Fake-News Classifier

A machine learning model to classify news articles as true or fake

Yocheved Ofstein, Shay Gali, Shalom Ofstein

Motivation

The rise of fake news impacts public trust and decision-making.

Goal: Classify news articles as **"true"** or **"fake."**

Focus: Use textual features for accurate classification.

Approach: Binary classification with advanced ML techniques.

Dataset Description

Sources:

Labeled news articles from true and fake news datasets.

- Kaggle "fake news" dataset

- Kaggle "WELFake" dataset

Features Used:

Processed text content (title, subject category, and publication date excluded).

Preprocessing:

Tokenization.

Removal of Class-exclusive words

Word2Vec embeddings for semantic representation.

Data Split:

70% training

15% validation

15% testing.

Model Frameworks

- **Baseline Model:**
  - Majority Class Classifier predicts the most common class.
- **Logistic Regression:**
  - Uses feature weights to classify articles as "true" or "fake."
- **Fully Connected Neural Network (FCNN):**
  - Captures non-linear patterns in text data.
- **Advanced Model:**
  - Incorporates Bi-LSTM, attention mechanisms, and dropout regularization.

Baseline Model

Baseline Model

Approach:

- Majority Class Classifier predicts the most frequent class ("true news").

Key Insights:

- Simple benchmark to evaluate improvements.
- Fails to address class imbalance or detect "fake news."

Baseline Model

Performance Metrics:

Metric:	Score:
Accuracy	51.5%
Precision (True News)	51.5%
Recall (True News)	100%
Recall (Fake News)	0%

Logistic Regression Model

Logistic Regression- Key stages:

Preprocessing:

Transforming raw text into numerical features.

Model Training:

Fitting the model using transformed data.

Logistic Regression- Preprocessing Steps

Eliminated dataset bias by removing the term **"Reuters"**, which skewed early results.

Removed duplicate entries from the dataset

Used **TF-IDF Vectorizer**:

- Extracted up to 5,000 features.
- Removed common stopwords for better focus.

Logistic Regression- Training Details

- Logistic Regression trained with:
  - **Maximum Iterations:** 1,000 for convergence.
  - **Refined Feature Representation:** Switched from Count Vectorizer to TF-IDF Vectorizer, improving validation accuracy from 90% to 92.9%.

Logistic Regression- Key Insights

- Initial results of 100% accuracy exposed dataset bias.
- Adjusting preprocessing revealed realistic performance metrics.
- Highlighted the importance of robust feature engineering and evaluation.

Performance Comparison:

Metric:	Baseline Model:	Logistic Regression Model:
Accuracy	51.5%	92.9%
Precision (True News)	51.5%	93%
Recall (True News)	100%	94%
F1-Score (True News)	67.95%	94%
Precision (Fake News)	N/A	93%
Recall (Fake News) 0%	0%	91%
F1-Score (Fake News)	N/A	92%

Neural Network Model

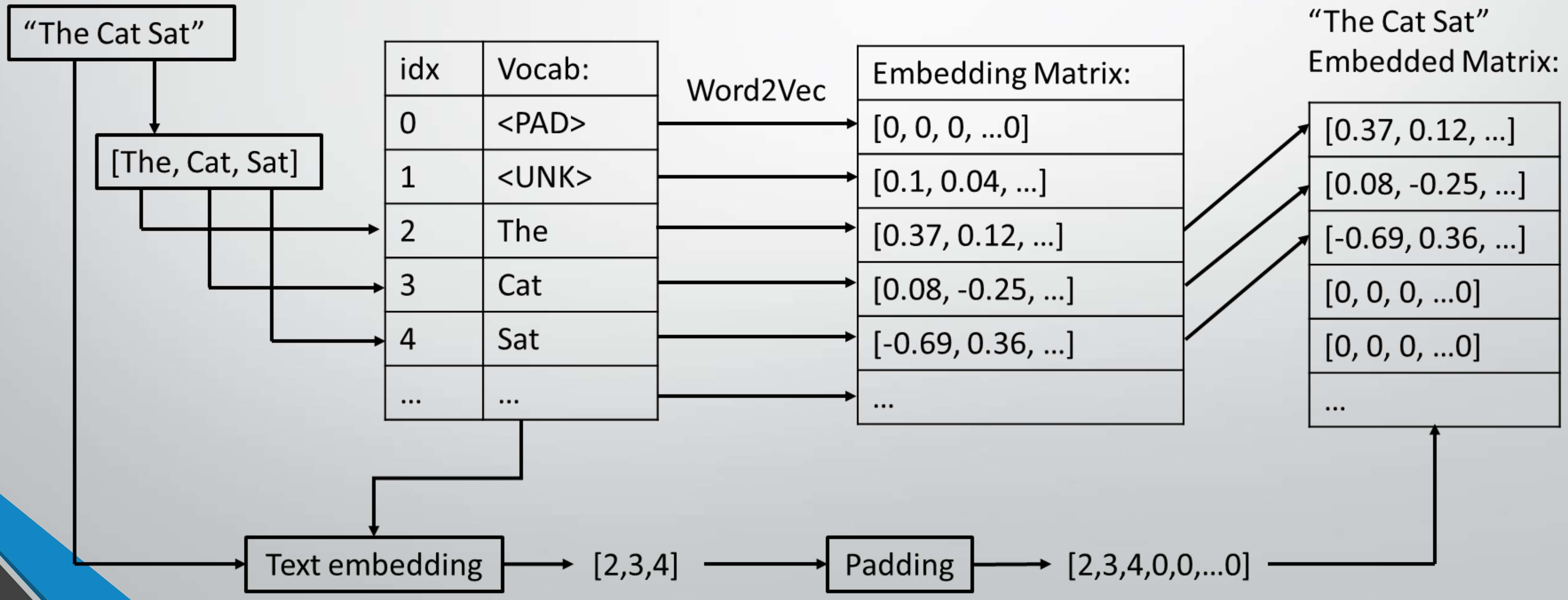
Fully Connected Neural Network (FCNN) Model

FCNN- Key stages:

- Preprocessing
- Model architecture
- Model Training

Fully Connected Neural Network

Preprocessing



Fully Connected Neural Network

Model Architecture

Hidden Layers:

- 3 Fully Connected Layers (512 → 256 → 128 units).
- ReLU activation for non-linearity.

Regularization:

- Dropout (50% for input, 20% for hidden layers).
- Layer normalization for stable training.

Output Layer:

- Sigmoid activation for binary classification

Fully Connected Neural Network

Model Architecture

Input Text	The news article
Embedding Layer	Converts word indices into dense, 300-dimensional vectors using pre-trained Word2Vec embeddings from 'GoogleNews vectors'
Flatten Layer	Flatten the embedding matrix to a 1-dimension vector
Hidden Layer 1	Connected with a ReLU Dropout Layer of 50%
Hidden Layer 2	Connected with a ReLU Dropout Layer of 20%
Hidden Layer 3	Connected with a ReLU Dropout Layer of 20%
Output layer	The final classification in a percentage using a sigmoid activation

Fully Connected Neural Network

Model Training

Loss Function:	Optimizer:	Learning Rate Strategy:	Training Parameters:
• Binary Cross-Entropy with class weights to address imbalance.	• AdamW with weight decay for better regularization.	• Warmup over initial steps. • Dynamic adjustment with ReduceLROnPlateau scheduler.	• Batch size: 32 • Epochs: Max 15 (Early stopping after 4 epochs of no improvement).

Initial Challenges and Fixes

Challenge: Exploding gradients in early training.	**Fix:** Applied gradient clipping to stabilize updates.
Challenge: High memory usage with embeddings.	**Fix:** Freezing pre-trained Word2Vec weights reduced memory load.
Challenge: Batch inconsistencies during processing.	**Fix:** Added compatibility checks for batch sizes.

FCNN Model - Key Insights

- **Initial Results:** Improved performance over Logistic Regression.
- **Key Features:**
  - Captures nuanced patterns in text data.
  - Regularization prevents overfitting, ensuring better generalization.

Performance Comparison:

Metric:	Baseline Model:	Logistic Regression Model:	Neural Network Model:
Accuracy	51.5%	92.9%	94.36%
Precision (True News)	51.5%	93%	94.95%
Recall (True News)	100%	94%	94.73%
F1-Score (True News)	67.95%	94%	94.84%
Precision (Fake News)	N/A	93%	93.66%
Recall (Fake News) 0%	0%	91%	93.92%
F1-Score (Fake News)	N/A	92%	93.79%

Advanced Model

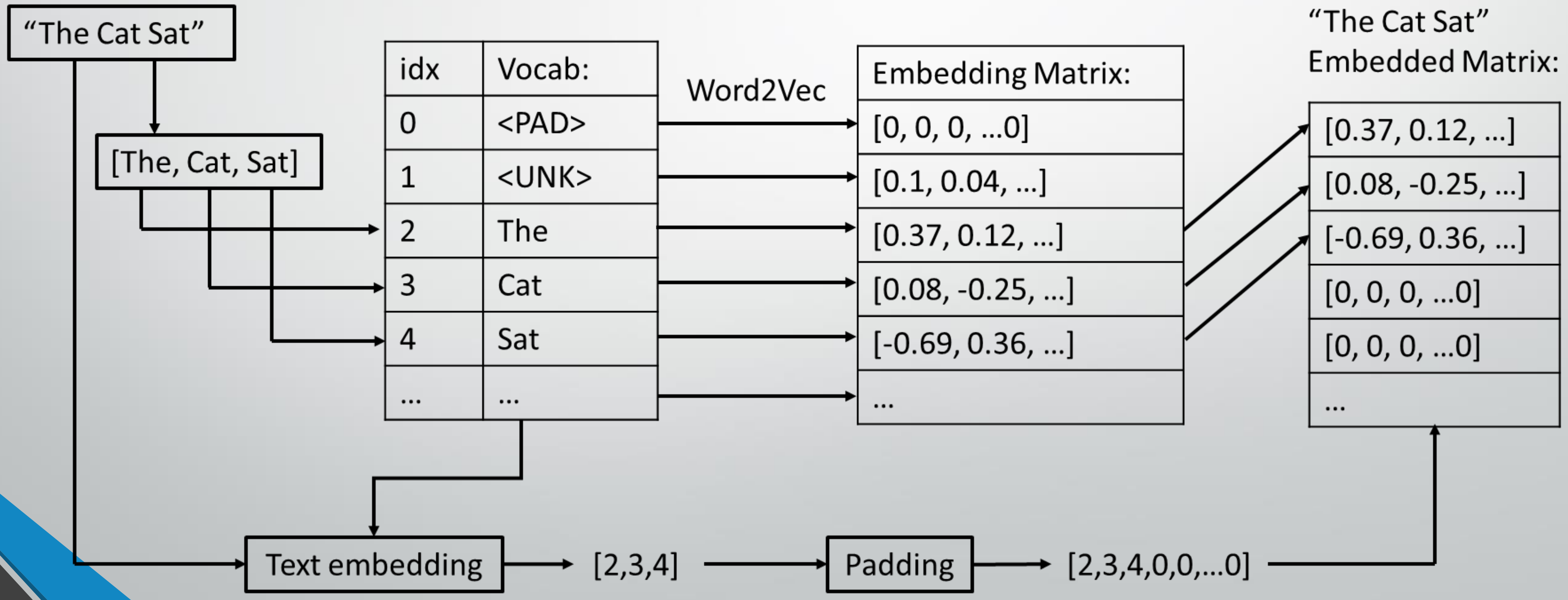
Advanced Implementation of Bi-Directional LSTM

Advanced Model - Key stages:

- Preprocessing and Embedding
- Model architecture
- Model Training

Advanced LSTM Model

Preprocessing and Embedding



Advanced LSTM model

Model Architecture

Input Text

Embedding Layer
- Converts word indices into vectors using pre-trained Word2Vec embeddings

Bi-Directional LSTM
- Captures both forward and backward contextual information in the text.

Attention Layer
- Computes weights for each word in the sequence to focus on the most relevant parts of the text.

First Dropout Layer
- Reduces overfitting by randomly deactivating 50% of neurons during training.

First Dense Layer
- 256 inputs from the LSTM/Attention layer, 128 outputs with ReLU activation.

Second Dropout Layer
- Reduces overfitting by randomly deactivating 50% of neurons during training.

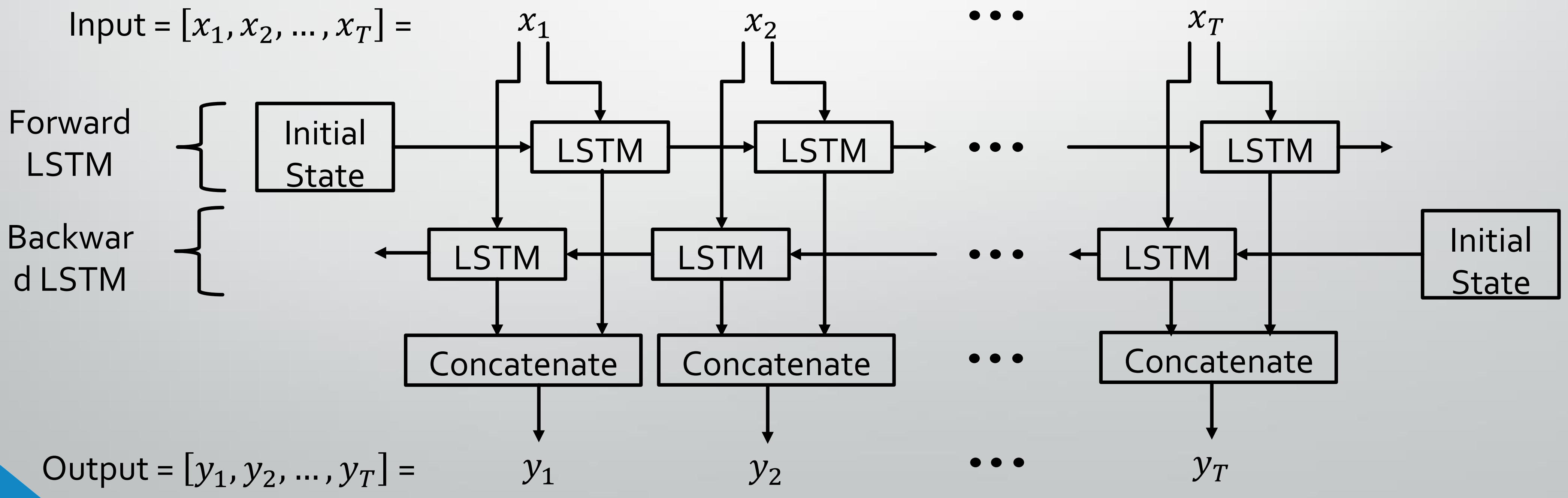
Second Dense Layer
- 128 inputs, 1 output with a Sigmoid activation for binary classification.

Output Layer

Advanced LSTM model

Model Architecture

Bi-Directional LSTM:



Advanced LSTM model

Model Architecture

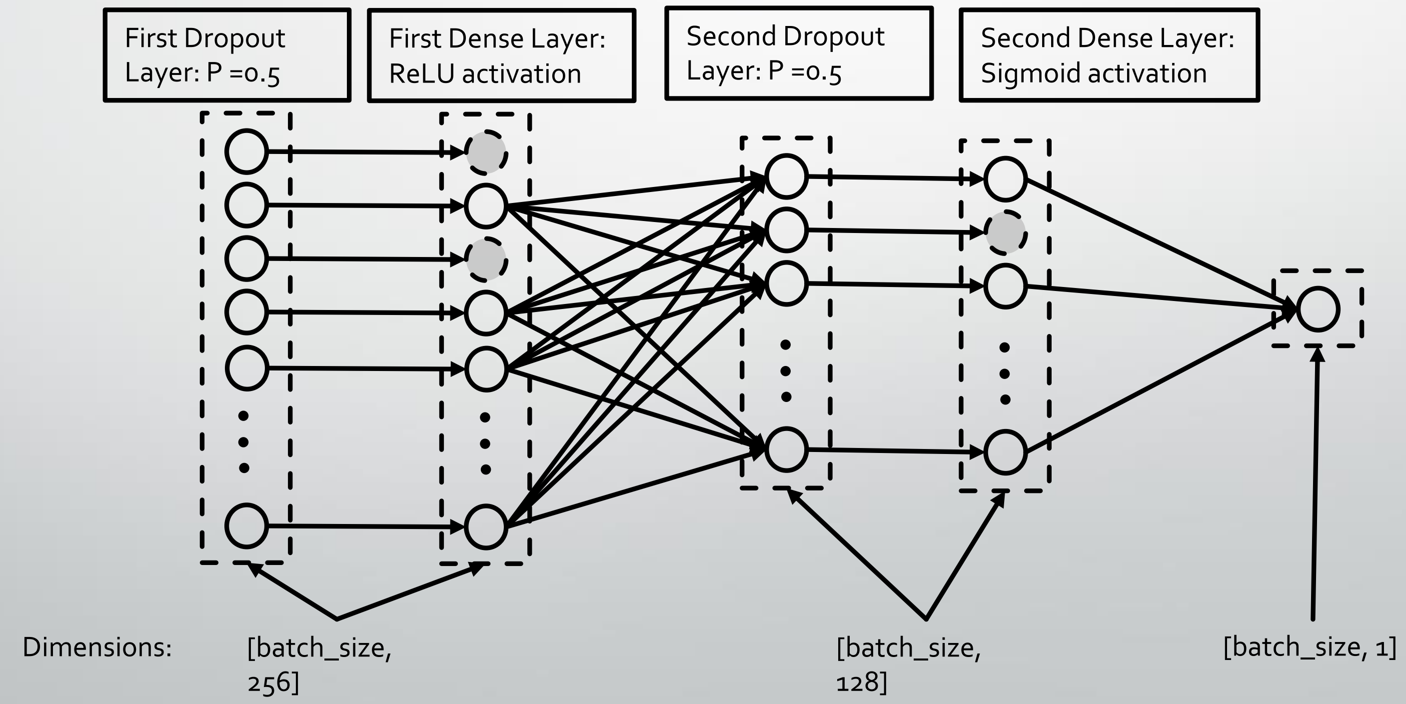
Attention Mechanism

- **How It Works:**
  - Assigns weights to words based on their importance.
  - Higher weights indicate greater relevance to the classification.
- **Impact:**
  - Enhances model accuracy and focus on critical parts of the input.

Advanced LSTM model

Model Architecture

Dropout and Dense Layers:



Advanced LSTM Model

Model Training

Loss Function:

- Binary Cross-Entropy.

Optimizer:

- Adam with learning rate 0.001.

Learning Rate Strategy:

- Warmup over initial steps.
- Dynamic adjustment with ReduceLROnPlateau scheduler.

Training Parameters:

- Batch size: 32
- Epochs: Max 15 (Early stopping after 4 epochs of no improvement).

Initial Challenges and Fixes

Challenge: Variable-Length Texts

Solution: Add uniform padding to tokenized text

Challenge: Inefficient learning rate adjustment.

Solution: Warmup scheduler with dynamic adjustment.

Challenge: Large gradients destabilized training.

Solution: Applied Gradient clipping.

Advanced LSTM Model - Key Insights

- **Initial Results:** Improved performance over FCNN
- **Key Insights:**
  - Attention mechanism improved interpretability and focus on critical text.
  - Bi-LSTM effectively captured sequential dependencies.
  - Highlighted the importance of fine-tuned embeddings and robust training strategies.

Performance Comparison:

Metric:	Baseline Model:	Logistic Regression Model:	Neural Network Model:	Advanced LSTM
Accuracy	51.5%	92.9%	94.36%	96%
Precision (True News)	51.5%	93%	94.95%	95%
Recall (True News)	100%	94%	94.73%	98%
F1-Score (True News)	67.95%	94%	94.84%	97%
Precision (Fake News)	N/A	93%	93.66%	98%
Recall (Fake News) 0%	0%	91%	93.92%	94%
F1-Score (Fake News)	N/A	92%	93.79%	96%

Summary of Results:

Models improved progressively from baseline to advanced architectures.

Advanced Bi-LSTM with attention mechanism achieved the best performance:

Accuracy: 95.78%

Precision: 96.21%

Recall: 94.57%

F1-Score: 95.38%.

Key Takeaways:

Preprocessing and bias removal are critical for reliable results.

Advanced techniques like Bi-LSTM and attention provide significant performance gains.

Future Work:

Explore additional datasets for better generalization.

Investigate transformer-based models for further improvement.

References:

- Goldberg, Y., & Levy, O. (2014). "Word2Vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method."
- Vaswani, A., et al. (2017). "Attention Is All You Need."
- Dataset Sources:
  - True and Fake News datasets: https://www.kaggle.com/code/therealsampat/fake-news-detection/notebook
  - WELFake Dataset: https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification/discussion/405485

Thank You For Listening