Learning to Optimize:
Learning the backwards pass using message-passing neural networks

Thijs Simons

Universiteit Leiden
The Netherlands

What is learning to optimize?

Very general:

Use machine learning to develop methods that are able to solve problems

But

Emphasis is on learning how to updating the state of the system that is optimized

Why?

- Lower convergence

  - Lower validation loss after T number of steps

- Faster

  - Wall clock speed

  - Number of steps

- Discover new optimization methods

- Learn specialized optimizers

  - Robustness

  - Transferability

Example of L2O

$$\boldsymbol{\theta}_{t+1} \quad \leftarrow \quad \boldsymbol{\theta}_t - \alpha \, \Delta \boldsymbol{\theta}_t$$

Example of L2O

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha \, \Delta \boldsymbol{\theta}_t$$

Parameterize!

Example of L2O

$$\boldsymbol{\theta}_{t+1} \quad \leftarrow \quad \boldsymbol{\theta}_t - \alpha \, \Delta\boldsymbol{\theta}_t$$

Parameterize!

$$\boldsymbol{\theta}_{t+1} \quad \leftarrow \quad \boldsymbol{\theta}_t - \alpha \, \boldsymbol{f}(\ldots, \boldsymbol{\phi})$$

Example of L2O

$$\boldsymbol{\theta}_{t+1} \quad \leftarrow \quad \boldsymbol{\theta}_t \; - \; \alpha \, \Delta\boldsymbol{\theta}_t$$

Parameterize!

$$\boldsymbol{\theta}_{t+1} \quad \leftarrow \quad \boldsymbol{\theta}_t \; - \; \alpha \, f(\ldots, \boldsymbol{\phi})$$

Neural net parameterized by $\boldsymbol{\phi}$

Not only the update function can be parameterized

- Forward pass

    - DARTS

- Initialization params

    - MAML
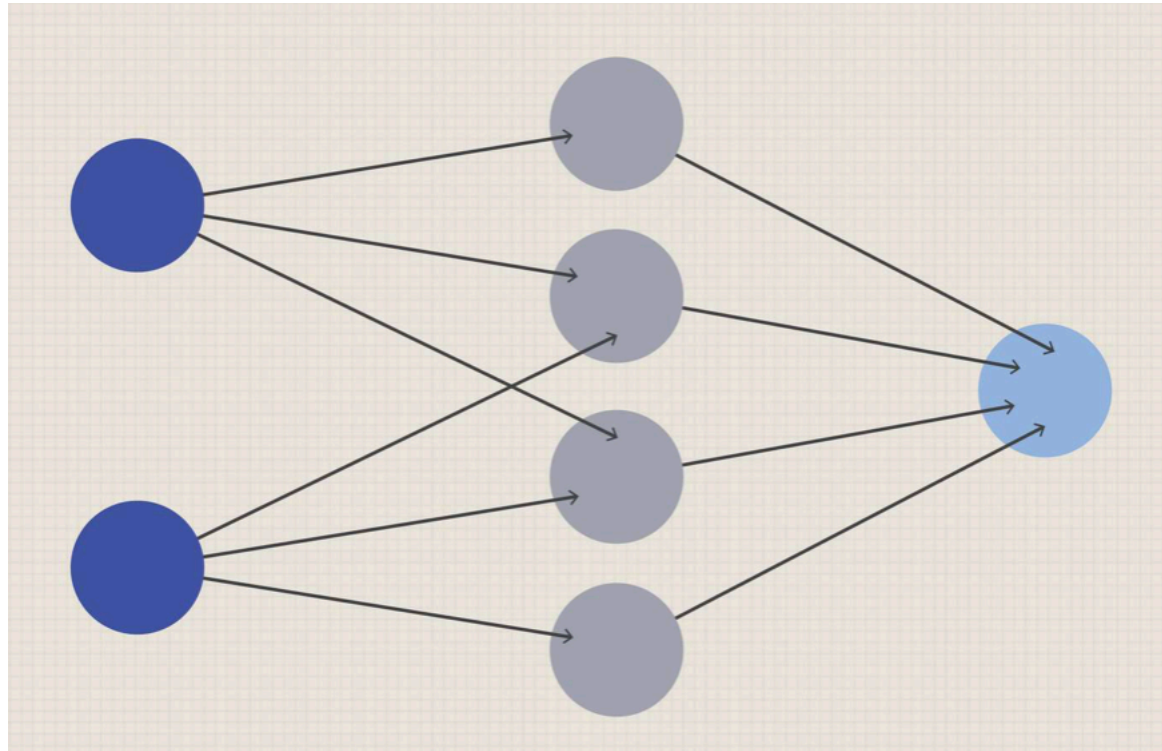
- Loss function

- Learning rate

Not only the update function can be parameterized

- Forward pass
  - DARTS
- Initialization params
  - MAML
- Loss function
- Learning rate
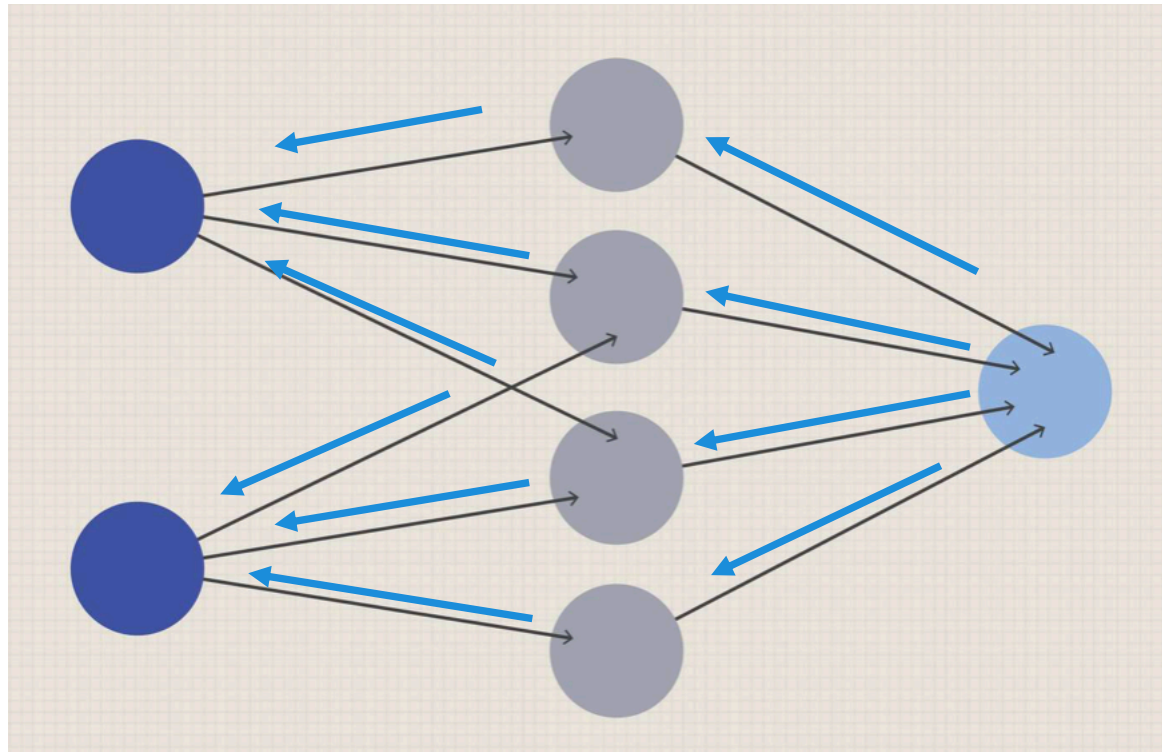- Backwards pass **(our method)**

How do we parameterize the backwards pass?

We look at the neural network from the perspective of graphs

How do we parameterize the backwards pass?

We pass messages through this graph using message passing functions

How do we parameterize the backwards pass?

We pass messages through this graph using message passing functions
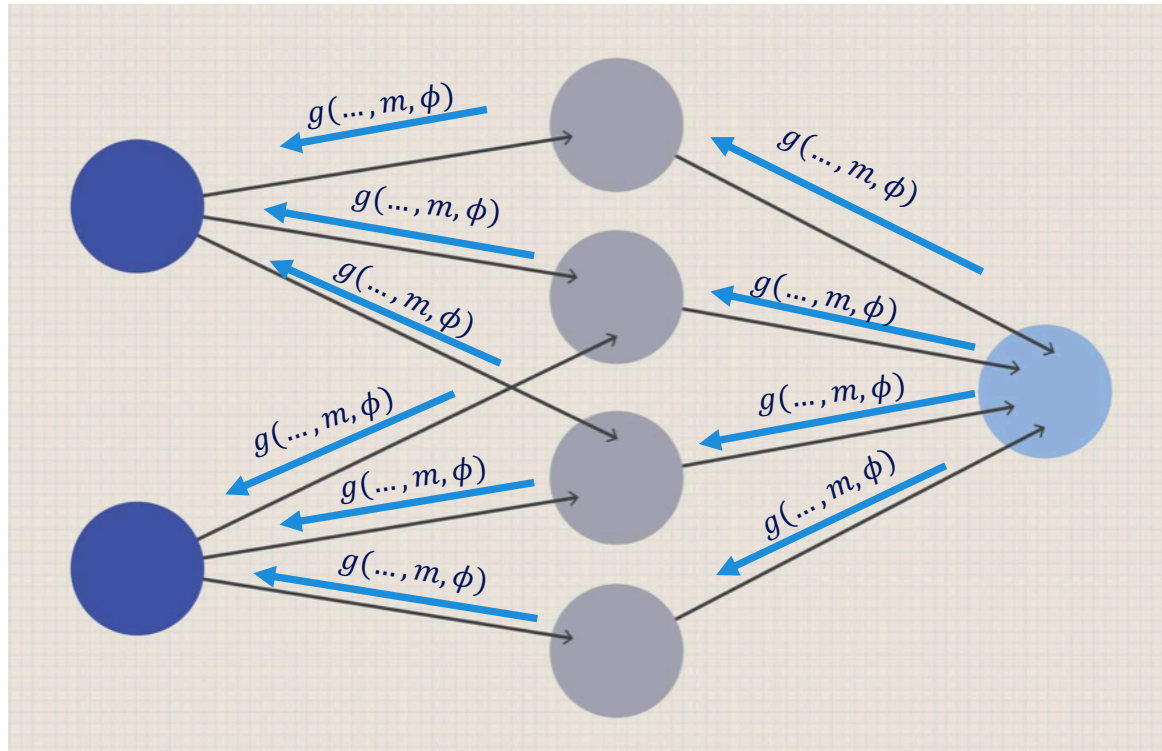
Why do we all this?

Mainly proof of concept

- Can we even train it?

- Can we learn from it?

Method

Forward θ_1

$L_1(x_1, \theta_1)$

Method

Forward $\boldsymbol{\theta}_1$

$L_1(x_1, \theta_1)$

Backwards

Method

Forward $\boldsymbol{\theta}_1$

$L_1(x_1, \theta_1)$

Backwards

$\boldsymbol{\theta}_2 \quad \leftarrow \quad \boldsymbol{\theta}_1 - \alpha\,\Delta\boldsymbol{\theta}_1$

Method

Forward $\boldsymbol{\theta}_1$

$L_1(x_1, \theta_1)$

Backwards

$\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_1 - \alpha \, \Delta\boldsymbol{\theta}_1$

Forward $\boldsymbol{\theta}_2$

$L_2(x_2, \theta_2)$

Backwards

Method

Forward $\boldsymbol{\theta}_1$

$L_1(x_1, \theta_1)$

Backwards

$\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_1 - \alpha \Delta\boldsymbol{\theta}_1$

Forward $\boldsymbol{\theta}_2$

$L_2(x_2, \theta_2)$

Backwards

$\boldsymbol{\theta}_3 \leftarrow \boldsymbol{\theta}_2 - \alpha \Delta\boldsymbol{\theta}_2$

repeat for **T** steps

Method

Forward $\boldsymbol{\theta}_1$ → $L_1(x_1, \theta_1)$

Backwards

$\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_1 - \alpha \, f(\dots, \boldsymbol{\phi})$

Forward $\boldsymbol{\theta}_2$ → $L_2(x_2, \theta_2)$

Backwards

$\boldsymbol{\theta}_3 \leftarrow \boldsymbol{\theta}_2 - \alpha \, f(\dots, \boldsymbol{\phi})$

repeat for **T** steps
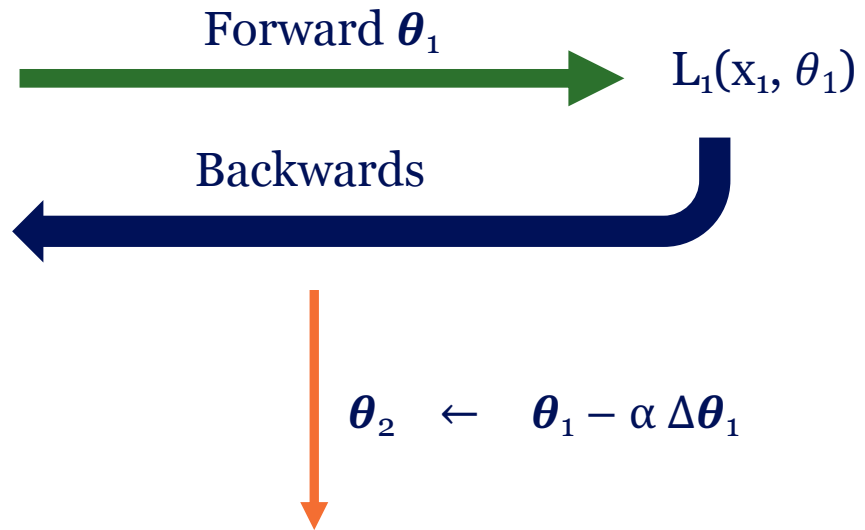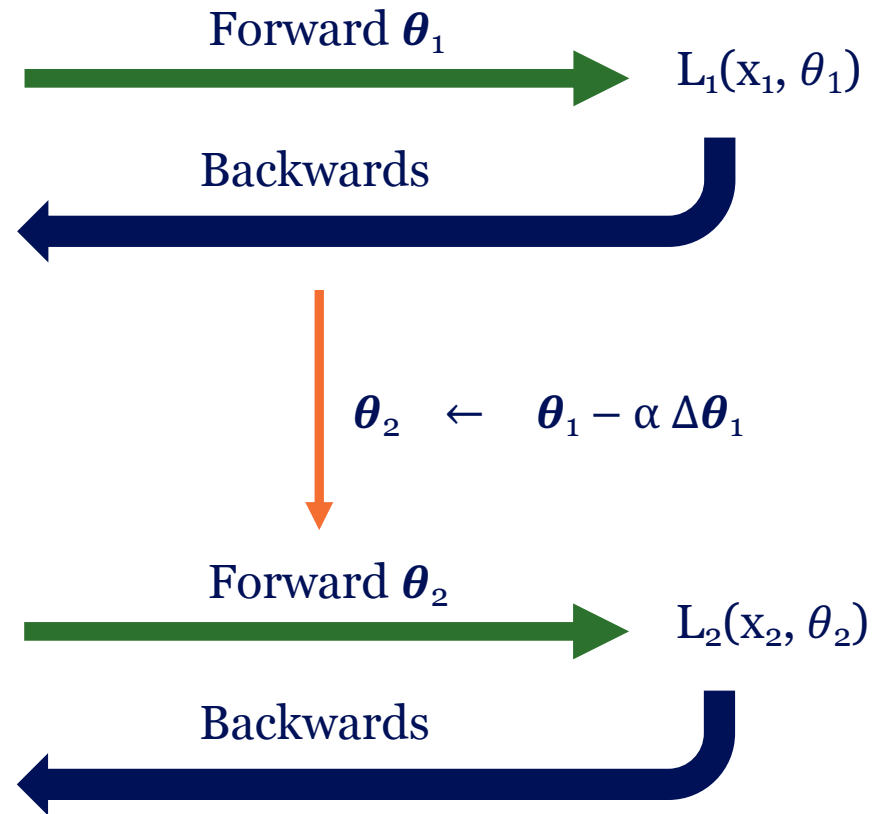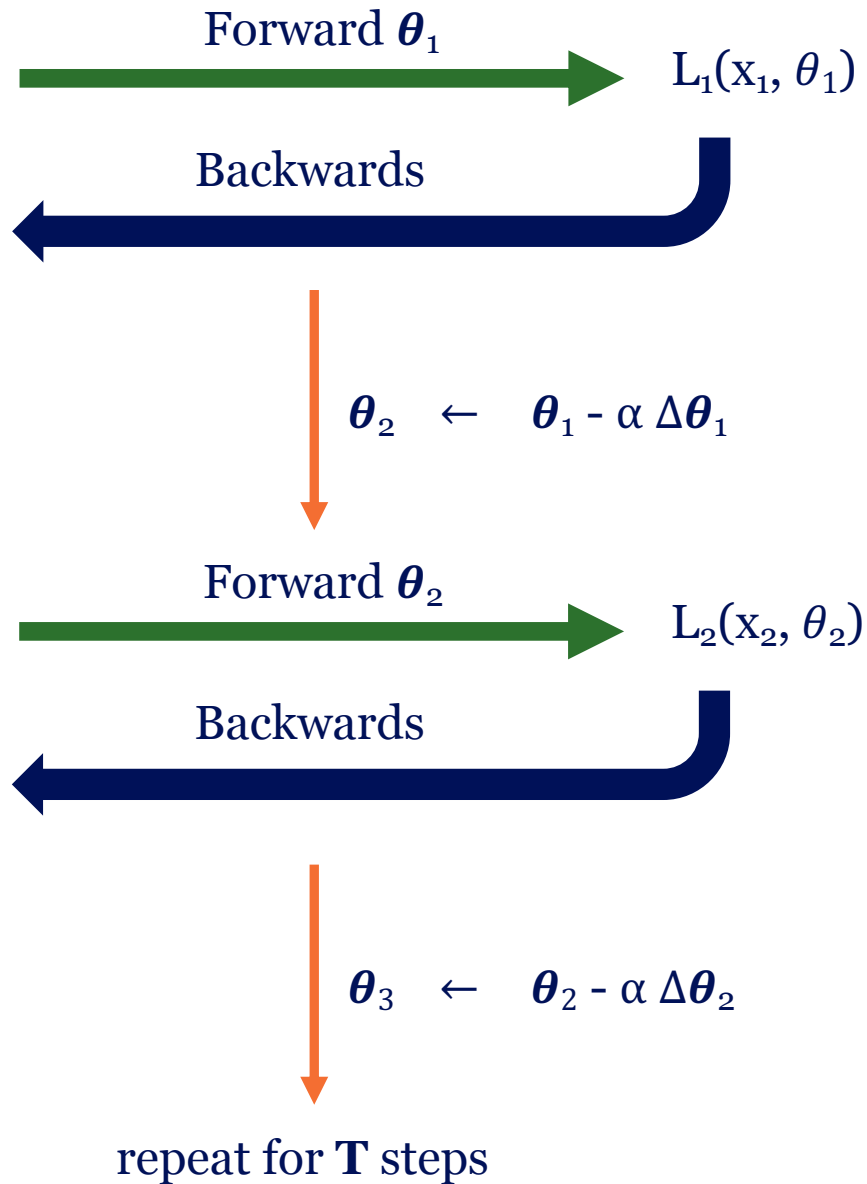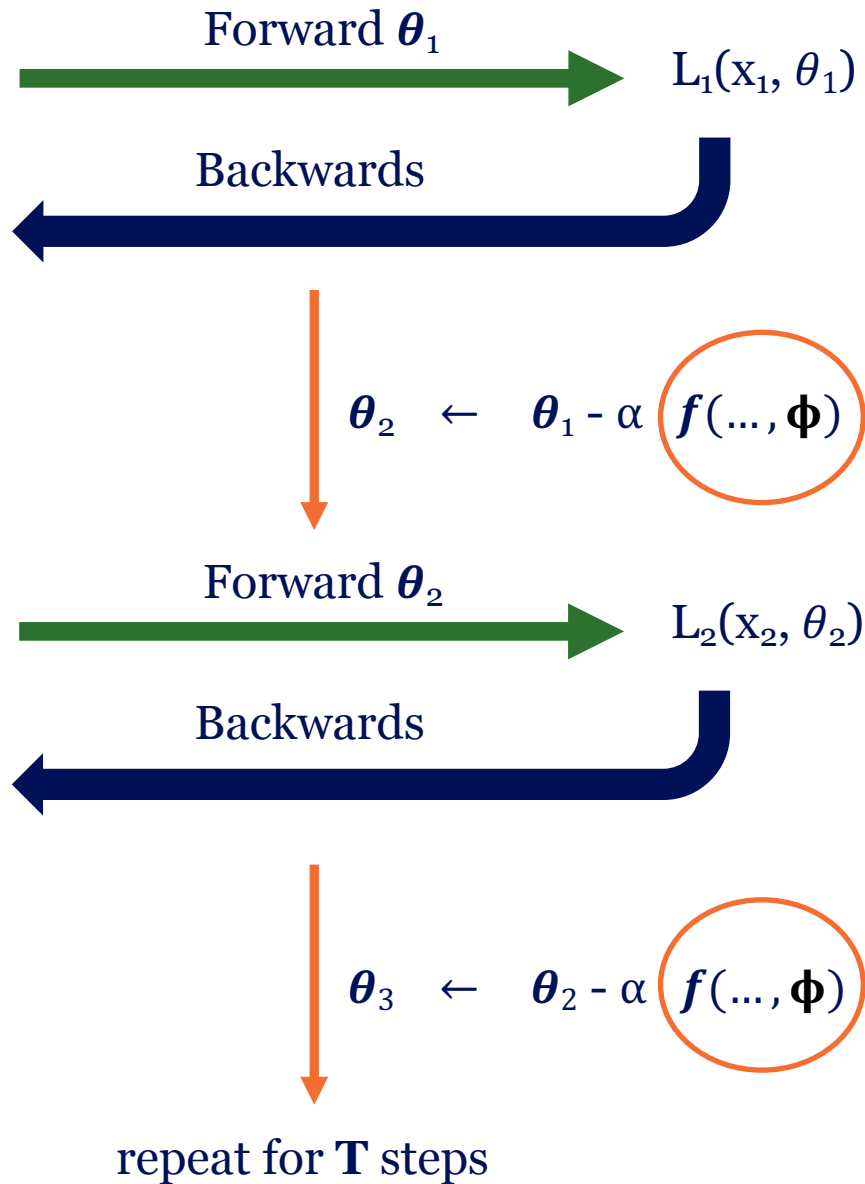
Parameterize the update rule!

Method

Forward $\boldsymbol{\theta}_1$

$L_1(x_1, \theta_1)$

Backwards

$\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_1 - \alpha\, \boldsymbol{f}(\dots, \boldsymbol{\phi})$

Forward $\boldsymbol{\theta}_2$

$L_2(x_2, \theta_2)$

Backwards

$\boldsymbol{\theta}_3 \leftarrow \boldsymbol{\theta}_2 - \alpha\, \boldsymbol{f}(\dots, \boldsymbol{\phi})$

repeat for **T** steps

$$\sum_t^T L_t(x_t, \theta_t, \boldsymbol{\phi})$$

Method

Forward $\boldsymbol{\theta}_1$

$L_1(x_1, \theta_1)$

Backwards

$\boldsymbol{\theta}_2 \quad \leftarrow \quad \boldsymbol{\theta}_1 - \alpha\, f(\ldots, \boldsymbol{\phi})$

Forward $\boldsymbol{\theta}_2$

$L_2(x_2, \theta_2)$

Backwards

$\boldsymbol{\theta}_3 \quad \leftarrow \quad \boldsymbol{\theta}_2 - \alpha\, f(\ldots, \boldsymbol{\phi})$

repeat for **T** steps

$$\sum_t^T L_t(x_t, \theta_t, \boldsymbol{\phi})$$

Differentiable
With respect to $\boldsymbol{\phi}$!!