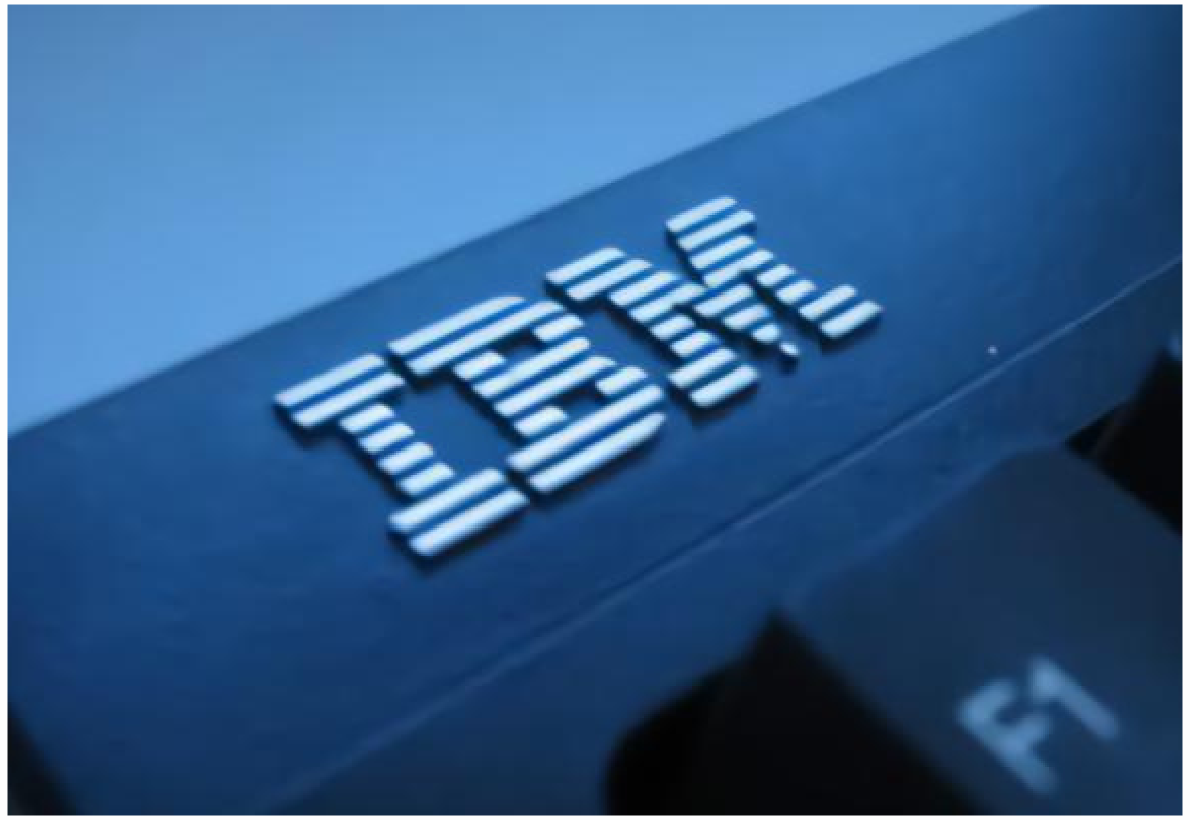
INFERENCE DRIFT ANALYSIS ON SQUEEZENET NEURAL NETWORK USING IBM AIHWKIT

PIERRE TESSIER | SERRANA AGIURREGARAY

High Performance Machine Learning
Columbia University, 2022

EXECUTIVE SUMMARY



Goal

Explore Analog AI's capabilities by evaluating inference drift on Hardware Aware Squeezenet.

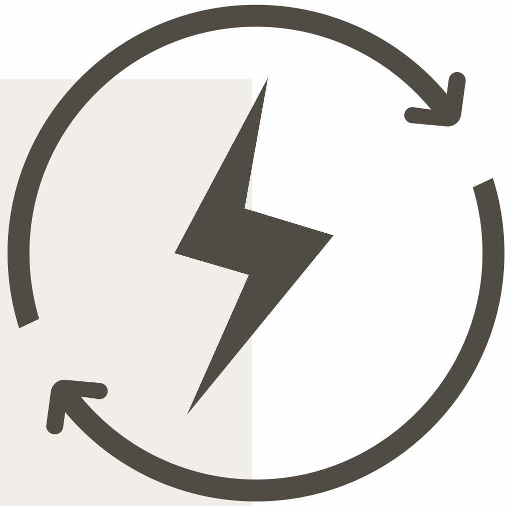
Solution

A digitally pre-trained Squeezenet over Visual Wake Words Dataset was re-trained using AIHWKIT.
Later, a parameter sweep was performed over RPU configurations to evaluate inference drift.

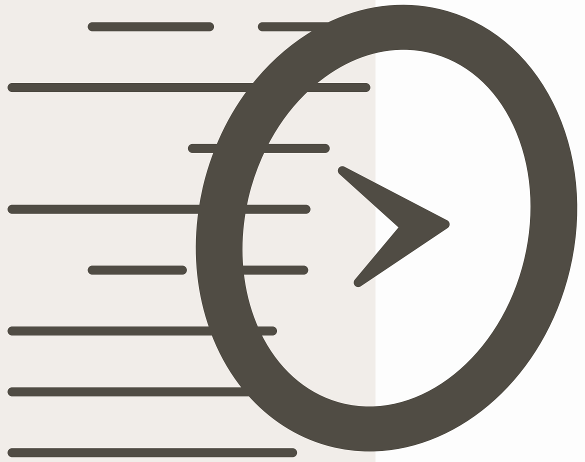
Value

Contribute to experimentation and documentation on IBM's Analog AI technology.

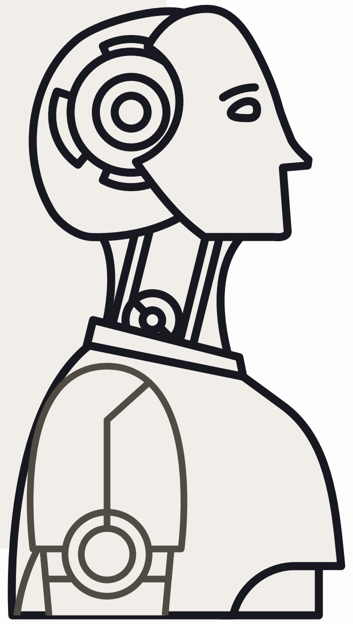
MOTIVATION



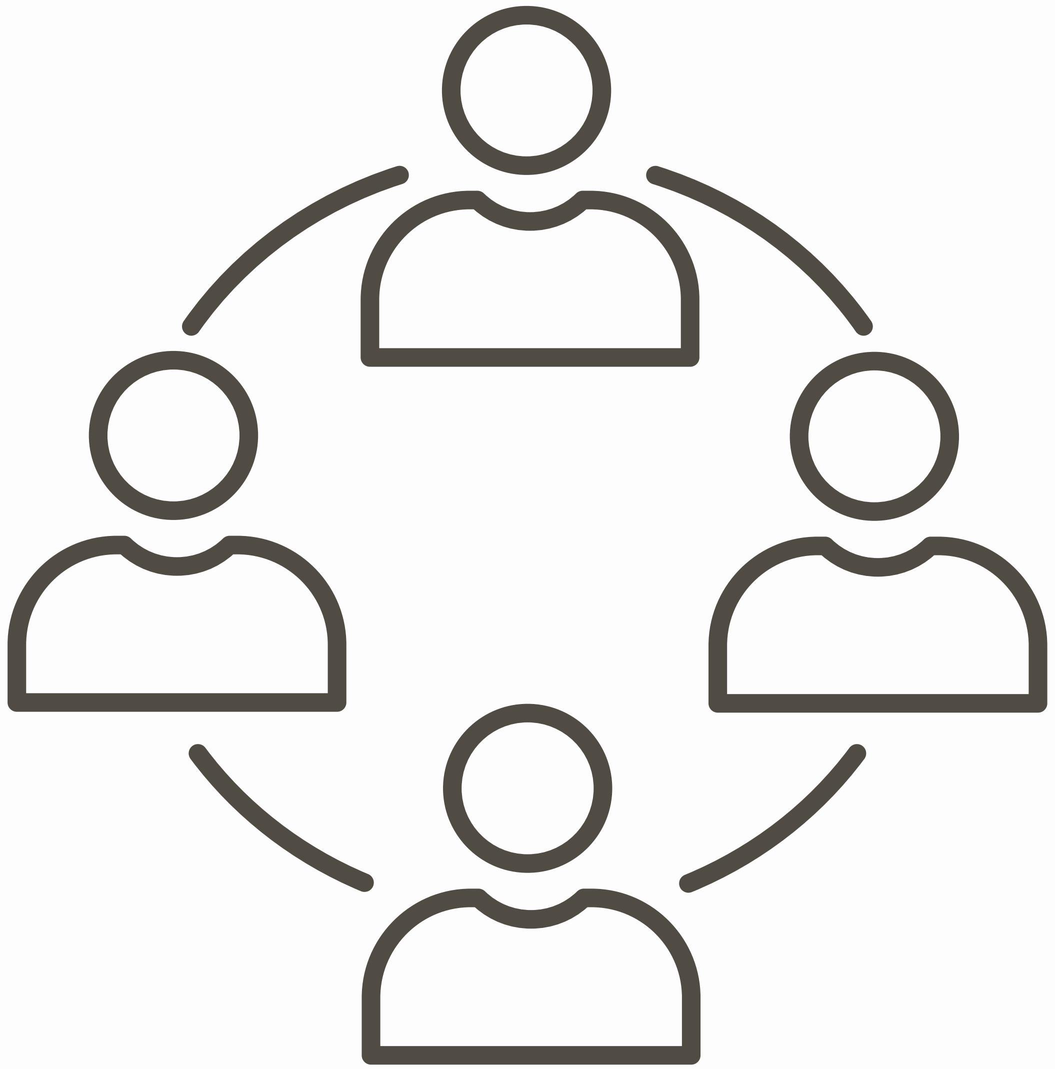
One of the biggest challenges in current Neural Networks is **energy consumption**.



Conventional computers perform calculations by **transferring data** between memory and processor, which **takes time and energy**.



In **Analog AI**, computation happens at the same place where the data is stored, this allows us to have the **speed and energy-efficiency** required to **move AI forward.**



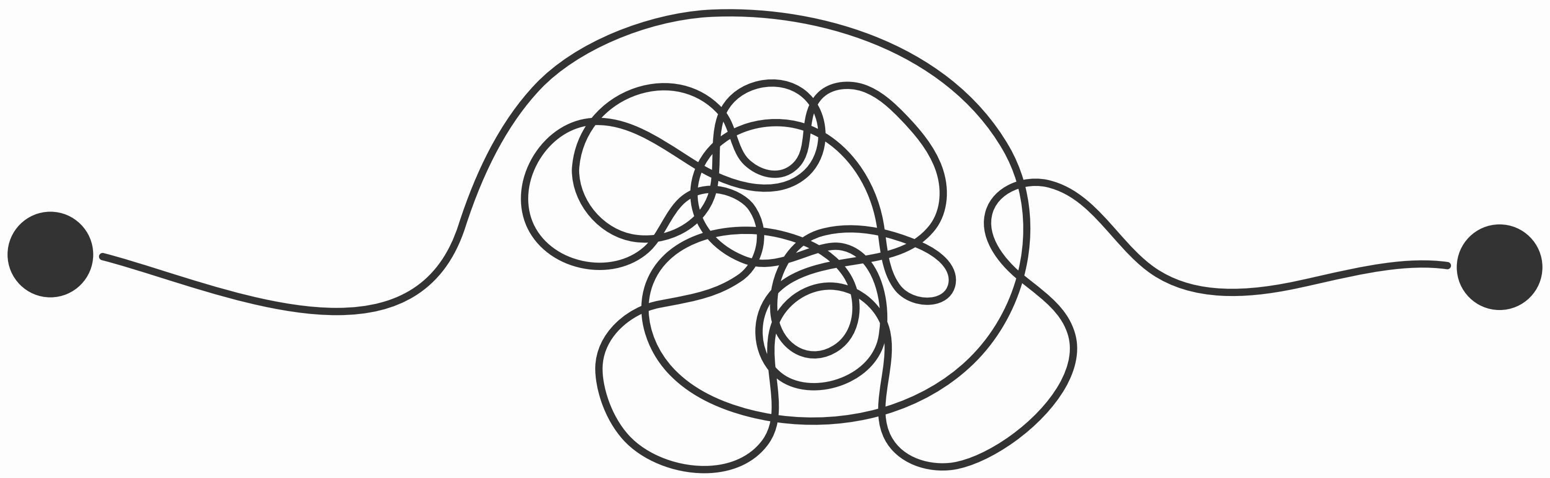
BACKGROUND WORK

AIHWKIT
IBM Analog Hardware Acceleration Kit is an **open source** Python **toolkit** for **exploring** and using the capabilities of **in-memory computing** devices in the context of artificial intelligence.

Squeezenet
Small deep neural network, it has **AlexNet-level accuracy** with **50x fewer parameters** and a **0.5MB model size**, which enables it to easily fit into computer memory.

Visual Wake Words
Adaptation of COCO Dataset that serves as a **benchmark** for the **tiny vision models** to deploy on microcontrollers and advance research in this area.

TECHNICAL CHALLENGES



Model size

The memory requirement to simulate the inference drift was very high.

This caused several models to be tested and modified before finding a suitable one, and computing times to be high.

AIHWKIT

Due to its on-going experimental status, documentation and usability are challenges on its own.

This caused changes in the initial plan due to some features not being implemented.

Hardware setup

Multiple hardware setups and GPU configurations were tested to obtain the correct one.

Due to the dataset and model size, it had to be stored and computed in the Cloud.

APPROACH

Build Visual Wake Words Dataset

By following the steps to repurpose COCO dataset.

Load VWW to Google Cloud

Test multiple work environments

Test configurations and versions to ensure a correct AIHWKIT and model compatibility.

Setup work environment

Setup VM in Google Cloud with an A100 GPU, 12 vCPUs and 85GB memory.
Install requirements with correct versions.

Pre-train digital Squeezenet model

Initialize with pre-trained features.

Design custom classifier with single output.

Add Sequential layer with features removing the last 4 layers.

Initialize weights of custom classifier.

APPROACH

Re-train Squeezenet to be Hardware aware with AIHWKIT

Transform model into analog model by changing the layers for the Analog layers provided by AIHWKIT.

Re-train for single epoch to simulate experimental analog conditions.

Perform parameter sweep over RPU configurations

Sweep RPU configurations using Weights & Biases.

Evaluate inference drift results

Use Weight & Biases results and Seaborn Python library to visualize and analize inference drift.

IMPLEMENTATION DETAILS

Modified Squeezenet last layer

```
├─Sequential: 1-2                          [1, 1]                --
│    └─Dropout: 2-10                       [1, 256, 13, 13]      --
│    └─Flatten: 2-11                       [1, 43264]            --
│    └─BatchNorm1d: 2-12                   [1, 43264]            86,528
│    └─Linear: 2-13                        [1, 1]                43,265
│    └─Sigmoid: 2-14                       [1, 1]                --
```

Model parameters and size

```
==========================================
Total params: 250,209
Trainable params: 250,209
Non-trainable params: 0
Total mult-adds (M): 163.42
==========================================
Input size (MB): 0.60
Forward/backward pass size (MB): 16.98
Params size (MB): 1.00
Estimated Total Size (MB): 18.59
==========================================
```

EXPERIMENT DESIGN

RPU configurations to test:

weight_noise: Noise applied to weights.
out_noise: Noise applied to outputs.
dac_res: Digital to analog conversion.
adc_res: Analog to digital conversion.

Inference times: 1 hour, 1 day, 1 month, 6 months, 1 year

Tested values:

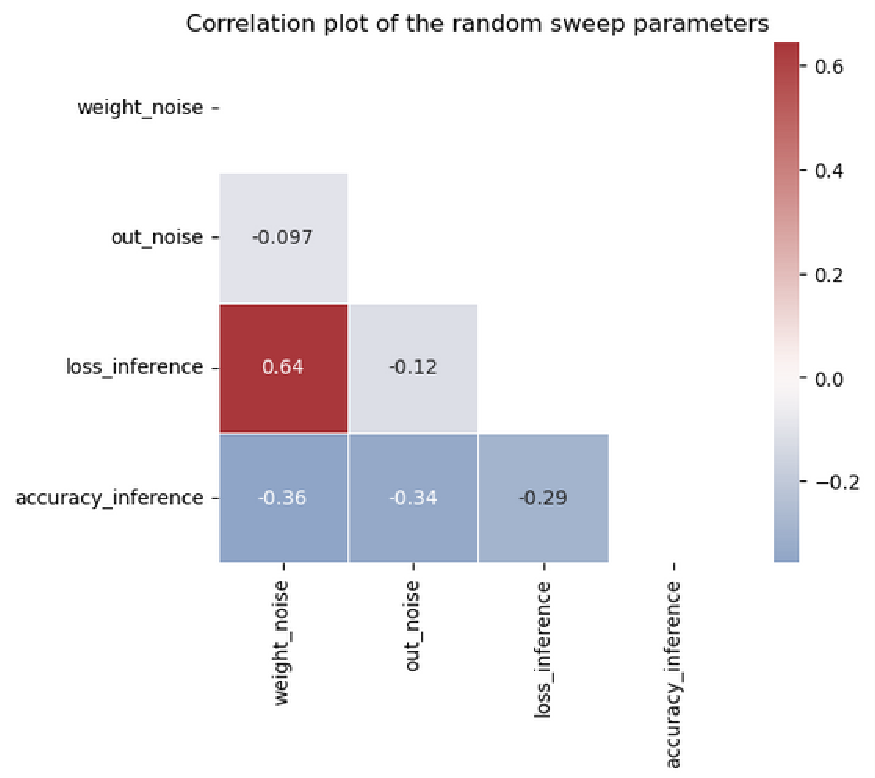
weight_noise: 0, 1e-2, 3e-2, 1e-1, 3e-1
out_noise: 0, 1e-2, 3e-2, 1e-1, 3e-1

Incorrect configuration to test both conversion resolutions, and we ran out of credits by the time we realized it.

EXPERIMENT RESULTS

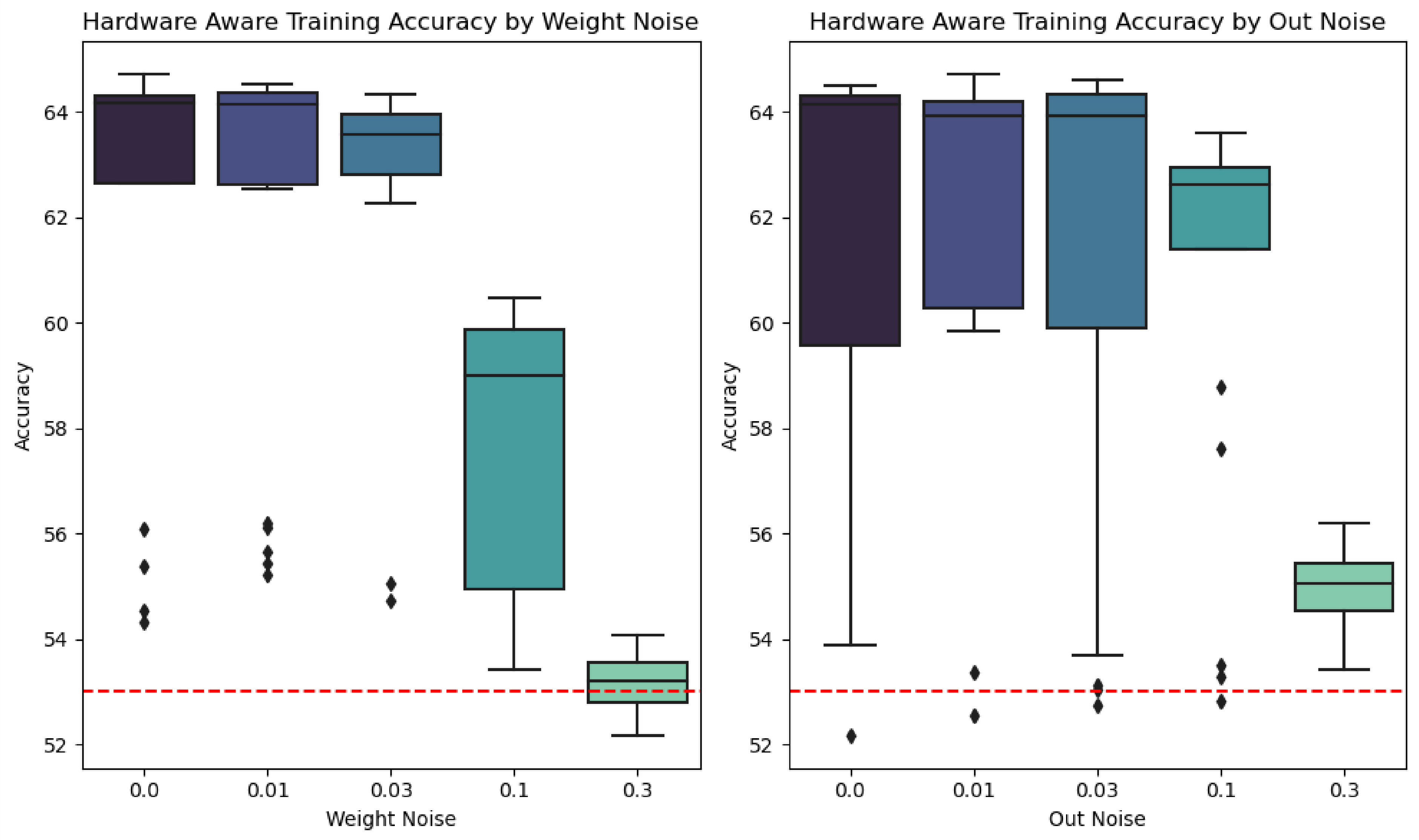
Random Sweep with W&B

- 97 runs with random configuration selection
- 1 epoch of hardware-aware fine-tuning each
- Great parameter distribution, except for the ADC resolution



Hardware-aware fine-tuning results

- Most important parameter: Weight Noise
- Output Noise is less significant for low values
- Threshold phenomenon for both noises above 0.1



EXPERIMENT RESULTS

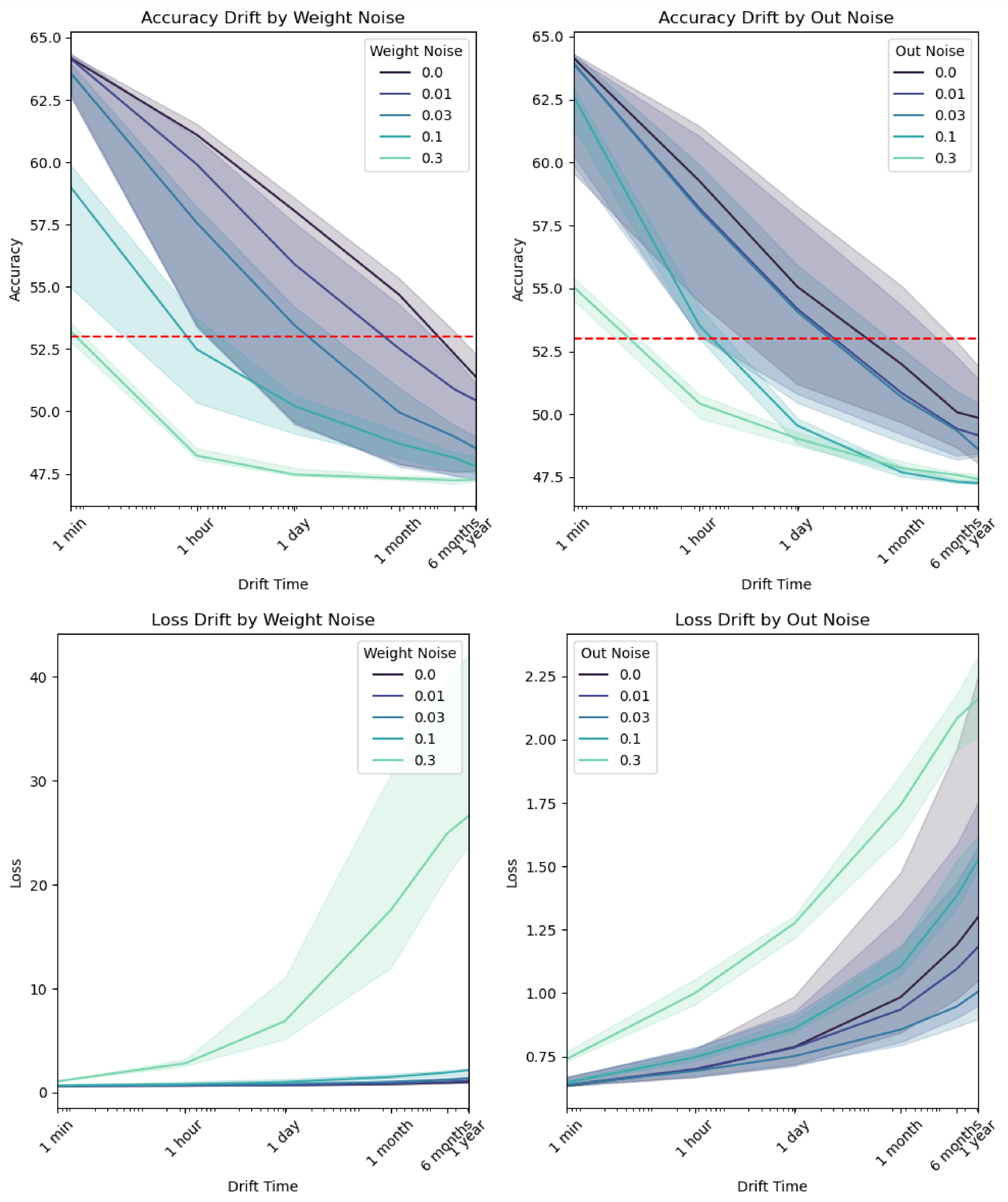
Inference results with drift

Simulate model drift from t=0 to 1 year.

Baseline performance at 53% accuracy, with a constant baseline.

Observations:

- A model always becomes unusable after 1 day ~ 1 month
- A lower weight noise gives a longer model longevity
- An out noise <= 0.03 is required, but further improving it is not significant
- A high weight noise causes the model to diverge.



CONCLUSION

Exciting new architecture

Analog AI is a promising lead and opens the door to many new research axes. IBM's toolkit is a great tool to experiment with virtual analog models.

Model size limitations

Simulating a model's drift requires an enormous amount of VRAM, even for small models. Thus usable models are barely able to process common vision datasets.

Model features limitations

IBM's toolkit does not support convolution grouping yet. Prevents many size optimizations developed on recent models.

Noise sensitivity

The model accuracy is highly dependant on the weight noise factor. This parameter also greatly affects the model longevity.

Observation of a threshold effect for the output noise factor: as long as this parameter is kept below some threshold, performance is not affected.

The sensitivity to the input and output conversion resolution can be easily tested with our code too.

THANK YOU

QUESTIONS?

Github Repository: https://github.com/saguirregaray/HPML-Columbia-2022