从神经网络到深度学习

唐都钰
dytang@ir.hit.edu.cn

2013.04.10

大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

深度学习

什么是机器学习

- Arthur Samuel (1959)
  – Field of study that gives computers the ability to learn without being explicitly programmed
- Tom Mitchell (1998)
  – A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E

什么是机器学习（续）

- 一封邮件是不是spam？
- 如何在杂乱的环境中识别三维物体？
- 如何判断一比信用卡交易是否正当？

- 我们希望---从数据中学习
  - 以有指导学习为例
    - 输入：实例+正确的输出
    - 输出：模型

机器学习

- Supervised learning
  - 给定输入，预测输出
  - 分类、回归
- Unsupervised learning
  - 挖掘输入数据的好的内在表示(representation)
- Reinforcement learning

有指导学习

- 首先初始化模型 $y = f(\mathbf{x};\mathbf{W})$
  - 借助数值化的参数W，把每个输入向量x，映射为预测的输出y
- 参数学习
  - 调整参数W
    - 减小在训练数据上标准输出结果与预测结果之间的不一致程度
  - 回归 $\frac{1}{2}(y - t)^2$
  - 分类

增强学习

- 输出是一个动作(action)或一系列动作，唯一的有指导信息就是报酬(reward)
  - 目标：选择未来回报最大的动作
- 很困难
  - 报酬一般是比较延时的(delayed)，所以很难知道是哪里做错/对
  - 一个标量的报酬并不能提供很多的信息
  - 不能学习millions 参数，只能学习约1,000量级的参数

无指导学习

- 近40年来被忽视，除了聚类(clustering)
  - Said by Hinton
- 很多学者认为聚类就是无指导学习
- 原因
  - 很难定义无指导学习的目标是什么
  - 传统的主要目的
    - 学习一种输入的内在表示，应用于后续的有指导学习或增强学习

无指导学习(续)

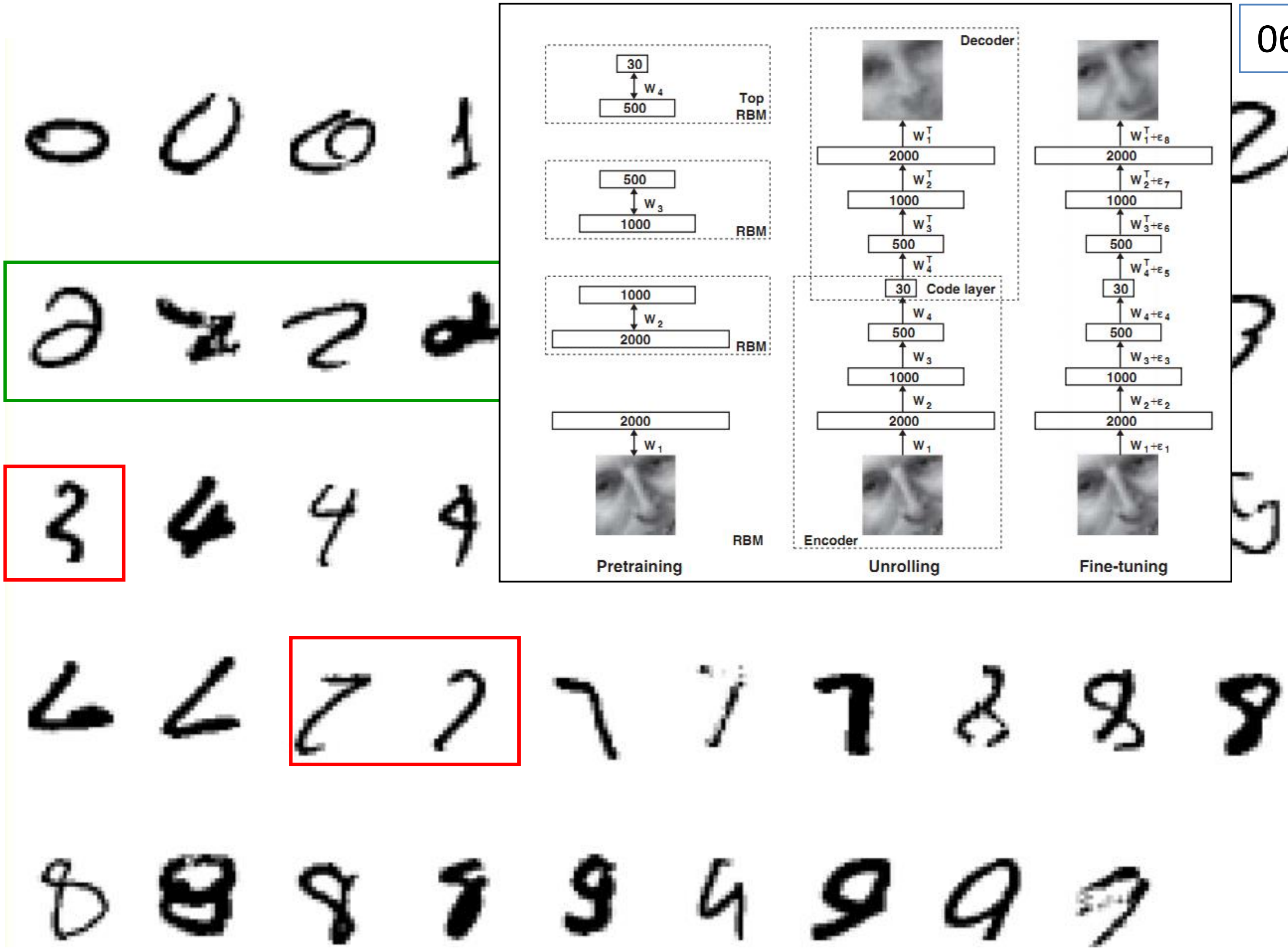
- 无指导学习的目的
  - 提供简洁的、低维度的表示来描述输入数据
  - 提供经济(economical)的高维特征表示
    - 0/1 binary特征
    - 实值特征但大多数为0
  - 聚类
    - 非常稀疏的表示，只有其中一维是1，其余全是0

Deep learning的现状

MNIST & The ImageNet task

- MNIST
  - 手写数字 0~9
  - 训练数据：60,000
  - 测试数据：10,000
- The ImageNet task
  - 来自Web的大规模高分辨率的图像
  - 1000个不同类别
  - 130万图片

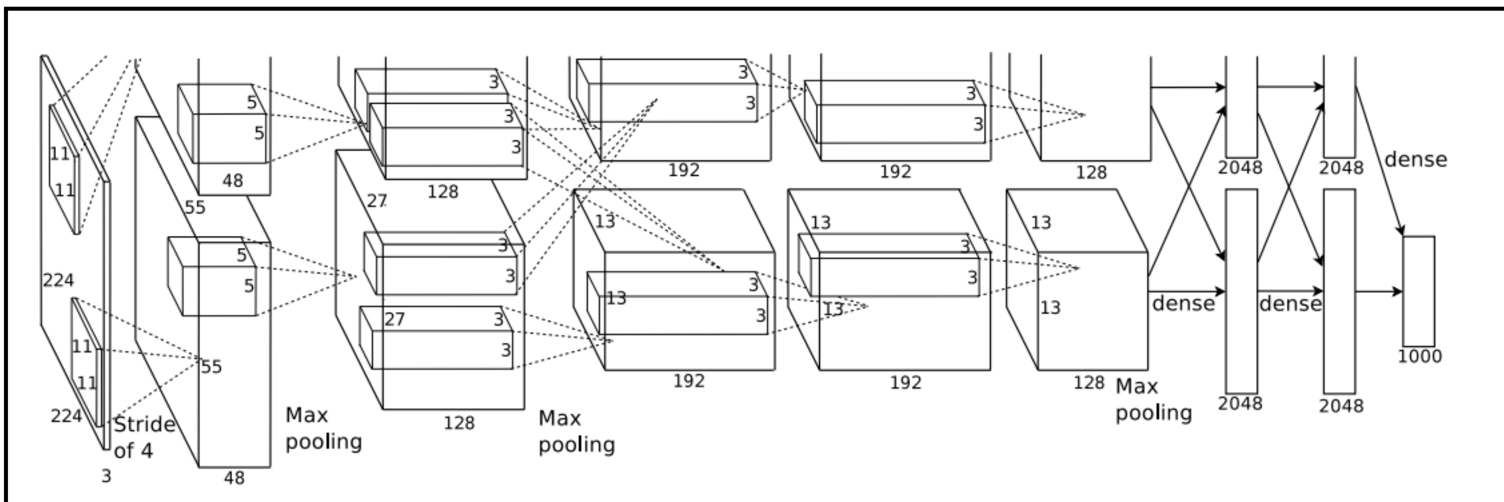
MNIST-识别数字“2”



06 Hinton

The ImageNet task

- 2010 竞赛最佳队伍
  - Top1 错误率=47% Top5 错误率=25%
- 2012 NIPS best paper
  - Top1 错误率=40% Top5 错误率=20%

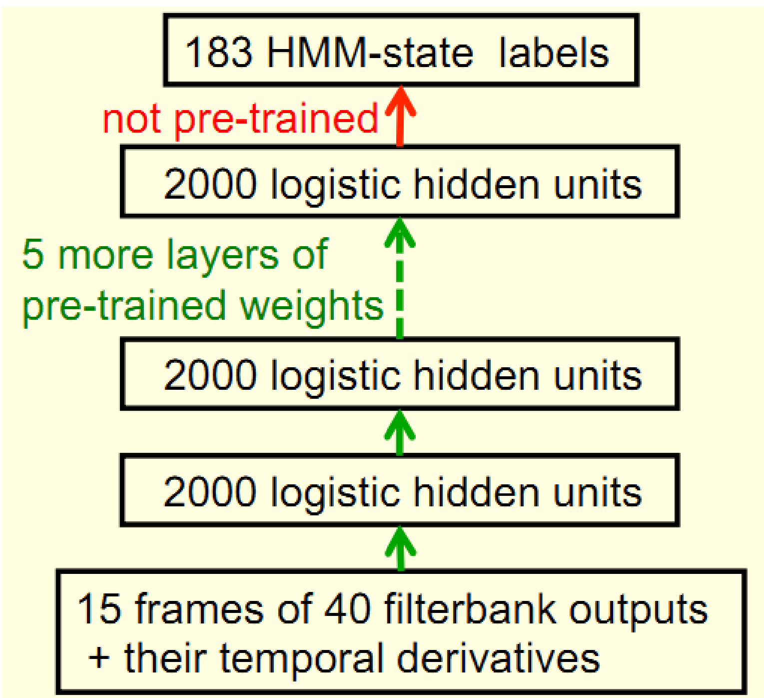


语音识别

- 传统语音识别系统(HMM based)
  - 信号处理
    - 将输入信号转换为向量形式的声学
  - 声学(acoustic)模型
    - 声音到音素的概率
  - 语言模型
    - 统计语言模型是用概率统计的方法来揭示语言单位内在的统计规律，N-Gram简单有效，被广泛使用。
  - 解码
    - 寻找满足声学模型和语言模型的最优序列，如Viterbi算法

语音识别 on TIMIT

- George Dahl 2012年的深度神经网络系统替换了传统的声学模型



- Dahl2012
  - 8层神经网络
  - 错误率：20.7%
- 之前最好结果
  - 错误率：24.4%
  - 融合多种方法

自然语言处理

- 很多机构在开展深度学习研究
- 但目前深度学习在自然语言处理方面还没有产生系统性的突破。
  - http://baike.baidu.com.cn/view/9964678.htm

大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习

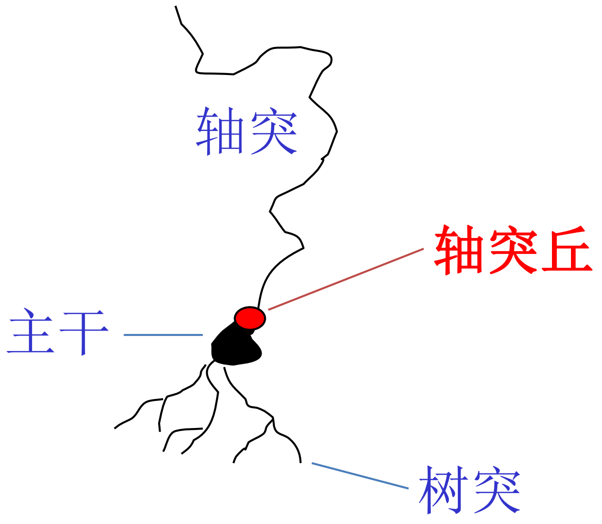
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

深度学习

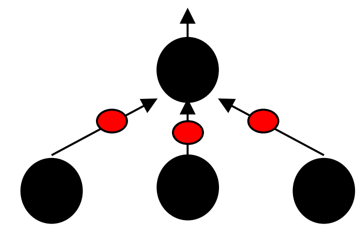
皮质神经元

- 物理结构
  - 每个神经元只有1个轴突
  - 每个神经元有很多树突，接受其他神经元的信号输入
  - 轴突丘生成脉冲信号



大脑如何工作

- 每个神经元从其他神经元接受信
- 神经元之间的影响程度取决于轴突上的权重



- 权重会适应整个网络来完成有效的计算
  - 识别物体、理解语言、控制身体
- 人大约有10^{11}神经元，每个神经元有大约10^4个权重

理想化神经元

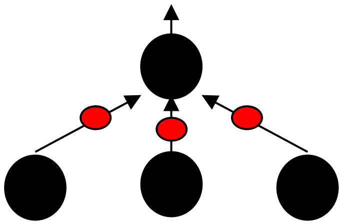
- To model things we have to idealize them (e.g. atoms) --Hinton
- 理想化的好处
  - 忽略很多与问题本质无关的琐碎细节
  - 可以通过数学运算和类比应用到类似的系统
  - 理解了基本的概念，很容易再添加复杂因素并使得系统更可信
- 理想化的问题
  - 模型本身是错的！但是我们需要记住自己是错的
  - 如：神经元传递式实数值的信号而不是离散的脉冲

神经元

- Linear Neurons(线性神经元)
- Binary Threshold Neurons
- Linear threshold Neurons
- Sigmoid neurons
- Stochastic binary neurons

(1) 线性神经元(Linear nearons)

- 简单+计算能力有限



bias

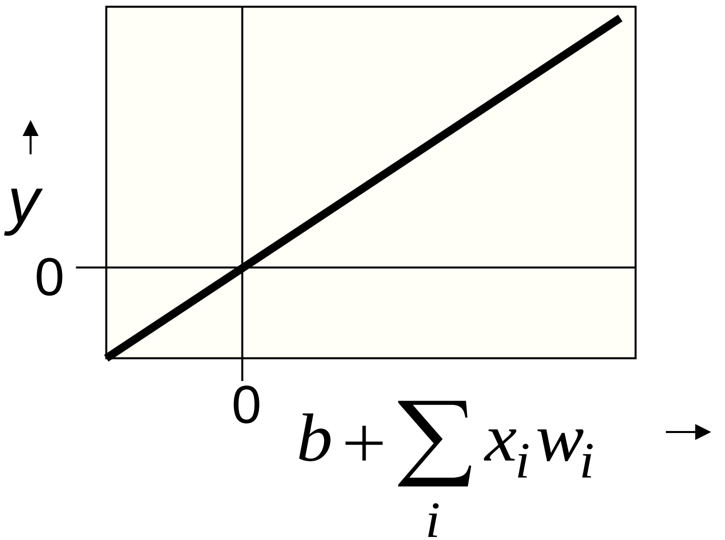
i th input

$$y = b + \sum_i x_i w_i$$

output

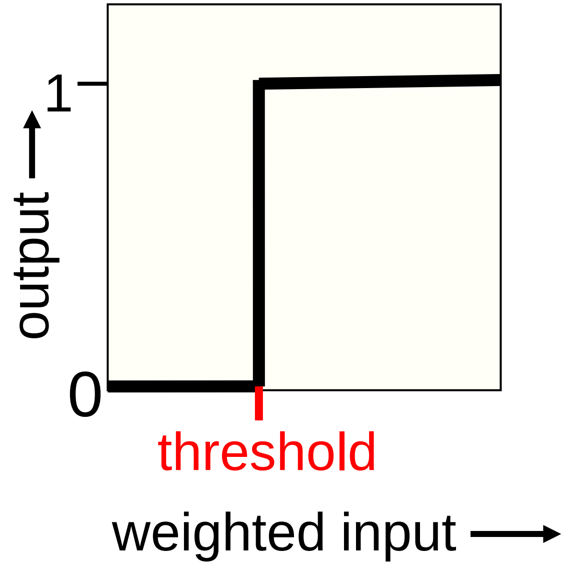
index over input connections

weight on i^{th} input



(2) Binary Threshold Neurons

- McCulloch-Pitts (1943)
  - 计算输入数据的加权求和
  - 如果加权和大于某个阈值，则输出固定值的脉冲，如“1”



(2) Binary Threshold Neurons(Cont.)

- 2种等价的方式定义Binary Threshold Neurons

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 \text{ if } z \geq \theta \\ 0 \text{ otherwise} \end{cases}$$

$$\boxed{\theta = -b}$$

$$z = b + \sum_i x_i w_i$$

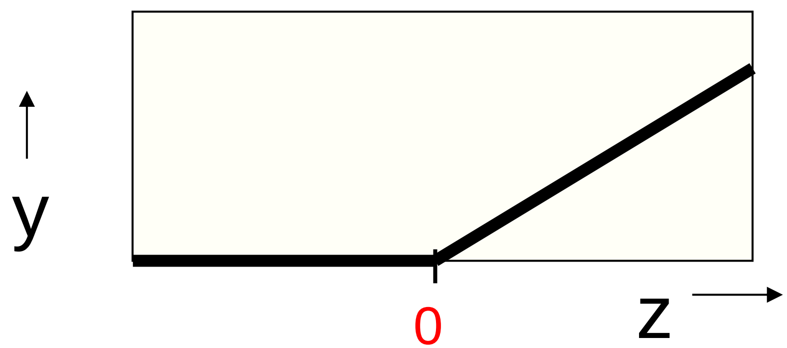
$$y = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

(3) Linear threshold Neurons

- 也叫Rectified Linear Neurons
- 流程
  - 首先计算输入数据的加权线性求和
  - 输出是相对输入的非线性函数

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$$

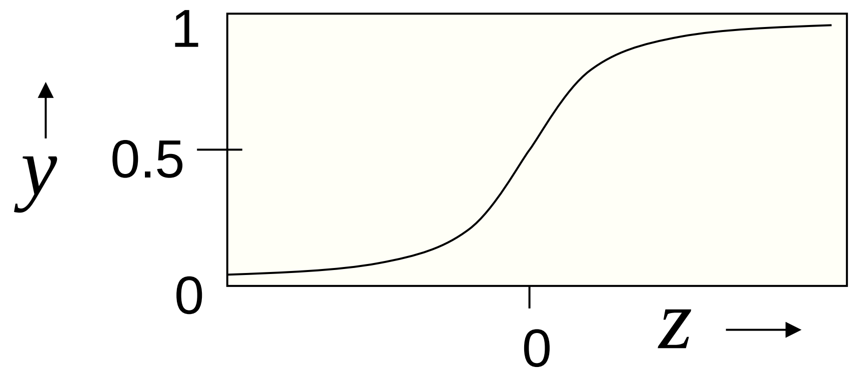


(4) Sigmoid neurons

- 输出是相对输入的实值、连续、有界的函数
  - 通常使用logistic函数
  - 导数形式很优美

$$z = b + \sum_i x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$

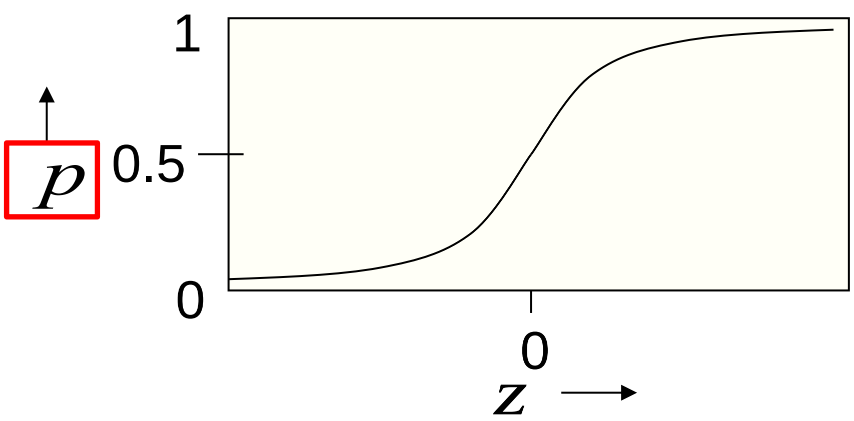


(5) Stochastic binary neurons

- 和Sigmoid Neurons公式一样
  - 但是输出为0、1，以p(s=1)概率输出1
  - 以sigmoid的结果输出值为1

$$z = b + \sum_i x_i w_i$$

$$p(s=1) = \frac{1}{1+e^{-z}}$$



大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

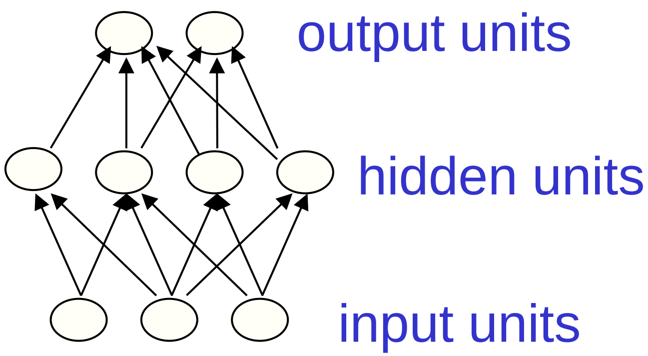
深度学习

神经网络

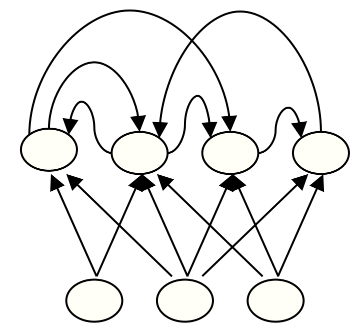
- 前馈神经网络
  - Feed-Forward Neuron Network
- 循环网络
  - Recurrent Networks
- 对称联通网络
  - Symmetrically connected networks

前馈神经网络

- 前馈神经网络是最普通的神经网络
  - 第一层是输入层
  - 最后一层是输出层
  - 如果包含多于一个隐含层
    - 称为深度神经网络
    - Deep Neuron Networks
- 每个神经元的激活值都由其下层输入神经元的非线性函数计算得到
  - 隐含层可以作为输入层的表示



循环网络



- 可以存在有向环
  - 有机会沿着有向边回到起始节点
- 很复杂 很难训练
  - 近年有很多研究工作寻找有效的训练方法
  - 原因
    - 很强大！
    - 能够很好地逼近生物学

对称联通网络

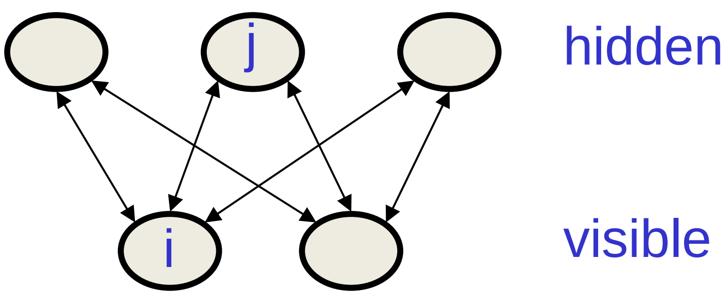
- 和循环网络(Recurrent Networks)类似
  - 但是节点之间的连接是对称的(Symmetric)
    - They have the same weight in both directions
  - John Hopfield表示
    - 对称网络比循环网络更容易分析
    - 但是同时对称网络更受限

对称联通网络(续)

- 不包含隐含节点的对称联通网络叫做Hopfield Nets
- 包含隐含节点的对称联通网络叫做玻尔兹曼机(Boltzmann machines)
- 玻尔兹曼机
  - 比Hopfield Nets更强大
  - 没有循环网络强大
  - 拥有漂亮、简单的学习算法

受限制的玻尔兹曼机

- Restricted Boltzmann Machine
- 只有一层隐含节点
- 隐含层之间没有连接



大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

深度学习

权重学习

- 线性神经元
- Sigmoid Neurons
- 反向传播算法

线性神经元

- 实值的输出
  - 输出是输入的线性加权求和

$$y = b + \sum_i x_i w_i \quad \Rightarrow \quad y = \sum_i w_i x_i = \mathbf{w}^T \mathbf{x}$$

  - 学习权重w
    - 在训练数据集上使得 (正确结果-预测结果)^2 最小

$$E = \frac{1}{2} \sum_{n \in training} (t^n - y^n)^2$$

线性神经元

- 简单推导

$$y = \sum_i w_i x_i = \mathbf{w}^T \mathbf{x}$$

$$E = \frac{1}{2} \sum_{n \in training} (t^n - y^n)^2$$

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_n \frac{\partial y^n}{\partial w_i} \frac{dE^n}{dy^n}$$

$$= - \sum_n x_i^n (t^n - y^n)$$

delta-rule

$$\Delta w_i = -\varepsilon \frac{\partial E}{\partial w_i} = \sum_n \varepsilon x_i^n (t^n - y^n)$$

例子—小沛的故事

- 小沛每天都在快餐店吃午饭
  - 只吃鱼(fish)、薯条(chips)和番茄酱(ketchup)
  - 每样都只会要几小份

  - 收银员只告诉小沛每天一共消费多少钱，但不告诉小沛每样菜每份多少钱
  - 小沛吃了很多天，有很多数据，想要算出鱼、薯条和番茄酱的单价。

小沛的故事(续)

- 价格计算公式

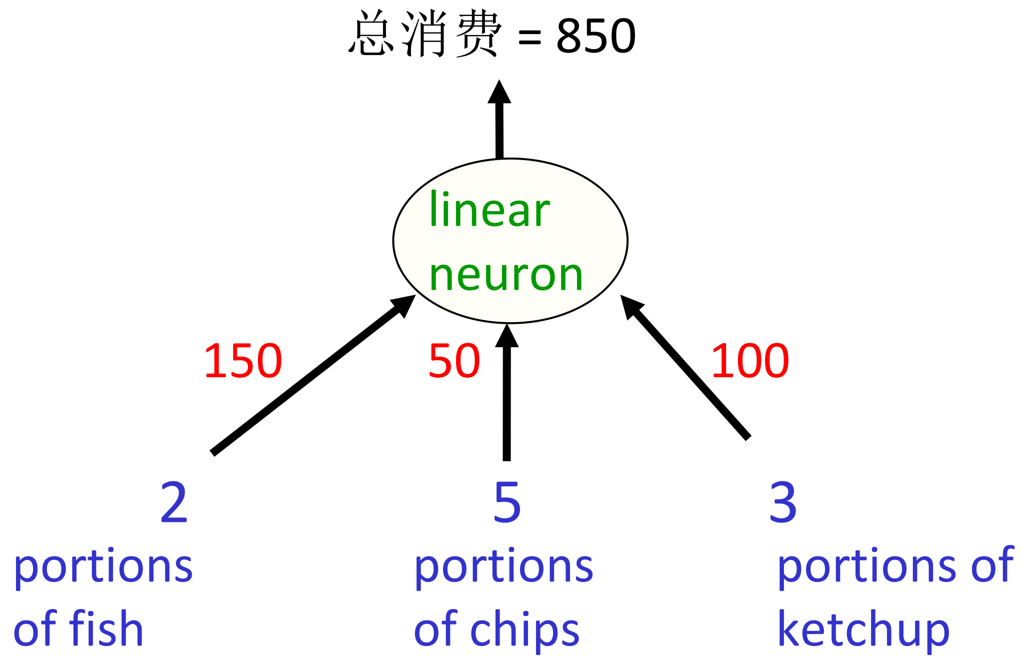
$$price = x_{fish}w_{fish} + x_{chips}w_{chips} + x_{ketchup}w_{ketchup}$$

  - 价格　　price
  - 每样的购买份数　　$\mathbf{X} = (x_{fish}, x_{chips}, x_{ketchup})$
  - 每份多少钱　　$\mathbf{w} = (w_{fish}, w_{chips}, w_{ketchup})$

  - 可以把每天的总消费price看作是购买份数X的线性加权求和，权重为W。即每份菜的价格

小沛的故事(续)

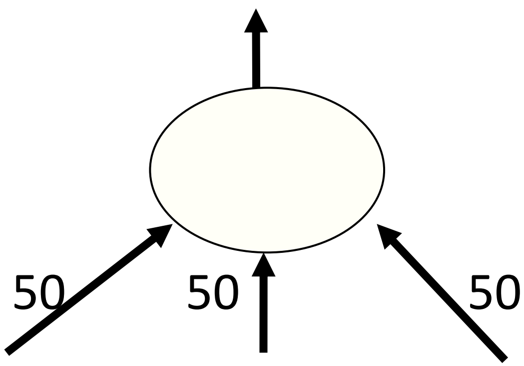
- 收银员
  - 实际的权重 150, 50, 100



小沛的故事(续)

- 任意随机初始值　　$\mathbf{w} = (w_{fish}, w_{chips}, w_{ketchup})$
  - 50, 50, 50

猜测的价格= 500



2 portions of fish　　5 portions of chips　　3 portions of ketchup

y-t = 850 – 500 = 350

delta-rule

$$\Delta w_i = -\varepsilon \frac{\partial E}{\partial w_i} = \sum_n \varepsilon x_i^n (t^n - y^n)$$

假设 ε = 1/35

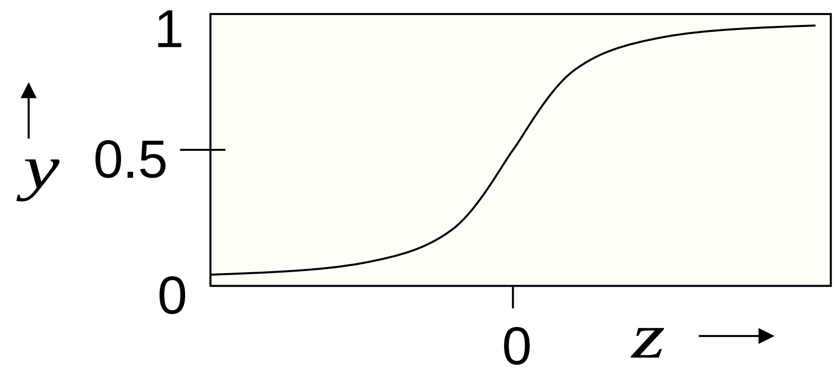
权重变化分别为：+20, +50, +30

新的权重为：70, 100, 80

Sigmoid Neurons

- 输出实数值
  - 输出是输入的非线性、平滑、有界的函数

$$z = b + \sum_i x_i w_i \qquad y = \frac{1}{1+e^{-z}}$$



$$\frac{\partial z}{\partial w_i} = x_i \qquad \frac{dy}{dz} = y(1-y)$$

Sigmoid Neurons

$$y = \frac{1}{1+e^{-z}} = (1+e^{-z})^{-1}$$

$$\frac{dy}{dz} = \frac{-1(-e^{-z})}{(1+e^{-z})^2} = \left(\frac{1}{1+e^{-z}}\right)\left(\frac{e^{-z}}{1+e^{-z}}\right) = y(1-y)$$

$$\frac{e^{-z}}{1+e^{-z}} = \frac{(1+e^{-z})-1}{1+e^{-z}} = \frac{(1+e^{-z})}{1+e^{-z}}\frac{-1}{1+e^{-z}} = 1-y$$

Sigmoid Neurons

- 推导—链式法则(Chain Rule)

$$\frac{\partial y}{\partial w_i} = \frac{\partial z}{\partial w_i}\frac{dy}{dz} = x_i\, y\,(1-y)$$

$$E = \frac{1}{2}\sum_{n \in training} (t^n - y^n)^2$$

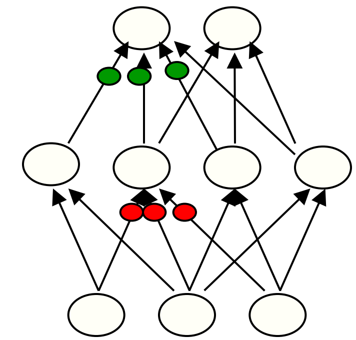
delta-rule

$$\frac{\partial E}{\partial w_i} = \sum_n \frac{\partial y^n}{\partial w_i}\frac{\partial E}{\partial y^n} = -\sum_n \boxed{x_i^n}\,\boxed{y^n\,(1-y^n)}\,\boxed{(t^n - y^n)}$$

extra term = slope of logistic

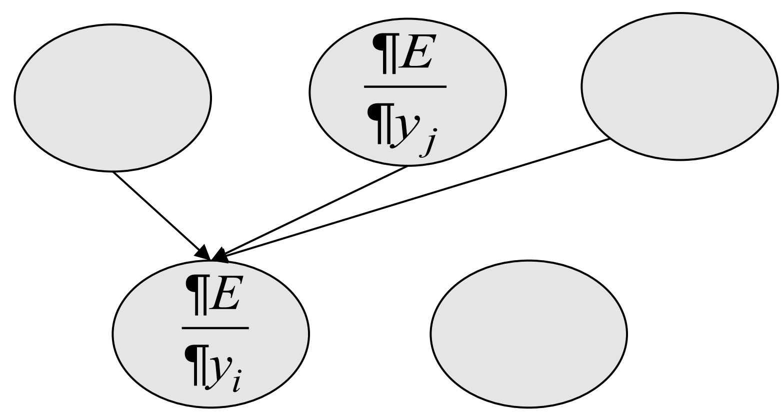
反向传播算法

- 处理包含隐含层的情况
- 初衷
  - 不知道隐含层应该输出什么
  - 但是隐含层的结果可以影响输出层

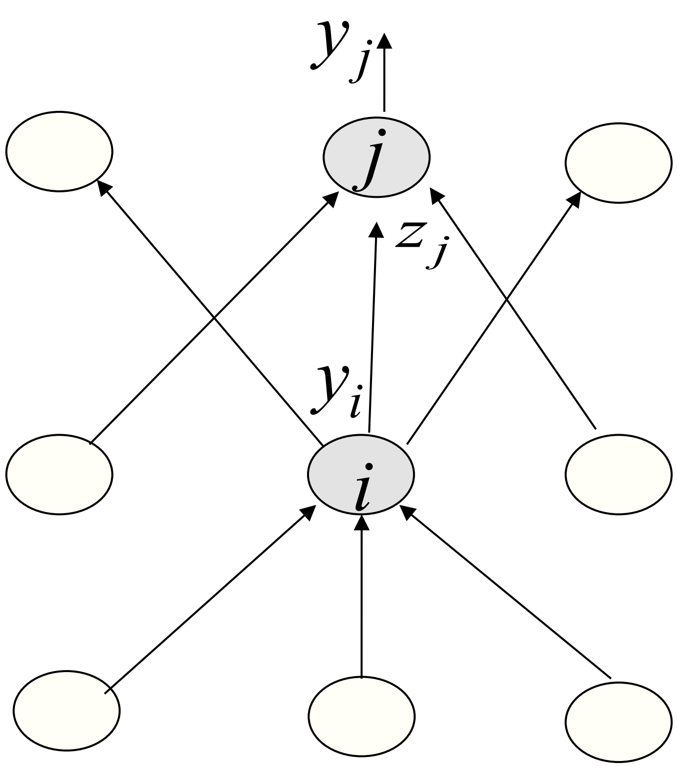


$$E = \frac{1}{2} \sum_{j \in output} (t_j - y_j)^2$$

$$\frac{\partial E}{\partial y_j} = -(t_j - y_j)$$



反向传播算法(续)



$$\frac{\partial E}{\partial z_j} = \frac{dy_j}{dz_j}\frac{\partial E}{\partial y_j} = y_j(1-y_j)\frac{\partial E}{\partial y_j}$$

$$\frac{\partial E}{\partial y_i} = \sum_j \frac{dz_j}{dy_i}\frac{\partial E}{\partial z_j} = \sum_j w_{ij}\frac{\partial E}{\partial z_j}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial z_j}{\partial w_{ij}}\frac{\partial E}{\partial z_j} = y_i\frac{\partial E}{\partial z_j}$$

大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

深度学习

深度学习(deep learning)

- Hinton

How to learn multi-layer generative models of unlabelled data by learning one layer of features at a time.

07 NIPS

- Yoshua Bengio

Deep learning algorithms attempt to learn multiple levels of representation of increasing complexity/abstraction

12 ICML

深度学习(Deep Learning)

- Andrew Ng

> "Deep learning" has had two big ideas:
> – Learning multiple layers of representation
> – Learning features from unlabeled data

2012 NIPS

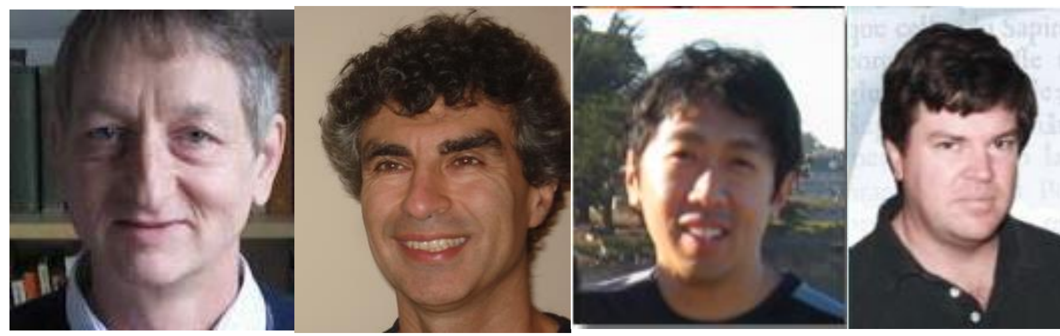
- Jeff Dean

> **Deep Learning**
>
> Algorithmic approach
>
> - automatically learn high-level representations from raw data
> - can learn from both labeled and unlabeled data

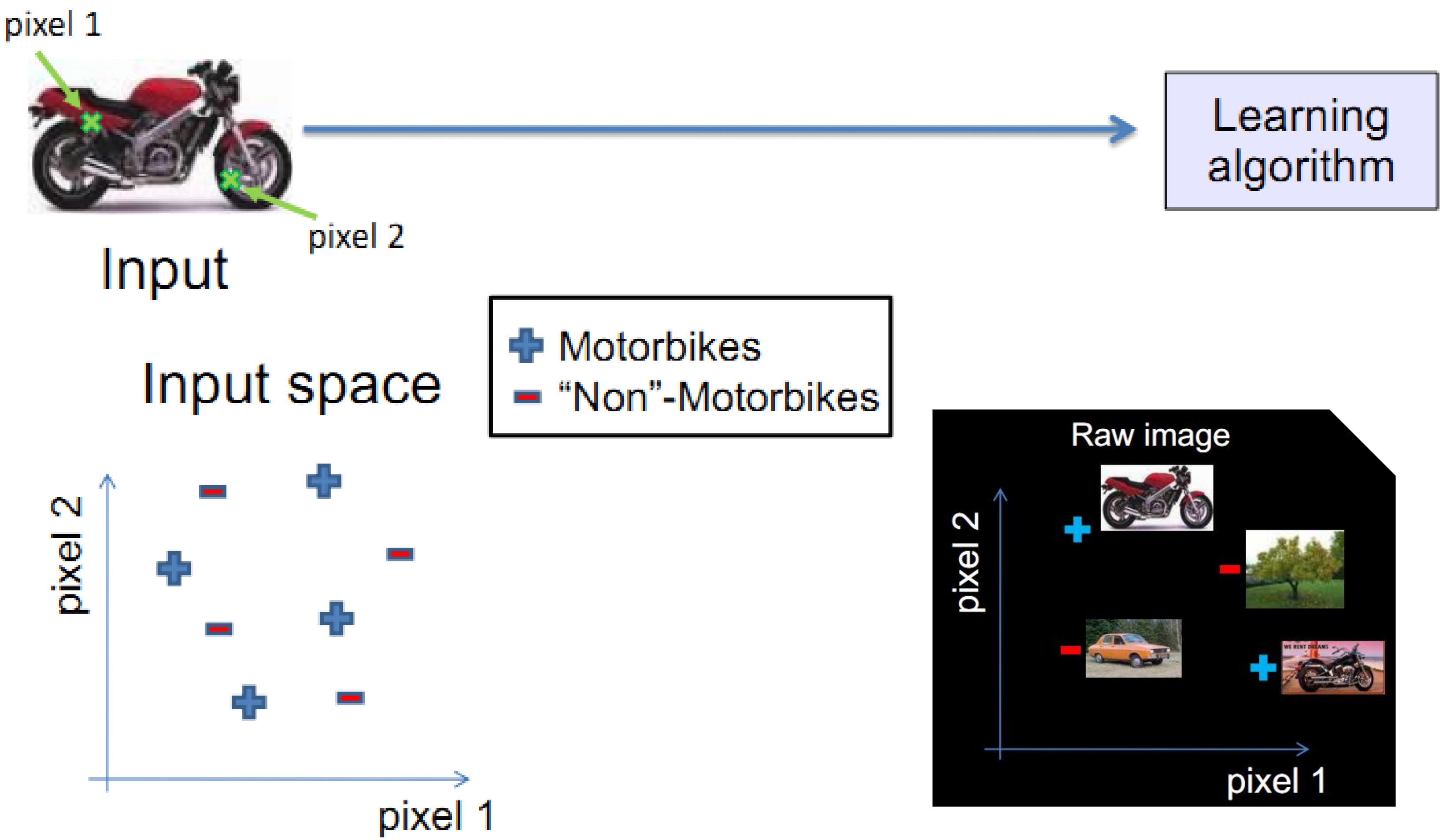
2013 Stanford

深度学习(Deep Learning)

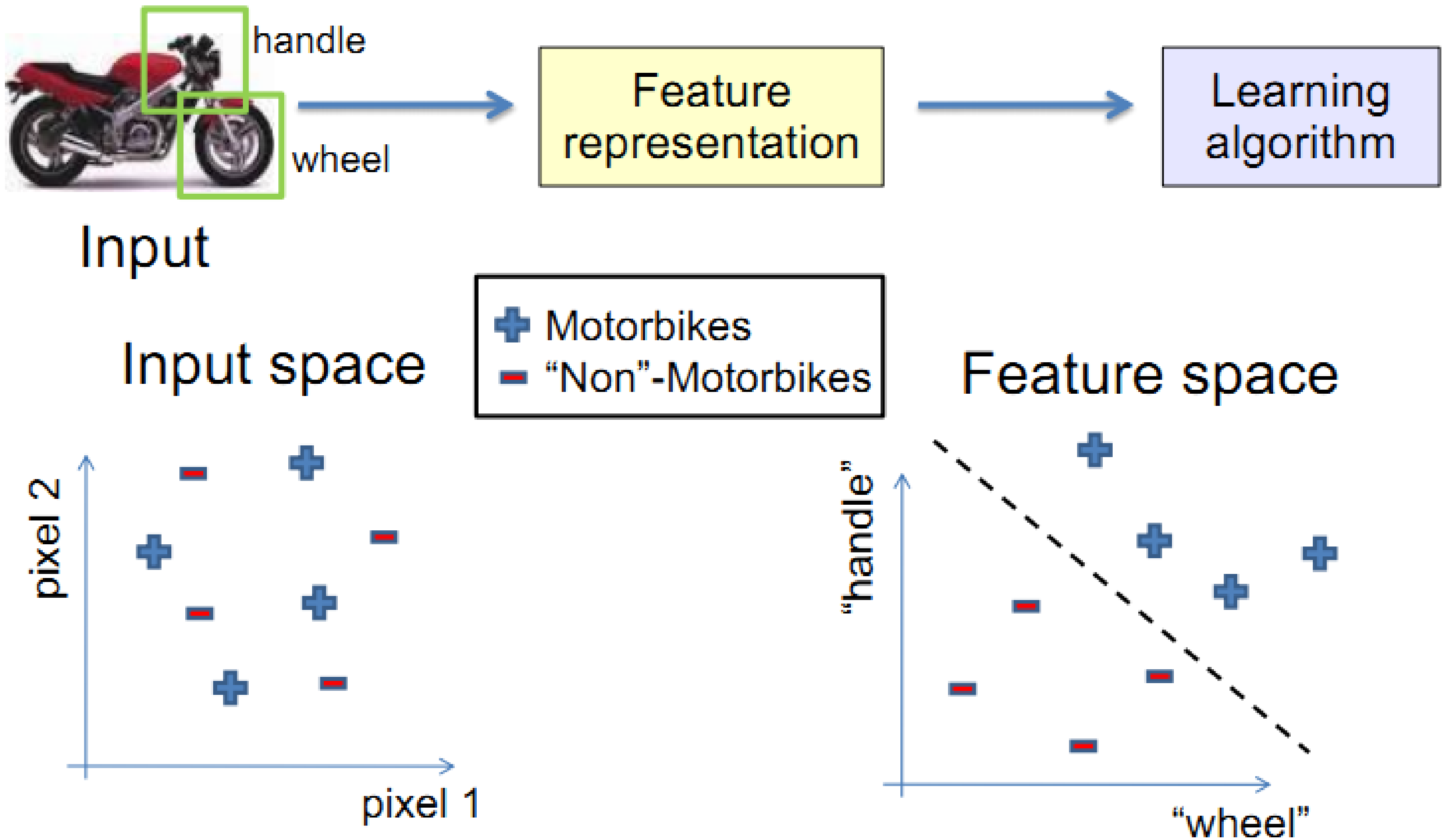
- 大神们
  - Geoff Hinton
  - Yoshua Bengio
  - Andrew Ng
  - Le Cun
  - Jeff Dean
  - Ruslan Salakhutdinov
  - Honglak Lee
  - ....



Feature representations



Feature representations



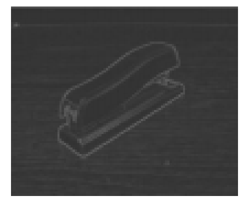
How is computer perception done?

State-of-the-art:
"hand-crafting"

Input data → Feature representation → Learning algorithm

Object detection

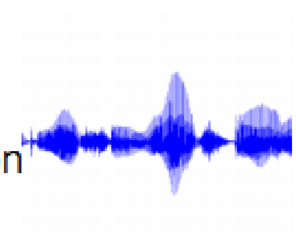
Image



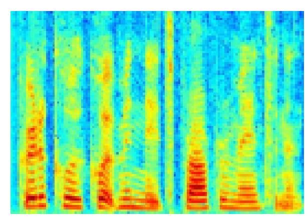
Low-level vision features (SIFT, HOG, etc.)

Object detection / classification

Audio classification



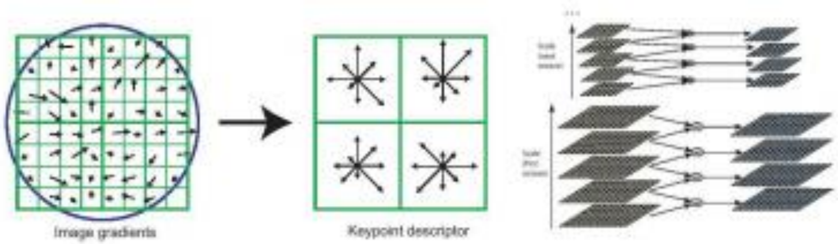
Audio



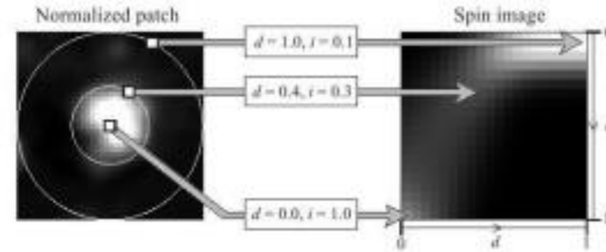
Low-level audio features (spectrogram, MFCC, etc.)

Speaker identification

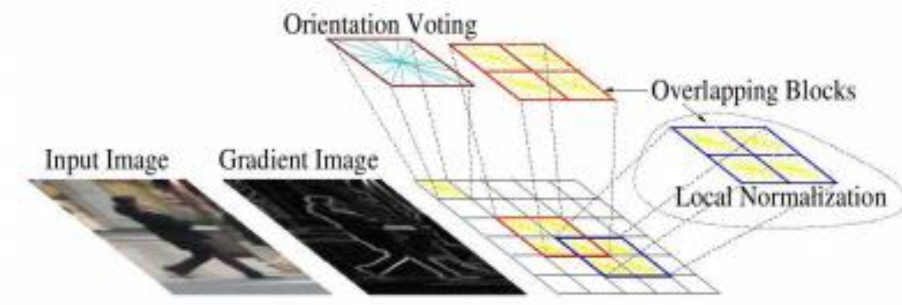
Computer vision features



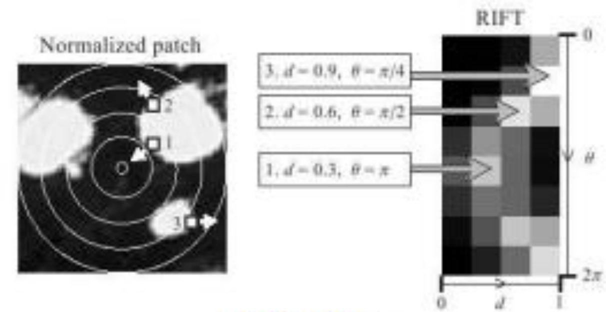
SIFT



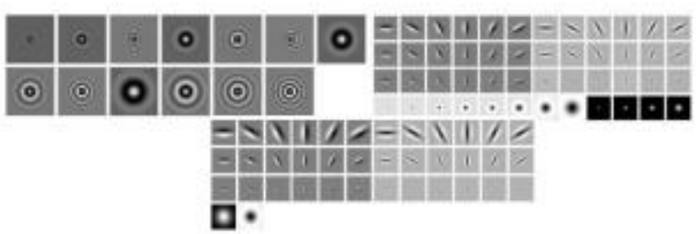
Spin image



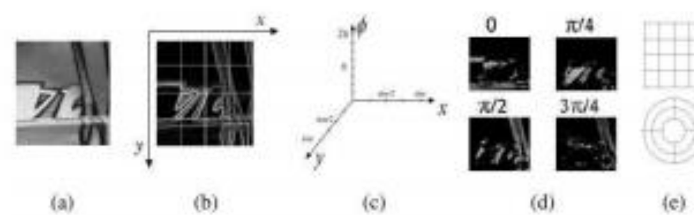
HoG



RIFT

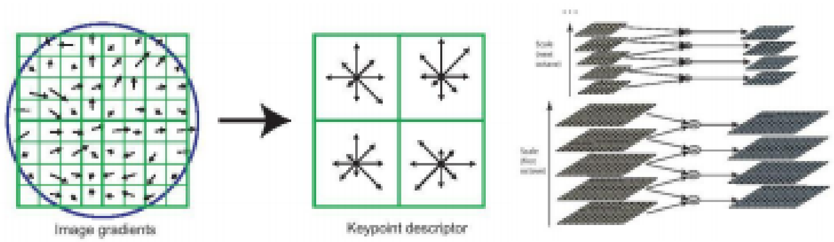


Textons

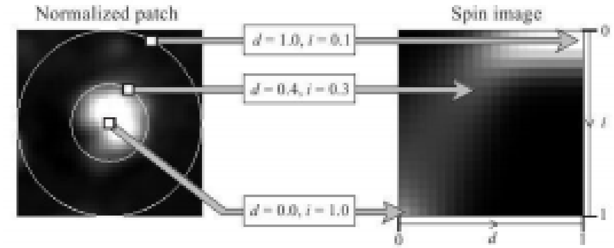


GLOH

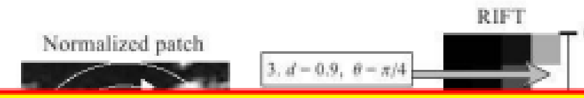
Computer vision features



SIFT



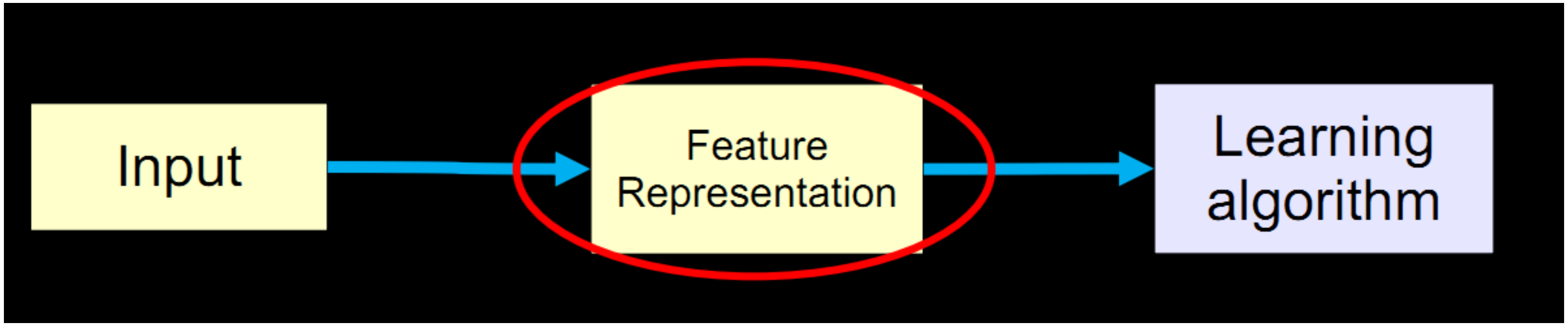
Spin image



Hand-crafted features:

1. Needs expert knowledge
2. Requires time-consuming hand-tuning
3. (Arguably) one of the limiting factors of computer vision systems

深度学习的目的



大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

深度学习

语言模型

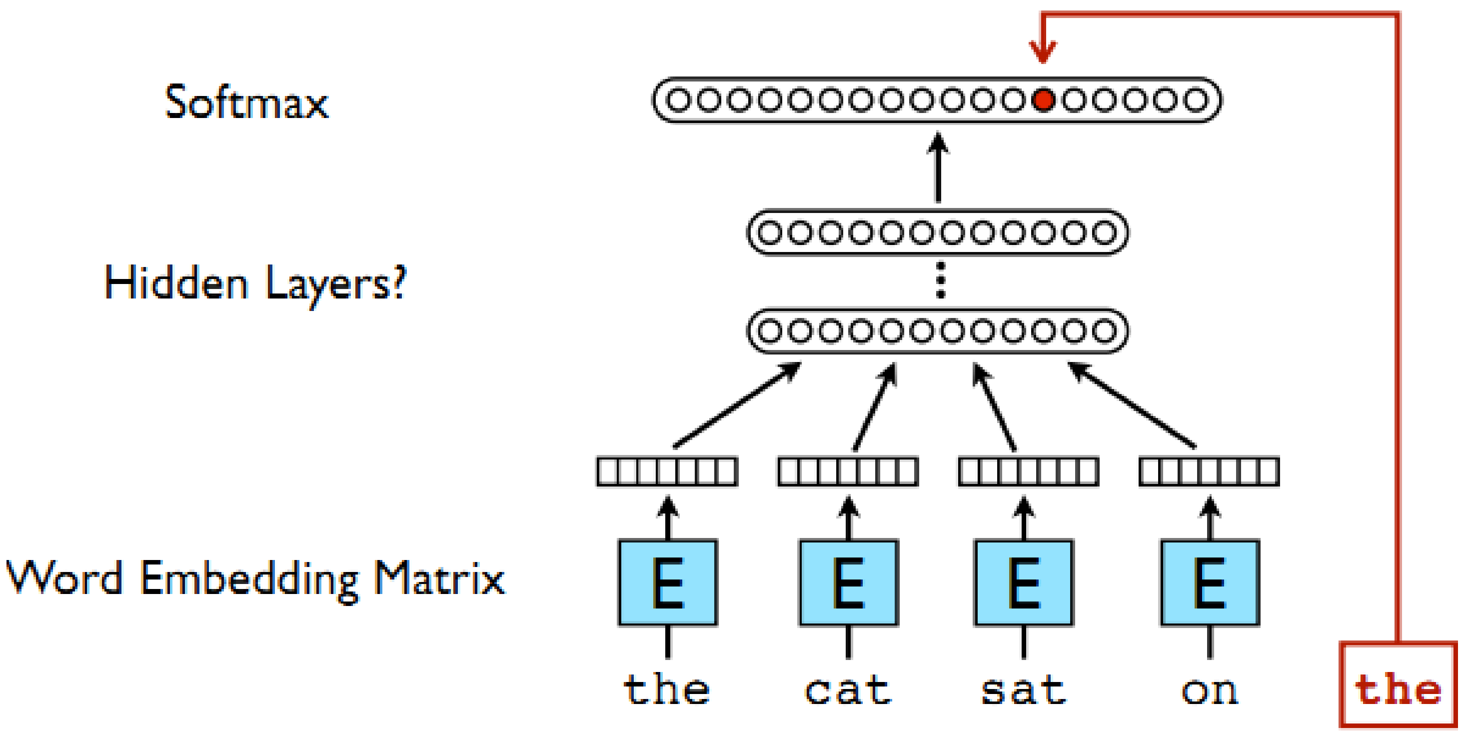
- 传统：Trigram 模型

$$p(w_i \mid w_{i-n} \ldots w_{i-1}) = \frac{C(w_{i-n} \ldots w_{i-1}, w_i)}{C(w_{i-n} \ldots w_{i-1})}$$

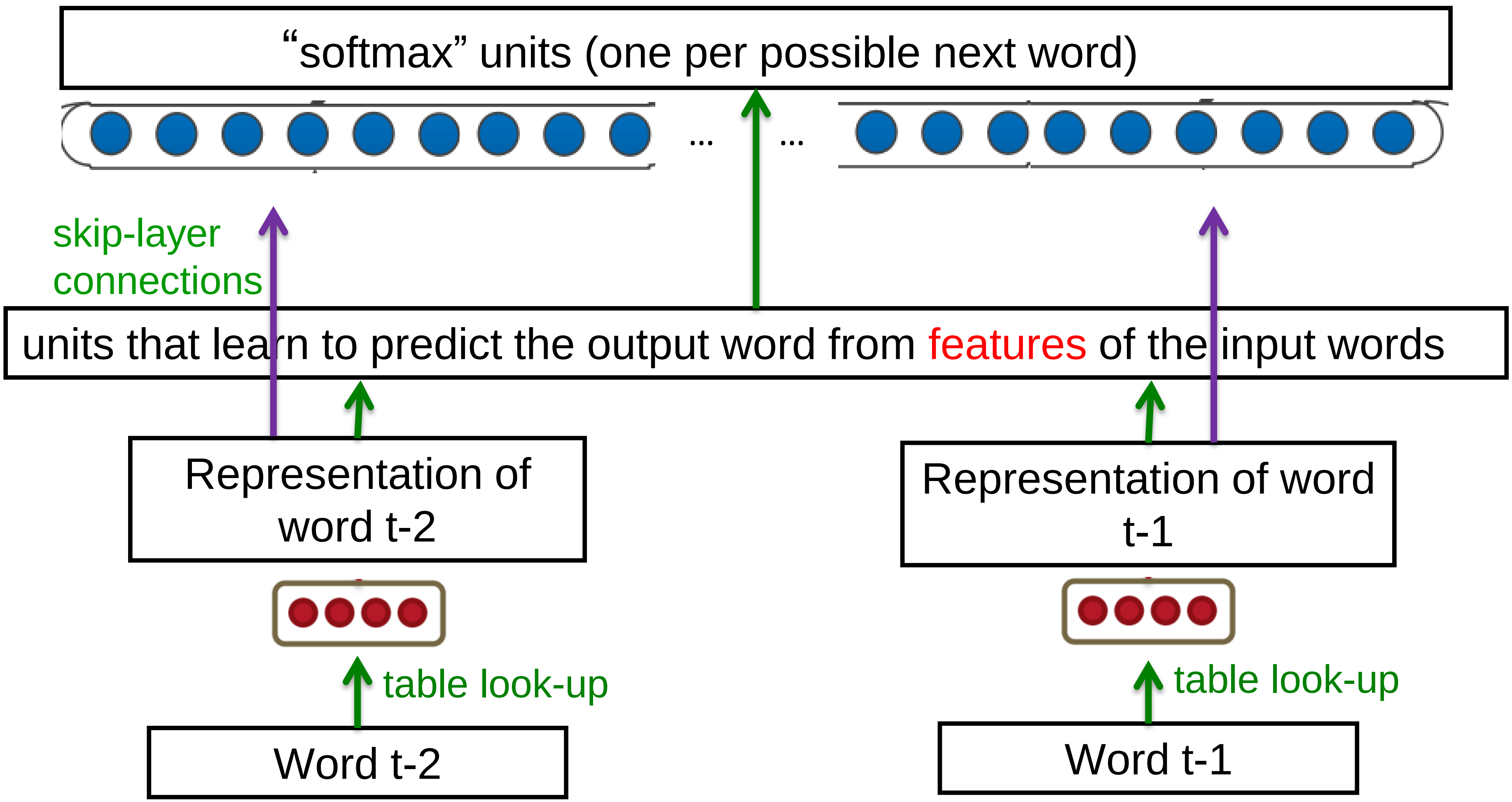
  – the cat sat in the garden on Friday
  – the dog sat in the yard on Monday

  – cat/dog garden/yard Friday/ Monday

Neuron Language Model

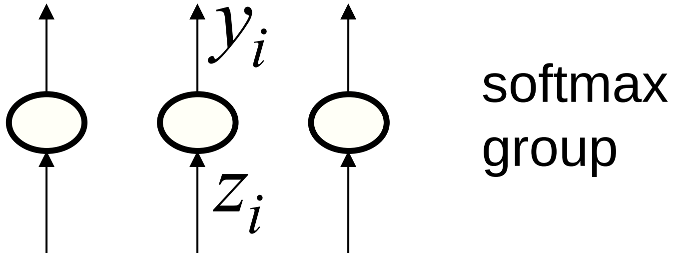


Bengio's Neuron Nets



SoftMax

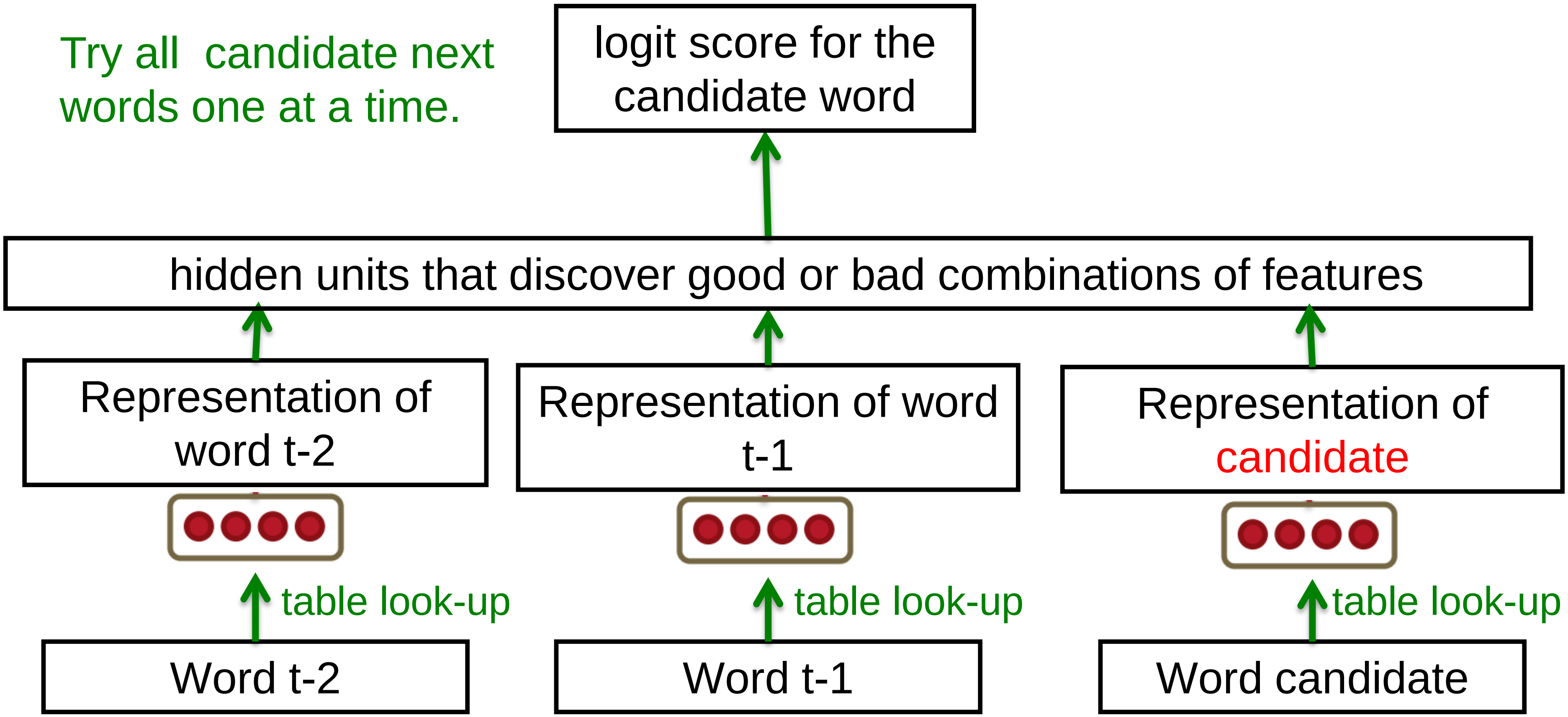
- 转换为一种概率表示



$$y_i = \frac{e^{z_i}}{\sum_{j \in group} e^{z_j}}$$

zi is called the “logit”

解决输出层节点太多


Try all candidate next words one at a time.
logit score for the candidate word
hidden units that discover good or bad combinations of features
Representation of word t-2
Representation of word t-1
Representation of candidate
table look-up
table look-up
table look-up
Word t-2
Word t-1
Word candidate

如何学习word embedding

传统的Word Representation

- One-hot
  - 向量空间中，1个1，很多0
  - 维度
    - 20K (speech)、50K (PTB)
    - 500K (big vocab)、13M (Google 1T)

  - 单词 motel 和 hotel

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Distributional similarity based representations

You can get a lot of value by representing a word by means of its neighbors

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

One of the most successful ideas of modern statistical NLP

government debt problems turning into	banking	crises as has happened in
saying that Europe needs unified	banking	regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

You can vary whether you use local or large context to get a more syntactic or semantic clustering

Neural word embeddings as a distributed representation

Similar idea

Combine vector space semantics with the prediction of probabilistic models (Bengio et al. 2003, Collobert & Weston 2008, Turian et al. 2010)

In all of these approaches, including deep learning models, a word is represented as a dense vector

$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Advantages of the neural word embedding approach

Compared to a method like LSA, neural word embeddings can become more meaningful through adding supervision from one or multiple tasks

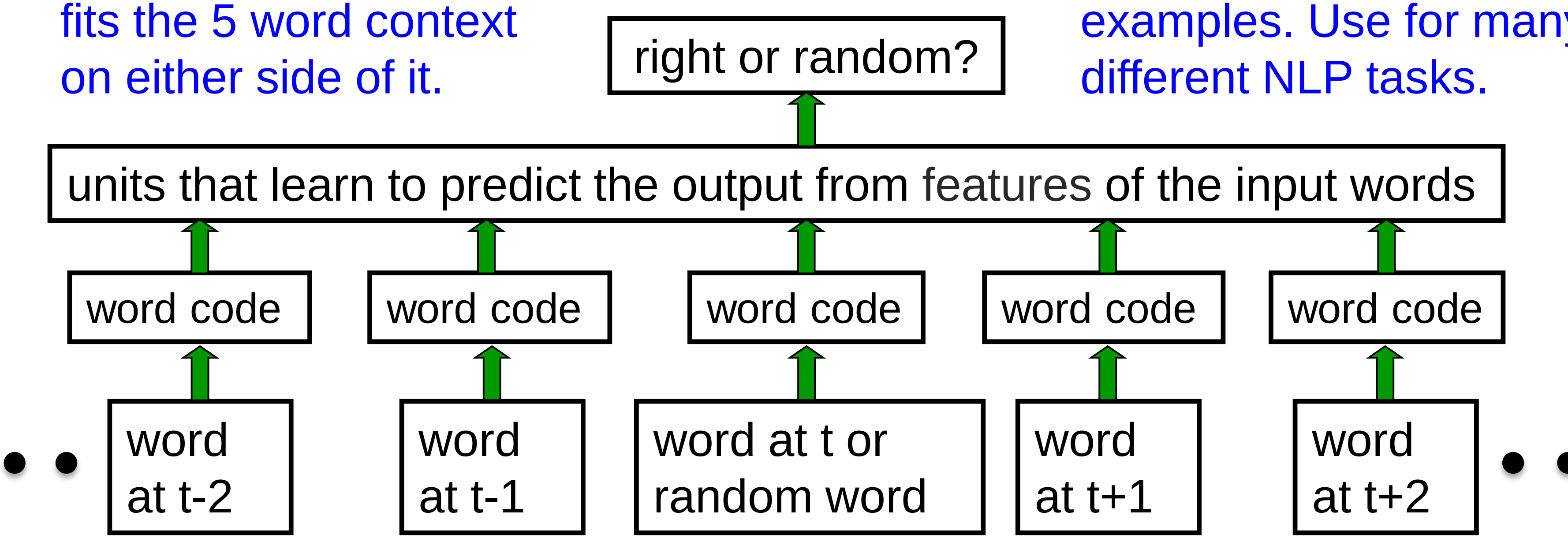
- For instance, sentiment is usually not captured in unsupervised word embeddings but can be in neural word vectors

如何学习每个单词的表示

- Collobert and Weston, 2008

Learn to judge if a word fits the 5 word context on either side of it.

Train on ~600 million examples. Use for many different NLP tasks.



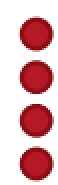
A neural network for learning word vectors

How do we formalize this idea? Ask that

score(cat chills on a mat) > score(cat chills Jeju a mat)

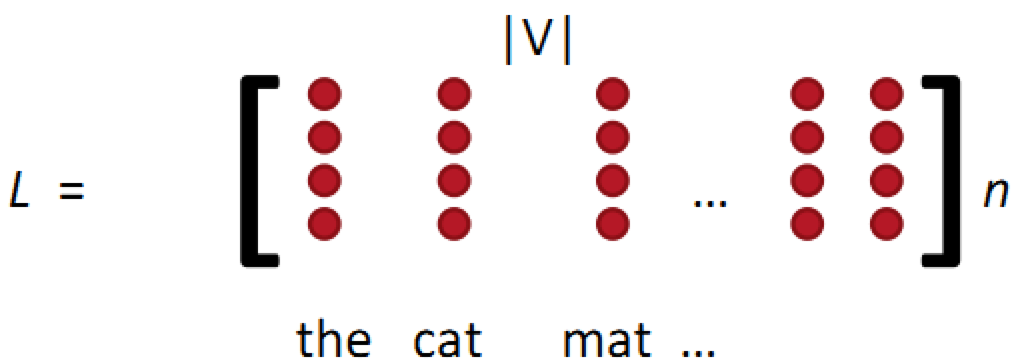
How do we compute the score?

- With a neural network
- Each word is associated with an *n*-dimensional vector



Word embedding matrix

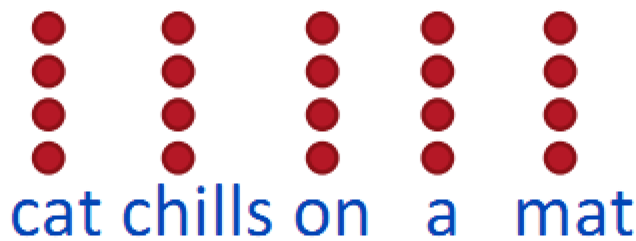
- Initialize all word vectors randomly to form a word embedding matrix $L \in \mathbb{R}^{n \times |V|}$

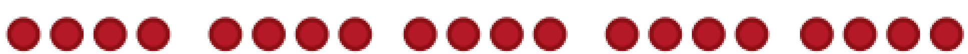


- These are the word features we want to learn
- Also called a look-up table
    - Conceptually you get a word's vector by left multiplying a one-hot vector *e* by *L*: $x = Le$

Word vectors as input to a neural network

- score(cat chills on a mat)
- To describe a phrase, retrieve (via index) the corresponding vectors from *L*



- Then concatenate them to 5*n* vector:
- x =[]
- How do we then compute score(x)?

A Single Layer Neural Network

- A single layer is a combination of a linear layer and a nonlinearity:

$$\begin{aligned} z &= Wx + b \\ a &= f(z) \end{aligned}$$

- The neural activations can then be used to compute some function.
- For instance, the score we care about:

$$score(x) = U^T a \in \mathbb{R}$$

大纲

- 背景介绍
- 神经元
- 神经网络
- 权重学习
- 深度学习概述
- 语言模型
- DBN & RNN

神经网络

深度学习

Deep Belief Network

思想

- Reconstruction
- RBM
  - Energy Model
  - 和Auto-encoder可互换

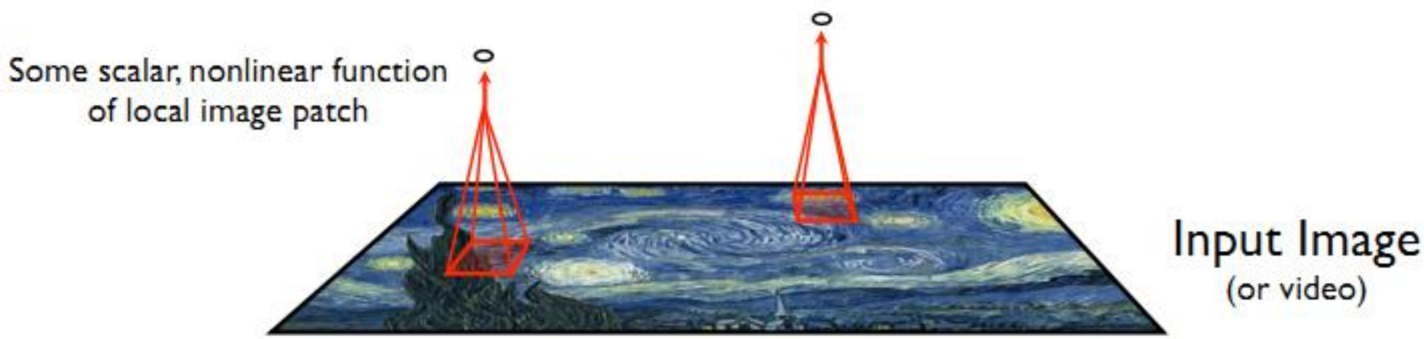
Reconstruction

Deep Networks

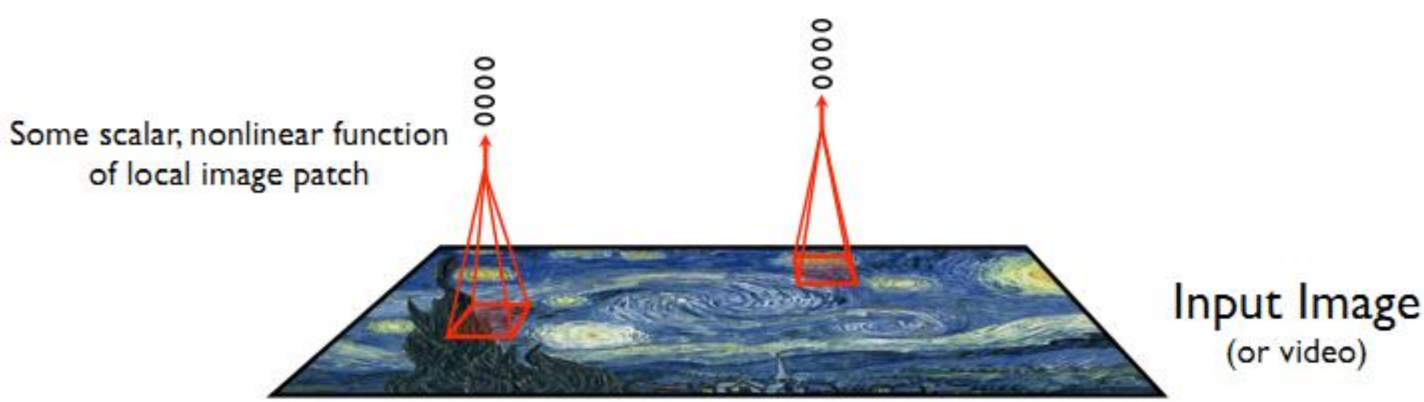
Input Image
(or video)

Deep Networks

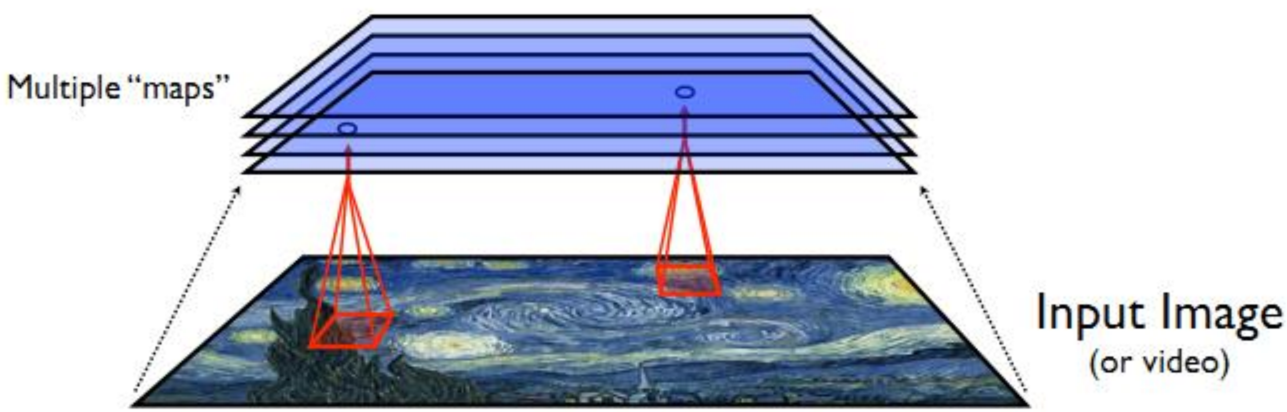
Deep Networks



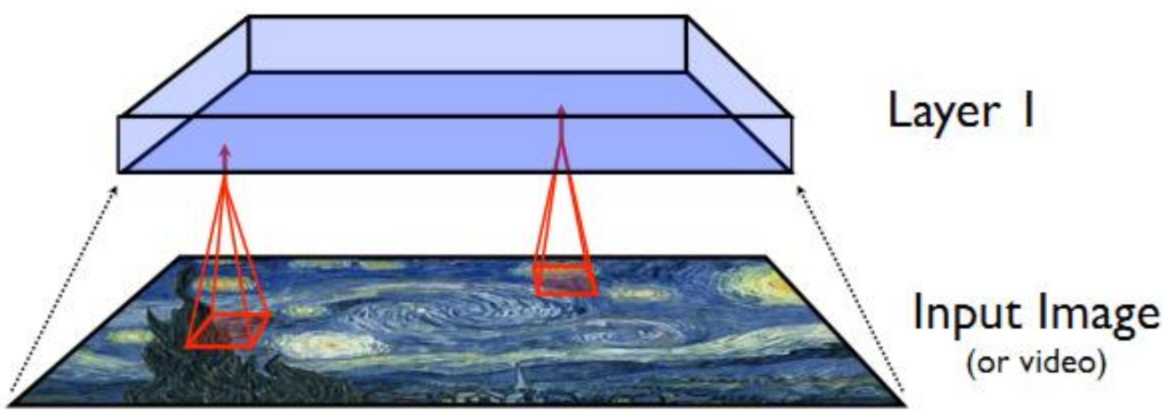
Deep Networks



Deep Networks

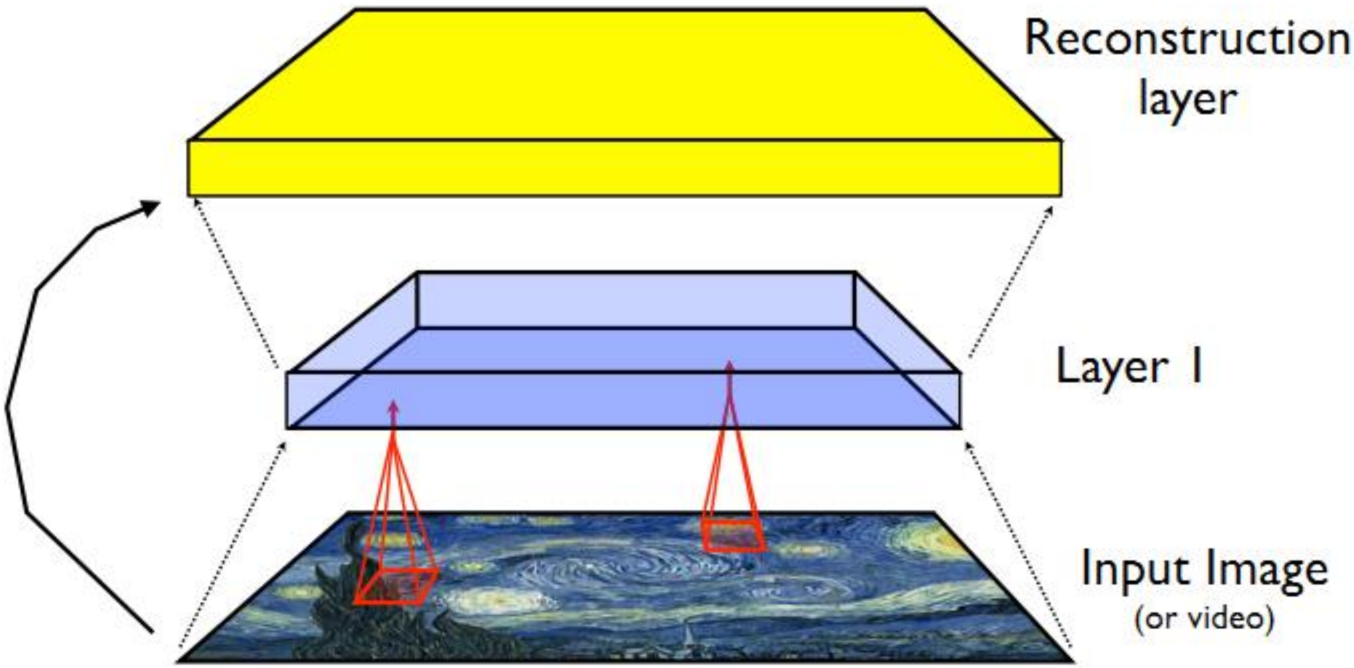


Deep Networks

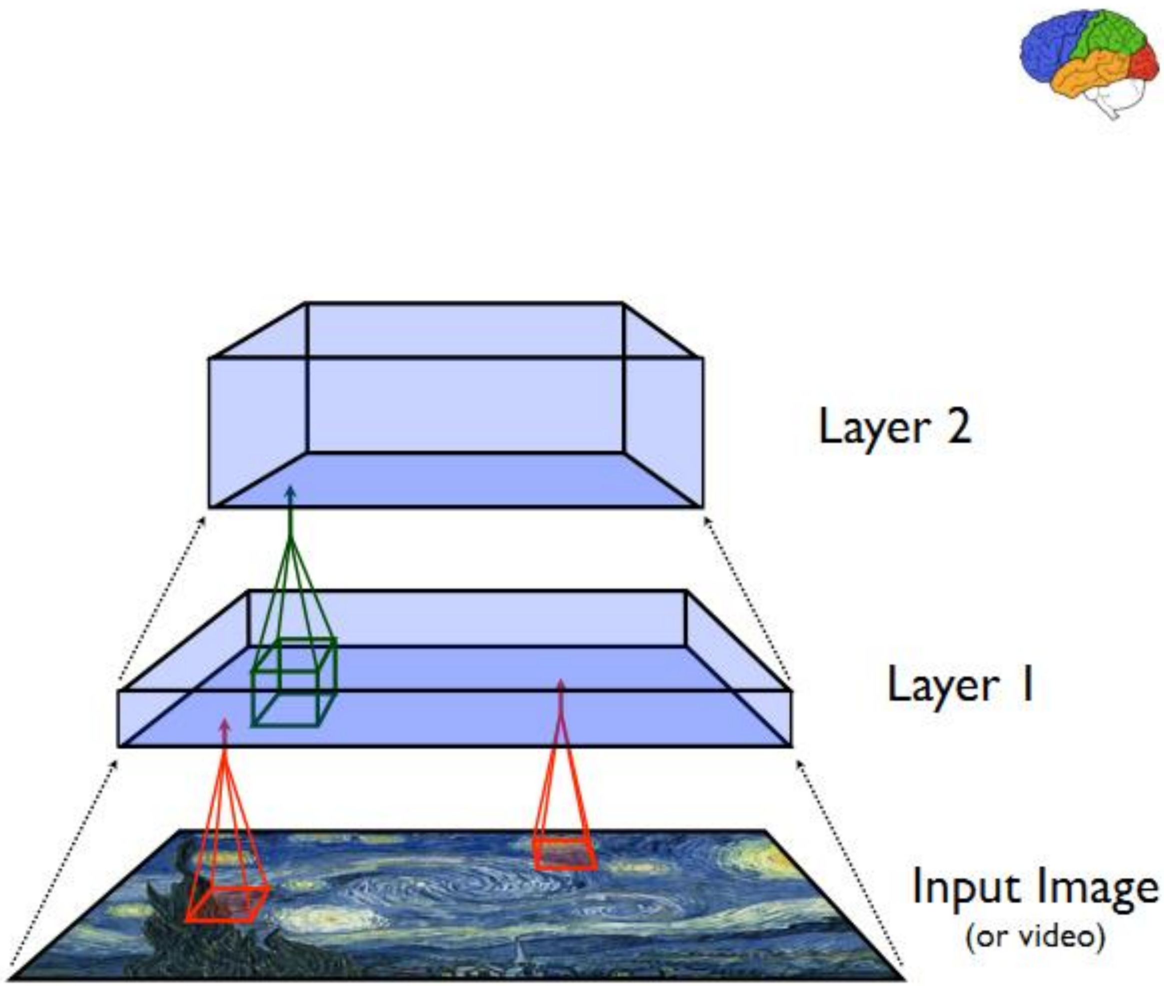


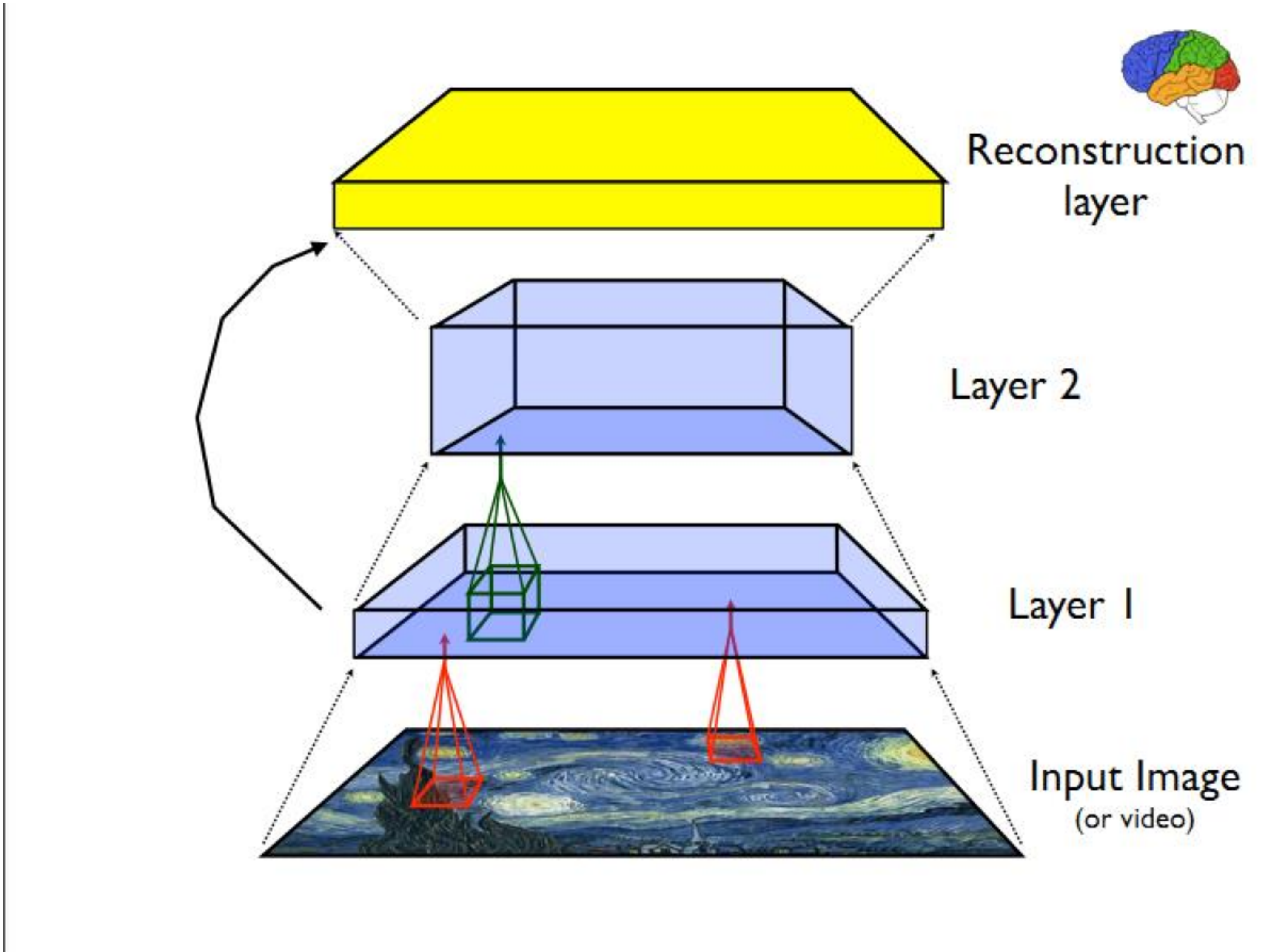
Unsupervised Training

Core idea: try to reconstruct input from just the learned representation



Due to Geoff Hinton, Yoshua Bengio, Andrew Ng, and others

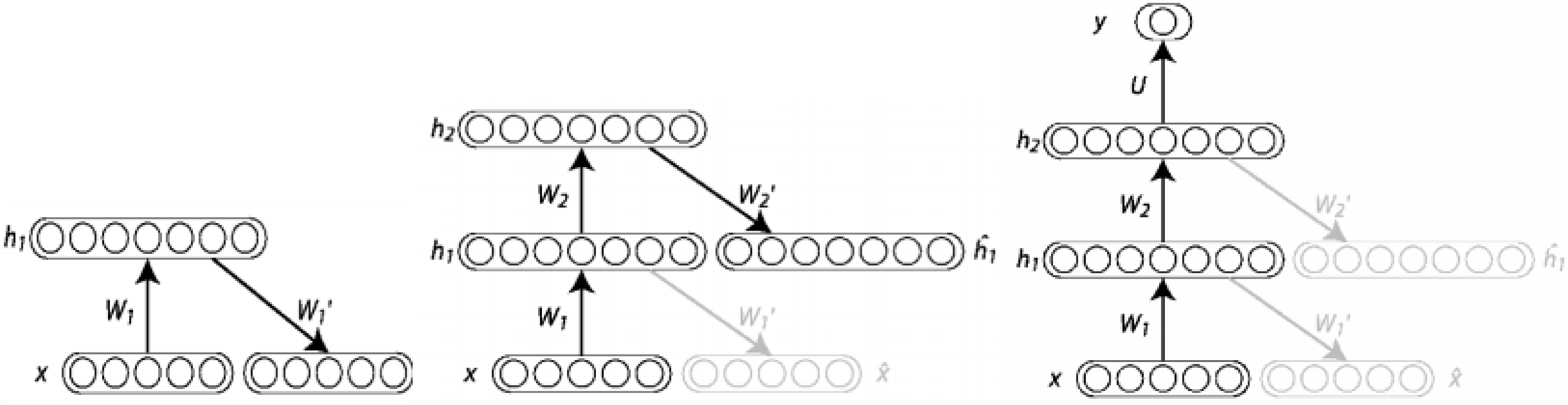

Layer 2
Layer 1
Input Image
(or video)


Reconstruction
layer
Layer 2
Layer 1
Input Image
(or video)

Auto-Encoder

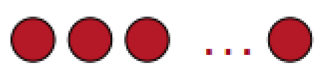
Stacking Auto-Encoders

- Can be stacked successfully (Bengio et al NIPS'2006) to form highly non-linear representations

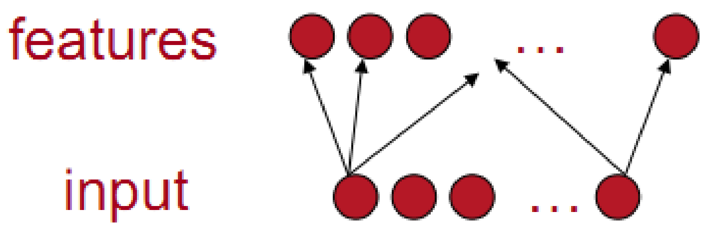


Layer-wise Unsupervised Learning

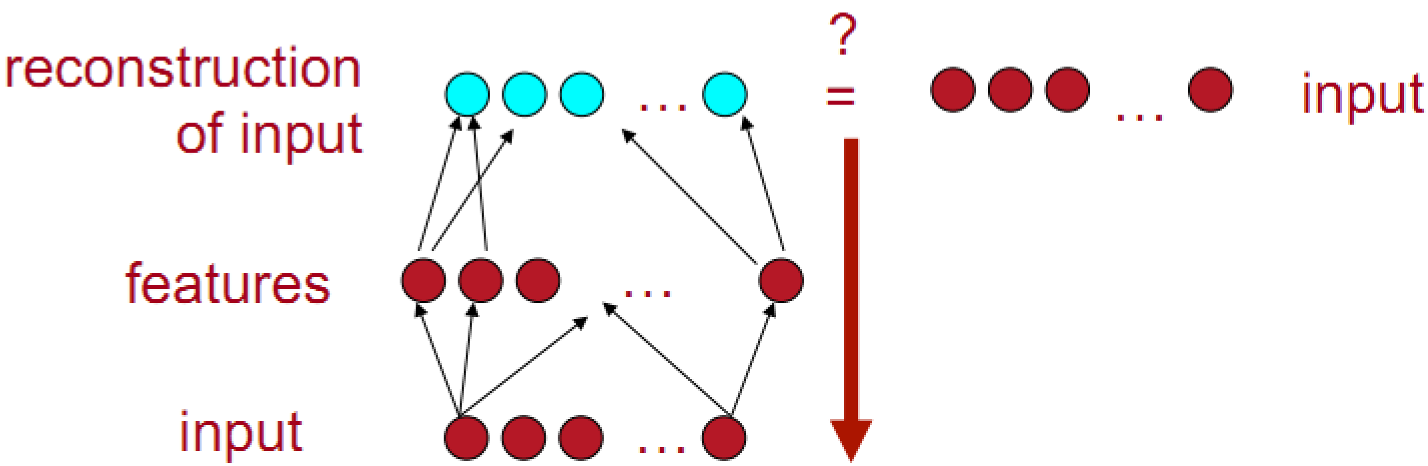
input



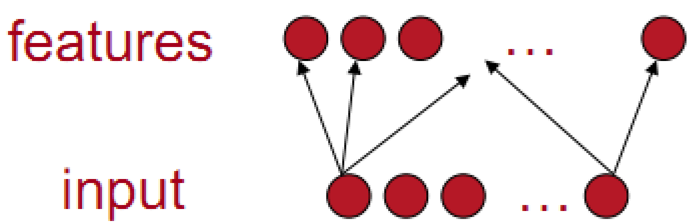
Layer-wise Unsupervised Pre-training



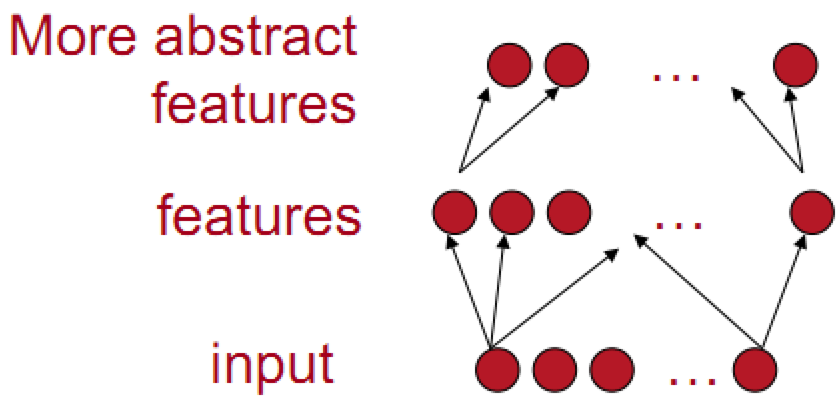
Layer-wise Unsupervised Pre-training



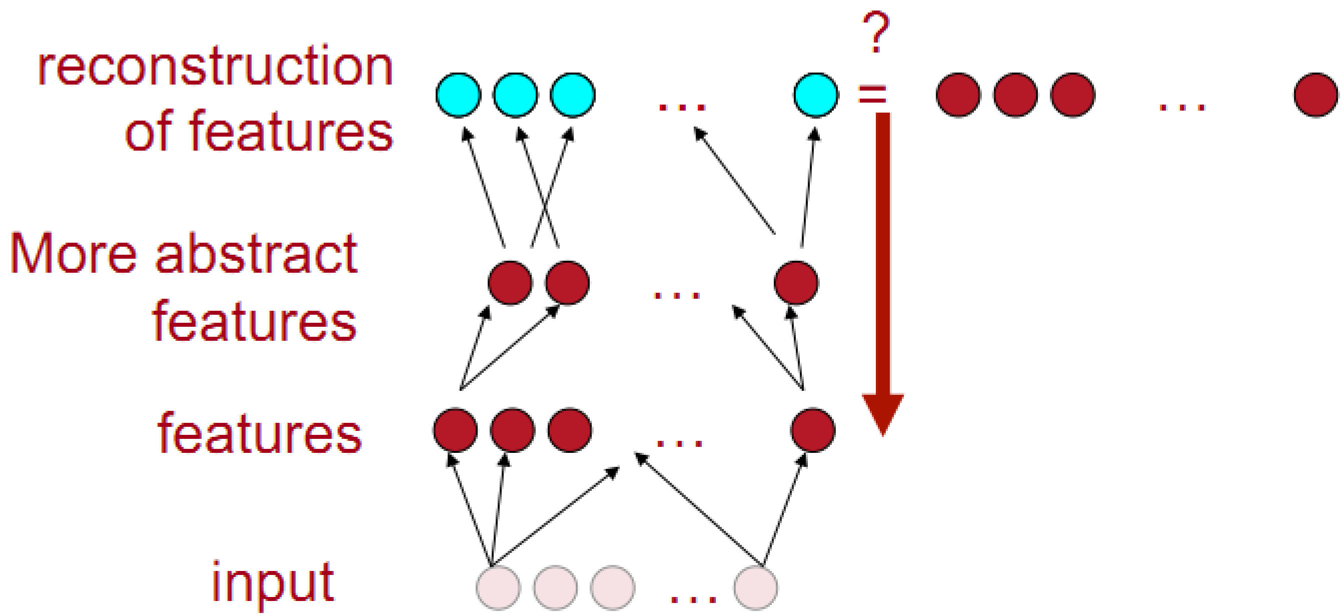
Layer-wise Unsupervised Pre-training



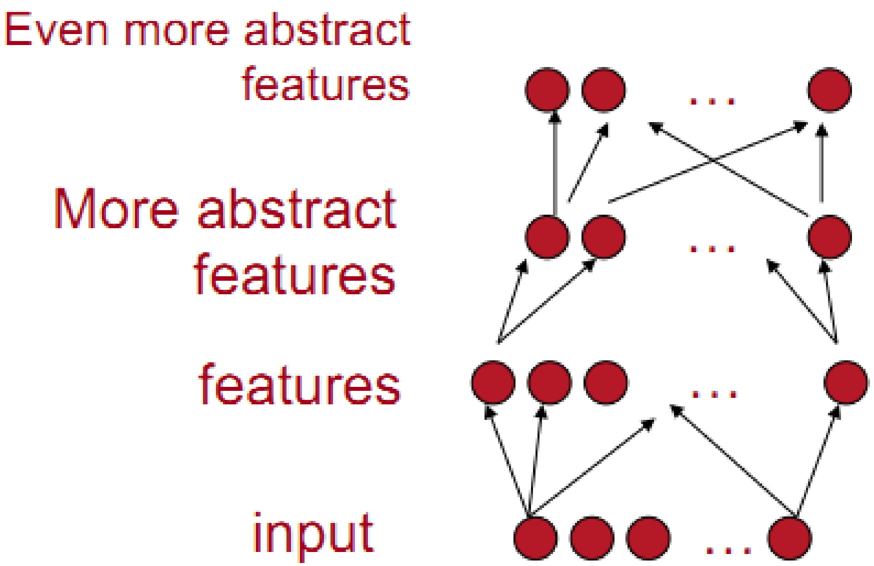
Layer-wise Unsupervised Pre-training



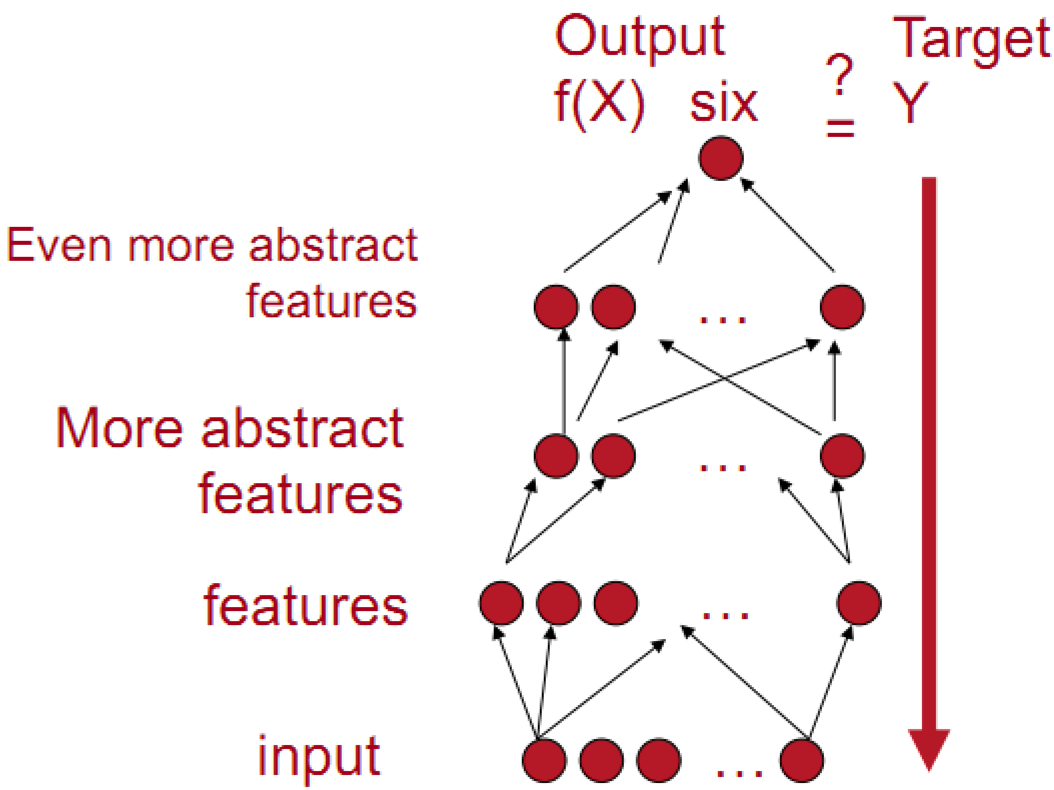
Layer-wise Unsupervised Learning



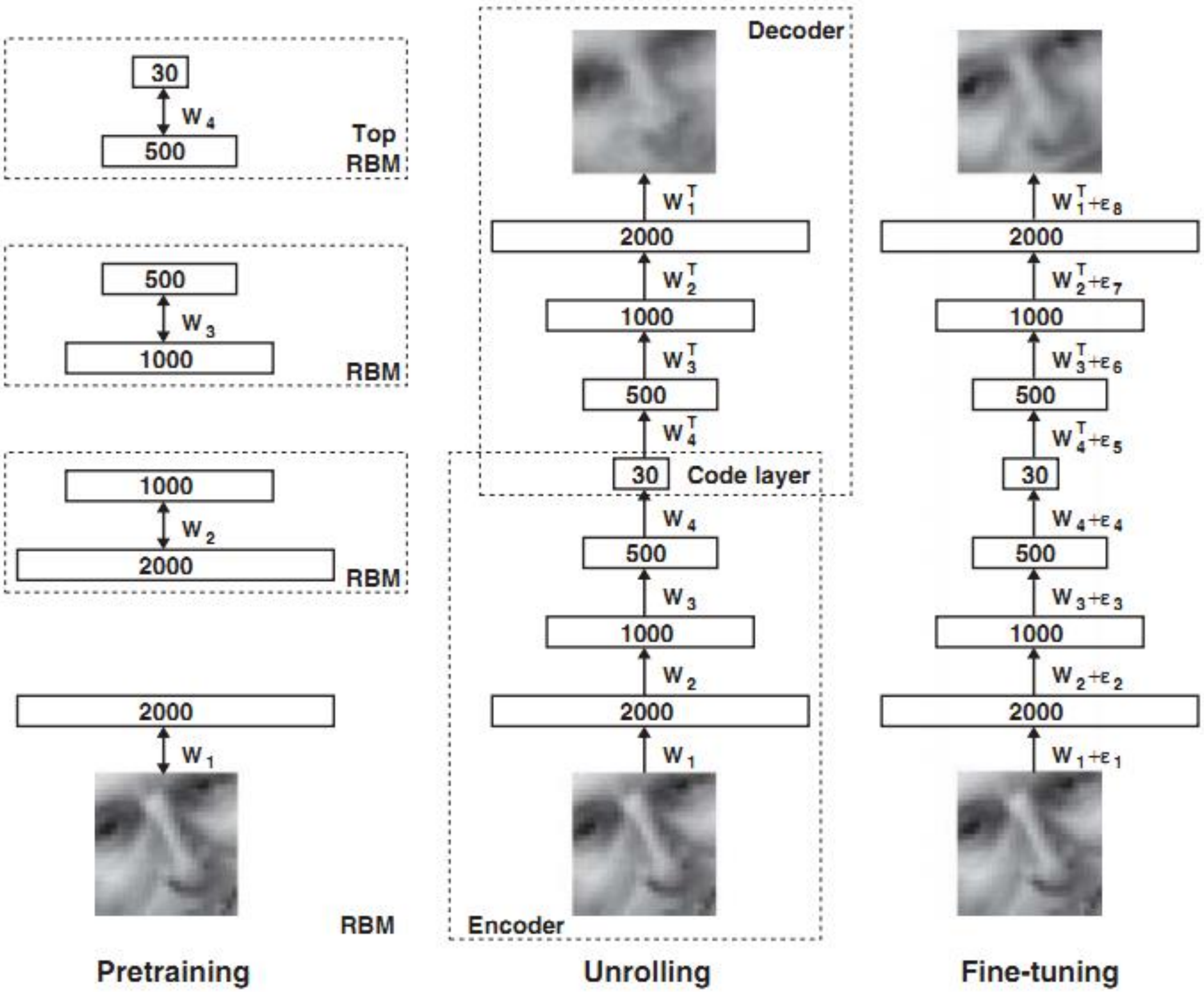
Layer-wise Unsupervised Learning



Supervised Fine-Tuning



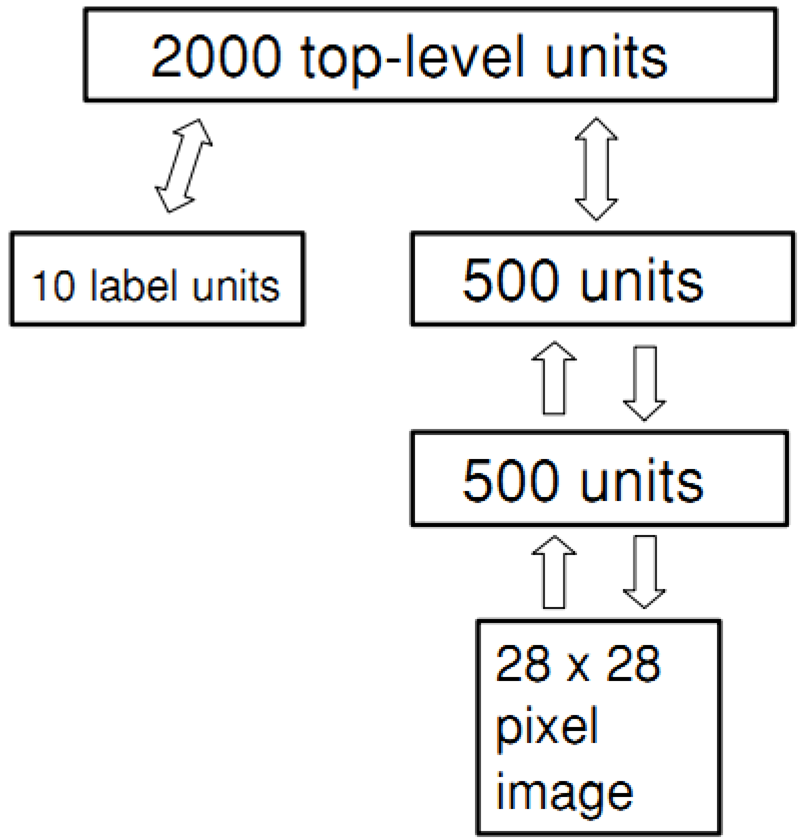
DBN模型

- 

30
W_4
500
Top RBM
500
W_3
1000
RBM
1000
W_2
2000
RBM
2000
W_1
RBM
Pretraining
Decoder
W_1^T
2000
W_2^T
1000
W_3^T
500
W_4^T
30
Code layer
W_4
500
W_3
1000
W_2
2000
W_1
Encoder
Unrolling
$W_1^T+\varepsilon_8$
2000
$W_2^T+\varepsilon_7$
1000
$W_3^T+\varepsilon_6$
500
$W_4^T+\varepsilon_5$
30
$W_4+\varepsilon_4$
500
$W_3+\varepsilon_3$
1000
$W_2+\varepsilon_2$
2000
$W_1+\varepsilon_1$
Fine-tuning

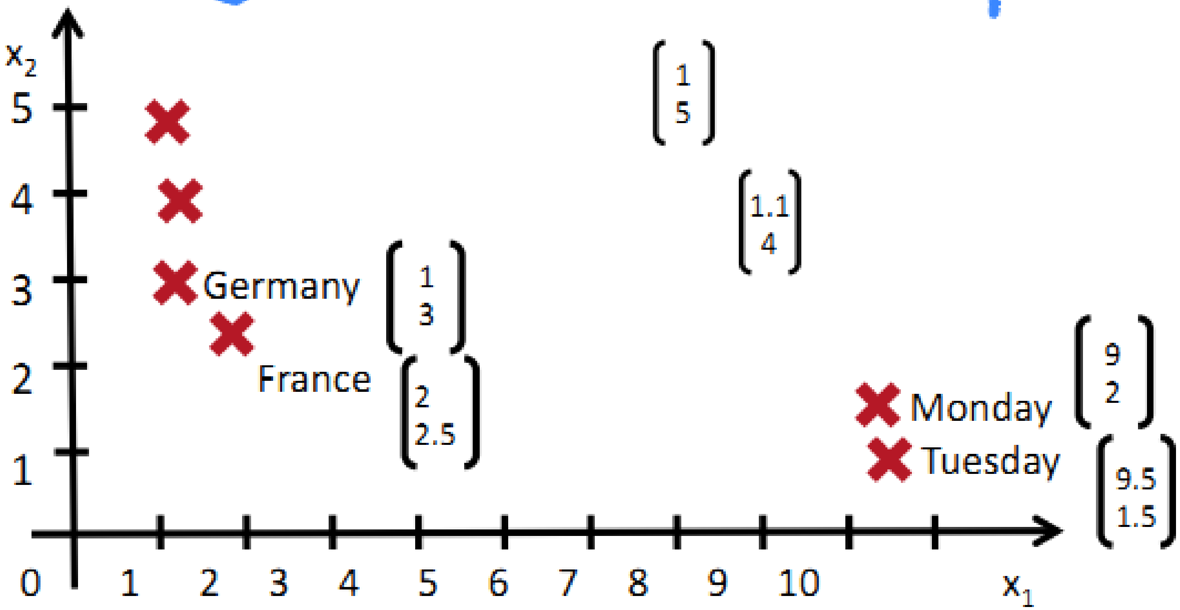
DBN模型

- Hinton 06 NC



Recursive Neuron Network

Building on Word Vector Space Models



the country of my birth

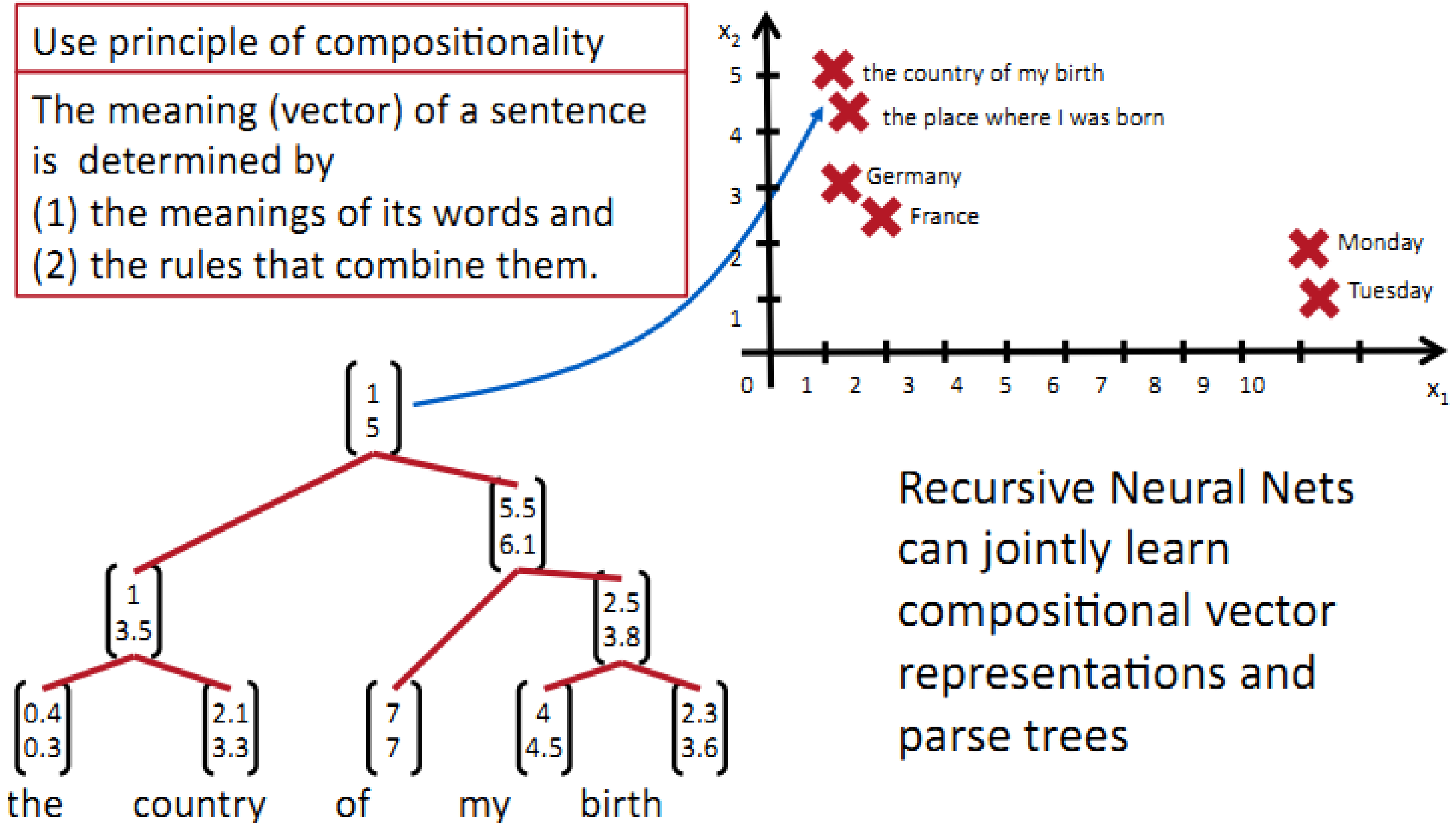
the place where I was born

But how can we represent the meaning of longer phrases?

By mapping them into the same vector space!

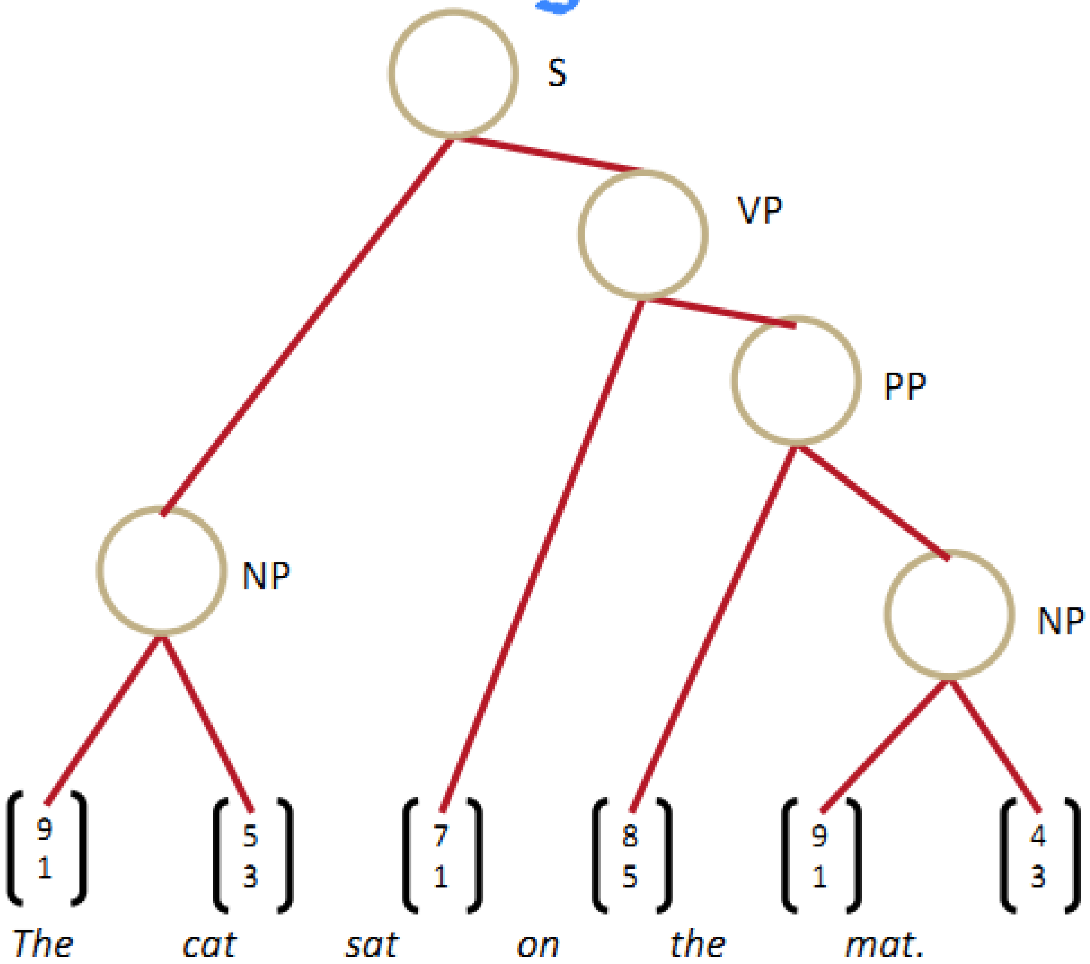
How should we map phrases into a vector space?

Use principle of compositionality
The meaning (vector) of a sentence is determined by (1) the meanings of its words and (2) the rules that combine them.

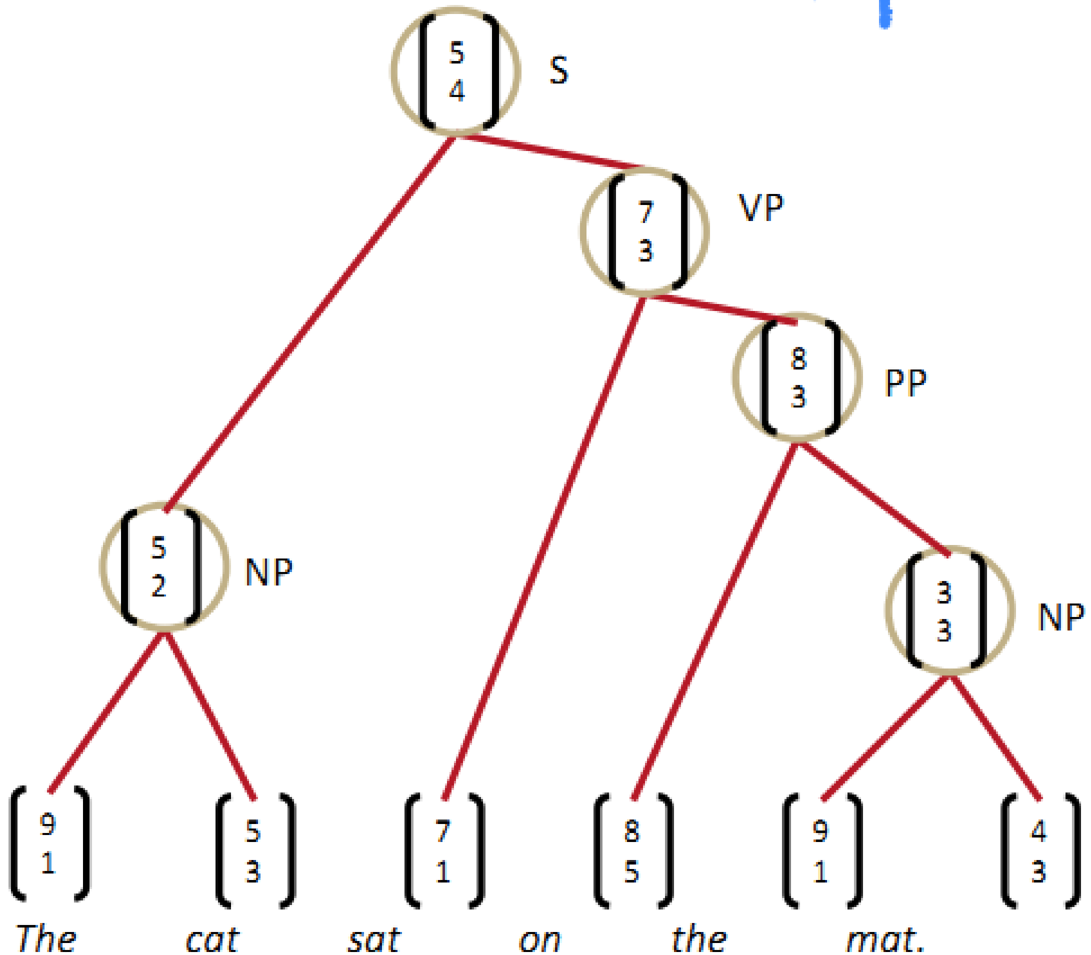


Recursive Neural Nets can jointly learn compositional vector representations and parse trees

Sentence Parsing: What we want



Learn Structure and Representation

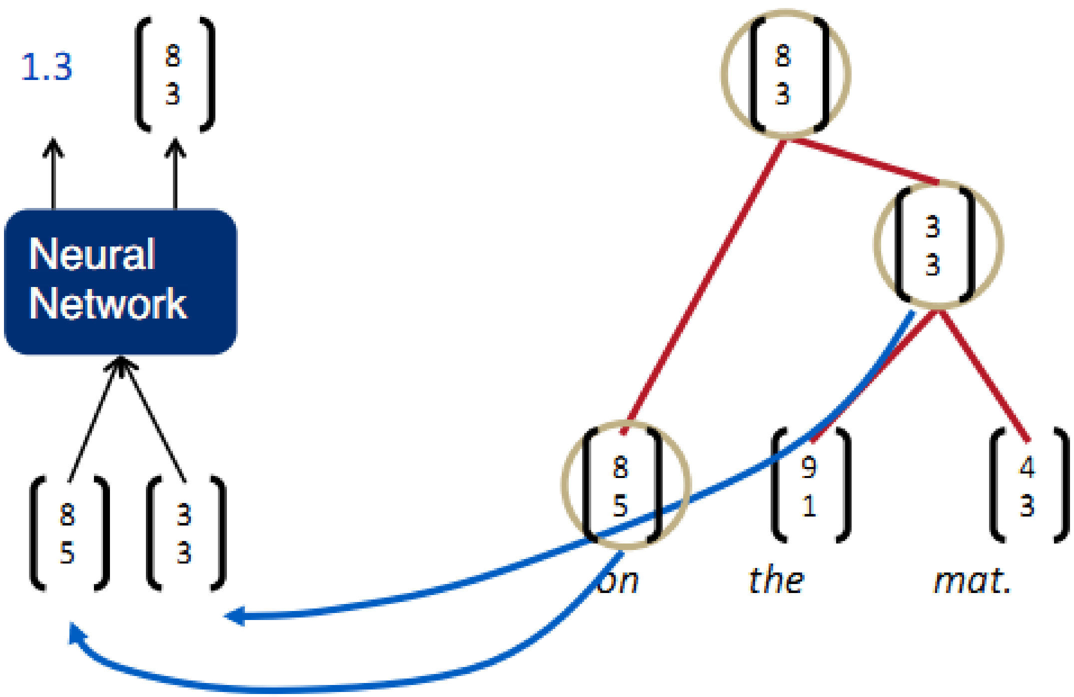


Recursive Neural Networks for Structure Prediction

Inputs: two candidate children's representations

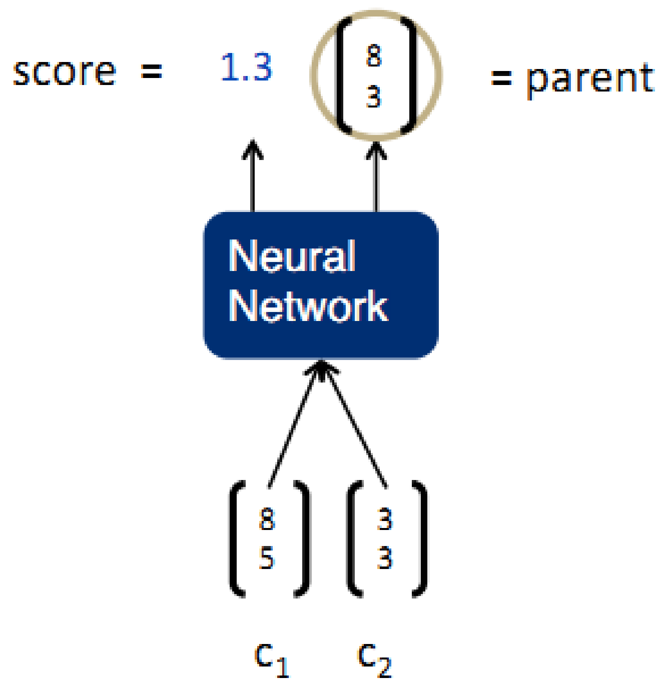
Outputs:

1. The semantic representation if the two nodes are merged.
2. Score of how plausible the new node would be.



103

Recursive Neural Network Definition

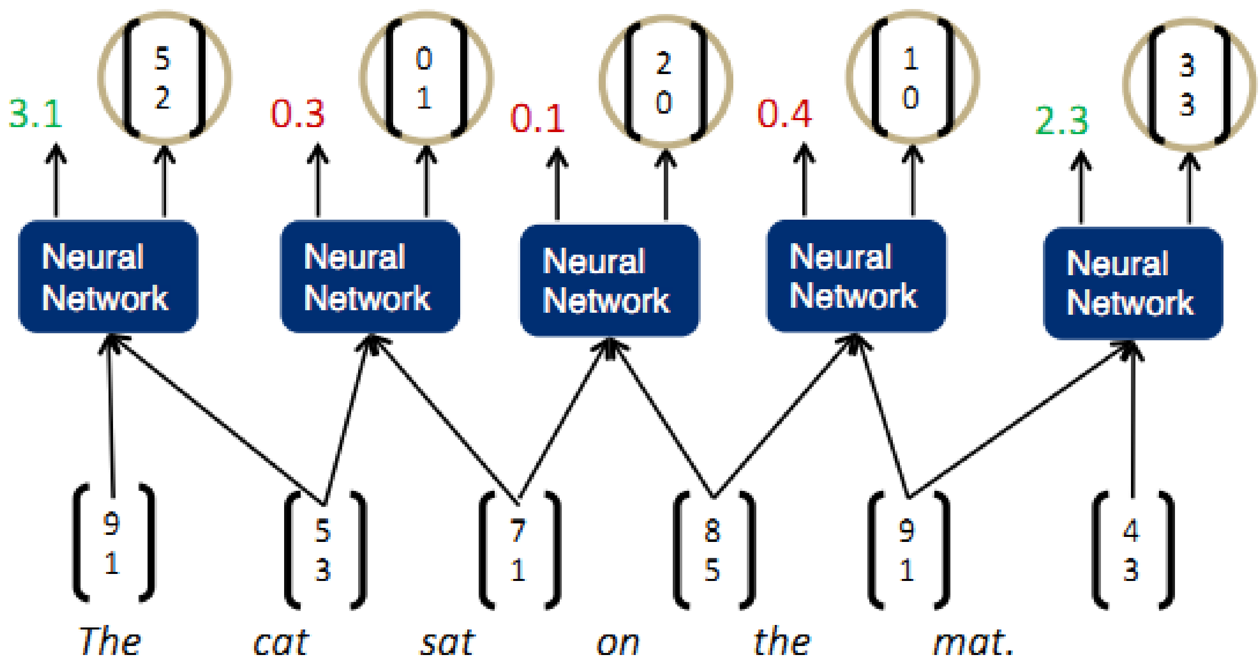


$$\text{score} = U^T p$$

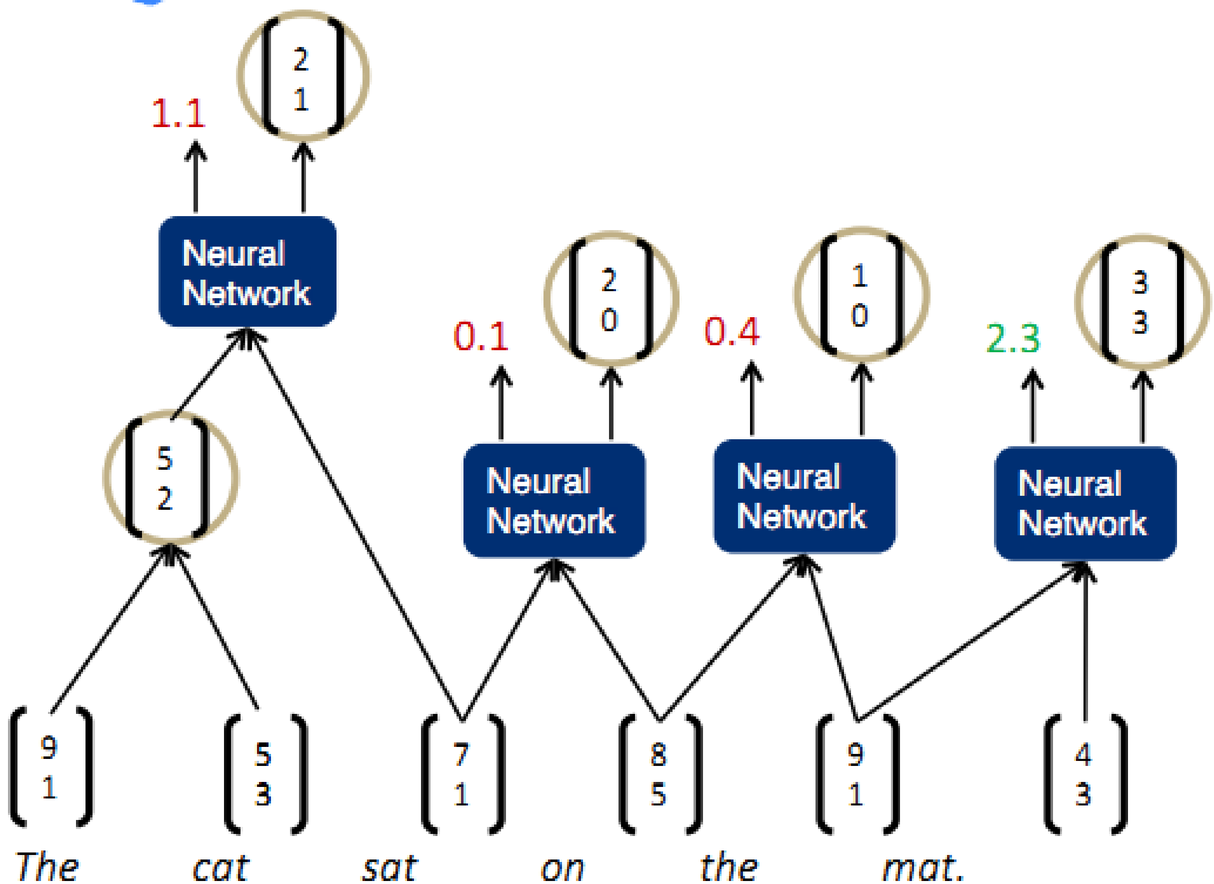
$$\boxed{p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),}$$

Same W parameters at all nodes of the tree

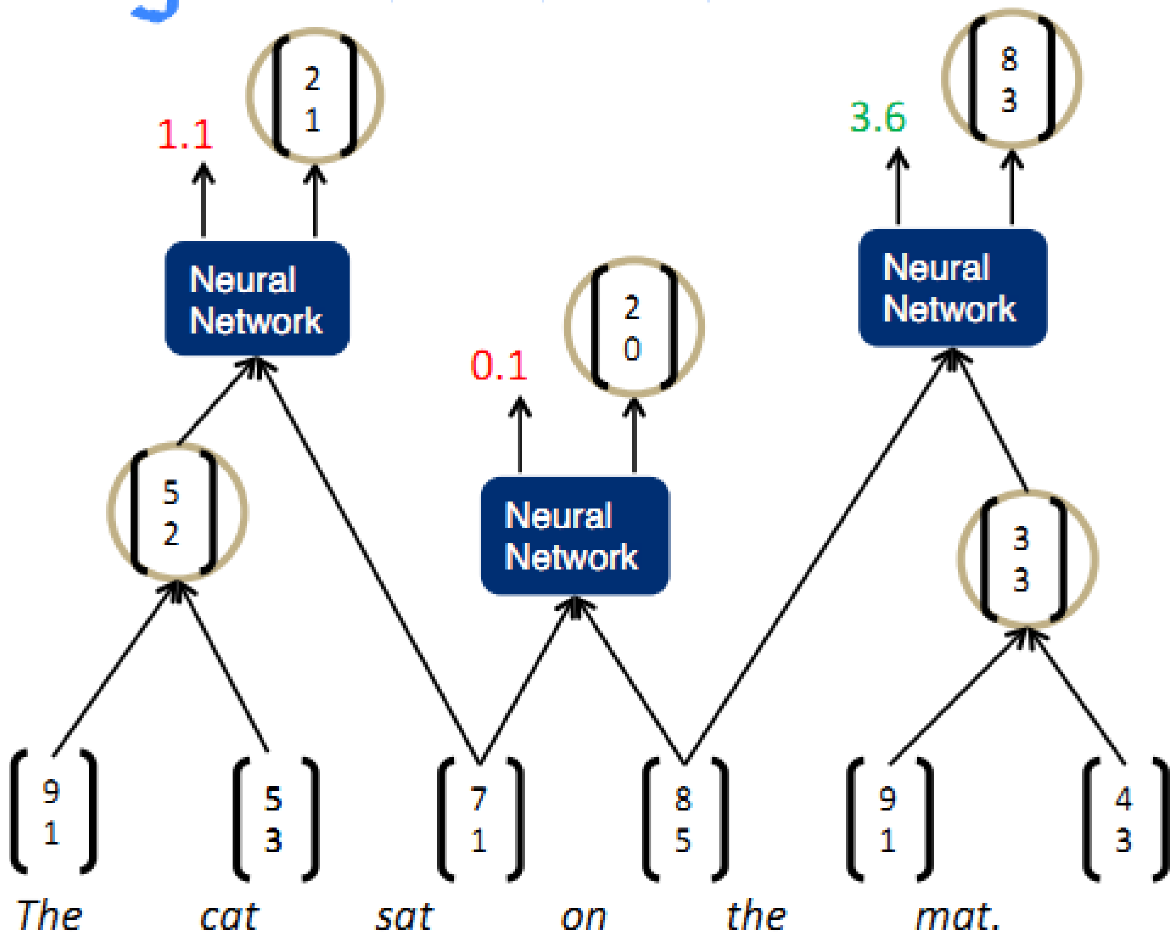
Parsing a sentence with an RNN



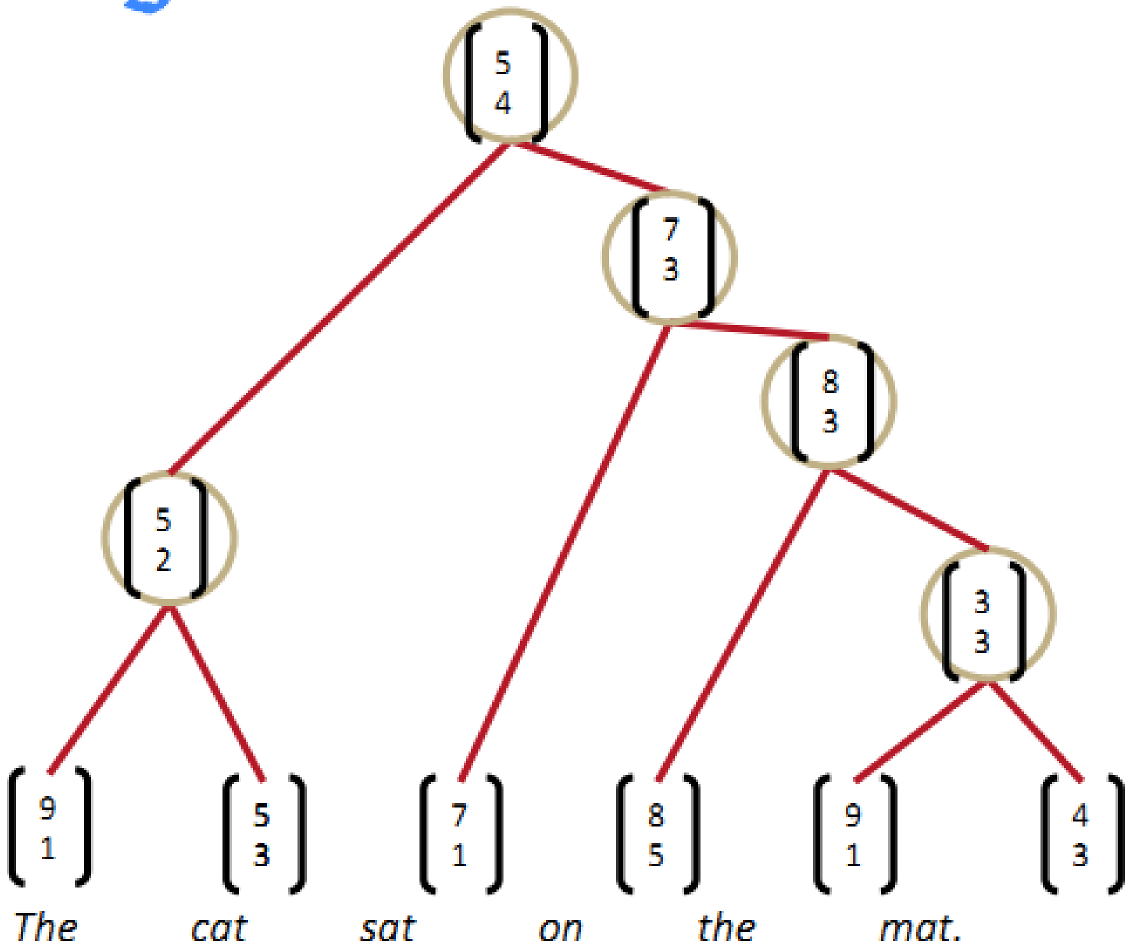
Parsing a sentence



Parsing a sentence



Parsing a sentence



Max-Margin Framework - Details

- $$\boxed{s(x_i, y_i) = \sum_{d \in T(y_i)} s_d(c_1, c_2)}$$

- Similar to max-margin parsing (Taskar et al. 2004), a supervised max-margin objective

Maximize J

$$J = \sum_i s(x_i, y_i) - \max_{y \in A(x_i)} \left(s(x_i, y) + \Delta(y, y_i) \right)$$

- Th $$\boxed{\Delta(y, y_i) = \sum_{d \in T(y)} \lambda \mathbf{1}\{d \notin T(y_i)\}}$$ ns

更多内容

http://deeplearning.net/

End