STAC Update:
Machine Learning

Bishop Brock
Head of Research, STAC
bishop.brock@STACresearch.com

Peter Nabicht
President, STAC
peter.nabicht@STACresearch.com

SECURITIES TECHNOLOGY ANALYSIS CENTER

STAC-ML Markets (Training) Benchmark : Underway

- Existing ML training benchmarks are not *specific* to Finance:
  - They typically focus on categorical decisions (e.g., most probable next word)
  - Finance often requires quantitative models (e.g., fair value of a derivative)
- Finance use cases may require training many, many models
  - Historical backtesting may involve models specific to points in time
  - This becomes a scale-out problem vs. scale-up (e.g., LLM training)
- Many use cases have been proposed and discussed, but may not satisfy all high-level requirements:
  - Is this an ongoing concern for many end-users?
  - Can performance and quality be reliably measured and compared?
  - Can we validate that the implementation conforms to the specifications?

Some ML Training Use Cases Being Considered

Model Type / Use case	Issues / Notes
Predict prices/returns/portfolio-weights from market data	• Obviously interesting use cases • Training / re-training very important ! Low signal-noise means models learn quickly and erratically – difficult to benchmark
Complex multi-dimensional functions (Derivative valuation, Model Calibration PDE solving)	• Also sees much current interest ! Not clear if training is the bottleneck for most use cases (train once and done?)
Synthetic market data generation	• Useful research and risk testing tool ! Quality evaluation may be difficult ! Again, not clear training is bottleneck
Reinforcement learning for (hedging, trading, …)	• Under investigation

Training: Tell us what *You* think

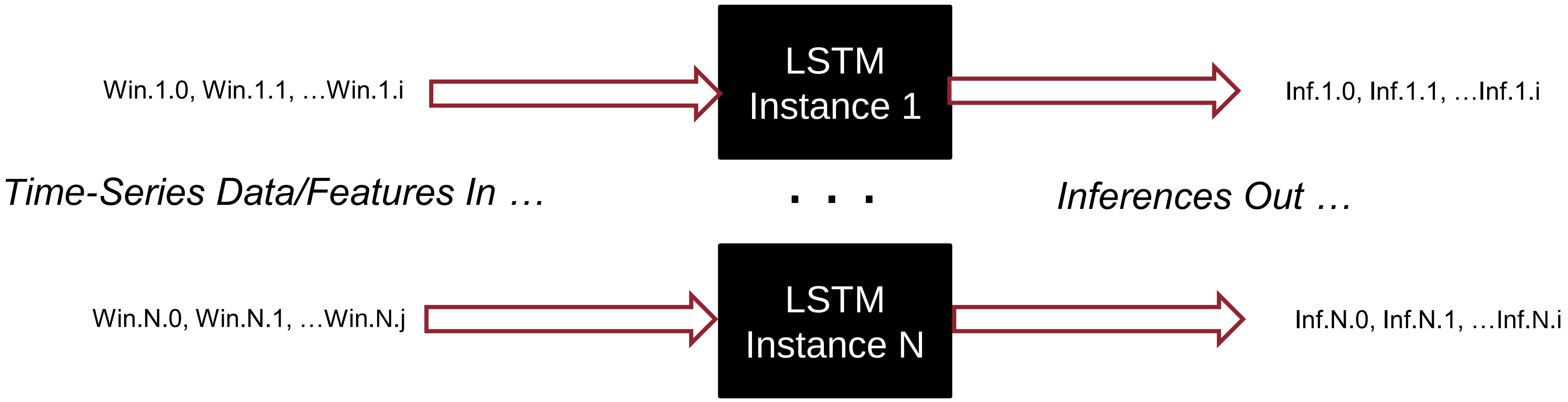
- STAC Benchmarks are defined by financial firms to reflect their needs
- What training workloads give you the insight you need?
- Find us today to talk more, or…
- **Join the Working Group!**

www.STACresearch.com/ML

STAC-ML Markets (Inference) : Basics

- LSTM models inferring on simulated market data features
- Goal: isolate <u>inference</u> performance
  - Inference engine software
  - Underlying processors, memory, accelerators, etc.
  - Anything required to optimally use the former with the latter (e.g., data transfer to processor memory)
- Metrics:
  - Latency, throughput, error, power efficiency, space efficiency, cost
- Benchmarks allow any level of precision (including mixed-precision)

Benchmark Schematic; Scaling Dimensions



- Model size
  - Three are currently specified
  - Input data window scales with model size
- Number of Model Instances running in parallel
  - As specified by the SUT provider
  - Performance / efficiency per model instance is key for co-located inference

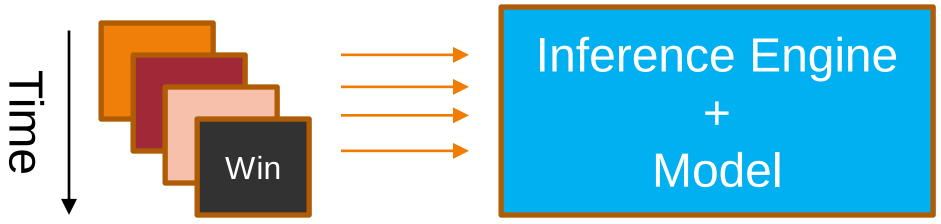
Use Cases and Optimizations

- Different Use Cases:
  - Trading – Latency Optimization
  - Backtesting – Throughput Optimization
- Optimization tradeoffs (latency vs throughput vs efficiency vs error) are up to the SUT provider
  - The benchmarks do not assume an inference application
  - The tests collect all metrics every time, no matter the optimization goal
  - Any quantization scheme allowed, if used consistently

Two benchmark suites

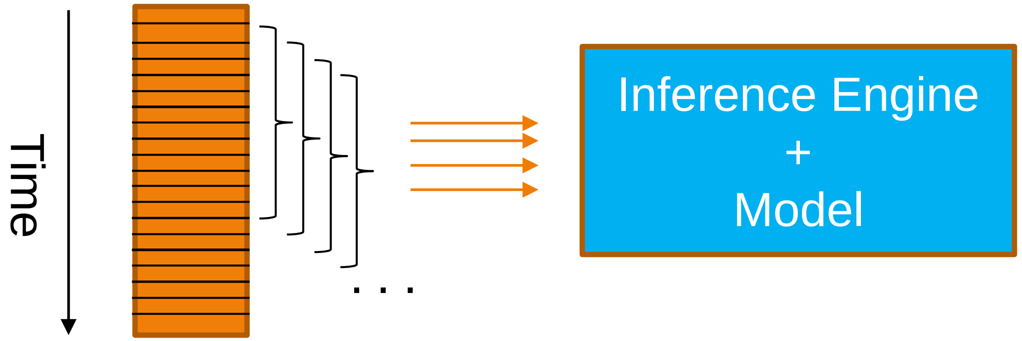
Sumaco

- Operates on fully populated, unique windows of time-series data/features
- Examples:
  - Inference over the recent past in response to an asynchronous event
  - One model may be used to reason about multiple instruments



Tacana

- Operates on sliding windows of a single time-series of data/features
- Example:
  - Inference every tick or bar
- May provide lowest possible tick-to-inference latency



STAC-ML Markets (Inference) - Comparability

- The benchmark is agnostic to the architecture of the SUT and inference engine, and the precision of the computation
- Report readers can explore latency / throughput / error / efficiency tradeoffs
- STAC only allows direct competitive comparisons if all the following are true:
  - Same suite (Tacana to Tacana, or Sumaco to Sumaco)
  - The same LSTM model
  - Error results are comparable
    - SUT A can compare to SUT B if SUT A's error is strictly less than, or only slightly greater than SUT B's
  - All performance comparisons must include an efficiency comparison to provide context
  - All latency comparisons must include a throughput comparison for context

Myrtle.ai tested the Tacana Suite with FPGA as accelerator

Last year did STAC-ML Sumaco (MRTL221125) and now Tacana!

- STAC-ML Pack for Myrtle.ai VOLLO™ (Rev B)
- VOLLO SDK 0.2.0
- VOLLO Accelerator 0.2.0
- Ubuntu Linux 20.04.5 LTS
- BittWare TeraBox™ 1402B (1U)
  - 4 x BittWare IA-840f-0001 each with
    - Intel® Agilex™ AGF027 FPGA
    - 4 x 16 GiB DDR4 @ 2666 MHz
  - 1 x Intel® Xeon® Platinum 8351N CPU @ 2.40 GHz
  - 4 x 8 GiB Micron DDR4 @ 2933 MHz (32GiB total)
- Latency-optimized, bfloat16 precision

www.STACresearch.com/MRTL230426

Results highlights – Myrtle.ai

- *For LSTM_A (the smallest model) the 99p latency was:*[1]
  - *5.07 µs – 5.08 µs Across 1, 2 & 4 model instances tested (NMI)*
  - *5.97 µs with 8 NMI*
  - *6.96 µs with 24 NMI*

- *For LSTM_B the 99p latency was:*[2]
  - *6.89 µs with 1 NMI*
  - *6.77 µs with 2 NMI*
  - *7.75 µs with 8 NMI*

Myrtle.ai

www.STACresearch.com/MRTL230426

1. STAC-ML.Markets.Inf.S.LSTM_A.[1,2,4,8,24].LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_B.[1,2,8].LAT.v1

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Results highlights – Myrtle.ai

- *For LSTM_C (the largest model) the 99p latency was:*[1]
  - *31.0 µs with 1 NMI*

- *LSTM_A with 24 NMI achieved the following throughput and efficiency:*[2]
  - *1.4M inferences / second*
  - *1.4M inferences / second / cubic foot*
  - *2.3M inferences / second / kW*

www.STACresearch.com/MRTL230426

1. STAC-ML.Markets.Inf.S.LSTM_C.[1].LAT.v1
2. STAC-ML.Markets.Inf.S.LSTM_A.12.[TPUT,SPACE_EFF,ENERG_EFF].v1