LAPORAN TUGAS BESAR 1C
IF3270 Pembelajaran Mesin

IMPLEMENTASI MINI-BATCH GRADIENT DESCENT

Disusun oleh:

Lukas Kurnia Jonathan / 13517006
Eginata Kasan / 13517030
Vivianni / 13517060
Rika Dewi / 13517147

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

2020

Daftar Isi

Bab 1 Implementasi

Pada tugas besar kali ini, kami melakukan implementasi Mini-Batch Gradient Descent. Mini-Batch Gradient Descent melakukan update terhadap weight di akhir setiap batchnya sehingga merupakan metode gradient descent yang lebih fleksibel dibandingkan dengan Batch Gradient Descent namun lebih efektif dibandingkan Stochastic Gradient Descent.

Implementasi Mini-Batch Gradient descent diawali dengan melihat jumlah kelas yang akan diklasifikasikan oleh model. Untuk setiap kelas, akan dibangun Neural Network yang menghasilkan probabilitas benar kelas tersebut dan bukan kelas tersebut. Pada kasus kali ini, digunakan dataset iris yang akan diacak terlebih dahulu lalu diklasifikasikan untuk 3 buah kelas yaitu setosa, versicolor, dan virginica. Sehingga akan terbangun 3 buah Neural Network yang dilatih dengan training data yang sama hanya saja berbeda di bagian encoding untuk kelas targetnya, misal untuk dataset yang akan digunakan mengklasifikasikan setosa maka hasil encoding kelas targetnya akan bernilai 1 jika setosa dan bernilai 0 jika bukan setosa, begitu pula untuk dataset kelas lainnya. Proses klasifikasi akan mengambil kelas dengan probabilitas tertinggi dari output setiap Neural Network.

Untuk setiap Neural Network yang telah dibentuk, dataset akan dibagi menjadi beberapa batch yang berukuran kecil. Setiap *instance data* dalam batch kemudian diambil feature valuenya dan diberikan kepada Feed Forward function sebagai parameter. Feed Forward function akan mengubah dan menghitung net dan output dari setiap simpul di dalam graph dimulai dari input layer dengan menggunakan fungsi aktivasi sigmoid. Setelah graph selesai dimodifikasi oleh Feed Forward function, akan dipanggil Back Propagation function dengan nilai target sebagai parameternya. Back Propagation function akan memodifikasi dan menghitung *delta weight* (dw) untuk setiap *edge* dalam *graph* mulai dari output layer. Ketika sebuah batch telah selesai diproses, maka akan diupdate *weight* setiap *edge* dari *graph* dengan menambahkannya dengan *learning weight* * *delta weight* (w' = w + learning_rate * dw). Proses training akan berhenti apabila iterasi (untuk 1 epoch) sudah melebihi *max iteration* atau error kumulatif berada di bawah *error threshold* yang diberikan.

Bab 2 Hasil Eksekusi

Model Gradient Batch Descent

Model dari Mini-Batch Gradient Descent hasil eksekusi dapat dilihat pada directory `test`. Berikut merupakan sample dari model yang digunakan:

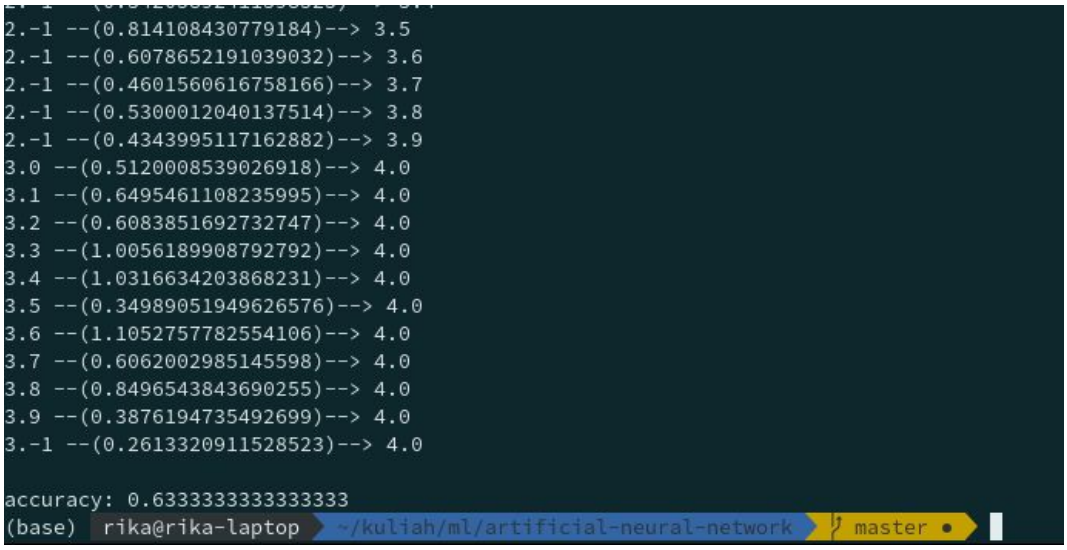
```
2.-1 --(0.7265048349664677)--> 3.0
```

1. Angka **2** merupakan **id layer** dari neural network (0 = input layer, 1 s.d n-1 = hidden layer, n = output layer)
2. Angka **-1** merupakan **id node** dari list of root. Dimana list of root berisi sejumlah node dari sebuah layer. (-1 = bias, else = id node)
3. Angka **0.7265048349664677** merupakan weight dari edge yang menghubungkan kedua node yang tertulis (node kiri dengan node kanan)
4. Angka **3** merupakan **id layer** dari neural network yang terhubung dengan layer sebelumnya. Pada contoh ini menghubungkan node 2 dan 3 (0 = input layer, 1 s.d n-1 = hidden layer, n = output layer)
5. Angka **0** merupakan **id node** dari list of root yang terhubung dari node sebelumnya. Dimana list of root berisi sejumlah node dari sebuah layer. (-1 = bias, else = id node)

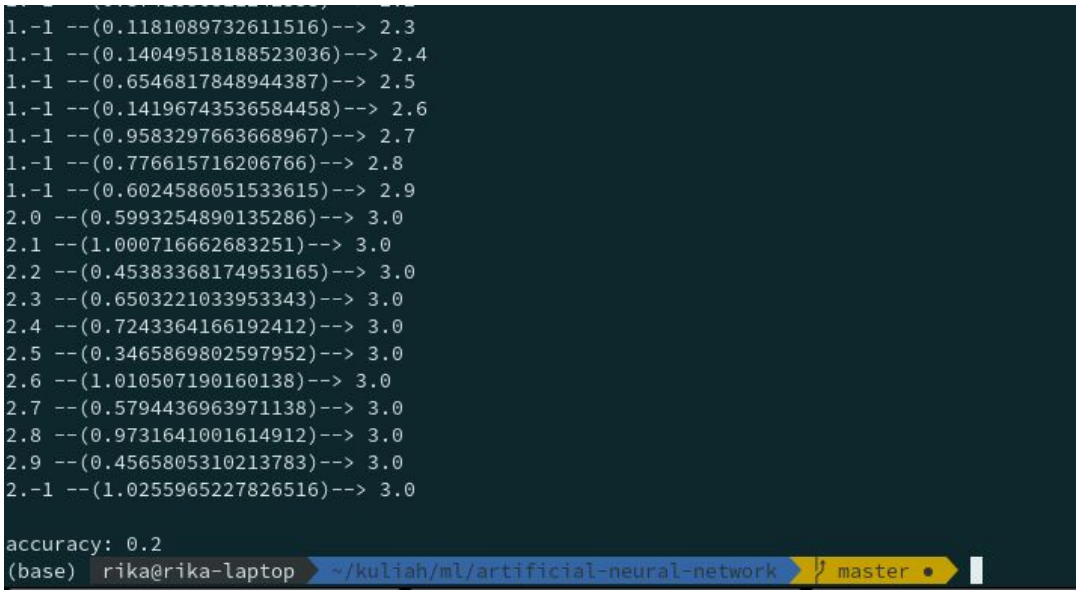
Hasil Percobaan

Berikut adalah beberapa hasil percobaan mini-batch gradient descent dengan berbagai parameter:

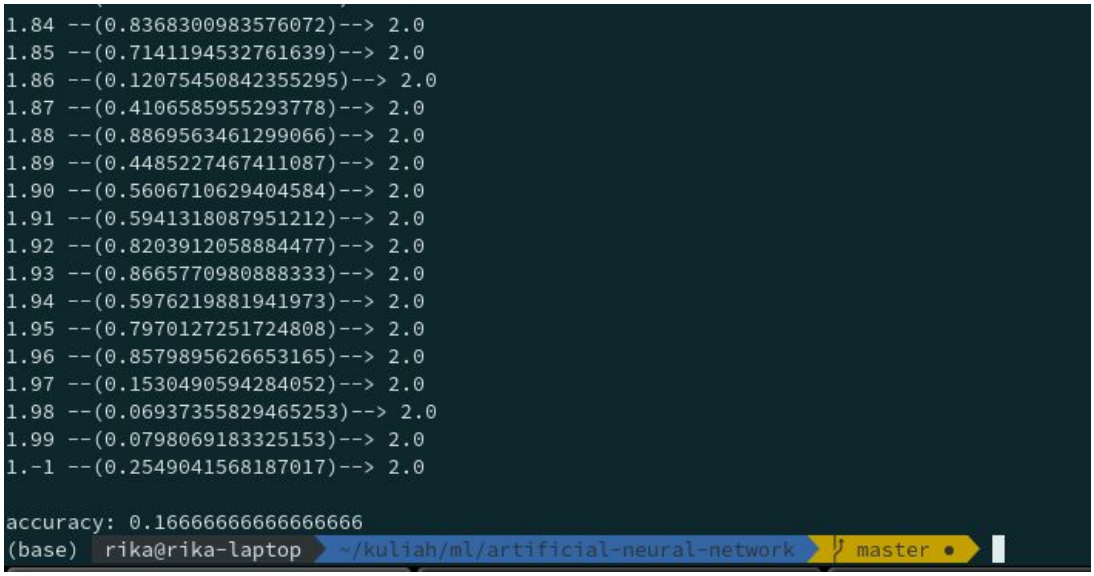
1. Dengan menggunakan learning rate = 0.1, jumlah hidden layer = 3, jumlah tiap node pada hidden layer = 10, batch size = 30, error threshold = 0.3, dan max iteration = 10, didapatkan akurasi sebesar 63%. Model dapat dilihat pada `test/1.txt`



2. Dengan menggunakan learning rate = 0.8, jumlah hidden layer = 2, jumlah tiap node pada hidden layer = 10, batch size = 100, error threshold = 0.3, dan max iteration = 20, didapatkan akurasi sebesar 20%. Model dapat dilihat pada `test/2.txt`



3. Dengan menggunakan learning rate = 0.6, jumlah hidden layer = 1, jumlah tiap node pada hidden layer = 100, batch size = 30, error threshold = 0.3, dan max iteration = 10, didapatkan akurasi sebesar 17%. Model dapat dilihat pada `test/3.txt`



Bab 3 Perbandingan dengan Library MLP sklearn

Perbandingan

Perbandingan dilakukan dengan menggunakan konfigurasi parameter yang sama.

1. Hasil eksekusi untuk contoh konfigurasi ke-1 pada sklearn

```
In [277]: print("Accuracy of MLPClassifier : ", accuracy(cm))
          Accuracy of MLPClassifier :  0.16666666666666666
```

2. Hasil eksekusi untuk contoh konfigurasi ke-2 pada sklearn

```
In [263]: print("Accuracy of MLPClassifier : ", accuracy(cm))
          Accuracy of MLPClassifier :  0.8
```

3. Hasil eksekusi untuk contoh konfigurasi ke-3 pada sklearn

```
In [291]: print("Accuracy of MLPClassifier : ", accuracy(cm))
          Accuracy of MLPClassifier :  0.36666666666666664
```

Analisis

Menurut hasil perbandingan antara hasil eksekusi program mini-batch gradient descent, dapat dilihat bahwa terdapat perubahan akurasi yang fluktuatif dengan berubahnya parameter hidden layer, jumlah node hidden layer, learning rate, batch size, max iteration dan error threshold.

Training yang dilakukan selalu mencapai maksimum iteration dengan kecenderungan error yang meningkat. Maka kelompok kami menyimpulkan bahwa dataset iris tidaklah konvergen.

Hal ini didukung oleh warning yang terlihat pada saat eksekusi melalui library sklearn yaitu:

```
/home/lukaskurnia/anaconda3/lib/python3.7/site-packages/sklearn/neural_network/
multilayer_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum
iterations (10) reached and the optimization hasn't converged yet.  %
self.max_iter, ConvergenceWarning)
```

Bab 4 Pembagian Tugas

Nama - NIM	Tugas
Lukas Kurnia J. 13517006 (25%)	Arsitektur Graph, Predict, MLP, Laporan
Eginata Kasan 13517030 (25%)	Integrasi (Mini-Batch), MLP, Laporan
Vivianni 13517060 (25%)	Feed Forward, MLP, NN, Laporan
Rika Dewi 13517147 (25%)	Back Propagation, Arsitektur Graph, MLP, NN, Laporan