**LAPORAN TUGAS BESAR**

**Klasifikasi Gambar Menggunakan *Demystifying Gradient Descent and Backpropagation via Logistic Regression***

Makalah ini ditujukan untuk memenuhi tugas mata Sistem Kendali Cerdas

S1 Teknik Elektro



Disusun oleh:

1. Shofiyyullah (1102164316)
2. Ravi Febrian Firjatullah (1102164288)

**FAKULTAS TEKNIK ELEKTRO**

**UNVERSITAS TELKOM**

**BANDUNG**

**2019**

# DAFTAR ISI

[DAFTAR ISI ii](#__RefHeading___Toc1075_572466509)

[BAB I 1](#__RefHeading___Toc1077_572466509)

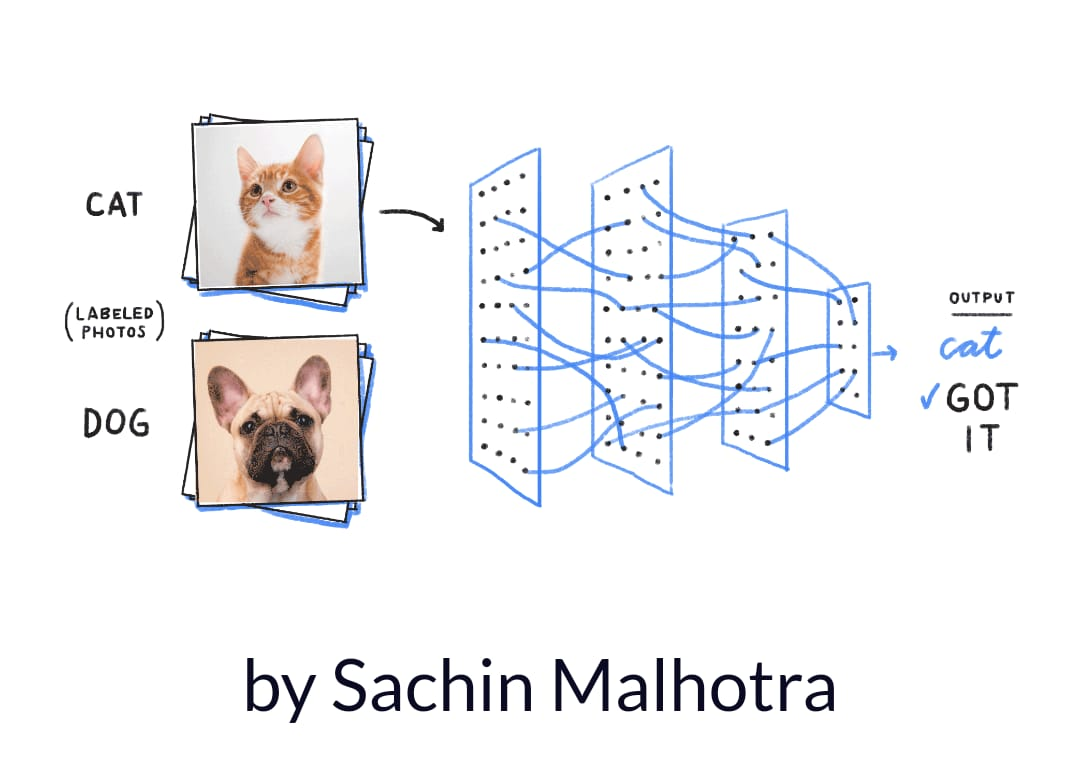
[BAB II 6](#__RefHeading___Toc1079_572466509)

[DAFTAR PUSTAKA 14](#__RefHeading___Toc1081_572466509)

# BAB I

**LATAR BELAKANG**

### Masalah



Salah satu problem dalam visi komputer yang telah lama dicari solusinya adalah klasifikasi objek pada citra secara umum. Bagaimana menduplikasi kemampuan manusia dalam memahami informasi citra, agar komputer dapat mengenali objek pada citra selayaknya manusia. Proses feature engineering yang digunakan pada umumnya sangat terbatas dimana hanya dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi pada jenis citra apapun. Hal tersebut diarenakan berbagai perbedaan antar citra antara lain perbedaan sudut pandang, perbedaan skala, perbedaan kondisi pencahayaan, deformasi objek dan sebagainya.

Kalangan akademisi yang telah lama bergelut pada problem ini. Salah satunya pendekatan yang berhasil digunakan adalah menggunakan Jaringan Syaraf Tiruan (JST) yang terinspirasi dari jaringan syaraf pada manusia. Konsep tersebut kemudian dikembangkan lebih lanjut dalam Deep Learning.

Pada tahun 1989, Yann LeCun dan teman-temannya berhasil melakukan klasifikasi citra kode zip menggunakan kasus khusus dari Feed Forward Neural Network dengan nama Convolution Neural Network (CNN)[1]. Karena keterbatasan perangkat keras, Deep Learning tidak dikembangkan lebih lanjut hingga pada tahun 2009dimana Jurgen mengembangkan sebuah Recurrent Neural Network (RNN) yang mendapatkan hasilsignifikan pada pengenalan tulisan tangan [2]. Semenjak itu, dengan berkembangnya komputasi pada perangkat keras Graphical Processing Unit (GPU), pengembangan DNN berjalan dengan pesat. Pada tahun 2012, sebuah CNN dapat melakukan pengenalan citra dengan akurasi yang menyaingi manusia pada dataset tertentu [3]. Dewasa ini, Deep Learning telah menjadi salah satu topik hangat dalam dunia Machine Learning karena kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti citra dan suara.

Metode Deep Learning yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah Convolutional Neural Network (CNN) [4]. Hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia [5] sehingga memiliki kemampuan mengolah informasi citra. Namun CNN, seperti metode Deep Learning lainnya, memiliki kelemahan yaitu proses pelatihan model yang lama. Dengan perkembangan perangkat keras, hal tersebut dapat diatasi menggunakan teknologi General Purpose Graphical Processing Unit (GPGPU).

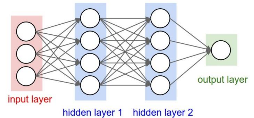
### Metoda

A. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik.CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang [4]. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeChun pada penelitiannyamengenai pengenalan angka dan tulisan tangan [1]. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi ImageNet Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode Deep Learning, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode Machine Learning lainnya seperti SVM pada kasus klasifikasi objek pada citra.

B. Konsep CNN

Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.

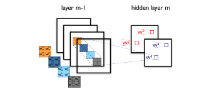


Sebuah MLP seperti pada Gambar memiliki i layer (kotak merah dan biru) dengan masing-masing layer berisi j, neuron (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasian data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data inputpada layerdilakukan operasi lineardengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linearyang disebut sebagai fungsi aktivasi.

Pada CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti padaGambar.2. Dimensi bobot pada CNN adalah :

neuron input x neuron output x tinggi x lebar

Karena sifat proses konvolusi, maka CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara.



C. Arsitektur Jaringan CNN

JST terdiri dari berbagai layer dan beberapa neuron pada masing-masing layer. Kedua hal tersebut tidak dapat ditentukan menggunakan aturan yang pasti dan berlaku berbeda-beda pada data yang berbeda [7].

Pada kasus MLP, sebuah jaringan tanpa hidden layer dapat memetakan persamaan linear apapun, sedangkan jaringan dengan satu atau dua hidden layer dapat memetakan sebagian besar persamaan pada data sederhana.

Namun pada data yang lebih kompleks, MLP memiliki keterbatasan. Pada permasalahan jumlah hidden layerdibawah tiga layer, terdapat pendekatan untuk menentukan jumlah neuron pada masing-masing layer untuk mendekati hasil optimal. Penggunaan layer diatas dua pada umumnya tidak direkomendasikan dikarenakan akan menyebabkan overfitting serta kekuatan backpropagation berkurang secara signifikan.

Dengan berkembangnya deep learning, ditemukan bahwa untuk mengatasi kekurangan MLP dalam menangani datakompleks, diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih mudah dimengerti oleh MLP. Hal tersebut memicu berkembangnya deep learningdimana dalam satu model diberi beberapa layer untuk melakukan transformasi data sebelum data diolah menggunakan metode klasifikasi. Hal tersebut memicu berkembangnya model neural network dengan jumlah layerdiatas tiga. Namun dikarenakan fungsi layerawal sebagai metode ekstraksi fitur, maka jumlah layer dalam sebuah DNN tidak memiliki aturan universal dan berlaku berbeda-beda tergantung dataset yang digunakan.

Karena hal tersebut, jumlah layerpada jaringan serta jumlah neuron pada masing-masing layerdianggap sebagai hyperparameterdan dioptimasi menggunakan pendekatan searching.

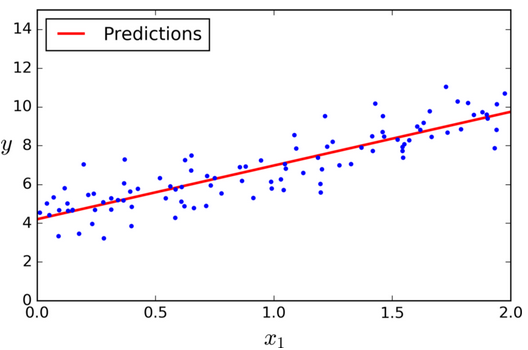
### Teknologi (The Age old Machine Learning Algorithm)

Mari kita mulai dengan pengantar yang sangat singkat tentang apa yang dilakukan salah satu algoritma tertua dalam Pembelajaran Mesin.

Ambil beberapa titik pada grafik 2D, dan gambar garis yang sesuai dengan mereka sebaik mungkin. Apa yang baru saja dilakukan adalah meng-generalisasikan dari beberapa contoh pasangan nilai input (X) dan nilai output (Y) ke fungsi umum yang dapat memetakan nilai input apapun ke nilai output.

Ini dikenal sebagai regresi linier, dan ini adalah teknik yang luar biasa untuk mengekstrapolasi fungsi umum dari beberapa pasangan input-ouput.

Oleh karena itu mengapa memiliki teknik seperti itu luar biasa: ada sejumlah fungsi yang tak terhitung di dunia nyata yang menemukan persamaan adalah tugas yang sangat sulit tetapi mengumpulan pasangan input-output adalah tugas yang relatif lebih mudah untuk dilakukan.



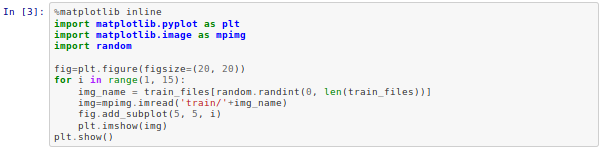
# BAB II

**LANGKAH KERJA**

### *Data Training*

- Dapatkan daftar nama file

- Mencetak gambar acak dari data

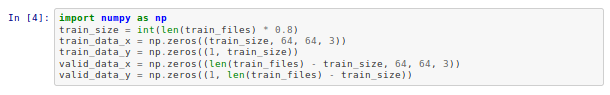
- Inisialisasi Numpy Vektor

a. Kita bagi data menjadi set train dan validasi

b. 80% untuk pelatihan dan 20% untuk validasi

c. Selain itu, kita ubah menjadi ukuraran gambar menjadi dimensi

(64 64 3). Menjadi dimensi array

- Gunakan imageio untuk mengonversi file gambar ke array numpy

Kita buah array Python tuple. Tupel berisi larik numpy untuk

gambar dan 1 jika itu anjing atau lainnya adalah 0

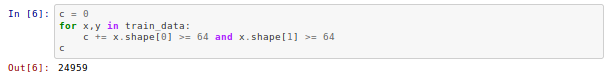


- Verifikasi apakah downscaling atau upscaling

Kita ulangi semua gambar dan kami menemukan bahwa 99%

gambar dapat dengan mudah diturunkan menjadi dimensi

(64 64 3)

- Ubah ukuran gambar

a. Iterasi atas daftar python untuk seluruh data

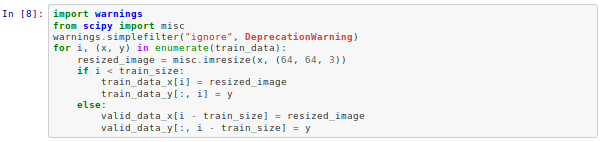
b. ubah ukuran gambar menjadi (64, 64, 3)

c. alokasikan data untuk pelatihan atau set validasi yang sesuai

d. Fungsi imresize dalam spicy sudah usang. Perlu menggunakan

skimage sebagai gantinya. Menggunakan paket peringatan untuk

menutup pesan peringatan yang sudah usang.

- Save data

Kami menggunakan numpy.savez untuk menyimpan banyak array

ke dalam file .npz tunggal, yang tidak terkompresi.



### *Testing Data*

*-*Import

a. Numpy adalah barebones dari jaringan

b. h4py digunakan untuk membaca dataset Coursera.

c. Matplot adalah untuk angka-angka di notebook.

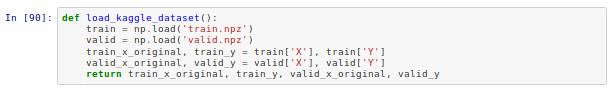
d. Kita juga buat sejajar sehingga dapat membuat grafik di sini di

notebook itu sendiri



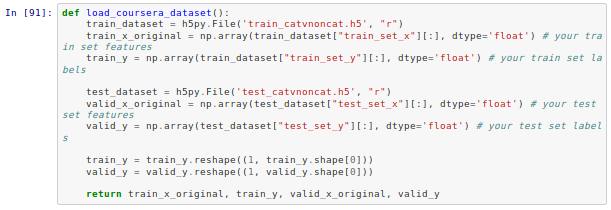
- Kaggle Dataset

Dataset menyediakan gambar dalam banyak ukuran yang bervariasi. Kami memproses data dan mengonversi semua gambar ke ukuran standar (64, 64, 3) dan kemudian menyimpan data ke dalam dua file train.npz dan valid.npz. Kami menggunakannya untuk memuat data.



- Coursera Dataset

Dataset ini dibuat sebagai bagian dari Spesialisasi Pembelajaran Jauh oleh Prof. Andrew NG. Dataset ini sangat kecil komparatifnya dan skrip pemuatan data telah diambil dari sumber kursus itu sendiri.

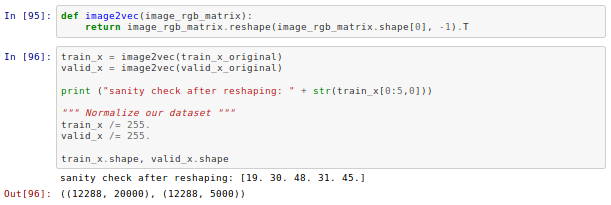


Muat dataset mana yang ingin Anda mainkan

 Mencetak bentuk semua array. Catatan: jika Anda memuat dataset Anda sendiri, pastikan untuk membawa semua gambar dalam format ini. Dimensi suatu gambar dapat berubah dengan jelas dan jumlah pelatihan dan sampel uji dapat bervariasi, tetapi urutannya harus sebagai berikut  
  
 X = (sample\_size, img\_x, img\_y, 3) Y = (1, sample\_size)

 - Image2Vec

Ratakan vektor seperti yang dijelaskan dalam sel sebelum dan juga kita perlu menormalkan dataset kami sehingga nilainya tidak menjadi terlalu besar untuk sigmoid. Sigmoid cenderung mengembalikan 0 untuk nilai yang sangat besar atau sangat kecil dan karenanya kami ingin menghindarinya di sini. Kami menggunakan fungsi sigmoid karena ini adalah masalah klasifikasi biner.



- Sigmoid Function

Di bawah fungsi adalah grafik yang menunjukkan plot untuk hal yang sama. Seperti yang dapat kita lihat dari gambar, untuk nilai yang sangat rendah atau sangat tinggi, sigmoid mengembalikan nilai 0. Karenanya, kita perlu menormalkan data gambar kita. Juga, untuk inisialisasi bobot dan bias, kita perlu memastikan mereka cukup kecil untuk belajar untuk melanjutkan pada kecepatan yang layak. Nilai bobot yang terlalu besar dapat menyebabkan kerusakan.

 - Mean Squared Loss Function

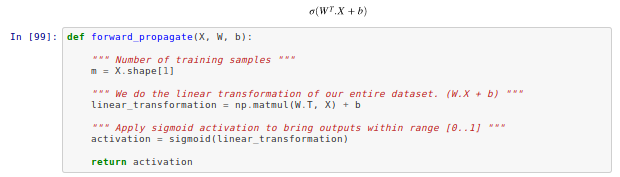
Di sini kita menggunakan fungsi kerugian kuadrat rata-rata. Bagian rata-rata dilakukan di dalam fungsi pemanggil.



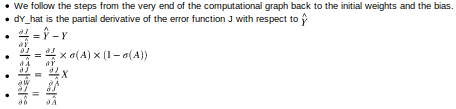
- Forward Propagation

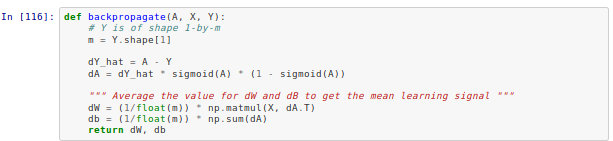
a. apply linear transformation

b. apply sigmoid activation



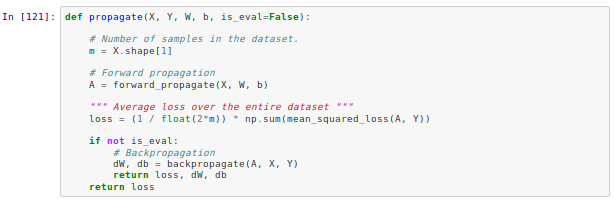
- Backpropagation





- Forward and Back-porpagation combined

Perhatikan penggunaan variabel is\_eval di sini. Kami menggunakannya sehingga kami dapat memanggil fungsi propagasi yang sama bahkan selama waktu pengujian. Selama waktu pengujian kita tidak benar-benar perlu melakukan backpropagation. Kami hanya perlu meneruskan propagasi dan perhitungan kerugian kemudian.

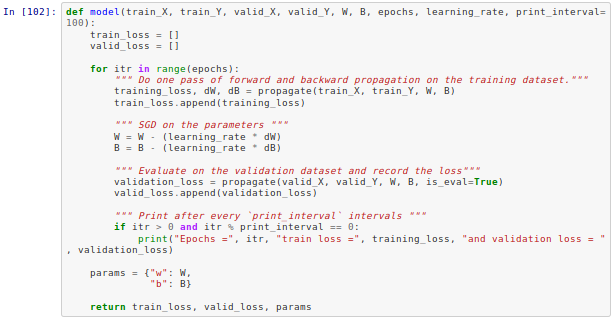
 - Menyatukan semuanya di bawah fungsi model

a. Iterasi di epochs

b. Lakukan maju dan perbanyakan kembali pada set pelatihan.

c. Lakukan gradient descent mis. Sesuaikan bobot dan bias.

d. Jalankan model pada set data validasi untuk mengukur kinerja aktual.



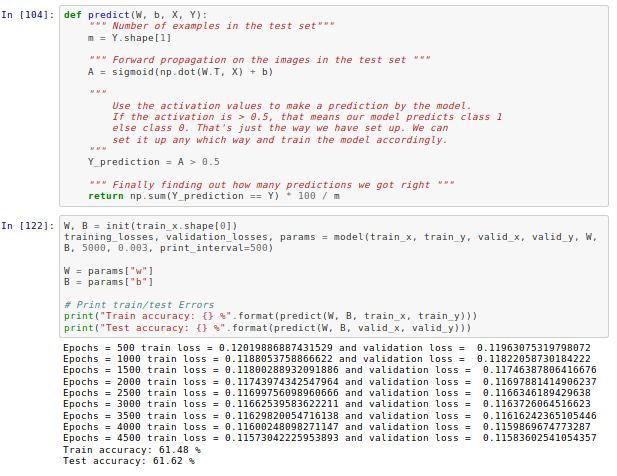
- Parameter inisialisasi model

Kita dapat menggunakan inisialisasi acak di sini juga, tetapi ini berfungsi sama baiknya untuk kasus penggunaan kami.

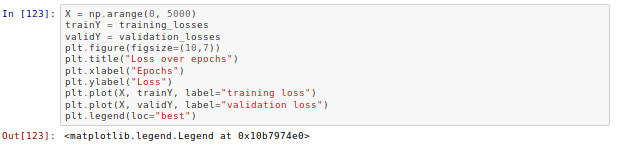


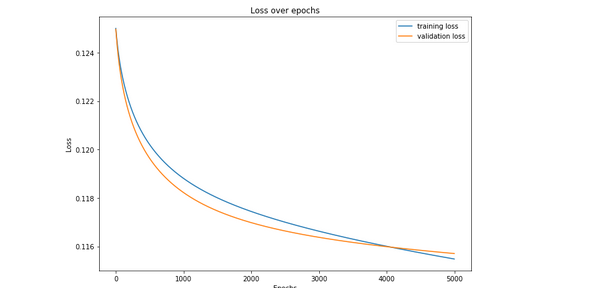
- Fungsi prediksi

Kita dapat memanggil fungsi ini untuk membuat prediksi aktual pada dataset validasi kami dan fungsi ini mengembalikan persentase gambar yang diklasifikasikan dengan benar.

 - Merencanakan kerugian epochs

Catatan: kehilangan validasi berkurang selama 5000 zaman yang tidak menunjukkan tanda-tanda overfitting apa pun. Juga, kehilangan validasi meniru kehilangan pelatihan di sini yang merupakan tanda model yang terlatih dengan baik.





- Prediksi pada gambar



# 

# DAFTAR PUSTAKA

[1] https://github.com/edorado93/Power-Of-A-Neuron

[2]https://www.freecodecamp.org/news/demystifying-gradient-descent-and-backpropagation-via-logistic-regression-based-image-classification-9b5526c2ed46/

[5]https://media.neliti.com/media/publications/191064-ID-klasifikasi-citra-menggunakan-convolutio.pdf?source=post\_page---------------------------