SEQUENTIAL ARCHITECTURE EXPERIMENTATION FOR TEXT CLASSIFICATION

Jamia Russell[1,†]

[1] Northwestern University School of Professional Studies
Master of Science in Data Science Program
633 Clark St. Evanston, IL 60208

Address to which correspondence should be addressed:
jamialashe@gmail.com

Abstract

To examine the effectiveness and efficiency of neural networks classification using Natural Language Processing (NLP) and deep learning techniques, different network topologies are required. Using the AG News data set the models placed through sequential experimentation are varied in hyperparameter settings and model architectures. These model architectures span from Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short Term Memory (LSTM) models, to Dense Neural Networks (DNNs). Architecture comparison based on accuracy, processing efficiency, and classification performance inform model selection for NLP tasks and text and conversational AI systems.

Deep learning techniques batch normalization, regularization, diverse optimizers, and bidirectional RNNs are utilized to alter network structure to depict the most efficient network in classifying images. Specifically, max pooling, dropout layers, L2 regularization, Adam, RMSprop and SGD optimizers, ReLU and Softmax activation functions are used.

This paper explores performance and accuracy to determine the effects of changing network architecture during sequential experimentation of AG News data classification by topic.

The models were created and examined using Python TensorFlow and Keras packages that offer examination through model layer modification. Results found that dense neural network (DNN) models compared to differing architectures perform best in classification and computational processing time..

Keywords: Neural Network Topologies, Hidden Layer Nodes, AG News Text Classification, Tokenization

Table of Contents

I. Introduction & Problem Statement

Existing text processing systems such as sentiment analysis, conversational AI, and topic categorization across multiple interfaces prove determining optimal network topology for fast and efficient classification advantageous for commercial and everyday use. Most notably, OpenAI's ChatGPT has established itself as an instrument to daily life averaging 300 million active users per week and processing over 1 billion queries per day (Jha, 2025). Most AI assistance is not reflective of image and number classification, in most cases it is interaction with questions or prompts relayed looking for an answer or solution to be pulled from differing sources.

For example, ChatGPT has decreased employee task completion time by 40%. A MIT study examining CHatGPTs impact on the workforce found that CHatGPT use in white collar work increased speed of work completion. Tasks include email and memo drafting, slide deck creation, and speech flow editing. It is expected that use of ChatGPT and other competing artificial intelligence text processing systems will only increase across many industries.

Using the AG News data set to design varying network topologies with adjusted hyperparameters and deep learning techniques to predict and classify with accuracy textual data of news articles across four classes bidirectional RNN and long short term memory (LSTM) are pertinent. Individually these techniques assist with vanishing gradients and text context allowing for models to learn dependencies on text and designate word meaning.

An existing challenge within conversational AI relating to chat bot assistants in the workplace is encompassing the scope of potential issue that may arise. The assistant will have to

be properly trained to include intents and model architectures reflective of user language as it can vary by user knowledge.

II. Literature Review

Text Classification

As a fundamental function of Natural Language Processing (NLP) text classification and processing into existing labels is necessary in every NLP task. Traditional machine learning methods and deep learning methods have allowed for differing processing approaches that can be tailored to the input data based on context and word choice. For example, traditional methods such as decision trees, support vector machines, and naive bayes compared to deep learning techniques convolutional neural networks, recurrent neural networks and transformer models require statistical techniques and handmade features. Deep learning allows for a more computationally efficient model that learns through model training.

Uysal and Gunal, (2014) examined classification accuracy through extensive preprocessing techniques. Preprocessing techniques such as tokenization, stop-word removal, lower case conversion and stemming in varying degrees were used to compare the effect on classification across English and Turkish textual data across two different domains - emails and news. The study found that compared to individual text preprocessing methods a combination proves more effective in classifying textual data. However, results depend on domain and language where particular combinations where words are derived from may present significant improvement in classification accuracy.

Displaying proficiency in image classification convolutional neural networks (CNNs), and recurrent neural networks (RNNs) ability to provide exemplary computational efficiency and

accuracy with textual data is often overlooked. Cai et al., (2018) examined RNN and CNN performance in text classification across news domains encompassing sports, social media and entertainment, and international and domestic political news. Models revealed CNNs to have the highest precision accuracy at 85.34% compared to RNNs at 82.73%. RNN model inability to parallelize text making it more suitable for short text processing.

Domain Variation and Text Classification

Existing research on text classification presents model accuracy and computational efficiency in text classification. A consistency across all model experimentation is the difference in results based on model architecture and text domain. Model experiments span across healthcare, social media, feedback analysis, and legal documentation depicting differing results with similar deep learning preprocessing and model structures applied.

Chen et al. (2022) investigated domain specific model performance on word embedding and information processing within the architecture, engineering and construction (AEC) industry. Models were pre-trained domain specific models with static and context word embedding based on the domain corpora. To compare results the corpora was also ran through non-domain trained models. Experiment results found that for text classification tasks domain corpora can increase both static and contextual word embedding DL models increasing accuracy by 11.4% and 6.4%, respectively.

Conversely, to address this existing discrepancy in model performance by domain and model training Cardie & Chen (2022) construct a multinomial adversarial network (MAN) to interact with data that reflects real world multi-domain data. Model creation and experimentation

was viewed from two perspectives, domain adaptation and multi-domain text classification. Balancing the idea of models trained on large and complex training data and labeled data for multiple domains exists the aim is to leverage all resources to increase system efficiency and accuracy over the domains. The MAN's architecture prioritizes domain invariant and shared features that contribute to classification. MAN experiments outperform existing domain adaptation and MDTC systems depicting feature extraction that can distinguish domain invariant and shared features increasing classification accuracy across domains with unlabeled data.

III. Methods

Data Overview

To examine text classification using deep learning and natural language techniques, the AG News topic classification dataset was used. This data set consists of 120,000 training and 7,600 test samples of news articles across four categories including Business, Science/Technology, Sports and World. Each class contains 30,000 training and 1,900 testing samples. The complexity, yet ease of use provides a benchmark for experimenting and analysis in model classification.

In this model experimentation, there is a need for diverse and accurate data for training models and providing a comprehensive analysis of model and network performance which informs the accuracy of model experiments. To interact with this data Python packages Pandas, NumPy, Matplotlib, TensorFlow, Keras, and SciKit-Learn were used for data model construction, training, evaluation, and visualization.

Preparing the AG News data set for model creation and experimentation several preprocessing steps took place. Removal of punctuation and special characters and conversion of

all text to lowercase prepared the data for tokenization where words were transformed into integer sequences. Standardization using sequence padding fixed all text to a length of 128 tokens. From there data was split into training, validation, and testing sets and exploratory data analysis displayed class distribution and determined the most frequently used words across categories.

Model Construction and Experimentation

To develop a comprehensive analysis and assessment of deep learning and natural language processing techniques neural network architectures were evaluated. The foundational model includes a dense neural network with an embedding layer with 64 dimensional vectors and fully connected 128 and 64 neuron hidden layers with ReLU activation. To mitigate overfitting L2 regularization and a dropout layer were used following each hidden layer. The output layer contained four neurons and implemented the softmax activation function and an Adam optimizer.

The recurrent neural network model processed text sequentially allowing for contextual dependencies to be identified, and embeds input words into dense vectors. A LSTM layer with 128 and 256 units and a hidden layer with 64 neurons with ReLU activation followed. The .3 dropout rate in the LSTM layer refined data and prepared the output layer with softmax and RMSprop to provide class probabilities.

Model architecture advanced including a long short term memory (LSTM) network and bidirectional long short term memory (biLSTM) to both capture long term dependencies and factor text past and future context. Doing this by processing in forwards and backwards directions the bidirectional LSTM layer consisting of 128 units and a .3 dropout rate, followed by a dense layer of 64 neurons with ReLU activation, and Adam and RMSprop optimizers. The

model controlled randomness to the network's activation and the final dropout layer and softmax output layer refined text.

Continuing, a one dimensional convolutional neural network (CNN) was created with a Conv1D layer containing 128 filters and kernel size variation of 3 and 5, and SGD and Adam optimizers. A max pooling layer reduced dimensionality, then a dense layer of 64 neurons, ReLU activation and dropout flattened features. Lastly, the softmax layer provided class probabilities. Building from this structure

Training and Evaluation

Models were trained across 10 epochs with a batch size of 100 to optimize training efficiency. The pertinent measures of effectiveness were accuracy, time for computational processing, and loss metrics.

IV. Results

The first experiment, the dense neural network shows rapid learning with an initial improvement with accuracy and loss across the first two epochs. Increasing from an accuracy of 51% and loss of 1.3829 to 86.77% and a loss decrease to .5581. Continuing across the epochs, training accuracy reaches a high of 93% and validation loss reaches its lowest point .4609. Later epochs depict a performance plateau, accuracy and validation loss fluctuation suggest overfitting and issues with generalization due to training patterns and interaction with unseen data. Despite this, we do see the effectiveness of dropout mid model where validation loss increase is slower.

Results of the second experiment, a recurrent neural network model, do not provide evidence of RNN efficacy of the text classification task. With a model accuracy that remains between 20-30% across the 10 epochs there aren't many meaningful patterns to pull from the data. In addition, results present fluctuation in validation loss and accuracy suggesting the model has limited ability to learn rather than issues with overfitting. Despite the inclusion of the LSTM layer the RNNs perform poorly, vanishing gradients may be causing improper weight updates.Including other deep learning techniques such as batch normalization, increased dropout, and other layer additions may help optimize model strategy and provide effective results.

Contrarily, BiLSTM with an Adam optimizer model results present accuracy, loss, and performance metrics performing very well. Consistent increases across 10 epochs with training accuracy reaching a high of 94.53%, validation accuracy wavering between 90-91%, and validation losses decreasing from .6751 to .1516 inform the model's learning capacity. Despite this, the rise in validation loss indicates overfitting. Test set accuracy at 89.52% and validation loss .3341 supporting the indication of overfitting. An increase in dropout rate would encourage early stopping and prevent validation loss increases.

Similarly, a BiLSTM model with RMSprop optimizer presented model efficiency over the 10 epochs. Training accuracy increased over 25% from 65.28% at the first epoch to 92.96% at the final epoch. A consistent decrease in training loss from.7986 to .2104, and validation loss .3412 to .2585 proves the model is learning. Model test accuracy reaching a high of 90.64% with a loss of .2789 overfitting seems to be a non-issue. Despite the accurate performance, training time does increase across epochs and suggest there are factors that are slowing down

computational efficiency. Potential mitigation includes reducing batch size, and varying optimizers.

CNN model performance with a size 5 kernel and SGD optimizer shows gradual increases across metrics over the 10 epochs. At the first epoch the model presents low training accuracy at 26.59% and loss of 1.3855. This result changes over the epochs as the model learns, reaching 67.64% at the tenth epoch. Decreases in training loss from 1.3855 to .7990, and validation loss from 1.3824 to .7130 for validation presents the model's optimization skill. This is due to the SGD optimizer's slower convergence. Validation and loss on the test set at 73.03% and .7628 respectively, indicate the model's ability to prevent overfitting. In addition, speedy epoch run times prove computational efficiency and fast training times.

CNN model performance with a size 3 kernel and Adam optimizer presents faster convergence and an initial high accuracy of 72.03% training accuracy in the first epoch. By the tenth epoch training reaches 98.82% depicting the model efficiency compared to the SGD optimizer. Comparatively, validation accuracy peaks at 90.68% and lowers to 88.86% at the tenth epoch. Validation loss increases over the 10 epochs from .2657 to .6857 indicating overfitting in the training data. Model results have room for improvement. Inclusion of dropout and L2 regularization techniques would help the model with overfitting.

Lastly, a BiLSTM model with batch normalization is evaluated showing proficient performance. Initial epochs show a training accuracy of 73.07% and reaching a high of 94.61% at the tenth epoch. Validation accuracy reaches a high in early epochs then plateaus at 90.40% Validation loss performance reveals minimal changes indicating the model generalizes well.

Regardless of performance metrics, processing time of several hours proves the model to be inefficient for training and use.

V. Conclusion

Experiments evaluating text classification performance by architecture offer lessons on patterns with accuracy, loss behavior, training efficiency, and generalization. The dense neural network with a .5 dropout layer displayed the effects of regularization, however the model does struggle with overfitting and generalization. RNN models show to be ineffective in text classification with these hyperparameters. Both varying unit models provide a training accuracy below 30%, the model is unable to learn meaningful representations from the data. In addition loss increases reveal the models suffer from vanishing gradients, with extensive processing times. CNN models with differing kernel sizer and optimizers present an overall higher validation and test accuracy compared to models examined at this point in experimentation. SGD optimizer convergence speed led to effective performance in early epochs, however towards later epochs there is an issue with model generalization. Bidirectional LSTM models are able to capture backward and forward contextual information proving them to be suitable for textual information and sequences. Despite that, extensive processing time disqualifies the model as optimal architecture for text classification. The sense neural network performance with relatively high accuracy, appropriate computational processing time, and room for hyperparameter tuning is the best choice.

On the contrary, a conversational agent used for customer support architecture would be best suited for a hybrid CNN and LSTM model. The high classification accuracy and lower processing times make it appropriate for use of user interactions. To facilitate use, the chat bot

requires intent identification, context aware responses, and scalability to perform such task. Source code will be necessary to define these and implement stories and actions the assistant will follow after receiving user inquiries. From there, a could based infrastructure would allow for proficiency in dealing with real life interactions and providing accurate responses. It is also imperative to include means for user feedback and model retraining to meet user needs and fulfill requests.

References

ag_news_subset | TensorFlow Datasets. (n.d.). TensorFlow. https://www.tensorflow.org/datasets/catalog/ag_news_subset

Cai, J., Li, J., Li, W., & Wang, J. (2018). Deeplearning Model Used in Text Classification. *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. https://doi.org/10.1109/iccwamtip.2018.8632592

Chen, X., & Cardie, C. (2018). Multinomial Adversarial Networks for Multi-Domain Text Classification. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1802.05694

Jha, V. (2025, February). *ChatGPT User Statistics and Market Performance: February 2025 Update - Aitechtonic*. AiTechtonic - Informative & Entertaining Text Media. https://aitechtonic.com/chatgpt-user-statistics/

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, *50*(1), 104–112. https://doi.org/10.1016/j.ipm.2013.08.006

Papers with Code - AG News Dataset. (2019). Paperswithcode.com. https://paperswithcode.com/dataset/ag-news

Winn, Z. (2023, July 14). *Study finds ChatGPT boosts worker productivity for some writing tasks*. MIT News | Massachusetts Institute of Technology. https://news.mit.edu/2023/study-finds-chatgpt-boosts-worker-productivity-writing-0714

Zheng, Z., Lu, X.-Z., Chen, K.-Y., Zhou, Y.-C., & Lin, J.-R. (2022). Pretrained domain-specific language model for natural language processing tasks in the AEC domain. *Computers in Industry*, *142*, 103733. https://doi.org/10.1016/j.compind.2022.103733

https://arxiv.org/pdf/1802.05694

https://www.tensorflow.org/datasets/catalog/ag_news_subset

Appendix

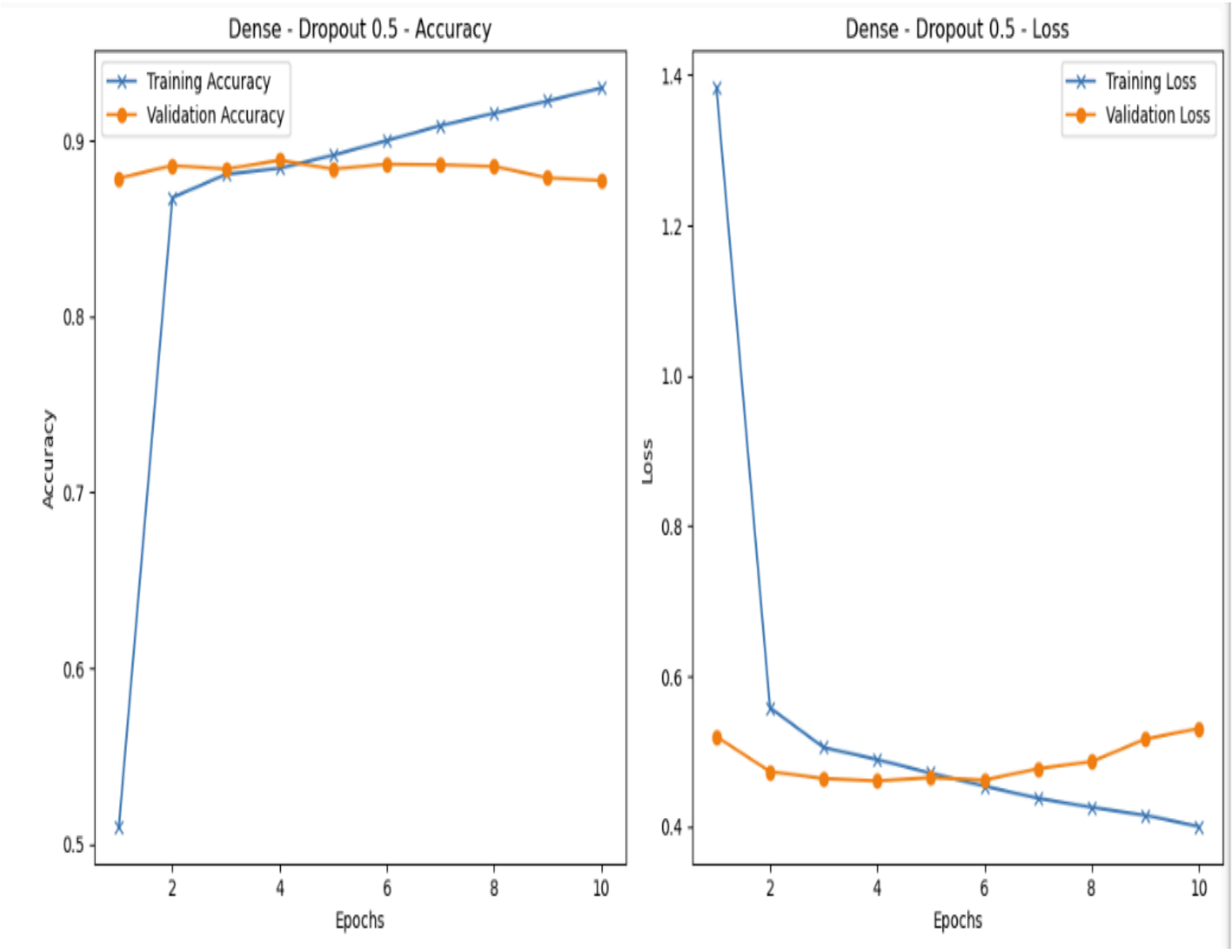
Model Result Data Frame

	Model	Train Accuracy	Validation Loss	\
0	Dense - Dropout 0.3	0.516108	0.886842	
1	Dense - Dropout 0.5	0.553883	0.873816	
2	RNN - 128 units	1.412246	0.250000	
3	RNN - 256 units	1.386312	0.250000	
4	BiLSTM - Adam	0.335320	0.894737	
5	BiLSTM - RMSprop	0.283088	0.906316	
6	CNN - Kernel 5, SGD	0.730651	0.726842	
7	CNN - Kernel 3, Adam	0.684650	0.885658	
8	BiLSTM - BatchNorm	0.332643	0.900395	

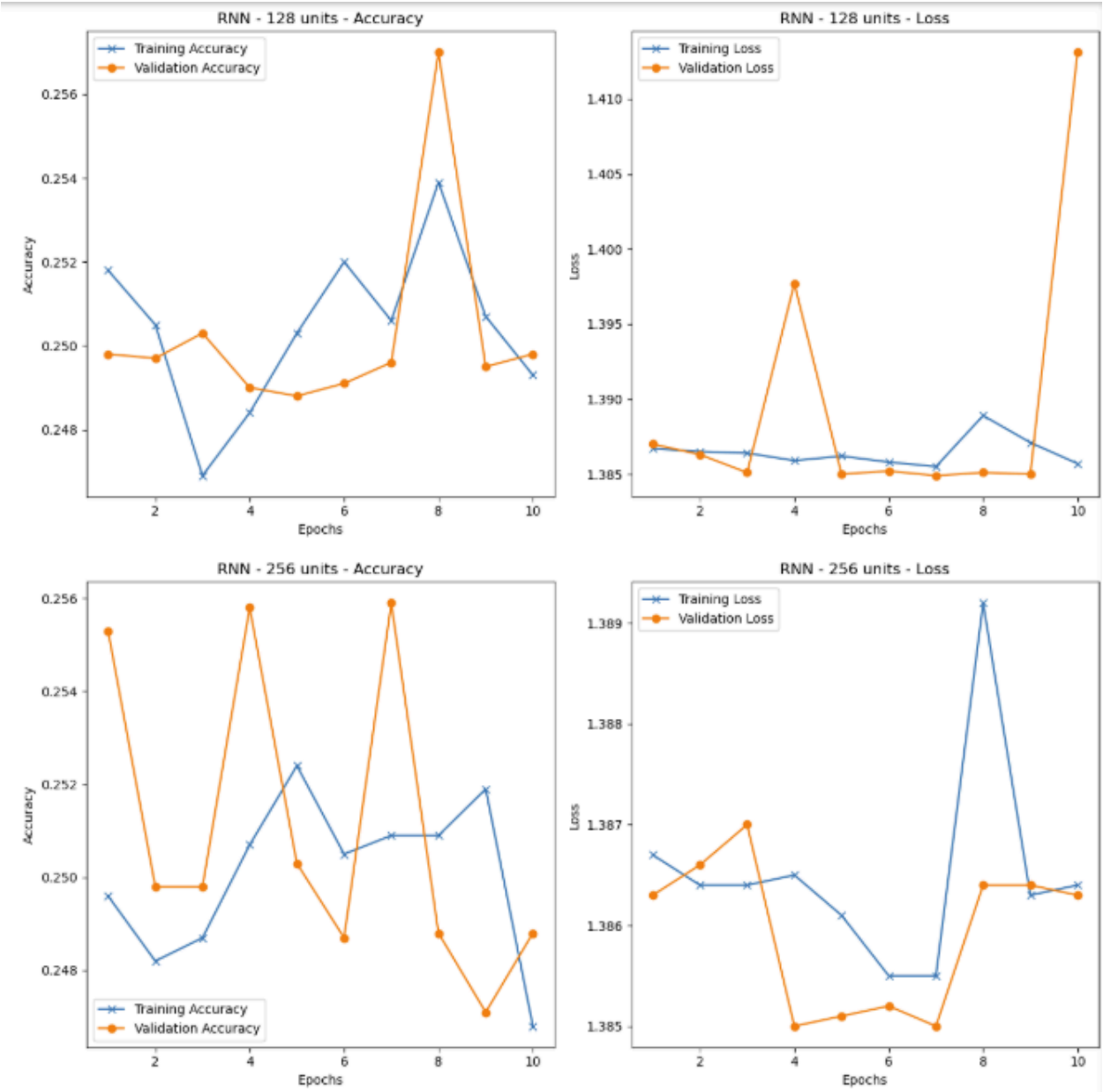
	Train Accuracy (experiment)	Validation Accuracy (experiment)	\
0	0.813593	0.892324	
1	0.681407	0.868852	
2	0.250194	0.250963	
3	0.249315	0.248120	
4	0.834750	0.903139	
5	0.798407	0.896769	
6	0.274065	0.302639	
7	0.840407	0.910287	
8	0.835704	0.903028	

	Processing Time (s)
0	1260.342844
1	157.559771
2	1441.206192
3	3080.122847
4	2834.701091
5	10058.485137
6	737.644118
7	749.594133
8	51904.231493

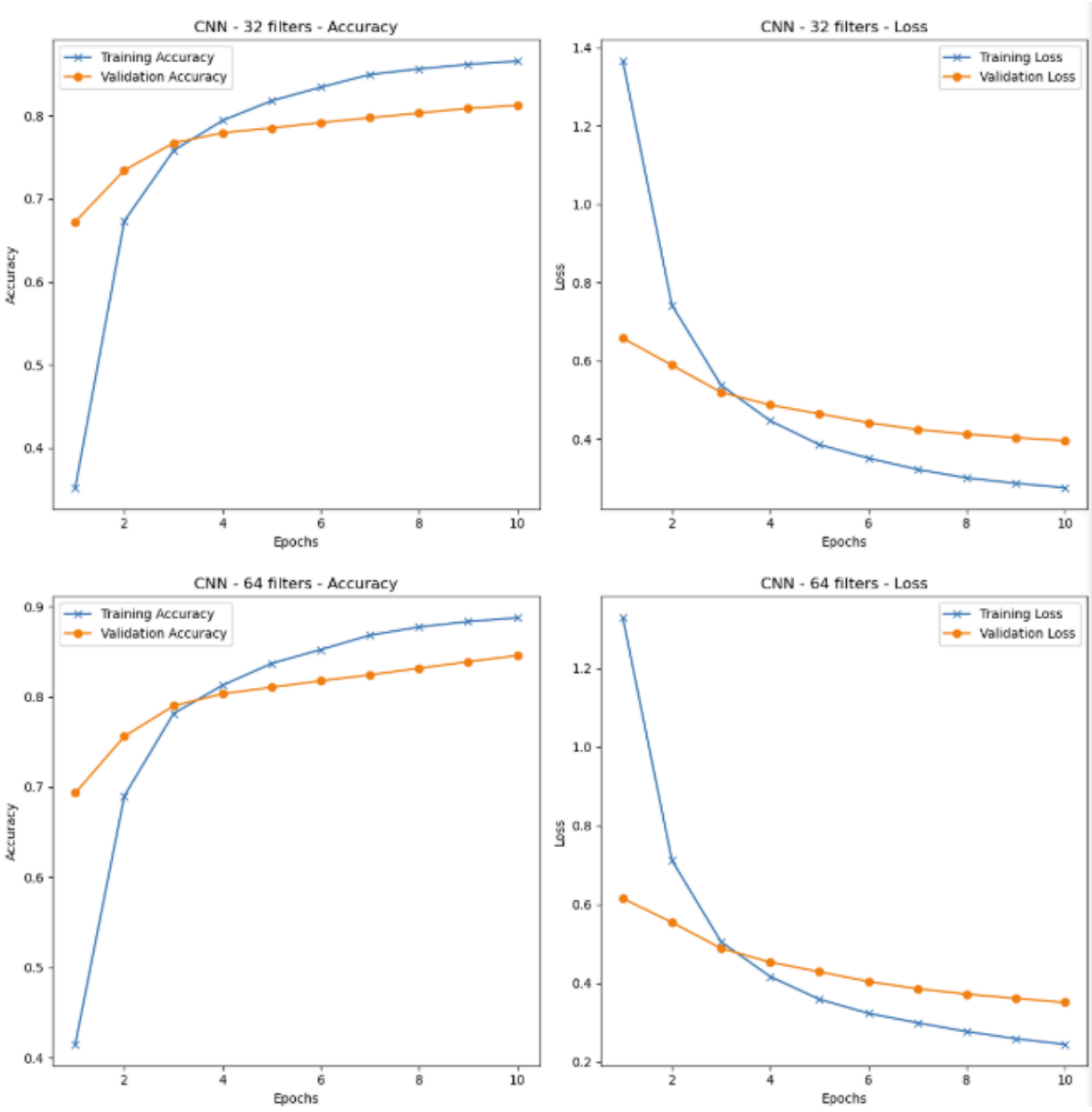
DNN Results


Dense - Dropout 0.5 - Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epochs
Dense - Dropout 0.5 - Loss
Training Loss
Validation Loss
Loss
Epochs

RNN Experiment Results



CNN Experiment Results



BiLSTM Experiment Results