北京大學

复习：深度学习部分

主讲人：董豪　讲义：董豪

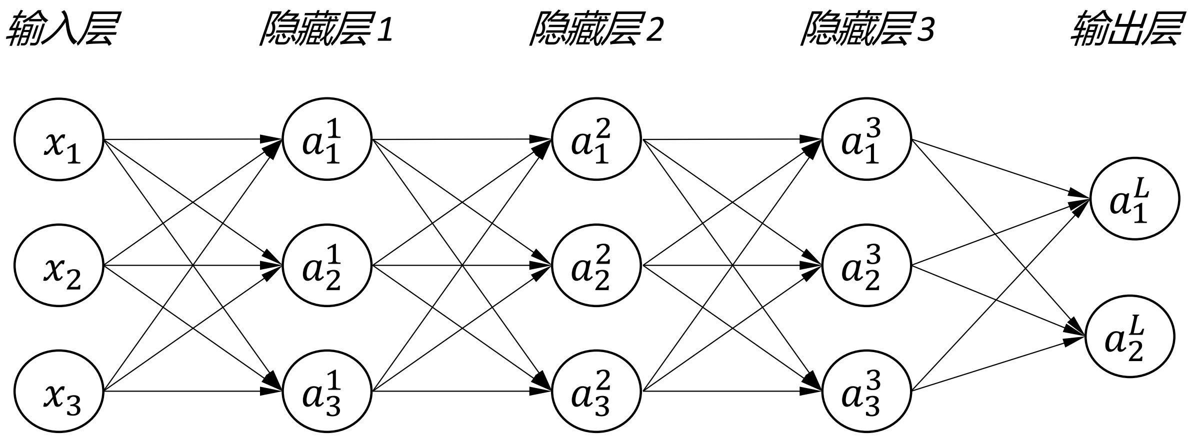
神经网络基础

- 单个神经元 Single Neuron
- 激活函数 Activation Functions
- 多层感知器 Multi-layer Perceptron
- 损失函数 Loss Functions
- 优化 Optimisation
- 正则化 Regularisation
- 实现 Implementation

误差反向传播

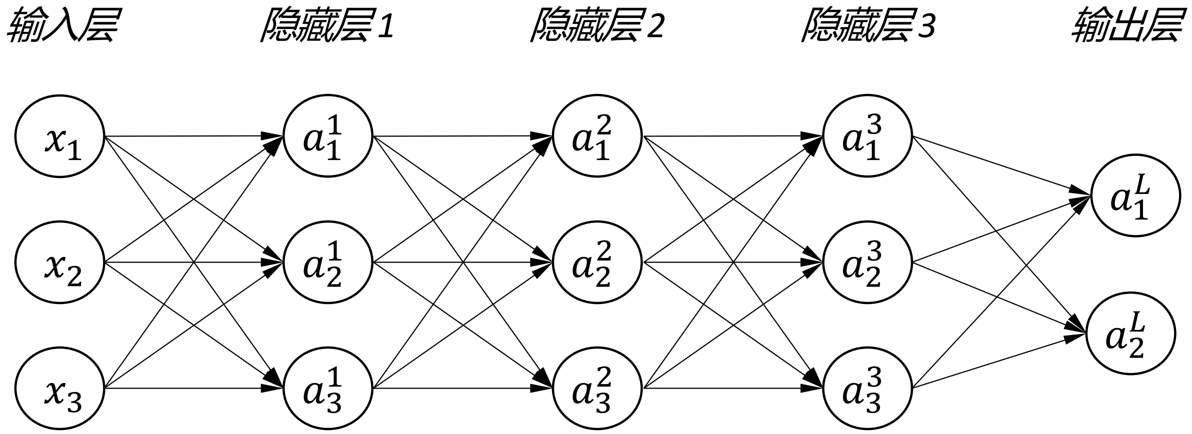
- 误差反向传播（Error Back-Propagation）

误差反向传播是用来计算网络中所有参数的梯度 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}$ 的方法。计算梯度时，引入对$\mathcal{L}$求输出值$\boldsymbol{z}$的偏导 $\delta = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}$，作为中间结果（intermediate result)，基于这个中间结果来计算出每一层的梯度$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}$。



误差反向传播

- 误差反向传播（Error Back-Propagation）



层索引（layer index） $l = 1 \dots L$ 代表第一层隐藏层（1）到输出层（L）

输入层用 $\boldsymbol{x} = \boldsymbol{a}^0$ 来表示

$$\boldsymbol{a}^l = f(\boldsymbol{z}^l) = \frac{1}{1+e^{-\boldsymbol{z}^l}}$$

我们通过这个简单模型和损失函数来讲解

$$\boldsymbol{z}^l = \boldsymbol{W}^{l^T}\boldsymbol{a}^{l-1} + \boldsymbol{b}^l$$

$$\mathcal{L} = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{a}^L)^2$$

误差反向传播

- 误差反向传播（Error Back-Propagation）：列格式（column format）教材常用

1. 已知

- $\boldsymbol{a}^l = f(\boldsymbol{z}^l) = \frac{1}{1+e^{-\boldsymbol{z}^l}}$
- $\boldsymbol{z}^l = \boldsymbol{W}^{l^T}\boldsymbol{a}^{l-1} + \boldsymbol{b}^l$
- $\mathcal{L} = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{a}^L)^2$

2. 则有如下求导

- $\frac{\partial \boldsymbol{a}^l}{\partial \boldsymbol{z}^l} = f'(\boldsymbol{z}^l) = \boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l)$
- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} = (\boldsymbol{a}^L - \boldsymbol{y})$
- $\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{W}^l} = \boldsymbol{a}^{l-1}$ and $\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{b}^l} = 1$

逐点相乘
Hadamard (element-wise) product

3. 输出层的中间结果 $\boldsymbol{l} = \boldsymbol{L}$

- $\delta^L = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^L} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L}\frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} = (\boldsymbol{a}^L - \boldsymbol{y}) \circ (\boldsymbol{a}^L \circ (1 - \boldsymbol{a}^L))$

链式法则（chain rule）

4. 其他层的中间结果 $\boldsymbol{l} = \boldsymbol{1} \dots \boldsymbol{L} - \boldsymbol{1}$

- $\delta^l = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{l+1}}\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l} = \delta^{l+1}\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l}$
  - $\boldsymbol{z}^{l+1} = \boldsymbol{W}^{l+1^T}\boldsymbol{a}^l + \boldsymbol{b}^{l+1}$
  - $\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l} = \boldsymbol{W}^{l+1^T} f'(\boldsymbol{z}^l) = \boldsymbol{W}^{l+1^T} \circ (\boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l))$
- $\delta^l = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{l+1}}\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l} = \boldsymbol{W}^{l+1^T}\delta^{l+1} \circ (\boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l))$

5. 则有梯度

- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l}\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{W}^l} = \delta^l\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{W}^l} = \delta^l\boldsymbol{a}^{l-1^T}$
- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l}\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{b}^l} = \delta^l\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{b}^l} = \delta^l$

6. 更新参数

$$\boldsymbol{W}^l \coloneqq \boldsymbol{W}^l - \alpha\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^l} \qquad \boldsymbol{b}^l \coloneqq \boldsymbol{b}^l - \alpha\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^l}$$

误差反向传播

- 误差反向传播（Error Back-Propagation）：行格式（row format）编程常用

1. 已知

- $\boldsymbol{a}^l = f(\boldsymbol{z}^l) = \frac{1}{1+e^{-\boldsymbol{z}^l}}$
- $\boldsymbol{z}^l = \boldsymbol{a}^{l-1}\boldsymbol{W}^l + \boldsymbol{b}^l$
- $\mathcal{L} = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{a}^L)^2$

2. 则有如下求导

- $\frac{\partial \boldsymbol{a}^l}{\partial \boldsymbol{z}^l} = f'(\boldsymbol{z}^l) = \boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l)$
- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} = (\boldsymbol{a}^L - \boldsymbol{y})$
- $\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{W}^l} = \boldsymbol{a}^{l-1}$ and $\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{b}^l} = 1$

3. 输出层的中间结果 $\boldsymbol{l} = \boldsymbol{L}$

- $\delta^L = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^L} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L}\frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} = (\boldsymbol{a}^L - \boldsymbol{y}) \circ (\boldsymbol{a}^L \circ (1 - \boldsymbol{a}^L))$

4. 其他层的中间结果 $\boldsymbol{l} = \mathbf{1} \ldots \boldsymbol{L} - \mathbf{1}$

- $\delta^l = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{l+1}}\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l} = \delta^{l+1}\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l}$
  - $\boldsymbol{z}^{l+1} = \boldsymbol{a}^l\boldsymbol{W}^{l+1} + \boldsymbol{b}^{l+1}$
  - $\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l} = \boldsymbol{W}^{l+1} \circ f'(\boldsymbol{z}^l) = \boldsymbol{W}^{l+1} \circ (\boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l))$
- $\delta^l = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{l+1}}\frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{z}^l} = \delta^{l+1}{\boldsymbol{W}^{l+1}}^T \circ (\boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l))$

5. 则有梯度

- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l}\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{W}^l} = \delta^l\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{W}^l} = {\boldsymbol{a}^{l-1}}^T\delta^l$
- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l}\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{b}^l} = \delta^l\frac{\partial \boldsymbol{z}^l}{\partial \boldsymbol{b}^l} = \delta^l$

6. 更新参数

$$\boldsymbol{W}^l := \boldsymbol{W}^l - \alpha\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^l} \qquad \boldsymbol{b}^l := \boldsymbol{b}^l - \alpha\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^l}$$

误差反向传播

- 梯度消失（Gradient Vanish）问题
- $$\delta^l = \frac{\partial \mathcal{L}}{\partial z^l} = \frac{\partial \mathcal{L}}{\partial z^{l+1}} \frac{\partial z^{l+1}}{\partial z^l} = \delta^{l+1} {\boldsymbol{W}^{l+1}}^T \circ (\boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l))$$

刚刚的例子中，中间结果δ中有一项$\left(\boldsymbol{a}^l \circ (1 - \boldsymbol{a}^l)\right)$，当激活输出$\boldsymbol{a}$接近0或者1时，中间结果$\delta$会变得很小。由于$\delta^l$又跟$\delta^{l+1}$有关，当反向传播时$\delta$比较小的话，会使得$\delta$ 越传播越小，使得靠近输入层的参数无法被更新，影响网络训练。

- 解决方法 1: 用 ReLU 代替 Sigmoid 函数（常见的方法）
- 解决方法 2: 逐层训练法（已经很少使用了）
- …

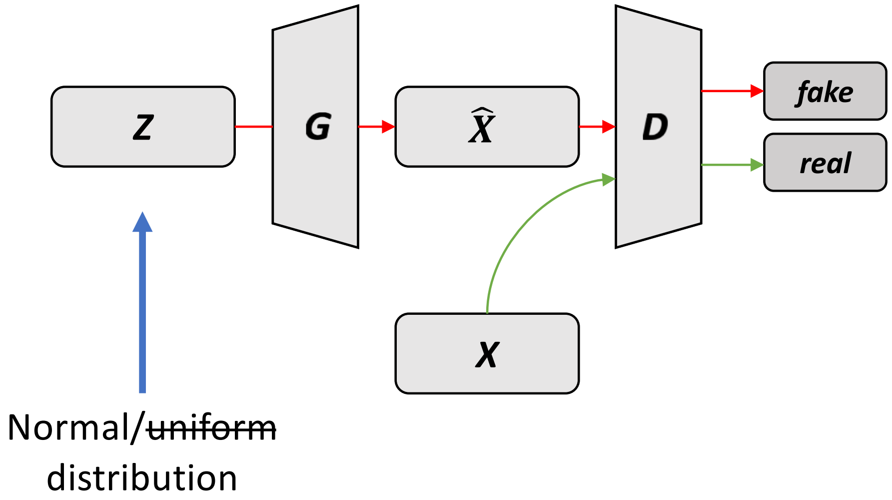
卷积神经网络

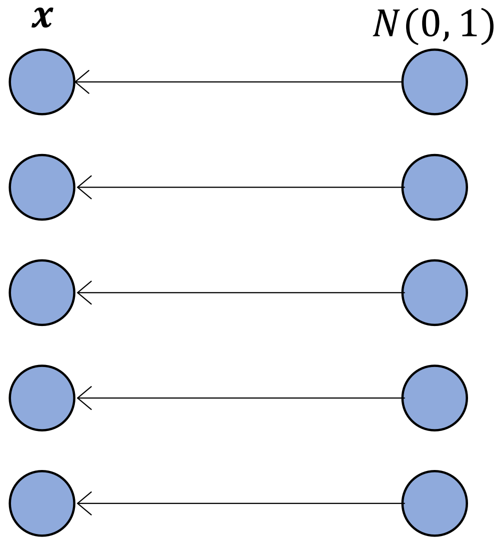
- 卷积算法 Convolutional Algorithm
- 池化算法 Pooling Algorithm
- 分层表示学习 Hierarchical Representation Learning
- 卷积神经网络结构 Convolutional Architectures
- 转置卷积（反卷积） Transposed Convolutional Algorithm

- 引入：生成式模型 Generative Models
- 朴素GAN
- 有条件GAN
- 对抗损失函数 VS 均方差 Adversarial Loss vs. MSE
- GAN面临的挑战 Challenges of GAN

朴素GAN

- 朴素GAN





Unidirectional Mapping

GAN: map a distribution to another distribution

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

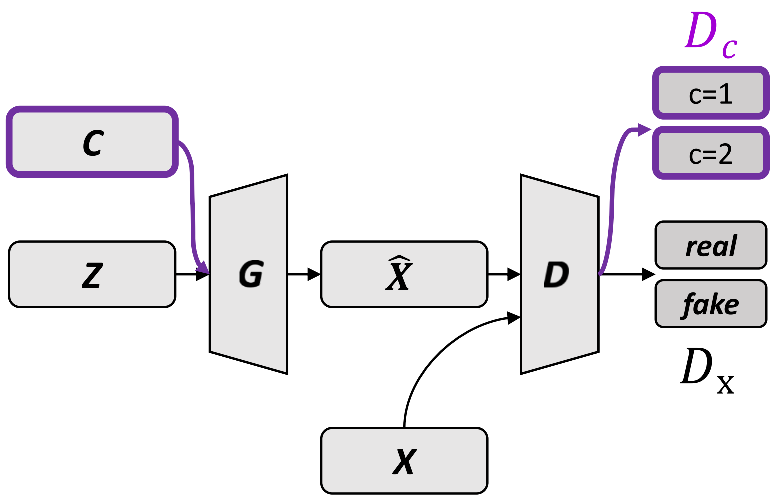
$$\mathcal{L}_D = -\boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})]} - \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))]}$$

$$\mathcal{L}_G = -\boxed{\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(G(\boldsymbol{z}))]}$$

对抗

有条件GAN

- 一个简单的例子：辅助分类器生成对抗网络



monarch butterfly goldfinch daisy

$$\mathcal{L}_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_{\mathrm{x}}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z, c\sim p_c}[\log(1 - D_{\mathrm{x}}(G(\boldsymbol{z},\boldsymbol{c}))] - \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_c(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z, c\sim p_c}[\log(1 - Dc(G(\boldsymbol{z},\boldsymbol{c}))]$$

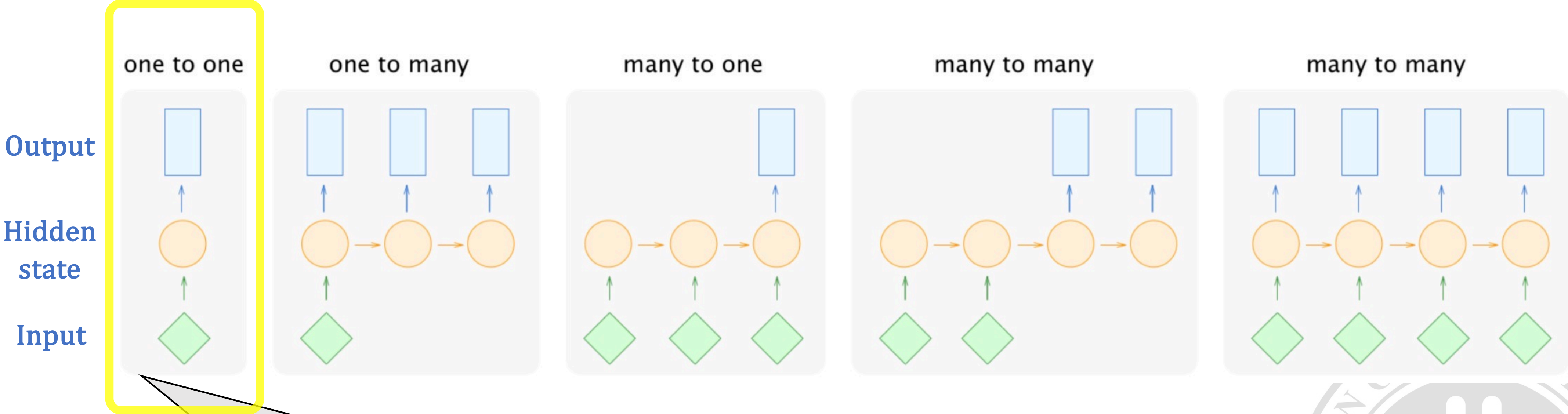
$$\mathcal{L}_G = -\mathbb{E}_{\boldsymbol{z}\sim p_z, c\sim p_c}\left[\log D_{\mathrm{x}}\left(G(\boldsymbol{z},\boldsymbol{c})\right)\right] - \mathbb{E}_{\boldsymbol{z}\sim p_z, c\sim p_c}\left[\log D_c\left(G(\boldsymbol{z},\boldsymbol{c})\right)\right]$$

循环神经网络

- 词的表示 Word Representation
- 序列数据 Sequential Data
- 朴素循环神经网络 Vanilla Recurrent Neural Network
- 长短期记忆网络 LSTM Long Short-Term Memory
- 序列生成模型 RNNs are Generative Models
- 时间序列应用 Time-series Applications

（以及后续的Transformer）

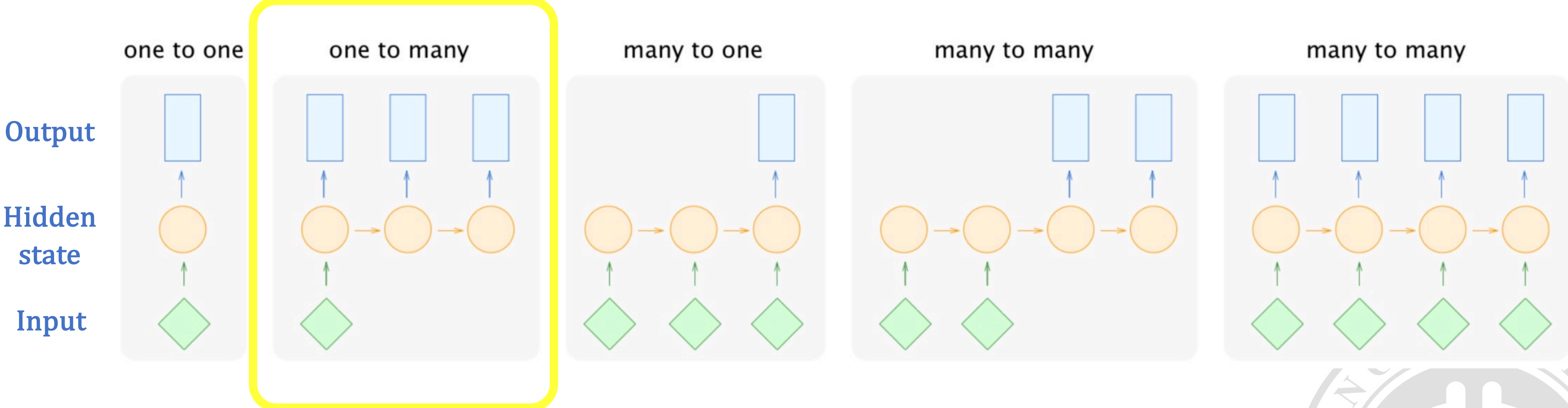
序列数据



Feedforward Neural Networks
前向神经网络

非时间序列问题
比如：图像分类、目标检测

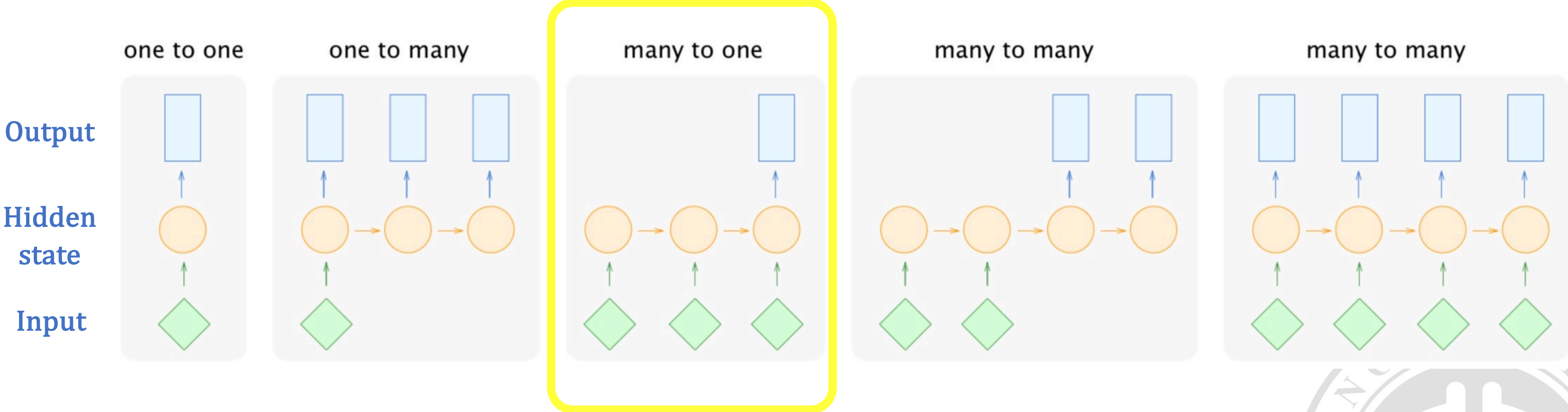
序列数据



输入一个数据，输出多个数据

图片描述：输入一张图片，生成一句话的描述

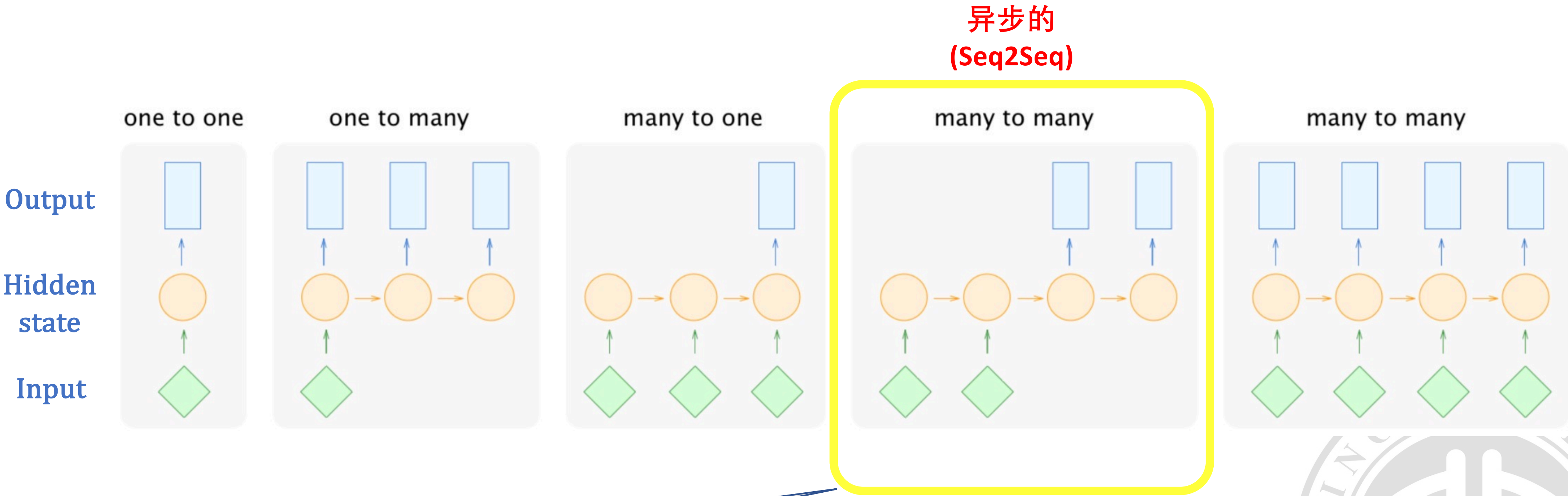
序列数据



输入多个数据，输出一个数据

情感分类任务：输入一个有序的句子，输出表示幸福概率的数值。

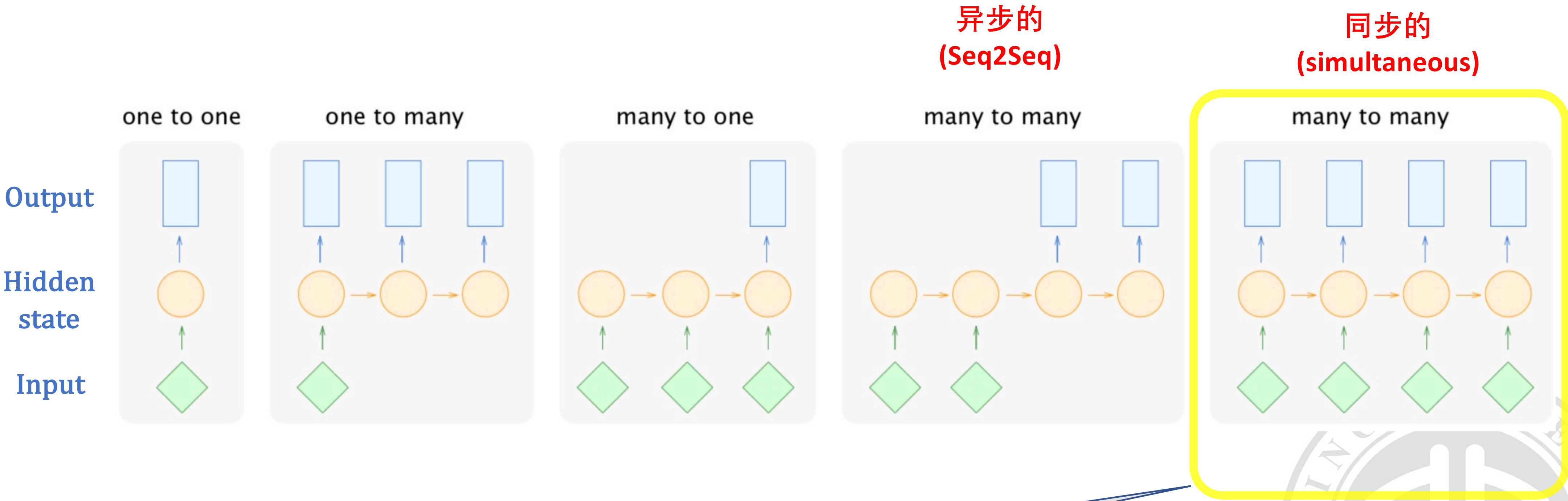
序列数据



多数据输入和多数据输出

语言翻译：在开始生成翻译句子之前，将整个句子输入到模型中。

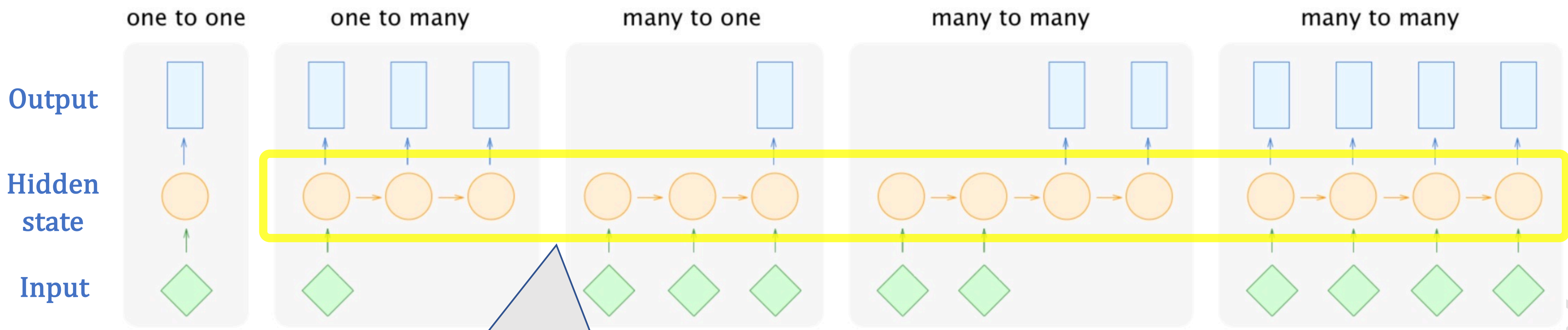
序列数据



多个数据输入和多个数据输出

天气预测：在每个时间步（time-step）输入信息到模型中，并输出预测的天气状况。

序列数据



循环神经网络（Recurrent Neural Nets）

存储和处理时序信息。

北京大學

考试加油