

# 13 人工神经网络3

---

## 13 人工神经网络3

### 13.1 本集内容简介

### 13.2 实验环节

#### 实验环节

#### 第一组实验

设置网络的层数和各层神经元数量

设置激活函数和参数

设置训练参数

生成训练样本

调用训练函数

#### 第二组实验

数据集A

数据集B

### 13.3 理论分析

### 13.4 理论分析-拟合能力

### 13.5 理论分析- 与神经系统的关系

### 13.6 实现细节问题

### 13.7 输入值和输出值设定

### 13.8 网络的规模

### 13.9 激活函数的选择

为什么需要激活函数？

什么样的函数可以用作激活函数？

### 13.10 损失函数的选择

### 13.11 权重的初始化

caffe

### 13.12 正则化

### 13.13 学习率的设定

### 13.14 动量项

### 13.15 挑战与改进措施

### 13.16 梯度消失问题

梯度消失问题

梯度爆炸问题

### 13.17 退化

### 13.18 局部极小值

### 13.19 鞍点问题

### 13.20 损失函数曲面分析

### 13.21 实际应用

### 13.22 实战项目

### 13.23 本集总结

## 13.1 本集内容简介

---

实验环节  
理论解释  
实现细节问题  
挑战与改进措施  
实际应用

- 实验环节 我们用C++代码来写
- 理论解释
- 实现细节问题
- 挑战与改进措施
- 实际应用

## 13.2 实验环节

---

### 实验环节

---

实验环节

设置网络的层数和各层神经元数量

设置激活函数和参数

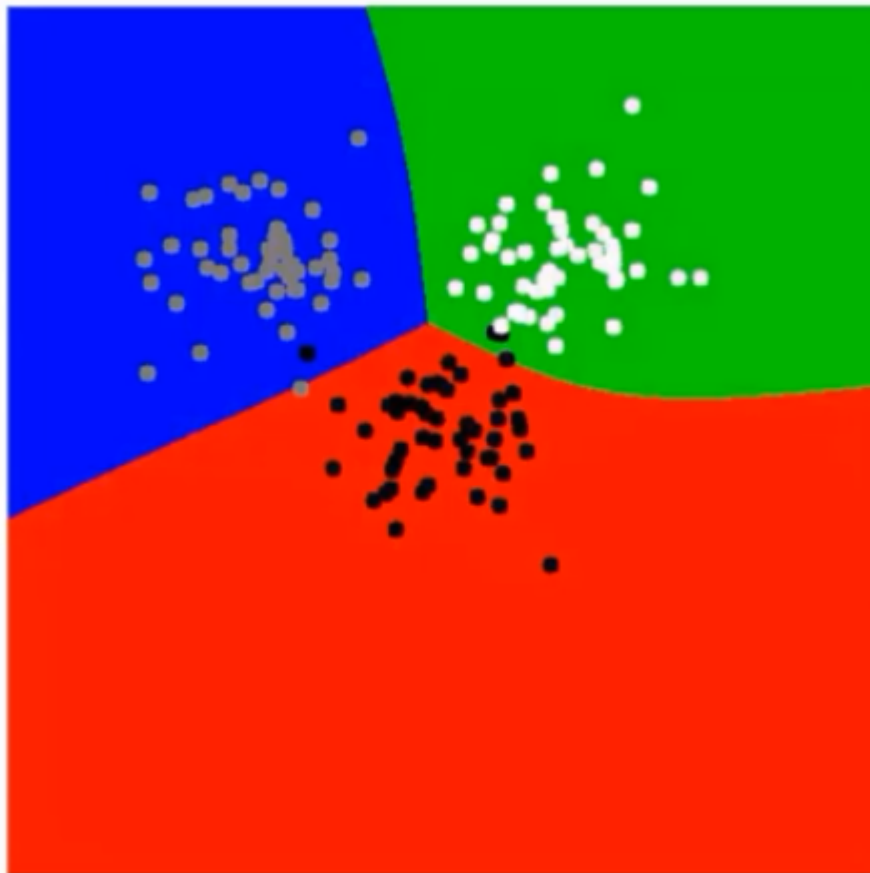
设置训练参数，包括最大迭代次数，迭代终止阈值，学习率，动量项系数等

生成训练样本，关键是设置样本标签值

调用训练函数

### 第一组实验

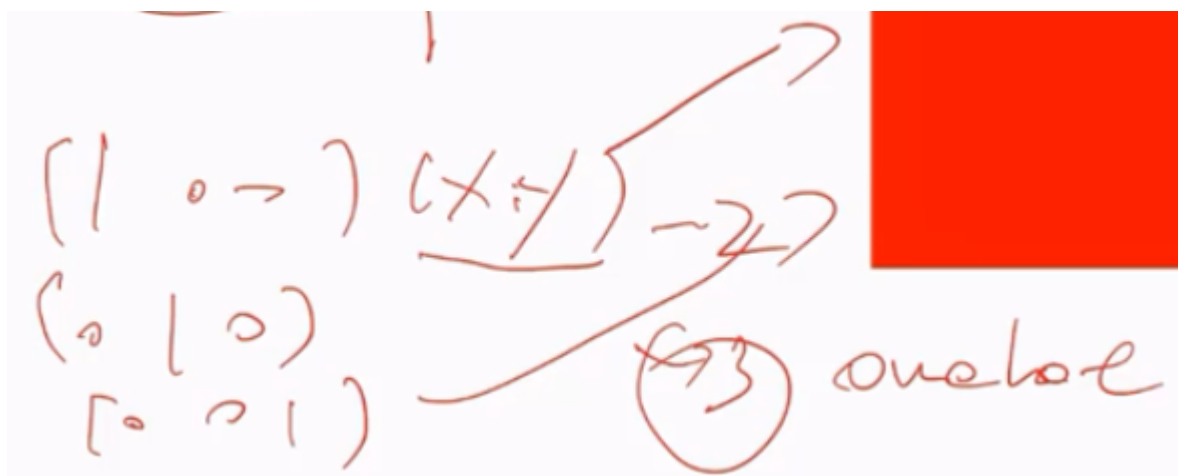
自己写 c++ 和基于openCV实现的神经网络



### 设置网络的层数和各层神经元数量

隐含层数	2
隐含层1神经元数	10
隐含层2神经元数	10

### 设置激活函数和参数



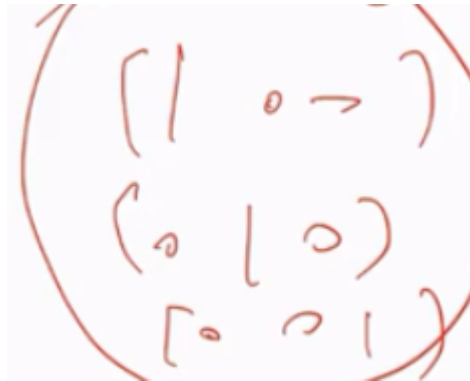
## 设置训练参数

包括最大迭代次数，迭代终止阈值，学习率，动量项系数等。

如果我们求出来的参数的梯度的值收敛。

## 生成训练样本

关键是设置样本标签值

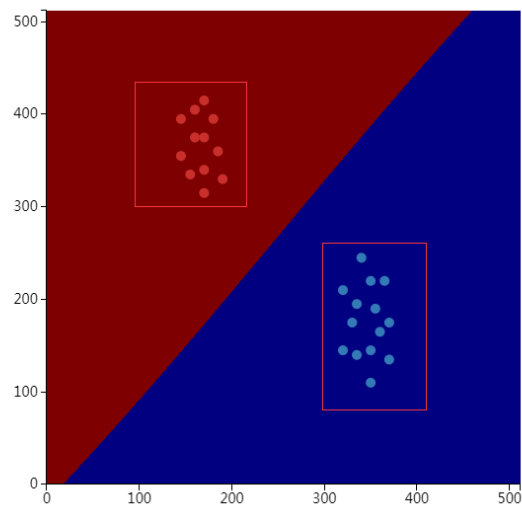


## 调用训练函数

```
Mat trainData(150, 2, CV_32FC1, trainDataArray);  
CvANN_MLP mlp,  
// 神经网络有3层，第一层2个神经元，对应于二维的特征向量。第二层有  
// 6个神经元，第三层有3个神经元，对应分类问题中的3个类别  
Mat layerSizes=(Mat_<int>(1,3) << 2, 6, 3);  
CvANN_MLP_TrainParams params;  
// 神经网络的训练参数，在后面会详细解释  
params.term_crit = cvTermCriteria( CV_TERMCRIT_ITER | CV_TERMCRIT_EPS,  
    1000, 0.001 );  
params.train_method = CvANN_MLP_TrainParams::BACKPROP;  
params.bp_dw_scale = 0.1;  
params.bp_moment_scale = 0.1;  
// 创建神经网络  
mlp.create(layerSizes, CvANN_MLP::SIGMOID_SYM);  
// 训练神经网络  
mlp.train(trainData, trainResponse, Mat(), Mat(), params);
```

## 第二组实验

## 数据集A



运行

清空

观看训练过程

隐含层数

2

隐含层1神经元数

10

隐含层2神经元数

10

添加样本

✓

使用预置数据集

数据集A

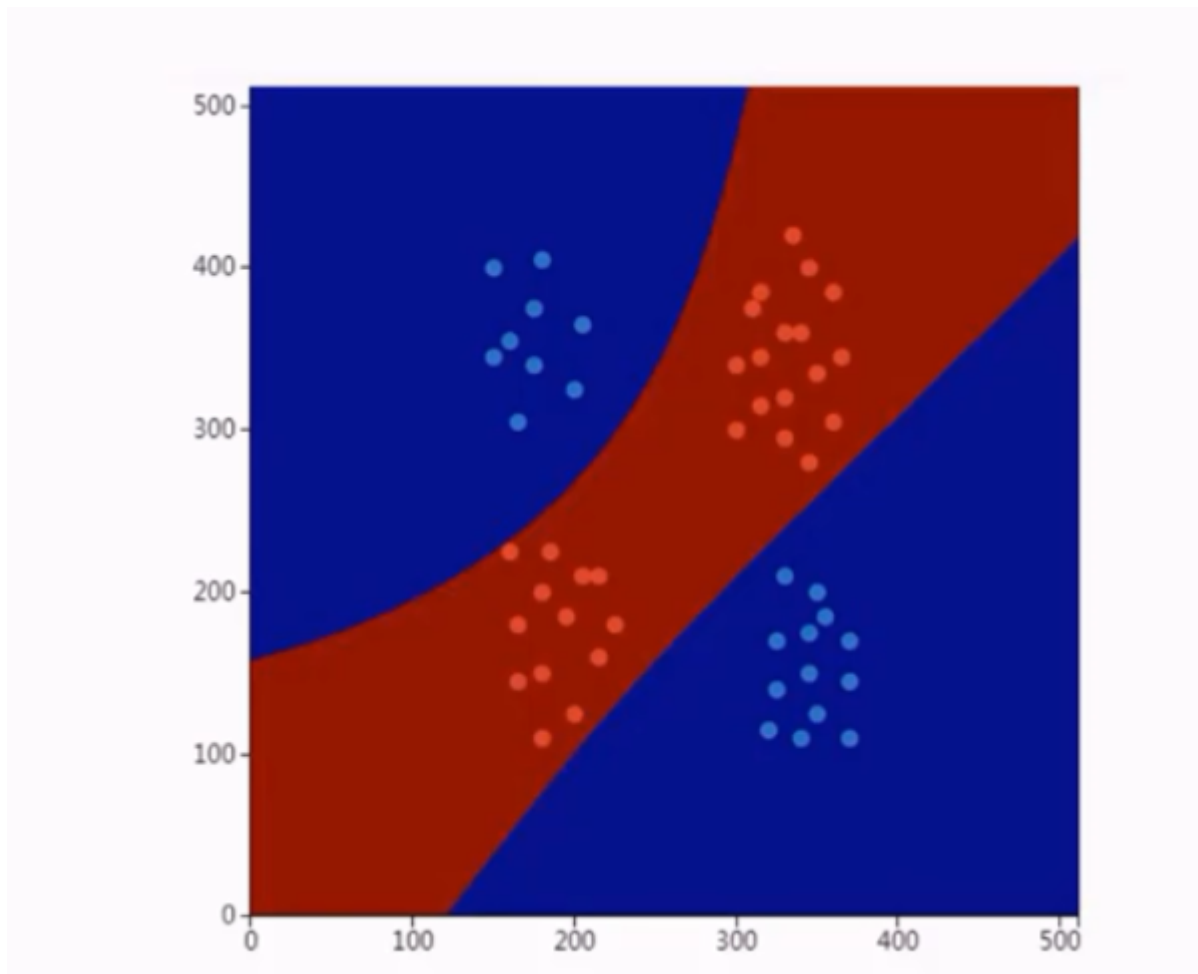
数据集B

数据集C

数据集D

## 数据集B

在云端实验室来完成，实验效果如下



1. batch

2. one

用的是随机梯度下降法

## 13.3 理论分析

---

理论解释

1.数学特性

2.与动物神经系统的关系

## 13.4 理论分析-拟合能力

---

神经网络的拟合能力

万能逼近定理

[1]Kurt Hornik. Approximation capabilities of multilayer feedforward networks. Neural Networks.1991.

[2]Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359-366, 1989.

$\varphi(x)$  非常数、有界且单调递增的连续函数

$I_m$   $m$ 维的单位立方体

$C(I_m)$  连续函数空间

$$F(x) = \sum_{i=1}^N v_i \varphi(w_i^T x + b_i)$$

$$|F(x) - f(x)| < \varepsilon$$

可以构造出上面这样的函数，逼近定义在单位立方体空间中的任何一个连续函数到任意指定的精度

先来看神经网络的数学特性

万能逼近定理

从理论上讲它的建模能力是非常强的。

## 13.5 理论分析- 与神经系统的关系

---

人工神经网络是对生物神经系统的模拟，但只是简单的模拟，在多个方面两者的机理是不同的。人脑的单个神经元有很复杂的结构，各个神经元在结构和功能上不是完全相同的，另外神经元之间的连接关系非常复杂。

在训练方式上，人脑的神经网络没有反向传播算法这种机制，在外界刺激下建立神经元之间连接通路的机制远比反向传播算法复杂。

前馈型人工神经网络本质上来说只是一个多层的复合函数。

人工神经网络本身是一种仿生的。

## 13.6 实现细节问题

- 1.输入与输出值的设定
- 2.网络的规模
- 3.激活函数的选择
- 4.损失函数的选择
- 5.权重的初始化
- 6.正则化
- 7.学习率的设定
- 8.动量项

实际上细节不止这些

## 13.7 输入值和输出值设定

数据归一化，输入值，输出值  
类别标签的设定

输入值与输出值设定

数据归一化，输入值，输出值  
类别标签的设定

$(0, 0, \dots, +1, 0, \dots, 0)$

- 数据归一化 (0,1] , (-1,+1)
- 输入值 x
- 输出值 y
- 类别标签的设定
  - 这同样也是需要考虑归一化
  - 标签数组
  - (0,0,...,+1,0,...0)

## 13.8 网络的规模

- 层数 早起很小，现在很深
- 各层神经元的数量，输入层和输出层是确定的，隐含层根据经验而定，一般情况下，设置为2的n次方，以提高计算效率

## 13.9 激活函数的选择

激活函数

sigmoid

tanh

ReLU

其他改进型

和梯度消失问题有关

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

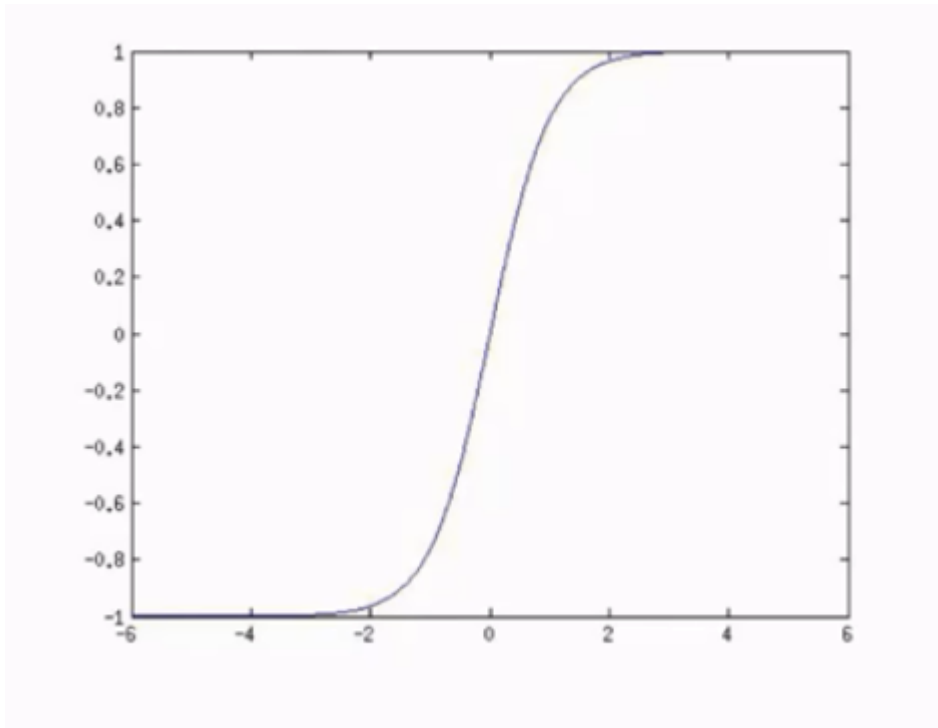
$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$\tanh'(x) = 1 - (\tanh(x))^2$$

$$\text{ReLU}(x) = \max(0, x)$$

sigmoid函数如下图





## 为什么需要激活函数？

$$h(x) = f\left(W^{(3)} f\left(W^{(2)} f\left(W^{(1)} x + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right)$$

$$W^{(3)} \left( W^{(2)} \left( W^{(1)} x + b^{(1)} \right) + b^{(2)} \right) + b^{(3)}$$

逐层

## 什么样的函数可以用作激活函数？

- 非线性
- 单调
- 连续函数
- 几乎处处可导
  - 因为神经网络中，我们要用梯度下降法来计算

如果我们学过计算机组成原理我们就会知道

饱和性-梯度消失问题

后面讲：什么样的函数是一个好的激活函数？

## 13.10 损失函数的选择

损失函数  
欧氏距离  
交叉熵  
对比损失

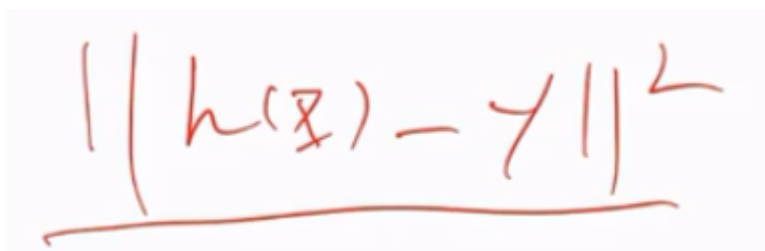
.....

对于分类问题为什么一般采用交叉熵而不是欧氏距离？

Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison.

- 损失函数

。


$$\|h(x) - y\|^2$$

- 欧氏距离
- 交叉熵
- 对比损失

对于分类问题为什么一般采用交叉熵而不是欧氏距离？

Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison.

实践证明，交叉熵的分类效果比欧式距离效果好。

所以以后我选交叉熵。

## 13.11 权重的初始化

用随机数进行初始化，均匀分布，正态分布。

caffe

常量初始化把权值或者偏置初始化为一个常数，这常数由用户自己定义  
 均匀分布初始化用均匀分布的随机数来初始化网络参数，均匀分布的上限和下限值由用户自己定义  
 高斯分布初始化用正态分布随机数初始化网络参数。正态分布的均值和方差由用户自己定义  
 positive\_unitball初始化也是用均匀分布的随机数初始化，但保证每个神经元与前一层所有神经元的连接的权重值之和为1  
 Xavier初始化采用了文献[1]中提出的方法，也是用均匀分布的随机数来初始化权重  
 MSRA初始化采用了文献[2]提出的方法，使用的是正态分布的随机数  
 Bilinear使用的双线程插值公式来初始化权重

Size

$W \rightarrow$

## 13.12 正则化

神经网络同样面临过拟合的问题。为了减轻过拟合，所以也需要正则化。

$$L(W) = \frac{1}{2m} \sum_{i=1}^m \|y_i - h(x_i)\|^2 + \lambda \frac{1}{2} \|W\|_2^2 \quad \lambda W_i$$

$$L(W) = \frac{1}{2m} \sum_{i=1}^m \|y_i - h(x_i)\|^2 + \lambda \|W\|_1 \quad \lambda \operatorname{sgn}(W_i)$$

- L2 正则化

- 对w进行求导，得到右边的式子

$$\lambda \frac{1}{2} \|W\|_2^2 \quad \lambda W_i$$

- L1

- $\lambda \|W\|_1 \quad \lambda \operatorname{sgn}(W_i)$

## 13.13 学习率的设定

为什么需要设置学习率？  
 一般设置为接近于0的正数  
 可以采用更复杂的策略，训练过程中动态调整

梯度下降法

$$W_{t+1} = W_t - \frac{2 \nabla_w L(W_t)}{\delta}$$

- 一般设置为接近于0的正数
- 可以采用更复杂的策略，训练过程中动态调整

## 13.14 动量项

为了加快算法的收敛速度减少震荡，引入了动量项  
动量项累积了之前的权重更新值

$$W_{t+1} = W_t + V_{t+1}$$

$$V_{t+1} = -\alpha \nabla_w L(W_t) + \mu V_t$$

正则化项与动量项的区别

$$W_{t+1} = W_t - 2 \nabla_w L(W_t)$$

$$V_{t+1} = -\alpha \nabla_w L(W_t) + \mu V_t$$

正则化项与动量项的区别

0.9

正则化项与动量项的区别

## 13.15 挑战与改进措施

### 挑战与改进措施

1. 梯度消失问题
2. 退化
3. 局部极小值
4. 鞍点问题

后面两个问题是最优化方法的问题。因为人工神经网络不是凸优化问题，所以不能产生全局最优解。

## 13.16 梯度消失问题

### 梯度消失问题

如果激活函数导数的绝对值小于1，多次连乘之后，误差项接近于0，导致根据它计算的参数梯度值接近于0，参数无法有效更新

$$\delta^{(l)} = \left(W^{(l+1)}\right)^T \delta^{(l+1)} \odot f'(u^{(l)})$$

与之相反的是梯度爆炸问题

ReLU一定程度上可以缓解梯度消失问题

激活函数的饱和性

$$\lim_{x \rightarrow -\infty} f'(x) = 0 \quad \lim_{x \rightarrow \infty} f'(x) = 0$$

$$x > c \quad f'(x) = 0 \quad x < c \quad f'(x) = 0$$

## 梯度消失问题

我可以理解为多次连乘之后，收敛于0.导致参数无法有效更新。

## 梯度爆炸问题

这问题不好解决



ReLU 一定程度上可以缓解梯度消失的问题

激活函数的饱和性

$$\lim_{x \rightarrow -\infty} f'(x) = 0 \quad \lim_{x \rightarrow \infty} f'(x) = 0$$
$$x > c \quad f'(x) = 0 \quad x < c \quad f'(x) = 0$$

## 13.17 退化

神经网络的训练误差和测试误差会随着层数的增加而增大

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. computer vision and pattern recognition. 2015.

## 13.18 局部极小值

局部极小值  
神经网络的优化目标不是一个凸优化  
有陷入局部极小值的风险

这问题，到目前为止也没很好的解决办法。

- [1] Sontag E. D, Sussman, H. J. Backpropagation can give rise to spurious local minima even for networks without hidden layers. Complex Systems, 3, 91-106, 1989.
- [2] Brady, M. I., Raghavan, R., Slawny, J. Backpropagation fails to separate where perceptrons succeed. IEEE Transactions on Circuits and Systems, 36(5), 665-674, 1989.
- [3] Gori, M, Tesi, A. On the problem of local minima in backpropagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(1), 76-86. 1992.

## 13.19 鞍点问题

鞍点虽然是个驻点，但是

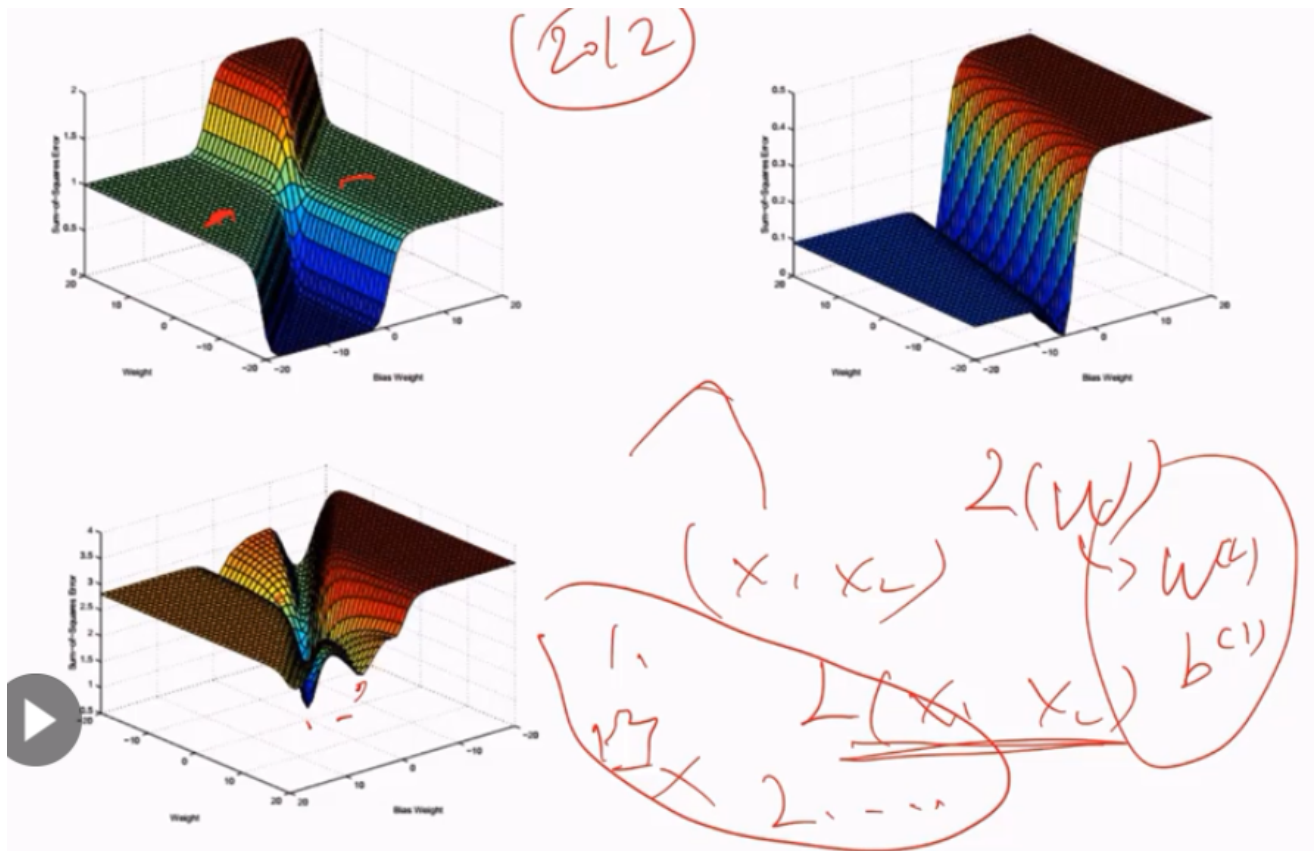


Hessian矩阵不定，不是局部极值点

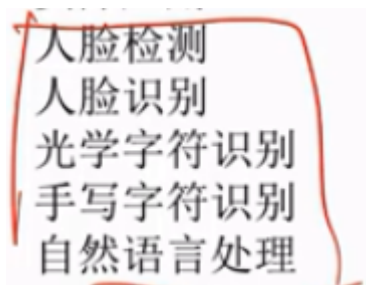
## 13.20 损失函数曲面分析

前面三篇文章

- [1] D R Hush, J M Slas, B Horne. Error surfaces for multi-layer perceptrons. international symposium on neural networks. 1991.
- [2] Choromanska A, Henaff M, Mathieu M, Ben Arous G, Le Cun Y. The loss surfaces of multilayer networks. arXiv:1412.0233. 2014.
- [3] Marcus Gallagher. Multi-layer Perceptron Error Surfaces: Visualization, Structure and Modelling. 2000.
- [4] Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., Bengio, Y. 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. NIPS 2014.



## 13.21 实际应用



这些论文有时间去找来看看

- [1] Henry A Rowley, Shumeet Baluja, Takeo Kanade. Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998.
- [2] Steve Lawrence, C L Giles, Ah Chung Tsoi, Andrew D Back. Face recognition: a convolutional neural-network approach. IEEE Transactions on Neural Networks. 1997.
- [3] Michael Sabourin, Amar Mitiche. Original Contribution: Optical character recognition by a neural network. Neural Networks. 1992.
- [4] M D Ganis, Charles L Wilson, James L Blue. Neural network-based systems for handprint OCR applications. IEEE Transactions on Image Processing. 1998.
- [5] Dong Xiao Ni. Application of Neural Networks to Character Recognition. 2007.
- [6] Neiva Maria Picinini Santos, E Da Costa Oliverira. A neural network for handwritten pattern recognition. 1996.
- [7] Anita Pal, Dayashankar Singh. Handwritten English Character Recognition Using Neural Network. 2010.
- [8] Mikaela Keller, Samy Bengio. A neural network for text representation. international conference on artificial neural networks. 2005.

## 13.22 实战项目

训练自己的神经网络，完成MNIST数据集的实验  
<http://yann.lecun.com/exdb/mnist/>  
尝试不同的层数，神经元数量  
尝试不同的激活函数  
尝试不同的训练参数

使用全连接神经网络

[yann.lecun.com/exdb/mnist](http://yann.lecun.com/exdb/mnist)





## 13.23 本集总结

### 13 神经网络3

#### 13.1 本集内容简介

#### 13.2 实验环节

##### 实验环节

##### 第一组实验

设置网络的层数和各层神经元数量

设置激活函数和参数

设置训练参数

生成训练样本

调用训练函数

##### 第二组实验

数据集A

数据集B

#### 13.3 理论分析

#### 13.4 理论分析-拟合能力

#### 13.5 理论分析- 与神经系统的关系

#### 13.6 实现细节问题

#### 13.7 输入值和输出值设定

#### 13.8 网络的规模

#### 13.9 激活函数的选择

为什么需要激活函数？

什么样的函数可以用作激活函数？

#### 13.10 损失函数的选择

#### 13.11 权重的初始化

caffe

#### 13.12 正则化

#### 13.13 学习率的设定

#### 13.14 动量项

#### 13.15 挑战与改进措施

#### 13.16 梯度消失问题

梯度消失问题

梯度爆炸问题

13.17 退化

13.18 局部极小值

13.19 鞍点问题

13.20 损失函数曲面分析

13.21 实际应用

13.22 实战项目

13.23 本集总结

1.