

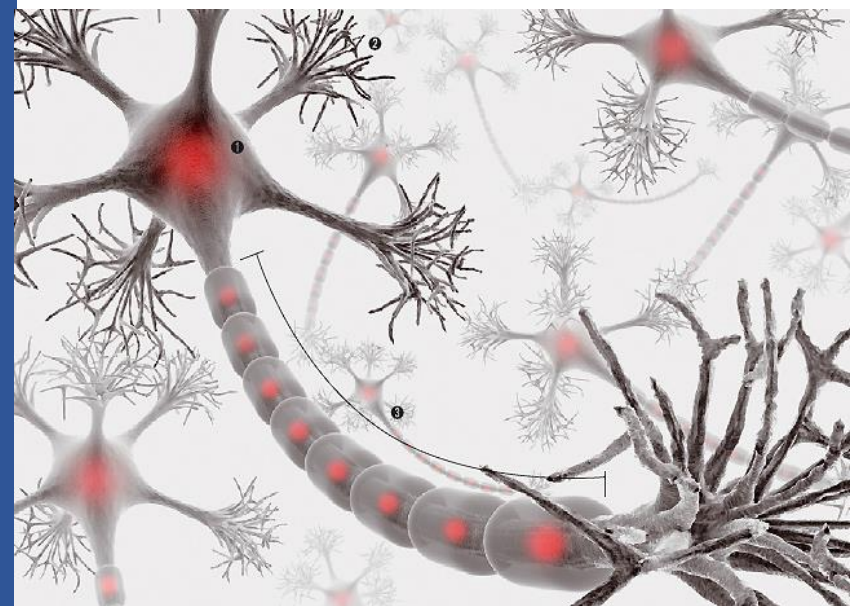
# TensorBoard 모니터링

## 학습 목표

- 신경망 모델을 훈련하면서 TensorBoard로 모니터링 해본다.

## 주요 내용

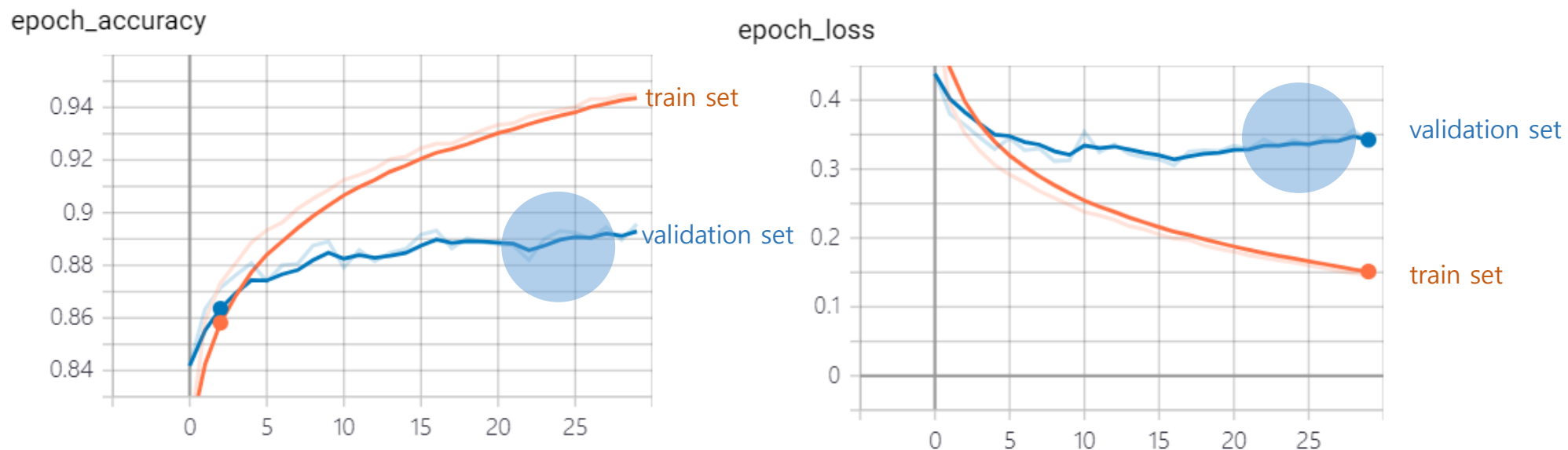
1. TensorBoard 소개
2. TensorBoard Callback
3. TensorFlow Summary 문제



# 문제

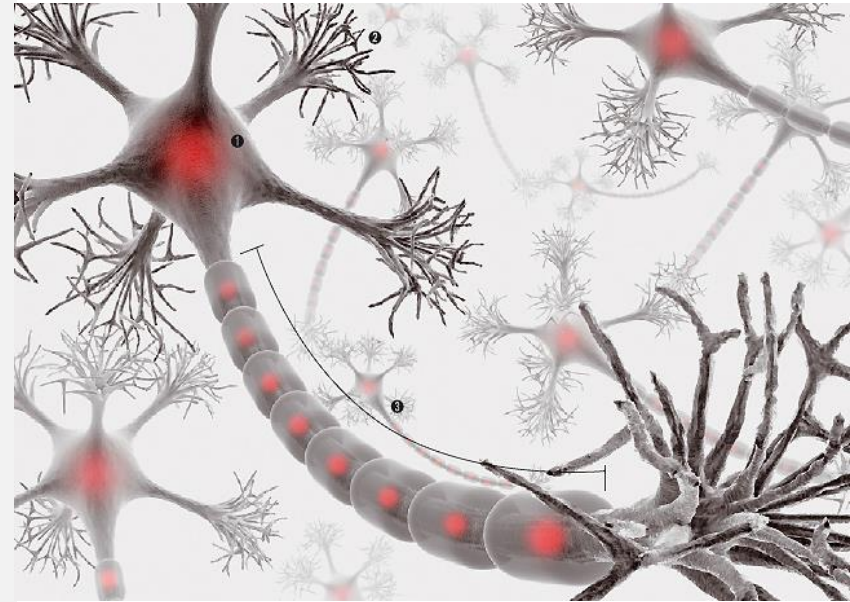


**Fashion MNIST 모델의 훈련 과정을 Tensorboard로 모니터링 해보자!**



단, `model.fit()`으로 훈련하므로 `tf.keras.callbacks.TensorBoard` 콜백을 사용하시오.  
`validation_split=0.1`를 사용해서 검증 그래프도 같이 나오게 하시오.

# 1 TensorBoard 소개

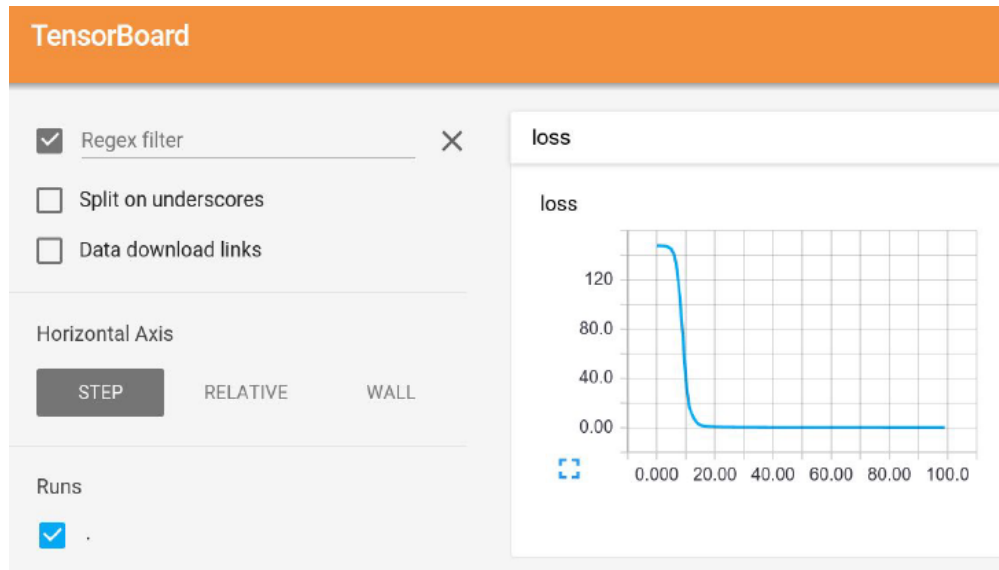


# TensorBoard

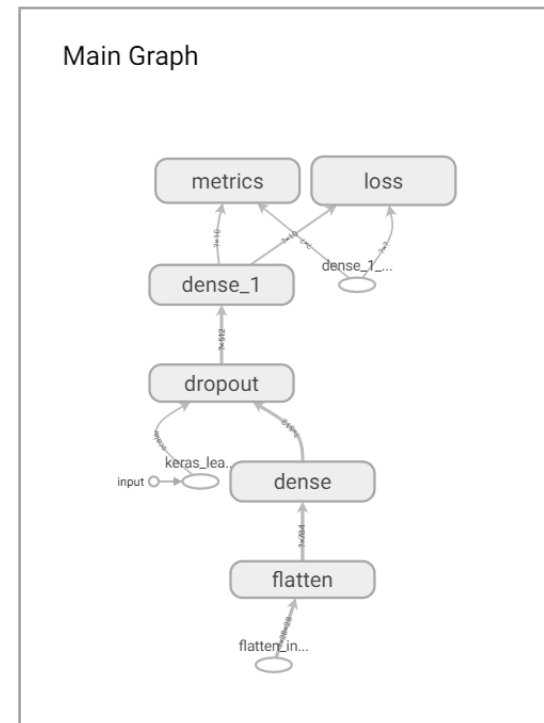
## TensorFlow에 포함된 Graph Visualization 소프트웨어

- Metrics 요약 그래프
- 계산 그래프 시각화
- 가중치/출력 히스토그램
- 프로파일링

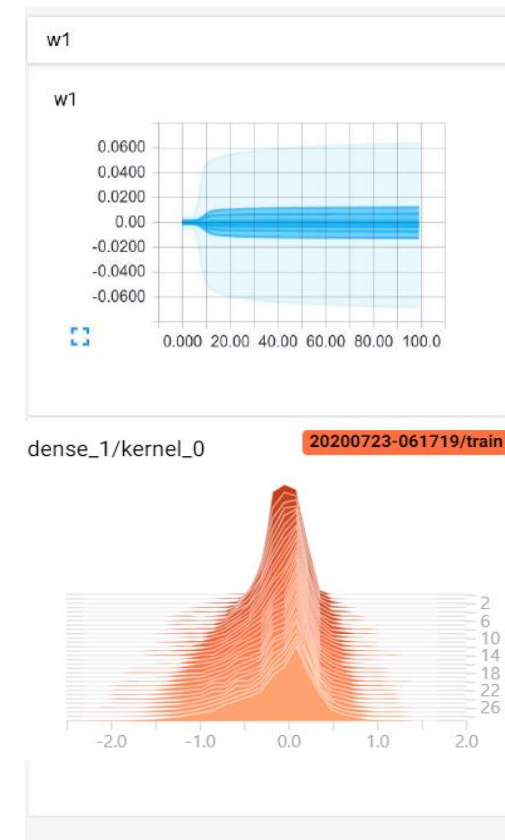
Loss 요약 그래프



계산 그래프 시각화



가중치 그래프/히스토그램



# TensorBoard 아키텍처

대량 데이터를 모니터링 하는 방식으로 File로 로깅한 후에 별도의 뷰어를 통해 확인



모니터링 유형 별로 메뉴 구성

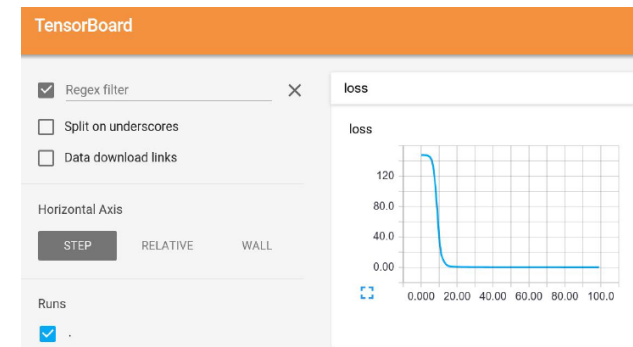
## 실행 프로그램

- `tf.summary.scalar(name, scalar)`
- `tf.summary.image(name, image)`
- `tf.summary.histogram(name, histogram)`
- `tf.summary.audio(name, audio)`
- `tf.summary.text(name, text)`

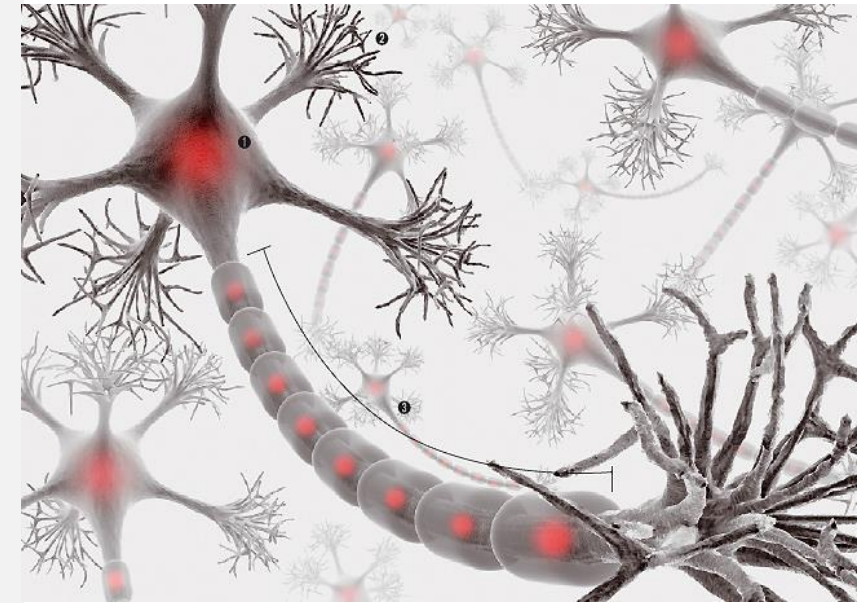
Scalar, Image,  
Histogram, audio, text

TensorBoard  
Log Data

## TensorBoard Viewer



## 2 TensorBoard Callback



# Tensorboard Callback

## Tensorboard 콜백 적용

```
import os
import datetime
model = create_model()
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

log_dir= os.path.join(os.getcwd(), 'logs', datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)

model.fit(x=x_train,
          y=y_train,
          epochs=5,
          validation_data=(x_test, y_test),
          callbacks=[tensorboard_callback])
```

Loss & accuracy 로깅, 각 Layer 별 가중치와 activation histogram 추적

## 참고 tf.keras.callbacks.TensorBoard

```
tf.keras.callbacks.TensorBoard(  
    log_dir='logs', histogram_freq=0, write_graph=True, write_images=False,  
    update_freq='epoch', profile_batch=2, embeddings_freq=0,  
    embeddings_metadata=None, **kwargs  
)
```

- **log\_dir**: 로깅 디렉토리 the path of the directory where to save the log files to be parsed by TensorBoard.
- **histogram\_freq**: 히스토그램 생성 주기 (epoch 단위) frequency (in epochs) at which to compute activation and weight histograms for the layers of the model. If set to 0, histograms won't be computed. Validation data (or split) must be specified for histogram visualizations.
- **write\_graph**: whether to visualize the graph in TensorBoard. The log file can become quite large when write\_graph is set to True.
- **write\_images**: whether to write model weights to visualize as image in TensorBoard.
- **update\_freq**: 로깅 주기 'batch' or 'epoch' or integer. When using 'batch', writes the losses and metrics to TensorBoard after each batch. The same applies for 'epoch'. If using an integer, let's say 1000, the callback will write the metrics and losses to TensorBoard every 1000 batches. Note that writing too frequently to TensorBoard can slow down your training.
- **profile\_batch**: Profile the batch to sample compute characteristics. By default, it will profile the second batch. Set profile\_batch=0 to disable profiling. Must run in TensorFlow eager mode.



# TensorBoard Viewer

## Command Line에서 실행

```
> tensorboard --logdir {log_dir}
```

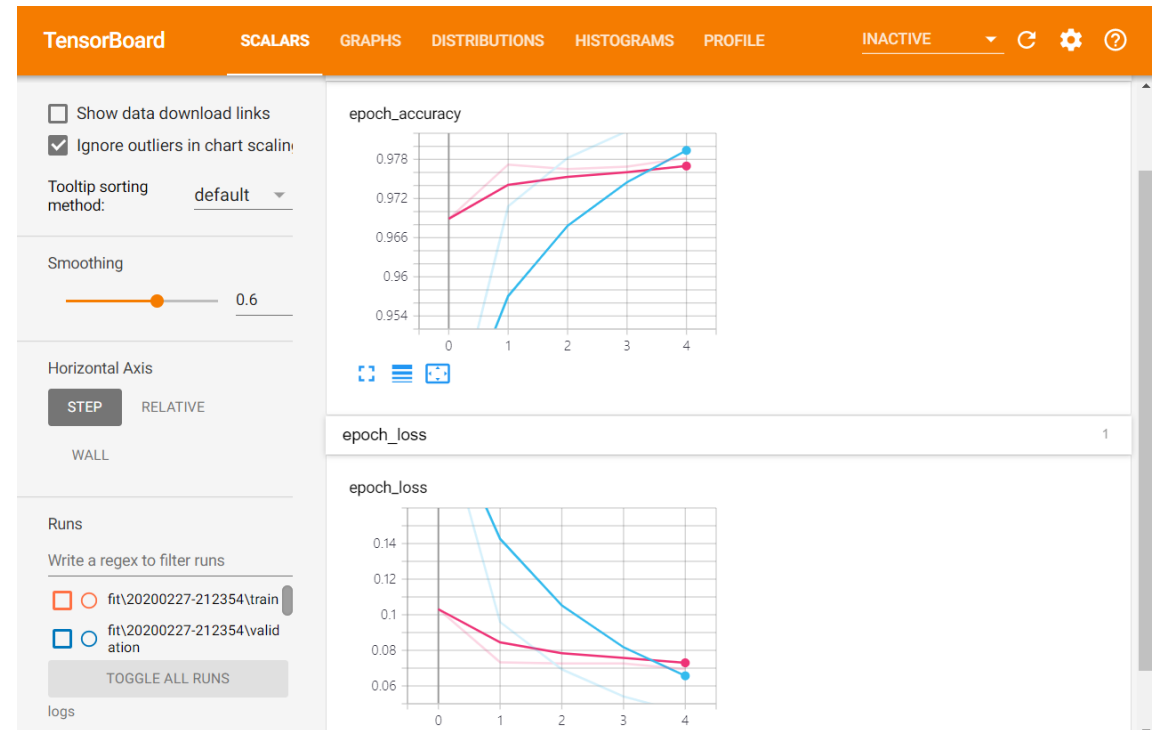
TensorBoard 2.0.0 at <http://localhost:6006/> (Press CTRL+C to quit)

- URL을 복사해서 브라우저에서 <http://localhost:6006/> 실행

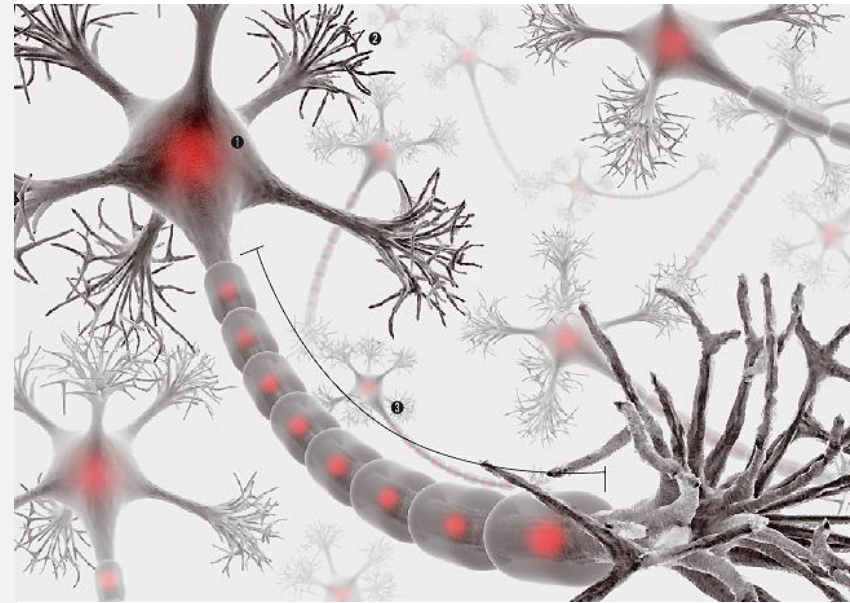
## Jupyter Notebook에서 실행

```
%load_ext tensorboard
%tensorboard --logdir {log_dir}
```

- Jupyter notebook extension으로 Tensorboard 로딩
- Jupyter notebook 내에서 Tensorboard Viewer 렌더링



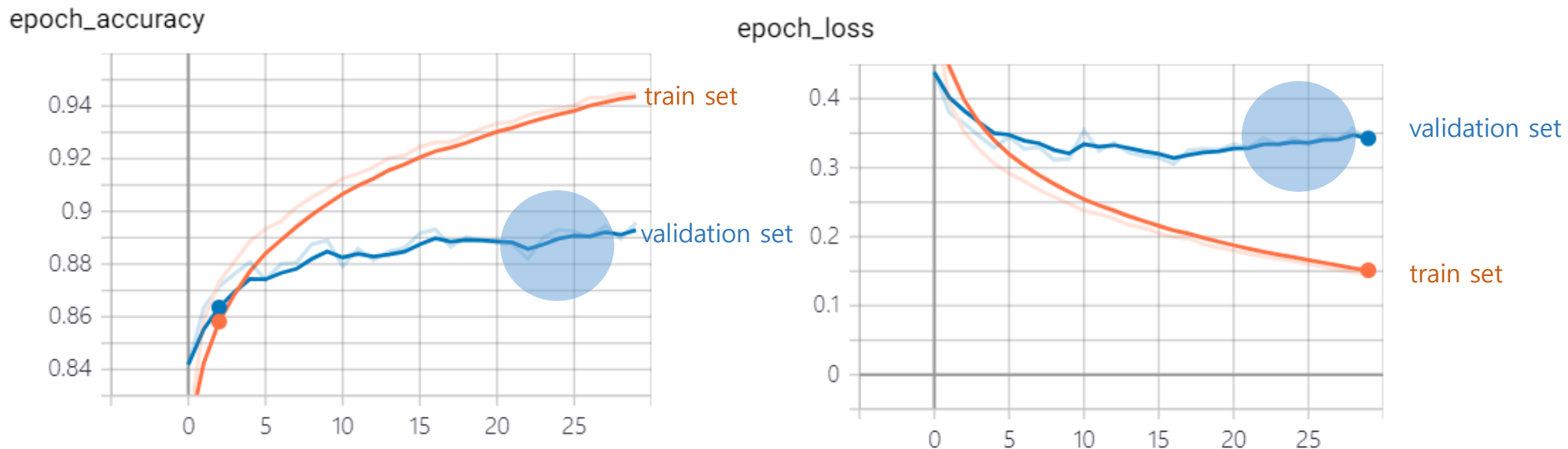
# 3 Tensorboard Callback 실습



# 문제



**Fashion MNIST 모델의 훈련 과정을 Tensorboard로 모니터링 해보자!**



단, `model.fit()`으로 훈련하므로 `tf.keras.callbacks.TensorBoard` 콜백을 사용하시오.  
`validation_split=0.1`를 사용해서 검증 그래프도 같이 나오게 하시오.

# 모델 훈련 (문제)



Tensorboard Callback을 이용해서 훈련 과정을 모니터링 해보시오.

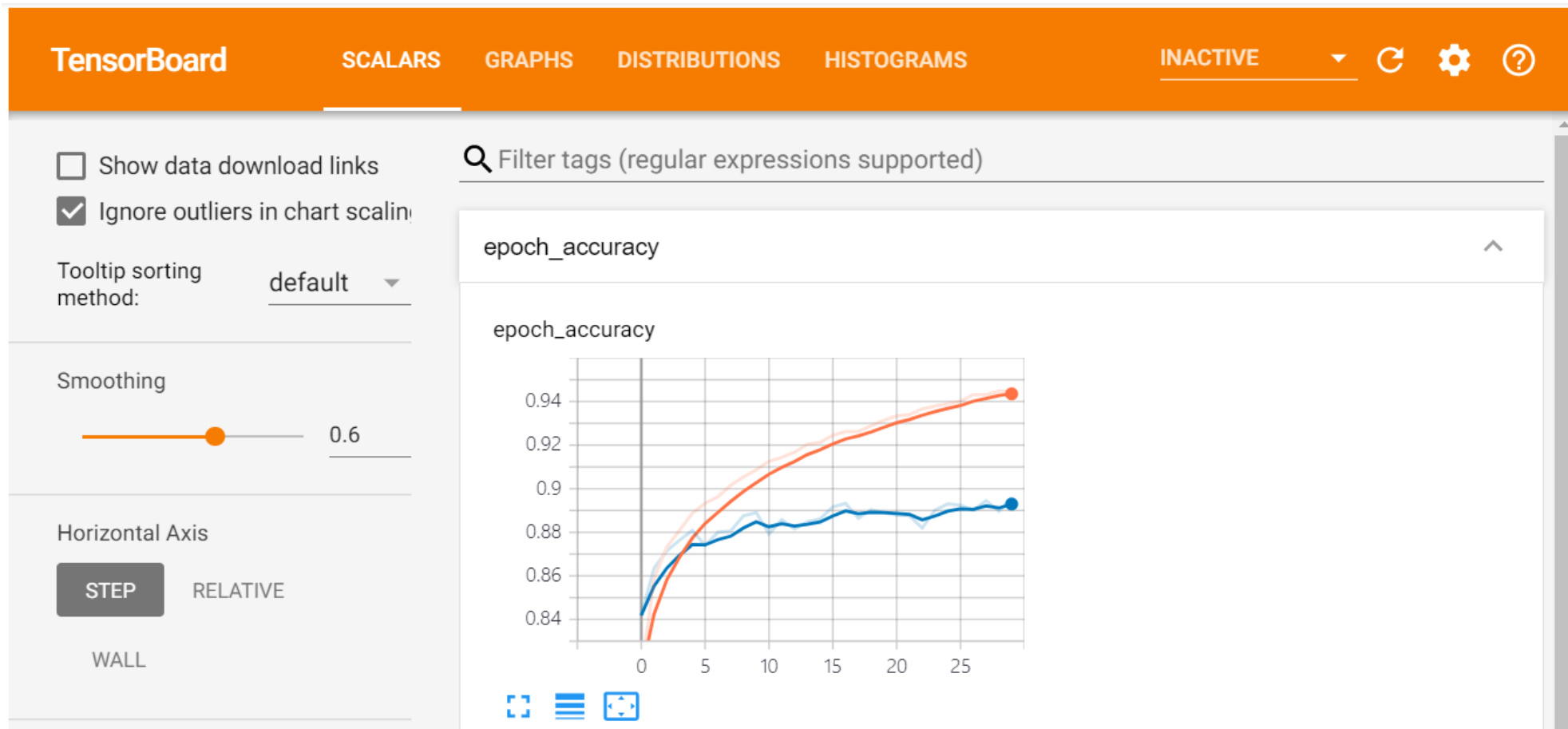
```
import datetime
import os
batch_size = 64

# Tensorboard
tensorboard_callback = # your code

model.fit(train_images, train_labels,
          batch_size=batch_size,
          epochs=30,
          validation_split=0.1,
          callbacks=[tensorboard_callback])
```

# 모델 훈련 모니터링

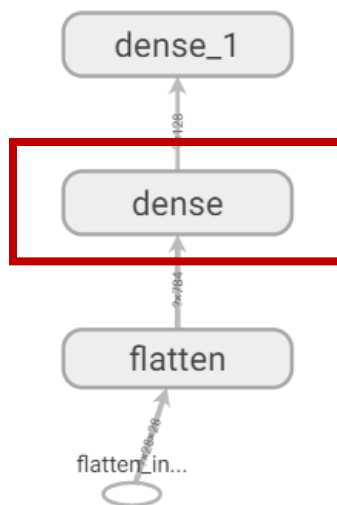
```
%tensorboard --logdir logs
```



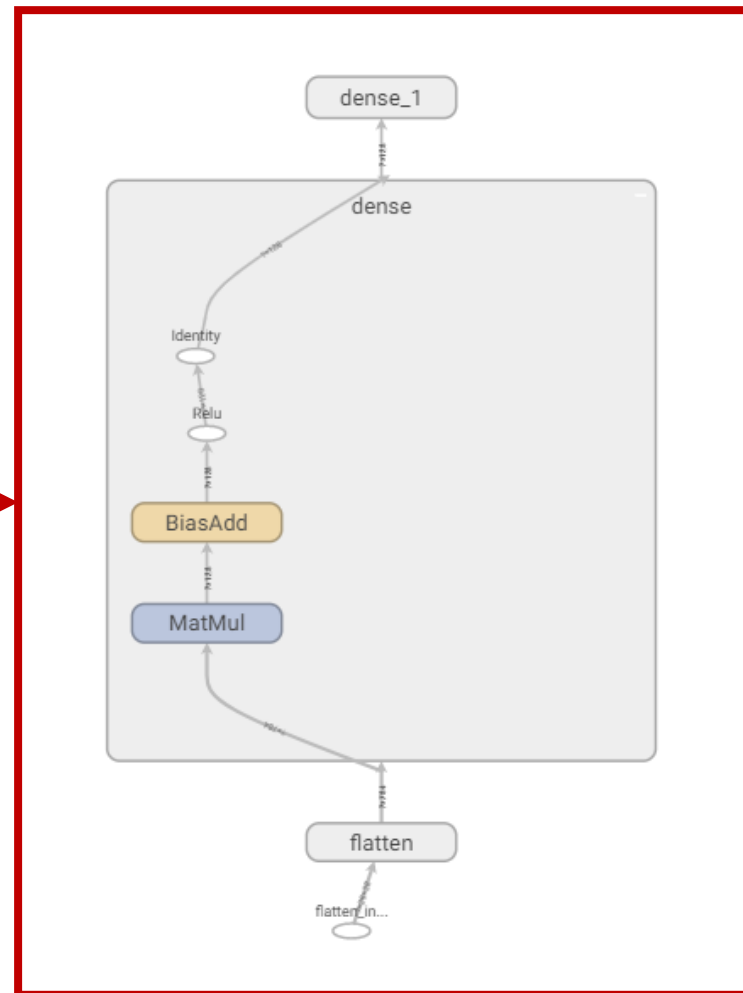
# 모델 훈련 모니터링

**Graph :** 훈련 그래프 시각화

Main Graph



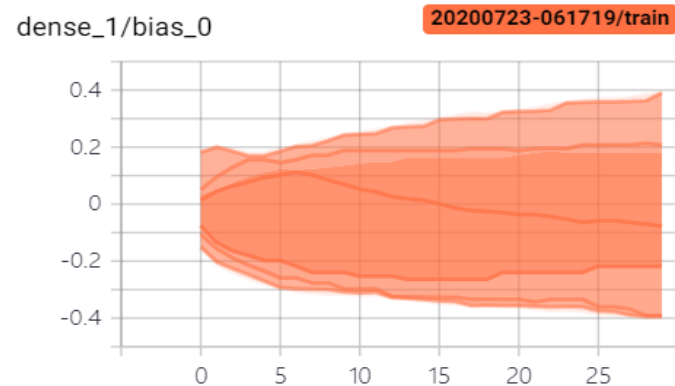
Auxiliary Nodes



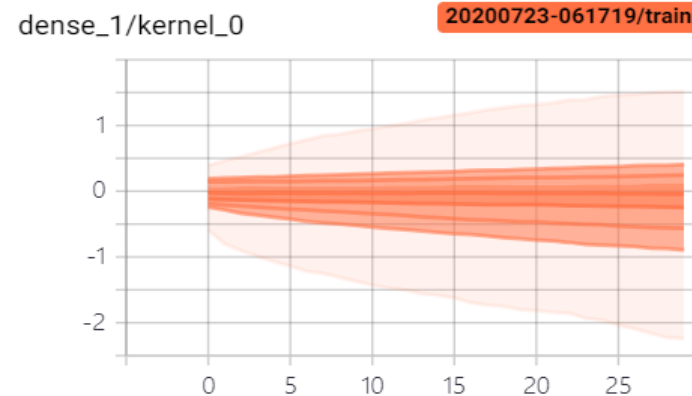
# 모델 훈련 모니터링

Graph

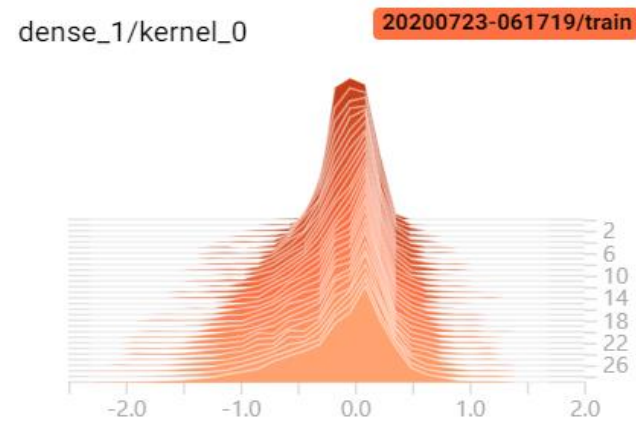
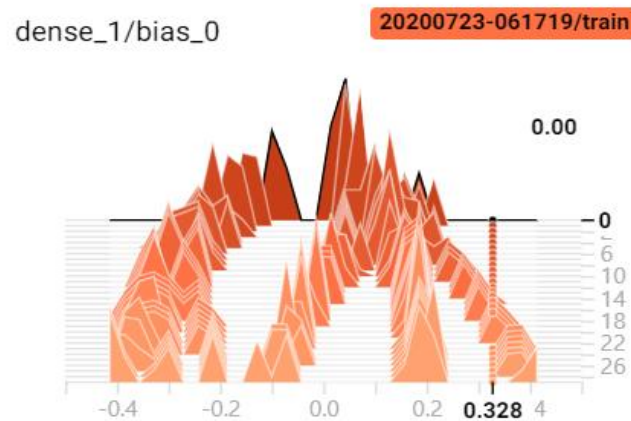
Bias



Weight



Histogram



**Thank you!**

