Getting started with Differentiable Programming in cryo–EM

\[
\frac{dy}{dx}
\]
Oli B. Clarke @OliBClarke

Beautiful structure... but what poor sod had the task of picking 270k particles manually!?? 😳 rdcu.be/KDbX

Bui lab @builab

Replying to @OliBClarke

@Yoshi__Ichikawa lol u can do it

5:45 PM - 7 Apr 2018

Alyazan Albarghash @Aly_Albarghash · Apr 5

Replying to @OliBClarke @kshbeckham

Oh dear ... Oh dear
not a very unusual task though :$

Joshua Lobo @boreas_cryo · Apr 5

Really ? :O Haven't crossed 9k yet
- Noisy
- Irregularly shaped
- Binary decision
Who would win against a machine?

**ImageNet1000**
- 1M+ natural images
- 1000 classes
- 5% human error rate

**This 1 particle boi**
- Noisy
- Irregularly shaped
- Binary decision
Who would win against a machine?

Driving a car

This 1 particle boi

- Noisy
- Irregularly shaped
- Binary decision

- &^%#@!
A graph of differentiable functions
How does a ConvNet work?
Residual networks

\[ F(x) + x \]

 würde

34-layer plain

34-layer residual
Not a black box

What Does the Network See?

Semantic dictionaries give us a fine-grained look at an activation: what does each single neuron detect? Building off this representation, we can also consider an activation vector as a whole. Instead of visualizing individual neurons, we can instead visualize the combination of neurons that fire at a given spatial location. (Concretely, we optimize the image to maximize the dot product of its activations with the original activation vector.)

https://distill.pub/2018/building-blocks/
To start: Don’t start from scratch!
```python
def boxnet_resnet_v2_generator(resnet_size, num_classes, data_format=None):
    num_blocks = (resnet_size - 2) // 6

    def model(inputs, is_training):
        inputs = conv2d_fixed_padding(
            inputs=inputs, filters=128, kernel_size=5, strides=3,
            data_format=data_format
        )
        inputs = tf.identity(inputs, 'initial_conv')

        inputs = block_layer(
            inputs=inputs, filters=16, block_fn=building_block, blocks=num_blocks,
            strides=1, is_training=is_training, name='block_layer1',
            data_format=data_format
        )

        inputs = block_layer(
            inputs=inputs, filters=32, block_fn=building_block, blocks=num_blocks,
            strides=2, is_training=is_training, name='block_layer2',
            data_format=data_format
        )

        inputs = block_layer(
            inputs=inputs, filters=64, block_fn=building_block, blocks=num_blocks,
            strides=2, is_training=is_training, name='block_layer3',
            data_format=data_format
        )

        inputs = batch_norm_relu(inputs, is_training, data_format)
        inputs = tf.layers.average_pooling2d(
            inputs=inputs, pool_size=8, strides=1, padding='VALID',
            data_format=data_format
        )
        inputs = tf.identity(inputs, 'final_avg_pool')
        inputs = tf.reshape(inputs, [-1, 64])
        inputs = tf.layers.dense(inputs=inputs, units=num_classes)
        inputs = tf.identity(inputs, 'final_dense')
        return inputs

    return model
```
```
Code, part 2

45 particle species, 1500–3000 particles each:

- 23 simulated from PDBs with InSilicoTEM
- 22 from EMPIAR & in-house
Augmentation

- Rotate
- Shear
- Noise
Training

• Rescale all images to 8 A/px
• Extract positive and negative examples
• Augment examples
• Look at each example 100 times in random order
• Gradually decrease learning rate from $10^{-3}$ to $10^{-5}$
Inference

- Rescale micrograph to 8 Å/px
- Extract and normalize a running window of $96^2$ px
- Send it through BoxNet
- Store SoftMax results for all positions
- Binarize (typically at > 0.9)
- Find connected components
- Centroids = particle positions
- Optionally, enforce minimum distance
**Integration**

**CTF**
- Window: 768 px
- Voltage: 300 kV
- Amplitude: 0.07
- Defocus: 0.2–8.0 μm
- Use Movie Sum
- C: 2.70 mm
- C: 2.70 mm
- ΔE: 0.70 eV
- Phase Shift
- Model Ice Ring

**Motion**
- Consider 0.02–0.25 Ny, weight with B = -600 Å²

**Models**
- Defocus: 2 x 2 x 1
- Motion: 4 x 4 x 20

**Pick Particles**
- Use BoxNet_20180122
- Expect 140 Å, cryo particles; use scores above 0.95
- Extract 256 px boxes, 1.0600 Å/px, invert, normalize

**Select BoxNet model**
- BoxNet_20180122
- HaukeNet
- SandraNet_2
- SandraNet

**Retrain BoxNet_20180122**
- New name: BoxNet_20180122_2
- Positive examples
- Select per-micrograph STAR files...
- False-positive examples
- Select per-micrograph STAR files...
- Particle diameter is 200 Å
- Also use 100324 examples in C:\Users\dtegno\Desktop\warp\boxnettraining

**Start training**
With EMPIAR-10097
Raw Data

Input:
- Pixel W/H: 0.5300/0.5300 Å, θ: 0.0°
- Bin: 1.00x (1.0600 Å/pix)
- Dose: 0.00 e/A²/frame

Preprocessing:
- Correct gain using: G:\particlerawl\dataset\temp\10078\SuperRef...
- CTF
  - Window: 768 px
  - Range: 0.11-0.75 Ny
  - Use Movie Sum
  - Voltage: 300 kV
  - S: 2.70 mm
  - Cc: 2.70 mm
  - Amplitude: 0.07
  - Ill. Aperture: 30 μrad
  - ΔE: 0.70 eV
- Defocus: 0-8.0 μm
- Phase Shift
- Model Ice Ring

Motion:
- Consider 0.02-0.25 Ny, weight with B = -0.00 Å²

Models:
- Defocus: 2 x 2 x 1
- Motion: 6 x 4 x 20

Pick Particles
- Use BoxNet, 20180122
- Expect 140 Å
- particles; use scores above 0.95
- Extract 256 px boxes, 1.0600 Å/pix
- Invert, normalize

Output:
- Skip first 0, last 0 frames
- Average
- Deconvolved average (strength = 1.00, tailoff = 1.00)
- Aligned stack, collapse every 1 frames

Overview:
- Fourier Space
- Real Space

Processing Status

Astigmatism (use up to 2.0 μm)

Defocus (use 0.35–5.00 μm)

Phase shift (use 0.10–0.70 π)

Estimated resolution (use better than 3.2 Å)

Average motion per frame in first ½ (use up to 1.5 Å)

Number of particles in BoxNet_20180122 – 8762 overall, 6248 good (use at least 1)

START PROCESSING
Things to try

• 3D map denoising
• GANs for realistic data simulation
• Autoencoders in 2D, 3D to deal with flexibility
• Refinement with better scoring metric
• Reconstruction
General challenges

• No training data for most problems
• Memory consumption in 3D
• Very little research applicable to cryo-EM