A contrastive rule for meta-learning

João Sacramento

with: Nicolas Zucchet, Simon Schug,
Johannes von Oswald, Dominic Zhao

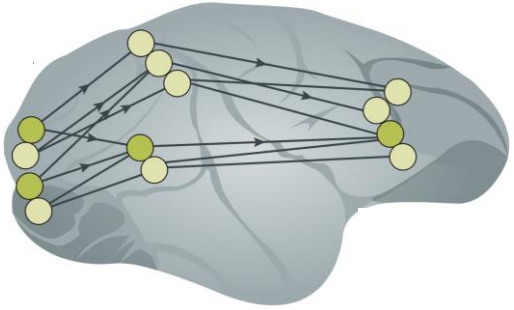
Institute of Neuroinformatics
University of Zürich & ETH Zürich

ETH zürich

University of Zurich[UZH]

institute of neuroinformatics

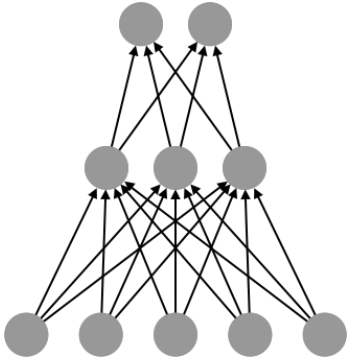
Neuroscience



testable hypotheses,
theories, interpret data

novel ideas, insight

Artificial intelligence



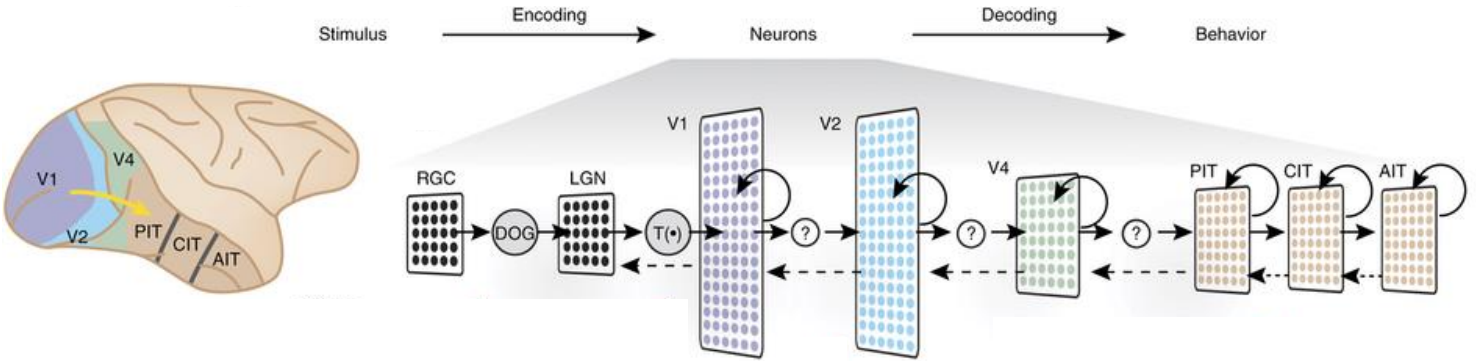
Exciting methodological advances
record thousands of neurons simultaneously
→ bridge the cellular and network level

Human-level performance in challenging tasks using artificial neural networks

Outline

- Credit assignment in artificial and biological neural networks. The deep learning recipe for credit assignment
- Meta-learning: slowly learning to learn new tasks quickly
- The credit assignment problem at the meta-level
- A new way to solve it? The contrastive meta-learning rule
- Discuss a many-systems model for fast learning

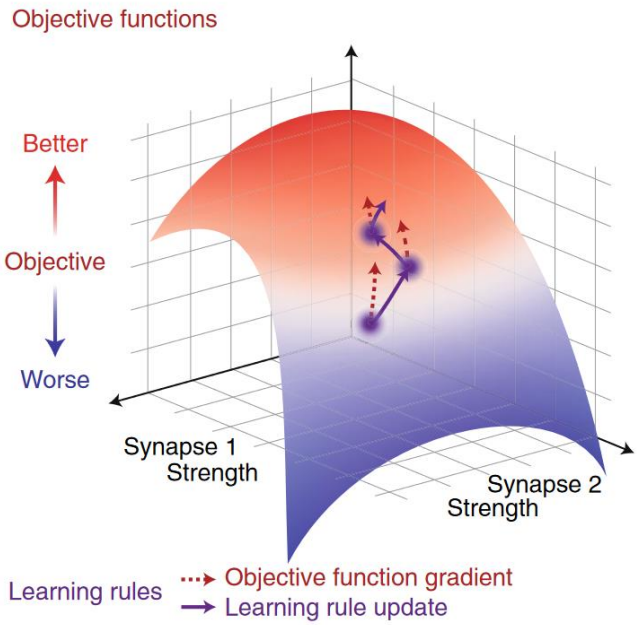
Credit assignment in neural networks



- Complicated mapping from stimulus to behavior
- How does a single synapse influence behavioral output?

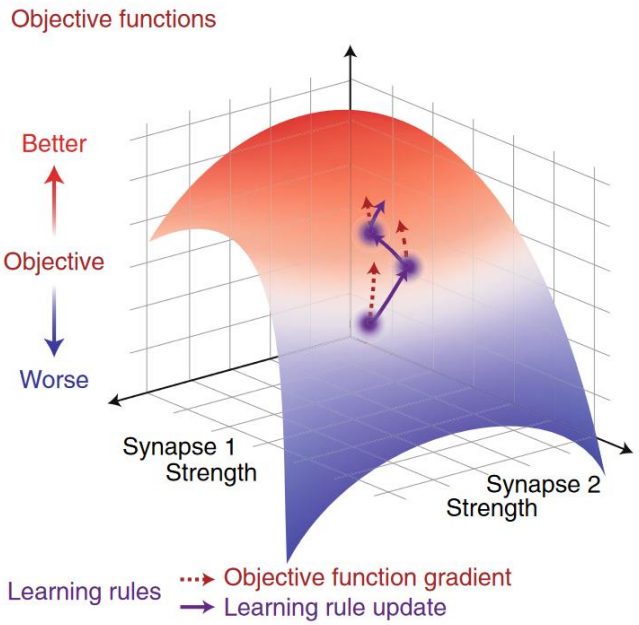
Adapted from Yamins & DiCarlo, *Nat Neurosci* 2016

Learning as optimization



Richards, Lillicrap, Therien, Kording, et al., *Nat Neurosci* 2020

Learning by gradient descent



Adjust synapses W to maximize objective function F by taking steps along its gradient:

$$W_{t+1} = W_t + \eta_t \nabla F(W_t)$$

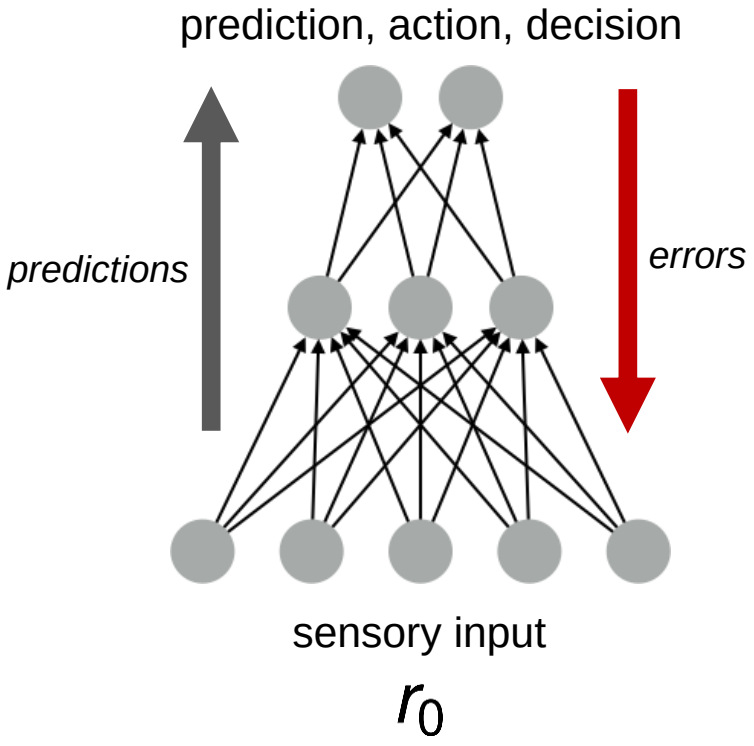
How is $\nabla F(W)$ determined?

Richards, Lillicrap, Therien, Kording, et al., *Nat Neurosci* 2020

Error backpropagation

$$r_2 = \rho(u_2)$$
$$u_2 = W_2\, r_1$$

$$r_1 = \rho(u_1)$$
$$u_1 = W_1 r_0$$



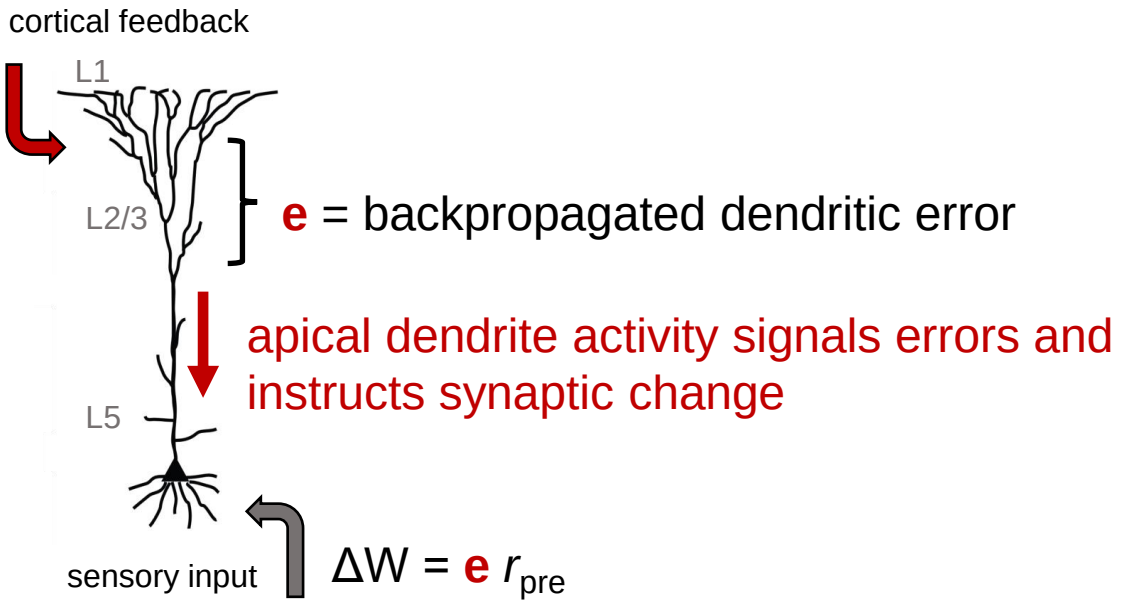
$$e_2 = y^* - r_2$$

$$\Delta W_2 \propto e_2\, r_1^T$$

$$e_1 = \rho'(u_1)\, W_2^T\, e_2$$

$$\Delta W_1 \propto e_1\, r_0^T$$

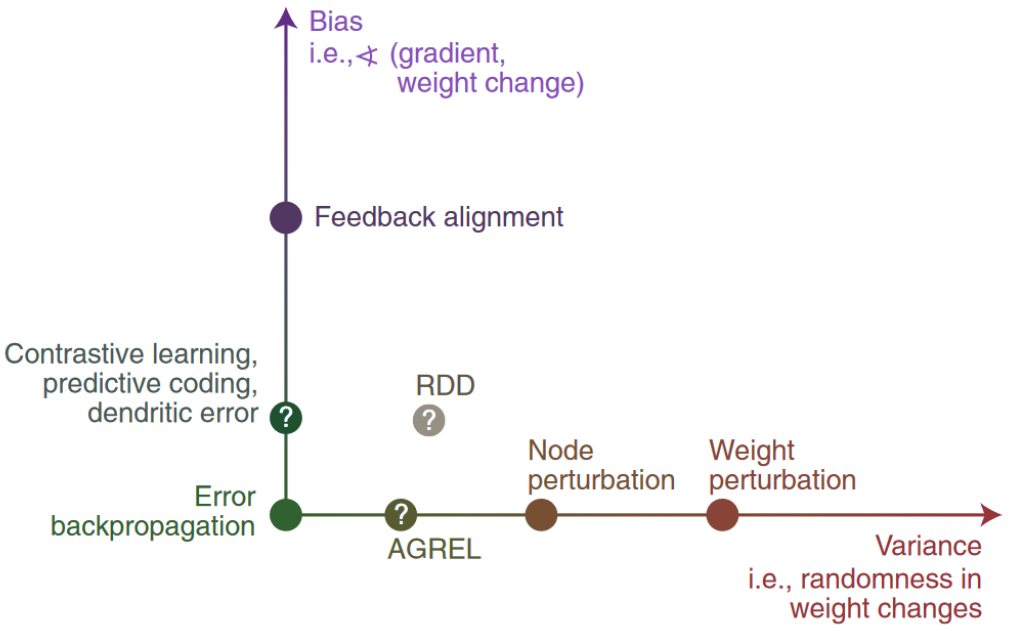
Error backpropagation in the brain?



Guerguiev et al., *eLife* 2017
Sacramento et al., *NeurIPS* 2018
Payeur, Guerguiev et al., *Nat Neurosci* 2020
Meulemans, Farinha et al., *arXiv* 2021
Körding & König, *J Comput Neurosci* 2001

Inspiration: Larkum, *Trends Neurosci* 2013

Approximate gradient descent



In the rest of the talk, we will assume that we have *some way* of optimizing an objective function.

Outline

- Credit assignment in artificial and biological neural networks. The deep learning recipe for credit assignment

- Meta-learning: slowly learning to learn new tasks quickly

- The credit assignment problem at the meta-level

- A new way to solve it? The contrastive meta-learning rule

- Discuss a many-systems model for fast learning

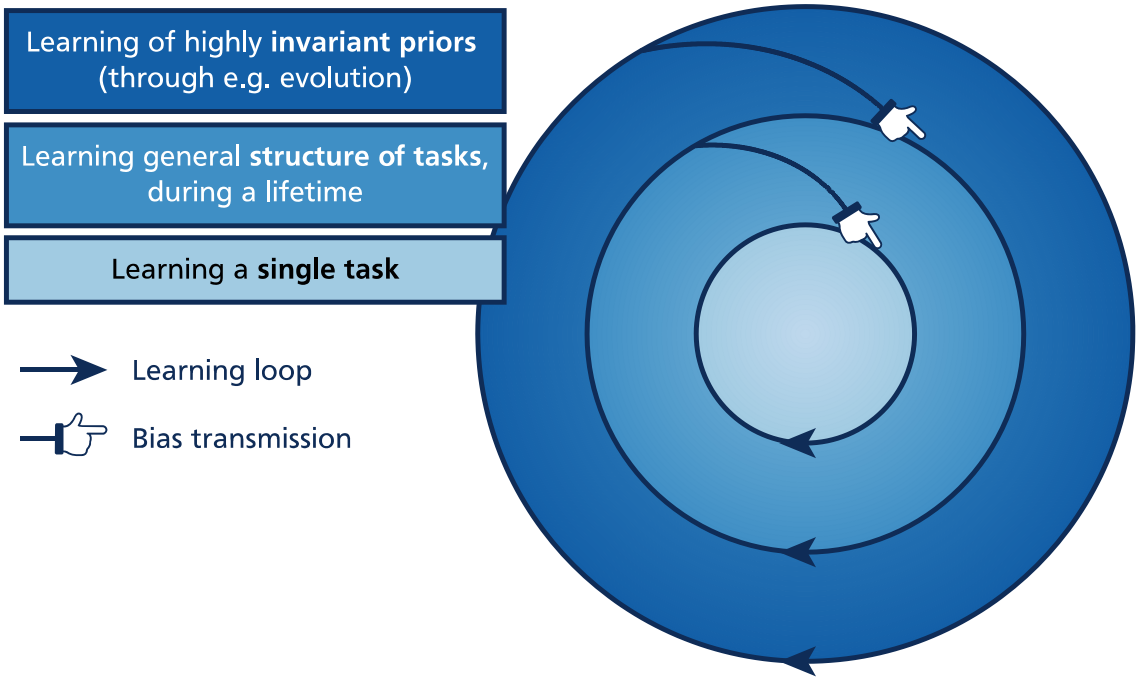
Meta-learning

Humans can learn from very little data; ANNs seem to require a lot of data

1. Evolution gives us strong inductive biases to efficiently learn the sort of problems we will likely encounter

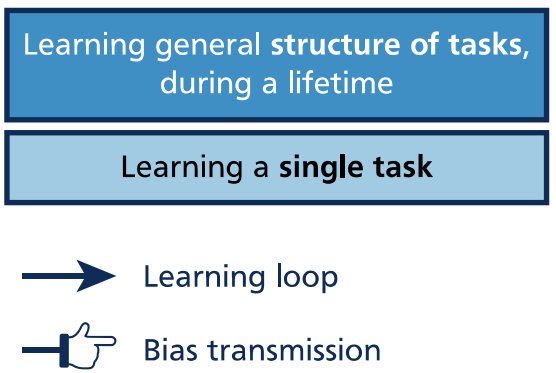
2. Throughout our lives we encounter not one, but many learning problems
   → This allows us to slowly (over many tasks) acquire biases that eventually lead to fast learning (of a new single task)

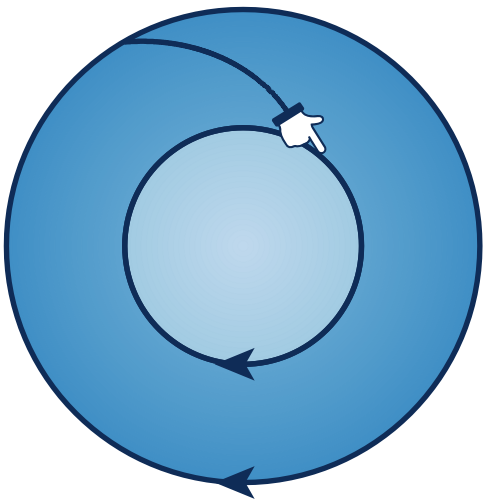
Meta-learning



Adapted from: Wang, *Curr Opin Behav Sci* 2021

Meta-learning





Adapted from: Wang, *Curr Opin Behav Sci* 2021

Fast and slow parameters

We distinguish between two types of parameters or “components”:

- Task-shared meta-parameters: θ (slow)
- Task-specific parameters: ϕ (fast)

Meta-parameters do not change when learning a single task

Learning as optimization

Learning: minimize a loss L function, evaluated on "training" data

$$\phi^* = \arg\min_{\phi} L(\theta, \phi, \mathcal{D}^{\text{train}})$$

fixed meta-parameters transmit inductive bias

Generalization: evaluate loss function on new "test" data

$$L(\theta, \phi^*, \mathcal{D}^{\text{test}})$$ should be low

Meta-learning: wrap around

Meta-learning as bilevel optimization

Learning: optimize parameters on current-task training loss

Meta-learning: optimize meta-parameters on expected loss over tasks evaluated on learned task-specific parameters

$$\min_{\theta} \overbrace{\mathbb{E}_{\tau \sim p(\tau)} \left[L(\theta, \phi_\tau^*, D_\tau^{\text{test}}) \right]}^{L^{\text{out}}}$$

approximate expectation
by randomly sampling tasks

$$\text{s.t.} \quad \phi_\tau^* = \arg\min_{\phi} \underbrace{L(\theta, \phi, \mathcal{D}_\tau^{\text{train}})}_{L^{\text{in}}}$$

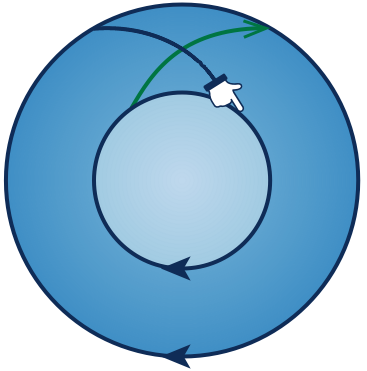
Outline

- Credit assignment in artificial and biological neural networks. The deep learning recipe for credit assignment
- Meta-learning: slowly learning to learn new tasks quickly
- The credit assignment problem at the meta-level
- A new way to solve it? The contrastive meta-learning rule
- Discuss a many-systems model for fast learning

Meta-level credit assignment

1. A learning system is presented with a task
2. The learning system solves the task (fits training data)
3. The performance of the learning system is evaluated (on new data)

How should the transmitted bias θ change,
to improve generalization performance of the
learning system?



Meta-learning by gradient descent

- Standard technique: gradient descent (GD), again
- How does it look like when using GD at both levels?

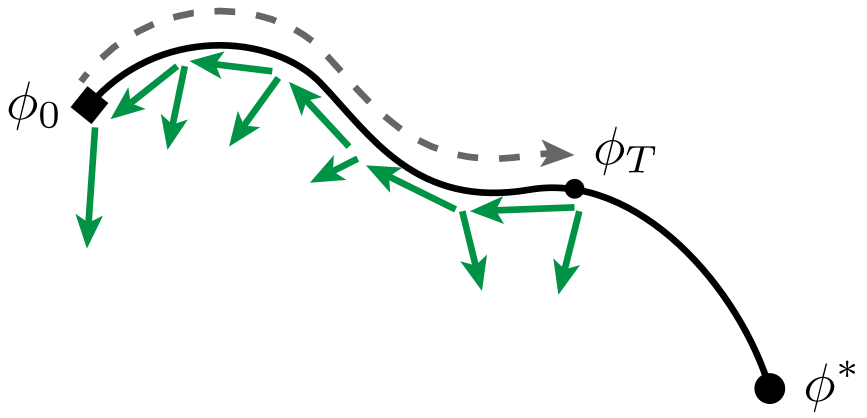
Learn task (multiple steps of GD): $\phi_{t+1} = \phi_t - \eta_\phi \nabla_\phi L(\theta, \phi_t, D^{\text{train}}) \quad t = 0, \ldots, T$

Small change (single step of GD): $\Delta\theta = -\eta_\theta \frac{d}{d\theta} L(\theta, \phi_T(\theta), D^{\text{test}})$

Schmidhuber, 1987
Bengio et al., 1991
Ravi & Larochelle, *ICLR* 2017
Finn et al., *ICML* 2017

How do we compute the "meta-gradient"?

Backpropagation-through-training



Backpropagation-through-training

- Need to store intermediate states of the parameter
- **Need to revisit parameter trajectory in reverse-time order**
- Need to evaluate second derivatives of the loss function

Crude first-order approximation: drop all nasty terms above

Meta-learning in the brain?

Minimal wish-list for biologically-plausible meta-learning:

- ✓ **Locality:** Use only first derivatives (assume these are available)
- ✓ **Causality**: calculations in reverse time order are forbidden
- ✓ **Flexibility**: be agnostic as to how a task is solved and how derivatives are computed

Outline

- Credit assignment in artificial and biological neural networks. The deep learning recipe for credit assignment
- Meta-learning: slowly learning to learn new tasks quickly
- The credit assignment problem at the meta-level
- A new way to solve it? The contrastive meta-learning rule
- Discuss a many-systems model for fast learning

Contrastive meta-learning

- Introduce an auxiliary augmented loss function (with $\beta \geq 0$):

$$\mathcal{L}(\theta, \phi, \beta) = L^{\text{in}}(\theta, \phi, D^{\text{train}}) + \beta\, L^{\text{out}}(\theta, \phi, D^{\text{test}})$$

- Learn at different levels of mixing: $\phi^*_\beta = \arg\min_\phi \mathcal{L}(\theta, \phi, \beta)$

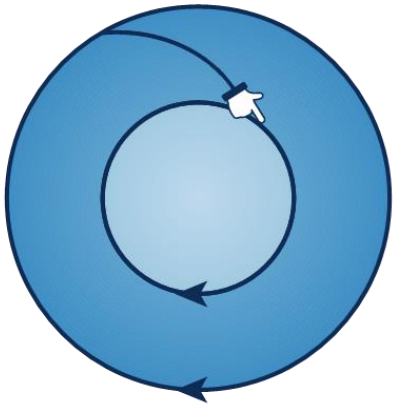
- Invoke equilibrium propagation theorem:

$$\frac{d}{d\theta} L^{\text{out}}(\theta, \phi^*_0, D^{\text{test}}) = \lim_{\beta \to 0} \frac{1}{\beta} \left(\frac{\partial \mathcal{L}}{\partial \theta}(\theta, \phi^*_\beta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\theta, \phi^*_0, 0) \right)$$

Scellier & Bengio, *Front Comput Neurosci* 2017

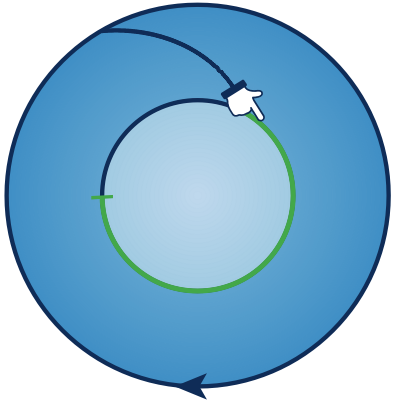
Contrastive meta-learning

1. Sample task(s)



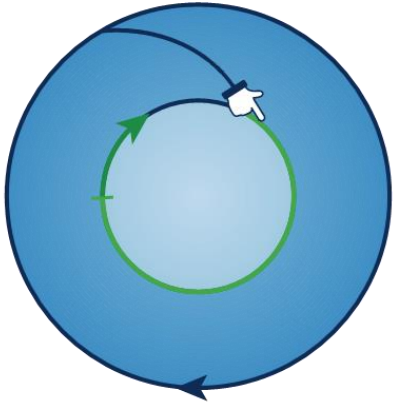
Contrastive meta-learning

1. Sample task(s)
2. Solve lower-level learning problem: $\phi_0^* = \arg\min_{\phi} \mathcal{L}(\theta, \phi, 0)$



Contrastive meta-learning

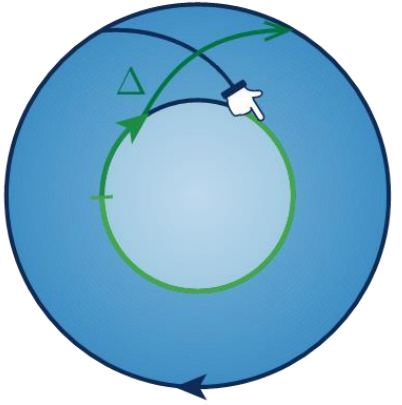
1. Sample task(s)
2. Solve lower-level learning problem: $\phi_0^* = \arg\min_{\phi} \mathcal{L}(\theta, \phi, 0)$
3. Solve augmented learning problem: $\phi_\beta^* = \arg\min_{\phi} \mathcal{L}(\theta, \phi, \beta)$



Contrastive meta-learning

1. Sample task(s)
2. Solve lower-level learning problem: $\phi_0^* = \arg\min_{\phi} \mathcal{L}(\theta, \phi, 0)$
3. Solve augmented learning problem: $\phi_\beta^* = \arg\min_{\phi} \mathcal{L}(\theta, \phi, \beta)$
4. Contrast gradients:

$$\Delta\theta = -\frac{\eta_\theta}{\beta}\left(\frac{\partial \mathcal{L}}{\partial \theta}(\theta, \phi_\beta^*, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\theta, \phi_0^*, 0)\right)$$



Properties of contrastive meta-learning

Contrastive meta-learning is…

- Agnostic to how each learning problem is solved
- Agnostic to how gradients are computed
- A causal learning rule
- Based on two points in time (requires memory)
- As accurate as needed: to improve meta-gradient estimate, improve lower-level learning process (cf. analytical results)

Meta-learning a complex synapse

Parameter: fast-changing plastic synaptic weight ϕ

Meta-parameters: slowly-changing long-term synaptic base weight ω
strength of attraction λ towards base weight

$$L^{\text{in}}(\theta, \phi, D^{\text{train}}) = L(\phi, D^{\text{train}}) + \sum_{i=1}^{N} \lambda_i (\phi_i - \omega_i)^2$$

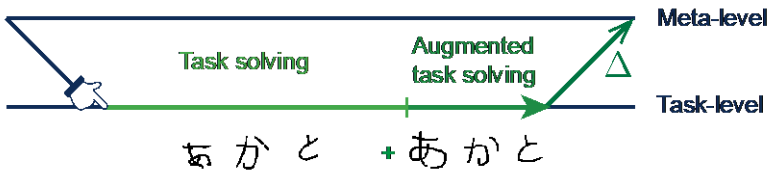
meta-learned regularizer keeps synaptic weights close to a synapse-specific internal value

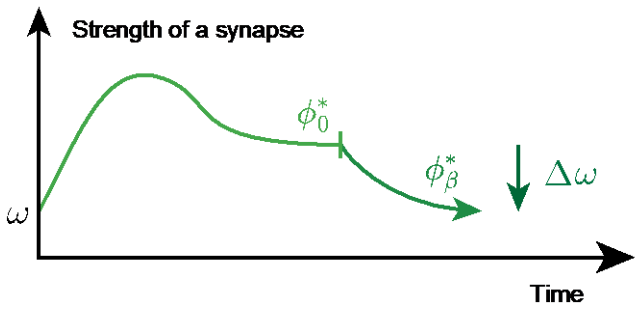
Can be combined with any neural network!
High-level model for synaptic consolidation

$$L^{\text{out}}(\theta, \phi, D^{\text{test}}) = L(\phi, D^{\text{test}})$$

Kirkpatrick et al., *PNAS* 2017
Rajeswaran et al., *NeurIPS* 2019

Meta-learning a complex synapse

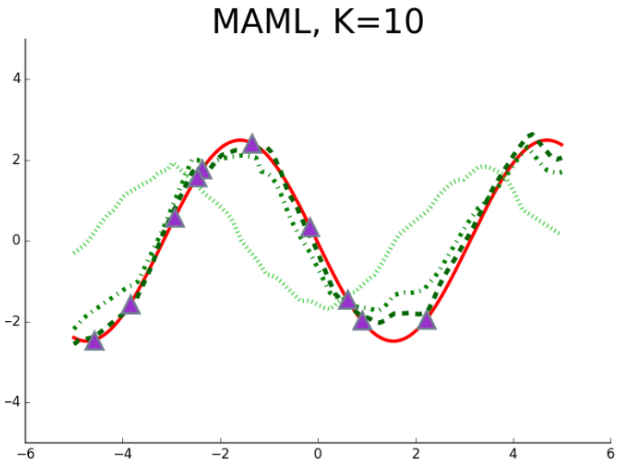




$$\Delta\omega \propto \lambda\left(\phi_\beta^* - \phi_0^*\right)$$

local meta-learning rule
update only depends on synaptic variables

Few-shot regression

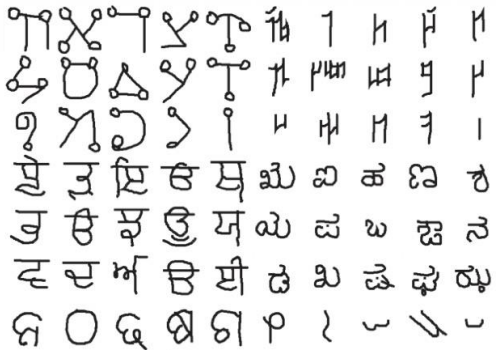


Method	Mean squared error (↓)
MAML (BPTT)	0.435±0.039
CML GD	0.305±0.101
CML SGD	0.258±0.070
From scratch	1.999±3.500

- Learn to regress sine waves from 10 points
- Randomly sample phase from [0, π] and amplitude from [0.1, 5]
- Two-hidden-layer neural network (1-40-40-1)

Few-shot visual classification

Omniglot



dataset of hand-written characters
1623 characters, 20 examples each

Method	20-way 1-shot	20-way 5-shot
MAML [10]	$95.8^{\pm 0.3}$	$98.9^{\pm 0.2}$
First-order MAML [10]	$89.4^{\pm 0.5}$	$97.9^{\pm 0.1}$
Reptile [28]	$89.43^{\pm 0.14}$	$97.12^{\pm 0.32}$
iMAML (GD) [26]	$94.46^{\pm 0.42}$	$98.69^{\pm 0.1}$
iMAML (Hessian-free) [26]	$96.18^{\pm 0.36}$	$99.14^{\pm 0.1}$
Contrastive meta-learning (syn.)	$94.16^{\pm 0.12}$	$98.06^{\pm 0.26}$

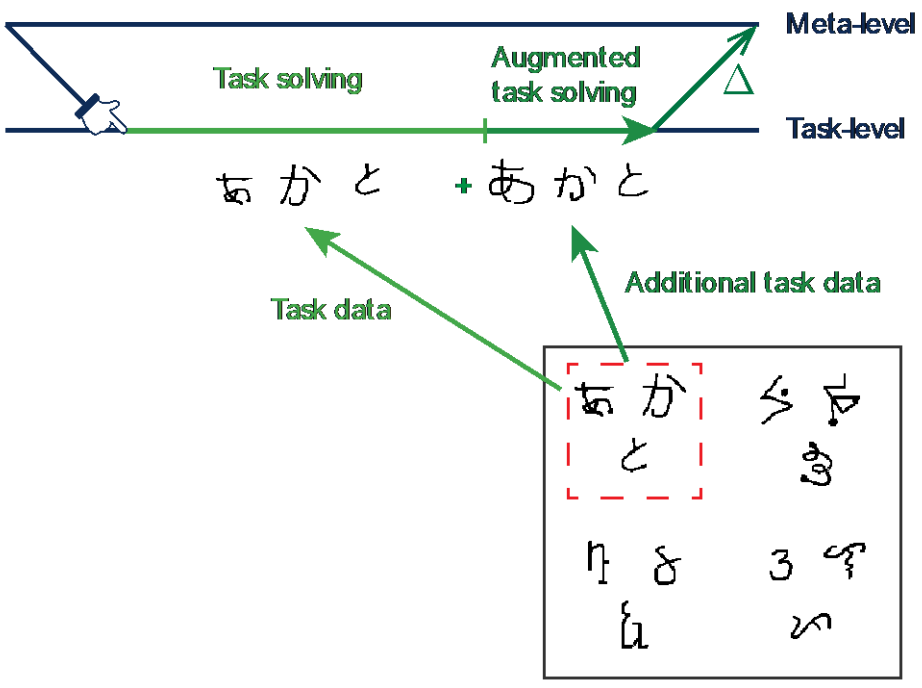
Neural network: 4 convolutional layers, batch normalization, rectified linear units

Performance close to second-order methods
Same for a more complex dataset (miniImageNet)

Outline

- Credit assignment in artificial and biological neural networks. The deep learning recipe for credit assignment
- Meta-learning: slowly learning to learn new tasks quickly
- The credit assignment problem at the meta-level
- A new way to solve it? The contrastive meta-learning rule
- Discuss a many-systems model for fast learning

Complementary learning systems view



consolidation

Cortex

plasticity

Hippocampus

Hold and replay data
Instruct and control sign of plasticity

McClelland et al., *Psychol Rev* 1995
Kumaran et al., *Trends Cog Sci* 2016

Conclusion

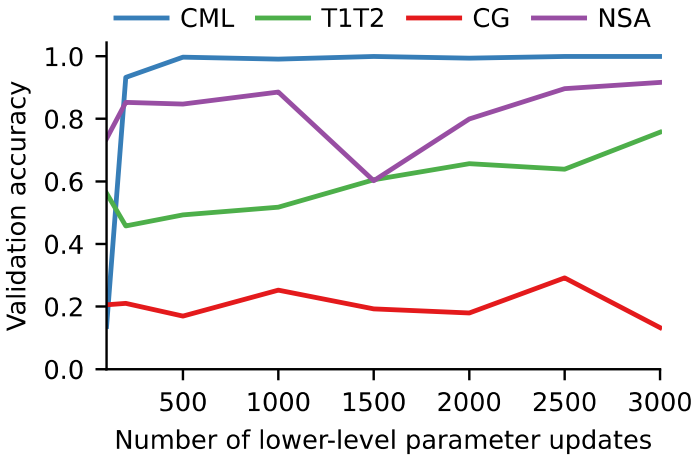
- Learning fast by slowly learning over many tasks
- Meta-level credit assignment by solving learning problems *twice*
- Our algorithm can build upon existing theories of backpropagation in the brain to implement meta-learning
- Critical role of hippocampus in enabling meta-learning in the brain?

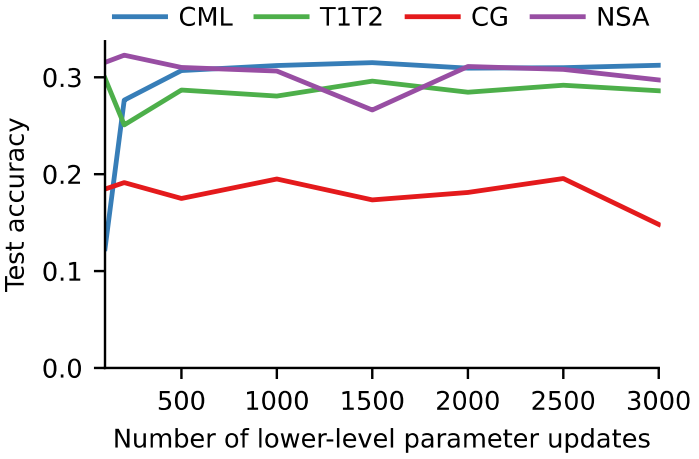
***Special thanks*:** Nicolas Zucchet, Simon Schug, Johannes von Oswald
Dominic Zhao

Funded by: Swiss National Science Foundation

ETH zürich

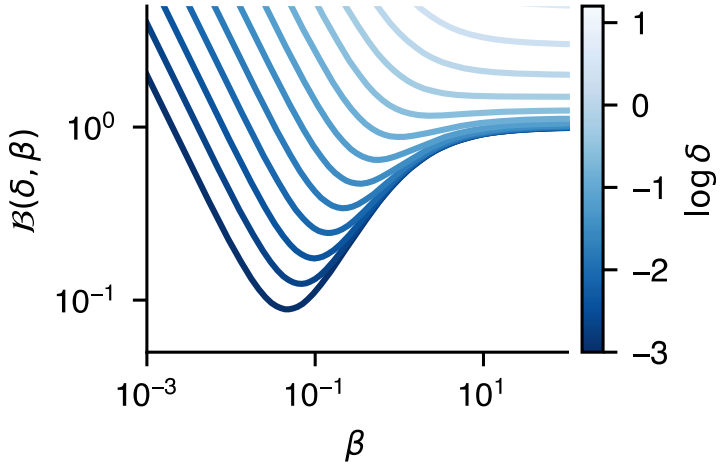
Overfitting on CIFAR-10 data

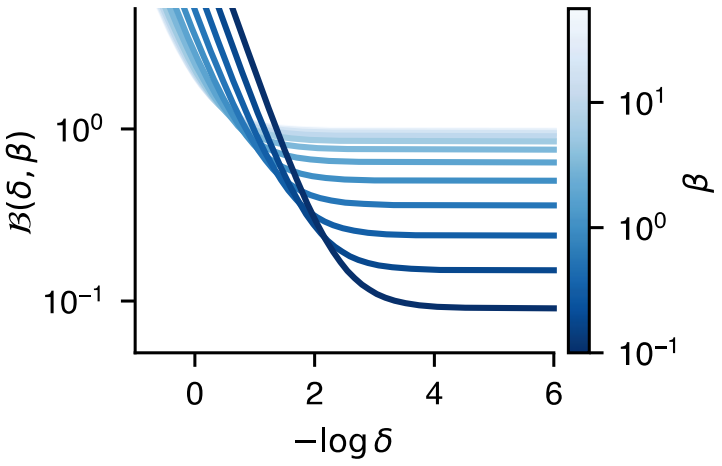




1000 examples, LeNet-5 model, 95% data on hold-out validation set

Theory


$\mathcal{B}(\delta, \beta)$
10^0
10^{-1}
10^{-3}
10^{-1}
10^{1}
β
1
0
−1
−2
−3
$\log \delta$


$\mathcal{B}(\delta, \beta)$
10^0
10^{-1}
0
2
4
6
$-\log \delta$
10^{1}
10^{0}
10^{-1}
β

Theory

Suppose that L^{in} and $\mathrm{L}^{\mathrm{out}}$, as functions of ϕ, are strongly convex, smooth and have Lipschitz Hessians. Assume their partial derivatives with respect to θ are B^{in}- and $\mathrm{B}^{\mathrm{out}}$-Lipschitz continuous and have Lipschitz Hessians.

Strong convexity and regularity assumptions

Let $\beta > 0$. Suppose that the fixed point estimates verify

$$\|\hat{\phi}_0 - \phi_0^*\| \leq \delta, \ \|\hat{\phi}_\beta - \phi_\beta^*\| \leq \delta'$$

Then, there exists a constant C s.t.,

$$\|\widehat{\nabla}_\theta - \nabla_\theta\| \leq B^{\mathrm{in}} \frac{\delta + \delta'}{\beta} + B^{\mathrm{out}} \delta' + C \frac{\beta}{1+\beta}$$

Equilibrium propagation

Theorem (Scellier and Bengio, 2017)

Suppose L^{in} and $\mathrm{L}^{\mathrm{out}}$ twice continuously differentiable such that $\phi^*_{\theta,\beta}$ is well defined.

Then,

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\frac{\partial \mathcal{L}}{\partial \beta}(\phi^*_{\theta,\beta},\theta,\beta) = \frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial \mathcal{L}}{\partial \theta}(\phi^*_{\theta,\beta},\theta,\beta)^\top$$

Transforms a derivative w.r.t. a **vector** into a derivative w.r.t. a **scalar**

Equilibrium propagation

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\frac{\partial \mathcal{L}}{\partial \beta}(\phi^*_{\theta,\beta}, \theta, \beta)\bigg|_{\beta=0} = \frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial \mathcal{L}}{\partial \theta}(\phi^*_{\theta,\beta}, \theta, \beta)^\top\bigg|_{\beta=0}$$

equilibrium propagation theorem applied at $\beta = 0$

The left-hand side is what we want to compute as

$$\frac{\partial \mathcal{L}}{\partial \beta}(\phi^*_{\theta,\beta}, \theta, \beta) = \frac{\partial\left[L^{\mathrm{in}} + \beta L^{\mathrm{out}}\right]}{\partial \beta}(\phi^*_{\theta,\beta}, \theta, \beta) = L^{\mathrm{out}}(\phi^*_{\theta,\beta}, \theta)$$

The right-hand side is easy to approximate:

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial \mathcal{L}}{\partial \theta}(\phi^*_{\theta,\beta}, \theta, \beta)\bigg|_{\beta=0} \approx \frac{1}{\beta}\left(\frac{\partial \mathcal{L}}{\partial \theta}(\phi^*_{\theta,\beta}, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\phi^*_{\theta,0}, \theta, 0)\right)$$

finite difference