Deep Learning and it’s Application in Audio and Speech Processing

Bill Xia

CCNT Biometric Lab,

College of Computer Science,
Zhejiang University

ibillxia@zju.edu.cn

Outline

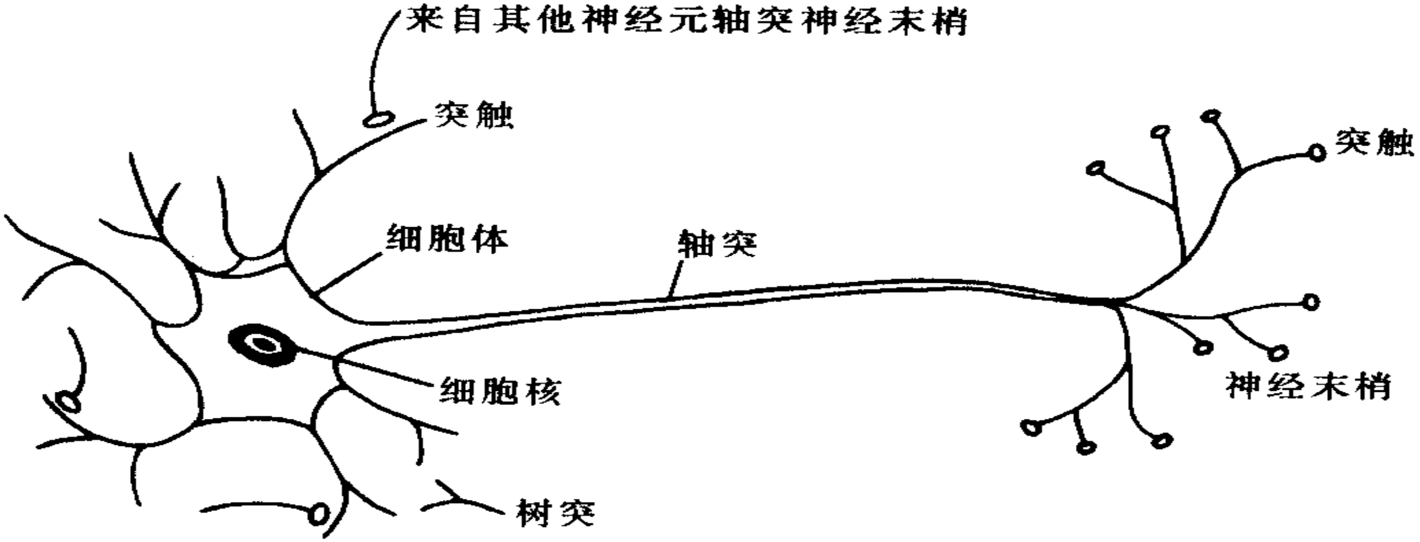
- Background
  - Human Auditory System
  - Neural Networks
- Deep Architectures
  - Motivations and Challenges
  - Attempts and Breakthrough
- Deep Learning
  - Autoencoders and Sparse Coding
  - Restricted Boltzmann Machines(RBMs)
  - Greedy Layer-Wise Training
  - Explaination and Justification
- Deep Learning in Audio and Speech
- References

Outline

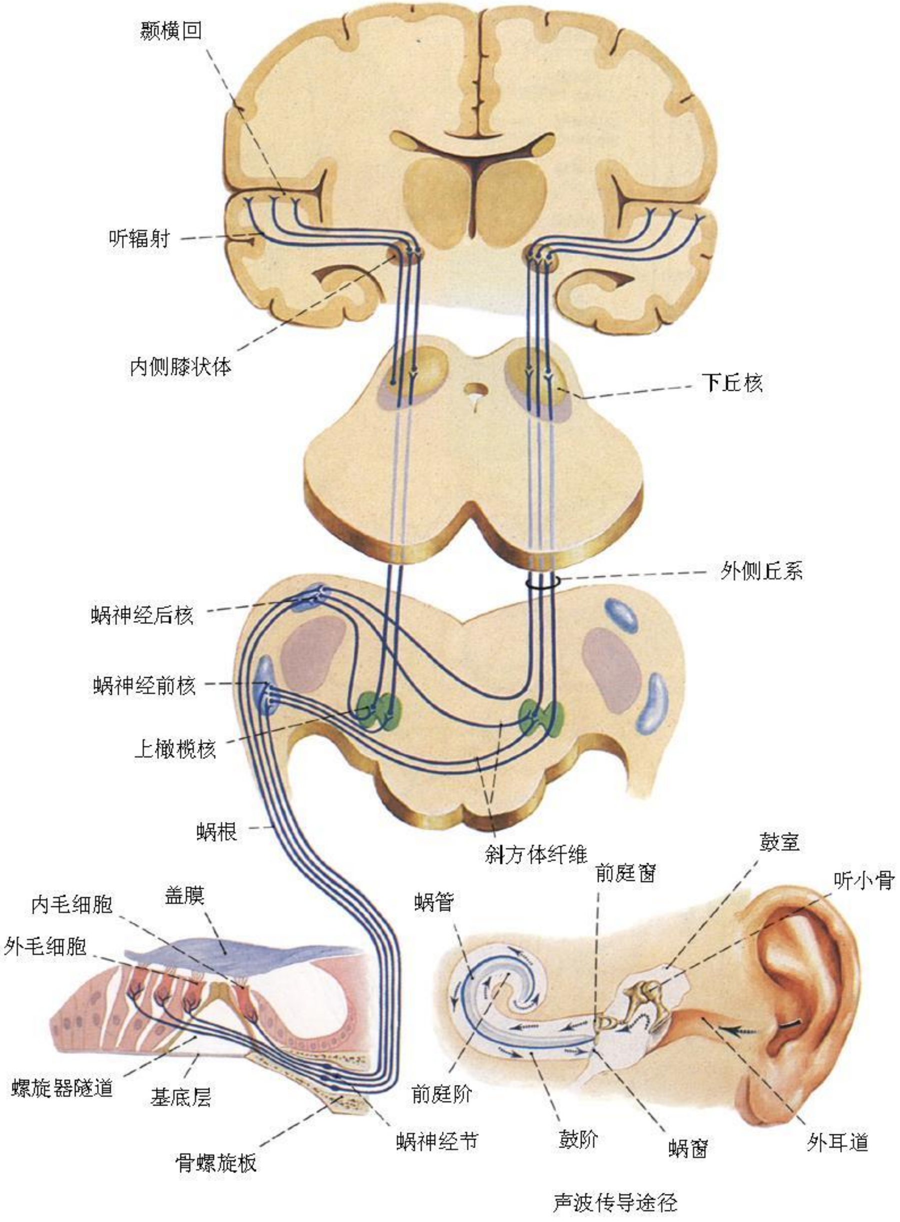
- Background
  - Human Auditory System
  - Neural Networks
- Deep Architectures
  - Motivations and Challenges
  - Attempts and Breakthrough
- Deep Learning
  - Autoencoders and Sparse Coding
  - Restricted Boltzmann Machines(RBMs)
  - Greedy Layer-Wise Training
  - Explaination and Justification
- Deep Learning in Audio and Speech
- References

Background - Human Auditory System

- 生物神经元
  - 神经元是大脑处理信息的基本单元，人脑大约由10^{11}个神经元组成，神经元互相连接成神经网络
- 功能和特性
  - 时空整合功能
  - 两种工作状态
  - 脉冲电位转换功能
  - 突触对神经冲动的传递有延时和不应期现象
  - 突触的传递作用有增强、减弱和饱和三种可能



Background - Human Auditory System



听觉系统-中枢部分
大脑中与听觉相关的部分称为听觉中枢，它纵跨脑干、中脑、丘脑的大脑皮层，是感觉系统中最长的中枢通路之一

自下向上主要环节包括：蜗神经核、上橄榄核、外侧丘系核、下丘核、丘脑的内侧膝状体、大脑皮层颞叶的听觉皮层等，左图所示为听觉中枢的传导通路。

Background - Neural Networks

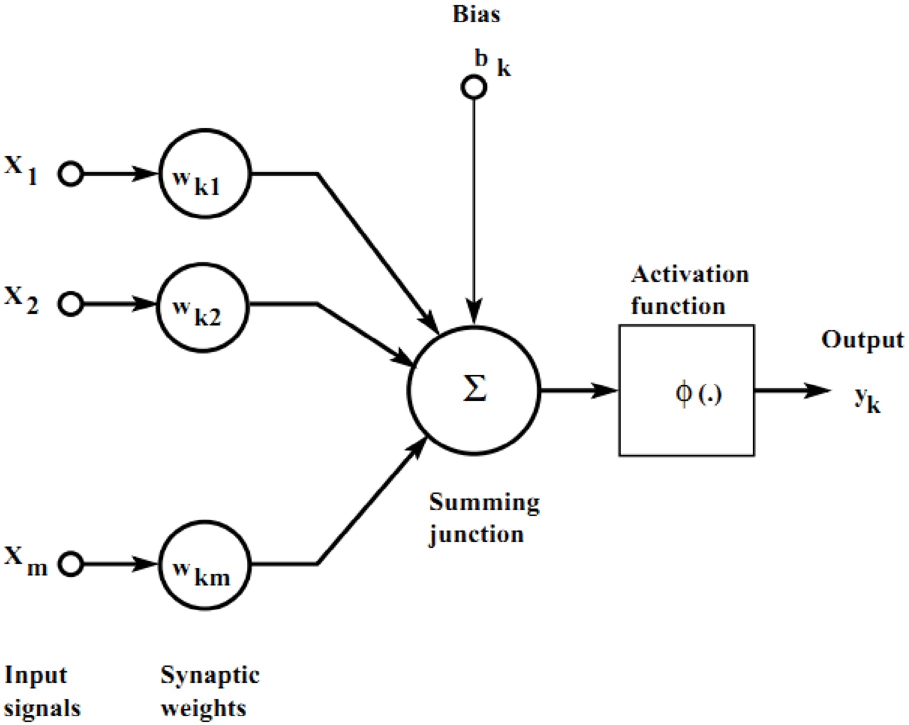
- Neuron Model
  - A set of synapses(weights)
  - Adder

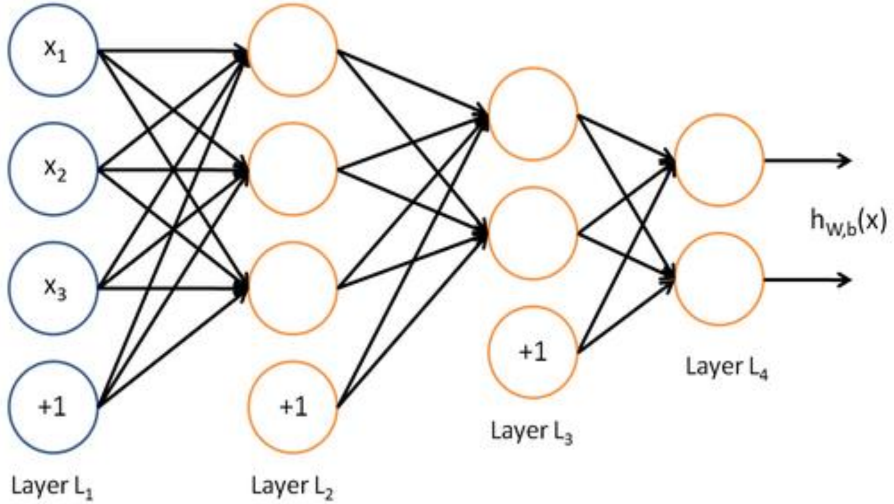
$$z = \sum_{j=1}^{n} w_{kj} x_j + b_k$$

  - Activation function

$$y_k = f(z) = \frac{1}{1 + exp(-z)}$$

- Neural Network Model
  - 4 basic features
  - Common categories
  - Learning Methods





Background - Neural Networks

- 四个基本特征
  - 非线性、非局限性、非常定性、非凸性
- 四种常见的神经网络模型
  - 前馈神经网络: 单/多层感知机
  - 反馈神经网络: Hopfield、Hamming、双向联想存储器(BAM)
  - 自组织神经网络: 自组织映射神经网络(SOM) 、对流神经网络(CPN)
  - 随机神经网络: Boltzmann机
- 五种常见的学习方法
  - 误差修正学习(Error-Correction Learning)
  - 基于记忆的学习(Memory-Based Learning)
  - 赫布型学习(Hebbian Learning)
  - 竞争学习(Competitive Learning)
  - 波尔兹曼学习(Boltzmann Learning)

Outline

- Background
  - Human Auditory System
  - Neural Networks
- Deep Architectures
  - Motivations and Challenges
  - Attempts and Breakthrough
- Deep Learning
  - Autoencoders and Sparse Coding
  - Restricted Boltzmann Machines(RBMs)
  - Greedy Layer-Wise Training
  - Explaination and Justification
- Deep Learning in Audio and Speech
- References

Deep Architectures - Motivations

- Why Deep Architectures?
  - Insufficient depth can hurt
    - Precision and Generalization
    - Neurons -> Memory space
    - Time consuming
  - The brain has a deep architecture
    - such as Auditory System
  - Cognitive processes seem deep:
    - Humans organize their ideas and concepts hierarchically.
    - Humans first learn simpler concepts and then compose them to represent more abstract ones.
    - Engineers break-up solutions into multiple levels of abstraction and processing

Ref: http://www.iro.umontreal.ca/~pift6266/H10/notes/deepintro.html

Deep Architectures - Challenges

- How to train Deep Architectures?
  - Layers
  - Neurons in each Layer
  - Initial Weights/Activation function
  - Learning methods

  These are general problems in NN design
- BUT poor training and generalization errors generally obtained using the standard random initialization.
- gradient-based training of deep supervised multi-layer neural networks (starting from random initialization) gets stuck in “apparent local minima or plateaus(高原，平台)”, and that as the architecture gets deeper, it becomes more difficult to obtain good generalization

Ref: Yoshua Bengio. Learning Deep Architectures for AI. Chap 4.2

Deep Architectures - Attempts

- Deep Convolutional(卷积的) Neural Networks(CNNs)
  - [1]Y. LeCun, etc. “Backpropagation applied to handwritten zip code recognition,” Neural Computation, 1989.
  - [2]Y. Le Cun,etc. “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, 1998.
  - [3]D. Simard, P. Y. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks,” in International Conference on Document Analysis and Recognition (ICDAR’03)
  - [4]M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, “Efficient learning of sparse representations with an energy-based model,” in Advances in Neural Information Processing Systems 19(NIPS’06)

Three architectural ideas to ensure some degree of shift, scale, and distortion(变形，扭曲) invariance: (Ref [2].)

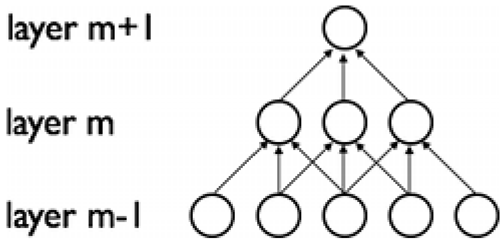
1) local receptive fields;
2) shared weights (or weight replication);
3) spatial or temporal subsampling.

Ref; Yoshua Bengio. Learning Deep Architectures for AI. Chap 4.2

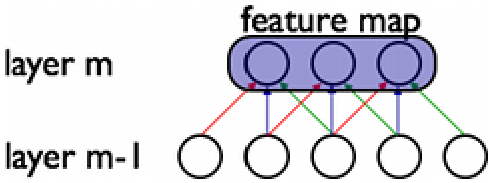
Deep Architectures - Attempts

- Deep Convolutional Neural Networks
  - Sparse Connectivity (a)
  - Shared Weights (b)
  - The Full Model (LeNet, c)

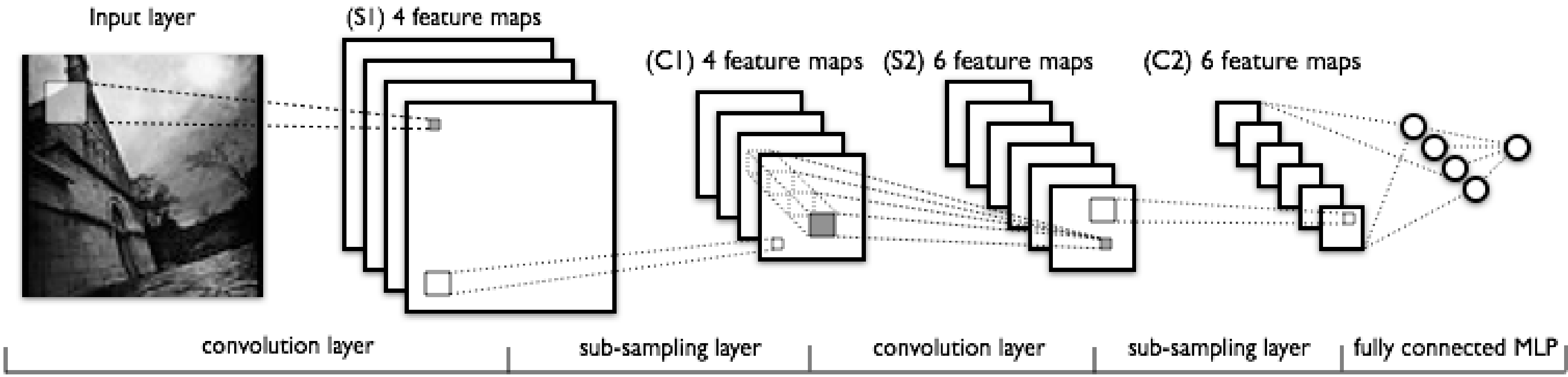
(a)



(b)



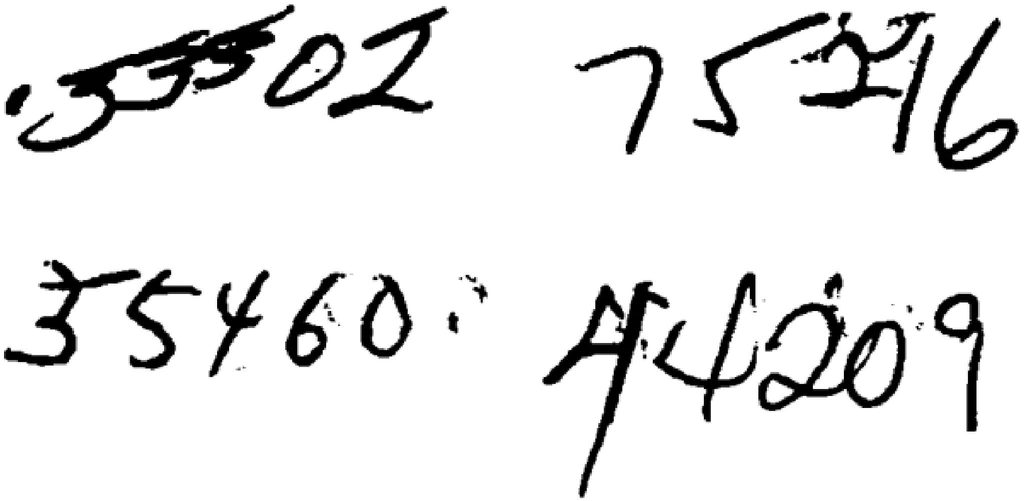
(c)



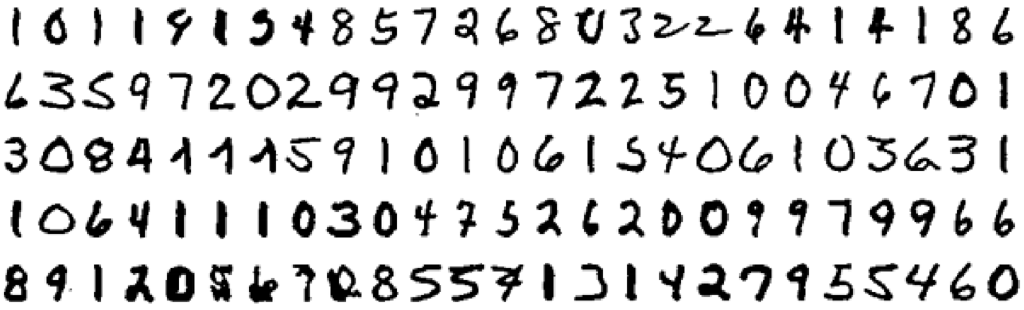
Ref: http://deeplearning.net/tutorial/lenet.html

Deep Architectures - Attempts

- Experiments and Results
  - Data set: 9298 segmented numerals digitized from handwritten zip codes that appeared on U.S. mail passing through the Buffalo, NY post office



original zip codes

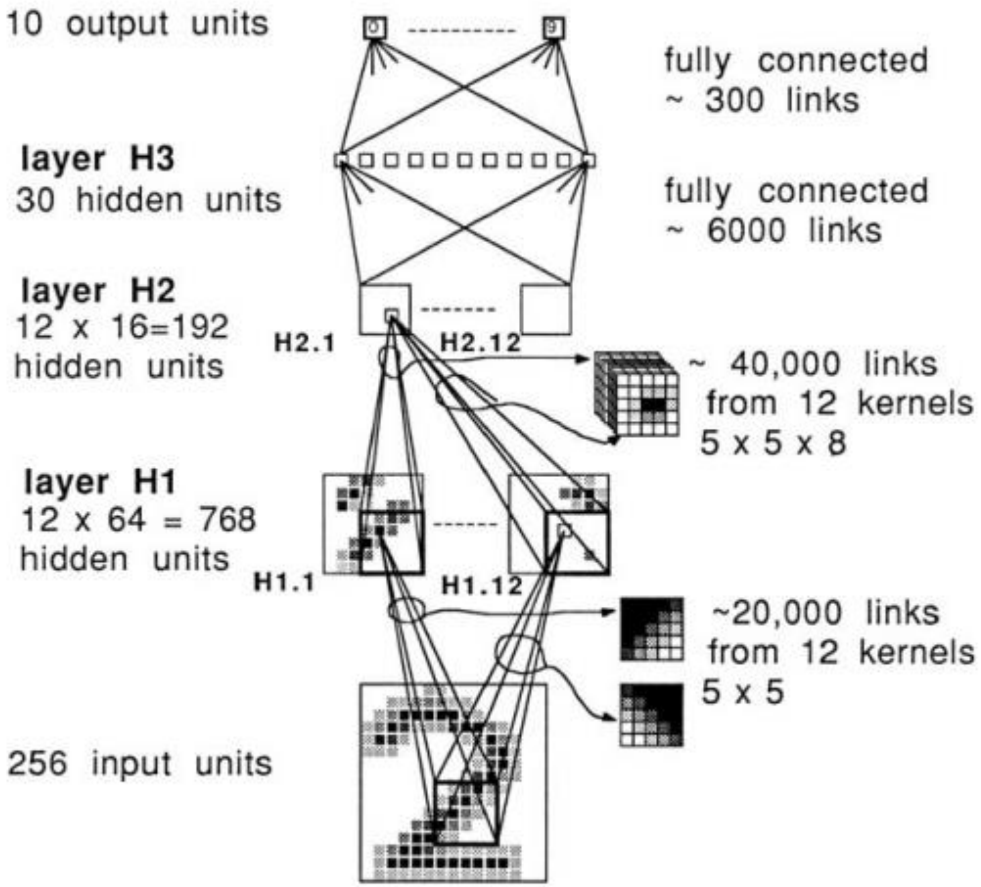


normalized digits

Deep Architectures - Attempts

- Experiments and Results

Network Architecture



Method: Backpropagation and stochastic gradient

Training Error: 0.14%
Test Error: 5.0%

While a fully connected network with one hidden layer of 40 units:
Training Error: 1.6%
Test Error: 8.1%

More Comparisons:
MNIST database

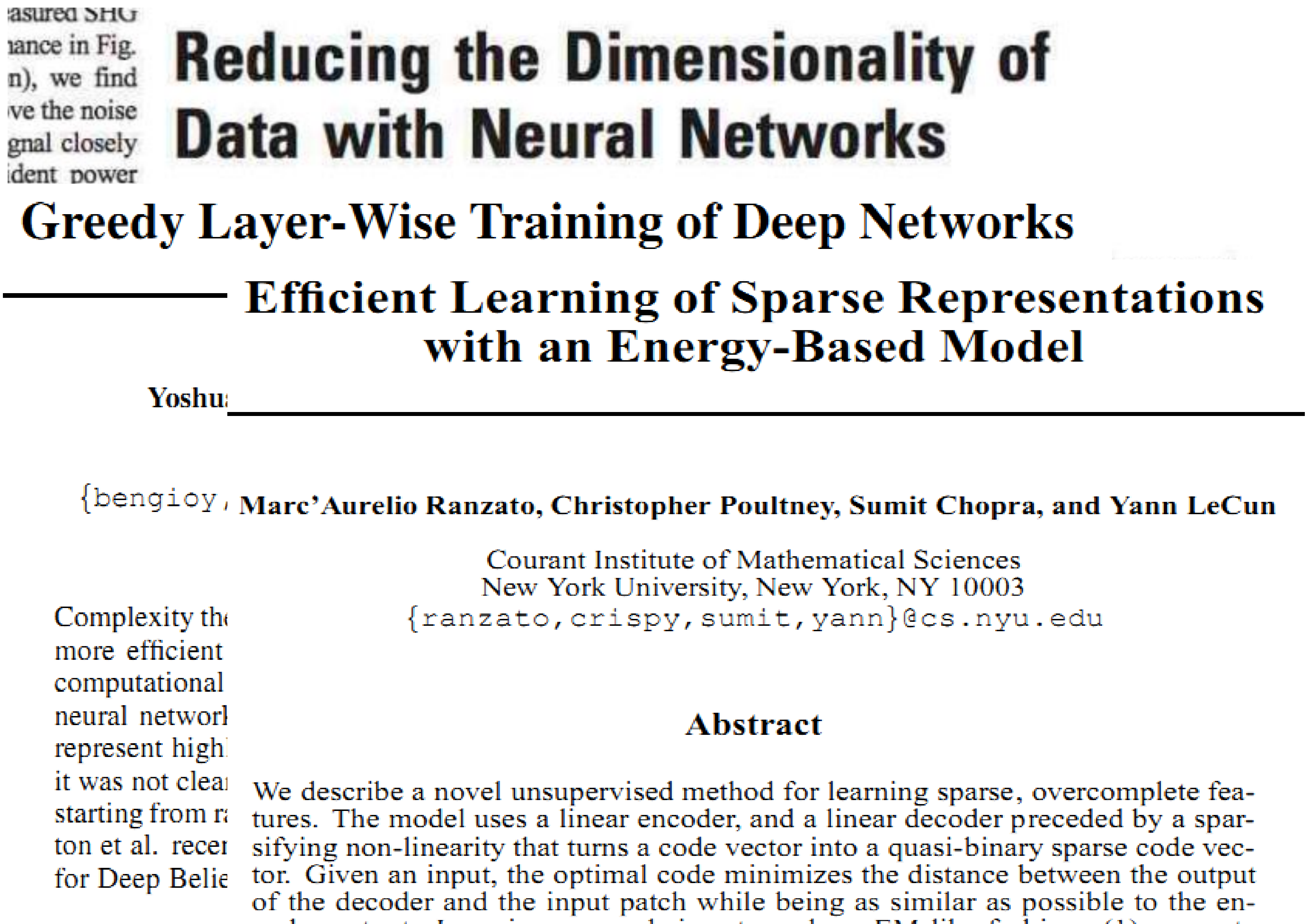
Ref: Y. LeCun etc. Backpropagation Applied to Handwritten Zip Code Recognition

Deep Architectures - Breakthrough

- better results could be achieved when pre-training each layer with an unsupervised learning algorithm, one layer after the other, starting with the first layer
- Breakthrough
  - G. E. Hinton, etc. Reducing the Dimensionality of Data with Neural Networks. Science 2006. (and Hinton, G. E., etc., A fast learning algorithm for deep belief nets. Neural Computation, 2006)
  - Yoshua Bengio,etc. Greedy Layer-Wise Training of Deep Networks. NIPS06
  - Marc'Aurelio Ranzato,Efficient Learning of Sparse Representations with an Energy-Based Model. NIPS06
- key principles in the papers
  - Unsupervised learning of representations is used to (pre-)train each layer.
  - Unsupervised training of one layer at a time, on top of the previously trained ones. The representation learned at each level is the input for the next layer.
  - Use supervised training to fine-tune all the layers (in addition to one or more additional layers that are dedicated to producing predictions).

Ref: http://www.iro.umontreal.ca/~pift6266/H10/notes/deepintro.html

Deep Architectures - Breakthrough



asured SHG
iance in Fig.
n), we find
ve the noise
gnal closely
ident power

Reducing the Dimensionality of Data with Neural Networks

Greedy Layer-Wise Training of Deep Networks

Yoshu

{bengioy,

Complexity th
more efficient
computational
neural networl
represent high
it was not clea
starting from r
ton et al. recei
for Deep Belie

Efficient Learning of Sparse Representations with an Energy-Based Model

Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun

Courant Institute of Mathematical Sciences
New York University, New York, NY 10003
{ranzato,crispy,sumit,yann}@cs.nyu.edu

Abstract

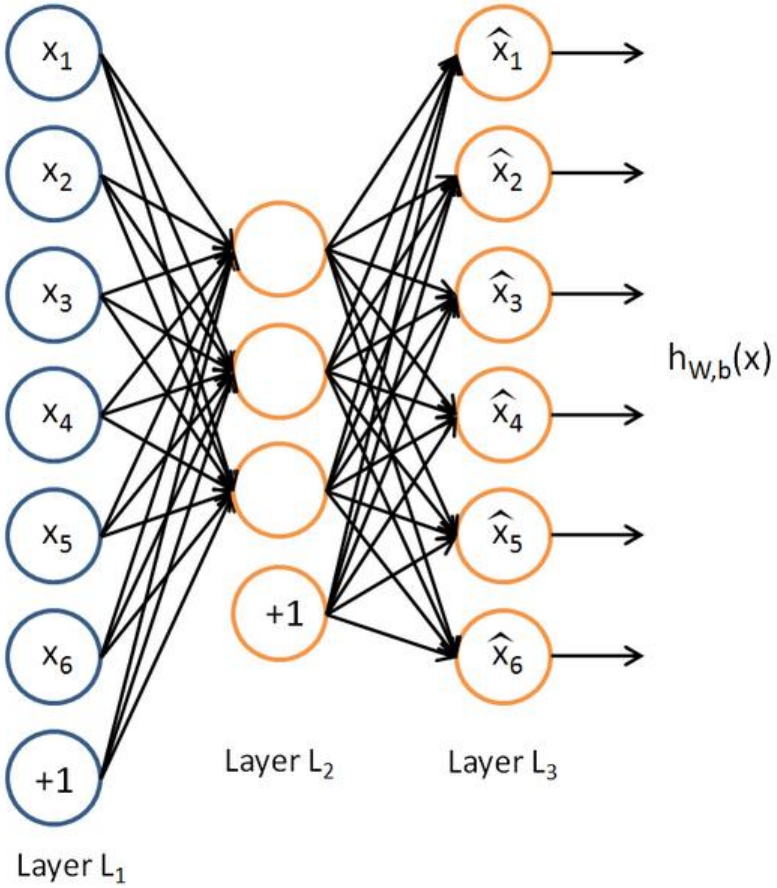
We describe a novel unsupervised method for learning sparse, overcomplete features. The model uses a linear encoder, and a linear decoder preceded by a sparsifying non-linearity that turns a code vector into a quasi-binary sparse code vector. Given an input, the optimal code minimizes the distance between the output of the decoder and the input patch while being as similar as possible to the en-

Outline

- Background
  - Human Auditory System
  - Neural Networks
- Deep Architectures
  - Motivations and Challenges
  - Attempts and Breakthrough
- Deep Learning
  - Autoencoders and Sparse Coding
  - Restricted Boltzmann Machines(RBMs)
  - Greedy Layer-Wise Training
  - Explaination and Justification
- Deep Learning in Audio and Speech
- References

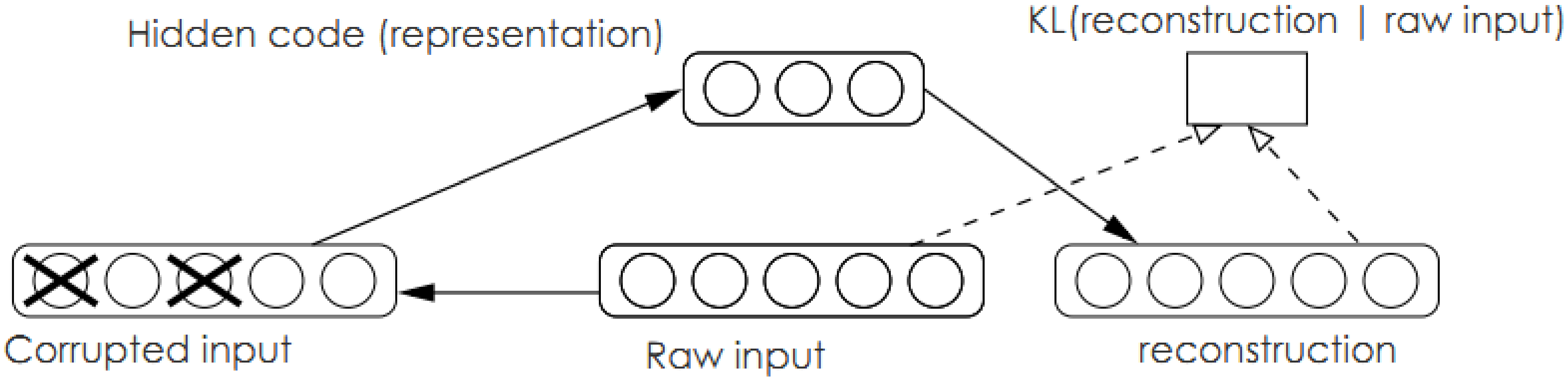
Deep Learning - Autoencoders

- Autoencoders
  - An **autoencoder** neural network is an **unsupervised learning** algorithm that applies backpropagation, setting the target values to be equal to the inputs. i.e., it uses $y^{(i)} = x^{(i)}$
  - The simple autoencoder often ends up learning a low-dimensional representation very **similar to PCA**
  - If the hidden layer is **non-linear**, the auto-encoder behaves **different from PCA**, with the ability to capture multi-modal aspects of the input distribution.
  - The departure becomes even more important when we consider stacking multiple encoders



Deep Learning - Autoencoders

- Denoising Autoencoders
  - In order to force the hidden layer to discover more robust features and prevent it from simply learning the identity, we train the autoencoder to *reconstruct the input from a corrupted (腐蚀的，破坏的) version of it*.
  - The denoising auto-encoder is a stochastic version of the auto-encoder
  - Intuitively, a denoising auto-encoder does two things:
    - **try to encode the input (preserve the information about the input),**
    - **and try to undo the effect of a corruption process stochastically applied to the input of the auto-encoder.**



Ref: http://deeplearning.net/tutorial/dA.html

Deep Learning - Sparse Coding

- Sparse Coding(Olshausen & Field,1996)
  - Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data efficiently.
  - The aim of sparse coding is to find a set of basis vectors such that we can represent an input vector as a linear combination of these basis vectors:

$$\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i \qquad (1)$$

  - Define the sparse coding cost function on a set of *m* input vectors as

$$\text{minimize}_{a_i^{(j)},\phi_i} \sum_{j=1}^{m} \left\| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)}) \qquad (2)$$

where *S*(.) is a sparsity cost function which penalizes(惩罚) a_i for being far from zero.

Deep Learning - Sparse Coding

- Measure of sparsity
  - L_0 norm ($S(a_i) = 1(|a_i| > 0)$): non-differentiable and difficult to optimize
  - L_1 penalty $S(a_i) = |a_i|_1$
  - log penalty $S(a_i) = \log(1 + a_i^2)$
- Generalization and Restriction

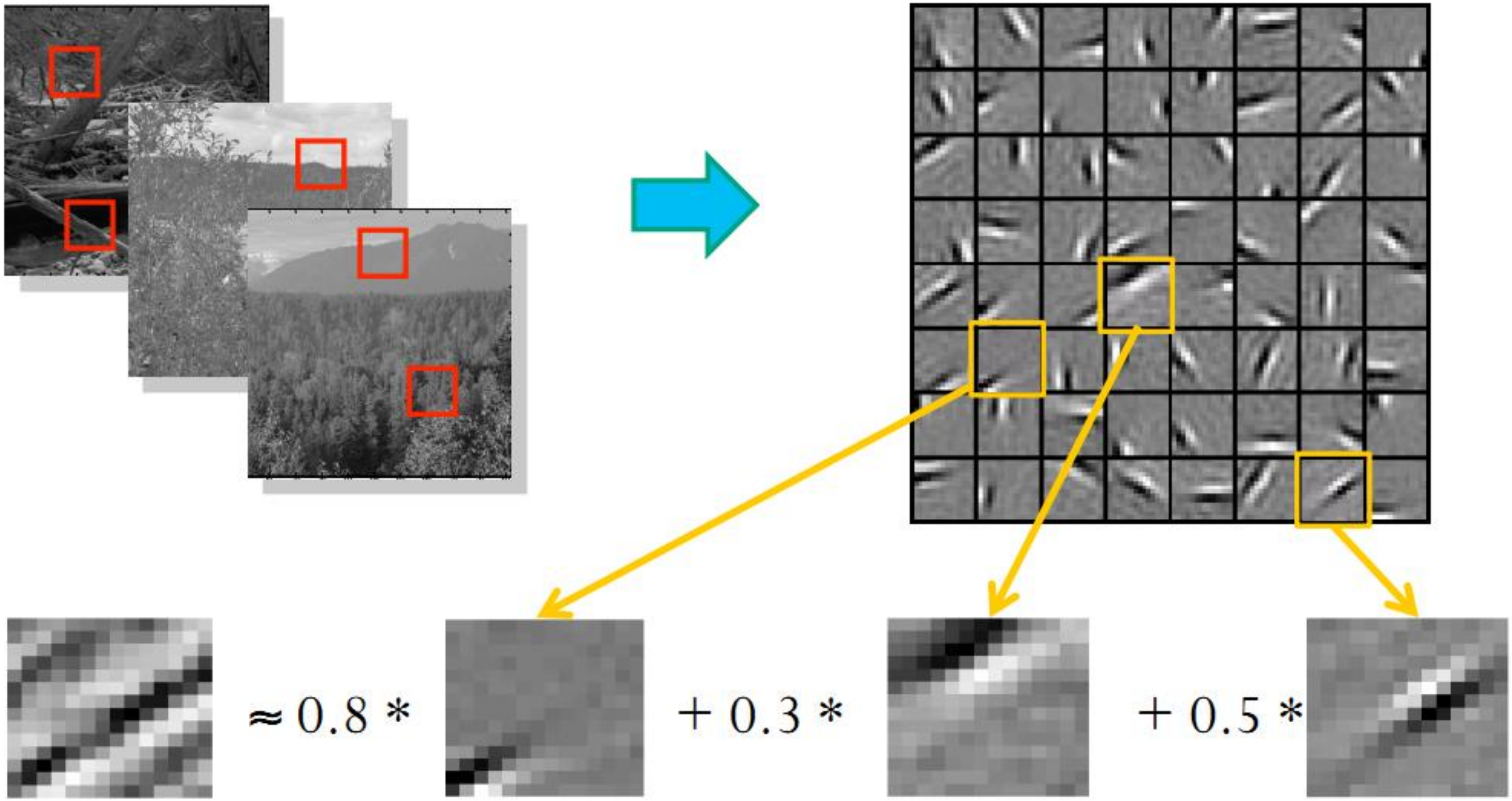
$$\begin{array}{ll} \text{minimize}_{a_i^{(j)},\phi_i} & \sum_{j=1}^{m} \left|\left|\mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)}\phi_i\right|\right|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)}) \\ \text{subject to} & \left|\left|\phi_i\right|\right|^2 \leq C, \forall i = 1, ..., k \end{array} \qquad (3)$$

- Learning
  - L1 penalty: gradient-based methods, such as conjugate(共轭) gradient methods
  - L2 norm penalty: Lagrange dual(拉格朗日对偶)

Ref: http://deeplearning.stanford.edu/wiki/index.php/Sparse_Coding

Deep Learning - Sparse Coding

- Sparse Coding in Vision

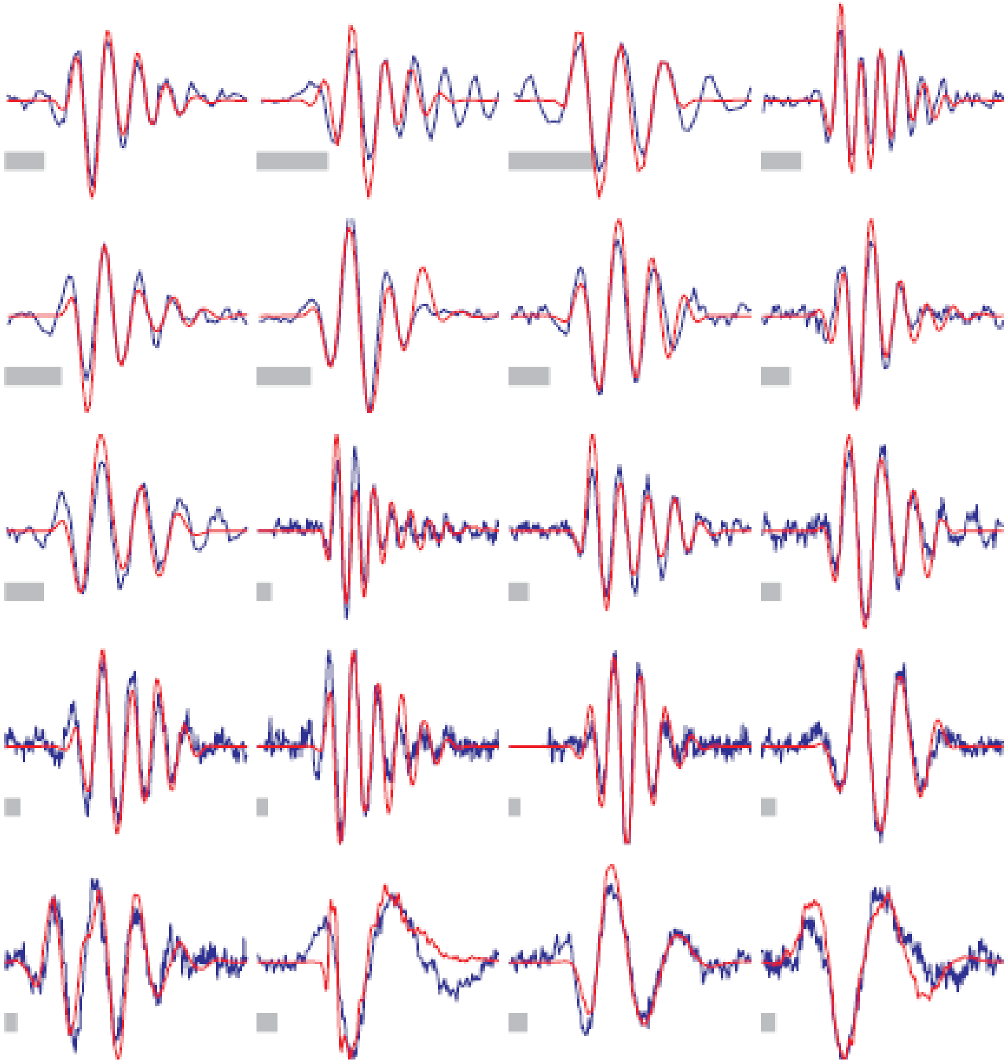


$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$
(feature representation)

Deep Learning - Sparse Coding

- Sparse Coding in Audio

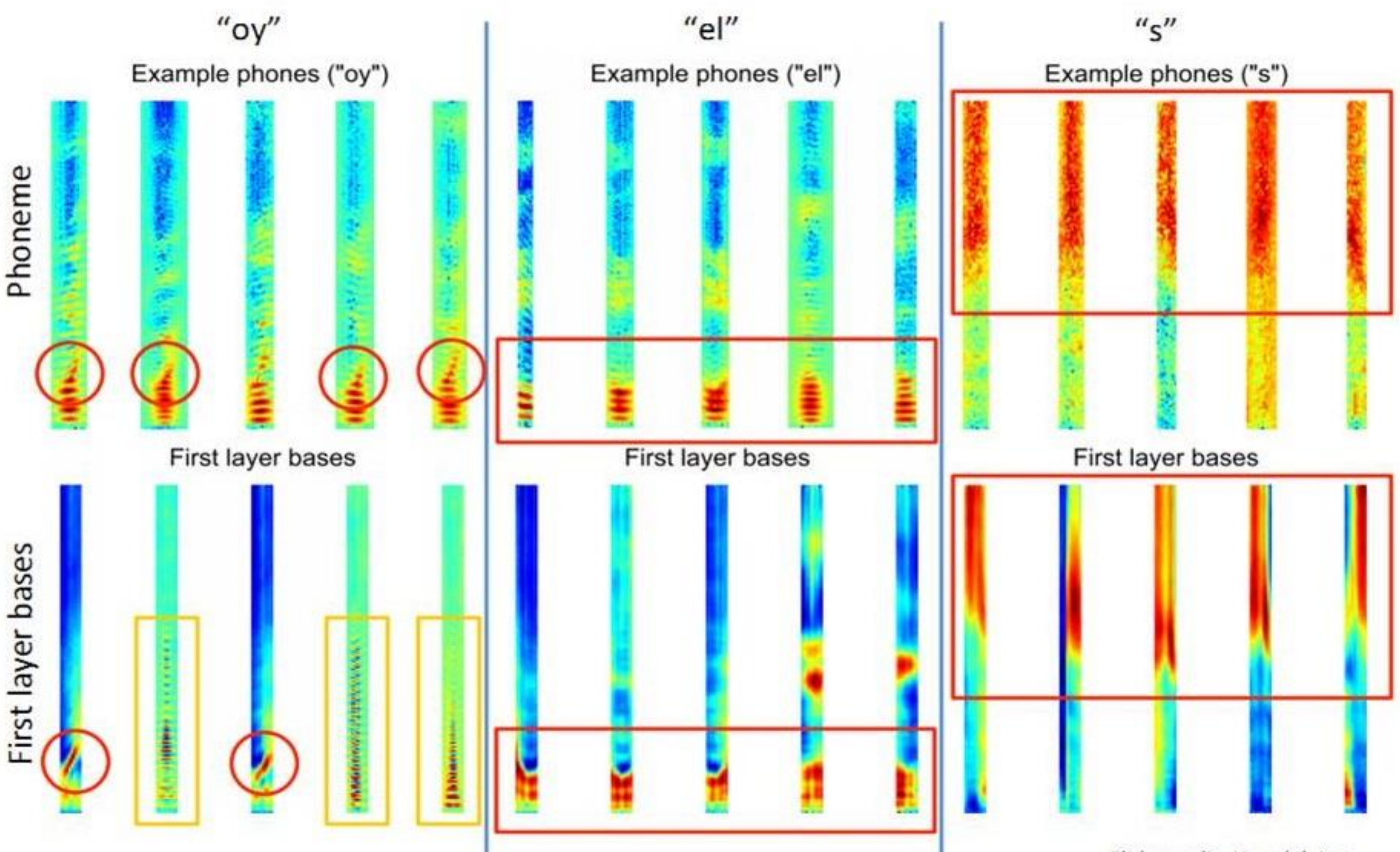
Image shows 20 basis functions learned from unlabeled audio.



[Evan Smith & Mike Lewicki. Efficient Auditory Coding Nature, 2006]

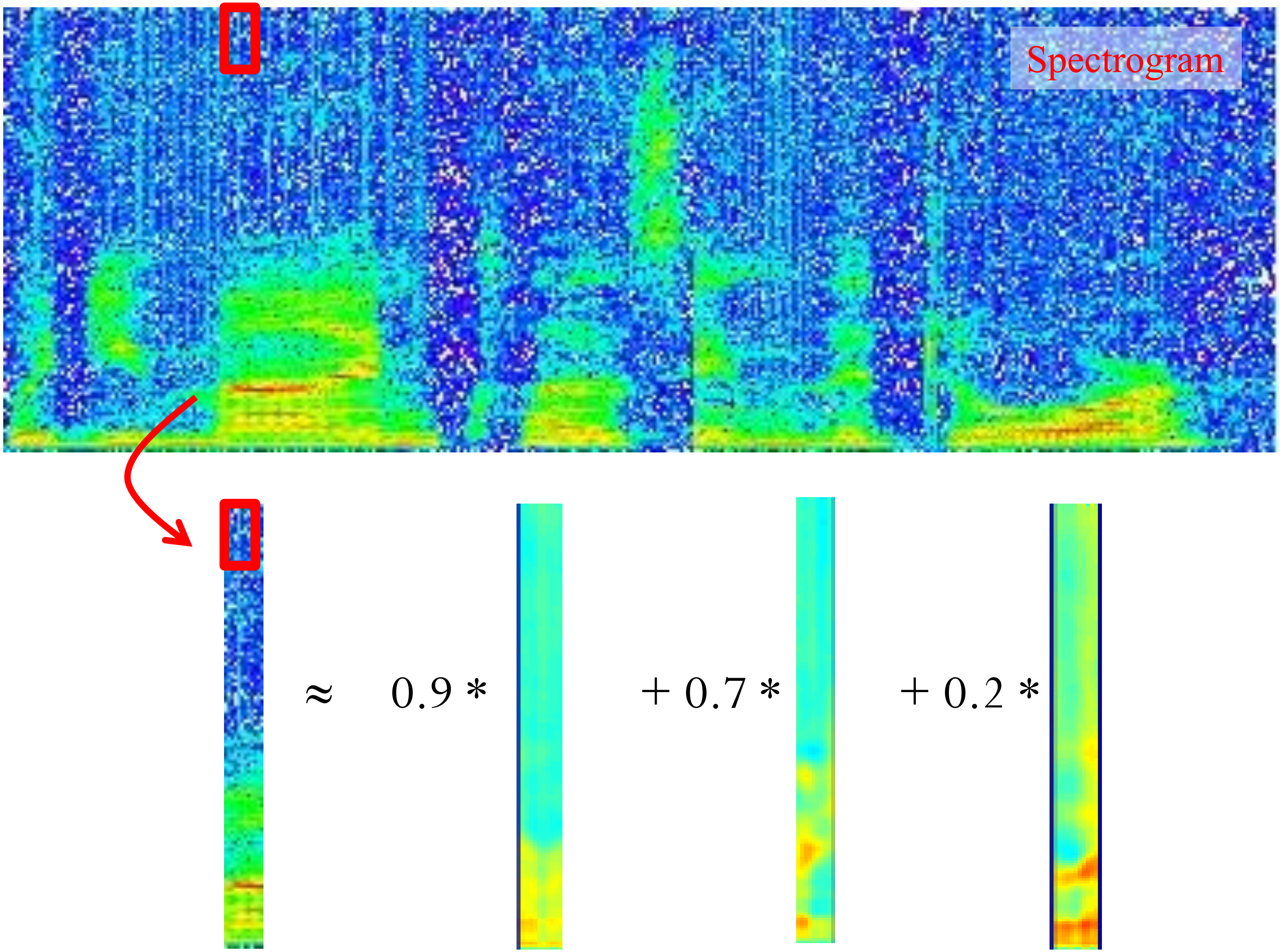
Deep Learning - Sparse Coding

- Comparison of bases to phonemes



Slide credit: Honglak Lee

Deep Learning - Sparse Coding



$$x \approx 0.9 * \phi_{36} + 0.7 * \phi_{42} + 0.2 * \phi_{63}$$

Andrew Ng.

Deep Learning - RBMs

- Energy-Based Models (EBMs)
  - **Energy-based** models associate a scalar energy to each configuration of the variables of interest
  - Learning corresponds to modifying that energy function so that its shape has desirable properties.
  - Energy-based probabilistic models define a probability distribution through an energy function

$$p(x) = \frac{e^{-E(x)}}{Z}. \quad Z = \sum_x e^{-E(x)} \qquad (4)$$

- EBMs with Hidden Units
  - In many cases of interest, we do not observe the example x fully, or we want to introduce some non-observed variables to increase the expressive power of the model. We introduce a **hidden** part $\boldsymbol{h}$

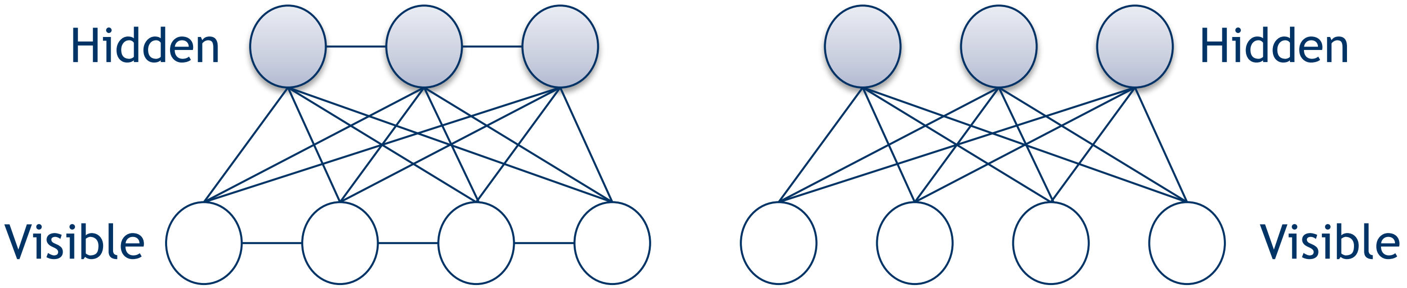
$$P(x) = \sum_h P(x,h) = \sum_h \frac{e^{-E(x,h)}}{Z}. \qquad (5)$$

  - Introduce $\mathcal{F}(x) = -\log \sum_h e^{-E(x,h)}$ then $P(x) = \frac{e^{-\mathcal{F}(x)}}{Z}$ with $Z = \sum_x e^{-\mathcal{F}(x)}$.

Ref: http://deeplearning.net/tutorial/rbm.html

Deep Learning - RBMs

- Restricted Boltzmann Machines(RBMs)
  - **Boltzmann Machines (BMs. Hinton, etc. 1986)** are a particular form of log-linear Markov Random Field (MRF), i.e., for which the energy function is linear in its free parameters.
  - **BMs with hidden variables**: can increase the modeling capacity of the Boltzmann Machine (BM)
  - **Restricted Boltzmann Machines(RBMs)** further restrict BMs to those **without visible-visible and hidden-hidden connections**



BM

RBM

Deep Learning - RBMs

- Restricted Boltzmann Machines(RBMs)
  - 能量函数

$$E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \qquad (6)$$

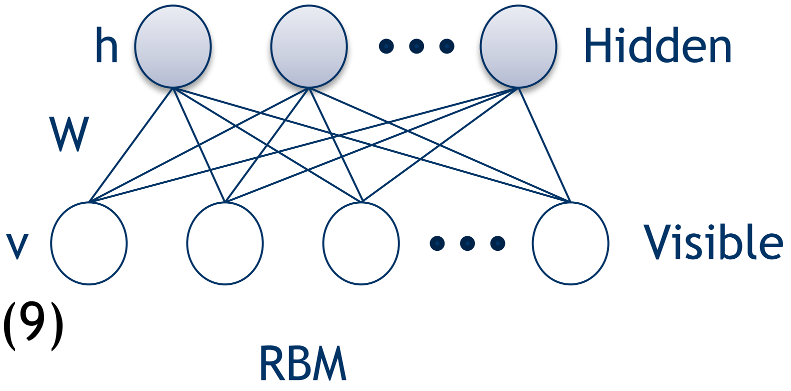
其中 $\boldsymbol{\theta} = \{W_{ij}, a_i, b_j\}$ 为模型参数

  - 从而(v, h)的联合概率分布

$$P(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}) = \frac{e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}, \quad Z(\boldsymbol{\theta}) = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})} \qquad (7)$$

  - 观测数据v的分布 $P(\mathbf{v}|\boldsymbol{\theta})$

$$P(\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})} \qquad (8)$$



  - 第j个隐单元的激活概率

$$P(h_j = 1|\mathbf{v},\boldsymbol{\theta}) = \sigma(b_j + \sum_i v_i W_{ij}) \qquad (9)$$

  - 第i个可见单元的激活概率

$$P(v_i = 1|\mathbf{h},\boldsymbol{\theta}) = \sigma(a_i + \sum_j W_{ij} h_j) \quad \sigma(x) = \frac{1}{1+\exp(-x)} \qquad (10)$$

© Bill Xia, College of Computer Science, Zhejiang University

Deep Learning - RBMs

- Learning RBMs - 模型参数估计
  - 学习RBM的任务是求出参数θ的值, 可以用最大似然法, θ的估计值为

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{t=1}^{T} \log P(\mathbf{v}^{(t)}|\boldsymbol{\theta}) \qquad (11)$$

  - 使用随机梯度上升法(stochastic gradient ascent)求解，关键是求偏导, 首先我们对 $\mathcal{L}(\boldsymbol{\theta})$进行一下变换:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= \sum_{t=1}^{T} \log P(\mathbf{v}^{(t)}|\boldsymbol{\theta}) = \sum_{t=1}^{T} \log \sum_{\mathbf{h}} P(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}) \\
&= \sum_{t=1}^{T} \log \frac{\sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})]}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h})|\boldsymbol{\theta}]} \\
&= \sum_{t=1}^{T} \left(\log \sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})] - \log \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h})|\boldsymbol{\theta}] \right) \qquad (12)
\end{aligned}$$

Deep Learning - RBMs

- Learning RBMs - 模型参数估计

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{t=1}^{T} \frac{\partial}{\partial \theta}\left(\log \sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})] - \log \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})]\right) \\
&= \sum_{t=1}^{T} \left(\sum_{\mathbf{h}} \frac{\exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})]}{\sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})]} \times \frac{\partial(-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right. \\
&\quad \left. - \sum_{\mathbf{v}} \sum_{\mathbf{h}} \frac{\exp[-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})]}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})]} \times \frac{\partial(-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right) \\
&= \sum_{t=1}^{T} \left(\left\langle \frac{\partial(-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right\rangle_{P(\mathbf{h}|\mathbf{v}^{(t)}, \boldsymbol{\theta})} - \left\langle \frac{\partial(-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right\rangle_{P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})} \right)
\end{aligned}
\tag{13}
$$

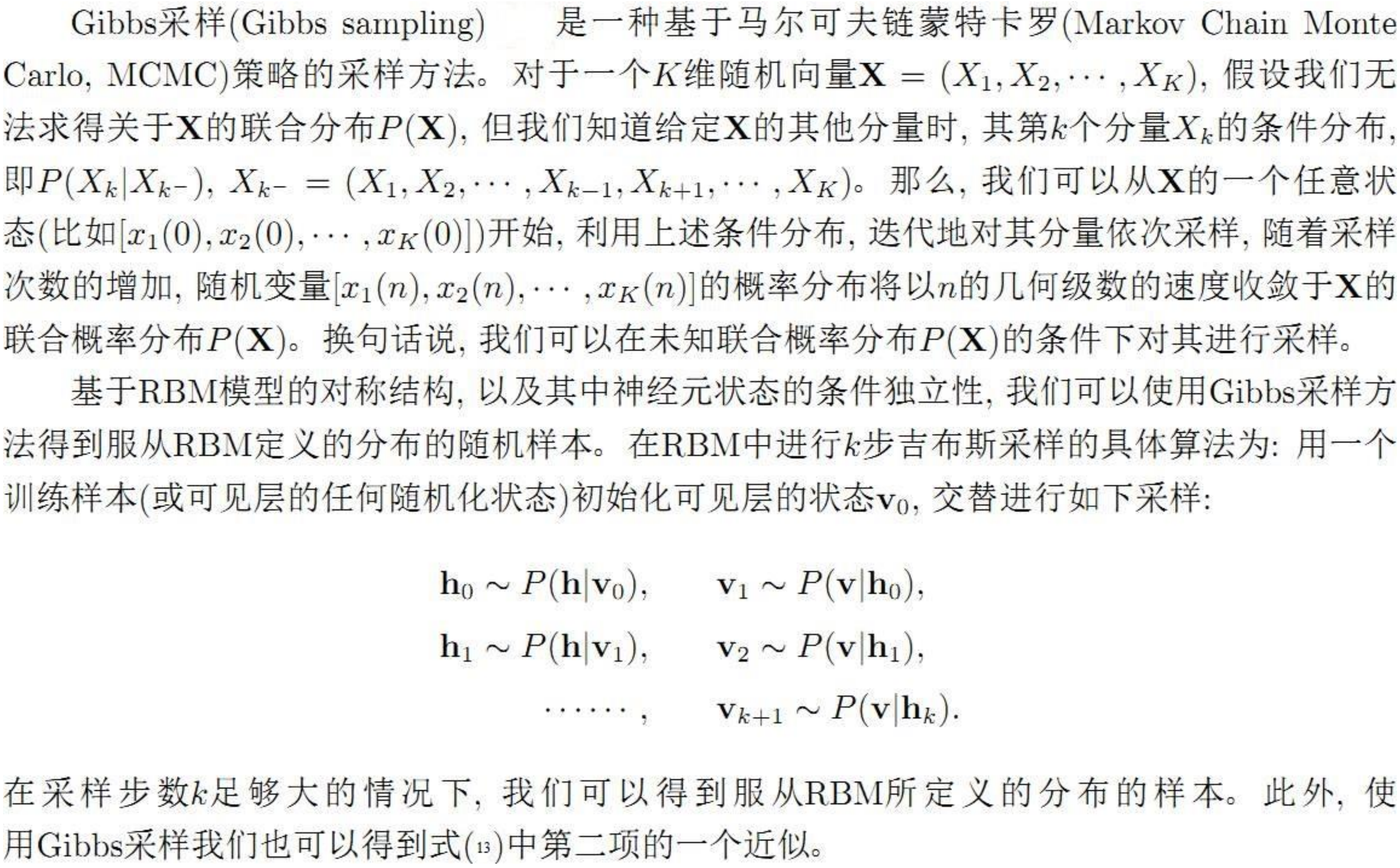
其中, $\langle \cdot \rangle_P$ 表示求关于分布 P 的数学期望

- 用 “data” 和 “model” 来简记 $P(\mathbf{h}|\mathbf{v}^{(t)}, \boldsymbol{\theta})$ 和 $P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})$, 则

$$
\begin{aligned}
\frac{\partial \log P(\mathbf{v}|\boldsymbol{\theta})}{\partial W_{ij}} &= \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}, \\
\frac{\partial \log P(\mathbf{v}|\boldsymbol{\theta})}{\partial a_i} &= \langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}, \\
\frac{\partial \log P(\mathbf{v}|\boldsymbol{\theta})}{\partial b_j} &= \langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}.
\end{aligned}
\tag{14}
$$

Deep Learning - RBMs

- Learning RBMs – Gibbs采样



Gibbs采样(Gibbs sampling) 是一种基于马尔可夫链蒙特卡罗(Markov Chain Monte Carlo, MCMC)策略的采样方法。对于一个K维随机向量$\mathbf{X} = (X_1, X_2, \cdots, X_K)$，假设我们无法求得关于$\mathbf{X}$的联合分布$P(\mathbf{X})$，但我们知道给定$\mathbf{X}$的其他分量时，其第$k$个分量$X_k$的条件分布，即$P(X_k|X_{k-})$，$X_{k-} = (X_1, X_2, \cdots, X_{k-1}, X_{k+1}, \cdots, X_K)$。那么，我们可以从$\mathbf{X}$的一个任意状态(比如$[x_1(0), x_2(0), \cdots, x_K(0)]$)开始，利用上述条件分布，迭代地对其分量依次采样，随着采样次数的增加，随机变量$[x_1(n), x_2(n), \cdots, x_K(n)]$的概率分布将以$n$的几何级数的速度收敛于$\mathbf{X}$的联合概率分布$P(\mathbf{X})$。换句话说，我们可以在未知联合概率分布$P(\mathbf{X})$的条件下对其进行采样。

基于RBM模型的对称结构，以及其中神经元状态的条件独立性，我们可以使用Gibbs采样方法得到服从RBM定义的分布的随机样本。在RBM中进行k步吉布斯采样的具体算法为：用一个训练样本(或可见层的任何随机化状态)初始化可见层的状态$\mathbf{v}_0$，交替进行如下采样：

$$\begin{aligned}
\mathbf{h}_0 &\sim P(\mathbf{h}|\mathbf{v}_0), & \mathbf{v}_1 &\sim P(\mathbf{v}|\mathbf{h}_0), \\
\mathbf{h}_1 &\sim P(\mathbf{h}|\mathbf{v}_1), & \mathbf{v}_2 &\sim P(\mathbf{v}|\mathbf{h}_1), \\
&\cdots\cdots, & \mathbf{v}_{k+1} &\sim P(\mathbf{v}|\mathbf{h}_k).
\end{aligned}$$

在采样步数k足够大的情况下，我们可以得到服从RBM所定义的分布的样本。此外，使用Gibbs采样我们也可以得到式(13)中第二项的一个近似。

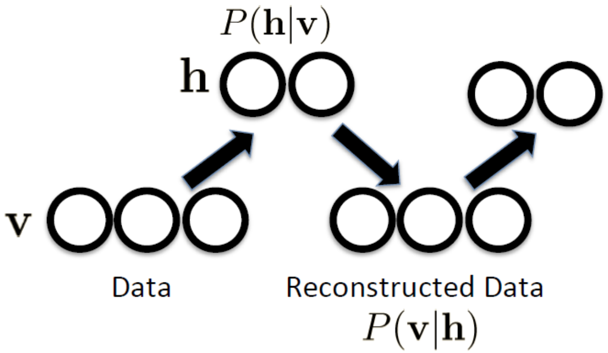
Deep Learning - RBMs

- Learning RBMs – Contrastive Divergence(对比散度, CD)
  - 尽管利用吉布斯采样我们可以得到对数似然函数关于未知参数梯度的近似, 但通常情况下需要使用较大的采样步数, 这使得RBM的训练效率仍旧不高,尤其是当观测数据的特征维数较高时
  - 02年，Hinton提出了RBM的一个快速学习算法，即对比散度，他指出当使用训练数据初始化v_0时,我们仅需要使用k(通常k=1)步吉布斯采样便可以得到足够好的近似
  - 在CD算法一开始, 可见单元的状态被设置成一个训练样本, 并利用式(9)计算所有隐层单元的二值状态。在所有隐层单元的状态确定之后, 根据式(10)来确定第i个可见单元v_i取值为1的概率, 进而产生可见层的一个重构(reconstruction)
  - 各参数的更新准则为

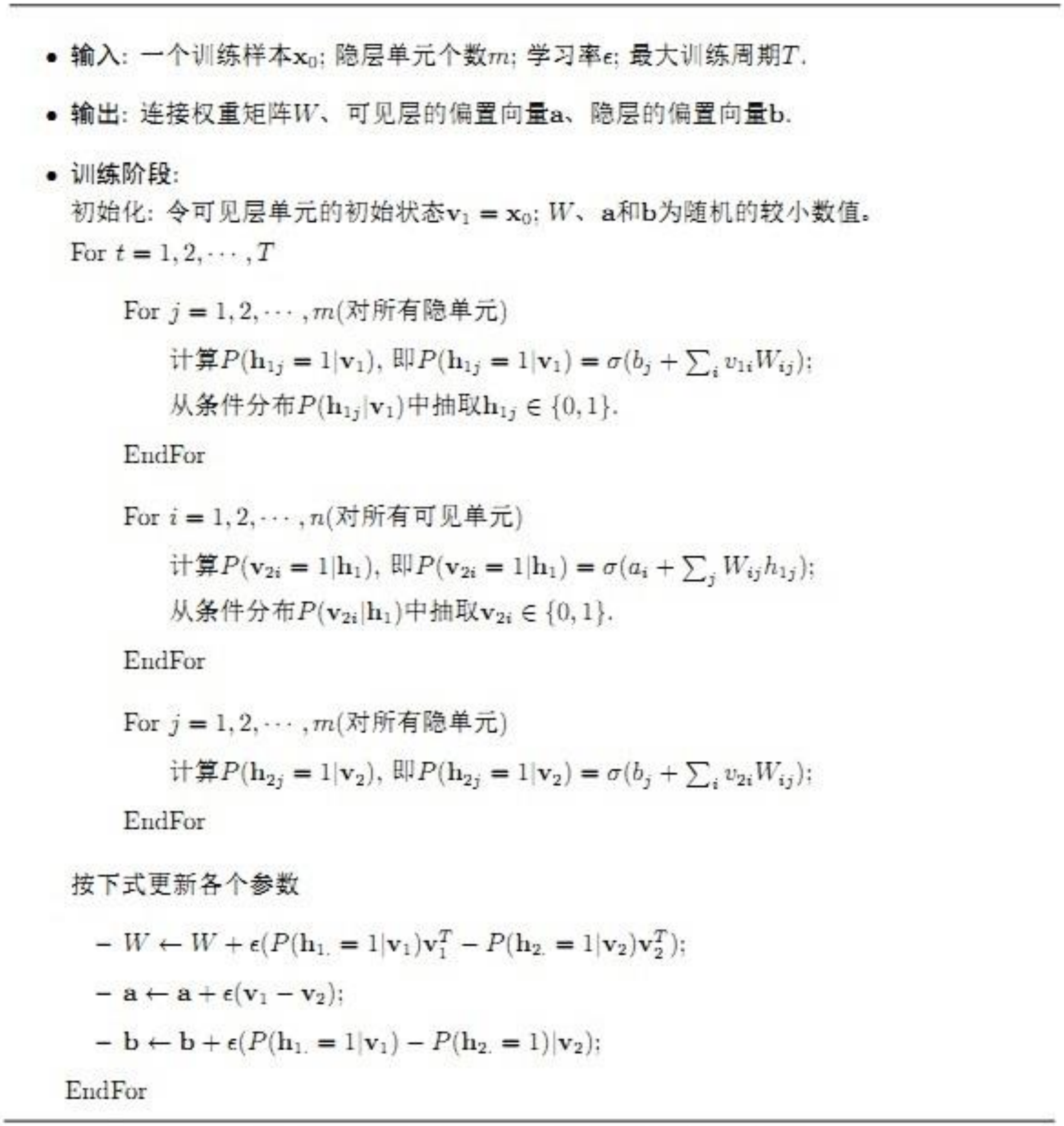
$$\Delta W_{ij} = \epsilon(\langle v_i h_j \rangle_{\mathrm{data}} - \langle v_i h_j \rangle_{\mathrm{recon}}),$$

$$\Delta a_i = \epsilon(\langle v_i \rangle_{\mathrm{data}} - \langle v_i \rangle_{\mathrm{recon}}),$$

$$\Delta b_j = \epsilon(\langle h_j \rangle_{\mathrm{data}} - \langle h_j \rangle_{\mathrm{recon}}), \qquad (15)$$



其中 ϵ是学习速率， $\langle\cdot\rangle_{\mathrm{recon}}$表示一步重构后模型定义的分布



- **输入**: 一个训练样本$\mathbf{x}_0$; 隐层单元个数m; 学习率ϵ; 最大训练周期T.
- **输出**: 连接权重矩阵W、可见层的偏置向量$\mathbf{a}$、隐层的偏置向量$\mathbf{b}$.
- **训练阶段**:

```
初始化: 令可见层单元的初始状态$\mathbf{v}_1 = \mathbf{x}_0$; $W$、$\mathbf{a}$和$\mathbf{b}$为随机的较小数值。
For $t = 1, 2, \cdots, T$
    For $j = 1, 2, \cdots, m$(对所有隐单元)
        计算$P(\mathbf{h}_{1j} = 1|\mathbf{v}_1)$, 即$P(\mathbf{h}_{1j} = 1|\mathbf{v}_1) = \sigma(b_j + \sum_i v_{1i}W_{ij})$;
        从条件分布$P(\mathbf{h}_{1j}|\mathbf{v}_1)$中抽取$\mathbf{h}_{1j} \in \{0, 1\}$.
    EndFor
    For $i = 1, 2, \cdots, n$(对所有可见单元)
        计算$P(\mathbf{v}_{2i} = 1|\mathbf{h}_1)$, 即$P(\mathbf{v}_{2i} = 1|\mathbf{h}_1) = \sigma(a_i + \sum_j W_{ij}h_{1j})$;
        从条件分布$P(\mathbf{v}_{2i}|\mathbf{h}_1)$中抽取$\mathbf{v}_{2i} \in \{0, 1\}$.
    EndFor
    For $j = 1, 2, \cdots, m$(对所有隐单元)
        计算$P(\mathbf{h}_{2j} = 1|\mathbf{v}_2)$, 即$P(\mathbf{h}_{2j} = 1|\mathbf{v}_2) = \sigma(b_j + \sum_i v_{2i}W_{ij})$;
    EndFor
    按下式更新各个参数
      – $W \leftarrow W + \epsilon(P(\mathbf{h}_{1.} = 1|\mathbf{v}_1)\mathbf{v}_1^T - P(\mathbf{h}_{2.} = 1|\mathbf{v}_2)\mathbf{v}_2^T)$;
      – $\mathbf{a} \leftarrow \mathbf{a} + \epsilon(\mathbf{v}_1 - \mathbf{v}_2)$;
      – $\mathbf{b} \leftarrow \mathbf{b} + \epsilon(P(\mathbf{h}_{1.} = 1|\mathbf{v}_1) - P(\mathbf{h}_{2.} = 1)|\mathbf{v}_2)$;
EndFor
```

RBM的基于CD的快速学习算法主要步骤.

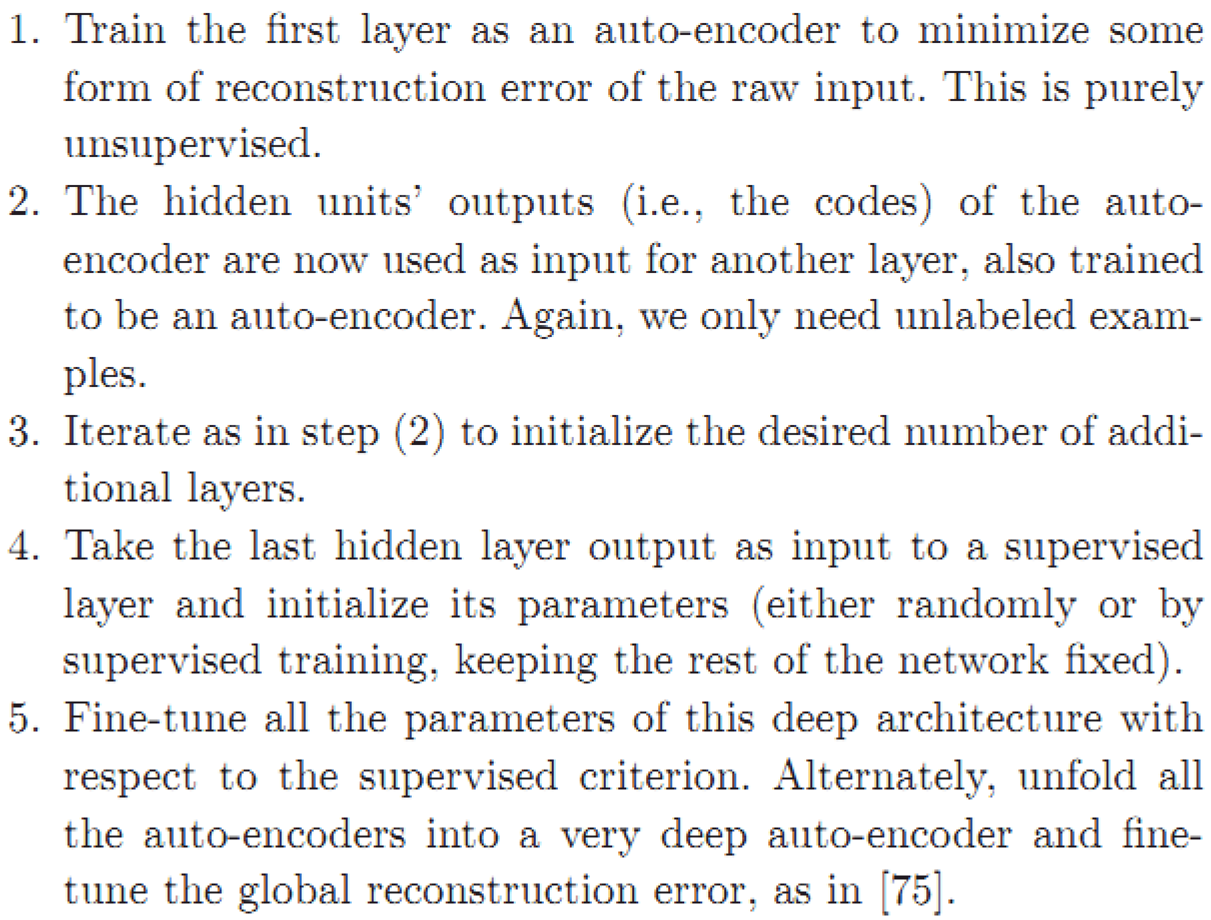
Deep Learning - RBMs

- Learning RBMs – CD
  - RBM的参数设置
    - 小批量数据及其容量
    - 学习率
    - 权重和偏置的初始值
    - 动量学习率
    - 权衰减
    - 隐单元个数
  - RBM的评估算法
    - 重构误差
    - 退火式重要性采样
  - 基本RBM模型的变形算法
    - 稀疏RBM
    - 稀疏组RBM
    - 分类RBM
    - 条件RBM等

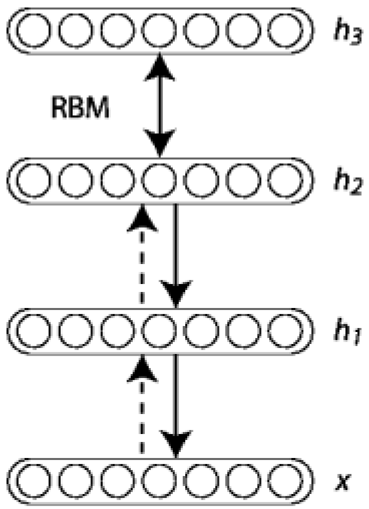
Ref:张春霞，姬楠楠，王冠伟. 受限波尔兹曼机简介.

Deep Learning - Training

- Greedy Layer-Wise Training
  - RBMs can be stacked(堆放，堆叠) and trained in a greedy manner to form so-called Deep Belief Networks (DBN)

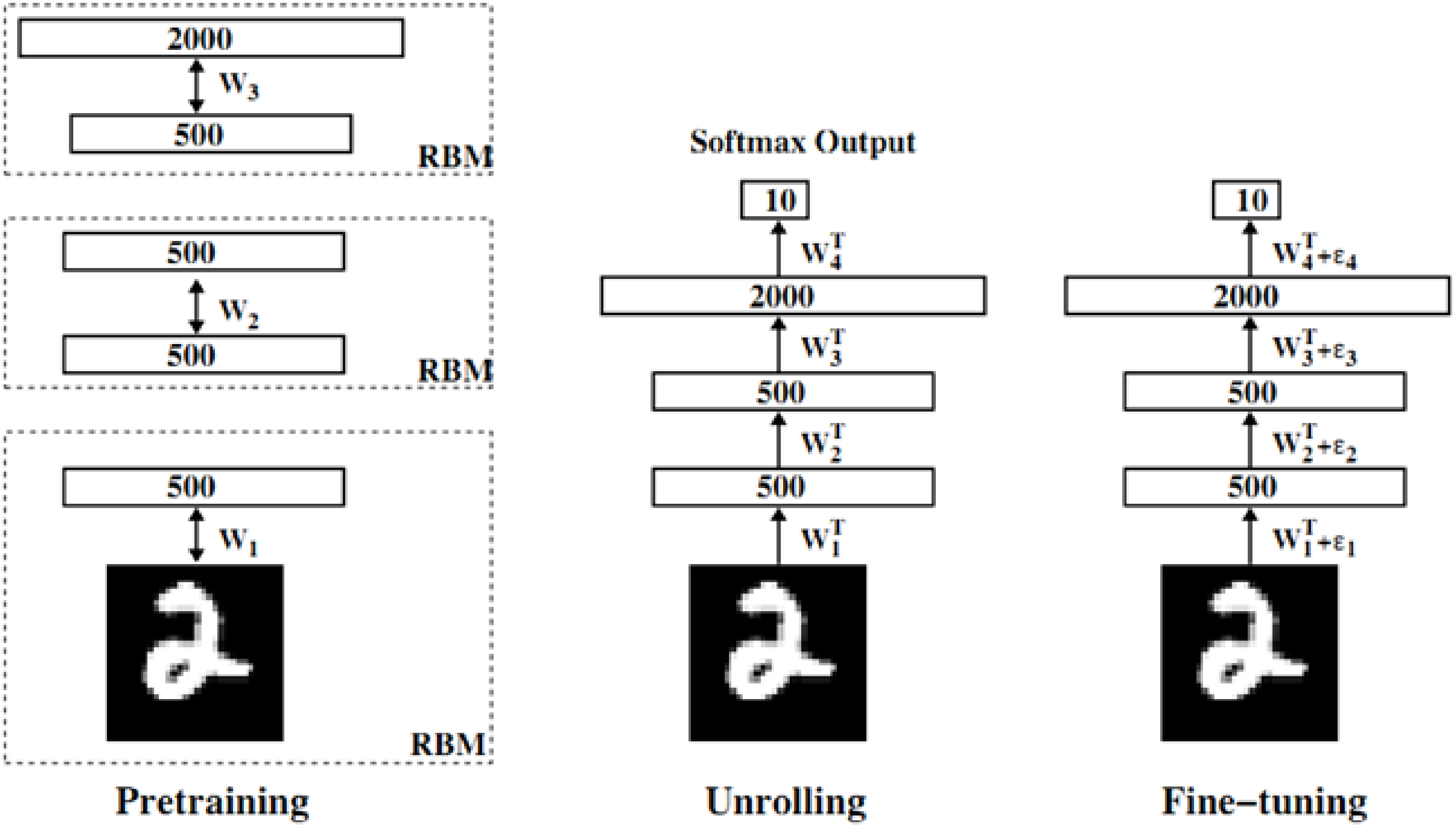


1. Train the first layer as an auto-encoder to minimize some form of reconstruction error of the raw input. This is purely unsupervised.
2. The hidden units' outputs (i.e., the codes) of the auto-encoder are now used as input for another layer, also trained to be an auto-encoder. Again, we only need unlabeled examples.
3. Iterate as in step (2) to initialize the desired number of additional layers.
4. Take the last hidden layer output as input to a supervised layer and initialize its parameters (either randomly or by supervised training, keeping the rest of the network fixed).
5. Fine-tune all the parameters of this deep architecture with respect to the supervised criterion. Alternately, unfold all the auto-encoders into a very deep auto-encoder and fine-tune the global reconstruction error, as in [75].



Deep Learning - Training

- Greedy Layer-Wise Training

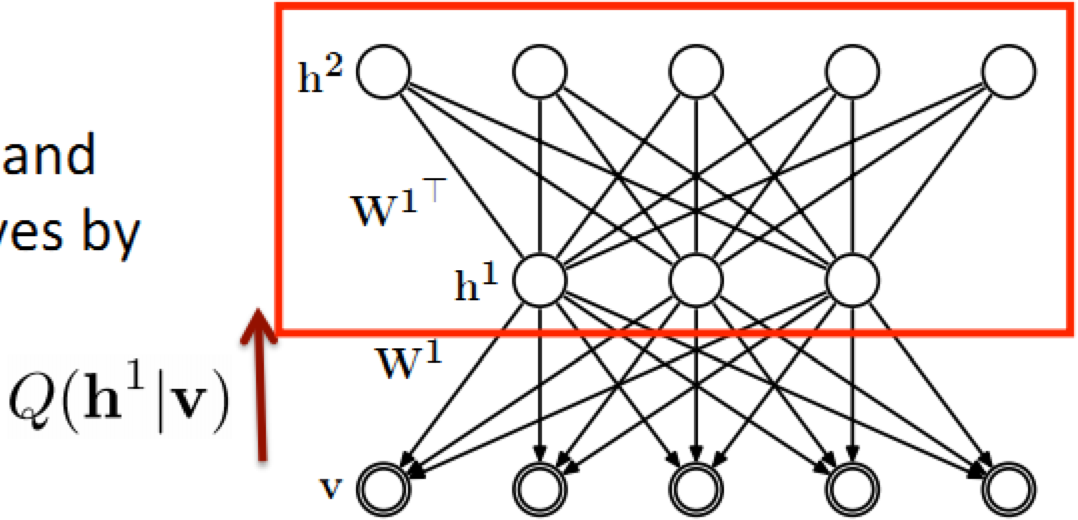


- After layer-by-layer **unsupervised pretraining**, discriminative fine-tuning by backpropagation achieves an error rate of 1.2% on MNIST. SVM's get 1.4% and randomly initialized backprop gets 1.6%.

Deep Learning - Explaination

- Why this Pre-training works?

- Greedy training improves variational lower bound.
- RBM and 2-layer DBN are equivalent when $W^2 = {W^1}^\top$.
- The lower bound is tight and the log-likelihood improves by greedy training.
- For any approximating distribution $Q(\mathbf{h}^1|\mathbf{v})$



$$\log P_\theta(\mathbf{v}) = \sum_{\mathbf{h}^1} P_\theta(\mathbf{v}, \mathbf{h}^1)$$

Train 2nd-layer RBM

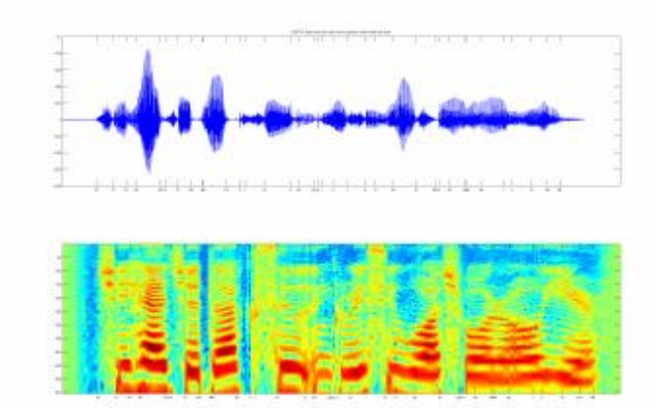
$$\geq \sum_{\mathbf{h}^1} Q(\mathbf{h}^1|\mathbf{v}) \Big[\log P(\mathbf{h}^1) + \log P(\mathbf{v}|\mathbf{h}^1) \Big] + \mathcal{H}(Q(\mathbf{h}^1|\mathbf{v}))$$

Outline

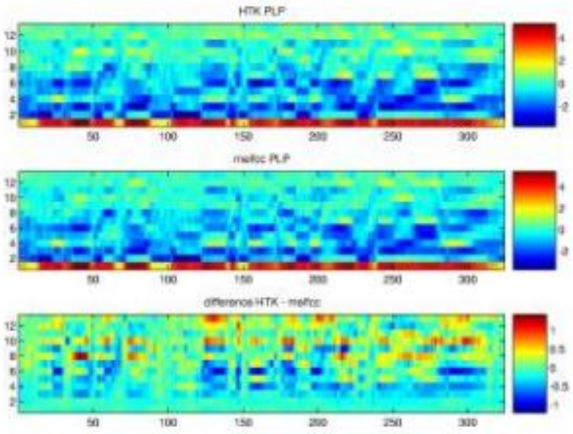
- Background
  - Human Auditory System
  - Neural Networks
- Deep Architectures
  - Motivations and Challenges
  - Attempts and Breakthrough
- Deep Learning
  - Autoencoders and Sparse Coding
  - Restricted Boltzmann Machines(RBMs)
  - Greedy Layer-Wise Training
  - Explaination and Justification
- Deep Learning in Audio and Speech
- References

Deep Learning in Audio and Speech

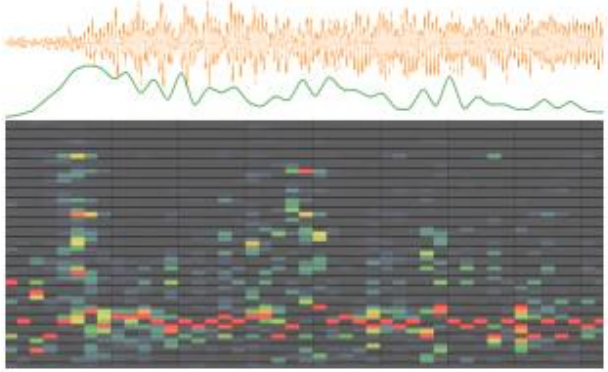
Audio features



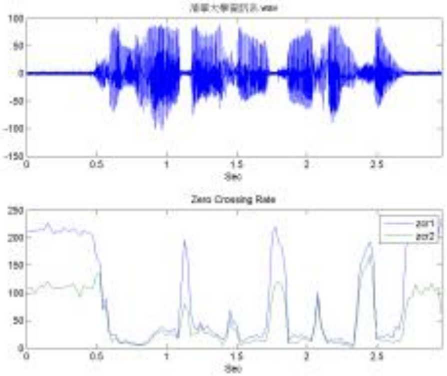
Spectrogram



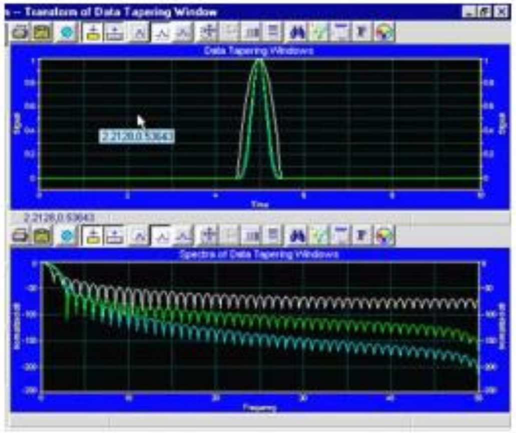
MFCC



Flux



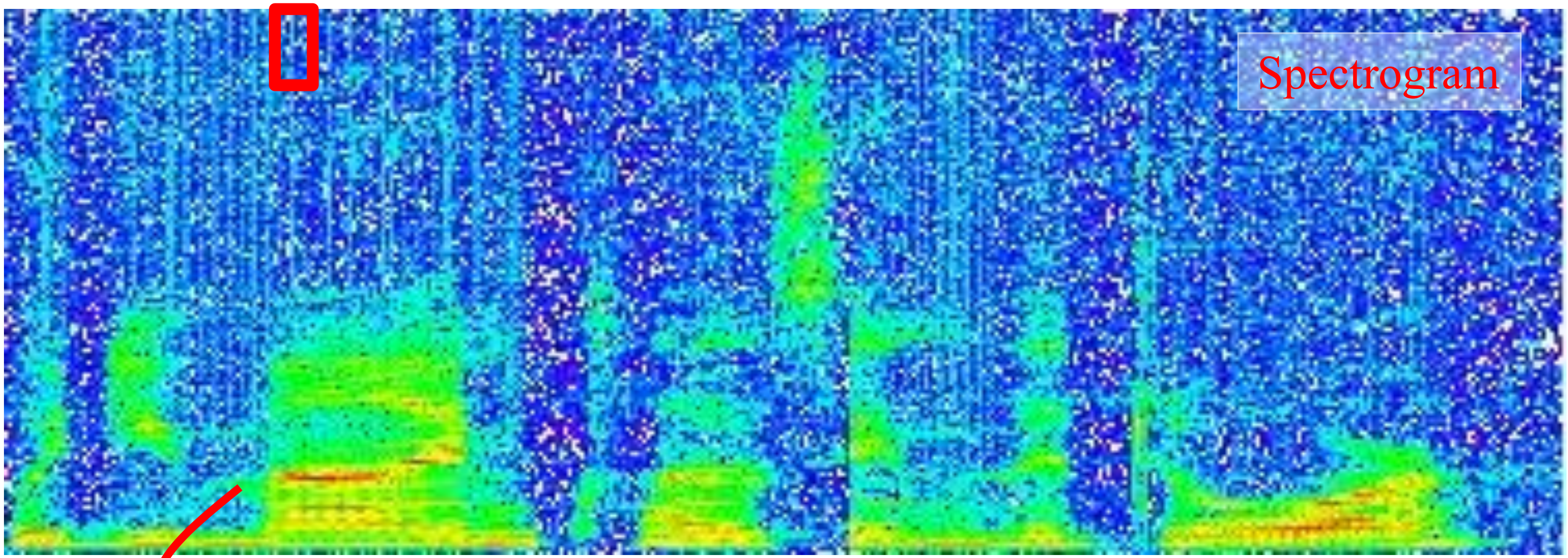
ZCR

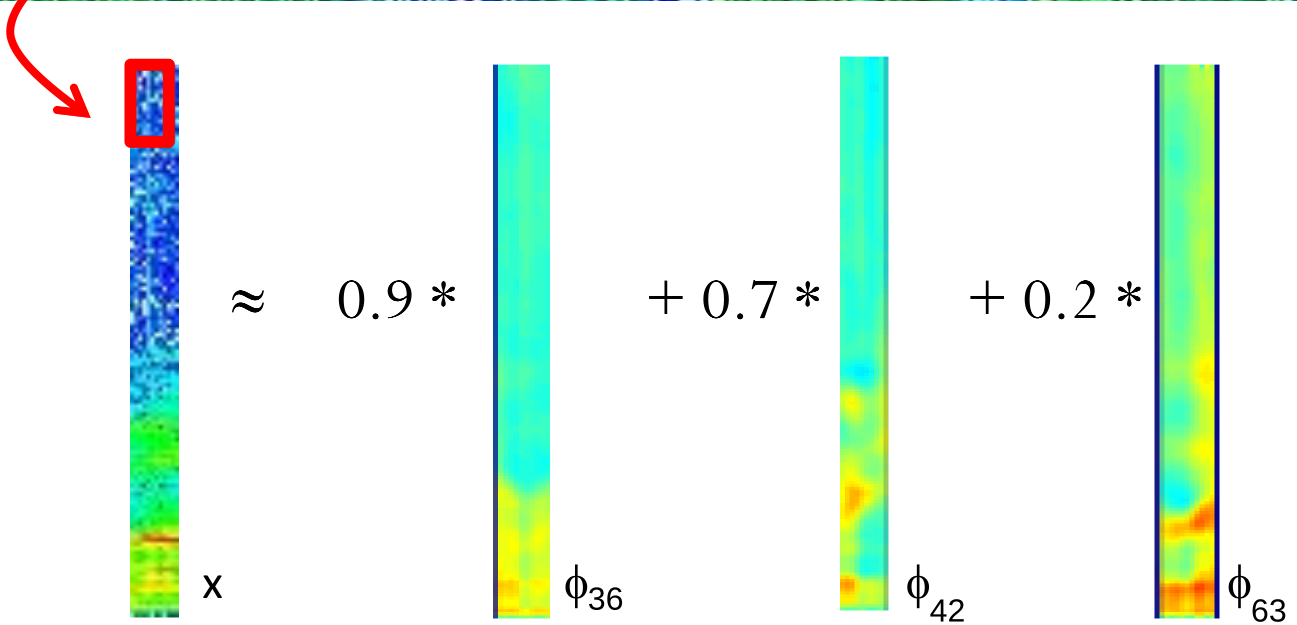

Transform of Data Tapering Window
Data Tapering Windows
Time
Spectra of Data Tapering Windows
Frequency

Rolloff

Deep Learning in Audio and Speech

- Sparse coding on audio (speech)



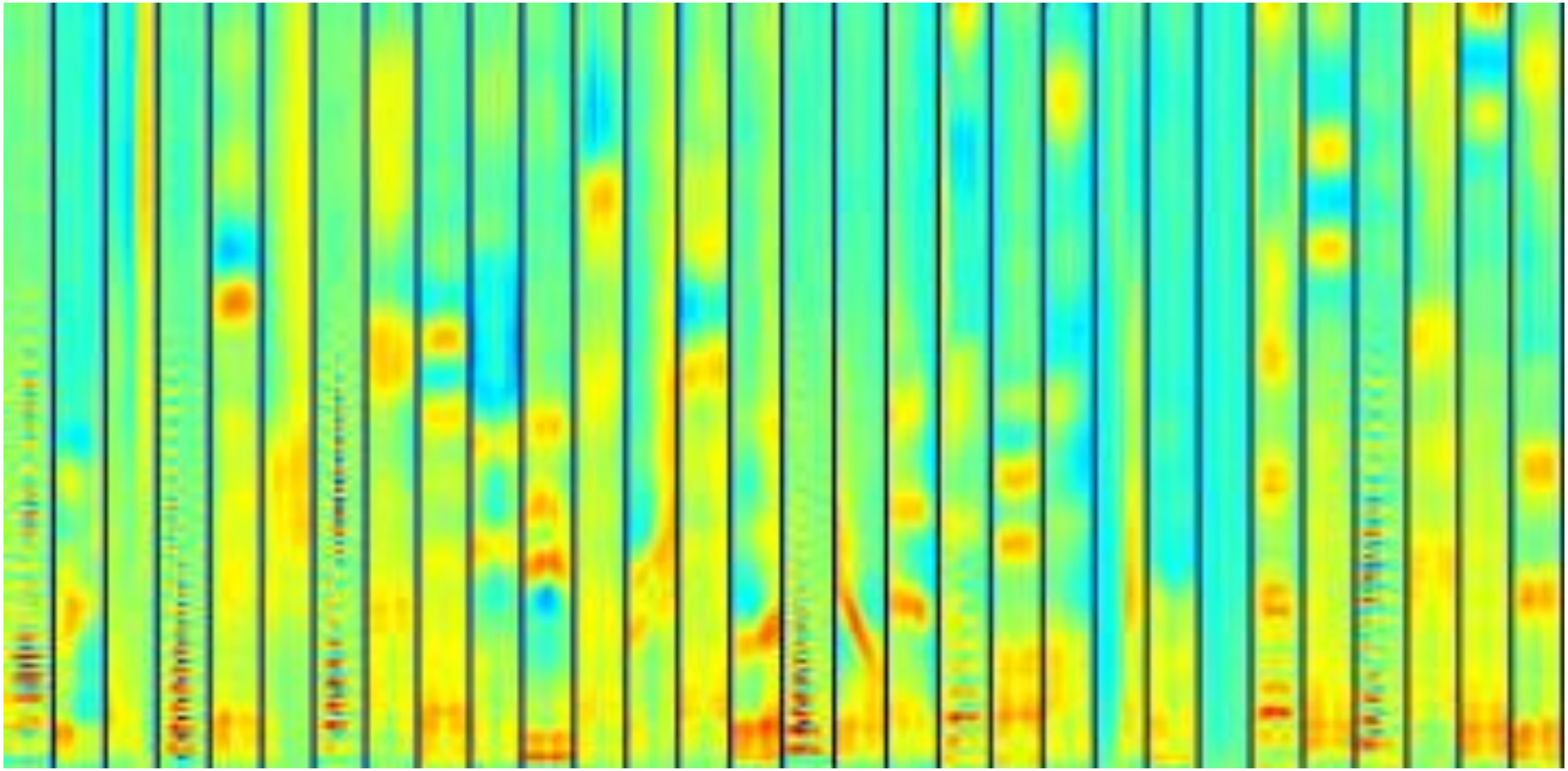


$$x \approx 0.9 * \phi_{36} + 0.7 * \phi_{42} + 0.2 * \phi_{63}$$

Andrew Ng.

Deep Learning in Audio and Speech

- Dictionary of bases ϕ_i learned for speech

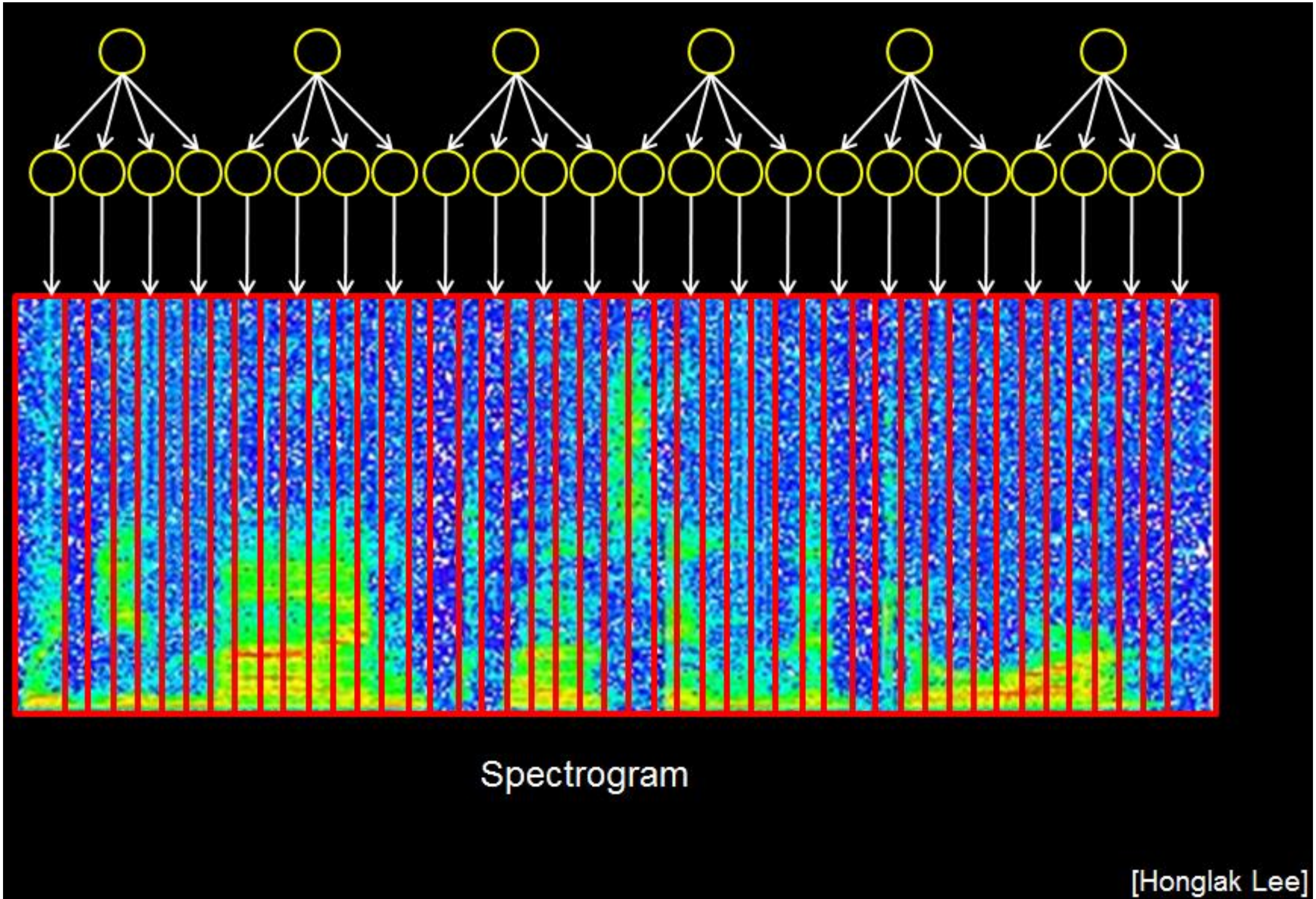


Many bases seem to correspond to phonemes.

Honglak Lee

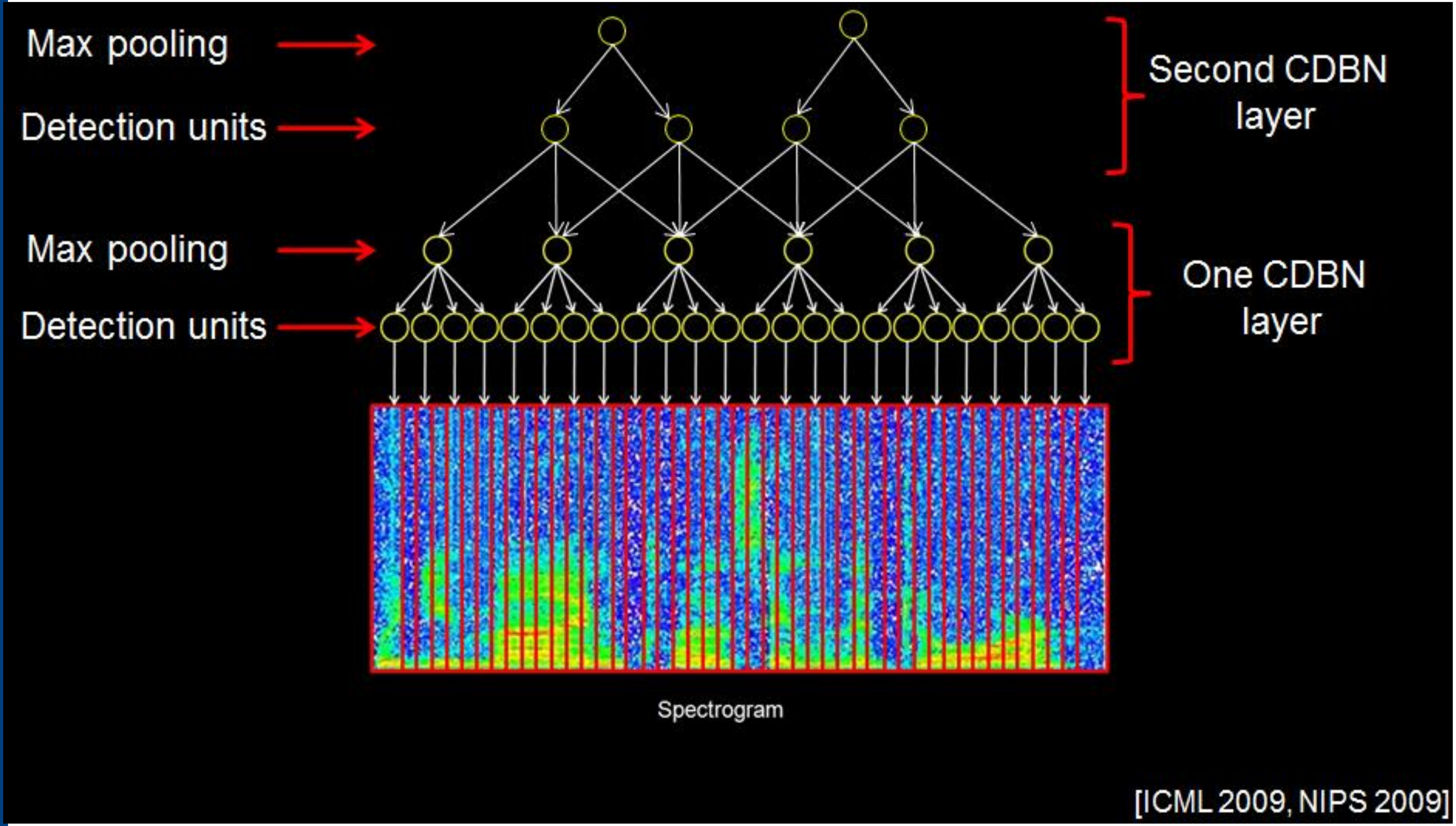
Deep Learning in Audio and Speech

- Hierarchical Sparse coding (sparse DBN) for audio



Deep Learning in Audio and Speech

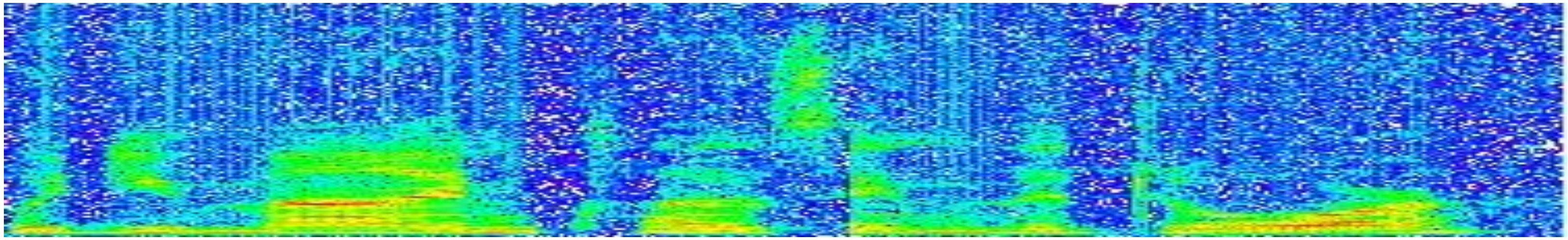
- Convolutional DBN for audio



[ICML 2009, NIPS 2009]

Deep Learning in Audio and Speech

- Phoneme Classification (TIMIT benchmark)



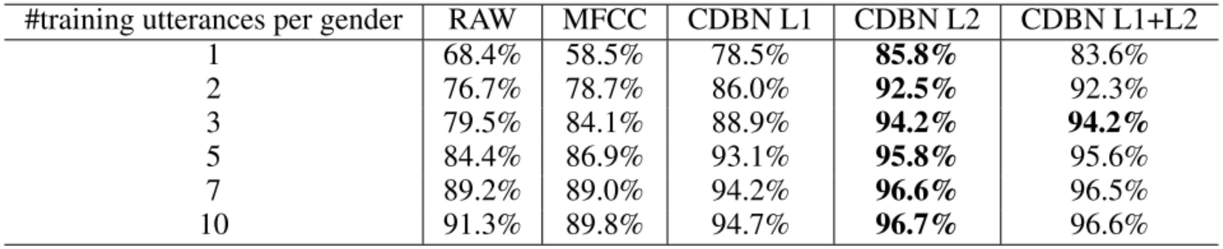
Method	Accuracy
Clarkson and Moreno (1999)	77.6%
Gunawardana et al. (2005)	78.3%
Sung et al. (2007)	78.5%
Petrov et al. (2007)	78.6%
Sha and Saul (2006)	78.9%
Yu et al. (2006)	79.2%
Unsupervised feature learning (our method)	**80.3%**

Unsupervised feature learning significantly improves on the previous state-of-the-art.

Deep Learning in Audio and Speech

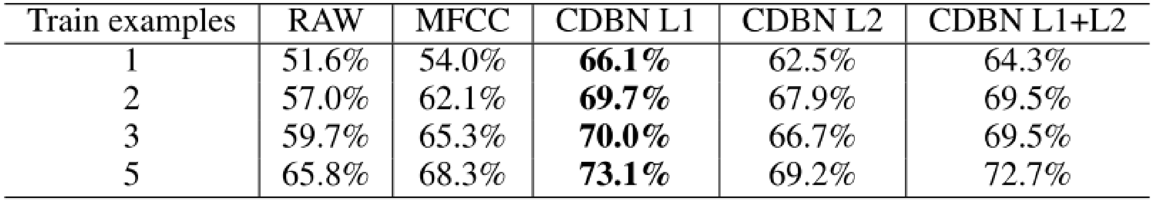
- Audio problems

Gender classification:



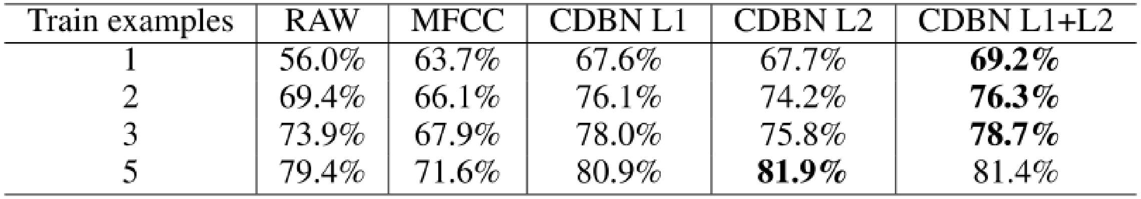
#training utterances per gender	RAW	MFCC	CDBN L1	CDBN L2	CDBN L1+L2
1	68.4%	58.5%	78.5%	**85.8%**	83.6%
2	76.7%	78.7%	86.0%	**92.5%**	92.3%
3	79.5%	84.1%	88.9%	**94.2%**	**94.2%**
5	84.4%	86.9%	93.1%	**95.8%**	95.6%
7	89.2%	89.0%	94.2%	**96.6%**	96.5%
10	91.3%	89.8%	94.7%	**96.7%**	96.6%

Music genre classification:



Train examples	RAW	MFCC	CDBN L1	CDBN L2	CDBN L1+L2
1	51.6%	54.0%	**66.1%**	62.5%	64.3%
2	57.0%	62.1%	**69.7%**	67.9%	69.5%
3	59.7%	65.3%	**70.0%**	66.7%	69.5%
5	65.8%	68.3%	**73.1%**	69.2%	72.7%

Music artist classification:

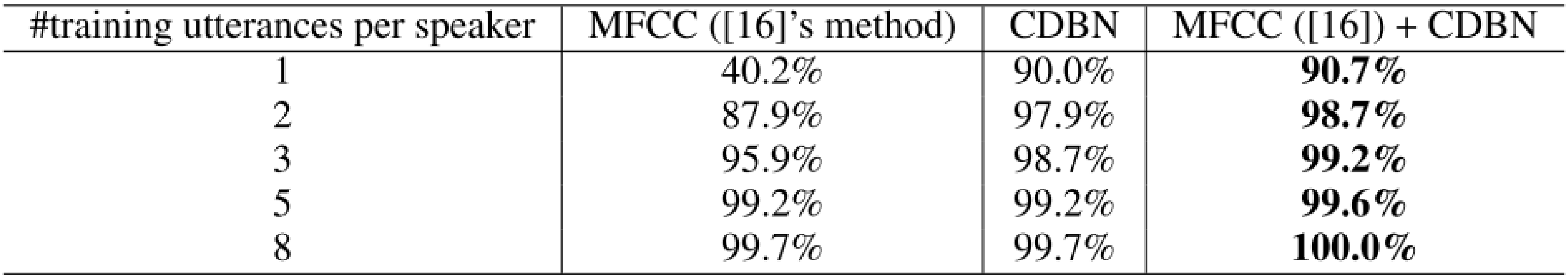


Train examples	RAW	MFCC	CDBN L1	CDBN L2	CDBN L1+L2
1	56.0%	63.7%	67.6%	67.7%	**69.2%**
2	69.4%	66.1%	76.1%	74.2%	**76.3%**
3	73.9%	67.9%	78.0%	75.8%	**78.7%**
5	79.4%	71.6%	80.9%	**81.9%**	81.4%

Outperforms MFCC baselines. Having a deeper network generally does better.

Deep Learning in Audio and Speech

- Speaker Identification



#training utterances per speaker	MFCC ([16]'s method)	CDBN	MFCC ([16]) + CDBN
1	40.2%	90.0%	**90.7%**
2	87.9%	97.9%	**98.7%**
3	95.9%	98.7%	**99.2%**
5	99.2%	99.2%	**99.6%**
8	99.7%	99.7%	**100.0%**

[16] D. A. Reynolds. *Speech Commun*,1995.

The CDBN features outperform the MFCC features, especially when the number of training examples is small.

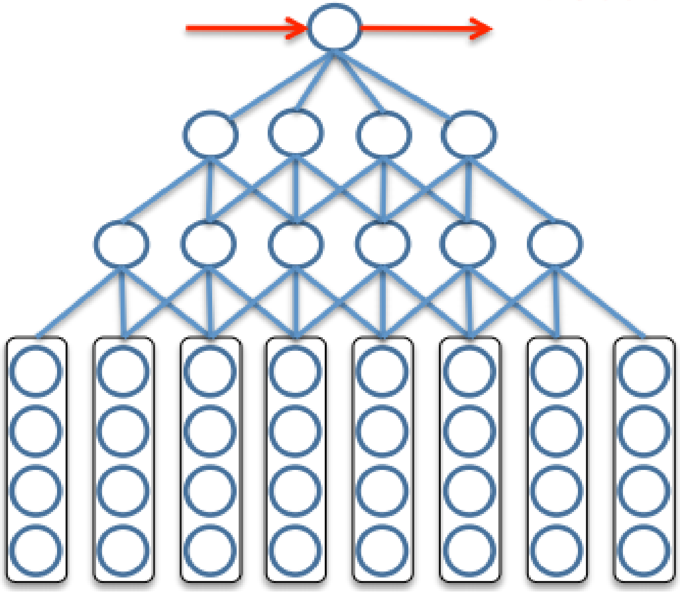
Deep Learning in Audio and Speech

Speech Recognition

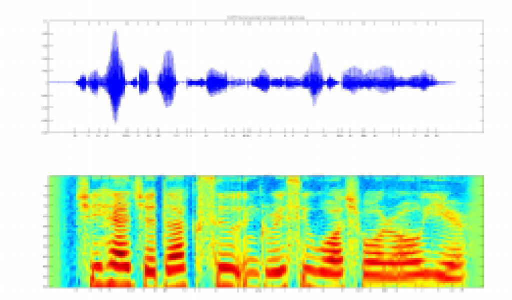
(Zhang, Salakhutdinov, Chang, Glass, ICASSP 2012)

61 phonetic labels

HMM decoder



25 ms windowed frames



- 630 speaker TIMIT corpus: 3,696 training and 944 test utterances.
- **Spoken Query Detection**:
  For each keyword, estimate utterance’s probability of containing that keyword.
- Performance: Average equal error rate (EER).

Learning Algorithm	AVG EER
GMM Unsupervised	16.4%
DBM Unsupervised	14.7%
DBM (1% labels)	13.3%
DBM (30% labels)	10.5%
DBM (100% labels)	9.7%

Outline

- Background
  - Human Auditory System
  - Neural Networks
- Deep Architectures
  - Motivations and Challenges
  - Attempts and Breakthrough
- Deep Learning
  - Autoencoders and Sparse Coding
  - Restricted Boltzmann Machines(RBMs)
  - Greedy Layer-Wise Training
  - Explaination and Justification
- Deep Learning in Audio and Speech
- References

References

- Simon Haykin. Neural Networks: A Comprehensive Foundation, 2009
- Yoshua Bengio. Learning Deep Architectures for AI.
- Andrew Ng. Machine Learning and AI via Brain simulations
- Honglak Lee,etc. Unsupervised feature learning for audio classification using convolutional deep belief networks
- 张春霞，姬楠楠，王冠伟. 受限波尔兹曼机简介.
- Introduction to Deep Learning Algorithms: http://www.iro.umontreal.ca/~pift6266/H10/notes/deepintro.html#introduction-to-deep-learning-algorithms
- Unsupervised Feature Learning and Deep Learning: http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial
- Deep Learning Tutorials: http://deeplearning.net/tutorial/contents.html

Thanks!