



P3. KONSEP JARINGAN SYARAF TIRUAN DAN BACKPROPAGATION ALGORITHM (22 MEI 2026)

Pengumpulan Tugas Ebook Deep Learning

Nama: Aris Kurniawan

NIM: 2311510438

Jenis Pengumpulan: Dokumen jawaban + lampiran aplikasi/program

Abstrak Laporan ini memaparkan penerapan konsep jaringan syaraf tiruan (feedforward dan backpropagation) untuk studi kasus prediksi jumlah panen padi. Disertakan penjabaran langkah perhitungan numerik (tabel forward dan backpropagation), rumus inti, serta informasi lampiran program implementasi.

1. Tujuan Menjelaskan tahapan feedforward dan backpropagation secara matematis dan numerik pada jaringan sederhana (arsitektur 3-4-1) yang memprediksi hasil panen berdasar tiga fitur input.
2. Arsitektur model dan notasi
 - Input $x \in \mathbb{R}^3 = \underline{x_1, x_2, x_3}$ – telah dinormalisasi.
 - Layer tersembunyi: 4 neuron, aktivasi $\sigma(z) = 1/(1+e^{-z})$ (sigmoid).
 - Output: 1 neuron (linear).
 - Bobot/bias:
 - $W1 \in \mathbb{R}^{4 \times 3}$, $b1 \in \mathbb{R}^4$
 - $W2 \in \mathbb{R}^{1 \times 4}$, $b2 \in \mathbb{R}$
 - Fungsi loss: $MSE L = \frac{1}{2} (y_{true} - y_{pred})^2$
 - Learning rate: $\eta = 0.1$

1. Rumus inti

- Forward:
 - $z^{\{1\}} = W1 x + b1$
 - $a^{\{1\}} = \sigma(z^{\{1\}})$
 - $z^{\{2\}} = W2 a^{\{1\}} + b2$
 - $y_{pred} = z^{\{2\}}$ (linear)
- Backpropagation (gradien):

- $\delta^{(2)} = y_{\text{pred}} - y_{\text{true}}$
- $\partial L / \partial W_2 = \delta^{(2)} (a^{(1)})^T$
- $\partial L / \partial b_2 = \delta^{(2)}$
- $\delta^{(1)} = (W_2^T \cdot \delta^{(2)}) * \sigma'(z^{(1)})$, where $\sigma'(z) = \sigma(z)(1 - \sigma(z))$
- $\partial L / \partial W_1 = \delta^{(1)} x^T$
- $\partial L / \partial b_1 = \delta^{(1)}$
- Update bobot:
 - $W \leftarrow W - \eta \partial L / \partial W$
 - $b \leftarrow b - \eta \partial L / \partial b$

1. Contoh numerik (satu sampel) Set awal dan data contoh:

- $x = \underline{0.5, 0.2, 0.1}$
- $y_{\text{true}} = 0.6$
- Inisialisasi:
 - $W_1 = [[0.1, -0.2, 0.3], [0.4, 0.1, -0.1], [-0.3, 0.2, 0.2], [0.2, -0.1, 0.05]]$
 - $b_1 = [0, 0, 0, 0]$
 - $W_2 = [0.2, -0.1, 0.05, 0.1]$
 - $b_2 = 0$

Langkah numerik (iterasi 1 — forward & backprop; disederhanakan ke 4 desimal):

- Forward (iterasi 1):
 - $z^{(1)} \approx [0.0400, 0.2100, -0.0900, 0.0850]$
 - $a^{(1)} \approx [0.5100, 0.5520, 0.4775, 0.5212]$
 - $y_{\text{pred}} \approx 0.1228$
 - $\text{Loss} \approx 0.1139$
- Backprop (iterasi 1): dihitung δ dan gradien → pembaruan bobot menghasilkan $W_2 \approx [0.2243, -0.0736, 0.0728, 0.1249]$, $b_2 \approx 0.0477$; W_1 dan b_1 juga diperbarui (nilai numerik lengkap di lampiran kode).

TABEL A — Forward propagation (ringkasan iterasi)

Iterasi	y_{pred}	Loss (MSE/2)
1	0.1228	0.1139
2	0.2215	0.0717
3	0.3000	0.0450

TABEL B — Ringkasan pembaruan bobot W_2 dan bias b_2 per iterasi

Iterasi	W2 (4 nilai)	b2
0 (awal)	[0.2000, -0.1000, 0.0500, 0.1000]	0.0000
1	[0.2243, -0.0736, 0.0728, 0.1249]	0.0477
2	[0.2437, -0.0528, 0.0909, 0.1446]	0.0856
3	(lanjutan; konvergen menuju prediksi lebih tinggi)	0.0856→0.???

Catatan: Tabel B menunjukkan arah dan magnitudo pembaruan. Nilai lengkap untuk W1, b1, W2, b2 pada setiap iterasi tersedia dalam implementasi program (lampiran) untuk verifikasi numerik langkah demi langkah.

1. Interpretasi hasil

- Dari contoh numerik, loss menurun dari $\approx 0.1139 \rightarrow \approx 0.0717 \rightarrow \approx 0.0450$ dalam tiga iterasi, menandakan model belajar menurunkan kesalahan.
- Pembaruan bobot menunjukkan kontribusi fitur melalui gradien; bobot W2 meningkat pada komponen yang membantu meningkatkan prediksi mendekati y_{true} .
- Untuk dataset nyata, diperlukan pembagian data (train/validation), normalisasi fitur, pengaturan hyperparameter (ukuran hidden, η , jumlah epoch), dan regularisasi bila perlu.

1. Evaluasi dan metrik (yang dianjurkan pada laporan akhir)

- RMSE, MAE, MAPE, dan R^2 untuk menilai performa regresi.
- Visualisasi kurva loss terhadap epoch (konvergensi).
- Analisis residual untuk mendeteksi bias model.

1. Langkah implementasi praktis (ringkas)

- Kumpulkan data historis panen + fitur cuaca & manajemen (minimal 50–200 sampel untuk eksperimen).
- Normalisasi fitur.
- Implementasi model seperti yang dijelaskan; jalankan training dan monitoring loss.
- Validasi menggunakan data terpisah; tuning hyperparameter bila perlu.

Lampiran program File implementasi lengkap (Python) disertakan sebagai lampiran program: `neural_network_implementation.py` — berisi kelas jaringan, fungsi forward/backward, loop training, normalisasi data contoh, dan perhitungan metrik evaluasi. Lampiran ini dapat dijalankan untuk mereproduksi angka tabel dan menghasilkan grafik konvergensi.

Lampiran laporan Format laporan terlampir berisi struktur pengisian (pendahuluan, tinjauan pustaka singkat, metodologi, hasil perhitungan numerik lengkap, diskusi, kesimpulan). Gunakan lampiran program untuk menyalin tabel per-iterasi yang terperinci.

Kesimpulan singkat Penjabaran matematis dan numerik ini menunjukkan secara jelas bagaimana feedforward menghasilkan prediksi awal, dan bagaimana backpropagation menggunakan aturan

rantai untuk menghitung gradien yang dipakai memperbarui bobot sehingga loss menurun. Lampiran program memberikan verifikasi numerik lengkap dan dapat dijalankan untuk menghasilkan tabel lengkap serta grafik konvergensi yang dapat dimasukkan ke dalam laporan akhir.

Lampiran yang diserahkan:

- Program implementasi: `neural_network_implementation.py` (kode dan cara menjalankan)
- File panduan format laporan dan contoh tabel untuk disalin ke berkas PDF laporan.

(catatan: seluruh perhitungan numerik ringkasan tercantum dalam isi; perhitungan lengkap per-parameter dan script untuk reproduksi tersedia di lampiran program.)