

---

# **Dandelion Documentation**

***Release 0.14.4***

**David Leon (Dawei Leng)**

**Apr 19, 2018**



---

## Contents

---

<b>1</b>	<b>Featuring</b>	<b>3</b>
<b>2</b>	<b>Why Another DL Framework</b>	<b>5</b>
<b>3</b>	<b>Project Layout</b>	<b>7</b>



A quite light weight deep learning framework, on top of Theano, offering better balance between flexibility and abstraction



- **Aiming to offer better balance between flexibility and abstraction.**
  - Easy to use and extend, support for any neural network structure.
  - Loose coupling, each part of the framework can be modified independently.
- **More like a handy library of deep learning modules.**
  - Common modules such as CNN, LSTM, GRU, Dense, and common optimization methods such as SGD, Adam, Adadelta, Rmsprop are ready out-of-the-box.
- **Plug & play, operating directly on Theano tensors, no upper abstraction applied.**
  - Unlike previous frameworks like Keras, Lasagne, etc., Dandelion operates directly on tensors instead of layer abstractions, making it quite easy to plug in 3rd part defined deep learning modules (layer defined by Keras/Lasagne) or vice versa.





---

### Why Another DL Framework

---

- The reason is more about the lack of flexibility for existing DL frameworks, such as Keras, Lasagne, Blocks, etc.
- By “flexibility”, we means whether it is easy to modify or extend the framework.
  - The famous DL framework Keras is designed to be beginner-friendly oriented, at the cost of being quite hard to modify.
  - Compared to Keras, another less-famous framework Lasagne provides more flexibility. It’s easier to write your own layer by Lasagne for small neural network, however, for complex neural networks it still needs quite manual works because like other existing frameworks, Lasagne operates on abstracted ‘Layer’ class instead of raw tensor variables.



## CHAPTER 3

---

### Project Layout

---

Python Module | Explanation ————— | ————— module | all neural network module definitions functional | operations on tensor with no parameter to be learned initialization | initialization methods for neural network modules activation | definition of all activation functions objective | definition of all loss objectives update | definition of all optimizers util | utility functions