Big data and machine learning for network research problems

Sabidur Rahman

Friday Group Meeting

December 22, 2016

krahman@ucdavis.edu

http://www.linkedin.com/in/kmsabidurrahman/

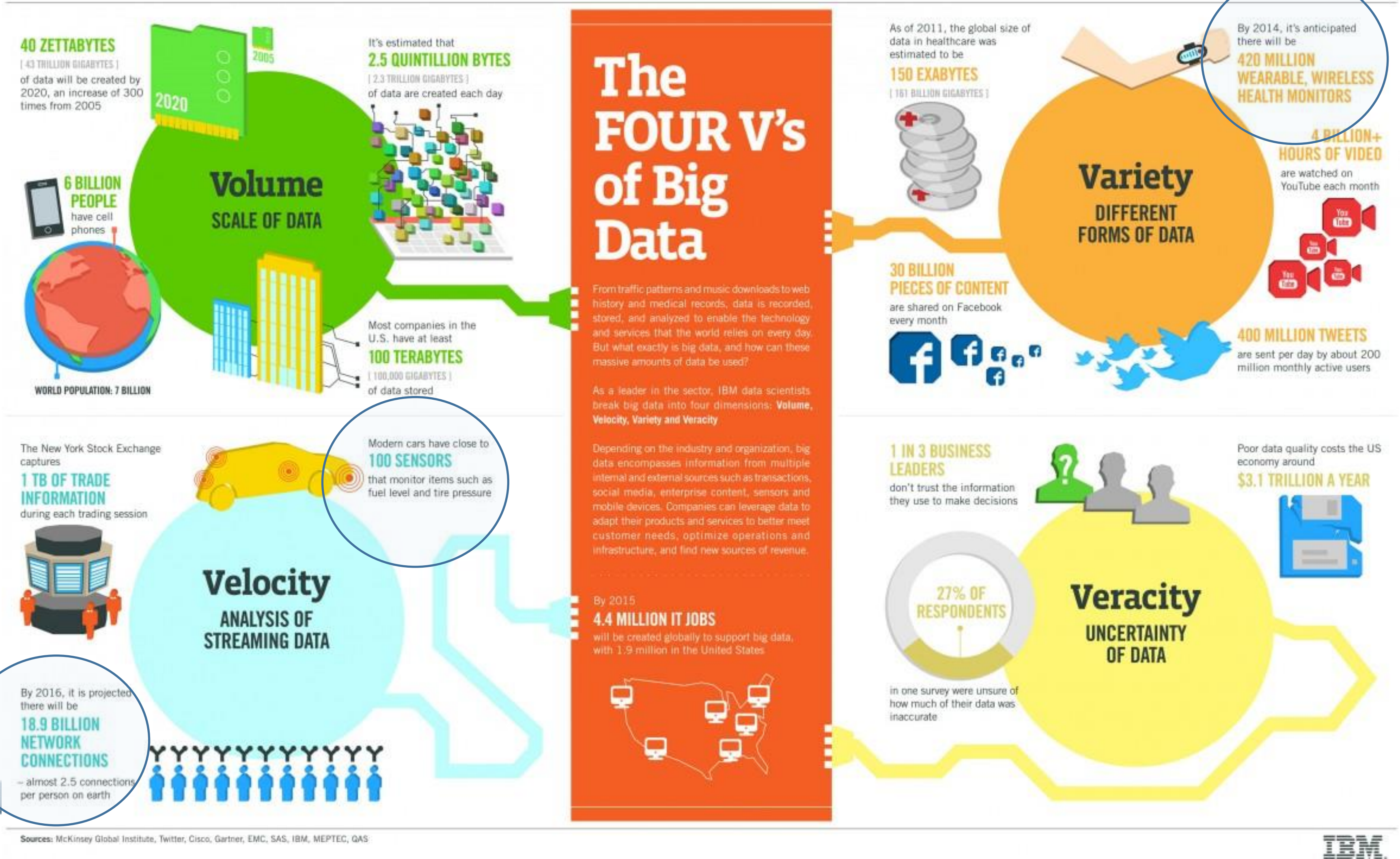
UCDAVIS
UNIVERSITY OF CALIFORNIA

Agenda

- Big data and machine learning
  - How these two fit into networking world?
- Random Neural Network
- Reinforcement learning
- Paper review: O. Brun, L. Wang, and E. Gelenbe, "**Big data for Autonomic Intercontinental Overlays**", IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, March 2016.

UCDAVIS

Big data



http://www.ibmbigdatahub.com/infographic/four-vs-big-data

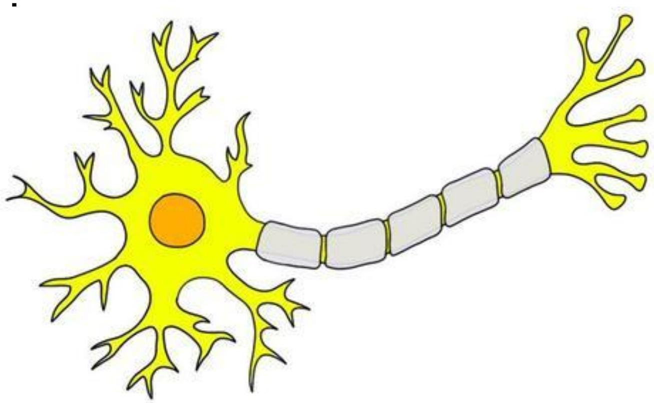
Machine learning

- Network today is producing large amount of data.
- What to do with the collected data?
  - Network performance enhancement
  - Smarter routing decision
  - Network management and planning
  - Network automation
- Analytics and automation of this huge amount of data depends largely on data analytics and machine learning techniques.

UCDAVIS

Random Neural Network (RNN)

- When a biophysical neuron is excited, it transmits a train of signals, called **spikes (firing of signals)**, along its axon to either **excite or inhibit** the receiving neurons.
- The combined effect of excitatory (positive) and inhibitory (negative) inputs change the potential level of the receiving neuron and determine whether it will become excited.
- Neurons in RNN interact by exchanging positive and negative spiking signals with certain probability.
- RNN model is used in pattern recognition, classification, image processing, combinatorial optimization and communica



Excitatory = causing, or constituting excitation
Inhibitory = suppress or restrain an impulse

Timotheou S. The random neural network: a survey. The computer journal. 2010 Mar 1;53(3):251-67.

Reinforcement learning

The basic reinforcement learning model consists of:

- Set of environment and agent states S
- Set of agent actions A

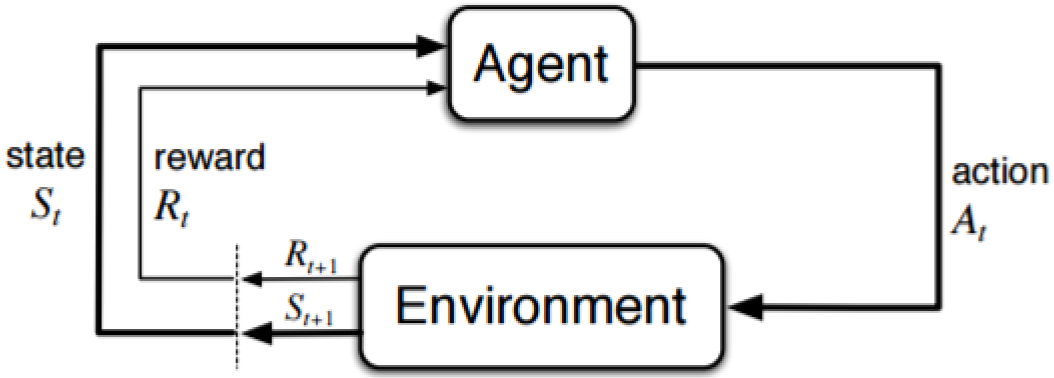


Figure 3.1: The agent–environment interaction in reinforcement learning.

- Policies of transitioning from states to actions
- Rules that determine r*eward (or punishment)* of a transition (G and R)

Sutton RS, Barto AG. Reinforcement learning: An introduction. Cambridge: MIT press; 1998 Mar 1.

UCDAVIS

Paper review

O. Brun, L. Wang, and E. Gelenbe, “**Big data for Autonomic Intercontinental Overlays**”, IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, March 2016.

Introduction

- Multihop intercontinental network that uses IP to communicate between nodes.
- IP routing often results in sub-optimal paths with respect to metrics such as end-to-end round trip delay.
- QoS of such routes can be optimized using the collected network statistics.
- **Objective**: Select route to provide better QoS than IP.
- **Machine learning** based scheme to exploit large scale data collected from communicating node pairs.

(Big) Data collection

- ICMP-based ping to collects round-trip delay data at the overlay nodes, every two minutes
- 24 hours with 20 overlay nodes: collect up to some **2.7 × 10^5 data points per day.**

Fig. 5. Geographical location of the 20 Source and Destination Nodes used in our experiments within the NLNog ring. These same nodes are also used for our direct IP based routing measurements, and also to evaluate the SMART Overlay Node based routing scheme.

UCDAVIS

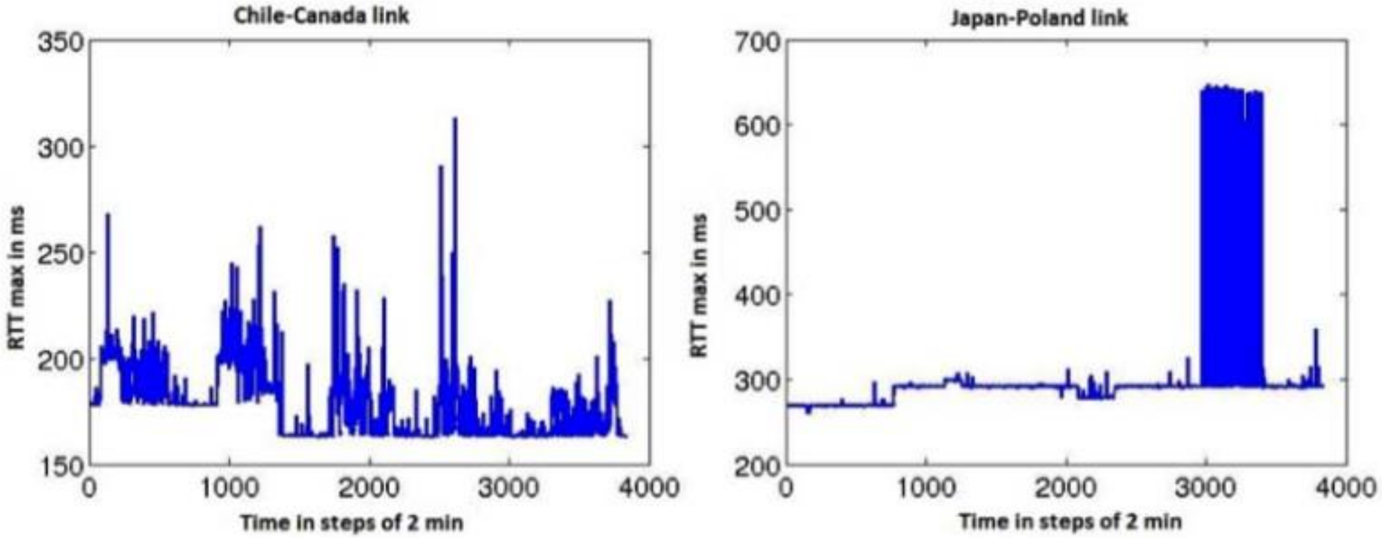


Fig. 6. The RTD in milliseconds using the IP protocol, given in averages over successive 2 minute intervals over an observation period; each unit along the x-axis corresponds to successive 2 minute intervals over the long time period being shown. The connections measured are between Chile-Canada (left) and Japan-Poland (right), and we observe the large variation in average RTD depending on the time period being considered.

UCDAVIS

Routing overlay

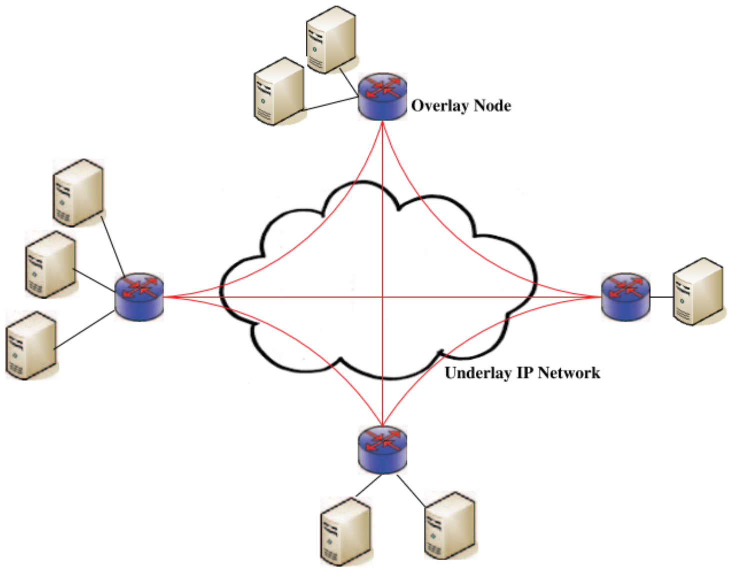


Fig. 1. Schematic description of the structure of a routing overlay, where the Overlay Nodes exchange packets with each other with packets that tunnel through the IP connections, while paths between Overlay Nodes may transit through intermediate Overlay Nodes.

Routing overlay

- Routing overlay is a method to improve performance without changing underlying network.
- Overlay network is formed by software routers deployed over the Internet.
- Operates by monitoring the quality of Internet paths (**latency, bandwidth, loss rate**) between overlay nodes.
- When primary path becomes unavailable or suffers from congestion, **re-routes packets** along an alternate path.
- Routing overlays can be used to **quickly recover** from path outages, and also **improve QoS** of data flows.
- This approach makes it possible to override the routes determined by Internet protocols and to route traffic based on metrics directly related to the performance needs of the application.

Self-healing and self-optimising routing overlay

- SMART overlay network is formed by software agents that are deployed at VMs.
- On each VM, a **Transmission (TA) and Reception Agent(RA)** run together with various Applications or tasks.
- Each VM's software router is the **Proxy** that monitors the quality of the overlay paths towards other destinations, selects the best paths, and forwards the packets over these paths.
- **TA** receives the packets that are being sent to other Applications at other sites.
- **RA** receives packets from the local Proxy, decapsulate and delivers them to the appropriate Application in the VM.
- TA, RA and Proxy enables the control of path of packets through the network, without the applications being aware that their data flows are routed by the overlay.

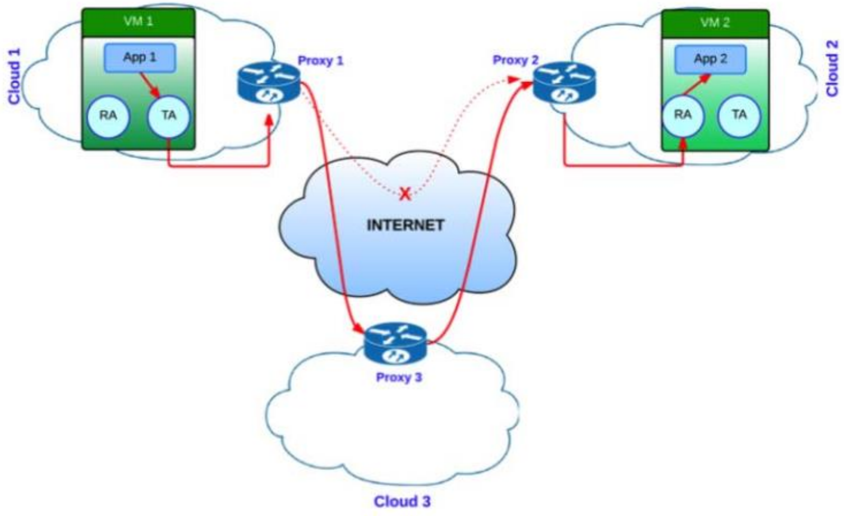
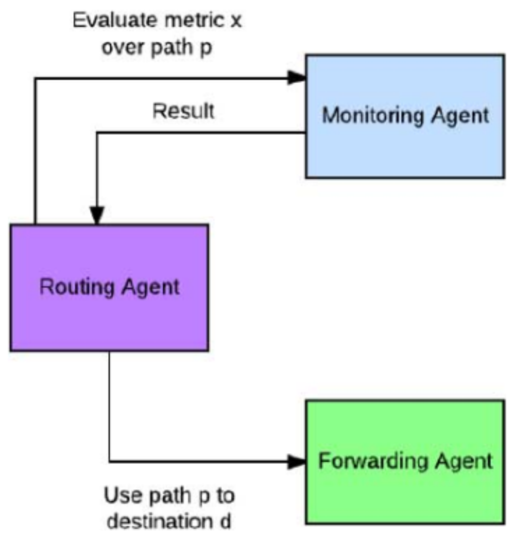


Fig. 2. Architecture of the Autonomic Communication Overlay: the Overlay Nodes used by SMART exchange packet streams via the Internet using the Internet Protocol (IP) either directly, or via intermediate Overlay Nodes. When a SMART path uses multiple Overlay Nodes, IP is used between adjacent Overlay Nodes.

UCDAVIS

Proxy: details

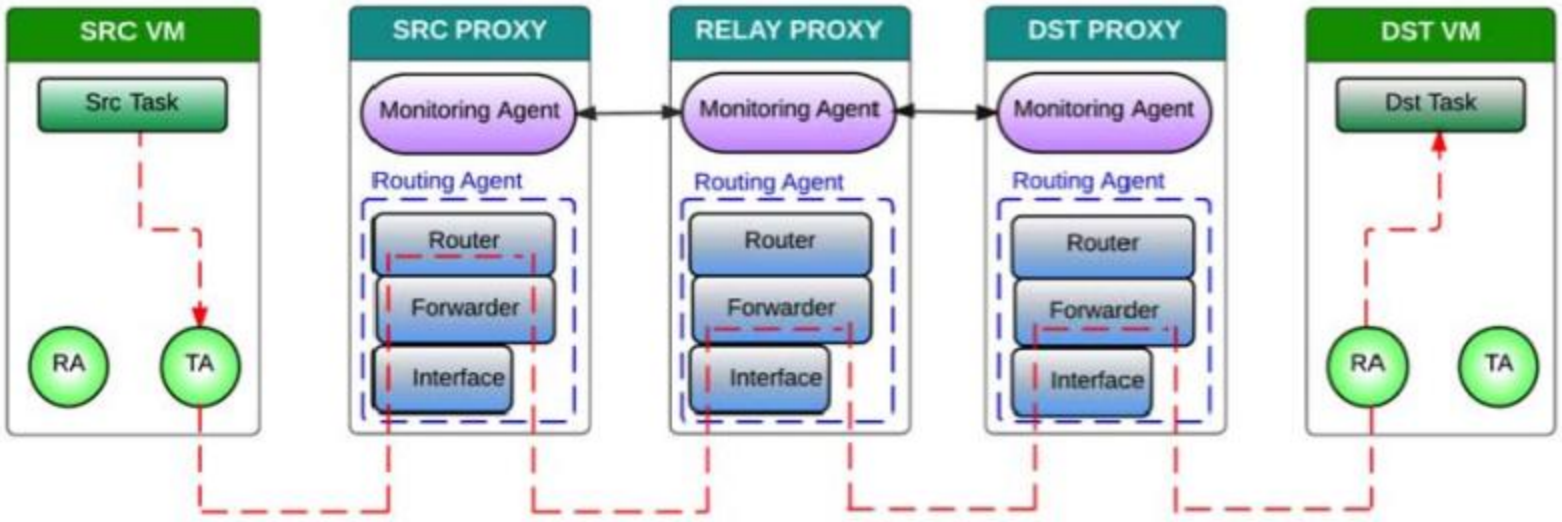
- **Monitoring Agent**: monitors the quality of the Internet paths between the local cloud and the other clouds.
- **Routing Agent**: drives the monitoring agent and uses the data it collects to discover an optimal path.
- **Forwarding Agent**: forwards each incoming packet to its destination on the path it was instructed to use by Routing agent.
- **Source routing:** Routing table of source proxy describes the complete path between overlay proxies to be followed by a packet to reach its destination, while the path between proxies is determined by the conventional IP protocol.



Interactions between the entities constituting the proxy.

Packet forwarding process

- SMART header contains the sequence of intermediate proxies.



Details of the SMART packet forwarding process.

Random Neural Network (RNN)

- At each time interval, the algorithm chooses a subset of paths to probe, and measures the sum of edge delays in the probed paths.
- The algorithm then sends its packet over the minimum latency path among those it has probed.
- Each neuron in RNN represents a path from source to destination.
- RNN helps to select K paths to destination out of N paths.
- State of n connected neurons are represented by a vector $\mathbf{k}(t) = [\mathbf{k}_1(t), \mathbf{k}_2(t), \ldots, \mathbf{k}_n(t)]$, where $\mathbf{k}_i(t)$ is a non-negative integer valued *random variable* representing the “potential” of the *i*-th neuron being excited at time *t*.

Random Neural Network

- A neuron *i* of the RNN is said to be excited whenever **k***i(t)* > 0, in which case it can fire and send signals at an average rate *ri*

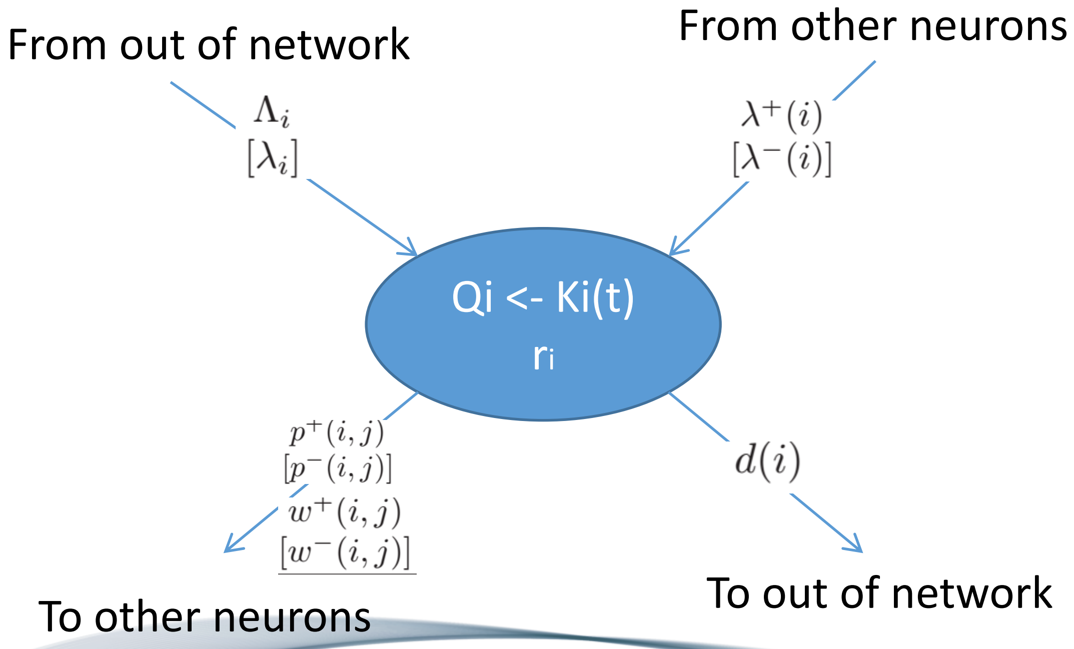


TABLE 1. List of RNN symbols

Notation	Definition
$k_i(t)$	Potential of neuron i at time t
q_i	Probability neuron i is excited at time t
Λ_i $[\lambda_i]$	External arrival rate of positive [negative] signals to neuron i
$\lambda^+(i)$ $[\lambda^-(i)]$	Average arrival rate of positive [negative] signals to neuron i
$p^+(i,j)$ $[p^-(i,j)]$	Probability neuron j receives a positive [negative] signal from firing neuron i
$w^+(i,j)$ $[w^-(i,j)]$	Rate of positive [negative] signals to neuron j from firing neuron i
$d(i)$	Probability a signal from firing neuron i departs from the network
r_i	Firing rate of neuron i
N	Number of neurons in the network

Positive signal received = higher probability of firing
Neg. signal received = higher probability of not firing

Timotheou S. The random neural network: a survey. The computer journal. 2010 Mar 1;53(3):251-67.

Reinforcement learning

- "Guide" RNN to choose better paths by:
  - "Reward" neurons representing good paths.
  - "Punish" neurons representing bad paths.

1. Increasing probability of it's "firing"
2. Reducing probability of others "firing"

1. Decrease probability of it's "firing"
2. Increasing probability of others "firing"

G = minimum delay

$R = G^{-1}$

Higher delay = lower reward

UCDAVIS

Algorithm

Algorithm 1. Learning optimal paths with a RNN and Reinforcement Learning.

1: **for** $tl = 1, 2, \ldots$ **do**
2: $\quad \mathcal{P}(t_l)$ is the set of K neurons with highest probabilities q_i at time t_l.
3: $\quad R_j(t_l) \leftarrow$ reward obtained with path j.
4: $\quad p^* \leftarrow \arg\max_{j \in \mathcal{P}(t_l)} R_j(t_l)$.
5: $\quad$ **for** $j \in \mathcal{P}(t_l)$ **do**
6: $\quad\quad \nu_j \leftarrow R_j(t_l)/T(t_l)$.
7: $\quad\quad$ **if** $R_j(t_l) \geq T(t_l)$ **then**
8: $\quad\quad\quad$ **for** $i = 1, \ldots, N$ **do**
9: $\quad\quad\quad\quad \Delta_i \leftarrow (\nu_j - 1)\, w^+_{i,j}$.
10: $\quad\quad\quad\quad w^+_{i,j} \leftarrow w^+_{i,j} + \Delta_i$.
11: $\quad\quad\quad\quad w^-_{i,k} \leftarrow w^-_{i,k} + \frac{\Delta_i}{N-2}, \forall k \neq j$.
12: $\quad\quad\quad$ **end for**

2. Select neurons with highest probability at time tl

3. Calculate rewards. $R = G^{-1}$ where G = minimize delay

4. p* is the best path

7. Good path: reward the neuron!

10.Increase positive weights going into the neuron

11. Increase negative weights leading to other neurons

Algorithm(continued..)

13: **else**
14: **for** $i = 1, \ldots, N$ **do**
15: $\Delta_i \leftarrow \left(1 - \nu_j\right) w^-_{i,j}$.
16: $w^-_{i,j} \leftarrow w^-_{i,j} + \Delta_i$.
17: $w^+_{i,k} \leftarrow w^+_{i,k} + \frac{\Delta_i}{N-2}, \forall k \neq j$.
18: **end for**
19: **end if**
20: **for** $i = 1, \ldots, N$ **do**
21: $r^*_i = \sum_{k=1}^{n} w^+_{i,k} + w^-_{i,k}$.
22: $w^+_{i,j} \leftarrow w^+_{i,j} \frac{r_i}{r^*_i}$.
23: $w^-_{i,j} \leftarrow w^-_{i,j} \frac{r_i}{r^*_i}$.
24: **end for**
25: Solve the non-linear system (2)
26: $T(t_{l+1}) \leftarrow \beta T(t_l) + (1 - \beta) R_j(t_l)$.
27: **end for**
28: **end for**

13. Bad path: punish the neuron!

16. Increase negative weights going into the neuron

17. Increase positive weights leading to other neurons (give other paths a higher chance of being selected)

22. Re-normalize w to avoid ever increasing weights

26. Update decision threshold T.

$\beta \in (0, 1)$ is used to introduce forgetfulness.
Higher β will give more importance lesson learned from previous events.

UCDAVIS

Results(1)

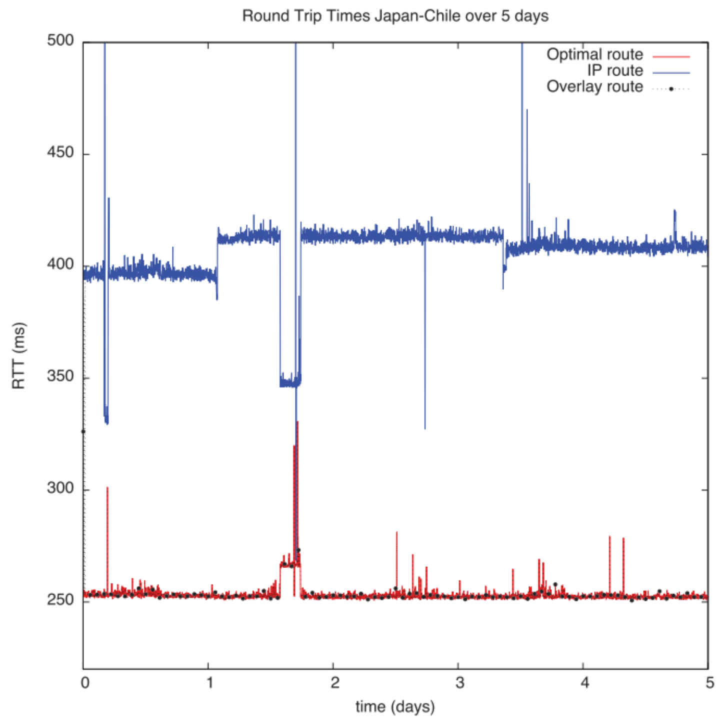


Fig. 9. RTD in millliseconds measured for the Japan-Chile connection in an experiment lasting 5 successive days. We see that measured RTD for the the SMART routing policy (in black dots) follows the values of the optimal (i.e. minimum) RTD very closely.

Results(2)

TABLE I

NON-OPTIMAL RTDS FOR ALL THE MEASUREMENTS: IP VS. SMART ROUTING WITH 2-HOP OVERLAYS. 16% OF 2-HOP OVERLAYS AND 50% OF IP PATHS ARE NON-OPTIMAL. RTDS FOR IP CAN SUBSTANTIALLY EXCEED THE MINIMA AND AVERAGES OVER ALL MEASUREMENTS

	Direct	2-hop Overlays
% non-optimal instances	50.08%	16.20%
Av'ge % diff. above min. latency	11.1%	4.24%

TABLE II

RELATIVE GAP IN PERCENTAGE OF RTD, TO THE MINIMUM OBSERVED RTD, FOR SOME PATHOLOGICAL ORIGIN TO DESTINATION PAIRS

	Direct IP	K=2
Singapore-Israel	26.86	0.34
Japan-Chile	60.73	0.08
Australia-Chile	26.03	0.30
Norway-Singapore	23.35	1.15
Poland-Brazil	24.32	0.39
Ireland-Moscow	119.39	0.18
Israel-Moscow	48.39	0.17

UCDAVIS

Results(3)

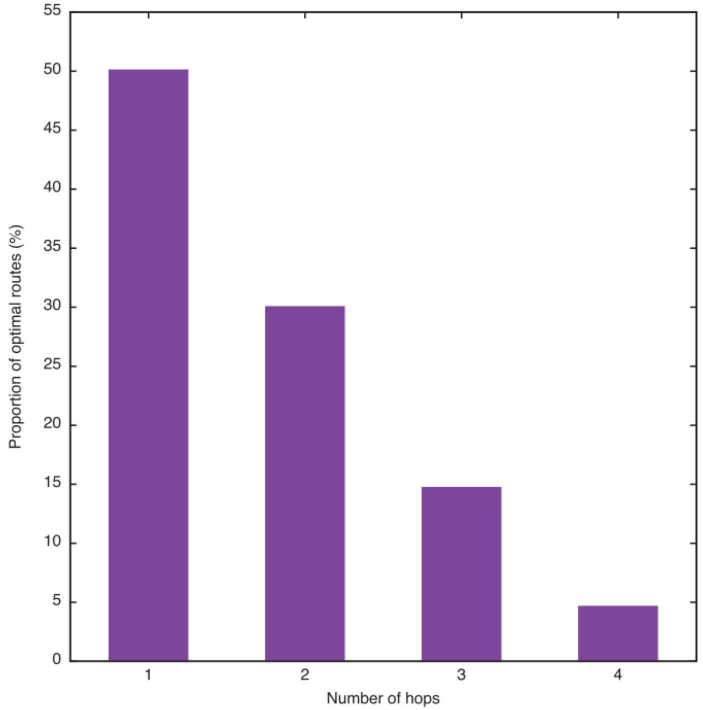


Fig. 7. Percentage of instances when the overlay path that minimises RTD, which uses IP paths between Overlay Nodes, is observed to include 1, 2, 3 or 4 Overlay Hops. We see that at most two overlay hops cover most of the optimal cases.

UCDAVIS

Summary

- Increasing amount of data collected from networks can help to improve performance.
- Machine learning tools are useful to analyze and learn from these huge data.
- Usefulness of other interesting machine/deep learning tools are yet to be explored in Network research.

UCDAVIS

Thanks!

Comments: krahman@ucdavis.edu

http://www.linkedin.com/in/kmsabidurrahman/

UCDAVIS