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 ➔

THE INTERNET

IS THIS A MEME?
What is meta learning?
(n.)
= learn to learn
(v.)
MAML
ML - Machine Learning
Meta learning - Definition

- Learn to learn

\[ x \xrightarrow{f} y \]  

learns

\[ F \xrightarrow{f} y \]  

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

algorithm model

Task
Motivations for meta learning

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

(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 → “general” AI
   ○ Related to:
     ■ transfer learning
     ■ domain adaptation
     ■ multi-task learning
     ■ life-long learning
     ■ ...

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 → “general” AI
   ○ Related to:
     ■ transfer learning
     ■ domain adaptation
     ■ multi-task learning
     ■ life-long learning
     ■ ...

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

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

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

https://drive.google.com/file/d/1DuHyotdwEAEhmuHQWwRosdiVBVGm8uYx/view

(即使資料不多，或許也 train 得起來)
How to do meta learning?
Recap

- Optimization based
  - Meta as LSTM
  - Meta as RNN

- Metric based
  - Siamese Network
  - Matching Network
  - Prototypical Network
  - Relation Network

- MetaLSTM
  - MAML
  - Reptile
  - MANN
  - SNAIL
(Interesting **Names** of) Models / Techniques

- MAML (*Model Agnostic Meta-Learning*)
- Reptile (???)
- SNAIL (*Simple Neural AttentIve Learner*)
Interesting Names of) Models / Techniques

- **PLATIPUS** (Probabilistic LATent model for Incorporating Priors and Uncertainty in few-Shot learning)
- **LLAMA** (Lightweight Laplace Approximation for Meta-Adaptation)
- **ALPaCA** (Adaptive Learning for Probabilistic Connectionist Architectures)
- **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) 豬
to 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: evolitional 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 β δ ι λ υ βγδε… ড ন ল র কি…
How about … “omniglot” of anime? ☺️

來自深淵

魔法少女小圓

ナルクノソコデマツ

BEATRICE


為美好的世界獻上祝福！（このすば）

What can we meta learn on? → Datasets

2. miniImageNet
   - from ImageNet but few-shot

3. CUB (Caltech-UCSD Birds)

CIFAR-FS, FC100, Fungi...
Categories

- **Black-box ((memory) Model) based**
- **Optimization (Gradient) based (Parametric)**
  - Learn to initialize
- **Non-parametric (Metric based)**
  - Learn to compare

(Not unified, but generally...)

**Hybrid**
- MANN
- SNAIL
- MetaNet...
- MAML
- Reptile...
- iMAML
- MAML++...
- Siamese Network
- Matching Network
- Prototypical Network
- Relation Network...
- IMP...

**Bayesian**
- PLATIPUS...

**Meta-LSTM**
- LEO...
Black-box

• 直接用 RNN 架構硬 train 一發！

希望中間的 RNN 去學出 distribution
Black-box

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

直接 train 就 train 不起來
→ 劣一些手腳
1. 加上 memory
   (Neural Turing Machine)
   (LST“M”)
2. 加上 attention
   (Transformer)
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

Loss Function:

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

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

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

\( \hat{\theta}^n \) depends on \( \phi \)

\[ \phi \leftarrow \phi - \eta \nabla_\phi L(\phi) \]

Considering one-step training:

\[ \hat{\theta} = \phi - \epsilon \nabla_\phi l(\phi) \]

Network Structure

init

Update

\[ \theta^1 \rightarrow \text{Update} \rightarrow \theta^2 \]

Learning Algorithm (Function \( F' \))

Only focus on initialization parameter \( \phi \)

Training Data

Compute Gradient

meta-learning

learning/adaptation

\[ \nabla L_1, \nabla L_2, \nabla L_3 \]

\[ \phi_1^*, \phi_2^*, \phi_3^* \]
Problems of MAML

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

- “Adaptive learning rate” version of MAML

Outer loop

- Meta params: $\theta$
- Learning rate: $\beta$
- Loss: $\Sigma L_{\text{test}(T_i)}(\theta_i)$

Inner loop

- Task params: $\phi^* (\theta' \text{ original})$
- Learning rate: $\alpha$ (vector)
- Loss:
  $L(\theta_i) = \frac{1}{|\text{data}|} \Sigma l(f_{\theta}(x), y)$
How to train your MAML?

**Problems of MAML**

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

- Implicit gradients $\rightarrow$ iMAML
- Closed-form on feature extraction $\rightarrow$ R2-D2
iMAML

\[
\theta_{\text{ML}}^* := \arg\min_{\theta \in \Theta} F(\theta), \text{ where } F(\theta) = \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}\left(\text{Alg}(\theta, D_{i}^{\text{tr}}), D_{i}^{\text{test}}\right).
\]

\[
\phi_i \equiv \text{Alg}(\theta, D_{i}^{\text{tr}}) = \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, D_{i}^{\text{tr}}). \quad \text{(inner-level of MAML)}
\]
First-Order MAML

This is just an **approximation**!
→ Not quite accurate
Can we do better? Consider the following...

- 如果要使 meta param. 發揮作用 我們的 $\phi$ 不應該離 $\theta$ 太遠！

$\text{Alg}^*(\theta, D_i^{tr}) = \arg\min_{\phi' \in \Phi} \mathcal{L}(\phi', D_i^{tr}) + \frac{\lambda}{2} \|\phi' - \theta\|^2.$  

(This is $\phi$.)

Proximal regularization!

Implicit MAML (this work)
How does it perform?

(a) Normalized error = $\frac{\|\hat{g}_r - g_r\|}{\|g_r\|}$

(b) GPU Memory (Normalized)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Compute</th>
<th>Memory</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML (GD + full back-prop)</td>
<td>$\kappa \log \left( \frac{D}{\delta} \right)$</td>
<td>$\text{Mem}(\nabla \hat{L}_i) \cdot \kappa \log \left( \frac{D}{\delta} \right)$</td>
<td>0</td>
</tr>
<tr>
<td>MAML (Nesterov’s AGD + full back-prop)</td>
<td>$\sqrt{\kappa} \log \left( \frac{D}{\delta} \right)$</td>
<td>$\text{Mem}(\nabla \hat{L}_i) \cdot \sqrt{\kappa} \log \left( \frac{D}{\delta} \right)$</td>
<td>0</td>
</tr>
<tr>
<td>Truncated back-prop [53] (GD)</td>
<td>$\kappa \log \left( \frac{D}{\delta} \right)$</td>
<td>$\text{Mem}(\nabla \hat{L}_i) \cdot \kappa \log \left( \frac{1}{\epsilon} \right)$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>Implicit MAML (this work)</td>
<td>$\sqrt{\kappa} \log \left( \frac{D}{\delta} \right)$</td>
<td>$\text{Mem}(\nabla \hat{L}_i)$</td>
<td>$\delta$</td>
</tr>
</tbody>
</table>
R2-D2: closed form solvers

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）
Black-box v.s Gradient based

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

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

\[ \phi_i = \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, D^{\text{tr}}_i, f(\theta, D^{\text{tr}}_i, \nabla_{\theta} \mathcal{L})) \]

\[ \text{Stored in } \theta \]

→ Meta-LSTM, even more general and unifying!
Metric-based / non-parametric

Learn to compare!

Network as an algorithm

A bigger function

Learning Algorithm (Function $F$)
Metric-based / non-parametric

Learn to compare!

Feature extractor (Embedding)

A bigger function

Network as an algorithm

KNN Cosine L2

Training Data

Testing Data
Metric-based / non-parametric

Learn to compare!

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

Why?

• Few-shot \(\rightarrow\) Just compare! No too complex models needed!
  (Even compare with something like L2-distances or pixel spaces)

• Too simple? \(\rightarrow\) 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!
No need to parametrize all the model, just the embedding / representation!

• Siamese network
• Prototypical network
• Matching network
• Relation network

Better representation
• IMP
• GNN

One reason: few-shot!
Metric-based / non-parametric

Learn to compare!

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

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

Better representation

- IMP
- GNN

One reason: few-shot!
Metric-based / non-parametric

Learn to **compare**!

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

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

Better representation
- IMP
- GNN

One reason: few-shot!
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)

- 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 $\rightarrow$ difficult to scale
- Limited to classification (only learning to compare)
Hybrid

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

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
  [Link to metaGAN](https://papers.nips.cc/paper/7504-metagan-an-adversarial-approach-to-few-shot-learning.pdf)

- Transfer learning: Meta-Transfer Learning (MTL)

Life-long learning, Domain adaption, Multi-task...
Reference