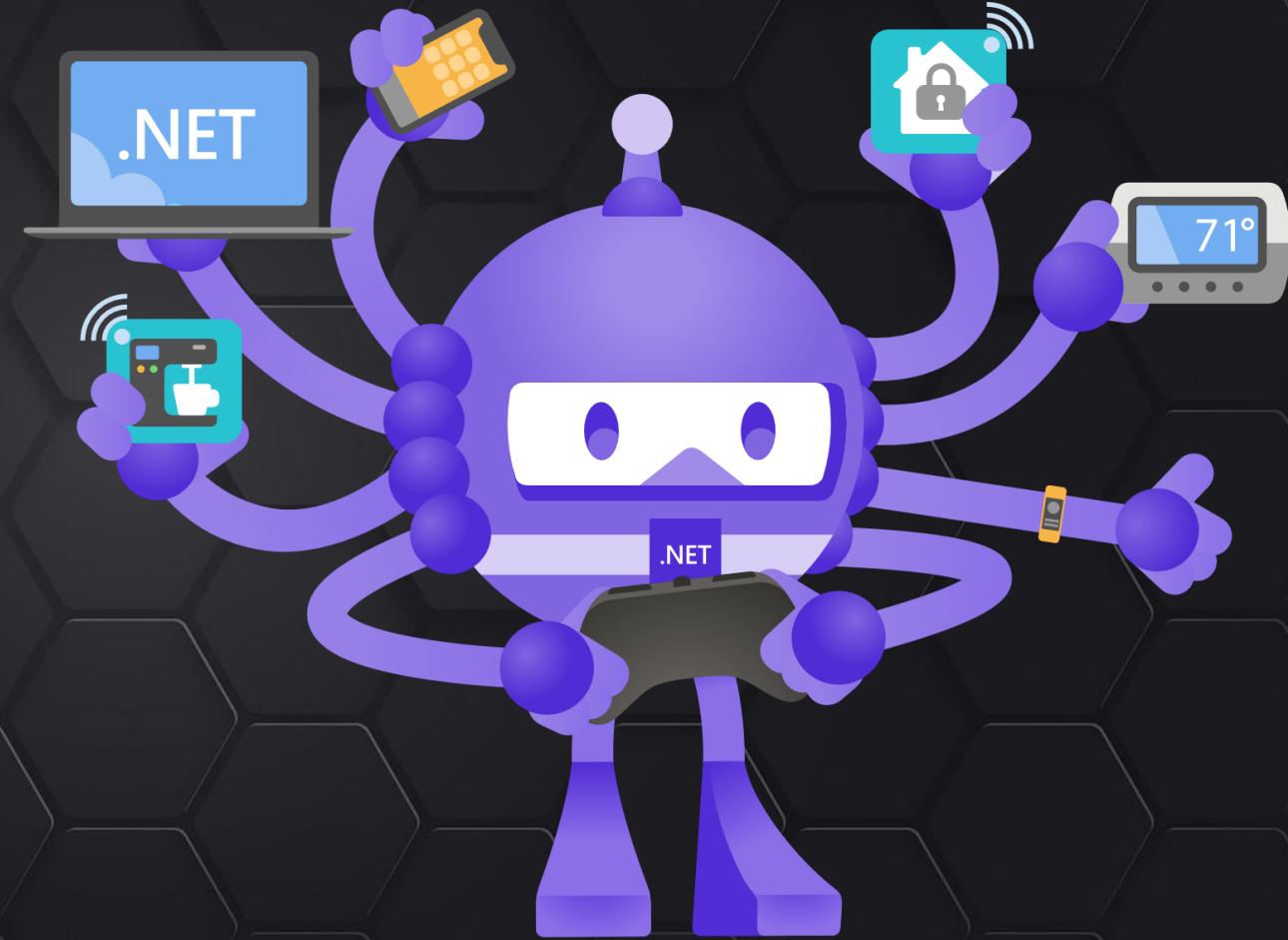


Reactor

基于.NET Core 的
人工智能动手实验



个人介绍



Kinfey Lo – (卢建晖)

Microsoft Cloud Advocate

前微软MVP、Xamarin MVP和微软RD，拥有超过10年的云原生、人工智能和移动应用经验，为教育、金融和医疗提供应用解决方案。 Microsoft Ignite, TechEd 会议讲师，Microsoft AI 黑客马拉松教练，目前在微软，为技术人员和不同行业宣讲技术和相关应用场景。

爱编程(Python , C# , TypeScript , Swift , Rust , Go)

专注于人工智能，云原生，跨平台移动开发

Github : <https://github.com/lokinfey>

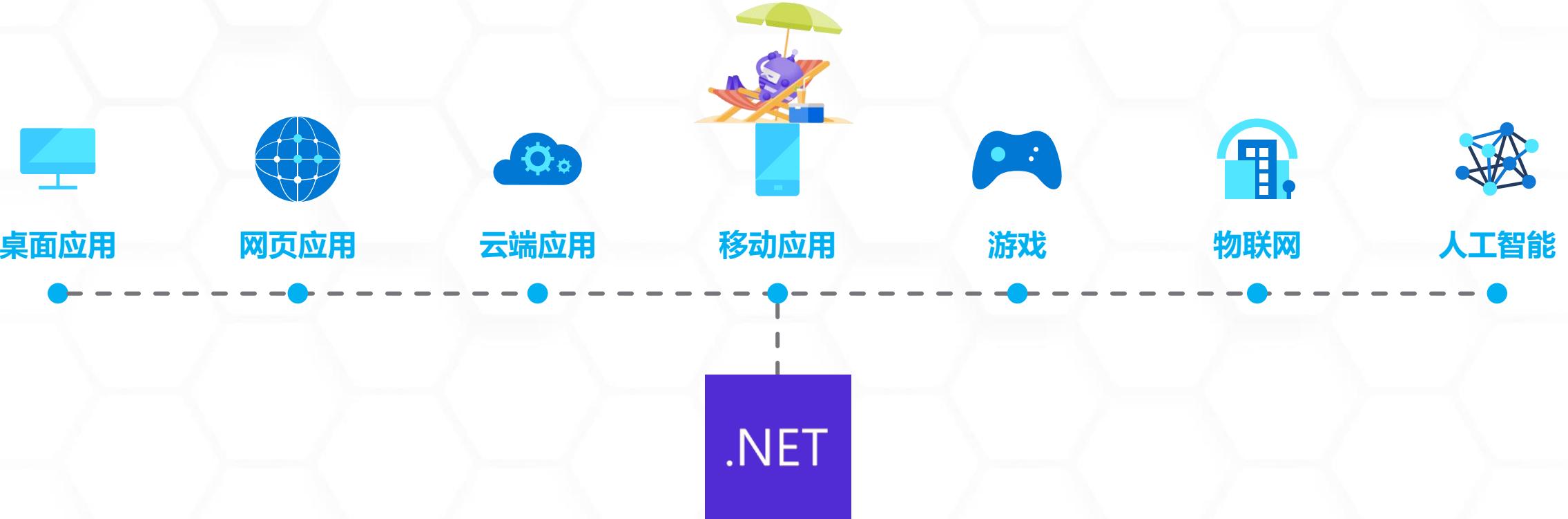
Email : kinfeylo@microsoft.com Blog : <https://blog.csdn.net/kinfey>

Twitter : @Ljh8304

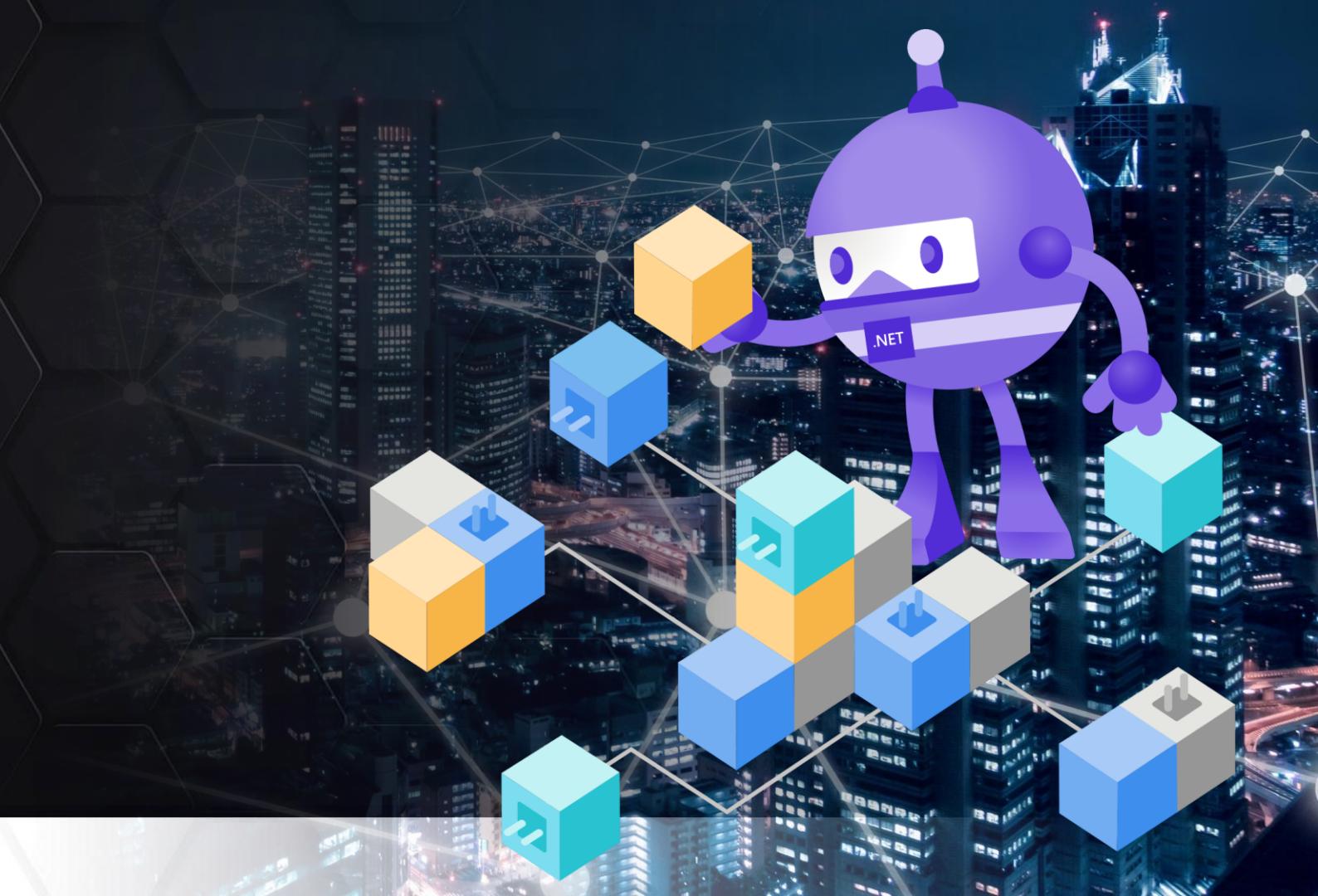


.NET

跨平台利器



00.人工智能介绍



人工智能场景无处不在



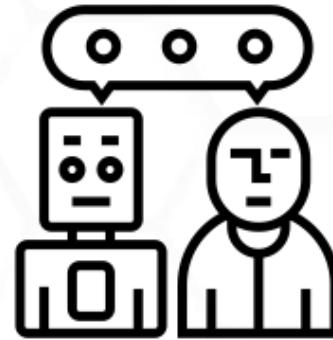
天气预测



电子商务



计算机视觉



自然语言

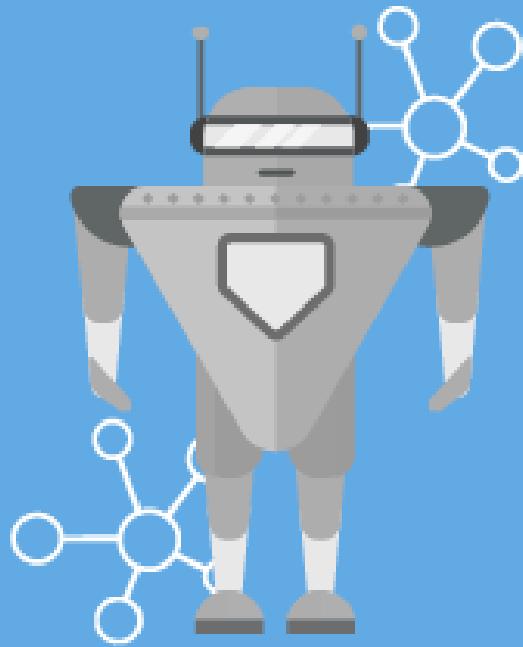


无人驾驶

机器学习？深度学习？人工智能？

人工智能

是一种让计算机能够模仿人类智能的技术。
它包括机器学习



1950's

1960's

1970's

1980's

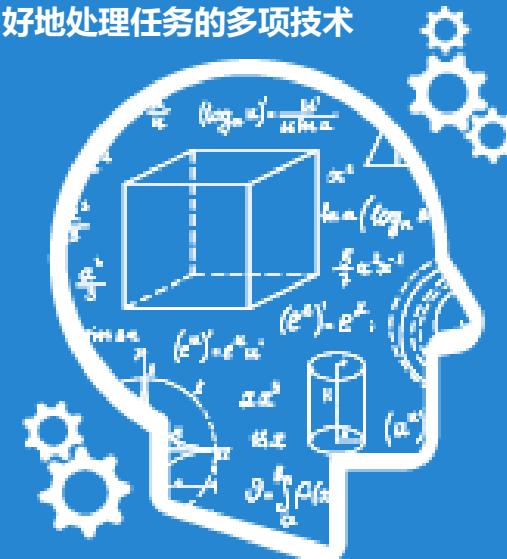
1990's

2000's

2010's

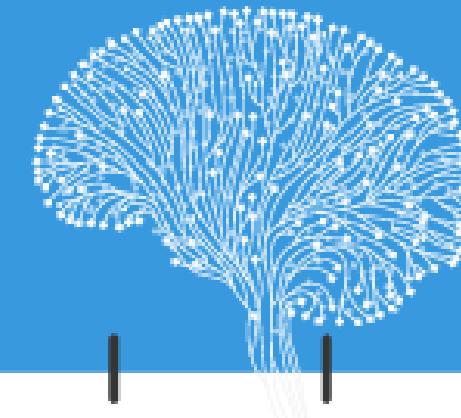
机器学习

是人工智能的一部分，它包括让计算机能够依靠经验更好地处理任务的多项技术

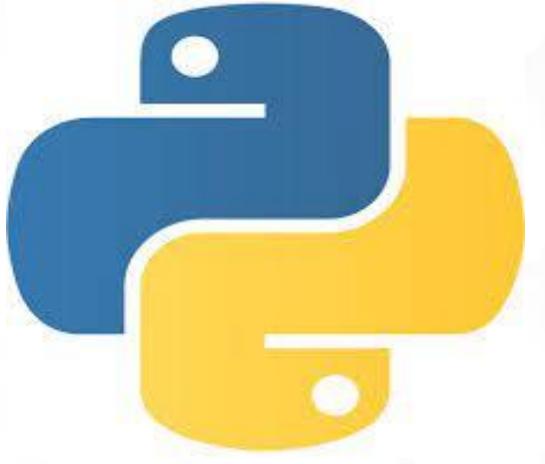


深度学习

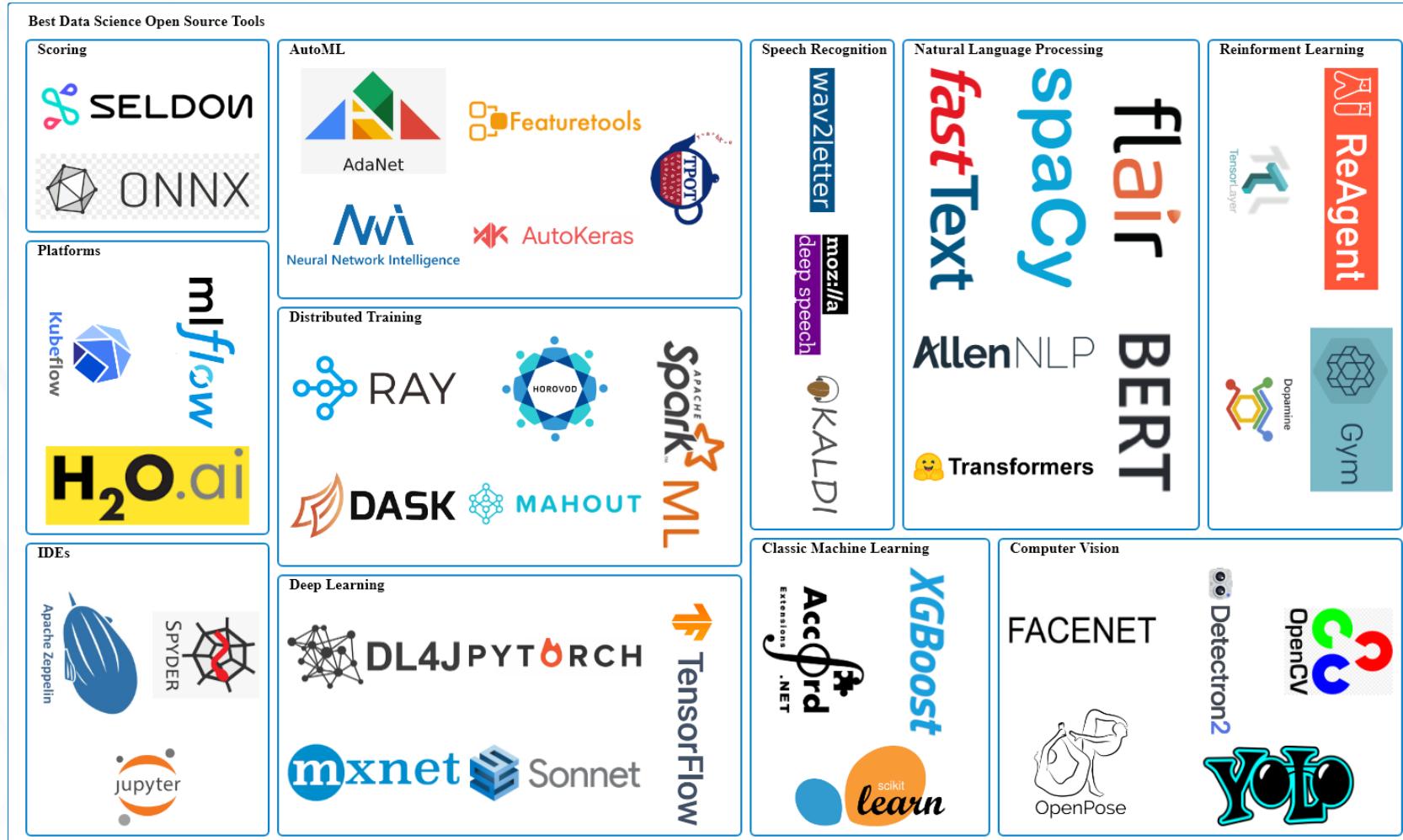
又是机器学习的一部分，它以人工神经网络为基础，让计算机能够自我训练。



人工智能主要技术栈



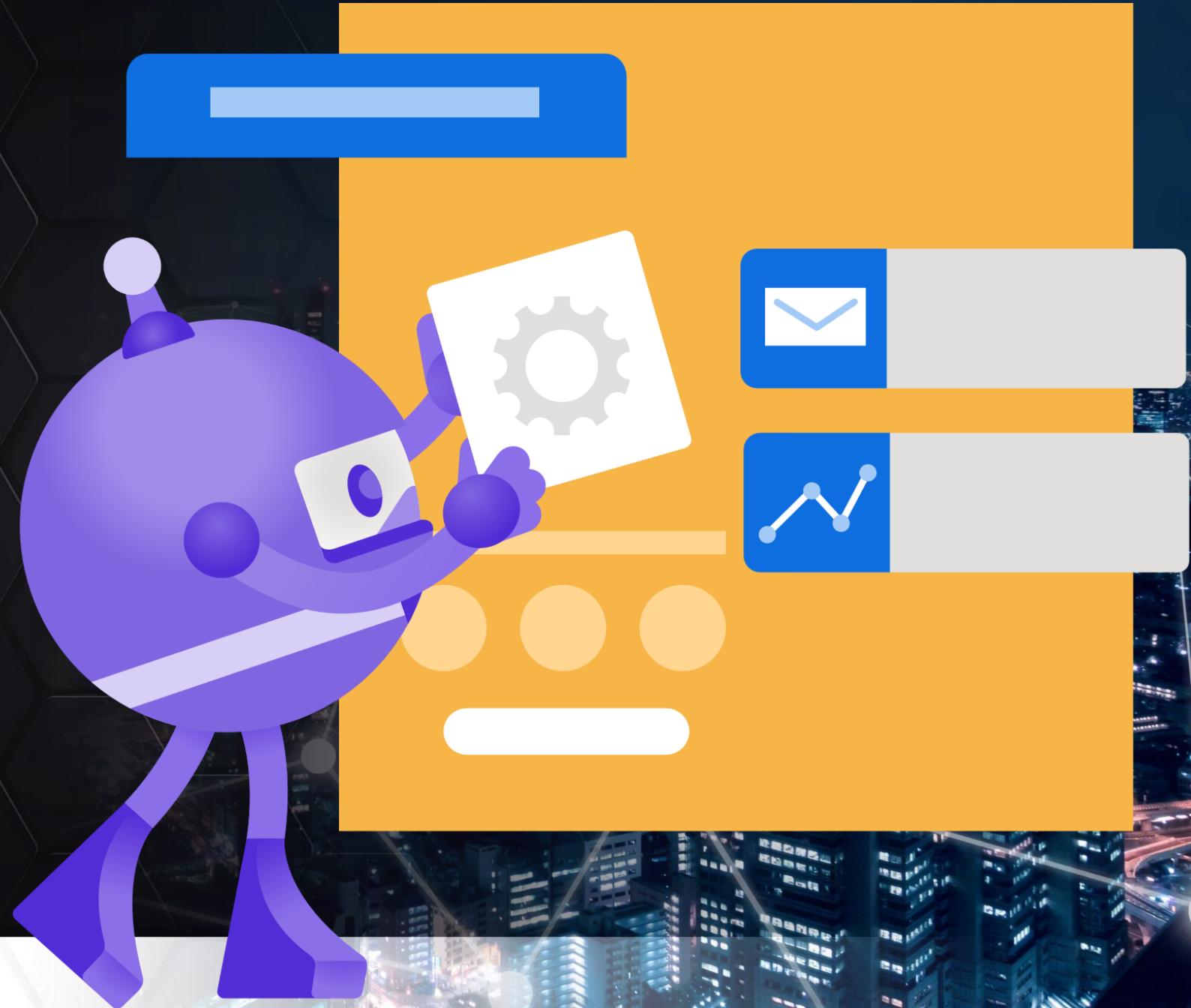
Python



.NET 程序员如何成为一名数据科学家?



01. 准备工作



Jupyter NoteBook



Jupyter notebook (<http://jupyter.org/>) 是一种 Web 应用，能让用户将说明文本、数学方程、代码和可视化内容全部组合到一个易于共享的文档中。

可选择语言: 支持超过40种编程语言，包括Python、R、Julia、Scala等。

分享笔记本: 可以使用电子邮件、Dropbox、GitHub和Jupyter Notebook Viewer与他人共享。

交互式输出: 代码可以生成丰富的交互式输出，包括HTML、图像、视频、LaTeX等等。

大数据整合: 通过Python、R、Scala编程语言使用Apache Spark等大数据框架工具。支持使用pandas、scikit-learn、ggplot2、TensorFlow来探索同一份数据。

.NET Interactive

<https://github.com/dotnet/interactive>

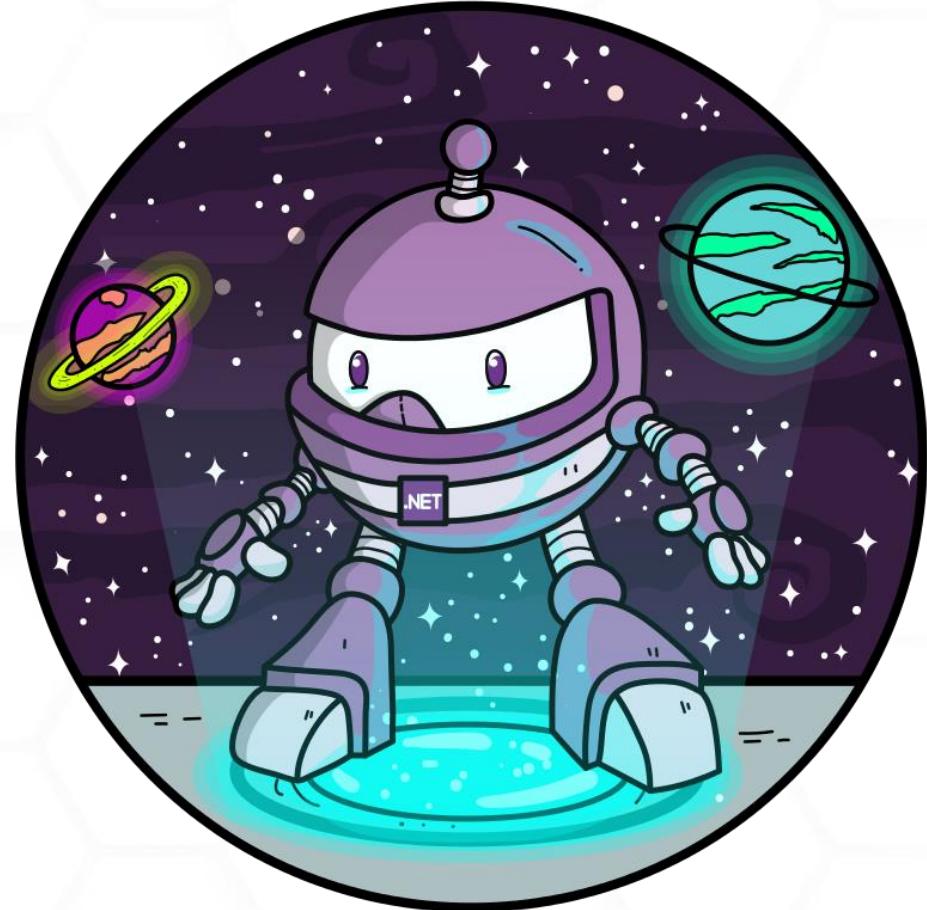
.NET Interactive 利用 .NET 的强大功能并将其嵌入到您的交互体验中。以前所未有的方式在您的应用中共享代码、探索数据、编写和学习。

Notebook : Jupyter、Interactive 和 Visual Studio Code

代码机器人

树莓派设备

可嵌入的脚本引擎REPL



进入动手环节的第一步-基础环境配置学习

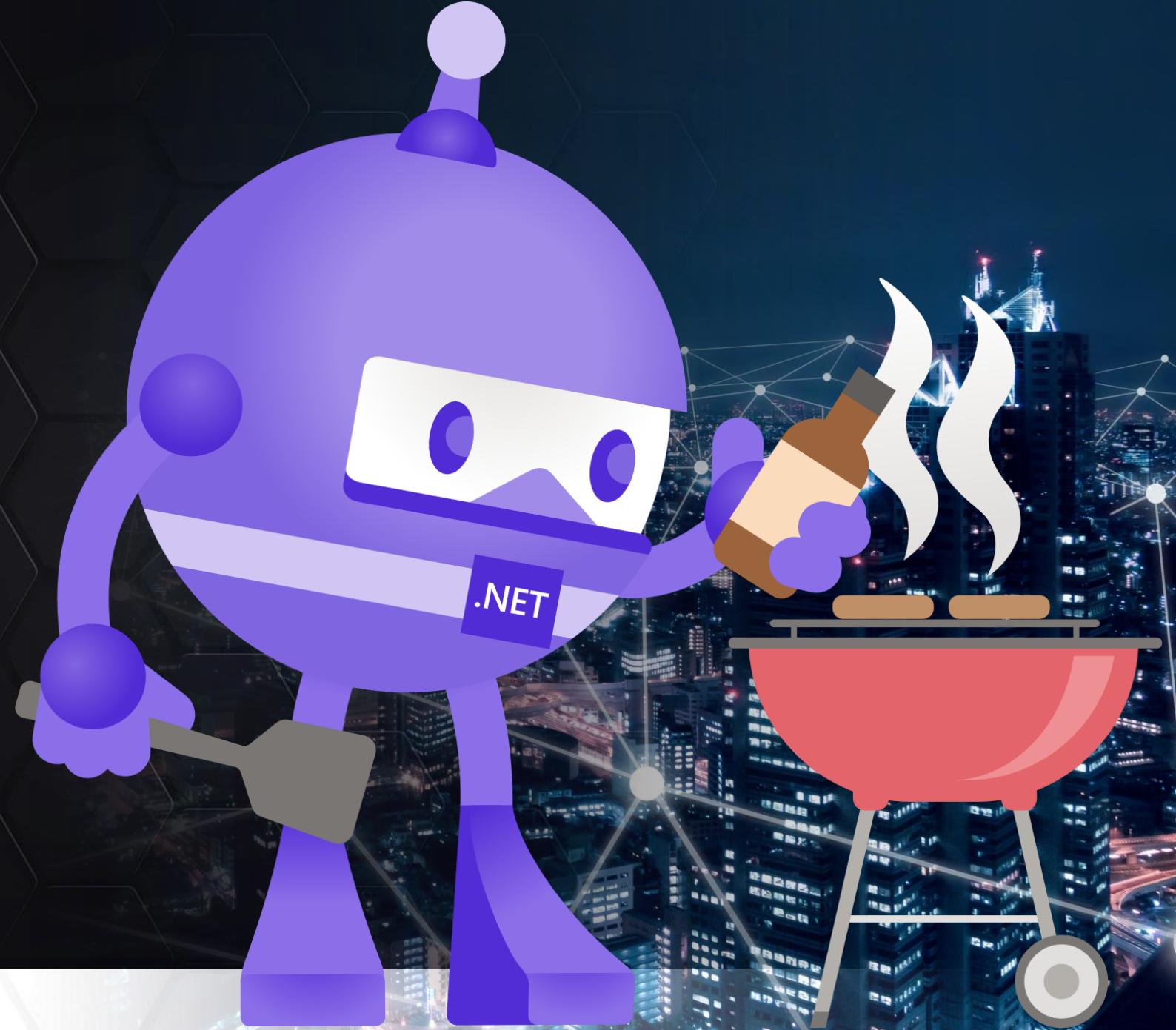
00_Install.ipynb

时间： 20分钟



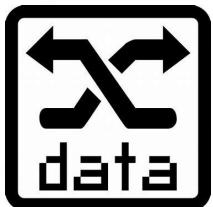
1. 扫码关注
2. 进入公众号
3. 输入mlnet 获取动手实验手册

02. 关于数据



机器学习大全

数据决定一切



场景

Supervised Learning
监督学习

Semi-supervised Learning
半监督学习

Transfer Learning
迁移学习

Unsupervised Learning
无监督学习

Structured Learning
结构化学习

Reinforcement Learning
强化学习

方法

Regression
(回归)

Classification
(分类)

Binary Classification
(二分类)

Multi-class Classification
(多分类)

GAN

结果

Linear Model
(线性模型)

None-Linear Model
(非线性模型)

深度学习

SVM

决策树

K-NN

在分析数据之前，你必须要了解认识数据

Python 数据分析的主要工具

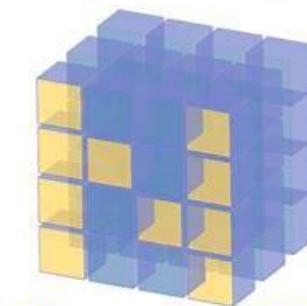


Numpy 介绍

NumPy(Numerical Python) 是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

NumPy (Numeric Python) 提供了许多高级的数值编程工具，如：矩阵数据类型、矢量处理，以及精密的运算库。专为进行严格的数字处理而产生。多为很多大型金融公司使用

```
import numpy as np  
  
arr = np.array([1, 2, 3, 4, 5])  
  
print(arr)
```



NumPy

Pandas 介绍

pandas 是基于NumPy 的一种工具，该工具是为解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。你很快就会发现，它是使Python成为强大而高效的数据分析环境的重要因素之一。

Pandas 适合处理一个规正的二维数据（一维也可以，应用较少），即有 N 行 N 列，类似于 SQL 执行后产出的，或者 无合并单元格Excel 表格 这样的数据。它可以把多个文件的数据合并在一起，如果结构不一样，也可以经过处理进行合并。

```
import pandas as pd
```

```
data={'state':['Ohi','Ohi','Ohi','Nev','Nev','Nev'],
      'year':[2000,2001,2002,2003,2004,2005],
      'pop':[1.5,3.4,3.0,1.2,2.9,3.2]}
```

```
frame=pd.DataFrame(data)
```

```
frame
```

	state	year	pop
0	Ohi	2000	1.5
1	Ohi	2001	3.4
2	Ohi	2002	3.0
3	Nev	2003	1.2
4	Nev	2004	2.9
5	Nev	2005	3.2

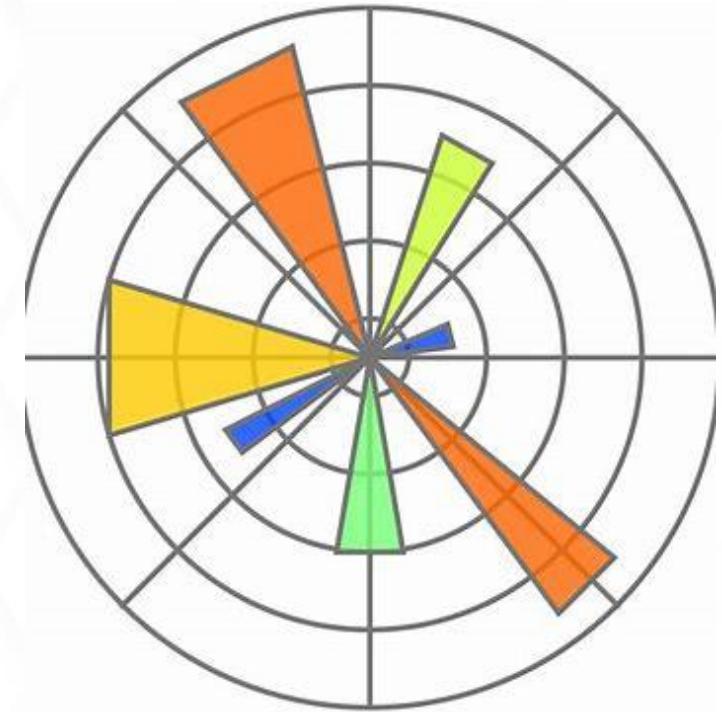
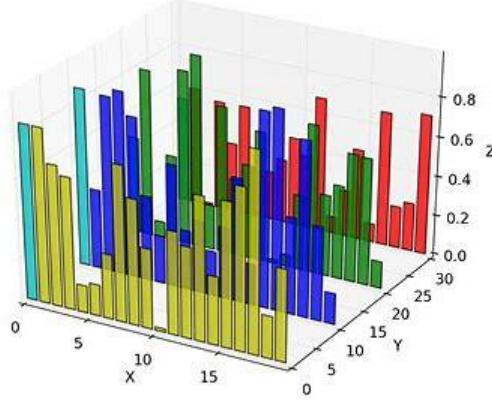
Pandas



Matplotlib 介绍

Matplotlib是一个Python的2D绘图库，它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形。通过Matplotlib，开发者可以仅需要几行代码，便可以生成绘图。一般可绘制折线图、散点图、柱状图、饼图、直方图、子图等等。Matplotlib使用Numpy进行数组运算，并调用一系列其他的Python库来实现硬件交互。

```
import matplotlib.pyplot as plt  
  
fig = plt.figure()
```



.NET Core 下如何做数据分析



NumSharp

NumSharp 是 NumPy 的 C# 版本，它尽可能与 NumPy 编程接口保持一致，包括函数名称和参数位置。通过引入 NumSharp 工具库，您可以轻松地将 Python 代码转换为 C# 或 F# 代码。



NumSharp

NumSharp (NS) 是面向 C# 的 NumPy 端口，面向 .NET Standard。NumSharp 是使用 C# 和 F# 进行科学计算所需的基本包

<https://github.com/SciSharp/NumSharp>

Microsoft.Data.Analysis

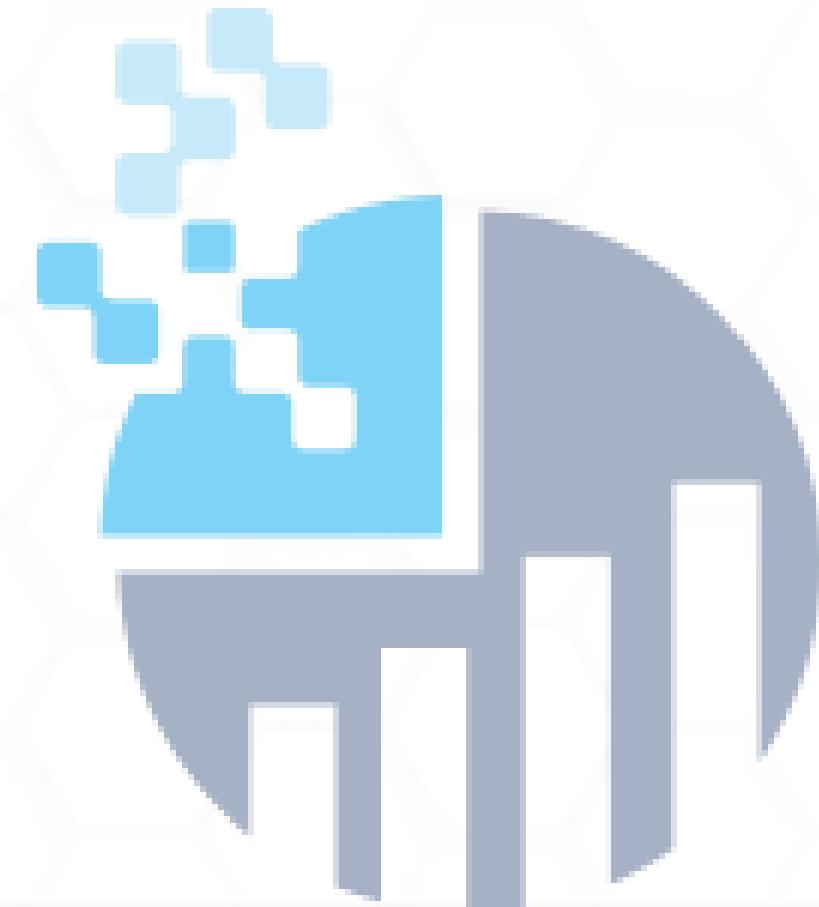
用于数据分析和转换的易于使用的高性能库。

<i>index</i>	DateTimes	Ints	Strings
0	2019-01-01 00:00:00Z	10	<null>
1	2019-01-01 00:00:00Z	100	Foo!
2	2019-01-02 00:00:00Z	<null>	<null>

XPlot

XPlot 是出色的 F# 和 .NET 数据可视化包。 使用 Plotly 和 Google Charts 作为“后端”来呈现精美的可视化效果。

<https://github.com/fslaborg/XPlot>



进入动手环节的第二步-数据整理学习

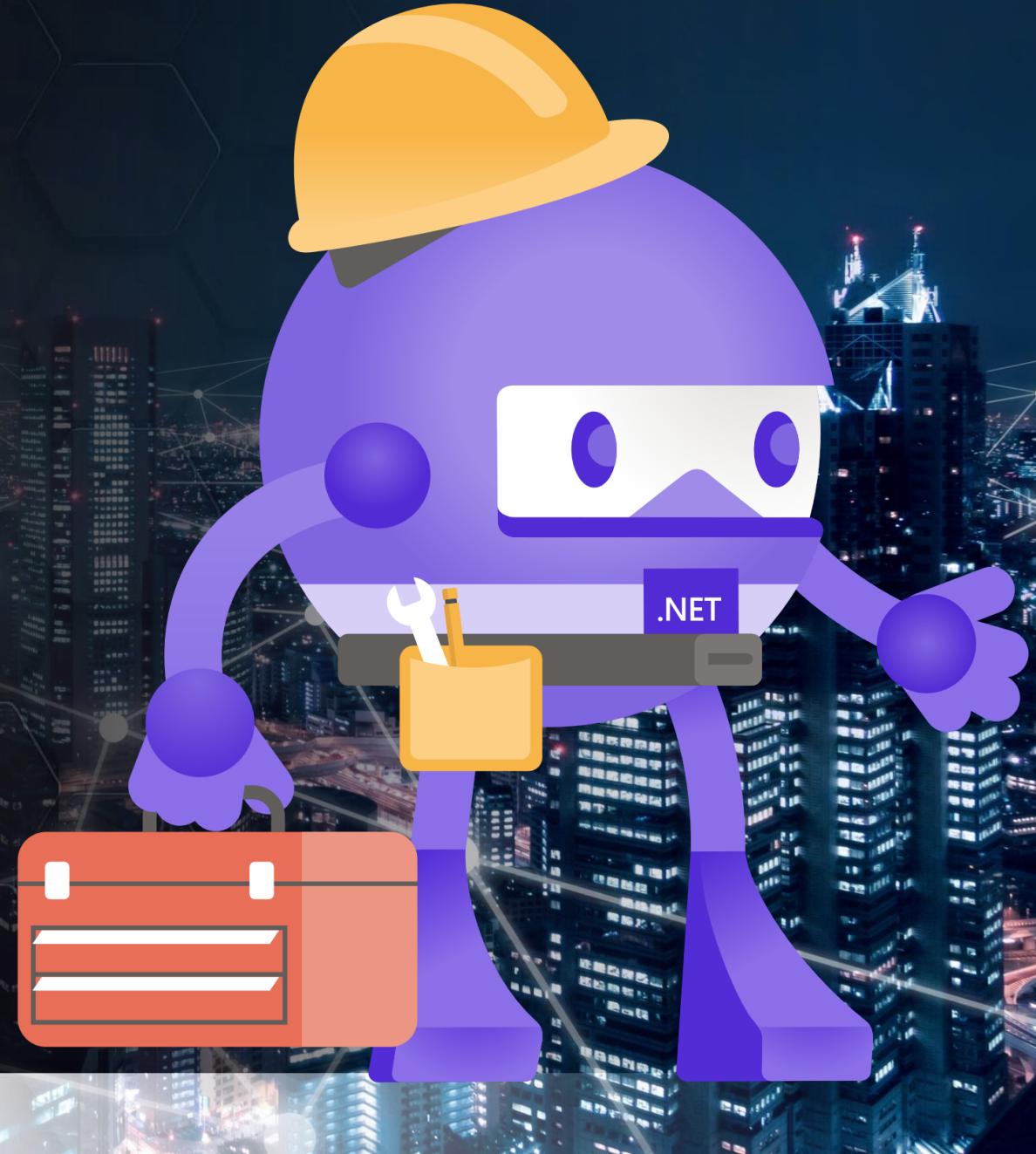
[01_Data_Analysis.ipynb](#)

时间： 20分钟



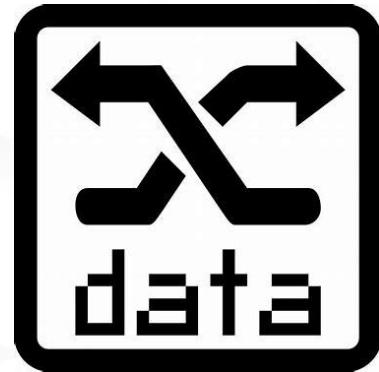
1. 扫码关注
2. 进入公众号
3. 输入mlnet 获取动手实验手册

03. 机器学习和ML.NET



监督学习

Supervised Learning – 预测未来



$$y = f([x_1, x_2, x_3, \dots])$$

经验决定一切，整合大量的标注数据，应用在预测价格，判断分类等场景上

需要找到一个基于特征数据，生成结果的方法

一般都有多个特征数据

x 值拟合到计算中，从而为训练数据集中的所有情况合理准确地生成 y 。

监督学习的常见解决方法

Regression (回归)

预测明天的气温是多少度

Classification (分类)

预测明天是阴、晴还是雨

输出	连续数据	离散数据
目的	定量- 找到最佳拟合	定性- 决策边界
评价	拟合度	精度
场景	预测房价, 天气	垃圾邮件, 物品分类

监督学习- 回归示例

问题导入

期望已有共享单车的信息，来预测某个季节，某个天气条件的投放的单车数量



监督学习- 分类

分类是有监督机器学习的一种形式，在这种学习中，您训练模型使用特征（我们函数中的x值）预测标签（y），该标签计算属于多个可能类别的观察案例的概率，并预测适当的标签。最简单的分类形式是二进制分类，其中标签为0或1，表示两个类中的一个；例如，“真”或“假”“内部”或“外部”“是”或“否”；等等

**Binary Classification
(二分类)**

**Multi - Classification
(多分类)**

多类分类可以看作是多个二进制分类器的组合。解决问题有两种方法：

一对一（OVR），其中为每个可能的类值创建一个分类器，对于预测为此类的情况，结果为正，对于预测为任何其他类的情况，结果为负

一对一（OVO），其中为每个可能的类对创建一个分类器。

监督学习- 二分类示例

问题导入

根据身体指标，判断是否患有糖尿病



监督学习- 多分类示例

问题导入

基于企鹅的生理指标判断企鹅类型, 'Adelie', 'Gentoo', 'Chinstrap'



无监督学习

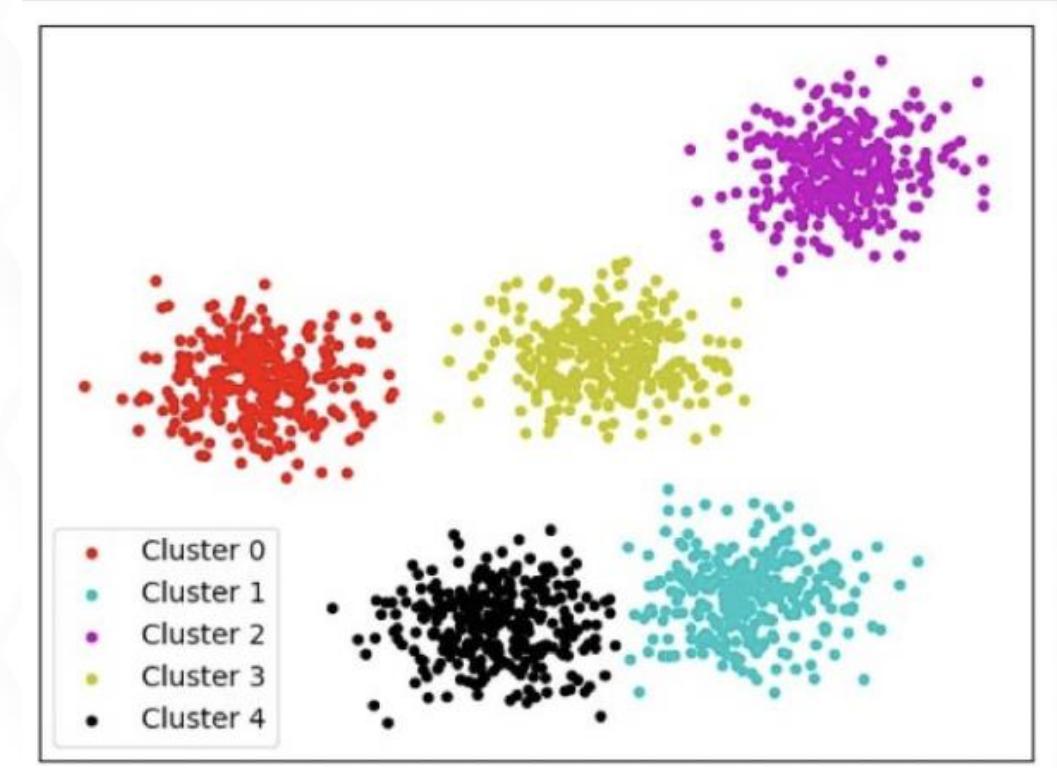
区别于监督学习，无监督学习希望机器做到无师自通，

在完全没有任何标签的情况下，机器到底能学到什么样的知识

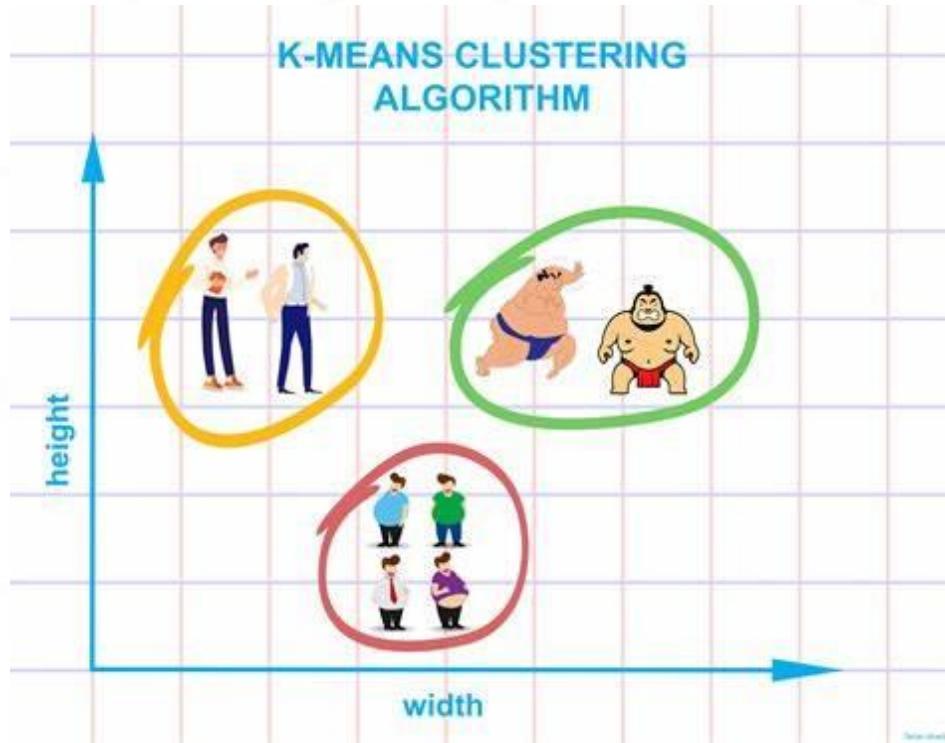
如：通过多维度的用户行为分析，广告投放

推荐系统

异常查找



无监督学习基本算法 – 聚类分析 K-means

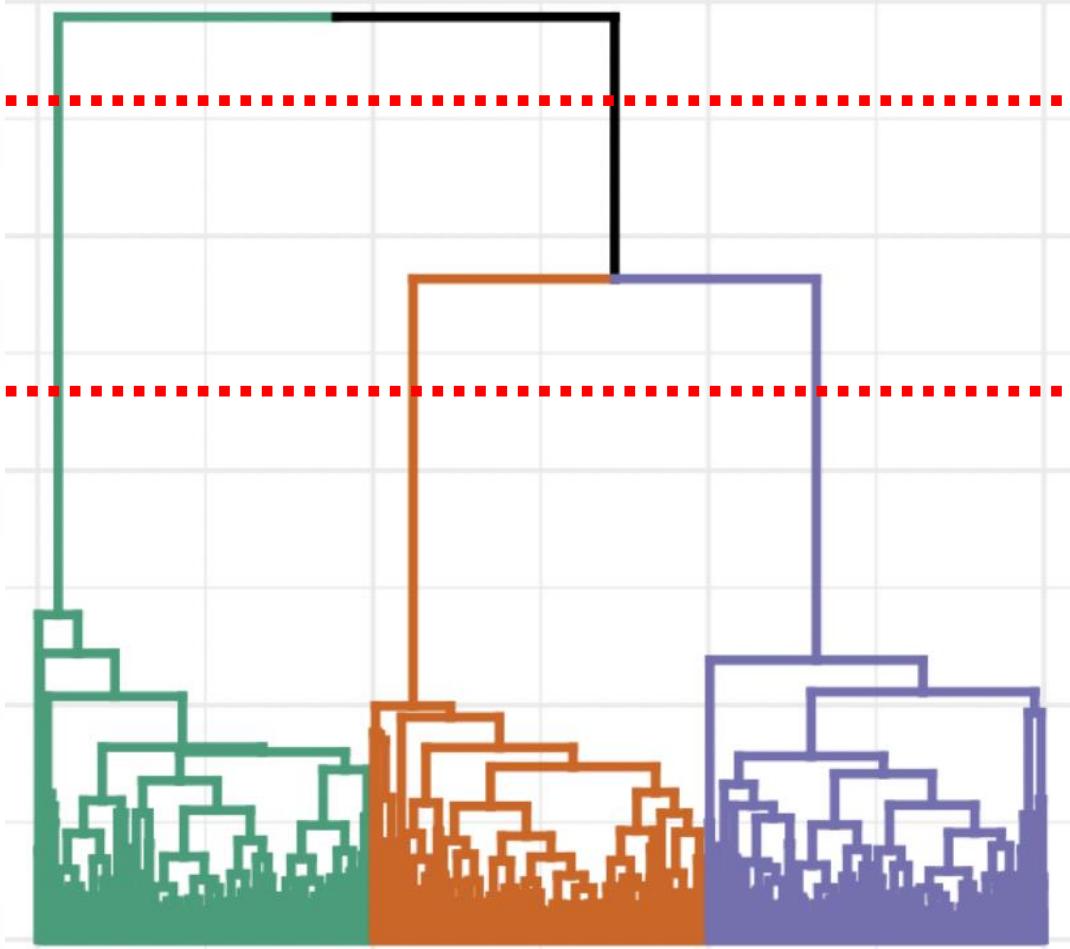


是在没有给定划分类别的情况下，根据数据的相似度进行分组的一种方法，分组的原则是组内距离最小化而组间距离最大化。

K-means算法是典型的基于距离的非层次聚类算法，在最小化误差函数的基础上将数据划分为预定的K类别，采用距离作为相似性的评级指标，即认为两个对象的距离越近，其相似度越大。

- 1、随机设置K个特征空间内的点作为初始的聚类中心
- 2、对于其他每个点计算到K个中心的距离，未知的点选择最近的一个聚类中心点作为标记类别
- 3、接着对着标记的聚类中心之后，重新计算出每个聚类的新中心点（平均值）
- 4、如果计算得出的新中心点与原中心点一样，那么结束，否则重新进行第二步过程

无监督学习基本算法 – 聚类分析 Hierarchical Clustering



分层聚类通过分割方法或聚集方法创建聚类。除法是一种“自上而下”的方法，从整个数据集开始，然后逐步查找分区。凝聚聚类是一种“自下而上”的方法。

- (1) 初始化：把每个样本各自归为一类（每个样本自成一类），计算每两个类之间的距离，在这里也就是样本与样本之间的相似度（本质还是计算类与类之间的距离）。
- (2) 寻找各个类之间最近的两个类，把它们归为一类（这样，类的总数就减少了一个）
- (3) 重新计算新生成的这个类与各个旧类之间的距离（相似度）
- (4) 重复 (2) (3) 步，直到所有的样本都归为一类，结束。

ML.NET



ML.NET

用于构建自定义 ML 模型的机器学习框架

技能重用

C# and F#

大规模实践

Azure, Office, Windows

定义ML模型更容易

Automated ML 和 Tools (Model Builder and CLI)

兼容性, 扩展性

TensorFlow, ONNX and Infer.NET

免费, 跨平台, 开源

<https://github.com/dotnet/machinelearning>

ML.NET 能做什么？



Sentiment analysis

Analyze the sentiment of customer reviews using a binary classification algorithm.



Product recommendation

Recommend products based on purchase history using a matrix factorization algorithm.



Price prediction

Predict taxi fares based on parameters such as distance traveled using a regression algorithm.



Customer segmentation

Identify groups of customers with similar profiles using a clustering algorithm.



Object detection

Recognize objects in an image using an ONNX deep learning model.



Fraud detection

Detect fraudulent credit card transactions using a binary classification algorithm.



Sales spike detection

Detect spikes and changes in product sales using an anomaly detection model.



Image classification

Classify images (for example, broccoli vs. pizza) using a TensorFlow deep learning model.

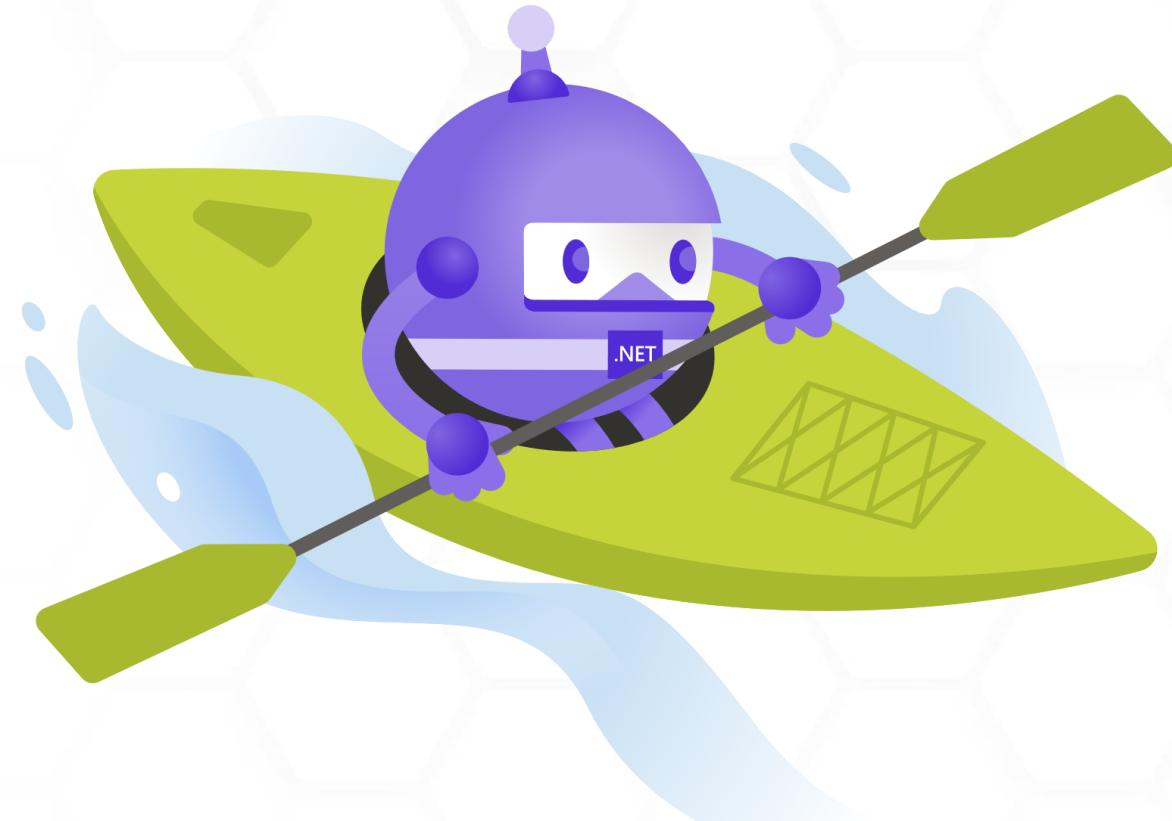


Sales forecasting

Forecast future sales for products using a regression algorithm.

<https://dot.net/ml>

进入动手环节的第三步-ML.NET学习



[02_ML.NET_Basic.ipynb](#)

[02_ML.NET_Case.ipynb](#)

时间：20分钟

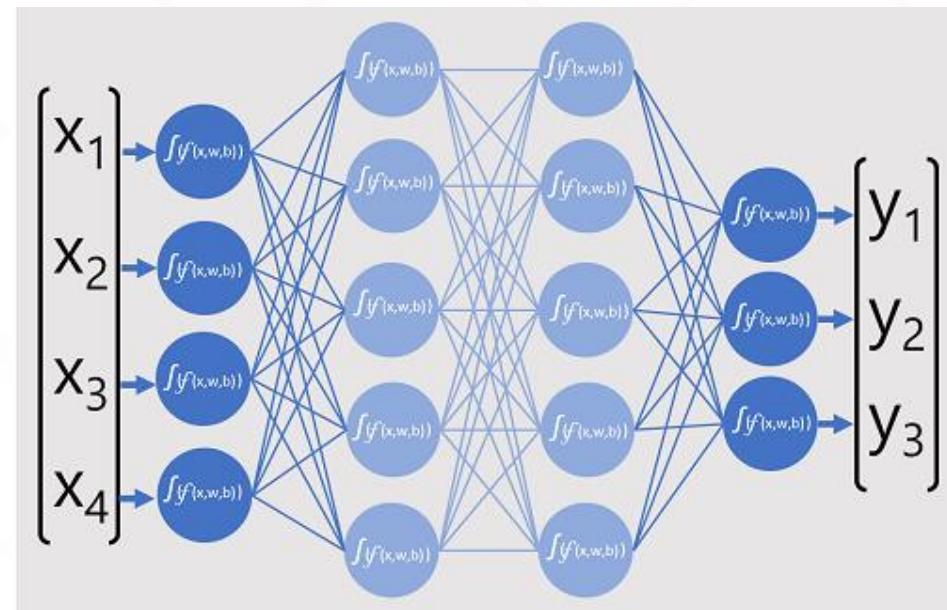


1. 扫码关注
2. 进入公众号
3. 输入mlnet 获取动手实验手册

04. 深度学习 和TensorFlow.NET



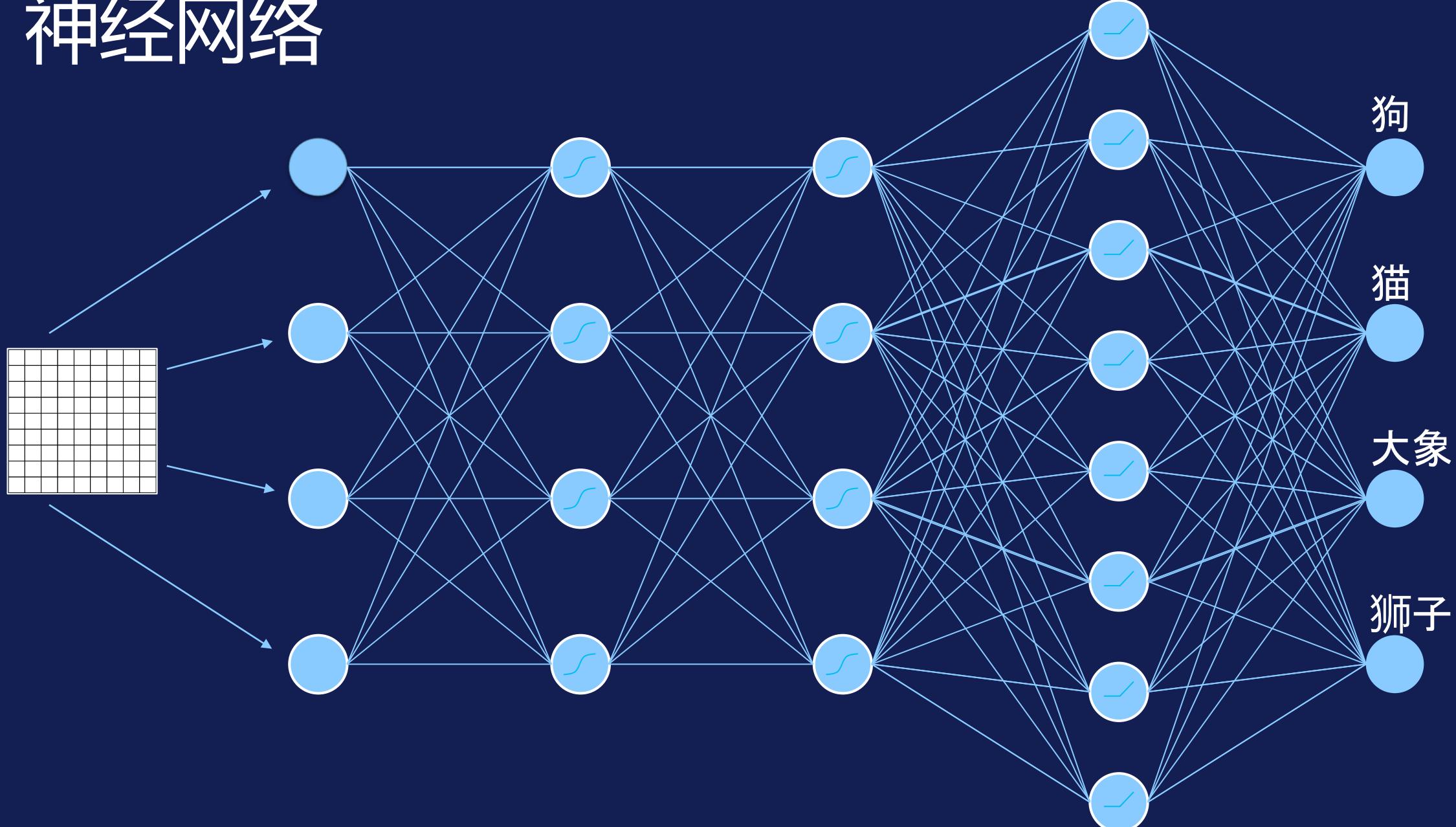
深度学习



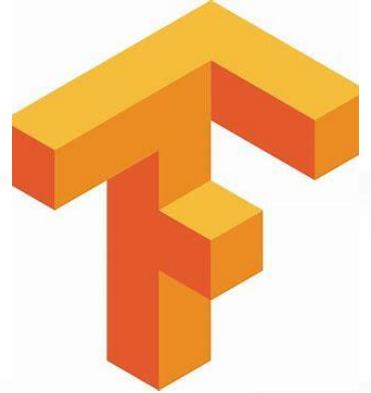
深度神经网络的训练过程包含多个称为“时期”的迭代。对于第一个时期，你首先要为权重分配随机初始化值 (w) 和偏差 (b) 值。然后，该过程如下所示：

1. 具有已知标签值的数据观察特征将提交到输入层。通常情况下，这些观测值分组为多个批次（通常称为小型批处理）。
2. 然后，神经元发挥其作用，并在激活后将结果传递到下一层，直到输出层生成预测。
3. 将预测与实际的已知值进行比较，并对预测值和真实值之间的差异量（称为“损失”）进行计算。
4. 根据结果，计算权重和偏差值的修订值以减少损失，并将这些调整反向传播到网络层中的神经元。
5. 下一个时期使用修改后的权重和偏差值重复批量训练向前传递，有望通过减少损失来提高模型的准确性。

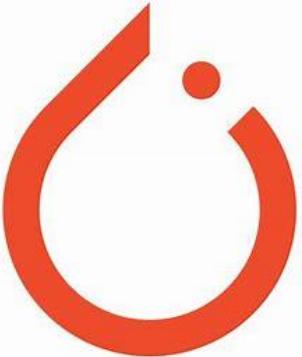
神经网络



常用的深度学习库



TensorFlow



PyTorch



PaddlePaddle

关于机器学习/深度学习 开发环境搭建

<https://blog.csdn.net/kinfey/article/details/117635067>

TensorFlow.NET



TensorFlow.NET

<https://github.com/SciSharp/TensorFlow.NET>

Install TF.NET and TensorFlow binary through NuGet.

```
### install tensorflow C#/F# binding
PM> Install-Package TensorFlow.NET
### install keras for tensorflow
PM> Install-Package TensorFlow.Keras

### Install tensorflow binary
### For CPU version
PM> Install-Package SciSharp.TensorFlow.Redist

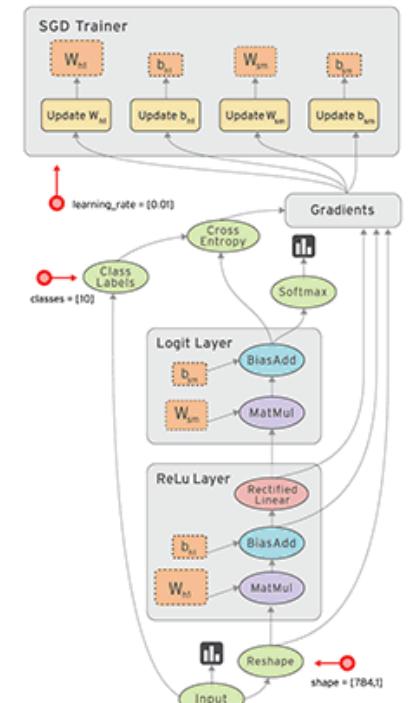
### For GPU version (CUDA and cuDNN are required)
PM> Install-Package SciSharp.TensorFlow.Redist-Windows-GPU
```

Import TF.NET and Keras API in your project.

```
using static Tensorflow.Binding;
using static Tensorflow.KerasApi;
using Tensorflow;
using NumSharp;
```

TensorFlow.NET (TF.NET)

为 TensorFlow 提供了 .NET Standard 绑定。它旨在用 C# 实现完整的 TensorFlow API，允许 .NET 开发人员使用跨平台的 .NET Standard 框架开发、训练和部署机器学习模型。TensorFlow.NET 内置了 Keras 高级接口，并作为独立包 TensorFlow.Keras 发布。



TensorFlow.NET

与其他项目相比，例如 TensorFlowSharp，它只提供 TensorFlow 的低级 C++ API，并且只能运行使用 Python 构建的模型，Tensorflow.NET 还实现了 TensorFlow 的高级 API，您可以使用 C# 或 F# 构建新的机器学习模型。

TensorFlow

Graph Manipulation
Layer (Python)

C++ API (Python)

Tensor Computation
Layer (C++)

TensorFlow.NET

Graph Manipulation
Layer (C#)

C++ API (C#)

Tensor Computation
Layer (C++)

TensorFlowSharp (by Microsoft)

C++ API (C#)

Tensor Computation
Layer (C++)

TensorFlow.NET

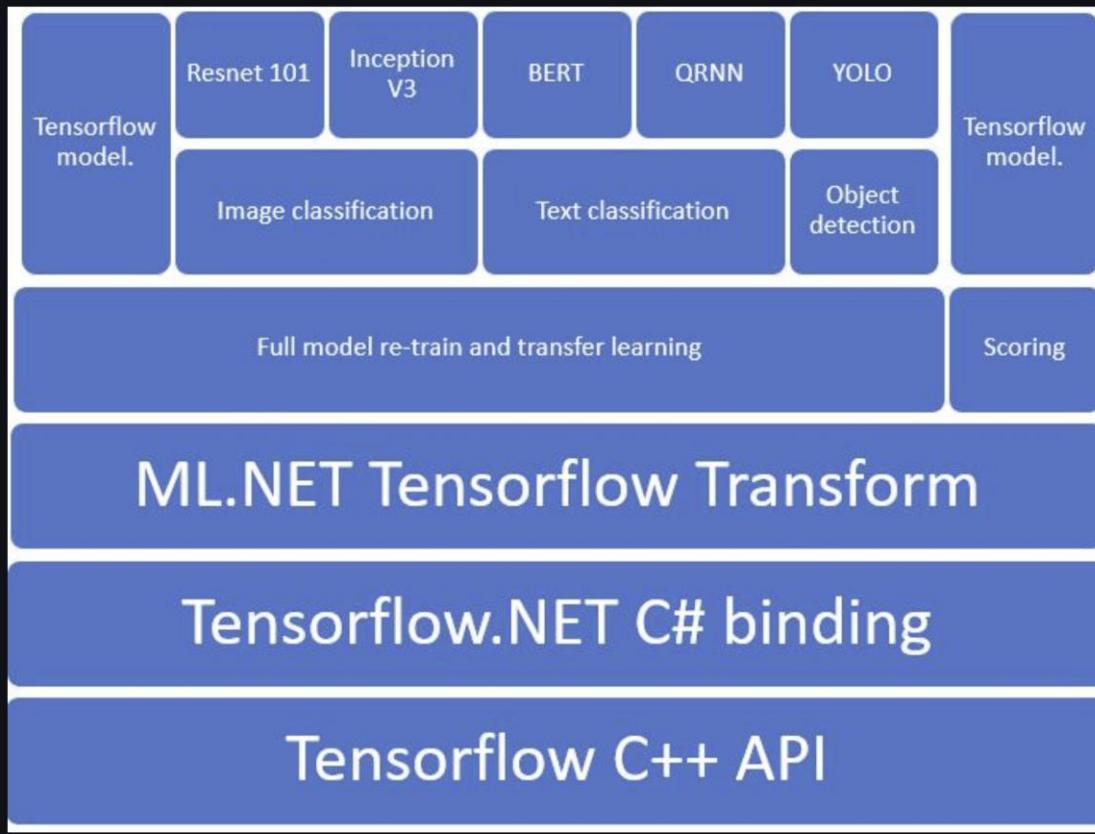
master branch is based on tensorflow v2.4, v0.3x branch is based on tensorflow v2.3, v0.15-tensorflow1.15 is from tensorflow1.15.

TensorFlow	tf native1.14, cuda 10.0	tf native 1.15, cuda 10.0	tf native 2.3, cuda 10.1	tf native 2.4, cuda 11
tf.net 0.4x, tf.keras 0.5				x
tf.net 0.3x, tf.keras 0.4			x	
tf.net 0.2x		x	x	
tf.net 0.15	x	x		
tf.net 0.14	x			

TensorFlow.NET 与 ML.NET

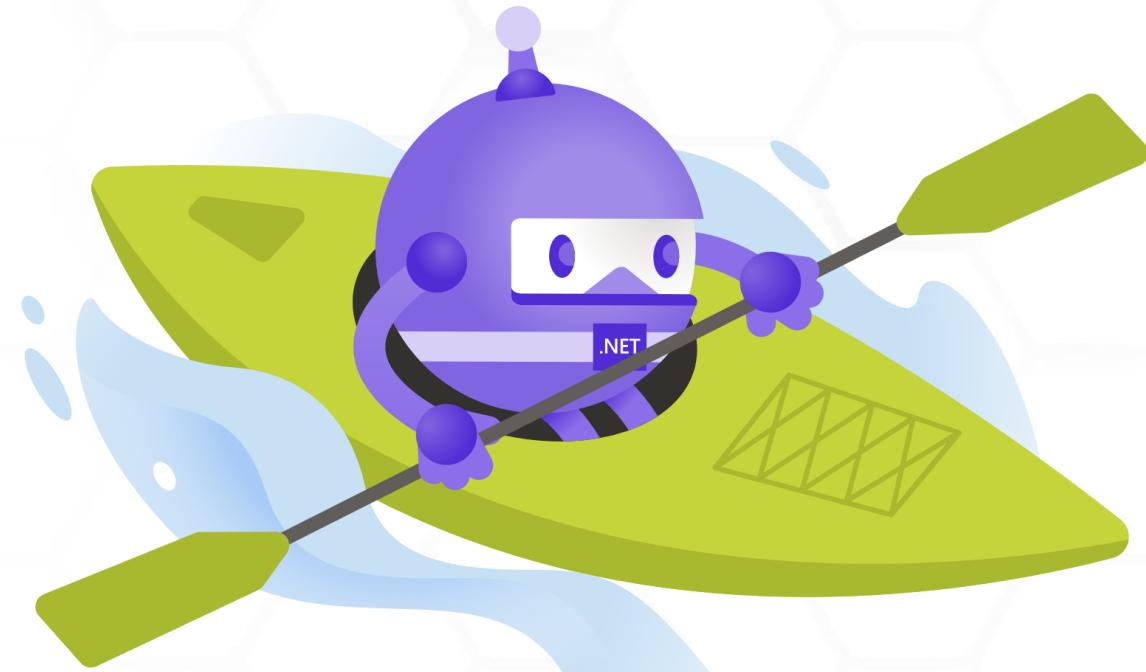
Solution

ML .NET does not have a DNN training infrastructure so we plan to use Tensorflow in the backend through the C# bindings created by Tensorflow .NET.



https://github.com/dotnet/machinelearning/blob/cd591dd492833964b6829e8bb2411fb81665ac6d/docs/specs/DNN/dnn_api_spec.md

进入动手环节的第四步-TensorFlow.NET学习



[**03_TensorFlow.NET_Basic.ipynb**](#)

[**03_TensorFlow.NET_ML.ipynb**](#)

[**03_TensorFlow.NET_DP.ipynb**](#)

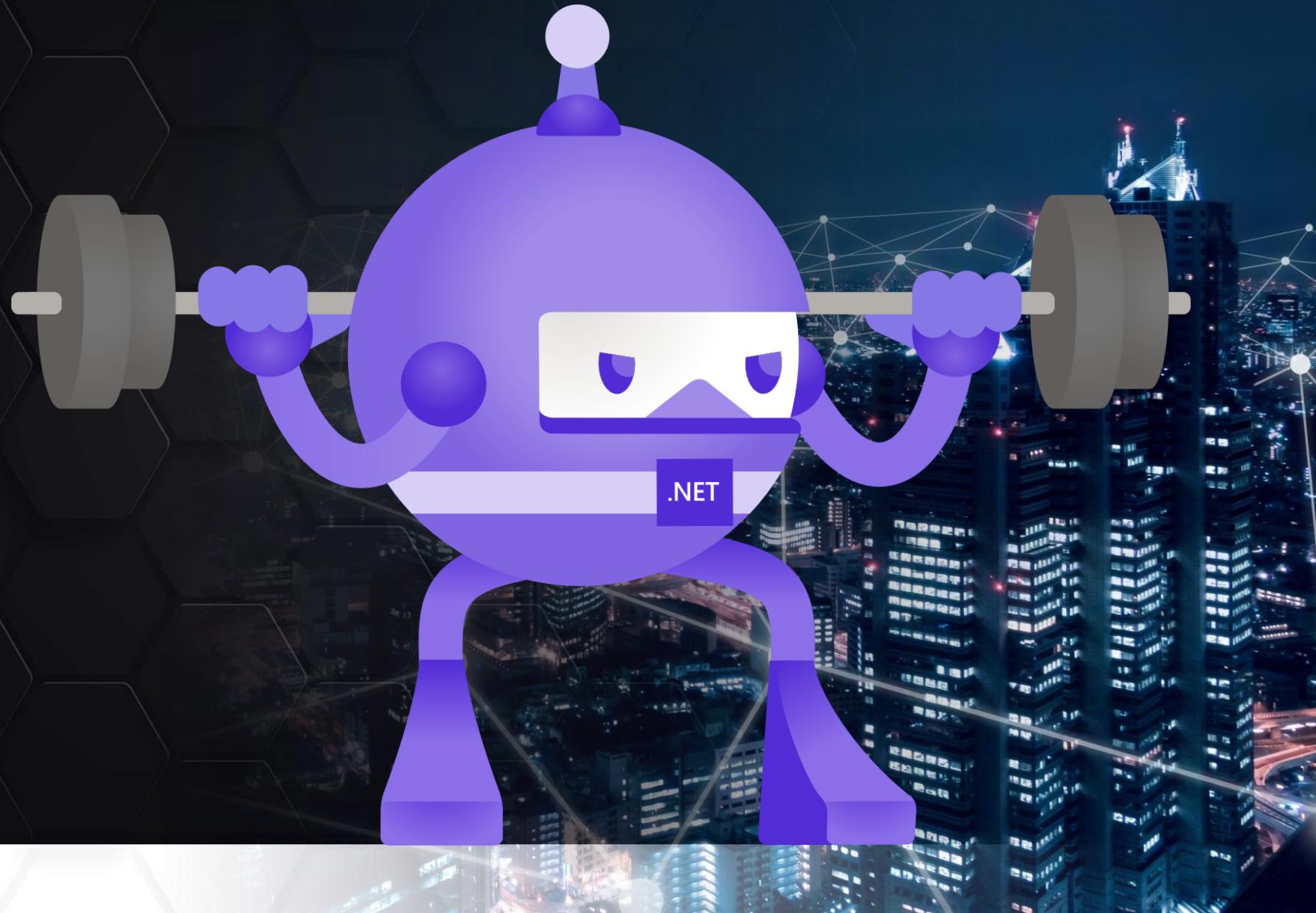
时间：20分钟



- 1. 扫码关注**
- 2. 进入公众号**
- 3. 输入mlnet 获取动手实验手册**

五. 小结

Q&A



MS Learn 的学习模块推荐



Introduction to machine learning

<https://docs.microsoft.com/zh-cn/learn/modules/introduction-to-machine-learning/>



训练和评估深度学习模型

<https://docs.microsoft.com/zh-cn/learn/modules/train-evaluate-deep-learn-models/>

Reactor

谢谢!

