# HW2, Vlad Seremet, seremetv@

Code : https://www.github.com/shervlad/hw2/

## Problem 1

a) source code is in hw2/train_inverse_model.py

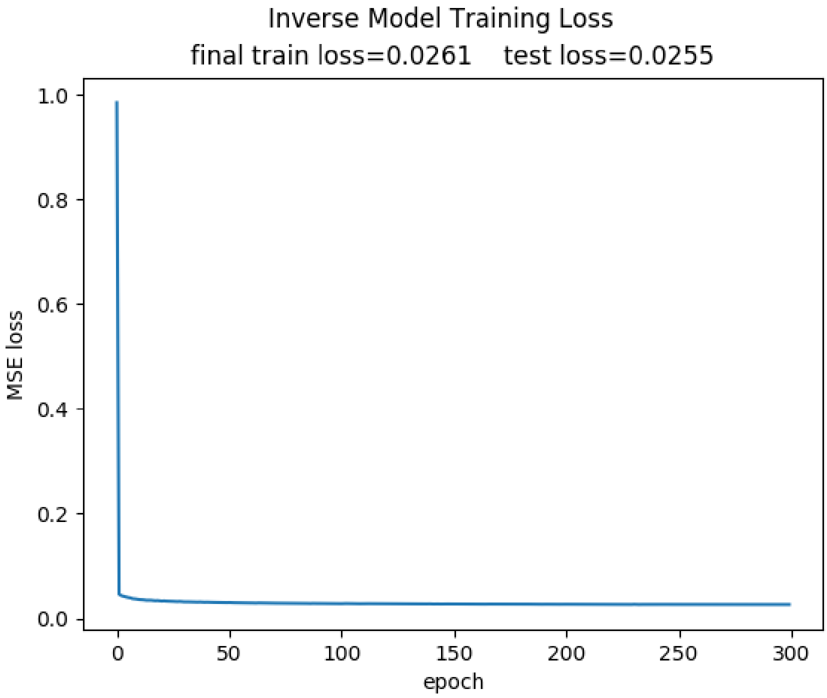
b) The model is a neural net with 3 hidden layers, 64 nodes each.

The input dimension is 4 (init_ob + goal_obj) and the output is 4 (push)

Mean Squared Error is used for the loss function.

Adam is used for updating the parameters.

c) Plot:



d) Video is in hw2/videos/plan_inverse_model.mp4

Distances from goal:

[0.0126, 0.0079, 0.0085, 0.0082, 0.0084, 0.0076, 0.0067, 0.0079, 0.0081, 0.0077]

## Problem 2

a) source code is in hw2/train_forward_model.py

b) At the heart of the model is a neural net just like for the inverse model except

the input has dimension 6 (init_ob + push) and the output has dimension 2 (goal_obj)

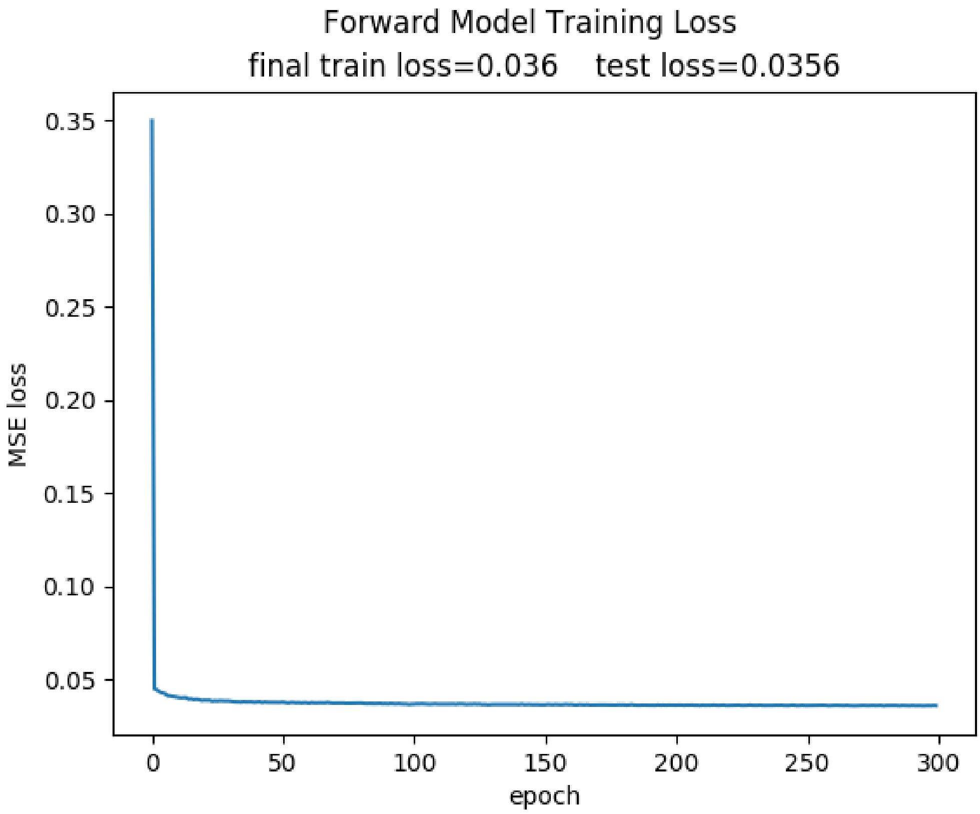
To infer an action from (init_obj, goal_obj), the CEM algorithm is used.

We start with a normal distribution over push_angle and push_length.

For 200 iterations, we take 200 samples, convert the to actions and calculate their performace.

We pick the top 20, and their mean and std become the mean and std of the normal distribution.

c) Plot



d) Video is in hw2/videos/plan_forward_mode.mp4

Distances from goal:

[0.0302, 0.0306, 0.0307, 0.0312, 0.0318, 0.031, 0.0321, 0.033, 0.0352, 0.0313]

## Problem 3

a) Video is in hw2/videos/plan_extrapolate_inverse_model.mp4

Distances from goal:

[0.0259, 0.0294, 0.0178, 0.0250, 0.0231, 0.0284, 0.0279, 0.0293, 0.0216, 0.0193]

b) Video is in ./videos/plan_extrapolate_forward_model.mp4

Distances from goal:

[0.0351, 0.0184, 0.0134, 0.0217, 0.0641, 0.0355, 0.0336, 0.0276, 0.0137, 0.0402]