深度学习平台与应用

第二讲：线性分类器

范琦

fanqi@nju.edu.cn

2024年9月11日

课程考核

- 课后作业：20%
- 课程项目：30%
- 期末考试：50%

大　　纲

- 图像分类
- K-NN 最近邻分类器
- 线性分类器

图像分类

- 图像分类是**最核心的计算机视觉任务**

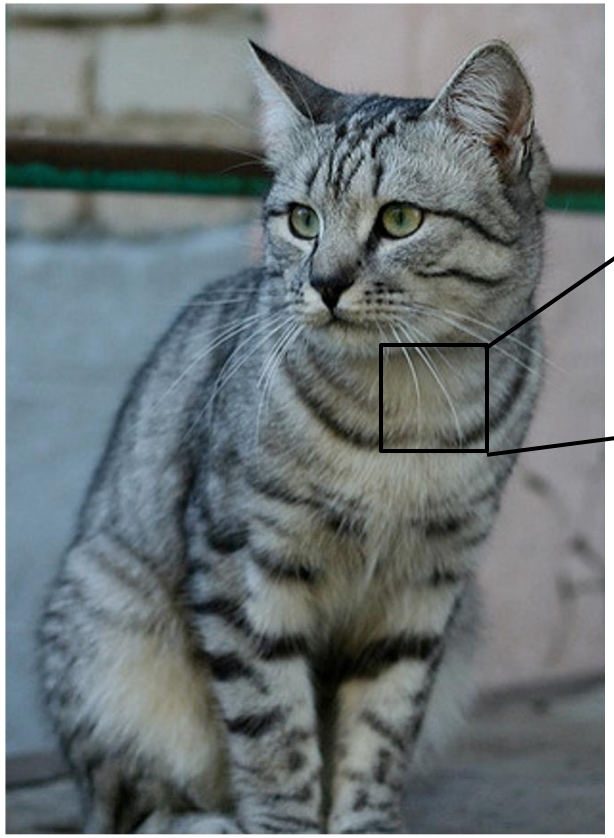
假设有一个标签集合
{人，狗，猫，汽车，…}

模型的分类预测 → 猫

图像分类

南京大学 NANJING UNIVERSITY

- 问题：语义鸿沟（Semantic Gap）



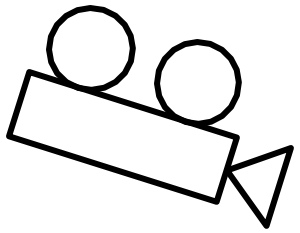
```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

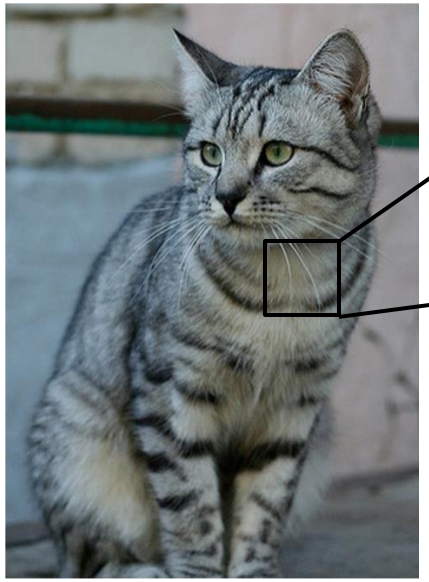
人眼看到的内容：
一只猫的图像

计算机看到的内容：
0-255的整数组成的矩阵
例如，RGB图像会表示为HxWx3的矩阵

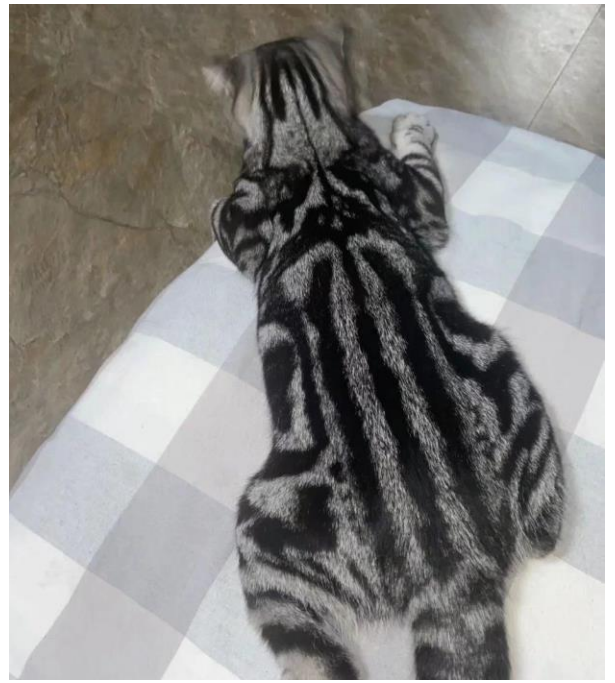
图像分类

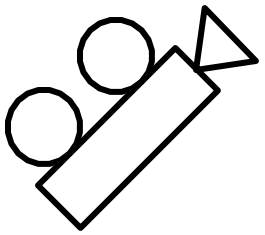
- 挑战：视角差异（Viewpoint Variation）

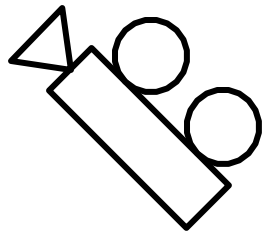




```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```







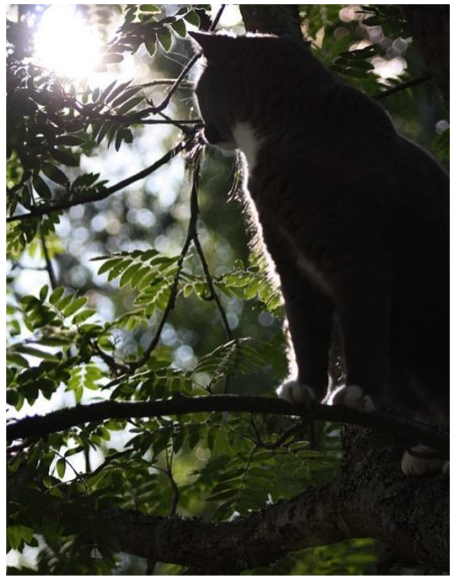
当相机视角改变时，
所有像素都会变化

图像分类

南京大学 NANJING UNIVERSITY

- 挑战：光照变化（Illumination）



当光照变化时，所有像素都会改变

图像分类

南京大学 NANJING UNIVERSITY

- 挑战：杂乱的背景（Background Clutter）

图像分类

南京大学 NANJING UNIVERSITY

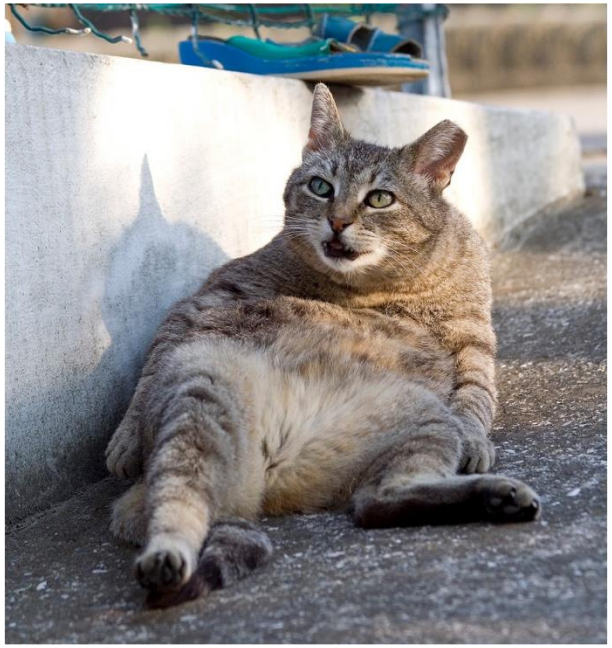
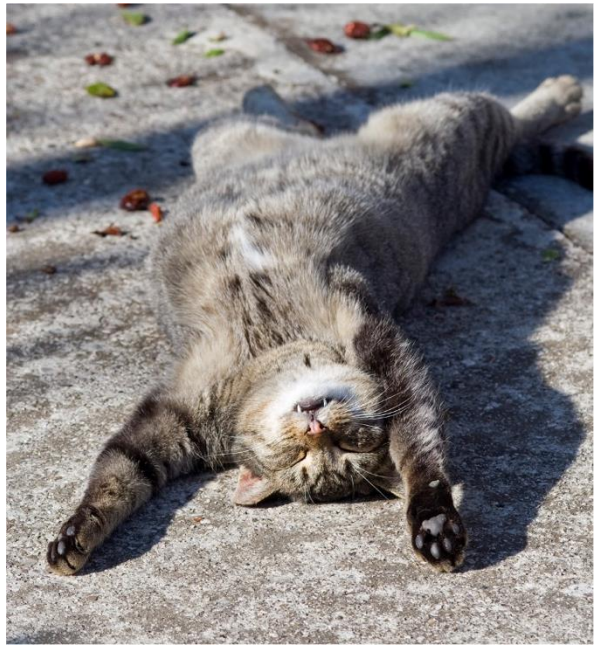
- 挑战：遮挡（Occlusion）

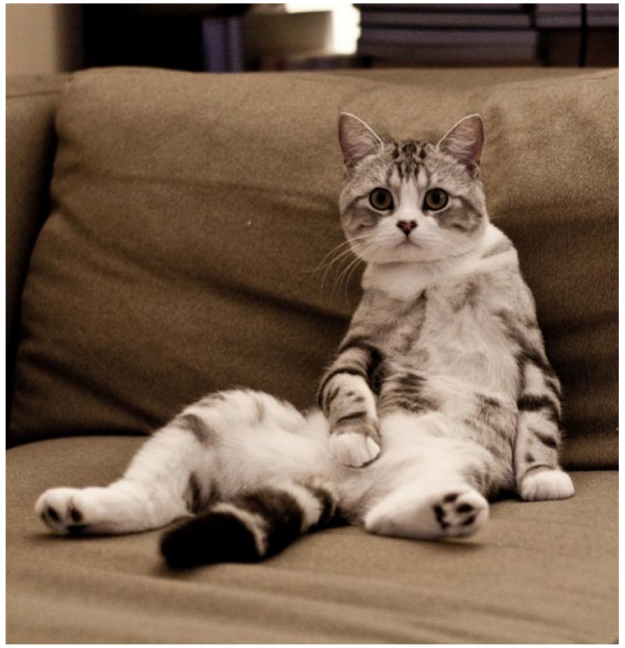


图像分类

南京大学 NANJING UNIVERSITY

- 挑战：形变（Deformation）



图像分类

- 挑战：**类内差异（Intra-Class Variation）**

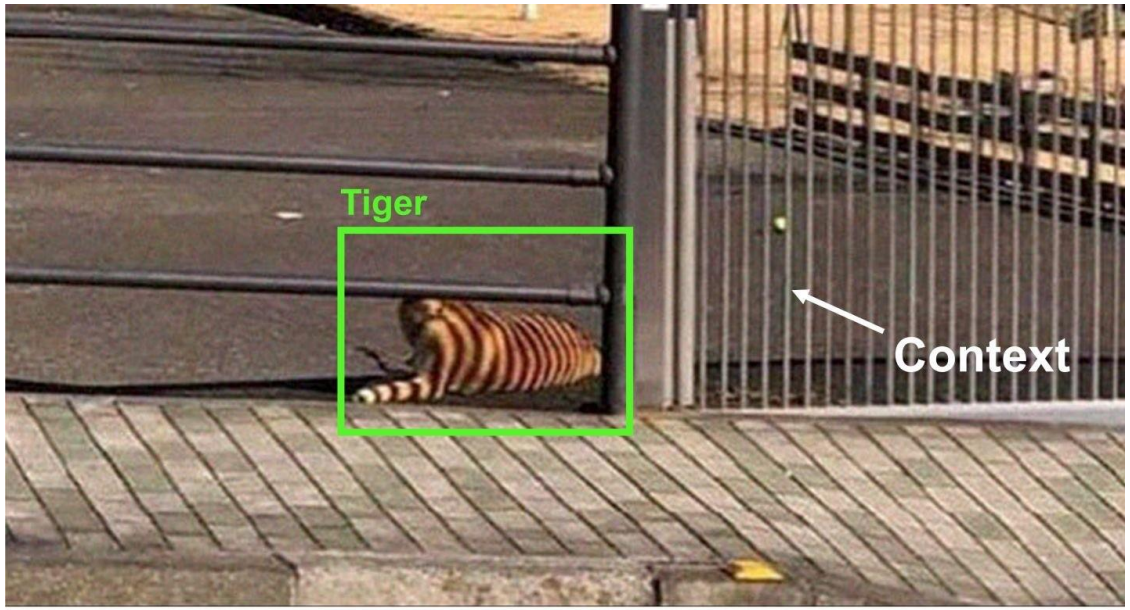
图像分类

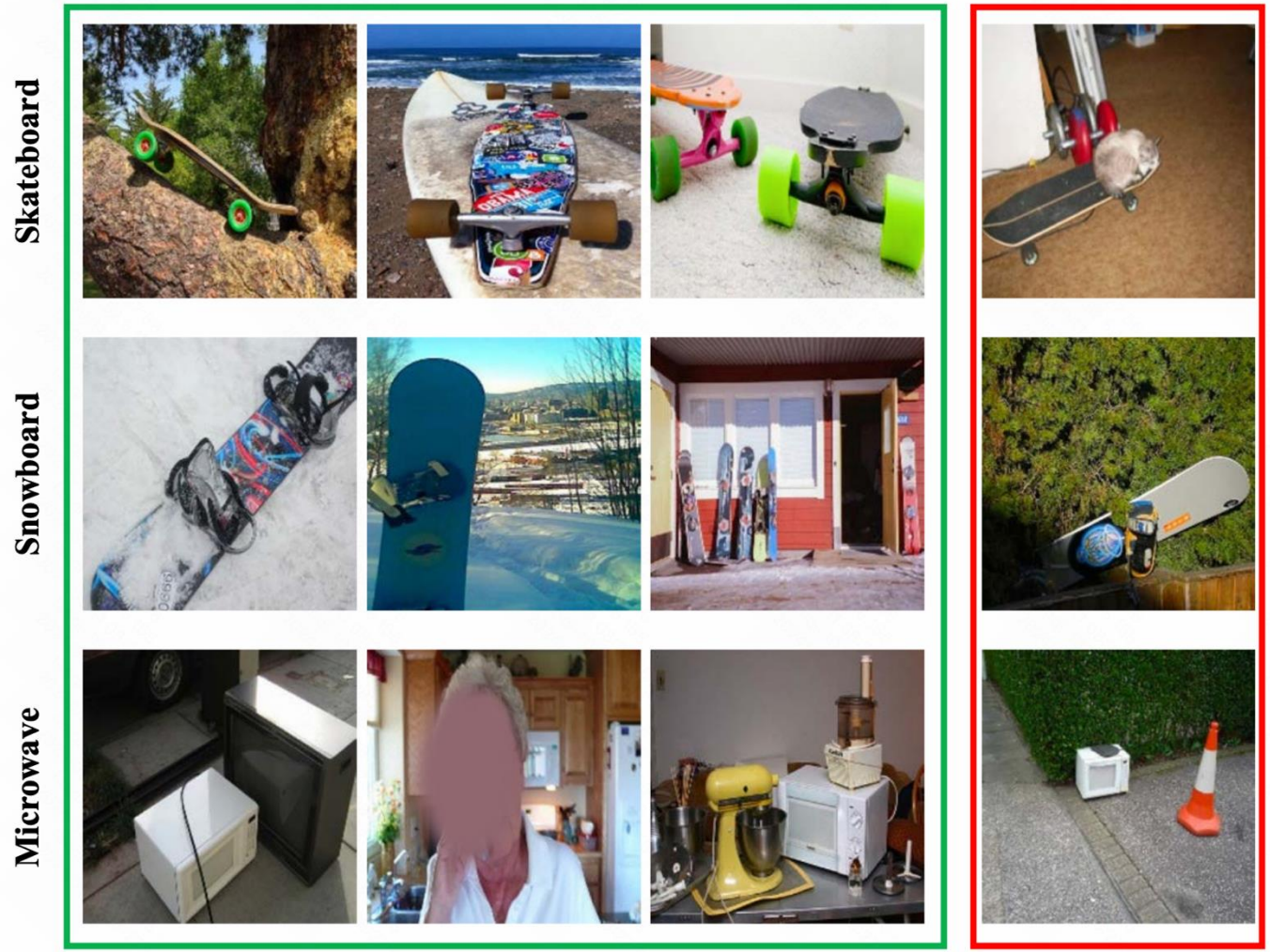
南京大学 NANJING UNIVERSITY

- 挑战：**类间相似（Inter-Class Similarity）**

图像分类

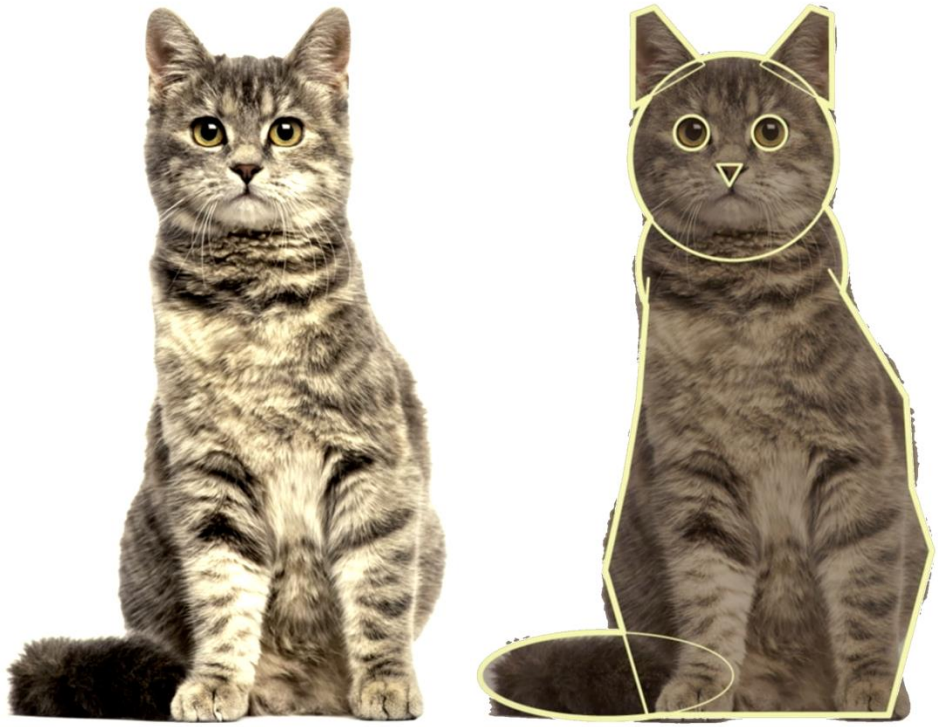
- 挑战：语义干扰（Context Disturbance）



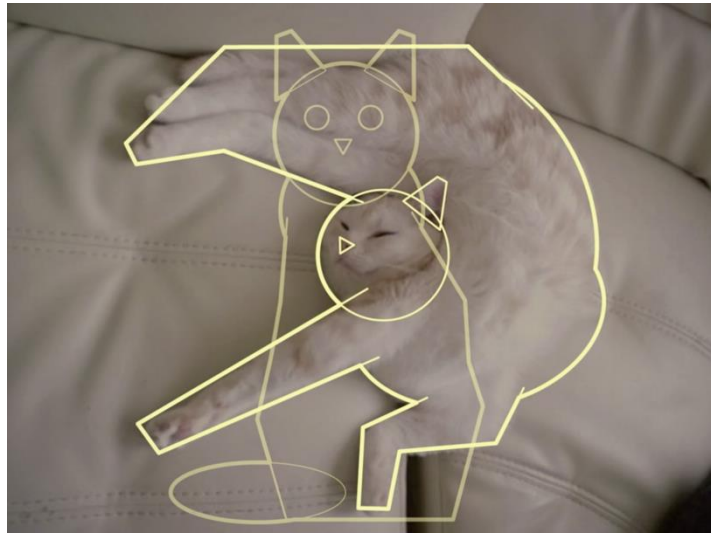


图像分类

- 图像分类器（Classifier）



与排序等算法不同，
无法对图像识别算法进行硬编码



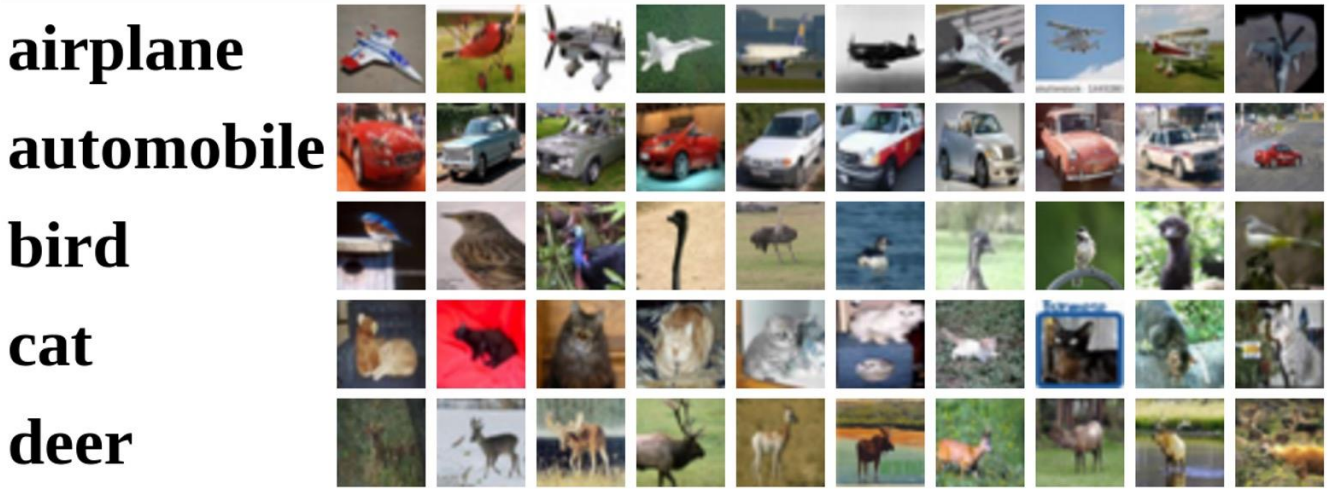
图像分类

南京大学 NANJING UNIVERSITY

基于规律的图像分类

图像分类

- 数据驱动（Data-Driven）的图像分类
  - 收集图像和标签的数据集
  - 使用机器学习算法训练分类器
  - 使用分类器对新图像进行分类




```
def train(images, labels):
  # Machine learning!
  return model
```

```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

大　　纲

- 图像分类
- K-NN 最近邻分类器
- 线性分类器

K-NN 最近邻分类器

南京大学 NANJING UNIVERSITY

- 最近邻分类器

```
def train(images, labels):
  # Machine learning!
  return model
```

→ 记住所有数据和标签

```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

→ 将测试图像预测为与其最相似的训练图像的标签

K-NN 最近邻分类器

- 最近邻分类器

训练图像 + 图像标签

测试图像

距离度量

$$\left| \quad , \quad \right| \rightarrow \mathbb{R}$$

K-NN 最近邻分类器

南京大学 NANJING UNIVERSITY

- **距离度量函数**

L1 距离: $$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

test image

56	32	10	18
90	23	128	133
24	26	178	200
2	0	255	220

-

training image

10	20	24	17
8	10	89	100
12	16	178	170
4	32	233	112

=

pixel-wise absolute value differences

46	12	14	1
82	13	39	33
12	10	0	30
2	32	22	108

add → 456

K-NN 最近邻分类器

南京大学 NANJING UNIVERSITY

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

记忆训练数据

对于每个测试数据：

- **找到与其最相似的训练数据**
- **将其预测为此训练数据的标签**

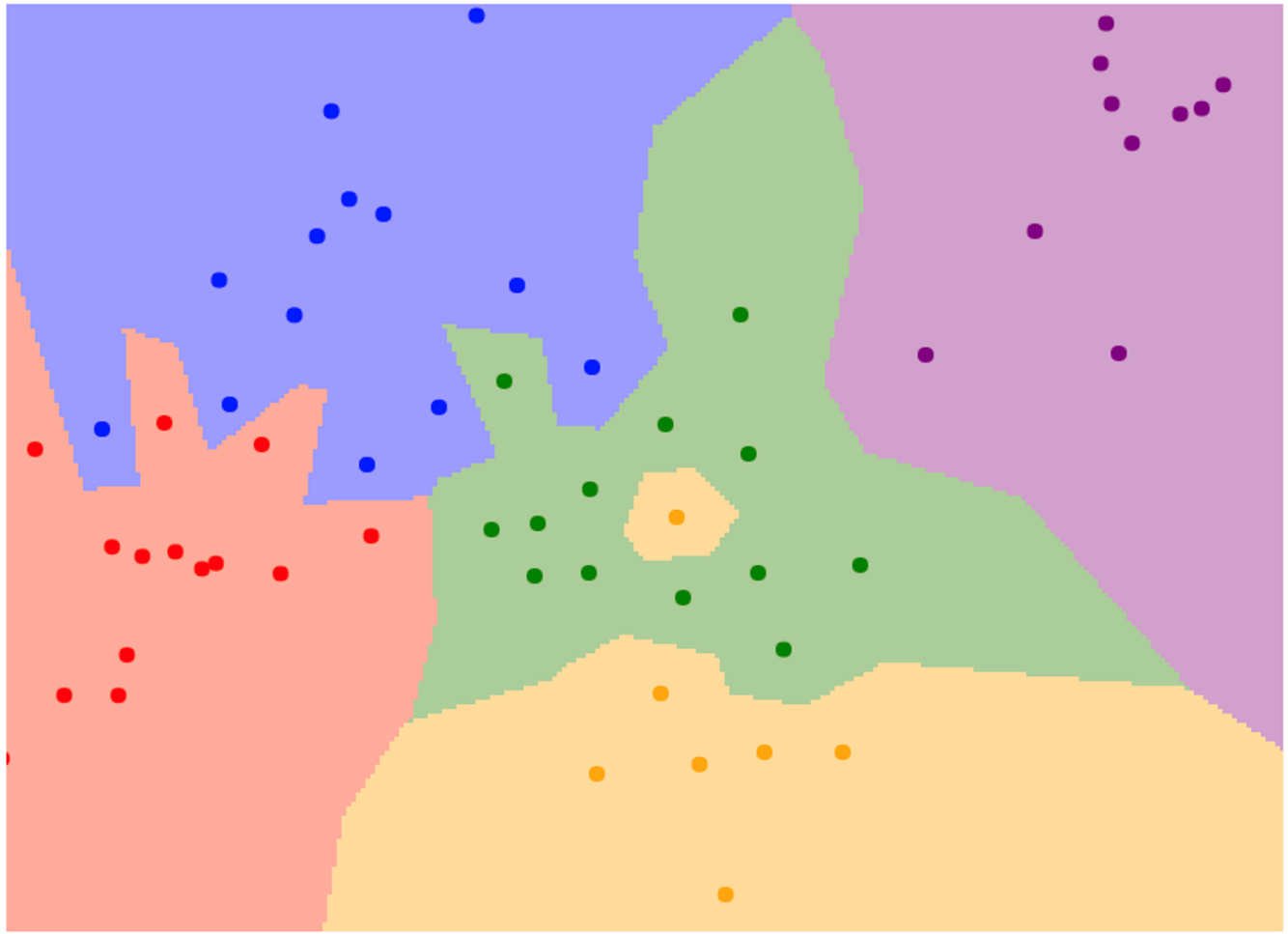
时间复杂度?

K-NN 最近邻分类器

- 1 近邻示例
- 存在的问题?

点：训练数据（2D）

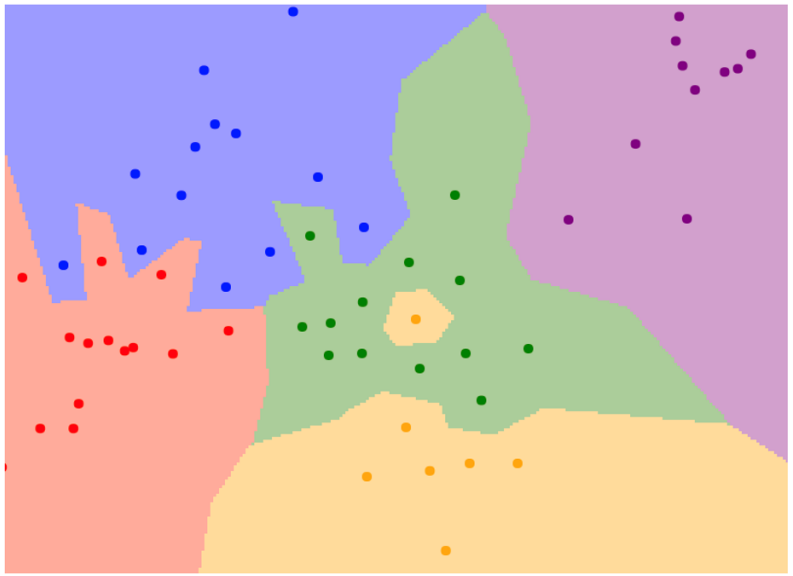
区域：类别空间



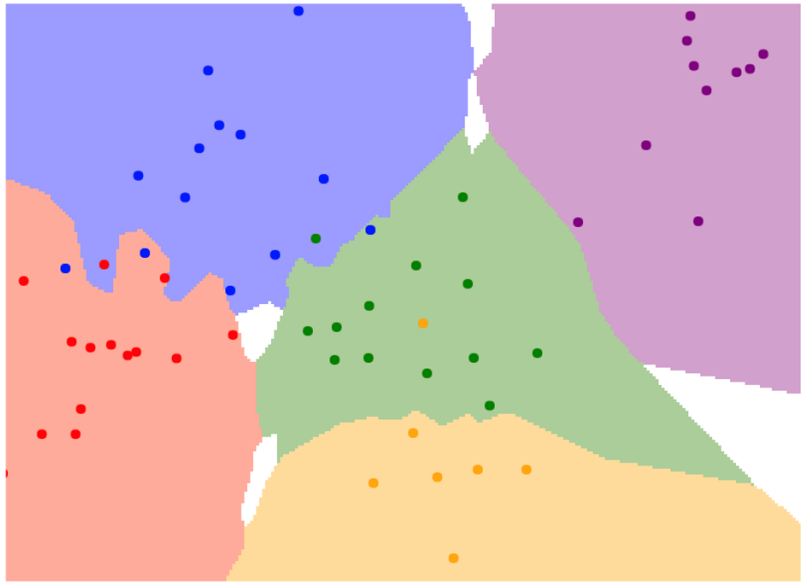
1-nearest neighbor

K-NN 最近邻分类器

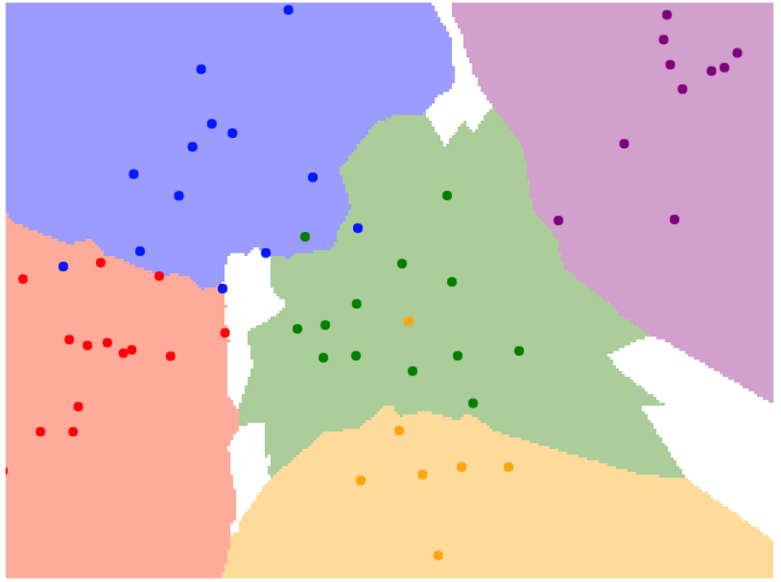
- K 近邻
- 问题：如果某个区域的 K 个最近邻是 K 个不同的类，怎么办?



K=1



K=3



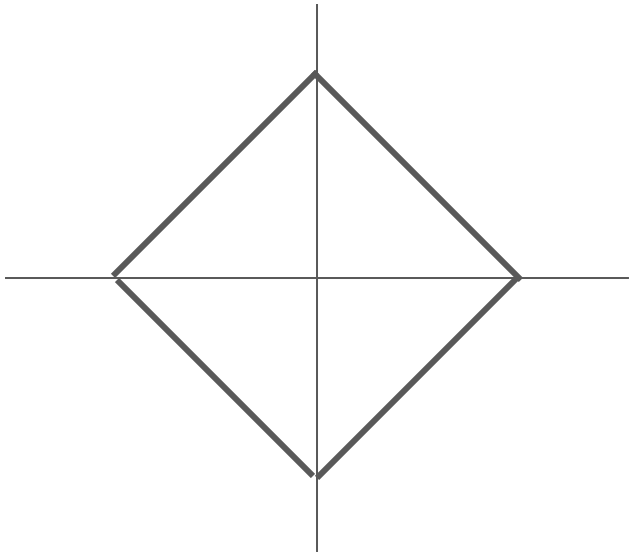
K=5

K-NN 最近邻分类器

- 距离度量函数

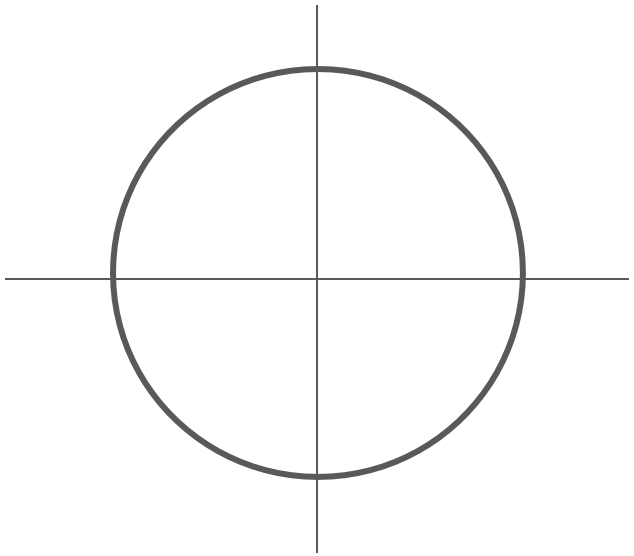
L1 (Manhattan) 距离

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



L2 (Euclidean) 距离

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

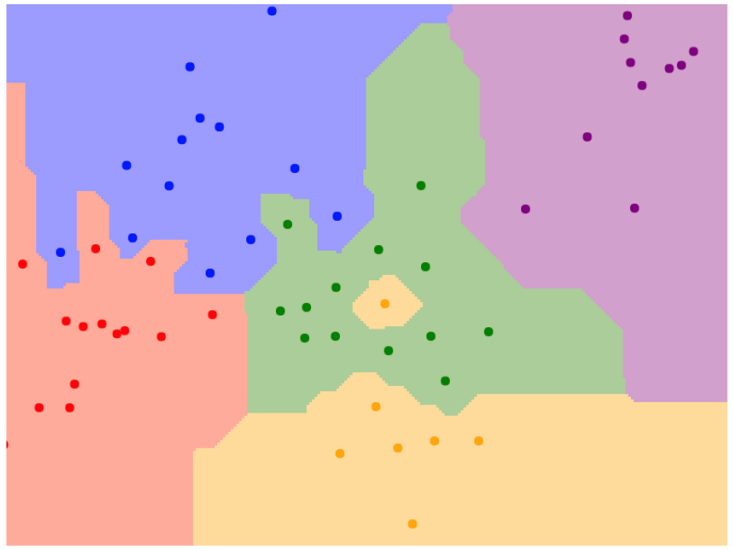


K-NN 最近邻分类器

- 距离度量函数

L1 (Manhattan) 距离

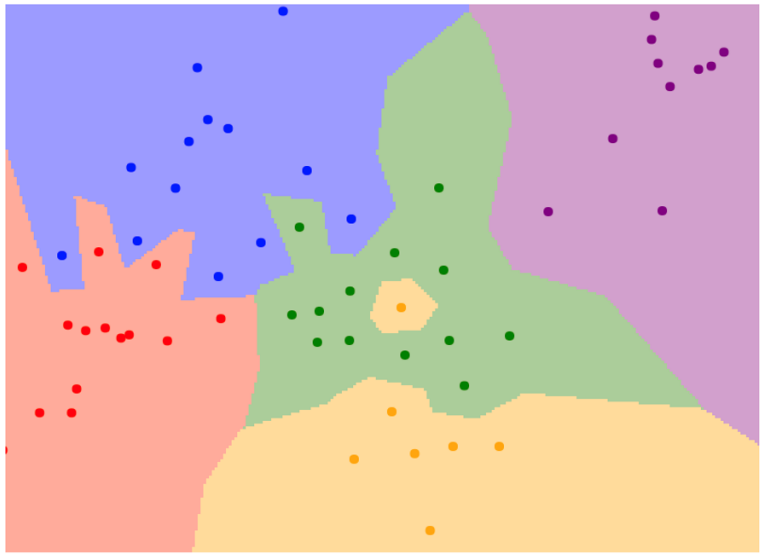
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



K=1

L2 (Euclidean) 距离

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



K=1

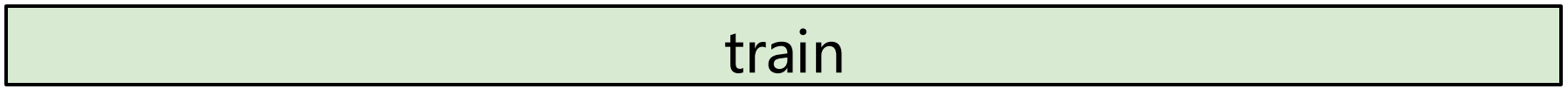
K-NN 最近邻分类器

- K-NN 超参数
  - 超参数：算法本身的设置
  - k的最佳值是什么?
  - 最好的距离度量函数是什么?
- 超参数的选择
  - 非常依赖问题/数据集
  - 尝试不同方法，看看哪种效果最好

K-NN 最近邻分类器

南京大学 NANJING UNIVERSITY

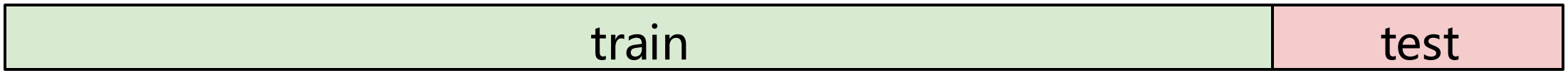
- 设置超参数
  - Idea 1：选择在训练集上表现最好的超参数
  - 不好的选择！为什么?
  - K=1 在训练集上是表现最好的超参数，但在测试集上的效果可能很差，难以泛化到测试集上



K-NN 最近邻分类器

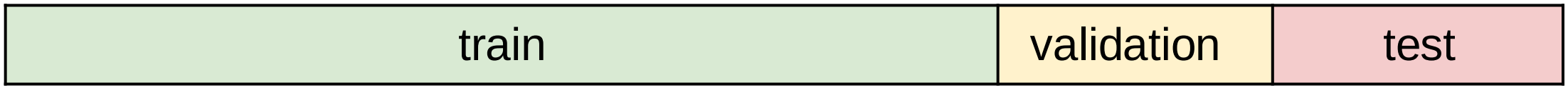
南京大学 NANJING UNIVERSITY

- 设置超参数
  - Idea 2：选择在测试集上表现最好的超参数
  - 被禁止的选择！为什么?
  - 数据泄露！
  - 不能在训练时以任何方式接触测试数据！



K-NN 最近邻分类器

- 设置超参数
  - Idea 3：将原始训练集 A 划分为训练集 B + 验证集 C
    - 在训练集 B 上训练模型
    - 在验证集 C 上选择模型超参数
    - 在测试集上进行测试



K-NN 最近邻分类器

- 交叉验证：将训练数据拆分为多个部分，分别使用每个部分作为验证，并将结果取平均值
- 一般用于小数据集，深度学习中不常用

fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

K-NN 最近邻分类器

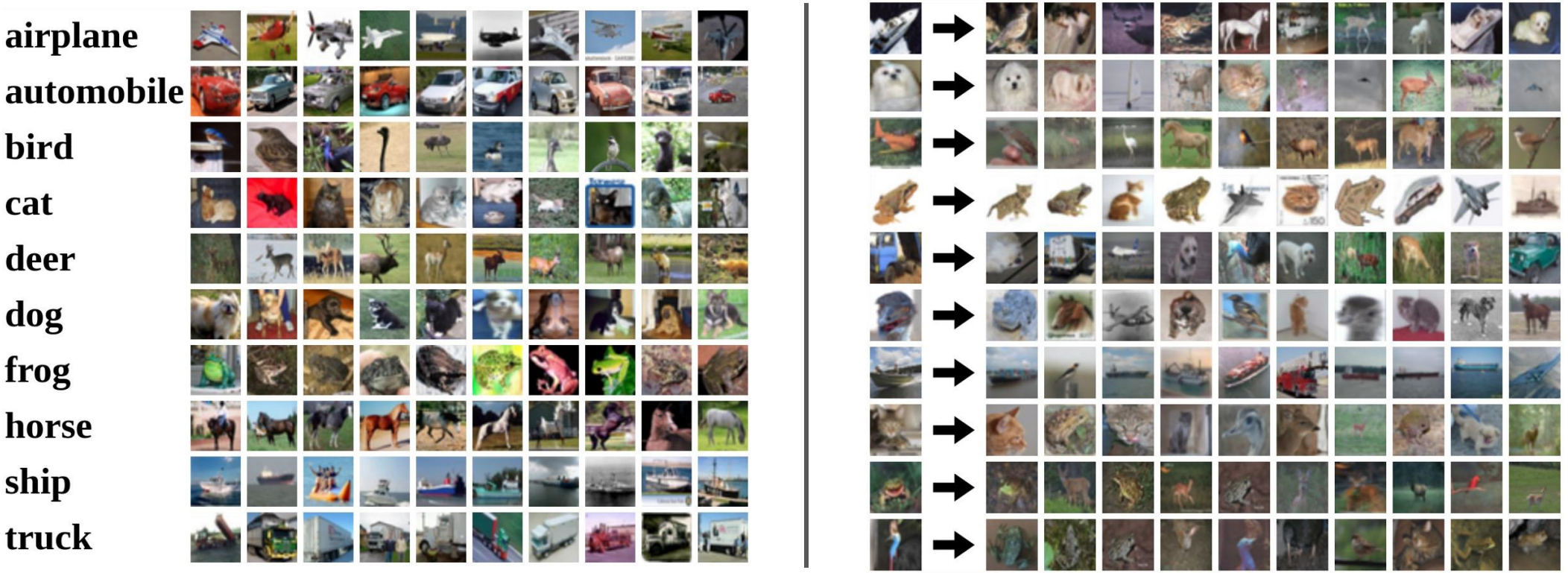
南京大学 NANJING UNIVERSITY

- 例子：CIFAR-10 数据集
  - 10 个类别，50000 张训练图像，10000 张测试图像

K-NN 最近邻分类器

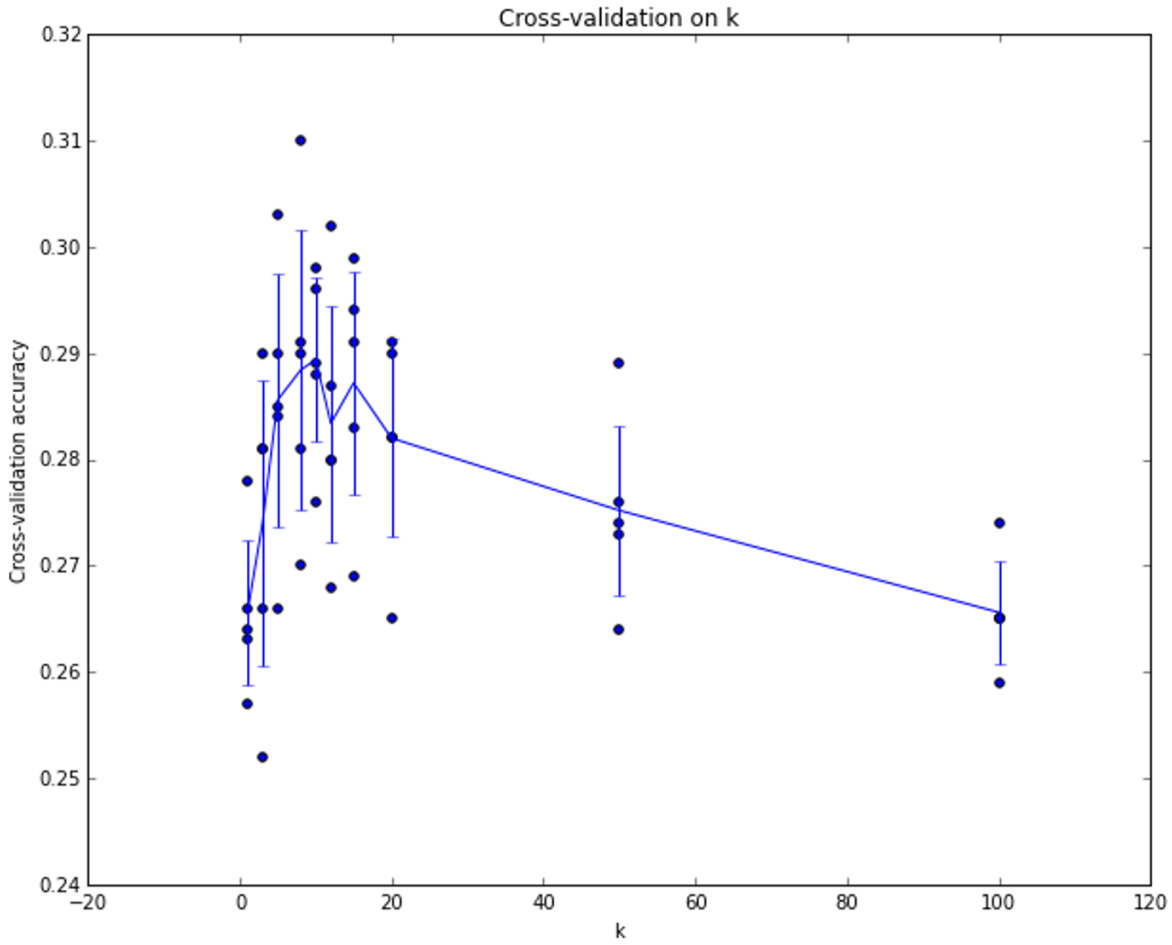
南京大学 NANJING UNIVERSITY

- 例子：CIFAR-10 数据集
  - 右侧：测试图像，及其最相似的训练图像



K-NN 最近邻分类器

- 例子：设置超参数
  - 5-折交叉验证
  - 点：单折验证结果
  - K=7 时效果最好



K-NN 最近邻分类器

- 最近邻分类效果
  - 看起来很像
  - 但存在错误

K-NN 最近邻分类器

- 这的结果像什么?
  - 图像检索!
  - 区别?
  - 预测目的不同!

K-NN 最近邻分类器

- 不要直接使用像素间的距离进行度量！
  - 像素间的距离度量的泛化性和稳定性都很差

Original

Occluded

Shifted (1 pixel)

Tinted

大　　纲

- 图像分类
- K-NN 最近邻分类器
- 线性分类器

线性分类器

- 线性分类器的定义

输入图像
32x32x3 的矩阵

f(x,W)

W
parameters or weights
参数或权重

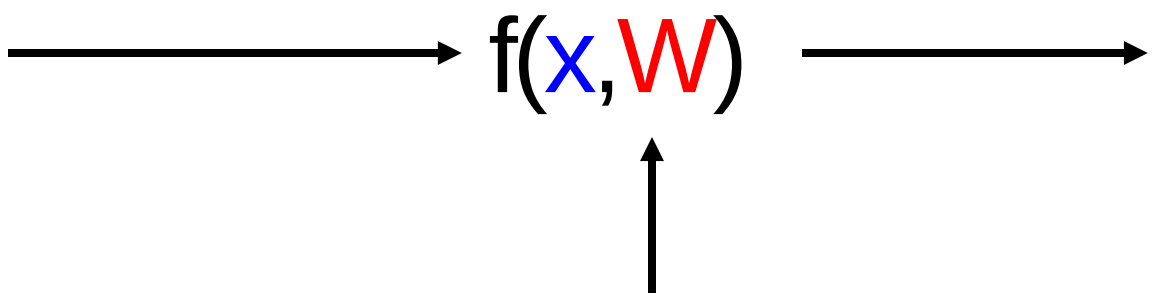
571.3	猫
1.25	狗
-132.2	船

线性分类器

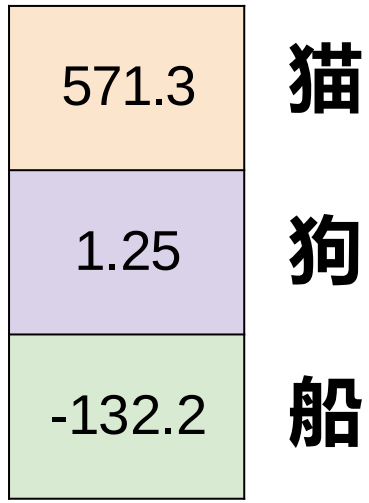
- 线性分类器的定义

$$f(x,W) = Wx$$

输入图像
32x32x3 的矩阵



W
parameters or weights
参数或权重



线性分类器

- 线性分类器的定义

$$f(x,W) = Wx$$

3072x1

3x1

3x3072

输入图像
32x32x3 的矩阵

f(x,W)

W

parameters or weights

参数或权重

571.3	猫
1.25	狗
-132.2	船

线性分类器

- 线性分类器的定义

$$f(x,W) = Wx + b$$

3072x1　　3x1

3x1　　3x3072

输入图像
32x32x3 的矩阵

f(x,W)

W

parameters or weights

参数或权重

571.3	猫
1.25	狗
-132.2	船

线性分类器

南京大学 NANJING UNIVERSITY

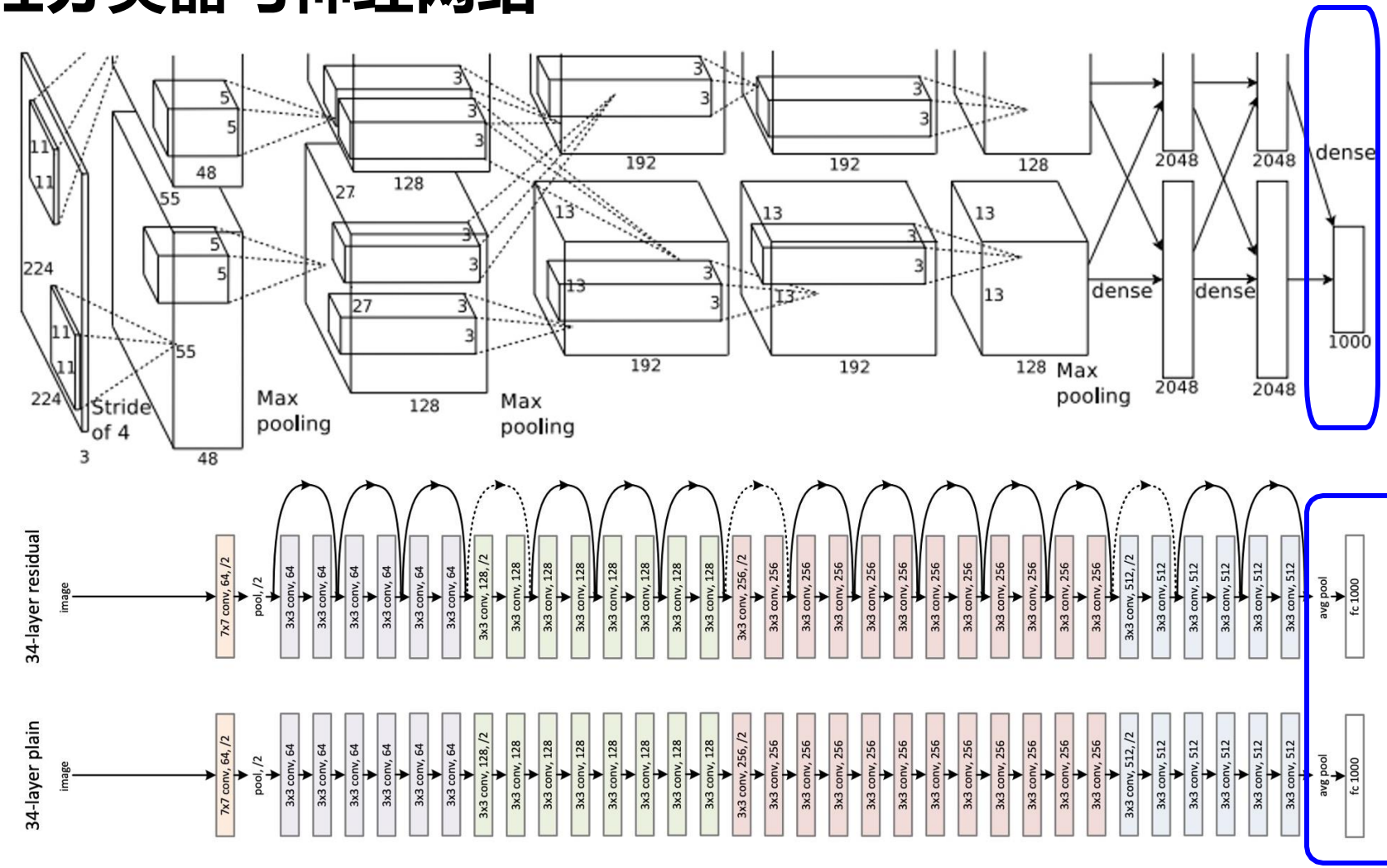
- 线性分类器与神经网络

线性分类器

神经网络

线性分类器

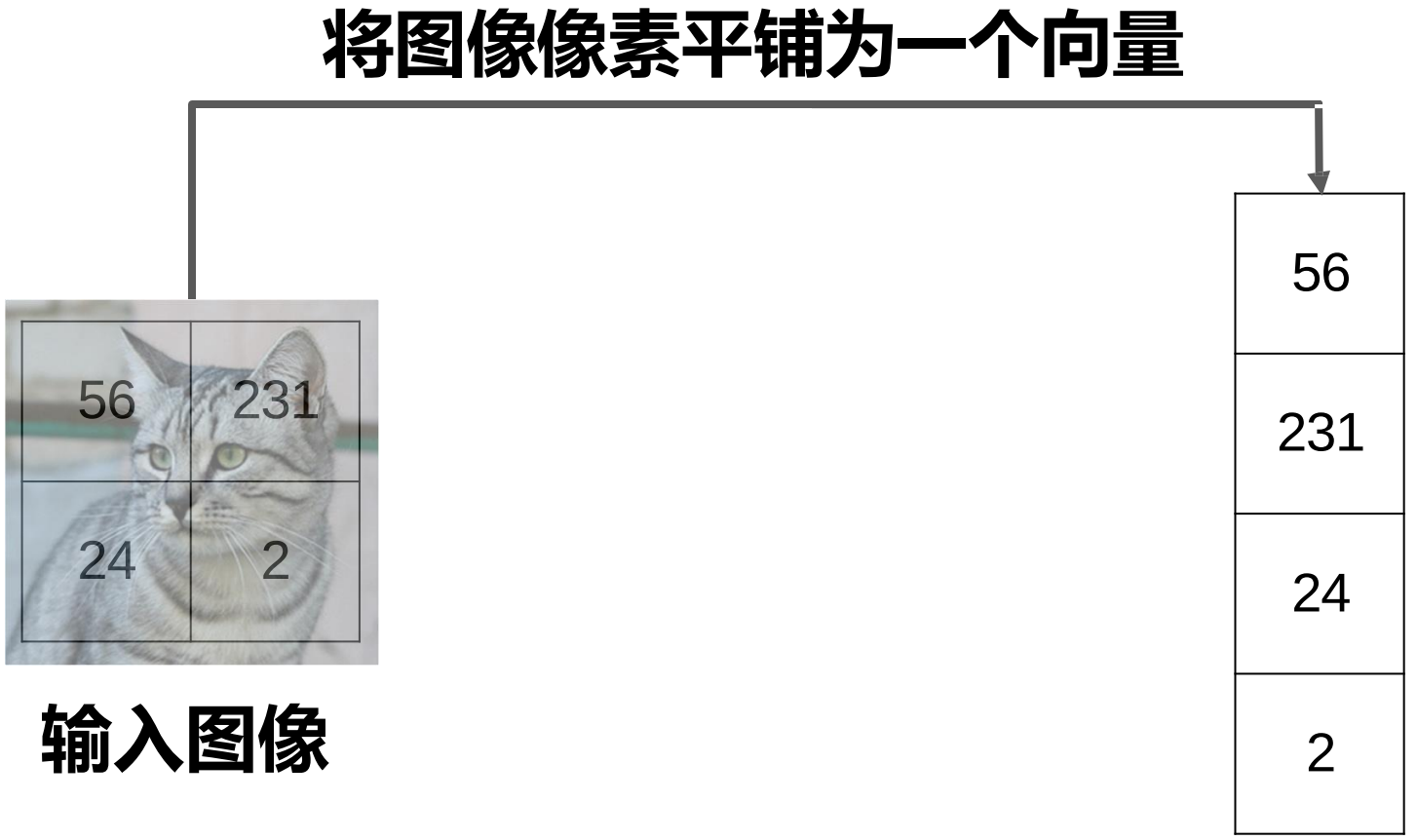
线性分类器与神经网络



线性分类器

线性分类器

- 例子：含有 2 x 2 像素的图像，数据集包含 3 个类别

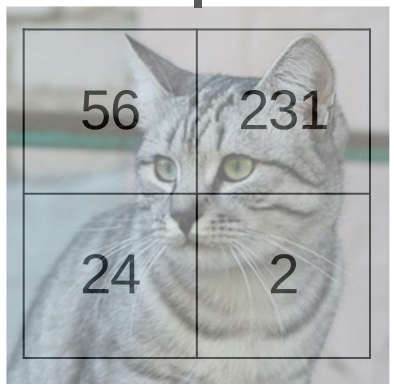


线性分类器

- 例子：含有 2 x 2 像素的图像，数据集包含 3 个类别

将图像像素平铺为一个向量

代数视角理解线性分类器



输入图像

0.2	-0.5	0.1	2.0
1.5	1.3	2.1	0.0
0	0.25	0.2	-0.3

W

56
231
24
2

+

1.1
3.2
-1.2

b

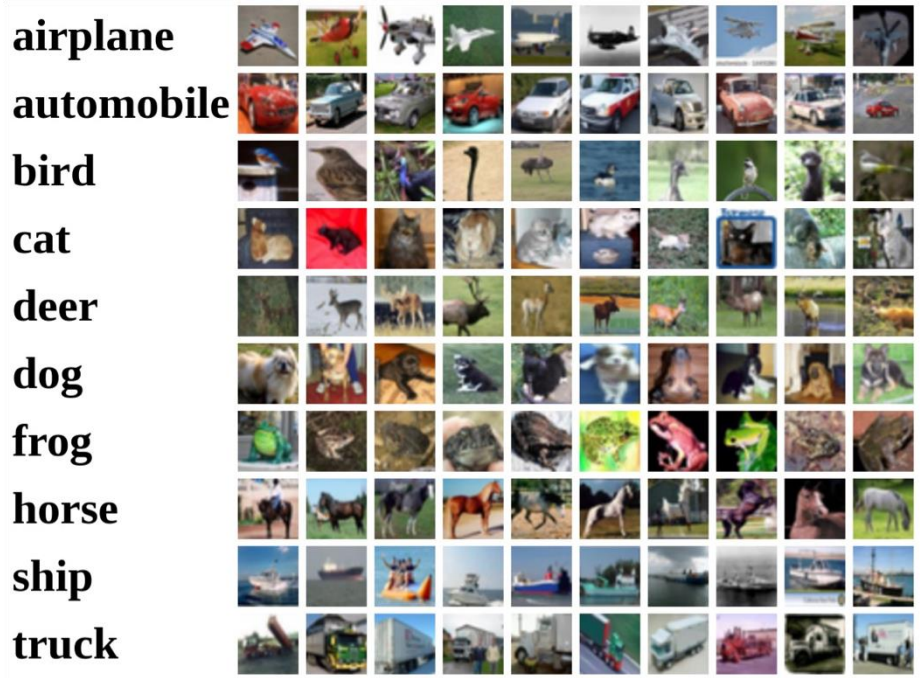
=

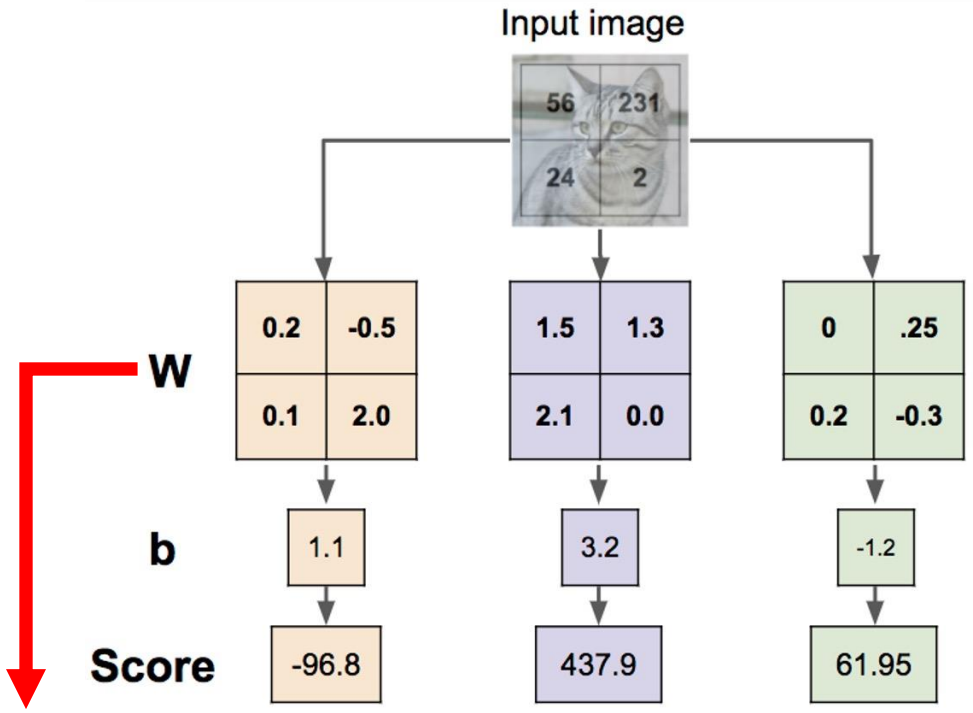
-96.8	Cat score
437.9	Dog score
61.95	Ship score

线性分类器

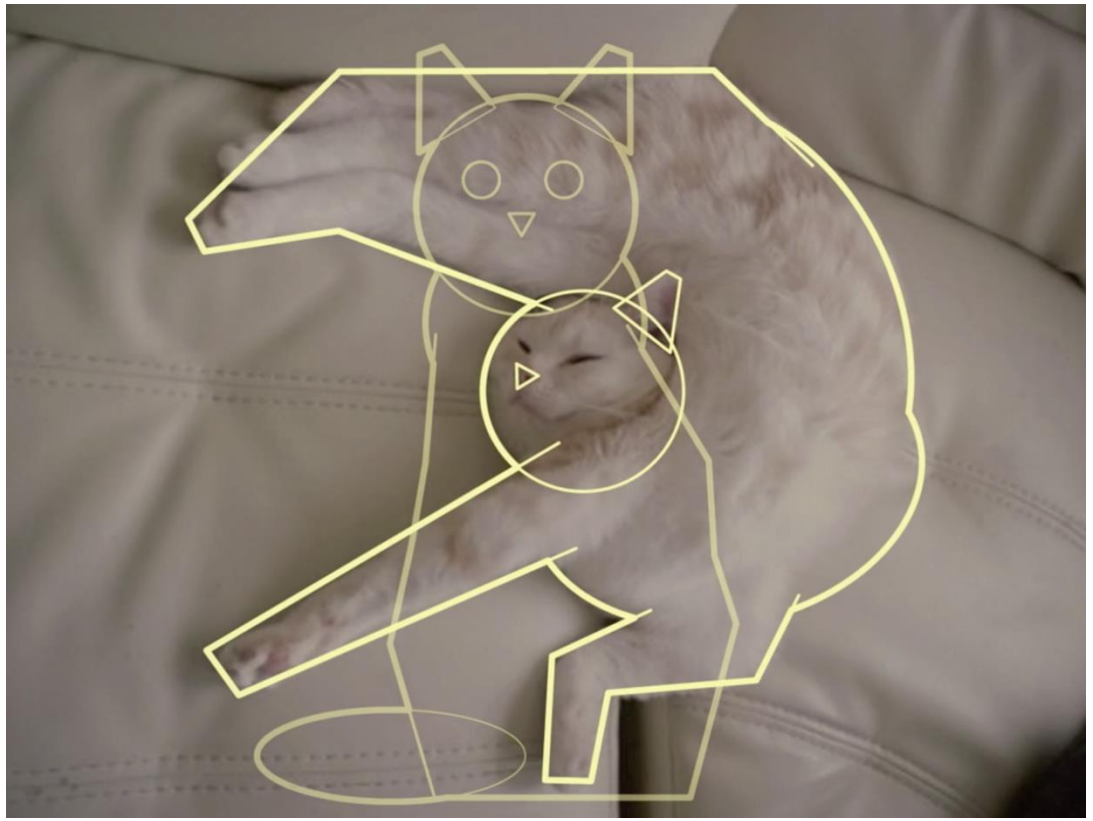
南京大学 NANJING UNIVERSITY

视觉视角理解线性分类器



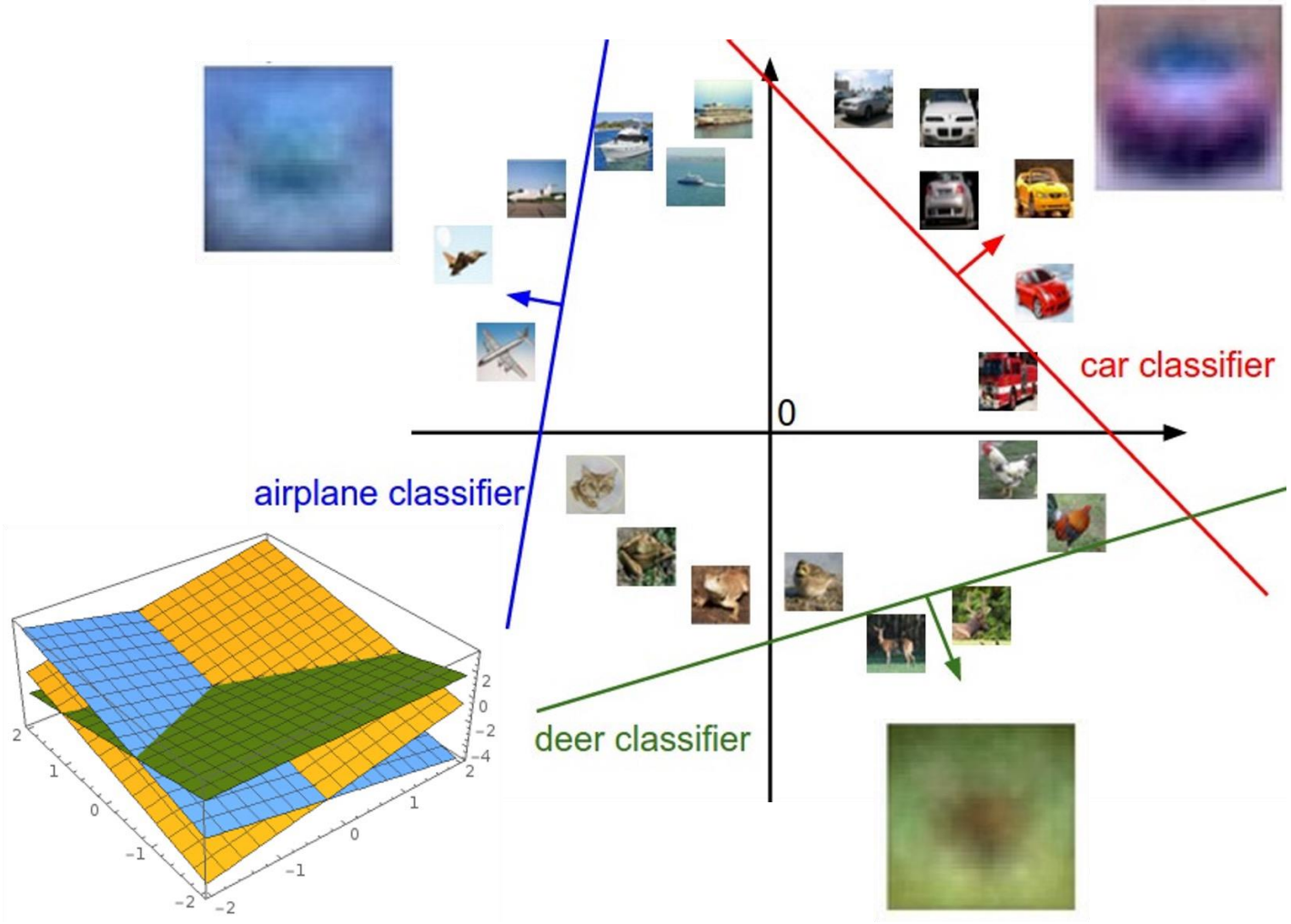


线性分类器



线性分类器

- **几何视角理解线性分类器**



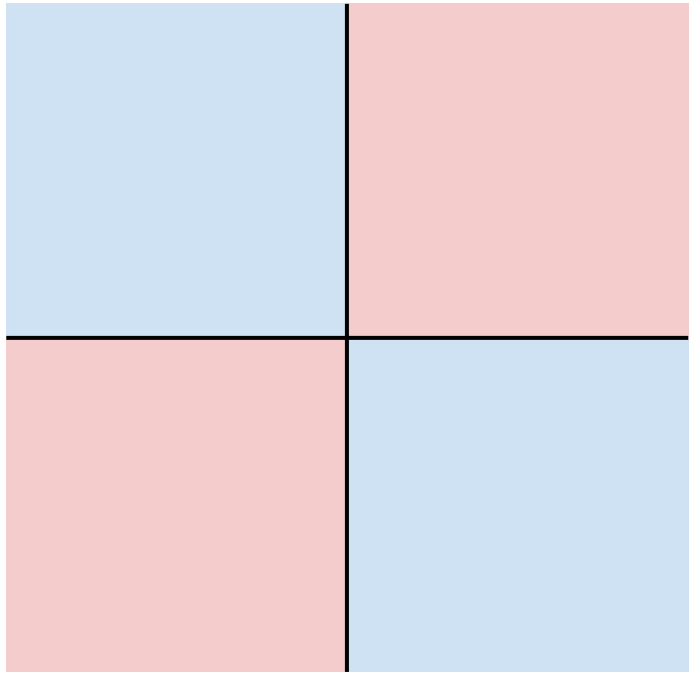
$$f(x,W) = Wx + b$$

输入图像
32x32x3 的矩阵

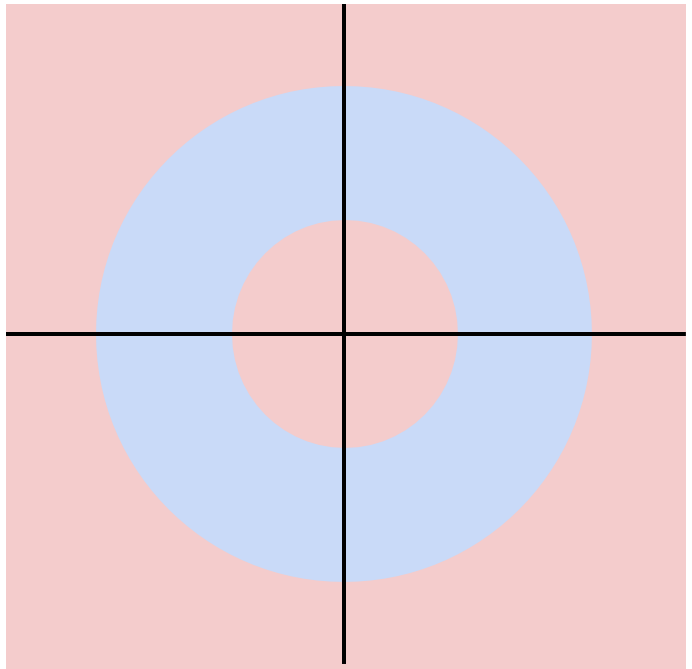
线性分类器

- 线性分类器难以处理的情况

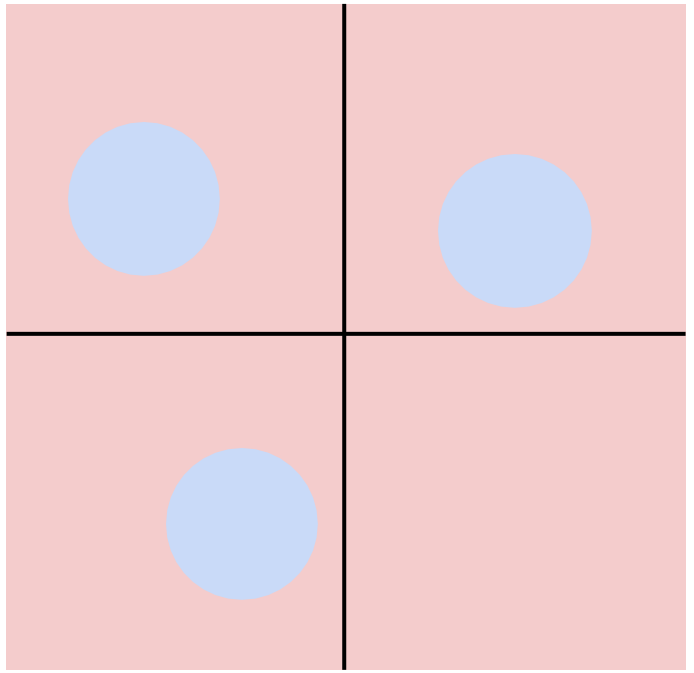
类别一：第一、三象限
类别二：第二、四象限



类别一：1<=L2 norm<=2
类别二：其他

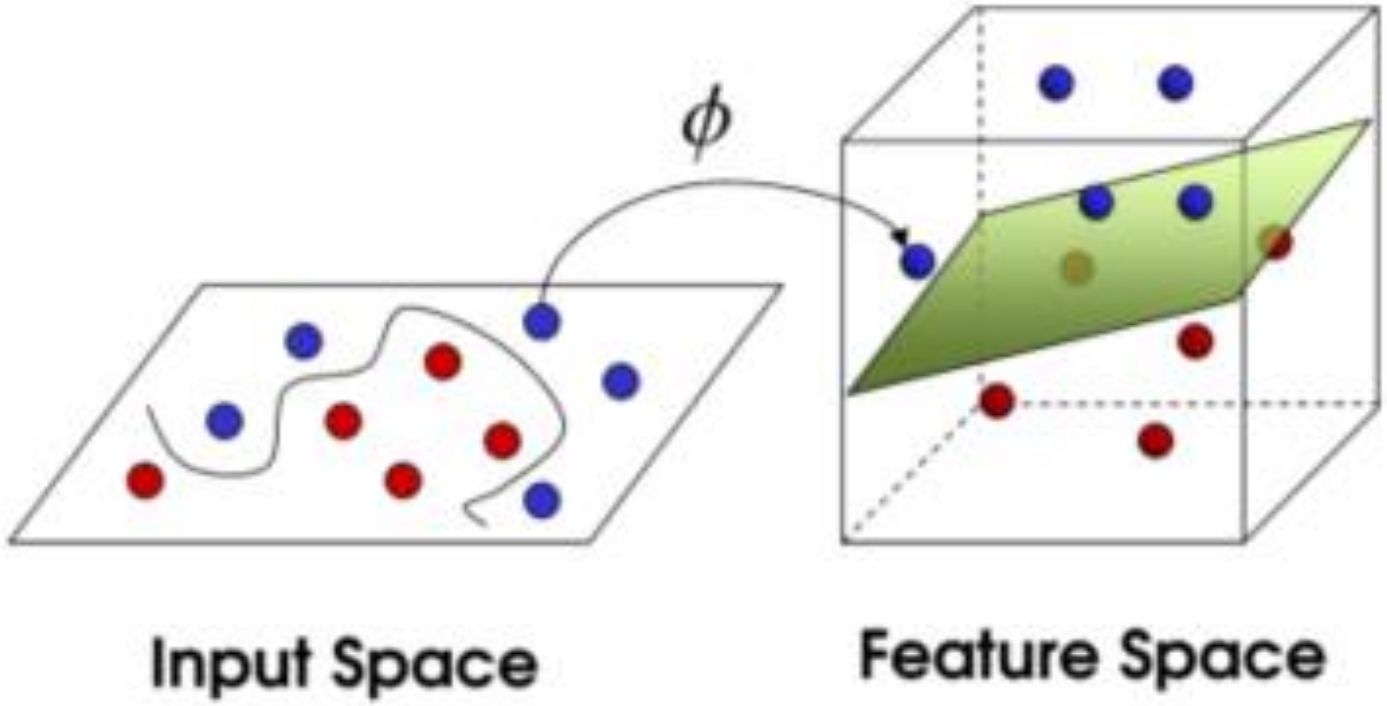


类别一：三个圆
类别二：其他



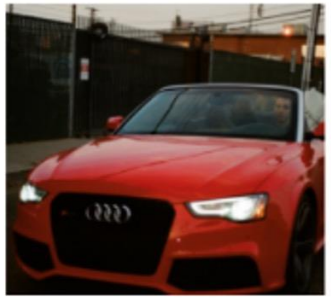
线性分类器

- 低维不可分，高维可分
- 映射到高维空间，依然可以进行线性分类



线性分类器

- 训练线性分类器的参数 W


airplane	-3.45	-0.51	3.42
automobile	-8.87	**6.04**	4.64
bird	0.09	5.31	2.65
cat	**2.9**	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	**-4.34**
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

- 定义一个损失函数，量化对训练数据中预测分数的满意度
- 提出一种有效的**优化方法找到最小化损失函数的参数**

线性分类器

- 训练线性分类器的参数 W

$$f(x, W) = Wx$$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

- 损失函数：衡量当前分类器的好坏
- 给定包含 N 个样本的数据集：

$$\{(x_i, y_i)\}_{i=1}^{N}$$

- 损失函数为：

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

线性分类器

- **多类别 SVM Loss**

$$f(x, W) = Wx$$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

- **对于一个样本：** (x_i, y_i)
- **预测得分为：** $s = f(x_i, W)$
- **SVM Loss 为：**

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

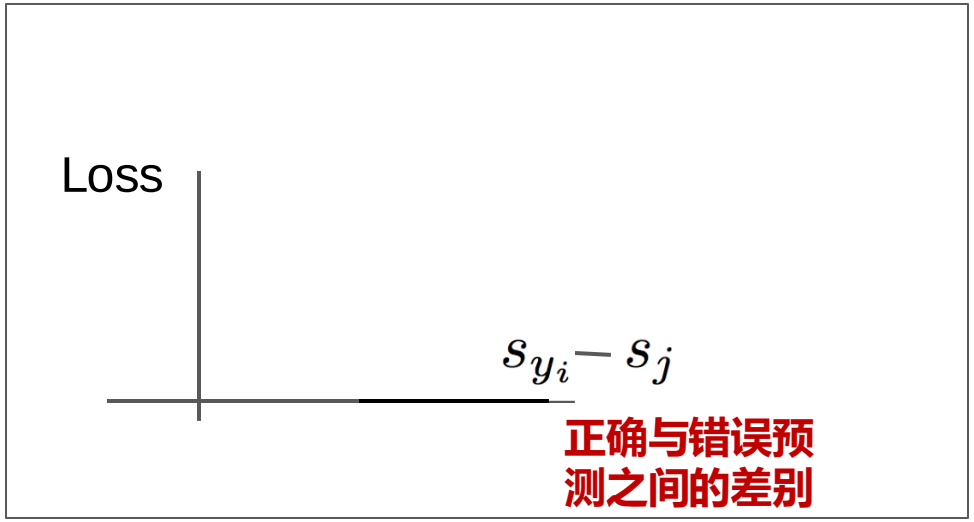
线性分类器

- **多类别 SVM Loss**

$$f(x, W) = Wx$$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1



$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

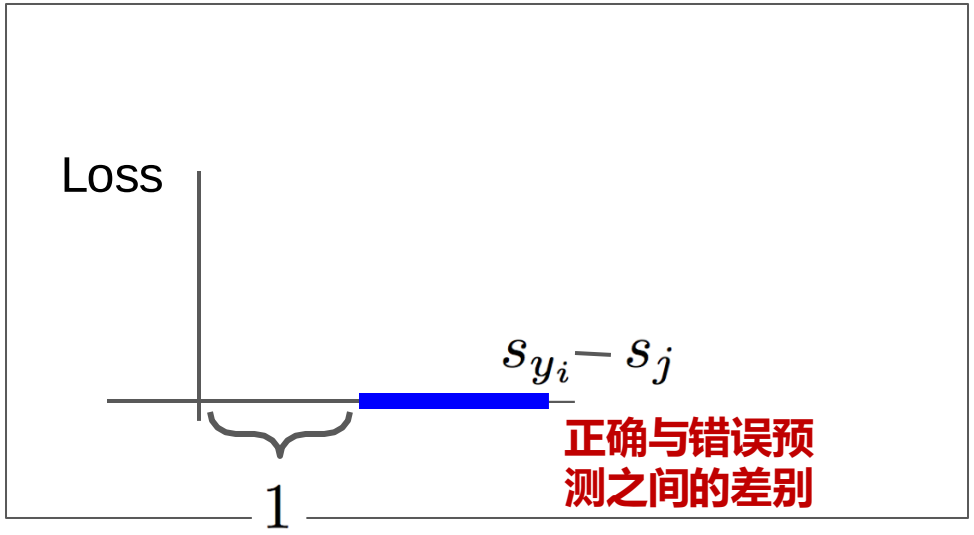
线性分类器

- **多类别 SVM Loss**

$$f(x, W) = Wx$$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1



$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

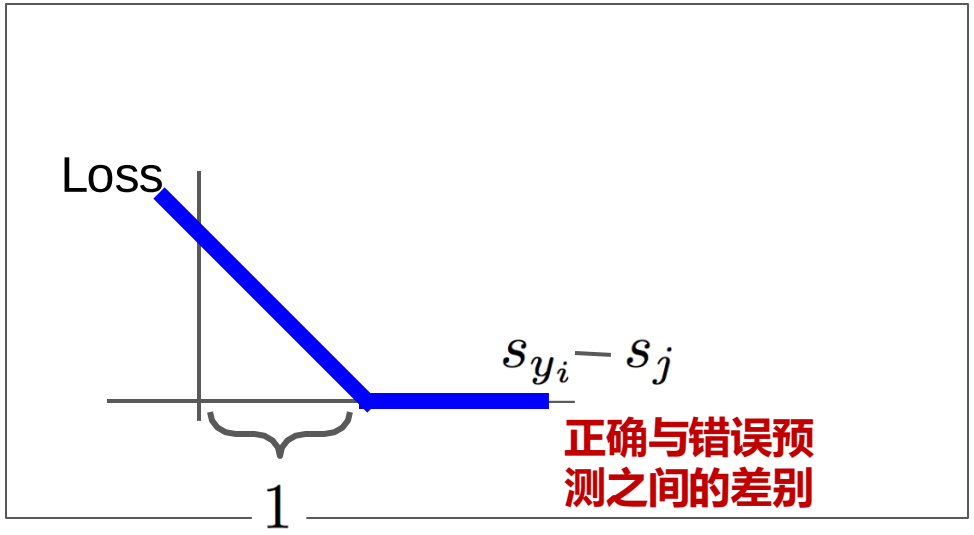
线性分类器

- **多类别 SVM Loss**

$$f(x, W) = Wx$$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1



$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**		

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

= max(0, 5.1 - 3.2 + 1)
+max(0, -1.7 - 3.2 + 1)
= max(0, 2.9) + max(0, -3.9)
= 2.9 + 0
= 2.9

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

= max(0, 1.3 - 4.9 + 1)
+max(0, 2.0 - 4.9 + 1)
= max(0, -2.6) + max(0, -1.9)
= 0 + 0
= 0

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

= max(0, 2.2 - (-3.1) + 1)
+max(0, 2.5 - (-3.1) + 1)
= max(0, 6.3) + max(0, 6.6)
= 6.3 + 6.6
= 12.9

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

最终的损失为：

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i$$

L = (2.9 + 0 + 12.9)/3
= 5.27

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$

cat	1.3
car	4.9
frog	2.0
Loss:	**0**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

- **如果 car 的预测得分降低0.5，会发生什么情况?**

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

- **损失函数的最小和最大值分别可能是多少?**

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

- **如果我们将损失函数中的 sum 替换为 mean，会出现什么情况?**

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$

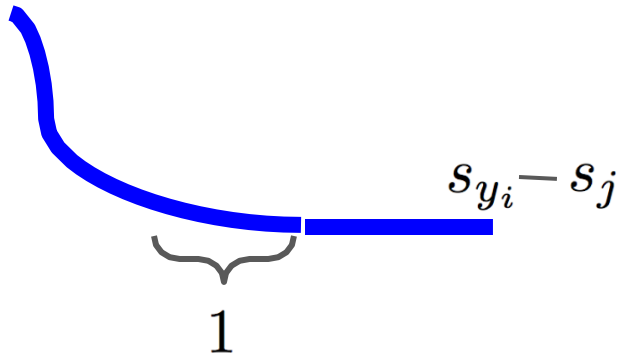


cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

- **如果我们将损失函数替换为下式?**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)^2$$



线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

- **训练初始时，W 非常小以至于 s 约为 0，那么 L_i 会是什么样的?**

线性分类器

- **多类别 SVM Loss** $f(x, W) = Wx$



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Loss:	**2.9**	**0**	**12.9**

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

- **如果我们对所有类别求和，会怎样？（包括 $j = y_i$ 的情况）**

线性分类器

南京大学 NANJING UNIVERSITY

- 多类别 SVM Loss 代码样例

```
def L_i_vectorized(x, y, W):
  scores = W.dot(x)
  margins = np.maximum(0, scores - scores[y] + 1)
  margins[y] = 0
  loss_i = np.sum(margins)
  return loss_i
```

线性分类器

南京大学 NANJING UNIVERSITY

- **线性分类器 Softmax Loss**

cat	3.2
car	5.1
frog	-1.7

线性分类器

南京大学 NANJING UNIVERSITY

- **线性分类器 Softmax Loss**

$$s = f(x_i; W)$$

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

cat 3.2

car 5.1

frog -1.7

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

概率必须 >=0

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

cat	3.2	exp	24.5
car	5.1		164.0
frog	-1.7		0.18

未归一化的 log-probabilities / logits

未归一化的 probabilities

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

概率必须
>=0

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax
函数

概率之和
必须=1

	未归一化的 log-probabilities / logits	exp	未归一化的 probabilities	normalize	概率
cat	3.2	→	24.5	→	0.13
car	5.1		164.0		0.87
frog	-1.7		0.18		0.00

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

概率必须 >=0

概率之和 必须=1

$$L_i = -\log P(Y = y_i|X = x_i)$$

	未归一化的 log-probabilities / logits	exp	未归一化的 probabilities	normalize	概率	
cat	3.2	→	24.5	→	0.13	→ L_i = -log(0.13) = 2.04
car	5.1		164.0		0.87	
frog	-1.7		0.18		0.00	

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

概率必须 >=0

概率之和 必须=1

$$L_i = -\log P(Y = y_i|X = x_i)$$

	3.2	exp	24.5	normalize	0.13	compare	1.00
cat							
car	5.1		164.0		0.87		0.00
frog	-1.7		0.18		0.00		0.00

未归一化的 log-probabilities / logits

未归一化的 probabilities

概率

正确标签的概率

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

概率必须 >=0

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

概率之和 必须=1

$$L_i = -\log P(Y = y_i|X = x_i)$$

	未归一化的 log-probabilities / logits		未归一化的 probabilities		概率		正确标签的概率
cat	3.2	exp →	24.5	normalize →	0.13	compare ←	1.00
car	5.1		164.0		0.87		0.00
frog	-1.7		0.18		0.00		0.00

Kullback–Leibler divergence

$$D_{KL}(P||Q) = \sum_y P(y) \log \frac{P(y)}{Q(y)}$$

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

概率必须 >=0

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

概率之和 必须=1

$$L_i = -\log P(Y = y_i|X = x_i)$$

	未归一化的 log-probabilities / logits		未归一化的 probabilities		概率		正确标签的概率
cat	3.2	exp →	24.5	normalize →	0.13	compare	1.00
car	5.1		164.0		0.87	Cross Entropy	0.00
frog	-1.7		0.18		0.00	$H(P,Q) = H(p) + D_{KL}(P\|Q)$	0.00

线性分类器

线性分类器 Softmax Loss

$$s = f(x_i; W)$$

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

cat 3.2

car 5.1

frog -1.7

最大化正确类别的概率：$L_i = -\log P(Y = y_i|X = x_i)$

最终的损失函数形式：$L_i = -\log(\frac{e^{sy_i}}{\sum_j e^{s_j}})$

线性分类器

南京大学 NANJING UNIVERSITY

- **线性分类器 Softmax Loss**

$$s = f(x_i; W)$$

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax 函数

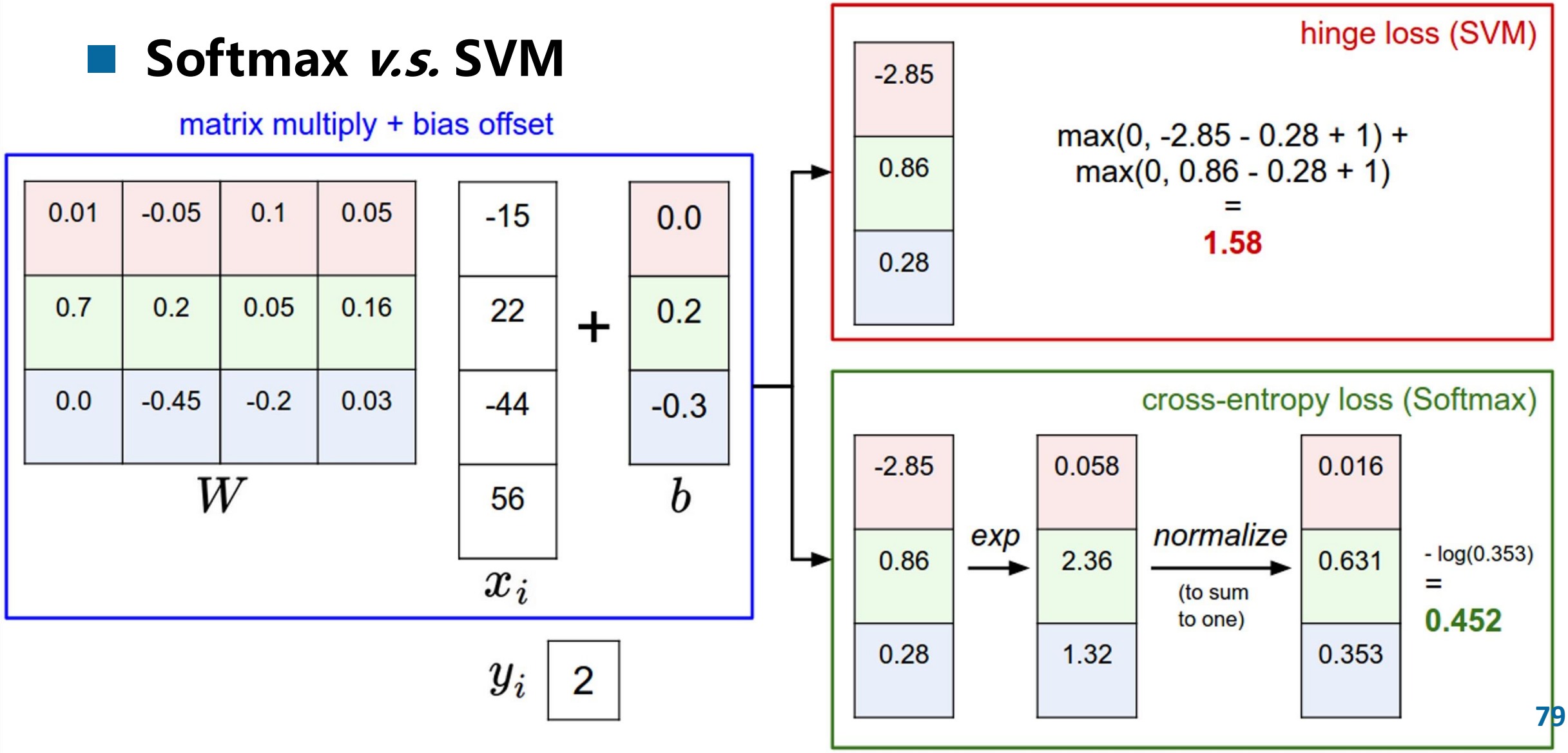
$$L_i = -\log\left(\frac{e^{sy_i}}{\sum_j e^{s_j}}\right)$$

cat	3.2
car	5.1
frog	-1.7

- **Softmax 损失 L_i 可能的最大值和最小值是多少?**
- **在初始化时，所有 s_j 将大致相等；假设有 C 个类别，softmax 损失 L_i 是多少?**
- **如果预测得分 s_j 发生轻微扰动， L_i 会发生什么?**

线性分类器

- **Softmax *v.s.* SVM**



线性分类器

- 下一节：**正则化与优化**