**LSTM Neural Network**

Pada topik ini berfokus pada penggunaan arsitektur Deep LSTM Neural Network untuk memberikan prakiraan deret waktu multidimensi menggunakan Keras dan Tensorflow - khususnya pada kumpulan data pasar saham untuk memberikan indikator momentum harga saham. Pada arsitektur ini akan secara singkat menyentuh sel neuron LSTM, memberikan contoh mainan untuk memprediksi gelombang sinus kemudian berjalan melalui aplikasi ke deret waktu stokastik. Artikel ini mengasumsikan pengetahuan dasar tentang jaringan saraf dalam sederhana.

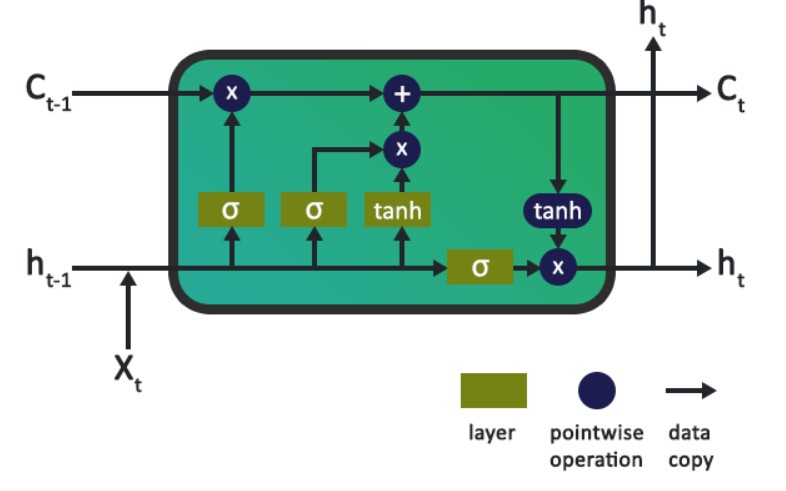
1. PENGERTIAN NEURON LSTM

Salah satu masalah mendasar yang melanda arsitektur jaringan saraf tradisional untuk waktu yang lama adalah kemampuan untuk menafsirkan urutan input yang saling mengandalkan untuk informasi dan konteks. Informasi ini bisa berupa kata-kata sebelumnya dalam kalimat untuk memungkinkan konteks memprediksi apa kata berikutnya, atau bisa juga informasi temporal dari suatu urutan yang memungkinkan konteks pada elemen berbasis waktu dari urutan itu. Sederhananya, jaringan saraf tradisional mengambil vektor data yang berdiri sendiri setiap saat dan tidak memiliki konsep memori untuk membantu mereka dalam tugas-tugas yang membutuhkan memori.

Upaya awal untuk mengatasi hal ini adalah dengan menggunakan pendekatan jenis umpan balik sederhana untuk neuron di jaringan di mana keluaran diumpankan kembali ke masukan untuk memberikan konteks pada masukan yang terakhir terlihat. Ini disebut Recurrent Neural Networks (RNNs). Sementara RNN ini bekerja sampai batas tertentu, mereka mengalami penurunan yang cukup besar sehingga setiap penggunaan yang signifikan dari mereka mengarah pada masalah yang disebut Masalah Gradien Hilang.

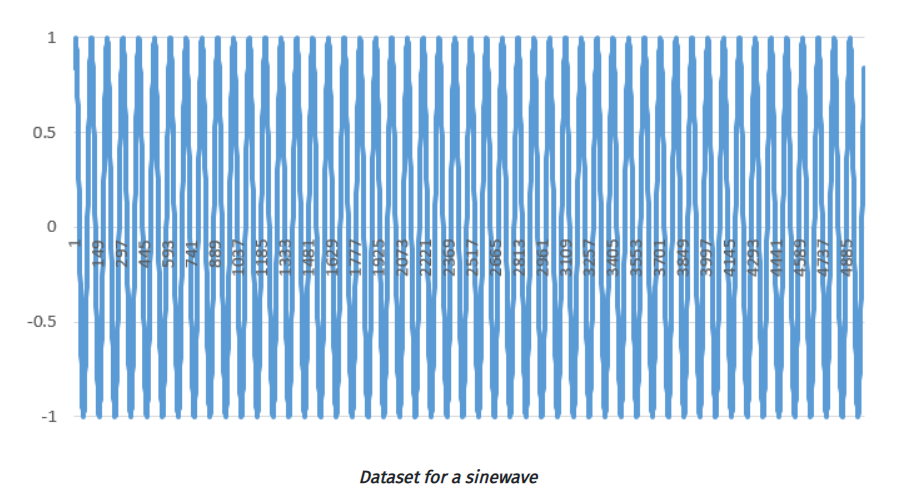
Di sini jaringan saraf Long Short Term Memory (LSTM) datang untuk menyelamatkan. Seperti neuron RNN, neuron LSTM menyimpan konteks memori dalam pipeline mereka untuk memungkinkan penanganan masalah sekuensial dan temporal tanpa masalah gradien menghilang yang memengaruhi kinerjanya. Pada artikel ini kami tidak akan membahas cara kerja LSTM yang kompleks karena kami lebih memperhatikan penggunaannya untuk masalah ini.

Dibawah ini merupakan diagram cara kerja bagian dalam neuron LSTM. Terdiri dari beberapa lapisan, dan operasi pointwise yang bertindak sebagai gerbang untuk input data, output dan status sel LSTM. Keadaan sel ini yang membuat memori dan konteks jangka panjang di seluruh jaringan dan input.



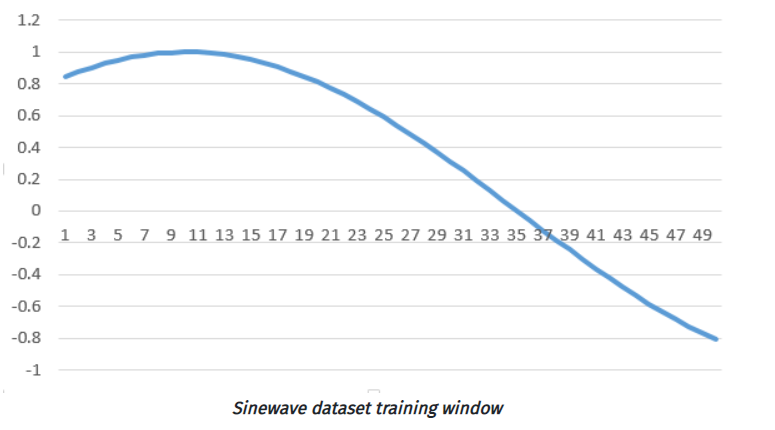
1. GELOMBANG SAMPING SEDERHANA

Untuk mendemonstrasikan penggunaan jaringan saraf LSTM dalam memprediksi deret waktu, dimulai dengan hal paling dasar yang dapat kita pikirkan yaitu deret waktu: gelombang sinus terpercaya. Data yang disediakan dalam folder data kode berisi file sinewave.csv yang kami buat yang berisi 5001 periode waktu gelombang sinus dengan amplitudo dan frekuensi 1 (memberikan frekuensi sudut 6,28) dan delta waktu 0,01. Hasilnya, jika diplot terlihat seperti ini:

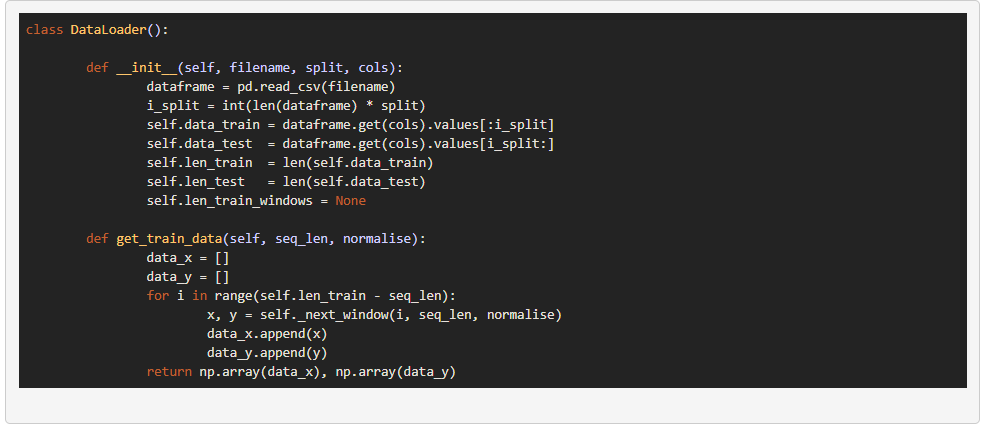


Selanjutnya apabila datanya sudah ada, dimulai dengan mengubah dan memuat data dari file CSV ke dataframe pandas yang kemudian akan digunakan untuk mengeluarkan array numpy yang akan memberi makan LSTM. Cara kerja lapisan Keras LSTM adalah dengan mengambil larik numpy 3 dimensi (N, W, F) di mana N adalah jumlah urutan pelatihan, W adalah panjang urutan dan F adalah jumlah fitur dari setiap urutan. Kami memilih untuk menggunakan panjang urutan (baca ukuran jendela) 50 yang memungkinkan jaringan sehingga dapat melihat sekilas bentuk gelombang sinus pada setiap urutan dan diharapkan akan membangun pola urutan berdasarkan pada windows sebelumnya yang diterima.

Urutannya sendiri adalah jendela geser dan karenanya bergeser 1 setiap kali, menyebabkan tumpang tindih yang konstan dengan jendela sebelumnya. Jendela pelatihan tipikal dengan panjang urutan 50, ketika diplot, ditampilkan di bawah ini:



Untuk memuat data ini kami membuat kelas DataLoader di kode kami untuk menyediakan abstraksi untuk lapisan pemuatan data, akan terlihat bahwa setelah inisialisasi objek DataLoader, nama file diteruskan, bersama dengan variabel terpisah yang menentukan persentase data yang akan digunakan. Pengujian dan variabel kolom yang memungkinkan untuk memilih satu atau lebih kolom data untuk analisis dimensi tunggal atau multidimensi.

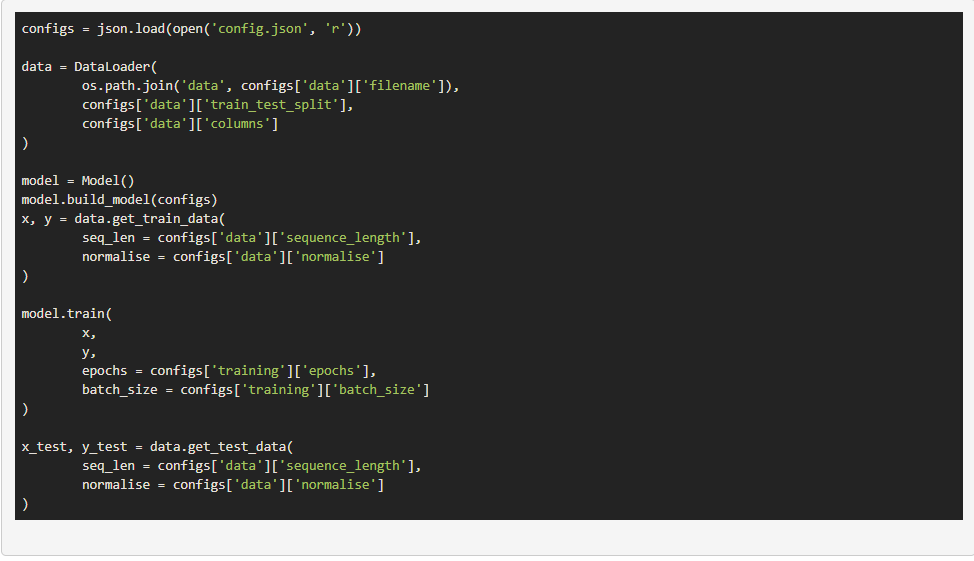


Setelah kita memiliki objek data yang memungkinkan kita memuat data, kita perlu membangun model jaringan saraf dalam. Sekali lagi untuk abstraksi, kerangka kode kami menggunakan kelas Model bersama file config.json untuk dengan mudah membuat instance model kami dengan arsitektur dan hyperparameter yang diperlukan yang disimpan dalam file konfigurasi. Fungsi utama yang membangun jaringan kita adalah fungsi build\_model () yang mengambil file konfigurasi parsing. Kode fungsi ini dapat dilihat di bawah dan dapat dengan mudah diperpanjang untuk digunakan di masa mendatang pada arsitektur yang lebih kompleks.

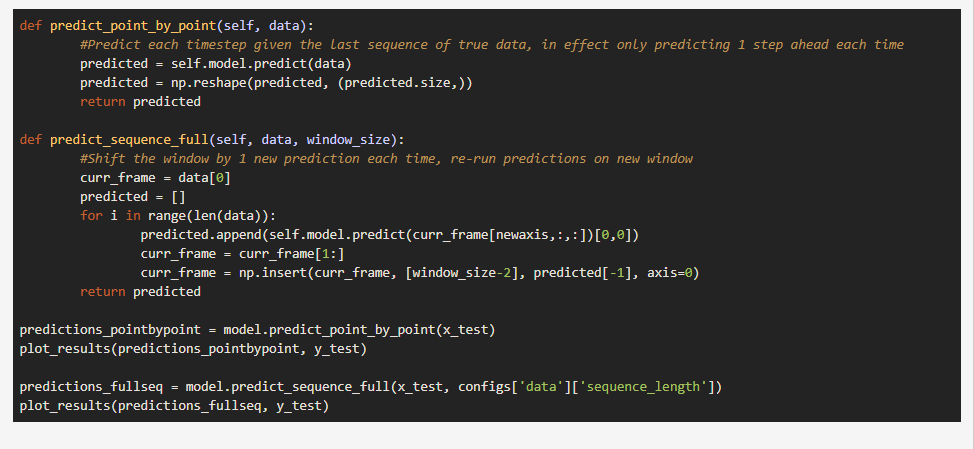


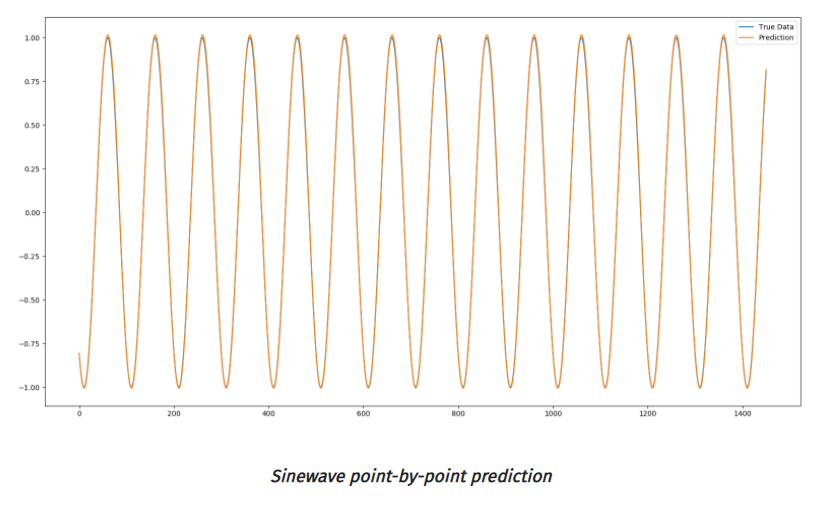
Dengan data yang dimuat dan model yang dibangun, sekarang kita dapat melanjutkan ke pelatihan model dengan training data. Untuk ini, kami membuat modul proses terpisah yang akan memanfaatkan abstraksi Model dan DataLoader kami untuk menggabungkannya untuk training, keluaran, dan visualisasi.

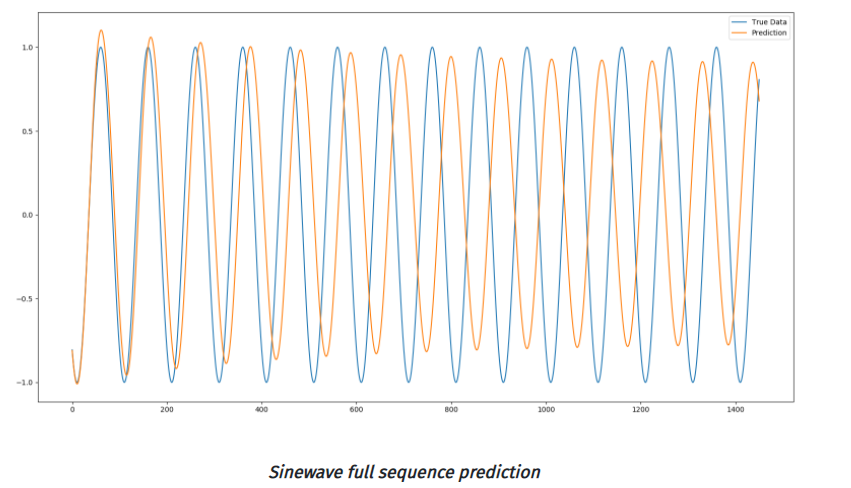
Di bawah ini adalah kode run thread umum untuk melatih model.



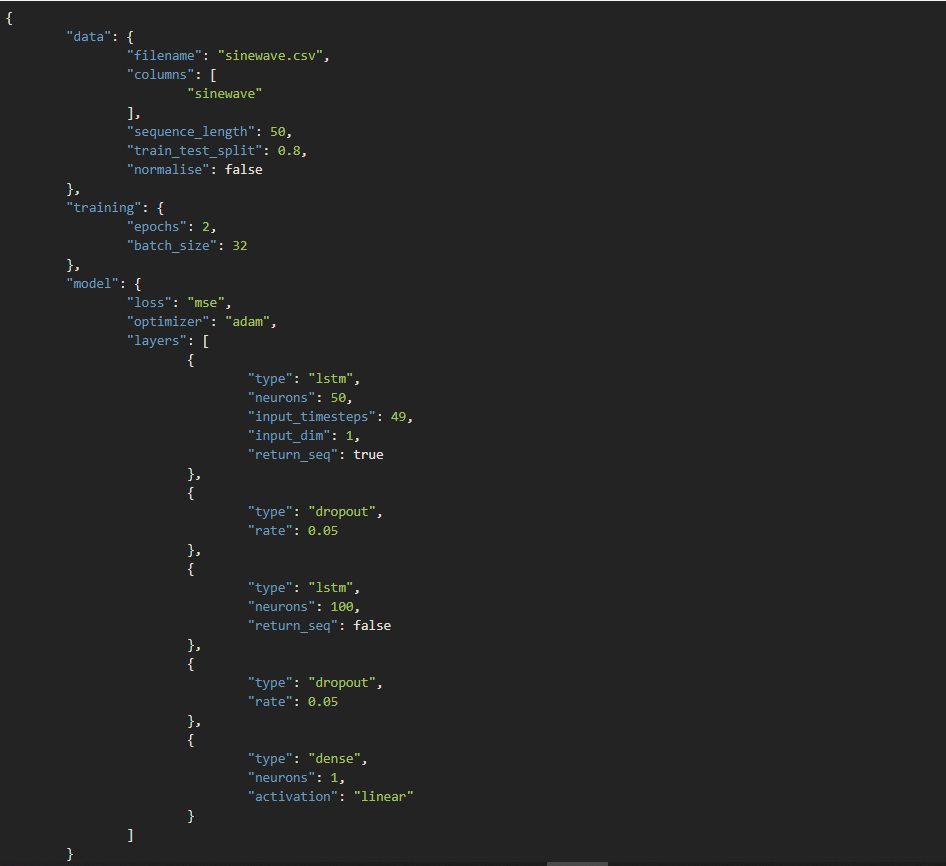
Untuk keluaran dijalankan dua jenis prediksi: yang pertama akan memprediksi dengan cara titik demi titik, yaitu hanya memprediksi satu titik di depan setiap kali, memplot titik ini sebagai prediksi, lalu melanjutkan jendela berikutnya. Dengan data pengujian lengkap dan memprediksi poin berikutnya sekali lagi. Prediksi kedua yang akan kita lakukan adalah memprediksi urutan lengkap hanya menginisialisasi training windows dengan bagian pertama dari training data sekali. Model kemudian memprediksi titik berikutnya untuk menggeser windows, seperti pada metode titik demi titik. Perbedaannya kemudian diprediksi menggunakan data pada prediksi sebelumnya. Pada langkah kedua ini berarti hanya satu titik data (titik terakhir) yang akan berasal dari prediksi sebelumnya. Pada prediksi ketiga, dua poin data terakhir akan berasal dari prediksi sebelumnya dan seterusnya. Setelah 50 prediksi, model selanjutnya akan memprediksi pada data sebelumnya. Hal ini memungkinkan penggunakan model untuk meramalkan banyak langkah waktu ke depan, yang nantinya akan meningkatkan tingkat kesalahan prediksi lebih jauh ke depan yang kita prediksi. Di bawah ini kita dapat melihat kode dan keluaran masing-masing untuk prediksi titik demi titik dan prediksi urutan lengkap.







Untuk referensi arsitektur jaringan dan hyperparameter yang digunakan untuk contoh sinewave dapat dilihat pada file config di bawah ini.



Dengan data true, kita dapat melihat bahwa hanya dengan 1 epoch dan kumpulan data pelatihan yang cukup kecil, jaringan saraf dalam LSTM telah melakukan pekerjaan yang cukup baik dalam memprediksi fungsi sinus. Kita dapat melihat bahwa saat memprediksi semakin banyak di masa depan, margin kesalahan meningkat karena kesalahan dalam prediksi sebelumnya semakin diperkuat saat digunakan untuk prediksi di masa mendatang. Dengan demikian dapat dilihat bahwa dalam contoh urutan lengkap, semakin jauh ke masa depan kita memprediksi, semakin kurang akurat frekuensi dan amplitudo prediksi dibandingkan dengan data yang sebenarnya. Namun karena fungsi sin adalah fungsi osilasi yang sangat mudah dengan noise nol, ia masih dapat memprediksinya dengan baik tanpa overfitting - ini penting, karena dapat dengan mudah menyesuaikan model dengan meningkatkan epoch dan mengeluarkan lapisan dropout untuk membuatnya hampir sepenuhnya akurat pada data pelatihan ini, yang memiliki pola yang sama dengan data pengujian, tetapi untuk contoh dunia nyata lainnya, overfitting model ke data pelatihan akan menyebabkan akurasi pengujian menurun karena model tidak akan digeneralisasi.

1. PASAR SAHAM YANG TIDAK SEDERHANA

Pada laporan ini diprediksikan beberapa ratus langkah waktu gelombang sinus secara akurat poin demi poin. Tidak seperti sinewave, deret waktu pasar saham bukan fungsi statis spesifik apa pun yang dapat dipetakan. Properti terbaik untuk mendeskripsikan pergerakan deret waktu pasar saham adalah jalan acak. Sebagai proses stokastik, true random walk tidak memiliki pola yang dapat diprediksi sehingga mencoba memodelkannya akan sia-sia. Terdapat argumen yang sedang berlangsung dari banyak pihak untuk mengatakan bahwa pasar saham bukanlah proses stokastik murni, yang memungkinkan untuk berteori bahwa deret waktu mungkin memiliki semacam pola tersembunyi. Dan pola tersembunyi inilah yang menjadi kandidat utama untuk diprediksi jaringan dalam LSTM.

Data yang akan digunakan pada laporan ini adalah file sp500.csv di folder data. File ini berisi harga Buka, Tinggi, Rendah, Tutup, serta Volume harian Indeks Ekuitas S&P 500 dari Januari 2000 hingga September 2018. Pada contoh pertama hanya akan membuat model satu dimensi dengan menggunakan harga Tutup saja. Menyesuaikan file config.json untuk mencerminkan data baru. Untuk penyimpanan sebagian besar parameter sama. Namun satu perubahan yang diperlukan tidak seperti sinewave yang hanya memiliki rentang numerik antara -1 hingga +1, harga penutupan adalah harga absolut yang terus bergerak dari pasar saham. Artinya, jika mencoba melatih model ini tanpa menormalkannya, model tersebut tidak akan pernah menyatu. Untuk mengatasi hal ini, diambil setiap windows training data / pengujian berukuran n dan menormalkan masing-masing untuk mencerminkan perubahan persentase dari awal jendela itu (sehingga data pada titik i = 0 akan selalu 0). Berikut merupakan persamaan untuk menormalkan dan kemudian melakukan de-normalisasi di akhir proses prediksi untuk mendapatkan angka dunia nyata dari prediksi:

n = daftar [jendela] perubahan harga yang dinormalisasi

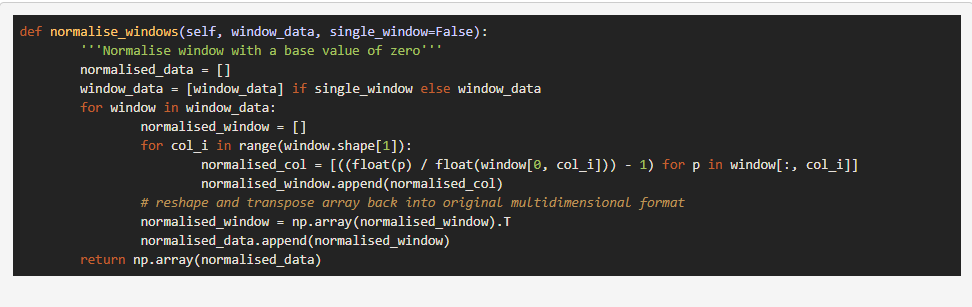
p = daftar mentah [jendela] harga pengembalian harian yang disesuaikan

Normalisasi:

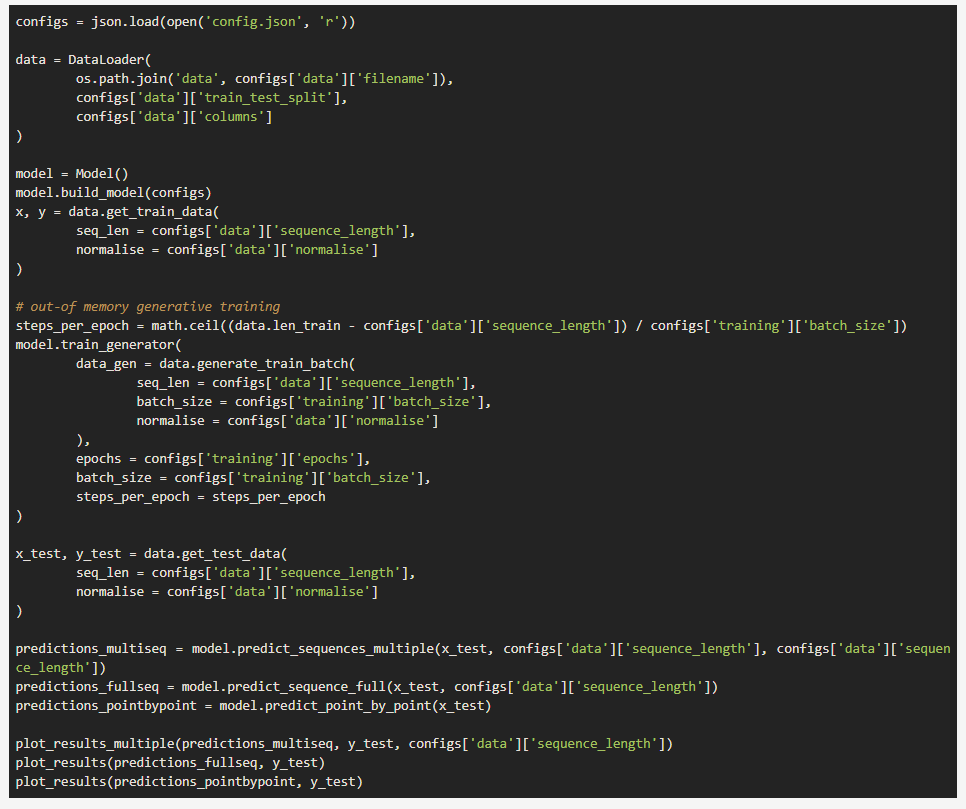


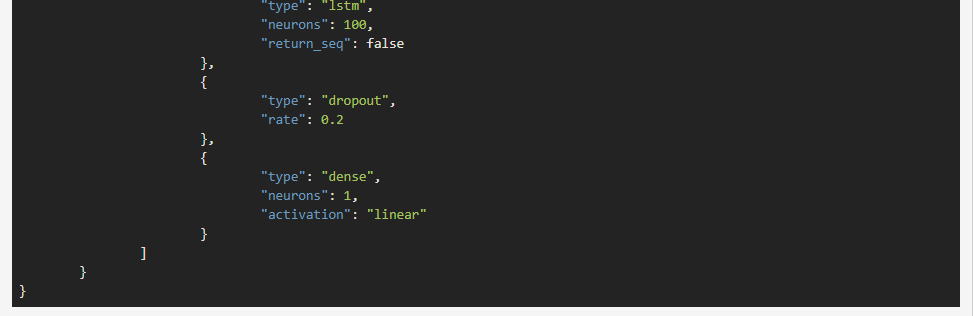
De-Normalisasi:

Kami telah menambahkan fungsi normalise\_windows () ke kelas DataLoader untuk melakukan transformasi ini, dan Boolean normalise flag terdapat dalam file konfigurasi yang menunjukkan normalisasi jendela ini.



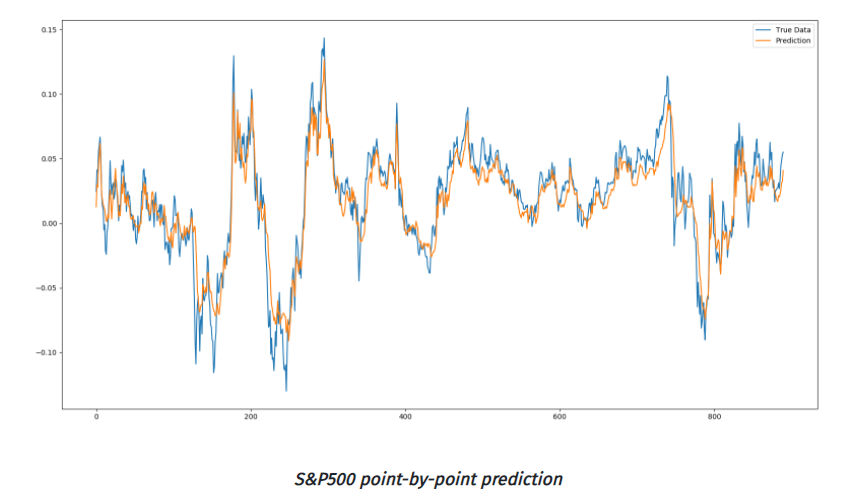
Dengan jendela yang dinormalisasi, kemudian dapat menjalankan model dengan cara yang sama seperti menjalankan terhadap data sinewave. Namun telah dibuat perubahan penting saat menjalankan data ini, dari menggunakan metode model.train () framework menjadi metode model.train\_generator () yang telah dibuat karena kehabisan memori saat mencoba melatih kumpulan data besar, karena fungsi model.train () memuat kumpulan data lengkap ke dalam memori, kemudian menerapkan normalisasi ke setiap jendela di memori, dengan mudah menyebabkan luapan memori. Jadi sebagai gantinya menggunakan fungsi fit\_generator () dari Keras untuk memungkinkan pelatihan dinamis dari kumpulan data menggunakan generator python untuk menggambar data, yang berarti pemanfaatan memori akan diminimalkan secara dramatis. Kode di bawah merinci run thread baru untuk menjalankan tiga jenis prediksi (poin demi poin, urutan penuh, dan beberapa urutan).



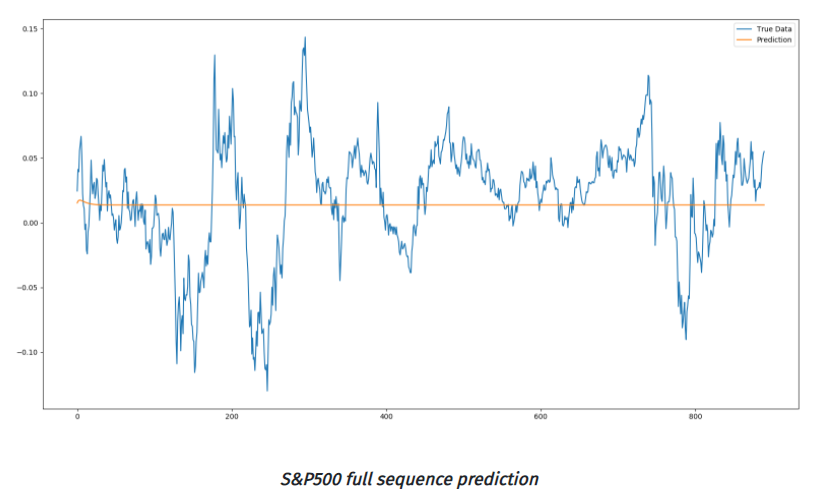


Menjalankan data pada satu prediksi poin demi poin seperti yang disebutkan di atas memberikan sesuatu yang cocok dengan hasil yang cukup dekat. Tapi sedikit seperti menipu. Setelah pemeriksaan lebih dekat, garis prediksi terdiri dari titik prediksi tunggal yang memiliki jendela riwayat nyata sebelumnya di belakangnya. Karena itu, jaringan tidak perlu mengetahui banyak tentang deret waktu itu sendiri selain itu setiap poin berikutnya kemungkinan besar tidak akan terlalu jauh dari poin terakhir. Jadi, meskipun prediksi untuk poinnya salah, prediksi berikutnya akan mempertimbangkan riwayat sebenarnya dan mengabaikan prediksi yang salah, lagi-lagi memungkinkan terjadinya kesalahan.

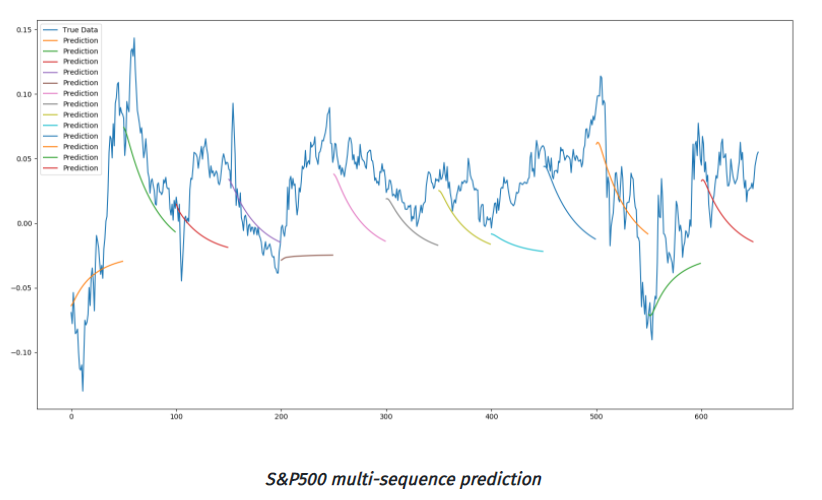
Informasi ini dapat digunakan dalam aplikasi seperti peramalan volatilitas (dapat memprediksi periode volatilitas tinggi atau rendah di pasar dapat sangat menguntungkan untuk strategi perdagangan tertentu), atau menjauh dari perdagangan, ini juga dapat digunakan sebagai indikator yang baik untuk deteksi anomali. Deteksi anomali dapat dicapai dengan memprediksi titik berikutnya, kemudian membandingkannya dengan data sebenarnya saat data tersebut masuk, dan jika nilai data sebenarnya berbeda secara signifikan dengan titik yang diprediksi, tanda anomali dapat dinaikkan untuk titik data tersebut.



Beralih ke prediksi urutan lengkap, tampaknya prediksi ini terbukti paling tidak berguna untuk jenis rangkaian waktu ini (setidaknya dilatih pada model ini dengan hyperparameter ini). Kita dapat melihat sedikit peningkatan pada awal prediksi di mana model mengikuti momentum, namun dengan sangat cepat kita dapat melihat model memutuskan bahwa pola yang paling optimal adalah menyatu ke beberapa ekuilibrium deret waktu. Pada tahap ini, mungkin tampak seperti tidak menawarkan banyak nilai, namun pedagang pengembalian berarti mungkin melangkah ke sana untuk menyatakan bahwa model tersebut hanya menemukan arti bahwa rangkaian harga akan kembali ke saat volatilitas dihapus.



Terakhir dibuat jenis prediksi ketiga untuk model ini, sesuatu yang disebut prediksi multi-urutan adalah campuran dari prediksi urutan lengkap dalam arti masih menginisialisasi jendela pengujian dengan data pengujian, memprediksi poin selanjutnya dan membuat jendela baru dengan poin berikutnya. Namun, setelah mencapai titik di mana jendela masukan terdiri sepenuhnya dari prediksi masa lalu, ia akanberhenti, menggeser maju satu jendela penuh, menyetel ulang jendela dengan data pengujian yang sebenarnya, dan memulai proses lagi. Intinya, ini memberikan beberapa prediksi seperti garis tren pada data pengujian untuk dapat menganalisis seberapa baik model tersebut dapat mengambil tren momentum masa depan.

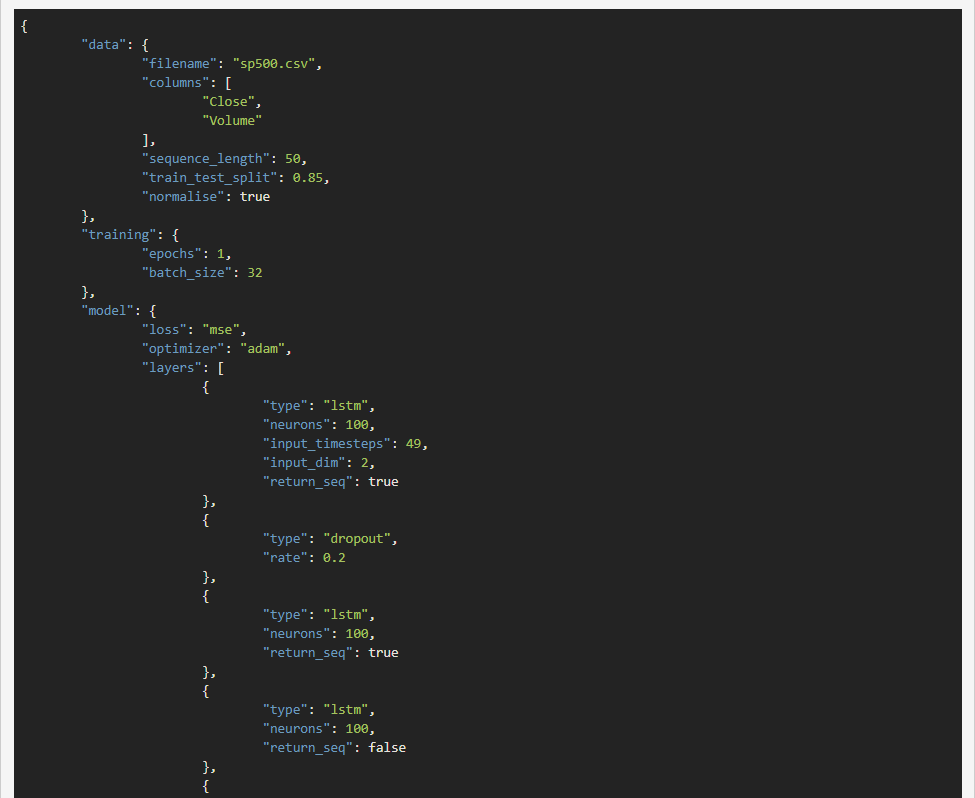


Dari gambar diatas dapat dilihat dari prediksi multi-urutan bahwa jaringan tampaknya benar-benar memprediksi tren (dan amplitudo tren) untuk sebagian besar deret waktu. Meskipun tidak sempurna, tetapi memberikan indikasi kegunaan jaringan saraf dalam LSTM dalam masalah deret waktu dan deret waktu. Akurasi yang lebih besar pasti bisa dicapai dengan penyetelan hyperparameter yang cermat.

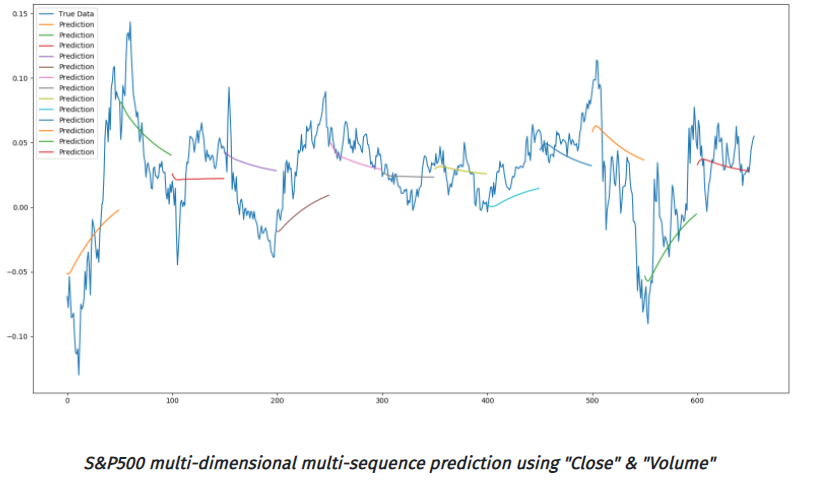
1. PREDIKSI LSTM MULTIDIMENSI

Sejauh ini model yang digunakan input satu dimensi (harga "Tutup" dalam kasus set data S & P500 kami). Tetapi dengan kumpulan data yang lebih kompleks, secara alami terdapat banyak dimensi berbeda untuk urutan yang dapat digunakan untuk meningkatkan kumpulan data dan karenanya meningkatkan akurasi model.

Dalam kasus kumpulan data S & P500, dapat dilihat bahwa kami memiliki Buka, Tinggi, Rendah, Tutup, dan Volume yang menyusun lima kemungkinan dimensi. Kerangka kerja yang telah dikembangkan memungkinkan set data masukan multi-dimensi untuk digunakan, jadi yang perlu dilakukan untuk memanfaatkan mengedit kolom dan nilai input dilapisan pertama lstm dengan tepat untuk menjalankan model. Dalam hal ini akan menjalankan model menggunakan dua dimensi; "Tutup" dan "Volume".







Pada gambar diatas dapat dilihat dengan dimensi "Volume" kedua yang ditambahkan disamping "Tutup" bahwa prediksi keluaran menjadi lebih terperinci. Garis tren prediktor tampaknya lebih akurat untuk memprediksi sedikit penurunan yang akan datang, tidak hanya tren yang berlaku dari awal dan akurasi garis tren juga tampaknya meningkat dalam kasus khusus ini.

1. KESIMPULAN

Pada laporan ini bertujuan untuk memberikan contoh kerja jaringan saraf dalam LSTM dalam praktiknya, artikel ini hanya menggambarkan permukaan potensi dan aplikasinya dalam masalah sekuensial dan temporal. LSTM telah berhasil digunakan dalam banyak masalah dunia nyata dari masalah deret waktu klasik seperti yang dijelaskan di sini, koreksi otomatis teks, deteksi anomali, dan deteksi penipuan, hingga pengembangan inti dalam teknologi mobil self-driving. .

Saat ini ada beberapa batasan dengan penggunaan LSTM vanilla yang dijelaskan di atas, khususnya dalam penggunaan deret waktu finansial, deret itu sendiri memiliki properti non-stasioner yang sangat sulit untuk dimodelkan (meskipun kemajuan telah dibuat dalam menggunakan metode Bayesian Deep Neural Netwok. Untuk menangani non-stasioneritas deret waktu). Untuk beberapa aplikasi, telah ditemukan bahwa kemajuan yang lebih baru dalam mekanisme berbasis perhatian untuk jaringan saraf telah mengungguli LSTM (LSTM yang digabungkan dengan memperhatikan mekanismenya, dapat mengungguli keduanya). Namun sampai sekarang, LSTM memberikan kemajuan yang signifikan pada pendekatan deret waktu statistik yang lebih klasik untuk dapat memodelkan hubungan secara non-linier dan mampu memproses data dengan beberapa dimensi secara non-linier.