

Nama : Ni Kadek Angelita Meilani

Nim : 662021001

Username Github: gekangel

### **ANN (Artificial Neural Network)**

Jaringan Saraf Tiruan (JST) dan Jaringan Saraf Tiruan (ANN) adalah kumpulan algoritma yang mirip dengan system saraf manusia, di mana neuron saling terhubung satu sama lain untuk memproses data. ANN biasa digunakan dalam hal prediksi pasar, diagnosis medis, pengenalan suara, dan pengenalan gambar. ANN belajar dari data melalui proses yang disebut training, di mana input yang masuk ke neuron ( $x_0, x_1, x_2$ ). Setiap input dikalikan dengan variable yang disebut sebagai "weight" ( $w_0, w_1, w_2$ ), dan kemudian ketiga input tersebut dijumlahkan. Selama proses pembelajaran, nilai dari setiap koneksi neuron akan berubah sampai model ANN menghasilkan output yang diinginkan. Setelah itu bisa ditambahkan bias  $b$  kedalam hasil penjumlahan diatas. Nilai bias ini tidak datang dari input layer. Bias seperti intercept dalam persamaan linear, ditambahkan untuk mengatur agar hasil perhitungan lebih akurat. Setelah melakukan semua perhitungan di atas, neuron akan dimasukkan ke dalam fungsi aktivasi. Fungsi ini menentukan apakah neuron harus aktif atau tidak.

ANN terorganisir dalam tiga lapisan, diantaranya :

- **input layer** : dalam lapisan ini berperan untuk membawa data masuk kedalam system untuk kemudian di proses pada layer selanjutnya.
- **hidden layer**: lapisan antara input layer dan output layer, dimana artificial neuron yang memiliki sekumpulan input pembobot 'weight' dan prosedur untuk menghasilkan output neuron melalui *activation function*. Dalam lapisan ini input akan diproses dan menghasilkan pola atau fitur.
- **output layer**: lapisan terakhir dari neuron yang menghasilkan output system.

Fungsi aktivasi dalam Jaringan Saraf Tiruan (ANN) berperan penting dalam menentukan output dari neuron. Berikut adalah beberapa contoh fungsi aktivasi yang sering digunakan dalam ANN:

- **Sigmoid atau Logistik:**

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

Fungsi sigmoid menghasilkan output antara 0 dan 1, membuatnya berguna untuk model probabilitas dan klasifikasi biner.

- **Tangens Hiperbolik (Tanh):**

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

Tanh menghasilkan output antara -1 dan 1, sering digunakan karena rentang outputnya yang simetris terhadap titik asal.

- **Rectified Linear Unit (ReLU):**

$$f(x) = \max(0, x)$$

ReLU memberikan output x jika x positif dan 0 jika x negatif. Ini populer karena efisiensi komputasinya dan membantu mengurangi masalah gradien yang menghilang.

- **Leaky ReLU:**

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01x, & \text{if } x \leq 0 \end{cases}$$

Leaky ReLU memodifikasi ReLU dengan memungkinkan gradien kecil untuk nilai negatif, yang membantu mengatasi masalah neuron “mati” pada ReLU tradisional.

- **Softmax:**

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Softmax biasanya digunakan pada lapisan output dari jaringan saraf untuk klasifikasi multikelas, di mana K adalah jumlah kelas.

Jenis-jenis ANN yang sering digunakan :

- **Feedforward Neural Networks (FNN):** jenis ANN paling sederhana karena koneksi antar neuron tidak membentuk siklus. Informasi bergerak ke depan dari lapisan input, melalui lapisan tersembunyi, dan akhirnya ke lapisan output.
- **Single Layer Perceptron:** bentuk ANN paling dasar dengan hanya satu lapisan output. Setiap input terhubung langsung ke neuron output melalui serangkaian bobot (weights).
- **Multilayer Perceptrons (MLP):** ANN dengan satu atau lebih lapisan tersembunyi antara lapisan input dan output. MLP dapat memodelkan fungsi yang lebih kompleks dibandingkan dengan single layer perceptron.

- Recurrent Neural Networks (RNN): RNN memiliki koneksi yang membentuk siklus, memungkinkan informasi dari output sebelumnya untuk digunakan sebagai input, yang membuatnya cocok untuk tugas-tugas seperti pemrosesan bahasa alami dan pengenalan ucapan.
- Convolutional Neural Networks (CNN): CNN sangat efektif untuk tugas-tugas pengolahan gambar dan video karena kemampuannya untuk mengenali pola spasial dalam data.
- Self-Organizing Maps (SOM): SOM digunakan untuk pengklasteran dan visualisasi data berdimensi tinggi. SOM memetakan input ke dalam grid dua dimensi dan mengklasifikasikan data tanpa pengawasan.
- Modular Neural Networks: Jaringan ini terdiri dari beberapa ANN yang bekerja secara independen dan bersama-sama untuk memecahkan masalah yang sama.
- Radial Basis Function Networks (RBFN): RBFN menggunakan fungsi basis radial sebagai fungsi aktivasi dan biasanya digunakan untuk tugas-tugas seperti interpolasi dan klasifikasi.

Berikut adalah kelebihan dan kekurangan dari ANN:

- Kelebihan ANN:
  1. Kemampuan Belajar dan Adaptasi: ANN dapat belajar dari data dan memperbaiki diri sendiri tanpa perlu diprogram secara eksplisit.
  2. Pemrosesan Paralel: ANN mampu melakukan lebih dari satu tugas secara bersamaan, berkat nilai numeriknya.
  3. Pengenalan Pola: ANN sangat efektif dalam mengenali pola yang kompleks, yang berguna dalam aplikasi seperti pengenalan wajah dan suara.
- Kekurangan ANN:
  1. Ketergantungan pada Data: ANN membutuhkan jumlah data latih yang besar untuk performa yang optimal, yang bisa menjadi masalah jika data terbatas.
  2. Kebutuhan Hardware: ANN memerlukan hardware dengan spesifikasi tinggi, terutama untuk jaringan yang besar dan kompleks.
  3. Waktu Pelatihan: ANN bisa membutuhkan waktu yang lama untuk dilatih, terutama jika jumlah data yang diolah besar.

Sumber :

<https://www.pengalaman-edukasi.com/2020/06/pembahasan-dasar-algoritma-artifical.html>

<https://sis.binus.ac.id/2022/04/21/mengenal-3-jenis-neural-network-pada-deep-learning/>

<https://dailyhabibie.com/kelebihan-dan-kekurangan-artificial-neural-network/>

<https://blog.algorit.ma/artificial-neural-networks/>

<https://yunusmuhammad007.medium.com/6-artificial-neural-network-ann-part-1-pengenalan-db487b8f8d85>

<https://rifqimulyawan.com/blog/pengertian-ann/>

<https://ichi.pro/id/fungsi-aktivasi-dasar-dasar-sigmoid-relu-leaky-relu-dan-softmax-untuk-neural-networks-dan-deep-learning-155703049115934>