

Tugas 11

Nama : muhammad makhlufi makbullah

Kelas : TK 45 01

NIM : 1103210171

eksperimen hyperparameter tuning pada model **Multilayer Perceptron (MLP)**. Dalam konteks pembelajaran mesin, **hyperparameter tuning** merujuk pada proses mencari konfigurasi terbaik untuk hyperparameter model yang digunakan. Hyperparameter adalah parameter yang ditentukan sebelum pelatihan dimulai dan mempengaruhi proses pelatihan dan kinerja model, tetapi tidak dapat dipelajari oleh model itu sendiri.

Pada percobaan ini, kita menguji pengaruh berbagai hyperparameter terhadap kinerja model MLP, yang diukur menggunakan akurasi pada dataset uji. Percobaan yang dilakukan melibatkan pengujian terhadap **5 hyperparameter utama: jumlah lapisan tersembunyi, fungsi aktivasi, jumlah epoch, learning rate, dan ukuran batch**.

Apa Itu Hyperparameter?

Hyperparameter adalah pengaturan yang diatur sebelum pelatihan model dan mempengaruhi cara model berlatih. Setiap hyperparameter memiliki dampak signifikan pada kinerja model dan proses pelatihannya. Berikut adalah definisi dari setiap hyperparameter yang diuji dalam percobaan ini:

1. Jumlah Lapisan Tersembunyi (Hidden Layers):

- Mengacu pada jumlah lapisan dalam jaringan saraf yang tidak langsung terhubung dengan input atau output.
- Lapisan ini bertanggung jawab untuk mengekstrak fitur dari data masukan. Jumlah lapisan tersembunyi yang lebih banyak memungkinkan model untuk menangkap pola yang lebih kompleks dalam data.

2. Fungsi Aktivasi (Activation Function):

- Fungsi yang menentukan apakah neuron akan diaktifkan atau tidak. Fungsi aktivasi mengatur non-linearitas dalam jaringan saraf, yang penting agar model dapat belajar dari data yang kompleks.
- Fungsi aktivasi yang diuji dalam percobaan ini termasuk Linear, Sigmoid, ReLU, Softmax, dan Tanh.

3. Jumlah Epoch (Epochs):

- Merupakan jumlah iterasi pelatihan di mana model akan melihat seluruh dataset.

- Epoch yang lebih banyak memungkinkan model untuk belajar lebih lama, tetapi setelah titik tertentu, kenaikan epoch tidak lagi memberikan manfaat yang signifikan dan dapat menyebabkan overfitting.

4. **Learning Rate:**

- Merupakan ukuran langkah yang diambil oleh model dalam memperbarui bobot selama proses pelatihan.
- Learning rate yang terlalu besar dapat menyebabkan model tidak stabil, sedangkan learning rate yang terlalu kecil dapat memperlambat konvergensi.

5. **Ukuran Batch (Batch Size):**

- Merujuk pada jumlah sampel yang diproses dalam satu langkah pembaruan bobot selama pelatihan.
- Ukuran batch yang lebih kecil dapat mempercepat konvergensi, tetapi bisa meningkatkan variabilitas dalam pembaruan model. Sebaliknya, ukuran batch yang lebih besar dapat memperlambat pembaruan model, tetapi memberikan estimasi yang lebih stabil.

Tujuan Percobaan Hyperparameter Tuning

Tujuan utama dari percobaan ini adalah untuk menemukan kombinasi hyperparameter yang memberikan hasil terbaik dalam hal **akurasi** pada dataset uji. Dengan mengubah setiap hyperparameter dan mengamati dampaknya terhadap akurasi model, kita dapat menyimpulkan konfigurasi yang optimal untuk model MLP pada tugas klasifikasi.

Mengapa Hyperparameter Tuning Penting?

Hyperparameter tuning sangat penting karena dapat secara signifikan mempengaruhi **kinerja model**. Sebuah model yang baik bukan hanya dilatih dengan data yang tepat, tetapi juga dikonfigurasi dengan hyperparameter yang tepat. Jika hyperparameter tidak disetel dengan benar, model bisa menjadi terlalu sederhana (underfitting) atau terlalu kompleks (overfitting), yang dapat menurunkan kinerjanya pada data baru (data uji). Oleh karena itu, pemilihan hyperparameter yang tepat sangat krusial untuk menghasilkan model yang akurat dan efektif.

Hasil Percobaan Hyperparameter

Dalam percobaan ini, kami menguji beberapa kombinasi hyperparameter untuk memahami pengaruhnya terhadap akurasi model. Hasil yang diperoleh menunjukkan bahwa:

- **Jumlah Lapisan Tersembunyi:** Model dengan 3 lapisan tersembunyi memberikan akurasi terbaik, menunjukkan bahwa penambahan lapisan tersembunyi memungkinkan model untuk mempelajari pola yang lebih kompleks.
- **Fungsi Aktivasi:** Fungsi aktivasi **ReLU** memberikan hasil terbaik dibandingkan dengan fungsi lainnya seperti Linear atau Sigmoid, yang menunjukkan keunggulan ReLU dalam menangani non-linearitas pada tugas klasifikasi.

- **Jumlah Epoch:** Semakin banyak epoch, semakin baik performa model, meskipun setelah mencapai sekitar 100 epoch, penambahan epoch tidak memberikan peningkatan signifikan, menunjukkan bahwa model sudah mencapai konvergensi.
- **Learning Rate:** Learning rate 0.01 memberikan hasil terbaik, sementara learning rate yang terlalu besar atau terlalu kecil dapat menurunkan kinerja model, mengindikasikan bahwa learning rate yang seimbang sangat penting.
- **Ukuran Batch:** Ukuran batch yang lebih kecil (16 atau 32) memberikan hasil terbaik, menunjukkan bahwa model lebih efektif ketika diperbarui lebih sering, meskipun dengan estimasi yang lebih bervariasi.

Kesimpulan Percobaan Hyperparameter

1. **Pemilihan Hyperparameter yang Tepat:** Kombinasi yang optimal untuk model ini terdiri dari **3 lapisan tersembunyi**, **ReLU** sebagai fungsi aktivasi, **100 epoch**, **learning rate 0.01**, dan **batch size 32**. Pengaturan ini menghasilkan akurasi yang tinggi pada tugas klasifikasi.
2. **Pentingnya Eksperimen:** Percobaan ini menegaskan pentingnya menguji beberapa hyperparameter untuk menemukan konfigurasi terbaik. Tanpa eksperimen yang tepat, kita tidak akan dapat mengoptimalkan model dan mungkin akan terjebak dalam konfigurasi yang suboptimal.
3. **Kinerja Model:** Dengan menggunakan konfigurasi yang tepat, kita dapat memperoleh hasil yang jauh lebih baik dibandingkan hanya mengandalkan satu konfigurasi default.

Pada percobaan ini, kita menguji pengaruh berbagai hyperparameter pada model **Multilayer Perceptron (MLP)** yang diterapkan pada masalah klasifikasi. Beberapa hyperparameter yang diuji adalah jumlah lapisan tersembunyi, fungsi aktivasi, jumlah epoch, laju pembelajaran (learning rate), dan ukuran batch. Berikut adalah penjelasan langkah demi langkah, hasil dari percobaan, dan kesimpulan dari masing-masing eksperimen:

1. Eksperimen dengan Jumlah Lapisan Tersembunyi (Hidden Layers)

- **Langkah:**
 - Menguji pengaruh jumlah lapisan tersembunyi (1, 2, atau 3 lapisan) pada akurasi model.
 - Setiap konfigurasi menggunakan jumlah neuron yang sama di setiap lapisan tersembunyi (16, 32, 64).
 - Menggunakan **ReLU** sebagai fungsi aktivasi dan pelatihan selama 20 epoch.
- **Hasil:**

- **Akurasi** untuk konfigurasi:
 - 1 Lapisan: 85.37%
 - 2 Lapisan: 87.32%
 - 3 Lapisan: 95.12%
 - **Kesimpulan:**
 - Peningkatan jumlah lapisan tersembunyi menunjukkan peningkatan akurasi model. Meskipun peningkatan akurasi dari 1 ke 2 lapisan tidak terlalu besar, penambahan lapisan ke-3 memberikan peningkatan signifikan pada akurasi. Ini menunjukkan bahwa model lebih kompleks dengan penambahan lapisan tersembunyi.
-

2. Eksperimen dengan Fungsi Aktivasi

- **Langkah:**
 - Menguji berbagai fungsi aktivasi seperti **Linear, Sigmoid, ReLU, Softmax, dan Tanh** pada model dengan konfigurasi yang sama (jumlah lapisan tersembunyi 2 lapisan, 16 dan 32 neuron).
 - Pelatihan dilakukan selama 20 epoch.
 - **Hasil:**
 - **Akurasi** untuk setiap fungsi aktivasi:
 - Linear: 80.98%
 - Sigmoid: 81.46%
 - ReLU: 94.63%
 - Softmax: 82.44%
 - Tanh: 87.80%
 - **Kesimpulan:**
 - **ReLU** memberikan hasil terbaik di antara fungsi aktivasi yang diuji, dengan akurasi mencapai 94.63%. Fungsi aktivasi ini cenderung lebih efektif pada masalah klasifikasi dibandingkan fungsi lainnya seperti Linear atau Sigmoid. Ini mungkin karena sifat non-linieritas yang lebih kuat yang diberikan oleh ReLU.
-

3. Eksperimen dengan Jumlah Epoch

- **Langkah:**

- Menguji pengaruh jumlah epoch terhadap performa model dengan konfigurasi yang sama (2 lapisan tersembunyi, 16 dan 32 neuron, ReLU).
- Menguji epoch dengan jumlah yang bervariasi (1, 10, 25, 50, 100, 250).
- **Hasil:**
 - **Akurasi** untuk berbagai jumlah epoch:
 - 1 Epoch: 75.61%
 - 10 Epoch: 85.85%
 - 25 Epoch: 91.22%
 - 50 Epoch: 93.17%
 - 100 Epoch: 95.61%
 - 250 Epoch: 96.10%
- **Kesimpulan:**
 - Peningkatan jumlah epoch berkontribusi pada peningkatan akurasi model, yang menunjukkan bahwa model memerlukan waktu lebih lama untuk konvergen dan mencapai hasil terbaik. Namun, setelah mencapai 100 epoch, kenaikan akurasi mulai melambat, menunjukkan bahwa pelatihan lebih lanjut mungkin tidak memberikan peningkatan signifikan.