

Analisis Kode dan Output

Struktur Kode Model MLP

Kode yang Anda berikan melibatkan pembuatan model **MLP (Multilayer Perceptron)** dengan dua lapisan tersembunyi yang masing-masing berukuran 4, 8, dan 16 neuron. Model ini menggunakan aktivasi **Softmax** untuk masalah klasifikasi, yang sangat sesuai untuk dataset yang memiliki beberapa kelas target. Kode juga mengonfigurasi beberapa hyperparameter penting, seperti jumlah **epoch**, **learning rate (LR)**, dan **batch size**.

Model dibangun dengan menggunakan **PyTorch**, di mana jaringan disusun menggunakan dua lapisan tersembunyi (hidden layers) dengan **ReLU** sebagai fungsi aktivasi, serta output layer menggunakan **Softmax**. Optimizer yang digunakan adalah **SGD** (Stochastic Gradient Descent), yang dipilih karena kesederhanaannya, namun seringkali lebih lambat dalam konvergensi dibandingkan dengan algoritma optimasi yang lebih canggih seperti **Adam**.

Di dalam loop pelatihan (training loop), terdapat perhitungan **loss** menggunakan fungsi **CrossEntropyLoss**, yang merupakan pilihan umum untuk masalah klasifikasi dengan banyak kelas. Model ini dilatih dengan iterasi sebanyak 250 epoch, yang bertujuan untuk mengurangi nilai loss dan meningkatkan akurasi pada data latih.

Performa Akurasi dan Loss

Akurasi yang diperoleh pada data latih berfluktuasi antara 45% hingga 54%, dengan beberapa peningkatan kecil selama pelatihan. Ini menunjukkan bahwa model dapat mencapai kinerja yang relatif konsisten setelah beberapa epoch, tetapi tidak ada lonjakan signifikan dalam akurasi meskipun jumlah epoch yang digunakan mencapai 250. Pada epoch pertama, akurasi sudah mencapai sekitar 42%, yang menunjukkan bahwa model dapat belajar dari data latih meskipun secara perlahan.

Loss menunjukkan penurunan yang stabil tetapi tidak signifikan. Nilai loss pada awal pelatihan relatif tinggi, seperti pada epoch pertama yang menunjukkan loss sekitar 1.66. Seiring berjalannya epoch, loss berkurang, tetapi pada epoch terakhir, loss hanya turun sedikit dari nilai awal, menunjukkan bahwa meskipun ada penurunan, model kesulitan untuk menemukan solusi yang lebih baik. Hal ini dapat mengindikasikan bahwa model tidak sepenuhnya mampu mengoptimalkan bobotnya untuk meminimalkan loss secara efektif.

Pengaruh Hyperparameter

Learning Rate (LR) memainkan peran penting dalam akurasi dan loss. Dengan **LR = 0.1**, akurasi mencapai sekitar 54% di epoch akhir, menunjukkan bahwa learning rate ini cukup baik untuk memungkinkan model bergerak menuju minimum loss. Namun, pada nilai LR yang lebih kecil (misalnya 0.001 dan 0.0001), akurasi menunjukkan penurunan tajam, dengan nilai akurasi yang tetap rendah, sekitar 29% hingga 32%. Ini mengindikasikan bahwa learning rate yang terlalu kecil membuat model kesulitan untuk melakukan pembaruan yang cukup besar pada bobot selama pelatihan, yang menyebabkan akurasi dan loss terjebak pada tingkat yang lebih buruk.

Batch Size juga mempengaruhi kecepatan pelatihan dan akurasi akhir. Dengan batch size yang lebih kecil, seperti 16 dan 32, model menunjukkan akurasi yang sedikit lebih baik, meskipun tetap berada di kisaran 45% hingga 54%. Sebaliknya, dengan batch size yang lebih besar, seperti 128, 256, dan 512, model cenderung menunjukkan akurasi yang lebih rendah, yang dapat mengindikasikan bahwa batch yang lebih besar memperlambat proses pembelajaran atau menyebabkan model terlalu cepat mengonvergensi ke solusi yang kurang optimal.

Overfitting dan Generalisasi

Overfitting dapat dilihat dari konsistensi akurasi yang tinggi pada data latih, namun tidak tercermin dalam akurasi uji. Sebagai contoh, pada beberapa konfigurasi, seperti dengan **batch size 16** dan **LR 0.1**, akurasi uji mencapai sekitar 52%, sementara akurasi pada data latih stabil pada sekitar 45%. Ini menunjukkan bahwa model mungkin lebih terlatih pada data latih dan tidak dapat menggeneralisasi dengan baik pada data uji.

Akurasi uji yang relatif rendah (di kisaran 44% hingga 52%) menunjukkan bahwa model belum dapat menggeneralisasi dengan baik ke data yang tidak terlihat selama pelatihan. Akurasi uji yang lebih rendah ini konsisten di hampir semua eksperimen, menunjukkan bahwa meskipun model berhasil belajar dari data latih, model ini kesulitan untuk diterapkan pada data uji yang tidak terlihat sebelumnya.

Fluktuasi selama Pelatihan

Terdapat fluktuasi dalam **akurasi dan loss** selama pelatihan. Misalnya, pada beberapa epoch seperti epoch 250, loss mungkin sedikit lebih tinggi daripada pada epoch sebelumnya, tetapi akurasi tetap berada dalam kisaran yang sama. Hal ini menunjukkan bahwa meskipun model terus dilatih, tidak ada kemajuan besar yang tercapai dalam meningkatkan kinerja model setelah sejumlah epoch tertentu.

Fluktuasi ini dapat terjadi karena model mencapai titik tertentu di mana ia kesulitan untuk menemukan minimum global atau local minima yang lebih baik. Pada titik ini, akurasi dan loss cenderung mencapai kestabilan, meskipun masih ada variasi kecil.

Perbedaan Akurasi Antara Epoch Awal dan Epoch Terakhir

Pada beberapa eksperimen, akurasi awal sudah menunjukkan angka yang cukup tinggi, misalnya sekitar 45%. Namun, selama pelatihan, akurasi cenderung tetap stagnan, tanpa peningkatan besar setelah beberapa epoch. Ini menunjukkan bahwa meskipun model belajar di awal, pada akhirnya model mencapai titik jenuh di mana ia tidak lagi bisa meningkatkan kinerjanya secara signifikan, baik pada data latih maupun data uji.

Pada beberapa eksperimen, seperti dengan **batch size 512** dan **learning rate 0.0001**, akurasi tidak pernah meningkat lebih dari 29%, dan loss hampir tidak berubah, yang menunjukkan bahwa model terjebak di minimum lokal dan kesulitan dalam proses pelatihan.