智能系统安全实践：有监督学习

复旦白泽智能

系统软件与安全实验室

大纲

- 理解有监督学习基本框架
- 模型定义、损失函数、梯度下降、训练-预测

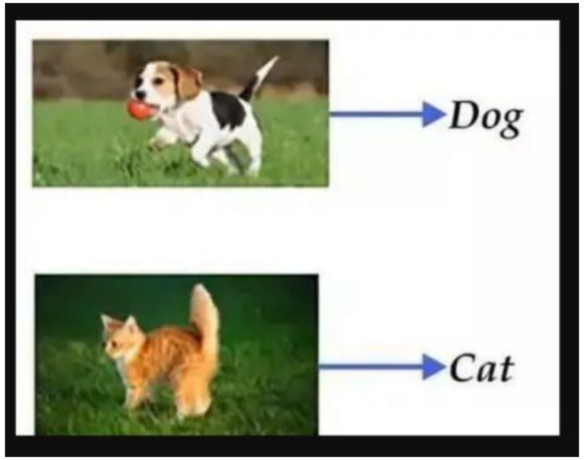
- 理解线性回归模型
- 有监督学习框架的一个例子

- 理解多维线性分类模型
- 多维线性模型、交叉熵损失函数

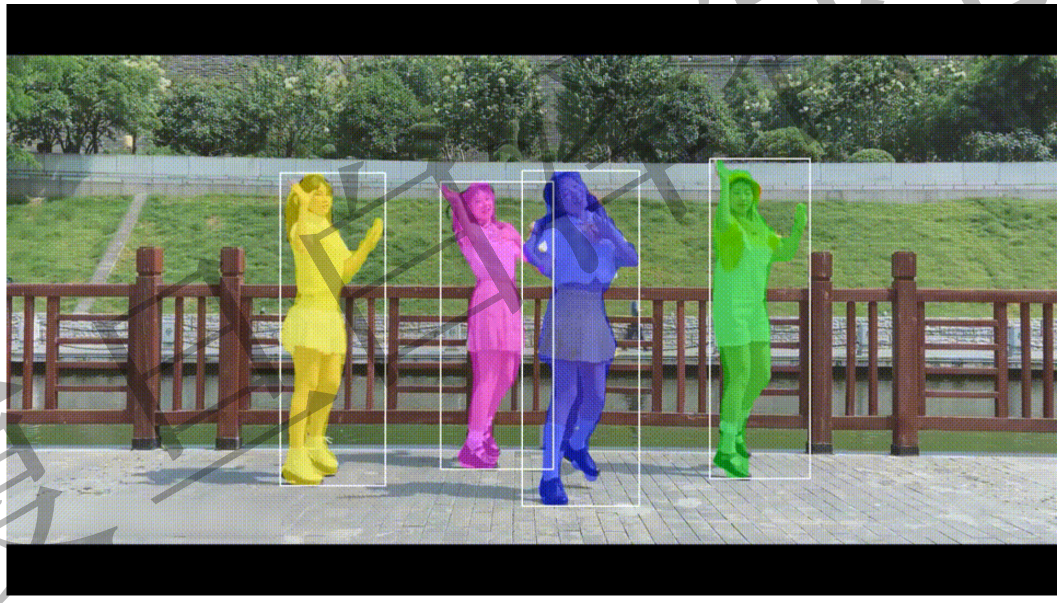
- 在红酒类型数据集上做实验

有监督学习

- 图像分类



- 实例分割

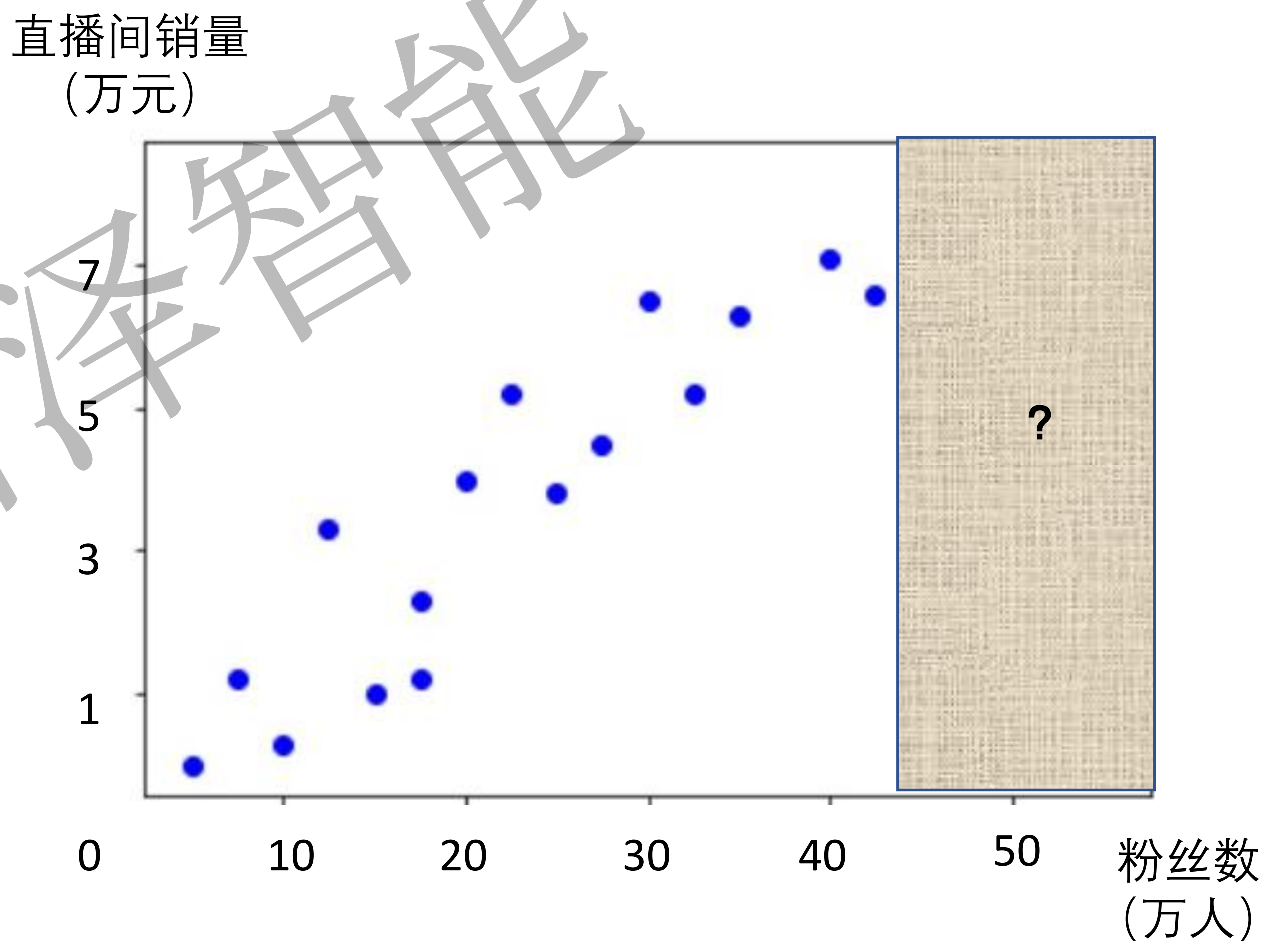


- 语音识别

- Q: 上述深度学习模型的原理？ A: 有监督学习

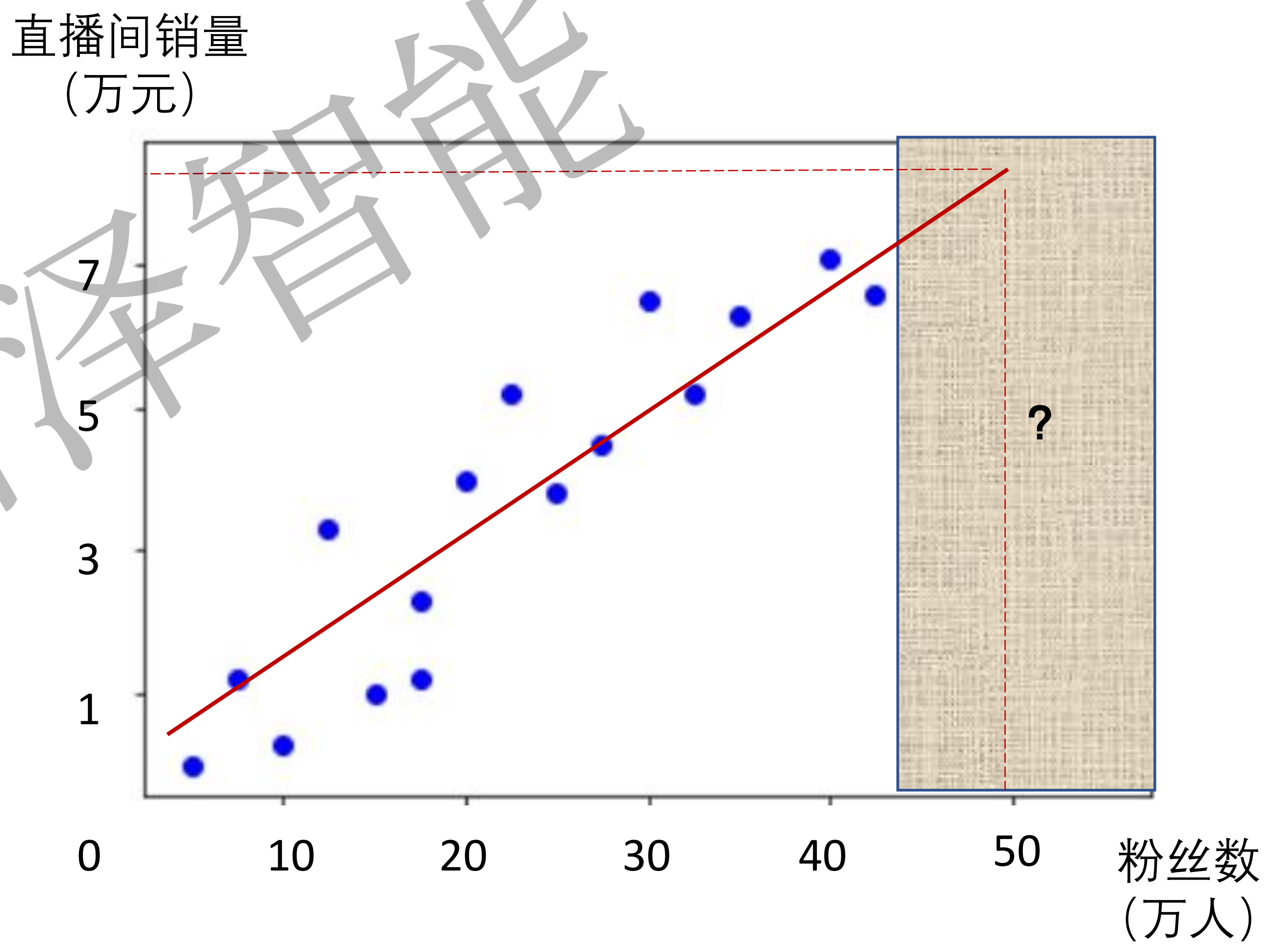
有监督学习：一个简单的例子

- 现收集了各个主播的数据
- 运营方想要知道当粉丝数为50万人直播间销量大概是多少
- 如何预测?



有监督学习：一个简单的例子

- 从数据规律来看是线性关系
- 拟合直线，然后看y轴值多少
- 如何用数学语言描述这个过程?



有监督学习：线性建模

- 假设已经有了如下N个数据点：

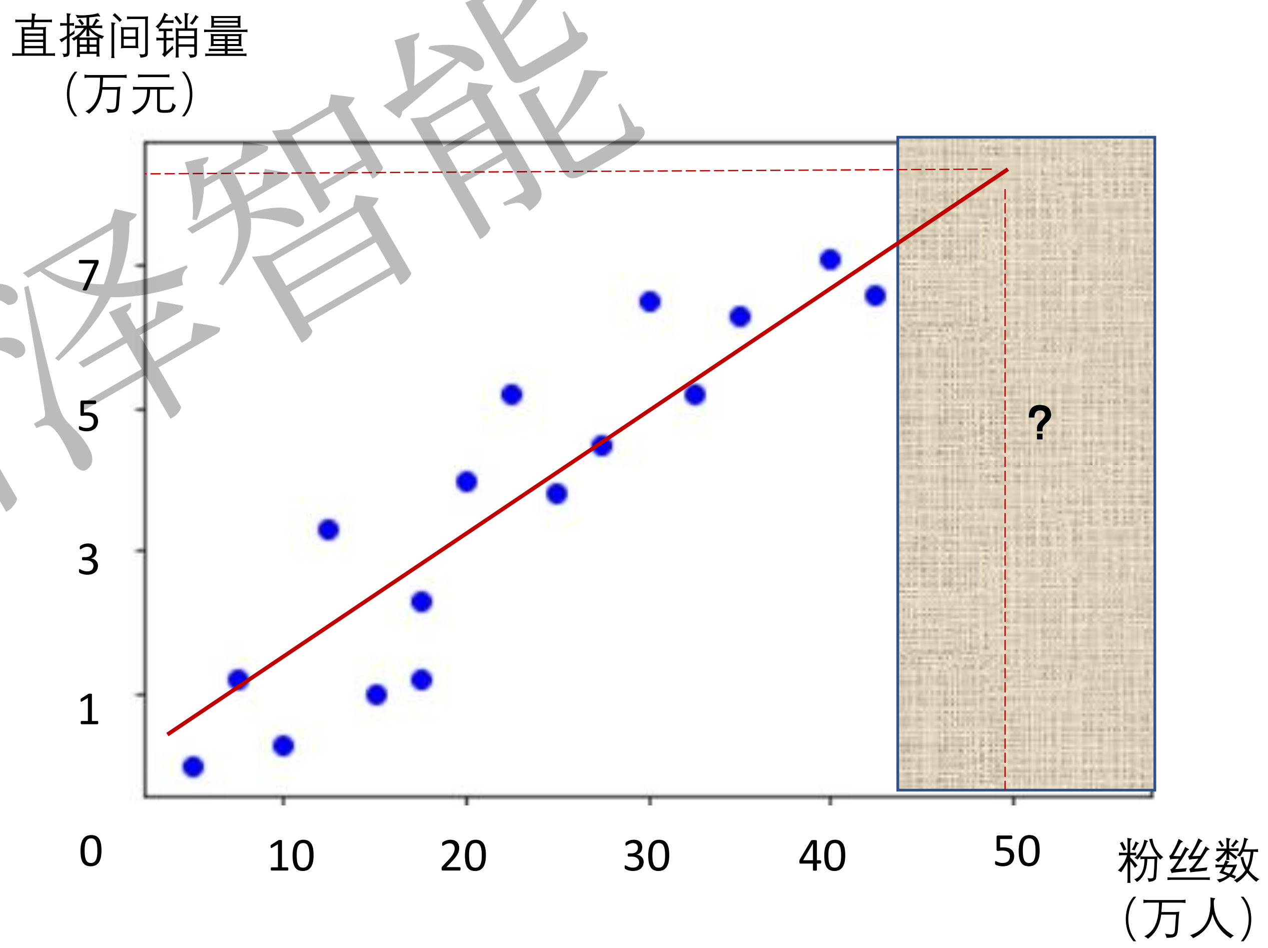
$$\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(N)}, y^{(N)}\right)$$

- 同时假设直线方程式为：

$$f(x) = wx + b$$

- 接下来求解方程的参数w, b：

$$\min_{w,b} \sum_{i=1}^{N} \left(y^{(i)} - f(x^{(i)})\right)^2$$

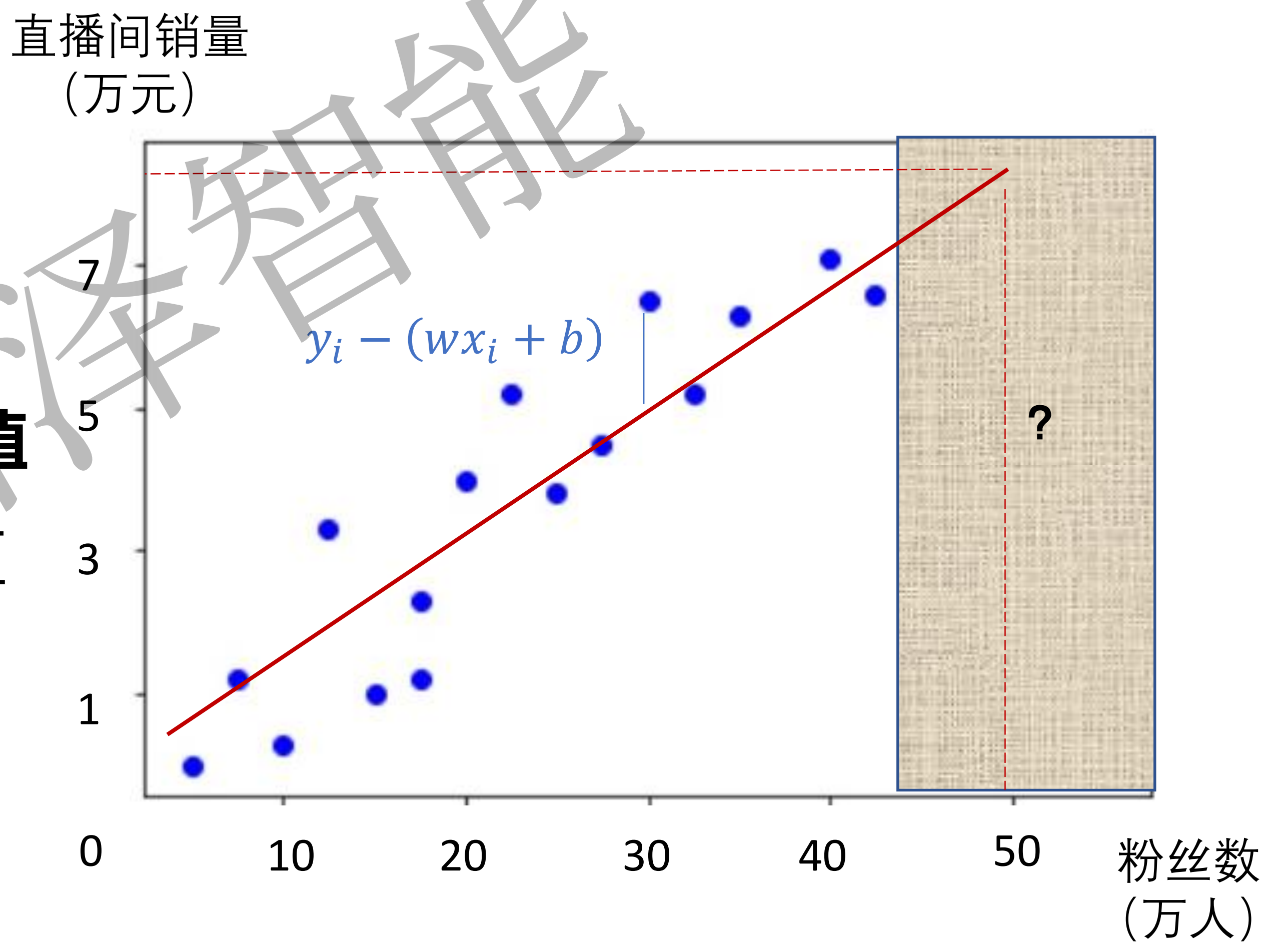


有监督学习：损失函数

- 这个优化目标被称作**损失函数**：

$$\min_{w,b}\sum_{i=1}^{N}(y^{(i)}-f(x^{(i)}))^2$$

- 从定义上来看
  - $f(x^{(i)})$表示用直线方程对$x^{(i)}$的**预测值**
  - $\left(y^{(i)}-f(x^{(i)})\right)^2$表示预测值与真实值的偏差（损失）
- 最小化损失函数⇔让所有点的预测都尽可能准确



有监督学习：梯度下降

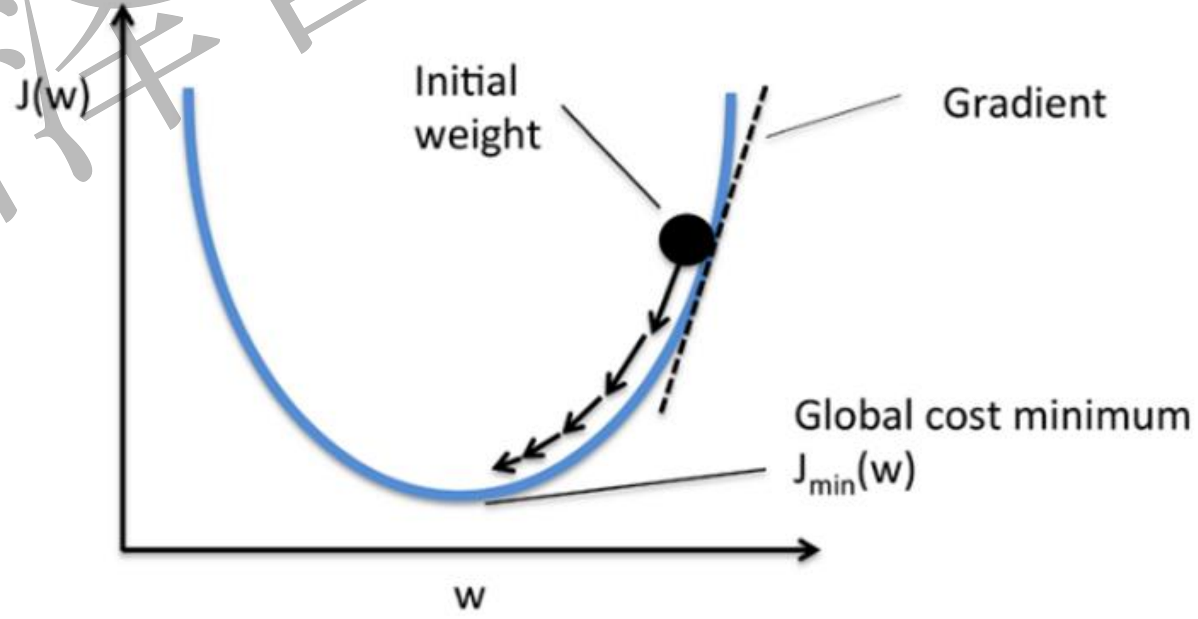
- 如何求解最小化损失函数的参数w, b？ -> 梯度下降法
  - 考虑函数$l(w) = (w-1)^2$的优化
    $$\min_w l(w)$$
  - 使用如下的算法：

    1. 随机初始化w

    2. 对当下的w求$l(w)$的导数$l'(w)$

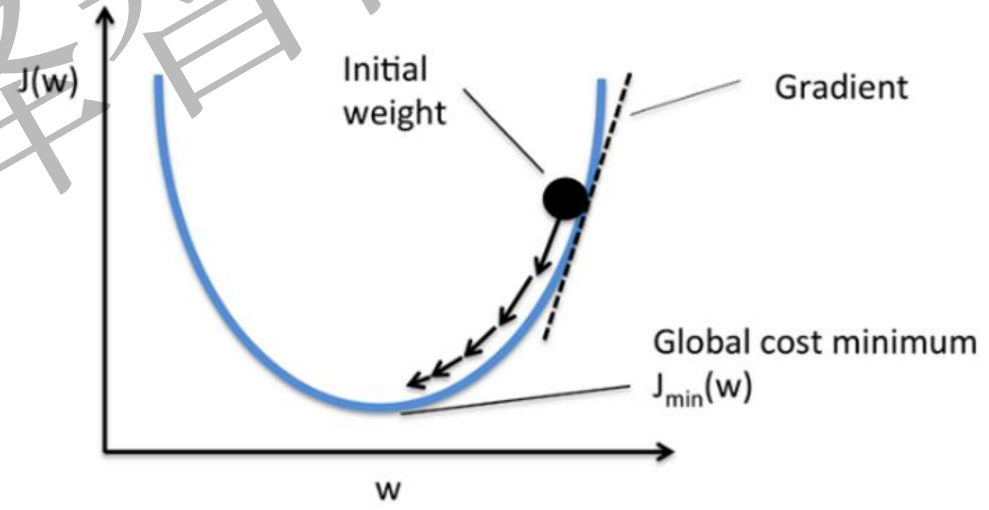
    3. 做如下的更新：
    $$w = w - \alpha \cdot l'(w)$$

    4. 回到第2步，直至w不再变化



有监督学习：梯度下降

- 如何理解梯度下降?
  - 考虑函数$l(w) = (w-1)^2$的最小化，学习速率$\alpha = 0.1$
    - 当$w = 3$，在最优解右边：
      $$l'(w) = 2(3-1) = 4$$
      $$w = 3 - 0.1 * 4 = 2.6$$
    - 当$w = -3$，在最优解左边：
      $$l'(w) = 2(-3-1) = -8$$
      $$w = -3 - 0.1 * (-4) = -2.2$$
    - 当$w = 1$，在最优解上：
      $$l'(w) = 2(1-1) = 0$$
      $$w = 1 - 0.1 * 0 = 1$$



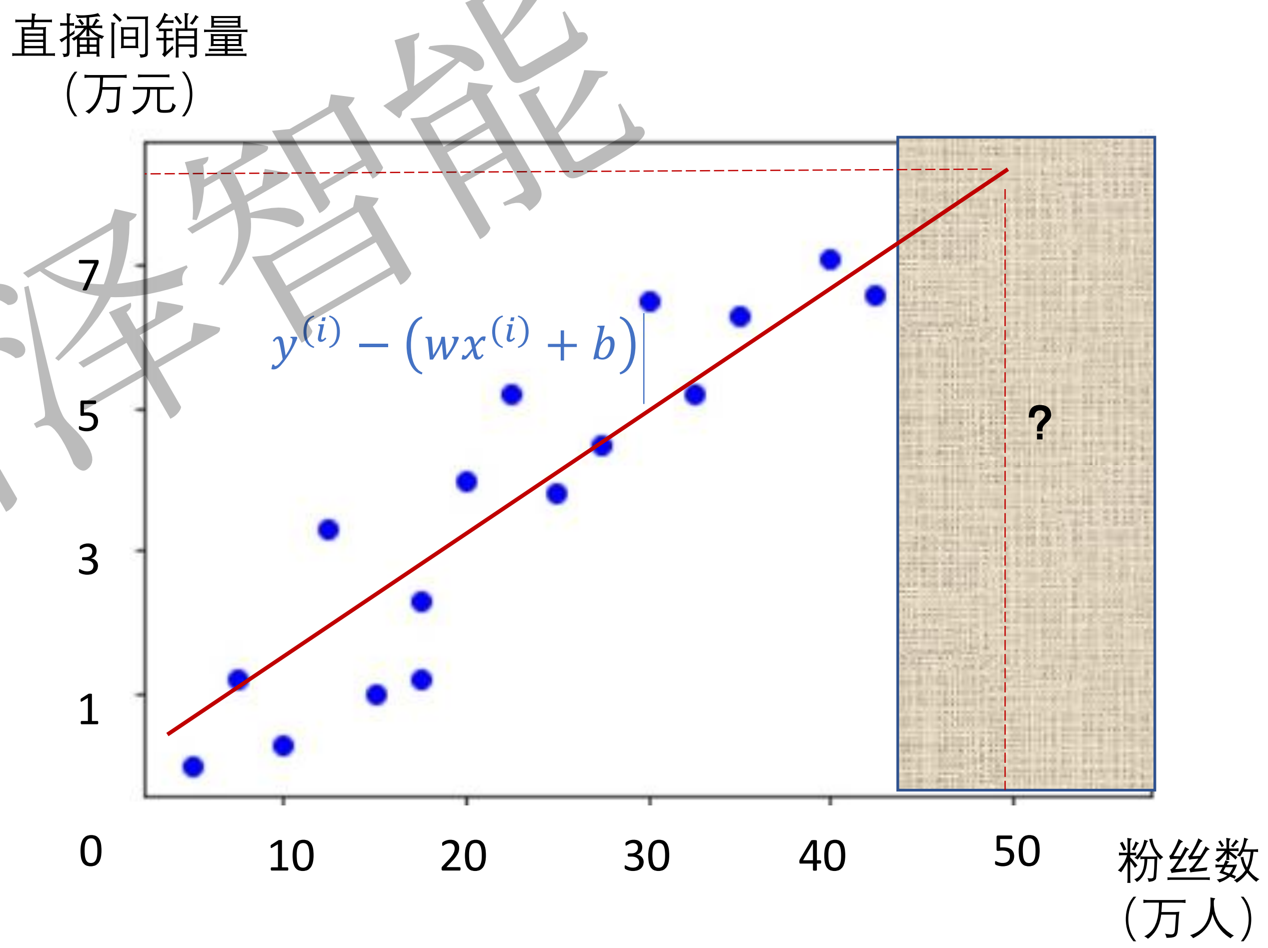
- 每次都在朝向最优解走一小步
- 走到最优解之后停止

有监督学习：梯度下降

- 如何应用在本问题中?
- 定义：$\ell(w,b)=\sum_{i=1}^{N}(y^{(i)}-f(x^{(i)}))^2$
  - 随机初始化w,b，指定学习速率α
  - 做下述迭代：

$$w=w-\alpha\cdot\nabla_w l(w,b)$$

$$b=b-\alpha\cdot\nabla_b l(w,b)$$



有监督学习：梯度下降

- 偏导数推导:
  - 损失函数：$\ell(w,b)=\sum_{i=1}^{N}(y^{(i)}-f(x^{(i)}))^2$
  - 求偏导（链式求导法则）：

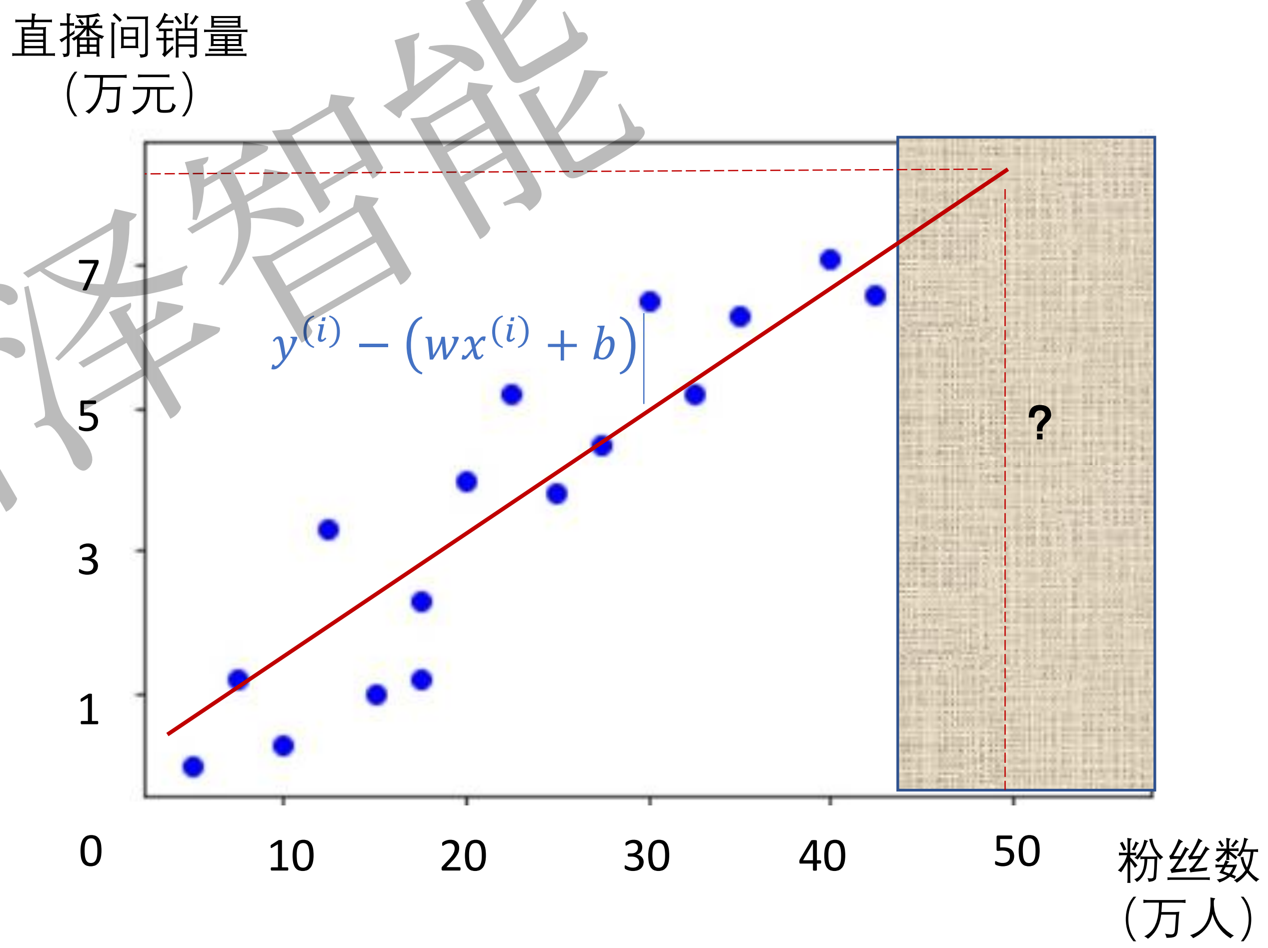
$$\nabla_w \ell(w,b)=\sum_{i=1}^{N}\nabla_w(y^{(i)}-f(x^{(i)}))^2=\sum_{i=1}^{N}\frac{\partial[(y^{(i)}-f(x^{(i)}))^2]}{\partial f(x^{(i)})}\cdot\frac{\partial f(x^{(i)})}{\partial w}$$

$$=-\sum_{i=1}^{N}2[y^{(i)}-f(x^{(i)})]\cdot x^{(i)}$$

$$\nabla_b \ell(w,b)=\sum_{i=1}^{N}\frac{\partial[(y^{(i)}-f(x^{(i)}))^2]}{\partial f(x^{(i)})}\cdot\frac{\partial f(x^{(i)})}{\partial b}=-\sum_{i=1}^{N}2[y^{(i)}-f(x^{(i)})]$$

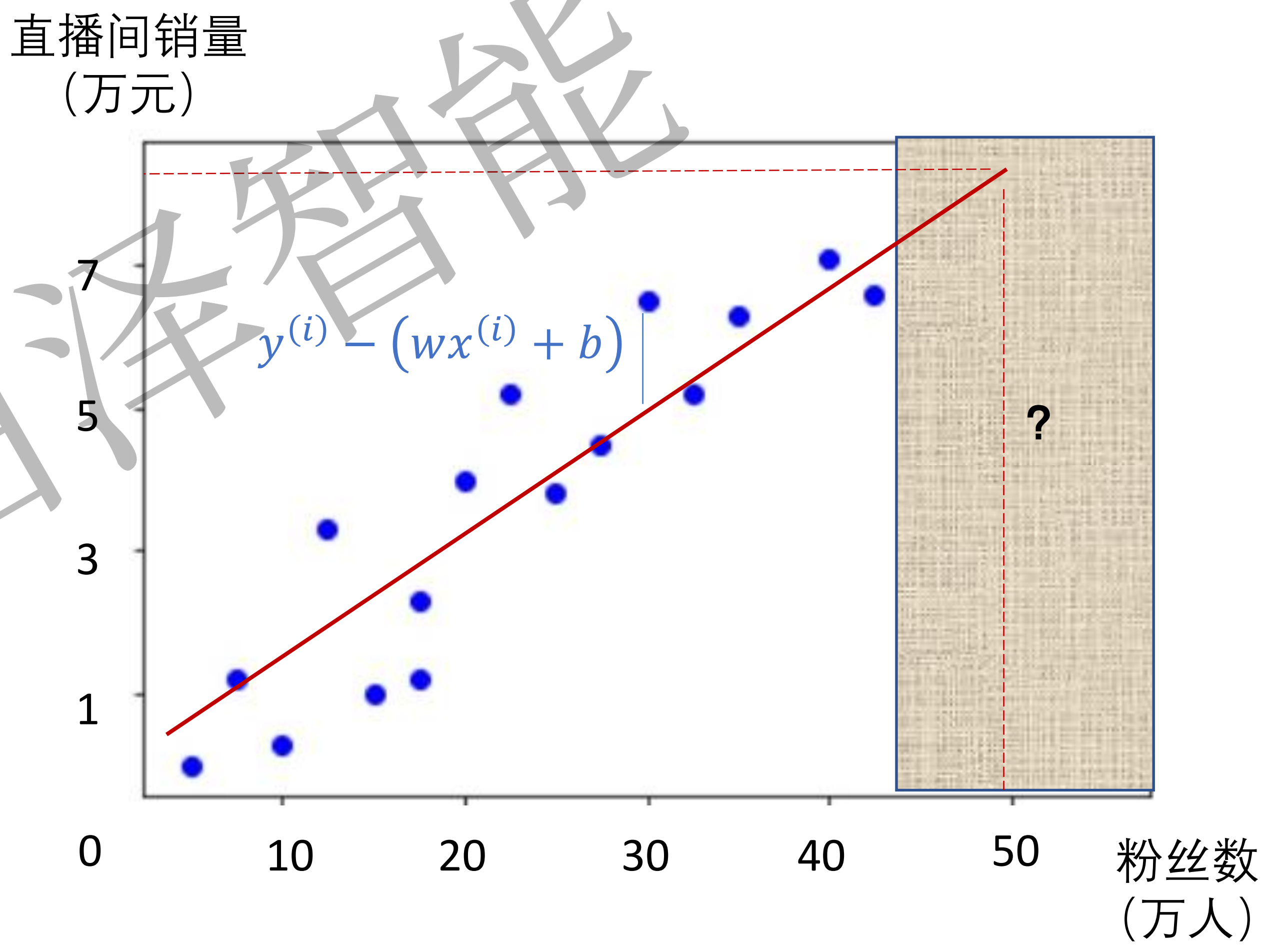
有监督学习：预测

- 完成上述优化之后：
  - 建立直线方程$f(x) = w_* x + b_*$
  - 其中w_*, b_*是优化之后的值
- 给定数据$x = 50$
  - 通过上述方程计算$f(x)$
  - 这个被称作**模型预测**过程



有监督学习：小结

- 有监督学习的几个要素
  - 准备工作：
    - 收集**训练集**$(x^{(i)}, y^{(i)})$
    - 建立模型$f(x)$
  - 模型**训练**：
    - 设定**损失函数**$\ell(\theta)$
    - 通过梯度下降最小化$\ell(\theta)$
  - 模型**预测**：在新数据点上计算$f(x)$



Q&A

有监督学习：红酒类型预测

- 应用上述框架：
  - 准备工作：
    - 收集**训练集**$(x^{(i)}, y^{(i)})$ -> 每款红酒参数和类型
    - 建立模型$f(x)$ -> ？
  - 模型**训练**：
    - 设定**损失函数**$\ell(\theta)$ -> ？
    - 通过梯度下降最小化$\ell(\theta)$
  - 模型**预测**：在新数据点上计算$f(x)$

属性	值
固定酸度	8.319637
挥发物	0.527821
柠檬酸	0.270976
糖分	2.538806
氯化物	0.087467
游离SO_2	15.874922
总SO_2	46.467792
密度	0.996747
PH值	3.311113
硫酸盐	0.658149
酒精度	10.422983

红酒类型：$\{0,1,2\}$

红酒类型预测：建立模型

- 建立模型$f_\theta(X)$
  - 输入定义：
  - $x \in \mathbb{R}^{13}$：13维的实数向量，描述红酒的属性
  - 标签定义：
  - $y \in \{0,1,2\}$：整数，代表红酒类型
  - 模型定义：
  - $f(x) = Wx + b \in \mathbb{R}^3$, 其中$W \in \mathbb{R}^{3\times13}$，$b \in \mathbb{R}^3$
  - $\theta = \{W, b\}$是模型的参数，需要优化

属性	值
固定酸度	8.319637
挥发物	0.527821
柠檬酸	0.270976
糖分	2.538806
氯化物	0.087467
游离SO_2	15.874922
总SO_2	46.467792
密度	0.996747
PH值	3.311113
硫酸盐	0.658149
酒精度	10.422983

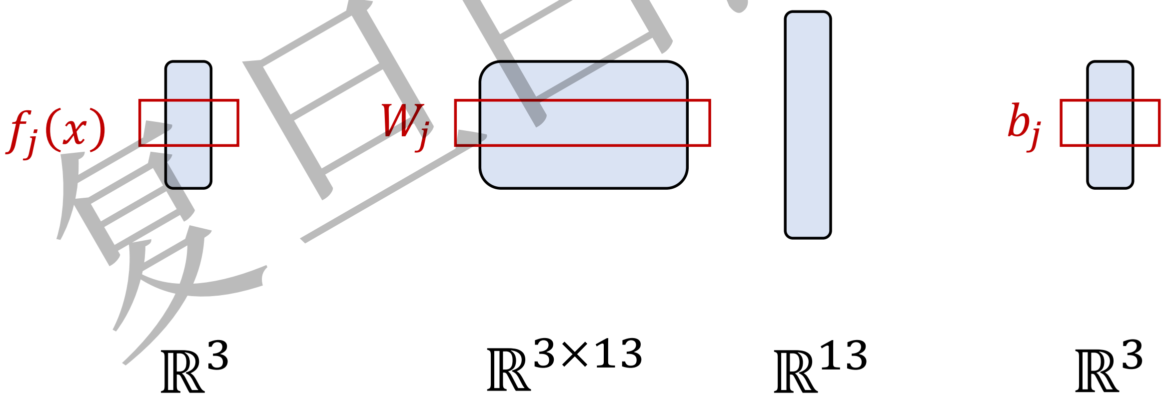
红酒类型：$\{0,1,2\}$

红酒等级预测：多维线性模型

- 理解多维线性模型：

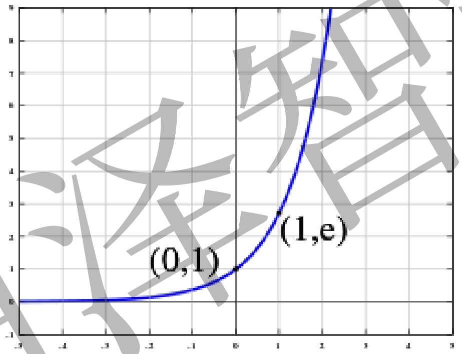
$f_j(x) = \sum_{k=1}^{13} W_{jk} x_k + b_j$，表示模型预测红酒为第$j$类型的置信度

$$f(x) \quad = \quad W \quad \cdot \quad x \quad + \quad b$$



红酒等级预测：损失函数

- 再$f(x)$做一种特定的转化：
- $\tilde{f}(x) = \mathrm{Softmax}\big(f(x)\big) \in \mathbb{R}^3$
- 其中$\tilde{f}_j(x) = \dfrac{\exp\big(f_j(x)\big)}{\sum_{j=1}^{5} \exp\big(f_j(x)\big)}$
- 先通过$\exp(\cdot)$函数把置信度都映射成正数
- 再对置信度做归一化
- 例如$\tilde{f}(x) = [0.7, 0.1, 0.2]$
- $\tilde{f}_j(x)$就表示模型认为此款红酒类型为j的概率



属性	值
固定酸度	8.319637
挥发物	0.527821
柠檬酸	0.270976
糖分	2.538806
氯化物	0.087467
游离SO_2	15.874922
总SO_2	46.467792
密度	0.996747
PH值	3.311113
硫酸盐	0.658149
酒精度	10.422983

红酒类型：{0,1,2}

红酒等级预测：损失函数

- 接下来是损失函数$\ell(\theta)$的构建
  - 先对y做一种特定的转化：
  - $\tilde{y} = \mathrm{OneHot}(y) \in \mathbb{R}^3$
  - 当$y = 0$时，$\tilde{y} = [1,0,0]$
  - 当$y = 2$时，$\tilde{y} = [0,0,1]$
  - y_j只有在$j = y$的位置等于1，否则等于0

属性	值
固定酸度	8.319637
挥发物	0.527821
柠檬酸	0.270976
糖分	2.538806
氯化物	0.087467
游离SO_2	15.874922
总SO_2	46.467792
密度	0.996747
PH值	3.311113
硫酸盐	0.658149
酒精度	10.422983

红酒类型：$\{0,1,2\}$

红酒等级预测：损失函数

- 接下来是损失函数$\ell(\theta)$的构建
  - 经过转换后
  - $\tilde{y} = \mathrm{OneHot}(y) = [0,0,1]$
  - $\tilde{f}(x) = \mathrm{Softmax}\big(f(x)\big) = [0.7, 0.1, 0.2]$
  - 交叉熵损失函数（Cross Entropy）
  - $\ell(\theta) = \sum_{j=3} -\tilde{y}_j \cdot \ln \tilde{f}_j(x)$
  - 让类型$j = y$的概率值尽可能的大
  - 当$\ell(\theta) = 0$时， $\tilde{f}(x) = [0,0,1]$

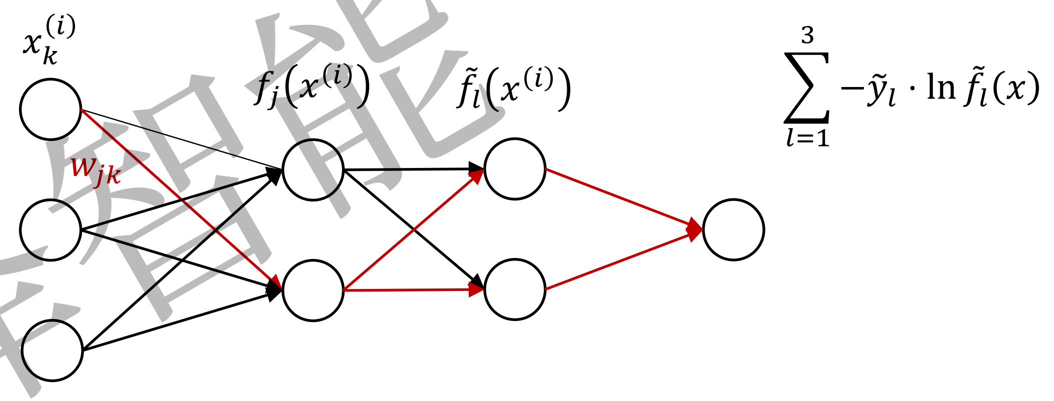
属性	值
固定酸度	8.319637
挥发物	0.527821
柠檬酸	0.270976
糖分	2.538806
氯化物	0.087467
游离SO_2	15.874922
总SO_2	46.467792
密度	0.996747
PH值	3.311113
硫酸盐	0.658149
酒精度	10.422983

红酒类型：$\{0,1,2\}$

红酒等级预测：梯度下降

- 考虑梯度下降算法中的偏导数：
  - 交叉熵损失函数：

$$\ell(w,b) = -\sum_{i=1}^{N}\sum_{l=1}^{3} \tilde{y}_l^{(i)} \ln \tilde{f}_l(x^{(i)})$$



  - 求偏导（链式求导法则）：

$$\nabla_{w_{jk}} \ell(w,b) = \sum_{i=1}^{N}\sum_{l=1}^{3} \frac{\partial[-\tilde{y}_l^{(i)} \cdot \ln \tilde{f}_l(x^{(i)})]}{\partial \tilde{f}_l(x^{(i)})} \cdot \frac{\partial \tilde{f}_l(x^{(i)})}{\partial f_j(x^{(i)})} \cdot \frac{\partial f_j(x^{(i)})}{w_{jk}}$$

交叉熵损失函数求导　Softmax函数求导　线性模型求导

$$\underbrace{\text{交叉熵损失函数求导} \quad \text{Softmax函数求导}}_{-\sum_{i=1}^{N}(\tilde{y}_j^{(i)} - \tilde{f}_j(x^{(i)}))}$$

$x_k^{(i)}$

红酒等级预测：梯度下降

- 考虑梯度下降算法中的偏导数:
  - 损失函数:

$$\ell(w,b) = -\sum_{i=1}\sum_{l=1}^{3}\tilde{y}_l^{(i)}\ln\tilde{f}_l(x^{(i)})$$

  - 求偏导（链式求导法则）：

$$\nabla_{w_{jk}}\ell(w,b) = -\sum_{i=1}^{N}(\tilde{y}_j^{(i)} - \tilde{f}_j(x^{(i)}))\cdot x_k^{(i)}$$

$$\nabla_{b_j}\ell(w,b) = -\sum_{i=1}^{N}(\tilde{y}_j^{(i)} - \tilde{f}_j(x^{(i)}))$$

有监督学习：红酒类型预测

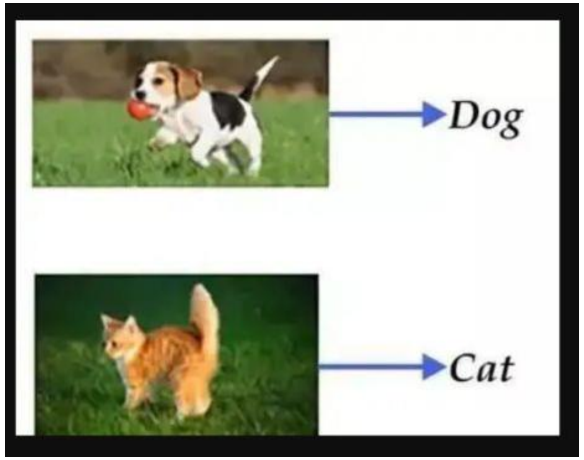
- 应用上述框架：
  - 准备工作：
    - 收集**训练集**$(x^{(i)}, y^{(i)})$ -> 收集每款红酒参数和类型
    - 建立模型$f(x)$ -> $f(x) = Wx + b$，多维线性模型
  - 模型**训练**：
    - 设定**损失函数**$\ell(\theta)$ -> 交叉熵损失函数
    - 通过梯度下降最小化$\ell(\theta)$
  - 模型**预测**：在新数据点上计算$f(x)$
    - $f_j(x)$最大的j即为预测的等级

属性	值
固定酸度	8.319637
挥发物	0.527821
柠檬酸	0.270976
糖分	2.538806
氯化物	0.087467
游离SO_2	15.874922
总SO_2	46.467792
密度	0.996747
PH值	3.311113
硫酸盐	0.658149
酒精度	10.422983

红酒类型：$\{0,1,2\}$

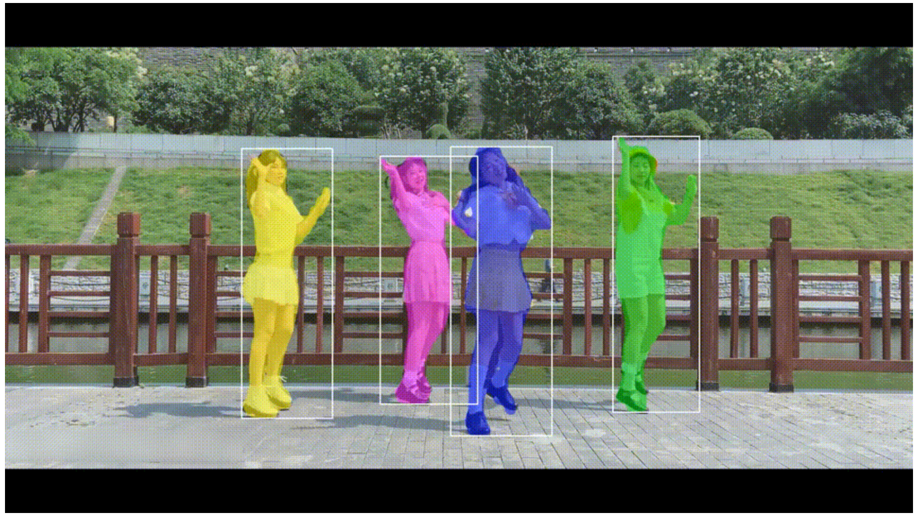
有监督学习：图像分类

- 应用上述框架：
  - 准备工作：
    - 收集**训练集**$(x^{(i)}, y^{(i)})$ -> 收集大量的图片和类别
    - 建立模型$f(x)$ -> 卷积神经网络
  - 模型**训练**：
    - 设定**损失函数**$\ell(\theta)$ -> 交叉熵损失函数
    - 通过梯度下降最小化$\ell(\theta)$ -> Batch SGD
  - 模型**预测**：在新数据点上计算$f(x)$



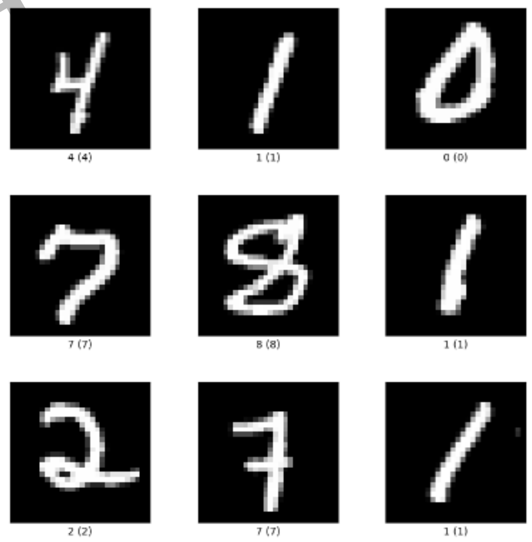
有监督学习：实例分割

- 应用上述框架：
  - 准备工作：
    - 收集**训练集**$(x^{(i)}, y^{(i)})$ -> 收集大量的图片和每个像素点的标注
    - 建立模型$f(x)$ -> 全卷积神经网络等复杂结构，输出每个像素点的类别
  - 模型**训练：**
    - 设定**损失函数**$\ell(\theta)$ -> 交叉熵损失函数
    - 通过梯度下降最小化$\ell(\theta)$
  - 模型**预测：**在新数据点上计算$f(x)$



有监督学习：应用范围

- 神经网络也不是万能的
- 测试数据需要跟训练数据比较接近
    - 比如运营方想要知道直播间10亿人时候销量会是多少
    - 比如输入一张旋转过的图片



Q&A

实验内容1：线性模型

- 上传jupyter notebook文件至服务器
- 实现下述两个模块：
-     softmax计算函数
-     模型预测函数
- 通过提供的实例验证实现的正确性

实验内容2：梯度下降

- 实现下述两个模块：
- 损失函数计算
- 梯度计算公式
- 通过提供的实例验证实现的正确性

实验内容3：线性模型

- 上传trainset.npy和testset.npy文件至服务器
- 实现训练阶段的核心代码
- 运行模块代码，通过观察损失函数的下降来验证实现的正确性

实验内容4：模型正确性验证

- 实现准确度计算函数
- 并计算准确度
- 验证模型在训练集和测试集上的准确度

实验内容5：超参数验证

- 尝试不同的训练轮数和学习速率
- 重新运行代码
- 记录训练集和测试集上的准确度

Q&A

复旦白泽智能

系统软件与安全实验室