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.)