Advanced Training Metrics and Mathematical Formulations

Reno-Vans Ensemble System

March 3, 2025

1 Introduction

This document presents the core mathematical formulations and metrics used in our ensemble training pipeline. Key topics include:

- Token-level and sequence-level entropy
- Logistic regression ensemble for quality scoring
- FAISS-based KNN retrieval for similar example search
- Cross-model alignment loss for latent space fusion
- Knowledge distillation loss for training a student model

2 Entropy Calculations

For a probability distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, the token-level entropy is defined as:

$$H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log p_i. \tag{1}$$

The sequence-level (mean) entropy over n tokens is given by:

$$H_{\text{seq}} = \frac{1}{n}\sum_{i=1}^{n} H(p_i). \tag{2}$$

3 Logistic Regression Ensemble

Our logistic regression ensemble combines features from multiple models. Given a feature vector $\mathbf{x} \in \mathbb{R}^7$, the prediction is:

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{x} + b), \tag{3}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function. The features include:

1. Primary confidence: $1 - H_{\text{seq}}$ from the primary model.
2. Secondary confidence: Derived from ensemble disagreement.
3. Evaluator confidence: $1 - H_{\text{seq}}$ from the evaluator model.

4. Raw primary sequence entropy.
5. Ensemble disagreement score.
6. KNN similarity score.
7. Example quality metadata.

4 KNN Retrieval with FAISS

We use FAISS to build an index for rapid retrieval of similar code examples. For two normalized embeddings $\mathbf{u}$ and $\mathbf{v}$, the cosine similarity is:

$$\text{similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}. \tag{4}$$

A higher similarity indicates a closer match between the query and stored examples.

5 Cross-Model Alignment Loss

To ensure consistent latent representations across models, we project their final hidden states into a common space and compute cosine similarities. The alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = \frac{1}{3}\left[(1 - \cos(\mathbf{z}_1, \mathbf{z}_2)) + (1 - \cos(\mathbf{z}_1, \mathbf{z}_3)) + (1 - \cos(\mathbf{z}_2, \mathbf{z}_3))\right], \tag{5}$$

where $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ are the projected representations from the primary, secondary, and evaluator models respectively.

6 Knowledge Distillation

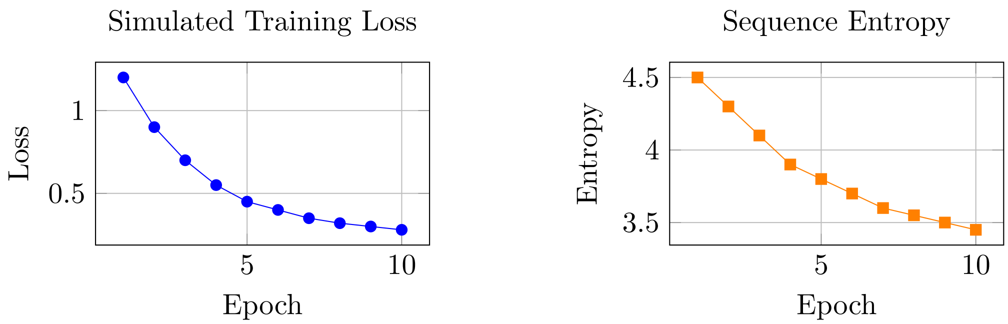
In our advanced knowledge distillation, a student MLP is trained to mimic the ensemble projection. The distillation loss is given by:

$$\mathcal{L}_{\text{distill}} = \|f_{\text{student}}(\mathbf{h}) - f_{\text{target}}(\mathbf{h})\|^2, \tag{6}$$

where $\mathbf{h}$ represents the hidden state from the primary model, and f_{student} and f_{target} are the student and target projection functions, respectively.

7 Visualization of Training Metrics

Below is an example plot of simulated training loss and sequence entropy over epochs.



(a) Training Loss

(b) Sequence Entropy

Figure 1: Simulated Training Metrics over 10 Epochs

8 Conclusion

This document has provided a detailed mathematical formulation of the key components of our ensemble training pipeline. By leveraging these formulations, we can monitor, analyze, and improve the system's performance continuously.