Meta Learning & More

講解助教：B05901111　陳建成

授課教授：　李宏毅　　教授

Outline

- What is meta learning?
- Why meta learning?
- How and what to do meta learning?
    - Categories
    - Datasets
    - Models
- Related machine learning topics

What is **meta learning**?

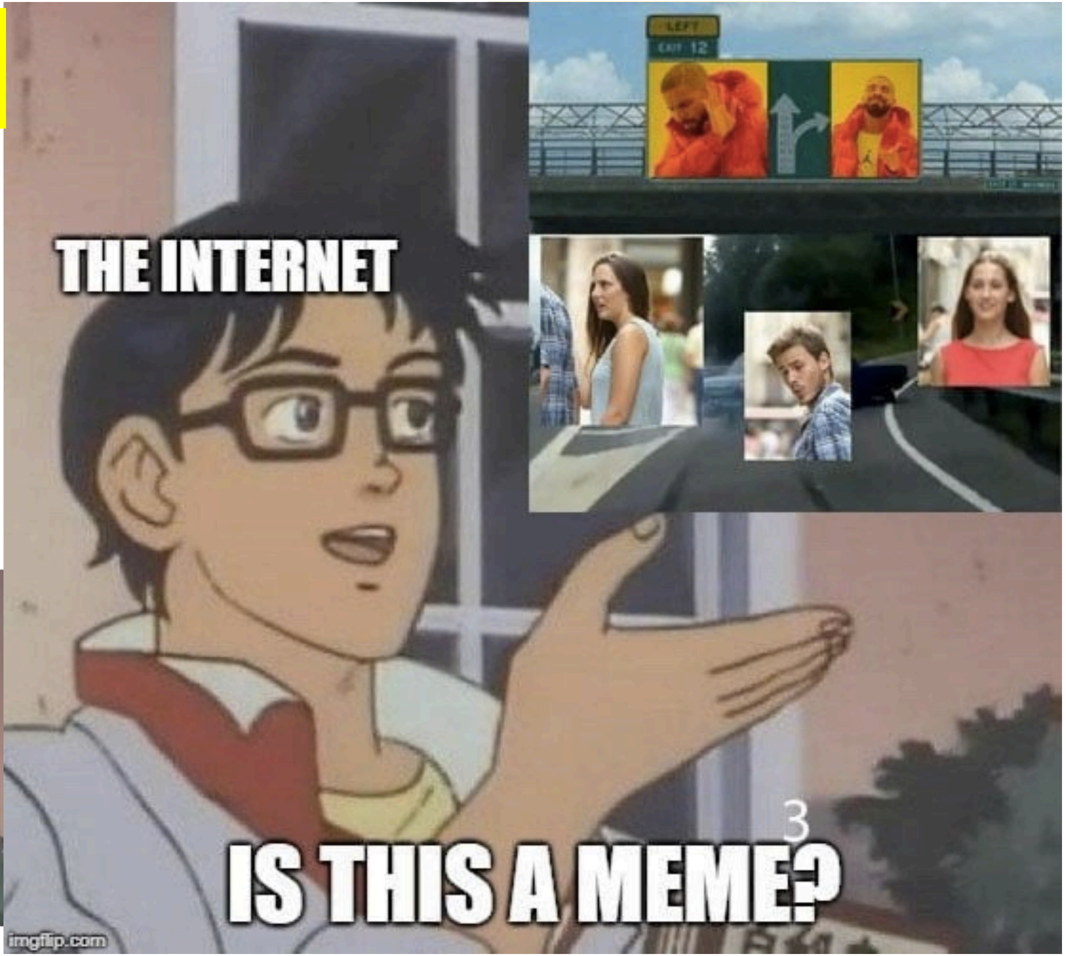
What is **meta learning**?

_______ about _______

meme

meme about meme

meta-meme →



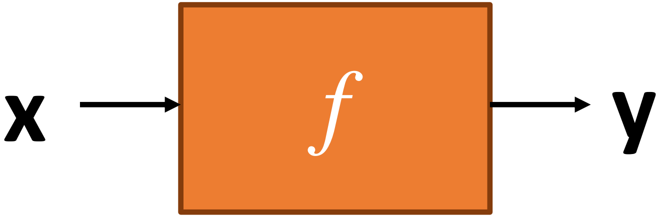
ning?

g (n.)

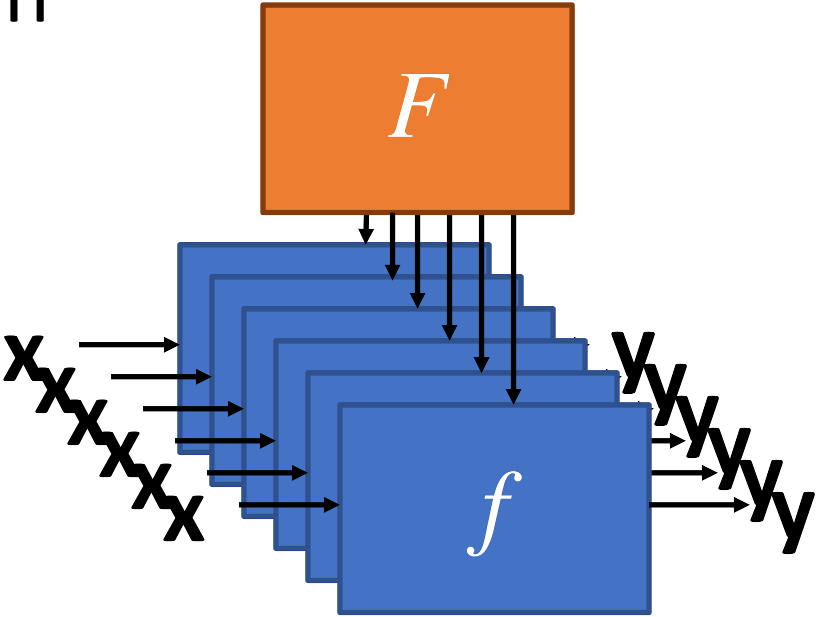
(v.)

Meta learning - Definition

- Learn to learn



learning



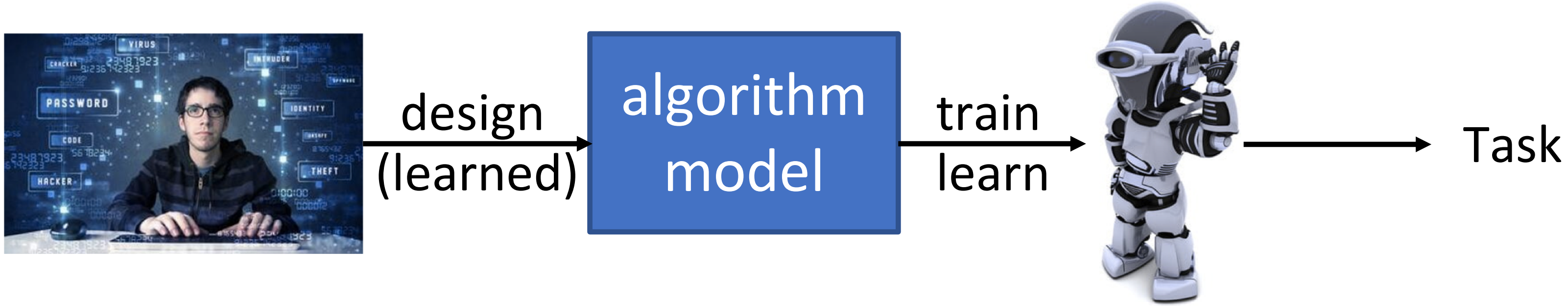
meta learning

- Usually considered to achieve **Few-shot learning** (but not limited to)

Why do we need meta learning?

Motivations for meta learning

1. Too many tasks to learn, to learn more efficiently → learning to learn



design
(learned)

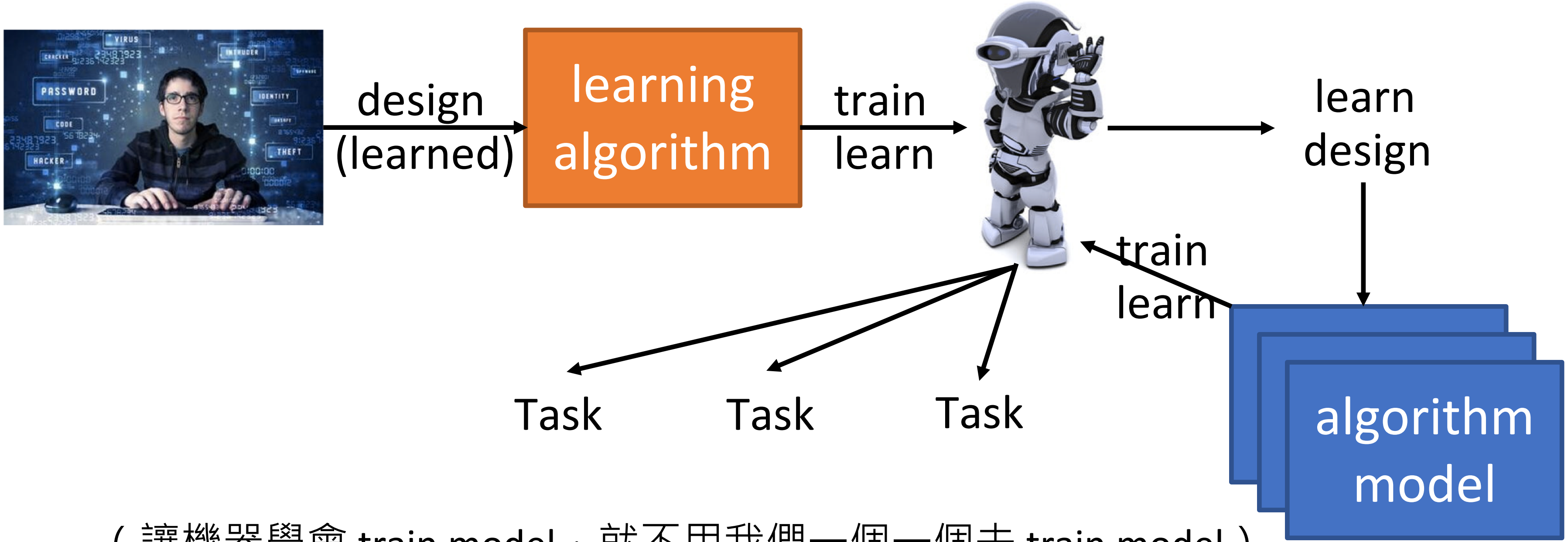
algorithm
model

train
learn

Task

Motivations for meta learning

1. Too many tasks to learn, to learn more efficiently → learning to learn



design
(learned)

learning
algorithm

train
learn

learn
design

train
learn

algorithm
model

Task

Task

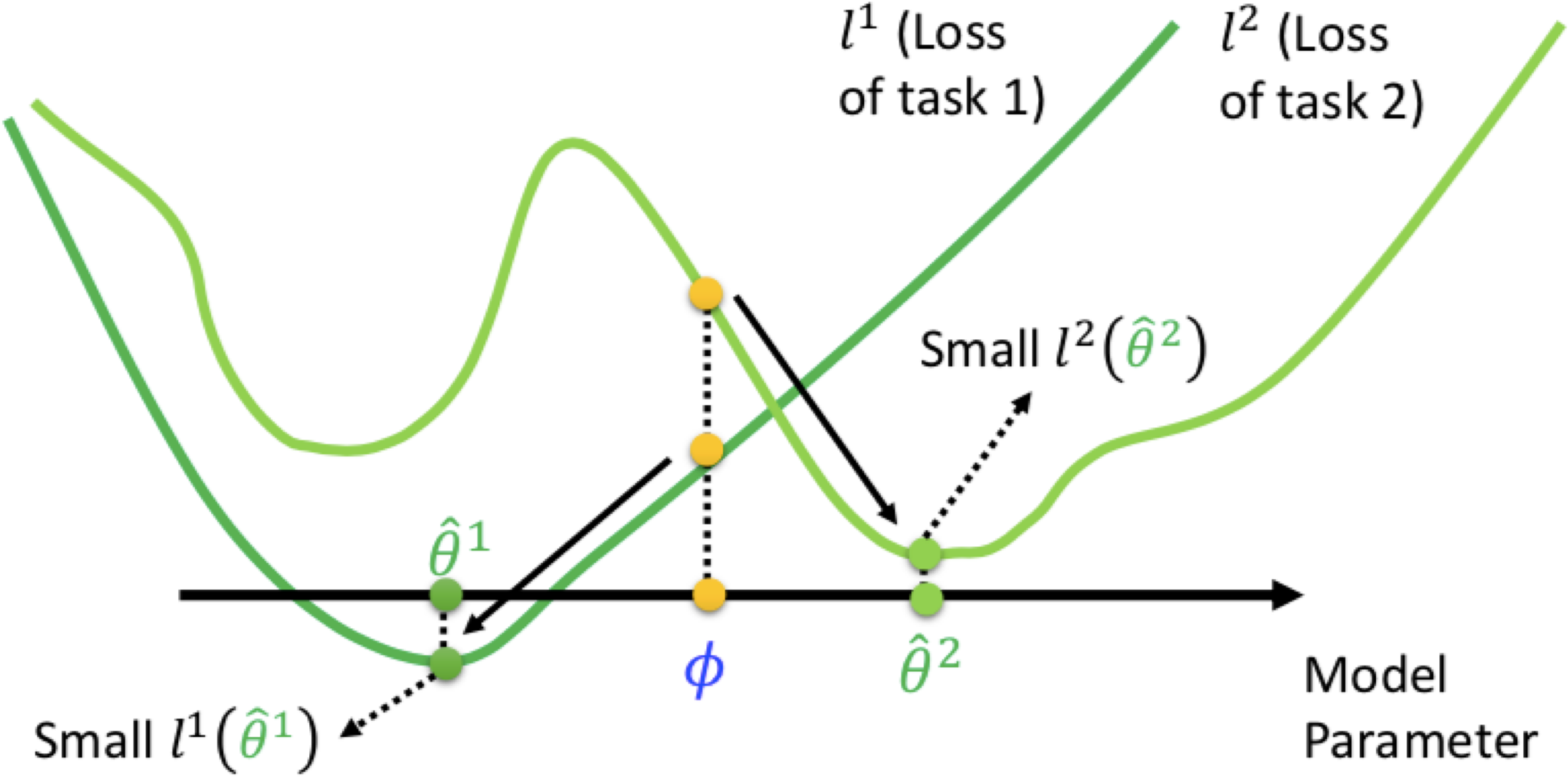
Task

（讓機器學會 train model，就不用我們一個一個去 train model）

Motivations for meta learning

1. Too many tasks to learn, to learn more efficiently → learning to learn
   - Faster learning methods (adaptation)
   - Better hyper-parameters / learning algorithms
   - Related to:
     - transfer learning
     - domain adaptation
     - multi-task learning
     - life-long learning
     - …

→ "general" AI


l^1 (Loss of task 1)
l^2 (Loss of task 2)
Small $l^2(\hat{\theta}^2)$
$\hat{\theta}^1$
ϕ
$\hat{\theta}^2$
Model Parameter
Small $l^1(\hat{\theta}^1)$

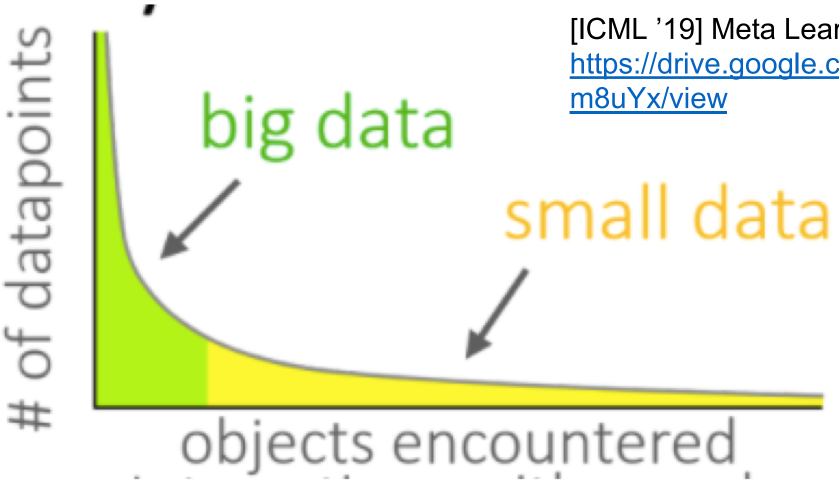
Motivations for meta learning

1. Too many tasks to learn, to learn more efficiently → learning to learn
   - Faster learning methods (adaptation)
   - Better hyper-parameters / learning algorithms
   - Related to:
     - transfer learning
     - domain adaptation
     - multi-task learning
     - life-long learning
     - …

→ "general" AI

2. Too little data, to fit more accurately → few-shot learning
   (Better learner, fit more quickly)
   - Traditional supervised may not work

Motivations for meta learning

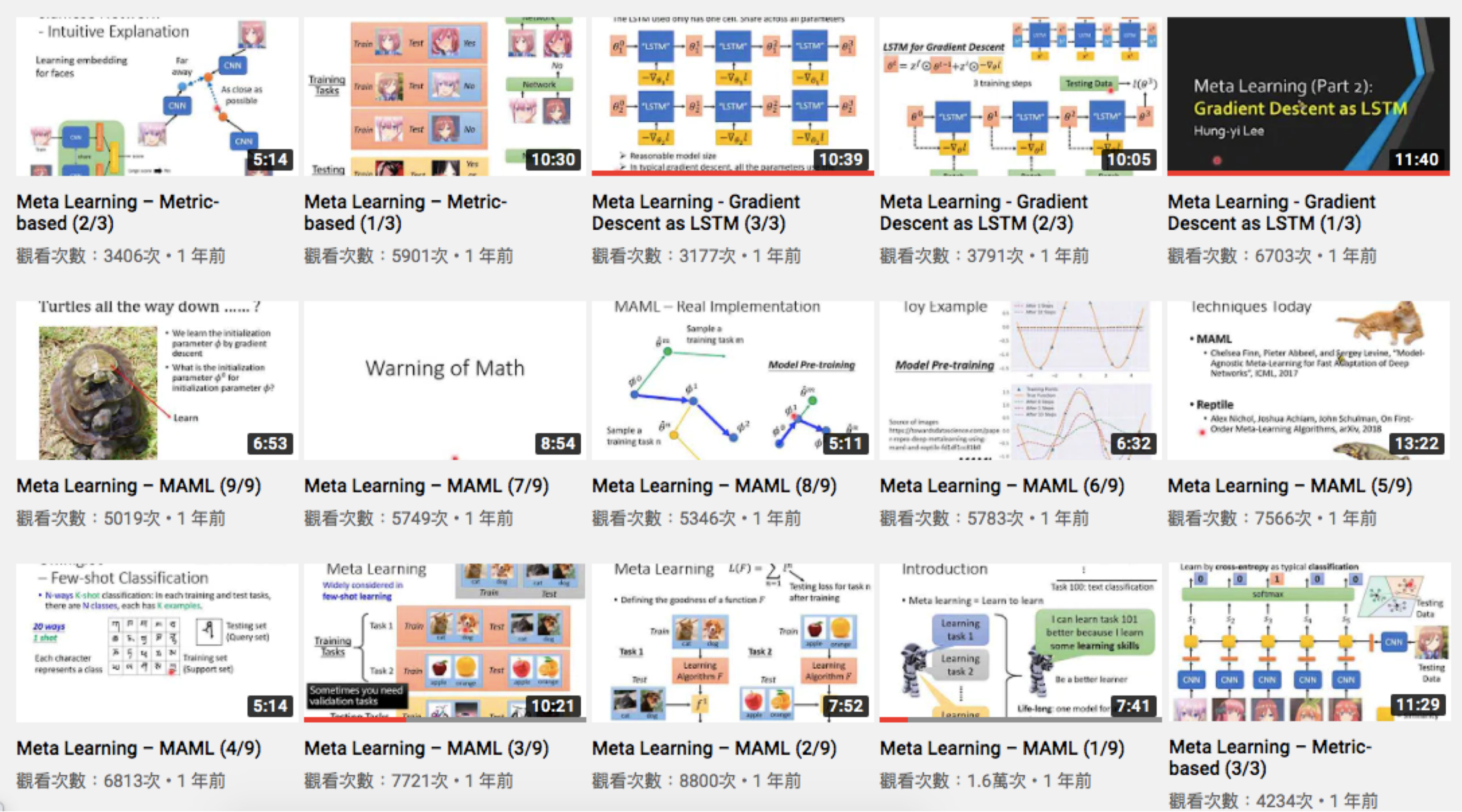


[ICML ’19] Meta Learning Tutorial
https://drive.google.com/file/d/1DuHyotdwEAEhmuHQWwRosdiVBVGm8uYx/view

2. Too little data, to fit more accurately → few-shot learning
   (Better learner, fit more quickly)
   - ○ Traditional supervised may not work

（即使資料不多，或許也 train 得起來）

How to do meta learning?


- Intuitive Explanation
Learning embedding for faces
Far away
As close as possible
CNN
Training Tasks
Train
Test
Yes
No
Network
Testing
LSTM
Reasonable model size
LSTM for Gradient Descent
3 training steps
Testing Data
Meta Learning (Part 2):
Gradient Descent as LSTM
Hung-yi Lee
Turtles all the way down …… ?
We learn the initialization parameter ϕ by gradient descent
What is the initialization parameter ϕ^0 for initialization parameter ϕ?
Learn
Warning of Math
MAML – Real Implementation
Sample a training task m
Model Pre-training
Sample a training task n
Toy Example
Model Pre-training
Techniques Today
MAML
Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", ICML, 2017
Reptile
Alex Nichol, Joshua Achiam, John Schulman, On First-Order Meta-Learning Algorithms, arXiv, 2018
– Few-shot Classification
N-ways K-shot classification: In each training and test tasks, there are N classes, each has K examples.
20 ways
1 shot
Each character represents a class
Testing set (Query set)
Training set (Support set)
Meta Learning
Widely considered in few-shot learning
Task 1
Task 2
Sometimes you need validation tasks
Defining the goodness of a function F
Testing loss for task n after training
Learning Algorithm F
Introduction
Meta learning = Learn to learn
Learning task 1
Learning task 2
I can learn task 101 better because I learn some learning skills
Be a better learner
Life-long: one model for
Task 100: text classification
Learn by cross-entropy as typical classification
softmax
CNN
Testing Data

5:14
Meta Learning – Metric-based (2/3)
觀看次數：3406次 • 1 年前

10:30
Meta Learning – Metric-based (1/3)
觀看次數：5901次 • 1 年前

10:39
Meta Learning - Gradient Descent as LSTM (3/3)
觀看次數：3177次 • 1 年前

10:05
Meta Learning - Gradient Descent as LSTM (2/3)
觀看次數：3791次 • 1 年前

11:40
Meta Learning - Gradient Descent as LSTM (1/3)
觀看次數：6703次 • 1 年前

6:53
Meta Learning – MAML (9/9)
觀看次數：5019次 • 1 年前

8:54
Meta Learning – MAML (7/9)
觀看次數：5749次 • 1 年前

5:11
Meta Learning – MAML (8/9)
觀看次數：5346次 • 1 年前

6:32
Meta Learning – MAML (6/9)
觀看次數：5783次 • 1 年前

13:22
Meta Learning – MAML (5/9)
觀看次數：7566次 • 1 年前

5:14
Meta Learning – MAML (4/9)
觀看次數：6813次 • 1 年前

10:21
Meta Learning – MAML (3/9)
觀看次數：7721次 • 1 年前

7:52
Meta Learning – MAML (2/9)
觀看次數：8800次 • 1 年前

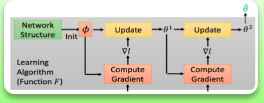
7:41
Meta Learning – MAML (1/9)
觀看次數：1.6萬次 • 1 年前

11:29
Meta Learning – Metric-based (3/3)
觀看次數：4234次 • 1 年前

Recap

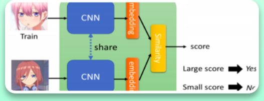
Optimization based

MAML
Reptile



Meta as LSTM

MetaLSTM



Metric based

Siamese Network
Matching Network
Prototypical Network
Relation Network

Meta as RNN

Backpropagated Signal
Predicted Label
Bind and Encode
Retrieve Bound Information
MANN: Memory Augmented Neural Network
https://arxiv.org/abs/1605.06065
SNAIL: A Simple Neural
7:47

Meta Learning - Train+Test as RNN

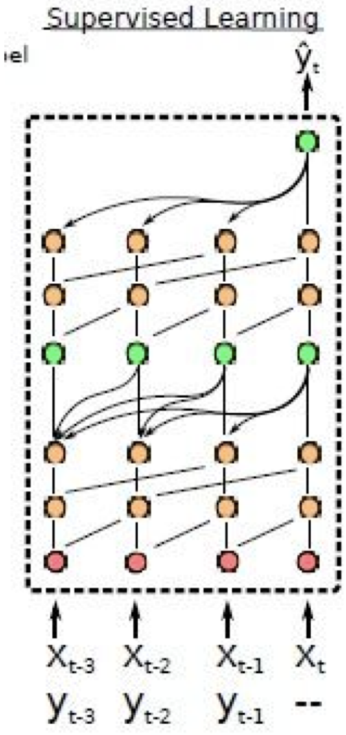
觀看次數：3909次・1 年前

MANN
SNAIL

(Interesting ***Names*** of) Models / Techniques

- MAML (Model Agnostic Meta-Learning)
- Reptile (???)
- SNAIL (Simple Neural AttentIve Learner)

mammal


Supervised Learning
el
$\hat{y}_t$
x_{t-3} x_{t-2} x_{t-1} x_t
y_{t-3} y_{t-2} y_{t-1} --

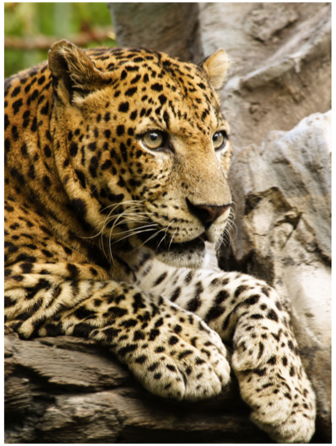
(Interesting ***Names*** of) Models / Techniques

鴨嘴獸（卵生哺乳類）

- PLATIPUS (Probabilistic LATent model for Incorporating Priors and Uncertainty in few-Shot learning)
- 駱馬 LLAMA (Lightweight Laplace Approximation for Meta-Adaptation)
- 羊駝 ALPaCA (Adaptive Learning for Probabilistic Connectionist Architectures)
- 駱駝 CAML (Conditional class-Aware Meta Learning)

(Interesting ***Names*** of) Models / Techniques

- LEO (Latent Embedding Optimization) （拉丁文）獅子；獅子座
- LEOPARD 豹
  (Learning to generate softmax parameters for diverse classification)
- CAVIA (Context Adaptation via meta-learning) （新拉丁文）豚鼠（屬）、天竺鼠
  (not CAML)
- R2-D2 (Ridge Regression Differentiable Discriminator) 星際大戰機器人



What can we “meta learn”?

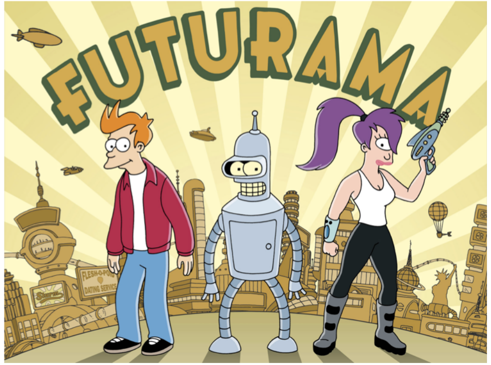
1. Model Parameters (suitable for Few-shot framework)
    - Initializations
    - Embeddings / Representations / Metrics
    - Optimizers
    - Reinforcement learning (Policies / other settings)
2. Hyperparameters (e.g. AutoML)
   (beyond the scope of today, but can be viewed as *kind of* meta learning)
    - Hyperparameters search ((training) settings)
      https://www.youtube.com/watch?v=kyX29rUntjM
    - Network architectures → Network architecture search (NAS)
      (related to: evolutional strategy, genetic algorithm…)
3. Others
    - Algorithm itself (literally, not a network)

…… (More in DLHLP)

What can we meta learn *on*? → Datasets

omni = all, glot = language

1. Omniglot
   - ○ Launched by linguist Simon Ager in 1998
   - ○ As a dataset by Lake in 2015, *Science*
   - ○ Concept learning

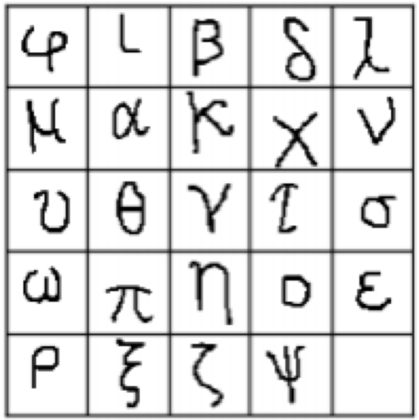


φ ι β δ λ...

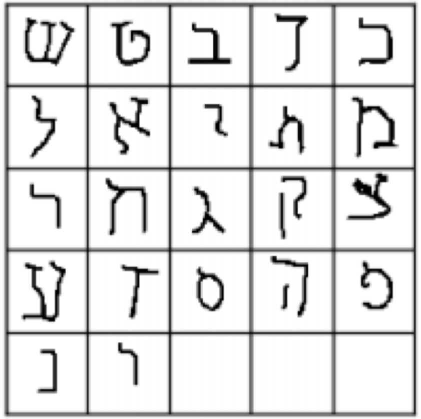
כ ל ב ט ש...

ড ন ল র কি...

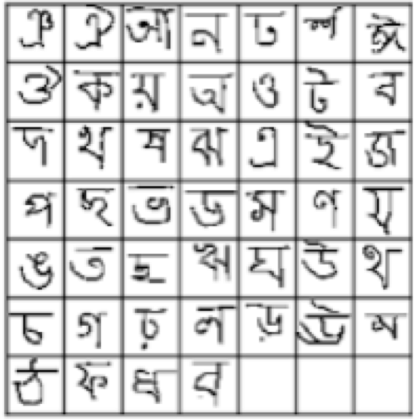
Greek



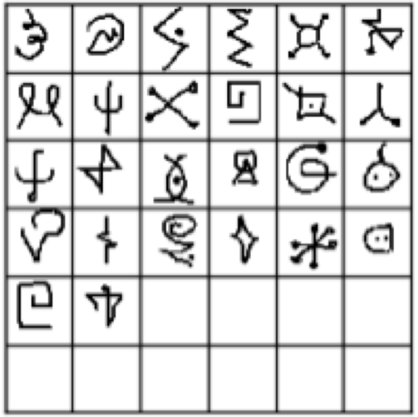
Hebrew



Bengali

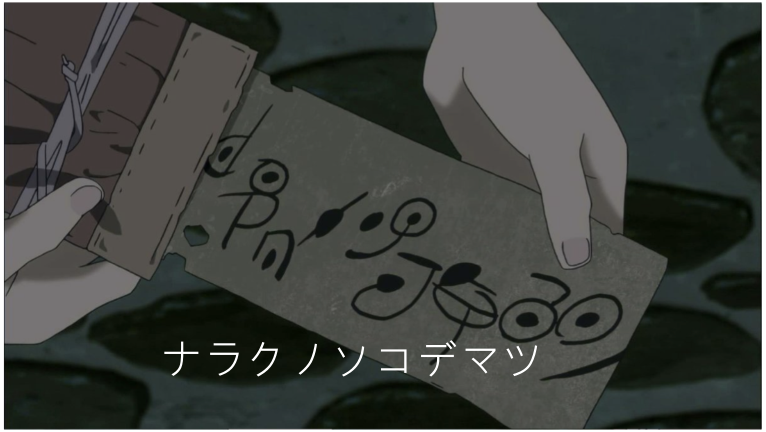


Futurama



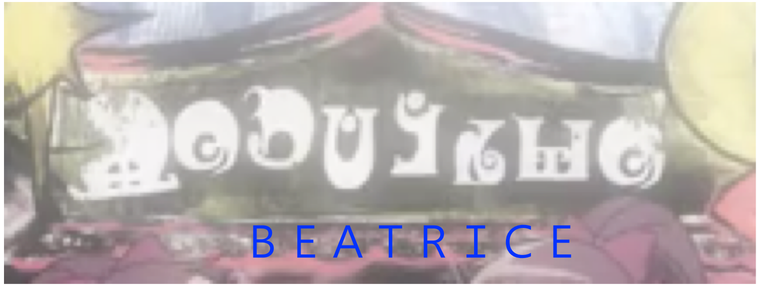
How about … “omniglot” of anime? ☺

來自深淵



https://twitter.com/tka24/status/888380568043274240?lang=ar

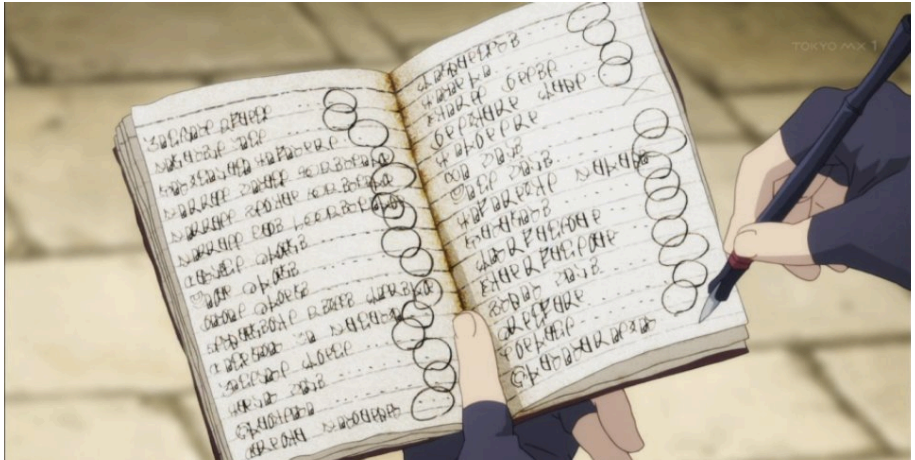
魔法少女小圓



https://ani.gamer.com.tw/animeVideo.php?sn=14454

為美好的世界獻上祝福！（このすば）

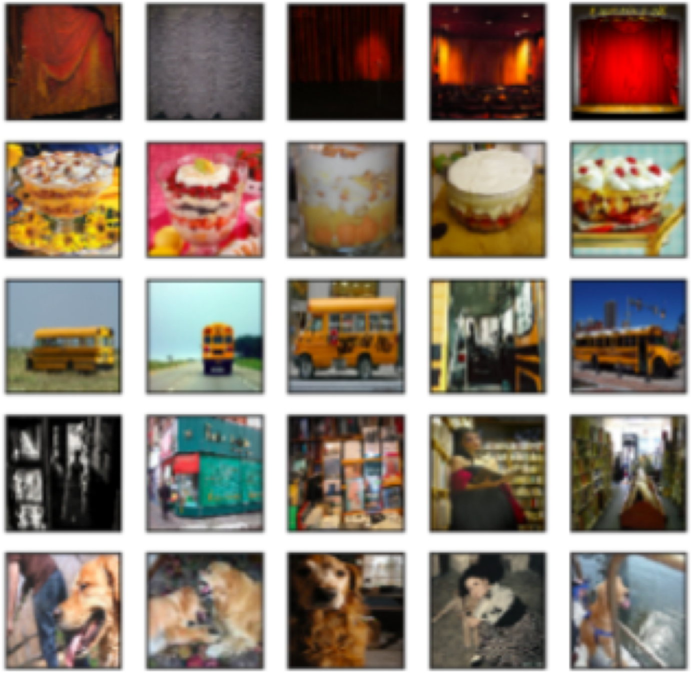
https://forum.gamer.com.tw/Co.php?bsn=60076&sn=41514705



What can we meta learn ***on***? → Datasets

2. miniImageNet

   ○ from ImageNet but few-shot

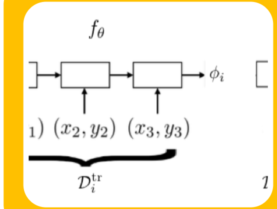


3. CUB (Caltech-UCSD Birds)

CIFAR-FS, FC100, Fungi...

Categories

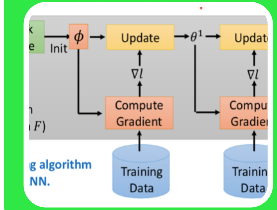
(Not unified, but generally…)



Black-box ((memory) Model) based

MANN
SNAIL
MetaNet…

Meta-LSTM

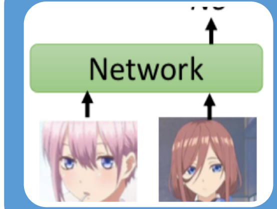


Optimization (Gradient) based (Parametric)

- Learn to initialize

MAML
Reptile…

iMAML
MAML++…

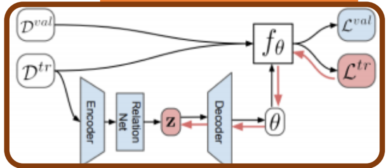


Non-parametric (Metric based)

- Learn to compare

Siamese Network
Matching Network
Prototypical Network
Relation Network…
IMP…

Hybrid



LEO…

Bayesian

PLATIPUS…

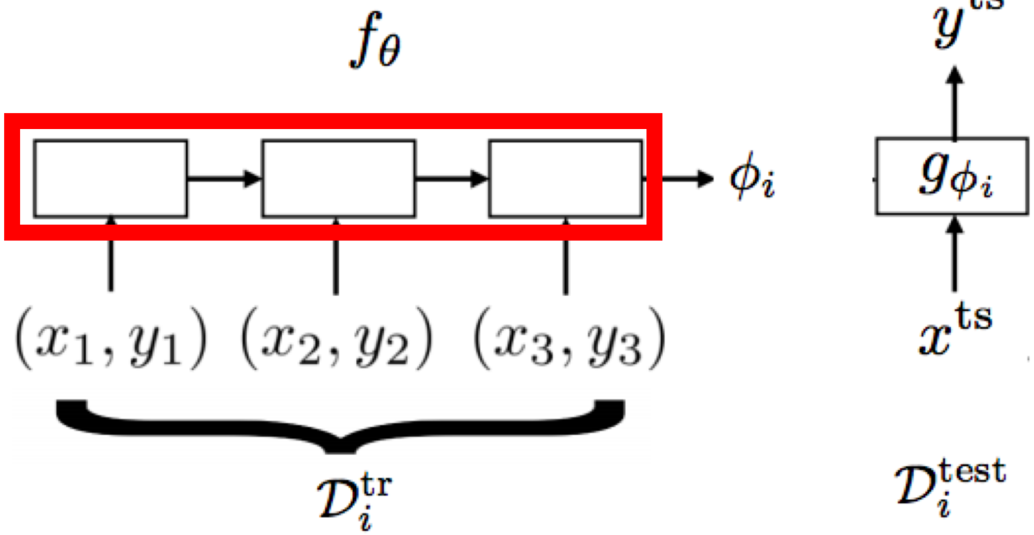
Black-box

- 直接用 RNN 架構硬 train 一發！

$$\phi_i = f_\theta(\mathcal{D}_i^{\text{tr}})$$

希望中間的 RNN 去學出 distribution

$$p(\phi_i | \mathcal{D}_i^{\text{tr}}, \theta)$$

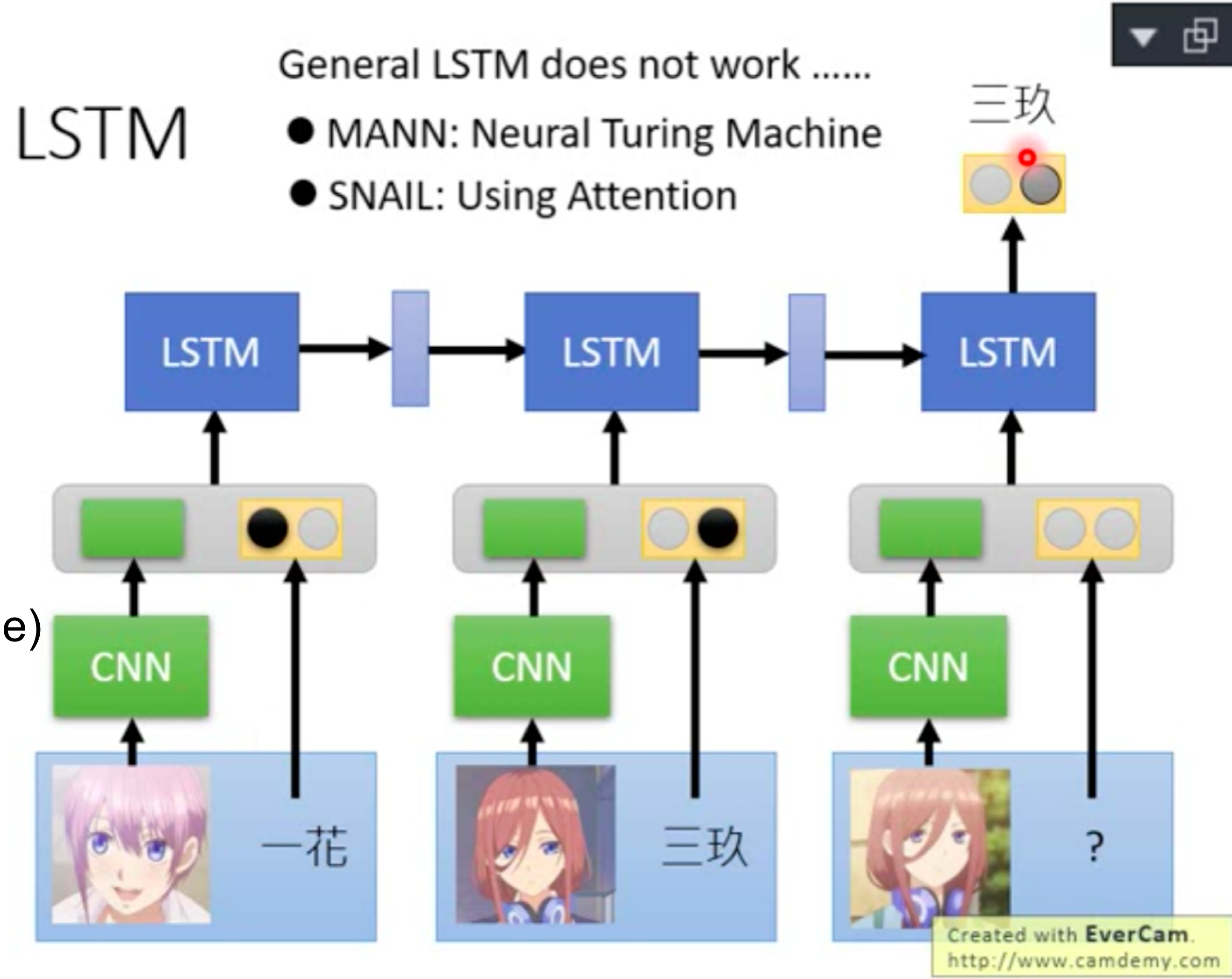


Black-box

直接用 LSTM 硬 train，
跟 “viewing as LSTM”
剛好反向操作

直接 train 就 train 不起來
→ 動一些手腳

1. 加上 memory
   (Neural Turing Machine)
   (LST“M”)
2. 加上 attention
   (Transformer)



LSTM

General LSTM does not work ......

- ● MANN: Neural Turing Machine
- ● SNAIL: Using Attention

Optimization / Gradient based

Learn model initialization

- MAML (Model Agnostic Meta Learning) *(recap)*
- Reptile
- Meta-LSTM *(can be also viewed as RNN black-box)*

different meta-parameters

- iMAML
- R2-D2 / LR-D2
- ALPaCA
- MetaOptNet

improvements of MAML

- Meta-SGD
- MAML++
- AlphaMAML
- DEML
- CAVIA

MAML

$\hat{\theta}^n$: model learned from task n

Loss Function:

$\hat{\theta}^n$ depends on ϕ

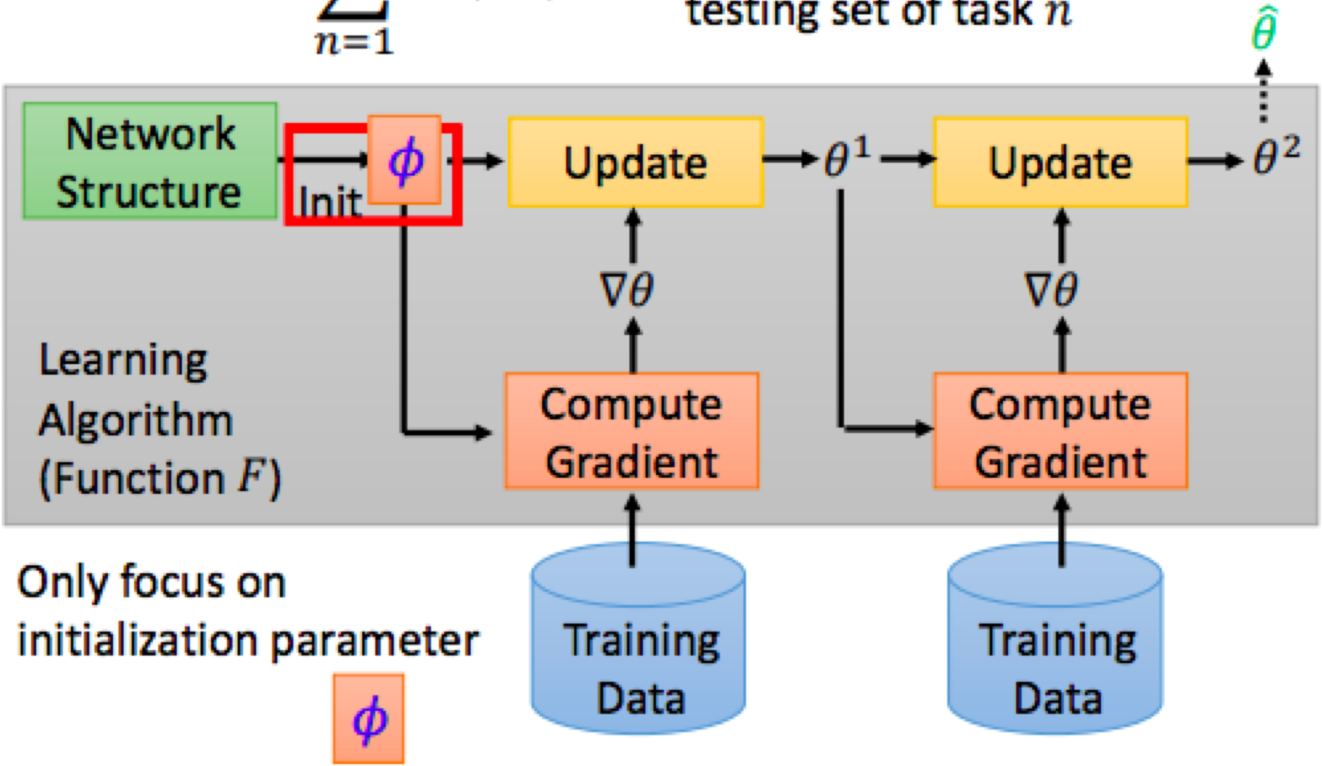
$$L(\phi) = \sum_{n=1}^{N} l^n(\hat{\theta}^n)$$

$l^n(\hat{\theta}^n)$: loss of task n on the testing set of task n

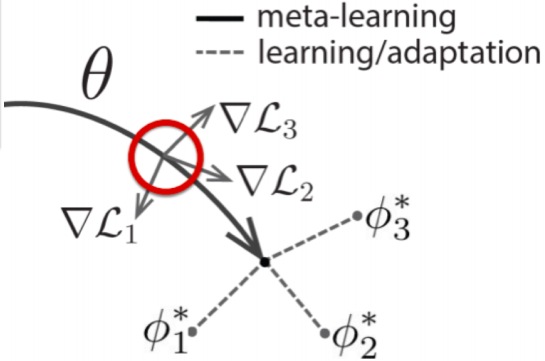
$$\phi \leftarrow \phi - \eta \nabla_\phi L(\phi)$$

Considering one-step training:

$$\hat{\theta} = \phi - \varepsilon \nabla_\phi l(\phi)$$

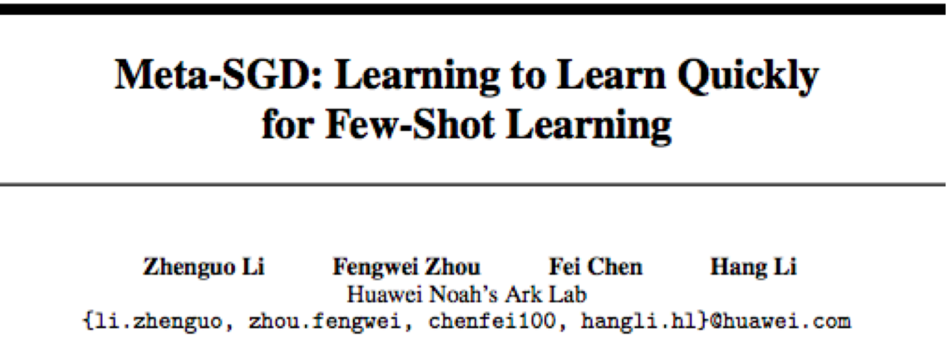


Only focus on initialization parameter ϕ



Problems of MAML

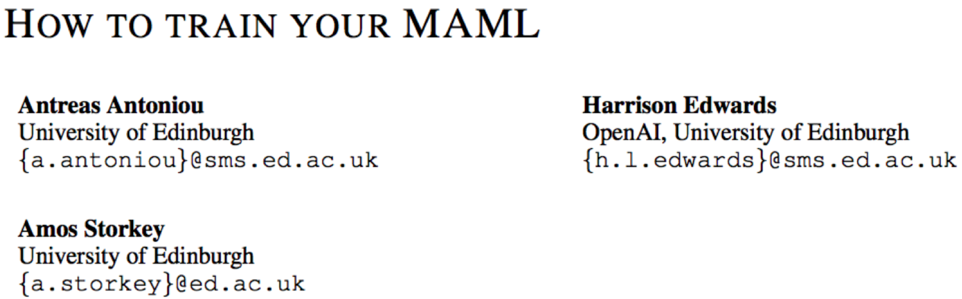
- Learning rate → Meta-SGD, MAML++
- Second-order derivatives (instability) → MAML++
- Batch Normalization → MAML++



Meta-SGD: Learning to Learn Quickly for Few-Shot Learning

Zhenguo Li **Fengwei Zhou** **Fei Chen** **Hang Li**
Huawei Noah's Ark Lab
{li.zhenguo, zhou.fengwei, chenfei100, hangli.hl}@huawei.com

CoRR, arXiv 1707.09835



HOW TO TRAIN YOUR MAML

Antreas Antoniou
University of Edinburgh
{a.antoniou}@sms.ed.ac.uk

Harrison Edwards
OpenAI, University of Edinburgh
{h.l.edwards}@sms.ed.ac.uk

Amos Storkey
University of Edinburgh
{a.storkey}@ed.ac.uk

ICLR '19

Meta-SGD

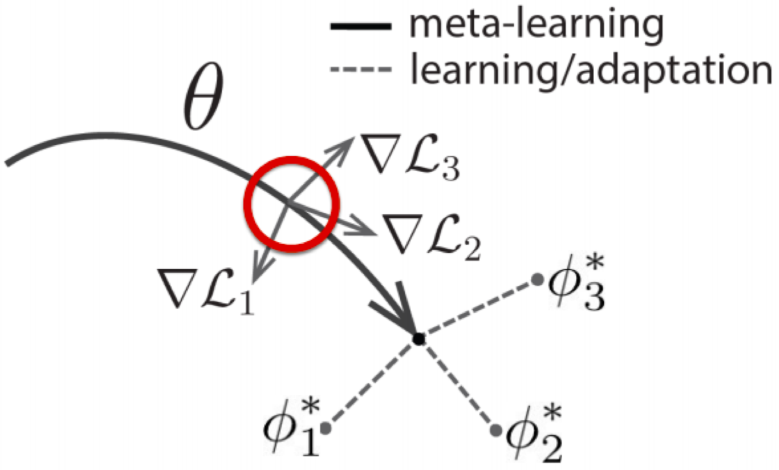
- “Adaptive learning rate” version of MAML

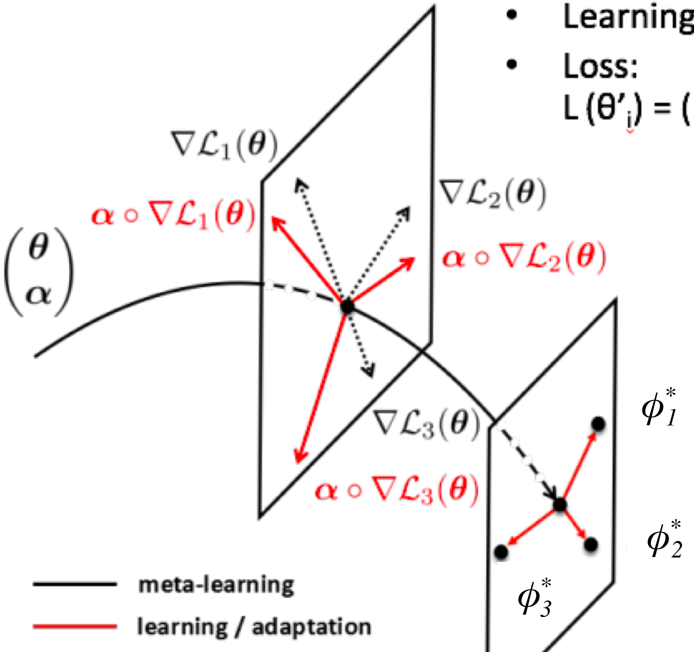
Outer loop

- Meta params: θ
- Learning rate: β
- Loss: Σ $L_{test(T\_i)}(\theta'_i)$

Inner loop

- Task params: ϕ* (θ’ original)
- Learning rate: **α** (vector)
- Loss:
  $L(\theta'_i) = (1/|data|)\ \Sigma\ l(f_\theta(x), y)$





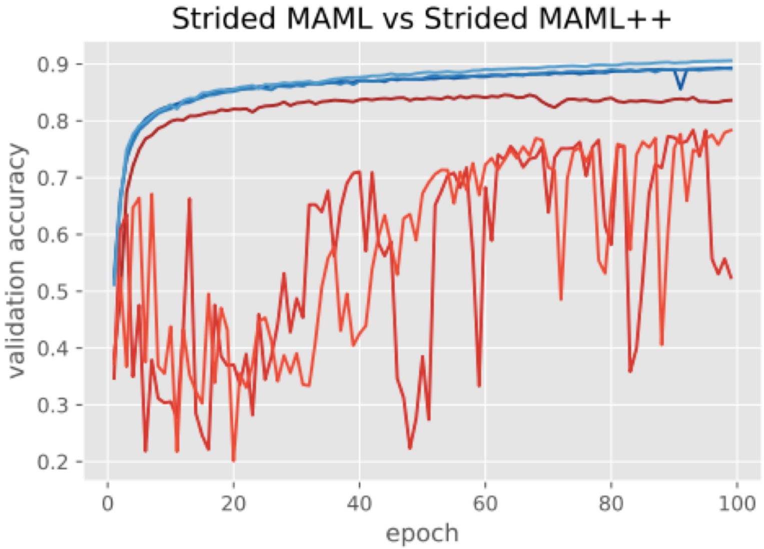
$\nabla\mathcal{L}_1(\boldsymbol{\theta})$
$\boldsymbol{\alpha}\circ\nabla\mathcal{L}_1(\boldsymbol{\theta})$
$\nabla\mathcal{L}_2(\boldsymbol{\theta})$
$\boldsymbol{\alpha}\circ\nabla\mathcal{L}_2(\boldsymbol{\theta})$
$\begin{pmatrix}\boldsymbol{\theta}\\ \boldsymbol{\alpha}\end{pmatrix}$
$\nabla\mathcal{L}_3(\boldsymbol{\theta})$
$\boldsymbol{\alpha}\circ\nabla\mathcal{L}_3(\boldsymbol{\theta})$
ϕ_1^*
ϕ_2^*
ϕ_3^*
meta-learning
learning / adaptation

$$\phi_i^* = \boldsymbol{\theta} - \boldsymbol{\alpha}\circ\nabla\mathcal{L}_i(\boldsymbol{\theta})$$

How to train your MAML?

Problems of MAML

1. Training Instability
   - Gradient issues
2. Second Order Derivative Cost
   - Expensive to compute
   - First-order → harmful to performance
3. Batch Normalization Statistics
   - No accumulation
   - Shared bias
4. Shared (across step and across parameter) inner loop learning rate
   - Not well scaled
5. Fixed outer loop learning rate



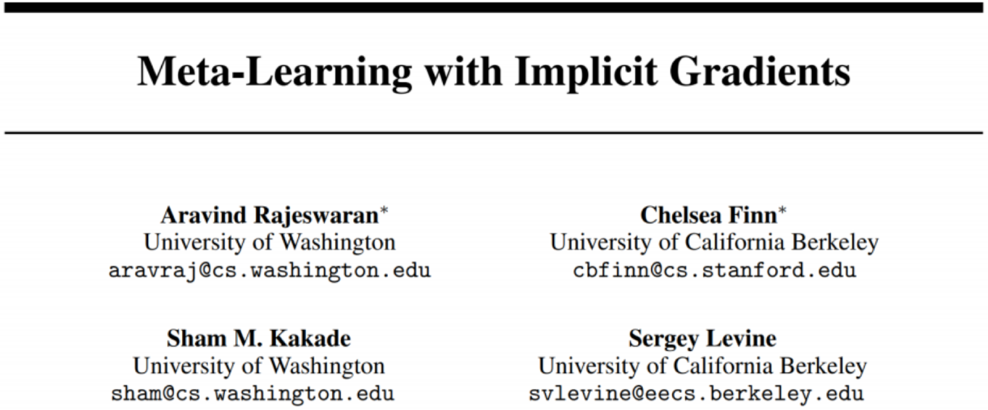
How to train your MAML?

Solutions proposed

1. Training Instability ⇒ Multi-Step Loss Optimization (MSL)
   - ○ Gradient issues
2. Second Order Derivative Cost ⇒ Derivative-Order Annealing (DA)
   - ○ Expensive to compute
   - ○ First-order → harmful to performance
3. Batch Normalization Statistics
   - ○ No accumulation ⇒ Per-Step Batch Normalization Running Statistics
   - ○ Shared bias ⇒ Per-Step Batch Normalization Weights & Biases
4. Shared (across step and across parameter) inner loop learning rate
   ⇒ Learning Per-Layer Per-Step Learning Rates & Gradient Directions (LSLR)
5. Fixed outer loop learning rate
   ⇒ Cosine Annealing of Meta-Optimizer Learning Rate (CA)

Different meta-parameters

- Implicit gradients → iMAML
- Closed-form on feature extraction → R2-D2



Meta-Learning with Implicit Gradients

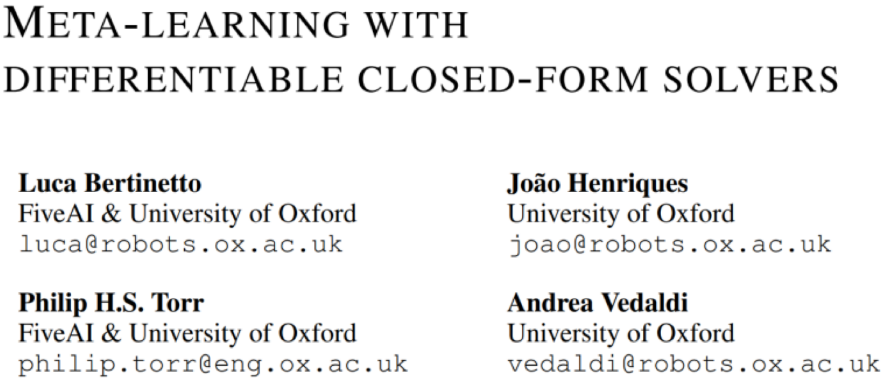
Aravind Rajeswaran*
University of Washington
aravraj@cs.washington.edu

Chelsea Finn*
University of California Berkeley
cbfinn@cs.stanford.edu

Sham M. Kakade
University of Washington
sham@cs.washington.edu

Sergey Levine
University of California Berkeley
svlevine@eecs.berkeley.edu

NIPS ‘19



META-LEARNING WITH DIFFERENTIABLE CLOSED-FORM SOLVERS

Luca Bertinetto
FiveAI & University of Oxford
luca@robots.ox.ac.uk

João Henriques
University of Oxford
joao@robots.ox.ac.uk

Philip H.S. Torr
FiveAI & University of Oxford
philip.torr@eng.ox.ac.uk

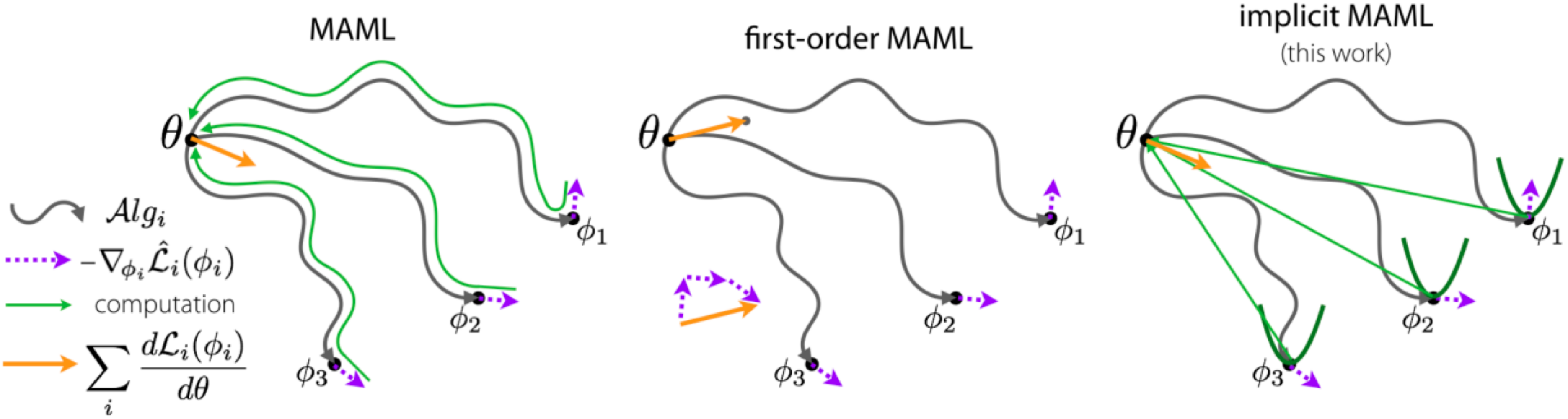
Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

ICLR ‘19

iMAML

$$\overbrace{\boldsymbol{\theta}^*_{\mathrm{ML}} := \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}}\, F(\boldsymbol{\theta})}^{\text{outer}-\text{level}}, \text{ where } F(\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} \mathcal{L}\Big(\overbrace{\mathcal{A}lg\big(\boldsymbol{\theta}, \mathcal{D}_i^{\mathrm{tr}}\big)}^{\text{inner}-\text{level}},\ \mathcal{D}_i^{\mathrm{test}}\Big).$$

$$\boldsymbol{\phi}_i \equiv \mathcal{A}lg(\boldsymbol{\theta}, \mathcal{D}_i^{\mathrm{tr}}) = \boldsymbol{\theta} - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_i^{\mathrm{tr}}). \qquad \text{(inner-level of MAML)}$$



$$\phi \leftarrow \phi - \eta \nabla_{\phi} L(\phi)$$

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

$$\boxed{\hat{\theta} = \phi - \varepsilon \nabla_{\phi} l(\phi)}$$

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

Prof. Hung-Yi Lee's

- Outer loop
  - Meta params: ϕ
  - Learning rate: η
  - Loss: L(ϕ) = Σ $l^n(\theta^n)$
- Inner loop
  - Task params: θ
  - Learning rate: ε
  - Loss: $l^n(\theta^n)$

$$\frac{\partial l(\hat{\theta})}{\partial \phi_i} = \sum_{j} \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \boxed{\frac{\partial \hat{\theta}_j}{\partial \phi_i}} \approx \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_i}$$

$$\hat{\theta}_j = \phi_j - \varepsilon \frac{\partial l(\phi)}{\partial \phi_j}$$

First-Order MAML

This is just an **approximation**!

→ Not quite accurate

$i \neq j$:

$$\frac{\partial \hat{\theta}_j}{\partial \phi_i} = -\varepsilon \cancel{\frac{\partial l(\phi)}{\partial \phi_i \partial \phi_j}} \approx 0$$

$$\frac{d\hat{\theta}}{d\phi} = \mathbf{I}$$

$i = j$:

$$\frac{\partial \hat{\theta}_j}{\partial \phi_i} = 1 - \varepsilon \cancel{\frac{\partial l(\phi)}{\partial \phi_i \partial \phi_j}} \approx 1$$

$$\nabla_{\phi} l(\hat{\theta}) = \begin{bmatrix} \partial l(\hat{\theta})/\partial \phi_1 \\ \partial l(\hat{\theta})/\partial \phi_2 \\ \vdots \\ \boxed{\partial l(\hat{\theta})/\partial \phi_i} \\ \vdots \end{bmatrix}$$

Can we do better? Consider the following…

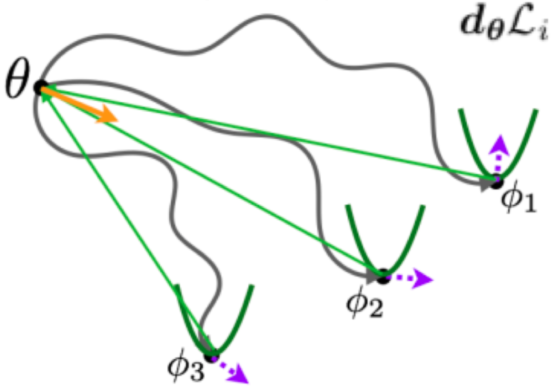
- 如果要使 meta param. 發揮作用 ⍰ 我們的 ϕ 不應該離 θ 太遠！

$$\mathcal{A}lg^\star(\boldsymbol{\theta}, \mathcal{D}_i^{\text{tr}}) = \underset{\boldsymbol{\phi}'\in\Phi}{\operatorname{argmin}}\ \mathcal{L}(\boldsymbol{\phi}', \mathcal{D}_i^{\text{tr}}) + \frac{\lambda}{2}\|\boldsymbol{\phi}' - \boldsymbol{\theta}\|^2.$$

(This is ϕ.)

implicit MAML

(this work)



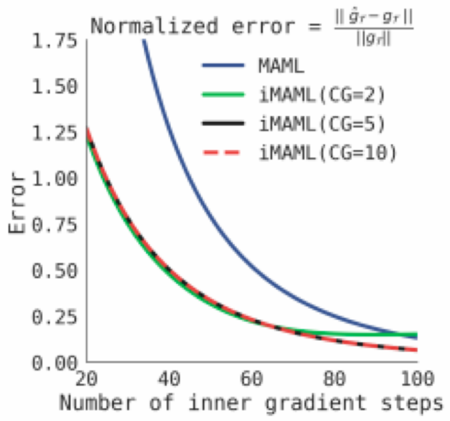
Proximal regularization!

$$d_{\boldsymbol{\theta}}\mathcal{L}_i(\mathcal{A}lg_i(\boldsymbol{\theta})) = \frac{d\mathcal{A}lg_i(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\nabla_{\boldsymbol{\phi}}\mathcal{L}_i(\boldsymbol{\phi})\mid_{\boldsymbol{\phi}=\mathcal{A}lg_i(\boldsymbol{\theta})} = \frac{d\mathcal{A}lg_i(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\nabla_{\boldsymbol{\phi}}\mathcal{L}_i(\mathcal{A}lg_i(\boldsymbol{\theta}))$$

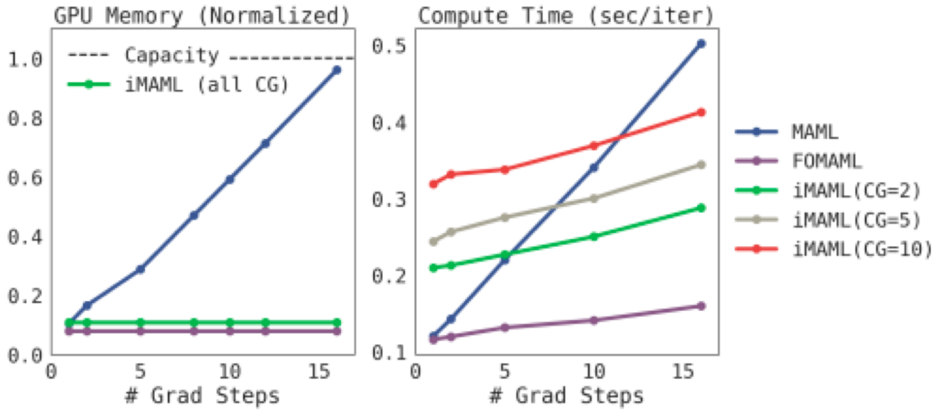
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\,\frac{1}{M}\sum_{i=1}^{M}\frac{d\mathcal{A}lg_i^\star(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\,\nabla_{\boldsymbol{\phi}}\mathcal{L}_i(\mathcal{A}lg_i^\star(\boldsymbol{\theta})).$$

$$\frac{d\mathcal{A}lg_i^\star(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \left(\boldsymbol{I} + \frac{1}{\lambda}\nabla_{\boldsymbol{\phi}}^2\hat{\mathcal{L}}_i(\boldsymbol{\phi}_i)\right)^{-1}.$$

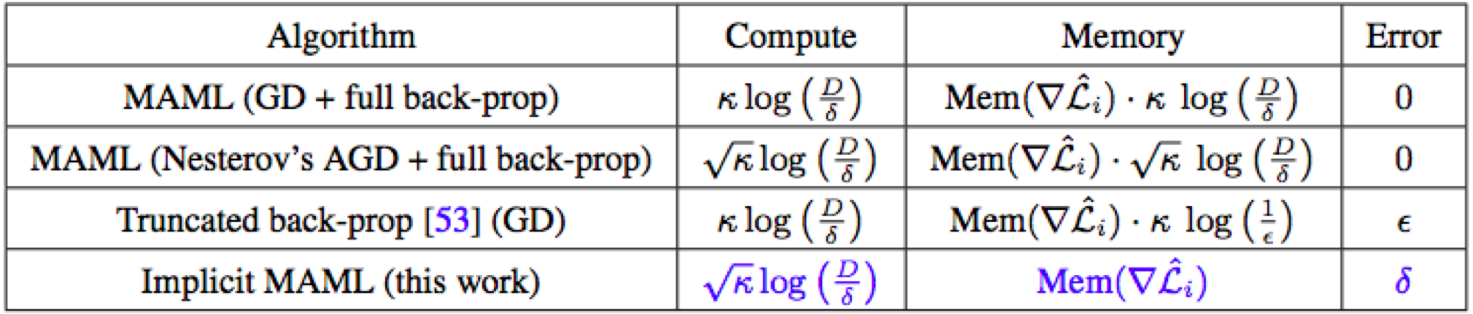
How does it perform?



(a)



(b)

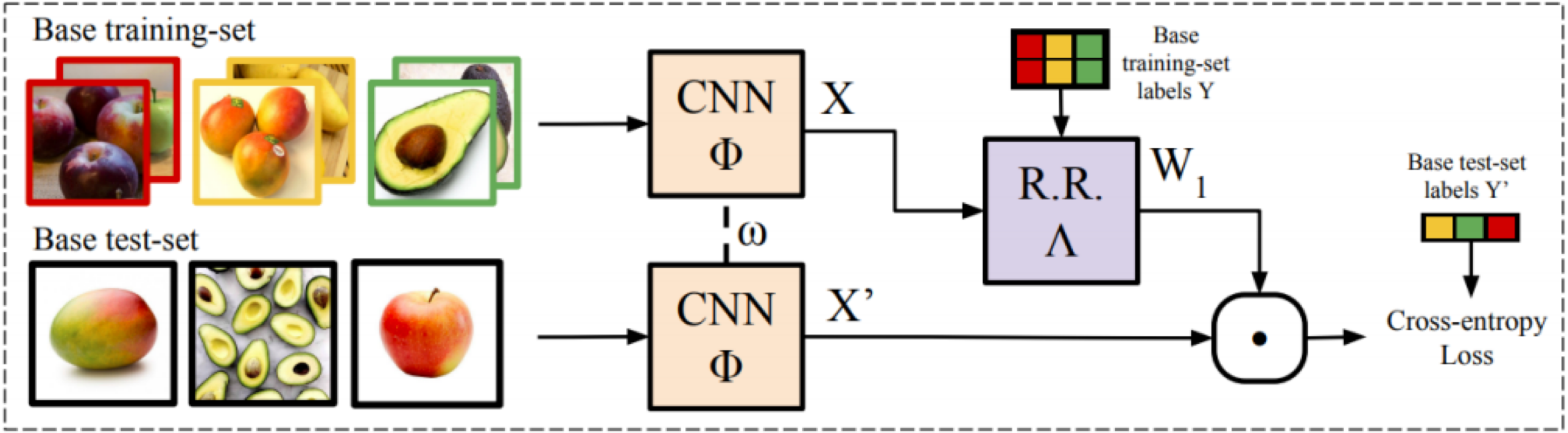


Algorithm	Compute	Memory	Error
MAML (GD + full back-prop)	$\kappa \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i) \cdot \kappa \log\left(\frac{D}{\delta}\right)$	0
MAML (Nesterov's AGD + full back-prop)	$\sqrt{\kappa} \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i) \cdot \sqrt{\kappa} \log\left(\frac{D}{\delta}\right)$	0
Truncated back-prop [53] (GD)	$\kappa \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i) \cdot \kappa \log\left(\frac{1}{\epsilon}\right)$	ϵ
Implicit MAML (this work)	$\sqrt{\kappa} \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i)$	δ

R2-D2: closed form solvers

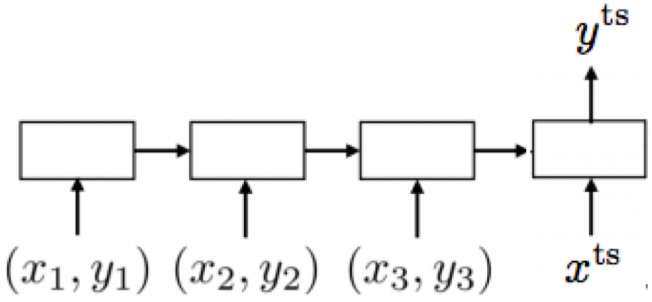
L2 regularization

將 inner loop 的 base learners 用 closed form solvers （例如 ridge regression ）取代，只訓練 feature extractors 的參數（當作 meta parameters ）



$$\min_{\omega,\rho} \frac{1}{|\mathbb{E}| \cdot |Z'_{\mathcal{E}}|} \sum_{\mathcal{E}\in\mathbb{E}} \sum_{(x',y')\in Z'_{\mathcal{E}}} L\left(f\left(\phi\left(x' ; \omega\right) ; w_{\mathcal{E}}\right), y'\right), \quad \text{with } w_{\mathcal{E}} = \Lambda(\phi(Z_{\mathcal{E}} ; \omega) ; \rho).$$

Black-box v.s Gradient based

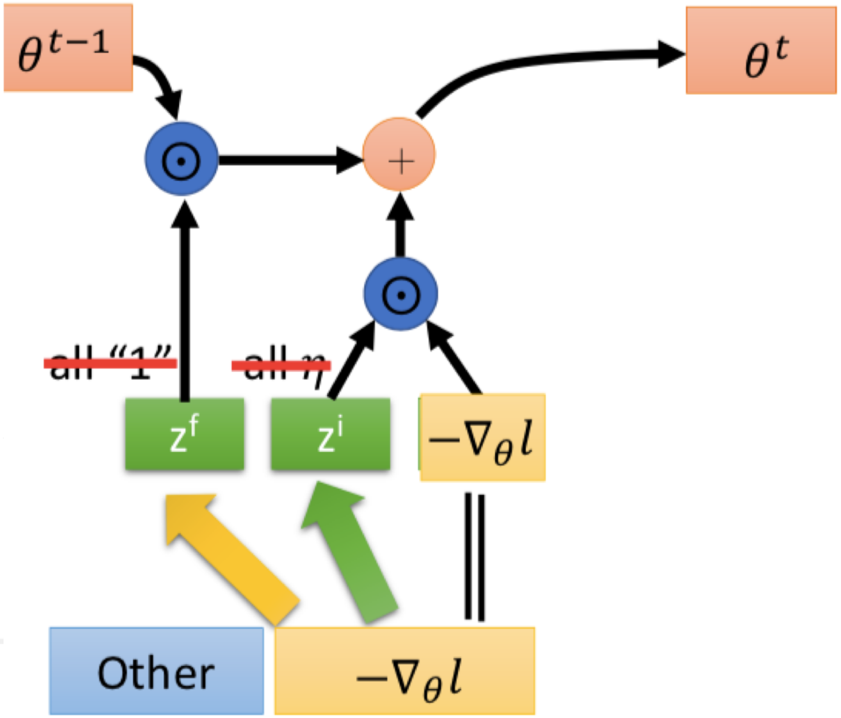


Stored in θ

$$y^{\text{ts}} = f_\theta(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}}) = f_\theta(\{(x_i, y_i)\}_{i=1}^{L}, x^{\text{ts}})$$

$$\mathcal{D}_i^{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{L}$$

$$\boxed{\begin{aligned}\phi_i = \theta - \alpha \cancel{\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})} \\ f(\theta, \mathcal{D}_i^{\text{tr}}, \nabla_\theta \mathcal{L})\end{aligned}}$$



(x^{ts})

step

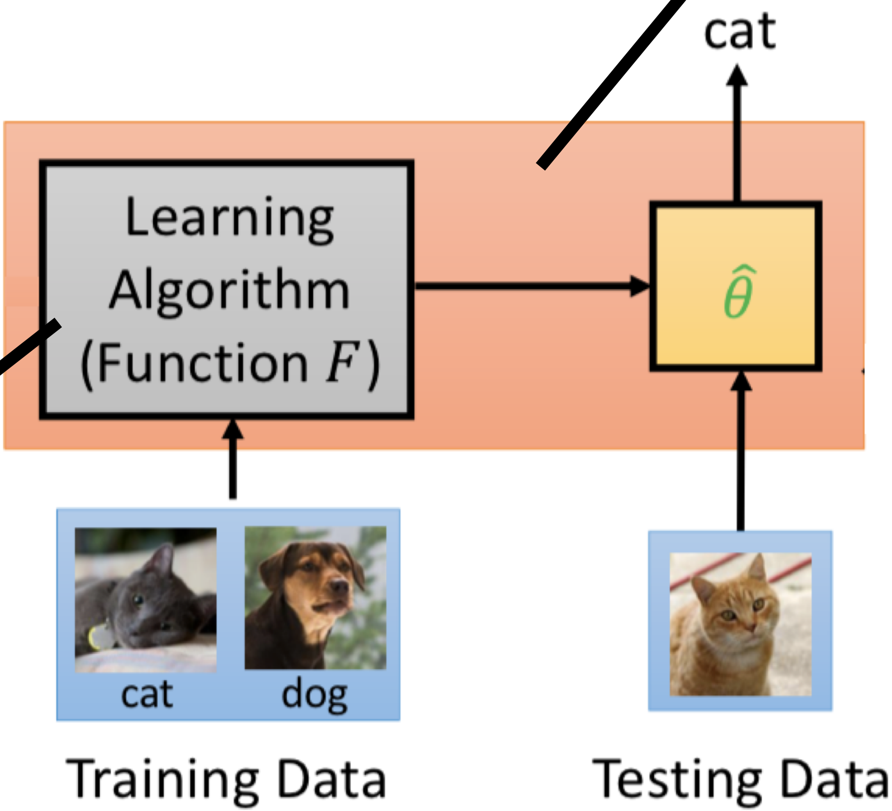
steps

→ Meta-LSTM, even more general and unifying!

Metric-based / non-parametric

A bigger function

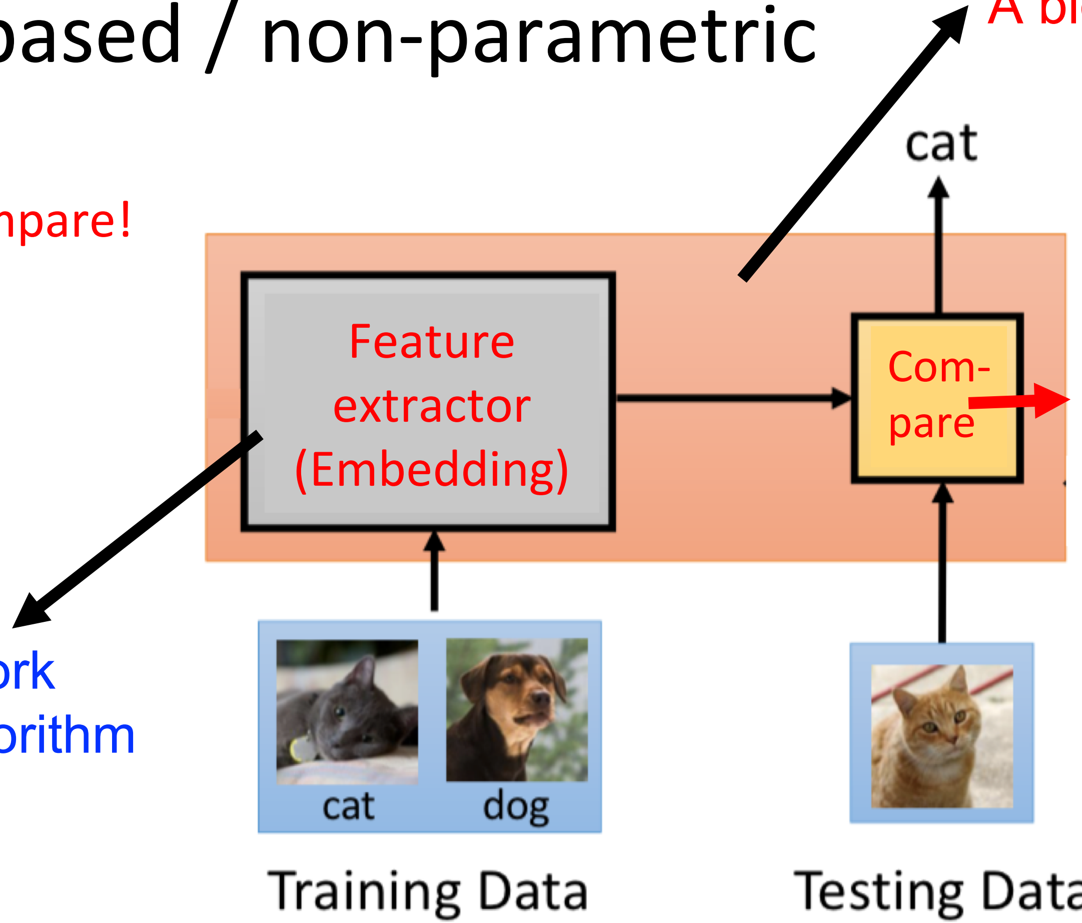
Learn to compare!



Network as an algorithm

Metric-based / non-parametric

Learn to compare!

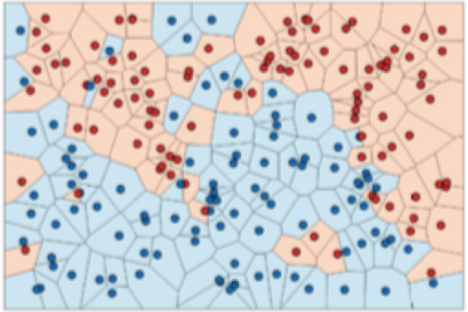


A bigger function

KNN
Cosine
L2

Network as an algorithm

Metric-based / non-parametric



Learn to compare!

No need to parametrize all the model, just the embedding / representation!

Why?

- Few-shot → Just compare! No too complex models needed!
  (Even compare with something like ~~L2-distances or pixel spaces~~)
- Too simple? → Learn from training! = learn a better “embedding”!

Metric-based / non-parametric

Learn to compare!

One reason: few-shot!

No need to parametrize all the model, just the embedding / representation!

- Siamese network
- Prototypical network
- Matching network
- Relation network

Better representation

- IMP
- GNN

Metric-based / non-parametric

Learn to compare!

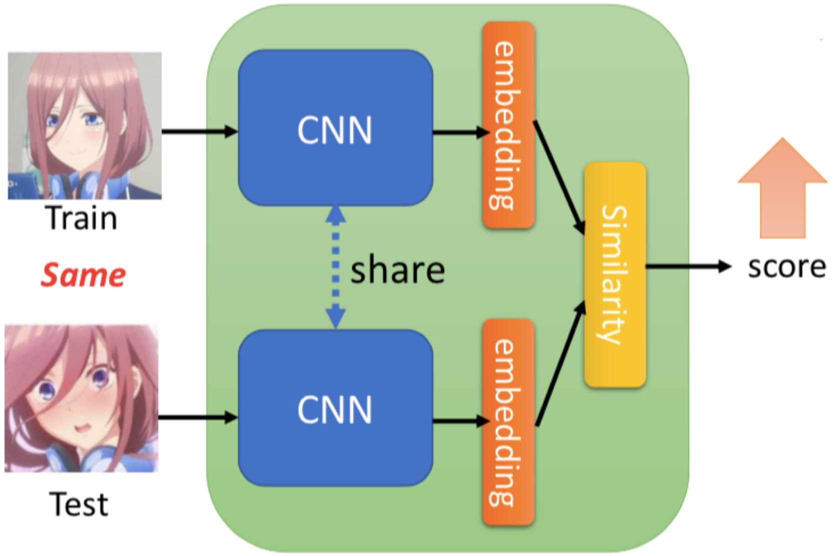
One reason: few-shot!

No need to parametrize all the model, just the embedding / representation!

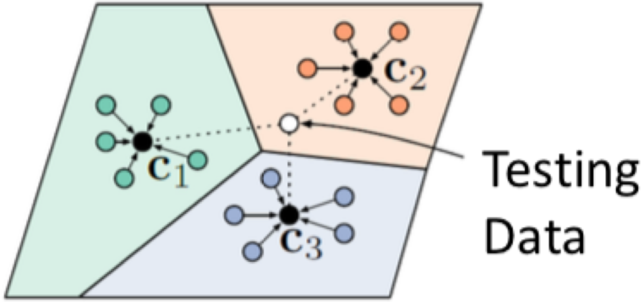
- Siamese network
- Prototypical network
- Matching network
- Relation network

Better representation

- IMP
- GNN



Metric-based / non-parametric



Learn to compare!

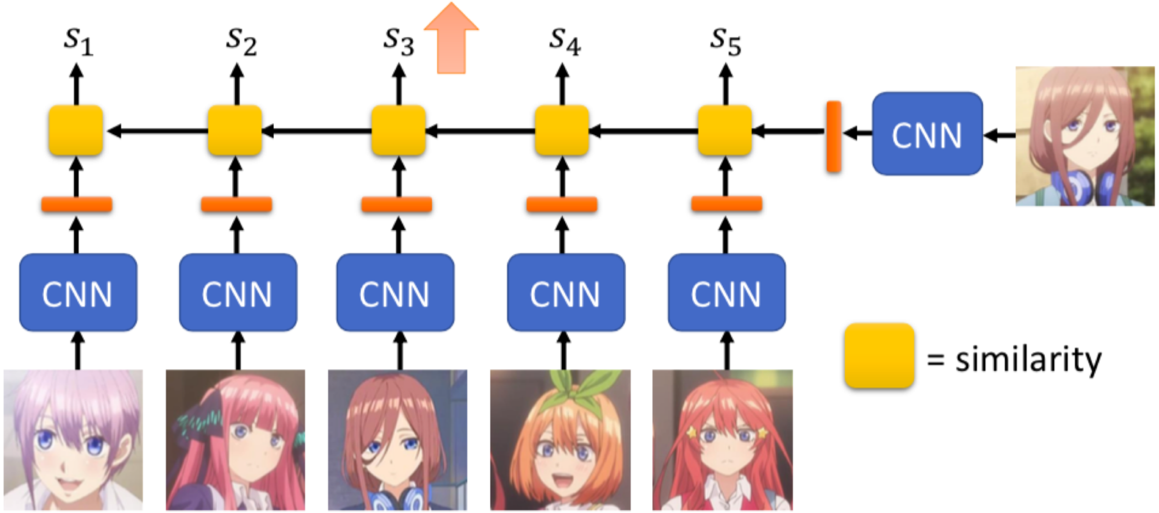
One reason: few-shot!

No need to parametrize all the model, just the embedding / representation!

- Siamese network
- **Prototypical network**
- Matching network
- Relation network

Better representation

- IMP
- GNN



Metric-based / non-parametric

Learn to compare!

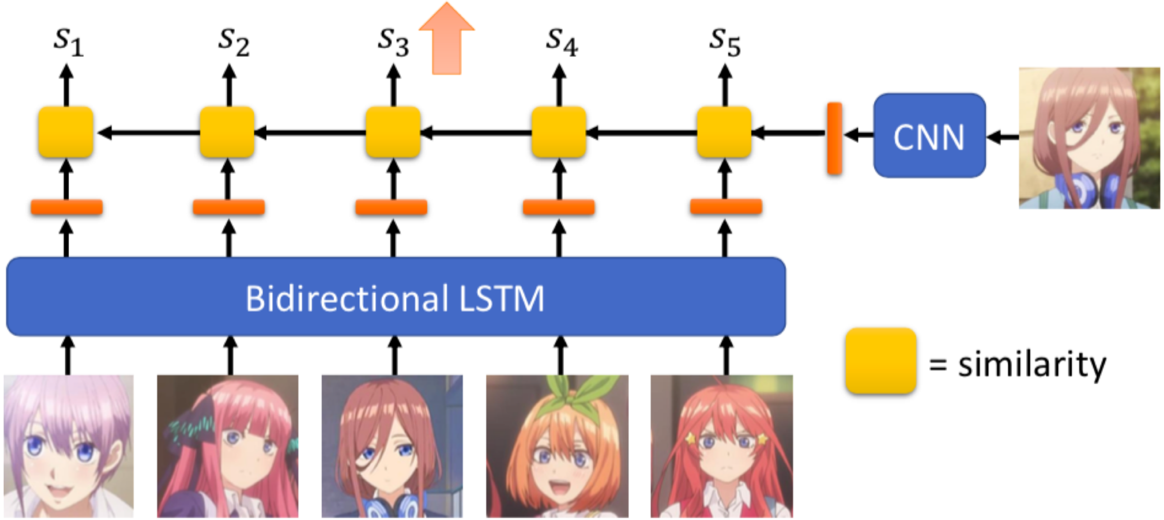
One reason: few-shot!

No need to parametrize all the model, just the embedding / representation!

- Siamese network
- Prototypical network
- **Matching network**
- Relation network

Better representation

- IMP
- GNN



Metric-based / non-parametric

Learn to compare!

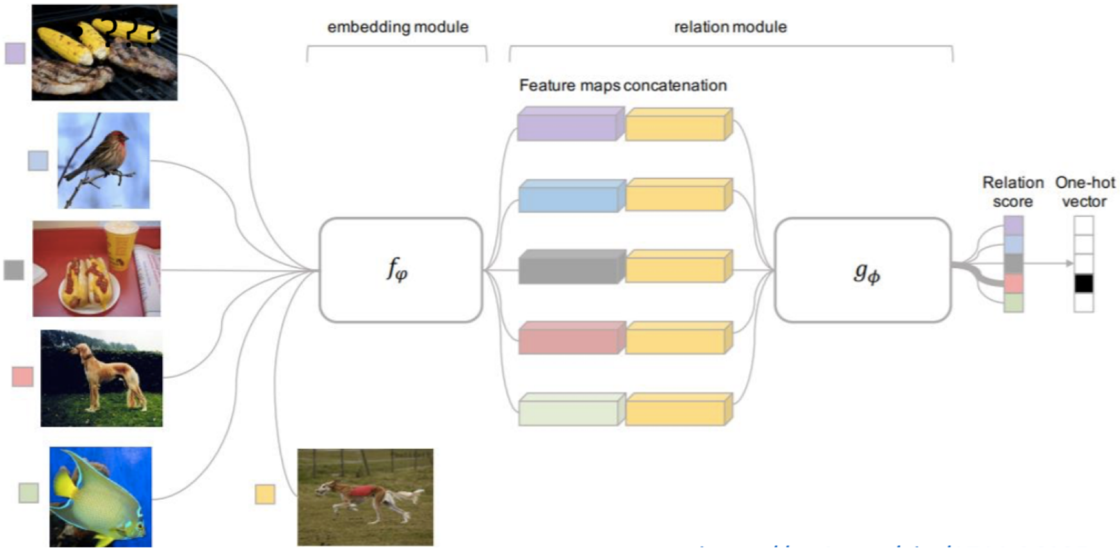
One reason: few-shot!

No need to parametrize all the model, just the embedding / representation!

- Siamese network
- Prototypical network
- Matching network
- **Relation network**

Better representation

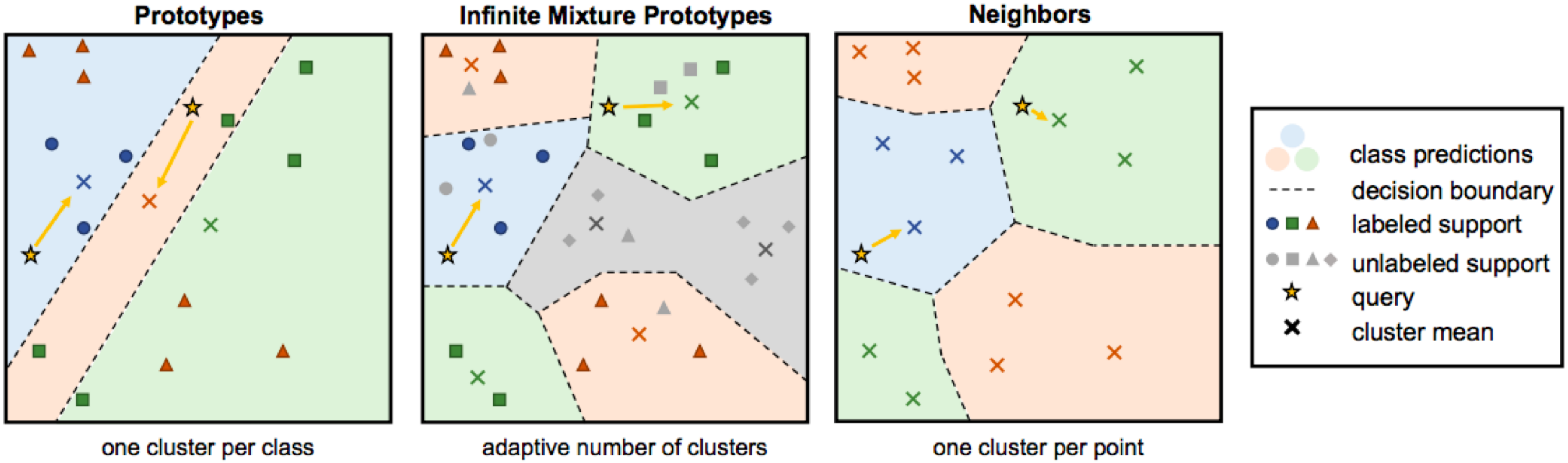
- IMP
- GNN



IMP (Infinite Mixture Prototypes)

https://arxiv.org/pdf/1902.04552.pdf

- Modified from prototypical
- The number of mixture determined from data through Bayesian nonparametric methods

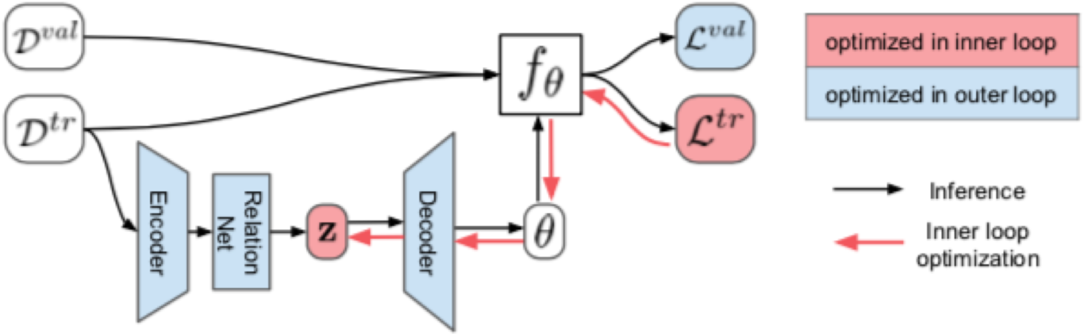


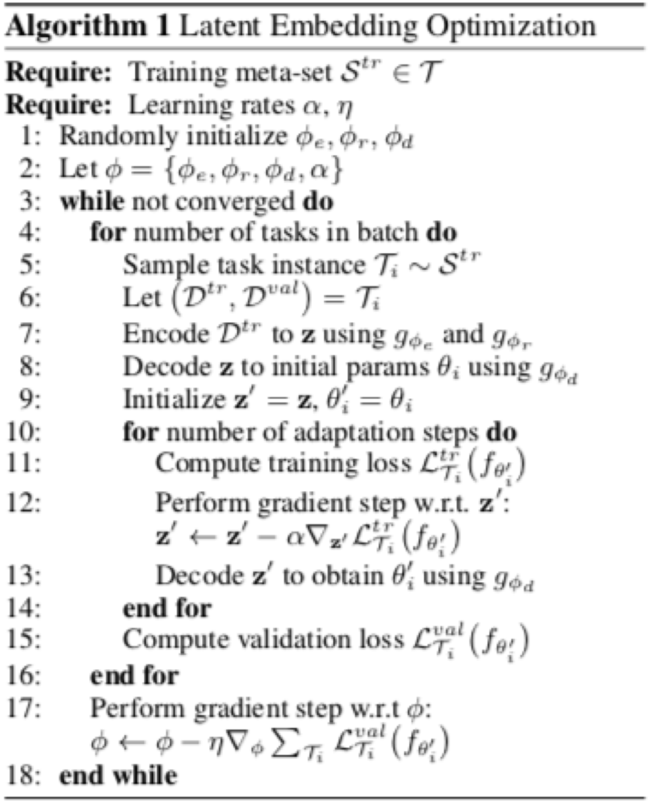
Problems of metric-based

- When the *K* in *N*-way *K*-shot large → difficult to scale
- Limited to classification (only learning to compare)

Hybrid

- CAML
- Proto-MAML
- LEO (Latent Embedding Optimization)





Algorithm 1 Latent Embedding Optimization

```
Require: Training meta-set $\mathcal{S}^{tr} \in \mathcal{T}$
Require: Learning rates $\alpha, \eta$
Randomly initialize $\phi_e, \phi_r, \phi_d$
Let $\phi = \{\phi_e, \phi_r, \phi_d, \alpha\}$
while not converged do
    for number of tasks in batch do
        Sample task instance $\mathcal{T}_i \sim \mathcal{S}^{tr}$
        Let $(\mathcal{D}^{tr}, \mathcal{D}^{val}) = \mathcal{T}_i$
        Encode $\mathcal{D}^{tr}$ to $\mathbf{z}$ using $g_{\phi_e}$ and $g_{\phi_r}$
        Decode $\mathbf{z}$ to initial params $\theta_i$ using $g_{\phi_d}$
        Initialize $\mathbf{z}' = \mathbf{z}$, $\theta'_i = \theta_i$
        for number of adaptation steps do
            Compute training loss $\mathcal{L}^{tr}_{\mathcal{T}_i}(f_{\theta'_i})$
            Perform gradient step w.r.t. $\mathbf{z}'$:
            $\mathbf{z}' \leftarrow \mathbf{z}' - \alpha \nabla_{\mathbf{z}'} \mathcal{L}^{tr}_{\mathcal{T}_i}(f_{\theta'_i})$
            Decode $\mathbf{z}'$ to obtain $\theta'_i$ using $g_{\phi_d}$
        end for
        Compute validation loss $\mathcal{L}^{val}_{\mathcal{T}_i}(f_{\theta'_i})$
    end for
    Perform gradient step w.r.t $\phi$:
    $\phi \leftarrow \phi - \eta \nabla_\phi \sum_{\mathcal{T}_i} \mathcal{L}^{val}_{\mathcal{T}_i}(f_{\theta'_i})$
end while
```

Optimization based on model + Metric based embedding (RelationNet **z**)

Bayesian meta-learning

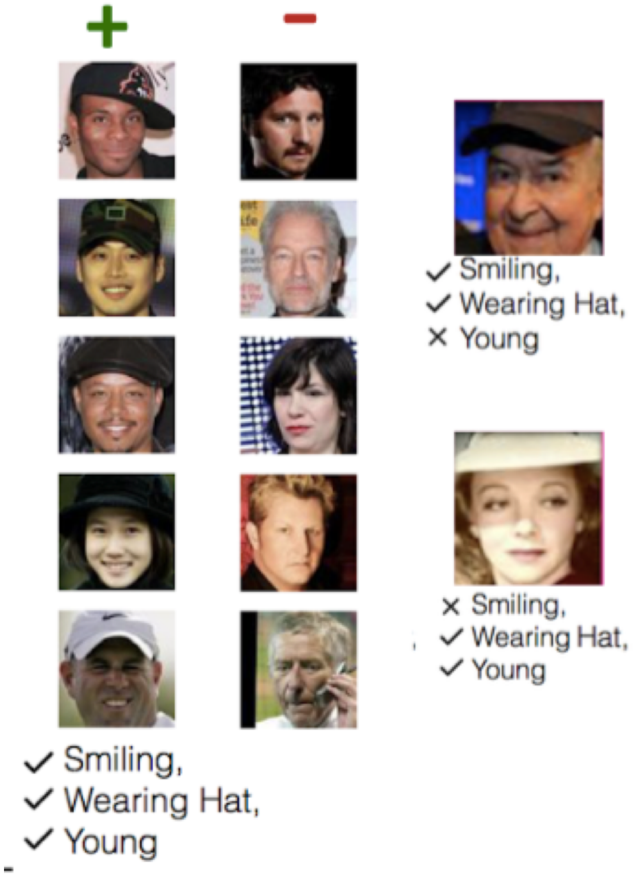
Uncertainty problems

Black-box:

- VERSA

Optimization:

- PLATIPUS
- Bayesian MAML (BMAML)
- Probabilistic MAML (PMAML)



What matters with meta learning?

Related techniques in deep machine learning

- Reinforcement learning
  - Meta-RL is a massive active research area!
- GNN & Adversarial attack
  - Metric-based, feature extraction
  - Attack on GNN by meta learning 😎

- GAN: metaGAN
  https://papers.nips.cc/paper/7504-metagan-an-adversarial-approach-to-few-shot-learning.pdf
- Transfer learning: Meta-Transfer Learning (MTL)
  http://openaccess.thecvf.com/content_CVPR_2019/papers/Sun_Meta-Transfer_Learning_for_Few-Shot_Learning_CVPR_2019_paper.pdf

Life-long learning, Domain adaption, Multi-task…

Reference

- https://icml.cc/media/Slides/icml/2019/halla(10-09-15)-10-13-00-4340-meta-learning_.pdf
- https://arxiv.org/abs/1810.09502
- https://papers.nips.cc/paper/8306-meta-learning-with-implicit-gradients.pdf
- https://arxiv.org/abs/1807.05960