Unsupervised Visual Representation Learning by Context Prediction

Summary

1. Learn representation in an unsupervised manner using the relative position prediction task.
2. The network may learn trivial shortcut to solve this task, therefore needs tricks to avoid trivial solution.

3. Learning Visual Context Prediction.

1. They aim to learn feature embedding for idv patches, such that visually similar patches would be close in embedding space.

2. To achieve this, they use a late-fusion architecture:
   - A pair of AlexNet-style architecture processes each patch separately.
   - Until a depth analogous to fc6 in AlexNet, after which the representations are fused.

3. To obtain training example given an img, they sample the 1st patch uniformly without reference to img content
   ↳ but it must leave enough space around to obtain the 2nd patch ?

4. Given the position of the 1st patch, they sample the 2nd patch randomly from the 8 possible neighboring locations.

3.1 Avoiding "trivial" solutions

1. Need to avoid shortcut for trivial solution:
   - boundary patterns or texture continuing bt patches
     ⇒ include a gap bt patches (approx. ½ the patch width)
   - long line spanning the patch
     ⇒ randomly jitter each location up to 7 pixels.
   - chromatic aberration (ConvNet can learn to locate the patch relatively to the lens) ↖ don't quite understand this yet.
     ⇒ 2 types of pre-processing was experimented with:
     1. Shift green & magenta towards gray ("called "projection")
     2. Randomly replace 2 of the 3 color ~~cut~~ channel with noise ("color dropping")
     3. They found both to perform similarly.

2. Other implementation details:

1. Randomly downsampling some patches to 100 total pixels, and then upsampling to build robustness to pixelation.
2. Use Batch Norm to allow for successful training.
3. High momentum (.999) accelerates training.

4. Experiments

4.1 Nearest neighbors

1. They use nearest neighbor to demonstrate that the network has learnt to associate similar patches.
2. They:
   - Sampling random patches.
   - Represent the patches using the learnt features.
   - find nearest neighbor using the normalized correlation of the features.
3. They compare against ImageNet features & untrained ConvNet feature.