Intelligent Data Labeling via Neural Network-Driven Active Learning for Enhanced Model Efficiency

Aditya Vijendra Girase - s240125[1], Ivor Baricevic - s232467[1], Mads Albert Alkjærsig - s203372[2], and Samyak Jain - s232883[1]

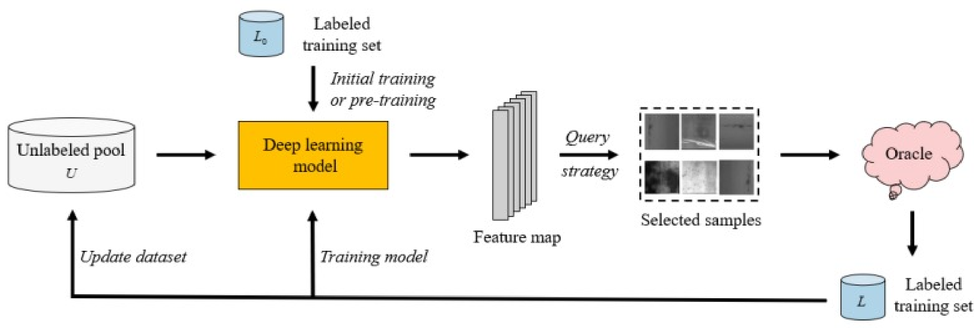
1 Autonomous Systems, Technical University of Denmark; 2 Acoustics, Technical University of Denmark

Introduction

Deep Learning (DL) excels at extracting data features but relies heavily on large labeled datasets, which are costly and time-consuming to create [1]. Active Learning (AL) addresses this by identifying the most informative samples for labeling, minimizing costs while maintaining high accuracy. The combination of DL and AL, referred to as Deep Active Learning (DeepAL), enables the efficient training of neural networks in data-scarce domains like medical imaging, robotics, and speech recognition.

Objectives:

- Optimize neural network training by minimizing labeled data requirements [1].
- Implement active learning strategies to focus on high-information samples for labeling [1].
- Balance exploration (diverse samples) and exploitation (uncertain samples) to refine decision boundaries [1].



(c) A typical example of deep active learning.

Figure 1: DeepAL Procedure [1]

.

Methodology

Active Learning Strategies

- **Uncertainty Sampling:** Focuses on samples with high prediction uncertainty using entropy and margin-based metrics [1].
- **Hybrid Query Strategies:** Combines uncertainty and diversity to avoid redundancy and ensure dataset representativeness [1].
- **Batch Mode DeepAL (BMDAL):** Queries diverse and informative samples in batches instead of single points [1].

Techniques Explored:

- **Bayesian Neural Networks:** Used for reliable uncertainty estimation [1].
- **Pseudo-Labeling:** Labels high-confidence samples automatically to expand training data [1].

Implementation Highlights:

- **Custom CNN Architecture:** Developed for CIFAR-10 classification.
  - ▷ Three convolutional layers with ReLU activation and max-pooling [1].
  - ▷ Fully connected layers for classification with dropout for regularization [1].
- **Entropy-Based Sampling:** Selected uncertain samples using entropy metrics [1].
- **Evaluation Framework:** Assessed performance on MNIST and CIFAR-10 datasets [1].

Bayesian Inference - Posterior Estimation

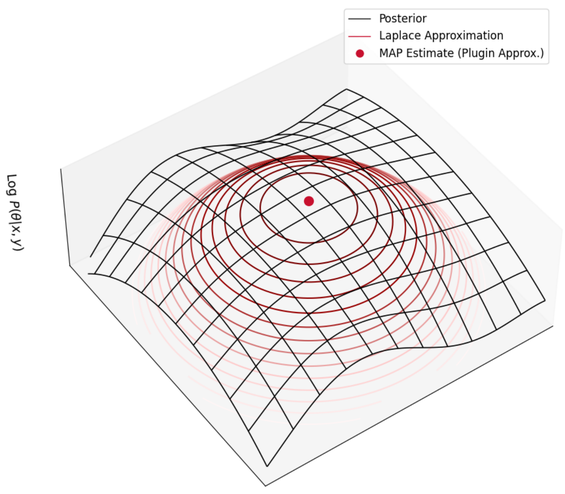


Figure 2: Example of Posterior Estimations

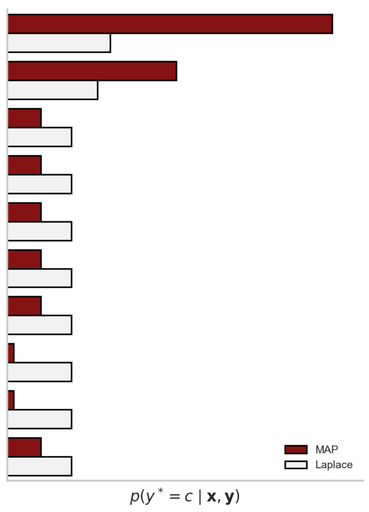


Figure 3: Example of Predictions

MAP-estimation:

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \mathcal{L}(\theta, x, y)$$

Loss function:

$$\mathcal{L}(\theta, x, y) = -\log p(\theta, y, x)$$

$$p(\theta, y, x) = p(y|\theta, x)_{model} p(\theta)_{L2\,reg}$$

where $p(\theta, y, x) \propto p(\theta|y, x)$

Laplace Approximation:

$$q(\theta|x, y) \approx \mathcal{N}(\hat{\theta}_{MAP}, H^{-1})$$

Monte Carlo Sampling:

$$p(y^* = c|x, y, x^*) \approx \frac{1}{M} \sum_{m=1}^{M} (p(y^* = c|\theta_m, x^*))$$

where $\theta \sim q(\theta|x, y)$

- Plugin Approximation
  - Conventional method of computing the class probabilities.
  - Simple Optimization Problem - Minimize the Loss function.
  - Tends to be overconfident in it's predictions - This can have consequences for sample selection.
- Laplace Approximation
  - Estimation of Posterior using Gaussian Distribution.
  - Requires computation of Hessian matrix with $\mathcal{O}(|\theta|^2)$ complexity. This matrix must be inverted - assuring invertability and stability is an issue.
  - Can be simplified to an Isotropic Gaussian Distribution which has $\mathcal{O}(|\theta|)$ complexity. This simplifies invertability and stability of the Hessian matrix.
  - $p(y^* = c|x, y)$ estimated using Monte Carlo sampling methods.

Query Strategies

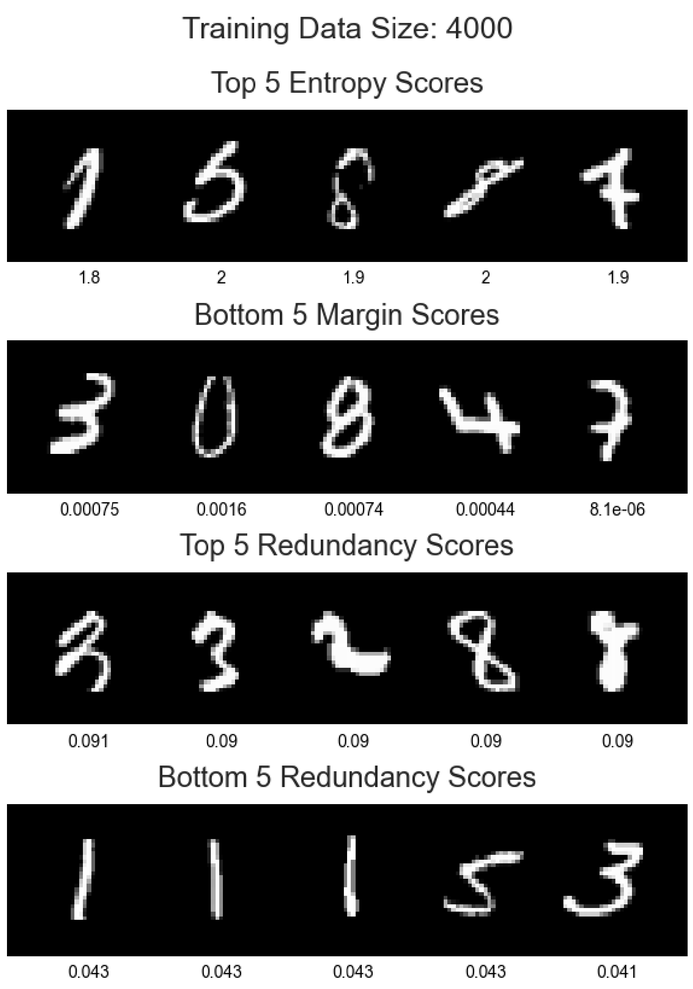


Figure 4: Top/Bottom 5 Query Scores - MNIST CNN model

- Entropy

$$H = -\sum_i p_i \log p_i$$

  - Represents overall uncertainty of predictions.
- Margin
  - Represents ambiguity between top predictions.

$$M = \max_i p_i - \max_{i \neq \text{argmax}_j p_j} p_i$$

- Redundancy
  - Measures similarity between features of other samples.

$$\mathcal{S} = \tilde{\mathbf{F}} \mathcal{M} \tilde{\mathbf{F}}^T, \quad \mathcal{M} = \mathbf{I}$$

where, $\tilde{\mathbf{F}} = \frac{\mathbf{F}}{\|\mathbf{F}\|_{\text{col}}}$.

$$\hat{\mathcal{S}} = \mathcal{S} - \text{diag}(\mathcal{S})$$

$$\mathcal{R} = \frac{\hat{\mathcal{S}}}{|\mathcal{D}|} \mathbf{1}$$

  - Entropy-Redundancy scoring:

$$T = H - \alpha \mathcal{R}$$

Model Performance

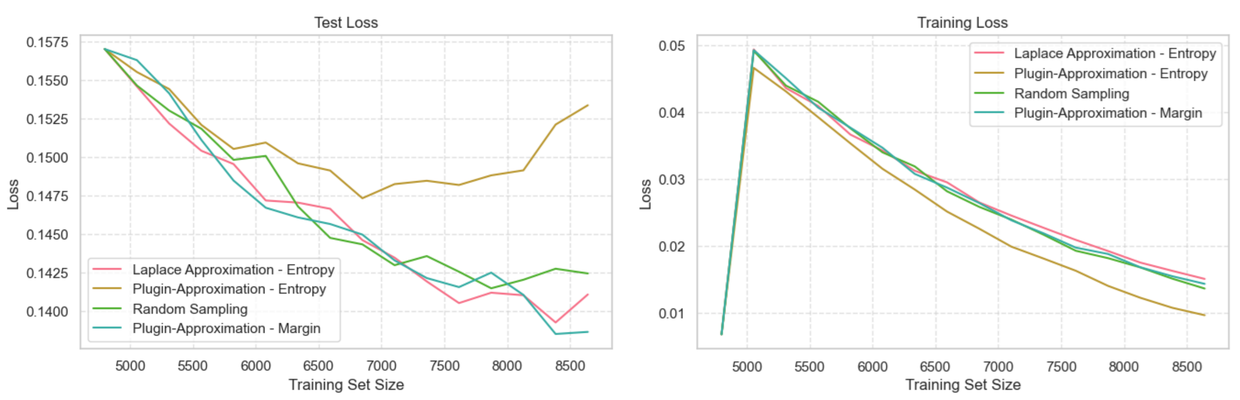


Figure 5: Effect of Query Strategies on Model Performance.

The figure illustrates the performance of the Active Learning strategies on the MNIST Dataset over 20 iterations, Key findings:

- **Training Loss:** All strategies show a consistent decline as the training set grows. Laplace Approximation (Entropy) achieves the lowest losses, indicating efficient learning.
- **Test Loss:** Plugin Approximation (Entropy) and Laplace Approximation (Entropy) consistently reduce test loss, demonstrating strong generalization.
- **Validation and Test Accuracy:** Accuracy improves steadily with training set size, with Laplace Approximation (Entropy) and Plugin Approximation (Entropy) slightly outperforming other methods.
- **Test Average Entropy:** Laplace Approximation (Entropy) maintains higher uncertainty throughout, suggesting more reliable uncertainty quantification.

Key Takeaways

- **Entropy Query Overfitting:** Entropy-based selection risks overfitting by utilizing obscure samples. This might lead to domination of similar samples with reduced diversity and lower training effectiveness.
- **Margin Scoring Simplifies Learning:** Margin scoring centers around ambiguous samples between two classes, making fine-tuning easier since less difficult samples are considered.
- **Over-Parametrization & Overtraining:** In DeepAL the model is often initialized with a smaller initial dataset meaning it is likely to overfit. It's therefore important to not overtrain in the pre-training phase and each cycle after querying new samples. Appending the query samples instead of replacing can result in higher performance but higher risk of overfitting the model.
- **More Conservative Predictions:** Laplace Approximation ensures more conservative estimates which results in more consistent entropy during querying, improving performance over classic entropy methods.
- **Sensitivity of Redundancy Scoring:** Redundancy scoring depends on feature space structure and may misinterpret particular classes (such as 1s) as non-redundant since they don't share features with other classes.

Reference

[1] Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., Wang, X. (2021). A Survey of Deep Active Learning. *ACM Computing Surveys*, 54(9), 1-40. Retrieved from `https://arxiv.org/pdf/2009.00236`