**M Reza Sunday, 22 December 2024**

**Machine Learning Neural Network & Keras’s Tensorflow**

1. **Neural Network?**

Neural Network adalah model pembelajaran mesin yang terdiri dari beberapa lapisan node (neuron) yang terhubung satu sama lain. Hubungan antar-neuron ini memiliki bobot (weight) dan bias yang dapat dioptimalkan selama pelatihan. Neural Network digunakan untuk menangkap pola kompleks dalam data, baik untuk prediksi, klasifikasi, maupun pengenalan pola.

Apa yang Terjadi Ketika Neural Network Dilatih?

1. Forward Pass: Data input dikirimkan melalui jaringan lapisan demi lapisan. Setiap neuron menghitung outputnya berdasarkan kombinasi linier dari input (z=Σ(weight×input)+biasz = \Sigma (weight \times input) + biasz=Σ(weight×input)+bias) yang diterapkan pada fungsi aktivasi (misalnya, sigmoid, ReLU).
2. Loss Calculation: Perbedaan antara output model (prediksi) dengan nilai sebenarnya dihitung menggunakan fungsi loss (misalnya, Mean Squared Error untuk regresi).
3. Backward Pass (Backpropagation): Gradien dihitung untuk setiap bobot menggunakan aturan rantai (chain rule) sehingga jaringan tahu ke arah mana harus memperbarui parameter untuk mengurangi kesalahan.
4. Parameter Update: Optimizer (misalnya, Adam) digunakan untuk memperbarui bobot berdasarkan gradien.
5. **Input Layer, Hidden Layer, dan Output Layer!**

Input Layer

Input layer menerima data mentah dalam bentuk tensor. Data ini biasanya diproses sebelumnya, seperti normalisasi atau standardisasi.

* Apa yang Terjadi? Input layer hanya mengalihkan data ke hidden layer tanpa operasi.
* Contoh: Jika Anda memiliki 3 fitur (misalnya, harga pembukaan, tertinggi, dan volume), maka input layer akan memiliki 3 neuron.

Hidden Layer

Hidden layer bertugas memproses data menggunakan bobot dan fungsi aktivasi. Data diubah menjadi representasi yang lebih abstrak.

* Apa yang Terjadi? Setiap neuron menghitung outputnya menggunakan kombinasi linier input dan fungsi aktivasi.
* Fungsi Aktivasi Umum:
  + ReLU (Rectified Linear Unit): f(x)=max(0,x)f(x) = max(0, x)f(x)=max(0,x) untuk menangani non-linearitas.
  + Sigmoid: Mengubah output menjadi probabilitas.

Output Layer

Output layer memberikan hasil akhir yang dapat berupa nilai kontinu (regresi) atau probabilitas kelas (klasifikasi).

* Apa yang Terjadi? Data dari hidden layer terakhir dihitung ulang menjadi output final.

1. **LSTM?**

Penjelasan Teoritis

LSTM adalah jenis khusus dari RNN yang mampu menangani masalah long-term dependencies. Dalam data sekuensial (misalnya, time-series), pola jangka panjang sering kali penting, tetapi RNN biasa sulit menangkapnya karena masalah "vanishing gradient".

Apa yang Terjadi di LSTM?

1. Forget Gate: Memutuskan informasi dari langkah waktu sebelumnya yang harus dilupakan.
2. Input Gate: Memilih informasi baru yang harus disimpan dalam memori.
3. Output Gate: Menentukan bagian mana dari memori yang digunakan untuk output saat ini.
4. **Pengaruh Optimizer, Batch Size, dan Epoch!**

Optimizer

* Adam: Menggunakan momentum dan rata-rata gradien terjaga, cepat.
* RMSProp: Menjaga laju belajar berbeda untuk setiap parameter, cocok untuk time-series.
* SGD: Sederhana, lambat tetapi stabil.

Batch Size

* Ukuran kecil: Memperbarui bobot lebih sering, sehingga bisa belajar pola kecil tetapi lebih lambat.
* Ukuran besar: Menghemat memori tetapi kurang baik dalam mendeteksi pola kecil.

Epoch

* Terlalu sedikit epoch: Model tidak belajar cukup.
* Terlalu banyak epoch: Model berisiko overfitting.

1. **Perbedaan Grid Search, Random Search, dan Bayesian Search?**

|  |  |  |  |
| --- | --- | --- | --- |
| Metode | Proses | Keunggulan | Kekurangan |
| Grid Search | Menguji semua kombinasi parameter. | Menemukan kombinasi optimal. | Butuh waktu komputasi tinggi. |
| Random Search | Memilih kombinasi parameter secara acak. | Cepat dan sederhana. | Tidak menjamin optimal. |
| Bayesian Search | Menggunakan probabilitas untuk mempersempit ruang parameter yang dicoba. | Efisien dan cepat. | Implementasi lebih kompleks. |

1. **Perbedaan Adamax, Adam, RMSProp, dan SGD!**

|  |  |  |  |
| --- | --- | --- | --- |
| Optimizer | Keterangan | Keunggulan | Kekurangan |
| Adam | Cepat, stabil, dan adaptif. | Cocok untuk banyak masalah. | Rentan overfitting. |
| Adamax | Variasi Adam yang lebih stabil. | Cocok untuk data dengan noise. | Lambat pada dataset besar. |
| RMSProp | Menyesuaikan learning rate adaptif. | Baik untuk time-series. | Tidak secepat Adam. |
| SGD | Update langsung berdasarkan gradien. | Stabil, cocok untuk dataset besar. | Lambat. |

1. **Pengaruh Batch Size!**

* Batch Size 4: Memungkinkan model mempelajari detail kecil, tetapi lambat dan memori tinggi.
* Batch Size 32: Lebih cepat tetapi bisa mengorbankan generalisasi model.

1. **Pengaruh Epoch!**

* Epoch 50: Cepat, tetapi berisiko underfitting.
* Epoch 100: Memberikan model lebih banyak kesempatan belajar, tetapi membutuhkan early stopping untuk mencegah overfitting.