

PyTorch and Neural Nets

CS285 Deep RL

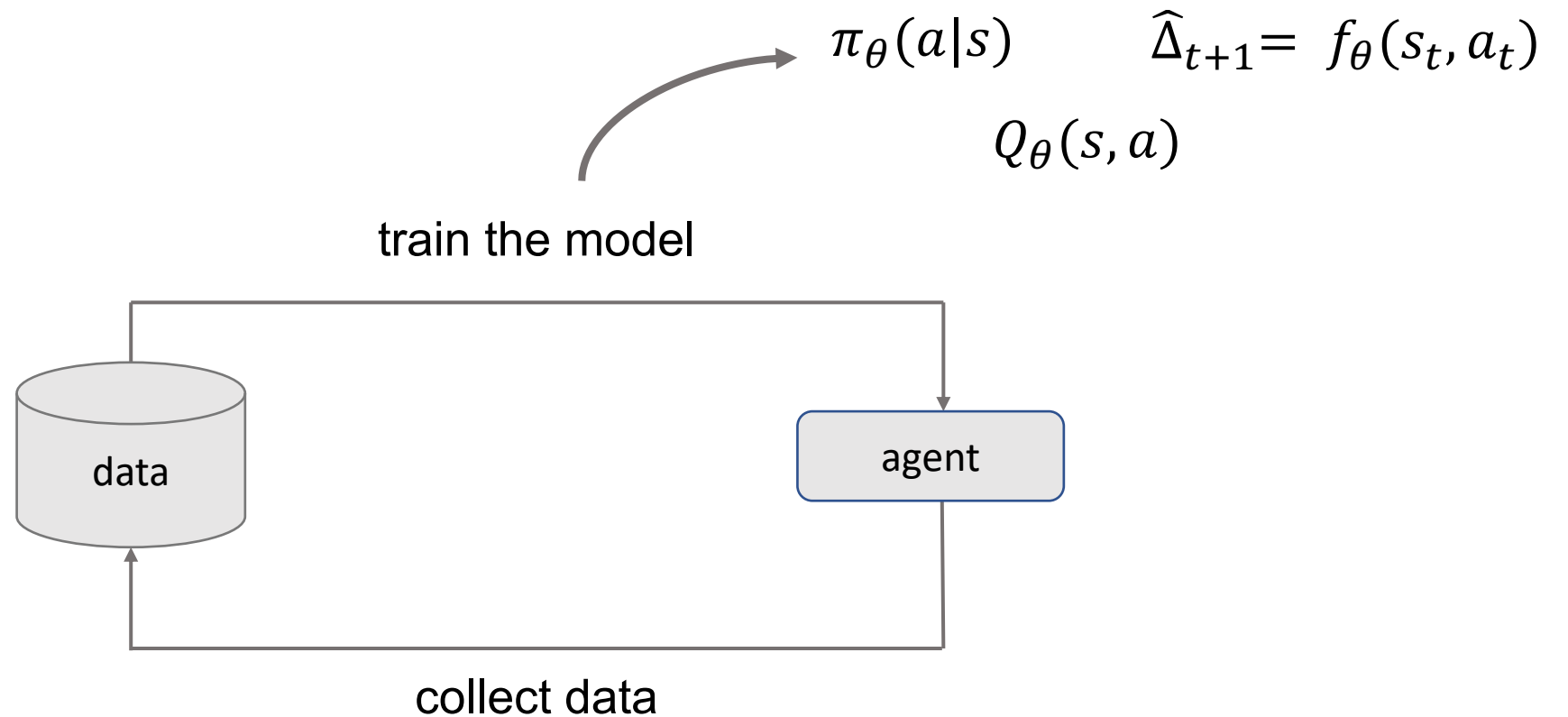
Instructor: Marwa Abdulhai



[Adapted from Kevin Li's CS285 Fa21 Slides]

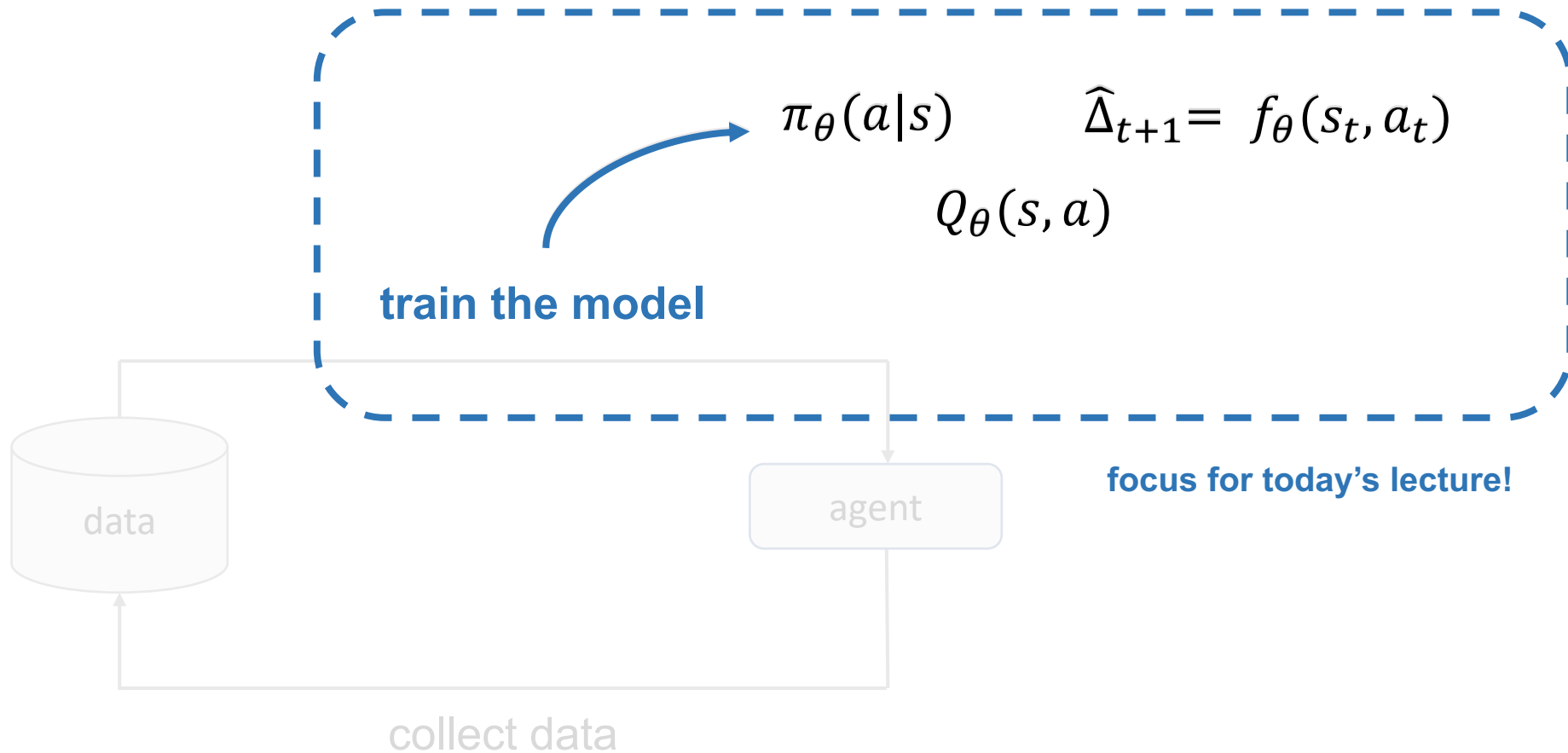
Goal of this course

Train an agent to perform useful tasks



Goal of this course

Train an agent to perform useful tasks



How do train a model?

$$\theta^* = \text{arg min}_{\theta} \sum_{(x,y) \in D} \mathcal{L}(f_{\theta}(x), y)$$

gradient descent

dataset

loss

neural network

PyTorch does all of these!

What is PyTorch?

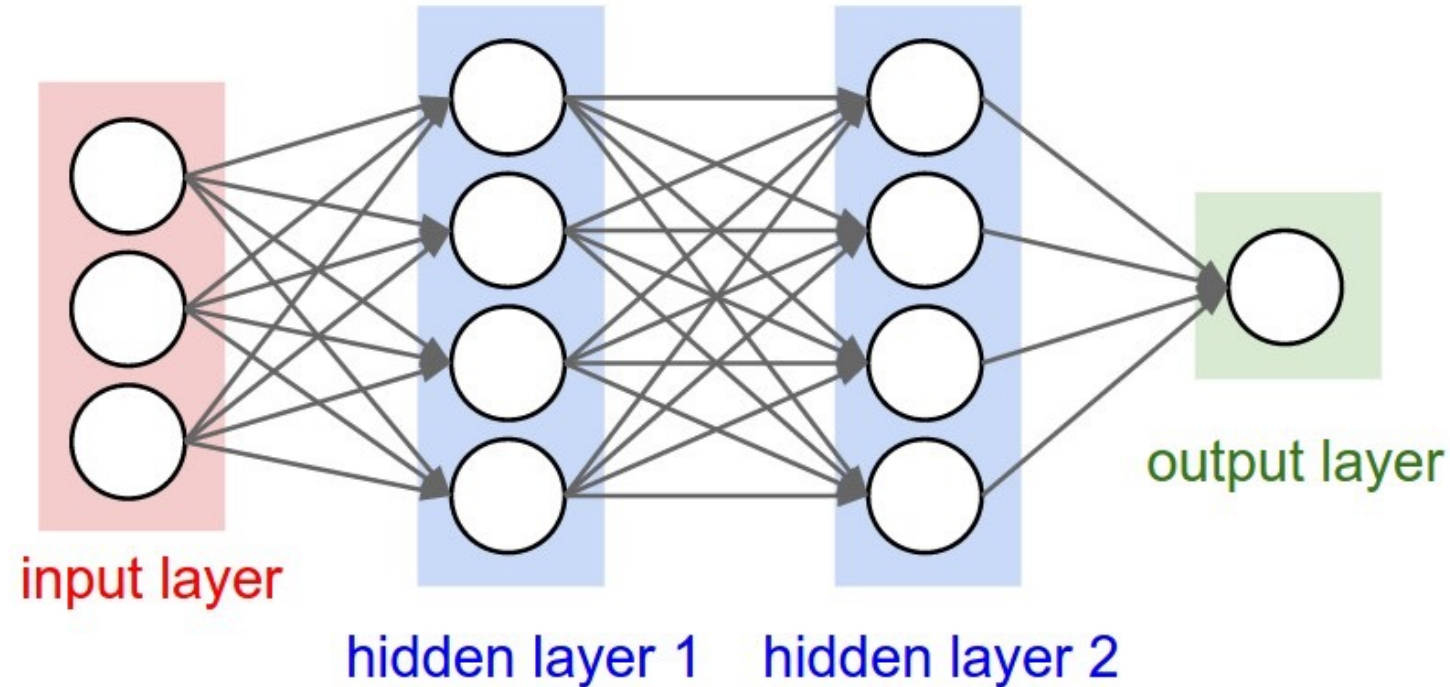
Python library for:

- Defining neural networks
- Automating computing gradients
- And more! (datasets, optimizers, GPUs, etc.)

$$\theta^* = \underset{\theta}{\operatorname{arg\,min}} \sum_{(x,y) \in D} \mathcal{L}(f_{\theta}(x), y)$$

The diagram illustrates the components of the equation. A green oval around $\operatorname{arg\,min}_{\theta}$ has a green arrow pointing to the text "gradient descent". A blue oval around $\sum_{(x,y) \in D}$ has a blue arrow pointing to the text "dataset". A red oval around \mathcal{L} has a red arrow pointing to the text "loss". An orange oval around f_{θ} has an orange arrow pointing to the text "neural network".

How does PyTorch work?



You define:

$$h_1 = \sigma(W_1 x) \quad h_2 = \sigma(W_2 h_1) \quad y = \sigma(W_3 h_2)$$

PyTorch computes:

$$\frac{\partial y}{\partial W_1} = \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_1} \quad \frac{\partial y}{\partial W_2} = \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial W_2} \quad \frac{\partial y}{\partial W_3}$$

PyTorch Tutorial (Colab)

https://colab.research.google.com/drive/1XQu1mUbGtvkQY-D7_YCOZIRzSnjp4u9f?usp=sharing

<https://bit.ly/3CM6lcf>