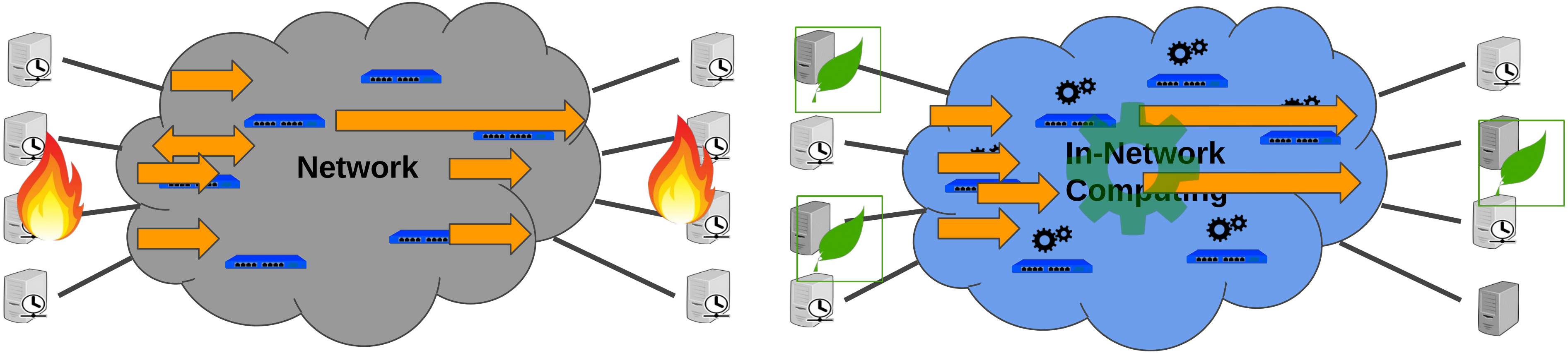
Problems and Strategies implementing in-network AI models

Presented by
Matthews Jose
INRIA, Nancy - Grand Est
Orange Labs, Chatillon

In-network Computing

Power savings
Latency reduction
Remove congestion

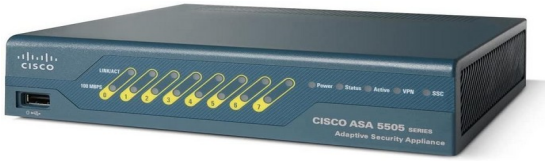


In-network AI and its applications

- Security applications : DDoS detection algorithms, anomaly detection using classification.
- Low Latency Applications: Services on connected cars like DNS
- Scheduling and congestion control algorithms
- Network aggregation: Accelerate training process for models
- Eg. Device recognition in IoT environments, BNN for analyzing processing overheads

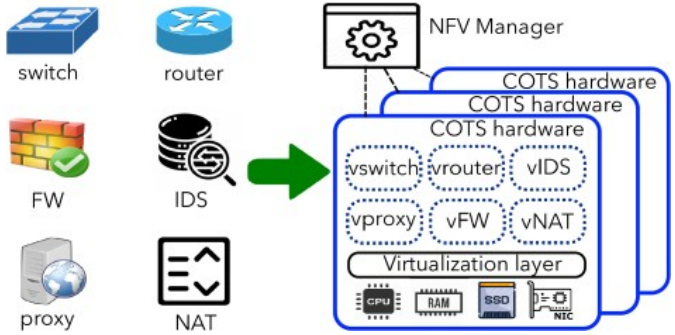
Hardware considerations

Dedicated Custom hardware



x86 middle boxes
implemented in software

NFV, virtualization software
based middle boxes



- Based on reconfigurable match tables (**RMT**) model in SDN.
- Languagues like P4 allow custom headers and action logic
- P4 code is compiled and deployed via the controller onto the switch

Limitations, drawbacks

Middle Boxes

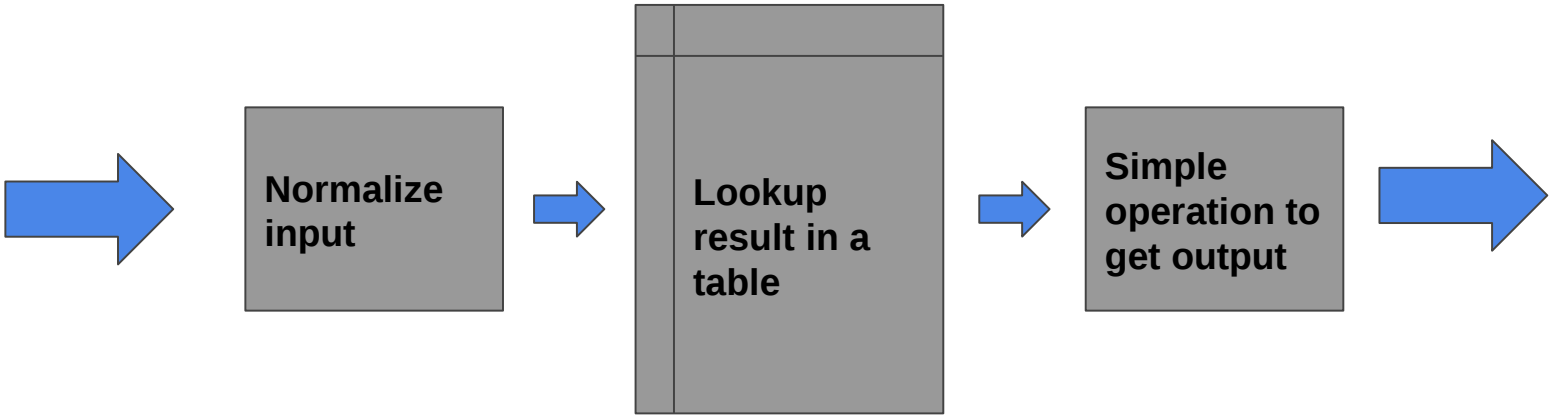
- Increase latency of traffic
- Middle boxes are generally purpose built and not flexible
- Additional costs for new hardware
- Networks get clogged with these devices

Programmable Switches

- Not much onboard memory
- No support for floating-point numbers and operations
- Program size and complexity limitations
- Basic instruction set
- Linear in nature; hard to implement structures that have loops
- Complex feature sets not available on switches

Proposed Solutions

Lookup Tables for Complex Elementary Operations



In-network Inference made easy (IIsy) is one such framework that makes use of this method to perform mathematical operations.

Zhaoqi Xiong and Noa Zilberman. 2019. Do Switches Dream of Machine Learning? Toward In-Network Classification. In Proceedings of the 18th ACM Workshop on Hot Topics in Networks (HotNets '19). Association for Computing Machinery, New York, NY, USA, 25–33.

Proposed Solutions

Using simpler encoding schemes like fixed point numbers instead of floating point numbers to represent real numbers

SwitchML is aggregates intermediate data in a distributed model training cluster. The aggregation is performed is partly performed in-network and is provides upto x2.1 speedup

A. Sapio, M. Canini, C.-y. Ho, J. Nelson, P. Kalnis, C. Kim, A. Krishnamurthy, M. Moshref, D. Ports, and P. Richtarik, “Scaling distributed machine learning with in-network aggregation,” in KAUST. KAUST, 2019.

Proposed Solutions

Addition Header fields to carry intermediate data and features

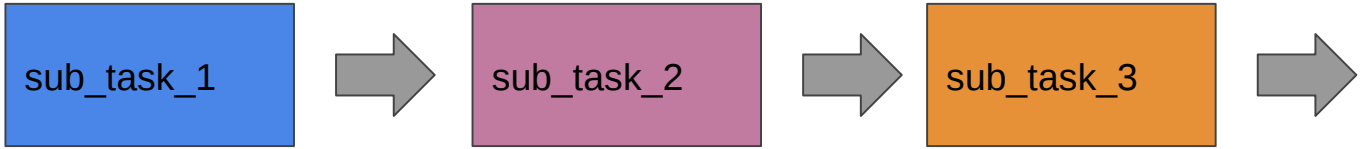
Custom values	Feature values	Ethernet	IP packet	…...

iSwitch is a solution that accelerates training by performing in-network aggregation for distributed RL training. The programmable switch is used to carry intermediate aggregated states.

Y. Li, I. Liu, Y. Yuan, D. Chen, A. Schwing and J. Huang, "Accelerating Distributed Reinforcement learning with In-Switch Computing," 2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, 2019, pp. 279-291.

Proposed Solutions

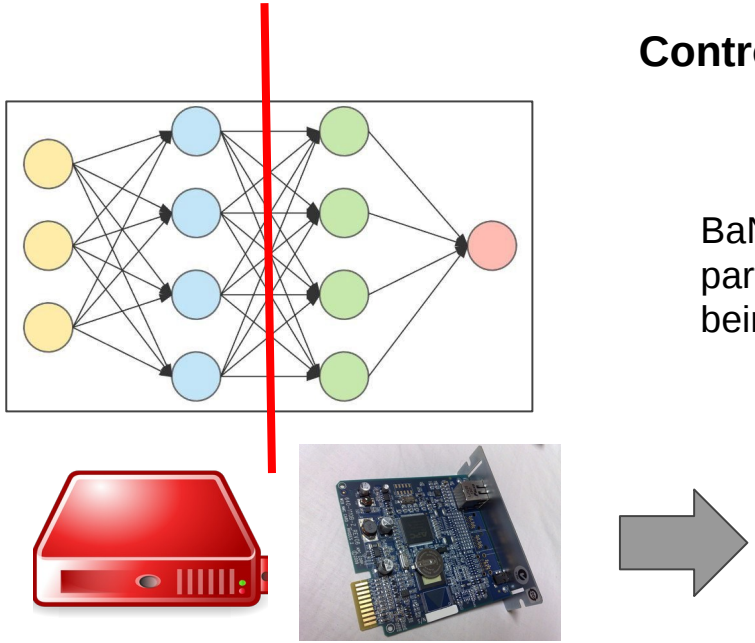
Dividing tasks amongst several switches



Divide tasks across multiple switches and use packet headers to carry intermediate states between them.

Mina Tahmasbi Arashloo, Yaron Koral, Michael Greenberg, Jennifer Rexford, and David Walker. 2016. SNAP: Stateful Network-Wide Abstractions for Packet Processing. In <i>Proceedings of the 2016 ACM SIGCOMM Conference. Association for Computing Machinery, New York, NY, USA, 29–43.

Proposed Solutions



Controller assisted model computation

BaNaNa Split is divides the processing of a BNN into 2 parts to be run on the CPU and on the NIC card before being written to a packet.

Davide Sanvito, Giuseppe Siracusano, and Roberto Bifulco. 2018. Can the Network be the AI Accelerator? In Proceedings of the 2018 Morning Workshop on In-Network Computing (NetCompute '18). Association for Computing Machinery, New York, NY, USA, 20–25.

INReC : enabling floating-point operations in-network

- A framework for generating pipeline for real-valued functions
- Enables deployment of small ML models in-network

1. Define procedures of compute elementary operations in P4

addition, (x + y) :

1. Find difference between exponent part of floating point numbers
2. Binary left shift manitssa of smaller number by difference
3. Add the resulting manitissa to mantissa of larger number
4. Compute the sign and normalize mantissa using conditional check

logarithm base 2, $\log_2$ (x) :

1. Use a lookup table to get value of log(mantissa of X)
2. Add the exponent of X to the result of step 1
3. Normalize result of step 2

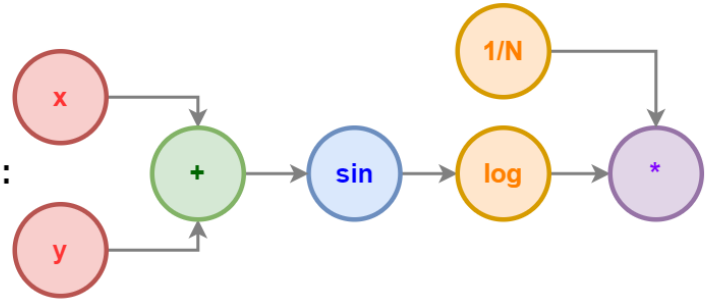
division, x / y :

1. Compute $\log_2$ (x) and $\log_2$ (y)
2. Subtract first result from the second
3. Compute 2 power the result from step 2

M. Jose, J. Franc ̧ois, and K. Lazri. (2021) In-REC: In-network REal Number Computation. Bordeaux, France. [Online]. Available: https://1drv.ms/b/s!AmSZGUizmAdemza-yiMklMCAWAWt?e=oWGhQQ

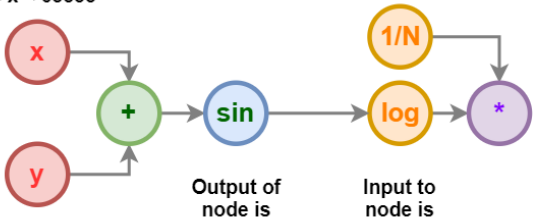
2. Decompose function into elemetary operations and map dependencies via a DAG

$f(x , y) = (1/N) * \log(\sin(x + y)) :$



3. Optimize DAG by placing constraints, aggregation and reducing

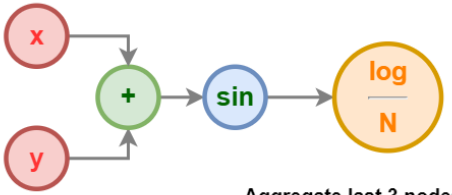
If x is the bandwidth of port then
0 < x < 65353



Output of node is between -1 and 1

Input to node is between -1 and 1

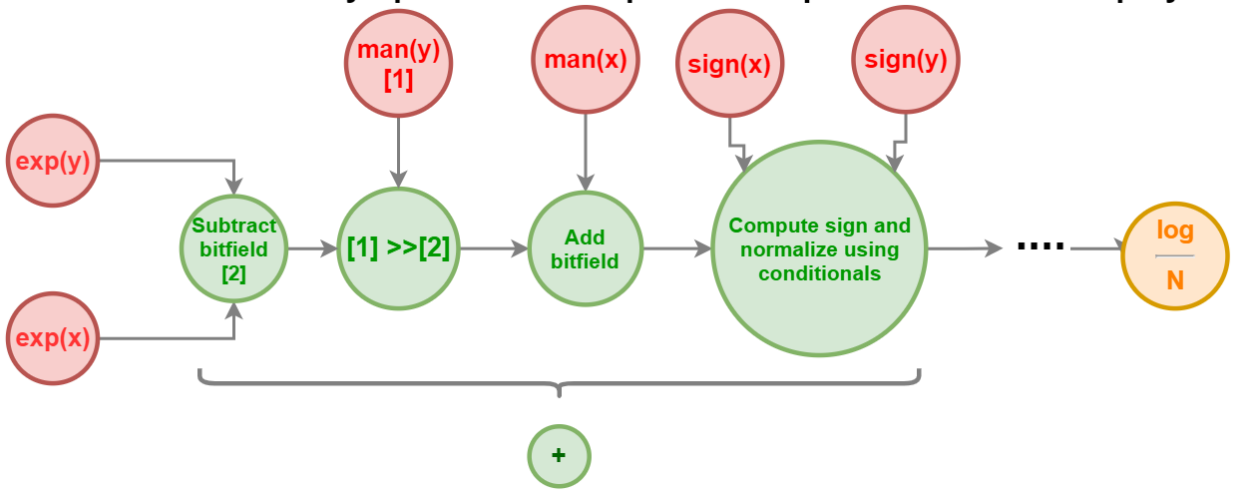
Place constraints



Aggregate last 3 nodes by replacing with a lookup table for: g(w) = log(w) / N where -1 < w < 1

Aggregate bounded nodes

4. Subtitute elementary operations with predefined procedures and simplify



5. Perform switch specific optimizations: parallel processing, memory optimizations, key match selection using

1) solving a linear program for selecting memory used for the lookup table
2) mapping to rectangle packing problem find optimal lookup table size

Conclusion

- In-network computing offers several advantages like: increase in throughput, latency reduction and power savings.
- Middle boxes and programmable switches are the 2 most common ways in which in-network computation is achieved.
- Programmable switches despite being fast are heavily resource limited.
- Several work arounds have been proposed to overcome this like: Lookup table computation, task splitting, assisted computation etc.
- INReC is a framework used to generate switch pipelines for real-valued functions.

Thank you!
Questions?