矩阵求导

覃雄派

矩阵求导

提纲

- 矩阵求导入门
- 矩阵求导实例

矩阵求导

- 矩阵求导参考资料

The Matrix Cookbook

[http://matrixcookbook.com]

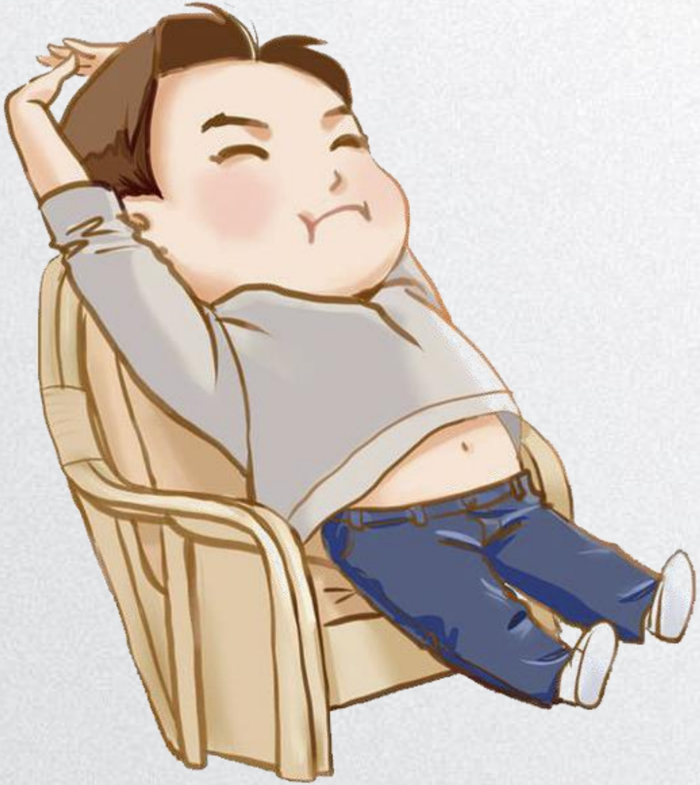
Kaare Brandt Petersen
Michael Syskind Pedersen

VERSION: NOVEMBER 15, 2012

这里先进行简单入门，然后直接使用该cook book的一些结论

https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

矩阵求导



矩阵求导

- 矩阵求导入门

按照“列”组织向量

$$\mathrm{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

$$\mathrm{x}^T\theta = \begin{bmatrix} 1 & x_1 & x_2 & \dots & x_d \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} = \theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$$

向量点乘

矩阵求导

- 矩阵求导入门

$$\mathrm{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

$$\mathrm{x}^T\theta = \begin{bmatrix} 1 & x_1 & x_2 & \dots & x_d \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} = \theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$$

1. $\theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$ 对θ_0、θ_1、θ_2、…、θ_d求导
2. 得到

$$1 \quad x_1 \quad x_2 \quad \dots \quad x_d$$

3. 按照列向量来组织为$\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

矩阵求导

- 矩阵求导入门

$$\mathrm{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

$$\mathrm{x}^T\theta = \begin{bmatrix} 1 & x_1 & x_2 & \dots & x_d \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} = \theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$$

1. $\theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$ 对θ_0、θ_1、θ_2、…、θ_d 求导

2.得到

$$1 \quad x_1 \quad x_2 \quad \dots \quad x_d$$

3.按照列向量来组织为 $\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

$$\frac{\partial \mathrm{x}^T\theta}{\partial \boldsymbol{\theta}} = \frac{\partial \theta^T \mathrm{x}}{\partial \boldsymbol{\theta}} = \mathrm{x}$$

第一个重要公式

矩阵求导

- 矩阵求导入门

$$\mathrm{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

$$\mathrm{x}^T\theta = \begin{bmatrix} 1 & x_1 & x_2 & \dots & x_d \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} = \theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$$

$\theta_0 + x_1\theta_1 + x_2\theta_2 + \dots + x_d\theta_d$ 对θ_0、θ_1、θ_2、…、θ_d 求导得到

$$1 \quad x_1 \quad x_2 \quad \dots \quad x_d$$

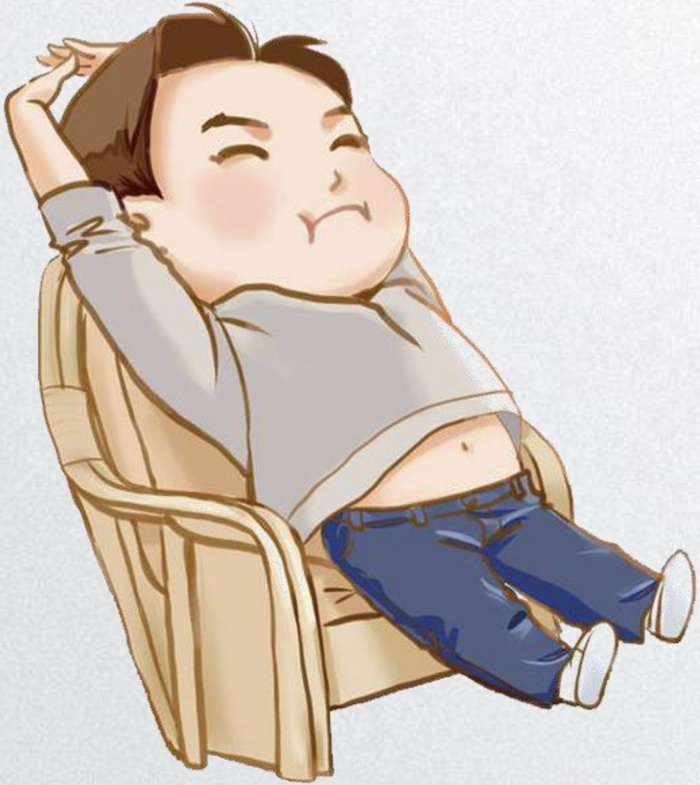
按照列向量来组织为 $\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

$\boldsymbol{x}$ 列向量

$\boldsymbol{a}$ 列向量

$$\frac{\partial \boldsymbol{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \boldsymbol{a}}{\partial \mathbf{x}} = \boldsymbol{a}$$

矩阵求导



矩阵求导

- 矩阵求导入门

$\boldsymbol{x}$ 列向量

$\boldsymbol{a}$ 列向量

$$\frac{\partial \boldsymbol{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \boldsymbol{a}}{\partial \mathbf{x}} = \boldsymbol{a}$$

$\mathbf{b}^T \mathbf{A}$ 看作一个整体

$$\frac{\partial \mathbf{b}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = \boldsymbol{A}^T \boldsymbol{b}$$

矩阵求导

- 矩阵求导入门

$\boldsymbol{x}$ 列向量

$\boldsymbol{a}$ 列向量

$$\frac{\partial \boldsymbol{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \boldsymbol{a}}{\partial \mathbf{x}} = \boldsymbol{a}$$

$$\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^T \boldsymbol{x}$$

有两个x，分别求导，累加

$\boxed{\mathbf{x}^T \mathbf{A}}\mathbf{x}$

$\mathbf{x}^T \boxed{\mathbf{A}\mathbf{x}}$

矩阵求导

- 矩阵求导入门

$\boldsymbol{x}$	列向量
$\boldsymbol{a}$	列向量

$$\frac{\partial \boldsymbol{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \boldsymbol{a}}{\partial \mathbf{x}} = \boldsymbol{a}$$

有两个x，分别求导，累加

$$\boxed{\mathbf{x}^T} \mathbf{A}\mathbf{x}$$

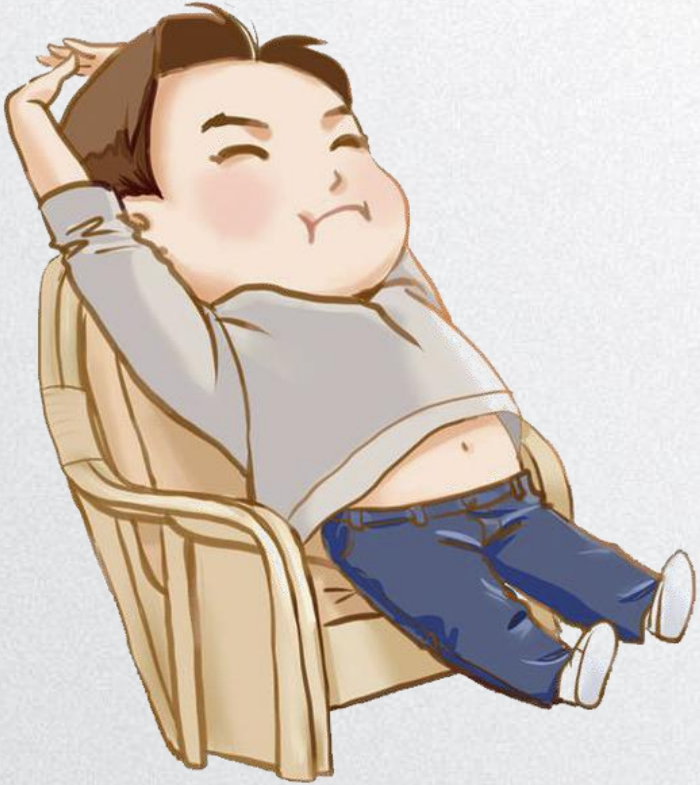
$$\mathbf{x}^T \boxed{\mathbf{A}\mathbf{x}}$$

$$\frac{\partial \mathbf{x}^T \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^T\boldsymbol{x}$$

如果A是对称矩阵

$$2\boldsymbol{A}\boldsymbol{x}$$

矩阵求导



矩阵求导

在SVM的梯度下降法求解过程中用到

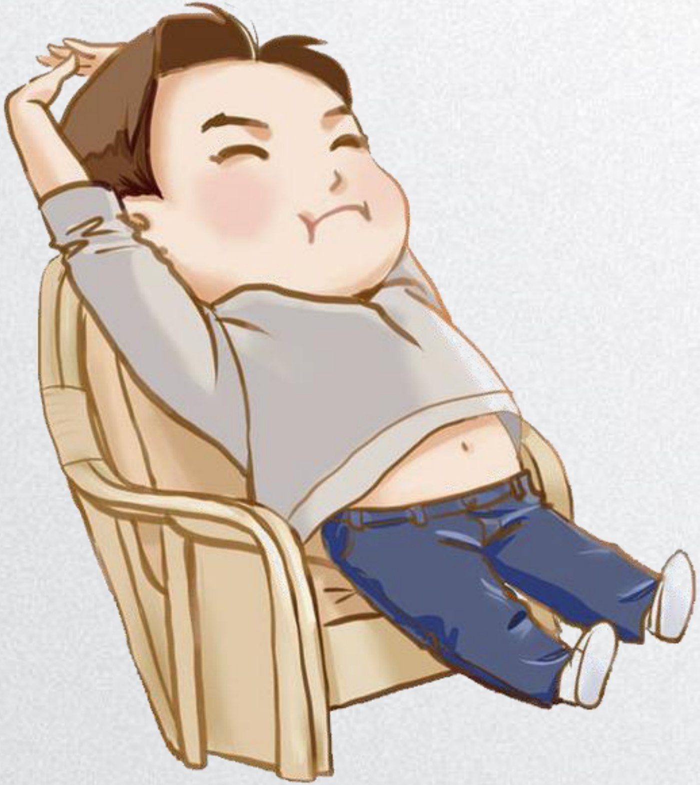
- Hinge loss → SVM
  - $\max(0,1-y_i w^T x_i)+\frac{\lambda}{2}||w||^2$
    - 对w求导
    - 注意y_i是标量，x_i是一个列向量， w是一个列向量（权重）
    - $||w||^2$为w和自己点乘，即$w^T\ w$
  - 分两个部分
  - $\frac{\partial}{\partial w}[\max(0,1-y_i w^T x_i)]=\begin{cases}-y_i x_i, if\ 1-y_i w^T x_i\geq 0\\ 0,, if\ 1-y_i w^T x_i<0\end{cases}$
  - $\frac{\partial}{\partial w}\left[\frac{\lambda}{2}||w||^2\right]=\frac{\partial}{\partial w}\left(\frac{\lambda}{2}w^T\ w\right)=\frac{\lambda}{2}(w+w)=\lambda w$

$$\frac{\partial \boldsymbol{a}^T\mathbf{x}}{\partial \mathbf{x}}=\frac{\partial \mathbf{x}^T\boldsymbol{a}}{\partial \mathbf{x}}=\boldsymbol{a}$$

$$\frac{\partial \boldsymbol{a}^T\mathbf{x}}{\partial \mathbf{x}}=\frac{\partial \mathbf{x}^T\boldsymbol{a}}{\partial \mathbf{x}}=\boldsymbol{a}$$

分别对$w^T\ w$的两个w求导，再加起来

矩阵求导



矩阵求导

当$\xi_i \geq 0$时
$1 - y_i(\mathrm{w}^T\ \mathrm{K}(\mathrm{x}, \mathrm{x}_i) + \mathrm{b})$ >0

- Hinge loss → kernelized SVM
  - $\frac{1}{2}\mathrm{w}^T K(x, x^T)\mathrm{w} + \ C\sum_{i=1}^{n} \max(0, 1 - y_i(\mathrm{w}^T\ \mathrm{K}(\mathrm{x}, \mathrm{x}_i) + \mathrm{b})\ )$
    - 对w求导，对b求导
  - $\frac{\partial \mathrm{L}}{\partial \mathrm{w}} = \ K\left(x, x^T\right)\mathrm{w} - \mathrm{C}\sum_{i=1,\xi_i\geq 0}^{n} y_i\ \mathrm{K}(\mathrm{x}, \mathrm{x}_i)$

$$\frac{\partial \mathbf{x}^T\mathbf{B}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{B} + \mathbf{B}^T)\mathbf{x}$$

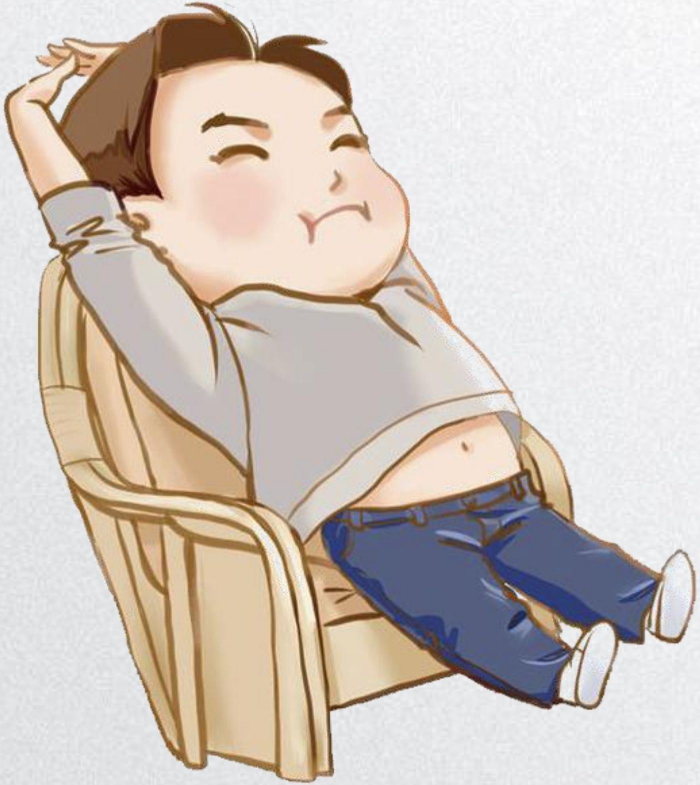
$K\left(x, x^T\right)$为一个对称矩阵

$$\frac{\partial \mathbf{x}^T\boldsymbol{a}}{\partial \mathbf{x}} = \boldsymbol{a}$$

把$\mathrm{K}(\mathrm{x}, \mathrm{x}_i)$看作一个列向量

  - $\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = -C\sum_{i=1,\xi_i\geq 0}^{n} y_i$
    - 注意b是标量，y_i是标量

矩阵求导



矩阵求导

- 一元/多元线性回归的解析解
  - $\frac{1}{2n}(\mathbb{Y}-\mathbb{X}\theta)^T(\mathbb{Y}-\mathbb{X}\theta)=\frac{1}{2n}(\mathbb{Y}^T-\theta^T\mathbb{X}^T)(\mathbb{Y}-\mathbb{X}\theta)$
    - 对θ求导
  - $\frac{1}{2n}(\mathbb{Y}^T\mathbb{Y}-\mathbb{Y}^T\mathbb{X}\boldsymbol{\theta}-\boldsymbol{\theta}^{\mathrm{T}}\mathbb{X}^T\mathbb{Y}+\boldsymbol{\theta}^{\mathrm{T}}\mathbb{X}^T\mathbb{X}\boldsymbol{\theta})$

$$\frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}}=\mathbf{a} \qquad \frac{\partial \mathbf{x}^T\mathbf{a}}{\partial \mathbf{x}}=\mathbf{a}$$

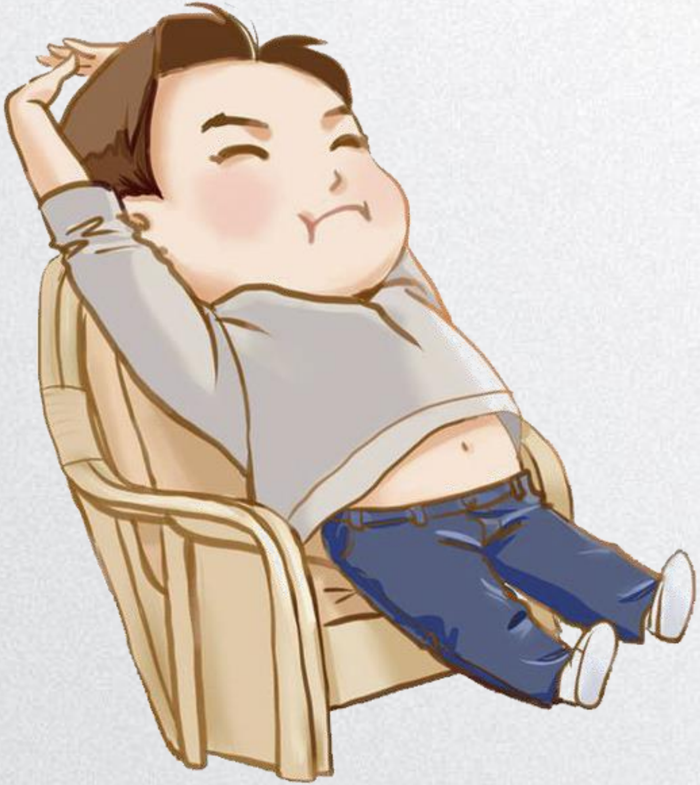
分别对两个求导

$$\boldsymbol{\theta}^{\mathrm{T}}\boxed{\mathbb{X}^T\mathbb{X}\boldsymbol{\theta}} \qquad \boxed{\boldsymbol{\theta}^{\mathrm{T}}\mathbb{X}^T\mathbb{X}}\boldsymbol{\theta}$$

  - $\frac{1}{2n}\left(0-\mathbb{X}^T\mathbb{Y}-\mathbb{X}^T\mathbb{Y}+\mathbb{X}^T\mathbb{X}\boldsymbol{\theta}+\mathbb{X}^T\mathbb{X}\boldsymbol{\theta}\right)$

$$\frac{\partial \mathbf{x}^T\mathbf{a}}{\partial \mathbf{x}}=\frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}}=\mathbf{a}$$

矩阵求导



矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]} \rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]} \rightarrow A^{[2]}$

损失函数J
二元交叉熵

- 神经网络
  - (1) $\boxed{Z^{[1]} = W^{[1]}X + b^{[1]} = \mathbf{(64*10)}(10*\ \mathbf{1}) + \mathbf{(64*1)} = \mathbf{(64*\ 1)}}$
  - (2) $A^{[1]} = \sigma(Z^{[1]}) = (64*\ \mathbf{1})$
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]} = \mathbf{(1*64)}(64*\ \mathbf{1}) + \mathbf{(1*1)} = (1*\ \mathbf{1})$
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]}) = \mathbf{(1*\ 1)}$
  - 处理过程 $A^{[0]} \rightarrow Z^{[1]} \rightarrow A^{[1]} \rightarrow Z^{[2]} \rightarrow A^{[2]}$
  - X即$A^{[0]}$

有1个样本，每个样本为10×1的列向量

从输入层计算隐藏层

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]} \rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]} \rightarrow A^{[2]}$

损失函数J
二元交叉熵

有1个样本，每个样本为10×1的列向量

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* **1**)+**(64*1)= (64* 1)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* **1**)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* **1**) +**(1*1)**=(1* **1**)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1* 1)**
  - 处理过程 $A^{[0]} \rightarrow Z^{[1]} \rightarrow A^{[1]} \rightarrow Z^{[2]} \rightarrow A^{[2]}$
  - X即$A^{[0]}$

隐藏层的非线性传导

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]} \rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]} \rightarrow A^{[2]}$

损失函数J
二元交叉熵

有1个样本，每个样本为10×1的列向量

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* 1)+**(64*1)= (64* 1)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* 1)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* 1) +**(1*1)**=(1* 1)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1* 1)**
  - 处理过程 $A^{[0]} \rightarrow Z^{[1]} \rightarrow A^{[1]} \rightarrow Z^{[2]} \rightarrow A^{[2]}$
  - X即$A^{[0]}$

从隐藏层计算输出层

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]} \rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]} \rightarrow \boxed{A^{[2]}}$

损失函数J
二元交叉熵

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* 1)+**(64*1)= (64* 1)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* 1)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* 1) +**(1*1)**=(1* 1)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1* 1)**
  - 处理过程 $A^{[0]} \rightarrow Z^{[1]} \rightarrow A^{[1]} \rightarrow Z^{[2]} \rightarrow A^{[2]}$
  - X即$A^{[0]}$

有1个样本，每个样本为10×1的列向量

输出层的非线性传导

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]} \rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]} \rightarrow A^{[2]}$

损失函数J
二元交叉熵

1个样本，每个样本为10×1的列向量

- 神经网络
  - 处理过程 $A^{[0]} \rightarrow Z^{[1]} \rightarrow A^{[1]} \rightarrow Z^{[2]} \rightarrow A^{[2]}$
  - X即$A^{[0]}$
  - 最后用$A^{[2]}$构造损失函数，，注意$A^{[2]}$即预测值$\hat{y}$
    - 二值分类器(0/1)的交叉熵损失函数的形式为
    - $\boxed{J = -\frac{1}{n}((Ylog(A^{[2]}) + (1-Y)\log(1-A^{[2]}))}$
    - Y=(1*，**1**)
    - n=1

根据损失函数计算损失值

矩阵求导

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* **1**)+**(64*1)**= **(64* 1)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* **1**)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* **1**) +**(1*1)**=(1* **1**)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1* 1)**
  - 处理过程$A^{[0]} \to Z^{[1]} \to A^{[1]} \to Z^{[2]} \to A^{[2]}$
  - 最后用$A^{[2]}$构造损失函数，，注意$A^{[2]}$即预测值$\hat{y}$
    - 二值分类器(0/1)的交叉熵损失函数的形式为
    - $J = -\frac{1}{n}((Ylog(A^{[2]}) + (1-Y)\log(1-A^{[2]}))$
  - 计算损失函数对$W^{[1]}$、$b^{[1]}$、$W^{[2]}$、$b^{[2]}$的导数

$$dZ^{[2]} = dA^{[2]}g'(Z^{[2]})$$ **(1*1) (1*1)**

$$dW^{[2]} = \frac{dJ}{dW^{[2]}} = \frac{\boldsymbol{dJ}}{\mathbf{dA^{[2]}}}\frac{\mathbf{dA^{[2]}}}{\mathbf{dZ^{[2]}}}\frac{dZ^{[2]}}{dW^{[2]}} = \frac{\boldsymbol{dJ}}{\mathbf{dZ^{[2]}}}\frac{dZ^{[2]}}{dW^{[2]}}$$

$$= dZ^{[2]}\left(A^{[1]}\right)^T$$

(1*1)(1*64)

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{b}}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^T$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T$$

$$\frac{dZ^{[2]}}{dW^{[2]}} = \frac{d(W^{[2]}A^{[1]}+b^{[2]})}{dW^{[2]}} = \frac{d(EW^{[2]}A^{[1]}+b^{[2]})}{dW^{[2]}} = E^T\left(A^{[1]}\right)^T$$

E为单位矩阵

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]}\to A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]}\to A^{[2]}$

损失函数J
二元交叉熵

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* 1)+**(64*1)**= **(64*** 1**)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* 1)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* 1) +**(1*1)**=(1* 1)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1*** 1**)**
  - 处理过程$A^{[0]} \to Z^{[1]}\to A^{[1]}\to Z^{[2]}\to A^{[2]}$
  - (5)最后用$A^{[2]}$构造损失函数，，注意$A^{[2]}$即预测值$\hat{y}$
    - 二值分类器(0/1)的交叉熵损失函数的形式为
    - $J = -\frac{1}{n}((Ylog(A^{[2]}) + (1-Y)\log(1-A^{[2]}))$
  - 计算损失函数对$W^{[1]}$、$b^{[1]}$、$W^{[2]}$、$b^{[2]}$的导数

$$\frac{dJ}{db^{[2]}} = \frac{dJ}{dA^{[2]}}\frac{dA^{[2]}}{dZ^{[2]}}\frac{dZ^{[2]}}{db^{[2]}}$$
$= [A^{[2]} - Y][1] = [A^{[2]} - Y]$
$= dZ^{[2]}$
(1*1)

$$\frac{dJ}{dA^{[1]}} = \frac{dJ}{dA^{[2]}}\frac{dA^{[2]}}{dZ^{[2]}}\frac{dZ^{[2]}}{dA^{[1]}} = \frac{dJ}{dZ^{[2]}}\frac{dZ^{[2]}}{dA^{[1]}}$$
$=dZ^{[2]}(W^{[2]})^T$即$(\boldsymbol{W}^{[2]})^{\boldsymbol{T}}\boldsymbol{dZ}^{[2]}$
(64*1)(1*1)

$$\frac{dZ^{[2]}}{dA^{[1]}} = \frac{d(W^{[2]}A^{[1]}+b^{[2]})}{dA^{[1]}} = (W^{[2]})^T$$

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]}\rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]}\rightarrow A^{[2]}$

损失函数J
二元交叉熵

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* 1)+**(64*1)**= **(64* 1)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* 1)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* 1) +**(1*1)**=(1* 1)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1* 1)**
  - 处理过程$A^{[0]} \rightarrow Z^{[1]}\rightarrow A^{[1]}\rightarrow Z^{[2]}\rightarrow A^{[2]}$
  - (5)最后用$A^{[2]}$构造损失函数，，注意$A^{[2]}$即预测值$\hat{y}$
    - 二值分类器(0/1)的交叉熵损失函数的形式为
    - $J = -\frac{1}{n}((Ylog(A^{[2]}) + (1-Y)\log(1-A^{[2]}))$
  - 计算损失函数对$W^{[1]}$、$b^{[1]}$、$W^{[2]}$、$b^{[2]}$的导数

$$dZ^{[1]} = dA^{[1]}g'(Z^{[1]})$$

(64*1) (64*1)

$$dW^{[1]} = \frac{dJ}{dW^{[1]}} = \frac{\boldsymbol{dJ}}{\mathbf{dA^{[2]}}}\frac{\mathbf{dA^{[2]}}}{\mathbf{dZ^{[2]}}}\frac{dA^{[2]}}{dA^{[1]}}\frac{dA^{[1]}}{dZ^{[1]}}\frac{dZ^{[1]}}{dW^{[1]}}$$

$$= \frac{\boldsymbol{dJ}}{\mathbf{dZ^{[1]}}}\frac{dZ^{[1]}}{dW^{[1]}}$$

$$= dZ^{[1]}(A^{[0]})^T$$

(64*1)(1*10)

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{b}}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^T$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T$$

$$\frac{dZ^{[1]}}{dW^{[1]}} = \frac{d(W^{[1]}A^{[0]}+b^{[1]})}{dW^{[1]}} = \frac{d(EW^{[1]}A^{[0]}+b^{[1]})}{dW^{[1]}} = E^T(A^{[0]})^T$$

E为单位矩阵

矩阵求导

X
10元素
$A^{[0]}$

$W^{[1]}$、$b^{[1]}$

隐藏层
64神经元
$Z^{[1]} \rightarrow A^{[1]}$

$W^{[2]}$、$b^{[2]}$

输出层
1个神经元
$Z^{[2]} \rightarrow A^{[2]}$

损失函数J
二元交叉熵

- 神经网络
  - (1) $Z^{[1]} = W^{[1]}X + b^{[1]}$=**(64*10)**(10* 1)+**(64*1)**= **(64* 1)**
  - (2) $A^{[1]} = \sigma(Z^{[1]})$=(64* 1)
  - (3) $Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$=**(1*64)**(64* 1) +**(1*1)**=(1* 1)
  - (4) $\hat{y} = A^{[2]} = \sigma(Z^{[2]})$ =**(1* 1)**
  - 处理过程$A^{[0]} \rightarrow Z^{[1]} \rightarrow A^{[1]} \rightarrow Z^{[2]} \rightarrow A^{[2]}$
  - (5)最后用$A^{[2]}$构造损失函数，，注意$A^{[2]}$即预测值$\hat{y}$
    - 二值分类器(0/1)的交叉熵损失函数的形式为
    - $J = -\frac{1}{n}((Ylog(A^{[2]}) + (1-Y)\log(1-A^{[2]}))$
  - 计算损失函数对$W^{[1]}$、$b^{[1]}$、$W^{[2]}$、$b^{[2]}$的导数

$db^{[1]} = \frac{dJ}{db^{[1]}} =$

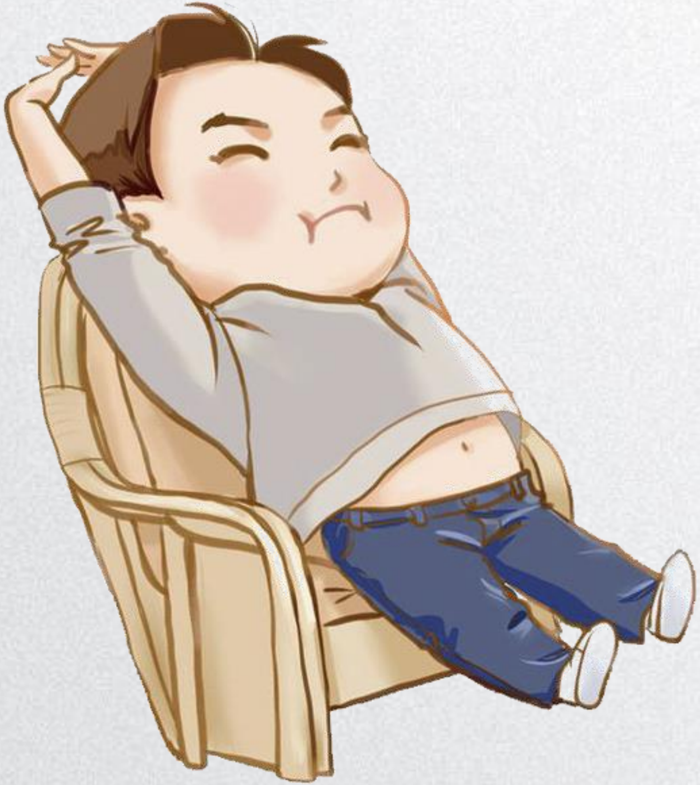
$\frac{\boldsymbol{dJ}}{\mathbf{dA}^{[2]}} \frac{\mathbf{dA}^{[2]}}{\mathbf{dZ}^{[2]}} \frac{dZ^{[2]}}{dA^{[1]}} \frac{dA^{[1]}}{dZ^{[1]}} \frac{dZ^{[1]}}{db^{[1]}} = \frac{\boldsymbol{dJ}}{\mathbf{dZ}^{[1]}} \frac{dZ^{[1]}}{db^{[1]}} =$

$dZ^{[1]}[1]$ =

$dZ^{[1]}$

(64*1)

矩阵求导



矩阵求导

- NMF梯度（矩阵求导）
- 假设我们把V分解为W和H
- $L = \min \frac{1}{2} ||V - WH||^2$
- $\frac{\partial L}{\partial w_{ik}} = \boxed{-[(V - WH)H^T]}_{ik}$
- $\frac{\partial L}{\partial h_{kj}} = -[W^T(V - WH)]_{kj}$

Frobenius norm

$$\frac{\partial}{\partial \mathbf{X}} ||\mathbf{X}||_F^2 = \frac{\partial}{\partial \mathbf{X}} \mathrm{Tr}(\mathbf{X}\mathbf{X}^H) = 2\mathbf{X}$$

$$||\mathbf{A}||_F = \sqrt{\sum_{ij} |A_{ij}|^2} = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^H)} \qquad \text{(Frobenius)}$$

$(V - WH)(V - WH)^T =$
$(V - WH)(V^T - H^T W^T) =$
$VV^T - WHV^T - VH^T W^T + WHH^T W^T$
对着W求导，得到
$0 - VH^T - VH^T + WHH^T + WHH^T$
$\boxed{= -2(V - WH) H^T}$

$\boxed{WHH^T}W^T$

$WH\boxed{H^T W^T}$

矩阵求导

- NMF梯度（矩阵求导）
- 假设我们把V分解为W和H
- $L = \min \frac{1}{2} ||V - WH||^2$
- $\frac{\partial L}{\partial w_{ik}} = -[(V - WH)H^T]_{ik}$
- $\frac{\partial L}{\partial h_{kj}} = \boxed{-[W^T(V - WH)}]_{kj}$

Frobenius norm

$$\frac{\partial}{\partial \mathbf{X}}||\mathbf{X}||_F^2 = \frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}\mathbf{X}^H) = 2\mathbf{X}$$

$$||\mathbf{A}||_F = \sqrt{\sum_{ij}|A_{ij}|^2} = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^H)} \qquad \text{(Frobenius)}$$

$(V - WH)^T(V - WH) =$

$(V^T - H^TW^T)(V - WH) =$

$V^TV - H^TW^TV - V^TWH + H^TW^TWH$

对着H求导，得到

$0 - W^TV - W^TV + W^TWH + W^TWH$

$\boxed{= -2W^T(V - WH)}$

$\boxed{H^TW^TW}H$

$H^T\boxed{W^TWH}$

矩阵求导