深蓝学院
shenlanxueyuan.com

神经网络

主讲人 夏至

纲要

- 第一部分：作业1 感知机
- 第二部分：作业2 神经网络
- 第三部分：作业3 参数计算
- 第四部分：手写数字分类器编程练习

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：作业1 感知机
- 第二部分：作业2 神经网络
- 第三部分：作业3 参数计算
- 第四部分：手写数字分类器编程练习

感知机

现有如下训练数据集，正样本点$M_1(3,3)$和$M_2(4,3)$，负样本点$M_3(1,1)$，请利用感知机算法求感知机模型$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w} \cdot \boldsymbol{x} + b)$。下方表格中已给出了前4步迭代的步骤，第5步迭代中选择M_2为误分类点，请补全后续详细的迭代过程。

迭代次数	误分类点	w	b	超平面
0		(0，0)	0	0
1	M_1	(3，3)	1	$3x_1 + 3x_2 + 1$
2	M_3	(2，2)	0	$2 + 2x_2$
3	M_3	(1，1)	-1	$x_1 + x_2 - 1$
4	M_3	(0，0)	-2	-2
5	M_2			
…				
N	无			

感知机

1. 取初值$\boldsymbol{w}_0 = \boldsymbol{0}, b_0 = 0$

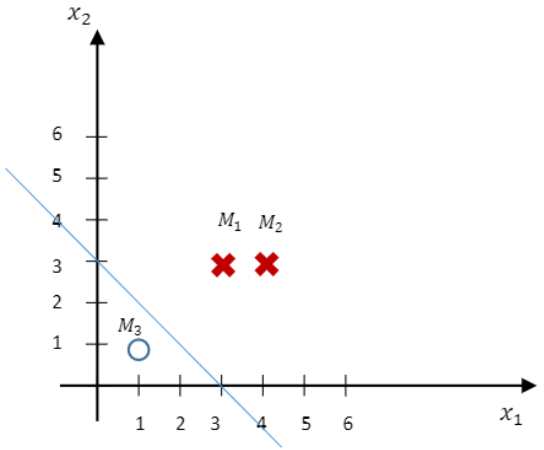
2. M_1对应的感知机模型输出为0，未被正确分类，更新：

$$\boldsymbol{w}_1 = \boldsymbol{w}_0 + y_1\boldsymbol{x}_{M1} = (3,3) \quad b_1 = b_0 + y_1 = 1$$

3. M_1和M_2已经满足，M_3对应感知机输出为1，未被正确分类，更新：

$$\boldsymbol{w}_2 = \boldsymbol{w}_1 + y_3\boldsymbol{x}_{M3} = (2,2) \quad b_2 = b_1 + y_3 = 0$$

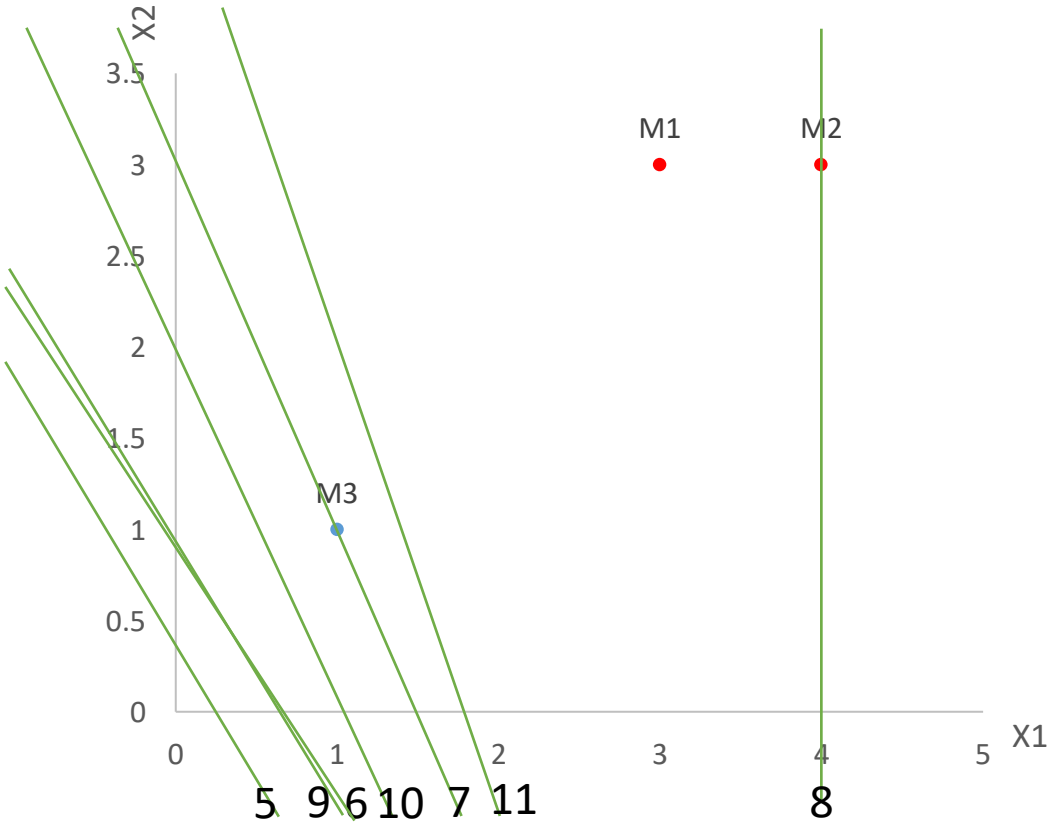
4. 重复更新步骤，直至没有误分类点。



参数更新：

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + y_i\boldsymbol{x}_i$$
$$b_{n+1} = b_n + y_i$$

感知机



迭代次数	误分类点	w	b	超平面
0		（0，0）	0	0
1	M1	（3，3）	1	$3x_1+3x_2+1$
2	M3	（2，2）	0	$2x_1+2x_2$
3	M3	（1，1）	-1	x_1+x_2-1
4	M3	（0，0）	-2	-2
5	**M2**	**（4，3）**	**-1**	**$4x_1+3x_2-1$**
6	**M3**	**（3，2）**	**-2**	**$3x_1+2x_2-2$**
7	**M3**	**（2，1）**	**-3**	**$2x_1+x_2-3$**
8	**M3**	**（1，0）**	**-4**	**x_1-4**
9	**M2**	**（5，3）**	**-3**	**$5x_1+3x_2-3$**
10	**M3**	**（4，2）**	**-4**	**$4x_1+2x_2-4$**
11	**M3**	**（3，1）**	**-5**	**$3x_1+x_2-5$**
12	无	（3，1）	-5	$3x_1+x_2-5$

感知机模型： $f(\mathbf{x})=\text{sign}(\mathbf{w}\bullet\mathbf{x}+b)$

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：作业1 感知机
- **第二部分：作业2 神经网络**
- 第三部分：作业3 参数计算
- 第四部分：手写数字分类器编程练习

神经网络

为什么多层神经网络可以拟合任意函数？其与浅层宽网络的区别是什么？

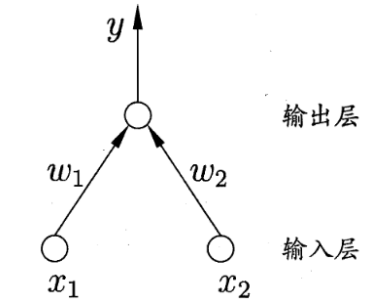
- 从感知机的角度看
- 从决策边界的角度看

神经网络

深蓝学院 shenlanxueyuan.com

●感知机

感知机由两层神经元组成，激活函数设为阶跃函数，如右图所示。



考虑简单的逻辑运算:

对于“与”运算（$x_1 \wedge x_2$）:

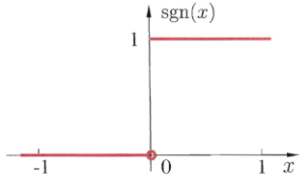
可令$w_1 = w_2 = 1$，$b = -1.5$，则$y = f(x_1 + x_2 - 1.5)$

对于“或”运算（$x_1 \vee x_2$）:

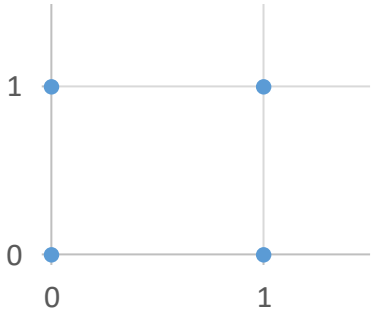
可令$w_1 = w_2 = 1$，$b = -0.5$，则$y = f(x_1 + x_2 - 0.5)$

对于“非”运算（$\neg x_1$）:

可令$w_1 = -1$，$w_2 = 0$，$b = 0.5$，则$y = f(-x_1 + 0.5)$



$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geqslant 0; \\ 0, & x < 0. \end{cases}$$

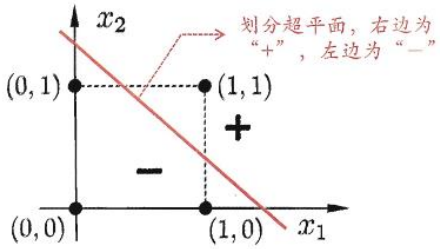


神经网络

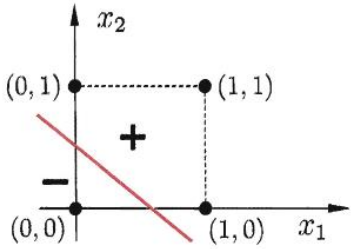
- 感知机

对于“异或”运算（$x_1 \oplus x_2$）？

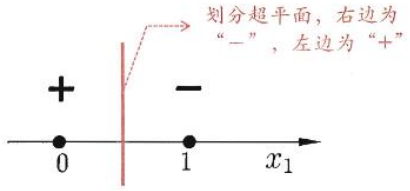
事实上，如果感知机只有一层神经元，学习能力非常有限。上述的与、或、非问题都是线性可分问题，可以证明存在一个线性超平面将其分开。但对于异或这样简单的非线性问题，单层感知机不能解决。



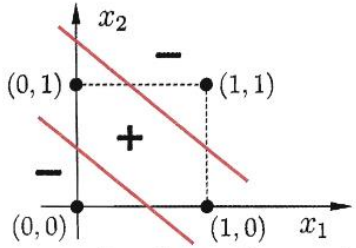
(a) “与”问题 $(x_1 \wedge x_2)$



(b) “或”问题 $(x_1 \vee x_2)$



(c) “非”问题 $(\neg x_1)$



(d) “异或”问题 $(x_1 \oplus x_2)$

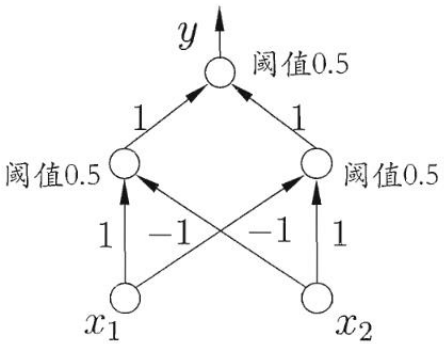
神经网络

●感知机

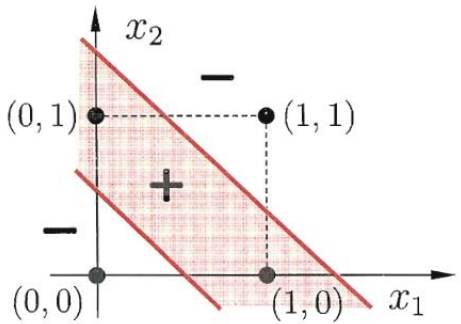
那么，多层感知机可以实现异或问题吗?

考虑在之前单层感知机的基础上增加一层隐含层，网络结构如下图所示:

只要我们设置好合适的权重和阈值，则异或问题迎刃而解。



(a) 网络结构



(b) 分类区域

神经网络

- 感知机

由此可见，单层神经网络可以解决线性问题，多层神经网络则将单层神经网络联立起来，解决了非线性问题。

万能近似定理表明，两层神经网络已经可以拟合任意函数。

神经网络

- 决策边界

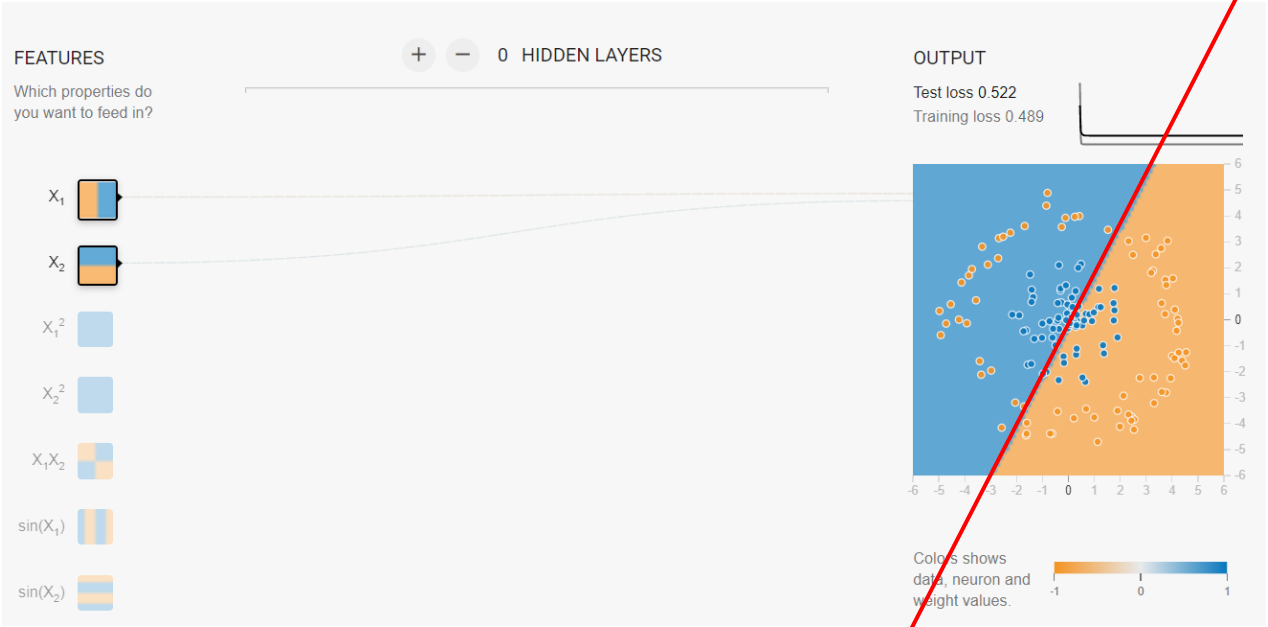
单层神经网络只能划分出一条超平面，可以解决线性可分问题；

多层神经网络可以划分出多条超平面，可以解决线性不可分问题；

神经网络

深蓝学院
shenlanxueyuan.com

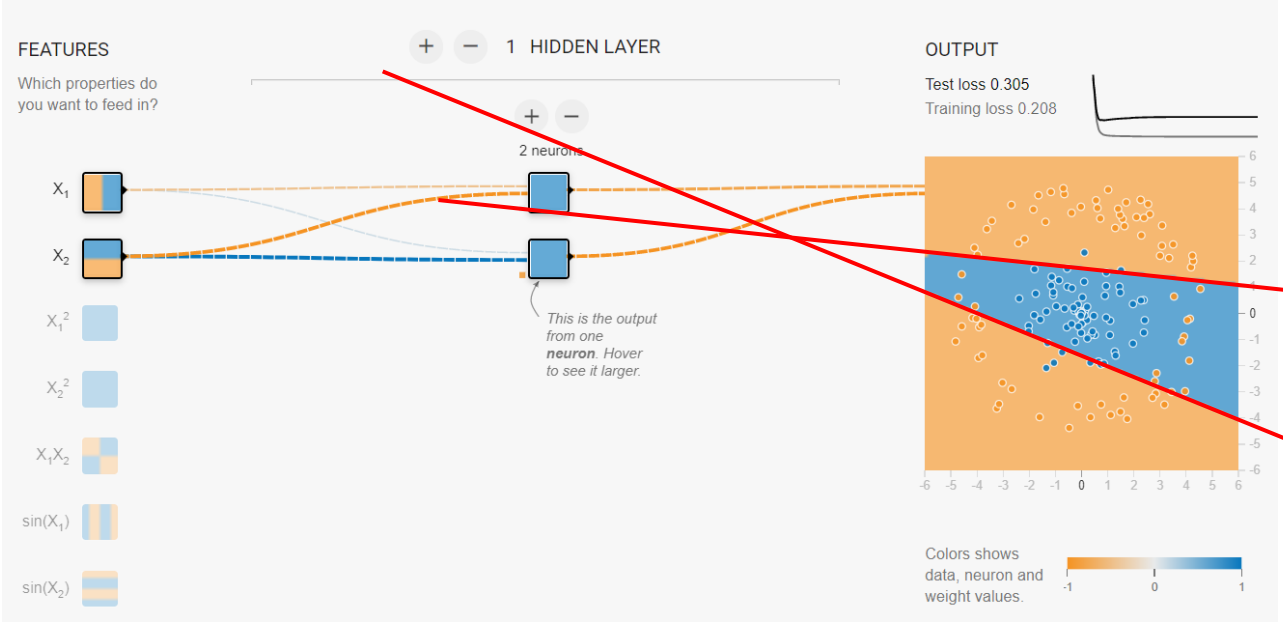
- 决策边界



神经网络

深蓝学院
shenlanxueyuan.com

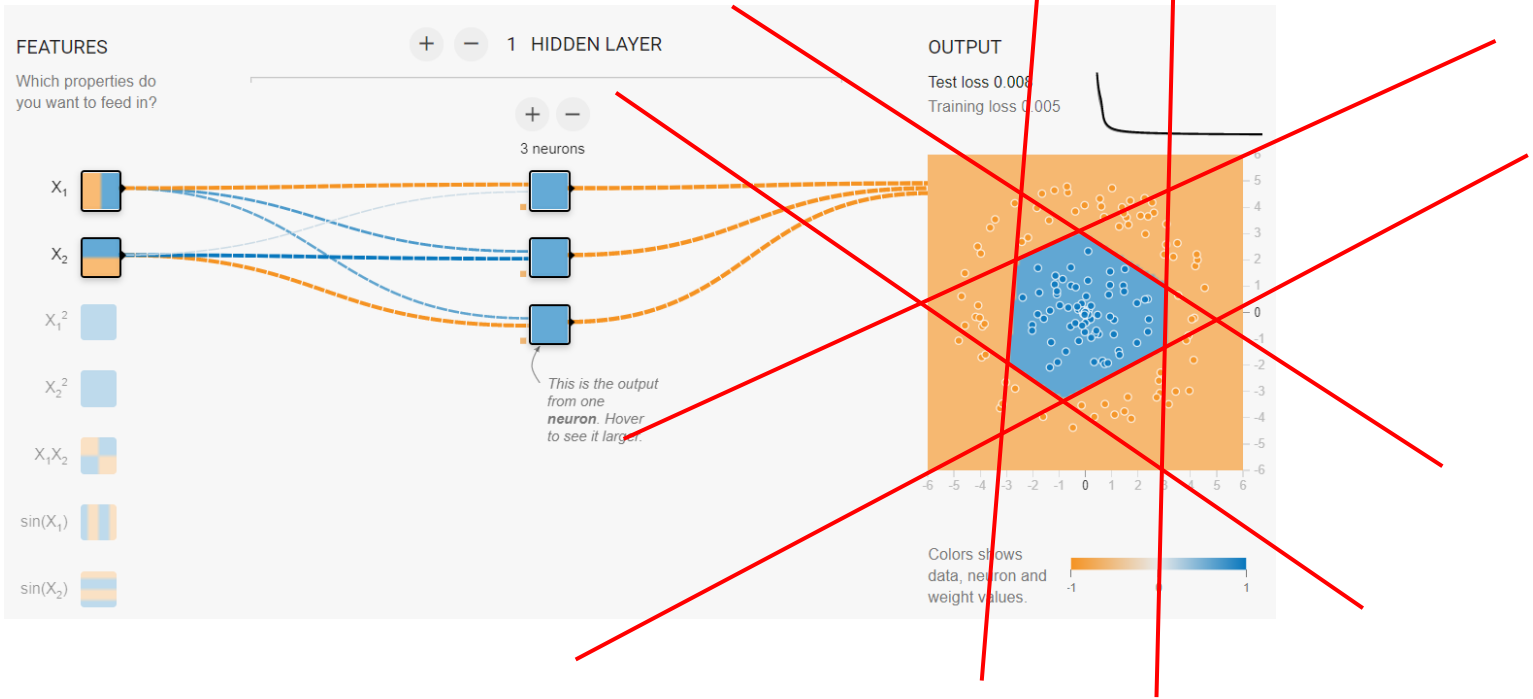
- 决策边界



神经网络

深蓝学院
shenlanxueyuan.com

- 决策边界



纲要

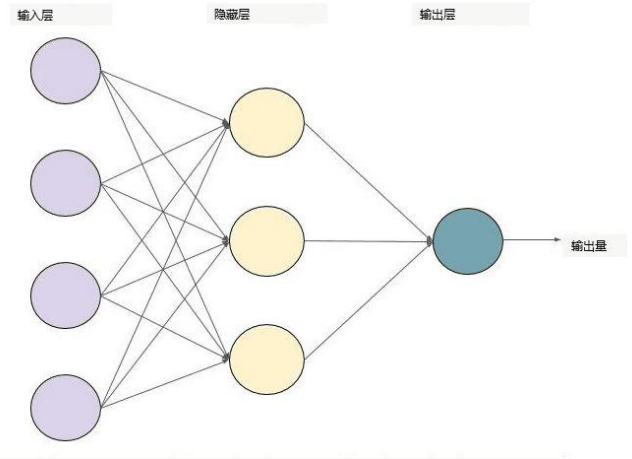
深蓝学院
shenlanxueyuan.com

- 第一部分：作业1 感知机
- 第二部分：作业2 神经网络
- **第三部分：作业3 参数计算**
- 第四部分：手写数字分类器编程练习

参数计算

- 手写数字识别案例中，如果输入的图片尺寸是28*28大小，则输入层维度为：28*28=784
- 整个网络结构为：[784 30 10]
- 则参数量为：
  - 输入层到隐藏层权重：784*30=23520
  - 输入层到隐藏层偏置：30
  - 隐藏层到输出层权重：30*10=300
  - 隐藏层到输出层偏置：10

  因此，参数量共为：(784+1)*30+(30+1)*10=23860

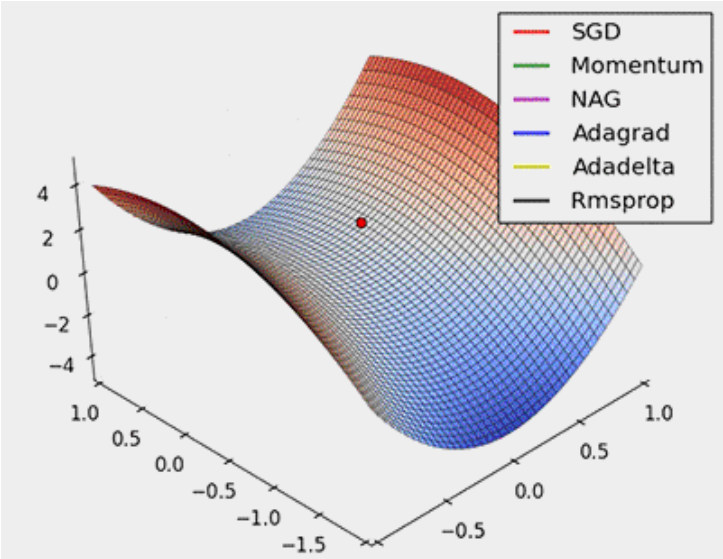


- 假设图片变成了256*256，则输入层维度为：256*256=65536
- 整个网络结构为：[65536 30 10]
- 计算方法与之前一样，参数量共为:(65536+1)*30+(30+1)*10=1966420

参数计算

深蓝学院
shenlanxueyuan.com

- 降低计算复杂度
  - 采用更合适的梯度下降方法
  - 减小输入数据维度
- 提升计算速度
  - 并行加速
  - GPU加速
  - 指令集加速



纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：作业1 感知机
- 第二部分：作业2 神经网络
- 第三部分：作业3 参数计算
- **第四部分：手写数字分类器编程练习**

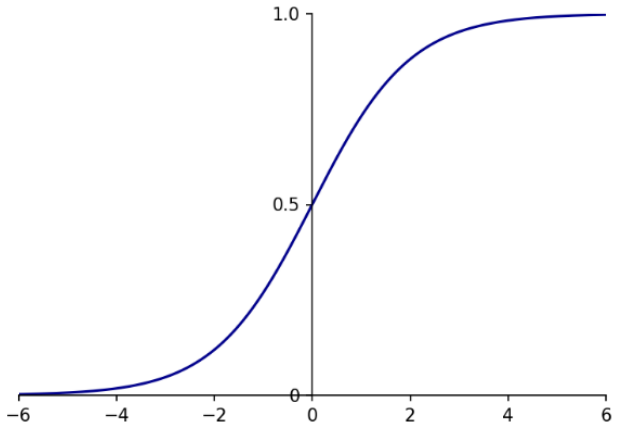
手写数字分类器编程练习

- def sigmoid(z):

  return 1/(1 + np.exp(-z))

- def sigmoid_prime(z):

  return sigmoid(z)*(1-sigmoid(z))



$$f(x) = \frac{1}{1+e^{-x}}$$

$$f'(x) = -\frac{1}{(1+e^{-x})^2} * e^{-x} * (-1)$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

$$= \frac{1}{1+e^{-x}} * \frac{e^{-x}}{1+e^{-x}}$$

$$= f(x) * (1 - f(x))$$

手写数字分类器编程练习

- train

  对于每轮训练：

  打乱训练集数据，将训练集划分成若干个batch；

  每次用batch更新神经网络参数：

  计算出每个batch平均的权重和偏置更新情况

  再乘上一个预设的学习率来控制梯度下降收敛情况

  然后更新权重和偏置

- evaluate

  前向计算神经网络输出，获取输出向量的最大索引，即为预测值。统计所有测试集数据的预测值与真实值差异返回准确率

手写数字分类器编程练习

深蓝学院 shenlanxueyuan.com

- [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15] training_data
- [2,5,7,9,13,12,0,8,10,3,1,15,4,14,6,11] random.shuffle
- [2,5,7,9],[13,12,0,8],[10,3,1,15],[4,14,6,11] mini_batches
- 假设先对第一个mini_batch[2,5,7,9]进行update
  - 初始设置nabla_b， nabla_w，然后遍历mini_batch的所有数据
    - training_data[2][0]: 784*1 training_data[2][1]:10*1
    - backprop计算delta_nabla_b，delta_nabla_w,然后更新nabla_b， nabla_w
    - 在这个过程中weights和biases保持不变
  - 完成所有mini_batch的遍历，我们可以获得最后的nabla_b， nabla_w，然后除以len(mini_batch)以获得mini_batch内数据的平均nabla_b， nabla_w，为了更好的收敛，我们需要通过设置学习率eta来控制梯度下降步长，因此，对每个参数进行更新：

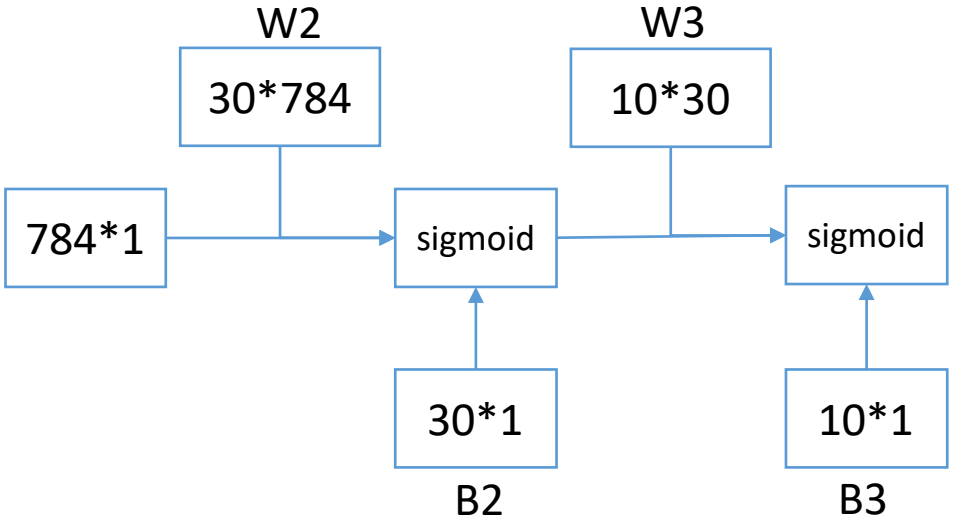
$$w_{ij}^{l} = w_{ij}^{l} - eta * \frac{\nabla w_{ij}^{l}}{len(\text{mini\_batch})}$$

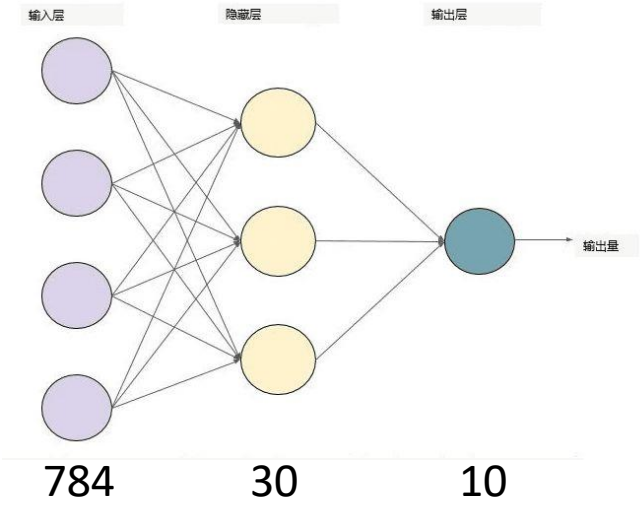
$$b_{i}^{l} = b_{i}^{l} - eta * \frac{\nabla b_{i}^{l}}{len(\text{mini\_batch})}$$

手写数字分类器编程练习

深蓝学院 shenlanxueyuan.com

- Backprop——前向计算





- 存储神经网络每层输出a和z，用于计算delta

手写数字分类器编程练习

深蓝学院 shenlanxueyuan.com

- Backprop——反向传播

cost_derivative sigmoid_prime

$$\delta_i^{(L)} = -\left(y_i - a_i^{(L)}\right) f'\left(z_i^{(L)}\right)$$ 输出层

$$\delta_i^{(l)} = \left(\sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \Theta_{ji}^{(l+1)}\right) f'\left(z_i^{(l)}\right)$$ 隐藏层

$$\frac{\partial E}{\partial \Theta_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)}$$

$$\frac{\partial E}{\partial b_i^{(l)}} = \delta_i^{(l)}$$

$$\Theta^{(l)} = \Theta^{(l)} - \mu \frac{\partial E_{total}}{\partial \Theta^{(l)}}$$

$$= \Theta^{(l)} - \frac{\mu}{N} \sum_{i=1}^{N} \frac{\partial E_{(i)}}{\partial \Theta^{(l)}}$$

$$\boldsymbol{b}^{(l)} = \boldsymbol{b}^{(l)} - \mu \frac{\partial E_{total}}{\partial \boldsymbol{b}^{(l)}}$$

$$= \boldsymbol{b}^{(l)} - \frac{\mu}{N} \sum_{i=1}^{N} \frac{\partial E_{(i)}}{\boldsymbol{b}^{(l)}}$$

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening