Mphasis
The Next Applied

Active Learning for Building and Maintaining High Performing Machine Learning Models

Whitepaper by Vibha Bhagchandani, AVP – AI & Cognitive Solutions, Mphasis NEXT Labs |
Manish Shukla, Senior Associate Business Analyst, Mphasis NEXT Labs

Contents

1.
Introduction

The lifecycle of a Machine Learning (ML) system typically starts from understanding the problem statement, collecting the necessary data and preprocessing the data, up to when the model is out of the production pipeline. Model building is not the end goal of ML system lifecycle, but only a phase of it. ML system lifecycle's end goal is to make the model production-ready, deploy the model in the production pipeline, monitor its performance, retrain the model to account for data and Concept Drift and redeploy to continue the lifecycle. As long as a model is in the production pipeline, it requires monitoring to obtain accurate and efficient results.

There are six phases in the ML system lifecycle:

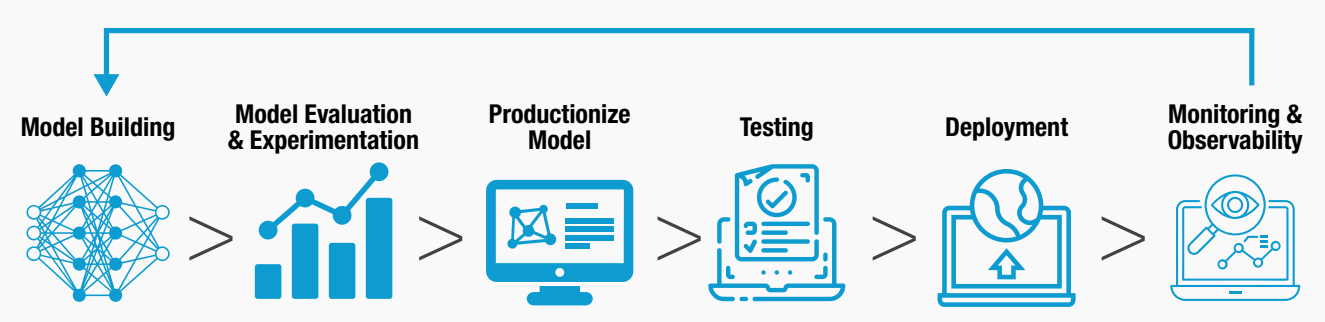

Fig 1: ML System Lifecycle

- **Model Building:** This phase includes understanding the problem, converting it into the ML framework, data preparation, feature engineering and primary model building.
- **Model Evaluation and Experimentation:** In this phase, the model is evaluated against available validation data. Accordingly, hyperparameters are tuned and different algorithms are used to check the performance of the model by using various metrices such as accuracy, precision, recall, f1-score, area under curve, etc.
- **Making Model Production-ready:** Once the model and hyperparameters are set, code is converted to be put in production.
- **Testing:** Model is tested in production environment and the results are compared with the results obtained during model evaluation and experimentation phase.
- **Deployment:** After the final testing model is deployed, real-time predictions can be made by accessing the Application Programming Interface (API).
- **Model Monitoring:** This is the final phase of the ML system lifecycle. Here, the model is monitored to ensure that it performs its intended job and maintains a desired level of performance. It is used to understand the performance of the model on real-time data. If the output of ML models in not monitored correctly, it may lead to inaccurate predictions, leading to loss in business.

2.
Why Model Monitoring is Important

During Machine Learning training, historical data is mapped to the target variable using a function like sigmoid or regression. This function learns the rules present in the data for mapping it to the target variable. One assumption taken during the training of the model is that the future data will be representative of past data. But in real case scenario, the inherent properties of data can change over time and the performance of the model can deteriorate. This phenomenon is called Model Drift. If Model Drift is not detected, it can lead to poor performance of production models and hence the overall pipeline. There are two major reasons for Model Drift known as Concept Drift and Data Drift.

Concept Drift: When the statistical properties or the definition of the target variable changes, it is called Concept Drift. For example, suppose a spam detector ML model is trained to predict which content contains spam. Over time, the definition of what is to be categorized as spam changes, then this scenario is known as Concept Drift. So, it can be concluded that Concept Drift happens when a new kind of spam emails starts appearing.

Data Drift: When the statistical properties of the predictor or the independent variable changes, it is called Data Drift. For example, in case of spam detector ML model, when spammers change the way of writing similar kind of spam emails, then this scenario is known as Data Drift.

There could be various reasons for the deterioration of model performance:

- **Unseen Data:** Machine Learning models are trained on labeled data and there will always be a limit to the data that can be manually tagged. Unseen data is the new data that might be different from the labeled data and models' predictions can be less accurate on unseen data.
- **Time:** Time is another important factor because of which monitoring of the Machine Learning model is required. With time, statistical properties of the data, variables, parameters and even the outcome of the model are likely to change.
- **Change in Data Format:** Every Machine Learning model expects data in a predefined format. Due to any unforeseen conditions, if the format of the data is manipulated, performance of the model will degrade.
- **Overall:** The model is trained on the data present at hand and all the corner cases are considered while building the model, but we can never be sure of all the possible scenarios that may happen in future. To incorporate this uncertainty in terms of data, problem statement, variables and parameters while maintaining the desired performance of the model, its monitoring is required.

Now that we know why model monitoring is required, the next big question is how we monitor ML models.

There are ways to detect the drift that happened in the model or data by analyzing the predictions and model accuracy, monitoring the statistical properties of data and how it affects or correlates with the prediction over time and various algorithms. If we identify that our model is not maintaining a desired level of accuracy, we need to work towards improving the model performance. In this paper, we explore, how to get started with improving your ML models by leveraging the concept of Active Learning.

3.
Active Learning

Active Learning is the process of sampling data points from a pool of unlabeled data for labeling so that the model performance can be improved significantly. Active Learning comes in handy while deciding which data points are to be labeled, in the order of priority. With Active Learning, we can sample data points for model building upfront. It is helpful in not just identifying the initial dataset, but also in choosing the right dataset for retraining the model for enhancements.

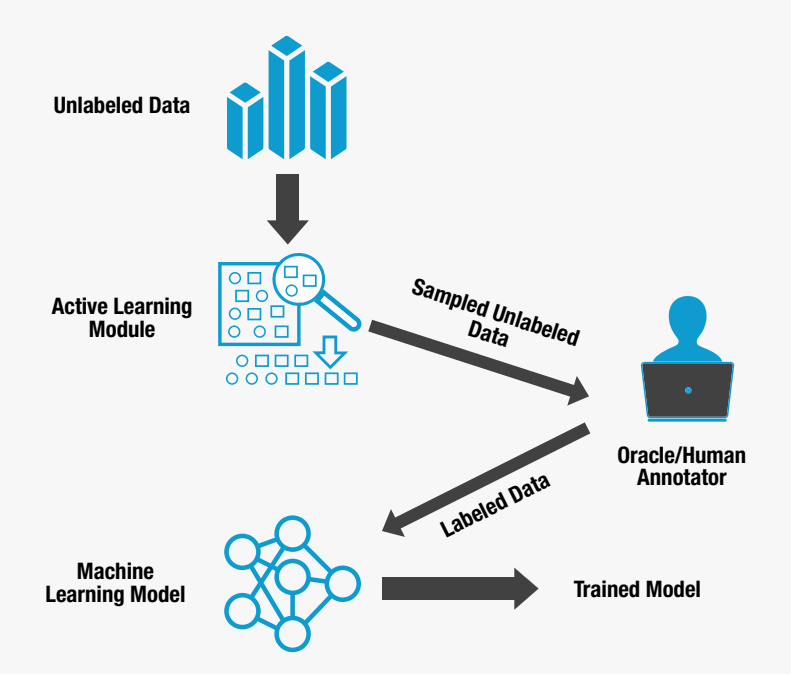

Fig 2: Active Learning Framework

There are various ways to do Active Learning, but the most generic steps involved are:

1. A small sample of data must be selected from the unlabeled data for manual tagging.
2. Once the data is tagged, it must be used to train the model. The model's performance will not be very good, but will provide insight into which areas of the parameter space should be annotated first.
3. Once the model is trained, it should be used to make prediction on the unlabeled data.

4. A prioritization score is calculated for each unlabeled data point based on the prediction of the model. There are various ways to calculate the prioritization score – explanation provided in the later part of this section.
5. Based on the prioritization score, data points can be sampled from the unlabeled data and annotated further.
6. If the performance of the trained ML model is not as expected, the process of sampling and retraining can be repeated several times until the desired level of performance is achieved.

There are various ways to do sampling known as sampling strategies. Based on the prioritization score and methodologies used for sampling, these strategies can be categorized as:

1. Uncertainty Sampling Strategy

Whenever a trained Machine Learning model makes a prediction, it gives a probability score representing the confidence of prediction. If the confidence of prediction is low, it is a good practice to seek human feedback to improve the model performance. This type of Active Learning where we seek human feedback in low confidence model predictions is known as **Uncertainty Sampling.** For example, in case of a binary classification problem, model prediction will be better for the data points that are significantly away from the decision boundary which separates the two classes. In this case, if we want to improve the model accuracy by increasing the training data size, it will be impractical to sample data points that can be well predicted by the classification model. Whereas data points falling near the decision boundary might confuse the ML model while making prediction and if these data points are validated by a human and classified into specific classes, then overall performance of the model improves. Active Learning with Uncertainty Sampling helps in identifying the data points falling near the decision boundary and sampling these data points for annotation will improve the overall performance of the ML model. While performing Active Learning using Uncertainty Sampling, a score is assigned to each data point known as confidence score.

There are various ways to calculate the confidence score of prediction by a Machine Learning model and some of them are:

a. Least Confidence

This is the simplest prioritization score method. In this method, predicted probabilities by ML model are used to calculate the confidence of prediction.

If probability of prediction is greater than 0.5, then:

$$\text{Confidence} = \text{probability of prediction}$$

And if the probability of prediction is less than 0.5, then:

$$\text{Confidence} = 1 - (\text{probability of prediction})$$

b. Margin of Confidence

In this methodology, prioritization score is calculated by subtracting highest probability and the second highest probability of the data point belonging to different classes. Data points are then selected for annotation based on the lowest margin sampling score, as these data points are least certain about most probable and next to most probable class. Let us consider the probability of prediction for two data points in case of a spam detector ML model.

Data Point	Spam Probability	Not Spam Probability
A	0.8	0.2
B	0.3	0.7

The margin of confidence for data point A will be 0.6 (0.8-0.2) and margin of confidence for data point B will be 0.4 (0.7-0.3). Data point B has lower margin of confidence, hence ranks higher in priority for sampling.

2. Diversity Sampling Strategy

The method of ensuring that you have diverse training data for your model is a type of Active Learning called Diversity Sampling. This type of Active Learning is useful in increasing the coverage of training data and ensuring that all kinds of data are included in the training sample.
In Uncertainty Sampling, the trained Machine Learning model is uncertain to make prediction on a data point, whereas in Diversity Sampling, the model is unaware of the data points. For example, training data for a multi-class classification problem must be sampled. But in the unlabeled data, there is significant class imbalance and if data is sampled randomly chances of data points belonging to majority class, will be more in the training data, as compared to the minority class. Hence, the minority class data points will have lower representation in the training data.
In another scenario, suppose, during Machine Learning model lifecycle, due to dynamicity new kind of data trends or classes start appearing for which model is not trained. If we want to sample these data points in the training data before the model retraining is done, then Diversity Sampling methods can be used.

Some of the Diversity Sampling Active Learning methods are:

a. Model-based Outliers

Sampling strategy to sample data points that are confusing to the model because of the lack of information or weightage in the training data.

b. Cluster-based Outliers

This sampling strategy is used to sample all kinds of trends present in the data and not just those where most of the data points are present. For this approach, unsupervised learning algorithms are used to pre-segment the data so that all trends are recognized. With this approach, data points that are not part of any trend or cluster (e.g., outliers) are also identified to increase coverage.

c. Representative Sampling

Sampling items that are representative of the target domain for your model, in relevance to your current training data. For example, during Machine Learning lifecycle, there is drift in the target variable and existing training data does not have representation for the new class. In such a scenario, this method helps in sampling data points that are more relevant for the new target domain.

d. Real World Diversity

This sampling strategy is used to incur fairness and support real world diversity in the training data. With the help of this sampling strategy, it is ensured that the training data is as fair as possible and free from real world biases such as demographic or gender biases.

3. Transfer Learning for Active Learning

Transfer Learning is the process of using knowledge gained while solving a ML or DL problem and applying it to another related problem. For example, knowledge learned in cat/dog classification model can be used while classifying human and pet. Transfer Learning has gained a lot of popularity because of its ability to train deep neural networks with comparatively lesser data. Transfer Learning enables the use of a pre-trained neural network for a new task by changing the last or last few layers of the network. For image classification use cases, ImageNet provides a mechanism to gain the benefits of Transfer Learning when large amount of image data is not available upfront to train the models from scratch. Similarly, for text analytics use cases, pre-trained model like BERT for natural language processing tasks add a lot of value to build production models at scale and speed. By altering the last layers of the neural network, the new labels can be any category that we want. By using the concepts of Transfer Learning, Active Learning can be performed in the following ways:

a. Active Transfer Learning for Uncertainty Sampling

The basic idea of using Transfer Learning with Uncertainty-Sampling-based Active Learning is to predict its own errors whenever the model is uncertain and less confident. Let's understand the concept using a deep learning model that classifies an input to one of the 4 output labels – A,B, C or D.

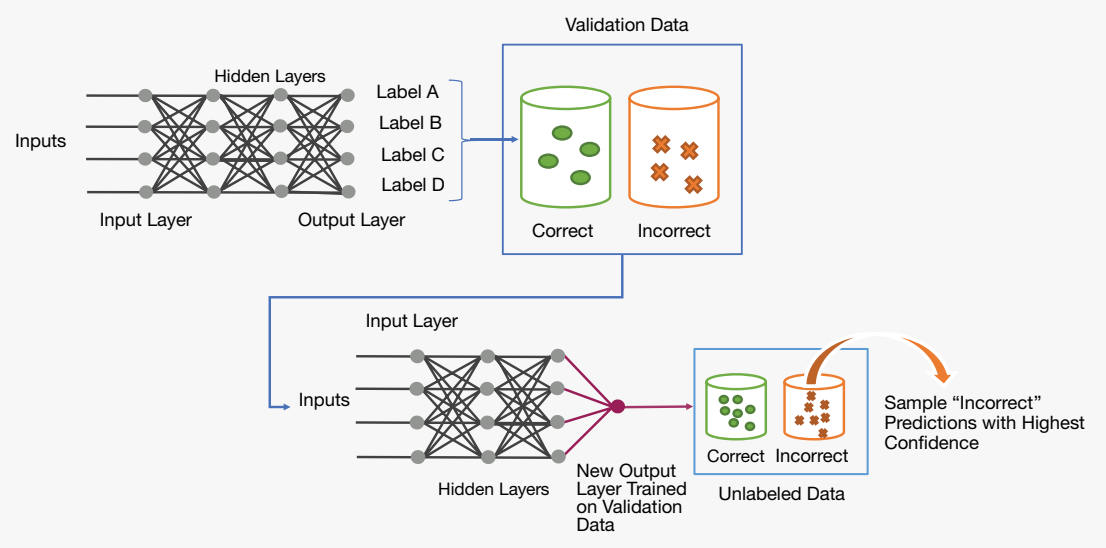

Fig 3: Active Transfer Learning for Uncertainty Sampling

Looking at validation data annotated for the four-class labels, correct and incorrect classification can be easily segregated by comparing actual label present in validation data against predicted labels. The final layer of the existing deep neural network is then changed and is trained with validation data having labels as “Correct” and “Incorrect”. After training, when this model makes prediction on unlabeled data, data points will be classified as “Correct” and “Incorrect”. Sampling the “Incorrect” predicted data points with the highest confidence make up the candidate dataset for human annotation that will improve the performance of the DL model.

b. Active Transfer Learning for Representative Sampling

We have already discussed drift and how it affects a Machine Learning model’s performance over time. The new data might not be of the same distribution as the training data. In such scenarios, we classify data into two categories such as training domain data and application domain data where application domain data is the new data after the drift has happened. Active Transfer Learning for Representative Sampling helps in increasing the representation of application domain data in the updated training dataset before the model retraining is done. Application domain data can belong to newer classes, as well as those yet to be identified. But that should not be a matter of concern because after the data is sampled, it will get a human label. For example, in case of cat/dog classification problem, with time, some images of rabbits start appearing due to drift. Existing AI model is not trained to classify rabbits and hence retraining is required. The existing validation data with cat and dog images will be “Training” domain data and unlabeled data with cat, dog and rabbit images will be labeled as “Application” domain data. Then, existing DL model is trained with this new data comprising of “Training” domain and “Application” domain data. Once trained, when this model is used to make predictions on unlabeled data, it will classify it into “Training” and “Application”, and sampling “Application” with highest confidence and providing with human label will increase the representation of new data in the training data.

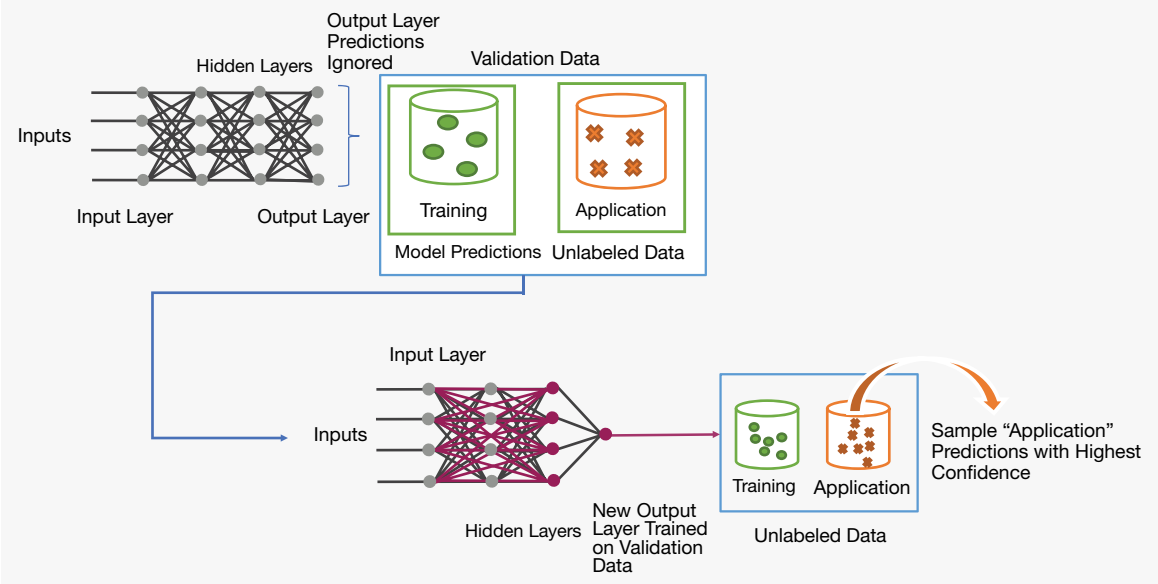

Fig 4: Active Transfer Learning for Representative Sampling

c. Active Transfer Learning for Adaptive Sampling

This sampling strategy combines Active Transfer Learning for Uncertainty Sampling and Active Transfer Learning for Representative Sampling. With this strategy, strategy both uncertain and diverse data points can be sampled. Let us understand the concept with the help of the following diagram.

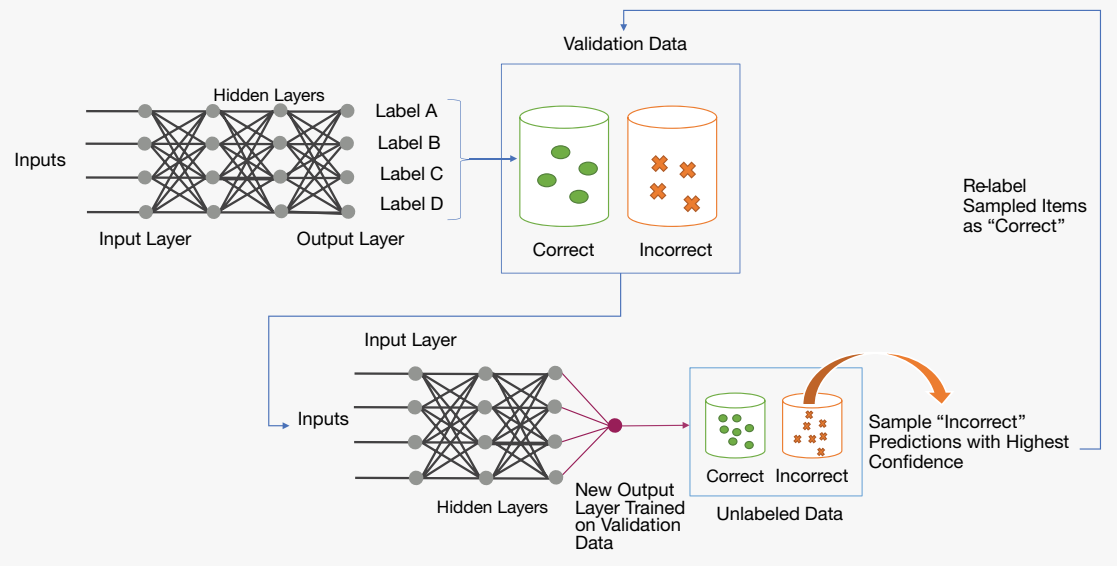

Fig 5: Active Transfer Learning for Adaptive Sampling

Like Uncertainty Sampling, the existing DL model is used to make prediction on validation data and when actual labels are compared against predicted label, correct and incorrect predictions can be found out. Final layer of the existing DL model is changed and is trained with validation data with "Correct" and "Incorrect" as the labels. When the new trained model makes prediction on unlabeled data, it will classify each data into "Correct" and "Incorrect" prediction. The "Incorrect" predictions with the highest confidence are sampled and re-labeled as "Correct". The same process is performed iteratively until sufficient data points are sampled. This is done because the model will make prediction as "Incorrect" for both kind of data points, which are either lying near the classification boundary, i.e., uncertain and those not seen by the model, i.e., outliers and newer trends after drift. Re-labeling these "Incorrect" as "Correct" will make sure that these data points will have representation in the training data even though the label of these data points is currently unknown. Performing the same task in the next iteration will result in sample data points that are different from the previous iteration, thus improving the coverage. Once sufficient data points are selected, all the sampled data points are provided with human label. The newly tagged data will have newer classes as well as data points on which initial model was uncertain to make prediction. This is how both the uncertainty and diversity of the data points are handled, with Active Transfer Learning for Adaptive Sampling.

4.
Active Learning for Incremental Learning

Incremental Learning is a Machine Learning paradigm where the learning process occurs whenever new example(s) emerge and adjust what has been learned according to new example(s). The most prominent difference of Incremental Learning from traditional Machine Learning is that it does not assume the availability of a sufficient training set before the learning process, but the training examples appear over time. Incremental Learning helps in improving the model performance in dynamic settings when newer data is available over time or whenever data size is out of system memory limits. The main task of Incremental Learning is to learn the new rules present in the newer data without forgetting the rules learned previously. The process of forgetting the previously learned weights and biases during retraining of the neural network is known as catastrophic forgetting and poses the biggest challenge in Incremental Learning. There are various ways to do Incremental Learning. For this paper, we have adopted a simple Incremental Learning technique. Instead of training the Machine Learning model for the complete training data, which consists of previous training data and newer sampled data using Active Learning we load the previously learned Machine Learning model from disk and retrain it with newer data only. By hyperparameter tuning, we control the number of epochs in such a way that the new rules can be learned without significantly changing the weights and biases learned previously. Hence, the objective is to learn new rules optimally without forgetting the already learned rules.

5.
Active Learning in Practice

In this section, we will discuss a Machine Learning model's performance, which is retrained incrementally on a sample of data selected using Active Learning methodology.

We trained a Machine Learning model on news headline data to classify the short description of news headline into the following six classes of News type: *Business, Crime, Education, Entertainment, Politics* and *Sports.* The overall accuracy of the trained model on validation dataset was at 82.1%. To check if the model performance improves with Active Learning followed by Incremental Learning we sampled 1000 news headlines short description from the pool of unlabeled data using Active Learning. We selected Uncertainty Sampling with least confidence score as our sampling strategy of choice. Based on confidence scores, 1000 data points of short description text were selected where the model was least confident. These 1000 news headlines were then labeled by an annotator into one of the above mentioned six categories. Then these 1000 reviews became the new training data, and our model was trained incrementally with the new training data set of 1000 annotated samples. When this retrained model was used to make prediction on same validation dataset, the overall accuracy of the model improved to 85.3%.

Thus, by just one iteration of sampling, we could increase the accuracy of the model by approximately 3%. The same process of sampling the data points from unlabeled data and retraining the model was repeated once more, and the overall accuracy of the model increased to 86.2%, i.e., 1% increase in overall accuracy. As we can see, the overall improvement in accuracy in the first iteration is higher than the overall improvement in accuracy in the second iteration. As we iteratively perform Active Learning followed by Incremental Learning, we will observe higher accuracies that will start diminishing with increasing number of iterations and will hit a point of saturation, where the model has learnt all patterns from the data it has seen.

6. Conclusion

When starting to build a supervised ML model, Diversity Sampling can be used to determine the set of data points to be labeled at priority. However, when trying to improve the performance of a model already in production, one can start with Representative Sampling.

When the model performance is deteriorated over time, due to drift in the data, Active Learning methodology can be used to improve the performance of model, by labeling considerably lower amount of data, as compared to when Active Learning methodologies are not used. This is achieved by sampling targeted data points having the most impact on the performance of model. Confidence score generated for the purpose of Active Learning sampling can be used as a metric to monitor the performance of the model. If the confidence score for a significant number of data points is low, then it signals for the need of retraining the deployed Machine Learning model. Understanding why the model performance has deteriorated over time is necessary to find out the correct Active Learning methodology. **Uncertainty Sampling and Diversity Sampling** methodologies are used in case of Data Drift and Concept Drift, respectively. If both data and Concept Drift happen simultaneously, Active Transfer Learning for Adaptive Sampling can be used to improve the model performance.

Active Learning reduces the rigorous task of manual tagging for a huge volume of data, thus saving a significant amount of manpower and time. This is achieved by prioritizing labeling with the help of targeted sampling. Incremental Learning technique also plays an important role in determining the performance of the deployed Machine Learning model. Hence the challenges faced by Incremental Learning, such as catastrophic forgetting, must be addressed appropriately. Active Learning followed by Incremental Learning can improve the performance of the model only up to a certain limit. Performing retraining iteratively will give decreasing returns. After a few iterations, the model performance will not improve any further once the model has learnt enough of all available data patterns. Model performance using Active Learning and Incremental Learning can only be improved in case the base model is not performing well. If the Machine Learning model performance is already high, then using active and Incremental Learning will contribute significantly to any further performance improvements.

Another important benefit of Active Learning is that it allows for iteratively labeling the data during the training cycle. Hence feedback on the model performance is obtained faster rather than waiting for all the training data to be labeled upfront. This allows for course correction and detecting any issues much faster in the ML model development lifecycle.

7. References

- https://freecontent.manning.com/active-transfer-learning-with-pytorch/#
- https://en.wikipedia.org/wiki/Active_learning_(machine_learning)
- https://en.wikipedia.org/wiki/Incremental_learning
- https://en.wikipedia.org/wiki/Transfer_learning
- https://towardsdatascience.com/uncertainty-sampling-cheatsheet-ec57bc067c0b
- https://towardsdatascience.com/https-towardsdatascience-com-diversity-sampling-cheatsheet-32619693c304
- https://towardsdatascience.com/knowledge-quadrant-for-machine-learning-5f82ff979890
- https://towardsdatascience.com/advanced-active-learning-cheatsheet-d6710cba7667

Authors

Vibha Bhagchandani

AVP – AI & Cognitive Solutions, Mphasis NEXT Labs

With 14+ years of experience in the technology industry, Vibha is responsible for solutioning and building IP components as well as Proof of Concepts for client-focused cognitive projects. Vibha led the efforts for onboarding Mphasis proprietary ML algorithms on various cloud provider marketplaces like AWS. She has been actively involved in the research, development and delivery of Natural Language Processing (NLP) focused product development initiatives for some of Mphasis' premier clients. She has done her B.E. in Computer Science and a post-graduate in Management from the Indian School of Business (ISB), Hyderabad.

Manish Shukla

Senior Associate Business Analyst, Mphasis NEXT Labs

Manish Shukla is part of the Mphasis innovation and research group - Mphasis NEXT Labs. He holds a M. Tech. degree from Indian Institute of Technology, Kanpur in Industrial & Management Engineering. His technical skills include Machine Learning, Deep Learning, Operations Research and Statistical Modelling. He has extensively worked in the field of Text Analytics, Natural Language Processing, Computer Vision, Email Order Processing and Information retrieval from unstructured documents. He has keen interest in advanced AI ML technologies and has been currently exploring different methods of Applied AI focused on NLP.

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ($C = X2C^{2}_{TM} = 1$) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. To know more, please visit www.mphasis.com

For more information, contact: marketinginfo.m@mphasis.com

USA
460 Park Avenue South
Suite #1101
New York, NY 10016, USA
Tel.: +1 212 686 6655

UK
1 Ropemaker Street, London
EC2Y 9HT, United Kingdom
T : +44 020 7153 1327

INDIA
Bagmane World Technology Center
Marathahalli Ring Road
Doddanakundhi Village
Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

www.mphasis.com

NR 12/03/21 US LETTER BAS16770