Meta-Learning

Nikita Konodyuk

Higher School of Ecomonics

What is meta-learning

- Any technique applying machine learning to the process of learning
- Examples:
  - one/few-shot learning
  - algorithm selection based on dataset metadata
  - trainable optimizers
  - ...

What is meta-learning

- Any technique applying machine learning to the process of learning

- Examples:
    - **one/few-shot learning**
    - algorithm selection based on dataset metadata
    - **trainable optimizers**
    - ...

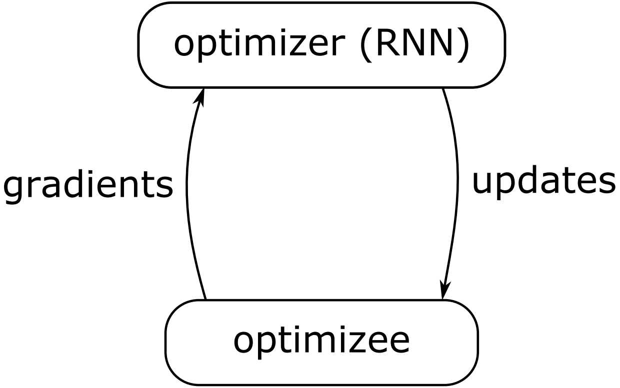
Manually designed optimizers

- $w^{(t)} = w^{(t-1)} - \eta_t \boldsymbol{\nabla}_w L\big(w^{(t-1)}\big)$
- $w^{(t)} = w^{(t-1)} - h_k\big(w^{(t-1)}\big)$
  where $\quad h_k = \alpha h_{k-1} + \eta_t \boldsymbol{\nabla}_w L\big(w^{(t-1)}\big)$
- $w_j^{(t)} = w_j^{(t-1)} - \frac{\eta_t}{\sqrt{G_{tj}+\varepsilon}}\Big(\boldsymbol{\nabla}_w L\big(w^{(t-1)}\big)\Big)_j$
  where $\quad G_{tj} = G_{t-1,j} + \Big(\boldsymbol{\nabla}_w L\big(w^{(t-1)}\big)\Big)_j^2$

Generic optimizer

$$w^{(t)} = w^{(t-1)} - g_t\left(\boldsymbol{\nabla}_w L\left(w^{(t-1)}\right)\right)$$

Learning to learn by gradient descent by gradient descent



Optimizer's loss

- $\mathcal{L}(\cdot)$: optimiz**er**'s loss function
- ϕ: optimiz**er**'s parameters
- $L(\cdot)$: optimiz**ee**'s loss function
- w^*: optimiz**ee**'s final parameters

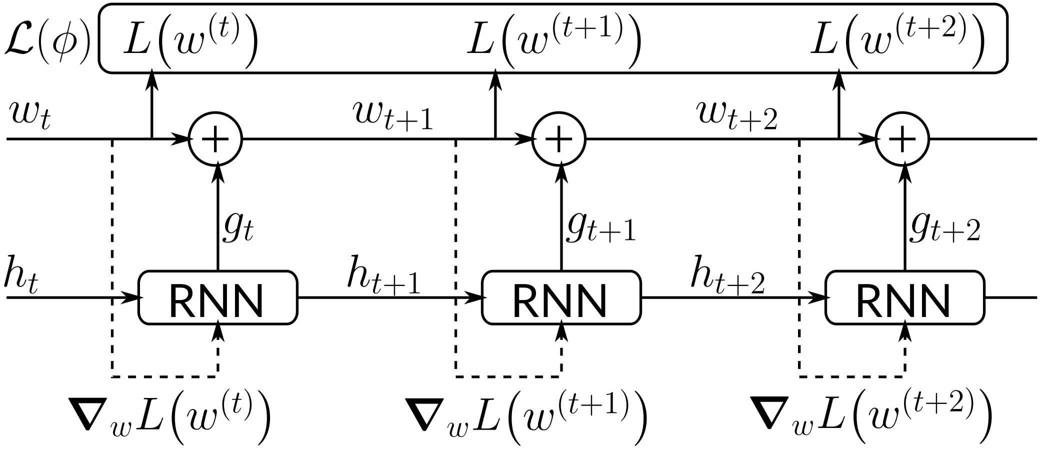
$$\mathcal{L}(\phi) = \mathbb{E}[L(w^*)]$$

Optimizer's loss

- $\mathcal{L}(\cdot)$: optimiz**er**'s loss function
- ϕ: optimiz**er**'s parameters
- $L(\cdot)$: optimiz**ee**'s loss function
- w^*: optimiz**ee**'s final parameters

$$\cancel{\mathcal{L}(\phi) = \mathbb{E}[L(w^*)]} \quad \mathcal{L}(\phi) = \mathbb{E}\left[\sum_{t=0}^{T} L\Big(w^{(t)}\Big)\right]$$

Computational graph



Scaling issues

- Number of optimiz**er**’s parameters grows too fast if RNN is fully-connected

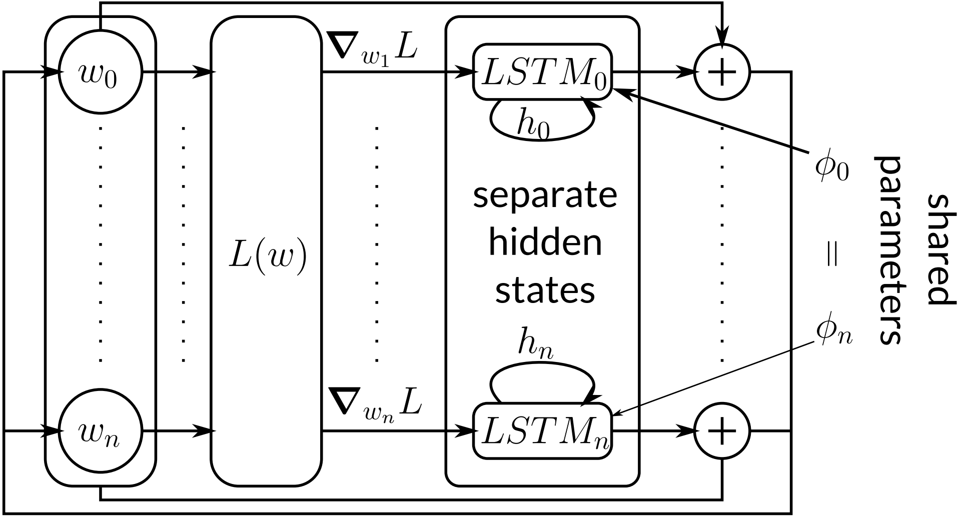
Scaling issues

- Number of optimiz**er**'s parameters grows too fast if RNN is fully-connected
- Need to reduce the growth speed

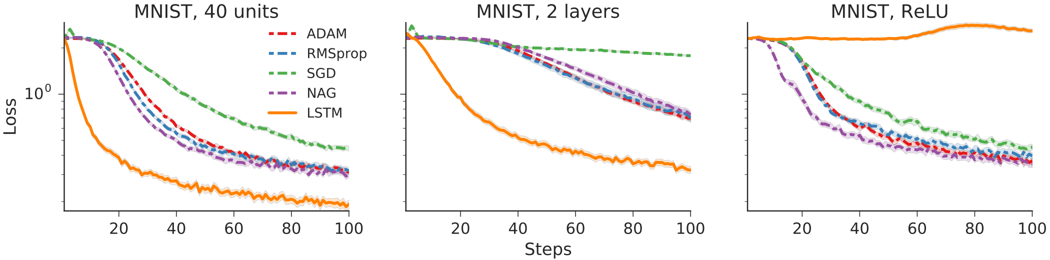
Scaling issues

- Number of optimiz**er**'s parameters grows too fast if RNN is fully-connected
- Need to reduce the growth speed
- Solution: use per-coordinate RNN

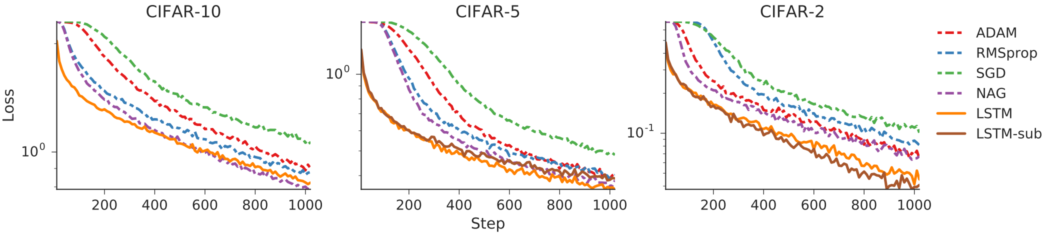
Coordinatewise LSTM



Benchmarks: MNIST



Benchmarks: CIFAR



Approach drawbacks

- Too many parameters

MAML/FOMAML/Reptile

- Do not increase the number of parameters
- Can be applied to any model optimized by gradient descent

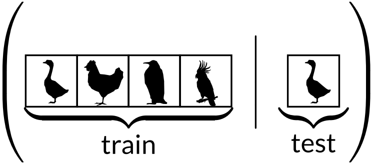
MAML/FOMAML/Reptile

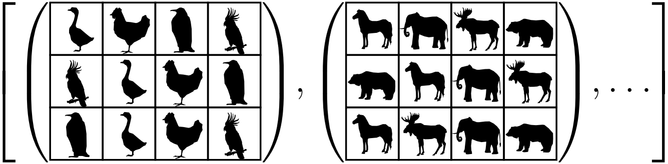
- Do not increase the number of parameters
- Can be applied to any model optimized by gradient descent
- Try to find an optimal initial point for gradient descent

MAML/FOMAML/Reptile

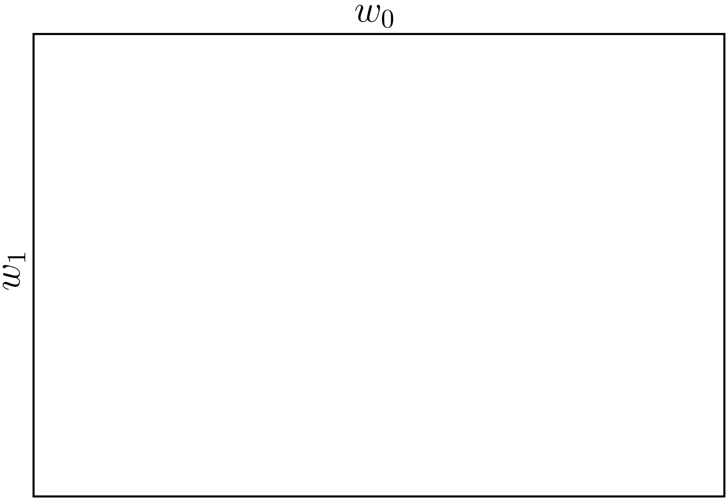
- **Learning** —optimizing the initial point
- **Adaptation** —gradient descent from the learned initial point for a particular task

Example: one/few-shot learning

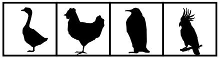
- The format of evaluation tasks: 

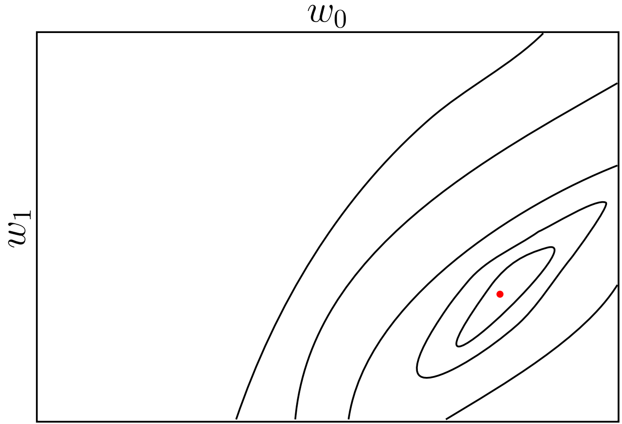
- Dataset of tasks: 

Parameter space

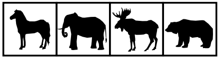


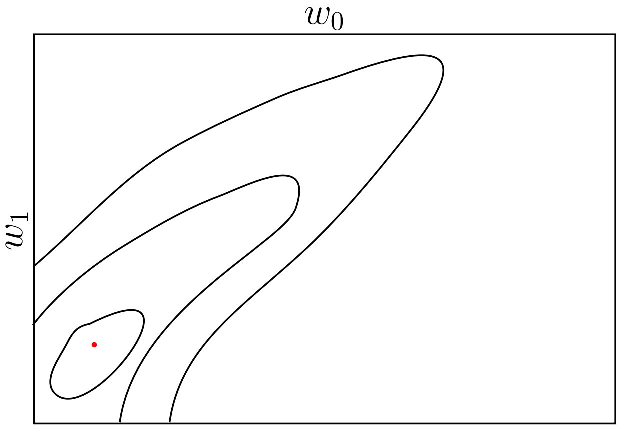
Task #1: birds



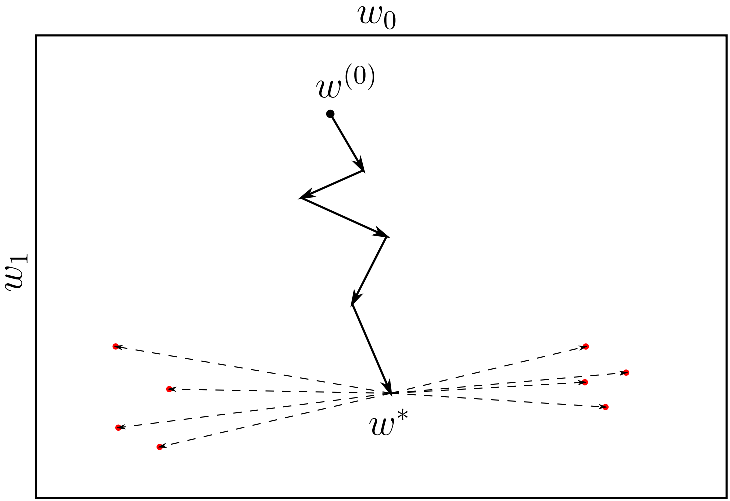


Task #2: wild animals





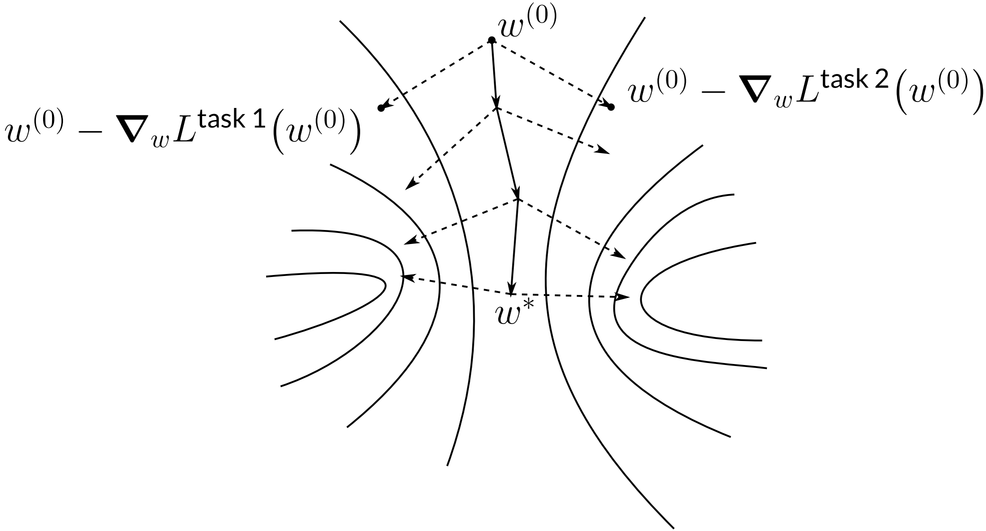
Meta-learning by gradient descent



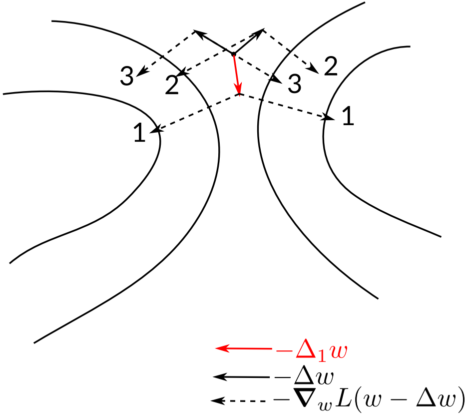
Model-agnostic Meta Learning (MAML)

$w \leftarrow$ init
for $batch$ **of** $tasks$ **do**
  for $task \in batch$ **do**
    $\mathcal{D}_{train}, \mathcal{D}_{test} \leftarrow$ sample$(task)$
    $w_{new} \leftarrow w - \nabla_w L(w, \mathcal{D}_{train})$
    $grad \leftarrow grad + \nabla_w L(w_{new}, \mathcal{D}_{test})$
  end for
  $w \leftarrow w - \alpha \cdot grad$
end for

MAML: Intuition

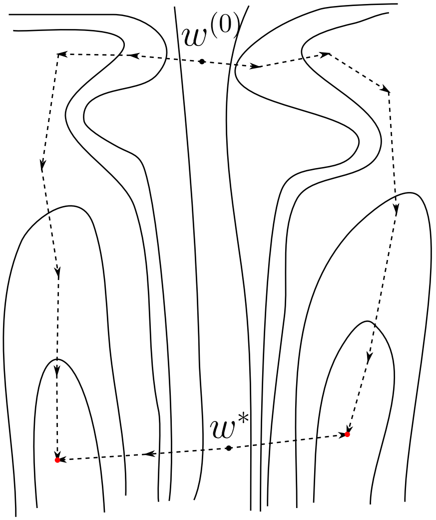


MAML: 2nd derivative intuition



$$\sum L(1) < \sum L(3) < \sum L(2) \Rightarrow \nabla_w L(w - \nabla_w(L(w))) = \Delta_1 w$$

Why can't we just step by antigradient?



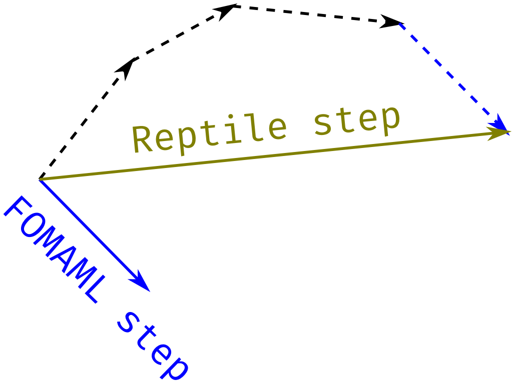
First-order MAML (FOMAML)

$w \leftarrow$ init
for $batch$ **of** $tasks$ **do**
    for $task \in batch$ **do**
        $\mathcal{D}_{train}, \mathcal{D}_{test} \leftarrow$ sample$(task)$
        $w_{new} \leftarrow w$
        for $j = 1 \rightarrow K$ **do**
            $w_{new} \leftarrow w_{new} - \boldsymbol{\nabla}_{w_{new}} L(w_{new}, \mathcal{D}_{train})$
        end for
        $grad \leftarrow grad + \boldsymbol{\nabla}_{w_{new}} L(w_{new}, \mathcal{D}_{test})$
    end for
    $w \leftarrow w - \alpha \cdot grad$
end for

Reptile

$w \leftarrow$ init
for $i = 1 \rightarrow N$ **do**
  $task \leftarrow$ choice$(tasks)$
  $w_{new} \leftarrow w$
  for $j = 1 \rightarrow K$ **do**
    $\mathcal{D} \leftarrow$ sample$(task)$
    $w_{new} \leftarrow w_{new} - \boldsymbol{\nabla}_w L(w, \mathcal{D})$
  end for
  $w \leftarrow w - \alpha \cdot (w_{new} - w)$
end for

FOMAML vs Reptile



Benchmarks: MAML on Omniglot

Omniglot	5-way Accuracy 1-shot	5-way Accuracy 5-shot	20-way Accuracy 1-shot	20-way Accuracy 5-shot
MANN, no conv	82.8%	94.9%	–	–
MAML, no conv (ours)	$\mathbf{89.7 \pm 1.1\%}$	$\mathbf{97.5 \pm 0.6\%}$	–	–
Siamese nets	97.3%	98.4%	88.2%	97.0%
matching nets	98.1%	98.9%	93.8%	98.5%
neural statistician	98.1%	99.5%	93.2%	98.1%
memory mod.	98.4%	99.6%	95.0%	98.6%
MAML (ours)	$\mathbf{98.7 \pm 0.4\%}$	$\mathbf{99.9 \pm 0.1\%}$	$\mathbf{95.8 \pm 0.3\%}$	$\mathbf{98.9 \pm 0.2\%}$

MAML/FOMAML on MiniImagenet

MiniImagenet	5-way Accuracy 1-shot	5-shot
fine-tuning baseline	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
nearest neighbor baseline	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$
matching nets	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
meta-learner LSTM	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML, first order approx. (ours)	$\mathbf{48.07 \pm 1.75\%}$	$\mathbf{63.15 \pm 0.91\%}$
MAML (ours)	$\mathbf{48.70 \pm 1.84\%}$	$\mathbf{63.11 \pm 0.92\%}$

MAML/FOMAML/Reptile on Omniglot

Algorithm	1-shot 5-way	5-shot 5-way	1-shot 20-way	5-shot 20-way
MAML + Transduction	$98.7 \pm 0.4\%$	$99.9 \pm 0.1\%$	$95.8 \pm 0.3\%$	$98.9 \pm 0.2\%$
FOMAML + Transduction	$98.3 \pm 0.5\%$	$99.2 \pm 0.2\%$	$89.4 \pm 0.5\%$	$97.9 \pm 0.1\%$
Reptile	$95.39 \pm 0.09\%$	$98.90 \pm 0.10\%$	$88.14 \pm 0.15\%$	$96.65 \pm 0.33\%$
Reptile + Transduction	$97.68 \pm 0.04\%$	$99.48 \pm 0.06\%$	$89.43 \pm 0.14\%$	$97.12 \pm 0.32\%$

MAML/FOMAML/Reptile on MiniImagenet

Algorithm	1-shot 5-way	5-shot 5-way
MAML + Transduction	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
FOMAML + Transduction	$48.07 \pm 1.75\%$	$63.15 \pm 0.91\%$
Reptile	$47.07 \pm 0.26\%$	$62.74 \pm 0.37\%$
Reptile + Transduction	$49.97 \pm 0.32\%$	$65.99 \pm 0.58\%$

Further reading

- Reptile Playground
- Learning to learn by GD by GD
- MAML
- Reptile