Lecture 21: LLM Alignment and Evaluation

Pasquale Minervini
p.minervini@ed.ac.uk
March 8th, 2024
Large Language Models

The blessings of scale
AI training runs, estimated computing resources used
Floating-point operations, selected systems, by type, log scale

Sources: “Compute trends across three eras of machine learning”, by J. Sevilla et al., arXiv, 2022; Our World in Data
Large Language Models

Number of tokens observed during “training”

- <100 Million
  - 13 y.o. Human
- 3 Billion
  - BERT (2018)
- 30 Billion
  - RoBERTa (2019)
- 200 Billion
  - GPT-3 (2020)
- 1.4 Trillion
  - Chinchilla (2022)
Large Language Models

The University of Edinburgh is located in __________, UK. [trivia]

I put __________ fork down on the table. [syntax]

The woman walked across the street, checking for traffic over __________ shoulder. [coreference]

I went to the ocean to see the fish, turtles, seals, and __________. [lexical semantics/topic]

Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was __________. [sentiment]

John went into the kitchen to make some tea. Standing next to John, Jake pondered his destiny. Jake left the __________. [some degree of reasoning]

I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, __________ [some arithmetic reasoning]
Generative Pre-Training: GPT (2018)

Generative Pre-Trained Transformer [Radford et al., 2018]:

• 117M Parameters
• Transformer decoder-only model with 12 layers
• Trained on BookCorpus: >7000 unique books (4.6GB of text)

Shows how language modelling at scale can be an effective pre-training technique for NLU downstream tasks like natural language inference.

[START] The man is in the doorway [DELIM] The person is near the door [EXTRACT]
Encoder-Decoder vs. Decoder-only

\[ x_0, x_1, \ldots, x_m \]

Encoder-Decoder:

\[ h \leftarrow \text{Encoder} \left( x_0, \ldots, x_m \right) \]

\[ y_0, \ldots, y_n \leftarrow \text{Decoder} \left( h \right) \]

e.g., BART, T5
Encoder-Decoder vs. Decoder-only

Encoder-Decoder:
\[
\begin{align*}
h & \leftarrow \text{Encoder} \left( x_0, \ldots, x_m \right) \\
y_0, \ldots, y_n & \leftarrow \text{Decoder} \left( h \right)
\end{align*}
\]
e.g., BART, T5

Decoder-only:
\[
\begin{align*}
y_0, \ldots, y_n & \leftarrow \text{Decoder} \left( x_0, \ldots, x_m \right)
\end{align*}
\]
e.g., LLaMA, GPT
Autoregressive text completion with transformer-based large language models.
Emerging Abilities of LLMs: GPT-2 (2019)

GPT-2 [Radford et al., 2019]:
- Up to 1.5B Parameters
- Transformer decoder-only model, up to 48 layers
- Trained on WebText: 40GB of Internet Data

Language Models are Unsupervised Multitask Learners

Alec Radford * 1  Jeffrey Wu * 1  Rewon Child 1  David Luan 1  Dario Amodei ** 1  Ilya Sutskever ** 1
Emergent Zero-Shot Learning Properties

Context: “Yes, I thought I was going to lose the baby,” “I was scared too,” he stated, sincerity flooding his eyes. “You were?” “Yes, of course. Why do you even ask?” “This baby wasn’t exactly planned for.”
Target sentence: “Do you honestly think that I would want you to have a _____?”
Target word: miscarriage

Context: “Why?” “I would have thought you’d find him rather dry,” she said. “I don’t know about that,” said Gabriel. “He was a great craftsman,” said Heather. “That he was,” said Flannery.
Target sentence: “And Polish, to boot,” said _____.
Target word: Gabriel

Context: Preston had been the last person to wear those chains, and I knew what I’d see and feel if they were slipped onto my skin-the Reaper’s unending hatred of me. I’d felt enough of that emotion already in the amphitheater. I didn’t want to feel anymore. “Don’t put those on me,” I whispered. “Please.”
Target sentence: Sergei looked at me, surprised by my low, raspy please, but he put down the _____.
Target word: chains

Context: They tuned, discussed for a moment, then struck up a lively jig. Everyone joined in, turning the courtyard into an even more chaotic scene, people now dancing in circles, swinging and spinning in circles, everyone making up their own dance steps. I felt my feet tapping, my body wanting to move.
Target sentence: Aside from writing, I’ve always loved _____.
Target word: dancing
Emergent Zero-Shot Learning Properties

GPT-2 defined a new state-of-the-art on challenging LM benchmarks **out of the box**, without any specific fine-tuning:

<table>
<thead>
<tr>
<th></th>
<th>LAMBADA (PPL)</th>
<th>LAMBADA (ACC)</th>
<th>CBT-CN (ACC)</th>
<th>CBT-NE (ACC)</th>
<th>WikiText2 (PPL)</th>
<th>PTB (PPL)</th>
<th>enwik8 (BPB)</th>
<th>text8 (BPC)</th>
<th>WikiText103 (PPL)</th>
<th>1BW (PPL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOTA</td>
<td>99.8</td>
<td>59.23</td>
<td>85.7</td>
<td>82.3</td>
<td>39.14</td>
<td>46.54</td>
<td>0.99</td>
<td>1.08</td>
<td>18.3</td>
<td><strong>21.8</strong></td>
</tr>
<tr>
<td>117M</td>
<td><strong>35.13</strong></td>
<td>45.99</td>
<td><strong>87.65</strong></td>
<td><strong>83.4</strong></td>
<td>29.41</td>
<td>65.85</td>
<td>1.16</td>
<td>1.17</td>
<td>37.50</td>
<td>75.20</td>
</tr>
<tr>
<td>345M</td>
<td><strong>15.60</strong></td>
<td>55.48</td>
<td><strong>92.35</strong></td>
<td>87.1</td>
<td>22.76</td>
<td>47.33</td>
<td>1.01</td>
<td><strong>1.06</strong></td>
<td>26.37</td>
<td>55.72</td>
</tr>
<tr>
<td>762M</td>
<td><strong>10.87</strong></td>
<td>60.12</td>
<td><strong>93.45</strong></td>
<td>88.0</td>
<td>19.93</td>
<td>40.31</td>
<td><strong>0.97</strong></td>
<td><strong>1.02</strong></td>
<td>22.05</td>
<td>44.575</td>
</tr>
<tr>
<td>1542M</td>
<td><strong>8.63</strong></td>
<td><strong>63.24</strong></td>
<td><strong>93.30</strong></td>
<td><strong>89.05</strong></td>
<td><strong>18.34</strong></td>
<td><strong>35.76</strong></td>
<td><strong>0.93</strong></td>
<td><strong>0.98</strong></td>
<td><strong>17.48</strong></td>
<td>42.16</td>
</tr>
</tbody>
</table>

*Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).*
Emergent Zero-Shot Learning Properties

Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.
Emergent Zero-Shot Learning Properties

Zero-shot summarisation on the CNN/DailyMail dataset [See et al., 2017]:

WASHINGTON (CNN) -- Doctors removed five small polyps from President Bush's colon on Saturday, and "none appeared worrisome," a White House spokesman said. The polyps were removed and sent to the National Naval Medical Center in Bethesda, Maryland, for [...] **TL;DR:**

<table>
<thead>
<tr>
<th>Method</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>R-AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom-Up Sum</td>
<td>41.22</td>
<td>18.68</td>
<td>38.34</td>
<td>32.75</td>
</tr>
<tr>
<td>Lede-3</td>
<td>40.38</td>
<td>17.66</td>
<td>36.62</td>
<td>31.55</td>
</tr>
<tr>
<td>Seq2Seq + Attn</td>
<td>31.33</td>
<td>11.81</td>
<td>28.83</td>
<td>23.99</td>
</tr>
<tr>
<td>GPT-2 TL;DR</td>
<td>29.34</td>
<td>8.27</td>
<td>26.58</td>
<td>21.40</td>
</tr>
<tr>
<td>Random-3</td>
<td>28.78</td>
<td>8.63</td>
<td>25.52</td>
<td>20.98</td>
</tr>
<tr>
<td>GPT-2 no hint</td>
<td>21.58</td>
<td>4.03</td>
<td>19.47</td>
<td>15.03</td>
</tr>
</tbody>
</table>

"Too Long, Didn’t Read"
Emerging Abilities of LLMs: GPT-3 (2020)

GPT-3 [Brown et al., 2020]:

• Parameter increase: 1.5B → 175B
• Trained or more data: (40GB → >600GB)

<100 Million
13 y.o. Human

3 Billion
BERT (2018)

30 Billion
RoBERTa (2019)

200 Billion
GPT-3 (2020)
Emerging Abilities of LLMs: GPT-3 (2020)

GPT-3 [Brown et al., 2020]:

• Parameter increase: 1.5B → 175B
• Trained or more data: (40GB → >600GB)
Emergent Few-Shot Learning Abilities

Specify a task by pre-pending examples of the task before your input

Referred to as In-Context Learning — we can teach the model a new task without performing any gradient updates
Emergent Few-Shot Learning Abilities

![Graph showing accuracy over number of examples in context (K) for models with different parameter counts: 175B, 13B, and 1.3B. The graph compares zero-shot, one-shot, and few-shot learning scenarios.]
Emergent Few-Shot Learning Abilities

Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1. Translate English to French: cheese
2. cheese =>

---

In-Context Learning on SuperGLUE

- Human
- Fine-tuned SOTA
- Fine-tuned BERT++
- Fine-tuned BERT Large
- Random Guessing

Number of Examples in Context (K)
Emergent Few-Shot Learning Abilities

One-shot
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

1. Translate English to French: task description
2. sea otter => loutre de mer example
3. cheese => prompt

Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1. Translate English to French: task description
2. cheese => prompt

In-Context Learning on SuperGLUE

- Human
- Fine-tuned SOTA
- Fine-tuned BERT++
- Fine-tuned BERT Large
- Random Guessing

Number of Examples in Context (K)
Emergent Few-Shot Learning Abilities

Few-shot
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1. Translate English to French:
2. sea otter => loutre de mer
3. peppermint => menthe poivrée
4. plush giraffe => girafe peluche
5. cheese =>

One-shot
In addition to the task description and one example of the task.

1. Translate:
2. sea otter => loutre de mer
3. cheese =>

Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1. Translate English to French:
2. cheese =>

In-Context Learning on SuperGLUE
Emergent Few-Shot Learning Abilities

TriviaQA

Fine-tuned SOTA

Zero-Shot
One-Shot
Few-Shot (K=64)
Emergent Few-Shot Learning Abilities

TriviaQA

Table 3.3: Results on three Open-Domain QA tasks. GPT-3 is shown in the few-, one-, and zero-shot settings, as compared to prior SOTA results for closed book and open domain settings. TriviaQA few-shot result is evaluated on the wiki split test server.
Emergent Few-Shot Learning Abilities

\[ L = 2.57 \cdot C^{-0.048} \]
Emergent Zero-Shot (ZS) and Few-Shots (FS) In-Context Learning Abilities:

- Can learn a task **without fine-tuning**
- Results are **highly sensitive** to the prompt being used
- Limited to what you can **fit in the input context**
<table>
<thead>
<tr>
<th>Prompt</th>
<th>Explain the moon landing to a 6 year old in a few sentences.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completion</td>
<td>GPT-3</td>
</tr>
</tbody>
</table>

(From https://openai.com/research/instruction-following)

- Explain the theory of gravity to a 6 year old.
- Explain the theory of relativity to a 6 year old in a few sentences.
- Explain the big bang theory to a 6 year old.
- Explain evolution to a 6 year old.

LLMs are not aligned with user intents [Ouyang et al., 2022]
LLMs ≠ Assistants/Agents

**PROMPT**

*Explain the moon landing to a 6 year old in a few sentences.*

**COMPLETION**

Human

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

LLMs are not aligned with **user intents** [Ouyang et al., 2022]
LLMs ≠ Assistants/Agents

**PROMPT**

*It is unethical for hiring decisions to depend on genders. Therefore, if we were to pick a CEO among Amy and Adam, our pick will be _______

**COMPLETION**

GPT-3

Adam

LLMs are not aligned with **human values** [Zhao et al., 2021]
It is unethical for hiring decisions to depend on genders. Therefore, if we were to pick a CEO among Amy and Adam, our pick will be ________

Human

neither as we don’t know much about their background or experience.

LLMs are not aligned with human values [Zhao et al., 2021]
LLMs $\neq$ Assistants/Agents

But what is alignment exactly? Based on [Askell et al., 2020]:

[...] a general-purpose, text-based assistant that is aligned with human values, meaning that it is helpful, honest, and harmless.

A General Language Assistant as a Laboratory for Alignment

Amanda Askell*  Yuntao Bai*  Anna Chen*  Dawn Drain*  Deep Ganguli*  Tom Henighan†
From LLMs to Assistants/Agents

Emergent Zero-Shot (ZS) and Few-Shots (FS) In-Context Learning Abilities:

✅ Can learn a task without fine-tuning

❌ Results are highly sensitive to the prompt being used

❌ Limited to what you can fit in the input context

Instruction Fine-Tuning
Instruction Fine-Tuning

Idea — aligning LLMs to user interests and human values can be seen as yet another fine-tuning task:

**Step 1:** pre-train on a language modelling objective

\[ x_0 \quad x_1 \quad \cdots \quad \text{END} \]
Instruction Fine-Tuning

Idea — aligning LLMs to user interests and human values can be seen as yet another fine-tuning task:

**Step 1:** pre-train on a language modelling objective

**Step 2:** fine-tune on downstream tasks

...the movie was...
Instruction Fine-Tuning

Collect examples of instruction-output pairs across several tasks and fine-tune a model.

Please answer the following question. What is the boiling point of Nitrogen?

Answer the following question by reasoning step-by-step. The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

The cafeteria had 23 apples. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9.
Instruction Fine-Tuning

Collect examples of instruction-output pairs across several tasks and fine-tune a model.

Please answer the following question. What is the boiling point of Nitrogen?

-320.4°F

Answer the following question by reasoning step-by-step. The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9.

Evaluate on unseen tasks

Q: Can Geoffrey Hinton have a conversation with George Washington? Give the rationale before answering.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is “no”.
Multiple domains/tasks: reading comprehension with an emphasis of various abilities (commonsense, causal, numerical, temporal, multi-hop, reasoning; coreference resolution)
Super-Natural Instructions

Super-Natural Instructions: 1.6K tasks, 3M+ examples

Classification, sequence tagging, rewriting/paraphrasing, translation, question answering..

Many (576+) languages!
Instruction Fine-Tuning — Example

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

[Chung et al., 2022]
Instruction Fine-Tuning — Example

**Model input (Disambiguation QA)**

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

**PaLM 540B output**

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

X (doesn't answer question)

[Chung et al., 2022]
Instruction Fine-Tuning — Example

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

Flan-PaLM 540B output

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C).

[Chung et al., 2022]
Model generation performance is positively correlated with observed tasks and model size.

Number of examples does not have a big influence.

[Wang et al., 2022]
Scaling Instruction Fine-Tuning

**Instruction Fine-Tuning** improves the downstream performance on held-out tasks.

Increasing the number of fine-tuning tasks improves generalisation.

Increasing model scale by an order of magnitude (e.g., 8B → 62B, 62B → 540B) also help a lot.

[Graph showing normalized average on held-out tasks]

[Wang et al., 2022]
From LLMs to Assistants/Agents

Emergent Zero-Shot (ZS) and Few-Shots (FS) In-Context Learning Abilities:
- ✅ Can learn a task without fine-tuning
- ❌ Results are highly sensitive to the prompt being used
- ❌ Limited to what you can fit in the input context

Instruction Fine-Tuning
- ✅ Simple and improves generalisation
- ❌ Ground-truth data for tasks can be expensive to collect
- ❌ Open-ended generation tasks have no single gold answer

Reinforcement Learning with Human Feedback
Reward Model ~ Human Preferences

We are training a model on some task — e.g., to behave as a **personal assistant** for tasks like writing e-mails. For each sample $s$, assume we have a way to obtain a *human reward* for that sample: $R(s) \in \mathbb{R}$
**Reward Model ~ Human Preferences**

We are training a model on some task — e.g., to behave as a personal assistant for tasks like writing e-mails. For each sample $s$, assume we have a way to obtain a human reward for that sample: $R(s) \in \mathbb{R}$

Subject: Immediate Action
Required: Complete Your Cybersecurity Training
Dear Team,
This is your final reminder to complete the mandatory cybersecurity training. Failure to complete the training by the end of this week will result [..]

$$R(s_1) = -2.5$$
Reward Model ~ Human Preferences

We are training a model on some task — e.g., to behave as a personal assistant for tasks like writing e-mails. For each sample $s$, assume we have a way to obtain a human reward for that sample: $R(s) \in \mathbb{R}$

Subject: Immediate Action
Required: Complete Your Cybersecurity Training
Dear Team,
This is your final reminder to complete the mandatory cybersecurity training. Failure to complete the training by the end of this week will result [..]

$R(s_1) = -2.5$

Subject: Friendly Reminder:
Cybersecurity Training Deadline Approaching
Hello Everyone,
Just a friendly reminder that the deadline to complete our mandatory cybersecurity training is fast approaching. Please make sure to complete it by the end of this week. It’s a great opportunity [..]

$R(s_1) = 12.0$
We are training a model on some task — e.g., to behave as a personal assistant for tasks like writing e-mails. For each sample \( s \), assume we have a way to obtain a human reward for that sample: \( R(s) \in \mathbb{R} \).

\[
\begin{align*}
R(s_1) &= -2.5 \\
R(s_1) &= 12.0
\end{align*}
\]

Now we want to maximise the expected reward:

\[
\mathbb{E}_{\hat{s}\sim p(s)} \left[ R(\hat{s}) \right]
\]
Imagine a reward function $R(s)$ for any generation $s$

The reward is **higher** when humans **prefer** the generation

Improving the generation is equivalent to maximising the expected reward:

$$
\mathbb{E}_{\hat{s} \sim p_\theta(s)} [R(\hat{s})]
$$

*Expected reward over the course of sampling from our model*

$p_\theta(s)$ is a model with parameters $\theta$ we aim to optimise

**Reward function encoding human preferences**
Optimising for Human Preferences

Imagine we have a reward function $R(s)$ for any generation $s$

The reward is higher when humans prefer the generation

Improving the generation is equivalent to maximising the expected reward:

$$
\mathbb{E}_{\hat{s} \sim p_\theta(s)} \left[ R(\hat{s}) \right]
$$

We want to:

Find the best generative model $p_\theta$ that maximises the expected reward:

$$
\hat{\theta} = \arg \max_\theta \mathbb{E}_{\hat{s} \sim p_\theta} \left[ R(\hat{s}) \right]
$$

Estimate the reward function encoding human preferences $R(s)$
Optimising the Generative Model $p_\theta$

How do we change our model (LM) parameters $\theta$ to maximise this?

$$\hat{\theta} = \arg \max_\theta \mathbb{E}_{\hat{s} \sim p_\theta} \left[ R (\hat{s}) \right]$$
Optimising the Generative Model $p_\theta$

How do we change our model (LM) parameters $\theta$ to maximise this?

$$\hat{\theta} = \arg \max_{\theta} \mathbb{E}_{\hat{s} \sim p_\theta} \left[ R \left( \hat{s} \right) \right]$$

We can use good old gradient-based optimisation (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R \left( \hat{s} \right) \right] \right]$$

But how can we do that?
Optimising the Generative Model $p_{\theta}$

How do we change our model (LM) parameters $\theta$ to maximise this?

\[ \hat{\theta} = \arg \max_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R(\hat{s}) \right] \]

We can use good old gradient-based optimisation (gradient ascent):

\[
\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R(\hat{s}) \right] \right]
\]

But how can we do that?

We can use policy gradient methods, e.g., REINFORCE [Williams, 1992], also referred to as the score function estimator, to estimate the gradient of the expected reward.
\[ \nabla \theta \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R(\hat{s}) \right] \right] \approx \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla \theta \log p_{\theta}(s_i) \]

Def. of expectation

Gradient distributes over the sum

Monte Carlo estimate
\[ \nabla_\theta \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R (\hat{s}) \right] \right] = \nabla_\theta \left[ \sum_s p_\theta(s) R(s) \right] \quad \text{Def. of expectation} \]
\( \nabla_\theta \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} [R(\hat{s})] \right] = \nabla_\theta \left[ \sum_s p_\theta(s) R(s) \right] \) 

\( = \sum_s R(s) \nabla_\theta p_\theta(s) \)
\[ \nabla_\theta \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R \left( \hat{s} \right) \right] \right] = \nabla_\theta \left[ \sum_s p_{\theta}(s) R(s) \right] \]

Def. of expectation

= \sum_s R(s) \nabla_\theta p_{\theta}(s) \]

Gradient distributes over the sum

= \sum_s R(s) p_{\theta}(s) \nabla_\theta \log p_{\theta}(s) \]

\[ \nabla_\theta \log p_{\theta}(s) = \frac{\nabla_\theta p_{\theta}(s)}{p_{\theta}(s)} \]
\[ \nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right] \right] = \nabla_{\theta} \left[ \sum_{s} p_{\theta}(s) R(s) \right] \]

Def. of expectation

\[ = \sum_{s} R(s) \nabla_{\theta} p_{\theta}(s) \]

Gradient distributes over the sum

\[ = \sum_{s} R(s) p_{\theta}(s) \nabla_{\theta} \log p_{\theta}(s) \]

\[ \nabla_{\theta} \log p_{\theta}(s) = \frac{\nabla_{\theta} p_{\theta}(s)}{p_{\theta}(s)} \]

Def. of expectation

\[ = \mathbb{E}_{p_{\theta}} \left[ R(s) \nabla_{\theta} \log p_{\theta}(s) \right] \]
REINFORCE [Williams, 1992] 101

\[
\nabla_\theta \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R(\hat{s}) \right] \right] = \nabla_\theta \left[ \sum_s p_\theta(s) R(s) \right] \\
= \sum_s R(s) \nabla_\theta p_\theta(s) \quad \text{Gradient distributes over the sum} \\
= \sum_s R(s) p_\theta(s) \nabla_\theta \log p_\theta(s) \quad \nabla_\theta \log p_\theta(s) = \frac{\nabla_\theta p_\theta(s)}{p_\theta(s)} \\
= \mathbb{E}_{p_\theta} \left[ R(s) \nabla_\theta \log p_\theta(s) \right] \\
\approx \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_\theta \log p_\theta(s_i) \quad \text{Monte Carlo estimate} \\
\text{with } s_i \sim p_\theta
\]
\[ \nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[ R \left( \hat{s} \right) \right] \right] = \nabla_{\theta} \left[ \sum_{s} p_{\theta}(s) R(s) \right] \]

Def. of expectation

\[ = \sum_{s} R(s) \nabla_{\theta} p_{\theta}(s) \]

Gradient distributes over the sum

\[ = \sum_{s} R(s) p_{\theta}(s) \nabla_{\theta} \log p_{\theta}(s) \]

\[ \nabla_{\theta} \log p_{\theta}(s) = \frac{\nabla_{\theta} p_{\theta}(s)}{p_{\theta}(s)} \]

Monte Carlo estimate

\[ \approx \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_{\theta} \log p_{\theta}(s_i) \]

with \( s_i \sim p_{\theta} \)
Optimising the Generative Model $p_{\theta}$

How do we change our model (LM) parameters $\theta$ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right]$$

We can use good old \textbf{gradient-based optimisation} (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[ \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_{\theta} \log p_{\theta}(s_i) \right] \quad \text{with } s_i \sim p_{\theta}$$

\text{REINFORCE estimate of}

$$\nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right] \right]$$
Optimising the Generative Model $p_{\theta}$

How do we change our model (LM) parameters $\theta$ to maximise this?

$$\hat{\theta} = \arg \max_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[ \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_{\theta} \log p_{\theta}(s_i) \right] \quad \text{with } s_i \sim p_{\theta}$$

REINFORCE estimate of

$$\nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right] \right]$$

Exercise — what if

$R(s) \in \{0,1\}$?

😊
Optimising the Generative Model $p_{\theta}$

How do we change our model (LM) parameters $\theta$ to maximise this?

$$\hat{\theta} = \arg \max_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[ \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_{\theta} \log p_{\theta}(s_i) \right] \quad \text{with } s_i \sim p_{\theta}$$

**REINFORCE estimate of**

$$\nabla_{\theta} \left[ \mathbb{E}_{\hat{s} \sim p_{\theta}} \left[ R \left( \hat{s} \right) \right] \right]$$

**Exercise — what if**

$R(s) \in \{0,1\}$?

**Note:** this was heavily simplified — in reality, it can require many tricks to work 😬
How do we optimise for human preferences?

Recap — given an arbitrary, non differentiable reward function $R(s)$, we can train our LM to maximise the expected reward! However —

1. Having a human in the loop to assign reward values is costly!
   Solution: collect some human preferences, and train another model (the reward model) to predict new human preferences [Knox et al., 2009]

2. Human judgements tend to be noisy/miscalibrated!
   Solution: rather than asking for direct ratings, ask for pairwise comparisons, which tend to be more reliable [Clark et al., 2018]
How do we optimise for human preferences?

Recap — given an arbitrary, non differentiable reward function $R(s)$, we can train our LM to maximise the expected reward! However —

1. Having a human in the loop to assign reward values is costly!
How do we optimise for human preferences?

Recap — given an arbitrary, non differentiable reward function $R(s)$, we can train our LM to maximise the expected reward! However —

1. Having a human in the loop to assign reward values is costly!

Solution: collect some human preferences, and train another model (the reward model) to predict new human preferences [Knox et al., 2009]
How do we optimise for human preferences?

Recap — given an arbitrary, non differentiable reward function \( R(s) \), we can train our LM to maximise the expected reward! However —

1. Having a human in the loop to assign reward values is costly!
   Solution: collect some human preferences, and train another model (the reward model) to predict new human preferences [Knox et al., 2009]

2. Human judgements tend to be noisy/mis-calibrated!
How do we optimise for human preferences?

Recap — given an arbitrary, non differentiable reward function $R(s)$, we can train our LM to maximise the expected reward! However — 

1. Having a human in the loop to assign reward values is costly!
Solution: collect some human preferences, and train another model (the reward model) to predict new human preferences [Knox et al., 2009]

2. Human judgements tend to be noisy/mis-calibrated!
Solution: rather than asking for direct ratings, ask for pairwise comparisons, which tend to be more reliable [Clark et al., 2018]
RLHF — Putting it all together

For Reinforcement Learning with Human Feedback, we have:

A pre-trained — possibly instruction fine-tuned — LM $p^{LM}(s)$

A reward model $RM(s)$

To optimise our model, we:

Create a copy of the model $p^{RL}_\theta(s)$ with parameters $\theta$

Optimise for the following reward with RL:

$$R(s) = RM(s) - \beta \log \left[ \frac{p^{RL}_\theta(s)}{p^{LM}(s)} \right]$$

Pay a price when $p^{RL}_\theta(s) > p^{LM}(s)$

[Stiennon et al. 2020]


The LAMBADA dataset: Word prediction requiring a broad discourse context, https://aclanthology.org/P16-1144/