

# 深度学习

## 在自然语言处理中的基本应用

曾梓航

mail:raymondtseng0912@gmail.com

## 一、什么叫深度学习

深度学习的概念源于人工神经网络的研究。含多隐层的多层感知器就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。

深度学习是机器学习中一种基于对数据进行表征学习的方法。观测值（例如一幅图像）可以使用多种方式来表示，如每个像素强度值的向量，或者更抽象地表示成一系列边、特定形状的区域等。而使用某些特定的表示方法更容易从实例中学习任务（例如，人脸识别或面部表情识别）。深度学习的好处是用非监督式或半监督式的特征学习和分层特征提取高效算法来替代手工获取特征。

同机器学习方法一样，深度机器学习方法也有监督学习与无监督学习之分，不同的学习框架下建立的学习模型很是不同，例如，卷积神经网络就是一种深度的监督学习下的机器学习模型，而深度置信网就是一种无监督学习下的机器学习模型。

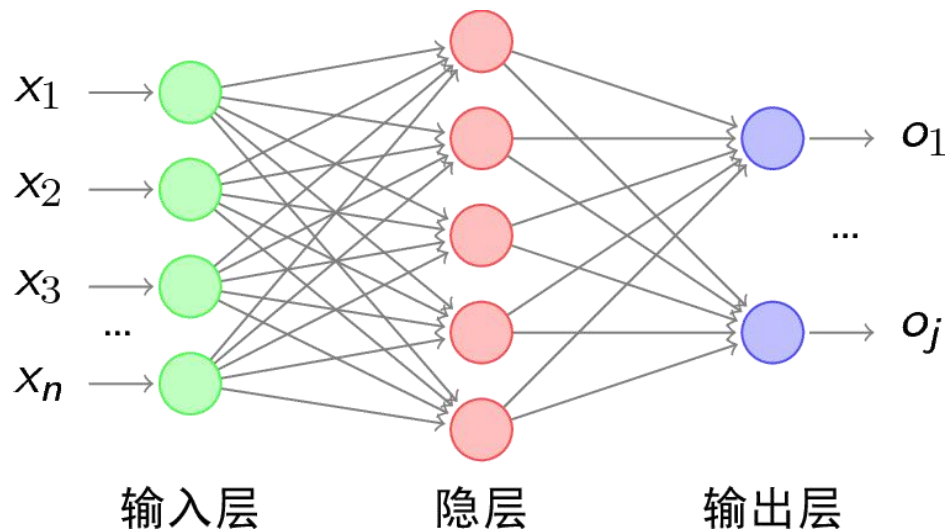
## 二、深度学习在自然语言处理中的基本应用

深度学习在自然语言处理中的应用有很多，如序列标注问题，包括分词，词性标注，命名实体识别等任务；如分类问题，包括文本主题分类，情感分类等。又如更高层次的机器翻译，自动问答等，均可用深度学习来解决。

### 2.1 BP ( Back Propagation ) 神经网络

其基本思想是：学习过程由信号正向传播与误差的反向回传两个部分组成；正向传播时，输入样本从输入层传入，经各隐层依次逐层处理，传向输出层，若输出

层输出与期望不符，则将误差作为调整信号逐层反向回传，对神经元之间的连接权矩阵做出处理，使误差减小。经反复学习，最终使误差减小到可接受的范围。



BP 神经网络结构图

具体步骤如下：

- 1、从训练集中取出某一样本，把信息输入网络中。
- 2、通过各节点间的连接情况正向逐层处理后，得到神经网络的实际输出。
- 3、计算网络实际输出与期望输出的误差。
- 4、将误差逐层反向回传至之前各层，并按一定原则将误差信号加载到连接权值上，使整个神经网络的连接权值向误差减小的方向转化。
- 5、对训练集中每一个输入—输出样本对重复以上步骤，直到整个训练样本集的误差减小到符合要求为止。

## 2.2 word2vec

word2vec 是 Google 于 2013 年开源推出的一个用于获取 word vector 的工具包，它简单、高效，因此引起了很多人的关注。其包含两种结构，分别为 CBOW 和 skip-gram，有两种解法，分别为 hierarchical softmax 和 negative sampling。

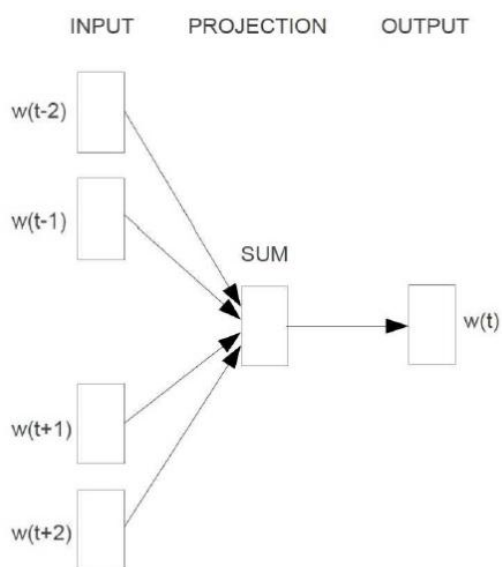


图 8 CBOW 模型

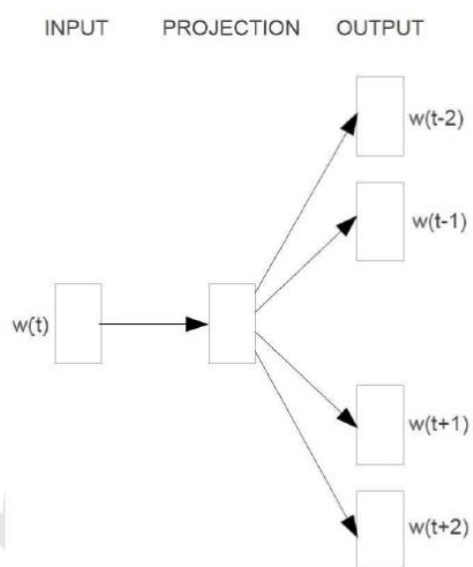
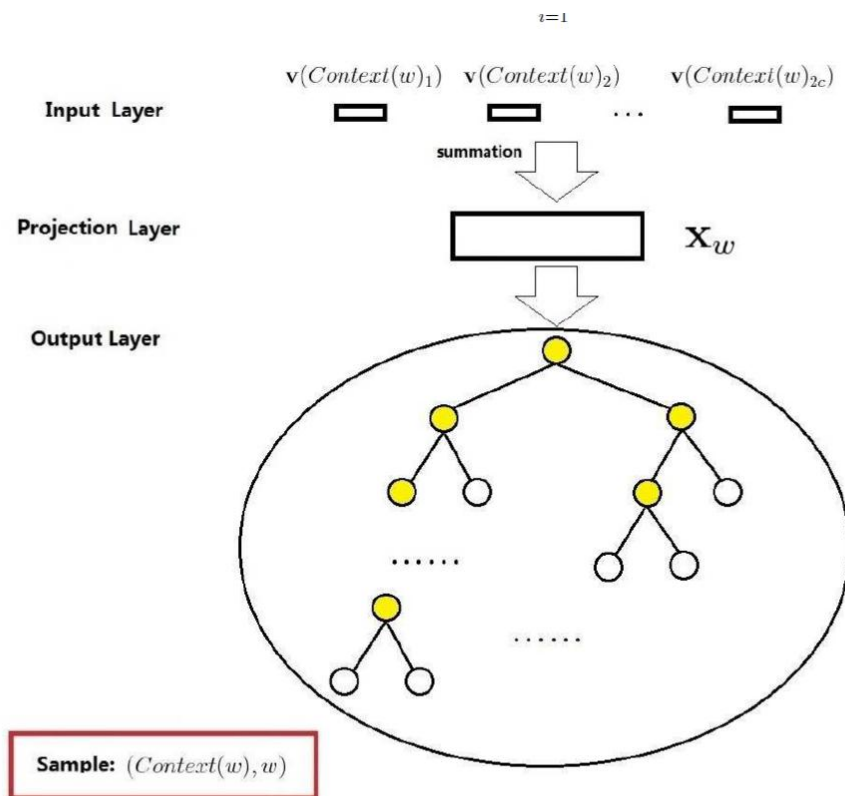
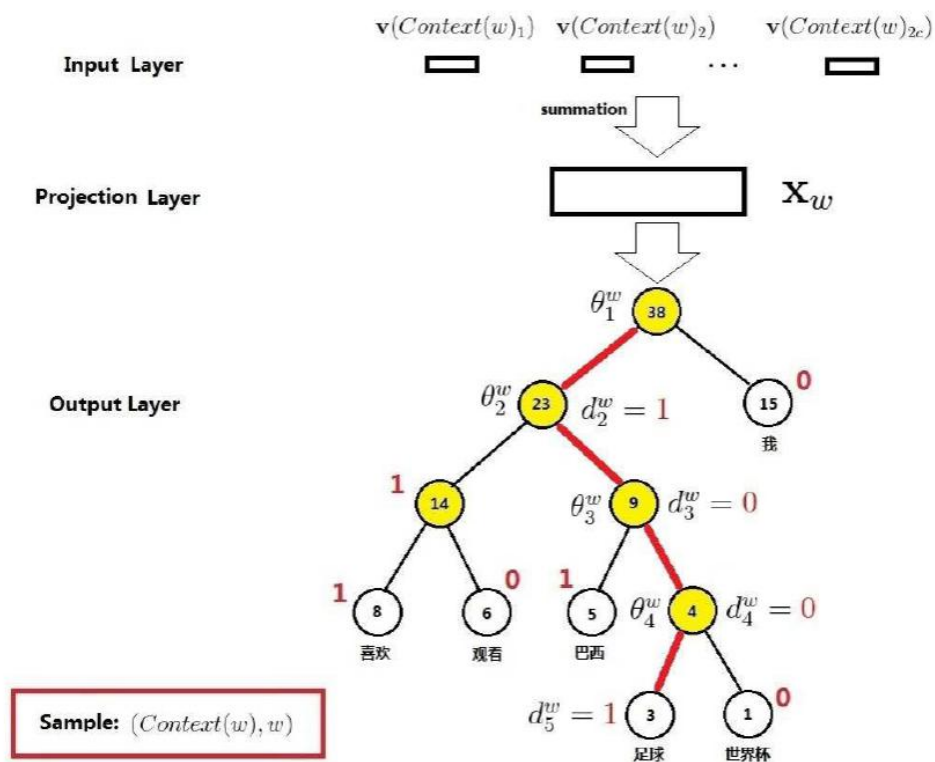


图 9 Skip-gram 模型



CBOW 模型网络结构示意图

Example:



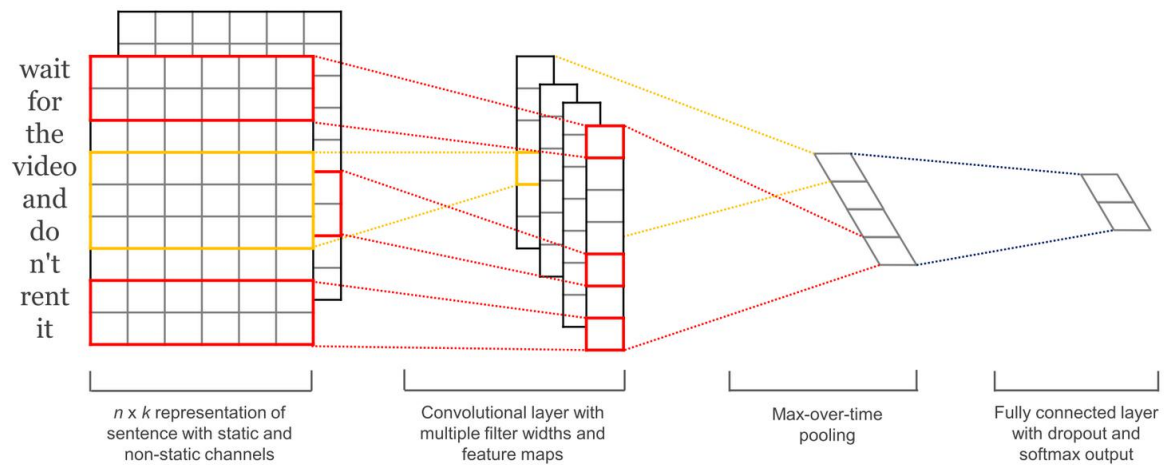
伪代码：

---

```
1.  $\mathbf{e} = \mathbf{0}$ .
2.  $\mathbf{x}_w = \sum_{u \in \text{Context}(w)} \mathbf{v}(u)$ .
3. FOR  $j = 2 : l^w$  DO
    {
        3.1  $q = \sigma(\mathbf{x}_w^\top \theta_{j-1}^w)$ 
        3.2  $g = \eta(1 - d_j^w - q)$ 
        3.3  $\mathbf{e} := \mathbf{e} + g\theta_{j-1}^w$ 
        3.4  $\theta_{j-1}^w := \theta_{j-1}^w + g\mathbf{x}_w$ 
    }
4. FOR  $u \in \text{Context}(w)$  DO
    {
         $\mathbf{v}(u) := \mathbf{v}(u) + \mathbf{e}$ 
    }
```

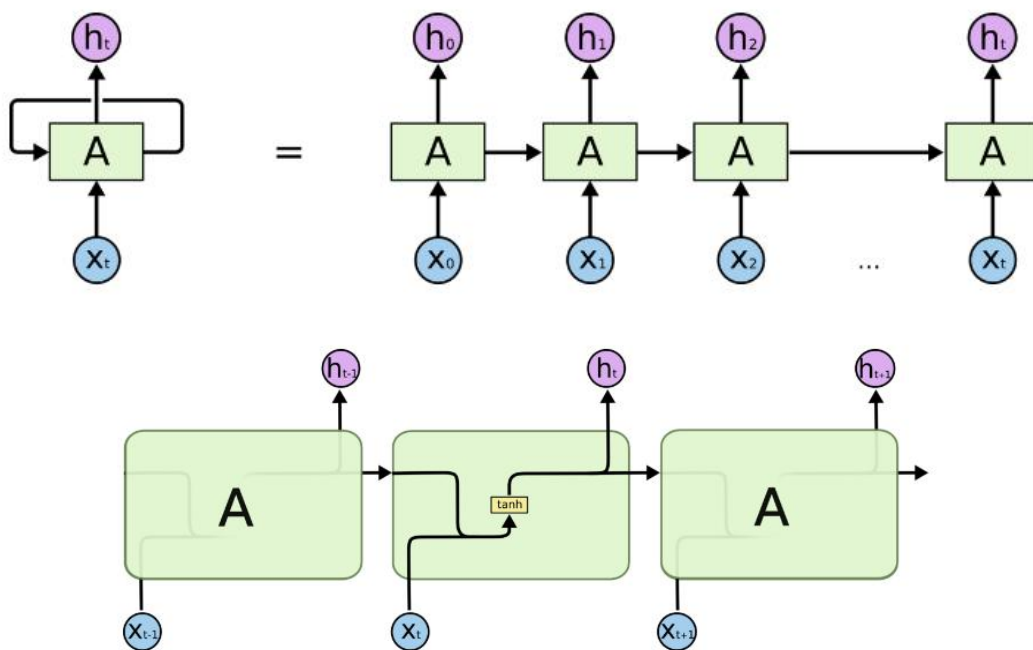
---

## 2.3 CNN ( Convolutional Neural Network )



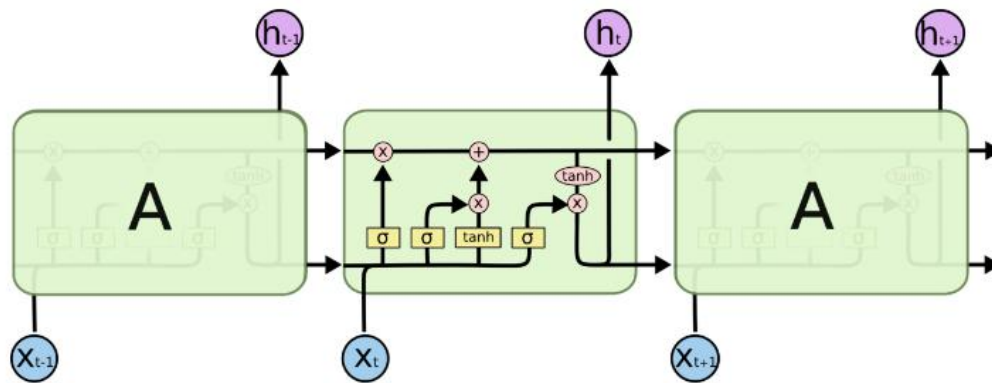
CNN 神经网络结构图（应用于文本）

## 2.4 RNN ( Recurrent Neural Network )

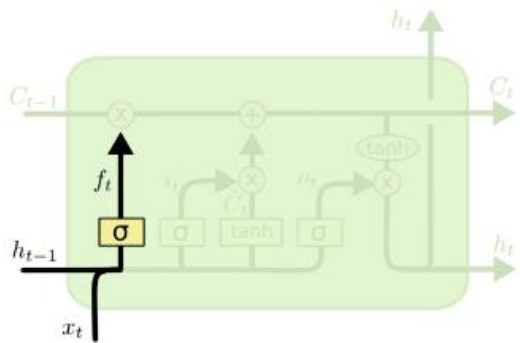


RNN 神经网络结构图

## 2.5 LSTM(Long Short Term Memories)

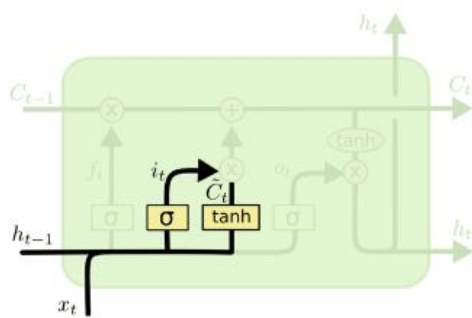


LSTM 神经网络结构图



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

遗忘门

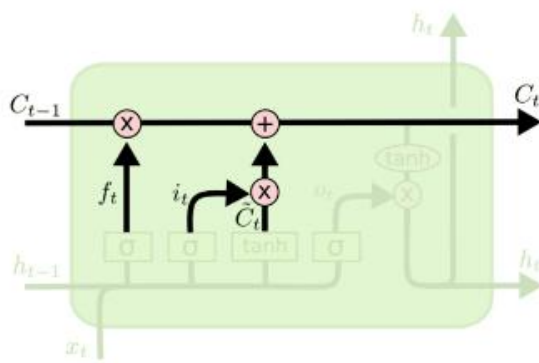


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

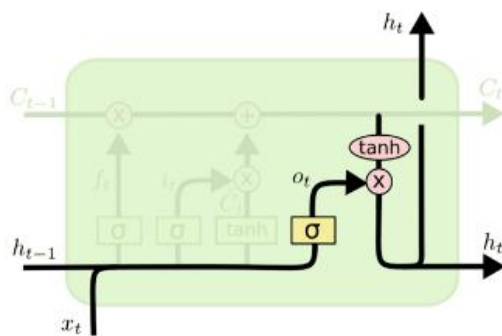
输入门





$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

更新细胞状态



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

输出门

目前代码已经实现了 CNN , RNN , LSTM 和 word2vec。代码地址为  
<https://github.com/RaymondTseng/DeepLearningForJava>