# 工作小结

姜扬扬

# 1. 深度学习基础

- 神经网络 Neural Network
  - 学习NN的基础结构，从二分分类，到2层的浅层NN，再到层数>2的深层NN；
  - 参数学习过程：前向传播计算各层神经元，得到预测结果，计算损失函数；然后应用后向传播，对损失函数求各参数的偏导，用梯度下降更新参数。

# 深度神经网络的优化

- 欠拟合问题：扩大网络规模，延长训练时间；
- 过拟合问题：增加训练数据（数据增强），正则化（L2正则化，Dropout正则化）；
- 加速训练：输入标准化
- 梯度消失/爆炸问题：初始化抑制权重快速上升/下降
- 梯度下降算法：批梯度下降，小批量梯度下降，随机梯度下降
- 优化算法：Momentum, RMSprop, Adam

# 深度神经网络的超参数

- 学习率 α；动量衰减参数β，各层单元数 #hidden units，mini-batch大小；学习率衰减 decay_rate，网络层数 #layers；若使用Adam优化算法，还涉及到β1, β2, ε

- 超参数搜索
  - 随机搜索 random search

- 批归一化 batch normalization：通过远离导数饱和区，缓解了梯度消失问题；通过对z的标准分数做线性操作，避免了激活函数线性过高；同时有一些正则化的效果。

# 2. 卷积神经网络 CNN

- 基本组成
  - 卷积层 filter_size, filters, stride, padding(valid/same)
  - 池化层（不含常规参数） filter_size, stride, max/avg
  - 全连接层：组合特征，进行分类；
- 优点
  - 参数共享，稀疏连接，平移不变。

# 经典CNN网络

- LeNet-5
- AlexNet
- VGG-16

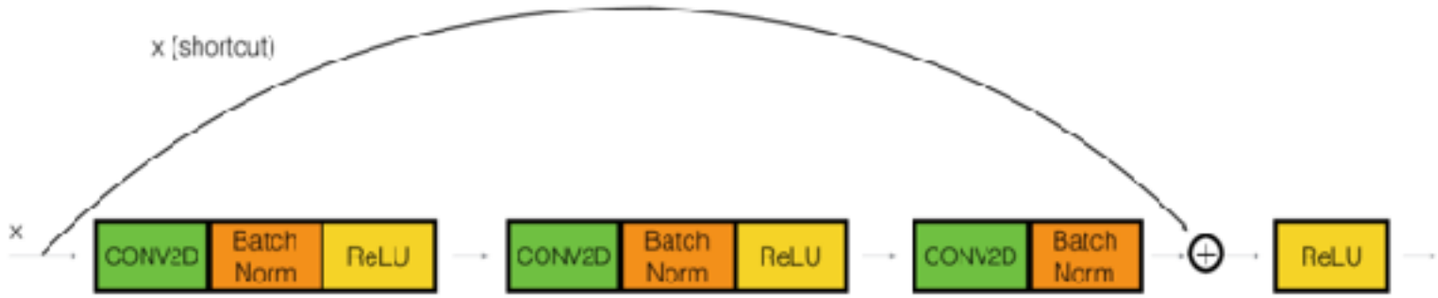
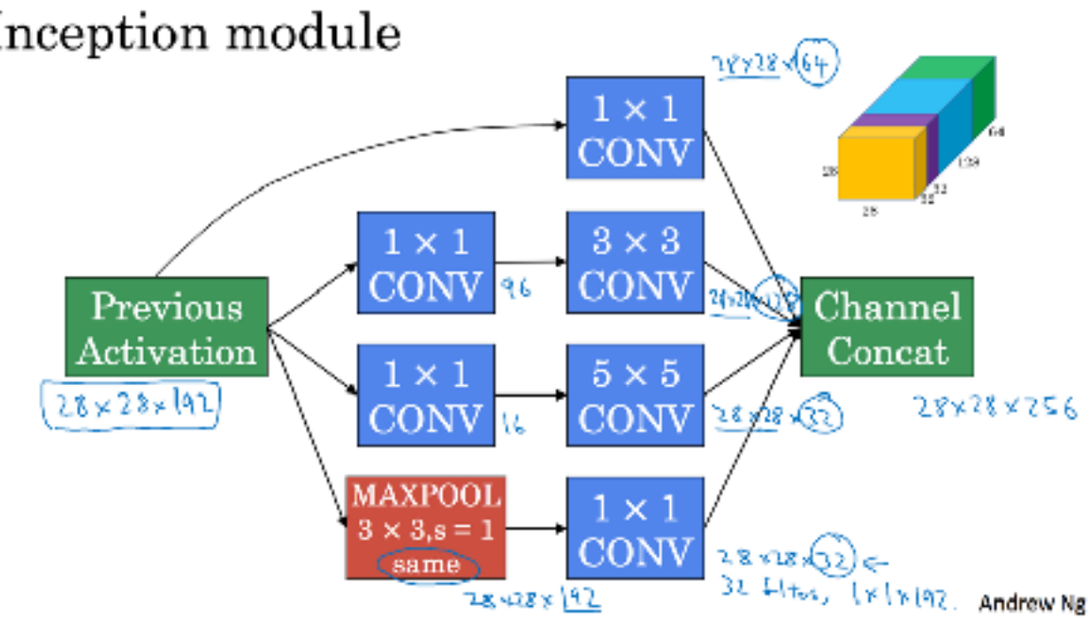



**Figure 4**: **Identity block.** Skip connection "skips over" 3 layers.




- 残差网络 ResNet，搭建捷径 short cut/skip connection；
- Inception，一个模块是多种过滤器的组合，堆叠结果，注意加入瓶颈层降低信道数量。

# 3. 实验部分

- 从头实现二分回归模型和4层的深度NN模型；
- 基于tensorflow实现3层NN模型；
- 基于keras，用简单的NN模型，解决IMDB二分类问题、reuters多分类问题、boston_housing回归问题；
- 基于keras，建立简单的CNN模型解决MNIST识别问题；应用数据增强和dropout优化CNN模型在猫狗识别问题的表现，再尝试用VGG-16模型做特征提取，训练分类器部分。

- 迁移学习
  - 特征提取 feature extraction，复用已有模型的卷积基；
  - 微调 fine-tuning，复用卷积基底部，解冻顶部调节。
- CV中的数据增强，对图像进行变换，包括：镜像，随机剪裁，旋转，局部弯曲，色彩变换等。
- 机器学习的流程为：定义问题，收集数据集；选择衡量指标metrics；确定评估方法；处理数据；开发比随机基准更好的模型；扩大模型规模，采取多轮训练，开发过拟合模型；最后进行数据增强和正则化，调节模型超参数。

# 4. 物体检测

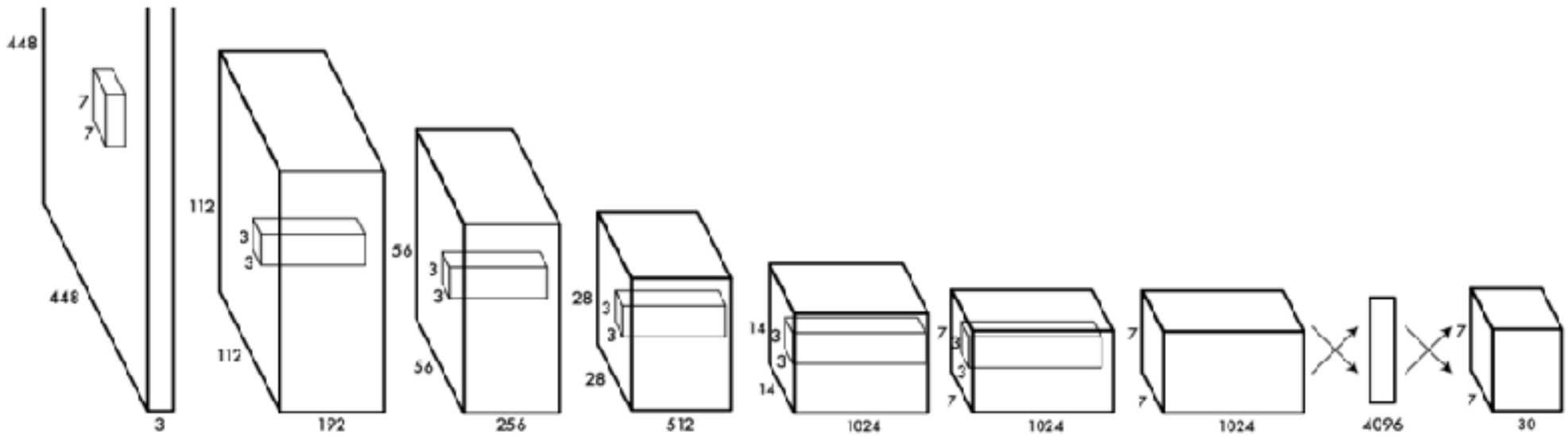
- 物体定位 Object Localisation + 物体分类 Object Classification

- 滑动窗口检测，设置一个窗口，以一定的步长，滑动遍历图像，将当前窗口截取的图像依次输入模型；

- 卷积滑动窗口，将整张图片输入模型，最后输出结果是各个窗口结果的组合，移动步长由中间的池化层决定；

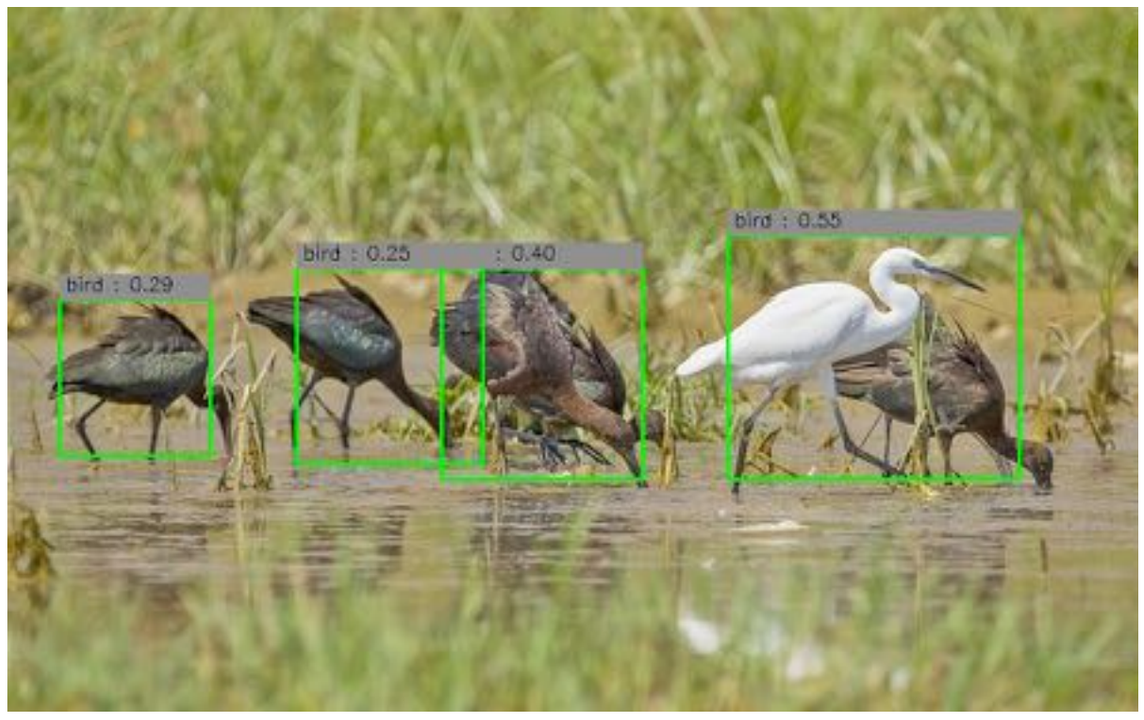
- 但是总的来说，采取滑动窗口的方法得到的边界框不够准确，而且计算开销较大。

# YOLO

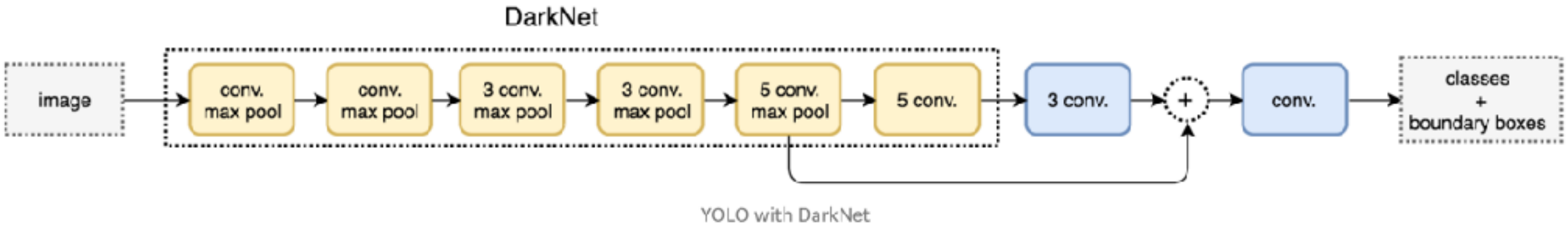



- 24xCONV + 2xFC

- 输入图片划分为 SxS 个格子，每个格子预测 (#boxes x 5) + #classes



# YOLOv2




YOLO with DarkNet

# YOLOv3

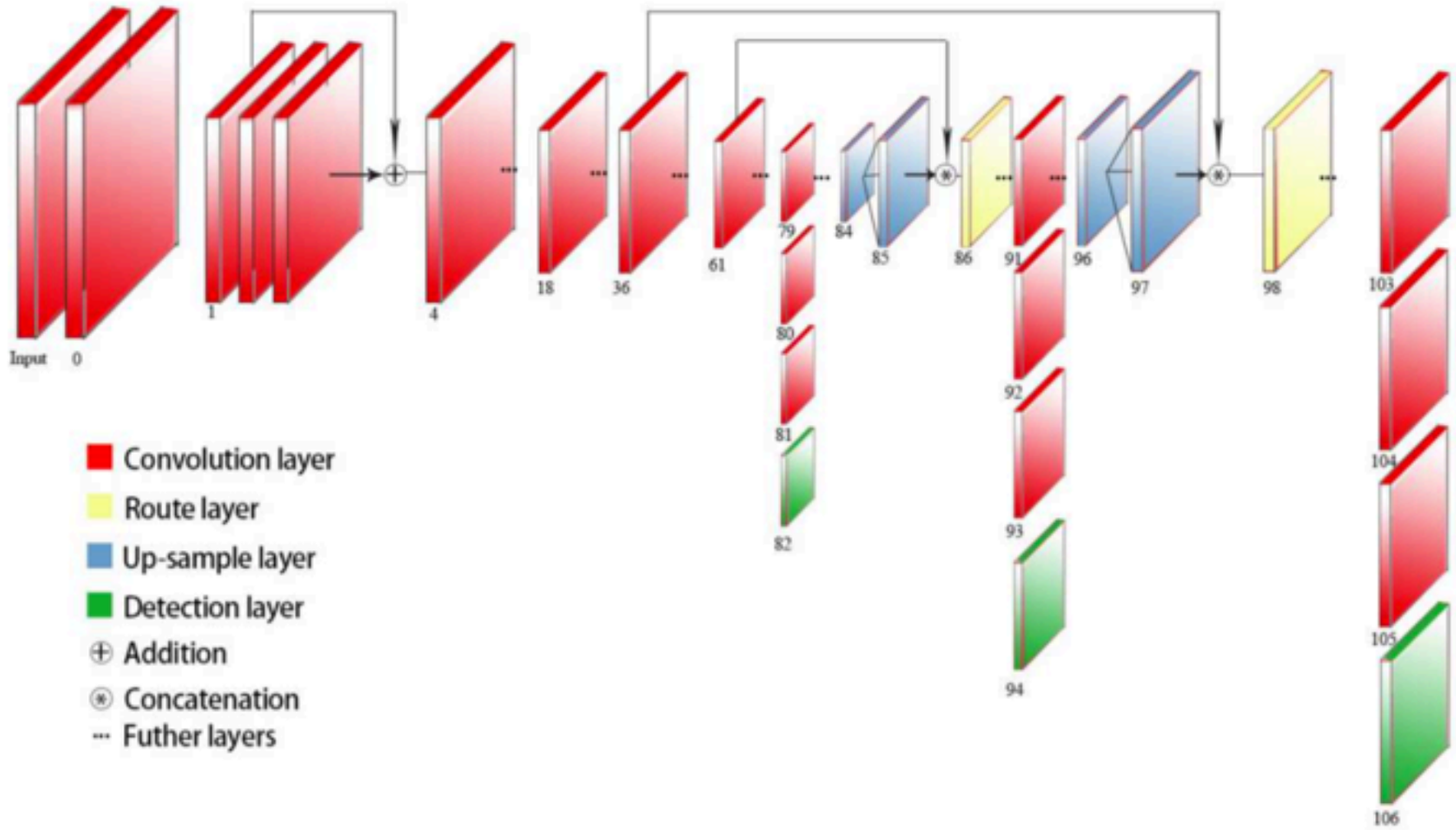
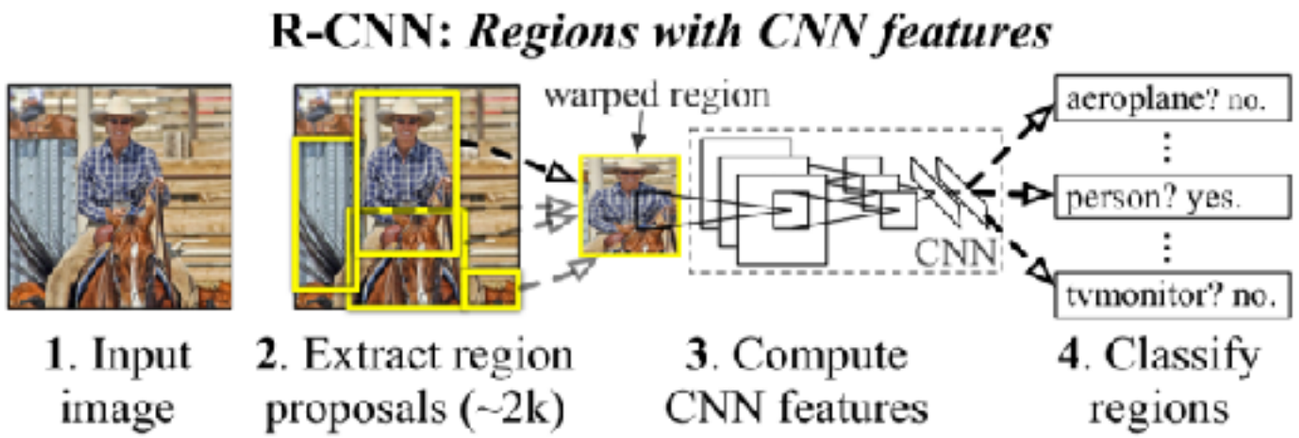
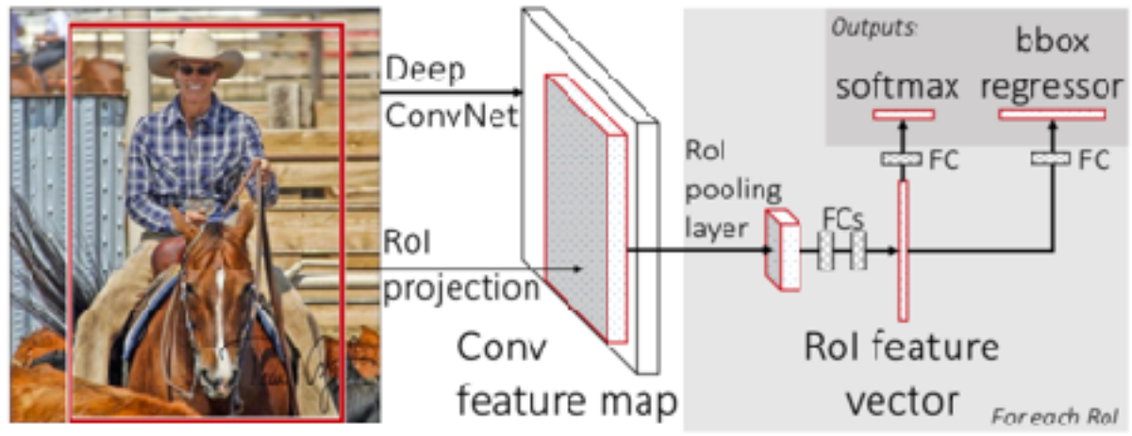



**Figure 3.** Network structure of you only look once (YOLOv3).

# R-CNN




# Fast R-CNN




# Faster R-CNN