机器学习2024
第11节

涉及知识点：
多任务学习、自动机器学习、迁移学习、元学习、机器学习的未来

从单一任务到多个任务学习

张伟楠 – 上海交通大学

课程安排

参数化有监督学习

1. 机器学习概述
2. 线性模型
3. 双线性模型
4. 神经网络

非参数化有监督学习

5. 支持向量机
6. 决策树
7. 集成学习与森林模型

无监督学习部分

8. 概率图模型
9. 无监督学习

学习理论部分

10. 学习理论与模型选择

前沿话题部分

11. 迁移、多任务、元学习
12. System 1&2 机器意识

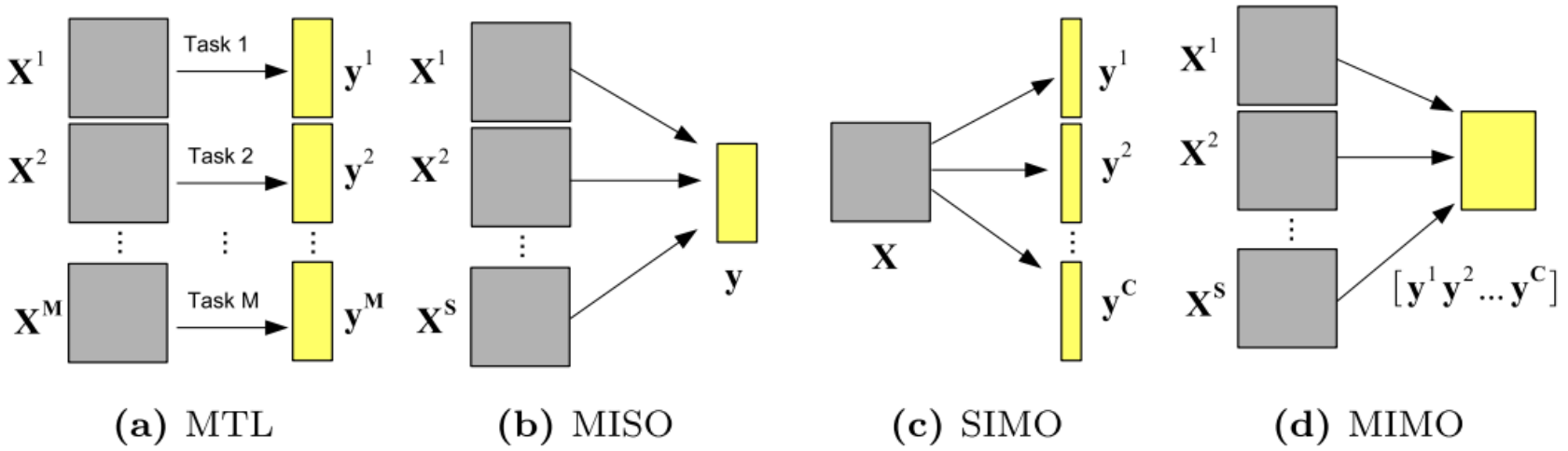
多任务学习

张伟楠 – 上海交通大学

什么是多任务学习？

- 什么是任务？
  - 使用单一输入源
  - 学习单个输出目标
- “多任务学习（Multi-task learning, MTL）”意味着：
  - 单一输入源 → 多个输出目标
  - 多输入源 → 单个输出目标
  - 或：两者的混合

多任务学习类型



- MISO（多输入单输出）：不能简单地简化为单任务学习（其他数据源在预测时可能不可用）
- SIMO（单输入多输出）：可简化为单任务学习（多分类）

为什么有效？

- 多任务学习（MTL）是机器学习的一个分支领域，它利用任务之间的共性和差异，同时解决多个学习任务

- 与单独训练模型相比，这可以提高特定任务模型的学习效率和预测精度

- 基本假设：
  - 学习中的所有任务，或者至少一个子集，是相互关联的

为什么有效？

- 隐式的数据增广（data augmentation）
  - 有效增加样本量
  - 不同任务具有不同噪声模式
- “窃取”（eavesdropping）
  - 一些特征G对任务B来说很容易，但对于A来说很难
  - 通过多任务学习，我们可以使A通过B学习G
- 表示偏差
  - 倾向更泛化的表示

相关研究领域

- 迁移学习（Transfer learning, TL）
  - 主要任务和辅助任务；可以被视为多任务学习的一种特例，多任务学习中对任务的处理是平等的
  - 假设：关联的二级任务可以为主要任务提供额外的信息，提高主要任务的泛化程度；几乎与多任务学习（MTL）相同
  - 非对称多任务学习（Asymmetric MTL）：当多个任务在进行联合训练时，有新任务加入
- 学会学习（Learning-to-learn, LTL/元学习 Meta learning）
  - LTL的目标是利用解决先验任务时获得的知识来处理新任务

表达式

$$\min_{\boldsymbol{W}=[\boldsymbol{w}^1 \ldots \boldsymbol{w}^M]} \sum_{m=1}^{M} L(\boldsymbol{X}^m, \boldsymbol{y}^m, \boldsymbol{w}^m) + \lambda \mathrm{Reg}(\boldsymbol{W})$$

$\boldsymbol{X}^m \in \mathbb{R}^{N_m \times D}$	第m个任务的输入矩阵
$\boldsymbol{y}^m \in \mathbb{R}^{N_m \times 1}$	相应的输出向量
$\boldsymbol{y}^m \approx \boldsymbol{X}^m \boldsymbol{w}^m$	对于回归问题
N_m, D, M	样本数量，特征数量，任务数量

概念

- 不同任务是否具有相同特征空间？
  - 同构特征多任务学习
  - 异构特征多任务学习
- 是否为不同类型的任务？
  - 例如：多分类，无监督学习，半监督学习，强化学习……
  - 同构多任务学习
  - 异构多任务学习
- 默认情况下：同构
  - 任务的特征和参数空间是相同的

多任务学习中的问题

- 什么情况下在任务间共享：单任务或多任务？
  - 人类专家
  - 模型选择问题
  - 使用可退化为单任务模型的多任务模型
- 在任务间共享什么：通过什么途径进行共享？
  - 特征层次（基于特征）
  - 实例层次（基于实例）
  - 任务层次（基于参数）

多任务学习中的问题

- 如何进行共享：共享知识的具体方法
  - 特征层次
    - 特征选择
    - 特征转换
  - 实例层次
  - 任务层次
    - 任务聚类
    - 低秩
    - 任务关系学习
  - 使用深度学习的多任务学习
    - 硬参数共享
    - 软参数共享
    - 共享-私有方法
    - 神经结构搜索等

表达式

$$\min_{\boldsymbol{W}=[\boldsymbol{w}^1 \ldots \boldsymbol{w}^M]} \sum_{m=1}^{M} L(\boldsymbol{X}^m, \boldsymbol{y}^m, \boldsymbol{w}^m) + \lambda \mathrm{Reg}(\boldsymbol{W})$$

- 数据保真度项
  - 计算目标预测与真实目标的匹配程度
- 正则化项
  - 正则化权重矩阵以获得不同任务之间的关系

多任务学习的典型例子

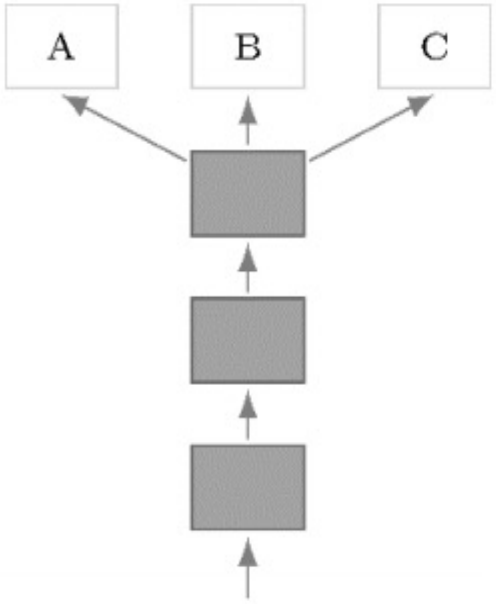
- 最常见的多任务学习目标函数
  - 对于两个任务，正则化任务参数

任务1的损失函数　　任务2的损失函数

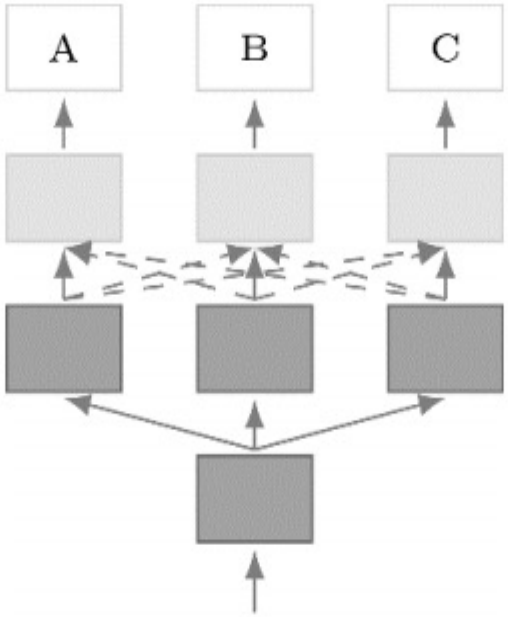
$$\min_{w^1,w^2} \quad \alpha L\left(\mathbf{Y}^1, f_{w^1}(\mathbf{X}^1)\right) + (1-\alpha) L\left(\mathbf{Y}^2, f_{w^2}(\mathbf{X}^2)\right)$$
$$+\lambda_1(\|\boldsymbol{w}^1\|^2 + \|\boldsymbol{w}^2\|^2) + \lambda_2\|\boldsymbol{w}^1 - \boldsymbol{w}^2\|^2$$

标准正则化　　软参数共享

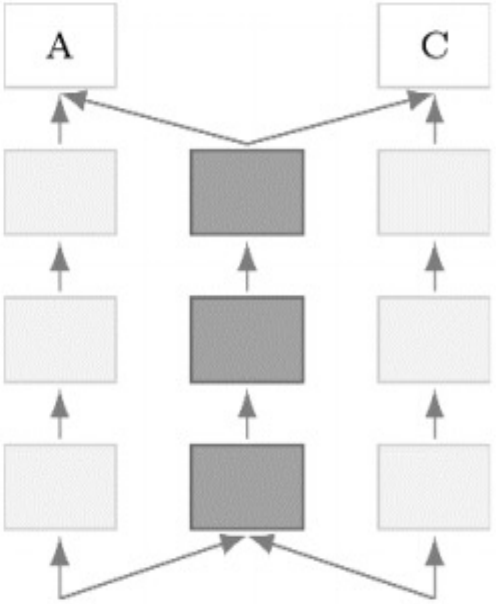
主要深度多任务学习方法



硬参数共享



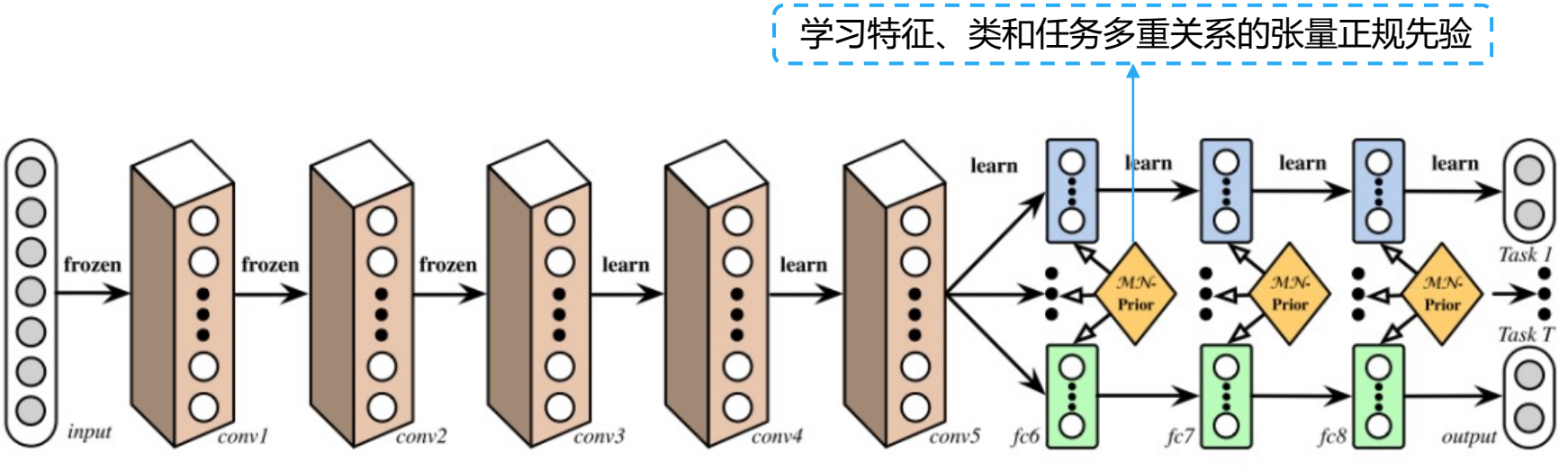
软参数共享



共享-私有方法

共享-私有方法

- 深度关系网络



- 共享-私有的部分：贝叶斯模型，矩阵先验
- 依赖于预定义的共享结构

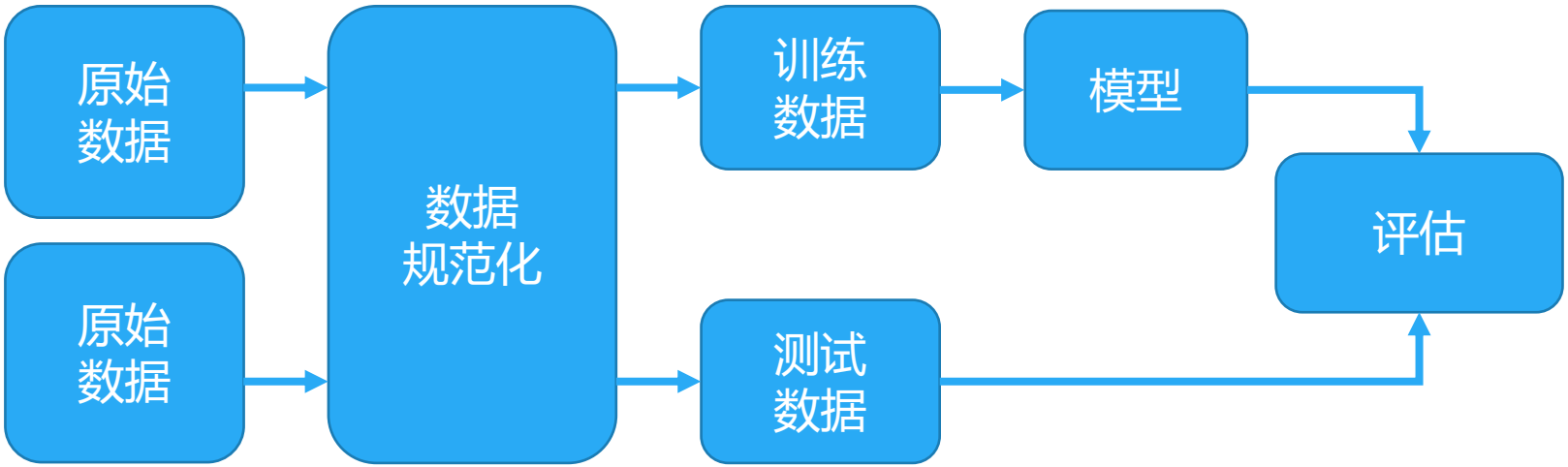
Long, Mingsheng, et al. "Learning Transferable Features with Deep Adaptation Networks." *ICML* 2015.

迁移学习

张伟楠 – 上海交通大学

机器学习基本假设

$$\min_{\theta} \frac{1}{N} \sum_{(x_i,y_i)\in D_{\text{train}}} \mathcal{L}\big(y_i, f_\theta(x_i)\big) + \lambda \|\theta\|_2^2$$



$$\text{Test Error} = \frac{1}{N} \sum_{(x_i,y_i)\in D_{\text{test}}} \mathcal{L}\big(y_i, f_\theta(x_i)\big)$$

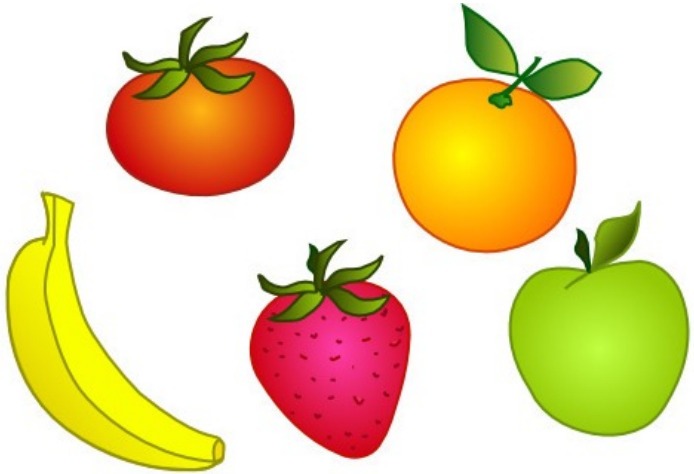
- 基本假设：训练和测试数据具有相同分布

实际案例

- 数据分布$p(x)$随任务知识域的不同而变化或随时间发生变化

$$\mathcal{X}_S \neq \mathcal{X}_T \text{ or } p_S(x) \neq p_T(x)$$

真实图片



卡通图片

实际案例

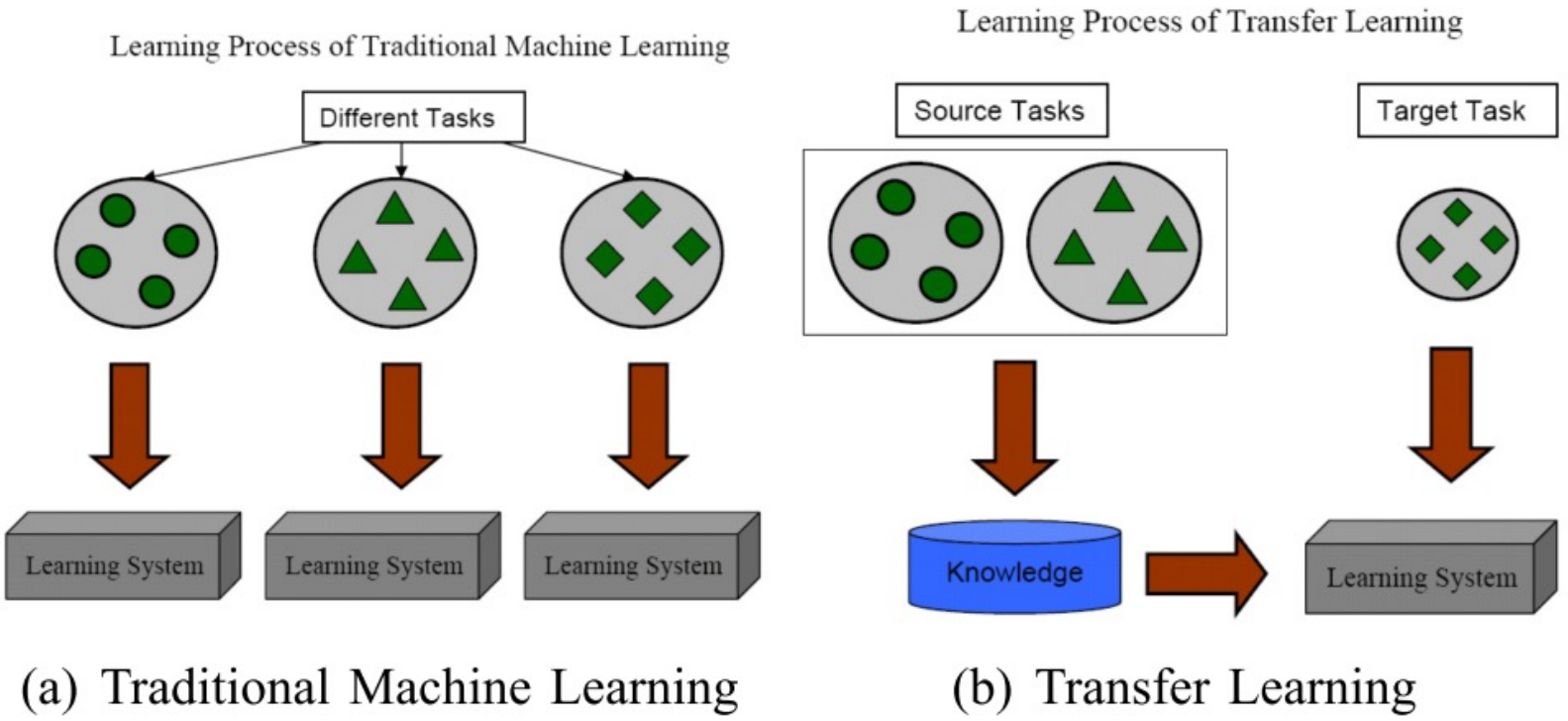
- 数据依赖$p(y|x)$也可能不同

$$\mathcal{Y}_S \neq \mathcal{Y}_T \text{ or } p_S(y|x) \neq p_T(y|x)$$

识别苹果

识别梨

迁移学习


Learning Process of Traditional Machine Learning
Different Tasks
Learning System
Learning System
Learning System
(a) Traditional Machine Learning
Learning Process of Transfer Learning
Source Tasks
Target Task
Knowledge
Learning System
(b) Transfer Learning

迁移学习的符号表和定义

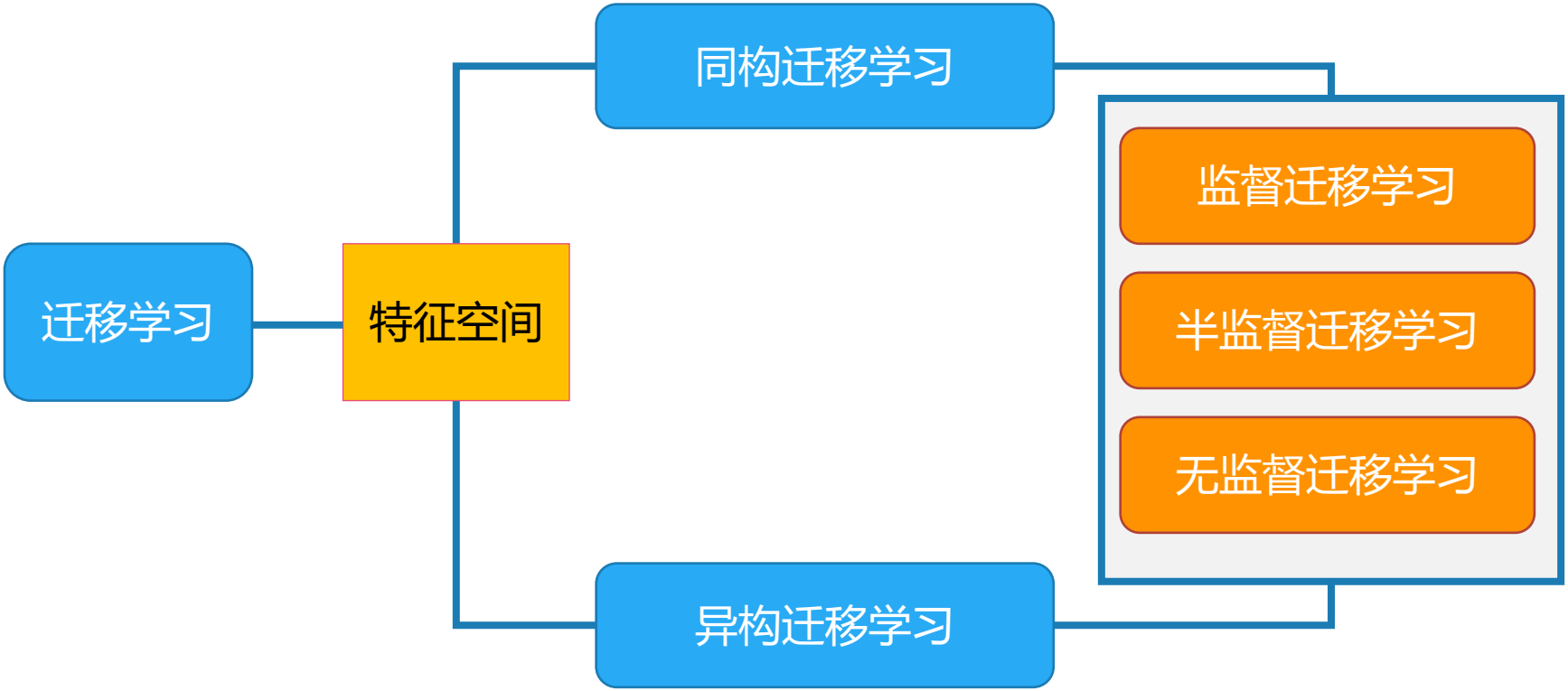
- 符号表
  - 一个**领域（domain）** $\mathcal{D} = \{\mathcal{X}, p(x)\}$
    - 特征空间$\mathcal{X}$
    - 数据分布$p(x)$
  - 一个**任务（task）** $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$
    - 标签空间$\mathcal{Y}$
    - 目标预测函数$f(\cdot)$
- 定义
  - 给定**源域（source domain）**$\mathcal{D}_S$和对应的学习任务$\mathcal{T}_S$，以及**目标域（target domain）**$\mathcal{D}_T$和对应的学习任务$\mathcal{T}_T$
  - **迁移学习**是通过使用$\mathcal{D}_S$和$\mathcal{T}_S$的相关信息来改进目标域预测函数$f(\cdot)$的学习过程，其中$\mathcal{D}_s \neq \mathcal{D}_T$或者$\mathcal{T}_s \neq \mathcal{T}_T$

一些解释

- 数据域不同 $\mathcal{D}_S \neq \mathcal{D}_T$
  - 数据空间不同 $\mathcal{X}_S \neq \mathcal{X}_T$
    - 异构迁移学习
    - 两份使用不同语言的文件
  - 数据分布不同 $P(X_S) \neq P(X_T)$
    - 领域自适应
    - 两份侧重于不同主题的文件
- 任务不同 $\mathcal{T}_S \neq \mathcal{T}_T$
  - 预测标签不同 $\mathcal{Y}_S \neq \mathcal{Y}_T$
    - 源域有两个类别：正例和负例；目标域新增一个类别：中性
  - 映射不同 $P_S(y|x) \neq P_T(y|x)$
    - 同一个词在两个领域中可以有不同的含义

迁移学习背景

- 同构/异构迁移学习



迁移学习方法

- 实例迁移（Instance Transfer）
  - 重新调整源域中实例的权重应用于目标域的数据
- 特征迁移（Feature Transfer）
  - 把源域和目标域的特征映射到一个共同的空间中
- 参数迁移（Parameter Transfer）
  - 根据源域模型学习目标域模型的参数

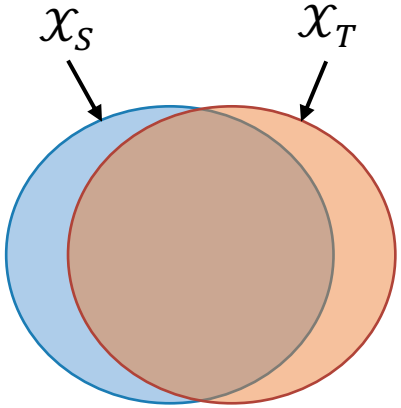
基于实例的迁移学习

- 基本假设
  - 源域和目标域具有大量重叠的特征或者共享相同的特征空间

$$x_S \simeq x_T$$

  - 标签空间是相同的

$$y_S \simeq y_T$$



- 应用举例
  - 不同科室的电子病历
  - 不同主题的情感分析

实例迁移学习 例1：领域自适应

- 问题设置
  - 给定带有标签的源域数据 $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$，目标域数据 $D_T = \{x_{T_i}\}_{i=1}^{n_T}$
  - 学习f_T使得目标数据上的损失最小

$$\sum_i \mathcal{L}(f_T(x_{T_i}), y_{T_i})$$

  - 其中y_{T_i}是未知的
- 假设
  - 相同的标签空间 $\mathcal{Y}_S = \mathcal{Y}_T$
  - 相同的依赖关系 $p(y_S|x_S) = p(y_T|x_T)$
  - （几乎）相同的特征空间 $\mathcal{X}_S \simeq \mathcal{X}_T$
  - 不同数据分布 $p_S(x) \neq p_T(x)$

领域自适应中的重要性采样

- 重要性采样

$$\begin{aligned}\theta^* &= \arg\min_\theta \mathbb{E}_{(x,y)\sim p_T}\left[\mathcal{L}\left(y, f_\theta(x)\right)\right] \\ &= \arg\min_\theta \int_{(x,y)} p_T(x)\mathcal{L}\left(y, f_\theta(x)\right)dx \\ &= \arg\min_\theta \int_{(x,y)} p_S(x)\frac{p_T(x)}{p_S(x)}\mathcal{L}\left(y, f_\theta(x)\right)dx \\ &= \arg\min_\theta \mathbb{E}_{(x,y)\sim p_S}\left[\frac{p_T(x)}{p_S(x)}\mathcal{L}\left(y, f_\theta(x)\right)\right]\end{aligned}$$

- 通过下式重新调整每个实例的权重

$$\beta(x) = \frac{p_T(x)}{p_S(x)}$$

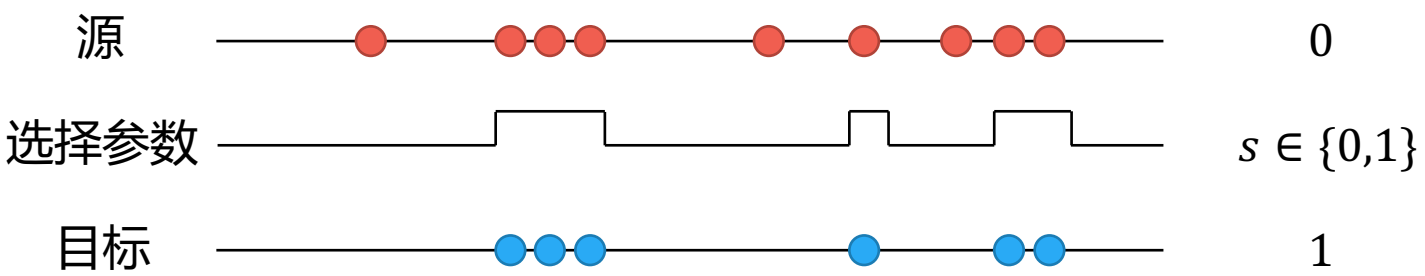
领域自适应中的重要性采样

如何估计$\beta(x) = \frac{p_T(x)}{p_S(x)}$

- 最简单的解决方法是先分别估计$p_S(x)$和$p_T(x)$，然后计算$\beta(x)$
  - 可能会面临高方差问题
- 一个更实用的解决方法是直接估计$\frac{p_T(x)}{p_S(x)}$

领域自适应中的重要性采样

- 设想一个拒绝采样过程，并且把目标域看作从源域采样获得的



- 概率密度函数（p.d.f.）关系

$$p_T(x) \propto p_s(x)p(s=1|x)$$

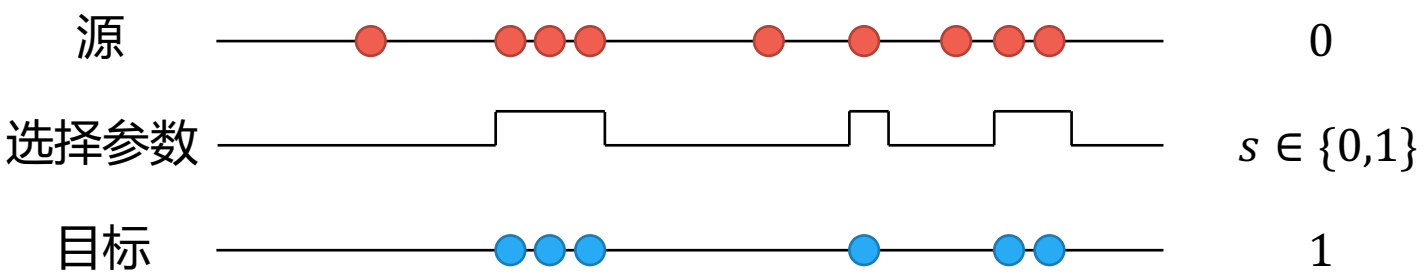
- 且我们把$p(s=1|x)$估计为一个二分类模型

$$\beta(x) = \frac{p_T(x)}{p_S(x)} \propto p(s=1|x)$$

Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004

领域自适应中的重要性采样

- 设想一个拒绝采样过程，并且把目标域看作从源域采样获得的



- 把$p(s=1|x)$估计为一个二分类模型
  - 将目标域中的实例标记为1
  - 将源域中的实例标记为0

$$\beta(x) = \frac{p_T(x)}{p_S(x)} \propto p(s=1|x)$$

Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004

领域自适应中的重要性采样

如何估计$\beta(x) = \frac{p_T(x)}{p_S(x)}$

- 用一系列基函数构建估计量

$$\hat{\beta}(x) = \sum_{l=1}^{b} \alpha_l \psi_l(x)$$

- 被估计的目标概率密度函数 $\hat{p}_T(x) = \hat{\beta}(x) p_S(x)$

- 最小化KL散度

$$\min_{\{\alpha_l\}_{l=1}^{b}} \mathrm{KL}[p_T(x) || \hat{p}_T(x)]$$

Sugiyama *et al*., Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation, NIPS 2007

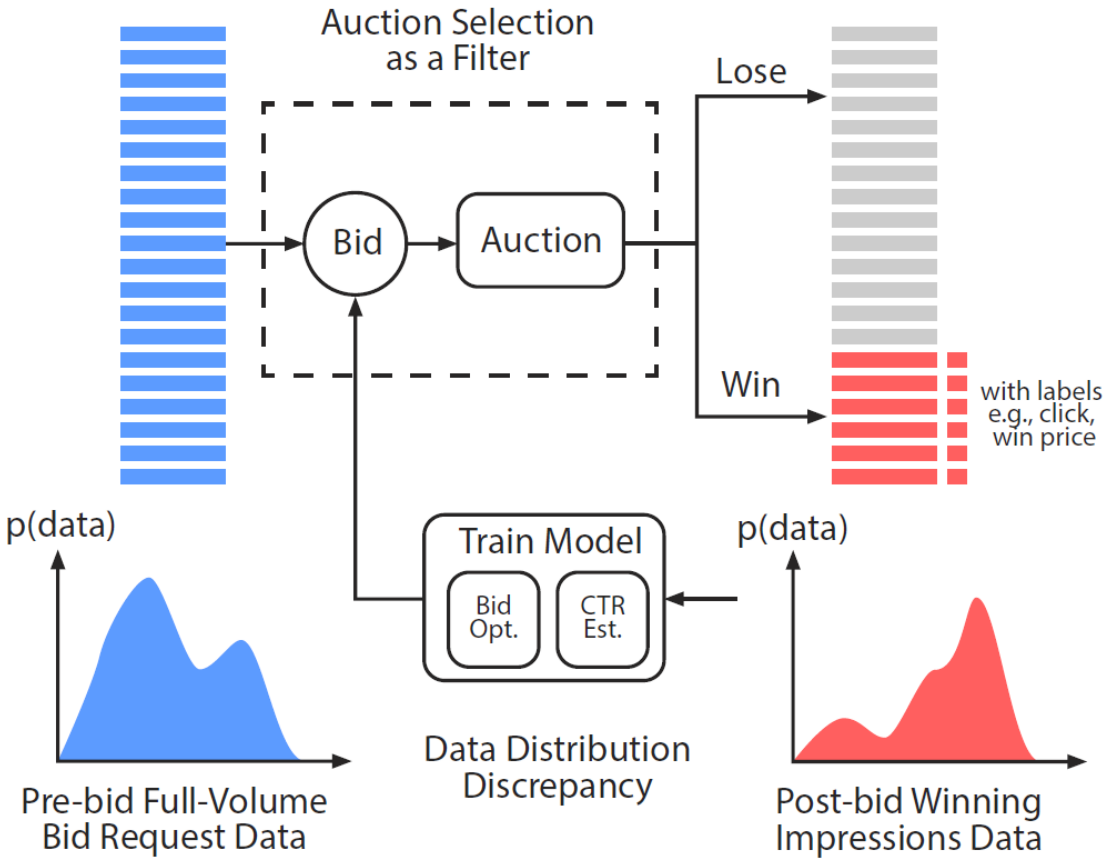
- 最小化平方误差

$$\min_{\{\alpha_l\}_{l=1}^{b}} \int_x \left(\hat{\beta}(x) - \beta(x)\right)^2 p_S(x) dx$$

Kanamori et al., A Least-squares Approach to Direct Importance Estimation, JMLR 2009

展示型广告中的无偏训练

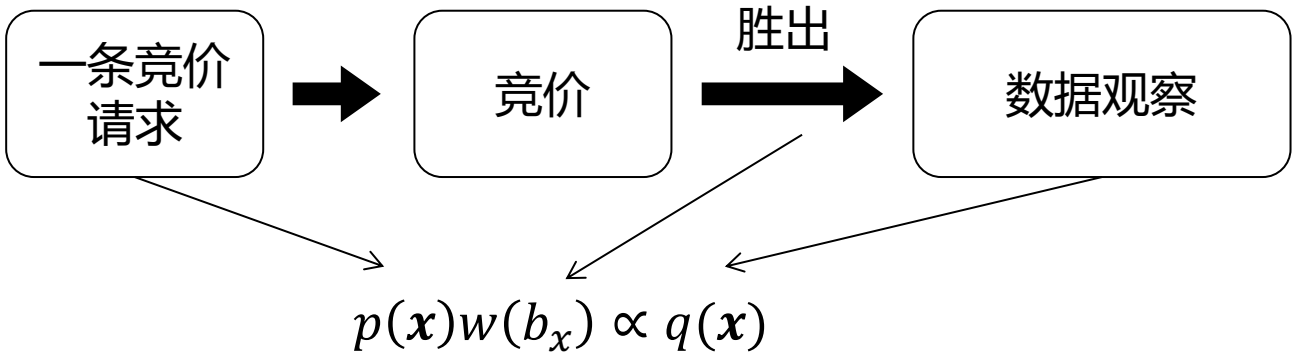
- 展示型广告中，标签数据只有当广告商赢得竞拍之后才能被广告商所观测到，所以是有偏的（biased）



Weinan Zhang et al. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. KDD 16

无偏学习框架

- 数据观测过程



- 重要性采样

$$\min_{\beta} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathcal{L}\left(y, f_\beta(\boldsymbol{x})\right)\right] = \min_{\beta} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\frac{\mathcal{L}\left(y, f_\beta(\boldsymbol{x})\right)}{w(b_x)}\right]$$

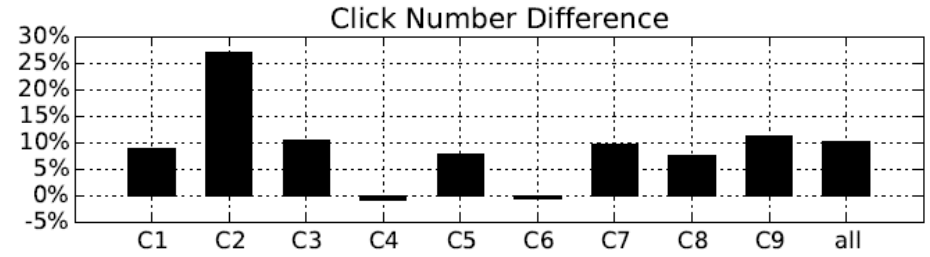
Weinan Zhang et al. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. KDD 16

与Yahoo！DSP的表现比较

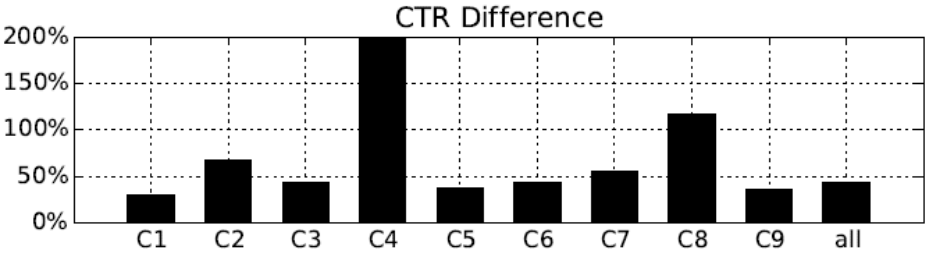
- 在Yahoo！美国上的A/B测试

2.97% AUC提升

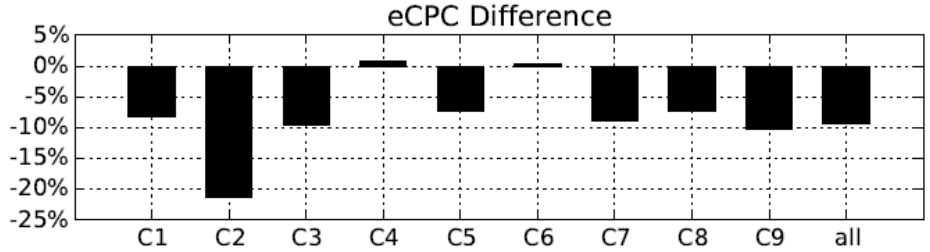
Camp.	BIAS AUC.	KMMP AUC	AUC Lift
C1	63.78%	64.12%	0.34%
C2	87.45%	88.58%	1.13%
C3	69.73%	75.52%	5.79%
C4	88.82%	89.55%	0.73%
C5	69.71%	72.29%	2.58%
C6	89.33%	90.70%	1.37%
C7	77.76%	78.92%	1.16%
C8	74.57%	76.98%	2.41%
C9	71.04%	73.12%	2.08%
all	73.48%	76.45%	2.97%



10.3% 更多点击数



42.8% 更高CTR



9.3% 更低eCPC

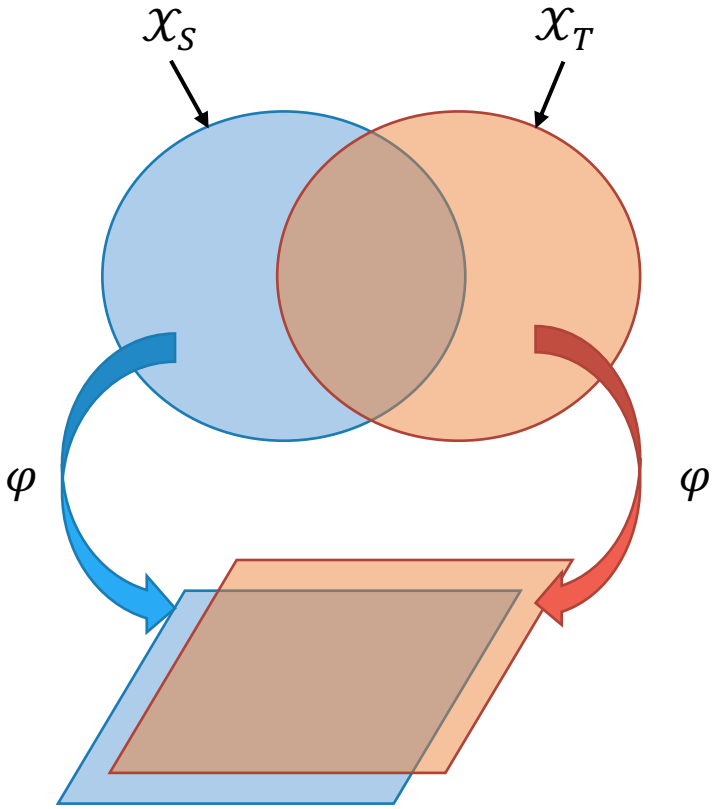
Weinan Zhang et al. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. KDD 16

迁移学习方法

- 实例迁移（Instance Transfer）
  - 重新调整源域中实例的权重应用于目标域的数据
- 特征迁移（Feature Transfer）
  - 把源域和目标域的特征映射到一个共同的空间中
- 参数迁移（Parameter Transfer）
  - 根据源域模型学习目标域模型的参数

基于特征的迁移学习

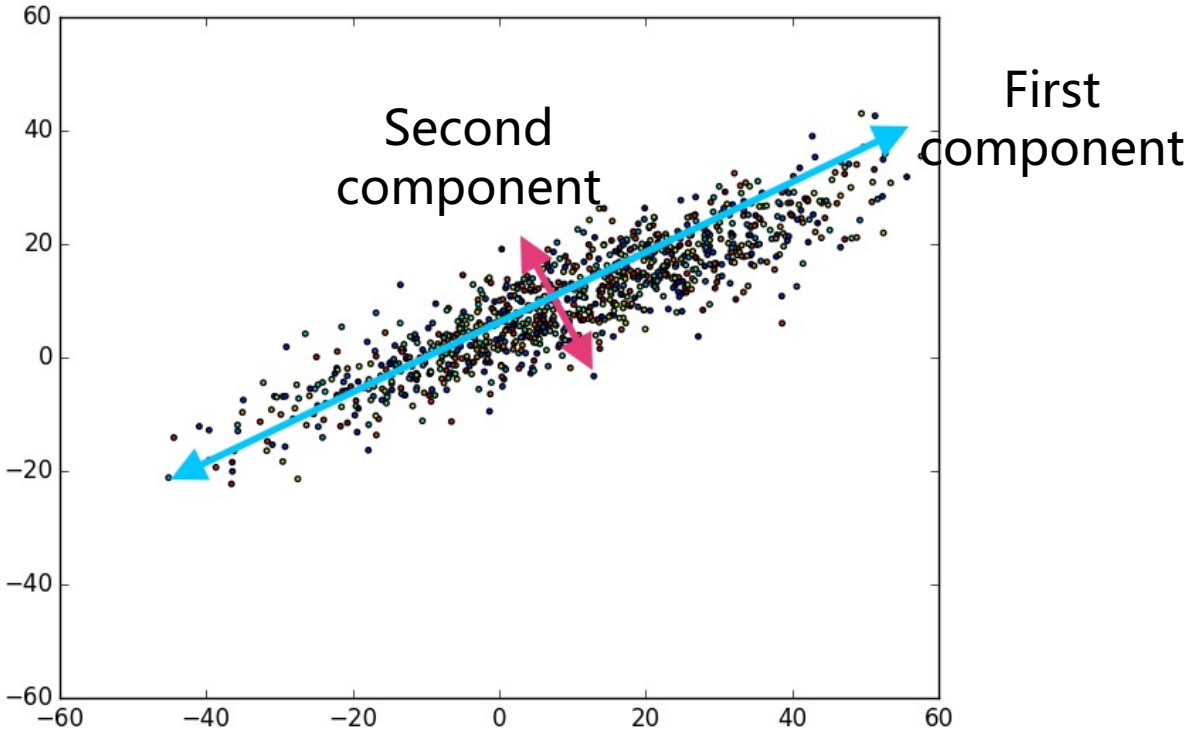
- 当源域和目标域只有部分重叠
  - 许多特征仅在源域（或）目标域中存在
- 可能的解决方法
  - 编码指定应用场景的先验知识
  - 学习出进行迁移的一般映射φ



基于特征的通用迁移学习方法

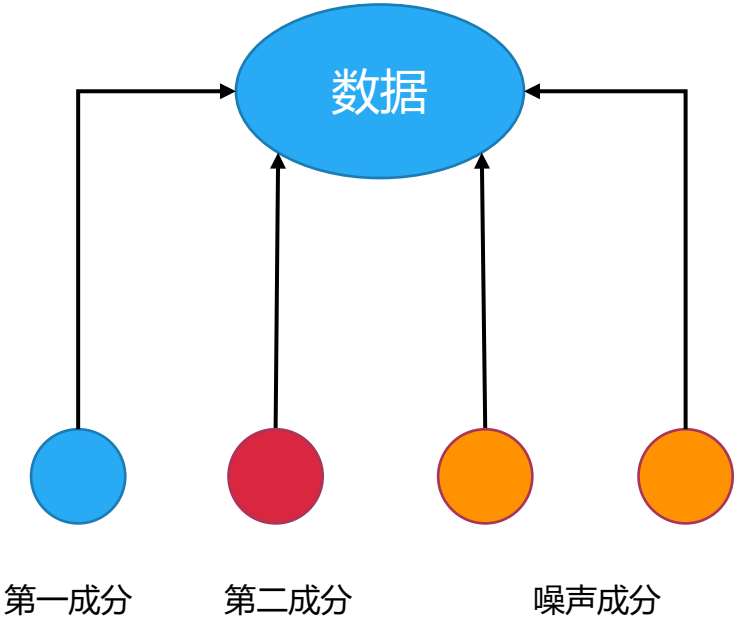
- 通过最小化两个领域分布之间的距离来学习新的数据表示
- 通过多任务学习（multi-task learning）来学习新的数据表示
- 通过自学习（self-taught learning）来学习新的数据表示

主成分分析（PCA）



- PCA使用正交变换，将一系列可能相关的变量观测值转换为一系列称为主成分的线性不相关变量值

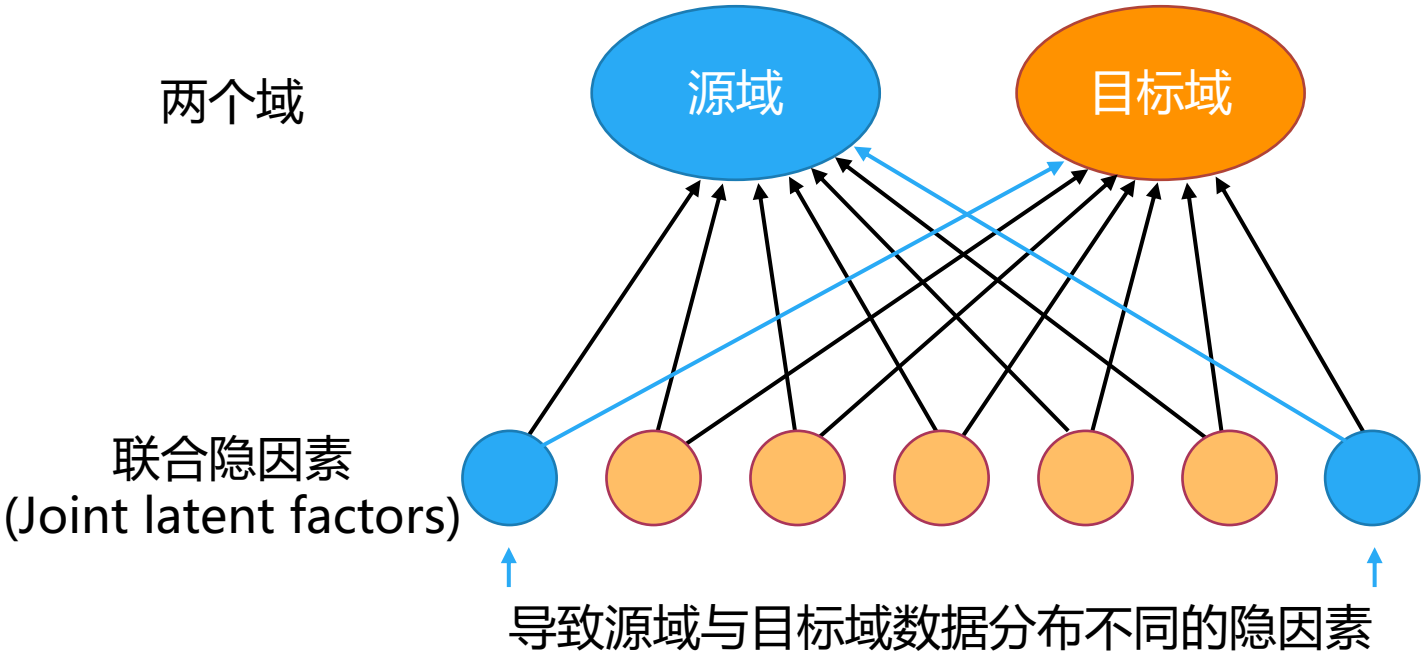
主成分分析（PCA）



- PCA使用正交变换，将一系列可能相关的变量观测值转换为一系列称为主成分的线性不相关变量值

迁移成分分析(Transfer Component Analysis)

- 动机
  - 通过将数据投影到学习的迁移成分（transfer components）上，将领域分布之间的距离最小化



Pan, Sinno Jialin, et al. "Domain adaptation via transfer component analysis." *IEEE Transactions on Neural Networks* 22.2 (2011): 199-210.

迁移成分分析(Transfer Component Analysis)

- 主要思想
  - 学习φ将源域和目标域数据映射到可以减少领域差异并保留原始数据结构的隐空间

$$\min_{\varphi} \mathrm{Dist}\left(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)\right) + \lambda\Omega(\varphi)$$

$$\text{s. t. constraints on } \varphi(\mathbf{X}_S) \text{ and } \varphi(\mathbf{X}_T)$$

迁移成分分析(Transfer Component Analysis)

- 最大平均差异（MMD）
  - 给定源域和目标域数据

$$\mathbf{X}_S = \{x_{S_i}\}_{i=1}^{n_S} \quad \mathbf{X}_T = \{x_{T_i}\}_{i=1}^{n_T}$$

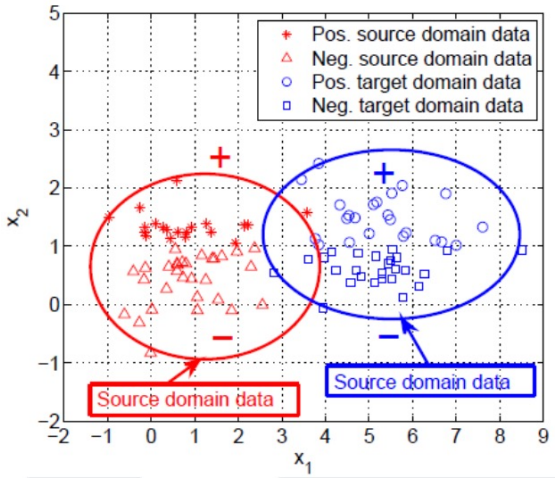
  - 分别从$P_S(x)$和$P_T(s)$抽样

$$\mathrm{Dist}\big(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)\big) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \Phi\left(\varphi\left(x_{S_i}\right)\right) - \frac{1}{n_T} \sum_{i=1}^{n_T} \Phi\left(\varphi\left(x_{T_i}\right)\right) \right\|_{\mathcal{H}}$$

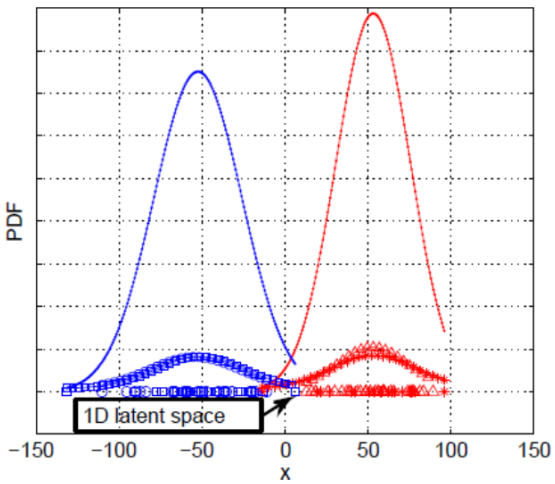
映射　　核函数

迁移成分分析(Transfer Component Analysis)

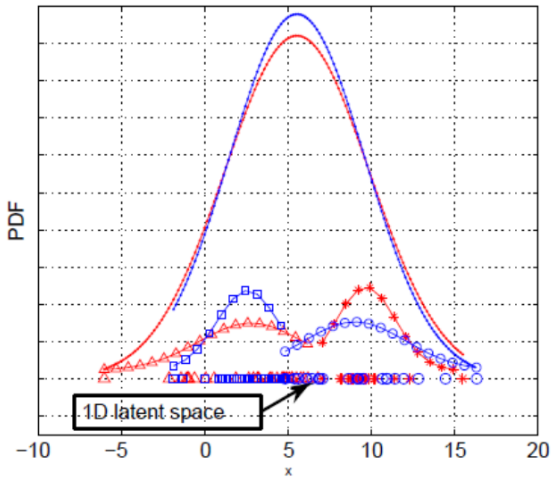
- 通过PCA和TCA学习隐特征的可视化示例



原始特征空间



PCA



TCA

参数迁移

张伟楠 – 上海交通大学

迁移学习方法

- 实例迁移（Instance Transfer）
  - 重新调整源域中实例的权重应用于目标域的数据
- 特征迁移（Feature Transfer）
  - 把源域和目标域的特征映射到一个共同的空间中
- 参数迁移（Parameter Transfer）
  - 根据源域模型学习目标域模型的参数

基于参数的迁移学习

- 从两个域上学习的θ-参数化函数

$$\theta_S^* = \arg\min_\theta \sum_{i=1}^{n_S} \mathcal{L}\left(y_{S_i}, f_\theta\left(x_{S_i}\right)\right) + \lambda\Omega(\theta)$$

$$\theta_T^* = \arg\min_\theta \sum_{i=1}^{n_T} \mathcal{L}\left(y_{T_i}, f_\theta\left(x_{T_i}\right)\right) + \lambda\Omega(\theta)$$

- 动机
  - 一个充分训练好的模型$f_{\theta_S^*}(x)$已经学习到了源域上的许多结构特征
  - 如果两个任务是相关的，这一结构可以被迁移用于学习目标域上的模型$f_{\theta_T^*}(x)$

多任务或协作学习

- 最小化两个任务上的联合损失函数和模型参数之间的距离

$$\min_{\theta_S,\theta_T} \alpha \frac{1}{N_S}\sum_{i=1}^{N_S}\mathcal{L}\left(y_i, f_{\theta_S}(x_i)\right) + (1-\alpha)\frac{1}{N_T}\sum_{j=1}^{N_T}\mathcal{L}\left(y_i, f_{\theta_T}(x_j)\right) + \lambda\Omega(\theta_S,\theta_T)$$

源任务损失　　目标任务损失　　参数距离

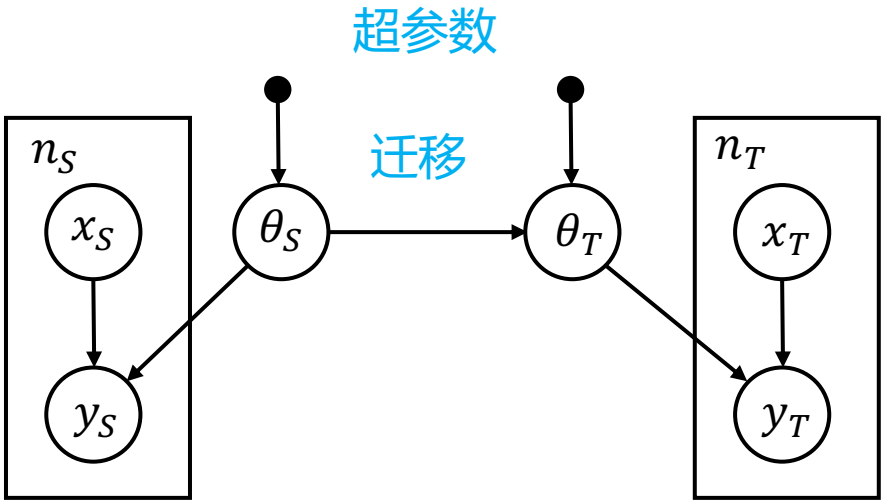
- 参数距离的不同定义

$$\Omega(\theta_S,\theta_T) = \|\theta_S - \theta_T\|^2$$

$$\Omega(\theta_S,\theta_T) = \sum_{t\in\{S,T\}}\left\|\theta_t - \frac{1}{2}\sum_{s\in\{S,T\}}\theta_s\right\|^2$$

分层贝叶斯网络

- 思想：源域参数被视为随机变量，充当目标域参数的先验



案例分析：从网页浏览到广告点击

- 源任务
  - 数据：用户浏览网页id数据
  - 任务：预测一名用户是否会愿意点开一个网页
- 目标任务
  - 数据：用户浏览网页id数据
  - 任务：预测一名用户是否会愿意点击一条广告

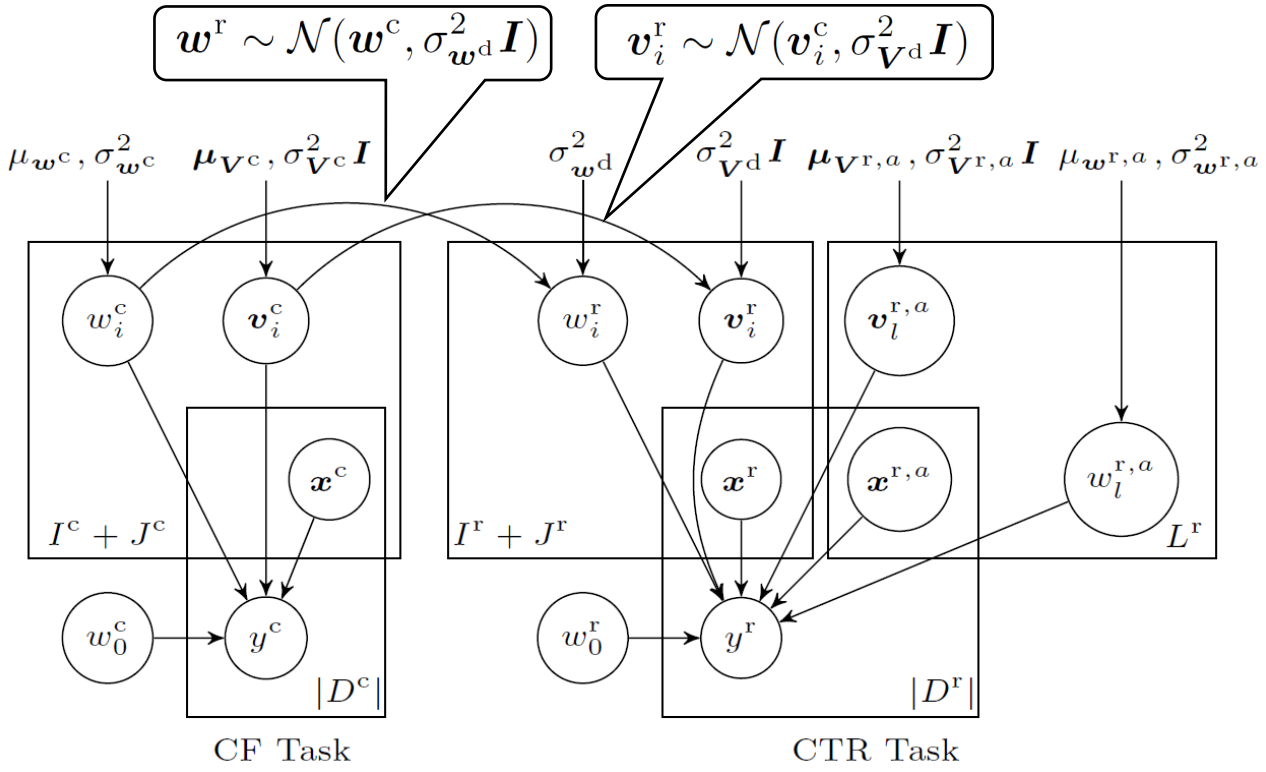
$$\min_{\theta_S,\theta_T} \alpha \frac{1}{N_S}\sum_{i=1}^{N_S}\mathcal{L}\left(y_i, f_{\theta_S}(x_i)\right) + (1-\alpha)\frac{1}{N_T}\sum_{j=1}^{N_T}\mathcal{L}\left(y_i, f_{\theta_T}(x_j)\right) + \lambda\Omega(\theta_S,\theta_T)$$

逻辑回归　　逻辑回归

Perlich, Claudia, et al. "Machine learning for targeted display advertising: Transfer learning in action." *Machine learning* 95.1 (2014): 103-127.

案例分析：从网页浏览到广告点击

- 以贝叶斯分层图模型为例说明



Zhang, Weinan et al. Implicit Look-alike Modelling in Display Ads: Transfer Collaborative Filtering to CTR Estimation. ECIR 2016.

元学习

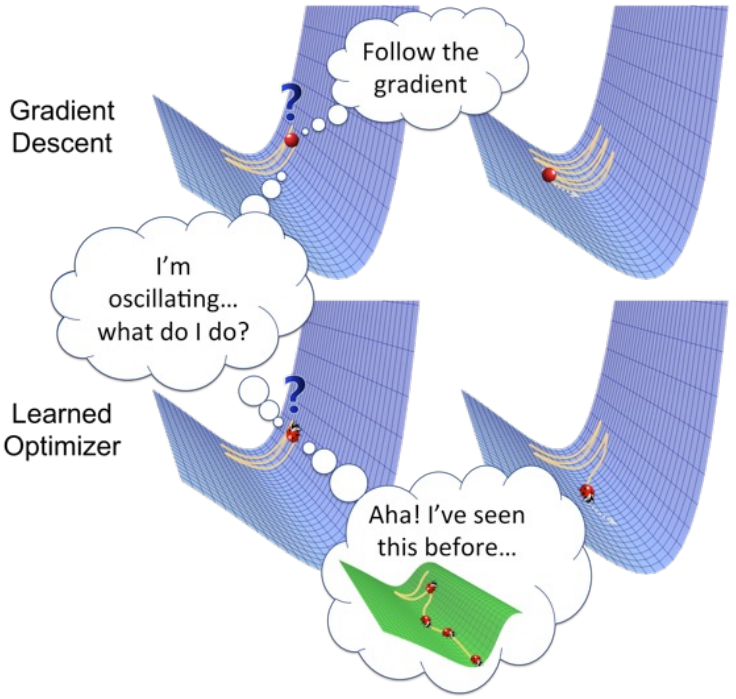
张伟楠 – 上海交通大学

什么是元学习？

- 如果你已经学习了100个任务，给定一个新任务，你能想出如何更有效地学习吗？
  - 拥有多任务现在成为了巨大优势

- 元学习（Meta-Learning）与多任务学习（multi-task learning）非常接近

- 元学习 = 学会学习（Meta-learning = learning to learn）



https://bair.berkeley.edu/blog/2017/09/12/learning-to-optimize-with-rl/

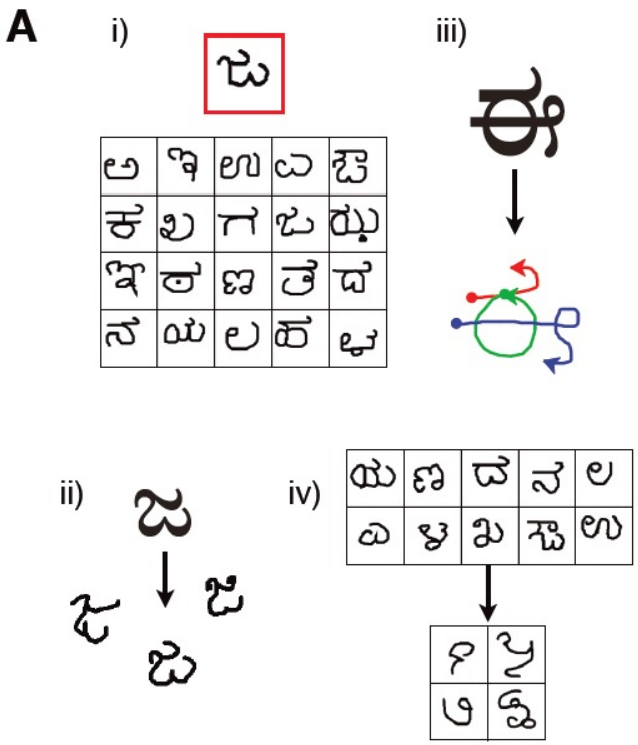
Slide credit: Sergey Levine

元学习的早期方法

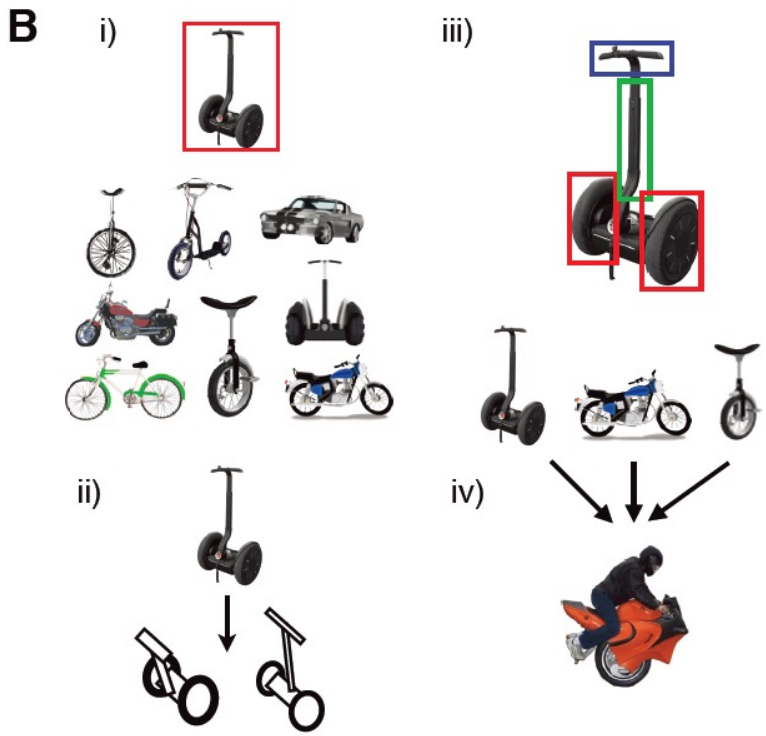
- Jürgen Schmidhuber
  - Genetic Programming. PhD thesis. 1987
  - Learning to control fast-weight memories: An alternative to dynamic recurrent networks. Neural Computation 1992
  - A neural network that embeds its own meta-levels. ICNN 1993
- Yoshua Bengio
  - Learning a synaptic learning rule. Univ. Montreal. 1990
  - On the search for new learning rules for ANN. Neural Processing Letters 1992
    - 使用SGD学习更新规则

元学习超越了传统的机器学习

- Lake等人有力地论证了元学习作为人工智能的基石的重要性



人类级别的新型手写字符学习



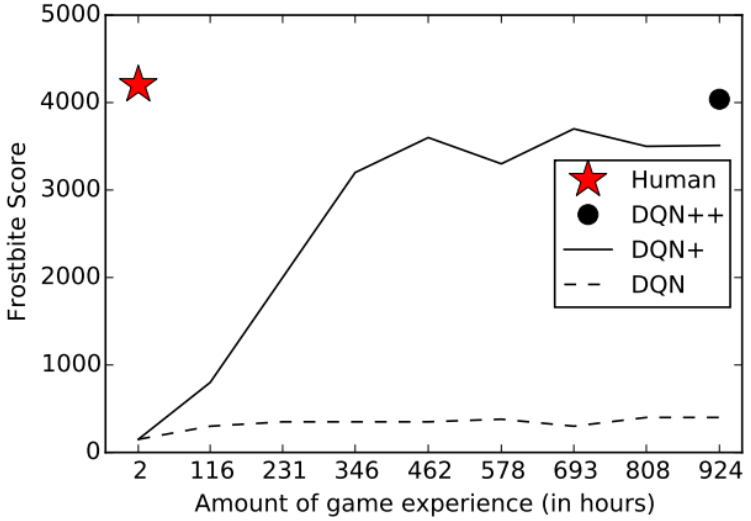
在新型两轮交通工具中也展现出相同的能力

Lake, Brenden M., et al. "Building machines that learn and think like people." Behavioral and Brain Sciences 40 (2017).

元学习超越了传统的机器学习

- Lake等人有力地论证了元学习作为人工智能基石的重要性

Atari 2600游戏Frostbite

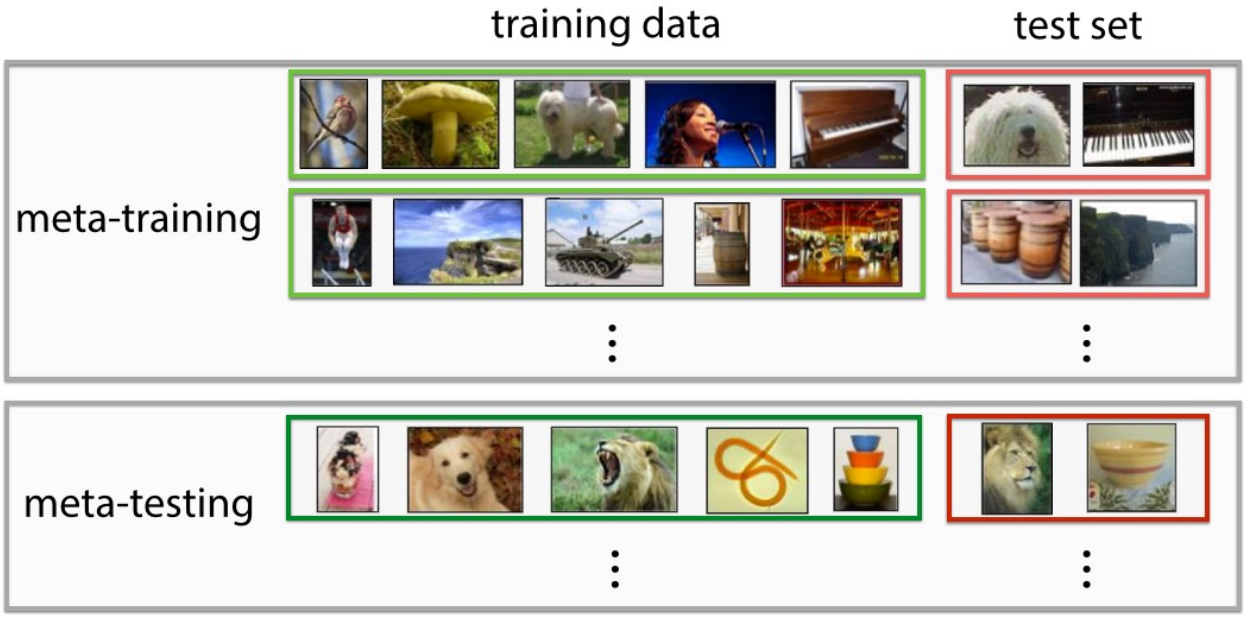


测试表现

人类知道如何进行学习，但是强化学习方法不能做到

Lake, Brenden M., et al. "Building machines that learn and think like people." Behavioral and Brain Sciences 40 (2017).

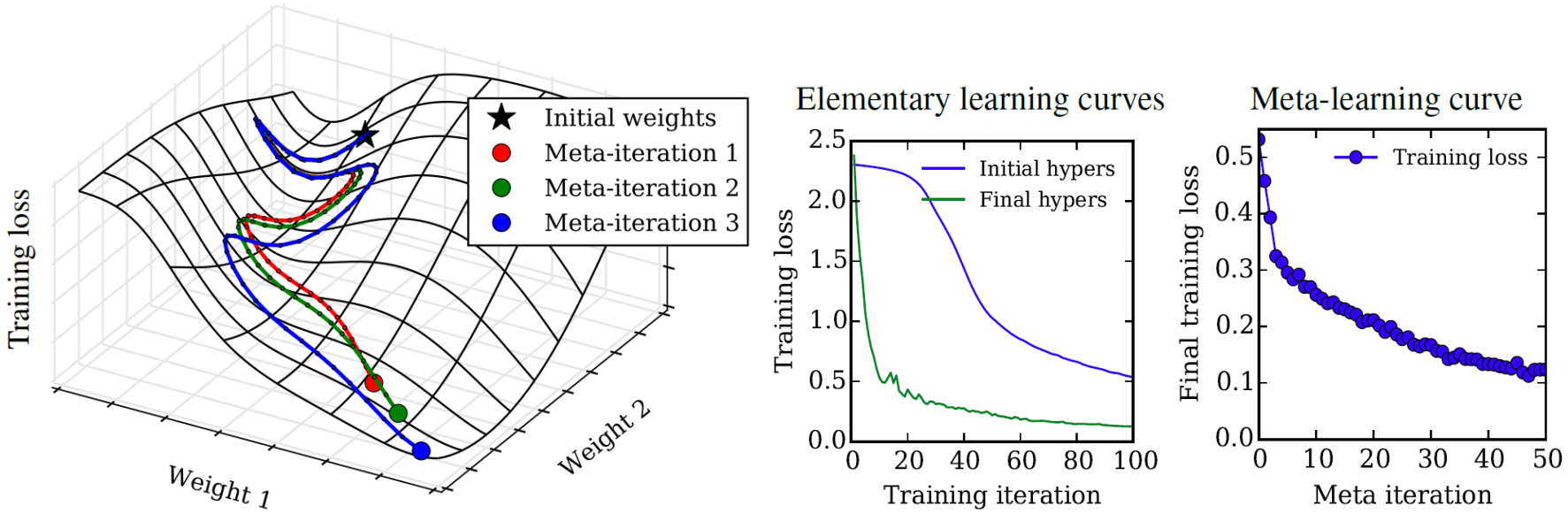
元学习的一般范式



元学习在小样本（few-shot）图像分类任务中的示例

- 在元学习过程中，模型在元训练集（meta-training set）中通过训练来学习任务。优化分为两个部分：
  - 学习器：学习新任务
  - 元学习器：训练学习器

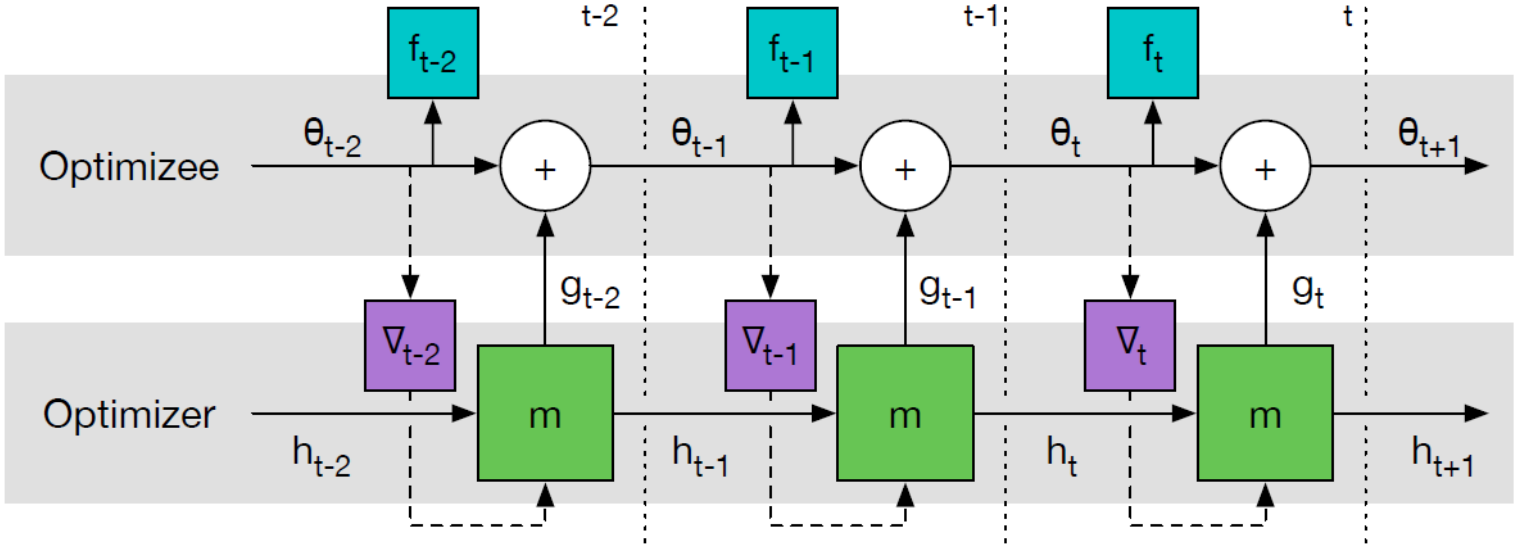
各种元学习任务



- 超参数优化
  - 通过在整个训练过程中反向链式求导，计算所有超参数下交叉验证表现的准确梯度

Maclaurin et al. Gradient-based Hyperparameter Optimization through Reversible Learning. ICML 2015.

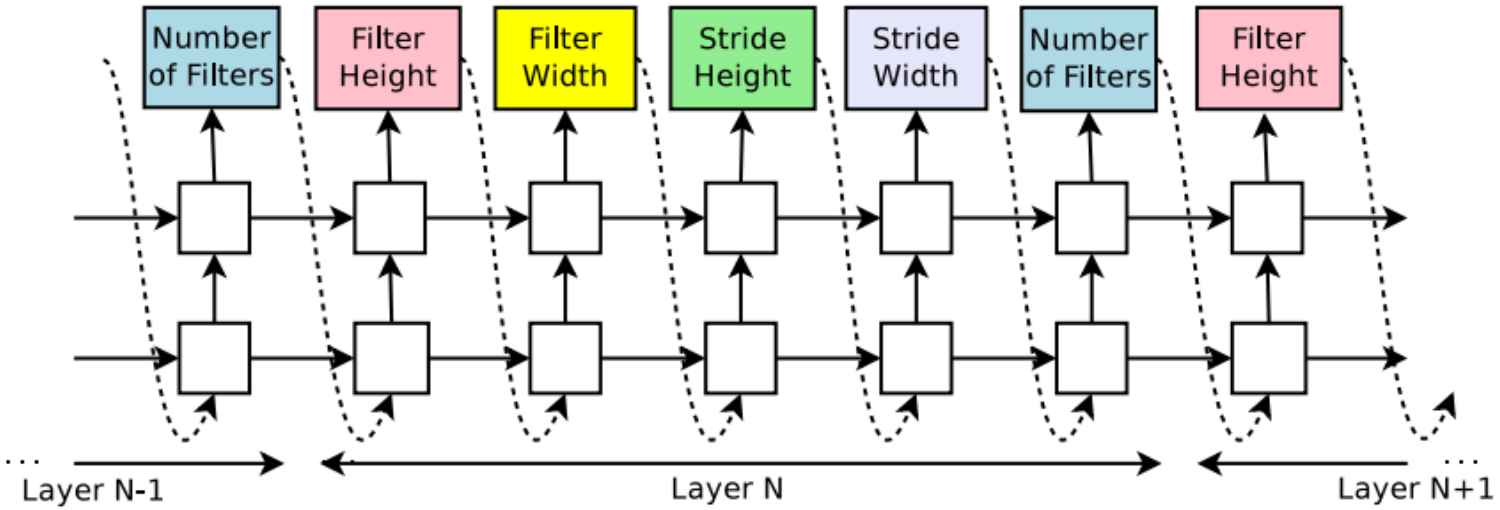
各种元学习任务



- 学会生成良好的梯度
  - 一个带有隐藏记忆单元的RNN，接收新的原始梯度，并输出一个调整后的梯度，以便更好地训练模型。

Andrychowicz et al. Learning to learn by gradient descent by gradient descent. ICML 2015.

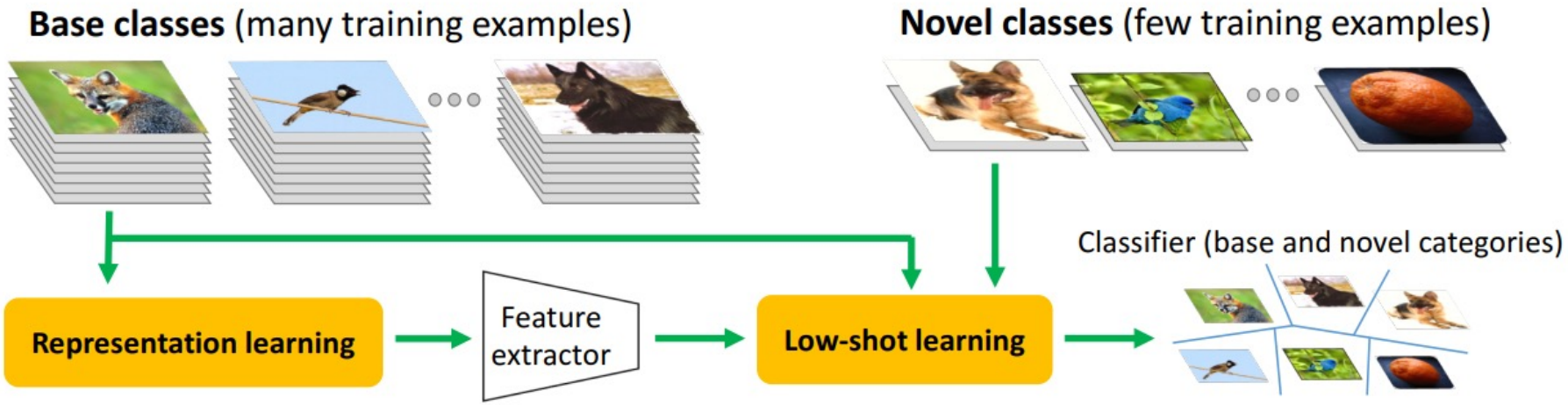
各种元学习任务



- 自动搜索良好的网络结构
  - 一个强化学习过程：选择动作以创建网络结构，并在数据集上训练和评估所建的网络以获得奖励

Barret Zoph and Quoc V. Le. Neural architecture search by reinforcement learning. ICLR 2017.

各种元学习任务



- 小样本学习（Few-shot learning）
  - 从大量数据集中学习模型，使其能够轻松适用于具有少量数据的新的类别

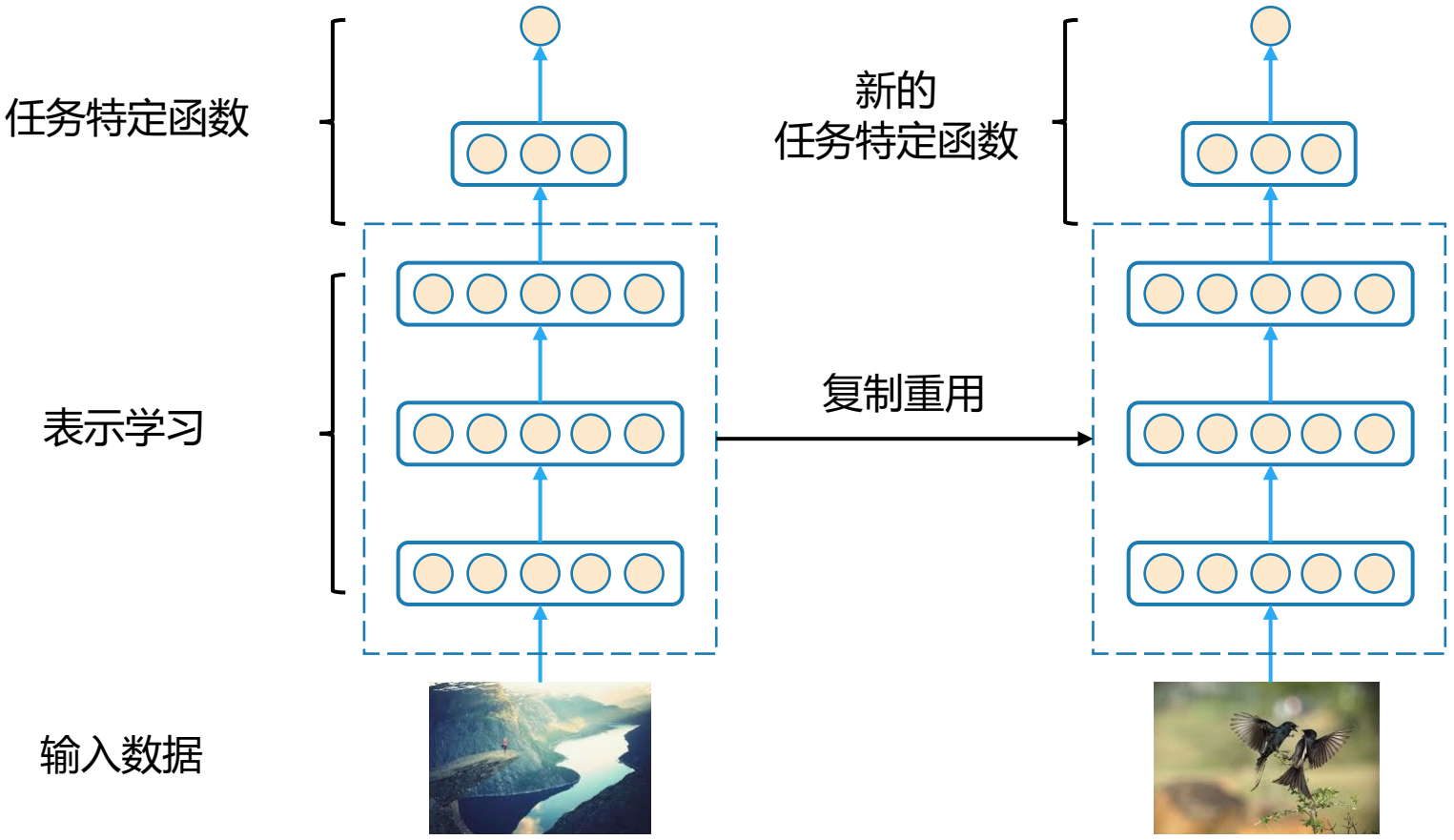
Hariharan and Girshick. Low-shot Visual Recognition by Shrinking and Hallucinating Features. ICCV 2017.

元学习方法

- 基于初始化的方法
  - 学习如何为新任务初始化模型
- 循环神经网络的方法
  - 学习如何通过自回归的方式产生良好的梯度
- 强化学习的方法
  - 学习如何通过强化学习的方式产生良好的梯度

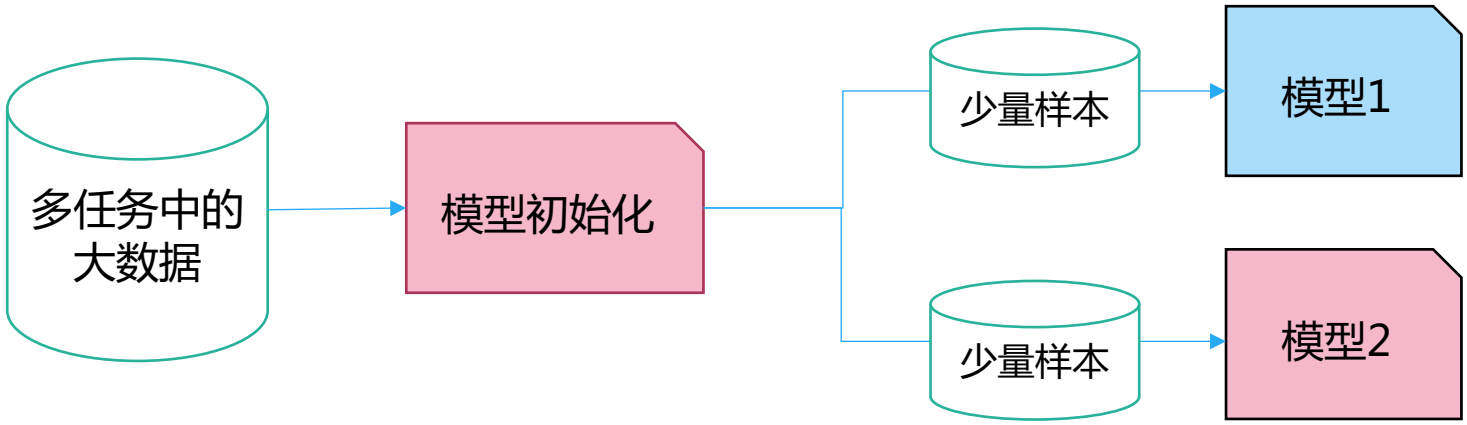
回顾：网络参数重用

- 把底层网络视为表示学习模块，并把他们重用做良好的特征提取器



模型不可知元学习（MAML）

- 目标：训练一个能借助少量样本快速适应不同任务的模型
- MAML的思想：直接优化一个能借助少量样本进行有效微调的初始表示



Finn et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML 2017.

模型不可知元学习（MAML）

- 目标：训练一个能借助少量样本快速适应不同任务的模型

传统机器学习的随机梯度下降（SGD）

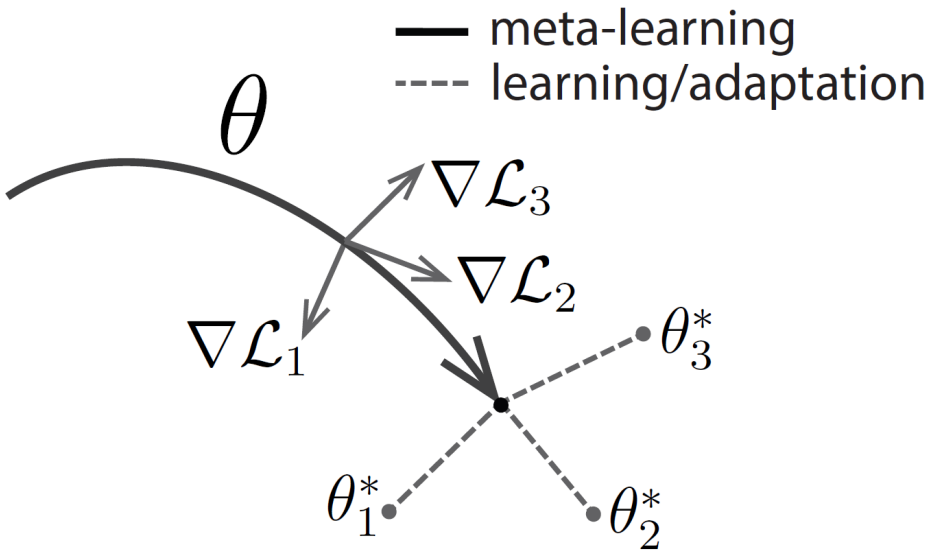
$$\theta \leftarrow \theta - \eta \sum_i \nabla_\theta L(\theta)$$

任务i的理想良好参数

$$\theta^i \leftarrow \theta - \eta \nabla_\theta L_i(\theta)$$

MAML SGD

$$\theta \leftarrow \theta - \eta \sum_i \nabla_\theta L_i\left(\theta - \eta \nabla_\theta L_i(\theta)\right)$$



Finn et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML 2017.

模型不可知元学习（MAML）

- MAML中元梯度（meta-gradient）的更新涉及梯度的梯度

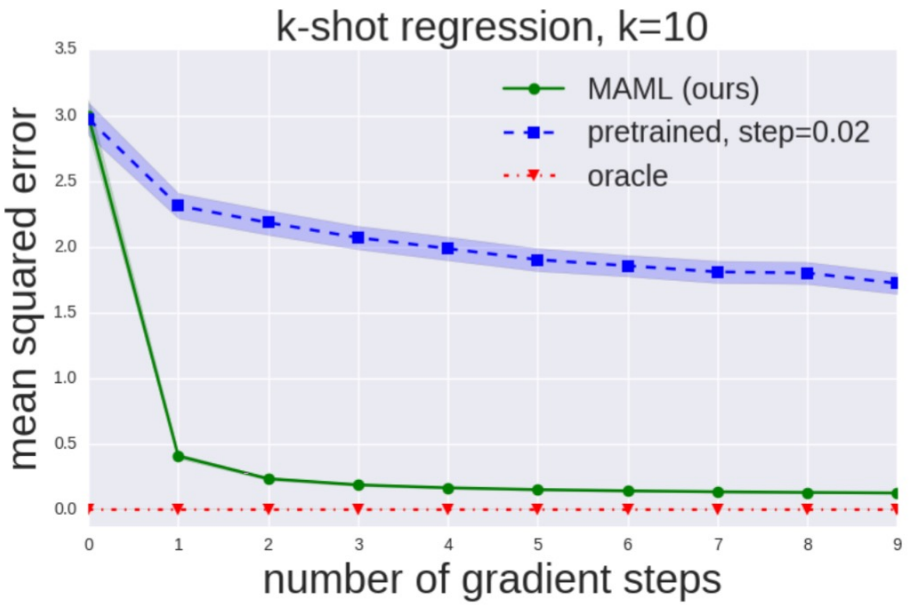
$$\theta \leftarrow \theta - \eta \sum_i \nabla_\theta L_i\left(\theta - \eta \nabla_\theta L_i(\theta)\right)$$

  - 这就需要额外的反向传递过程 f 来计算Hessian-向量
  - 标准深度学习库（如：TensorFlow）支持这些操作

Finn et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML 2017.

利用MAML实现小样本学习

- 预训练：使用预训练网络作为初始化，并进行传统的随机梯度下降（SGD）
- MAML能够取得比RNN元学习（稍后讲解）更好的效果



MiniImagenet (Ravi & Larochelle, 2017)	5-way Accuracy	
	1-shot	5-shot
fine-tuning baseline	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
nearest neighbor baseline	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$
matching nets (Vinyals et al., 2016)	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
meta-learner LSTM (Ravi & Larochelle, 2017)	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML, first order approx. (ours)	$\mathbf{48.07 \pm 1.75\%}$	$\mathbf{63.15 \pm 0.91\%}$
MAML (ours)	$\mathbf{48.70 \pm 1.84\%}$	$\mathbf{63.11 \pm 0.92\%}$

元学习方法

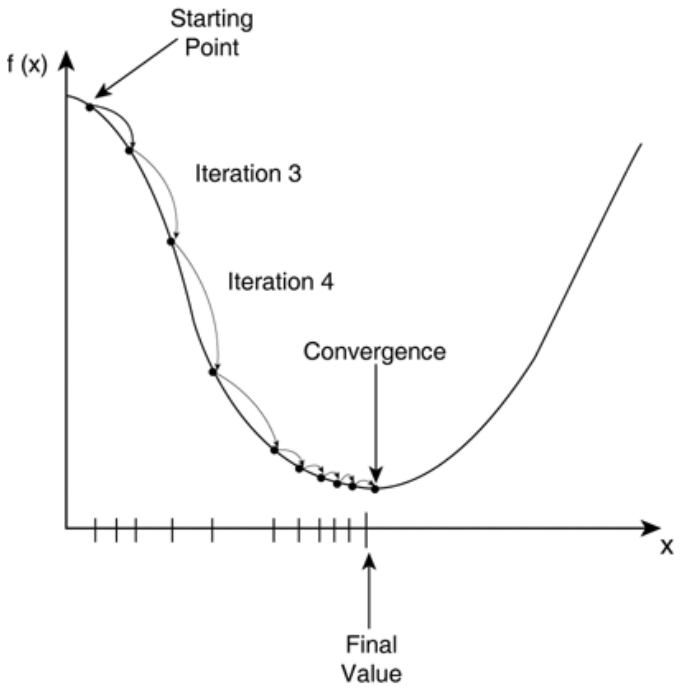
- 基于初始化的方法
  - 学习如何为新任务初始化模型
- 循环神经网络的方法
  - 学习如何通过自回归的方式产生良好的梯度
- 强化学习的方法
  - 学习如何通过强化学习的方式产生良好的梯度

关于梯度学习的重新思考

- 传统机器学习中的梯度

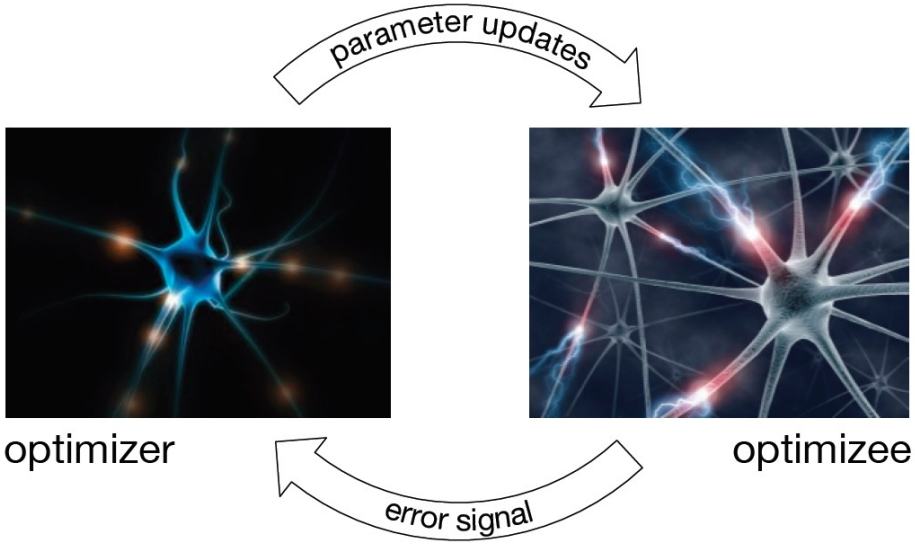
$$\theta_{t+1} = \theta_t - \eta_t \nabla_{\theta_t} L(\theta_t)$$

- 存在的问题
  - 学习率固定不变(or)随启发式规则改变
  - 没有考虑二阶信息（甚至更高阶）
- 可行的思想
  - 记忆历史梯度以更好地确定下一个梯度



Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." *NIPS* 2016.

元优化器决定优化器如何进行优化



- 两个组成部分：元优化器（optimizer）、优化器（optimizee）
- 元优化器接受优化器的表现，并指导优化器进行能使其表现提升的更新

Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." *NIPS* 2016.

使用循环网络的元学习

- 优化器以自回归的方式确定梯度，以循环神经网络（RNN）的形式实现

$$\theta_{t+1} = \theta_t - \eta_t \nabla_{\theta_t} L(\theta_t)$$

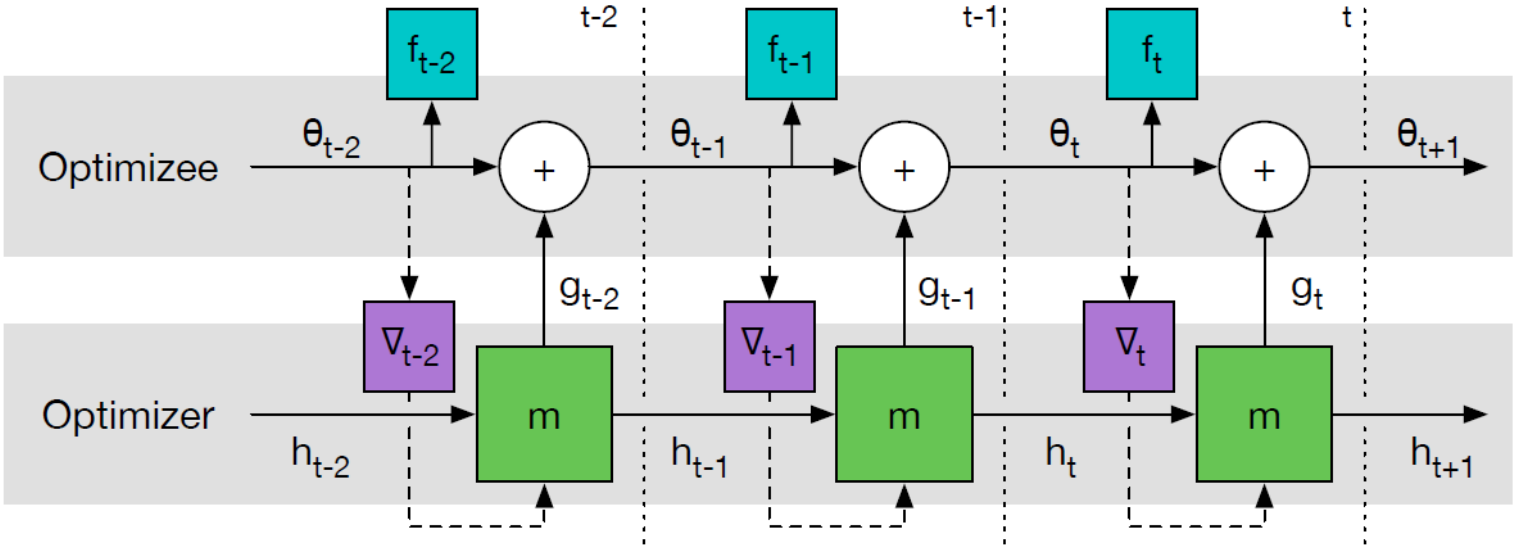
↓

$$\theta_{t+1} = \theta_t - g_t\left(\nabla_{\theta_t} L(\theta_t), \phi\right)$$

g_t 可以通过一个循环神经网络（RNN)实现

Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." *NIPS* 2016.

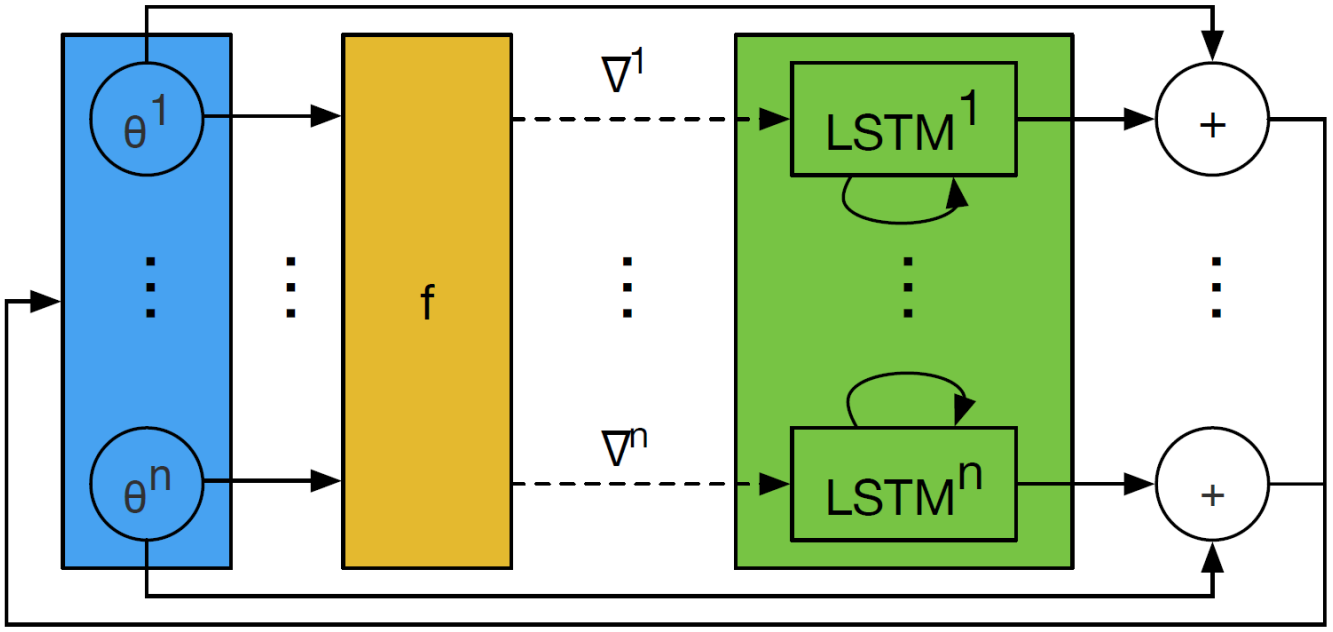
使用循环网络的元学习



- 通过使用RNN，优化器就可以记住隐藏层中的历史梯度信息
- RNN可以通过损失函数直接使用反向传播算法进行更新

Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." *NIPS* 2016.

单变量（Coordinatewise）LSTM优化器



- 通常来说，参数n较大，因此不适合训练全连接的循环神经网络
- 上图展示了协同$LSTM$的结构。即，为每个单独的参数θ^i使用一个$LSTM^i$网络，这些网络共享$LSTM$参数

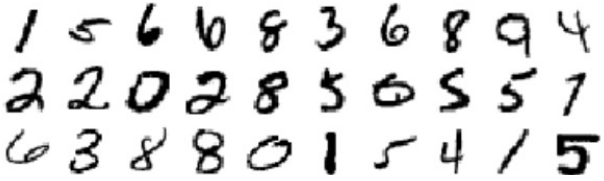
Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." *NIPS* 2016.

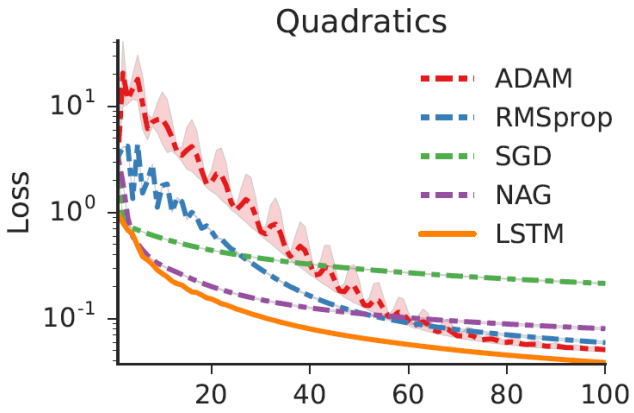
实验结果

- 二次函数

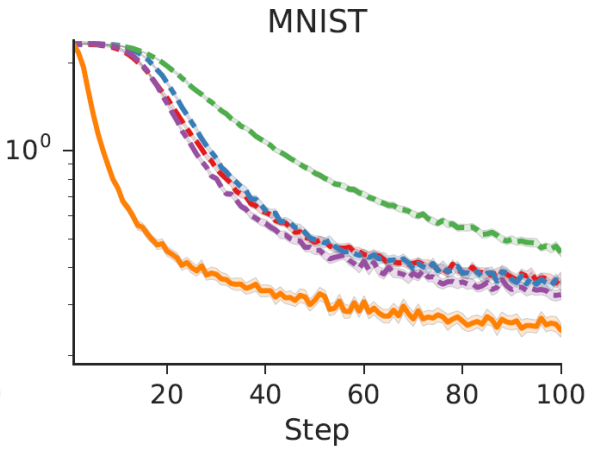
$$f(\theta) = \|W\theta - y\|^2$$

- MNIST

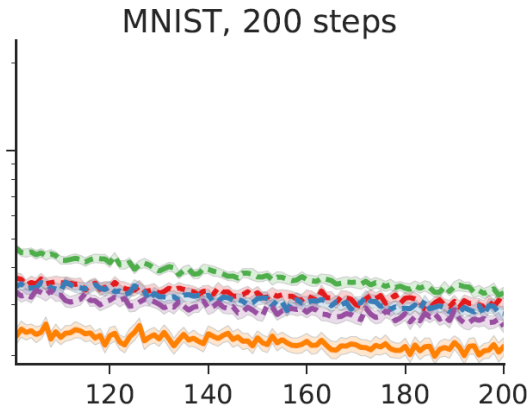




不同优化器在随机抽样的
10-维二次函数上的表现



MNIST上的表现
LSTM比其他算法表现更好



100-200步训练的学习曲线
（延续上图）

Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." *NIPS* 2016.

元学习方法

- 基于初始化的方法
  - 学习如何为新任务初始化模型
- 循环神经网络的方法
  - 学习如何通过自回归的方式产生良好的梯度
- 强化学习的方法
  - 学习如何通过强化学习的方式产生良好的梯度

机器学习优化算法的高层理解

Algorithm 1 General structure of optimization algorithms

Require: Objective function f
  $x^{(0)} \leftarrow$ random point in the domain of f
  for $i = 1, 2, \ldots$ **do**
    $\Delta x \leftarrow \phi(\{x^{(j)}, f(x^{(j)}), \nabla f(x^{(j)})\}_{j=0}^{i-1})$
    if stopping condition is met **then**
      return $x^{(i-1)}$
    end if
    $x^{(i)} \leftarrow x^{(i-1)} + \Delta x$
  end for

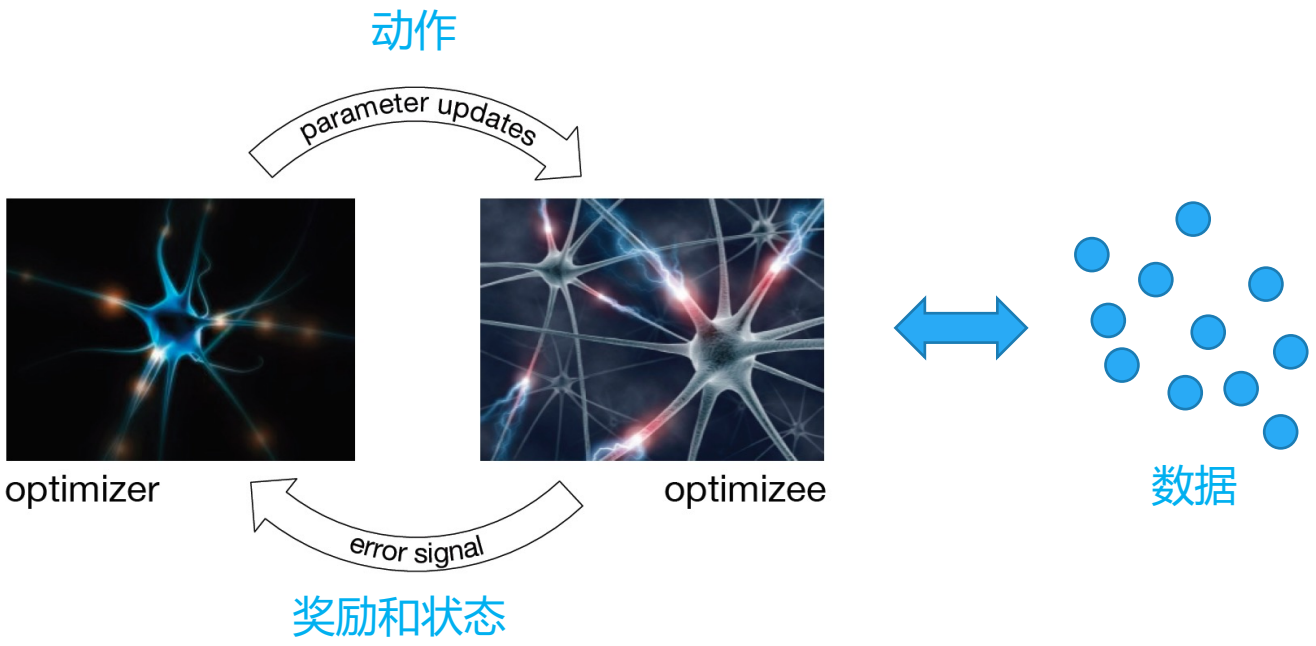
梯度下降	$\phi(\cdot) = -\eta \nabla f(x^{(i-1)})$
动量法	$\phi(\cdot) = -\eta(\sum_{j=0}^{i-1} \eta^{i-1-j} \nabla f(x^{(j)}))$
学习算法	$\phi(\cdot) =$ Neural Net

- 如何设计好： $\Delta x \leftarrow \phi(\{x^{(j)}, f(x^{(j)}), \nabla f(x^{(j)})\}_{j=0}^{i-1})$
- 元学习的关键是设计一个良好的函数，满足：
  - 接受先验观察值和学习行为
  - 输出适当的梯度以让机器学习模型进行更新

Li, Ke, and Jitendra Malik. "Learning to optimize." *arXiv preprint arXiv:1606.01885* (2016).

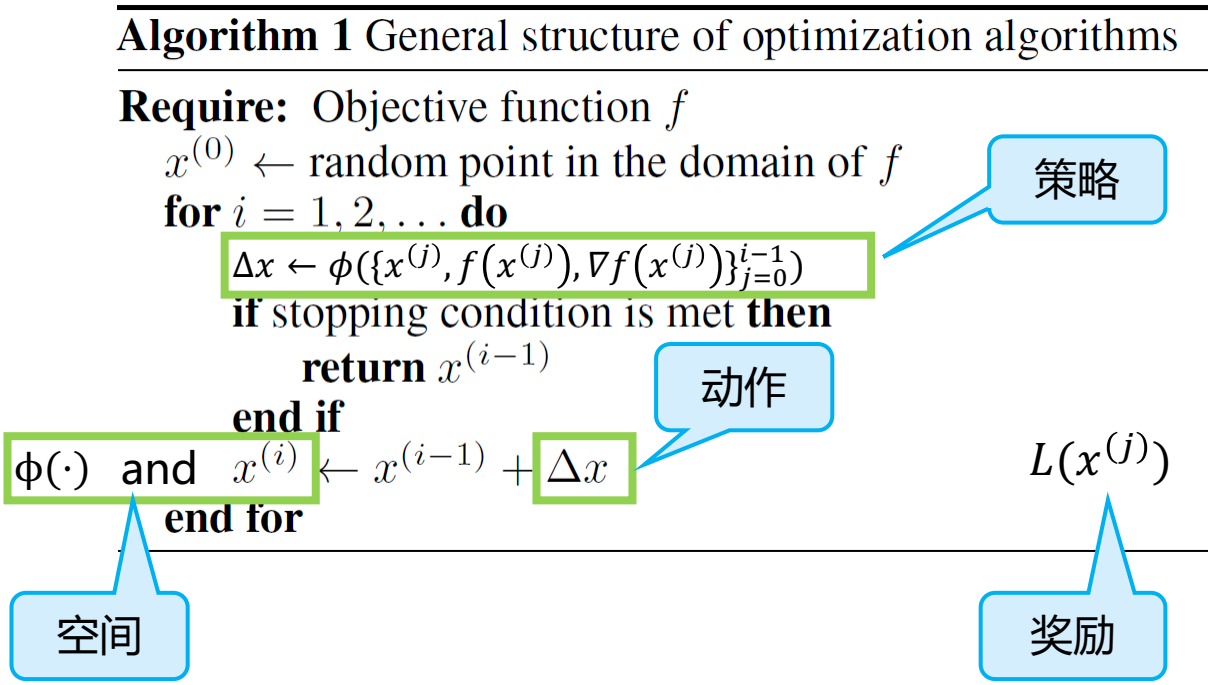
使用强化学习的元学习

- 思想：每个时间步长中，元学习者（meta-learner）学习向学习者（learner）给出优化动作，并观察学习者的表现。这个过程和强化学习的思想非常相似



Li, Ke, and Jitendra Malik. "Learning to optimize." *arXiv preprint arXiv:1606.01885* (2016).

公式化为强化学习问题



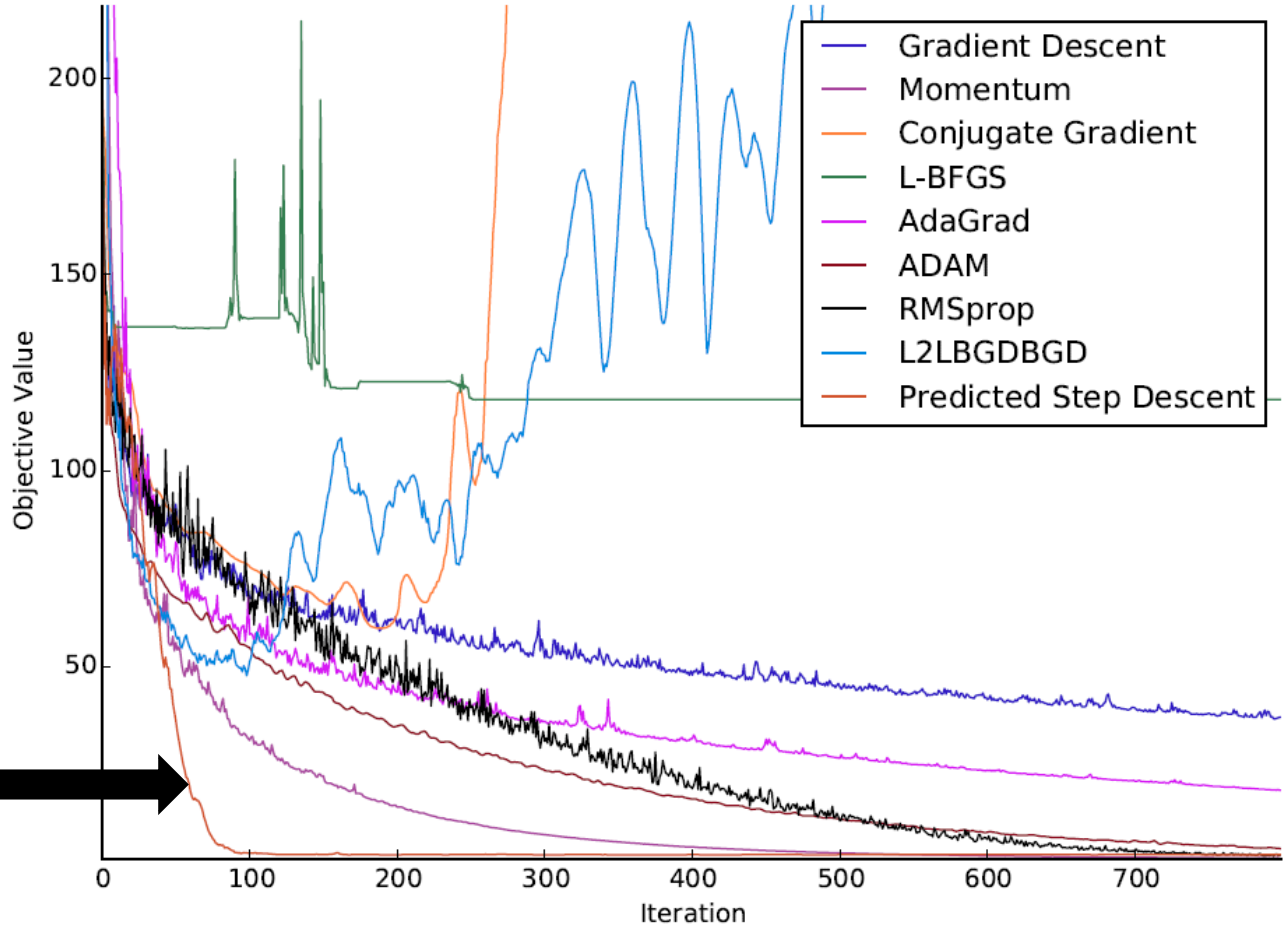
- 状态表示由一个函数Φ(·)生成，该函数将观察到的数据和学习行为映射到一个潜在的表示（latent representation）中
- 动作是策略输出的梯度
- 奖励是与当前模型参数相关的损失函数

Li, Ke, and Jitendra Malik. "Learning to optimize." *arXiv preprint arXiv:1606.01885* (2016).

实验结果

浅红色曲线是一个使用强化学习训练的优化器

与使用监督学习训练的优化器不同，它不会在后续的迭代中发散



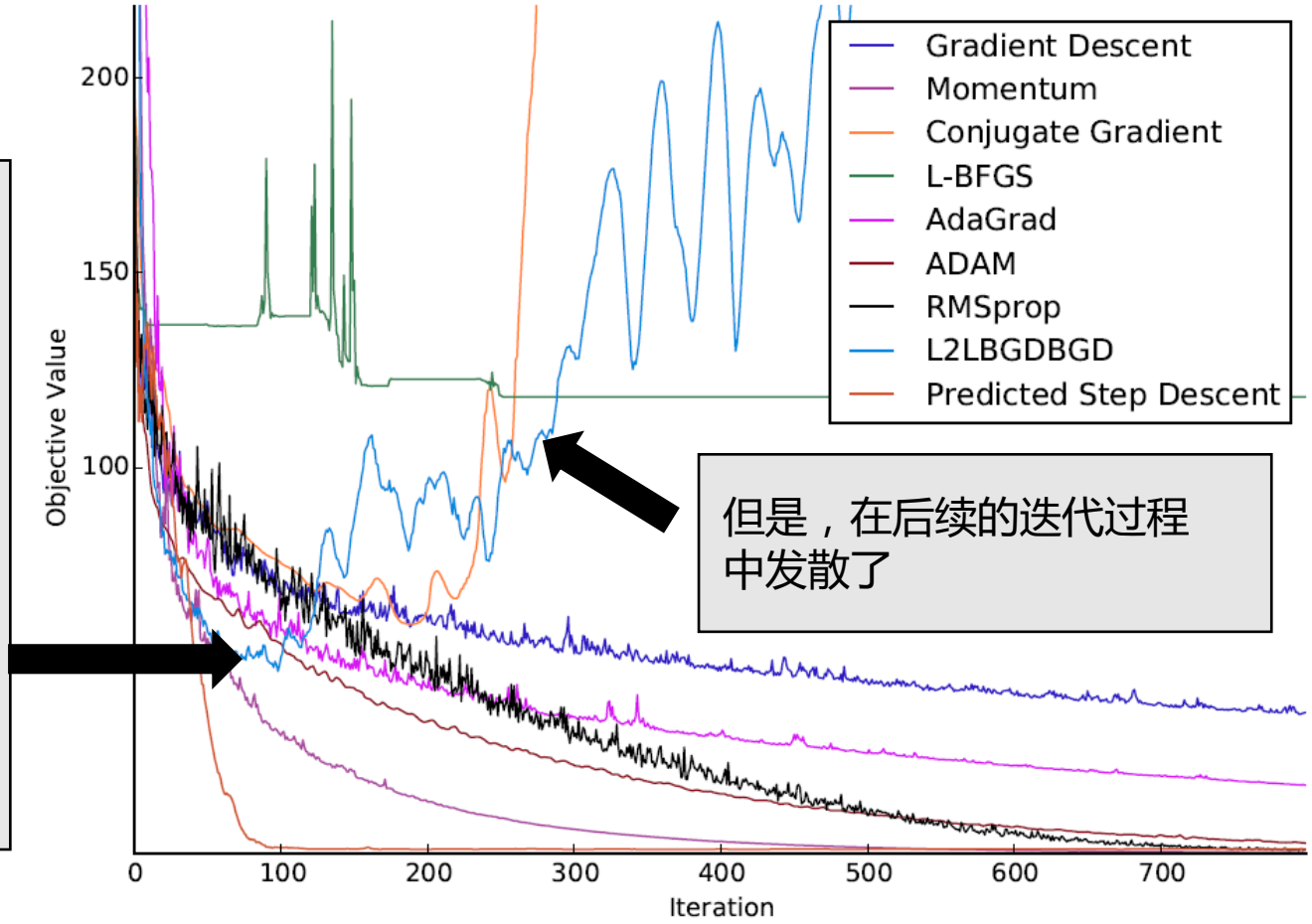
数据：随机投影和标准化版本的MNIST数据集，维度为48维，每一维上具有单位方差

https://bair.berkeley.edu/blog/2017/09/12/learning-to-optimize-with-rl/

实验结果

浅蓝色曲线是一个使用监督学习训练的优化器

在这里，它被用来训练一个新任务的神经网络。其最初阶段的表现相当好



数据：随机投影和标准化版本的MNIST数据集，维度为48维，每一维上具有单位方差

https://bair.berkeley.edu/blog/2017/09/12/learning-to-optimize-with-rl/

元学习参考文献

- 优秀博客
  - https://medium.com/huggingface/from-zero-to-research-an-introduction-to-meta-learning-8e16e677f78a
  - https://bair.berkeley.edu/blog/2017/09/12/learning-to-optimize-with-rl/
  - https://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/
- Sergey Levin的强化学习课程
  - http://rail.eecs.berkeley.edu/deeprlcourse-fa17/f17docs/lecture_16_meta_learning.pdf
- 两篇值得读读的论文
  - *Learning to learn by gradient descent by gradient descent*
  - *Learning to Learn without Gradient Descent by Gradient Descent*

自动机器学习与神经网络架构搜索

张伟楠 – 上海交通大学

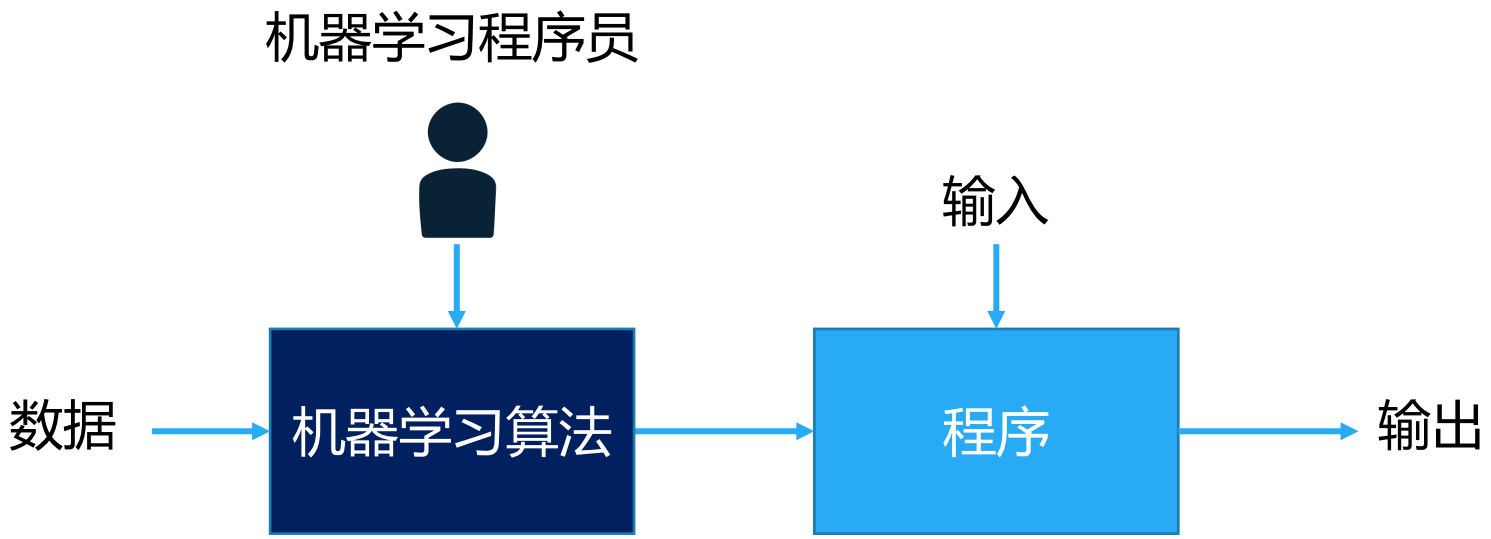
神经网络架构搜索和自动机器学习的推广

- 一篇关于神经网络架构搜索（neural architecture search, NAS）的有影响力的论文
  - Zoph, Barret, and Quoc Le. *Neural Architecture Search with Reinforcement Learning.* ICLR 2017.

- 人工智能民主化革命
  - 李飞飞，时任谷歌云自动机器学习（Google Cloud AutoML）负责人
  - Published Jan 17, 2018
  - https://www.blog.google/products/google-cloud/cloud-automl-making-ai-accessible-every-business/

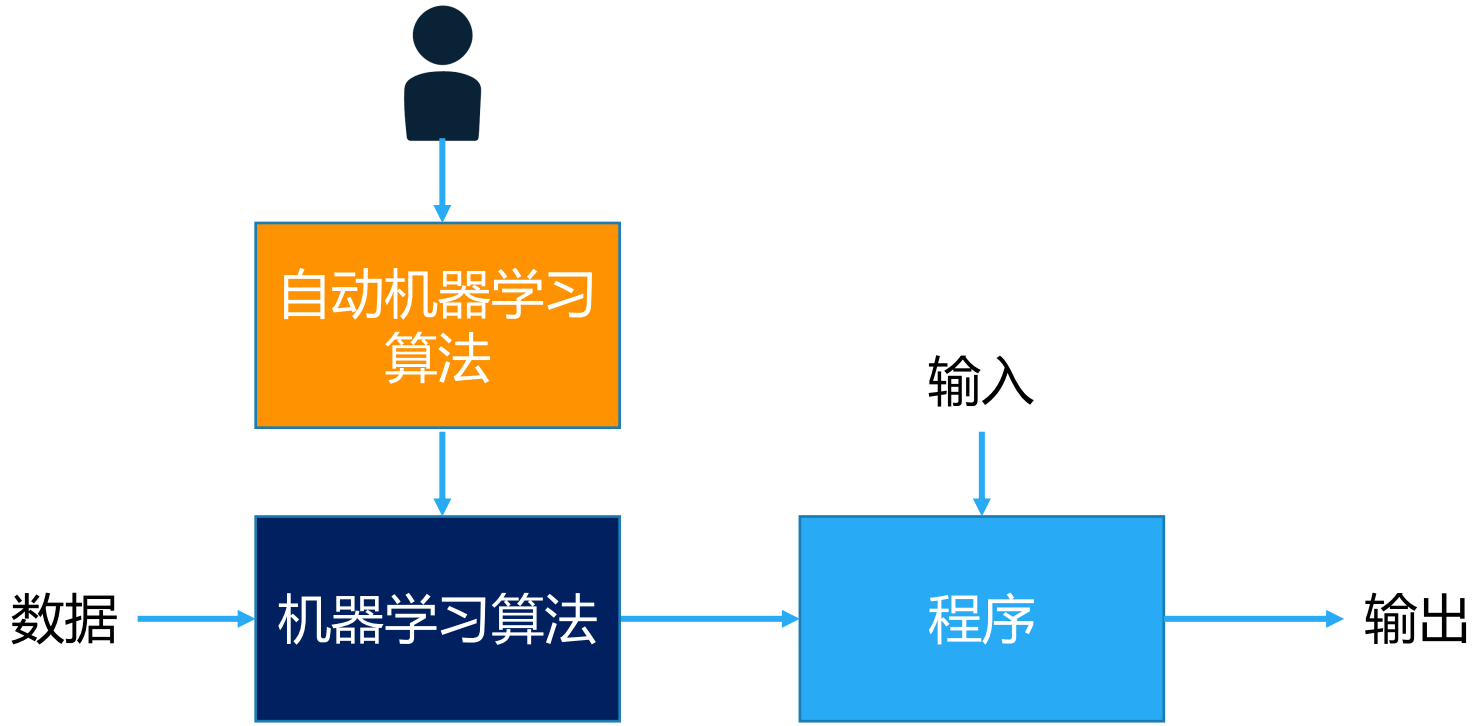
从机器学习到自动机器学习

- 机器学习
  - 机器学习程序员仍需要凭经验进行模型选择、超参数调整等工作

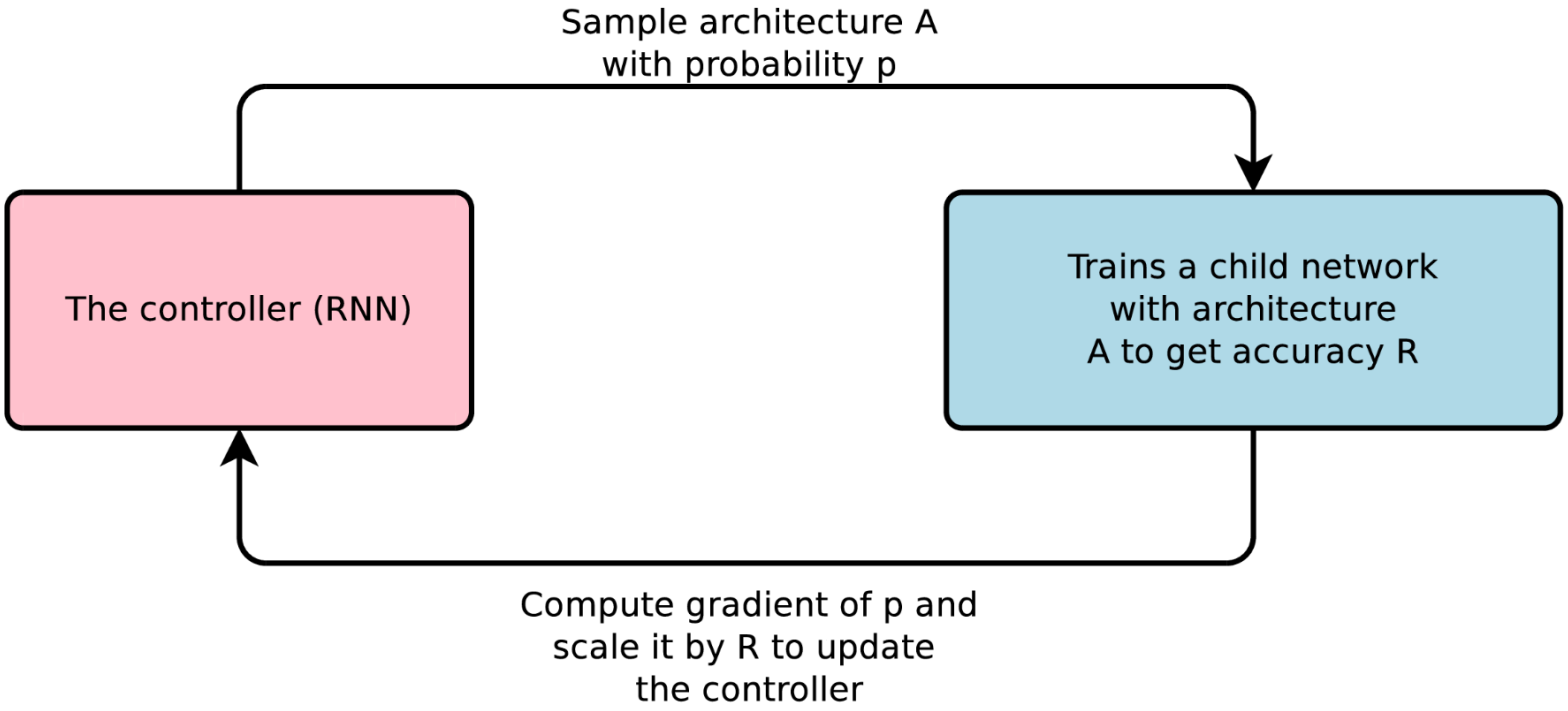


从机器学习到自动机器学习

- 自动机器学习
  - 机器学习程序员几乎不需要做任何多余的工作



神经网络架构搜索（NAS）框架



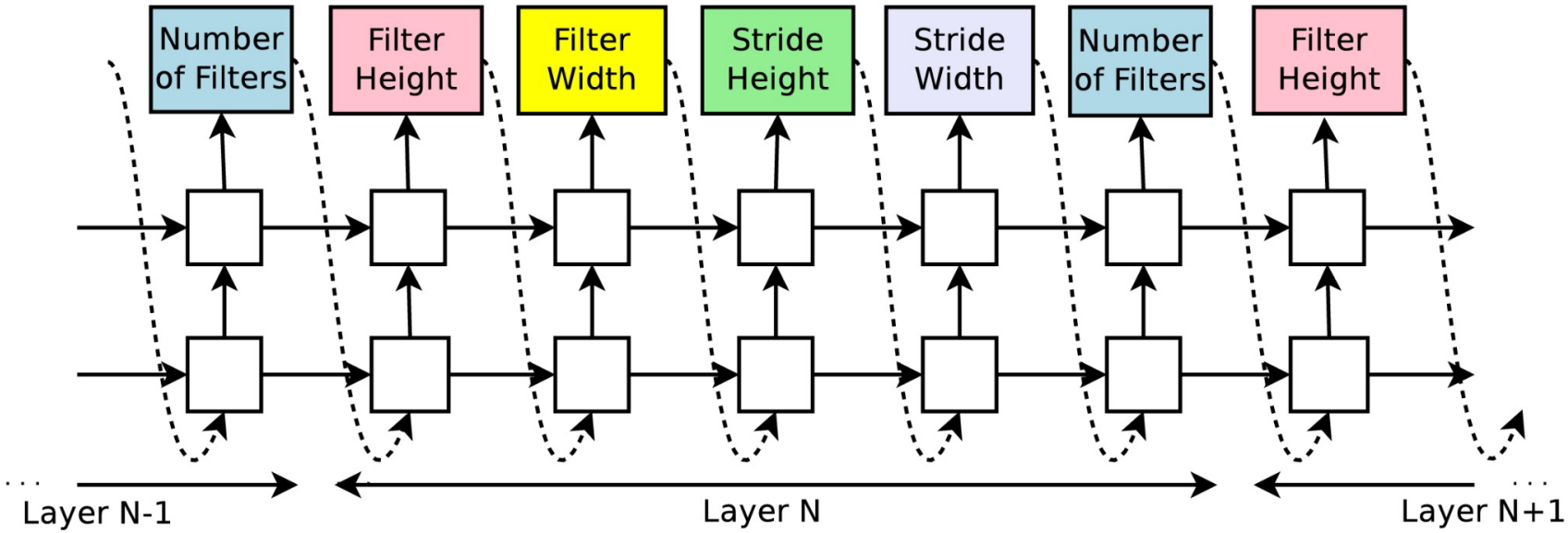
- 控制器（controller）决定接下来尝试哪个子网络，并观察相应的表现作为反馈
- 通过多次尝试-反馈，控制器学习如何选择有效的子网络

神经网络架构搜索（NAS）框架

- **观测值（Observation）**：一个神经网络的结构和连通性能够由可变长度的字符串描述
- **控制器（Controller）**：一个能够生成上述字符串的循环神经网络（RNN）
- **子网络（Child network）**：训练由上述字符串指定的神经网络
- 验证集上子网络的结果作为更新控制器的奖励

卷积架构搜索

- 使用循环神经网络（RNN）控制器进行卷积神经网络（CNN）架构搜索



对于每一层，控制器能够决定：过滤器（或称卷积核filter/kernel）的高度、宽度，间隔（stride）的高度、宽度，以及过滤器的个数

Zoph, Barret, and Quoc Le. Neural Architecture Search with Reinforcement Learning. ICLR 2017.

卷积架构搜索

1. 当层数超过某个值时此过程停止（该值随训练过程的进行而增加）
2. 训练生成的网络直到其收敛
3. 记录验证集上的准确率
4. 更新控制器循环神经网络（RNN）的参数θ

控制器策略的目标

- 控制器生成的一系列词符（token）是一系列动作$a_{1:T}$
- 准确率$R(a_{1:T})$作为奖励信号
- 控制器的目标是最大化期望奖励：

$$J(\theta_c) = \mathbb{E}_{\pi(a_{1:T};\theta_c)}[R(a_{1:T})]$$

使用REINFORCE进行训练

- 因为R是不可微的，我们可以使用REINFORCE规则

$$\nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^{T} \mathbb{E}_{\pi(a_{1:T};\theta_c)}\left[\nabla_{\theta_c} \log \pi(a_t | a_{1:t-1}; \theta_c) R(a_{1:T})\right]$$

- 上式的经验近似为

$$\nabla_{\theta_c} J(\theta_c) \simeq \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} \nabla_{\theta_c} \log \pi\left(a_t^{(k)} \middle| a_{1:t-1}^{(k)}; \theta_c\right) R\left(a_{1:T}^{(k)}\right)$$

其中m为使用批处理时一个批次中对架构的采样数

使用REINFORCE进行训练

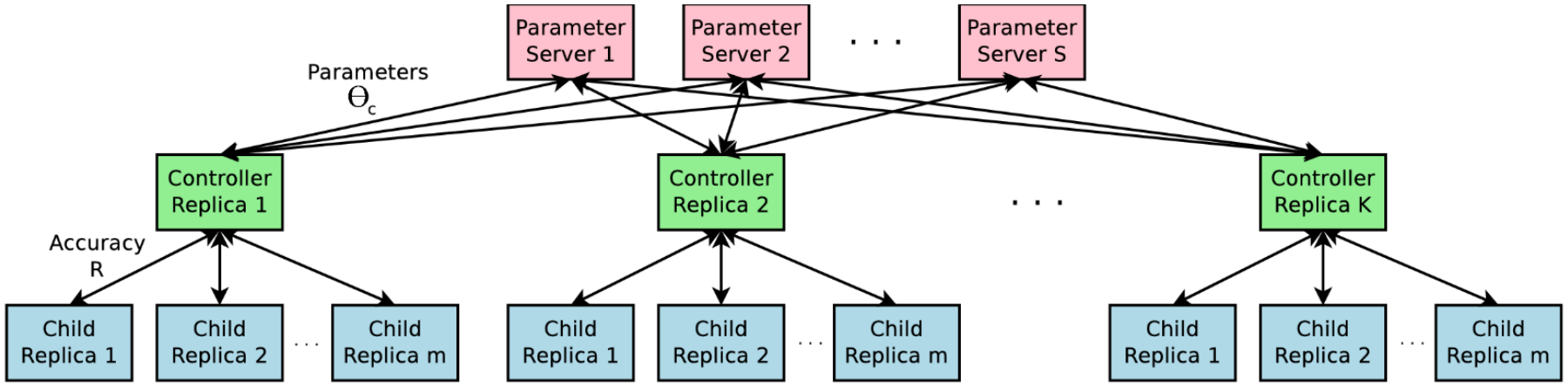
- 该估计是无偏的，但方差较大，解决办法是使用基线函数（baseline function）：

$$\nabla_{\theta_c} J(\theta_c) \simeq \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} \nabla_{\theta_c} \log \pi \left(a_t^{(k)} \middle| a_{1:t-1}^{(k)}; \theta_c \right) \left(R\left(a_{1:T}^{(k)} \right) - b \right)$$

  - 在实际中，b作为基线函数，可以是先前实验中准确率的移动窗口的平均值

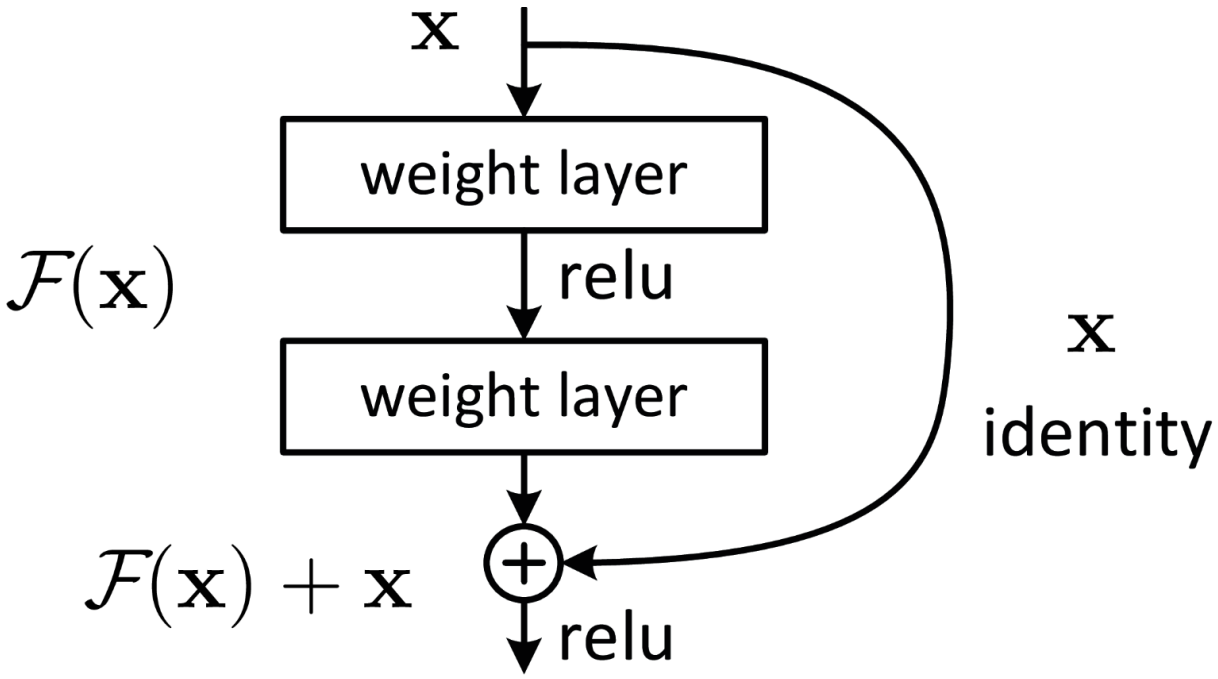
并行训练和异步更新

- 训练子网络可能花费很长的时间
- 使用分布式训练和异步参数更新能够加速这一过程



参数集合 S 储存并将参数传输给 K 个控制器副本
每个控制器副本采样 m 个架构且并行地运行多个子模型

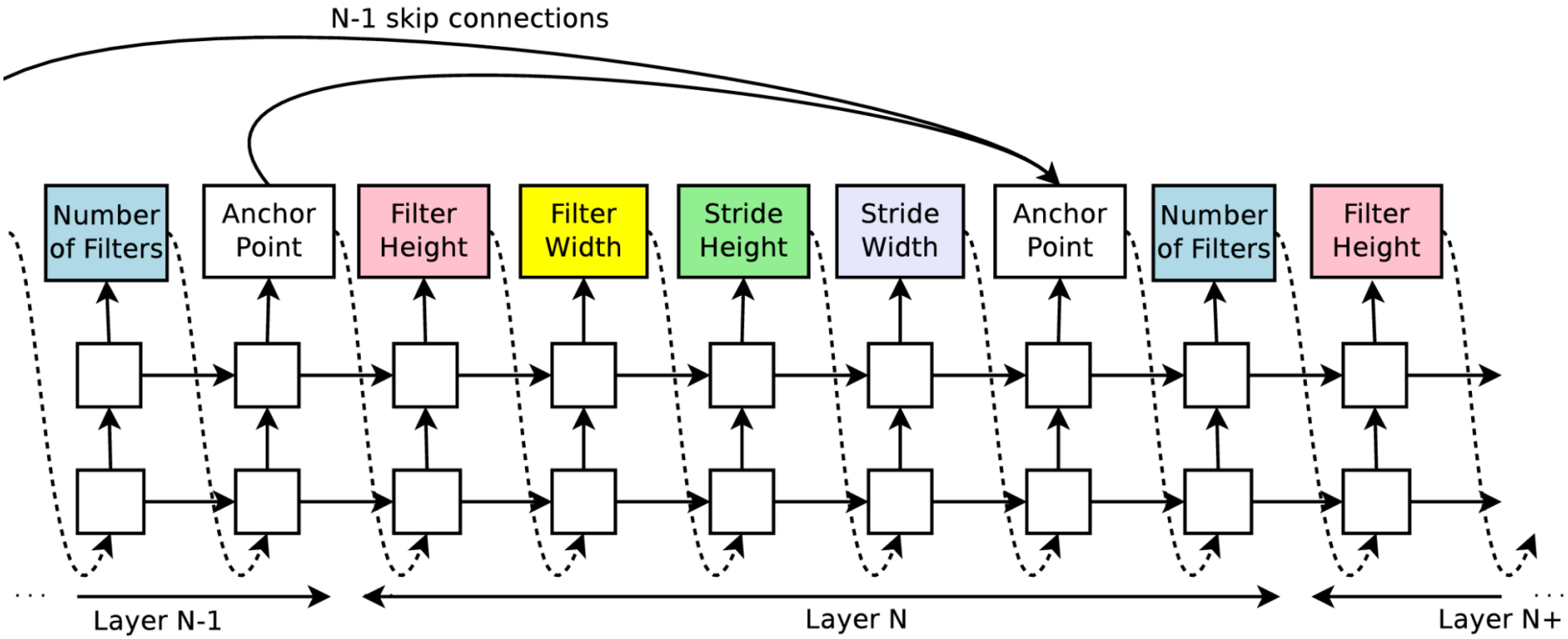
跳跃式连接（Skip Connections）



残差神经网络架构

He, Kaiming, et al. "Deep residual learning for image recognition." *CVPR*. 2016.

跳跃式连接（Skip Connections）



带锚点的控制器决定跳跃式连接

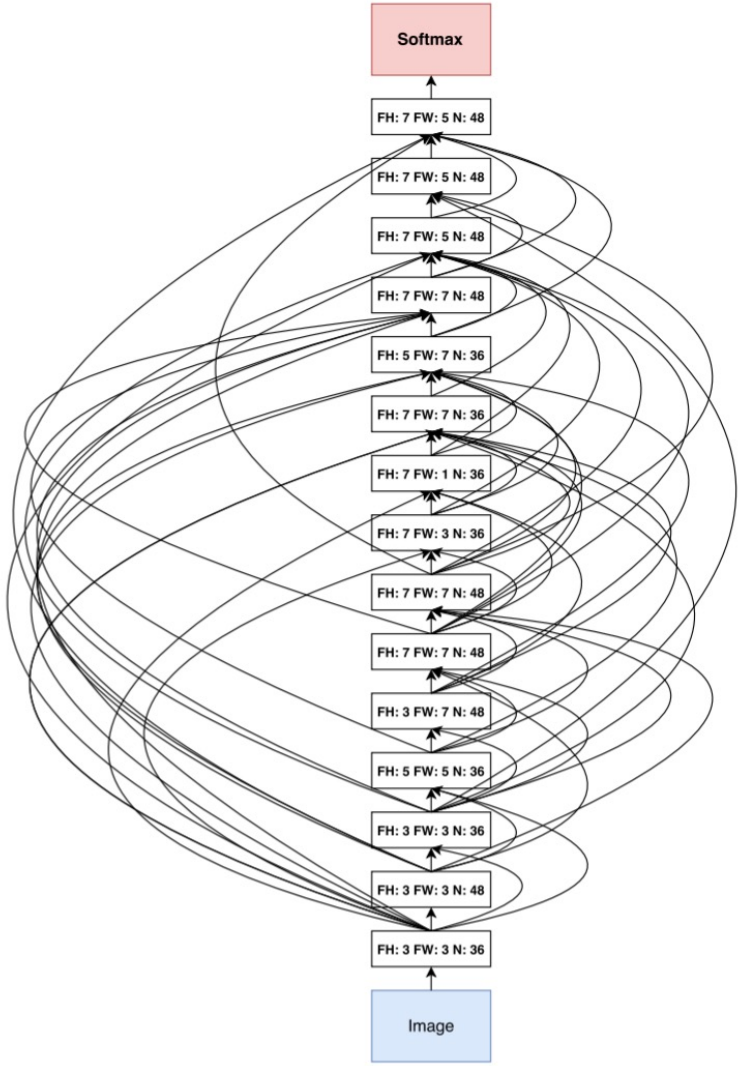
$$\pi(\text{layer } j \text{ is an input of layer } i) = \text{sigmoid}\left(v^{\mathrm{T}}\tanh\left(W_{\text{prev}}h_j + W_{\text{curr}}h_i\right)\right)$$

不仅限于卷积层

- 其他类型的层也可以通过在控制器中添加RNN单元来预测使用哪种类型的层
  - 池化（Pooling）
  - 批标准化（Batch norm）
  - …

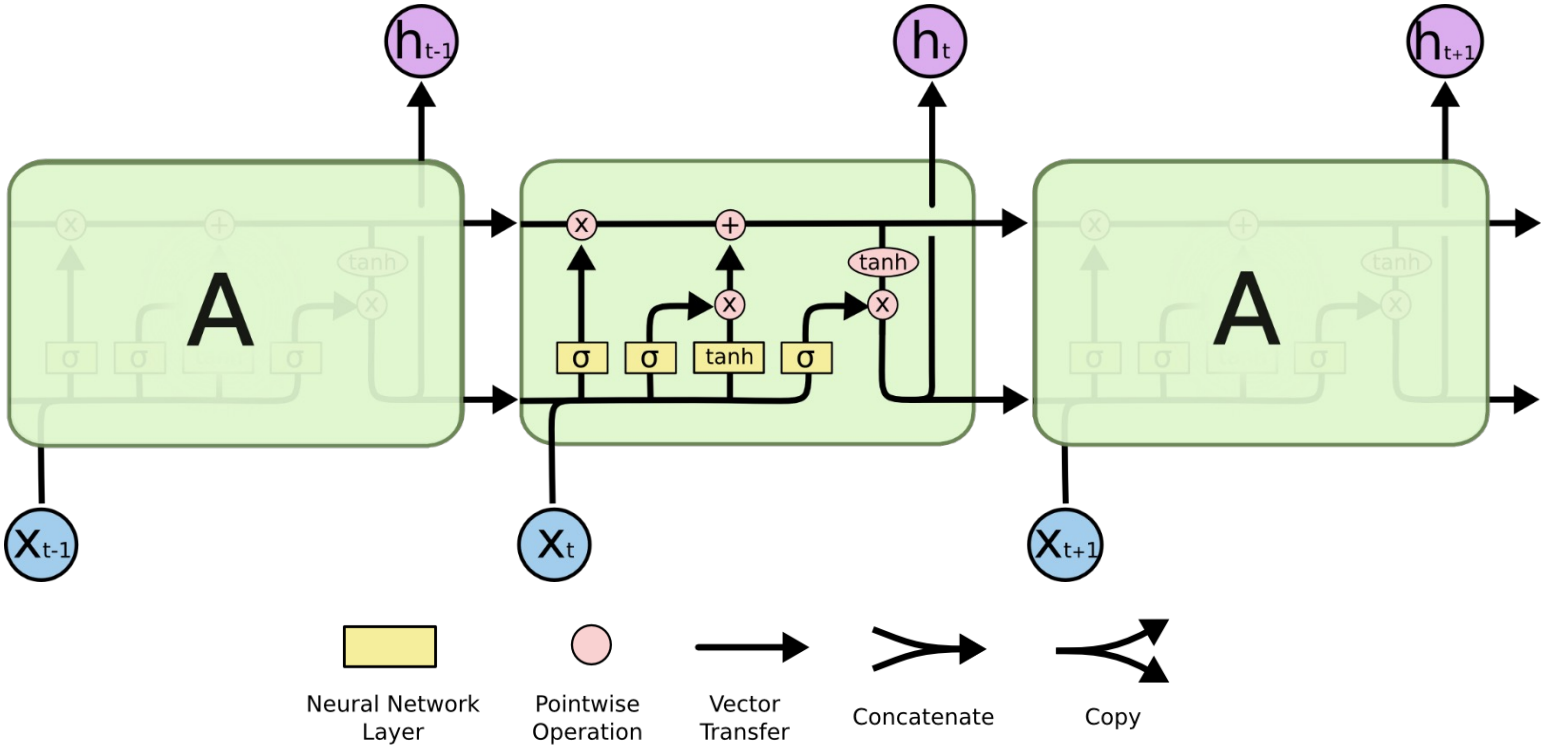
发现CNN架构

- FH为过滤器高度
- FW为过滤器宽度
- N为过滤器数量
- 无间隔（stride）或池化层(pooling layer)
- 跳跃式连接为非残差连接

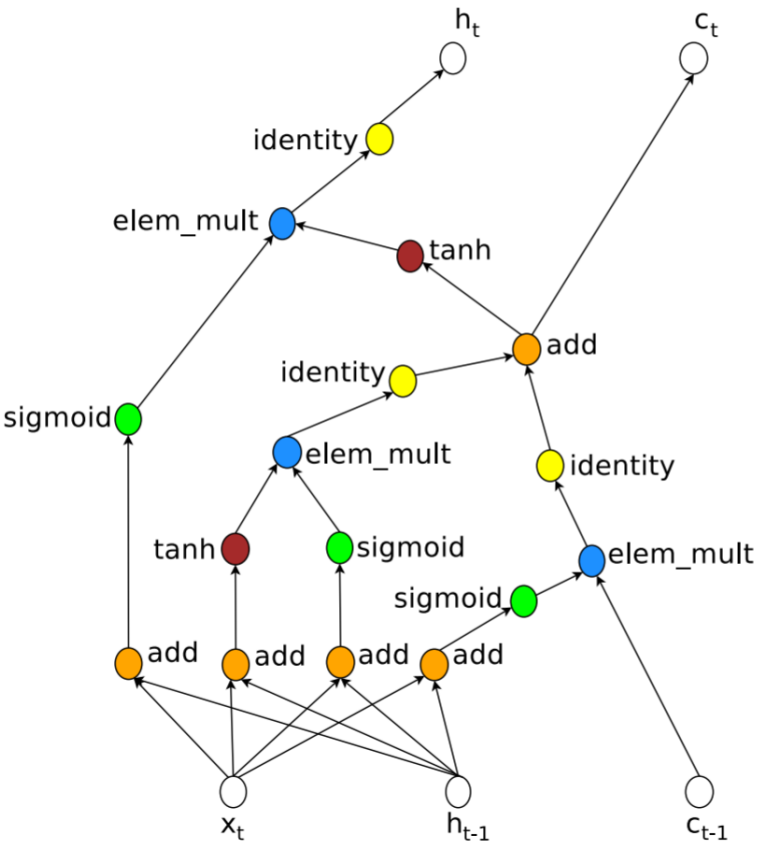


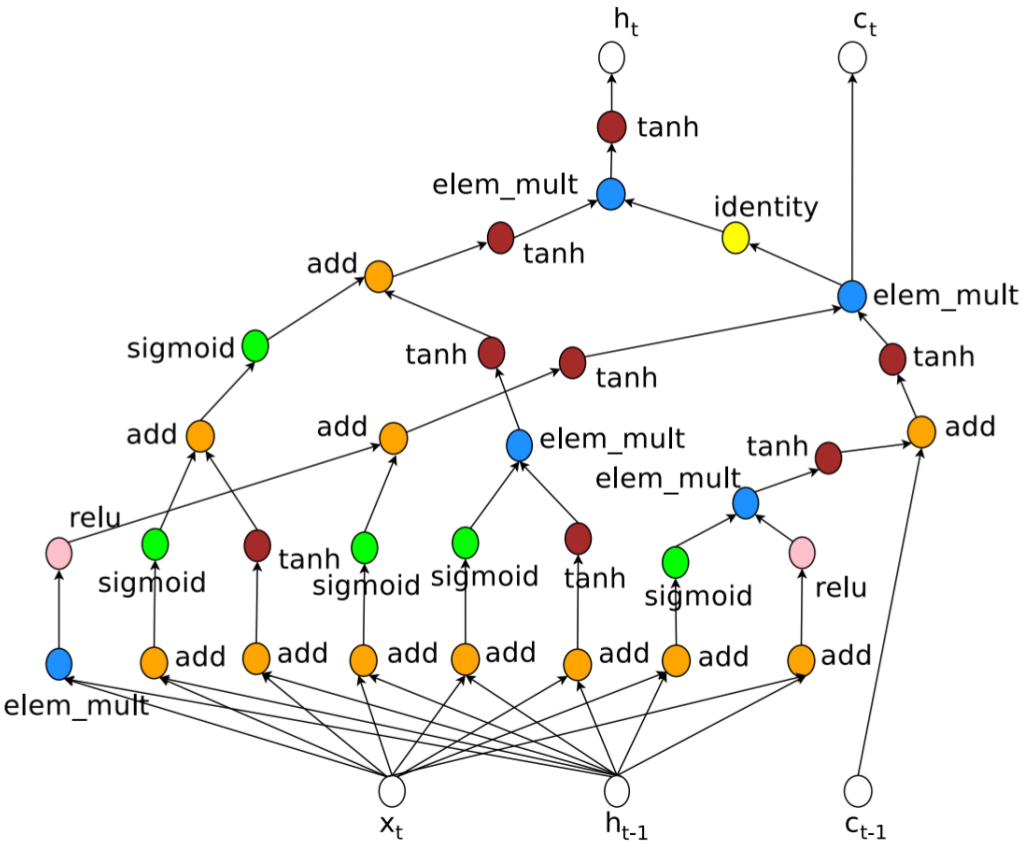
RNN单元往往用固定的几种设计

- 标准recurrent cell：$h_t = f(h_{t-1}, x_t)$
- LSTM cell：$(h_t, c_t) = f(h_{t-1}, c_{t-1}, x_t)$

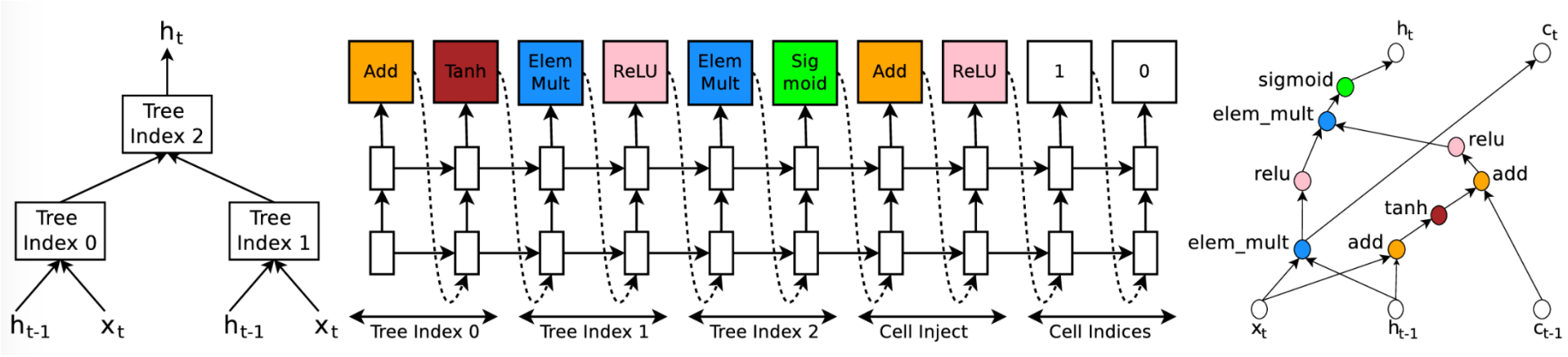


树状LSTM Cell


h_t
c_t
identity
elem_mult
tanh
add
identity
sigmoid
elem_mult
identity
tanh
sigmoid
elem_mult
sigmoid
add
add
add
add
x_t
h_{t-1}
c_{t-1}


h_t
c_t
tanh
elem_mult
identity
add
tanh
elem_mult
sigmoid
tanh
tanh
tanh
add
add
elem_mult
tanh
add
elem_mult
relu
sigmoid
tanh
sigmoid
sigmoid
tanh
sigmoid
relu
elem_mult
add
add
add
add
add
add
add
add
x_t
h_{t-1}
c_{t-1}

Recurrent Cell搜索



CIFAR-10上的表现

Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016c))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016c)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

Penn Treebank上的表现

Model	Parameters	Test Perplexity
Mikolov & Zweig (2012) - KN-5	2M‡	141.2
Mikolov & Zweig (2012) - KN5 + cache	2M‡	125.7
Mikolov & Zweig (2012) - RNN	6M‡	124.7
Mikolov & Zweig (2012) - RNN-LDA	7M‡	113.7
Mikolov & Zweig (2012) - RNN-LDA + KN-5 + cache	9M‡	92.0
Pascanu et al. (2013) - Deep RNN	6M	107.5
Cheng et al. (2014) - Sum-Prod Net	5M‡	100.0
Zaremba et al. (2014) - LSTM (medium)	20M	82.7
Zaremba et al. (2014) - LSTM (large)	66M	78.4
Gal (2015) - Variational LSTM (medium, untied)	20M	79.7
Gal (2015) - Variational LSTM (medium, untied, MC)	20M	78.6
Gal (2015) - Variational LSTM (large, untied)	66M	75.2
Gal (2015) - Variational LSTM (large, untied, MC)	66M	73.4
Kim et al. (2015) - CharCNN	19M	78.9
Press & Wolf (2016) - Variational LSTM, shared embeddings	51M	73.2
Merity et al. (2016) - Zoneout + Variational LSTM (medium)	20M	80.6
Merity et al. (2016) - Pointer Sentinel-LSTM (medium)	21M	70.9
Inan et al. (2016) - VD-LSTM + REAL (large)	51M	68.5
Zilly et al. (2016) - Variational RHN, shared embeddings	24M	66.0
Neural Architecture Search with base 8	32M	67.9
Neural Architecture Search with base 8 and shared embeddings	25M	64.0
Neural Architecture Search with base 8 and shared embeddings	54M	62.4

总结多任务学习、元学习、自动机器学习

- 和迁移学习仅仅优化目标任务的效能不同，多任务学习旨在优化模型在多个任务上面的效能

- 元学习旨在研究机器如何能找到任务之间的内在联系，做到任务间的泛化。在元学习中，每一个任务的训练和测试仅仅是元学习的一个数据实例。

- 自动机器学习旨在元层面优化学习任务的各个非学习的参数，目前自动机器学习的one-shot方法就是一种元学习思维。

- 相比于元学习的科学系高度，自动机器学习更加“亲民”，作为一种自动化机器学习的服务存在于各个云计算平台上，已经为诸多企业的机器学习业务带来便利

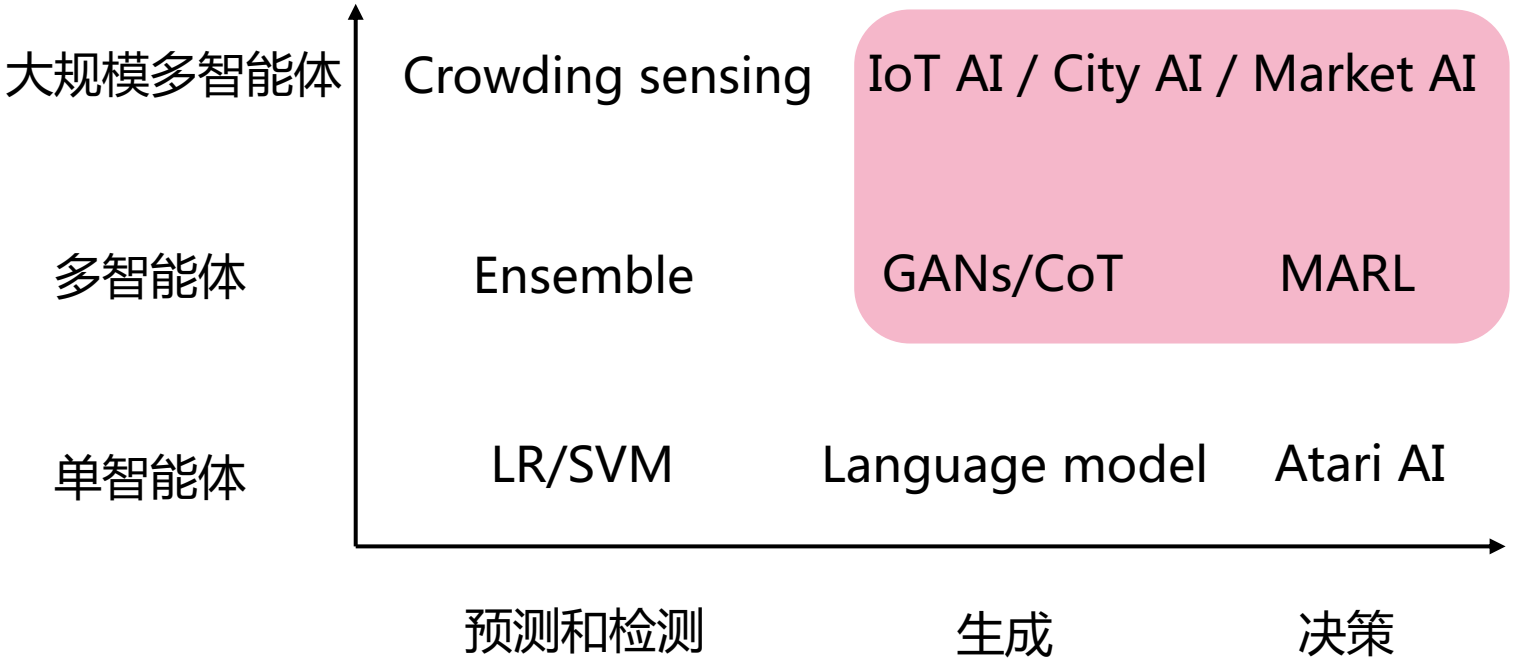
机器学习的未来

张伟楠 – 上海交通大学

机器学习范式的扩展

面向更分散的服务

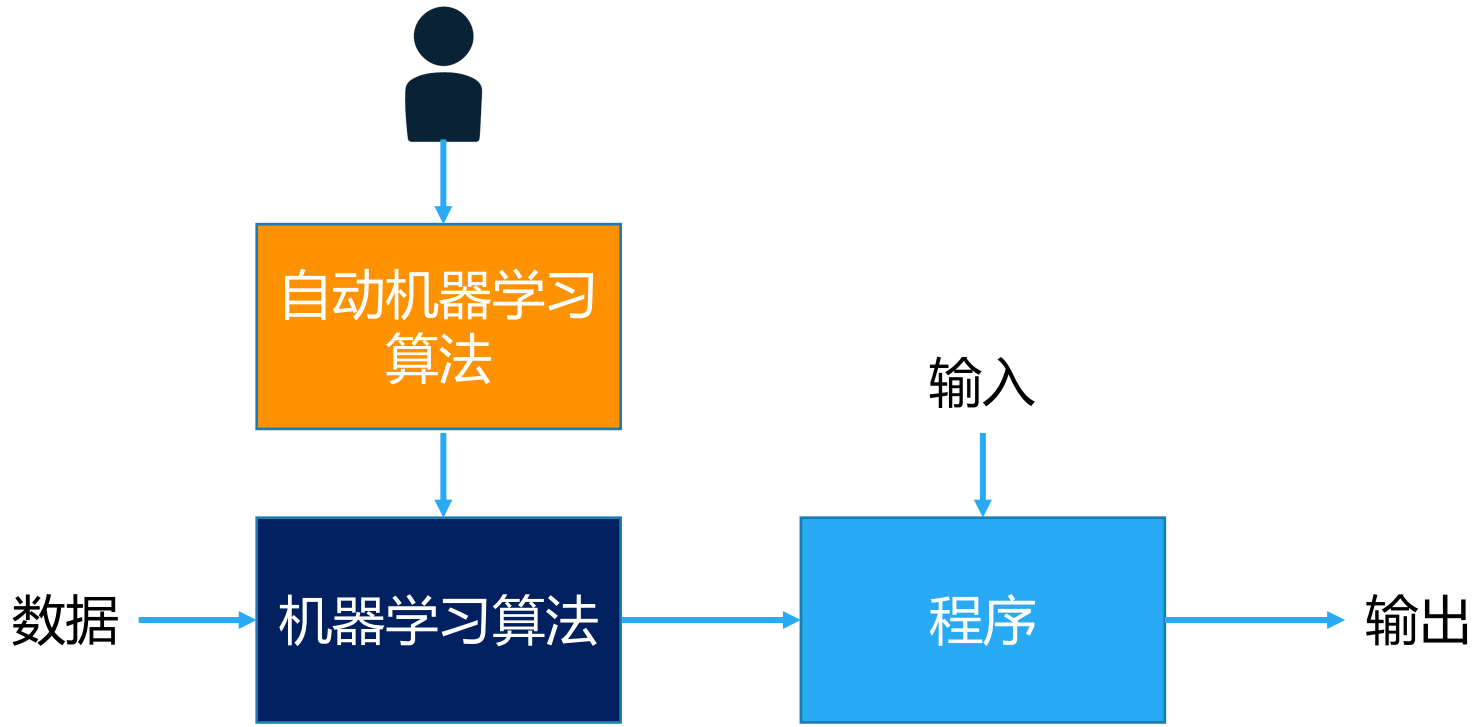
该领域受到越来越多的关注！



提供更多机器访问权限

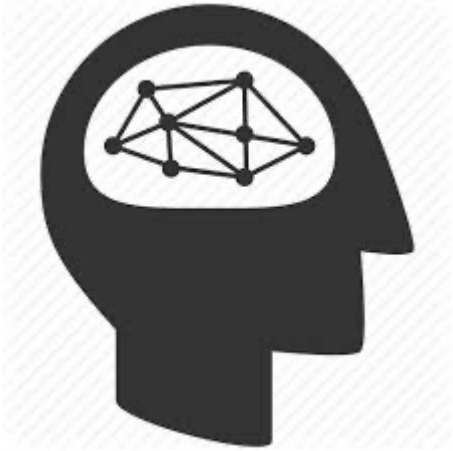
机器学习范式的扩展

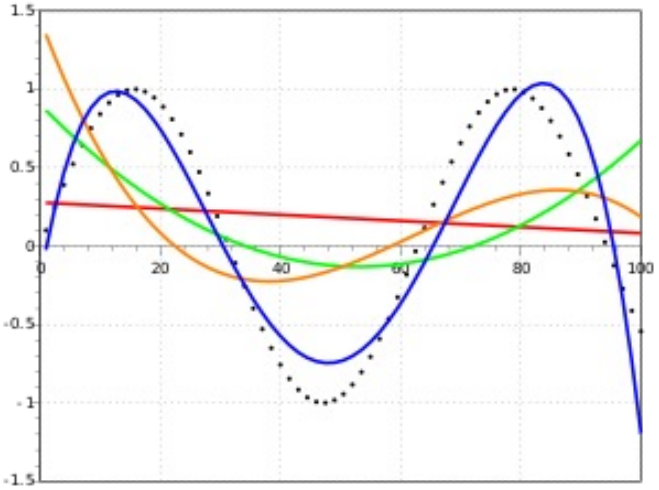
- 自动机器学习使AI更广泛地被使用



符号人工智能 vs. 统计人工智能

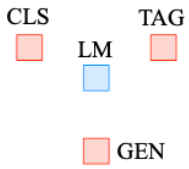
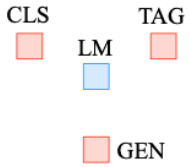
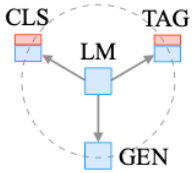
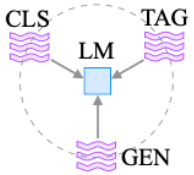
- 组合泛化
  - 常识
  - 知识
  - 逻辑
  - …
- 统计泛化
  - 拟合函数曲线
  - 目标函数优化
  - 正则化
  - 先验估计
  - …





机器学习范式的扩展

- 大模型开始展示强大的智能

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

Pengfei Liu et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

通用人工智能的到来

- 微软对GPT-4的分析报告

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research

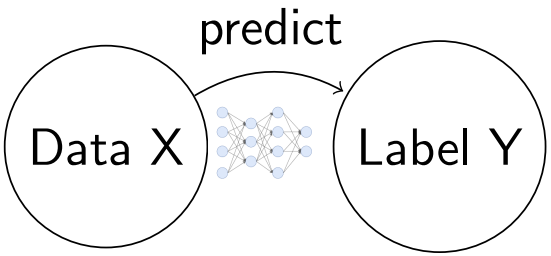
Abstract

Artificial intelligence (AI) researchers have been developing and refining large language models (LLMs) that exhibit remarkable capabilities across a variety of domains and tasks, challenging our understanding of learning and cognition. The latest model developed by OpenAI, GPT-4 [Ope23], was trained using an unprecedented scale of compute and data. In this paper, we report on our investigation of an early version of GPT-4, when it was still in active development by OpenAI. We contend that (this early version of) GPT-4 is part of a new cohort of LLMs (along with ChatGPT and Google's PaLM for example) that exhibit more general intelligence than previous AI models. We discuss the rising capabilities and implications of these models. We demonstrate that, beyond its mastery of language, GPT-4 can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more, without needing any special prompting. Moreover, in all of these tasks, GPT-4's performance is strikingly close to human-level performance, and often vastly surpasses prior models such as ChatGPT. Given the breadth and depth of GPT-4's capabilities, we believe that it could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence (AGI) system. In our exploration of GPT-4, we put special emphasis on discovering its limitations, and we discuss the challenges ahead for advancing towards deeper and more comprehensive versions of AGI, including the possible need for pursuing a new paradigm that moves beyond next-word prediction. We conclude with reflections on societal influences of the recent technological leap and future research directions.

Sébastien Bubeck et al. Sparks of Artificial General Intelligence: Early experiments with GPT-4. Mar. 2023.

System 1 & System 2

Current AI: System 1



刺激无响应 - 意识的，肌肉记忆的

Future AI: System 2

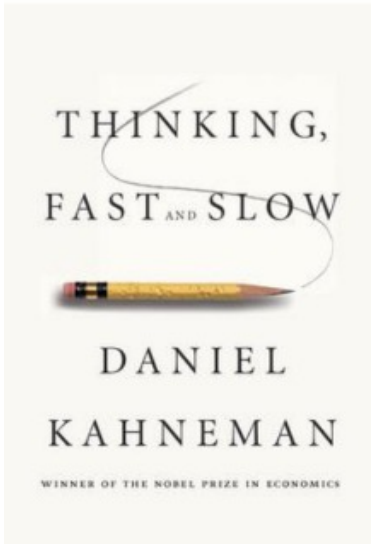
有意识的，逻辑的

Source: https://www.youtube.com/watch?v=2KVFMc7q2qs

Driving over Roundabout: Look others, assess, decide, act…

Gweon, H., and R. Saxe. "Developmental cognitive neuroscience of theory of mind." *Neural circuit development and function in the brain*. 2013.
Yoshua Bengio DEEP LEARNING FOR SYSTEM 2 PROCESSING and From Conscious Processing to System 2 Deep Learning

System 1 & System 2

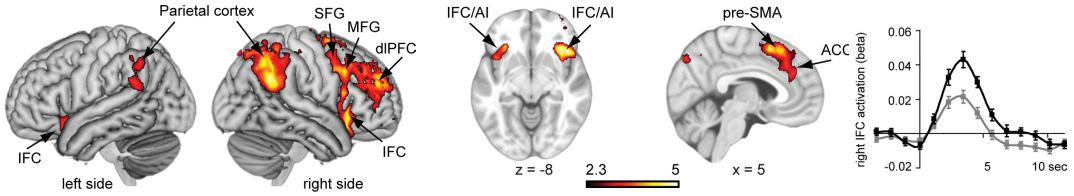


Think Fast and Slow: a book with 400+ pages by **Daniel Kahneman**

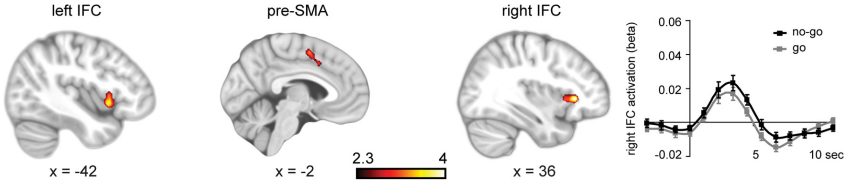
System 2:
- 缓慢的
- 有序的
- 有逻辑的
- **有意识的**
- 有语言表达的
- 有算法合理规划的
- 显性知识

System 1:
- 快速的
- 单步并行的
- **直觉的**
- 无意识的
- 无语言表达的
- 惯常的
- 隐性知识

System 1和System 2在脑部激活的区域是完全不同的



System 2 Conscious control



System 1 Nonconscious control

Van Gaal, et al. "Unconscious activation of the prefrontal no-go network." (2010)

机器意识的层次和路径

机器意识的级别	哲学问题归属	功能性描述	应用
C4：感受意识	困难问题	具有主观性感受和情感体验	**自主交流交互型，服务型机器人**
C3：自我意识	困难问题	能够自我识别与自我定位 具有内省反思能力和元认知	**环境自适应机器人，元学习**
C2：机制层面	容易问题	意识产生的机制基础 超长记忆和注意力，超强推理决策能力	**复杂系统下安全可靠的感知决策智能**
C1：认知层面	容易问题	有一定记忆、注意力、推理和决策能力	**自然语言处理，语音识别，多模态大模型，辅助决策智能，简单的群体智能，AIGC**
C0：感知层面	容易问题	无意识的应激反应，简单的映射	**计算机视觉，OCR识别，声音感知，触觉感知**

秦瑞林等. 机器意识研究综述. 自动化学报. 2021.

未来具身智能的智能范式融合

- 机器意识、认知智能、感知智能、控制智能的融合

1~3hz

Consciousness 意识层
建立机器自主意识

3~10hz

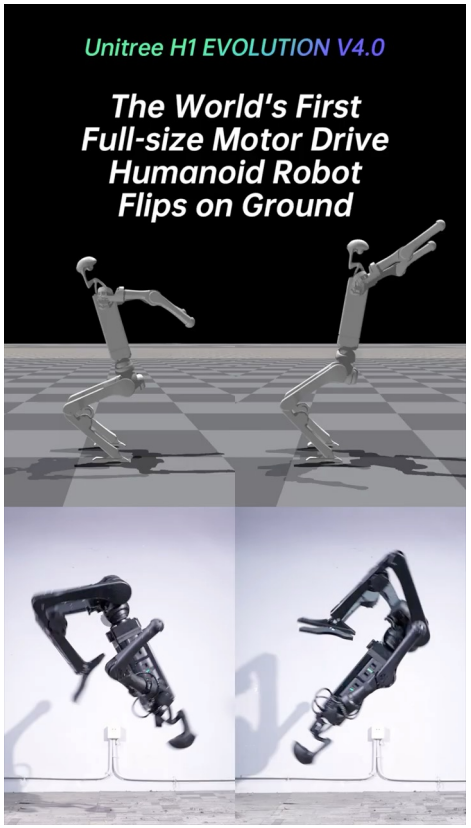
Cognition 认知层
基于LLM Agent做认知推理、任务规划

10~100hz

Perception 感知层
多模态感知环境

Control 控制层
给出控制信号

Intelligence



Agility

课程结语

- 2022年机器学习课，我写到：“我们仍然离强人工智能（通用人工智能，Artificial General Intelligence, AGI）很遥远，还有很长的路需要走”
- 2023年机器学习课，我写到：“我们已经可以看到通用人工智能的早期版本就在眼前”
- 2024年机器学习课，同学们的一项大作业是使用大模型来完成机器学习大作业，我们已经正式进入机器学习新时代
- 机器学习是迄今为止我们拥有的最强大的AI工具
- 祝愿同学们怀揣机器学习这一利器在今后的科研之路上大展宏图！

THANK YOU