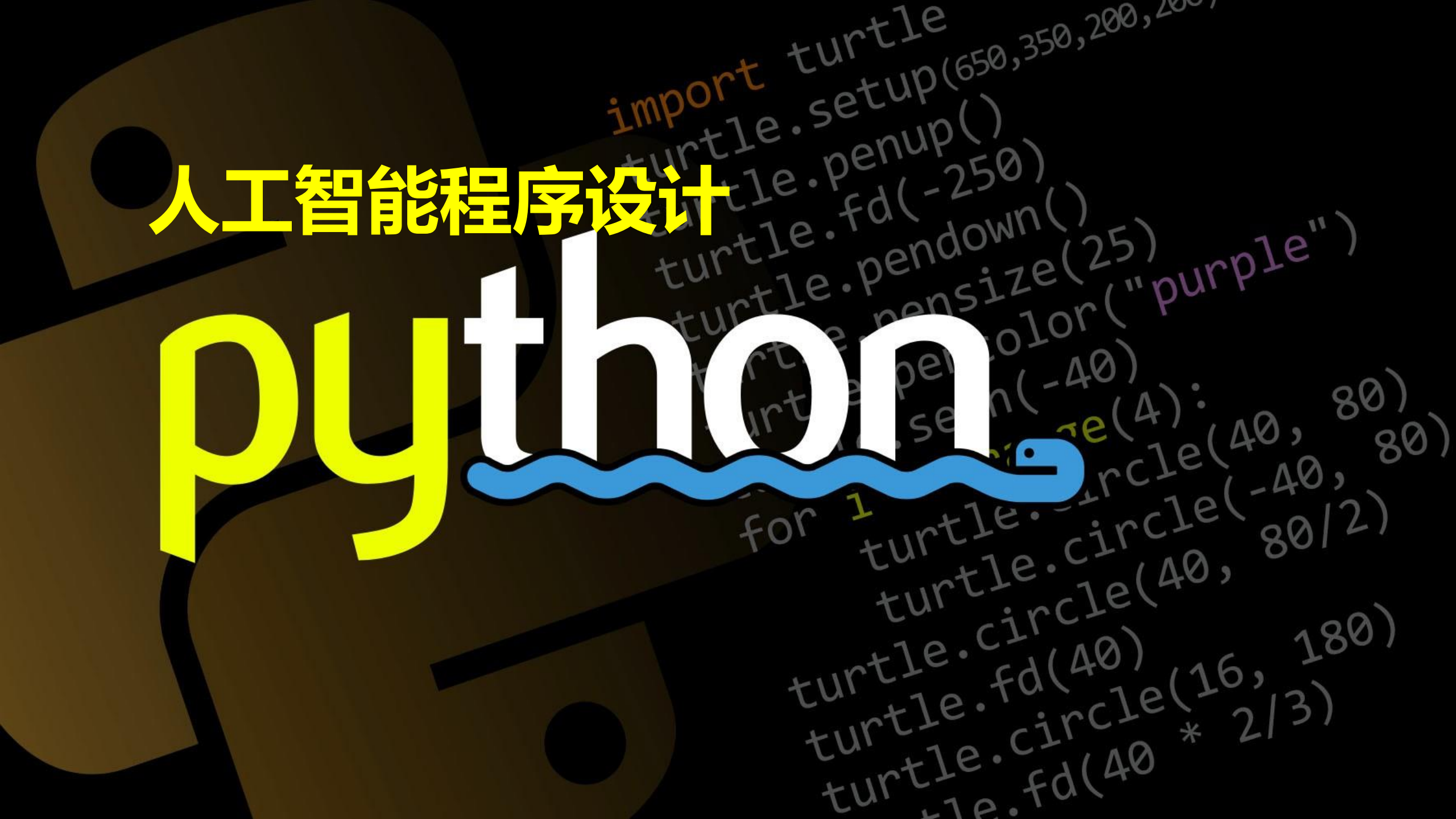
人工智能程序设计


python
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for 1 in range(4):
turtle.circle(40, 80)
turtle.circle(-40, 80)
turtle.fd(40)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)

人工智能程序设计

11.1 从传统编程到机器学习

北京石油化工学院 人工智能研究院

刘 强

11.1.1 编程范式的根本转变

传统编程和机器学习代表了两种截然不同的问题解决思路：

- **传统编程**：程序员定义规则，计算机执行规则
- **机器学习**：计算机从数据中学习规则

理解这个转变是学习机器学习的关键第一步。

传统编程：规则驱动

在传统编程中，程序员需要明确定义处理逻辑。规则很明确，计算机严格按照规则执行：

这里的规则很明确：除以2余数为0的数就是偶数。

```
def is_even(number):
    if number % 2 == 0:
        return True
    else:
        return False
```

机器学习：数据驱动

判断图片是否包含猫，用传统编程很难描述"什么是猫"：

- 猫有四条腿？但桌子也有四条腿
- 猫有毛？但狗也有毛
- 猫会叫？但声音在图片中无法体现

机器学习提供了全新的解决思路：给机器大量数据样本，让机器自己学习规律。

机器学习的思路

机器学习不需要程序员定义规则，而是从数据中自动学习：

```
## 机器学习的思路（伪代码）
## 1. 收集数据：1000张猫的图片 + 1000张非猫的图片
## 2. 训练模型：让机器分析这些图片，找出猫的特征规律
## 3. 预测新图片：机器根据学到的规律判断新图片是否包含猫

## 实际使用时的简化流程
model = train_cat_classifier(cat_images, non_cat_images)
result = model.predict(new_image)  # True表示是猫，False表示不是猫
```

11.1.2 机器学习的核心价值

机器学习解决"人无法明确写出规则"的问题：

- **图像识别**：识别照片中的物体、人脸、文字等
- **推荐系统**：根据用户历史行为推荐商品、音乐、视频
- **预测分析**：预测股价、天气、销量等未来趋势
- **自然语言处理**：机器翻译、情感分析、智能客服

共同特点：规律存在但难以用传统编程方式描述，需要从大量数据中自动发现模式。

11.1.3 机器学习基本概念

为了更好地理解机器学习的核心概念，我们用"学生考试"这个熟悉的场景来类比：

- **数据（Data）**：就像"考试题库"
- **特征（Features）**：就像"题目中的关键信息"
- **标签（Labels）**：就像"题目的标准答案"
- **训练（Training）**：就像"学生做题库学知识点"
- **预测（Prediction）**：就像"学生用学到的知识点答新题"

特征（Features）

特征是用来描述数据的关键信息。比如要预测"明天是否下雨"，相关特征可能包括：

- 今天的温度
- 湿度
- 云量
- 风速

特征的选择和质量直接影响模型的预测效果。

分类与回归：两种基本任务类型

机器学习的监督学习任务主要分为两类，可以用考试题型来类比：

分类任务：选择题

- 结果是离散的类别
- 例子：邮件是否为垃圾邮件、图片中的动物类型、学生是否及格

回归任务：填空题

- 结果是连续的数值
- 例子：房价预测、温度预测、学生考试分数

分类任务示例

分类任务预测的是离散的类别：

```
## 分类任务示例
## 输入：学生的平时成绩、出勤率、作业完成率
## 输出：及格 或 不及格

features = [85, 0.9, 0.95]  # [平时成绩, 出勤率, 作业完成率]
prediction = model.predict([features])
print(prediction)  # 输出：'及格' 或 '不及格'
```

回归任务示例

回归任务预测的是连续的数值：

```
## 回归任务示例
## 输入：房屋面积、地段、房龄
## 输出：具体房价（连续数值）

features = [120, 8.5, 5]  # [面积(平米), 地段评分, 房龄(年)]
prediction = model.predict([features])
print(prediction)  # 输出：285.6（万元）
```

数据集的组成

机器学习的数据集由特征矩阵和标签数组两部分组成。以"预测学生是否挂科"为例：

- **特征矩阵**：平时成绩、出勤次数、作业完成率
- **标签数组**：挂科用0表示，不挂科用1表示

学生	平时成绩	出勤次数	作业完成率	结果
学生1	85分	28次	95%	不挂科
学生2	60分	15次	70%	挂科
学生3	92分	30次	100%	不挂科
学生4	45分	8次	30%	挂科

任务类型的判断

分类任务：预测"挂科"或"不挂科"这样的离散类别

回归任务：预测具体的考试分数（连续数值）

数据规模：实际项目中通常需要成百上千甚至更多的样本来训练一个可靠的模型。

实践练习

练习 11.1.1：编程思路对比

解释传统编程和机器学习在解决问题思路上的根本差异

实践练习

练习 11.1.2：应用场景判断

举例说明什么情况下适合用传统编程，什么情况下适合用机器学习

实践练习

练习 11.1.3：任务类型识别

区分以下任务是分类还是回归：

- 预测明天的最高温度
- 判断邮件是否为垃圾邮件
- 预测股票价格
- 识别图片中的数字

本节小结

- 传统编程：规则驱动，程序员定义规则
- 机器学习：数据驱动，从数据中学习规则
- 分类任务：预测离散类别（选择题）
- 回归任务：预测连续数值（填空题）
- 核心概念：数据、特征、标签、训练、预测