

图像分类的两个方法：

### 1. KNN(K-最近邻分类器)

- i. 训练：直接记住所有的训练数据和标签
- ii. 预测：对于一张新的测试图片，在训练集中找到与之最“相似”(距离最近) 的图片，并将其标签作为预测结果。
- iii. 量化：

L1距离 (曼哈顿距离)： $d_1(I_1, I_2) = \text{sum}(|I_1 - I_2|)$

## Distance Metric to compare images

L1 distance:

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

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

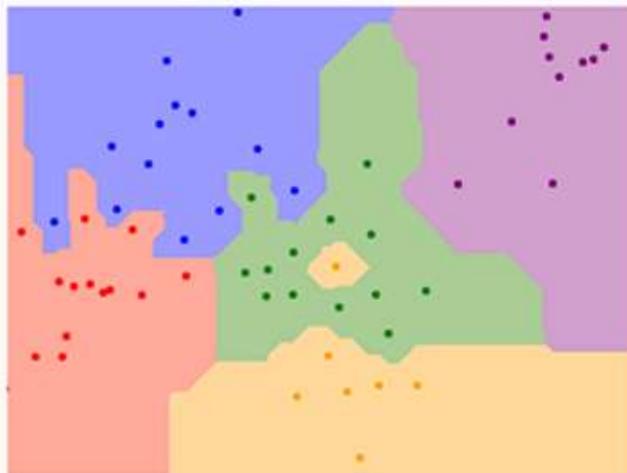
add → 456

L2距离 (欧氏距离)： $d_2(I_1, I_2) = \sqrt{\text{sum}((I_1 - I_2)^2)}$ 。两点之间的直线距离。

# K-Nearest Neighbors: Distance Metric

L1 (Manhattan) distance

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



$K = 1$

L2 (Euclidean) distance

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



$K = 1$

## iv. 超参数调优

将数据分为训练集、验证集和测试集。

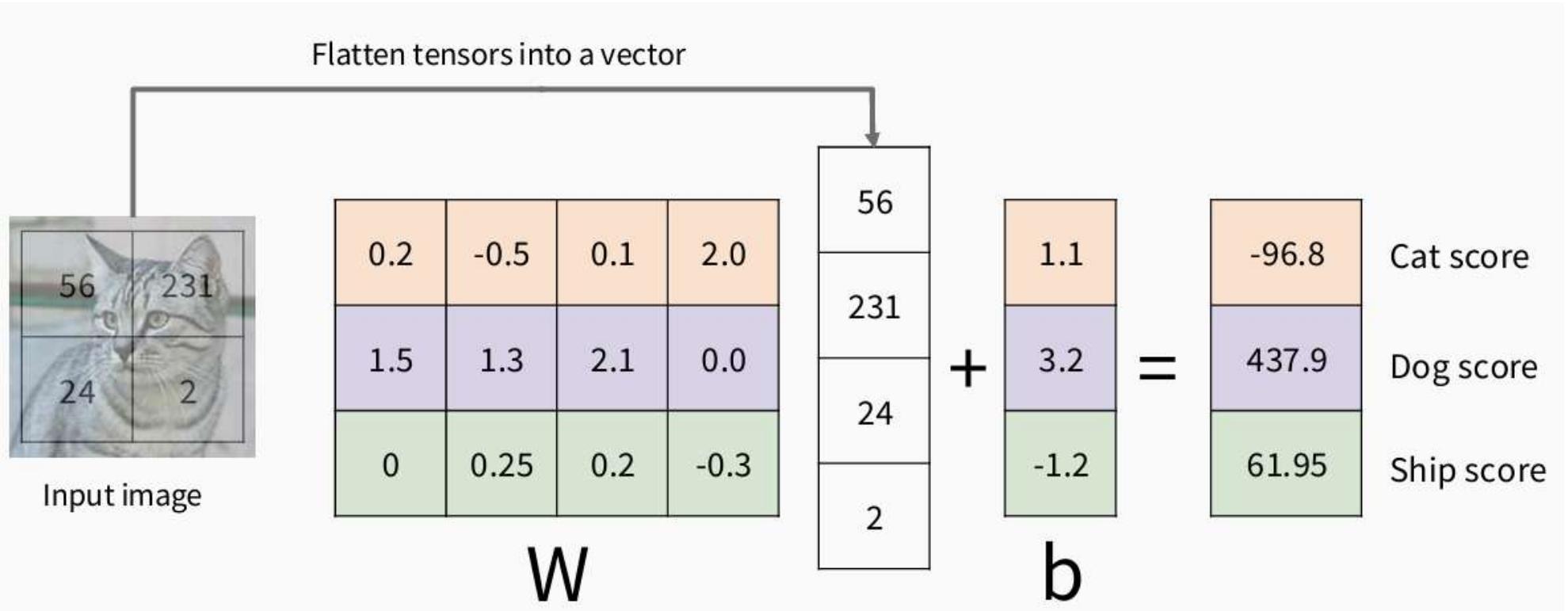
在训练集上训练不同的模型（使用不同的超参数）。

在验证集上评估这些模型，选择表现最好的超参数。

最终，在测试集上仅运行一次，以报告模型的最终性能。

## 2. Linear Classifier(线性分类器)

i. 不再存储训练数据，而是定义一个模型  $f(x, W) = Wx + b$ ，其中  $x$  是输入图像（被展平为一个长向量）， $W$  是权重矩阵， $b$  是偏置向量。



ii. 训练线性分类器的方法:

a. 多类 SVM

对单样本  $i$  (真实标签  $y_i$ )，定义 margin:

$$m_{ij} = s_j - s_{y_i} + \Delta, \quad \Delta \text{ 常取 } 1$$

损失:

$$L_i = \sum_{j \neq y_i} \max(0, m_{ij})$$

总体训练损失 (加 L2 正则化):

$$L = \frac{1}{N} \sum_{i=1}^N L_i + \lambda \frac{1}{2} \|W\|_F^2$$

SVM 关注得分差，常用于最大化类别间的 margin。

b. Softmax + 交叉熵 (Multinomial Logistic Regression)

先把 logits 变为类别概率：

$$p_j = \frac{e^{s_j}}{\sum_k e^{s_k}}$$

单样本交叉熵损失：

$$L_i = -\log p_{y_i}$$

总体损失 (含 L2)：

$$L = \frac{1}{N} \sum_{i=1}^N -\log p_{y_i} + \lambda \frac{1}{2} \|W\|_F^2$$

Softmax 输出概率、对数似然优化，常与概率解释结合使用。

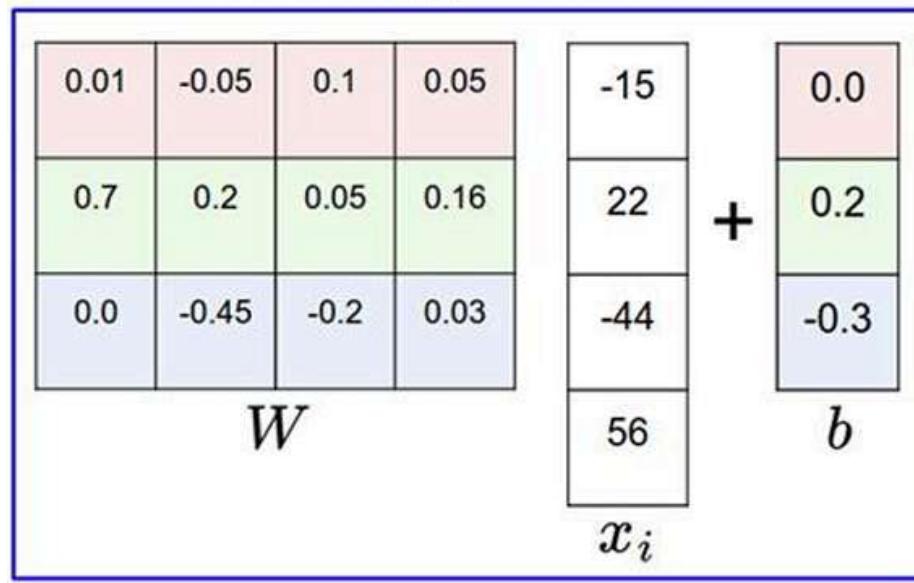
c. Softmax与SVM方法的对比

对于矩阵计算出来的分值，SVM分类器将它们看做是分类评分，它的损失函数鼓励正确的分类的分值比其他分类的分值高出至少一个边界值。

Softmax分类器将这些数值看做是每个分类没有归一化的对数概率，鼓励正确分类的归一化的对数概率变高，其余的变低。

# Softmax vs. SVM

matrix multiply + bias offset



$$y_i \quad 2$$

