

Laporan Teori - Chapter 4: Training Models

Pendahuluan

Chapter 4 membahas secara mendalam berbagai algoritma dan teknik untuk melatih model, dengan fokus utama pada model linear. Bab ini menjelaskan mekanisme inti di balik Regresi Linear, beberapa varian Gradient Descent sebagai metode optimisasi, cara menangani data non-linear menggunakan Regresi Polinomial, serta teknik-teknik fundamental untuk regularisasi guna mencegah *overfitting*. Terakhir, bab ini beralih ke model klasifikasi populer, yaitu Regresi Logistik dan Regresi Softmax.

1. Regresi Linear

Model Regresi Linear memprediksi nilai target dengan menghitung jumlah bobot dari fitur input, ditambah sebuah *bias term*. Pelatihan model ini berarti mencari nilai parameter (bobot dan *bias*) yang meminimalkan *cost function* tertentu, biasanya **Mean Squared Error (MSE)**.

- The Normal Equation

Normal Equation adalah sebuah formula matematis yang memberikan hasil akhir secara langsung. Ini adalah solusi analitis untuk menemukan nilai parameter optimal yang meminimalkan cost function.

- **Kelebihan:** Tidak memerlukan *hyperparameter* dan tidak ada proses iteratif. Hasilnya didapat secara instan.
- **Kekurangan:** Sangat tidak efisien secara komputasi ketika jumlah fitur sangat besar (misalnya, lebih dari 100.000 fitur). Kompleksitasnya meningkat drastis seiring dengan jumlah fitur.

- Gradient Descent (GD)

Gradient Descent adalah algoritma optimisasi iteratif yang bertujuan untuk menemukan nilai minimum dari sebuah cost function. Identya adalah dengan mengubah-ubah parameter secara bertahap ke arah penurunan gradien yang paling curam, seperti menuruni lembah.

- **Learning Rate:** Ukuran langkah pada setiap iterasi disebut *learning rate*. Ini adalah *hyperparameter* yang krusial. Jika terlalu kecil, algoritma akan membutuhkan waktu sangat lama untuk konvergen. Jika terlalu besar, algoritma bisa "melompat-lompat" melewati titik minimum dan bahkan divergen.
- **Varian Gradient Descent:**
 1. **Batch Gradient Descent:** Menggunakan **seluruh dataset** pelatihan pada setiap langkah iterasi untuk menghitung gradien. Metode ini konvergen secara stabil ke titik minimum, tetapi sangat lambat pada dataset yang besar.
 2. **Stochastic Gradient Descent (SGD):** Hanya menggunakan **satu instance acak** dari dataset pada setiap langkah. Ini membuatnya jauh lebih cepat. Namun, lintasanya menuju minimum tidak stabil dan "berisik". Sifat acaknya ini membantunya keluar dari minimum lokal, tetapi juga berarti model tidak pernah benar-benar "tenang" di titik

minimum. Biasanya, *learning rate* dikurangi secara bertahap (*learning schedule*) untuk membantu model konvergen.

3. **Mini-batch Gradient Descent:** Sebuah kompromi antara Batch GD dan SGD. Pada setiap langkah, algoritma ini menggunakan sekelompok kecil instance acak yang disebut *mini-batch*. Metode ini lebih stabil daripada SGD dan mendapatkan keuntungan dari optimisasi perangkat keras (GPU). Ini adalah varian yang paling umum digunakan dalam praktik.

2. Regresi Polinomial

Untuk data yang lebih kompleks dan non-linear, Regresi Polinomial dapat digunakan. Teknik ini tetap menggunakan model linear, tetapi dengan menambahkan **pangkat dari setiap fitur** sebagai fitur baru. Misalnya, untuk satu fitur x , kita dapat menambahkan x^2 , x^3 , dst. sebagai fitur baru. Setelah transformasi ini, model linear biasa dapat dilatih pada set fitur yang diperluas ini. Namun, penggunaan pangkat yang terlalu tinggi dapat dengan cepat menyebabkan model mengalami *overfitting*.

3. Kurva Pembelajaran (Learning Curves)

Kurva pembelajaran adalah alat diagnostik yang kuat untuk memeriksa apakah sebuah model mengalami *overfitting* atau *underfitting*. Kurva ini memplot *error* model pada *training set* dan *validation set* sebagai fungsi dari ukuran *training set*.

- **Model Underfitting:** Kedua kurva (training dan validasi) akan mencapai titik *plateau*, berada pada level *error* yang tinggi, dan saling berdekatan. Ini menandakan model terlalu sederhana.
- **Model Overfitting:** Terdapat celah (gap) yang signifikan antara kurva *training* (yang memiliki *error* sangat rendah) dan kurva validasi (yang memiliki *error* jauh lebih tinggi). Ini menandakan model terlalu kompleks dan tidak dapat menggeneralisasi dengan baik.

4. Regularisasi Model Linear

Regularisasi adalah teknik untuk membatasi model agar tidak terlalu kompleks, sehingga mengurangi risiko *overfitting*. Hal ini dicapai dengan menambahkan *term* regularisasi ke dalam *cost function* selama pelatihan.

- Ridge Regression

Menambahkan term regularisasi yang setara dengan norma l_2 dari vektor bobot. Tujuannya adalah untuk menjaga agar bobot model sekecil mungkin. Hyperparameter α mengontrol seberapa besar regularisasi yang diterapkan. Ridge Regression adalah pilihan regularisasi default yang baik.

- Lasso Regression

Lasso (Least Absolute Shrinkage and Selection Operator) Regression menggunakan norma l_1 dari vektor bobot. Karakteristik utamanya adalah kemampuannya untuk menghilangkan sepenuhnya bobot dari fitur yang paling tidak penting (yaitu, mengaturnya menjadi nol). Ini menjadikannya teknik yang berguna untuk melakukan seleksi fitur secara otomatis.

- Elastic Net

Merupakan jalan tengah antara Ridge dan Lasso. Term regularisasinya adalah campuran dari term Ridge dan Lasso, yang dikendalikan oleh hyperparameter rasio campuran, α . Elastic Net lebih disukai daripada Lasso ketika jumlah fitur lebih besar dari jumlah sampel pelatihan atau ketika beberapa fitur sangat berkorelasi.

- Early Stopping

Bentuk regularisasi yang berbeda untuk algoritma iteratif seperti Gradient Descent. Idennya sangat sederhana: hentikan pelatihan segera setelah error pada validation set mencapai titik minimum dan mulai meningkat lagi. Ini mencegah model terus belajar dan menjadi overfitting pada data pelatihan.

5. Regresi Logistik dan Softmax

- Regresi Logistik

Meskipun namanya "regresi", ini adalah algoritma klasifikasi biner. Model ini menghitung probabilitas sebuah instance termasuk dalam kelas tertentu. Probabilitas ini dihitung dengan memasukkan hasil dari model linear ke dalam fungsi logistik (sigmoid), yang menghasilkan output antara 0 dan 1. Pelatihannya bertujuan untuk meminimalkan cost function yang disebut log loss. Model ini menghasilkan batas keputusan (decision boundary) yang linear.

- Regresi Softmax

Juga dikenal sebagai Multinomial Logistic Regression, ini adalah generalisasi dari Regresi Logistik untuk mendukung klasifikasi multikelas secara langsung, tanpa perlu menggunakan strategi One-vs-Rest. Model ini menghitung skor untuk setiap kelas, kemudian melewatkannya melalui fungsi softmax untuk mendapatkan probabilitas setiap kelas. Cost function yang digunakan adalah Cross-Entropy, yang akan menghukum model jika ia memberikan probabilitas rendah pada kelas target yang benar.