Lessons learned of working on Deep Prediction Learning – Phillip Lippe – 08/18/2017

Lessons learned

Deep Tracking

- Comparing recurrent convolutions (as in the DeepTracking paper) and ConvLSTMs on the Deep Tracking dataset, the ConvLSTMs clearly outperform

- Deep Tracking does not properly work with RGB because there is no default value for “free space”

- Occlusion is one of the main problems for predicting objects. Deep Tracking gives a good start point but unseen objects that come out of the occlusion remain as an unsolved challenge

(Video) prediction

- Networks only consisting of multiple stacked ConvLSTMs like Chelsa Finns Video Prediction are not capable to predict whole RGB images as next frame. The problem is that only some parts of the image are changed in the next frame and therefore does not strongly depend on the past.

- For generating realistic RGB images use a GAN. Even it is hard to train sometimes you just need luck. Tips and tricks:
  - Look at https://github.com/soumith/ganhacks, have some nice suggestions how to stabilize. It depends on the network if some tricks are working or not
  - For discriminator with many time frames input try to insert recurrence. It mostly helps.
  - Scale down the image as much as possible in the discriminator so that the last fully connected layer is not too huge.
  - Train the discriminator and generator similarly or for example discriminator twice as much. Don't try to balance the loss via statistics (for example if the loss of the discriminator is to high, than stop training generator until it goes down again)
  - Discriminator and generator have to be trained separately. Make sure that the gradients calculated for the discriminator, does not flow through the generator and other way round.

- The more input you give to the prediction network the more it would have to predict by itself (to feed it back in while predicting). Alternatively take a default/most likely value but remind yourself that the network can then distinguish between prediction and seeing ground truth.

- Predicting a rotation is not possible/very very hard with a neural network because there is no standard filter that can be applied on the whole image (like the translation just requires one single filter over the whole image). Moreover the network has to learn a different filter for every pixel what makes it hard to predict it

- Loss calculation is the most important part of your network when it has to solve multiple different challenges.