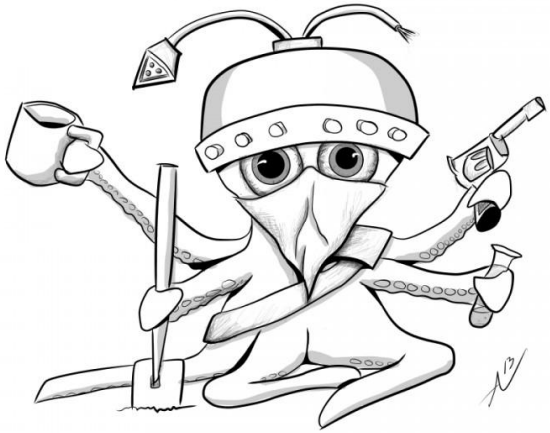
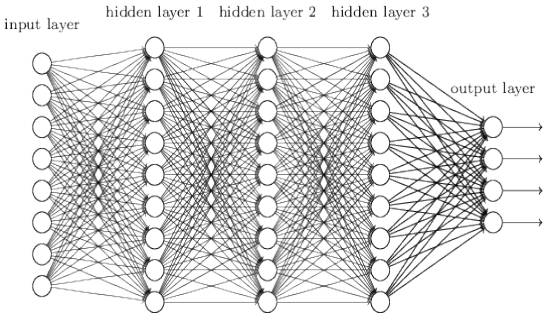
# Multi-Armed Bandit Solutions as Neural Net Learning Algorithms

By:
River Ludington,
Tristan Caputo,
Paul Lenharth



# What are Neural Networks?

- Program which is a much simplified version of a brain
- Contains nodes (“neurons”) connected to each other with arcs
- Neurons contain continuous functions with inputs from previous neurons
- The program “learns” by changing weights of arcs



# Neural Network Learning

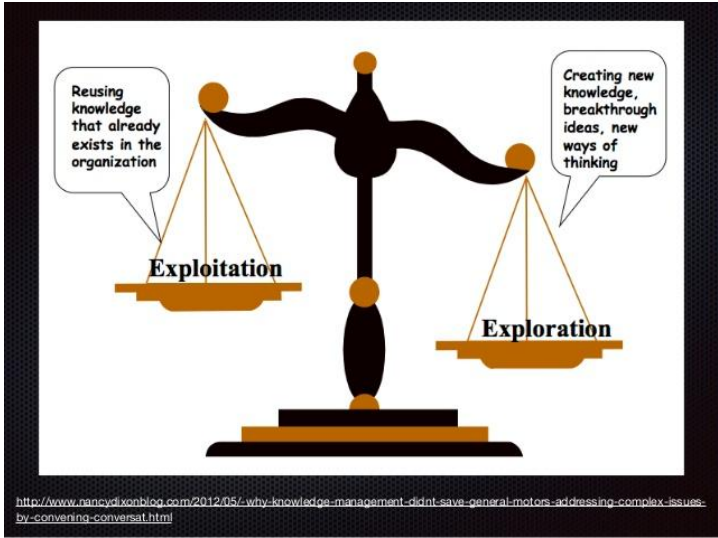
Two main ways to learn: **Unmonitored** and **Monitored**

**Unmonitored** Nets have nodal systems that learn without use of an outside program.

**Monitored** Nets have a seperate program that makes small changes to get closer to a provided result

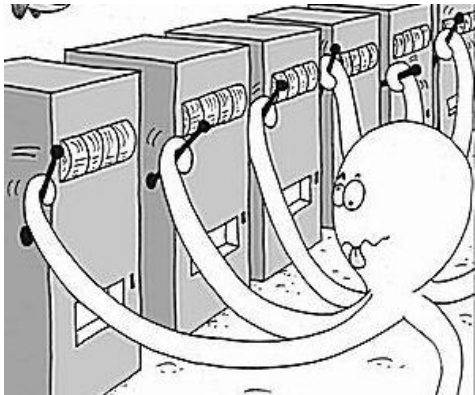
# Neural Network Learning

- Functions are modified starting at the last layers and working back: backpropagation
- Requires an algorithm to determine whether to refine the current functions or make larger changes: exploitation vs. exploration



# Multi-armed Bandit Problem

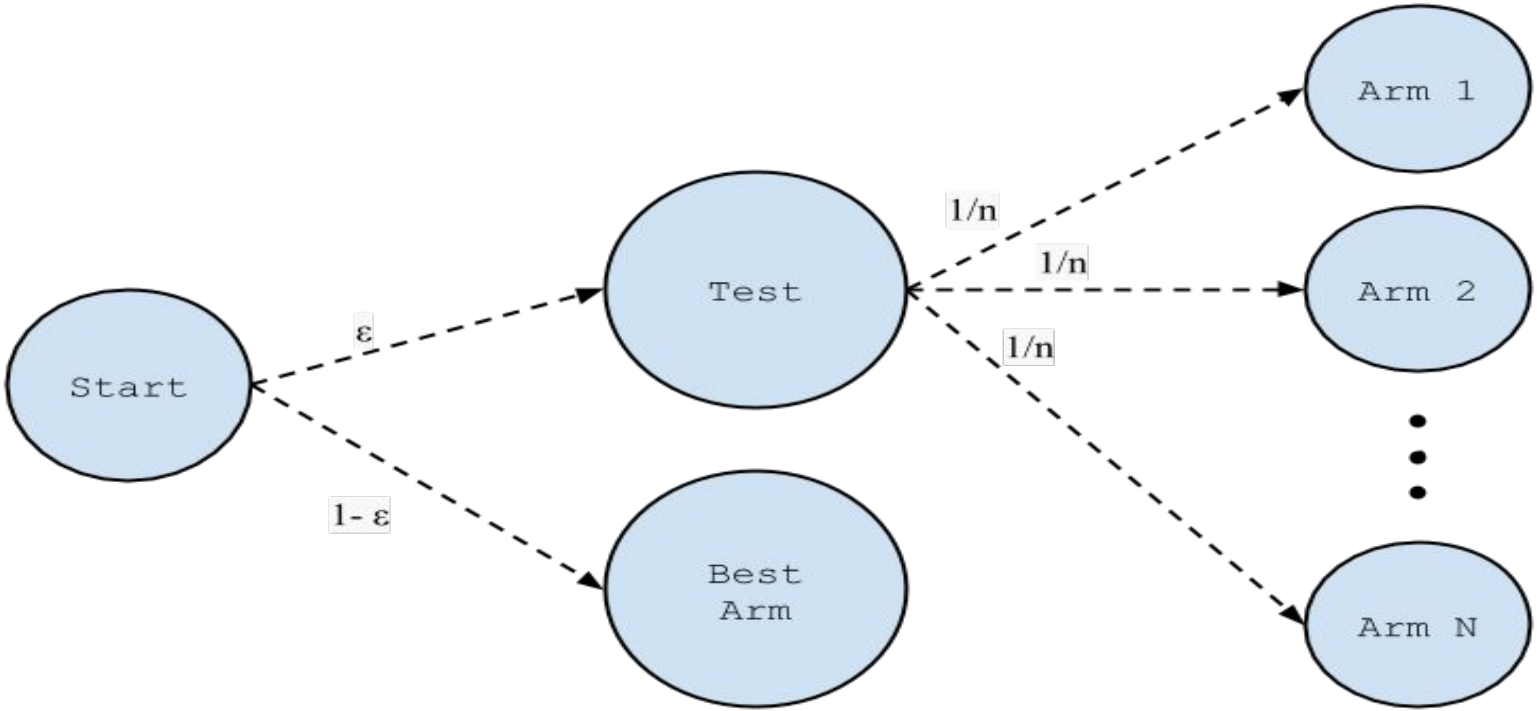
- Multi-armed bandit is a model used for the exploitation/exploration problem
- Formulated with slot machines of different, unknown payouts
- Goal is to maximize money
- Requires quickly finding the “best” machine and playing only it
- Probability makes knowledge of best machine uncertain, must keep testing others
- Testing nets less money



# Most Popular Bandit Solutions

1. Epsilon-Greedy (many variations)
2. Softmax (Also known as Boltzmann)

# Epsilon Greedy



# Softmax

- Fixes Problem from Epsilon-Greedy algorithm
- Problem: All non-best arms are treated equally
- Solution: Select non-best arms with Softmax distribution

# Softmax

Function of experimental average payout of that arm, $m_a$

$$P(a) = \frac{e^{m_a}}{\sum_a e^{m_a}}$$

Probability of selecting random arm

Normalization of arms

# Measure of Success

1. Make the most money possible
2. Reduce “Regret”

The most popular performance measure for bandit algorithms is the *total expected regret*, defined for any fixed turn $T$ as:

$$R_T = T\mu^* - \sum_{t=1}^{T} \mu_{j(t)}$$

where $\mu^* = \max_{i=1,\dots k} \mu_i$ is the expected reward from the best arm.

Alternatively, we can express the total expected regret as

$$R_T = T\mu^* - \mu_{j(t)} \sum_{k=1}^{K} \mathbb{E}(T_k(T))$$

where $T_k(T)$ is a random variable denoting the number of plays of arm $k$ during the first $T$ turns.

Citation: Kuleshov, Volodymyr, and Doina Precup. "Algorithms for multi-armed bandit problems." *arXiv preprint arXiv:1402.6028* (2014).

# Room for Growth

## According to our Research:

> “simple heuristics such as ε-greedy and Boltzmann exploration outperform theoretically sound algorithms on most settings by a significant margin.” - Volodymyr Kuleshov, Doina Precup

Note: The higher end algorithms are pursuit, reinforcement comparison, UCB1, UCB1-Tuned

# Implementation Plan

- Devise a code simulating a player trying his luck on K slot machines with respective probability distributions and see which one he picks after each turn.
- Algorithms will tell the player which slot machine (“arm”) to select after each turn, after a certain number of tries, we will measure the total regret (i.e. $$$ lost compared to only playing the best machine).
- Our main parameters will be K (number of arms) and the reward variance
- After obtaining working code, various algorithms will be tested

# Applications: Clinical Trials, Recommendations Alg.

- Identify the best treatment (best slot machine) and ensure that as many patients as possible can receive it in the clinical trial. (less regret)

- Identify the best recommendations (best slot machine) and ensure that the user has a good chance of clicking on it. (less regret)

NETFLIX

# Work Allocation

Tristan: Coding algorithms

River: Algorithm development

Paul: Performance analysis of algorithms on Bandit Problem

# References


1. Vermorel, Joannes, and Mehryar Mohri. "Multi-armed bandit algorithms and empirical evaluation." *European conference on machine learning*. Springer, Berlin, Heidelberg, 2005.
2. Kuleshov, Volodymyr, and Doina Precup. "Algorithms for multi-armed bandit problems." *arXiv preprint arXiv:1402.6028* (2014).
3. Raja, Sudeep. “Multi Armed Bandits and Exploration Strategies.” *Multi Armed Bandits and Exploration Strategies – Sudeep Raja – MS/Phd Student at UMass Amherst*, 28 Aug. 2016, sudeepraja.github.io/Bandits/.
4. “Multi-Armed Bandits.” *The Data Incubator MultiArmed Bandits Comments*, blog.thedataincubator.com/2016/07/multi-armed-bandits-2/.