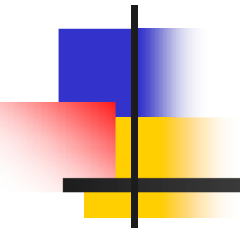
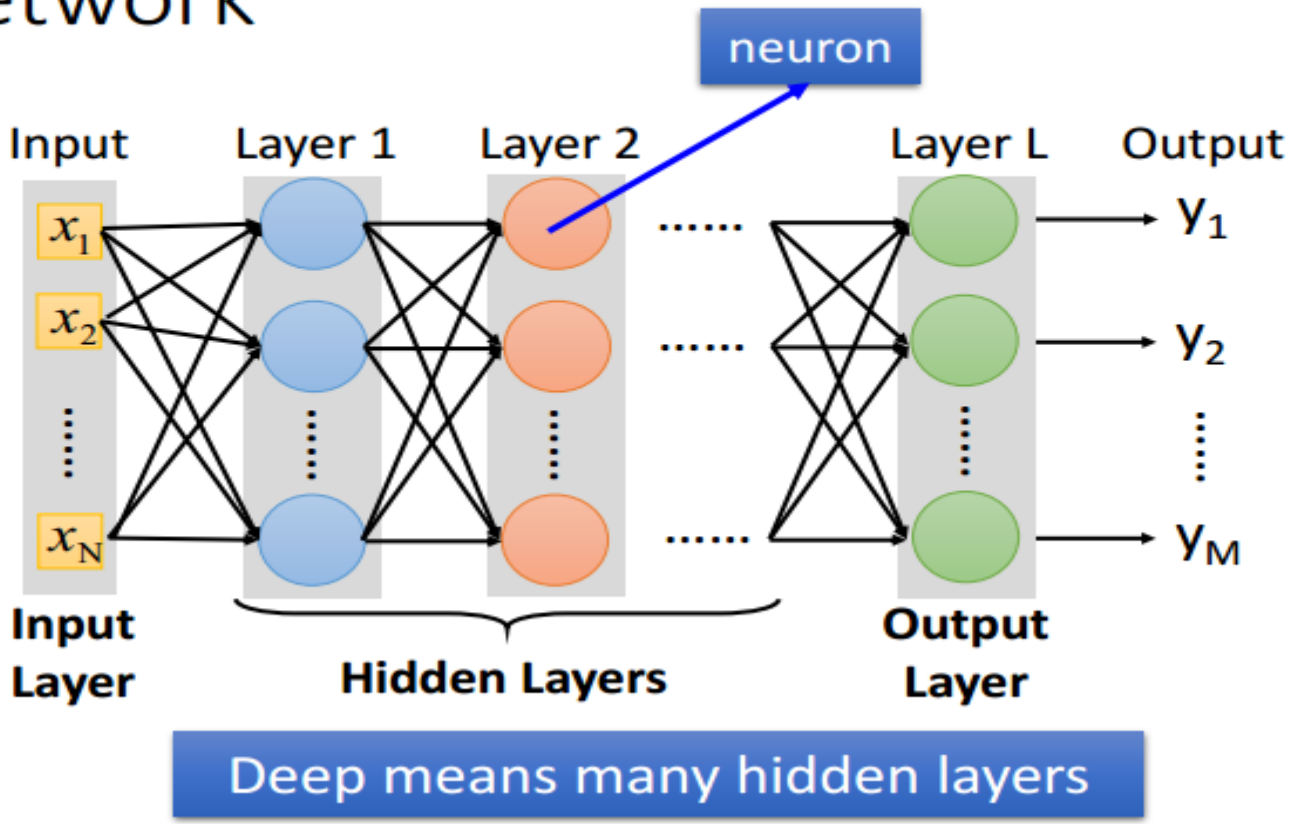


深度神经网络

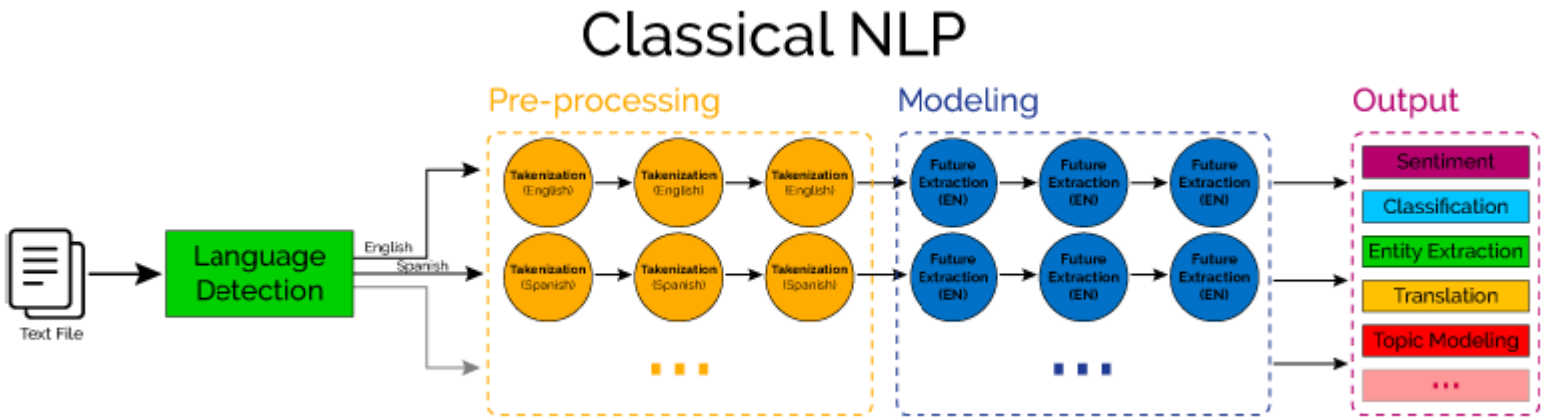
深度神经网络

Fully Connect Feedforward Network

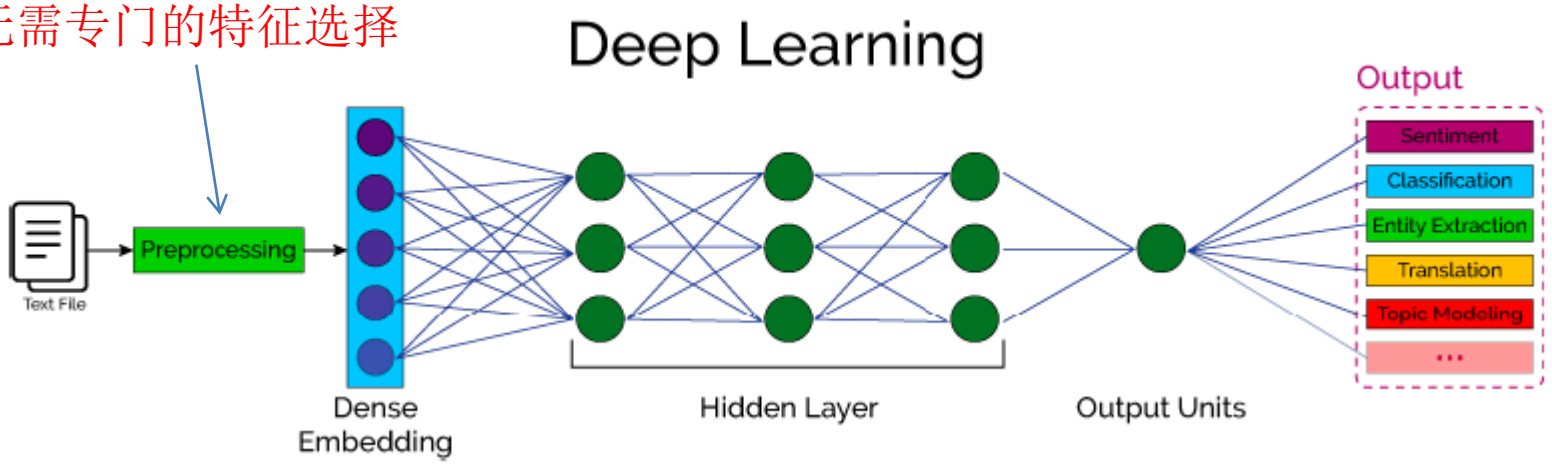


Deep means many hidden layers

深度学习与传统机器学习



无需专门的特征选择

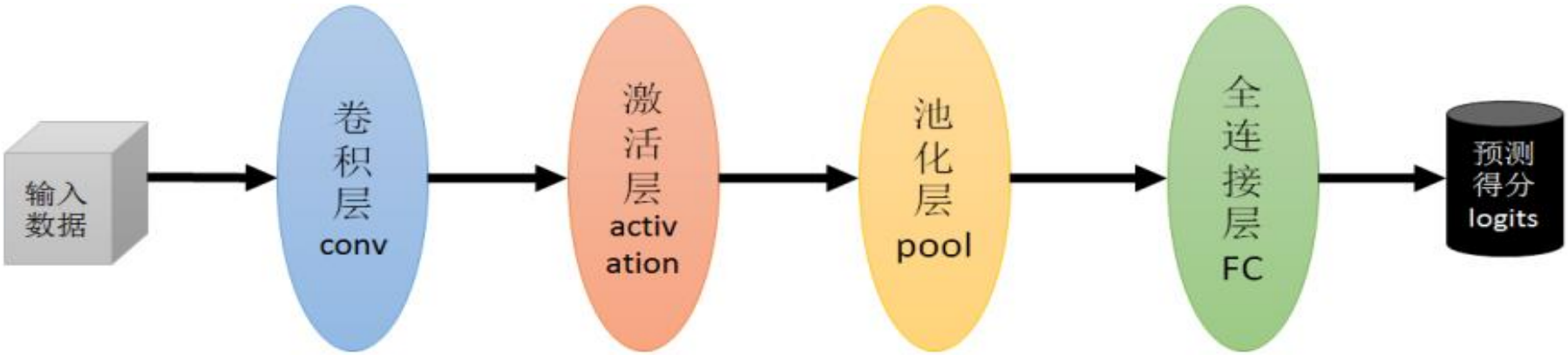


卷积神经网络分析

一个简单的卷积神经网络

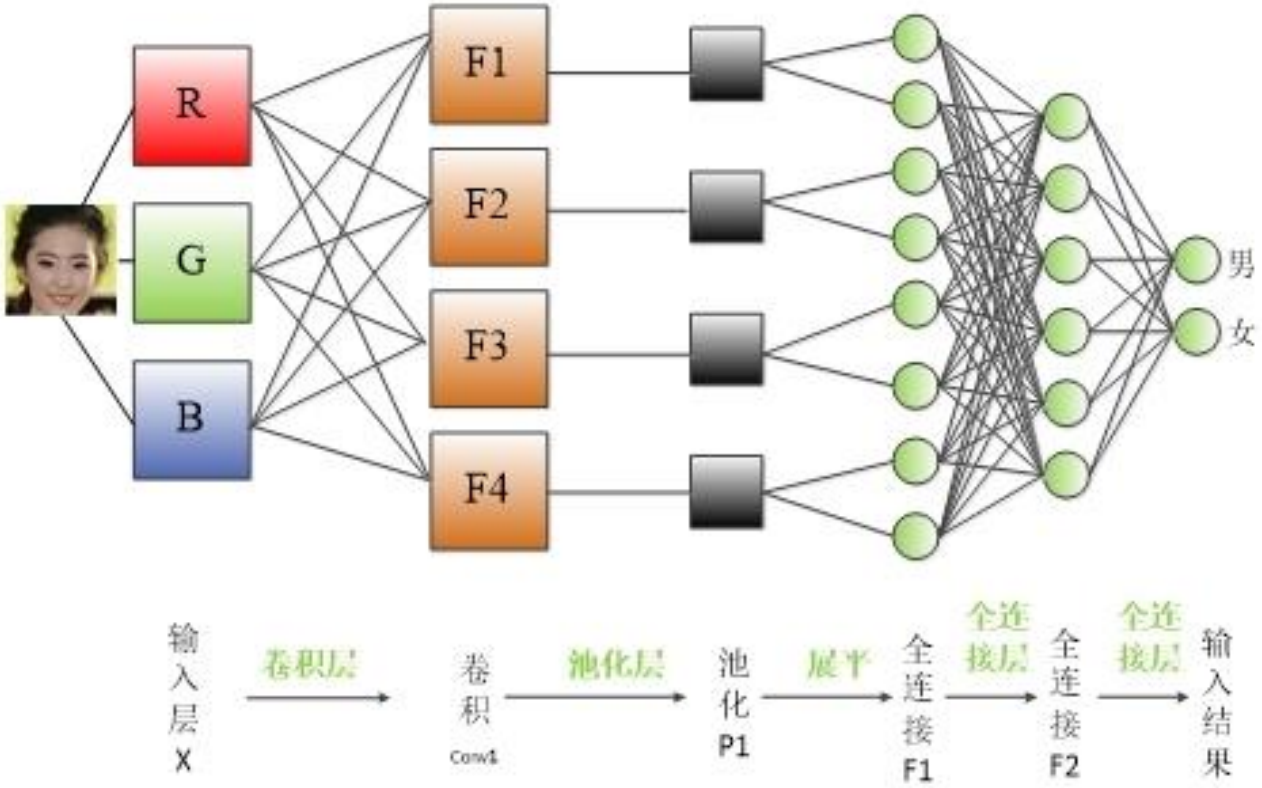
构成：

- 数据输入层　　卷积层
- 激活层　　池化层
- 全连接层



卷积神经网络分析

一个简单的卷积神经网络



卷积神经网络分析

卷积层

- 卷积层是卷积神经网络的核心模块。
- 作用：进行特征提取(选出最重要的特征)
- 这里特征提取是自动而非通过人工指定的，主要通过filter（也称卷积核）在模式（如图像）上平移来提取。就好比人观察图片，通过上下扫描图片，提取想要的信息。

卷积神经网络分析

卷积层

- 局部感知

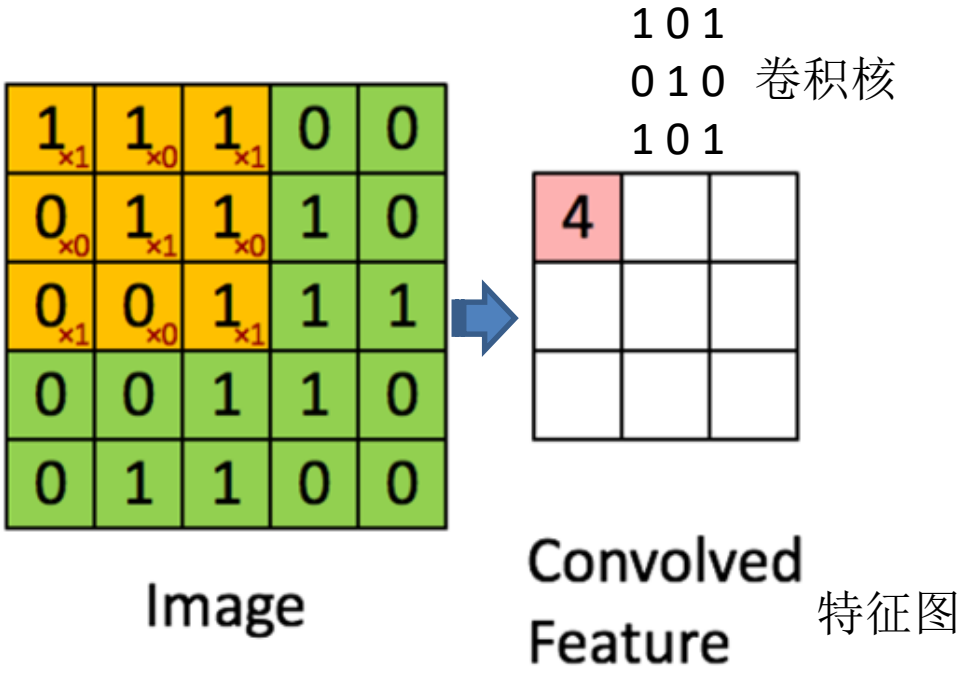
人识别图片的过程中，并不是整张图同时识别，而是对于图片中的每一个特征首先局部感知，然后更高层次对局部信息进行综合，从而得到全局信息。

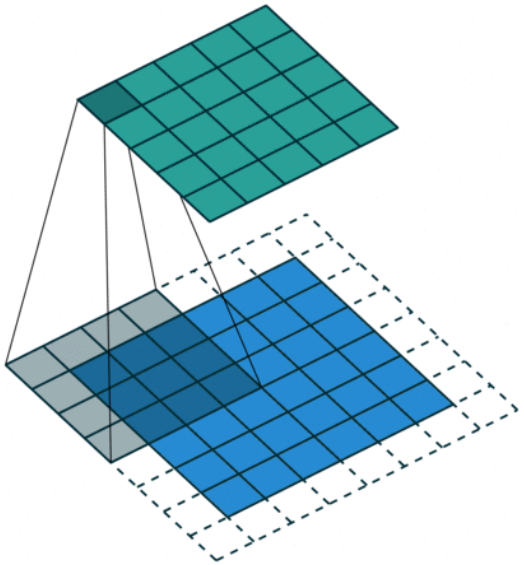
类似于蚂蚁寻食，每个蚂蚁通过局部感知，找到它认为最大个的食物，再通过综合，获得全局最大食物。

卷积神经网络分析

卷积层

- 通过卷积运算得到特征图





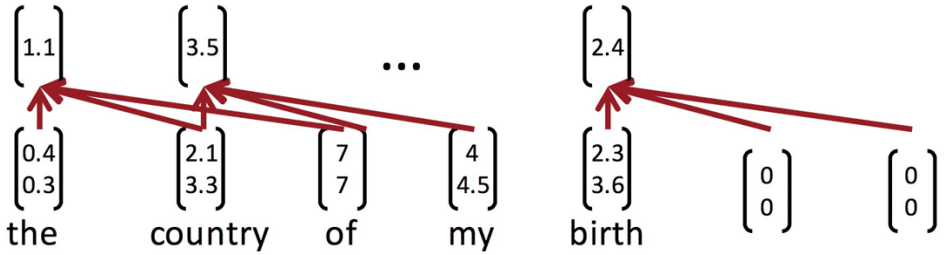
把每次移动的距离称为步幅s

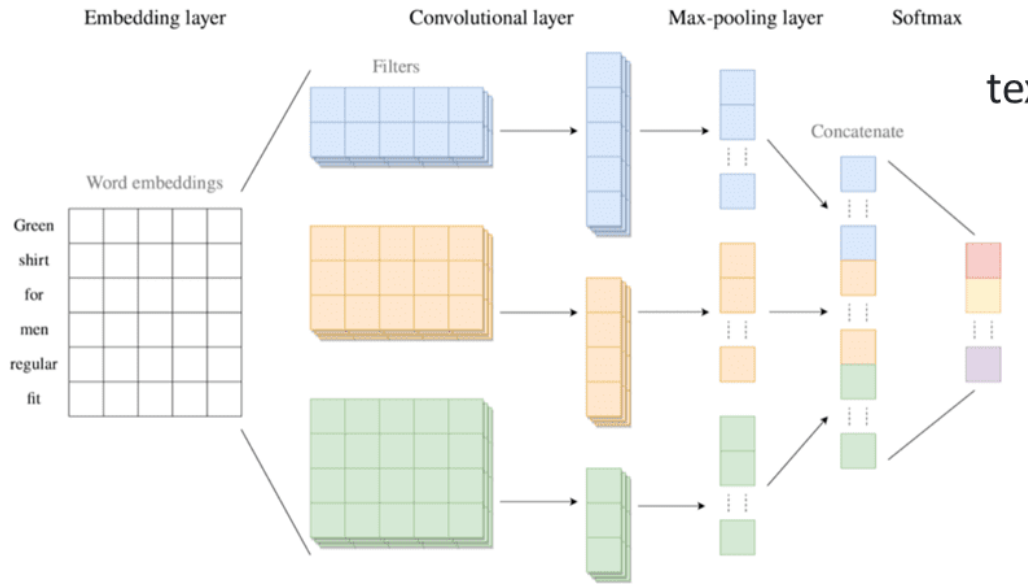
$$1*1 + 1*0 + 1*1 + 0*0 + 1*1 + 1*0 + 0*1 + 0*0 + 1*1 = 4$$

卷积神经网络分析

卷积层

- NLP中的卷积示意





text cnn中的多卷积核

卷积神经网络分析

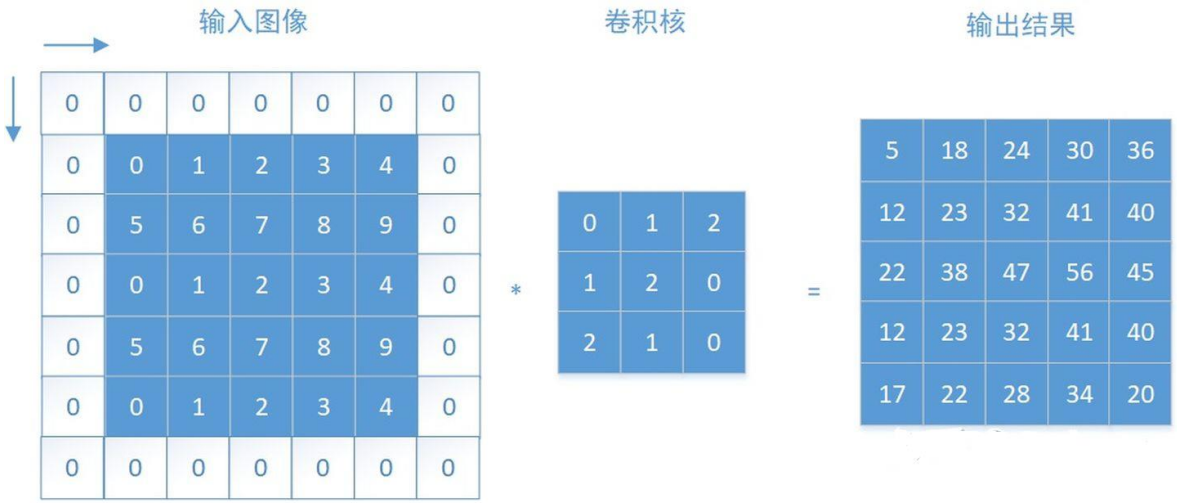
卷积层

- 填充padding

  在输入矩阵周围填充0。

- 目的

  保持边界信息　　使得输入输出图像尺寸一致



卷积神经网络分析

- 卷积层

卷积神经网络分析

- 卷积层

1	2	1
0	0	0
-1	-2	-1

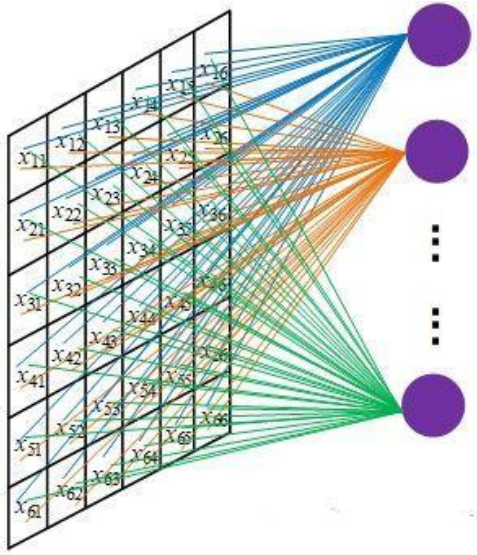
1	0	-1
0	0	0
-1	0	1

卷积神经网络分析

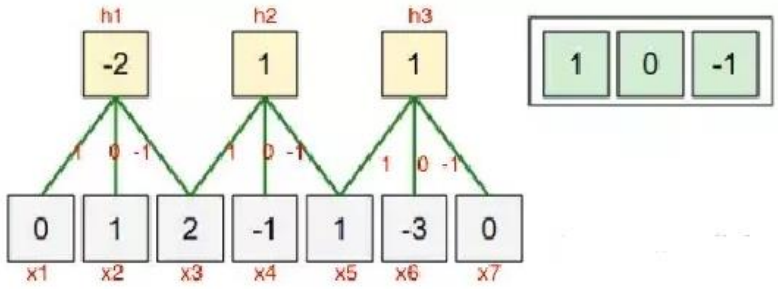
与全连接层相比，卷积层的两个主要优势：

（1）参数共享：　　卷积核在数据上移动　使用相同的权值向量来做卷积

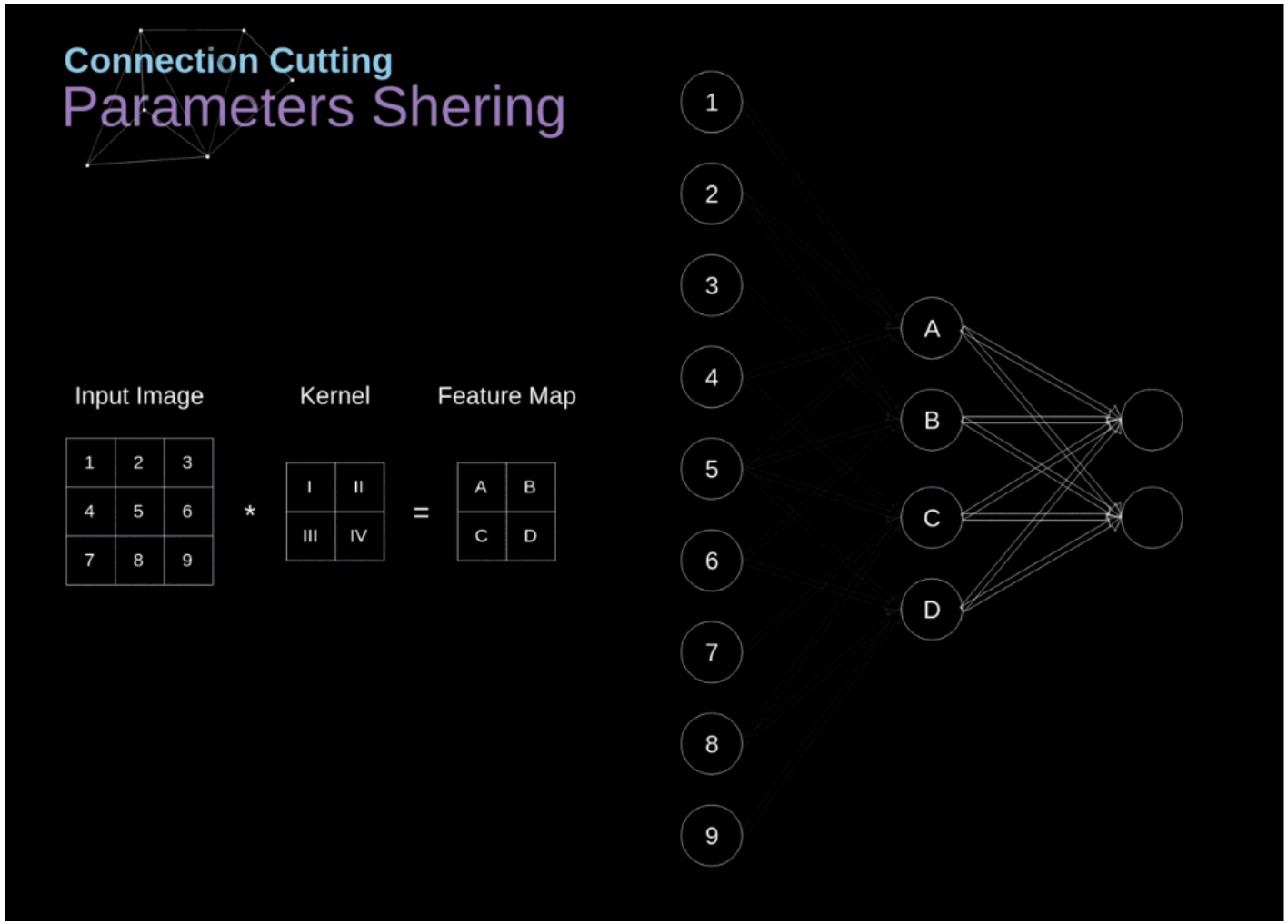
（2）稀疏连接：　　卷积核仅与数据中的某些个部分连接



对比



卷积神经网络分析


Connection Cutting
Parameters Shering
Input Image
Kernel
Feature Map
1 2 3
4 5 6
7 8 9
*
I II
III IV
=
A B
C D
1
2
3
4
5
6
7
8
9
A
B
C
D

卷积神经网络分析

激活层

激活层一般紧接在卷积层后，是对卷积层的输出结果做一次非线性映射。

如果不用激活函数，每一层的输出都只是上一层输入的线性函数。

常用的激励函数有：

(1) Sigmoid函数

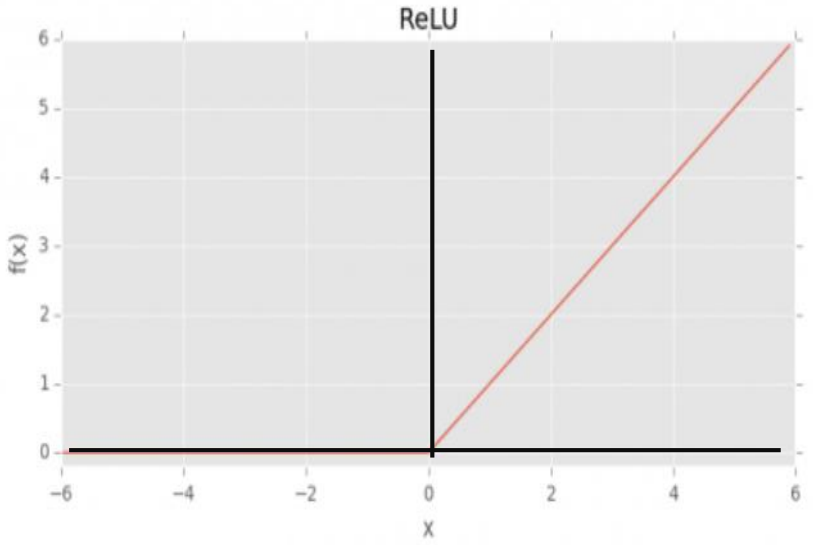
(2) Tanh函数

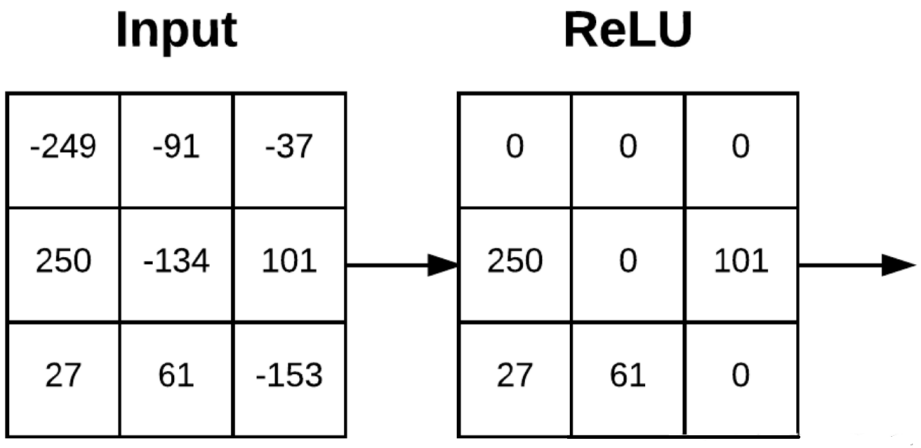
(3) ReLU

卷积神经网络分析

激活层

Sigmoid函数
Tanh函数
ReLU





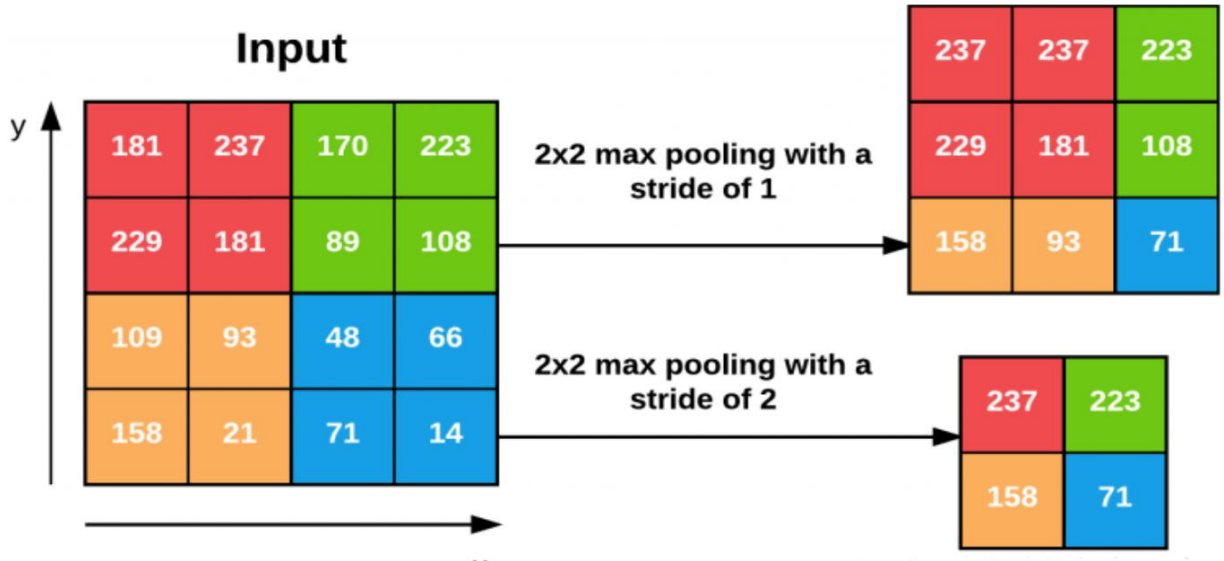
卷积神经网络分析

池化层

- 主要用于特征降维，压缩数据和参数的数量，减小过拟合，同时提高模型的容错性。主要有：

  Max Pooling：最大池化, 选取最大值

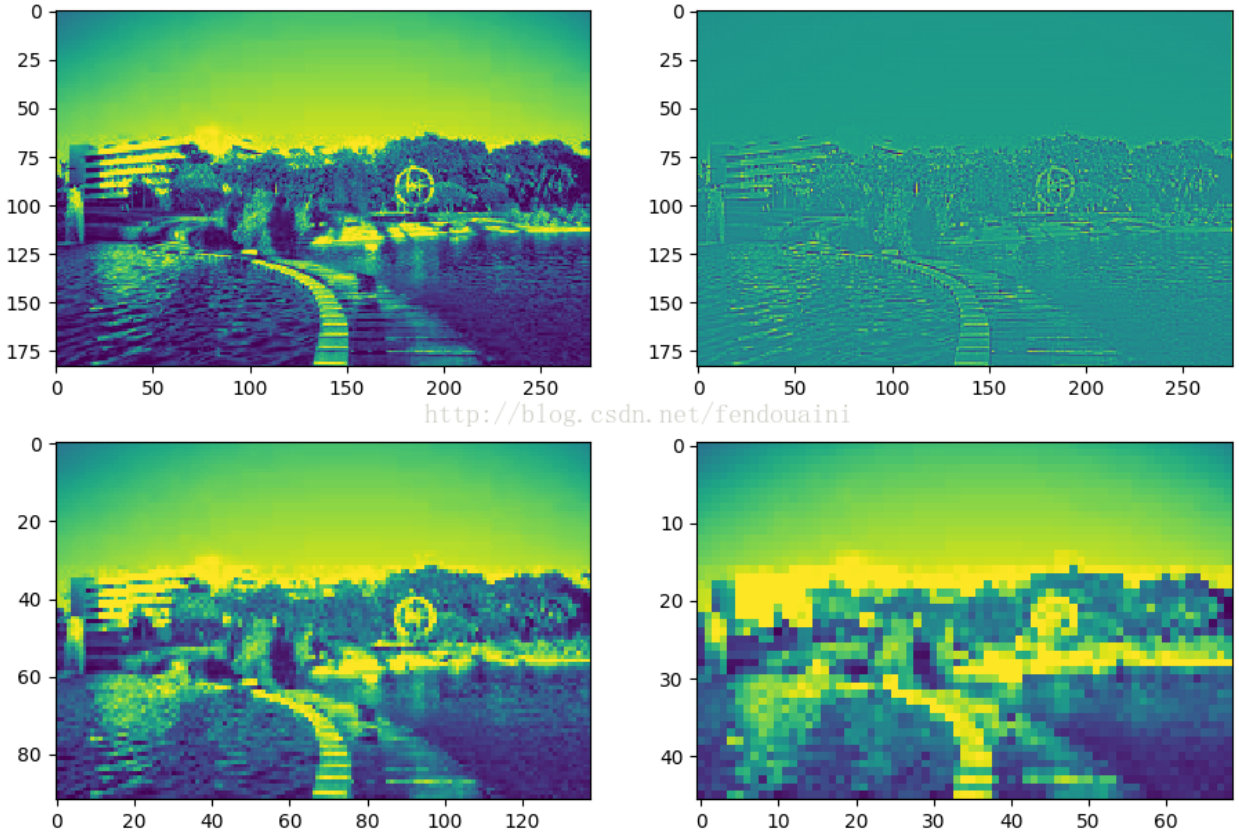
  Average Pooling：平均池化，选取平均值



卷积神经网络分析

池化层

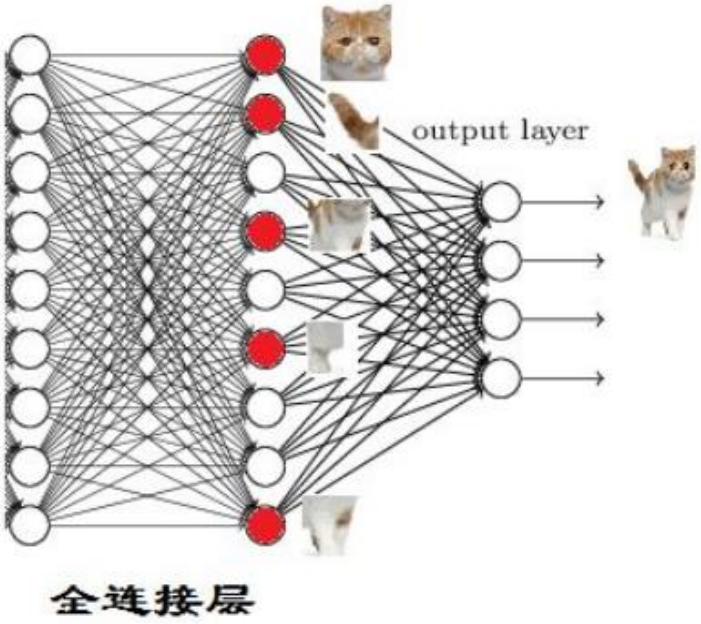
卷积和池化效果



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，作用是把前边提取到的特征综合起来。一般全连接层的参数是最多的。

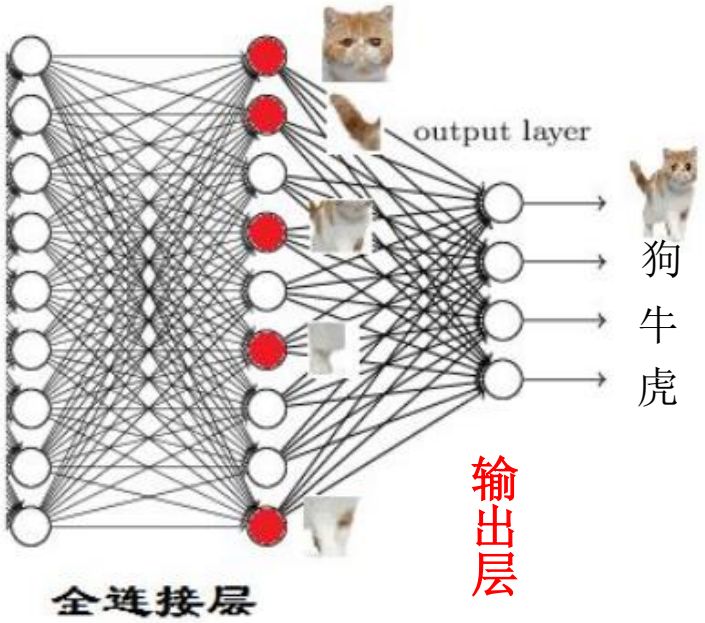


例如VGG16中，第一个全连接层FC1有4096个节点，上一层POOL2是25088个节点，则需要4096*25088个权值，需要很大的内存。

卷积神经网络分析

模型输出

卷积神经网络的最后一层往往是全连接层+Softmax（分类网络），通过它给每种类别赋予一个概率。

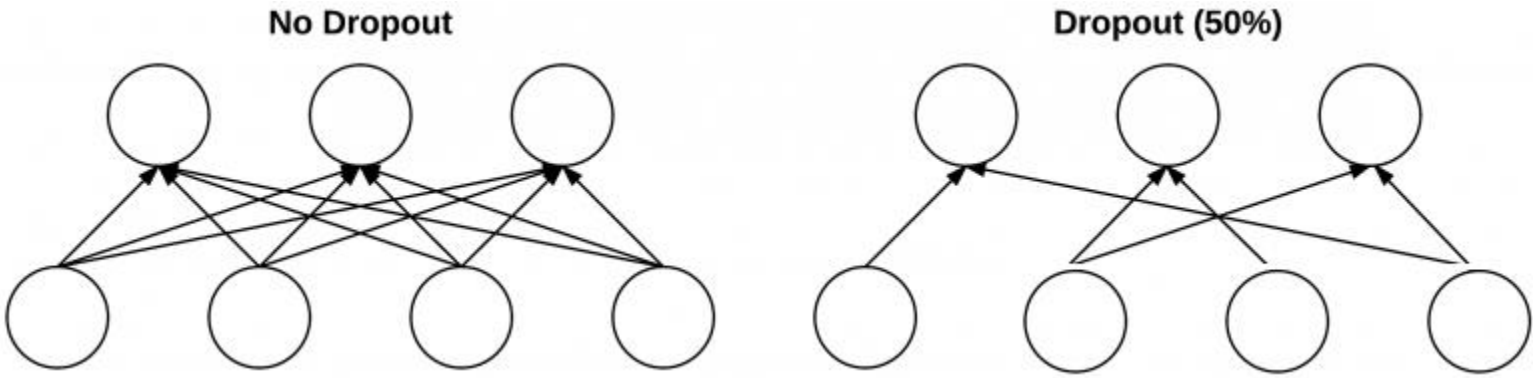


$$softmax(z_j) = \frac{e^{z_j}}{\sum_K e^{z_j}}$$

卷积神经网络分析

Dropout

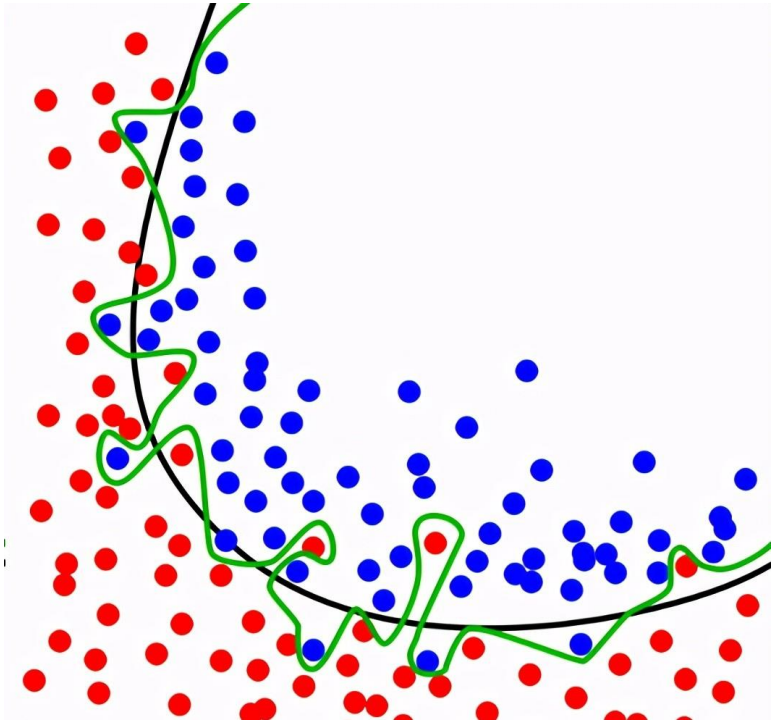
Dropout 实际上是一种正则化形式，旨在防止过拟合。对于训练集中的每个batch，dropout 以概率p随机断开网络中从前一层到下一层的部分输入。



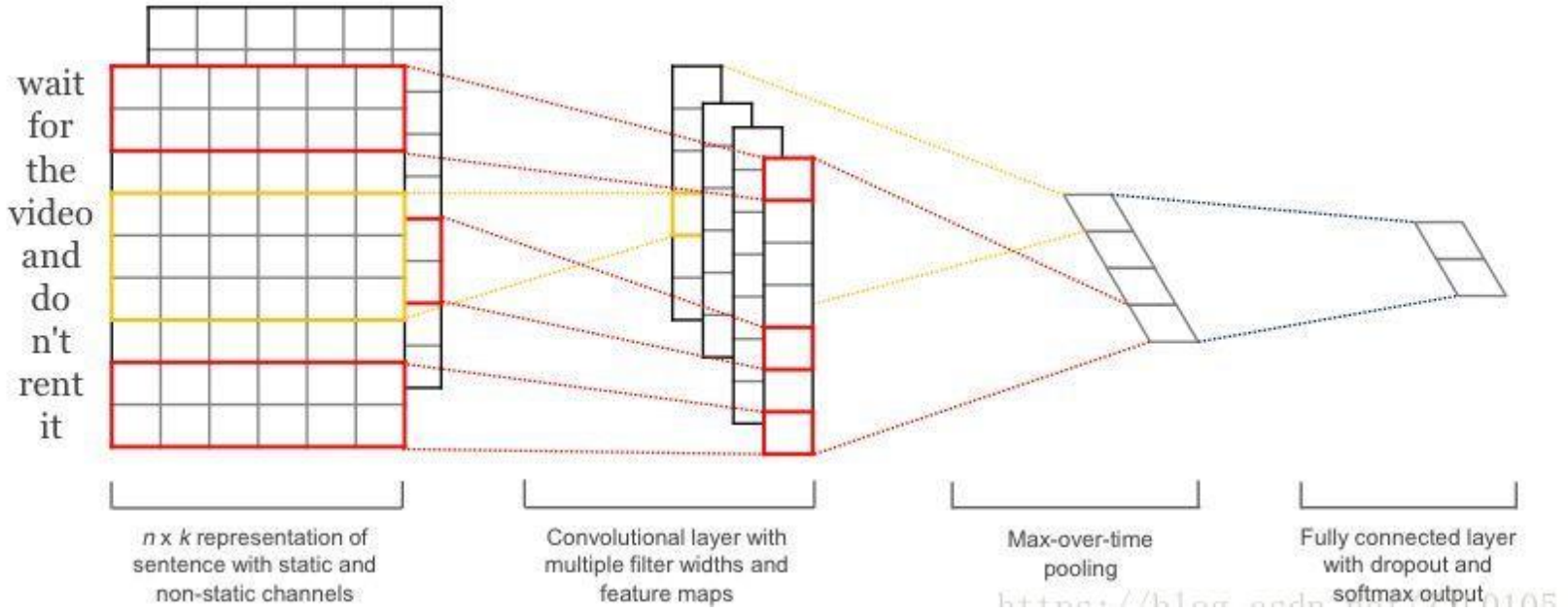
卷积神经网络分析

Dropout

图中黑色曲线是正常模型，绿色曲线是overfitting模型。尽管绿色曲线精确地区分了训练数据，但是对新数据的适应性较差。



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。

卷积神经网络实现

Keras 中文文档: https://keras-zh.readthedocs.io/

```
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Conv1D可加入卷积层，换成MaxPooling1D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

循环神经网络RNN

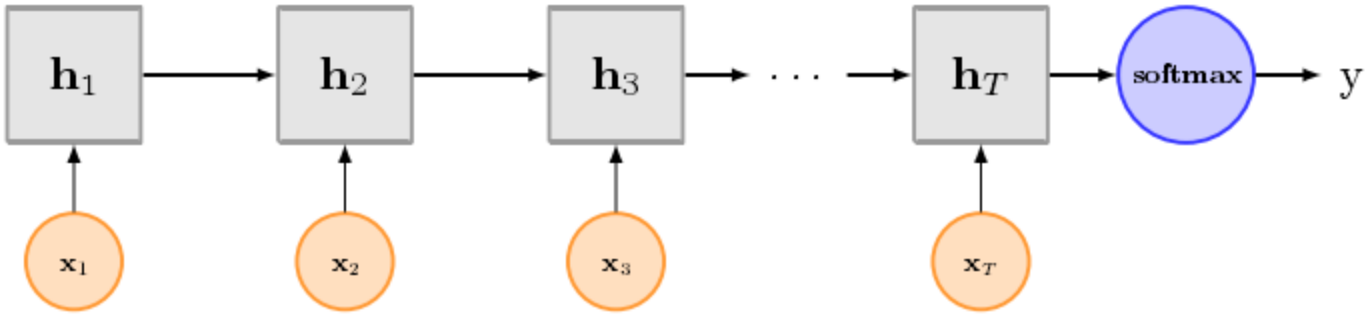
- 动机：很多任务需要处理序列关系。比如，当在理解一句话的意思时，孤立地理解这句话的每个词是不够的，需要处理这些词连接起来的整个序列；
- 时序预测问题（timeseries），诸如预测天气、温度

普通的神经网络假设输入和输入之间是相互独立的。

RNN中不仅每层之间的神经元存在连接，同一层之间的神经元也存在连接，上一个时刻的状态能作用于下一个时刻的状态。

RNN模型应用

- 文本分类
  方式1：使用最后的隐变量作为句向量



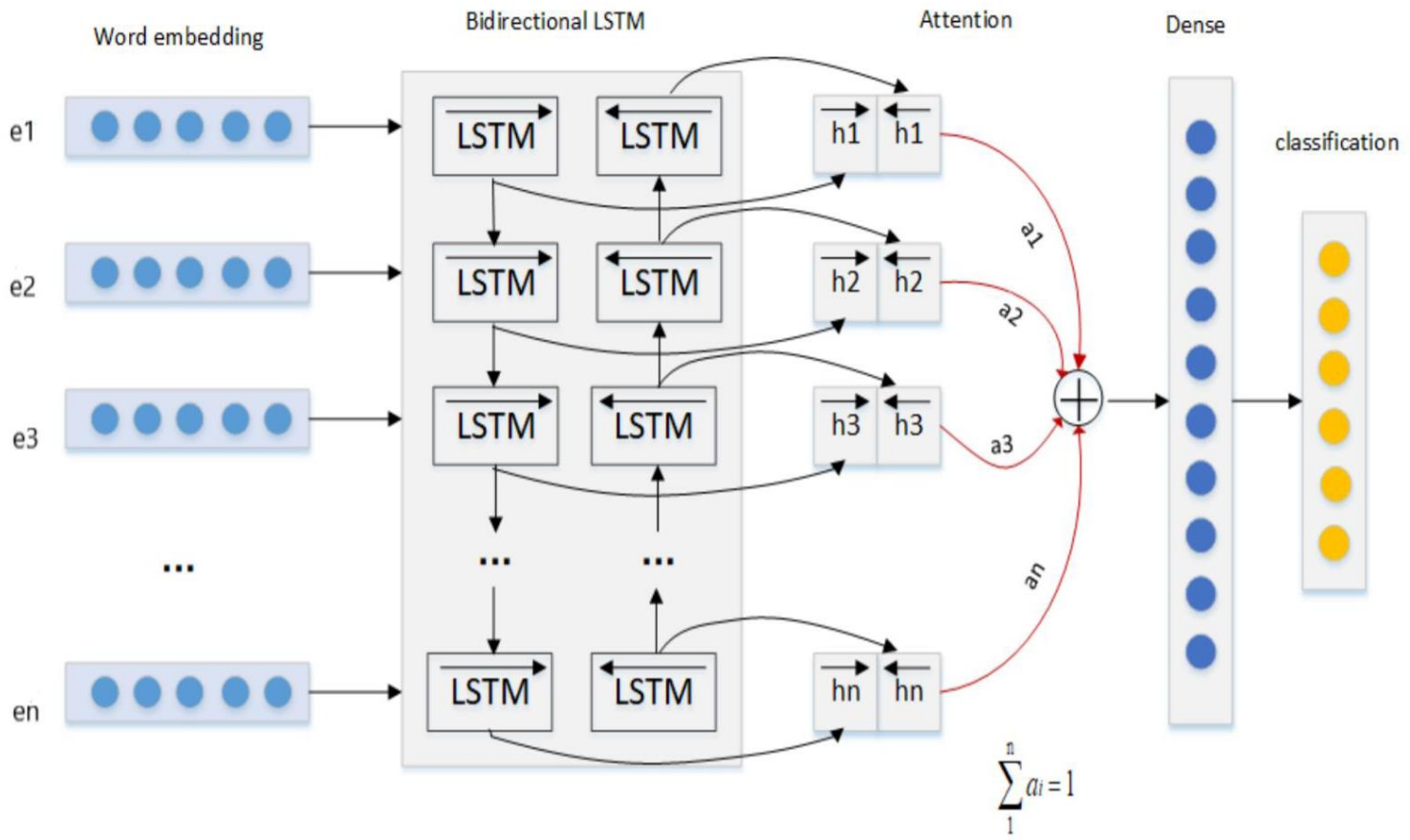
数据　　挖掘

基于Keras实现LSTM

```
max_features = 20000        #max_features给出了最多使用的单词数
batch_size = 32
(x_train, y_train), (x_test, y_test) = imdb.load_data(path='./imdb.npz',num_words=max_features)

model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(128, dropout=0.2, return_sequences=false, recurrent_dropout=0.2))  #输出层的维度128，即每个门的输出。 return_sequences=True表示返回每个时间片的隐状态，false返回最后时刻的隐状态。
model.add(Dense(1, activation='sigmoid'))          #1个输出，二分类，如果多分类用softmax

model.compile(loss='binary_crossentropy',  optimizer='adam',  metrics=['accuracy']) #使用Adam优化器

model.fit(x_train, y_train,  batch_size=batch_size,  epochs=15)
score, acc = model.evaluate(x_test, y_test, batch_size=batch_size)
```

RNN模型应用

- 文本分类　　方式2：使用每个词的隐变量拼成句向量



RNN模型应用

- 双向LSTM实现

```
model = Sequential()
model.add(Bidirectional(LSTM(10)))
model.add(Dense(5))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy',optimizer='rmsprop')
```

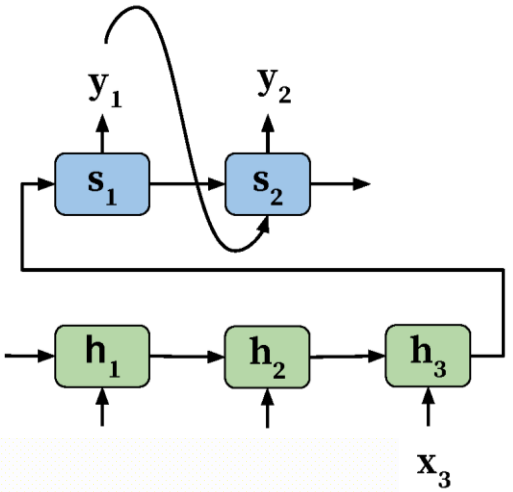
```
keras.layers.Bidirectional(layer, merge_mode='concat', weights=None)
```

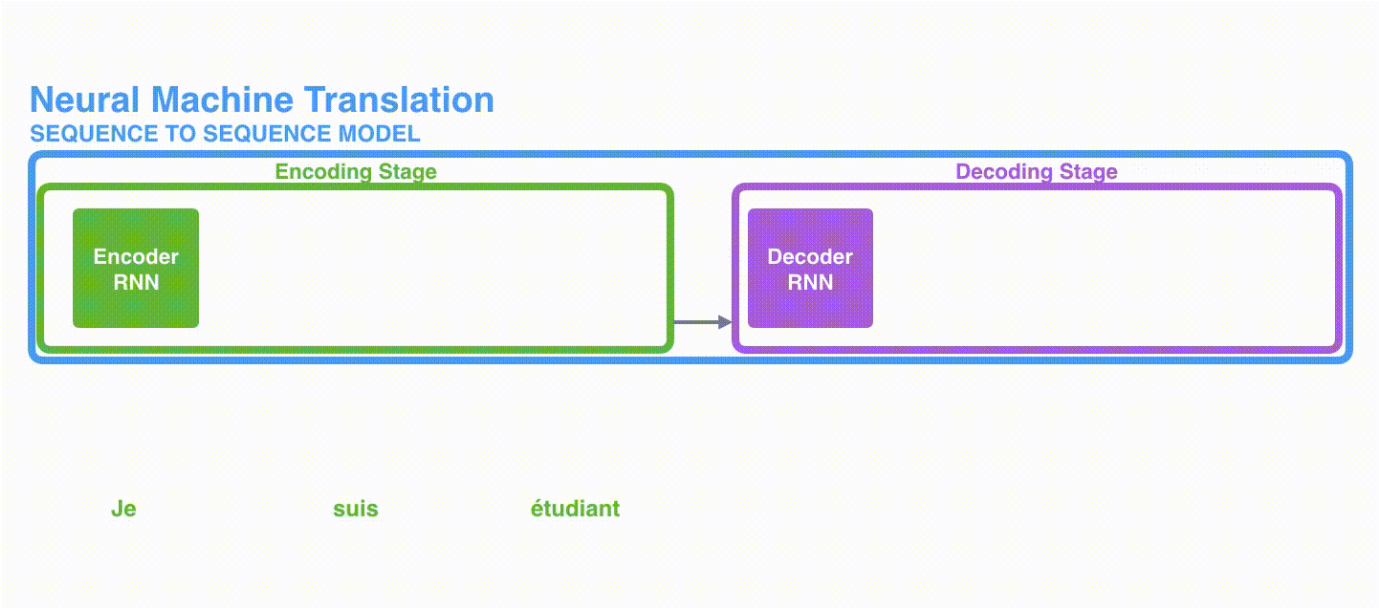
merge_mode: 前向和后向 RNN 的输出的结合模式。 为 {'sum', 'mul', 'concat', 'ave', None} 其中之一。

RNN模型应用

- 神经网络机器翻译

翻译器由编码器和解码器构成：





RNN模型应用

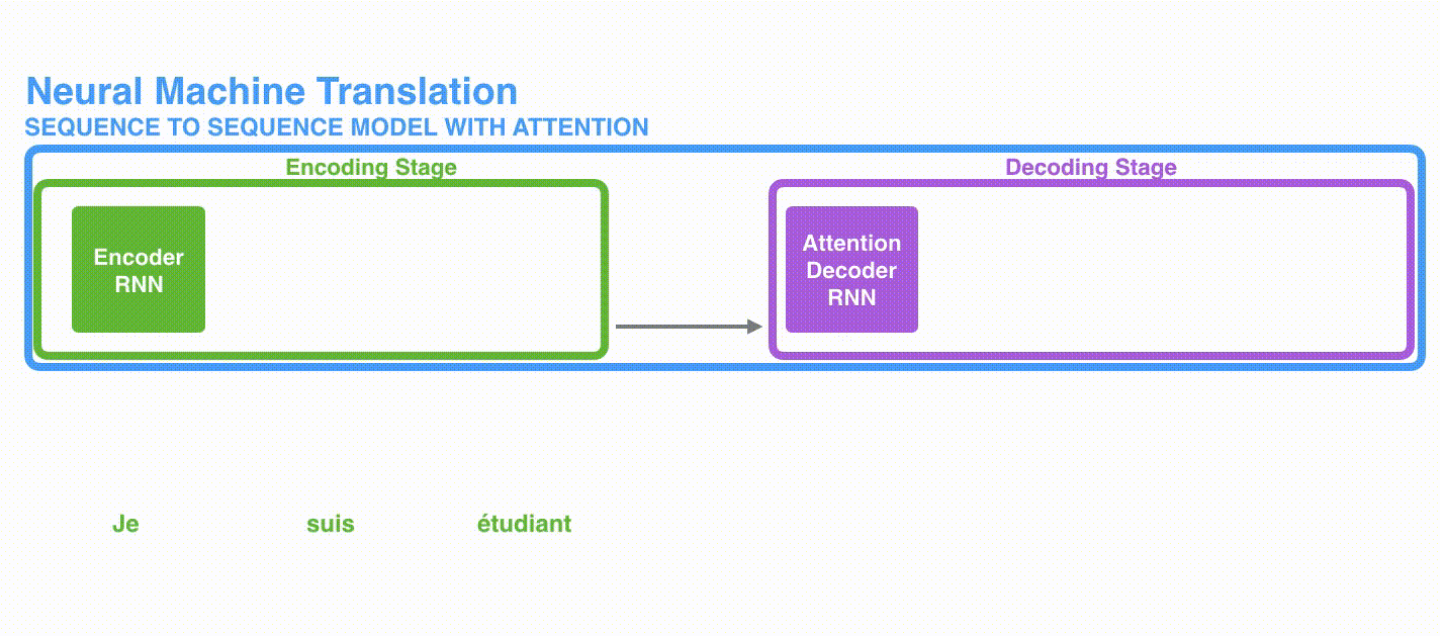
- 神经网络机器翻译

编码器-解码器结构有两个缺陷：

（1）编码器需将所有输入压缩成固定长度的向量 ， 造成输入信息的丢失。

（2）不能对输入序列和输出序列的(位置)对应关系进行建模。

解决方法：引入注意力机制。



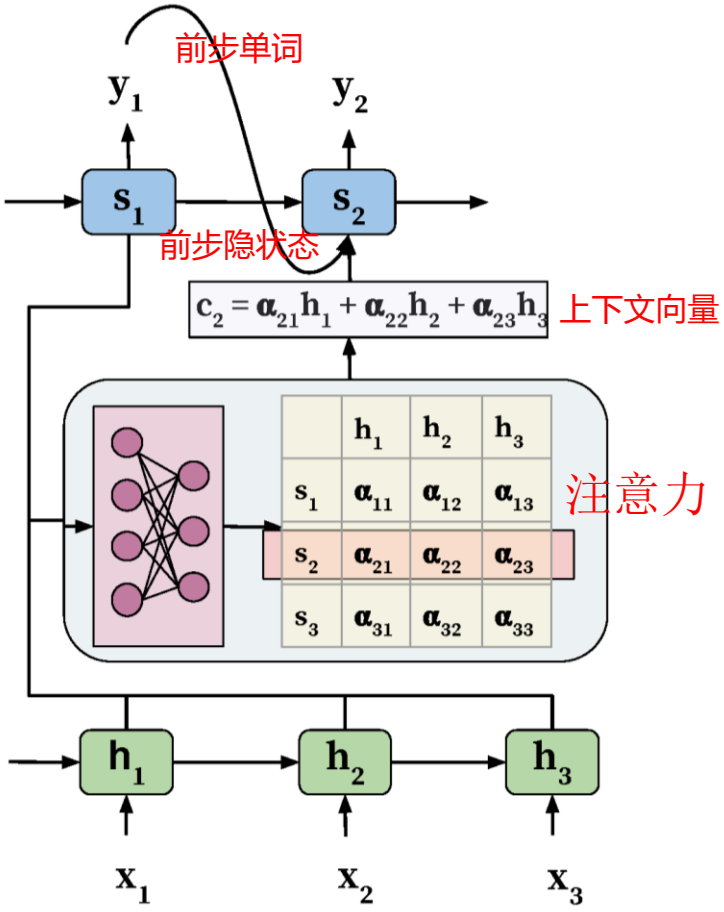
RNN模型应用

- 神经网络机器翻译

利用注意力改进：

（1）编码器不只传递编码阶段的最后一个隐藏状态，而是将所有隐藏状态传递给解码器：

（2）解码器计算当前隐状态与编码器传入的每个隐状态的相关性，产生注意力权值；



主流深度学习框架

目前最主流的是TensorFlow

华为的AI计算框架：
MindSpore

常见的深度学习框架

Thanks

谢谢！