Mathematical Formulations and Proofs for mBART Representation and SAE Dimensionality Reduction

team 20

December 8, 2024

1 Introduction

This document provides mathematical formulas and derivations related to extracting and processing hidden representations from an mBART model, applying Stacked Autoencoders (SAEs) for dimensionality reduction, and analyzing the resulting distributions and loss functions.

2 mBART Encoder and Decoder Attention Layers

The mBART model follows the Transformer architecture, using multi-head self-attention in the encoder and cross-attention in the decoder.

2.1 Encoder Self-Attention

Given an input sequence of tokens, let the embeddings be:

$$X \in \mathbb{R}^{T \times d},$$

where T is the sequence length and $d = 1024$ is the model dimension.

For a single attention head:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V,$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_h}$ and $d_h = d/H$ for H heads.

The attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right)V.$$

For multi-head attention:

$$\text{MultiHead}(X) = [\text{head}_1; \text{head}_2; \ldots; \text{head}_H]W^O,$$

where $W^O \in \mathbb{R}^{(Hd_h) \times d}$.

Adding the residual connection and layer normalization:

$$\hat{X} = \text{LayerNorm}(X + \text{MultiHead}(X)).$$

Then a position-wise feed-forward network (FFN) is applied:

$$X_{\text{out}} = \text{LayerNorm}(\hat{X} + \text{FFN}(\hat{X})).$$

The output of the final encoder layer $X_{\text{out}} \in \mathbb{R}^{T \times d}$ captures the contextualized embeddings.

Applying mean pooling across the time dimension:

$$h_{\text{enc}} = \frac{1}{T} \sum_{t=1}^{T} X_{\text{out}}^{(t)} \in \mathbb{R}^d.$$

2.2 Decoder Cross-Attention

In the decoder, given the encoder output X_{enc} and the decoder states X_{dec}, cross-attention is computed similarly:

$$Q_{\text{dec}} = X_{\text{dec}} W_{\text{dec}}^{Q}, \quad K_{\text{enc}} = X_{\text{enc}} W_{\text{enc}}^{K}, \quad V_{\text{enc}} = X_{\text{enc}} W_{\text{enc}}^{V}.$$

$$\text{Attention}(Q_{\text{dec}}, K_{\text{enc}}, V_{\text{enc}}) = \text{softmax}\left(\frac{Q_{\text{dec}} K_{\text{enc}}^{\top}}{\sqrt{d_h}}\right) V_{\text{enc}}.$$

After processing through the decoder layers, we similarly obtain:

$$h_{\text{dec}} = \frac{1}{T'} \sum_{t=1}^{T'} X_{\text{dec out}}^{(t)} \in \mathbb{R}^d.$$

3 Stacked Autoencoder (SAE) for Dimensionality Reduction

We use an SAE to reduce the dimension from $d = 1024$ to $d' = 128$.

3.1 SAE Encoder and Decoder

The SAE consists of an encoder and a decoder. The encoder maps:

$$f_{\text{enc}} : \mathbb{R}^{1024} \to \mathbb{R}^{128}.$$

A single layer can be represented as:

$$z = \sigma(Wx + b),$$

where $x \in \mathbb{R}^{1024}, W \in \mathbb{R}^{128\times1024}, b \in \mathbb{R}^{128}$, and $\sigma(\cdot)$ is an activation function (e.g., ReLU).

The decoder reconstructs back to the original dimension:

$$\hat{x} = \sigma'(W'z + b'),$$

where $W' \in \mathbb{R}^{1024\times128}, b' \in \mathbb{R}^{1024}$, and $\sigma'(\cdot)$ may be chosen depending on the output domain.

3.2 Variance Retention

Under the assumption that the input distribution is isotropic Gaussian, each of the $D = 1024$ dimensions contributes equally to total variance. Reducing to $d' = 128$ dimensions retains:

$$\frac{128}{1024} = 0.125 = 12.5\%$$

of the variance.

4 Sparsity and Variance Threshold Inequalities

4.1 Sparsity in the SAE

A sparse autoencoder aims to maintain low average activation of hidden neurons. Let $\hat{\rho}_j$ be the average activation of the j-th neuron:

$$\hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N} h_j^{(i)},$$

where $h_j^{(i)}$ is the activation of neuron j for the i-th sample. Given a desired sparsity ρ, we impose:

$$|\hat{\rho}_j - \rho| \leq \epsilon \quad \text{or more strictly} \quad \hat{\rho}_j \leq \rho.$$

4.2 Variance Threshold

If $\lambda_1, \lambda_2, \ldots, \lambda_D$ are eigenvalues (variances along principal components), and we want to retain a fraction $\alpha \in (0, 1)$ of the variance, we need:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{D} \lambda_j} \geq \alpha.$$

For an isotropic Gaussian where all dimensions contribute equally, this reduces to:

$$\frac{k}{D} \geq \alpha \implies k \geq \alpha D.$$

5 Distribution and Loss Functions in Reduced Space

5.1 Distribution in 128-D Space

After dimension reduction:

$$x \in \mathbb{R}^{128}, \quad x \sim \mathcal{N}(\mu, \Sigma),$$

with $\mu \in \mathbb{R}^{128}$ and $\Sigma \in \mathbb{R}^{128\times128}$.

5.2 Loss Functions

Reconstruction Loss:

$$\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|_2^2.$$

Sparsity Penalty (KL-Divergence):

$$\sum_j \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}.$$

Supervised Loss: For a ground truth y and prediction $\hat{y}$:

$$\mathcal{L}_{\text{sup}} = \|y - \hat{y}\|_2^2 \quad \text{or for classification:} \quad \mathcal{L}_{\text{CE}} = -\sum_c y_c \log(\hat{y}_c).$$

Combined Loss:

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \beta \sum_j \text{KL}(\rho \| \hat{\rho}_j) + \gamma \|y - \hat{y}\|_2^2,$$

where $\beta, \gamma \geq 0$ are weighting coefficients.

6 Conclusion

These formulas and inequalities can be incorporated into academic manuscripts or presentation slides to rigorously support the methodology of analyzing mBART's hidden representations, applying SAEs for dimensionality reduction, and controlling sparsity and variance retention.