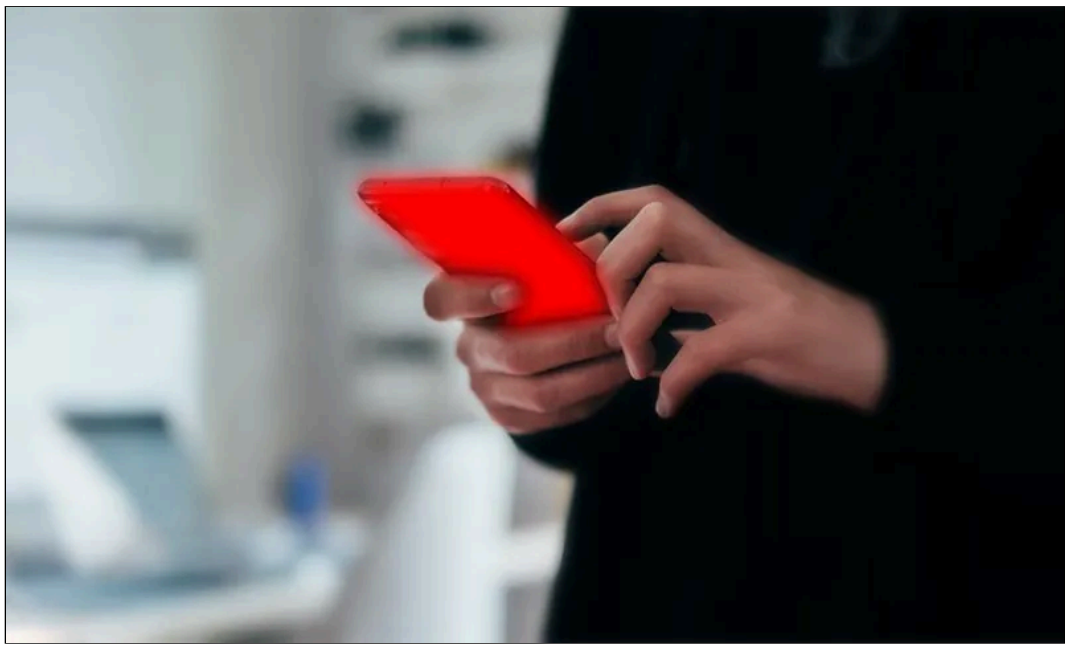
How to Keep Smartphones Cool When They're Running Machine Learning Models

Originally published at https://www.unite.ai/how-to-keep-smartphones-cool-when-theyre-running-machine-learning-models/

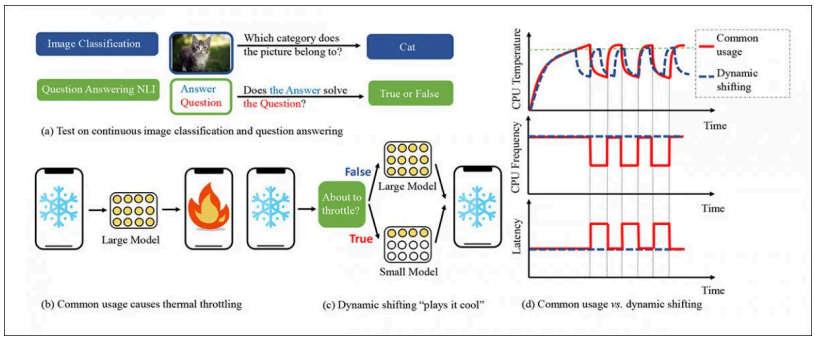


Researchers from the University of Austin and Carnegie Mellon have proposed a new way to run computationally expensive machine learning models on mobile devices such as smartphones, and on lower-powered edge devices, without triggering thermal throttling – a common protective mechanism in professional and consumer devices, designed to lower the temperature of the host device by slowing down its performance, until acceptable operating temperatures are obtained again.

The new approach could help more complex ML models to run inference and various other types of task without threating the stability of, for instance, the host smartphone.

The central idea is to use *dynamic networks*, where the weights of a model can be accessed by both a 'low pressure' and 'full intensity' version of the local machine learning model.

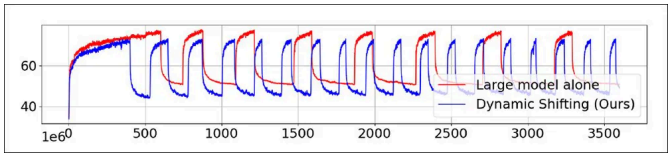
In cases where the operation of the local installation of a machine learning model should cause the temperature of the device to rise critically, the model would dynamically switch to a less demanding model until the temperature is stabilized, and then switch back to the full-fledged version.



The test tasks consisted of an image classification job and a question-answering natural language inference (QNLI) task – both the kind of operation likely to enga
Source: https://arxiv.org/pdf/2206.10849.pdf

The researchers conducted proof-of-concept tests for computer vision and Natural Language Processing (NLP) models on a 2019 Honor V30 Pro smartphone, and a Raspberry Pi 4B 4GB.

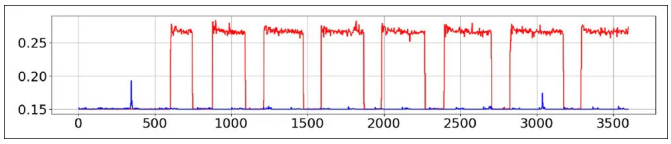
From the results (for the smartphone), we can see in the image below the temperature of the host device rising and falling with usage. The red lines represent a model running *without* Dynamic Shifting.



Though the results may look quite similar, they're not: what's causing the temperature to undulate for the *blue* lines (i.e. using the new paper's method) is the switching back and forth between simpler and more complex model versions. At no point in the operation is thermal throttling ever triggered.

What's causing the temperature to rise and fall in the case of the *red* lines is the automatic engagement of thermal throttling in the device, which slows down the model's operation and raises its latency.

In terms of how usable the model is, we can see in the image below that the latency for the unaided model is significantly higher while it is being thermally throttled:



At the same time, the image above shows almost no variation in latency for the model that's managed by Dynamic Shifting, which remains responsive throughout.

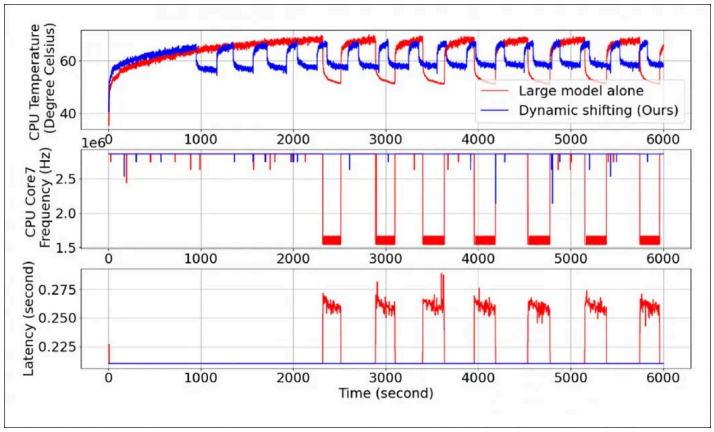
For the end user, high latency can mean increased waiting time, which may cause abandonment of a task and dissatisfaction with the app hosting it.

In the case of NLP (rather than computer vision) systems, high response times can be even more unsettling, since the tasks may rely on prompt response (such as auto-translation, or utilities to help disabled users).

For truly time-critical applications – such as real-time VR/AR – high latency would effectively kill the model's core usefulness.

The researchers state:

> *'We argue that thermal throttling poses a serious threat to mobile ML applications that are latency-critical. For example, during real-time visual rendering for video streaming or gaming, a sudden surge of processing latency per frame will have substantial negative effect on user experience. Also, modern mobile operating systems often provide special services and applications for vision impaired individuals, such as VoiceOver on iOS and TalkBack on Android.*
>
> *'The user typically interacts with mobile phones by relying completely on speech, so the quality of these services is highly dependent on the responsiveness or the latency of the application.'*



Graphs demonstrating the performance of BERT w50 d50 unaided (red), and helped by Dynamic Shifting (blue). Note the evenness of latency in Dynamic Shifting

The paper is titled *Play It Cool: Dynamic Shifting Prevents Thermal Throttling*, and is a collaboration between two researchers from UoA; one from Carnegie Mellon; and one representing both institutions.

CPU-Based Mobile AI

Though Dynamic Shifting and multi-scale architectures are an established and active area of study, most initiatives have concentrated on higher-end arrays of computational devices, and the locus of effort at the current time is divided between intense optimization of local (i.e. device-based) neural networks, usually for the purposes of inference rather than training, and the improvement of dedicated mobile hardware.

The tests performed by the researchers were conducted on CPU rather than GPU chips. Despite growing interest in leveraging local GPU resources in mobile machine learning applications (and even training directly on mobile devices, which could improve the quality of the final model), GPUs typically draw more power, a critical factor in AI's effort to be independent (of cloud services) and useful in a device with limited resources.

Testing Weight Sharing

The networks tested for the project were slimmable networks and DynaBERT, representing, respectively, a computer vision and an NLP-based task.

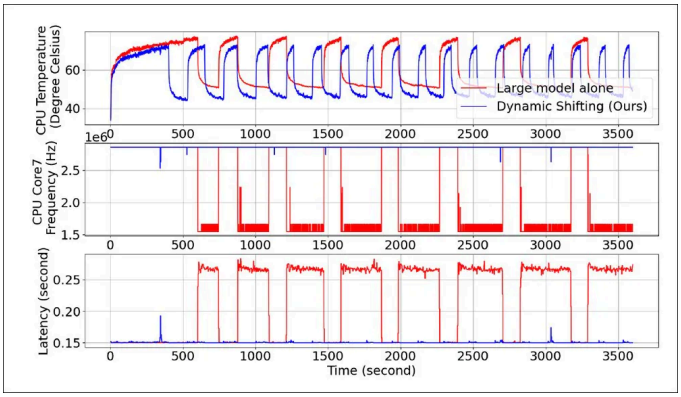
Though there have been various initiatives to make iterations of BERT that can run efficiently and economically on mobile devices, some of the attempts have been criticized as tortuous workarounds, and the researchers of the new paper note that using BERT in the mobile space is a challenge, and that 'BERT models in general are too computationally intensive for mobile phones'.

DynaBERT is a Chinese initiative to optimize Google's powerful NLP/NLU framework into the context of a resource-starved environment; but even this implementation of BERT, the researchers found, was very demanding.

Nonetheless, on both the smartphone and the Raspberry PI device, the authors ran two experiments. In the CV experiment, a single, randomly-chosen image was processed continuously and repetitively in ResNet50 as a classification task, and was able to run stably and without invoking thermal throttling for the entire hour of the experiment's runtime.

The paper states:

> *'Although it may sacrifice some accuracy, the proposed Dynamic Shifting has a faster inference speed. Most importantly, our Dynamic Shifting approach enjoys a consistent inference.'*



Running ResNet50 unaided and with Dynamic Shifting between Slimmable ResNet50 x1.0 and the x0.25 version on a continuous image classification task, for sixty minutes.

For the NLP tests, the authors set the experiment to shift between the two smallest models in the DynaBERT suite, but found that at 1.4X latency, BERT throttles at around 70°. They therefore set the down-shift to occur when the operating temperature reached 65°.

The BERT experiment involved letting the installation run inference continuously on a question/answer pair from GLUE's ONLI dataset.

The latency and accuracy trade-offs were more severe with the ambitious BERT task than for the computer vision implementation, and accuracy came at the expense of a more severe need to control the device temperature, in order to avoid throttling:

	Latency (s)	Accuracy
ResNet50 1.0x (alone)	0.205	0.761
Slimmable ResNet50 (DS)	0.150 (-26.8%)	0.695 (-0.066)
BERT d0.5 w0.5 (alone + 1.4x latency)	0.217	0.900
DynaBERT (DS + 1.4x latency)	0.211 (-2.88%)	0.893 (-0.008)

Latency vs accuracy for the researchers' experiments across the two sector tasks.

The authors observe:

> *'Dynamic Shifting, in general, cannot prevent BERT models from thermal throttling because of the model's enormous computational intensity. However, under some limitations, dynamic shifting can still be helpful when deploying BERT models on mobile phones.'*

The authors found that BERT models cause the Honor V30 phone's CPU temperature to rise to 80° in under 32 seconds, and will invoke thermal throttling in under six minutes of activity. Therefore the authors used only half-width BERT models.

The experiments were repeated on the Raspberry PI setup, and the technique was able also in that environment to prevent the triggering of thermal throttling. However, the authors note that the Raspberry PI does not operate under the same extreme thermal constraints as a tightly-packed smartphone, and appear to have added this raft of experiments as a further demonstration of the method's effectiveness in modestly-outfitted processing environments.

First published 23rd June 2022.

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More