



中国科学院大学
University of Chinese Academy of Sciences

深度学习基础





图像识别为什么使用深度学习

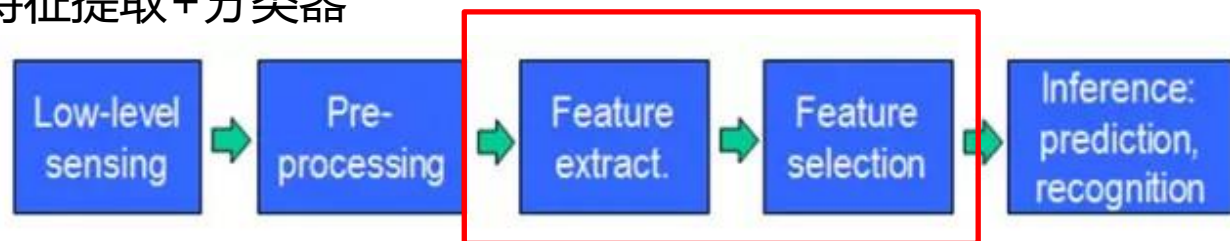
AI DISCOVERY

测量空间 $\xrightarrow{\text{red arrow}}$ 特征空间 $\xrightarrow{\text{blue arrow}}$ 类别空间

核心任务

传统方法

- 人工特征提取+分类器



深度学习

- 手工地选取特征是一件非常费力、启发式（需要专业知识）的方法，能不能选取好很大程度上靠经验和运气，而且它的调节需要大量的时间

AI DISCOVERY



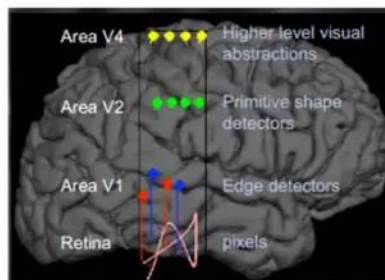
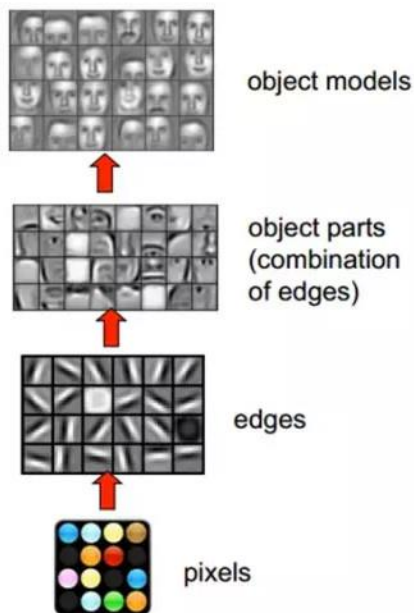
图像识别为什么用深度学习



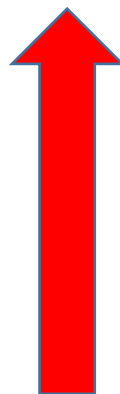
AI DISCOVERY

人脑视觉机理：神经-中枢-大脑

- 从原始信号摄入（瞳孔摄入像素 Pixels）→初步处理（大脑皮层某些细胞发现边缘和方向）→抽象（大脑判定，眼前的物体的形状，是圆形的）→进一步抽象（大脑进一步判定该物体是人脸）
- 从原始信号，做低级抽象，逐渐向高级抽象迭代



特征空间（概念）



测量空间（表象）



AI DISCOVERY

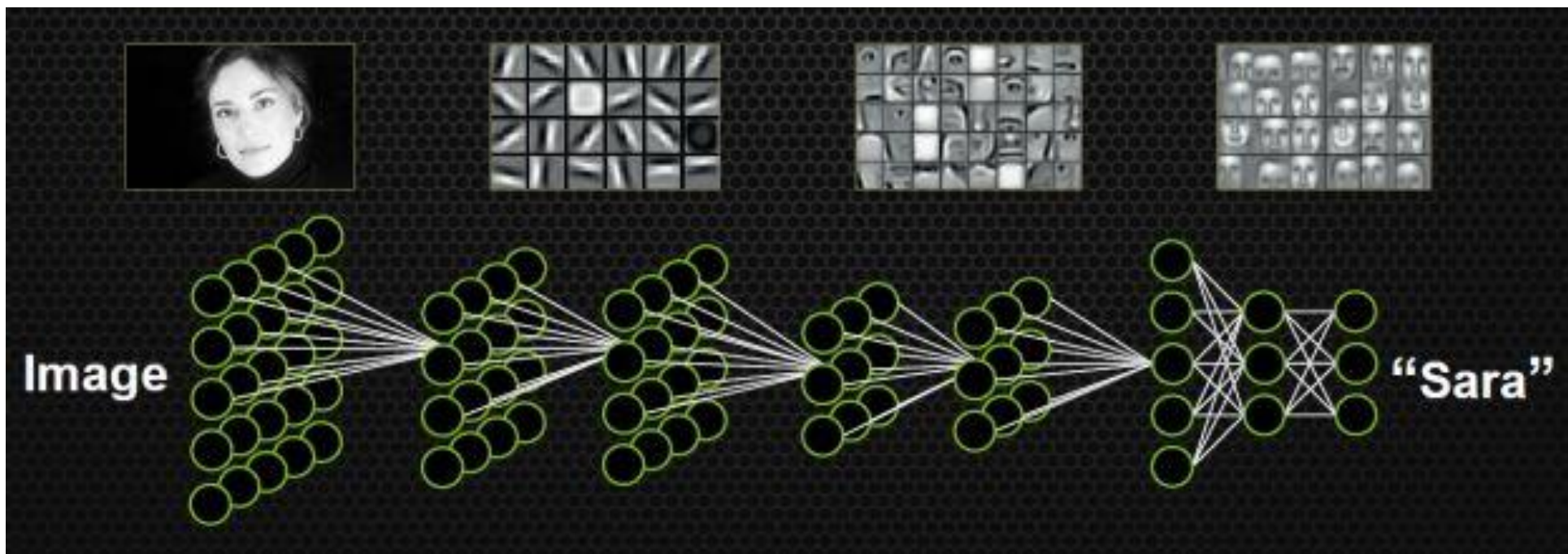


图像识别为什么用深度学习



AI DISCOVERY

深度学习模仿人类的视觉系统



AI DISCOVERY



使用步骤

AI DISCOVERY

★ 建立模型

- 选择什么样的网络结构
- 选择多少层数，每层选择多少神经元

★ 损失函数

- 选择常用损失函数，平方误差，交叉熵....

★ 参数学习

- 梯度下降
- 反向传播算法

AI DISCOVERY

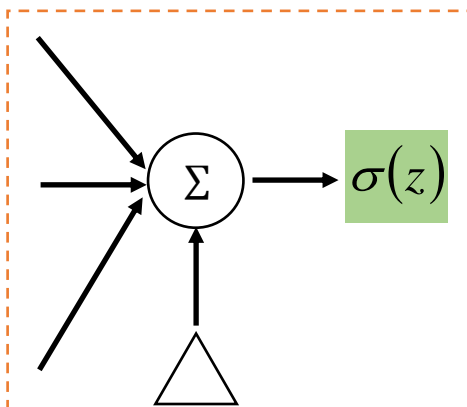
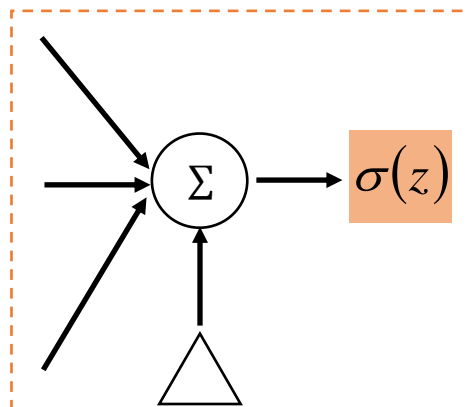
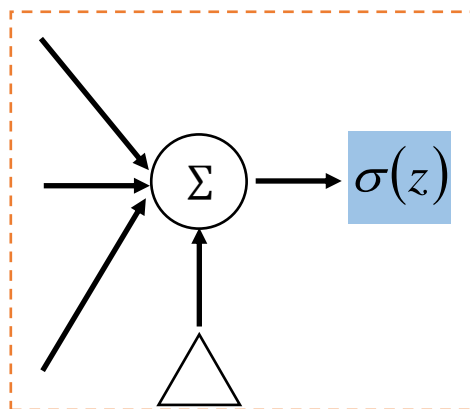
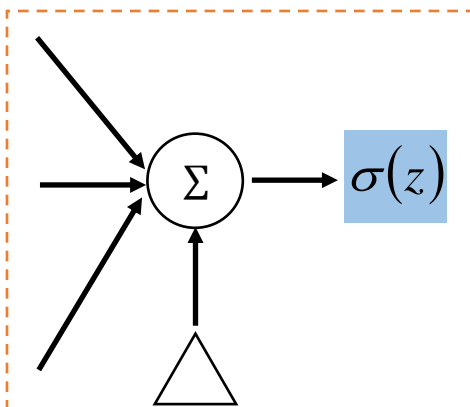


建立模型

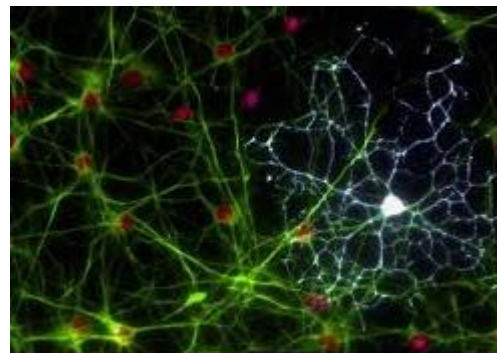
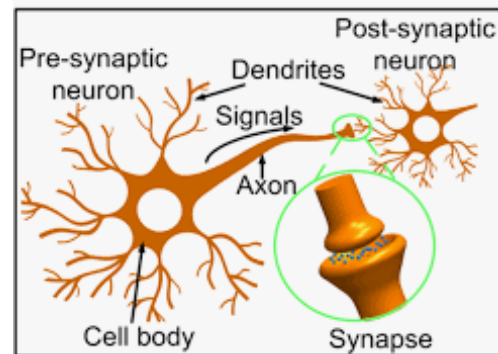
AI DISCOVERY

★ 网络结构

◆ 神经元不同的连接方式构成不同的网络结构



每个神经元都有自己的权重和偏置参数





建立模型

AI DISCOVERY

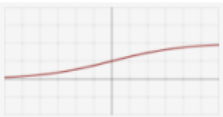
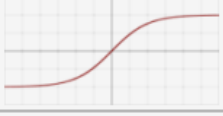


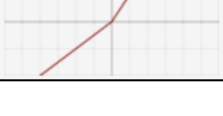


建立模型

神经元

◆ 为什么引入激活函数

- 为了增强网络的表达能力，我们需要激活函数来将线性函数->非线性函数
- 非线性的激活函数需要有连续性。因为连续非线性激活函数可以可导的，所以可以用最优化的方法来求解

Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU) ^[7]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) ^[8]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

常用激活函数示例

AI DISCOVERY



建立模型



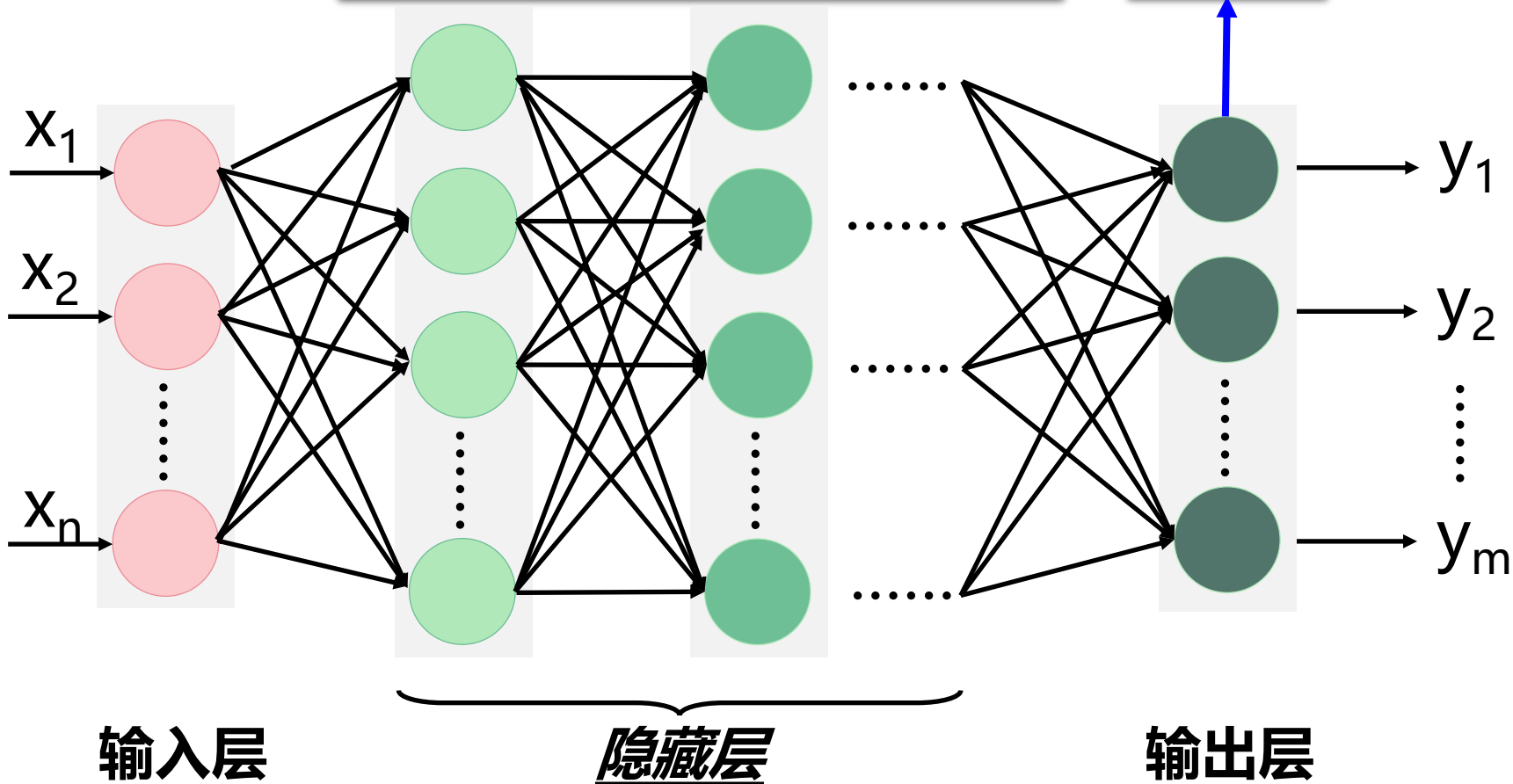
AI DISCOVERY



前馈神经网络

隐藏层数很多意味着网络越深

神经元



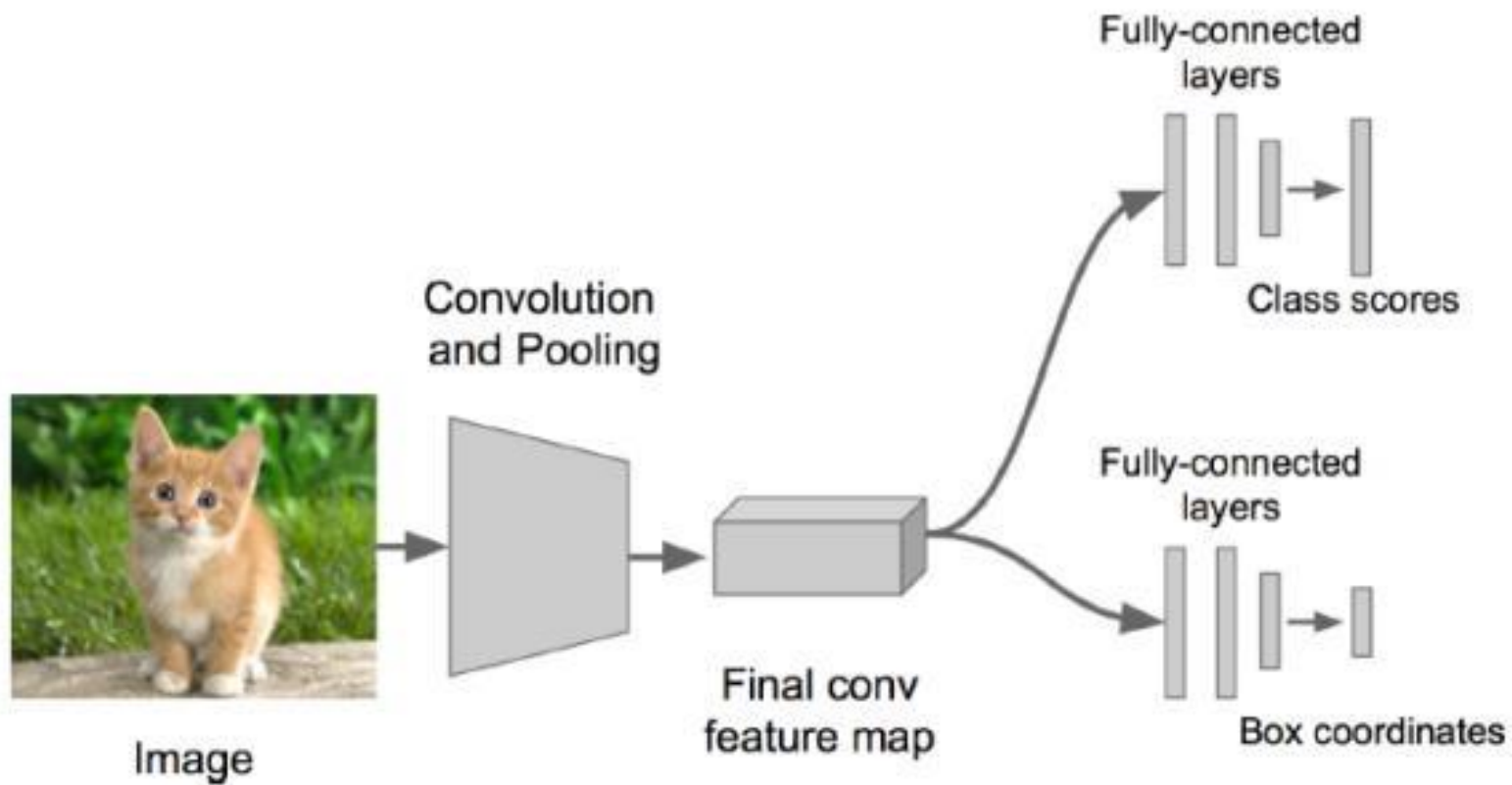
AI DISCOVERY



损失函数



AI DISCOVERY



分类损失

回归损失

损失函数的设计依赖于具体的任务



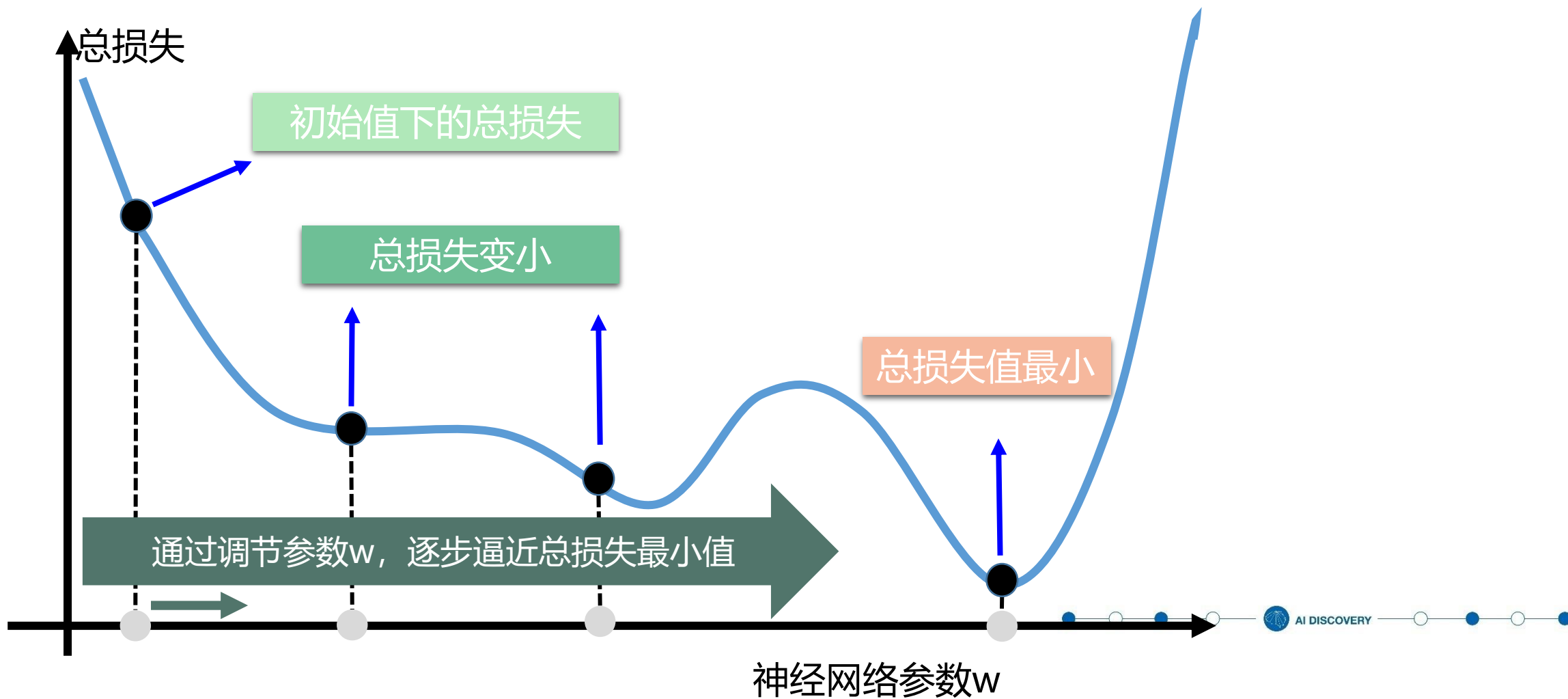
AI DISCOVERY



参数学习

AI DISCOVERY

★ 梯度下降法





参数学习

反向传播算法



只剩一个问题：
怎么求 梯度 $\partial L / \partial w$?

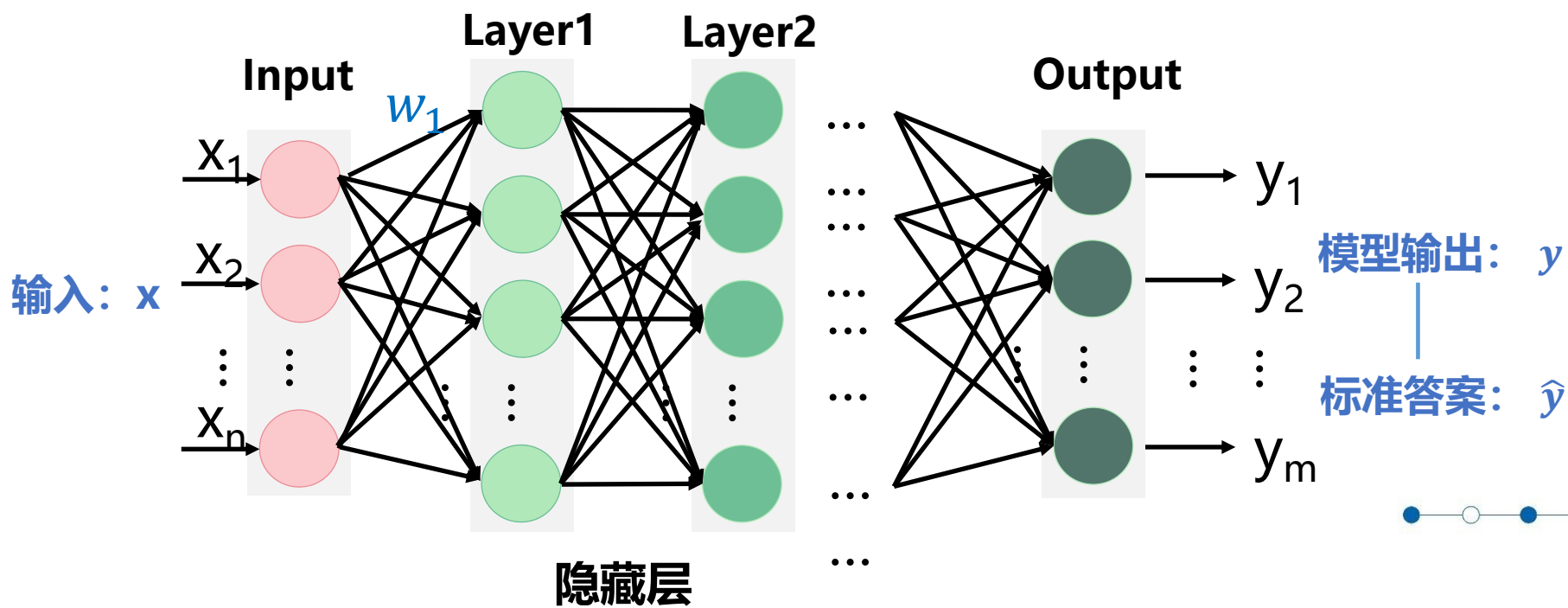
$$w_1 - \eta \partial L / \partial w_1$$

更新 w_1

计算 $\partial L / \partial w_1$

总误差 L

以此类推更新 w_2 w_3





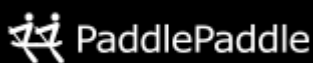
主流的深度学习开源工具

AI DISCOVERY

PaddlePaddle是百度提供的国内首个开源深度学习框架，是基于“深度学习编程语言”的新一代深度学习框架，在兼具性能的同时，极大的提升了框架对模型的表达能力。框架本身具有易学、易用、安全、高效四大特性，是最适合中国开发者和企业的深度学习工具。

GitHub项目地址：

<https://github.com/PaddlePaddle/Paddle>



Caffe是由神经网络中的表达式、速度及模块化产生的深度学习框架。Caffe是一个基于C++/CUDA架构框架，开发者能够利用它自由的组织网络，目前支持卷积神经网络和全连接神经网络（人工神经网络）。在Linux上，C++可以通过命令行来操作接口，运算上支持CPU和GPU直接无缝切换。

但官方文档和教程支持不是很充分，对初学者来说门槛较高。

GitHub项目地址：

<https://github.com/BVLC/caffe>

Caffe

TensorFlow是谷歌基于C++开发、发布的第二代机器学习系统。开发目的是用于进行机器学习和深度神经网络的研究。目前Google的Google App的语音识别、Gmail的自动回复功能、Google Photos的图片搜索等都在使用TensorFlow。

但内部概念众多，结构复杂，API繁重上手困难，版本迭代快API兼容性问题。

GitHub项目地址：

<https://github.com/tensorflow/tensorflow>



Keras是基于Python开发的极其精简并高度模块化的神经网络库，在TensorFlow或Theano上都能够运行，是一个高度模块化的神经网络库，支持GPU和CPU运算。Keras侧重于开发快速实验，用可能最少延迟实现从理念到结果的转变，即为做好一项研究的关键。

GitHub项目地址：

<https://github.com/fchollet/keras>





PaddlePaddle计算模型

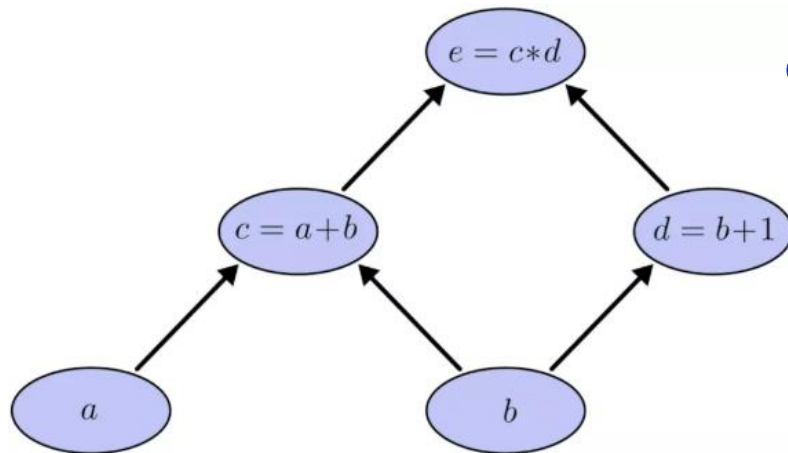


AI DISCOVERY

计算图

描述计算的经典方式：

- ✓节点代表计算
- ✓边代表计算之间的依赖关系



$$e=(a+b)*(b+1)$$

PaddlePaddle Fluid去掉静态计算图的概念，采用Program的形式动态描述计算过程。

在Fluid程序实例中，区分编译时和运行时：

- ✓编译时：用户像写程序一样描述计算。
- ✓运行时：执行规划好的计算

Fluid中不再有模型的概念，
提倡像写程序一样描述计算



AI DISCOVERY



PaddlePaddle计算模型



AI DISCOVERY

Program

在Fluid中，一个神经网络任务（训练/预测）被描述为一段Program

Fluid通过提供顺序、分支和循环三种执行结构的支持，让用户可以通过组合描述任意复杂的模型。

➤ 顺序执行

```
x = fluid.layers.data(name='x', shape=[13], dtype='float32')
y_predict = fluid.layers.fc(input=x, size=1, act=None)
y = fluid.layers.data(name='y', shape=[1], dtype='float32')
cost = fluid.layers.square_error_cost(input=y_predict, label=y)
```

➤ 条件分支

```
with fluid.layers.control_flow.Switch() as switch:
    with switch.case(global_step == zero_var):
        fluid.layers.tensor.assign(input=one_var, output=lr)
    with switch.default():
        fluid.layers.tensor.assign(input=two_var, output=lr)
```

网络结构修改的灵活性、模型搭建的便捷性