Ads Ranking at Facebook

model 1
model 2 batch 1
model 2 batch 2
model 3
... model k

predictions

ad 1
ad 2
ad 3
... ad n
Parallel execution between model evaluation
Each model runs on a single thread
For each model, there can be multiple batches executing at the same time. In this case, weights are global and shared between threads, but activations are thread local
Model weights are refreshed every few hours. Therefore, activations needs to be released at the end of each inference to avoid running out of memory
Batch size is dynamic
C++ only
Multiple CPU architectures: avx512, avx2
Model Architecture

MLP: Multilayer perceptron (sequence of FC + activation function)

https://ai.facebook.com/blog/dlrm-an-advanced-open-source-deep-learning-recommendation-model/
Ads Ranking Models

**Implementation**
- JIT (not AOT): because models are updated periodically
- Graph runtime does not manage memory
  - weights are shared between threads for the same model
  - activations are shared by instances of all graph runtimes
  - release activation after each iteration to avoid OOM

**Performance**
- Use MKL for FC for simplicity
- 5-10% speedup from fusion
- Runtime overhead eats into speedup
What’s Next

**Relay VM**
- Handles dynamic shapes
- JIT compilation
- Dynamic memory allocation

**Performance**
- Autotuning at scale
- FBGEMM for fp16 and int8
- Embedding lookup