IN4310 mandatory 2

Erlend Kristensen

25.04.2022

1 Models and parameters

We have made 4 different models, first using simple RNN layers, second using double GRU-cells, third with double LSTM-cells, and the fourth and last one which is a more complex two-layered neural network using LSTM-cells with maxpooling and an attention layer.
The models were made with a batch size of 128, number of epochs was 99 for task 1-3, and 50 for task 4. Learning rate was 0.001, weight decay 0.00001, embedding size 300, vocabulary size 10000, truncated backprop length 25, hidden state sizes 512, num rnn layers (1 for task 1, 2 for rest), scheduler milestones [75,90] and scheduler factor 0.2.

2 Loss curves and METEOR scores

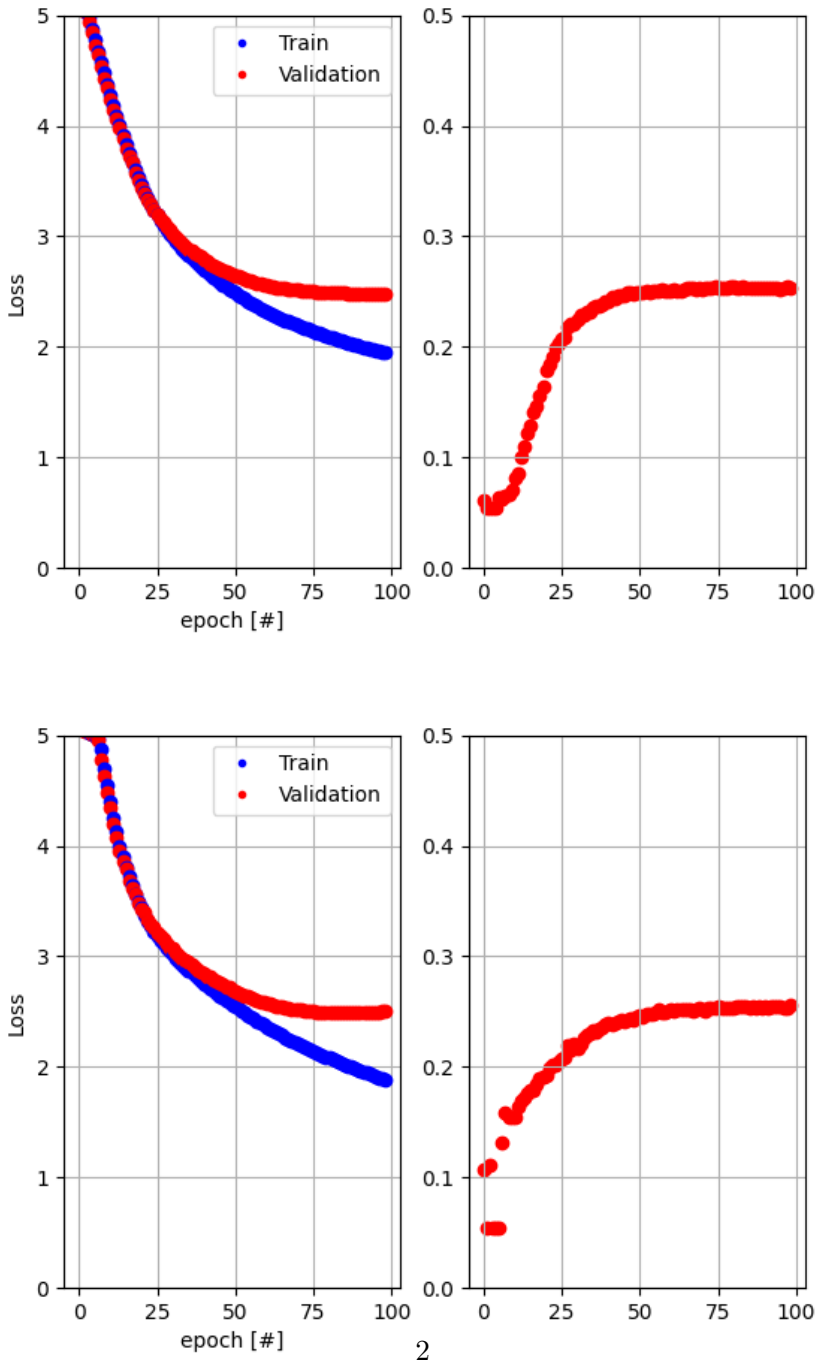


Figure 1: Loss curves and METEOR validation scores for task 1 and 2. Task 1 is the first image.

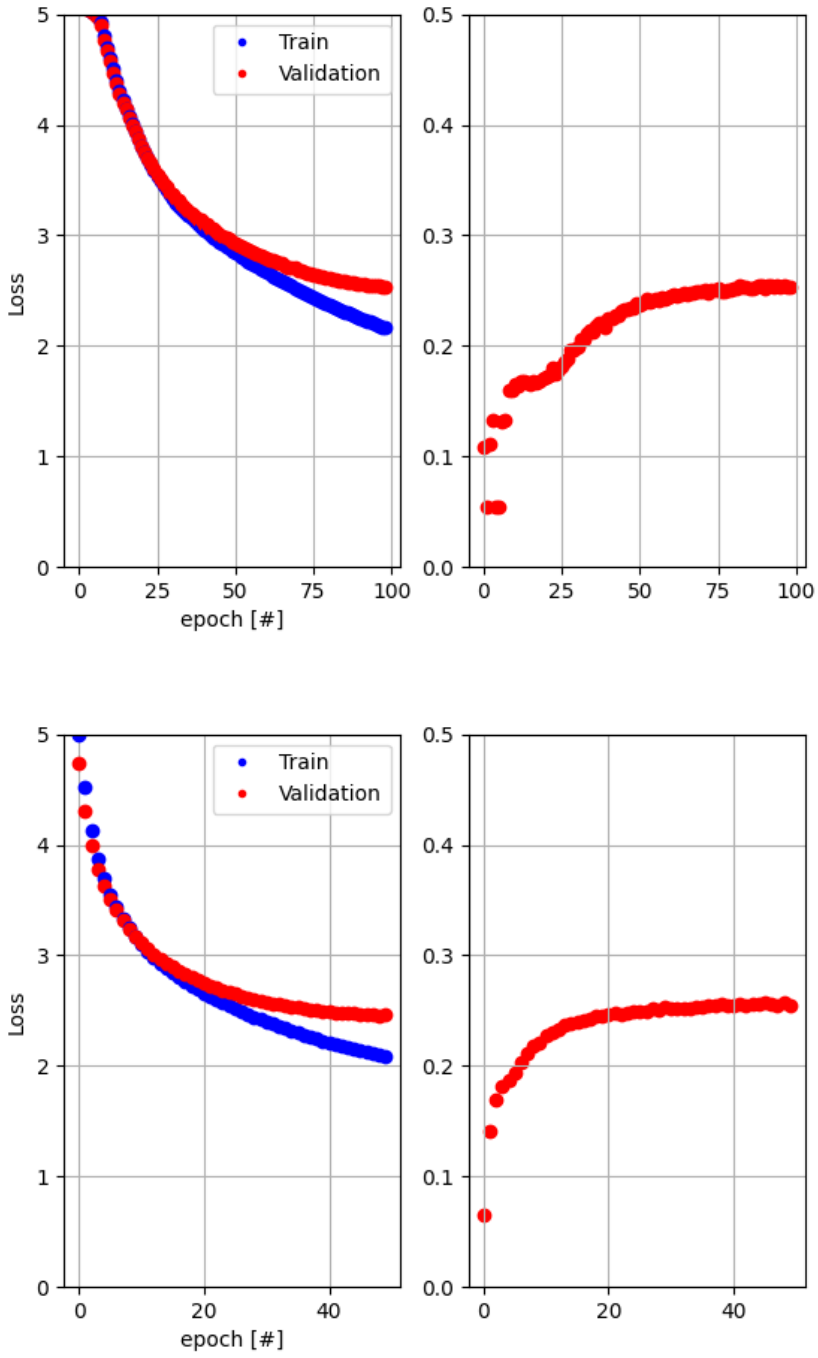


Figure 2: Loss curves and METEOR validation scores for task 3 and 4. Task 3 is the first image.

The first model from task 1, with simple rnn layer, is the one that uses the most epochs to get to a good METEOR value, while we observe that the one from task 4, with a complex doubled LSTM layer, uses the least amount of epochs. Loss curves seem to be pretty similar for all the models, as well as the METEOR ending at about the same values.

The METEOR and BLEU scores for each model were:
model 1:
Best METEOR: 0.25426976184548733
Last METEOR: 0.2533475725527003
Last BLEU: 0.2734293963092999

model 2:
Best METEOR: 0.2552882439261155
Last METEOR: 0.2552882439261155
Last BLEU: 0.264689740228381

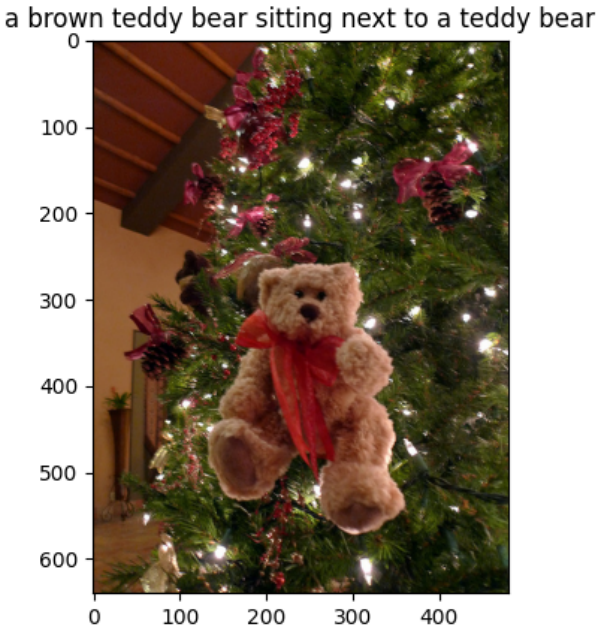
model 3:
Best METEOR: 0.2552186424397039
Last METEOR: 0.25323939192346034
Last BLEU: 0.2676470009787992

model 4:
Best METEOR: 0.25722736095009135
Last METEOR: 0.25501580615260927
Last BLEU: 0.2803739391936391

So model 4 with the complex LSTM layers performed the best, which was expected.

3 Images

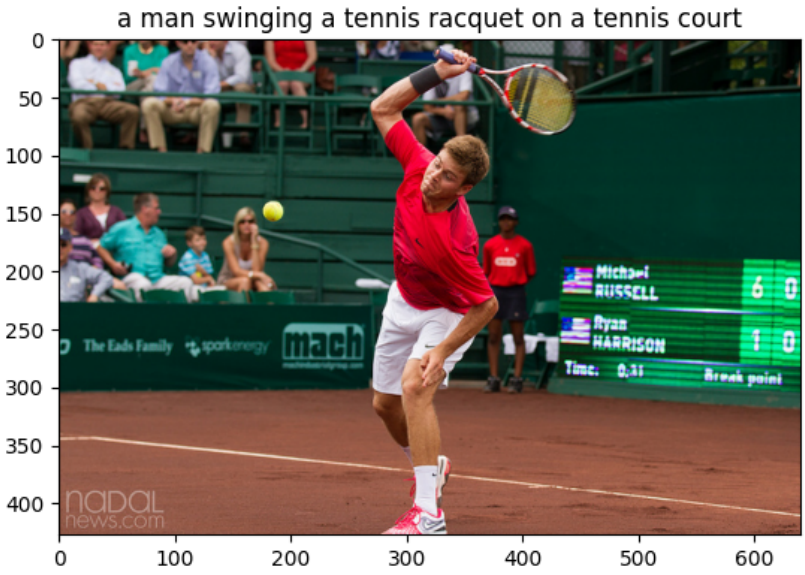
Here are some resulting images with generated captions.


a brown teddy bear sitting next to a teddy bear
0
100
200
300
400
500
600
0
100
200
300
400


a baseball player swinging a bat at a ball
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600


a refrigerator freezer with the door open and has food inside
0
100
200
300
400
0
100
200
300
400
500
600


a train traveling down tracks next to a forest
0
100
200
300
400
0
100
200
300
400
500
600



Some of the captions seem very well generated, like the train traveling down the tracks next to a forest, while other miss a bit, like the teddy bear on the Christmas tree, which was captioned 'a brown teddy bear sitting next to a teddy bear'. It still managed to see it was a brown teddy bear in the image, which is not too bad.

The generating of these images can be made with the validation file (check Readme). I added the model from task 2 to generate with.