深度学习技术与应用（8）

Deep Learning: Techniques and Applications (8)

Ge Li

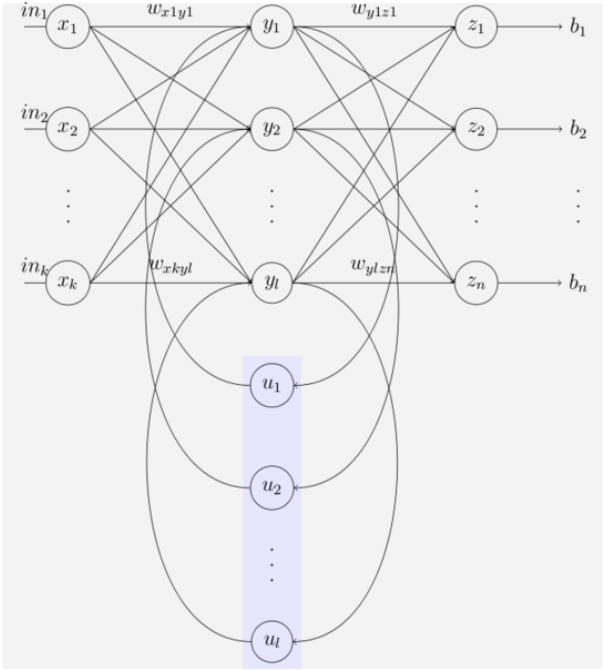
Peking University

Is Feed-forward Neural Networks Powerful Enough?

- Feed-forward Neural Networks isn't Powerful Enough
  - There were no undirected cycles in the connectivity patterns;
  - Although neurons in the brain do contain undirected cycles as well as connections within layers;
  - We chose to impose these restrictions to simplify the training process at the expense of computational versatility.
- In order to create more powerful compuational systems, **directed cycles should be allowed in neural networks**
  - neurons can be connected to themselves.
  - RNNs are this kind of NN

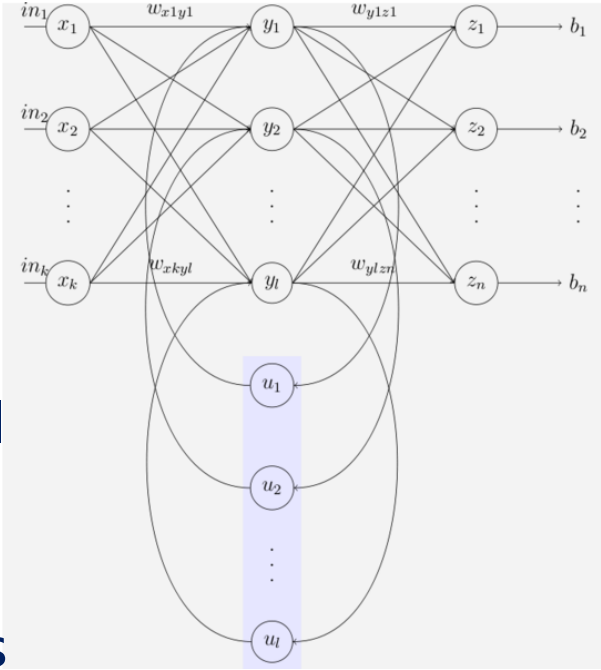
Is Feed-forward Neural Networks Powerful Enough?

- To create more powerful compuational systems, we allow RNNs to break these artificially imposed rules.
  - Thus, RNNs do not have to be organized in layers and directed cycles are allowed.
  - In fact, **neurons are actually allowed to be connected to themselves**.
  - One quite promising solution to tackling the problem of learning sequences of information.



Recurrent Neural Network

- RNN(Recurrent Neural Network)
  - all biological neural networks are recurrent;
  - RNNs implement dynamical systems mathematically, ;
- Some Old View :
  - RNNs can approximate **arbitrary** dynamical systems with **arbitrary** precision;
  - RNNs are (not often) proposed in technical articles as "in principle promising" solutions for difficult tasks.



State of usage in applications

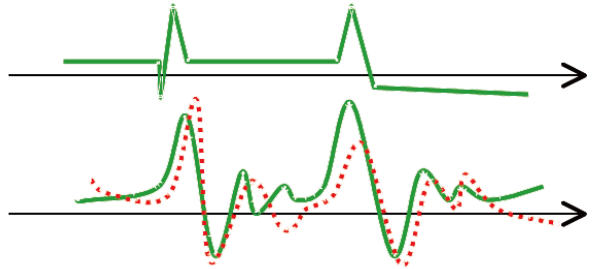
- Some relevant application areas:
  - Natural Language Processing
  - Image/Vision Processing
- Others
  - telecommunication
  - control of chemical plants
  - control of engines and generators
  - fault monitoring, biomedical diagnostics and monitoring
  - speech recognition
  - video data analysis

RNN的作用

A. Training

Teacher:

Model:

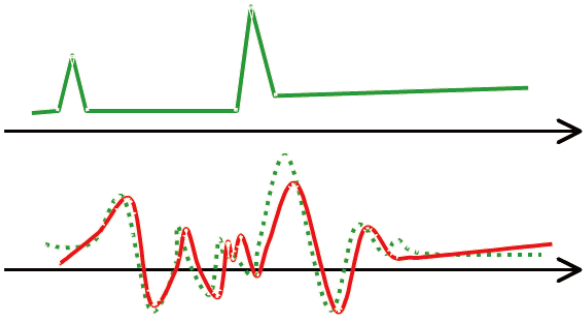


B. Exploitation

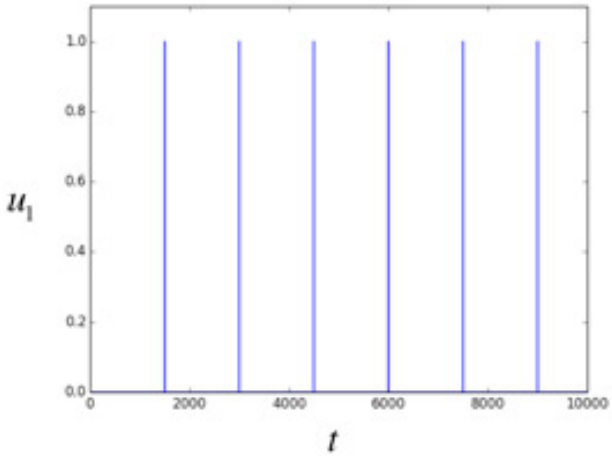
Input:

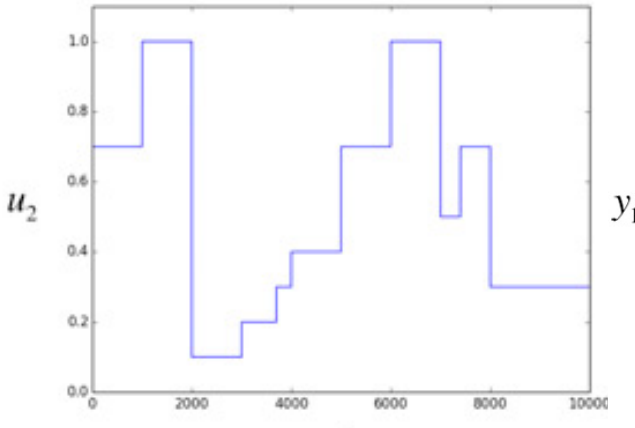
Correct (unknown) output:

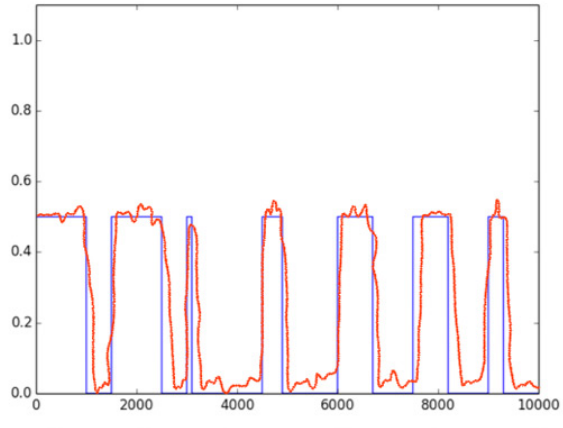
Model:

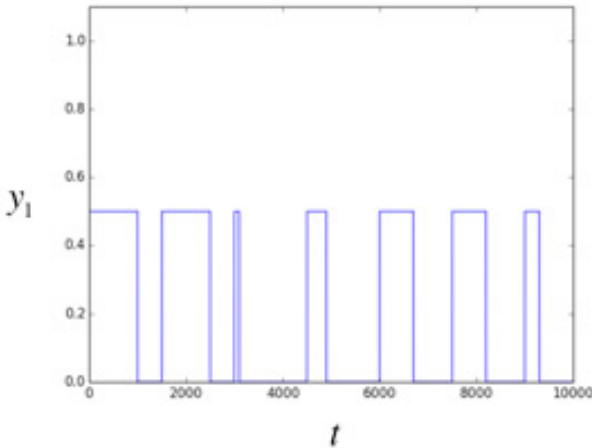


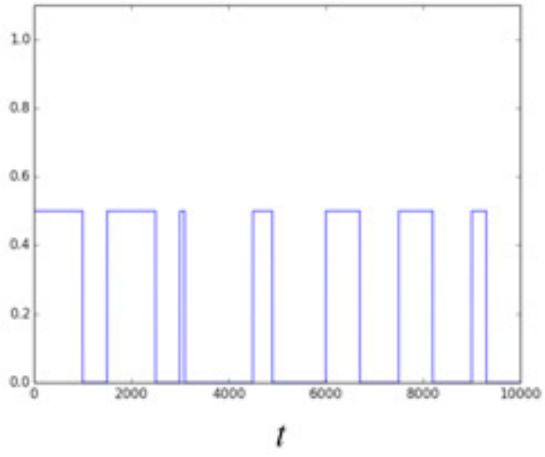
一个典型的例子


u_1
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
t


u_2
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000

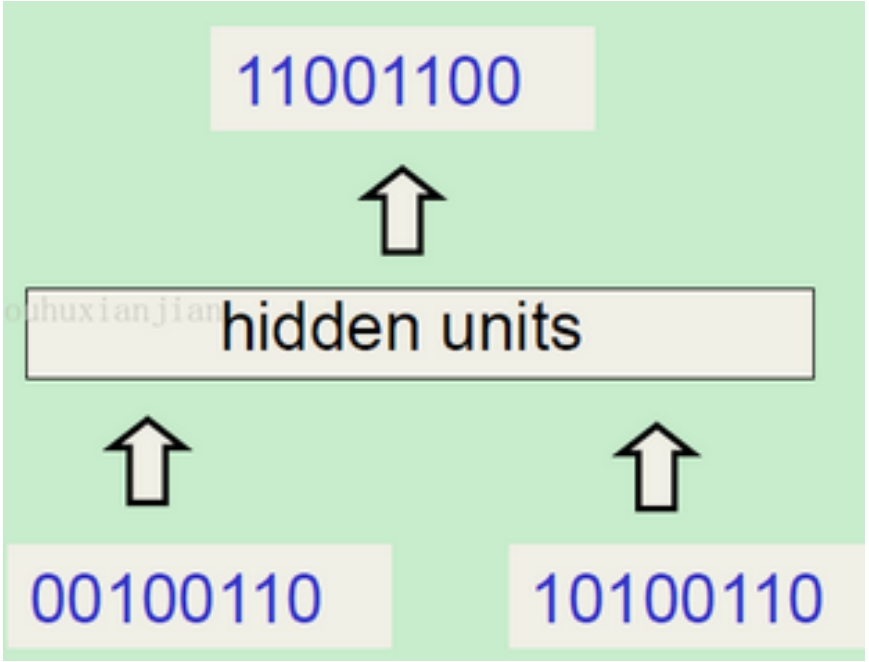

y_1
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000


y_1
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
t


y_1
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
t

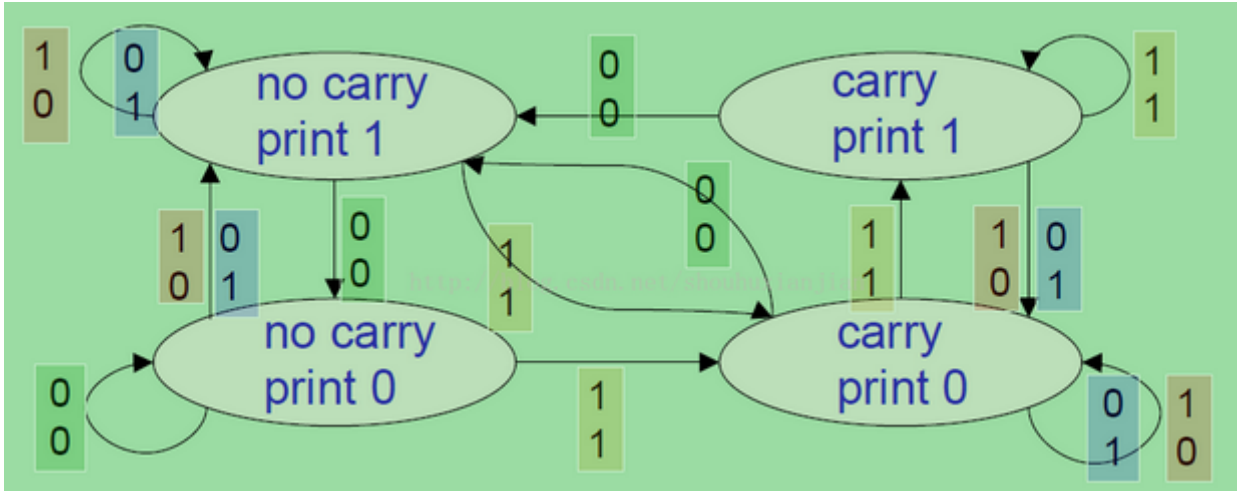
一个应用示例

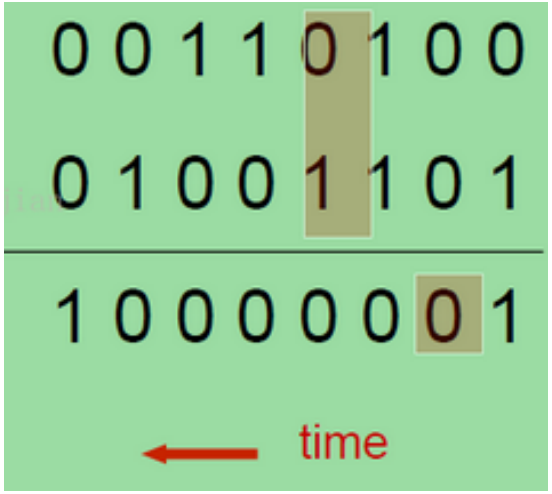
- 想一下
  - 如何构造一个前馈神经网络完成如下的学习？有什么问题？



一个应用示例

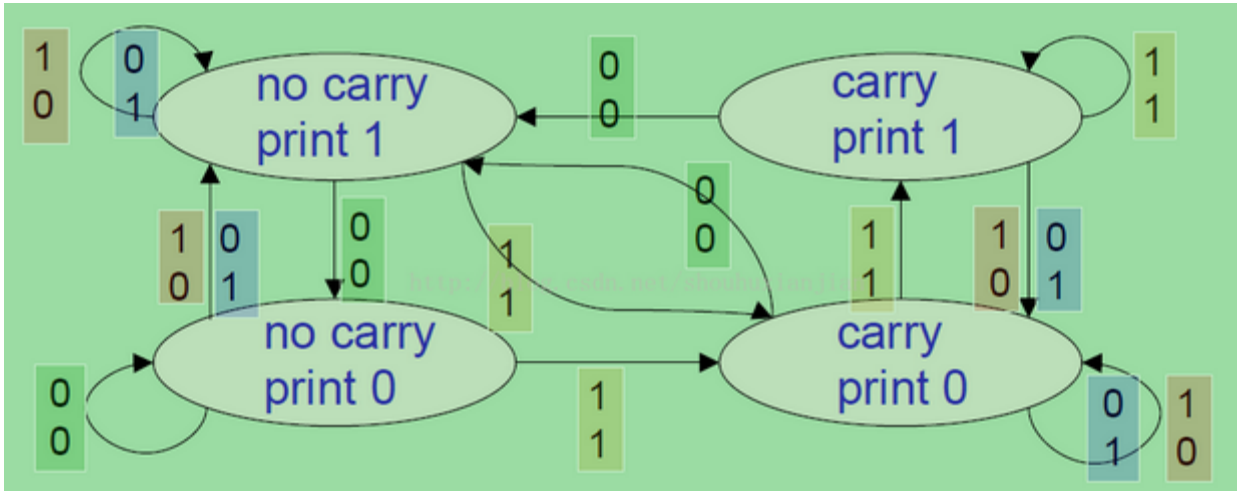
- 想一下
  - 如何构造一个前馈神经网络完成如下的学习？有什么问题？

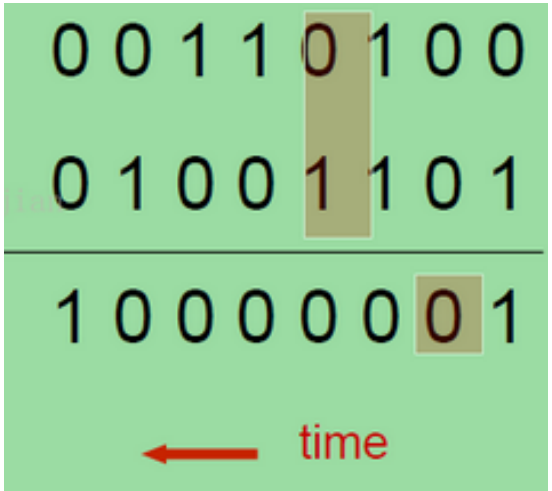




一个应用示例

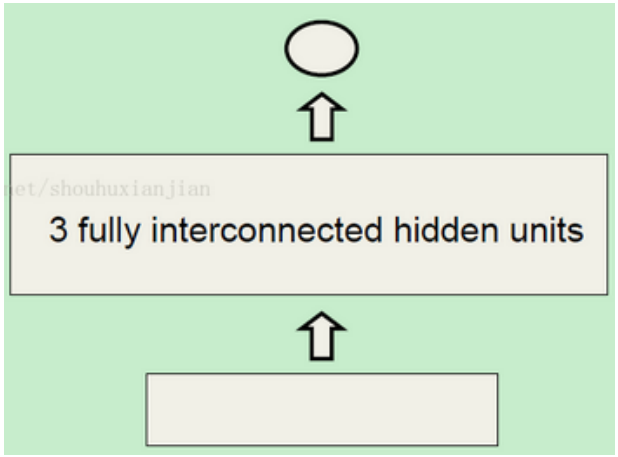
- 是否可以用RNN完成？
  - 几个输入？
  - 几个隐藏层？

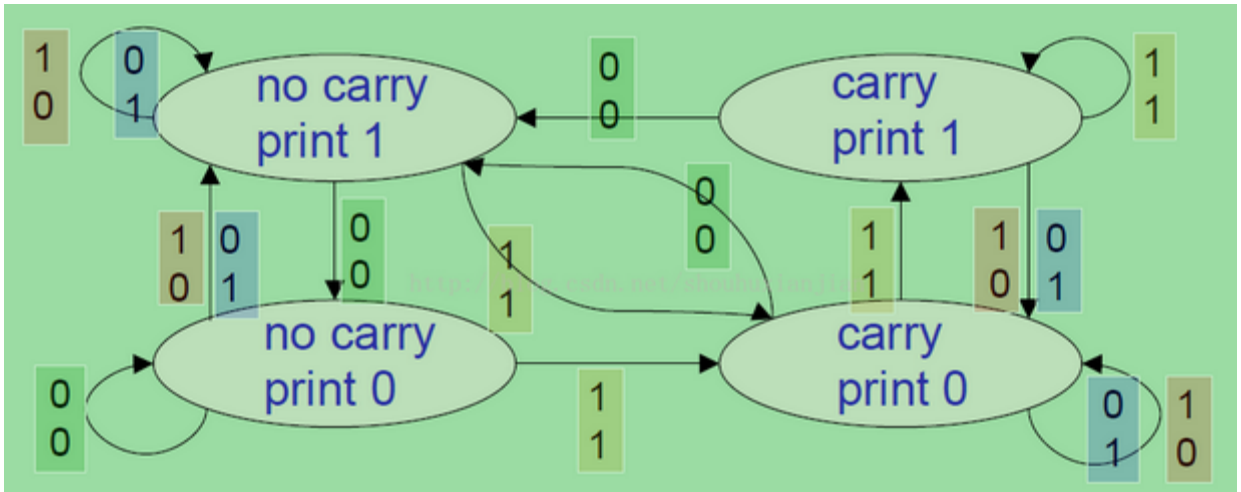


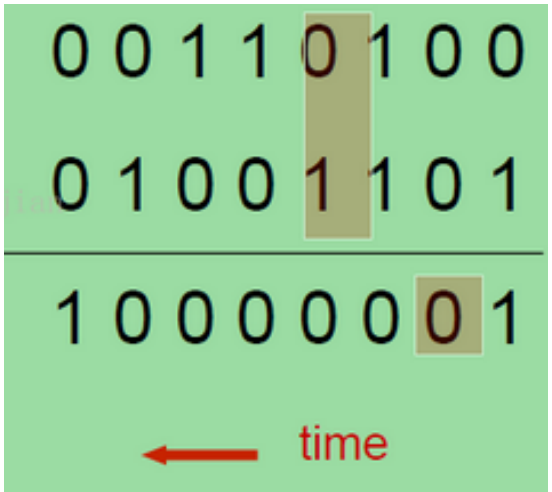


一个应用示例

- 是否可以用RNN完成？
  - 几个输入？
  - 几个隐藏层？

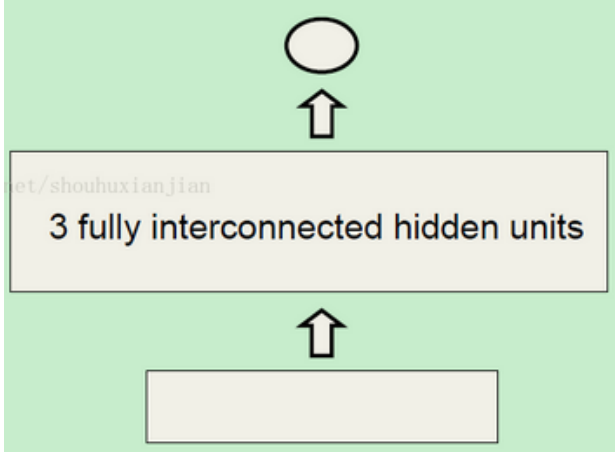


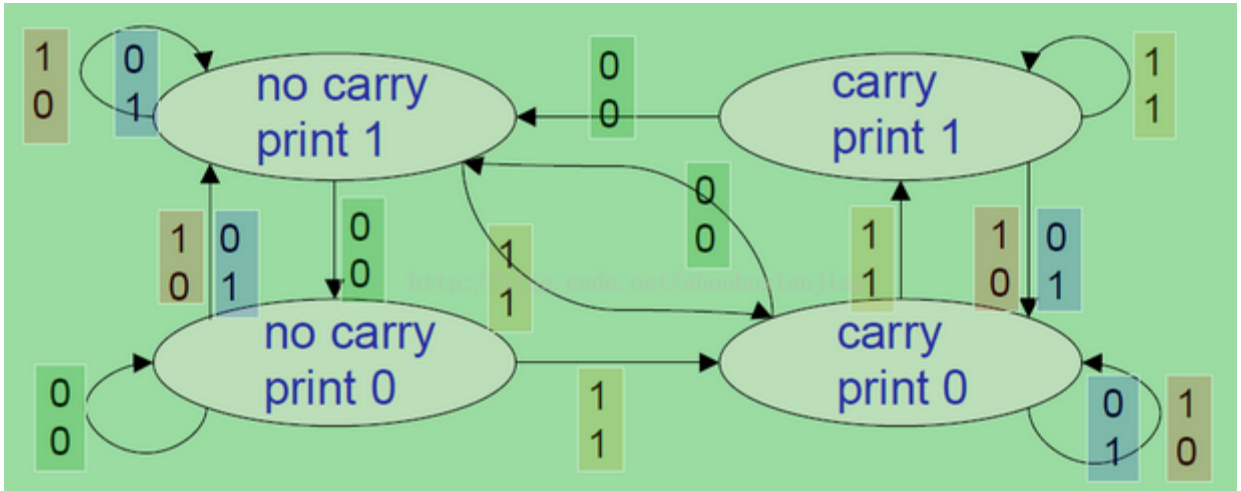


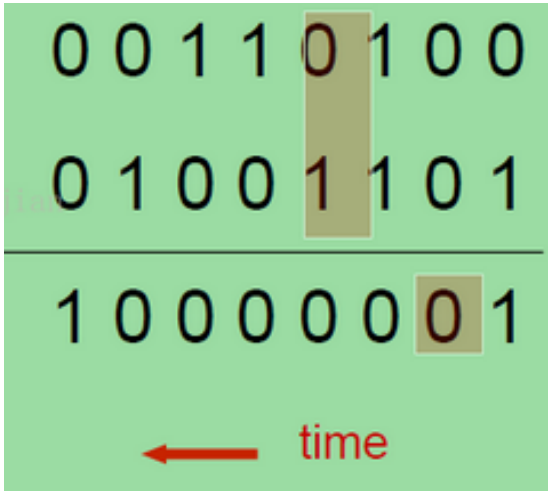


一个应用示例

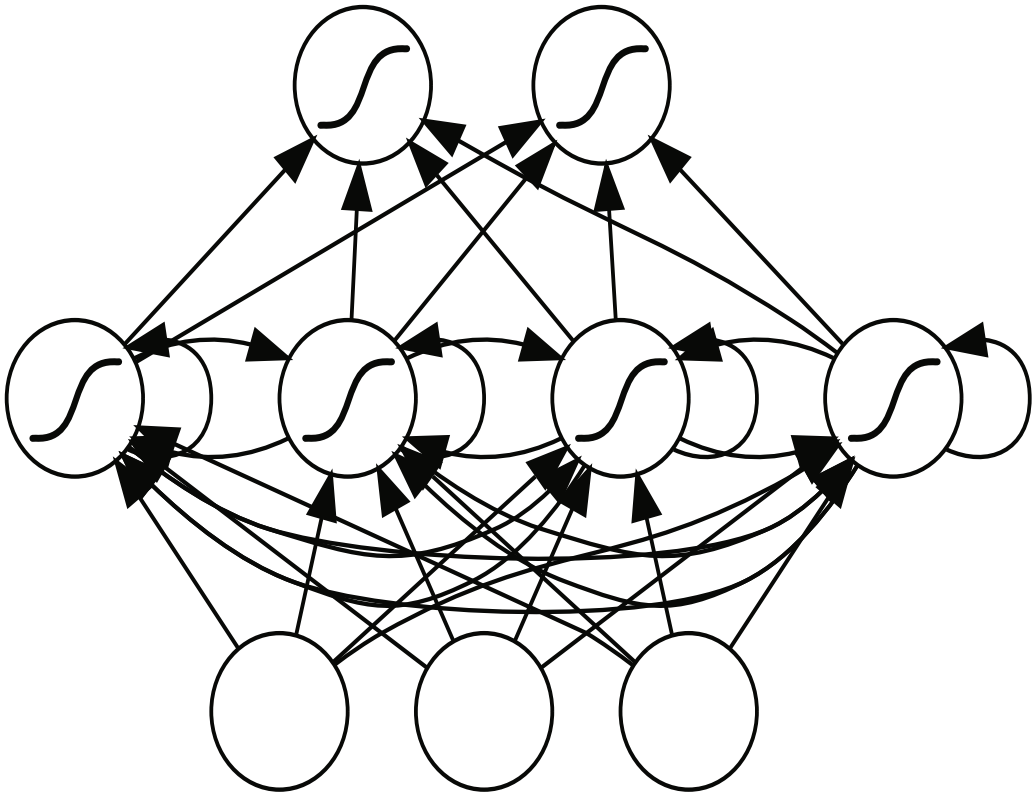
- RNN的使用场景
  - 需要抓取时间序列中隐藏的规律
  - 拓展到其他线性序列规律
  - “规律”隐藏在神经网络的循环层中







RNN

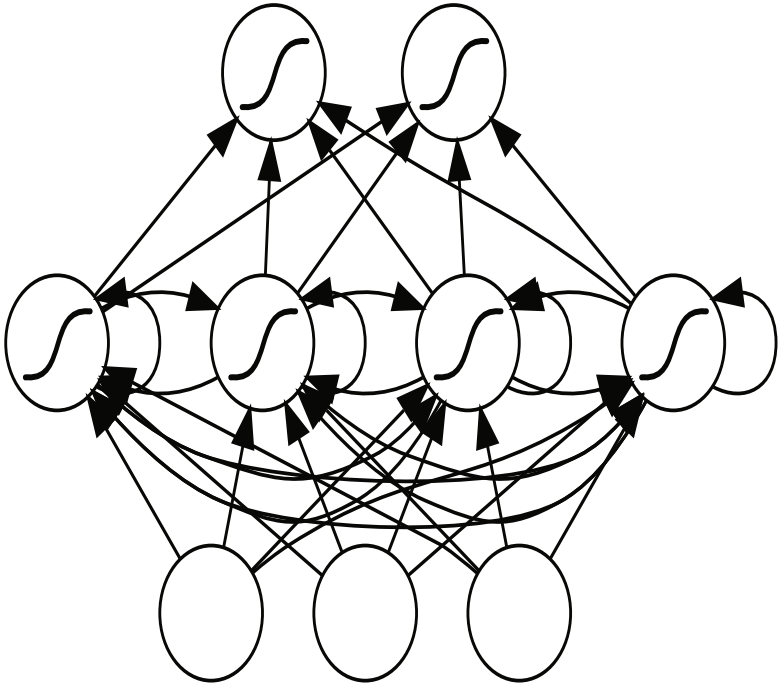


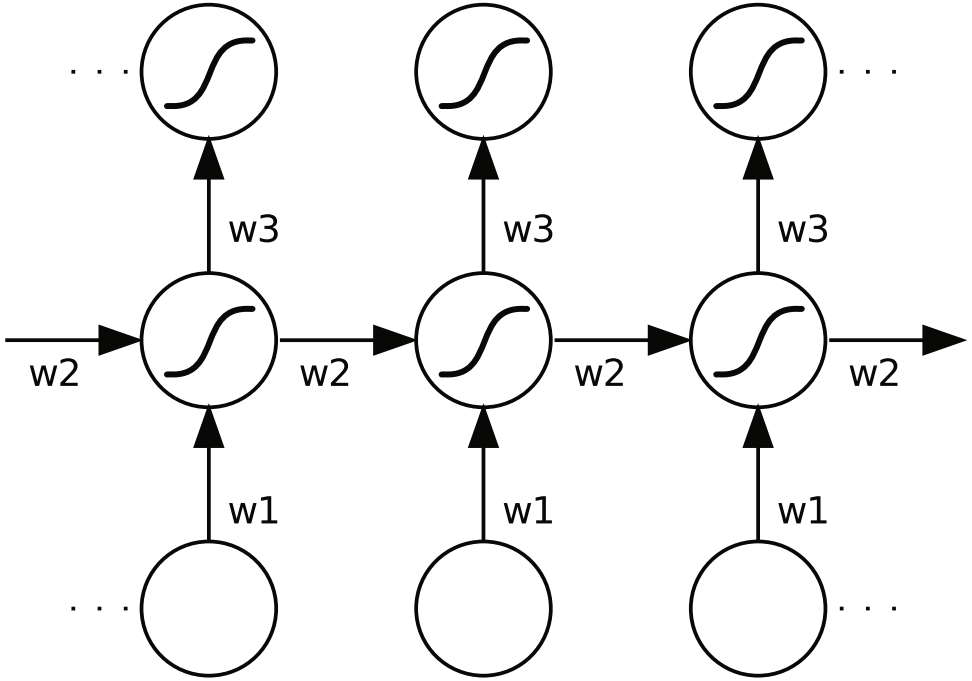
Output Layer

Hidden Layer

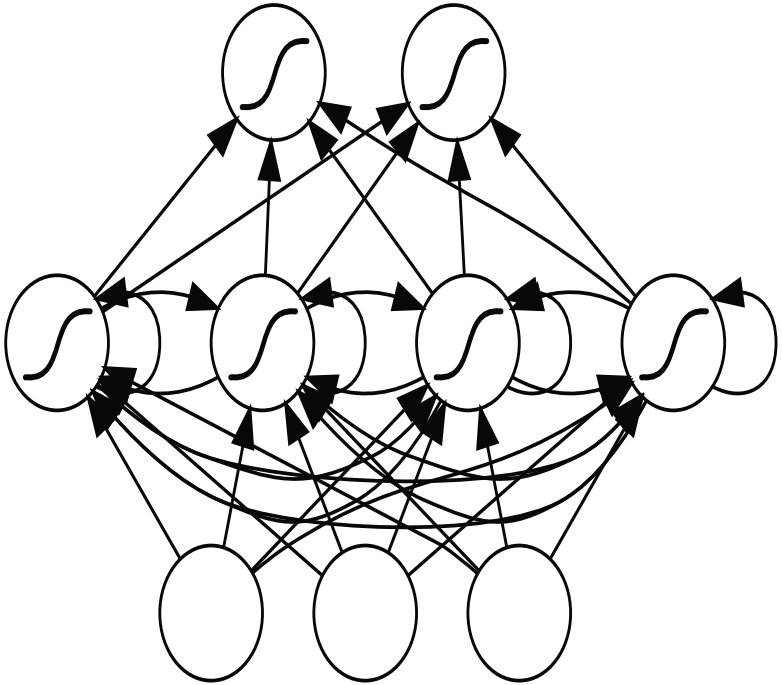
Input Layer

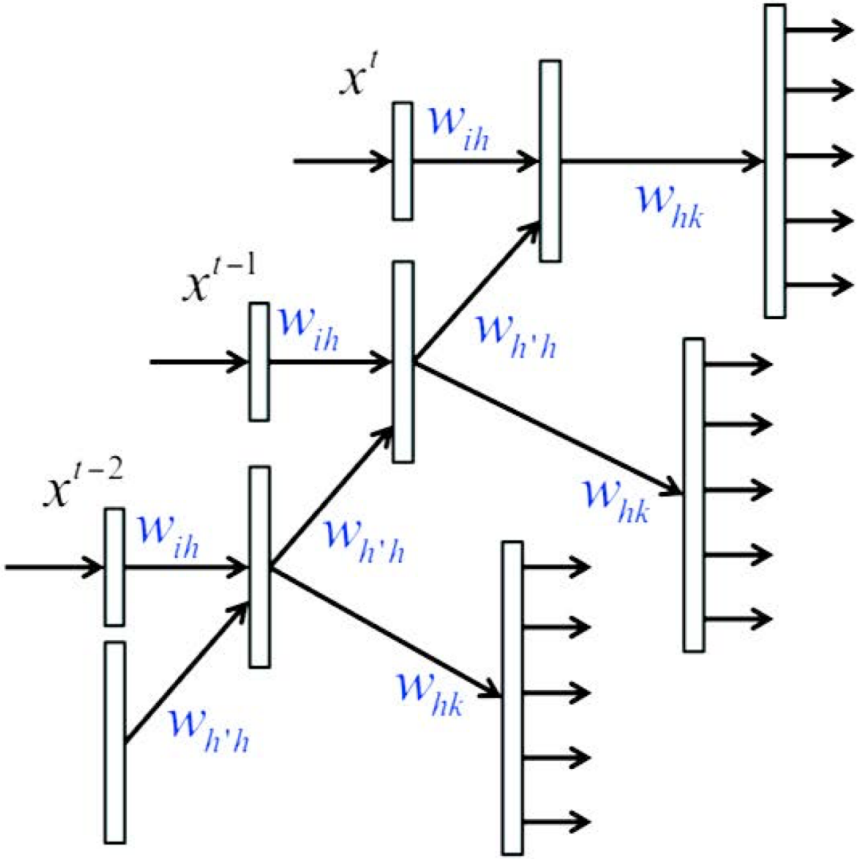
RNN



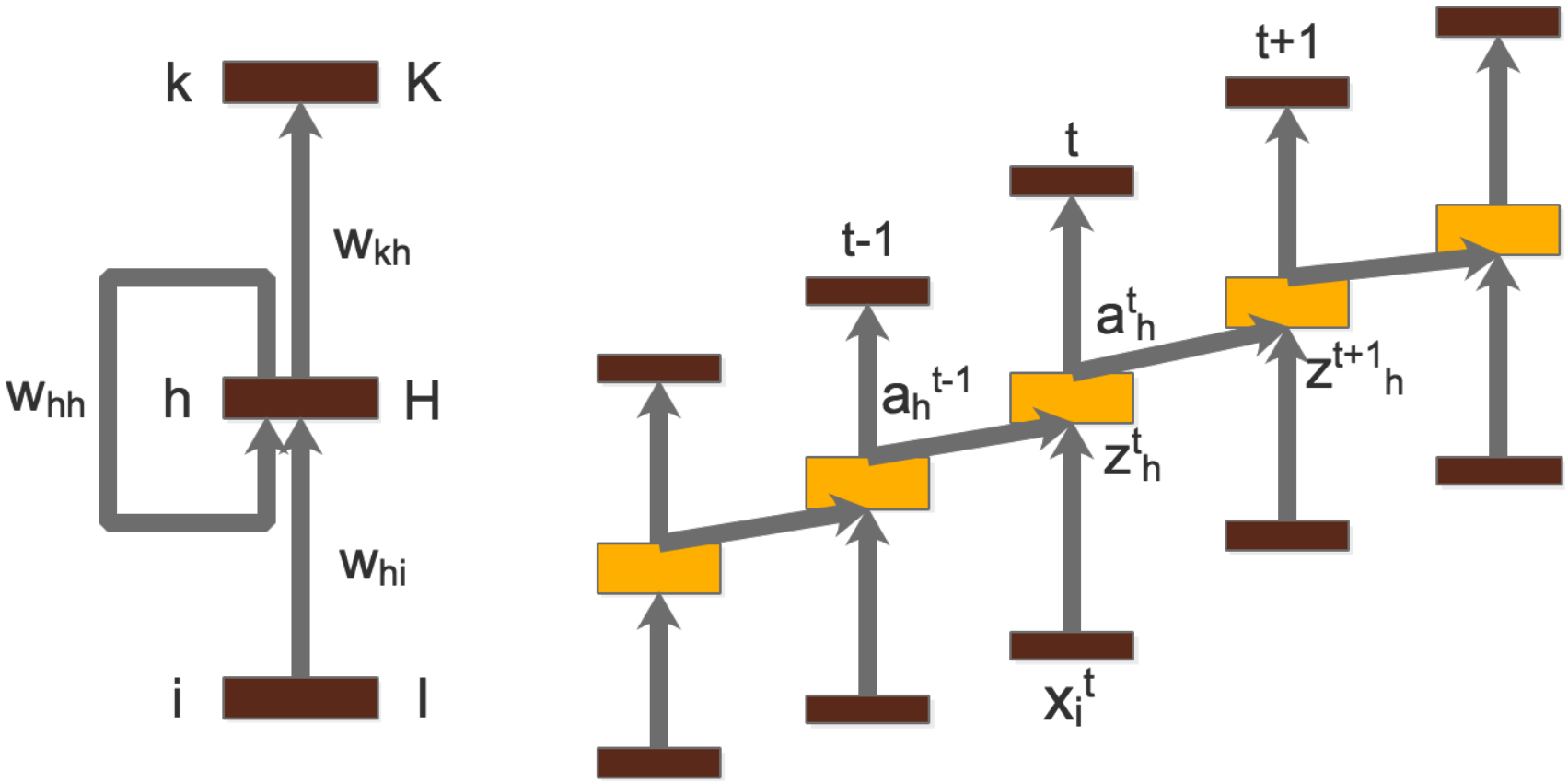


RNN





RNN 符号体系



RNN 前向传播推导

x: 长度为 T 的输入；x_i^t:t 时刻的输入 x 的第 i 维；
I: 输入层神经元个数；H: 隐藏层神经元个数；K: 输出层神经元个数；
z_j^t: 神经元 j 在 t 时刻的待激活输入；
a_j^t: 神经元 j 在 t 时刻的激活值；
J^t: 用 t 时刻的输出计算的代价函数；

$$z_h^t = \sum_{i=1}^{I} w_{hi} x_i^t + \sum_{h'=1}^{H} w_{hh'} a_{h'}^{t-1} \qquad a_h^t = f_h(z_h^t)$$

$$z_k^t = \sum_{h=1}^{H} w_{kh} a_h^t$$

$$J = \sum_{t=1}^{T} J^t(W, b)$$

其中，a_i^0 需要进行初始化，可以选择 0，也可以选择非零初始值。

RNN 后向传播推导

$$
\begin{aligned}
w_{kh} &= w_{kh} - \frac{\partial J}{\partial w_{kh}} \\
&= w_{kh} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial w_{kh}} \\
&= w_{kh} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_k^t} \frac{\partial z_k^t}{\partial w_{kh}}
\end{aligned}
$$

因为：$z_k^t = \sum_{h=1}^{H} w_{kh} a_h^t$ 所以：

$$
w_{kh} = w_{kh} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_k^t} a_h^t
$$

RNN 后向传播推导

$$w_{hh'} = w_{hh'} - \frac{\partial J}{\partial w_{hh'}} = w_{hh'} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial w_{hh'}}$$

$$= w_{hh'} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_h^t} \frac{\partial z_h^t}{\partial w_{hh'}}$$

因为：$z_h^t = \sum_{i=1}^{I} w_{hi} x_i^t + \sum_{h'=1}^{H} w_{hh'} a_{h'}^{t-1}$ **所以**：

$$= w_{hh'} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_h^t} a_{h'}^{t-1}$$

设：$\delta_h^t = \frac{\partial J^t(W,b)}{\partial z_h^t}$ **则**：$w_{hh'} = w_{hh'} - \sum_{t=1}^{T} \delta_h^t a_{h'}^{t-1}$

RNN 后向传播推导

$$w_{hi} = w_{hi} - \frac{\partial J}{\partial w_{hi}} = w_{hi} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial w_{hi}}$$

$$= w_{hi} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_h^t} \frac{\partial z_h^t}{\partial w_{hi}}$$

因为：$z_h^t = \sum_{i=1}^{I} w_{hi} x_i^t + \sum_{h'=1}^{H} w_{hh'} a_{h'}^{t-1}$ **所以**：

$$= w_{hi} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_h^t} x_i^t$$

设：$\delta_h^t = \frac{\partial J^t(W,b)}{\partial z_h^t}$ **则**：$w_{hi} = w_{hi} - \sum_{t=1}^{T} \delta_h^t x_i^t$

RNN 后向传播推导

因为：

$$z_h^t = \sum_{i=1}^{I} w_{hi} x_i^t + \sum_{h'=1}^{H} w_{hh'} a_{h'}^{t-1}$$

所以：

$$\begin{aligned}
\delta_h^t = \frac{\partial J(W,b)}{\partial z_h^t} &= \sum_{k=1}^{K} \frac{\partial J^t}{\partial z_k^t} \frac{\partial z_k^t}{\partial z_h^t} + \sum_{h=1}^{H} \frac{\partial J^{t+1}}{\partial z_h^{t+1}} \frac{\partial z_h^{t+1}}{\partial z_h^t} \\
&= \sum_{k=1}^{K} \frac{\partial J^t}{\partial z_k^t} \frac{\partial z_k^t}{\partial a_h^t} \frac{\partial a_h^t}{\partial z_h^t} + \sum_{h=1}^{H} \frac{\partial J^{t+1}}{\partial z_h^{t+1}} \frac{\partial z_h^{t+1}}{\partial a_{h'}^t} \frac{\partial a_{h'}^t}{\partial z_h^t} \\
&= \sum_{k=1}^{K} \delta_k^t w_{kh} f_h'(\cdot) + \sum_{h=1}^{H} \delta_h^{t+1} w_{hh'} f_h'(\cdot) \\
&= \left(\sum_{k=1}^{K} \delta_k^t w_{kh} + \sum_{h=1}^{H} \delta_h^{t+1} w_{hh'} \right) f_h'(\cdot)
\end{aligned}$$

RNN 后向传播推导

$$w_{hk} = w_{hk} - \sum_{t=1}^{T} \frac{\partial J^t(W,b)}{\partial z_k^t} a_h^t$$

$$w_{hi} = w_{hi} - \sum_{t=1}^{T} \delta_h^t x_i^t$$

$$w_{h'h} = w_{h'h} - \sum_{t=1}^{T} \delta_h^t a_{h'}^{t-1}$$

$$\delta_h^t = \left(\sum_{k=1}^{K} \delta_k^t w_{kh} + \sum_{h'=1}^{H} \delta_h^{t+1} w_{hh'} \right) f_h'(\cdot)$$

Jump out of Neural Network

Acknowledgement: Thanks Lili Mou' Contribution.

Probability

Kolmogorov (1933):

- Nonnegative

$$p(x) \geq 0, \forall x$$

- Normalized

$$\sum_x p(x) = 1$$

- Finitely/countably additive
  Let $A_1, A_2, \cdots, A_n$ be disjoint events,

$$p\left(\bigcup A_i\right) = \sum_i p(A_i)$$

- The probability that a toss of a coin gives the head
- The probability that it will rain tomorrow
- The probability that the speed of light lies in 2.9-3.1$\times 10^8$m/s

Probability

- Frequentist
  - The limit of frequency provided that the number of samples goes to infinity (Recall the Law of Large Numbers)
- Bayesian
  - The degree of ones subjective belief
  - Is belief necessarily a kind of probability?
  - Is belief admissible in scientific research? or even unavoidable?

Probability

- Some even argue that the frequency concept never applies, it being impossible to have an infinite sequence of i.i.d repetitions of any situation, except in a certain imaginary (subjective) sense. “(James, 1985)
- The subjectivist states his judgements, whereas the objectivist sweeps them under the carpet by calling assumptions knowledge, and he basks under the glorious objectivity of science." (Good, 1973)

James O. Berger, Statistical Decision Theory and Bayesian Analysis (2 ed.), Springer-Verlag, 1985.

Bifurcation of the Two Schools

- We have the data D, and the model, parametrized by θ.

$$\mathcal{D} \sim p_{\Theta}(\cdot)$$

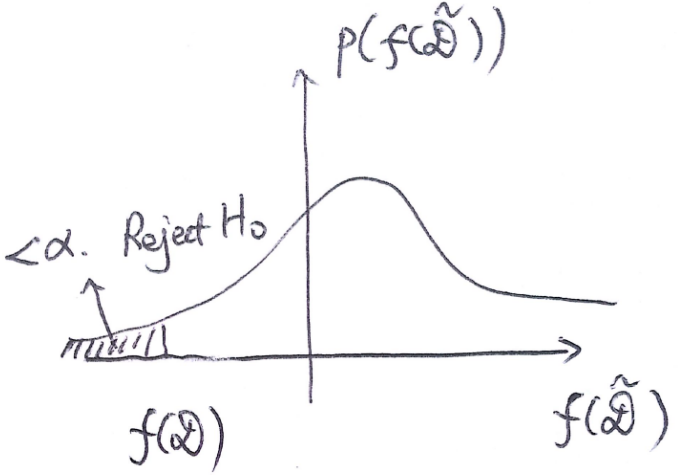
- What do we take expectation on (for learning, inference, etc)?
- Frequentist:
  - D, because there is nothing random about θ
  - No random, no cry!
- Bayesian: ,
  - θ, because D is known
  - Everything unknown is a random variable!

Pathologies of Frequentist

- Hypothesis test
  - $H_0 \leftrightarrow H_1$
  - Data $\mathcal{D}$
  - Test statistic: $f(\mathcal{D})$
  - p-value$(\mathcal{D}) = \Pr\{f(\tilde{\mathcal{D}}) > f(\mathcal{D}) | \tilde{\mathcal{D}} \sim H_0\}$
  - Reject H_0, if p-value$< \alpha$
  - Cannot reject H_0, if p-value$\geq \alpha$

Pathologies of Frequentist

- Hypothesis test
  - Tail Area Probability



Assume $\mathcal{N}(\mu, 1)$

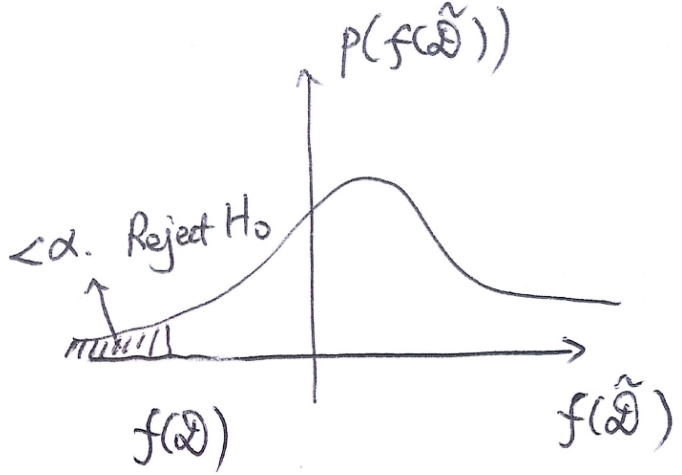
- $H_0 : \mu = 0 \leftrightarrow H_1 : \mu \neq 0$
- True probability: $\mathcal{N}(0.2, 1)$
- $\mathcal{D}_1 = \{0.2\}$, $\mathcal{D}_2$=10000 samples with mean 0.2

Given $\mathcal{D}_1$, we cannot reject H_0.
Given $\mathcal{D}_2$, we do reject H_0.

Pathologies of Frequentist

- Hypothesis test
  - Tail Area Probability

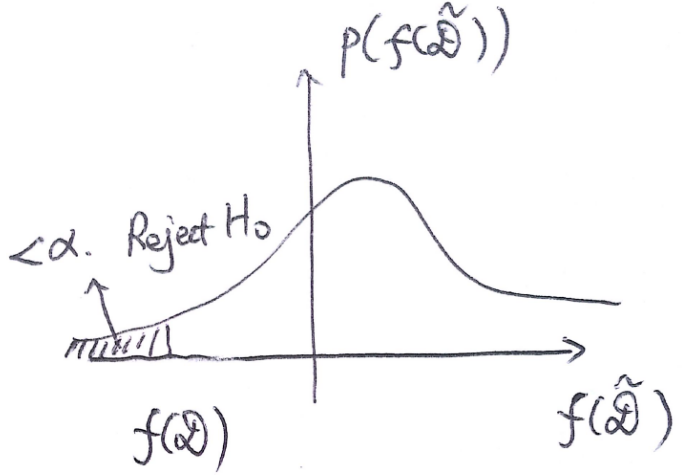
Who cares about nonoccurrence?



First peculiar property: Because p-value relies on the tail area probability, a hypothesis which may be true may be rejected because it has not predicted observable results which have not occurred." (Jeffreys, 1961)

Pathologies of Frequentist

- Hypothesis test
  - Tail Area Probability
  - For most scientific problems, the only thing that matters is the sample size, which is under control of researchers.



- "In a recent survey, 58% of researchers admitted to having collected more data after looking to see whether the results were significant and 22% admitted to stopping an experiment early because they had found the result that they were looking for." (Sanborn et al., 2014)

Pathologies of Frequentist

Confidence Interval

$$C_\alpha(\theta) = (l, u) : \Pr\{l(\tilde{\mathcal{D}}) \leq \theta \leq u(\tilde{\mathcal{D}}) | \tilde{\mathcal{D}} \sim \theta\} = 1 - \alpha$$

Counter-intuitive explanation:

- The confidence level (e.g., 95%) is **NOT** the probability that θ lies in the interval, given $\mathcal{D}$.
- It is the probability that the interval covers θ if we repeatedly draw datasets $\tilde{\mathcal{D}}$ (in addition to $\mathcal{D}$ *per se*).
- However, we notice that $\mathcal{D}$ is **KNOWN**. Who cares about nonoccurrence?

Pathologies of Frequentist

- An Interesting Story
  - “Suppose a substance to be analyzed can be sent either to a laboratory in New York or a laboratory in California. The two labs seem equally good, so a fair coin is flipped to choose between them, which "heads" denoting that the lab in New York will be chosen. The coin is flipped and comes up tails, so the California lab is used.
  - After a while, the experimental results come back and a conclusion and report must be developed. Should this conclusion take into account the fact that the coin could have been heads, and hence that the experiment in New York might have been performed instead?"

For Machine Learning

Linear Classification

Assume

$$p(y = i|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x})$$

Given a set of training samples $\boldsymbol{x}_i, y_i$, we would like to predict $y_* =?$ for a new sample $\boldsymbol{x}_*$.

For Machine Learning

Frequentists' Viewpoint

Minimize the expected risk (loss) over $\boldsymbol{x}_*$

$\Downarrow$

Minimize the empirical risk over $\boldsymbol{x}_i, y_i$ (training samples)

$\Downarrow$

Maximize the likelihood of $\boldsymbol{x}_i, y_i$

Maximum likelihood estimation

Training:

$$\boldsymbol{w}^* \leftarrow \arg\max_{\boldsymbol{w}} p(y_i|\boldsymbol{w}, \boldsymbol{x}_i)$$

Predicting:

$$y_* \leftarrow \arg\max_{y} p(y_*|\boldsymbol{x}_*, \boldsymbol{w}^*)$$

For Machine Learning

Bayesian Learning

- There does not exist $\boldsymbol{w}^*$.
- Rather, $\boldsymbol{w}$ is a random variable that we have to marginalize out.
- Predictive density

$$p(y_*|\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{w}\ p(\boldsymbol{w}|\boldsymbol{y})p(y_*|\boldsymbol{w})$$

Let $\boldsymbol{y} \in \{0,1\}^m$ denote the labels of training data $\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_m$

Prior $p(\boldsymbol{w})$, which is a \$64,000,000 question

Likelihood

$$p(\boldsymbol{y}|\boldsymbol{w}) = \prod_{i=1}^{m} p(y^{(i)}|\boldsymbol{w}) = \prod_{i=1}^{m} \sigma\left(\boldsymbol{w}^T \boldsymbol{\phi}^{(i)}\right)^{t^{(i)}} \left(1 - \sigma\left(\boldsymbol{w}^T \boldsymbol{\phi}^{(i)}\right)\right)^{1-t^{(i)}}$$

Posterior

$$p(\boldsymbol{w}|\boldsymbol{y}) = \frac{p(\boldsymbol{w})p(\boldsymbol{y}|\boldsymbol{w})}{p(\boldsymbol{y})} \underset{\boldsymbol{w}}{\propto} p(\boldsymbol{w})p(\boldsymbol{y}|\boldsymbol{w}) = p(\boldsymbol{w}) \prod_{i=1}^{m} \sigma(\cdot)^{t^{(i)}} (1-\sigma(\cdot))^{1-t^{(i)}}$$

Predictive density

$$p(y_*|\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{w}\ p(y_*|\boldsymbol{w}) \cdot p(\boldsymbol{w}|\boldsymbol{y})$$

$$\underset{y_*}{\propto} \int \mathrm{d}\boldsymbol{w}\ \sigma\left(\boldsymbol{w}^T \boldsymbol{\phi}_*\right) \cdot p(\boldsymbol{w}) \prod_{i=1}^{m} \sigma(\cdot)^{t^{(i)}} (1 - \sigma(\cdot))^{1-t^{(i)}}$$

Bayesian Learning

$$p(\mathcal{C}_1|\boldsymbol{t}) \simeq \int \sigma(a)p(a)\,\mathrm{d}a = \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)\,\mathrm{d}a$$

$$\simeq \int \Phi(\lambda a)\mathcal{N}(a|\mu_a, \sigma_a^2)\,\mathrm{d}a = \Phi\left(\frac{\mu_a}{(\lambda^{-2}+\sigma_a^2)^{1/2}}\right) \simeq \sigma\left(\kappa(\sigma_a^2)\mu_a\right)$$

$$\mu_a = \mathbb{E}[a] = \int p(a)a\,\mathrm{d}a = \int q(\boldsymbol{w})\boldsymbol{w}^T\boldsymbol{\phi}_*\,\mathrm{d}\boldsymbol{w} = \boldsymbol{w}_{\mathsf{MAP}}^T\boldsymbol{\phi}_*$$

$$\sigma_a^2 = \mathsf{var}[a] = \int p(a)\left\{a^2 - \mathbb{E}[a]^2\right\}\mathrm{d}a \qquad \kappa(\sigma_a^2) = (1+\pi\sigma_a^2/8)^{-1/2}$$

$$= \int q(\boldsymbol{w})\left\{(\boldsymbol{w}^T\boldsymbol{\phi})^2 - (\boldsymbol{m}_N^2\boldsymbol{\phi})^2\right\}\mathrm{d}\boldsymbol{w}$$

$$= \phi^T\boldsymbol{S}_N\boldsymbol{\phi}$$

Prior: Gaussian, which is natural[1]

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{S}_0)$$

Conclusions

- Make a choice between:

$$\text{Sufficiency} + \text{Weak Conditionality Principle} \Rightarrow \text{Bayesian Analysis}$$

  - Frequentist takes expectation on (known) data while conditioning on (unknown) θ
  - Conditionalist (Bayesian) takes expectation on (unknown) θ while conditioning on (known) data
  - Bayesian treatment is fundamentally correct by computationally non-trivial.

L_pPooling

L_pPooling[1] 是一种泛化能力较强的 Pooling 方法；

$$y_{i,j,k} = \left[\sum_{(m,n) \in R_{ij}} (a_{m,n,k})^p \right]^{\frac{1}{p}}$$

- 当 $p = 1$ 时，L_pPooling 相当于“Average Pooling”;
- 当 $p = \infty$ 时，L_pPooling 相当于“Max Pooling”;

[1]A. Hyvarinen, U. Koster, Complex cell pooling and the statistics of natural images, in: NCNS, 2007.

Mixed Pooling

Mix Pooling[1] 是一种 Max Pooling 与 Mean Pooling 混合的方式：

$$y_{i,j,k} = \lambda \, max_{(m,n)\in R_{ij}} a_{m,n,k} + (1-\lambda)\frac{1}{|R_{ij}|}\sum_{(m,n)\in R_{ij}} a_{m,n,k}$$

- Inspired by random Dropout and DropConnect.
- Experiments in show that mixed pooling can better address the overfitting problems and it performs better than max pooling and average pooling.

[1]D. Yu, H. Wang, P. Chen, Z. Wei, Mixed pooling for convolutional neural networks, in: Rough Sets and Knowledge Technology, 2014.

Stochastic Pooling

- Stochastic pooling[1] randomly picks the activations according to a multinomial distribution.
- Stochastic pooling first computes the probabilities p for each region R_j by normalizing the activations within the region:

$$p_i = \frac{a_i}{\sum_{k \in R_j}(a_k)}$$

- Compared with max pooling, stochastic pooling can avoid overfitting due to the stochastic component.

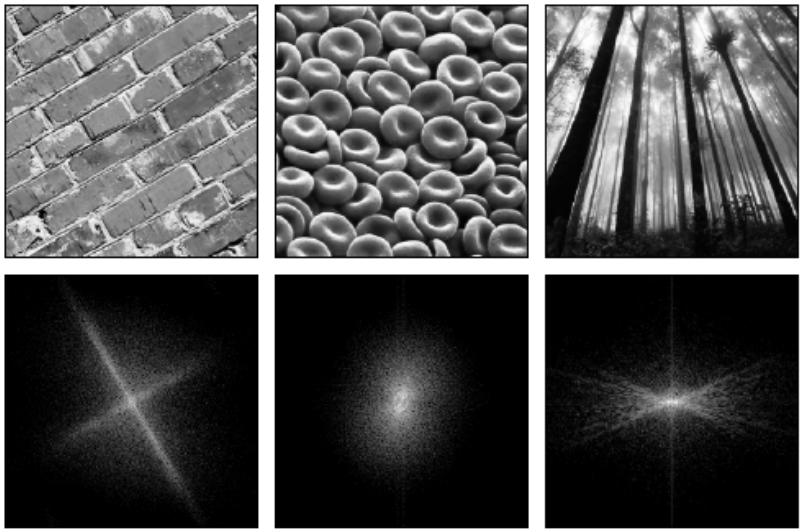
[1]M. D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, in: ICLR, 2013.

Spectral Pooling

通过离散傅立叶变换，将特征图像转换到频域再进行 Pooling[1]：
对于输入 $x \in C^{M \times N}$，其对应的傅立叶转换为：

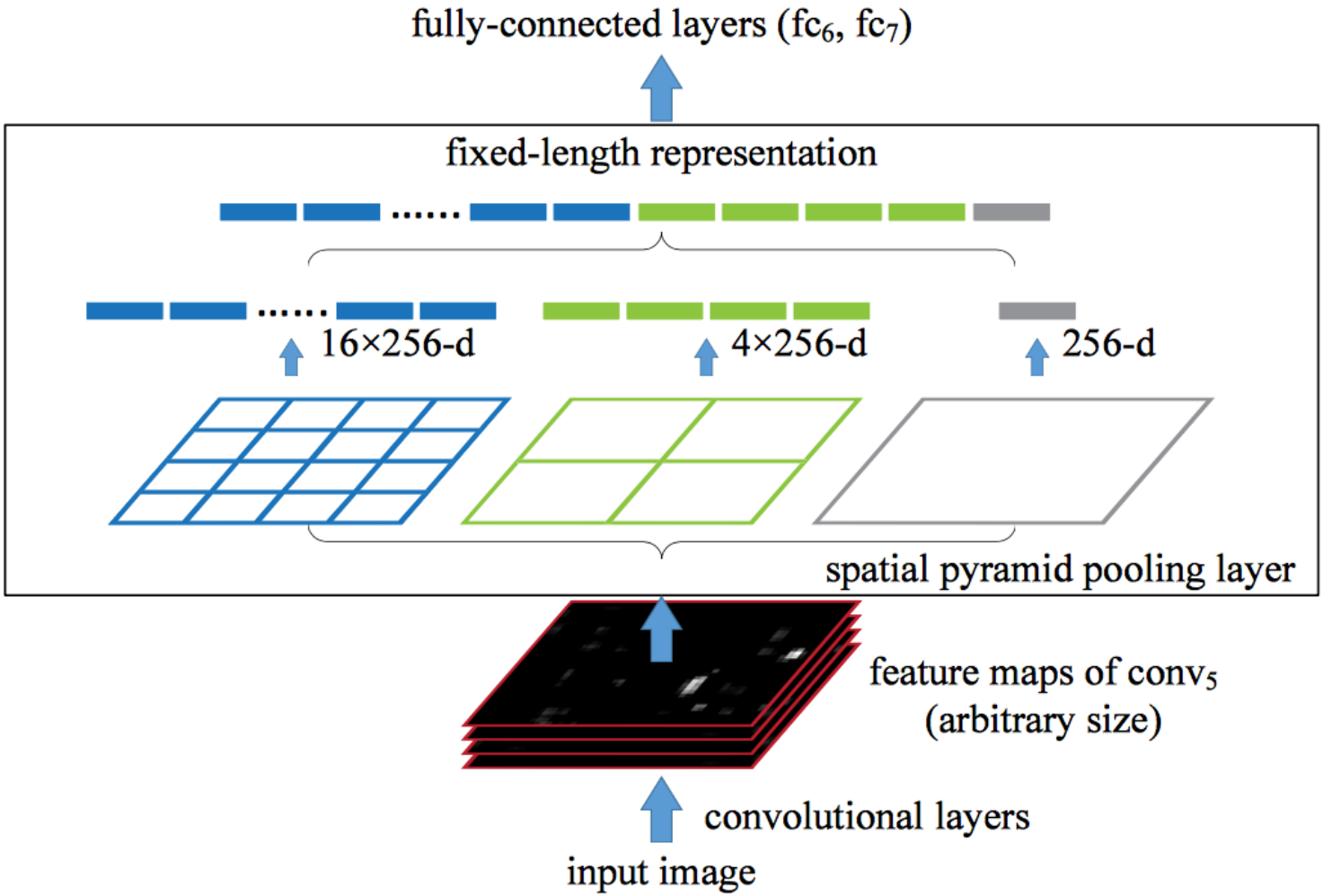
$$F(x)_{hw} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{mn} e^{-1\pi i(\frac{mh}{M})+\frac{nw}{N}}$$

$$\forall h \in 0, \ldots, M-1, \forall w \in 0, \ldots, N-1.$$



[1] O. Rippel, J. Snoek, R. P. Adams, Spectral representations for convolutional neural networks, in: NIPS, 2015.

Spatial Pyramid Pooling



[1] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: ECCV, 2014.

Thanks.