

基础工具集与常用数据集

杨沐昀

语言技术研究中心
哈尔滨工业大学

基础工具集

- tiktoken 子词切分工具
- NLTK
- LTP
- PyTorch

常用数据集

- Wikipedia
- Common Crawl

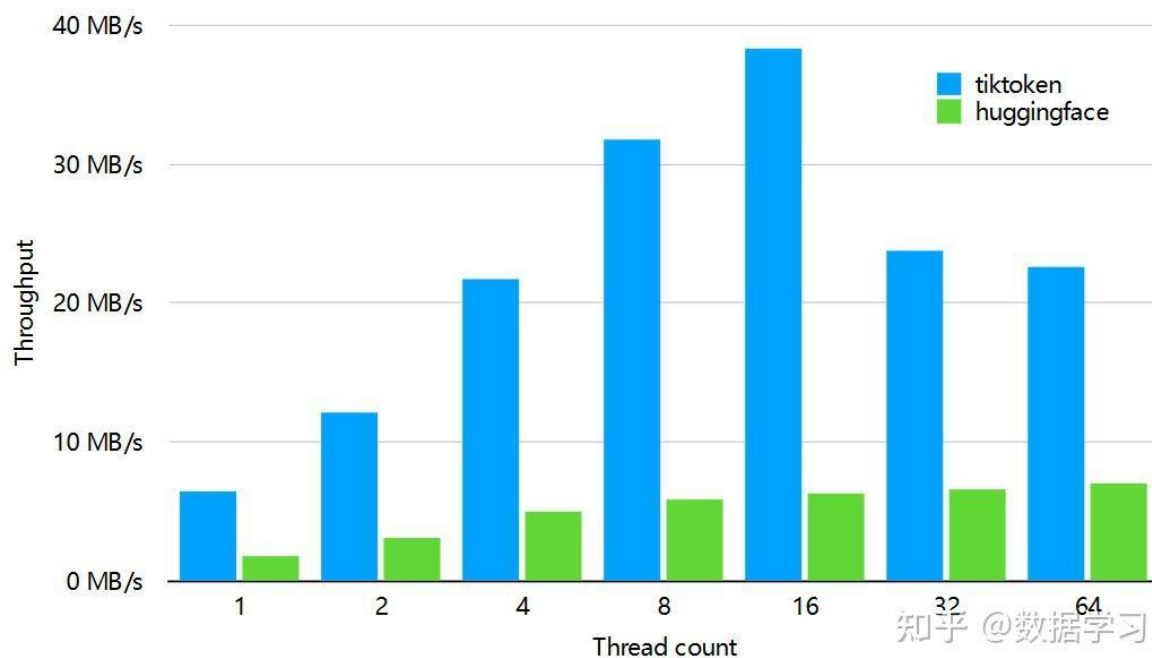
tiktoken 子词切分工具

❑ tiktoken 由 OpenAI 开发的开源子词切分工具，高效地实现了 BPE (Byte Pair Encoding) 算法，能处理各种自然语言文本

❑ BPE用于GPT-2、RoBERTa、XLM、FlauBERT等语言模型中

❑ tiktoken 是GPT-2中使用的BPE算法实现，并以Python库的形式开源

❑ 相比较HuggingFace的tokenizer，其速度提升了好几倍



处理1GB数据的效率对比
(横轴为并行数)

❑ tiktoken 提供了 OpenAI 系列 GPT 模型所使用的三类词表

词表名	模型名	词表大小
cl100k_base	gpt-4, gpt-3.5-turbo, text-embedding-ada-002	10 万
p50k_base	Codex 模型, text-davinci-002, text-davinci-003	5 万
r50k_base (or gpt2)	GPT-3 模型	5 万

❑ 加载词表: `enc = tiktoken.get_encoding("cl100k_base")`

❑ 编码: `enc.encode("Hello tiktoken!")`

❑ 解码: `enc.decode([9906, 87272, 5963, 0])`

用 tiktoken 处理中文

❑ 对中文字符串进行编码

- ▣ `>>> enc.encode("我爱北京天安门! ")`

- ▣ `[37046, 76207, 109, 70090, 36827, 51385, 65789, 6447]`

❑ 使用 `decode_single_token_bytes` 对每个 token 进行解码，会输出字节码

- ▣ `>>> [enc.decode_single_token_bytes(i) for i in [37046, 76207, 109, 70090, 36827, 51385, 65789, 6447]]`

- ▣ `[b'\xe6\x88\x91', b'\xe7\x88', b'\xb1', b'\xe5\x8c\x97\xe4\xba\xac', b'\xe5\xa4\xa9', b'\xe5\xae\x89', b'\xe9\x97\xa8', b'\xef\xbc\x81']`

❑ 使用字符串的 `decode` 方法对每个字节码进行解码，以输出相应的中文字符

- ▣ `>>> [enc.decode_single_token_bytes(i).decode(errors='ignore') for i in [37046, 76207, 109, 70090, 36827, 51385, 65789, 6447]]`

- ▣ `['我', ' ', ' ', '北京', '天', '安', '门', '!', ' ']`

- ▣ 为什么输出两个空字符串？

□ Natural Language Toolkit

□ <https://www.nltk.org/>

□ 多种语料库和词典资源

□ 生文本、PennTreebank样例

□ WordNet

□ 基本的自然语言处理工具集

□ 分句

□ 词元解析

□ 词性标注

□ 句法分析

□ 更多英文自然语言处理工具集

□ CoreNLP、spaCy等



<https://data-flair.training/blogs/nltk-python-tutorial/>

语言技术平台 (Language Technology Platform, LTP)

<http://ltp.ai>

高效、高精度中文自然语言处理基础平台

中文词法、句法、语义分析等6项自然语言处理核心技术

2020年7月发布4.0版本

基于预训练模型

多任务学习机制

安装

\$ pip install ltp



在线演示
直观了解语言技术平台能为你做什么

国务院总理李克强调研上海外高桥时提出，支持上海积极探索新机制。

句子视图 篇章视图 XML视图

☒ 词性标注 ☒ 命名实体 ☒ 句法分析 ☒ 语义角色标注 ☒ 语义依存分析

段落1句子1: 国务院总理李克强调研上海外高桥时提出，支持上海积极探索新机制。

标签释义

Tag	关系类型	Description
Agt	施事关系	Agent
Dir	趋向角色	Direction
ePurp	目的关系	event Purpose
Feat	描写角色	Description
Mann	方式角色	Manner
mPunc	标点标记	Punctuation Marker
mTime	时间标记	Time
Nmod	名字修饰角色	Name-modifier
Prod	成事关系	Product
Root	根节点	Root

LTP调用示例

```
>>> sentences = ltp.sent_split(["南京市长江大桥。", "汤姆生病了。他去了医院。"])  
# 分句  
>>> print(sentences)  
['南京市长江大桥。', '汤姆生病了。', '他去了医院。']  
>>> segment, hidden = ltp.seg(sentences)  
>>> print(segment)  
[['南京市', '长江', '大桥', '。'], ['汤姆', '生病', '了', '。'], ['他', '去', '了', '医院', '。']]  
>>> pos_tags = ltp.pos(hidden) # 词性标注  
>>> print(pos_tags) # 词性标注的结果为每个词所对应的词性, LTP使用的词性标记集与  
# NLTK不尽相同, 但基本大同小异  
[['ns', 'ns', 'n', 'wp'], ['nh', 'v', 'u', 'wp'], ['r', 'v', 'u', 'n', 'wp']]
```


□ Facebook开发的开源深度学习库

□ <https://pytorch.org/>

□ 基于张量（Tensor）的数学运算工具包

□ GPU加速的张量计算

□ 自动进行微分计算

□ PyTorch的优势

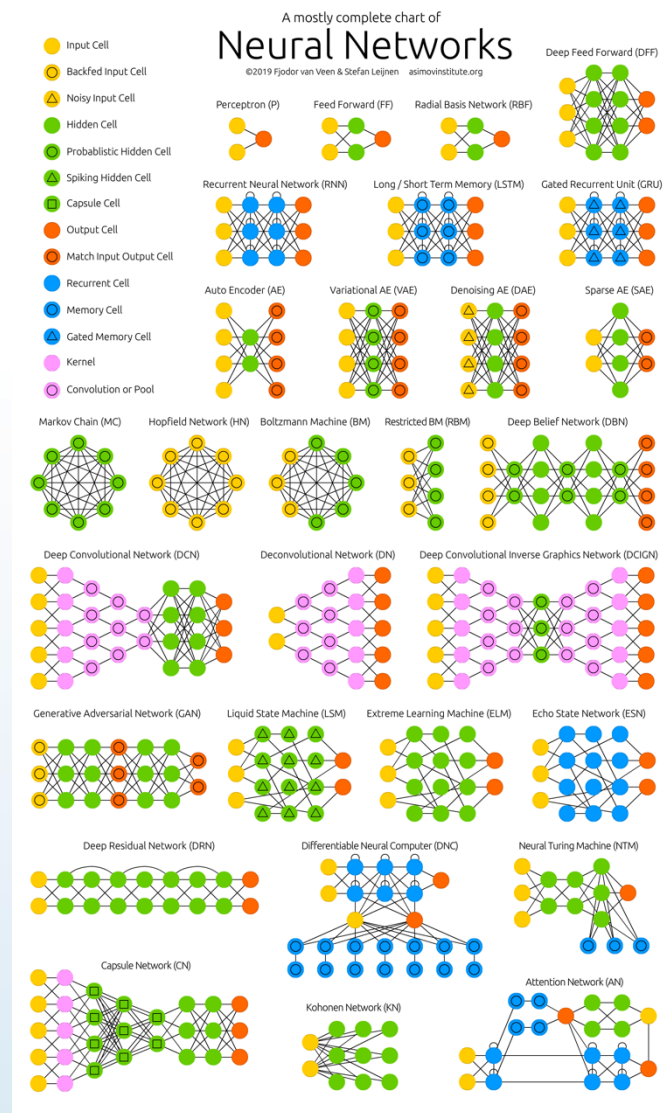
□ 框架简洁

□ 入门简单，容易上手

□ 支持动态神经网络构建

□ 与Python语言无缝结合

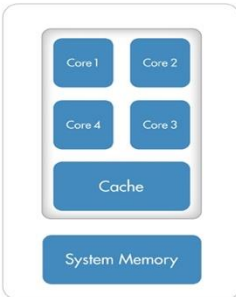
□ Debug方便



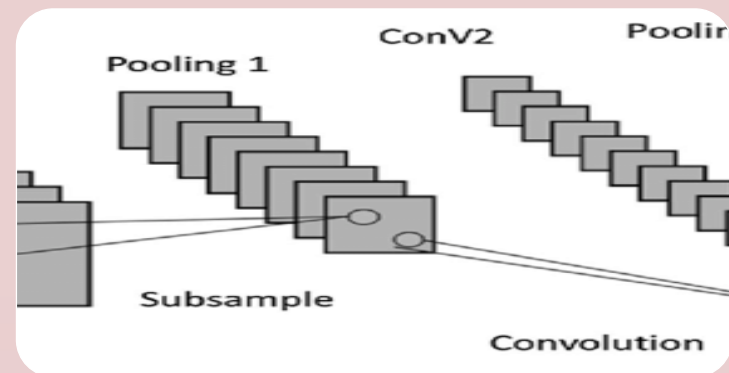
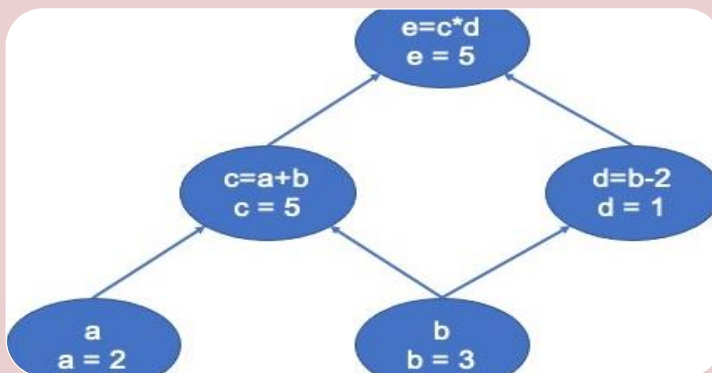
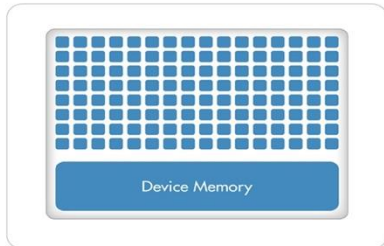
<https://www.asimovinstitute.org/neural-network-zoo/>

PyTorch的三要素

CPU (Multiple Cores)



GPU (Hundreds of Cores)



张量 (Tensor)

- 数据存储
- 支持GPU

表达式 (Expression)

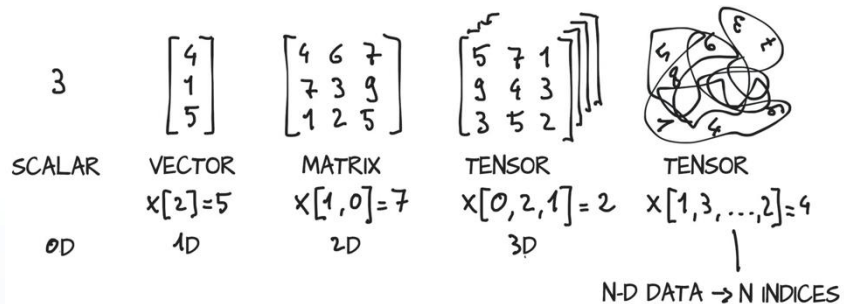
- 数据运算
- 自动微分计算

模块 (Module)

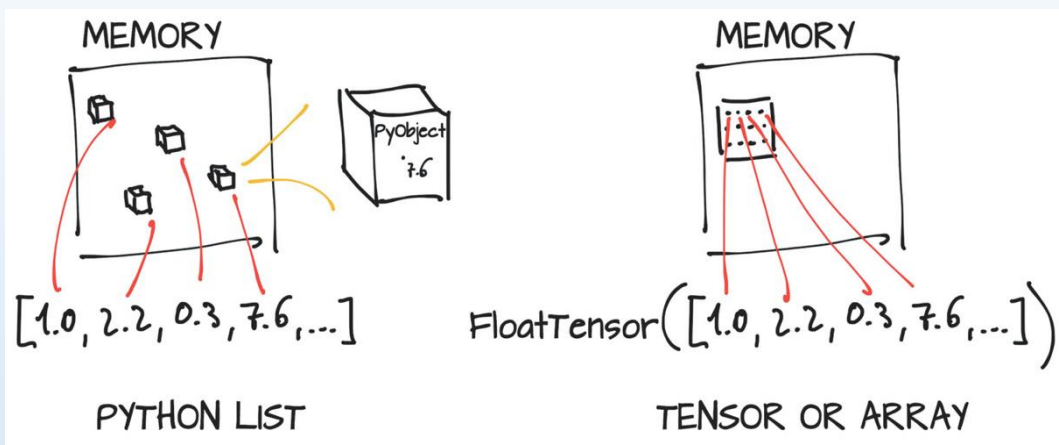
- 神经网络层
- 支持自定义

张量 (Tensor)

□ 又称为多维数组

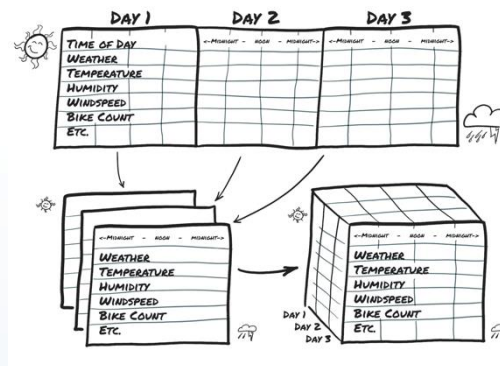


□ 与Python列表的区别

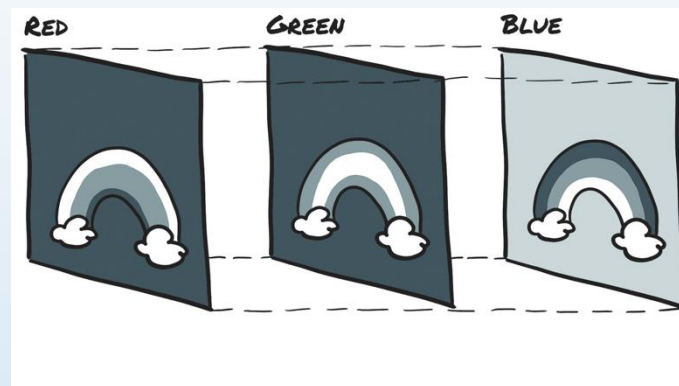


□ 可表示现实世界中的各种数据

□ 表格、时间序列



□ (彩色) 图像



`batch = torch.zeros(100, 3, 256, 256, dtype=torch.uint8)`

PyTorch作为张量库

□支持各种张量操作（类似NumPy）

- 创建张量
- 索引、切片

□支持GPU

- 快!

$$M * M * M \quad M \in \mathbb{R}^{1000 \times 1000}$$

Numpy

```
In [2]: M = numpy.random.randn(1000,1000)
In [3]: timeit -n 500 M.dot(M).dot(M)
500 loops, best of 3: 30.7 ms per loop
```

PyTorch

```
In [4]: N = torch.randn(1000,1000).cuda()
In [5]: timeit -n 500 N.mm(N).mm(N)
500 loops, best of 3: 474 μs per loop
```

表达式 (Expression)

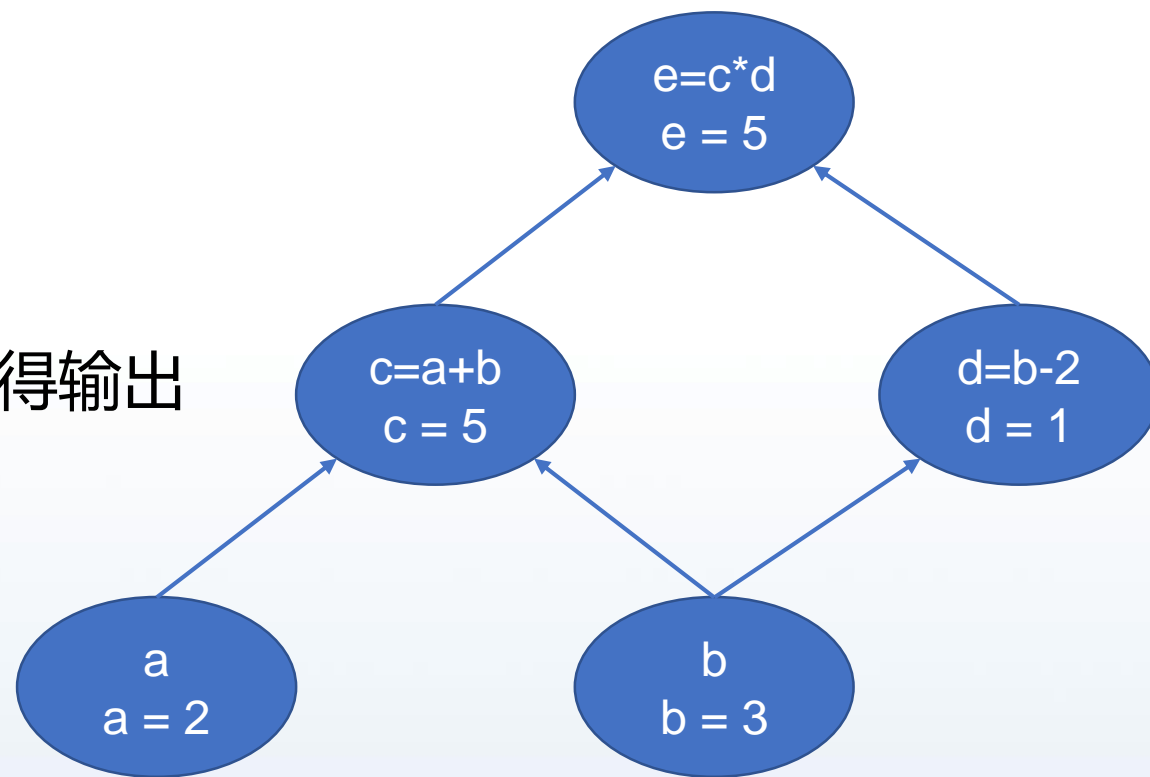
□ 通过**计算图**描述表达式

□ 如: $e = (a + b) * (b - 2)$

□ 计算图中每个节点执行一个运算

□ 前向运算 (Forward) : 根据输入获得输出

```
>>> a = torch.tensor([2.])
>>> b = torch.tensor([3.])
>>> c = a + b
>>> d = b - 2
>>> e = c * d
>>> print(c, d, e)
tensor([5.], grad_fn=<AddBackward0>)
tensor([1.], grad_fn=<SubBackward0>)
tensor([5.], grad_fn=<MulBackward0>)
```



表达式 (Expression)

反向传播 (Back-propagation)

计算输出对每个输入的导数

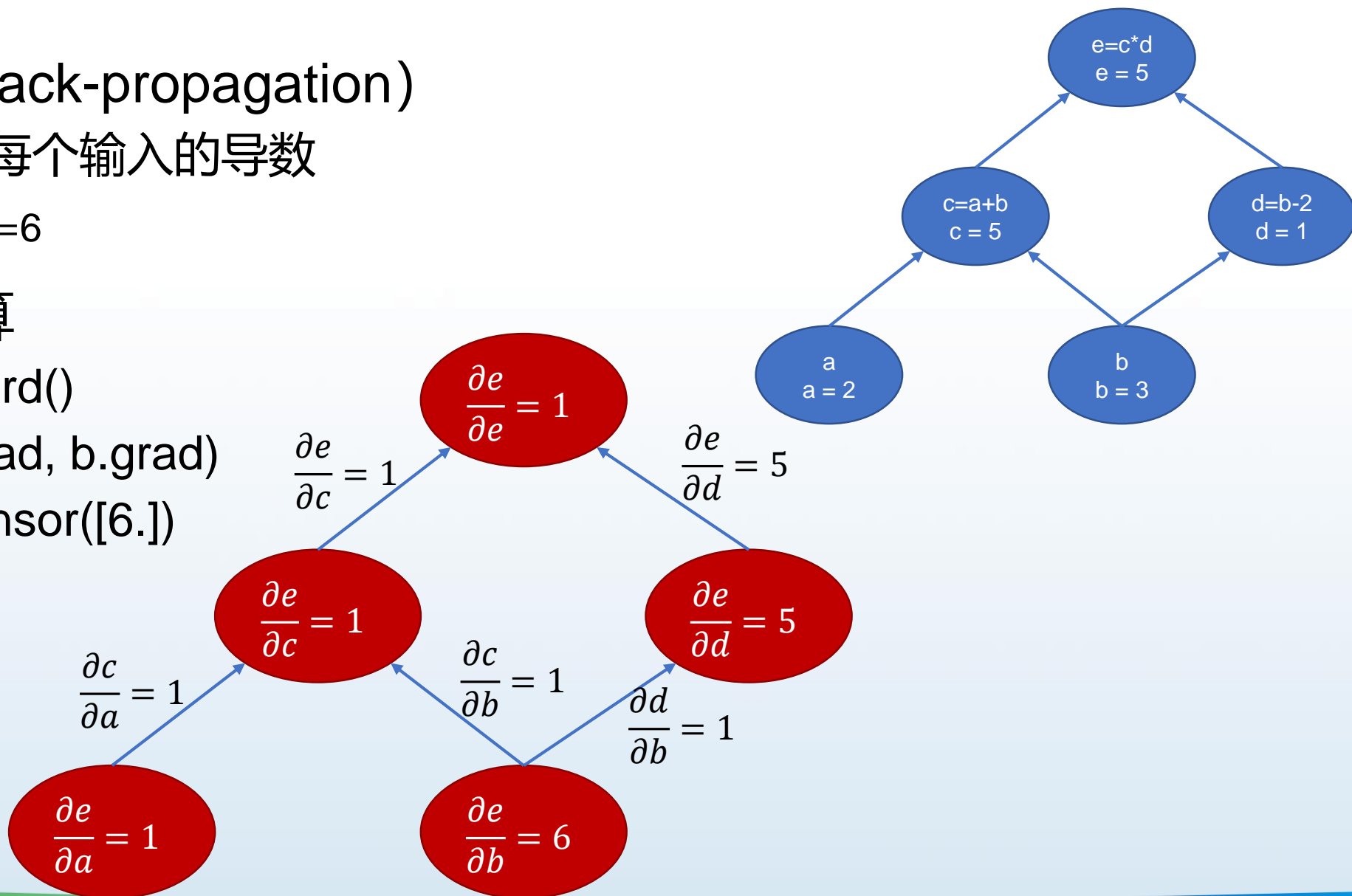
$$\frac{\partial e}{\partial a} = 1, \frac{\partial e}{\partial b} = 6$$

自动微分计算

```
>>> e.backward()
```

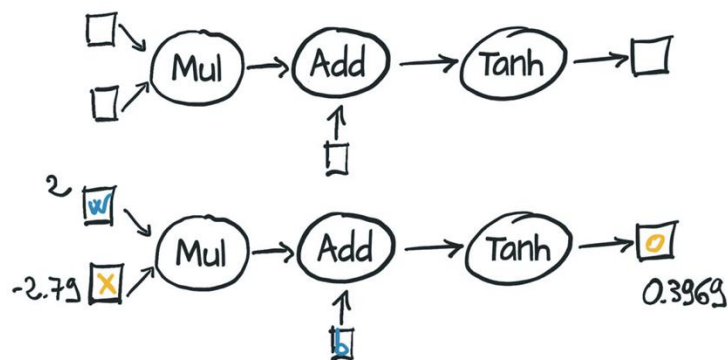
```
>>> print(a.grad, b.grad)
```

```
tensor([1.]) tensor([6.])
```



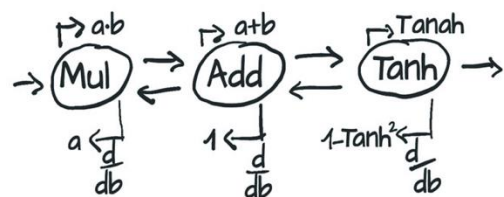
静态图与动态图

STATIC GRAPH $o = \text{Tanh}(wx + b)$



COMPILE
SYMBOLIC
GRAPH

EVALUATE



AUTOMATIC
DIFFERENTIATION

$$\frac{do}{dw} = \frac{d\text{Tanh}}{d(wx+b)} \frac{d(wx+b)}{dw} = (1 - \text{Tanh}^2(wx+b))x$$

$o = \text{Tanh}(wx + b)$

$$x = -2.79$$

$$x_1 = wx = 2 \times (-2.79) = -5.58$$

$$x_2 = x_1 + b = -5.58 + 6 = 0.42$$

$$o = \text{Tanh } x_2 = \text{Tanh } 0.42 = 0.3969 \dots$$

GREEDY EVALUATION
(NO GRAPH)

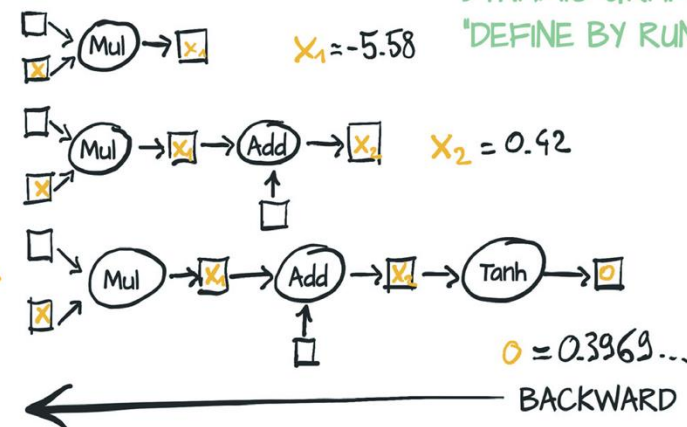
DYNAMIC GRAPH
'DEFINE BY RUN'

STATEMENTS

$$x_1 = wx$$

$$x_2 = x_1 + b$$

$$o = \text{Tanh } x_2$$



模块 (Module)

包名	功能	描述
torch	Tensor / Expression	类似NumPy的张量库，支持GPU以及自动微分
torch.nn	Module	灵活的神经网络库，提供多种神经网络层
torch.optim	Module	多种优化算法
torch.utils	Module	数据集、数据加载等辅助工具

大规模预训练数据集

□ 维基百科 (Wikipedia)

□ 快照 (2020年10月23日) <https://dumps.wikimedia.org/>

文件名	内容	大小/MB
zhwiki-latest-abstract.xml.gz	所有词条摘要	≈ 147
zhwiki-latest-all-titles.gz	所有词条标题	≈ 33
zhwiki-latest-page.sql.gz	所有词条标题及摘要	≈ 204
zhwiki-latest-pagelinks.sql.gz	所有词条外链	≈ 890
zhwiki-latest-pages-articles.xml.bz2	所有词条正文	≈ 1,952

□ 处理流程

□ 纯文本抽取 → 中文繁简转换 → 数据清洗

□ Common Crawl数据 (<https://commoncrawl.org/>)

□ 大规模网络爬虫数据集

□ 使用Facebook的CC-Net工具下载和处理

□ https://github.com/facebookresearch/cc_net

更大规模的语料库集合

□ HuggingFace Datasets

□ 特点

- **数据集数目多**：截至 2023 年 9 月，共收录了近 200 种语言的 6 万个数据集，涵盖了文本分类、机器翻译和阅读理解等众多自然语言处理任务。之所以能有如此多的数据，主要依赖于社区的贡献，任何用户都可以共享相关的数据集。除了支持用户可以直接使用这些公开的数据集，还支持其方便地调用自己私有的数据集
- **兼容性好**：可以直接被 PyTorch、TensorFlow 等深度学习框架，以及 pandas、NumPy 等数据处理工具调用，同时支持 CSV、JSON 等数据格式的读取，并提供了丰富、灵活的调用接口和数据处理接口
- **数据读取效率高**：在仅占用少量内存的条件下，高速地读取大量的数据
- **丰富的评价方法**：针对不同的数据集提供了更有针对性的评价方法

谢谢!



语言技术紫丁香

微信扫描二维码，关注我的公众号

