Deep learning for stochastic control

Jonathan Haag and Wille Viman Vestberg

Model Description

Consider the following model for a warehouse

$$s_{t+1} = s_t + p_t - d_{t+1} \tag{1}$$

where s_t is the stock at time t, p_t is the amount purchased and d_t is a random variable describing the demand, i.e, how many parts are sold at time t. We want to find a control law

$$p_t = f(s_t),$$

that is how much to buy given the current stock in order to minimize our overall expenses. The cost at time t are described by

$$c_t = g(t)p_t$$

where $g(t)$ is a function describing the purchase cost per item at time t and thus the cumulative cost at a time τ are given by

$$C_\tau = \sum_{t=0}^{\tau} c_t.$$

For the purpose of this project we defined

$$g(t) = \begin{cases} 1, & \text{for } t \leq 5 \\ 3, & \text{for } t > 5. \end{cases}$$

Note that we assume that we can sell each item for a fixed cost, i.e., our income only depends on the demand and is thus not further considered here.

To ensure that the model is realistic, we introduce a set of constraints

$$\begin{aligned} &s_t \in [0, s_{max}], \text{ where } s_{max} \text{ is the maximum storage capacity,} \\ &p_t \in [0, p_{max}], \text{ where } p_{max} \text{ is the maximum purchase amount,} \\ &s_t + p_t \geq d_{t+1}, \text{which ensures that the demand is always met.} \end{aligned}$$

We can reformulate these as five inequality constraints of form

$$h_i(s_t, p_t) \geq 0, \quad i = 1, ..., 5. \tag{2}$$

Problem formulation

The cost minimization problem for finite horizon T can be stated as a optimization problem of the form

$$\begin{aligned} &\min_{p_t, t=0,\ldots,T-1} \mathbb{E}\{\sum_{t=0}^{T-1} g(t)p_t + c_T | s_0\} \\ &\text{s.t. } (1) \text{ and } (2), \end{aligned} \tag{3}$$

where we want to find actions that minimize the expected total cost given an initial state s_0 while satisfying the system dynamics and the constraints. Here, we add a final cost c_T to the purchasing cost for the T time steps from before to obtain the total cost.

Deep learning approach

The key observation is that the control law can be approximated by a feedforward neural network that approximately solves the minimization problem (3). Thus we setup a network of T subnetworks, each of which is a fully connected N-hidden layer neural network with one input and one output node. The idea is that subnetwork t takes a state s_t and predicts the optimal purchase p_t which, together with the demand, is then used to update the state accordingly.

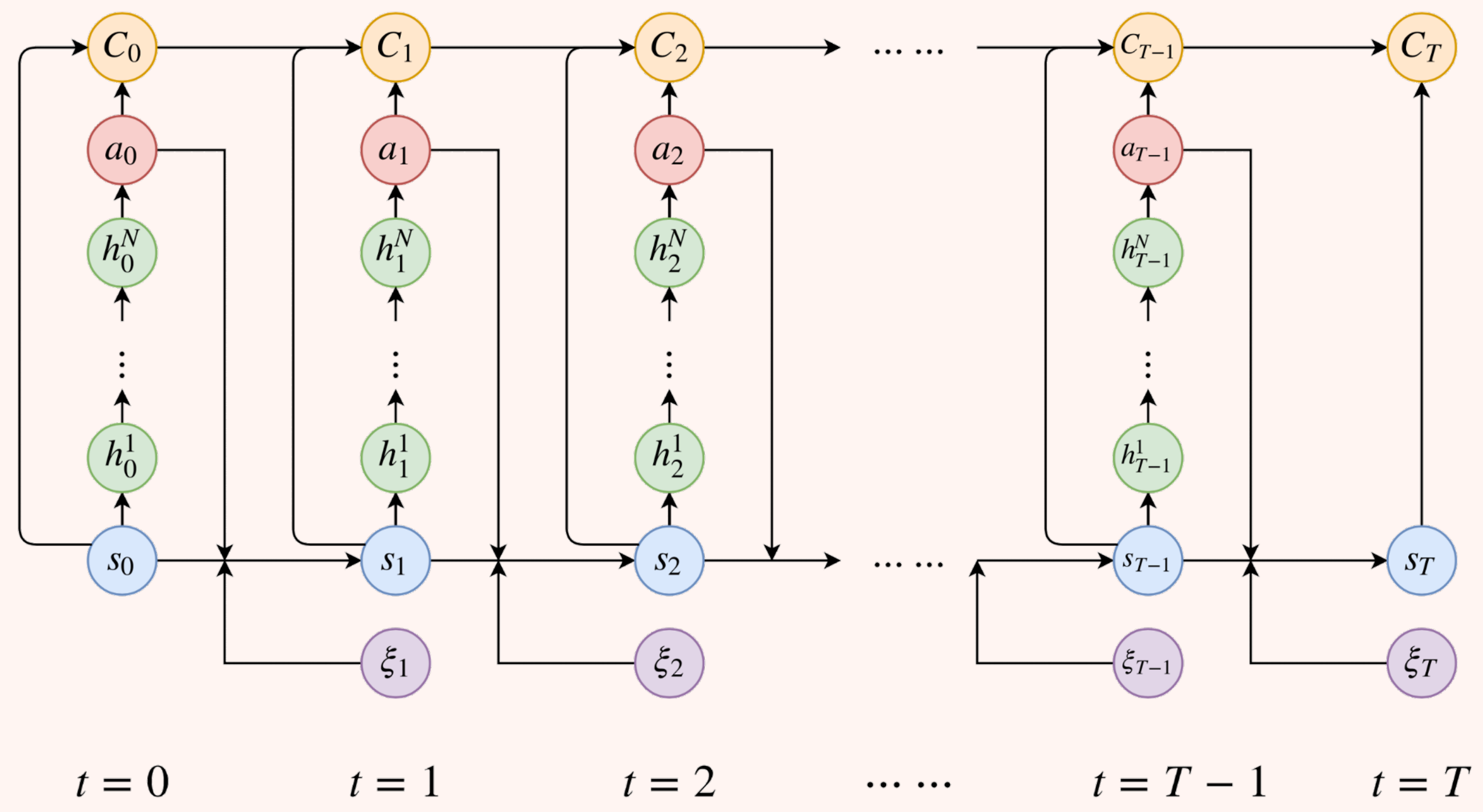

Figure 1. The used network architecture as established in [1]. Notice that they use ξ and a where we use d and p.

We use the cost function as before but add penalty terms to ensure constraint satisfaction, such that the full cost is given by

$$C_{total} = \sum_{t=0}^{T-1}(g(t)p_t + \sum_{i=1}^{5} \gamma_i \cdot abs(min\{0, h_i(s_t, p_t)\})) + C_T.$$

where the final cost C_T only ensure constraint satisfaction. As we set out target values to zero this corresponds to a mean absolute error as loss function.

Implementation details

The neural network architecture was implemented in TensorFlow using two hidden layers with 50 units in each subnetwork. As the authors of [1] suggest, ReLu was used as activiation function for all layers and batch normalization was applied after the linear transform but before the activation. The network is trained with the Adam optimizer for 1000 epochs with a learning rate of 0.003. With a batch size of 50 and 20000 total iteration steps, the training set is of size 1000.

Each training sample consists of the initial value $s_0 = 50$ and a sequence of demands of length T sampled from a uniform distribution with lower bound 5 and upper bound 35. The limits of the storage and purchase where set to $s_{max} = 100$ and $p_{max} = 30$. The penalty parameters γ_i for the constraints are all set to 10^4.

Example results

The suggested model was tested for different end times $T = 10, 15, 20$ resulting in differently sized networks.

Learning

Figure 2 clearly shows a two-stage learning process: An initial period of around 25 epochs with many constraint violations and high losses, followed by a long slow convergence phase with occasional spikes when the model gets too aggressive. The learning is very similar for different T, however computation time increases from 119 to 172 and finally 230 seconds for increasing T.

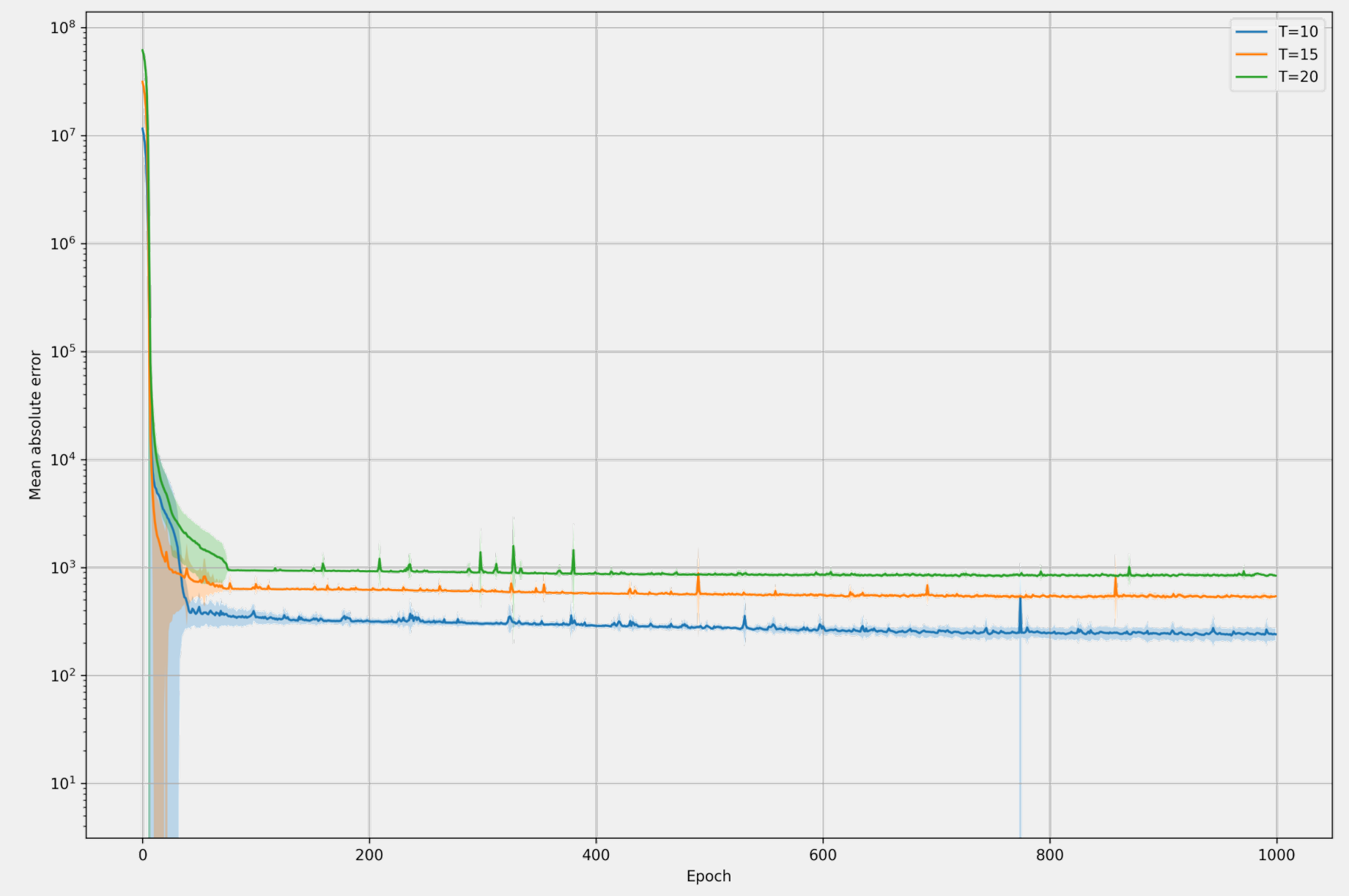

Figure 2. Training loss (mean absolute error) for different T and 5 different random seeds. Shown is always the obtained mean ± the standard deviation.

Trajectory

Figure 3 shows one obtained cost and prediction for $T = 15$. We can easily see that the constraints are satisfied at all times and that the network is able to take advantage of the cheaper prices early on. This indicates validity of the found solution.

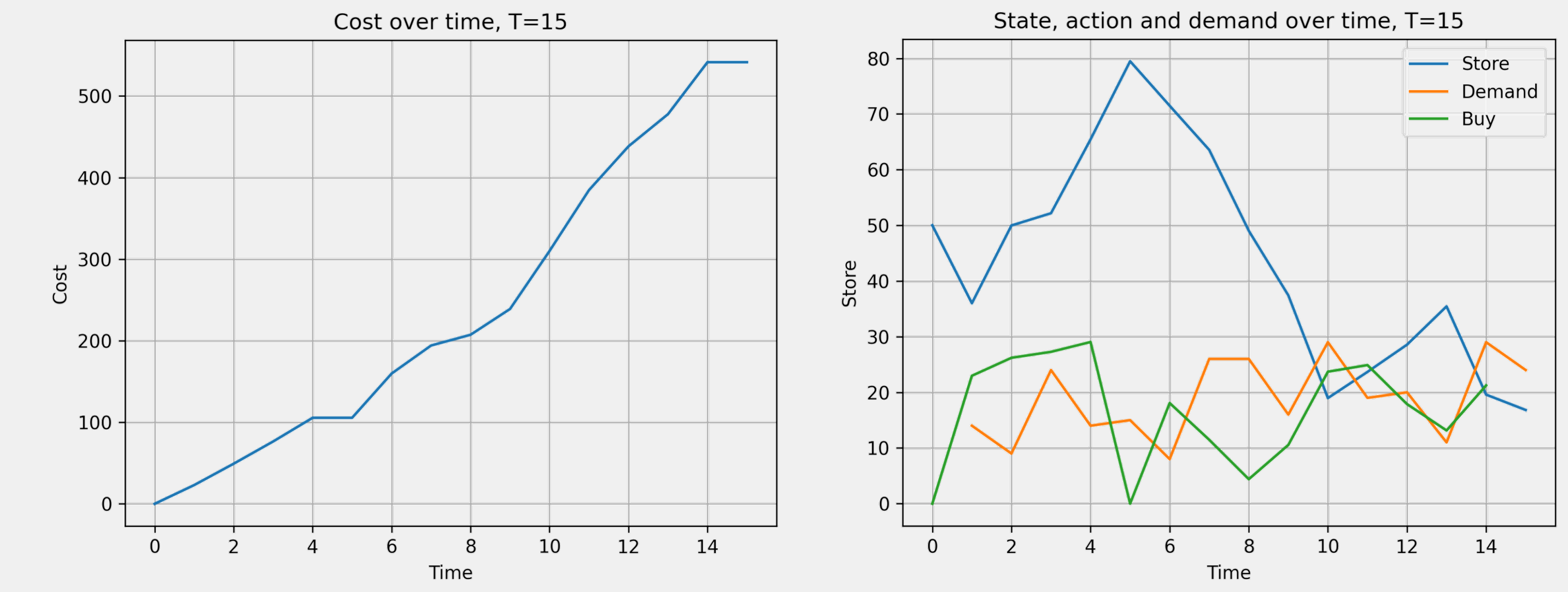

Figure 3. Cumulative cost (left) and state, purchase and demand over time (right) for $T = 15$.

References

[1] Jiequn Han et al. Deep learning approximation for stochastic control problems. *arXiv preprint arXiv:1611.07422*, 2016.

Github: https://github.com/jonahaag/DL-Approx-Stochastic-Control

jmhaag@kth.se, willev@kth.se