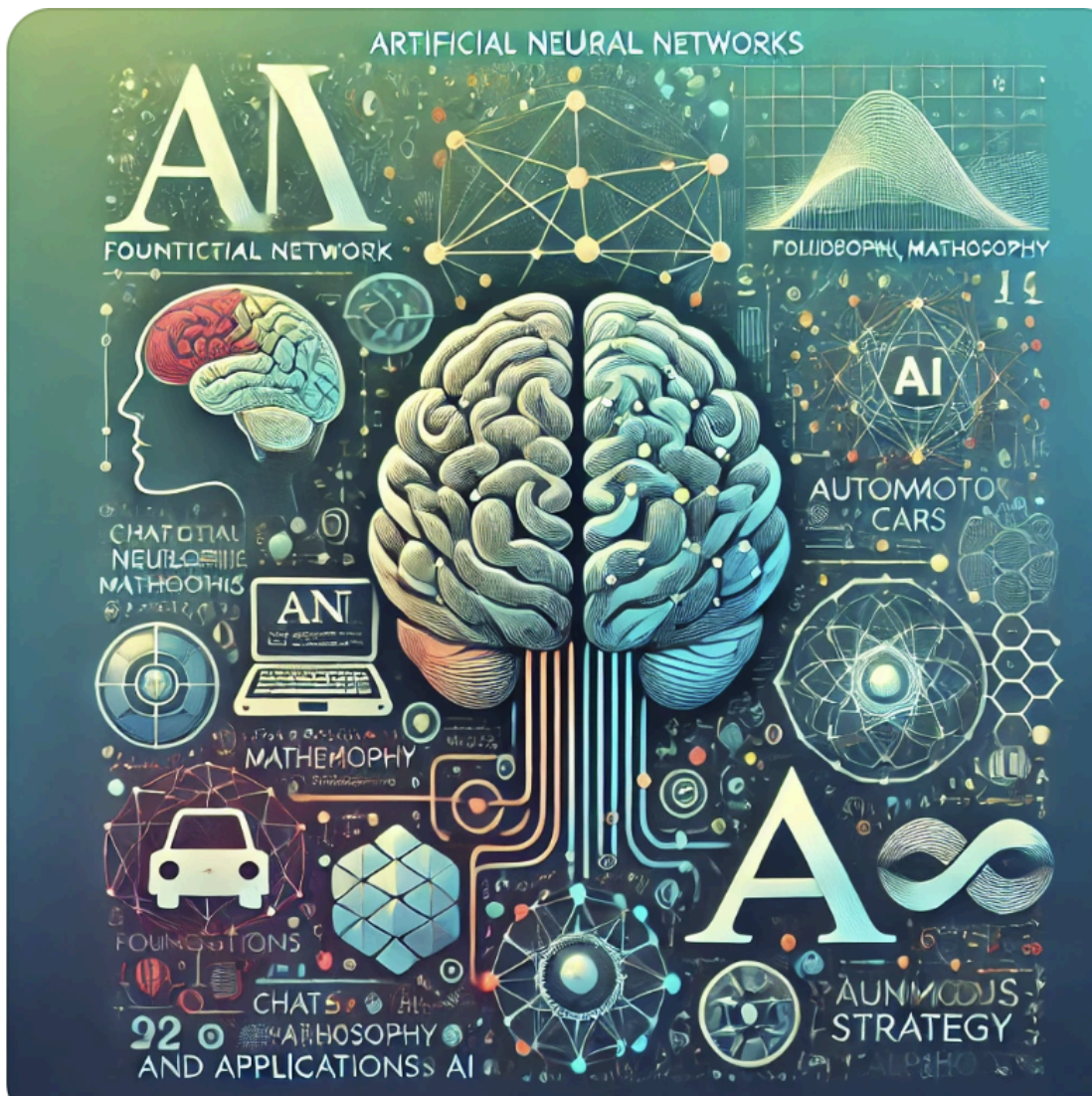


Artificial Neural Network

Dasar, Sejarah, dan Aplikasinya



Muchamad Kurniawan

#informatika_ITATS

Pendahuluan

Artificial Neural Network (ANN) adalah salah satu komponen inti dalam perkembangan kecerdasan buatan (AI) modern. Terinspirasi dari cara kerja otak manusia, ANN memungkinkan mesin untuk belajar dari data, mengenali pola, dan mengambil keputusan. Dari algoritma sederhana seperti **perceptron** hingga model kompleks dalam **deep learning**, ANN memainkan peran penting dalam berbagai aplikasi, mulai dari sistem rekomendasi hingga kendaraan otonom.

Konsep pembelajaran dalam ANN berakar pada filosofi lama yang menganggap bahwa pengetahuan berasal dari pengalaman. **Aristoteles** menekankan pentingnya pengalaman dalam memperoleh pengetahuan, sementara **Jean Piaget** menggambarkan pembelajaran sebagai proses aktif yang melibatkan interaksi dengan lingkungan. Filosofi ini menjadi dasar dalam pembelajaran mesin, di mana komputer juga belajar dan beradaptasi melalui data.

Sejarah Artificial Neural Networks (ANN): Dari Awal Hingga Deep Learning

Artificial Neural Networks (ANN) adalah salah satu cabang utama dalam kecerdasan buatan (AI) yang mengubah cara komputer belajar dan membuat keputusan. Terinspirasi oleh cara kerja otak manusia, ANN telah berkembang pesat sejak diperkenalkan pada pertengahan abad ke-20. Transformasi dari teori dasar menjadi model **deep learning** yang mendukung aplikasi modern, seperti pengenalan gambar dan pemrosesan bahasa alami, adalah hasil dari penelitian bertahun-tahun. Artikel ini merangkum sejarah ANN, mulai dari ide awal hingga munculnya **Deep Learning**.

1. Awal Mula: Inspirasi dari Otak Manusia

- **1943: Model McCulloch-Pitts**
 - ANN pertama kali diperkenalkan oleh **Warren McCulloch** dan **Walter Pitts** pada tahun 1943. Mereka mengembangkan model matematis sederhana untuk neuron biologis, yang dikenal sebagai **McCulloch-Pitts Neuron**.
 - Model ini menggunakan logika Boolean untuk menunjukkan bagaimana neuron dapat memproses informasi melalui koneksi yang saling terhubung. Ini menjadi dasar bagi konsep jaringan saraf buatan.
- **1958: Perceptron oleh Frank Rosenblatt**

- **Frank Rosenblatt** menciptakan **Perceptron**, algoritma ANN pertama yang dapat mempelajari pola dari data input. Perceptron dirancang untuk menyelesaikan masalah klasifikasi linier.
- Perceptron dianggap sebagai lompatan besar karena memperkenalkan mekanisme pembelajaran berdasarkan bobot yang diperbarui melalui **gradient descent**.

Kelebihan Awal:

- Perceptron mampu menangani data sederhana dan mempelajari pola linier.
- Konsep ini menjadi dasar untuk pengembangan model ANN yang lebih kompleks.

Tantangan:

- Perceptron tidak dapat menangani masalah yang tidak linier, seperti masalah **XOR**, yang menghambat kemajuannya pada waktu itu.
-

2. Stagnasi: "AI Winter" dan Kritik

- **1969: Kritik dari Minsky dan Papert**
 - Buku "**Perceptrons**" karya **Marvin Minsky** dan **Seymour Papert** mengkritik keterbatasan Perceptron. Mereka menunjukkan bahwa model ini tidak dapat memecahkan masalah non-linier seperti **XOR**, yang memerlukan arsitektur multi-layer.
 - Kritik ini mengarah pada penurunan minat dalam penelitian jaringan saraf buatan, yang dikenal sebagai **AI Winter**.
 - **1970-an: Fokus Beralih**
 - Penelitian AI beralih ke pendekatan berbasis logika dan aturan, seperti sistem pakar (expert systems), yang dianggap lebih dapat diandalkan pada waktu itu.
-

3. Kebangkitan ANN: Penemuan Backpropagation

- **1986: Kembalinya ANN dengan Backpropagation**
 - **Geoffrey Hinton**, bersama **David Rumelhart** dan **Ronald Williams**, memperkenalkan kembali ANN melalui algoritma **Backpropagation**.

- Backpropagation memungkinkan pelatihan jaringan saraf multi-layer dengan cara menghitung error di output dan mendistribusikan koreksi ke layer sebelumnya.
- Penemuan ini membuka jalan untuk menyelesaikan masalah non-linier yang sebelumnya tidak dapat ditangani oleh Perceptron.

Keunggulan Backpropagation:

- Meningkatkan akurasi ANN pada dataset yang lebih kompleks.
 - Memungkinkan jaringan saraf untuk belajar secara iteratif melalui data besar.
-

4. Era Deep Learning: Revolusi Jaringan Saraf Dalam

- **1990-an: Restricted Boltzmann Machines dan Neural Networks**
 - **Geoffrey Hinton** mengembangkan **Restricted Boltzmann Machines (RBM)**, yang membantu pembelajaran fitur secara hierarkis. Teknik ini menjadi dasar dari **Deep Belief Networks (DBN)**, cikal bakal jaringan saraf dalam.
- **2010-an: Ketersediaan Data Besar dan GPU**
 - Kemajuan teknologi, seperti ketersediaan **Big Data** dan **komputasi GPU**, mempercepat pelatihan model ANN yang lebih kompleks. Hal ini memungkinkan penerapan jaringan saraf dalam skala besar.
- **2012: Revolusi AlexNet**
 - **Alex Krizhevsky**, bersama Hinton, memperkenalkan **AlexNet**, model deep learning berbasis **Convolutional Neural Network (CNN)**, yang memenangkan kompetisi ImageNet. CNN dirancang untuk mengenali pola dalam data visual.
 - AlexNet menggunakan GPU untuk mempercepat pelatihan, membuktikan bahwa deep learning dapat mengungguli metode tradisional.
- **2014: Generative Adversarial Networks (GANs)**
 - **Ian Goodfellow** memperkenalkan **GANs**, yang memungkinkan jaringan saraf untuk menghasilkan data baru yang realistis, seperti gambar atau suara.
- **2017: Transformer dan Era Pemrosesan Bahasa**
 - **Google** memperkenalkan arsitektur **Transformer**, yang menjadi dasar untuk model AI bahasa besar seperti **GPT (Generative Pre-trained**

Transformer) dan **BERT (Bidirectional Encoder Representations from Transformers).**

- Transformer memungkinkan pemrosesan data sekuensial dengan cara yang lebih efisien, membawa kemajuan besar dalam pemrosesan bahasa alami (NLP).
-

5. Faktor-Faktor Pendukung Deep Learning

Beberapa faktor utama yang mendorong era Deep Learning adalah:

1. **Big Data:** Ketersediaan dataset besar dari berbagai domain, seperti citra, teks, dan suara.
2. **Komputasi GPU:** Kemampuan GPU untuk memproses data paralel mempercepat pelatihan model.
3. **Inovasi Algoritma:** Kemajuan seperti backpropagation, dropout (untuk mencegah overfitting), dan optimasi seperti Adam Optimizer.
4. **Ekosistem Open-Source:** Framework seperti TensorFlow, PyTorch, dan Keras membuat pengembangan model deep learning lebih mudah diakses.

Man vs. Machine: Perbedaan AlphaGo dan Deep Blue

Deep Blue adalah superkomputer IBM yang didesain khusus untuk bermain catur. Pada tahun 1997, Deep Blue berhasil mengalahkan juara dunia catur Garry Kasparov. Algoritma yang digunakan oleh Deep Blue berbeda dengan teknik pembelajaran yang digunakan dalam AI modern. Komputer ini menggunakan **algoritma brute-force search**, yang memungkinkan Deep Blue untuk menganalisis jutaan kemungkinan langkah setiap detik.

- **Algoritma Utama:** Brute-force search dan heuristic evaluation.
- **Keunggulan:** Deep Blue menggunakan kekuatan komputasi untuk mengevaluasi semua kemungkinan langkah hingga kedalaman tertentu, memungkinkan komputer ini untuk memprediksi gerakan terbaik berdasarkan setiap situasi.
- **Keterbatasan:** Deep Blue tidak memiliki kemampuan adaptif atau pembelajaran dari pengalaman. Semua pengetahuan catur yang dimiliki mesin ini diperoleh

dari fungsi evaluasi dan aturan yang dirancang manusia.

Deep Blue tidak "belajar" dari permainan, tetapi memanfaatkan kecepatan dan kapasitas komputasi untuk mengevaluasi gerakan yang optimal. Meskipun berhasil, pendekatan ini masih memiliki keterbatasan karena tidak bisa beradaptasi atau belajar dari pengalaman.

AlphaGo (Google DeepMind) - Go, 2017

AlphaGo membawa pendekatan baru yang lebih modern dalam AI. Menghadapi permainan **Go**, yang lebih kompleks daripada catur karena jumlah kemungkinan langkah yang hampir tak terbatas, AlphaGo menggunakan **deep neural networks** dan **reinforcement learning** untuk mempelajari strategi permainan.

- **Algoritma Utama:** Kombinasi dari deep neural networks, reinforcement learning, dan Monte Carlo Tree Search (MCTS).
- **Keunggulan:** AlphaGo mengembangkan kemampuan untuk memahami pola dalam permainan Go, mengurangi ruang pencarian dengan lebih efisien, dan bahkan menciptakan strategi baru yang tidak pernah terpikirkan oleh pemain manusia.
- **Adaptasi:** AlphaGo belajar dari simulasi permainan berulang, memainkan ribuan permainan dengan dirinya sendiri untuk menyempurnakan strateginya.

Kemenangan AlphaGo menunjukkan bahwa kecerdasan buatan modern mampu melakukan lebih dari sekadar menghitung kemungkinan; ia dapat belajar dan beradaptasi dari pengalaman, yang lebih mendekati cara manusia bermain dan belajar.

Perbandingan antara **Deep Blue** dan **AlphaGo** menunjukkan perbedaan besar dalam pendekatan AI. Deep Blue mengandalkan komputasi kasar, sedangkan AlphaGo memanfaatkan pembelajaran mendalam untuk menghasilkan kemampuan yang lebih fleksibel dan adaptif. Kemenangan AlphaGo menjadi tonggak penting dalam AI, menunjukkan potensi jaringan saraf dalam mengatasi tantangan kompleks yang sebelumnya dianggap terlalu sulit bagi mesin.

Dasar-Dasar Artificial Neural Network

ANN terdiri dari beberapa komponen utama:

1. **Neuron:** Unit pemrosesan dasar yang menghitung output berdasarkan input, bobot, dan bias.
2. **Bobot (Weights):** Menentukan pengaruh setiap input pada output.
3. **Bias:** Menambahkan fleksibilitas ke model.
4. **Fungsi Aktivasi:** Menghasilkan keluaran non-linear yang menentukan apakah neuron aktif.

ANN bekerja melalui dua proses utama:

1. **Forward Propagation:** Data mengalir dari input ke output.
2. **Backward Propagation:** Bobot diperbarui berdasarkan error untuk meminimalkan kesalahan prediksi.

Perceptron: Dasar dari Artificial Neural Network

Perceptron adalah algoritma dasar dalam ANN yang diperkenalkan oleh **Frank Rosenblatt** pada tahun 1958. Perceptron digunakan untuk klasifikasi linier dan bekerja dengan cara menghitung output berdasarkan input, bobot, dan bias.

Formula Perceptron:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

Perhitungan Backpropagation pada Perceptron

1. Hitung Forward Pass:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

Dengan fungsi aktivasi sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}}$$

2. Hitung Error:

$$E = \frac{1}{2}(t - y)^2$$

3. Hitung Gradien untuk Perbaruan Bobot:

$$\frac{\partial E}{\partial w_i} = -(t - y) \cdot y \cdot (1 - y) \cdot x_i$$

4. Perbarui Bobot dan Bias:

$$w_i = w_i - \alpha \cdot \frac{\partial E}{\partial w_i}$$

$$b = b - \alpha \cdot \frac{\partial E}{\partial b}$$

Contoh Perhitungan Manual Forward Propagation dan Backward Propagation

1. Dataset yang Digunakan

Kita akan menggunakan dataset logika sederhana **AND** untuk mempermudah perhitungan. Target keluaran (output) untuk fungsi AND adalah 1 hanya jika kedua input adalah 1; sebaliknya, output adalah 0.

x_1	x_2	Target (t)
1	1	1

2. Inisialisasi Parameter

- Bobot (Weights): $w_1 = 0.5, w_2 = -0.5$
- Bias (Bias): $b = 0$
- Learning Rate (α): 0.1

3. Forward Propagation

Forward propagation adalah langkah awal di mana input dikalikan dengan bobot masing-masing, kemudian bias ditambahkan, dan hasilnya diteruskan melalui fungsi aktivasi untuk menghasilkan output prediksi.

1. Hitung Input Teragregasi (z):

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b$$

Substitusi nilai:

$$z = (0.5 \cdot 1) + (-0.5 \cdot 1) + 0 = 0$$

2. Fungsi Aktivasi: Kita menggunakan fungsi aktivasi sigmoid untuk menghitung output y :

$$y = f(z) = \frac{1}{1 + e^{-z}}$$

Karena $z = 0$:

$$y = \frac{1}{1 + e^0} = \frac{1}{2} = 0.5$$

Output prediksi y dari forward pass adalah **0.5**.

4. Hitung Error (Loss)

Untuk menghitung error antara target sebenarnya dan prediksi, kita menggunakan fungsi **Mean Squared Error (MSE)** sebagai berikut:

$$E = \frac{1}{2}(t - y)^2$$

Substitusi nilai $t = 1$ dan $y = 0.5$:

$$E = \frac{1}{2}(1 - 0.5)^2 = \frac{1}{2}(0.5)^2 = 0.125$$

Error ini akan digunakan untuk memperbarui bobot dan bias selama backpropagation.

5. Backward Propagation

Backward propagation adalah langkah di mana kita menghitung gradien dari error terhadap setiap parameter (bobot dan bias) untuk memperbarui nilai mereka. Kita akan menggunakan **gradient descent** untuk mengurangi error.

Langkah-Langkah Backpropagation

1. Gradien Error terhadap Output (y):

$$\frac{\partial E}{\partial y} = -(t - y) = -(1 - 0.5) = -0.5$$

2. Gradien Output terhadap Input Teragregasi (z): Karena kita menggunakan fungsi aktivasi sigmoid, turunan dari sigmoid adalah:

$$\frac{\partial y}{\partial z} = y \cdot (1 - y)$$

Substitusi $y = 0.5$:

$$\frac{\partial y}{\partial z} = 0.5 \cdot (1 - 0.5) = 0.5 \cdot 0.5 = 0.25$$

3. Gradien dari Error terhadap z :

$$\frac{\partial E}{\partial z} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial z}$$

Substitusi:

$$\frac{\partial E}{\partial z} = -0.5 \cdot 0.25 = -0.125$$

4. **Gradien Error terhadap Bobot (w_1 dan w_2):** Untuk memperbarui setiap bobot, kita hitung turunan dari error terhadap setiap bobot.

Untuk w_1 :

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial w_1} = -0.125 \cdot x_1 = -0.125 \cdot 1 = -0.125$$

Untuk w_2 :

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial w_2} = -0.125 \cdot x_2 = -0.125 \cdot 1 = -0.125$$

5. **Gradien Error terhadap Bias (b):**

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial z} = -0.125$$

Perbarui Bobot dan Bias

Setelah menghitung gradien, kita dapat memperbarui bobot dan bias menggunakan formula:

$$w_i = w_i - \alpha \cdot \frac{\partial E}{\partial w_i}$$
$$b = b - \alpha \cdot \frac{\partial E}{\partial b}$$

Substitusi dengan $\alpha = 0.1$:

- **Perbarui w_1 :**

$$w_1 = 0.5 - 0.1 \cdot (-0.125) = 0.5 + 0.0125 = 0.5125$$

- **Perbarui w_2 :**

$$w_2 = -0.5 - 0.1 \cdot (-0.125) = -0.5 + 0.0125 = -0.4875$$

- **Perbarui Bias (b):**

$$b = 0 - 0.1 \cdot (-0.125) = 0 + 0.0125 = 0.0125$$

Hasil Akhir Setelah Backpropagation (1 Epoch)

Setelah satu iterasi forward dan backward propagation, nilai parameter diperbarui sebagai berikut:

- $w_1 = 0.5125$
- $w_2 = -0.4875$
- $b = 0.0125$

Dengan perhitungan ini, bobot dan bias telah diperbarui untuk menghasilkan prediksi yang lebih baik pada epoch berikutnya. Proses ini akan berulang hingga error mencapai nilai minimum yang diinginkan.

Implementasi ANN dalam Produk Populer

1. **ChatGPT (OpenAI):** Menggunakan ANN berbasis Transformer untuk pemrosesan bahasa alami.
2. **Tesla Autopilot:** Kendaraan otonom yang menggunakan ANN untuk pemrosesan data sensor.
3. **Netflix:** Rekomendasi konten berbasis ANN.
4. **Google Assistant:** Menggunakan ANN untuk pengenalan suara dan respons berbasis konteks.
5. **AlphaGo:** Strategi permainan berbasis deep neural networks.

Kesimpulan

Forward propagation dan backward propagation adalah langkah-langkah utama dalam ANN untuk mempelajari pola dari data dan mengurangi error dalam prediksi. Dengan melakukan perhitungan manual seperti ini, kita dapat memahami bagaimana setiap parameter diperbarui dalam jaringan saraf dan bagaimana ANN "belajar" dari data untuk membuat keputusan yang lebih akurat.