On the necessity of learning early layers

- Can we exploit physical laws and avoid learning some invariances?
- Do the first layers need to be learned?

Short review of the Scattering Transform

- Cascade of complex wavelets transforms followed by moduli and finally a linear averaging parametrized by its invariance scale \( J \):
  \[ x(u,\cdot) \rightarrow \mathcal{W}_J \rightarrow \mathcal{A}_J \rightarrow S_J x(\frac{\cdot}{J}) \]

- Additive, diffeomorphisms and translation stabilities...

Nice mathematical foundations!

- Characteristics for an RGB input batch for the scale \( J = 2 \):

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Timing (TitanX)</th>
<th>Timing (CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J = 2 )</td>
<td>32x32x3x128</td>
<td>8x8x24x128</td>
<td>0.63s</td>
</tr>
<tr>
<td>( J = 2 )</td>
<td>128x128x3x128</td>
<td>32x32x24x128</td>
<td>0.26s</td>
</tr>
<tr>
<td>( J = 2 )</td>
<td>256x256x3x128</td>
<td>64x64x4x128</td>
<td>0.71s</td>
</tr>
</tbody>
</table>

Shared Local Encoder (SLE) for Imagenet

- \( \text{Scat. + } 1 \times 1 \text{ convolution } \rightarrow \text{Non-overlapping Scat. } + \text{ supervised encoder (FCs) } \)

- Extremely constrained

ResNet and Scattering on CIFAR10 and Imagenet

- **CIFAR10:**
  - Good prior baseline for image classification
  - Input size: 224x224 x 3
  - \( \text{Unsupervised Conv. } \rightarrow \text{WideResNet } \)

- **Imagenet:**
  - 224x224 x 3
  - \( \text{Unsupervised Conv. } \rightarrow \text{WideResNet } \)

Incorporating geometrical invariances limits settings

- We provide a python implementation on GPUs which mixes CuPy and PyTorch.

- Soon: gradients, fast reconstruction, GANs, cache system...

Conclusion

- Scattering does not lose important information and is an initialisation that improves in the context of limited samples.
- No-mixing of spatial information permits to obtain good performances. Can we incorporate more structure?
- Can we incorporate knowledge of groups beyond euclidean ones?

Contacts

- Emails: edouard.oyallon@ens.fr, eugene.belilovsky@inria.fr
- Website + codes: https://edouard.oyallon.github.io/