RL Agents Inside Edge Devices

Zhan Xu

Director, Research Team of IC Lab, Vancouver

Huawei Technologies Canada Co., Ltd.

Contents

1. AI is moving from the clouds to the edge devices;
2. Apply RL to improve the user experience based on user operations;
3. Apply RL to help to automate some decision making and search process;
4. The major challenges we are facing to apply RL into our applications.

AI moves toward edge devices, it is the production requirements

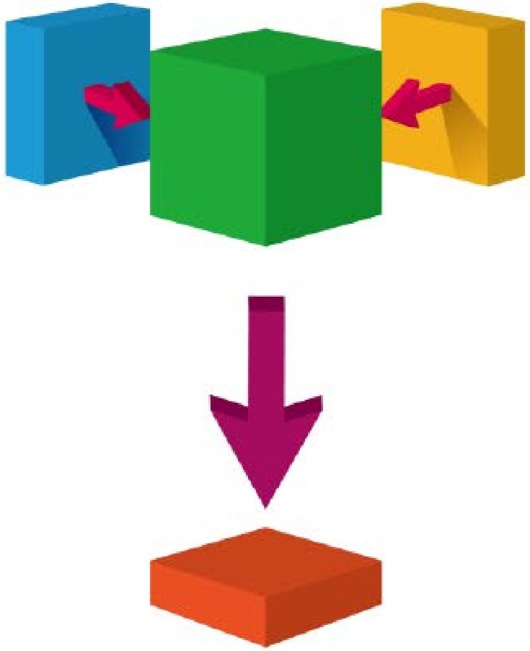
1. In the end, all the AI algorithms needed to be integrated into a real product.
2. With the advance of 5G networks, IoT devices will be everywhere.
3. Following are the major advantages when AI functions running inside a edge device :

- Low Cost (for high volume),
- Reliability and quick to respond,
- Privacy,
- Low Latency,
- Network bandwidth.

With the help of ASIC for stronger computation power, AI function on edge become possible!

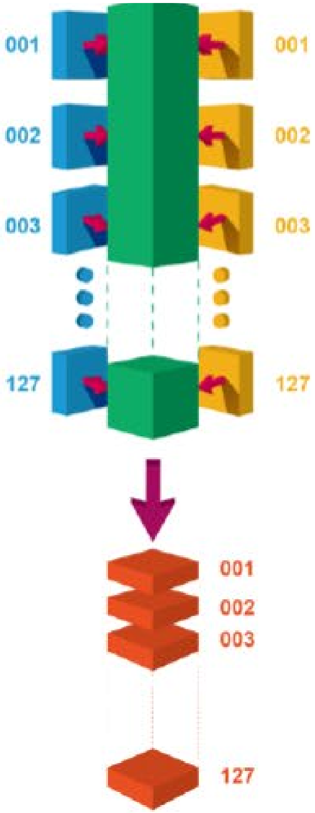
CPU

scalar (numeric) operations



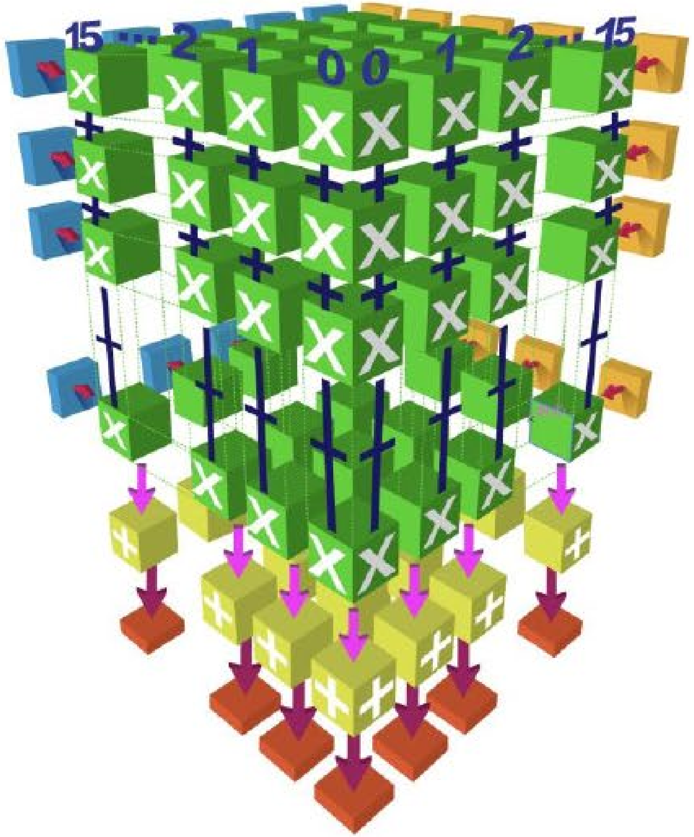
GPU

Vector operations



TPU / NPU

Matrix cube operations



Challenges of AI on Edge Device: How to adopt to the environment?

- Current industrial successes are mostly based on supervised learning, we are still unable to discover higher-level abstractions. That’s the limitation of CNN, that’s why we need so many training data, like ImageNet (14 M images!), we need hundreds or thousands dog images in order to recognize dog.
- Those algorithms need to learn from the environment individually, like model update and customization, federated learning, to improve the one-to-all generic model.

Challenges of AI on Edge Device: Online learning

- After the algorithms and NN models are deployed into hundred millions of devices, they do not change to adapt to each individual user, not evolve with the different environment.

- Each device (smart phone, auto drive car and video surveillance camera etc.) can be deployed in different places, but normally it will keep working in the same environment (or used by the same person) afterward. This makes online learning possible!

Could RL come to help?

Naturally, every edge device can act like RL agent, interacting with its surrounding environment.

With help of RL, how to improve the user experience based on user operations and event incidents?

How to improve the user experience based on user operations?

Some examples:

- Learn from mistakes of AI modules on smart phone;
- Learn to save battery power on smart phones;
- Learn to reduce false alarms for smart home cameras;
- Enhance user engagement for game playing.

Learn from mistakes on smart phone

One example of smart phone application - document photography:

- In this mode, the phone automatically detect the document area, crop and correct the distortion, to create rectangle document;
- However, some cases, the cropping can be wrong;
- After user delete the wrong picture and take again, the cropping still wrong.
- How can the agent inside learn from those mistakes?

Learn to save battery power on smart phones

How to balance the user experience and battery saving?

- On smart phone, apps running at the background will be killed after certain amount of time, to save battery power.
- However, when the user opens this app, he/she will to wait to restart the app from scratch, not very user friendly!
- How to apply RL agent, to learn user behaviors, knows how long to terminate app, and which app should be left running for longer time.

Learn to reduce false alarms for smart home cameras

- False alarm is by-far the biggest problem in smart home applications;
- Some time the same false alarms being triggered and reported again and again;
- Learn to reduce the false alarm: for reported alarms, if user can response and give feedback, it can an explicit reward; if the user never response for same repeated alarm, it can be treated as an inexplicit reward;

Enhance user engagement for game playing

- To provide the best user experience during the game playing, it is best to play at the similar level of the player;
- Learn the user’s level as soon as possible, by trying to explore different moves at different skill levels.;
- The game agent lifts the skill level when the user’s skill grows;

With the help of RL, how to automate some decision making and search process?

Two examples:

- RL agent inside SDN network routers, to alleviate the network congestion and improve load balance;
- Use RL for neural network architecture auto-search, vs. hand crafted neural network structures;

RL for SDN-based network flow control

Problems to solve:

- Congestion alleviation, better load balancing, attack defense.
- Best policy to select packet size, local buffer size, number of ports, routing tables etc.
- The rewards will be the congestion measurement, bandwidth usage, time delay etc.

RL for Neural Architecture Search (NAS)

Problems to solve:

- NAS is now a very active research area, as of mete learning.
- Compared to VGGNet, ResNet, DenseNet, etc. hand crafted architecture, we can use RL for application-specific NN architecture search, it will potentially generate better results, and quick adoption.
- The parameters to learn will be layers of NN, Convolution filter size/stride, number of channels, pruning, weight sharing, activation function type, pooling function types etc.

The major challenges we are facing to apply RL into our applications

The challenges include:

- How to effectively define and formulate the problems as an abstract MDP models?
- How to collect rewards when there is sparse feedback or even no explicit rewards at all?
- How to create a light-weight RL framework or engine, able to run at the background inside the devices? Considering foot-size of memory and CPU usage, general enough to handle different problems etc.

Thank you!