# Theory Exercise 4

## 1.1 Task 1

### 1.1.1 Question

For 5 time steps

$$ (20 \cdot 128 + 128 \cdot 10 )\cdot 5 + 128 \cdot 4 = 19712. $$

For 10 time steps

$$ (20 \cdot 128 + 128 \cdot 10 )\cdot 10 + 128 \cdot 9 = 39552. $$

### 1.1.2 Question 2

No. The parameters are the same during a forward roll they are only used more times depending on the number of time setps you are taking. During a backward roll the parameter will change for every time step but no new parameters are added.

## 1.2 Task

### 1.2.1 Question 1

The vanishing or exploding gradient problem. Because you are multiplying with the same scalar every time step the value will vanish or explode if the scalar is not equal to 1.

### 1.2.2 Question 2

LSTM. Has a cell state that is kept, forgotten or added to in every time step and a hidden state that it passes to the next time step.

GRU. Only has a hidden state that it feeds forward or forgets.

## 2.1 Task 1

### 2.1.1 Question 1

It is something that we can train and the gates can learn. This leads it to be able to forget or make use of different things from the last time step without having to worry that much about the problems with the gradient.

### 2.1.2 Question 2

The LSTM will not remember anything from the last time step and now the cell state should only contain zeros.

### 2.1.3 Question 3

The problem can still occur but is now much less likely. If it occurs I think the problem arises in the forget gate and it is designed to forget things. But the biggest change is that now we are using the cell state and it is updated by ADDITION and not MULTIPLICATION. So no mather what the weights are they will not have that fast impact as they used to have in a normal RNN.

### 2.1.4 Question 4

As long as you have a hidden layer of a size bigger the 1 you can lower the parameters.

### 2.1.5 Question 5

It's a RNN with 2 LSTM layers and one of the layers is trained starting at the last time step and works it's way backwards instead of forward. This will help the network to predict context better.