Deep Learning Training on Distributed Embedded Systems

Sihyeong Park, Jemin Lee, Hyungshin Kim
Chungnam National University
Introduction

• Deep Learning
  • Machine learning based on multi-layer artificial neural network
  • Largely divided into two: Training, Inference

• Excellent performance of Deep Learning
  • Computer vision, natural language processing, speech recognition, self-driving car, etc.
Introduction

• **Training**
  • Generate a model with input data
  • Consists of repeatedly calculating multi-layer
  • Typically executed on a high-performance distributed system that supports multiple GPUs

• **Inference**
  • Perform for certain tasks such as object detection

• **Embedded System has limited resource**
  • Existing studies are not considered full-training on embedded systems without cloud offloading
Introduction

• Smart Home Systems
  • Diffusion of Internet-of-Things (IoT)
  • Many home appliances are applying deep learning
    • High-performance CPU like quad-core is used for appliances
    • Make easier to implement a distributed processing system using embedded systems

• Risk of personal information leakage
  • Devices collect data from their sensors and then send them to servers for training
  • Smart home systems incorporate an amount of sensitive information
Proposal

• Distributed Deep Learning using Smart Home System
  • In-Home local training, without server
  • Online learning about change of environment such as change of furniture layout and change of living pattern of people
  • Privacy protection possible
  • Reduced server usage costs
Experiment Environment

• Distributed Deep Learning framework
  • MXNet
    • Provides data and model parallel processing
    • Mobile devices to multi-GPU, multi-Devices
    • Distributed key-value storage based on parameter servers for synchronization

• Distributed computing
  • Raspberry Pi 3 Model B: 1~11
  • Connected with 100 Mbps LAN switch
Experiment Environment

- **MNIST dataset**
  - Handwritten image of 0-9
  - 60,000 32x32 size images

- **LeNet-5**
  - 3 Convolution layers
  - 2 sub-sampling layers
  - 1 fully-connected layer
Experiment Result

- **Layer size**: 64
- **Learning rate**: 0.05
- **Number of epochs**: 10

![Graphs showing time costs and training/validation accuracy](image_url)
Conclusion and Future Works

• Trained LeNet model in distributed embedded system and measured execution time and accuracy for a handwritten dataset

• Using a larger model and a heterogeneous embedded system
  • LesNet, AlexNet, etc.

• Using Wi-Fi to distributed computing