CSS545: Mobile Computing

Marina Rosenwald

Homework 4

Due: 5/19/2024

Machine Learning: Hardware Constraints

Industry Trends and Needs/Current Solutions

In mobile computing, machine learning is becoming a growing tool in applications and built-in functions. Built-in functions such as voice assistants like Siri and Google Assistant use machine learning concepts like voice recognition, natural language processing, and contextual understanding. Also, modern day biometric authentication such as Face ID and fingerprint scanner are more examples of everyday built-in functions in mobile devices. Biometric authentication adds convenience to the user and an added layer of security. Photo and video processing applications implement object detection and photo enhancement editing. These machine learning features are becoming increasingly popular with users making it crucial for apps such as Snapchat and Instagram to implement them. Another example of machine learning in mobile applications is in translation apps. Machine learning is utilized in speech to text recognition giving real-time transcription services. This furthers human connection by allowing more people around the world to communicate with one another, thus also enhancing accessibility.

The above examples give evidence of how machine learning is used frequently every day by mobile device users. The added accessibility and convenience comes at a price for the device that it is our job as developers to help mitigate and address as we push forward into the future with Machine Learning. This paper will address the resource constraints that make utilizing machine learning on mobile devices more complex than in a typical computing environment. The three main resource constraints mentioned in this paper are processing power, memory and battery life.

Processing power is the ability the device has to process data or perform a series of operations on information. In machine learning, processing power is used constantly at a high level making high processing capabilities essential to a small runtime. However, mobile devices have limited processing power compared to the typical computer environment making running machine learning models slow and inefficient. To help mitigate this issue software engineers must optimize their model by performing techniques such as pruning, quantization, and knowledge distillation to make the models more efficient while still maintaining accuracy.

Memory is the amount of information that can be stored on the device while a program is executing. In machine learning, large amounts of memory are utilized when running and building models, oftentimes coming close to or meeting the device’s memory limit. Coming close to or meeting the device’s memory limit will likely cause the device to crash, interrupt other applications running on the device, cause the device to slow significantly, or cause errors on the device. All of the previously mentioned situations are ones we want to avoid as developers. The most common way to best avoid the overuse of memory is to develop a smaller model using the techniques previously discussed. Another approach is to use offloading and caching, offloading is essentially utilizing the cloud to run more computationally intensive tasks on servers with more memory and processing power than having the results sent back to the mobile device. Caching involves having data stored in a temporary location to make the data more readily available (i.e. the computer doesn’t have to retrieve the data or do the same computation to create it over and over again). In machine learning on mobile devices, caching is often used to store intermediate machine learning results to avoid any unnecessary recalculations.

Battery life is how long the mobile device can operate before needing to be recharged. Machine learning drains battery life quickly on mobile devices due to the large amount of complex computations. This negatively affects the usage of the device as users are not always able to charge their devices and oftentimes will have consecutive hours when they need access to the device and no ability to charge it. To help mitigate machine learning’s battery consumption on mobile devices developers have implemented energy-aware scheduling. This includes techniques such as task scheduling, having machine learning tasks run when the device is plugged in and charging with low device activity, and background processing limits, having machine learning processes reduce in frequency and duration. Both techniques significantly decrease the battery consumption of machine learning processes, thus assisting in the increase in the device's battery life on a single charge.

Critical Analysis

The solutions mentioned above are optimizing the models used in machine learning tasks, using offloading and caching, and implementing energy-aware scheduling. Some pros and cons of optimizing machine learning models are that the time in which the model takes to return results decreases, less hardware resources will be consumed (i.e. less memory, battery, and processing power will be needed to run the model), and the model will be easier to deploy making it more scalable. Some drawbacks of optimizing the models are that there is some loss in accuracy, it takes longer to train as data points have to be cherry picked and tested for the best model performance, and due to the added complexity that comes with smaller models, their decision making process is harder to comprehend.

Positives of offloading and caching include it decreases the need for local hardware as large computational tasks are performed in the cloud, lower retrieval times as important data points are stored in temporary memory that is easy to access and it makes scaling the models a lot easier as all tasks are done in high performance computing environments that are well-suited for large amounts of data. Drawbacks include, the model relies on the devices ability to connect to the same network the cloud services are on, caching needs to be done efficiently, or it risks consuming much of the device’s resources (memory), and when using cloud services to run the model over user data the user’s data is exposed to privacy risks.

Pros and cons of energy-aware scheduling are as follows: Pros - the model’s implementation becomes more user-friendly as it does not drain the user’s device’s battery, other applications on the device have space and battery capacity to run, and the model’s performance is optimized by having the optimal amount of resources available to it when it does run. Cons include - the model’s accuracy may decrease due to a lower use of the device’s processing units, the integration of the scheduling algorithm is complex and may cause unforeseen problems in the future, and there are challenges that arise in how to accurately allocate resources without having under or over consumption issues.

New Solutions

Other solutions to the resource constraints in regards to machine learning on mobile devices are to improve battery technology so batteries last longer. Longer lasting batteries help elevate the battery life constraint and allow for models to run at larger capacities without causing the device to die when the user needs it. Another solution could be to install GPUs with more processing units in mobile devices. This would give the device more processing power thus allowing for larger models to be run locally on the device. Finally, we can increase the memory on mobile devices allowing more space for data on the device locally alleviating the need for remote servers and cutting operation costs on machine learning implementations.

Citations

1. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/4/html/introduction_to_system_administration/s1-bandwidth-processing#:~:text=3.2.-,Processing%20Power,CPUs%20supporting%20smaller%20word%20sizes.
2. https://us.norton.com/blog/how-to/how-to-free-up-ram#:~:text=That's%20because%20full%20RAM%20usage,extremely%20frustrating%20and%20time%2Dconsuming.

3. https://ieeexplore.ieee.org/document/9498725
4. https://par.nsf.gov/servlets/purl/10199522
5. https://developer.android.com/docs/quality-guidelines/build-for-billions/battery-consumption
6. https://blog.linuxplumbersconf.org/2016/ocw/system/presentations/3693/original/LPC-%20EAS%20for%20Android.pdf
7. https://www.linkedin.com/pulse/mechanics-pros-cons-machine-learning-optimization-sofia-m%C3%A9ndez/
8. https://www.telusinternational.com/insights/ai-data/article/three-ways-your-smartphone-is-programmed-to-use-ai-and-machine-learning
9. https://medium.com/@dmennis/introduction-to-machine-learning-on-mobile-36845619c56
10. https://docs.snap.com/lens-studio/references/guides/lens-features/machine-learning/ml-overview
11. https://link.springer.com/article/10.1007/s10619-018-7231-7
12. https://ieeexplore.ieee.org/abstract/document/9344830