

原

三分钟带你了解 Softmax 划重点

2018年06月14日 09:18:28

红色石头Will

阅读数：12402

标签：

机器学习

Softmax

更多

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/red_stone1/article/details/80687921

个人网站：[红色石头的机器学习之路](#)
CSDN博客：[红色石头的专栏](#)
知乎：[红色石头](#)
微博：[RedstoneWill的微博](#)
GitHub：[RedstoneWill的GitHub](#)
微信公众号：[AI有道](#)（ID：redstonewill）



1. 什么是Softmax

Softmax 在机器学习和深度学习中有非常广泛的应用。尤其在处理多分类（ $C > 2$ ）问题，分类器最后的输出单元需要Softmax 函数进行数值处理。函数的定义如下所示：

$$S_i = \frac{e^{V_i}}{\sum_i^C e^{V_i}}$$

其中， V_i 是分类器前级输出单元的输出。 i 表示类别索引，总的类别个数为 C 。 S_i 表示的是当前元素的指数与所有元素指数和的比值。Softmax 将多分转化为相对概率，更容易理解和比较。我们来看下面这个例子。

一个多分类问题， $C = 4$ 。线性分类器模型最后输出层包含了四个输出值，分别是：

$$V = \begin{bmatrix} -3 \\ 2 \\ -1 \\ 0 \end{bmatrix}$$

经过Softmax处理后，数值转化为相对概率：

$$S = \begin{bmatrix} 0.0057 \\ 0.8390 \\ 0.0418 \\ 0.1135 \end{bmatrix}$$

很明显，Softmax 的输出表征了不同类别之间的相对概率。我们可以清晰地看出， $S_1 = 0.8390$ ，对应的概率最大，则更清晰地可以判断预测为第1类的。Softmax 将连续数值转化成相对概率，更有利于我们理解。

实际应用中，使用 Softmax 需要注意数值溢出的问题。因为有指数运算，如果 V 数值很大，经过指数运算后的数值往往可能有溢出的可能。所以，需些数值处理：即 V 中的每个元素减去 V 中的最大值。

$$D = \max(V)$$
$$S_i = \frac{e^{V_i - D}}{\sum_i^C e^{V_i - D}}$$

相应的python示例代码如下：

```
1 scores = np.array([123, 456, 789]) # example with 3 classes and each having large scores
2 scores -= np.max(scores) # scores becomes [-666, -333, 0]
3 p = np.exp(scores) / np.sum(np.exp(scores))
```

2. Softmax 损失函数

我们知道，线性分类器的输出是输入 x 与权重系数的矩阵相乘： $s = Wx$ 。对于多分类问题，使用 Softmax 对线性输出进行处理。这一小节我们来探讨损失函数。

$$S_i = \frac{e^{S_{ik}}}{\sum_{j=1}^C e^{S_j}}$$

由于 log 运算符不会影响函数的单调性，我们对 S_i 进行 log 操作：

$$S_i = \log \frac{e^{S_{y_i}}}{\sum_{j=1}^C e^{S_j}}$$

我们希望 S_i 越大越好，即正确类别对应的相对概率越大越好，那么就可以对 S_i 前面加个负号，来表示损失函数：

$$L_i = -S_i = -\log \frac{e^{S_{y_i}}}{\sum_{j=1}^C e^{S_j}}$$

对上式进一步处理，把指数约去：

$$L_i = -\log \frac{e^{S_{y_i}}}{\sum_{j=1}^C e^{S_j}} = -(s_{y_i} - \log \sum_{j=1}^C e^{s_j}) = -s_{y_i} + \log \sum_{j=1}^C e^{s_j}$$

这样，Softmax 的损失函数就转换成了简单的形式。

举个简单的例子，上一小节中得到的线性输出为：

$$V = \begin{bmatrix} -3 \\ 2 \\ -1 \\ 0 \end{bmatrix}$$

假设 $i = 1$ 为真实样本，计算其损失函数为：

$$L_i = -2 + \log(e^{-3} + e^2 + e^{-1} + e^0) = 0.1755$$

$$L_i = 3 + \log(e^{-3} + e^2 + e^{-1} + e^0) = 5.1755$$

3. Softmax 反向梯度

推导了 Softmax 的损失函数之后，接下来继续对权重参数进行反向求导。

Softmax 线性分类器中，线性输出为：

$$S_i = Wx_i$$

其中，下标 i 表示第 i 个样本。

求导过程的程序设计分为两种方法：一种是使用嵌套 for 循环，另一种是直接使用矩阵运算。

使用嵌套 for 循环，对权重 W 求导函数定义如下：

```
1 def softmax_loss_naive(W, X, y, reg):
2     """
3     Softmax loss function, naive implementation (with loops)
4
5     Inputs have dimension D, there are C classes, and we operate on minibatches
6     of N examples.
7
8     Inputs:
9     - W: A numpy array of shape (D, C) containing weights.
10    - X: A numpy array of shape (N, D) containing a minibatch of data.
11    - y: A numpy array of shape (N,) containing training labels; y[i] = c means
12        that X[i] has label c, where 0 <= c < C.
13    - reg: (float) regularization strength
14
15    Returns a tuple of:
16    - loss as single float
17    - gradient with respect to weights W; an array of same shape as W
18    """
19    # Initialize the loss and gradient to zero.
20    loss = 0.0
21    dW = np.zeros_like(W)
22
23    num_train = X.shape[0]
24    num_classes = W.shape[1]
25    for i in xrange(num_train):
26        # Compute the loss and gradient for this example
```

```

29     loss += -scores_shift[right_class] + np.log(np.sum(np.exp(scores_shift)))
30     for j in xrange(num_classes):
31         softmax_output = np.exp(scores_shift[j]) / np.sum(np.exp(scores_shift))
32         if j == y[i]:
33             dW[:,j] += (-1 + softmax_output) * X[i,:]
34         else:
35             dW[:,j] += softmax_output * X[i,:]
36
37     loss /= num_train
38     loss += 0.5 * reg * np.sum(W * W)
39     dW /= num_train
40     dW += reg * W
41
42     return loss, dW

```

5

2

使用矩阵运算，对权重 W 求导函数定义如下：

```

1 def softmax_loss_vectorized(W, X, y, reg):
2     """
3     Softmax loss function, vectorized version.
4
5     Inputs and outputs are the same as softmax_loss_naive.
6     """
7     # Initialize the loss and gradient to zero.
8     loss = 0.0
9     dW = np.zeros_like(W)
10
11     num_train = X.shape[0]
12     num_classes = W.shape[1]
13     scores = X.dot(W)
14     scores_shift = scores - np.max(scores, axis = 1).reshape(-1,1)
15     softmax_output = np.exp(scores_shift) / np.sum(np.exp(scores_shift), axis=1).reshape(-1,1)
16     loss = -np.sum(np.log(softmax_output[range(num_train), list(y)]))
17     loss /= num_train
18     loss += 0.5 * reg * np.sum(W * W)
19
20     dS = softmax_output.copy()
21     dS[range(num_train), list(y)] += -1
22     dW = (X.T).dot(dS)
23     dW = dW / num_train + reg * W
24
25     return loss, dW

```

实际验证表明，矩阵运算速度要比嵌套循环快很多，特别是在训练样本数量多的情况下。我们使用 CIFAR-10 数据集中约5000个样本对两种求导方

比：

```

1 tic = time.time()
2 loss_naive, grad_naive = softmax_loss_naive(W, X_train, y_train, 0.000005)
3 toc = time.time()
4 print('naive loss: %e computed in %fs' % (loss_naive, toc - tic))
5
6 tic = time.time()
7 loss_vectorized, grad_vectorized = softmax_loss_vectorized(W, X_train, y_train, 0.000005)
8 toc = time.time()
9 print('vectorized loss: %e computed in %fs' % (loss_vectorized, toc - tic))
10
11 grad_difference = np.linalg.norm(grad_naive - grad_vectorized, ord='fro')
12 print('Loss difference: %f' % np.abs(loss_naive - loss_vectorized))
13 print('Gradient difference: %f' % grad_difference)

```

结果显示为：

```

naive loss: 2.362135e+00 computed in 14.680000s
vectorized loss: 2.362135e+00 computed in 0.242000s
Loss difference: 0.000000
Gradient difference: 0.000000

```

显然，此例中矩阵运算的速度要比嵌套循环快60倍。所以，当我们在编写机器学习算法模型时，尽量使用矩阵运算，少用 嵌套循环，以提高运算速度。

4. Softmax 与 SVM

Softmax线性分类器的损失函数计算相对概率，又称交叉熵损失「Cross Entropy Loss」。线性 SVM 分类器和 Softmax 线性分类器的主要区别在于损失函数。SVM 使用 hinge loss，更关注分类正确样本和错误样本之间的距离「 $\Delta = 1$ 」，只要距离大于 Δ ，就不在乎到底距离相差多少。而 Softmax 的损失函数都会影响其损失函数的大小。举个例子来说明，类别个数 $C = 3$ ，两个样本的得分函数分别为 $[10, -10, -10]$ ， $[10, -20, -20]$ ，真实标签为第0类。说，这两个 L_i 都为0；但对于Softmax来说，这两个 L_i 分别为0.00和0.55，差别很大。

关于 SVM 线性分类器，我在上篇文章里有所介绍，传送门：

[基于线性SVM的CIFAR-10图像集分类](#)

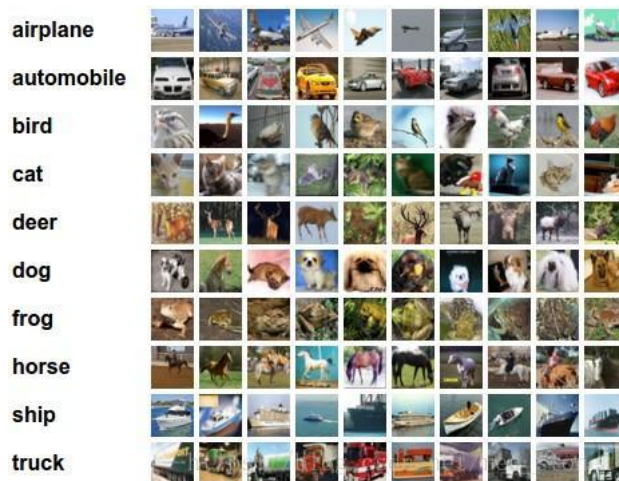
接下来，谈一下正则化参数 λ 对 Softmax 的影响。我们知道正则化的目的是限制权重参数 W 的大小，防止过拟合。正则化参数 λ 越大，对 W 的限制越大，某3分类的线性输出为 $[1, -2, 0]$ ，相应的 Softmax 输出为 $[0.7, 0.04, 0.26]$ 。假设，正类类别是第0类，显然，0.7远大于0.04和0.26。

若使用正则化参数 λ ，由于限制了 W 的大小，得到的线性输出也会等比例缩小： $[0.5, -1, 0]$ ，相应的 Softmax 输出为 $[0.55, 0.25, 0.33]$ 。显然，正确样本之间的相对概率差距变小了。

也就是说，正则化参数 λ 越大，Softmax 各类别输出越接近。大的 λ 实际上是「均匀化」正确样本与错误样本之间的相对概率。但是，概率大小的相改变，这点需要留意。因此，也不会影响到对 Loss 的优化算法。

5. Softmax 实际应用

使用 Softmax 线性分类器，对 CIFAR-10 图片集进行分类。



使用交叉验证，选择最佳的学习因子和正则化参数：

```
1 # Use the validation set to tune hyperparameters (regularization strength and
2 # learning rate). You should experiment with different ranges for the learning
3 # rates and regularization strengths; if you are careful you should be able to
4 # get a classification accuracy of over 0.35 on the validation set.
5 results = {}
6 best_val = -1
7 best_softmax = None
8 learning_rates = [1.4e-7, 1.5e-7, 1.6e-7]
9 regularization_strengths = [8000.0, 9000.0, 10000.0, 11000.0, 18000.0, 19000.0, 20000.0, 21000.0]
10
11 for lr in learning_rates:
12     for reg in regularization_strengths:
13         softmax = Softmax()
14         loss = softmax.train(X_train, y_train, learning_rate=lr, reg=reg, num_iters=3000)
15         y_train_pred = softmax.predict(X_train)
16         training_accuracy = np.mean(y_train == y_train_pred)
17         y_val_pred = softmax.predict(X_val)
18         val_accuracy = np.mean(y_val == y_val_pred)
19         if val_accuracy > best_val:
```

```
22         results[(lr, reg)] = training_accuracy, val_accuracy
23
24     # Print out results.
25     for lr, reg in sorted(results):
26         train_accuracy, val_accuracy = results[(lr, reg)]
27         print('lr %e reg %e train accuracy: %f val accuracy: %f' % (
28             lr, reg, train_accuracy, val_accuracy))
29
30     print('best validation accuracy achieved during cross-validation: %f' % best_val)
```

5
2

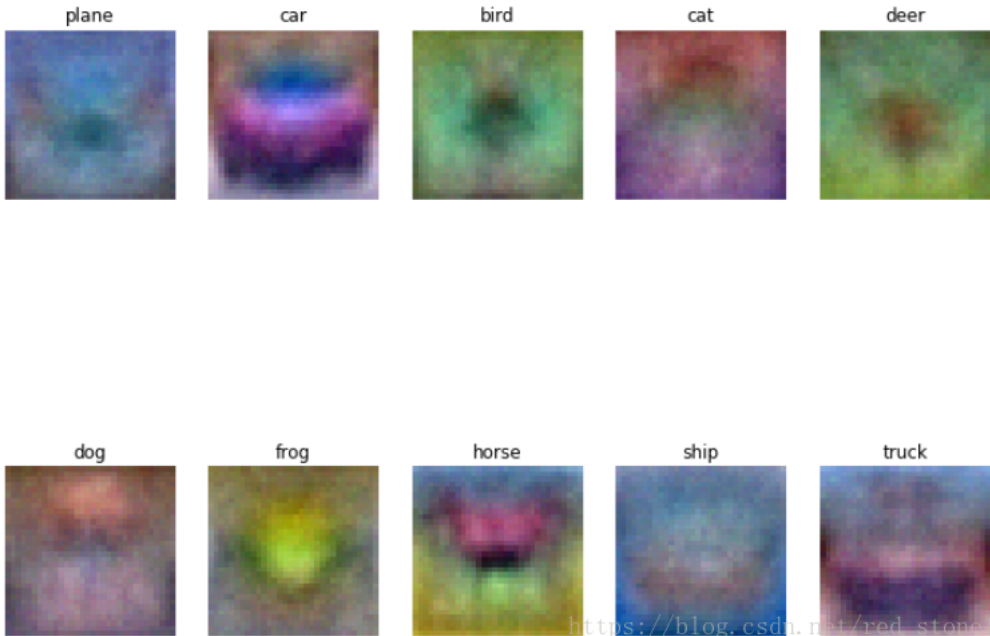
训练结束后，在测试图片集上进行验证：

```
1 # evaluate on test set
2 # Evaluate the best softmax on test set
3 y_test_pred = best_softmax.predict(X_test)
4 test_accuracy = np.mean(y_test == y_test_pred)
5 print('softmax on raw pixels final test set accuracy: %f' % (test_accuracy, ))
```

softmax on raw pixels final test set accuracy: 0.386000

权重参数 W 可视化代码如下：

```
1 # Visualize the learned weights for each class
2 w = best_softmax.W[:-1,:] # strip out the bias
3 w = w.reshape(32, 32, 3, 10)
4
5 w_min, w_max = np.min(w), np.max(w)
6
7 classes = ['plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']
8 for i in range(10):
9     plt.subplot(2, 5, i + 1)
10
11     # Rescale the weights to be between 0 and 255
12     wimg = 255.0 * (w[:, :, :, i].squeeze() - w_min) / (w_max - w_min)
13     plt.imshow(wimg.astype('uint8'))
14     plt.axis('off')
15     plt.title(classes[i])
```



很明显，经过训练学习，W 包含了相应类别的某些简单色调和轮廓特征。

源码



「AI有道」

ID: redstonewill

机器学习、深度学习干货汇总
注重理论与实战相结合
为你提供切实可行的AI学习路线。

5

2

参考文献：

<http://cs231n.github.io/linear-classify/>

想对作者说点什么？

我来说两句

qq_38824451：你这篇文章三分钟能看完？

(06-21 08:23 #1楼)

[查看回复\(1\)](#)

Softmax的通俗讲解

697

0 前言Softmax在机器学习中应用非常广泛，尤其在处理多分类问题，分类器最后的输出单元需要Softmax 函数进行数值处理。但是刚刚接触机器学习的...

手打例子一步一步带你看懂softmax函数以及相关求导过程

351

这几天学习了一下softmax激活函数，以及它的梯度求导过程，整理一下便于分享和交流！softmax函数softmax用于多分类过程中，它将多个神经元的输...

各种激活函数比较

103

除了下述三种激活函数，还有softmax、elu、softplus、softsign、relu、tanh、sigmoid、hard_sigmoid、linear1、sigmod函数 y值在【0，1】...

sigmoid和softmax总结

9.1万

sigmoid函数（也叫逻辑斯谛函数）： 引用wiki百科的定义： A logistic function or logistic curve is a common “S” shape (si...

logistic回归和softmax回归

2442

logistic回归 在 logistic 回归中，我们的训练集由 个已标记的样本构成：。由于 logistic 回归是针对二分类问题的，因此类标记。假设函数(hypothesis fun...

Softmax回归问题分析

157

在本节中，我们介绍Softmax回归模型，该模型是logistic回归模型在多分类问题上的推广，在多分类问题中，类标签 可以取两个以上的值。Softmax回归...

ATPG中经常用到的名词SPF

2.5万

ATPG中经常用到的名词SPFSTIL procedure file (SPF)TetraMAX ATPG uses STIL in several different contexts. Desig...

平均值(Mean)、方差(Variance)、标准差(Standard Deviation)

2.1万

本文目录 本文目录 开篇明志 平均值 方差标准差 为什么使用标准差 贝赛尔修正 公式的选择 平均值与标准差的适用范围及误用开篇明志对于一维数据的分...

C++ - Vector 计算 均值(mean) 和 方差(variance)

2.7万

Vector 计算 均值(mean) 和 方差(variance) 本文地址: http://blog.csdn.net/caroline_wendy vector的数组, 计算均值和方差的最...

量子纠缠可能是真实存在

189

量子纠缠可能是真实存在 红朝儒生 2018-4-5 关键字：量子纠缠 真实 简介：从测不准原理来看，量子纠缠必须存在。 量子纠缠的大意...

文章热词

机器学习算法框架简介 机器学习中的优化器 ai 跟机器学习 ng 机器学习笔记 机器学习资料数学

相关热词

demo三分钟 三分钟命令 三分钟五步 三分钟了解 mysql三分钟

博主推荐

 AI之路
关注 144篇文章

 悟乙己
关注 295篇文章

 zhiyong_will
关注 171篇文章

CS231n课程学习笔记（三）——Softmax分类器的实现

翻译笔记：<https://zhuanlan.zhihu.com/p/21930884?refer=intelligentuni> 参考资料：http://blog.csdn.net/pjia_1008...

卷积神经网络系列之softmax，softmax loss和cross entropy的讲解

我们知道卷积神经网络（CNN）在图像领域的应用已经非常广泛了，一般一个CNN网络主要包含卷积层，池化层（pooling），全连接层，损失函数层...

softmax的多分类

关于多分类我们常见的逻辑回归、SVM等常用于解决二分类问题，对于多分类问题，比如识别手写数字，它就需要10个分类，同样也可以用逻辑回归或SVM...

斯坦福大学深度学习公开课cs231n学习笔记（1）softmax函数理解与应用

我学习使用的是带中文翻译字幕的网易课程，公开课地址：<http://study.163.com/course/courseLearn.htm?courseId=1003223001#/learn/vid...>

Softmax的理解与应用

Softmax 理解 应用

简单易懂的softmax交叉熵损失函数求导

来写一个softmax求导的推导过程，不仅可以给自己理清思路，还可以造福大众，岂不美哉~ softmax经常被添加在分类任务的神经网络中的输出层，神经网络...

softmax——logistic的泛化

logistic回归：训练集为{(x(1),y(1)),...,(x(m),y(m))}，其中m为样本数，x(i)为特征。logistic回归是针对二分类问题的，因此类标y(i)∈{0,1}...

神经网络四：Softmax以及与Sigmoid的关系

Softmax 是神经网络中另一种输出层函数，计算输出层的值。主要用于神经网络最后一层，作为输出层进行多分类，与Logistic多分类函数相对应。两者各有千秋...

softmax

对于神经元间的非线性变化可以使用softmax函数：可以看出对于某一层的神元的所有值的和为1，这样的话每一个神经元的结构也可以表示成概率。 ...

Coursera | Andrew Ng (02-week3-3.9)—训练一个 Softmax 分类器

该系列仅在原课程基础上部分知识点添加个人学习笔记，或相关推导补充等。如有错误，还请批评指教。在学习了 Andrew Ng 课程的基础上，为了更方便...

softmax到底怎么实现分类——知乎大神讲解

作者：忆臻 链接：<https://www.zhihu.com/question/23765351/answer/240869755> 来源：知乎 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

量子纠缠和人类意识

朱清时：谢谢慈雄先生的介绍。今天我很荣幸能够应邀在南老师九十六岁诞辰日来作这一次报告。南老师一生用了七十年来教化众生，被他教化的人无...

深度学习与卷积神经网络（直观理解）

好吧，读了男神哥哥们的博客，自己写不来更好的。附上链接：[凌风探梅的卷积神经网络（CNN）新手指南](http://blog.csdn.NET/real_myth/article/details/21930884) http://blog.csdn.NET/real_myth/article/details/21930884

深度学习（DL）：卷积神经网络（CNN）：从原理到实现

序深度学习现在大火，虽然自己上过深度学习课程、用过keras做过一些实验，始终觉得理解不透彻。最近仔细学习前辈和学者的著作，感谢他们的无私...

吴恩达deeplearning之CNN—卷积神经网络入门

1.边界检测示例假如你有一张如下的图像，你想让计算机搞清楚图像上有什么物体，你可以做的事情是检测图像的垂直边缘和水平边缘。如下是一个6*6的图像...

人脸识别系列（十六）：AMSoftmax	 3627
论文链接：Additive Margin Softmax for Face Verification AMSoftmax Softmax ASoftmax 这是在SphereFac...	
机器学习之Softmax回归模型	5 495
Softmax在机器学习中有非常广泛的应用，但是刚刚接触机器学习的人可能对Softmax的特点以及好处并不理解，其实你了解了以后就会发现，	2 x计...
卷积神经网络系列之softmax loss对输入的求导推导	4123
我们知道卷积神经网络（CNN）在图像领域的应用已经非常广泛了，一般一个CNN网络主要包含卷积层，池化层（pooling），全连接层，损失	虽...
softmax loss对输入的求导推导	92
转载自: https://blog.csdn.net/u014380165/article/details/79632950我们知道卷积神经网络（CNN）在图像领域的应用已经非常广泛了，一般	4N...
通俗详解softmax函数及其求导过程	820
通俗详解softmax函数及其求导过程 陆续的将以前的一些文章原创保护和重新排版一下！以及它的梯度求导过程，整理一下便于分享和交流！本文目录如...	
softmax代价函数的导数计算	 8037
对于softmax的理解请参考Ufld教程，本文仅对代价函数求导部分进行推导softmax regression 代价函数： $J(\theta) = -1/m \sum_{i=1}^m \sum_{j=1}^K 1\{y(i)=j\} \log \theta_{Tj}$	
Softmax代价函数求导过程	 1557
Cost Function参考UFLDL的Softmax回归，SoftMax回归的损失函数形为： $J(\theta) = -1/N \sum_{i=1}^N \sum_{k=1}^K 1\{y(i)=k\} \log \theta_{Tj} x(i) \sum_{k=1}^K 1e^{\theta_{Tj} x(i)}$	
Softmax	 1.1万
简介 其实吧，一般在神经网络中用的最多的是sigmoid和tanh，当然也有用relu的。这是针对“是”和“否”的分类，但当进行多分类时，就要用到softmax。...	
深度学习笔记8：softmax层的实现	 3.4万
如果有什么疑问或者发现什么错误，欢迎在评论区留言，有时间我会一一回复 softmax简介 Softmax回归模型是logistic回归模型在多分类问题上的推广，...	
Softmax与SoftmaxWithLoss原理及代码详解	 3362
一直对softmax的反向传播的caffe代码看不懂，最近在朱神的数学理论支撑下给我详解了它的数学公式，才豁然开朗 SoftmaxWithLoss的由来 SoftmaxWit...	
Softmax函数与交叉熵	 2.6万
Softmax函数与交叉熵	
深度学习基础（一）——softmax及logsoftmax	 1万
softmax：重新定义了多层神经网络的输出层（output layer），注意仅和输出层有关系，和其他层无关。softmax function，也称为 normalized expone...	
卷积神经网络的训练方法	 2018
卷积神经网络的训练方法	
Softmax的解释	 389
作者：忆臻 链接：https://www.zhihu.com/question/23765351/answer/240869755 来源：知乎 著作权归作者所有。商业转载请联系作者获得授权，非...	
深度学习常用激活函数之— Sigmoid & ReLU & Softmax	 4.4万
深度学习常用激活函数-ReLU	
【深度学习】卷积神经网络（CNN）	 3242
卷积神经网络（Convolutional Neural NetWork，CNN）：自然语言处理、医药发现、灾难气候发现、人工智能程序。输入层：整个神经网络的输入 ...	
卷积神经网络的训练	 4789
卷积神经网络的训练 和全连接神经网络相比，卷积神经网络的训练要复杂一些。但训练的原理是一样的：利用链式求导计算损失函数对每个权重的偏导...	
<div>下载</div> softmax分类器简介	05-14
对softmax分类器简单的介绍，适合初学者快速了解softmax分类器的基本原理。	

详解softmax函数以及相关求导过

这几天学习了一下softmax激活函数，以及它的梯度求导过程，整理一下便于分享和交流！ softmax函数 softmax用于多分类过程中，它将多个神经元的输出...

下载

Matlab实现softmax分类

用matlab实现softmax回归

Softmax 和 为什么要用偏置值

1. 激活函数Rectified Linear Unit(ReLU) - 用于隐层神经元输出Sigmoid - 用于隐层神经元输出Softmax - 用于多分类神经网络输出Linear - 用于回

softmax求导，你求对了吗

问题提出 按定义(一般资料中的定义)，softmax的似然函数为： $\prod_i \prod_k P_k(x_i) I(y(i)=k)$ 其中 $k \in \{1,2,\dots,K\}$, $i \in \{1,2,\dots,N\}$, $P_k(x_i)$ 按soft...

从头理解softmax，笔记

损失函数指数损失函数（Adaboost）平方差损失函数：yolo交叉熵损失函数：softmaxloss线性分类： $y=w_0x_0+w_1x_1+\dots+w_nx_n=w$ 转置x激活函数sigmoid函...

CNN 简史与 Keras 增量实现（一）—— Softmax 分类器

在学习 CNN 之前，我们先实现一个 Softmax 多类分类器，作为 baseline。实现 Softmax 过程中涉及的参数初始化、梯度下降和一些实验观察有助于后面...

机器学习->监督学习->logistic回归,softMax回归

本篇博文来总结一下回归模型里面两个非常重要的模型。 logistic回归 softmax回归 Logistic回归logistics回归虽然有“回归”两字但是却是分类模型，并且...

2934

07-15

122

1122

2046

红色石头Will

关注

向TA提问

博客专家

原创

91

粉丝

1500

喜欢

425

评论

336

等级： 博客 6

访问： 55万+

积分： 5198

排名： 7626

勋章：

欢迎关注『AI有道』公众号

A.I.

博主专栏

吴恩达深度学习专项课程

阅读量：8503118 篇

林轩田机器学习课程笔记

阅读量：14616634 篇

机器学习这件小事

登录

注册

×

https://blog.csdn.net/red_stone1/article/details/80687921

9/11

redstone1

redstone1

斯坦福cs231n课程剖析

阅读量：3564

4 篇

展开

热门文章

Jupyter notebook入门教程（上）

阅读量：100171

多元函数的泰勒(Taylor)展开式

阅读量：73872

Ubuntu16.04环境下PyTorch简易安装教程

阅读量：23436

Jupyter notebook入门教程（下）

阅读量：19401

matlab svmtrain和svmclassify函数使用示例

阅读量：17304

最新文章

通俗解释协方差与相关系数

如何在 Google 免费云端运行 Python 深度学习框架？

机器学习笔试题精选（六）

机器学习笔试题精选（五）

白话生成对抗网络 GAN，50 行代码玩转 GAN 模型！【附源码】

个人分类

机器学习70篇

台湾大学林轩田机器学习基石18篇

台湾大学林轩田机器学习技法17篇

深度学习28篇

吴恩达深度学习专项课程18篇

展开

归档

2018年9月2篇

2018年8月2篇

2018年7月9篇

2018年6月8篇

2018年5月4篇

展开

最新评论

如何在 Google 免费云端运行...

sjz_hahalala479：感谢！

基于线性SVM的CIFAR-10图...

weixin_41831956：请问上面的代码是OVA的方法吧？如果ovo 的话，应该怎么实现呢？

Coursera吴恩达《神经网络与...

ZengTudou：在11. Random Initialization点中，为什么当第一层的权重w为0时，第一层的a...

【通俗易懂】机器学习中的 L1 和

5
2

Zoey_7900 : 协方差看过不少资料, 这个最最清楚明白了. 真心赞啊 买了博主深度学习的课程, 正在死磕中

联系我们



扫码联系客服

关于 招聘 广告服务 网站地图
京ICP证09002463号
百度提供站内搜索
©2018 CSDN版权所有

✉kefu@csdn.net ☎400-660-0108

🗨QQ客服 🗨客服论坛

经营性网站备案信息 网络110报警服务

中国互联网举报中心 北京互联网违法和不良信息举报中心

违法和不良信息举报中心



5
2