November 28, 2018

Distributed Transfer Learning on Embedded Devices

Hanumant Prasad Singh

Distributed Transfer Learning on Embedded Devices

Abstract:

An Internet-of-Things (IoT) platform that enables the retraining of machine learning models on embedded devices is described. The IoT platform utilizes transfer learning to retrain models in a cluster of IoT products connected to each-other in a local-area network (LAN), personal-area network (PAN), or wireless personal-area network (WPAN), to be reused for a similar purpose. Unlike current IoT platforms, the distributed transfer learning IoT platform does not need to utilize a centralized computing system, such as a cloud-computing server or a network server to perform model training, but rather execute this training in the cluster of IoT products. To reach this goal, in addition to transfer learning, the described IoT platform supports application programming interfaces (APIs) that specify a small portion of the existing pretrained model to be retrained, specify a data pipeline in the cluster of IoT devices to be used to retrain the model, and tune the model.

Keywords:

Internet-of-Things (IoT), transfer learning, distributed transfer learning, machine learning, application programming interface (API), embedded devices, pretrained model, neural network, convolution neural network (CNN)

Background:

Transfer learning is used to enable pretrained machine learning models to be utilized to create models for a related data set by modifying a small portion of a pretrained model and performing retraining with the new data set on the modified portion to produce a new, derived model. Thus, transfer learning reduces the computationally intensive task of training an entire model by training a small portion of the model. This allows a pretrained machine-learning model, which classifies certain types of images to be reused for other purposes.

An IoT platform can employ different machine-learning architectures, including, neural networks and convolutional neural networks (CNN) to support machine learning. Fig. 1 demonstrates an example of a neural network for image recognition using machine learning.

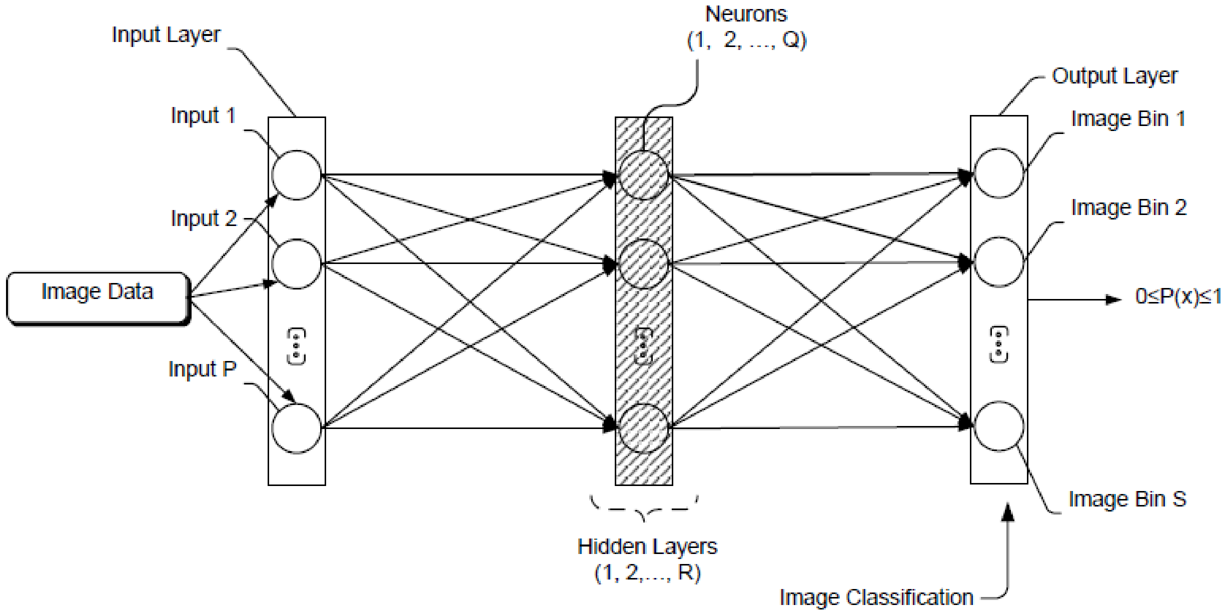

Fig. 1

The neural network in Fig. 1, illustrates an input layer, several hidden layers, and an output layer. The input layer includes P number of inputs. There are R hidden layers with up to Q neurons in each layer. The quantity of neurons in each hidden layer can differ. The output layer includes S bins with different probabilities as the correct outcome. The bin with the closest probability to one (1) is interpreted as the correct output.

With advancements in communication technologies and with computing/sensing electronics embedded in a myriad of devices, the ability for devices to collect and exchange data with one another is escalating. Devices such as smart phones, voice-recognizing personal assistants, computers, automobiles, home entertainment systems/appliances, and the like, are able to communicate with one another either directly, in a machine-to-machine environment, or indirectly over a network. Such communications and exchange of data across the myriad of devices is commonly referred to as the Internet-of-Things (IoT). The communications and exchange of data can have purposes that include, for example, collecting usage data for vendor analytics, remote initiation/shut-down of an operating system, automating a home environment, monitoring a person's health, and so forth.

A view of an example IoT environment is represented in Fig. 2 below:

Internet of Things (IoT)

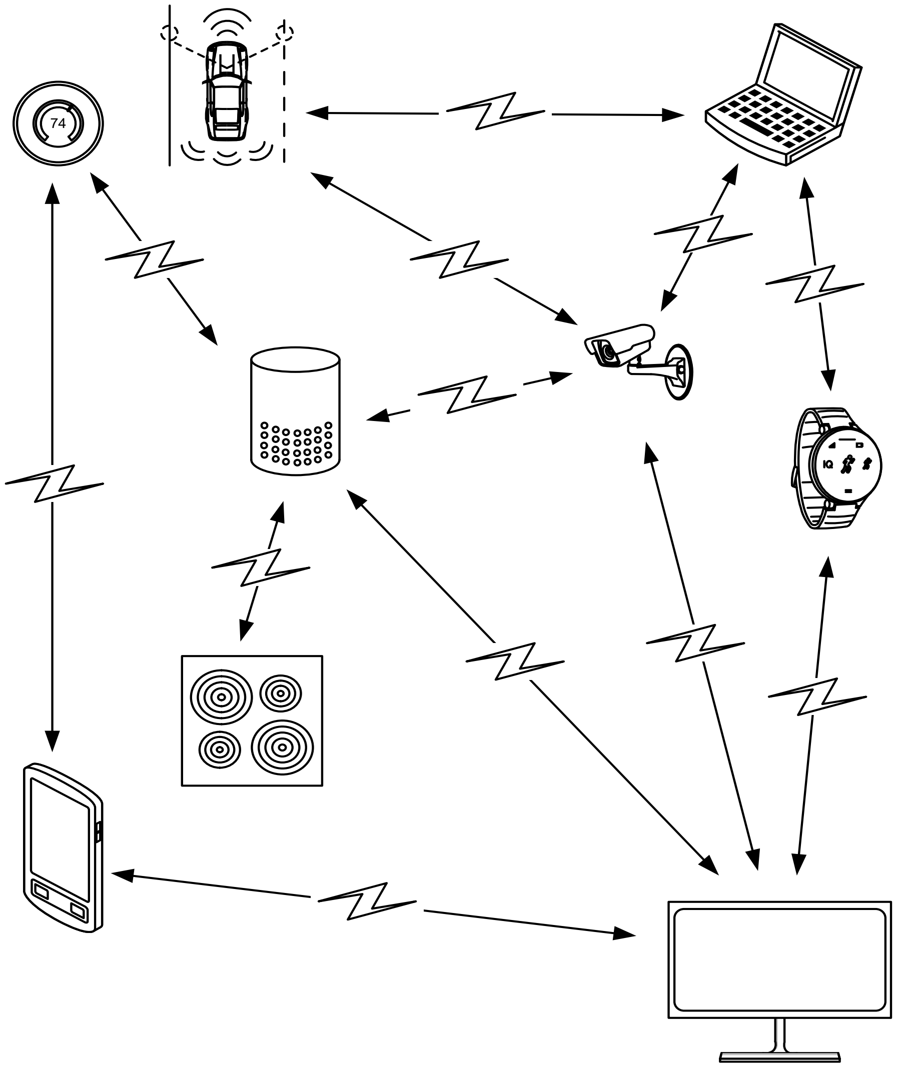

Fig. 2

In the IoT environment of Fig. 2, data may be collected by sensors of a device and shared with another device. Processing of data may be performed local to the device collecting the data or remote from the device collecting the data. Combinations of hardware (*e.g.*, sensors, microprocessors, memory), software (*e.g.,* algorithms, GUI's), and services (*e.g.*, communication networks) may be used to sense, collect, and exchange data. Large amounts of data are expected to be exchanged, as part of the IoT, across a horizon that is developing and changing frequently.

Detection mechanisms that may be built into IoT devices, such as light sensors, radar systems, proximity sensors, imaging sensors, cameras, or microphones, may measure conditions of an environment surrounding the IoT devices. Furthermore, and in some instances, computing algorithms may be applied to the conditions, as measured by the IoT devices, to assess aspects of the environment, examples of which include identifying a person who might be within the environment, quantifying movement of an object within the environment, or detecting a manufacturing anomaly within the environment.

Currently, Internet-of-Things (IoT) allows vendors and end users to deploy pretrained models in their products. A centralized computing system, such as a cloud-computing server or a network server performs the model training. However, the use of such centralized computing system with the IoT devices has multiple drawbacks. As one example, extensive use of computing power of the centralized computing system may circumvent other, higher-priority operations or computations the computing system may be tasked to perform. As another example, the end user might have privacy concerns. And, as yet another example, there may be no available cloud-computing server to perform the model training; whether a biologist performing a field-study in a remote area of North America, or a farmer, or rancher monitoring the crop, or herd, in an emerging market. In summary, utilizing the computing power, albeit limited, of a system with IoT devices to perform model training can be beneficial.

Description:

Current IoT platforms already support the exportation of application programming interfaces (APIs) that allow vendors and end users to use pretrained models on IoT products. However, the training of the model itself is not done on IoT products due to the insufficient computing power on a single IoT device. Given that transfer learning significantly reduces the task of training a model, it becomes feasible to retrain the model in a cluster of IoT products connected to each-other in a LAN, PAN, or WPAN, to be reused for a similar purpose.

To achieve model retraining on IoT embedded devices, the IoT platform is enhanced by exporting the following additional APIs:

a) APIs that specify a small portion of the existing pretrained model to be retrained. These APIs, exported on embedded devices, enable the end user to utilize transfer learning and significantly reduce the portion of the model that needs retraining.

b) APIs that specify a data pipeline in the cluster of IoT devices used to retrain the model. To increase the robustness of this retraining exercise, these APIs determine whether any single IoT embedded device fails and still use the rest of the IoT devices to retrain the model.

c) APIs to tune the model. These APIs enable machine learning by considering alternate values for the training parameters to be performed on the embedded devices.

These APIs in the enhanced IoT platform retrain the model in a cluster of devices. Fig. 3 illustrates how such an IoT platform could be used in the market.

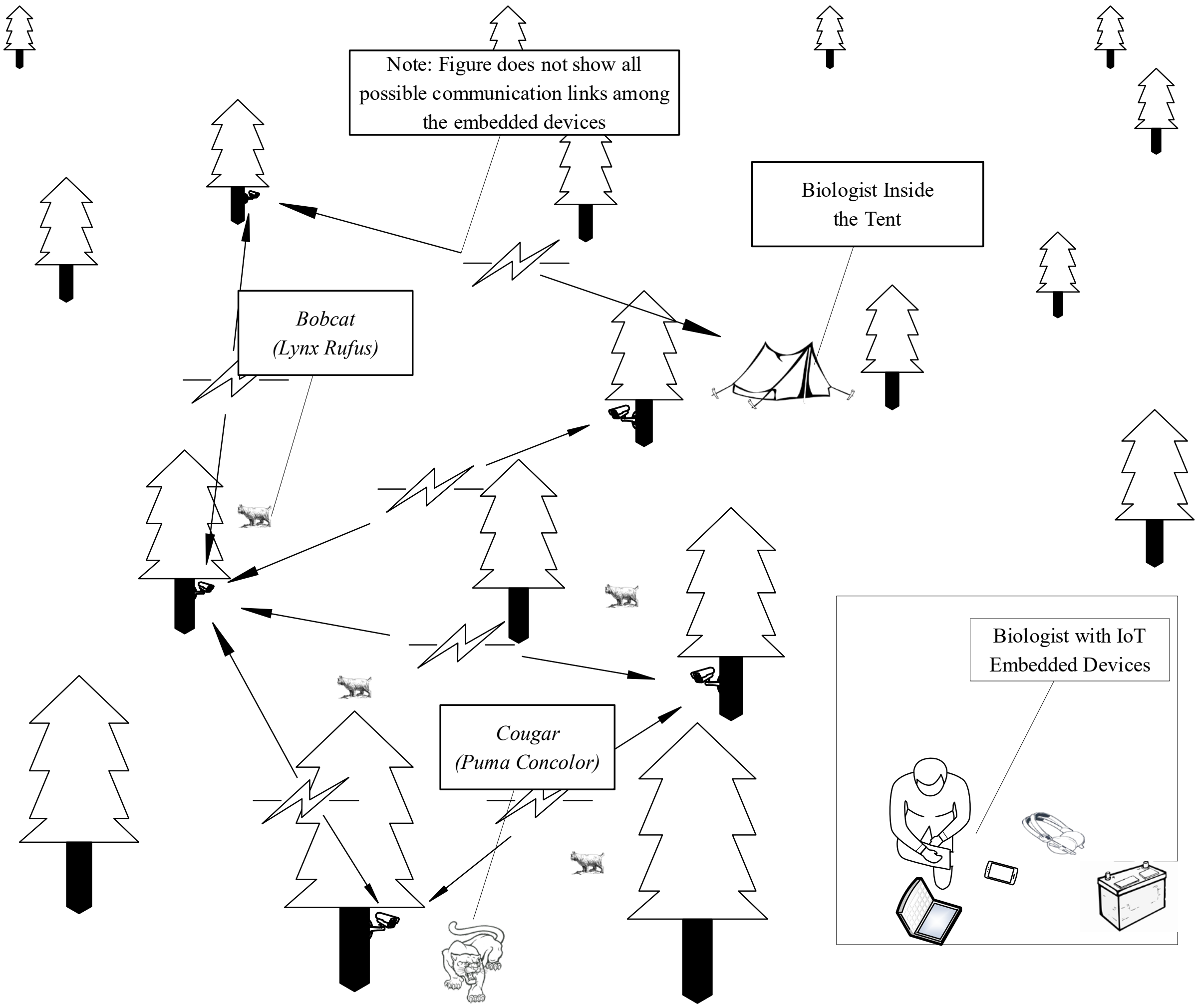
Note: Figure does not show all possible communication links among the embedded devices
Biologist Inside the Tent
Bobcat (Lynx Rufus)
Biologist with IoT Embedded Devices
Cougar (Puma Concolor)

Fig. 3

In Fig. 3, consider the biologist, who for years has visited a specific remote area in North America to study and monitor the bobcat *(lynx rufus)* population. The only way to reach the area is by hiking or by helicopter. The biologist uses a pretrained model to identify bobcats. Now consider that during a scheduled trip, the biologist notices something unexpected; besides the normal sightings of bobcats, there have been sightings of cougars *(puma concolor)* in an area where cougars had not been seen for decades. The area has no available cloud-computing support. The enhanced new IoT platform, utilizing transfer learning, would allow the end user to retrain the bobcat identifying application to identify cougars by using the computing power of the biologist's embedded devices (e.g., smartphone, digital camera, notebook, smartwatch, high-end headphones, etc.).

As demonstrated by the example in Fig. 3, the distributed transfer learning across the embedded devices allows the end user to avoid the use of cloud-computing services. The privacy of the end user is protected. In addition, the increased utilization of embedded devices for such computations helps slow down the ever-increasing need for the construction of new cloud-computing servers. Furthermore, such an IoT platform benefits end users who live and work in an area without cloud-computing support.