

Laporan Tugas 2 IF4074 Pembelajaran Mesin Lanjut
Long Short-Term Memory



Disusun oleh:

Jaya Mangalo	13520015
Jevant Jedidia	13520133
Vincent Christian	13520136

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

Kode Program

Berikut adalah kode program yang khusus berhubungan untuk tubes LSTM ini.

1. LSTM Layer

LSTM Layer menerima 3 parameter input: `num_timestep` yaitu jumlah timestep data, `input_size` yaitu besarnya input, dan `num_cell` yaitu jumlah cell LSTM yang dibuat dan juga jumlah output LSTM layer.

LSTM Layer memiliki 4 tipe fungsi :info/getters, init, execution dan save & load

- Info/getters, terdapat fungsi **getParamNum** yang mengembalikan jumlah parameters dan **getOutputShape** yang mengembalikan bentuk output.
- Init, fungsi *utility* yang untuk melakukan init awal, **initGatesStates**, menginisiasi nilai pada gates untuk timestep 0, dan **initWeightsBias**, menginisiasi bobot U, W, dan bias dengan nilai random pada setiap gates.
- Execution, fungsi utama yang digunakan oleh user, **setWeightsBias** dapat digunakan untuk mengisi bobot dan bias secara manual, serta **forward** untuk melakukan Forward Propagation.
- Save & Load, fungsi **saveModel** digunakan untuk mencetak model menjadi dict (digunakan oleh sequential model), sedangkan fungsi **loadModel** digunakan me-load model yang telah di save.

2. Sequential Class

Sequential Class melanjutkan yang dibuat pada Tubes1, perubahan utama adalah di penambahan LSTM untuk fungsi `addLayer`, `saveModel`, `loadModel`, `forward`, dan `info`. Fungsi lainnya seperti `train`, `updateWeight` tidak ditambahkan LSTM karena akan memerlukan backpropagation yang diluar scope tubes ini.

Hasil Eksperimen

1. Save Model

Save dilakukan dengan memanggil fungsi `saveModel` pada kelas `Sequential` dengan parameter berupa nama file. Berikut adalah kode untuk save model.

```
nn = Sequential(1,5)

lstm = LSTMLayer((4,5), 64)
Dense1 = DenseLayer(5, 'sigmoid', 0.5)

nn.addLayer(lstm)
nn.addLayer(Dense1)

nn.info()
```

Layer Type	Output Shape	Params
LSTMLayer	64	17920
DenseLayer	5	325
Total Parameters: 18245		

```
nn.saveModel("LSTM1.json")
```

Berikut adalah model yang disimpan pada file "LSTM1.json".

```
[
  {
    "type": "LSTMLayer",
    "parameters": {
      "num_timestep": 4,
      "input_size": 5,
      "num_cell": 64
    },
    "weights": {
      "U_f": [...],
      "U_i": [...],
      "U_c": [...],
      "U_o": [...],
      "W_f": [...],
      "W_i": [...],
      "W_c": [...],
      "W_o": [...]
```

```
    },
    "bias": {
        "b_f": [...],
        "b_i": [...],
        "b_c": [...],
        "b_o": [...]
    }
},
{
    "type": "DenseLayer",
    "parameters": {
        "units": 5,
        "activation": "sigmoid",
        "learning": 0.5
    },
    "weights": [...]
}
]
```

2. Load Model

Load dilakukan dengan memanggil fungsi loadModel pada kelas Sequential dengan parameter berupa nama file. Berikut adalah kode untuk load model yang sudah di save.

```
nn2 = Sequential(1,5)
nn2.loadModel("LSTM1.json")
nn2.info()
```

Layer Type	Output Shape	Params
LSTMLayer	64	17920
DenseLayer	5	325
Total Parameters: 18245		

3. Eksperimen Data Stock Market BMBA

```
dataset2 = create_sequences_dataset(merged_data,4)
dataset3 = create_sequences_dataset(merged_data,2)
dataset4 = create_sequences_dataset(merged_data,7)
```

a. Timestep = 4

```
predicted_data2 = []
for (data,_) in dataset2:
    prediction = nn.forward(data)
    predicted_data2.append(prediction)
```

predicted_data2

```
/tmp/ipykernel_132/3379207600.py:6: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-x))
```

```
[array([ 0.85218166, 0.85912889, 0.48562015, 0.10395347, 0.16195219]),
array([ 0.83926817, 0.85301794, 0.45513369, 0.10659532, 0.15262934]),
array([ 0.83730003, 0.85266752, 0.44238644, 0.11494859, 0.15854952]),
array([ 0.83658267, 0.85258306, 0.43903317, 0.1169137, 0.15955   ]),
array([ 0.83554749, 0.85240828, 0.43007435, 0.12402292, 0.16488474]),
array([ 0.86781726, 0.88807165, 0.5694343, 0.08364289, 0.15969533]),
array([ 0.82375905, 0.8518473, 0.43182596, 0.10247817, 0.13211627]),
array([ 0.82825663, 0.85223721, 0.44331737, 0.09938394, 0.13367023]),
array([ 0.82513429, 0.85109005, 0.43043831, 0.10581673, 0.13652188]),
array([ 0.83553806, 0.8524119, 0.42997895, 0.12411861, 0.16491905]),
array([ 0.83541107, 0.85240002, 0.42961398, 0.12425602, 0.16489019]),
array([ 0.83761953, 0.85260817, 0.43600329, 0.12187581, 0.16539328]),
array([ 0.83319963, 0.85219355, 0.42330725, 0.12665325, 0.1643939 ]),
(array([ 0.83777767, 0.85241337, 0.42336665, 0.12665325, 0.1643939 ]),
```

b. Timestep = 2

```

predicted_data3 = []
for (data,_) in dataset3:
    prediction = nn3.forward(data)
    predicted_data3.append(prediction)

```

predicted_data3

```

/tmp/ipykernel_132/3379207600.py:6: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-x))

```

```

[array([0.83096473, 0.3544509 , 0.18162076, 0.50725105, 0.79576588]),
 array([0.79249363, 0.28979186, 0.18251972, 0.49560502, 0.77195364]),
 array([0.82645514, 0.34547029, 0.1821167 , 0.50683817, 0.79295352]),
 array([0.7978179 , 0.29827864, 0.18325414, 0.49801266, 0.77511127]),
 array([0.79660901, 0.28895805, 0.18372697, 0.49824433, 0.77303716]),
 array([0.79835614, 0.29002197, 0.18323906, 0.49848592, 0.77384524]),
 array([0.79913811, 0.29044052, 0.1829736 , 0.49855748, 0.77419398]),
 array([0.8329316 , 0.37286063, 0.19288494, 0.49908656, 0.80117004]),
 array([0.78637704, 0.29689149, 0.18444993, 0.49450837, 0.78214287]),
 array([0.7999198 , 0.29115028, 0.18279065, 0.49868943, 0.77459699]),
 array([0.79957951, 0.29075607, 0.18284601, 0.49861427, 0.7744057 ]),
 array([0.80027604, 0.29126659, 0.1826476 , 0.49870642, 0.77474229]),
 array([0.79978167, 0.29090589, 0.18278902, 0.4986414 , 0.77450368]),
 array([0.79946092, 0.2908334 , 0.18292682, 0.49863278, 0.77437893]),
 array([0.79964012, 0.29079985, 0.18282858, 0.49862214, 0.77443486]),

```

c. Timestep = 7

```

predicted_data4 = []
for (data,_) in dataset4:
    prediction = nn4.forward(data)
    predicted_data4.append(prediction)

```

predicted_data4

```

/tmp/ipykernel_132/3379207600.py:6: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-x))

```

```

[array([0.01697855, 0.09918238, 0.23109454, 0.85118677, 0.02228673]),
 array([0.01678952, 0.09892222, 0.23119275, 0.85194037, 0.02230177]),
 array([0.04421295, 0.17230127, 0.35574193, 0.84608338, 0.02477957]),
 array([0.01612846, 0.10383888, 0.24748674, 0.84375431, 0.02253283]),
 array([0.01754053, 0.09533687, 0.21858661, 0.85788901, 0.02211236]),
 array([0.01751958, 0.09373238, 0.21431873, 0.86086366, 0.02206338]),
 array([0.01754918, 0.09371341, 0.21413057, 0.8607852 , 0.0220565 ]),
 array([0.01752945, 0.0937841 , 0.21439919, 0.86069743, 0.02206148]),
 array([0.01775209, 0.09421509, 0.21502024, 0.86022784, 0.02206956]),
 array([0.02058337, 0.0765117 , 0.15921706, 0.88973389, 0.02117605]),
 array([0.02107632, 0.07730594, 0.16047205, 0.88888305, 0.02119851]),
 array([0.02177716 , 0.0788602 , 0.16371993, 0.88794543, 0.02128678]),
 array([0.03066947, 0.10259087, 0.21940831, 0.87889309, 0.02278349]),
 array([0.0206103 , 0.0811874 , 0.17193104, 0.88268852, 0.02139776]),
 array([0.01005323, 0.08000630, 0.17016790, 0.88064023, 0.02126025]),

```

Kesimpulan Eksperimen

Program yang dibuat dapat menyimpan serta me-load model dengan baik. Hasil Forward Propagation kurang baik untuk memprediksi data berapapun timestep yang digunakan. Hal tersebut dikarenakan bobot pada model diinisiasi secara acak sehingga berapapun timestep yang digunakan tidak akan menghasilkan nilai yang baik karena tidak ada pembelajaran. Jika backward propagation sudah diimplementasikan, bobot baru bisa diupdate dan memungkinkan untuk mencapai nilai prediksi yang baik. Akan tetapi dapat dilihat bahwa program telah dapat menjalankan forward propagation untuk LSTM dengan baik yang ditandai dengan dapatnya berjalan program tanpa adanya error.

Pembagian Tugas

NIM	Nama	Tugas
13520015	Jaya Mangalo	LSTMLayer, Laporan, Video
13520133	Jevant Jedidia	Experiment, Laporan, Video
13520136	Vincent Christian	Sequential Model, Laporan, Video

Lampiran

Link Github: <https://github.com/JayaMangalo/LSTM-CNN-Implementation>

Link Penjelasan Video: <https://youtu.be/w6IL1STW6Eg>

Link Google Colab:

<https://colab.research.google.com/drive/1ovIoUrjOjWaZ8XBMXG1HqiQLPMTafF-O#scrollTo=I2ottXk1vZfR>