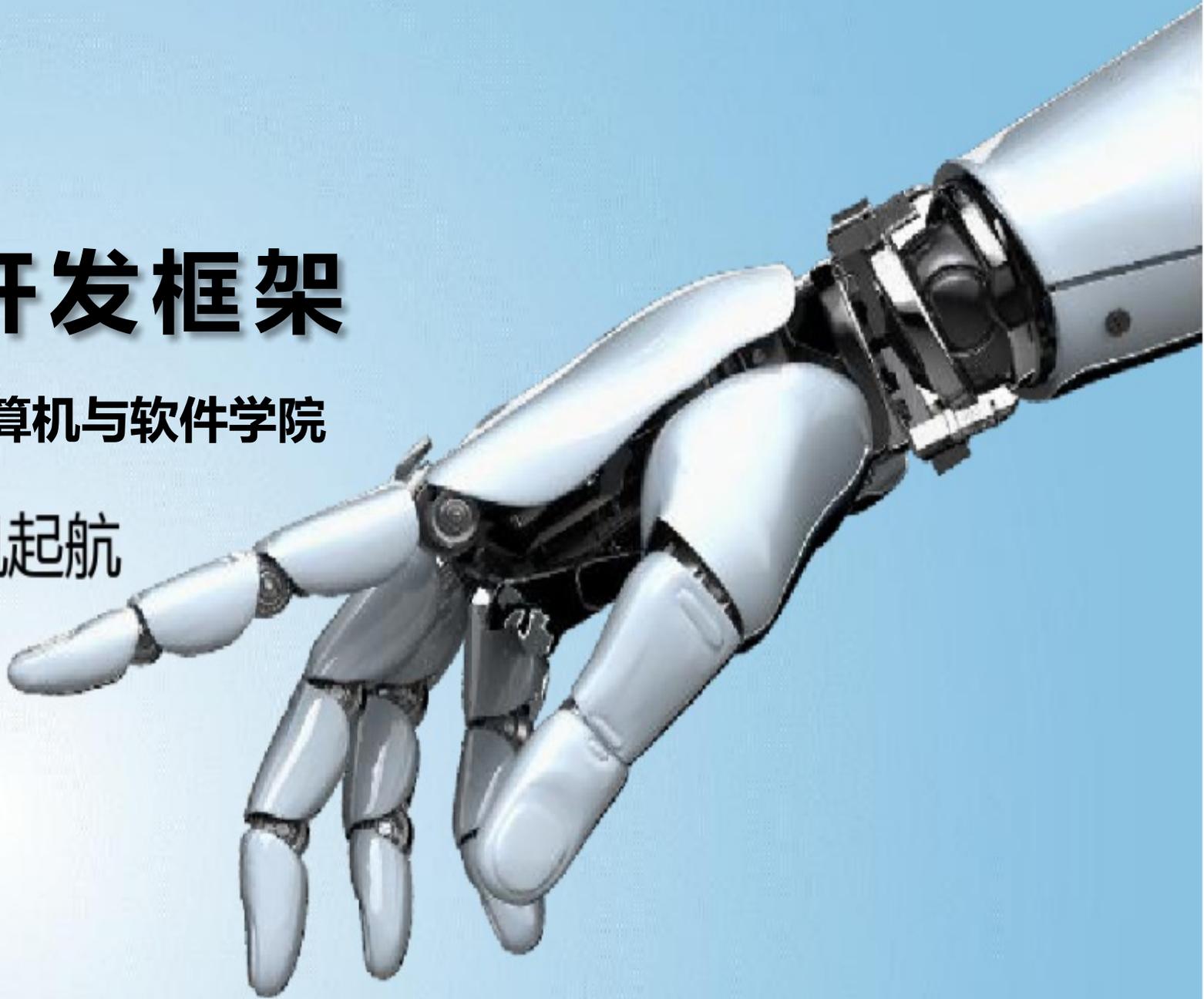
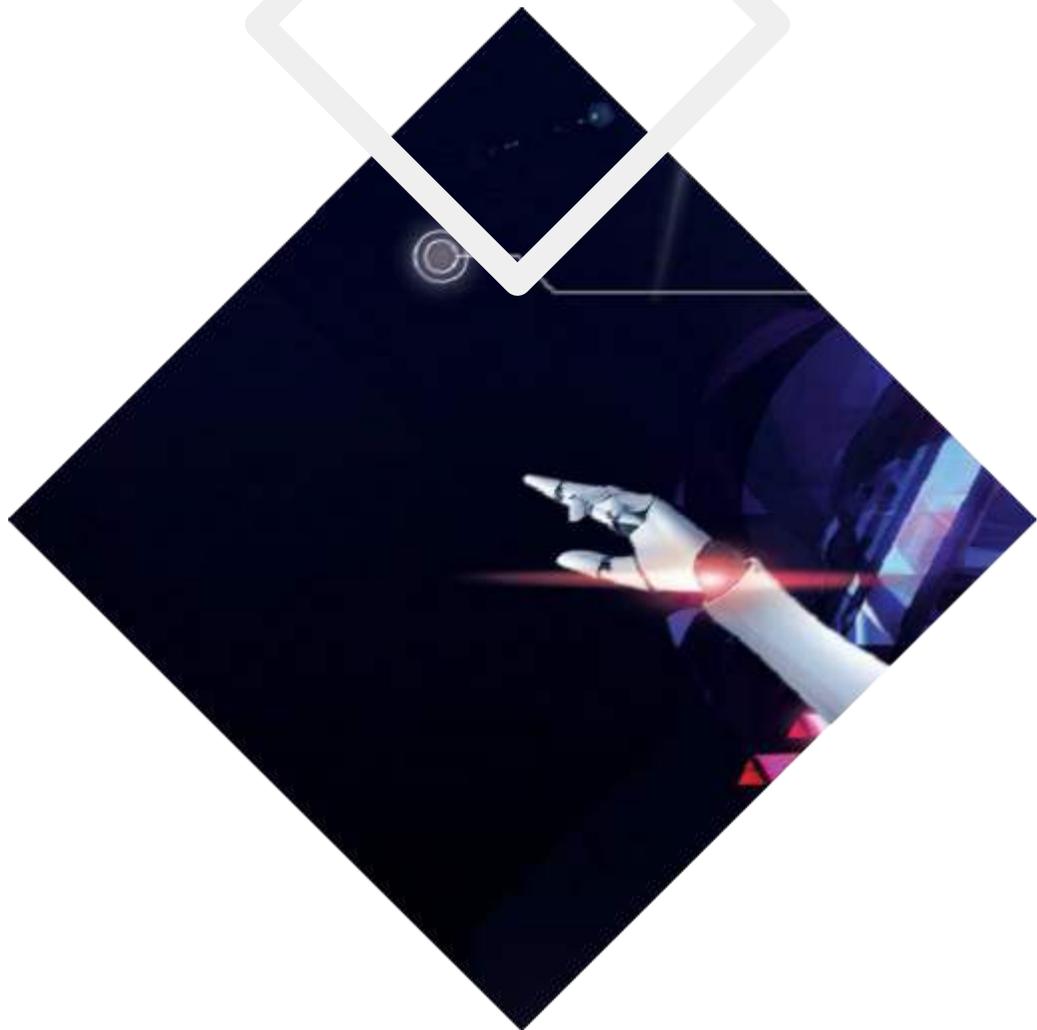


人工智能技术开发框架

计算机与软件学院

扬帆起航





01 AI应用开发概述

02 Flask WEB应用开发

03 数据处理与分析工具

04 机器学习框架Scikit-Learn

05 深度学习框架Tensorflow2

数据处理与分析

- 数据处理
- 数据可视化

数据处理工具

- Numpy
- Pandas

Numpy

Numpy 是Python科学计算的基础库，主要提供了高性能的N维数组实现以及计算能力，还提供了和其他语言如C/C++集成的能力，此外还实现了一些基础的数学算法，如线性代数相关、傅里叶变换及随机数生成等。

- 创建数组
- 查看数组维度和类型
- 数组元素的访问及索引
- Numpy的运算
- 数组的维度变换
- 数组的排序及索引
- 数组的文件操作（数据文件存储、数据文件导入）

| Numpy : 创建数组

- 通过python的list列表创建数组
- 通过Numpy的函数创建数组

| Numpy : 通过python的list列表创建数组

```
a = np.array([1 , 2 , 3 , 4])
b = np.array([[1 , 2], [3 , 4], [5 , 6]])

# 查看array 的属性 , 包括数据的维度和类型
print(b.ndim)
print(b.shape)
print(b.dtype)
```

Numpy : 通过Numpy的函数创建数组

创建等差一维数组，步长为1

```
c = np.arange(10)
```

```
print("c=",c)
```

创建等差一维数组，[0, 2]分成11等分后的数组

```
d = np.linspace(0, 2, 11)
```

```
print("d=",d)
```

创建3 × 3 的全1数组

```
e = np.ones((3, 3))
```

```
print("e=",e)
```

创建3 × 6 的全零数组

```
f = np.zeros((3, 6))
```

```
print("f=",f)
```

创建4 × 4 的对角数组

```
g = np.eye(4)
```

```
print("g=",g)
```

创建6 × 4 的随机数组

```
h = np.random.randn(6, 4)
```

```
print("h=",h)
```

Numpy : 数组元素的访问及索引——1维数组

```
a= np.arange(10)
```

```
print(a)
```

```
print(a[0] , a[3] , a[-1])
```

```
print(a[:4])
```

```
print(a[3:7])
```

```
print(a[6:])
```

```
print(a[2:8:2]) # 3 个参数表示起始、结束和步长，不包含结束位置
```

```
print(a[2::2]) # 结束位置可以省略
```

```
print(a[::3]) # 开始和结束都省略
```

Numpy : 数组元素的访问及索引——2维数组

```
a= np.arange(0 , 51 , 10).reshape(6 , 1) + np.arange(6) # 以下为对二维数组矩阵进行索引切片操作
# 一维向量 , 1×6数组矩阵
print(np.arange(0 , 51 , 10))
# 转变成6×1数组矩阵
print(np.arange(0 , 51 , 10).reshape(6 , 1))
# 一维向量 , 1×6数组矩阵
print(np.arange(6))
# 两个矩阵相加运算
print(a)
print(a[0 , 0] , a[2 , -1])
print(a[0 , 2 : 5])
print(a[ : 3, 3 : ])
print(a[2 , : ])
print(a[ : , 3]) # 结果应该是列向量 , Numpy 自动转换行向量
print(a[ : , :: 2])
print(a[ :: 2 , :: 3])
```

Numpy : Numpy的运算

- 数组和标量的运算
- 数组与数组之间的运算
- 矩阵的内积运算
- 数组的比较运算
- 数组的内置函数
- 数组的统计运算

Numpy : 数组和标量的运算

- 最简单的数值计算是数组和标量进行计算，计算过程是直接把数组里的元素和标量逐个进行计算

```
a = np.arange(6)
```

```
print(a)
```

```
print(a + 5)          # 数组和标量加法
```

```
# 创建1~5之间的20个随机整数，形成一维向量，即1×20的数组矩阵
```

```
b = np.random.randint(1, 5, 20)
```

```
print(b)
```

```
# 将1×20的数组矩阵构建出4×5的数组矩阵
```

```
c = b.reshape(4, 5)
```

```
print(c)
```

```
print(c * 3)         # 数组和标量乘法
```

Numpy : 数组与数组之间的运算

- 数组和数组的运算，如果数组的维度相同，那么在组里对应位置进行逐个元素的数学运算。

```
# 创建5×4数组矩阵，元素是1~5随机整数
a = np.random.random_integers(1, 5, (5, 4))
print(a)
# 创建一个5×4的数组矩阵，每个元素都是1
b = np.ones((5, 4), dtype=int)
print(b)
# 数组加法
a + b
print(a + b)
```

```
# 创建3×4数组矩阵，元素是1~5的随机整数
c = np.random.random_integers(1, 5, (3, 4))
print(c)
# 创建3×4的数组矩阵，元素是1~5随机整数
d = np.random.random_integers(1, 5, (3, 4))
print(d)
# 数组相乘，逐元素相乘，不是矩阵内积运算
c * d
print(c * d)
```

Numpy : 矩阵的内积运算

- 数组的乘法是对应元素相乘，不是矩阵内积，矩阵内积使用的是np.dot()函数

```
# 创建一个3×2的数组矩阵，每个元素是1~5的随机整数，random_integers函数包含末尾数5
```

```
a = np.random.random_integers(1, 5, (3, 2))
```

```
print(a)
```

```
# 创建一个2×3的数组矩阵，每个元素是1~5的随机整数
```

```
b = np.random.random_integers(1, 5, (2, 3))
```

```
print(b)
```

```
# 矩阵内积
```

```
np.dot(a, b)
```

```
print(np.dot(a, b))
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/148123116104007015>