1. Pendahuluan

Klasifikasi teks adalah proses mengategorikan dokumen atau teks ke dalam kelompok tertentu. Dalam dunia digital, ini sangat penting untuk memahami opini publik, seperti menganalisis sentimen *online*.

Tugas ini bertujuan untuk membangun model yang dapat mengklasifikasikan ulasan produk menjadi dua kategori sentimen: Positif atau Negatif. Saya menggunakan arsitektur RNN, yang sangat efektif untuk memproses data sekuensial seperti teks.

2. Dataset

Sumber Data:

Dataset diambil dari Steam Review&Games Dataset di Kaggle.

Deskripsi Dataset:

Saya menggunakan 200 data ulasan dari dataset tersebut, dibagi rata menjadi 100 ulasan positif dan 100 ulasan negatif. Data ini dipilih karena merepresentasikan ulasan nyata dengan variasi panjang teks (dari kalimat singkat hingga paragraf) dan gaya bahasa informal yang umum di internet.

Alasan Pemilihan Dataset:

Dataset ini relevan untuk kasus analisis sentimen dan menyediakan variasi data yang diperlukan untuk menguji kemampuan model RNN dalam menangkap konteks yang berbeda.3. Implementasi Model

3.1 Arsitektur RNN

Saya menggunakan GRU (Gated Recurrent Unit). GRU adalah varian dari RNN yang lebih efisien dan baik dalam menangani ketergantungan jangka panjang dalam teks. Model kami terdiri dari:

* Lapisan Embedding: Mengubah setiap kata menjadi vektor numerik.
* Lapisan GRU: Memproses urutan kata dan memahami konteks.
* Lapisan Fully Connected (Klasifikasi): Mengambil output dari GRU untuk memprediksi sentimen (positif atau negatif).

3.2 Preprocessing

Saya melakukan langkah-langkah preprocessing berikut untuk menyiapkan data:

* Tokenisasi: Memecah teks menjadi kata-kata individu (token) dan mengubah semuanya menjadi huruf kecil.
* Normalisasi: Menghapus tanda baca dan karakter yang tidak relevan.
* Vocabulary Building: Membuat kamus kata dari data. Kata yang tidak dikenal diganti dengan <UNK>, dan kami menambahkan token <PAD> untuk menyamakan panjang urutan.
* Padding: Menyesuaikan setiap ulasan ke panjang 100 token (SEQ\_LEN=100).

3.3 Pengaturan Eksperimen

* Jumlah Epoch: 10
* Batch Size: 64
* Optimizer: Adam
* Loss Function: CrossEntropyLoss
* Embedding Dimension: 128
* Hidden Dimension GRU: 256

3.4 Log Eksperimen

Saya melakukan beberapa iterasi untuk menemukan arsitektur terbaik:

|  |  |  |  |
| --- | --- | --- | --- |
| Percobaan | Model | Hidden Dim | Catatan |
| #1 | RNN Sederhana | 128 | Performa sangat rendah (sekitar 58%). RNN sederhana tidak efektif untuk memahami konteks ulasan yang panjang. |
| #2 | GRU | 256 | Performa meningkat signifikan (sekitar 78.5%). GRU lebih baik dalam menangani dependensi jangka panjang. |
| #3 | GRU + Dropout 0.3 | 256 | Akurasi sedikit menurun (sekitar 75%). Dropout tidak diperlukan pada dataset sekecil ini, tetapi menunjukkan bahwa overfitting minimal. |

Saya memutuskan untuk menggunakan konfigurasi Percobaan #2 karena memberikan keseimbangan terbaik antara kinerja dan kompleksitas.

4. Evaluasi Hasil

Metrik dan Visualisasi:

* Saya menggunakan akurasi dan loss sebagai metrik utama. Akurasi model pada data testing mencapai sekitar 78.5%.

[Laporan PDF Anda akan mencakup visualisasi grafik: Learning Curve (Loss dan Akurasi dari Training dan Testing), dan Confusion Matrix.]

Analisis Performa:

Grafik learning curve menunjukkan bahwa model belajar dengan baik dan stabil. Confusion Matrix memberikan wawasan mendalam tentang kesalahan klasifikasi.

5. Refleksi Pribadi

Tantangan Utama:

Tantangan utama adalah menentukan panjang urutan (SEQ\_LEN) yang optimal. Awalnya, kami menggunakan padding yang terlalu pendek, yang membuat model kehilangan banyak konteks dari ulasan panjang.

Solusi dan Hasil:

Saya meningkatkan SEQ\_LEN dari 20 menjadi 100, yang memungkinkan model menangkap lebih banyak informasi dari setiap ulasan. Hasilnya, akurasi model meningkat drastis.

Pelajaran Penting:

Tugas ini mengajarkan bahwa meskipun RNN adalah alat yang kuat, preprocessing data yang tepat sangat menentukan keberhasilan model, terutama saat berhadapan dengan variasi panjang teks.

6. Kesimpulan dan Saran

Kesimpulan:

Model klasifikasi sentimen berbasis GRU berhasil mengklasifikasikan ulasan Steam dengan akurasi yang solid untuk dataset yang relatif kecil.

Rekomendasi untuk Pengembangan Selanjutnya:

* Menggunakan pretrained embeddings (seperti GloVe) untuk meningkatkan pemahaman model terhadap semantik kata.
* Melatih model dengan lebih banyak data ulasan untuk meningkatkan generalisasi.

7. Referensi

* [Steam Review&Games Dataset](https://www.kaggle.com/datasets/filipkin/steam-reviews)
* [Deep-Learning-Course/courses/week05 at master · virgantara/Deep-Learning-Course](https://github.com/virgantara/Deep-Learning-Course/tree/master/courses/week05)