Microsoft

Microsoft Tech Summit 2018

微软技术暨生态大会

Microsoft Tech Summit 2018

微软技术暨生态大会

由零开始用AI

DAIL03

胡浩

Microsoft

Agenda

- 从生物到数学 一些有趣的概念
- 不能更简单的 入门之路
- AI 的 'Hello, World!'

开始之前

☺ 请放松，这是个轻松的动手实验

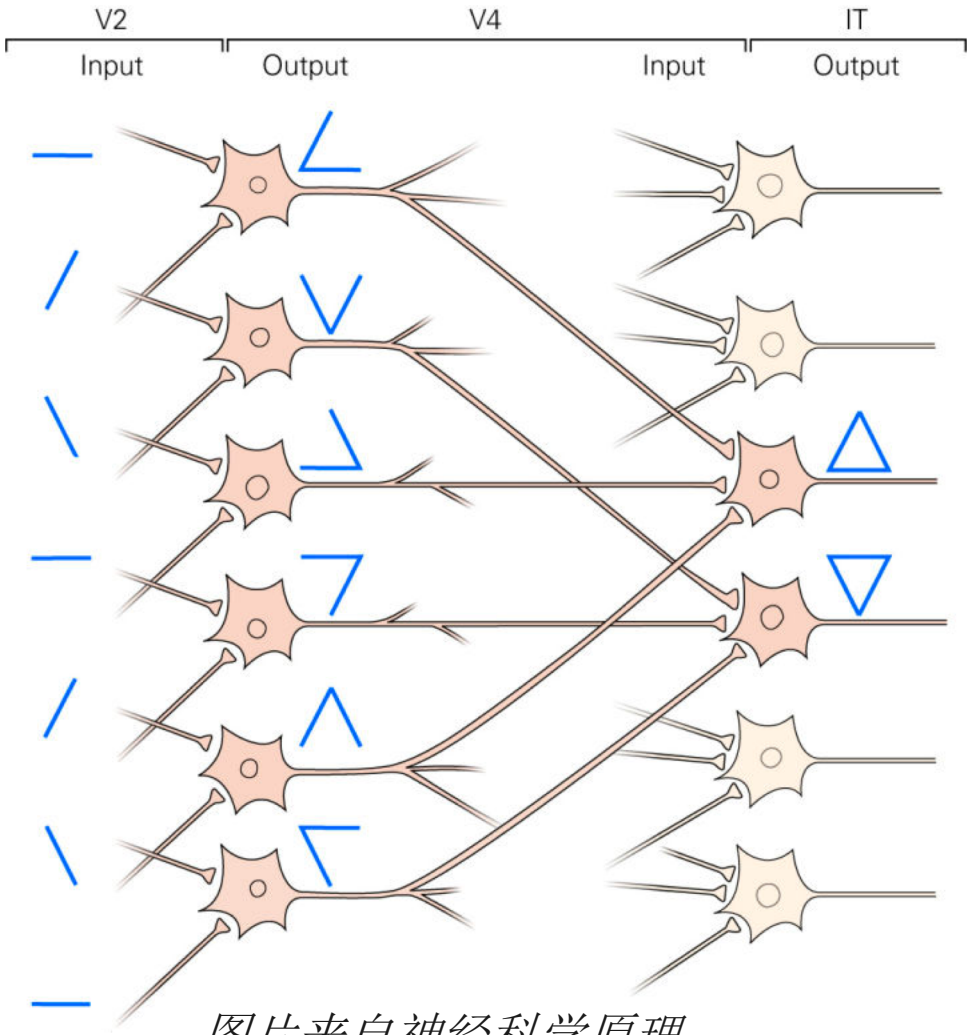
☺ 记得填写反馈表，给个好评哦亲

☺ 时间太紧张？没做完可以打包哦

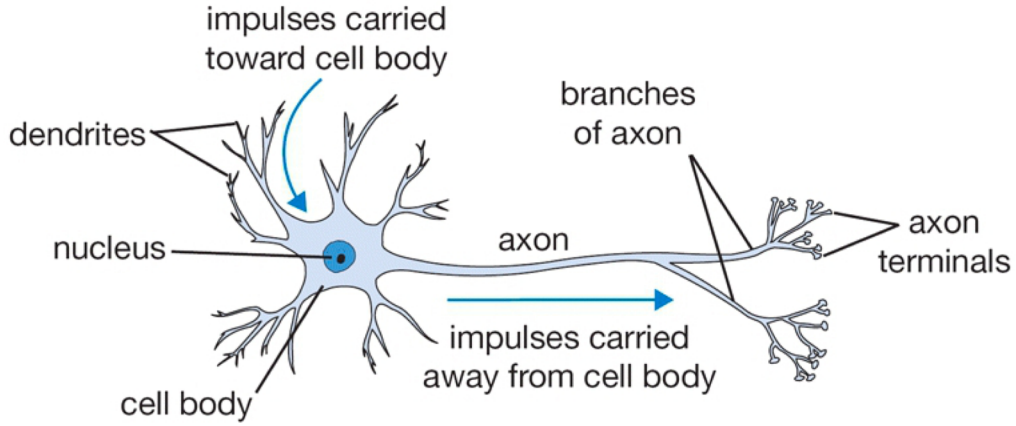
从生物到数学

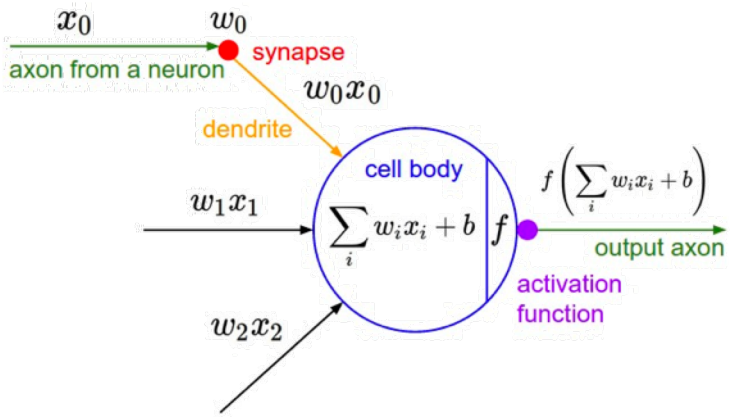
一些有趣的概念

神经元—从生物到数学

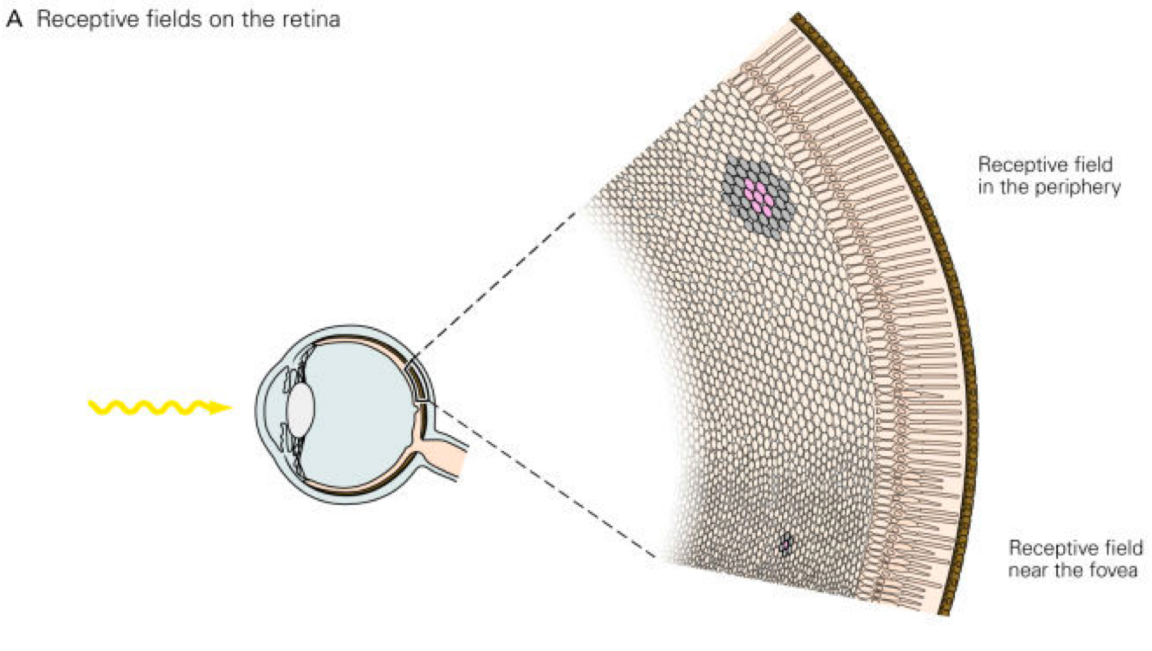


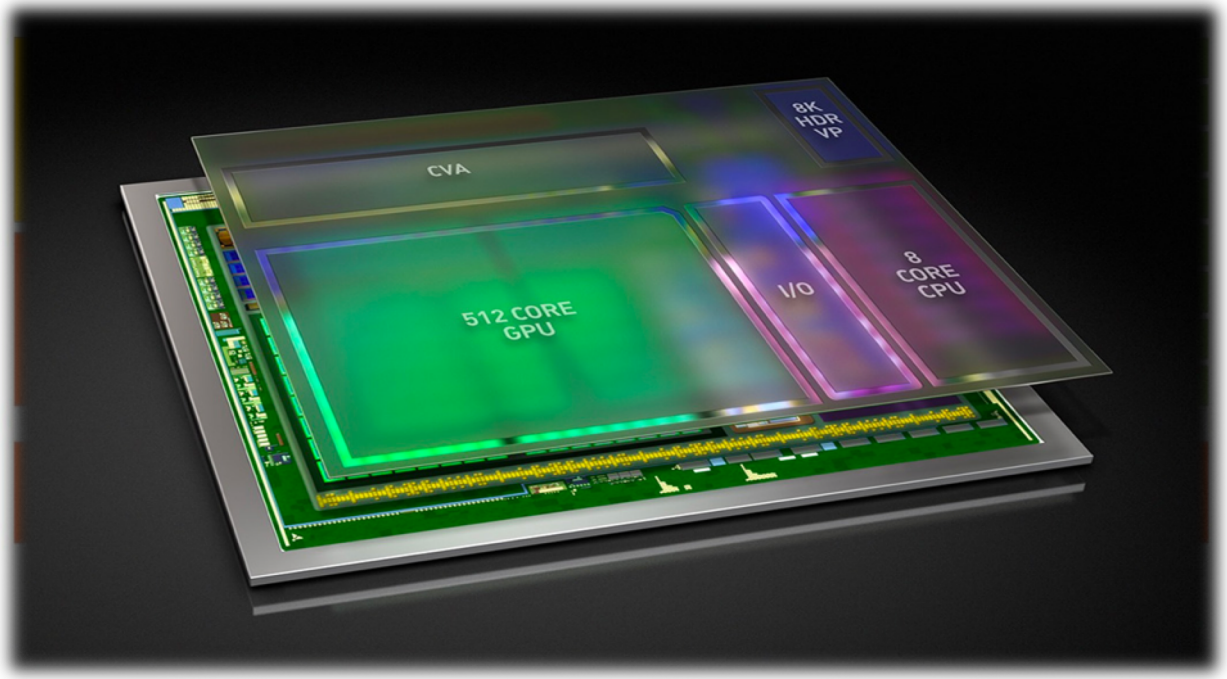
图片来自神经科学原理

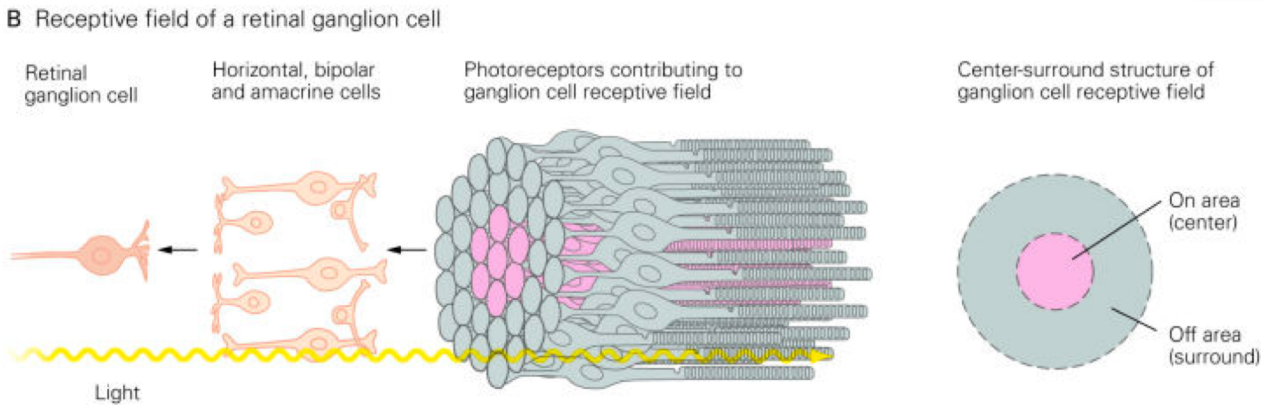


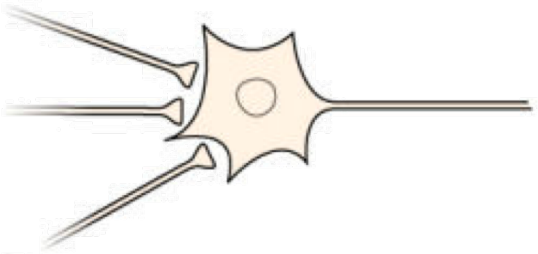


触发——从生物到数学

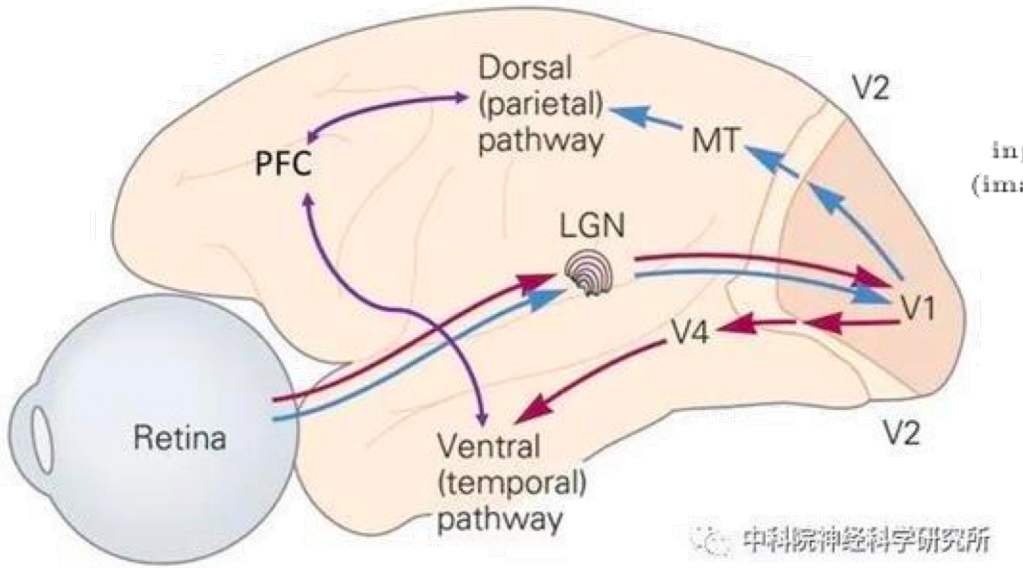

A Receptive fields on the retina
Receptive field
in the periphery
Receptive field
near the fovea


8K
HDR
VP
CVA
512 CORE
GPU
I/O
8
CORE
CPU

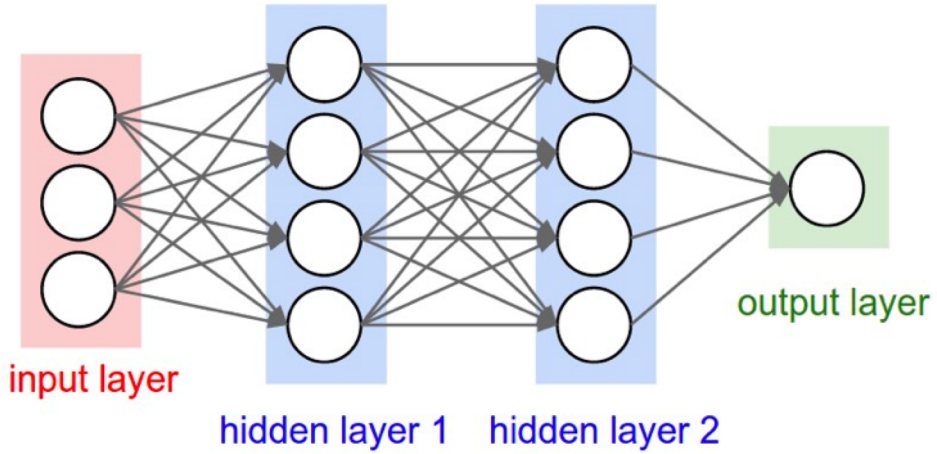

B Receptive field of a retinal ganglion cell
Retinal
ganglion cell
Horizontal, bipolar
and amacrine cells
Photoreceptors contributing to
ganglion cell receptive field
Center-surround structure of
ganglion cell receptive field
On area
(center)
Off area
(surround)
Light

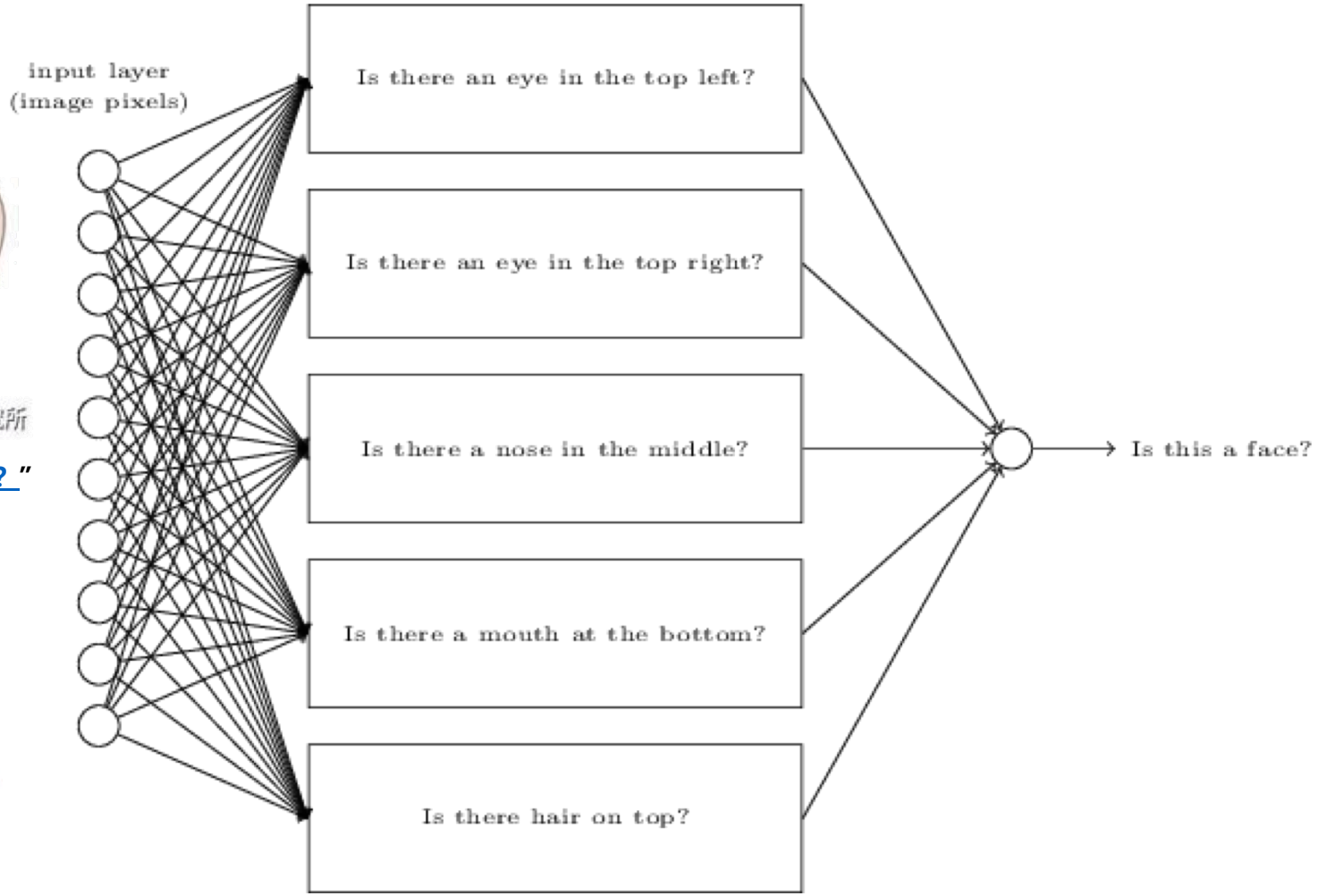


视觉过程——从生物到神经网络



原图 中科院神经科学研究所 公众号文章“集中注意！做个决定！咦，哪来的小松鼠？”





不能更简单的
入门之路

Experiment created by Hao Hu

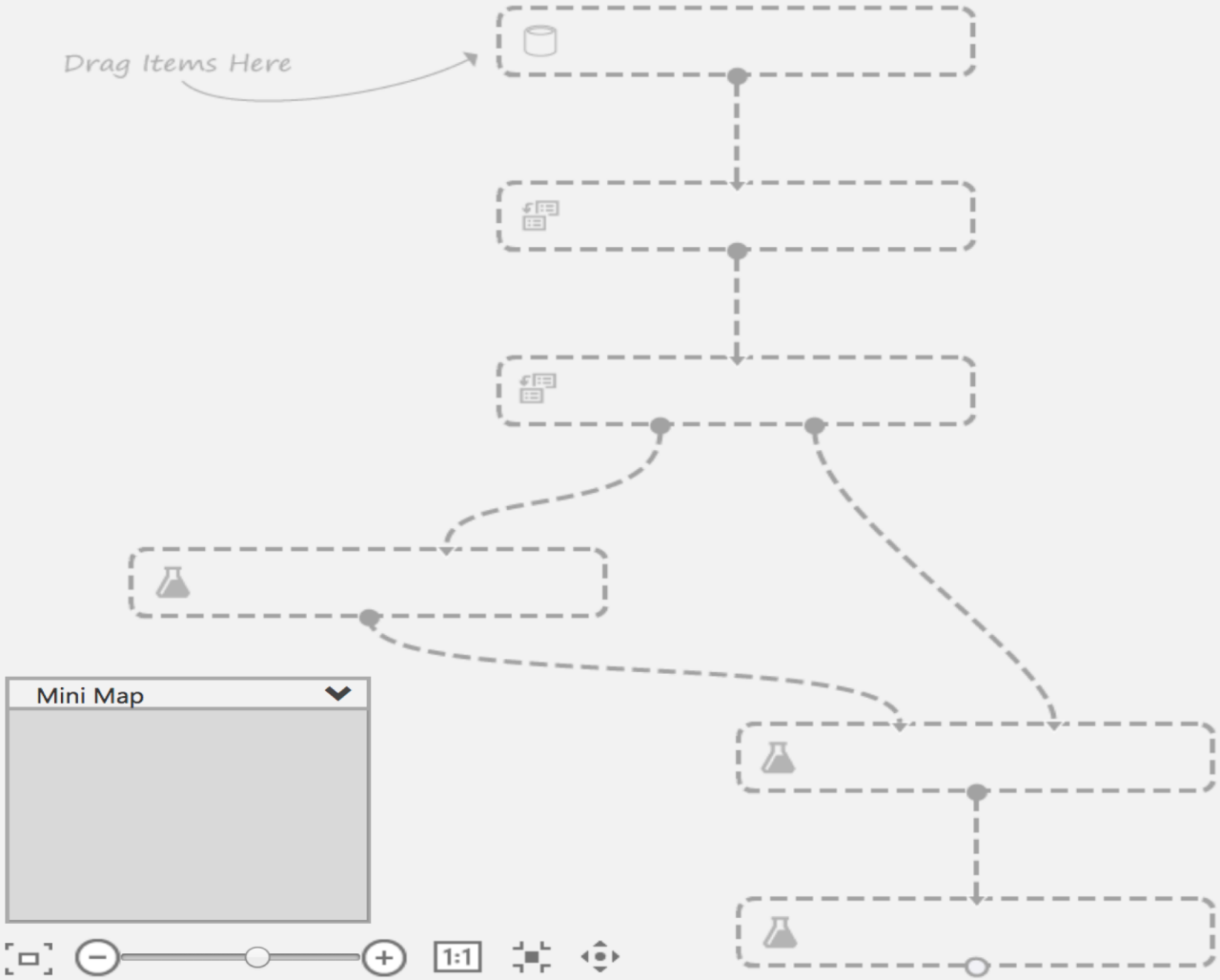
In draft

Search experiment items

- Saved Datasets
- Trained Models
- Transforms
- Data Format Conversions
- Data Input and Output
- Data Transformation
- Feature Selection
- Machine Learning
- OpenCV Library Modules
- Python Language Modules
- R Language Modules
- Statistical Functions
- Text Analytics
- Time Series
- Web Service
- Deprecated

To create your experiment, drag and drop datasets and modules here

Drag Items Here



Properties Project

Experiment Properties

STATUS CO... InDraft

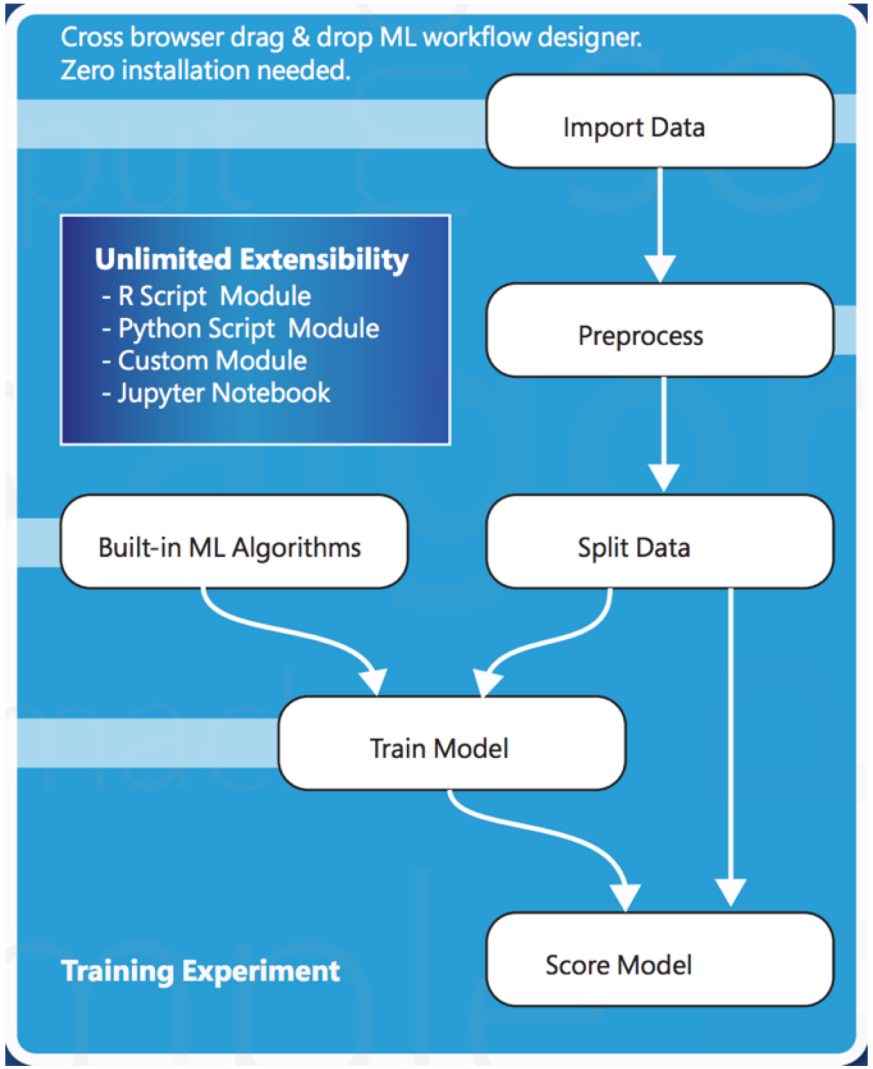
Summary

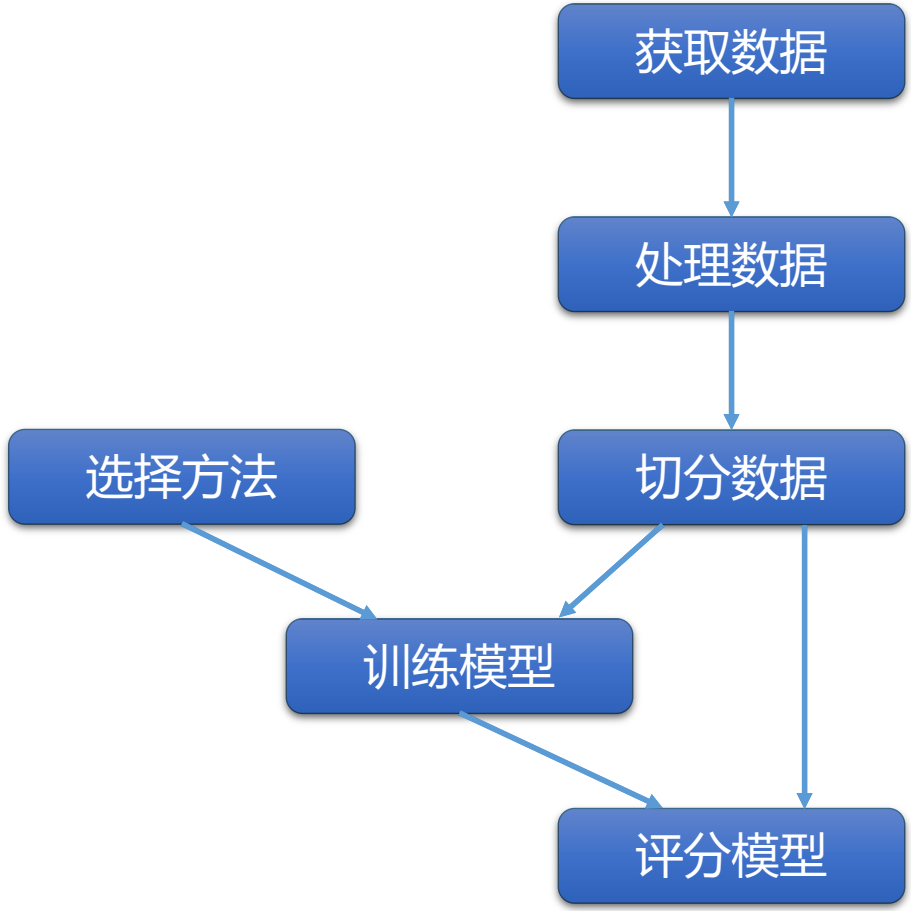
Enter a few sentences describing your experiment (up to 140 characters).

Description

Quick Help

Azure ML Studio的套路





Machine Learning in ML Studio

Anomaly Detection
- One-class Support Vector Machine
- Principal Component Analysis-based Anomaly Detection
- Time Series Anomaly Detection*

Classification

Two-class Classification
- Averaged Perceptron
- Bayes Point Machine
- Boosted Decision Tree
- Decision Forest
- Decision Jungle
- Logistic Regression
- Neural Network
- Support Vector Machine

Multi-class Classification
- Decision Forest
- Decision Jungle
- Logistic Regression
- Neural Network
- One-vs-all

Clustering
- K-means Clustering

Recommendation
- Matchbox Recommender

Regression
- Bayesian Linear Regression
- Boosted Decision Tree
- Decision Forest
- Fast Forest Quantile Regression
- Linear Regression
- Neural Network Regression
- Ordinal Regression
- Poisson Regression

Statistical Functions
- Descriptive Statistics
- Hypothesis Testing T-Test
- Linear Correlation
- Probability Function Evaluation

Text Analytics
- Feature Hashing
- Named Entity Recognition
- Vowpal Wabbit

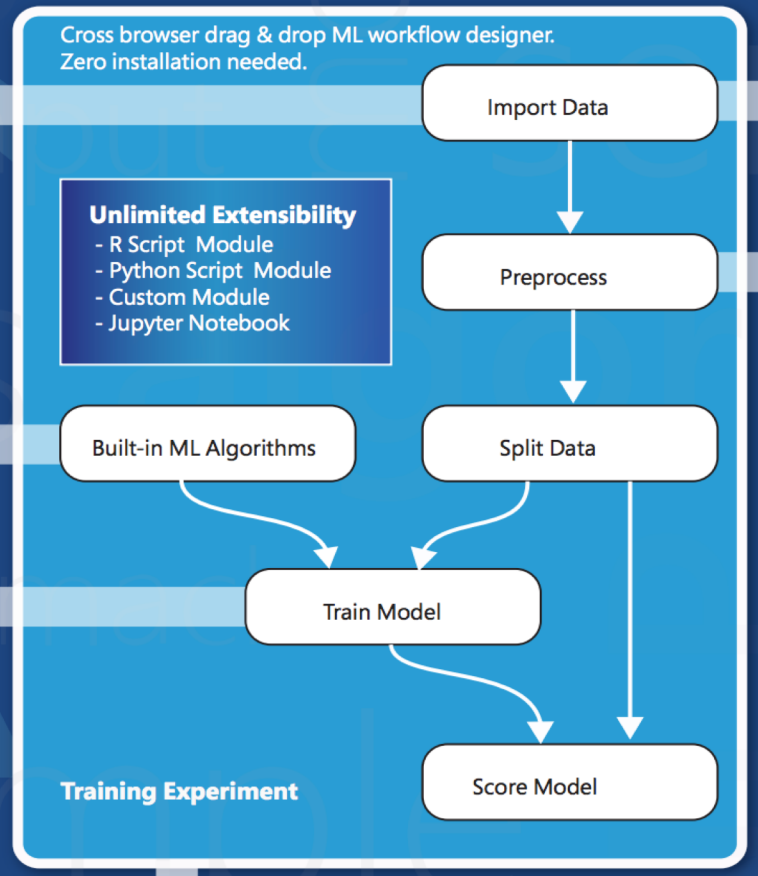
Computer Vision
- OpenCV Library

https://studio.azureml.net

Guest Access Workspace: Free trial access without logging in.
Free Workspace: Free persisted access, no Azure subscription needed.
Standard Workspace: Full access with SLA under an Azure subscription.

Data/Model Visualization
- Scatterplots
- Bar Charts
- Box plots
- Histogram
- R and Python Plotting Libraries
- REPL with Jupyter Notebook
- ROC, Precision/Recall, Lift
- Confusion Matrix
- Decision Tree*

Training
- Cross Validation
- Retraining
- Parameter Sweep



Data Source
- Azure Blob Storage
- Azure SQL DB
- Azure SQL DW*
- Azure Table
- Desktop Direct Upload
- Hadoop Hive Query
- Manual Data Entry
- OData Feed
- On-prem SQL Server*
- Web URL (HTTP)

Data Format
- ARFF
- CSV
- SVMLight
- TSV
- Excel
- ZIP

Data Preparation
- Clean Missing Data
- Clip Outliers
- Edit Metadata
- Feature Selection
- Filter
- Learning with Counts
- Normalize Data
- Partition and Sample
- Principal Component Analysis
- Quantize Data
- SQLite Transformation
- Synthetic Minority Oversampling Technique

Enterprise Grade Cloud Service
- SLA: 99.95% Guaranteed Up-time
- Azure AD Authentication
- Compute at Large Scale
- Multi-geo Availability
- Regulatory Compliance*

One-click Operationalization

Predictive Experiment

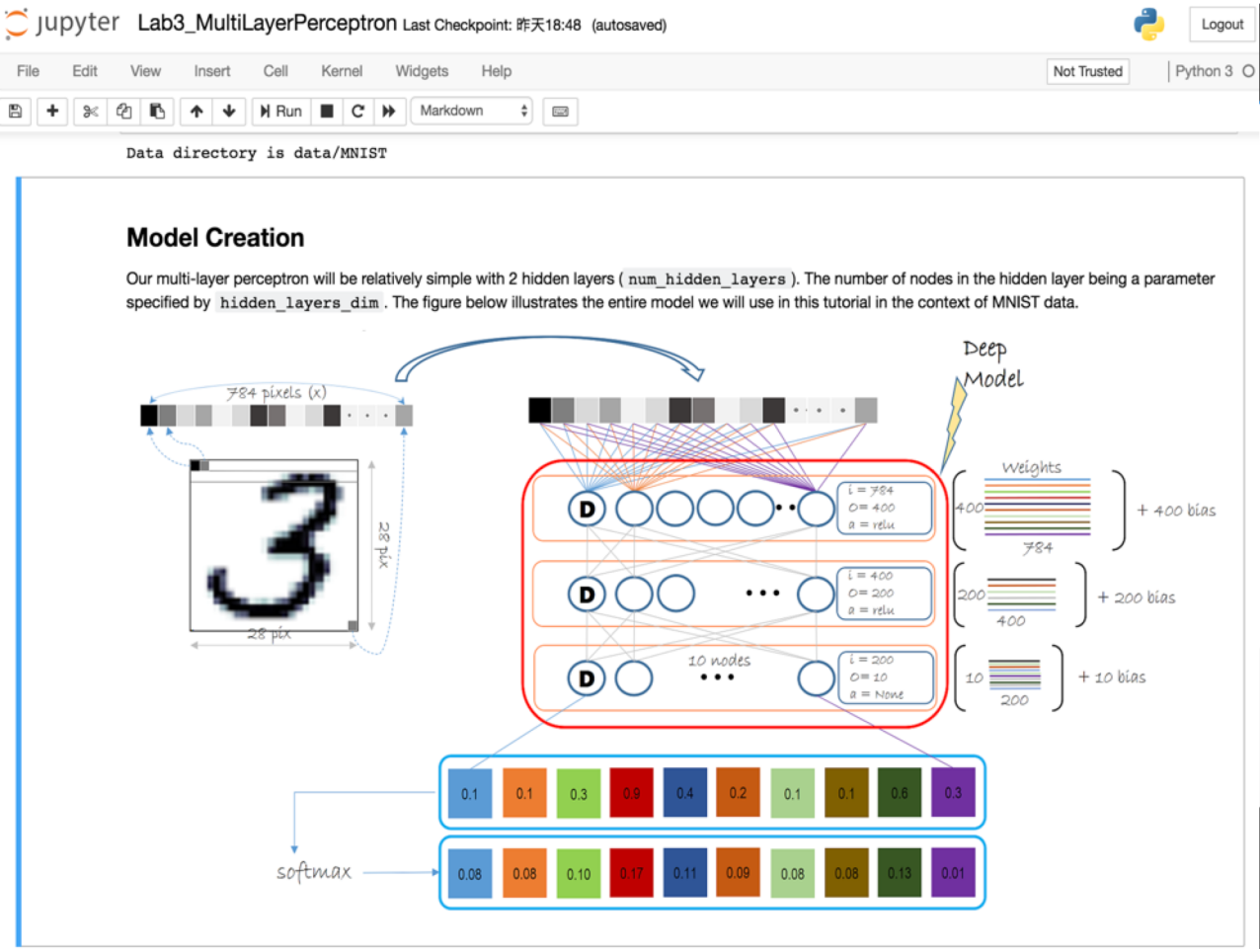
Make Prediction with Elastic APIs
- Request-Response Service (RRS)
- Batch Execution Service (BES)
- Retraining API

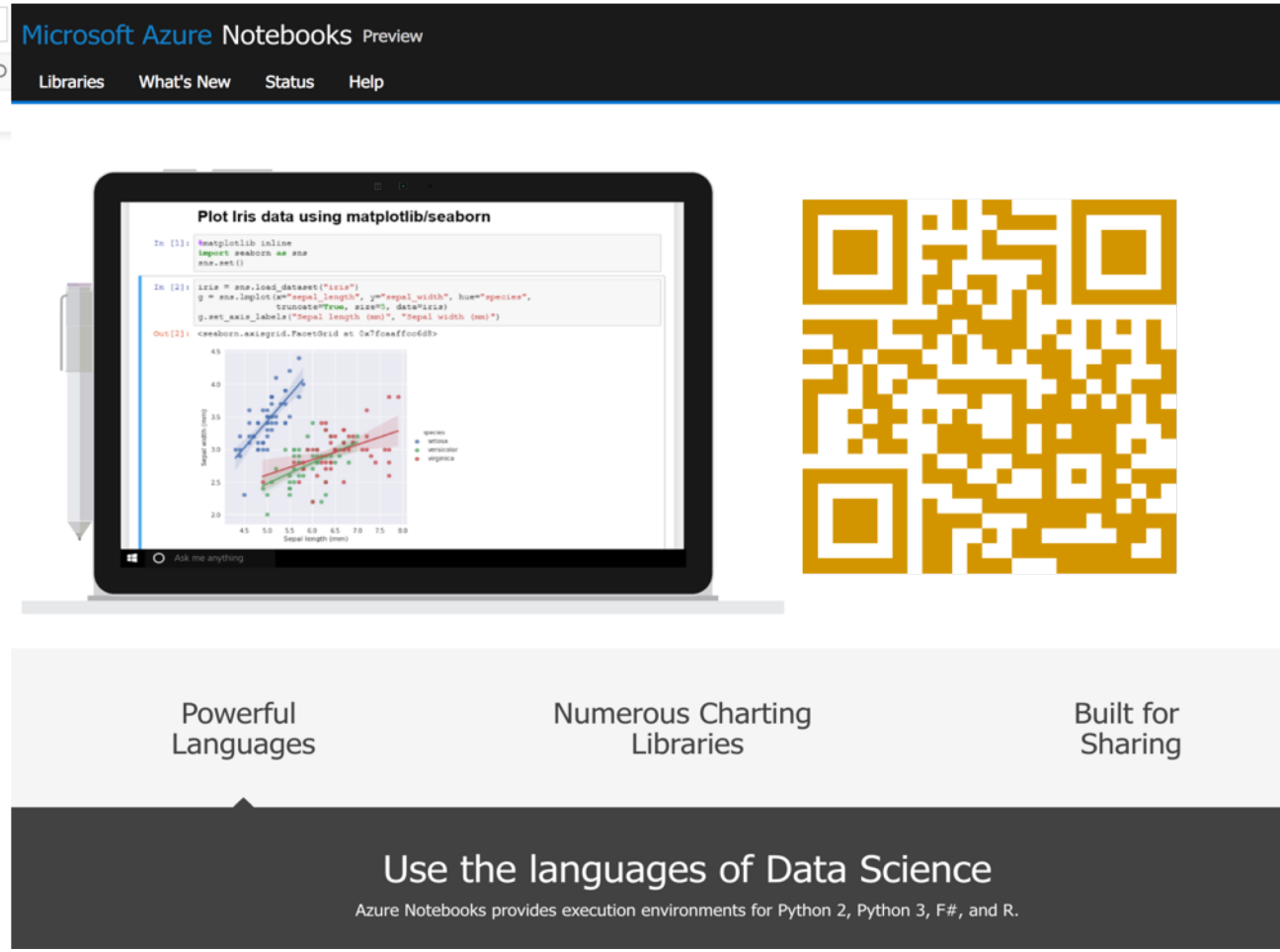
Community
- Gallery (http://gallery.azureml.net)
- Samples & Templates
- Workspace Sharing and Collaboration
- Live Chat & MSDN Forum Support

* Feature Coming Soon

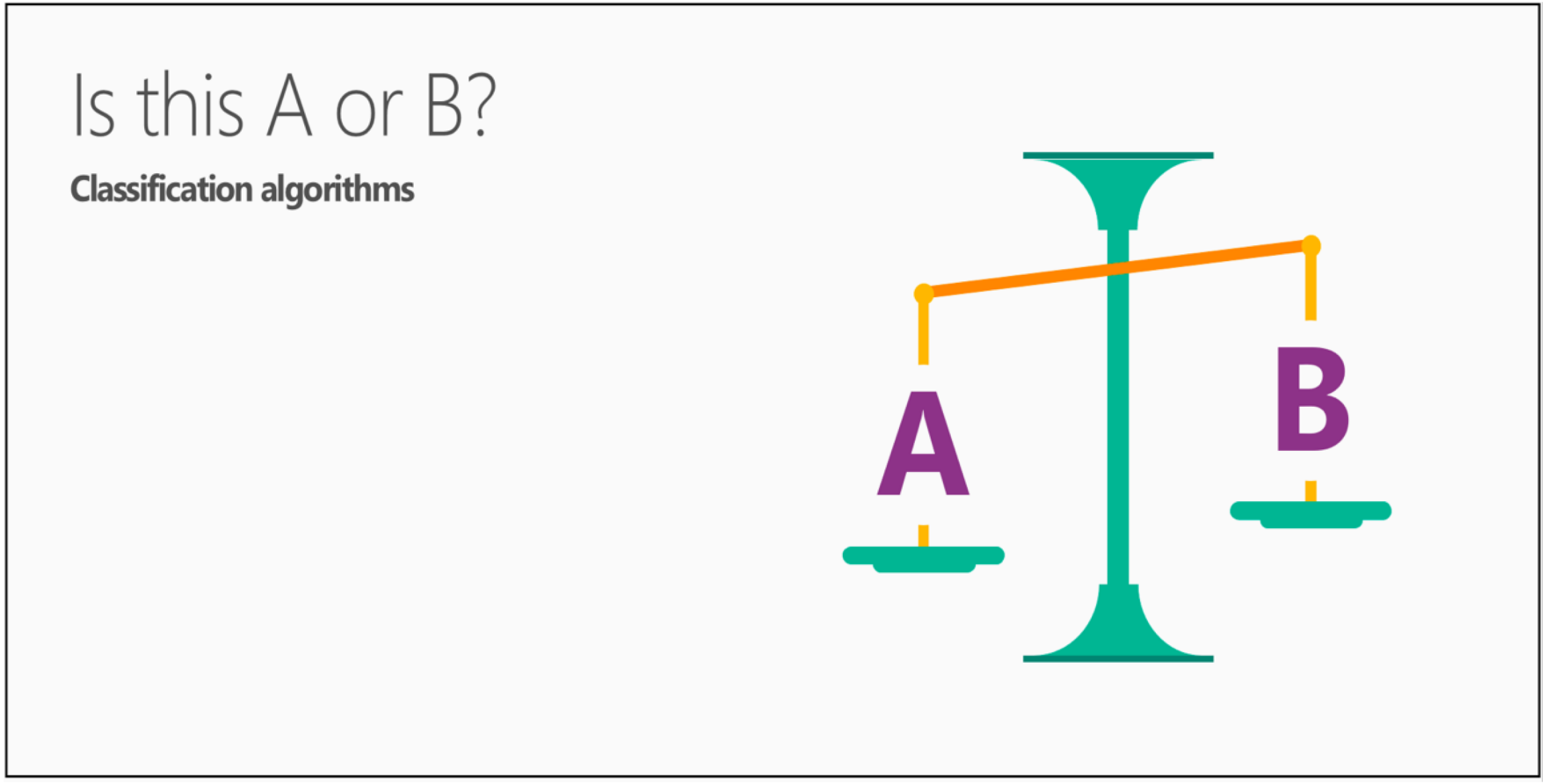
Jupyter Notebooks

如果不愿意自己搭环境，可以使用在线免费的～ CNTK



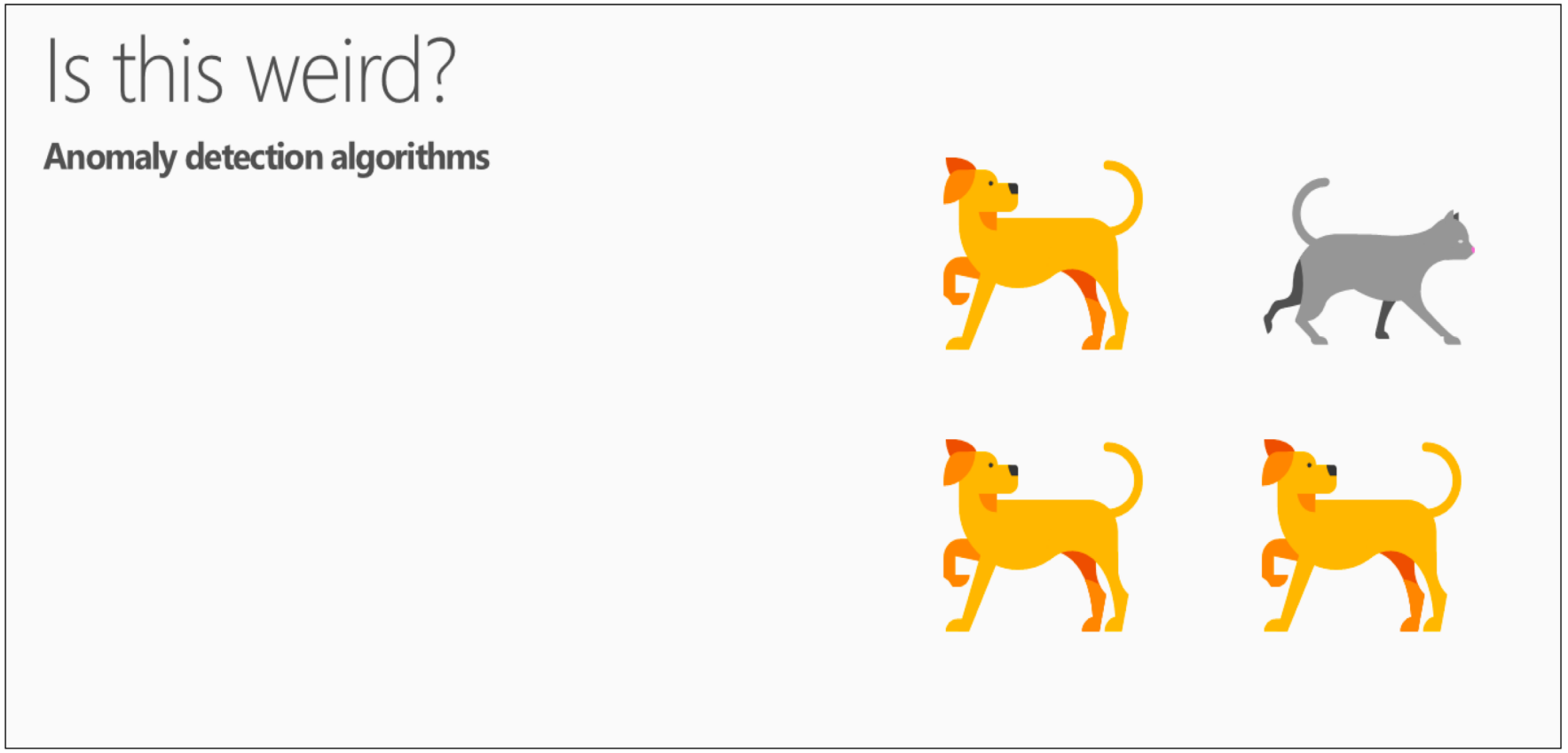


这是A还是B?



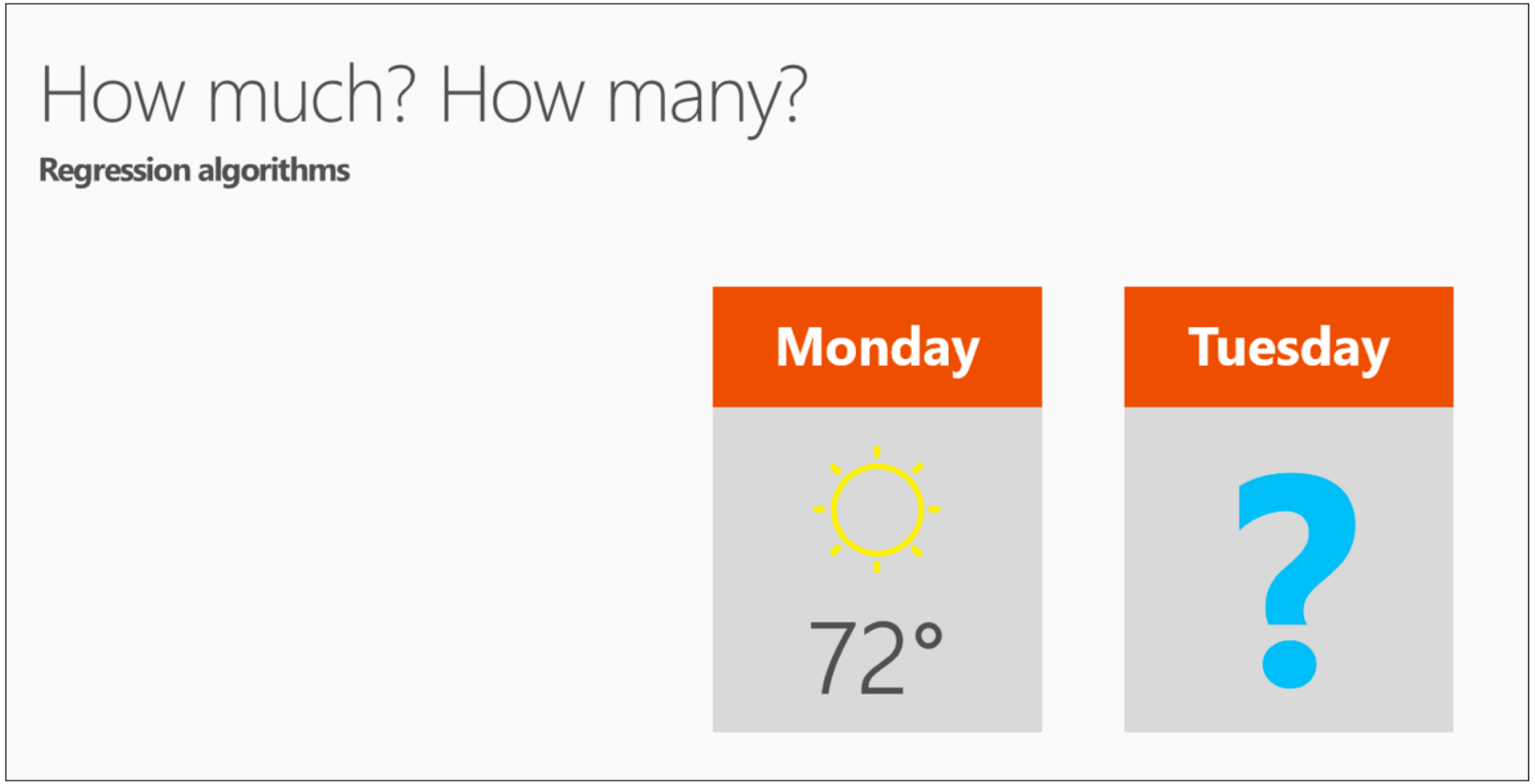
分类

这有点奇怪吗?



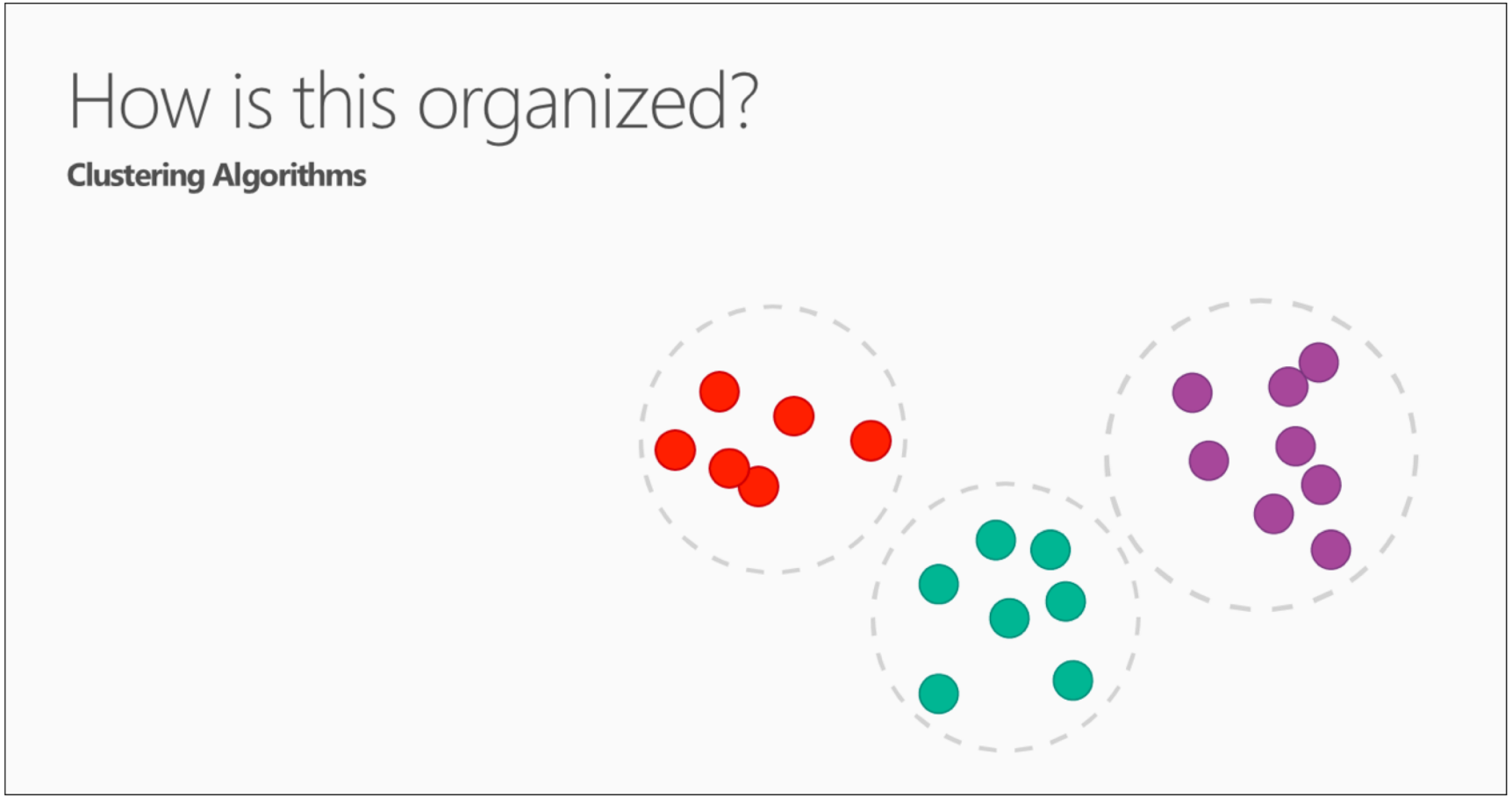
异常检测

有多少？是多少？



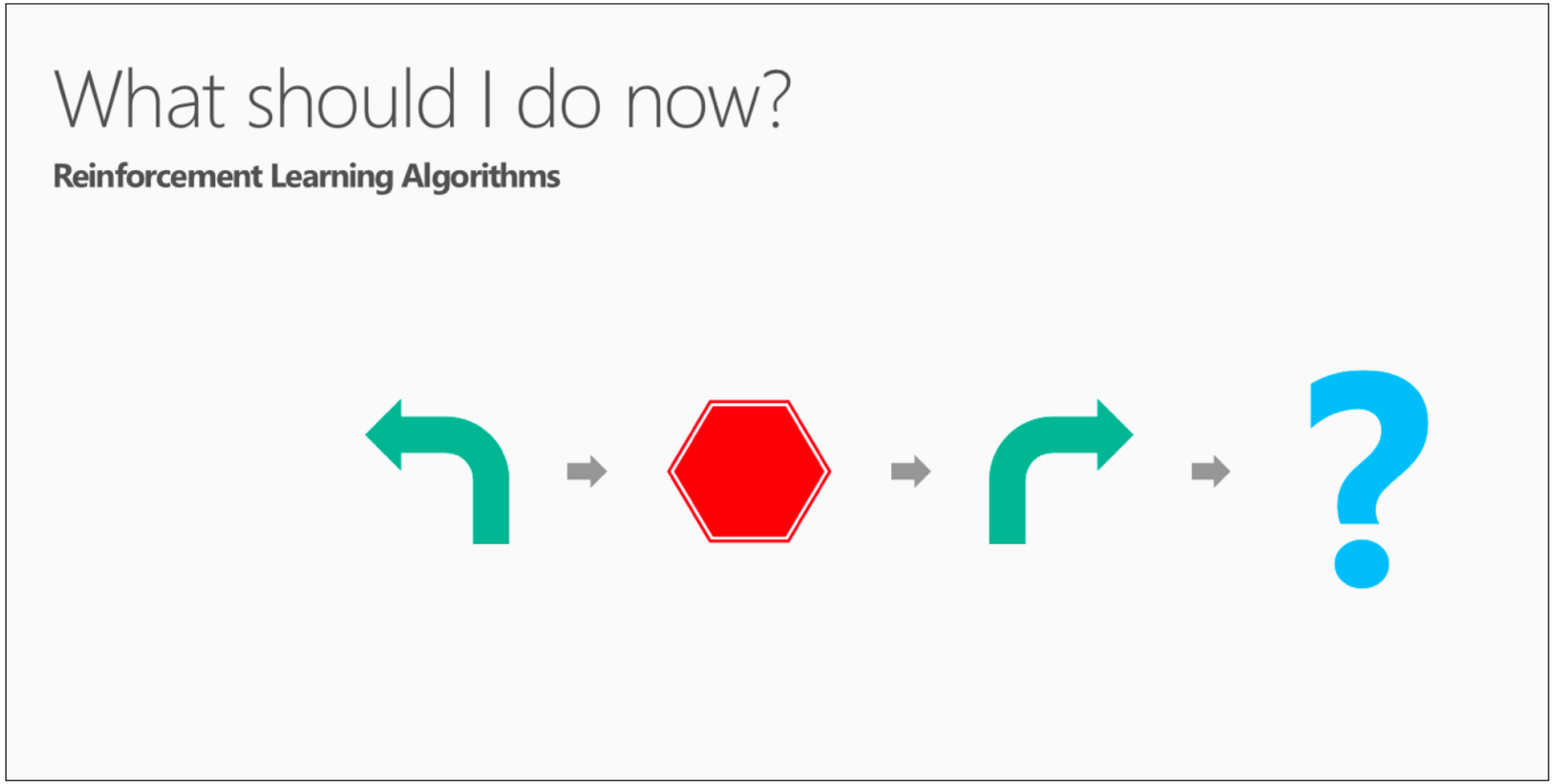
回归

事物是如何组织的?



聚类

我该怎么办？我该做什么？



强化学习

Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.

START

Finding unusual data points

Discovering structure

Predicting categories

Three or more

Two

Predicting values

ANOMALY DETECTION

- **One-class SVM** — >100 features, aggressive boundary
- **PCA-based anomaly detection** — Fast training

CLUSTERING

- **K-means**

MULTI-CLASS CLASSIFICATION

- Fast training, linear model — **Multiclass logistic regression**
- Accuracy, long training times — **Multiclass neural network**
- Accuracy, fast training — **Multiclass decision forest**
- Accuracy, small memory footprint — **Multiclass decision jungle**
- Depends on the two-class classifier, see notes below — **One-v-all multiclass**

REGRESSION

- **Ordinal regression** — Data in rank ordered categories
- **Poisson regression** — Predicting event counts
- **Fast forest quantile regression** — Predicting a distribution
- **Linear regression** — Fast training, linear model
- **Bayesian linear regression** — Linear model, small data sets
- **Neural network regression** — Accuracy, long training time
- **Decision forest regression** — Accuracy, fast training
- **Boosted decision tree regression** — Accuracy, fast training, large memory footprint

TWO-CLASS CLASSIFICATION

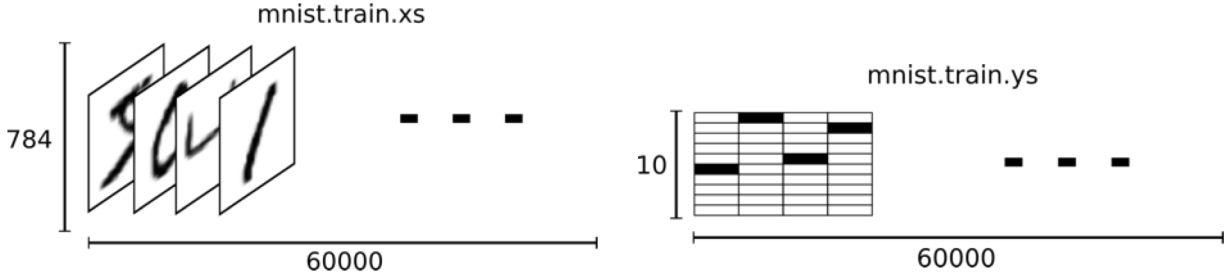
- **Two-class SVM** — >100 features, linear model
- **Two-class averaged perceptron** — Fast training, linear model
- **Two-class logistic regression** — Fast training, linear model
- **Two-class Bayes point machine** — Fast training, linear model

- Accuracy, fast training — **Two-class decision forest**
- Accuracy, fast training, large memory footprint — **Two-class boosted decision tree**
- Accuracy, small memory footprint — **Two-class decision jungle**
- >100 features — **Two-class locally deep SVM**
- Accuracy, long training times — **Two-class neural network**

AI 的
'Hello, World! '

MNIST：AI 的 Hello World

- MNIST数据集
  - 由70,000个手写数字的灰度图像组成
  - 该数据集可在MNIST网站上公开获取
  - 60000行的训练数据集（mnist.train）
  - 10000行的测试数据集（mnist.test）



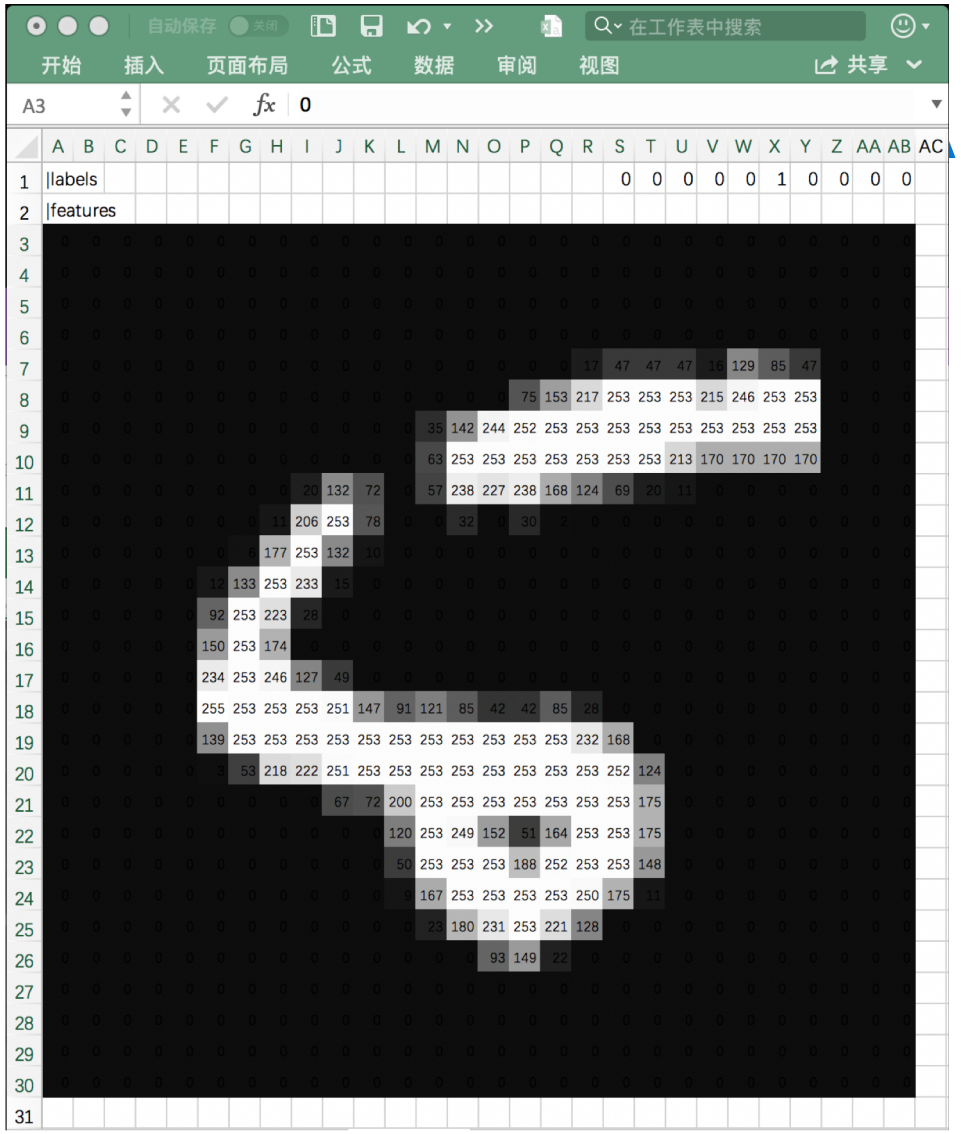
Neural Network: Basic convolution › MNIST Test 10k 28x28 dense › dataset

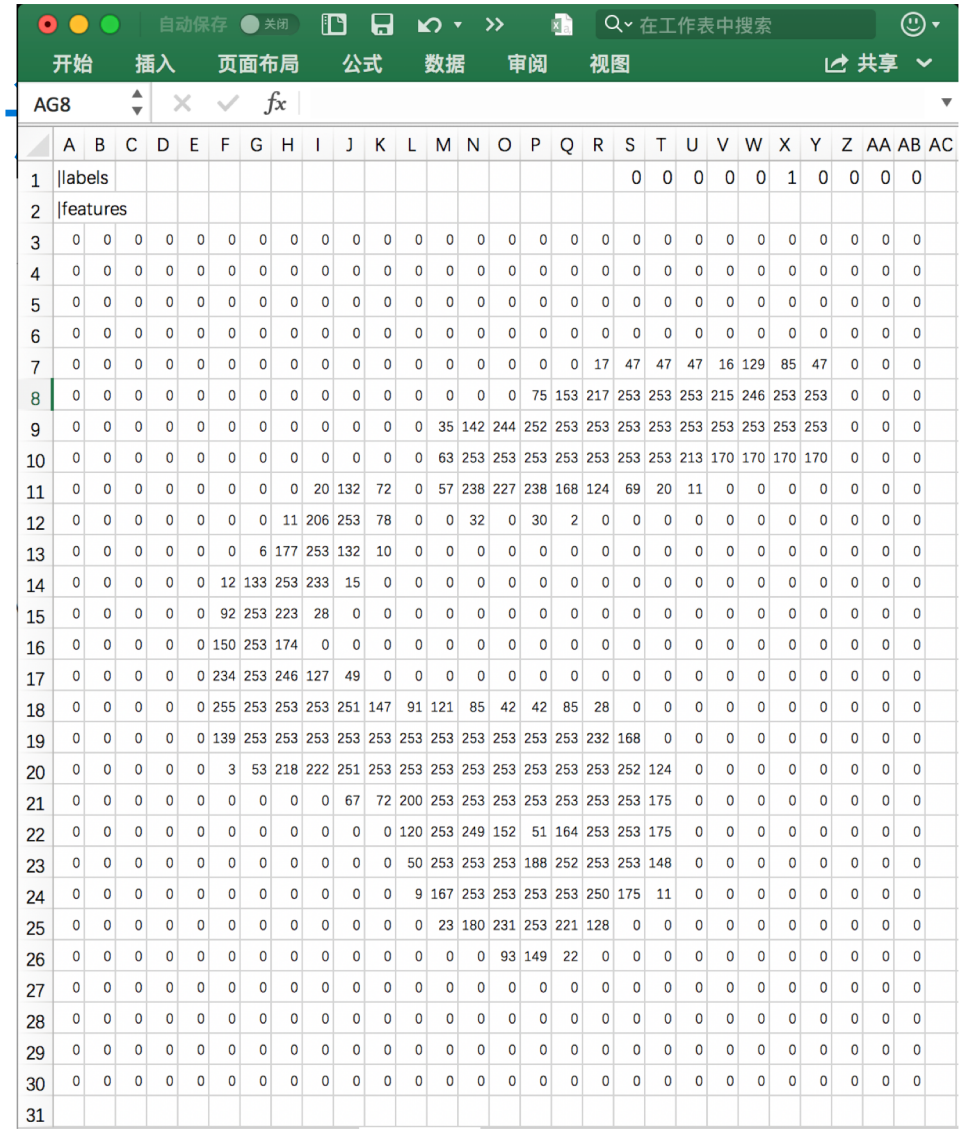
rows 10000 columns 785

Pixels + Label=28x28+1=784+1=785

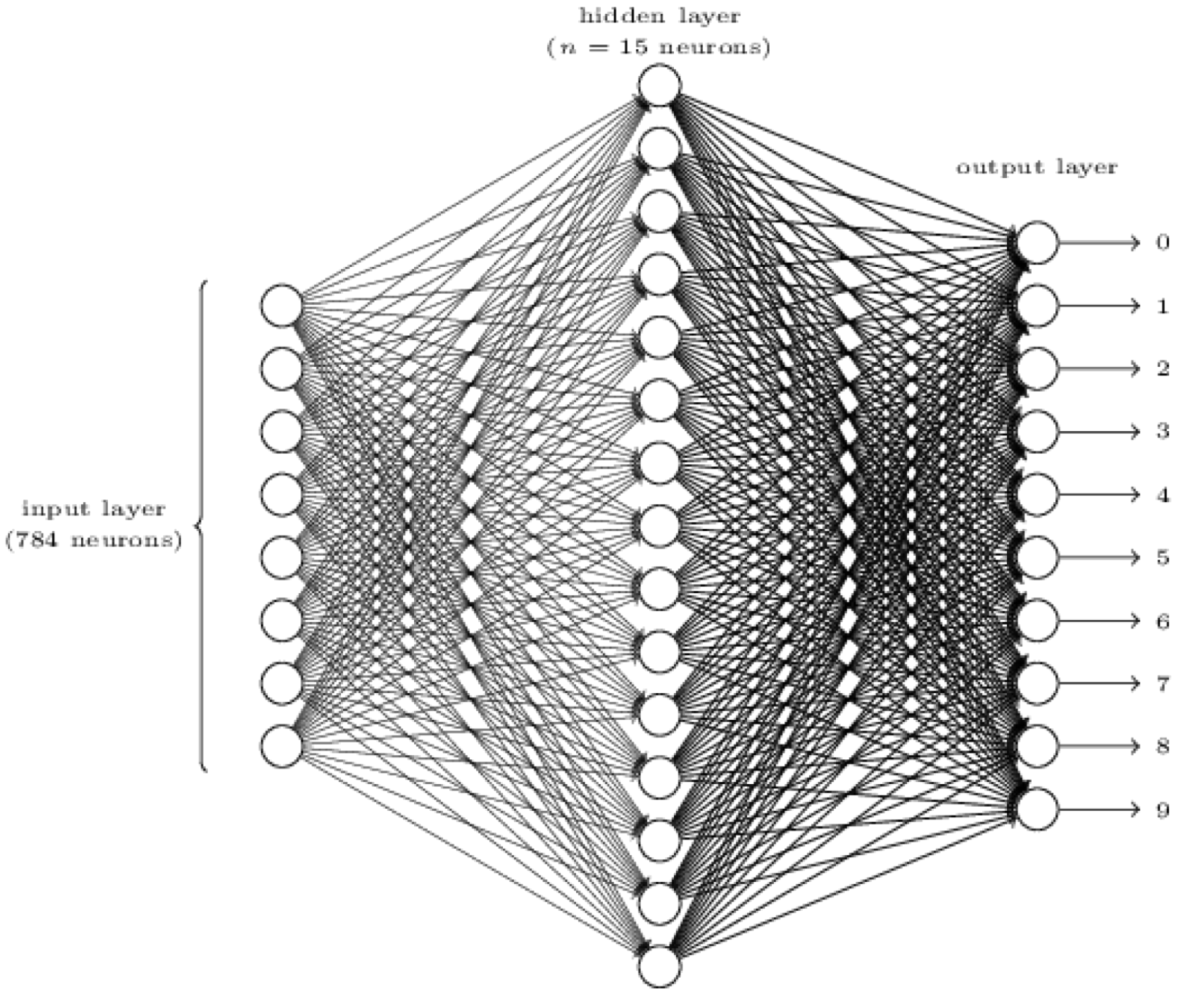
Label	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9
view as										
7	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0

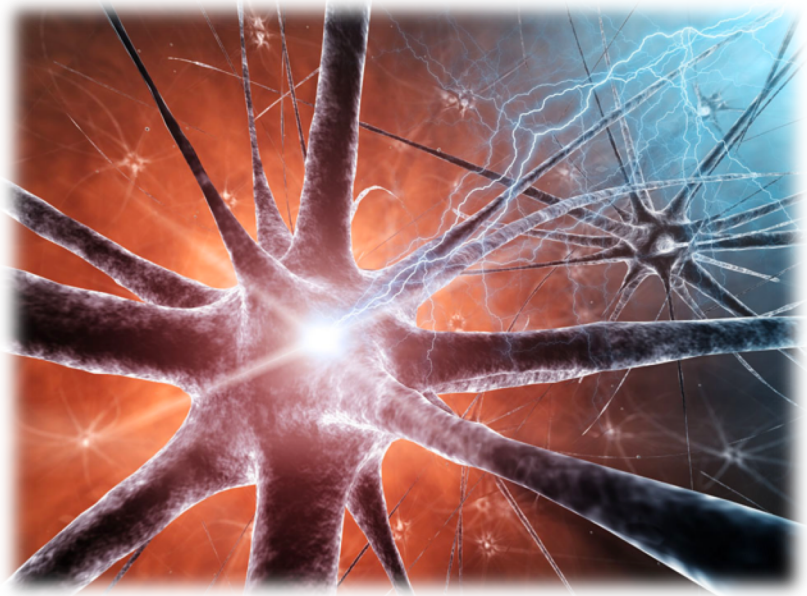
从图像到数字


自动保存
在工作表中搜索
开始
插入
页面布局
公式
数据
审阅
视图
共享
A3
0
|labels
|features


自动保存
在工作表中搜索
开始
插入
页面布局
公式
数据
审阅
视图
共享
AG8
|labels
|features

MNIST 的神经网络示意

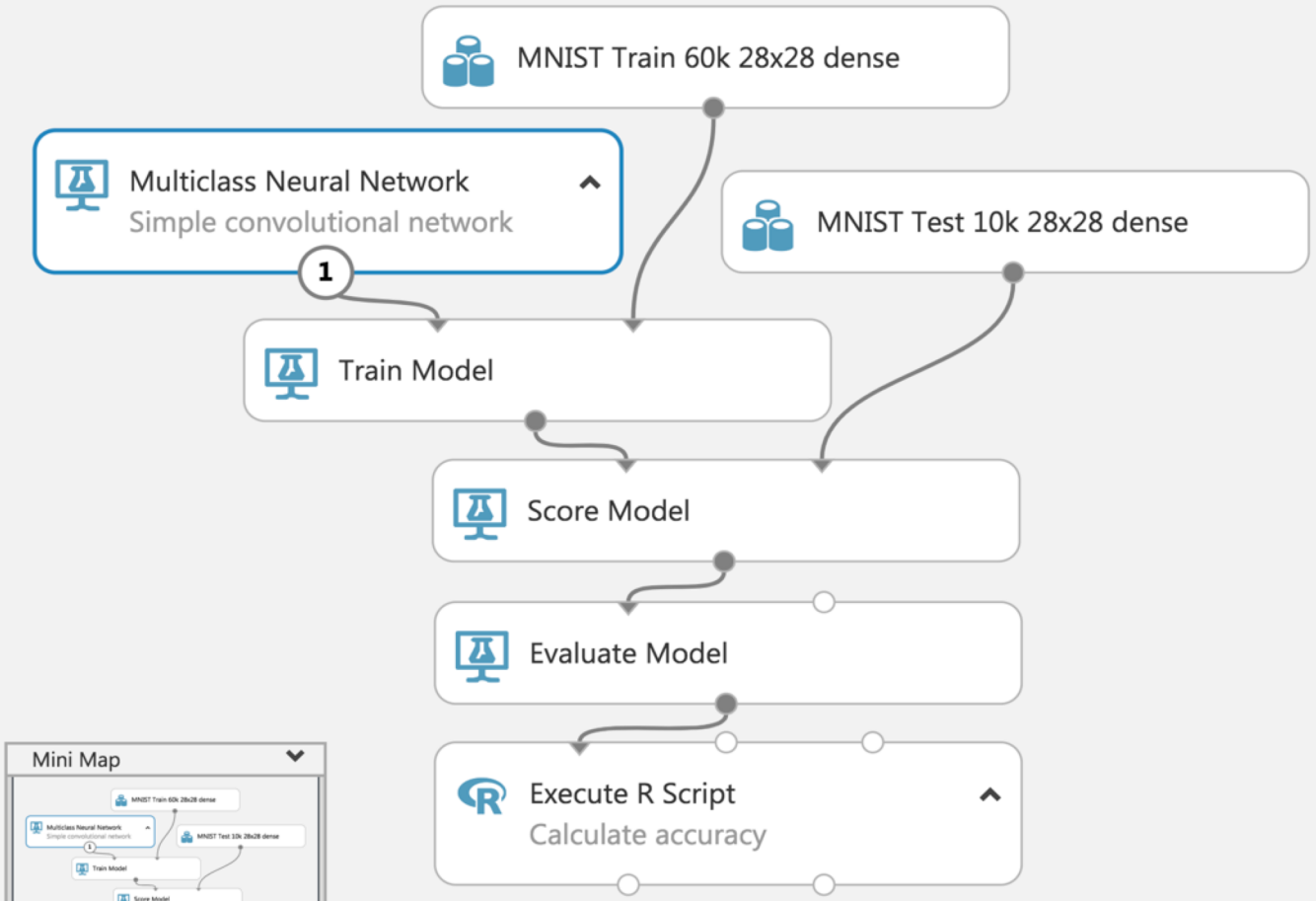




Neural Network: Basic convolution

In draft

Draft saved at 下午5:49:31



Search experiment items

- Saved Datasets
- Trained Models
- Transforms
- Data Format Conversions
- Data Input and Output
- Data Transformation
- Feature Selection
- Machine Learning
- OpenCV Library Modules
- Python Language Modules
- R Language Modules
- Statistical Functions
- Text Analytics
- Time Series
- Web Service
- Deprecated

Properties | Project

Multiclass Neural Network

Create trainer mode: Single Parameter

Hidden layer specification: Custom definition script

Neural network definition

```
const { T = true; F = false; }

input Picture [28, 28];

hidden C1 [5, 12, 12]
  from Picture convolve {
    InputShape  = [28, 28];
    KernelShape = [ 5,  5];
    Stride      = [ 2,  2];
    MapCount = 5;
  }
```

The learning rate: 0.01

Number of learning iterations: 30

The initial learning weights diameter: 1

The momentum: 0

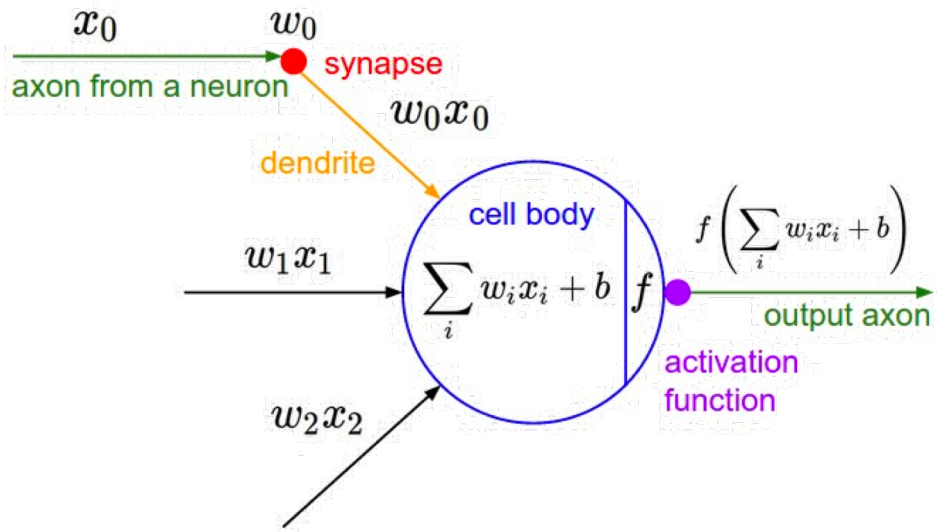
The type of normalizer: Min-Max normalizer

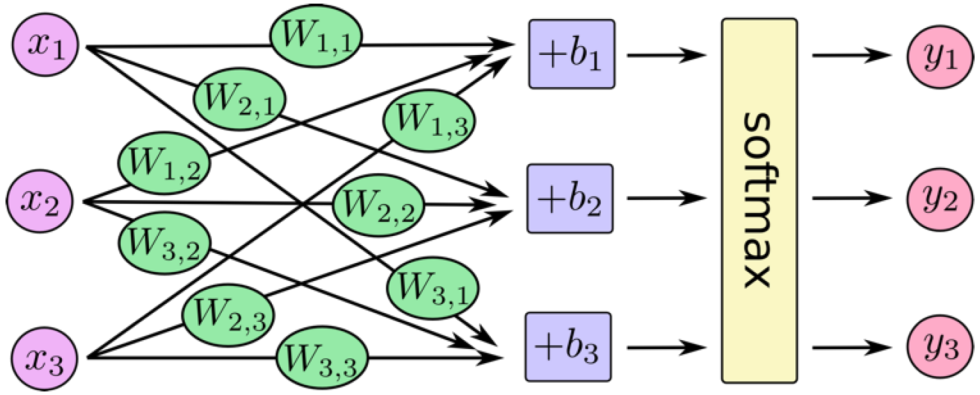
Shuffle examples

Random number seed: 1

Quick Help

模拟神经元计算





$$\text{evidence}_i = \sum_j W_{i,j} x_j + b_i$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{matrix} W_{1,1}x_1 + W_{1,2}x_1 + W_{1,3}x_1 + b_1 \\ W_{2,1}x_2 + W_{2,2}x_2 + W_{2,3}x_2 + b_2 \\ W_{3,1}x_3 + W_{3,2}x_3 + W_{3,3}x_3 + b_3 \end{matrix} \right)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

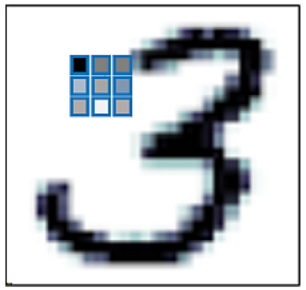
$$y = \text{softmax}(Wx + b)$$

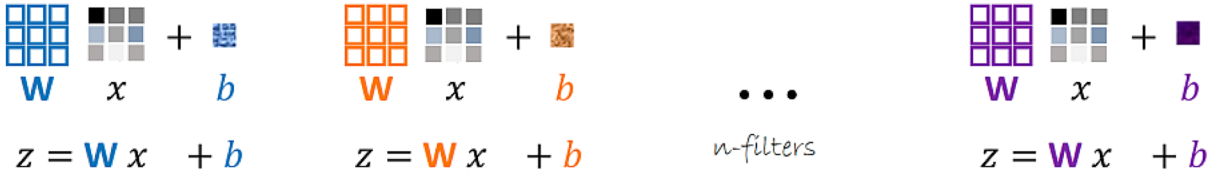
MNIST-双隐藏层网络

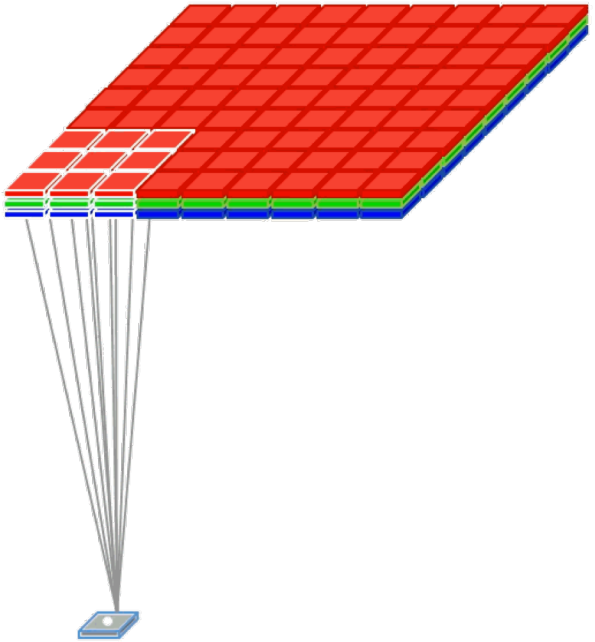
- 输出函数（激活函数）
  - sigmoid
  - linear
  - softmax
  - rlinear
  - square
  - sqrt
  - srlinear
  - abs
  - tanh
  - brlinear

```
input Picture [28,28];
hidden H1 [200] from Picture all;
hidden H2 [200] from H1 all;
output Result [10] softmax from H2 all;
```

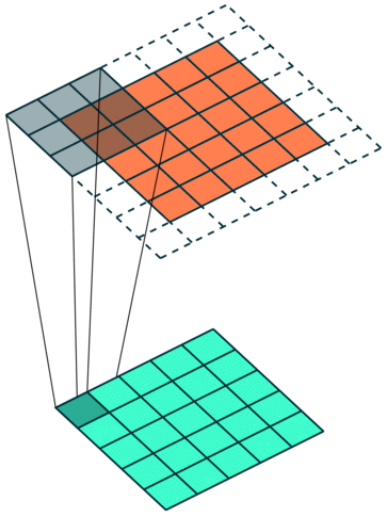
卷积计算



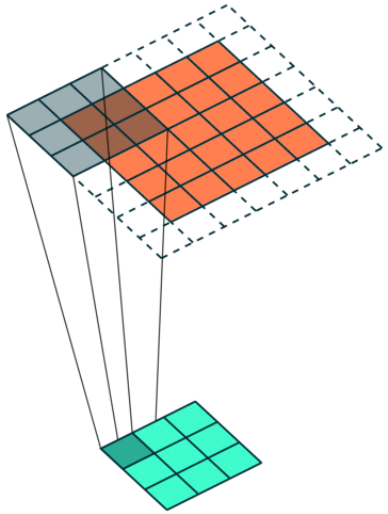




Stride=1



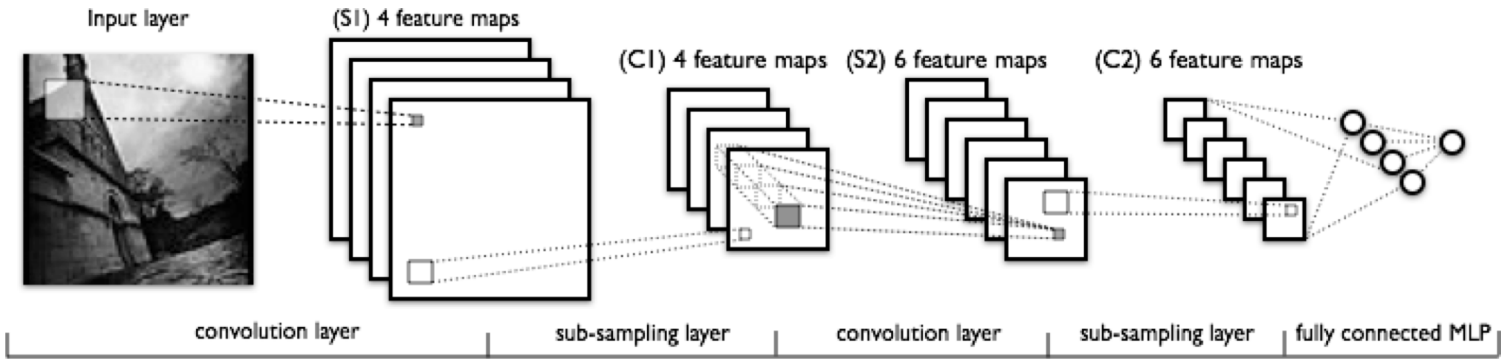
Stride=2

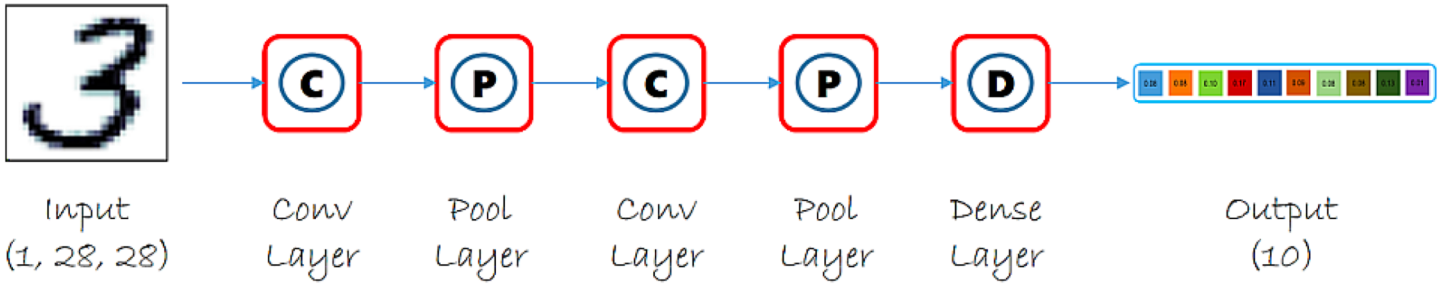


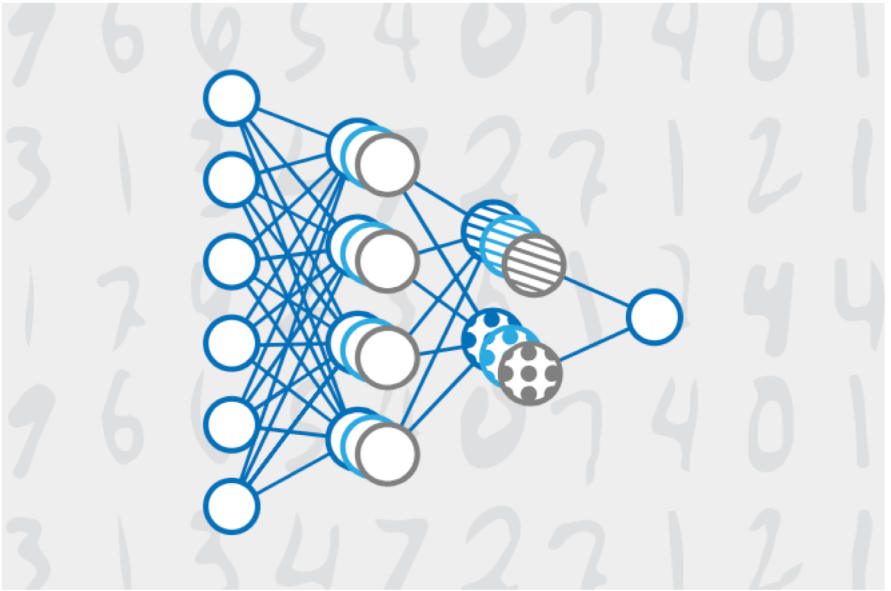
Max Pooling

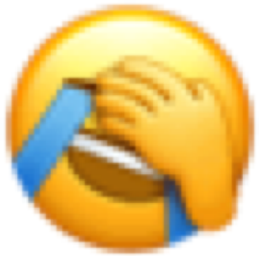
Average Pooling

MNIST-卷积池深度网络









好复杂，我吹不下去了…

真的挺复杂...

```
const { T = true; F = false; }
const {
// input image size
ImgW = 28;
ImgH = 28;
// first convolutional layer parameters
C1Maps = 5;
C1KernW = 5;
C1KernH = 5;
C1StrideW = 1;
C1StrideH = 1;
// The following formula computes dimensions with padding enabled.
C1OutW = (ImgW - 1) / C1StrideW + 1;
C1OutH = (ImgH - 1) / C1StrideH + 1;
// first pooling layer parameters
P1KernW = 2;
P1KernH = 2;
P1StrideW = 2;
P1StrideH = 2;
// The following formula computes dimensions with no padding.
P1OutW = (C1OutW - P1KernW) / P1StrideW + 1;
P1OutH = (C1OutH - P1KernH) / P1StrideH + 1;
// second convolutional layer parameters
C2Maps = 10;
C2KernW = 5;
C2KernH = 5;
C2StrideW = 1;
C2StrideH = 1;
// The following formula computes dimensions with padding enabled.
C2OutW = (P1OutW - 1) / C2StrideW + 1;
C2OutH = (P1OutH - 1) / C2StrideH + 1;
// Since Z dimension of the kernel is 1 and sharing is disabled in Z dimension
// total number of maps is a product of input maps and layer maps.
C2OutZ = C2Maps * C1Maps;
// second pooling layer parameters
P2KernW = 2;
P2KernH = 2;
P2StrideW = 2;
P2StrideH = 2;
// The following formula computes dimensions with no padding.
P2OutW = (C2OutW - P2KernW) / P2StrideW + 1;
P2OutH = (C2OutH - P2KernH) / P2StrideH + 1;
}
input Picture [ImgH, ImgW];
hidden C1 [C1Maps, C1OutH, C1OutW] from Picture convolve {
InputShape  = [ImgH, ImgW];
KernelShape = [C1KernH, C1KernW];
Stride  = [C1StrideH, C1StrideW];
Padding = [T, T];
MapCount = C1Maps;
}
hidden P1 [C1Maps, P1OutH, P1OutW] from C1 max pool {
InputShape  = [C1Maps, C1OutH, C1OutW];
KernelShape = [1, P1KernH, P1KernW];
Stride  = [1, P1StrideH, P1StrideW];
}
hidden C2 [C2OutZ, C2OutH, C2OutW] from P1 convolve {
InputShape  = [C1Maps, P1OutH, P1OutW];
KernelShape = [1, C2KernH, C2KernW];
Stride  = [1, C2StrideH, C2StrideW];
Sharing = [F, T, T];
Padding = [F, T, T];
MapCount = C2Maps;
}
hidden P2 [C2OutZ, P2OutH, P2OutW] from C2 max pool {
InputShape  = [C2OutZ, C2OutH, C2OutW];
KernelShape = [1,  P2KernH, P2KernW];
Stride  = [1,  P2StrideH, P2StrideW];
}
hidden H3 [100] from P2 all;
output Result [10] softmax from H3 all;
```

Net# 神经网络语言

- InputShape
  - 可定义用于此卷积捆绑的源层维数
- KernelShape（必需）
  - 为卷积捆绑定义每个内核的维数
- Stride（可选）
  - 定义卷积的滑动步大小（每维度每步大小）即中央节点之间的距离
- Sharing（可选）
  - 定义卷积的每个维度的权重共享
- MapCount（可选）
  - 为卷积捆绑定义功能映射数
- Weights（可选）
  - 定义捆绑的初始权重

InputShape
>
KernelShape
>
Stride

MVP Community Meetup Shanghai

佛系更新的公众号

用到的手册和代码

包邮区活动微信群

该二维码7天内(10月31日前)有效，重新进入将更新


MVP
Community
Meetup
Shanghai
vă mulţumesc
Salamat sa iyo
Terima kasih
Cảm ơn bạn
Vinaka vakalevu
Níib óolal
Dankie
Gràcies
hvala
Dank u
Grazie
Спасибо
תודה
ευχαριστώ
Thank you
多謝
köszönöm
감사합니다
धन्यवाद
Teşekkür ederiz
Jamädi
Gracias
Merci
谢谢
شكرا
ありがとう
Tak
Aitäh
Dziękuję
Mauruuru ia oe
Danke
Obrigado
dioleh
Благодаря
köszönöm
Ευχαριστούμε
Hvala ti
ขอบคุณ
謝謝
Děkuju
Дякую