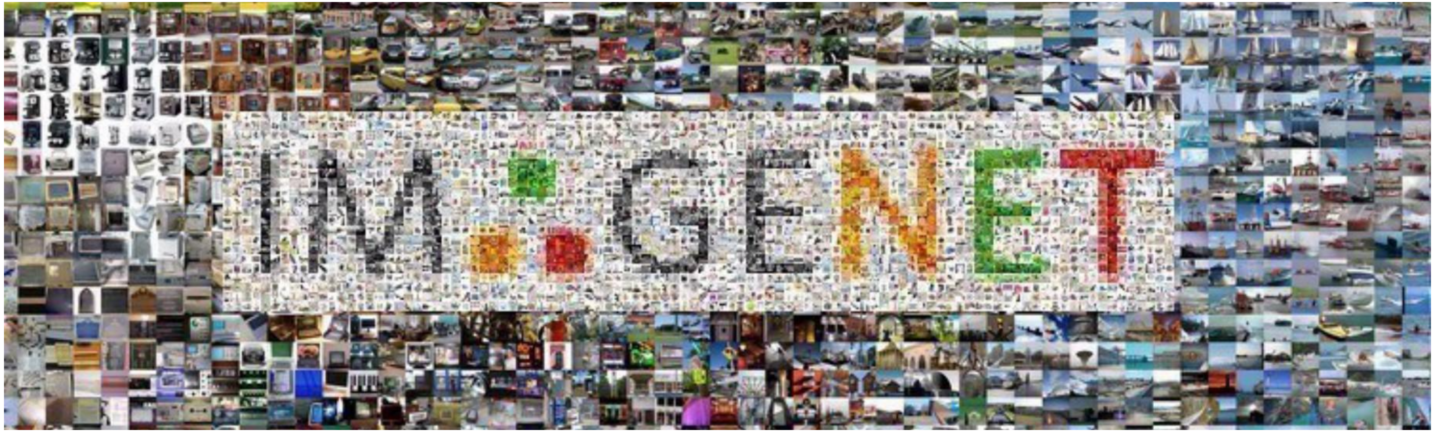
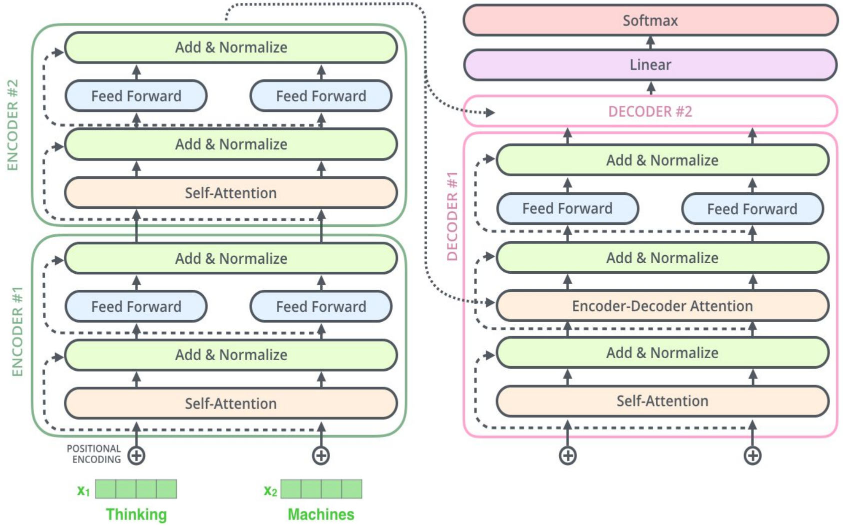
Meta Learning Is All You Need

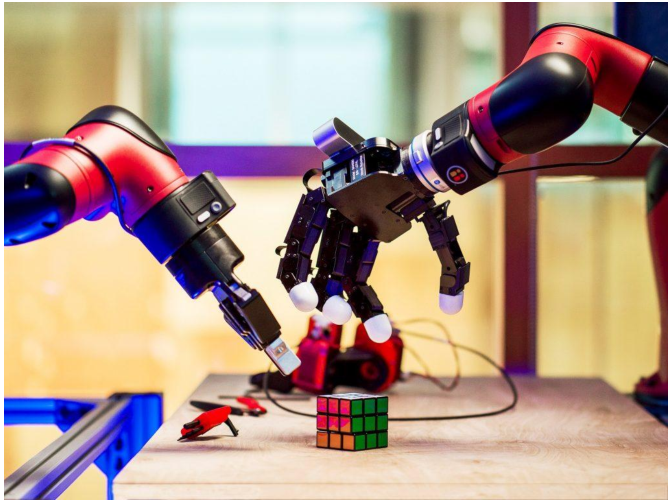
James Le
06/17/2020

1 - Motivation For Meta-Learning




The scientist named the population, after their distinctive horn, Ovid's Unicorn.
GPT-2


Softmax
Linear
DECODER #2
DECODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Encoder-Decoder Attention
Add & Normalize
Self-Attention
ENCODER #2
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
ENCODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
POSITIONAL ENCODING
x1
x2
Thinking
Machines



What If We Don’t Have A Large Dataset?

What If Our Data Has A Long Tail?

What If We Want to Quickly Learn Something New?

Multitask Learning*

RICH CARUANA

Multitask Learning (MTL) is an inductive transfer mechanism whose principle goal is to improve generalization performance. MTL improves generalization by leveraging the domain-specific information contained in the training signals of *related* tasks. It does this by training tasks in parallel while using a shared representation. In effect, the training signals for the extra tasks serve as an inductive bias. Section 1.2 argues that inductive transfer is important if we wish to scale tabula rasa learning to complex, real-world tasks. Section 1.3 presents the simplest method we know for doing multitask inductive transfer, adding extra tasks (i.e., extra outputs) to a backpropagation net. Because the MTL net uses a shared hidden layer trained in parallel on all the tasks, what is learned for each task can help other tasks be learned better. Section 1.4 argues that it is reasonable to view training signals as an inductive bias when they are used this way.

Caruana, 1997

Is Learning The n-th Thing Any Easier Than Learning The First?

Sebastian Thrun[1]

They are often able to generalize correctly even from a single training example [2, 10]. One of the key aspects of the learning problem faced by humans, which differs from the vast majority of problems studied in the field of neural network learning, is the fact that humans encounter a whole stream of learning problems over their entire lifetime. When faced with a new thing to learn, humans can usually exploit an enormous amount of training data and experiences that stem from other, related learning tasks. For example, when learning to drive a car, years of learning experience with basic motor skills, typical traffic patterns, logical reasoning, language and much more precede and influence this learning task. The transfer of knowledge across learning tasks seems to play an essential role for generalizing accurately, particularly when training data is scarce.

Thrun, 1998

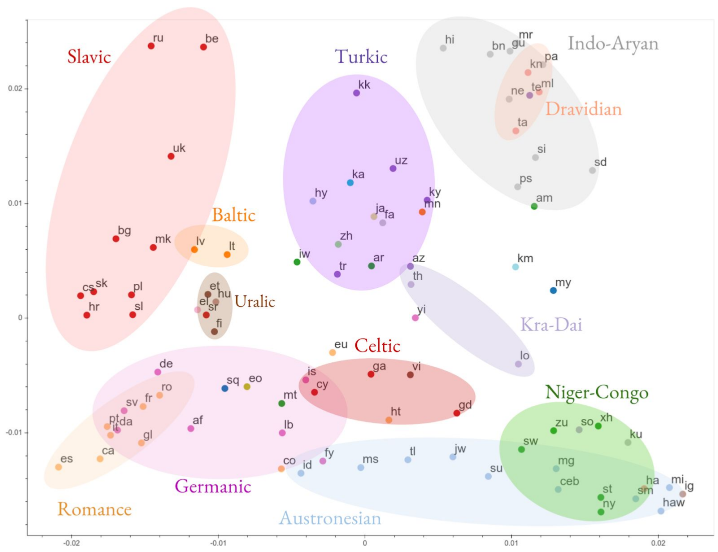
On the Optimization of a Synaptic Learning Rule

Samy Bengio Yoshua Bengio Jocelyn Cloutier Jan Gecsei

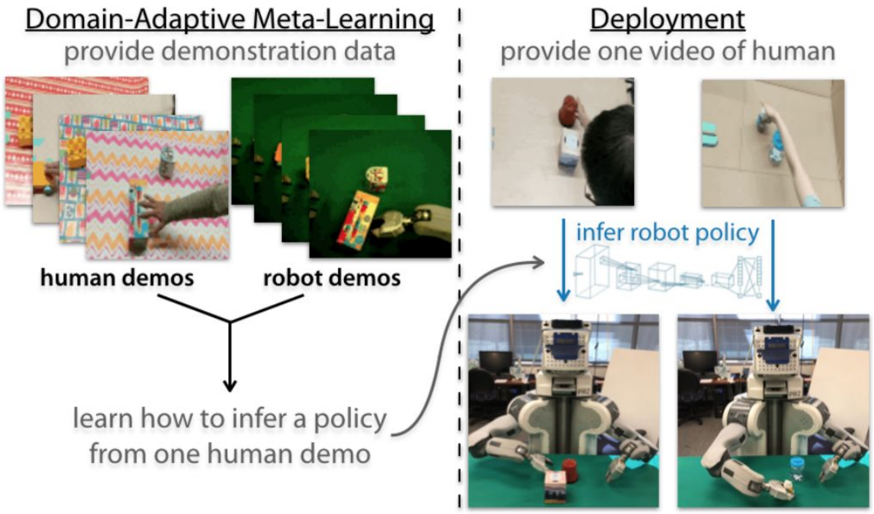
Université de Montréal, Département IRO

This paper presents a new approach to neural modeling based on the idea of using an automated method to optimize the parameters of a synaptic learning rule. The synaptic modification rule is considered as a parametric function. This function has *local* inputs and is the same in many neurons. We can use standard optimization methods to select appropriate parameters for a given type of task. We also present a theoretical analysis permitting to study the *generalization* property of such parametric learning rules. By generalization, we mean the possibility for the learning rule to learn to solve *new* tasks. Experiments were performed on three types of problems: a

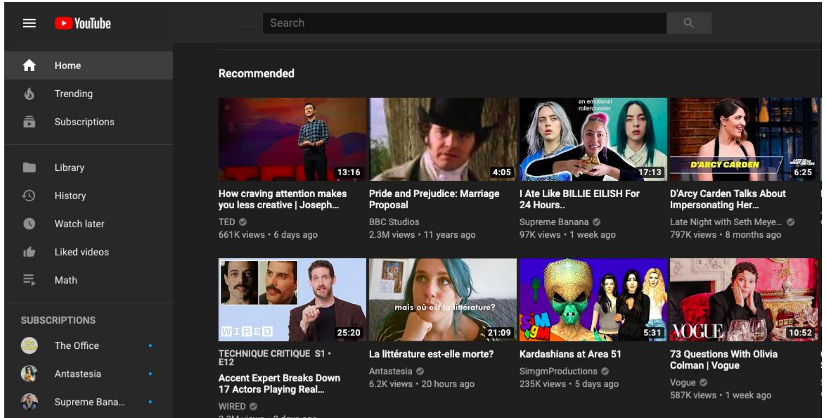
Bengio et al. 1992



Aharoni et al., Massively Multilingual Neural Machine Translation, 2019



Yu et al., Domain-Adaptive Meta-Learning, 2018



YouTube, Recommending What Video To Watch Next, 2019

2 - Basics Of Meta-Learning

Supervised Learning

$$arg \max_{\phi} logp(\phi|D)$$

$$= arg \max_{\phi} logp(D|\phi) + logp(\phi)$$

$$= arg \max_{\phi} \sum_{i} logp(y_i|x_i, \phi) + logp(\phi)$$

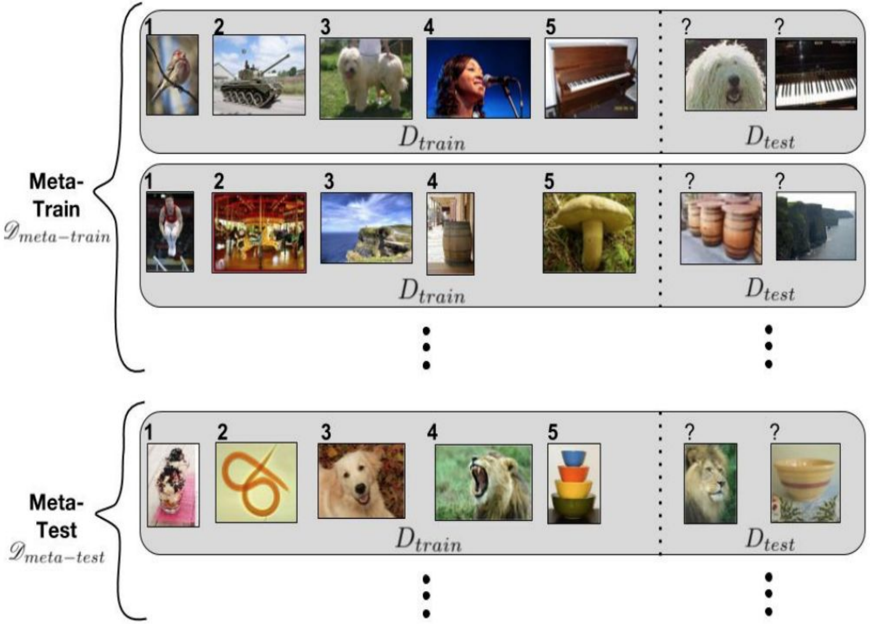
- Big models require large amounts of labeled data
- Labeled data for some tasks may be very limited
- Can we incorporate **additional** data?

Supervised Meta-Learning

$$\arg\max_{\phi} logp(\phi|D, D_{meta-train})$$

$$logp(\phi|D, D_{meta-train}) = log \int_{\Theta} p(\phi|D,\theta)p(\phi|D_{meta-train})d\theta$$

$$\approx \underbrace{logp(\phi|D,\theta^*)}_{\text{Adaptation Task}} + \underbrace{logp(\theta^*|D_{meta-train})}_{\text{Meta-Training Task}}$$



Ravi and Larochelle, ICLR 2017

Meta-Learning Optimization

$$D_{meta-train} = \{(D_1^{train}, D_1^{test}), \cdots, (D_n^{train}, D_n^{test})\}$$

Meta-Training Phase:

ϕ* = arg max log p(ϕ|D,θ*)

$$D_i^{train} = \{(x_1^i, y_1^i), \cdots, (x_k^i, y_k^i)\}; D_i^{test} = \{(x_1^i, y_1^i), \cdots, (x_l^i, y_l^i)\}$$

Adaptation Phase:

θ* = max log p(θ|D_{meta-train})

Learn θ such that:

$$\theta^* = \max_\theta \sum_{i=1}^{n} logp(\phi_i | D_i^{test})$$

ϕ_i = f_{θ} (D_i^{tr})

is good enough for D_i^{ts}

The Recipe to Design a Meta-Learning Algorithm

1. Choose a form of $p(\phi_i | D_i^{tr}, \theta)$ (**adaptation task**)
2. Choose how to optimize θ with respect to maximum-likelihood objective using D_{meta-train} (**meta-training task**)

3 - Black-Box Meta-Learning

Formulation

$$\max_{\theta} \sum_{T_i} \sum_{(x,y)\sim D_i^{test}} log g_{\phi_i}(y|x)$$

$$\sum_{(x,y)\sim D_i^{test}} log g_{\phi_i}(y|x) = L(\phi_i, D_i^{test})$$

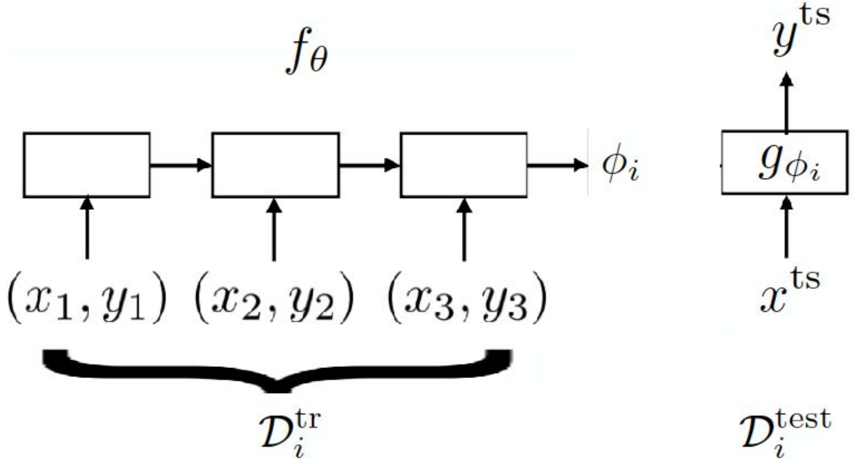
Train a neural network to represent $p(\phi_i | D_i^{tr}, \theta)$:

$\phi_i = f_{\theta}(D_i^{tr})$

$$\max_{\theta} \sum_{T_i} L(f_\theta(D_i^{train}), D_i^{test})$$

Train another neural network for inference on test set:

$D_i^{ts} = g_{\phi_i}$



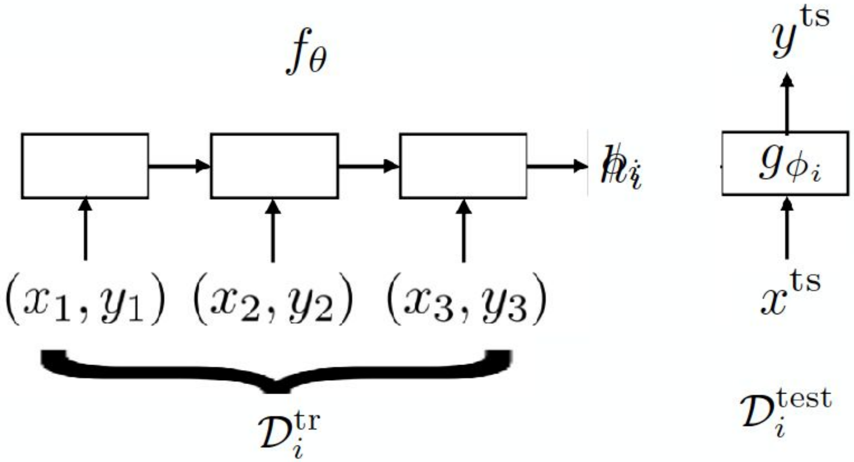
Black-Box Meta-Learning Algorithm

1. Sample a task T_i (or mini batch of tasks)
2. Sample disjoint sets D_i^{tr} and D_i^{t} from D_i
3. Compute $\phi_i \leftarrow$ f_{θ} (D_i^{tr})
4. Update θ using ∇_{θ} $L(\phi_i, D_i^{t})$

Challenge

Outputting all neural net parameters won't be scalable!

Idea: Only output the sufficient statistics, not **all** parameters of the network (Santoro et al. MANN, Mishra et al. SNAIL)



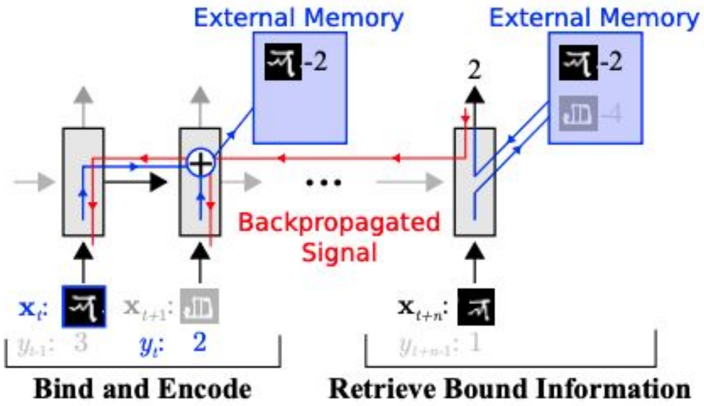
The low-dimensional vector h_i represents contextual task information

$\boldsymbol{\phi_i = \{h_i, \theta\}}$

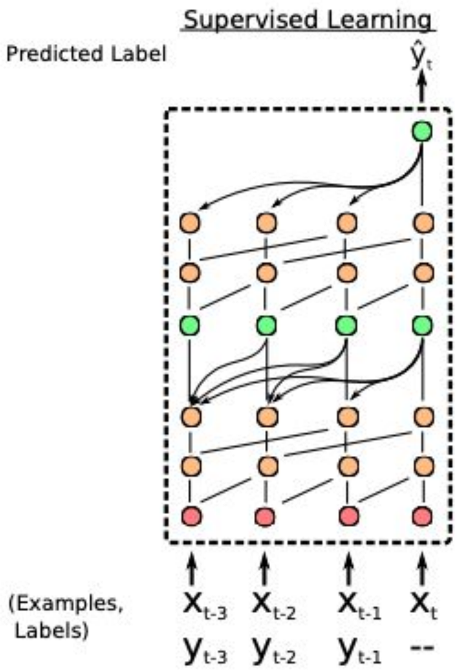
General form:

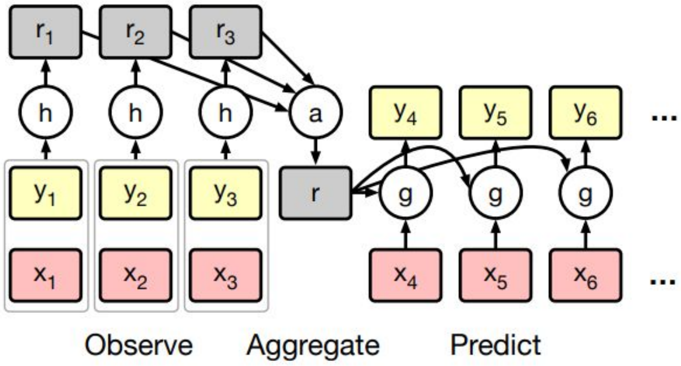
$$y^{ts} = f_\theta(D_i^{train}, x^{ts})$$

Black-Box Architectures



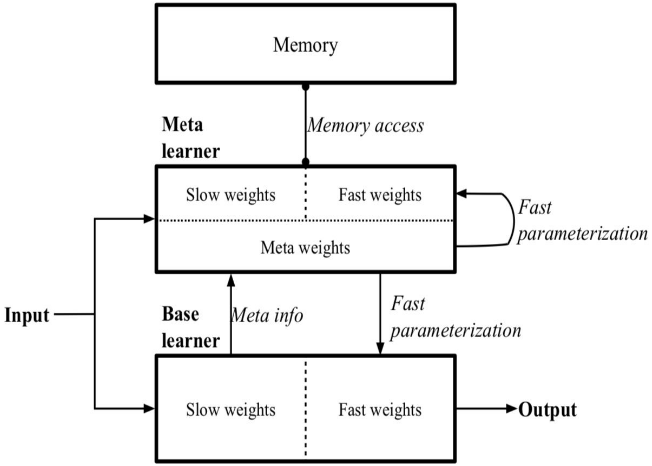
MANN, ICML'16





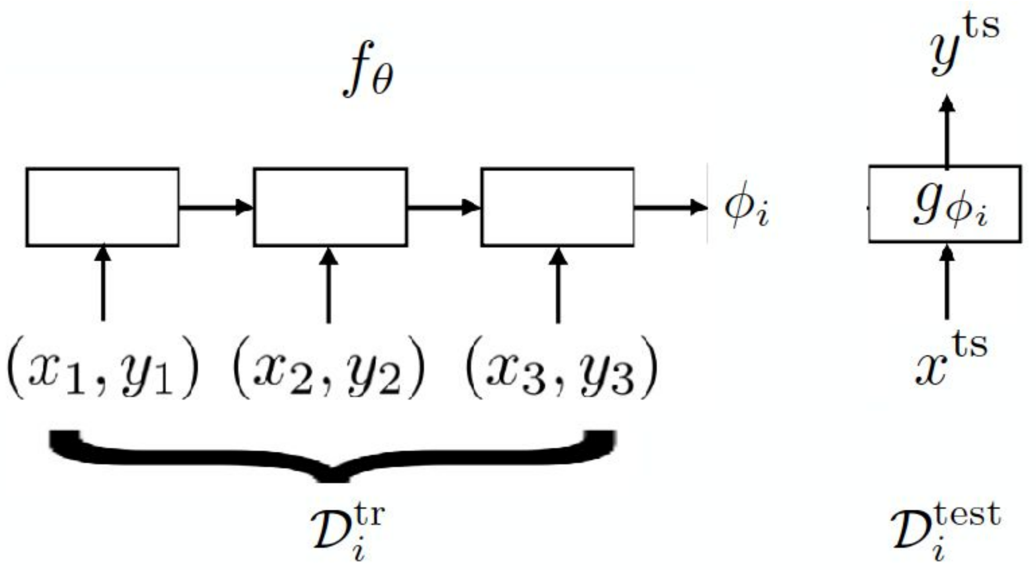
Conditional Neural Processes, ICML'18

Meta Networks, ICML'17



SNAIL, ICLR'18

Black-Box Adaptation



- \+ Expressive
- \+ Easy to combine with variety of learning problems
- \- Complex model with complex task -> challenging optimization problem
- \- Data-inefficient

=> How else can we represent $p(\phi_i | D_i^{tr}, \theta)$ in a scalable way?

4 - Optimization Based Meta-Learning

Formulation

$$\max_{\phi_i} logp(D_i^{train}|\phi_i) + logp(\phi_i|\theta)$$

Acquire ϕ_i through **optimization**

pre-trained parameters

Meta-parameters θ are **pre-trained**

$$\phi \leftarrow \theta - \alpha \nabla_\theta L(\theta, D^{train})$$

Training data for new task

Model-Agnostic Meta Learning (Finn et al., ICML’17):

- **Fine-tuning** using pre-trained parameters θ and train data
- **Meta-training** includes the loss between the results from fine-tuning and test data
- Pre-trained parameters come from **publicly large available datasets**

$$\min_{\theta} \sum_{task_i} L(\theta - \alpha \nabla_\theta L(\theta, D_i^{train}), D_i^{test})$$

Optimization-Based Meta-Learning Algorithm

1. Sample a task T_i (or mini batch of tasks)
2. Sample disjoint sets D_i^{tr} and D_i^t from D_i
3. Optimize $\phi_i \leftarrow \theta - \alpha \nabla\_\theta L(\theta, D_i^{tr})$
4. Update θ using $\nabla\_\{\theta\} L(\phi_i, D_i^t)$

$$y^{ts} = f_\theta(D_i^{train}, x^{ts})$$ ← Black-Box Adaptation

Optimization-Based Adaptation → $$y^{test} = f_{MAML}(D_i^{train}, x^{test}) = f_{\phi_i}(x^{test})$$

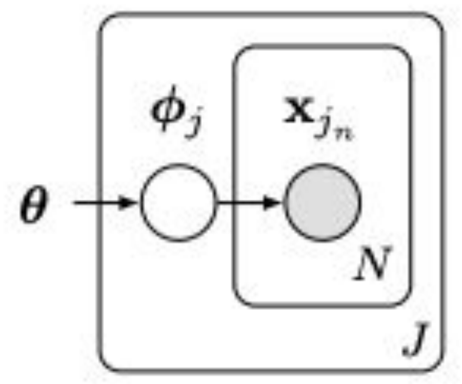
For a sufficient deep f, MAML can approximate **any** function of D_i^{tr}, x^{ts} (Finn and Levine, ICLR 2018)

Assumptions:

- Non-zero learning rate
- Loss function gradient does not lose information about the label
- Data points in training set are unique

MAML has the benefit of inductive bias without losing expressive power

Probabilistic Version Of Optimization-Based Inference



Grant et al., ICLR'18

$$\max_{\theta} log \prod_j p(D_j|\theta)$$

$$= log \prod_j \int p(D_j|\phi_j)p(\phi_j|\theta)d_{\phi_j}$$

Empirical Bayes

$$\approx log \prod_j p(D_i|\hat{\phi}_j)p(\hat{\phi}_j|\theta)$$

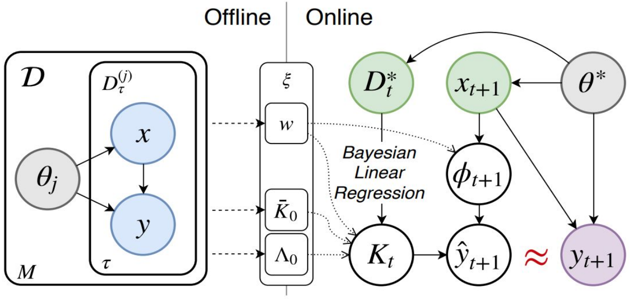
MAP Estimate

How to compute MAP estimate?
Gradient descent with early stopping (MAML): *implicit Gaussian prior*

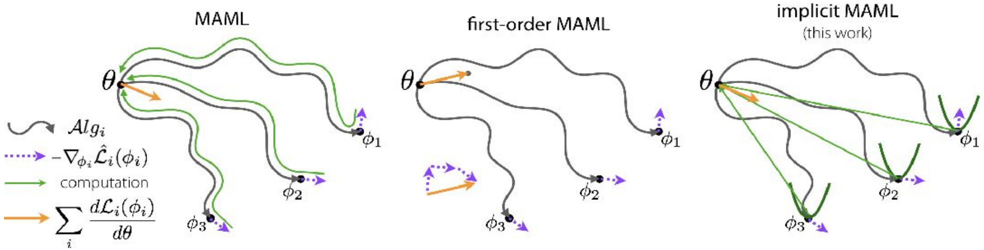
$$\phi \leftarrow \theta - \alpha \nabla_{\theta} L(\theta, D^{train})$$

Probabilistic Version Of Optimization-Based Inference

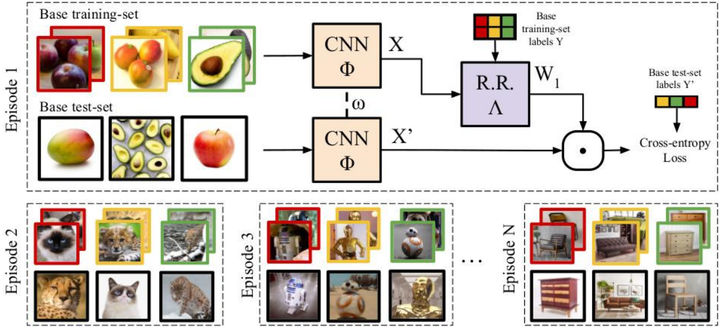
Other Ways to Compute MAP Estimate?



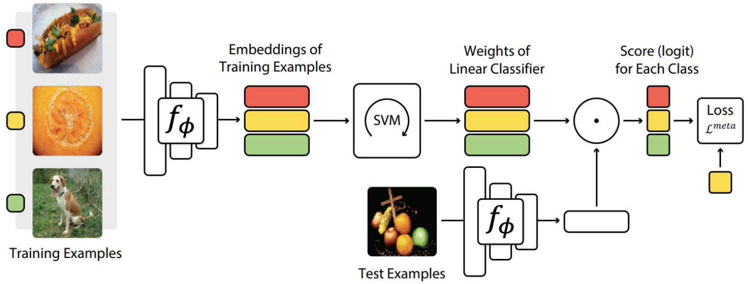
Bayesian linear regression on learned features
(Harrison et al. ALPaCA '18)



Explicit Gaussian Prior (Rajeswaran et al, Implicit MAML '19)



Ridge/logistic regression on learned features
(Bertinetto et al. R2-D2 '18)



Support Vector Machine on learned features
(Lee et al. MetaOptNet '19)

Challenges

How to choose architecture that is effective for inner gradient-step?

Idea: Progressive neural architecture search + MAML (Kim et al., Auto-Meta, NIPS’2018)

- Finds highly non-standard architecture with deep & narrow layers
- Different from architectures that work well for standard supervised learning

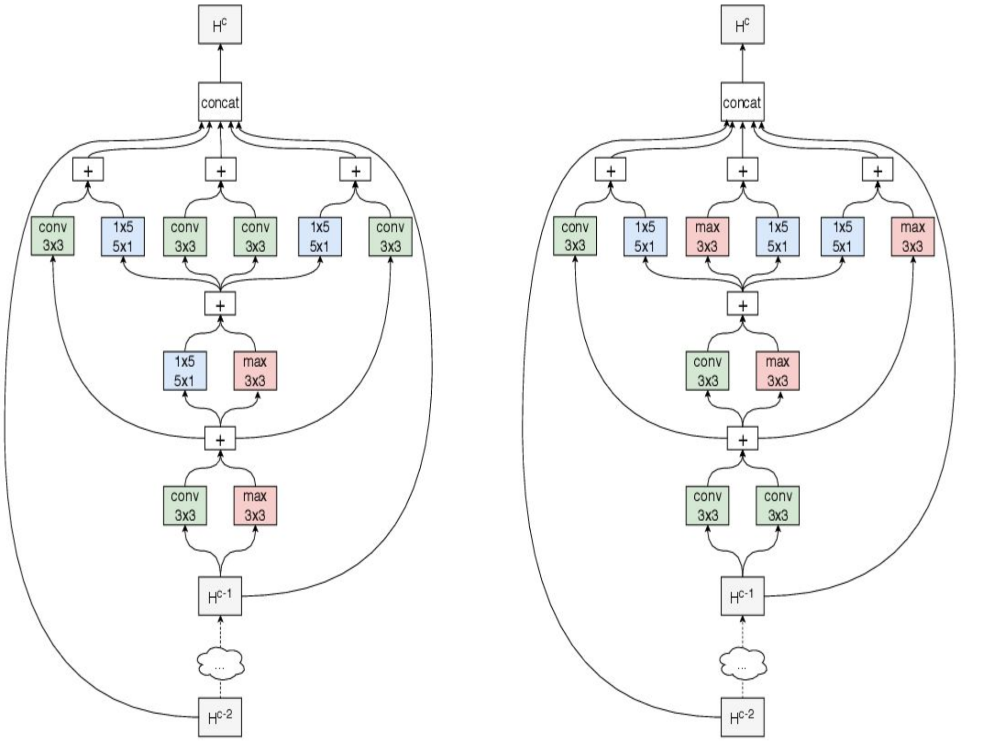


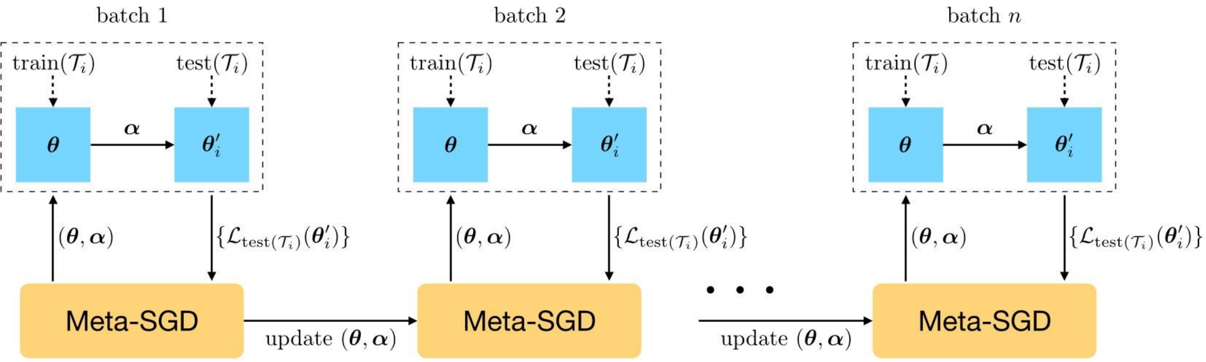
Figure 1: The best cell architectures for 1-shot 5-way Mini-Imagenet tasks in the small (left) and

Challenges

Bi-level optimization can exhibit instabilities

Idea: Automatically learn inner vector learning rate, tune outer learning rate

Li et al., Meta-SGD'17



6th ICML Workshop on Automated Machine Learning (2019)

Alpha MAML: Adaptive Model-Agnostic Meta-Learning

Harkirat Singh Behl HARKIRAT@ROBOTS.OX.AC.UK
Atılım Güneş Baydin GUNES@ROBOTS.OX.AC.UK
Philip H.S. Torr PHST@ROBOTS.OX.AC.UK
University of Oxford

arXiv:1905.07435v1 [cs.LG] 17 May 2019

Abstract

Model-agnostic meta-learning (MAML) is a meta-learning technique to train a model on a multitude of learning tasks in a way that primes the model for few-shot learning of new tasks. The MAML algorithm performs well on few-shot learning problems in classification, regression, and fine-tuning of policy gradients in reinforcement learning, but comes with the need for costly hyperparameter tuning for training stability. We address this shortcoming by introducing an extension to MAML, called Alpha MAML, to incorporate an online hyperparameter adaptation scheme that eliminates the need to tune meta-learning and learning rates. Our results with the Omniglot database demonstrate a substantial reduction in the need to tune MAML training hyperparameters and improvement to training stability with less sensitivity to hyperparameter choice.

1. Introduction

Meta-learning—or "learning to learn"—concerns machine learning models that can improve their learning quality by altering aspects of the learning process such as the model architecture, optimization rules, initialization, or learning hyperparameters (Thrun and Pratt, 2012; Schmidhuber, 1987; Hochreiter et al., 2001). An important application of meta-learning is in few-shot learning problems (Vinyals et al., 2016; Behl et al., 2018), where one is concerned with developing methods able to learn new concepts from one or only a few instances (Lake et al., 2015). In this paper we focus on the state-of-the-art *model-agnostic meta-learning* (MAML) (Finn et al., 2017) method, which is a conceptually simple and general algorithm that has been shown to outperform existing approaches in tasks including few-shot image classification and few-shot adaptation in reinforcement learning (Antoniou et al., 2019). MAML aims to solve the few-shot learning problem by being just few gradient descent steps away from any new concepts, doing so by making the assumption that learning a new concept will just involve few parameter updates (Algorithm 1). In other words, MAML is based on learning an initial representation that can be efficiently fine-tuned for new tasks in a few steps.

The generality of MAML comes with the difficulty of choosing hyperparameters to achieve stable training in practice (Antoniou et al., 2019). MAML has two important hyper-parameters, namely the learning rate α and the meta-learning rate β, thus increasing any hyperparameter grid search computation by an order, and making it significantly more time and resource consuming than comparable methods. Another complication to this problem is the fact that it is currently not established whether the technique can benefit from a conventional decaying schedule for the inner learning rate α. Furthermore, a good value of α in MAML is even more important than for any conventional stochastic gradient descent (SGD) optimization, because only a handful of samples are available in the few-shot learning case. This has significant consequences, making it difficult to scale this algorithm

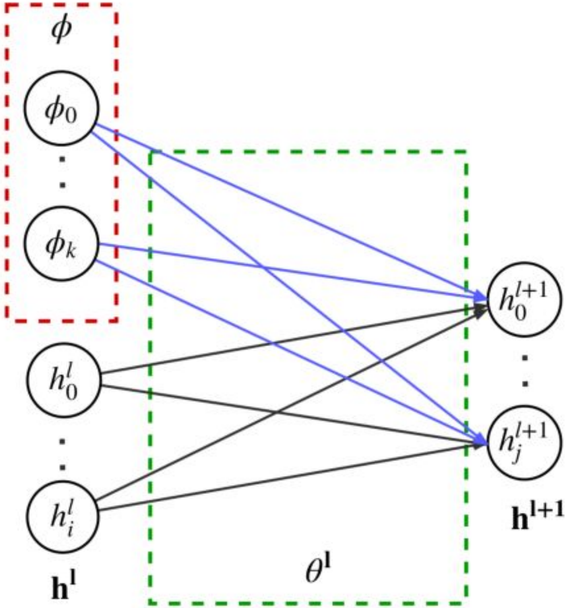
Behl et al., AlphaMAML, ICML'19

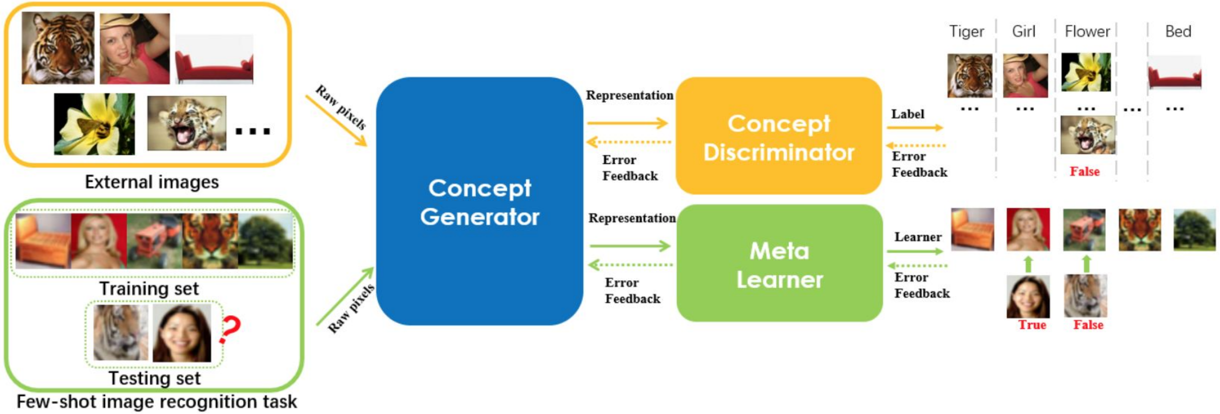
Zintgraf et al., ICML'19

Challenges

Bi-level optimization can exhibit instabilities

Idea: Optimize only a subset of the parameters in the inner loop





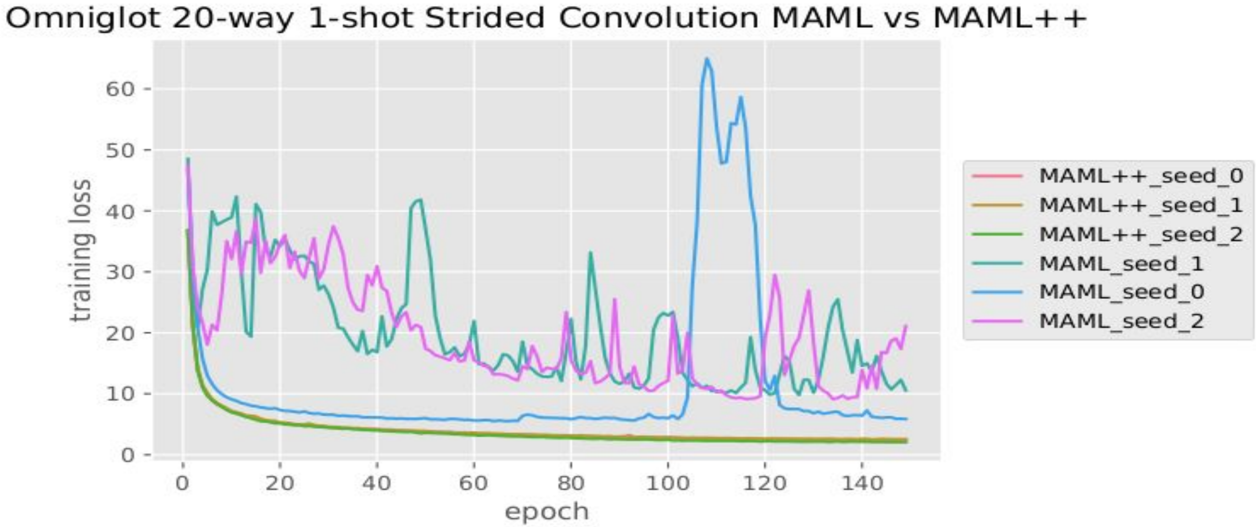
Zhou et al., DEML'18

Challenges

Bi-level optimization can exhibit instabilities

Idea: Decouple inner learning rate, Batch-Norm statistics per-step

MAML++, Antoniou'18



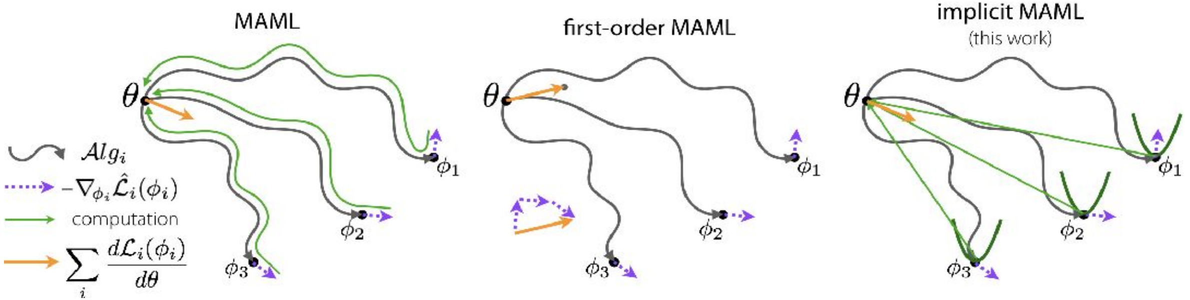
Challenges

Back-propagating through many inner gradient steps is compute- and memory-intensive

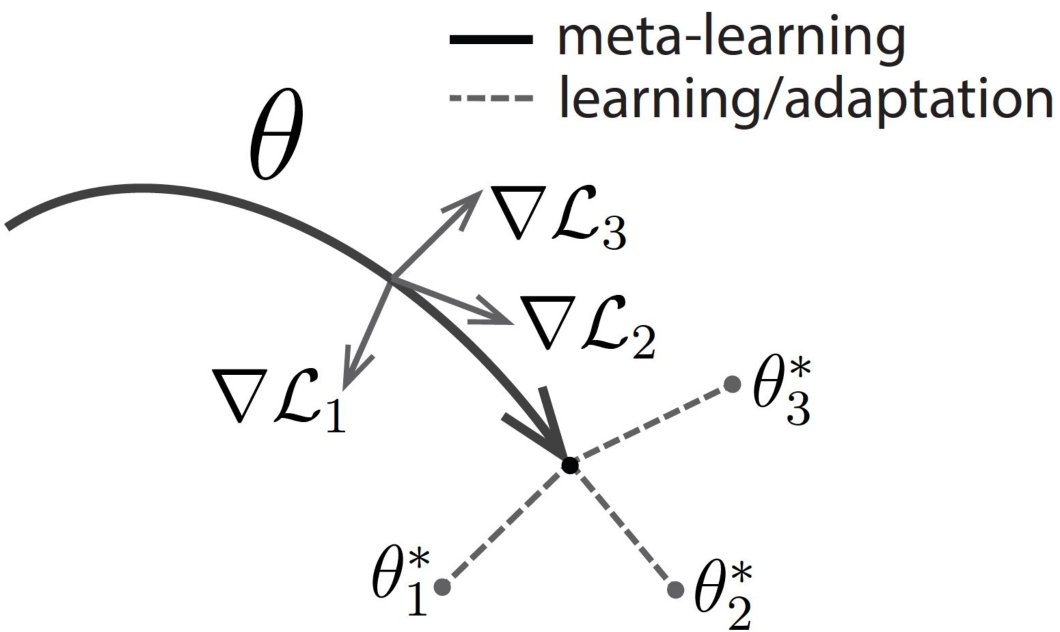
Idea: Approximate d_ϕ_i/d_θ as identity (Finn etl al. 1st-order MAML’17, Nichol et al. Reptile’18)

=> Works for simple few-shot problems, but not for more complex problems

Idea: Derive meta-gradient using implicit function theorem (Rajeswaran et al. Implicit MAML’19)



Optimization-Based Inference

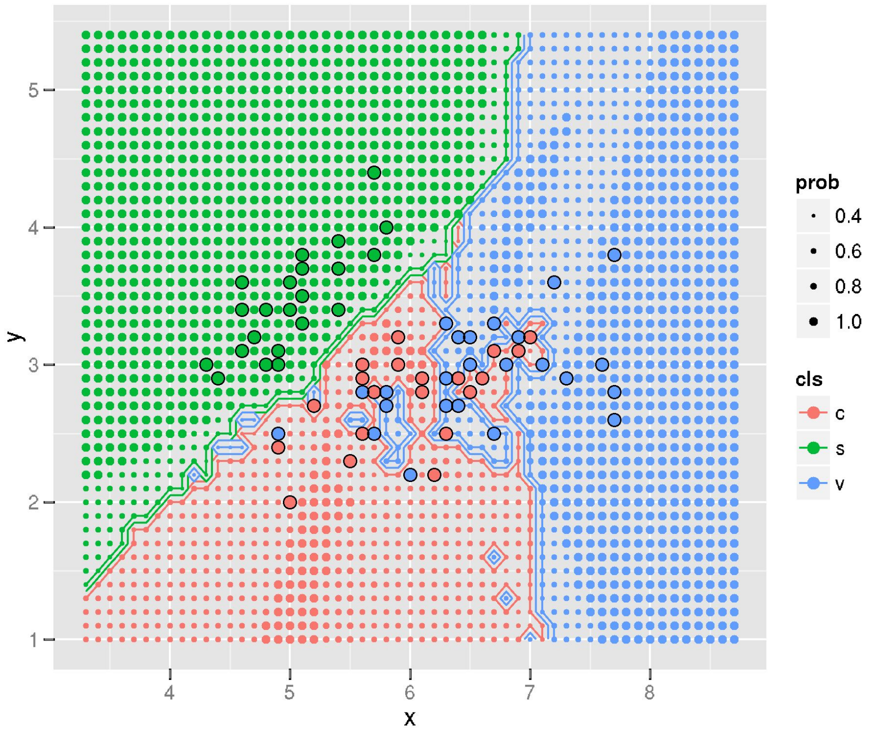


- \+ Bi-level optimization
- \+ Positive inductive bias at the start of meta-learning
- \+ Consistent procedure -> extrapolates better
- \+ Maximally expressive with sufficiently deep network
- \+ Model-agnostic
- \- Requires 2nd-order optimization
- \- Compute and/or memory intensive

5 - Non-Parametric Meta-Learning

Why Non-Parametric?

- In low data regimes, non-parametric methods are simple, work well
- Parametric during meta-training
- Non-parametric during meta-test



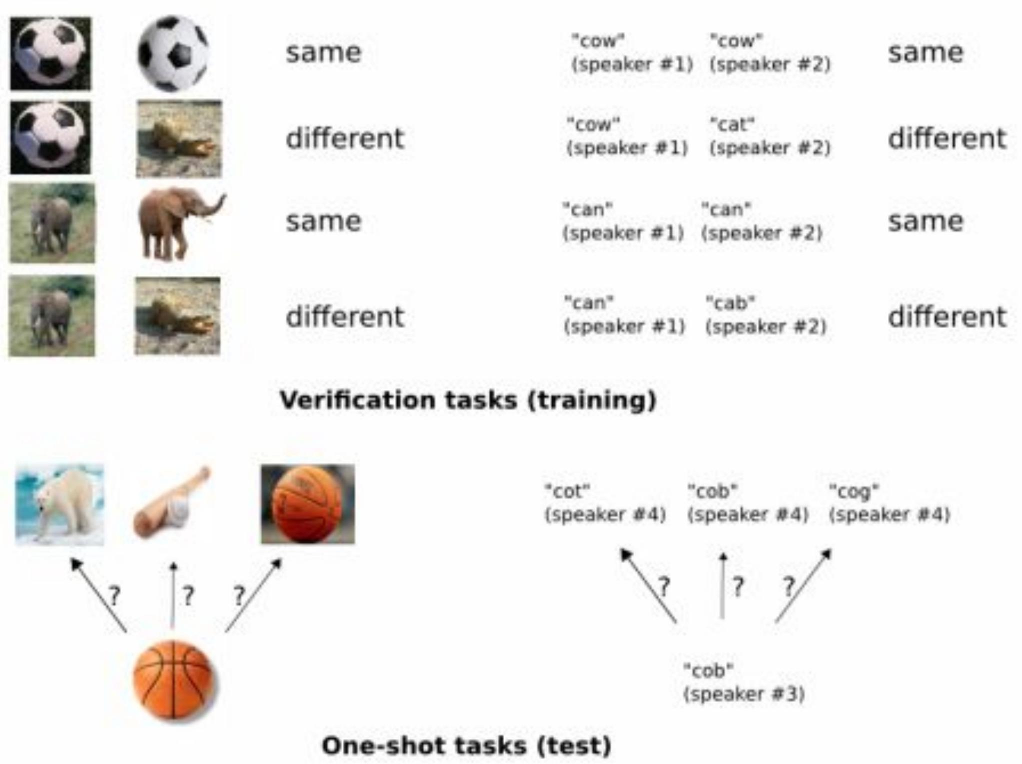
Non-Parametric Meta-Learning Algorithm

1. Sample a task T_i (or mini batch of tasks)
2. Sample disjoint sets D_i^{tr} and D_i^{t} from D_i
3. Compute $y^{ts} = \sum\{x\_k, y\_k \in D^{tr}\}\ f\_\theta\ (x^{ts}, x\_k)\ y\_k$
4. Update θ using $\nabla\_\{\theta\}\ L(y^{ts}, y^{ts})$

=> Task-specific parameters ϕ integrated out, hence non-parametric

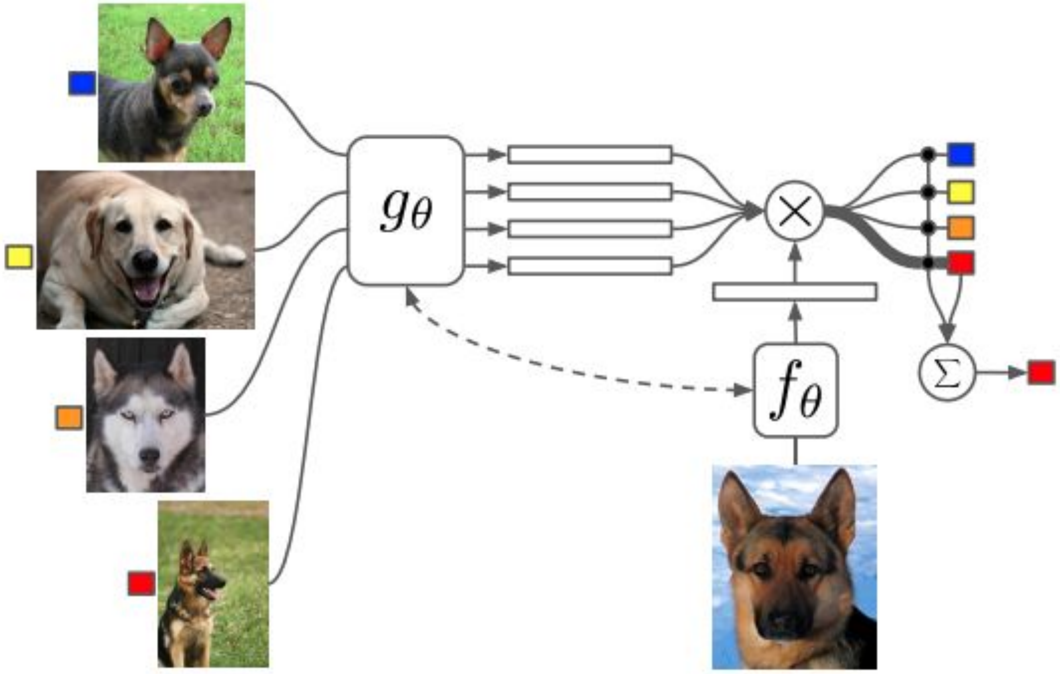
Koch et al., ICML'15

- Train a **Siamese network** to predict whether or not two images are the same class
- **Meta-Training**: Binary classification
- **Meta-Test**: N-way classification



Vinyals et al., NIPS'16

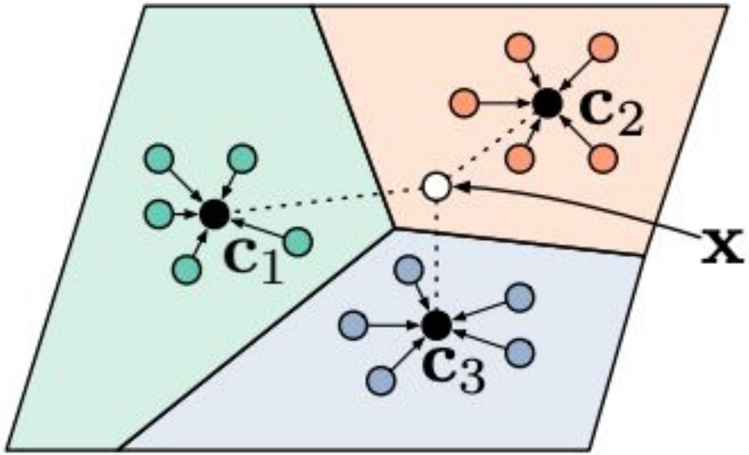
- Can we match meta-train and meta-test?
- Nearest neighbor in learned embedding space
- **Matching Networks:** Convolutional Encoder + Bi-Directional LSTM



$$\hat{y}^{test} = \sum_{x_k, y_k \in D^{train}} f_\theta(x^{test}, x_k) y_k$$

Snell et al., NIPS'17

- Can we aggregate class information to create a prototypical embedding?
- D = Distance metric between f_θ and c_k



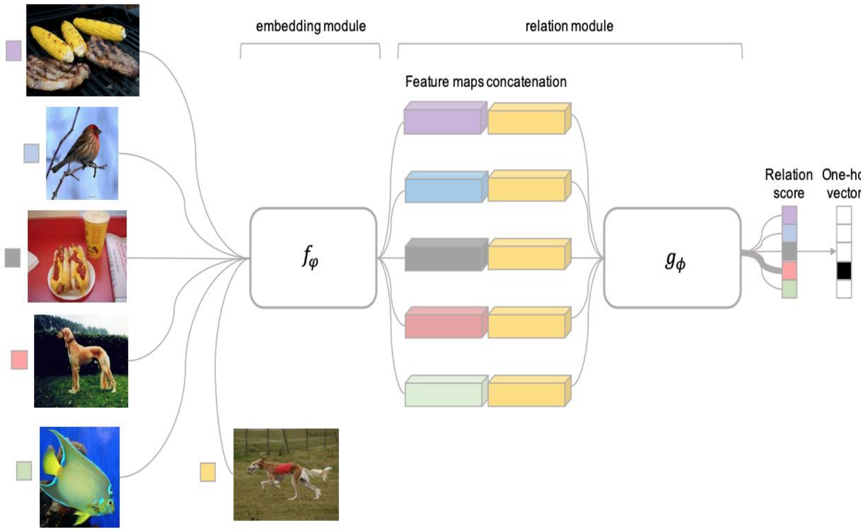
$$c_k = \frac{1}{D_i^{train}} \sum_{(x,y)\in D_i^{train}} f_\theta(x)$$

$$p_\theta(y=k|x) = \frac{softmax(-D(f_\theta(x), c_k))}{\sum_{k'} softmax(-D(f_\theta(x), c_{k'}))})$$

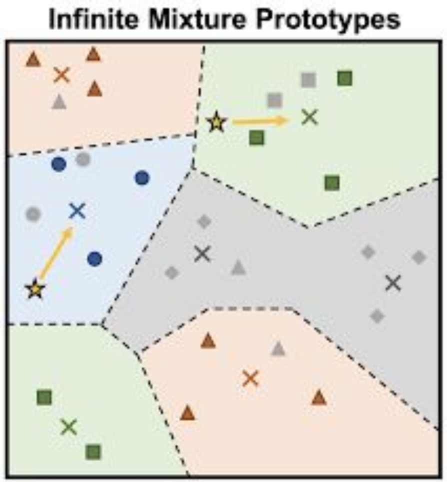
Challenge

What if we need to reason about **more complex relationships** between data points?

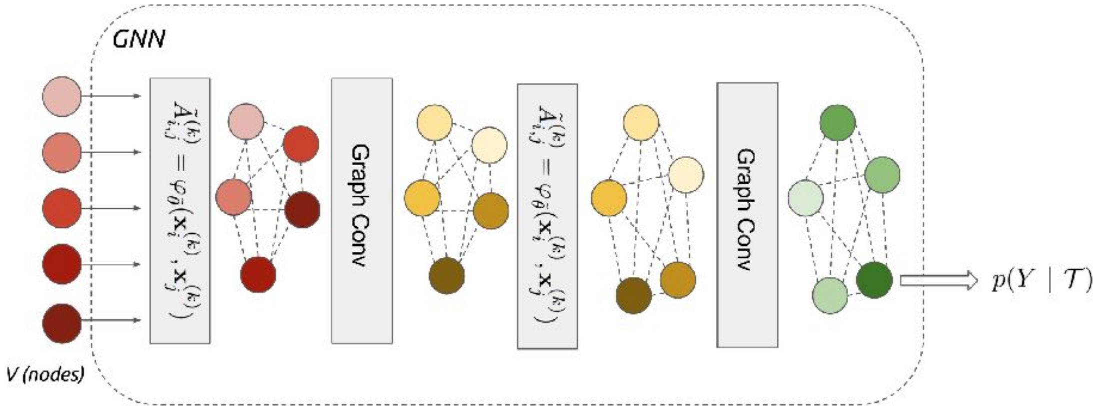
Sung et al., RelationNet (CVPR'18)



Allen et al., IMP (ICML'19)



Garcia and Bruna., GraphNet (ICLR'18)



6 - Takeaways

Takeaways (1/3)

Black-Box Meta-Learning

1. Complete expressive power
2. Not consistent
3. Easy to combine with a variety of learning problems
4. Data-inefficient

Takeaways (2/3)

Optimization-Based Meta-Learning

1. Consistent via gradient descent
2. Expressive for very deep models
3. Positive inductive bias at the start of meta-learning
4. Model-agnostic
5. Compute and memory Intensive

Takeaways (3/3)

Non-Parametric Meta-Learning

1. Expressive for most network architectures
2. Consistent under certain conditions
3. Computationally fast and easy to optimize
4. Harder to generalize and scale

Thank You!

Link to Blog Post: https://jameskle.com/writes/meta-learning-is-all-you-need