8. 计算机视觉I

图像增广、微调、锚框

WU Xiaokun 吴晓堃

xkun.wu [at] gmail

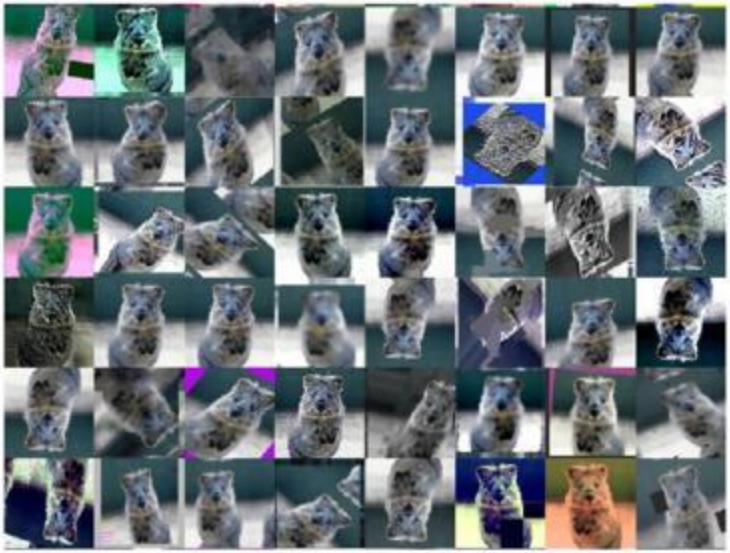
2021/04/11

图像增广

什么是图像增广?

对已有数据集扩充，使其更多样

- 语音：加入背景噪音
- 图像：颜色、形状、噪音



为什么需要图像增广?

Consumer Electronics Show (CES) 展会案例

- 智能售货机：现场演示效果很差
  - 灯光色温
  - 桌面光照反射

- 解决方案：（连夜）收集会场数据
  - 训练新模型
  - 更换新桌布

为什么需要图像增广?

Consumer Electronics Show (CES) 展会案例

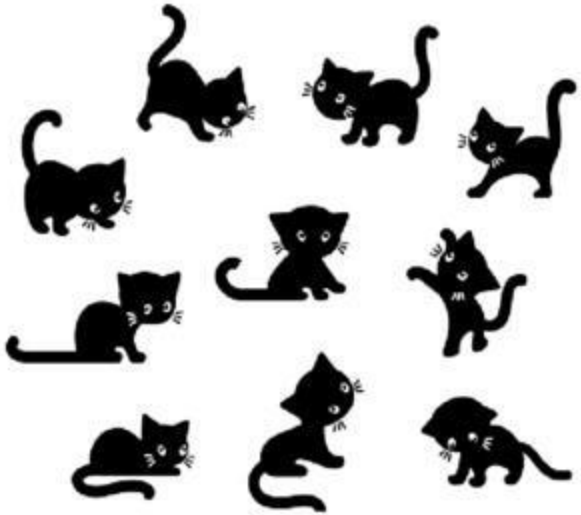
- 智能售货机：现场演示效果很差
  - 灯光色温
  - 桌面光照反射

- 解决方案：（连夜）收集会场数据
  - 训练新模型
  - 更换新桌布

问题：每次展会都提前去训练模型?

图像增广：几何变形

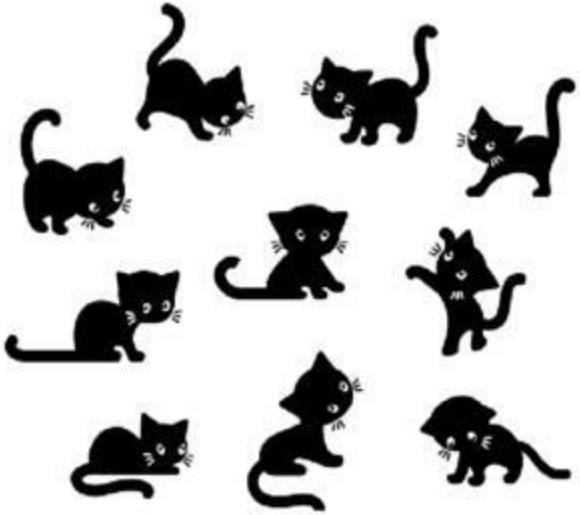
图像内容、位置的微小变形、偏移



- 不同姿态

图像增广：几何变形、剪裁

图像内容、位置的微小变形、偏移

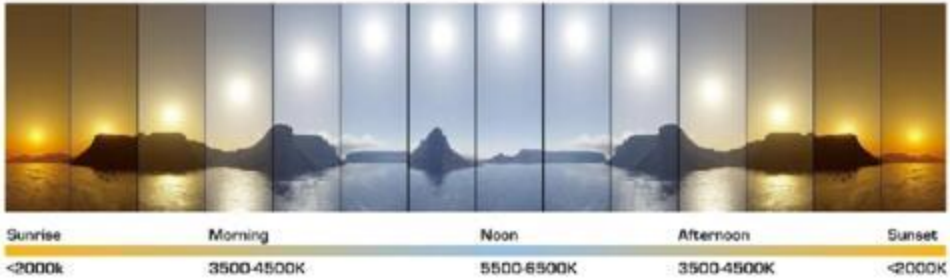


- 不同姿态

- 调整FOV：等价于剪裁

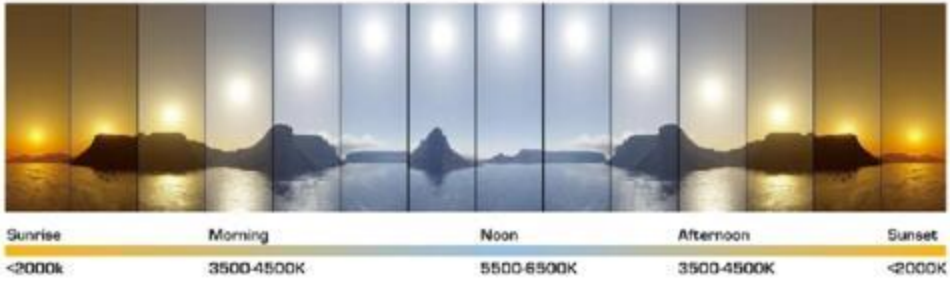
图像增广：摄影参数

色温：暖（黄）、冷（蓝）色调



图像增广：摄影参数

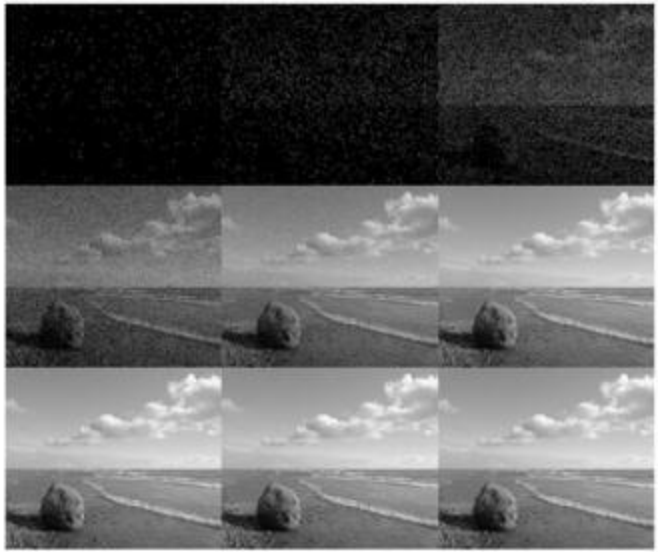
色温：暖（黄）、冷（蓝）色调



曝光：整体明度

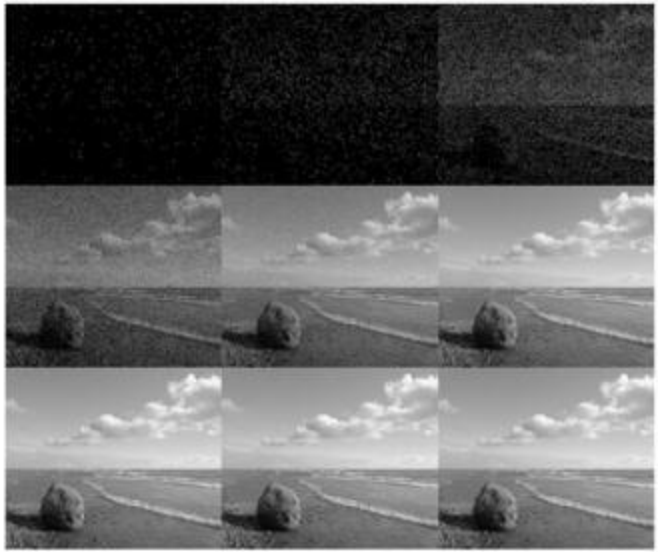
-2EV -1EV 0EV +1EV +2EV

图像增广：硬件



- 噪音：传感器质量、ISO

图像增广：硬件、后期



- 噪音：传感器质量、ISO

- 饱和度：颜色的纯正程度或鲜明程度

图像增广：意义

回顾：数据集大小**间接决定模型效能**

- 数据比模型更复杂：不容易过拟合
- AlexNet 的成功因素之一

图像增广：意义

回顾：数据集大小**间接决定模型效能**

- 数据比模型更复杂：不容易过拟合
- AlexNet 的成功因素之一

图像增广可以降低模型对数据的**敏感性**

- 图像内容、位置的微小变形、偏移
  - 卷积网络：池化有类似作用

图像增广：意义

回顾：数据集大小**间接决定模型效能**

- 数据比模型更复杂：不容易过拟合
- AlexNet 的成功因素之一

图像增广可以降低模型对数据的**敏感性**

- 图像内容、位置的微小变形、偏移
  - 卷积网络：池化有类似作用
- 亮度：曝光程度；颜色：色温、色差

图像增广：意义

回顾：数据集大小**间接决定模型效能**

- 数据比模型更复杂：不容易过拟合
- AlexNet 的成功因素之一

图像增广可以降低模型对数据的**敏感性**

- 图像内容、位置的微小变形、偏移
  - 卷积网络：池化有类似作用
- 亮度：曝光程度；颜色：色温、色差

减小模型对特定特征的依赖：提高模型的**泛化能力**

- 假设固定特征数量：数据信息量少时，特征之间容易相关

图像增广：物理定律

“倒立的凳子”也是凳子

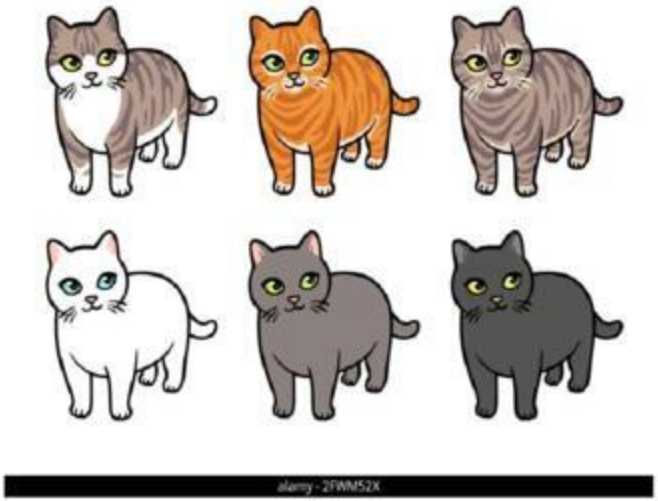
图像增广：物理定律、常识

“倒立的凳子”也是凳子

这是只名贵的猫，因为~~它会倒立~~

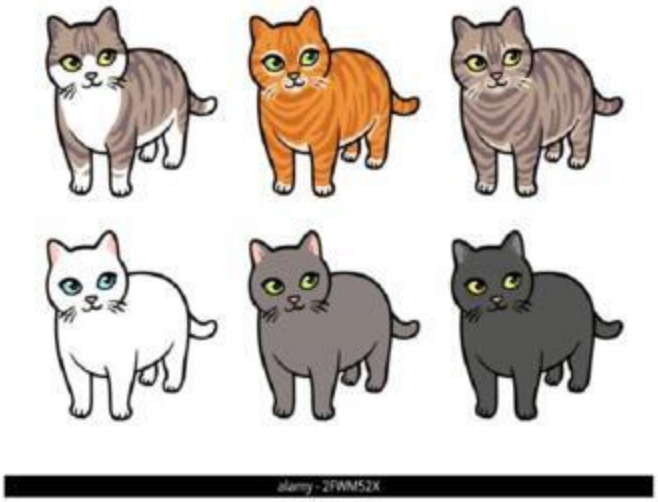
- 它会让你把图片倒过来看

图像增广：生物事实



- 理论上猫只有黑/白/橙、纯色/斑纹的组合

图像增广：生物事实、装饰



- 理论上猫只有黑/白/橙、纯色/斑纹的组合

- 印度洒红节

imgaug

imgaug: https://github.com/aleju/imgaug

实验：图像增广

小结：图像增广

图像增广：变换、变形以获得更多样的数据

- 提高模型的泛化性能

常见操作：仿射变换、翻转、剪裁；色调、饱和度、明度

微调

模型、数据的两难

现实任务可以认为有**无限可能情况**

- 例如猫的图片：稍微调整角度后拍摄

模型、数据的两难

现实任务可以认为有**无限可能情况**

- 例如猫的图片：稍微调整角度后拍摄
- 推论：模型容量需要非常大
  - **训练成本**高；不同任务不能通用

模型、数据的两难

现实任务可以认为有**无限可能情况**

- 例如猫的图片：稍微调整角度后拍摄
- 推论：模型容量需要非常大
  - **训练成本**高；不同任务不能通用
- 推论：数据规模需要非常大
  - **标注成本**高；可用于相关任务；通常公司视为核心资产

	ImageNet	通常	MNIST
样本数	1.2 M	500K	60 K
类别数	1,000	100	10

模型、数据的两难

现实任务可以认为有**无限可能情况**

- 例如猫的图片：稍微调整角度后拍摄
- 推论：模型容量需要非常大
  - **训练成本**高；不同任务不能通用
- 推论：数据规模需要非常大
  - **标注成本**高；可用于相关任务；通常公司视为核心资产

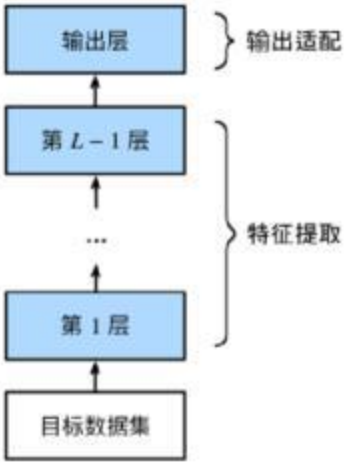
	ImageNet	通常	MNIST
样本数	1.2 M	500K	60 K
类别数	1,000	100	10

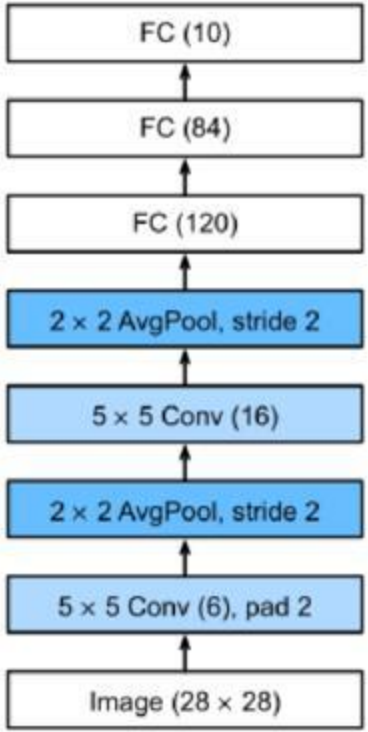
问题：难道模型只能是一次性产品？标注数据不够导致过拟合（需要重新训练）怎么办？

网络架构解析

神经网络可以划分成两个组件

1. **特征提取**：看成自动化特征工程
2. **输出适配**：例如分类器输出概率



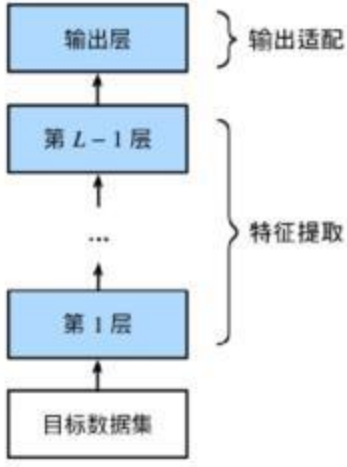


也可以认为是两个**处理阶段**

微调：特征提取

微调的本质：将训练好的模型当作**特征提取器**

- 图像的特征描述大同小异
  - 任务之间可以**共享特征**
- 但不同深度的**特征含义不同**
  - 如何取舍?



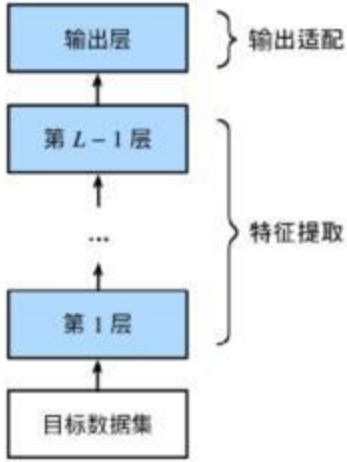

IM..GENET

微调：输出适配

微调的本质：将训练好的模型当作**特征提取器**

- 图像的特征描述大同小异
  - 任务之间可以**共享特征**
- 但不同深度的**特征含义不同**
  - 如何取舍?

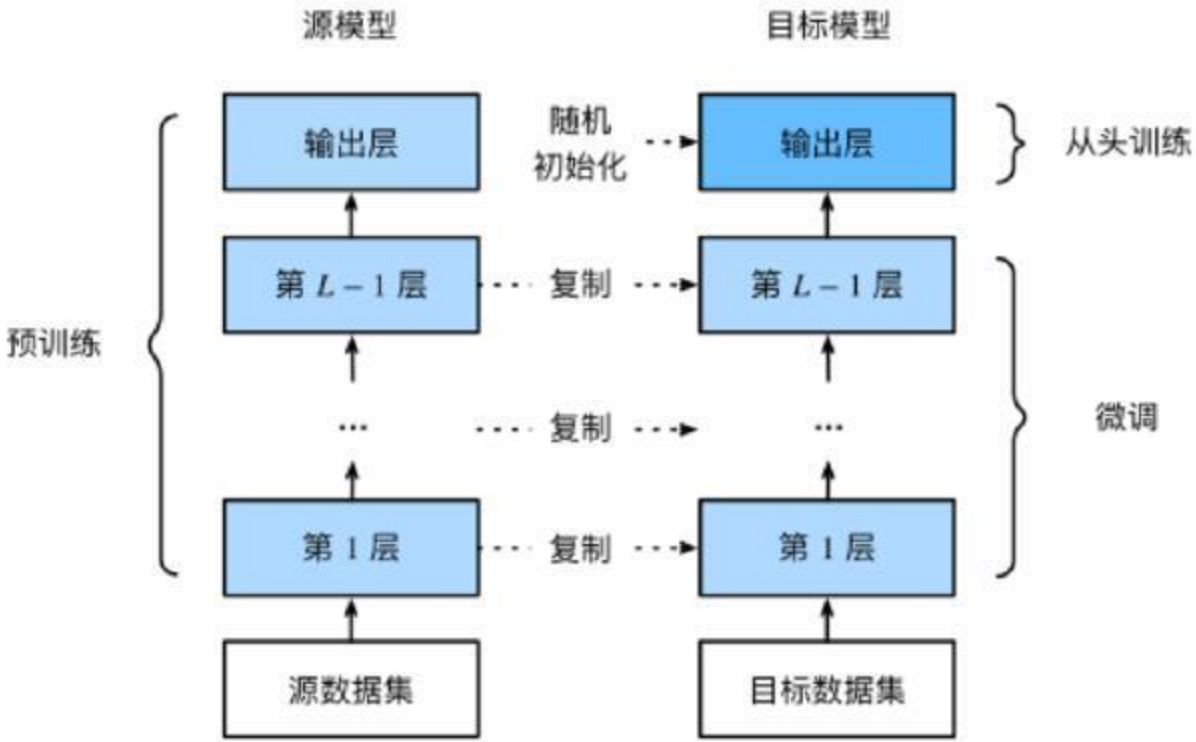
- 输出适配：**重新构造**、替换
  - 任务变化：不再是分类问题
  - 类别数、标签含义都可能变




IM GENET

微调：权重初始化

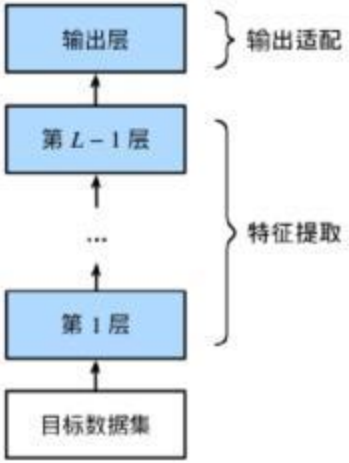
首先看目标模型重用**全部**特征提取器



微调：训练

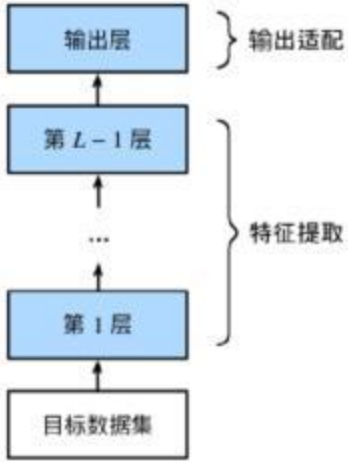
在目标数据集上正常训练

- 需要**更强的正则化**：避免参数剧烈变动
- **小学习率**：已经在最优点附近
- 通常只需更少的数据迭代



微调：训练

在目标数据集上正常训练



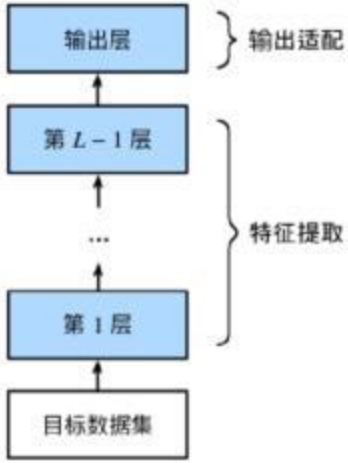
- 需要**更强的正则化**：避免参数剧烈变动
- **小学习率**：已经在最优点附近
- 通常只需更少的数据迭代

比直接训练：**速度更快、精度更高**

- 源数据集、模型远比目标复杂时，效果更好
  - 有可能出现过拟合，但也可能覆盖测试集

微调：训练

在目标数据集上正常训练



- 需要**更强的正则化**：避免参数剧烈变动
- **小学习率**：已经在最优点附近
- 通常只需更少的数据迭代

比直接训练：**速度更快、精度更高**

- 源数据集、模型远比目标复杂时，效果更好
  - 有可能出现过拟合，但也可能覆盖测试集

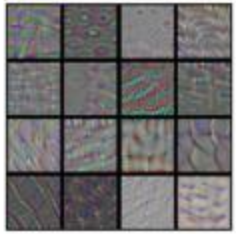
迁移学习 transfer learning：将从源数据集学到的知识迁移到目标数据集

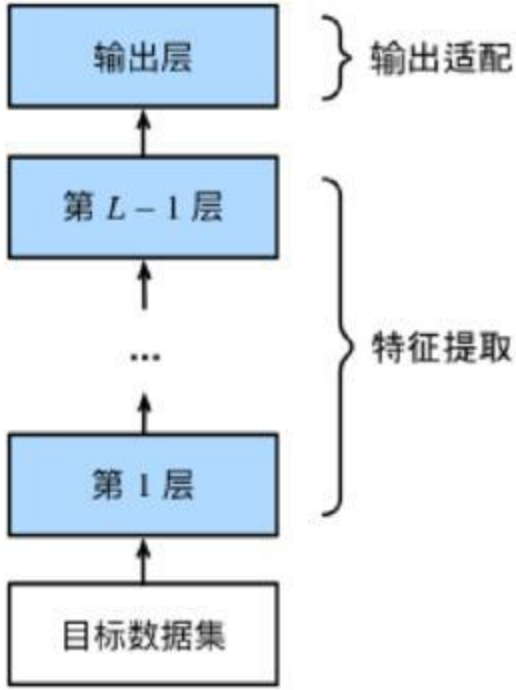
微调：部分重用 I

根据任务的**差异幅度**选择提取层级

后面层：**概括**、**特性**特征

- 观察范围（感受野）更大
- 与任务更相关，必须替换



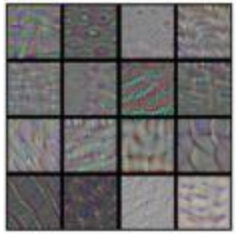


微调：部分重用 II

根据任务的**差异幅度**选择提取层级

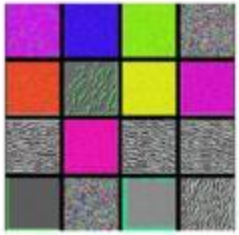
后面层：**概括**、**特性**特征

- 观察范围（感受野）更大
- 与任务更相关，必须替换

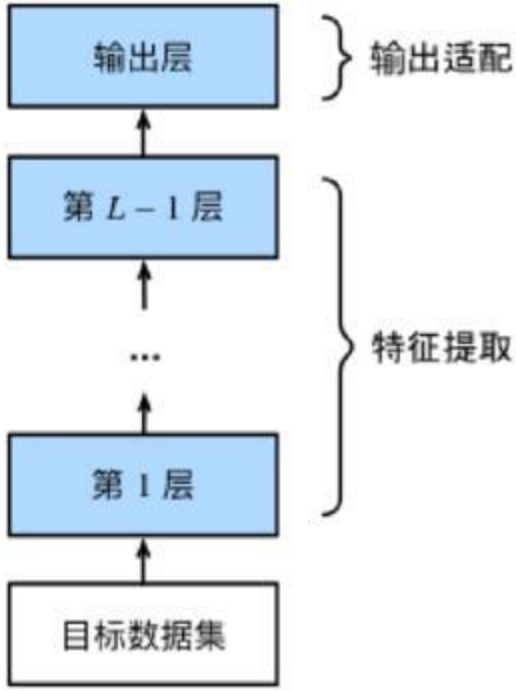


前面层：**低级**、**共性**特征

- 只能观察局部
- 颜色、边缘、形状等



可以固定参数：不参与训练

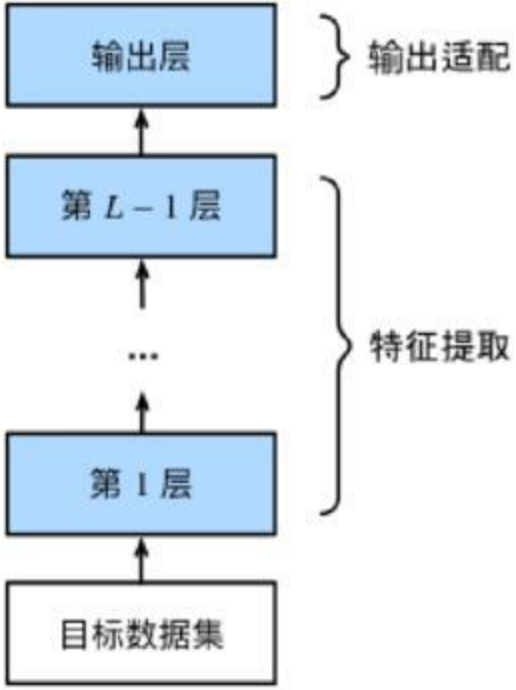


微调：重用分类器

特殊情况：源数据集含目标数据集的部分标签

- 含“赛车”类

- 可以使用**对应标签**提取的特征表示（向量）



实验：微调

小结：微调

- 微调：使用预训练的模型当作特征提取器
  - 替代目标模型的部分模块
  - 层的选取取决于数据差异程度
- 预训练模型的质量很关键
- 通常速度更快、精度更高

实战 Kaggle 比赛：图像分类

图像分类（CIFAR-10）

狗的品种识别（ImageNet Dogs）

目标检测和边界框

图片分类、目标检测

图片**分类**：输出类别标签

- 图像中只有一个目标对象

目标**检测**：输出类别标签、**位置**

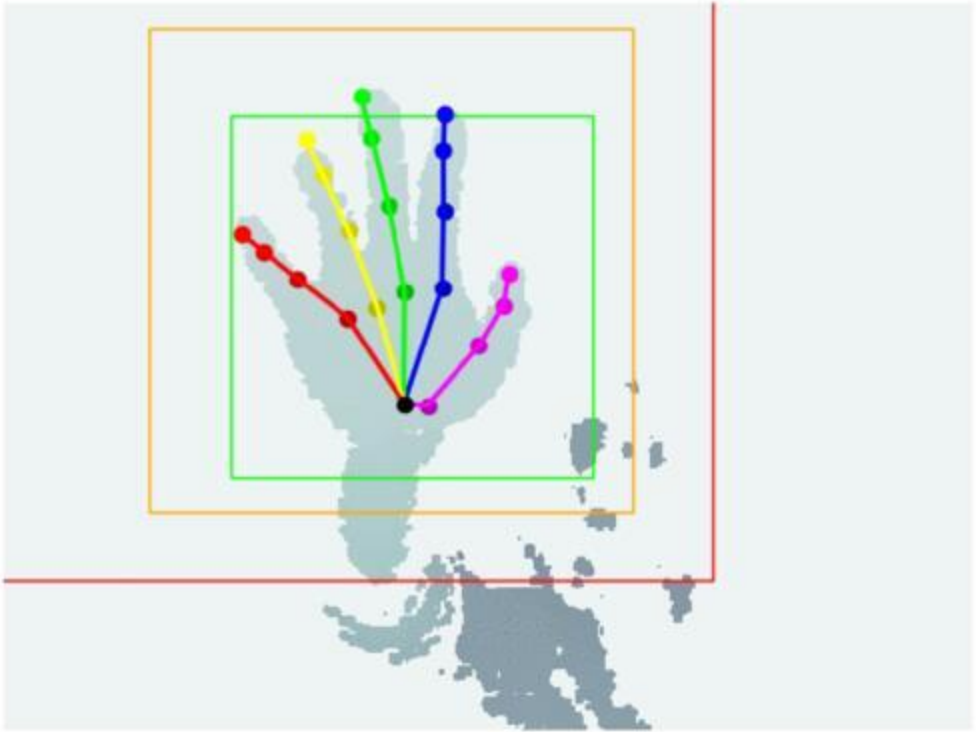
- 图像中有多个目标对象
- 如何确定、表示目标的位置?

目标检测：位置的不同粒度

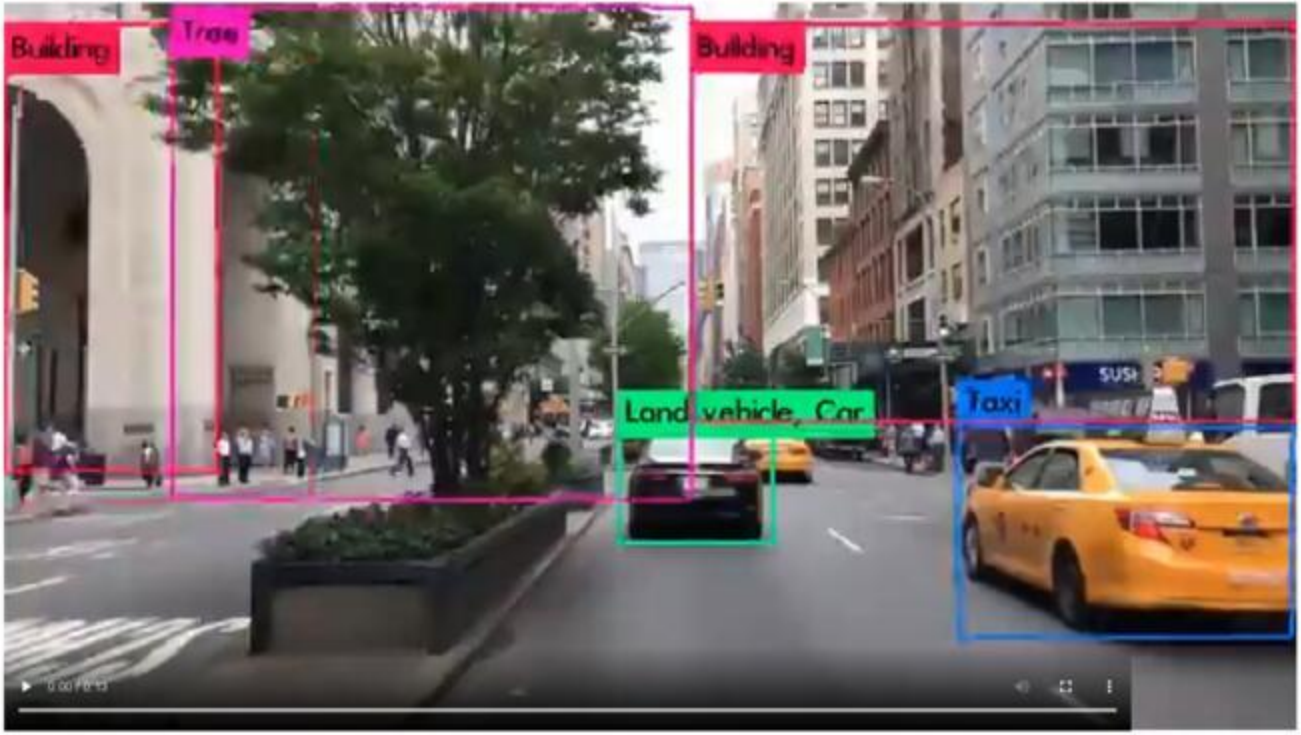
目标**检测**：输出类别标签、**位置**

- 位置的描述有不同层次
  - 单点？歧义过大，无法评测
  - 边缘？成本过大，计算复杂
  - 边界框：折衷方案，业界常用

目标检测：手势实例



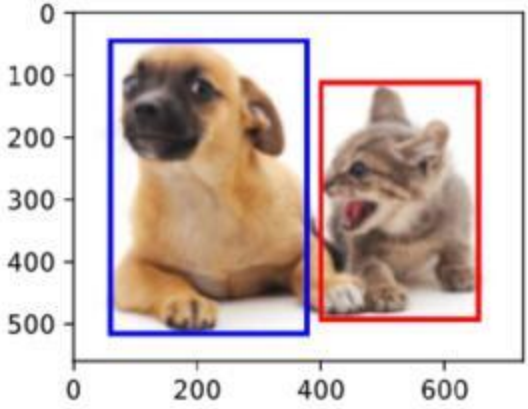
目标检测：街头实例



目标检测：边界框

定义：只需4个数字

- 左上(x, y)，右下(x, y)
- 左上(x, y)，宽、高(w, h)
  - 或中心点 + 宽、高

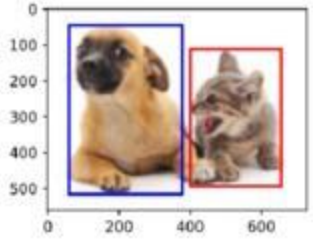


实验：边界框

目标检测：数据集

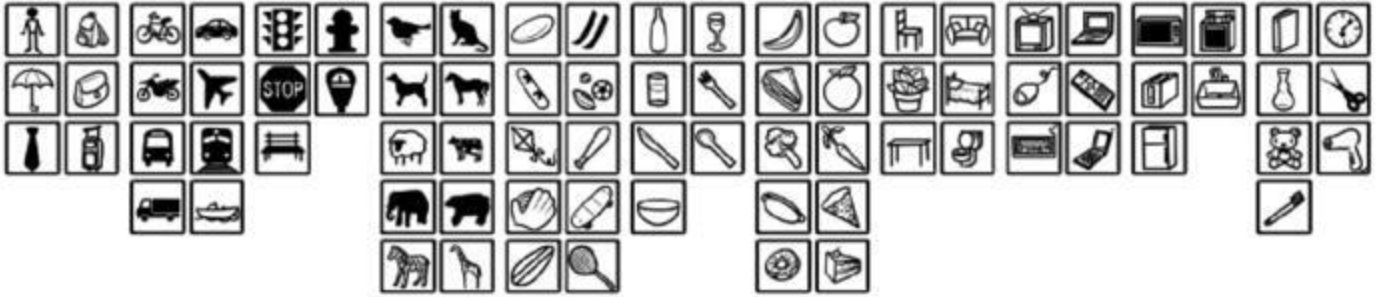
多个目标：每行一个物体

- 文件名、类别、边界框



COCO (cocodataset.org)

- 80物体类别，330K图片；1.5M物体



实验：目标检测数据集

小结：目标检测和边界框

- 目标检测：识别图片中多个物体的类别、位置
- 位置表示：常用边界框、边缘轮廓

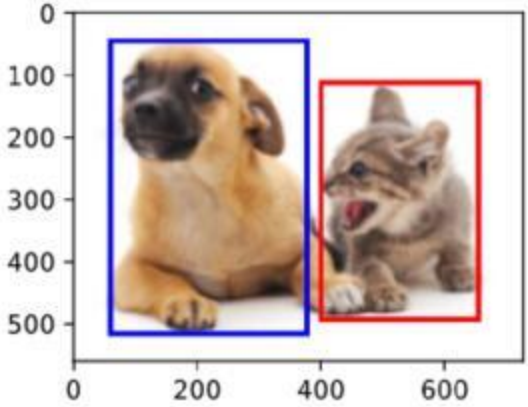
锚框

目标检测：两个阶段

目标检测可以看成两个阶段

1. 输出**可能**含物体的边界框
2. 判定边界框中物体的类别

因此边界框也称**备选区域 Region of Interest (ROI)**

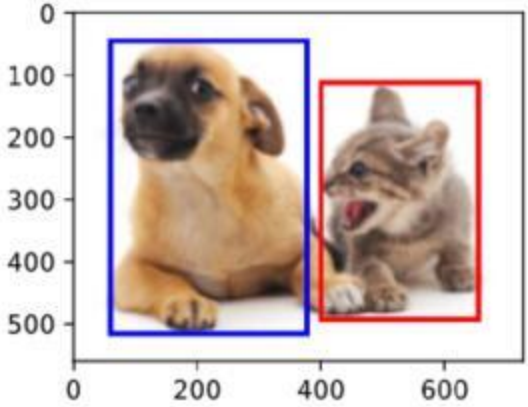


目标检测：两个阶段

目标检测可以看成两个阶段

1. 输出**可能**含物体的边界框
2. 判定边界框中物体的类别

因此边界框也称**备选区域 Region of Interest (ROI)**



ROI 调整算法：

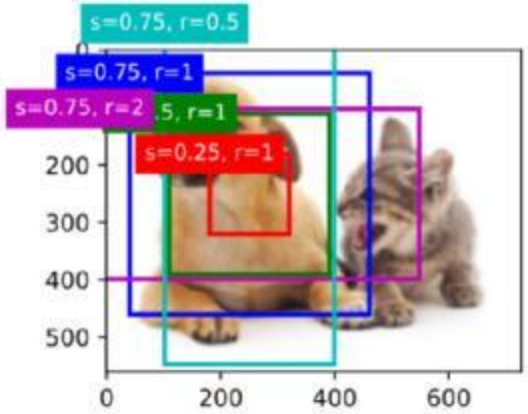
1. 采样大量备选框，并判定是否包含任何物体
2. 调整边界：计算、输出目标物体的真实边界框

锚框：完全覆盖法

以**每个像素**为中心生成不同形状的边界框

- 图宽w、高h，缩放比$s \in (0, 1]$，宽高比$r > 0$
  - 锚框：宽$ws\sqrt{r}$、高$hs/\sqrt{r}$

中心点称为“**锚点 anchor**”



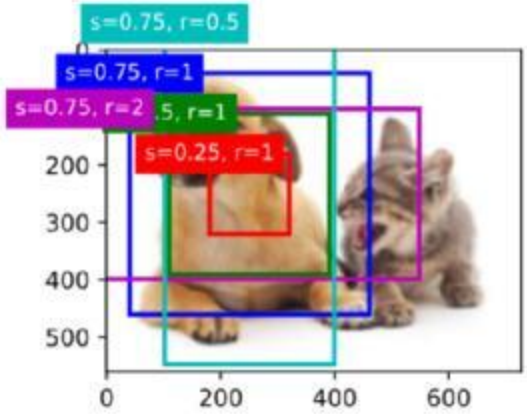
锚框：完全覆盖计算量

以**每个像素**为中心生成不同形状的边界框

- 图宽w、高h，缩放比$s \in (0, 1]$，宽高比$r > 0$
  - 锚框：宽$ws\sqrt{r}$、高$hs/\sqrt{r}$

中心点称为“**锚点 anchor**”

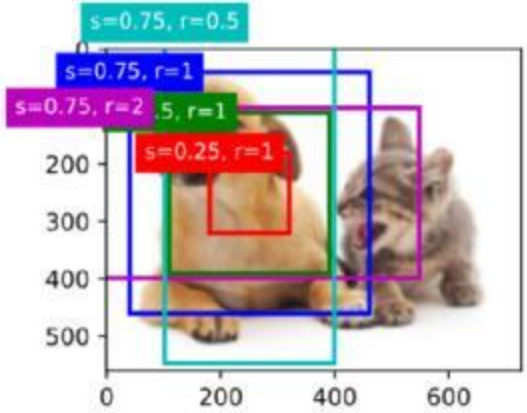
- 实际计算中缩放比、宽高比取有限的一系列值
  - $s \in \{s_1, .., s_n\}, r \in \{r_1, .., r_m\}$



锚框：完全覆盖计算量

以**每个像素**为中心生成不同形状的边界框

- 图宽w、高h，缩放比$s \in (0, 1]$，宽高比$r > 0$
  - 锚框：宽$ws\sqrt{r}$、高$hs/\sqrt{r}$



中心点称为“**锚点 anchor**”

- 实际计算中缩放比、宽高比取有限的一系列值
  - $s \in \{s_1, .., s_n\}, r \in \{r_1, .., r_m\}$

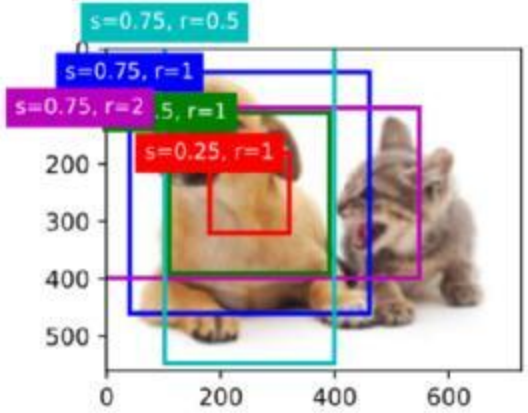
问题：计算复杂度太高，不可能计算

- 锚框总数：$whnm$，例如：$500 \times 500 \times 9 = 2.25$ M

锚框：完全覆盖计算量

以**每个像素**为中心生成不同形状的边界框

- 图宽w、高h，缩放比$s \in (0,1]$，宽高比$r > 0$
  - 锚框：宽$ws\sqrt{r}$、高$hs/\sqrt{r}$



中心点称为“**锚点 anchor**”

- 实际计算中缩放比、宽高比取有限的一系列值
  - $s \in \{s_1, .., s_n\}, r \in \{r_1, .., r_m\}$

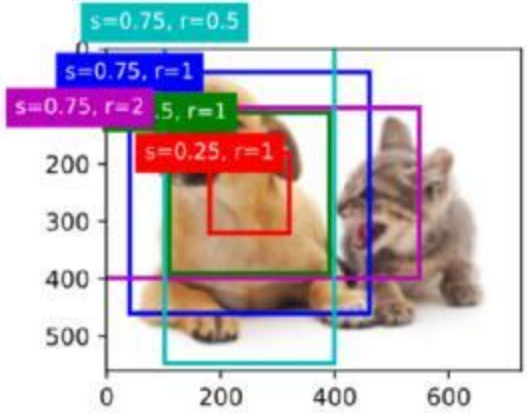
问题：计算复杂度太高，不可能计算

- 锚框总数：$whnm$，例如：$500 \times 500 \times 9 = 2.25$ M
- 实践中只考虑包含s_1或r_1的组合：$(s_1, r_1), .., (s_1, r_m), (s_2, r_1), .., (s_n, r_1)$
  - 每个锚点对应$m + n - 1$个锚框

交并比（IoU）

右图：蓝色框以$s = 0.75, r = 1$为参数似乎不错

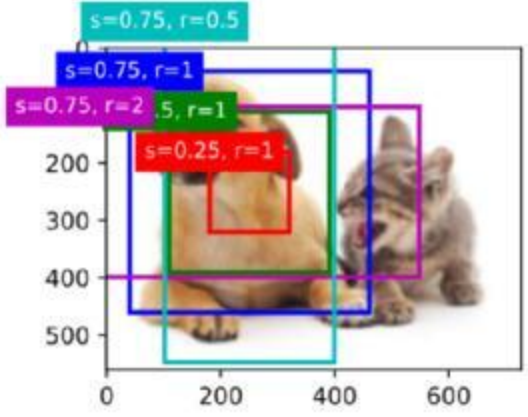
- 那么如何量化误差呢?
  - 本质：计算两个矩形面积的相对差异



交并比（IoU）

右图：蓝色框以$s = 0.75, r = 1$为参数似乎不错

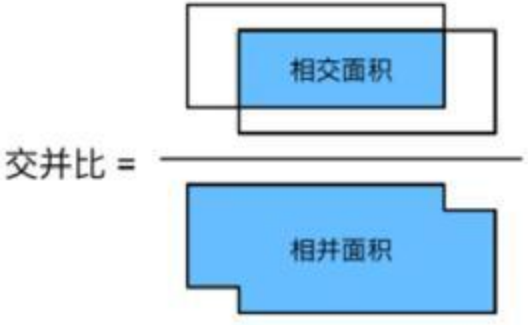
- 那么如何量化误差呢?
  - 本质：计算两个矩形面积的相对差异



IoU，也称**Jaccard相似度**：计算框之间的相似度

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

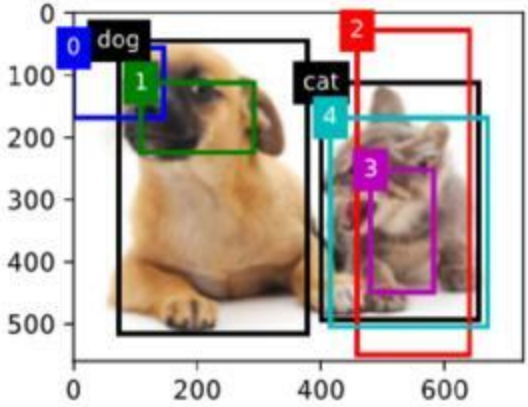
- 相似度取值范围：$[0, 1]$



锚框标注：训练、预测

训练：**每个锚框**构造、标注一个训练样本

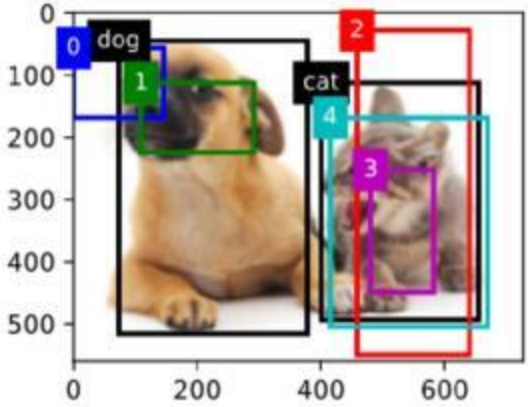
- 标签：锚框中物体的类别，或背景
  - 背景：可能生成大量负例
- 偏移量：真实边界框的**相对位移**
  - $(x, y) + (\Delta x, \Delta y)$



锚框标注：训练、预测

训练：**每个锚框**构造、标注一个训练样本

- 标签：锚框中物体的类别，或背景
  - 背景：可能生成大量负例
- 偏移量：真实边界框的**相对位移**
  - $(x, y) + (\Delta x, \Delta y)$



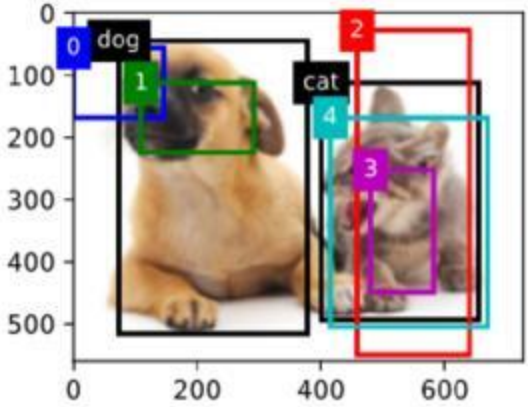
预测：生成多个锚框，逐个预测类别、偏移量

1. 计算、合并预测边界框
2. 根据**置信度**决定是否输出

锚框标注：训练、预测

训练：**每个锚框**构造、标注一个训练样本

- 标签：锚框中物体的类别，或背景
  - 背景：可能生成大量负例
- 偏移量：真实边界框的**相对位移**
  - $(x, y) + (\Delta x, \Delta y)$



预测：生成多个锚框，逐个预测类别、偏移量

1. 计算、合并预测边界框
2. 根据**置信度**决定是否输出

首先考虑标注训练样本（锚框）：分配到**最接近**的真实边界框

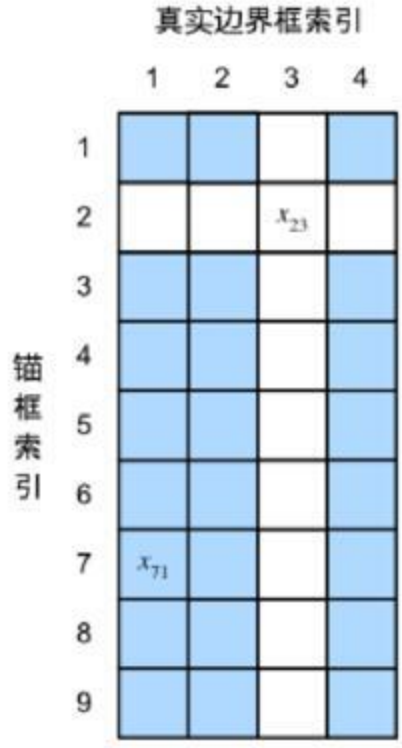
锚框标注：分配算法 I

每个锚框：分配**最接近**的真实边界框

- 锚框：$A_1, .., A_s$，真实：$B_1, .., B_t$
  - x_{ij}：A_i、B_j的IoU

1. 找出**最大元素**$x_{i_1 j_1}$：将A_{i_1}分配给B_{j_1}

- 丢弃i_1行、j_1列的剩余元素



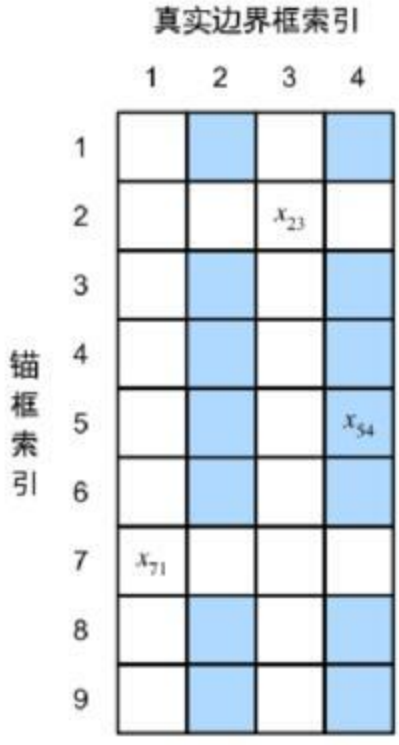
锚框标注：分配算法 II

每个锚框：分配**最接近**的真实边界框

- 锚框：$A_1, .., A_s$，真实：$B_1, .., B_t$
  - x_{ij}：A_i、B_j的IoU

1. 找出**最大元素**$x_{i_1 j_1}$：将A_{i_1}分配给B_{j_1}
2. 继续找出**最大元素**$x_{i_2 j_2}$：将A_{i_2}分配给B_{j_2}

- 丢弃i_2行、j_2列的剩余元素



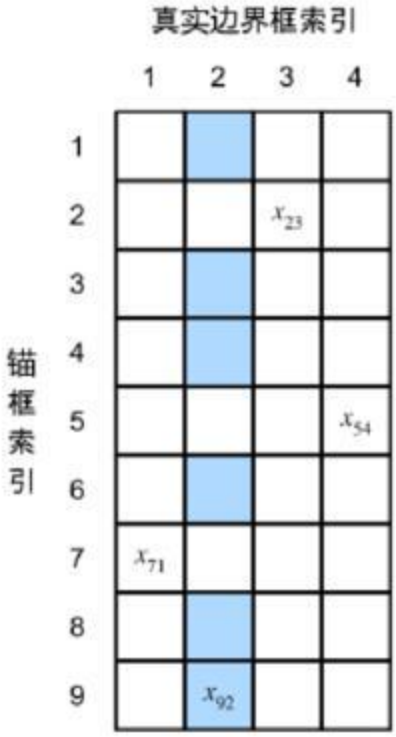
锚框标注：分配算法 III

每个锚框：分配**最接近**的真实边界框

- 锚框：$A_1, .., A_s$，真实：$B_1, .., B_t$
  - x_{ij}：A_i、B_j的IoU

1. 找出**最大元素**$x_{i_1 j_1}$：将A_{i_1}分配给B_{j_1}
2. 继续找出**最大元素**$x_{i_2 j_2}$：将A_{i_2}分配给B_{j_2}
3. 继续以上步骤，直到所有t个**真实边界框**都已被分配

- 此例：只有4个锚框被分配



锚框标注：分配算法 IV

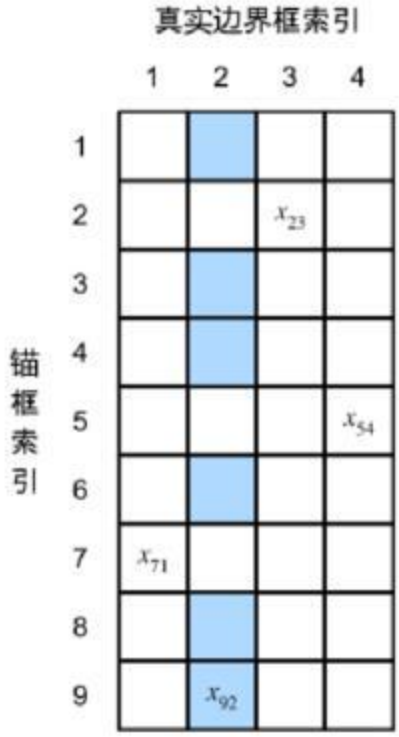
每个锚框：分配**最接近**的真实边界框

- 锚框：$A_1, .., A_s$，真实：$B_1, .., B_t$
  - x_{ij}：A_i、B_j的IoU

1. 找出**最大元素**$x_{i_1 j_1}$：将A_{i_1}分配给B_{j_1}
2. 继续找出**最大元素**$x_{i_2 j_2}$：将A_{i_2}分配给B_{j_2}
3. 继续以上步骤，直到所有t个**真实边界框**都已被分配
4. 遍历其余$s-t$个锚框：只有当IoU**大于阈值**时分配

输出：每个真实边界框对应多个锚框

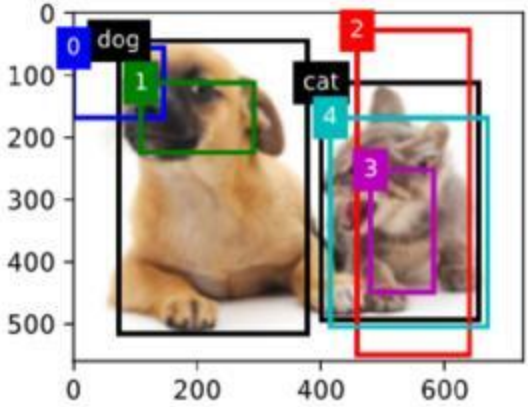
- 按照IoU的**降序排列**



锚框标注：类别

类别：按照分配情况标记

- 没有分配：标记为“**背景**”，**即负例**
  - IoU 小于阈值



锚框标注：类别、偏移量

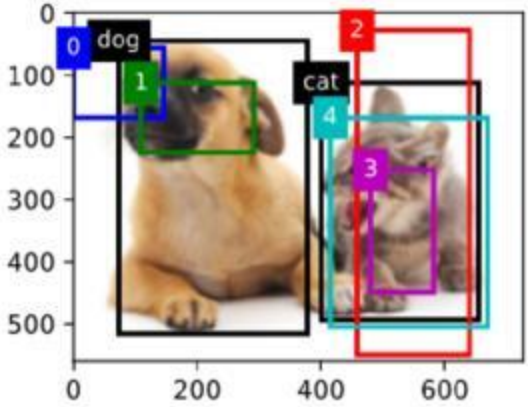
类别：按照分配情况标记

- 没有分配：标记为“**背景**”，**即负例**
  - IoU 小于阈值

偏移量：中心点、相对大小

$$\left(\frac{x_b - x_a}{w_a}, \frac{y_b - y_a}{h_a}, \frac{w_b}{w_a}, \frac{h_b}{h_a}\right)$$

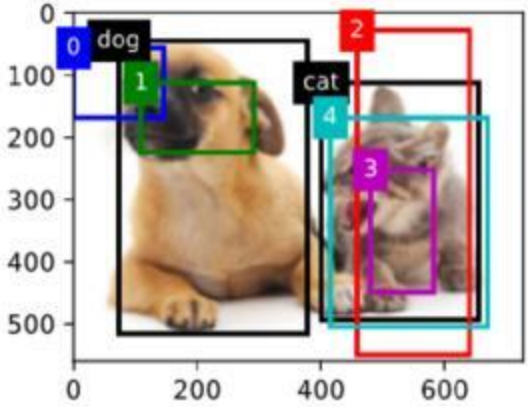
- 问题：数据复杂时**数值差异过大**、难以拟合



锚框标注：类别、偏移量

类别：按照分配情况标记

- 没有分配：标记为“**背景**”，**即负例**
  - IoU 小于阈值



偏移量：中心点、相对大小

$$\left(\frac{x_b - x_a}{w_a}, \frac{y_b - y_a}{h_a}, \frac{w_b}{w_a}, \frac{h_b}{h_a}\right)$$

- 问题：数据复杂时**数值差异过大**、难以拟合

解决方案：变换位置、大小，使其**分布更均匀**

$$\left(\frac{\frac{x_b - x_a}{w_a} - \mu_x}{\sigma_x}, \frac{\frac{y_b - y_a}{h_a} - \mu_y}{\sigma_y}, \frac{\log\frac{w_b}{w_a} - \mu_w}{\sigma_w}, \frac{\log\frac{h_b}{h_a} - \mu_h}{\sigma_h}\right)$$

锚框预测：边界框

然后看锚框预测：生成多个锚框，逐个预测类别、偏移量

1. 计算、合并预测边界框
2. 根据**置信度**决定是否输出

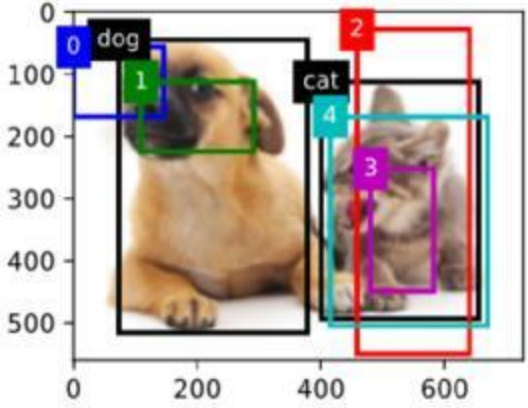
锚框预测：边界框

然后看锚框预测：生成多个锚框，逐个预测类别、偏移量

1. 计算、合并预测边界框
2. 根据**置信度**决定是否输出

计算边界框：根据偏移量反向调整

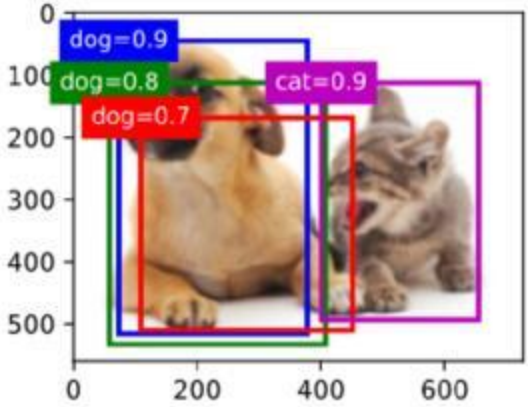
- 锚框A**逆向偏移**后得到预测边界框B
  - 前一页公式反向操作



锚框预测：类别置信度

计算每个类别的概率：取最大的作为预测类别

- 称为预测边界框的**置信度（confidence）**



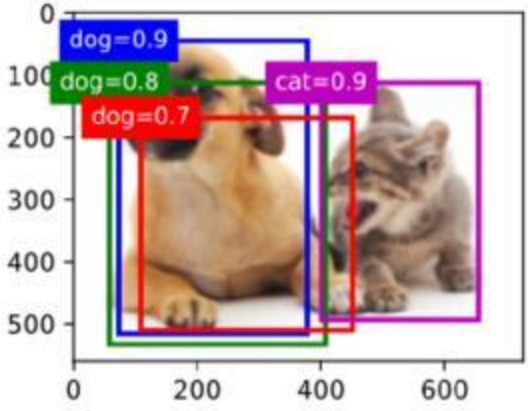
锚框预测：非极大值抑制

计算每个类别的概率：取最大的作为预测类别

- 称为预测框的**置信度（confidence）**

生成锚框过于稠密，且相似：增加不必要计算量

- 合并预测框：相同物体、相似边界框
  - **非极大值抑制（non-maximum suppression，NMS）**



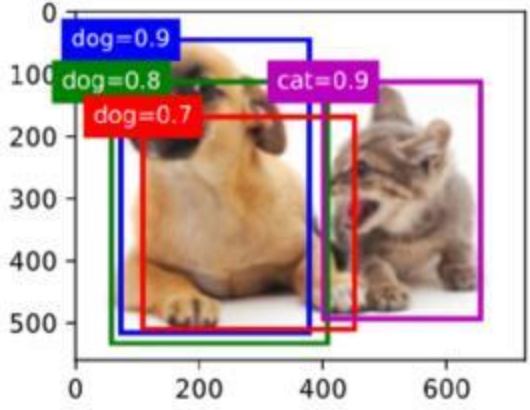
锚框预测：非极大值抑制

计算每个类别的概率：取最大的作为预测类别

- 称为预测框的**置信度（confidence）**

生成锚框过于稠密，且相似：增加不必要计算量

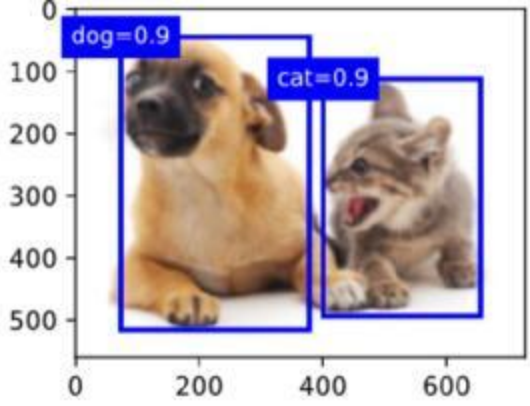
- 合并预测框：相同物体、相似边界框
  - **非极大值抑制（non-maximum suppression，NMS）**



NMS：预测框按**置信度降序**排列，可截断

1. 取置信度最高的边界框B
   1. 将B加入**保留列表**
   2. 移除所有与B的**IoU超过阈值**的预测框
2. 重复，直至所有预测框都在保留列表：输出

输出中任意一对预测框**都不相似**：因IoU小于阈值


dog=0.9
cat=0.9
0
100
200
300
400
500
0
200
400
600

实验：锚框

小结：锚框

- 目标检测：锚点附近的边界框 + 类别标签
- 每个锚框构造、标注一个样本
  - 锚框标注：类别标签、偏移量
- 预测：根据预测偏移量还原
  - 置信度：确定类别，去除负类
  - NMS：去除冗余预测

多尺度目标检测

锚框：计算复杂度

以**每个像素**为中心生成不同形状的边界框

- 图宽w、高h，缩放比s，宽高比r

锚框：计算复杂度

以**每个像素**为中心生成不同形状的边界框

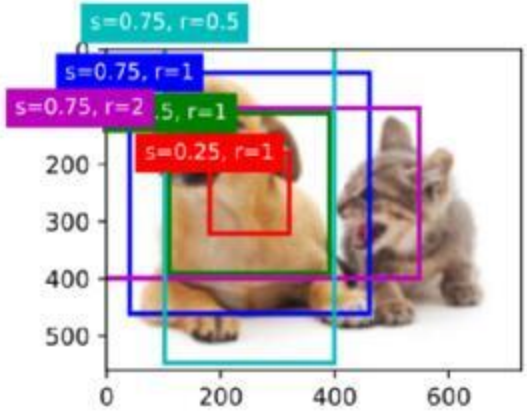
- 图宽w、高h，缩放比s，宽高比r

问题：计算复杂度太高，不可能计算

- 锚框总数：$whnm$

例如：561x728的图像，3种缩放比、宽高比

- $561 \times 728 \times 9 > 367$ 万



锚框：计算复杂度

以**每个像素**为中心生成不同形状的边界框

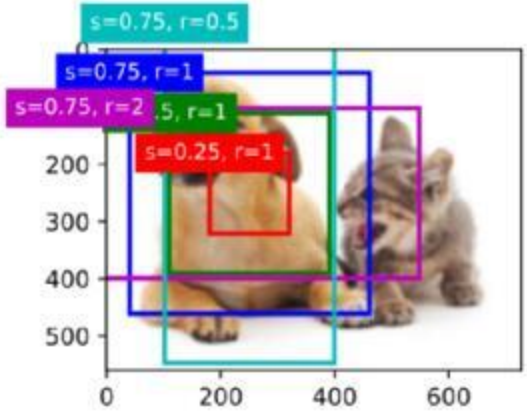
- 图宽w、高h，缩放比s，宽高比r

问题：计算复杂度太高，不可能计算

- 锚框总数：$whnm$

例如：561x728的图像，3种缩放比、宽高比

- $561 \times 728 \times 9 > 367$ 万

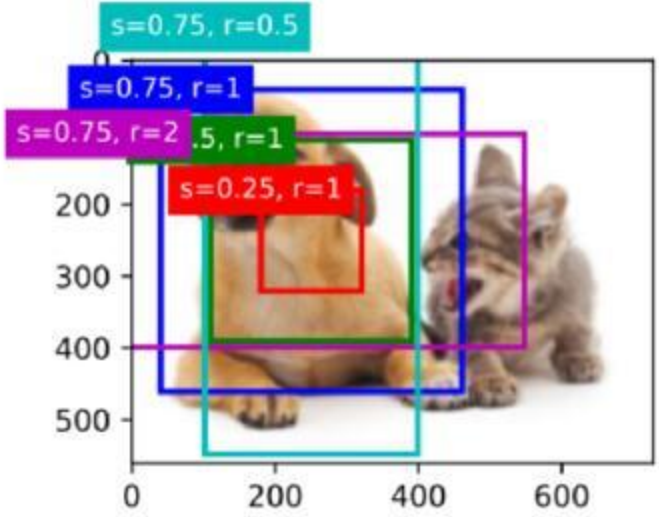


密集采样完全没必要：生成大量冗余锚框

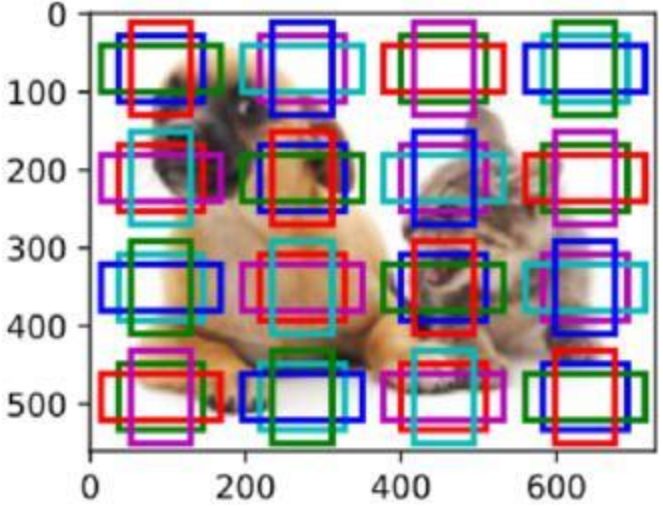
- **均匀间隔采样**作为锚点

均匀间隔采样

每个像素点都是锚点



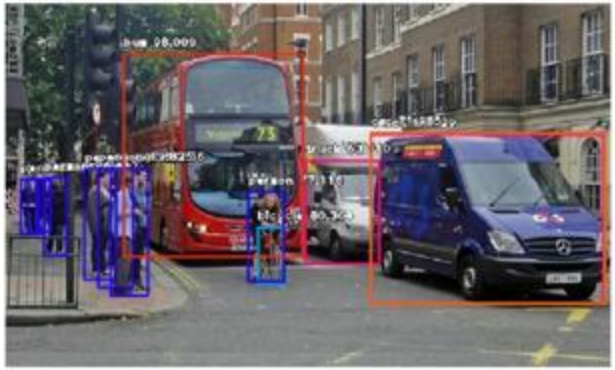
固定间隔、相同宽高比



- 行、列都只有4个采样点

目标、锚框尺寸

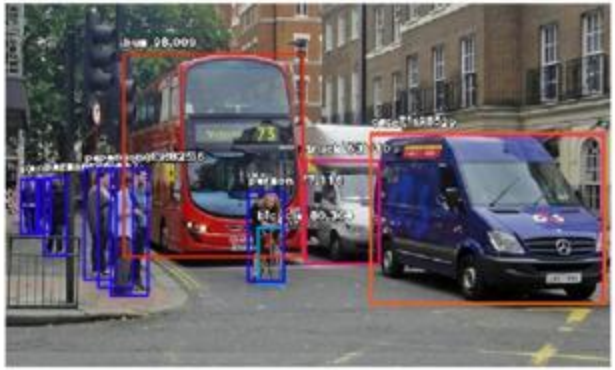
目标尺寸不同：锚框大小可以、且应该**区别对待**



- 目标较小：使用小锚框也有足够的可选尺寸
  - 例如：2x2图像可以填充4种1x1、2种1x2、1种2x2
  - 使用大锚框反倒会圈入**干扰信息**

目标、锚框尺寸

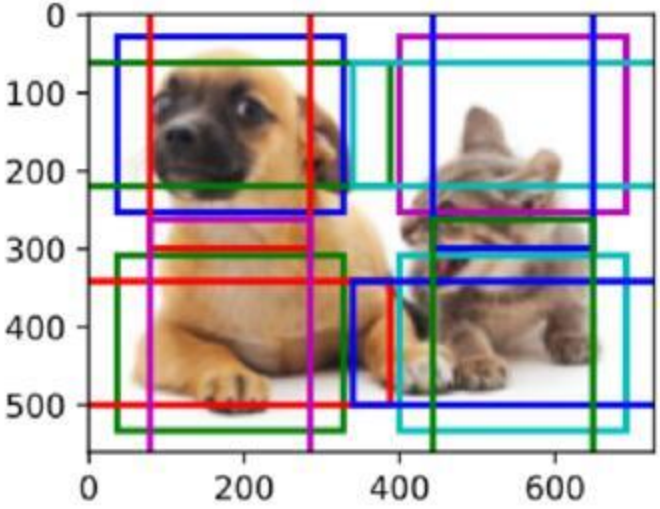
目标尺寸不同：锚框大小可以、且应该**区别对待**



- 目标较小：使用小锚框也有足够的可选尺寸
  - 例如：2x2图像可以填充4种1x1、2种1x2、1种2x2
  - 使用大锚框反倒会圈入**干扰信息**
- 目标较大：必须使用**相匹配**的锚框尺寸
  - 由于均匀间隔，采样数可以相应减少

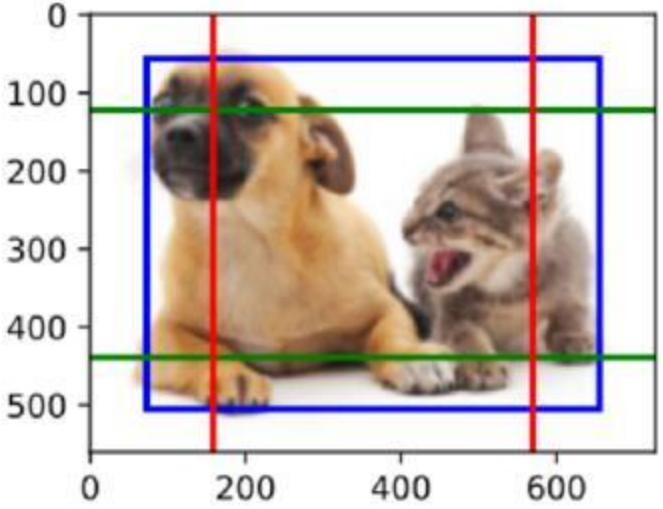
多尺度锚框

行、列都只有2个采样点



- 锚框尺寸必须相应增加
  - 更大的观察窗口：检测更大目标

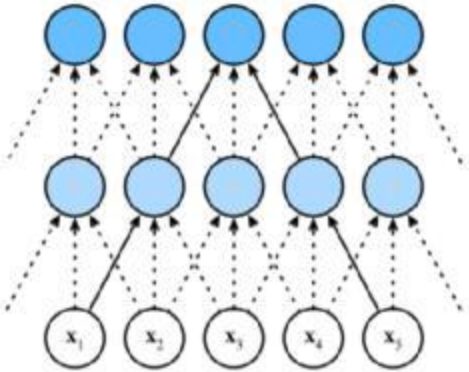
行、列都只有1个采样点



- 极端情况：锚点即图像中心点

多尺度检测

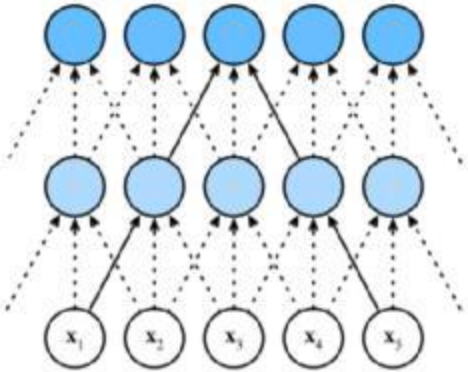
锚框大小：等价于**感受野尺寸**，“站得高看得远”



- 回顾：神经网络可以看成特征提取器
  - 深度不同的层：提取不同**抽象级别**的特征

多尺度检测

锚框大小：等价于**感受野尺寸**，“站得高看得远”



- 回顾：神经网络可以看成特征提取器
  - 深度不同的层：提取不同**抽象级别**的特征

锚框本质上是利用**感受野接受到的信息**进行预测

- 较深的层：感受较大规模的目标
  - 必然需要较大的感受野、锚框

实验：多尺度目标锚框

小结：多尺度目标检测

- 目标尺寸不同：需要相应尺寸的锚框
- 减少冗余计算：均匀间隔采样
- 多尺度检测：感受野大小不同、提取出特征的抽象级别不同

Review

本章内容

图像增广。微调。实战 Kaggle 比赛：图像分类。目标检测和边界框。锚框。多尺度目标检测。

重点：图像增广；微调；锚框；多尺度均匀间隔采样。

难点：部分重用式微调。

学习目标

- 理解图像增广的原因、主要方法。
- 理解微调的动机、原理、方法。
- 理解目标检测的特点（多物体）和表示方法（边界框）。
- 理解锚框的表示、标注方法，及其在训练、预测中的应用方法。
- 理解锚框的多尺度均匀间隔采样的方法、含义。

问题

简述图像增广的原因、主要方法。

简述微调的动机、原理、方法。

简述目标检测问题的特点和表示方法。

简述锚框的表示、标注方法，及其在训练、预测中的应用方法。

简述锚框的多尺度均匀间隔采样的方法、含义。