1 Introduction

- There is rising interest in using RISC-V to do DL training [3]
- Library optimized for floating-point computations is needed
- Library should follow specifications of popular APIs for DL

2 Existing works

<table>
<thead>
<tr>
<th>Extensions</th>
<th>Pulp-NN</th>
<th>XNNPACK</th>
<th>OneDNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector &quot;V&quot;</td>
<td>xPulp</td>
<td>Vector &quot;V&quot;</td>
<td>Vector &quot;V&quot;</td>
</tr>
<tr>
<td>Subbyte-quantized integer</td>
<td>Floating-point</td>
<td>Floating-point</td>
<td></td>
</tr>
<tr>
<td>Softmax, Pooling, Conv, LSTM, SVd, Relu, Sigmoid</td>
<td>Add, Pooling, Linear, MatMul</td>
<td>Sqr, Sqr, Abs, Neg, HSwish, Clamp</td>
<td>Pooling</td>
</tr>
</tbody>
</table>

3 Target Platform: Snitch [4]

- SDMA: Snitch asynchronous data movement
- SmallFloat [6]: Support of fp8, fp16, fp32, fp64
- FREP: Floating-point repetition
- TCDM: Tightly Coupled Data

4 Optimization example: LayerNorm

```plaintext
for h in range(b * k):
    if h + a < b:
        read src
        shapes: [B, 2, N, 1]
        strides: [B, 2, N, 1]
        b + a
    elif h + b < b:
        read dst
        shapes: [B, 2, N, 1]
        strides: [B, 2, N, 1]
        b + a
    else:
        read src
        shapes: [B, 2, N, 1]
        strides: [B, 2, N, 1]
        b + a
        b

The use of SSR and FREP extensions to optimize LayerNorm performance
```

5 Operations per cycle in the Snitch core

<table>
<thead>
<tr>
<th>Operation</th>
<th>Peak ops/cycle</th>
<th>latency [cycles]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDMUL</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DIVSORT</td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td>COMP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SDMA</td>
<td></td>
<td>60</td>
</tr>
</tbody>
</table>

6 Evaluation

Performance of kernels on Snitch platform.

7 End-to-end model support

References