NLP Final Presentation

Abdelrahman Khaled

May 19, 2024

Outline

Introduction

Project Overview

- The aim of the project is to predict product prices based on various attributes.
- Understanding price dynamics aids in market analysis and decision-making.

Introduction to Dataset and Features

- Origin: The dataset originates from Amazon and comprises top-selling laptop accessories.
- Features: Title, description, brand, price, stars, reviews count, etc.
- Size: The dataset includes 500 rows.

Simple Model Development

Preprocessing Steps

- Cleaning and Tokenization
  - Remove special characters
  - Tokenize text into words
- Lemmatization
  - Normalize tokens to their base form
  - Reduces dimensionality
- Concatenation of Title and Description
  - Combine title and description into a single sequence
- Encoding and Padding
  - Convert words to numerical representations
  - Pad sequences to a fixed length of 150 tokens

Model Architecture

- Input Layer: Receives tokenized text sequences
- Embedding Layer: Converts tokens to dense vectors (100 dimenstions)
- LSTM Layer: Bidirectional LSTM units capture long-range dependencies (128 units)
- Dense Layer: Applies ReLU activation for non-linearity (64 units)
- Output Layer: Single neuron for price prediction (1 unit)

Training Process

- **Dataset Split:**
  - Training set: 80%
  - Validation set: 10%
  - Test set: 10%
- **Batch Size:**
  - Set to 32, balancing computational efficiency and training stability.
- **Epochs:**
  - Trained for 10 epochs, allowing iterative refinement of model weights.
- **Optimizer Settings:**
  - Adam optimizer with a learning rate of 0.001.
  - Chosen for its effectiveness in training deep neural networks.
- **Loss Function:**
  - Mean Squared Error (MSE) loss function.
  - Measures the average squared difference between predicted and actual prices.

Training and Testing Results: Simple Model

- **Training Results:**

Epoch	Training Loss	Validation Loss
1	3941.1536	2769.0745
2	3058.5154	1872.2606
3	2730.6440	1620.5831
4	2676.9148	1635.8971
5	2665.6924	1654.4270
6	2671.7400	1688.1238
7	2662.6802	1660.9471
8	2666.4351	1626.2347
9	2611.1729	1580.2756
10	2437.9158	1486.4126

- **Testing Results:**
  - Test Loss: 2644.9639

Residual Plots

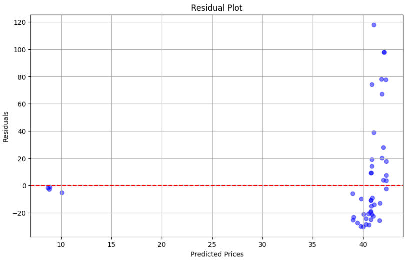

Figure: Residual Plot for Validation Dataset

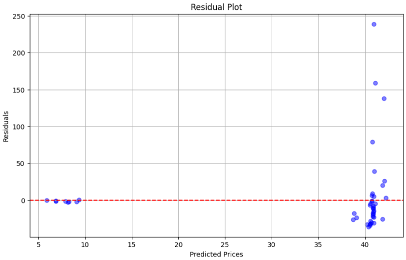

Figure: Residual Plot for Test Dataset

Pre-trained Model Fine-tuning

Fine-tuning with Pretrained Model

- **Choice of Pretrained Model:**
  - Utilized BERT (Bidirectional Encoder Representations from Transformers) as the pretrained model.
  - BERT is suitable for prediction tasks as it provides contextualized embeddings without the need for a decoder.
- **Preprocessing Steps:**
  - Concatenate product titles and descriptions.
  - Tokenize text using the BERT tokenizer.
  - Pad or truncate sequences for uniform length.
  - Encode tokens into numerical IDs and attention masks.
  - Encode brand information using label encoding.

Bert Integration

- Input data into BERT model consists of:
  - Token IDs: Numerical representations of tokens.
  - Attention Masks: Indicate valid tokens vs. padding.
- After BERT processing, pass BERT outputs combined with the encoded brand through two fully connected layers:
  - The First layer combines BERT output with encoded brand information.
  - The Second layer performs regression prediction with a single neuron.
- Mean Squared Error (MSE) is used as a loss function.

Training and Evaluation Results

- Training Loss:

Epoch	Training Loss	Validation Loss
1	4191.5744	4418.9487
2	4011.9904	4201.7373
3	3843.6064	4000.6934
4	3686.5248	3807.8892
5	3541.4464	3625.7041
6	3404.0112	3457.5764
7	3285.9200	3305.0376
8	3163.2736	3119.1428
9	3018.6064	2964.7173
10	2920.6848	2830.8445

Table: Training and Validation Losses for Pretrained Model

- Test Loss: 2758.7461

Residual Plots

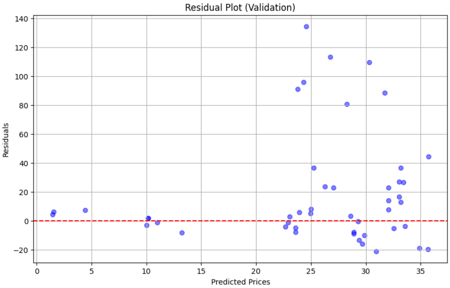

Figure: Residual Plot for Validation Dataset (Pretrained Model)

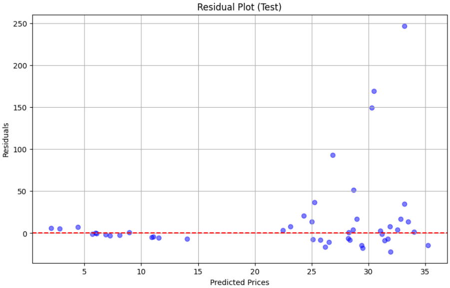

Figure: Residual Plot for Test Dataset (Pretrained Model)

Discussion

Discussion

- Comparison of Results:
  - The pretrained model achieved higher training and testing loss compared to the basic model.
  - While the basic model's plot showed predictions concentrated within narrow ranges, the pretrained model's plot exhibited a wider spectrum of predicted prices.
- Limitations and Future Direction:
  - The relatively small dataset size might have limited the pretrained model's ability to learn complex patterns effectively.
  - Limited computational resources.
  - Further optimization of hyperparameters such as learning rate and batch size could potentially improve the performance of both models.
  - Exploring more sophisticated architectures might lead to better results.

Conclusion

Conclusions

- Training differences emphasize the trade-offs between basic and pretrained models.
- Pretrained models offer richer semantic information but pose training challenges.
- The pretrained model's higher training losses underscore the need for thorough experimentation.
- Future iterations may explore alternative architectures, optimization, or preprocessing techniques.


Thank you