Inception, ResNet dan DenseNet

Rahmi, Eka Fitri Ramadani, Alika Oktaviani
Departmen Matematika
Universitas Hasanuddin

CONTENTS

I Introduction 1

II Problem Definition 1

III Proposed Solutions 1
III-A Inception . . . 1
III-B ResNet . . . 1
III-C DenseNet . . . 1

IV Research Methodology 1

V Analysis and Interpretation 2

VI Conclusions and Recommendations 2

References 2

LIST OF FIGURES

1 perbandingan akurasi model . . . 2

Inception, ResNet dan DenseNet

***Abstract*—Pada laporan ini membahas tentang perbandingan arsitektur Convolutional Neural Network (CNN) untuk klasifikasi dataset CIFAR10. Arsitektur CNN yang diujicobakan yaitu, Inception/GoogleNet, ResNet, DenseNet. Hasil uji coba menunjukkan kinerja terbaik ditunjukkan oleh arsitektur ResNet dengan akurasi validasi sebesar 91.84 dan akurasi test sebesar 91.06 dengan Num parameter 272.378.**

I. Introduction

Convolutional neural network (CNN, atau ConvNet) adalah bagian dari artificial neural network (ANN), yang paling sering diterapkan untuk menganalisis citra visual. CNN memiliki beberapa jenis arsitektur diantaranya Inception, ResNet dan DenseNet yang merupakan pengembangan dari ResNet. Laporan ini membahas pendekatan arsitektur CNN, yaitu Inception, ResNet dan DenseNet untuk membangun model. Dataset yang digunakan yaitu CIFAR10 yang dibagi menjadi data training untuk melatih model dan data validasi untuk memeriksa keakuratan model dan melakukan perbandingan akurasi. Dengan demikian, laporan ini akan membahas pendekatan secara rinci dan hasil dari perbandingan akurasi ketiga arsitektur CNN tersebut.

II. Problem Definition

Dataset yang digunakan adalah CIFAR10. Data ini terdiri dari 60000 gambar berwarna 32x32 dalam 10 kelas, dengan 6000 gambar per kelas yang terbagi menjadi 50.000 data train dan 10.000 data validasi. Dimana pada data ini akan dilakukan klasifikasi.

III. Proposed Solutions

Untuk melakukan klasifikasi pada dataset CIFAR10, digunakan arsitektur CNN yaitu Inception, ResNet dan DenseNet.

A. Inception

Inception merupakan pengembangan dari Convolutional Neural Network (CNN) yang pertama kali diperkenalkan oleh Szegedy, dkk., pada tahun 2014 dalam paper berjudul "Going Deeper with Convolutions". Very deep convolutional networks telah menjadi pusat pengembangan dalam performa image recognition belakangan ini. Contohnya adalah arsitektur Inception yang menghasilkan performa yang sangat baik dengan komputasi yang relatif rendah.

Blok Inception ini menerapkan empat blok konvolusi secara terpisah pada satu blok yang sama: konvolusi 1x1, 3x3, 5x5, dan operasi max poolin. Saat melakukan pelatihan dengan data CIFAR10 dengan ukuran gambar 32x32, tidak diperlukan arsitektur yang berat. Jumlah channel untuk reduksi dimensi dan keluaran per filter (1x1, 3x3, 5x5, dan max pooling) perlu ditentukan secara manual dan dapat diubah. Umumnya filter terbanyak untuk konvolusi adalah 3x3, karena filter tersebut cukup baik mempertimbangkan parameter yang dibutuhkan hanya sekitar sepertiga dari parameter konvolusi 5x5.

Pelatihan model ditangani oleh PyTorch Lightning, dan hanya perlu memanggil fungsi train model untuk memulai pelatihan. Kemudian dilakukan pelatihan model pada hampir 200 epochs, yang memakan waktu sekitar satu jam pada GPU default Lisa (GTX1080Ti).

B. ResNet

ResNet memiliki dua diantaranya blok yaitu 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.

Arsitektur ResNet keseluruhan terdiri dari penumpukan beberapa blok ResNet, di mana beberapa di antaranya melakukan downsampling input. Blok ResNet biasanya dikelompokkan dari 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 dengan tiga kelompok yang berpoerasi yaitu group 1 (32 X 32) group 2 (16 X 16) dan group 3 (8 X 8).

C. DenseNet

Layer di DenseNet dibagi menjadi tiga bagian: DenseLayer, DenseBlock, dan TransitionLayer. Modul DenseLayer mengimplementasikan satu layer di dalam dense block. Ini menerapkan konvolusi 1x1 untuk pengurangan dimensi dengan konvolusi 3x3 berikutnya. Selanjutnya Batch Normalization diterapkan sebagai layer pertama dari setiap blok. Ini memungkinkan aktivasi yang sedikit berbeda untuk fitur yang sama ke layer yang berbeda, tergantung pada apa yang dibutuhkan. Setiap dense layer mengambil input asli sebagai input yang digabungkan dengan semua peta fitur lapisan sebelumnya. Kemudian TransitionLayer mengambil output akhir dari dense block sebagai input dan mengurangi dimensi channelnya menggunakan konvolusi 1x1.

Untuk mengurangi dimensi tinggi dan lebar, diambil pendekatan yang berbeda dari di ResNet yaitu menerapkan average pooling (penyatuan rata-rata) dengan ukuran kernel 2 dan stride 2. Parameter ini lebih efisien daripada menggunakan konvolusi 3x3 dengan stride 2. Untuk menentukan jumlah layer, digunakan notasi yang sama seperti di ResNet. Serta diterapkan layer transisi untuk mengurangi dimensi sebesar 2 kecuali untuk dense block yang terakhir. Setelah itu, melatih networknya dengan optimizer.

IV. Research Methodology

Dataset yang akan dilatih dan dievaluasi modelnya adalah dataset CIFAR10. Sebelum datanya dilatih dan dievaluasi, dilakukan normalisasi dengan cara menghitung nilai mean dan standar deviasi dari dataset CIFAR10 untuk setiap channel red, green dan blue. Selain itu, dilakukan pula augmentasi data selama pelatihan/training yang berfungsi untuk mengurangi risiko eoverfitting dan membantu CNN dalam melakukan generalisasi. Disini diterapkan dua augmentasi acak.

Augmentasi pertama, yaitu membalik gambar secara horizontal dengan peluang 50 persen (transforms.RandomHorizontalFlip) tidak mengubah objeknya. Augmentasi kedua, menggunakan (transforms.RandomResizedCrop) yang berfungsi untuk menskalakan gambar dalam rentang kecil, mengubah rasio, dan memotongnya dalam ukuran sebelumnya. Maka nilai piksel aslinya berubah sementara konten keseluruhan gambarnya tetap sama.

Selanjutnya, kita membagi secara acak dataset training menjadi set pelatihan/training dan set validasi. Set validasi akan digunakan untuk menentukan penghentian awal. Setelah training selesai, dilakukan pengujian model pada set test CIFAR10.

Kemudian digunakan juga library PyTorch Lightning . PyTorch Lightning adalah kerangka kerja yang menyederhanakan kode yang Anda perlukan untuk melatih, mengevaluasi, dan menguji model di PyTorch. Ini juga menangani login ke TensorBoard , toolkit visualisasi untuk eksperimen Machine Learning, dan menyimpan model checkpoints secara otomatis dengan overhead kode minimal dari pihak kami.

Bagian penting lainnya dari PyTorch Lightning adalah konsep callback. Callback adalah fungsi mandiri yang berisi logika non-esensial dari Lightning Module yang digunakan. Mereka biasanya dipanggil setelah menyelesaikan pelatihan epoch, tetapi juga dapat memengaruhi bagian lain dari lingkaran pelatihan yang dilakukan. Misalnya, akan digunakan dua callback yang telah ditentukan sebelumnya: LearningRateMonitor dan ModelCheckpoint. Monitor learning rate menambahkan learning rate saat ini ke TensorBoard, yang membantu memverifikasi bahwa penjadwal learning ratenya berfungsi dengan benar. Model checkpoint callback memungkinkan untuk menyesuaikan rutinitas penyimpanan checkpoints. Misalnya, berapa banyak checkpoints yang harus disimpan, kapan harus disimpan, metrik mana yang harus diperhatikan, dan lain-lain.

V. Analysis and Interpretation

Pengukuran performa untuk akurasi arsitektur CNN terdapat pada tabel dibawah. Dari hasil percobaan, akurasi tertinggi ditunjukkan oleh arsitektur ResNet. ResNet ini juga memiliki dua blok yaitu blok ResNet asli dan blok ResNet pra-aktivasi. Blok ResNet asli dalam visualiasinya sudah menunjukkan lapisan apa saja yang diterapkan sehingga memudahkan ketika kita ingin mengurangi dimensi gambar dalam hal lebar dan tinggi. Blok ResNet Pra-Aktivasi ini menerapkan kinerja ResNet menjadi lebih baik karena aliran yang dijamin memiliki matriks identitas dan menerapkan downsampling. ResNet juga merupakan arsitektur yang tercepat dari DenseNet dan GoogleNet karena memiliki lebih banyak lapisan yang diterapkan secara berurutan dalam implementasi primitif.

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

Fig. 1. perbandingan akurasi model

Dapat dilihat bahwa semua model memiliki kinrja yang cukup baik. GoogleNet adalah model untuk mendapatkan kinerja terendah pada 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 mengingat jumlah hyperparameter yang besar. ResNet mengungguli DenseNet dan GoogleNet lebih dari 1 persen pada validasi, sementara ada perbedaan kecil antara kedua versi, versi asli dan versi pra-aktivasi. Dari sini kita dapat menyimpulkan bahwa jaringan dangkal, tempat dari fungsi aktivasi tidak terlalu penting.

Secara umum, kita dapat menyimpulkan bahwa ResNet adalah arsitektur yang sederhana namun kuat. Jika kita akan menerapkan model pada yang lebih kompleks dengan gambar yang lebih besar serta lebih banyak lapisan dalam jaringan, kemungkinan kita akan melihat tugas yang lebih besar antara GoogleNet dan melewatkan koneksi yang lebih besar antara GoogleNet dan 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, Dual Path Network (Chen dkk), sebenarnya merupakan kombinasi dari ResNet dan DenseNet yang menunjukkan bahwa keduanya menawarkan keuntungan yang berbeda.

VI. Conclusions and Recommendations

Laporan ini membahas tentang arsitektur CNN yang diuji dengan menggunakan Dataset CIFAR10. Dari hasil pengujian, arsitektur ResNet memiliki kinerja yang lebih baik daripada Inception/ GoogleNet dan DenseNet. Jadi, memulai dengan ResNet adalah ide yang bagus mengingat kinerja yang lebih unggul dan implementasinya yang sederhana.

References

[1] R, Aditya Yanuar. 10 Agustus 2018. *Inception Network*. https://machinelearning.mipa.ugm.ac.id/2018/08/10/inception-network/?msclkid=1ea2d066cd4f11ecb5f6081eb6fe6d44. Diakses pada : April. 27, 2022.

[2] Sasha Targ, Diogo Almeida, Kevin Lyman, *ResNet:Residual Network*. 2016. [E-book] Tersedia: googlescholar, https://doi.org/10.48550/arXiv.1603.08029/. Diakses pada : April. 27, 2022.

[3] P. Jasman, D. A. L . Putra Implementasi DenseNet Untuk Mengidentifikasi Kanker Kulit Melanoma , JuTISI 2020, 20 Desember 2020, Bandung, Indonesia. Tersedia: Google Scholar, http://dx.doi.org/10.28932/jutisi.v6i3.2814. [Diakses pada: 26 April 2022].