FORTH
INSTITUTE OF MOLECULAR BIOLOGY & BIOTECHNOLOGY

Meta-learning synaptic plasticity rules to approximate gradient descent

Christos Karageorgiou Kaneen
Neuroscience M.Sc., School of Medicine,
University of Crete

Committee:
Yiota Poirazi (**supervisor**)
Konrad Körding
Emmanouil Froudarakis

Heraklion, Crete
February 2024

Dedicated to the art and science of learning the art and science of learning

Acknowledgements

This work would not have been possible without the tremendous support and invaluable feedback of Konrad Körding, Yiota Poirazi, Ilenna Jones and Spiros Chavlis, to whom I am grateful for all their motivation and encouragement throughout the year.

Christos Karageorgiou Kaneen

Neuroscience & AI Researcher, Software Engineer ckarageorgkaneen@gmail.com

github.com/ckarageorgkaneen

linkedin.com/in/ckarageorgkaneen

twitter.com/ckaraneen

Education

- **University of Crete** Heraklion, GR
  M.Sc., Neuroscience, School of Medicine *Oct. 2021 - March 2024 (expected)*
- **Technical University of Crete** Chania, GR
  M.Eng., Electrical & Computer Engineering, School of ECE *Sept. 2014 - Dec. 2019*

Research Experience

- **Banerjee Lab, Cold Spring Harbor Laboratory** New York, USA
  Neuroscience & AI Researcher (NeuroAI Research Intern) *(**upcoming**) June 2024 - Aug. 2024*
  Behavior-driven neuro-evolution.
- **Kording Lab, University of Pennsylvania** Philadelphia, USA
  Neuroscience & AI Researcher (Visiting Scholar) *Dec. 2022 - March 2023*
  Led a project of meta-learning synaptic plasticity rules for exploring how gradient descent can best be approximated.
- **Poirazi Lab, IMBB-FoRTH** Heraklion, GR
  Neuroscience & AI Researcher *July 2022 - (ongoing)*
  - Wrote a research proposal.
  - Explored basic dendritic properties of neurons computationally.
  - Continued the work I started at the Kording Lab as a thesis project.

Work Experience

- **mist.io** (acquired by Dell) Remote
  Software & Infrastructure Engineer *April 2021 - July 2022*
  Developed features and fixed issues having to do with machines, stacks, networks, scripts and other resources managed by the platform (mainly within the *api* and *tests* submodules of both community and enterprise editions). I also worked on outlining and implementing the API and functional tests of a distributed micro-services project for Juniper Networks. See my public contributions on mist-ce.
- **Greek Research & Technology Network** Remote
  Software Engineer & Maintainer *May 2020 - April 2021*
  Contributed to reducing Greece's public sector bureaucracy: built, alongside one lead developer, v1 of mitos: the National Registry of Public Services (Εθνικό Μητρώο Διαδικασιών). See diavlos.
- **Myrmex** (acquired by Ocado) Athens, GR
  Software Engineer *Sept. 2019 - March 2020*
  - **Power monitor application**: Implemented a real-time GUI that displays the current battery state of each running robot.
  - **gRPC clients**: Implemented gRPC client proxies for the APIs of two key system components.
  - **Bug-reporting tool**: Implemented a bug-reporting automation tool for gathering diagnostics for whenever a bug or problem occurred. This tool decreased the time required for data collection from more than 20 min. to 15 sec.

Internship Experience

- **Larkum Lab, Humboldt University of Berlin** Remote
  Software & Hardware Engineer *Jan. 2021 - Aug. 2021*
  Implemented a prototype of an experimental neuroscience protocol for the Larkum and Kremkow labs. This work was done in my free time, on weekends and it was based on my previous work. See airtrack.
- **Technical University of Crete** Chania, GR
  Network Administrator *Nov. 2018 - June 2019*
  - Learned to do various system and network administration tasks.
  - Secured network printers, etc.
  - Read the *UNIX and Linux System Administration Handbook*.
- **Oracle** Athens, GR
  Database Administrator *Oct. 2018 - Nov. 2018*
  - Familiarized myself with Oracle DB 12c internals and did day-to-day DBA tasks.
  - Brushed up (PL/)SQL skills.
- **Google Summer of Code** Athens, GR
  Software Engineer *May 2018 - Sept. 2018*
  Using NLP and ML techniques, I implemented a tool for the Open Technologies Alliance that extracts Responsibilities assigned to Public Administration Organizations from Greece's Official Government Gazette documents. See ggx.
- **Open Lab Athens** Athens, GR
  Software Engineer *July 2017 - Sept. 2017*
  - Debugged the beta version of the *Iris SMS* Android app.
  - Implemented a smart pen (Neo N2) GUI for getting stats and syncing form data.

Volunteer Experience

- **Poirazi Lab** Remote
  Software Engineer *July 2020 - May 2021*
  Volunteered on PyBpod, a programming interface for the Bpod: a system for precise measurement of small animal behavior. This work was done in my free time, on weekends. See pybpod.

Supervisor Experience

- *Henry J. Flynn* @ Poirazi Lab, *April – August 2023*: Drexel University mechanical engineering student who under my guidance built upon my volunteer contribution and improved the Mouse2AFC protocol.

Awards, Fellowships & Funding

- 2023 – 2024
  - *IMBB* ***fellowship*** (*July 2023 – April 2024*, Poirazi Lab): *5000€*
- 2022 – 2023
  - *Erasmus+*/IKY **funding** (*Dec. 2022 – March 2023*, USA travel & stay): *4500€*
  - *Visiting Scholar* **funding** (*Dec. 2022 – March 2023*, Kording Lab) : $*6000*

Conferences & Summer Schools

- MLSS2024: *Machine Learning Summer School 2024*, Okinawa, Japan, *March 4 – 15, 2024*

Publications

1. C.K. Kaneen, "*Methodology for designing GDPR compliant IoT applications*", Diploma Work, School of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece, 2019 (https://doi.org/10.26233/heallink.tuc.84151)

2. C.K. Kaneen, E.G.M. Petrakis, "*Towards evaluating GDPR compliance in IoT applications*", Procedia Computer Science, Volume 176, 2020, Pages 2989-2998 (https://doi.org/10.1016/j.procs.2020.09.204)

Last update: *27th February 2024*. For latest CV, **click here**.

Abstract

When synaptic plasticity leads to effective learning, neurons are assigned the credit that appropriately corresponds to their specific contribution. In hierarchical neural networks (NNs), the difficulty of distinguishing between credit-related and non-credit-related activity render credit assignment a non-trivial problem. Although speculative biological solutions have been proposed, these are not as potent as the optimization algorithm most widely used in deep learning. Gradient descent (GD) changes the synapses of an artificial NN using the gradient of performance, a quantity that signifies the direction of most rapid improvement of performance. GD is successful because it efficiently optimizes performance with the least modification to irrelevant parameters of a NN. Consequently, many have proposed that the brain may use an algorithm that approximates GD. What the proposed biologically-plausible implementations of GD have in common is that they all assume that the brain uses a sufficiently simple scheme that a human scientist can readily understand and describe and thus allows for mathematical proof that the system does GD. However, there is little reason to assume that evolution might derive learning mechanisms that are easily interpretable by humans. Instead, we may expect that inhibition, bursting and reward, among many other neuronal elements, all jointly contribute to efficient GD-like credit assignment. Here, we apply meta-learning, an algorithm that can be seen as an approximation of what evolution does, on synaptic plasticity rules with many degrees of freedom and explore how GD can best be approximated. We show how our method can solve non-trivial problems better than the non-meta-learned rules with a performance comparable to the performance of GD. Our results provide insights into how vastly diverse mechanisms of physiology and plasticity may enable efficient biologically-plausible credit assignment.

Περίληψη

Όταν η συναπτική πλαστικότητα συνεπάγεται αποτελεσματική μάθηση, στους νευρώνες ανατίθεται το εύσημο που κατάλληλα αντιστοιχεί στην συγκεκριμένη τους συνεισφορά. Στα ιεραρχικά νευρωνικά δίκτυα (ΝΔ), η δυσκολία της διάκρισης μεταξύ νευρωνικής δραστηριότητας σχετιζόμενης με το εύσημο και ασυσχέτιστης με αυτό καθιστά το πρόβλημα της ανάθεσης ευσήμου ένα μη-τετριμμένο πρόβλημα. Παρόλο που έχουν προταθεί βιολογικές λύσεις, αυτές δεν είναι τόσο αποτελεσματικές όσο ο κύριος αλγόριθμος βελτιστοποίησης που χρησιμοποιείται ευρέως στο πεδίο της βαθιάς μάθησης. Η μέθοδος της κατάβασης μέγιστης κλίσης (ΜΚ) αλλάζει τις συνάψεις ενός τεχνητού ΝΔ χρησιμοποιώντας τη κλίση (παράγωγο), μια ποσότητα που υποδηλώνει την κατεύθυνση της πιο γρήγορης βελτίωσης της απόδοσης. Η επιτυχία της ΜΚ οφείλεται στην αποτελεσματική βελτιστοποίηση της απόδοσης με την ελάχιστη τροποποίηση των μη-χρήσιμων παραμέτρων ενός δικτύου. Κατά συνέπεια, πολλοί ερευνητές προτείνουν ότι το εγκέφαλος ενδέχεται να χρησιμοποιεί έναν αλγόριθμο που προσεγγίζει την ΜΚ. Το κοινό στοιχείο που μοιράζονται οι προτεινόμενες βιολογικά-εφικτές υλοποιήσεις της ΜΚ είναι ότι όλες τους υποθέτουν ότι ο εγκέφαλος χρησιμοποιεί έναν επαρκώς απλό κανόνα μάθησης που ένας επιστήμονας μπορεί να κατανοήσει και να περιγράψει εύκολα, αλλά και που να επιτρέπει μια μαθηματική απόδειξη ότι το εν λόγω σύστημα μπορεί να υλοποιήσει την ΜΚ. Ωστόσο, δεν υπάρχουν σημαντικοί λόγοι για να υποθέσουμε ότι η εξέλιξη παράγει μηχανισμούς μάθησης που είναι εύκολα ερμηνεύσιμοι από ανθρώπους. Αντιθέτως, υποθέτουμε ότι κάποιοι νευρικοί μηχανισμοί όπως η αναστολή, η πυροδότηση κατά ριπές και η ανταμοιβή, μεταξύ πολλών άλλων, συνεισφέρουν από κοινού στην αποτελεσματική ανάθεση ευσήμου όμοια της ΜΚ. Στην παρούσα εργασία, εφαρμόζουμε μετα-μάθηση, έναν αλγόριθμο που μπορεί να θεωρηθεί ως μια προσέγγιση του τι κάνει η εξέλιξη στο βιολογικό εγκέφαλο, σε κανόνες συναπτικής πλαστικότητας με πολλές παραμέτρους και εξερευνούμε πως να προσεγγίσουμε την ΜΚ με τον καλύτερο δυνατό τρόπο. Δείχνουμε πως η μέθοδός μας μπορεί να λύσει μη-τετριμμένα προβλήματα καλύτερα από αντίστοιχους κανόνες πλαστικότητας που δεν έχουν υποστεί μετα-μάθηση, με απόδοση συγκρίσιμη με την απόδοση της ΜΚ. Τα αποτελέσματά μας υποδεικνύουν πως ένα ευρύ φάσμα ποικιλόμορφων μηχανισμών φυσιολογίας και πλαστικότητας μπορεί να καταστήσει δυνατή την αποδοτική βιολογικά-εφικτή ανάθεση ευσήμου.

Contents

1. Introduction

1.1. Motivation

The brain is a remarkably efficient learning system whose vast and seemingly incomprehensible complexity inspires the awe of the beautiful in the ambitious explorers aspiring to understand it. Although much about how the brain learns remains to be discovered, synaptic plasticity mechanisms have been largely thought to form the neurochemical foundations of learning and memory (Magee & Grienenberger, 2020).

In deep learning, the backpropagation of error algorithm (backprop) has been widely successful in training artificial neural networks (ANNs). However, its biological plausibility has been a subject of debate. Therefore, it has become crucial to investigate learning mechanism alternatives to backprop that are more in line with biological reality, such as synaptic plasticity. Inspired by backprop's success and what we already know about the neuroscience of learning, the motivation for this work stems from the quest to better understand the brain's plasticity apparatus and their computational implications (Lillicrap et al., 2020).

In this context, we aim to explore whether associative synaptic plasticity rules can approximate gradient descent (GD), the optimization method used by backprop, without relying on the backprop-driven computations of gradients during the training process. The central question we address is whether a set of Hebbian learning rules, emulating the biological process of synaptic strengthening and weakening based on correlated neural activity, can be harnessed to achieve GD-like performance in ANNs. We also seek to elucidate the conditions under which unsupervised or globally supervised Hebbian learning rules can approximate GD driven by the need to bridge the gap between the effectiveness of backprop and the brain's more efficient learning systems.

To the best of our knowledge, no one has yet used associative synaptic plasticity to train multilayer perceptrons (MLPs) on non-trivial tasks to clearly demonstrate an approximation of GD performance. We illustrate that a set of Hebbian learning rules with sufficient complexity and degrees of freedom can be optimized in an online manner using a meta-learning system, akin to what evolution does, to approximate the gradient-related weight deltas produced by a teacher network trained with backprop. In summary, we show the conditions under which straight-forward, parametrized Hebbian rules can approximate GD performance feedforward-only training, without requiring any backward passes that calculate vectorized error signals or explicit neuron-specific gradients (e.g. backprop) during the training phase.

1.2. Backpropagation of error

Although first described in the 1970s, efficient backprop was outlined explicitly and applied to NNs in the following decade. Whereas, around 1986, researchers popularized backprop after using it to demonstrate the emergence of useful internal representations in hidden layers (Linnainmaa, 1970, 1976; Werbos, 1974; Speelpenning, 1980; Werbos, 1982; Parker, 1985; LeCun, 1985; Rumelhart et al., 1986).

Backprop is a gradient estimation method used for training NNs, such as MLPs. The goal of training is for the network to learn complex representations from input and target output data by encoding an approximate "in-to-out" mapping into its parameters. Essentially, to make the network yield output data that is closest to target data when given input data not seen during the training process.

During training, using the chain rule, backprop recursively calculates the gradient of an error function, also commonly referred to as a loss function, that quantifies the performance of a NN, with respect to (w.r.t) its parameters (e.g. synaptic weights), starting at the output layer and moving backwards until the first hidden layer (Leibniz, 1673). By propagating an error signal (that is a function of the outputs obtained from the last layer) back through the previous layers (i.e. "backpropagating"), this process enables the calculation of gradients. After calculation, these can be used by an optimization algorithm such as GD to update the parameters of the network (i.e. synaptic weights) in order to minimize the loss: the differences between prediction values and target values.

GD is an optimization algorithm that finds combinations of parameters that constitute local minima in the landscape of a differentiable multivariate function. It is widely used for training ANNs, wherein the goal is to minimize a parameter-dependent loss function that quantifies performance. Using the gradient that is usually calculated through backprop, GD iteratively updates the network's parameters that contribute to the loss (i.e. synaptic weights) by taking small steps in the negative direction of the gradient. In other words, GD makes the parameters descend the gradient. Thus, changing the parameters in the direction that yields the steepest decrease of the value of the loss at each given point (i.e. set of parameter values). In that regard, GD efficiently optimizes performance with the least modification to irrelevant parameters of a NN: a key attribute of its success.

Today, stochastic GD, a simple extension of GD, serves as the most popular algorithm most widely used to train deep NNs. Of course, many other variants exist, such as the popular Adam optimizer (Bisong 2019; LeCun et al., 2015; Ruder, 2017).

1.3. Credit assignment in the brain

During learning, the brain appropriately assigns credit or blame to its neurons in order to influence or modify future decisions and actions (e.g. increase performance during learning). The difficulty to distinguish credit-related activity from other activity that is unrelated to credit is known as the credit assignment problem, a challenge that is fundamental in learning in hierarchical NNs (Richards & Lillicrap, 2018).

Several biologically-plausible learning mechanism candidates have been proposed to solve the credit assignment problem, with reinforcement learning and error-driven learning being two of the most prominent examples. In similar vein, various experimental studies have also investigated the responsible neural mechanisms and brain areas associated with backprop, such as the prefrontal cortex and medial frontal cortex (Akaishi, 2016; Stolyarova, 2018; Lamba et al, 2023).

Biological versions of backprop have also been suggested as possible candidate solutions to credit assignment. However, despite promising proposals (e.g. dendritic solutions) in that direction, limitations remain (e.g. weight symmetry) that make the algorithm biologically implausible. Therefore, the existence of functional equivalents of this algorithm in the brain are still heavily debated. Recently, nevertheless, Mark Harnett's group measured somato-dendritic signals that hinted at gradient-calculation capabilities of cortical neurons, with this evidence culminating in the first experimental data from the brain that suggest a credit assignment solution comparable to backprop (Lillicrap et al. 2016; Sacramento et al. 2018; Richards & Lillicrap, 2018; Whittington & Bogacz, 2019; Lillicrap et al. 2020; Payeur et al., 2021; Greedy et al., 2022; Francioni et al., 2023).

As a general rule, in any system that learns via small changes, performance improves when these small changes correlate with the gradient of an objective function that quantifies its performance. Although it is highly uncertain whether the brain learns using a backprop-like solution, a combination of neurophysiological mechanisms such as Hebbian plasticity and dopaminergic neuromodulation may allow it to approximate gradients. Thus, it is not unreasonable to hypothesize that the plasticity mechanisms responsible for useful changes during learning may be approximating GD. And gradients may just be the perfect mathematical abstraction for formally studying synaptic plasticity (Richards & Kording, 2023).

1.4. Associative learning

At the end of the 19th century, William James first alluded to a synaptic learning principle by proposing a “law of neural habit” as a physiological basis for earlier versions of a proposed law of mental association by continuity, described by others a century earlier. Later, Cajal, Tanzi, Freud, Pavlov and, most-evidently, McDougall, presented their own ideas and theories of how we learn and remember, all very reminiscent to the associative models of synaptic plasticity we nowadays refer to as Hebbian. Later on, in 1949, Donald Hebb, largely influenced by the work of his doctoral advisor, Karl Lashley, proposed his own, now-renown, neuropsychological theory, seeking to explain synaptic plasticity: “*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.*” (Brown et al., 2021; Hebb, 1949).

A historically prevalent and foundational learning model, Hebbian theory postulates that the repeated, persistent and simultaneous pre- and post-synaptic activities of two or more neurons increases the efficacy (strength) of their synapses. Serving as the biological basis of unsupervised learning, it is considered a fundamental principle in the study of NNs and cognitive function (Hebbian theory (Wikipedia); McClelland, 2006). Furthermore, to account for learning phenomena that occur across seconds, as opposed to millisecond time scales, researchers have also proposed three-factor learning frameworks, as extensions to classic associative learning models, in an attempt to reconcile them with widely-observed neuromodulatory (e.g. dopaminergic) signaling mechanisms. For a mathematical formulation of Hebbian learning, see the Methods section (chapter 2.1) (Lisman et al., 2011; Kusmierz et al., 2017; Gerstner et al., 2018).

1.5. Meta-learning

1.5.1. Overview

In the past, artificial intelligence researchers used to design algorithms that solve specific problems by hand. However, being too inflexible and unscalable of an approach, such hand-crafting has been largely superseded by ANNs that learn to solve problems without explicit instruction, using backprop. These surpass hand-coded solutions, and often humans, in performance. Yet, contrary to human-like learning, deep-learning-based methods remain largely inefficient and do not generalize, requiring manual tuning of hyperparameters and re-training across different tasks (e.g. due to catastrophic forgetting). One way to overcome such limitations is to automatically optimize the learning process through meta-learning (Kirkpatrick et al., 2017; Lansdel & Kording, 2019).

Meta-learning means learning to learn. A setup that learns to learn usually consists of two optimization systems: a learner and a meta-learner. The learner trains a model to perform a task (learns, e.g. an inner loop that optimizes a model). The meta-learner modifies the learner (meta-learns, e.g. an outer loop that optimizes the parameters of the learner). The goal of meta-learning is to change the learner itself so that the modified version becomes better (e.g. more efficient) than the original one. See Figure 1. (Schmidhuber 1987, 1992; Thrun & Pratt, 1998; Schaul & Schmidhuber, 2010; J.X. Wang et al., 2016; Hospedales et al., 2022).

Over the years, meta-learning approaches have ranged from learning a network's initial weights to continual, scalable and generalized learning, as well as frameworks for discovering or improving biologically-plausible plasticity models (Schmidhuber, 1992; Finn et al., 2015; Lansdell & Kording, 2019; Javed & White, 2019, Lindsey & Litwin-Kumar, 2019, Miconi et al. 2018, 2020). Our work falls into the latter category: that of hyperparameter optimization. In the following sections, we compare and contrast our work to similar work, highlighting our contributions, before presenting our methods and results.

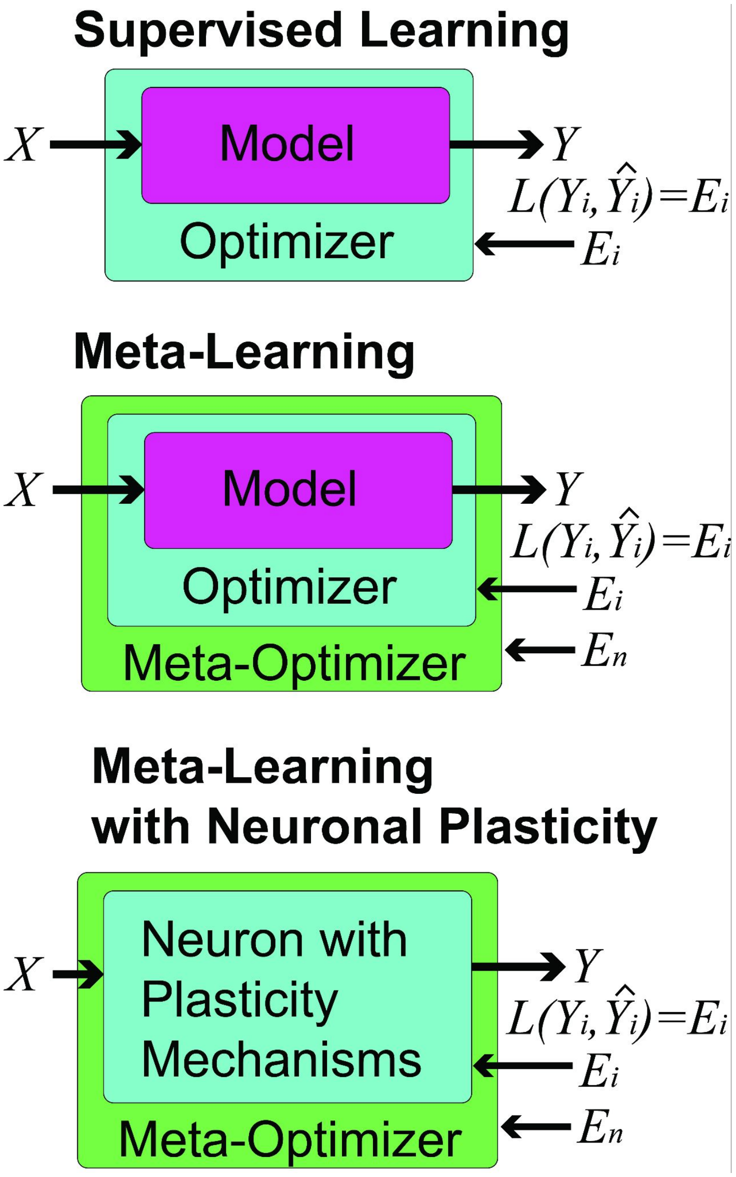


Fig. 1. Comparison of types of learning and meta-learning. In supervised learning (*upper*), an optimizer changes a model. In "general" meta-learning (*middle*), a meta-optimizer changes the optimizer itself. In meta-learning that improves a plasticity model (*lower*), a meta-optimizer changes the parameters of the plasticity model (e.g. "Neuron with plasticity mechanisms"). X: input data, Y: output data, Y_i: output value, $\hat{Y}_i$: target value, E_i: loss (error) value, E_n: meta-loss (error) value.

1.5.2. Related work

Within the research direction of meta-learning synaptic plasticity rules, relevant work falls within two broad categories:

1. learning the feedforward pathway's plasticity rule parameters
2. learning a combination of feedforward and feedback pathway rule parameters

In the earliest known work from the first category, Bengio et al., 1995 sum the loss of multiple networks trained on different tasks (e.g. boolean, 2D classification) and learn a parametric learning rule that learns those tasks successfully. More recently, Andrychowicz et al., 2016 meta-learned the weights of a Recurrent NN (RNN) that parametrizes a learning rule, demonstrating how dynamic rules can be trained. Also using RNNs, Miconi et al., 2018, 2020 made their network plastic by incorporating a Hebbian component into the weights and optimized the coefficients using backprop. Confavreux et al., 2020, 2023 used meta-learning in rate-based and recurrent spiking models and -instead of optimizing or discovering- they *recovered* known plasticity rules that satisfy predefined biological constraints. In the context of reinforcement learning, a team from Copenhagen (Najarro & Risi, 2020; Pedersen & Risi, 2023) evolved Hebbian rules through meta-learning, allowing for a continuous self-organization of weights. In practice, this enables their agents to adaptively learn two control tasks.

Within the second category, Gu et al., 2019 meta-learned rules that update error-propagating feedback connections, whereas Lindsey et al., 2020 meta-learned modulatory feedback pathways and initial weight values. On the other hand, Metz et al., 2019, optimized plasticity rules that update weights of feedforward and feedback connections that project into hidden states. Similarly, Sandler et al., 2021 defined parameterized update rules and then learned the meta-parameters that control both forward and backward passes in the respective connections. Last but not least, Shervani-Tabar & Rosenbaum, 2023 meta-learned a parameterized learning rule that depends on a term based on Oja's rule combined with pseudo-gradient and error-based Hebbian terms, for training networks with fixed feedback connections, using both feedforward and backward passes.

1.5.3. Contributions

Compared to the most similar work:

- we demonstrated an approximation to GD through a clear and direct comparison. *Lindsey et al., 2020* and *Shervani-Tabar & Rosenbaum, 2023* for example do not,
- we showed that there are coefficient configuration for one of our meta-learned Hebbian rules that can generalize across models with different initial weights on a non-linear regression task, without requiring explicit training on random weights beforehand, nor a repeat of the meta-learning process. Whereas, the results of *Lindsey et al., 2020*, for example, depend on initial weights and do not demonstrate any example to the contrary.
- we interpreted the similarity of meta-learned Hebbian rule across synaptic connections per layer by providing a variability analysis of the evolution of coefficients during meta-training and as well as of distributions of the coefficient values after meta-training. We also discussed the biological interpretation of our findings. On the other hand, *Miconi et al., 2018, 2019* and *Najarro et al., 2020*, who also meta-learn different plasticity rules per connection, do not analyze or discuss biological plausibility, which renders their results difficult to interpret.
- we relied on a simple feedforward MLP architecture. *Metz et al., 2019*, *Sandler et al., 2021*, *Shervani-Tabar & Rosenbaum, 2023* and Confavreux et al., 2023 employ less simple architectures with feedback connections.
- we meta-learned Hebbian learning rules that depend on pre- and postsynaptic activity terms, with only one of our rules having global error modulation. *Shervani-Tabar & Rosenbaum, 2023* meta-learned composite parametrized rules that contain a pseudo-gradient and error terms, both depending on feedback pathways and neuron-specific error signals.
- we achieved GD-like performance only through forward passes without requiring backward error-propagating passes during training. Miconi et al., 2018, 2020 optimize rule coefficients and learn using backprop. Whereas in *Shervani-Tabar & Rosenbaum, 2023*, weight updates depend on vectorized error signals derived from backward passes.

2. Methods

2.1. Mathematical formulation of Hebbian learning

To mathematically describe Hebb's postulate and Hebbian learning rules that build upon it, we must first clarify the following constituents:

- w_{ij}, the efficacy of the synapse transmitting signals from a presynaptic neuron j to a postsynaptic neuron i, or the simply the *weight*
- v_j, the presynaptic neuron activity
- v_i, the postsynaptic neuron activity

Since our work deals only with rate-based models, we make the assumption that v_j and v_i refer to time-independent average firing rates.

Two properties characterize Hebb's postulate and act as criteria for characterizing learning rules as Hebbian:

1. *locality*, meaning the change of synaptic efficacy must only depend on information such as w_{ij}, v_j and v_i, available locally at the synapse, and not on activity of other neurons or synapses
2. *cooperativity*, implying that any weight change depends of the joint activity of the pre- and postsynaptic neurons, activating simultaneously

The first property suffices for defining the general Hebbian formula for weight change:

$$\frac{dw_{ij}}{dt} = F(w_{ij}, v_i, v_j)$$

Using the second property, we can make F more specific, by expanding it into a Taylor series (about $v_j = v_i = 0$, if it is sufficiently well-behaved):

$$\frac{dw_{ij}}{dt} = c_0(w_{ij}) + c_1^{pre}(w_{ij})v_j + c_1^{post}(w_{ij})v_i + c_2^{pre}(w_{ij})v_j^2 + c_2^{post}(w_{ij})v_i^2 + c_{11}^{corr}(w_{ij})v_iv_j + \mathcal{O}(v^3)$$

So that any term that includes the joint pre- and postsynaptic activity term v_iv_j (e.g. $c_{11}^{corr}(w_{ij})v_iv_j$) satisfies the *cooperativity* condition. Without such a term, the learning rule is called *non-Hebbian.* Generally speaking, all *Hebbian* learning rules derive from this Taylor series, with many combinations of terms being possible (Gestner & Kistler, 2014; Gerstner et al., 2014).

2.2. Plasticity Rules

2.2.1. Original rule

The simplest Hebbian plasticity rule formulation is the prototype, Hebb's rule:

$$F_0 = F(w_{ij}; \theta; v_i, v_j) = cv_iv_j$$
$$\theta = \{c\}$$

Although coefficients may depend on w_{ij} (e.g. c(w_{ij}), e.g. to limit weight growth), they can also be plain constants (c), as is the case for F_0.

2.2.2. Including synaptic depression

Since Hebb’s rule only accounts for weight increase but not for weight decrease, we can use our general formulation to derive a rule that combines synaptic potentiation in addition to synaptic depression:

$$F_1 = F(w_{ij}; \theta; v_i, v_j) = \gamma_2(1 - w_{ij})v_iv_j - \gamma_0 w_{ij}$$
$$\theta = \{\gamma_2, \gamma_0\}$$

Note that in the absence of any stimulation, $v_iv_j = 0$ and thus $\frac{dw_{ij}}{dt} = -\gamma_0 w_{ij}$. In that case, the synaptic weights decay to zero over time. And to represent some constant neuronal background input or baseline level of neurotransmitter release, we can introduce a constant bias coefficient term to F_1 that influences synaptic change independently of the weights and neural activities:

$$F_2 = F(w_{ij}; \theta; v_i, v_j) = \gamma_2(1 - w_{ij})v_iv_j - \gamma_0 w_{ij} + b$$
$$\theta = \{\gamma_2, \gamma_0, b\}$$

2.2.3. Quadratic Hebbian rule

Given the freedom to formulate a wide variety of complex rules, one can increase the flexibility of parameterization. One example of doing so involves taking the Taylor series expansion and defining a quadratic Hebbian rule with six constant coefficient terms:

$$F_3 = F(w_{ij}; \theta; v_i, v_j) = c_0 + c_1v_j + c_2v_i + c_3v_j^2 + c_4v_i^2 + c_5v_iv_j$$
$$\theta = \{c_0, c_1, c_2, c_3, c_4, c_5\}$$

2.2.4. Accounting for neuromodulation

Although not yet fully understood, neuromodulation signaling in the brain, such as reward, novelty and prediction errors are known to improve or steer learning, but also to influence metaplasticity factors (Doya, 2002; Yu & Dayan, 2005; Schultz, 2016; Iigaya, 2016; Farashahi, 2017; Roelfsema & Holtmaat, 2018).

To incorporate neuromodulation into a Hebbian learning context, we can extend a rate-based rule such as F_3 by including a global third factor that represents a signal broadcasted among all neurons during the learning process. This performance-related signal M (e.g. the error) can have a multiplicative influence on the weight change:

$$F_4 = F(w_{ij};\theta;v_i,v_j) = \left(c_0 + c_1 v_j + c_2 v_i + c_3 v_j^2 + c_4 v_i^2 + c_5 v_i v_j\right) M$$
$$\theta = \{c_0, c_1, c_2, c_3, c_4, c_5\}$$

2.3. Tasks

In our meta-learning experiments, we used backprop and Hebbian learning rules F_1, F_2, F_3, F_4 (see 2.2.) to train a noisy sinusoidal wave regression task (*wavereg*): $y = \sin(X) + noise$, where X is a vector of 2048 evenly-spaced values within $[-1.0, 1.0]$ and $noise = \frac{K}{L}$ is a vector where K and L are vectors and $K_i \sim N(0,1), L_i \sim U\,(4,12)$.

Although most of our results (3.1 through 3.4, see chapter 3) come from the *wavereg* task, to show proof of concept in a different context, we also gathered preliminary loss and accuracy data for the first result (3.1) by also learning and meta-learning a binary MNIST classification task (*bmnist*, digits: 0 or 1).

2.4. Models

For the *wavereg* task experiments, we employed a shallow 1-512-1 MLP. Whereas for the *bmnist* experiments, we used a shallow 784-512-2 MLP. The initial weights of both models were initialized using the Kaiming uniform initialization and we chose the tanh as each neuron's non-linear activation function (He et al., 2015).

2.5. Optimization

Our meta-training relied on the condition of initially giving the same rule to each synapse but allowing the rule coefficients to change per synapse. Therefore, after meta-training, each synapse ended up with unique sets of rules (coefficients) per synapse. For specific training and meta-training details (e.g. learning rates), see **Appendix A** - Implementation details.

2.6. Meta-learning

```
Algorithm 1: Metalearning algorithm
for $mep \leftarrow 0, \ldots MAXMEP$ do
    $student\_model \leftarrow clone\,(teacher\_model)$
    $\Delta W_{GD} \leftarrow train\,(teacher\_model, method = backprop, epochs = 1)$
    $\Delta W_{HEBB} \leftarrow train\,(student_model, method = F_*, epochs = 1)$
    $L_{\Delta W} = L\,(\Delta W_{GD}, \Delta W_{HEBB})$
    $\theta \leftarrow \theta - \eta \nabla_\theta L_{\Delta W}$
end for
```

Starting with two identical MLP networks, within an outer meta-learning loop, we iterate over a number of meta-learning epochs (*mep*s). During each *mep*, within an inner learning loop, we train for 1 epoch the first network using backprop and the second network using a parametrized Hebbian learning rule $F_* \in \{F_1, F_2, F_3, F_4\}$ (see 2.2.), both in an online manner. After training them both for 1 training epoch, we calculate the corresponding weight delta tensors, ΔW_{GD} and ΔW_{HEBB} and pass them to a loss function and obtain a differentiable loss $L_{\Delta W}$: the *meta-loss*. Then, we automatically calculate the gradient of the *meta-loss* w.r.t. the coefficient parameters (θ) of the Hebbian rule ($\nabla_\theta L_{\Delta W}$), using an automatic differentiation tool (e.g. torch.autograd). At the end of the *mep* iteration, using this gradient ($\nabla_\theta L_{\Delta W}$) and a meta-optimizer (e.g. GD-based), we calculate a $\Delta\theta$ tensor (e.g. $-\,\eta \nabla_\theta L_{\Delta W}$) and update θ. This process repeats for a chosen number of *meps*.

Upon completion, we derive a modified, meta-learned Hebbian rule that is theoretically better for training the same model on a specific task than the one we started with. Meaning, a rule that yields weight change vectors ΔW_{HEBB} more closely aligned with ΔW_{GD}. After meta-learning, we produce coefficients that constitute a Hebbian learning rule that can approximate GD performance using only feedforward passes, without backward, error-propagating passes required during the training phase. For an intuitive schematic of the meta-learning process, see Fig. 2.

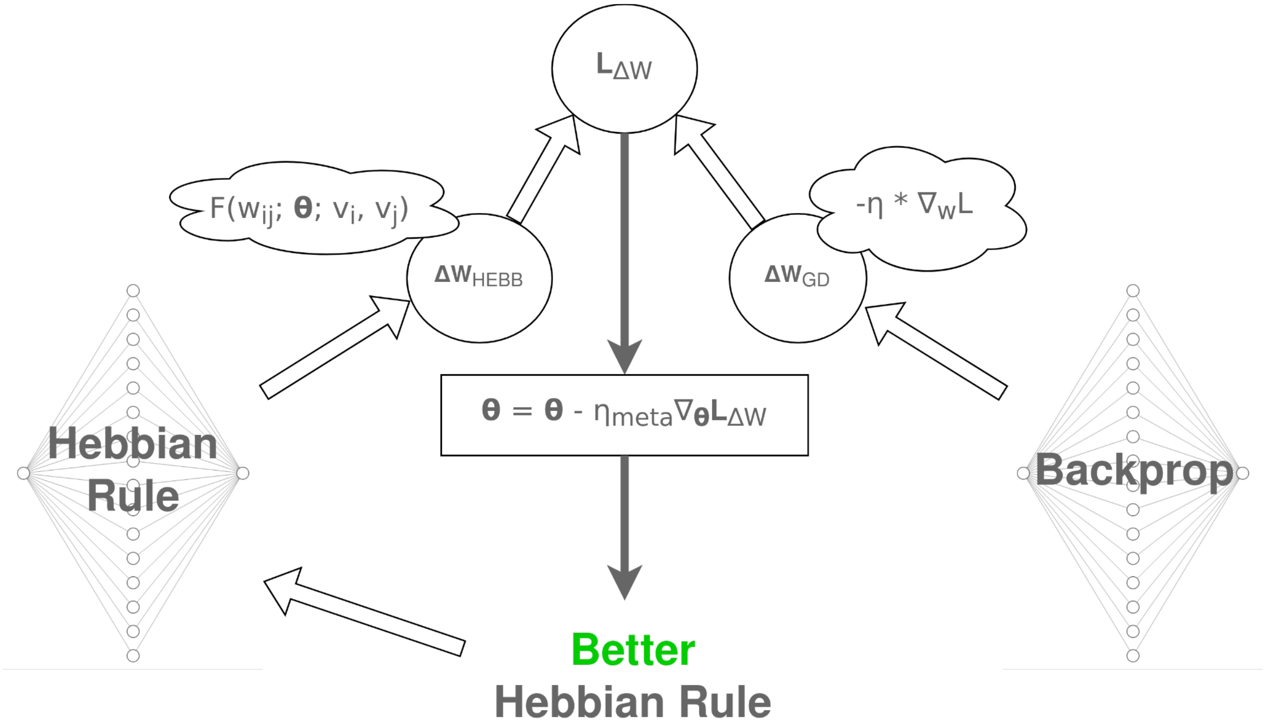

$L_{\Delta W}$
$F(w_{ij}; \boldsymbol{\theta}; v_i, v_j)$
$-\eta * \nabla_W L$
ΔW_{HEBB}
ΔW_{GD}
$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta_{meta} \nabla_{\boldsymbol{\theta}} L_{\Delta W}$
Hebbian Rule
Backprop
Better Hebbian Rule

Fig. 2. Meta-learning algorithm. Schematic of the meta-training iteration: two identical networks are trained for one training epoch, one with a Hebbian learning rule and the other with GD (using backprop). After training, a loss we call the *meta-loss* $L_{\Delta W}$ is calculated using the ΔW_{HEBB} and ΔW_{GD} weight delta vectors. Using $L_{\Delta W}$, the vector of the coefficient parameters of the Hebbian learning rule θ is optimized (e.g. using GD). Repeating this process optimizes the Hebbian learning rule. F: Hebbian learning rule function, i: presynaptic neuron index, j: postsynaptic neuron index, w_{ij}: synaptic weight between neuron i and neuron j, θ: coefficient parameters of Hebbian learning rule function, v_j: activity of presynaptic neuron j, v_i: activity of postsynaptic neuron i, η: learning rate used during GD training, η_{meta}: learning rate used during GD meta-training.

3. Results

3.1. Meta-learned Hebbian plasticity approximates GD

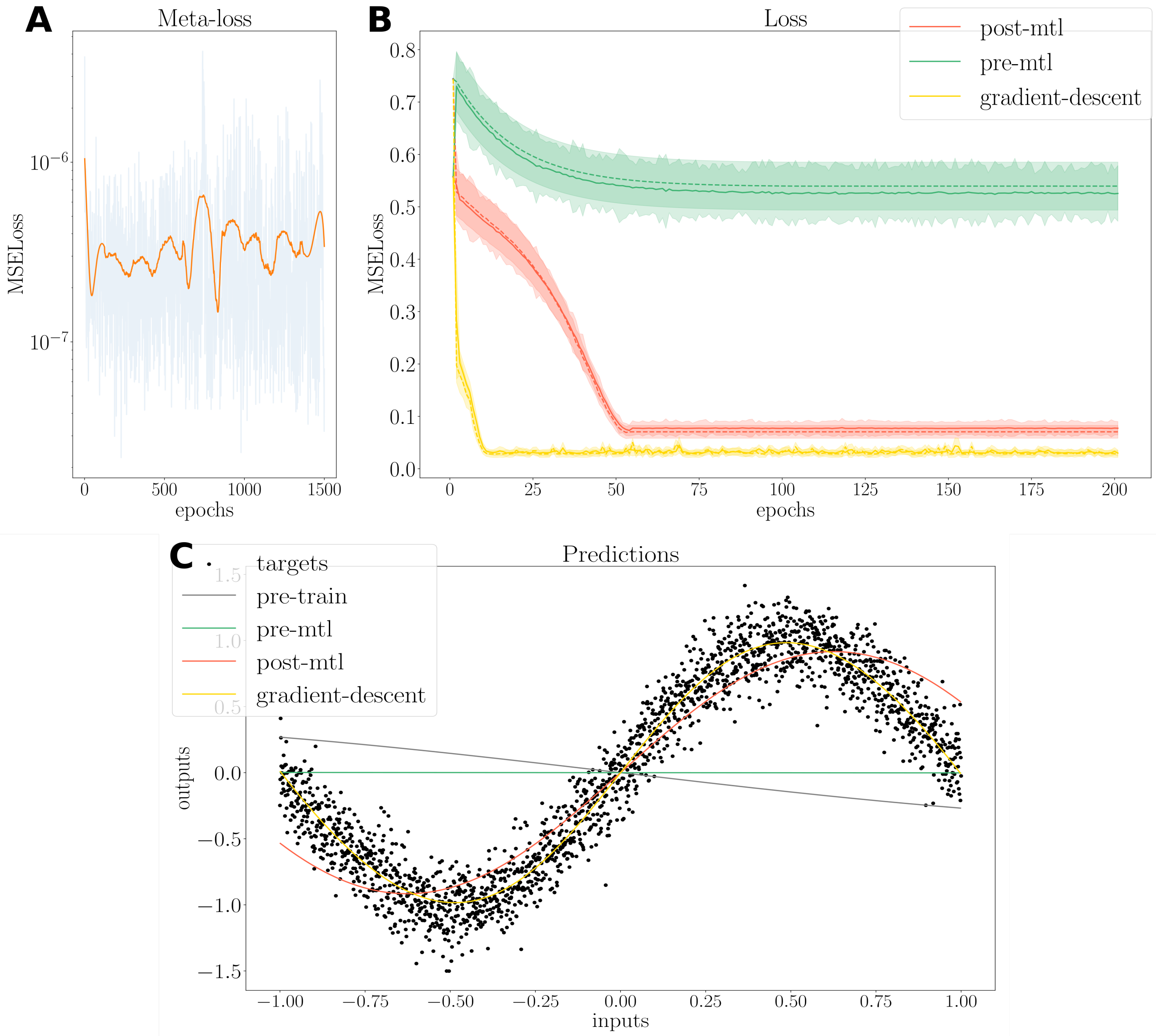


Fig. 3. Meta-learned rule F_1 approximates GD on the *wavereg* task. A. Smoothed meta-loss in log scale (epochs imply *meta-training* epochs). B. Training (solid) and validation (dotted) loss (epochs imply *training* epochs). C. Prediction curves.

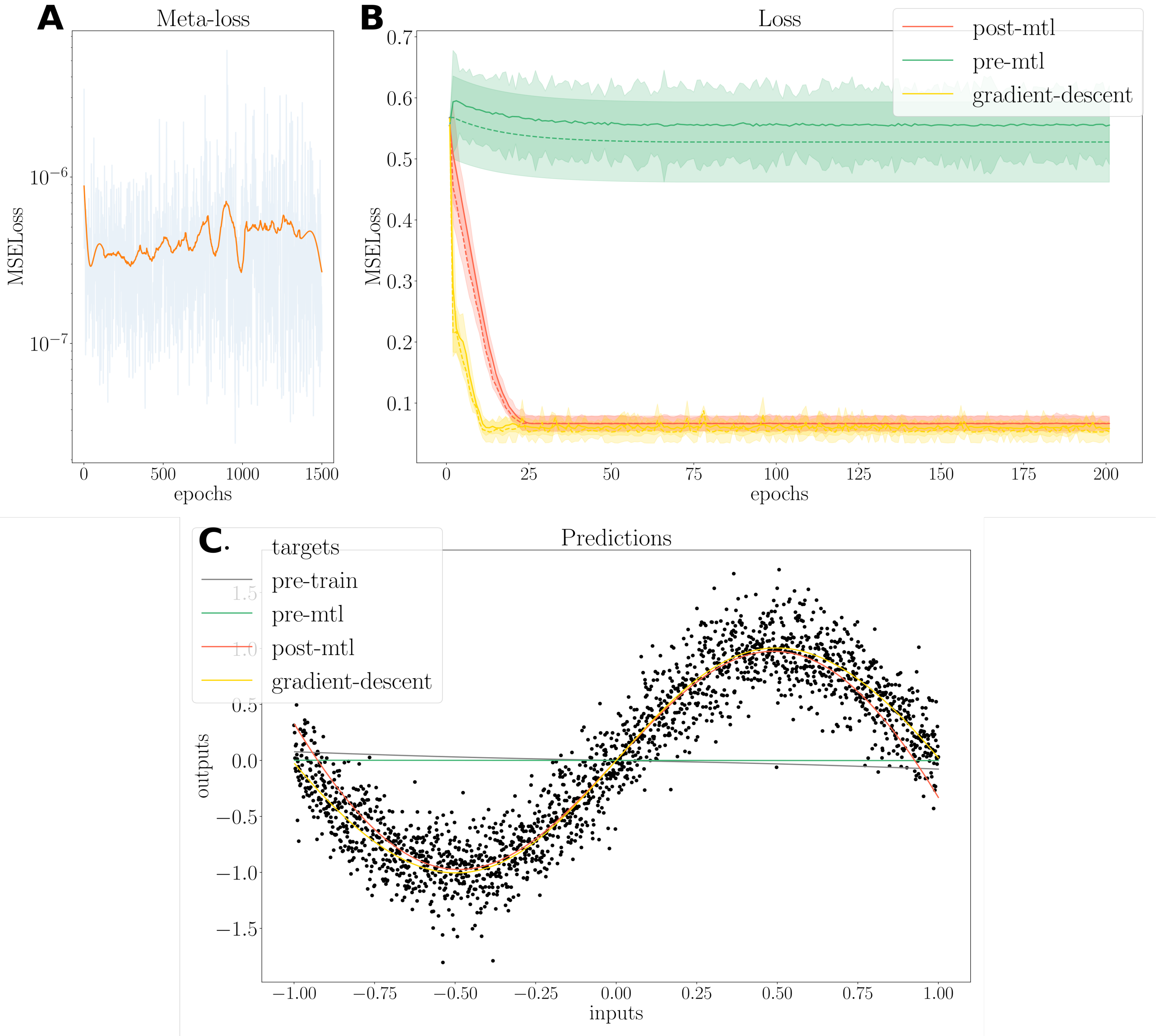


Fig. 4. Meta-learned rule F_2 approximates GD on the *wavereg* task. A. Smoothed meta-loss (epochs imply *meta-training* epochs). B. Training (solid) and validation (dotted) loss (epochs imply *training* epochs). C. Prediction curves.

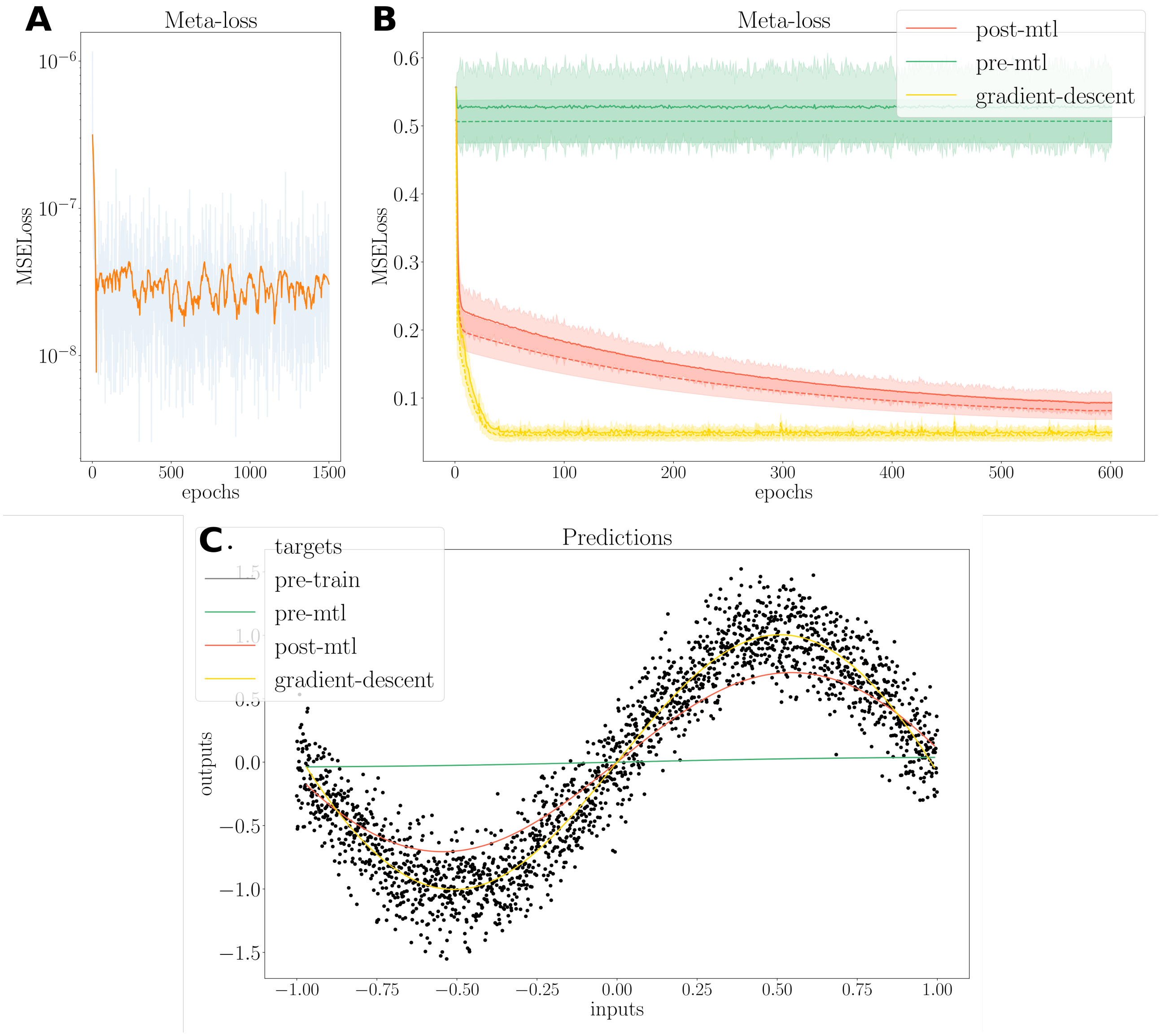


Fig. 5. Meta-learned rule F_4 approximates GD on the *wavereg* task. A. Smoothed meta-loss in log scale (epochs imply *meta-training* epochs). B. Training (solid) and validation (dotted) loss (epochs imply *training* epochs). C. Prediction curves.

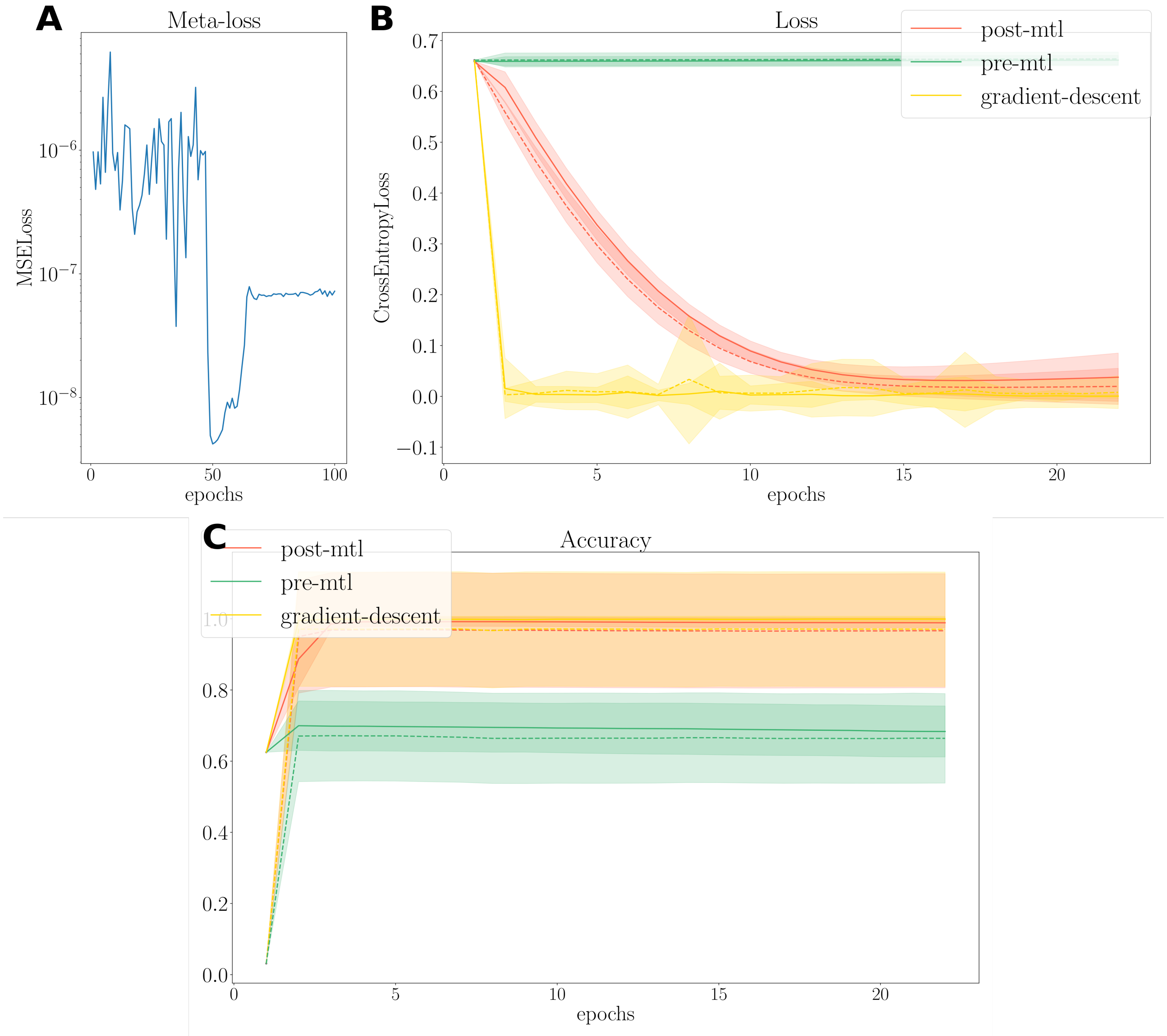


Fig. 6. Meta-learned rule F_3 approximates GD on the *bmnist* task. A. Meta-loss in log scale (epochs imply *meta-training* epochs). B. Training (solid) and validation (dotted) loss (epochs imply *training* epochs). C. Training (solid) and validation (dotted) accuracy.

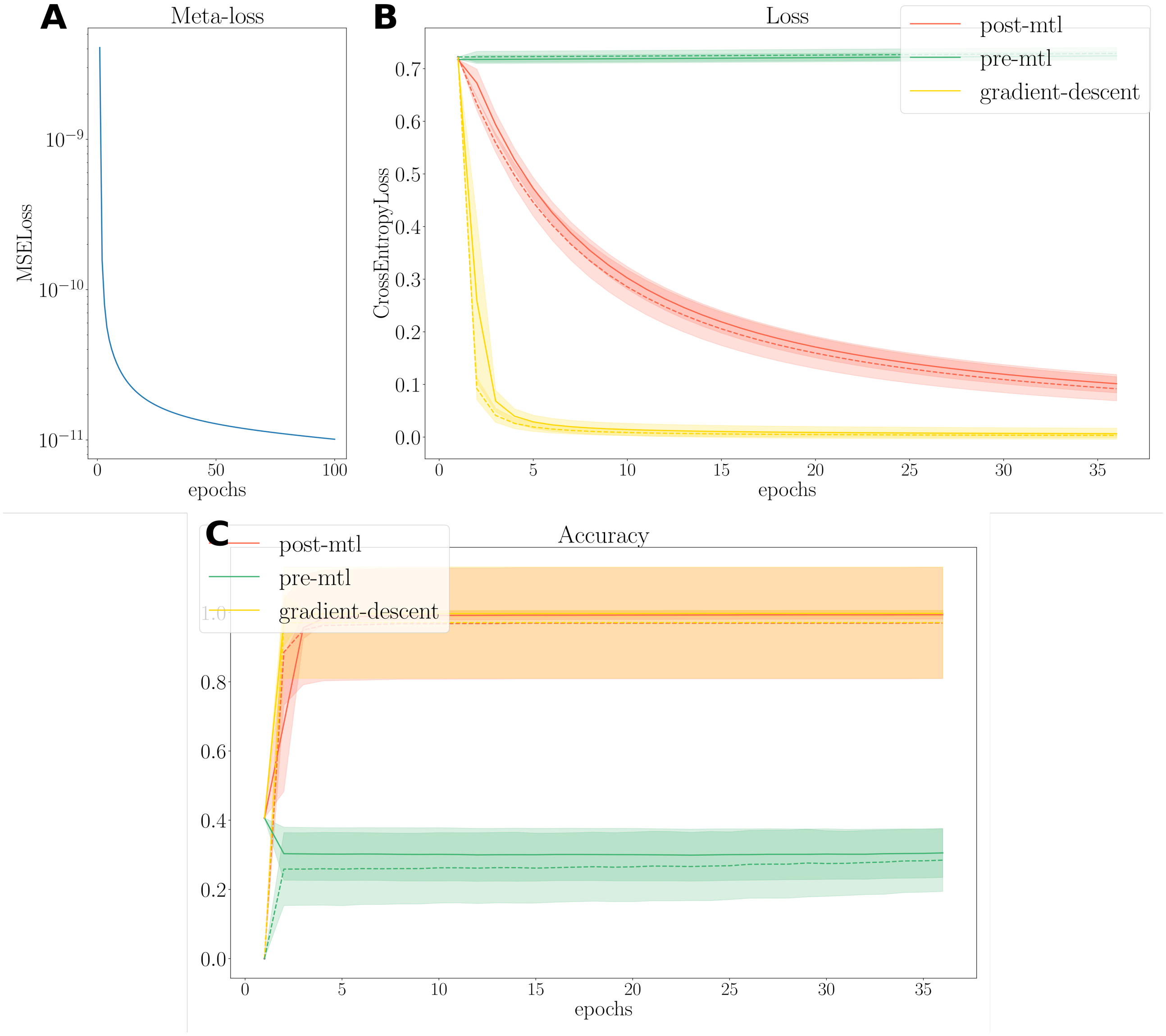


Fig. 7. Meta-learned rule F_4 approximates GD on the *bmnist* task. A. Meta-loss in log scale (epochs imply *meta-training* epochs). B. Training (solid) and validation (dotted) loss (epochs imply *training* epochs). C. Training (solid) and validation (dotted) accuracy.

To learn the *wavereg* task using Hebbian plasticity, we first meta-trained the learning rules F_1, F_2 and F_4 for 1500 epochs each (see sections 2.2 and 2.6). In subfigures 3A, 4A and 5A, the small drop in the meta-loss curve signifies that the error between Hebbian and GD weight delta vectors becomes approximately smaller as meta-training progresses. After meta-learning, the optimized rules F_1**,** F_2 and F_4 effectively approximate GD during training compared to the respective ineffective performance of their non-meta-learned counterparts (see loss subfigures 3B, 4B, 5B and output subfigures 3C, 4C, 5C). Both rules F_1 and F_2 were trained for 200 training epochs with learning rate decay enabled. Whereas, F_4 was early-stopped after being trained for 600 epochs. Although we carried out experiments on rule F_3 as well (see Sup. Fig. 1, **Appendix C**), including the global error third factor (as in F_4) turned out to be noticeably more effective in terms of loss convergence.

For the *bmnist* task, we meta-learned rules F_3 and F_4 for 100 epochs each and produced meta-loss curves that were very unstable, in the one case, and surprisingly smooth in the other case (see subfigures 6A, 7A). Despite this discrepancy, the loss curve of meta-learned rule F_3 converges close to GD's at around 20 training epochs, whereas the loss curve of the optimized rule F_4 still continues to slowly decrease even at training epoch 35, without having fully converged (see subfigures 6B, 7B). Nevertheless, for both meta-learned rules, practical approximation of GD is achieved if one judges by the indistinguishable high-accuracy values that the post-meta-learning accuracy curves reach, which almost approach GD's.

3.2. Meta-learned Hebbian coefficients vary slightly across synapses

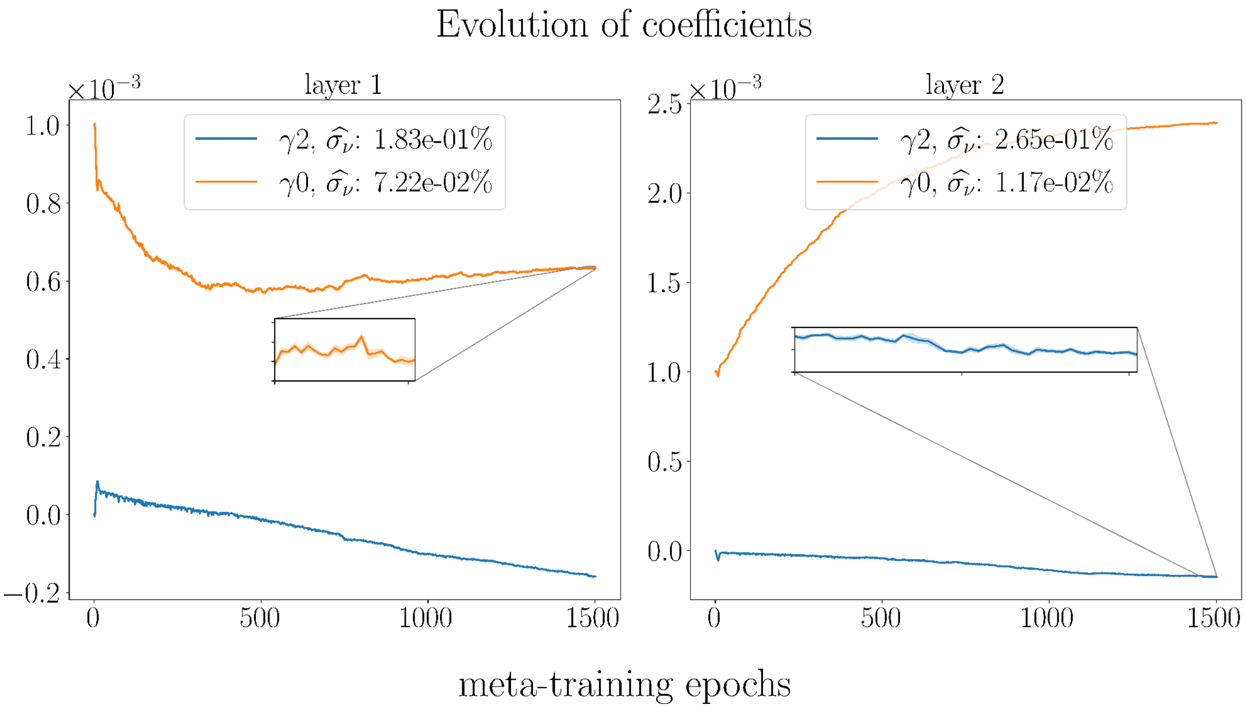


Fig. 8. Evolution of the mean value of rule F_1 coefficients γ_2, γ_0 across synapses during meta-training on the *wavereg* task. Variability around the mean plotted as standard deviation (σ) shade around the mean and quantified using the percentage of $\widehat{\sigma_v}$: the coefficient of variation, also known as the relative standard deviation (inset plots are zoomed regions).

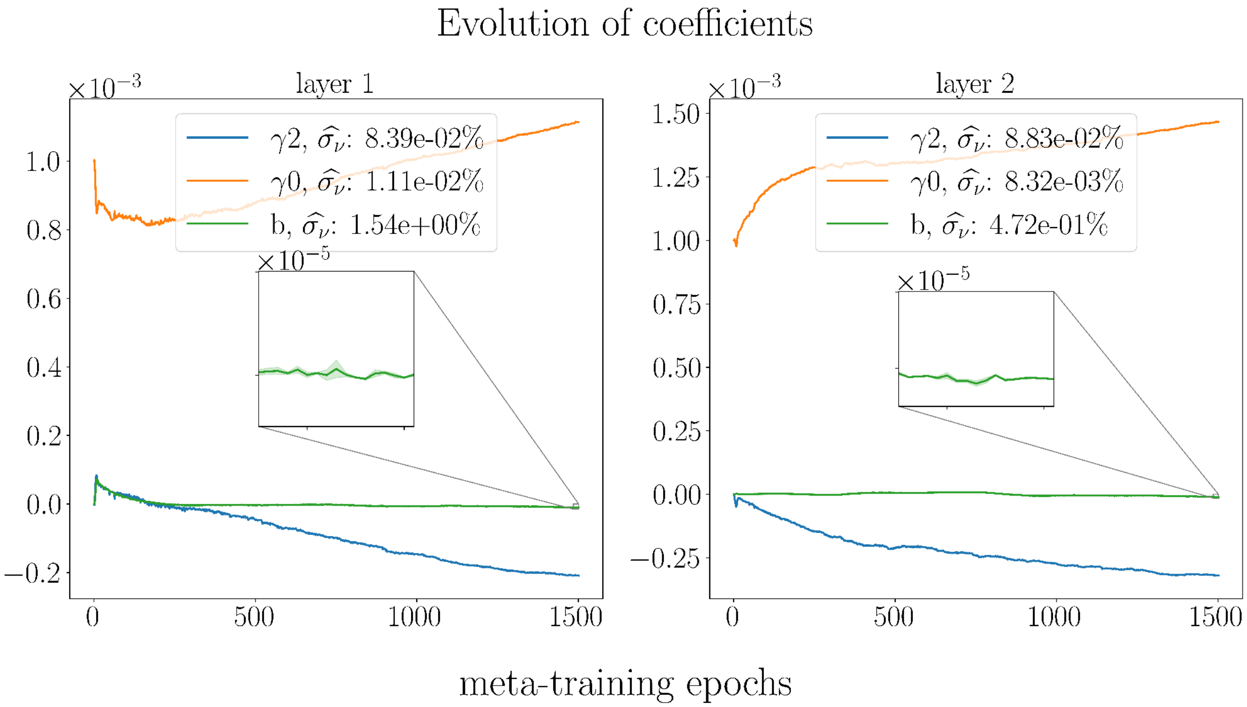


Fig. 9. Evolution of the mean value of rule F_2 coefficients γ_2, γ_0, b across synapses during meta-training on the *wavereg* task. Variability around the mean plotted as standard deviation (σ) shade around the mean and quantified using the percentage of $\widehat{\sigma_v}$: the coefficient of variation, also known as the relative standard deviation (inset plots are zoomed regions).

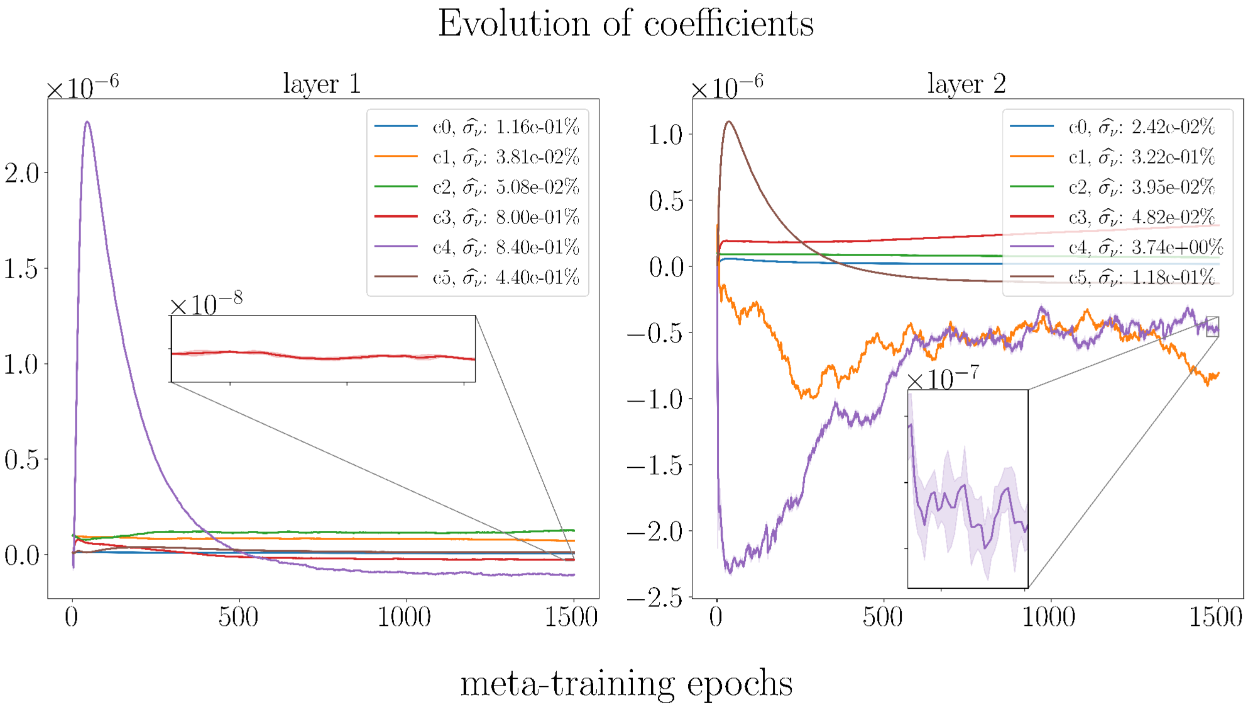


Fig. 10. Evolution of the mean value of rule F_4 coefficients c_0, c_1, c_2, c_3, c_4, c_5 across synapses during meta-training on the *wavereg* task. Variability around the mean plotted as standard deviation (σ) shade around the mean and quantified using the percentage of $\widehat{\sigma_v}$: the coefficient of variation, also known as the relative standard deviation (inset plots are zoomed regions).

Figures 8-10 depict the evolution during meta-learning of the mean values of coefficients of Hebbian learning rules F_1, F_2, F_4 (see sections 2.2.2 and 2.2.4), as well as the standard deviation as a shaded area around those means (std), across all synapses per layer. From these plots, two are the most important takeaways.

First, close to the end of meta-training, we notice the very small shaded std areas around the mean, as seen through the zoomed inset plots for selected coefficients. We quantified these using a percentage of the coefficient of variation, $\widehat{\sigma_v}$ (also known as the relative std). These values, calculated after meta-training, represent a standardized measure of the variability around the mean. Such tiny $\widehat{\sigma_v}$ values, with a maximum of 3.72% (for c_4 in layer 2 of Fig. 10), clearly illustrate very minute variabilities for each coefficient across synapses of the same layer. This means that although each learning rule term multiplier is free to fluctuate for each synaptic connection, the optimization process settles on solutions with coefficient values that vary only slightly among the mean of all synapses in the same layer.

The other key observation is that all meta-learned Hebbian learning rules F_1, F_2 and F_4 are characterized by prevailing Hebbian and Anti-Hebbian terms whose interplay during learning facilitates the desired outcome of GD approximation. Such a phenomenon is intriguing, perhaps alluding to the emergence of learning components of different neuron groups (e.g. slow- and fast-firing) that are responsible for long-term potentiation (LTP, strengthening of synapses) and long-term depression (LTD, weakening of synapses), respectively. Further analysis could reveal which synapses in particular are responsible for either form of plasticity.

3.3. Distributions of optimized coefficients depend on initial model weights

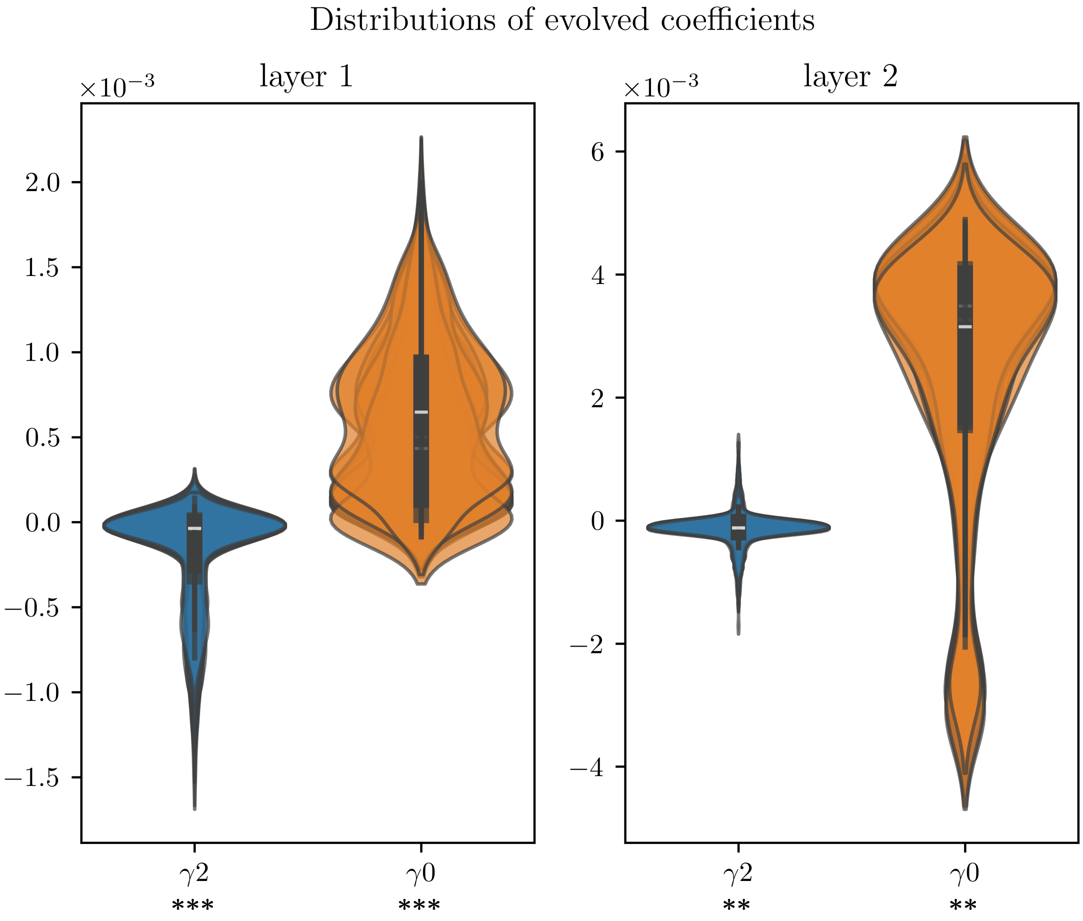


Fig. 11. Overlayed distributions of F_1 coefficients γ_2, γ_0 across synapses per layer, meta-learned on the *wavereg* task (10 random runs of same model with different initial weights). Each set of overlayed coefficient distributions is marked by a statistical significance that marks the statistical difference between runs.

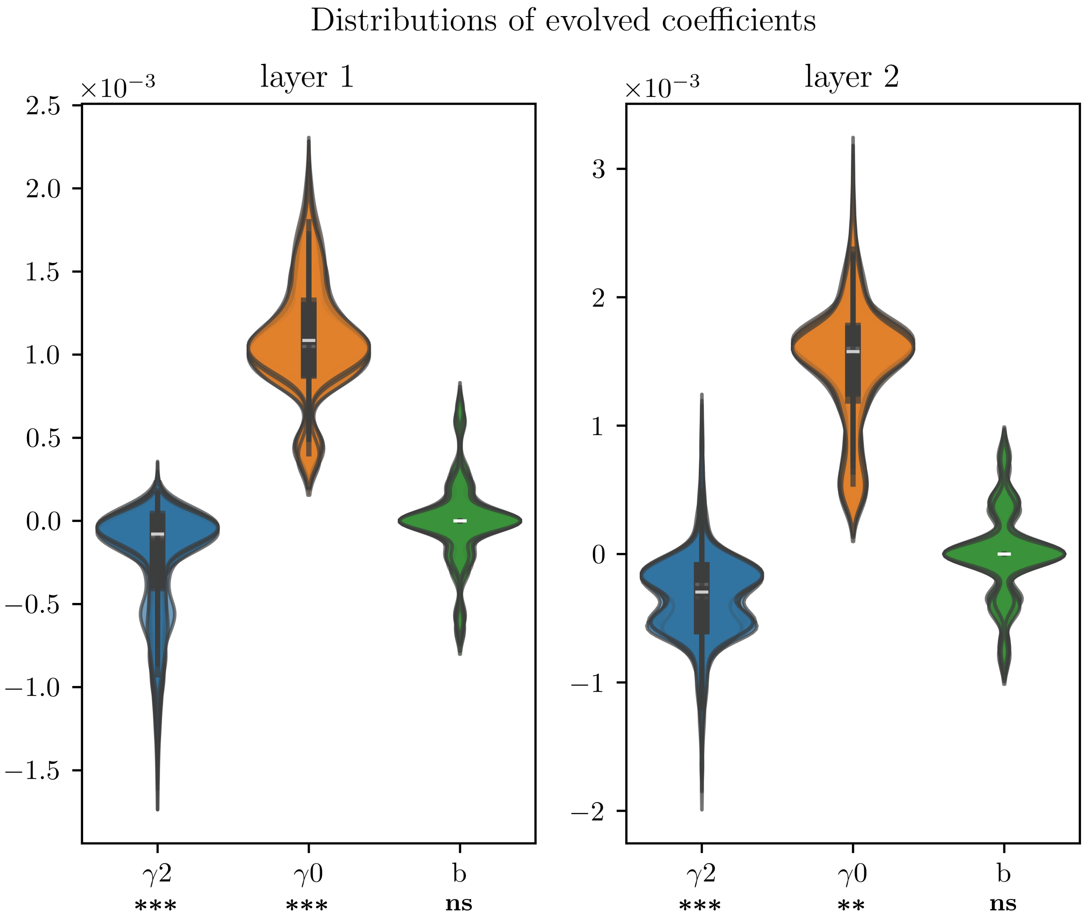


Fig. 12. Overlayed distributions of F_2 coefficients γ_2, γ_0, b across synapses per layer, meta-learned on the *wavereg* task (10 random runs of same model with different initial weights). Each set of 10 overlayed coefficient distributions is marked by a statistical significance that marks the statistical difference between runs.

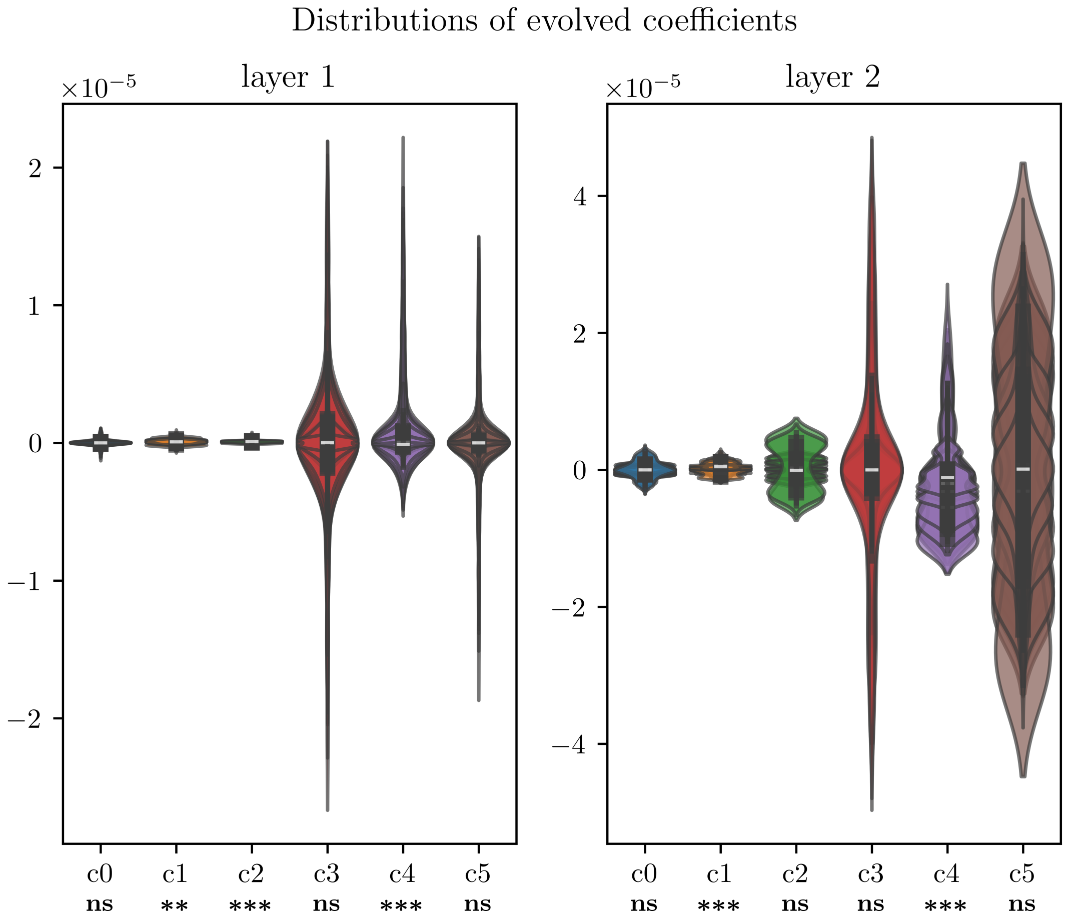


Fig. 13. Overlayed distributions of F_4 coefficients c_0, c_1, c_2, c_3, c_4, c_5 across synapses per layer, meta-learned on the *wavereg* task (10 random runs of same model with different initial weights). Each set of overlayed coefficients is marked by a statistical significance that represents the statistical difference between runs.

One of the main questions that arose after achieving favorable results (i.e. GD approximation) was whether the resulting optimized coefficients originated from distributions that are independent of initial model weights. To test this hypothesis, we meta-learned each rule F_1, F_2, and F_4 (see sections 2.2 and 2.6) 10 different times, with a unique seed and thus a unique model with different initial weights per run (Figures 11-13). Then, we:

1. overlayed the distributions (violin plots) of the meta-learned coefficients
2. ran, for each set of 10 coefficient distributions, the Kruskal-Wallis test and calculated a statistical significance (p-value) that determines whether the coefficient samples originate from the same underlying distribution (Kruskal & Wallis, 1952)

For F_1 (Fig. 11) and F_2 (Fig. 12), although the overlayed coefficient distributions intuitively look quite similar, the corresponding high statistical significance indicates that at least one of them stochastically dominates over another one among the 10 (with the exception of F_2's b coefficient). Whereas, for F_4 (Fig. 13), the results are mixed, with around half of the coefficient distributions satisfying the null hypothesis and the rest containing at least one distribution that is significantly different. Under the assumption that the chosen statistical test is appropriate for addressing our question, these results sufficiently falsify our hypothesis, demonstrating that for the *wavereg* task the distributions of meta-learned rule coefficients depend on the initial weights of the underlying model used for training and meta-training.

3.4. Meta-learned rules can generalize to models with different initial weights

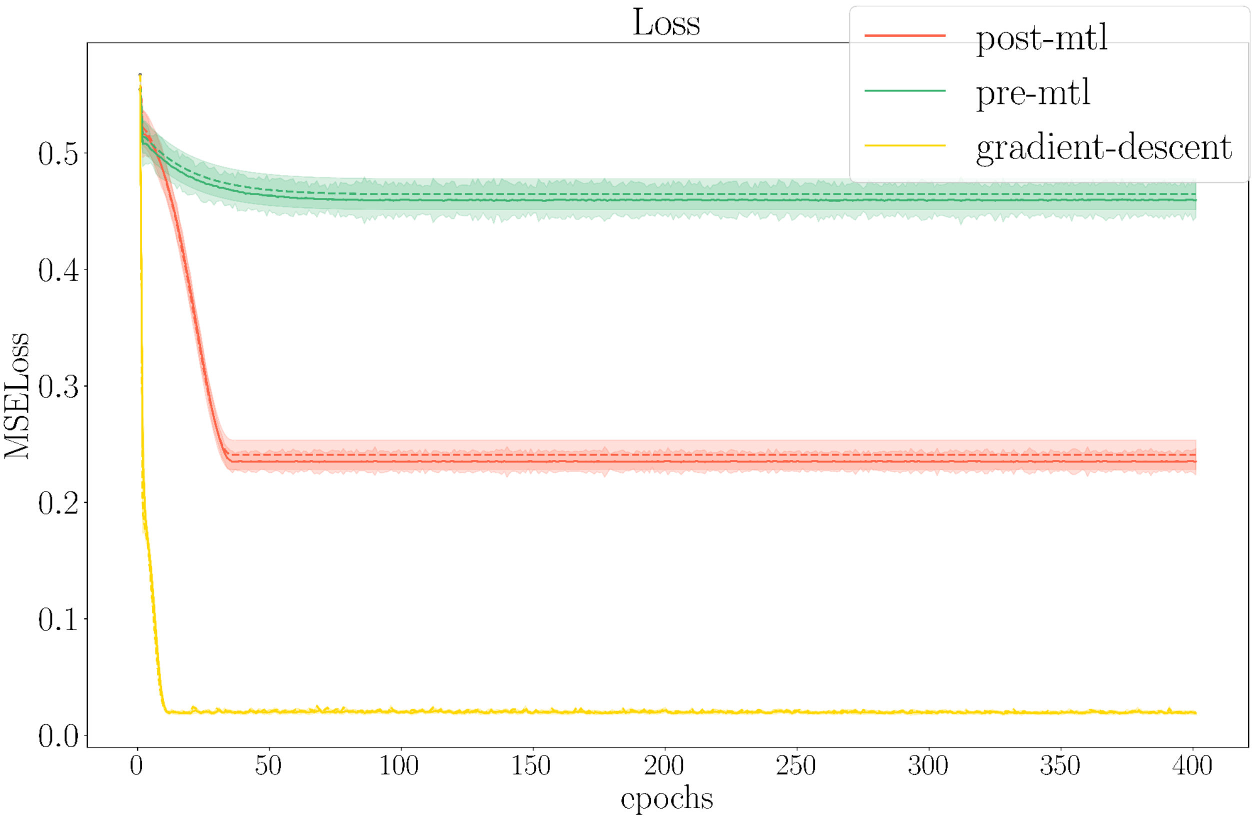


Fig. 14. Mean training (solid) and validation (dotted) loss using pre-optimized rule F_2 coefficients to learn the *wavereg* task on 10 unseen models with different initial weights.

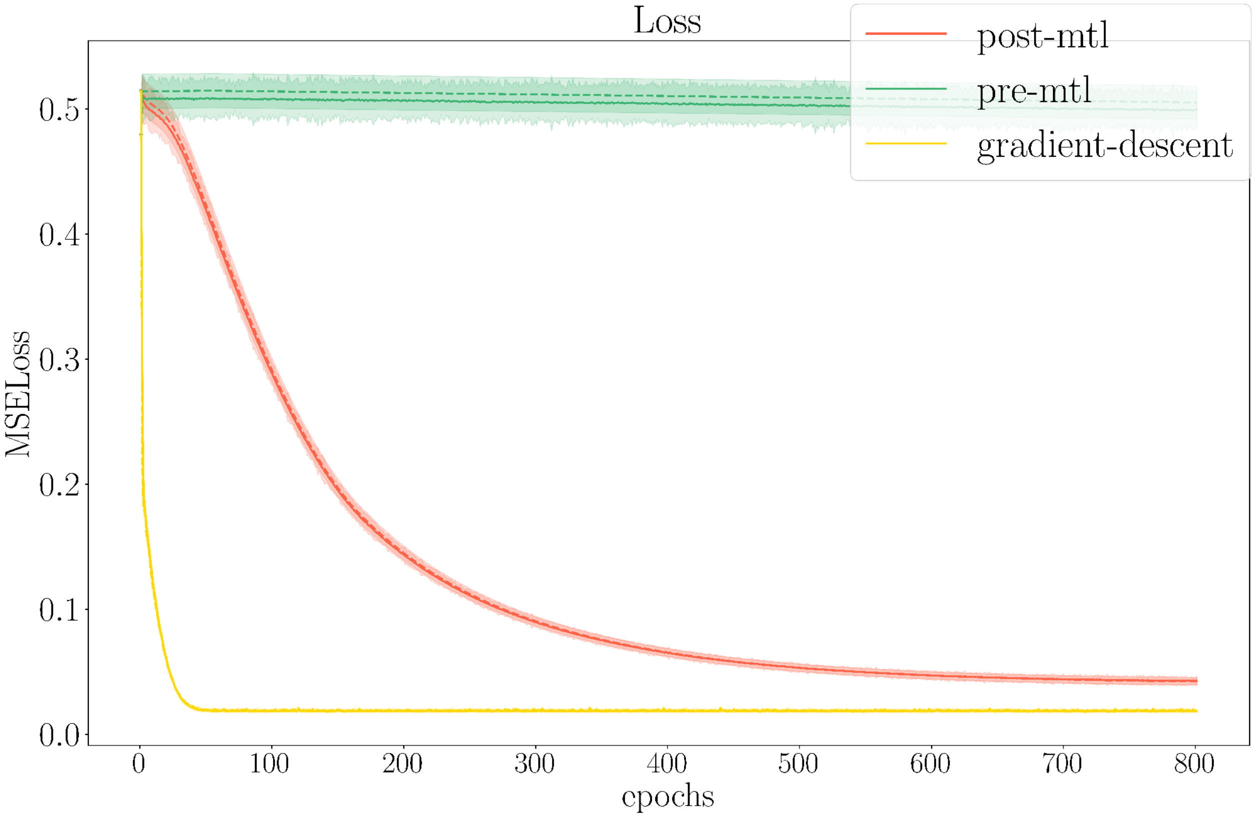


Fig. 15. Mean training (solid) and validation (dotted) loss using pre-optimized rule F_4 coefficients to train the *wavereg* task on 10 unseen models with different initial weights.

Next, we addressed the question: can meta-learned rules (optimized coefficients) trained on one model A successfully train on the same task using other models B, C, D, etc. that have different initial weights, without the need to repeat the meta-training process? Despite a clear indication from results 3.3 that our meta-learning algorithm yields coefficient distributions that depend on initial model parameters, the results of 3.4 affirm this question.

Here we meta-learned rules F_1, F_2 and F_4 (see sections 2.2 and 2.6) and used them to train the *wavereg* task on 10 new, unseen models, all initialized with different initial weight values. We then plotted the mean of the GD, pre- and post-meta-learning training and validation loss curves for each rule. After discarding inadequate F_1 results (not shown), we produced Figures 14 and 15 for rules F_2 and F_4, respectively.

F_2 and, more profoundly, F_4 suggest that our meta-learning algorithm can in fact yield sets of optimized rules capable of training models with different initial weights on the same task without requiring further meta-training on those unseen models, effectively yielding a form of generalized learning independent of initial model weights.

4. Discussion

4.1. Summary

The exact mechanisms that allow biological NNs to assign credit to their neurons and to learn efficiently are still unknown. Yet, the method that enables ANNs to learn effectively is known as backprop and its success can be attributed to the ability to increase performance by minimizing a clearly-defined loss function via GD optimization. Naturally, the wide adoption of GD for training ANNs through backprop has led to speculations about the existence of corresponding neurophysiological mechanisms that either approximate or surpass GD and as a result facilitate successful learning behaviors in brains.

Recently, researchers have suggested that gradients may be a surprisingly appropriate mathematical paradigm for studying synaptic plasticity and that the brain's learning functionalities may support gradient calculation (Richards & Kording, 2023). However, since learning in the brain most likely relies on the interplay of a vast array of complex mechanisms such as Hebbian plasticity, inhibition, high-frequency spike bursting and reward modulation, biologically-plausible learning rules that can train NNs as effectively as GD (if at all) are not guaranteed to be easily discoverable, formulatable or interpretable by human researchers. In fact, the perplexing variability found in complex natural systems (e.g. mammalian nervous systems) makes it intuitively unlikely that even neighboring synapses strengthen or weaken in *exactly* the same way (e.g. one shared plasticity rule). In that sense, although a human can indeed design mathematical plasticity models and tune their coefficients by hand, their neurophysiological plasticity correlates in brains are immensely more complex and thus most likely arise and are dynamically tuned by evolutionary processes or by compensatory processes acting within the lifespan of the organism (e.g. during development or learning).

Over the course of millennia, mutations to neurophysiological mechanisms lead to more adaptable learning capabilities in animals. Wanting to computationally investigate how the brain may be efficiently assigning credit, we set forth to mimic such overarching evolutionary processes by exploring whether associative plasticity models can approximate GD and under which conditions. To these ends, we implemented a meta-learning system that optimizes a set of Hebbian learning rules with many degrees of freedom. We then demonstrated the condition under which these rules can approximate GD-like performance on two non-trivial tasks. Additionally, in order to achieve the normative goal of approaching gradient-based learning, we used meta-learning in an online manner to automatically find optimal rule coefficients that minimize the error between Hebbian and GD weight delta quantities.

Our results indicate that the key condition for Hebbian learning to approximate GD is assigning per synapse a unique set of coefficients of the same rule that slightly vary across all synapses of the same layer. This *condition* arose out of attempts to learn the *wavereg* task with *layer-specific* plasticity rules (that share rule coefficients per layer) in which mode, we tried both to optimize the per-layer coefficients but also to set their values to the mean of all optimized coefficient values from our *synapse-specific condition*. However, approximating GD performance in "*layer mode*" proved unsuccessful (see Sup. Fig. 2, **Appendix C**). Furthermore, however, we found that the distributions of optimized coefficients depend on initial model weights, but also that meta-learned Hebbian rule F_4 can be used to learn the *wavereg* task on models with different initial weights: a form of generalization.

4.2. Interpretation

In our work, the *condition* for approximating GD-like performance is meta-learning *synapse-specific* Hebbian plasticity rules for training shallow MLPs. When discussing the biological implications of our findings under this *condition*, the matter of synaptic heterogeneity becomes central. What does it mean for a synapse to change its efficacy according to a rule whose coefficient parameters are unique to that synapse and not shared among others? And what does it mean for the values of such coefficients and the rules that they constitute to slightly differ between synapses of the same layer?

The brain's intricate assembly of circuits made up of diverse neurons, each with their unique dendritic structures, varying membrane and spine properties and distinct ionic channel distributions, densities and conductances, casts doubt on a hypothetical singular learning rule that changes all synapses of one or more of its networks in the same manner. Although plasticity rules can undoubtedly be modeled as unambiguous functions based on underlying blueprints containing associativity and other activity-related terms, strict mathematical formulations may just be a convenience for modelers. If one considers the immense complexity of neurons, quite apparent even at the synaptic level, one-size-fits-all learning rules may just only be able to encapsulate extremely crude approximations of useful effects of plasticity processes, especially in networks that lack realistic biophysical structures or error-carrying feedback mechanisms, such as sparse burst multiplexing: a theoretical framework that uses dendrite-dependent bursting combined with short-term plasticity to support multipleing of feedback signals from higher-order areas and feedforward signals, wherein the former can steer synaptic plasticity without affecting the communication of the latter (Payeur et al., 2021).

When neuron-specific instructive signals that correlate with the gradient of an objective function that quantifies performance are not calculated or utilized during learning, Hebbian plasticity rules lack the appropriate data to accurately assign credit at the fine-grained neuronal level. In such unsupervised or globally supervised contexts, optimizing a rule that is shared among all synapses of a network or layer may not be as effective a method as the optimization of distinct rules that are uniquely assigned to each synapse. Within this latter setting, the inherent advantage of using a supervised teacher method such as backprop to meta-learn a student network's learning rules is that the many synapse-specific coefficients parameters cumulatively become adequate encoders (e.g. through meta-plasticity or genomically through evolution) that can capture context-related information for one or more tasks (e.g. combination of input and target data). In other words, our meta-learning method under this *condition* provides a way to encode the dynamics of a teacher network doing GD (e.g. direction of gradients) into many synaptic plasticity parameters: the Hebbian rule coefficients. Moreover, effective meta-learning under this *condition* may also be a compensation for useful neurophysiological mechanism diversity that is usually found in real neurons and circuits but that is absent from our simplistic perceptron networks.

4.3. Neuronal heterogeneity & learning

There is no guarantee that learning in the brain obeys a simple set of principles. If no two synapses are exactly the same, then, by necessity, no two neurons are the same. Each synapse and, as a consequence, each neuron behaves and responds, at least minutely, uniquely to the same stimuli. This means that since the inherent structural differences of real synapses translate to a necessary behavioral heterogeneity of post-synaptic responses, it only makes sense that, however slight those deviations (e.g. among neighboring synapses of same-type neurons), this diversity is reflected in synapse-specific learning rules that govern plasticity. More simply put, if each synapse and by extension each neuron is physically dissimilar, each one must yield slightly unique responses. Then, by necessity, every synapse must also *learn* distinctly, with plasticity learning rules that are at least marginally different.

Computational neuroscience research has only just begun to skim the surface of the benefits of heterogeneous elements in learning with ANNs (Doty et al., 2021). A recent study supports the idea that neural heterogeneity may extend beyond being a mere byproduct of noisy processes to potentially serving a fundamental role in allowing animals to learn flexibly under uncertainty. It may also contribute to metabolically efficient strategies that allow brains to represent richer sets of functions (Perez-Nieves et al., 2021). Along adjacent research directions, Choudhary et al., 2023 showed that meta-learning different activation functions can enhance conventional NNs by making them more expressive and accurate. As it turns out, discovering heterogeneous arrangements of activations outperforms homogeneous counterparts and facilitates adaptability through efficient function approximation. Similarly, in our work we meta-learned heterogeneity in the form of slight coefficient variability of synaptic learning rules which facilitates the approximation of GD performance. Our emerging useful diversity perhaps compensates for the lack of feedback pathways and other realistic elements within the rich diversity of neuronal circuitry not considered in our experiments, such as dendritic processing and spiking dynamics.

Despite work highlighting the learning benefits of neuronal heterogeneity in ANNs, there has also been experimental support for the explicit statement that neuronal correlates of Hebbian learning rule coefficients might vary across synapses. Basal dendrites of neocortical pyramidal neurons for instance, contain two compartments with distinct plasticity rules, potentially supporting different learning functions (Gordon et al., 2006). Whereas, in spike timing-dependent synaptic plasticity, learning rules depend on synapse location within the dendritic tree, with synapses undergoing plasticity according to local rather than global learning rules (Letzkus et al., 2006). Another study on locally dynamic synaptic learning rules in pyramidal neuron dendrites found that LTP at individual synapses reduces the threshold for LTP at nearby synapses, indicating that synapses within distinct dendritic neighborhoods may undergo plasticity differently (Harvey & Svoboda, 2007). Further empirical findings and control-theoretic considerations suggest that compensatory meta-plasticity processes capable of counteracting the effects of Hebbian plasticity must act on similar or even faster timescales, hinting at the variability of branch- or even synapse-specific plasticity (Zenke & Gerstner, 2017; Zenke et al., 2017). More recently, one study showed that sequence learning with narrow learning windows requires overlapping firing fields, which suggests that neighboring synapses may undergo plasticity using slightly different learning rules (Reifenstein et al., 2021). Whereas, Yaeger et al. 2022 highlighted that cortical neurons possess dendrite-specific integration and plasticity rules that may enable robust yet flexible cortical computation in adults. Together, these studies suggest that Hebbian learning rules may vary across synapses, supporting the argument that neighboring synapses learn via slightly distinct learning rules.

Moreover, the work of Eve Marder and her collaborators raises many arguments in favor of heterogeneity's central role in adaptive behavior. Based on their findings of two- to sixfold variation of synaptic parameters (e.g. conductance densities) in same-type cells, they emphasized that inherent variability may favor multiple beneficial solutions (sets of parameters) for neurons to produce similar circuit performance, allowing organisms to respond flexibly in altered environments and maintain resilience. They also presented evidence that variation in neuronal excitability across neurons of same populations potentially serves to increase a circuit's information transfer capabilities (Marder & Goaillard, 2006; Marder & Taylor, 2011; Marder, 2011). In later work, they mentioned that diversity may enhance a network's ability to respond to a broader range of inputs. One that would otherwise be narrower if for example neurons were more similar to one another. Adding that, rather than hindering network function, the high variability of ion channel properties from neuron to neuron increases biophysical diversity, which can in turn improve network information storage and robustness in response to perturbations (Goaillard & Marder, 2021). Their work highlights the idea that evolution diversifies neurons, leading to a variability of multiple solutions that plays a key role in flexible, resilient behavior. Along similar lines, through a computational meta-learning example, our work shows that variability at the synaptic level of coefficients that could theoretically correlate with metaplasticity parameters, favors efficient learning (e.g. approximation of GD), broadening the aforementioned argument of the significance of neuronal variability in flexible behavior to encompass the domains of synaptic plasticity and learning.

4.4. Limitations & future directions

Despite possible biological explanations, one major limitation of our work is our *condition* of assigning a distinct learning rule (unique set of coefficients) per synapse. This presents problems of scalability, overfitting, generalization and interpretability (e.g. increasing numbers of synapses requires re-meta-training). Although we presented the surprising results of very slight variability of coefficients and generalization of a specific meta-learned rule across models with the different initial weights, we have only demonstrated these phenomena on only one task (*wavereg*). Additionally, the simplicity of our tasks and models render our results inconclusive as to whether our method successfully scales to deeper or different networks and more complex problems. Investigating whether results 3.2, 3.3. and 3.4 can also be produced for the *bmnist* task would be a reasonable next first step towards demonstrating aspects of our method's robustness. In the future, we could also address whether our method works for an MNIST task with more difficult digits or perhaps three, four or more classes and perhaps describe new limitations before moving onto more challenging architectures and tasks (e.g. CIFAR-10).

With regard to the architectures we experimented with, one important discussion to have involves a type of connectivity that we have not yet implemented: feedback pathways. Although rule F_4 (see 2.2.4.) supports a modulatory third factor mechanism that globally broadcasts an error on all synapses during learning, it does not support vectorized error signaling capabilities. Consequently, our system has to rely on vectorized synaptic learning rules with numerous, unique coefficients parameters assigned to each synaptic connection in order to encode task-specific information and the intricate supervised learning dynamics of gradient calculation and descent (i.e. carried out by an identical teacher network) in order to successfully learn. Even though we found that the variability of coefficients across synapses is tiny and such a configuration may hold biological significance, the wide prevalence of feedback-informed networks in the cortical hierarchy presents a more promising alternative for efficient credit assignment (Markov et al., 2014; Harris & Shepherd, 2015) Therefore, our next goal is to introduce an architecture with feedback pathways through which target information can flow and inform the change of synapses at upstream layers. Then, by meta-learning Hebbian plasticity rules in this context, we can evaluate how well they perform compared to bio-inspired alternatives like Equilibrium Propagation (Scellier & Bengio, 2017).

Towards making evolutionary optimization more realistic, an interesting extension would also be to add an option for performing stochastic, instead of GD-based, coefficient parameter updates, using an algorithm such as the covariance matrix adaptation evolution strategy (Hansen, 2023). Whereas, for loosening the per-synapse condition, perhaps adding dendritic functionality to the point-neuron perceptron layers could also help remove some of the burden of useful heterogeneity from the rule coefficients by introducing an element of sparsity and stability to the network (Grewal et al., 2021; Wilmes & Clopath, 2023; Yaeger et al., 2024).

4.5. Conclusion

Using meta-learning we explored the idea of dynamically modifying model plasticity rules themselves in order to explore how GD performance can be reached. Inspired by evolution giving rise to the immense complexity of neurophysiological plasticity mechanisms responsible for efficient learning in the brain, this work contributes to the ongoing dialogue between neuroscience and AI, highlighting meta-learning as a fruitful tool for addressing problems inherent in the remarkable biochemical machinery that makes biological credit assignment possible. To quote Richard Sutton:

“*... the actual contents of minds are tremendously, irredeemably complex; we should stop trying to find simple ways to think about the contents of minds, such as simple ways to think about space, objects, multiple agents, or symmetries. All these are part of the arbitrary, intrinsically-complex, outside world. They are not what should be built in, as their complexity is endless; instead we should build in only the meta-methods that can find and capture this arbitrary complexity. Essential to these methods is that they can find good approximations, but the search for them should be by our methods, not by us. We want AI agents that can discover like we can, not which contain what we have discovered. Building in our discoveries only makes it harder to see how the discovering process can be done.*”

References

Akaishi, R., Kolling, N., Brown, J. W., & Rushworth, M. (2016). Neural Mechanisms of Credit Assignment in a Multicue Environment. The Journal of neuroscience : the official journal of the Society for Neuroscience, 36(4), 1096–1112. https://doi.org/10.1523/JNEUROSCI.3159-15.2016

Andrychowicz et al. (2016). Learning to learn by gradient descent by gradient descent. arXiv. https://doi.org/10.48550/arXiv.1606.04474

Bengio, S., Bengio, Y. & Cloutier, J. On the search for new learning rules for ANNs. Neural Process Lett 2, 26–30 (1995). https://doi.org/10.1007/BF02279935

Bisong, E. (2019). Optimization for Machine Learning: Gradient Descent. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_16

Brown, R. E., Bligh, T. W. B., & Garden, J. F. (2021). The Hebb Synapse Before Hebb: Theories of Synaptic Function in Learning and Memory Before , With a Discussion of the Long-Lost Synaptic Theory of William McDougall. Frontiers in behavioral neuroscience, 15, 732195. https://doi.org/10.3389/fnbeh.2021.732195

Choudhary, A., Radhakrishnan, A., Lindner, J.F. et al. Neuronal diversity can improve machine learning for physics and beyond. Sci Rep 13, 13962 (2023). https://doi.org/10.1038/s41598-023-40766-6

Confavreux et al. (2020). A meta-learning approach to (re)discover plasticity rules that carve a desired function into a neural network. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 1376, 16398–16408. https://proceedings.neurips.cc/paper/2020/hash/bdbd5ebfde4934142c8a88e7a3796cd5-Abstract.html

Confavreux et al. (2023). Meta-learning families of plasticity rules in recurrent spiking networks using simulation-based inference. Neural Information Processing Systems (NIPS'23). https://nips.cc/virtual/2023/poster/72268

Doty et al. (2021). Heterogeneous ‘cell types’ can improve performance of deep neural networks. bioRxiv. https://doi.org/10.1101/2021.06.21.449346

Doya K. (2002). Metalearning and neuromodulation. Neural networks : the official journal of the International Neural Network Society, 15(4-6), 495–506. https://doi.org/10.1016/s0893-6080(02)00044-8

Farashahi, S., Donahue, C. H., Khorsand, P., Seo, H., Lee, D., & Soltani, A. (2017). Metaplasticity as a Neural Substrate for Adaptive Learning and Choice under Uncertainty. Neuron, 94(2), 401–414.e6. https://doi.org/10.1016/j.neuron.2017.03.044

Finn, C., Abbeel, P.; Levine, S.. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. Proceedings of the 34th International Conference on Machine Learning, in Proceedings of Machine Learning Research, 70:1126-1135. https://proceedings.mlr.press/v70/finn17a.html

Francioni, V., Tang, V. D., Brown, N. J., Toloza, E. H. S., & Harnett, M. (2023). Vectorized instructive signals in cortical dendrites during a brain-computer interface task. bioRxiv : the preprint server for biology, 2023.11.03.565534. https://doi.org/10.1101/2023.11.03.565534

Gerstner, W., Kistler, W. M., Naud, R., & Paninski, L. (2014). Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition. Cambridge: Cambridge University Press. Chapter 19. https://neuronaldynamics.epfl.ch/online/Ch19.S2.html#Ch19

Gerstner, W., Kistler, W. Mathematical formulations of Hebbian learning. Biol Cybern 87, 404–415 (2002). https://doi.org/10.1007/s00422-002-0353-y

Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., & Brea, J. (2018). Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules. Frontiers in neural circuits, 12, 53. https://doi.org/10.3389/fncir.2018.00053

Goaillard, J. M., & Marder, E. (2021). Ion Channel Degeneracy, Variability, and Covariation in Neuron and Circuit Resilience. Annual review of neuroscience, 44, 335–357. https://doi.org/10.1146/annurev-neuro-092920-121538

Gordon, U., Polsky, A., & Schiller, J. (2006). Plasticity compartments in basal dendrites of neocortical pyramidal neurons. The Journal of neuroscience : the official journal of the Society for Neuroscience, 26(49), 12717–12726. https://doi.org/10.1523/JNEUROSCI.3502-06.2006

Greedy, W. et al. (2022). Single-phase deep learning in cortico-cortical networks. arXiv. https://doi.org/10.48550/arXiv.2206.11769

Grewal et al. (2021). Going Beyond the Point Neuron: Active Dendrites and Sparse Representations for Continual Learning. BioRxiv. https://doi.org/10.1101/2021.10.25.465651

Gu et al. (2019). Meta-Learning Biologically Plausible Semi-Supervised Update Rules. bioRxiv. https://doi.org/10.1101/2019.12.30.891184

Hansen (2016). The CMA Evolution Strategy: A Tutorial. ArXiv. https://doi.org/10.48550/arXiv.1604.00772

Harris, K., Shepherd, G. The neocortical circuit: themes and variations. Nat Neurosci 18, 170–181 (2015). https://doi.org/10.1038/nn.3917

Harvey, C., Svoboda, K. Locally dynamic synaptic learning rules in pyramidal neuron dendrites. Nature 450, 1195–1200 (2007). https://doi.org/10.1038/nature06416

He et al. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. arXiv. https://doi.org/10.48550/arXiv.1502.01852.

Hebb, D. O. (1949). The organization of behavior. New York: Wiley.

Hebbian theory article. Wikipedia: https://en.wikipedia.org/wiki/Hebbian_theory

Iigaya K. (2016). Adaptive learning and decision-making under uncertainty by metaplastic synapses guided by a surprise detection system. eLife, 5, e18073. https://doi.org/10.7554/eLife.18073

Javed & White (2019). Meta-Learning Representations for Continual Learning. ArXiv. https://doi.org/10.48550/arXiv.1905.12588

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences of the United States of America, 114(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114

Kruskal, W. H., & Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. Journal of the American Statistical Association, 47(260), 583–621. https://doi.org/10.2307/2280779

Kuśmierz, Ł., Isomura, T., & Toyoizumi, T. (2017). Learning with three factors: modulating Hebbian plasticity with errors. Current opinion in neurobiology, 46, 170–177. https://doi.org/10.1016/j.conb.2017.08.020

Lamba, A., Nassar, M. R., & FeldmanHall, O. (2023). Prefrontal cortex state representations shape human credit assignment. eLife, 12, e84888. https://doi.org/10.7554/eLife.84888

Lansdell, B. J., & Kording, K. P. (2019). Towards learning-to-learn. Current Opinion in Behavioral Sciences, 29, 45–50. https://doi.org/10.1016/j.cobeha.2019.04.005

LeCun, Y. (1985) Une procédure d'apprentissage pour réseau a seuil asymmetrique (a Learning Scheme for Asymmetric Threshold Networks), Proceedings of Cognitiva 85, 599-604, Paris, France. http://yann.lecun.com/exdb/publis/pdf/lecun-85.pdf

LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

Leibniz (Author) (2007). The Early Mathematical Manuscripts of Leibniz. Merchant Books, 2007. See p. 126: the chain rule appeared in a 1676 memoir by Leibniz.

Letzkus, J. J., Kampa, B. M., & Stuart, G. J. (2006). Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. The Journal of neuroscience : the official journal of the Society for Neuroscience, 26(41), 10420–10429. https://doi.org/10.1523/JNEUROSCI.2650-06.2006

Lillicrap, T. P. et al. (2016). Random synaptic feedback weights support error backpropagation for deep learning. Nat. Commun. 7, 13276 doi: https://doi.org/10.1038/ncomms13276

Lillicrap, T.P., Santoro, A., Marris, L. et al. (2020) Backpropagation and the brain. Nat Rev Neurosci 21, 335–346. https://doi.org/10.1038/s41583-020-0277-3

Lindsey & Litwin-Kumar (2019). Learning to Learn with Feedback and Local Plasticity. arXiv. https://doi.org/10.48550/arXiv.2006.09549

Linnainmaa, S. (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki. https://people.idsia.ch/~juergen/linnainmaa1970thesis.pdf

Linnainmaa, S. (1976). Taylor expansion of the accumulated rounding error. BIT 16, 146–160. https://doi.org/10.1007/BF01931367

Lisman, J., Grace, A. A., & Duzel, E. (2011). A neoHebbian framework for episodic memory; role of dopamine-dependent late LTP. Trends in neurosciences, 34(10), 536–547. https://doi.org/10.1016/j.tins.2011.07.006

Magee, J. C., & Grienberger, C. (2020). Synaptic Plasticity Forms and Functions. Annual review of neuroscience, 43, 95–117. https://doi.org/10.1146/annurev-neuro-090919-022842

Marder, E. (2011). Variability, compensation, and modulation in neurons and circuits. Proceedings of the National Academy of Sciences of the United States of America, 108 Suppl 3(Suppl 3), 15542–15548. https://doi.org/10.1073/pnas.1010674108

Marder, E., Goaillard, JM. Variability, compensation and homeostasis in neuron and network function. Nat Rev Neurosci 7, 563–574 (2006). https://doi.org/10.1038/nrn1949

Marder, E., Taylor, A. Multiple models to capture the variability in biological neurons and networks. Nat Neurosci 14, 133–138 (2011). https://doi.org/10.1038/nn.2735

Markov, N. T., Vezoli, J., Chameau, P., Falchier, A., Quilodran, R., Huissoud, C., Lamy, C., Misery, P., Giroud, P., Ullman, S., Barone, P., Dehay, C., Knoblauch, K., & Kennedy, H. (2014). Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. The Journal of comparative neurology, 522(1), 225–259. https://doi.org/10.1002/cne.23458

McClelland, J. L. (2006). 'How far can you go with Hebbian learning, and when does it lead you astray?', in Yuko Munakata, and Mark H Johnson (eds), Processes of Change in Brain and Cognitive Development: Attention and Performance XXI. https://doi.org/10.1093/oso/9780198568742.003.0002

Metz et al. (2019). Meta-Learning Update Rules for Unsupervised Representation Learning. arXiv. https://doi.org/10.48550/arXiv.1804.00222

Miconi et al. (2020). Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. arXiv. https://doi.org/10.48550/arXiv.2002.10585

Miconi, T., Stanley, K., Clune, J.. (2018). Differentiable plasticity: training plastic neural networks with backpropagation. Proceedings of the 35th International Conference on Machine Learning, in Proceedings of Machine Learning Research 80:3559-3568 Available from https://proceedings.mlr.press/v80/miconi18a.html

Najarro & Risi (2020). Meta-Learning through Hebbian Plasticity in Random Networks. arXiv. https://doi.org/10.48550/arXiv.2007.02686

Parker, D.B. (1985) Learning-Logic: Casting the Cortex of the Human Brain in Silicon. Technical Report Tr-47, Center for Computational Research in Economics and Management Science. MIT Cambridge, MA.

Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A., & Naud, R. (2021). Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. Nature neuroscience, 24(7), 1010–1019. https://doi.org/10.1038/s41593-021-00857-x

Pedersen & Risi (2023). Learning to Act through Evolution of Neural Diversity in Random Neural Networks. arXiv. https://doi.org/10.1145/3583131.3590460

Perez-Nieves, N., Leung, V.C.H., Dragotti, P.L. et al. Neural heterogeneity promotes robust learning. Nat Commun 12, 5791 (2021). https://doi.org/10.1038/s41467-021-26022-3

Reifenstein, E. T., Bin Khalid, I., & Kempter, R. (2021). Synaptic learning rules for sequence learning. eLife, 10, e67171. https://doi.org/10.7554/eLife.67171

Rich Sutton (2019). The Bitter Lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Richards, B. A., & Kording, K. P. (2023). The study of plasticity has always been about gradients. The Journal of physiology, 601(15), 3141–3149. https://doi.org/10.1113/JP282747

Richards, B. A., & Lillicrap, T. P. (2018). Dendritic solutions to the credit assignment problem. Current opinion in neurobiology, 54, 28–36. https://doi.org/10.1016/j.conb.2018.08.003

Roelfsema, P., Holtmaat, A. Control of synaptic plasticity in deep cortical networks. Nat Rev Neurosci 19, 166–180 (2018). https://doi.org/10.1038/nrn.2018.6

Ruder, S. (2017). An overview of gradient descent optimization algorithms. arXiv. https://doi.org/10.48550/arXiv.1609.04747

Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 533–536 (1986). https://doi.org/10.1038/323533a0

Sacramento, J. et al. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. arXiv. https://doi.org/10.48550/arXiv.1810.11393

Sandler et al. (2021). Meta-Learning Bidirectional Update Rules. arXiv. https://doi.org/10.48550/arXiv.2104.04657

Scellier, B., & Bengio, Y. (2017). Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. Frontiers in computational neuroscience, 11, 24. https://doi.org/10.3389/fncom.2017.00024

Schmidhuber, J. (1987). Evolutionary principles in self-referential learning. (On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich. https://people.idsia.ch/~juergen/diploma1987ocr.pdf

Schmidhuber, J. (1992). Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. Neural Comput 1992; 4 (1): 131–139. https://doi.org/10.1162/neco.1992.4.1.131

Schmidhuber, J. (1992). Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. Neural Comput; 4 (1): 131–139. https://doi.org/10.1162/neco.1992.4.1.131

Schultz W. (2016). Dopamine reward prediction error coding. Dialogues in clinical neuroscience, 18(1), 23–32. https://doi.org/10.31887/DCNS.2016.18.1/wschultz

Shervani-Tabar, N., Rosenbaum, R. (2023). Meta-learning biologically plausible plasticity rules with random feedback pathways. Nat Commun 14, 1805. https://doi.org/10.1038/s41467-023-37562-1

Speelpenning, B. (1980). Compiling fast partial derivatives of functions given by algorithms. United States. https://doi.org/10.2172/5254402

Stolyarova A. (2018). Solving the Credit Assignment Problem With the Prefrontal Cortex. Frontiers in neuroscience, 12, 182. https://doi.org/10.3389/fnins.2018.00182

T. Hospedales, A. Antoniou, P. Micaelli and A. Storkey (2022). "Meta-Learning in Neural Networks: A Survey" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 44, no. 09, pp. 5149-5169. https://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3079209

Thrun, S., Pratt, L. (1998). Learning to Learn: Introduction and Overview. In: Thrun, S., Pratt, L. (eds) Learning to Learn. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-5529-2_1

Tom Schaul & Juergen Schmidhuber (2010), Scholarpedia, 5(6):4650. http://dx.doi.org/10.4249/scholarpedia.4650

Wang, J. X. et al. (2016). Learning to reinforcement learn. arXiv. https://doi.org/10.48550/arXiv.1611.05763

Werbos, P. J. (1982) Applications of advances in nonlinear sensitivity analysis. In R. Drenick, F. Kozin, (eds): System Modeling and Optimization: Proc. IFIP, Springer. http://werbos.com/Neural/SensitivityIFIPSeptember1981.pdf

Werbos, P.J. (1974) Beyond Recognition, New Tools for Prediction and Analysis in the Behavioural Sciences. Ph.D. Thesis, Harvard University, Cambridge.

Whittington, J. C. R., & Bogacz, R. (2019). Theories of Error Back-Propagation in the Brain. Trends in cognitive sciences, 23(3), 235–250. https://doi.org/10.1016/j.tics.2018.12.005

Wilmes, K.A., Clopath, C. (2023). Dendrites help mitigate the plasticity-stability dilemma. Sci Rep 13, 6543. https://doi.org/10.1038/s41598-023-32410-0

Yaeger et al. (2022). Dendritic compartmentalization of input-specific integration and plasticity rules across cortical development. bioRxiv. https://doi.org/10.1101/2022.02.02.478840

Yu, A. J., & Dayan, P. (2005). Uncertainty, neuromodulation, and attention. Neuron, 46(4), 681–692. https://doi.org/10.1016/j.neuron.2005.04.026

Zenke, F., & Gerstner, W. (2017). Hebbian plasticity requires compensatory processes on multiple timescales. Philosophical transactions of the Royal Society of London. Series B, Biological sciences, 372(1715), 20160259. https://doi.org/10.1098/rstb.2016.0259

Zenke, F., Gerstner, W., & Ganguli, S. (2017). The temporal paradox of Hebbian learning and homeostatic plasticity. Current opinion in neurobiology, 43, 166–176. https://doi.org/10.1016/j.conb.2017.03.015

Appendix A – Implementation details

Tables 1, 2 list the parameter values chosen for producing the results (see chapter 3.). Table 1 contains the learning and meta-learning optimizers, learning rates and number of epochs chosen for meta-training and training with each Hebbian learning rule. Table 2 contains the initial coefficient parameter values for each Hebbian learning rule set prior to meta-learning. Note that in Table 2, $*$ corresponds to the *bmnist* task.

For backprop training, we used the SGD and Adam optimizers. For Hebbian training, we used Hebbian learning rules F_1, F_2, F_3, F_4 (see 2.2.). During meta-optimization, the rule coefficients were auto-tuned using the Adam and SGD optimizers. For rule F_4, in particular, we chose the modulatory third factor M to be the loss of the network. For the *wavereg* experiments we used the mean squared error (MSE) as a loss function and for the *bmnist* task the cross entropy loss was chosen.

Our experiments were implemented using pytorch 2.1.0+cu121 and ran on a 6-core Intel(R) Core(TM) i7-8700K @ 3.70GHz CPU and on a NVIDIA GeForce GTX 1080 Ti GPU.

Table 1: Training and meta-training optimizers (optimizer type, lr, epochs)

	wavereg		*bmnist*	
RULE	TRAINING	META-TRAINING	TRAINING	META-TRAINING
F_1	*Adam*, 4e-3, 200	*Adam*, 5e-6, 1500	-	-
F_2	*Adam*, 4e-3, 200	*Adam*, 5e-6, 1500	-	-
F_3	-	-	*Adam*, 4e-3, 35	*Adam*, 3e-9, 100
F_4	*SGD*, 4e-3, 600	*SGD*, 5e-2, 1500	*SGD*, 4e-3, 35	*SGD*, 5e-1, 100

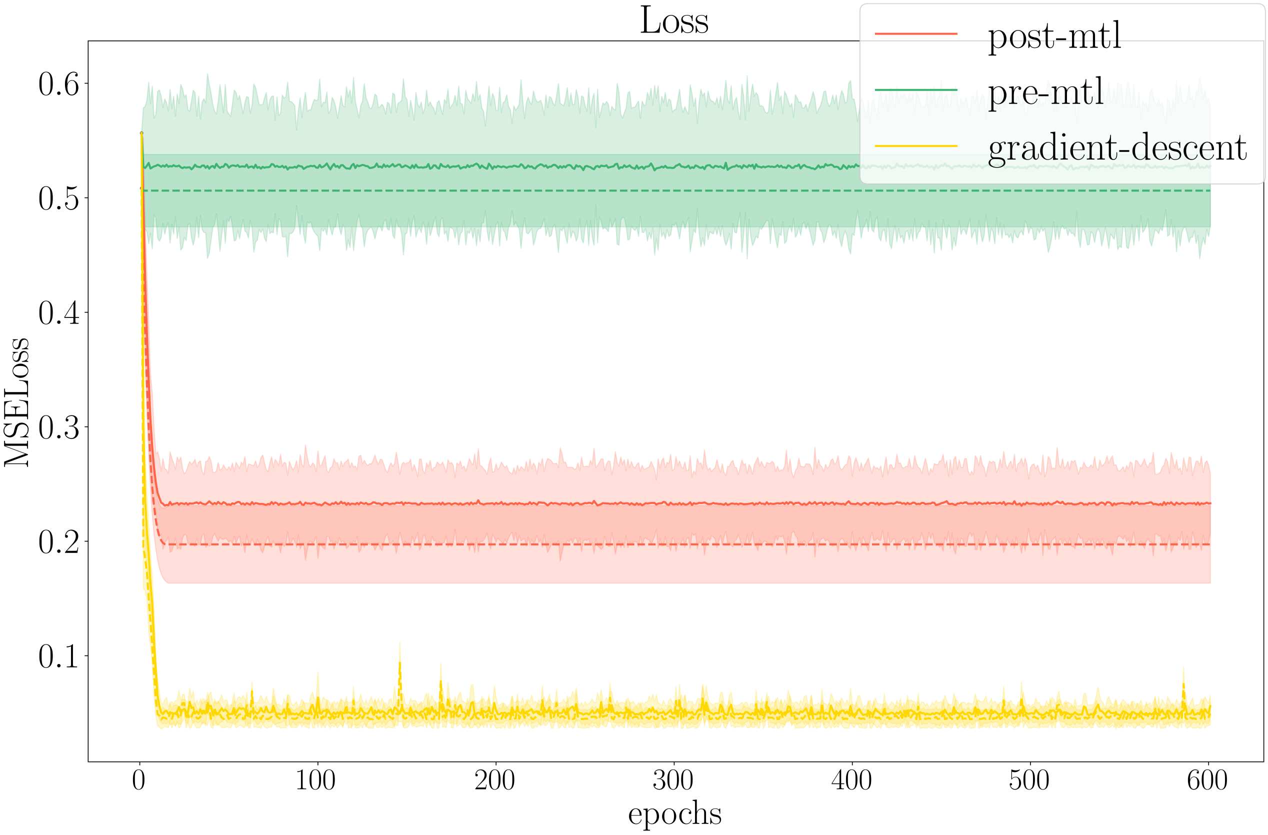
Table 2: Learning rule coefficients

RULE TERM	COEFF. NAME	F_1	F_2	F_3*	F_4
$(1 - w_{ij})v_iv_j$	γ_2	1e-8	1e-8	-	-
w_{ij}	γ_0	1e-3	1e-3	-	-
1	b	-	0	-	-
1	c_0	-	-	1e-8$*$	1e-8
v_j	c_1	-	-	1e-8$*$	1e-7/1e-8$*$
v_i	c_2	-	-	1e-8$*$	1e-7/1e-8$*$
v_j^2	c_3	-	-	1e-8$*$	1e-8
v_i^2	c_4	-	-	1e-8$*$	1e-8
v_iv_j	c_5	-	-	1e-8$*$	1e-8

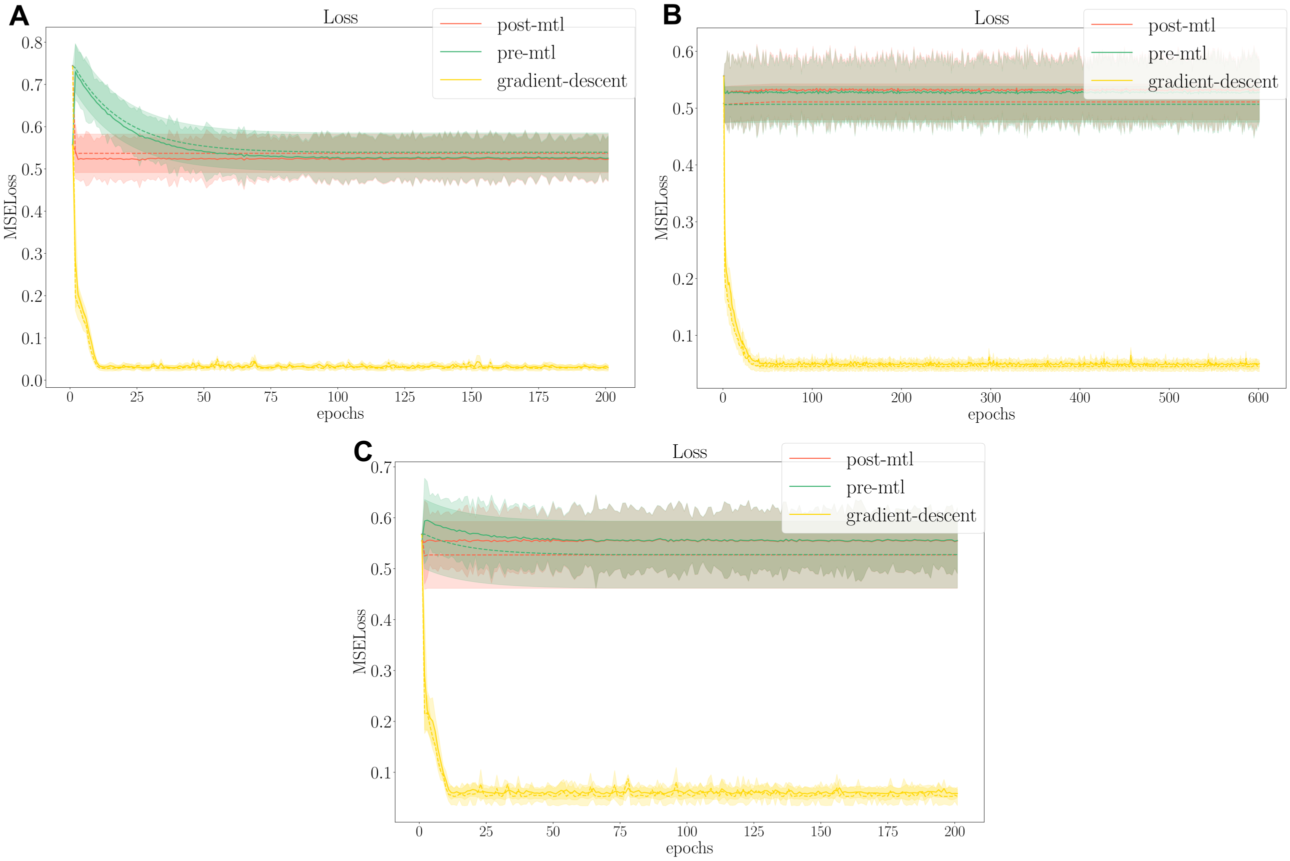
Appendix B – Abbreviations

Abbreviation	Full form	Description
Adam	Adaptive Moment Estimation	a GD-based stochastic optimization method based on adaptive learning rates
ANN	Artificial neural network	a machine learning model made up of artificial neurons, which loosely model the neurons in a biological brain
Backprop	Backpropagation (of error)	a gradient estimation method used to train neural network models
GD	Gradient descent	an iterative algorithm for finding a local minimum of a differentiable multivariate function
LTP	Long-term potentiation	an activity-dependent *increase* in the efficacy of neuronal synapses lasting hours or longer following a long patterned stimulus
LTD	Long-term depression	an activity-dependent *reduction* in the efficacy of neuronal synapses lasting hours or longer following a long patterned stimulus
MLP	Multilayer perceptron	a modern feedforward artificial neural network, consisting of fully connected neurons with a non-linear kind of activation function, organized in at least three layers
NN	Neural network	an interconnected population of either biological or artificial neurons
SGD	Stochastic gradient descent	a stochastic approximation of gradient descent optimization that replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data)

Appendix C – Supplementary figures



Sup. Fig. 1. Meta-learned rule F_3 on *wavereg* task, training (solid) and validation (dotted) loss.



Sup. Fig. 2. Training (solid) and validation (dotted) loss of meta-learned rules F_1 (**A**), F_2 (**B**), F_4 (**C**) on *wavereg* task i*n "layer mode*". In this mode, each layer of the network is trained using a rule whose coefficients values were set to the mean value of coefficients across synapses of the corresponding "*synapse mode*" rules in Fig. 3, 4, 5.