

Artificial Neural Network

Procedure 方法

Artificial Neural Network adalah sebuah model machine learning yang bekerja menyerupai otak manusia. ANN terdiri dari 3 jenis layer: Input, Hidden, dan Output. Linear layer berisi neuron-neuron yang mengandung weight dan bias yang dimiliki per layer. Activation layer berfungsi untuk melakukan transformasi pada output tiap layer agar bisa membangun sebuah space yang lebih kompleks menurut Universal Approximation Theorem, yang menyatakan bahwa NN dapat mengaproksimasi segala fungsi kontinu dengan aktivasi dan layer yang cukup. Tanpa activation layer, NN akan mirip sebuah linear regressor.

1. Membangun Arsitektur

Arsitektur dari NN ditentukan di-awal dengan susunan layernya. Input layer memiliki input size kolom data. Hidden layer merupakan satu atau lebih lapisan diantara input dan output yang memproses data. Biasanya disusun selang-seling oleh linear dan activation layer. Output layer memiliki out size sesuai jumlah kelas yang diprediksi, atau 1 untuk regresi.

2. Inisialisasi Weights dan Biases

Weights dan Biases tiap layer linear di-inisialisasi sesuai metode yang diinginkan, misalnya kaiming uniform, xavier_uniform, atau lainnya.

3. Forward Propagation

Data di-proses oleh NN dengan cara melakukan perkalian dengan weights, ditambah dengan bias, kemudian dioper ke layer selanjutnya. Atau pada kasus activation layer akan diterapkan activation function.

4. Backward Propagation

Setelah sampai di output layer, error dihitung sesuai dengan loss function yang dipilih. Lalu gradient dioper ke belakang untuk dilakukan batch gradient descent. Pengoperan dan update gradien didasarkan pada konsep chain rule yang diaplikasikan kepada gradien weight dan gradien bias untuk menghasilkan gradien input yang dioper ke belakang.

5. Ulangi 3-5 hingga jumlah epoch yang ditentukan

Prediksi dilakukan dengan hasil voting semua classifier. Per-classifier prediksi dilakukan dengan menentukan di sisi mana dari hyperplane prediksi tersebut jatuh.

6. Prediksi

Prediksi dilakukan dengan memproses data melalui semua layer hingga mendapatkan output.

VS Pytorch

Hasil di notebook DoE menghasilkan nilai F1 yang lebih tinggi. Hal tersebut menandakan implementasi ANN from scratch sudah cukup baik.

Potential Improvements

- Menambahkan optimizer seperti Adam, Adagrad, dll.
- Menambahkan jenis layer seperti Batch Normalization atau dropout

Bonus Convolutional Neural Network

Convolution Layer

Convolution secara matematis berarti sebuah sum dari diagonal perkalian terbalik oleh 2 baris angka. Atau juga bisa direpresentasikan sebagai perkalian 2 buah polinom, dengan koefisiennya menjadi nilai konvolusi. Dari motivasi tersebut, Convolutional Layer dalam image processing bekerja dengan membuat sebuah grid window yang disebut kernel atau filter, yang memiliki bobot dalam tiap kotaknya. Kernel ini bekerja dengan melakukan in-place multiplication pada pixel yang didalamnya, kemudian melakukan penjumlahan dari seluruh kotak dalam gridnya. Dari penggeseran ke seluruh gambar didapatkan sebuah feature map. Tergantung pada bobot kernel, fungsi dari convolution bisa menghasilkan output seperti edge detection, sharpening, gaussian blur, dll. Untuk kecepatan operasi, sebenarnya Convolution bisa didapatkan bukan hanya dari looping naif, namun juga bisa dengan menggunakan Fast Fourier Transform.

Max Pooling Layer

Max Pooling Layer adalah sebuah layer sliding window lagi, yang berfungsi untuk mereduksi dimensionalitas, pada kasus ini 2D menjadi 1D. Caranya cukup simple, pada setiap pergeseran sliding window, ambil nilai pixel maksimum dari seluruh window tersebut.

Convolution Layer vs Fully Connected Layer

1. FC Layer menerima input gambar sebagai flat array, sedangkan Convolution Layer langsung bekerja pada ruang 2D dari gambar. Karena itu, Convolution Layer dapat mempelajari hubungan sebuah pixel dari pixel berdekatannya sehingga dapat mempelajari hubungan lebih kompleks dibandingkan FC Layer yang melihat pixel satu-per-satu.
2. Karena Convolution menggunakan satu kernel dengan weights yang sama yang digeser ke seluruh gambar, jumlah parameter jauh lebih sedikit dibandingkan FC

Layer yang harus sesuai dengan shape input. Hal tersebut jauh mengurangi kompleksitas dan kerentanan overfit.

3. Sejalan dengan kerentanan overfit, karena Convolution Layer menggunakan weights yang sama di seluruh gambar, maka Convolution Layer akan bersifat invariant terhadap translasi dari sebuah gambar. Weights Convolution Layer tidak terikat pada posisi-posisi pixel sehingga akan bekerja sesuai ekspektasi jika gambar mengalami translasi, misalnya identifikasi jenis baju, dan ada yang di tengah, kiri, dan kanan. Secara feature map, hasil akan sesuai yang diinginkan. Berbeda dengan FC Layer yang weightsnya terikat pada pixel, sehingga translasi akan langsung membuat FC Layer tidak berguna.