Neural Nearest Neighbors Networks

Tobias Plötz
Stefan Roth

Motivation
- K-nearest neighbors (KNN) selection is building block for many methods, e.g. non-local image restoration algorithms [1, 3]
- however: KNN selection is not differentiable
- this prevents learning the feature space for matching

We propose a deterministic and differentiable relaxation to KNN and leverage it in a novel non-local processing layer (N³ block).

Contributions
- Unlikely soft attention mechanisms, e.g. [6], retrieve multiple weighted averages, yielding a set of non-local neighbors.
- Go beyond existing differentiable k-nearest neighbors classifiers [2, 5], retrieving a set of non-local neighbors instead of non-local averaging of labels.
- Do not assume fixed set of target neighbors unlike [7].
- Back propagating through the non-local layer enables feature learning in the context of non-local processing [1, 3].

Code available! github.com/visinf/n3net

Step 1 – K-nearest neighbors selection
q: query item
x_i: database items
x_i \rightarrow \text{p}_i: nearest neighbor

\text{For KNN, } X', \text{ given by weighted sum:}
\forall \text{ deterministic and 1-hot coded}

Step 2 – Stochastic k-nearest neighbor selection
Sample u from sequence of discrete distributions:
\text{P[}u_{i+1} = i|u_{i+1} = i] = \exp(a_i(u_i) / \sum_j \exp(a_j(u_i))

First logits correspond to distances to query item:
a_i(u) = -d(x_i, x_q)

Update logits such that same weight cannot be sampled again:
a_{i+1} = a_i + \log(1 - u_i) \rightarrow a_{i+1} = a_i + \log(1 - u_i)

if \ y \neq i \rightarrow a_i = a_i + \log(1 - u_i)

if \ y = i \rightarrow a_i = a_i

Step 3 – Continuous relaxation
Relax discrete u to continuous expectation::
\text{E}[u_{i+1} = i|u_{i+1} = i] = \mathbb{P}[u_{i+1} = i|u_{i+1} = i]

Relax discrete logit update:
a_{i+1} = a_i + \log(1 - u_i) \rightarrow \text{a_i is differentiable}

Use \text{u_i} to define continuous NN: X' = \sum_i w_i x_i

Recover hard KNN as \text{u_i} \rightarrow 1

Continuous nearest neighbors
Differentiable deterministic relaxation of KNN selection

Goal: Calculate nearest neighbor feature volumes

Self-similarity: Queries and database features selected from same features. Predict temperature t

Full differentiability

N²Net
Interleave local network blocks with N³ blocks
Local blocks derived from base architectures, e.g. DnCNN [9]
N²Net can learn to non-linearly aggregate neighbor information

N³Net: Neural Nearest Neighbors Network
Improved accuracy from augmenting local networks with N³ block

Neural nearest neighbor selection
Embedding network
Neural nearest neighbor block
Continuous nearest neighbors selection
Neighborhood feature volumes are concatenated

References

Experiments: Gaussian Denoising
- Baseline architecture: DnCNN (9)
- 3x 6-layer DnCNN interleaved with 2x N³ block (k=7)
- Training on 40000 samples of BSD68 dataset
- Individual model per noise level

Results:
- Continuous nearest neighbors with learned embedding outperforms KNN selection on static embedding
- Significant and consistent gain over DnCNN baseline
- Outperforms other strong local denoisers
- Outperforms competing non-local methods with KNN selection on fixed feature space

Experiments: RAW Image Denoising
- Baseline architecture: DnCNN (6)
- 3x 6-layer DnCNN interleaved with 2x N³ block (k=7)
- Trained on 400 images of BSDS, 800 images of DIV2K, and 3793 images of Waterloo dataset
- Single model for wide range of noise level functions
- Data augmentations to approximately invert camera processing

Results:
- New state of the art on raw image denoising benchmark DND (4)
- Outperforms baseline DnCNN
- Outperforms dedicated sRGB denoising methods

Experiments: Correspondence Classification
- Baseline architecture: Context Normalization Net (8) (CNNet)
- Added one more N³ block (k=7) at middle layer of CNNet
- Input: Set of pairs of image coordinates
- Output: Classification score for each putative correspondence

Results: N³ block significantly improves accuracy across all settings

Acknowledgements.

References