Implementing Cross Entropy Method for TensorForce

Tom Brady
TensorForce

- Open Source (Apache 2.0) Reinforcement Learning library
- Built on top of TensorFlow and compatible with Python 2.7 and >3.5
- Goal: clear APIs, readability and modularisation
- Differentiator:
  - "strict separation of environments, agents and update logic that facilitates usage in non-simulation environments"
  - Everything optionally configurable to be able to quickly experiment with new models.
- Integrates with OpenAI Gym API, OpenAI Universe, DeepMind lab, ALE and Maze explorer

* Find out more: https://github.com/reinforceio/tensorforce
Sample Usage

- Clear APIs
- Readable
- Modular

```python
from tensorflow.agents import PPOAgent

# Create a Proximal Policy Optimization agent
agent = PPOAgent(
    states_spec=dict(type='float', shape=(10,)),
    actions_spec=dict(type='int', num_actions=10),
    network_spec=[
        dict(type='dense', size=64),
        dict(type='dense', size=64)
    ],
    batch_size=1000,
    step_optimizer=dict(
        type='adam',
        learning_rate=1e-4
    )
)

# Get new data from somewhere, e.g. a client to a web app
client = MyClient('http://127.0.0.1', 8080)

# Poll new state from client
state = client.get_state()

# Get prediction from agent, execute
action = agent.act(state)
reward = client.execute(action)

# Add experience, agent automatically updates model according to batch size
agent.observe(reward=reward, terminal=False)
```
Cross Entropy Method

- Probabilistic Stochastic Optimization Method
- Neural network parametrizes the distribution of solutions
- Intuition: Iteratively sampling and refining a distribution of solutions
- High Level Procedure:
  - Assume a distribution of the problem space (e.g. Gaussian, with specified mean and variance)
  - While not converged:
    - Sample domain by generating candidate solutions from distribution
    - Evaluate the generated candidates
    - Update distribution based on the better candidate solutions discovered, minimizing the cross entropy
- Open source implementations available (e.g. https://github.com/rll/rllab/blob/master/rllab/algos/cem.py)
Aim: Implement X-Entropy Method for TensorForce

- **Goal**: Implement Cross Entropy pure TensorFlow in the TensorForce architecture
  - Following TensorFlow’s philosophy: clear APIs, readability and modularisation
  - Allow for experimentation with and deployment of RL models using X-entropy method using TensorForce

- **Validation**: Run x-entropy method on a simple OpenAI gym environment (e.g. CartPole)
  - Compare performance to other methods
Getting to the Goal

**Goal:** Implement Cross Entropy pure TensorFlow in the TensorForce architecture

Very little done so far & very little planned to do in the next week.

From Monday onwards - I have a plan!

- **Analysis**
  - Reading about Cross Entropy Method
  - Reading through TensorForce source, familiarizing myself with architecture
- **Cross Entropy in TensorForce**
- **Test implementation on a simple OpenAI gym environment (e.g. CartPole)**
  - Compare performance to other methods
- **Hopefully get a PR merged into TensorForce to give this functionality to users**
Thank you.

Questions?