Gonçalo Aroeira Gonçalves, 99226
Matilde Heitor, 99284

Homework 2 - Group 33

Question 1

1.

1. Input $\underbrace{[\,]_D\ [\,]_D\ \cdots\ [\,]_D}_{L}$ INPUT $\in \mathbb{R}^{L\times D}$

THE LENGHT OF THE INPUT SEQUENCE IS L AND THE HIDDEN SIZE IS D

$P = Q K^T$, $Q \in \mathbb{R}^{L\times D}$, $K \in \mathbb{R}^{L\times D}$, $P \in \mathbb{R}^{L\times L}$

THE COMPLEXITY OF MULTIPLYING Q BY K^T IS $O(L^2 \times D)$

$q_1 \times K_1^T \rightarrow O(D)$

$q_i \times K_1^T$, $i = 1, \ldots, L \rightarrow O(L \times D)$

$q_i \times K_j^T$, $i, j = 1, \ldots, L \rightarrow O(L^2 \times D)$

WHERE q_i AND K_i DENOTE THE i^{TH} ROWS OF Q AND K, RESPECTIVELY.

SINCE THE COMPLEXITY OF APPLYING THE SOFTMAX AS THE ACTIVATION FUNCTION IS LINEAR

$SOFTMAX(P) \rightarrow O(L \times L) = O(L^2)$

$Z = SOFTMAX(P)V$, $V \in \mathbb{R}^{L\times D}$

AS BEFORE, THE COMPLEXITY OF THIS MULTIPLICATION IS

$O(L^2 \times D + L^2) = O(L^2 \times D)$

THE OVERALL COMPLEXITY BECOMES PROBLEMATIC FOR LONG SEQUENCES DUE TO IT'S QUADRATIC DEPENDENCE ON L.

2.

2.

FOR THE FEATURE MAP $\Phi: \mathbb{R}^D \rightarrow \mathbb{R}^M$
ASSUMING $q^T k << 1$ $q, k \in \mathbb{R}^D$, THEN

$$\exp(q^T k) \approx 1 + q^T k + \frac{(q^T k)^2}{2}$$

$$\Phi(q)^T \Phi(k) = 1 + q^T k + \frac{(q^T k)^2}{2}$$

$$= 1 + \sum_{i=1}^{D} q_i k_i + \frac{(\sum_{i=1}^{D} q_i k_i)^2}{2}$$

$$= 1 + \sum_{i=1}^{D} q_i k_i + \frac{(\sum_{i=1}^{D} q_i k_i)(\sum_{j=1}^{D} q_j k_j)}{2}$$

$$= 1 + \sum_{i=1}^{D} q_i k_i + \frac{\sum_{i=1}^{D}\sum_{j=1}^{D} q_i k_i q_j k_j}{2}$$

ASSUMING ONLY THE 1ST TERM OF THE MCLAURIN SERIES, $K = 1$

$\Phi(q) = [1]$, DIMENSIONALITY $M = 1$

ASSUMING $K = 2$

$$\Phi(q) = \begin{bmatrix} 1 \\ q \end{bmatrix}, \quad q \in \mathbb{R}^D \quad M = D + 1$$

ASSUMING $K = 3$

$$\Phi(q) = \begin{bmatrix} 1 \\ q \\ \frac{1}{2} q_1 q_1 \\ \frac{1}{2} q_1 q_2 \\ \vdots \\ \frac{1}{2} q_1 q_D \\ \frac{1}{2} q_2 q_1 \\ \frac{1}{2} q_2 q_2 \\ \vdots \\ \frac{1}{2} q_2 q_D \\ \vdots \\ \frac{1}{2} q_D q_1 \\ \vdots \\ \frac{1}{2} q_D q_D \end{bmatrix}, \quad q \in \mathbb{R}^D, \quad M = D^2 + D + 1$$

USING THE APPROXIMATION OF THE 3 1ST TERMS ON MCLAURIN SERIES WE OBTAIN THE DIMENSIONALITY $M = D^2 + D + 1$ OF THE FEATURE SPACE.
FOR $K \geq 3$ THE SAME LOGIC IS APPLIED AND THE DIMENSIONALITY OF THE FEATURE SPACE IS

$$M = D^K + D^{K-1} + \dots + 1$$

3.

$$Z = PV = \text{SOFTMAX}(\phi(Q)\phi(K)^T)V$$

$$\phi(Q)\phi(K)^T = [\phi(q_i)^T\phi(k_j)]_{i,j=1,\dots,L} \approx [\exp(q_i^T k_j)]_{i,j=1,\dots,L}$$

$$\phi(Q)\phi(K)^T 1_L = [\textstyle\sum_j \phi(q_i)^T\phi(k_j)]_{i=1,\dots,L} \approx [\textstyle\sum_j \exp(q_i^T k_j)]_{i=1,\dots,L}$$

$$D = \text{Diag}(\phi(Q)\phi(K)^T 1_L) = \text{Diag}([\textstyle\sum_j \phi(q_i)^T\phi(k_j)]_{i=1,\dots,L}) \approx \text{Diag}([\textstyle\sum_j \exp(q_i^T k_j)]_{i=1,\dots,L})$$

$$D^{-1} = \text{Diag}\left(\left[\frac{1}{\sum_j \phi(q_i)^T\phi(k_j)}\right]_{i=1,\dots,L}\right) \approx \text{Diag}\left(\left[\frac{1}{\sum_j \exp(q_i^T k_j)}\right]_{i=1,\dots,L}\right)$$

$$D^{-1}\phi(Q)\phi(K)^T = \left[\frac{1}{\sum_j \phi(q_i)^T\phi(k_j)}\phi(q_i)^T\phi(k_j)\right]_{i,j=1,\dots,L} \approx \left[\frac{1}{\sum_j \exp(q_i^T k_j)}\exp(q_i^T k_j)\right]_{i,j=1,\dots,L}$$

$$= D^{-1}\phi(Q)\phi(K)^T = \left[\frac{\phi(q_i)^T\phi(k_j)}{\sum_j \phi(q_i)^T\phi(k_j)}\right]_{i,j=1,\dots,L} \approx \left[\frac{\exp(q_i^T k_j)}{\sum_j \exp(q_i^T k_j)}\right]_{i,j=1,\dots,L} = [\text{SOFTMAX}(q_i^T k_j)]_{i,j=1,\dots,L}$$

$$D^{-1}\phi(Q)\phi(K)^T V = \left[\frac{\phi(q_i)^T\phi(k_j)}{\sum_j \phi(q_i)^T\phi(k_j)}\right]_{i,j=1,\dots,L} V \approx [\text{SOFTMAX}(q_i^T k_j)]_{i,j=1,\dots,L} V$$

$$= D^{-1}\phi(Q)\phi(K)^T V \approx \text{SOFTMAX}(\phi(Q)\phi(K)^T)V = PV$$

$$= D^{-1}\phi(Q)\phi(K)^T V \approx Z$$

4.

FOR THE APPROXIMATION

$$Z \approx D^{-1}\phi(Q)\phi(K)^T V$$

WHERE $D^{-1} \in \mathbb{R}^{L\times L}$, $\phi(Q), \phi(K) \in \mathbb{R}^{L\times M}$, $V \in \mathbb{R}^{L\times D}$

THE MATRIX MULTIPLICATIONS FOR $\phi(Q)\phi(K)^T$ CAN BE COMPUTED INDEPENDENTLY FOR EACH ROW AND SO THIS MULTIPLICATION BECOMES ASSOCIATIVE. THEREFORE, STARTING THE MULTIPLICATIONS FROM THE RIGHT TO THE LEFT:

$\phi(K)^T V \rightarrow O(MLD)$

$\phi(Q)(\phi(K)^T V) \rightarrow O(LMD)$

$D^{-1}(\phi(Q)(\phi(K)^T V))$ AS D^{-1} IS A DIAGONAL MATRIX $\rightarrow O(LD)$

THE OVERALL COMPLEXITY RESULT IS

$$O(MLD + LMD + LD) = O(LMD)$$

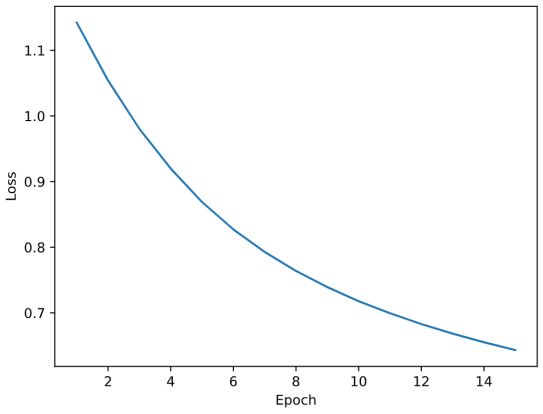
THIS APPROXIMATION ALLOWS FOR MORE EFFICIENT COMPUTATION BECOMING LINEAR IN TERMS OF L, COMPARING TO THE QUADRATIC FROM THE SOFTMAX APPLICATION.

Question 2

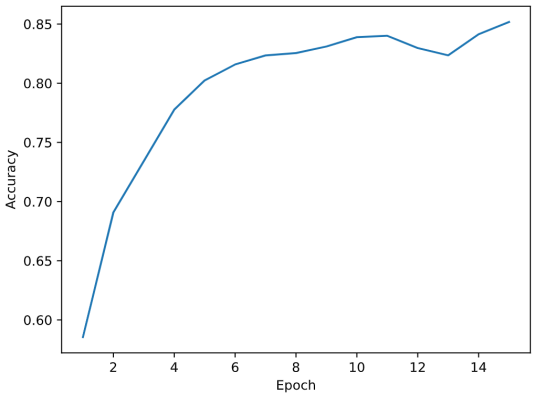
1.

learning_rate = 0.1 => Valid acc = 0.7705 => Final Test acc = 0.7675
learning_rate = 0.01 => Valid acc = 0.8517 => Final Test acc = 0.7921
learning_rate = 0.001 => Valid acc = 0.6910 => Final Test acc = 0.7051

Our best result in terms of validation accuracy was 0.8517 and final test accuracy was 0.7921, both for the model in which we used 0.01 for the learning rate.



Training loss - Figure 6

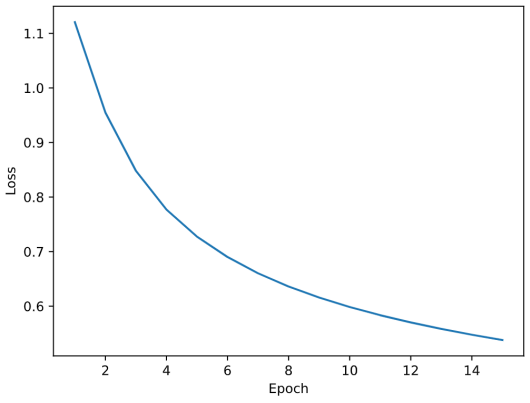


Validation accuracy - Figure 7

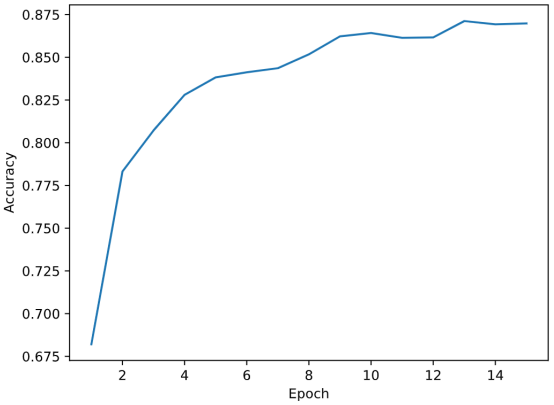
2.

learning_rate = 0.1 => Valid acc = 0.8251 => Final Test acc = 0.8015
learning_rate = 0.01 => Valid acc = 0.8698 => Final Test acc = 0.8242
learning_rate = 0.001 => Valid acc = 0.7660 => Final Test acc = 0.7486

Our best result in terms of validation accuracy was 0.8698 and final test accuracy was 0.8242, both for the model in which we used 0.01 for the learning rate.



Training Loss - Figure 8



Validation Accuracy - Figure 9

3.

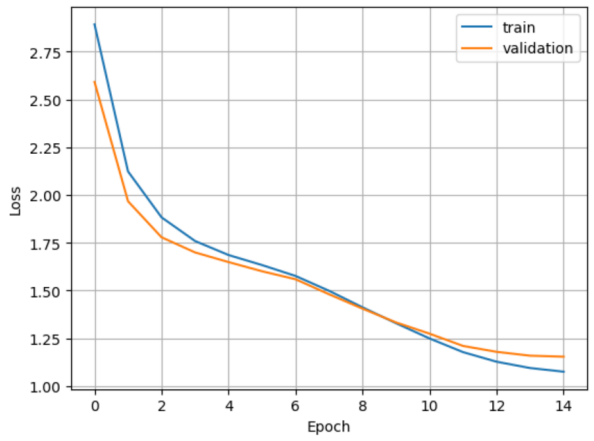
The number of trainable parameters is 224892 for both types of CNNs. While the number of trainable parameters is the same, the architectural differences in terms of max-pooling and stride have a significant impact on how these models learn and represent features. The CNN where there is max-pooling might be better at capturing detailed information and learning hierarchical representations due to the inclusion of it with smaller strides. The second CNN, with larger strides, might sacrifice some spatial details and context, potentially affecting its performance.

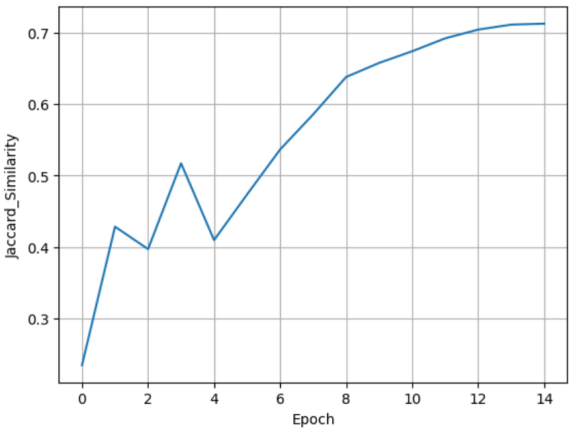
The best model for this depends on its objective. If it´s for recognizing objects based on their local features and spatial relationships, the original CNN may be more appropriate. If fine-grained details are essential, the modified CNN without max-pooling may be a better choice.

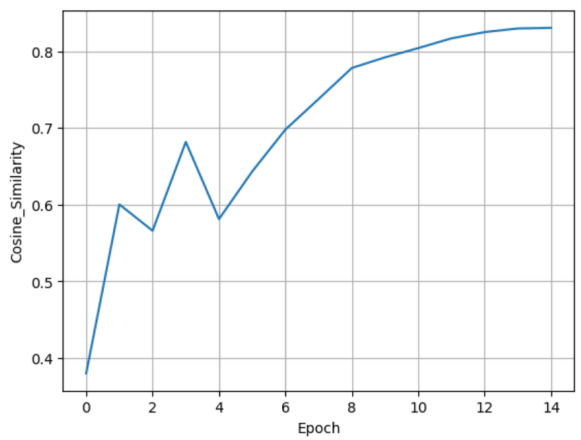
Question 3

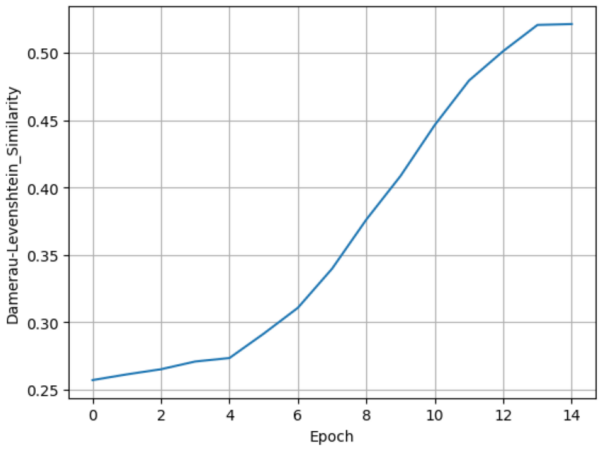
1.

Final test loss: 1.1616197032172506
Jaccard similarity score: 0.7184034646635822
Cosine similarity score: 0.8346791754594555
Damerau-Levenshtein similarity score: 0.5204482737993288



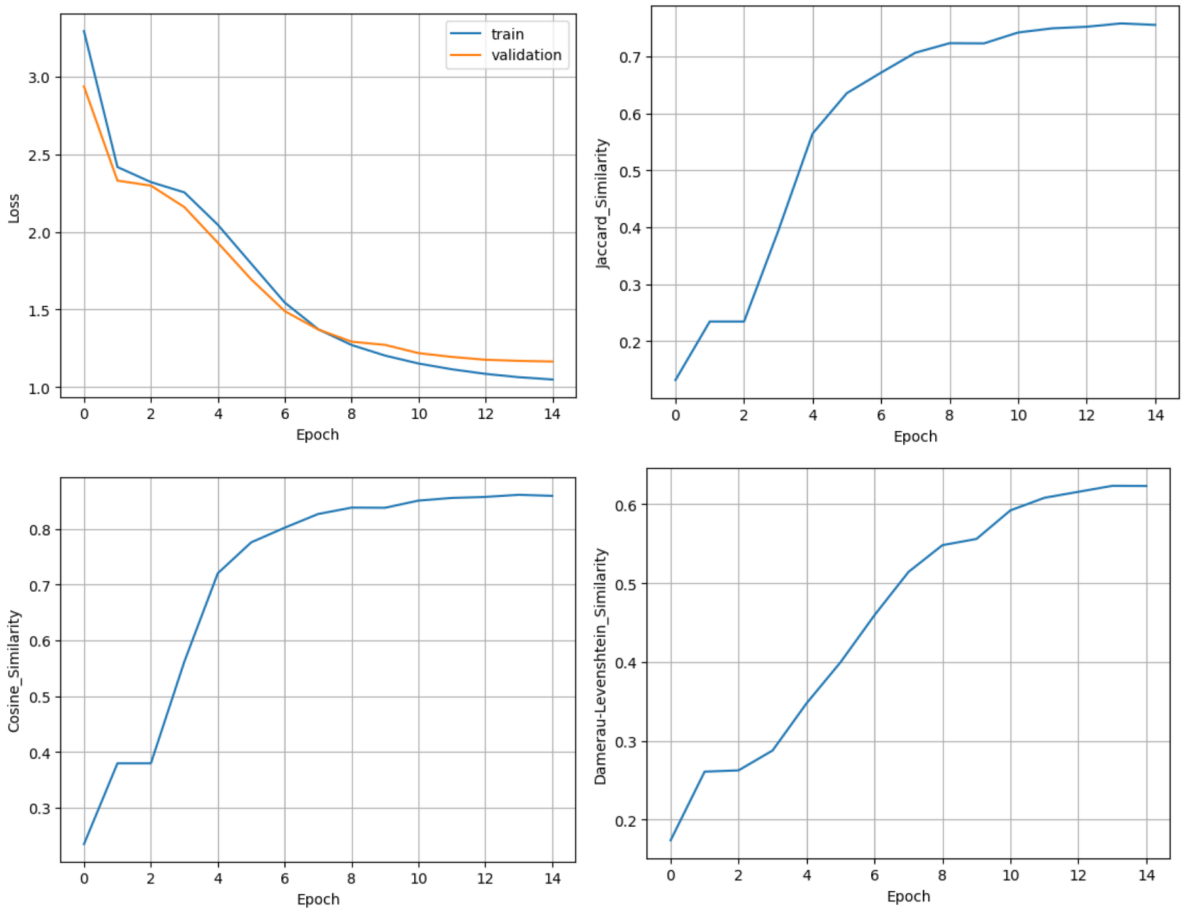






2.

Final test loss: 1.1604668236360318
Jaccard similarity score: 0.7655570957245197
Cosine similarity score: 0.8660184556068071
Damerau-Levenshtein similarity score: 0.6341917111051074



3.

For the LSTM, the input is processed sequentially, the hidden state updated based on the current input and the previous hidden state, capturing sequential dependencies.

Attention mechanisms, on the other hand, process the input text in parallel, assigning different weights to different parts of the input sequence, saving positional information and capturing dependencies beyond sequential order.

The differences in test results stem from the inherent characteristics of the two architectures. While the LSTM excels in sequential modeling, it may lead to word-level inaccuracies and loss of coherence, incorrectly predicting or skipping words and resulting in deviations from the target. On the other hand, while not aligning with real text, the attention mechanism's ability to focus on relevant parts of the input sequence allows it to capture more nuanced relationships between words, resulting in more contextually accurate and coherent text generation and proximity to the actual content.

The choice between these architectures might depend on the nature of the data and the desired modeling capabilities. Evaluating the performance metrics the self attention performs slightly better but the distinction is the analysis of the resulting outputs and despite the fact that neither of the mechanisms align precisely with the target text, the self-attention demonstrates a superior ability to retain the overall meaning and structure of the input.

4.

LSTM (question 1)
Final test loss: 1.1616
Jaccard similarity score: 0.7184
Cosine similarity score: 0.8347
Damerau-Levenshtein similarity score: 0.5204

Self attention (question 2)
Final test loss: 1.1605
Jaccard similarity score: 0.7656
Cosine similarity score: 0.8660
Damerau-Levenshtein similarity score: 0.63

The Jaccard Similarity assesses shared vocabulary, the Cosine Similarity prioritizes vector alignment, and the Damerau-Levenshtein Similarity penalizes sequence transformation complexity.

The worst presented metric for both mechanisms is Damerau-Levenshtein. As it focuses on token precision, it highly punishes the LSTM, yielding the lowest score when whole tokens are predicted incorrectly. Self-Attention, with positional information, maintains proximity to the target even with incorrect predictions, resulting in a comparatively higher Damerau-Levenshtein score. Jaccard is the second worst metric, highlighting shared vocabulary, tends to increase as models learn similar vocabularies but can be influenced by the length of the predicted and target sequences, potentially inflating the score. The best results are with Cosine, emphasizing vector alignment, remains high despite occasional token errors, offering a forgiving measure.

The differences in their behavior highlight the strengths and weaknesses of each metric in capturing various aspects of text generation, contributing to a comprehensive evaluation of model performance.

Contributions

In this project, both members collaborated effectively to address the diverse set of tasks across all three questions. Gonçalo took the lead in implementing and training the Image classification with CNNs (2) and Matilde completed both text decoders for Automatic Speech Recognition but both members equally contributed to the critical analysis of difference in terms of performance between the networks (2.3), and test results and differences for LSTM and Self-Attention and related similarity scores (3.3 and 3.4).
In Question 1, both members jointly tackled the self-attention layer of a transformer and various approximations as well as the study of its computational complexities. Throughout the project, both members worked collaboratively and simultaneously, leveraging their individual strengths and expertise to contribute effectively to each task.