ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2020 Prof. R. Wattenhofer

Computational Thinking
Exercise 12

1 Neural Networks on Small Devices

You are tasked with designing a neural network to classify pictures taken with a 8 megapixel phone camera, that is pictures of 3771×2121 pixels (each of which has a red, blue and green value). Obviously, the network should run on the phone which has 4GB of RAM and a sufficiently strong CPU. As you know that computations within a neural network do not have to be extremely precise, you choose to use an 8bit encoding for floating point numbers.

a) Calculate the memory requirement of an MLP that simply flattens the image, passes it to a hidden layer of 256 nodes and then to an output layer consisting of 10 output nodes (representing one class each). Do you see an issue?

b) In the lecture you learned about convolutional neural networks (CNNs). Give two reasons why a CNN might be useful here.

c) Given a network of 10 CNN layers, each of which has 128 3×3 filters for each input channel, what is the memory requirement to store the network weights? You can assume that the output classification layer has a negligible amount of parameters.[1] **Hint:** The number of output channels is equal to the number of input channels for the next layer.

d) The output of a convolutional layer with n filters and stride 1, i.e. how many positions the filter slides each time, is a tensor of shape $[b, H, W, n]$ where b is the batch size (number of training examples in the batch) and H and W are the height and width of the input image. What memory issue do we face if we try to train the 10 layer CNN from before on full resolution images with a batch size of 1 image at a time on the given hardware? **Hint:** Think about the steps that we need to take during training.

e) What solution do you propose?

2 Zurich's Temperature

You are required to predict the temperature in Zurich from satellite images. You have a dataset for training that consists of RGB images of resolution 128×128 of different parts of Zurich taken every hour and labeled with the corresponding temperature. The dataset spans over a time of 1 year. Your model should take one of these images as input and output the temperature in Celsius.

a) First, you decide to use as model an MLP that receives as input a flattened image. However, you observe a very poor test performance. What is most likely the problem with your model? How would would you modify it?

[1] This can be achieved by taking the average of the output tensor over the image dimensions, followed by a 128*10 fully connected layer.

b) After changing your model, you observe that the performance improved, but it is still far from what you need. You decide to examine your data and you find that the model produces wrong temperature values for images with snow. You decide to include your knowledge in the network architecture: for each image you count the amount of white pixels, i.e., pixels with intensity above a certain threshold value. You wish to train this threshold, so you set it as a trainable parameter. The count you provide as an additional input to your model, a feature which you hope the network can pic up on to make a better prediction. List technical issues with this approach.

c) Next you notice that you are processing the images individually, however they form a sequence, i.e., there is a temporal ordering. How could you modify the architecture so that you include this information?

d) Still your network is not performing well enough, but you get access to an additional dataset consisting of satellite images of Zurich taken over a span of 5 years. However, these images do not have temperature labels. How could you change the architecture so that you can use these images to improve your performance?

3 Short Sighted Agent

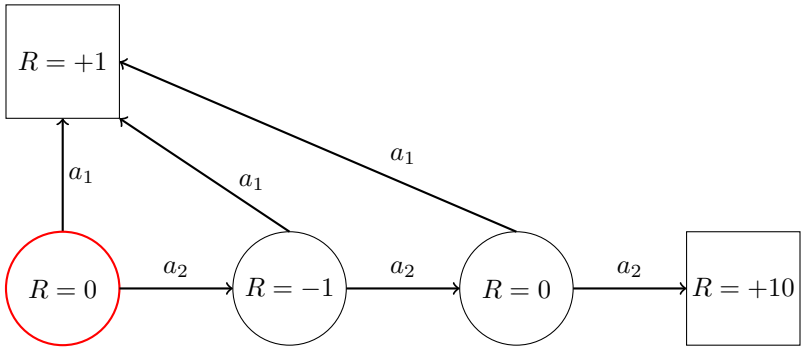

Figure 1: A simple deterministic MDP. The rectangles are terminal states, the red circle is the initial state.

Figure 1 shows a simple deterministic Markov decision process (MDP). Compared to the example from the lecture we give the rewards here on the states, and not on the transitions. We seek to investigate the effect of the discount factor γ for this given task.

a) What is the optimal policy if we wish to maximize the undiscounted cumulative reward $\sum_t R(s_t)$?

b) Use the value iteration algorithm to calculate the optimal value V_{π^*} of the three non-terminal states for $\gamma = 0.1$. **Hint:** It may help to do the calculations in a specific order.

c) What is the optimal strategy for $\gamma = 0.1$?

d) For which values of γ does the optimal policy end up in the top terminal state with $R = +1$?