

Laporan Tugas UTS Deeplearning, Tutorial 5 : Inception, ResNet and DenseNet

Lopinta Sarungallo
Irene Maritsa
Stella Alithea Sura Parinding
Sistem Informasi - Universitas Hasanuddin

CONTENTS

I	Pendahuluan	1
II	Definisi Permasalahan	1
III	Arsitektur CNN	1
III-A	GoogleNet	1
III-B	ResNet	2
III-C	DenseNet	2
IV	Metodologi Penelitian	2
IV-A	Dataset	2
IV-B	Data Augmentation	2
IV-C	Model	3
V	Analisis dan Inerpretasi	3
VI	Kesimpulan dan Saran	3
	References	3

LIST OF FIGURES

1	Inception Block	1
2	Arsitektur GoogleNet	1
3	Resnet Block	2
4	Arsitektur DenseNet	2

LIST OF TABLES

I	Konfigurasi optimizer dan hyperparameter tiap Arsitektur	3
II	Performa masing-masing model CNN	3

Laporan Tugas UTS Deeplearning, Tutorial 5 : Inception, ResNet and DenseNet

Abstract—Ada banyak arsitektur yang berbeda telah diusulkan selama beberapa tahun terakhir pada arsitektur CNN. Beberapa yang paling berdampak, 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 terbaik saat diusulkan, dan ide inti dari jaringan ini adalah fondasi untuk sebagian besar arsitektur terbaik saat ini. Oleh karena itu, penting untuk memahami arsitektur ini secara detail dan mempelajari cara mengimplementasikannya.

I. PENDAHULUAN

CNN adalah salah satu bidang deep learning yang digunakan terutama untuk pemrosesan gambar. CNN telah mengalami kemajuan yang hebat dalam satu dekade terakhir. Kursus ini mengajarkan CNN mulai dari yang paling mendasar sampai pemrograman CNN yang mutakhir seperti pengenalan mobil pada self-driving car, pengenalan wajah, dan sebagainya.

Sesuai dengan namanya, 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.

II. DEFINISI PERMASALAHAN

Dalam laporan ini, difokuskan untuk membandingkan beberapa arsitektur CNN yakni GoogLeNet, ResNet dan DenseNet pada dataset CIFAR-10. Khususnya untuk mengetahui bagaimana performa dari masing-masing arsitektur CNN pada dataset CIFAR-10, dan arsitektur mana yang memberikan performa terbaik pada dataset CIFAR-10?

III. ARSITEKTUR CNN

CNN memiliki banyak arsitektur yang terus berkembang hingga saat ini. Diantaranya adalah GoogLeNet [1], ResNet[2], dan DenseNet [4]. Dalam report ini, kami akan membandingkan tiga arsitektur CNN yakni GoogLeNet/Inception, ResNet, dan DenseNet received.

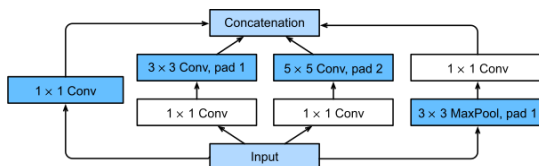


Fig. 1. Inception Block

A. GoogLeNet

Pada tahun 2014, GoogLeNet memenangkan ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14), mengusulkan sebuah struktur yang menggabungkan kelebihan NiN dan paradigma blok berulang. Salah satu fokus dari GoogLeNet adalah mencari ukuran kernel konvolusi terbaik untuk menghindari masalah overfit dan meminimalisir biaya komputasi yang terjadi ketika menambah ukuran Neural Network untuk meningkatkan performa. Cara mendasar untuk menyelesaikan kedua masalah tersebut adalah dengan beralih dari arsitektur yang sepenuhnya terhubung ke arsitektur yang jarang terhubung, bahkan di dalam konvolusi. Arsitektur Inception dipilih setelah dilakukan penyetelan parameter dan berhasil menunjukkan performa yang optimal.

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. Tentu saja, mempelajari konvolusi 5x5 saja secara teoritis akan lebih kuat. Namun, ini tidak hanya lebih banyak komputasi dan memori, tetapi juga cenderung lebih mudah digunakan.

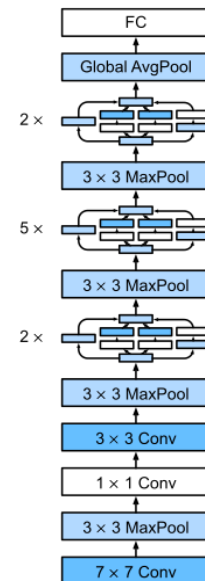


Fig. 2. Arsitektur GoogLeNet

B. ResNet

ResNet merupakan salah satu jenis deep learning yang diperkenalkan pada tahun 2015 oleh Kaiming He, Xiangyu Zhang, Shaoqing Ren dan Jian Sun. Hingga saat ini paper ResNet merupakan salah satu makalah AI yang paling banyak dikutip, dan telah menjadi dasar untuk jaringan saraf dengan lebih dari 1.000 lapisan [3]. Inti dari ResNet yang diusulkan adalah gagasan bahwa setiap lapisan tambahan harus lebih mudah memuat fungsi identitas sebagai salah satu elemennya. Pertimbangan ini agak mendalam tetapi mereka mengarah pada solusi yang sangat sederhana yaitu blok residual. Alih-alih memodelkan $x_l + 1 = F(x_l)$, Dengan memodelkan $x_l + 1 = x_l + F(x_l)$ dimana F adalah sebuah pemetaan non-linier (biasanya urutan modul NN seperti konvolusi, fungsi aktivasi, dan normalisasi).

Bias terhadap matriks identitas menjamin propagasi gradien yang stabil menjadi kurang terpengaruh dengan sendirinya. Ada banyak varian ResNet yang diusulkan, yang sebagian besar menyangkut fungsi, atau operasi yang diterapkan pada penjumlahan. Di sini akan digunakan varian blok ResNet asli dan blok ResNet Pra-Aktivasi.

Pada Fig.3 dapat kita lihat bahwa Blok ResNet asli menerapkan fungsi aktivasi non-linier, biasanya ReLU, setelah koneksi lewat. Sebaliknya, blok ResNet pra-aktivasi menerapkan non-linearitas di awal. Keduanya memiliki kelebihan dan kekurangan masing-masing. Namun, untuk jaringan yang sangat dalam, pra-aktivasi ResNet telah menunjukkan kinerja yang lebih baik karena aliran gradien dijamin memiliki matriks identitas seperti yang dihitung di atas, dan tidak dirugikan oleh aktivasi non-linier yang diterapkan padanya. Sebagai perbandingan, di notebook ini, kami menerapkan kedua jenis ResNet sebagai jaringan dangkal.

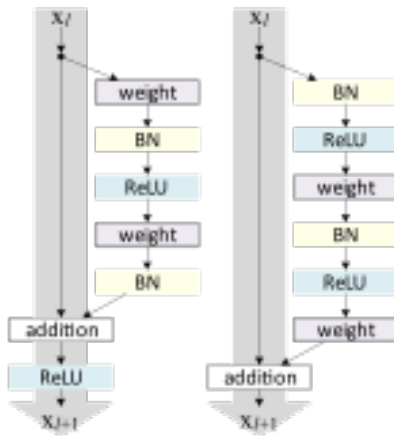


Fig. 3. Resnet Block

C. DenseNet

DenseNet merupakan arsitektur lain yang dapat mengaktifkan jaringan saraf yang sangat dalam dan mengambil perspektif yang sedikit berbeda pada koneksi residual. 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. Namun, beberapa pola kompleks terdiri dari kombinasi fitur abstrak (misalnya tangan, wajah, dll.), dan fitur tingkat rendah (misalnya tepi, warna dasar, dll.). 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.

Sebagai ilustrasinya, dapat dilihat pada Fig.4 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.

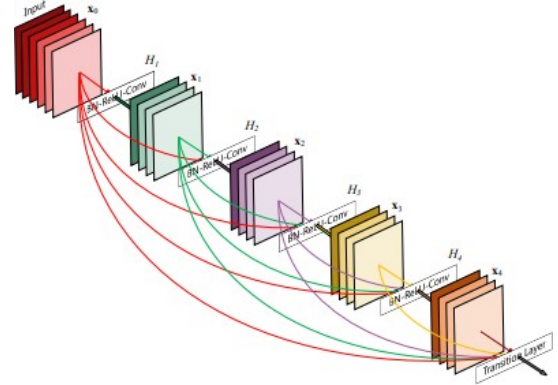


Fig. 4. Arsitektur DenseNet

IV. METODOLOGI PENELITIAN

A. Dataset

Dalam laporan ini dataset yang digunakan adalah dataset CIFAR-10. CIFAR-10 merupakan dataset yang berisi kumpulan gambar yang terdiri dari 60000 gambar yang dibagi menjadi 10 kelas dimana masing-masing kelas terdiri dari 6000 gambar. Masing-masing data merupakan gambar berwarna berukuran 32x32. Dataset ini terdiri dari 50000 data latih dan 10000 data uji. 10 kelas yang terdapat pada dataset CIFAR10 adalah airplane, automobile, cat, deer, dog, frog, horse, ship, truck. Sebelum dataset tersebut digunakan, maka perlu dilakukan normalisasi pada dataset tersebut menggunakan nilai mean dan std dari dataset tersebut.

B. Data Augmentation

Demi mencegah terjadinya overfitting, maka akan dilakukan data augmentation. Pada tahap ini, akan dilakukan 2 proses

Model	Activation	Optimizer	lr	Momentum	Weight Decay
GoogleNet	ReLU	Adam	0.001	-	0.0001
ResNet	ReLU	SGD	0.1	0.9	0.0001
ResNetPreAct	ReLU	SGD	0.1	0.9	0.0001
DenseNet	ReLU	Adam	0.001	-	0.0001

TABLE I. KONFIGURASI OPTIMIZER DAN HYPERPARAMETER TIAP ARSITEKTUR

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

TABLE II. PERFORMA MASING-MASING MODEL CNN

data augmentasi. Pertama, proses data augmentasi dilakukan dengan melakukan flip pada gambar secara horizontal sehingga diperoleh gambar baru dengan perspektif berbeda. Umumnya, proses flip tidak akan mengubah ukuran dari gambar tersebut. Proses yang augmentasi kedua adalah melakukan resize pada gambar. Proses resize dapat mengubah skala dan aspect ratio dari gambar. Oleh karena itu, setelah proses ini, gambar akan dicrop dengan ukuran 32x32 agar selanjutnya dapat dimasukkan kedalam model CNN.

C. Model

Untuk membangun model CNN, kami menggunakan tiga arsitektur CNN yakni GoogleNet, ResNet, dan DenseNet. Untuk arsitektur ResNet, akan digunakan 2 jenis block yakni original ResNet Block dan Pre-Activation ResNet Block. Model GoogleNet pada penelitian ini akan menggunakan activation function yakni ReLU. Model GoogleNet menggunakan optimizer Adam dengan learning rate 0.001 dan weight decay 0.0001. Untuk model ResNet, activation function yang digunakan adalah ReLU dan Stochastic Gradient Descent (SGD). Selain itu, hyperparameter yang digunakan adalah learning rate yang digunakan sebesar 0.1, momentum sebesar 0.9 dan weight decay sebesar 0.0001. Optimizer dan hyperparameter tersebut berlaku untuk 2 jenis ResNet block yang digunakan. Untuk model DenseNet, optimizer yang digunakan adalah Adam dan activation function yang digunakan adalah ReLU. Selain itu, hyperparameter yang digunakan adalah learning rate yang digunakan sebesar 0.001 dan weight decay sebesar 0.0001. Secara sederhana, optimizer dan hyperparameter yang digunakan dapat dilihat pada tabel I.

V. ANALISIS DAN INERPRETASI

Setelah melakukan proses pembangun model dan pengujian model, performa dari masing-masing model dapat dilihat pada Tabel II

Secara umum, dapat dilihat bahwa semua model memberikan performa yang cukup baik. Dari sini, dapat dilihat bahwa GoogleNet mendapatkan kinerja paling rendah di bidang validation accuracy dan test accuracy masing-masing sebesar 90,40 persen dan 89,70 persen. DenseNet memberikan

performa yang sedikit lebih unggul jika dibandingkan dengan GoogleNet di mana terdapat peningkatan sebesar 0,32 persen pada validation accuracy dan 0,53 persen pada test accuracy. Terakhir, kita dapat melihat performa masing-masing model ResNet yang mampu mengalahkan model GoogleNet dan DenseNet dimana mungkin ada perbedaan lebih dari 1 persen pada bagian validation accuracy dan perbedaan lebih dari 1 persen untuk test accuracy dengan GoogleNet dan kontras mendekati 1 persen jika dibandingkan dengan DenseNet.

VI. KESIMPULAN DAN SARAN

Berdasarkan hasil yang didapatkan, kita dapat menyimpulkan bahwa ResNet merupakan arsitektur yang sederhana namun memiliki performa yang lebih baik dibandingkan dengan 2 arsitektur lainnya. Jika kita perlu menerapkan arsitektur untuk task yang lebih kompleks dengan gambaran yang lebih besar, maka kita dapat melihat kontras yang sangat besar antara GoogleNet dan arsitektur skip connection seperti ResNet dan DenseNet. Dengan demikian, jika kita perlu memilih arsitektur untuk task baru, melihat bagaimana kinerja ResNet pada dataset CIFAR10 maka arsitektur ResNet maka arsitektur ResNet dapat menjadi pertimbangan.

REFERENCES

- [1] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). *Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [3] Crew, B. (2020, July 13). *Google scholar reveals its most influential papers for 2020. Nature Index*. Retrieved April 29, 2022, from <https://www.natureindex.com/news-blog/google-scholar-reveals-most-influential-papers-research-citations-twenty-twenty>
- [4] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). *Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).