

Tugas 10

Nama : muhammad makhlufi makbullah

Kelas : TK 45 01

NIM : 1103210171

1. Penjelasan Langkah-Langkah dalam Program

A. Persiapan Data

1. Memuat Dataset:

- Dataset **bank.csv** yang berisi data pelanggan bank digunakan untuk memprediksi apakah seorang pelanggan akan berlangganan produk bank atau tidak.
- Dataset ini berisi berbagai fitur (fitur numerik dan kategorikal) yang menggambarkan karakteristik pelanggan.

2. Preprocessing Data:

- Kolom **kategorikal** (misalnya job, marital, education, dll) di-*encode* menggunakan **One-Hot Encoding**.
- Fitur numerik tetap dipertahankan dalam format numerik.
- Data hasil encoding dikombinasikan menjadi satu set fitur numerik yang akan digunakan oleh model.

3. Splitting Data:

- Data dibagi menjadi dua set: data pelatihan (X_train dan y_train) dan data pengujian (X_test dan y_test).
- Pemisahan ini dilakukan dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian.

B. Model Neural Network (MLP)

1. Arsitektur Model:

- Model yang digunakan adalah **Vanilla MLP (Multilayer Perceptron)**, yang terdiri dari beberapa lapisan tersembunyi dan satu lapisan output dengan fungsi aktivasi **Sigmoid** untuk klasifikasi biner.
- **Lapisan input:** Menerima input dengan jumlah fitur yang sesuai.

- **Lapisan tersembunyi:** Ada dua lapisan tersembunyi yang masing-masing memiliki 32 dan 16 neuron. Fungsi aktivasi yang digunakan adalah **ReLU** (Rectified Linear Unit).
- **Lapisan output:** Hanya memiliki satu neuron dengan fungsi aktivasi **Sigmoid** untuk menghasilkan nilai antara 0 dan 1, yang diinterpretasikan sebagai probabilitas.

2. Fungsi Loss:

- Digunakan **Binary Cross-Entropy Loss (BCELoss)** karena ini adalah masalah **klasifikasi biner** (apakah pelanggan berlangganan produk bank atau tidak).

3. Optimisasi:

- Optimizer yang digunakan adalah **Adam** dengan learning rate 0.001, yang sering digunakan karena kemampuannya dalam memperbaiki konvergensi model.

C. Pelatihan Model

1. Proses Pelatihan:

- Model dilatih selama **50 epoch**. Setiap epoch, model memperbarui bobotnya berdasarkan **loss** yang dihitung dari prediksi dan nilai aktual (target).
- Proses pelatihan juga mencatat nilai **loss** pada setiap epoch untuk dianalisis.

2. Evaluasi Model:

- Setelah pelatihan selesai, model diuji pada data pengujian untuk mengukur kinerja menggunakan **loss** dan **akurasi**.
- **Akurasi** dihitung dengan membandingkan prediksi model dengan label yang sebenarnya, dengan menganggap prediksi benar jika nilai lebih dari 0.5 (probabilitas lebih besar dari 50%).

2. Hasil Program

Setelah program dijalankan, hasil yang akan diperoleh adalah:

- **Training Loss** untuk setiap epoch.
- **Test Loss** setelah evaluasi di data pengujian.
- **Akurasi** pada data pengujian.

Contoh output dari program:

Epoch 1/50, Loss: 0.6789

Epoch 2/50, Loss: 0.6745

...

Test Loss (BCELoss): 0.3400

Test Accuracy: 86.45%

3. Penjelasan Hasil

1. Training Loss:

- **Training loss** menurun dari epoch ke epoch. Ini menunjukkan bahwa model mulai belajar dan memperbaiki prediksinya selama pelatihan.
- Loss yang lebih rendah mengindikasikan bahwa model lebih baik dalam meminimalkan kesalahan prediksi selama pelatihan.

2. Test Loss:

- **Test loss** menunjukkan seberapa baik model dalam memprediksi data yang belum pernah dilihat selama pelatihan.
- Jika test loss lebih tinggi daripada training loss, itu bisa menunjukkan bahwa model **overfitting** pada data pelatihan (model terlalu banyak menyesuaikan data pelatihan).
- Nilai test loss yang rendah menunjukkan bahwa model dapat generalisasi dengan baik pada data baru.

3. Akurasi:

- **Akurasi** mengukur persentase prediksi yang benar dibandingkan dengan total data.
- Sebagai contoh, jika akurasi 86.45%, ini berarti 86.45% dari prediksi model pada data pengujian cocok dengan label yang sebenarnya.

4. Perbandingan dan Evaluasi Kinerja

Beberapa faktor yang dapat Anda bandingkan untuk mengevaluasi model:

1. Perbandingan Antara Epoch:

- Anda dapat mencoba melatih model dengan jumlah **epoch yang berbeda** (misalnya 10, 25, 100, atau 250 epoch) untuk melihat bagaimana **training loss** dan **test loss** berubah seiring waktu.
- Biasanya, lebih banyak epoch akan meningkatkan akurasi pelatihan, tetapi jika terlalu banyak, model bisa **overfitting**. Anda dapat memantau test loss untuk menghindari hal ini.

2. Perbandingan Dengan Learning Rate:

- Anda bisa mencoba variasi **learning rate** (misalnya 0.01, 0.001, 0.0001) untuk melihat seberapa cepat model belajar dan seberapa baik model beradaptasi dengan data.
- **Learning rate yang lebih besar** mungkin menyebabkan model gagal untuk mengonvergensi (overshooting), sementara **learning rate kecil** bisa membuat pelatihan lebih lambat.

3. Perbandingan Dengan Batch Size:

- Penggunaan **batch size** yang lebih besar (misalnya 128 atau 256) mungkin mempercepat pelatihan tetapi dapat mengurangi akurasi.
- Batch size yang lebih kecil (misalnya 16 atau 32) dapat membuat pelatihan lebih stabil, tetapi lebih lambat.

4. Perbandingan Dengan Aktivasi Lainnya:

- Anda dapat mencoba berbagai fungsi aktivasi seperti **Sigmoid**, **Tanh**, dan **ReLU** pada hidden layers dan mengamati perubahan dalam **train loss**, **test loss**, dan **accuracy**.

5. Kesimpulan

- Hasil **training loss** dan **test loss** akan menunjukkan seberapa baik model belajar dari data.
- **Akurasi** memberikan gambaran umum tentang seberapa baik model mengklasifikasikan data dengan benar.
- Untuk mengoptimalkan model lebih lanjut, Anda bisa bereksperimen dengan berbagai konfigurasi model, jumlah epoch, learning rate, batch size, dan fungsi aktivasi untuk melihat perubahan dalam performa.