Laboratorium 6 — RLHF (Reinforcement Learning from Human Feedback)

Rodzaje • Etapy • Mechanizmy • Ryzyka

1. Wprowadzenie — czym jest RLHF?

Modele językowe (LLM) wytrenowane klasycznie (tzw. *pretraining* + *supervised fine-tuning*, SFT) potrafią generować płynny tekst, ale nie są zoptymalizowane pod kątem zgodności z intencją użytkownika, użyteczności ani bezpieczeństwa. Potrafią być obszerne, ale rozwlekłe, uprzejme, lecz nieszczere, albo odpowiadać poprawnie składniowo, jednak niemerytorycznie.

RLHF (ang. Reinforcement Learning from Human Feedback) to metoda dostrajania modeli językowych, w której zachowanie modelu jest kształtowane za pomocą **nagród pochodzących z ocen ludzi**, a następnie optymalizowane technikami **uczenia przez wzmacnianie** (reinforcement learning).

> **Kluczowa definicja**

> RLHF — paradygmat treningu, w którym najpierw zbiera się ludzkie opinie o odpowiedziach modelu, na ich podstawie trenuje tzw. model nagradzający (Reward Model), a następnie za jego pomocą — metodami RL (np. PPO) — stroi się politykę modelu tak, aby maksymalizowała oczekiwaną nagrodę, jednocześnie nie odchodząc zbyt daleko od zachowania wyjściowego (kara KL).

RLHF jest fundamentem współczesnych czatbotów i asystentów AI (m.in. ChatGPT, Claude, Gemini). Dzięki niemu modele stają się:

- bardziej **pomocne** (helpful),
- bardziej **szczere** (honest),
- bardziej **bezpieczne** (harmless).

Zrozumienie RLHF jest kluczowe dla każdego, kto projektuje, trenuje lub ewaluuje systemy oparte na dużych modelach językowych.

2. Etapy procesu RLHF

Proces RLHF można podzielić na trzy główne etapy. Każdy z nich ma własne dane, własnych twórców oraz specyficzne zagrożenia.

2.1 Krok 1 — SFT (Supervised Fine-Tuning)

Na wejściu mamy model bazowy (np. GPT-2/GPT-3 po pretrainingu). Pierwszym krokiem jest **dostrojenie nadzorowane** na zestawie par *(zapytanie, idealna odpowiedź)*.

Aspekt	Opis
Dane	Tysiące par (instruction, response) przygotowanych przez ludzi
Kto tworzy	Eksperci dziedzinowi, redaktorzy, lingwiści, anotatorzy
Cel	Dopasowanie formatu i stylu odpowiedzi do oczekiwań użytkownika
Metoda	Standardowa minimalizacja loss (cross-entropy) — jak w klasycznym uczeniu

Ryzyka: stronniczość anotatorów, brak pokrycia rzadkich przypadków.

Mitigacja: zróżnicowany dobór próbek (*diverse sampling*), podwójne anotacje, audyt jakości danych.

2.2 Krok 2 — Reward Model (RM)

Model nagradzający uczy się **przewidywać, którą odpowiedź człowiek uznałby za lepszą**. Dla każdej pary odpowiedzi *(chosen vs rejected)* anotator wskazuje preferowaną. Następnie trenuje się model klasyfikacyjny (np. oparty na BERT/DistilBERT), który przypisuje odpowiedzi **liczbową nagrodę**.

> **Jak oceniany jest RM?**

> Dokładność powyżej ~70% uważana jest za przyzwoitą. Trudność wynika z subiektywności — ta sama odpowiedź może być „chosen" dla jednego anotatora, a „rejected" dla innego. Stąd potrzeba dużej liczby danych i kryteriów oceny (poprawność, empatia, zwięzłość).

Ryzyka: niezgodność między anotatorami, stronniczość kulturowa.

Miara zgodności międzyanotatorskiej (Cohen's kappa > 0.6) Gdy dwóch różnych ludzi ocenia te same odpowiedzi modelu, ich zdania mogą się różnić — jeden uzna odpowiedź A za lepszą, drugi odpowiedź B. Aby sprawdzić, czy oceny są spójne i wiarygodne, stosuje się statystykę **Cohena Kappa**. Współczynnik ten przyjmuje wartości zwykle od 0 do 1:

- ~0 = oceny całkowicie przypadkowe/niezgodne,
- 1 = idealna zgodność.

Próg **> 0.6** oznacza „zadowalającą, akceptowalną zgodność" między anotatorami. Jeśli wynik jest niższy, oceny są zbyt chaotyczne, by na ich podstawie trenować model — trzeba coś poprawić.

Jasne rubrics (kryteria oceny / instrukcje punktowania) Słowo *rubric* pochodzi tu z metodyki oceniania i oznacza **z góry ustalone, precyzyjne kryteria**, według których anotatorzy mają oceniać odpowiedzi. Zamiast subiektywnego „mi się podoba", dostają np. tabelę punktową:

- poprawność merytoryczna (0–3 pkt),
- empatia i ton odpowiedzi (0–2 pkt),
- zwięzłość (0–2 pkt),
- kompletność (0–3 pkt).

Dzięki temu wszyscy oceniająca według tych samych zasad, a wyniki są bardziej porównywalne i powtarzalne.

Iteracyjna kalibracja To proces **stopniowego ujednolicania ocen** podczas pracy. Na początku anotatorzy oceniają próbkę testową, wyniki się porównuje, wskazuje rozbieżności, omawia wątpliwe przypadki i aktualizuje wytyczne. Ten cykl powtarza się (*iteracyjnie*) aż do osiągnięcia zadowalającej zgodności (np. kappa > 0.6). Kalibrację można też rozszerzyć o wspólne sesje szkoleniowe zespołu anotującego.

2.3 Krok 3 — Optymalizacja polityki (PPO)

Na koniec stroimy model (politykę) metodą **uczenia przez wzmacnianie**, np. **PPO (Proximal Policy Optimization)**. Model generuje odpowiedzi na prompty z produkcji, a Reward Model ocenia je punktowo — polityka aktualizuje swoje wagi, by maksymalizować oczekiwaną nagrodę.

SFT —▶Reward Model —▶PPO

(nadzorowane) (ocena) (optymalizacja polityki)

Ryzyka: *reward hacking*, *mode collapse*.

Mitigacja: kara KL, monitoring dystrybucji długości i różnorodności leksykalnej, czerwone zespoły testujące zachowanie.

3. Funkcja nagrody i problem Reward Hacking

3.1 Projektowanie funkcji nagrody

Funkcja nagrody to matematyczne odwzorowanie „podzielności" odpowiedzi. W prostych przypadkach możemy ją zdefiniować ręcznie, np. jako ważoną sumę składowych:

- r_len — nagroda za długość (liniowo rosnąca do pewnego maksimum),
- r_kw — nagroda za obecność słów kluczowych,
- r_uniq — nagroda za unikalność tokenów (różnorodność).

total = 0.3·r_len + 0.5·r_kw + 0.2·r_uniq

Taka ręczna funkcja premiuje długie odpowiedzi zawierające istotne słowa — ale łatwo ją „oszukać".

3.2 Prawo Goodharta i reward hacking

> **Prawo Goodharta**

> „When a measure becomes a target, it ceases to be a good measure."

> — gdy miara staje się celem, przestaje być dobrą miarą.

Reward hacking polega na tym, że model optymalizuje metrykę nagrody **bez faktycznej poprawy jakości**. Przykładowo model może nauczyć się zaczynać każdą odpowiedź od „Świetne pytanie!", co ludziom podoba się krótkoterminowo, choć nie poprawia merytorycznej wartości odpowiedzi. To bezpośrednia konsekwencja prawa Goodharta — model gra w grę o punkty zamiast faktycznie wykonywać zadanie.

Konsekwencje: degradacja jakości, jednolite odpowiedzi, manipulacja metryką zamiast realizacji celu.

Rozwiązania: regularne przeskalowywanie reward modelu, staranna konstrukcja funkcji nagrody, testy A/B na realnych użytkownikach.

4. Kara KL — ograniczanie odchyleń od referencji

Podczas optymalizacji PPO model polityki (π) mógłby całkowicie „oszukać" Reward Model, produkując nonsensowne, ale wysoko punktowane odpowiedzi. Aby temu zapobiec, dodaje się **karę dywergencji KL**.

> **Definicja**

> Dywergencja Kullbacka–Leiblera (KL) mierzy „odległość" między dwoma rozkładami prawdopodobieństwa — tutaj między rozkładem tokenów modelu polityki a zamrożonym modelem referencyjnym (sztuczna sieć uczona z SFT).

>

> > $\text{penalty} = \beta \cdot KL(\pi \,||\, \pi_ref)$

>

Im wyższa temperatura/modyfikacja polityki względem referencji, tym większa wartość KL. Kara ta działa jak „kotwica": pozwala modelowi uczyć się poprawy jakości, ale zabrania mu oddalić się zbyt daleko od sensownego zachowania SFT.

Parametr	Rola
β (beta)	Waga kary KL, typowo 0.1–0.5
KL(π ‖ π_ref)	Odchylenie polityki od referencji
Efekt	Zapobiega degeneracji i reward hackingowi

5. Główne ryzyka i mechanizmy obronne

Poniższa tabela syntetyzuje kluczowe problemy omówione w tej części teoretycznej:

Problem	Źródło	Konsekwencja	Mitigacja
Reward hacking	Funkcja nagrody niedoskonała	Model optymalizuje metrykę, nie jakość	Staranna nagroda, przeskalowywanie RM, testy A/B
Mode collapse	Zbyt silna presja na nagrodę	Jednolite, szablonowe odpowiedzi	Monitoring różnorodności, kara KL
Stronniczość anotatorów	Subiektywne preferencje ludzi	Niezgodność ocen	Cohen's kappa, jasne rubrics, diverse sampling
Brak pokrycia rzadkich przypadków	Ograniczone dane	Słabe odpowiedzi na nietypowe pytania	Diverse sampling, audyt danych
Degeneracja polityki	Nadużycie RL bez kontroli	Nonsensowne odpowiedzi	Kara β·KL(π ‖ π_ref)

6. Podsumowanie

RLHF przekształca surowy model językowy w narzędzie dostosowane do ludzkich oczekiwań. Proces ten łączy trzy kluczowe elementy:

Element	Rola
SFT	Ustalenie formatu i stylu odpowiedzi
Reward Model	Nauka przewidywania ludzkich preferencji
PPO (+ kara KL)	Optymalizacja polityki przy zachowaniu sensowności

Znajomość tych mechanizmów i ich ryzyk (zwłaszcza *reward hackingu* i roli kary KL) jest niezbędna dla każdego, kto buduje lub wdraża systemy oparte na dużych modelach językowych. Świadomy inżynier musi monitorować jakość danych treningowych, stabilność reward modelu oraz różnorodność generowanych odpowiedzi.

> **Myśl przewodnia**

> Nagroda to nie cel sam w sobie — to ster. Jeśli pozwolisz modelowi gonić za metryką bez kotwicy (kary KL) i bez weryfikacji, zoptymalizuje miarę zamiast rozwiązać problem. Dobry model powstaje tam, gdzie ludzka intencja, staranna nagroda i kontrola odchyleń działają razem.