
강의 목표

1. TensorBoard를 이용한 학습과정 Visualization 필요성을 이해한다.
2. TensorFlow2.0을 이용해서 TensorBoard 로그를 저장하고 시각화하는 방법을 살펴본다.

학습과정 Visualization의 필요성과 TensorBoard

- 터미널 로그 등을 이용해서 학습 과정을 모니터링 할 경우, 한눈에 학습 과정의 문제점을 파악하기 쉽지 않습니다.
- 따라서 TensorFlow에서는 학습과정 시각화를 위해 **TensorBoard**라는 기능을 제공합니다.

```
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tensorflow/python/util/deprecation.py:574: calling map_fn_v2 (from tensorflow.python.ops.map_fn) with dtype is deprecated and will be removed in a future version.
```

Instructions for updating:

Use fn_output_signature instead

반복 (Epoch): 100, 트레이닝 데이터 정확도 : 0.920000

반복 (Epoch): 200, 트레이닝 데이터 정확도 : 0.980000

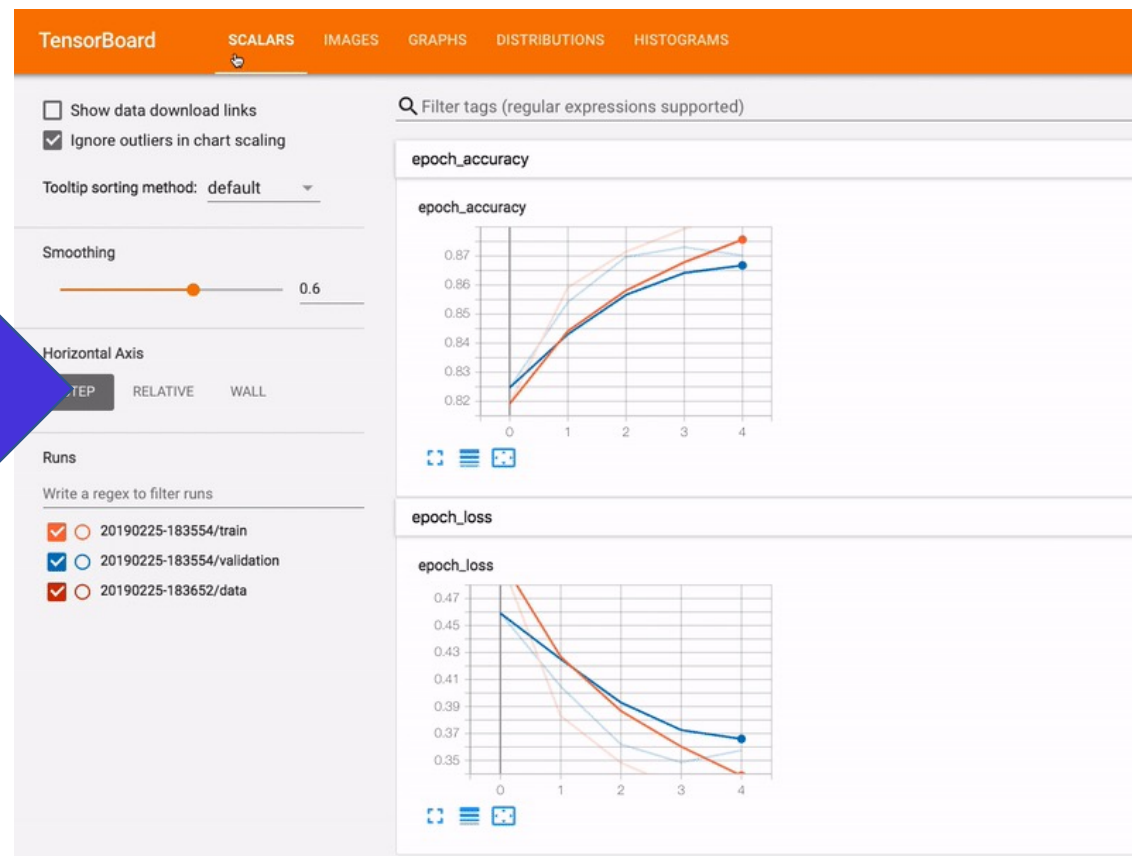
반복 (Epoch): 300, 트레이닝 데이터 정확도 : 0.920000

반복 (Epoch): 400, 트레이닝 데이터 정확도 : 0.940000

반복 (Epoch): 500, 트레이닝 데이터 정확도 : 0.980000

반복 (Epoch): 600, 트레이닝 데이터 정확도 : 0.960000

□



tf.summary 로그의 형태

- summary 로그의 형태

- ① **tf.summary.scalar** : scalar값 형태의 로그 저장
- ② **tf.summary.histogram** : histogram 형태의 로그 저장
- ③ **tf.summary.image** : 이미지 형태의 로그 저장

TensorBoard 로그를 저장하는 코드 추가방법

- TensorFlow 2.0을 이용해서 텐서보드(TensorBoard) 로그를 저장하는 방법은 다음과 같습니다.

[텐서보드 로그를 저장하는 법]

- ① 인자값으로 텐서보드 **로그 파일을 저장할 경로**를 지정해서 File Writer 생성

e.g.) `summary_writer = tf.summary.create_file_writer('./tensorboard_log')`

- ② 요약 정보를 남기고 싶은 값을 Writer scope 내에서 `tf.summary.*` API로 추가

e.g.)

```
with summary_writer.as_default():  
    tf.summary.scalar('loss', loss, step=optimizer.iterations)
```

TensorBoard 실행방법

- TensorBoard를 실행하는 터미널 명령어는 아래와 같습니다. (logdir에 summary file이 있는 위치를 argument로 지정합니다.)

tensorboard --logdir=path\to\log-directory

- 웹브라우저에서 TensorBoard를 실행결과를 보려면 URL창에 아래 주소를 입력합니다. (기본 실행 port는 6006 포트입니다.)

localhost:6006

TensorBoard 로그 저장 예제

- TensorBoard 를 이용해서 학습결과를 시각화하는 방법을 살펴봅시다.
- https://github.com/solaris33/deep-learning-tensorflow-book-code/blob/master/Appendix/mnist_classification_using_cnn_v2_keras_with_tensorboard.py

Chapter 12 - 강화학습(Reinforcement Learning)

- DQN을 이용한 게임 에이전트 구현 - CatchGame ([Code](#)) ([TF v2 Code](#)) ([TF v2 Keras Code](#))

Chapter 13 - 파인튜닝(Fine-Tuning)과 Pre-Trained 모델을 이용해서 실제 문제 해결하기

- Inception v3 Retraining을 이용해서 나만의 분류기 만들어보기 ([Code](#))
- Pre-Trained 모델을 이용해서 Object Detection 수행하기 ([Code](#))

Appendix

- CNN을 이용한 MNIST 숫자 분류기 구현 + 텐서보드(TensorBoard) ([Code](#))

Thank you!
