WARP: On the Benefits of Weight Averaged Rewarded Policies

Amit Kumar

AI/NLP Engineer at E42.ai

Table of contents

1. Reinforcement learning from human feedback (RLHF)
2. Challenges in RLHF
3. Weight Averaged Rewarded Policies (WARP)
4. Stage 1: Exponential Moving Average (EMA)
5. Stage 2: Spherical Linear intERPolation of task vectors (SLERP)
6. Stage 3: Linear Interpolation Towards Initialization (LITI)
7. Results

Introduction

- **RLHF Alignment**: Reinforcement Learning from Human Feedback (RLHF) aligns large language models (LLMs) by encouraging outputs that receive high rewards based on human preferences.
- **KL Regularization Challenge**: RLHF often uses KL regularization to prevent the model from forgetting its pre-trained knowledge, but this can limit reward optimization.
- **WARP Strategy**: Introduces Weight Averaged Rewarded Policies (WARP) to address the trade-off between KL regularization and reward optimization.
- **Iterative Refinement**: Applies WARP iteratively, using the final model of each iteration to further improve performance in the next.
- **Improved Performance**: Experiments demonstrate that WARP enhances model quality and alignment, outperforming other open-source LLMs.

RLHF

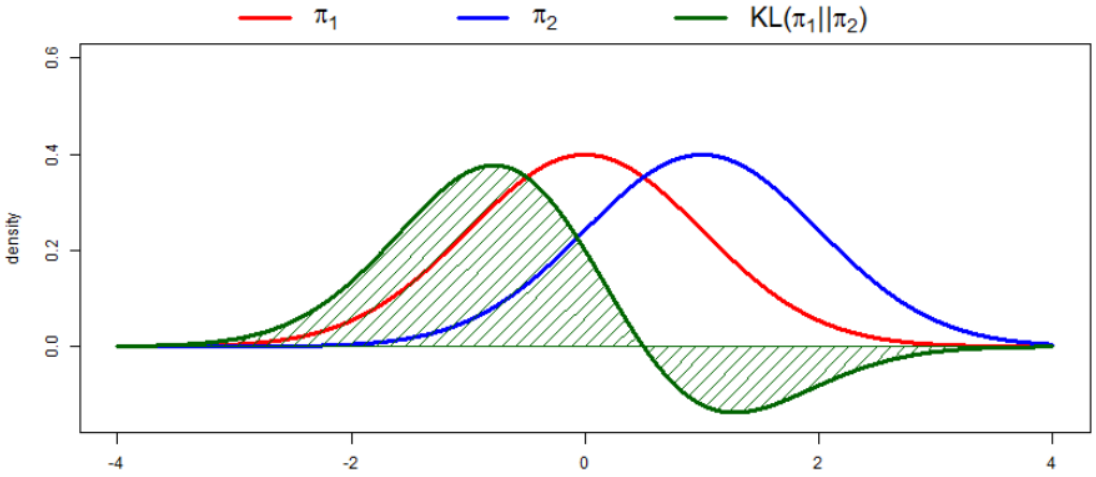
- RLHF can be seen as these 3 points:
  1. Collect human data:
     X = "Can you help me with 1+1?"
     Y1 = " .... something 2" (1)
     Y2 = " .... something 3" (0)
  2. Train your reward model on human data using a pre-trained model with LORA. For the final token, use sigmoid binary classification (0/1), or softmax if there are multiple classes.
  3. Fine-tune the base model M with RLHF (PPO/AC2) using a dataset of prompts other than human-generated data, and aim to maximize rewards through this process.

$$\underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{x \in \mathcal{X}} \left[\mathbb{E}_{y \sim \pi_\theta(\cdot|x)} r(x, y) - \beta \mathrm{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_{\theta_{\text{anchor}}}(\cdot \mid x)\right) \right],$$

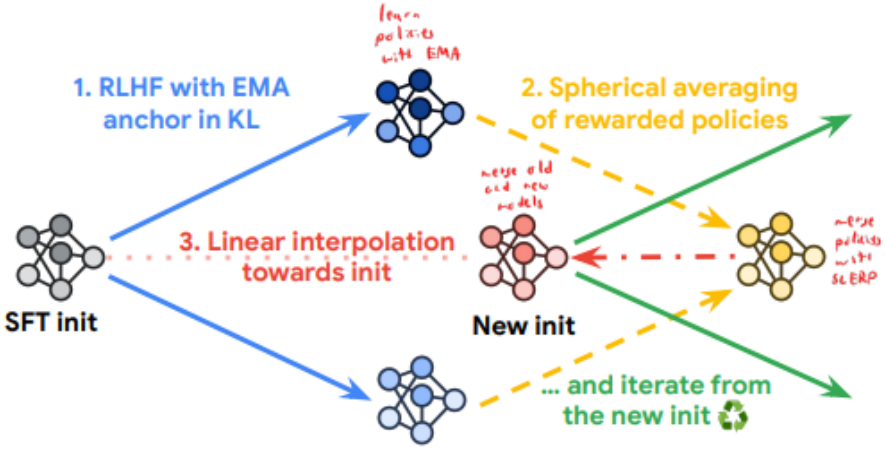
Limitations of RLHF

- **Excessive Specialization**: Fine-tuning with RLHF on small datasets can cause the model to forget its broad, pre-trained knowledge, leading to a loss in its overall reasoning capabilities.
- **Reward Hacking**: The model might exploit weaknesses in the reward system, producing flawed, verbose, or overly agreeable responses, raising concerns about alignment and safety.
- **Reduced Diversity**: RLHF can limit the variety of the model's responses, making it less effective for creative tasks and sometimes even causing it to refuse to answer certain prompts.

Kullback-Leibler divergence (KL)



WARP peek


learn policy with EMA
1. RLHF with EMA anchor in KL
2. Spherical averaging of rewarded policies
merge old and new models
3. Linear interpolation towards init
merge policies with SLERP
SFT init
New init
... and iterate from the new init

Weight Averaged Rewarded Policies (WARP)

Stage 1: *Exponential Moving Average* (*EMA*). During RL fine-tuning, instead of regularizing the policy towards the SFT initialization, *WARP* uses the policy's own exponential moving average [100] as a dynamic updatable anchor in the KL. This stage enables stable exploration with distillation from a mean teacher [127] and annealed constraint.

Stage 2: *Spherical Linear intERPolation of task vectors* (*SLERP*). Considering M policies RL fine-tuned independently with their own *EMA* anchor, we merge them by spherical linear interpolation [118] of their task vectors [53]. This stage creates a merged model with higher reward by combining the strengths of the M individual policies.

Stage 3: *Linear Interpolation Towards Initialization* (*LITI*). Considering the merged policy from *SLERP*, *WARP* linearly interpolates towards the initialization, akin to WiSE-FT [138]. This stage allows to run through an improved Pareto-front simply by adjusting the interpolating coefficient η between 1 (high reward but high KL) and 0 (small KL but small reward). Critically, selecting an intermediate value for $0 < \eta < 1$ offers a balanced model that can serve as a new, improved initialization for subsequent iterations of *WARP*.

Stage 1: Exponential Moving Average (EMA)

- KL penalty over the exponential moving average of the policy instead of the old policy as an anchor as RLHF

$$\theta_{\text{ema}} \leftarrow (1 - \mu) \cdot \theta_{\text{ema}} + \mu \cdot \theta_{\text{policy}}.$$

- Unlike a static SFT anchor, the dynamic nature of an EMA anchor induces relaxation of the KL regularization making it bit softer than original.

Stage 2: Spherical Linear intERPolation of task vectors (SLERP)

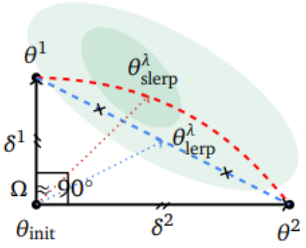
- While EMA helps for a single RL and a fixed compute budget, it faces limitations due to the similarity of the weights collected along a single fine-tuning.
- In this second stage, we merge weights RL fine-tuned independently (each with their own EMA anchor).
- Weight Average improves generalization, and that task vectors (the difference between fine-tuned weights and their initialization) can be arithmetically manipulated by linear interpolation (LERP)

$$\theta = \theta_{\text{init}} + \lambda\delta_1 + (1-\lambda)\delta_2$$

SLERP vs. LERP

$$\text{slerp}\left(\theta_{\text{init}}, \theta^1, \theta^2, \lambda\right) = \theta_{\text{init}} + \frac{\sin[(1-\lambda)\Omega]}{\sin\Omega} \cdot \delta^1 + \frac{\sin[\lambda\Omega]}{\sin\Omega} \cdot \delta^2,$$

- where Ω is the angle between the two task vectors $\delta_1 = \theta_1 - \theta_{\text{init}}$ and $\delta_2 = \theta_2 - \theta_{\text{init}}$, and λ is the interpolation coefficient.



SLERP

- **SLERP (Spherical Linear Interpolation)**: Increases rewards but slightly raises the KL divergence, as shown by empirical evidence and theoretical insights.
- **LERP (Linear Interpolation)**: Lowers the KL divergence but has a smaller impact on boosting rewards, supported by empirical and theoretical analysis.
- **Task Vectors**: The task vectors δ are nearly orthogonal (angle $\approx 90°$), while the full weight vectors θ are collinear.

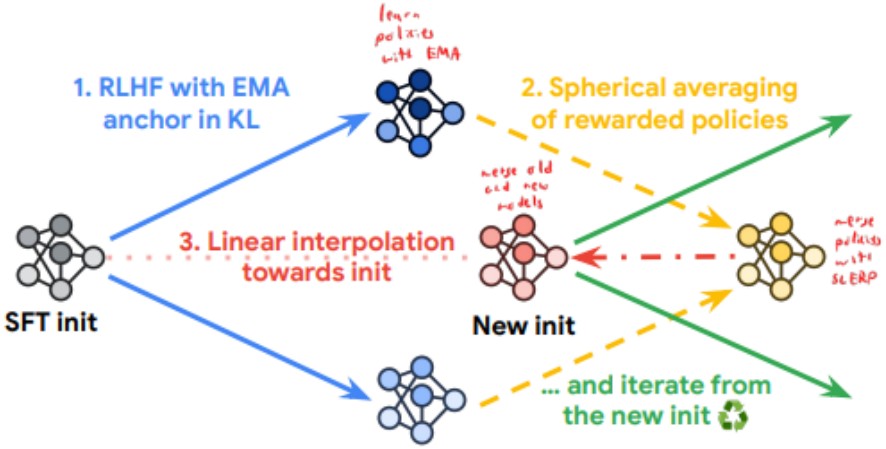
Stage 3: Linear Interpolation Towards Initialization (LITI)

- In the previous stage, SLERP combines multiple policies into one with higher rewards and slightly higher KL. This third stage we interpolates from the merged model towards the initialization:

$$\theta^{\eta} \leftarrow (1-\eta) \cdot \theta_{\text{init}} + \eta \cdot \theta_{\text{slerp}}.$$

- Adjusting the interpolating coefficient $\eta \in [0, 1]$ trades off between some newly acquired behaviors leading to high rewards vs. general knowledge from the SFT initialization.

WARP complete



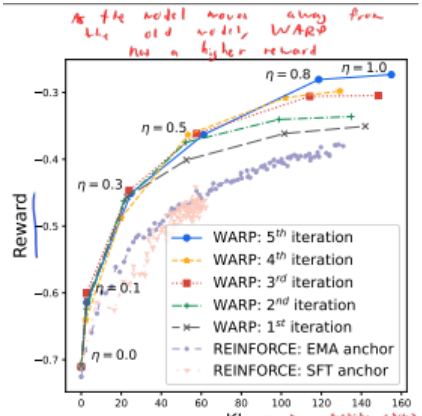
Code

Input: Weights θ_{sft} pre-trained and supervised fine-tuned
Reward model r, prompt dataset $\mathcal{X}$, optimizer Opt
I iterations with M RL runs each for T training steps
μ *EMA* update rate, η *LITI* update rate

Define $\theta_{\text{init}} \leftarrow \theta_{\text{sft}}$
for iteration i from 1 to I **do**
  for run m from 1 to M **do** ▷ Run in parallel
    Define $\theta^m, \theta^m_{\text{ema}} \leftarrow \theta_{\text{init}}$
    for step t from 1 to T **do**
      Generate completion $\boldsymbol{y} \sim \pi_{\theta^m}(\cdot \mid \boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}$
      Compute $r_\beta(\boldsymbol{y}) \leftarrow r(\boldsymbol{x}, \boldsymbol{y}) - \beta \log \frac{\pi_{\theta^m}(\boldsymbol{y}|\boldsymbol{x})}{\pi_{\theta^m_{\text{ema}}}(\boldsymbol{y}|\boldsymbol{x})}$ ▷ KL regularized reward
      Update $\theta^m \leftarrow \text{Opt}(\theta^m, r_\beta(\boldsymbol{y}) \nabla_\theta [\log \pi_{\theta^m}(\boldsymbol{y} \mid \boldsymbol{x})])$ ▷ Policy gradient
      Update $\theta^m_{\text{ema}} \leftarrow (1-\mu) \cdot \theta^m_{\text{ema}} + \mu \cdot \theta^m$ ▷ Equation (*EMA*): update anchor
    end for
  end for
  Define $\theta^i_{\text{slerp}} \leftarrow \text{slerp}\left(\theta_{\text{init}}, \{\theta^m\}_{m=1}^M, \lambda = \frac{1}{M}\right)$ ▷ Equation (*SLERP*): merge M weights
  Update $\theta_{\text{init}} \leftarrow (1-\eta) \cdot \theta_{\text{init}} + \eta \cdot \theta^i_{\text{slerp}}$ ▷ Equation (*LITI*): interpolate towards init
end for

Output: KL-reward Pareto front of weights $\{(1-\eta) \cdot \theta_{\text{sft}} + \eta \cdot \theta^I_{\text{slerp}} \mid 0 \leq \eta \leq 1\}$

Results : Fine-tuning trajectories



(b) KL-reward Pareto front.

Benchmark results

- WARP has particularly strong results on mathematics benchmarks suggesting higher analytical capabilities

Methods	MBPP	MMLU	GSM8K	MATH	HumanEval	BBH
Gemma "7B" 1.1	39.0	56.4	55.6	25.6	46.9	53.1
WARP	**45.4**	**57.6**	**66.8**	**31.0**	**50.0**	**58.8**

Conclusion

1. Weight Averaged Rewarded Policies (WARP) is a new RLHF strategy designed to align large language models (LLMs) through a three-stage model merging process.
2. WARP involves using an exponential moving average as a dynamic anchor, spherical interpolation to merge independently rewarded policies, and interpolation toward the shared initialization.
3. This approach enhances alignment by improving the balance between the model's knowledge and reward optimization, outperforming current methods.
4. WARP aims to scale alignment in AI systems safely while preserving pre-trained knowledge.

Thank You