

Tugas Besar I IF4074 - Pembelajaran Mesin Lanjut

Mini-batch Gradient Descent (*Feed Forward Neural Network*)

Senapati Sang Diwangkara / 13516107

Hafizh Budiman / 13516137

Christian Wibisono / 13516147

1. Penjelasan Kode

Kode kami dibagi dalam 3 modul, yaitu

- Fully connected layer (feedforward.py)
- Activation layer (activation.py)
- Loss function (loss.py)

Ketiga modul ini diimplementasikan sebagai class dg method `__call__`, untuk forward pass, dan `calc_grad` untuk back propagation. Ketiga modul tersebut disusun untuk membentuk kelas model neural network.

Model neural network dibuat dengan menyusun fully connected layer dan activation layer berturut2 tergantung konfigurasi yang diberikan pada saat inisialisasi model (berapa layer dengan berapa node di masing-masing layer).

Forward pass dilakukan dengan memasukkan input ke masing-masing layer yang telah dinistansiasikan, secara terurut. Dari luar, forward pass akan diperlakukan seperti fungsi, yaitu dengan pemanggilan menggunakan tanda kurung (`model(X)`) yang akan mengembalikan output dari batch yang dimasukkan.

Backpropagation dilakukan dengan memanggil method `calc_grad` dari masing-masing modul yang telah diinstansiasikan di dalam model tersebut, secara terbalik (dari akhir ke awal). Koreksi weight dilakukan berbarengan dengan perhitungan gradien di backward pass.

`calc_grad` dibuat sebagai representasi dari 1 suku di aturan rantai pada gradient descent. Misal, untuk meng-update weight di layer terakhir, akan dihitung $d(\text{loss})/d(w_n)$ yang, dengan aturan rantai, akan di-expand menjadi:
$$= d(\text{loss})/d(a_n) * d(a_n)/d(z_n) * d(z_n)/d(w_n)$$

Satu suku di persamaan tersebut dihitung dengan `calc_grad`
$$= \text{loss.calc_grad}() * \text{sigmoid.calc_grad}() * a_{n-1}$$

2. Eksperimen

a. Dataset

Eksperimen ini menggunakan dataset weather yang terdiri dari 5 fitur dan 14 observasi.

```
@relation weather
```

```

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no

```

b. Data Partitioning dan Evaluation

Dalam eksperimen ini kami membagi data menjadi 2 bagian yaitu train dan test dengan teknik holdout untuk melakukan evaluasi model. Kami menggunakan modul `train_test_split` dari `scikit-learn` dengan `test_size=0.2` dan `random_state=1`.

c. Arsitektur Neural Network

Berikut merupakan arsitektur *Feed Forward Neural Network* yang akan diujikan dalam eksperimen kali ini.

Arsitektur A	3 layer : 1 input layer (6 nodes), 1 hidden layer (6 nodes), 1 output layer (1 nodes)
Arsitektur B	4 layer : 1 input layer (6 nodes), 2 hidden layer ([6,3] nodes), 1 output layer (1 nodes)
Arsitektur C	5 layer : 1 input layer (6 nodes), 3 hidden layer ([6,3,2] nodes), 1 output layer (1 nodes)
Arsitektur D	6 layer : 1 input layer (6 nodes), 4 hidden layer ([6,3,2,1] nodes), 1 output layer (1 nodes)

d. Parameter Neural Network

Berikut merupakan parameter untuk *Feed Forward Neural Network* yang akan diujikan dalam eksperimen kali ini.

Konfigurasi 1	Learning rate: 0.01 Epoch : 100 Momentum: 0.9 Batch Size:2
Konfigurasi 2	Learning rate: 0.01 Epoch : 50 Momentum: 0.9 Batch Size:2
Konfigurasi 3	Learning rate: 0.01 Epoch: 50 Momentum: 0.85 Batch Size:2
Konfigurasi 4	Learning rate: 0.05 Epoch: 100 Momentum: 0.9 Batch Size:2
Konfigurasi 5	Learning rate:0.01 Epoch:100 Momentum: 0.9 Batch Size: 4

e. Hasil Eksperimen

	1	2	3	4	5
A	0.66666	0.66666	0.66666	0.66666	0.66666
B	0.66666	0.66666	0.66666	0.66666	0.66666
C	0.66666	0.66666	0.66666	0.66666	0.66666
D	0.66666	0.66666	0.66666	0.66666	0.66666

f. Perbandingan dengan Library Lain

Skenario	ScikitLearn MLPClassifier	Pytorch	Keras / Tensorflow	Model Kami
A.1	0.66666	0.66666	0.66666	0.66666
A.2	0.66666	0.66666	0.66666	0.66666
B.1	0.66666	0.66666	0.66666	0.66666

B.4	0.66666	0.66666	0.66666	0.66666
------------	---------	---------	---------	---------

3. Analisis Hasil Eksperimen

Berdasarkan hasil eksperimen dapat dilihat bahwa pengubahan parameter maupun arsitektur tidak mempengaruhi akurasi model. Setelah kami menyelidiki hasil prediksi dari model ternyata model memprediksi label 1 untuk semua data test dan data train. Hal ini bisa disebabkan data hanya terdiri dari 14 baris sehingga model hanya dapat belajar dari sampel yang sangat minim. Hal ini menyebabkan model rentan terhadap *over-fitting*. Selain itu, adanya *outlier* dan *noise* pada dataset yang kecil akan menjadi masalah besar. Algoritma deep learning lebih cocok diterapkan untuk data yang banyak sehingga bisa mempelajari hubungan-hubungan non-linear dari dataset yang ada. Apabila dibandingkan dengan library-library *machine learning* populer lainnya model kami menghasilkan performa yang sama.

4. Pembagian Tugas

Nama	NIM	Tugas
Senapati Sang Diwangkara	13516107	Implementasi forward pass, backpropagation
Hafizh Budiman	13516137	Implementasi layer, desain eksperimen, implementasi momentum
Christian Wibisono	13516147	Implementasi minibatch, activation function, desain eksperimen