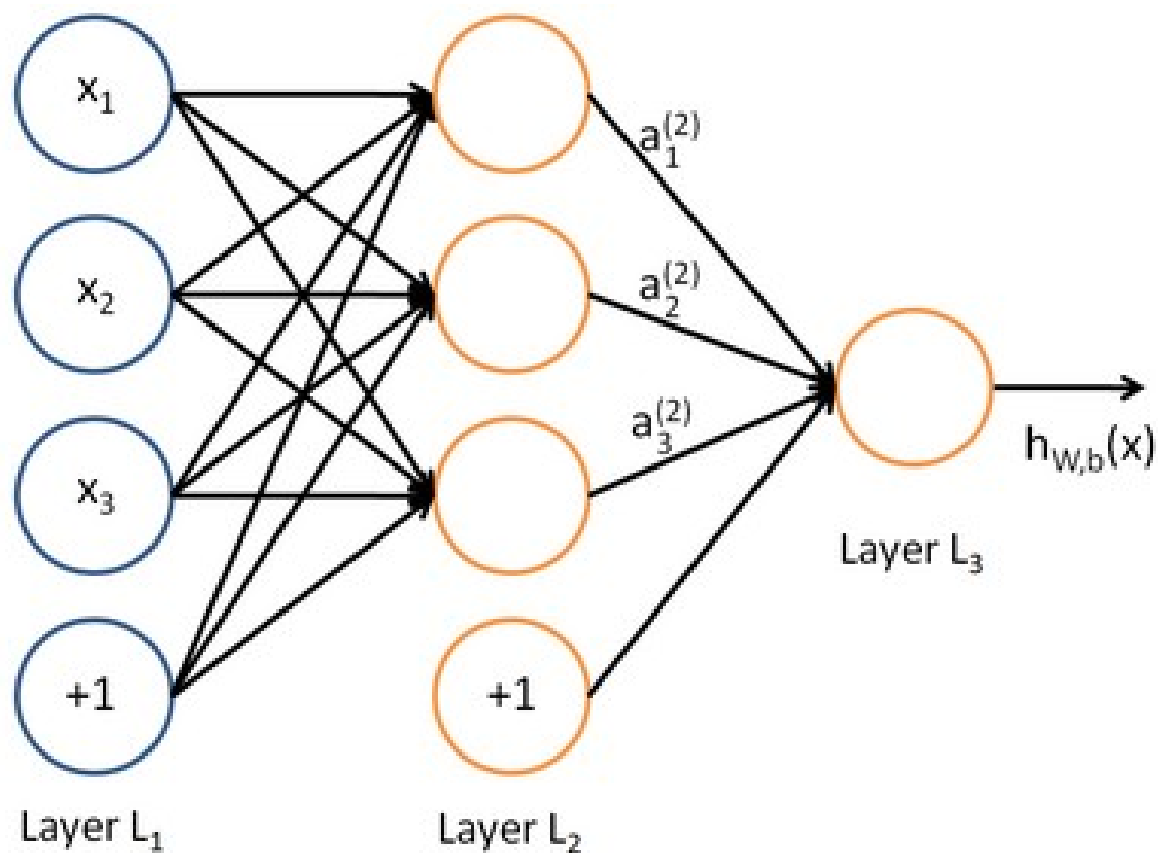# 全连接神经网络及实现

By 12330238
马毅辉

神经网络简介

内容

- 1. 数据预处理
- 2. 参数设置
- 3. 网络结构
- 4. 矩阵实现
- 5. 文件存取
- 6. 实验结果
- 7. 错误解析

数据预处理

1. 原数据 0-255 ，为什么要归一化？
    避免 sigmoid 函数没有变化。
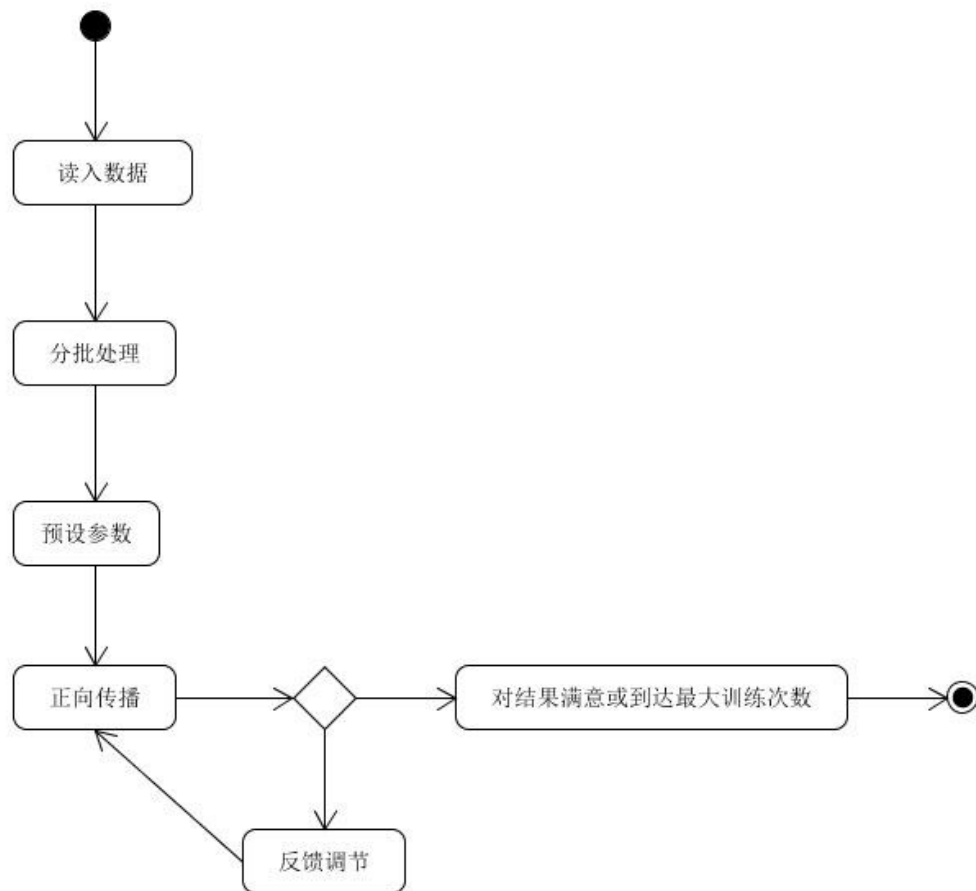 $f(z) = 1/(1+e-z)$
2. 得到感兴趣区域 (ROI)
    减少输入节点个数

参数设置

- 主要的问题：
    权值不能初始化过大，最好是 0.01
之后更小的数量级，过大会导致 sigmoid 函
数输出接近 0.5.

学习率： 0.05
权值初始： 0.01-0.09
Bias 初始： 0

网络结构（流程图）

读入数据

分批处理

预设参数

正向传播 → ◇ → 对结果满意或到达最大训练次数 → ●

反馈调节

网络结构（类图）

net
-firstnerous :int -col   :int -h_hide   :int -rate   :int -e_hide<vector> :Layer
+Net() +Net(filename:string,labelname:string, tfilename:string,tlabelname:string, hide:int,col,int,firstnerous:int) +ReadDate() :bool +ReadTest() :bool +Compute(i:int) :int +Back(i:int)   :void +Check(t:int) :void +CTest():void +Train(i;int) :void +Save(i:int) :bool +date() :void +Load(t:int) :void;

Layer
-numofbefore :int -numofsize   :int -output   :matrix -W   :matrix -bias   :matrix -residual   :matrix
+Layer() +Layer(nofbefore:int,nofsize); +SetStart():void +SetTStart();void +GetSize():int +GetOutPut():matrix +GetBias():matrix +GetW():matrix +Getresidual:matrix(); +Compute(date:matrix):void +UpdateWofend(label:matrix,frontout:matrix):void +UpdateW(residual:matrix,frontout:matrix):void +Load(file: ifstream&):void +Save(file: ofstream&):void

矩阵实现 (Armadillo)

安装部署：
与 opencv 的设置一样，包含目录，包含库．

能用到的函数和符号：见 armadillo.txt

容易出错的地方：
* 为乘法符号， % 为点乘符号。

文件存取

- 1. 矩阵的存取
      M.save(filename)   保存 matrix 为二进制文件
      M.load(filename)   读取 filename 内容为 matrix

- 2.Layer 的保存
      两个 int 参数，四个矩阵，矩阵的名字为 m.n 。
      四个矩阵的名字

实验结果对比 (1 循环）

- 361*100*10

```
You can load the last save network,if you input 'y'm
coast time:12
0.0946322 True:0.909383
times: 1
test true: 0.9483
请按任意键继续. . .
```

- 361*100*70*10

```
You can load the last save network,if you input 'y'n
coast time:13
0.12685 True:0.678083
times: 1
test true: 0.8937
请按任意键继续. . .
```

实验结果对比 (20 循环)

实验结果对比 (100 循环)

```
You can load the last save network,if you input 'y'y
szie:0*361
szie:361*100
szie:100*10
szie:0*361
szie:361*100
szie:100*10
test true: 0.9959
请按任意键继续. . .
```

```
coast time:26
0.00649698 True:0.994483
times: 95
coast time:27
0.00640577 True:0.994217
times: 96
coast time:27
0.00690133 True:0.993517
times: 97
coast time:27
0.00653059 True:0.994133
times: 98
coast time:27
0.00605201 True:0.99455
times: 99
coast time:27
0.00592533 True:0.995183
times: 100
szie:0*361
szie:361*100
szie:100*70
szie:70*10
save: 100
test true: 0.9966
请按任意键继续. . .
```

实验结果对比 （784 与 361 ）


```
coast time:61
0.0128803 True:0.991117
times: 15
coast time:60
0.0117412 True:0.991867
times: 16
coast time:61
0.0114335 True:0.992417
times: 17
coast time:60
0.0109795 True:0.99255
times: 18
coast time:61
0.00996118 True:0.992917
times: 19
coast time:60
0.00995929 True:0.993483
times: 20
szie:0*361
szie:361*300
szie:300*204
szie:204*10
save: 20
test true: 0.9937
请按任意键继续. . .
```

```
coast time:99
0.0103398 True:0.993183
times: 15
coast time:99
0.00966451 True:0.993433
times: 16
coast time:99
0.00867447 True:0.99445
times: 17
coast time:99
0.00878781 True:0.99415
times: 18
coast time:99
0.00834669 True:0.994667
times: 19
coast time:99
0.00801041 True:0.994767
times: 20
szie:0*784
szie:784*300
szie:300*204
szie:204*10
save: 20
test true: 0.9946
请按任意键继续. . .
```

实验中最好的结果


```
You can load the last save network,if you input 'y'y
szie:0*784
szie:784*300
szie:300*204
szie:204*10
szie:0*784
szie:784*300
szie:300*204
szie:204*10
test true: 0.9988
请按任意键继续. . .
```

- 学习率： 0.5
- W 权值： x>=0 && x<=0.01 的随机数
- 一条数据为一批
- Bias ： x>=0 && x<=0.01 的随机数
- 网络大小： 784*300*204*10

错误解析

- 1.W 赋值过大
- 2. 反馈函数写错
- 3. 矩阵操作 +- 是矩阵对应矩阵操作， */ 可以使用实数