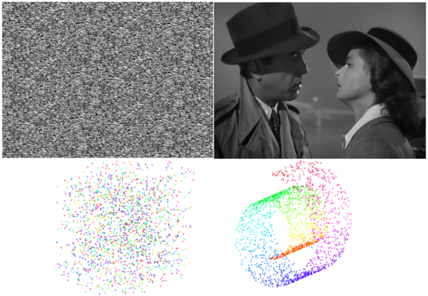
Factorization of High Dimensional Data using a Temporal Auto-Encoder Framework

Ross Goroshin   Joan Bruna   Arthur Szlam

December 18, 2013

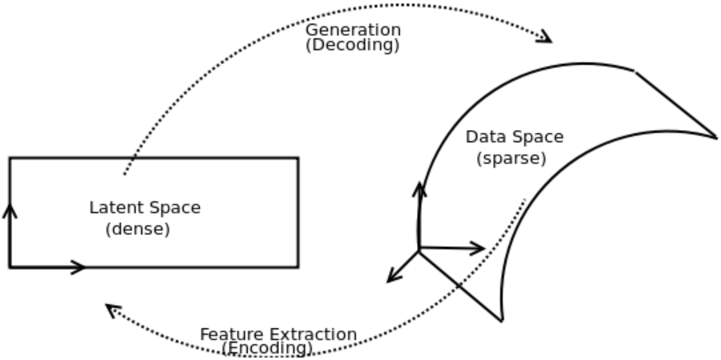
NEW YORK UNIVERSITY

Structure in Natural Data & Statistical Dependence



- Suppose we have a 42 second video played at 24 frames/second, with a resolution of 1000 by 1000 pixels
- In theory each pixel can vary independently from frame to frame, which implies that there are $\approx 10^9$ degrees of freedom
- If all pixels in natural images were i.i.d then natural images would fill the space. Sampling from this distribution would produce natural images.

Structure in Natural Data & Statistical Dependence



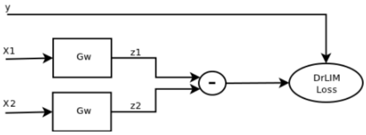
- This illustration is representative of many processes
- However, dependence can be introduced without increasing the dimensionality
- Latent representation is NOT unique for generative processes of interest

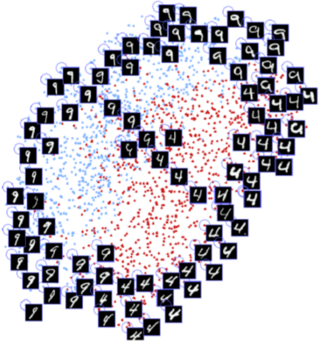
Unsupervised Learning

- Unsupervised learning algorithms should, in some sense, *parameterize* the data manifold
- These parameters are referred to as *features or factors*
- A good feature representation should satisfy the following criteria: (1) the features are mutually independent, (2) the representation is *not one that is invariant* but *linearly equivariant* to the set of transformation groups present in the data. This makes it possible to formulate any classification, detection, or regression task as a linear problem in feature space. Finally, (3) because we do not know the task a priori, the represntation should be loss-less, i.e. is information preserving.

Dimensionality Reduction by Learning an Invariant Mapping (DrLIM)

- We wish to find a mapping $G_W(X_i) : \mathbb{R}^D \to \mathbb{R}^d$, where $D > d$ which translates labeled similarity relationships in the input space to Euclidean distances in the output space
- If (X_1, X_2) are similar then $Y = 0$, otherwise $Y = 1$
- Let $D_W(X_1, X_2) = \|G_W(X_1), G_W(X_2)\|_2$
- $L(W, Y, X_1, X_2) = (1 - Y)\frac{1}{2}D_W^2 + Y\frac{1}{2}\{max(0, m - D_W)\}^2$



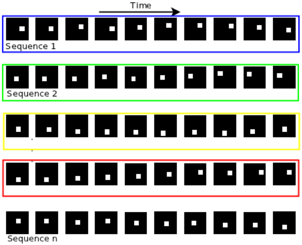


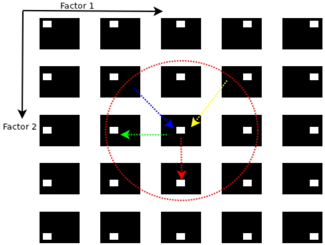
Although DrLIM can be used to extract some of the underlying factors that describe a high-dimensional dataset, there are several limitations which preclude it from being a general purpose feature learning algorithm.

- Where do we obtain the similarity labels Y?
- DrLIM does not include a reconstruction cost, and thus produces a non-invertible (lossy) feature set, i.e. the mapping $G_w()$ is trained to be invariant
- The features extracted are not guaranteed to be independent (factorized representation). This becomes a problem as the number of features increases

Problem 1 or 3: Similarity Labels and the Role of Time

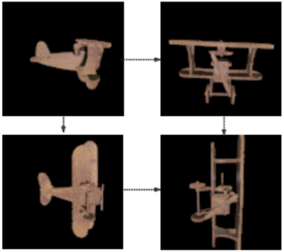
- If the objects in consecutive frames do not overlap then all samples are equidistant from each other
- *Without prior knowledge*, meaningful neighborhood relationships can only be deduced from temporally coherent sequences of images (i.e. movie clips)

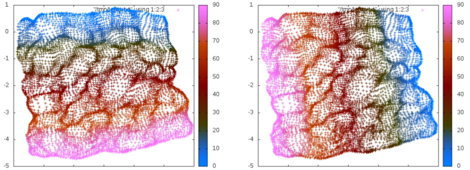




DrLIM Applied to Temporally Coherent Data

- 2-dimensional manifold living in a $\approx 10,000$-dimensional space (96x96 images)
- Similarity relationships can be naturally assigned via adjacent frames in a video

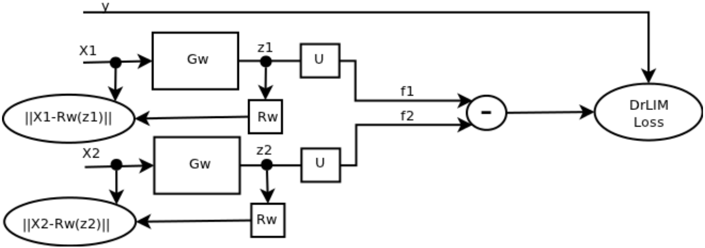




Problem 2 of 3: *Invariance VS Equivariance*

- The mapping $G_w()$ is trained to extract the underlying factors which generate a particular temporal sequence
- However, this mapping is trained to be invariant to any other variations that may be present in the data
- The mapping $G_w()$ is *not necessarily invertible*, thus it is not information preserving

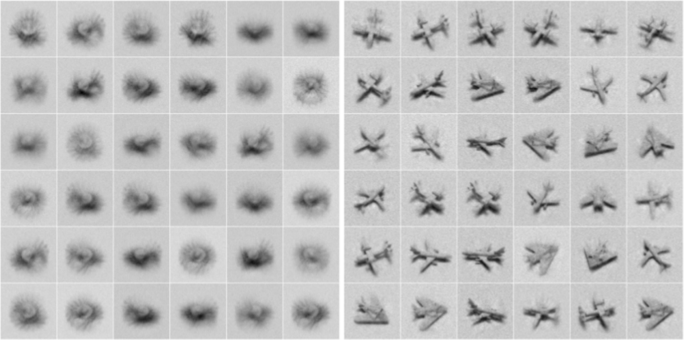
- $Gw()$ and $Rw()$ are trainable functions called the 'encoder' and 'decoder', respectively
- z_i is an intermediate, high dimensional representation
- U is a trainable *linear* map



$$L = (1-y)\|U\left(G_w(X_1) - G_w(X_2)\right)\| + y\ max(0,\ m - \|U\left(G_w(X_1) - G_w(X_2)\right)\|)$$
$$+\ \alpha\left[\|R_w(G_w(X_1)) - X_1\|^2 + \|R_w(G_w(X_2)) - X_2\|^2\right]$$

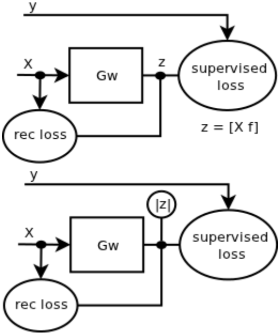
DrLIM + Reconstruction Loss

- Encoder has two stages: $z = G_w(X)$ and $f = Uz$
- The representation z is one from which we can: (i) reconstruct the input and (ii) linearly extract the factors of variation
- Thus z is a factorized representation in which the factors have been disentangled from the rest of the information



A Simpler Experiment

- Assume that $G_w()$ produces a large (over-complete) feature set
- $z = [X \;\; f]$ is a possible solution to the unregularized network. Where f are the relavant features for the supervised task
- Since f is causally extracted from X they are not independent

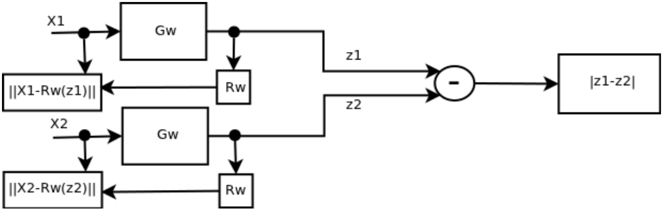


Problem 3 of 3: Promoting Independent Features

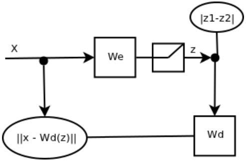
- A 'factorized' representation implies that the features extracted are independent
- Independence can be encouraged by maximizing the sparsity of $z_1 - z_2$
- This means that only a small number of features change between adjacent video frames
- Use a sparifying norm $\|z_1 - z_2\|_p$ where $p \leq 1$

Slow, Sparse Transition Features

- $|z_1 - z_2|_1$ not only encourages the features to vary slowly in time, but also encourages $z_1 - z_2$ to be as sparse as possible
- This is also called the total-variation (TV-norm)
- The implicit prior corresponding to this penalty is that *only a small set of latent factors vary between adjacent frames*



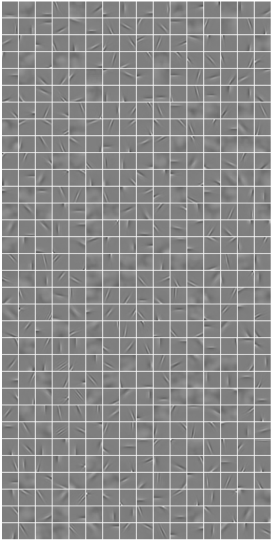
Slow Features Experiment #1



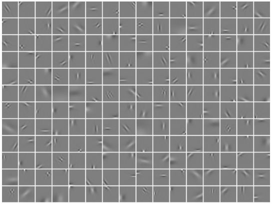
Encoder: ReLU

Decoder: Norm Linear

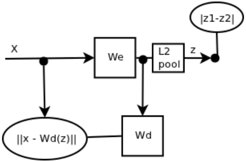
L_1



SF

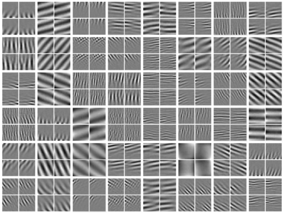


Slow Features Experiment #2

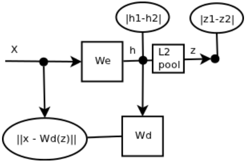


Encoder: Linear
Decoder: Norm Linear

L_2-pooling: $z_{11} = \sqrt{\sum_{i=1}^{4}(W_i^e x)^2}$

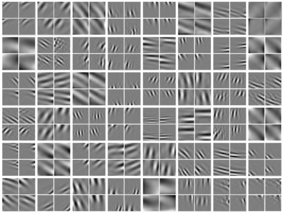


Slow Features Experiment #3

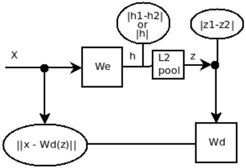


Encoder: Linear
Decoder: Norm Linear

L_2-pooling: $z_{11} = \sqrt{\sum_{i=1}^{4}(W_i^e x)^2}$

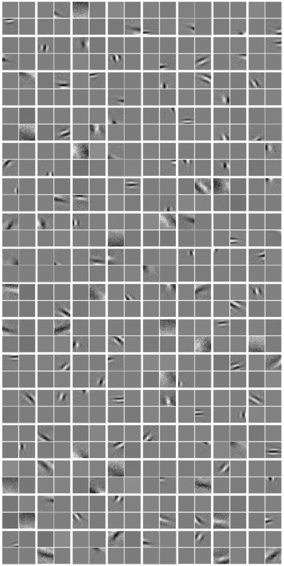


Slow Features Experiment #4

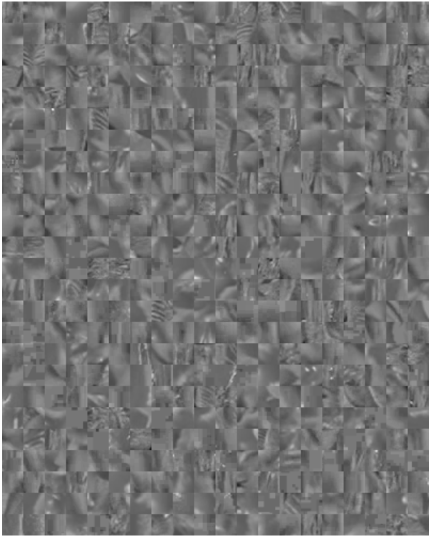


Encoder: Linear
Decoder: Norm Linear

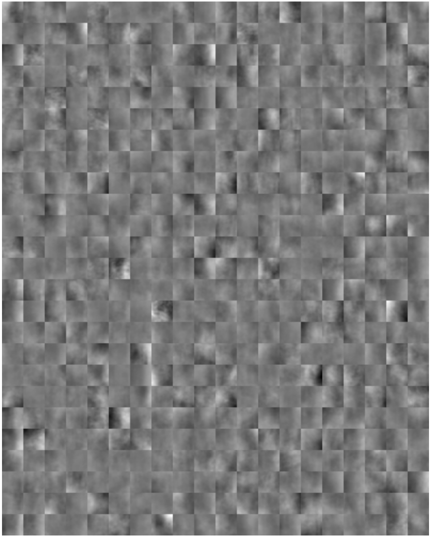
L_2-pooling: $z_{11} = \sqrt{\sum_{i=1}^{4}(W_i^e x)^2}$



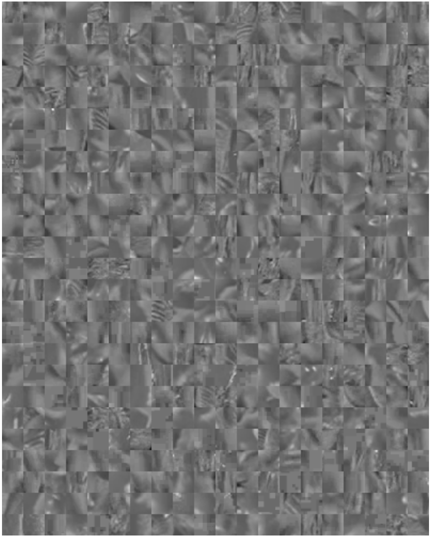
Input Data



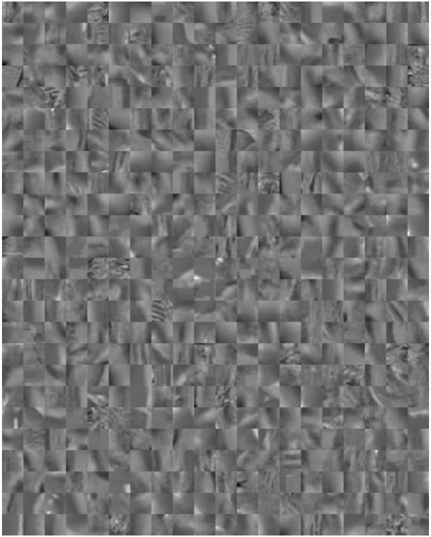
Linear Reconstruction



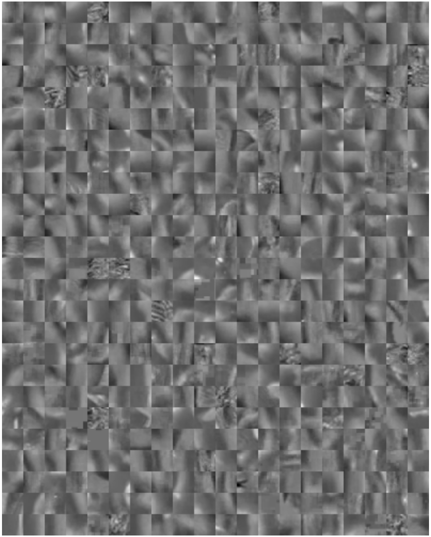
Input Data

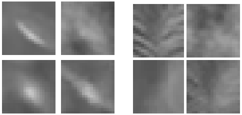


kNN-Code Space Reconstruction



kNN-Input Space Reconstruction

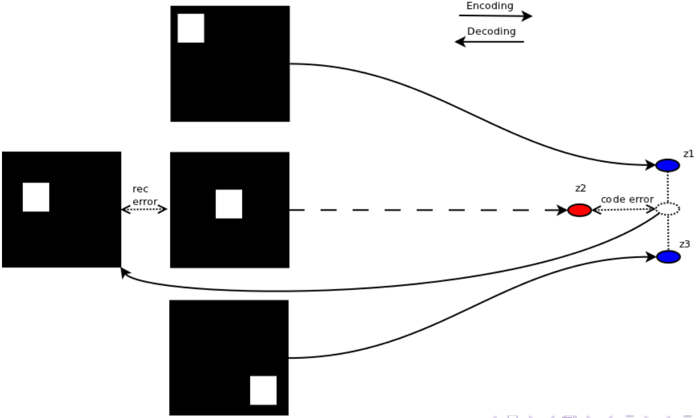




input	pinv
kNN input	kNN code

Curvature

- If we are after linearly equivariant features then it makes sense to minimize their curvature
- This requires three samples: $\|2z_2 - z_1 - z_3\|$
- We can also test the flatness of the representation and the quality of the decoder as follows:



Thank You

THE END