Introduction

This report details the computational learning experiments conducted for predicting the publication year of scientific papers based on their metadata. The process involves feature engineering, the implementation of a learning algorithm, hyperparameter tuning, and a discussion on the performance of the developed solution.

Feature Engineering

1. **Data Preprocessing:**

- Textual data in columns were preprocessed by converting to lowercase and removing punctuation.
- Feature ablation analysis demonstrated that the **'abstract'**, **'publisher'** and **'title'** are crucial features, in contrast, **'ENTRYTYPE'** and **'author'** have a lesser but still measurable impact.
- Converted certain columns to appropriate data types (e.g., 'ENTRYTYPE' to categorical and 'year' to numeric). Merged lists of authors into single strings.
- Various text fields concatenated to create new composite features such as **'author_publication'** and **'title_abstract'**, aiming to enhance the model's predictive capabilities.
- Null values were filled with empty strings, and specific column (**'editor'**) with 90% missing value was dropped.
- Split the dataset into training and validation sets, with stratification based on **'year'**, having a representative distribution of data from different years in both sets.

2. **Text Vectorization:**

   A pre-trained Word2Vec model (**word2vec-google-news-300**) was employed to create document vectors for the **'abstract'** feature while capturing semantic information from the text.

3. **TF-IDF Transformation:**

   Applied Term Frequency-Inverse Document Frequency (**TF-IDF**) vectorization separately to the **'title'**, **'author'**, **'publisher'** columns, while **OneHotEncoder** was applied to the categorical **ENTRYTYPE** field.

4. **Target Variable Transformation:**

   The target variable **'year'** was first transformed using a square and then a standard transformation, aiming to normalize its distribution and improve model performance.

Learning Algorithm(s)

Model Architecture

The model is a feedforward neural network with the following layers:

1. **Input Layer:** Accepts data shaped according to the number of input features.
2. **First Dense Layer:** 128 neurons with ReLU activation, allowing the model to learn complex, non-linear relationships.
3. **Dropout Layer:** 30% dropout rate, reducing overreliance on specific patterns and preventing overfitting.
4. **Batch Normalization Layer:** Stabilizes the learning process by normalizing the activations.
5. **Second Dense Layer:** 64 neurons with ReLU activation, including L2 regularization (0.001) to penalize large weights and mitigate overfitting.
6. **Another Dropout Layer:** An additional 30% dropout rate for further prevention of overfitting.
7. **Output Layer:** A single neuron for regression output.

Optimization and Loss Function

- **Adam Optimizer:** Used for efficient optimization due to its robustness and efficiency in handling noisy problems.
- **Mean Absolute Error (MAE):** Employed as the loss function to reduce outliers' impact in skewed datasets.

Hyperparameter Tuning

- **Regularization (L2 Regularization):** Incorporated in the second dense layer to prevent overfitting by discouraging large weights.
- **Learning Rate Scheduler:** Adjusts the learning rate after the first three epochs to refine the training process.
- **Early Stopping:** Implemented to terminate training when there's no improvement in validation loss for 3 epochs, preventing overtraining.

Performance Observations

- **Training Process:** The model was trained on a normalized target variable 'year', with a 25% validation split and a batch size of 32 for 15 epochs.
- **Training and Validation Loss**: A plot of training and validation loss over epochs was generated to visually assess the learning process and convergence. (Figure 1&2)
- **Model Evaluation:** The model's performance on the validation set was evaluated using MAE, with a reported value of 2.7365. This suggests reasonable accuracy with potential for improvement.
- **Post-Training Analysis:** The predictions on the validation and test sets were processed with inverse normalization and square root transformation, ensuring the accuracy and relevance of the model's output.

Discussion

The developed solution demonstrates a reasonable performance of predicting the publication year since the evaluation of the prediction submitted (MAE = 2.72) was the same as the validation set (MAE=2.73). The combination of feature engineering, model's architecture, with its emphasis on preventing overfitting and efficient learning, and strategic hyperparameter tuning contributed to its practical generalization. Further refinement and experimentation may enhance its predictive accuracy, making it a more robust tool for this application.

Figure 1- without L2 regularize and learning rate scheduler.

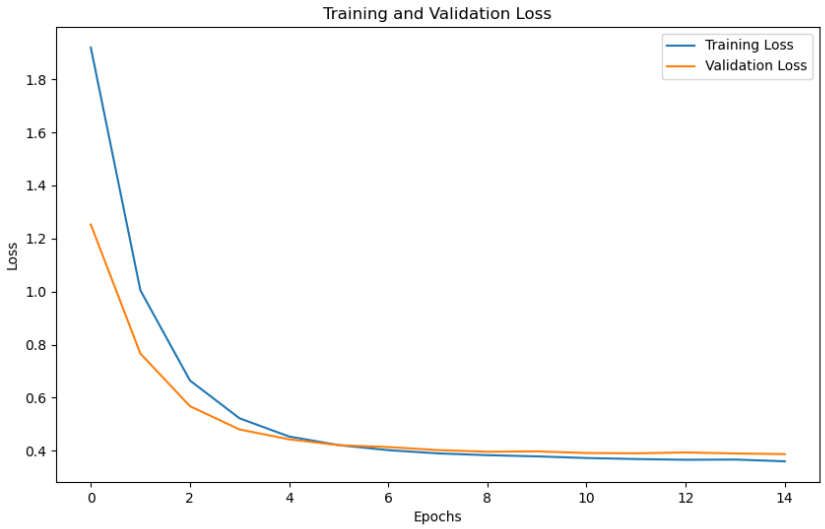

Figure 2- with L2 regularize and learning rate scheduler.