

Nama: Pratyenggo Damar I.P.

NIM: 1103194141

Technical Report: PyTorch Neural Network Classification

Pendahuluan

Masalah klasifikasi merupakan permasalahan di mana tujuannya adalah untuk mengelompokkan atau mengkategorikan data ke dalam kelas atau kategori yang berbeda berdasarkan fitur atau atribut tertentu. Tujuan utamanya adalah membuat prediksi atau penentuan terhadap kelas atau kategori yang tepat untuk setiap data baru berdasarkan pola atau informasi yang ditemukan dalam data pelatihan. Dalam berbagai bidang seperti ilmu komputer, statistik, dan machine learning, masalah klasifikasi sering digunakan untuk aplikasi seperti pengenalan pola, analisis data, dan prediksi.

Langkah-Langkah Menggunakan PyTorch

1. **Arsitektur Neural Network Klasifikasi:** Dalam PyTorch, arsitektur jaringan saraf untuk klasifikasi dapat dibuat menggunakan modul `torch.nn`. Ini mencakup lapisan-lapisan seperti `torch.nn.Linear` untuk lapisan-lapisan linier, fungsi aktivasi seperti `torch.nn.ReLU` untuk non-linearitas, dan `torch.nn.Softmax` untuk output layer pada masalah klasifikasi.
2. **Persiapan Data Klasifikasi:** Untuk memulai, kita perlu mengetahui dimensi fitur input dan jumlah kelas/output yang ingin diprediksi. Transformasi data menjadi format tensor dan pembagian menjadi set pelatihan dan pengujian dapat dilakukan menggunakan PyTorch.
3. **Membangun Model, Setup Loss Function dan Optimizer:** Model dibangun dengan menggunakan PyTorch, menyertakan lapisan-lapisan linear dan fungsi aktivasi non-linear. Loss function seperti `torch.nn.CrossEntropyLoss()` untuk klasifikasi multi-kelas dan optimizer seperti `torch.optim.SGD` digunakan untuk mengoptimalkan bobot jaringan.
4. **Melakukan Prediksi dan Evaluasi Model:** Setelah pelatihan model, prediksi dapat dilakukan pada data baru. Evaluasi model dilakukan dengan metrik-metrik seperti akurasi, presisi, recall, dan F1-score untuk mengukur performa klasifikasi.
5. **Peningkatan Model dan Non-Linearitas:** Model dapat ditingkatkan dengan menambahkan lapisan-lapisan baru, mengubah jumlah neuron dalam lapisan, menggunakan teknik regularisasi seperti dropout, atau melakukan fine-tuning pada hyperparameter. Non-linearitas diperlukan untuk memungkinkan model memahami pola-pola kompleks dalam data.
6. **Klasifikasi Multi-Kelas:** Dengan menggabungkan konsep-konsep sebelumnya, kita dapat membangun model klasifikasi multi-kelas menggunakan PyTorch. Proses ini

melibatkan pembuatan arsitektur jaringan, kompilasi loss function, dan pelatihan model menggunakan data.

Studi Kasus: CircleModelV2

Pembuatan Data Multi-Class: Data multi-class dibuat menggunakan metode `make_blobs` dari Scikit-Learn, diubah menjadi tensor, dan dibagi menjadi set pelatihan dan pengujian.

Pembuatan Model Multi-Class Classification: Model dibangun menggunakan PyTorch dengan tiga lapisan linear dan dapat menangani data multi-class.

Fungsi Kerugian dan Optimizer: Fungsi kerugian `nn.CrossEntropyLoss()` dan optimizer SGD dengan laju pembelajaran 0.1 digunakan.

Perolehan Probabilitas Prediksi untuk Model Multi-Class: Prediksi dilakukan dengan model, dan fungsi softmax digunakan untuk mengonversi logits menjadi probabilitas prediksi. Argmax dari probabilitas prediksi memberikan label prediksi.

Pelatihan dan Evaluasi Model: Model dilatih selama 100 epoch dengan evaluasi setiap 10 epoch. Performa model diukur menggunakan akurasi.

Evaluasi Tambahan dengan Metric Klasifikasi: Metric klasifikasi tambahan seperti presisi, recall, F1-score, confusion matrix, dan classification report diperkenalkan untuk memberikan pemahaman lebih mendalam tentang performa model.

Visualisasi Prediksi Model: Prediksi model divisualisasikan menggunakan `plot_decision_boundary()`.

Kesimpulan

Dengan mengikuti langkah-langkah di atas, PyTorch memfasilitasi pengembangan model klasifikasi neural network dengan kemampuan untuk menangani masalah klasifikasi multi-kelas. Penerapan non-linearitas, evaluasi dengan metrik klasifikasi, dan visualisasi prediksi membantu memahami dan meningkatkan performa model. Studi kasus CircleModelV2 memberikan contoh konkret dari penggunaan PyTorch untuk klasifikasi multi-kelas dengan hasil yang lebih memuaskan melalui pemahaman pola non-linear dalam data.