UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE
KATHMANDU

6CS012 - Artificial Intelligence and Machine Learning

Sarcasm Detection in Headlines using RNN, LSTM, and Word2Vec Embeddings

Team Members : Hrikiss Chitrakar

Student Number : 2329260

Course : Bachelor’s (Hons) in Computer Science

Module Leader : Simon Giri

Tutor : Durga Pokharel

Q.N.1. You are a Machine Learning Engineer at a growing e-commerce company preparing to implement and scale machine learning systems.

◦ List and explain at least three real-world challenges you expect during ML model development, deployment, or maintenance (e.g., data drift, imbalanced data, system latency)

◦ For each challenge:

◦ Discuss potential consequences if not properly addressed.

◦ Propose technical or organizational solutions you would implement (e.g., retraining pipelines, feature monitoring, distributed serving, MLOps practices)

◦ Finally, reflect on how cross-functional collaboration (between data scientists, engineers, product teams) can help mitigate these challenges more effectively.

Answer:

The creation and implementation of machine learning systems in e-commerce platforms encounter numerous practical obstacles. Data drift, imbalanced data, and system latency represent three major challenges.

1. Data Drift

Customer behavior patterns along with product preferences and seasonal trends undergo transformations over time. The model's predictive accuracy diminishes when it relies on obsolete training data. The system may generate inadequate recommendations which then negatively impact both user satisfaction and business revenue. I propose establishing systematic surveillance of input data distributions combined with periodic model retraining whenever significant deviations emerge. This process finds support through tools like Evidently, AI alongside cloud platforms' inherent drift detection features.

2. Imbalanced Data

Numerous e-commerce applications including fraud detection and return prediction face situations where certain outcomes manifest at significantly lower frequencies than others. The model could potentially disregard these minority cases completely if they remain unaddressed. The potential for serious oversight exists where fraudulent activity goes undetected. This problem requires the application of data balancing techniques which include oversampling methods like SMOTE, under-sampling approaches, and modifying loss functions to assign greater weight to infrequent events.

3. System Latency

Intricate large model architectures cause excessive response delays during real-time interactions. The shopping experience on e-commerce platforms suffers from even minimal delays. To tackle this issue, I would employ model optimization techniques such as quantization or transform models into lightweight formats like TensorRT. Examining distributed model serving emerges as an option to achieve quicker response times.

Cross-functional Collaboration

The collaborative efforts among data scientists, engineers, and product managers enable more effective problem-solving. Data scientists work on accuracy improvements while engineers manage deployment processes and product teams define meaningful outcomes. Through combined efforts this partnership guarantees the system achieves technical excellence while providing tangible business benefits.

Q.N.2. In the context of machine learning:

• Define and differentiate between overfitting and underfitting.

• Explain why both are problematic for model performance.

• Illustrate your explanation with simple examples (e.g., overfitting a training dataset, underfitting a complex pattern).

Answer:

In machine learning:
A model experiences overfitting when it memorizes training data intricacies including irrelevant noise and minor details. The system shows exceptional performance with training data yet fails to deliver satisfactory results when faced with new data sets.

Underfitting occurs when a model's simplicity prevents it from capturing data patterns. The system exhibits substandard performance across both training and test datasets.

Overfitting represents a scenario where the model extracts excessive information from the training set whereas underfitting indicates that the model fails to acquire sufficient knowledge. The intricacy of overfitted models stands in stark contrast to the simplicity found in underfitted models.

The performance of the model suffers from the detrimental impacts of both issues. The phenomenon of overfitting results in deficient generalization capabilities which renders the model untrustworthy for practical applications. Underfitting produces unreliable predictions alongside poor accuracy metrics.

Envision the construction of a mechanism for spam email detection. The overfitted model demonstrates its tendency to store specific expressions such as “Win a prize now!” and miss other spam that uses different words like “Claim your reward. An underfitted model tends to classify every message containing the word "offer" as spam which includes beneficial communications such as job offers. The model proves ineffective in both scenarios. An expertly developed model achieves equilibrium to identify new spam messages while avoiding excessive responses and maintaining focus.

Q.N.3. What are the main differences between normal Neural Network architectures and autoencoders? Explain in brief with example, How autoencoders can be applied in various contexts (problems at least one)? Explain how they help addressing these problems.

Answer:

A standard neural network processes input data to produce mapped outputs such as image classifications or numerical predictions. Through exposure to labeled data it acquires the ability to perform designated tasks. An autoencoder represents a distinct neural network category designed for unsupervised learning applications. The process involves reducing the input into a condensed version through encoding and subsequently restoring it to its original state via decoding.

Normal neural networks aim to make predictions whereas autoencoders concentrate on developing meaningful data representations.

Example:
An ordinary neural network functions to determine if an image depicts either a cat or dog. An autoencoder processes a cat image by compressing it into a smaller representation and then attempts to reconstruct the original image from this condensed form.

Application:
Autoencoders find extensive application in anomaly detection across domains such as fraud detection and medical imaging. The systems acquire an understanding of typical data patterns. When a new input fails to reconstruct correctly, it gets marked as abnormal. The ability to detect rare cases becomes possible even with limited examples through this method.

Autoencoders enable powerful problem-solving in real-world scenarios with unlabeled or imbalanced data by detecting patterns without labels.