Autoencoder

Xiaogang Wang

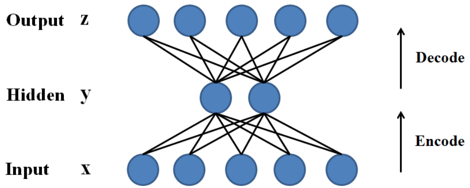
xgwang@ee.cuhk.edu.hk

February 14, 2015

Outline

Autoencoder

- An autoencoder takes an input $\mathbf{x} \in [0,1]^d$ and first maps it (with an encoder) to a hidden representation $\mathbf{y} \in [0,1]^{d'}$ through a deterministic mapping
$$\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b})$$
where s is a non-linear activation function (such as sigmoid).
- $\mathbf{y}$ is mapped back (with a decoder) into a reconstruction $\mathbf{z}$ of the same shape as $\mathbf{x}$,
$$\mathbf{z} = s(\mathbf{W}'y + \mathbf{b}')$$
$\mathbf{z}$ is seen as a prediction of $\mathbf{x}$.



Autoencoder

- Encoder

$$\mathbf{y} = f_\theta(\mathbf{x})$$

- Decoder

$$\mathbf{z} = g_\theta(\mathbf{y})$$

$$\theta = \{\mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}'\}$$

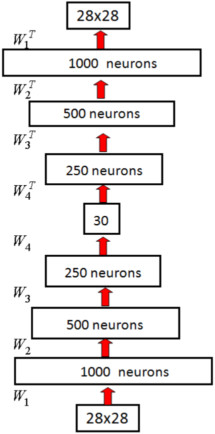
- It is important to add regularization in the training criterion or the parametrization to prevent the auto-encoder from learning the identity function, which would lead zero reconstruction error everywhere
- A particular form of regularization consists in constraining the code to have a low dimension, and this is what the classical auto-encoder or PCA do.

Autoencoder

- Optionally, the weight matrix $\mathbf{W}'$ of the reverse mapping may be constrained to be the transpose of the forward mapping: $\mathbf{W}' = \mathbf{W}^T$, referred to as **tied weights**
- The objective function measures the reconstruction error
  - Squared error: $J(\mathbf{x}, \mathbf{z}) = ||\mathbf{x} - \mathbf{z}||^2$
  - Cross-entropy: $J(\mathbf{x}, \mathbf{z}) = -\sum_{i=1}^{d} [x_i \log z_i + (1 - x_i) \log(1 - z_i)]$
- $\mathbf{y}$ is expected a distributed representation that captures the main factors of variation in data.
- If there is one linear hidden layer and the mean squired error criterion is used to train the network, the k hidden unites learn to project the input in the span of the first k principal components of data.
- Autoencoder gives low reconstruction error on test examples from the same distribution as the training examples, but generally high reconstruction error on samples randomly chosen from the input space
- Autoencoder is a multi-layer neural network. The only difference is that the size of its output layer is the same as the input layer and the objective function

Deep autoencoder

- Stack multiple encoders (and their corresponding decoders)



Deep autoencoder

- Very difficult to optimize deep autoencoders using backpropagation
- Pre-training + fine-tuning
  - First train a stack of RBMs
  - Then "unroll" them
  - Then fine-tune with backpropagation

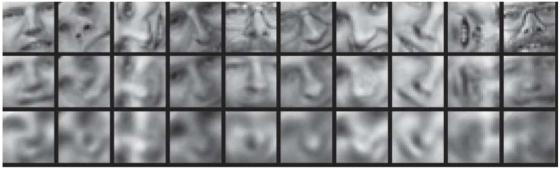
Comparison of methods of compressing images

real data

30-D deep auto

30-D logistic PCA

30-D PCA



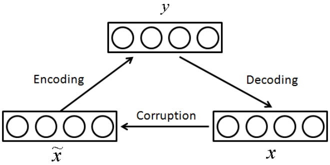
real data

30-D deep auto

30-D PCA

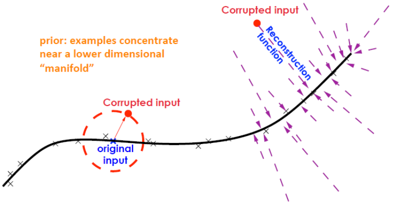
Denoising autoencoder

- In order to force the hidden layer to discover more robust features and prevent it from simply learning the identity function, train the autoencoder to reconstruct the input from a corrupted version of it
  - Encode the input (preserve the information about the input)
  - Undo the effect of a corruption process stochastically applied to the input of the auto-encoder
- To convert the autoencoder to a denoising autoencoder, all we need to do is to add a stochastic corruption step operating on the input
  - Randomly sets some of the inputs (as many as half of them) to zero. Hence the denoising auto-encoder is trying to predict the corrupted (i.e. missing) values from the uncorrupted (i.e., non-missing) values, for randomly selected subsets of missing patterns.
  - The input can be corrupted in other ways



Denoising autoencoder

- The learner must capture the structure of the input distribution in order to optimally undo the effect of the corruption process, with the reconstruction essentially being a nearby but higher density point than the corrupted input
- The denoising autoencoder is learning a reconstruction function that corresponds to a vector field pointing towards high-density regions (the manifold where examples concentrate)
- Denosing autoencoder basically learns in $r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}$ a vector pointing in the direction $\frac{\partial \log P(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}$



Predictive Sparse Decomposition

- Sparse coding
  - Solving the encoder f_θ is non-trivial because of L_1 minimization and entails an iterative optimization

$$\mathbf{y}^* = f_\theta(\mathbf{x}) = \arg\min_{\mathbf{y}} ||\mathbf{x} - \mathbf{W}\mathbf{y}||_2^2 + \lambda||\mathbf{y}||_1$$

$$J_{SC} = \sum_n ||\mathbf{x}^{(n)} - \mathbf{W}\mathbf{y}^{*(n)}||_2^2$$

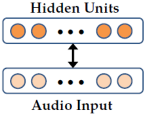
- Predictive sparse decomposition
  - Approximation to sparse coding
  - Add sparse penalty to auto-encoder
  - Replace the costly and highly non-linear encoding step by a fast non-iterative approximation
  - The training criterion is simultaneously optimized with respect to the hidden codes (representation) $\mathbf{y}^{(n)}$ and with respect to the parameters θ

$$J_{PSD} = \sum_n \lambda||\mathbf{y}^{(n)}||_1 + ||\mathbf{x}^{(n)} - \mathbf{W}\mathbf{y}^{(n)}||_2^2 + ||\mathbf{y}^{(n)} - f_\theta(\mathbf{x}^{(n)})||_2^2$$

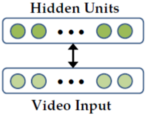
$$f_\theta(\mathbf{x}^{(n)}) = \sigma(\mathbf{b} + \mathbf{W}^T\mathbf{x}^{(n)})$$

Multimodal deep learning

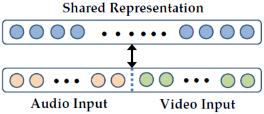
- Ngiam et al. ICML'11 (audio-visual speech recognition)
- Multimodal fusion: data from all modalities is available at all phases
- Cross modality learning: data from multiple modalities is available only during feature learning; during supervised training and testing, only data from a single modality is provided. The aim is to learn better single modality representations given unlabeled data from multiple modalities.
- Matching across different modalities
- A direct approach is to train a RBM over the concatenated audio and video data. Limited as a shallow model, it is hard for a RBM to learn the highly nonlinear correlations and form multimodal representations
- It was found that learning a shallow bimodal RBM results in hidden units that have strong connections to variables from individual modality but few units that connect across the modalities.



(a) Audio RBM



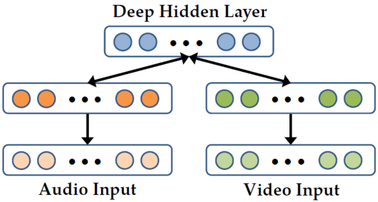
(b) Video RBM



(c) Shallow Bimodal RBM

Multimodal deep learning

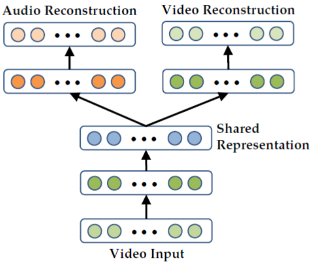
- Bimodal DBN: greedily layerwise training a RBM over the pre-trained layers for each modality; by representing the data through learned multilayer representations, it can be easier for the model to learn higher-order correlations across modalities. In (d), the first layer representations correspond to phonemes and visemes and the second layer models the relationships between them.
- Problems
  - It is possible for the model to find representations such that some hidden units are tuned only for audio while others are tuned only for video
  - It is not applicable in a cross modality learning setting where only one modality is present during supervised training and testing.



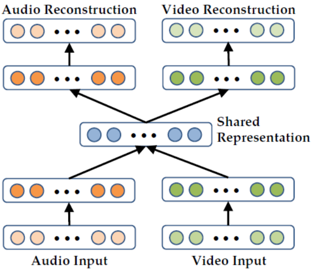
(d) Bimodal DBN

Multimodal deep auto-encoder

- In (a), the deep auto-encoder is trained to reconstruct both modalities when given only video data and thus discovers correlations across the modalities.
- Initialize the deep autoencoder with the bimodal DBN weights and discard any weights that are no longer present. The middle layer can be used as the new feature representation.
- Model (a) is used when only a single modality is present at supervised training and testing



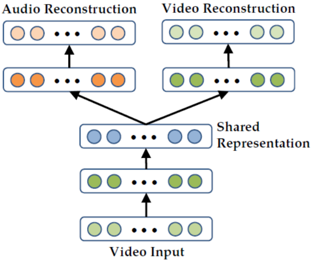
(a) Video-Only Deep Autoencoder



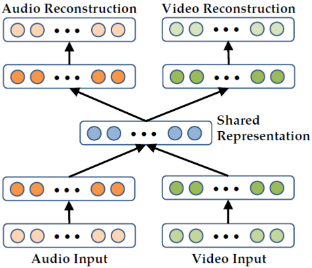
(b) Bimodal Deep Autoencoder

Multimodal deep auto-encoder

- Inspired by denoising auto-encoder, train model (b) with an augmented dataset
  - One-third of the training data has only video for input (setting zero values for the audio data)
  - Another one-third of the data has only audio
  - The last one-third of the data has both audio and video



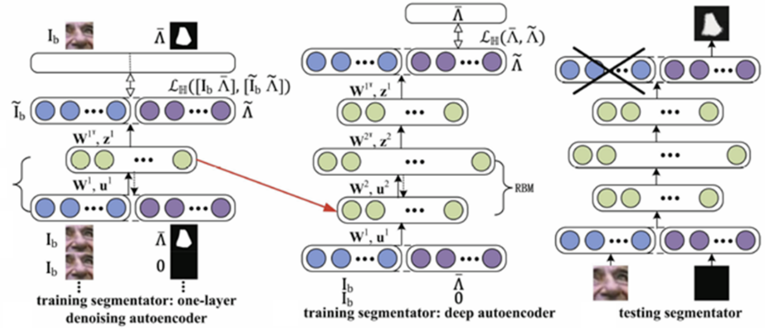
(a) Video-Only Deep Autoencoder



(b) Bimodal Deep Autoencoder

Multimodal deep auto-encoder

- For image segmentation: Luo et al. CVPR'12



Reading materials

- G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, Vol. 313, pp. 504-507, July 2006.
- K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "Fast Inference in Sparse Coding Algorithms with Applications to Object Recognition," CBLL-TR-2008-12-01, NYU, 2008.
- J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal Deep Learning," ICML 2011.
- P. Luo, X. Wang, and X. Tang, "Hierarchical Face Parsing via Deep Learning," CVPR 2012.