# RL with factored States and Actions

Anirudh Goyal

# Problem Trying to solve!

- Estimating the value of a state-action pair in large state and action spaces.
- Selecting good actions given these estimates.
- Find successful policies in problems with high-dimensional state or action spaces.
    - Advantageous to learn representations of the state and action spaces that enable a policy to achieve high performance.

# Energy based Policies

In order to use Energy based policies, necessary to address several challenges.

- Parameterized representations can be highly nonlinear, leading to non-convex objectives with multiple optima
- Often intractable to compute the partition function.
- High variance since the performance depends on accumulated rewards.
- Address this issue, by using Value based RL.

# Value based RL

- Approximate the action value function by the free energy of an energy-based model.
- Train it by temporal difference (TD) learning.

# Graphical Models and Approximate Inference

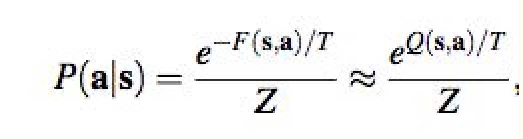
- Value function approximator is based on an undirected graphical model called a product of experts.
- Value of a state-action pair is modeled as the negative free energy of the state-action under the product model.

# RBM’s as Product of Experts Model

- Products of experts are probabilistic models that combine simpler models by multiplying their distributions together.
- RBM’s are undirected graphical models
  - Specifies joint distribution over input.
  - Computation of free energy is tractable!
  - intractable to compute the conditional distribution over actions
    - Use MCMC sampling (no theoretical convergence guarantees)
- Can be used to complete a visible vector that is only partially specified.

# Product of Experts as Function Approximator

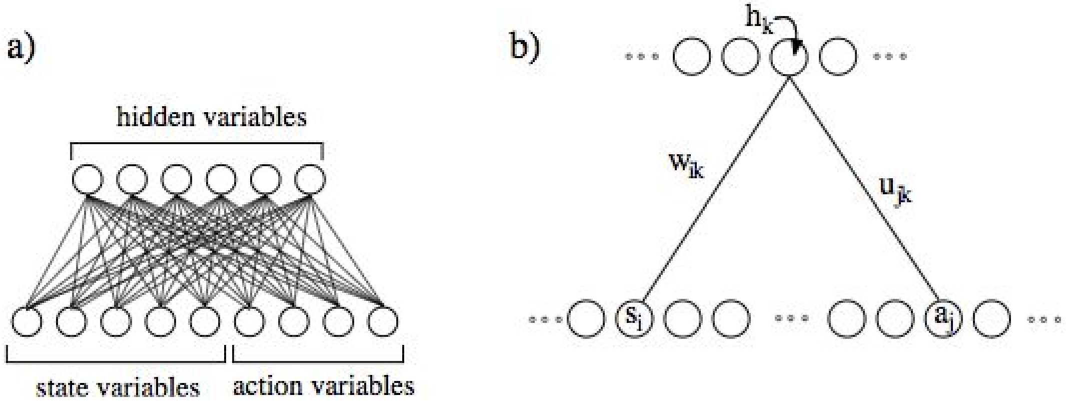
- Visible variables are state and action variables.
- Sample actions according to a Boltzmann exploration policy, conditioned on settings of the state variables.
- Create a correspondence between the value of a state-action pair, and its negative free energy under the Boltzmann machine model.
- Use MCMC sampling to select actions.
- Probability of sampling an action while holding the state fixed



$$P(\mathbf{a}|\mathbf{s}) = \frac{e^{-F(\mathbf{s},\mathbf{a})/T}}{Z} \approx \frac{e^{Q(\mathbf{s},\mathbf{a})/T}}{Z},$$

Good actions will become more probable under the model, and bad actions will become less probable under the model.

# Problem Formulation



$$F(\mathbf{s},\mathbf{a}) = -\sum_{k=1}^{K}\left(\sum_{i=1}^{N}(w_{ik}s_i\langle h_k\rangle)+\sum_{j=1}^{M}(u_{jk}a_j\langle h_k\rangle)\right) + \sum_{k=1}^{K}\langle h_k\rangle\log\langle h_k\rangle+(1-\langle h_k\rangle)\log(1-\langle h_k\rangle).$$

# Learning Model Parameters

- Model parameters are adjusted so that the negative free energy of a state-action pair under the product model approximates its action-value.
- Use SARSA Update!

$$\Delta w_{ik} \propto \left(r^t + \gamma \widehat{Q}(\mathbf{s}^{t+1}, \mathbf{a}^{t+1}) - \widehat{Q}(\mathbf{s}^t, \mathbf{a}^t)\right) s_i^t \langle h_k^t \rangle .$$

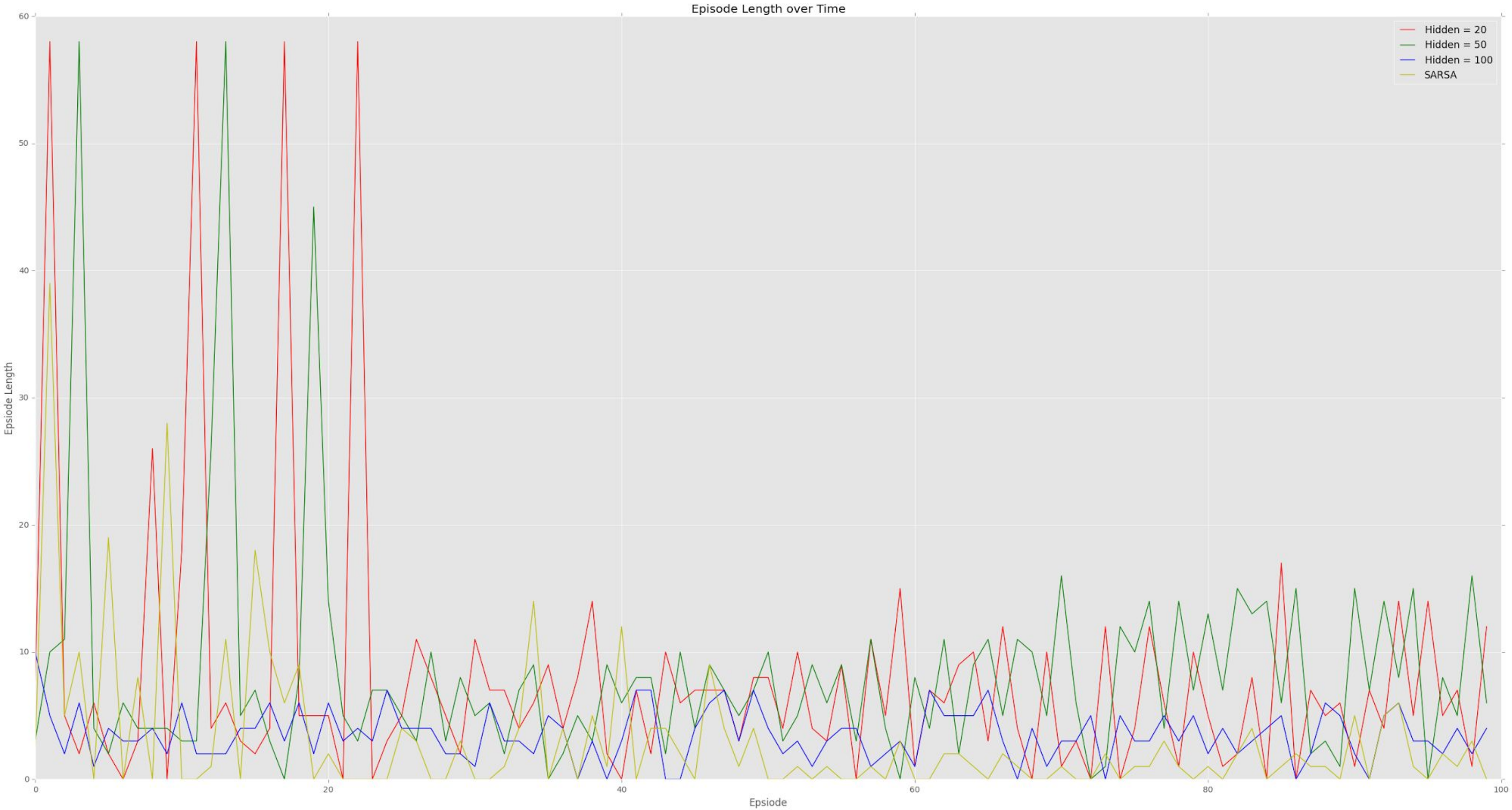
- **Exploration**
    - Use Boltzmann exploration!
    - Can move from exploration to exploitation by adjusting the “temperature” parameter T

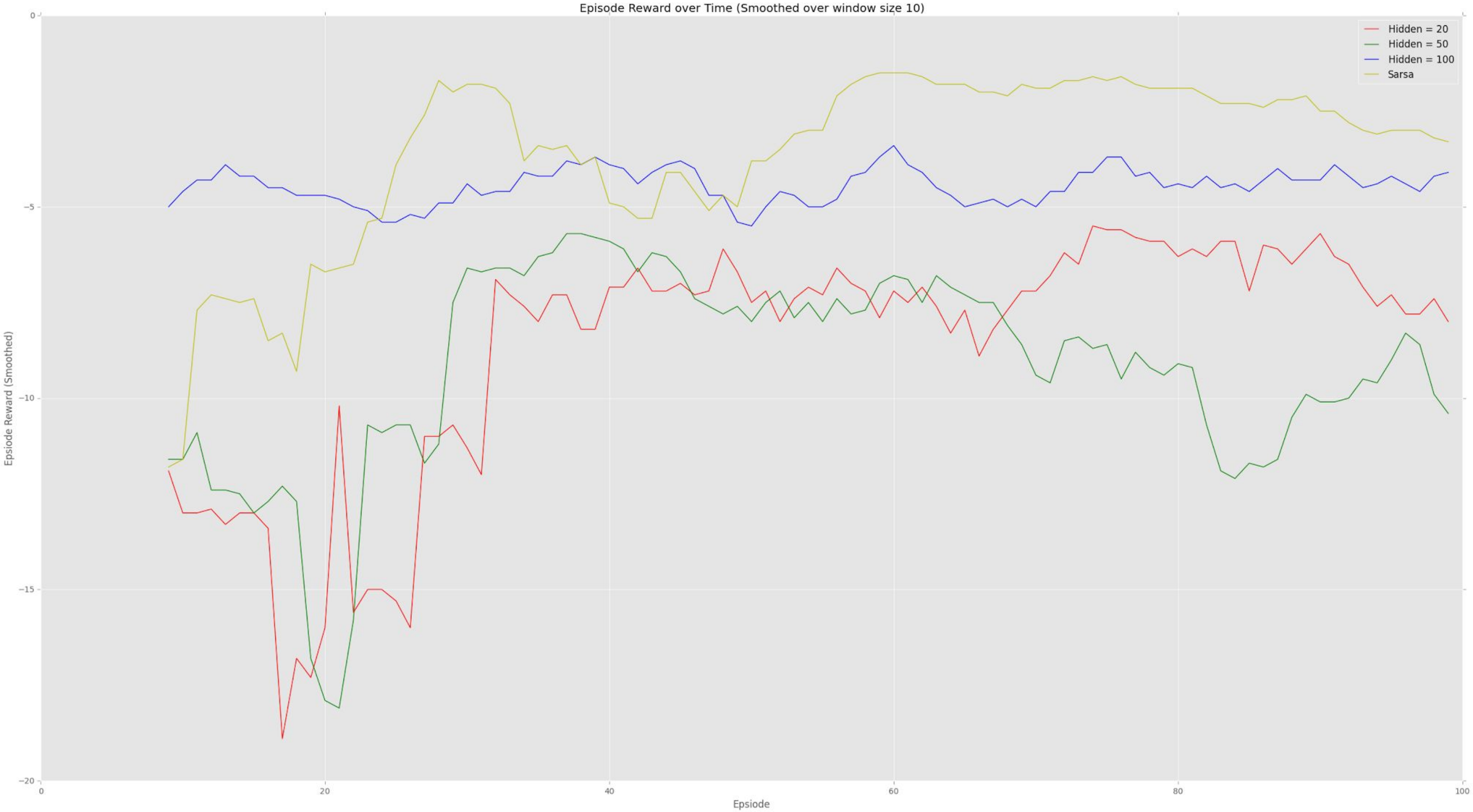
# Experiments

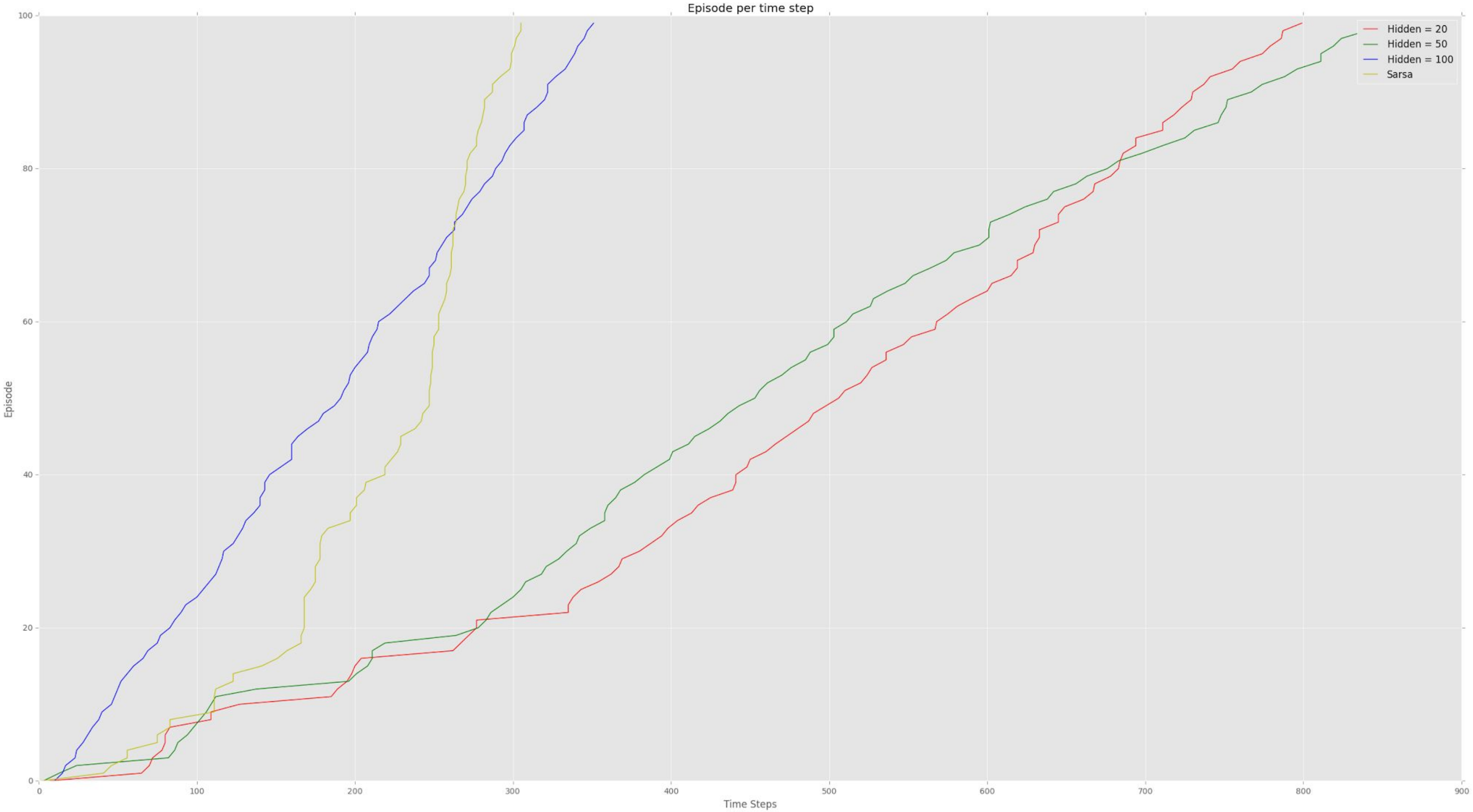
- Use TD learning with PoE model on GridWorld.
- 4 * 4 gridworld.
- Reward is -1 for each move, except it is zero when the agent reach its destination.
- 4 actions, 16 states.
- Compared to SARSA.
- SARSA performs better as compared to Energy based SARSA.
- It is possible that the experiment setup I have used is not ideal for Energy based SARSA.

# Co-operative multi-agent system

- Natural to represent both the state and action as sets of variables (one for each agent)
- Agent's action might be largely independent of the other agents' exact states and actions (at least for some regions of the combined state)
- Factored representation of the Q-value function might be appropriate:
    - The original representation of the combined states and combined actions is factored, and the ways in which the optimal actions of one agent are dependent on the states and actions of other agents might be well captured by a small number of "hidden" factors rather than the exponential number required to express arbitrary mappings


Episode Length over Time
Hidden = 20
Hidden = 50
Hidden = 100
SARSA
Epsiode Length
Epsiode
0
10
20
30
40
50
60
0
20
40
60
80
100


Episode Reward over Time (Smoothed over window size 10)
Hidden = 20
Hidden = 50
Hidden = 100
Sarsa
Epsiode Reward (Smoothed)
0
−5
−10
−15
−20
0
20
40
60
80
100
Epsiode


Episode per time step
Episode
Time Steps
0
20
40
60
80
100
0
100
200
300
400
500
600
700
800
900
Hidden = 20
Hidden = 50
Hidden = 100
Sarsa

# Some thoughts!

- Can use directed graphical model too.
- Approximate inference technique with a directed model to approximate the value, and also use the approximate inference technique to sample actions from the network.
- **Advantage! Distribution over actions could be evaluated, (not in PoE)**
- Has the flavour of Actor - Critic.
    - Free Energy - Plays the role of critic
    - Approximate inference -  Plays the role of actor.

# Future Work ?

- TD learning diverges when using non-linear value functions.
- Sensitive to the parameterization of the policy.
  - May be Natural gradient ?
- High variance in the gradient estimates.
  - Use Actor Critic!

# References

- Reinforcement Learning with Factored States and Actions http://www.jmlr.org/papers/volume5/sallans04a/sallans04a.pdf
- Actor-Critic Reinforcement Learning with Energy-Based Policies http://www.jmlr.org/proceedings/papers/v24/heess12a/heess12a.pdf
- Reinforcement Learning with Deep Energy-Based Policies https://arxiv.org/pdf/1702.08165.pdf