线性神经网络

线性回归

1. 线性模型

1.1 基本概念

- 给定 n 维输入 $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$
- 线性模型有一个 n 维权重和一个标量偏差

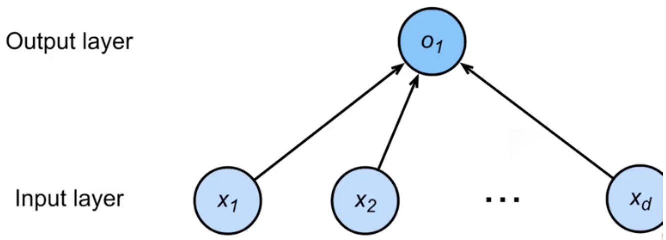
$$\mathbf{w} = [w_1, w_2, \ldots, w_n]^T, \quad b$$

- 输出是输入的加权和

$$y = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$$

向量版本：$y = \langle \mathbf{w}, \mathbf{x} \rangle + b$

- 线性模型可以看做是单层的神经网络



1.2 衡量预估质量

比较真实值和预估值，例如：平方损失

$$\ell(y, \hat{y}) = \frac{1}{2}\left(y - \hat{y}\right)^2$$

1.3 参数学习

- 训练损失(**n 表示的是样本数量**)

$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \langle \mathbf{x}_i, \mathbf{w} \rangle - b\right)^2 = \frac{1}{2n}\left\| \mathbf{y} - \mathbf{X}\mathbf{w} - b \right\|^2$$

- 最小化损失来学习参数

$$\mathbf{w}^*, \mathbf{b}^* = \arg\min_{\mathbf{w}, b} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b)$$

- 显示解

  - 将偏差加入权重 $\mathbf{X} \leftarrow [\mathbf{X}, \mathbf{1}]$ $\quad \mathbf{w} \leftarrow \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$

$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{1}{2n}\left\| \mathbf{y} - \mathbf{X}\mathbf{w} \right\|^2 \quad \frac{\partial}{\partial \mathbf{w}}\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{1}{n}\left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)^T \mathbf{X}$$

- 损失是凸函数，所以最优解满足

$$\frac{\partial}{\partial \mathbf{w}}\ell(\mathbf{X},\mathbf{y},\mathbf{w}) = 0$$

$$\Leftrightarrow \frac{1}{n}\left(\mathbf{y}-\mathbf{X}\mathbf{w}\right)^T\mathbf{X} = 0$$

$$\Leftrightarrow \mathbf{w}^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}\mathbf{y}$$

注：线性模型 -> 损失是凸函数 -> 具有最优解（**唯一一个具有最优解的模型**）

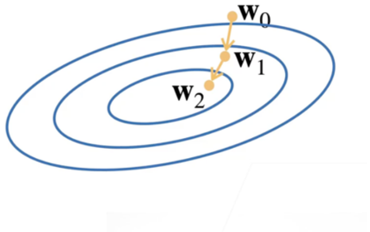
2. 基础优化方法

2.1 梯度下降

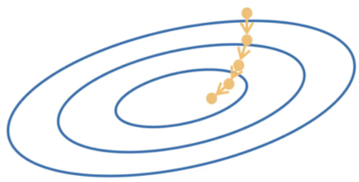
- 基本概念
  梯度下降通过不断沿着反梯度方向更新参数求解
  学习率：步长的超参数

  - 挑选一个初始值 $\mathbf{w}_0$
  - 重复迭代参数 t=1,2,3

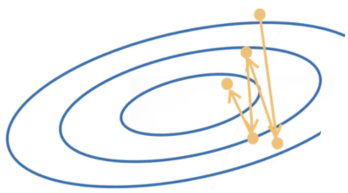
  $$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta\frac{\partial \ell}{\partial \mathbf{w}_{t-1}}$$



- 学习率选择
  a) 不能太小



  b) 不能太大



2.2 小批量随机梯度下降

- 小批量随机梯度下降算法是深度学习默认的求解算法
- 在整个训练集上算梯度太贵
  一个深度学习的模型可能需要数分钟甚至数小时
- 可以随机采样 b 个样本$i_1, i_2 \dots i_b$来近似损失，，b 是批量大小，需要设置的超参数

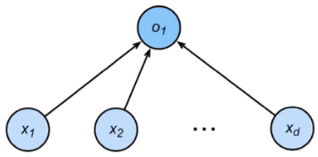
$$\frac{1}{b}\sum_{i\in I_b}\ell(\mathbf{x}_i, y_i, \mathbf{w})$$

Softmax 回归

1. 回归和分类

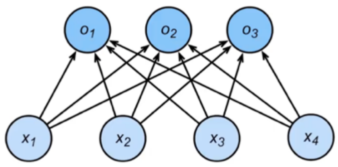
- **回归估计一个连续值**

  - 单连续数值输出
  - 自然区间 $\mathbb{R}$
  - 跟真实值的区别作为损失



- **分类预测一个离散类别**
  输出值$\boldsymbol{o_i}$当作预测类别是$\boldsymbol{i}$ 的置信度

  - 通常多个输出
  - 输出 i 是预测为第 i 类的置信度



2. 从回归到多类分类

2.1 均方损失

- 对类别进行一位有效编码(i.e. one-hot)

$$\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$$
$$y_i = \begin{cases} 1 \text{ if } i = y \\ 0 \text{ otherwise} \end{cases}$$

- 使用均方损失训练
- 将值最大的输出所对应的类作为预测输出

$$\hat{y} = \operatorname*{argmax}_i o_i$$

2.2 无效验比例

- 需要更置信的识别正确类(大余量)

$$o_y - o_i \geq \Delta(y, i)$$

其中，$\boldsymbol{o_i}$是预测其他类的概率，$\boldsymbol{o_y}$是预测正确类的概率

2.3 效验比例

- 输出**匹配概率**(非负，和为 1)

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o})$$

$$\hat{y}_i = \frac{\exp(o_i)}{\sum_k \exp(o_k)}$$

2.4 softmax 和交叉熵损失

- 交叉熵常用来衡量两个概率的区别

$$H(\mathbf{p}, \mathbf{q}) = \sum_i - p_i \log(q_i)$$

- 将其作为损失

$$l(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_i y_i \log \hat{y}_i = - \log \hat{y}_y$$

- 其梯度是真实概率和预测概率的区别

$$\partial_{o_i} l(\mathbf{y}, \hat{\mathbf{y}}) = \mathrm{softmax}(\mathbf{o})_i - y_i$$