Last Time

- ☐ RL approx
- ☐ SSL approx

Lecture 7

EAIS S'25

Andrea Bajcsy

This Time:

- ☐ wrap up SSL approx
- ☐ online updates!

Announcement: (1) don't forget to talk to us about course project before spring break to get xtra credit ⇒ fill out canvas assignment

(2) paper reading day on Wed! Submit summaries before class

Recap: Last time we talked about "neural reachable tubes" as approximations to the safe set + safety policy. (i.e. value function!)

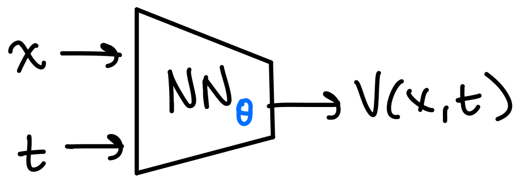
We covered $\boxed{\text{RL approximations}}$ by reformulating the safety Bellman backup to be a contraction mapping

$\Rightarrow$ this unlocked RL algorithms (e.g. Q-learning, REINFORCE ....)

$$V_\theta(x) = (1-\gamma)\,\ell(x) + \gamma \min \left\{ \ell(x), \max_{u \in \mathcal{U}} V_\theta(f(x,u)) \right\}$$

We also discussed $\boxed{\text{SSL approximations}}$ where the main trick was how to fit $V_\theta(x,t)$ when we do NOT have labels!

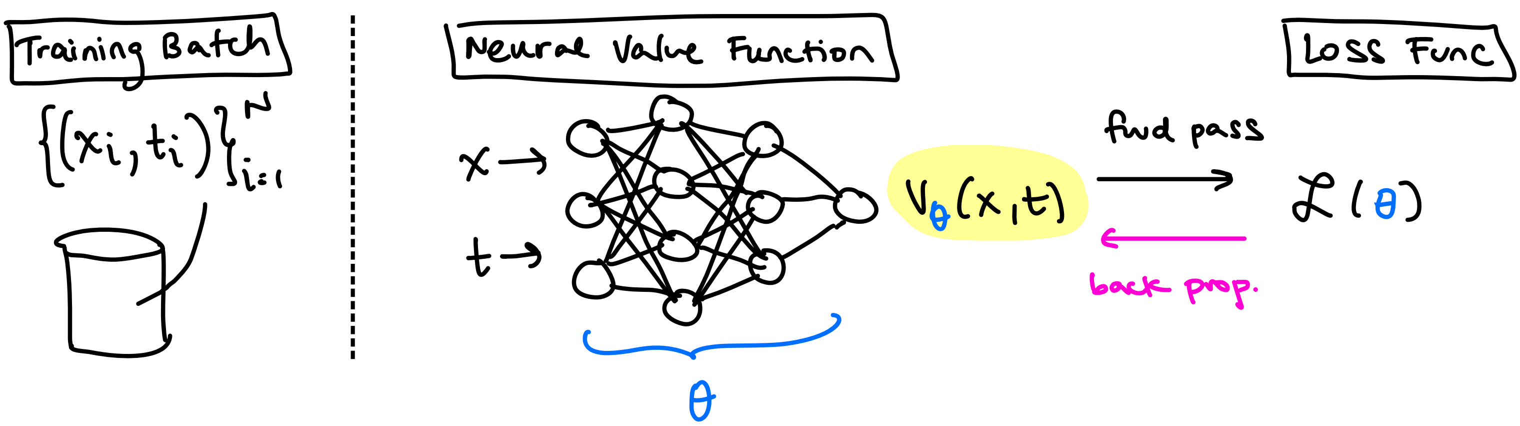
e.g.



; $\mathcal{D} = \{(x_i, t_i, V_i)\}_i$ — training data

If I was doing supervised learning, I'd need labels of V_i for each x_i, t_i!

! KEY IDEA: Use the HJB-VI as the signal which automatically tells you if the $V_\theta(x,t)$ is valid!



the HJ-VI should = 0! $V(x,T) = \ell(x)$

$$\mathcal{L}(x_i, t_i; \theta) = \text{HJ-VI Violation Error} + \lambda \text{ Initial Condition}$$

$$= \mathcal{L}_1(x_i, t_i; \theta) + \lambda \mathcal{L}_2(x_i, t_i; \theta)$$

$$\mathcal{L}_1 = \sum_i \underbrace{\left\| \min\left\{ \ell(x_i) - V_\theta(x_i,t_i), \frac{\partial V_\theta(x_i,t_i)}{\partial t} + \max_u \nabla_x V_\theta(x_i,t_i)^T f(x_i,u) \right\} \right\|}$$

we know this should = 0 when V is optimal ⇒ bigger the norm, the higher the loss! ⇒ This is HJI-VI!

$$\mathcal{L}_2 = \sum_i \|V_\theta(x_i,t_i) - \ell(x_i)\| \mathbb{1}\{t_i = T\}$$

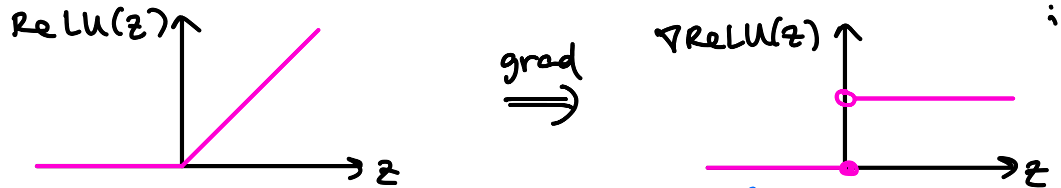
indicator

ground truth value only available @ final time

<u>update rule</u>: $\theta \leftarrow \theta + \nabla \mathcal{L}(\cdot\,; \theta)$

The loss depends on the time $\left(\frac{\partial V_\theta}{\partial t}\right)$ and the spatial $(\nabla_x V_\theta)$ gradients of the value function! Thus, our NN should represent both the value function + its gradients well

Popular activation: $\text{ReLU} = \max\{z, 0\}$ is <u>piecewise linear</u>, but gradient is <u>piecewise constant</u>!

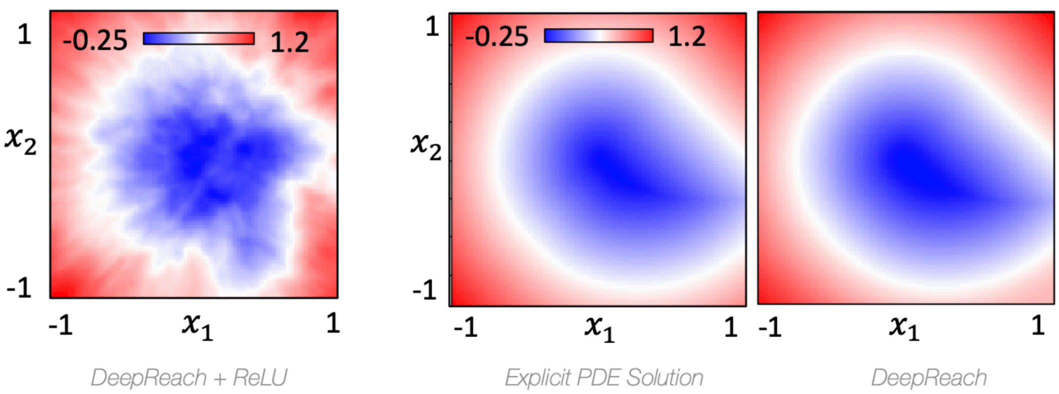


↑ struggle to represent $\frac{\partial V}{\partial t}$, $\nabla_x V(x,t)$

💡 <u>IDEA</u>: use sinusoidal activations in the NN value function!

⇒ Also helps with opt. ctrl: $u^*(x,t) = \arg\max_u \nabla_x V_\theta(x,t)^T f(x,u)$

Value Function Comparison



Comparison graphic
by Somil Bansal (USC).