Meta Learning

Samuel Cheng

University of Oklahoma

April 14, 2024

Table of Contents

Meta Learning: Learn to learn

- Learn meta-knowledge that shares among tasks
- Often associate with few-shot learning

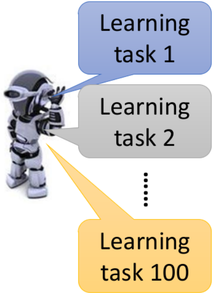


Image credit: Hung-yi Lee

Meta Learning: Learn to learn

- Learn meta-knowledge that shares among tasks
- Often associate with few-shot learning

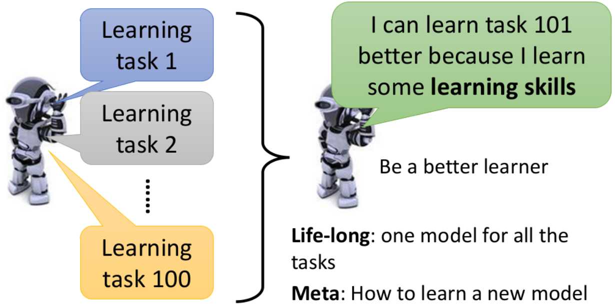


Image credit: Hung-yi Lee

Meta Learning vs Machine Learning

- Machine learning: given input and output, find a function f that maps input to output

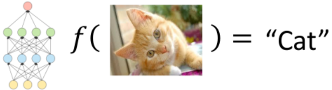
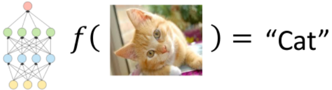


Image credit: Hung-Yi Lee

Meta Learning vs Machine Learning

- Machine learning: given input and output, find a function f that maps input to output

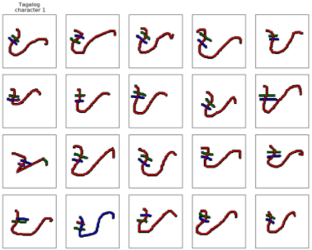


- Meta learning: given task training data, find a function F that maps training data to a good ML function $\boldsymbol{f}$

Image credit: Hung-Yi Lee

MNIST of Meta Learning: Omniglot

- 1623 characters
- 20 examples each
- Github



Other datasets: miniImageNet, CUB

Jargons

- N-ways K-shots: N classes and K samples each

- N-ways K-shots: N classes and K samples each
- In each task,

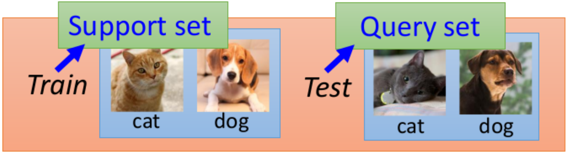
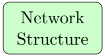


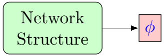
Image credit: Hung-yi Lee

- Focus only on F with same network structure but different initialization ϕ



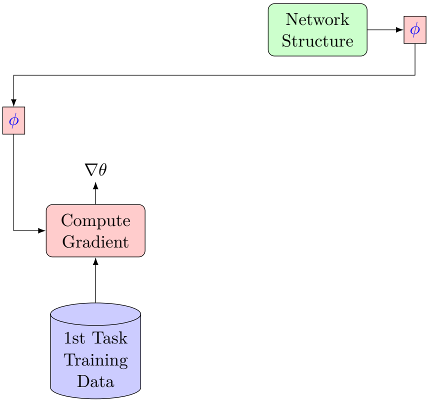
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



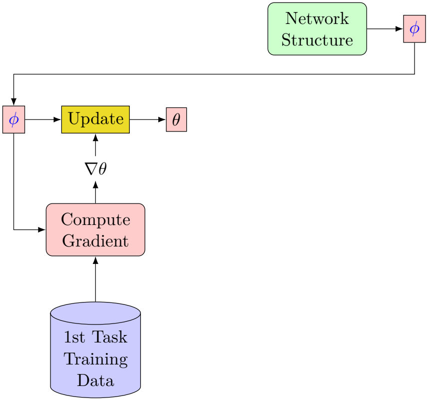
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



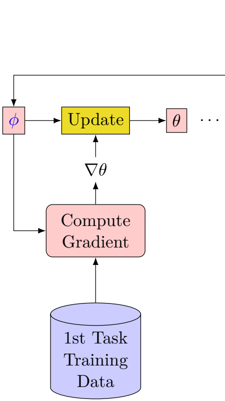
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



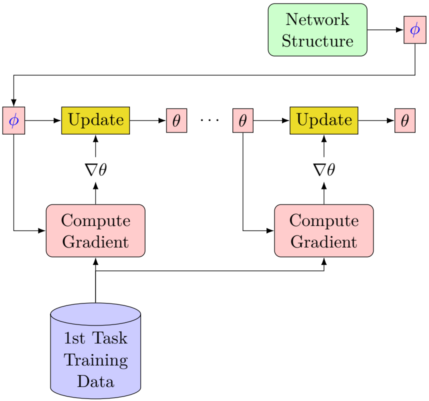
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



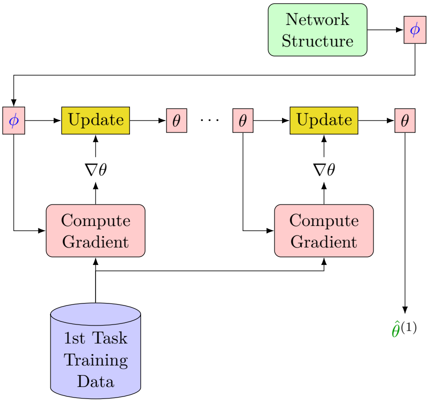
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



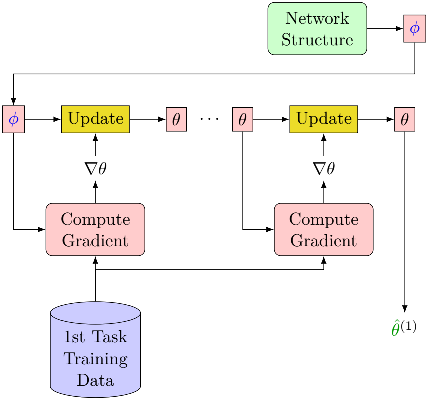
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



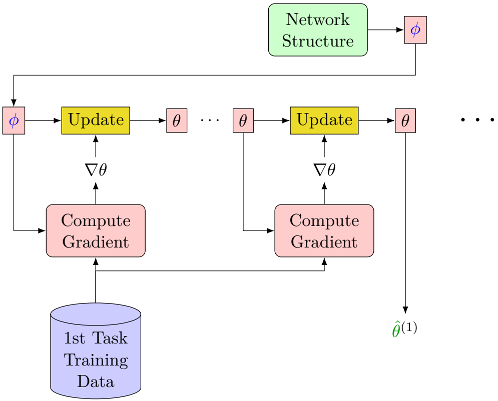
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



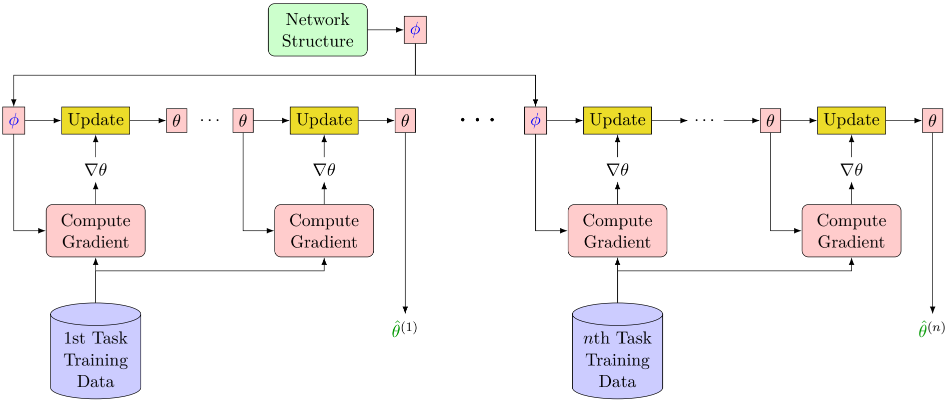
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



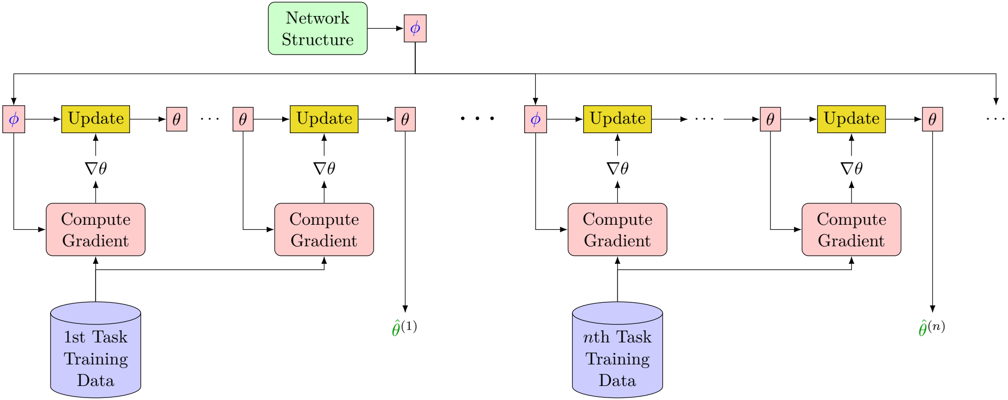
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ



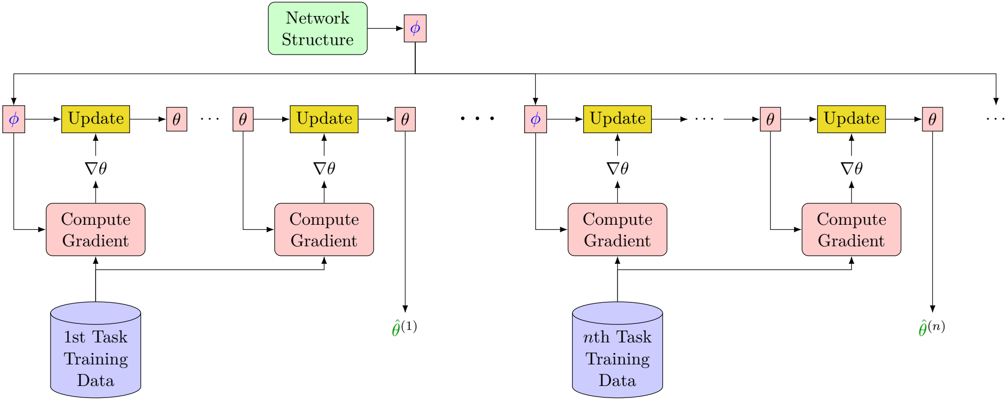
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ
- Minimize $L(\phi) = \sum_n l^{(n)}(\hat{\theta}^{(n)})$



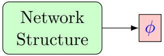
Gradient-based Approach

- Focus only on F with same network structure but different initialization ϕ
- Minimize $L(\phi) = \sum_n l^{(n)}(\hat{\theta}^{(n)})$ (c.f. $L(\phi) = \sum_n l^{(n)}(\phi)$ for pre-training)



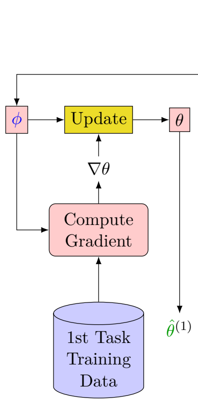
(FO)MAML

- MAML (Model Agnostic Meta Learning): optimize task θ with only 1 gradient update
- FOMAML (First Order MAML): 1st order approximation. Get rid of 2nd order terms



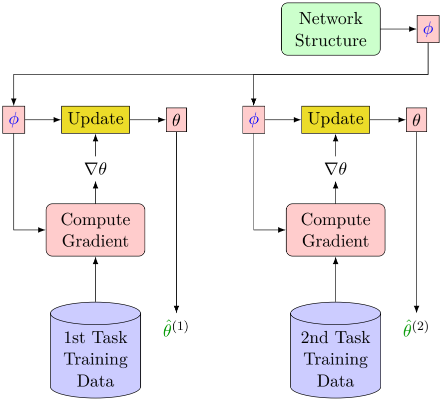
(FO)MAML

- MAML (Model Agnostic Meta Learning): optimize task θ with only 1 gradient update
- FOMAML (First Order MAML): 1st order approximation. Get rid of 2nd order terms



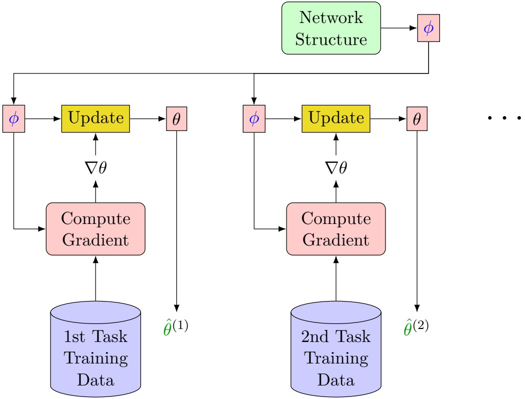
(FO)MAML

- MAML (Model Agnostic Meta Learning): optimize task θ with only 1 gradient update
- FOMAML (First Order MAML): 1st order approximation. Get rid of 2nd order terms



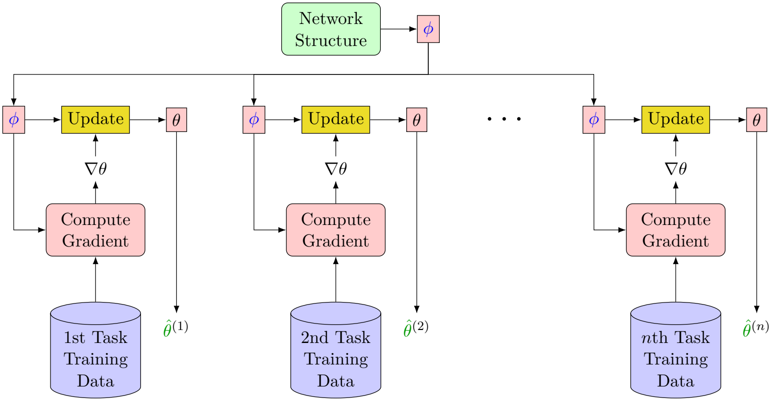
(FO)MAML

- MAML (Model Agnostic Meta Learning): optimize task θ with only 1 gradient update
- FOMAML (First Order MAML): 1st order approximation. Get rid of 2nd order terms



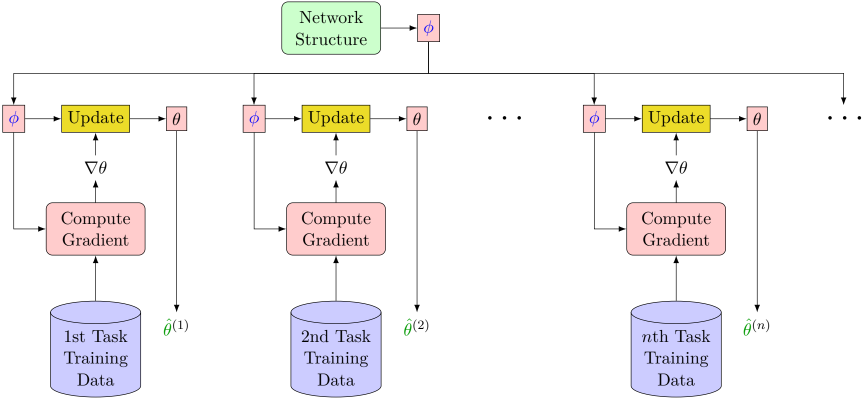
(FO)MAML

- MAML (Model Agnostic Meta Learning): optimize task θ with only 1 gradient update
- FOMAML (First Order MAML): 1st order approximation. Get rid of 2nd order terms



(FO)MAML

- MAML (Model Agnostic Meta Learning): optimize task θ with only 1 gradient update
- FOMAML (First Order MAML): 1st order approximation. Get rid of 2nd order terms



(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_{\phi} l^{(n)}(\phi)$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

$$\frac{\partial l(\hat{\theta})}{\partial \phi_i} = \sum_j \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

$$\frac{\partial l(\hat{\theta})}{\partial \phi_i} = \sum_j \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

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

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_{\phi} l^{(n)}(\phi)$$

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

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

$$\frac{\partial l(\hat{\theta})}{\partial \phi_i} = \sum_j \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

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

$$\begin{cases} i \neq j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} \\ \\ \end{cases}$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

$$\frac{\partial l(\hat{\theta})}{\partial \phi_i} = \sum_j \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

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

$$\begin{cases} i \neq j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = -\epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \\ \end{cases}$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

$$\frac{\partial l(\hat{\theta})}{\partial \phi_i} = \sum_j \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

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

$$\begin{cases} i \neq j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = -\epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 0 \\ i = j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = 1 - \epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 1 \end{cases}$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

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

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

$$\begin{cases} i \neq j : \dfrac{\partial \hat{\theta}_j}{\partial \phi_i} = -\epsilon \dfrac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 0 \\ i = j : \dfrac{\partial \hat{\theta}_j}{\partial \phi_i} = 1 - \epsilon \dfrac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 1 \end{cases}$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

$$\nabla_\phi l(\hat{\theta}) = \begin{pmatrix} \partial l(\hat{\theta})/\partial \phi_1 \\ \partial l(\hat{\theta})/\partial \phi_2 \\ \vdots \\ \partial l(\hat{\theta})/\partial \phi_i \\ \vdots \end{pmatrix} \approx \begin{pmatrix} \partial l(\hat{\theta})/\partial \hat{\theta}_1 \\ \partial l(\hat{\theta})/\partial \hat{\theta}_2 \\ \vdots \\ \partial l(\hat{\theta})/\partial \hat{\theta}_i \\ \vdots \end{pmatrix}$$

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

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

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

$$\begin{cases} i \neq j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = -\epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 0 \\ i = j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = 1 - \epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 1 \end{cases}$$

(FO)MAML

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

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

$$\hat{\theta}^{(n)} = \phi - \epsilon \nabla_\phi l^{(n)}(\phi)$$

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

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

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

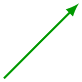
$$\nabla_\phi l(\hat{\theta}) = \begin{pmatrix} \partial l(\hat{\theta})/\partial \phi_1 \\ \partial l(\hat{\theta})/\partial \phi_2 \\ \vdots \\ \partial l(\hat{\theta})/\partial \phi_i \\ \vdots \end{pmatrix} \approx \begin{pmatrix} \partial l(\hat{\theta})/\partial \hat{\theta}_1 \\ \partial l(\hat{\theta})/\partial \hat{\theta}_2 \\ \vdots \\ \partial l(\hat{\theta})/\partial \hat{\theta}_i \\ \vdots \end{pmatrix} = \nabla_{\hat{\theta}} l(\hat{\theta})$$

$$\begin{cases} i \neq j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = -\epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 0 \\ i = j : \frac{\partial \hat{\theta}_j}{\partial \phi_i} = 1 - \epsilon \frac{\partial^2 l(\phi)}{\partial \phi_i \partial \phi_j} \approx 1 \end{cases}$$

(FO)MAML vs Pretraining

MAML

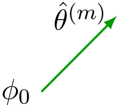
Model Pre-training

ϕ_0 

 : sample from Task m

(FO)MAML vs Pretraining

MAML



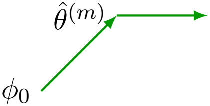
Model Pre-training

→ : sample from Task m

(FO)MAML vs Pretraining

MAML

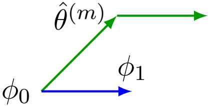
Model Pre-training



→ : sample from Task m

(FO)MAML vs Pretraining

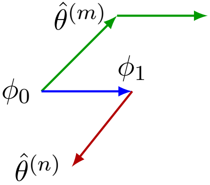
MAML



Model Pre-training

(FO)MAML vs Pretraining

MAML



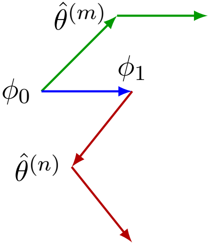
Model Pre-training

→ : sample from Task m

→ : sample from Task n

(FO)MAML vs Pretraining

MAML



Model Pre-training

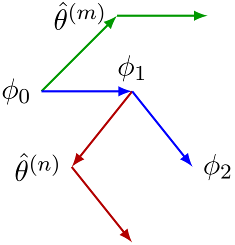
→ : sample from Task m

→ : sample from Task n

(FO)MAML vs Pretraining

MAML

Model Pre-training

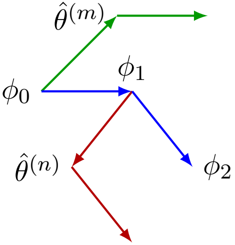


→ : sample from Task m

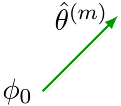
→ : sample from Task n

(FO)MAML vs Pretraining

MAML



Model Pre-training

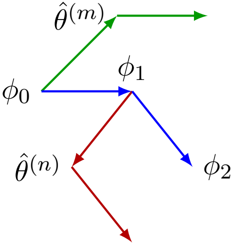


→ : sample from Task m

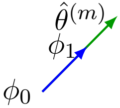
→ : sample from Task n

(FO)MAML vs Pretraining

MAML



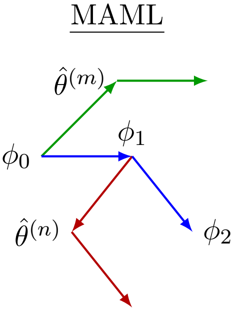
Model Pre-training

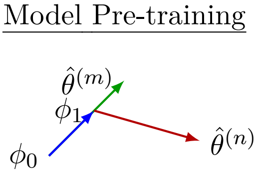


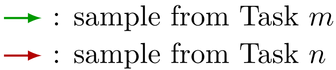
→ : sample from Task m

→ : sample from Task n

(FO)MAML vs Pretraining

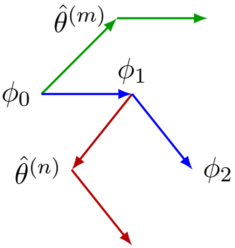




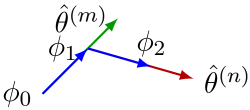


(FO)MAML vs Pretraining

MAML



Model Pre-training



: sample from Task m

: sample from Task n

MAML vs Pre-training

MAML optimizes the potential of ϕ: $L(\phi) = \sum_n l^{(n)}(\hat{\theta}^{(n)})$

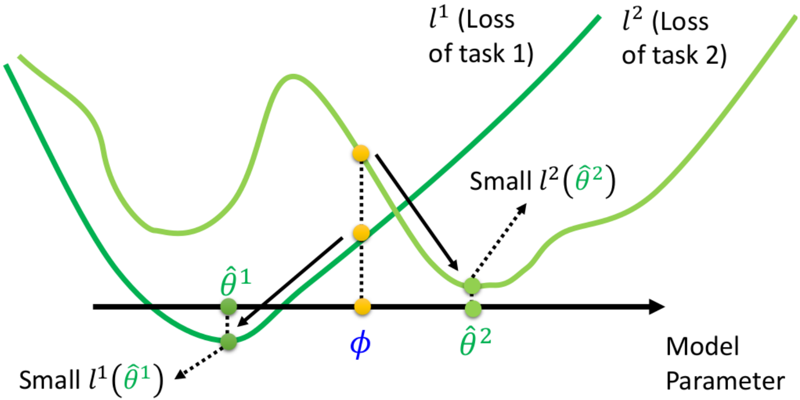


Image credit: Hung-yi Lee

MAML vs Pre-training

Pre-training optimizes the current ϕ for all tasks: $L(\phi) = \sum_n l^{(n)}(\phi)$

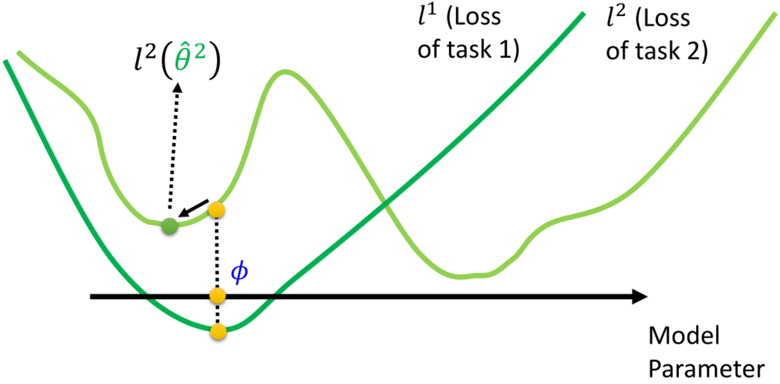


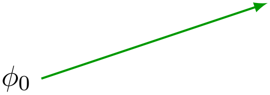
Image credit: Hung-yi Lee

Reptile

ϕ_0

→ : sample from Task m

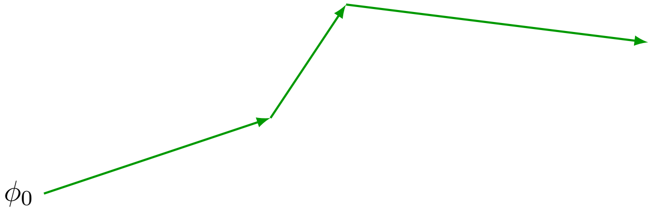
Reptile



→ : sample from Task m

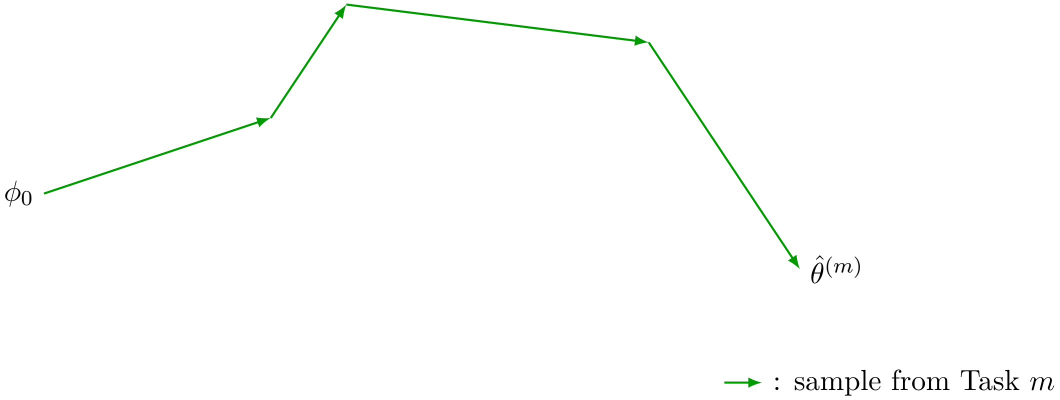
Reptile

Reptile

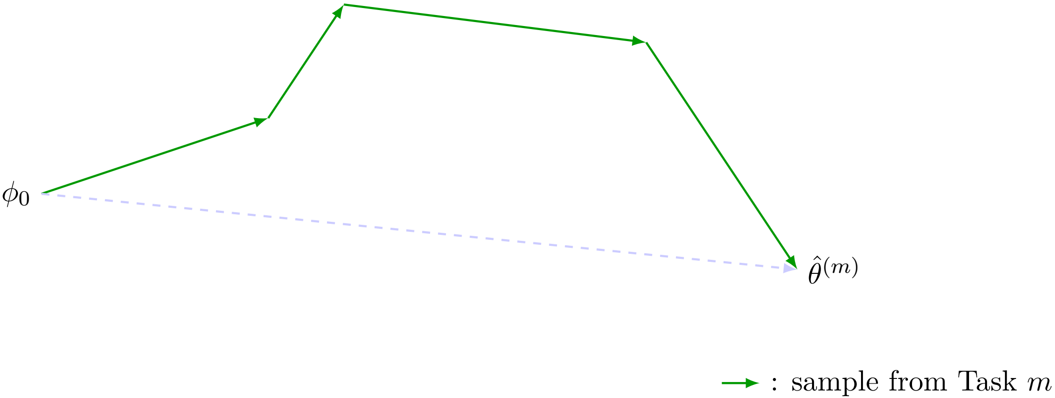


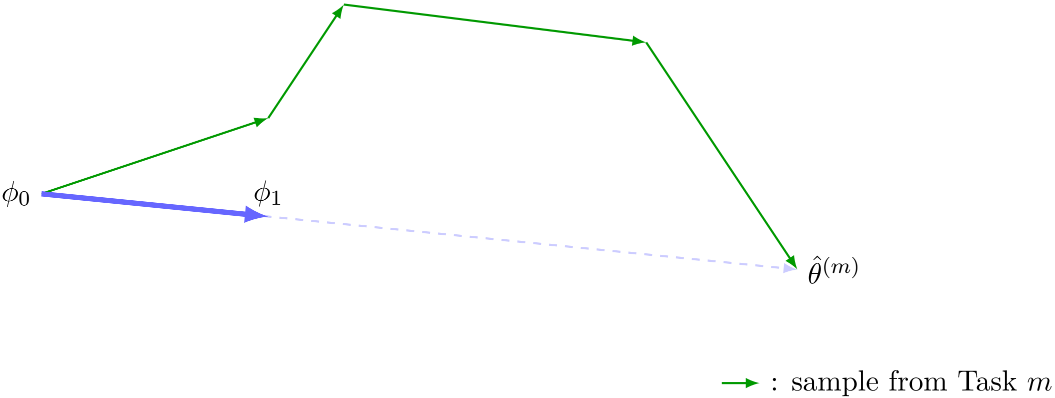
→ : sample from Task m

Reptile

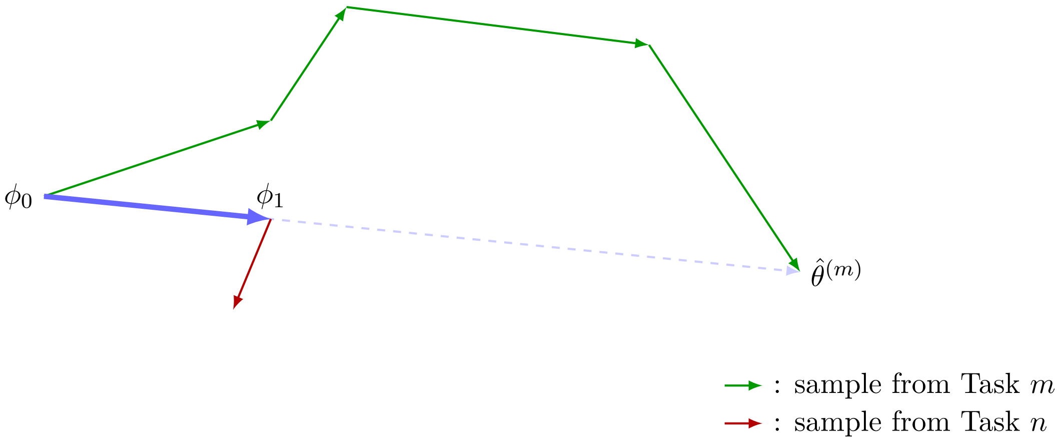


Reptile

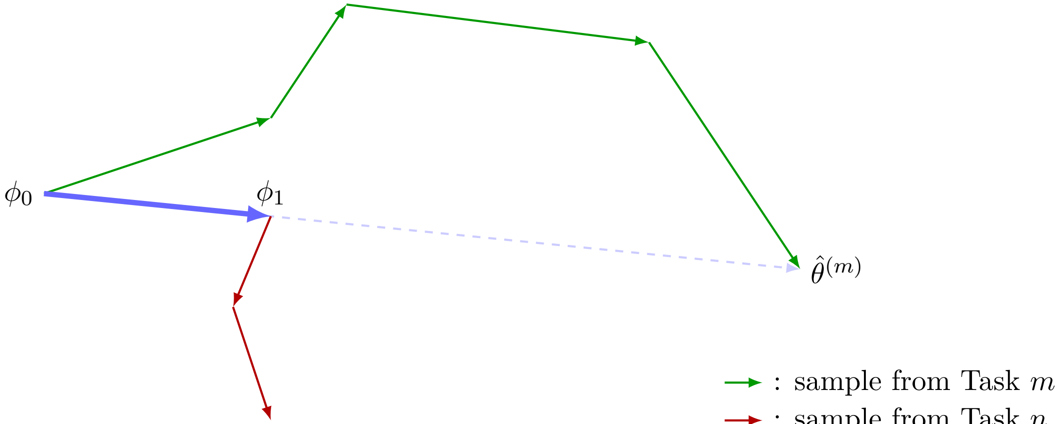



ϕ_0
ϕ_1
$\hat{\theta}^{(m)}$
: sample from Task m

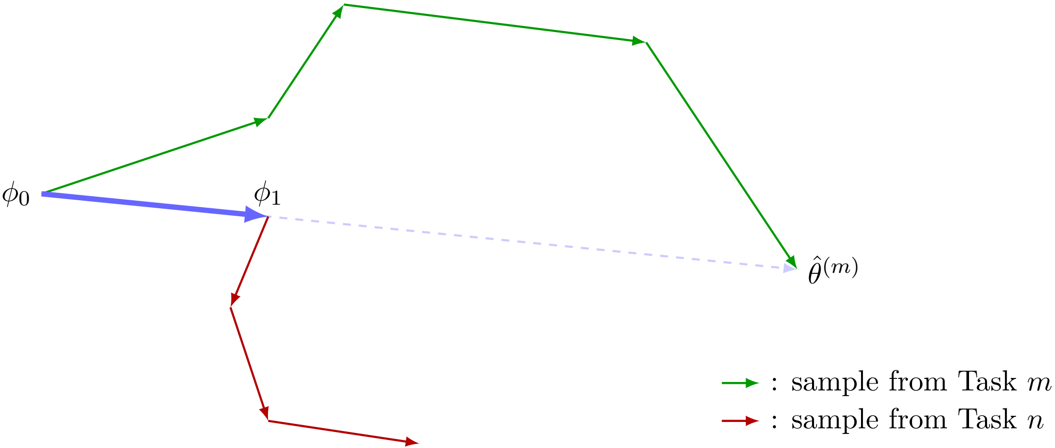
Reptile



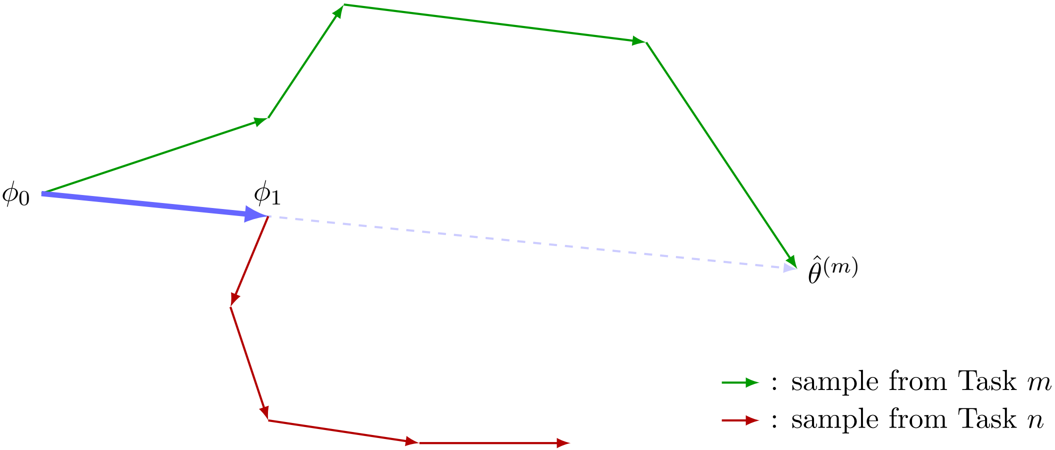
Reptile



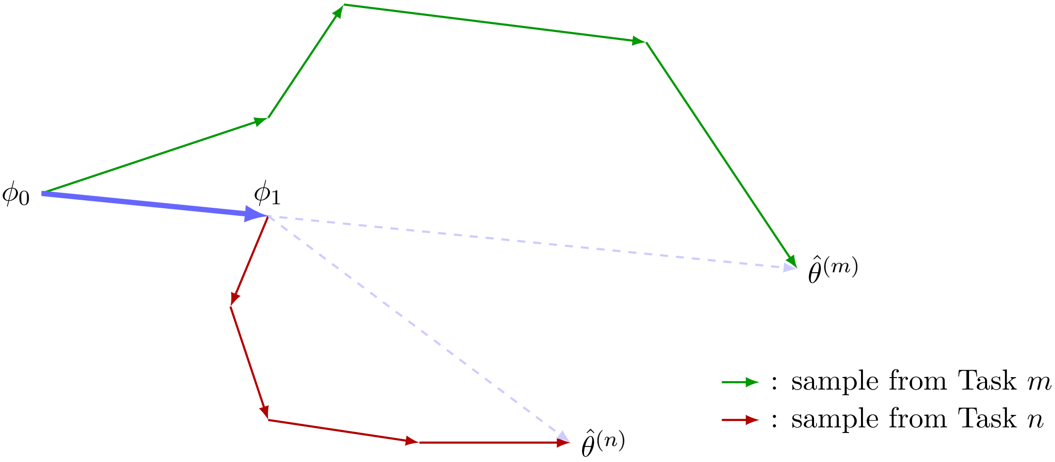
Reptile



Reptile

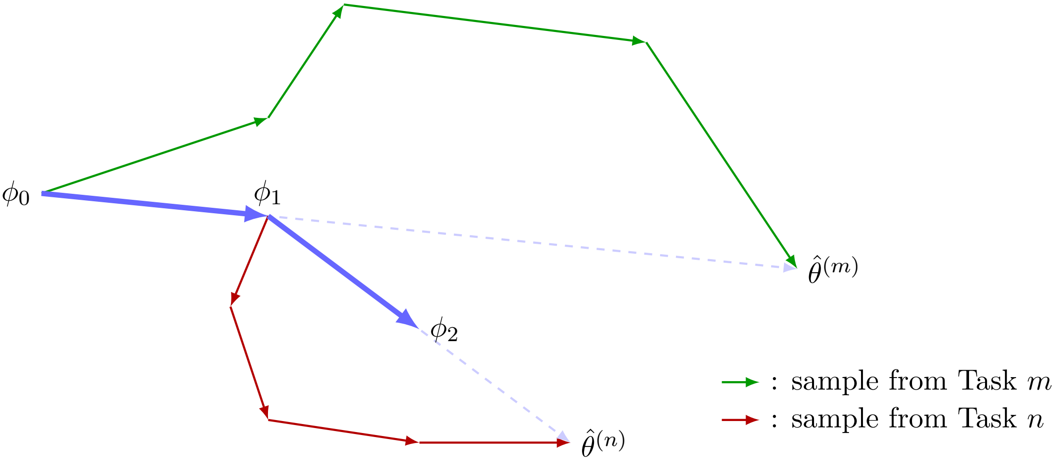


Reptile



ϕ_0
ϕ_1
$\hat{\theta}^{(m)}$
$\hat{\theta}^{(n)}$
: sample from Task m
: sample from Task n

Reptile



ϕ_0
ϕ_1
ϕ_2
$\hat{\theta}^{(m)}$
$\hat{\theta}^{(n)}$
: sample from Task m
: sample from Task n

Reptile Result

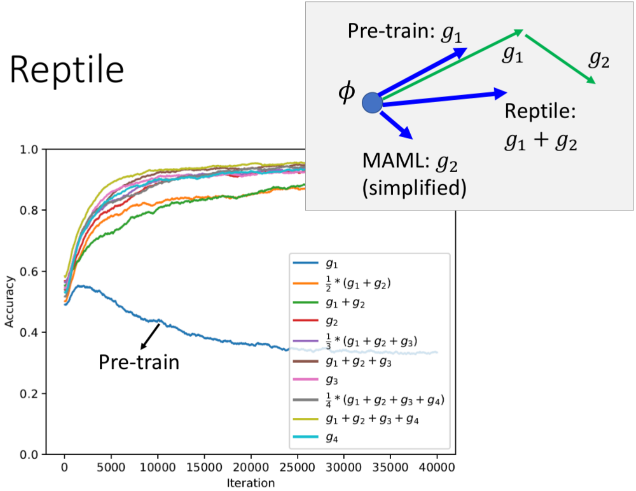


Image credit: Hung-yi Lee

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_{\theta} l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_{\theta} l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_{\theta} l^{(\tau)}(\theta)$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_\theta l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_\theta l^{(\tau)}(\theta)$$

$$\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \phi_i}{\partial \phi_j} - \frac{1}{\lambda}\sum_k \frac{\partial^2 l^{(\tau)}}{\partial \theta_k \partial \theta_i}\frac{\partial \theta_k}{\partial \phi_j}$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_\theta l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_\theta l^{(\tau)}(\theta)$$

$$\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \phi_i}{\partial \phi_j} - \frac{1}{\lambda}\sum_k \frac{\partial^2 l^{(\tau)}}{\partial \theta_k \partial \theta_i}\frac{\partial \theta_k}{\partial \phi_j} \Rightarrow \frac{d\theta}{d\phi} = I - \frac{1}{\lambda}H(l^{(\tau)}(\theta)\frac{d\theta}{d\phi}$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_{\theta} l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_{\theta} l^{(\tau)}(\theta)$$

$$\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \phi_i}{\partial \phi_j} - \frac{1}{\lambda}\sum_k \frac{\partial^2 l^{(\tau)}}{\partial \theta_k \partial \theta_i}\frac{\partial \theta_k}{\partial \phi_j} \Rightarrow \frac{d\theta}{d\phi} = I - \frac{1}{\lambda} H(l^{(\tau)}(\theta)\frac{d\theta}{d\phi} \Rightarrow \frac{d\theta}{d\phi} = \left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)^{-1}$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_\theta l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_\theta l^{(\tau)}(\theta)$$

$$\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \phi_i}{\partial \phi_j} - \frac{1}{\lambda}\sum_k \frac{\partial^2 l^{(\tau)}}{\partial \theta_k \partial \theta_i}\frac{\partial \theta_k}{\partial \phi_j} \Rightarrow \frac{d\theta}{d\phi} = I - \frac{1}{\lambda}H(l^{(\tau)}(\theta))\frac{d\theta}{d\phi} \Rightarrow \frac{d\theta}{d\phi} = \left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)^{-1}$$

Thus, the meta-gradient (from task τ) is

$$\nabla_\phi l^{(\tau)}(\theta) = \frac{d\theta}{d\phi}\nabla_\theta l^{(\tau)}(\theta)$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_\theta l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_\theta l^{(\tau)}(\theta)$$

$$\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \phi_i}{\partial \phi_j} - \frac{1}{\lambda}\sum_k \frac{\partial^2 l^{(\tau)}}{\partial \theta_k \partial \theta_i}\frac{\partial \theta_k}{\partial \phi_j} \Rightarrow \frac{d\theta}{d\phi} = I - \frac{1}{\lambda}H(l^{(\tau)}(\theta))\frac{d\theta}{d\phi} \Rightarrow \frac{d\theta}{d\phi} = \left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)^{-1}$$

Thus, the meta-gradient (from task τ) is

$$\nabla_\phi l^{(\tau)}(\theta) = \frac{d\theta}{d\phi}\nabla_\theta l^{(\tau)}(\theta) = \left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)^{-1}\nabla_\theta l^{(\tau)}(\theta) \triangleq g$$

iMAML: Implicit Gradient

For each task τ, we don't want the model parameter θ too far from the meta-parameter ϕ. Consequently, consider

$$\min_{\theta} l^{(\tau)}(\theta) + \frac{\lambda}{2}\|\theta - \phi\|^2 \Rightarrow \nabla_\theta l^{(\tau)}(\theta) + \lambda(\theta - \phi) = 0 \Rightarrow \theta = \phi - \frac{1}{\lambda}\nabla_\theta l^{(\tau)}(\theta)$$

$$\frac{\partial \theta_i}{\partial \phi_j} = \frac{\partial \phi_i}{\partial \phi_j} - \frac{1}{\lambda}\sum_k \frac{\partial^2 l^{(\tau)}}{\partial \theta_k \partial \theta_i}\frac{\partial \theta_k}{\partial \phi_j} \Rightarrow \frac{d\theta}{d\phi} = I - \frac{1}{\lambda}H(l^{(\tau)}(\theta))\frac{d\theta}{d\phi} \Rightarrow \frac{d\theta}{d\phi} = \left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)^{-1}$$

Thus, the meta-gradient (from task τ) is

$$\nabla_\phi l^{(\tau)}(\theta) = \frac{d\theta}{d\phi}\nabla_\theta l^{(\tau)}(\theta) = \left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)^{-1}\nabla_\theta l^{(\tau)}(\theta) \triangleq g$$

This involves inverting n^2-size matrix, which is infeasible to compute directly. Consider instead

$$\underbrace{\left(I + \frac{H(l^{(\tau)}(\theta))}{\lambda}\right)}_{A} g = \underbrace{\nabla_\theta l^{(\tau)}(\theta)}_{b},$$

which can be solved using conjugate gradient method

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$
- A-conjugate directions are linearly independent if A is positive definite

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$
- A-conjugate directions are linearly independent if A is positive definite
  - Assume $\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k = 0$, $p_j^\top A(\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k) = 0 \Rightarrow p_j^\top A p_j = 0$

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$
- A-conjugate directions are linearly independent if A is positive definite
  - Assume $\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k = 0$, $p_j^\top A(\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k) = 0 \Rightarrow p_j^\top A p_j = 0$
- Let x_0 be an initial estimate of x, such that $Ax = b$. Assume $x - x_0 = \sum_{k=0}^N \alpha_k p_k$,

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$
- A-conjugate directions are linearly independent if A is positive definite
  - Assume $\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k = 0$, $p_j^\top A(\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k) = 0 \Rightarrow p_j^\top A p_j = 0$
- Let x_0 be an initial estimate of x, such that $Ax = b$. Assume $x - x_0 = \sum_{k=0}^{N} \alpha_k p_k$,
  - $p_i^\top A(x - x_0) = \sum_{k=0}^{n} \alpha_k p_i^\top A p_k$

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$
- A-conjugate directions are linearly independent if A is positive definite
  - Assume $\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k = 0$, $p_j^\top A(\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k) = 0 \Rightarrow p_j^\top A p_j = 0$
- Let x_0 be an initial estimate of x, such that $Ax = b$. Assume $x - x_0 = \sum_{k=0}^N \alpha_k p_k$,
  - $p_i^\top A(x - x_0) = \sum_{k=0}^n \alpha_k p_i^\top A p_k = \alpha_i p_i^\top A p_i \Rightarrow \alpha_i = \frac{p_i^\top (b - Ax_0)}{p_i^\top A p_i} = \frac{\langle p_i, b - Ax_0 \rangle}{\langle p_i A p_i \rangle}$
  - Here we use the bracket notation commonly used in physics, $p^\top q = \langle p, q \rangle$ and $p^\top A q = \langle p, Aq \rangle = \langle pAq \rangle$

A-conjugacy

- Let A be $N \times N$
- We say p A-conjugate with q if $p^\top A q = 0$. And denote $p \perp_A q$
- A-conjugate directions are linearly independent if A is positive definite
  - Assume $\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k = 0$, $p_j^\top A(\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_k p_k) = 0 \Rightarrow p_j^\top A p_j = 0$
- Let x_0 be an initial estimate of x, such that $Ax = b$. Assume $x - x_0 = \sum_{k=0}^{N} \alpha_k p_k$,
  - $p_i^\top A(x - x_0) = \sum_{k=0}^{n} \alpha_k p_i^\top A p_k = \alpha_i p_i^\top A p_i \Rightarrow \alpha_i = \frac{p_i^\top (b - Ax_0)}{p_i^\top A p_i} = \frac{\langle p_i, b - Ax_0 \rangle}{\langle p_i A p_i \rangle}$
  - Here we use the bracket notation commonly used in physics, $p^\top q = \langle p, q \rangle$ and $p^\top A q = \langle p, Aq \rangle = \langle pAq \rangle$
- If we can keep generating conjugate directions p_k, we can find the solution x that satisfies $Ax = b$ by simply computing $\alpha_i = \frac{\langle p_i, b - Ax_0 \rangle}{\langle p_i A p_i \rangle}$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$
\begin{aligned}
r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\
r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\
r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\
& & \cdots & &
\end{aligned}
$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\dots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k = Ax_k - b$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$

$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$

$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$

$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k = Ax_{k-1} + \alpha_{k-1} A p_{k-1} - b$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\ldots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k = Ax_{k-2} + \alpha_{k-2}Ap_{k-2} + \alpha_{k-1}Ap_{k-1} - b$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\dots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k = Ax_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1} - b$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$

$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$

$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$

$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \ = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$.

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \ = -p_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
    - For $k = 0$, $p_1 = -r_1 + \beta_0 p_0$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \ = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $p_1 = b - Ax_1 + \beta_0 p_0$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$

$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$

$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$

$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \;\; = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $\quad p_1 = b - Ax_0 - \alpha_0 A p_0 + \beta_0 p_0$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \ = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $\quad p_1 = p_0 - \alpha_0 A p_0 + \beta_0 p_0$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \; = -p_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $\quad p_1 = p_0 - \alpha_0 Ap_0 + \beta_0 p_0 \;\Rightarrow Ap_0 = \frac{1}{\alpha_0}[p_0 + \beta_0 p_0 - p_1] \in \text{span}\{p_0, p_1\}$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$
\begin{aligned}
r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\
r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\
r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\
& & \cdots & &
\end{aligned}
$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \ = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $\quad p_1 = p_0 - \alpha_0 A p_0 + \beta_0 p_0 \ \Rightarrow A p_0 = \frac{1}{\alpha_0}[p_0 + \beta_0 p_0 - p_1] \in \text{span}\{p_0, p_1\}$
  - Assume $Ap_{k-1} \in \text{span}\{p_0, p_1, \cdots, p_k\}$,

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$
\begin{aligned}
r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\
r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\
r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\
&& \cdots &&&
\end{aligned}
$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \; = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $\quad p_1 = p_0 - \alpha_0 A p_0 + \beta_0 p_0 \;\Rightarrow A p_0 = \frac{1}{\alpha_0}[p_0 + \beta_0 p_0 - p_1] \in \text{span}\{p_0, p_1\}$
  - Assume $Ap_{k-1} \in \text{span}\{p_0, p_1, \cdots, p_k\}$, $p_{k+1} = -r_{k+1} + \beta_k p_k$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$
\begin{aligned}
r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\
r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\
r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\
& & \cdots & &
\end{aligned}
$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k = -p_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $p_1 = p_0 - \alpha_0 Ap_0 + \beta_0 p_0 \Rightarrow Ap_0 = \frac{1}{\alpha_0}[p_0 + \beta_0 p_0 - p_1] \in \text{span}\{p_0, p_1\}$
  - Assume $Ap_{k-1} \in \text{span}\{p_0, p_1, \cdots, p_k\}$, $p_{k+1} = p_0 - \sum_{i=0}^{k-1} \alpha_i Ap_i - \alpha_k Ap_k + \beta_k p_k$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$
\begin{aligned}
r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 & (1)\\
r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 & (2)\\
r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 & (3)\\
& & \cdots & & &
\end{aligned}
$$

- From the middle equations, $\{p_0, p_1, \cdots, p_n\}$ and $\{r_0, r_1, \cdots, r_n\}$ span the same space
- $r_k \quad = -p_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1}$.
- $Ap_k \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$ for $k \geq 0$. We can show that with induction
  - For $k = 0$, $\quad p_1 = p_0 - \alpha_0 Ap_0 + \beta_0 p_0 \Rightarrow Ap_0 = \frac{1}{\alpha_0}[p_0 + \beta_0 p_0 - p_1] \in \text{span}\{p_0, p_1\}$
  - Assume $Ap_{k-1} \in \text{span}\{p_0, p_1, \cdots, p_k\}$, $p_{k+1} = p_0 - \sum_{i=0}^{k-1} \alpha_i Ap_i - \alpha_k Ap_k + \beta_k p_k$ $\Rightarrow Ap_k = \frac{1}{\alpha_k}[p_0 - \sum_{i=0}^{k-1} \alpha_i Ap_i + \beta_k p_k - p_{k+1}] \in \text{span}\{p_0, p_1, \cdots, p_{k+1}\}$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$ and $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$
\begin{aligned}
r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\
r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\
r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\
& & \cdots & &
\end{aligned}
$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$ and $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$

- From the choice of α_n, we see that $r_n \to 0$ and $x_n \to x$ as long as $p_i \perp_A p_j, i \neq j$

Conjugate gradient method

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$ and $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$

- From the choice of α_n, we see that $r_n \to 0$ and $x_n \to x$ as long as $p_i \perp_A p_j, i \neq j$
- We will show that in the next several slides with induction, note that we also have $p_i \perp r_j$ for $i < j$. It is convenient to show them together

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0\rangle}{\langle p_n A p_n\rangle} = \frac{\langle p_n, p_0\rangle}{\langle p_n A p_n\rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n\rangle}{\langle p_n A p_n\rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- From the choice of β, we have $\langle p_{k+1} A p_k\rangle = \langle -r_{k+1} + \beta_k p_k, Ap_k\rangle$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$

$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$

$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$

$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- From the choice of β, we have $\langle p_{k+1} A p_k \rangle = \langle -r_{k+1} + \frac{\langle r_{k+1} A p_k \rangle}{\langle p_k A p_k \rangle} p_k, A p_k \rangle = 0$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0\rangle}{\langle p_n A p_n\rangle} = \frac{\langle p_n, p_0\rangle}{\langle p_n A p_n\rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n\rangle}{\langle p_n A p_n\rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- From the choice of β, we have $\langle p_{k+1} A p_k\rangle = \langle -r_{k+1} + p_k, Ap_k\rangle = 0 \Rightarrow p_{k+1} \perp_A p_k$. In particular, $p_1 \perp_A p_0$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- From the choice of β, we have $\langle p_{k+1} A p_k \rangle = \langle -r_{k+1} + p_k, A p_k \rangle = 0 \Rightarrow p_{k+1} \perp_A p_k$. In particular, $p_1 \perp_A p_0$
- And $\langle p_0, r_1 \rangle = \langle p_0, p_0 - \alpha_0 A p_0 \rangle$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- From the choice of β, we have $\langle p_{k+1} A p_k \rangle = \langle -r_{k+1} + p_k, Ap_k \rangle = 0 \Rightarrow p_{k+1} \perp_A p_k$. In particular, $p_1 \perp_A p_0$
- And $\langle p_0, r_1 \rangle = \langle p_0, p_0 - \frac{\langle p_0, p_0 \rangle}{\langle p_0 A p_0 \rangle} A p_0 \rangle$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$

$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$

$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$

$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n Ap_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n Ap_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} Ap_n \rangle}{\langle p_n Ap_n \rangle}$ and $r_k = -p_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1}$

- From the choice of β, we have $\langle p_{k+1} Ap_k \rangle = \langle -r_{k+1} + p_k, Ap_k \rangle = 0 \Rightarrow p_{k+1} \perp_A p_k$. In particular, $p_1 \perp_A p_0$
- And $\langle p_0, r_1 \rangle = \langle p_0, p_0 - \frac{\langle p_0, p_0 \rangle}{\langle p_0 Ap_0 \rangle} Ap_0 \rangle = 0$. Thus, $p_0 \perp r_1$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n, Ap_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n, Ap_n \rangle}$, $\beta_n = \frac{\langle r_{n+1}, Ap_n \rangle}{\langle p_n, Ap_n \rangle}$ and $r_k = -p_0 + \alpha_0 Ap_0 + \alpha_1 Ap_1 + \cdots + \alpha_{k-1} Ap_{k-1}$

- For $i < j \le k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
  - $\langle p_i, r_{k+1} \rangle = \langle p_i, -p_0 + \alpha_0 A p_0 + \cdots + \alpha_k A p_k \rangle$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$\begin{aligned} r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 && (1)\\ r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 && (2)\\ r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 && (3)\\ & & \cdots \end{aligned}$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0\rangle}{\langle p_n A p_n\rangle} = \frac{\langle p_n, p_0\rangle}{\langle p_n A p_n\rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n\rangle}{\langle p_n A p_n\rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \le k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
    - $\langle p_i, r_{k+1}\rangle = -\langle p_i, p_0\rangle + \alpha_i \langle p_i A p_i\rangle$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$\begin{aligned} r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1) \\ r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2) \\ r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3) \\ & & \cdots & & & \end{aligned}$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \le k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
  - $\langle p_i, r_{k+1} \rangle = 0$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$\begin{aligned} r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\ r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\ r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\ & & \cdots & & & \end{aligned}$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \le k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown
    $p_{k+1} = -r_{k+1} + \beta_k p_k$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown

    $p_{k+1} = -r_{k+1} - \beta_k r_k + \beta_k \beta_{k-1} p_{k-1}$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$\begin{aligned} r_0 &= Ax_0 - b, & p_0 &= -r_0, & x_1 &= x_0 + \alpha_0 p_0 \quad (1)\\ r_1 &= Ax_1 - b, & p_1 &= -r_1 + \beta_0 p_0, & x_2 &= x_1 + \alpha_1 p_1 \quad (2)\\ r_2 &= Ax_2 - b, & p_2 &= -r_2 + \beta_1 p_1, & x_3 &= x_2 + \alpha_2 p_2 \quad (3)\\ & & \cdots & & & \end{aligned}$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$,
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown

    $$p_{k+1} = -r_{k+1} - \beta_k r_k - \beta_k \beta_{k-1} r_{k-1} - \cdots - \beta_k \beta_{k-1} \cdots \beta_{i+1} r_{i+1} + \beta_k \beta_{k-1} \cdots \beta_i p_i$$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \quad p_0 = -r_0, \quad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$

$$r_1 = Ax_1 - b, \quad p_1 = -r_1 + \beta_0 p_0, \quad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$

$$r_2 = Ax_2 - b, \quad p_2 = -r_2 + \beta_1 p_1, \quad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$

$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$, ($Ap_i \in \text{span}\{p_0, p_1, \cdots, p_{i+1}\}$)
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown

$$p_{k+1} = -r_{k+1} - \beta_k r_k - \beta_k \beta_{k-1} r_{k-1} - \cdots - \beta_k \beta_{k-1} \cdots \beta_{i+1} r_{i+1} + \beta_k \beta_{k-1} \cdots \beta_i p_i$$

$$\Rightarrow \langle p_i A p_{k+1} \rangle \stackrel{?}{=} -(\beta_k \beta_{k-1} \cdots \beta_{i+1})(\langle p_i A r_{i+1} \rangle - \beta_i \langle p_i A p_i \rangle)$$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \tag{1}$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \tag{2}$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \tag{3}$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k + 1$, ($Ap_i \in \text{span}\{p_0, p_1, \cdots, p_{i+1}\}$)
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown

    $p_{k+1} \quad = -r_{k+1} - \beta_k r_k - \beta_k \beta_{k-1} r_{k-1} - \cdots - \beta_k \beta_{k-1} \cdots \beta_{i+1} r_{i+1} + \beta_k \beta_{k-1} \cdots \beta_i p_i$

    $\Rightarrow \langle p_i A p_{k+1} \rangle \stackrel{?}{=} -(\beta_k \beta_{k-1} \cdots \beta_{i+1})(\langle p_i A r_{i+1} \rangle - \beta_i \langle p_i A p_i \rangle) = 0$

Conjugacy proof

Consider an initial guess x_0 for the problem $Ax = b$. And iterate as follow

$$r_0 = Ax_0 - b, \qquad p_0 = -r_0, \qquad x_1 = x_0 + \alpha_0 p_0 \qquad (1)$$
$$r_1 = Ax_1 - b, \qquad p_1 = -r_1 + \beta_0 p_0, \qquad x_2 = x_1 + \alpha_1 p_1 \qquad (2)$$
$$r_2 = Ax_2 - b, \qquad p_2 = -r_2 + \beta_1 p_1, \qquad x_3 = x_2 + \alpha_2 p_2 \qquad (3)$$
$$\cdots$$

where $\alpha_n = \frac{\langle p_n, b - Ax_0 \rangle}{\langle p_n A p_n \rangle} = \frac{\langle p_n, p_0 \rangle}{\langle p_n A p_n \rangle}$, $\beta_n = \frac{\langle r_{n+1} A p_n \rangle}{\langle p_n A p_n \rangle}$ and $r_k = -p_0 + \alpha_0 A p_0 + \alpha_1 A p_1 + \cdots + \alpha_{k-1} A p_{k-1}$

- For $i < j \leq k$, assume that we have $p_i \perp_A p_j$ and $p_i \perp r_j$
- For $i < j = k+1$, ($Ap_i \in \text{span}\{p_0, p_1, \cdots, p_{i+1}\}$)
  - $\langle p_i, r_{k+1} \rangle = 0$
  - Assume $i < k$ as $p_k \perp_A p_{k+1}$ was already shown
    $$p_{k+1} = -r_{k+1} - \beta_k r_k - \beta_k \beta_{k-1} r_{k-1} - \cdots - \beta_k \beta_{k-1} \cdots \beta_{i+1} r_{i+1} + \beta_k \beta_{k-1} \cdots \beta_i p_i$$
    $$\Rightarrow \langle p_i A p_{k+1} \rangle \stackrel{?}{=} -(\beta_k \beta_{k-1} \cdots \beta_{i+1})(\langle p_i A r_{i+1} \rangle - \beta_i \langle p_i A p_i \rangle) = 0$$
  - Thus, by induction, $p_i \perp_A p_j$ (and $p_i \perp r_j$) for all $i < j$

Remark

- Equivalent of minimizing $f(x) = \frac{1}{2}x^\top Ax - x^\top b + c$

Remark

- Equivalent of minimizing $f(x) = \frac{1}{2}x^\top Ax - x^\top b + c$
- Given initial $x = x_0$, the gradient is $Ax_0 - b \propto p_0$. The remaining search directions are all conjugate to p_0. Thus the name conjugate gradient

Remark

- Equivalent of minimizing $f(x) = \frac{1}{2}x^\top Ax - x^\top b + c$
- Given initial $x = x_0$, the gradient is $Ax_0 - b \propto p_0$. The remaining search directions are all conjugate to p_0. Thus the name conjugate gradient
- There is no need to actually compute the Hessian, the update only involves Hessian-vector product Hv, we can compute

$$Hv = \left.\frac{d}{dt}\right|_{t=0} \nabla f(x + tv)$$

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update
  - Reptile: use multiple inner updates but only cares the final state

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update
  - Reptile: use multiple inner updates but only cares the final state
  - iMAML: can use any methods to optimize task model. Use CG to solve meta-gradient effectively

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update
  - Reptile: use multiple inner updates but only cares the final state
  - iMAML: can use any methods to optimize task model. Use CG to solve meta-gradient effectively
- Other meta learning approaches

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update
  - Reptile: use multiple inner updates but only cares the final state
  - iMAML: can use any methods to optimize task model. Use CG to solve meta-gradient effectively
- Other meta learning approaches
  - Optimization-based approach: learn optimizer directly (e.g., learning to learn by gradient descent by gradient descent, use RNN to learn task-specific update rules for optimizer)

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update
  - Reptile: use multiple inner updates but only cares the final state
  - iMAML: can use any methods to optimize task model. Use CG to solve meta-gradient effectively
- Other meta learning approaches
  - Optimization-based approach: learn optimizer directly (e.g., learning to learn by gradient descent by gradient descent, use RNN to learn task-specific update rules for optimizer)
  - Metric-based approach: learn metric useful for classification

Conclusion

- Meta Learning: learn to learn
  - Learn shared meta-knowledge among different tasks
  - Often associated with few-shot learning
- Gradient-based approach
  - Optimize learning algorithm (parametrized by initial parameter) with gradients directly
  - Meta-level learning (outer loop): learn meta-knowledge (initial parameter)
  - Task-level learning (inner loop): learn task specific parameter
  - MAML: only one inner loop update
  - Reptile: use multiple inner updates but only cares the final state
  - iMAML: can use any methods to optimize task model. Use CG to solve meta-gradient effectively
- Other meta learning approaches
  - Optimization-based approach: learn optimizer directly (e.g., learning to learn by gradient descent by gradient descent, use RNN to learn task-specific update rules for optimizer)
  - Metric-based approach: learn metric useful for classification
  - Model-based approach: design network that avoids overfitting

Reference

- Hung-yi Lee's MAML lecture
- An Interactive Introduction to Model-Agnostic Meta-Learning