Enabling Distributed DNNs for the Mobile Web Over Cloud, Edge and End Devices

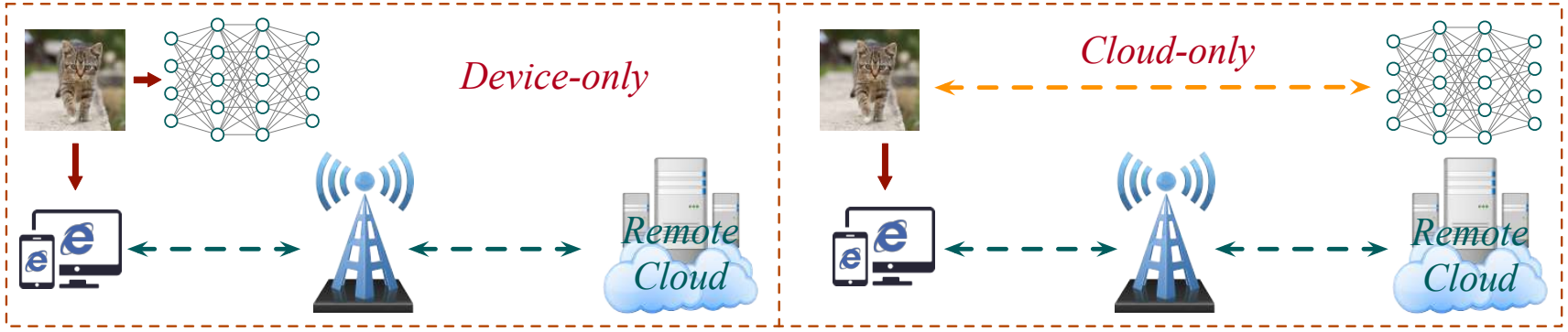
Yakun Huang & Xiuquan Qiao
Beijing University of Posts & Telecommunications

Outline

- Overview of executing DNNs on the Web
- Enabling Distributed DNNs for the Web with edge computing
- Thinking and Discussion

Overview of executing DNNs on the Web

- Deep neural networks (DNNs) show great promise in providing more intelligence to the web applications.
- Two typical DNNs execution schemes on the Web



Executing the whole DNNs on the remote Cloud

- Large amounts of data (e.g., image, audio and video) are sent to the cloud
- Increasing the computing pressure
- Raising new privacy concerns for users (e.g., home security cameras)

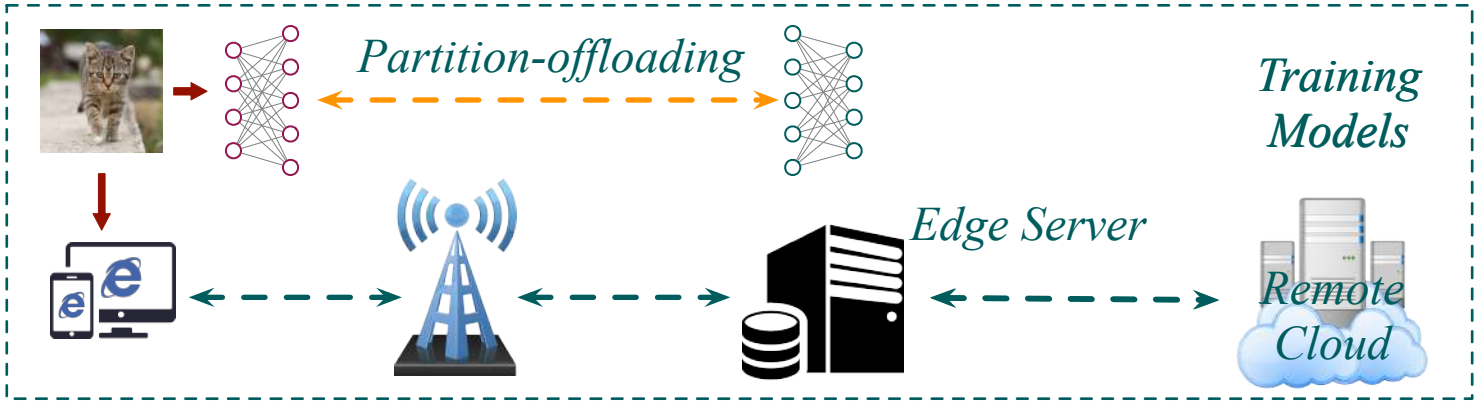
Executing the whole DNNs on the Web

- Limited computing resource
- Heavy DNN models (e.g., Tensorflow.js's ResNet50 deep learning model, whose size can be up to 97.8 MB)

Caffe.js | ONNX.js | Keras.js | WebDNN | Paddle.js | Tensorflow.js

Accelerating Distributed DNNs for the Web with edge computing

- Partition-offloading dynamically distributes the computations between the Web and the edge server.
  - Partitioning and performing the computation that can be done within the Web.
  - Protecting data privacy and reducing the computing pressure of the edge server.



Benefits of the edge computing

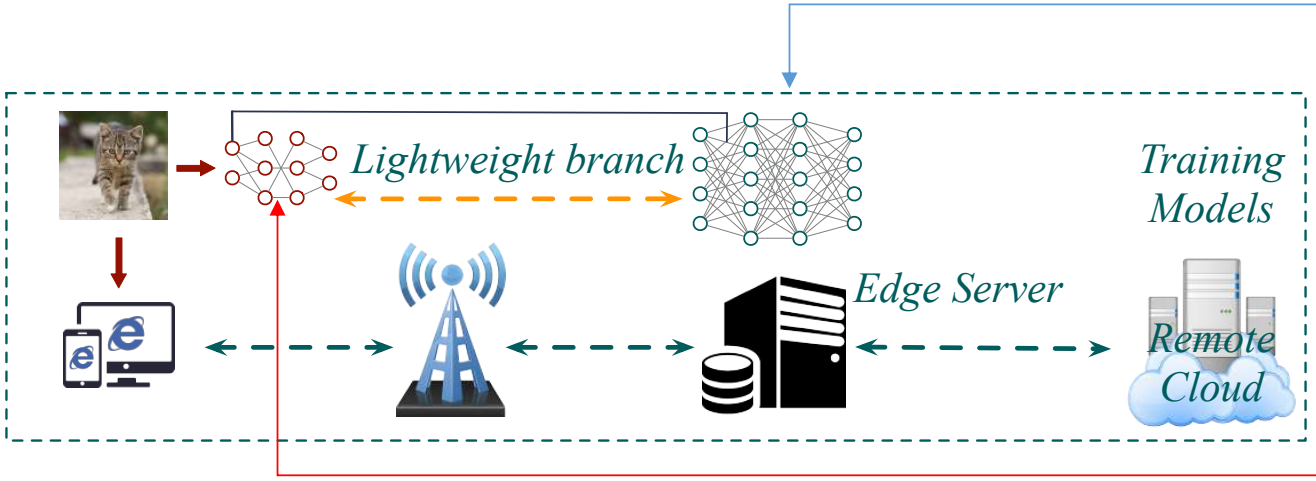
- High network bandwidth
- Low transmission latency

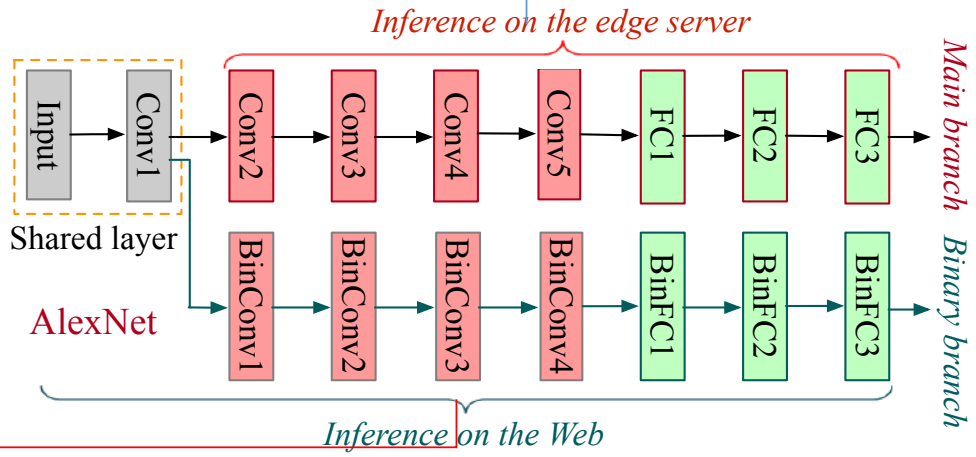
- Providing dynamic DNN partition to cope with various tasks, network conditions and the computing capacity of the Web

Accelerating Distributed DNNs for the Web

- Adding an efficient branch to the traditional DNNs for executing inference on the Web independently
  - providing a collaborative mechanism with the edge server for accuracy compensation.
  - Reducing model size and accelerating inference on the Web.

3) Otherwise, it asks the edge server for collaboration by posting outputs of the shared layer.

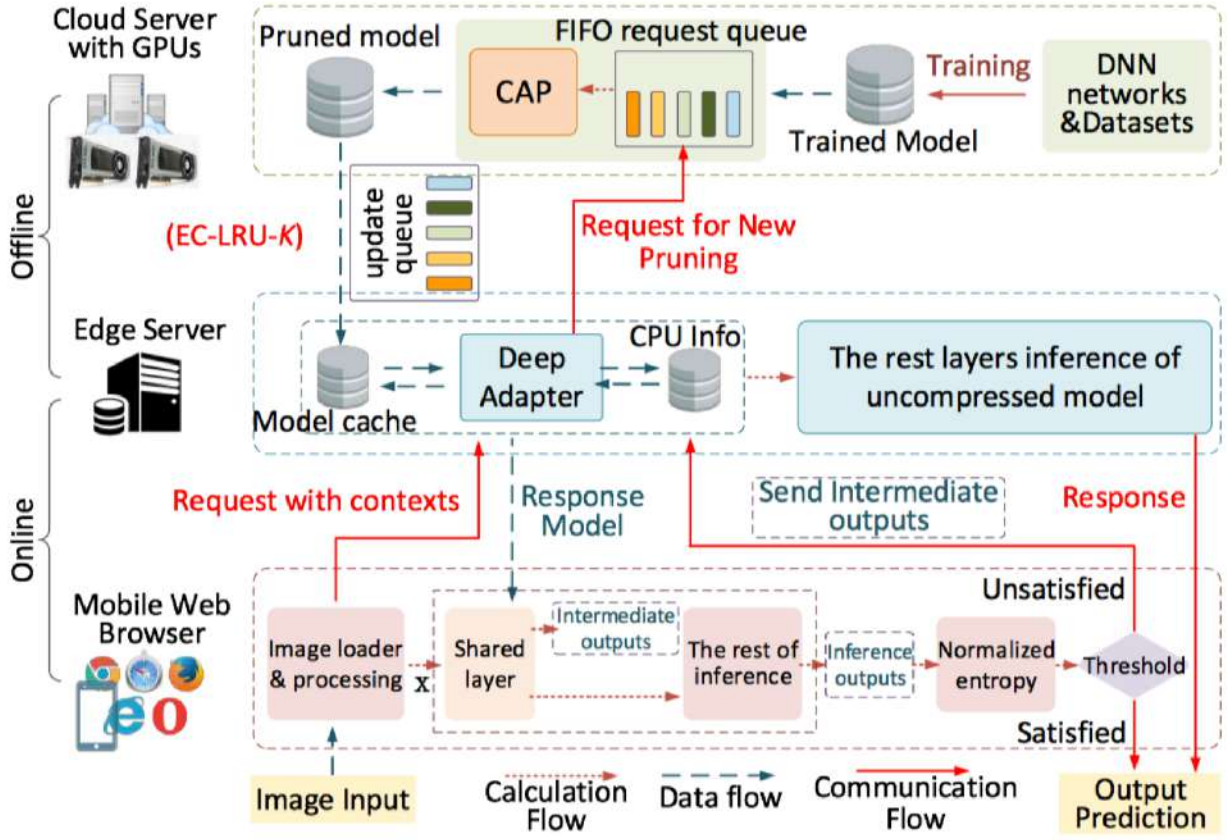




1) Measuring the confidence of inference results using the normalized entropy.
2) Exiting from the lightweight branch if there is confidence in the inference result.

Accelerating Distributed DNNs for the Web

- Providing a context-aware pruning algorithm that incorporates the latency, the network condition and the computing capability of the mobile device.



- **Offline Phase**
  - Incorporate dynamic contexts in the Network pruning.
  - Establish a synchronization mechanism to support the provision timely models.
- **Online Phase**
  - Employ a context-aware runtime adapter to provide appropriate pruned models.
  - Obtain the pruning requirement from the request queue, and then executing pruning.

Thinking and Discussion

- What role should the edge server play in providing processing support for intelligent web applications requiring heavy computation?
- How to web developers use the edge server more easily for accelerating DNNs and collaborating with Web apps?
- How can the edge server deploy and offload DNN computations more easily?
  - How Web apps can monitor changes in network conditions, and response to the edge server?
  - How the edge server can perceive the computing capability of the device, and distribute appropriate computations to Web apps for execution in real time?

Thanks!