Deep metric learning using Triplet network

Elad Hoffer, Nir Ailon

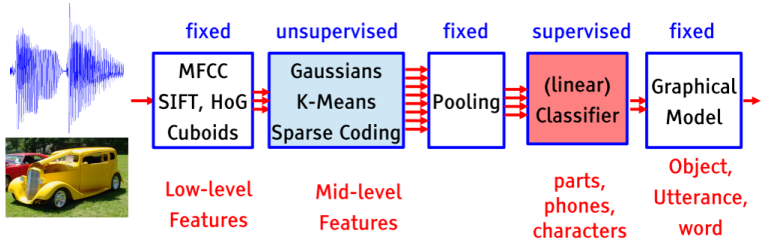
August 2015

Outline

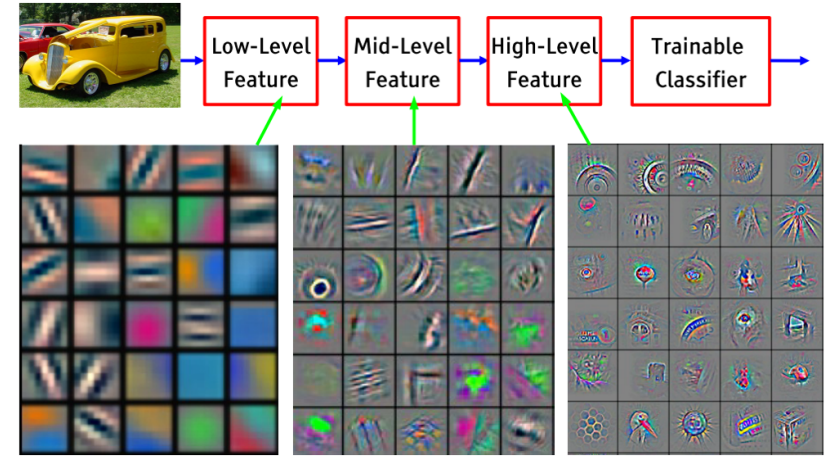
Outline

Deep Learning

Deep learning is a set of machine learning algorithms based on neural networks. These algorithms mostly concern multi-layered hierarchies (hence "Deep") that aim at finding abstract features representations from vast information.



Traditional ML



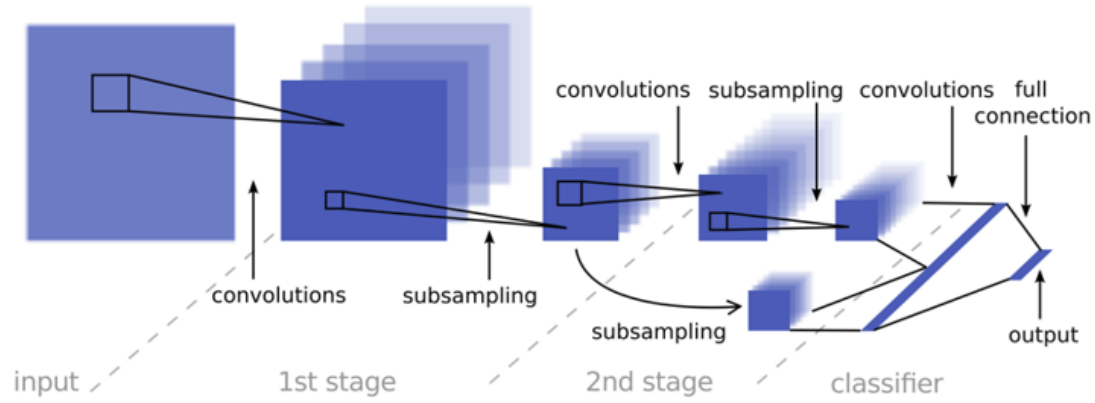
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

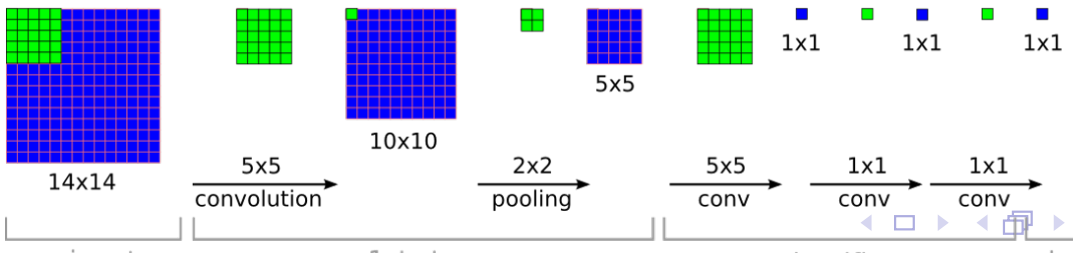
Deep Learning

Convolutional Neural Networks

Deep convolutional neural network represent the main approach of deep learning in computer vision tasks.

- Trainable weight kernels produce their output value by cross-correlating with input data.
- A non-linearity (ReLU, Sigmoid) is applied between layers
- Pooiling layers alow dimension-reduction+invariance

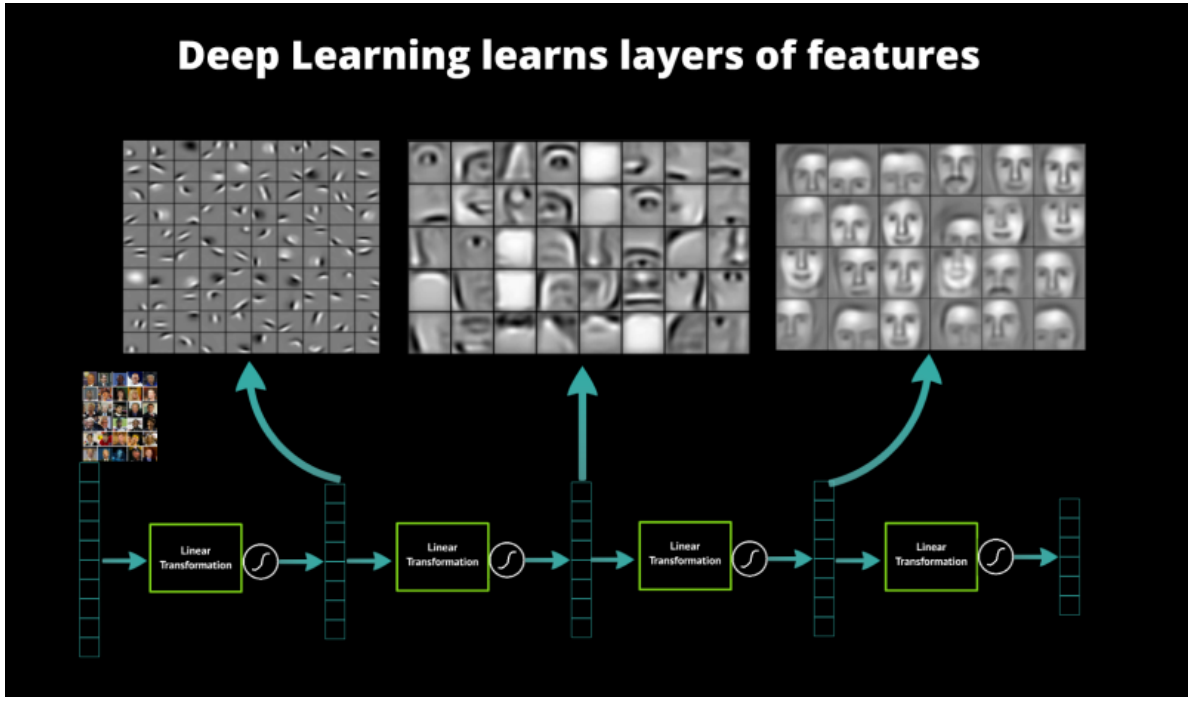




Outline

Deep Metric Learning

- Deep learning has proven itself as a successful set of models for learning useful semantic representations of data.
- These, however, are mostly implicitly learned as part of a classification task.



Deep Metric Learning

In this work, we aim to learn useful representations by distance comparisons.
Accordingly, we try to fit a metric embedding S so that:

$$S(x, x_1) > S(x, x_2), \quad \forall x, x_1, x_2 \in \mathbb{P} \quad \text{for which } r(x, x_1) > r(x, x_2).$$

where $r(x, x')$ is a rough similarity measure given by an oracle.
We focus on finding an L_2 embedding, by learning a function $F(x)$ for which $S(x, x') = \|F(x) - F(x')\|_2$.

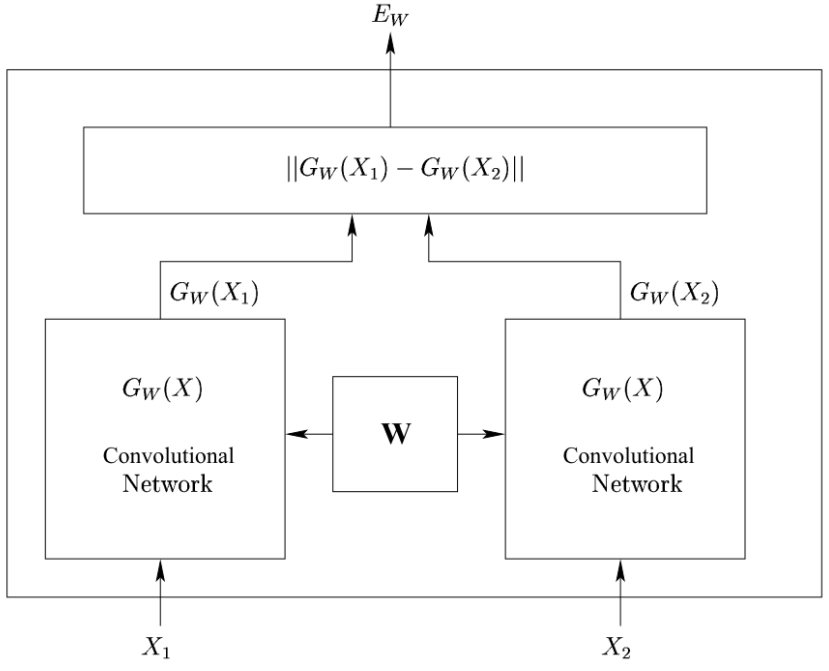
Outline

Siamese network

Siamese networks were used since the 90s to learn feature embedding with neural networks.
Training is done using a *contrastive loss*:

$$Loss(x_1, x_2; y = \{0, 1\}) = y \cdot E_w^2 + (1 - y) \cdot \max(0, \alpha - E_w)^2$$

They exhibit several problems on modern datasets and tasks.



Outline

Triplet network

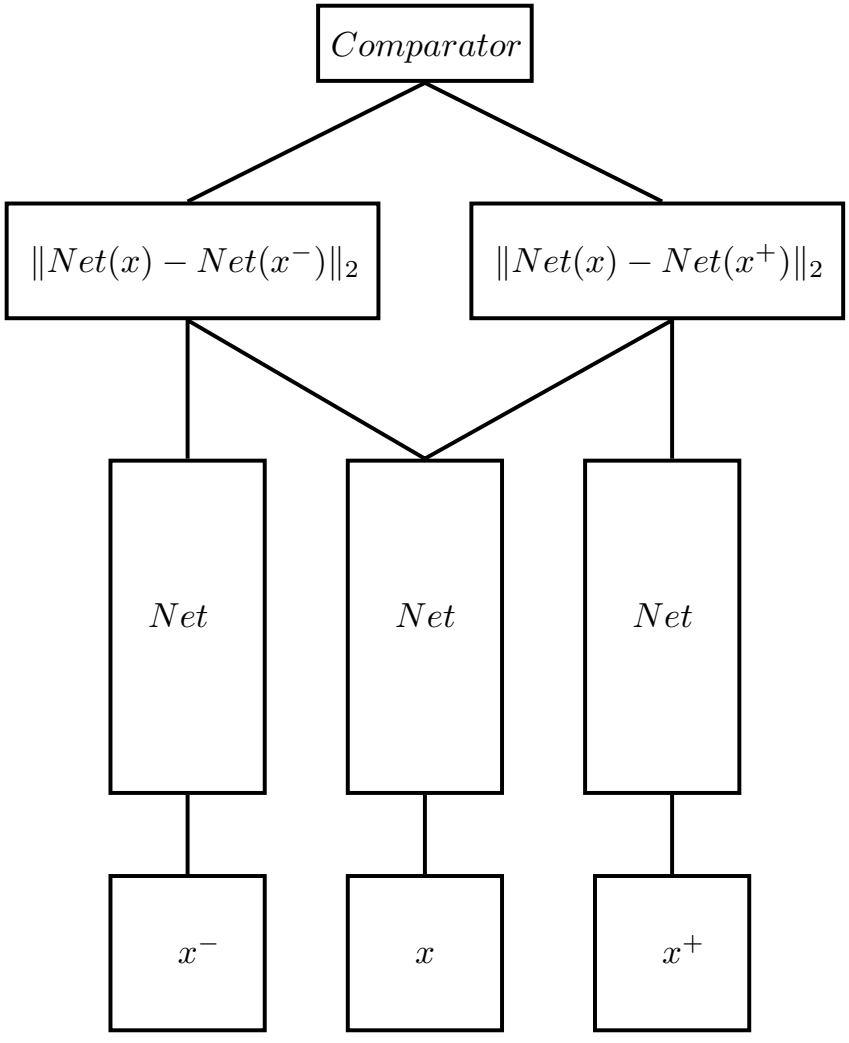
A *Triplet network* is comprised of 3 instances of the same feed-forward network (with shared parameters).

- When fed with 3 samples, the network outputs 2 intermediate values - the L_2 distances between the embedded representation of two of its inputs from the representation of the third.

$$TripletNet(x, x^-, x^+) = \begin{bmatrix} \|Net(x) - Net(x^-)\|_2 \\ \|Net(x) - Net(x^+)\|_2 \end{bmatrix} \in \mathbb{R}_+^2 \ .$$

- In words, this encodes the pair of distances between each of x^+ and x^- against the *reference* x.

Triplet network



Outline

Training the network

- Training is preformed by feeding the network with samples where x and x^+ are of the same class, and x^- is of different class.
- A direct expansion from Siamese network will lead us to

$$L(x, x^{\pm}) = \left[\left\| Net(x) - Net(x^+) \right\|_2^2 - \left\| Net(x) - Net(x^-) \right\|_2^2 + \alpha \right]_+$$

where α is a margin that is enforced between positive and negative pairs.
But this will lead to a negative hard sampling need, and a pre-defined margin.

Training the network

- Instead, a SoftMax function is applied on both outputs - effectively creating a ratio measure.

$$Loss(d_+, d_-) = \|(d_+, d_- - 1)\|_2^2 = const \cdot d_+^2$$

where

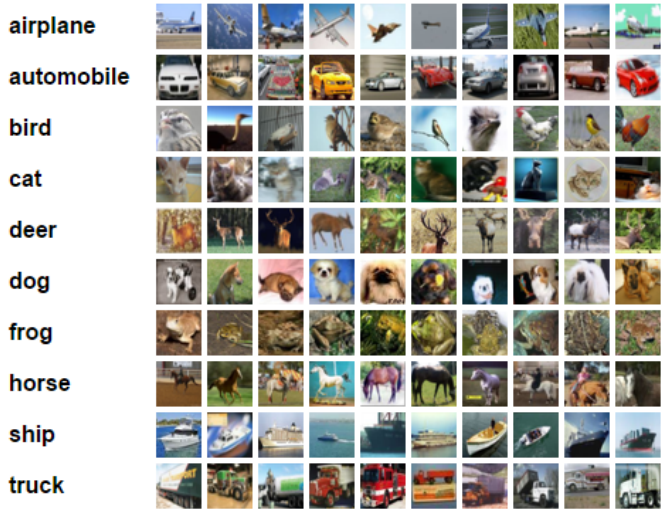
$$d_\pm = \frac{e^{\|Net(x) - Net(x^\pm)\|_2}}{e^{\|Net(x) - Net(x^+)\|_2} + e^{\|Net(x) - Net(x^-)\|_2}}$$

- We note that $Loss(d_+, d_-) \to 0$ iff $\frac{\|Net(x) - Net(x^+)\|}{\|Net(x) - Net(x^-)\|} \to 0$, which is the required objective.
- By using the same shared parameters network, we allow the back-propagation algorithm to update the model with regard to all three samples simultaneously.

Experiments

We experimented with 4 datasets

- *Cifar10* - 60K 32x32 color images of 10 classes.
- *MNIST* - 60K 28x28 gray-scale images of digits 0-9
- *Street-View-House-Numbers (SVHN)* - 600K 32x32 color images of house-number digits 0-9.
- *STL10* - similar to Cifar10, only with 5K labeled, 100K unlabeled and a bigger 96x96 image size.



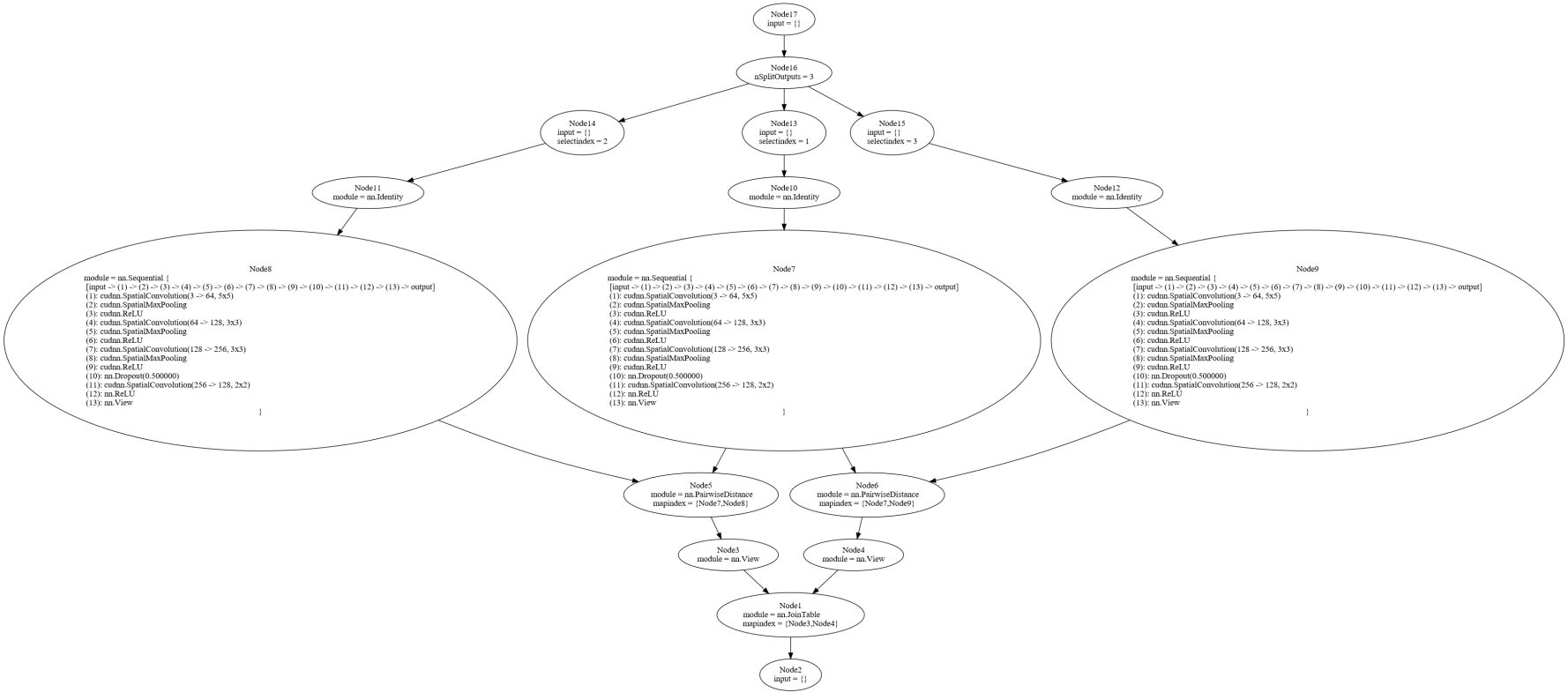
Cifar10 Dataset

SVHN Dataset

Outline

The Embedding Net

The full Triplet Network - 3 instances of embedding network, L_2 distance measure and SoftMax comparison.



The Embedding Net

Node7

```
module = nn.Sequential {
 [input -> (1) -> (2) -> (3) -> (4) -> (5) -> (6) -> (7) -> (8) -> (9) ->
 (1): cudnn.SpatialConvolution(3 -> 64, 5x5)
 (2): cudnn.SpatialMaxPooling
 (3): cudnn.ReLU
 (4): cudnn.SpatialConvolution(64 -> 128, 3x3)
 (5): cudnn.SpatialMaxPooling
 (6): cudnn.ReLU
 (7): cudnn.SpatialConvolution(128 -> 256, 3x3)
 (8): cudnn.SpatialMaxPooling
 (9): cudnn.ReLU
 (10): nn.Dropout(0.500000)
 (11): cudnn.SpatialConvolution(256 -> 128, 2x2)
 (12): nn.ReLU
 (13): nn.View
                                                    }
```

Outline

Results

Dataset	TripletNet	SiameseNet	Best known result
Mnist	99.54±0.08%	97.9±0.1%	99.61%
Cifar10	87.1±0.07%	-	90.22%
SVHN	95.37±0.08%	-	98.18%
STL10	70.67±0.1%	-	67.9%

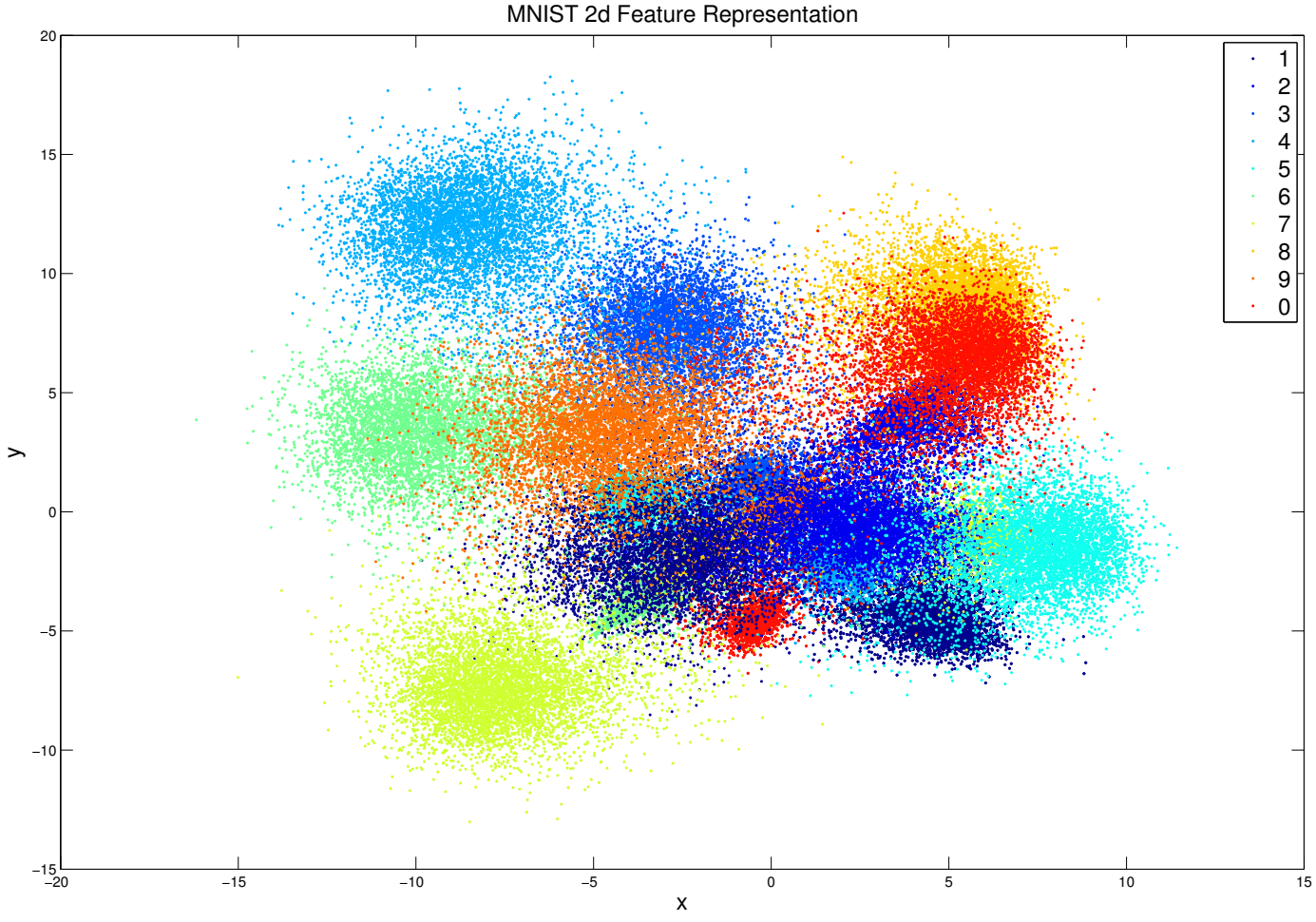
Table: Classification accuracy (no data augmentation)

- These results are comparable to state-of-the-art results with a deep learning model trained explicitly to classify samples, without using any data augmentation.
- We note that similar results are achieved when the embedded representations are classified using a linear SVM model or KNN classification with up to 0.5% deviance from the results.

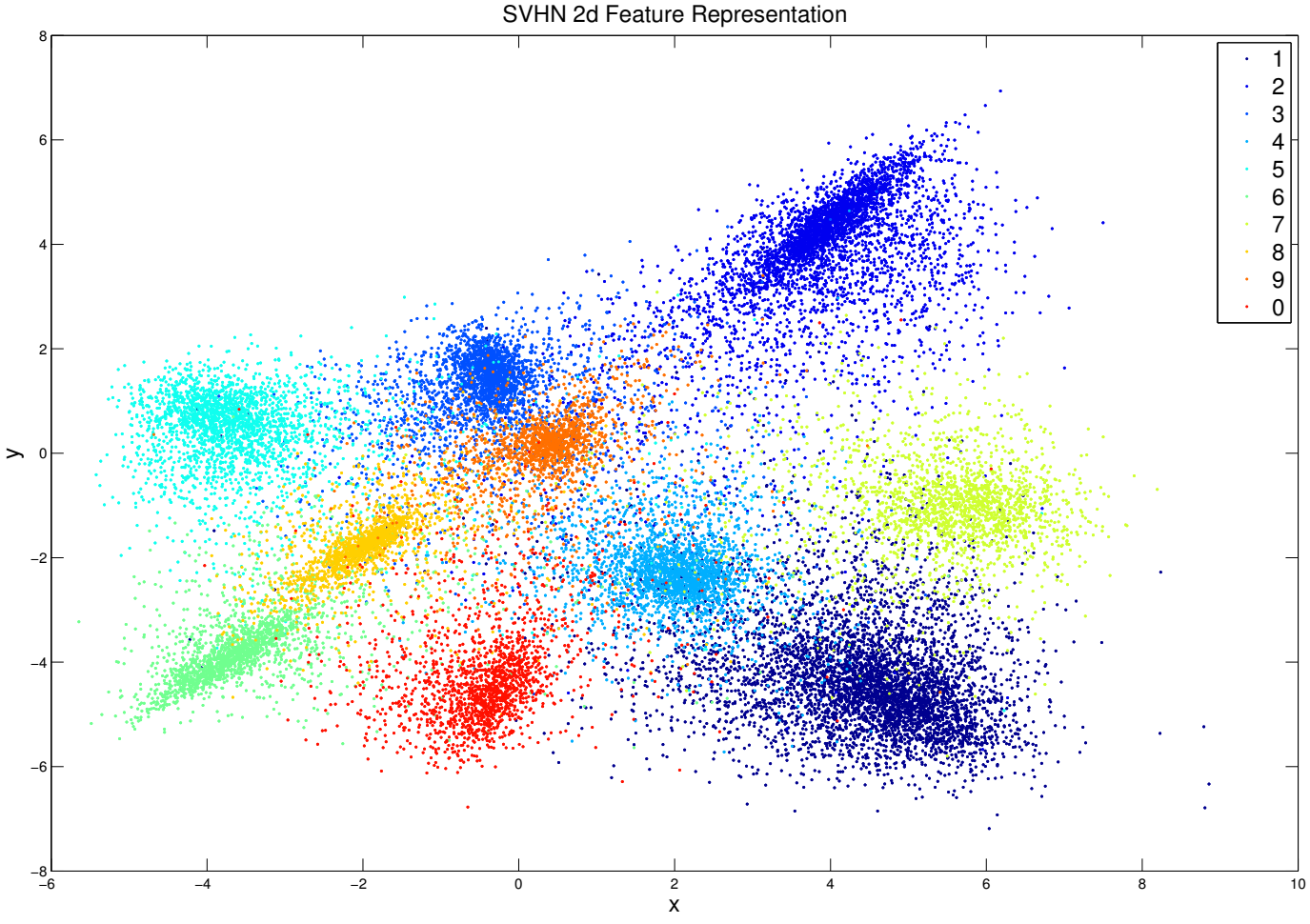
Results

- We were also able to show that the Triplet Network provided better results than its immediate competitor, the Siamese network, which was trained using a contrastive loss and the same embedding network.
- By projecting to a 2d plane (using PCA) we can see that learned features are discriminative between classes.

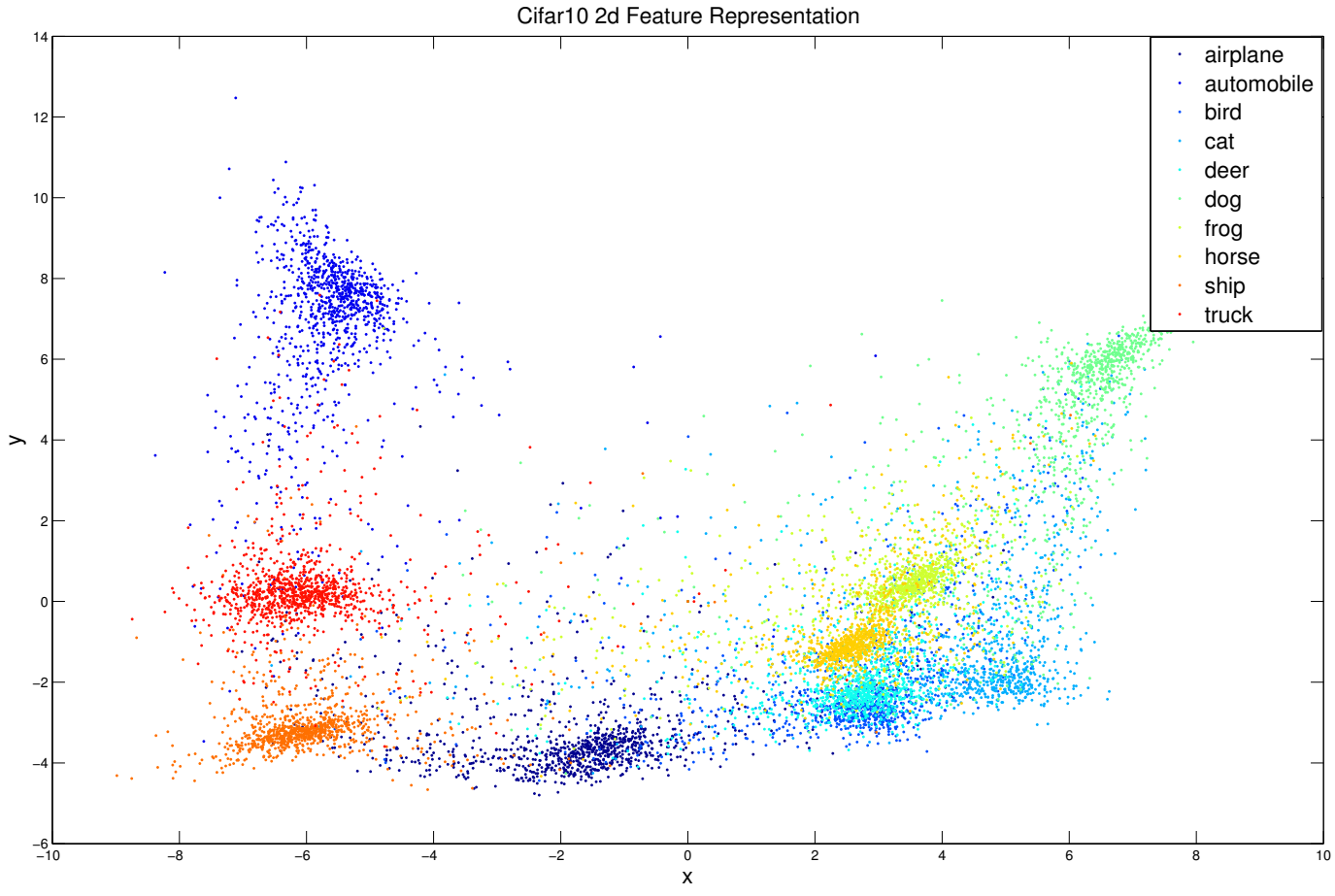
MNIST - Euclidean representation



SVHN - Euclidean representation



CIFAR10 - Euclidean representation



Other benefits of TripletNets

- Learning rate can be much higher than usual (x10). Possibly due to normalization of gradients
- No classification layer - which usually causes fast overfitting
- Feature vectors were found to be rather sparse - about 25% non zero values.
- No need for a fixed number of classes
- Very useful for ranking and data-retrieval tasks.
- Currently very popular for face verification (Google's FaceNet, Facebook)

Future Research

As the Triplet net model allows learning by comparisons of samples instead of direct data labels, usage as an unsupervised learning model is possible.

- **Using spatial information.** Objects and image patches that are spatially near are also expected to be similar from a semantic perspective.
- **Using temporal information.** Two consecutive video frames are expected to describe the same object, while a frame taken 10 minutes later is less likely to do so.

It is also well known that humans tend to be better at accurately providing comparative labels. Our framework can be used in a crowd sourcing learning environment. Comparisons over similarity measures are also much easier to attain.

Video Triplets - Using temporal information

Examples - triplets taken from videos

Summary

In this work it was shown that

- This work provides evidence that the representations that were learned are useful for classification in a way that is comparable with a network that was trained explicitly to classify samples.
- We have also shown how this model learns using only comparative measures instead of labels, which we can use in the future to leverage new data sources for which clear out labels are not known or do not make sense (e.g hierarchical labels).

Further research is needed to improve upon those findings and allow better results that are competitive with state-of-the-art results of conventional deep networks.

For Further Reading

- Original paper (with all mentioned citations):
  “Deep metric learning using Triplet network”
  `http://arxiv.org/abs/1412.6622`
- Full code (Torch) is available at:
  `https://github.com/eladhoffer/TripletNet`