

TensorBoard

Comprehensive TensorBoard usage untuk experiment tracking, performance profiling, dan model monitoring. Essential tool untuk ML development lifecycle.

Data Visualization dan Scalars Tracking

Basic Setup:

```
log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir, histogram_freq=1)

model.fit(train_ds, validation_data=val_ds, callbacks=[tensorboard_callback])
```

Custom Scalars:

```
class CustomMetricsCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        with self.writer.as_default():
            tf.summary.scalar('learning_rate', self.model.optimizer.lr, step=epoch)
            tf.summary.histogram('weights', self.model.weights[0], step=epoch)
```

Model Tracking dan Hyperparameter Tuning

HPParams Plugin:

```
HP_NUM_UNITS = hp.HParam('units', hp.Discrete([128, 256, 512]))
HP_DROPOUT = hp.HParam('dropout', hp.RealInterval(0.0, 0.5))
HP_OPTIMIZER = hp.HParam('optimizer', hp.Discrete(['adam', 'sgd']))

def run_experiment(hparams):
    model = create_model(hparams[HP_NUM_UNITS], hparams[HP_DROPOUT])
    model.compile(optimizer=hparams[HP_OPTIMIZER], ...)

    with summary_writer.as_default():
        hp.hparams_config(hparams, trial_id=trial_id, metrics=METRICS)

    model.fit(...)
    val_acc = ...

    with summary_writer.as_default():
        tf.summary.scalar('val_accuracy', val_acc, step=0)
```

Performance Profiling

Profiler Plugin:

```
%load_ext tensorboard
%tensorboard --logdir logs/fit
```

```
# Python profiling
tf.profiler.experimental.start(log_dir)
model.fit(train_ds)
tf.profiler.experimental.stop()
```

Input Pipeline Optimization:

```
# Memory viewer menunjukkan bottlenecks
# Optimal pattern: cache() -> shuffle() -> batch() -> prefetch()
```

Advanced Features

Mixed Precision Training:

```
tf.keras.mixed_precision.set_global_policy('mixed_float16')
```

Word Embeddings Projector:

```
# Project 2D/3D embeddings visualizations
with embedding_writer.as_default():
    tf.summary.create_embedding_matrix(
        vocab, metadata=vocab,
        embedding=tf.keras.layers.Embedding(vocab_size, 128).weights[0]
    )
```

Kesimpulan

TensorBoard adalah mission-critical tool untuk systematic ML development dan production monitoring.¹