

Gama Candra Tri K

15/378060/PA/16535

22 March 2018

Backpropagation

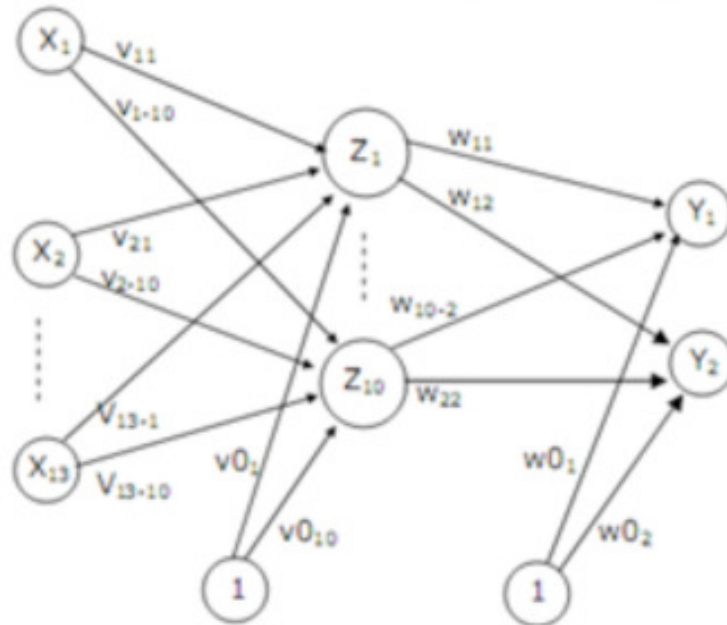
multilayer neural-network

Backpropagation adalah metode yang digunakan pada jaringan syaraf tiruan untuk menghitung gradien yang dibutuhkan dalam perhitungan bobot yang akan digunakan dalam jaringan. Hal ini biasa digunakan untuk melatih jaringan syaraf dalam, sebuah istilah yang digunakan untuk menjelaskan jaringan syaraf tiruan dengan lebih dari satu lapisan *hidden layer*.

Backpropagation adalah kasus khusus dari teknik yang lebih tua dan lebih umum yang disebut **automatic differentiation**. Dalam konteks *machine learning*, backpropagation umumnya digunakan oleh algoritma optimasi gradien descent untuk mengatur bobot neuron dengan menghitung gradien fungsi loss. Teknik ini juga terkadang disebut *error propagasi balik*, karena kesalahan dihitung pada output dan didistribusikan kembali melalui lapisan jaringan.

Arsitektur algoritma backpropagation terdiri dari tiga layer, yaitu input layer, hidden layer dan output layer. Pada input layer tidak terjadi proses komputasi, namun pada input layer terjadi pengiriman sinyal input X ke hidden layer. Pada hidden dan output layer terjadi proses komputasi terhadap bobot dan bias dan dihitung pula besarnya output dari hidden dan output layer tersebut berdasarkan fungsi aktivasi tertentu. Dalam algoritma backpropagation ini digunakan fungsi aktivasi sigmoid biner, karena output yang diharapkan bernilai antara 0 sampai 1.

Input Layer Hidden Layer Output Layer



Tiga layer backpropagation adalah input layer, hidden layer dan output layer. Pada input layer, inputan divariabelkan dengan X_n . Pada hidden layer, terdapat bobot (V_{ij}) dan bias (V_{0j}), serta Z sebagai data hidden layer. Pada output layer juga demikian, terdapat bobot (W_{ij}) dan bias (W_{0j}) dengan data output divariabelkan dengan Y .

Algoritma backpropagation adalah sebuah algoritma untuk memperkecil tingkat error dengan menyesuaikan bobot berdasarkan perbedaan output dan target yang diinginkan.

Secara umum algoritmanya terdiri dari tiga langkah utama, yaitu :

1. Pengambilan input
2. Penelusuran error
3. Penyesuaian bobot

Pada pengambilan input, terlebih dahulu dilakukan inisialisasi bobot, kemudian masuk ke dalam algoritma proses backpropagation yang terdiri dari komputasi maju yang bertujuan untuk menelusuri besarnya error dan komputasi balik untuk mengupdate dan menyesuaikan bobot. Dalam mengupdate bobot dapat dilakukan dengan dua cara, yaitu tanpa momentum dan dengan momentum. Namun, yang dijelaskan di bawah ini dalam mengupdate bobotnya dilakukan tanpa memperhatikan besarnya momentum. Dengan demikian dalam metode backpropagation, algoritma yang harus dilakukan adalah inisialisasi bobot, komputasi feed

forward dan backpropagation dan inisialisasi kondisi stopping berdasarkan nilai batas error atau jumlah batas epoch. Epoch merupakan rangkaian langkah dalam pembelajaran ANN (Artificial Neural Network). Satu epoch diartikan sebagai satu kali pembelajaran ANN.

Kadang-kadang disebut sebagai fungsi *cost* atau fungsi *error* (berbeda dengan *Gaussian error function*), fungsi loss adalah fungsi yang memetakan nilai dari satu atau lebih variabel ke bilangan real yang secara intuitif mewakili beberapa "cost" yang terkait dengan nilai-nilai tersebut. Untuk backpropagation, fungsi loss menghitung selisih antara output jaringan dan output yang diharapkan, setelah kasus menyebar melalui jaringan.

Algoritma optimasi mengulangi siklus update dua fase, propagasi dan weight. Ketika sebuah vektor masukan ditampilkan ke jaringan, ia diteruskan ke depan melalui jaringan, lapis demi lapis, hingga mencapai lapisan terluar. Output jaringan kemudian dibandingkan dengan output yang diinginkan, menggunakan fungsi **loss**. Nilai error yang dihasilkan dihitung untuk masing-masing neuron di lapisan output. Nilai error kemudian disebarkan dari output kembali melalui jaringan, sampai setiap neuron memiliki nilai kesalahan terkait yang mencerminkan kontribusinya terhadap output asli.

Backpropagation menggunakan nilai-nilai error ini untuk menghitung gradien dari fungsi loss. Pada tahap kedua, gradien ini dipindahkan ke metode optimasi, yang pada gilirannya menggunakannya untuk memperbarui weight, dalam upaya untuk meminimalkan fungsi loss.

Algoritma training dapat dibagi menjadi dua fase: propagasi dan weight.

Tahap 1: Propagation

Setiap propagasi melibatkan langkah-langkah berikut:

1. Propagasi maju melalui jaringan untuk menghasilkan nilai output (s)
2. Perhitungan cost (*error term*)
3. Propagasi aktivasi output kembali melalui jaringan menggunakan target pola training untuk menghasilkan delta (perbedaan antara nilai output yang ditargetkan dan aktual) dari semua output dan neuron tersembunyi.

Tahap 2: Weight update

Untuk setiap weight, langkah-langkah berikut harus diikuti:

1. Delta keluaran weight dan aktivasi input dikalikan untuk menemukan weight gradient.
2. Rasio (persentase) dari weight gradient dikurangi dari weight.

Rasio ini (persentase) mempengaruhi kecepatan dan kualitas training yang disebut juga tingkat training. Semakin besar rasio, semakin cepat neuron melatih, tetapi semakin rendah rasio, semakin akurat pelatihannya. Tanda gradien berat menunjukkan apakah kesalahan bervariasi secara langsung dengan, atau berbanding terbalik dengan, weight. Oleh karena itu, weight harus diperbarui ke arah yang berlawanan, "menurunkan" gradient. Training diulang (pada batch baru) hingga jaringan bekerja dengan baik.