LAPORAN TUGAS BESAR 1

IF4074 Pembelajaran Mesin Lanjut

Feed Forward Nerual Network

Dion Saputra - 13516045
Restu Wahyu Kartiko - 13516155

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2019

Laporan Tugas Besar 1 IF4174 Pembelajaran Mesin Lanjut
Mini-Batch Gradient Descent (Feed Forward Neural Network)

Dion Saputra - 13516045
Restu Wahyu Kartiko - 13516155

Penjelasan Algoritma

Algoritma *Mini-Batch Gradient Descent* dengan *Feed Forward Neural Network* diimplementasikan dalam satu kelas Python yang diberi nama kelas NeuralNetwork. Kelas ini memiliki beberapa atribut dan fungsi yang digunakan untuk algoritma *learning*-nya. Beberapa atribut dan fungsi yang diimplementasikan menggunakan Pustaka python bernama Numpy untuk memudahkan penyimpanan data dalam bentuk matriks dan operasi-operasi terkait matriks.

Kelas NeuralNetwork terdiri atas 9 atribut, yaitu *weights, biases, layers_size, activations, previous_gradients, learning_rate, momentum, epochs,* dan *batch_size*. Empat atribut terakhir dapat diinisialisasi dengan menggunakan parameter yang disediakan oleh konstruktor kelas Neural Network, sementara atribut lainnya diinisialisasi dengan list kosong.

Atribut *weights, biases,* dan *layers_size* pada kelas NeuralNetwork berturut-turut ditujukan untuk menyimpan nilai bobot, bias, dan banyak neuron pada *layer*. Atribut *weights* berupa *list* dari matriks dengan dimensi *M x N*, dengan *M* merupakan banyak neuron di *layer* sebelumnya dan *N* adalah banyak neuron di *layer* berikutnya. Atribut *biases* berupa *list* dari matriks dengan ukuran 1 x *M*, dimana M merupakan banyak neuron di *layer* tersebut. Sedangkan atribut *layers_size* berupa *list* dari *integer* yang menyatakan jumlah neuron tiap *layer*.

Atribut *activations* dan *previous_gradient* digunakan untuk menyimpan nilai fungsi aktivasi menggunakan fungsi sigmoid tiap neuron dan nilai dari gradient hasil learning epoch sebelumnya untuk digunakan di momentum. Kedua atribut ini juga berupa *list* dari matriks. Atribut *learning_rate*, momentum, *epochs,* dan *batch_size* sama dengan atribut *neural network* pada umumnya.

Fungsi yang ada pada kelas NeuralNetwork adalah konstruktor, *add_hidden_layer*, *add_input_layer*, *add_output_layer*, *sigmoid, feed_forward, sigmoid_derivation*, *cross_entropy, error, back_propagation, train, predict,* dan *get_accuration*. Fungsi *sigmoid, sigmoid_derivation, cross_entropy,* dan *get_accuration* merupakan fungsi pembantu yang secara berturut-turut menyatakan nilai sigmoid dari suatu bilangan, nilai dari turunan fungsi

sigmoid, menentukan perbedaan dari hasil prediksi dan label, dan menentukan akurasi dari model *learning*.

Fungsi *add_hidden_layer, add_input_layer*, dan *add_output_layer* digunakan untuk menambahkan *layer* pada *neural network*. Pada saat fungsi-fungsi ini dipanggil, program secara dinamis akan melakukan inisialisasi dari *weight* dan *bias* dari *layer-layer* terkait. Urutan penggunaan ketiga fungsi ini dimulai dari *add_hidden_layer*, *add_output_layer*, dan *add_input_layer*.

Fungsi *feed_forward* digunakan untuk melakukan perhitungan nilai output dari bobot *neural network* saat ini. Hal tersebut diimplementasikan dengan melakukan *update* nilai *activation* setiap neuron menggunakan perkalian titik dari nilai *activation* neuron-neuron pada *layer* sebelumnya dengan nilai bobot dari *layer* terkait. Hasil perkalian titik tersebut kemudian ditambahkan dengan nilai biasnya.

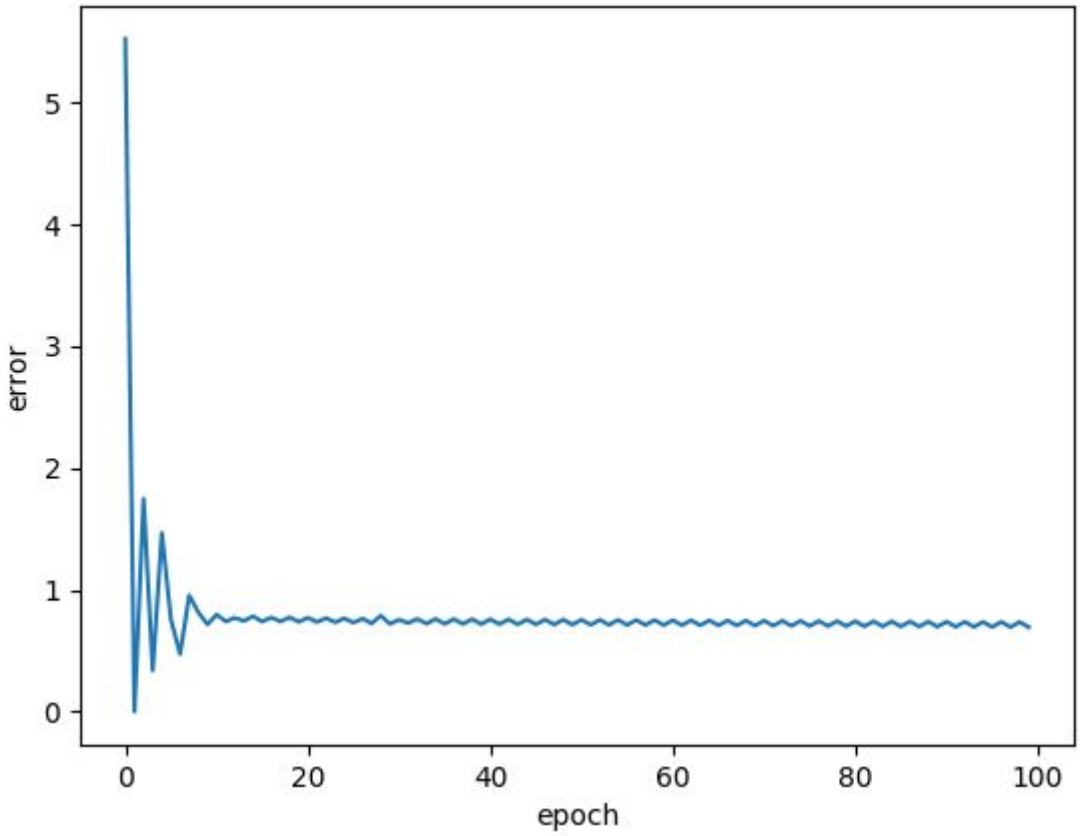
Fungsi *back_propagation* digunakan untuk melakukan *update* bobot dari neural network. Ada variabel lokal *delta* yang digunakan untuk mempropagasi turunan error terhadap bobot ke *layer-layer* sebelumnya. Variabel tersebut diinisialisasi dari hasil *cross_entropy* dari nilai *activation layer* output dengan label dari data *training*. Nilai *delta* ini digunakan untuk melakukan perhitungan *gradient* berdasarkan *learning_rate, delta,* dan momentum. Nilai *gradient* akan digunakan untuk melakukan *update* bobot dan bias.

Fungsi *train* dan *predict* secara berturut-turut digunakan untuk melakukan *learning* model dari data *train* dan melakukan prediksi dari data *test*. Fungsi train dilakukan dengan menggunakan pendekatan Mini-Batch dimana data dibagi dalam beberapa *batch*, lalu *feed_forward* pada setiap *instance* data dalam *batch_size*, sedangkan *back_propagation* dilakukan di akhir *batch.*

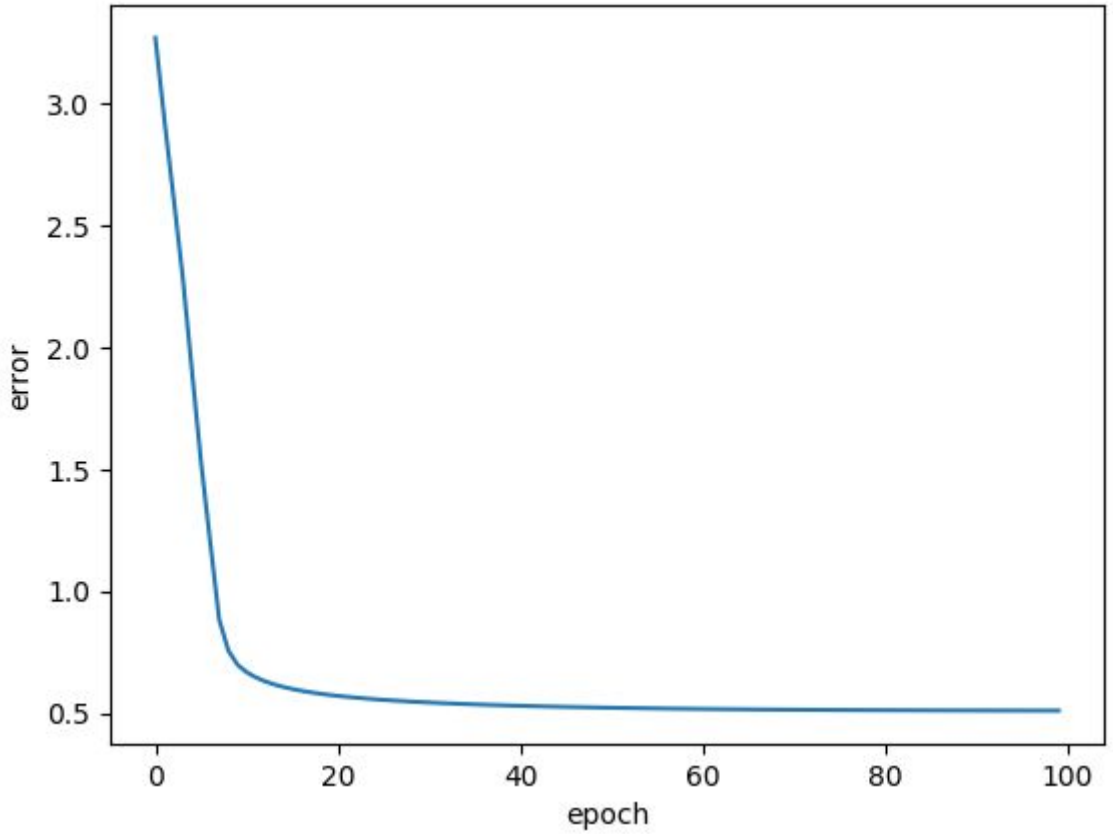
Eksperimen Klasifikasi

Eksperimen dilakukan dengan membagi data yang menjadi data train (11 instances), dan data test(3 instances). Hal ini karena sedikitnya data yang tersedia.

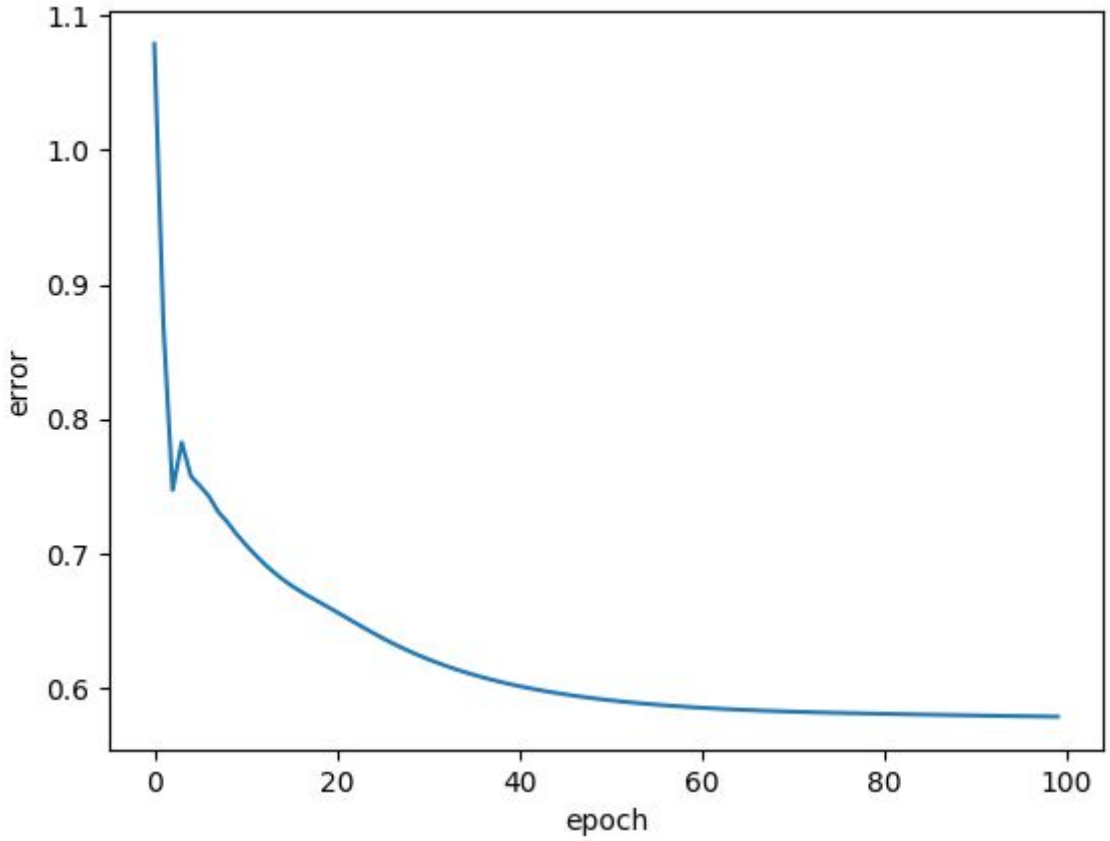
- Eksperimen 1
  epochs = 100
  learning_rate = 0.1
  momentum = 1
  batch_size = 5
  Training accuracy : 100%
  Test accuracy : 100%



- Eksperimen 2
  epochs = 100
  learning_rate = 0.001
  momentum = 0.01
  batch_size = 5
  Training accuracy : 100%
  Test accuracy : 100%



- Eksperimen 3
  epochs = 100
  learning_rate = 0.001
  momentum = 1
  batch_size = 5
  Training accuracy : 100%
  Test accuracy : 100%



Analisis Hasil Eksperimen

Dari tiga eksperimen yang dijalankan terdapat beberapa yang bisa diamati. Yang pertama adalah semakin besar learning rate (0.1 pada kasus ini), hasil training lebih cepat untuk *converge,* bisa dilihat di eksperimen 1. Namun dalam eksperimen 1 bisa dilihat bahwa nilai error terus mengalami perubahan yang relatif signifikan. Hal ini karena momentum yang dimasukkan bernilai 1.

Pada eksperimen 2, model lebih lama untuk *converge* dibanding eksperimen 1. Hal ini karena learning ratenya lebih kecil yaitu 0.001. Bisa dilihat grafik errornya relatif halus (tidak lompat lompat) karena momentumnya kecil.

Pada eksperimen 3, dengan learning rate 0.001 dan momentum 1, model lebih lama untuk *converge* dibanding eksperimen 1 dan 2. Selain itu, ada sedikit perubahan nilai error yang signifikan di epoc 5-10. Hal ini karena nilai memontumnya 1.

Namun demikian, model hasil training terus mengalami perubahan setiap kali dijalankan. Hal ini karena data set yang dimiliki sangat kecil sehingga tidak akurat.

Pembagian Tugas

13516045: 50%
13516155: 50%