AI for Networking: an Engineering Perspective

Mariam Kiran
Research Scientist
Energy Sciences Network
Lawrence Berkeley National Laboratory

INDIS Workshop @SC20
12th Nov 2020

Funding Support: DOE ASCR, LBNL
Outline

• My Background:
  – Working with Science R&E Networks
  – Focused on deep learning solutions for a ‘self-learning’ network
• Motivation: AI applications for Network Operations
• Quick ML/Al overview
  – Supervised, unsupervised and reinforcement learning
• Mathematical Challenges of AI
• Software: NetPredict (congestion patterns), DeepRoute (traffic engineering)
• Summary
Why use AI?
Why use AI?

- Networks are growing and becoming more complex
- Multiple vendors and their hardware
- Growing demand for reliable connectivity (science and home networks)
- Help engineers focus on innovation rather than mundane everyday tasks
AI Proposals for Networks

Improving Traffic Engineering

Predicting Network Surges

- Security Anomalies
- Elephant versus mice flows
- And more ...

Improving the TCP Congestion Model
Science WANs Challenge 1: Resources are often underutilized

- Traffic is highly variable and ‘bursty’ (big versus small transfers)
- Flow congestion leads to packet loss
- Networks are designed to be resilient
- Under-utilized resources

Designed for Resilience: Capped at 40% Utilization

ESnet Year 2019 usage on 100GB links
Science WANs Challenge 2: TCP Performance can be Fragile

- Science uses Transmission Control Protocol (TCP) to send data
- 95% of science traffic uses TCP
Science WANs Challenge 3: TCP Congestion Control for Performance

- Each TCP uses complex algorithms to optimize performance (slow down or quicken data sending)*
- Examples include video buffering on Youtube, VOIP (zoom) or big file transfers in few hours (Astronomy/physics from LHC)

*Cubic BBRv1

*Building Classifiers for TCP congestion algorithms, Machine Learning Journal
The Fusion Challenge: Network + AI/ML

- Deep Learning trend
  - Image recognition e.g. Cats 2011
  - Self-playing game e.g. AlphaGo 2016

- Hardware acceleration
  - GPU advances
  - FPGAs

- Industry and Academic Efforts
  - Smart NICs e.g. Barefoot
  - AI @ Control Plane e.g. juniper, cisco)
  - AI enabled TCP e.g. [1, 2]
  - Traffic patterns e.g. [3]

Quick Background of AI
This talk focuses on Deep Learning (DL) compared to general Machine Learning (ML)

Trained to recognize cats, “object identification”

Trained to play games, “Best strategies for winning game”

Deep learning (neural networks - NN) introduces ‘data-driven learning’ to build bespoke solutions
Supervised versus Unsupervised ML

When Labelled Data is Available
- Classification of good versus bad TCP flows with labeled data

When NO Labelled data is Available
- Learn underlying rules in the data
- Classification of abnormal behavior in unlabeled data* (Covid effect on NERSC)

*Unsupervised Traffic Analysis of Data Centers, SMC 2020
Reinforcement learning (RL)

When NO data is available
Learn via trial and error

• Agent views the state and chooses an action
• Gets a reward
• Over time, learns optimal actions for best rewards

Self-discovery: Learning flow performance over various paths
Mathematics of AI
AI has Much Randomness

- AI researchers still exploring:
  - Explainable AI
  - Uncertainty quantification
- Choosing appropriate ML algorithms – different accuracies
- Hyperparameter tuning can dramatically improve the accuracy
- AI training needs large training data, often gets stuck in local optima
- Optimization functions can affect how well the NN learns

Levemberg-Marquadt (LM) can lead to better solutions than Stochastic Gradient Descent (SGD)*

AI is Power Hungry

- Access to advanced hardware matters for training complex models
- For (training) Learning-at-the-edge, using GPUs and FPGAs

---

* Deploy Models into Production
Accessing the ‘Right’ Data from Hardware

- Testbeds and Hardware capability available (e.g. kind of switches, sflow-enabled)
- Most monitoring can introduce ‘noise’ (e.g. UDP, ICMP packets)*
- Available training data for research

*NetGraf : A collaborative Network Monitoring Stack for Network Experimental Testbeds SC20 poster
Handling Streaming AI for Massive Network Data

- Leveraging data tools e.g. splunk
- Building fast streaming solutions
- Security and access
- Build streaming AI solutions becomes a software challenge

<table>
<thead>
<tr>
<th>Packet size</th>
<th>Rate</th>
<th>Telemetry PPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500B</td>
<td>10Gb/s</td>
<td>812K</td>
</tr>
<tr>
<td>1500B</td>
<td>100Gb/s</td>
<td>8,127K</td>
</tr>
<tr>
<td>9000B</td>
<td>10Gb/s</td>
<td>138K</td>
</tr>
<tr>
<td>9000B</td>
<td>100Gb/s</td>
<td>1,383K</td>
</tr>
</tbody>
</table>

*R. Cziva’s (ESnet) work
Learning Features: Correlation Patterns

Correlation = \frac{\text{Cov}(x,y)}{\sigma_x \ast \sigma_y}

First 16 weeks of year 2019 and shows network is extremely dynamic
Learning Communities: Forming and Dispersing

\[ C_i(t_k, t_{k+1}) \propto ||S_i \epsilon(t_i, E_i, N_i)|| \]

Clustering Coefficient with temporal perturbation recording edge-change

Learn ‘Day and Night’ Traffic patterns
Graph Neural Networks for Traffic Prediction

+ features

Input
\( W=36 \) hrs

Multiple layers of bi-directional recurrent NNs

Output
\( W=24 \) hrs

- **Benchmarking** Prediction algorithms
- No need to learn long term traffic patterns (short term is enough)
- Deployed for internal testing with real-time data

*Data-driven Learning to Predict WAN Network Traffic, SNTA’20
*Dynamic Graph Neural Network for Traffic Forecasting in WAN, IEEE BigData20
Model Development Cycle

- Use HPC to optimize model training and “save model”
  - Costly and time consuming
- Model shift: Testing to update where the model is not performing well and needs a rethink
NetPredict: Predict traffic 1 week ahead

An easy to use ‘Google-map’ style to offer Bandwidth Usage Prediction for Planning and Scheduling.
NetPredict: Google Map for ESnet

Source and Destination

Hourly traffic prediction for next 1 week

Alternate path at a different time

4:00pm

5:00pm

Gives the least congested path in that hour

*SC19 Demo Network Research Exhibitions
DeepRoute: Reinforcement Learning for Traffic Engineering

- Multiple ways between source and destination
- Learning the Q function (value function or Bellman equation):
  \[ Q(s, a) = R(s, a) + \gamma \max_{a'} Q(s', a') \]
- Learned States: Current utilization, loss and flow details \((u, l, f)\)

<table>
<thead>
<tr>
<th>state</th>
<th>action</th>
<th>Reward (flow end time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>State0</td>
<td>Move path1</td>
<td>10</td>
</tr>
<tr>
<td>((u, l, f))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>state1</td>
<td>Move path2</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>..</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>..</td>
</tr>
</tbody>
</table>
Improving Flow Completion Time

One dashboard for all monitored data
(loss, utilization, flow)

Learn optimal allocations for all kinds of flows

*DeepRoute: Herding Elephant and Mice Flows with Reinforcement Learning, MLN 2019
Data-driven Learning for Better Paths

- Can perform multi-objective optimization

Multi-objective optimization = Min (flow time) Min (packet loss)

- MAMRL – Multi-agent meta reinforcement learning: Higher network load better copes with our approach

Assumption: one flow per iteration

*Multi-agent meta reinforcement learning for packet routing in dynamic network environments, SC20*
Summary

- Self-driving (autonomous) networks are within reach, with clear objectives
- Share solutions among groups, data and software tools
  - Research papers often omit key aspects of deploying the ‘AI’
  - Having Off-the-shelf solutions such as data and tools
- PROPOSAL: Extending NetPredict to other R&E Networks and Groups
- Hardware is expensive, but open testbeds are helping and are necessary (e.g. FABRIC, Chameleon)
Thankyou for listening!

- More information:
  - Open for Collaboration!
  - Deep Learning and Artificial Intelligence for High-Performance Networks (DAPHNE Lab)
    
    [https://sites.google.com/lbl.gov/daphne](https://sites.google.com/lbl.gov/daphne)
• Extra slides
How to action

• Networks engineers should work with software engineers
• Software engineers are trained to work with data processing challenges, streaming data, tools, testing, etc
• We should be forming a community:
  – Think of what data you can make available
  – Think of what software library also
• How to start: Find a problem with concrete goals of what the ML application should do
  – Collaborate with ML enthusiastic (e.g. students)
  – Deploy, test and test
    • “Search for Bounds at which the AI will fail”
More Problems to Explore

- Integrating AI with Control plane but also AI at the Edge
  - Filter traffic using classification techniques
- Optimizing network configurations
  - Checks, configs
  - Debugging
- Optimize engineer’s times
  - Timely alerts if something wrong, bad behaving routers/switches
  - Vendors offering this but it is quite easy to do yourself
- Unexpected versus expected network behaviors
  - Topology or flow changes, day and night patterns
- Intent-driven networks
Testing Multiple ML models

Multiple models tested in real time

MSE is growing, model not relevant any more
Supervised Learning

When Labelled Data is Available

- Learning to recognize the labels
- Classification, object detection, anomaly detection
- Works very well if we can identify clear class boundaries

Good versus bad flows

Plot of anomalous TCP traffic
Unsupervised Learning

When NO Labelled data in Available

- Learn underlying rules in the data
- Clustering, feature identification, recognize anomalies in test data
- Need domain scientists after clusters are recognized

Visualizing NERSC flows into clusters

Plot of NERSC netflow transfers

*S. Campbell, F. Bannatwala, SMC 2020*
Keep Trying different AI models

- No need to learn long term traffic patterns (short term is enough) [4]
- Developed our own Graph neural networks to improve prediction results
- Can still be active research
  - Improving time predicted
  - Improving AI model itself

• NN models pretrained on HPC facilities (NERSC)
  – 2 years of SNMP data
  – Save all AI models
• Deployed on Google Cloud Platform
  – Buckets save tensorflow models
  – VMs to do inference
  – App to visualize results

*SC19 Demo Network Research Exhibitions
LSTM to model time as Sequences

- Time as discrete sequences
Add features to LSTM to predict future

- Further signals which might affect data, e.g. time of day, weekday, weekend