Deep Autoencoder

Ke Chen

Department of Computer Science, The University of Manchester

Ke.Chen@manchester.ac.uk

Outline

Introduction
Limitation of shallow autoenoders, challenges in deep autoencoders, other application

Hybrid Learning Strategy
Greedy layerwise pre-training, fine-tuning with BP algorithm, relevant issues

Deep Autoencoder
Architecture, learning procedure, RMB-based deep autoencoder, illustrate example

Case Study: learning speaker-specific representation
Background, regularised Siamese architecture, loss function, visualisation

Introduction

- In general, shallow autoencoders (AEs) have too limited capacity to deal with challenging problems in real worlds, hence deep AEs are demanded.
- Traditional AEs may be extended to deep AEs by adding more hidden layers, but extension of other shallow AEs, e.g., RBMs, to deep AEs is not straight-forward.
- Training deep NNs with the standard BP algorithm often fails to work in practice.
- In 2006, G. Hinton and his research students invented a hybrid learning strategy to tackle this challenge, as demonstrated with deep AEs constructed with RBMs.
- This breakthrough published in *Science* leads to resurgence of NNs and heralds a new era, nowadays called deep learning.
- Since 2006, deep AEs have become one of central themes in deep learning, which has led to a new ML research area named representation learning.
- Deep AEs not only work independently for representation learning but also serve as an important "ingredient" incorporated into other deep learning models for diversified representation learning tasks.

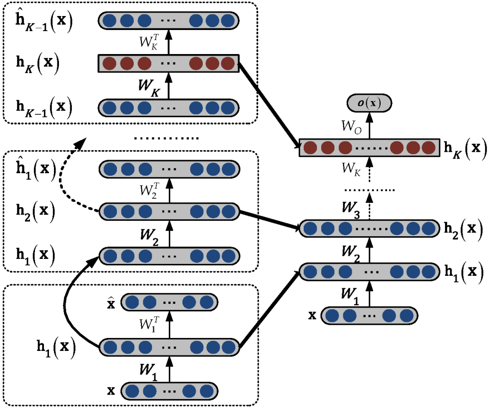
Hybrid Learning Strategy

- Motivation
  - Training deep NNs is a notorious non-convex optimisation problem where the parameter space is often huge and there are many local optimums.
  - Randomly initialisation of parameters and gradient-guided local search often cause the learning to end up with an unwanted local optimum.
  - Other issues such as saturation, vanish and exploration of gradients in the back-propagation process make training of deep NNs even harder.
- General idea behind hybrid learning
  - Two-stage learning strategy combining unsupervised and supervised learning
  - **Pre-training**: use unlabelled data (input instances only) to seek initial parameters that should be "better"than random initialisation in a greedy layerwise manner
  - **Fine-tuning**: starting with the pre-trained model, learn all parameters with training examples (input, target) for a classification or regression task

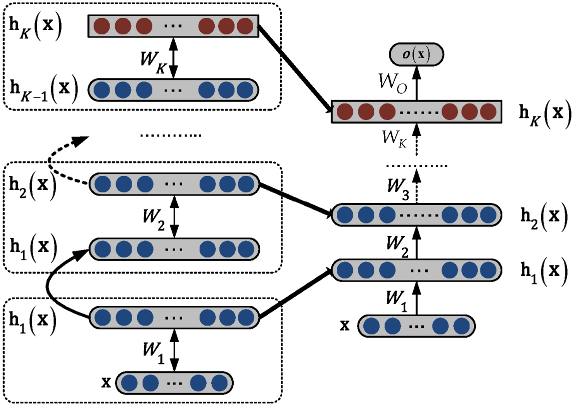
Unsupervised Pre-training

- Greedy layerwise representation learning with stacked shallow autoencoders
- Shallow autoencoders acted as "building blocks" to construct a deep neural network

Construction with AEs



Construction with RBMs



Greedy Layerwise Pre-training Procedure

Given a training dataset, $\mathcal{D} = \{\boldsymbol{x}_t)\}_{t=1}^{|\mathcal{D}|}$, randomly initialise the parameters of a chosen building block (e.g., AE or RBM) and pre-set all hyper-parameters in the building block.

For $k = 1, 2, \cdots, K$ do

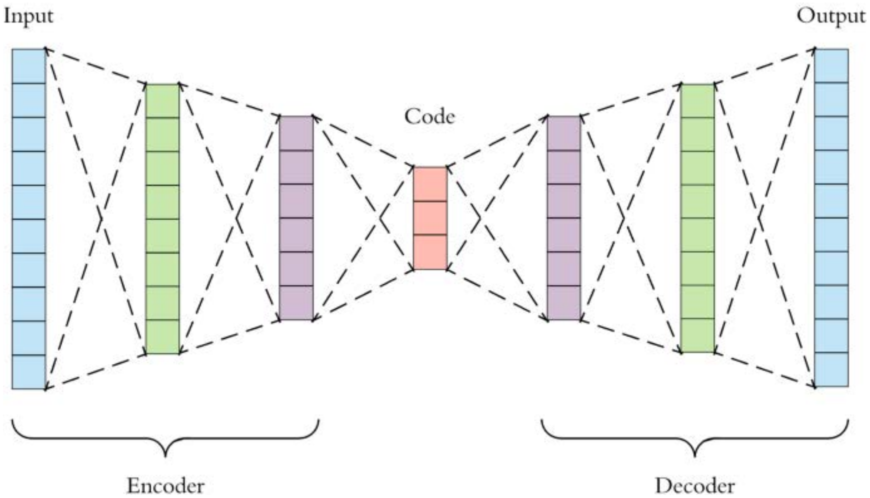
1. **Train a building block for hidden layer k**
   - set the number of neurons required by hidden layer k to be the dimension of the coding (hidden) layer in the chosen building block (shallow autoencoder)
   - with the training dataset, $\{\boldsymbol{h}_{k-1}(\boldsymbol{x}_t)\}_{t=1}^{|\mathcal{D}|}$, train the building block to achieve its optimal parameters (for $k = 0$, set $\{\boldsymbol{h}_0(\boldsymbol{x}_t)\}_{t=1}^{|\mathcal{D}|} = \{\boldsymbol{x}_t\}_{t=1}^{|\mathcal{D}|}$)
2. **Construct a DNN up to hidden layer k**
   - from the trained building block, discard its decoder and associated parameters
   - stack its hidden layer and associated parameters on top of the existing DNN

Finally, the output layer, $\boldsymbol{o}(\boldsymbol{x})$, is stacked onto hidden layer K with randomly initialised parameters to complete the DNN construction and its parameter initialisation.

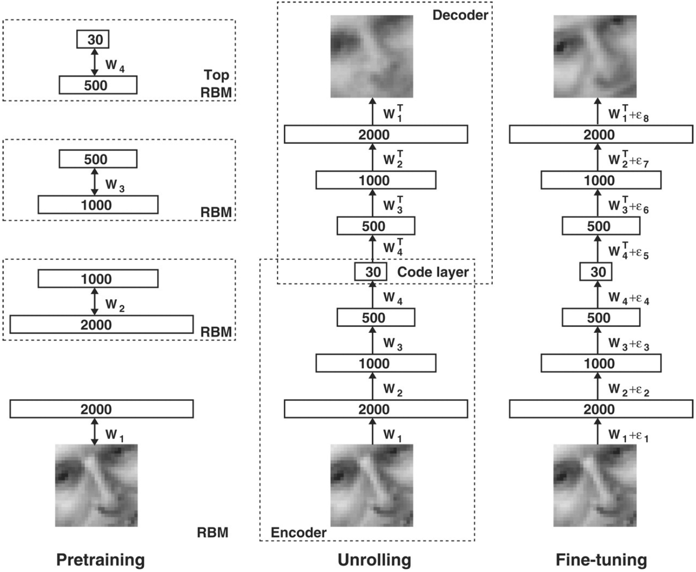
- **Why does this strategy work?**
  - Regularisation hypothesis: a probabilistic perspective
    - Unsupervised pre-training pushes model close to its natural distribution, $P(\boldsymbol{x})$
    - Representations good for $P(\boldsymbol{x})$ may also benefit to modelling $P(\boldsymbol{y}|\boldsymbol{x})$.
  - Optimisation hypothesis: a computational perspective
    - Unsupervised pre-training leading to near better local optimum of $P(\boldsymbol{y}|\boldsymbol{x})$
    - Higher likelihood to reach those better local minima not achievable by random initialisation due to the complex non-convex landscape
- **New insight into this strategy**
  - Many novel practical techniques have been invented for training DNNs much more effectively, such as new activation functions, e.g. ReLU familiy, batch and layer normalisation, drop-out and other regularisations, residual connections, $\cdots$
  - Since 2012, this strategy has been gradually abandoned in most circumstances unless (1) no similarity information in input feature; (2) very few labelled data in semi-supervised learning; (3) deep generative models.

Deep Autoencoder

- **Architecture**: symmetric configuration with respect to coding layer and tied weights
- **Learning algorithm**: (1) BP with supportive techniques; (2) hybrid learning strategy



RBM-based deep autoencoder via hybrid learning (Hinton and Salahutdinov, 2006)



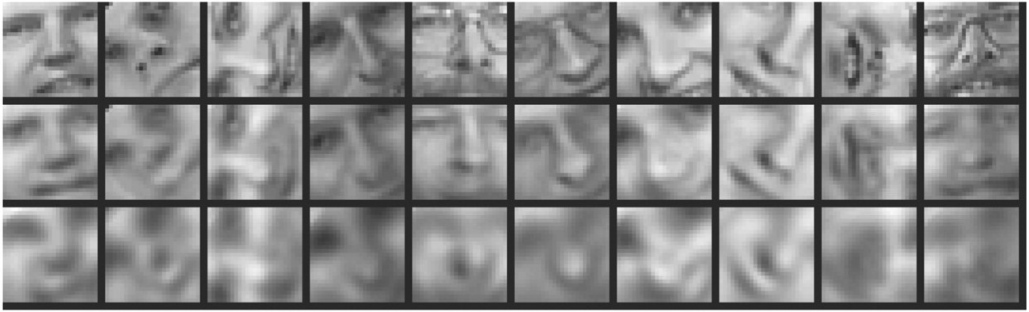
RBM-based deep autoencoder via hybrid learning (Hinton and Salahutdinov, 2006)

- **Data Compression**
  - Data set: grey-level image patches derived from the Olivetti face dataset
  - Model: 625-2000-1000-500-30 for encoder, linear neurons in coding layer and sigmoid neurons in all hidden layers, trained with hybrid learning
  - Comparison: Deep AE versus PCA (compressed code length, $p = 30$)

Original 25 X 25

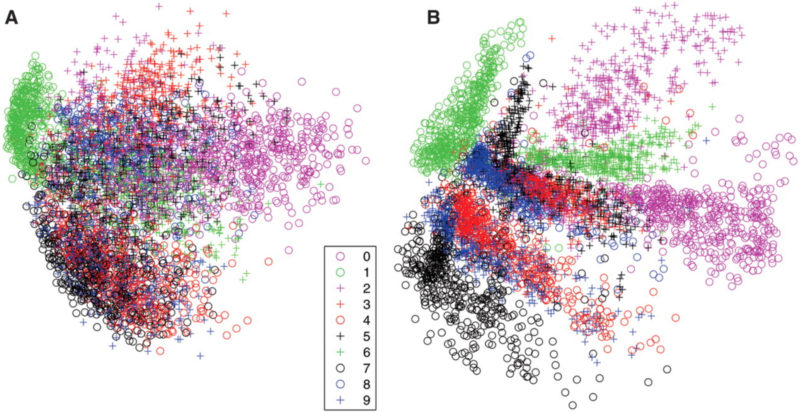
Deep-AE *p* = 30

PCA *p* = 30



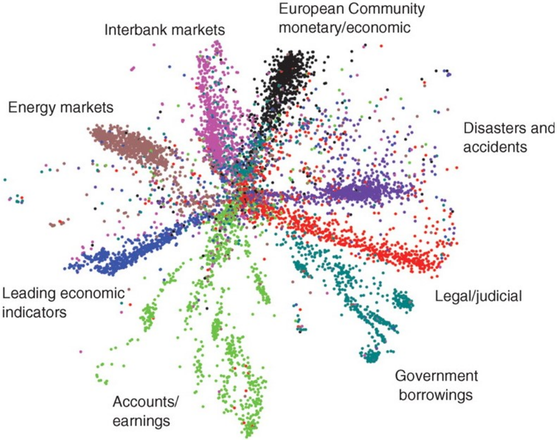
RBM-based deep autoencoder via hybrid learning (Hinton and Salahutdinov, 2006)

- **Visualisation**: (A) PCA versus (B) Deep AE
  - Handwritten digit: 28×28 grey-level images in MNIST dataset
  - Model: 768-1000-500-250-2 for encoder, linear neurons in coding layer and sigmoid neurons in all hidden layers, trained with hybrid learning



RBM-based deep autoencoder via hybrid learning (Hinton and Salahutdinov, 2006)

- **Visualisation**: Deep AE ($p = 10$) $\Rightarrow$ PCA ($p = 2$)
  - Text document: newswire stories in Reuter corpus
  - Model: 2000-500-250-125-10 for encoder, linear neurons in coding layer and sigmoid neurons in all hidden layers, trained with hybrid learning

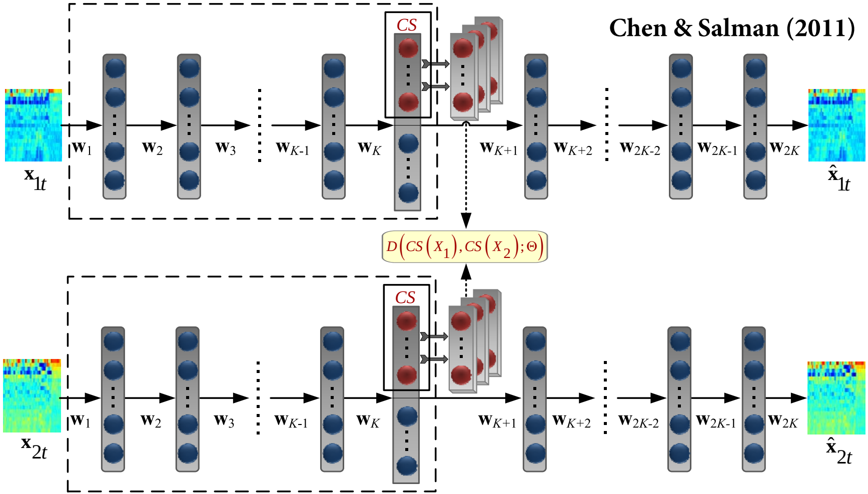


Background

- Speech signal: carrying various yet mixed information
  - Verbal: linguistic information (pre-dominated information source)
  - Non-verbal: speaker-specific, emotional and environmental information
  - Information components: entangled, dynamic and unequally distributed
- Various tasks: demanding specific information components
  - Speech recognition: speaker-independent linguistic information
  - Speaker recognition: text-independent speaker-specific information
  - Emotion recognition: non-verbal and/or verbal linguistic information
  - Unfortunately, all above tasks use the common speech representation, e.g., MFCC.
- Difference between data and information
  - Data: simply a carrier of mixed information components
  - Data component analysis, e.g., PCA, unable to disentangle information components
  - "Information component analysis" required for information disentanglement
- Research problem: how to extract speaker-specific information from speech signals?

Solution: Representation learning via novel deep neural architecture

- Key idea: learn speaker-specific distance regularised by preserving all information
- Regularised Siamese deep network (RSDN) proposed to carry out the idea



Loss Function

- Multi-objectives: (1) learn speaker-specific distance (2) minimise information loss
- Speaker-specific distance learning: weakly supervised contrastive learning
- Information loss minimisation: self-supervised learning via deep autoencoders
- Construction training dataset: $(X_1, X_2; \mathcal{I})$ where $\mathcal{I}$ is the label defined as $\mathcal{I} = 1$ if two speech segments, X_1 and X_2, are spoken by the same speaker and $\mathcal{I} = 0$ otherwise.

$$\mathcal{L}(X_1, X_2; \Theta) = \alpha[\mathcal{L}_R(X_1; \Theta) + \mathcal{L}_R(X_2; \Theta)] + (1 - \alpha)\mathcal{L}_D(X_1, X_2; \Theta),$$

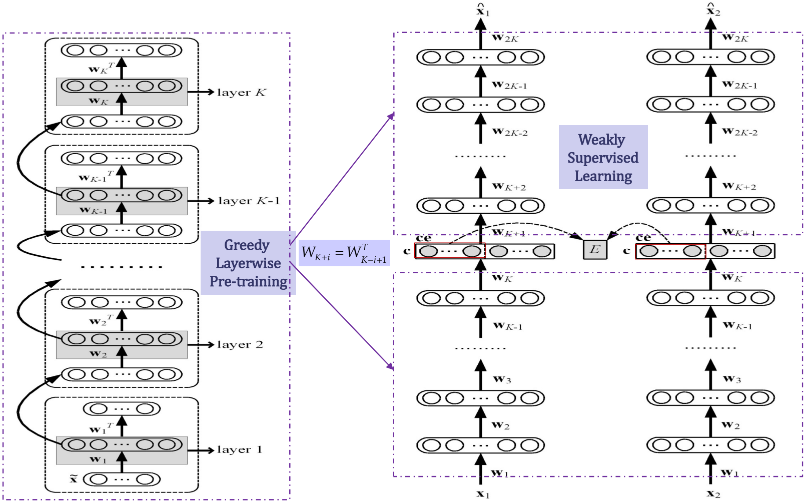
$$\mathcal{L}_R(X_i; \Theta) = \frac{1}{|\mathcal{B}|}\sum_{t=1}^{|\mathcal{B}|}||\boldsymbol{x}_{it} - \hat{\boldsymbol{x}}_{it}||^2 \quad i = 1, 2$$

$$\mathcal{L}_D(X_1, X_2; \Theta) = \mathcal{I}D + (1 - \mathcal{I})(e^{-\frac{D_m}{\lambda_m}} + e^{-\frac{D_S}{\lambda_S}}).$$

For CS in coding layer, $D = D_m + D_S$, $D_m = ||\boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)}||^2$, $D_S = ||\Sigma^{(1)} - \Sigma^{(2)}||_F^2$.

RSDN Training via Hybrid Learning

- Pre-training: DAEs with unlabelled data corrupted by Gaussian noise
- Fine-tuning: Stochastic gradient descent on the loss function with shared weights



Application of Speaker-specific Representation

- **Speaker modelling**
  For speech segment of $|B|$ frames,
  - Extract representation $CS(\boldsymbol{x}_t)\}_{t=1}^{|B|}$
  - Estimate mean and covariance matrix

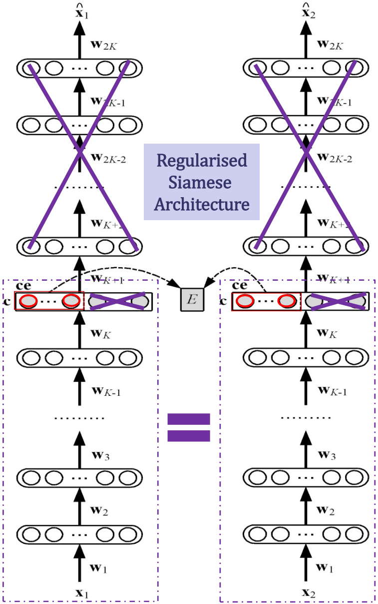
$$\boldsymbol{m} = \frac{1}{|B|-1}\sum_{t=1}^{|B|} CS(\boldsymbol{x}_t)$$

$$\Sigma = \frac{1}{|B|-1}\sum_{t=1}^{|B|} (CS(\boldsymbol{x}_t) - \boldsymbol{m})(CS(\boldsymbol{x}_t) - \boldsymbol{m})^T$$

- **Speaker-specific distance**
  For any two speaker models: $\mathrm{SM}_i = \{\boldsymbol{m}_i, \Sigma_i\}\ i = 1, 2$

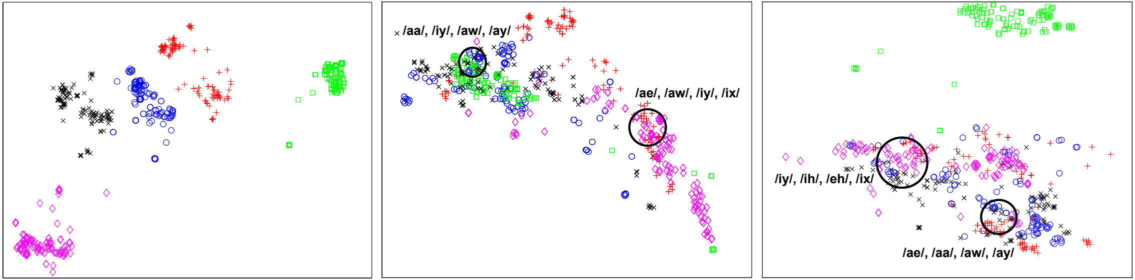
$$d(\mathrm{SM}_1, \mathrm{SM}_2) = \mathrm{Tr}\left((\Sigma_1^{-1} + \Sigma_2^{-1})(\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T\right)$$

If *d*(SM_1, SM_2)<*d*_0, SM_1 =SM_2.



Visualisation: Learned Representation versus MFCCs on TIMIT Data Set

- **Vowel**: main carrier of speaker-specific information in speech signals
- Speaker: 1 female and 4 male speakers (utterances of all 20 vowels in English)
- Comparison (2-D with t-SNE): (a) *CS* representation (speaker-specific), (b) $\overline{CS}$ representation (speaker-independent), (c) MFCCs (common speech representation)



(a) (b) (c)

If you want to deepen your understanding and learn something beyond this lecture, you can self-study the optional references below.

[Goodfellow et al., 2016] Goodfellow I., Bengio Y., and Courville A. (2016): *Deep Learning*, MIT Press. (Section 15.1)

[Chen, 2015] Chen K. (2015): Deep and modular neural networks. In *Springer Handbook of Computational Intelligence*, Chapter 28, pp. 473-492. (Sections 28.1-28.2)

[Hinton & Salakhutdinov, 2006] Hinton G. and Salakhutdinov R. (2006): Reducing the dimensionality of data with neural networks. *Science*, Vol. 313, pp. 504-507.

[Chen & Salman 2011] Chen K. and Salman A. (2011): Extracting speaker-specific information with a regularized Siamese deep network. In *Advances in Neural Information Processing Systems 25* (NIPS'11), MIT Press, pp. 298-306.