META-LEARNING WITH NEGATIVE LEARNING RATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning models require a large amount of data to perform well. When data is scarce for a target task, we can transfer the knowledge gained by training on similar tasks to quickly learn the target. A successful approach is *meta-learning*, or *learning to learn* a distribution of tasks, where *learning* is represented by an outer loop, and *to learn* by an inner loop of gradient descent. However, a number of recent empirical studies argue that the inner loop is unnecessary and more simple models work equally well or even better. We study the performance of MAML as a function of the learning rate of the inner loop, where zero learning rate implies that there is no inner loop. Using random matrix theory and exact solutions of linear models, we calculate an algebraic expression for the test loss of MAML applied to mixed linear regression and nonlinear regression with overparameterized models. Surprisingly, while the optimal learning rate for adaptation is positive, we find that the optimal learning rate for training is always negative, a setting that has never been considered before. Therefore, not only the performance increases by decreasing the learning rate to zero, as suggested by recent work, but it can be increased even further by decreasing the learning rate to negative values. These results help clarifying under what circumstances meta-learning performs best.

1 INTRODUCTION

Deep Learning models represent the state-of-the-art in several machine learning benchmarks (LeCun et al. (2015)), and their performance doesn't seem to stop improving when adding more data and computing resources (Rosenfeld et al. (2020), Kaplan et al. (2020)). However, they require a large amount of data and compute to start with, which are often not available to practitioners. The approach of *fine-tuning* has proved very effective to address this limitation: pre-train a model on a source task, for which a large dataset is available, and use this model as the starting point for a quick additional training (fine-tuning) on the small dataset of the target task (Pan & Yang (2010), Donahue et al. (2014), Yosinski et al. (2014)). This approach is popular because pre-trained models are often made available by institutions that have the resources to train them.

In some circumstances, multiple source tasks are available, all of which have scarce data, as opposed to a single source task with abundant data. This case is addressed by *meta-learning*, in which a model gains experience over multiple source tasks and uses it to improve its learning of future target tasks. The idea of meta-learning is inspired by the ability of humans to generalize across tasks, without having to train on any single task for long time. A meta-learning problem is solved by a bi-level optimization procedure: an outer loop optimizes meta-parameters across tasks, while an inner loop optimizes parameters within each task (Hospedales et al. (2020)).

The idea of meta-learning has gained some popularity, but a few recent papers argue that a simple alternative to meta-learning is just good enough, in which the inner loop is removed entirely (Chen et al. (2020a), Tian et al. (2020), Dhillon et al. (2020), Chen et al. (2020b), Raghu et al. (2020)). This is surprising, because this approach merges all source tasks into a single big training set, and it does not even distinguish between different source tasks during training. Other studies find the opposite (Goldblum et al. (2020)). It is hard to resolve the debate because there is very little theory available to explain these findings.

In this work, using random matrix theory and exact solutions of linear models, we derive an algebraic expression of the average test loss of MAML, a simple and successful meta-learning algorithm (Finn et al. (2017)), as a function of its hyperparameters. In particular, we study its performance as a

function of the inner loop learning rate during meta-training. Setting this learning rate to zero is equivalent to removing the inner loop, as advocated by recent work (Chen et al. (2020a), Tian et al. (2020), Dhillon et al. (2020), Chen et al. (2020b), Raghu et al. (2020)). Surprisingly, we find that the optimal learning rate is negative, thus performance can be increased by reducing the learning rate below zero. In particular, we find the following:

- In the problem of mixed linear regression, we prove that the optimal learning rate is always negative in overparameterized models. The same result holds in underparameterized models for small values of the learning rate. We validate the theory by running extensive experiments.
- We extend these results to the case of nonlinear regression and wide neural networks, in which the output of can be approximated by a linear function of the parameters (Jacot et al. (2018), Lee et al. (2019)). While in this case we cannot prove that the optimal learning rate is always negative, preliminary experiments suggest that the result holds in this case as well.

2 Related work

The field of meta-learning includes a broad range of problems and solutions, see Hospedales et al. (2020) for a recent review focusing on neural networks and deep learning. In this context, meta-learning received increased attention in the past few years, several new benchmarks have been introduced, and a large number of algorithms and models have been proposed to solve them (Vinyals et al. (2017), Bertinetto et al. (2019), Triantafillou et al. (2020)). Despite the surge in empirical work, theoretical work is still lagging behind.

Similar to our work, a few other studies used random matrix theory and exact solutions to calculate the average test loss for the problem of linear regression (Advani & Saxe (2017), Hastie et al. (2019), Nakkiran (2019)). To our knowledge, our study is the first to apply this technique to the problem of meta-learning with multiple tasks. Our results reduce to those of linear regression in the case of one single task. Furthermore, to our knowledge we are the first to apply the framework of Neural Tangent Kernel to the problem of meta-learning (Jacot et al. (2018), Lee et al. (2019)).

Similar to us, a few theoretical studies looked at the problem of mixed linear regression in the context of meta-learning. In Denevi et al. (2018), a meta-parameter is used to bias the task-specific parameters through a regularization term. Kong et al. (2020) looks at whether many tasks with small data can compensate for a lack of tasks with big data. However, none of these studies look into the effect of learning rate on performance, which is our main focus.

In this work, we focus on MAML, a simple and successful meta-learning algorithm (Finn & Levine (2018)). A few theoretical studies have investigated MAML, looking at: universality of the optimization algorithm (Finn & Levine (2018)), bayesian inference interpretation (Grant et al. (2018)), proof of convergence (Ji et al. (2020)), difference between convex and non-convex losses (Saunshi et al. (2020)), global optimality (Wang et al. (2020)). Again, none of these studies look at the effect of the learning rate, the main subject of our work.

3 Meta-learning and MAML

In this work, we follow the notation of Hospedales et al. (2020), and we use MAML (Finn et al. (2017)) as the meta-learning model in all our experiments and derivations. We assume the existence of a distribution of tasks τ and, for each task, a distribution of data points $\mathcal{D}^\tau$ and a loss function $\mathcal{L}^\tau$. The loss function of the meta-learning problem, $\mathcal{L}^{meta}$, is defined as an average over both distributions of tasks and data points. The goal of meta-learning is to minimize the loss function with respect to a vector of meta-parameters $\boldsymbol{\omega}$

$$\mathcal{L}^{meta}\left(\boldsymbol{\omega}\right) = \mathop{\mathbb{E}}_{\tau} \mathop{\mathbb{E}}_{\mathcal{D}_t^\tau} \mathop{\mathbb{E}}_{\mathcal{D}_v^\tau} \mathcal{L}^\tau\left(\boldsymbol{\theta}^\tau(\boldsymbol{\omega}, \mathcal{D}_t^\tau); \mathcal{D}_v^\tau\right) \tag{1}$$

Note that each task is characterized by a different distribution of the data, and we separate the training $\mathcal{D}_t$ and validation data $\mathcal{D}_v$. The vector of parameters $\boldsymbol{\theta}$ is task-specific, it depends on the meta-parameters $\boldsymbol{\omega}$, and the training data $\mathcal{D}_t$. Different meta-learning algorithms correspond to a

different choice of $\boldsymbol{\theta}^\tau$. We describe below the choice of MAML, the main subject of this study. Optimization of 1 is referred to as the *outer loop*, while computation of $\boldsymbol{\theta}^\tau$ is referred to as the *inner loop* of meta-learning.

In this work, we assume that the loss function is the same for all tasks, $\mathcal{L}^\tau = \mathcal{L}$. During the meta-training outer loop, the loss is evaluated on a sample of m tasks, and a sample of n_v validation data points for each task

$$\mathcal{L}^{meta}(\boldsymbol{\omega}) = \frac{1}{mn_v} \sum_{i=1}^{m} \sum_{j=1}^{n_v} \mathcal{L}\left(\boldsymbol{\theta}^{(i)}(\boldsymbol{\omega}); \mathcal{D}_j^{(i)}\right) \tag{2}$$

In the inner loop, the parameters $\boldsymbol{\theta}$ are learned by a set of n_t training data points for each task, usually distinct from the validation data. During meta-testing, a new (target) task is given and the parameters $\boldsymbol{\theta}$ are learned by a set of n_r target data points. The final performance of the model is computed on test data of the target task.

MAML aims at finding a good initial value $\boldsymbol{\omega}$ for adaptation to the target task, where adaptation corresponds to a few steps of gradient descent. During meta-training, MAML updates parameters by a few steps of gradient descent to each single training task, with a given learning rate α_t. In the simple case of a single gradient step, this is equal to

$$\boldsymbol{\theta}^{(i)}(\boldsymbol{\omega}) = \boldsymbol{\omega} - \frac{\alpha_t}{n_t} \sum_{j=1}^{n_t} \frac{\partial \mathcal{L}^{(i)}}{\partial \theta}\bigg|_{\boldsymbol{\omega}; \mathcal{D}_j^{(i)}} \tag{3}$$

If the learning rate α_t is zero, then parameters are not adapted during meta-training and $\boldsymbol{\theta}(\boldsymbol{\omega}) = \boldsymbol{\omega}$. In that case, a single set of parameters in learned across all data without distinguishing between different tasks, and there is no inner loop. A setting similar to this has been advocated in a few recent studies (Chen et al. (2020a), Tian et al. (2020), Dhillon et al. (2020), Chen et al. (2020b), Raghu et al. (2020)). It is important to note that, at testing time, parameters are adapted to the target task, with a few steps of gradient descent, using a distinct learning rate α_r.

We show that, intuitively, the optimal learning rate at testing time α_r is always positive. Surprisingly, in the family of problems considered in this study, we find that the optimal learning rate during training α_t is instead negative. We note that the setting $\alpha_t = 0$ effectively does not use the n_t training data points, therefore we could in principle add this data to the validation set, but we do not consider this option here since we are more interested in a wide range of possible values of α_t as opposed to the specific case $\alpha_t = 0$.

4 MIXED LINEAR REGRESSION

In this section we study MAML applied to the problem of mixed linear regression. Note that the goal here is not to solve the problem of mixed linear regression, but to probe the performance of MAML as a function of its hyperparameters.

In mixed linear regression, each task is characterized by a different linear function, and a model is evaluated by the mean squared error loss function. We assume a generative model in the form of $y = \mathbf{x}^T\mathbf{w} + z$, where $\mathbf{x}$ is the input vector (of dimension p), y is the output (scalar), z is noise (scalar), and $\mathbf{w}$ is a vector of generating parameters (of dimension p), therefore p represents both the number of parameters and the input dimension. Different tasks correspond to different values of generating parameters $\mathbf{w}$. Therefore, the distribution of tasks is on the generative parameters $\mathbf{w}$, while the distribution of data is on the input and output $(\mathbf{x}, y)$, all assumed Gaussian:

$$\mathbf{w} \sim \mathcal{N}\left(\mathbf{w}_0, \frac{\nu^2}{p} I_p\right) \qquad \mathbf{x} \sim \mathcal{N}(0, I_p) \qquad y|\mathbf{x}, \mathbf{w} \sim \mathcal{N}\left(\mathbf{x}^T\mathbf{w}, \sigma^2\right) \tag{4}$$

Using random matrix theory and exact solutions of linear models, we calculate the test loss as a function of the following hyperparameters: the number of training tasks m, number of data points per task for training (n_t), validation (n_v) and target (n_r), learning rate for training α_t and for adaptation to target α_r. Furthermore, we have the hyperparameters specific to the mixed linear regression problem: p, ν, σ. Since we use exact solutions to the linear problem, our approach is equivalent to running the outer loop optimization until convergence (see Appendix for details). We derive results in two limiting cases: overparameterized $p \gg n_v m$ and underparameterized $p \ll n_v m$. We distinguish these two cases in the following sections.

4.1 OVERPARAMETERIZED CASE

In the overparameterized case, the number of parameters p is much larger than the total number of validation data across tasks $n_v m$. In this case, since the data does not fully constrain the parameters, the optimal value of $\boldsymbol{\omega}$ found during meta-training depends on the initial condition used for optimization. This initial condition is denoted by $\boldsymbol{\omega}_0$. The average test loss is calculated in the Appendix, and the result is

$$\overline{\mathcal{L}}^{test} \simeq \frac{\sigma^2}{2}\left(1+\frac{\alpha_r^2 p}{n_r}\right) + h^r \left[\frac{\nu^2}{2}\left(1+\frac{n_v m}{p}\right) + \left(1-\frac{n_v m}{p}\right)|\boldsymbol{\omega}_0 - \mathbf{w}_0|^2 + \frac{\sigma^2 n_v m}{2p}\frac{1+\frac{\alpha_t^2 p}{n_t}}{h^t}\right] \tag{5}$$

where we have defined the following expressions for convenience of notation

$$h^t = (1-\alpha_t)^2 + \alpha_t^2 \frac{p+1}{n_t} \tag{6}$$

$$h^r = (1-\alpha_r)^2 + \alpha_r^2 \frac{p+1}{n_r} \tag{7}$$

The expression 5 is valid for $p \gg n_v m$, and we do not attempt to derive error bounds. The loss always increases with the output noise σ and task variability ν. One term in 5 is proportional to the distance between the initial condition for the optimization of $\boldsymbol{\omega}_0$ and the ground truth mean of the generating model $\mathbf{w}_0$. Adding more validation data n_v and tasks m may increase or decrease the loss depending on the size of this term relative to the noise (Nakkiran (2019)), as it does reducing the number of parameters p. However, the loss always decreases with the number of data points for the target task n_r, as that data only affects the adaptation step.

Our main focus is studying how the loss is affected by the learning rates, during training α_t and adaptation α_r. The loss is a quadratic and convex function of α_r, therefore it has a unique minimum. While it is possible to compute the optimal value of α_r from 5, here we just note that the loss is a sum of two quadratic functions, one has a minimum at $\alpha_r = 0$ and another has a minimum at $\alpha_r = 1/\left(1+(p+1)/n_r\right)$, therefore the optimal learning rate is in between the two values and is always positive. This is intuitive, since a positive learning rate for adaptation implies that the parameters get closer to the optimum for the target task. An example of the loss as a function of the adaptation learning rate α_r is shown in Figure 1a, where we also show the results of experiments where we run MAML empirically. The good agreement between theory and experiment suggest that the approximation 5 of the loss is accurate.

However, the training learning rate α_t shows the opposite: by taking the derivative of 5 with respect to α_t, it is possible to show that it has a unique absolute minimum for a negative value of α_t. This can be proved by noting that this function has the same finite value for large positive or negative α_t, its derivative is always positive at $\alpha_t = 0$, and it has one minimum $(-)$ and one maximum $(+)$ at values

$$\alpha_t^{\pm} = -\frac{n_t+1}{2p} \pm \sqrt{\left(\frac{n_t+1}{2p}\right)^2 + \frac{n_t}{p}} \tag{8}$$

Note that the argmax α_t^+ is always positive, while the argmin α_t^- is always negative. This result is counter-intuitive, since a negative learning rate pushes parameters towards higher values of the loss. However, learning of the meta-parameter $\boldsymbol{\omega}$ is performed by the outer loop, for which there is no learning rate since we are using the exact solution to the linear problem and thus we are effectively training to convergence. Therefore, it remains unclear whether the inner loop should push parameters towards higher or lower values of the loss. An example of the loss as a function of the training learning rate α_r is shown in Figure 1b, where we also show the results of experiments where we run MAML empirically. Here the theory slightly underestimate the experimental loss, but the overall shapes of the curves are in good agreement, suggesting that the approximation 5 of the loss is accurate.

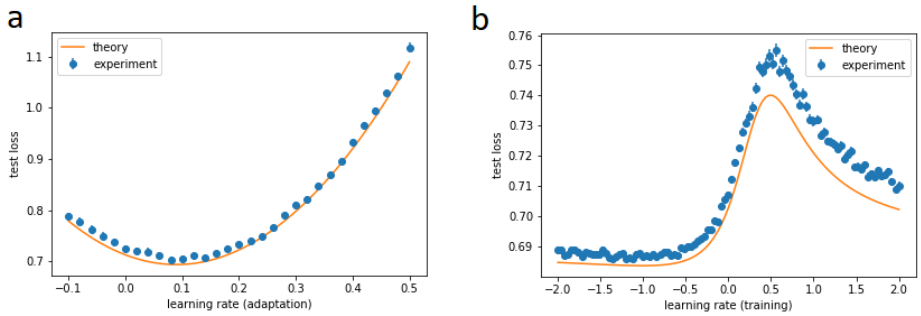

Figure 1: Average test loss of MAML as a function of the learning rate, on overparameterized mixed linear regression, as predicted by our theory and confirmed in experiments. a) Effect of learning rate α_r during adaptation. b) Effect of learning rate α_t during training. The optimal learning rate during adaptation is always positive, while that during training is always negative. Values of parameters: $n_t = 30, n_v = 2, n_r = 20, m = 3, p = 60, \sigma = 1., \nu = 0.5$, $\boldsymbol{\omega}_0 = \mathbf{0}$, $\mathbf{w}_0 = \mathbf{0}$. In panel a) we set $\alpha_t = 0.2$, in panel b) we set $\alpha_r = 0.2$. In the experiments, each run is evaluated on 100 tasks of 50 data points each, and each point is an average over 100 runs (a) or 1000 runs (b).

4.2 UNDERPARAMETERIZED CASE

In the underparameterized case, the number of parameters p is much smaller than the total number of validation data across tasks $n_v m$. In this case, since the data fully constrains the parameters, the optimal value of $\boldsymbol{\omega}$ found during meta-training is unique. The average test loss is calculated in the Appendix, and the result is

$$\overline{\mathcal{L}}^{test} \simeq \frac{\sigma^2}{2}\left(1 + \frac{\alpha_r^2 p}{n_r}\right) + \tag{9}$$

$$+ \frac{h^r \nu^2}{2} + \frac{h^r}{2h^{t2}} \frac{p}{n_v m}\left\{\sigma^2\left[h^t + \frac{\alpha_t^2}{n_t}\left[(n_v+1)\, g_1 + p g_2\right]\right] + \frac{\nu^2}{p}\left[(n_v+1)\, g_3 + p g_4\right]\right\} \tag{10}$$

where h^r, h^t are defined as in previous section, and g_i are polynomials in α_t and depend on p, n_t as well (see Appendix). This expression is valid for $p \ll n_v m$, and we do not attempt to derive error bounds. Again, the loss always increases with the output noise σ and task variability ν. Furthermore, in this case the loss always decreases with the number of data points n_v, n_r, and tasks m. Note that, for a very large number of tasks m, the loss does not depend on meta-training hyperparameters α_t, n_v, n_t. When the number of tasks is infinite, it doesn't matter whether we do the inner loop, and how much data we have for each task.

As in the overparameterized case, the loss is a quadratic and convex function of the adaptation learning rate α_r, and there is a unique minimum. While the value of the argmin is different, in this case as well the loss is a sum of two quadratic functions, one with minimum at $\alpha_r = 0$ and another with a minimum at $\alpha_r = 1/\left(1 + (p+1)/n_r\right)$, therefore the optimal learning rate is again in between the same two values and is always positive. Similar comments applies to this case: a positive learning rate for adaptation implies that the parameters get closer to the optimum for the target task. An example of the loss as a function of the adaptation learning rate α_r is shown in Figure 2a, where we also show the results of experiments where we run MAML empirically. The good agreement between theory and experiment suggest that the approximation 9 of the loss is accurate.

As a function of the training learning rate α_t, the loss 9 is the ratio of two fourth order polynomials, therefore it is not straightforward to determine its behaviour. However, it is possible to show that the following always holds

$$\left.\frac{\partial \overline{\mathcal{L}}^{test}}{\partial \alpha_t}\right|_{\alpha_t = 0} = \frac{\sigma^2 p}{n_v m} \geq 0 \tag{11}$$

suggesting that performance is always better for negative values of α_t around zero. Even if counter-intuitive, this finding aligns with that of previous section, and similar comments apply. An example of the loss as a function of the training learning rate α_r is shown in Figure 2b, where we also show the results of experiments where we run MAML empirically. A good agreement is observed between theory and experiment, again suggesting that the approximation 9 of the loss is accurate.

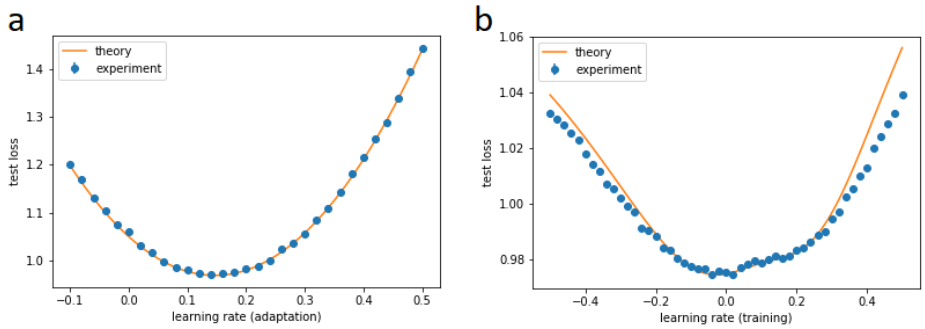

Figure 2: Average test loss as a function of the learning rate, on underparameterized mixed linear regression, as predicted by our theory and confirmed in experiments. a) Effect of learning rate α_r during testing. b) Effect of learning rate α_t during training. The optimal learning rate during testing is always positive, while that during training is negative. Values of parameters: $n_t = 5, n_v = 25, n_r = 10, m = 40, p = 30, \sigma = 0.2, \nu = 0.2$. In panel a) we set $\alpha_t = 0.2$, in panel b) we set $\alpha_r = 0.2$. In the experiments, the model is evaluated on 100 tasks of 50 data points each, and each point is an average over 100 (a) or 1000 (b) runs.

5 Non-Gaussian theory in overparameterized models

In the previous section we studied the performance of MAML applied to the problem of mixed linear regression. It remains unclear whether the results in the linear case are relevant for the more interesting case of nonlinear problems. Inspired by recent theoretical work, we consider the case of nonlinear regression with squared loss

$$\mathcal{L}(\boldsymbol{\omega}) = \mathop{\mathbb{E}}_{\mathbf{x}} \mathop{\mathbb{E}}_{y|\mathbf{x}} \left[y - f(\mathbf{x}, \boldsymbol{\omega})\right]^2 \tag{12}$$

where y is a target output and $f(\mathbf{x}, \boldsymbol{\omega})$ the output of a neural network with input $\mathbf{x}$ and parameters $\boldsymbol{\omega}$. The introduction of the Neural Tangent Kernel showed that, in the limit of infinitely wide neural networks, the output is a linear function of its parameters during the entire course of training (Jacot et al. (2018), Lee et al. (2019)). This is expressed by a first order Taylor expansion

$$f(\mathbf{x}, \boldsymbol{\omega}) \simeq f(\mathbf{x}, \boldsymbol{\omega}_0) + \mathbf{k}(\mathbf{x}, \boldsymbol{\omega}_0)^T (\boldsymbol{\omega} - \boldsymbol{\omega}_0) \tag{13}$$

$$\mathbf{k}(\mathbf{x}, \boldsymbol{\omega}_0) = \nabla_{\boldsymbol{\omega}} f(\mathbf{x}, \boldsymbol{\omega})|_{\mathbf{x}, \boldsymbol{\omega}_0} \tag{14}$$

The parameters $\boldsymbol{\omega}$ remain close to the initial condition $\boldsymbol{\omega}_0$ during the entire course of training, a phenomenon referred to as *lazy training* (Chizat et al. (2020)), and therefore the output can be linearized around $\boldsymbol{\omega}_0$. Intuitively, in a model that is heavily overparameterized, the data does not constrain the parameters, and a parameter that locally minimizes the loss 12 can be found in the vicinity of any initial condition $\boldsymbol{\omega}_0$. Note that, while the output of the neural network is linear in the parameters, it remains a nonlinear function of its input, through the vector of nonlinear functions $\mathbf{k}$ in 14.

By substituting 13 into 12, the nonlinear regression becomes effectively linear, in the sense that the loss is a quadratic function of the parameters $\boldsymbol{\omega}$, and all nonlinearities are contained in the functions $\mathbf{k}$ in 14, that are fixed by the initial condition $\boldsymbol{\omega}_0$. This suggests that we can carry over the theory developed in the previous section to this problem. However, in this case the input to the linear regression problem is effectively $\mathbf{k}(\mathbf{x})$, and some of the assumptions made in the previous section

are not acceptable. In particular, even if we assume that $\mathbf{x}$ is Gaussian, $\mathbf{k}(\mathbf{x})$ is a nonlinear function of $\mathbf{x}$ and cannot be assumed Gaussian.

In the Appendix, we extend the theory of the overparameterized case to a non-Gaussian input. However, we still assume that the target output can be expressed as $y = \mathbf{w}^T \mathbf{k}(\mathbf{x}, \boldsymbol{\omega}_0) + z$, for some value of $\mathbf{w}$ and some noise z of zero mean and variance σ^2. We do not assume any form for the distributions of $\mathbf{k}$ and $\mathbf{w}$, except that the fourth moment of $\mathbf{k}$ and the second moment of $\mathbf{w}$ exist. For simplicity, we assume $\boldsymbol{\omega}_0 = \mathbf{0}$, $f(\mathbf{x}, 0) = 0$, $\mathbb{E}\mathbf{w} = \mathbf{0}$, $\mathbb{E}\mathbf{k} = \mathbf{0}$. Then, the average test loss is equal to

$$\overline{\mathcal{L}}^{test} \simeq \frac{1}{2}\mathrm{Tr}\left(\Sigma_w H^r\right) + \frac{\sigma^2}{2}\left[1 + \frac{\alpha_r^2}{n_r}\mathrm{Tr}\left(\Sigma^2\right)\right] + \tag{15}$$

$$+ \frac{1}{2} n_v m \frac{\mathrm{Tr}\left(H^r H^t\right)\left\{\mathrm{Tr}\left(\Sigma_w H^t\right) + \sigma^2\left[1 + \frac{\alpha_t^2}{n_t}\mathrm{Tr}\left(\Sigma^2\right)\right]\right\}}{\mathrm{Tr}\left(H^t\right)^2} \tag{16}$$

where we define the following matrices

$$H^t = \left[\Sigma\left(I - \alpha_t \Sigma\right)^2 + \frac{\alpha_t^2}{n_t}\left(F - \Sigma^3\right)\right] \tag{17}$$

$$H^r = \left[\Sigma\left(I - \alpha_r \Sigma\right)^2 + \frac{\alpha_r^2}{n_r}\left(F - \Sigma^3\right)\right] \tag{18}$$

$$\Sigma = \mathbb{E}\,\mathbf{k}\mathbf{k}^T \tag{19}$$

$$\Sigma_w = \mathbb{E}\,\mathbf{w}\mathbf{w}^T \tag{20}$$

$$F = \mathbb{E}\left(\mathbf{k}^T \Sigma \mathbf{k}\right)\mathbf{k}\mathbf{k}^T \tag{21}$$

This expression is valid for p much larger than the total number of validation data across tasks $n_v m$. Note that all matrices involved, Σ, Σ_w, F, H^t, H^r are all of shape $p \times p$. However, as mentioned in the appendix, the validity of the expression may depend more on the covariance Σ rather than the actual value of p, since correlations in the input may reduce its effective dimensionality. Note that these expressions reduce to 5, 6, 7 when $\Sigma = I$, $\Sigma_w = I\nu^2/p$, $F = I(p+2)$, $\boldsymbol{\omega}_0 = \mathbf{0}$, $\mathbf{w} = \mathbf{0}$.

This expression for the loss is more difficult to analyze than those given in the previous sections, because it involves traces of nonlinear functions of matrices, all elements of which are free hyperparameters. Nevertheless, it is possible to show that, as a function of the adaptation learning rate α_r, the loss 15 is still a quadratic function. As a function of the adaptation learning rate α_r, the loss 15 is the ratio of two fourth order polynomials, but it is difficult to draw any conclusions since their coefficients do not appear to have simple relationships.

Even if the influence of the hyperparameters is not easy to predict, the expression 15 can still be used to quickly probe the behavior of the loss empirically, by using example values for the Σ, Σ_w, F, since computing the expression is very fast. Here we choose values of Σ, Σ_w by a single random draw from a Wishart distribution

$$\Sigma \sim \mathcal{W}(I, p) \qquad \Sigma_w \sim \frac{\nu^2}{p}\mathcal{W}(I, p) \tag{22}$$

Note that the number of degrees of freedom of the distribution is equal to the size of the matrices, p, therefore this covariances display significant correlations, although pairs of correlated components of $\mathbf{k}$, $\mathbf{w}$ are random. Furthermore, we choose $F = 2\Sigma^3 + \Sigma\mathrm{Tr}\left(\Sigma^2\right)$, which is the value taken when $\mathbf{k}$ follows a Gaussian distribution. Therefore, we effectively test the loss 15 for a Gaussian distribution, as in previous section, but we stress that the expression is valid for any distribution of $\mathbf{k}$ with existing fourth moment. Since we test the Gaussian case, we also run experiments of MAML, applied again to mixed linear regression, but now using the covariance matrices drawn in 22. Figure 3 shows the loss 15 as a function of the learning rates, during adaptation (panel a) and training (panel b). Qualitatively, we observe a similar behaviour as in section 4.1: the adaptation learning rate has a unique minimum for a positive value of α_r, while the training learning rate shows better performance for negative values of α_t. Again, there is a good agreement between theory and experiment, suggesting that the approximation 15 is valid in this case as well.

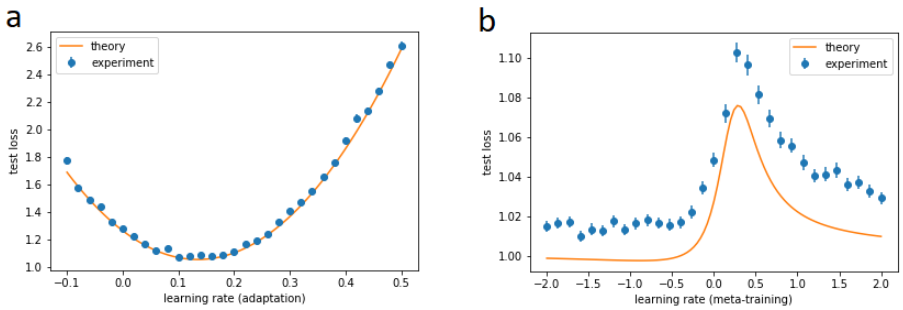

Figure 3: Average test loss of MAML as a function of the learning rate, on overparameterized mixed linear regression with Wishart covariances, as predicted by our theory and confirmed in experiments. a) Effect of learning rate α_r during adaptation. b) Effect of learning rate α_t during training. The optimal learning rate during adaptation is positive, while that during training appears to be negative. Values of parameters: $n_t = 30, n_v = 2, n_r = 20, m = 3, p = 60, \sigma = 1., \nu = 0.5$, $\boldsymbol{\omega}_0 = \mathbf{0}$, $\mathbf{w}_0 = \mathbf{0}$. In panel a) we set $\alpha_t = 0.2$, in panel b) we set $\alpha_r = 0.2$. In the experiments, each run is evaluated on 100 tasks of 50 data points each, and each point is an average over 100 runs (a) or 500 runs (b).

6 Discussion

We calculated algebraic expressions for the average test loss of MAML applied to a simple family of linear models, as a function of the hyperparameters. Surprisingly, we showed that the optimal value of the learning rate of the inner loop during training is negative. This finding seems to carry over to more interesting nonlinear models in the overparameterized case. However, additional work is necessary to establish the conditions under which the optimal learning rate may be positive, for example by probing more extensively Eq.15.

A negative optimal learning rate is surprising and counter-intuitive, since negative learning rates push parameters towards higher values of the loss. However, the meta-training loss is minimized by the outer loop, therefore it is not immediately obvious whether the learning rate of the inner loop should be positive, and we show that in some circumstances it should not. However, perhaps obviously, we also show that the learning rate during adaptation at test time should always be positive, otherwise the target task cannot be learned.

In this work, we considered the case of nonlinear models in the overparameterized case. However, typical applications of MAML (and meta-learning in general) implement relatively small models due to the heavy computational load of running bi-level optimization, including both outer and inner loop. Our theory also assumes a limited number of tasks where data is independently drawn in each task, while some applications use a large number of tasks with correlated draws (for example, images may be shared across tasks in few-shot image classification, see Bertinetto et al. (2019)). Furthermore, our theory is valid at the exact optimum of the outer loop, which is equivalent to training the outer loop to convergence, therefore overfitting may occur in the outer loop of our model.

To the best of our knowledge, nobody has considered before training meta-learning models with negative learning rates in the inner loop. Given that some studies advocate removing the inner loop altogether, which is similar to setting the learning rate to zero, then they may as well try a negative one. On the other hand, it is possible that a negative learning rate does not work in nonlinear and non-overparameterized models, a setting that is outside the the theory presented in this work.

7 Acknowledgements

We would like to thank *anonymous* for the help in calculating a averages over random matrix products of order higher than six.

REFERENCES

Madhu S. Advani and Andrew M. Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv:1710.03667 [physics, q-bio, stat]*, October 2017. URL http://arxiv.org/abs/1710.03667. arXiv: 1710.03667.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv:1805.08136 [cs, stat]*, July 2019. URL http://arxiv.org/abs/1805.08136. arXiv: 1805.08136.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A Closer Look at Few-shot Classification. *arXiv:1904.04232 [cs]*, January 2020a. URL http://arxiv.org/abs/1904.04232. arXiv: 1904.04232.

Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A New Meta-Baseline for Few-Shot Learning. *arXiv:2003.04390 [cs]*, March 2020b. URL http://arxiv.org/abs/2003.04390. arXiv: 2003.04390.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On Lazy Training in Differentiable Programming. *arXiv:1812.07956 [cs, math]*, January 2020. URL http://arxiv.org/abs/1812.07956. arXiv: 1812.07956.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Learning To Learn Around A Common Mean. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10169–10179. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8220-learning-to-learn-around-a-common-mean.pdf.

Guneet S. Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A Baseline for Few-Shot Image Classification. *arXiv:1909.02729 [cs, stat]*, March 2020. URL http://arxiv.org/abs/1909.02729. arXiv: 1909.02729.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *ICML*, pp. 9, 2014.

Chelsea Finn and Sergey Levine. Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm. *arXiv:1710.11622 [cs]*, February 2018. URL http://arxiv.org/abs/1710.11622. arXiv: 1710.11622.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*, March 2017. URL http://arxiv.org/abs/1703.03400. arXiv: 1703.03400.

Micah Goldblum, Steven Reich, Liam Fowl, Renkun Ni, Valeriia Cherepanova, and Tom Goldstein. Unraveling Meta-Learning: Understanding Feature Representations for Few-Shot Tasks. *arXiv:2002.06753 [cs, stat]*, March 2020. URL http://arxiv.org/abs/2002.06753. arXiv: 2002.06753.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. RECASTING GRADIENT-BASED META-LEARNING AS HIERARCHICAL BAYES. pp. 13, 2018.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in High-Dimensional Ridgeless Least Squares Interpolation. *arXiv:1903.08560 [cs, math, stat]*, November 2019. URL http://arxiv.org/abs/1903.08560. arXiv: 1903.08560.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-Learning in Neural Networks: A Survey. *arXiv:2004.05439 [cs, stat]*, April 2020. URL http://arxiv.org/abs/2004.05439. arXiv: 2004.05439.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv:1806.07572 [cs, math, stat]*, June 2018. URL http://arxiv.org/abs/1806.07572. arXiv: 1806.07572.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Multi-Step Model-Agnostic Meta-Learning: Convergence and Improved Algorithms. *arXiv:2002.07836 [cs, math, stat]*, February 2020. URL http://arxiv.org/abs/2002.07836. arXiv: 2002.07836.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *arXiv:2001.08361 [cs, stat]*, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv: 2001.08361.

Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. *arXiv:2002.08936 [cs, stat]*, February 2020. URL http://arxiv.org/abs/2002.08936. arXiv: 2002.08936.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14539. URL http://www.nature.com/articles/nature14539.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. *arXiv:1902.06720 [cs, stat]*, February 2019. URL http://arxiv.org/abs/1902.06720. arXiv: 1902.06720.

Preetum Nakkiran. More Data Can Hurt for Linear Regression: Sample-wise Double Descent. *arXiv:1912.07242 [cs, math, stat]*, December 2019. URL http://arxiv.org/abs/1912.07242. arXiv: 1912.07242.

Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010. ISSN 1041-4347. doi: 10.1109/TKDE.2009.191. URL http://ieeexplore.ieee.org/document/5288526/.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. *arXiv:1909.09157 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1909.09157. arXiv: 1909.09157.

Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A CONSTRUCTIVE PREDICTION OF THE GENERALIZATION ERROR ACROSS SCALES. pp. 30, 2020.

Nikunj Saunshi, Yi Zhang, Mikhail Khodak, and Sanjeev Arora. A Sample Complexity Separation between Non-Convex and Convex Meta-Learning. *arXiv:2002.11172 [cs, math, stat]*, February 2020. URL http://arxiv.org/abs/2002.11172. arXiv: 2002.11172.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need? *arXiv:2003.11539 [cs]*, June 2020. URL http://arxiv.org/abs/2003.11539. arXiv: 2003.11539.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. *arXiv:1903.03096 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1903.03096. arXiv: 1903.03096.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *arXiv:1606.04080 [cs, stat]*, December 2017. URL http://arxiv.org/abs/1606.04080. arXiv: 1606.04080.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. On the Global Optimality of Model-Agnostic Meta-Learning. *arXiv:2006.13182 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2006.13182. arXiv: 2006.13182.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv:1411.1792 [cs]*, November 2014. URL http://arxiv.org/abs/1411.1792. arXiv: 1411.1792.