

- filter
 - 模板匹配 (Template Matching)
 - 图像增强 (Image Enhancement)
 - Averaging Filtering
 - 高斯滤波 (Gaussian Filtering)
 - 特征提取 (Feature Extraction)
 - Sobel 检测器
 - Canny 边缘检测器
- Texture (纹理)
 - 纹理由重复的局部视觉模式组成
 - 具体步骤
 - 拉普拉斯算子
 - 拉普拉斯算子如何工作
 - 实际应用中的拉普拉斯算子
 - 滤波器组 (Filter Bank)
 - 常见的滤波器组
- corner
 - Harris Detector 角检测法
 - 基本原理
 - Harris角点检测的步骤
 - Harris角点检测的优点和缺点
- SIFT
 - Automatic Scale Selection
 - Scale-Normalized LoG for Multiscale Blob Detection
 - SIFT: Scale-invariant feature transform
 - SIFT for Matching
- Lecture 07
 - Bag-of-Word for Image Retrieval
 - Visual Dictionary Construction
- Linear Classifier
 - 线性分类器的定义
 - 线性分类器的种类
 - 损失函数
 - Softmax Loss (交叉熵损失)
 - Softmax函数
 - 交叉熵损失
 - SVM Loss (支持向量机损失)

- 铰链损失
- 区别与比较
 - 输出范围
 - 目标函数
 - 优化目标
- Optimization
 - 优化方法
 - 基于梯度的方法
 - 启发式方法
- Neural Nets
- CNN

filter

模板匹配（Template Matching）

模板匹配的基本原理是滑动窗口：将模板图像在待匹配图像上滑动，并在每个位置计算模板与图像区域之间的相似度。最常见的方法是通过计算相似度度量（如相关性系数、平方差等）来评估模板与图像区域的匹配程度。

图像增强（Image Enhancement）

- 直方图均衡化 通过调整图像的灰度直方图，使其灰度值分布更加均匀，从而增强图像的对比度
- 平滑处理（Smoothing） 使用滤波器（如均值滤波、高斯滤波、中值滤波）
- 锐化处理（Sharpening） 使用锐化滤波器（如拉普拉斯滤波、Unsharp Masking）来增强图像的边缘和细节
- 频域增强 通过傅里叶变换，将图像从空间域转换到频域进行处理（如滤波），然后再转换回空间域。

Averaging Filtering

通过计算图像局部区域内像素值的平均值来平滑图像。该滤波器通过卷积操作对图像进行处理，使用一个固定大小的滤波器窗口（也称为卷积核）在图像上滑动，对窗口内所

有像素的值进行平均，然后将结果赋值给中心像素

给定一个 $n \times n$ 的滤波器窗口，窗口内所有像素的值用 I 表示，则滤波后的像素值 I' 可以表示为：

$$I'(x, y) = \frac{1}{n^2} \sum_{i=-k}^k \sum_{j=-k}^k I(x + i, y + j)$$

其中， $k = \frac{n-1}{2}$ 。

高斯滤波（Gaussian Filtering）

高斯滤波是一种基于高斯分布函数的平滑滤波器，通过给邻域内的像素赋予不同的权重来计算平均值。高斯滤波器对中心像素赋予更大的权重，对远离中心的像素赋予较小的权重，从而更有效地保留图像的边缘信息。通过提取特征，可以将高维度的数据转化为低维度的特征向量，从而降低计算复杂度，提高算法的性能和准确性

给定一个大小为 $n \times n$ 的高斯滤波器窗口，高斯权重函数 $G(x, y)$ 可以表示为：

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

其中， σ 是标准差，决定了高斯分布的宽度。

高斯滤波后的像素值 I' 可以表示为：

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k G(i, j) \cdot I(x + i, y + j)$$

其中， $k = \frac{n-1}{2}$ 。

特征提取（Feature Extraction）

从原始数据中提取出有用特征，以便于进行进一步分析和处理的过程。

Sobel 检测器

Sobel 检测器是基于一阶导数的边缘检测方法，它使用 Sobel 算子（或 Sobel 滤波器）进行卷积操作，得到图像的梯度值

-1	-2	-1
0	0	0
1	2	1

Vertical
Change

-1	0	1
-2	0	2
-1	0	1

Horizontal
Change

计算步骤

1. 灰度化处理：如果输入图像是彩色图像，需要先将其转换为灰度图像。
2. 卷积操作：使用水平和垂直方向的Sobel卷积核分别对灰度图像进行卷积，得到两个梯度图像
3. 计算梯度幅值：结合水平和垂直梯度，计算每个像素的梯度幅值。梯度幅值（平方加根）
4. 非极大值抑制（可选）：进一步处理梯度图像，抑制非边缘上的梯度值，保留边缘处的梯度峰值。
5. 阈值处理（可选）：应用双阈值法或其他方法，提取出明确的边缘

Canny 边缘检测器

1. Filter image with derivatives of Gaussian in horizontal and vertical directions

用高斯导数在水平方向和垂直方向上过滤图像 解释：首先，图像通过高斯滤波器进行平滑处理，以减少噪声的影响。然后，计算图像在水平方向（x方向）和垂直方向（y方向）的导数。这个步骤的目的是检测图像中的变化，导数的计算可以突出边缘。

2. Find magnitude and orientation of gradient

翻译：找到梯度的大小和方向 解释：对于每个像素，计算梯度的大小（magnitude）和方向（orientation）。梯度的大小表示像素值变化的速率，方向表示变化的方向。

梯度的大小可以通过下式计算：

$$\text{Magnitude} = \sqrt{(G_x)^2 + (G_y)^2}$$

的方向可以通过下式计算：

$$\text{Orientation} = \arctan \left(\frac{G_y}{G_x} \right)$$

¹ G_x 和 G_y 分别是水平方向和垂直方向的导数。

3. Non-maximum suppression

翻译：非极大值抑制 解释：这一步骤的目的是细化边缘。对于每一个像素点，检查其梯度方向上是否是局部最大值。如果不是，则抑制（去除）该像素点。这样可以得到更精确的边缘线。

4. Thresholding and linking

翻译：阈值化和链接 解释：应用双阈值方法进行边缘检测。使用高阈值和低阈值区分强边缘和弱边缘。强边缘是指梯度值大于高阈值的边缘，弱边缘是指梯度值在低阈值和高阈值之间的边缘。然后，通过边缘连接算法，将弱边缘连接到强边缘，形成最终的边缘图像。

Texture （纹理）

纹理表示是指在数字图像处理和计算机图形学中，用于描述物体表面特征和外观的方法和技术

纹理由重复的局部视觉模式组成

1. 找到模式（Find the patterns）：

- 使用看起来像模式的滤波器（**Use filters that look like patterns**）：
 - 使用类似于斑点、条纹或原始补丁的滤波器来识别图像中的模式。
- 考虑响应的幅度（**Consider magnitude of response**）：
 - 关注滤波器在图像中不同位置的响应强度，以确定这些位置是否包含纹理模式。

2. 在每个局部窗口中描述它们的统计特性（Describe their statistics within each local window）：

- 统计度量（**Statistical measure**）：
 - 均值（**Mean**）：局部窗口内纹理特征值的平均值。
 - 标准差（**Standard deviation**）：局部窗口内纹理特征值的标准差，表示纹理变化的程度。
- “原型”特征出现的直方图（**Histogram of “prototypical” feature occurrences**）：

- 计算局部窗口内“原型”特征（如滤波器响应）的出现频率，并绘制直方图。这些原型特征是代表性强的局部模式，如特定方向的条纹或特定大小的斑点。

具体步骤

- 应用滤波器：**在图像上应用设计好的滤波器，这些滤波器可以是简单的卷积核，用来检测特定的模式（如Gabor滤波器检测条纹，LoG滤波器检测斑点）。
- 计算响应：**对于图像的每个局部窗口，计算滤波器的响应，通常通过计算滤波器与图像局部区域的卷积得到。
- 统计分析：**在每个局部窗口内，计算响应的统计特性，如均值和标准差。这些统计特性描述了该窗口内纹理的整体性质。
- 直方图表示：**对“原型”特征进行直方图统计，分析其出现频率，帮助理解整个图像中的纹理分布。

拉普拉斯算子（Laplacian Detector）是一种用于图像处理的边缘检测算子，基于图像的二阶偏导数。以下是对拉普拉斯算子的详细解释：

拉普拉斯算子

- 定义：**拉普拉斯算子是一个二阶微分算子，应用于图像中时，它计算的是图像亮度的二阶导数。
- 公式：**在二维空间中，拉普拉斯算子表示为 $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ 其中， I 是图像的亮度函数， (x) 和 (y) 是图像的空间坐标。

拉普拉斯算子如何工作

1. 计算图像的二阶偏导数：

- 计算图像在 (x) 和 (y) 方向上的二阶偏导数，分别记为 $\frac{\partial^2 I}{\partial x^2}$ 和 $\frac{\partial^2 I}{\partial y^2}$ 。
- 将这两个二阶偏导数相加得到拉普拉斯算子： $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$

2. 检测边缘：

- 拉普拉斯算子可以有效地检测图像中的边缘，因为在边缘处，图像亮度会发生显著变化，导致二阶导数值较大。
- 通过查找拉普拉斯算子结果中的零交叉点（zero-crossings），可以确定图像中的边缘位置。

实际应用中的拉普拉斯算子

- 离散拉普拉斯算子：在实际应用中，图像是离散的，常用离散拉普拉斯算子
- 边缘检测：拉普拉斯算子常与高斯平滑（Gaussian Smoothing）结合使用，以减少噪声对边缘检测的影响，这种组合称为拉普拉斯-高斯（Laplacian of Gaussian, LoG）滤波器。

滤波器组（Filter Bank）

计算机视觉应用：

- 特征提取：滤波器组用于从图像中提取特征，如边缘、角点、纹理等，这些特征可以用于图像分类、目标检测等任务。
- 纹理合成与识别：通过滤波器组分析图像纹理，实现纹理的合成和识别。

常见的滤波器组

1. Gradient detector:

- 特征：包含一组用于计算图像梯度的滤波器，主要目标是检测图像中灰度变化最大的地方，即边缘。
- 用例：Sobel算子，Prewitt算子

2. 小波滤波器组：

- 特征：使用小波函数对信号进行多尺度分解，适用于时间-频率分析。
- 应用：信号压缩、去噪、特征提取等。

3. Laplacian of Gaussian（LoG）滤波器组：

- 特征：结合高斯平滑和拉普拉斯算子，用于边缘检测。
- 应用：边缘检测和特征提取。

4. Steerable滤波器组:

- 特征: 滤波器的方向响应可以通过线性组合获得, 适用于方向选择性特征提取。
- 应用: 边缘检测和方向纹理分析。

corner

角点 (Corner) 是一种关键特征, 通常位于图像中亮度发生剧烈变化的地方

Harris Detector 角检测法

基本原理

Harris角点检测器的基本思想是通过计算每个像素点的自相关矩阵, 分析图像灰度值在小范围内的变化情况, 从而确定该点是否为角点。

- 自相关矩阵

自相关矩阵是图像灰度值在某个像素点附近的变化情况的描述, 定义如下:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

其中, I_x 和 I_y 分别是图像在x方向和y方向的梯度, \sum 表示在局部窗口内求和。

- 角点响应函数 Harris角点检测器通过计算角点响应函数 (Harris Response) 来判断一个点是否为角点。角点响应函数定义如下:

$$R = \det(M) - k \cdot (\text{trace}(M))^2$$

其中, $\det(M)$ 是矩阵的行列式, $\text{trace}(M)$ 是矩阵的迹 (对角线元素之和), (k) 是经验常数, 通常取值为0.04到0.06。

- 判断角点 根据角点响应函数 (R) 的值, 可以判断图像中的像素点类型:

- 当 (R) 较大时，说明该点是角点。
- 当 (R) 较小时，说明该点是平坦区域。
- 当 (R) 为负时，说明该点是边缘。

Harris角点检测的步骤

1. 计算图像梯度：使用Sobel算子计算图像在x方向和y方向的梯度 (I_x) 和 (I_y) 。
2. 构建自相关矩阵：在每个像素点处计算自相关矩阵 (M) 。
3. 计算角点响应函数：计算每个像素点的角点响应值 (R) 。
4. 非极大值抑制：对角点响应图进行非极大值抑制，提取局部极大值点作为角点。

Harris角点检测的优点和缺点

优点：

- 计算效率高：由于使用了简单的矩阵运算，Harris角点检测器计算速度较快。
- 鲁棒性强：对图像旋转和光照变化具有较好的鲁棒性。

缺点：

- 对尺度变化不敏感：Harris角点检测器对尺度变化不敏感，难以检测到不同尺度的角点。
- 需要非极大值抑制：角点检测结果中可能包含大量冗余的角点，需要进行非极大值抑制。

通过Harris角点检测器，可以有效地提取图像中的角点特征，为后续的图像处理任务提供可靠的特征点。

SIFT

Automatic Scale Selection

在图像中，同一物体可能以不同的尺度出现，而且不同的尺度可能会导致特征的外观变化。自动尺度选择的目标是在不同尺度下检测到相同的特征

自动尺度选择算法通常包括以下步骤

1. 尺度空间构建：通过对图像应用一系列不同尺度的滤波器（如高斯滤波器）来构建尺度空间。
2. 特征检测：在尺度空间中，对每个尺度下的图像进行特征检测，通常使用边缘检测、角点检测等技术。
3. 尺度选择：通过比较不同尺度下检测到的特征，选择最具代表性的特征或最适合任务的特征。
4. 特征描述：对所选特征进行描述，通常采用局部描述符（如SIFT、SURF等）。
5. 特征匹配：将图像中的特征与参考图像或其他图像进行匹配，从而实现目标检测、跟踪等任务。

Scale-Normalized LoG for Multiscale Blob Detection

一种用于多尺度斑点检测的技术。斑点通常指图像中的亮点或暗点，它们在不同尺度下可能具有不同的大小。这种技术的目标是在图像中准确地检测到不同尺度下的斑点，并且能够有效地区分它们与噪声或其他图像结构的区别。

以下是Scale-Normalized LoG技术的主要步骤：

1. 尺度空间构建：首先，通过在原始图像上应用一系列不同标准差的高斯滤波器来构建尺度空间。每个高斯滤波器的标准差决定了尺度大小。
2. **Laplacian of Gaussian (LoG) 计算**：对每个尺度下的图像应用LoG滤波器，LoG滤波器是先应用高斯滤波器，然后计算其二阶导数（拉普拉斯算子）。这一步可以帮助突出图像中的斑点。
3. 尺度归一化：对于每个尺度下的LoG响应，将其归一化为相应尺度的方差。这一步是为了消除不同尺度下斑点大小的影响，使得在不同尺度下的斑点可以进行比较。
4. 斑点检测：在归一化的LoG响应中寻找局部极大值，这些极大值对应着图像中的斑点。通常会应用阈值来筛选斑点。
5. 精确定位：对检测到的斑点进行精确定位，通常采用亚像素精确度的技术来找到斑点的中心位置。

SIFT: Scale-invariant feature transform

SIFT算法的主要特点是它能够在不同尺度和旋转下稳定地检测到图像中的特征点，并且对于一些仿射变换（如平移、旋转、缩放）也具有一定的不变性。

- 1. 尺度空间极值检测：** 使用高斯金字塔构建图像的尺度空间。这一步骤是构建高斯金字塔，通过在原始图像上应用一系列不同标准差的高斯滤波器来实现的。高斯金字塔的每一层代表了图像在不同尺度下的平滑版本。
- 2. 关键点定位：** 在差分金字塔中寻找尺度空间极值点，这些极值点对应于图像中的关键点。差分金字塔是通过对高斯金字塔相邻层之间进行差分操作而得到的，其中的差分图像代表了图像中不同尺度下的边缘信息。
- 3. 方向分配：** 在每个关键点周围的邻域内，根据关键点周围的梯度方向，为关键点分配一个主导方向。这一步骤是在尺度空间极值检测之后，在关键点定位的基础上完成的。
- 4. 关键点描述：** 在每个关键点周围的邻域内，通过统计局部图像梯度方向的直方图来描述关键点的局部特征。这一步骤是在确定了关键点的方向之后进行的。
- 5. 特征匹配：** 使用描述子之间的距离来衡量特征之间的相似度，并进行特征匹配。这一步骤是在完成了关键点描述之后进行的。

所以，SIFT算法中在尺度空间极值检测和关键点定位阶段，使用的是高斯金字塔。差分金字塔则在后续的步骤中，特别是在关键点方向分配和关键点描述阶段会用到。

SIFT for Matching

SIFT (Scale-Invariant Feature Transform) 在图像匹配中是一种非常经典和有效的方法，它能够在不同尺度、旋转和光照条件下稳定地检测和描述图像中的局部特征，并且具有较高的鲁棒性。

SIFT匹配主要包括以下步骤：

- 1. 特征提取：** 在两幅图像中分别应用SIFT算法，检测并描述每幅图像中的关键点及其对应的特征描述子。这些描述子是对关键点周围局部区域的描述，通常是128维的向量，描述了关键点周围的梯度方向和强度等信息。
- 2. 特征匹配：** 对于第一幅图像中的每个特征描述子，寻找在第二幅图像中与之最相似的特征描述子。这可以通过计算描述子之间的距离（如欧氏距离、汉明距离等）来实现。一般采用最近邻匹配算法，即为每个特征描述子在另一幅图像中找到最接近的特征描述子。
- 3. 匹配筛选：** 对特征匹配结果进行筛选和优化，以去除不可靠的匹配对。常见的筛选方法包括基于距离的阈值筛选、基于比率的阈值筛选、基于几何约束的筛选等。例

如，使用阈值来过滤掉距离过大的匹配对，或者通过RANSAC算法来估计并筛选出符合几何变换模型的匹配对。

4. **匹配结果展示：** 将保留下来的匹配对可视化展示，通常以线段或箭头的形式连接两幅图像中匹配的特征点，以便于直观地观察匹配结果。

Lecture 07

Bag-of-Word for Image Retrieval

Bag-of-Words (BoW) 模型是一种经典的特征表示方法，最早用于文本分析，但也广泛应用于图像检索。它通过将图像表示为一组无序的局部特征（“单词”），从而实现图像特征提取和相似性匹配。以下是关于Bag-of-Words模型在图像检索中的详细介绍：

1. **BoW模型的基本原理** Bag-of-Words模型的核心思想是将图像表示为局部特征的集合。这些局部特征可以通过特征检测和描述算法（如SIFT、SURF等）提取，然后通过量化过程将其映射到有限数量的视觉词汇（视觉单词）上。整个过程包括以下几个步骤：
 2. **特征提取** 首先，对每幅图像进行特征提取，通常使用SIFT（尺度不变特征变换）或SURF（加速鲁棒特征）等算法。这些算法能够提取出图像中的关键点，并为每个关键点生成一个特征向量（描述子）。
 3. **特征量化** 将提取到的特征向量通过聚类算法（如K-Means聚类）进行量化。聚类算法将所有特征向量分成若干个簇，每个簇的中心称为一个视觉单词。这个视觉单词的集合构成了视觉词汇表。
 4. **构建图像的BoW表示** 对于每幅图像，将其特征向量映射到视觉词汇表上，统计每个视觉单词在该图像中出现的次数，生成一个直方图。这个直方图就是图像的BoW表示，类似于文本分析中的词频直方图。
 5. **相似性度量** 在图像检索过程中，通过比较查询图像的BoW表示与数据库中图像的BoW表示，可以计算它们之间的相似度。常用的相似性度量方法包括欧氏距离、余弦相似度等。
6. **检索过程**
 1. **离线阶段：**
 - 对数据库中的所有图像进行特征提取。
 - 进行聚类，生成视觉词汇表。

- 将每幅图像转换为BoW表示，并存储到数据库中。

2. 在线阶段：

- 对查询图像进行特征提取。
- 将特征向量量化，生成BoW表示。
- 计算查询图像与数据库中图像的相似度，并返回最相似的图像。

Visual Dictionary Construction

视觉词典构建是图像处理和计算机视觉领域中的一个重要任务，通常用于Bag-of-Words (BoW) 模型视觉表示方法中。

概述 在图像处理中，视觉词典是一个关键的概念。它类似于文本处理中的字典，但是在图像领域，它是由一组视觉单词组成的，用于描述图像的内容。构建一个有效的视觉词典可以帮助我们更好地理解并表示图像的特征。

构建过程

1. 首先，从图像中提取特征。常用的特征包括SIFT、SURF、ORB等。这些特征描述了图像中的局部外观信息，如角点、边缘等。
 2. 特征聚类将提取到的特征进行聚类，将相似的特征归为一类。常用的聚类算法包括K-Means、Mean-Shift、DBSCAN等。聚类过程中，每个聚类中心代表一个视觉单词。
 3. 词典生成.将聚类得到的聚类中心作为视觉词典的单词，构建出一个词汇表。词汇表的大小通常由聚类算法的参数决定，也可以手动设定。
 4. 有时候，为了提高词典的质量，还可以对词汇表进行优化
- 在Bag-of-Words模型中，构建一个有效的视觉词典可以帮助我们将图像转换为稠密的特征向量，从而实现图像的相似性比较和检索。

Linear Classifier

线性假设 (Linear Hypothesis) 或线性分类器 (Linear Classifier) 是机器学习和统计学中的一种基本模型。

它用于将数据点分为不同类别，基于输入特征线性组合的结果进行分类。

线性分类器的定义

线性分类器通过一个线性函数将输入空间划分为不同的区域，每个区域对应一个类别。线性函数可以表示为：

$$[y = w * x + b]$$

其中：

- (y) 是输出（可以是分类的标签或预测的值）。
- (w) 是权重向量（coefficients）。
- (x) 是输入特征向量。
- (b) 是偏置（intercept）。

对于二分类问题，线性分类器的决策边界是一个超平面，将特征空间划分为两个部分。

线性分类器的种类

几种常见的线性分类器包括：

- **感知机（Perceptron）**：感知机是最简单的线性分类器，通过迭代更新权重来找到分离超平面。
- **支持向量机（SVM）**：SVM通过最大化类间间隔（margin）来找到最优的分离超平面。
- **线性回归（Linear Regression）**：尽管主要用于回归问题，线性回归也可以用于二分类任务，方法是将输出映射到分类标签。
- **逻辑回归（Logistic Regression）**：逻辑回归通过逻辑函数将线性组合的结果映射到概率值，适用于二分类任务。

损失函数

两种常见的损失函数是Softmax损失（Softmax Loss）和支持向量机损失（SVM Loss）。

Softmax Loss（交叉熵损失）

Softmax损失，通常与交叉熵损失结合使用，是多分类问题中的一种常见损失函数。Softmax函数将模型的输出转化为概率分布，然后使用交叉熵计算预测概率与真实标签之间的差异。

Softmax函数

Softmax函数将其转化为概率分布

$$P(Y = y_i | X = \mathbf{x}_i) = \frac{\exp(s_{y_i})}{\sum_j \exp(s_{ij})}$$

交叉熵损失

交叉熵损失衡量预测概率分布 \hat{y} 和真实分布 (y) 之间的差异：

$$\text{Cross-Entropy: } H(\mathbf{p}, \mathbf{q}) = - \sum_x p(x) \log q(x)$$

其中， (y_i) 是真实标签的指示函数， (\hat{y}_i) 是通过Softmax函数计算的预测概率。

SVM Loss（支持向量机损失）

支持向量机损失通常用于SVM模型，主要用于二分类和多分类问题。其目标是最大化类间间隔（margin），同时最小化分类错误。常用的SVM损失是铰链损失（hinge loss）。

铰链损失

对于二分类问题，铰链损失定义如下：

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

其中， (y) 是真实标签（取值为+1或-1）， $(f(x))$ 是模型的预测输出。

区别与比较

输出范围

- **Softmax Loss:** 将logits转化为概率分布，输出在 $[0, 1]$ 之间，总和为1。
- **SVM Loss:** 直接使用模型输出的分数，不转化为概率。

目标函数

- **Softmax Loss:** 最小化预测概率分布与真实分布之间的交叉熵差异。
- **SVM Loss:** 最大化类间间隔，同时最小化分类错误。

优化目标

- **Softmax Loss:** 更注重概率预测的准确性，适合需要概率输出的任务。
- **SVM Loss:** 更注重分类边界的优化，适合需要硬分类的任务。

Optimization

优化问题通常可以表示为最小化一个损失函数。目标是找到一组参数 (θ) 使得 $L(\theta)$ 达到最小值。

优化方法

基于梯度的方法

a. 梯度下降法（Gradient Descent）

梯度下降法是一种简单而有效的优化算法，通过沿着损失函数梯度的反方向更新参数：

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t)$$

其中， (η) 是学习率，控制参数更新的步长。

b. 随机梯度下降法（SGD, Stochastic Gradient Descent）

随机梯度下降法在每次迭代中仅使用一个样本来计算梯度：

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t; x_i, y_i)$$

这种方法可以加快收敛速度，但更新路径较为不稳定。

c. 小批量梯度下降法（Mini-Batch Gradient Descent）

小批量梯度下降法结合了批量梯度下降和随机梯度下降的优点，每次迭代使用一小批样本计算梯度：

$$[\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t; X_B, Y_B)]$$

d. 动量法（Momentum）

动量法通过引入动量项，积累历史梯度的加权平均来加速收敛：

$$[v_t = \beta v_{t-1} + (1 - \beta) \nabla_{\theta} L(\theta_t)] [\theta_{t+1} = \theta_t - \eta v_t]$$

其中， (β) 是动量系数。

e. 自适应方法（Adaptive Methods）

- **Adagrad**: 通过调整学习率来适应每个参数的变化。
- **RMSprop**: 改进Adagrad，防止学习率过快下降。
- **Adam**: 结合动量和RMSprop的优点，使用一阶矩和二阶矩的估计。

启发式方法

这些方法不依赖梯度信息，通常用于非凸优化问题或离散优化问题：**a. 遗传算法（Genetic Algorithm） b. 模拟退火（Simulated Annealing） c. 粒子群优化（Particle Swarm Optimization）**

Neural Nets

CNN
