Modern CNN : Inception, ResNet dan DenseNet

Andi Nur Salsabila Syamsu (H071191015) dan Anugrah Lestari (H071191059)
Program Studi Ilmu Komputer
Universitas Hasanuddin

CONTENTS

Modern CNN : Inception, ResNet dan DenseNet

I. Pendahuluan

Convolutional Neural Network (CNN) adalah salah satu jenis Neural Network yang biasa digunakan pada data image. Arsitektur CNN sekarang ada dimana-mana bidang visi komputer, dan telah menjadi begitu dominan sehingga hampir tidak ada orang saat ini yang akan mengembangkan aplikasi komersial atau mengikuti kompetisi yang terkait dengan pengenalan gambar, deteksi objek, atau segmentasi semantik, tanpa membangun dari pendekatan ini.

CNN cenderung efisien secara komputasi, baik karena memerlukan parameter yang lebih sedikit daripada arsitektur yang terhubung penuh dan karena konvolusi mudah diparalelkan di seluruh inti GPU. Akibatnya, praktisi sering menerapkan CNN bila memungkinkan, dan semakin mereka muncul sebagai pesaing yang kredibel bahkan pada tugas-tugas dengan struktur urutan satu dimensi, seperti audio, teks, dan analisis deret waktu, di mana jaringan saraf berulang secara konvensional digunakan. Beberapa adaptasi CNN yang cerdas juga telah membawa mereka pada data terstruktur grafik dan dalam sistem pemberi rekomendasi.

Arsitektur modern CNN menjadi model dasar di mana banyak proyek penelitian dan sistem yang digunakan dibangun. Masing-masing jaringan ini secara singkat merupakan arsitektur yang dominan dan banyak yang menjadi pemenang atau runner-up dalam kompetisi ImageNet, yang telah menjadi barometer kemajuan pembelajaran yang diawasi dalam visi komputer sejak 2010. Model-model ini termasuk GoogLeNet, yang menggunakan jaringan dengan rangkaian paralel; jaringan sisa (ResNet), yang tetap menjadi arsitektur paling populer dalam visi komputer; dan jaringan yang terhubung secara padat (DenseNet), yang mahal untuk dihitung tetapi telah menetapkan beberapa tolak ukur baru-baru ini.

II. Inception

GoogleNet diusulkan pada tahun 2014 dan memenangkan ImageNet challenge karena menggunakan modul inception. Arsitektur GoogleNet terdiri dari beberapa blok Inception dengan sesekali max pooling untuk mengurangi tinggi dan lebar peta fitur. GoogleNet asli dirancang untuk ukuran gambar ImageNet (224x224 piksel) dan memiliki hampir 7 juta parameter.

Blok Inception menerapkan empat blok konvolusi secara terpisah pada peta fitur yang sama: konvolusi 1x1, 3x3, dan 5x5, dan operasi max pool. 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. Konvolusi 5x5, 25 kali lebih mahal daripada konvolusi 1x1, dapat menghemat banyak komputasi dan parameter dengan mengurangi dimensi sebelum konvolusi besar.

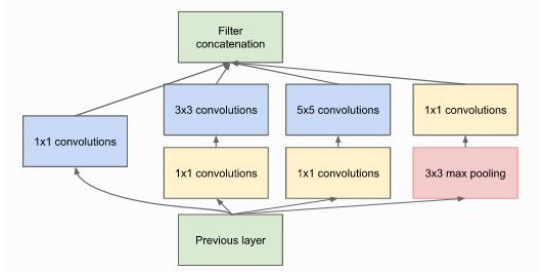

Fig. 1. Inception module with dimension reductions

III. ResNet

Makalah ResNet adalah salah satu makalah AI yang paling banyak dikutip, dan telah menjadi dasar untuk jaringan saraf dengan lebih dari 1.000 lapisan. Terlepas dari kesederhanaannya, ide koneksi residual sangat efektif karena mendukung propagasi gradien yang stabil melalui jaringan. Alih-alih memodelkan $x_{l+1} = F(x_l)$, kita memodelkan $x_{l+1} = x_l + F(x_l)$ dimana F adalah pemetaan non-linier (biasanya urutan modul Neural Network seperti konvolusi, fungsi aktivasi, dan normalisasi). Jika kita melakukan backpropagation pada koneksi residual tersebut, kita memperoleh:

$$\frac{\partial x_{l+1}}{\partial x_l} = \mathbf{I} + \frac{\partial F(x_l)}{\partial x_l}$$

Bias terhadap matriks identitas menjamin propagasi gradien yang stabil menjadi kurang dipengaruhi oleh F itu sendiri. Ada banyak varian ResNet yang diusulkan, yang sebagian besar menyangkut fungsi F, atau operasi yang diterapkan pada penjumlahan. Dua di antaranya: blok ResNet asli, dan blok ResNet Pra-Aktivasi.

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 karena aliran gradien dijamin memiliki matriks identitas seperti yang dihitung di atas, dan tidak dirugikan oleh aktivasi non-linier yang diterapkan padanya.

Blok ResNet asli. Ketika ingin mengurangi dimensi gambar dalam hal lebar dan tinggi. Blok dasar ResNet membutuhkan $F(x_l)$ untuk memiliki bentuk yang sama dengan x_l. Jadi perlu mengubah dimensi x_l sebelum menambahkan ke $F(x_l)$. Implementasi asli menggunakan pemetaan identitas dengan langkah 2 dan menambahkan dimensi fitur tambahan dengan 0. Namun, implementasi yang lebih umum adalah menggunakan konvolusi 1x1 dengan langkah 2 karena memungkinkan kita untuk mengubah dimensi fitur sekaligus efisien dalam parameter dan biaya komputasi.

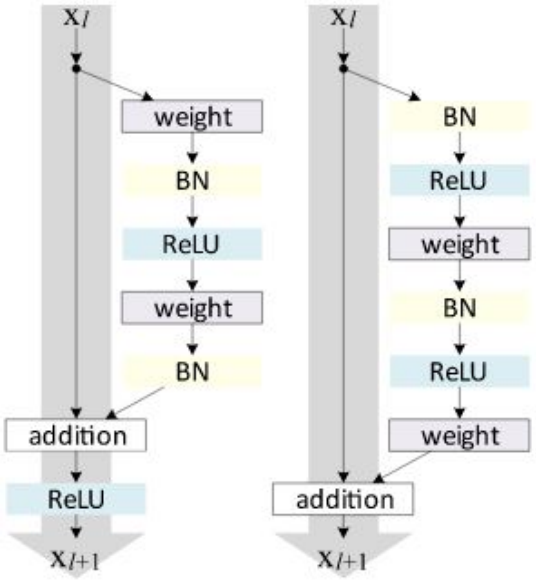

Fig. 2. Resnet

Blok ResNet Pra-Aktivasi. Untuk ini, perlu mengubah urutan layer di self.net dan tidak menerapkan fungsi aktivasi pada output. Selain itu, operasi downsampling harus menerapkan non-linearitas serta input, x_l, belum diproses oleh non-linearitas.

Arsitektur ResNet keseluruhan terdiri dari penumpukan beberapa blok ResNet, dimana beberapa diantaranya melakukan downsampling input. Pengelompokan blok ResNet di seluruh jaringan biasanya berdasarkan bentuk keluaran yang sama. Oleh karena itu, jika ResNet memiliki blok [3,3,3] itu berarti kita memiliki 3 kali kelompok 3 blok ResNet, dimana subsampling terjadi di blok keempat dan ketujuh.

ResNet telah terbukti menghasilkan lebih halus loss surface daripada jaringan tanpa skip connection. Visualisasi yang mungkin dari loss surface with/out skip connection berikut ini :

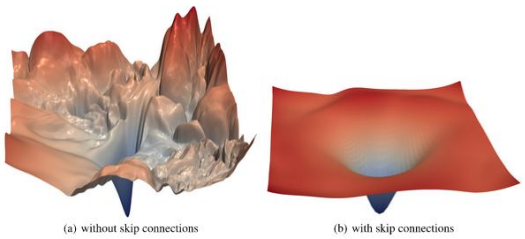

Fig. 3. DenseNet

Sumbu x dan y menunjukkan proyeksi ruang parameter, dan sumbu z menunjukkan nilai kerugian yang dicapai oleh nilai parameter yang berbeda. Pada permukaan halus seperti yang ada disebelah kanan, mungkin tidak memerlukan tingkat pembelajaran adaptif seperti yang disedikan optimasi Adam. Sebaliknya, Adam bisa terjebak di local optima sementara SGD menemukan minima yang lebih luas yang cenderung menggeneralisasi lebih baik.

IV. DenseNet

DenseNet adalah arsitektur lain untuk mengaktifkan kembali neural network yang sangat dalam dan memiliki perspektif yang sedikit berbeda dari koneksi residual. DenseNet menyediakan cara efisien untuk menggunakan kembali fitur dengan membuat setiap konvolusi bergantung pada semua fitur sebelumnya tetapi hanya menambahkan sedikit filter di dalamnya.

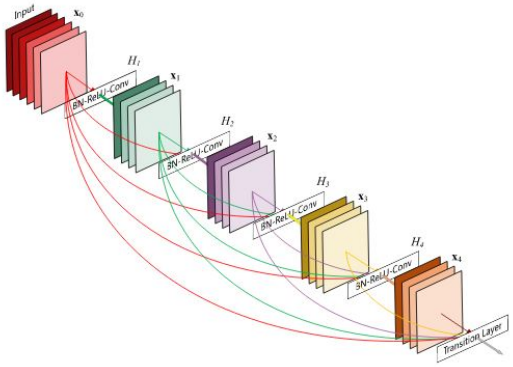

Fig. 4. DenseNet

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.

Implementasi lapisan di DenseNet dibagi menjadi tiga bagian: DenseLayer, dan DenseBlock, dan TransitionLayer. Modul DenseLayer mengimplementasikan satu lapisan di dalam dense block. Diterapkan konvolusi 1x1 untuk pengurangan dimensi dengan konvolusi 3x3 berikutnya. Hasil output digabungkan ke aslinya dan dikembalikan. Perhatikan bahwa kami menerapkan Normalisasi Batch sebagai lapisan pertama dari setiap blok. Ini memungkinkan aktivasi yang sedikit berbeda untuk fitur yang sama ke lapisan yang berbeda, tergantung pada apa yang dibutuhkan.

V. Eksperimen

Varian arsitektur modern CNN yaitu Inception, ResNet dan DenseNet akan diimplimentasikan dan dibandingkan.

A. Dataset

Dataset yang digunakan untuk melatih dan mengevaluasi model adalah CIFAR-10. Dataset CIFAR-10 terdiri dari 60.000 gambar berwarna berukuran 32x32 piksel yang dibagi ke dalam 10 kelas, dengan 6.000 gambar per kelas. Terdapat 50.000 gambar pelatihan dan 10.000 gambar uji. Untuk preprocessing, dilakukan normalisasi dengan mean dan standar deviasi dari dataset CIFAR-10. Selain itu, akan dilakukan augmentasi data selama pelatihan. Hal ini bertujuan mengurangi risiko overfitting dan membantu CNN untuk menggeneralisasi lebih baik.

Secara khusus dilakukan dua augmentasi acak. Pertama, membalik setiap gambar secara horizontal dengan peluang 50% (transforms.RandomHorizontalFlip). Kelas objek biasanya tidak berubah saat membalik gambar. Augmentasi kedua yang digunakan adalah transforms.RandomResizedCrop. Transformasi ini menskalakan gambar dalam rentang kecil, sementara pada akhirnya mengubah rasio aspek, dan memotongnya ke ukuran sebelumnya. Oleh karena itu, nilai piksel aktual berubah, sementara konten atau semantik keseluruhan gambar tetap sama.

Fig. 5. Contoh Augmentasi CIFAR-10

B. PyTorch Lightning

PyTorch Lightning adalah framework yang menyederhanakan kode untuk training, evaluasi, dan testing suatu model di PyTorch. PyTorch juga menangani TensorBoard yaitu sebuah toolkit untuk memvisualisasikan eksperimen Machine Learning dan menyimpan model checkpoints secara otomatis. Pada PyTorch Lightning, didefinisikan pl.LightningModule yang mengorganisasikan kode yang digunakan menjadi 5 bagian utama yaitu, Initialization, Optimizers, Training loop, Validation loop, dan Test loop.

C. Training

Dataset CIFAR10 di training menggunakan tiga model arsitektur modern CNN yaitu Inception, ResNet dan DenseNet.

Inception/GoogleNet. Arsitektur yang digunakan untuk melatih dataset CIFAR10 dengan ukuran gambar 32x32 tidak memerlukan arsitektur yang berat, tetapi menerapkan versi yang diperkecil. Jumlah saluran untuk pengurangan dimensi dan output per filter (1x1, 3x3, 5x5, dan max pooling) ditentukan secara manual. Training model dilakukan oleh PyTorch Lightning dan melatih hampir 200 epochs. Pada tahap ini diterapkan metode optimasi untuk meningkatkan presentase akurasi. Metode optimasi yang digunakan adalah Adaptive Moment Optimization (Adam).

ResNet. Terdapat perbedaan pada pelatihan GoogleNet yaitu metode optimasi yang digunakan bukan Adam melainkan menggunakan SGD dengan momentum sebagai pengoptimal. Adam sering menghasilkan akurasi yang sedikit lebih buruk pada ResNet yang sederhana dan dangkal. Sebagai perbandingan, diterapkan dua jenis ResNet sebagai jaringan dangkal yaitu ResNet asli dan ResNet pra-aktivasi.

Visualisasi ResNet dengan blok [3,3,3] pada dataset CIFAR10 :

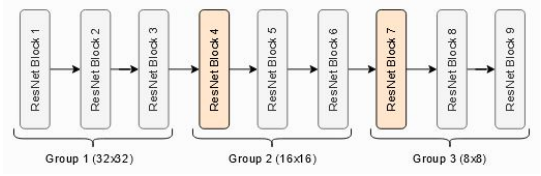

Fig. 6. Inception module with dimension reductions

Tiga grup beroperasi pada resolusi 32×32, 16×16 dan 8×8 masing-masing. Blok berwarna oranye menunjukkan blok ResNet dengan downsampling.

DenseNet. Berbeda dengan ResNet, DenseNet tidak menunjukkan masalah apapun dengan Adam, sehingga akan digunakan Adam untuk pengoptimalnya. Hyperparameter lainnya dipilih untuk menghasilkan jaringan dengan ukuran parameter yang sama seperti ResNet dan GoogleNet. Umumnya, ketika merancang jaringan yang sangat dalam, DenseNet lebih efisien parameter daripada ResNet saat mencapai kinerja yang serupa atau bahkan lebih baik.

D. Hasil dan Pembahasan

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

Semua model berkinerja cukup baik. GoogleNet adalah model yang mendapatkan kinerja terendah pada set validasi dan pengujian, meskipun sangat dekat dengan DenseNet. Pencarian hyperparameter yang tepat untuk semua ukuran saluran di GoogleNet kemungkinan akan meningkatkan akurasi model ke tingkat yang sama, tetapi ini juga mahal mengingat sejumlah besar hyperparameter. ResNet mengungguli DenseNet dan GoogleNet lebih dari 1% pada set validasi, sementara ada perbedaan kecil antara kedua versi, ResNet asli dan pra-aktivasi. Kita dapat menyimpulkan bahwa untuk jaringan dangkal, tempat fungsi aktivasi tampaknya tidak terlalu penting.

VI. KESIMPULAN

Secara umum, dapat disimpulkan bahwa ResNet adalah arsitektur yang sederhana namun kuat. Jika diterapkan model yang lebih kompleks dengan gambar yang lebih besar dan lebih banyak lapisan di dalam jaringan, mungkin dapat dilihat kesenjangan yang lebih besar antara GoogleNet dan arsitektur skip-connection seperti ResNet dan DenseNet.