Premise

- 2017 – ML still hasn't been properly explored for networking
- Goal: Preliminary work exploring ML for routing
- Lots of past work on flow optimisation – for given topology and traffic demands, optimal routing configurations can be computed
  - Fail miserably in dynamic situations
    - Given past traffic conditions, optimise with respect to them and hope it works well in the future
    - Find a good (static) routing configuration for a range of traffic scenarios
High Level Questions

- Learn next input ("demand") then calculate routing, or learn routing configuration directly?
- What should a learning algorithm produce?
- Will supervised learning work?
- Can we use reinforcement learning?
Model

- Goal: repeatedly select routing configuration minimizing congestion
- Model network as a directed graph where edge weights are link capacities
- **Routing Strategy**: for all source $s$ and destination $d$ pairs, at any vertex $v$, how traffic going from $s$ to $t$ through $v$ is split across neighbors of $v$
  - $|V|^2 \times |E|$ variables overall
  - Require loop-free routing
- **Demand Matrix**: specifies traffic demand between all (source, destination) pairs
Strategies

- Given a demand matrix $D$, we can calculate an optimal routing strategy via linear programming

- Supervised Learning
  - Predict next demand matrix, given past $D$'s
  - Calculate routing strategy from result

- Reinforcement Learning
  - Learn routing strategy directly from sequence of last $k$ $D$'s
Supervised Learning

- Assume regularity in network demand (daily, weekly cycles etc)
- Input: last $k$ demand matrices, predict next $D$
- Various neural nets: fully connected; CNN; NAR-NN (nonlinear auto-regressive)
- Generate “actual” demand matrix sequences
  1) Deterministically generated from prior $D$'s (cyclic of length $q$)
  2) Independently generated from fixed prob dist.
- Result: only NAR-NN succeeds for only cyclic scenario if $q < k$
Reinforcement Learning

- $|V|^2|E|$ is too large
- Destination-based routing: each vertex splits traffic across neighbors based only on destination: $|V|^*|E|$
- TRPO [4] learning on 3 layer, fully connected NN
- Reward = $(\text{max-link-utilization} / \text{optimal-max-link-utilization})$ given the next real $D$
- Learn mapping from $k$ past $D$'s to per-vertex traffic splitting ratios ($\text{softmax}(\text{real output}) \rightarrow \text{routing probabilities}$)
- Result: 700 epochs, still produces high $\text{max-link utilization}$
- Authors propose number of output parameters is too large
Reinforcement Learning – Softmin Routing

- Only learn 1 parameter per edge: $|E|$ parameters
- Given parameters $p$, and vertex $v$, we can calculate edge weights via shortest-path intermediate step, then apply *softmin* to calculate splitting probabilities
- Reward same as before
- Compare to three baselines: softmin routing based directly on prior $D$, based on average of last $D$'s, and oblivious [3] (routing strategy independent of past $D$'s)

$$\text{softmin}_\gamma(\alpha)_i = \frac{e^{-\gamma \alpha_i}}{\sum_{j=1}^r (e^{-\gamma \alpha_j})}$$
$\text{softmin}_\gamma(\alpha)_i = \frac{e^{-\gamma \alpha_i}}{\sum_{j=1}^{r} (e^{-\gamma \alpha_j})}$
It works!

Figure 2: Representative Results for softmin-Routing
Discussion

• Claims to be a very early paper in ML applied to routing
  – Very broad, shallow analysis
  – Not much evaluation presented except for select cases
  – Try to cover a huge configuration space: different ways to generating individual demand matrices, sequences of matrices, neural net architectures, supervised/reinforcement...

• There have been starts at using learning in networking in the 90s [2]

• Only recently a few papers published with modern ML techniques
Discussion

- Could spawn lots of future work
  - Different types of networks (including less simplified models)
  - More training time, different architectures (RNN?)
  - Different supervised learning approaches

- Big Idea
  - ML has lots of potential for generating efficient, dynamic routing strategies
References

1) A. Valadarsky et al.: A Machine Learning Approach to Routing, arXiv, 2017.

2) Boyan, Justin A., and Michael L. Littman. "Packet routing in dynamically changing networks: A reinforcement learning approach." Advances in neural information processing systems. 1994.

3) Azar, Yossi, et al. "Optimal oblivious routing in polynomial time." Proceedings of the thirty-fifth annual ACM symposium on Theory of computing. ACM, 2003.

4) Schulman, John, et al. "Trust region policy optimization." Proceedings of the 32nd International Conference on Machine Learning (ICML-15). 2015.