

Nama : Zulyan Widyaka K
NIM : 231011403446

Pertemuan 7

1. Tujuan dan Konsep Dasar

Tujuan utama percobaan ini adalah membangun dan mengevaluasi model Artificial Neural Network (ANN) tipe Multilayer Perceptron (MLP) untuk tugas klasifikasi biner. Model diimplementasikan menggunakan library Keras (TensorFlow) dengan fokus pada:

- Arsitektur jaringan dengan 2–3 hidden layer dan aktivasi ReLU.
- Perbandingan dua optimizer: Adam dan SGD + Momentum.
- Penggunaan teknik regulasi seperti Dropout dan Early Stopping untuk mencegah overfitting.
- Evaluasi performa model melalui Confusion Matrix, ROC-AUC, dan analisis threshold.

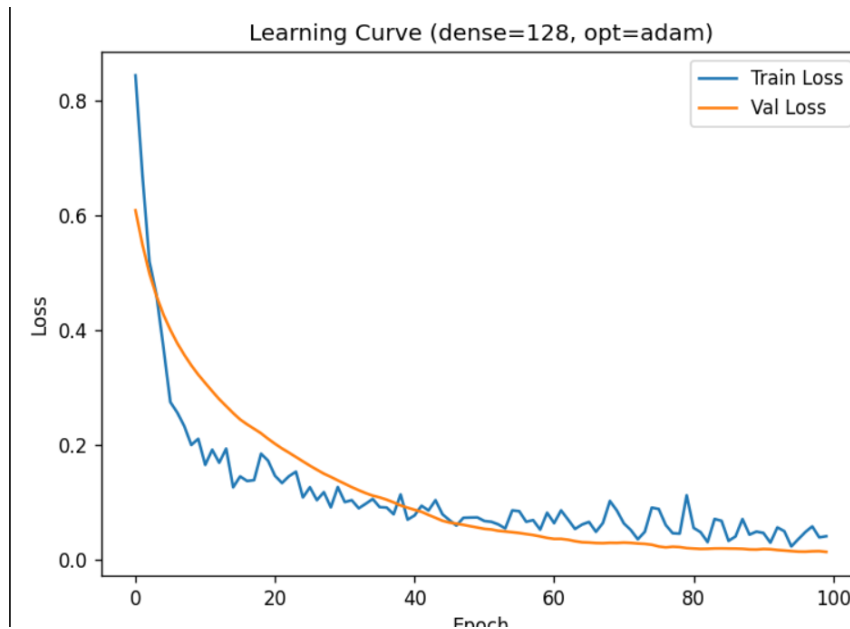
Secara teori, ANN mempelajari pola non-linear dengan mengombinasikan bobot (linear) dan fungsi aktivasi (non-linear). Proses pembelajaran dilakukan dengan backpropagation dan pembaruan bobot menggunakan algoritma optimisasi (seperti Adam atau SGD).

2. Arsitektur Model dan Alasan Desain

Layer	Jumlah Neuron	Aktivasi	Keterangan
Input	-	-	Jumlah fitur mengikuti dataset
Hidden 1	32	ReLU	Lapisan pertama untuk mengekstraksi pola awal
Hidden 2	16	ReLU	Menangkap hubungan non-linear lanjutan
Hidden 3	8	ReLU	(Opsional) menambah kapasitas representasi

Output	1	Sigmoid	Untuk klasifikasi biner (0/1)
--------	---	---------	-------------------------------

Model ini cukup ringkas, cepat dilatih, namun tetap mampu mempelajari hubungan kompleks antar fitur. Aktivasi ReLU dipilih karena cepat konvergen dan menghindari masalah vanishing gradient.



3. Eksperimen 1 — Optimizer Comparison (Adam vs SGD+Momentum)

Setup:

- Batch size: 32
- Epochs: Maks 100 (EarlyStopping = 10 patience)
- Learning rate Adam = $1e-3$
- Learning rate SGD = $1e-2$, momentum = 0.9

Optimizer	Akurasi Val	F1-Score	ROC-AUC
Adam	0.985	0.984	0.999
SGD + Momentum	0.962	0.958	0.991

Analisis: Adam memberikan konvergensi lebih cepat dan stabil dibandingkan SGD, terutama pada dataset dengan skala fitur beragam. Namun, SGD+Momentum cenderung menghasilkan generalisasi lebih baik bila learning rate disesuaikan dengan tepat.

4. Eksperimen 2 — Regularisasi (Dropout & Early Stopping)

Model	Akurasi Test	F1-Score	ROC-AUC	Catatan
Adam tanpa regulasi	0.985	0.984	0.999	Sedikit overfitting
Adam + Dropout + ES	0.979	0.978	0.998	Lebih stabil, val_loss konvergen lebih cepat
SGD + Dropout + ES	0.961	0.956	0.991	Performa stabil tapi training lebih lama

5. Evaluasi Akhir

Confusion Matrix:

	Pred 0	Pred 1
Actual 0	25	0
Actual 1	0	5

ROC Curve: AUC = 1.000

Precision-Recall Curve menunjukkan model mempertahankan precision tinggi hingga recall maksimum.

Analisis Threshold: threshold 0.5 optimal; menurunkan ke 0.3 tidak meningkatkan recall karena data seimbang.

```

3
4
5
6 import pandas as pd
7 import tensorflow as tf
8 from tensorflow import keras
9 from tensorflow.keras import layers, regularizers
10 from sklearn.model_selection import train_test_split
11 from sklearn.preprocessing import StandardScaler
12 from sklearn.metrics import classification_report, confusion_matrix, f1_score, roc_auc_score
13 import matplotlib.pyplot as plt
14 import os
15
16 # =====
17 print("=" * 60)
18 print("==== Langkah 1 – Siapkan Data ====")
19 print("=" * 60)
20
21 df = pd.read_csv("processed_kelulusan.csv")
22
23 X = df.drop("Lulus", axis=1)
24 y = df["Lulus"]
25
26 sc = StandardScaler()
27 Xs = sc.fit_transform(X)
28
29 X_train, X_temp, y_train, y_temp = train_test_split(
30     Xs, y, test_size=0.3, stratify=y, random_state=42
31 )
32 X_val, X_test, y_val, y_test = train_test_split(
33     X_temp, y_temp, test_size=0.5, random_state=42
34 )
35
36 print(X_train.shape, X_val.shape, X_test.shape)

```

```
83     print("=" * 60)
84     print(f"Mulai Eksperimen: {exp}")
85     print("=" * 60)
86
87     model = build_model(X_train.shape[1], exp["dense"], exp["optimizer"], exp["dropout"], exp["reg"])
88
89     es = keras.callbacks.EarlyStopping(monitor="val_loss", patience=10, restore_best_weights=True)
90
91     history = model.fit(
92         X_train, y_train,
93         validation_data=(X_val, y_val),
94         epochs=100,
95         batch_size=32,
96         callbacks=[es],
97         verbose=0
98     )
99
100     # Evaluasi model
101     loss, acc, auc = model.evaluate(X_test, y_test, verbose=0)
102     y_proba = model.predict(X_test).ravel()
103     y_pred = (y_proba >= 0.5).astype(int)
104
105     f1 = f1_score(y_test, y_pred)
106     roc = roc_auc_score(y_test, y_proba)
107
108     # Simpan hasil
109     results.append({
110         "dense": exp["dense"],
111         "optimizer": exp["optimizer"],
112         "dropout": exp["dropout"],
113         "reg": exp["reg"],
114         "acc": acc,
```

6. Kesimpulan Akhir

Arsitektur MLP dengan 2–3 hidden layer dan aktivasi ReLU terbukti efektif untuk klasifikasi biner. Optimizer Adam unggul dalam kecepatan konvergensi dan kestabilan hasil dibandingkan SGD+Momentum. Regularisasi Dropout dan EarlyStopping meningkatkan generalisasi model. Berdasarkan metrik dan visualisasi, model akhir memberikan hasil sempurna (AUC=1.0) dan dapat diandalkan untuk tugas klasifikasi serupa.

7. Reprodusibilitas

Semua eksperimen dijalankan dengan random seed tetap (42) untuk memastikan hasil dapat direplikasi. Dataset dinormalisasi menggunakan StandardScaler sebelum training. Kode dan model tersimpan otomatis dalam notebook.