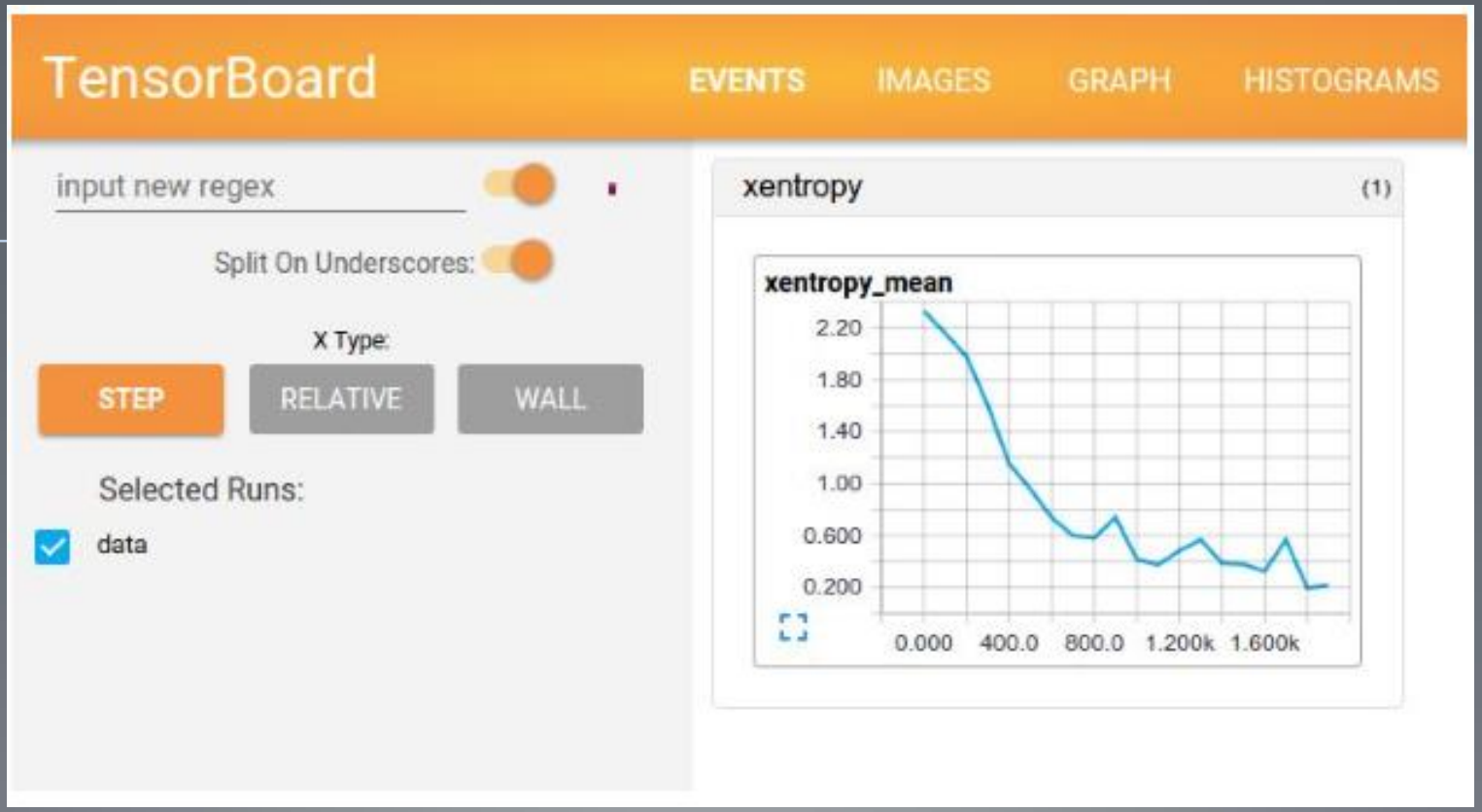
<MNIST 데이터베이스>

Image (문제)	Label (답)
train0.png	7
train1.png	3
train2.png	4
train3.png	6
train4.png	1
train5.png	8
train6.png	1

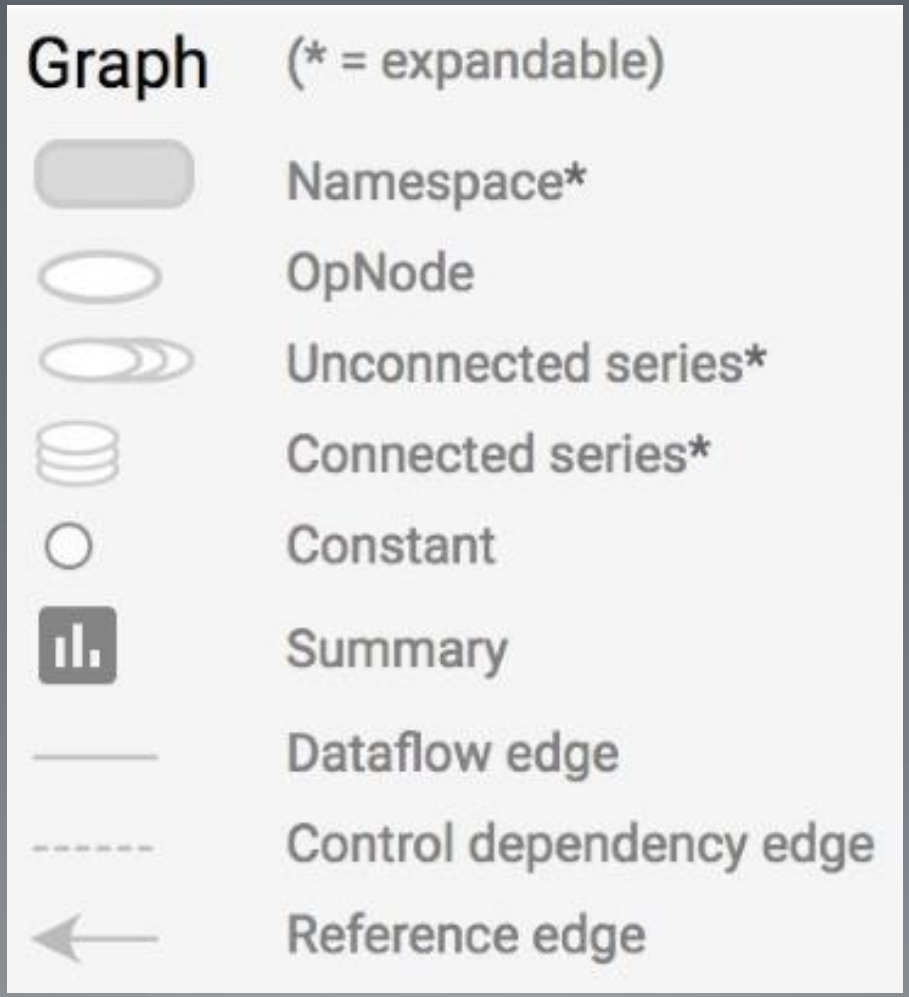
코드

https://github.com/chuckgu/nabi

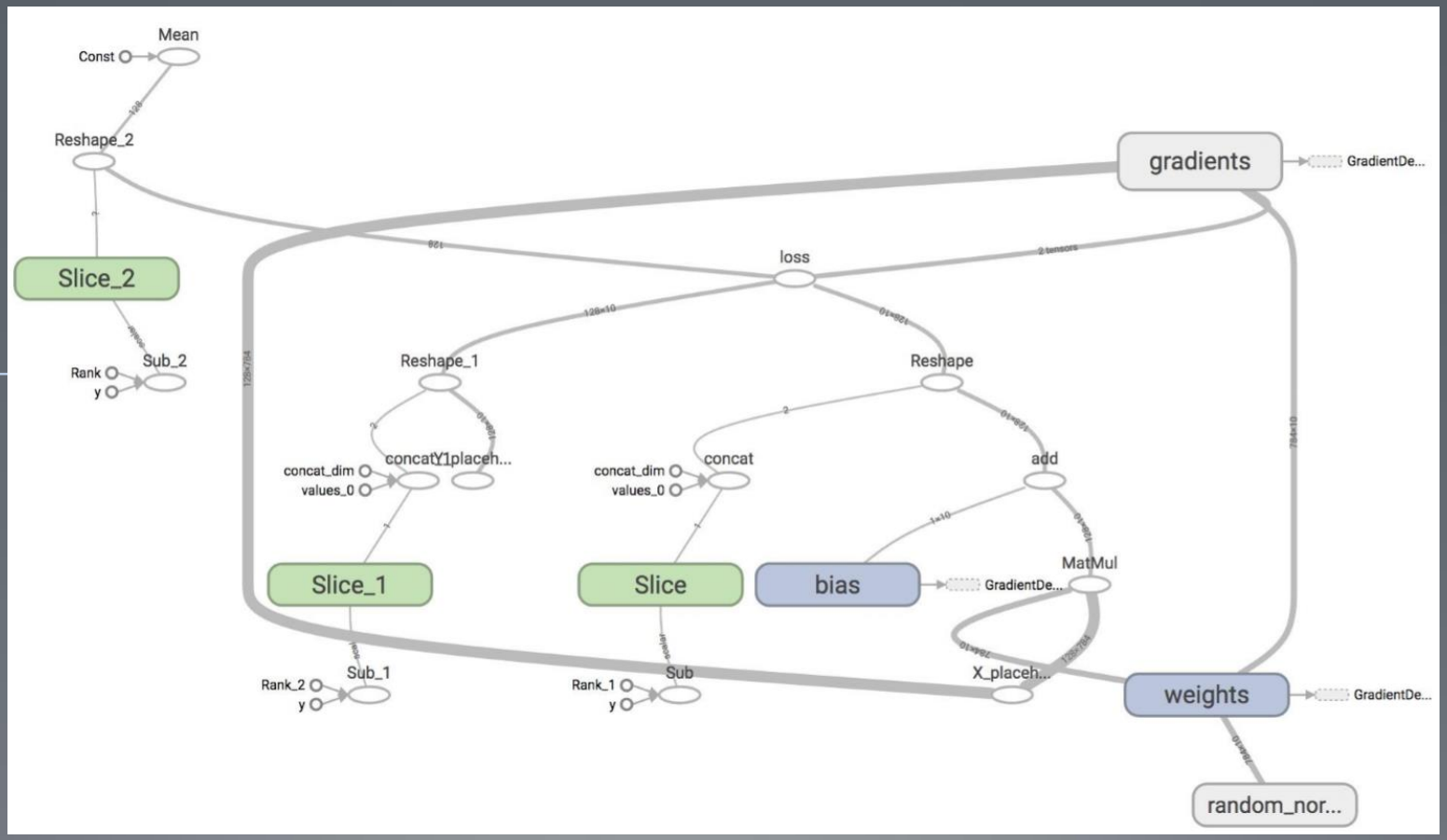
텐서보드
: 그래프 시각화 + 로그 시각화



텐서보드의 노드와 엣지의 종류



텐서보드의 그래프



art center nabi v.do

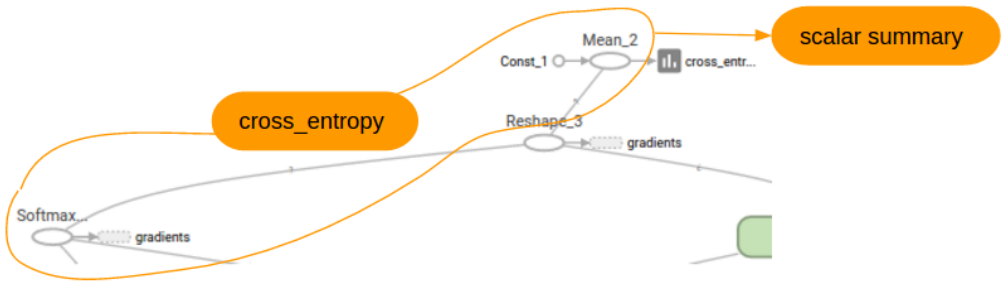
1. Summarize하기

Summary Operations에서 제공하는 오퍼레이션의 종류

- tensor_summary
- scalar
- histogram
- image

```python
# define loss: cross entropy
# cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=y_, logits=y))

# summary scalar for loss
tf.summary.scalar('cross_entropy', cross_entropy)
```

```
# cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=y_, logits=y))
```




```
# summary scalar for loss
tf.summary.scalar('cross_entropy', cross_entropy)
```

1. Summarize하기

```
# summary scalar for W
mean = tf.reduce_mean(W)
stddev = tf.sqrt(tf.reduce_mean(tf.square(W - mean)))
tf.summary.scalar('mean', mean)
tf.summary.scalar('stddev', stddev)
tf.summary.scalar('max', tf.reduce_max(W))
tf.summary.scalar('min', tf.reduce_min(W))
```

```
# summary histogram for W
tf.summary.histogram('histogram', W)
```

```
# summary image for x
image_shaped_input = tf.reshape(x, [-1, 28, 28, 1])
tf.summary.image('input', image_shaped_input, 10)
```

2. 병합하기

```
# merge all summary ops into a single op
merged = tf.summary.merge_all()
```

오퍼레이션을 개별로 평가할거 없이
합쳐서 하나의 오퍼레이션으로 처리 가능

3. 파일로 저장

```
# write summary to disk for TensorBoard visualization
train_writer = tf.summary.FileWriter('/home/itrocks/Downloads/train', sess.graph)
```

저장될 파일의 위치와 그래프를 표시하기 위해 세션의 그래프를 전달

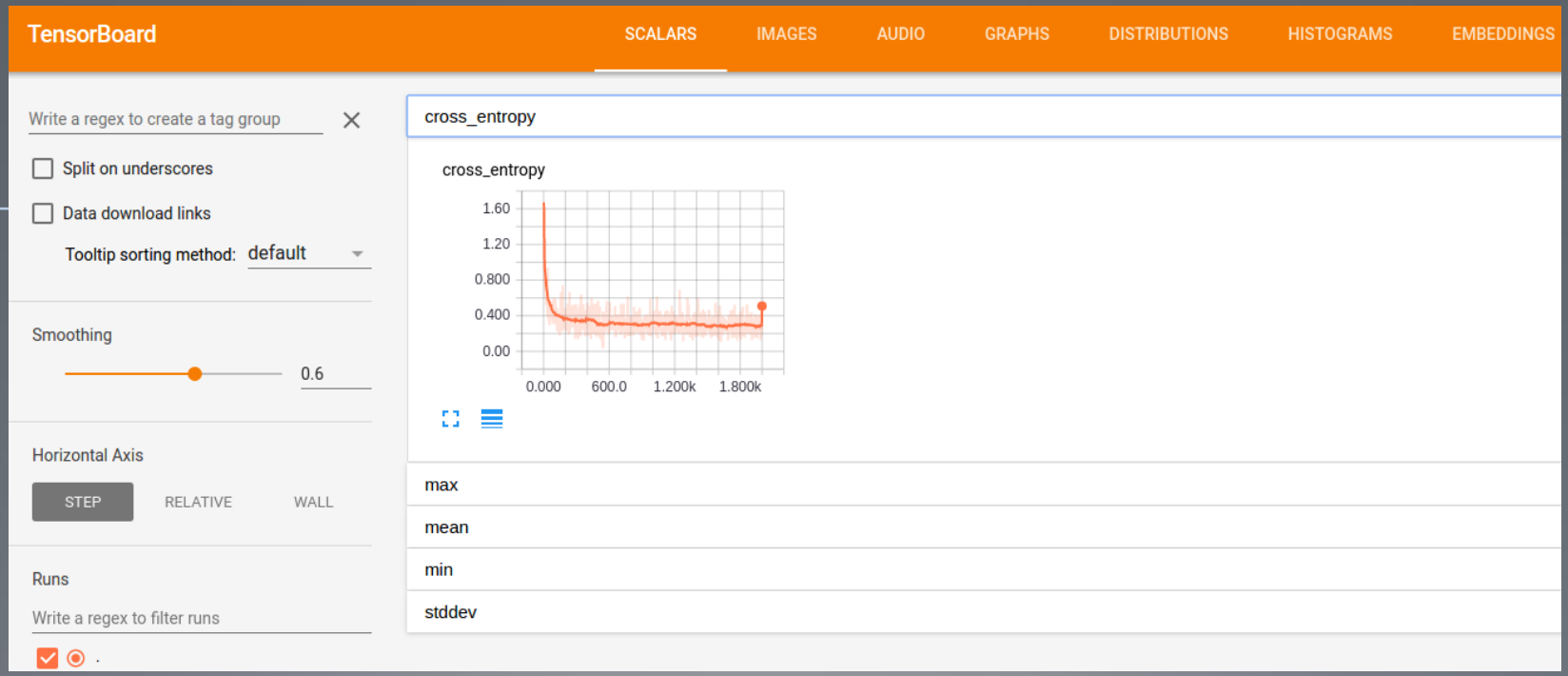
4. 평가 및 실행

```python
for i in range(2000):
  batch_xs, batch_ys = mnist.train.next_batch(100)
  # run summary op and train op
  summary, _ = sess.run([merged, train_step], feed_dict={x: batch_xs, y_: batch_ys})

  # write summary events to disk
  train_writer.add_summary(summary, i)

train_writer.close()
```

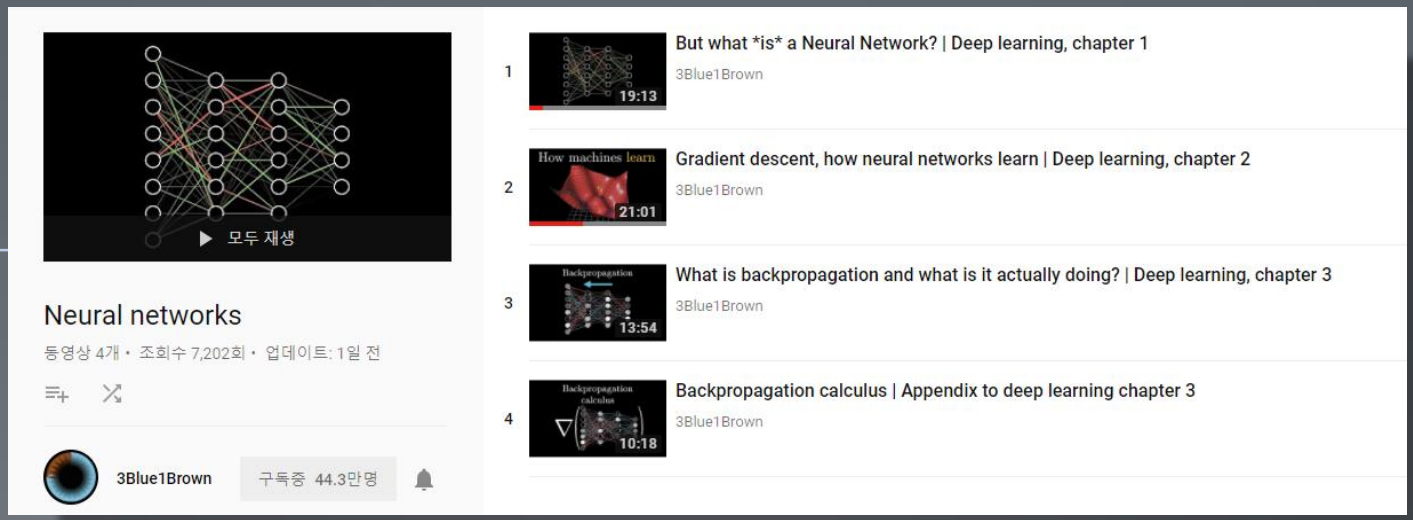
```
$ tensorboard --logdir=/home/itrocks/Downloads/train
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcublas.so.8.0 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcudnn.so.5 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcufft.so.8.0 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcurand.so.8.0 locally
Starting TensorBoard 41 on port 6006
(You can navigate to http://127.0.1.1:6006)
```

4. 평가 및 실행

기본주소(http://localhost:6006)로 웹브라우저에 접속



Gradient descent



https://youtu.be/IHZwWFHWa-w

art center nabi v.do

감사합니다