

Penjelasan Mengenai Arsitektur Model



Nama Anggota :

Fatwaraga Rafsanjani (3337210031)

Elga Syahira (3337210018)

Berliani Nasywa Fitri (3337210067)

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNIK
UNIVERSITAS SULTAN AGENG TIRTAYASA
2023**

Custom Text Classification using TensorFlow Lite

Install TF Lite Model Maker

Install **TensorFlow Lite Model Maker** library. TF Lite Model Maker memudahkan pelatihan model pada kumpulan data khusus dan mengurangi waktu pelatihan dengan menggunakan Pembelajaran Transfer pada model yang telah dilatih sebelumnya..

```
!pip install -q tflite-model-maker
```

Instal perpustakaan yang diperlukan

```
import numpy as np
from numpy.random import RandomState
import pandas as pd
import os

from tflite_model_maker import model_spec
from tflite_model_maker import text_classifier
from tflite_model_maker.config import ExportFormat
from tflite_model_maker.config import QuantizationConfig
from tflite_model_maker.text_classifier import AverageWordVecSpec
from tflite_model_maker.text_classifier import DataLoader

import tensorflow as tf
assert tf.__version__.startswith('2')
tf.get_logger().setLevel('ERROR')
```

Impor dataset

Impor kumpulan data pelatihan dan pengujian dan membacanya sebagai file CSV menggunakan perpustakaan Pandas.

```
df_train = pd.read_csv('train.csv', error_bad_lines=False,
engine="python")

df_test = pd.read_csv('test.csv', error_bad_lines=False,
engine="python")
```

Lihat kumpulan data

Periksa kumpulan data Anda dan lihat apakah data tersebut diimpor dengan benar atau tidak.

```
In [ ]:      df_train.head()
```

```
In [ ]:      df_test.head()
```

Pilih arsitektur model

Pilih salah satu model arsitektur pilihan Anda dan beri komentar pada sisanya. Setiap arsitektur model berbeda satu sama lain dan akan memberikan hasil yang berbeda. Model MobileBERT membutuhkan lebih banyak waktu untuk dilatih karena arsitekturnya cukup kompleks. Namun, jangan ragu untuk bermain-main dengan arsitektur yang berbeda hingga Anda menemukan hasil terbaik.

```
spec = model_spec.get('average_word_vec')
# spec = model_spec.get('mobilebert_classifier')
# spec = model_spec.get('bert_classifier')
# spec = AverageWordVecSpec(wordvec_dim=32)
```

Sesuaikan hyperparameter model MobileBERT

Note: Jalankan sel ini hanya jika Anda telah memilih arsitektur model MobileBERT Classifier.

Parameter model yang dapat Anda sesuaikan adalah:

- seq_len: Panjang urutan yang akan dimasukkan ke dalam model.
- initializer_range: Standar deviasi dari truncated_normal_initializer untuk menginisialisasi semua matriks bobot.
- dapat_dilatih: Boolean yang menentukan apakah lapisan yang telah dilatih sebelumnya dapat dilatih.

Parameter alur pelatihan yang dapat Anda sesuaikan adalah:

- model_dir: Lokasi file pos pemeriksaan model. Jika tidak disetel, direktori sementara akan digunakan.
- dropout_rate: Angka putus sekolah.
- learning_rate: Kecepatan pembelajaran awal untuk pengoptimal Adam.
- tpu: alamat TPU yang ingin disambungkan.

Misalnya, Anda dapat mengatur seq_len=256 (defaultnya adalah 128). Hal ini memungkinkan model untuk mengklasifikasikan teks yang lebih panjang.

```
# spec.seq_len = 256
```

Memuat data pelatihan dan pengujian

Muat file CSV data pelatihan dan pengujian untuk mempersiapkan proses pelatihan model. Pastikan parameter is_training untuk test_data diatur ke False.

```
train_data = DataLoader.from_csv(
    filename='train.csv',
    text_column='sentence',
    label_column='label',
    model_spec=spec,
    is_training=True)
```

```
test_data = DataLoader.from_csv(
```

```
filename='test.csv',  
text_column='sentence',  
label_column='label',  
model_spec=spec,  
is_training=False)
```

Train model

Mulai pelatihan model pada kumpulan data kereta. Jangan ragu untuk bermain-main dengan no yang berbeda. epoch hingga Anda menemukan nilai epoch ideal yang memberikan hasil terbaik.

```
model = text_classifier.create(train_data, model_spec=spec, epochs=100)
```

Periksa struktur model Anda - Lapisan jaringan saraf

```
model.summary()
```

Evaluasi modelnya

Evaluasi keakuratan model pada data pengujian dan lihat sendiri apakah model memerlukan beberapa penyesuaian seperti peningkatan kumpulan data atau penyetelan hyperparameter untuk meningkatkan akurasi.

```
loss, acc = model.evaluate(test_data)
```

Ekspor model TF Lite

Model akhir diekspor sebagai model TF Lite yang dapat diunduh dan langsung diterapkan di aplikasi Android Anda.

```
model.export(export_dir='average_word_vec')
```