

**TUGAS BESAR 2 IF3270 PEMBELAJARAN MESIN
CONVOLUTION NEURAL NETWORK DAN RECURRENT
NEURAL NETWORK**



Kelompok 3:

Maria Flora Renata S.	13522010
M. Athaullah Daffa Kusuma M.	13522044
Christopher Brian	13522106

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

DESKRIPSI PERSOALAN

Convolution neural network (CNN) adalah sebuah jenis arsitektur machine learning yang bertujuan untuk memproses data yang memiliki struktur grid (contoh : gambar dua dimensi). CNN terdiri dari beberapa jenis layer utama seperti convolution layer, activation layer, pooling layer, dan fully connected layer. Sama seperti neural network lainnya, CNN bekerja dalam dua tahap besar, yaitu forward propagation dan backward propagation. Forward propagation adalah tahap dimana data input diproses melalui operasi konvolusi, aktivasi, dan pooling untuk mengambil fitur penting dari data tersebut, hasil akhirnya dilewatkan ke fully connected layer untuk menghasilkan prediksi akhir. Backward propagation adalah tahap dimana parameter-parameter sebelumnya di-*train* berdasarkan error yang diperoleh dari hasil forward propagation sebelumnya.

Selain CNN, disini kita juga membahas tentang RNN. Recurrent Neural Network (RNN) adalah salah satu jenis arsitektur neural network yang dirancang untuk menangani data berurutan, seperti teks, data terkait waktu, dll. Berbeda dari jaringan feedforward biasa, RNN memiliki hubungan balik yang memungkinkan informasi dari timestamp sebelumnya 'diingat' dan digunakan untuk timestamp berikutnya. RNN bekerja dalam dua tahap, yaitu forward propagation dan backward propagation through time (BPTT). Forward propagation dilakukan secara sekuensial, setiap timestamp, hidden state pada timestamp sebelumnya digunakan untuk menghasilkan output dari timestamp sekarang serta menghasilkan hidden state yang baru. Backward propagation through time (BPTT) merupakan backward propagation yang dilakukan berproses mundur sepanjang timestamp, dimana gradien dihitung terhadap semua parameter jaringan dengan memperhatikan kontribusi setiap timestamp terhadap error total. RNN mempunyai satu masalah besar, yaitu exploding/vanishing gradient problem, dimana semakin besar timestamp nya, semakin besar/kecil gradient yang digunakannya. Masalah ini dapat diminimalisir dengan menggunakan Long Short-Term Memory (LSTM).

Long Short-Term Memory (LSTM) merupakan varian RNN yang dirancang untuk mengingat informasi dari timestamp sebelumnya dalam jangka panjang. LSTM menggunakan cell state yang menyimpan memori jangka panjang. Selain itu, LSTM mempunyai tiga gerbang utama, yaitu Forget gate

yang memutuskan informasi mana yang akan dilupakan dari cell state, Input gate yang menentukan informasi baru apa yang akan dimasukkan ke dalam cell state, dan output gate yang mengontrol bagian mana dari cell state yang akan digunakan untuk menghasilkan output saat ini.

Dalam persoalan kali ini, diminta CNN dapat menerima input berupa dataset CIFAR-10 yang terdiri atas gambar yang merepresentasikan 1 dari 10 objek. Model minimal memiliki jenis layer konvolusi 2 dimensi, layer pooling, layer flatten/global pooling, dan layer dense. Loss function yang digunakan adalah sparse categorical crossentropy. Optimizer yang digunakan adalah Adam. Model harus dapat menyimpan bobot hasil pelatihan menggunakan keras, dan mengolahnya kembali untuk menjalankan forward propagation dari scratch (tanpa menggunakan library machine learning).

Untuk RNN dan LSTM, diminta model bertujuan untuk text classification dengan dataset NusaX-Sentiment (Bahasa Indonesia) yang terdiri dari kalimat yang panjangnya tidak menentu. Model minimal memiliki Embedding layer, Bidirectional LSTM/RNN layer, Dropout layer, dan Dense layer. Loss function yang digunakan adalah Sparse Categorical Crossentropy dan optimizer yang digunakan adalah Adam. Model harus dapat menyimpan bobot hasil pelatihan menggunakan keras, dan mengolahnya kembali untuk menjalankan forward propagation dari scratch.

PEMBAHASAN

Penjelasan Implementasi

Untuk CNN, pertama-tama, dilakukan training model CNN pada dataset CIFAR-10 menggunakan Keras. Dataset di preprocess dengan train-test split menjadi 80 persen gambar untuk training dan 20 persen gambar untuk validasi. Kemudian, dibangun model CNN dengan arsitektur layer konvolusi 2D -> max pooling -> konvolusi 2D -> max pooling -> flatten -> dense -> dense. Bobot hasil training model menggunakan keras kemudian disimpan dalam sebuah file.

File tersebut kemudian digunakan untuk inisialisasi bobot pada proses forward propagation dari scratch. Forward propagation kemudian memproses dataset dari setiap layer ke layer berikutnya. Layer konvolusi 2D melakukan konvolusi dari dataset menjadi feature map berdasarkan bobot dan bias dari file tadi. Lalu, deteksi dilakukan dengan fungsi aktivasi ReLU, dan pooling digunakan dengan memilih angka terbesar pada setiap subset input. Layer flatten kemudian merubah bentuk input 3D menjadi array 1D. Deteksi tahap terakhir digunakan dengan fungsi aktivasi softmax yang berfungsi penentu akhir prediksi kelas input dataset.

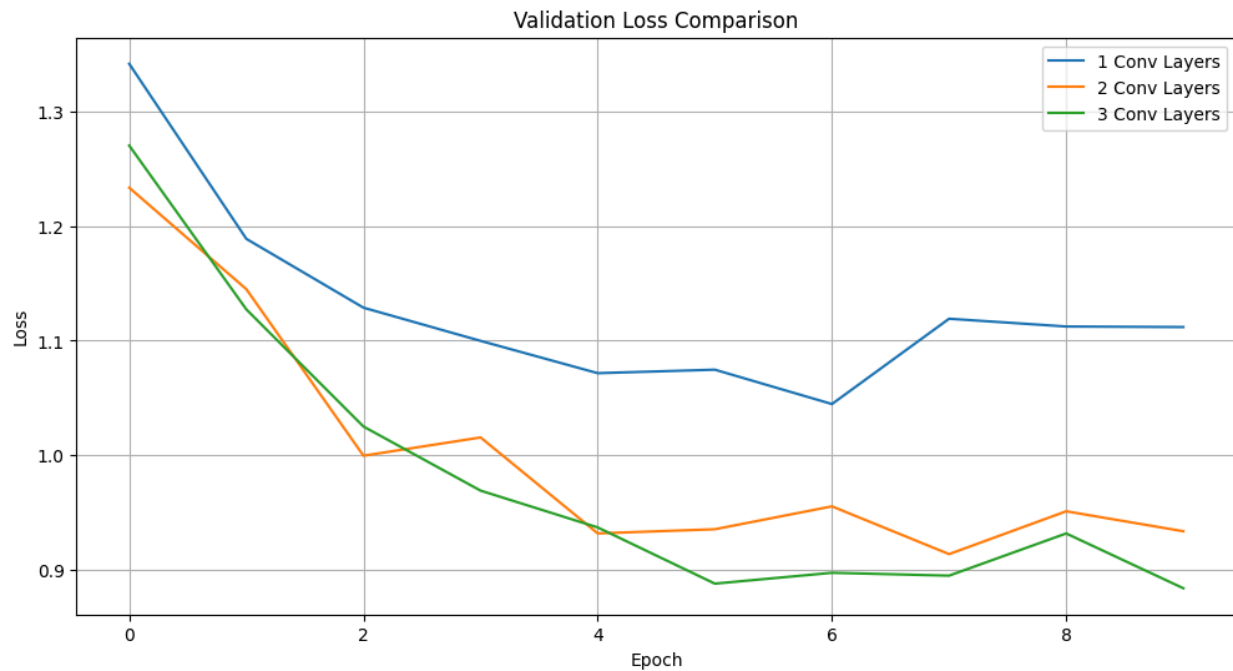
Untuk RNN, dilakukan training model RNN pada dataset NusaX-Sentiment. Karena dataset sudah di-split, preprocessing hanya dilakukan dengan melakukan label encoding (agar dapat menggunakan loss function sparse categorical crossentropy) dan melakukan tokenization menggunakan layer tokenization. Kemudian, dibangun model dengan struktur dasar layer tokenization, embedding, simpleRNN, dropout, dan dense untuk output. Selanjutnya dilakukan variasi pada depth, width, dan direction dari layer simpleRNN.

Untuk LSTM, dibuat sebuah kelas yang menyimpan model kerasnya, dan juga weight dari model keras. Cara mengekstraksi bobotnya yaitu dengan menggunakan `layer.get_weights` dimana layer merupakan iterasi dari `model.layer` dari keras tersebut. Setelah mengekstraksi seluruh bobot dari seluruh layer, baru bisa dibuat sebuah fungsi forward untuk menghitung cell state dan hidden state yang baru, setelah itu baru dilakukan backward untuk trainingnya.

Hasil pengujian

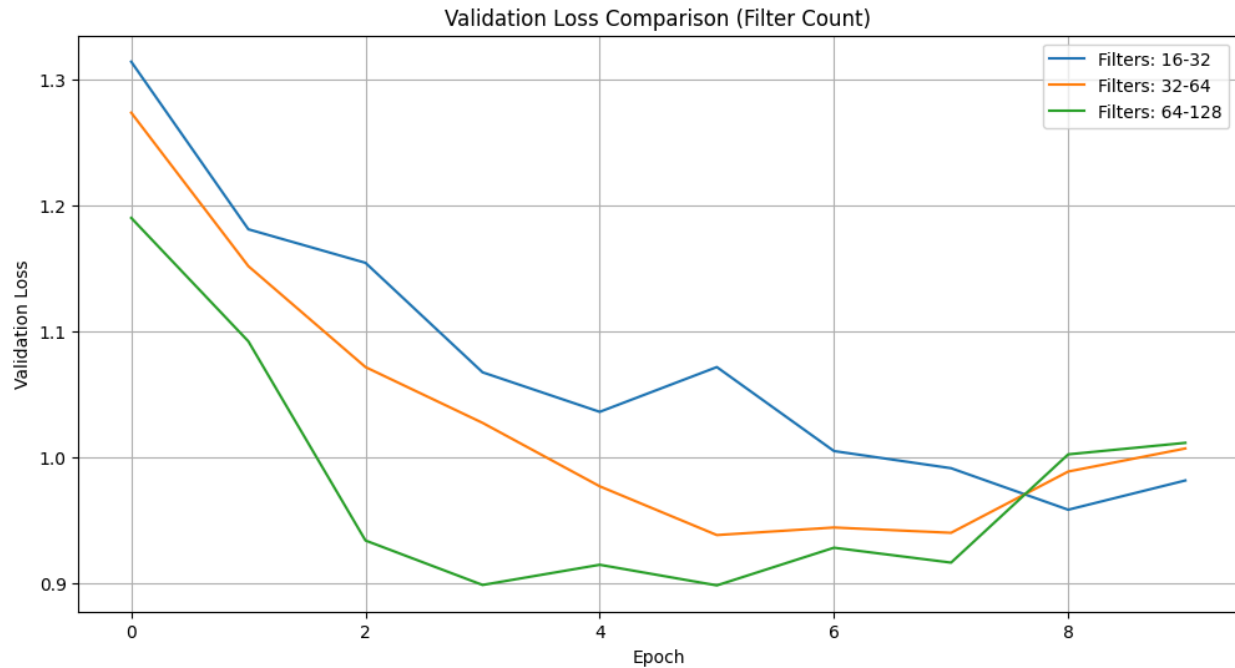
CNN

Pengaruh jumlah layer konvolusi



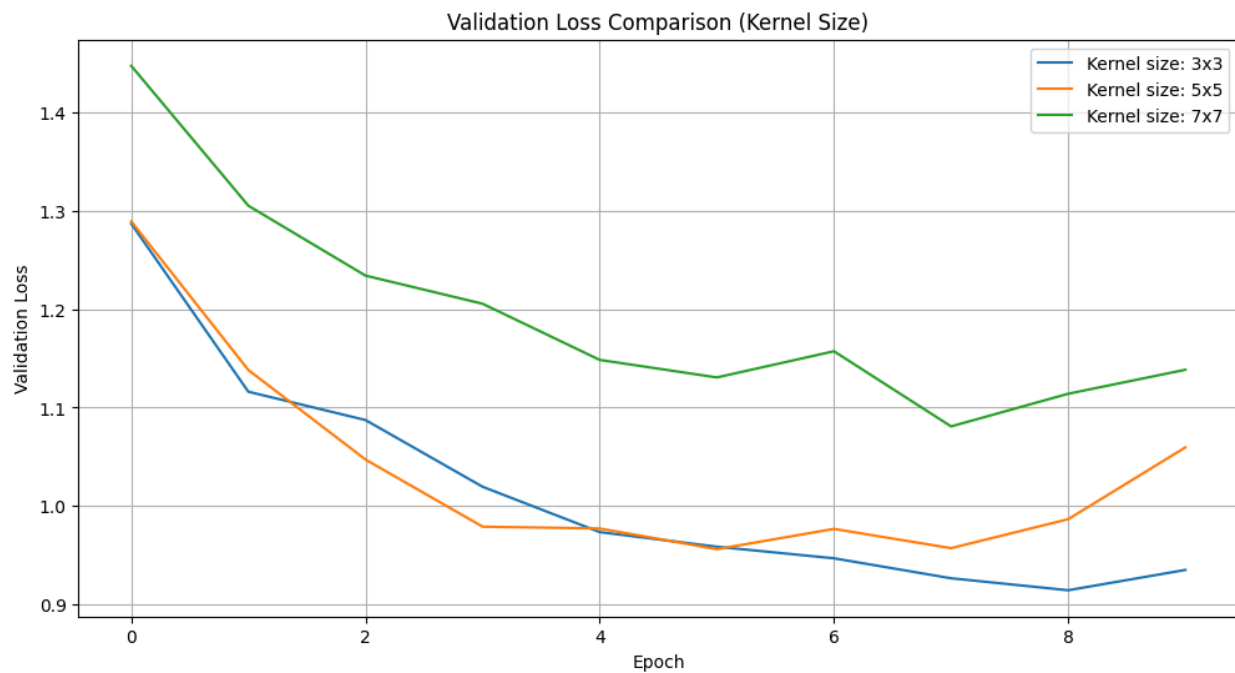
Dilakukan pengujian terhadap 3 konfigurasi model, masing-masing dengan 1, 2, dan 3 jumlah layer konvolusi. Terlihat bahwa validation loss lebih rendah pada model dengan jumlah konvolusi layer lebih besar, mengindikasikan jumlah layer konvolusi berkorelasi positif dengan kinerja model.

Pengaruh banyak filter per layer konvolusi



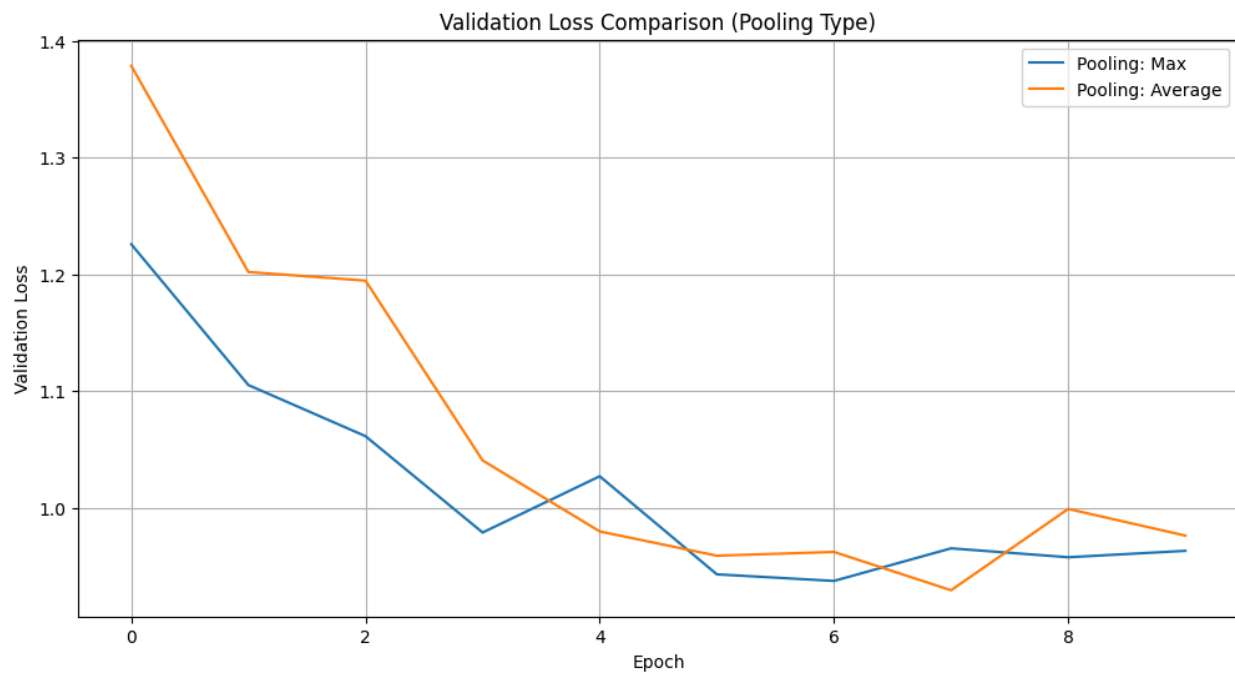
Dilakukan pengujian terhadap 3 konfigurasi model dengan jumlah filter per layer konvolusi masing-masing 16-32, 32-64, dan 64-128. Terlihat bahwa validation loss lebih rendah pada model dengan jumlah filter per layer konvolusi lebih besar. Hal ini mengindikasikan jumlah filter per layer konvolusi berkorelasi positif dengan kinerja model.

Pengaruh ukuran filter per layer konvolusi



Dilakukan pengujian terhadap 3 konfigurasi model, masing-masing dengan ukuran filter/kernel per layer konvolusi 3 kali 3, 5 kali 5, dan 7 kali 7. Terlihat bahwa validation loss lebih rendah pada model dengan ukuran filter per layer konvolusi lebih rendah. Hal ini mengindikasikan ukuran filter per layer konvolusi berkorelasi negatif dengan kinerja model.

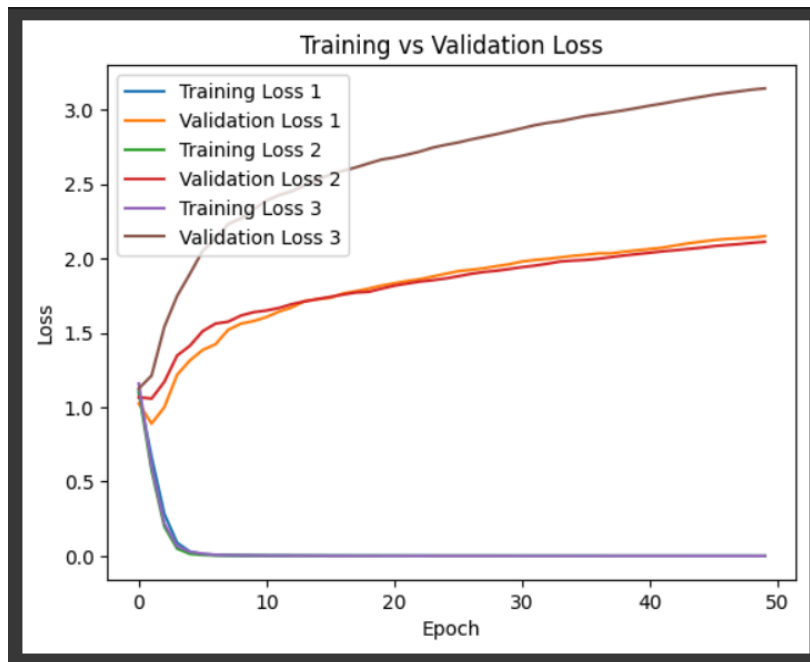
Pengaruh jenis pooling layer



Dilakukan pengujian pada 2 konfigurasi model, masing-masing dengan jenis pooling max dan average. Terlihat bahwa pooling max memiliki validation loss yang lebih rendah pada epoch awal, tetapi pada epoch-epoch berikutnya validation loss menjadi relatif sama dengan model yang menggunakan pooling average. Hal ini mengindikasikan jenis pooling tidak berpengaruh signifikan terhadap kinerja model.

RNN

Pengaruh depth

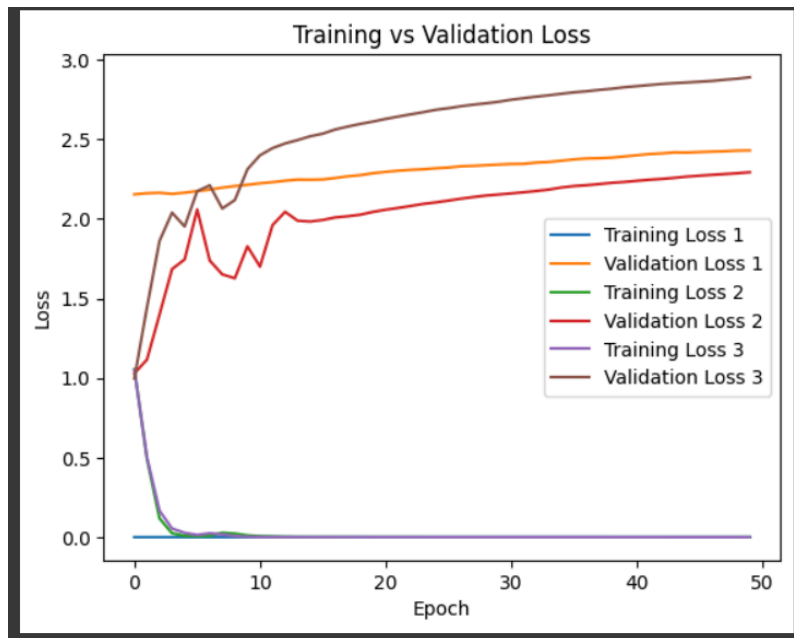


Pada gambar di atas, model 1 menggunakan 2 layer simpleRNN. Model 2 menggunakan 3 layer dan model 3 menggunakan 4 layer.

LSTM

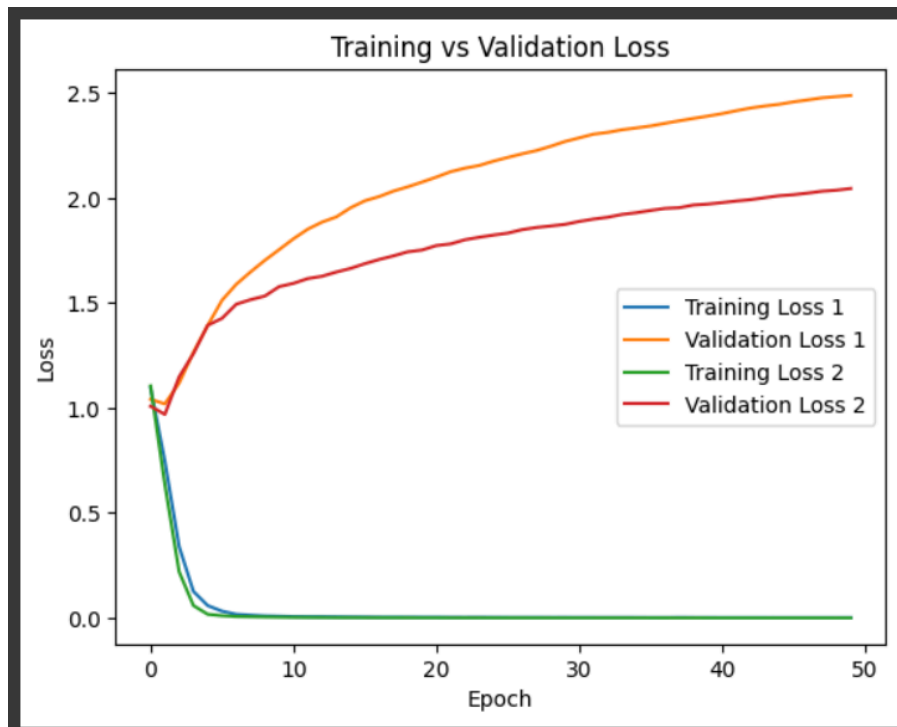
Pengaruh

Pengaruh width



Pada gambar di atas, model 1 menggunakan 2 layer simpleRNN dengan 64 cell masing-masing. Model 2 menggunakan 128 cell masing-masing dan model 3 menggunakan 256 cell masing-masing.

Pengaruh arah



Pada gambar di atas, model 1 menggunakan 2 layer simpleRNN unidirectional. Model 2 menggunakan 2 layer bidirectional.

LSTM

KESIMPULAN DAN SARAN

Kesimpulan

Untuk CNN, didapat bahwa jumlah layer konvolusi dan banyak filter per layer konvolusi berkorelasi positif dengan kinerja layer, sementara itu ukuran filter per layer konvolusi berkorelasi negatif dan jenis pooling layer tidak berpengaruh signifikan.

Saran

Sebaiknya saat mengimplementasikan forward propagation CNN/RNN/LSTM, harus lebih jeli dalam mengambil bobot-bobot yang disediakan oleh model keras. Selain itu, Sebaiknya dialokasikan lebih banyak waktu untuk mengerjakan tugas besar ini, dengan itu tugas besar ini akan selesai dengan sempurna. Sebaiknya juga tidak meremehkan tugas besar sekecil apapun scopenya.

PEMBAGIAN TUGAS

NIM	Tugas
13522010	RNN
13522044	LSTM
13522106	CNN

REFERENSI