

Inception, ResNet and DenseNet

Dicky Ikbal Pratama
Fakultas Matematika dan Ilmu Pengatahuan Alam
Prodi Sistem Informasi
Universitas Hasanuddin

CONTENTS

I	Introduction	1
II	CNN architectures	1
II-A	Inception	1
II-B	ResNet	1
II-C	DenseNet	2
III	Research Methodology	2
III-A	DataSet	2
III-B	Augmentasi Data	2
IV	Result And Conclusion	3
	References	3

Inception, ResNet and DenseNet

Abstract—Dalam technical report ini, kami akan mengimplementasikan dan membahas varian arsitektur CNN modern. Ada banyak arsitektur yang berbeda telah diusulkan selama beberapa tahun terakhir. Beberapa yang paling berpengaruh, dan masih relevan hingga saat ini, adalah sebagai berikut: arsitektur GoogleNet/Inception (pemenang ILSVRC 2014), ResNet (pemenang ILSVRC 2015), dan DenseNet (pemenang makalah terbaik CVPR 2017). Semuanya adalah model tercanggih saat diusulkan, dan ide inti dari jaringan ini adalah fondasi untuk sebagian besar arsitektur tercanggih saat ini. Oleh karena itu, penting untuk memahami arsitektur ini secara detail dan mempelajari cara mengimplementasikannya.

I. INTRODUCTION

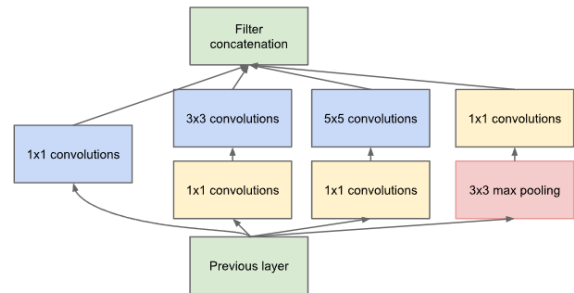
Deep Learning telah menjadi tantangan untuk didefinisikan oleh banyak orang dan para peneliti karena telah berubah bentuk secara perlahan selama dekade terakhir. Salah satu *Architectures Deep Learning* yang paling terkenal yaitu *Convolution Neural Networks* (CNNs), tujuan dari CNN adalah mempelajari fitur tingkat tinggi dalam data melalui konvolusi. CNN sangat cocok untuk pengenalan objek melalui gambar dan memegang kompetisi klasifikasi gambar teratas secara konsisten. CNN dapat mengidentifikasi wajah, individu, rambu jalan, platipus, dan banyak aspek dari data visual. Kemampuan CNN dalam pengenalan gambar adalah salah satu alasan utama mengapa dunia mengakui kehebatan *Deep Learning*. CNN memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan. Kelebihan CNN adalah algoritma yang dapat mengenali informasi tersembunyi dari suatu objek baik itu gambar, suara, teks dan sebagainya walaupun objek tersebut diposisikan dimana saja dalam input. Terdapat beberapa model arsitektur dari CNN salah satunya adalah Inception/GoogleNet, ResNet, dan DenseNet. Pada technical report ini akan dilakukan perbandingan performa dari ketiga model tersebut dengan menggunakan dataset CIFAR-10.

II. CNN ARCHITECTURES

A. Inception

GoogleNet, yang diusulkan pada tahun 2014, memenangkan Tantangan ImageNet karena penggunaan modul Inception. Blok Inception menerapkan empat blok konvolusi secara terpisah pada peta fitur yang sama: konvolusi 1x1, 3x3, dan 5x5, dan operasi kumpulan maks. Ini memungkinkan jaringan untuk melihat data yang sama dengan bidang reseptif yang berbeda. Konvolusi 1x1 tambahan sebelum konvolusi 3x3 dan 5x5 digunakan untuk reduksi dimensi. Ini sangat penting karena peta fitur dari semua cabang digabungkan setelahnya,

dan kami tidak ingin ada ledakan ukuran fitur. Karena konvolusi 5x5 25 kali lebih mahal daripada konvolusi 1x1, kami dapat menghemat banyak komputasi dan parameter dengan mengurangi dimensi sebelum konvolusi besar GoogleNet asli

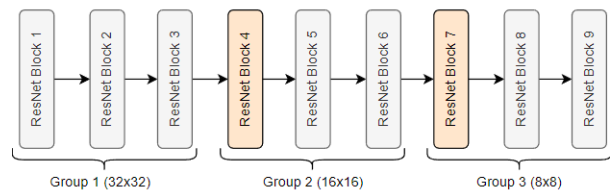


dirancang untuk ukuran gambar ImageNet (224x224 piksel) dan memiliki hampir 7 juta parameter. Saat kami berlatih di CIFAR10 dengan ukuran gambar 32x32. Jumlah saluran untuk pengurangan dimensi dan keluaran per filter (1x1, 3x3, 5x5, dan pooling maks) perlu ditentukan secara manual dan dapat diubah jika tertarik. Intuisi umumnya adalah memiliki filter terbanyak untuk konvolusi 3x3, karena filter tersebut cukup kuat untuk mempertimbangkan konteks sementara membutuhkan hampir sepertiga dari parameter konvolusi 5x5.

B. ResNet

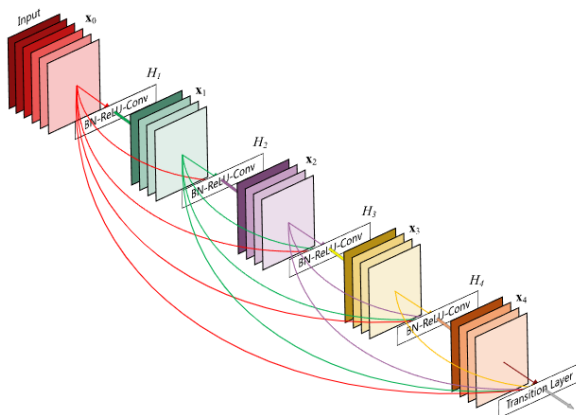
ResNet adalah salah satu makalah AI yang paling banyak dikutip, dan telah menjadi dasar untuk jaringan saraf dengan lebih dari 1.000 lapisan. Blok ResNet asli menerapkan fungsi aktivasi non-linear, biasanya ReLU, setelah koneksi lewati. Sebaliknya, blok ResNet pra-aktivasi menerapkan non-linearitas di awal F. Keduanya memiliki kelebihan dan kekurangan masing-masing. Namun, untuk jaringan yang sangat dalam, pra-aktivasi ResNet telah menunjukkan kinerja yang lebih baik. Mulai dengan blok ResNet asli. Visualisasi di atas sudah menunjukkan lapisan apa saja yang termasuk dalam F. Satu kasus khusus yang harus kita tangani adalah ketika kita ingin mengurangi dimensi gambar dalam hal lebar dan tinggi. Implementasi asli menggunakan pemetaan identitas dengan stride 2 dan menambahkan dimensi fitur tambahan dengan 0. Namun, implementasi yang lebih umum adalah menggunakan konvolusi 1x1 dengan stride 2 karena memungkinkan kita untuk mengubah dimensi fitur sekaligus efisien dalam parameter dan biaya komputasi. Arsitektur ResNet keseluruhan terdiri dari penumpukan beberapa blok ResNet, di mana beberapa di antaranya melakukan downsampling input. Biasanya dikelompokkan berdasarkan bentuk keluaran yang sama. Oleh karena itu, jika kita mengatakan ResNet memiliki [3,3,3] blok, itu

berarti bahwa kita memiliki 3 kali kelompok 3 blok ResNet, di mana subsampling terjadi di blok keempat dan ketujuh. ResNet dengan [3,3,3] blok di CIFAR10 divisualisasikan di bawah ini.



C. DenseNet

Densnet adalah arsitektur lain untuk mengaktifkan jaringan saraf yang sangat dalam dan mengambil perspektif yang sedikit berbeda tentang koneksi residual. Alih-alih memodelkan perbedaan antar lapisan, DenseNet menganggap koneksi residual sebagai cara yang memungkinkan untuk menggunakan kembali fitur di seluruh lapisan. Namun, beberapa pola kompleks terdiri dari kombinasi fitur abstrak (misalnya tangan, wajah, dll.), dan fitur tingkat rendah (misalnya tepi, warna dasar, dll.). DenseNet menyediakan cara yang efisien untuk menggunakan kembali fitur dengan membuat setiap konvolusi bergantung pada semua fitur input sebelumnya, tetapi hanya menambahkan sedikit filter ke dalamnya. Alih-alih memodelkan perbedaan antar lapisan, DenseNet menganggap koneksi residual sebagai cara yang memungkinkan untuk menggunakan kembali fitur di seluruh lapisan, menghilangkan kebutuhan apa pun untuk mempelajari peta fitur yang berlebihan. Jika kita masuk lebih dalam ke jaringan, model mempelajari fitur abstrak untuk mengenali pola. Untuk menemukan fitur tingkat rendah ini di lapisan dalam, CNN standar harus belajar menyalin peta fitur tersebut, yang membuang banyak kompleksitas parameter. DenseNet menyediakan cara yang efisien untuk menggunakan kembali fitur dengan membuat setiap konvolusi bergantung pada semua fitur input sebelumnya, tetapi hanya menambahkan sedikit filter ke dalamnya.



Lapisan terakhir, yang disebut lapisan transisi, bertanggung jawab untuk mengurangi dimensi peta fitur dalam tinggi, lebar, dan ukuran saluran. Meskipun secara teknis merusak

identitas backpropagation, hanya ada beberapa di jaringan sehingga tidak banyak mempengaruhi aliran gradien. Membagi implementasi lapisan di DenseNet menjadi tiga bagian: DenseLayer, dan DenseBlock, dan TransitionLayer. Modul DenseLayer mengimplementasikan satu lapisan di dalam blok padat. Ini menerapkan konvolusi 1x1 untuk pengurangan dimensi dengan konvolusi 3x3 berikutnya. Saluran keluaran digabungkan ke aslinya dan dikembalikan. Menerapkan Normalisasi Batch sebagai lapisan pertama dari setiap blok. Memungkinkan aktivasi yang sedikit berbeda untuk fitur yang sama ke lapisan yang berbeda, tergantung pada apa yang dibutuhkan.

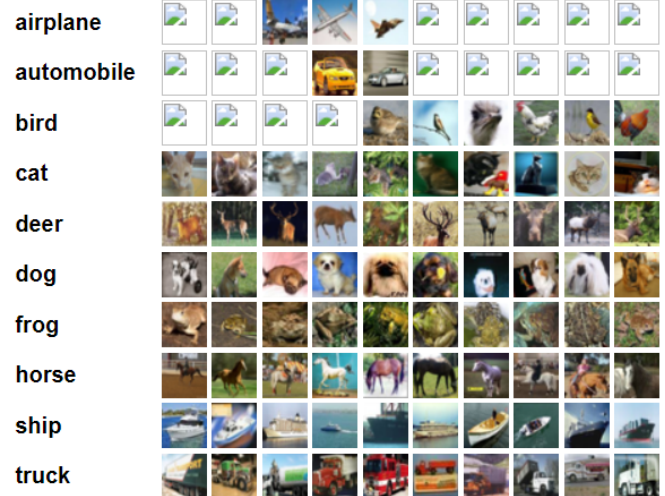
III. RESEARCH METHODOLOGY

A. DataSet

Dataset CIFAR-10 terdiri dari 60.000 gambar berwarna 32x32 dalam 10 kelas, dengan 6000 gambar per kelas. Ada 50000 gambar pelatihan dan 10.000 gambar uji. Mereka dikumpulkan oleh **Alex Krizhevsky, Vinod Nair, dan Geoffrey Hinton**

Dataset dibagi menjadi lima batch pelatihan dan satu batch pengujian, masing-masing dengan 10.000 gambar. Kumpulan tes berisi tepat 1000 gambar yang dipilih secara acak dari setiap kelas. Kumpulan pelatihan berisi gambar yang tersisa dalam urutan acak, tetapi beberapa kumpulan pelatihan mungkin berisi lebih banyak gambar dari satu kelas daripada yang lain. Di antara mereka, kumpulan pelatihan berisi tepat 5000 gambar dari setiap kelas.

Berikut adalah kelas dalam dataset, serta 10 gambar acak dari masing-masing:



B. Augmentasi Data

Augmentasi data adalah seperangkat teknik untuk meningkatkan jumlah data secara artifisial dengan menghasilkan titik data baru dari data yang ada. Ini termasuk

membuat perubahan kecil pada data atau menggunakan model pembelajaran mendalam untuk menghasilkan titik data baru. Teknik augmentasi data seperti crop-ing, padding, dan flipping horizontal umumnya digunakan untuk melatih jaringan neural besar.

IV. RESULT AND CONCLUSION

Model sederhana seperti yang telah diterapkan dalam praktik mencapai kinerja yang jauh lebih rendah, yang selain jumlah parameter yang lebih rendah juga dikaitkan dengan pilihan desain arsitektur. GoogleNet adalah model untuk mendapatkan kinerja terendah pada set validasi dan pengujian, meskipun sangat dekat dengan DenseNet. ResNet mengungguli DenseNet dan GoogleNet lebih dari 1% pada set validasi, sementara ada perbedaan kecil antara kedua versi, asli dan pra-aktivasi.

Secara umum, kita dapat menyimpulkan bahwa ResNet adalah arsitektur yang sederhana namun kuat. Jika kita akan menerapkan model pada tugas yang lebih kompleks dengan gambar yang lebih besar dan lebih banyak lapisan di dalam jaringan, kemungkinan kita akan melihat kesenjangan yang lebih besar antara GoogleNet dan arsitektur skip-connection seperti ResNet dan DenseNet. Perbandingan dengan model yang lebih dalam di CIFAR10 misalnya dapat ditemukan di sini. Menariknya, DenseNet mengungguli ResNet asli pada pengaturan mereka tetapi berada di belakang ResNet Pra-Aktivasi. Model terbaik, Jaringan Jalur Ganda, sebenarnya merupakan kombinasi dari ResNet dan DenseNet yang menunjukkan bahwa keduanya menawarkan keuntungan yang berbeda.

Model	Val Accuracy	Test Accuracy	Num Parameters
GoogleNet	90.40%	89.70%	260,650
ResNet	91.84%	91.06%	272,378
ResNetPreAct	91.80%	91.07%	272,250
Densenet	90.72%	90.23%	239,146

REFERENCES

- [1] suki rahmat, and Budi Nugroho "Pemrograman Deep Learning Dengan Python" Desember 2020.
- [2] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. <https://www.cs.toronto.edu/~kriz/cifar.html>.