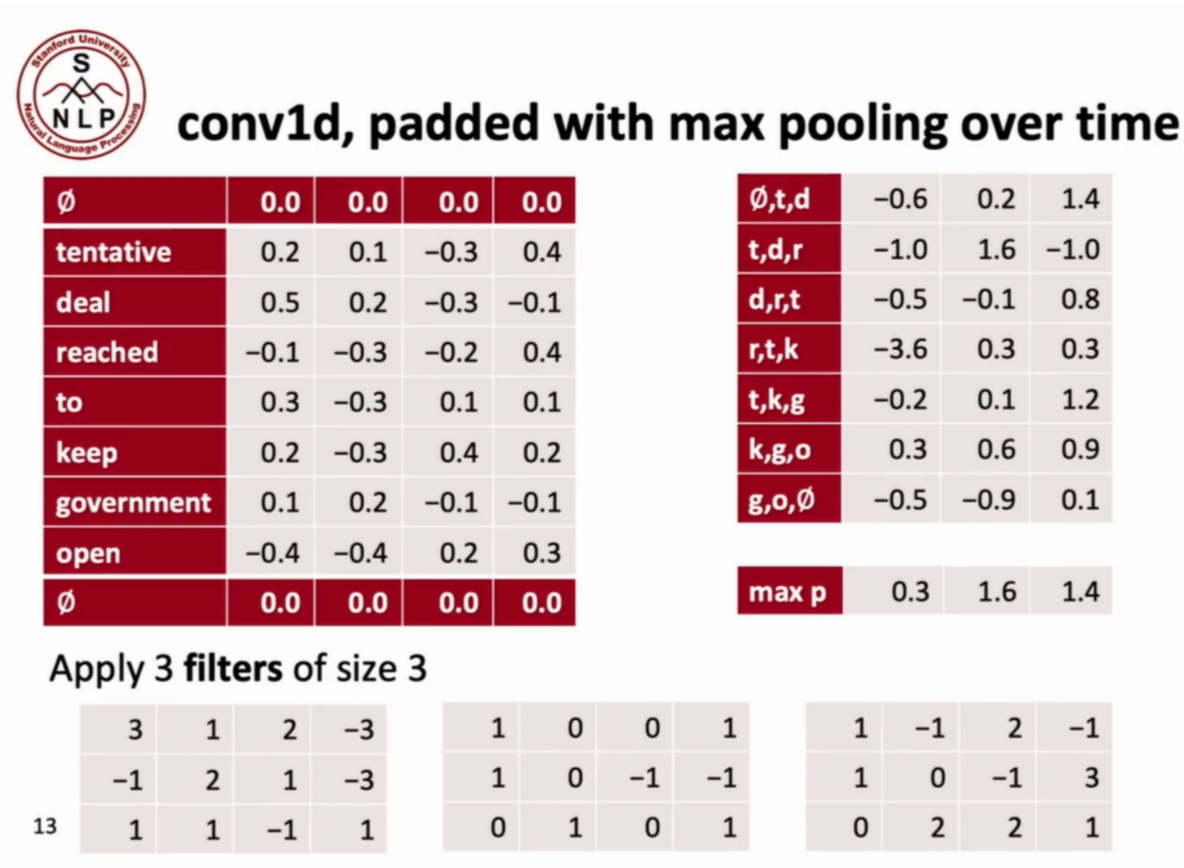
CNN

RNN不能提取一个span的特征，并且过度关注最后一个状态，但CNN可以关注每个span，适合分类任务。

1D convolution for text

以word数量为长度，wordvec维数为in channel数量，filter个数为out channel数量:



conv1d, padded with max pooling over time

∅	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

∅,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,∅	−0.5	−0.9	0.1
max p	0.3	1.6	1.4

Apply 3 **filters of size 3**

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

13

max pooling over time对时间维度(word)做max pooling，比如表示整句话是否表达polite。average pooling同理，表示整句话总体的polite程度。通常max pooling在分类时效果最好。

在CNN中，如果输入128(channel)x32x32的图片，希望输出512x30x30，则Pytorch生成512个128x3x3的卷积核。在这里的1d卷积中同理。

```
batch_size = 16
word_embed_size = 4
seq_len = 7
input = torch.randn(batch_size, word_embed_size, seq_len)
conv1 = Conv1d(in_channels=word_embed_size, out_channels=3,
                    kernel_size=3) # can add: padding=1
hidden1 = conv1(input)
hidden2 = torch.max(hidden1, dim=2)  # max pool
```

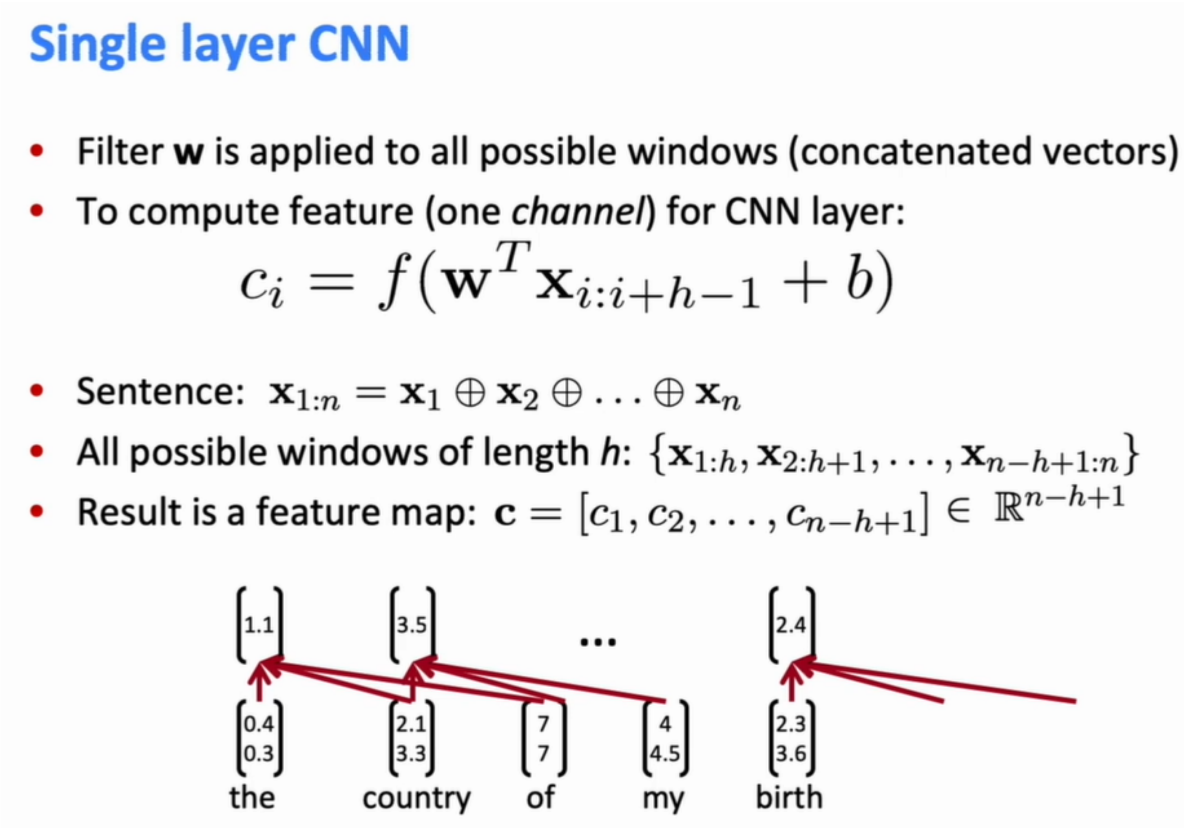
直观上，希望每个channel关注不同的特征。

压缩特征图的方法

- k-max pooling: 留下每个channel的前k个，按原来顺序排列
- stride: 步长
- dilation: 空洞卷积，在kernel size不变的情况下作用到更大的范围

Single Layer CNN for Sentecne Classification

和上述CNN相比，本文把window size=h的h个wordvec(k维)直接连起来变成一维向量，即输入的in channel=1。



对channel进行max pooling，window size(kernel size)可以有多种。

Pooling and channels

- Pooling: max-over-time pooling layer
- Idea: capture most important activation (maximum over time)
- From feature map $\mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}] \in \mathbb{R}^{n-h+1}$
- Pooled single number: $\hat{c} = \max\{\mathbf{c}\}$

- Use multiple filter weights **w**
- Useful to have different window sizes *h*
- Because of max pooling $\hat{c} = \max\{\mathbf{c}\}$, length of **c** irrelevant

$$\mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}] \in \mathbb{R}^{n-h+1}$$

- So we could have some filters that look at unigrams, bigrams, tri-grams, 4-grams, etc.

Multi-channel CNN

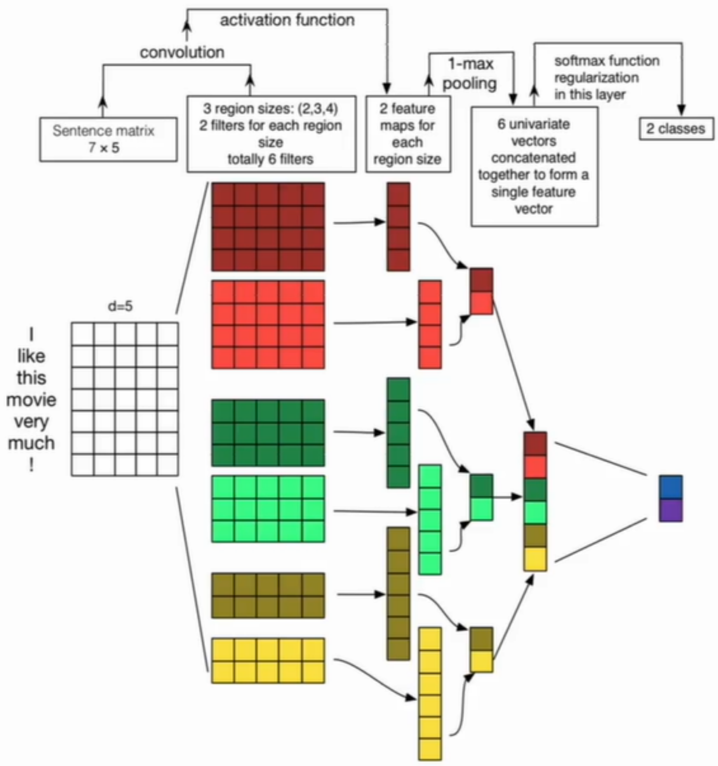
输入的wordvec做两份copy，一份冻结，一份可更新，最后特征合到一起做分类。

From:

Zhang and Wallace (2015) A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification

https://arxiv.org/pdf/1510.03820.pdf

(follow on paper, not famous, but a nice picture)



Some details

- Gated units used vertically (residual block)
- BatchNorm: 减少数据的扰动影响

- 1x1 conv: 和fully connected类似，可以用来只增加channel

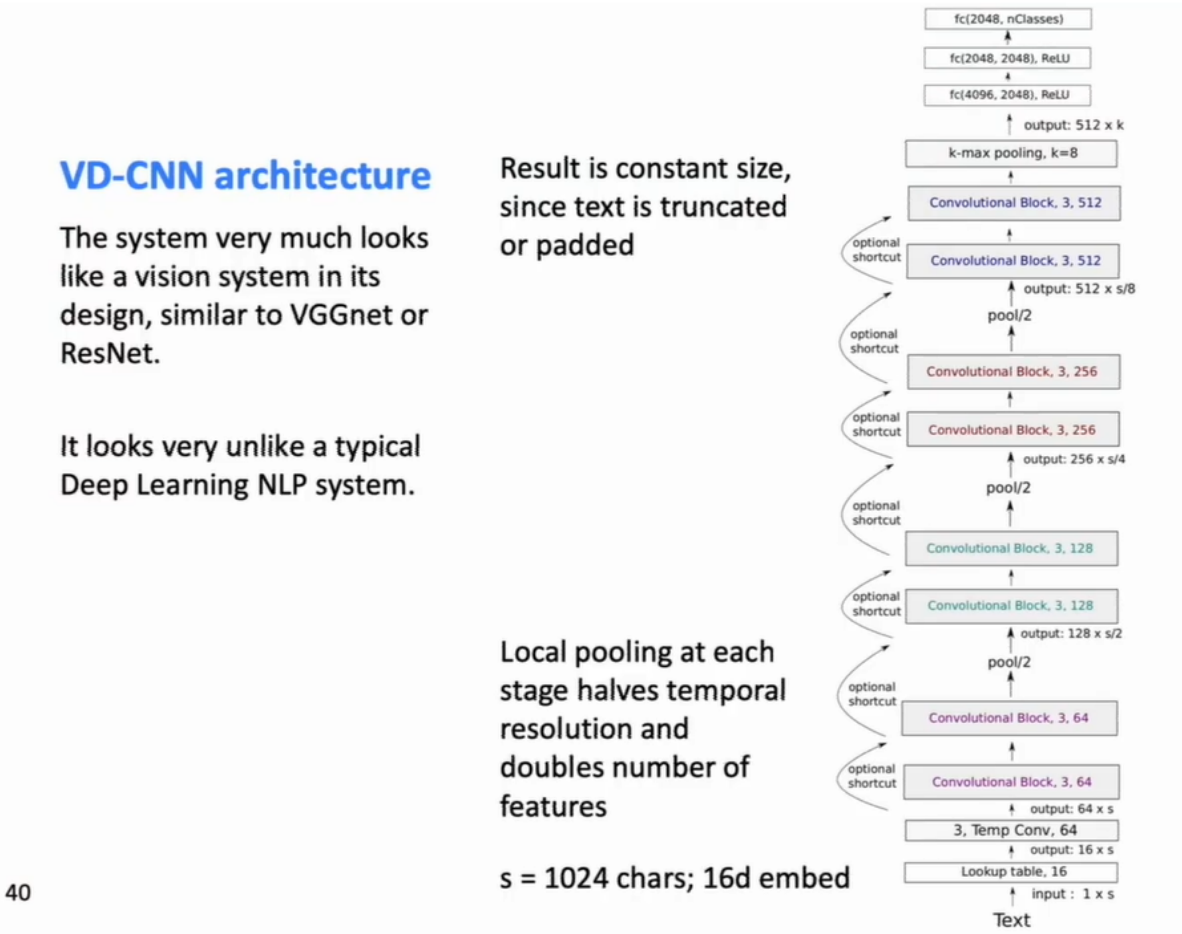
CNN application: translation

在seq2seq之前，有人用CNN作encoder，RNN作decoder做NMT，这是NMT新世纪的第一篇文章。

Very Deep CNN for Text Classification

2017年ResNet的出现在cv中提了很多分，但nlp中lstm却层数很少，因此本文尝试了在nlp中使用多层网络。

输入1024x16的char-level embedding，1024个char，16维特征，然后经过一个他们设计的ResNet。



在实验中，maxpooling效果最好。29层已经是效果最好的深度了，更深的网络效果变差。在cv中29层并不深，ResNet可以到152层。

Quasi-RNN

RNN不能并行，Quasi-RNN结合CNN和RNN。