Deep Learning (IST, 2024-25)

Homework 2

Diogo Ribeiro
ist1102484

Vasco Paisana
ist1102533

Question 1

1.

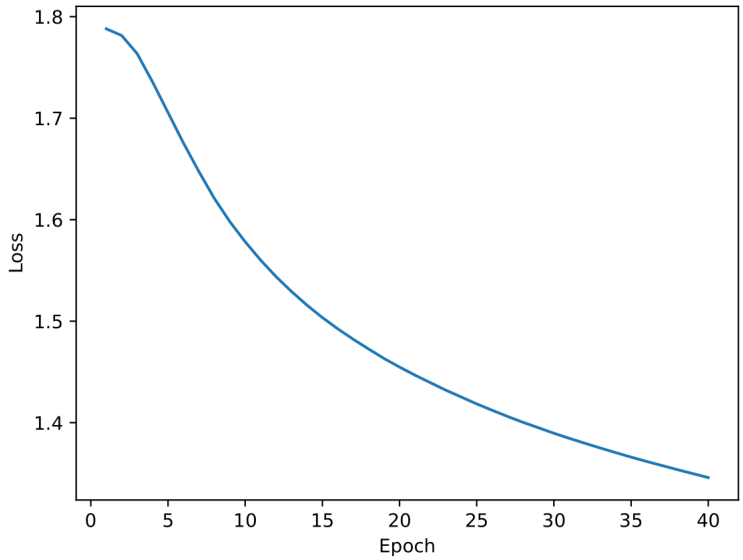
2.

3.

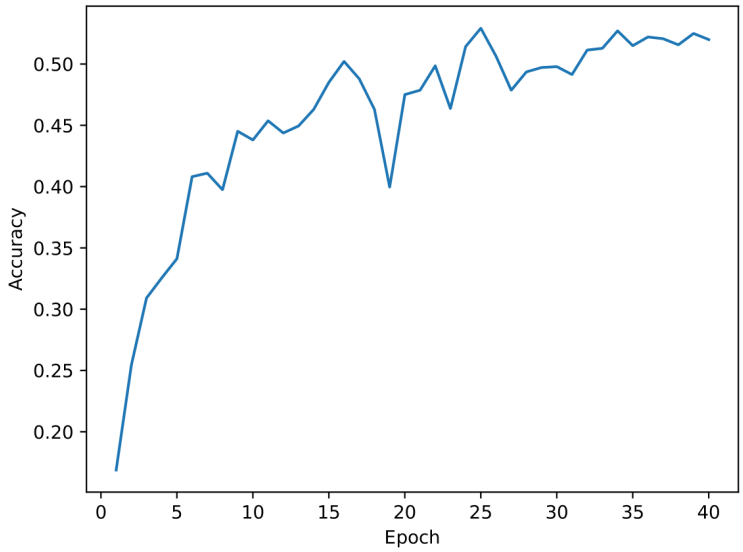
Question 2

1. After training the model for 40 epochs using 3 different values for learning rate (0.1; 0.01; 0.001), we obtained that the learning rate of the best configuration is 0.01, with a final test accuracy of 0.5243.

Training loss as function of epoch number for the best configuration

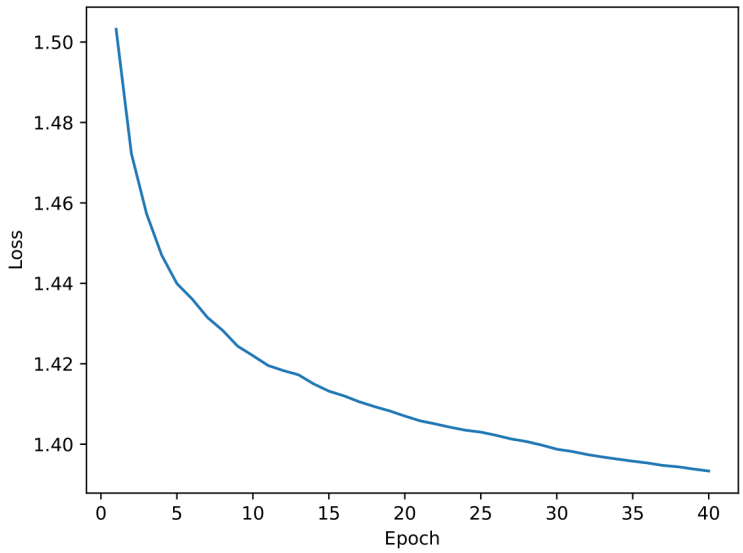


Validation accuracy as function of epoch number for the best configuration

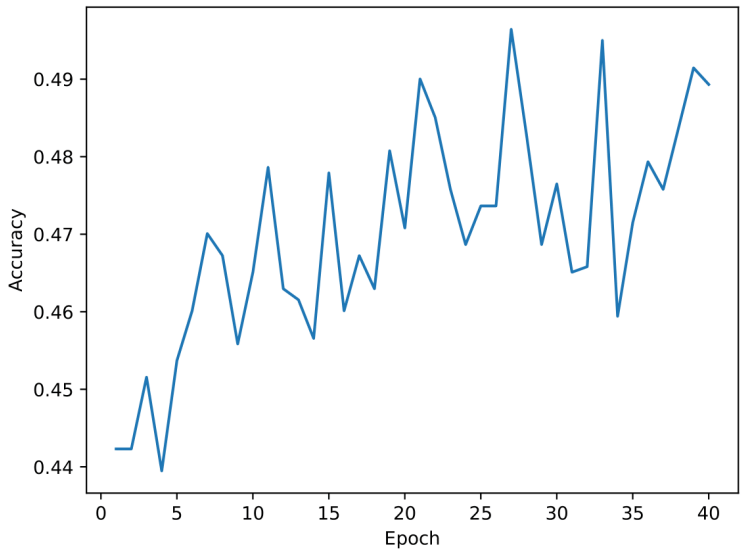


2. Using the optimal configuration from the previous exercise i.e. learning rate 0.01, and training the model for 40 epochs we obtained a final test accuracy of 0.4743.

Training loss as function of epoch number for the best configuration



Validation accuracy as function of epoch number for the best configuration



3.

We obtained 5247494 trainable parameters for exercise 1. While for exercise 2 we got 660230 trainable parameters, which is considerably lower.

By replacing transformation of flattening the output of the convolutional blocks with a global average pooling layer reduces the number of parameters, given that this comes as another pooling layer. Although this reduces the computational costs, the overall accuracy decreases, given the loss of some spatial and feature details important for capturing fine-grained patterns. Despite this, applying the pooling keeps good accuracy values even with much fewer parameters.

However, in the first epochs the model from exercise 2 achieves even higher accuracy rates faster. By using batch normalization, it stabilizes the gradients, thus accelerating convergence. This makes the second model a good choice when a faster training (less epochs) is needed.

4.

Using small kernels reduces the number of parameters compared to kernel of (width x height), which brings numerous advantages. Given less parameters, we are able to generalize and prevent overfitting. Moreover, training and inference require fewer computation steps, reducing the time and the power consumption to train the model.

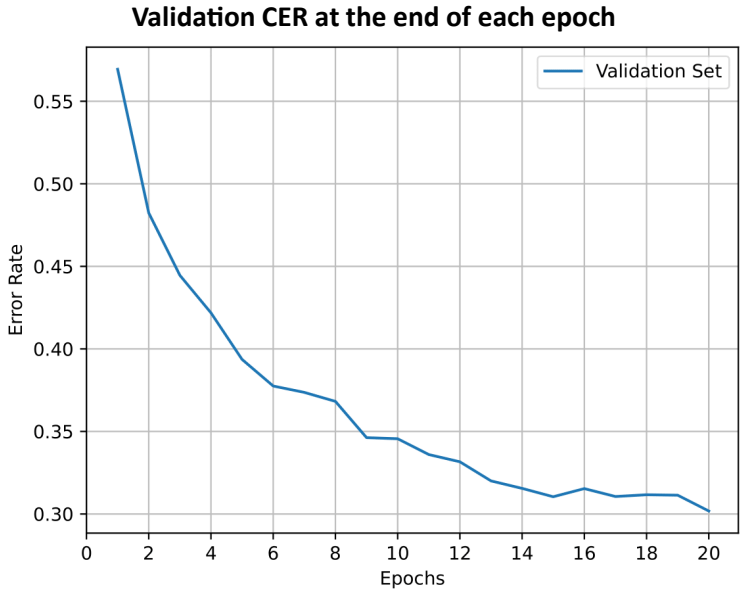
Smaller kernels are more effective at capturing local features because excessively large kernels can lead to over smoothing and the loss of fine-grained details in the input data. By stacking multiple small kernels, we can achieve an equivalent receptive field to a larger kernel while maintaining a more granular focus on local patterns, since each small kernel is followed by an activation function that introduces non-linear transformations between layers. This allows the network to learn more complex and abstract features than otherwise.

Pooling layers simplify feature maps by reducing their size, which lowers computational costs by decreasing the number of calculations and memory required for subsequent layers. They also highlight important features by emphasizing the most significant activations in a region, and help prevent overfitting by reducing the number of learnable parameters, making the network more focused on general patterns rather than noise or fine-grained details.

Question 3

1.

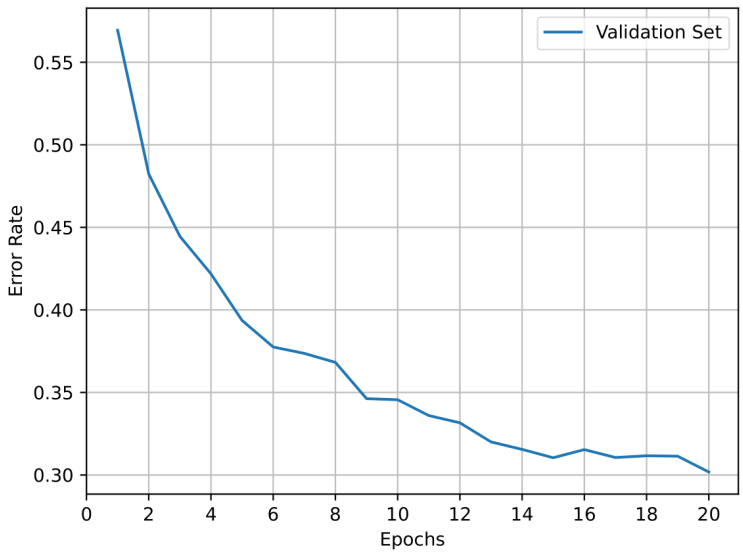
a) After training we obtained the following plot:



Then, running the checkpoint (minimum CER) on the test set:

- **CER –** 0.3006
- **WER –** 0.7980

b) After training, now with attention mechanism, we obtained the following plot:



Then, running the checkpoint (minimum CER) on the test set:

- **CER –** 0.2041
- **WER –** 0.7190

c) Using the same checkpoint as b) on the test set but now with nucleus sampling instead of greedy decoding:

- **CER –** 0.2269
- **WER –** 0.7630
- **WER@3 –** 0.6500

Contribution of each member & bibliography

➔ TODO

https://www.geeksforgeeks.org/how-to-choose-kernel-size-in-cnn/

https://data-ai.theodo.com/blog-technique/2019-10-31-convolutional-layer-convolution-kernel