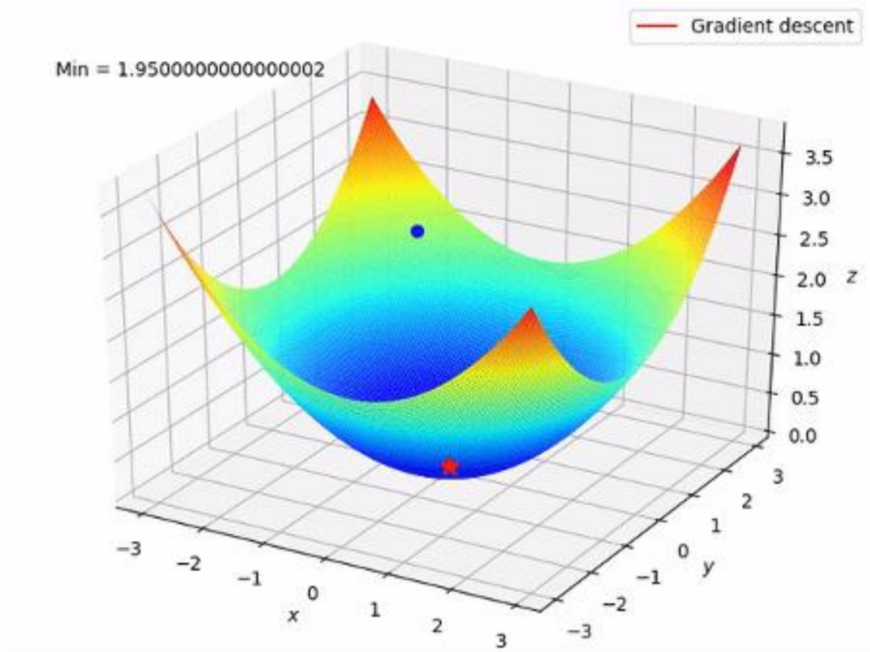
LIVERPOOL JOHN MOORES UNIVERSITY

A Comparative study of Optimizers for training Deep Neural Networks in Text Classification Tasks.



Shailendra Kumar
M.S. (Machine Learning & Artificial Intelligence)
Liverpool John Moores University
Oct 2023

Topics

Background of Study

•The wide-ranging applications of Natural Language Processing (NLP) in artificial intelligence have spurred significant research efforts aimed at enabling machines to comprehend and interpret human language.

•Notably, within the domain of text classification, numerous studies have contributed to the development of sophisticated algorithms and methodologies.

•The low performance of the widely employed Stochastic Gradient Descent (SGD) optimization algorithm in this context has underscored the need for a more comprehensive investigation into the intricacies of optimization techniques in NLP.

Literature Review

Introduction to NLP

- **Overview of the challenges and importance of NLP:**
  - Handling unstructured data, ambiguity, and context understanding.
  - Role of NLP in automating language-based tasks and facilitating communication.
- **Introduction to NLP sub-tasks:**
  - Text classification for categorizing textual data.
  - Sentiment analysis for understanding opinions and emotions.
  - Machine translation for enabling multilingual communication.

- **Impact of data availability and technological advancements:**
  - Growth of NLP with the increase in data availability.
  - Technological advancements leading to sophisticated NLP applications in various fields.

- **Contributions of text classification to various industries:**
  - Business: Market analysis, customer feedback analysis, and trend prediction.
  - Information Technology: Document organization and search optimization.
  - Healthcare: Patient record management and medical data analysis.

Literature Review

Text Preprocessing Techniques

- **Overview of essential preprocessing techniques:**
    - Tokenization: Breaking text into words or phrases.
    - Stop Word Removal: Eliminating common words with little semantic value.
    - Case Normalization: Converting all text to either lowercase or uppercase for consistency.

- **Importance of each step in preparing textual data for analysis:**
    - Ensuring uniformity in the text for accurate analysis.
    - Improving the efficiency of feature extraction and classification algorithms.

- **Trade-offs between stemming and lemmatization:**
    - Stemming: Faster but less accurate, may result in non-words.
    - Lemmatization: Slower but more accurate, provides meaningful lemmas.

Short Overview of Literature Review

Feature Extraction Techniques and Word Embeddings

•Count Vectorizer transforms text into a matrix, providing a simple representation, yet it lacks semantic understanding.

•TF-IDF (Term Frequency - Inverse Document Frequency) efficiently combines term frequency and document rarity.

•Word embeddings, such as Word2Vec, capture semantic and contextual word meanings, enhancing NLP tasks significantly.

•Word2Vec employs CBOW and Skip-gram architectures, enabling effective word mapping in vector space.

Short Overview of Literature Review

Text Classification Algorithms and Classification Approaches

Algorithms Process of Development:

- Feature Extraction, Training & Model Building, Testing & Evaluation, Inference, and Fine-Tuning & Deployment are key steps in text classification algorithms.

Classification of Algorithms:

- Classification techniques encompass Machine Learning Algorithms (Statistical Models, Ensemble Methods, Rule-Based Models, and Distance-Based Algorithms) and Deep Learning Architecture (Feedforward Neural Networks, CNNs, RNNs, and Transformers).
- These algorithms differ in data availability, classification types, approach, text data representation, and interpretability.
- Ensemble Techniques like Bagging and Boosting contribute to enhancing model performance.

Model Evaluation Parameters:

- Evaluation metrics like Accuracy, Precision, Recall, F1-Score, and Classification Report aid in the comprehensive assessment of model performance in text classification tasks.

Literature Review

Optimization in NLP Classification Tasks

1. Role of Optimizers:
   1. Adjust model parameters during training.
   2. Impact convergence speed, stability, and performance.

2. Key Functions of Optimizers:
   1. Parameter Updates based on gradients.
   2. Influence Convergence Speed and Stability.
   3. Adaptability and Hyperparameter Tuning.

3. Transfer Learning and Generalization:
   1. Optimizers influence model generalization to new data.
   2. Adaptive optimizers ensure better generalization performance.

LIVERPOOL JOHN MOORES UNIVERSITY

Problem Statement

•The continuous evolution of deep learning methodologies in Natural Language Processing (NLP) has brought forth a pressing challenge regarding the comprehensive understanding of the efficacy of available optimization algorithms and their implications for training deep neural networks in text classification tasks.

•The low performance of the Stochastic Gradient Descent (SGD) classifier algorithm have highlighted the need for an extensive exploration of alternative optimization techniques tailored to the nuances of NLP applications.

•Understanding on suitable optimization technique can significantly help in saving time, energy and compute resources, in other word It will also help in preserving environment.

Problem Statement

Aim:

The primary aim of this study is to comprehensively evaluate the efficacy of various optimization algorithms, available in Keras Library, in the training of deep neural networks for text classification tasks.

Objective:

•Objectives: Design and implement a comprehensive experimental framework for evaluating the performance of optimizers in training various deep learning models for text classification.

•Assess and compare the convergence dynamics, generalization performance, and computational efficiency of the identified optimizers through rigorous experimentation and analysis.

•Provide actionable insights and recommendations for practitioners and researchers regarding the selection and application of optimization algorithms for text classification tasks, thereby contributing to the broader understanding of optimization methodologies in the field of Natural Language Processing.

Research Methodology

Data Pre-Processing:

Steps like text-normalization, removal of punctuation, symbols non-alphanumeric characters, URLs, Additional process like stop word removal, Lemmatization.

Feature Engineering:

Creation of Word Embeddings for training Deep Neural Networks.

Model Selection:

Basic Neural Network, RNN with LSTM & RNN with GRU

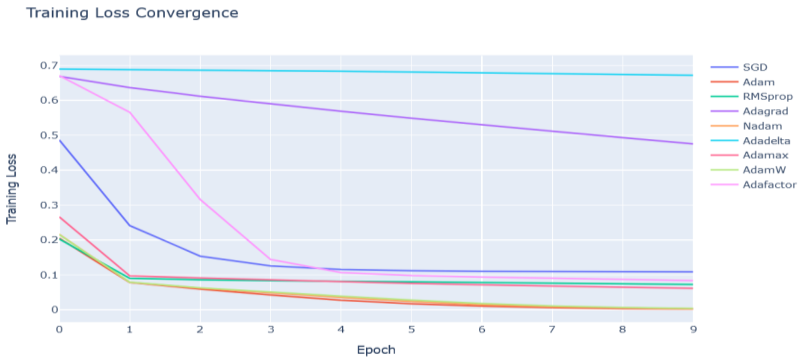
Hyperparameter Tuning:

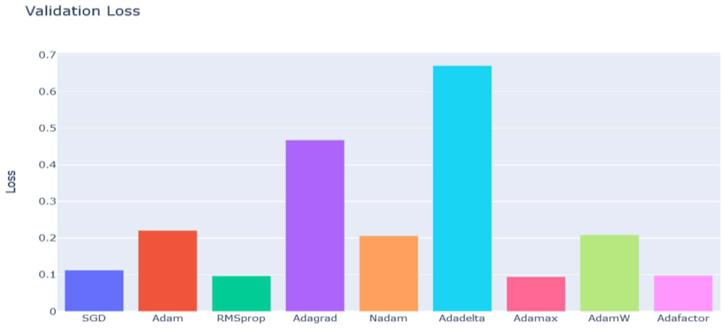
Optimizer parameters tuned for performance and training on same Basic Network.

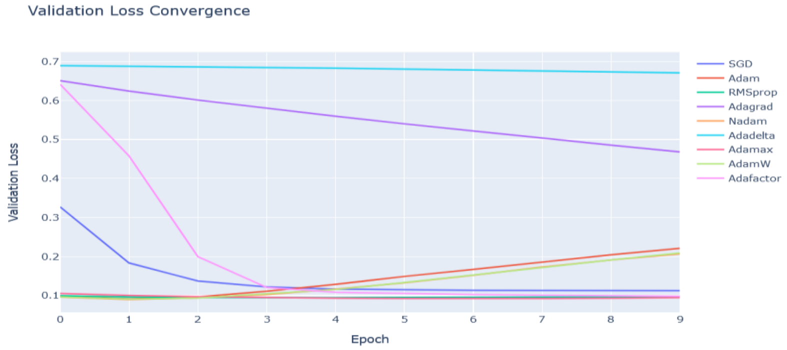
Evaluation Metrics:

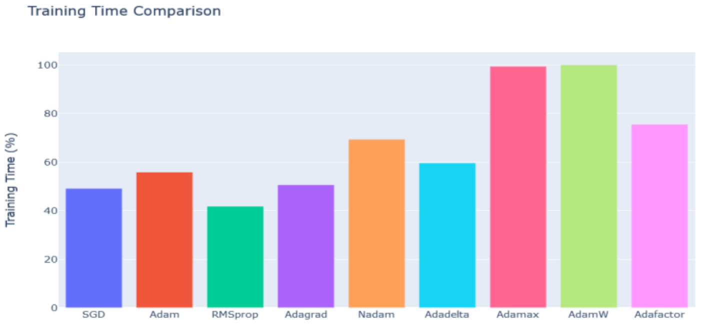
Monitoring Training & Validation Loss along with change in Accuracy.
Training time comparison.
Individual analysis of Training curve for each of optimizers.

Results & Discussion









Results & Discussion

From the graphs (Training Loss) and study, In General:

- **Adadelta & Adagrad** showed nearly a flat loss curve. It can be concluded that these optimizers failed to train these models well and performed very poorly since no significant training happened.

- **Adafactor** optimizer started with high loss value then showed significant reduction in training loss initially and then stabilized.

- **SGD, RMSprop, Adamax** started with moderate loss values and loss reduced in initial training phase and then model stabilized with slow reduction in loss values.

- **Adam, AdamW & Nadam** proved to be best optimizers since they started with quite less initial loss and good learning happened in early training phase and then afterwards also loss value kept reducing.

Results & Discussion

From the graphs (Validation Loss) and study, In General:

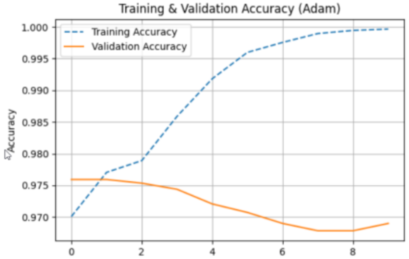
- **Adadelta & Adagrad** performed poorly in validation loss also since it is evident that no significant training happened with these optimizers.

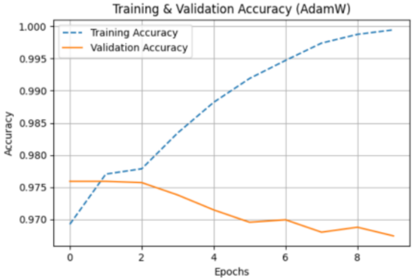
- **Adafactor** validation loss was quite high initially but decreased fast with training then stabilized. This shows that the model can generalize on unseen data, and it is able to learn the patterns in data.

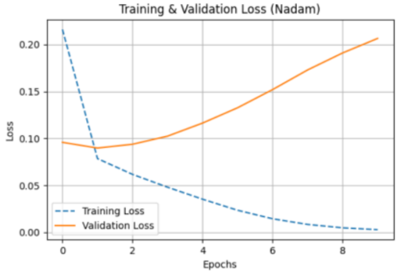
- **SGD, RMSprop, Adamax** achieved less validation loss since starting and kept nearly same values till end of training. This shows that models trained with these optimizers have good generalizing ability and this ability didn't face any change due to further training.

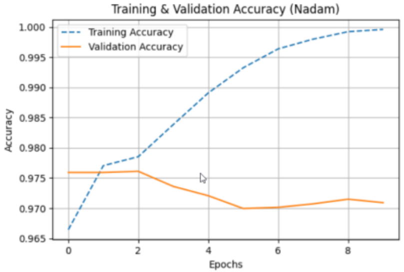
- **Adam, AdamW & Nadam** started with less validation loss values, which further reduced but after some time validation loss started increasing consistently. This shows that with further training models lost their ability to generalize and started overfitting. Early stopping should be employed in this case.

Results & Discussion









Adam, AdamW & Nadam their curves shown that these algorithms are very efficient in learning and learning happened very quickly in initial steps itself and further training lead to overfitting only. With Early Stopping these algorithm will perform really well on text classification tasks.

Conclusion

Detailed Research Analysis

•Comprehensive evaluation involved training four distinct deep learning models with various optimizers.

•**Adam, AdamW, and Nadam** demonstrated superior convergence and generalization performance.

•These optimizers exhibited stability, reduced fluctuations in training and validation losses, and faster convergence.

•Efficient training times were observed with Adam, AdamW, and Nadam, without compromising model performance.

•The study showcased the pivotal role of adaptive learning rates in capturing intricate textual patterns and nuances.

•Findings indicate that **Adam, AdamW, and Nadam** are optimal for text classification tasks, expediting training without compromising predictive power.

Future Work

Exploring Advanced Architectures and Optimizers
Study the interplay between Adam, AdamW, and Nadam with advanced models like BERT and GPT.

Diverse and Challenging Datasets
Incorporate challenging datasets with linguistic nuances, noisy text, and domain-specific complexities. Evaluate optimizers' performance across a spectrum of real-world text classification scenarios, ensuring robustness and adaptability.

Comprehensive Evaluation with Diverse Textual Data
Conduct a holistic analysis across multiple diverse textual datasets, spanning different domains, genres, and languages.

Hyperparameter Optimization Strategies
Investigate the impact of hyperparameter settings on the optimization process, offering insights into fine-tuning strategies for optimal performance.

LIVERPOOL JOHN MOORES UNIVERSITY

Thank You Very for Your Precious Attention & Time !!