

Nama : Muhamad Hilmi Haidar

NIM : 1103213005

Berikut adalah analisis hyperparameter hasil klasifikasi MLP :

1. Analisis Epochs

Peningkatan jumlah epochs memiliki dampak yang signifikan pada akurasi model. Pada epochs yang rendah (1 atau 10), akurasi masih cukup rendah karena model belum memiliki cukup waktu untuk mempelajari pola dalam data. Ketika jumlah epochs ditingkatkan menjadi 50, model mencapai performa optimal dengan peningkatan akurasi yang signifikan. Pada jumlah epochs yang lebih tinggi (100 atau 250), peningkatan akurasi mulai melambat, menunjukkan bahwa model telah mencapai titik konvergensi.

- **Epochs Rendah (1, 10):** Akurasi rendah karena waktu pelatihan tidak cukup untuk belajar pola.
- **Epochs Menengah (50):** Akurasi optimal tercapai.
- **Epochs Tinggi (100, 250):** Akurasi stabil dengan peningkatan minimal.

Rekomendasi: Gunakan jumlah epochs antara **50–100** untuk hasil terbaik tanpa risiko overfitting.

2. Analisis Hidden Layers

Jumlah dan ukuran hidden layers menentukan kapasitas model untuk menangkap pola kompleks dalam data. Penambahan jumlah neuron meningkatkan akurasi, tetapi hanya sampai batas tertentu. Setelah mencapai kapasitas yang cukup, peningkatan jumlah neuron memberikan keuntungan yang minimal atau bahkan menyebabkan overfitting.

- **Jumlah Neuron Rendah (4, 8):** Model underfitting karena kapasitas tidak mencukupi.
- **Jumlah Neuron Menengah (16, 32):** Performa optimal dengan akurasi tinggi.

- **Jumlah Neuron Tinggi (64, 128):** Peningkatan tidak signifikan, dengan risiko waktu pelatihan lebih lama.

Rekomendasi: Hidden layers dengan **16–32 neuron** memberikan keseimbangan terbaik antara kapasitas dan efisiensi.

3. Analisis Learning Rates

Learning rate menentukan kecepatan model dalam mencapai konvergensi. Dalam eksperimen ini, learning rate menengah (0.01 atau 0.1) memberikan hasil terbaik, sedangkan nilai yang terlalu tinggi atau rendah menurunkan akurasi.

- **Learning Rate Tinggi (10, 1):** Model tidak stabil dan gagal mencapai solusi optimal.
- **Learning Rate Menengah (0.01, 0.1):** Performa terbaik dengan stabilitas pelatihan yang baik.
- **Learning Rate Rendah (0.001, 0.0001):** Model terlalu lambat belajar, menghambat konvergensi.

Rekomendasi: Gunakan learning rate **0.01** untuk hasil terbaik dengan stabilitas optimal.

4. Analisis Activation Functions

Fungsi aktivasi memengaruhi bagaimana model menangkap non-linearitas data. Dalam eksperimen ini, ReLU terbukti menjadi pilihan terbaik karena menangani masalah vanishing gradient dengan baik.

- **ReLU:** Memberikan akurasi tertinggi dan performa stabil.
- **Sigmoid:** Underperforming karena masalah vanishing gradient.
- **Tanh:** Sedikit lebih baik daripada Sigmoid, tetapi masih kalah dari ReLU.
- **Softmax:** Tidak cocok untuk hidden layers, meskipun efektif di output layer.

Rekomendasi: Gunakan **ReLU** untuk hidden layers dalam jaringan.

5. Analisis Batch Sizes

Ukuran batch memengaruhi waktu pelatihan dan sensitivitas gradien terhadap data. Batch size kecil memberikan akurasi terbaik karena gradien diperbarui lebih sering, meskipun membutuhkan waktu pelatihan lebih lama. Batch size besar mempercepat pelatihan tetapi mengurangi akurasi.

- **Batch Size Kecil (16, 32):** Akurasi terbaik, tetapi waktu pelatihan lebih lama.
- **Batch Size Menengah (64, 128):** Kompromi terbaik antara kecepatan dan akurasi.
- **Batch Size Besar (256, 512):** Pelatihan cepat tetapi akurasi menurun.

Rekomendasi: Gunakan batch size **32 atau 64** untuk keseimbangan optimal.

Kesimpulan

Berdasarkan analisis di atas, konfigurasi terbaik untuk klasifikasi MLP pada dataset ini adalah:

- **Epochs:** 50–100
- **Hidden Layers:** 2–3 layers dengan 16–32 neuron per layer
- **Learning Rate:** 0.01
- **Activation Function:** ReLU
- **Batch Size:** 32 atau 64

Pendekatan ini memberikan hasil yang optimal dengan waktu pelatihan yang efisien, tanpa risiko overfitting atau underfitting. Konfigurasi ini ideal untuk memaksimalkan performa model pada dataset yang diuji.