Deep Learning Lab Course

Christian Zimmermann and Silvio Galesso

Due: 8. May, 2019

Outline The goal of this exercise is to understand main concepts and core components of Semantic Segmentation and Human Pose Estimation using Deep Learning methods. We provide you with a reduced and simplified version of the COCO[1] dataset, with according data readers and a simple baseline network architecture. Based on the code and data provided we ask for a series of experiments that will make implementation of additional code necessary. To pass this exercise create a short report summarizing your findings that can not exceed 2 pages in length as well as the source code used to create the results and submit it trough our Ilias group.[2] For questions during the exercise please use the Slack channel.[3] The report must include the following items:

- Task1: A diagram showing how the MPJPE[4] evolves over training epochs for the training and evaluation set starting from ImageNet initialization.
- Task1: How is final performance influenced by the ImageNet initialization of training?
- Task2: Provide an qualitative and quantitative comparison between the scalar regression and the softargmax variant.
- Task2: Give some qualitative results for the predicted pose overlayed on the input image and the latent heat map representations the network learned. Discuss common failure cases and overfitting.
- Task3: Compare the three listed network architectures and give qualitative examples.

1 Task 1

First, we ask you to train and evaluate the given network architecture on the prepared dataset. In the first task we want you to use the direct scalar regression

[1] `http://cocodataset.org`

[2] `https://ilias.uni-freiburg.de/goto.php?target=crs_1275088&client_id=unifreiburg`

[3] `http://dl-lab.informatik.uni-freiburg.de/`

[4] Mean per joint position error, see lecture for details.

approach that is already implemented in the given code base. For this purpose, setup up the environment according to the given instructions [5]. Next, you need to implement a training procedure which may include, but is not limited to, defining the loss, setting up the optimizer and saving snapshots of the network weights while training. You also have to think of an evaluation strategy that allows you to report how the networks MPJPE performance evolves during training on both the training and the withhold evaluation set.

We recommend the following setting

- Use an squared L2 loss to formulate the training objective. Make sure missing keypoint information is handeled accordingly.
- ADAM solver with a learning rate in the range of 1e-4 ususally is a good starting point.

You should be able to archieve below 13 pixel MPJPE on the evaluation set with this setup.

2 Task 2

For the second task you should implement the Softargmax loss and adapt the network architecture accordingly. With this loss formulation you can reach below 11 pixel MPJPE. Compare the two network architectures both in terms of MPJPE and qualitatively.

3 Task 3

For the last task you should implement an encoder-decoder network for semantic segmentation, using a cross entropy loss. For the encoder you can use the pretrained fully-convolutional ResNet model that was used as the base network in Task 1. For the decoder, we ask you to implement and train three different variants:

1. A single upsampling layer as decoder[6].
2. A convolutional decoder (at least 3 layers, using transposed convolutions)
3. A convolutional decoder as above, but with skip connections from the encoder.

For each variant you should provide quantitative (IoU) and qualitative validation results, and provide a comparison between them. Note that for a meaningful comparison between the two

[5] `https://github.com/aisrobots/dl-lab-ss19/blob/master/exercise1_CV/code/README.md`

[6] You can use a 1×1 convolutional layer to adapt the number of channels.

convolutional decoders (i.e. 2 and 3) is only possible when the two have similar architectures and number of parameters. The models should be trained for a reasonable time, but you should be able to draw your conclusions already after a few (<10) training epochs.