INTELLIGENT DATA LABELING VIA NEURAL NETWORK-DRIVEN ACTIVE LEARNING FOR ENHANCED MODEL EFFICIENCY

Aditya Girase - s240125, Ivor Baricevic - s232467, Mads Alkjærsig - s203372, Samyak Jain - s232883

Technical University of Denmark

ABSTRACT

In this report learning methods for improving data labeling work flows is explored. This includes both active learning methods like querying strategies for selecting samples for the oracle to label. A hybrid approach combining uncertainty estimation with diversity is proposed to refine sample selection. Bayesian neural networks employing Monte Carlo sampling are utilized to address overconfidence issues when using uncertainty as a query. The concept of pseudo-labeling is explored as a potential means of expanding labeled datasets without manual intervention. Experimental evaluations on MNIST suggest that random sampling performs comparably in high-quality datasets, with simpler query-strategies like margin and entropy performing worse. Laplace approximation showed some promising results. It is theorized that combining pseudo-labeling with active learning, particularly entropy-based querying and Laplace approximation, could enhance performance and efficiency in lower-quality datasets.

***Index Terms*—** Active Learning, Semi-Supervised Learning, Neural Networks, Pseudo Labeling, Uncertainty Sampling, Bayesian Neural Networks, Clustering Techniques, MNIST, CIFAR-10, Model Efficiency, Limited Data.

1. ACTIVE LEARNING

Active learning works in repeated cycles where the model asks an oracle, usually a human expert, for labels of the most useful samples from a dataset that does not have labels. The main steps in each of these cycles are three: the model picks samples to label using a query strategy, the oracle gives the labels for these samples, and the model is retrained with the new, larger labeled dataset. Traditional active learning requests one sample at a time, but deep neural networks tend to perform better in batch mode sampling: that is, choosing a few samples at a time while taking into consideration how useful and diverse each sample is to avoid duplicating labels. The challenge is to balance picking samples that the model finds uncertain (exploitation) and those representing the main data patterns (exploration), while balancing performance.[1]

1.1. Querying Strategies

The query strategies entail the scoring method for selection of samples.

1.1.1. Entropy

The entropy strategy evaluates uncertainty by measuring the entropy of the predicted class probabilities, with higher entropy indicating greater uncertainty.[1]

$$H = -\sum_i p_i \log p_i \tag{1}$$

1.1.2. Margin

The margin strategy selects samples with the smallest difference between the top two predicted class probabilities, indicating high uncertainty.[1]

$$M = \max_i p_i - \max_{i \neq \text{argmax}_j\, p_j} p_i \tag{2}$$

1.1.3. Redundancy

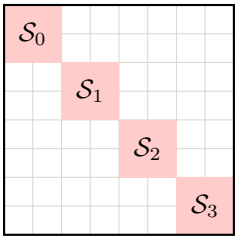
The redundancy computes the sample's similarity with other unlabeled samples using the feature space of the pre-trained model. [1]

$$\mathcal{S} = \tilde{\mathbf{F}}\mathcal{M}\tilde{\mathbf{F}}^T, \quad \mathcal{M} = \mathbf{I}, \quad \mathbf{F} = \begin{bmatrix}\mathbf{f}_1 & \dots & \mathbf{f}_M\end{bmatrix}^T \tag{3}$$

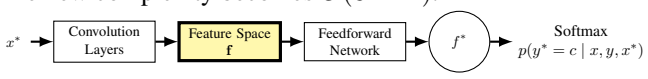
where, $\tilde{\mathbf{F}} = \frac{\mathbf{F}}{\|\mathbf{F}\|_{\text{col}}}$.

$$\hat{\mathcal{S}} = \mathcal{S} - \text{diag}(\mathcal{S}), \quad \mathcal{R} = \frac{\hat{\mathcal{S}}}{|\mathcal{D}|}\mathbf{1} \tag{4}$$

An issue with this algorithm is that it has a complexity of $\mathcal{O}(U^2)$ - U being the size of the unlabeled dataset. Memory issues can be reduced by computing blocks of the matrix at a time and combining them; however, this does not solve computational complexity. To address this, $\mathcal{S}_n$ is computed from M samples.



The unlabeled dataset can then be sliced into N batches. These batches must be large enough to be representative of the true dataset but small enough to reduce computational costs. The new complexity becomes $\mathcal{O}(U \cdot M)$.



1.2. Hybrid-Querying

Redundancy is best combined with an uncertainty estimate such as entropy. By combining them and finding the optimal hyper-parameter it's theoretically possible to balance between challenge and diversity in the set of samples.

$$C = H - \alpha \cdot \mathcal{R} \tag{5}$$

1.3. BNNs using Monte Carlo Methods

The standard approach for estimating optimal parameters θ_{MAP} involves maximizing the posterior distribution using optimization methods, known as the plugin approximation. While straightforward, this often leads to overconfident predictions, which can be problematic for querying. Bayesian inference suggests that the most accurate estimate of θ comes from integrating over the entire parameter space, weighting each value by its posterior probability. However, posteriors are typically intractable, making analytical solutions impractical. Assuming the posterior is centered around a local maximum, it can be approximated by a Gaussian distribution, simplifying computation and enabling feasible sampling. The Laplace approximation provides such a Gaussian approximation to the posterior of neural network parameters.

$$q(\theta|x, y) = \mathcal{N}(\hat{\theta}_{\text{MAP}}, H^{-1}) \tag{6}$$

where $\hat{\theta}_{\text{MAP}}$ is obtained by minimizing:

$$\mathcal{L}(\theta, x, y) = -\log p(\theta, y, x) \tag{7}$$

Even with Laplace approximation issues arise, specifically with the inverted hessian matrix. The primary issue lies in invertabillity which is not assured. The complexity of this hessian is also $\mathcal{O}(|\theta|^2)$, raising both computational and memory issues. Assuming the Laplace approximation is an isotropic Gaussian avoids this, as the hessian will be a diagonal matrix. This reduces complexity to only $\mathcal{O}(|\theta|)$ and makes invertibillity trivial.
Predictive probabilities are estimated using Monte Carlo sampling:

$$p(y^* = c|x, y, x^*) \approx \frac{1}{M} \sum_{m=1}^{M} p(y^* = c|\theta_m, x^*) \tag{8}$$

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

2. SEMI SUPERVISED LEARNING

2.1. Pseudo-Labeling

The pseudo-labeling technique was implemented to enhance the performance of our CNN model. This strategy expands the labeled dataset by assigning high-confidence labels from the base CNN model to the unlabeled data. The initial dataset contained an equal number of classes. By removing high confidence samples from the unlabeled dataset the goal is to gain model performance without requiring labeling from the oracle.[1] The trainer utilizes Early Stopping and Learning Rate Reduction to assure proper convergence to a solution with good generalization capability - avoiding potential overfitting.

To enhance the performance of the CNN model for semi-supervised learning, we implemented two key callbacks to optimize the training process:

3. RESULTS

3.1. Impact of Active Learning:

The following section contains the primary figures illustrating the performance of the active learning model.

Experiments evaluated different active learning strategies on the MNIST dataset. To assure comparable metrics they were averaged over 40 iterations with constant seeds each iteration.

3.1.1. Query Strategies

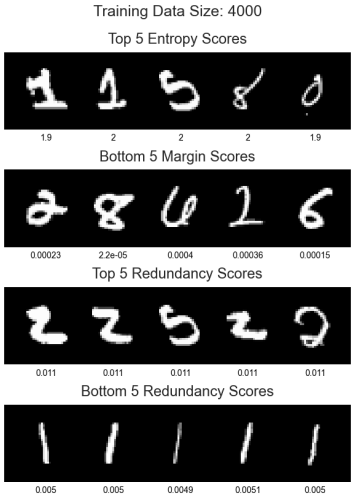


Fig. 1: Top/Bottom Query Strategies - MNIST Dataset

In Figure 1 the top 5 candidates are shown for Entropy, Margin and Redundancy (Top and Bottom). In Figure **??** the performance of query strategies can be seen including entropy utilizing a Laplace approximation. Each cycle involves

adding 256 new samples to the training set. To avoid overfitting early stopping is utilized.

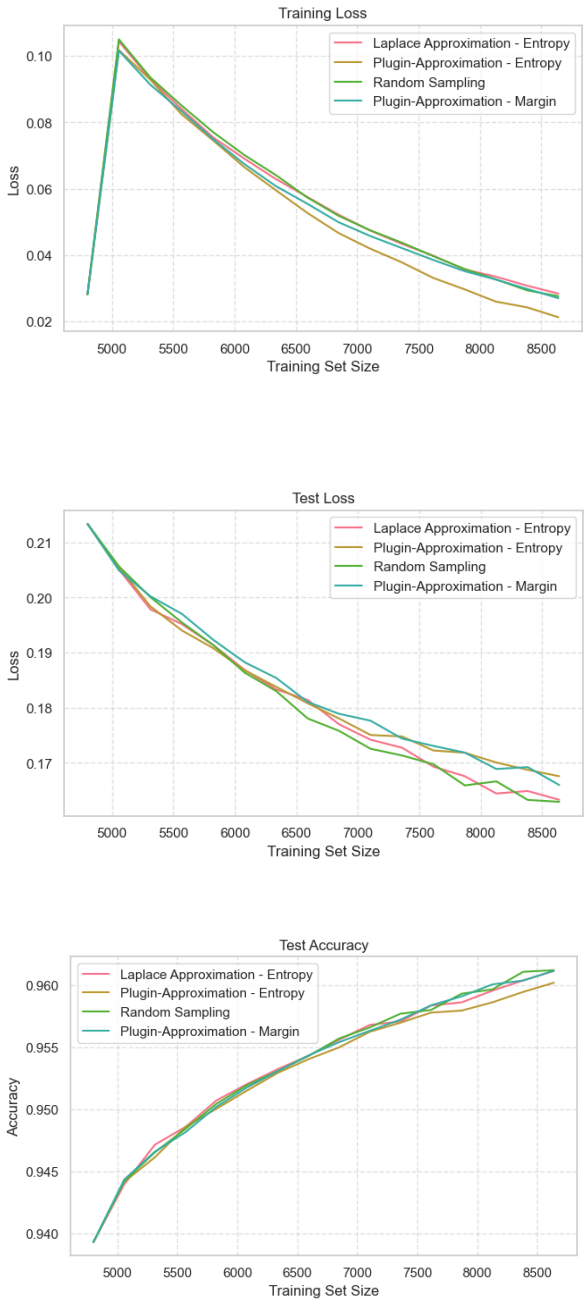


Fig. 2: Performance metrics of the model using all the strategies except redundancy: Training Accuracy, Test Loss and Test Accuracy.

3.2. Hybrid-Querying

In Figure 3 the pre-determined optimal α-value ($\alpha = 5$) is utilized in the hybrid-query strategy combining entropy and redundancy.

Fig. 3: Performance metrics of the model using the entropy and redundancy strategies: Test Accuracy, Test Loss, and Training Loss.

3.3. Pseudo-Labeling:

In Figure 4 the accuracy of the model as it train on the pseudo-labels are plotted.

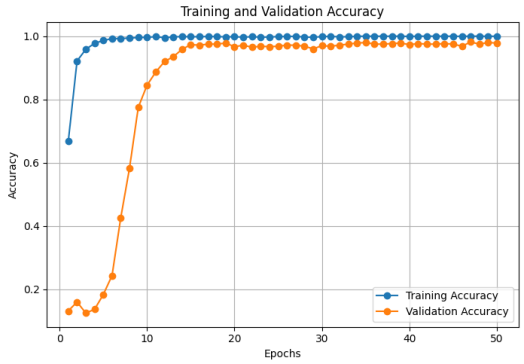


Fig. 4: Training and Validation Accuracy of the model

4. DISCUSSION

4.0.1. Methodology

A noticeable bottleneck in ability to acquire measurements is the volatility of model performances dependent on the train-

ing batches. This results in averaging over multiple iterations being the best option. However, to compute the mean of each metric a substantial amount of runs must be computed and averaged. This results in a computational bottleneck. It was chosen to do 40 runs for each metric. Due to this bottleneck some metric might be closer or contain unwanted noise which can make analyzing the model performances much more difficult.

4.0.2. Overfitting challenges

Deep active learning faces significant overfitting challenges primarily due to initialization with limited labeled data, where the model's ability to learn generalizable features is constrained by the small initial dataset size. The experimental results suggest that entropy-based querying tends to overfit as displayed by its higher training loss. During previous measurements of model performance similar behavior was seen suggesting this is a general trend. This indicates that Laplace Approximation's more conservative estimates provide better stability. This highlights the critical trade-off between maximizing information gain through active sampling and maintaining model generalization, particularly during the crucial early stages of training. Margin seemed to have comparable test loss to entropy but training loss was not aggressive, meaning the poor performance might be less due to overfitting and more due to sample selection being less effective.

4.0.3. Random Sampling Performance

What is noticeable in the results is that random sampling seems to result in very competitive results only challenged by the Laplace approximation using entropy. A reason why this could be is that datasets such as MNIST and CIFAR10 often contain particularly great sample quality. A case could be made that datasets that are of poorer quality would perform much worse when randomly sampled as they often contain redundancies and noninformative data that active learning methods are designed to avoid.

4.0.4. Feature Space and Hybrid-Query performance

In Figure 1, the samples with lowest redundancy score, seem quite redundant. A reason this might be is the nature of how the convolution layers are structured. It's very likely that vertical lines activate very few nodes in the feature space layer. Hence, since similarity is a dot product of feature spaces the similarity the majority of other samples is very small. A balance between entropy and redundancy should eliminate most of these samples due to entropies being very low.

4.0.5. Pseudo-labeling

After incorporating pseudo-labeling, the model achieved an accuracy of 99%, which was consistent with its initial performance. This indicates that the model was able to effectively integrate pseudo-labeled data without any significant degradation in performance. As can be seen Figure 4 the model generalizes well suggesting that the pseudo-labels do not result in overfitting. Hence, the method resulted in improved performance without requiring any manual labeling from the Oracle.

4.0.6. Combination of Learning Methods

A diverse set of different methods have been discussed including active learning and semi-supervised learning methods. However, due to time-constraints and technical difficulties (including some scope-creep) it was not feasible to combine these methods into one, which restricts the conclusions that can be made. However, it can be theorized that a Laplace approximation using entropy-querying and pseudo-labeling would perform well. The pseudo-labeling could help with good confidence-estimations, however the entropy tends to decrease as a model trains - the Laplace approximation is much more resistant to this. The redundancy scoring did not result in a noticeable performance increase. Due to this it would be unwise to use, as computing a decent estimate of the similarity matrix is very expensive.

5. CONCLUSION

From the findings it cannot be concluded that the implemented Active Learning strategies have a significant impact on performance, however Laplace approximation seemed consistently better than plugin approximations - reason likely being the overconfidence of the latter. Redundancy did not have a significant impact either. It is theorized that active learning has bigger impacts when the data quality is poor. MNIST might have such diverse and informative samples that random sampling becomes a decent strategy. Pseudo-labeling resulted in a decent increase in performance with very little trade-off. It's possible that a combination of pseudo-labeling and a active learning using entropy-querying and Laplace approximation would give best results.

Github Repository: https://github.com/Trojan74/Active-Learning-Strategies/tree/main

6. REFERENCES

[1] P. Ren, Y. Xiao, X. Chang, P.Y. Huang, Z. Li, B.B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," in *ACM Computing Surveys*. ACM, 2021, vol. 54, pp. 1–40.

7. APPENDIX

7.1. Active Learning Architecture

- **Input:** Single-channel grayscale image (28×28 pixels).
- **Convolutional Layers:**
  - Conv1: $1 \to 16$, 3×3 kernel, stride 1, padding 1.
  - Conv2: $16 \to 32$, 3×3 kernel, stride 1, padding 1.
  - Conv3: $32 \to 64$, 3×3 kernel, stride 1, padding 1.
  - All conv layers: BatchNorm, ReLU, MaxPooling.
- **Feature Space:** 64 nodes.
- **Feed-Forward Layers:** $64 \to 512 \to 10$, with Dropout for regularization, ReLU is used.
- **Hyperparameters:** Batch size: 64, Learning rate: 10^{-4}, Weight decay: 10^{-5}, Optimizer: ADAM, Criterion: Cross-Entropy.

7.2. Semi-Supervised Learning Architecture

- **Input:** Single-channel grayscale image (28×28 pixels).
- **Convolutional Layers:**
  - Conv1: $1 \to 32$, 3×3 kernel, stride 1, padding 1, L2 regularization ($\lambda = 0.001$).
  - BatchNorm, ReLU.
  - Conv2: $32 \to 32$, 3×3 kernel, stride 1, padding 1, L2 regularization ($\lambda = 0.001$).
  - BatchNorm, ReLU.
  - MaxPooling: 2×2.
  - Conv3: $32 \to 64$, 3×3 kernel, stride 1, padding 1, L2 regularization ($\lambda = 0.001$).
  - BatchNorm, ReLU.
  - Conv4: $64 \to 64$, 3×3 kernel, stride 1, padding 1, L2 regularization ($\lambda = 0.001$).
  - BatchNorm, ReLU.
  - MaxPooling: 2×2.
- **Feed-Forward Layers:**
  - Dense1: $3136 \to 128$, L2 regularization ($\lambda = 0.001$), Dropout (20%), ReLU.
  - Dense2: $128 \to K$, L2 regularization ($\lambda = 0.001$), Softmax.
- **Hyper parameters:**
  - Batch size: Default.
  - Learning rate: Adjusted by Adam optimizer.
  - Weight decay: L2 regularization ($\lambda = 0.001$).
  - Optimizer: Adam.
  - Criterion: Categorical Cross-Entropy.

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]

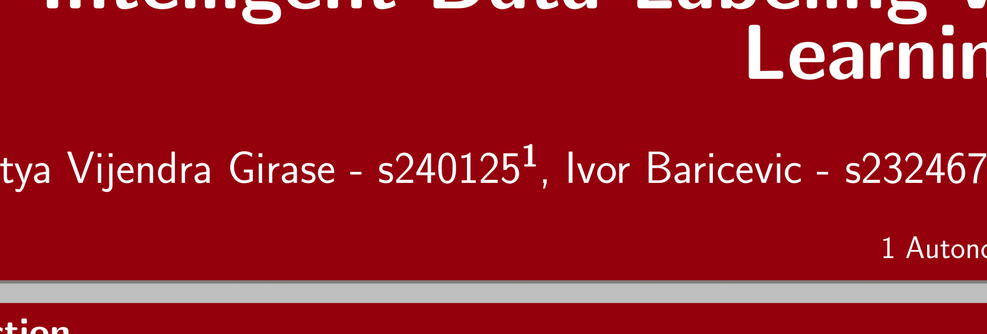
1 Autonomous Systems, Technical University of Denmark; 2 Acoustics, Technical University of Denmark

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

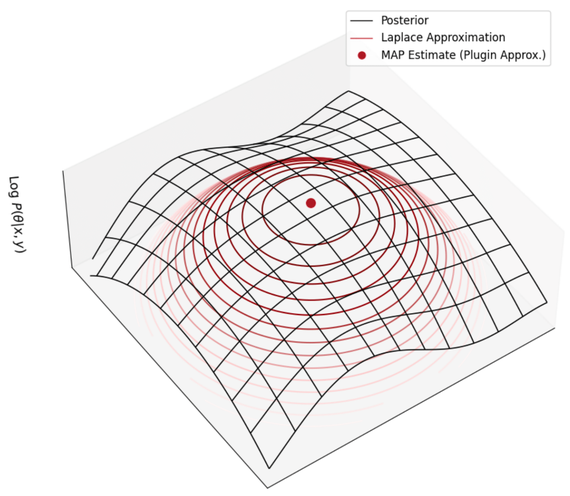


Figure 2: Example of Posterior Estimations

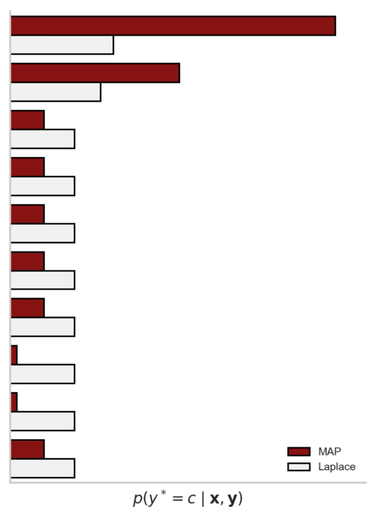


Figure 3: Example of Predictions

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

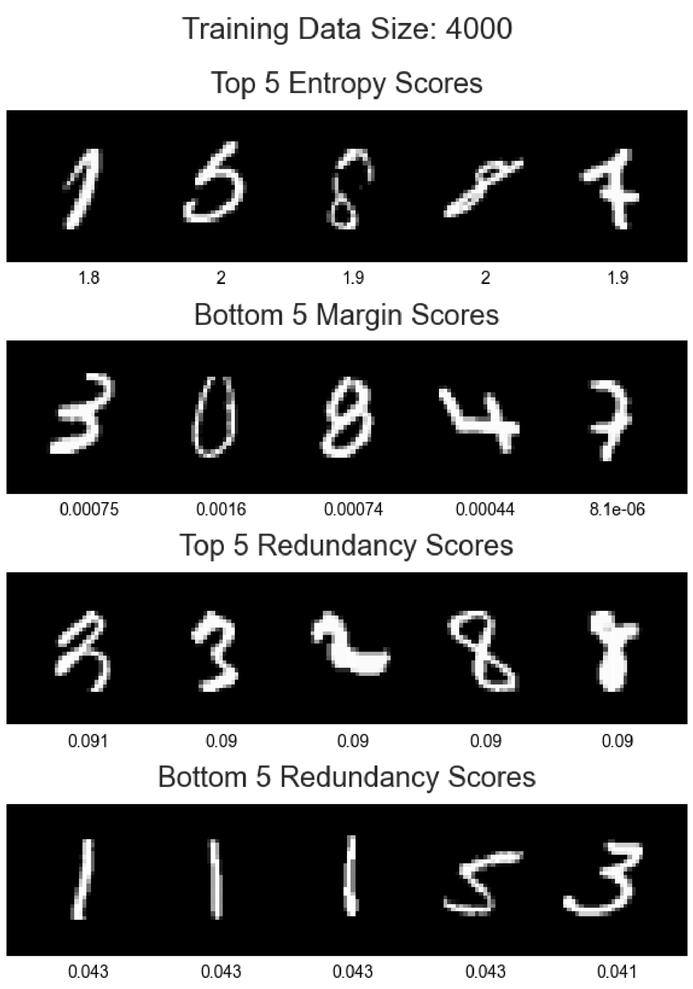


Figure 4: Top/Bottom 5 Query Scores - MNIST CNN model

- 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

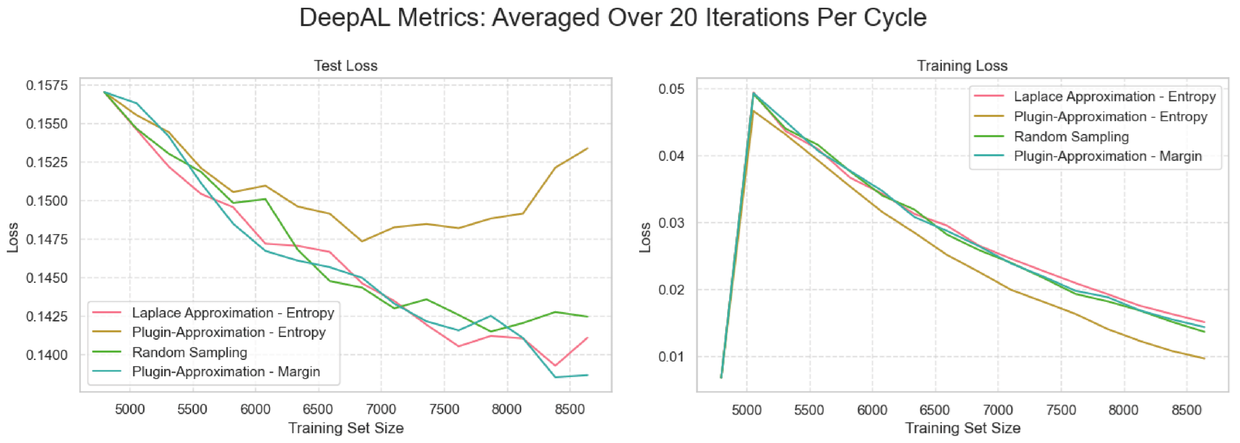


Figure 5: Effect of Query Strategies on Model Performance.

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 Takeways

- **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`