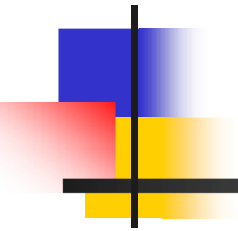


词向量与深度神经网络

主要内容

- 词向量
- 深度神经网络

词向量表示

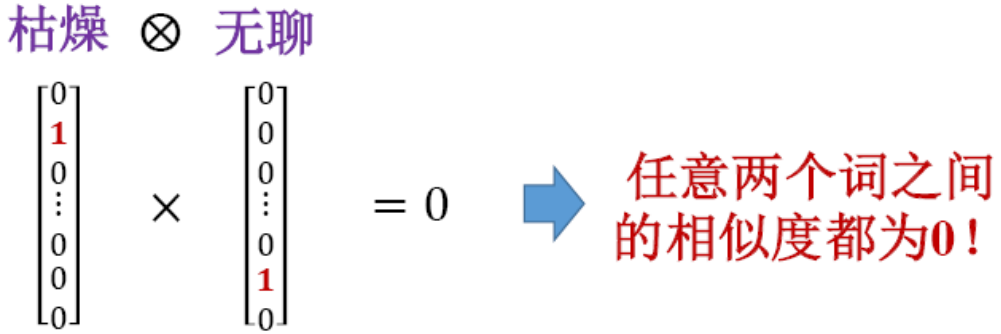
词向量表示的方法主要有 2 类：

独热编码 | one-hot representation
词嵌入　　| word embedding

独热编码

- 假如文本中一共出现了4个词：猫、狗、牛、羊。向量里每一个位置都代表一个词。所以用 one-hot 来表示就是：

  猫：［1，0，0，0］
  狗：［0，1，0，0］
  牛：［0，0，1，0］
  羊：［0，0，0，1］



- One-hot 的缺点：

  无法表达词语之间的关系；

  这种过于稀疏的向量，向量可能会非常长。其中99%以上都是 0；

词嵌入

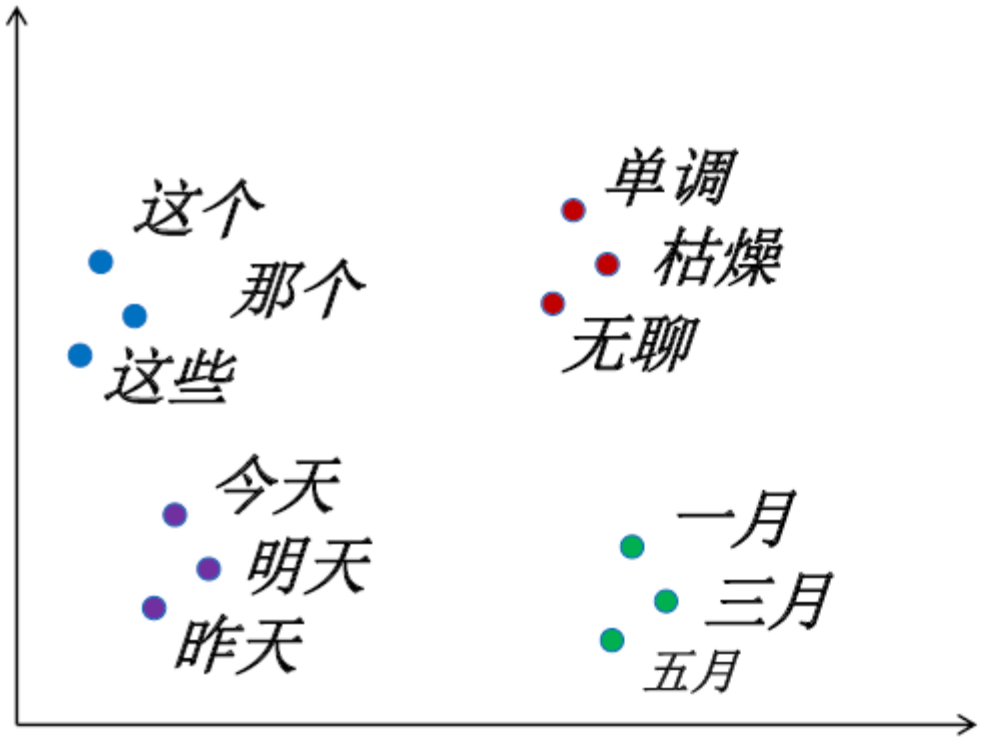
- 词嵌入 | word embedding

可以将每个词通过一个低维向量来表达，不像 one-hot 那么长。

语意相似的词在向量空间上也会比较相近。

可计算性很强，可以用在不同的任务中。

词嵌入



低维、稠密的连续实数空间

枯燥 $\begin{bmatrix} 0.24 \\ 0.15 \\ 0.42 \\ 0.51 \\ 0.21 \end{bmatrix}$ 无聊 $\begin{bmatrix} 0.25 \\ 0.12 \\ 0.39 \\ 0.46 \\ 0.26 \end{bmatrix}$

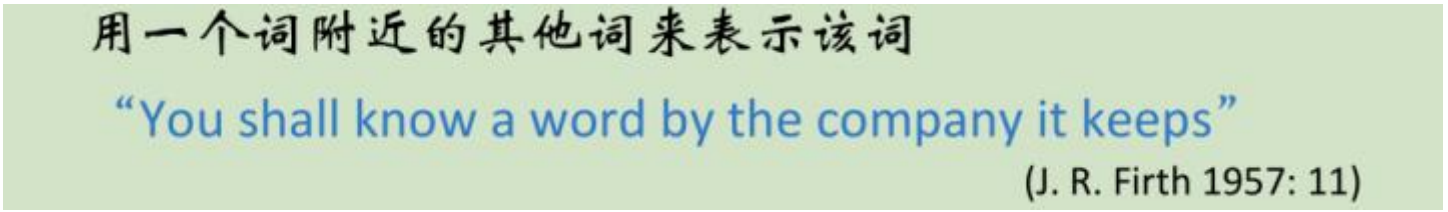
Word2Vec
Word embedding

Word2vec

- Word2vec 是 Word Embedding 的方法之一。它是2013 年由谷歌提出的一套词嵌入方法。
- 这种算法有2种训练模式：

  CBOW- 通过上下文来预测当前词

  Skip-gram- 通过当前词来预测上下文



Word2vec

- 一个词向量的简单实例

Word vectors	Dimensions			
dog	-0.4	0.37	0.02	-0.34
cat	-0.15	-0.02	-0.23	-0.23
lion	0.19	-0.4	0.35	-0.48
tiger	-0.08	0.31	0.56	0.07
elephant	-0.04	-0.09	0.11	-0.06
cheetah	0.27	-0.28	-0.2	-0.43
monkey	-0.02	-0.67	-0.21	-0.48
rabbit	-0.04	-0.3	-0.18	-0.47
mouse	0.09	-0.46	-0.35	-0.24
rat	0.21	-0.48	-0.56	-0.37

Word2vec

- CBOW(Continuous Bag-of-Words Model)

假设文本如下： the florid prose of **the** nineteenth century.

想象有个滑动窗口，红色的词是关键词，两边用相等长m的词串帮助分析。

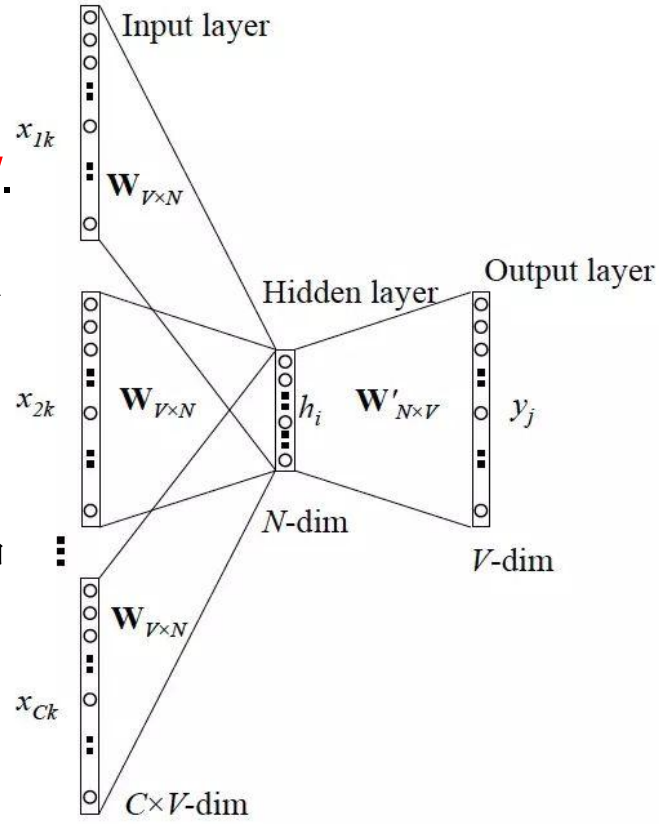
训练目标是：最大化在给定前后4个词的情况下输出正确关键词的概率，用公式表示为

$$P(\text{“ the ”} \mid (\text{“ prose ”}, \text{“ of ”}, \text{“ nineteenth ”}, \text{“ century ”}))$$

可理解为：给定多个上下文词，预测缺失词出现的概率

Word2vec

- CBOW(Continuous Bag-of-Words Model)
  - 输入层：上下文单词的onehot. {假设单词总数为V，上下文单词个数为C}
  - 所有one hot分别乘以共享的输入权重矩阵W. {V*N矩阵，N为自定义值}
  - 所得的向量相加求平均作为隐层向量, size为1*N.
  - 乘以输出权重矩阵W' {N*V}
  - 得到向量 {1*V}， 激活函数处理得到V-dim，概率最大的index所指示的单词为预测出的中间词（target word）
  - 与true label的onehot做比较，误差越小越好
  - 反向传播误差，不断调整W和W '的值

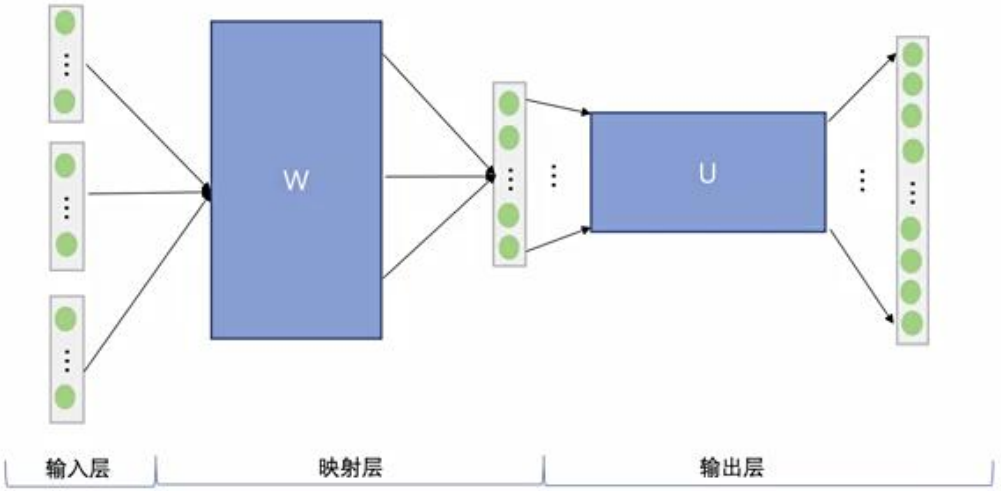


Word2vec

- CBOW(Continuous Bag-of-Words Model)

词典={我，喜欢，到处，旅游}

句子=我　喜欢　到处　旅游



Step1.得到上下文及输出的onenot向量

Word2vec

- CBOW(Continuous Bag-of-Words Model)

输入层的每个单词与矩阵W相乘得到的向量，就是我们想要的词向量（word embedding），这个矩阵也叫做look up table。

也就是说，任何一个单词的one hot乘以这个矩阵W都将得到自己的词向量。

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

W

词向量也就是矩阵W中的某一行参数

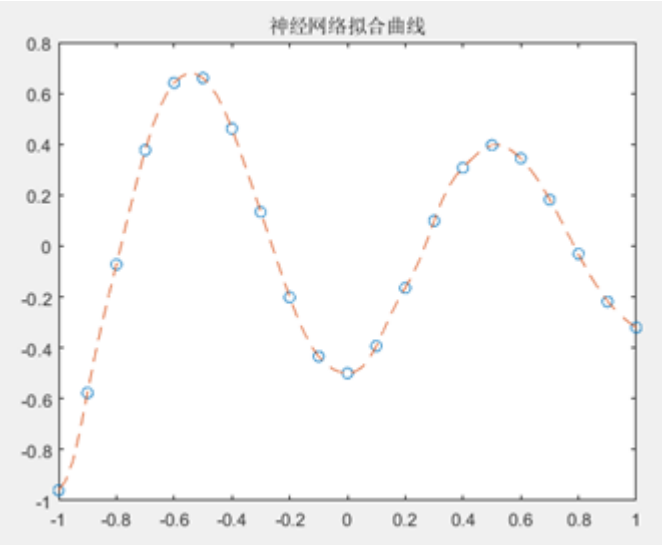
人工神经网络

- 人工神经网络主要由大量的神经元以及它们之间的有向连接构成。因此考虑三方面：
- 神经元的激活规则
  - 主要是指神经元输入到输出之间的映射关系，一般为非线性函数。
- 网络的拓扑结构
  - 不同神经元之间的连接关系。
- 学习算法
  - 通过训练数据来学习神经网络的参数。

人工神经网络

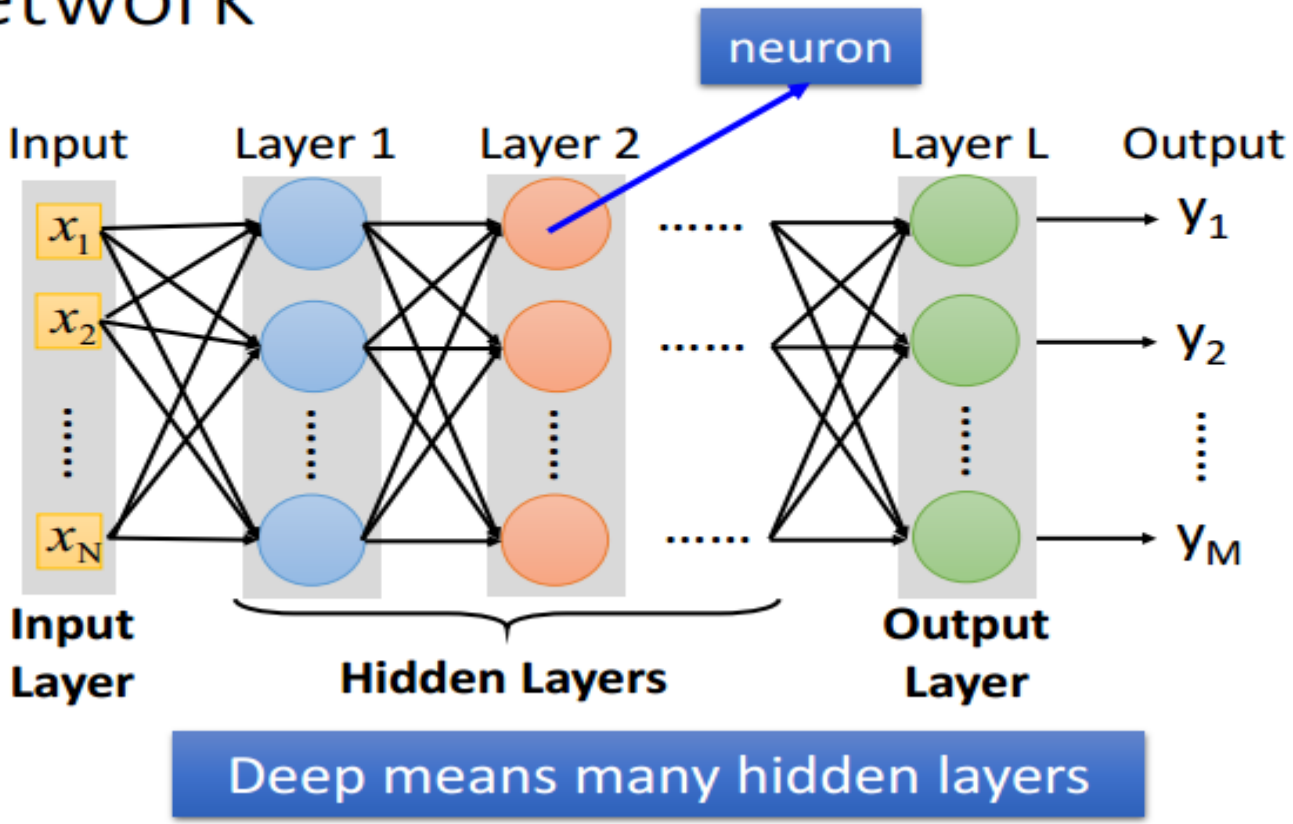
- 对于具有线性输出层和至少一个使用“挤压”性质的激活函数的隐藏层组成的前馈神经网络，只要其隐藏层神经元的数量足够，它可以以任意精度来近似任何从一个定义在实数空间中的有界闭集函数。

一个两层的神经网络可以模拟任何函数。



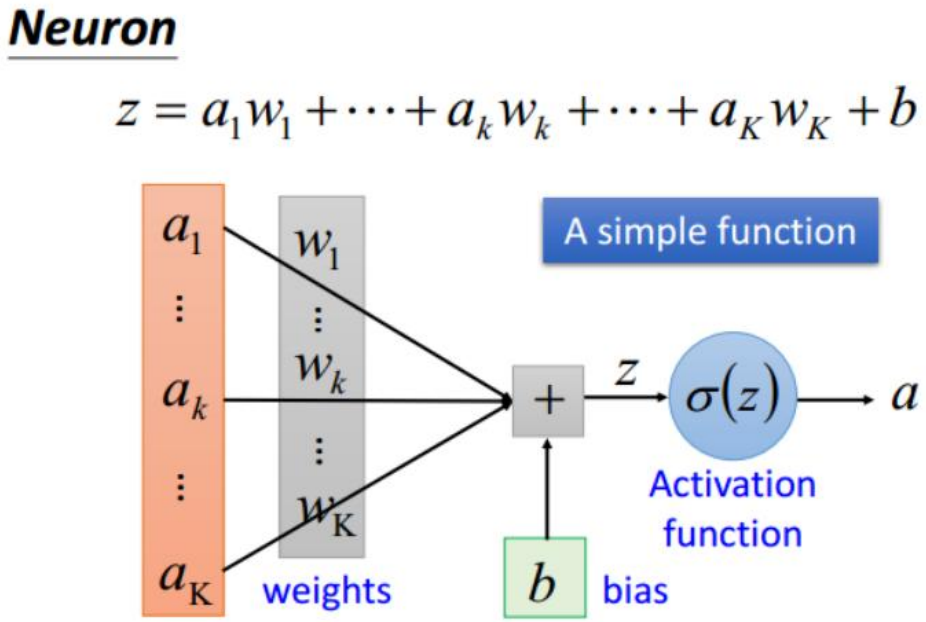
深度神经网络

Fully Connect Feedforward Network



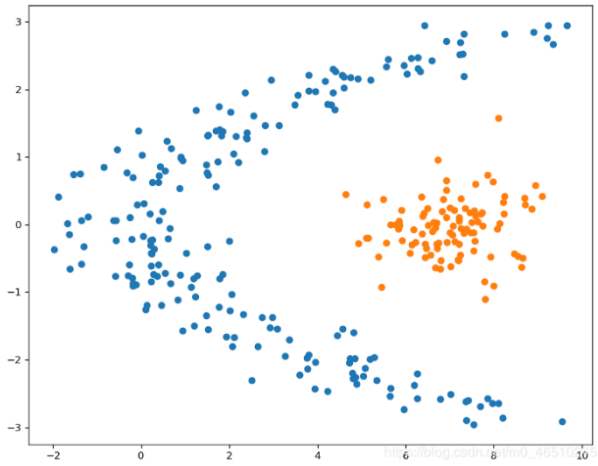
Deep means many hidden layers

神经元及其连接



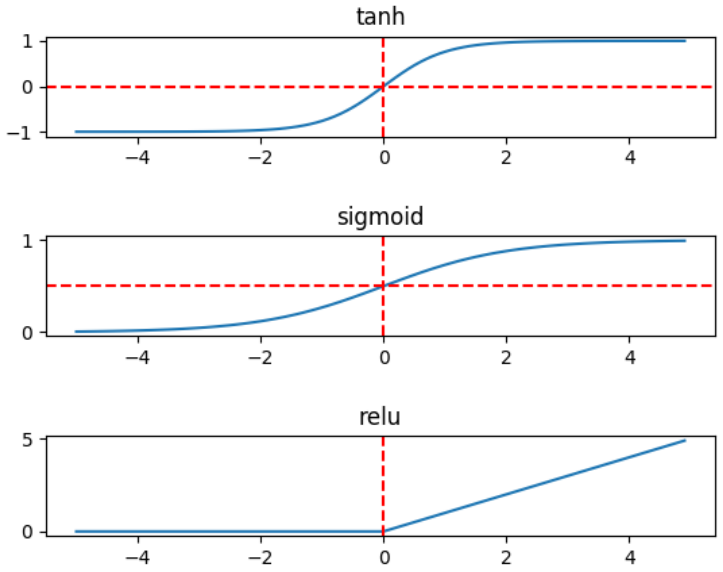
- a表示输入，w表示神经元之间的连接权值。输入端信号经过加权求和、传递后，输出端信号变成a*w+b；
- 函数sigma（激活函数）将矩阵线性运算的结果变为非线性，同时将其值域转换到了0到1之间。

神经元及其连接



图中这个数据集不是线性可分的。

常用的激活函数有三种，分别是Sigmoid、tanh和ReLU。

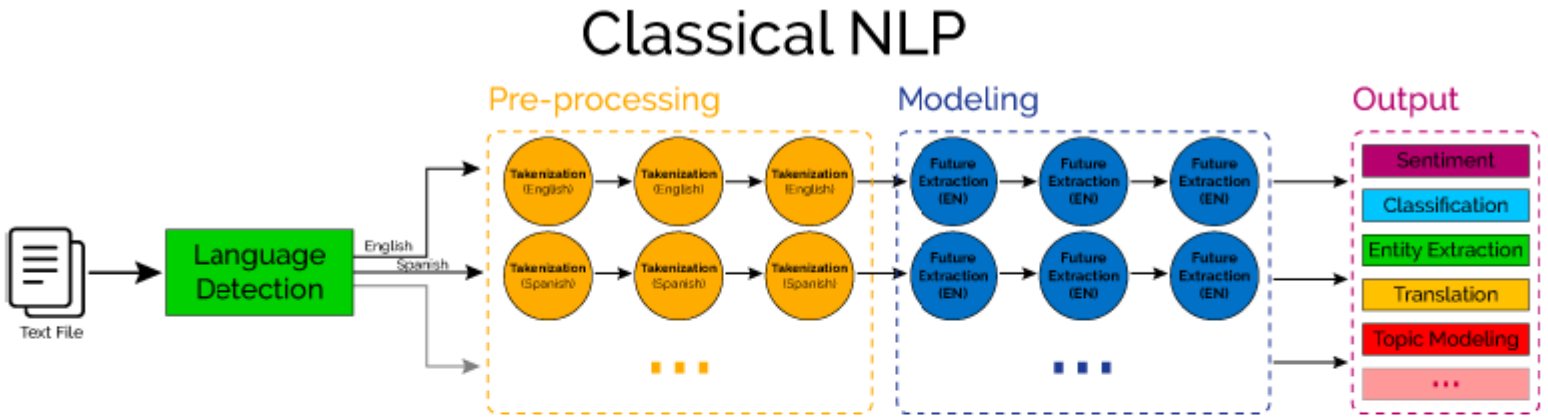


sigmoid: $y = 1/(1 + e^{-x})$

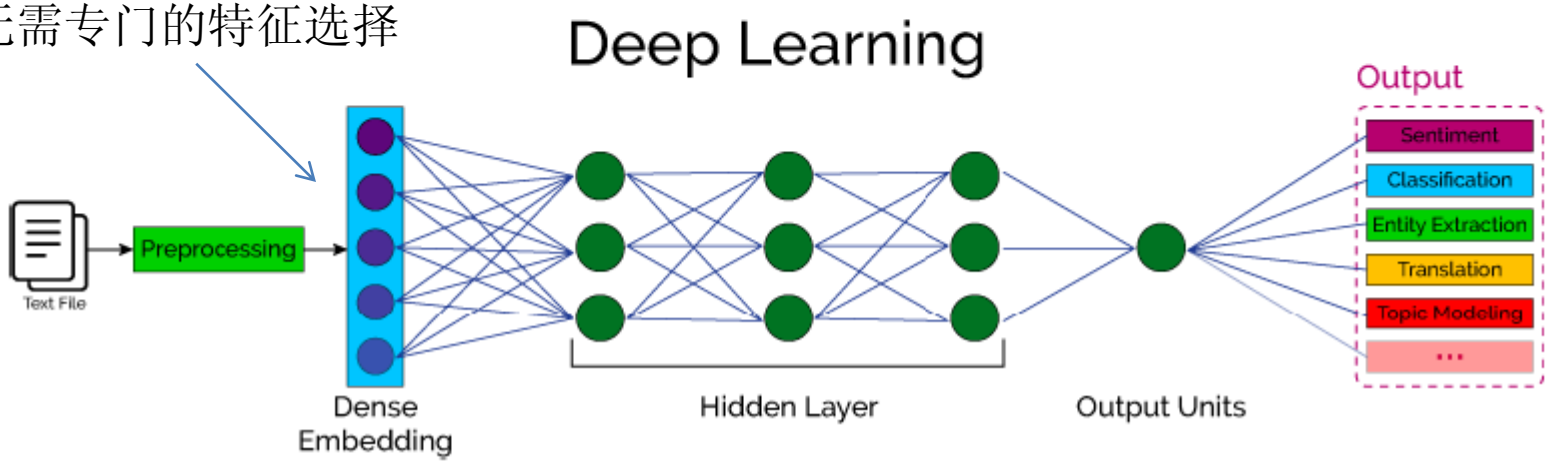
tanh: $y = (e^{x} - e^{-x})/(e^{x} + e^{-x})$

relu: $y = \max(0, x)$

深度学习与传统机器学习



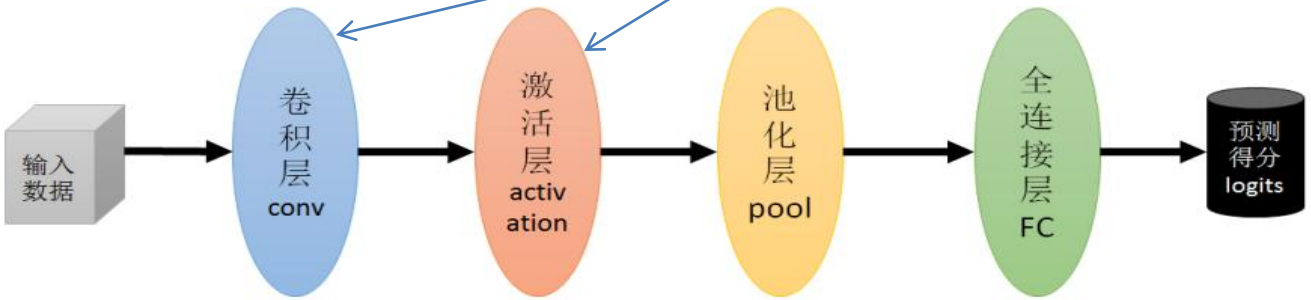
无需专门的特征选择

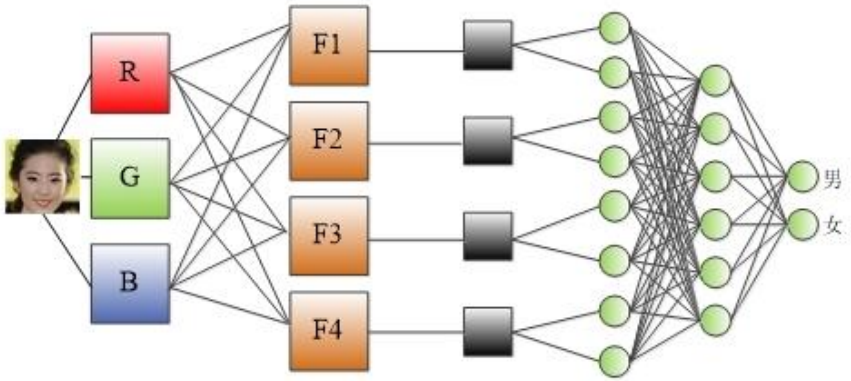


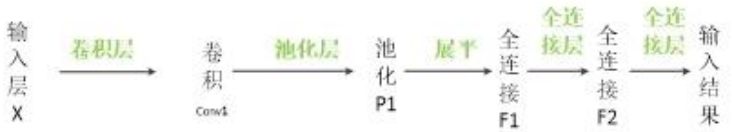
卷积神经网络分析

一个简单的卷积神经网络

可合并为一层







卷积神经网络分析

- 卷积层

原图

垂直
边缘

卷积是图像处理中一种基本方法. 卷积核(过滤器)是一个f*f的矩阵. 通常n取奇数,使得卷积核有中心点.

对图像中每个点取以其为中心的f阶方阵, 将该方阵中各值与卷积核中对应位置的值相乘, 并用它们的和作为结果矩阵中对应点的值.

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

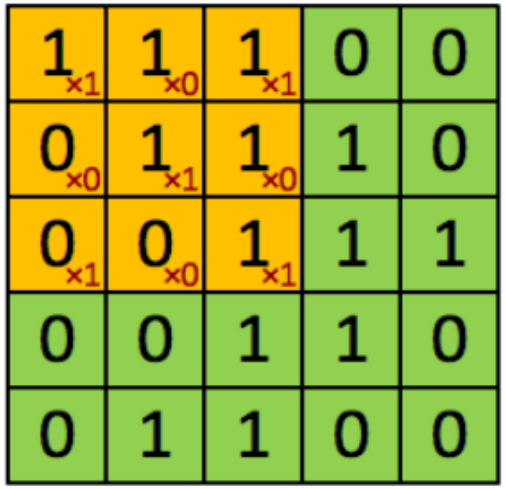
1	0	-1
1	0	-1
1	0	-1

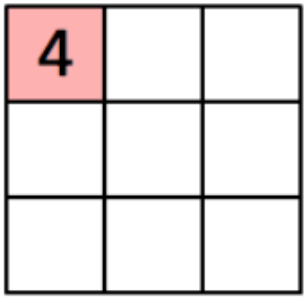
=

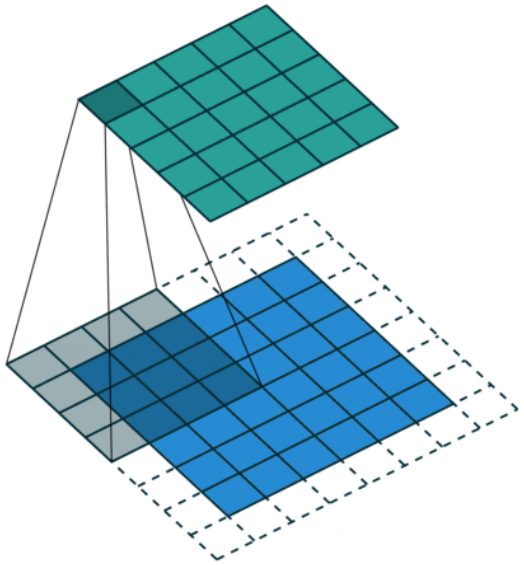
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

卷积神经网络分析

- 卷积运算







$1*1 + 1*0 + 1*1 + 0*0 + 1*1 + 1*0 + 0*1 + 0*0 + 1*1 = 4$

把每次移动的距离称为步幅s

卷积神经网络分析

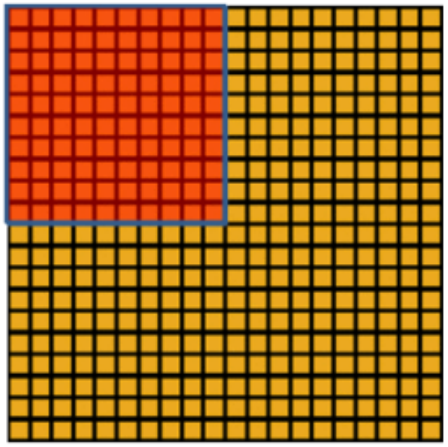
与全连接层相比，卷积层的两个主要优势：

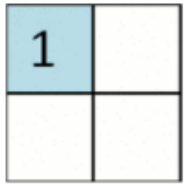
（1）参数共享：　　卷积核在数据上移动

（2）稀疏连接：　　卷积核仅与数据中的某些个部分连接

卷积神经网络分析

- 池化层

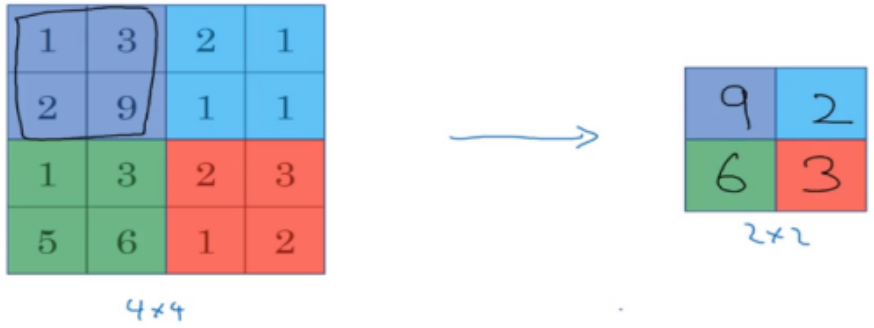




Pooling层对于卷积层进行了个一个***降维***的操作。 Max Pooling是对一个卷积层抽取的特征值取最大的值作为这个层的保留值， 其他值全部抛弃，这个值代表了特征值种***最显著的特征***。他可以减少模型的参数数量，减少过拟合的问题。

Convolved feature

Pooled feature

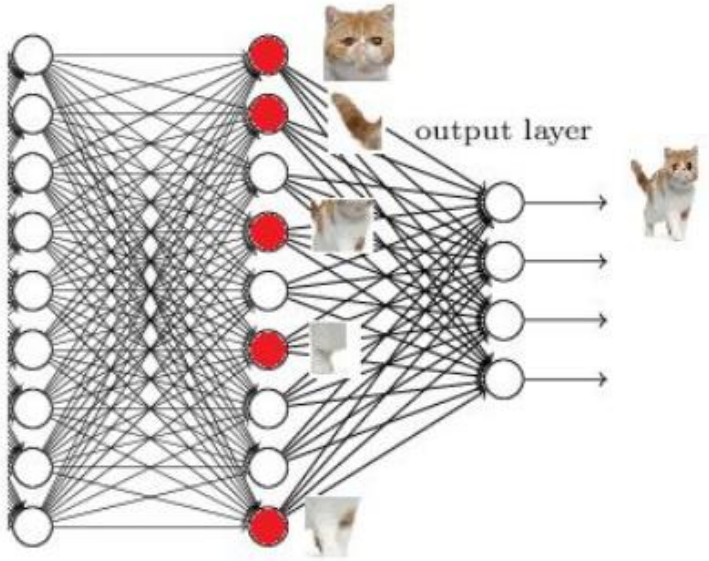


卷积神经网络分析

- 全连接层

全连接层的每一个结点都与上一层的所有结点相连，用来把前边提取到的特征综合起来。由于其全相连的特性，一般全连接层的参数也是最多的。

例如在VGG16中，第一个全连接层FC1有4096个节点，上一层POOL2是25088个节点，则需要4096*25088个权值，需要很大的内存



全连接层

如果全连接层的参数过多，导致无法训练，一个替代方法是不做全连接，使用全局平均值。

卷积神经网络分析

卷积网络在形式上有一点点像咱们正在召开的“人民代表大会制度”。

卷积核：相当于候选人，图像中不同的特征会激活不同的“候选人”（卷积核）。

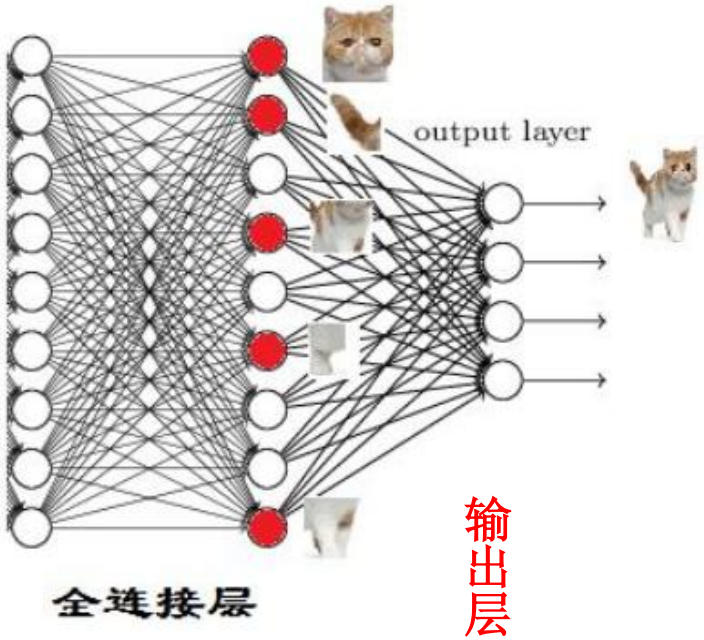
池化层：（仅指最大池化）起着类似于“合票”的作用，不同特征在对不同的“候选人”有着各自的喜好。

全连接层：相当于是“代表普选”。所有被各个区域选出的代表，对最终结果进行“投票”，所有代表都有对最终结果影响的权利。

卷积神经网络分析

- 输出层

为了达到识别的目的，需要定义一个特殊的函数Softmax函数，通过它给每种类别判断一个概率。



softmax函数，又称归一化指数函数。

（1）先通过指数函数exp()将模型原来预测结果转换为非负值。

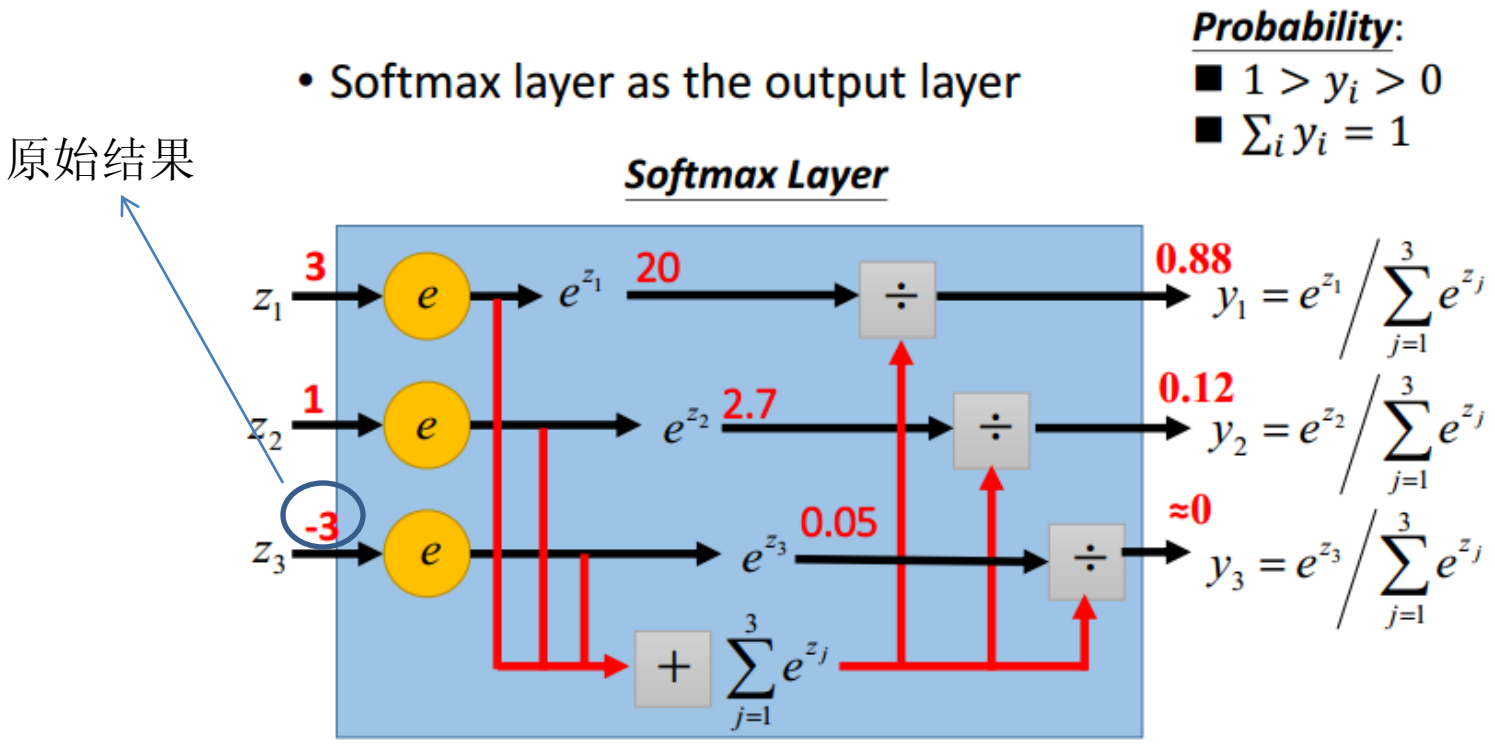
（2）再归一化到（0,1）区间内

$$\mathrm{softmax}(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^{K} \exp(x_i)}$$

卷积神经网络分析

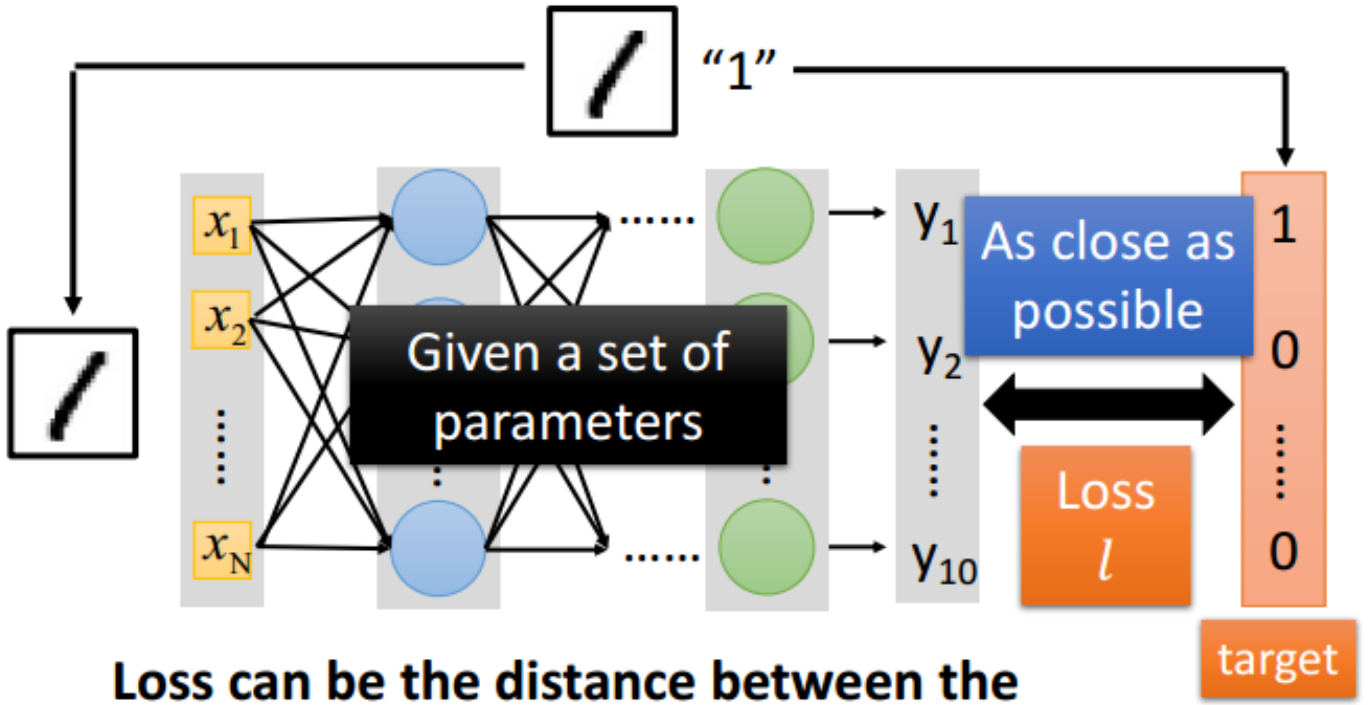
- 输出层

$$\mathrm{softmax}(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^{K}\exp(x_i)}$$



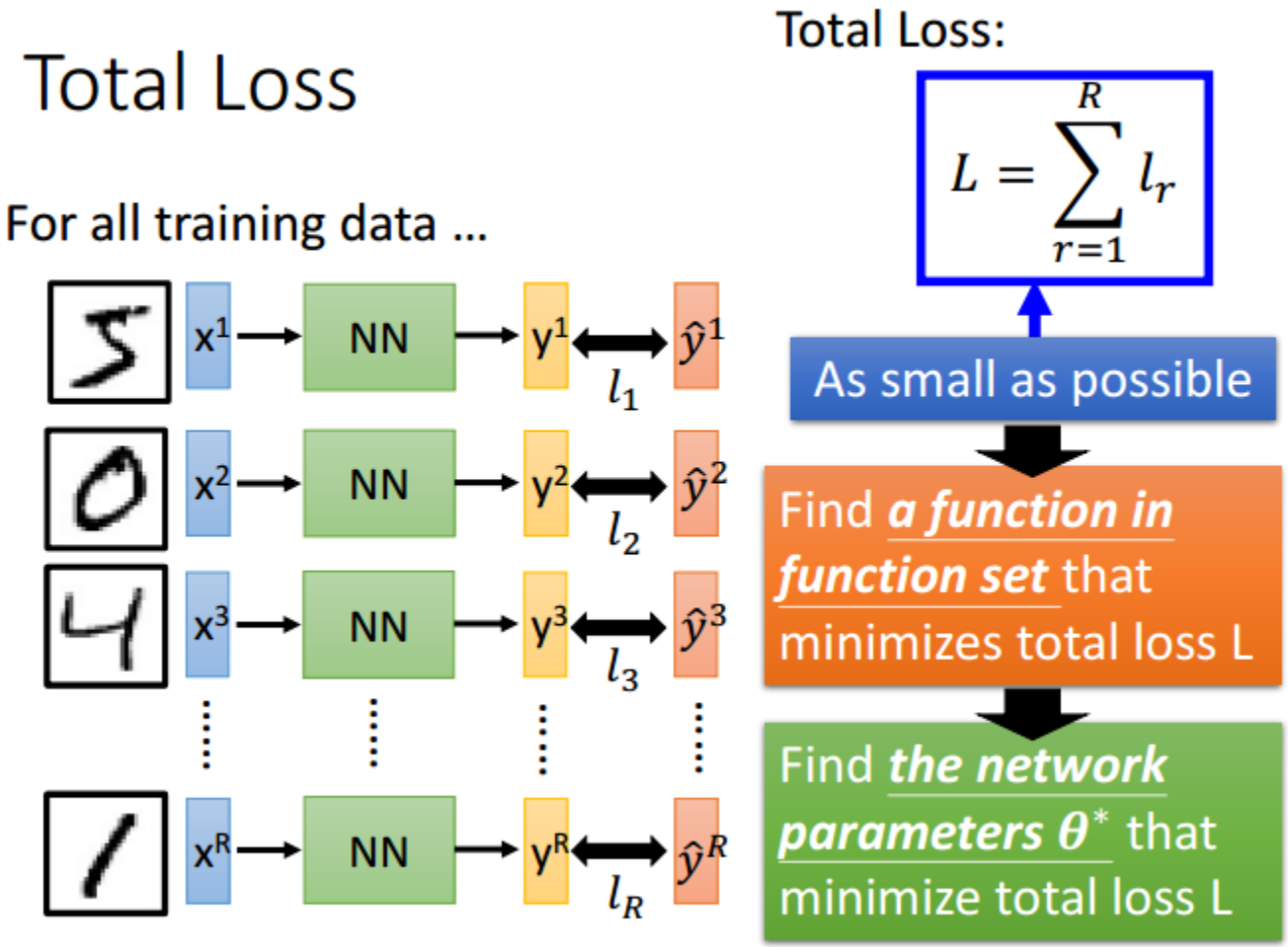
卷积神经网络分析

- 模型训练



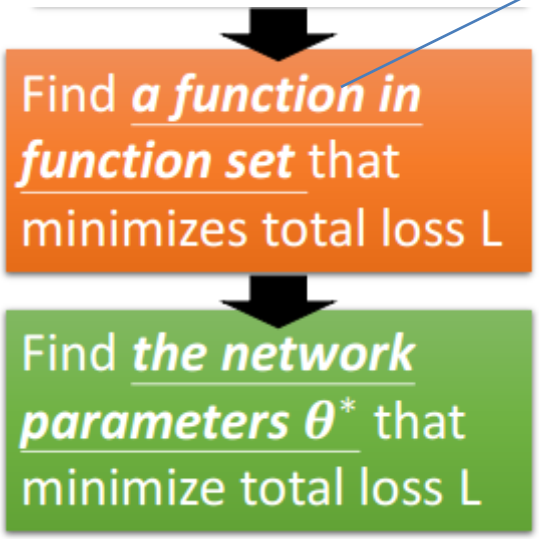
卷积神经网络分析

- 模型训练



卷积神经网络分析

- 模型训练



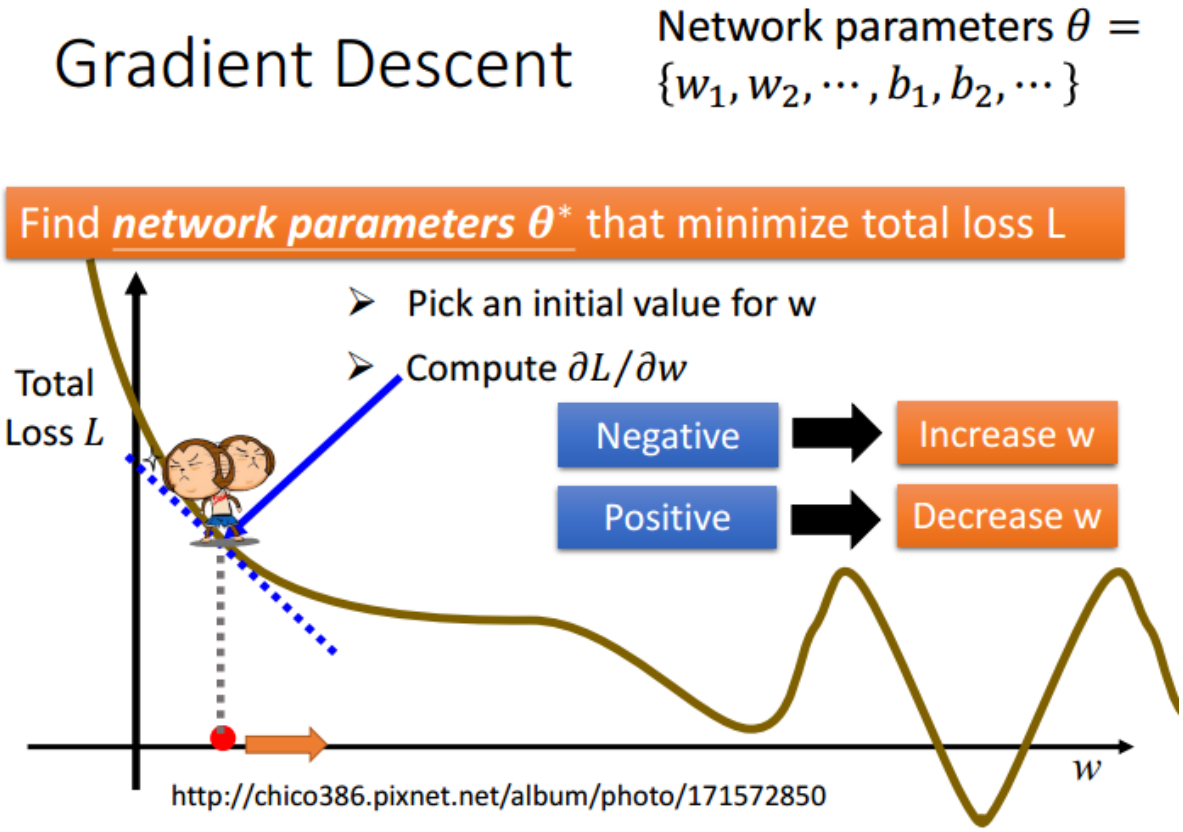
损失函数用来评价模型的**预测值**和**真实值**不一样的程度，如交叉熵损失函数：

$$C = -\frac{1}{n}\sum_{x}[y \ln a + (1-y)\ln(1-a)]$$

x表示样本，y 表示真实的标签，a表示预测的标签

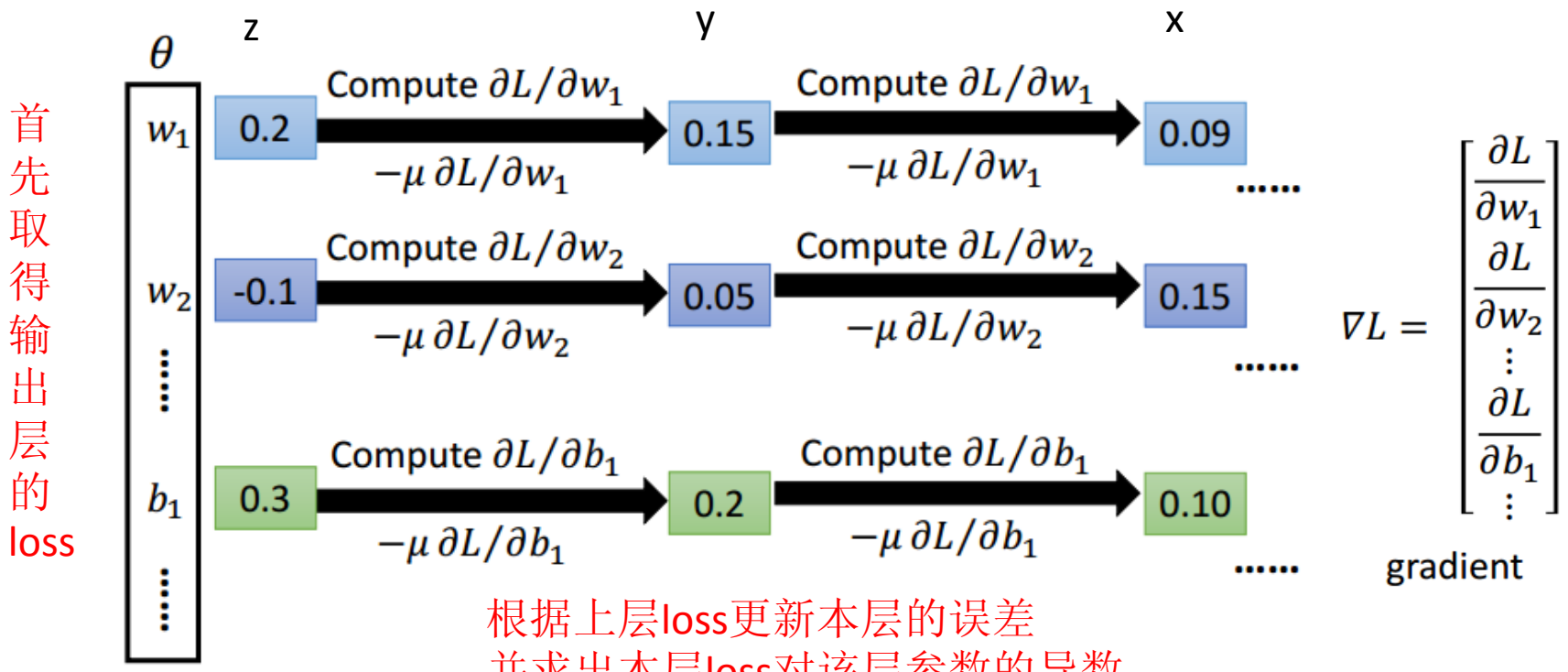
卷积神经网络分析

- 训练　为求得最小的损失函数，常使用梯度下降算法



卷积神经网络分析

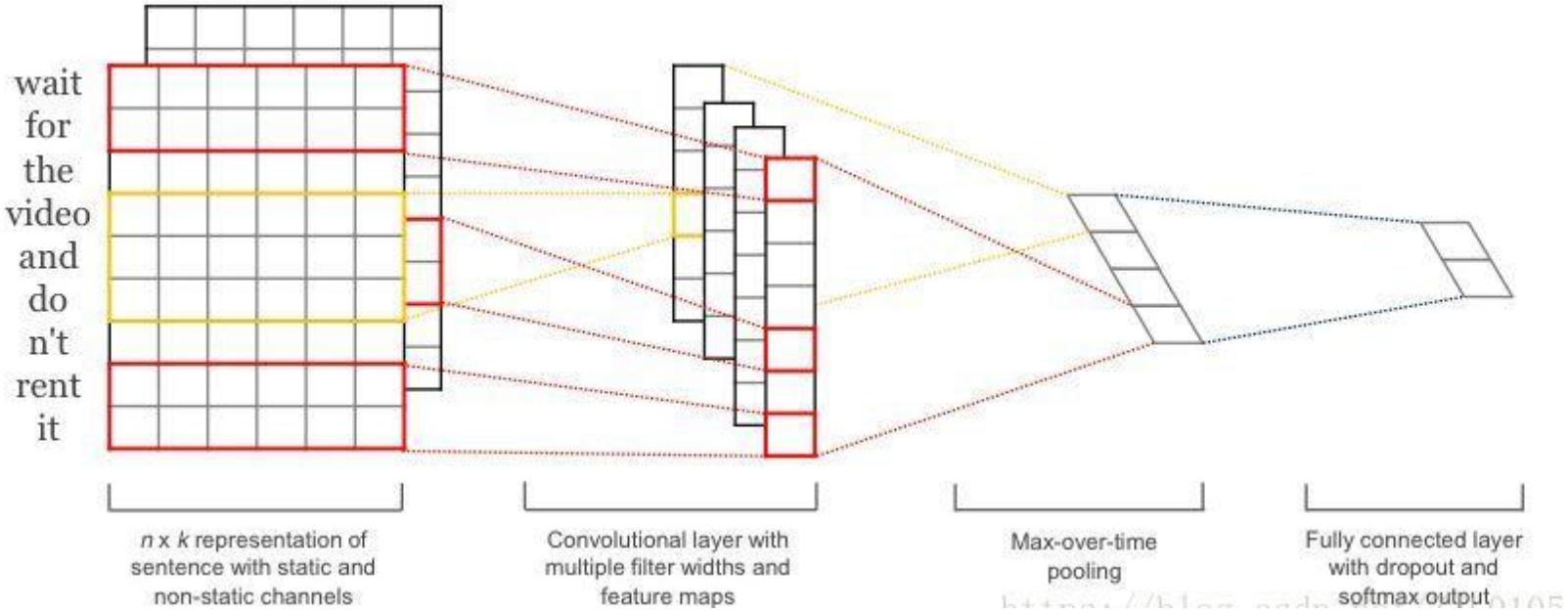
- 考虑到网络的深度，需使用反向传播算法
- 核心是链式法则 $y = g(x) \quad z = h(y)$ $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$



首先取得输出层的loss

根据上层loss更新本层的误差
并求出本层loss对该层参数的导数

NLP中的卷积神经网络



- 将卷积神经网络用于文本分类。
- 输入句子对应的词向量矩阵，经过一层卷积层和一层Max Pooling层，得到句向量表示，再送入到全连接层，最后softmax输出。

NLP中的卷积神经网络

- 优点

  卷积神经网络擅长提取重要的局部特征。

  在文本分类中，可以理解为不同大小的卷积核在提取不同n-gram特征。

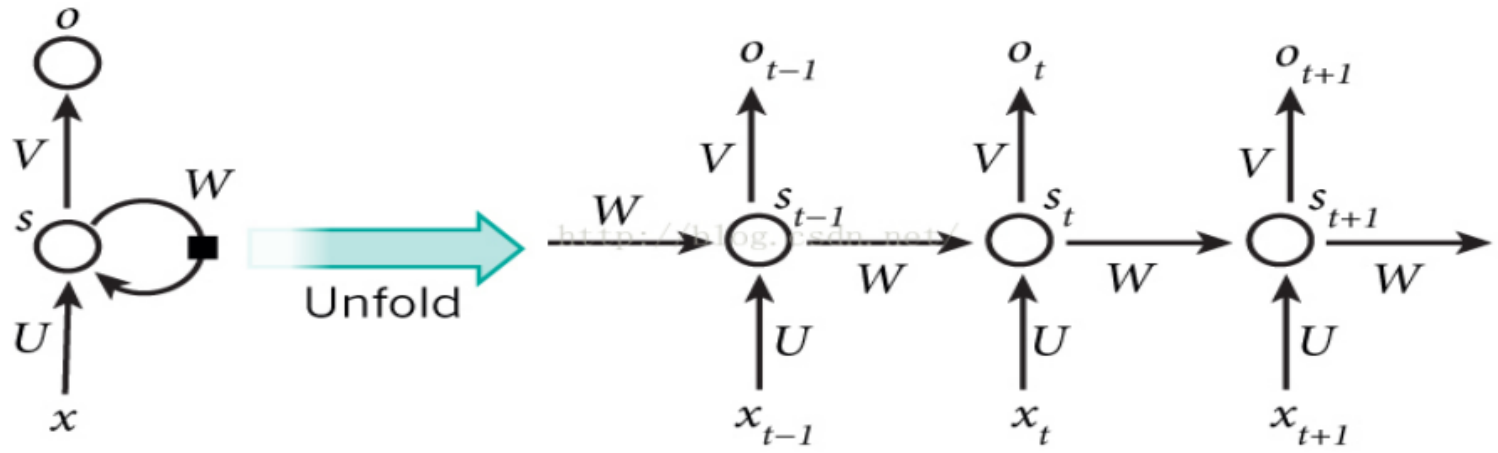
- 缺点

  卷积神经网络无法考虑长距离的依赖信息，且没有考虑词序信息，在有限的窗口下提取句子特征，会损失一些语义信息。

循环神经网络

全连接神经网络和卷积神经网络他们都只能单独的取处理一个个的输入，前一个输入和后一个输入是完全没有关系的。但是，某些任务需要能够更好的处理序列的信息，即前面的输入和后面的输入是有关系的。比如，当我们在理解一句话意思时，孤立的理解这句话的每个词是不够的，我们需要处理这些词连接起来的整个序列；当我们处理视频的时候，我们也不能只单独的去分析每一帧，而要分析这些帧连接起来的整个序列。这时，就需要用到深度学习领域中另一类非常重要神经网络：循环神经网络(Recurrent Neural Network)。

循环神经网络

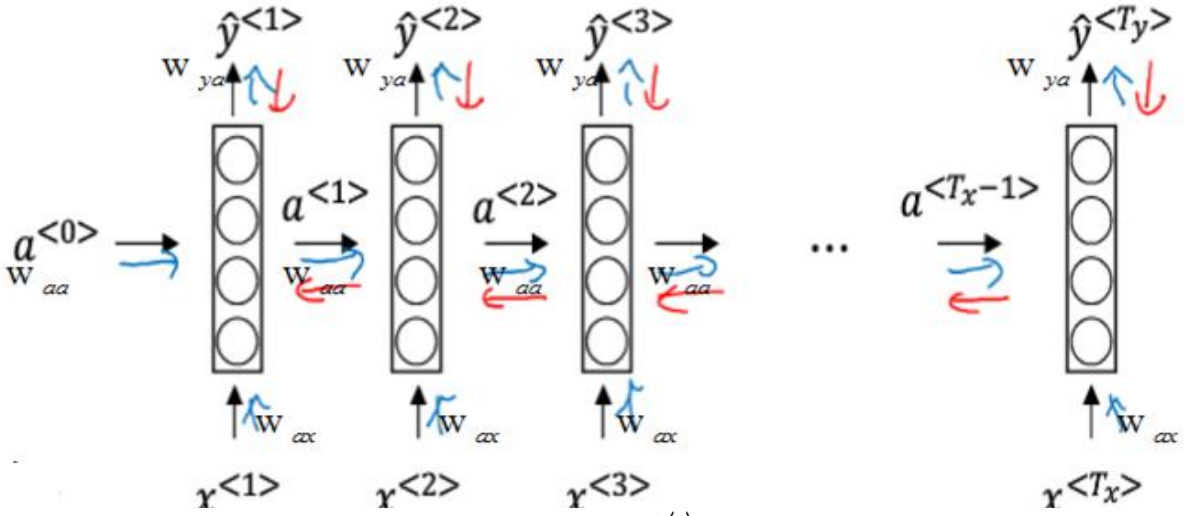


这个网络在t时刻接收到输入Xt之后，隐藏层的值是St，输出值是Ot。关键一点是，St的值不仅仅取决于Xt，还取决于St-1。

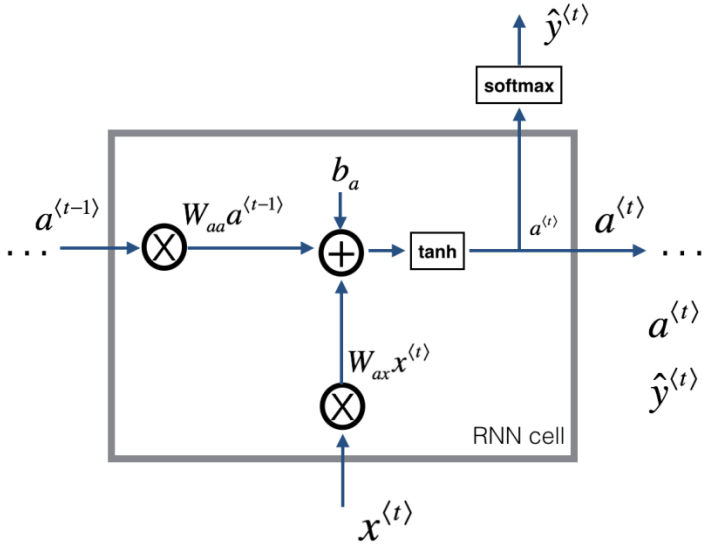
循环神经网络的本质是：像人一样拥有记忆的能力。它的输出依赖于当前的输入和之前的记忆。

循环神经网络

网络展开结构



其中每个神经元结构



$$a^{\langle t\rangle} = \tanh(W_{ax}x^{\langle t\rangle} + W_{aa}a^{\langle t-1\rangle} + b_a)$$

$$\hat{y}^{\langle t\rangle} = soft\max(W_{ya}a^{\langle t\rangle} + b_y)$$

循环神经网络

根据神经元内部结构的不同，分为：

GRU(门控制循单元)
LSTM（长短期记忆）

The ***cat***, which ....., ***was*** full.
The ***cats***, which ....., ***were*** full.

两个单词离得太远了，互相影响太小。

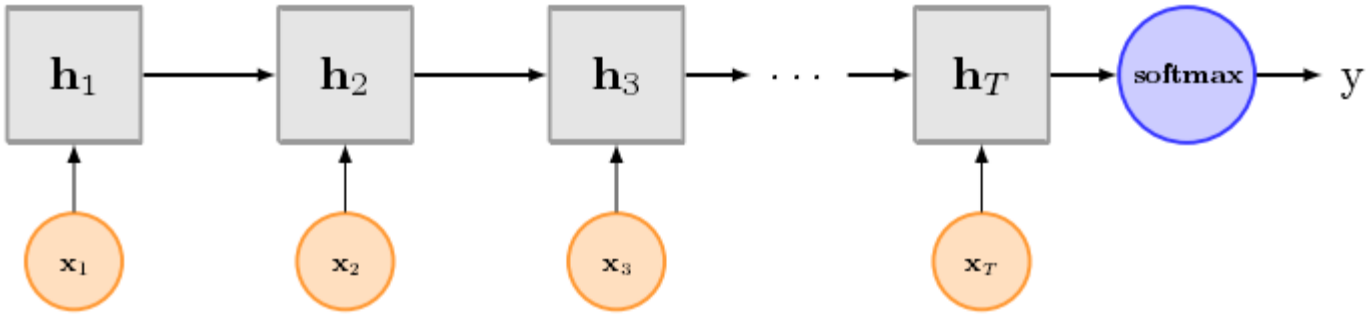
BRNN（双向RNN）

这是**佟德超**。
这是**佟德超市**。

考虑到前面和后面的影响。

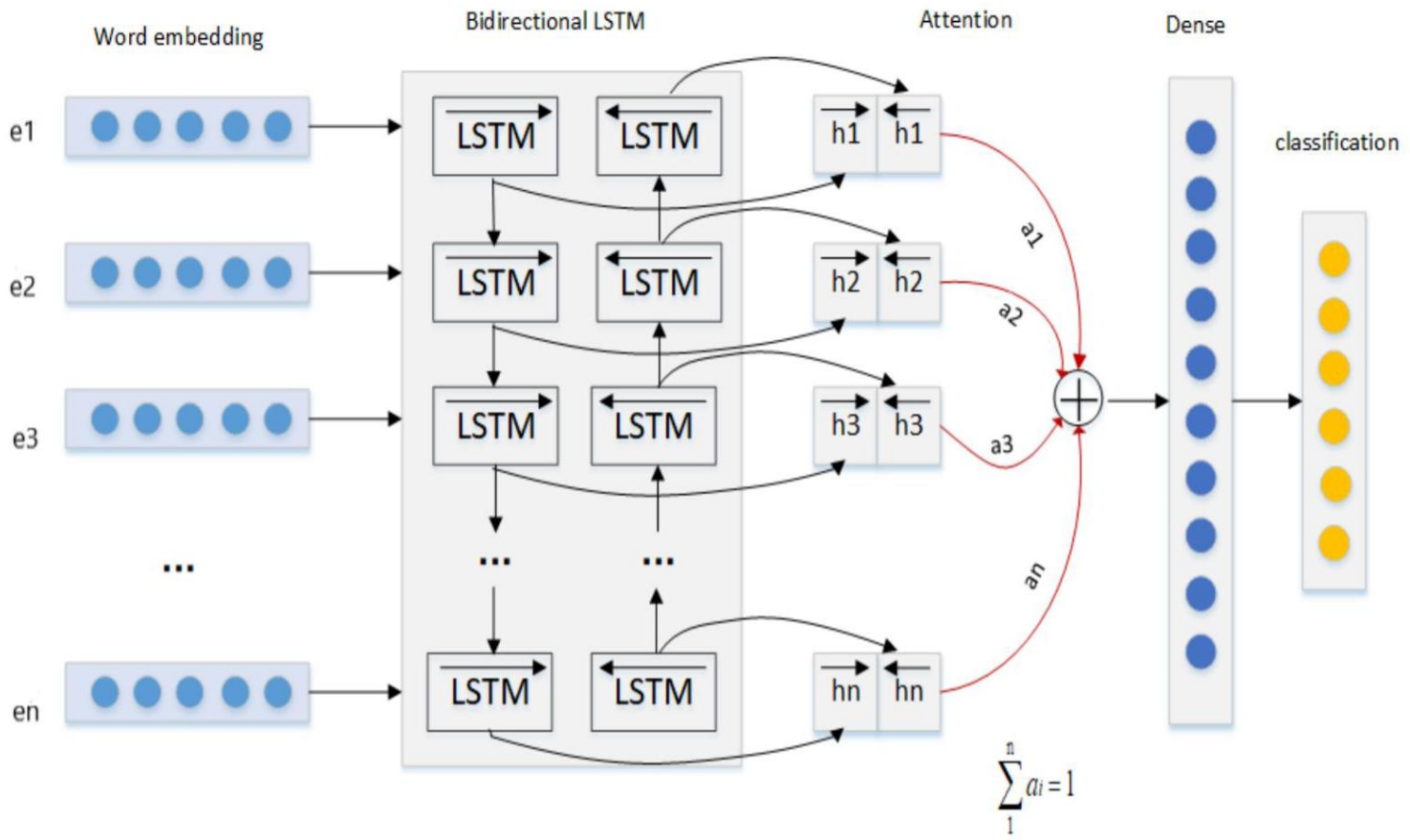
循环神经网络

- 文本分类
  方式1：使用最后的隐变量作为句向量



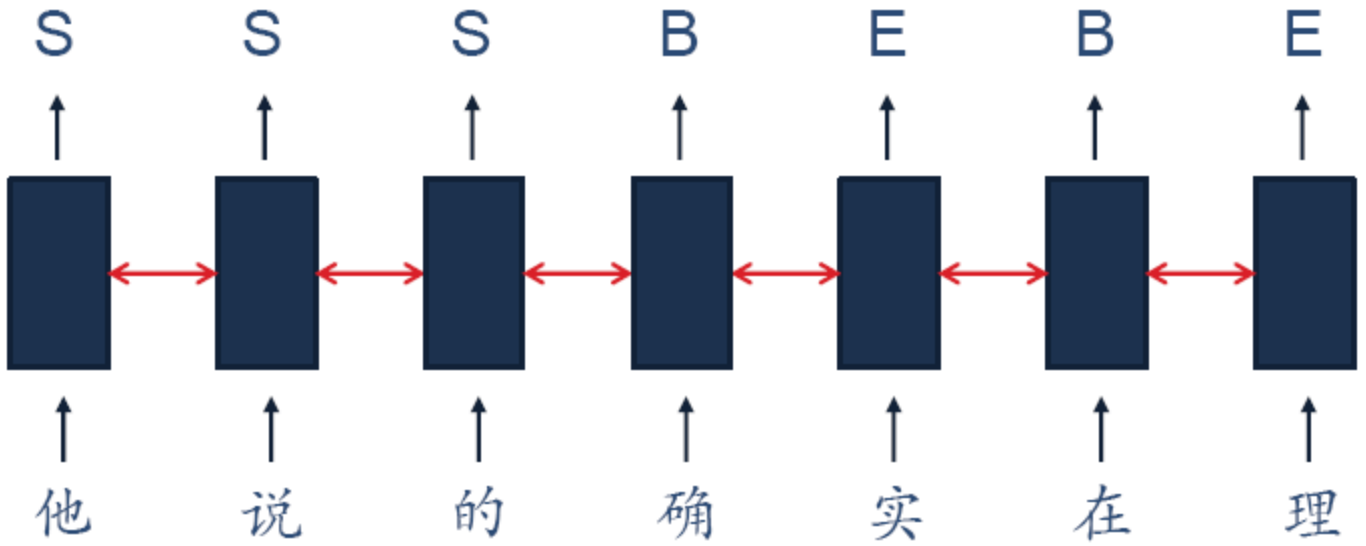
循环神经网络

- 文本分类　　方式2：使用每个词的隐变量拼成句向量



循环神经网络

- 中文分词



主流深度学习框架

目前最主流的是TensorFlow

常见的深度学习框架

主流深度学习框架

```
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.optimizers import SGD

model = Sequential()
model.add(Dense(output_dim=64, input_dim=100))
model.add(Activation("relu"))
model.add(Dense(output_dim=10))
model.add(Activation("softmax"))

model.compile(loss='categorical_crossentropy',
  optimizer='sgd', metrics=['accuracy'])

model.fit(X_train, Y_train, nb_epoch=5, batch_size=32)

loss = model.evaluate(X_test, Y_test, batch_size=32)
```

加入全连接层，将dense换成Conv2D可加入卷积层，换成MaxPooling2D可加入池化层

加入全连接层2

指定模型设置：如使用交叉熵和随机梯度下降优化器

赋予模型训练数据，循环训练5轮，每轮将数据分为大小32的子集

Thanks

谢谢！