DEEP LEARNING

INSTITUTO SUPERIOR TÉCNICO

1ST SEMESTER - 2021/22

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

Homework 2

Group 44:

Diogo Matos	86981
Diogo Moura	86976
Felicia Ekenstam	101636

January 12th, 2022

Question 1

1

How many parameters are there in total in the network with an input image of size 28x28x3?
Convolutional layer: This layer only takes the filter into consideration regarding the number of parameters, meaning that it has 5*5*3 = 75 parameters times 8, since there are 8 kernels and also no padding. Then there are is bias for each kernel so in total there will be 75*8+8 = 608 parameters in the convolutional layer.
After the convolutional layer the image size is:

$$\frac{input_{size} - kernel_{size}}{stride} + 1 = \frac{28 - 5}{1} + 1 = 24$$

The output size is then 24x24x8, since there are 8 kernels.

Max-pooling layer: In a max-pooling layer there are no trainable parameters since it only passes the maximum value within each rectangle to the next layer.
After the max-pooling layer the image size is:

$$\frac{input_{size} - pooling_{size}}{stride} + 1 = \frac{24 - 4}{2} + 1 = 11$$

The output size is then 11x11x8. With the image now being 11x11x8, it has to be flattened before entering the linear layer. This means that, in order to obtain the output vector size, the dimensions of the image are multiplied by the number of filters , 11*11*8 = 968.

Linear output layer: With the input vector of size 968x1 and the 10 output neurons, there are 968*10 + 10 = 9 690 parameters, since there is a bias for each output neuron. With the following softmax transformation, there are no trainable parameters since it is just a mathematical operation.

Therefore, the total number of parameters in the network is **608 + 0 + 9690 = 10 298.**

2

How many parameters are there in total if we replace the convolutional and max-pooling layers by a feedforward, fully connected layer with hidden size 100 and output size 10?
The input is an image of size 28x28x3, meaning we have to flatten it before passing it through the feedforward, fully-connected layer: 28*28*3 = 2 353. Giving an input size of 2352x1.

Feedforward, fully-connected layer: With the input size (i) of 2352x1, the hidden size (h) of 100 and 10 output neurons (o) we get:

$$\begin{aligned}(i * h + h * o) + h + o &= (2352 * 100 + 100 * 10) + 100 + 10 \\ &= 236310 \ parameters.\end{aligned}$$

Therefore, the total number of parameters in the network is **236 310**, which is way more parameters compared to the model in 1.1. This is because in the fully connected layers every neuron from one layer is connected to every neuron of the other layer.

3

Let X $\in$ R^{Lxn} be an input matrix for a sequence of length L, where n is the embedding size. Assume two self-attention heads have projection matrices W_Q^h, W_K^h, W_V^h $\in$ R^{nxd} for h $\in$ {1, 2}, with d $\leq$ n.

3.1

Show that the self-attention probabilities can be written as the rows of a matrix of the form below, and provide an expression for $A^{(h)}$:

$$P^{(h)} = Softmax(XA^{(h)}X^T) \tag{1}$$

Since the probabilities can be written as the following, using the transpose rule of $(AB)^T = B^T * A^T$:

$$\begin{aligned} P^{(h)} &= Softmax(S) \\ &= Softmax(QK^T) \\ &= Softmax((XW_Q^h)(XW_K^h)^T) \\ &= Softmax(X(W_Q^h(W_K^h)^T)X^T) \end{aligned}$$

Then we have the expression for the matrix $A^{(h)} = (W_Q^h(W_K^h)^T)$, with the rank($A^{(h)}$) $\leq$ d. Since W$_Q^h$ and W$_K^h$ $\in$ R^{nxd}, meaning that (W$_K^h$)^T $\in$ R^{dxn}. The multiplication of these two matrices gives the matrix $A^{(h)}$ with dimensions (n x n).

3.2

Show that, if $W_Q^{(2)} = W_Q^{(1)}B$ and $W_K^{(2)} = W_K^{(1)}B^{-T}$, the self-attention probabilities are exactly the same for the two attention heads.

From the previous question we have the following expression for the self-attention probabilities:

$$P^{(h)} = Softmax(X(W_Q^h(W_K^h)^T)X^T)$$

With the cases of $W_Q^{(2)} = W_Q^{(1)}B$ and $W_K^{(2)} = W_K^{(1)}B^{-T}$, and the inverse rule that $AA^{-1} = I$, this would give us:

$$P^{(1)} = Softmax(X(W_Q^{(1)}(W_K^{(1)})^T)X^T)$$

$$\begin{aligned} P^{(2)} &= Softmax(X(W_Q^{(2)}(W_K^{(2)})^T)X^T) \\ &= Softmax(X((W_Q^{(1)}B)(W_K^{(1)}B^{-T})^T)X^T) \\ &= Softmax(X((W_Q^{(1)}B)((B^{-T})^T(W_K^{(1)})^T)X^T) \\ &= Softmax(X(W_Q^{(1)}BB^{-1}(W_K^{(1)})^T)X^T) \\ &= Softmax(X(W_Q^{(1)}I(W_K^{(1)})^T)X^T) \\ &= Softmax(X(W_Q^{(1)}(W_K^{(1)})^T)X^T) \end{aligned}$$

Question 2

1

CNNs are good at capturing translation or shift invariances.

2

According to table 1, the learning rate which works best is 0.01.

Table 1: Training Loss and Validation accuracy of the CNN according to the learning rate

Learning Rate	Training Loss	Valid. Acc.	Final Test Acc.
0.01	0.3268	0.9111	0.9060
0.001	0.6465	0.8572	0.8517
0.1	0.2542	0.9016	0.8986

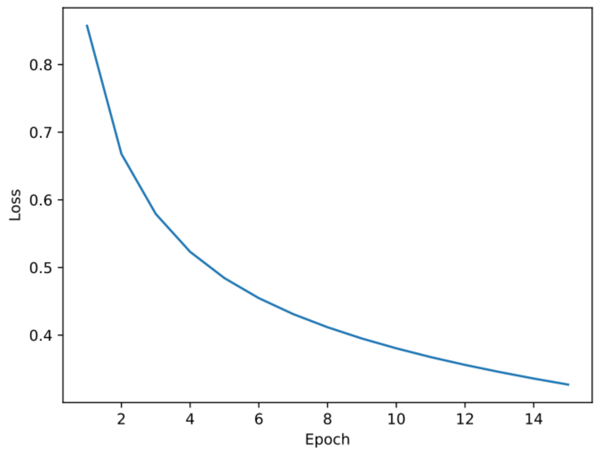

Figure 1: Plot of the loss as a function of the epoch number, with a learning rate of 0.01

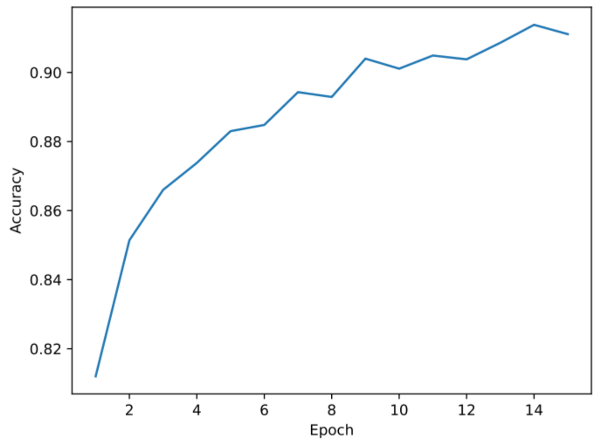

Figure 2: Plot of the accuracy as a function of the epoch number, with a learning rate of 0.01

3

In general, in bottom layers, are captured low-level features which contain both edge/shape filters and color constrictive information, which are highly consistent with the shape and opponent information in the human visual system.

In general, in top-layers, are captured high-level semantic features, such as objects or parts of objects in the image that have semantic meaning to humans.

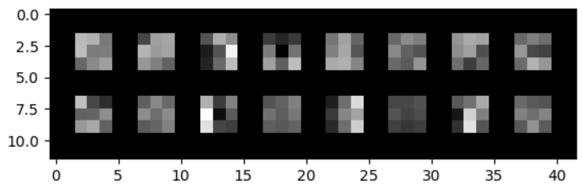

Figure 3: Plot of the first convolutional layer of the CNN

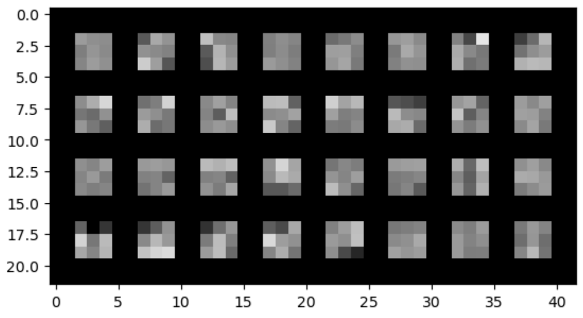

Figure 4: Plot of the second convolutional layer of the CNN, with a learning rate of 0.01

Question 3

1

1 a)

Without the attention mechanism, the final training Loss is 0.8908, the final validation Bleu-4 score is 0.4921 and the final test Bleu-4 score is 0.5150.

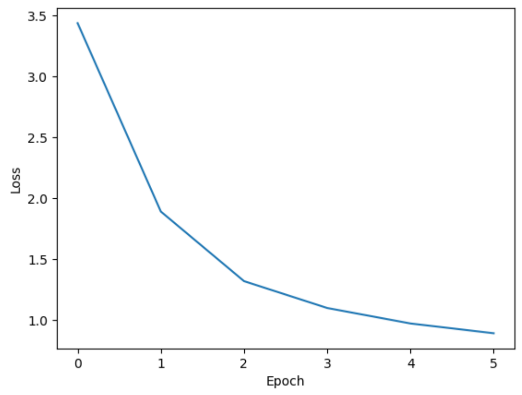

Figure 5: Plot of the loss as a function of the epoch number, without the attention mechanism

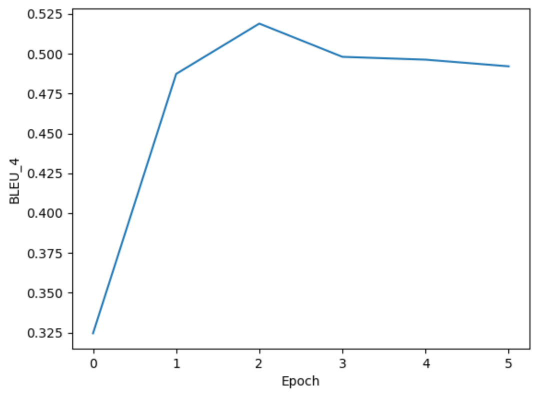

Figure 6: Plot of the validation bleu-4 score as a function of the epoch number, without the attention mechanism

1 b)

With the attention mechanism, the final training Loss is 0.8420, the final validation Bleu-4 score is 0.5283 and the final test Bleu-4 score is 0.5349.

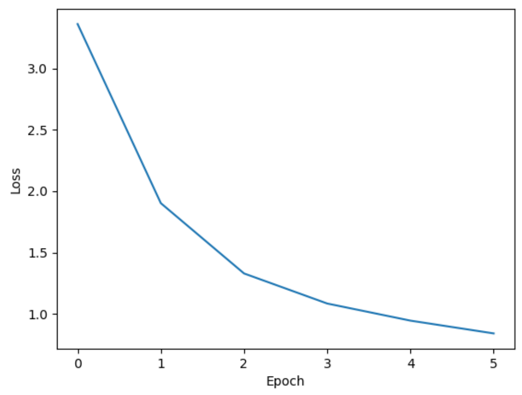

Figure 7: Plot of the loss as a function of the epoch number, with the attention mechanism

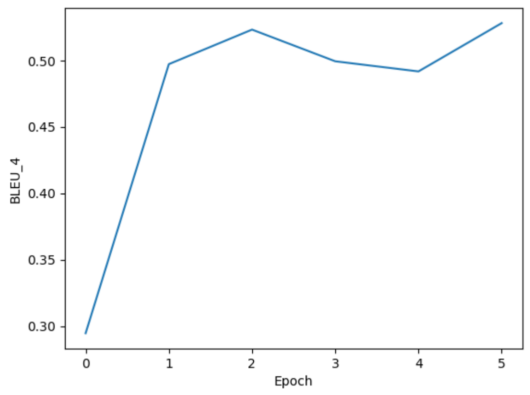

Figure 8: Plot of the validation bleu-4 score as a function of the epoch number, with the attention mechanism

1 c)

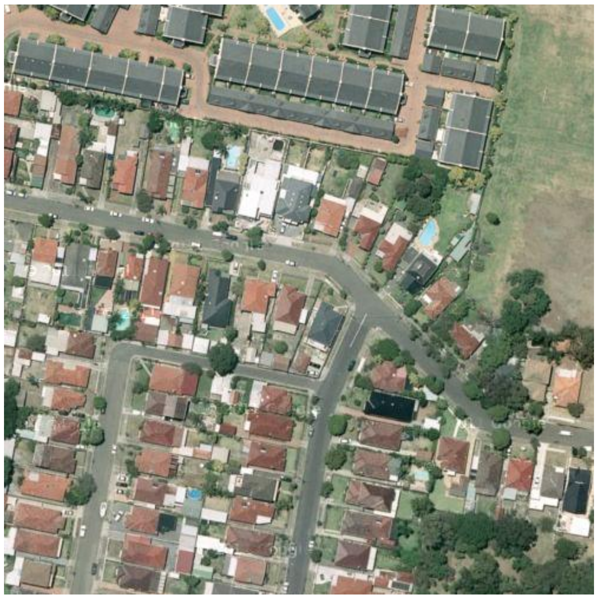

Figure 9: Figure 219.tif of the dataset. The generated caption was: "a residential area with houses arranged neatly and some roads go through this area".

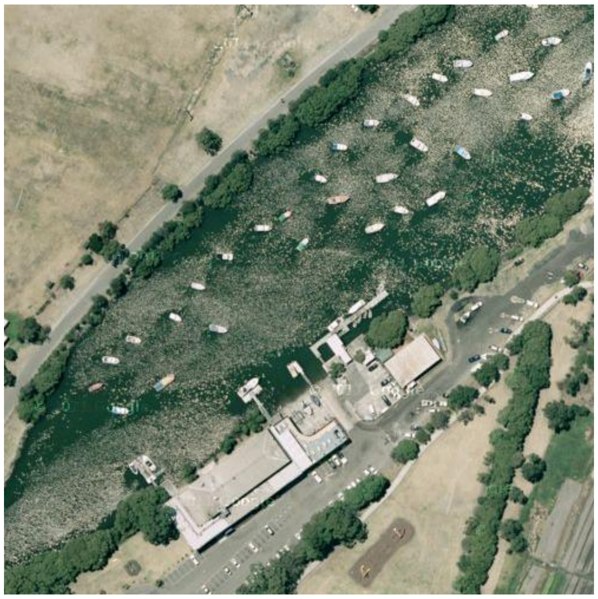

Figure 10: Figure 357.tif of the dataset. The generated caption was: "a curved river with some green waters and a highway passed by".

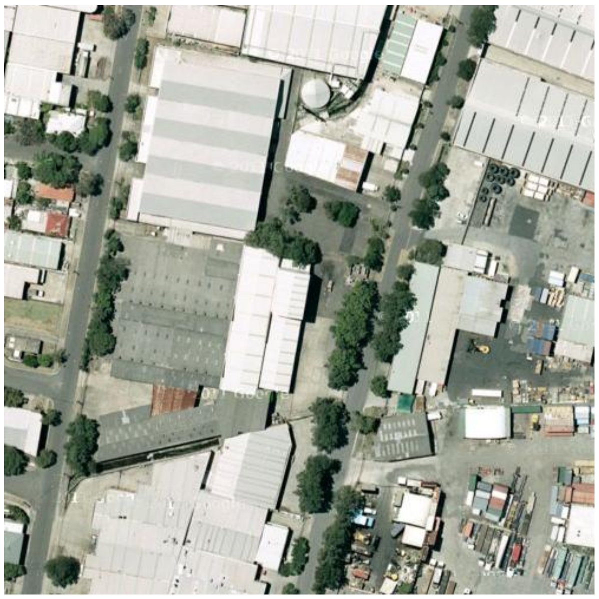

Figure 11: Figure 540.tif of the dataset. The generated caption was: "an industrial area with many white buildings and some roads go through this area".