META-LEARNING AS LEARNING THE META: A VIDEOGAME-THEORETIC PERSPECTIVE ON LEARNING TO LEARN

Anonymous authors
Paper under double-blind review

ABSTRACT

The field of meta-learning involves, not the training of a model for a particular task, but using training on a variety of related tasks to develop a model that transfers readily and generally to any of a set of similar tasks from some distribution. Similarly, top players of competitive videogames are required to contend with frequent updates or patches to their game of choice, and are expected to adapt to the changes quickly and without a significant decrease in skill. By considering each patch of a competitive game to be a separate task, drawn from a distribution describing versions thereof, the adaptation of players to new patches becomes analogous to the adaptation of models to new tasks. This paper seeks to describe the process of players adapting to updates of their games in the language of meta-learning, and then to analyze that process to inspire future modifications to the meta-learning paradigm.

1 INTRODUCTION

1.1 META-LEARNING, OR LEARNING TO LEARN

Meta-learning is the field of study associated with training models on a distribution of related learning tasks, such that it will be especially capable of transferring to new tasks with minimal resources, differing from ordinary transfer learning in that the base model's objective is specifically to transfer well, rather than to perform well on some other task, and hoping that the priors transfer Vilalta & Drissi (2002). For example, rather than training a model to identify images of cars, and then fine-tuning it to identify pickup trucks, meta-learning involves preparing a model that can quickly adapt to identifying cars, or trucks, or RVs, or boats, and so on.

Taking Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) as our formulation, the process is as follows:

This produces a parameterization θ that, in theory, minimizes the loss of its child ϕ_k when fine-tuned to a task $\mathcal{T}_k$ from the same distribution $p(\mathcal{T})$, under all the same restraints as existed in the inner loop, e.g. using only a limited number of update steps with a small number of batches. If $p(\mathcal{T})$ contains tasks like identifying cars and identifying trucks, then $\mathcal{T}_k$ could be a task like identifying RVs, and the expectation would be that θ could be fine-tuned into ϕ_k with only a few labeled pictured of RVs. This is achieved, in some sense, by imbuing θ with a general understanding of the nature of the tasks its children are charged with, as well as commonly applicable tricks for specific tasks, that either do not interfere with or can be ignored when a child is trained for another task.

1.2 PATCHES IN COMPETITIVE GAMES

For competitive videogames, like League of Legends, DotA 2, Overwatch, or even Pokémon, frequently the publishers of a game will make modifications to their mechanics in order to expand, promote, or de-emphasize certain aspects of gameplay noa (a;b;c;d). Typically, this will include things like the introduction of new characters or items, changes to existing elements of the game, or technological improvements and fixes. It is unusual, but not particularly common, for elements to be removed from a game, rather than modified. Groups of changes, fixes, and additions are ordinarily

Algorithm 1: Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: the distribution from which tasks can be sampled;
Require: $\mathcal{L}$: a differentiable loss function, that potentially differs by task;
Require: $\mathcal{A}lg_{inner},\ \mathcal{A}lg_{outer}$: update algorithms, e.g. SGD, Adam;
Randomly initialize θ, parameterizing a model $h(\theta;\ \cdot\ )$;
while θ *not converged* **do**
  Sample a batch of tasks $b \coloneqq \{\mathcal{T}_1,\ \mathcal{T}_2, \ldots,\ \mathcal{T}_n\}$ from $p(\mathcal{T})$;
  for $\mathcal{T}_i \in b$ **do**
    Create a new child model with parameters $\phi_i \leftarrow \theta$;
    In a limited fashion, train ϕ_i on its task $\mathcal{T}_i$ using $\mathcal{A}lg_{inner}$, e.g. by sampling only a few batches (x_{ij}, y_{ij}) from $\mathcal{T}_i$ and setting $\phi_i \leftarrow \mathcal{A}lg_{inner}\Big(\phi_i, \nabla_{\phi_i}\mathcal{L}\big(y_{ij}, h(\phi_i; x_{ij})\big)\Big)$;
  end
  Update θ based on its contribution to the errors of its child models on their tasks, e.g. by sampling a new batch (x_{ij}, y_{ij}) from each task $\mathcal{T}_i \in b$, and setting $\theta \leftarrow \mathcal{A}lg_{outer}\Big(\theta, \nabla_\theta \sum_i \mathcal{L}\big(y_{ij}, h(\phi_i; x_{ij})\big)\Big)$;
end

applied simultaneously, in the form of a ”patch,” with the specific version of the game being denoted as ”patch k,” after the k^{th} patch has been applied. For example, on March 23$^{\text{rd}}$, 2016, League of Legends received Patch 6.6, introducing the playable character ”Aurelion Sol,” slightly modifying items like the ”Tiamat” and the ”Zz’Rot Portal,” and tweaking the capabilities of player characters like ”Kalista,” ”Azir,” and ”Master Yi.” From that date until April 6$^{\text{th}}$, when Patch 6.7 was released, you would say that you were playing on Patch 6.6 noa (e). It is worth noting that it is extremely rare for any particular patch to modify, add, or remove the most core mechanics from the game to which they are applied.

During a particular patch, what is called a ”meta-game,” often abbreviated to ”meta,” will develop. This meta-game refers to processes of decision making that arise from, but are not inherent to, the game to which they apply Pell (1993); Donaldson (2017). For example, during a patch wherein characters that deal large and consistent amounts of damage, called ”carries,” are capable of easily knocking out opponents, and characters expected to be capable of absorbing damage, called ”tanks,” have greater difficulty doing so, the meta-game may revolve around the effective utilization of carries, as well as characters themselves capable of swiftly cutting down carries with instantaneous bursts of damage, called ”assassins.” But, in the next patch, if the prospects of tanks improve, whether by improvements to their statistics or the addition of items that aid in damage absorption, then the prior state of the meta-game will adapt to a new one with a lesser emphasis on carries, and therefore reward assassins to a lesser degree. However, despite the fact that the meta-game is subject to the game’s mechanics, it is a function of the playerbase, and no amount of theorizing or historical comparison will allow a pro-player to avoid actually playing matches to determine the present state of affairs.

2 Patch Adaptation as Sequential Learning

Those playing competitively have to deal with these changes between patches, not only becoming skilled with new characters or mechanics, but also adapting their understanding of the meta-game. However, because each individual patch will only have relatively minor changes compared to previous ones, skilled players with large quantities of prior knowledge will continue to be more skilled than those without. In addition, as skilled players increase the total amount of time that they have played, the over-all skill level increases. This is partially due to an increasing level of understanding of how the meta-game changes due to particular elements of a patch, as somebody who has seen how increasing the power of carries results in a greater value assigned to assassins will more readily choose to play assassins when they see similar changes in the future.

Thus, the process of becoming skilled at a competitive game may be modeled as a sequential learning and adaptation problem, with a focus on using historical data to understand future changes McCloskey & Cohen (1989); Beaulieu et al. (2020).

Algorithm 2: Competitive Videogame Career Practice

Require: $\mathcal{T}_0$: the initial content of the game;
Require: $\mathcal{R}$: a differentiable reward function, used to describe the level of skill in one's play;
Require: $\mathcal{A}lg_\phi, \mathcal{A}lg_\mathcal{M}$: update algorithms, e.g. SGD or Adam;
Randomly initialize $\theta_\mathcal{P}$, parameterizing a presently unskilled game-playing agent $\mathcal{P}(\theta_\mathcal{P};\ \cdot\ )$;
Randomly initialize $\theta_\mathcal{M}$, parameterizing a model $\mathcal{M}(\theta_\mathcal{M};\ \cdot\ )$ representing an understanding of how to change one's play to adapt to a new patch;
while *actively playing* **do**
  Apply patch δ_n to the previous game content $\mathcal{T}_{n-1}$, to get $\mathcal{T}_n$;
  Anticipate how your play will change to adapt to the new mechanics and meta-game, $\phi_n \leftarrow \theta_\mathcal{P} + \mathcal{M}(\theta_\mathcal{M}; \mathcal{T}_{n-1}, \delta_n, \theta_\mathcal{P})$;
  while *game content is still* $\mathcal{T}_n$ **do**
    Play a match as agent $\mathcal{P}(\phi_n;\ \cdot\ )$, recording the moment-to-moment game-states (s_t);
    Update your agent to increase your expected reward at different game-states, $\phi_n \leftarrow \mathcal{A}lg_\phi\Big(\phi_n, -\nabla_{\phi_n}\mathcal{R}\big(s_t, \mathcal{P}(\phi_n; s_t)\big)\Big)$;
  end
  Judge what the appropriate reaction to δ_n would've been, $\Delta_n \leftarrow \phi_n - \theta_\mathcal{P}$, and update your understanding of meta-games to more closely resemble that $\theta_\mathcal{M} \leftarrow \mathcal{A}lg_\mathcal{M}\big(\theta_\mathcal{M}, \nabla_{\theta_\mathcal{M}}||\Delta_n - \mathcal{M}(\theta_\mathcal{M}; \mathcal{T}_{n-1}, \delta_n, \theta_\mathcal{P})||\big)$;
  Update your understanding of the game in general, based on how you got better over the season, accounting for how you would now would've changed to approach the meta-game, $\theta_\mathcal{P} \leftarrow \phi_n - \mathcal{M}(\theta_\mathcal{M}; \mathcal{T}_{n-1}, \delta_n, \theta_\mathcal{P})$;
end

---

3 Implications for meta-learning

We can now try to translate the competitive videogame career practice algorithm into something more closely resembling the meta-learning problem statement, in order to compare and contrast it with existing approaches, and to inspire future work in the field.

This combined algorithm, here noted as SSCA, ends up greatly resembling a number of prior algorithms. It fine-tunes child models to their particular tasks via gradient descent, which it shares with MAML (Finn et al., 2017). It provides each child model with an initial modification to prepare it for a task, which epistemologically resembles the tokens from Contextual meta-learning algorithms (Zhang et al., 2020). The update scheme for the parent model ends up being quite similar to REPTILE, in that it becomes feasible to simply take the spacial mean of the child models once their recommended initial modifications have been subtracted out (Nichol et al., 2018).

4 Conclusions and Future Work

Adapting to new patches in videogames, as discussed, represent an interesting real-world case of learning to learn, and so provide a rich field to explore to gain intuitions about new meta-learning methodologies; we have demonstrated that an examination of the processes involved can quickly yield a combination of the core elements of a number of prior meta-learning algorithms. Thus, it may yield significant fruit to interview top competitive players and their coaches, in order to understand their thought processes to a greater extent. It might also be valuable to design a new game, under more sterile conditions, to study the entire process from beginning to end.

Algorithm 3: Semi-Sequential-Contextual-Adaptive Meta-Learning

Require: $p(\mathcal{T})$: the distribution from which tasks can be sampled;
Require: $\mathcal{L}$: a differentiable loss function, that potentially differs by task;
Require: $\mathcal{A}lg_h,\ \mathcal{A}lg_c$: update algorithms, e.g. SGD, Adam;
Randomly initialize θ_h, parameterizing a model $h(\theta_h;\ \cdot\ )$;
Randomly initialize θ_c, parameterizing a context embedder $c(\theta_c;\ \cdot\ )$;
while θ_h, θ_c *not converged* **do**
  Sample a batch of tasks $b \coloneqq \{\mathcal{T}_1,\ \mathcal{T}_2, \ldots,\ \mathcal{T}_n\}$ from $p(\mathcal{T})$;
  for $\mathcal{T}_i \in b$ **do**
    Create a new child model from the parent with suggested contextual changes $\phi_i \leftarrow \theta_h + c(\theta_c; \mathcal{T}_i, \theta_h)$;
    In a limited fashion, train ϕ_i on its task $\mathcal{T}_i$ using $\mathcal{A}lg_{inner}$, e.g. by sampling only a few batches (x_{ij}, y_{ij}) from $\mathcal{T}_i$ and setting $\phi_i \leftarrow \mathcal{A}lg_{inner}\Big(\phi_i, \nabla_{\phi_i}\mathcal{L}\big(y_{ij}, h(\phi_i; x_{ij})\big)\Big)$;
    Determine what the appropriate contextual change would've been, as $\Delta_i = \phi_i - \theta_h$;
  end
  Update θ_c based on how far off its suggestions were from the appropriate changes, $\theta_c \leftarrow \mathcal{A}lg_c\big(\theta_c, \nabla_{\theta_c} \sum_i ||\Delta_i - c(\theta_c; \mathcal{T}_i, \theta_h)||\big)$;
  Update θ_h to what would now have been the recommended starting point, $\theta_h \leftarrow \frac{1}{n}\sum_i \big(\phi_i - c(\theta_c; \mathcal{T}_i, \theta_h)\big)$;
end

REFERENCES

Game Updates, a. URL `https://www.dota2.com/news/updates/`.

Overwatch Patch Notes, b. URL `https://playoverwatch.com/en-us/news/patch-notes/`.

Patch Notes - League of Legends, c. URL `https://na.leagueoflegends.com/en-us/news/tags/patch-notes/`.

Patches & Updates - Pokémon Sword & Shield, d. URL `https://www.serebii.net/swordshield/patch.shtml`.

V6.6, e. URL `https://leagueoflegends.fandom.com/wiki/V6.6`.

Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. *arXiv preprint arXiv:2002.09571*, 2020.

Scott Donaldson. Mechanics and metagame: Exploring binary expertise in league of legends. *Games and Culture*, 12(5):426–444, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Barney Pell. *Strategy generation and evaluation for meta-game playing*. PhD thesis, Citeseer, 1993.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 2020.