Deep Learning Model for Emotion Classification

1. Introduction

This report provides an overview of the process followed to develop a deep learning model for emotion classification using textual data. The objective was to explore and preprocess the dataset, construct an effective neural network, and analyze its performance using standard metrics. Challenges and improvements were addressed iteratively to achieve optimal results.

2. Data Exploration and Preprocessing

Dataset Description:

The dataset consists of two columns:

1. Text: Strings containing emotional expressions.
2. Label: Corresponding emotional categories (e.g., fear, sadness, joy).

Exploration:

- A label distribution plot revealed class imbalance, with 'joy' and 'sadness' dominating.
- Text length analysis indicated varying lengths, necessitating padding during preprocessing.

Preprocessing Steps:

1. Tokenized the text into sequences using Keras' Tokenizer, limiting vocabulary to the most frequent 10,000 words.
2. Padded sequences to a fixed length of 100 for uniformity.
3. Encoded labels into numerical values using LabelEncoder.
4. Split data into training (80%) and validation (20%) sets.

3. Model Architecture

The deep learning model was built using TensorFlow/Keras:

Embedding Layer: Converts text tokens into dense vectors of size 50.

Global Average Pooling Layer: Reduces dimensionality to capture average representation efficiently.

Dense Layers:

- Two layers with ReLU activation (64 and 32 neurons, respectively).
- Dropout (30%) applied for regularization.

Output Layer: Softmax activation to handle multi-class classification (6 classes).

Optimizer: Adam optimizer with a learning rate of 0.0005 for gradual convergence.

Loss Function: Sparse categorical crossentropy.

4. Model Training

Training Configuration:

Epochs: 20

Batch Size: 64
Early Stopping: Halts training when validation loss fails to improve for 3 consecutive epochs.
Learning Rate Reduction: Reduces learning rate on plateau to enhance performance.

Performance Metrics:
Training Accuracy: 91.97%
Validation Accuracy: 90.05%
Training Loss: 0.1495
Validation Loss: 0.1929

5. Model Evaluation

Classification Report:
- The model achieved high precision, recall, and F1-scores for dominant classes like 'joy' and 'sadness.'
- Smaller classes, such as 'love' and 'surprise,' showed lower performance due to class imbalance.

Confusion Matrix:
- Visualized misclassifications and identified patterns where smaller classes were often mispredicted as larger classes.

Training and Validation Curves:
- Both accuracy and loss curves showed convergence with minimal overfitting, indicating effective regularization.

6. Challenges and Improvements

1. Class Imbalance:
- Used class weights during training to give more importance to underrepresented classes.

2. Overfitting:
- Added dropout layers and implemented early stopping.

3. Optimization:
- Reduced learning rate dynamically during training to stabilize performance.

7. Findings and Conclusions

The model successfully achieved strong performance across all key metrics.
Challenges related to class imbalance and overfitting were addressed effectively using regularization and training strategies.
Further improvements could include:
1. Data augmentation for smaller classes.
2. Experimenting with more complex architectures like GRUs or Bidirectional LSTMs.