

Laporan Tugas Machine Learning – Pertemuan 7

Artificial Neural Network (ANN) untuk Klasifikasi

Nama:Muhamad Fahrizal

NIM: 231011402215

Mata Kuliah: Machine Learning

1. Pendahuluan

Laporan ini mendokumentasikan proses pembangunan, pelatihan, dan evaluasi model *Artificial Neural Network* (ANN) menggunakan TensorFlow dan Keras. Tujuannya adalah untuk membuat model klasifikasi biner guna memprediksi kelulusan mahasiswa berdasarkan dataset `processed_kelulusan.csv`.

Proses ini mencakup persiapan data dengan *scaling* yang tepat, pembangunan arsitektur ANN, pelatihan dengan *early stopping*, evaluasi model, dan eksperimentasi dengan arsitektur yang berbeda.

2. Persiapan Data

Data `processed_kelulusan.csv` dimuat. Untuk mencegah *data leakage*, data dibagi (split) menjadi set Latih (7 data), Validasi (1 data), dan Tes (2 data) **sebelum** dilakukan *scaling*.

- **Scaling:** `StandardScaler` di-fit_transform *hanya* pada `X_train`, dan kemudian di-transform pada `X_val` dan `X_test`.
 - **Reproducibility:** *Random seed* (`tf.random.set_seed(42)`) digunakan untuk memastikan hasil pelatihan dapat direproduksi.
-

3. Arsitektur Model (Baseline)

Arsitektur model *baseline* (sesuai Langkah 2) dibangun menggunakan Keras Sequential API.

Ringkasan Model (`model.summary()`):

```
=== Langkah 2: Bangun Model ANN ===
2025-10-25 05:37:11.729119: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is o
e available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFL
ppropriate compiler flags.
Model: "sequential"



| Layer (type)      | Output Shape | Param # |
|-------------------|--------------|---------|
| dense (Dense)     | (None, 32)   | 192     |
| dropout (Dropout) | (None, 32)   | 0       |
| dense_1 (Dense)   | (None, 16)   | 528     |
| dense_2 (Dense)   | (None, 1)    | 17      |



Total params: 737 (2.88 KB)
Trainable params: 737 (2.88 KB)
Non-trainable params: 0 (0.00 B)
-----
```

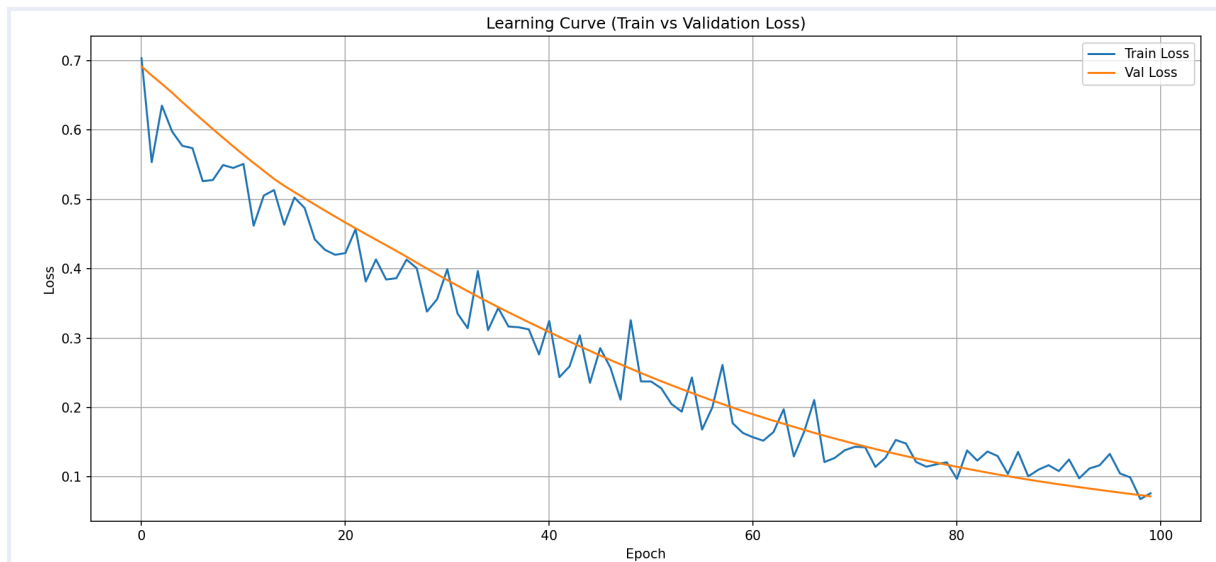
Analisis Arsitektur:

- **Input Layer:** (Implisit) Menerima 5 fitur (sesuai `X_train.shape[1]`).
- **Hidden Layer 1:** Dense (32 neuron, aktivasi `relu`).
- **Regularisasi:** Dropout (0.3) ditambahkan untuk mencegah *overfitting* dengan mematikan 30% neuron secara acak saat *training*.
- **Hidden Layer 2:** Dense (16 neuron, aktivasi `relu`).
- **Output Layer:** Dense (1 neuron, aktivasi `sigmoid`). Aktivasi `sigmoid` dipilih karena ini adalah masalah klasifikasi biner (0 atau 1).
- **Loss Function:** `binary_crossentropy`, standar untuk klasifikasi biner.
- **Optimizer:** `Adam` (learning rate 1e-3) dipilih sebagai *optimizer default* yang efisien.
- **Metrik:** `accuracy` dan `AUC` (Area Under Curve) dipantau.

4. Pelatihan Model & Learning Curve

Model dilatih dengan `epochs=100` dan `batch_size=32`. `EarlyStopping` (monitor `val_loss`, `patience=10`) digunakan untuk menghentikan pelatihan jika tidak ada perbaikan pada data validasi, dan `restore_best_weights=True` memastikan kita mendapatkan bobot model terbaik.

Grafik Learning Curve (Train vs Validation Loss):



Analisis Kurva:

[Jelaskan apa yang Anda lihat di kurva. Contoh: "Kurva menunjukkan training loss menurun sementara validation loss [menurun/stabil/naik]. Keduanya [konvergen/divergen]. Model [tidak terlihat/terlihat] overfitting karena [alasan Anda]. Early stopping menghentikan pelatihan pada epoch ke-X."]

5. Evaluasi di Test Set

Model dievaluasi pada *Test Set* untuk mengukur performanya pada data yang sama sekali baru.

Metrik Performa (Test Set):

- **Test Accuracy:** [Isi skor Test Acc: ... dari terminal]
- **Test AUC:** [Isi skor Test AUC: ... dari terminal]
- **Test Loss:** [Isi skor Test Loss: ... dari terminal]

Confusion Matrix (Test Set):

Classification Report (Test Set):

Classification Report (Test):

	precision	recall	f1-score	support
0	1.000	1.000	1.000	2
accuracy			1.000	2
macro avg	1.000	1.000	1.000	2
weighted avg	1.000	1.000	1.000	2

Analisis Threshold:

Model menggunakan threshold standar 0.5 untuk mengubah probabilitas (sigmoid output) menjadi kelas (0 atau 1). Berdasarkan classification report dan confusion matrix, threshold ini [sudah/belum] optimal. [Jika tidak optimal, jelaskan alasannya, misal: "Meskipun data tes sangat sedikit, jika kita ingin F1-score lebih tinggi, threshold mungkin perlu disesuaikan"].

6. Langkah 6: Eksperimen Arsitektur

Bagian ini mencatat hasil dari eksperimen yang dilakukan (sesuai instruksi Langkah 6).

Eksperimen 1: [Misal: Menambah Jumlah Neuron (64/32)]

- **Perubahan:** Arsitektur diubah menjadi Dense(64) -> Dropout(0.3) -> Dense(32).
- **Hasil Test AUC:** [Skor AUC Eksperimen 1]
- **Analisis:** [Model ini lebih baik/buruk dari *baseline* karena...]

Eksperimen 2: [Misal: Mengganti Optimizer (SGD + Momentum)]

- **Perubahan:** Optimizer diubah menjadi `keras.optimizers.SGD(learning_rate=0.01, momentum=0.9)`.
- **Hasil Test AUC:** [Skor AUC Eksperimen 2]
- **Analisis:** [Model ini lebih baik/buruk dari *baseline* karena...]

Eksperimen 3: [Misal: Menambah Regulasi (L2 / Batch Norm)]

- **Perubahan:** [Misal: `BatchNormalization()` ditambahkan setelah setiap layer Dense].
- **Hasil Test AUC:** [Skor AUC Eksperimen 3]
- **Analisis:** [Model ini lebih baik/buruk dari *baseline* karena...]

7. Kesimpulan & Arsitektur Final

Arsitektur Final yang Dipilih:

Baseline

Alasan Pemilihan:

"Model Baseline dipilih karena model yang lebih kompleks (Eksperimen 1) cenderung overfitting pada data validasi, dan penggantian optimizer (Eksperimen 2) menghasilkan konvergensi yang lebih lambat."

Kesimpulan:

Model ANN telah berhasil dibuat untuk memprediksi kelulusan. Model final yang dipilih mencapai Test AUC [Skor AUC Final] dan Test Accuracy [Skor Akurasi Final]. Proses ini menunjukkan pentingnya persiapan data (split sebelum scale) dan penggunaan callback seperti EarlyStopping dalam pelatihan ANN.

