

# Artificial Neural Network

## 1. Cara kerja algoritma

ANN bekerja dengan memproses data melalui jaringan neuron yang terorganisir dalam lapisan. Setiap neuron menerima input, menerapkan bobot dan bias, dan menggunakan fungsi aktivasi untuk menghasilkan output. Jaringan belajar dengan menyesuaikan bobot berdasarkan kesalahan prediksi.

### 1. Membangun arsitektur jaringan

ANN terdiri dari tiga jenis lapisan:

- Input layer: Lapisan pertama yang menerima input data.
- Hidden layer: Satu atau lebih lapisan di antara input dan output yang memproses data. Jumlah neuron dan hidden layer dapat disesuaikan berdasarkan kompleksitas masalah.
- Output layer: Lapisan terakhir yang menghasilkan prediksi. Jumlah neuron di lapisan ini biasanya sesuai dengan jumlah kelas yang diprediksi.

### 2. Inisialisasi bobot dan bias

Setiap koneksi antara neuron memiliki bobot yang diinisialisasi secara acak. Setiap neuron juga memiliki bias yang diinisialisasi.

### 3. Forward propagation

Data input diproses melalui jaringan. Setiap neuron menghitung outputnya dengan mengalikan input dengan bobot, menambahkan bias, dan menerapkan fungsi aktivasi (seperti ReLU, sigmoid, dll).

### 4. Menghitung kesalahan

Output dari jaringan dibandingkan dengan label sebenarnya untuk menghitung kesalahan menggunakan fungsi loss (seperti cross-entropy untuk klasifikasi).

### 5. Backward propagation

Kesalahan digunakan untuk memperbarui bobot dan bias melalui proses yang disebut backpropagation. Algoritma optimasi seperti stochastic gradient descent (SGD) atau Adam digunakan untuk menyesuaikan bobot guna meminimalkan kesalahan.

### 6. Pelatihan jaringan

Proses forward dan backward propagation diulang untuk beberapa iterasi (epochs) hingga jaringan mencapai tingkat kesalahan yang dapat diterima.

### 7. Klasifikasi data baru

Setelah pelatihan, ANN dapat digunakan untuk mengklasifikasikan data baru dengan memprosesnya melalui jaringan dan menghasilkan prediksi di output layer.

## 8. Evaluasi

Setelah model dilatih, performanya dievaluasi menggunakan metrik seperti accuracy, precision, recall, dan F1-score pada data uji. Jika hasilnya tidak memuaskan, dapat dilakukan parameter tuning.

## 2. Perbandingan model buatan sendiri dan library

Berdasarkan hasil evaluasi, model ANN library tensorflow memiliki akurasi 0.83, sedangkan model ANN implementasi sendiri memiliki akurasi 0.86. Ini menunjukkan bahwa implementasi sendiri memiliki performa yang sedikit lebih baik dalam hal akurasi. Hal ini bisa disebabkan karena di implementasi sendiri, proses backpropagationnya lebih tepat dan konsisten dengan teori serta bisa juga disebabkan oleh penggunaan batch size dan epochs yang digunakan.

## 3. Improvement yang bisa dilakukan

- Weight initialization  
Selain He initialization, bisa dipertimbangan untuk menggunakan Xavier/Glorot initialization untuk lapisan dengan aktivasi sigmoid atau tanh
- Hyperparameter tuning  
Gunakan grid search atau random search untuk menemukan kombinasi hyperparameter yang optimal, seperti ukuran batch, learning rate, dan arsitektur jaringan.
- Early stopping  
Implementasikan early stopping untuk menghentikan pelatihan ketika performa pada data validasi mulai menurun.
- Activations functions  
Eksperimen dengan fungsi aktivasi lain seperti Leaky ReLU, ELU, atau tanh untuk melihat apakah mereka memberikan hasil yang lebih baik.
- Learning rate schedule  
Implementasikan scheduling learning rate yang lebih canggih, seperti exponential decay atau adaptive learning rates untuk membantu konvergensi yang lebih baik.
- Batch normalization  
Tambahkan batch normalization setelah setiap lapisan untuk membantu stabilitas pelatihan dan mempercepat konvergensi.