

Muhammad Fathir Fadillah 121450098 Tugas PBA

1. Dataset yang digunakan adalah yahoo_answer dengan 10 kelas
2. Model yang digunakan adalah GRU, Transformer, BERT, dan Fasttext
3. Analisis waktu komputasi

Hyperparameter yang digunakan adalah sebagai berikut

BATCH_SIZE = 2048 #ukuran batch

SEQUENCE_LENGTH = 2 #panjang sequence

EPOCHS = 2 #jumlah epoch

EMBED_DIM = 2 #dimensi representasi tiap kata

N_LAYERS = 2 #banyak layer

DROPOUT_RATE = 0.4 #probabilitas menonaktifkan neuron

OPTIMIZER = Adam #metode untuk pembaharuan bobot fase backpropagation

LR = 0.029 #kecepatan pembaharuan bobot

LOSS = crossentropy #fungsi loss mengukur goodness of fit model

Num_head = 2 #banyak kepala atensi multihead yang dipakai pada decoder BERT dan transformer

```
GRU

[ ] model.GRU()
history = model.fit(epochs=EPOCHS, optimizer=OPTIMIZER, lr=LR, loss=LOSS)

Training: 100% | 665/665 [05:03<00:00, 2.19batch/s]
Validation: 100% | 30/30 [00:12<00:00, 2.42batch/s]
Epoch 1/2 | Train Loss: 2.3038 | Train Acc: 0.1003 | Val Loss: 2.3032 | Val Acc: 0.1000

Training: 100% | 665/665 [05:10<00:00, 2.14batch/s]
Validation: 100% | 30/30 [00:13<00:00, 2.15batch/s]Epoch 2/2 | Train Loss: 2.3030 | Train Acc: 0.0994 | Val Loss: 2.3031 | Val Acc: 0.1000

Restored model to the best state based on validation loss.

[ ] y_true, y_pred = model.eval()

Validation: 100% | 30/30 [00:24<00:00, 1.21batch/s]
```

```
Fasttext

[ ] model.FASTTEXT()
history = model.fit(epochs=EPOCHS, optimizer=OPTIMIZER, lr=LR, loss=LOSS)

Training: 100% | 665/665 [05:08<00:00, 2.15batch/s]
Validation: 100% | 30/30 [00:18<00:00, 2.75batch/s]
Epoch 1/2 | Train Loss: 2.3040 | Train Acc: 0.1002 | Val Loss: 2.3028 | Val Acc: 0.1000

Training: 100% | 665/665 [04:59<00:00, 2.22batch/s]
Validation: 100% | 30/30 [00:12<00:00, 2.36batch/s]Epoch 2/2 | Train Loss: 2.3030 | Train Acc: 0.1004 | Val Loss: 2.3027 | Val Acc: 0.1000

Restored model to the best state based on validation loss.

[ ] y_true, y_pred = model.eval()

Validation: 100% | 30/30 [00:11<00:00, 2.59batch/s]
```

```
Transformer

[ ] model.TRANSFORMER(num_heads=2)
history = model.fit(epochs=EPOCHS, optimizer=OPTIMIZER, lr=LR, loss=LOSS)

Training: 100% | 665/665 [05:12<00:00, 2.13batch/s]
Validation: 100% | 30/30 [00:14<00:00, 2.08batch/s]
Epoch 1/2 | Train Loss: 2.2983 | Train Acc: 0.1149 | Val Loss: 2.2970 | Val Acc: 0.1184

Training: 100% | 665/665 [05:08<00:00, 2.16batch/s]
Validation: 100% | 30/30 [00:14<00:00, 2.14batch/s]Epoch 2/2 | Train Loss: 2.2997 | Train Acc: 0.1115 | Val Loss: 2.2999 | Val Acc: 0.1121

Restored model to the best state based on validation loss.
```

```

v BERT

[ ] model.BERT(num_heads=2)
  history = model.fit(epochs=EPOCHS, optimizer=OPTIMIZER, lr=LR, loss=LOSS)

Training: 100% | 665/665 [05:05<00:00, 2.18batch/s]
Validation: 100% | 30/30 [00:13<00:00, 2.19batch/s]
Epoch 1/2 | Train Loss: 2.3040 | Train Acc: 0.1002 | Val Loss: 2.3030 | Val Acc: 0.1000

Training: 100% | 665/665 [05:05<00:00, 2.18batch/s]
Validation: 100% | 30/30 [00:13<00:00, 2.14batch/s]Epoch 2/2 | Train Loss: 2.3032 | Train Acc: 0.1005 | Val Loss: 2.3030 | Val Acc: 0.1000

Restored model to the best state based on validation loss.

```

Berikut rerata waktu pelatihan dan pengujian yang dibutuhkan untuk tiap model

Model	Mean Waktu Pelatihan (menit)
GRU	5.1
Fasttext	5.05
Transformer	5.16
BERT	5.083

Secara berturut-turut dari waktu pelatihan tercepat yakni Fasttext, BERT, GRU, dan Transformer. Hal ini dikarenakan fasttext memiliki kemampuan partisi kata sehingga menjadikannya cepat dalam hal komputasinya. Meskipun begitu, fasttext belum mampu menangkap konteks tiap kata. BERT (Bidirectional Encoder Representation Transformer) memodelkan tiap kata dengan pertimbangan konteksnya. BERT komputasinya lebih lama karena adanya proses bidirectional dan atensi pada bagian encodernya. Selanjutnya adalah GRU, dengan kedua gerbangnya yakni update dan reset. Untuk kasus ini model GRU membutuhkan waktu pelatihan yang lebih lama dibandingkan dengan fasttext dan BERT. Hal ini kemungkinan terjadi karena GRU yang merupakan jaringan syaraf sekuensial yang menjadikannya lebih lambat dari pada BERT maupun fasttext. Terakhir, yakni transformer dengan atensi pada encoder dan decodernya. Model ini membutuhkan waktu paling lama di antara yang lainnya. Hal ini disebabkan oleh arsitekturnya yang amat kompleks. Namun, di balik itu transformer memiliki peluang yang lebih besar perihal akurasinya disbanding model-model yang lain. Hal ini disebabkan atensi pada encoder dan decoder yang diterapkan pada transformer bukan hanya mampu menangkap konteks pada input melainkan juga pada target.

4. Analisis kebutuhan data

Data yahoo_asnwer dengan train dan test masing-masing sebesar 32 dan 745 MB. Data yang digunakan relatif cukup besar dan sudah lumayan cukup untuk melatih dan menguji model. Akan tetapi, jumlah epoch yang sedikit hanya menghasilkan akurasi yang relatif kurang bagus. Untuk itu, diperlukan penambahan epoch untuk menjadikan model lebih banyak belajar kembali meskipun dalam satu kali pembelajaran sudah menggunakan banyak data. Di sisi lain, hypeparameter seperti batch size, jumlah neuron, banyak layer, dll juga mempengaruhi performa tiap model. Misalnya, selain meningkatkan epoch kita juga dapat meningkatkan

jumlah layer dan/atau neuron yang digunakan pada model. Terkhusus untuk BERT dan transformer juga dapat ditambahkan banyaknya kepala pada multihead atensinya.

5. Generalisasi

Dengan menambahkan faktor performa tiap model atas data uji dan validasi dengan analisis poin 3 dan 4 dihasilkan analisis secara menyeluruh untuk tiap model.

GRU : Val Loss: 2.3031 | Val Acc: 0.1000

Fasttext : Val Loss: 2.3027 | Val Acc: 0.1000

Transformer : Val Loss: 2.2999 | Val Acc: 0.1121

BERT : Val Loss: 2.3030 | Val Acc: 0.1000

Keempat model menghasilkan akurasi yang hampir sama saat dilakukan pengujian atas data uji. Berdasarkan performa urutan terhadap data validasi secara berturut-turut tertinggi ke rendah adalah transformer, fasttext, BERT, dan yang terakhir GRU. Transformer berdasarkan performanya menempati urutan teratas. Hal ini berbanding terbalik dengan waktu pelatihannya yang menjadikan model ini menempati urutan terakhir dalam hal kecepatan komputasi. Kemudian untuk model fasttext dan BERT masing-masing menempati urutan ke-2 dan 3 perihal performa. Namun, saat pelatihan fasttext dan BERT masing-masing menempati urutan pertama dan ke-2 berdasar kecepatan komputasinya. Dan yang terakhir yakni model GRU yang menempati posisi terakhir pada performanya atas data validasi dan posisi ke-3 berdasar kecepatan komputasinya.

Dari keempat performa dan waktu pelatihan model-model ini menggambarkan bahwa tidak selamanya model yang lebih kompleks butuh sumber daya yang lebih. Namun, tidak berarti sebaliknya jika berdasar performanya atas data uji. Model yang lebih kompleks seperti transformer mampu mengungguli model-model lain yang lebih sederhana darinya. Kompleksitas yang lebih pada suatu model menjadikannya mampu dalam melakukan pembelajaran, atensi terhadap target, dan adaptasi yang lebih terhadap data baru. Meskipun begitu, ada harga yang perlu dibayarkan seperti waktu dan sumber daya komputasi yang berlebih. Baik performa maupun kebutuhan sumber daya komputasi atas pembangunan sebuah model dipengaruhi oleh *hyperparameter* seperti *epoch*, *batch_size*, dimensi embedding, dll.

6. Tautan github

https://github.com/fadillah180403/PBA_FathirFadillah098