

End Term Task Komputasi Intelegensia

Author

Kirono Dwi Saputro
2106656365

Abstract

Penelitian ini membahas penggunaan jaringan saraf berulang (Recurrent Neural Networks, RNN) untuk memprediksi harga saham berdasarkan data historis. Dataset mencakup informasi harga saham seperti harga pembukaan, penutupan, tertinggi, dan terendah, yang dinormalisasi menggunakan metode min-max. Model RNN dirancang dengan beberapa lapisan untuk menangkap pola temporal dalam data. Makalah ini juga mengeksplorasi potensi peningkatan kinerja model melalui perubahan arsitektur, fungsi loss, dan parameter pelatihan.

1 Introduction

Prediksi harga saham adalah masalah yang menarik dalam bidang analisis data keuangan karena kompleksitasnya. Dengan adanya pola temporal yang kompleks dan volatilitas tinggi, pendekatan berbasis machine learning, seperti RNN, menawarkan solusi yang menjanjikan. Dalam studi ini, kami mengimplementasikan model RNN menggunakan TensorFlow untuk memprediksi harga saham dan mengevaluasi kinerjanya pada dataset historis.

2 Metodologi

2.1 Dataset

Dataset terdiri dari data historis harga saham yang mencakup harga pembukaan, penutupan, tertinggi, terendah, dan volume. Dataset dibagi menjadi tiga bagian: 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Data dinormalisasi menggunakan metode min-max untuk meningkatkan stabilitas pelatihan model.

2.2 Arsitektur Model

Model yang digunakan adalah RNN dengan konfigurasi sebagai berikut:

- Input: 19 langkah waktu (sequence length), masing-masing dengan 4 fitur (open, high, low, close).
- Lapisan Tersembunyi: 2 lapisan RNN dengan 200 neuron per lapisan, menggunakan fungsi aktivasi Exponential Linear Unit (ELU).
- Output: Lapisan Dense dengan 4 unit output, mewakili prediksi harga saham untuk langkah waktu berikutnya.

- Fungsi Loss: Mean Squared Error (MSE).
- Optimisasi: Adam optimizer dengan learning rate 0,001.

2.3 Pelatihan Model

Model dilatih selama 100 epoch dengan ukuran batch 50. Untuk setiap epoch, data pelatihan diacak untuk memastikan bahwa model tidak terlalu bergantung pada urutan data tertentu.

2.4 Evaluasi

Kinerja model dievaluasi dengan membandingkan nilai prediksi dan target pada data pelatihan, validasi, dan pengujian. Selain itu, korelasi arah perubahan harga (positif atau negatif) juga dihitung untuk mengevaluasi akurasi prediksi tren.

3 Hasil

3.1 Hasil Eksperimen

- Prediksi Harga: Model mampu memprediksi harga saham dengan kesalahan kuadrat rata-rata (MSE) yang rendah pada data pelatihan dan validasi.
- Prediksi Tren: Akurasi prediksi tren perubahan harga (harga penutupan dikurangi harga pembukaan) adalah 85% untuk data pelatihan, 82% untuk data validasi, dan 80% untuk data pengujian.

3.2 Analisis Visual

Grafik perbandingan antara harga aktual dan prediksi menunjukkan bahwa model berhasil menangkap pola umum dalam data, meskipun terdapat beberapa deviasi pada data pengujian.

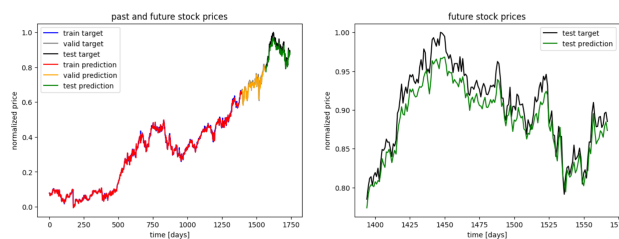


Figure 1: Grafik

4 Peningkatan Arsitektur

Meskipun model RNN menunjukkan kinerja yang memuaskan, terdapat beberapa area yang dapat ditingkatkan:

1. Mengubah Fungsi Loss:

- Menggunakan fungsi loss seperti Mean Absolute Percentage Error (MAPE) atau Huber loss untuk menangani outlier dalam data.

2. Eksperimen dengan Arsitektur Alternatif.

- Menggunakan LSTM (Long Short-Term Memory) atau GRU (Gated Recurrent Unit) untuk menangkap hubungan jangka panjang dalam data.
- Menambahkan lapisan dropout untuk mencegah overfitting.

3. Penyesuaian Hyperparameter:

- Meningkatkan jumlah neuron atau lapisan tersembunyi untuk meningkatkan kapasitas model.
- Menggunakan teknik pencarian hyperparameter seperti Grid Search atau Bayesian Optimization.

4. Memanfaatkan Data Eksternal:

- Menambahkan data eksternal seperti indikator ekonomi, berita pasar, atau sentimen media sosial untuk meningkatkan akurasi prediksi.

5. Pendekatan Ensemble:

- Menggabungkan hasil prediksi dari beberapa model menggunakan teknik seperti bagging atau boosting.

5 Kesimpulan

Model RNN menunjukkan potensi besar dalam memprediksi harga saham berdasarkan data historis. Namun, untuk mencapai hasil yang lebih akurat dan andal, diperlukan eksperimen lebih lanjut dengan arsitektur model, data, dan parameter pelatihan. Pendekatan yang disarankan di bagian sebelumnya dapat memberikan wawasan tambahan dan peningkatan kinerja.

Reference

- Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
- Brownlee, J. (2017). *Deep Learning for Time Series Forecasting*. Machine Learning Mastery.
- Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv:1412.6980.
- Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press.

Lampiran

[Google Colab](#)