《神经网络与深度学习》

模型独立的学习方式

https://nndl.github.io/

模型独立的学习方式

- 这些学习方式不限于具体的模型
  - 前馈神经网络、循环神经网络还是其他模型

- 然而一种学习方式往往会对符合某种特性的模型更加青睐
  - 集成学习往往和方差大的模型组合时效果显著。

内容

- 集成学习
- 自训练和协同训练
- 多任务学习
- 迁移学习
  - 归纳迁移学习
  - 转导迁移学习
- 元学习
- 终身学习

集成学习

- 通过某种策略将多个模型集成起来，通过群体决策来提高决策准确率。
- 三个臭皮匠赛过诸葛亮

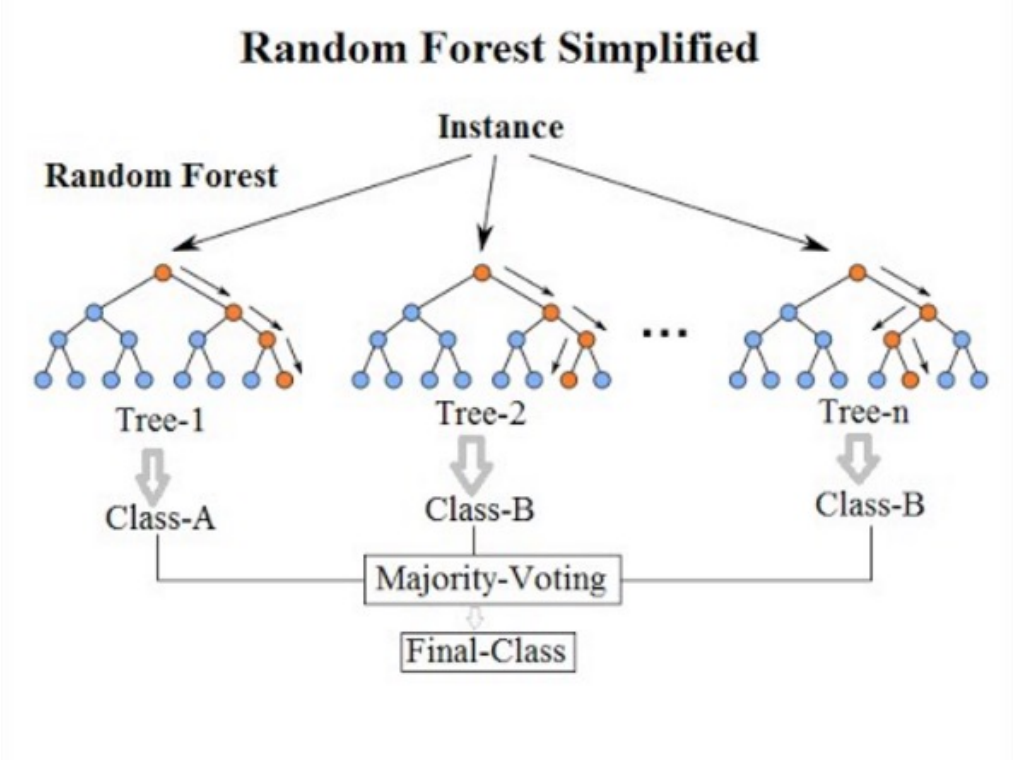
定理 10.1：对于 M 个不同的模型 $f_1(\mathbf{x}), \cdots, f_M(\mathbf{x})$，平均期望错误为 $\bar{\mathcal{R}}(f)$。基于简单投票机制的集成模型 $f^{(c)}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} f_m(\mathbf{x})$，其期望错误在 $\frac{1}{M}\bar{\mathcal{R}}(f)$ 和 $\bar{\mathcal{R}}(f)$ 之间。

$$
\begin{aligned}
\mathcal{R}(F) &= \mathbb{E}_{\boldsymbol{x}}\left[\left(\frac{1}{M}\sum_{m=1}^{M} f_m(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}_{\boldsymbol{x}}\left[\left(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right)^2\right] \\
&= \frac{1}{M^2}\mathbb{E}_{\boldsymbol{x}}\left[\sum_{m=1}^{M}\sum_{n=1}^{M} \epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right] \\
&= \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} \mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right],
\end{aligned}
$$

集成方式

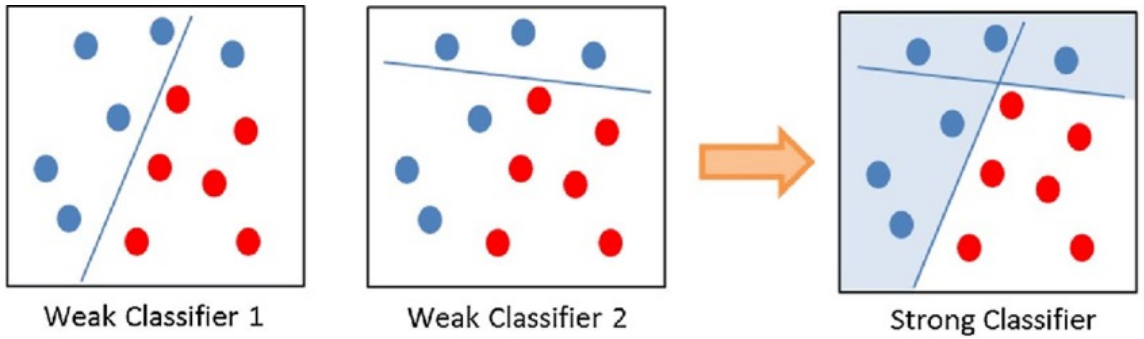
- Bagging类
  - Bagging（Bootstrap Aggregating）是一个通过不同模型的训练数据集的独立性来提高不同模型之间的独立性。我们在原始训练集上进行有放回的随机采样，得到M 比较小的训练集并训练M 个模型，然后通过投票的方法进行模型集成。
  - 随机森林（Random Forest）是在Bagging的基础上再引入了随机特征，进一步提高每个基模型之间的独立性。在随机森林中，每个基模型都是一棵决策树。

随机森林


Random Forest Simplified
Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

集成方式

- Boosting类
  - 按照一定的顺序来先后训练不同的基模型，每个模型都针对前序模型的错误进行专门训练。根据前序模型的结果，来调整训练样本的权重，从而增加不同基模型之间的差异性。
  - AdaBoost（Adaptive Boosting）算法



AdaBoost

Input :

- A training set $S = ((\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m))$.

Initialization :

- Maximum number of iterations T;
- initialize the weight distribution $\forall i \in \{1, \ldots, m\}, D^{(1)}(i) = \frac{1}{m}$.

for $t = 1, \ldots, T$ **do**

- Learn a classifier $f_t : \mathbb{R}^d \to \{-1, +1\}$ using distribution $D^{(t)}$
- Set $\epsilon_t = \sum_{i: f_t(\mathbf{x}_i) \neq y_i} D^{(t)}(i)$
- Choose $a_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
- Update the weight distribution over examples

$$\forall i \in \{1, \ldots, m\}, D^{(t+1)}(i) = \frac{D^{(t)}(i) e^{-a_t y_i f_t(\mathbf{x}_i)}}{Z^{(t)}}$$

where $Z^{(t)} = \sum_{i=1}^{m} D^{(t)}(i) e^{-a_t y_i f_t(\mathbf{x}_i)}$ is a normalization factor such that $D^{(t+1)}$ remains a distribution.

Output : The voted classifier $\forall \mathbf{x}, F(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} a_t f_t(\mathbf{x})\right)$

自训练和协同训练

- 是一种半监督学习方法。利用少量标注数据和大量无标注数据进行学习。

自训练

- 自训练(self-training/bootstrapping)
  - 使用标注数据来训练一个模型，并使用该模型预测无标签样本的标签，把置信度较高的样本及其预测的为标签加入训练集，然后重新训练新的模型

自训练

算法 10.2　自训练的训练过程

输入: 标注数据集 $\mathcal{L} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$;

无标注数据集 $\mathcal{U} = \{\boldsymbol{x}^{(m)}\}_{m=1}^{M}$;

迭代次数 T; 每次迭代增加样本数量 P;

for $t = 1 \cdots T$ **do**

　根据训练集 $\mathcal{L}$, 训练模型 f;

　使用模型 f 对无标注数据集 $\mathcal{U}$ 的样本进行预测, 选出预测置信度高的 P 个样本 $\mathcal{P} = \{(\boldsymbol{x}^{(p)}, f(\boldsymbol{x}^{(p)}))\}_{p=1}^{P}$;

　更新训练集:

$$\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{P}, \qquad \mathcal{U} \leftarrow \mathcal{U} - \mathcal{P}.$$

end

输出: 模型 f

协同训练

- 协同训练(co-training)
  - 自训练的改进方法，通过基于不同视角(view)的分类器来互相促进
  - 协同训练要求两种视角是条件独立的，如果两种视角完全一样，则退化为自训练。

协同训练

算法 10.3　协同训练的训练过程

输入: 标注数据集 $\mathcal{L}=\{(\boldsymbol{x}^{(n)},y^{(n)})\}_{n=1}^{N}$;

无标注数据集 $\mathcal{U}=\{\boldsymbol{x}^{(m)}\}_{m=1}^{M}$;

迭代次数 T; 候选池大小 K; 每次迭代增加样本数量 $2P$;

for $t=1\cdots T$ **do**

　根据训练集 $\mathcal{L}$ 的视角 V_1 训练训练模型 f_1;

　根据训练集 $\mathcal{L}$ 的视角 V_2 训练训练模型 f_2;

　从无标注数据集 $\mathcal{U}$ 上随机选取一些样本放入候选池 $\mathcal{U}'$, 使得 $|\mathcal{U}'|=K$;

　for $f\in f_1,f_2$ **do**

　　使用模型 f 预测候选池 $\mathcal{U}'$ 中的样本的伪标签;

　　for $p=1\cdots P$ **do**

　　　根据标签分布, 随机选取一个标签 y;

　　　从 $\mathcal{U}'$ 中选出伪标签为 y, 并且预测置信度最高的样本 $\boldsymbol{x}$;

　　　更新训练集:

$$\mathcal{L}\leftarrow\mathcal{L}\cup\{(\boldsymbol{x},y)\},\qquad \mathcal{U}'\leftarrow\mathcal{U}'-\{(\boldsymbol{x},y)\}.$$

　　end

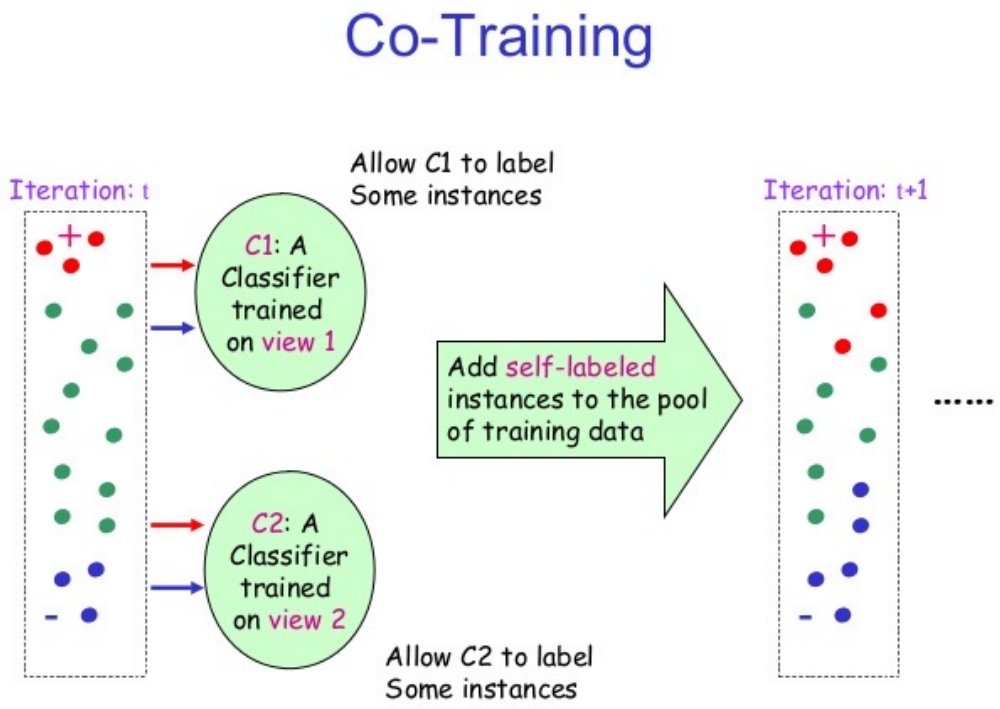
　end

end

输出: 模型 f_1,f_2

协同训练（ Co-Training ）

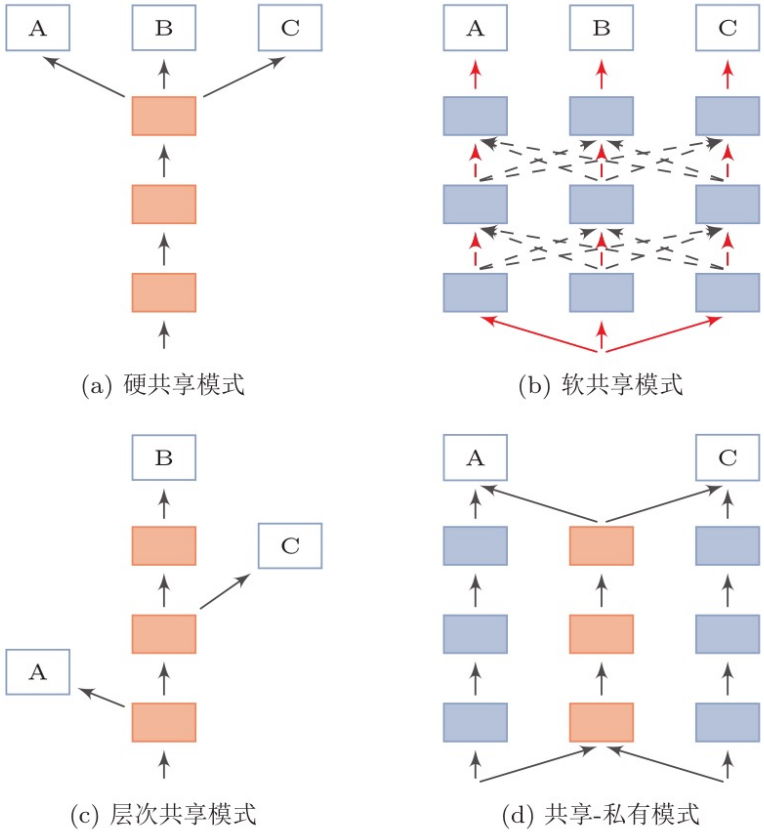
- Multi-View Learning



多任务学习

- 多任务学习(multi-task learning)
  - 同时学习多个相关任务，让这些任务在学习过程中共享知识，利用多个任务之间的相关性来改进模型在每个任务上的性能和泛化能力。

多任务学习（ Multitask Learning ）



多任务学习（ Multitask Learning ）

- 学习步骤

假设这 M 个任务对应的模型分别为 $f_m(\boldsymbol{x};\theta), 1 \le m \le M$，多任务学习的联合目标函数为所有任务损失函数的线性加权.

$$\mathcal{L}(\theta) = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \eta_m \mathcal{L}_m\left(f_m(x^{(m,n)};\theta), y^{(m,n)}\right), \tag{10.25}$$

其中 $\mathcal{L}_m(\cdot)$ 为第 m 个任务的损失函数，η_m 是第 m 个任务的权重，θ 表示包含了共享模块和私有模块在内的所有参数. 权重可以根据不同任务的重要程度来赋值，也可以根据任务的难易程度来赋值. 通常情况下，所有任务设置相同的权重，即 $\eta_m = 1, 1 \le m \le M$.

- 1）联合训练阶段
  - 每次迭代，随机条挑选一个任务，从这个任务随机选择一些训练样本，计算梯度并更新参数
- 2）单任务精调阶段
  - 基于多任务学习得到的参数，分别在每个单独任务上进行精调(fine-tuning)

多任务学习

算法 10.4　多任务学习中联合训练过程

输入: M 个任务的数据集 $\mathcal{D}_m, 1 \le m \le M$;
每个任务的批量大小 $K_m, 1 \le m \le M$;
最大迭代次数 T,学习率 α;

随机初始化参数 θ_0;
for $t = 1 \cdots T$ **do**
　// 准备 M 个任务的数据
　for $m = 1 \cdots M$ **do**
　　将任务 m 的训练集 $\mathcal{D}_m$ 中随机划分为 $c = \frac{N_m}{K_m}$ 个小批量集合:
　　$\mathcal{B}_m = \{\mathcal{I}_{m,1}, \cdots, \mathcal{I}_{m,c}\}$;
　end
　合并所有小批量样本 $\bar{\mathcal{B}} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_M$;
　随机排序 $\bar{\mathcal{B}}$;
　foreach $\mathcal{I} \in \bar{\mathcal{B}}$ **do**
　　计算小批量样本 $\mathcal{I}$ 上的损失 $\mathcal{L}(\theta)$;　// 只计算 $\mathcal{I}$ 在对应任务上的损失
　　更新参数:$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \nabla_\theta \mathcal{L}(\theta)$;
　end
end
输出: 模型 $f_m, 1 \le m \le M$

多任务学习 vs. 单任务学习

- 多任务学习通常比单任务学习有更好的泛化能力，原因如下：
  - 1）训练集更大，多任务之间有相关性，相当于隐式的数据增强
  - 2）共享模块需要兼顾所有任务，一定程度上避免了模型过拟合
  - 3）获得更好的表示
  - 4）每个任务“选择性”利用其他任务中学习到的隐藏特征，提高自身的能力

迁移学习

- 迁移学习（transfer learning)
  - 将相关任务的训练数据中的可泛化知识迁移到目标任务上。

表 10.1　迁移学习和标准机器学习的比较

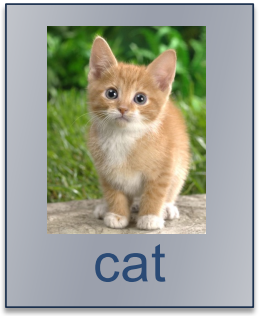
学习类型	样本空间	概率分布
标准机器学习	$\mathcal{X}_S = \mathcal{X}_T, \mathcal{Y}_S = \mathcal{Y}_T$	$p_S(\boldsymbol{x}, y) = p_T(\boldsymbol{x}, y)$
迁移学习	$\mathcal{X}_S \neq \mathcal{X}_T$ 或 $\mathcal{Y}_S \neq \mathcal{Y}_T$ 或 $p_S(\boldsymbol{x}, y) \neq p_T(\boldsymbol{x}, y)$	

Transfer Learning

http://weebly110810.weebly.com/396403913129399.html

http://www.sucaitianxia.com/png/cartoon/200811/4261.html

Dog/Cat Classifier



Data ***not directly related to*** the task considered

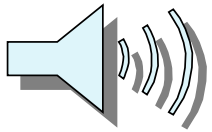
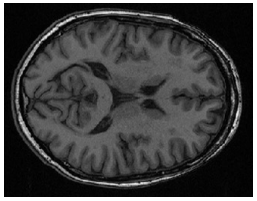
Similar domain, different tasks

Different domains, same task

Why?

http://www.bigr.nl/website/structure/main.php?page=researchlines&subpage=project&id=64

http://www.spear.com.hk/Translation-company-Directory.html

	Task Considered	Data not directly related
Speech Recognition	 广东话	 English Chinese ……
Image Recognition	 Medical Images	
Text Analysis	 Specific domain	 Webpages

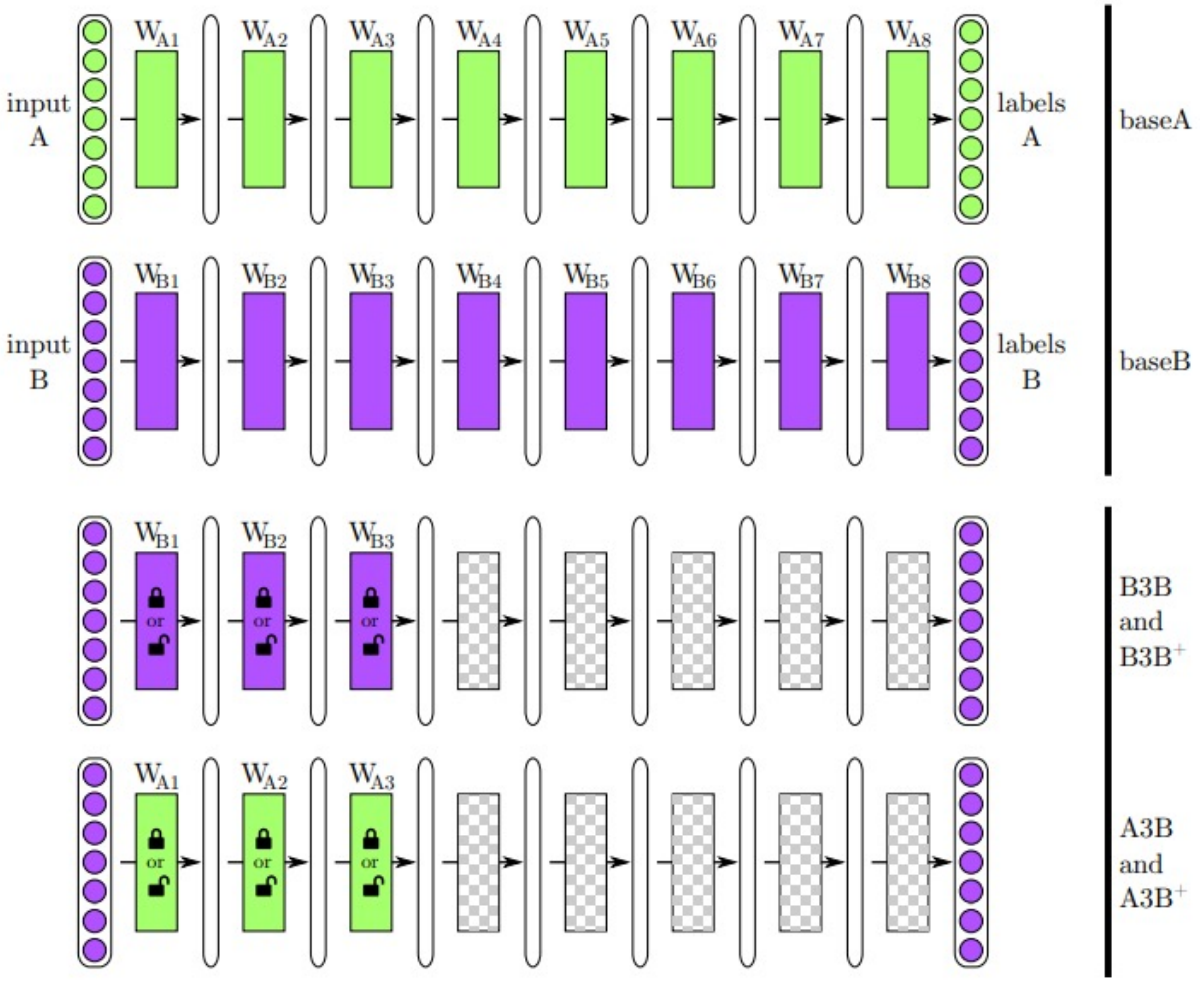
迁移学习

- 归纳迁移学习(inductive transfer learning)
  - $X_S = X_T, Y_S \neq Y_T$

- 当D^S有大量标注数据，可直接将源领域训练模型迁移到目标领域
  - 在image net上预训练的模型AlexNet, VGG等
- 当D^S有大量无标注数据，源领域任务可转换为无监督学习任务
  - Word2vec, GloVe, ELMO, openAI GPT, BERT等

- 基于特征的方式
- 精调的方式

Layer Transfer - Image

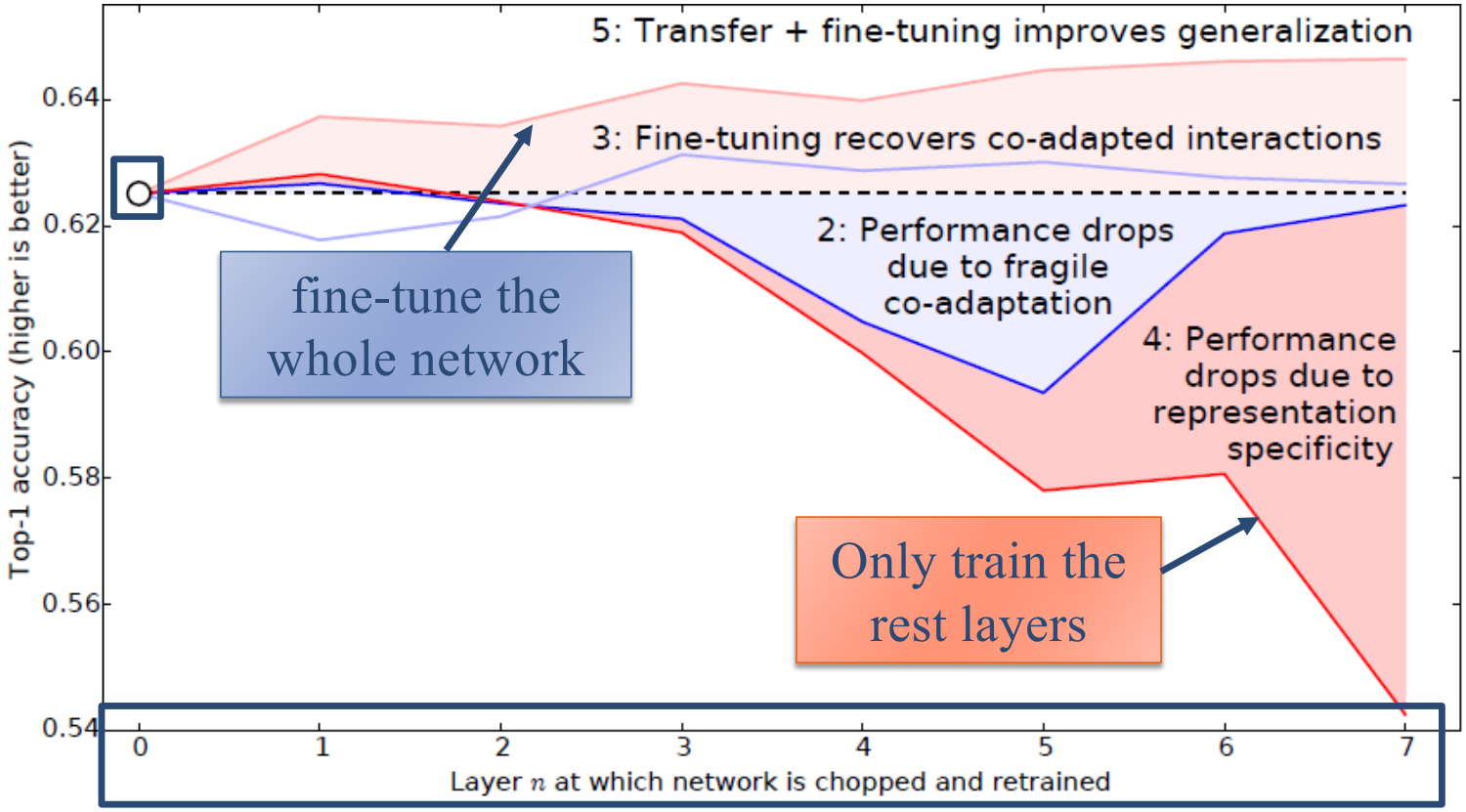


Source: 500 classes from ImageNet

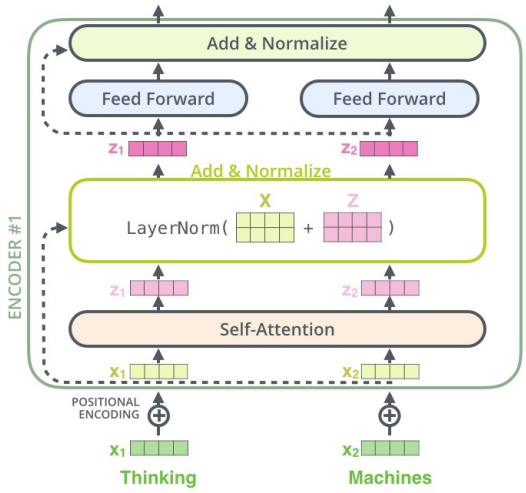
Target: another 500 classes from ImageNet

Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, “How transferable are features in deep neural networks?”, NIPS, 2014

Layer Transfer - Image



BERT



In this work, we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A.[3] We primarily report results on two model sizes: $\textbf{BERT}_{\textbf{BASE}}$ (L=12, H=768, A=12, Total Parameters=110M) and $\textbf{BERT}_{\textbf{LARGE}}$ (L=24, H=1024, A=16, Total Parameters=340M).

Task #1: Masked Language Model

- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.

Task #2: Next Sentence Prediction

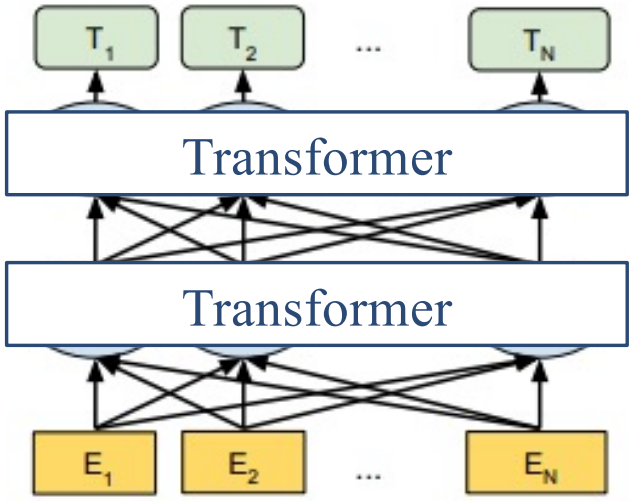
预测句子B是否承接句子A

BERT

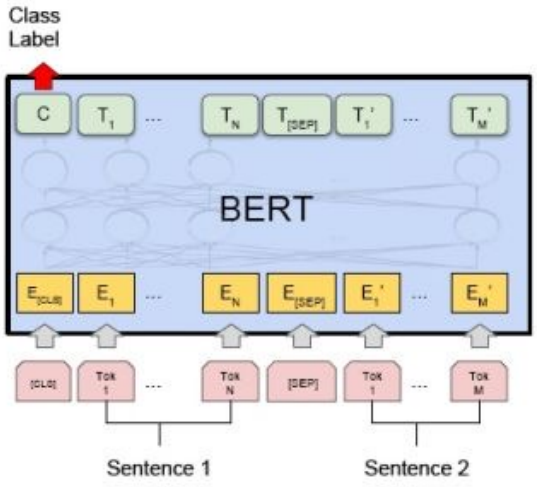
Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019

Corpus: the black dog **jumped** over the puddle, followed by a boy
Mask prediction & sentence prediction: [CLS] the black dog **[MASK]** over the [MASK] [SEP] followed [MASK] a boy [CLS] → label = 1
[CLS] the black dog **[MASK]** over the [MASK] [SEP] We have [MASK] amazing progress [MASK] NLP [CLS] → label = 0

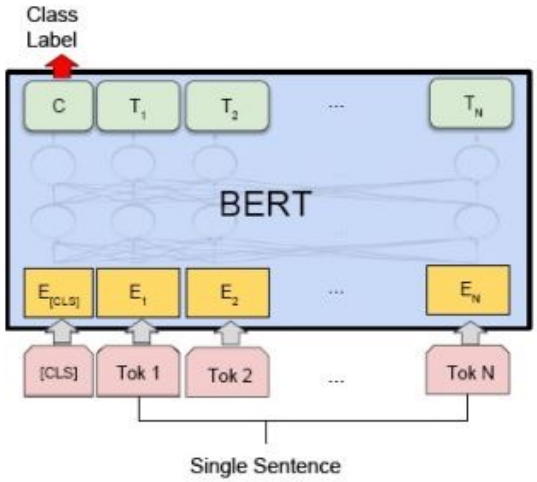
$$t_1, t_2, .., t_N = BERT(e_1, e_2, \dots, e_N; \theta)$$



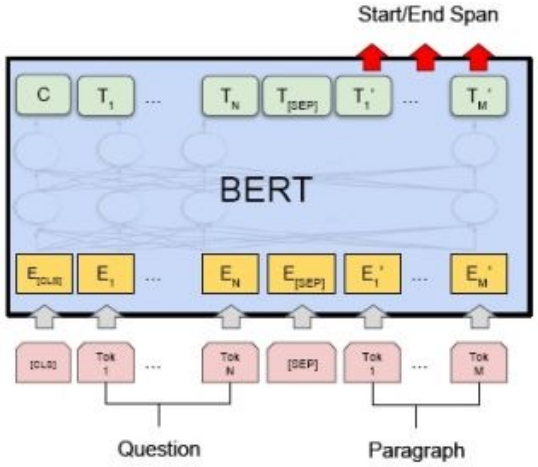
BERT-finetuning



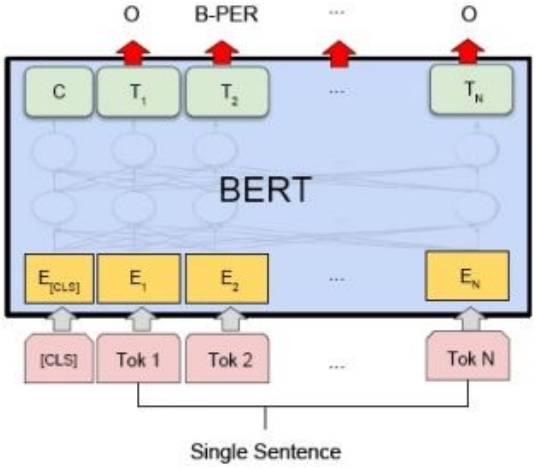
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA

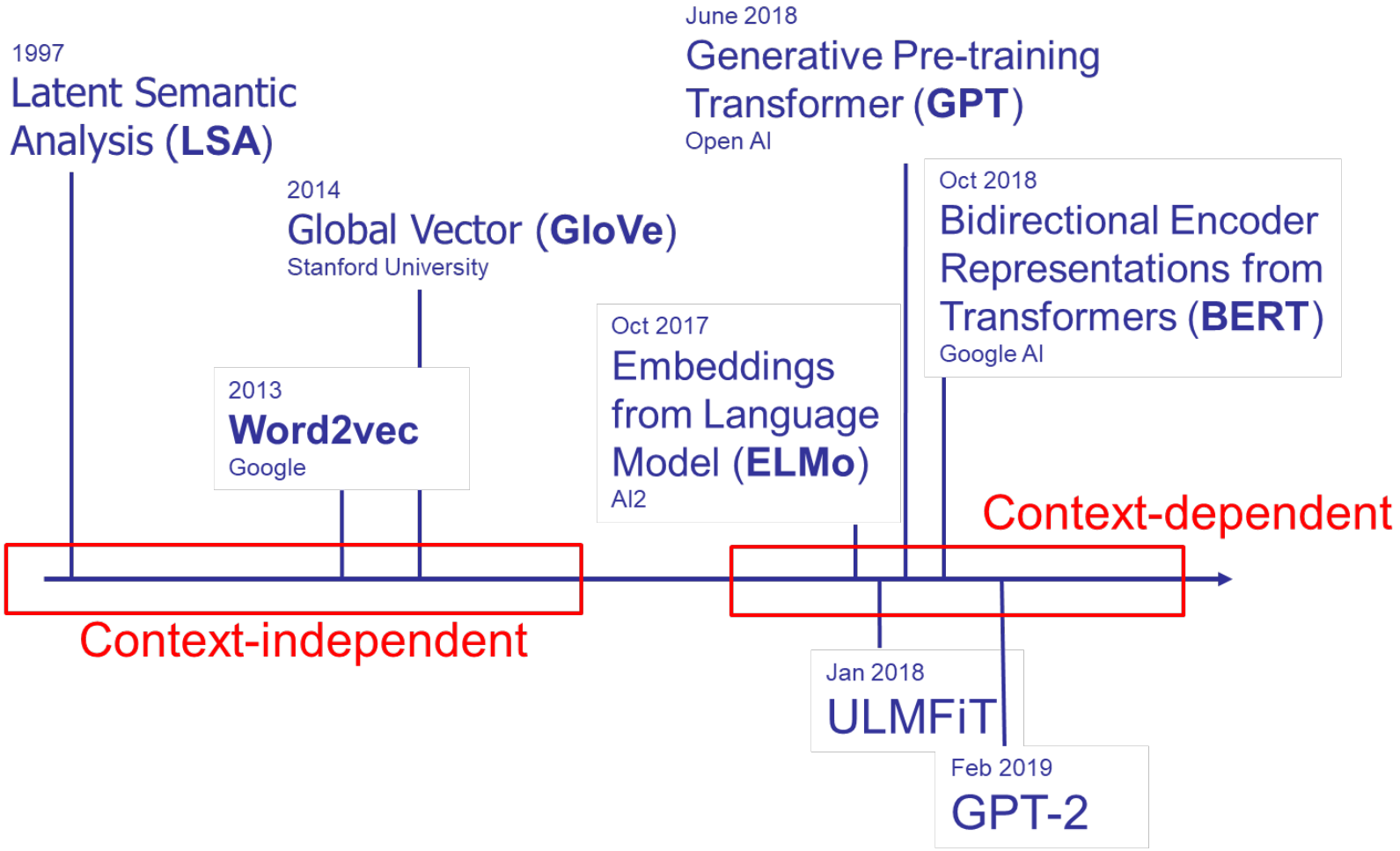


(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Timeline of word embedding



归纳迁移学习 vs. 多任务学习

- 主要区别如下：
  - 1）多任务学习是同时学习多个不同任务，归纳迁移学习通常分为两个阶段
  - 2）归纳迁移学习是单向的知识迁移，而多任务学习一般是希望提高所有任务的性能

迁移学习

- 转导迁移学习(transductive transfer learning)
  - $X_S \neq X_T, Y_S = Y_T$
  - $P_S(x, y) \neq P_T(x, y)$

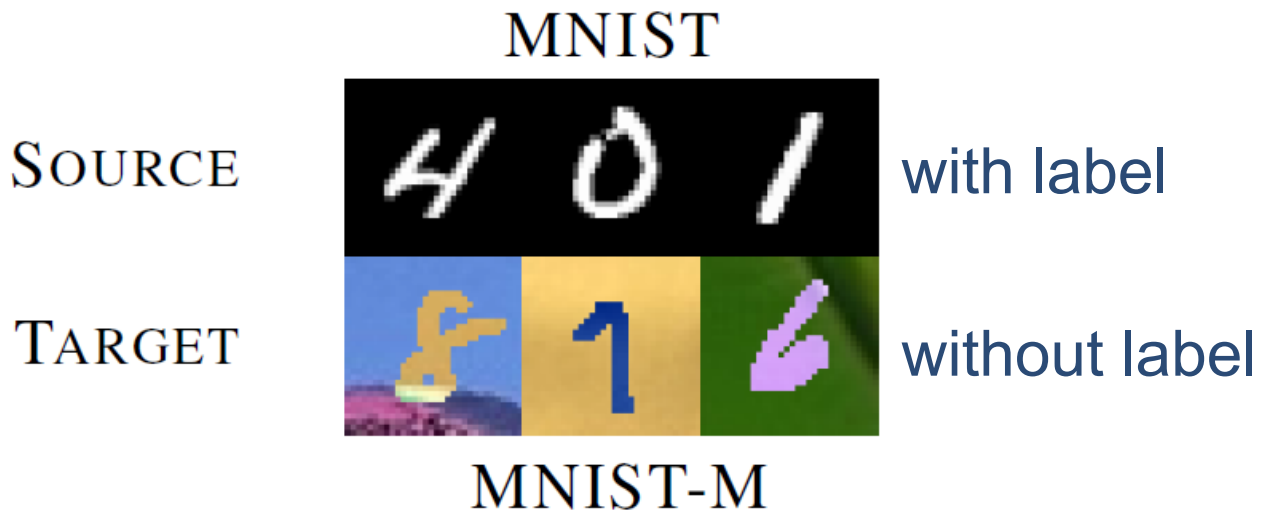
- 领域对抗学习(domain adversarial learning)

Task description

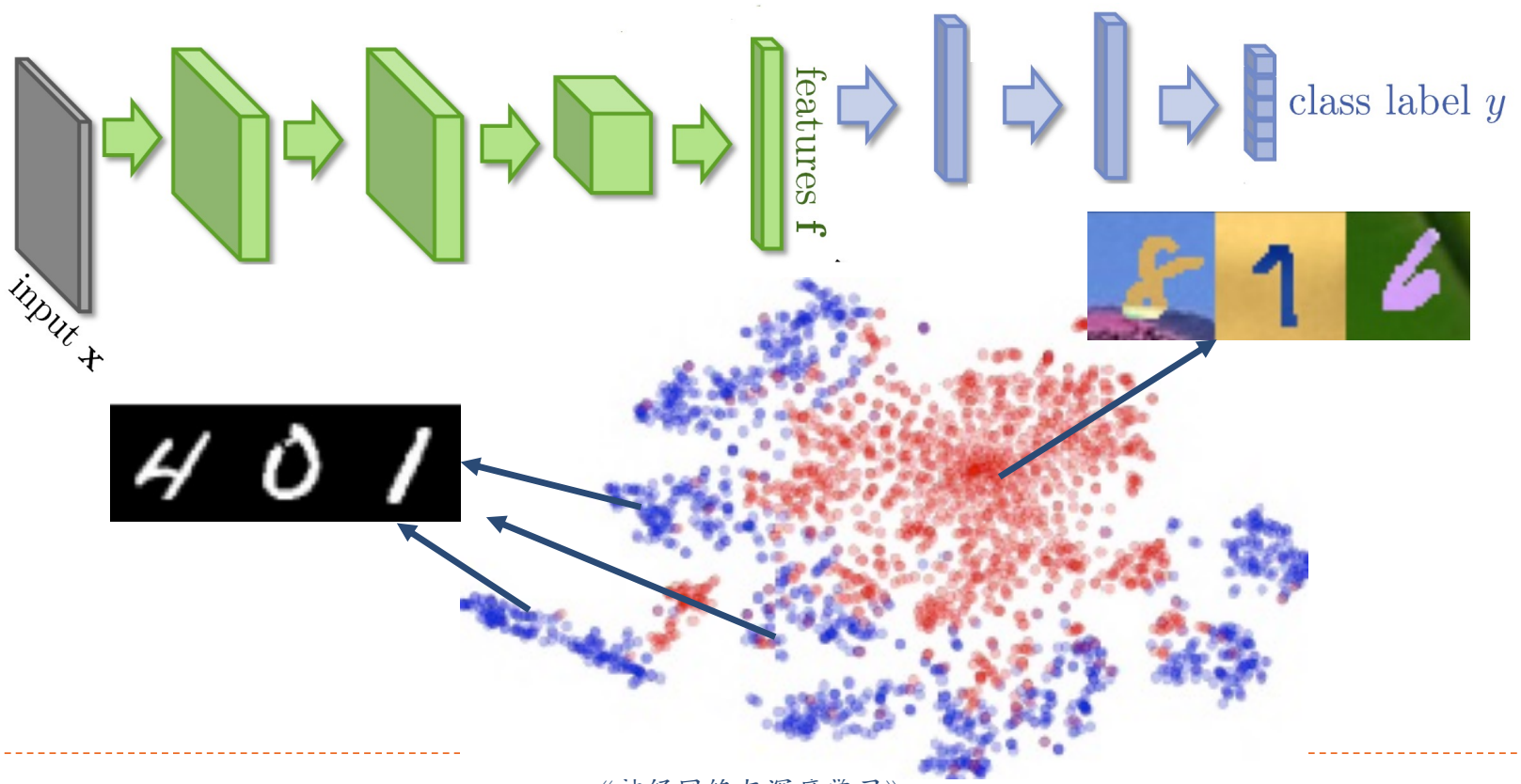
Bousmalis K, Trigeorgis G, Silberman N, et al., 2016. Domain separation networks. NIPS.

- Source data: (x^s, y^s)
- Target data: (x^t)

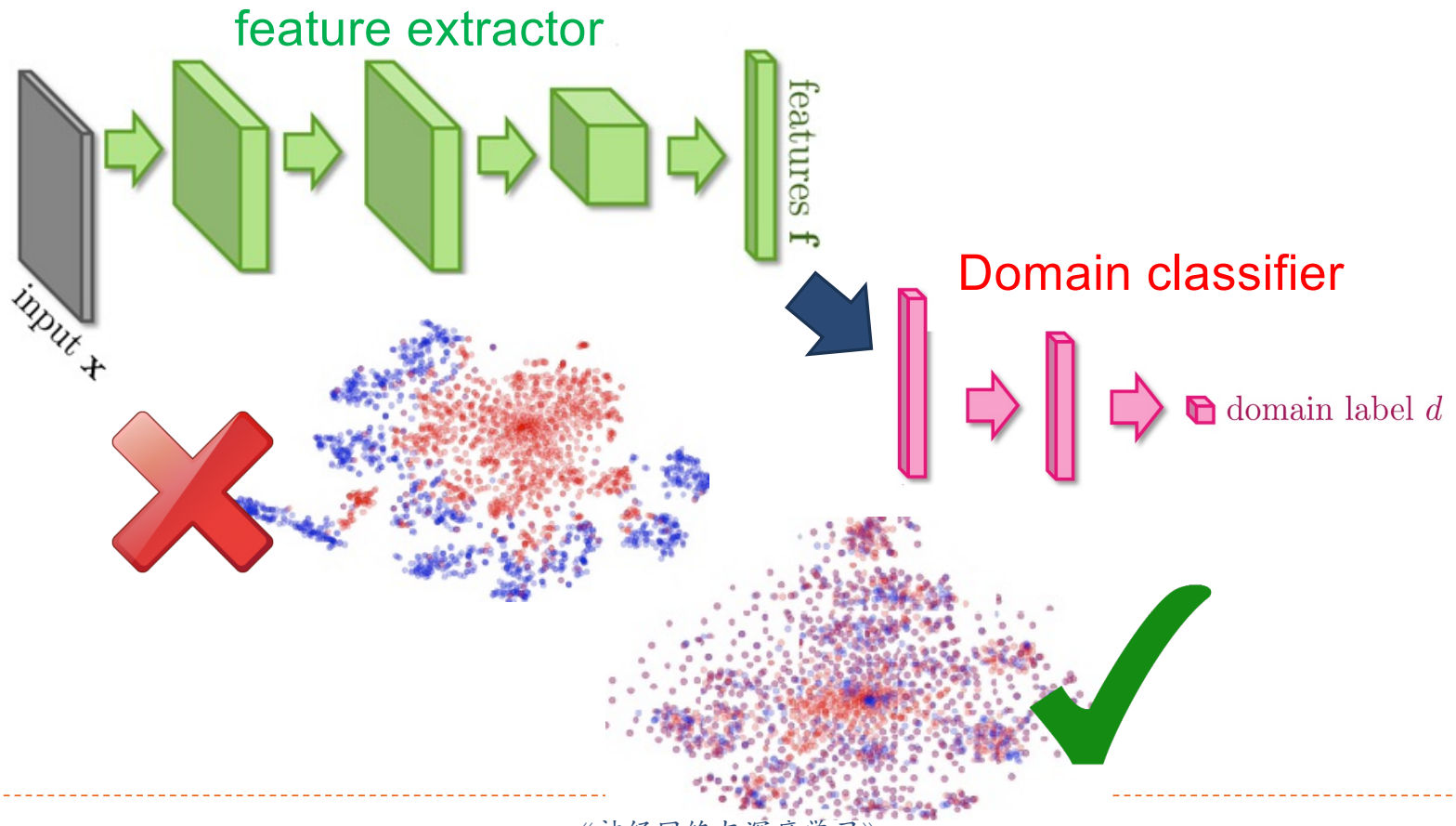
} Same task, mismatch



Domain-adversarial training



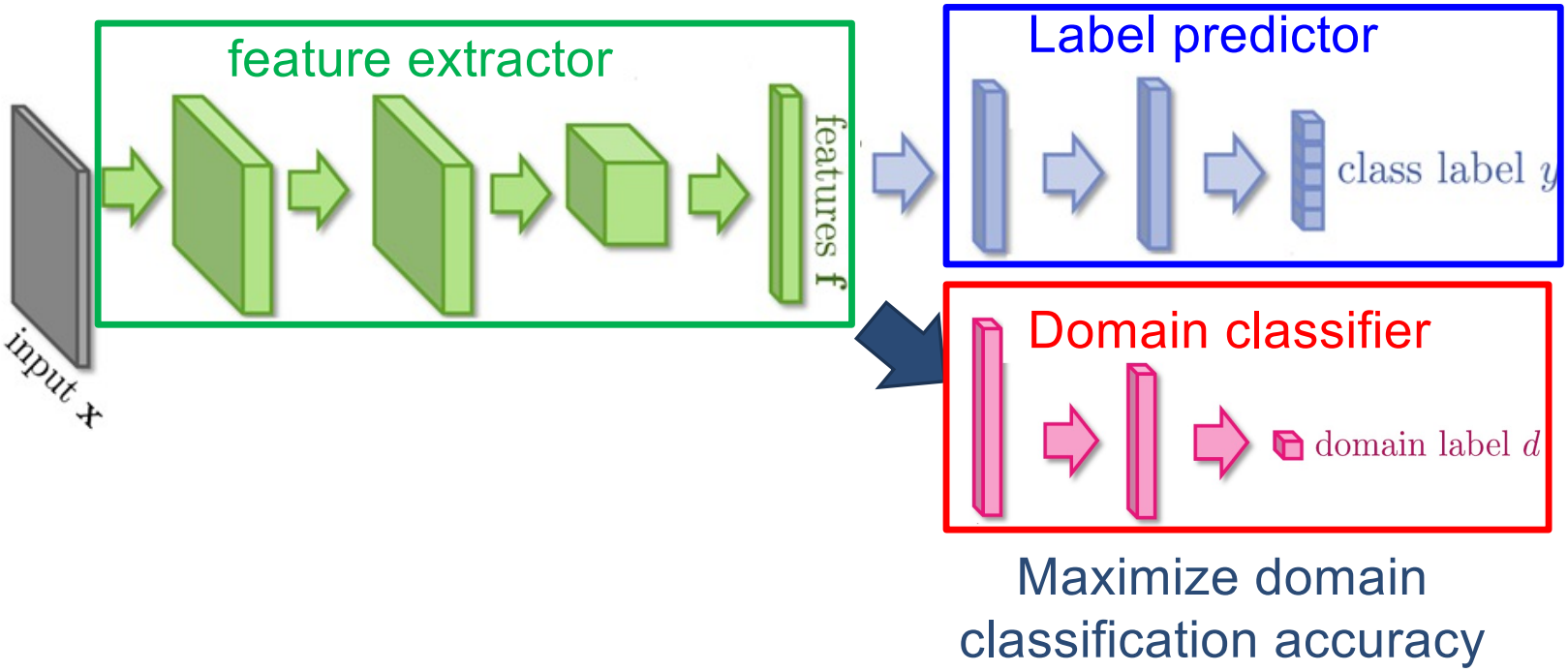
Domain-adversarial training



Domain-adversarial training

Maximize label classification accuracy + minimize domain classification accuracy

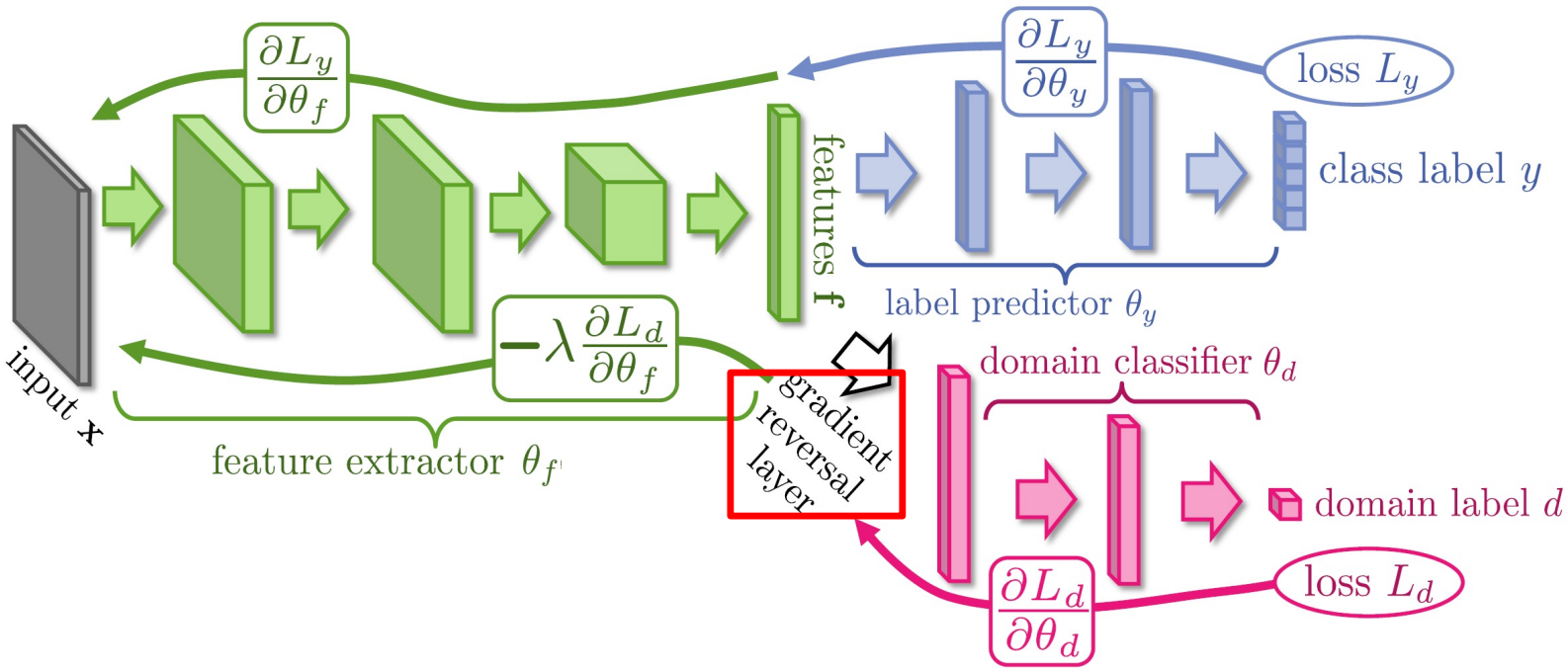
Maximize label classification accuracy



Maximize domain classification accuracy

This is a big network, but different parts have different goals.

Domain-adversarial training



Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016

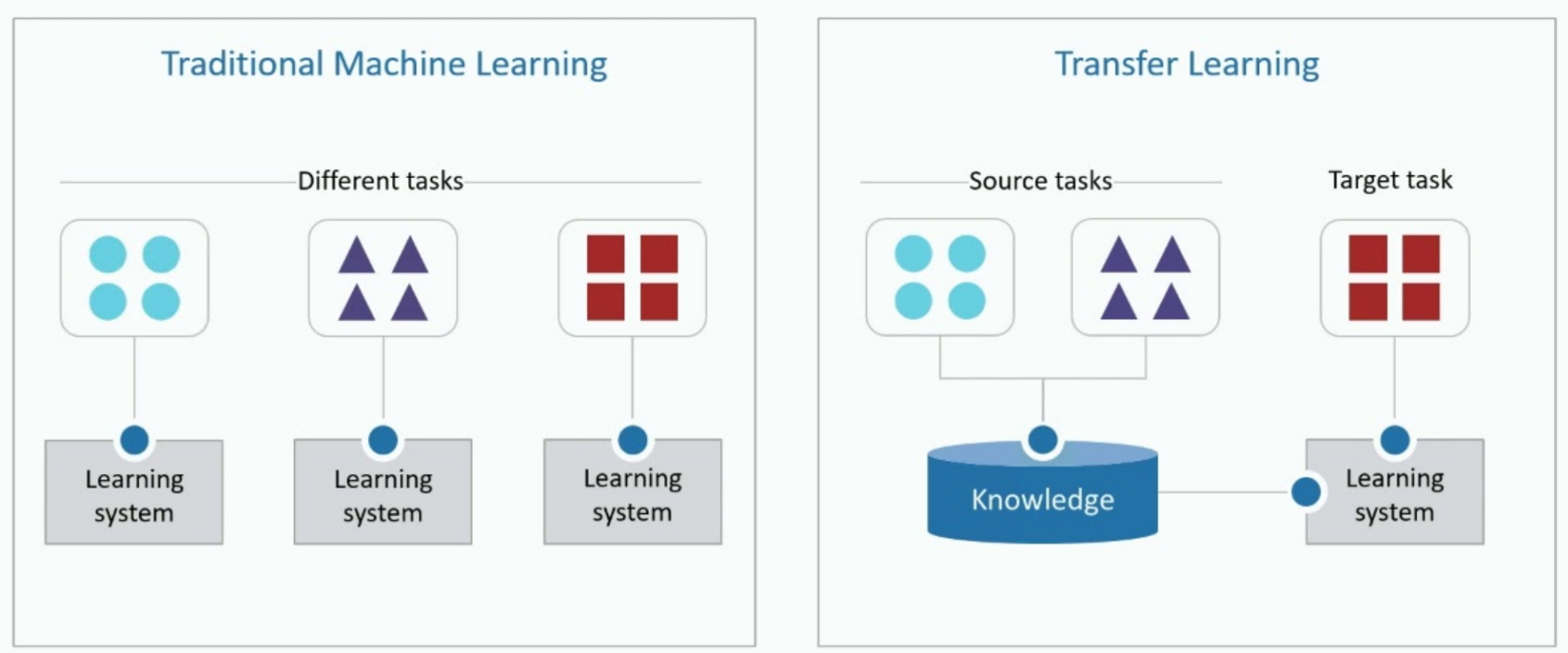
Domain-adversarial training

METHOD	SOURCE / TARGET	MNIST / MNIST-M	SYN NUMBERS / SVHN	SVHN / MNIST	SYN SIGNS / GTSRB
SOURCE ONLY		.5749	.8665	.5919	.7400
SA (FERNANDO ET AL., 2013)		.6078 (7.9%)	.8672 (1.3%)	.6157 (5.9%)	.7635 (9.1%)
PROPOSED APPROACH		**.8149** (57.9%)	**.9048** (66.1%)	**.7107** (29.3%)	**.8866** (56.7%)
TRAIN ON TARGET		.9891	.9244	.9951	.9987

Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016

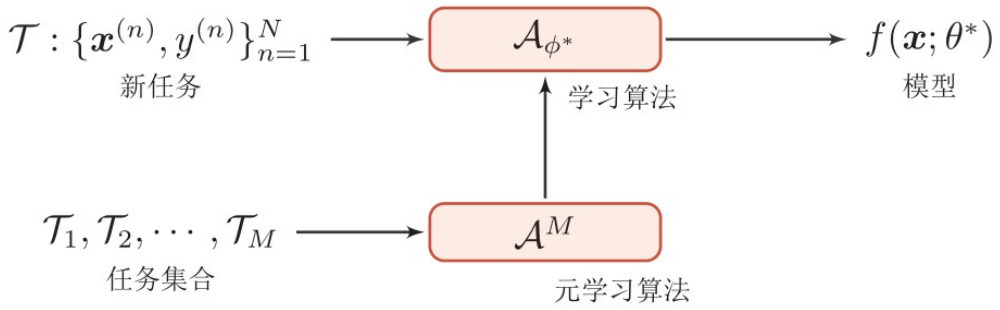
迁移学习（Transfer Learning）



元学习

- 元学习（meta learning)
  - 从已有任务中学习元知识，可以加速新任务的学习

- 基于优化器的学习



$$\theta_{t+1} = \theta_t - \alpha \nabla L(\theta_t) \quad \Longrightarrow \quad \theta_{t+1} = \theta_t + g_t(\nabla \mathcal{L}(\theta_t); \phi)$$

- 模型无关的元学习: Model-Agnostic Meta Learning(MAML)

迁移学习 vs. MAML

- 主要区别如下：
  - 迁移学习聚焦于寻找当前任务的最好参数，而MAML聚焦寻找一个能够快速调整到其他任务上的参数

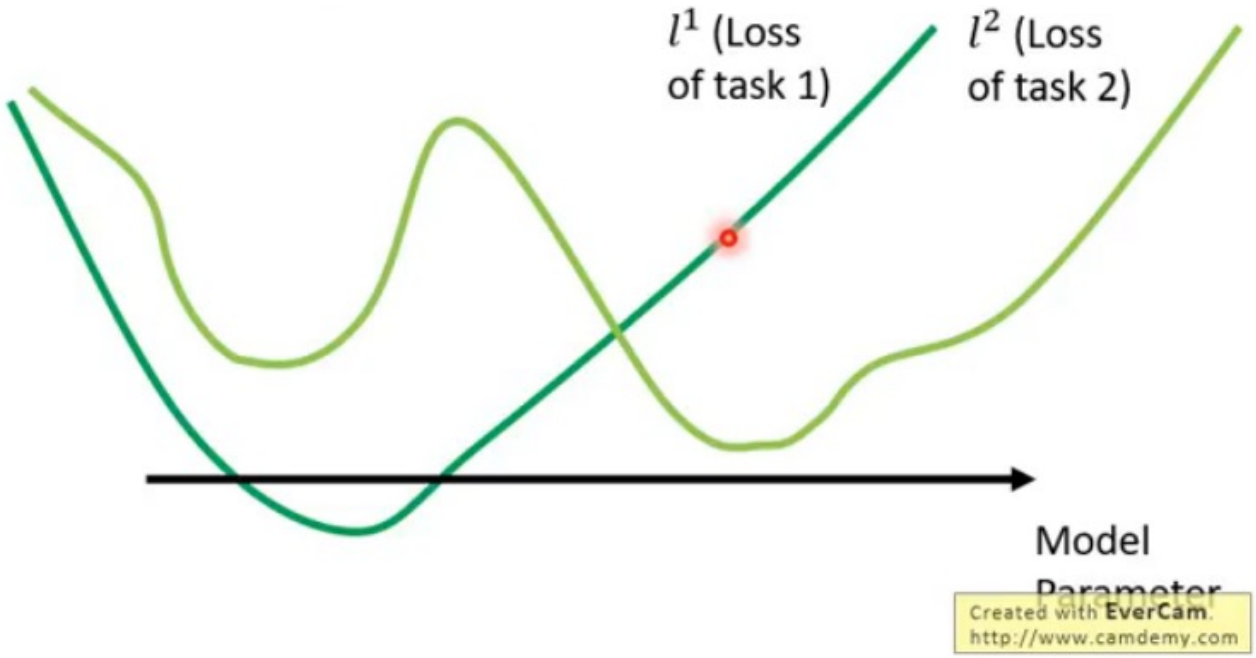
- MAML

$$L(\phi) = \sum_{n=1}^{N} l^n(\hat{\theta}_n)$$

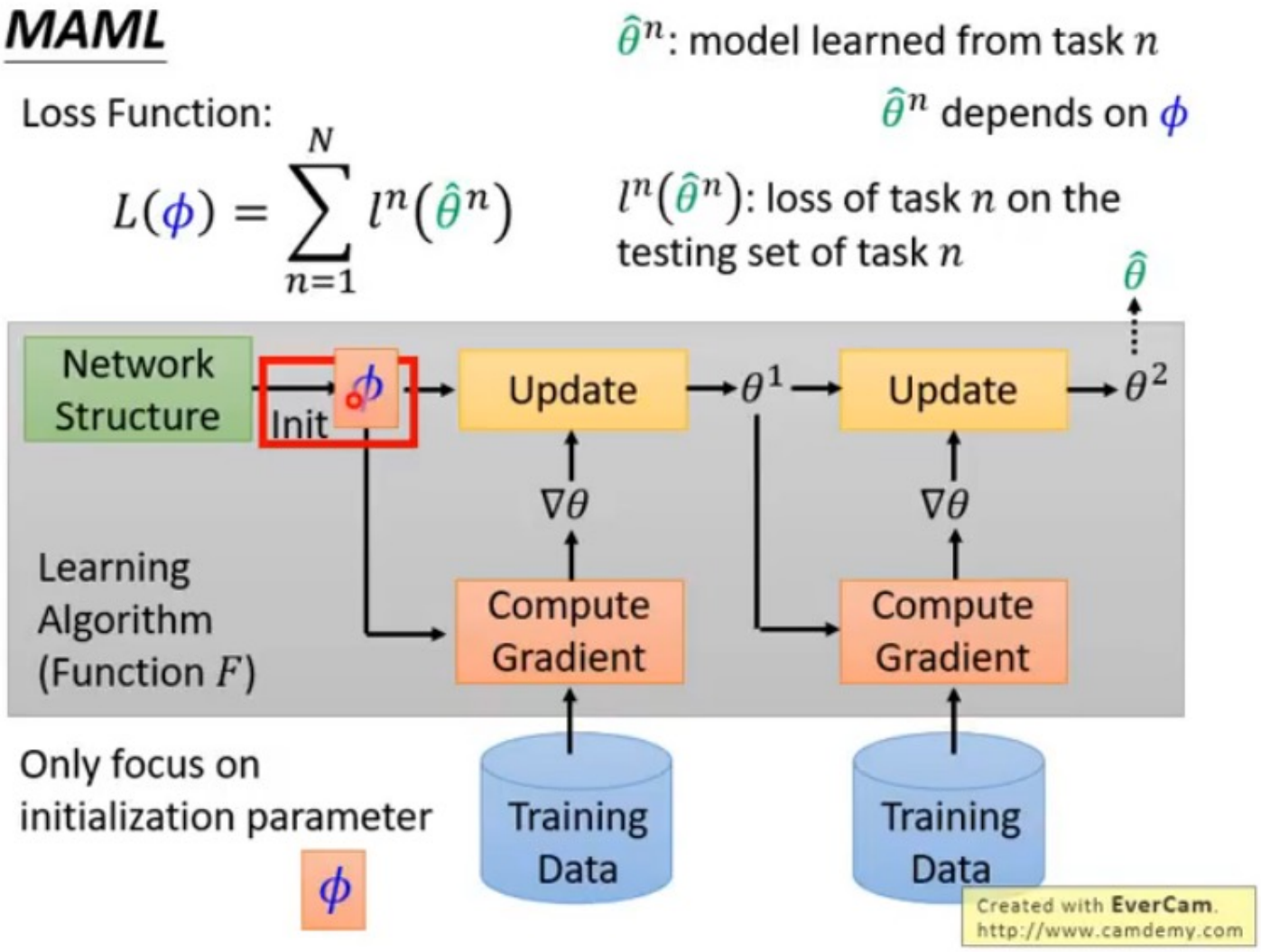
- Model pre-training

$$L(\phi) = \sum_{n=1}^{N} l^n(\phi)$$

迁移学习 vs. MAML

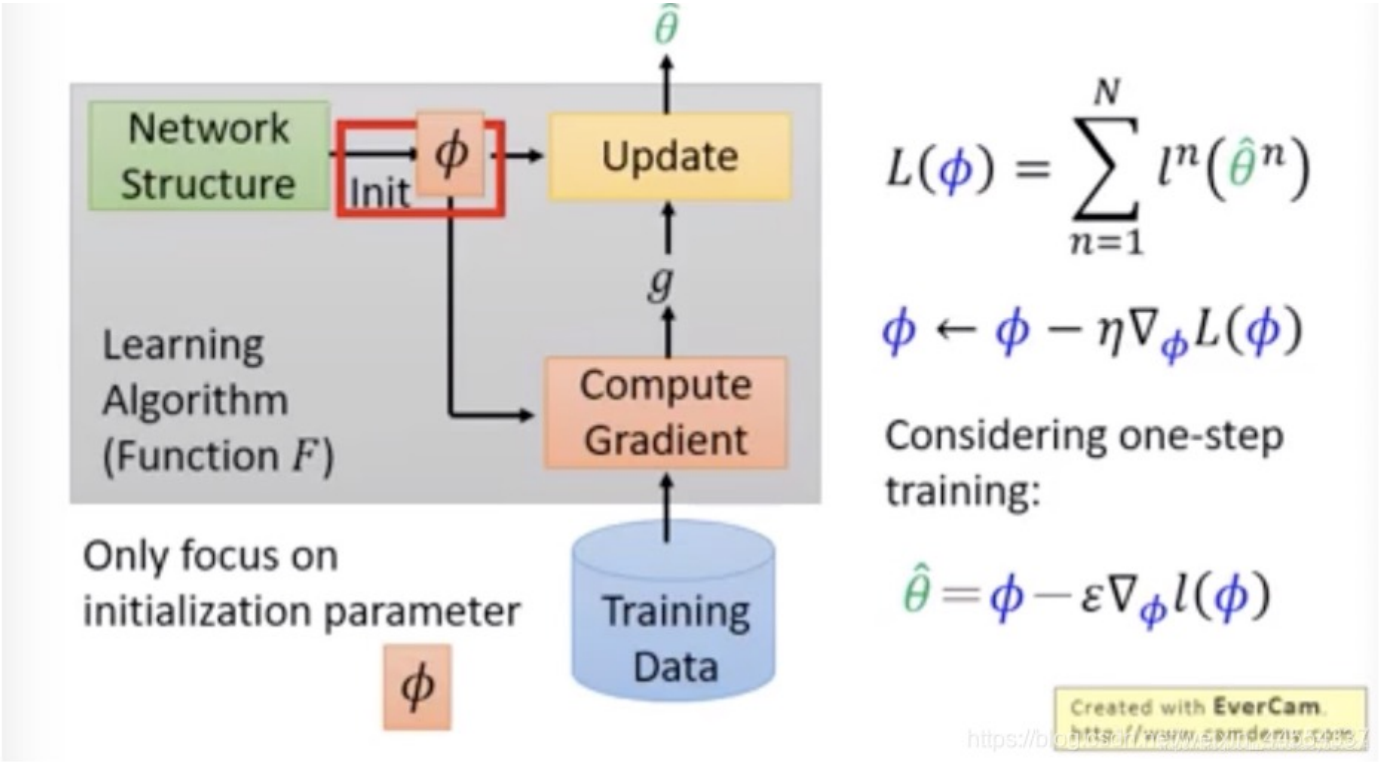

l^1 (Loss of task 1)
l^2 (Loss of task 2)
Model Parameter
Created with EverCam.
http://www.camdemy.com

MAML



Finn C, Abbeel P, Levine S, 2017. Model-agnostic meta-learning for fast adaptation of deep networks. JMLR

MAML



Finn C, Abbeel P, Levine S, 2017. Model-agnostic meta-learning for fast adaptation of deep networks. JMLR

MAML

算法 10.5　模型无关的元学习过程

输入: 任务分布 $p(\mathcal{T})$;

　　最大迭代次数 T, 学习率 α, β;

随机初始化参数 θ;

for $t = 1 \cdots T$ **do**

　根据 $p(\mathcal{T})$ 采样一个任务集合 $\{\mathcal{T}_m\}_{m=1}^{M}$;

　for $m = 1 \cdots M$ **do**

　　计算 $\nabla_\theta \mathcal{L}_{\mathcal{T}_m}(f_\theta)$;

　　计算任务适配的参数: $\theta'_m \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_m}(f_\theta)$;

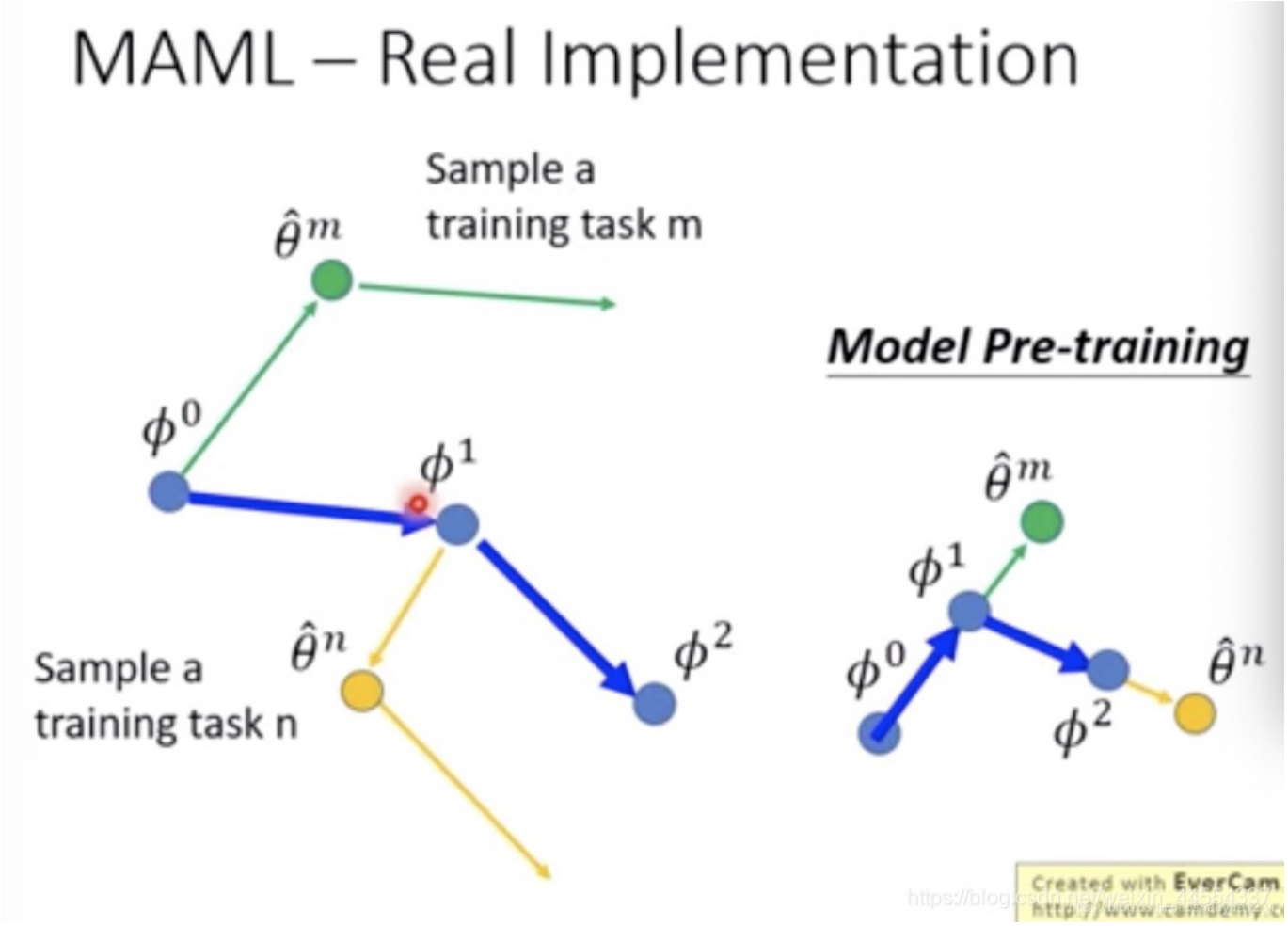
　end

　更新参数: $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{m=1}^{M} \mathcal{L}_{\mathcal{T}_m}(f_{\theta'_m})$;

end

输出: 模型 f_θ

迁移学习 vs. MAML


MAML – Real Implementation
Sample a training task m
$\hat{\theta}^m$
ϕ^0
ϕ^1
ϕ^2
$\hat{\theta}^n$
Sample a training task n
Model Pre-training
$\hat{\theta}^m$
ϕ^1
ϕ^0
ϕ^2
$\hat{\theta}^n$
Created with EverCam

MAML

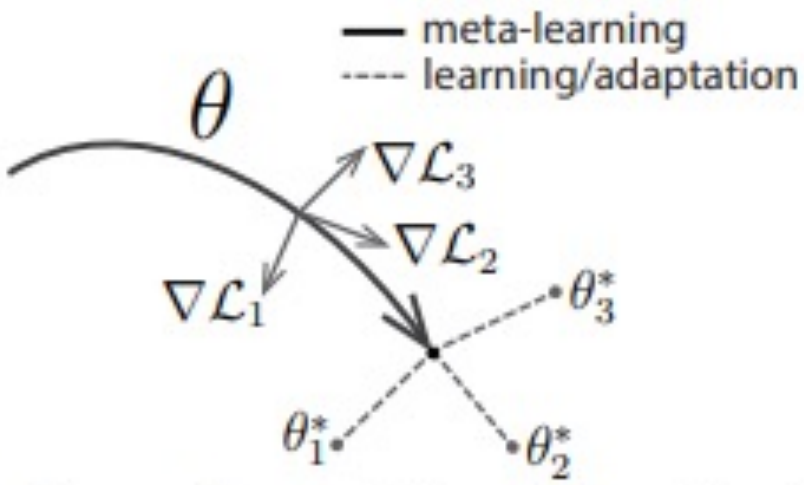
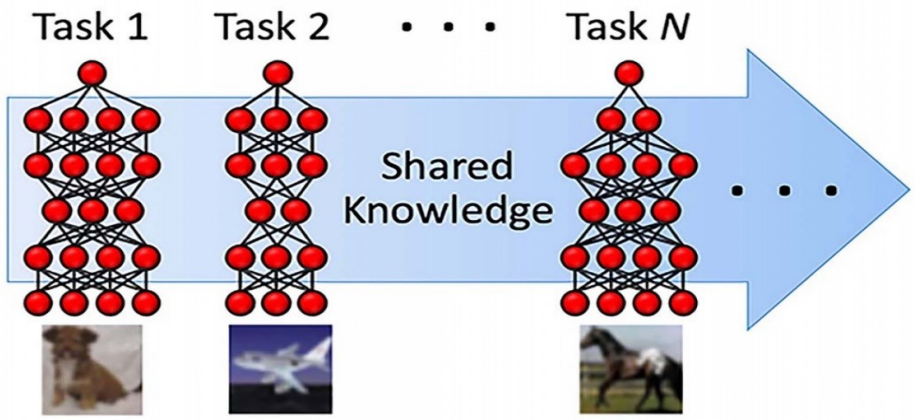


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

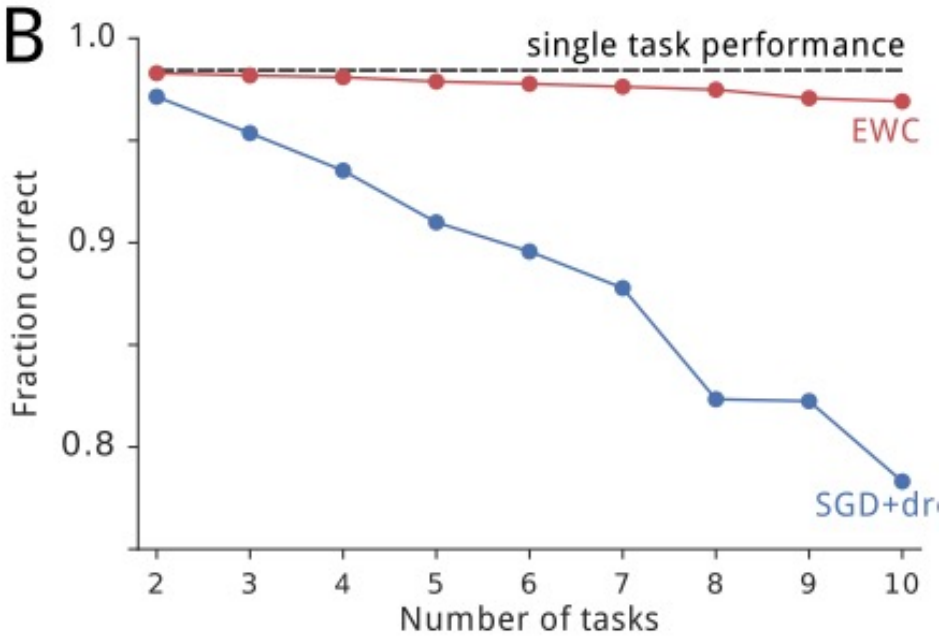
终身学习

- 终身学习（lifelong learning/continuous learning)
  - 根据历史任务中学到的经验和知识来帮助学习不断出现的新任务，并且不会忘记旧的知识。
  - 在网络容量有限时，防止学习一个新的任务会遗忘旧任务



  - 弹性权重巩固（Elastic Weight Consolidation）

终身学习


B
1.0
0.9
0.8
Fraction correct
single task performance
EWC
SGD+dr
2
3
4
5
6
7
8
9
10
Number of tasks

终身学习 vs. 归纳迁移学习和多任务学习

- 终身学习和归纳迁移学习主要区别如下：
  - 1）归纳迁移学习聚焦于优化目标任务的性能，而不关心知识的累积，终身学习目标是持续的学习和知识累计

- 终身学习和多任务学习
  - 1）终身学习不在所有任务上同时学习，而是一个一个学，多任务学习是在所有任务上同时学习

内容

- 集成学习
- 自训练和协同训练
- 多任务学习
- 迁移学习
  - 归纳迁移学习
  - 转导迁移学习
- 元学习
- 终身学习