

2023-2024学年第一学期本科生课程

# 《神经网络与深度学习》

## 课程实验作业(二)

---

**主讲人：戴金晟**(副教授，博士生导师)

daijincheng@bupt.edu.cn

神经网络与深度学习课程组



北京邮电大学  
Beijing University of Posts and Telecommunications

# Food-11 数据集

❑ food-11数据集是深度学习中常用的一类数据集，其中包含11类食物的图片



该数据集存在的问题:

- ① 标签需要自行提取
- ② 图片大小不一致
- ③ 图片数量有限，容易过拟合

- 数据类别: Bread, Dairy product, Dessert, Egg, Fried food, Meat, Noodles/Pasta, Rice, Seafood, Soup, and Vegetable/Fruit
- 数据集中包含9866张训练集, 3430张验证集, 3347张训练集
- training和validation目录下的命名格式为 “[类别]\_[编号].jpg”

任务点

根据已有数据集, 对 training 和 validation 进行处理, 构造训练数据集

# 数据增强 (Augmentation Implementation)

- ❑ 为了保证数据的训练性能，请利用 `torchvision.transforms` 设计属于你的 `train_transform`，`train_transform` 模版已给出
  - 你所设计的 `train_transform` 需要指定你所选择的特征图大小，不同特征图大小会影响实验结果
  - `train_transform` 应至少由五种不同 `transform` 组成



数据增强可以极好地防止过拟合的情况出现，值得说明的是，有些 `transform` 由于其效果与现实相悖，所以在处理中并不适用

任务点

- 1、根据个人情况设计自己的 `train_transform`
  - 2、对自己的 `train_transform` 效果进行可视化展示
- 注意：测试时无需使用 `train_transform`

# 模型选择

- 在搭建模型时，可以在本实验所提供的baseline基础上进行修改，或者是在[torchvision.models](https://pytorch.org/vision/0.13/models.html)中选择已有的网络结构，并在此基础上进行修改

## TORCHVISION.MODELS

The models subpackage contains definitions of models for addressing different tasks, including: image classification, pixelwise semantic segmentation, object detection, instance segmentation, person keypoint detection and video classification.

### • NOTE

Backward compatibility is guaranteed for loading a serialized `state_dict` to the model created using old PyTorch version. On the contrary, loading entire saved models or serialized `ScriptModules` (serialized using older versions of PyTorch) may not preserve the historic behaviour. Refer to the following [documentation](#)

## Classification

The models subpackage contains definitions for the following model architectures for image classification:

- AlexNet
- VGG
- ResNet

**不得使用预训练的权重！**  
若 torchvision 版本小于0.13，  
设置 `pretrained=False`；大于  
0.13，设置 `weights=False`

```
self.CNN = nn.Sequential(
    nn.Conv2d(3, 64, 3, 1, 1),
    nn.BatchNorm2d(64),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0),

    nn.Conv2d(64, 128, 3, 1, 1),
    nn.BatchNorm2d(128),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0),

    nn.Conv2d(128, 256, 3, 1, 1),
    nn.BatchNorm2d(256),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0),

    nn.Conv2d(256, 512, 3, 1, 1),
    nn.BatchNorm2d(512),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0),

    nn.Conv2d(512, 512, 3, 1, 1),
    nn.BatchNorm2d(512),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0),
```

# VGG系列模型

- VGG 系列模型是 AlexNet 的一种改进，其采用连续的几个  $3 \times 3$  的卷积核来代替 AlexNet 中较大的卷积核，如  $11 \times 11$ 、 $7 \times 7$  等

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 LRN	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

## 任务点

- 1、从torchvision.models中调用VGG系列模型，并打印模型结构
- 2、使用该模型对测试集进行预测，并将结果输出到ans\_vgg.csv中

# 你所关心的给分问题

□ 测试集上的准确度将会影响得分（不考虑VGG的准确率）

得分	准确率	建议
1分	0.6	运行基本模型即可
2分	0.7	进行数据增强；延长训练时间
3分	0.8	残差网络；交叉验证；dropout；更长的训练时间；其它你所知道的有用的方式
4分	0.85	