In this video you’ll learn how to:

1. Create OpenAI Gym environments like CartPole

2. Build a Deep Learning model for Reinforcement Learning using Tensorflow and Keras

3. Train a Reinforcement Learning model using Deep Q Policy based learning using Keras-RL

Github Repo for the Project: https://github.com/nicknochnack/Tenso...​

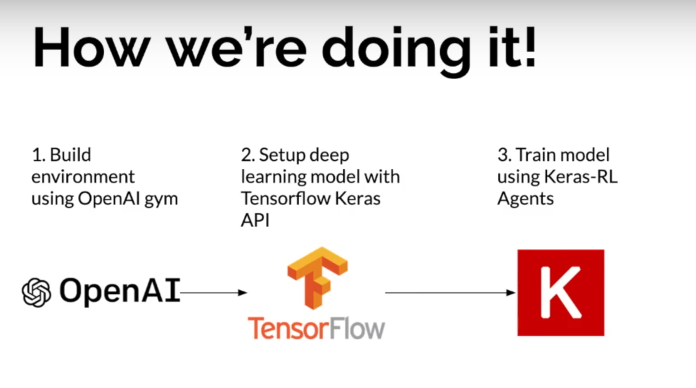
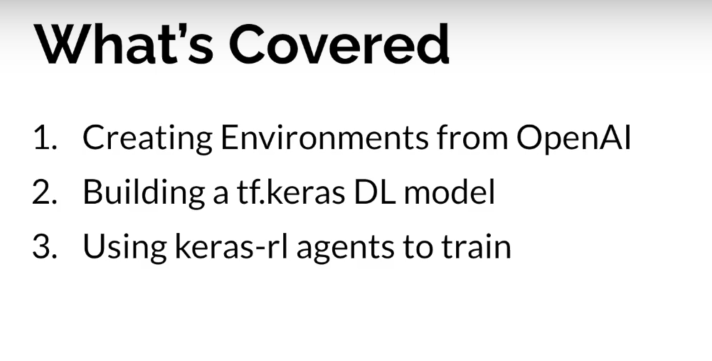
Want to learn more about it all:

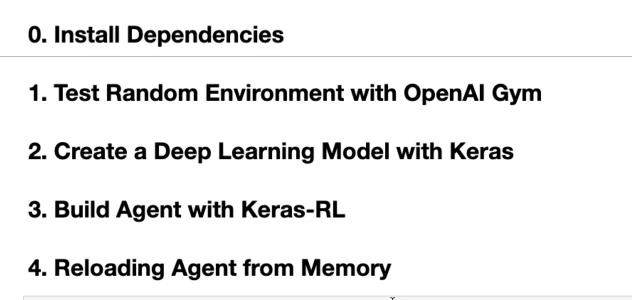
Open AI Gym: <https://gym.openai.com/envs/>​

Keras RL: <https://keras-rl.readthedocs.io/>

AREA:

* Action
* Reward
* Environment
* Agent





* Install OpenAI’s gym
* Keras Reinforcement Learning Library

**Test Random Environment with OpenAI Gym**

* **OpenAI Gym** comes with a bunch of prebuilt environment – to test RL
* Import gym and random
* Set up Environment
  + Extract the states available within the environment
  + Extract Actions
* Build Random Environment
  + Render the Environment where the cart in action is moving left and right 🡪 env.render()
  + Action 🡪 0 or 1 🡪 left or right (Random choice to see how it impacts the environment)
  + env.step(action) 🡪 Applying the action to env to get some results
  + Based on our steps we get rewards
  + If we fail 🡪 done will be set to true

**Create a Deep Learning Model with Keras**

* Deep Learning Model is going to learn the best action to take
* Pass 2 arguments 🡪 States , actions
* Flat node with 4 different states
* Dense layer 🡪 action (Linear)

**Build Agent with Keras-RL**

* Take the DL Model created and train in using Keras RL
* Keras Agent selected : DQNAgent (test with other available agents)
* Policy : BoltzmannQPolicy
  + Value based RL/ Policy based RL
* SequentialMemory class is used to maintain the memory
* Setup the agent and wrap it inside the function and we can reload it in the memory
  + Names the function Build\_Agent 🡪 pass DL model and action
  + Setup policy
  + Setup memory
  + Setup DQNAgent 🡪 passed the DL model, policy…
* Train the RL Model using dqn
  + Instantiate Dqn model – compile – configure
  + Mae -metrics 🡪 mean absolute error
* dqn.test method can be used to visualize the episodes

**Reloading Agent from Memory**

* Save and deploy the model
* Del the previously created model, env, dqnAgent
* Save the weights from dqn model and rebuild it by reloading the weights
* Save\_weights 🡪 Generated 2 .h5 files
* Re-Instantiated the model

Source:

<https://www.youtube.com/watch?v=cO5g5qLrLSo&list=PLgNJO2hghbmjlE6cuKMws2ejC54BTAaWV>