# 机器学习与数据科学

Machine Learning and Data Science

主讲: 李春光

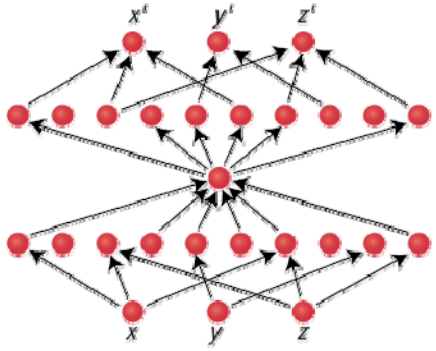
www.pris.net.cn/teacher/lichunguang

模式识别与智能系统实验室

信息与通信工程学院 网络搜索教研中心

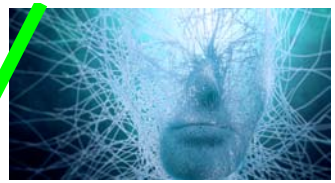
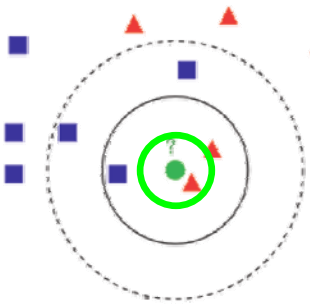
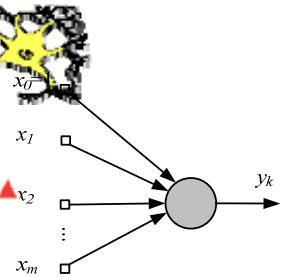
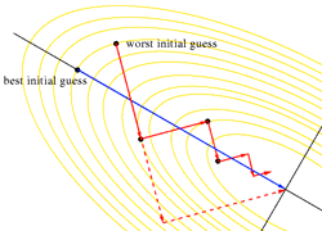
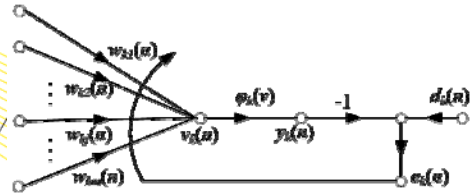
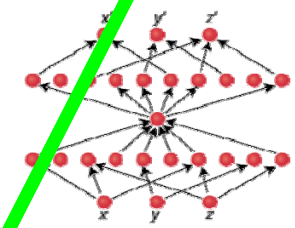
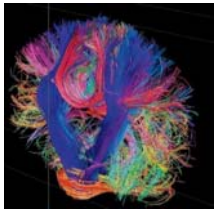
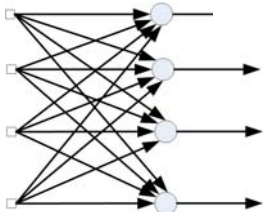
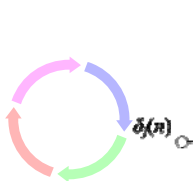
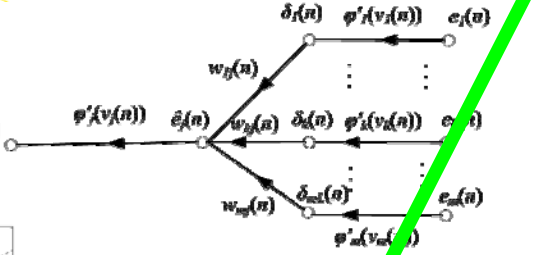
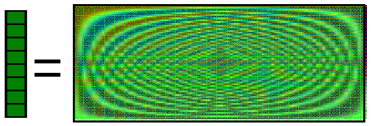
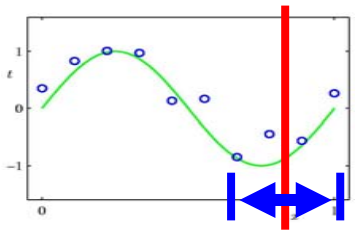
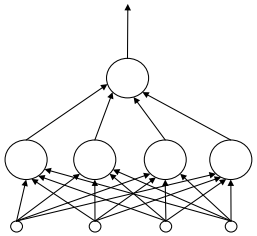
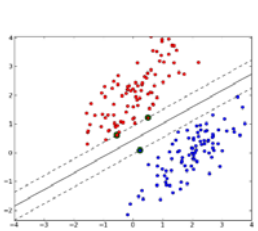
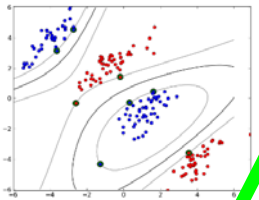
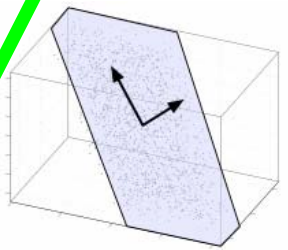
北京邮电大学

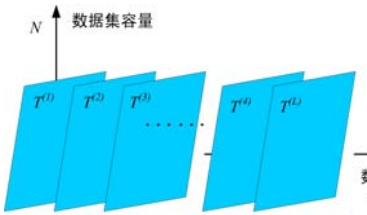
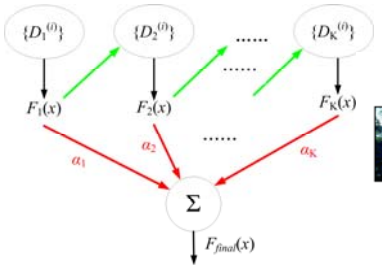
# 专题 四：深度学习(Deep Learning)

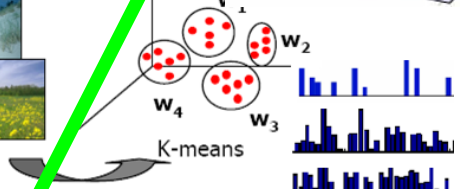
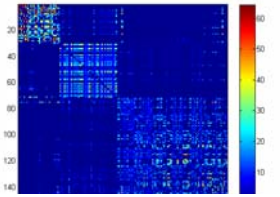
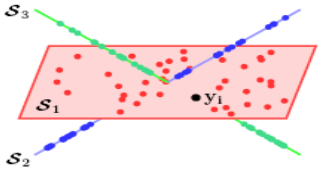
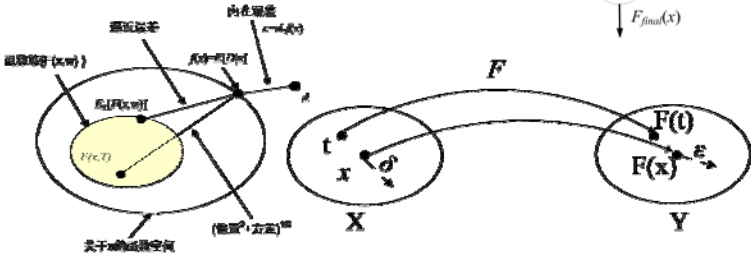
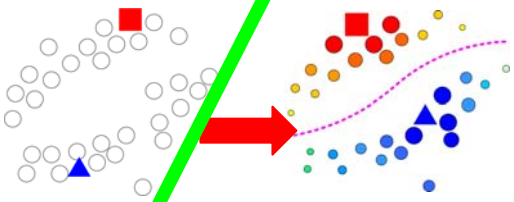
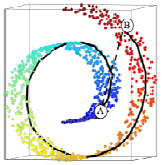
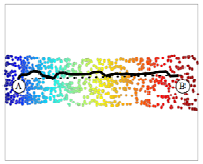
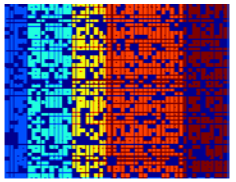


- **内容提要**
  - 引言
  - 深度学习的基本结构
    - Auto-Encoding Networks
    - CNN
    - RBM
  - 网络训练的新技术

# MLDS2018

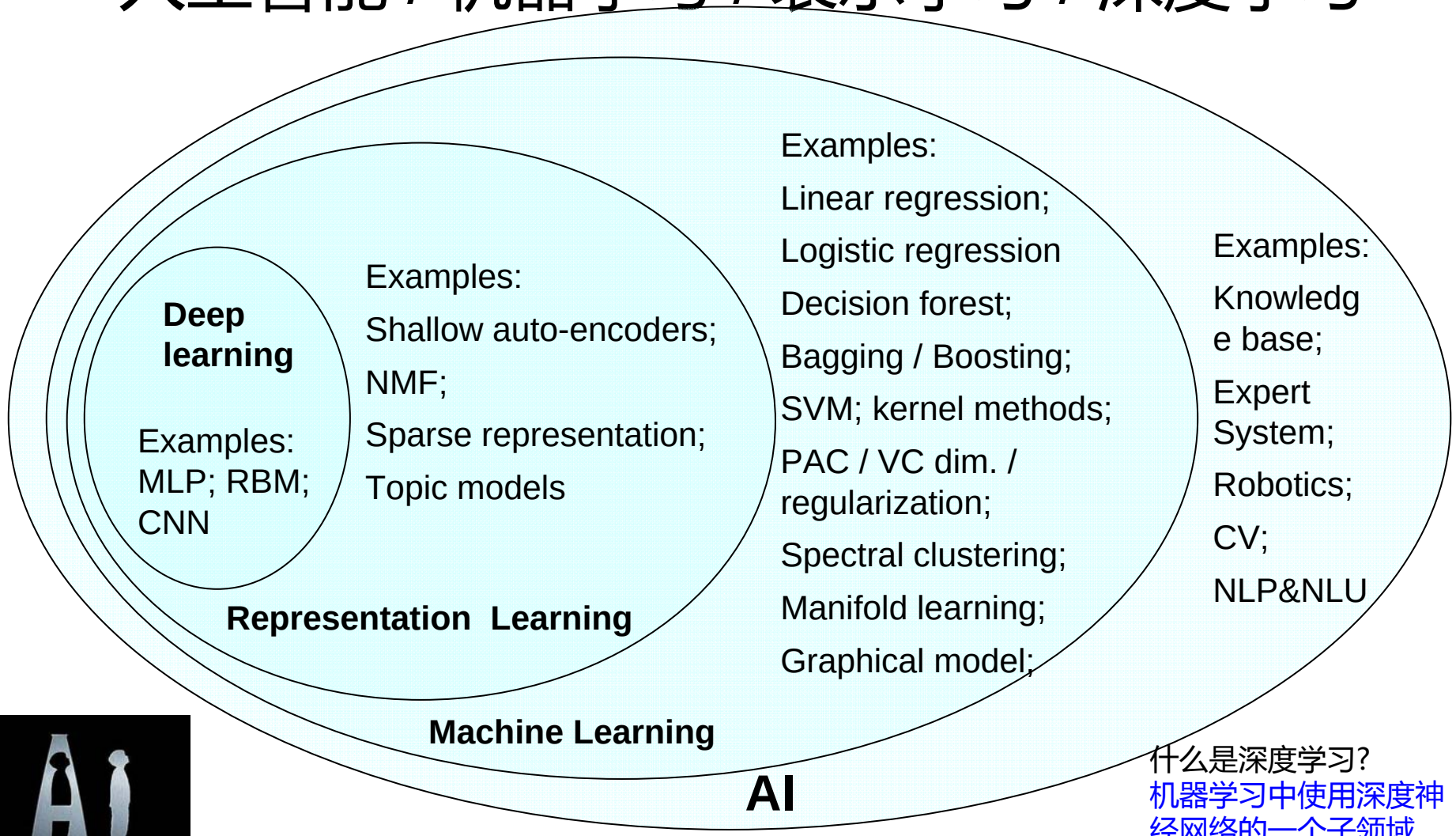


















# 深度学习

- 人工智能 / 机器学习 / 表示学习 / 深度学习




什么是深度学习?
机器学习中使用深度神经网络的一个子领域.

# 回顾: 神经网络发展简史

- 1943: S. McCulloch & W. Pitts
  - **Neuron model**
- 1958: F. Rosenblatt
  - **Perceptron**
- 1960: B. Widrow & M.Hoff
  - **ADALine**
- 1969: M. Minsky &S. Papert
  - **XOR problem**
- 1986: D. Rumelhart, G.Hinton, & R.Wiliams
  - **Backpropagation algorithm for Multi-Layered Perceptron**
- 1992: V.Vapnik & C.Cortes
  - **(kernel) SVM**
- 2006: G. Hinton & S. Ruslan
  - **Restricted Boltzman Machine**
- **2012**: CNN → Visual Object Categorization (VOC) on ImageNet

# 回顾：里程碑事件——反向传播算法

- 神经网络的第2次研究热潮
  - 以训练多层感知器的反向传播(BP)算法的提出为标志
    - 1985年Parker和1985年LeCun分别独立给出反向传播训练算法
    - 1986年**Rumelhart，Hinton和Williams**反向传播算法被再次独立发现，论文发表于Nature上

      Rumelhart, Hinton & Williams, "Learning representations by back-propagating errors", Nature, 1986
- 神经网络的第2次研究低谷
  - 在20世纪90年代中期之后神经网络的研究陷入研究低谷
    - 以支持向量机、正则化框架和贝叶斯网络为代表的统计学习在90年代的兴起
      - 前两者对应于统计学习理论
      - 后者对应于贝叶斯理论

# 回顾：里程碑事件——RBM用于降维

- 以G. E. Hinton的Science论文发表为曙光

## New Life for Neural Networks

Garrison W. Cottrell

Recent advances in machine learning have caused some to consider neural networks obsolete, even dead. This work suggests that such announcements are premature.

[1] Garrison W. Cottrell, New Life for Neural Networks, SCIENCE,Vol.313, July 2006, pp.454-455.
[2] **Hinton G.E. and Salakhutdinov R.R., Reducing the dimensionality of data with neural networks, SCIENCE, Vol.313, July 2006, pp.504-507.**

# 深度学习(Deep Learning)

- 3篇标志性工作
  - 深度置信网络**(DBNs)**
    - 2006：Hinton’ s revolutionary work on Deep Belief Networks (DBNs)

Hinton et al., “A fast learning algorithm for deep belief nets”, Neural Computation, 18:1527-1554, 2006.

  - 自编解码网络
    - 2006：encoder-decoder

Yoshua Bengio et al., “Greedy Layer-Wise Training of Deep Networks”, NIPS 2006.

  - 稀疏自编解码网络
    - 2006：Sparse encoder-decoder

Yann LeCun, “Efficient Learning of Sparse Representations with an Energy-Based Model”, NIPS 2006.

Sepp Hochreiter, Jurgen Schmidhuber, "Long short-term memory." Neural computation, Vol.9, 1997, pp.1735-1780.

# 深度学习综述

- LeCun Yann, Yoshua Bengio, Geoffrey Hinton, "Deep Learning", **Nature**, Vol. 521, pp.436-444, May 2015.

- Juergen Schmidhuber, "**Deep learning in neural networks: An overview**", Neural Networks, Vol. 61, Jan. 2015, pp. 85–117.

# 请思考...

- 有监督学习的基本问题:
  - 基于观测数据(X,Y), 寻找F(x): X→Y；希望在给定数据集X上得到的F(x)能推广到X中所不包括的新数据上

- 回顾: MLP能做什么?
  - 理论上证明：包含一个Sigmoid隐藏层的MLP可以作为通用的函数逼近器，逼近任何函数

- 疑问:
  - 为什么神经网络的研究会陷入研究低谷?
  - 为什么会在十几年后又重新兴起?

G. E. Hinton: “我们把MLP和BP算法发明得太早了。因为那时候还没有找到像现在这么有效的训练方法、没有像现在这么充足的训练数据、也没有足够强大的计算能力...”

# 深度学习兴起的原因

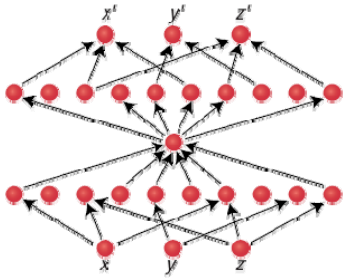
- 改进的训练方法与结构
  - **Layer-wise training**
  - **Batch Normalization**
  - **DropOut / DropConnect**
  - **ReLU**
  - **AlexNet / VGGNet / ResNet / DenseNet ….**
- 更强大的计算能力
  - **GPU**
- 更多的训练数据
  - **ImageNet / Places / …**

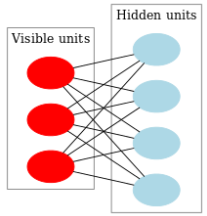
**在图像分类、语音识别等典型应用问题上显著地提升了分类准确率**

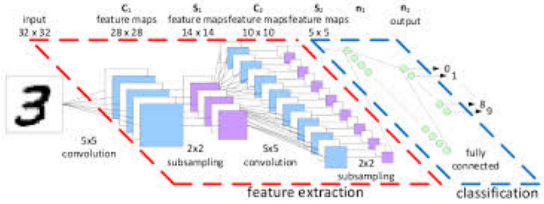
- ImageNet 2012 分类任务:
  - Error rate (@Top 5): 26.1% → 16.4%

[1] Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In Proc. Advances in Neural Information Processing Systems 25 1090–1098 (2012).

- **内容提要**
  - 引言
  - 深度学习的基本结构
    - Auto-Encoding Networks
    - CNN
    - RBM
    - RNN / LSTM
  - 网络训练的新技术
    - Layer-wise training
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...

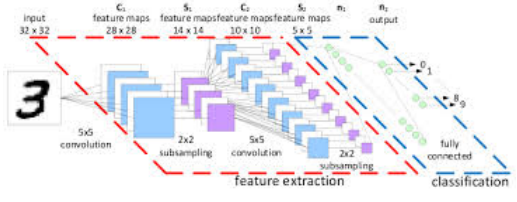


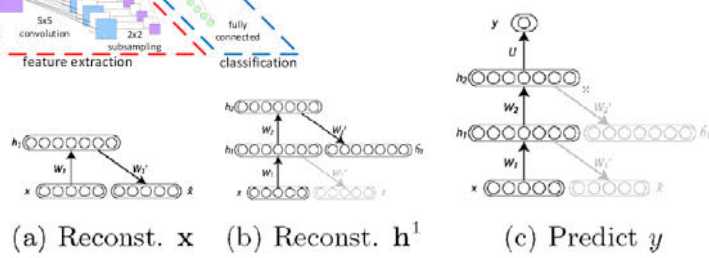


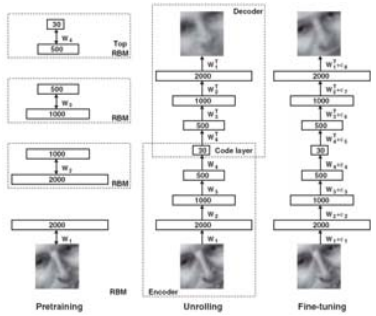



# 深度学习的基本思路

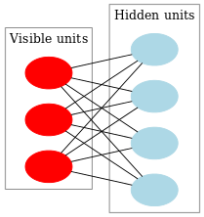
- 借助多层模型获得更强的特征表达
  - 结构特点：无监督学习 + 有监督学习
    - 使用逐层训练(Layer-wise training)获得高度非线性的特征
    - 使用有监督学习进行精细调整(fine-tuning)
  - 典型结构举例
    - 卷积神经网络 (CNN)
    - 堆栈的(stacked)自动编码器 (Auto-Encoder)
    - 堆栈的(stacked)受限波尔兹曼机 (Restricted Boltzmann Machine)
      - 深度置信网络 (Deep Belief Networks)

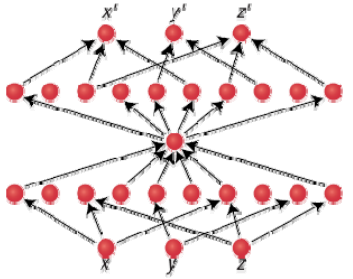


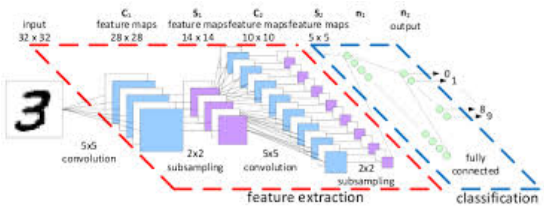



- **内容提要**
  - 引言
  - 深度学习的基本结构
    - Auto-Encoding Networks
    - CNN
    - RBM
    - RNN / LSTM
  - 网络训练的新技术
    - Layer-wise
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...



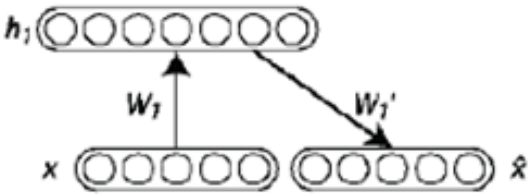



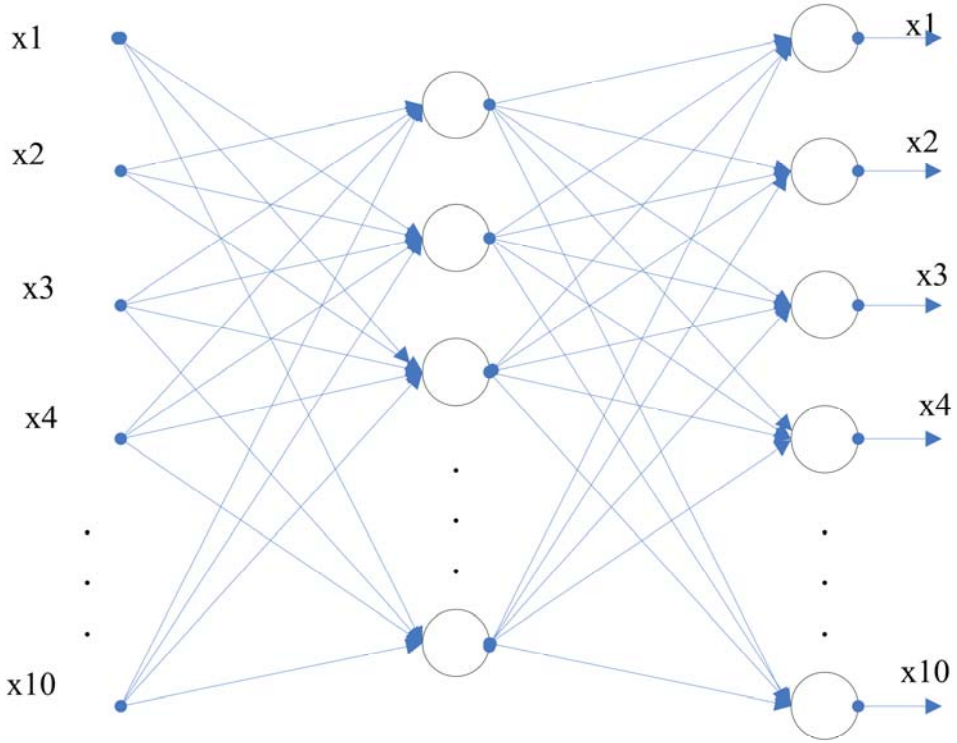


# 自编解码网络

- Train a network to predict the inputs itself.
  - 把输入当作为目标输出，即

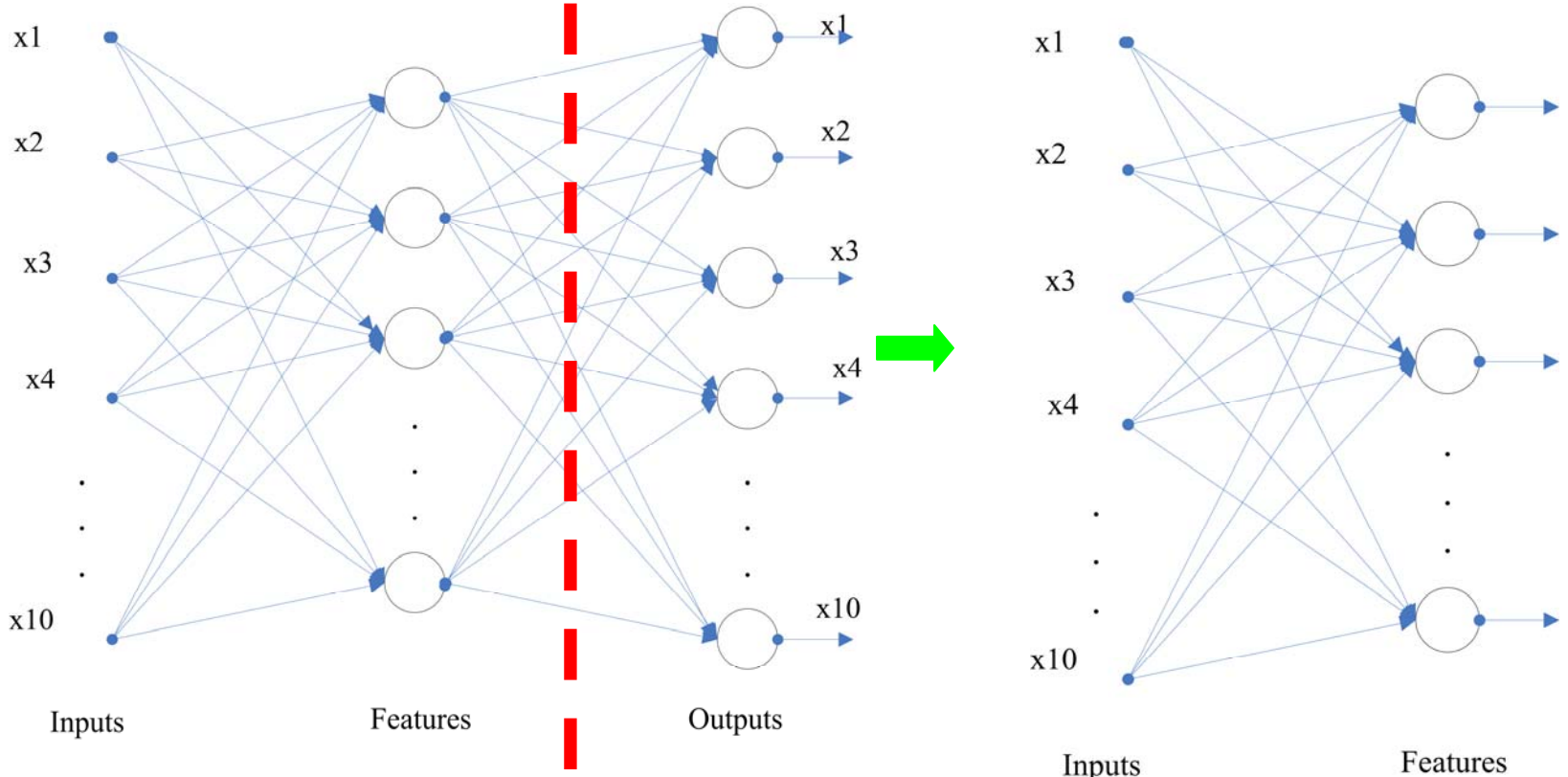
$$\min_{W} \sum_{\mathbf{x} \in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x}, W\right), W\right)\right)$$



# 自编解码网络用于特征学习

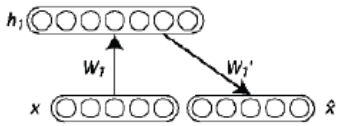
- Train a network to predict the inputs itself.
  - 把输入当作为目标输出
  - 若从中间切开，则可获得所学习的特征表示



# 自编解码网络中的不变性

- Train a network to predict the inputs itself.
  - 把输入当作为目标输出，即

$$\min_W \sum_{\mathbf{x}\in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x},W\right),W\right)\right)$$




  - 为增加**Robustness**，可加入由**1**阶或高阶信息定义的正则化项

$$\min_W \sum_{\mathbf{x}\in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x},W\right),W\right)\right) + \lambda\left\|\nabla_{\mathbf{x}} h\left(\mathbf{x},W\right)\right\|_F^2$$

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. : Contracting auto-encoders: Explicit invariance during feature extraction. ICML 2011.

$$\min_W \sum_{\mathbf{x}\in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x},W\right),W\right)\right) + \lambda\left\|J\left(\mathbf{x}\right)\right\|_F^2 + \gamma \mathrm{E}_{\varepsilon\sim N\left(0,\sigma^2 I\right)}\left[\left\|J\left(\mathbf{x}\right)-J\left(\mathbf{x}+\varepsilon\right)\right\|_F^2\right]$$

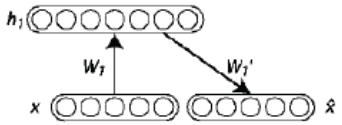
其中 $J\left(\mathbf{x}\right)=\nabla_{\mathbf{x}} h\left(\mathbf{x},W\right)$

Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y., and Glorot, X. : Higher order contractive auto-encoder. ECML PKDD 2011.

# 自编解码网络中加入半监督

- Train a network to predict the inputs itself.
  - 把输入当作为目标输出，即

$$\min_{W} \sum_{\mathbf{x}\in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x}, W\right), W\right)\right)$$



Reconst. x

  - 在自重构误差项基础上，增加半监督信息约束

$$\min_{W} \sum_{\mathbf{x}\in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x}, W\right), W\right)\right) + \lambda \sum_{i,j} \left(-1\right)^{1\{y_i \neq y_j\}} \left\| h\left(\mathbf{x}_i, W\right) - h\left(\mathbf{x}_j, W\right) \right\|_2^2$$

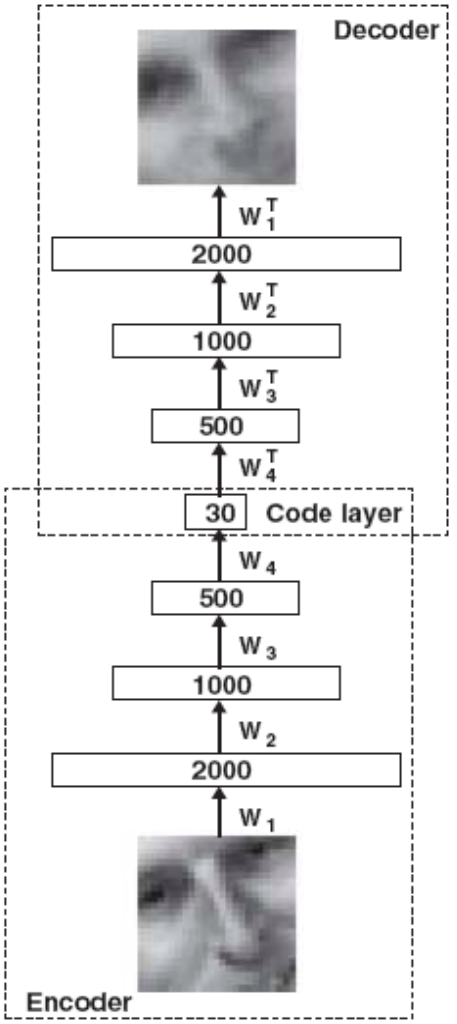
效果: 使得标签一致的样本的特征表达尽量邻近，标签不一致的样本的特征表达尽量远离

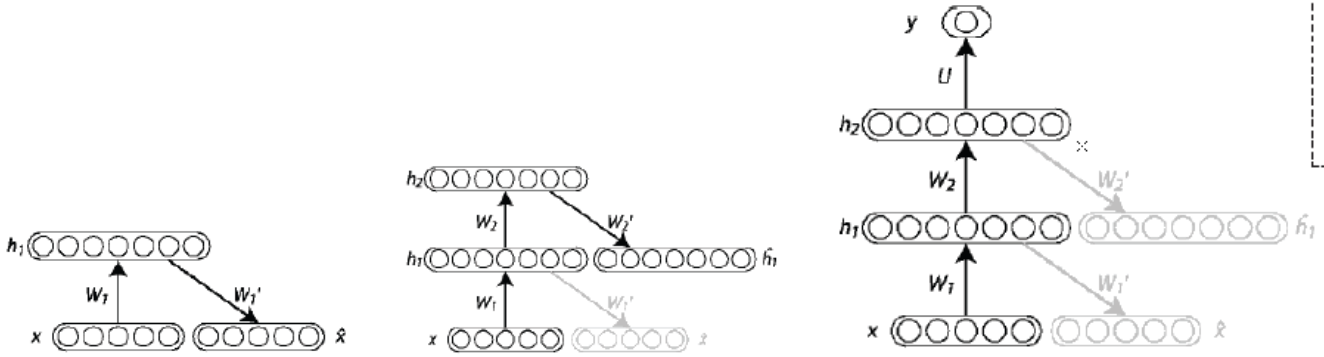
Weston, J., Ratle, F., and Collobert, R. Deep learning via semi-supervised embedding. ICML2008.

# 深度自编解码网络

- **Deep Auto-Encoder Networks:**
  - 堆栈的**(Stacked)Auto-Encoder**

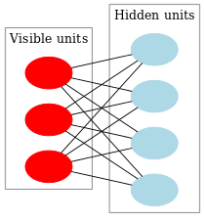
- 训练策略：
  - 逐层训练 + Fine-tuning

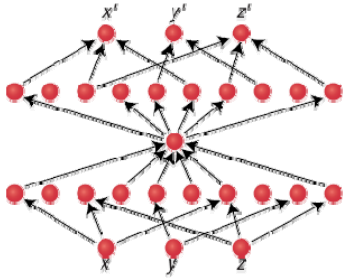


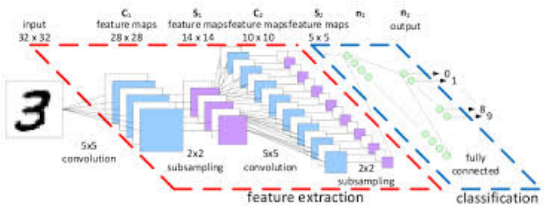




(a) Reconst. $\mathbf{x}$ (b) Reconst. $\mathbf{h}^1$ (c) Predict $y$

- **内容提要**
  - 引言
  - 深度学习的基本结构
    - Auto-Encoding Networks
    - CNN
    - RBM
    - RNN / LSTM / GAN
  - 网络训练的新技术
    - Layer-wise
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...



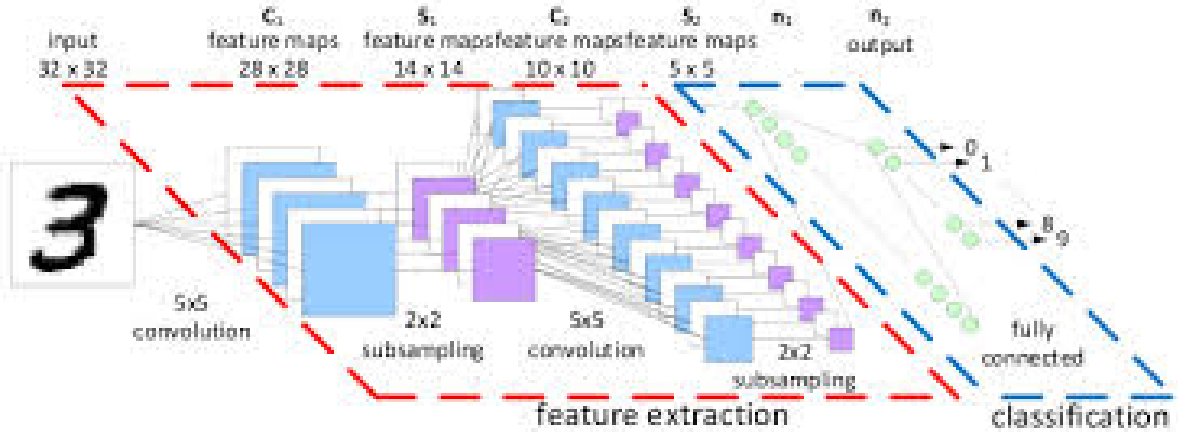


# CNN的不同实现

- LeNet
  - **[1] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." Neural computation, 1989.**
  - **[2] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE (1998): 2278-2324.**
- AlexNet
  - **[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." NIPS. 2012.**
- VGGNet
- GoogleNet
- ResNet
  - **He, K., Ren, S., Sun, J., & Zhang, X.: Deep Residual Learning for Image Recognition, CVPR 2016.**

# 卷积神经网络

- Convolutional Neural Networks (CNN)
  - **LeNet**



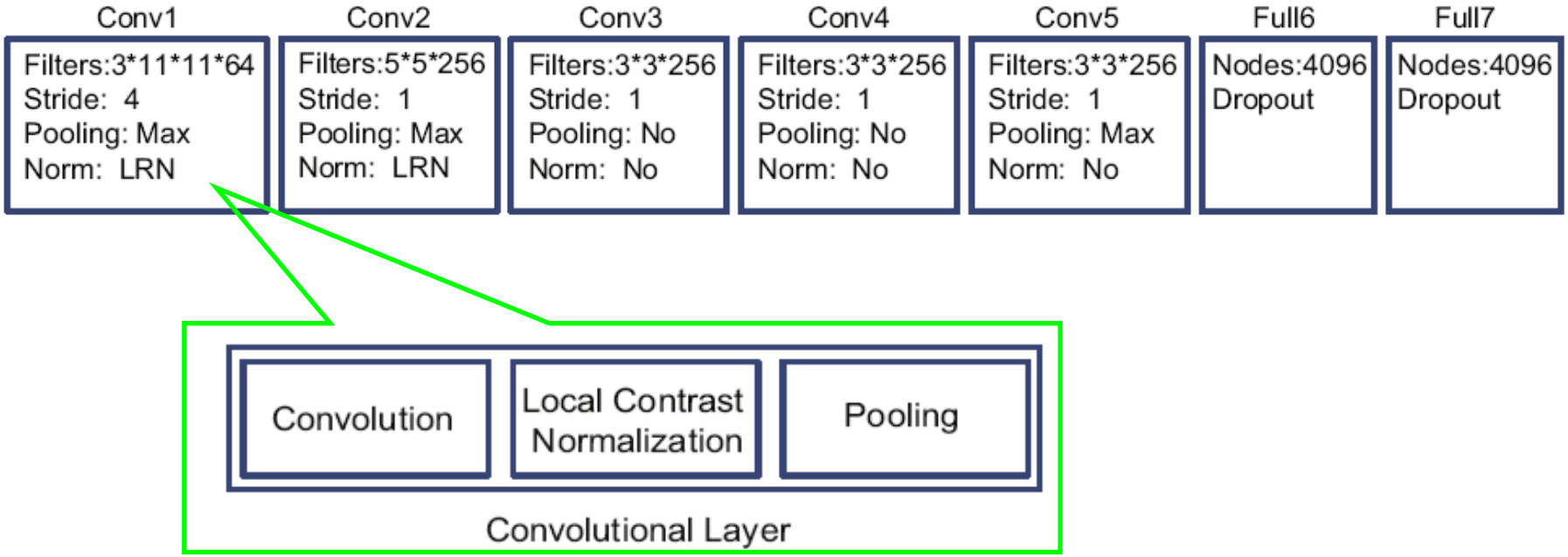

[1] Yann LeCun et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 1998, pp.2278-2324.

# CNN的不同实现

- LeNet
  - **[1] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." Neural computation, 1989.**
  - **[2] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE (1998): 2278-2324.**
- AlexNet
  - **[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." NIPS. 2012.**
- VGGNet
- GoogleNet
- ResNet
  - **He, K., Ren, S., Sun, J., & Zhang, X.: Deep Residual Learning for Image Recognition, CVPR 2016.**

# CNN的网络结构

- CNN用于图像分类
  - **AlexNet**




[1] Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In Proc. Advances in Neural Information Processing Systems 25 1090–1098 (2012).

# CNN卷积模块的构成

- CNN的每层由下述4个基本单元构成:
  - 卷积**(Convolution):**
    - 使用一组学习到的filters检测局部特征
  - **ReLU**校正**(Rectification):**
    - ReLU(x) = max(x, 0)
  - 在邻域内汇集**(Pooling):**
    - 把一定区域内的特征融合起来
  - 局部对比规范化**(Local Contrast Normalization)**

[1] Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In Proc. Advances in Neural Information Processing Systems 25 1090–1098 (2012).

# 1个8层CNN的体系结构

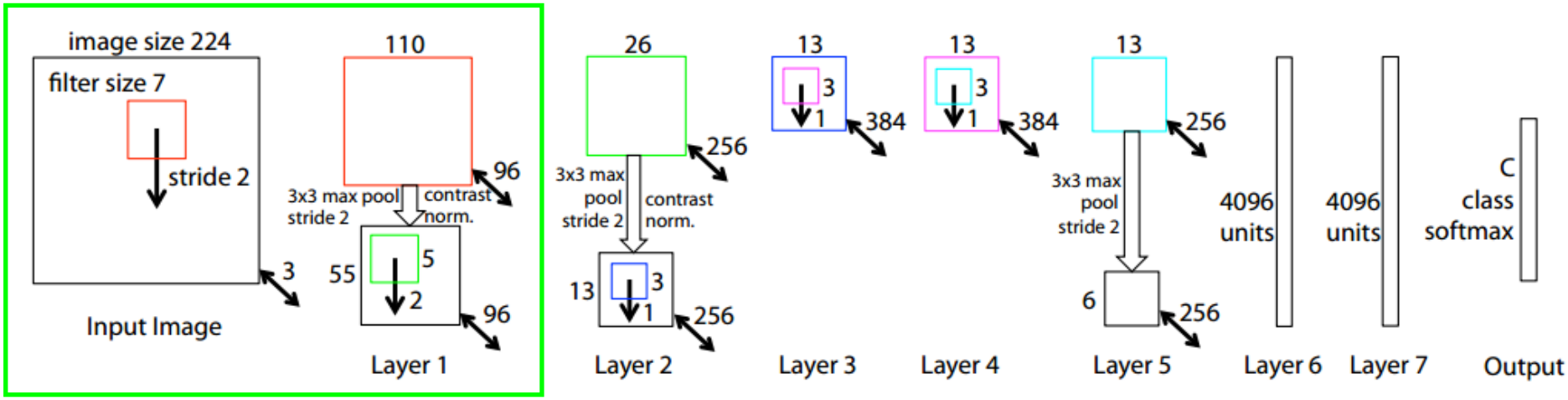



*Figure 3.* Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a $C$-way softmax function, $C$ being the number of classes. All filters and feature maps are square in shape.

Matthew D. Zeiler and Rob Fergus, "Visualizing and Understanding Convolutional Networks", http://arxiv.org/pdf/1311.2901.pdf.

# 1个8层CNN的体系结构

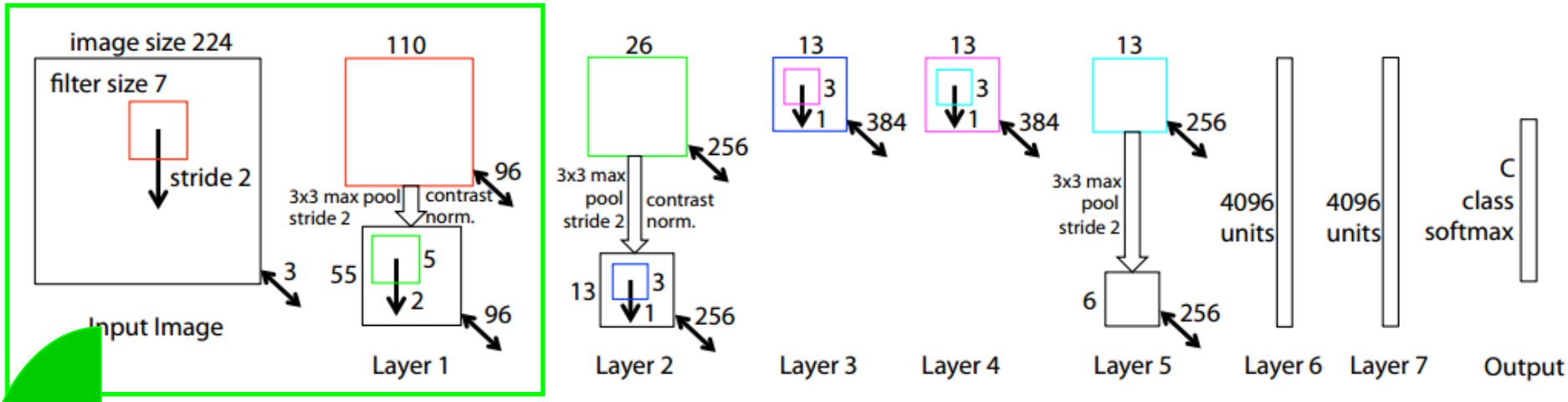
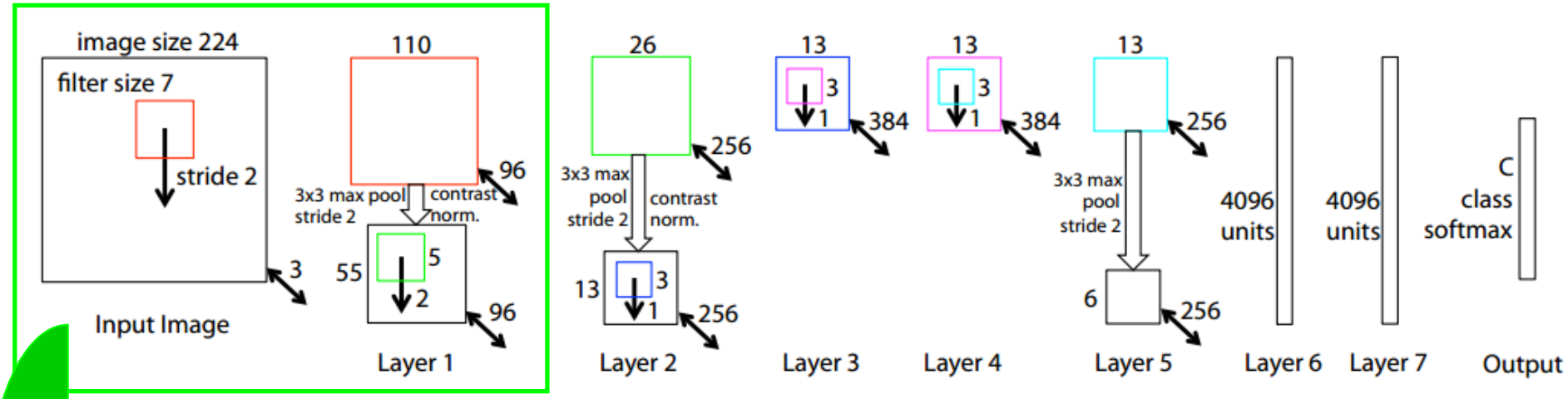



*Figure 3.* Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a $C$-way softmax function, $C$ being the number of classes. All filters and feature maps are square in shape.

- Stride 2: 相邻窗口每次滑动2个像素(或单位), i.e. 5个像素重叠(overlapping)
- Filter size 7: 使用7x7的kernel作用在每个窗口内，输出一个响应值
- 3x3 max pooling: 下采样(Downsampling)步骤, 在一个3x3邻域内的9个响应值里选取最大值作为输出
  - **3x3 max pooling stride 2:** 相邻的**3x3**窗口每次滑动**2**个像素，即重叠**1**个像素

# 1个8层CNN的体系结构




*igure 3.* Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a $C$-way softmax function, $C$ being the number of classes. All filters and feature maps are square in shape.

- 输入为彩色图像，包含3个通道
- 110 x 110 → 55 x 55: 使用3x3 max pooling with stride 2之后尺寸减半
- 3通道 → 96 通道: 选用96个7x7x3的filters对224x224x3的图像卷积，每个filter给出一个特征图(feature map)

# 1个8层CNN的体系结构

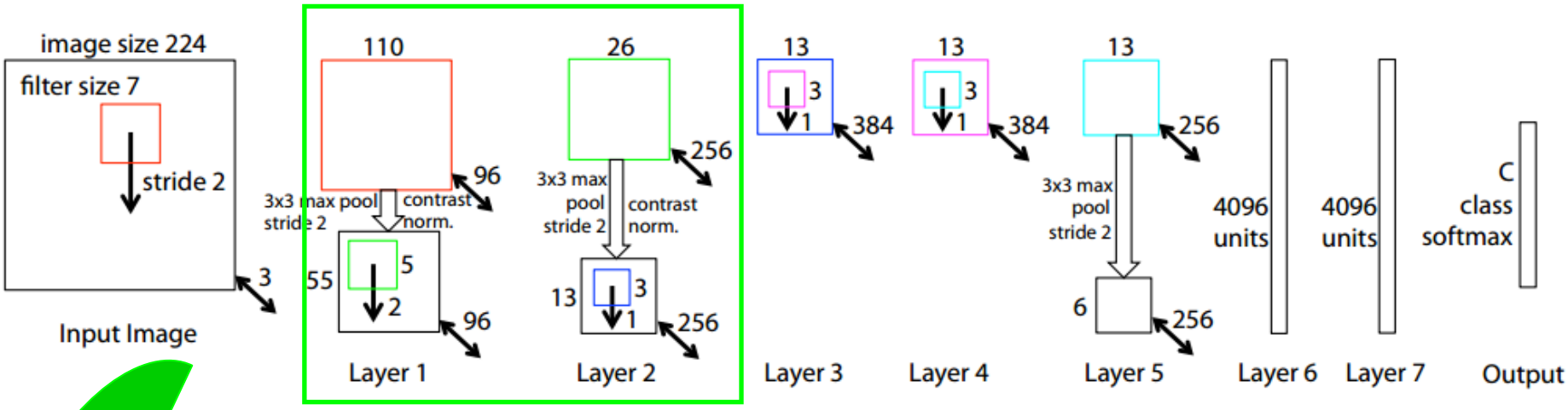



*Figure* hitecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the put. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a $C$-way softmax function, $C$ being the number of classes. All filters and feature maps are square in shape.

- 第1层输出为96个55x55的feature map，可以看做96通道的“特征图”
- 第2层选用256个filters，其中每个filter是5x5x96，作用在55x55的特征图上，得到256个26x26的feature maps, 然后MaxPooling之后变成13x13x256
- 第6层和第7层为全连接层(Full Connected Layers)

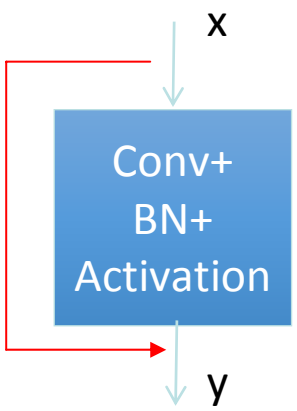
# CNN的不同实现

- LeNet
  - **[1] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." Neural computation, 1989.**
  - **[2] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE (1998): 2278-2324.**
- AlexNet
  - **[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." NIPS. 2012.**
- VGGNet
- GoogleNet
- ResNet
  - **He, K., Ren, S., Sun, J., & Zhang, X.: Deep Residual Learning for Image Recognition, CVPR 2016.**

# Residual network (ResNet)

- ResNet的基本模块

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}.$$



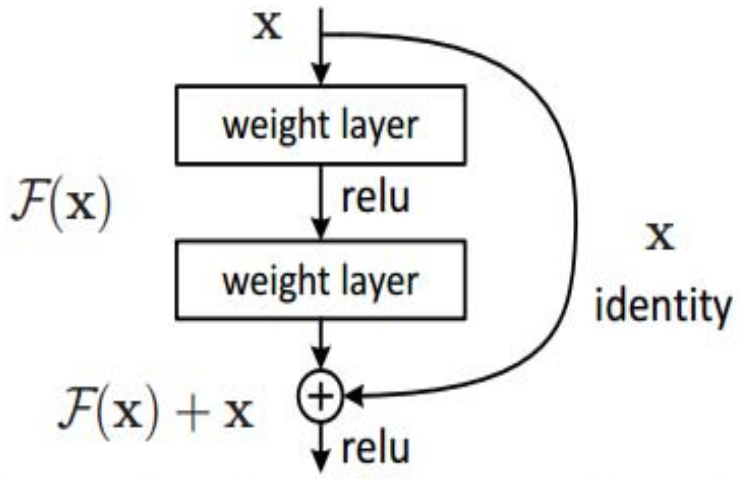




Figure 2. Residual learning: a building block.

He, K., Ren, S., Sun, J., & Zhang, X.: Deep Residual Learning for Image Recognition, CVPR 2016.

# 应用举例: CNN for 句子表达与分类

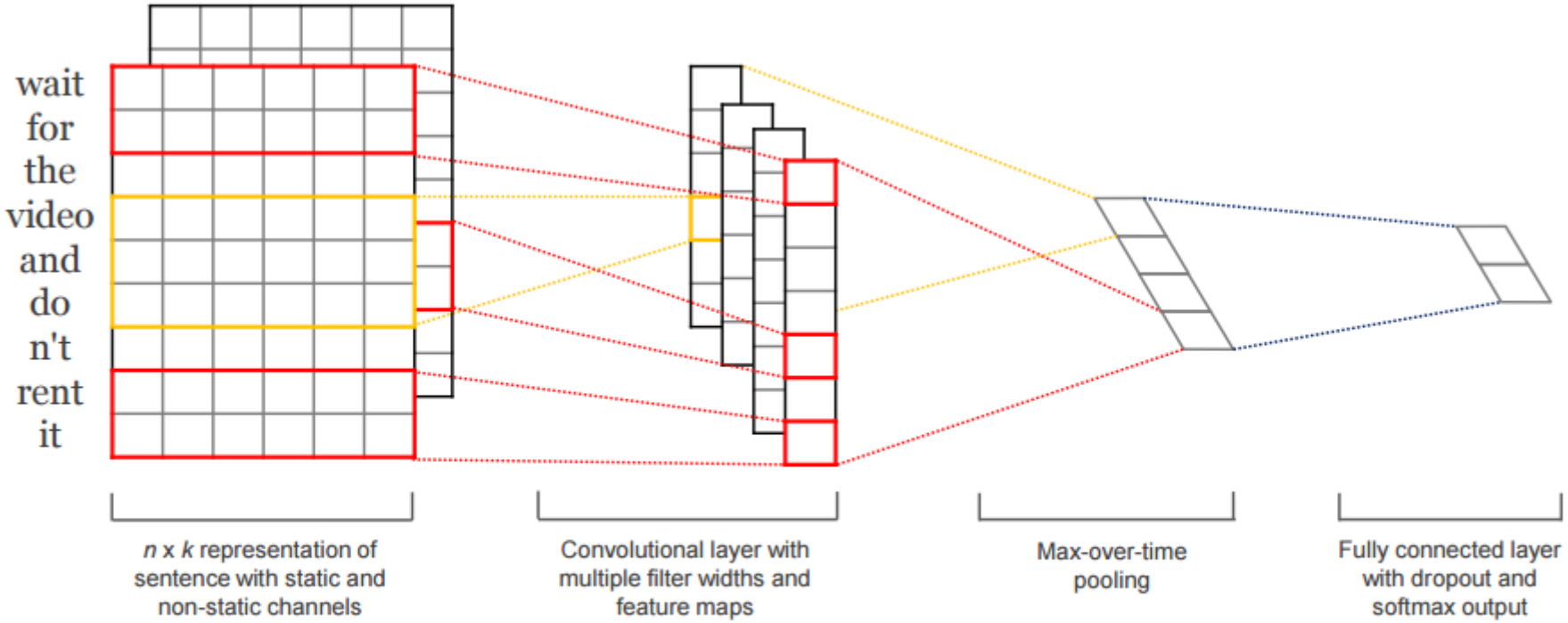



Figure 1: Model architecture with two channels for an example sentence.

Yoon Kim: "Convolutional Neural Networks for Sentence Classification", http://arxiv.org/pdf/1408.5882.pdf

# 应用举例: CNN for 动态场景/纹理分类

- TCoF: Transferred CovNet Feature

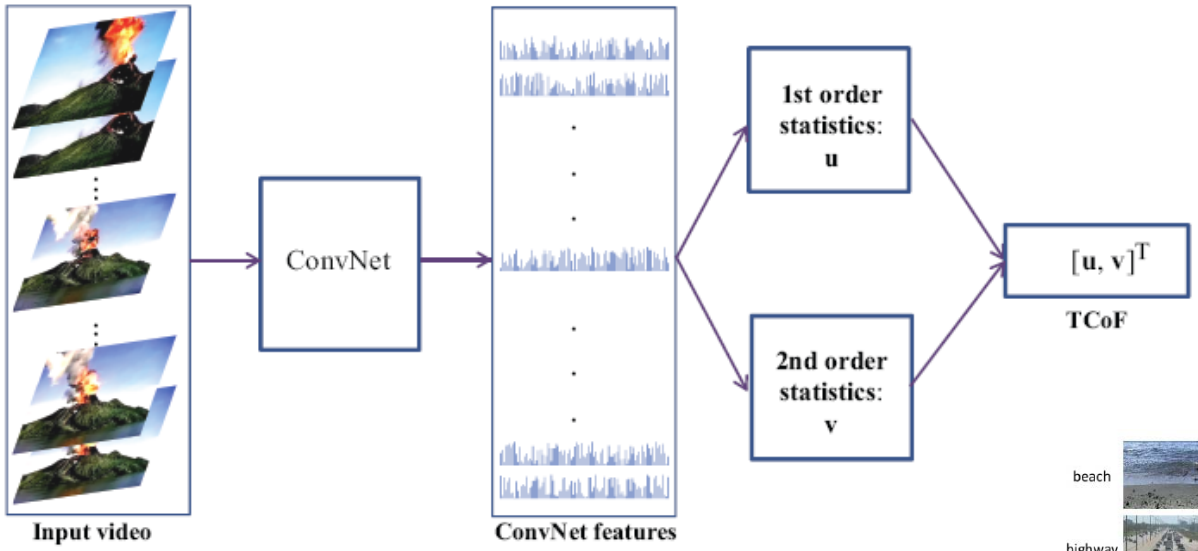



Figure 4: An illustration of our TCoF scheme.

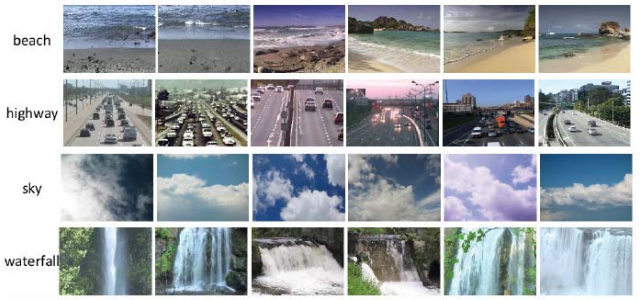
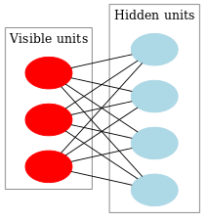
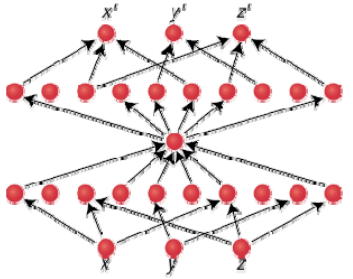


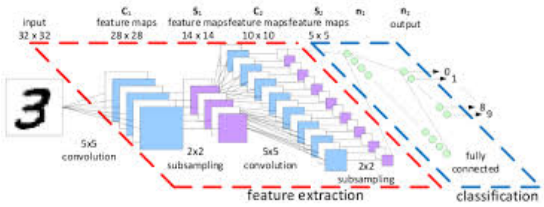

Figure 1: Sample images from dynamic scene data set YUPENN. Each row corresponds a category.

[1] X. Qi, C.-G. Li, G. Zhao, X. Hong, M. Pietikainen, "Dynamic texture and scene classification by transferring deep image features", Neurocomputing, vol.171, 2016, pp:1230-1241.

- **内容提要**
  - 引言
  - 深度学习的基本结构
    - Auto-Encoding Networks
    - CNN
    - RBM
    - RNN / LSTM / GAN
  - 网络训练的新技术
    - Layer-wise
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...





# Boltzmann学习

- 源于统计力学
  - 属于随机学习算法
  - 基于**Boltzmann**学习规则设计的神经网络称为**Boltzmann**机
    - 神经元构成递归结构，以二值方式运作
    - 能量函数定义为：

$$E = -\frac{1}{2}\sum_{j}\sum_{i \neq j} w_{ij} x_i x_j$$

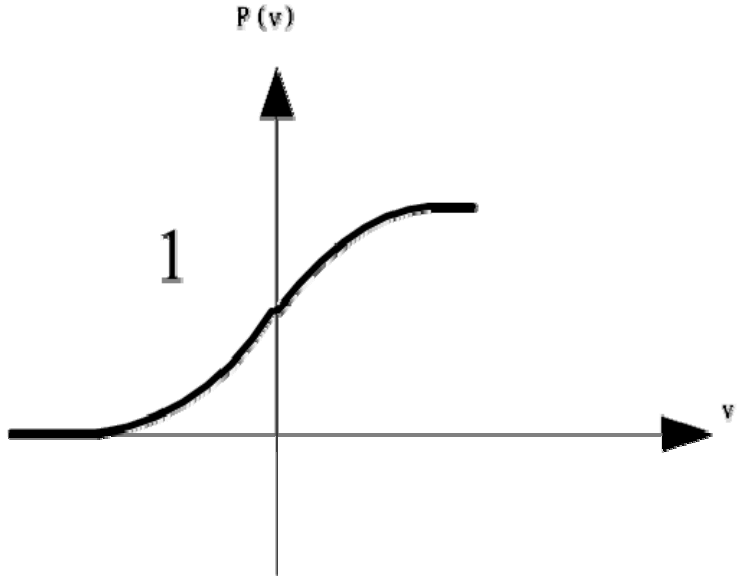
      - 能量函数的值由机器的个体神经元$i$占据的特定状态 $x_i$ 所决定

# 随机神经元模型

- 把McCulloch-Pitts模型的激活函数用概率分布来实现
  - $s$ 表示神经元的状态

$$s=\begin{cases}1 & \text{以概率} p(v)\\ 0 & \text{以概率} 1-p(v)\end{cases}$$

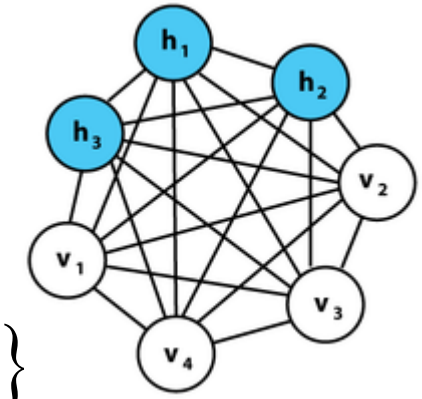
其中 $$p(v)=\frac{1}{1+\exp(-v/\mathrm{T})}$$




T是伪温度，控制噪声水平和点火(firing)的不确定性

# 玻尔兹曼机 (Boltzmann Machine)

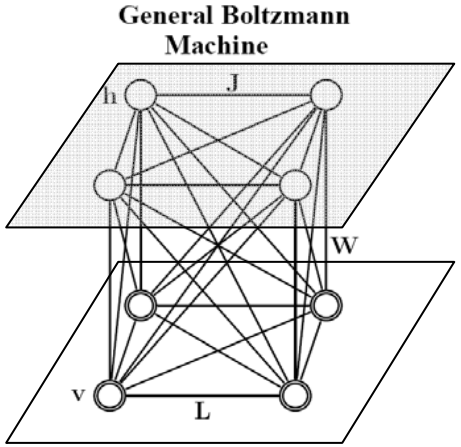
- BM是由二值神经元构成的神经网络
  - h: 隐含神经元集合; v: 可见神经元集合




– 在网络上定义一个能量函数 $E(\mathbf{h},\mathbf{v})$

– 每个神经元的状态是二值的: $s_i \in \{0,1\}$

– 每个神经元是随机的




$$P(v_i=1)=\sigma\left(\frac{\Delta E_i}{T}\right)=\frac{1}{1+\exp\left(-\frac{\Delta E_i}{T}\right)}$$

$$P(h_j=1)=\sigma\left(\frac{\Delta E_j}{T}\right)=\frac{1}{1+\exp\left(-\frac{\Delta E_j}{T}\right)}$$

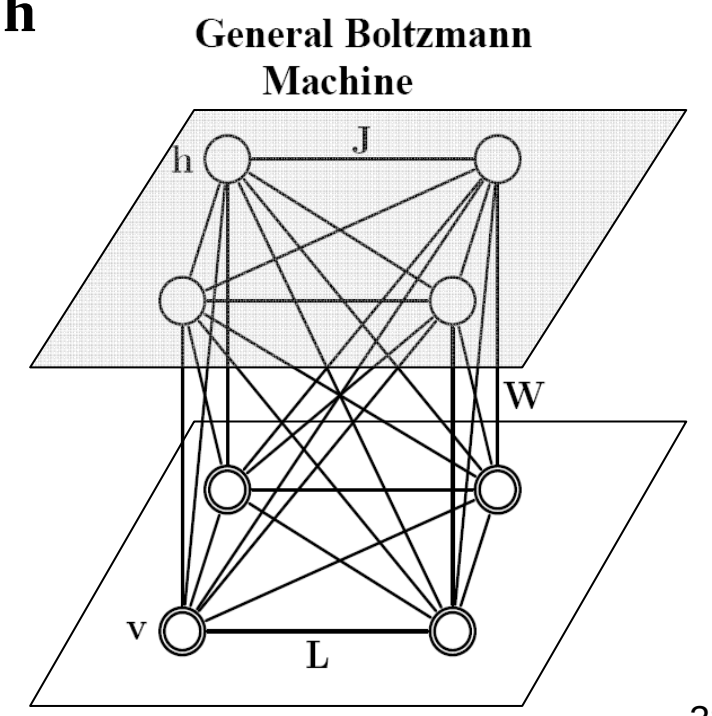
# 玻尔兹曼机 (Boltzmann Machine)

- BM是由随机神经元构成的随机学习模型
  - 网络结构与目标函数
    - 由对称成对儿随机二值单元构成的网络
    - 包含一组可见单元{ v }和隐藏单元{ h }
    - 状态的能量函数为:

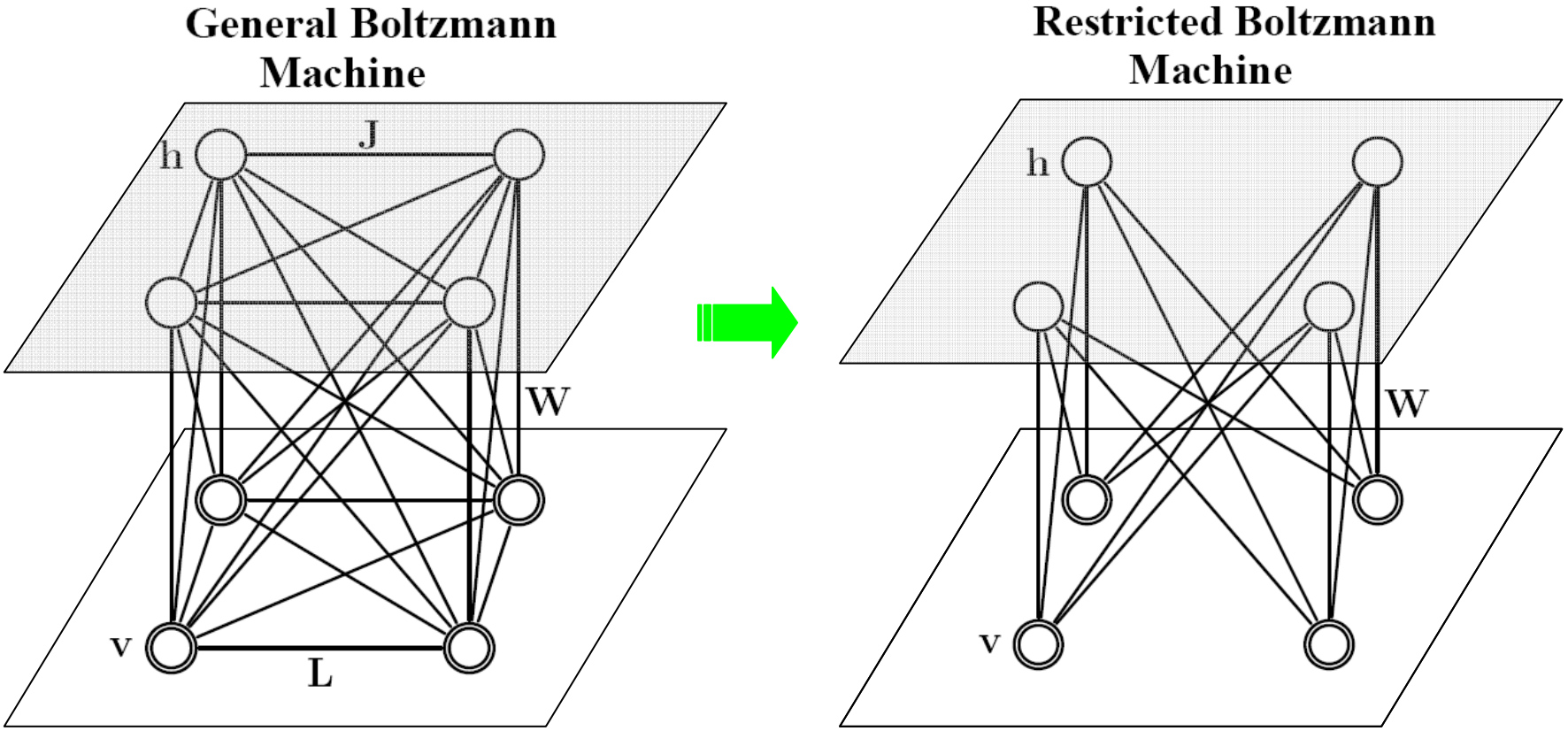
$$E(\mathbf{v},\mathbf{h};\Theta) = -\frac{1}{2}\mathbf{v}^T\mathbf{L}\mathbf{v} - \frac{1}{2}\mathbf{h}^T\mathbf{J}\mathbf{h} - \frac{1}{2}\mathbf{v}^T\mathbf{W}\mathbf{h}$$

其中 $\Theta = \{\mathbf{L},\mathbf{J},\mathbf{W}\}$

- 训练方法
  - 最大化对数似然函数，基于梯度上升法
    - 突触权值的调整仅使用两个不同条件下的局部可观测量



# BM vs. RBM



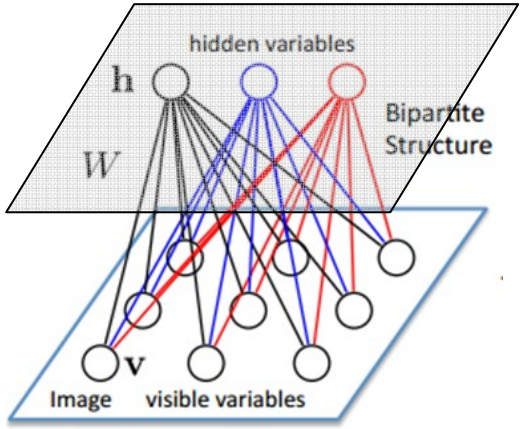

- 在**RBM**中，可见单元之间和隐藏单元之间均无突触权值连接
  - RBM是带隐含变量的MRF (Markov Random Field)且构成无向二部图(bipartite graph)

# 受限玻尔兹曼机 (RBM)

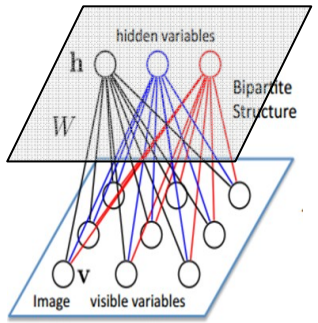
- 网络结构
  - 可见层和隐藏层构成的两层网络
    - 可见(visible)单元{ $v_i$, i=1,...,m }对应于网络输入
    - 隐藏(hidden)单元{ $h_j$, j=1,...,n }对应于特征检测器
      - 输入连接到随机的二值特征检测器，使用对称的权值连接
- 网络的能量函数E(v,h)

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i\in pixels} b_i v_i - \sum_{j\in features} c_j h_j - \sum_{i,j} w_{ij} v_i h_j$$

$$= -\mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{v}^T\mathbf{W}\mathbf{h}$$



# 受限玻尔兹曼机 (RBM)




- 给定1个样本时，网络的似然度

$$p(\mathbf{v},\mathbf{h})=\frac{1}{Z}\exp\{-E(\mathbf{v},\mathbf{h})\}=\frac{1}{Z}\exp\{-\mathbf{b}^T\mathbf{v}-\mathbf{c}^T\mathbf{h}-\mathbf{v}^T\mathbf{W}\mathbf{h}\}$$

  - 其中Z 是partition函数(相当于normalization constant)

– 隐藏单元与可见单元之间相互独立

$$p(\mathbf{v}|\mathbf{h})=\prod_{i=1}^{m}p(v_i|\mathbf{h}),\quad p(\mathbf{h}|\mathbf{v})=\prod_{j=1}^{n}p(h_j|\mathbf{v})$$

– 单个节点的激活概率

$$p(v_i=1|\mathbf{h})=\sigma\left(b_i+\sum_{j=1}^{n}w_{ij}h_j\right),\quad p(h_j=1|\mathbf{v})=\sigma\left(c_j+\sum_{i=1}^{m}w_{ij}v_i\right)$$

其中$\sigma(t)=\frac{1}{1+e^{-t}}$ logistic函数

[1] M. Á. Carreira-Perpiñán and G. Hinton. On contrastive divergence learning. AI&STAT 2005.
[2] Asja Fischer and Christian Igel. Training Restricted Boltzmann Machines: An Introduction. Pattern Recognition 47, pp. 25-39, 2014.

# 受限玻尔兹曼机 (RBM)的训练

- 给定1个样本时，网络可见部分的似然函数

$$p(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp\{-E(\mathbf{v},\mathbf{h})\}$$

  - 其中Z 是partition函数

- 给定一组i.i.d.训练样本V时，似然函数： $\prod_{\mathbf{v}\in V} p(\mathbf{v})$

- 采用最大似然法 $\arg\max_{W,b,c} \log\left(\prod_{\mathbf{v}\in V} p(\mathbf{v})\right)$

  – 计算对数似然函数的梯度

    - 利用Gibbs采样计算近似梯度，基于梯度上升法更新参数
      – E.g. Single Step Contrastive Divergence (CD-1)

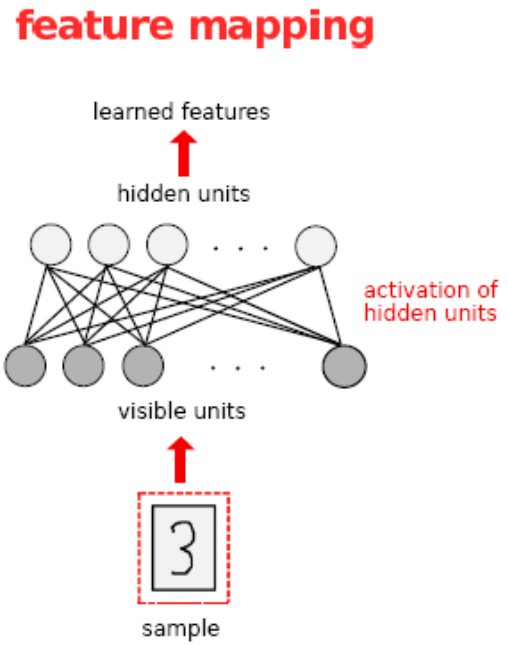
[1] M. Á. Carreira-Perpiñán and G. Hinton. On contrastive divergence learning. AI&STAT 2005.
[2] Asja Fischer and Christian Igel. Training Restricted Boltzmann Machines: An Introduction. Pattern Recognition 47, pp. 25-39, 2014.

# 举例: RBM trained on MNIST

- 数据集 MNIST:

  - **70000个 28x28 手写数字的二值图像**

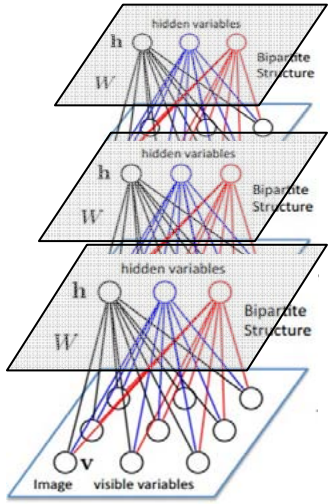
- RBM 结构:
  - **784 x 16**
    - 可见层784个节点
      - 对应于784个像素
    - 隐藏层16个节点(人为设定)
      - 右图对训练好的RBM的权值向量进行可视化

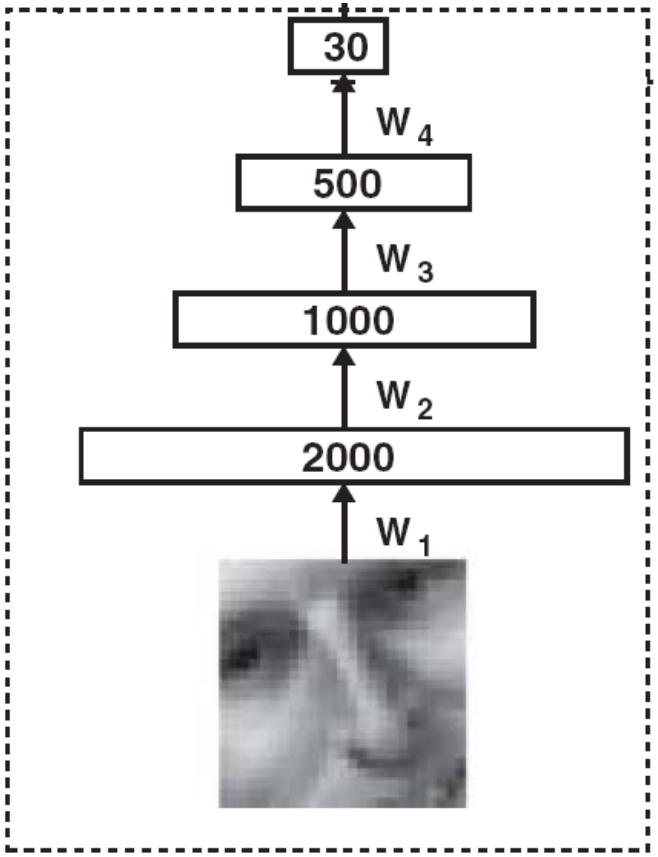



[1] Asja Fischer and Christian Igel. Training Restricted Boltzmann Machines: An Introduction. Pattern Recognition 47, pp. 25-39, 2014.

# 深度的受限玻尔兹曼机

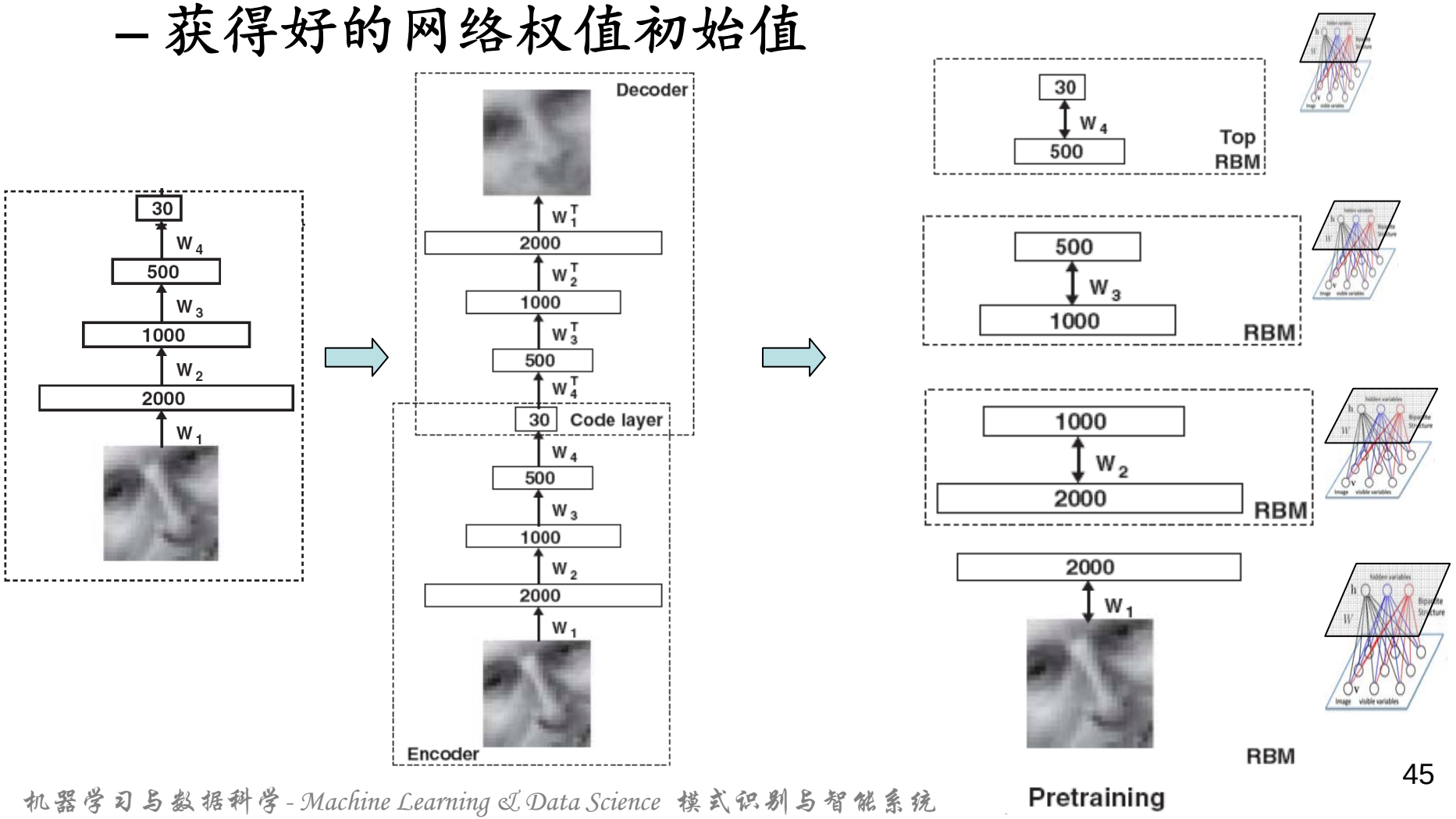
- 堆栈多个RBM：
  - 把上一层**RBM**的输出为下一层**RBM**的输入



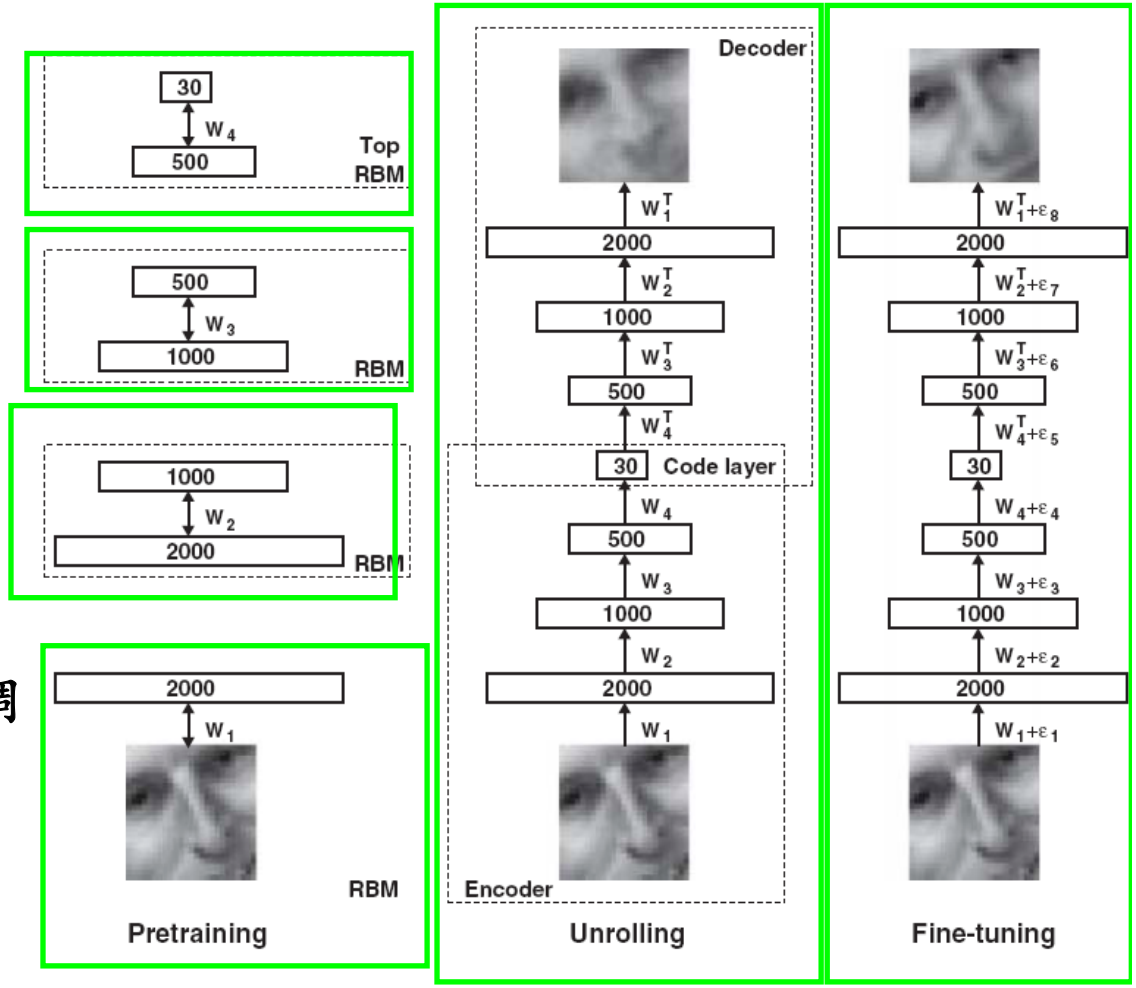
# 深度的受限玻尔兹曼机的预训练

- 逐层训练(Layer-wise Training)
  - 获得好的网络权值初始值



# 深度的受限玻尔兹曼机的完整训练策略

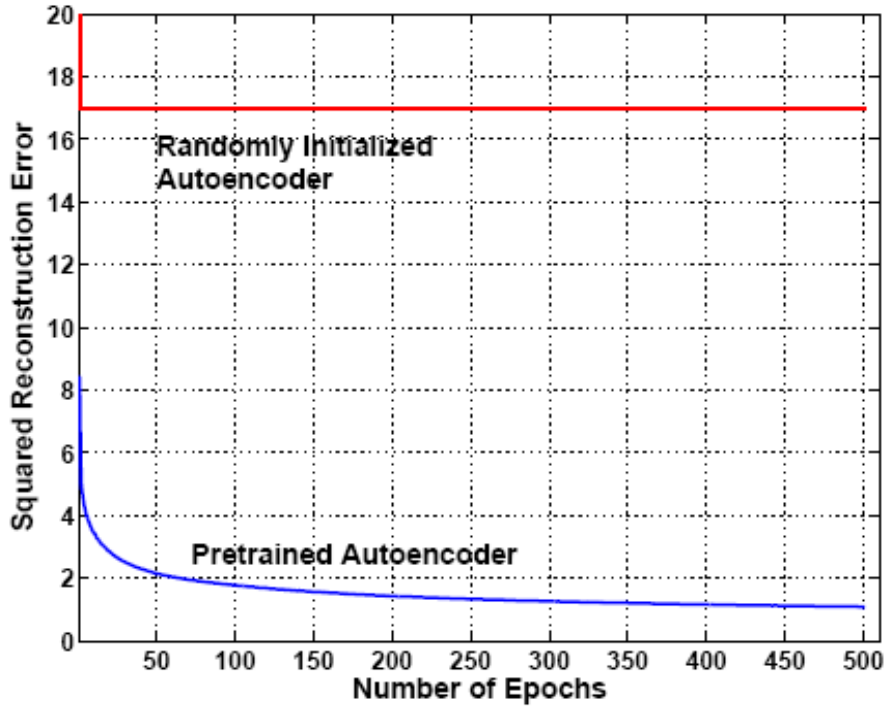
- 1. 逐层预训练
  - 从输入开始逐层训练
- 2. 细调(fine-tuning)
  - 构成深度自编解码
  - 使用反传(BP)算法细调




[1] Hinton G.E. and Salakhutdinov R.R., Reducing the dimensionality of data with neural networks, SCIENCE, Vol.313, July 2006, pp.504-507.

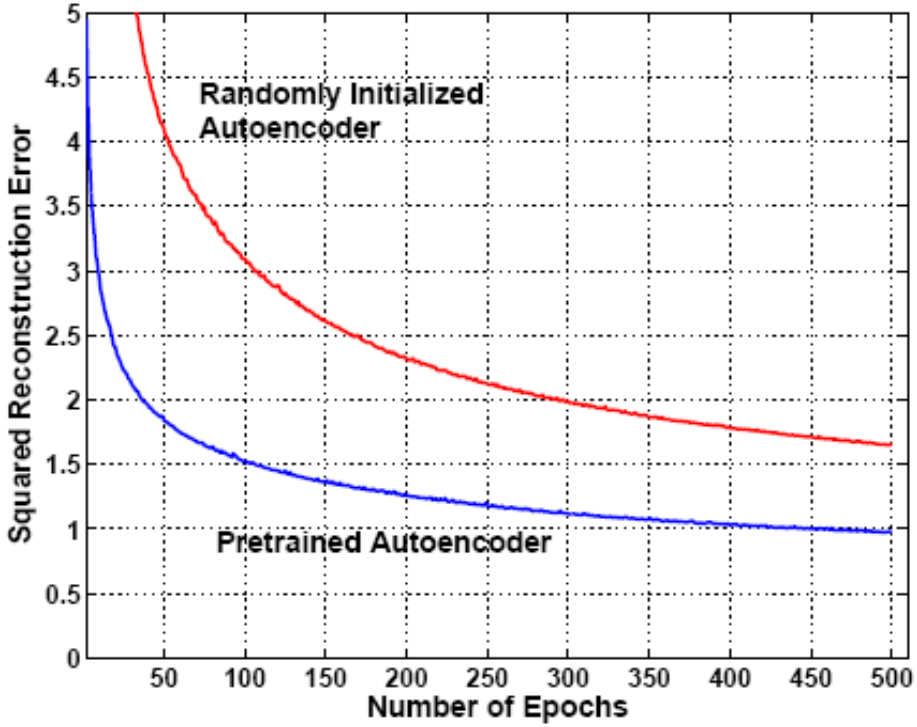
# 深度网络中预训练的作用

- Deep auto-encoder network: 784-400-200-100-50-25-6
  - **makes rapid progress after pre-training**
  - **but no progress without pre-training**

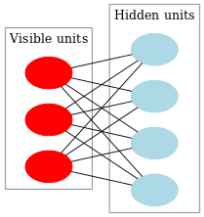


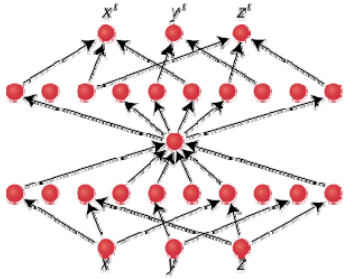
# 浅层网络中预训练的作用

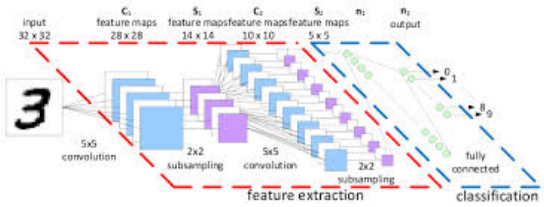
- Shallow auto-encoder: 784 x 532 x 6
  - **can learn without pre-training**
  - **but pre-training makes the fine-tuning much faster**



- **内容提要**
  - 引言
  - 深度学习的基本结构
    - RBM
    - Auto-Encoding Networks
    - CNN
    - RNN / LSTM / GAN / ...
  - 网络训练的新技术
    - 逐层预训练+细调 (Layer-wise pretraining + fine-tuning)
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...





# 深度置信网络 (Deep Belief Networks)

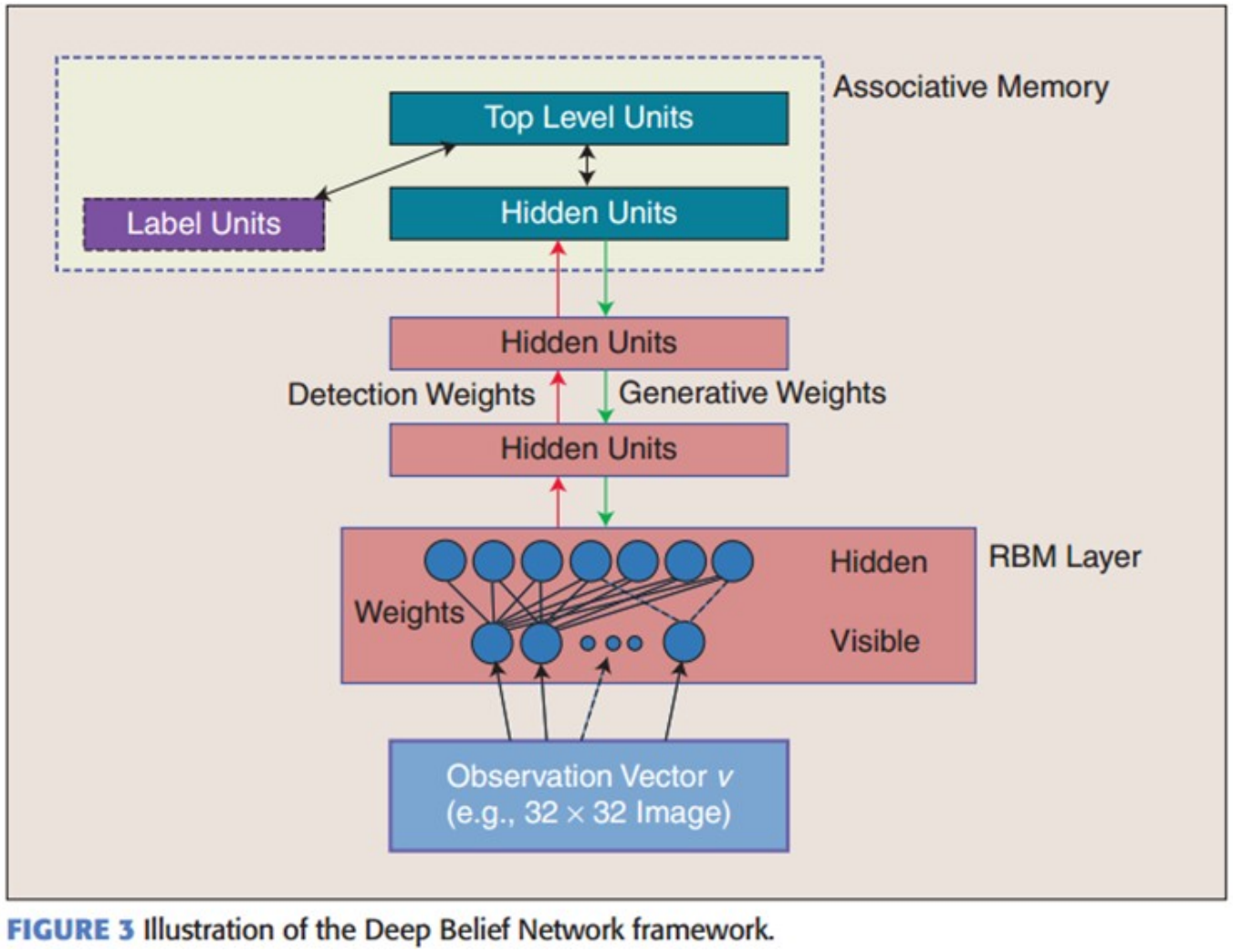
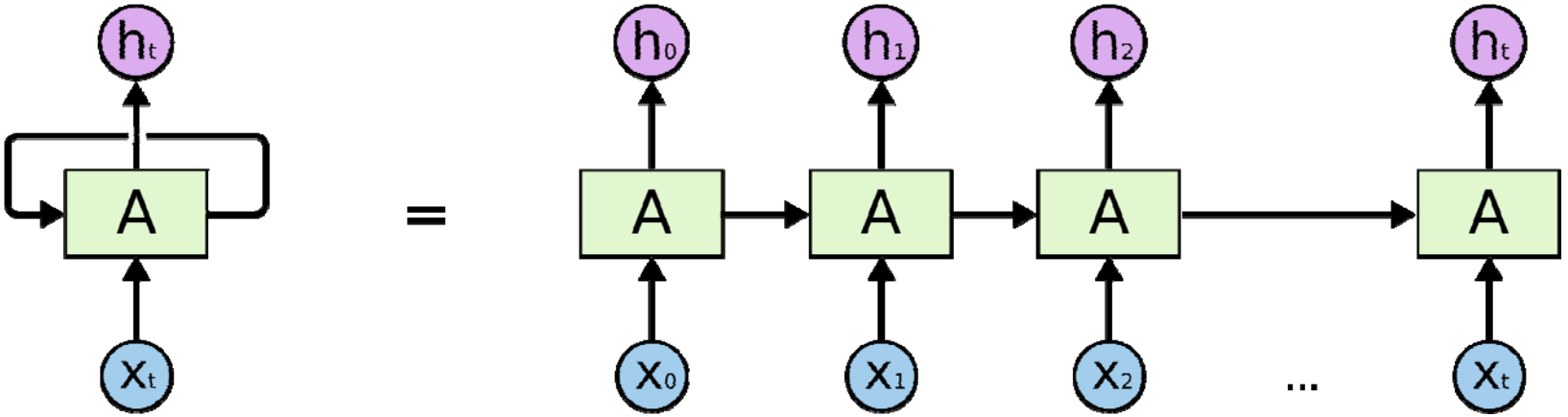
- 


**FIGURE 3** Illustration of the Deep Belief Network framework.

[1] Hinton et al: “A fast learning algorithm for deep belief nets”,Neural Computation,18:1527-1554,2006.

# RNN (Recurrent Neural Networks)

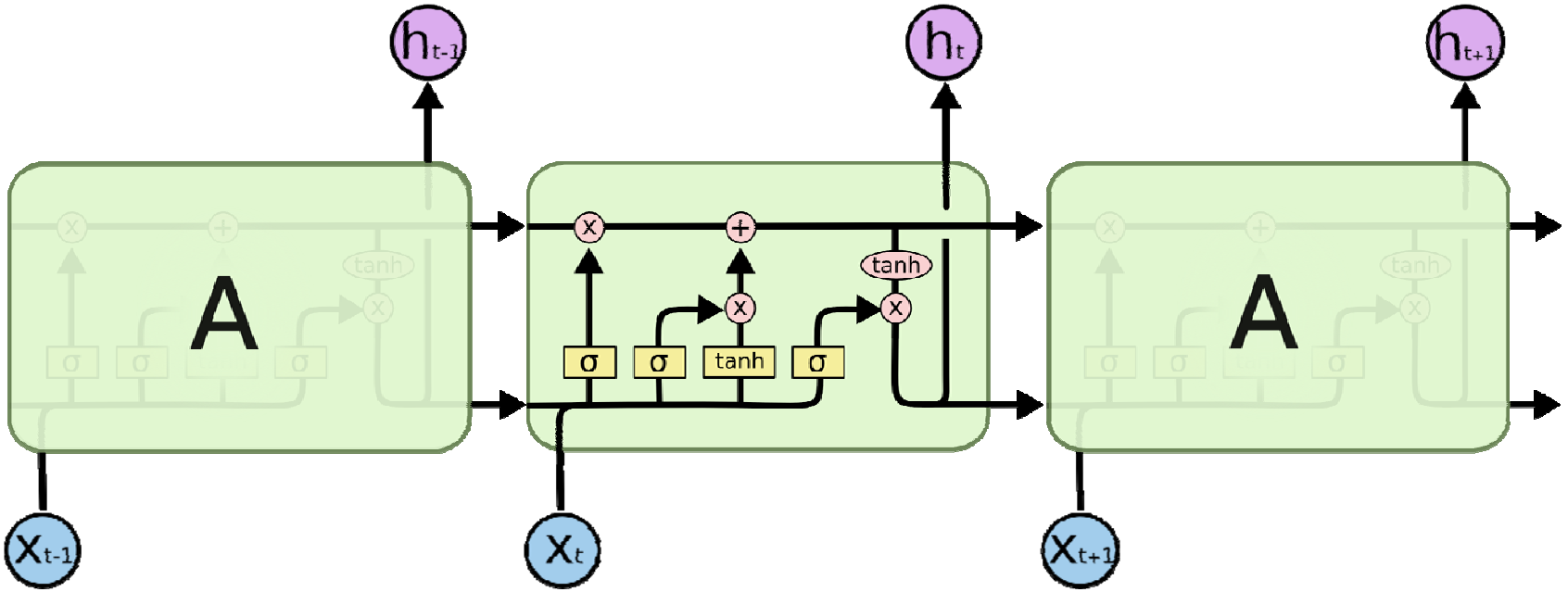
- 递归神经网络




http://qiita.com/KojiOhki

# LSTM (Long Short-Term Memory)

- LSTM




http://qiita.com/KojiOhki

[1] Sepp Hochreiter, Jurgen Schmidhuber, "Long short-term memory." Neural Computation, vol.9, 1997, pp.1735-1780.

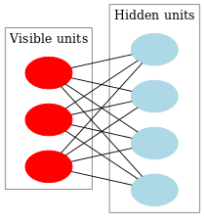
# Generative Adversarial Network

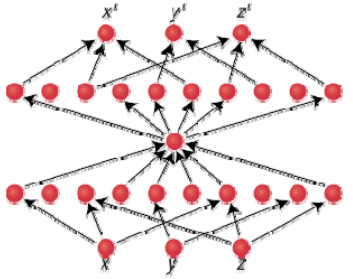
- 生成式对抗网络(GAN)
  - 构造一对模型:
    - 生成模型(G)和判别模型(D)，其中生成模型G用于生成样本，判别模型D用于评价G所生成的模型
    - 两个模型构成一个对抗过程，G生成样本尽量让判别器D分错

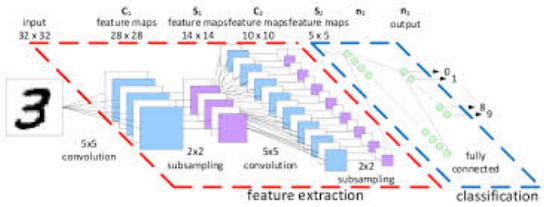
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

[1] Ian J. Goodfellow et al.: Generative Adversarial Networks, NIPS 2014.

- **内容提要**
  - 引言
  - 深度学习的基本结构
    - RBM
    - Auto-Encoding Networks
    - CNN
    - LSTM
  - 网络训练的新技术
    - 逐层预训练+细调 (Layer-wise pre-training + fine-tuning)
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...



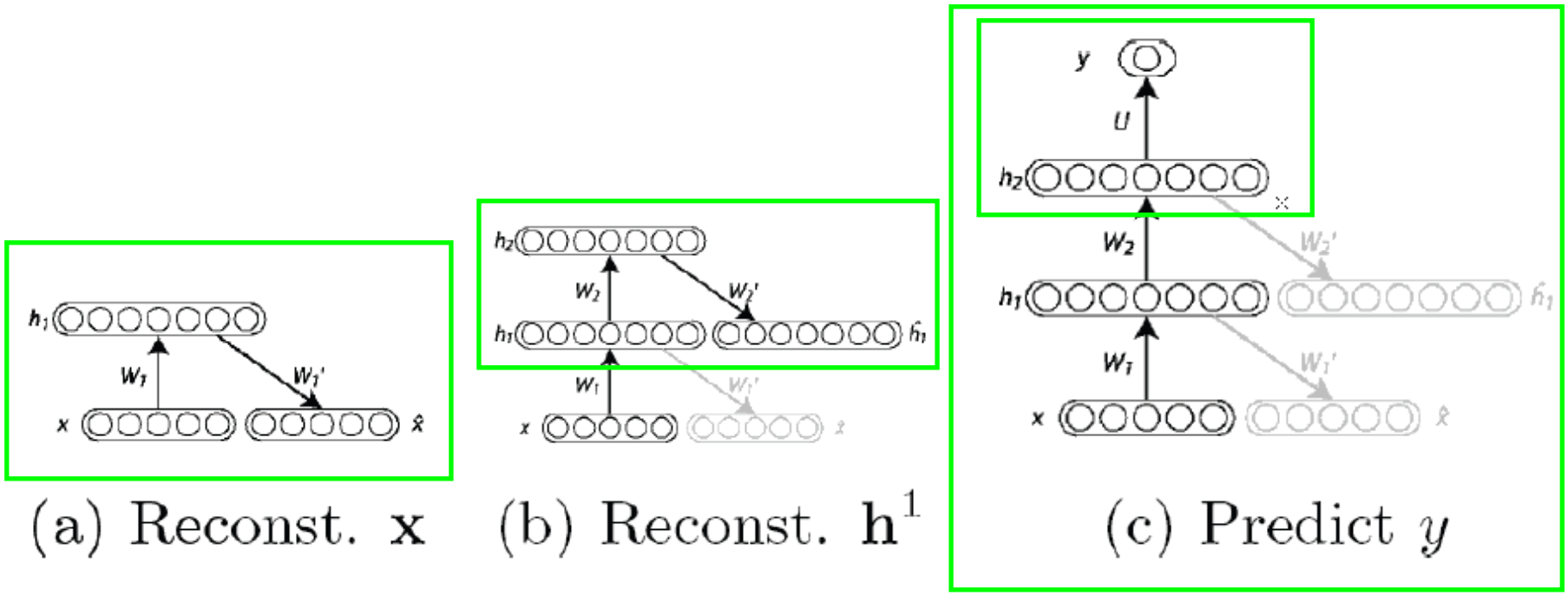





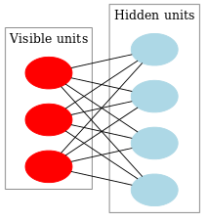
# 深度网络的逐层训练策略

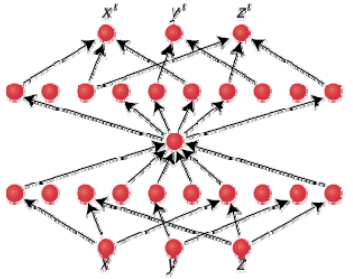
- **Deep Networks:**
  - Encoder-Decoder + Fine-tuning, e.g.,

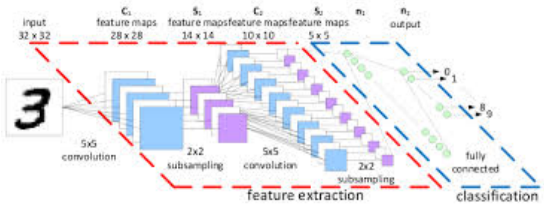
$$\min_{W} \sum_{\mathbf{x}\in D} L\left(\mathbf{x}, g\left(h\left(\mathbf{x}, W\right), W\right)\right) + \Omega\left(h\right)$$



- **内容提要**
  - 引言
  - 深度学习的基本结构
    - RBM
    - Auto-Encoding Networks
    - CNN
    - LSTM
  - 网络训练的新技术
    - 逐层预训练+细调 (Layer-wise pretraining + fine-tuning)
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...

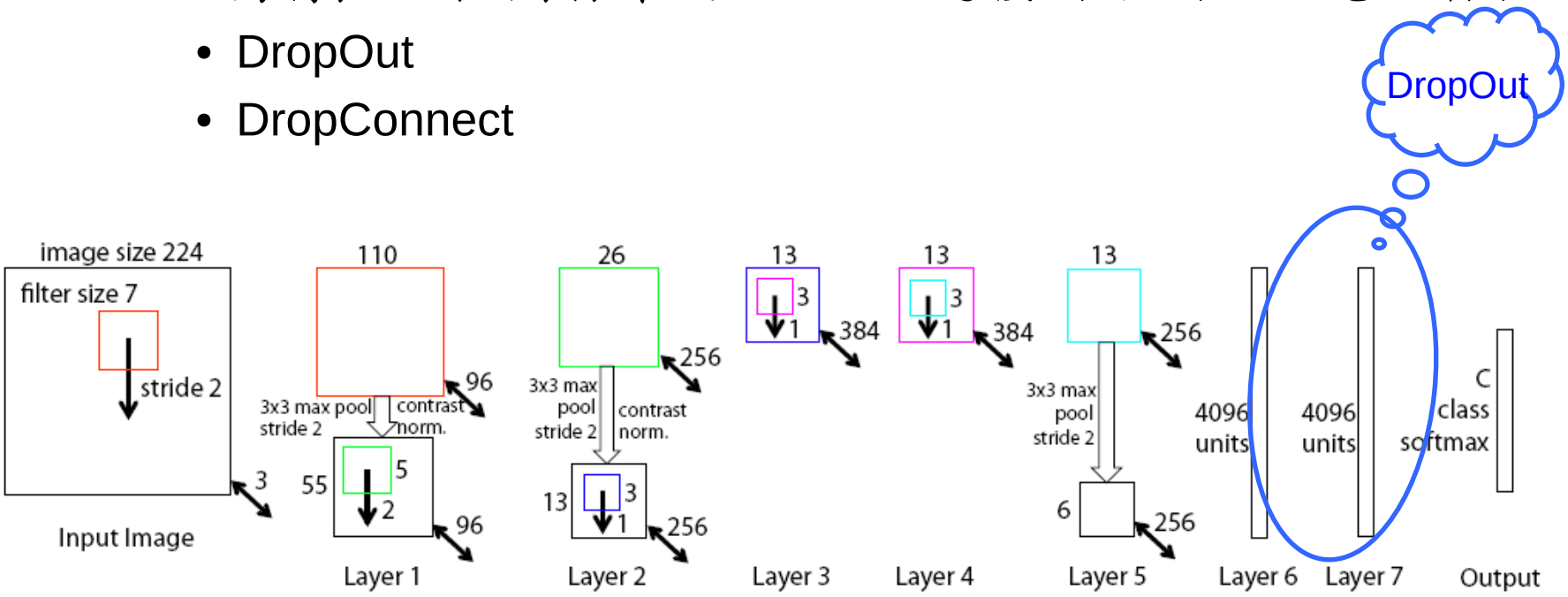




# 网络训练的新技术

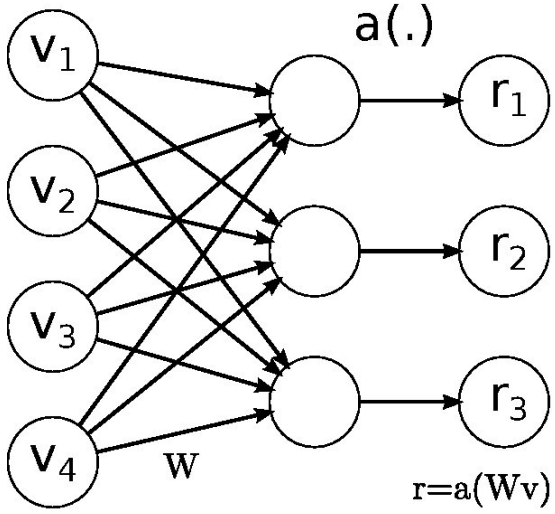
- 在训练大容量全连接网络时，通过“随机化”策略进行“正则化”
  - 结构特点：使用标准的**2-3**层全连接网络（比如感知器）
    - DropOut
    - DropConnect




[1] Hinton, G.E., Krizhevsky, A., Srivastava, N., Sutskever, I., & Salakhutdinov, R. : Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, 15, 1929-1958, 2014.

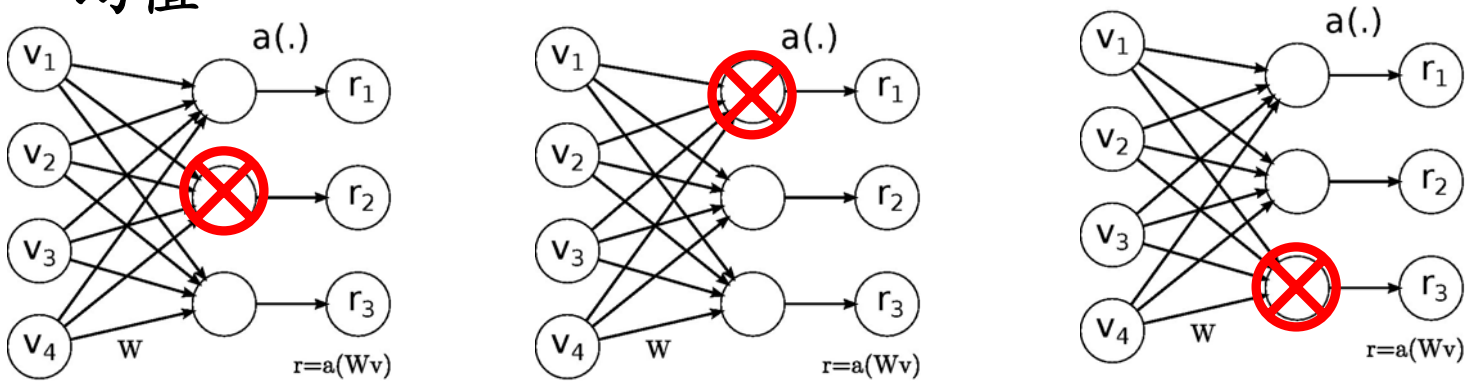
# 网络结构

- 全连接层
  - 比如**3**层感知器网络，**784 x 800 x 800 x 10**
    - 输入维数 784，输出层 10个神经元
    - 具有2个隐藏层，各800个神经元



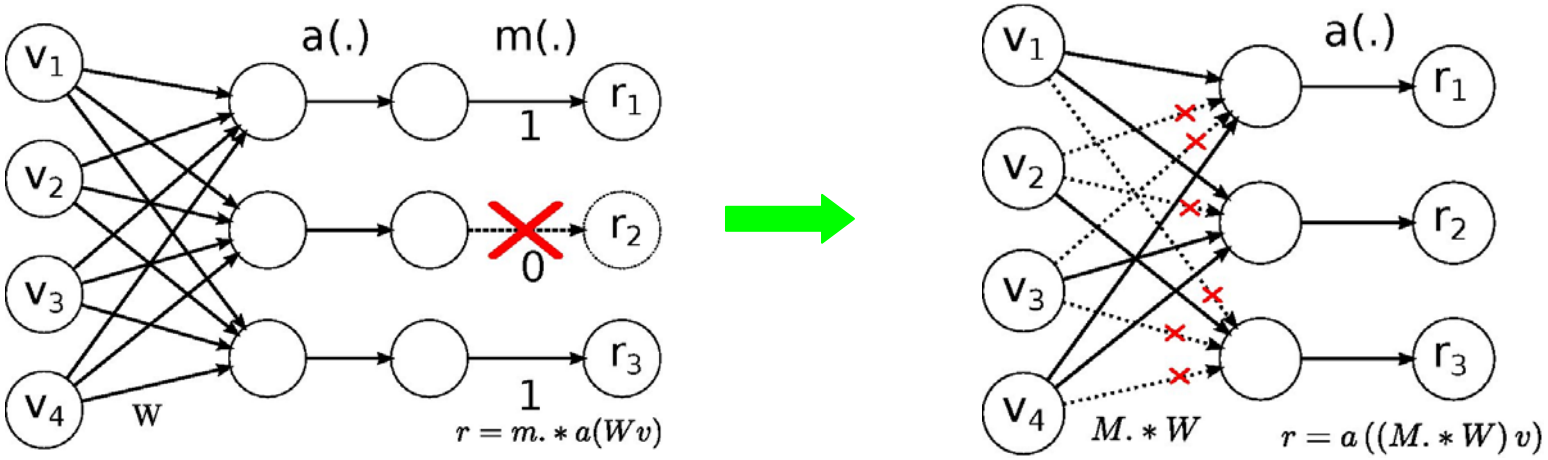
# 提高网络泛化能力的训练策略-1

- DropOut (G. E. Hinton, 2014)
  - 考虑使用**On-line**方式训练具有**1**个隐藏层的**2**层网络
  - 训练阶段: 每次呈现一个样本，以概率**0.5**随机地忽略各个隐藏层神经元
    - 等效于我们随机地从$2^H$个不同体系结构中对结构采样
    - 正则化效果：使权值趋向其它模型想要的值
  - 测试阶段: 从不同结构中采样，取不同输出分布的几何均值

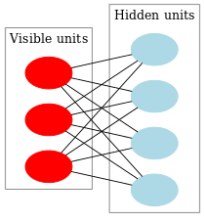


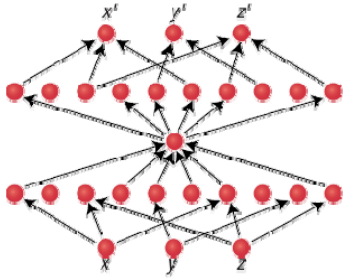
# 提高网络泛化能力的训练策略-2

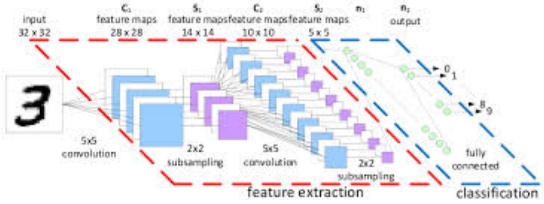
- DropConnect (L. Wan et al. ICML2013)
  - 仅限于在全连接层**(full-connect layer)**使用
  - 在训练时，以概率**p (p=0.5)**随机地忽略各个网络中的连接权值**($w_{ij}$=0)**
  - 在测试时，从网络结构中采样，取均值



- **内容提要**
  - 引言
  - 深度学习的基本结构
    - RBM
    - Auto-Encoding Networks
    - CNN
    - LSTM
  - 网络训练的新技术
    - 逐层预训练+细调 (Layer-wise pretraining + fine-tuning)
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...

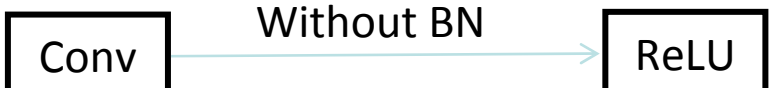




# Batch Normalization

- 在训练过程的mini-batch中，对卷积的输出规范化





$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

[1] Sergey Ioffe, Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift". ICML, 2015.

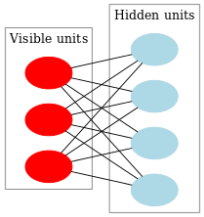
# 其它有用技术

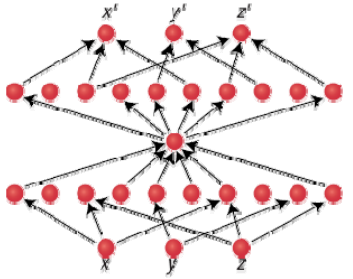
- 初始化
- 激活函数
  - **ReLU**
- 随机梯度下降
  - **Stochastic gradient decent (SGD)**
- 动量随机梯度下降
  - **Momentum SGD**
- Nesterov加速梯度下降
  - **Nesterov accelerated gradient (NAG)**

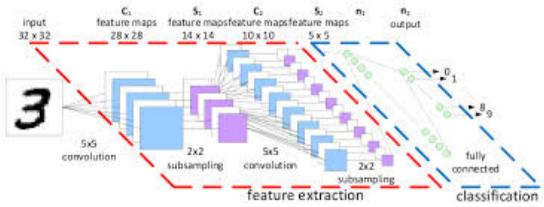
# 举例: ResNet 的部分技术细节

- 图片resize：短边长random.randint(256,480)
- 裁剪：224 * 224随机采样，含水平翻转
- 减均值
- 标准颜色扩充 [2]
- conv和activation间加batch normalization[3]
  - 帮助解决**vanishing/exploding**问题
- minibatch-size: 256
- learning-rate: 初始0.1, error饱和了之后学习速率要除以10
- weight decay：0.0001
- momentum：0.9

- **内容提要**
  - 引言
  - 深度学习的基本结构
    - RBM
    - Auto-Encoding Networks
    - CNN
    - LSTM
  - 网络训练的新技术
    - 逐层预训练+细调 (Layer-wise  pretraining + fine-tuning)
    - 正则化(Regularization), e.g. drop XYZ
    - Batch Normalization
    - ...





# 深度学习主要工具箱

- PyTorch
- Tensoflow (Google)
  - **Python, C++**
- Theano (Université de Montréal)
  - **Python**
- Torch7 (Ronan Collobert, Koray Kavukcuoglu, Clement Farabet)
  - **LUA**
- Caffe (Yangqing Jia, Facebook)
  - **Python, C++, Matlab**
- MxNet (Tianqi Chen et al.)
  - **Python, C++,**
- Matconvnet (VGG Group)
  - **Matlab**
- Keras (François Chollet, Google)
  - **Python**

# 扩展阅读资源

- Books
  - **Deep Learning**
  - **Neural Network and Deep Learning**
- Video courses
  - **CS231n: Convolutional Neural Networks for Visual Recognition (Stanford)**

# Q / A

- Any Question? …