动手一天学深度学习

1 基础 · 2 卷积网络 · 3 计算 · 4 计算机视觉

深度学习实训营 2019

李沐

http://1day-zh.d2l.ai

aws

大纲

- 深度学习介绍
- 安装
- 线性代数和张量计算
- 自动求导
- 线性回归
- 优化
- Softmax 回归
- 多层感知机 (训练 MNIST)

http://1day-zh.d2l.ai

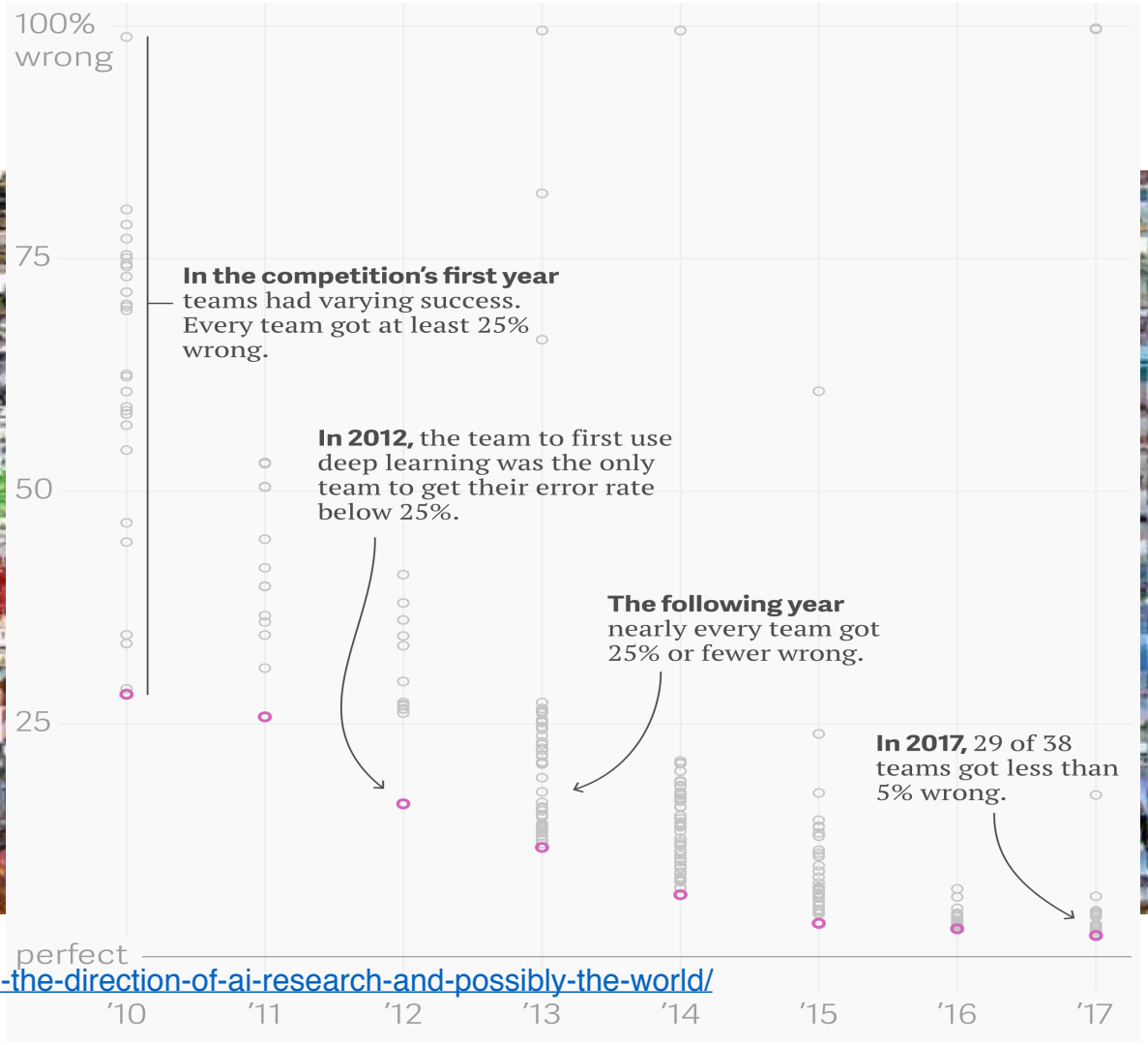

深度学习

图片分类

http://www.image-net.org/

http://1day-zh.d2l.ai

图片分类



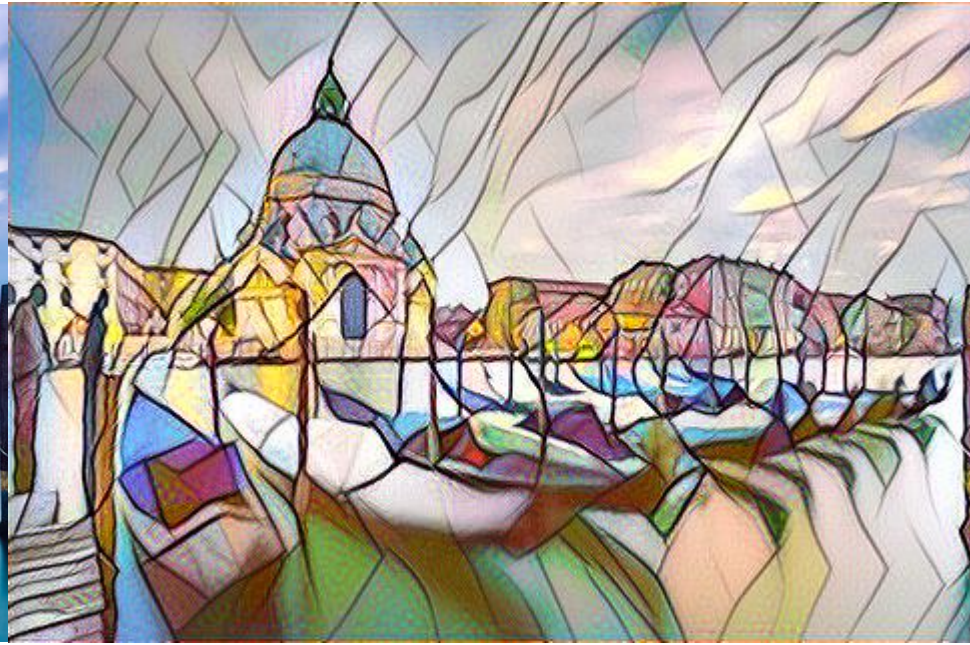
Yanofsky, Quartz
https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/
http://1day-zh.d2l.ai

物体检测和分割

http://1day-zh.d2l.ai

https://github.com/matterport/Mask_RCNN

样式迁移



https://github.com/zhanghang1989/MXNet-Gluon-Style-Transfer/

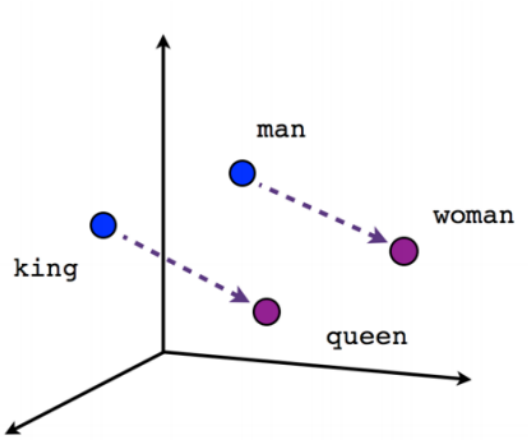
人脸合成

http://1day-zh.d2l.ai

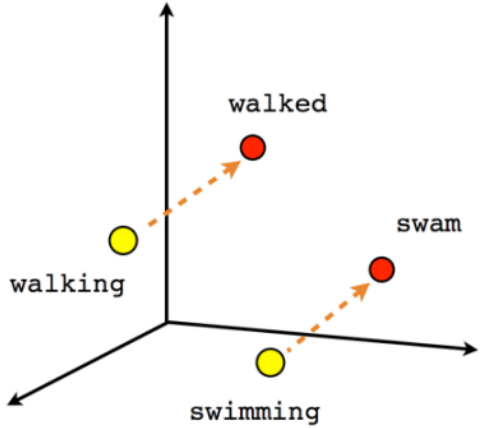
Karras et al, arXiv 2019

aws

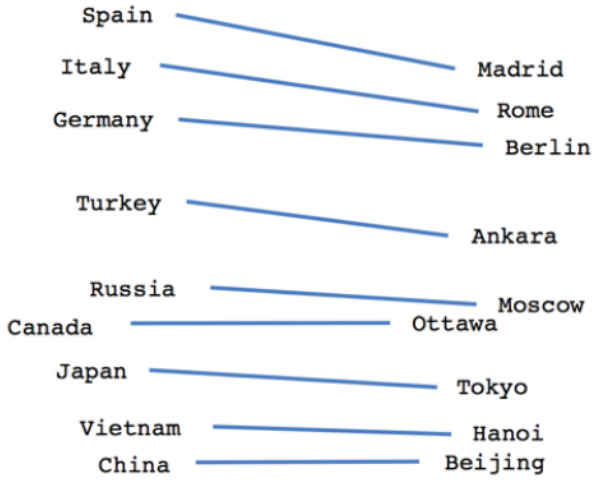
対比



Male-Female



Verb tense

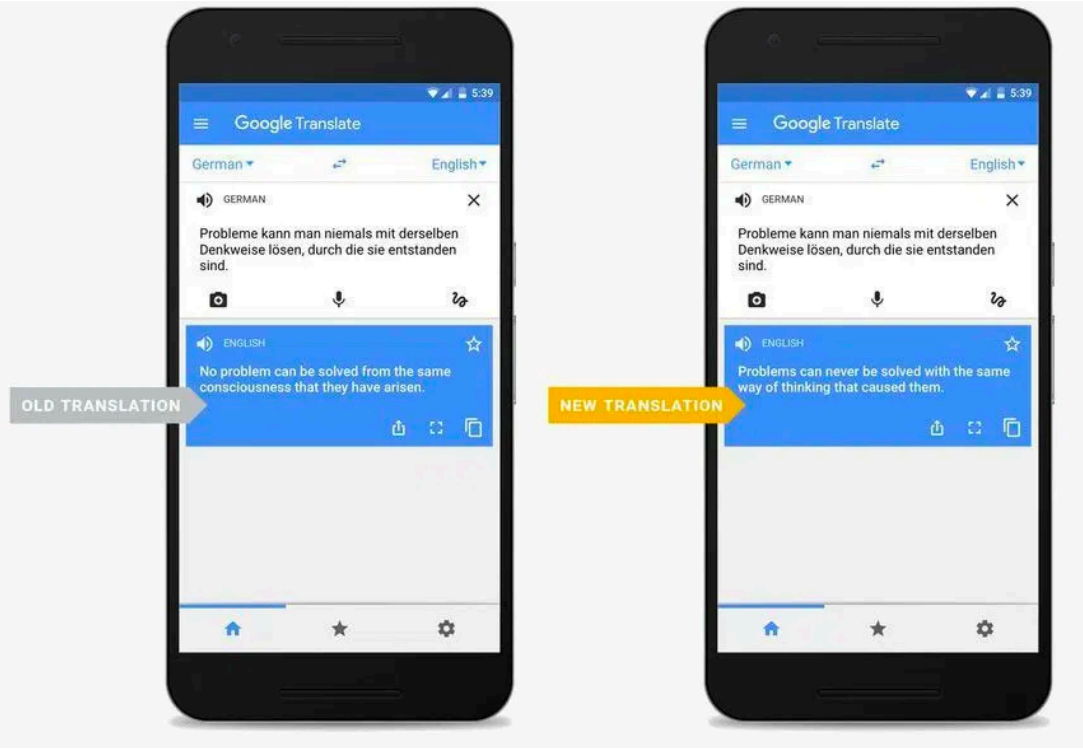


Country-Capital

https://www.tensorflow.org/tutorials/word2vec

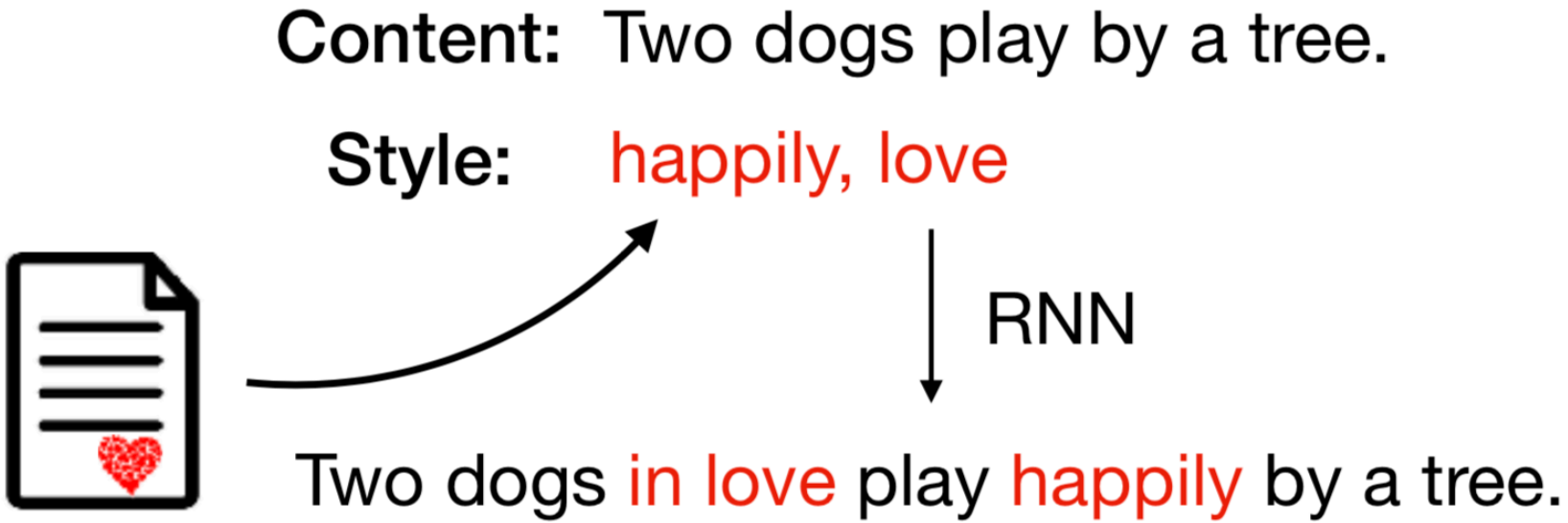
http://1day-zh.d2l.ai

机器翻译



https://www.pcmag.com/news/349610/google-expands-neural-networks-for-language-translation

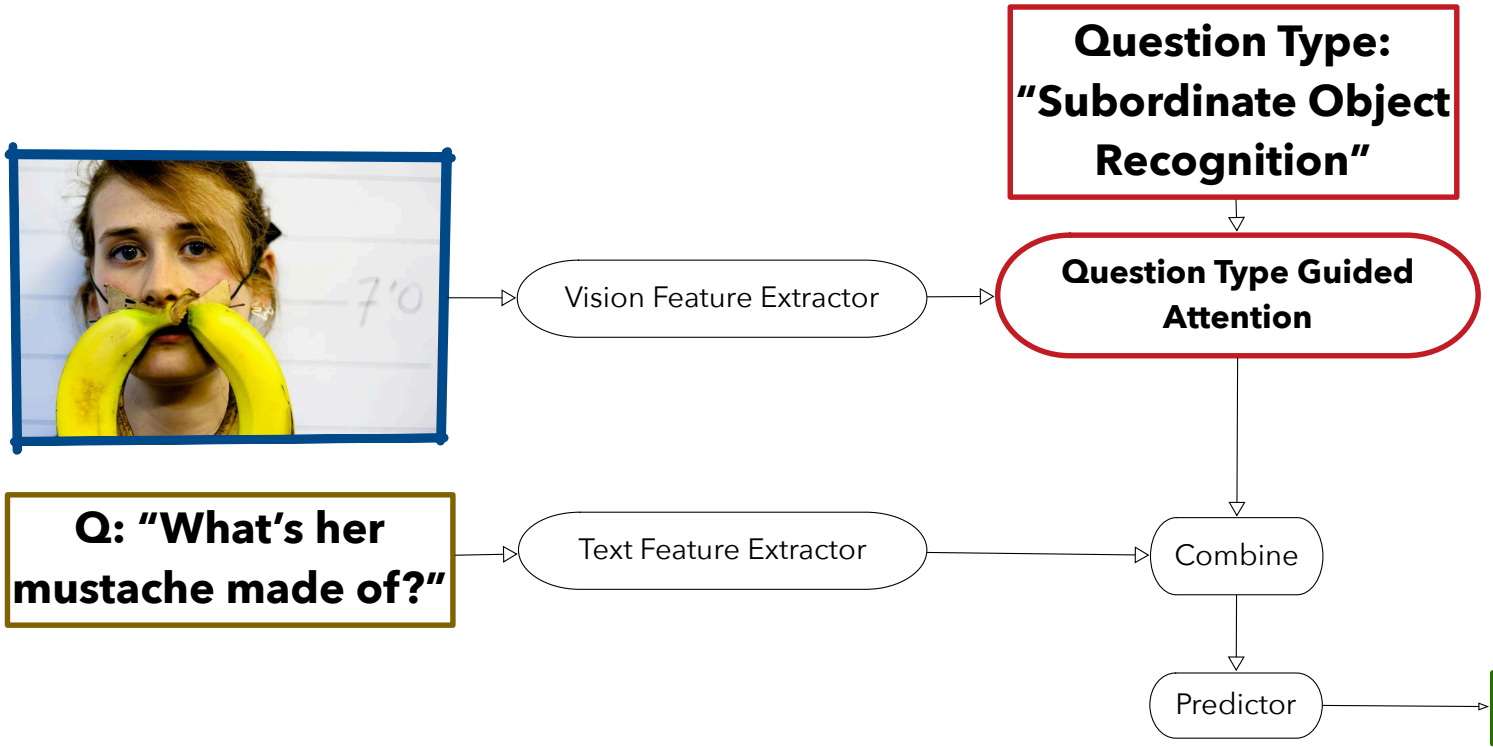
文本生成



Li et al, NACCL, 2018

http://1day-zh.d2l.ai

问答

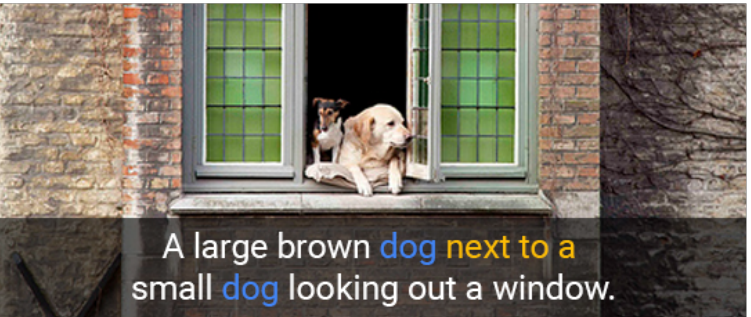


Shi et al, 2018, Arxiv

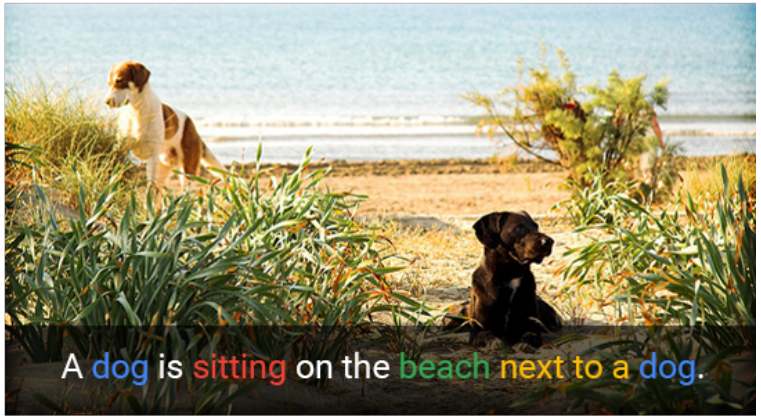
http://1day-zh.d2l.ai

图片文字生成

Human captions from the training set

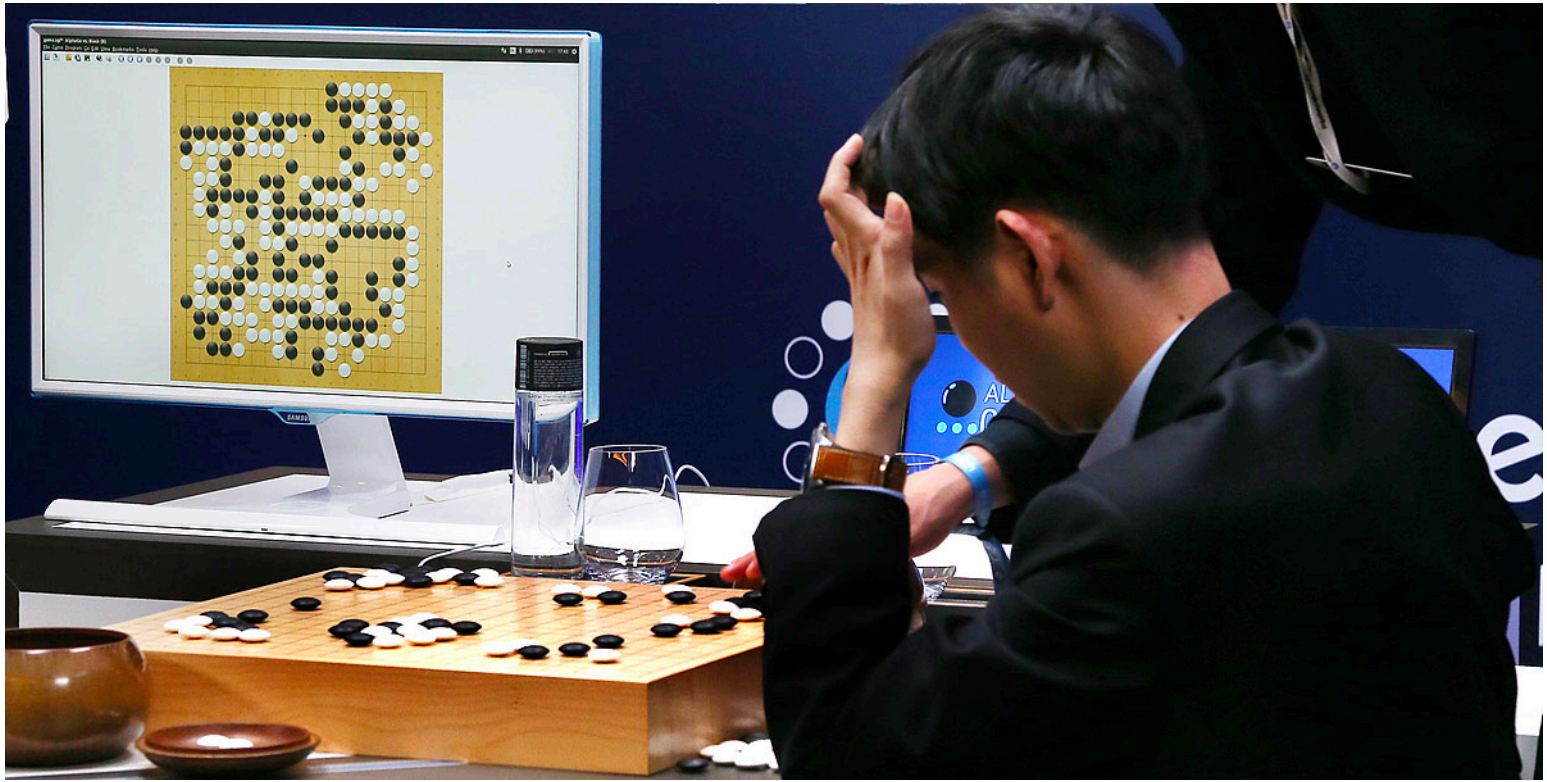


Automatically captioned



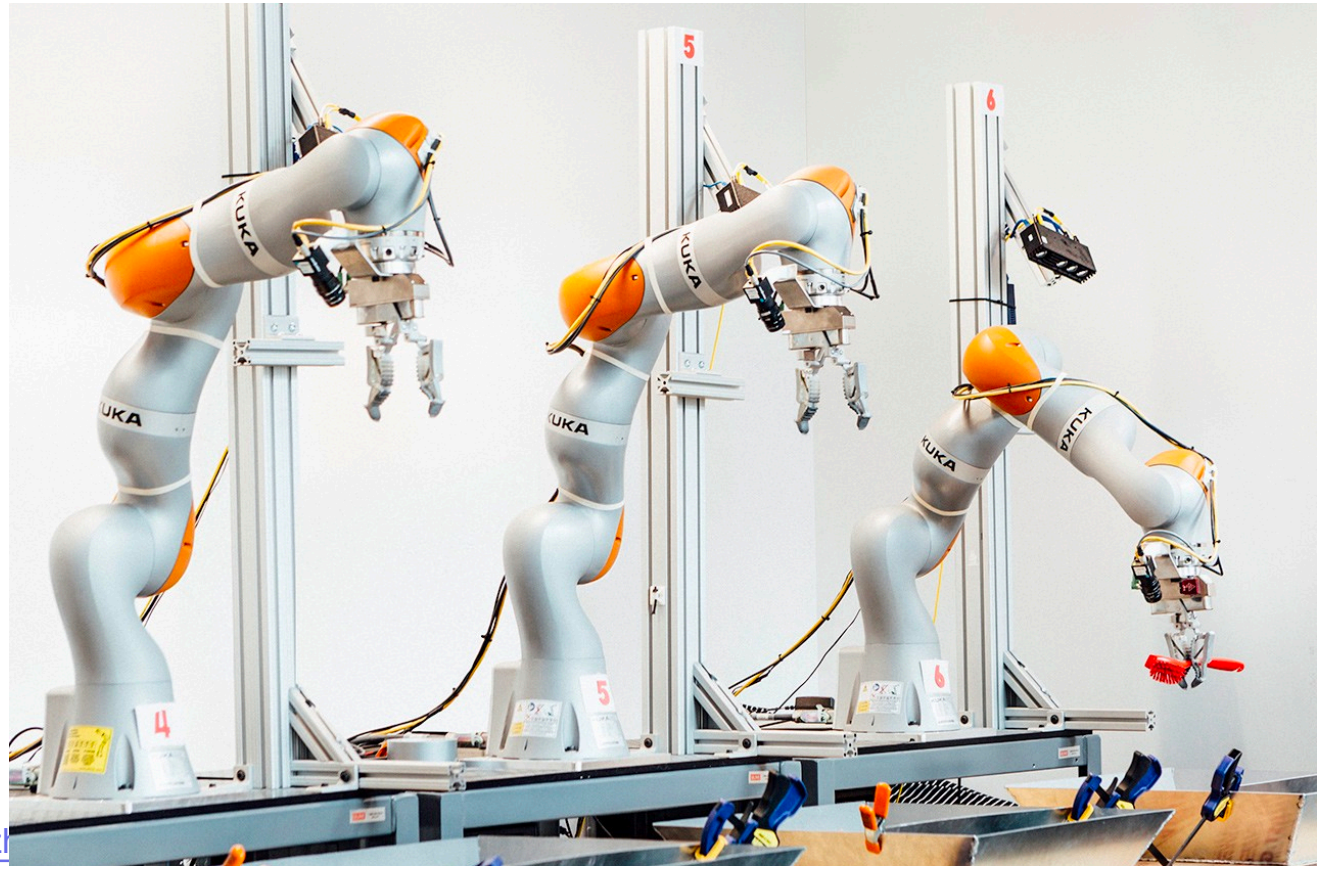
Shallue et al, 2016
https://ai.googleblog.com/2016/09/show-and-tell-image-captioning-open.html

下棋



https://www.technologyreview.com/s/604273/finding-solace-in-defeat-by-artificial-intelligence/

机器人手臂控制



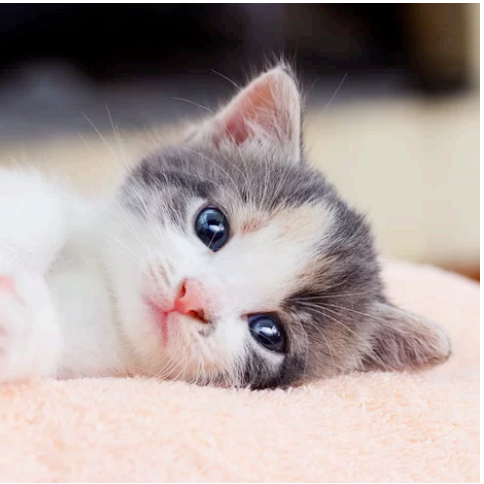
http://1day-zl

无人驾驶


LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
MOTION FLOW
LANE LINES
LANE LINES
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

http://

我们今天要解决的问题：分类



猫

狗

兔子

仓鼠

给定图片 x 来估计标号 y

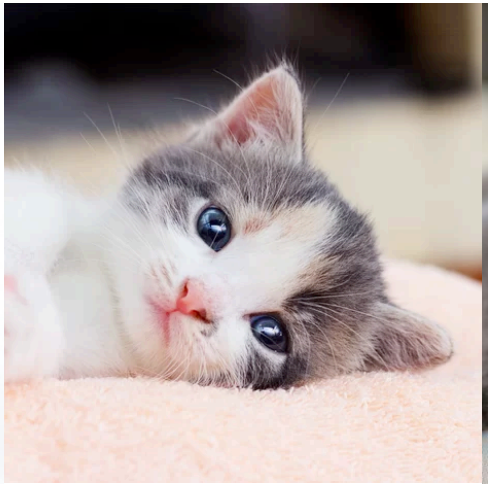
$$y = f(x) \text{ 这里 } y \in \{1, \ldots N\}$$

http://1day-zh.d2l.ai

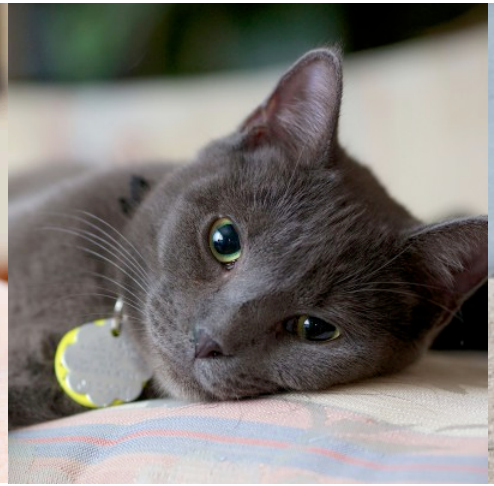
aws

我们今天将解决的问题：回归

0.4kg



2kg



4kg

10kg

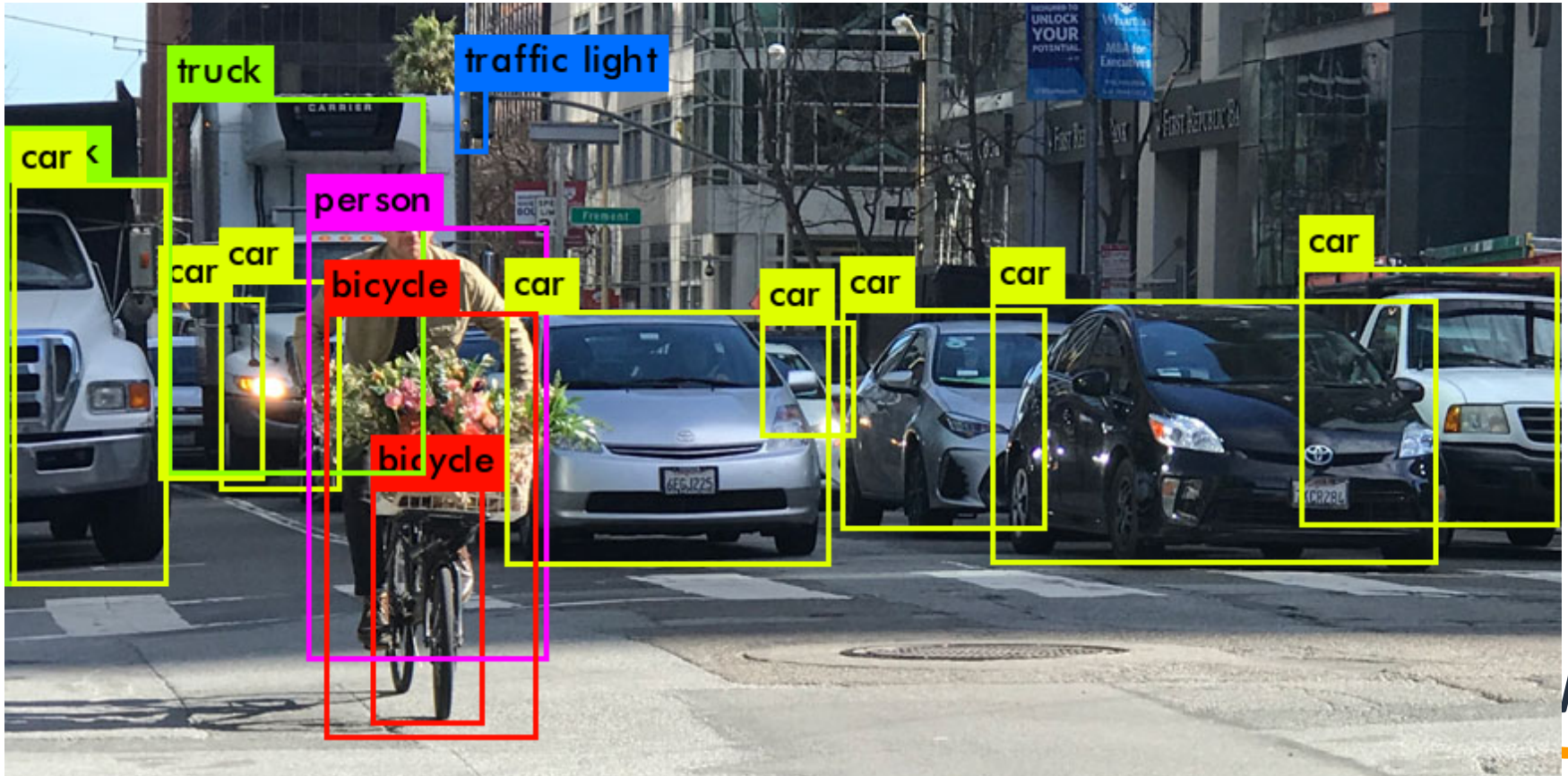
给定图片 x 来估计标号 y

$$y = f(x) \text{ 这里 } y \in \mathbb{R}$$

http://1day-zh.d2l.ai

aws

我们今天将解决的问题： 检测




安装

工具

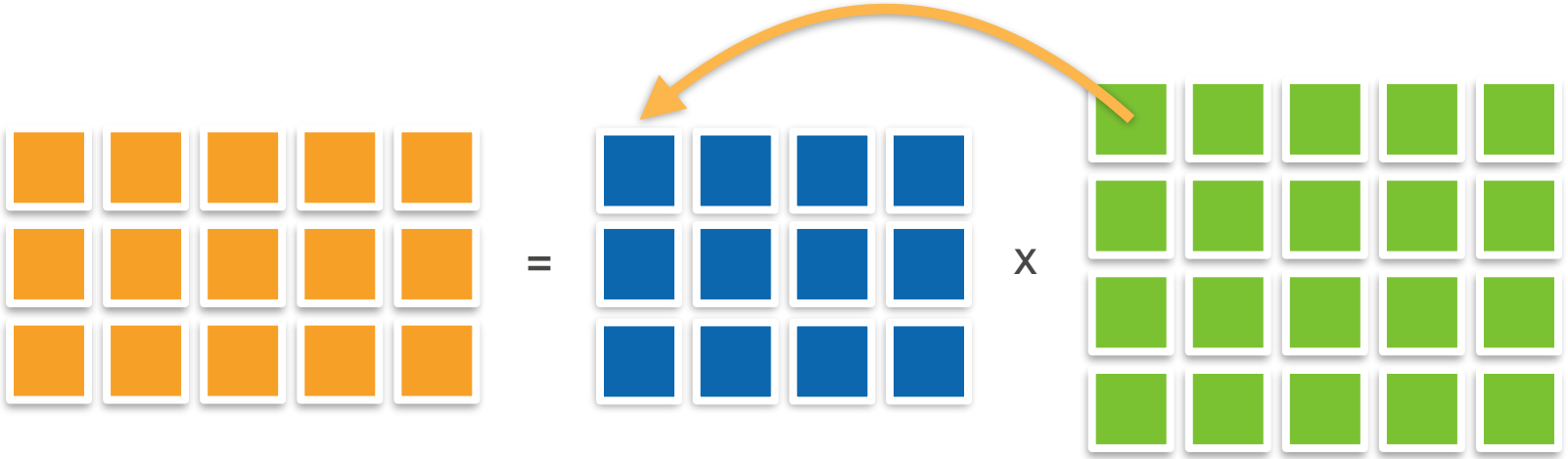
- Python
- Jupyter
- MXNet
  - 基于 nightly build 来使用 DeepNumpy
- D2L
  - 基于英文版的新NumPy版本 http://numpy.d2l.ai/

获取 GPU 实例

- 发送邮件到

- 等待几分钟后会收到邮件，点击链接即可

线性代数



http://1day-zh.d2l.ai

标量

- 操作

$$c = a + b$$

$$c = a \cdot b$$

$$c = \sin a$$

- 长度

$$|a| = \begin{cases} a & \text{if} a > 0 \\ -a & \text{otherwise} \end{cases}$$

$$|a + b| \leq |a| + |b|$$

$$|a \cdot b| = |a| \cdot |b|$$

向量

- 操作

$$c = a + b \quad \text{where } c_i = a_i + b_i$$
$$c = \alpha \cdot b \quad \text{where } c_i = \alpha b_i$$
$$c = \sin a \quad \text{where } c_i = \sin a_i$$

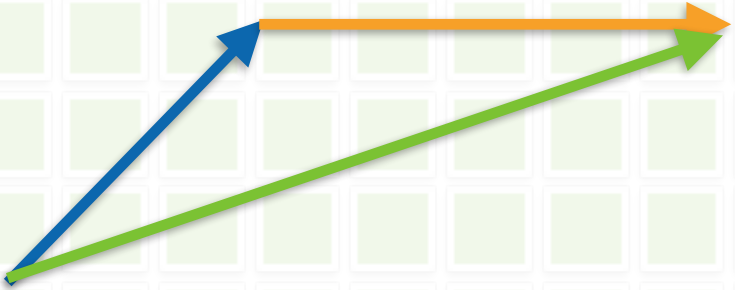
- 长度

$$\|a\|_2 = \left[\sum_{i=1}^{m} a_i^2\right]^{\frac{1}{2}}$$

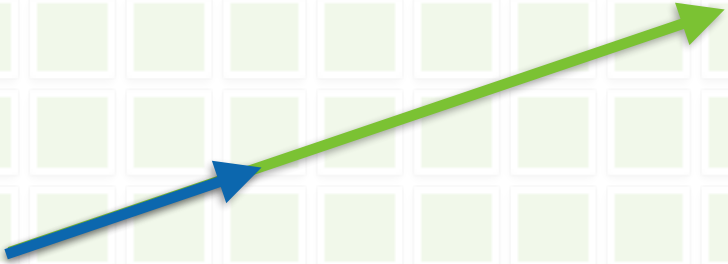
三角公式

$$\|a\| \geq 0 \text{ for all } a$$
$$\|a + b\| \leq \|a\| + \|b\|$$
$$\|a \cdot b\| = |a| \cdot \|b\|$$

向量



$$c = a + b$$

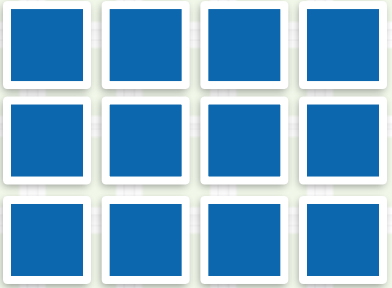


$$c = \alpha \cdot b$$

点积

$$a^\top b = \sum_i a_i b_i$$

矩阵



- 操作

$$C = A + B \quad \text{where } C_{ij} = A_{ij} + B_{ij}$$

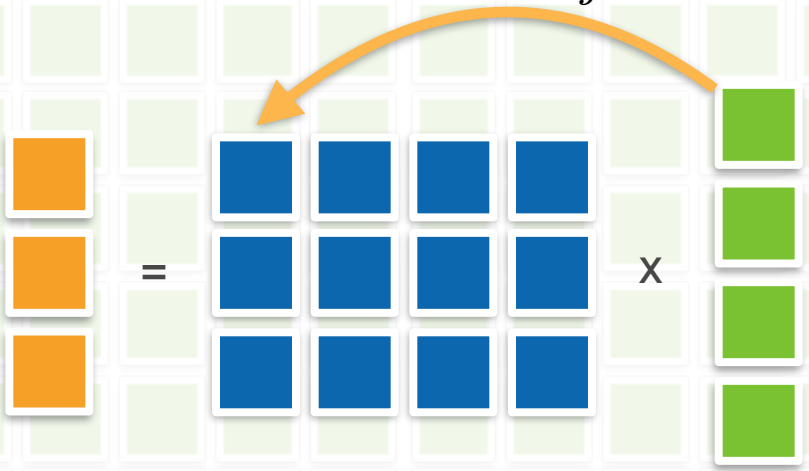
$$C = \alpha \cdot B \quad \text{where } C_{ij} = \alpha B_{ij}$$

$$C = \sin A \quad \text{where } C_{ij} = \sin A_{ij}$$

矩阵

- 乘法（矩阵乘以向量）

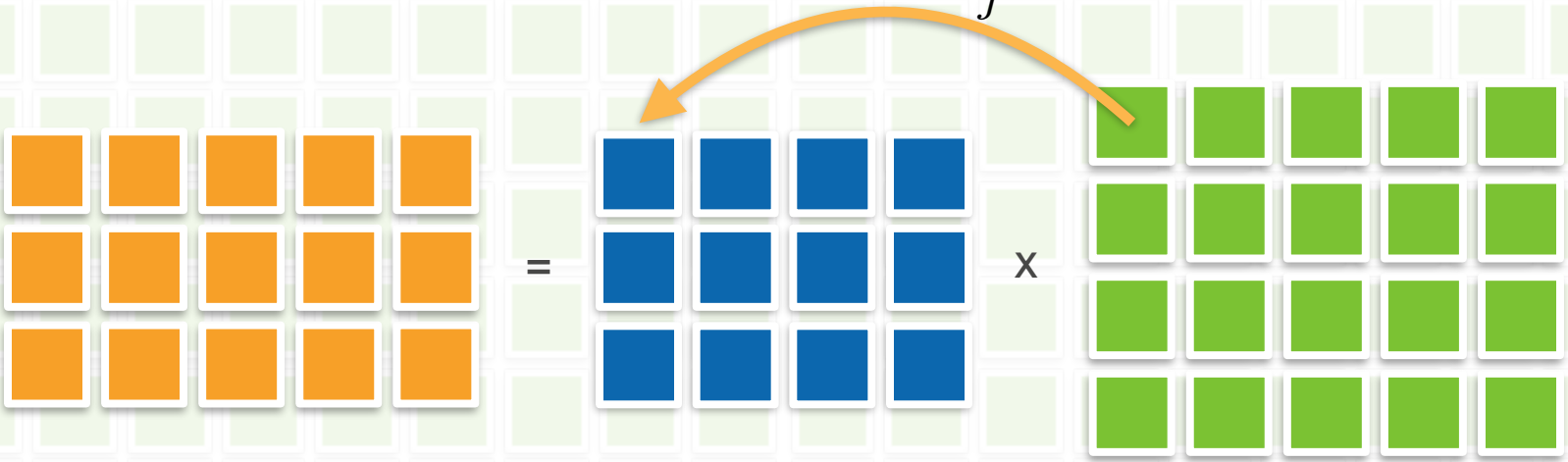
$$c = Ab \text{ where } c_i = \sum_j A_{ij} b_j$$

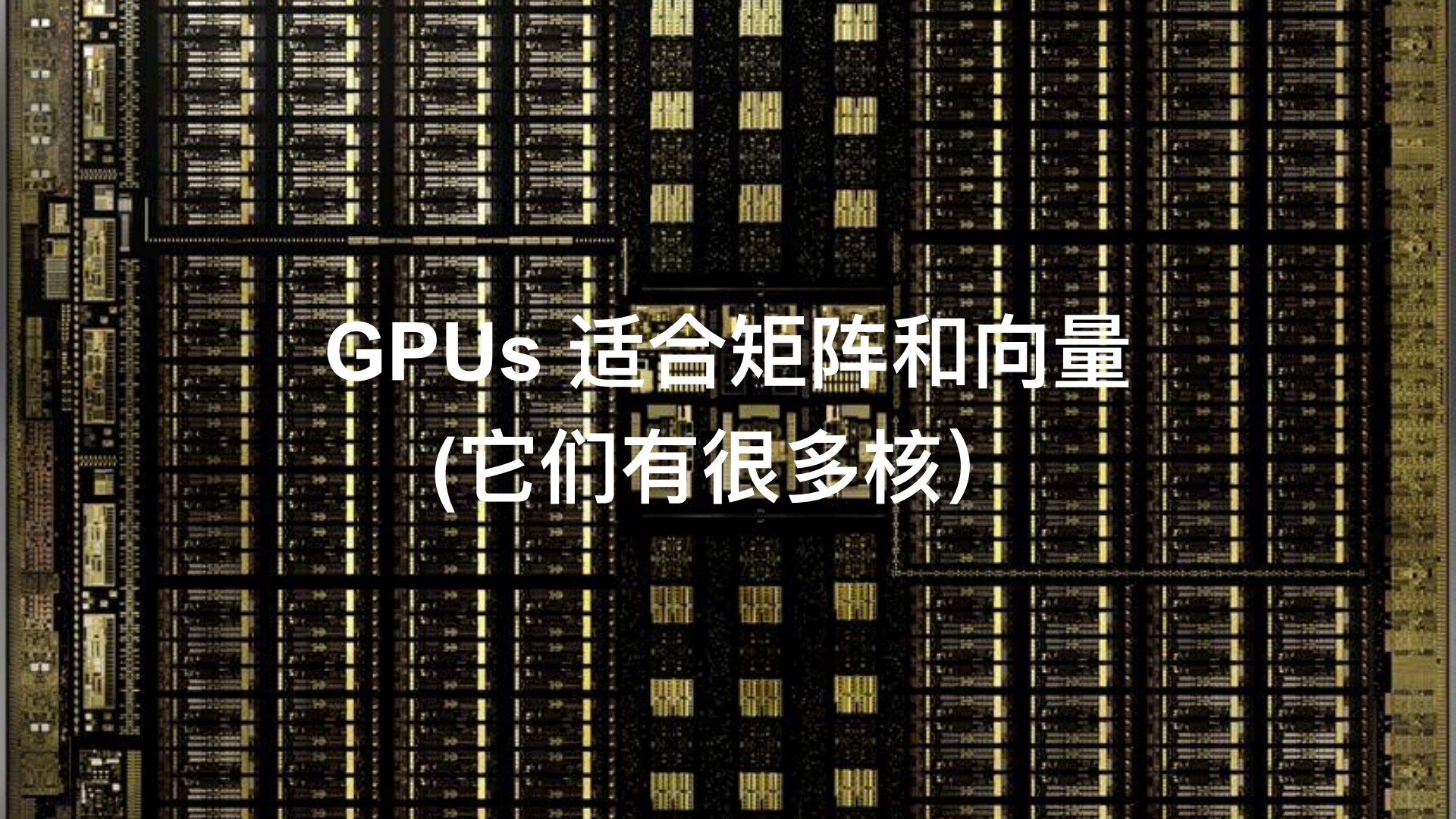


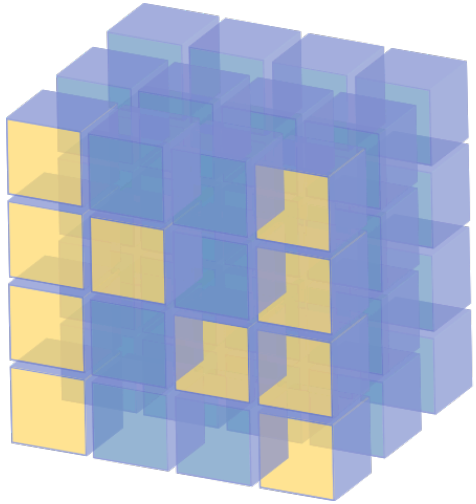
矩阵

- 乘法（矩阵乘以矩阵）

$$C = AB \text{ where } C_{ik} = \sum_j A_{ij} B_{jk}$$




GPUs 适合矩阵和向量
(它们有很多核)

Deep  NumPy

http://1day-zh.d2l.ai

aws

N-维数组

N-维数组是机器学习和深度学习的主要数据结构

0-d (标量)

```
1.0
```

一个类的标号

1-d (向量)

```
[1.0, 2.7, 3.4]
```

一个特征向量

2-d (矩阵)

```
[[1.0, 2.7, 3.4]
 [5.0, 0.2, 4.6]
 [4.3, 8.5, 0.2]]
```

样本-特征数据矩阵

http://1day-zh.d2l.ai

N-维数组

3-d

```
[[[0.1, 2.7, 3.4]
  [5.0, 0.2, 4.6]
  [4.3, 8.5, 0.2]]
 [[3.2, 5.7, 3.4]
  [5.4, 6.2, 3.2]
  [4.1, 3.5, 6.2]]]
```

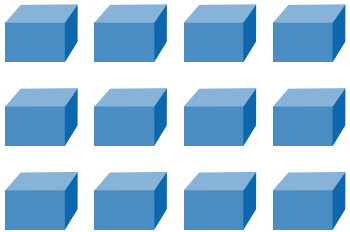
RGB 图片 (宽 x 高 x 通道)

4-d

```
[[[[. . .
    . . .
    . . .]]]]
```

一个 RGB 图片批量 (批量 x 宽 x 高 x 通道)

5-d



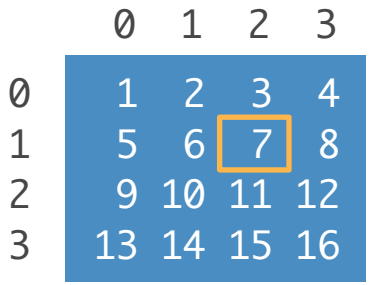
```
[[[[[. . .
     . . .
     . . .]]]]]
```

一个视频批量 (批量 x 时间 x 宽 x 高 x 通道)

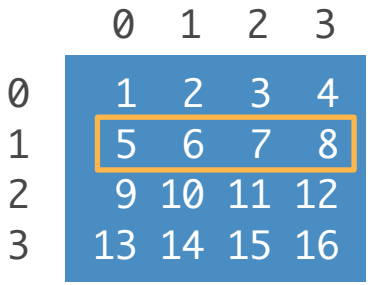
http://1day-zh.d2l.ai

元素読取

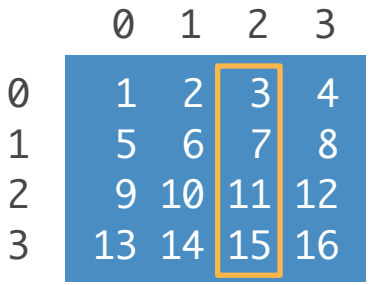
単个元素: [1, 2]

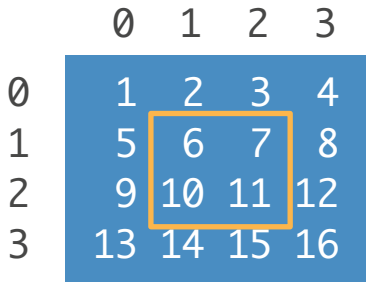


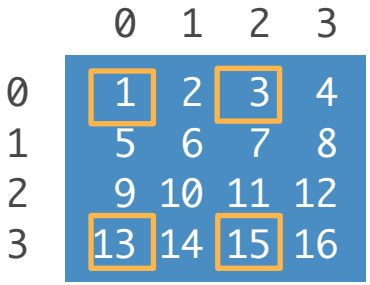
行: [1, :]



列: [1, :]







http://1day-zh.d2l.ai

NumPy notebook

http://1day-zh.d2l.ai

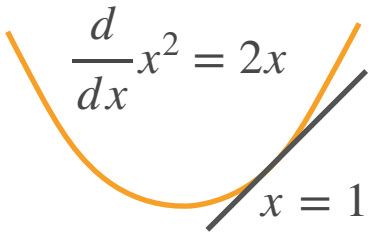
自动求导

http://1day-zh.d2l.ai

微积分 101

y	a	x^n	$\exp(x)$	$\log(x)$	$\sin(x)$
$\frac{dy}{dx}$	0	nx^{n-1}	$\exp(x)$	$\frac{1}{x}$	$\cos(x)$

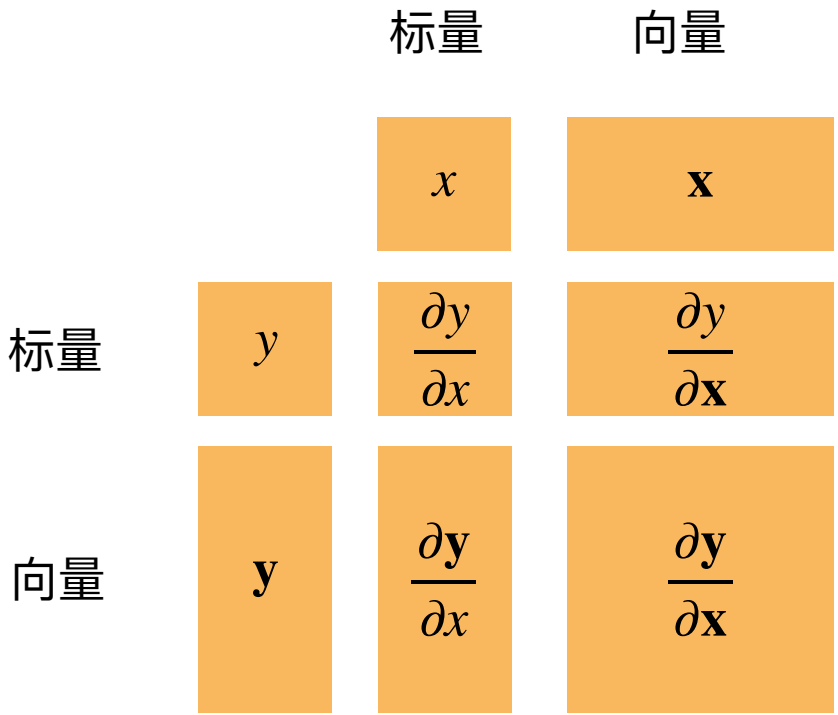
(梯度的斜率)



y	$u + v$	uv	$y = f(u),\ u = g(x)$
$\frac{dy}{dx}$	$\frac{du}{dx} + \frac{dv}{dx}$	$\frac{du}{dx}v + \frac{dv}{dx}u$	$\frac{dy}{du}\frac{du}{dx}$

http://1day-zh.d2l.ai

向量的导数



$$\left[\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right]_{ij} = \frac{\partial y_i}{\partial x_j}$$

http://1day-zh.d2l.ai

aws

例子

y	a	au	$\text{sum}(\mathbf{x})$	$\|\mathbf{x}\|^2$
$\frac{\partial y}{\partial \mathbf{x}}$	$\mathbf{0}^T$	$a\frac{\partial u}{\partial \mathbf{x}}$	$\mathbf{1}^T$	$2\mathbf{x}^T$

0 和 **1** 是向量

y	$u + v$	uv	$\langle \mathbf{u}, \mathbf{v} \rangle$
$\frac{\partial y}{\partial \mathbf{x}}$	$\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}$	$\frac{\partial u}{\partial \mathbf{x}}v + \frac{\partial v}{\partial \mathbf{x}}u$	$\mathbf{u}^T\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^T\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$

链式法则

标量

$$y = f(u),\ u = g(x) \qquad \frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x}$$

向量

$$\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial \mathbf{x}} \qquad \frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \qquad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$$

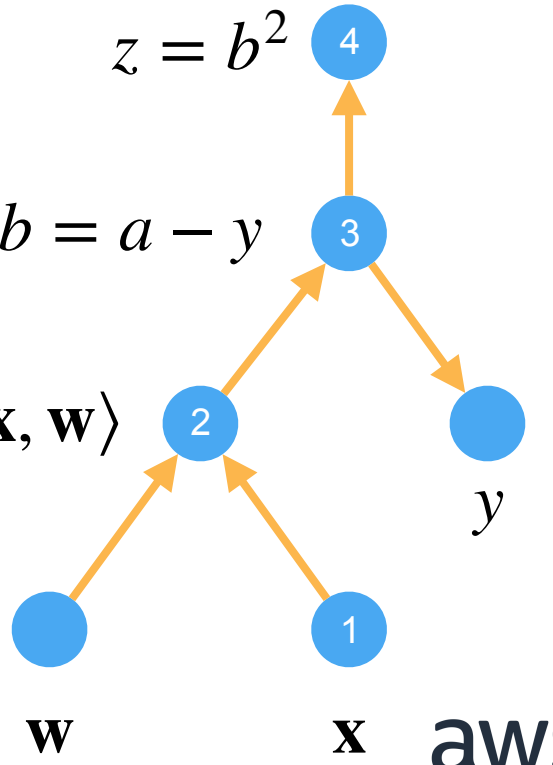
$$(1,n) \quad (1,)\ (1,n) \qquad (1,n) \quad (1,k)\ \ (k,n) \qquad (m,n)\ \ (m,k)\ (k,n)$$

http://1day-zh.d2l.ai

自动求导

$$z = \left(\langle \mathbf{x}, \mathbf{w} \rangle - y\right)^2$$

- 手动计算梯度很难，我们可以自动化过程
- 计算图
- 显示构建 (TensorFlow, MXNet Symbol)
- **隐式构建**
  (Chainer, PyTorch, DeepNumpy)
- **链式法则 (通过后项传递 backprop)**



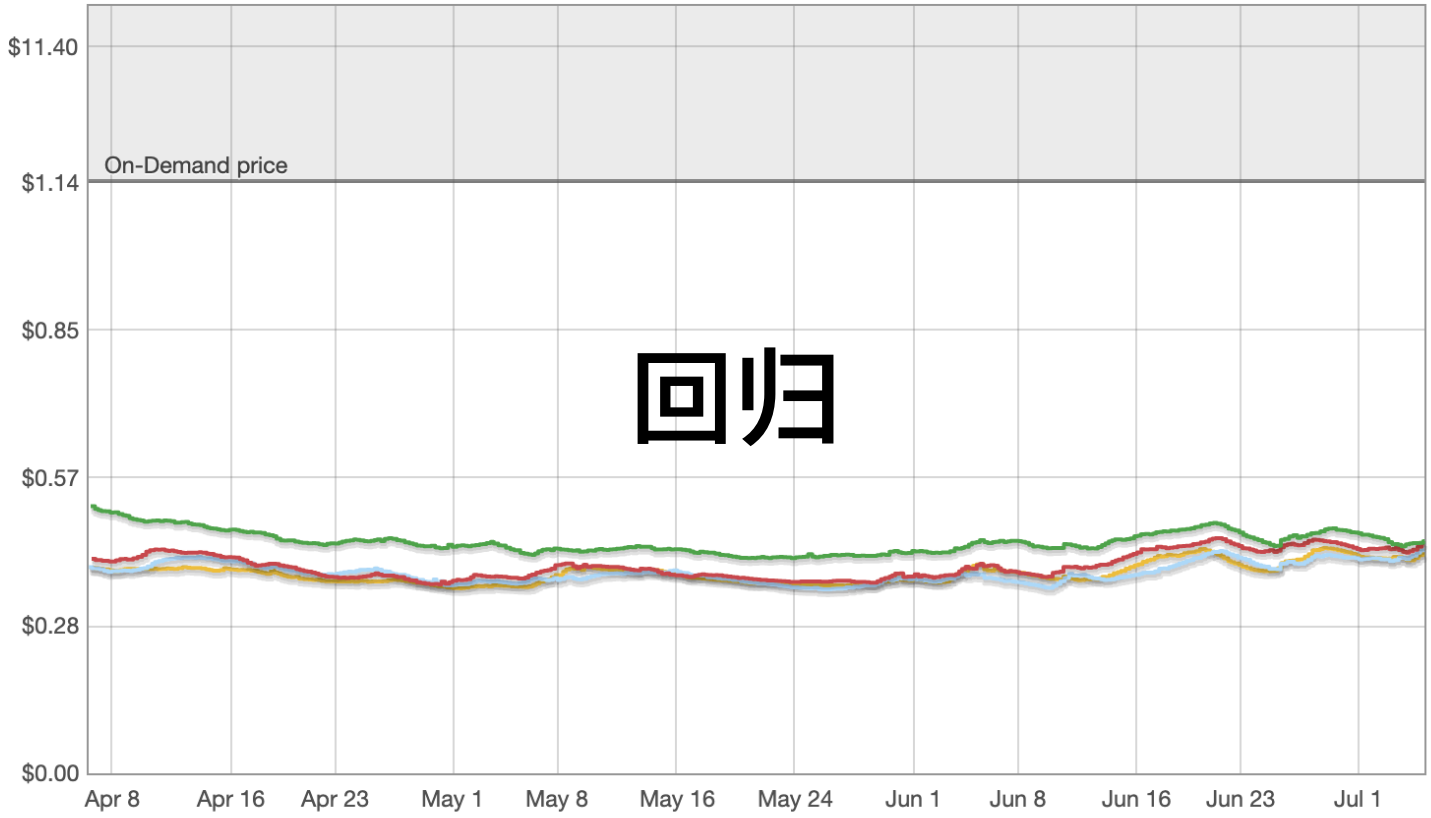
$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u_n}\frac{\partial u_n}{\partial u_{n-1}}\cdots\frac{\partial u_2}{\partial u_1}\frac{\partial u_1}{\partial x}$$

AutoGrad notebook

http://1day-zh.d2l.ai

Spot Instance Pricing History

Product: Linux/UNIX Instance type: g3.4xlarge Date range: 3 months



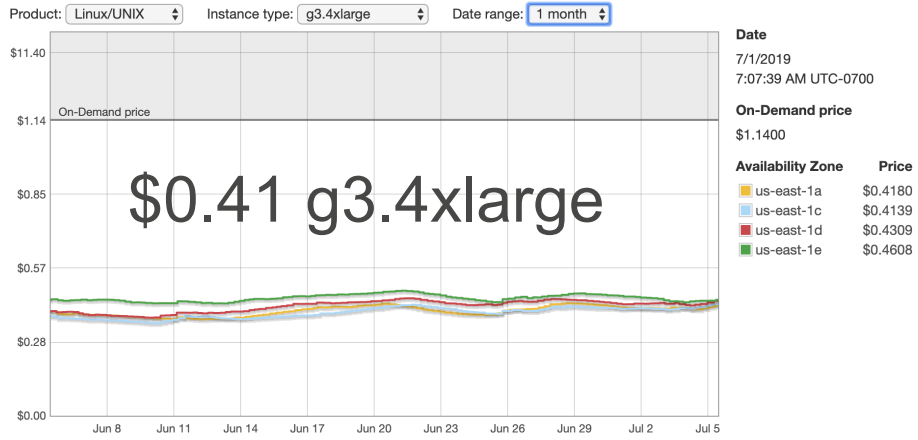
Date

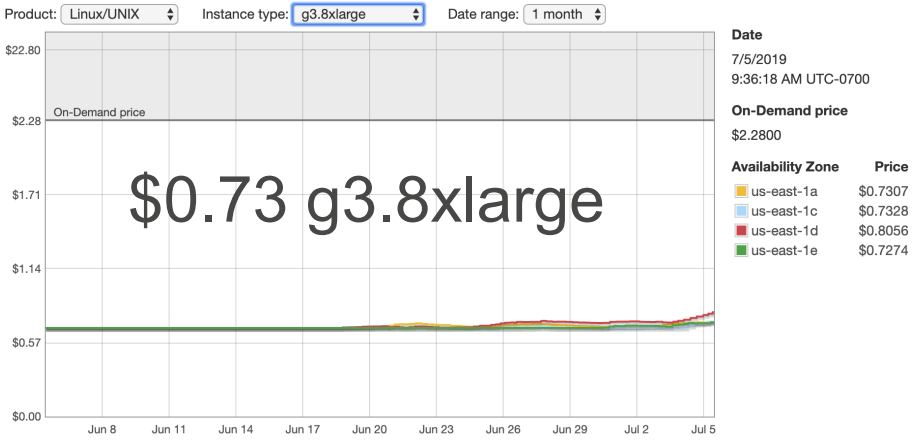
6/10/2019
9:47:38 PM UTC-0700

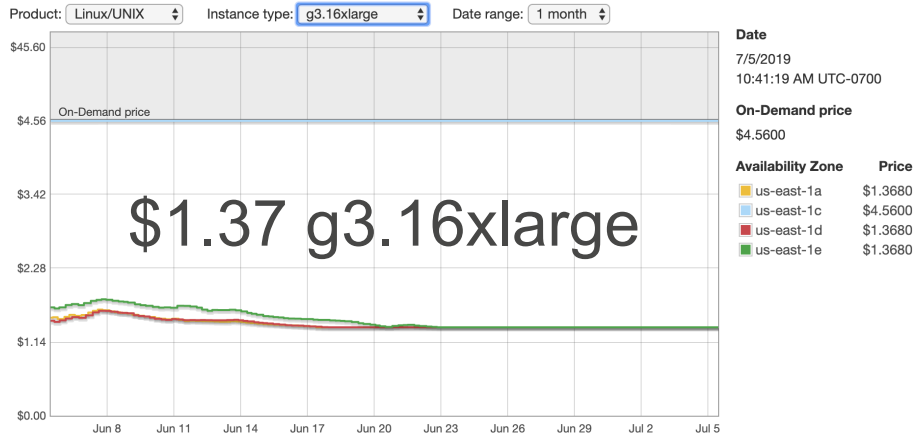
On-Demand price

$1.1400

Availability Zone	Price
us-east-1a	$0.3741
us-east-1c	$0.3710
us-east-1d	$0.3893
us-east-1e	$0.4353







我们可以估计价格（时间，机器，区域）吗？？

$$p = w_{\text{time}} \cdot t + w_{\text{server}} \cdot s + w_{\text{region}}[r]$$

http://1day-zh.d2l.ai

线性模型

- n-维输入 $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$
- 线性模型

$$\hat{y} = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$$

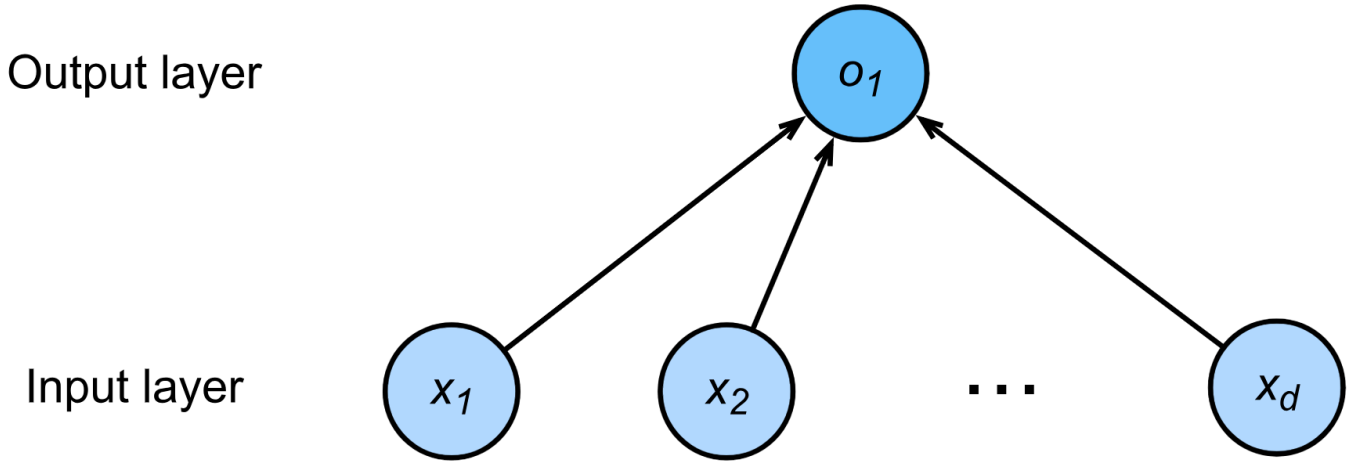
- 权重和偏移 $\mathbf{w} = [w_1, w_2, \ldots, w_n]^\top$ 和 b
- 向量化的版本

$$\hat{y} = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

- 损失函数（衡量拟合的好坏）

$$l(y, \hat{y}) = (y - \hat{y})^2$$

线性模型是一个单层神经网络



我们可以堆积神经层来得到深层神经网络

http://1day-zh.d2l.ai

aws

训练和测试

- 收集多个数据点来拟合参数
  (spot instance 过去6个月的价格)
- 训练数据 — 越多越好

$$\mathbf{X} = \left[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\right]^T \qquad \mathbf{y} = \left[y_0, y_1, \ldots, y_n\right]^T$$

- 测试数据 — 当使用模型时

$$\mathbf{X}' = \left[\mathbf{x}'_0, \mathbf{x}'_1, \ldots, \mathbf{x}'_{n'}\right]^T$$

在预测时我们不会知道标号（这就是机器学习的用途）

训练

- 目标是最小化训练误差

$$\underset{w}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \hat{y}_i)$$

- 代入 $\hat{y} = \langle \mathbf{w}, \mathbf{x} \rangle + b$ 可以得到

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b)^2 = \frac{1}{n} \left\| \mathbf{y} - \mathbf{X}\mathbf{w} - b \right\|^2$$

http://1day-zh.d2l.ai

Linear Regression notebook

http://1day-zh.d2l.ai

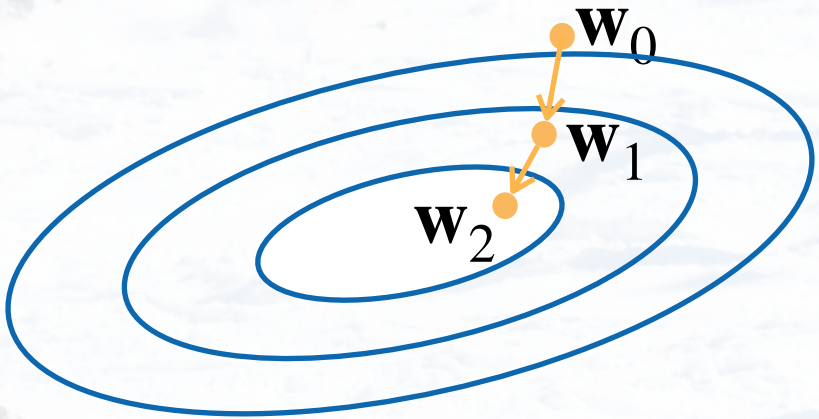
随机梯度下降

梯度下降

选择一个起始点 $\mathbf{w}_0$

重复下面的权重更新

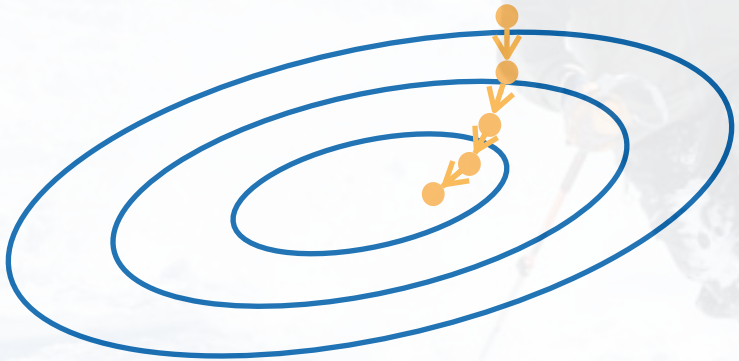
$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \partial_w l(w_{t-1})$$



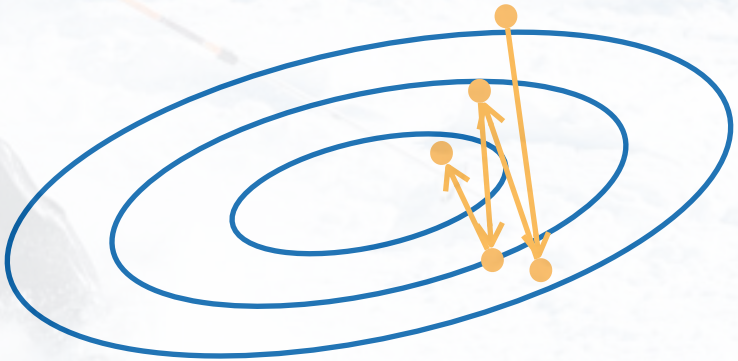
- 梯度方向指向增加函数值
- 学习率调节步长

学习率的取舍

太小



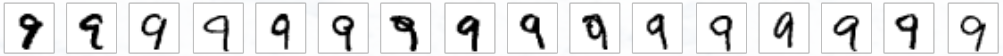
太大



http://1day-zh.d2l.ai

随机梯度下降（SGD）

- 在所有数据点上计算目标函数太复杂
- 数据里面可能有大量相似数据（例如相似数字）



- 单个样本计算不适合CPU/GPU
- 采样 b 样本 $i_1, \ldots i_b$ 来近似损失和梯度

$$\frac{1}{b}\sum_{i \in I_b} l(\mathbf{x}_i, y_i, \mathbf{w}) \text{ and } \frac{1}{b}\sum_{i \in I_b} \partial_{\mathbf{w}} l(\mathbf{x}_i, y_i, \mathbf{w})$$

b 是小批量大小

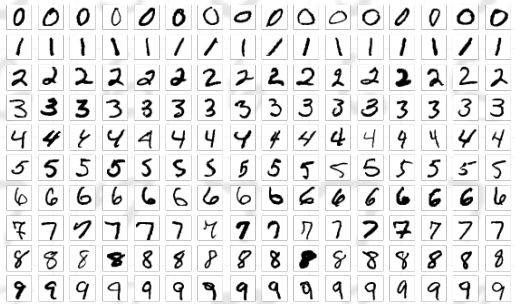

Logistic 回归

回归和分类

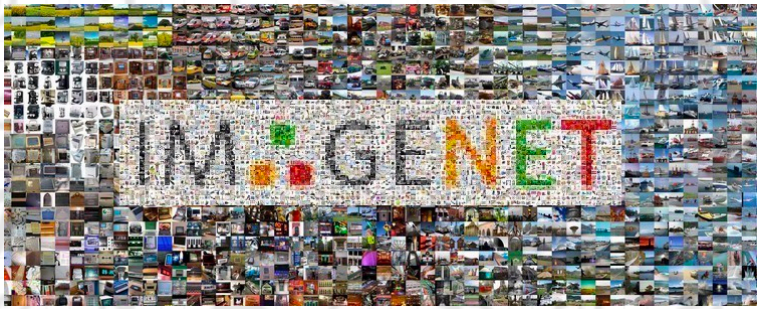
回归估计一个连续值

分类预测一个离散的类别

MNIST: 手写数字识别
(10 类)



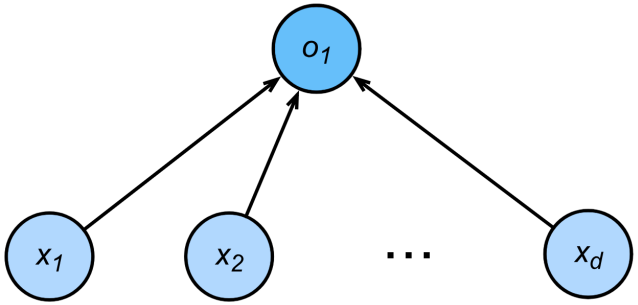
ImageNet: 分类自然物体
(1000 类)



从回归到多类分类

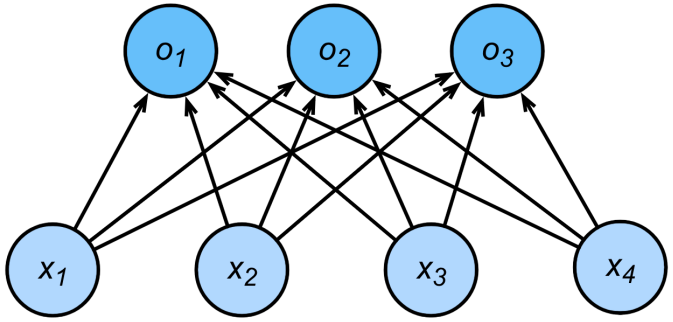
回归

- 单个连续值输出
- 自然区间 $\mathbb{R}$
- 损失是差值 $y - f(x)$



分类

- 多个类别，通常多个输出
- 值通常反映置信度



从回归到多类分类

均方误差

- 对每类进行 one-hot 编码

$$\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$$

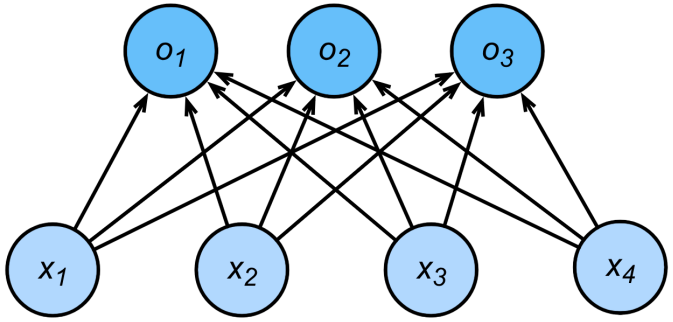
$$y_i = \begin{cases} 1 \text{ if } i = y \\ 0 \text{ otherwise} \end{cases}$$

- 使用均方误差函数
- 值最大的是预测类

$$\hat{y} = \underset{i}{\operatorname{argmax}}\, o_i$$

分类

- 多个类别，通常多个输出
- 值通常反映置信度



http://1day-zh.d2l.ai

从回归到多类分类

校正值大小

- 使得输出是概率值
  （非负，和为1）

$$p(y|o) = \text{softmax}(o)$$

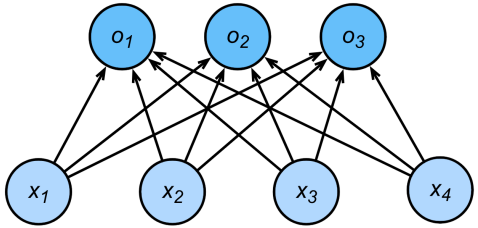
$$= \frac{\exp(o_y)}{\sum_i \exp(o_i)}$$

- 负的对数似然度

$$-\log p(y|o) = \log \sum_i \exp(o_i) - o_y$$

分类

- 多个类别，通常多个输出
- 值通常反映置信度



Softmax 和交叉熵损失函数

- 负的对数似然度 (给定标号 y)

$$-\log p(y \mid o) = \log \sum_i \exp(o_i) - o_y$$

- 交叉熵损失 (对于概率分布 y)

$$l(y, o) = \log \sum_i \exp(o_i) - y^\top o$$

- **梯度**

预计的分布跟真实的区别

$$\partial_o l(y, o) = \frac{\exp(o)}{\sum_i \exp(o_i)} - y$$

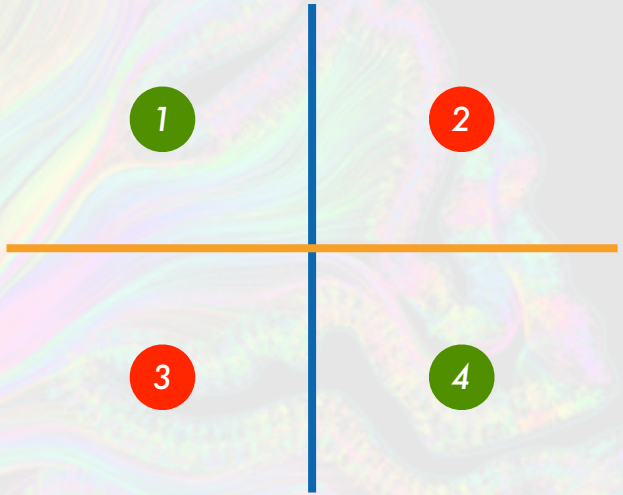
Logistic Regression Notebook

http://1day-zh.d2l.ai

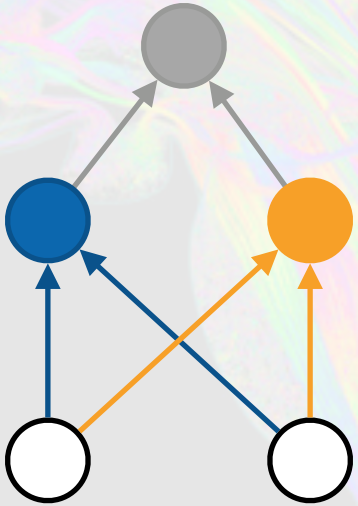

多层感知机

拟合 XOR

	1	2	3	4
	+	-	+	-
	+	+	-	-
product	+	-	-	+



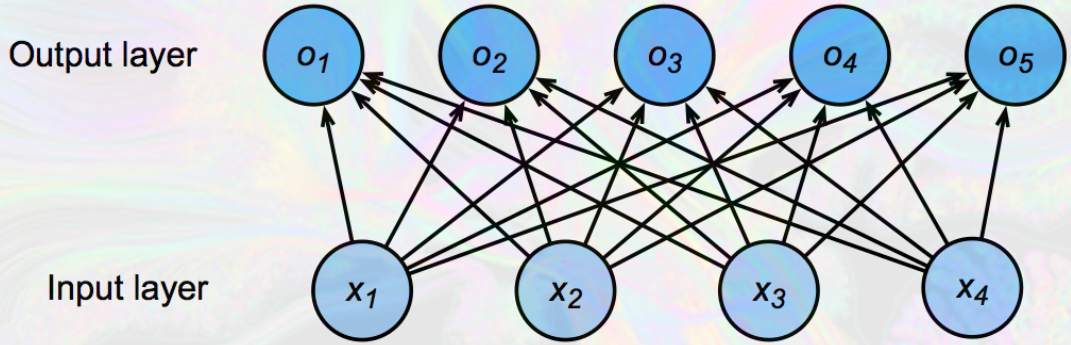
$$w_{11}x_1 + w_{12}x_2 + b_1$$



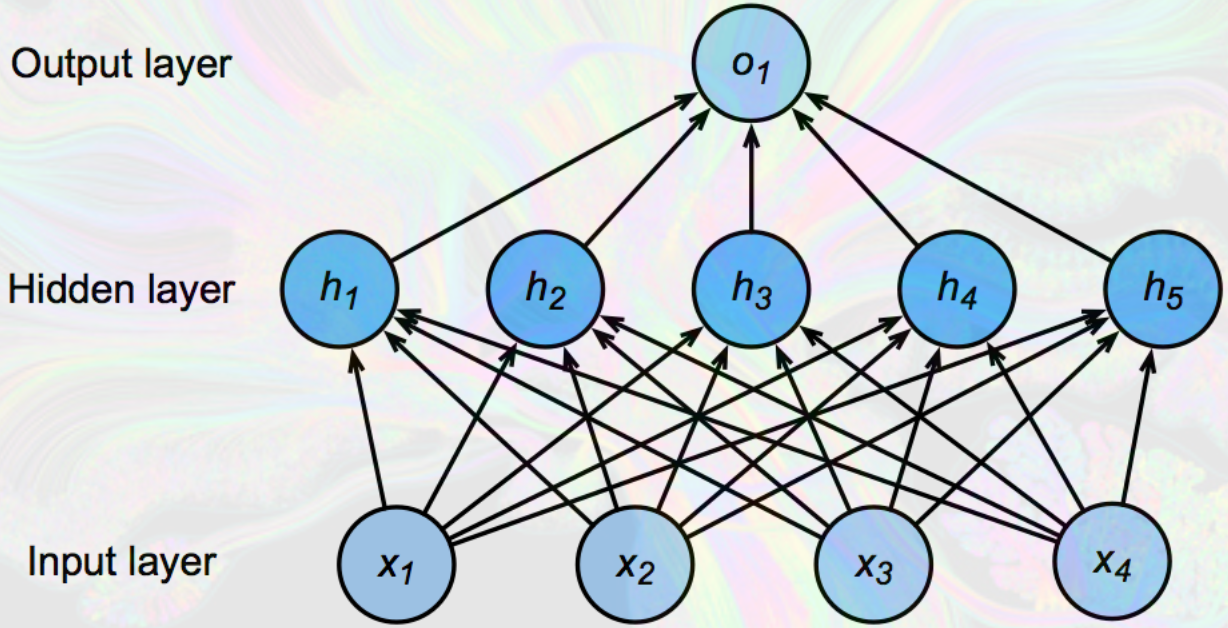
$$w_{21}x_1 + w_{22}x_2 + b_2$$

http://1day-zh.d2l.ai

单层网络



单隐藏层


Output layer
Hidden layer
Input layer
o_1
h_1
h_2
h_3
h_4
h_5
x_1
x_2
x_3
x_4

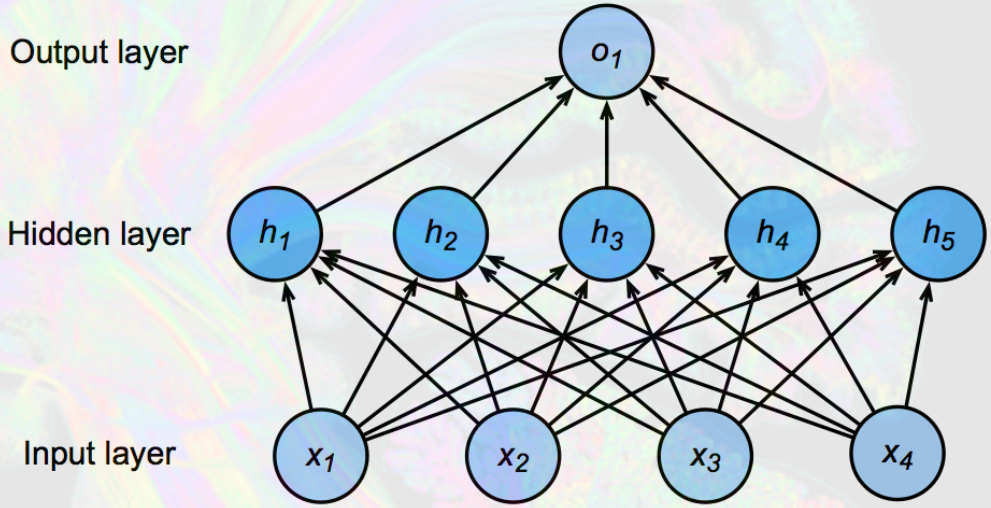
单隐藏层

- 输入　$\mathbf{x} \in \mathbb{R}^n$
- 隐藏层　$\mathbf{W}_1 \in \mathbb{R}^{m \times n}, \mathbf{b}_1 \in \mathbb{R}^m$
- 输出　$\mathbf{w}_2 \in \mathbb{R}^m, b_2 \in \mathbb{R}$

$$\mathbf{h} = \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{o} = \mathbf{w}_2^T \mathbf{h} + b_2$$

σ 是一个按元素的激活函数



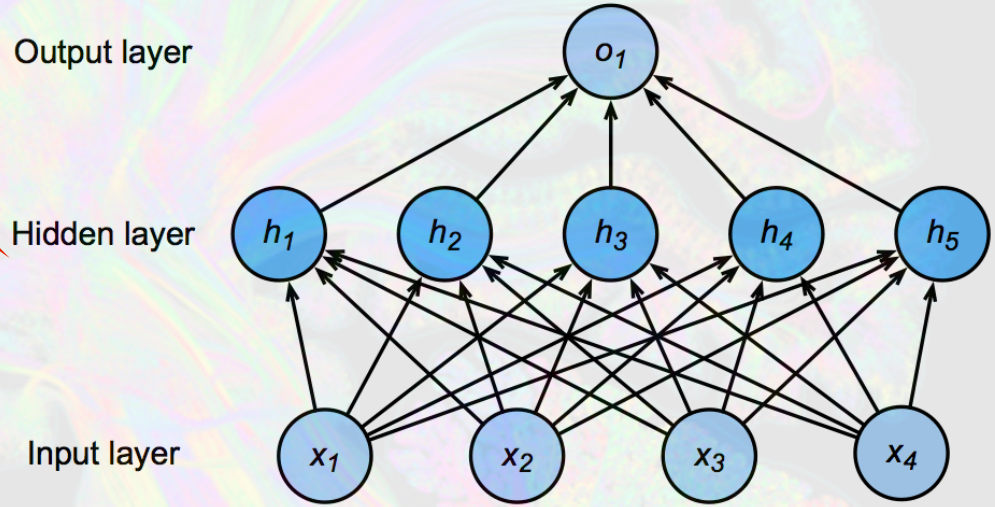
单隐藏层

为什么需要非线性的激活函数?

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{o} = \mathbf{w}_2^T\mathbf{h} + b_2$$

σ 是一个按元素的激活函数



单隐藏层

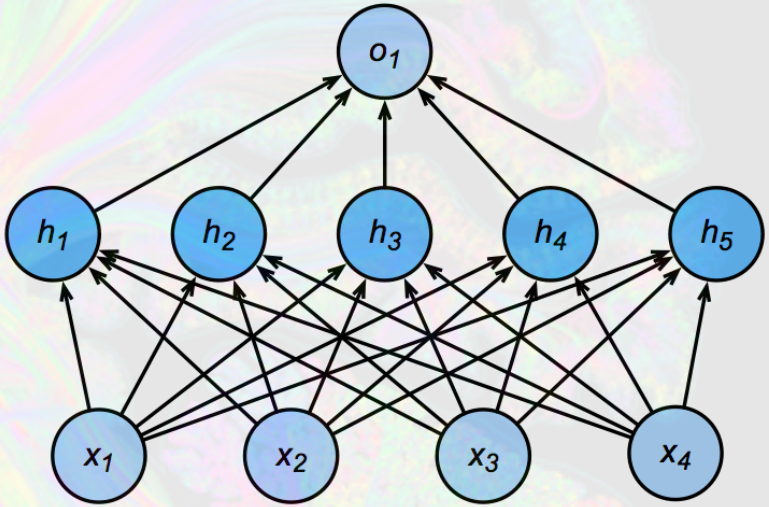
为什么需要非线性的激活函数?

$$\mathbf{h} = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1$$

$$\mathbf{o} = \mathbf{w}_2^T\mathbf{h} + b_2$$

$$\text{hence } o = \mathbf{w}_2^\top\mathbf{W}_1\mathbf{x} + b'$$

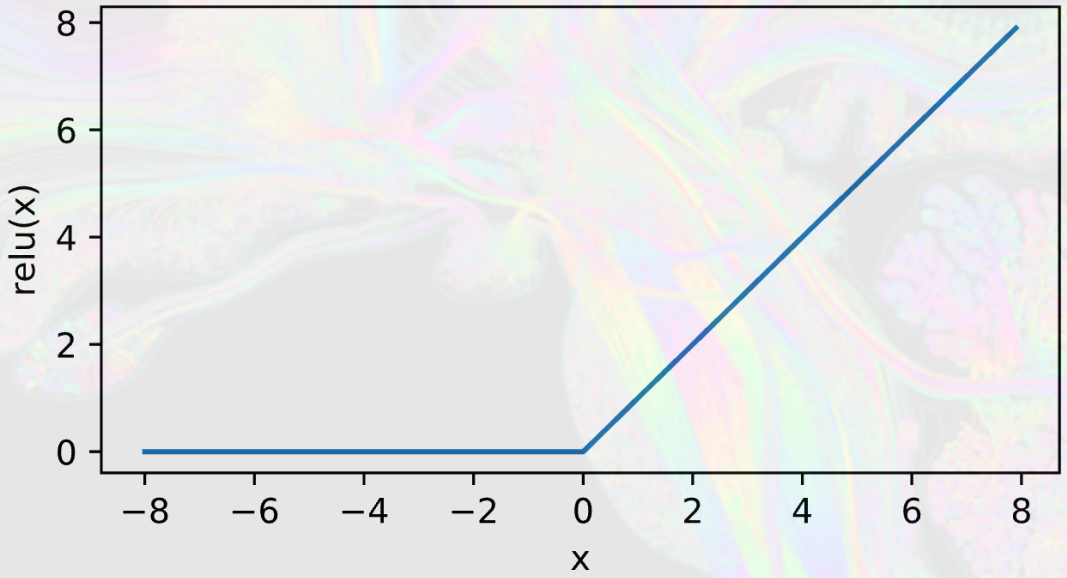
线性 ...



ReLU 激活函数

ReLU: rectified linear unit

$$\mathrm{ReLU}(x) = \max(x, 0)$$



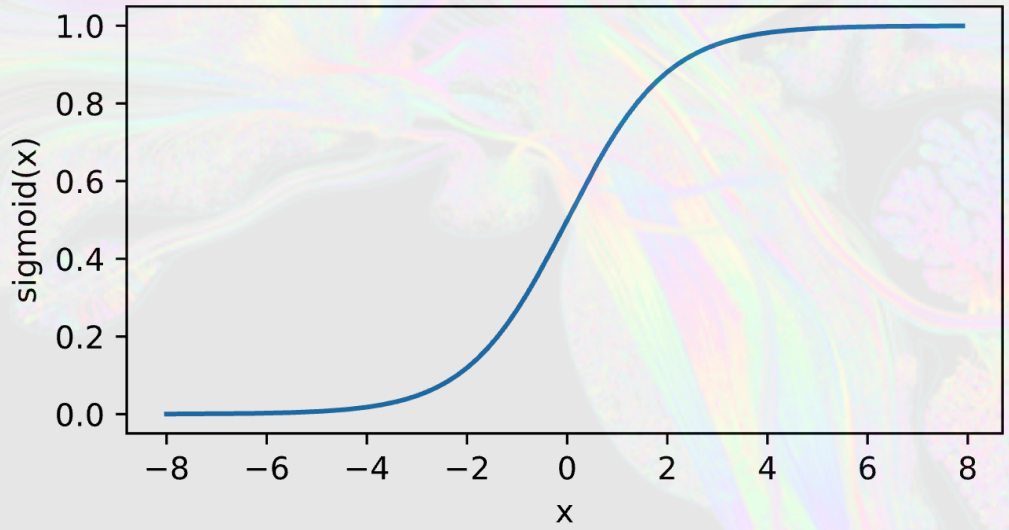
http://1day-zh.d2l.ai

Sigmoid 激活函数

将输入映射到 (0, 1), 是平滑版的

$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

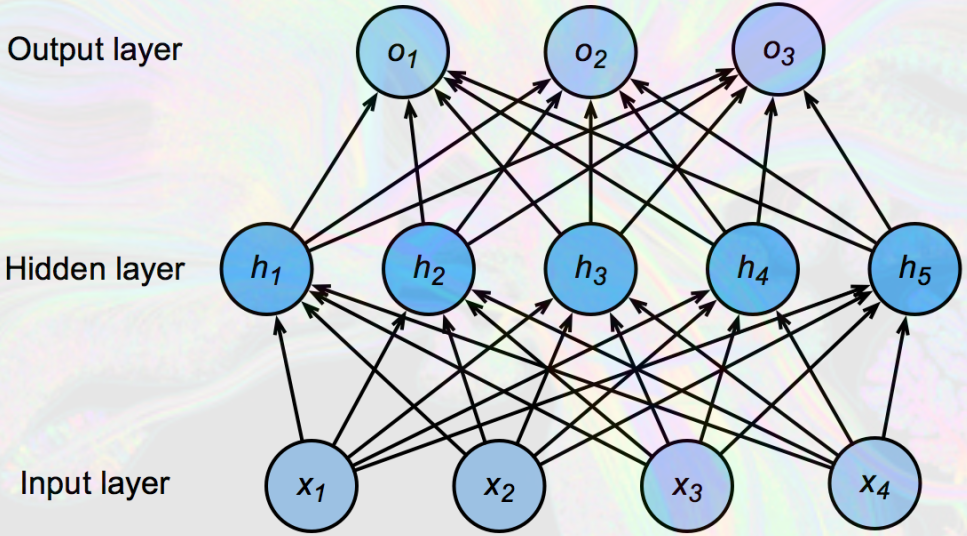
$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$



http://1day-zh.d2l.ai

多类分类

$$y_1, y_2, \ldots, y_k = \text{softmax}(o_1, o_2, \ldots, o_k)$$

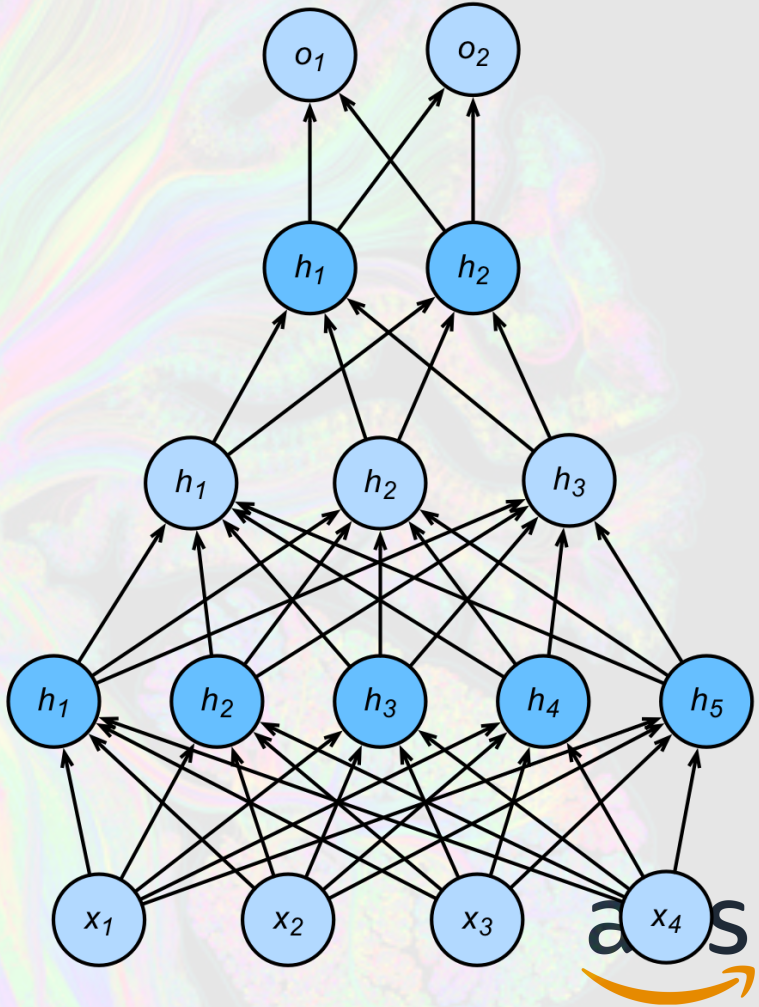


多隐藏层

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$
$$\mathbf{h}_3 = \sigma(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3)$$
$$\mathbf{o} = \mathbf{W}_4\mathbf{h}_3 + \mathbf{b}_4$$

超参数

- 隐藏层数
- 每个隐藏层的大小



MLP Notebook

http://1day-zh.d2l.ai

总结

- 深度学习介绍
- 安装
- 线性代数和张量计算
- 自动求导
- 线性回归
- 优化
- Softmax 回归
- 多层感知机 (训练 MNIST)

http://1day-zh.d2l.ai