LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, Danqi Chen

Motivation

- Instruction tuning is useful when we want to build models that follow instructions.
  Data: Instruction: "Translate to French”. Input: "Hello”. Output: "Bonjour”
  Train to maximize

  log prob("Bonjour” | "Translate to French”, "Hello”)

Motivation

- Instruction tuning is useful when we want to build models that follow instructions.
  Data: Instruction: "Translate to French”. Input: "Hello”. Output: "Bonjour”
  Train to maximize

  log prob("Bonjour” | "Translate to French”, "Hello”)

- Instruction tuning on a diverse set of instructions can deteriorate performance if we care about only a subset of tasks downstream [Wang et al. (2023b)]

Research Question

Given some examples of downstream tasks,
how can we select relevant fine-tuning data
from a large database of instruction data?

Solution: Influence functions!

Influence Functions for SGD

If z is a single training data point, the SGD update for model parameter θ is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^{t})$$

Influence Functions for SGD

If z is a single training data point, the SGD update for model parameter θ is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^{t})$$

If z' is a validation datapoint, then its difference in loss using Taylor expansion is:

$$\ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

Influence Functions for SGD

If z is a single training data point, the SGD update for model parameter θ is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^{t})$$

If z' is a validation datapoint, then its difference in loss using Taylor expansion is:

$$\ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

Plugging in the first equation for change in model params we get:

$$\ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t+1}) - \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) \approx -\eta_t \langle \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^{t}), \nabla \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) \rangle$$

Influence Functions for SGD

If z is a single training data point, the SGD update for model parameter θ is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^{t})$$

If z' is a validation datapoint, then its difference in loss using Taylor expansion is:

$$\ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

Plugging in the first equation for change in model params we get:

$$\ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t+1}) - \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) \approx -\eta_t \langle \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^{t}), \nabla \ell(\boldsymbol{z}'; \boldsymbol{\theta}^{t}) \rangle$$

Influence function:

$$\boxed{\mathrm{Inf}_{\mathrm{SGD}}(\boldsymbol{z}, \boldsymbol{z}') \triangleq \sum_{i=1}^{N} \bar{\eta}_i \langle \nabla \ell(\boldsymbol{z}'; \boldsymbol{\theta}_i), \nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}_i) \rangle}$$

Problems

1. LLMs are trained on batches of data using ADAM not SGD

2. $\nabla \ell(\boldsymbol{z}; \boldsymbol{\theta}^t)$ gradient is computed for a sequence = average gradient of all tokens. Empirically, it is observed that the gradient value is larger for shorter response.

3. We need lots of compute!

P1: Influence function generalization to ADAM

- Reminder: z' is a validation datapoint, then loss reduction is approximated as:

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

P1: Influence function generalization to ADAM

- Reminder: z' is a validation datapoint, then loss reduction is approximated as:

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

- Adam update is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \Gamma(\boldsymbol{z}, \boldsymbol{\theta}^{t})$$

$$\Gamma(\boldsymbol{z}, \boldsymbol{\theta}^{t}) \triangleq \frac{\boldsymbol{m}^{t+1}}{\sqrt{\boldsymbol{v}^{t+1} + \epsilon}}$$

$$\boldsymbol{m}^{t+1} = (\beta_1 \boldsymbol{m}^{t} + (1-\beta_1)\nabla \ell(\boldsymbol{z};\boldsymbol{\theta}^{t}))/(1-\beta_1^{t})$$

$$\boldsymbol{v}^{t+1} = (\beta_2 \boldsymbol{v}^{t} + (1-\beta_2)\nabla \ell(\boldsymbol{z};\boldsymbol{\theta}^{t})^2)/(1-\beta_2^{t})$$

P1: Influence function generalization to ADAM

- Reminder: z' is a validation datapoint, then loss reduction is approximated as:

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

- Adam update is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \Gamma(\boldsymbol{z}, \boldsymbol{\theta}^{t})$$

$$\Gamma(\boldsymbol{z}, \boldsymbol{\theta}^{t}) \triangleq \frac{\boldsymbol{m}^{t+1}}{\sqrt{\boldsymbol{v}^{t+1} + \epsilon}}$$

$$\boldsymbol{m}^{t+1} = (\beta_1 \boldsymbol{m}^{t} + (1-\beta_1)\nabla \ell(\boldsymbol{z};\boldsymbol{\theta}^{t}))/(1-\beta_1^{t})$$

$$\boldsymbol{v}^{t+1} = (\beta_2 \boldsymbol{v}^{t} + (1-\beta_2)\nabla \ell(\boldsymbol{z};\boldsymbol{\theta}^{t})^2)/(1-\beta_2^{t})$$

Warmup!

P1: Influence function generalization to ADAM

- Reminder: z' is a validation datapoint, then loss reduction is approximated as:

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} \rangle$$

- Adam update is:

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^{t} = -\eta_t \Gamma(\boldsymbol{z}, \boldsymbol{\theta}^{t})$$

$$\Gamma(\boldsymbol{z}, \boldsymbol{\theta}^{t}) \triangleq \frac{\boldsymbol{m}^{t+1}}{\sqrt{\boldsymbol{v}^{t+1} + \epsilon}}$$

$$\boldsymbol{m}^{t+1} = (\beta_1 \boldsymbol{m}^{t} + (1-\beta_1)\nabla \ell(\boldsymbol{z};\boldsymbol{\theta}^{t}))/(1-\beta_1^{t})$$

$$\boldsymbol{v}^{t+1} = (\beta_2 \boldsymbol{v}^{t} + (1-\beta_2)\nabla \ell(\boldsymbol{z};\boldsymbol{\theta}^{t})^2)/(1-\beta_2^{t})$$

Warmup!

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}_i), \Gamma(\boldsymbol{z}, \boldsymbol{\theta}_i) \rangle$$

P2: Adjusting instruction lengths

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}_i), \Gamma(\boldsymbol{z},\boldsymbol{\theta}_i) \rangle$$

P2: Adjusting instruction lengths

$$\ell(\boldsymbol{z}';\boldsymbol{\theta}^{t+1}) \approx \ell(\boldsymbol{z}';\boldsymbol{\theta}^{t}) + \langle \nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}_i), \Gamma(\boldsymbol{z},\boldsymbol{\theta}_i) \rangle$$

Normalize by replacing dot product with cosine similarity

$$\mathrm{Inf}_{\mathrm{Adam}}(\boldsymbol{z},\boldsymbol{z}') \triangleq \sum_{i=1}^{N} \bar{\eta}_i \cos(\nabla \ell(\boldsymbol{z}';\boldsymbol{\theta}_i), \Gamma(\boldsymbol{z},\boldsymbol{\theta}_i))$$

P3: Compute reduction

1. Update LLM parameters using LoRA, i.e. decompose model parameters to a product of low-ranked matrices and get the ADAM update.

$$\hat{\Gamma}(\cdot, \boldsymbol{\theta})$$

P3: Compute reduction

1. Update LLM parameters using LoRA, i.e. decompose model parameters to a product of low-ranked matrices and get the ADAM update.

$$\hat{\Gamma}(\cdot, \boldsymbol{\theta})$$

2. Reduce the dimensionality of this ADAM update by randomly projecting it on to a low dimensional space:

$$\tilde{\Gamma}(\boldsymbol{z}, \cdot) = \Pi^{\top} \hat{\Gamma}(\boldsymbol{z}, \cdot)$$

$$\Pi_{ij} \sim \mathcal{U}(\{-1, 1\}$$

Johnson-Lindenstrauss Lemma; dot-product is preserved (???)

P3: Compute reduction

- Do the same for validation data but with regular gradient. Take average across all data for a given task.

$$\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i) = \frac{1}{|\mathcal{D}_{\text{val}}^{(j)}|}\sum_{\boldsymbol{z}'\in\mathcal{D}_{\text{val}}^{(j)}}\tilde{\nabla}\ell(\boldsymbol{z}';\boldsymbol{\theta}_i)$$

P3: Compute reduction

- Do the same for validation data but with regular gradient. Take average across all data for a given task.

$$\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i) = \frac{1}{|\mathcal{D}_{\text{val}}^{(j)}|}\sum_{\boldsymbol{z}'\in\mathcal{D}_{\text{val}}^{(j)}}\tilde{\nabla}\ell(\boldsymbol{z}';\boldsymbol{\theta}_i)$$

- Finally, influence is given by:

$$\text{Inf}_{\text{Adam}}(\boldsymbol{z},\mathcal{D}_{\text{val}}^{(j)}) = \sum_{i=1}^{N}\bar{\eta}_i\frac{\langle\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i),\tilde{\Gamma}(\boldsymbol{z},\boldsymbol{\theta}_i)\rangle}{\|\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i)\|\|\tilde{\Gamma}(\boldsymbol{z},\boldsymbol{\theta}_i)\|}$$

P3: Compute reduction

- Do the same for validation data but with regular gradient. Take average across all data for a given task.

$$\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i) = \frac{1}{|\mathcal{D}_{\text{val}}^{(j)}|}\sum_{\boldsymbol{z}'\in\mathcal{D}_{\text{val}}^{(j)}}\tilde{\nabla}\ell(\boldsymbol{z}';\boldsymbol{\theta}_i)$$

- Finally, influence is given by:

$$\text{Inf}_{\text{Adam}}(\boldsymbol{z},\mathcal{D}_{\text{val}}^{(j)}) = \sum_{i=1}^{N}\bar{\eta}_i\frac{\langle\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i),\tilde{\Gamma}(\boldsymbol{z},\boldsymbol{\theta}_i)\rangle}{\|\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i)\|\|\tilde{\Gamma}(\boldsymbol{z},\boldsymbol{\theta}_i)\|}$$

- Rank based on: $\max_j \text{Inf}_{\text{Adam}}(\boldsymbol{z},\mathcal{D}_{\text{val}}^{(j)})$

Experiments - Dataset

- MMLU: MCQ questions in CS, elementary math, US history, law etc.
- TYDIQA: multi-lingual Q and A with question and passage. Task is extracting answer from the passage.
- BBH: challenging tasks from BIG-Bench selected to evaluate reasoning capabilities.

Dataset	# Shot	# Tasks	$\lvert\mathcal{D}_{\text{val}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$	Answer Type
MMLU	5	57	285	18,721	Letter options
TYDIQA	1	9	9	1,713	Span
BBH	3	23	69	920	COT and answer

Experiments

- Models: LLAMA-2-7B, LLAMA-2-13B, MISTRAL-7B

- Transfer learning (**LESS-T**): compute influence using a smaller model (LLAMA-2-7B) but train selected data on the bigger models (any 3).

Baselines

- Random selection: self-explanatory.
- BM25: featurize example based on word frequency statistics and select top k most similar training data point.
- DSIR: use n-gram features to also rank training data.
- RDS: use model’s hidden representation as features.

Results

	MMLU				TyDiQA				BBH			
Data percentage	**Full** (100%)	**Rand.** (5%)	**LESS-T** (5%)	**LESS** (5%)	**Full** (100%)	**Rand.** (5%)	**LESS-T** (5%)	**LESS** (5%)	**Full** (100%)	**Rand.** (5%)	**LESS-T** (5%)	**LESS** (5%)
Llama-2-7B	*51.6*	46.5 (0.5)	-	**50.2** (0.5)	*54.0*	52.7 (0.4)	-	**56.2** (0.7)	*43.2*	38.9 (0.5)	-	**41.5** (0.6)
Llama-2-13B	*54.5*	53.4 (0.1)	**54.6** (0.3)	54.0 (0.7)	*54.3*	53.0 (1.3)	**57.5** (0.8)	54.6 (0.3)	*50.8*	47.0 (1.6)	49.9 (0.5)	**50.6** (0.6)
Mistral-7B	*60.4*	60.0 (0.1)	60.6 (0.3)	**61.8** (0.4)	*57.7*	56.9 (0.2)	**61.7** (1.7)	60.3 (2.4)	*53.0*	54.5 (0.1)	**56.0** (0.8)	**56.0** (1.0)

	Rand.	BM25	DSIR	RDS	LESS	Δ
MMLU	46.5 (0.5)	47.6	46.1 (0.3)	45.0 (1.0)	**50.2** (0.5)	↑2.6
TyDiQA	52.7 (0.4)	52.7	44.5 (1.7)	46.8 (1.3)	**56.2** (0.7)	↑3.5
BBH	38.9 (0.5)	39.8	36.8 (0.1)	36.7 (1.3)	**41.5** (0.6)	↑1.7

***** More experiments in paper for compute time, warm-up, LoRA efficiency, projection dimension selection, qualitative analysis, etc.**