Implementasi Deep Learning Dalam Peforma Penggunaan Model Inception, ResNet dan DenSet Pada Dataset CIFAR

Muh. Risqullah Awaluddin (H071191053), Sakinah Yunus (H071191046), Indira Septianita Larasati (H071191023)
Sistem Informasi
Universitas Hasanuddin

Table Of Content

Implementasi Deep Learning Dalam Peforma Penggunaan Model Inception, ResNet dan DenSet Pada Dataset CIFAR

Muh. Risqullah Awaluddin (H071191053), Sakinah Yunus (H071191046), Indira Septianita Larasati (H071191023)
Sistem Informasi
Universitas Hasanuddin

***Abstract*—** Deep Learning adalah bagian dari CNN. Model kerja pada CNN yakni GoogleNet, ResNet dan DesNet. Pemilihan arsitektur CNN yang sesuai terkadang menimbulkan masalah tersendiri. Implementasi model ketiga tersebut akan menggunakan dataset CIFAR. Oleh karena itu diperlukan performa yang bagus untuk menghasilkan akurasi dengan cara mengvalidasi akurasi dan test akurasi untuk melihat peforma model yang mana dapat dipakai dari ketiga model tersebut.

I. Introduction

Deep Learning merupakan bagian dari pembelajaran mesin yang menggunakan Convolutional Neural Networks (CNN)yang dimana menggunakan metode pembelajaran mesin dengan meniru cara kerja sistem saraf otak manusia. Salah satu pembagian model kerja dari CNN yakni Inception, ResNet, dan DenSet. Convolutional neural network dapat menerima input berupa gambar, menentukan aspek atau obyek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk "belajar" mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya. Dengan arsitektur seperti itu, CNN dapat dilatih untuk memahami detail sebuah gambar dengan lebih baik. Dengan begitu, CNN dapat menangkap dependensi Spasial dan Temporal dalam sebuah gambar setelah kamu memberikan filter yang relevan. Arsitektur pada Convolutional Neural Network (CNN) memiliki kemampuan untuk mengekstraksi fitur secara otomatis. Banyak macam arsitektur Convolutional Neural Network (CNN) yang populer untuk digunakan salah satunya GoogLeNet, yang menggunakan jaringan dengan rangkaian paralel jaringan sisa (ResNet), dan jaringan yang terhubung secara padat (DenseNet). Adapun dataset yang digunakan yakni dataset CIFAR.

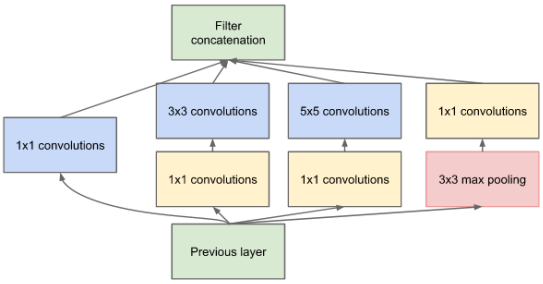
II. Problem Formulasi

1. Formula apa yang sesuai digunakan pada data set CIFAR ?

III. Discussion

A. GoogLeNet

GoogleNet mendapatkan predikat sebagai arsitektur kinerja paling baik. Kelebihan googlenet terletak pada *inception modules*. *Inception modules* terdiri dari sejumlah *convolution* kecil yang digunakan untuk mereduksi



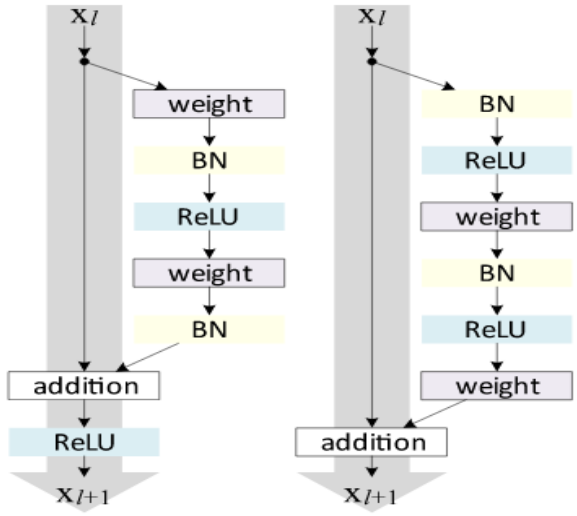
Picture 1. Inception Block

Berdasarkan Fig.1, Blok Inception menerapkan empat blok konvolusi secara terpisah pada peta fitur yang sama yakni konvolusi 1x1, 3x3, dan5x5, dan operasi max pool. Ini akan 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 fitur dari semua cabang digabungkan setelahnya, dan jika tidak ingin ada perbedaan ukuran fitur. Konvolusi 5x5, 25 kali lebih mahal daripada konvolusi 1x1, dapat menghemat banyak komputasi dan parameter dengan mengurangi dimensi sebelum konvolusi besar.

B. Resnet

Model ini berfokus pada computational accuracy yang merupakan model dengan lapisan paling banyak. Hal ini memungkinkan karena resnet menggunakan Residual Network sehingga tidak perlu mengkhawatirkan permasalahan diminishing gradient yang membuat performa model menurun jika model terlalu banyak lapisan. Resnet sendiri memiliki akurasi yang tinggi dibanding dengan model lainnya namun kecepatan training modelnya sendiri juga cepat meskipun tidak secepat GoogleNet ataupun Inception dengan memberikan solusi berupa blok residual. Adapun modelnya

$x_l + 1 = F(x_l)$ *dengan kita modelkan menjadi* $x_l + 1 = x_l + F(x_l)$ yang dimana F merupakan pemetaan non liner . Jika kita melakukan backpropagation pada koneksi residual maka kita memperoleh :

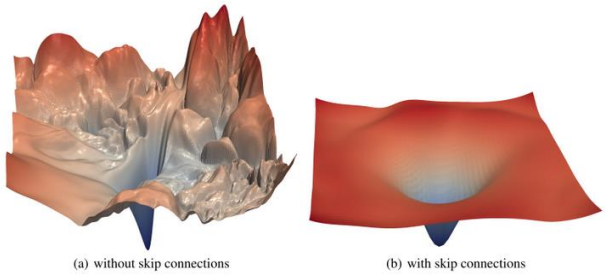


Picture. 2. ResNet Block

Jika kita melakukan backpropagation pada koneksi residual tersebut, kita memperoleh:

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

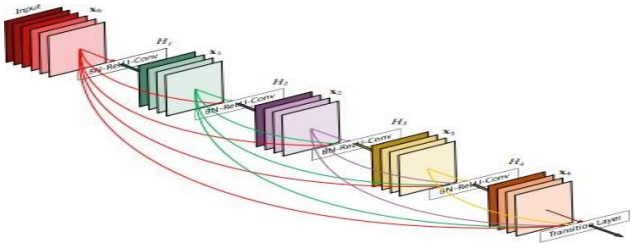
ResNet terbukti berhasil menghasilkan visualisasi lebih halus yang dapat mengeneralisasikan (with skip connection) dibandingkan jaringan pembelajaran tingkat lanjut(without skip connection), seperti sumbu x dan y pada gambar dibawah ini :



Picture 3. Visualisasi ResNet

C. DenseNet

DenseNet terbagi menjadi tiga bagian yakni DenseLayer, DenseBlock dan TransitionLayer. Pada DenseNet sendiri berfokus pada masalah yang timbul seiring dengan bertambahnya *layer* dimana informasi yang terdapat pada input akan menghilang seiring dengan informasi tersebut melewati banyak *layer* pada *neural network*. Pada DenseNet, hal ini dilakukan dengan membuat suatu *simple connectivity pattern* yang menghubungkan semua *layer* secara langsung satu sama lain. Setiap dense blocks, terdapat transition layers yang berguna untuk melakukan proses konvolusi dan pooling.



Picture 4: Arsitektur DenseNet

Pada gambar 4 arsitektur DenSet terdapat lima blok pertumbuhan k = 4, setiap lapisan sebagai input semua fitur (Huang et al., 2017). Pada gambar 1, x_0 yang dilewatkan melalui *Convolution Network*. Jaringan terdiri dari lapisan L, masing-masing yang mengimplementasikan transformasi non-linier Hl(·), di mana l adalah indeks lapisan. Hl (·) dapat menjadi fungsi komposit operasi seperti *Batch Normalization* (BN) (Ioffe & Szegedy, 2015), unit linier yang diperbaiki (ReLU) (Xavier Glorot, Antoine Bordes, 2011), *Pooling* (Y. LeCun, L. Bottou, Y. Bengio, 1998), atau *Convolution* (Konv). Output dari `lapisan ke-th disebut sebagai x`.

Diterapkan konvolusi 1x1 untuk pen- gurangan 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, pada tergantung apa yang dibutuhkan.

IV. Research Methodology

A. Dataset

Dataset ini menggunakan dataset CIFAR-10. Data set digunakan untuk mentrain dan mengevaluasi data dari 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 piksel. Dataset ini terdiri dari 50.000 data latih dan 10000 data uji yang terbagi menjadi 10 kelas yang terdapat pada dataset CIFAR-10. Sebelum dataset tersebut digunakan perlu dilakukan normalisasi pada dataset tersebut menggunakan nilai mean dan menormalisasikan dari dataset tersebut . Hal ini dilakukan untuk mengurangi risiko overfitting dan membantu proses pengerjaan CNN dalam mengeneralisasikan menghasilkan ouput yang terbaik.

B. Data Augmentasi

Dilakukan dua augmentasi secara random dengan tujuan untuk menghindari terjadinya overfitting. Oleh karena itu dilakukan pada beberapa tahap. Pertama, proses data augmentasi dilakukan dengan melakukan flip pada gambar secara horizontal dengan pendekatan 50% 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 resolusi 32x32 agar selanjutnya dapat dimasukkan kedalam model CNN

C. Model

Dalam membangun model CNN diperlukan tiga arsitektur CNN yakni GoogleNet, ResNet, dan DenseNet. Untuk arsitektur ResNet, akan digunakan dua 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 digu- nakan adalah ReLU dan *Stochastic Gradient Descent* (SGD). Selain itu, *hyperparameter* yang digunakan adalah *learning rate* 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 sehingga menghasilkan akurasi pada masing masing model yang dimana GoogleNet adalah model untuk mendapatkan kinerja terendah pada set validasi dan pengujian, meskipun sangat dekat dengan DenseNet. Pencarian hyperparameter yang tepat untuk semua ukuran untuk mwlalui GoogleNet kemungkinan akan meningkatkan akurasi model ke tingkat yang sama, tetapi ini juga rumit mengingat banyaknya hyperparameter. ResNet mengungguli DenseNet dan GoogleNet lebih dari 1% pada set validasi, sementara ada perbedaan kecil antara kedua versi, asli dan pra-aktivasi. Adapun tabel masing masing tingkat akurasi model

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

Picture 5. Tingkat akurasi setiap model

D. Pytorch Lightning

Pytorch Lightning merupakan framework yang digunakan pada pengujian model arsitektur GoogleNet, ResNet dan DesNet. 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.

V. Conclusion

Berdasarkan penjelasan yang didapatkan menghasilkan kesimpulan berupa model pada Resnet yang merupakan arsitektur sederhana namun akurat sehingga menghasilkan peforma lebih baik dibandingkan GoogleNet dan DesNet. Jika ingin menerapkan pada konteks yang kompleks dan gambar ukuran lebih besar maka adanya ketidakseimbangan antara GoolheNet dan arsitektur skipconnection seperti ResNet dan DesNet. Jika diliat pada implementasi penggunaan dari dataset CIFAR10 maka DesNet lebih unggul dibandingkan ResNet. Namun menunjukkan kombinasi ResNet dan DesNet yang saling memiliki masing masing keunggulan tersendiri.