Optimization in Machine Learning

Topic: Model Uncertainty Based Reweighting for Robust Convergence

Team Members :-
Snegha A (23M2160)
Hemanth Kotaprolu (23M2164)

Course Project Presentation

TA MENTOR: Prateek Chanda
INSTRUCTOR: Prof. Ganesh Ramakrishnan

ARXIV SUBMISSION UPDATE

Model Uncertainty Based Reweighting for Robust Convergence

Snegha A
CMInDS, IIT Bombay
snegha.a@iitb.ac.in

Hemanth Kotaprolu
CMInDS, IIT Bombay
hemanth.kotaprolu@iitb.ac.in

Prateek Chanda
CSE, IIT Bombay
prateek.chanda@iitb.ac.in

Ganesh Ramakrishnan
CSE, IIT Bombay
ganramkr@iitb.ac.in

Abstract

Large Language Models have achieved remarkable success across diverse NLP benchmarks, yet their performance remains sensitive to the quality of pretraining data. Web-scale corpora often contain noisy or low-quality samples that can impede convergence and degrade final model accuracy. In this work, we propose an **Meta-EM Reweighting Algorithm** that mitigates this issue by adjusting the importance of each training example based on its estimated uncertainty. Our method employs a small proxy model to compute batch-level loss statistics (mean and variance) for each training sequence, which are then clustered into distinct categories i.e., "easy", "hard", and "noisy" using a Gaussian Mixture Model. These category assignments inform sample-specific weights that are applied during the pretraining of a larger auto-regressive LLM. Our method provides a principled mechanism for quality-aware training through: (1) predictive uncertainty estimation, (2) probabilistic sample categorization, and (3) dynamic weight modulation. Experimental results in a 5-shot setting across five benchmarks demonstrate consistent improvements over the baseline. Notably, the weighted model achieves **+0.30 on LogiQA, +0.40 on PiQA, and +0.30 on ARC-Challenge**, validating the efficacy of uncertainty-based reweighting in enhancing LLM training. Code will be made publicly available.


```
Dear Hemanth Kotaprolu,

Thank you for submitting your work to arXiv.

Your submission has been received and is under consideration. The temporary submission number is:
submit/6407612.

As with all submissions, this work will go through technical and moderation checks. You will be
contacted by arXiv when the work is announced or if any issues are identified.

Our goal is to screen and announce papers as quickly as possible while ensuring that papers meet
long-term archival standards. Generally, this process takes two business days, with announcements
occurring at 20:00 ET, Sunday through Thursday.

You can make changes and view the current status of the submission from your user dashboard:
https://arxiv.org/user/.

Below is a copy of the submission information.

Regards,
arXiv Support
```

OUTLINE

- Motivation
- Preliminaries
  - Expectation Maximization
  - Gaussian Mixture Models
- Approach
- Experimental Setup
- Result and Analysis
- Conclusion

MOTIVATION

LLM pretraining predominantly involve two phases:

- ❏ Heavy data curation.
- ❏ Training with uniform sampling on the constructed corpus.

The increased emphasis on **data quantity has made it challenging to assess the importance** of each data sample during training process.

We leverage **Model uncertainty measures** to guide training data reweighting.
It prioritizes or de-emphasizes certain training examples based on how uncertain the model is about them.

Uncertainty occurs in two situation:

- ❏ Due to lack of knowledge about certain data patterns — **Prioritizes**
- ❏ Due to inherent randomness or noise in the data — **De-emphasizes**

Common way of measuring this uncertainty is through **loss.**

PRELIMINARIES

GAUSSIAN MIXTURE MODELS

- Gaussian Mixture Model (GMM) is a probabilistic model that assumes data points are generated from a mixture of several Gaussian distributions, each with its own mean and variance.

- GMM is used when we have complex data distributions i.e, when data clusters are not well-separated or not linearly separable.

$$p(x) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

K: Number of Gaussian Components

$\boldsymbol{\pi_k}$: Weight (mixing coefficient) for the k-th Gaussian (sums to 1)

$\boldsymbol{\mu_k}$: Mean of k-th gaussian

$\boldsymbol{\Sigma_k}$: Covariance of k-th gaussian

- K-Means is a special case of Gaussian, with covariance matrix as Identity.

EXPECTATION MAXIMIZATION

- Expectation Maximization (EM) is an iterative algorithm used to find the parameters of a model with latent (hidden) variables
  - Like GMMs, where we don't know which point belongs to which Gaussian, mean and variance of each gaussian component

- Two repeating steps:
  - **E-Step (Expectation)**: For each data point x_i, compute the probability that it belongs to each cluster *k* using

$$\gamma_{ik} = \frac{\pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \cdot \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}$$

  - **M-step (Maximization)**: Update the parameters using the responsibilities:

$$\mu_k = \frac{\sum_i \gamma_{ik} x_i}{\sum_i \gamma_{ik}} \qquad \Sigma_k = \frac{\sum_i \gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \gamma_{ik}} \qquad \pi_k = \frac{1}{N} \sum_i \gamma_{ik}$$

  - Repeat step E and M until convergence (log-likelihood stops improving).

APPROACH

OVERVIEW & ALGORITHM

Algorithm 1 Meta-EM Reweighting Algorithm for Noisy/Hard Sample Selection

Input:

- Training data $\mathcal{D}_{\text{train}} = \{x_i\}_{i=1}^{M}$
- Proxy model $\hat{f}$ for estimating tokenwise losses
- Main model $f(x;\theta)$
- Validation set $\mathcal{D}_{\text{val}}$
- Number of clusters K
- Learning rates η (for model), η_ϕ (for meta)
- Max sample weight $\omega_{\max} = 2/M$

Initialize:

- Model parameters θ_0
- Cluster parameters $\phi = \{\boldsymbol{c}_k, \lambda_k, \pi_k\}_{k=1}^{K}$

for each training step $t = 0$ to $T-1$ **do**
  Sample batch $\mathcal{B}_t \subset \mathcal{D}_{\text{train}}$
  for each sample $x_i \in \mathcal{B}_t$ **do**
    Compute tokenwise losses $\{\ell_{ij}\}_{j=1}^{L_i} \leftarrow \hat{f}(x_i)$
    Compute proxy statistics:

$$\boldsymbol{s}_i = (\mu_i, \sigma_i^2), \quad \mu_i = \frac{1}{L_i}\sum_j \ell_{ij}, \quad \sigma_i^2 = \frac{1}{L_i}\sum_j (\ell_{ij} - \mu_i)^2$$

  end for
  ▷ **E-step: Cluster Responsibility**
  for each $x_i \in \mathcal{B}_t$ **do**
    $\gamma_{ik} \leftarrow \frac{\pi_k \cdot \exp(-\lambda_k \|\boldsymbol{s}_i - \boldsymbol{c}_k\|^2)}{\sum_{j=1}^{K} \pi_j \cdot \exp(-\lambda_j \|\boldsymbol{s}_i - \boldsymbol{c}_j\|^2)}$
  end for
  ▷ **Weighting**
  for each $x_i \in \mathcal{B}_t$ **do**
    $w_i \leftarrow \sum_{k=1}^{K} \gamma_{ik} \cdot \exp(-\lambda_k \|\boldsymbol{s}_i - \boldsymbol{c}_k\|^2)$
    $w_i \leftarrow \min(w_i, \omega_{\max})$ ▷ Enforce Theorem 1 constraint
  end for
  ▷ **Model Update (Inner Loop)**
  $\theta_{t+1} \leftarrow \theta_t - \eta \cdot \sum_{x_i \in \mathcal{B}_t} w_i \cdot \nabla_\theta f(x_i;\theta_t)$
  ▷ **Meta Update (Outer Loop)**
  Sample validation minibatch z_t^{val}
  Compute validation improvement objective:

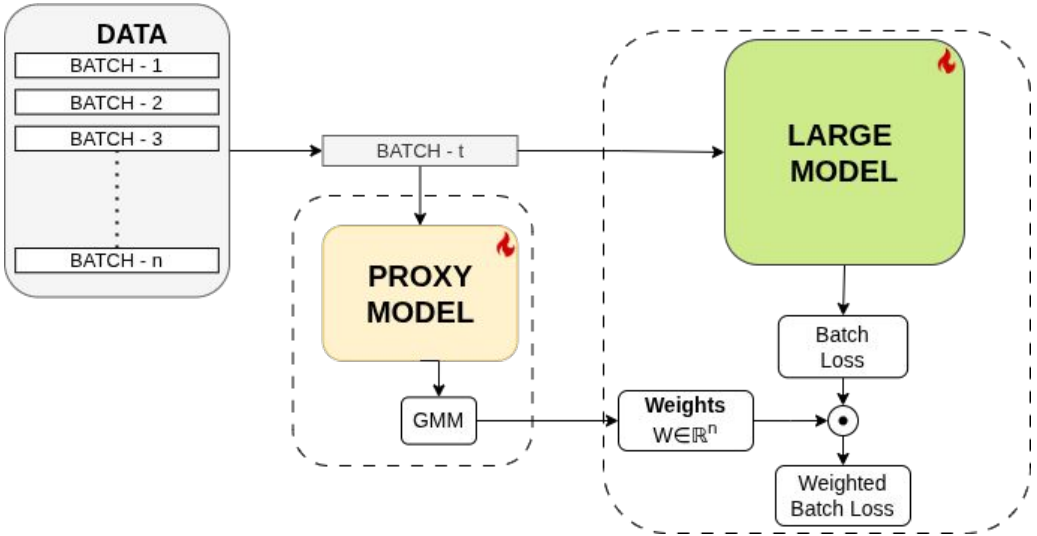
$$U_t(\phi) = \ell(z_t^{\text{val}}, \theta_t) - \ell(z_t^{\text{val}}, \theta_{t+1}(\phi))$$

  Compute gradient $\nabla_\phi U_t(\phi)$ via unrolled backpropagation
  Update cluster parameters:

$$\phi \leftarrow \phi + \eta_\phi \cdot \nabla_\phi U_t(\phi)$$

end for

Output: Trained model parameters θ_T, learned reweighting parameters ϕ



ALGORITHM MOTIVATION

3.3 Small and Large Model for Reweighting

Pretraining of Large Language Models (LLMs) often requires huge computational resources. The training dynamics of different training samples are consistent across differently sized models Yang et al. (2024b); Xia et al. (2023). This is verified in the theorem below.

Theorem *If examples i and j have similar loss trajectories on the proxy model, i.e., $\|\boldsymbol{L}_i^{proxy} - \boldsymbol{L}_j^{proxy}\| \leq \epsilon$, and their loss trajectories on the proxy and target model is similar, i.e., $\|\boldsymbol{L}_p^{proxy} - \boldsymbol{L}_p^{target}\| \leq \delta$ for $p \in \{i, j\}$, then i and j have similar gradients throughout training the target model:*

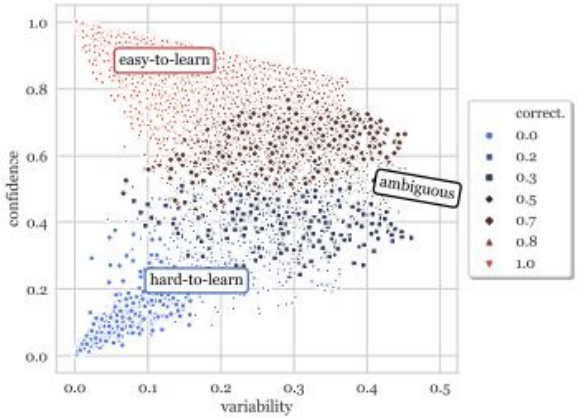
$$\|\nabla \mathcal{L}_i^{\text{target}}(\boldsymbol{\theta}) - \nabla \mathcal{L}_j^{\text{target}}(\boldsymbol{\theta})\| \leq \frac{2\epsilon' + 2CD^2}{d} = \Delta.$$

where $\epsilon' = \epsilon + 2\delta$ and $\|\boldsymbol{\theta}\| \leq D$ for all t.

This theorem shows that if two samples have similar loss trajectories on a small proxy model and align with the target model, their gradients during target model training will also be similar, enabling efficient sample reweighting.

Proxy models can reliably predict training dynamics of larger models, reducing computational costs for sample selection !!!

ALGORITHM MOTIVATION



Data Cartography

The behavior of the model on individual instances during training (training dynamics) for building data maps. This yields two intuitive measures for each example—the model's confidence in the true class, and the variability of this confidence.

- Ambiguous regions with respect to the model, which contribute the most towards out-of-distribution generalization.
- The most populous regions in the data are easy to learn for the model, and play an important role in model optimization

WEIGHT CALCULATION

- GMM is use to cluster training samples into three groups based on their sample-level mean and loss. These clusters are hypothesized to correspond to *easy*, *hard*, and *noisy* samples.
- However, GMM does not explicitly label each cluster. To infer these categories, we analyze the cluster centers in terms of their global mean, variance, and relative positioning. Based on this analysis, we assign a category label—easy, hard, or noisy—to each sample.

Once categorized, we assign weights to the samples. The weighting depends on two factors:

- The distance between a sample's (mean,variance) and the global batch (mean,variance).
- The angle between the sample (mean,variance) vector and the global (mean,variance) vector.

We combine these using a simple heuristic:

- For ***hard*** samples: **weight = exp(angle) + distance**
- For ***noisy*** samples: **weight = exp(angle) - distance**
- *Easy* samples are not weighted (i.e., weight = 1).

Finally, all weights are normalized within the batch and used to modulate the large model's loss, thereby guiding the gradient updates.

EXPERIMENTAL SETUP

SETUP & EVALUATION

Training Configurations:

- Main Model: GPT-style autoregressive LM with 355M parameters
- Proxy Model: Smaller model with 124M parameters for uncertainty estimation.
- Dataset: OpenWebText (high-quality English web content) : 8M documents

Evaluation:

We evaluate the pretrained models in a 5-shot setting.

These benchmarks span a diverse set of reasoning challenges:

- **Logical Reasoning** → LOGIQA
- **Physical Commonsense** → PIQA
- **Scientific QA** → SCIQ
- **Plausibility Reasoning** → HELLASWAG
- **Deductive Reasoning** → ARC-CHALLENGE

RESULTS

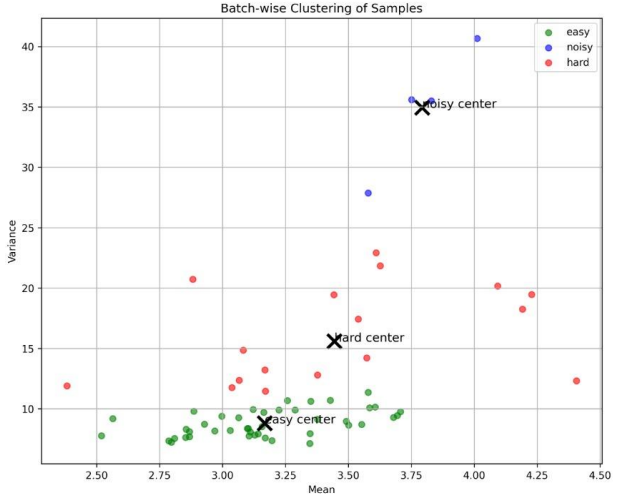
RESULT

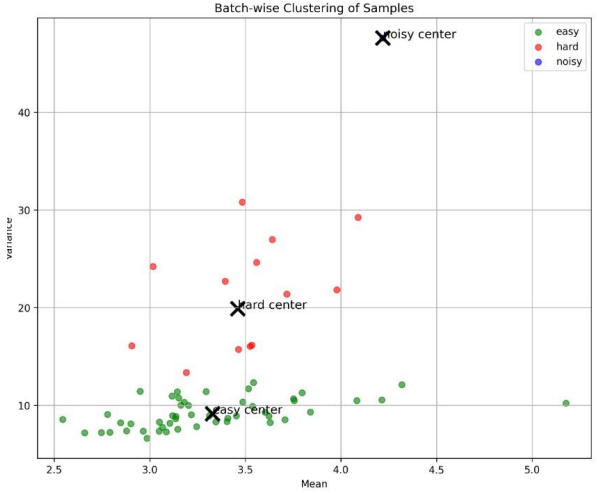
- **Reweighting improves or maintains performance across most benchmarks**, with notable gains on LogiQA and PIQA—highlighting its strength in logical and commonsense reasoning tasks.
- Moderate improvement on ARC-Challenge and stable performance on HellaSwag indicate the method's ability to generalize to knowledge-intensive and narrative reasoning tasks.
- A performance drop on SciQ suggests that reweighting may suppress simple yet crucial patterns, indicating the need for careful tuning on fact-based datasets.

Model	Benchmark	Baseline	Weighted Model
GPT-2 Medium	SciQ	23.30	22.70
	LogiQA	24.30	**24.60**
	PiQA	50.30	**50.70**
	HellaSwag	27.60	27.20
	ARC-Challenge	27.20	**27.50**

Table 1: Performance comparison of GPT-2 Medium on various benchmarks in 5 shot setting

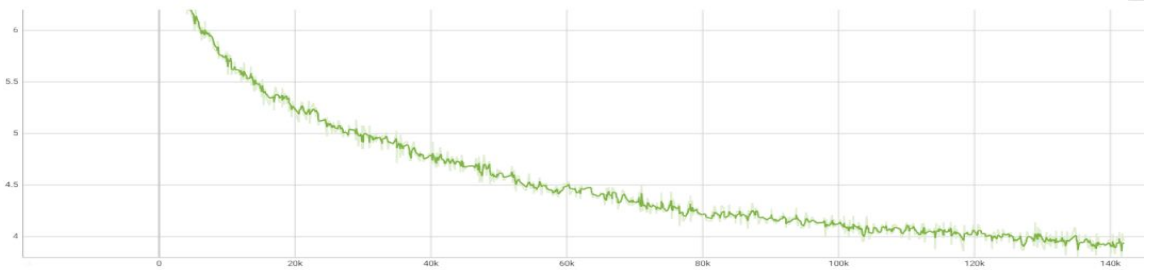
ANALYSIS





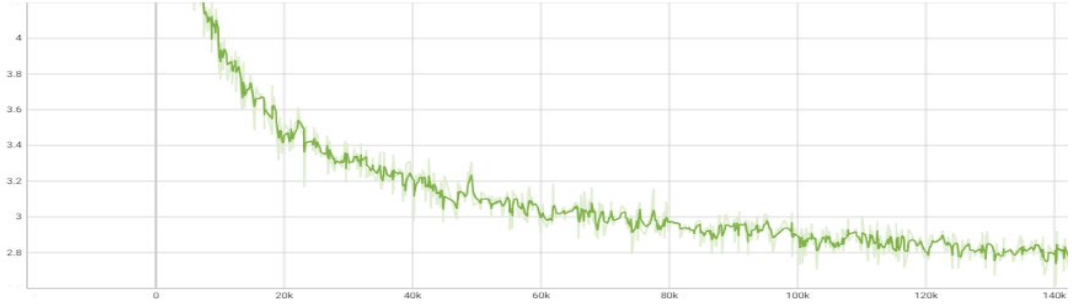
GMM clustering of a Random Batch

ANALYSIS



Baseline Loss curve

Weighted Model Loss curve



CONCLUSION

CONCLUSION

- This work demonstrates that using predictive uncertainty to reweight training sequences enhances performance in few-shot reasoning tasks.
- By clustering loss statistics into easy, hard, and noisy categories, the method emphasizes informative examples and downplays noisy outliers.
- Empirical improvements on LogiQA, PiQA, and ARC-Challenge highlight its effectiveness for logical and commonsense inference.
- Underperformance on SciQ and HellaSwag suggests that reweighting strength may need task-specific calibration for optimal results.

REFERENCE

- Sow, Daouda, et al. "Dynamic Loss-Based Sample Reweighting for Improved Large Language Model Pretraining." *arXiv preprint arXiv:2502.06733* (2025).
- Kumar, Ramnath, et al. "Stochastic re-weighted gradient descent via distributionally robust optimization." *arXiv preprint arXiv:2306.09222* (2023).
- Thakkar, Megh, et al. "Self-influence guided data reweighting for language model pre-training." *arXiv preprint arXiv:2311.00913* (2023).
- Swayamdipta, Swabha, et al. "Dataset cartography: Mapping and diagnosing datasets with training dynamics." *arXiv preprint arXiv:2009.10795* (2020).
- Yang, Y., Mishra, S., Chiang, J. N., and Mirzasoleiman, B. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024b. URL https://openreview.net/forum?id=K9IGIMQpif.
- Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N.A. and Choi, Y., 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*.

THANK YOU!