Lightning Quick Performance on Apple M1 with TVM

Phil Mazenett
Head of Field Engineering, OctoML
Agenda

- Recent results on the new M1 chips (posted in our blog)
- How to replicate
- Q&A
Recent results on the new M1 chips
OctoML’s BERT Model Acceleration Proves Apple M1 Pro and Max Chips Make AI Accessible to Everyone

Phil Mazenett  Jared Rosch
Dec. 9, 2021

OctoML, the company behind the leading Machine Learning (ML) Deployment Platform, has recently spearheaded ML acceleration work with Hugging Face’s implementation of BERT to automatically optimize its performance for the “most powerful chips Apple has ever built.” OctoML is able to provide an average 3X speedup over Apple’s TensorFlow plugin by leveraging its expertise in Apache TVM, the open source stack for ML performance and portability. The details of these technical achievements will be presented at TVMConf, the open source ML acceleration conference on December 17th.
Apple's Most Powerful Chips with 3X CPU Performance Speedups with Apache TVM

(lower bars mean faster inference)
Apple's Most Powerful Chips with ~2X GPU Performance Speedups with Apache TVM

(lower bars mean faster inference)

<table>
<thead>
<tr>
<th>Acceleration Engine on Apple M1 Chips</th>
<th>GPU Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TensorFlow: mini M1</td>
<td>25.37 ms</td>
</tr>
<tr>
<td>TensorFlow: MBP16 M1 Pro</td>
<td>18.64 ms</td>
</tr>
<tr>
<td>TensorFlow: MBP16 M1 Max</td>
<td>19.68 ms</td>
</tr>
<tr>
<td>TVM: mini M1</td>
<td>18.28 ms</td>
</tr>
<tr>
<td>TVM: MBP16 M1 Max</td>
<td>12.32 ms</td>
</tr>
<tr>
<td>TVM: MBP16 M1 (FP16)</td>
<td>12.94 ms</td>
</tr>
<tr>
<td>TVM: MBP16 (FP16)</td>
<td>10.88 ms</td>
</tr>
</tbody>
</table>

OctoML
How to replicate
Clone the Repo

Things to Note:

● Miniforge continues to be the best way to run all the necessary packages on M1
● While Python 3.9 is the default version installed at this time, all tests have been done with 3.8
● When using TVM, ensure you are building with METAL and LLVM set to ON and OPENMP set to gnu
● There is a CoreML script you can use to run some benchmarks using the CoreML libraries. The coremltools library is used for these tests
Running Benchmarks

Things to Note:

- In order to achieve the best results (results highlighted in the blog post) you will need to set the TVM_NUM_THREADS environment variable to the number of your Performance cores; 8 in the Pro and Max
- You should not be running any other CPU intensive app at the same time or apps that can limit performance. The results shown were achieved on a vanilla, clean install of MacOS Monterey with a local account (No background iCloud processes running)
Thank you!

Q&A following session

Also feel free to email me at phil@octoml.ai if you have any other questions