April, 2020
Ekkono Solutions AB

Incremental Learning at the Edge

The traditional big data approach to Machine Learning (ML) for IoT (Internet of Things) is that you upload sensor data from many connected devices to the cloud, process it to find common denominators, and learn one generic machine learning model; E.g. to explain the battery consumption and the remaining range of an electrical bus. The advantage of this type of centralized learning is that the model can generalize from the population of devices and hence learn faster. However, this also entails that the data can explain all variations in the devices and their environment, which fast becomes extremely hard to ensure if you do not have absolute control of repairs and the surrounding environment.

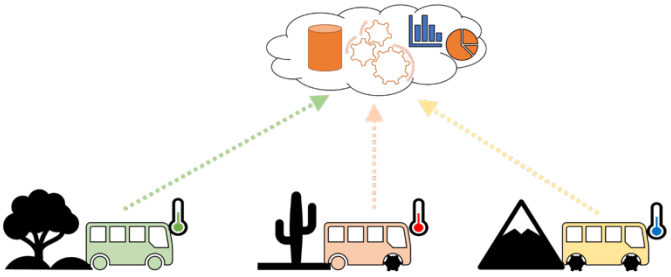

Figure 1 – Traditional Centralized Learning.

Now imagine that you train the model on streaming data to continuously learn more over time as you get more data and insights. Like a Tamagotchi that gets better the more you feed it. By doing this at the edge, i.e. on the device, you can even learn individually per device. Now you can learn the remaining range for the specific bus, based on the climate, landscape, and road and traffic conditions where it is operated. Instant and actionable insights, and you can start with small data. This is the potential of Incremental Learning at the edge.

1. Edge Machine Learning

Ekkono does Edge Machine Learning*. Here Ekkono's unique ability to do incremental learning, i.e. to continuously train the ML model on streaming data, facilitates to learn an individual model for each device. In practice this simplifies the learning task since each model only needs to be able to explain what is normal for itself and not how it varies compared to all

* Edge Machine Learning means running machine learning (ML) at the edge of the network – onboard the connected device. Ekkono develops an Edge Machine Learning software. In Ekkono's case, it is possible to do incremental learning at the edge, which means that the ML model continuously gets better but also that it gets personalized as it is fed with sensor data while in production.

other devices. Other advantages of this decentralized incremental learning are that the models can adapt to changes over time, that they can be applied in real-time since only the processed data (which is often much smaller than the raw data in size) needs to be transmitted to the cloud, thus reducing bandwidth and connectivity related problems. Naturally, these advantages represent a significant value to companies that operate globally in different conditions and environments.

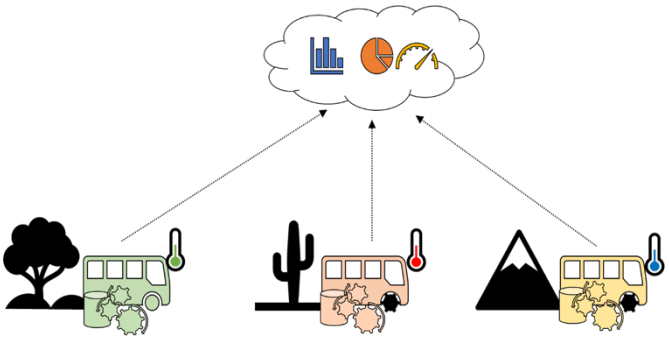

Figure 2 – Edge Learning in different environments, e.g. different temperatures, locations and configurations.

2. Incremental vs Batch Learning[†]

A traditional machine learning approach relies on the fact that data has been collected and is readily available in some kind of data storage. Based on the recorded data, a model is then trained to minimize a score function e.g. the root means square error, for the target variable e.g. fuel consumption. The reason for this approach is that a larger sample of data should be more representative to the true underlying distribution. Hence more data is normally better since it should be closer to the true representation. This also entails that the algorithm can assume that the distribution of data found in the training data also will hold in the future and make a decision upon this fact. However, since the data storage will contain a large amount of data that needs to be processed during training, it will put a high load on the CPU until the training is complete, typically creating a bottleneck. When training is complete, the stored data can be discarded and further execution of the model, i.e. the inference and prediction will be cheap and use data directly from the sensors on the device. In practice a small temporary storage is usually needed to keep a memory of seen instances to facilitate calculation of moving averages and other statistics that can enrich the data and simplify the learning task.

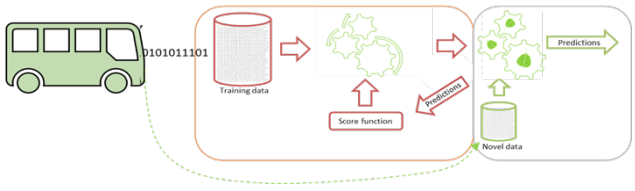

Figure 3 – Batch Learning.

Batch learning has been around since the beginning of machine learning and will continue to be the dominant part of ML. For certain tasks, i.e. tasks where the device or its environment is

[†] Ekkono's Edge Machine Learning software supports both incremental and batch training.

subject to change, an online/incremental approach to training is however more suitable. Example tasks motivating incremental learning are:

- Device performance may be influenced by seasonal related factors e.g. temperature, humidity. Without incremental learning data will need to be collected for at least a year before training can start.
- The device itself may change due to repairs or upgrades. At each change new data must be collected and then retrained.
- IoT devices are often mobile and may hence operate in very different environments. New sites would require more data collection and training of a new model.
- Machine configuration is often optimized for worst case scenario and not for normal operations, which may result in suboptimal performance. If the model is not trained on site with real operating condition it will not perform optimally.

Based on these examples it should be clear that it sometimes may be better to continuously train the model using incremental learning as depicted in Figure 4.

Figure 4 – Incremental Learning.

Incremental learning does not require that a large amount of data has been collected before the model is trained. Instead learning starts with a very simple model typically predicting the average value seen so far. When new data examples arrive, the model is trained to represent more complex patterns to be able to predict new examples. The big difference from batch learning is of course that since we only make use of one or a very small set of examples, we cannot assume that the data is representative for the whole population, i.e. we do not know the underlying distribution of the data. Hence, we cannot be as sure of what is noise from the training data and what is the true underlying signal from the concept we are trying to model, and thereby cannot minimize the error as aggressively as in batch training.

2.1 Learning rate

When designing incremental algorithms, finding a proper learning rate is important. The effect of a too high (too aggressive) learning rate can be seen in Figure 5 where the underlying concept seems to be learned very fast but large deviation remains even after 2,000 instances has been observed. Instead, if a smaller step towards making a good prediction for the new example is taken, it will result in a slower but more general learning rate, which can be seen in Figure 6. If the example is important it will often occur again, and for each time the model sees a similar

example it will become a little better on that type of example. Over time it will learn the recurring patterns and thereby the true underlying concept. In contrast, the downside as compared to batch learning is of course that the learning may take longer since we cannot make assumptions about the underlying distribution. In practice, especially for industrial applications, the process cycles that need to be learnt are often rather short and changing frequently, which makes a slightly longer training time insignificant. The advantages of incremental learning are however large:

- No data needs to be stored or sent to the cloud since it is no longer needed after it has been used to update the model.
- Training is distributed over time, which removes CPU bottlenecks from training.
- The model may adapt to changing configurations or environment.

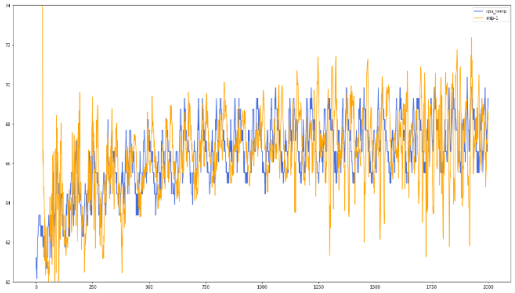

Figure 5 – Too fast learning rate.

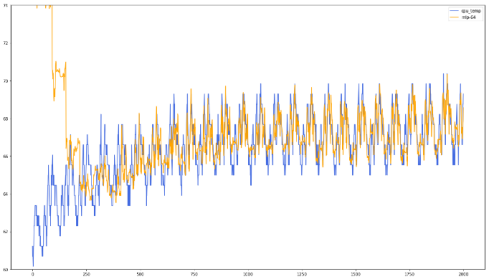

Figure 6 – Appropriate learning rate.

Similar to life, there are no free lunches: Machine learning models that are easy (fast) to train often suffer from the problem of too large memory footprints. This phenomenon can be very well illustrated by the contrast between RF (Random Forest) and MLP (Multilayer Perceptron).

- RFs are fast to train and easy to configure, but memory expensive (MB)
- MLPs are slower to train and hard to configure, but have a small footprint (kB)

3. Conclusion

Ekkono's Edge Machine Learning solves the overarching issue of unmanageable data volumes being generated from IoT sensors. While you want to analyze high-frequency sensor data on a millisecond level, it is unfeasible to upload such data volumes to the cloud; E.g. a modern ABB electrical substation can generate 5GB/s, and most of it loses its value quickly. Edge Machine Learning processes this high-frequency data, to generate a machine learning model that represents the individual learnings from the asset. The ML model is a fraction in size (a few kB) compared to the raw sensor data and represents the actual learnings from all the granular data.

Incremental Learning at the edge comes with several benefits:

- Learn individually per device/asset – what is normal for the specific device, and predict or confirm when something deviates from normal/expected
- You can *learn at work*, continuously, after deployment – Tamagotchi-style
- You can start with little or no data
- You can send the ML model to the cloud, which is a representation of the sensors data, but a fraction in size
- Train the ML model on real-time, high-frequency sensor data – valuable for automation
- Reduce the amount of data stored both in the cloud and at the edge, which saves CPU cycles and battery through less communication and read/write
- Training is distributed over time, which removes CPU bottlenecks from training
- The model may adapt to changing configurations or environment;
    - Device performance may be influenced by seasonal factors like temperature and humidity – without incremental learning, data will need to be collected for at least a year before training can start
    - The device itself may change due to repairs or upgrades, which requires new data to be collected and for the ML model to be re-trained
    - IoT devices are often mobile and may hence operate in very different environments; Incremental Learning can adapt to a new site or a new rental customer

***

Ekkono #openfika is a short open, online fika[‡] session, hosted by Ekkono, on hot, contemporary and relevant topics, where a 15 minutes presentation is followed by discussion and Q&A. Keep an eye on www.ekkono.ai and LinkedIn for the next #openfika session.

Ekkono Solutions AB is a software company that develops Edge Machine Learning. Our product is the result of seven years of research at the University of Borås, Sweden, and assists product OEMs in different industries to rapidly develop smart features onboard their products, using machine learning to make them self-learning and predictive. For more information, visit www.ekkono.ai.

[‡] fika (wikipedia.org); Swedes have fika (pronounced [ˈfiːka]), meaning "coffee break". The tradition has spread throughout Swedish businesses around the world. Fika is a social institution in Sweden and a common practice at workplaces in Sweden. Fika may also function partially as an informal meeting between co-workers and management people, and it may even be considered impolite not to join in.