

深度学习早期总结的知识

朱梦

初稿于 2024 年 9 月 17 日，修改于 2025-05-10

1. 卷积与池化

pytorch:

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \cdot padding - kernel_size}{stride} \right\rfloor + 1 \quad (1)$$

tensorflow:

$$H_{out} = \begin{cases} \left\lceil \frac{H_{in}}{stride} \right\rceil, & \text{If padding is 'same',} \\ \left\lceil \frac{H_{in} - kernel_size + 1}{stride} \right\rceil, & \text{If padding is 'valid'.} \end{cases} \quad (2)$$

2. Warning

Warning1。BatchNormalization 和 Dropout 等层 (模块) 在训练和预测会有不同的行为。

- (1) **pytorch:** 通过 `model.train()` 和 `model.eval()` 控制;
- (2) **tf1.0:** 通过 `model.train_on_batch()` 和 `model.predict_on_batch()` 控制;
- (3) **tf2.0:** 通过 `model(x, training=True)` 和 `model(x, training=False)` 控制;

Warning2。当数据集的数量不能整除 `batch_size`，最好丢掉最后一批数据。

Warning3。对训练集使用数据增强时，要注意会改变数据集分布的增强操作，比如：图像增强中的 `RandomBrightness`。

3. PyTorch 小工具

- (1) `netron` 和 `tensorwatch`，可视乎模型结构图;
- (2) `Python-OpCounter` 计算模型复杂度;
- (3) `visdom` 类似于 `tensorboard`。

4. Python 高效开发 API

- (1) `from pathlib import Path`

- (2) `from pprint import pprint`
- (3) `setattr, getattr, hasattr`
- (4) 定义回调函数: `__call__`
- (5) 定义可迭代对象: `__getitem__` 或 `__iter__`
- (6) 定义迭代器: `__iter__` 和 `__next__`
- (7) 定义生成器: `yield`

5. 下采样算子

下采样的目的就是增大感受野。

- (1) 最大值池化或均值池化, 池化核一般取 2, 步幅一般等于池化核。对旋转产生的位置微小变化具有鲁棒性, 一般用于图像领域, 在 NLP 领域效果不好。
- (2) 跨步卷积, 卷积核一般取 3 或 5, 步幅一般取 2, 能保持相对位置的不变性, 即可以用于图像领域, 也可以用于 NLP 领域。

6. 机器学习库

- (1) `scikit-learn`: `pip install scikit-learn; from sklearn import *`;
- (2) `mglearn`: `pip install mglearn; from mglearn import *`;

7. 图像处理库

- (1) `opencv`: 以 BGR 读入, 以 RGB 写出;
- (2) `scikit-image`: `pip install scikit-image; from skimage import *`。以 RGB 读入, 以 RGB 写出;
- (3) `pillow`: 以 RGB 读入, 以 RGB 写出。