Efficient Low-Latency Dynamic Licensing for Deep Neural Network Deployment on Edge Devices

Toan Pham Van[1] Hoang Pham Minh[1] Ngoc N. Tran[1]
Tam Nguyen Minh [1] Ta Minh Thanh[2]

[1]R&D Department
Sun-Asterisk Inc.

[2]Faculty of Computer Science
Le Quy Don Technical University

The 3rd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2020)

Table of Contents

Table of Contents

Motivations

Commercial AI applications are becoming mainstream.

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
  - Complete server-side solutions require unrealistic server scaling.

Sun*

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
  - Complete server-side solutions require unrealistic server scaling.
  - Network bandwidth will be severely strained.

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
  - Complete server-side solutions require unrealistic server scaling.
  - Network bandwidth will be severely strained.

  $\rightarrow$ Push workload to the edge!

Sun*

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
    - Complete server-side solutions require unrealistic server scaling.
    - Network bandwidth will be severely strained.

    $\rightarrow$ Push workload to the edge!
- Edge devices' computational power is limited.

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
  - Complete server-side solutions require unrealistic server scaling.
  - Network bandwidth will be severely strained.

    $\rightarrow$ Push workload to the edge!
- Edge devices' computational power is limited.

  $\rightarrow$ Need model optimization!

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
  - Complete server-side solutions require unrealistic server scaling.
  - Network bandwidth will be severely strained.

    $\rightarrow$ Push workload to the edge!
- Edge devices' computational power is limited.

  $\rightarrow$ Need model optimization!
- Commercial AI apps requires versioning and licensing.

Motivations

Commercial AI applications are becoming mainstream.

- Traditional infrastructures will not be able to keep up.
  - Complete server-side solutions require unrealistic server scaling.
  - Network bandwidth will be severely strained.

  $\rightarrow$ Push workload to the edge!
- Edge devices' computational power is limited.
  $\rightarrow$ Need model optimization!
- Commercial AI apps requires versioning and licensing.
  $\rightarrow$ We can extend our system to work with that.

Preliminaries

Cloud-based AI vs Edge-based AI

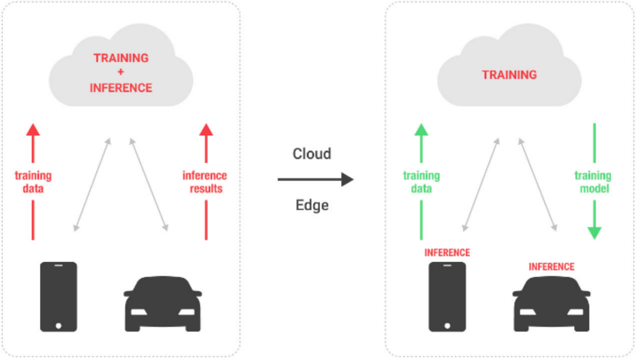

Figure: Cloud-based DNN (left) vs Edge-based DNN architecture (right)

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called “deep” because it has many layers

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called "deep" because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called “deep” because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights
    - An extreme example: GPT-3 has 175 **billion** parameters!

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called "deep" because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights
    - An extreme example: GPT-3 has 175 **billion** parameters!
    - Various optimizations needed

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called "deep" because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights
    - An extreme example: GPT-3 has 175 **billion** parameters!
    - Various optimizations needed
- Model compression techniques

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called "deep" because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights
    - An extreme example: GPT-3 has 175 **billion** parameters!
    - Various optimizations needed
- Model compression techniques
  - Model pruning

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called "deep" because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights
    - An extreme example: GPT-3 has 175 **billion** parameters!
    - Various optimizations needed
- Model compression techniques
  - Model pruning
  - Quantization

Preliminaries

Deep Neural Networks

- Basically a complex composition of simpler functions
  - Called "deep" because it has many layers
  - Represented by (lots and) lots of numbers called parameters/weights
    - An extreme example: GPT-3 has 175 **billion** parameters!
    - Various optimizations needed
- Model compression techniques
  - Model pruning
  - Quantization
  - Weight sharing

Preliminaries

Database and Query

- Database

Preliminaries

Database and Query

- Database
  - Relational database

Preliminaries

Database and Query

- Database
  - Relational database
- Query

Preliminaries

Database and Query

- Database
  - Relational database
- Query
  - RESTful API

Preliminaries

Database and Query

- Database
  - Relational database
- Query
  - RESTful API
  - GraphQL

Table of Contents

Our architecture

- We split the traditional unified cloud into a training server and a weight storage server.

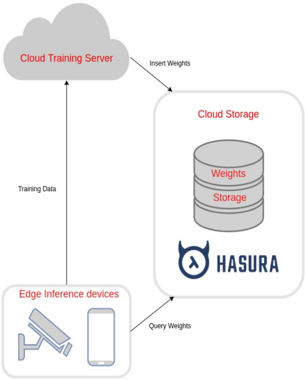

Figure: Our architecture with weight storage in database

Our architecture

- We split the traditional unified cloud into a training server and a weight storage server.
- Weight database is designed for extensive versioning.

Our architecture

- We split the traditional unified cloud into a training server and a weight storage server.
- Weight database is designed for extensive versioning.
  - Store each model version with their last update time

Our architecture

- We split the traditional unified cloud into a training server and a weight storage server.
- Weight database is designed for extensive versioning.
  - Store each model version with their last update time
  - Also store individual weights with their last update time

Sun*

Our architecture

- We split the traditional unified cloud into a training server and a weight storage server.
- Weight database is designed for extensive versioning.
  - Store each model version with their last update time
  - Also store individual weights with their last update time

  → Allows for model updates only when needed, whichever part needed.

Model Compression

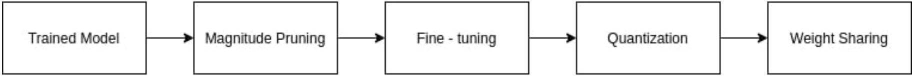

Figure: Model compression pipeline

Weight Licensing

Algorithm 1 Pruning model based on accuracy

divide weight range into k smaller equal-sized intervals
initialize a list of cut-off intervals
for all intervals **do**
    for all model's layers **do**
        cut off weights that have values in that interval
        append interval into cut-off interval list
        if accuracy of pruned model is close to the target **then**
            break the pruning process
        end if
    end for
end for
return uncut interval lists

Table of Contents

Sun*

Why choose our method?

- Efficient Deployment

Why choose our method?

- Efficient Deployment
- Version Management

Why choose our method?

- Efficient Deployment
- Version Management
- Low-Latency Update

Why choose our method?

- Efficient Deployment
- Version Management
- Low-Latency Update
- Dynamic and Static Licensing

Table of Contents

Sun*

System setup

- Django Framework
- Keras/TensorFlow
- PostgreSQL
- Hasura
- Docker

Results

Table: The cost of memory storage

No. of params	Full params	Pruning 80%	+ Quantization
109386	13MB	2.92MB	2.34MB
101770	12MB	2.65MB	2.09MB

Thank you for listening!