Solving with simple dqn

Solving with ddqn

solving with dqn + replay memory

universal approximation theorem

does there exist a simpler less computationally expensive algo?

Learning rate decay

Relu vs sigmoid

**Advantage:**

* Sigmoid: not blowing up activation
* Relu : not vanishing gradient
* Relu : More computationally efficient to compute than Sigmoid like functions since Relu just needs to pick max(0,xx) and not perform expensive exponential operations as in Sigmoids
* Relu : In practice, networks with Relu tend to show better convergence performance than sigmoid. ([Krizhevsky et al.](http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf))

**Disadvantage:**

* Sigmoid: tend to vanish gradient (cause there is a mechanism to reduce the gradient as "aa" increase, where "aa" is the input of a sigmoid function. Gradient of Sigmoid: S′(a)=S(a)(1−S(a))S′(a)=S(a)(1−S(a)). When "aa" grows to infinite large , S′(a)=S(a)(1−S(a))=1×(1−1)=0S′(a)=S(a)(1−S(a))=1×(1−1)=0).
* Relu : tend to blow up activation (there is no mechanism to constrain the output of the neuron, as "aa" itself is the output)
* Relu : Dying Relu problem - if too many activations get below zero then most of the units(neurons) in network with Relu will simply output zero, in other words, die and thereby prohibiting learning.(This can be handled, to some extent, by using Leaky-Relu instead.)

Which optimizer to use?

Graph: Reward at each training episode while training your agent and discussion of results

* Pair this with the steps graph, pair this with exploration
* Plot 3 different seeds

Explanation of pitfalls and problems you encountered.

* Tried without target network
* Tried with different losses
* Different optimizer methods
* Size of the network
* Different activation function

Graph: Eﬀect of hyperparameters and discussion of the results.

* High learning rate – pair with loss explosion
* Low learning rate
* Replay memory size
* Target network updates

Explanation of algorithms used: what worked best? what didn’t work?

* Without target network
* Without replay memory

Hovering problem

* Limiting solves hovering problem at a cost and makes it more likely to
* Training unbounded for longer times
  + Some seeds are lucky and converge fast
  + But converge to a better policy
* Stuck problem
* Learning gets worse after some time

Abstract

Describe rl

Describe the game

Describe q learning describe q learning

Describe dqn and neural networks from goodfellow

Describe the working experiment

GRAPH

Training reward steps per training

When to stop

Reward worsens on continual training talk about optimal policy and overfitting

A study on overfitting check briefly on generalization

GRAPH

Reward per episode for 100 consecutive episodes on trained agent

Pick the agent from the training and maximized reward and minimized steps

GRAPH

Effect of replay memory size

Effect of alpha

Effect of epsilon decay

Effect of gamma

Effect of network size

Large networks almost fail to learn

Small networks do not converge

Exporation problem, always had to keep exporation at 0.05 which had both advantages and disadvantages

Slow decay would cause loss explosion

Fast decay wouldn’t let agent learn optimal policy

Effect of mini batch size

Explain pitfalls

* Hovering problem – limit episodes
  + Found this to converge faster
  + Found a few non limited trainings that converged
  + But with number of trainings limited
    - Quantifying this with a concrete number would be wrong since we haven’t explored the seed space
    - Conclude based on the limited sample
* Unlearning problem
  + Premature exit
* Loss explosion
* Hyperparameter tuning
* Loss function
* Tried with different optimizer
* Stuck on edge problem

Explanation of algorithm used

* Replay memory size
* Without replay memory
  + Graphs and why
* Without target network
  + Graphs and why
* **On-policy if possible**

future possibilities

cnn into the picture

why

because it can see the hill

agent is only sensing through touch and not sight, we’re only aware of the ground when we touch it

would solve stuck on edge problem

on-policy lol if I didn’t try it already