3.4 文本分类

林洲汉
上海交通大学
2024年秋季学期

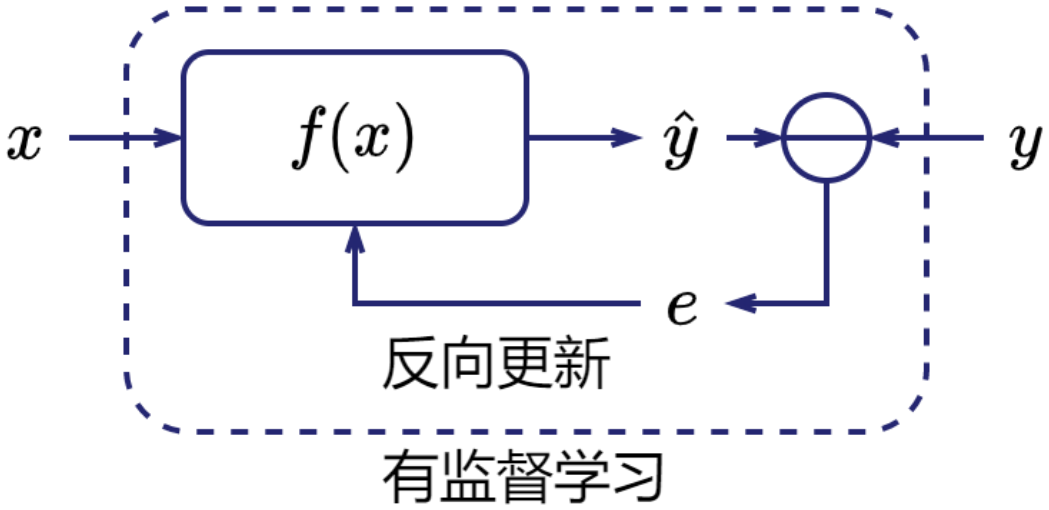
目录

任务分类

任务分类

- 有监督文本分类
  - 依靠有正确类别标签的训练数据训练分类模型
  - **监督信号**：样本的正确标签
  - **典型训练方式**：误差梯度的反向传递
  - 本课程主要介绍有监督的文本分类方法

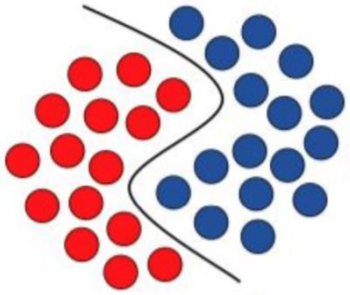
- 有监督文本分类
  - 依靠有正确类别标签的训练数据训练分类模型
  - **监督信号**：样本的正确标签
  - **典型训练方式**：误差梯度的反向传递



任务分类

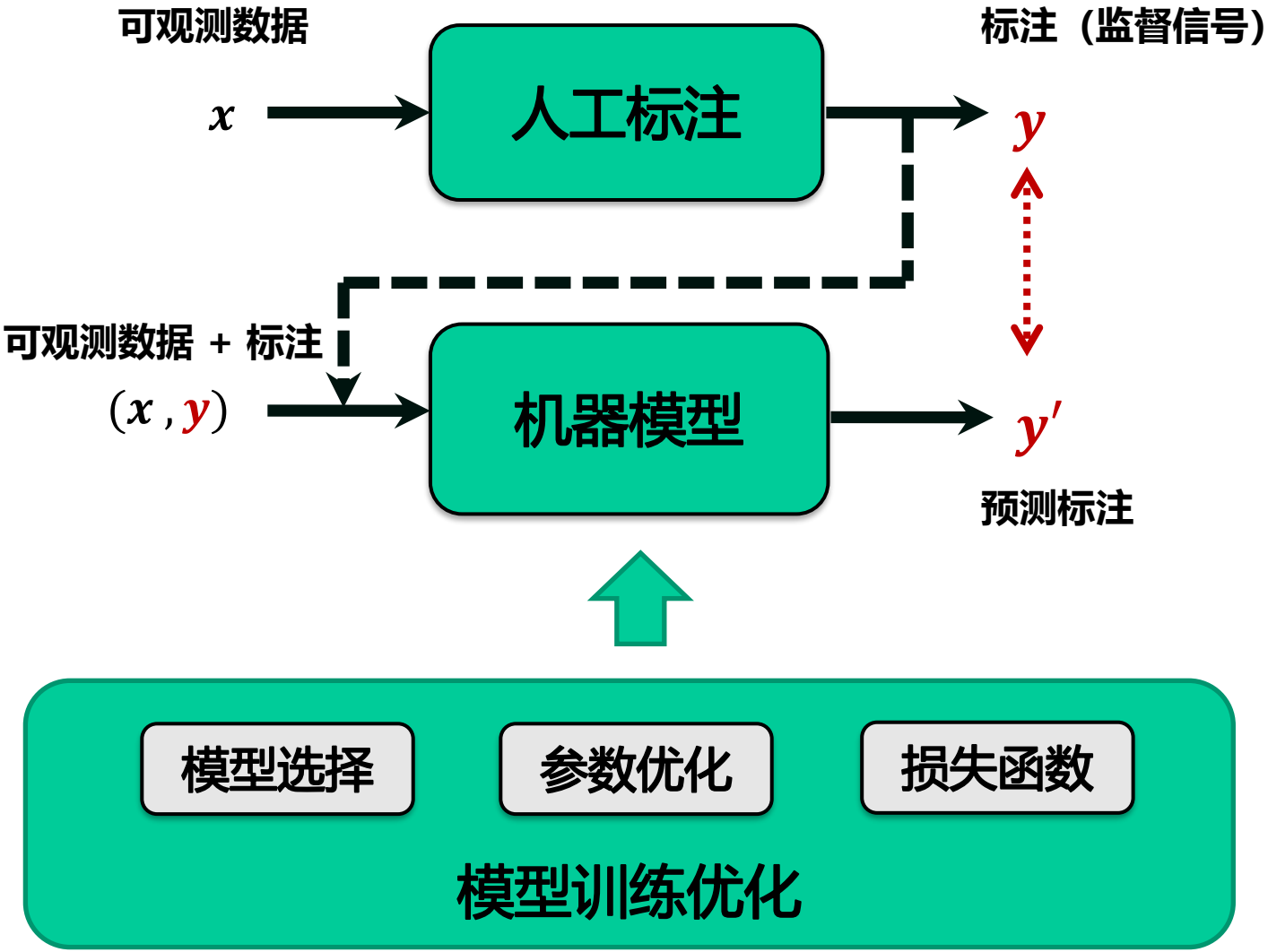
有监督文本分类

有监督学习

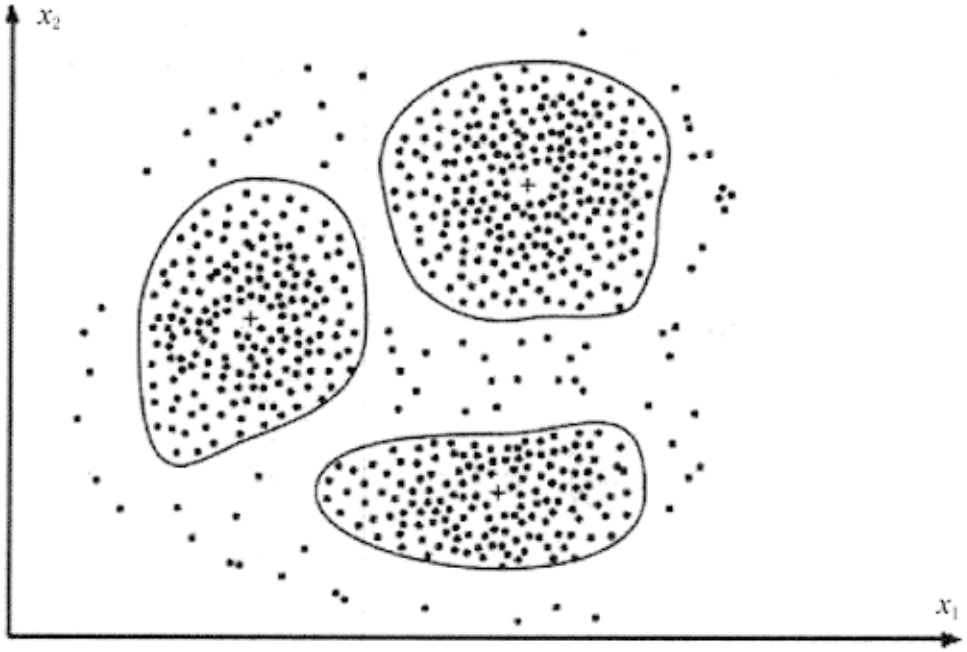


(x, y)

监督信号来自人工标注



- 无监督文本分类
  - 无监督文本分类模型在训练中，样本的正确标签，即监督信号是不可见的
  - 统计模型需要在不依赖于分类标签的情况下优化特定的目标函数已取得理想的分类性能
  - 传统的文档主题模型，如pLSA，LDA等，多是无监督分类方法，本课程不做介绍



文本分类

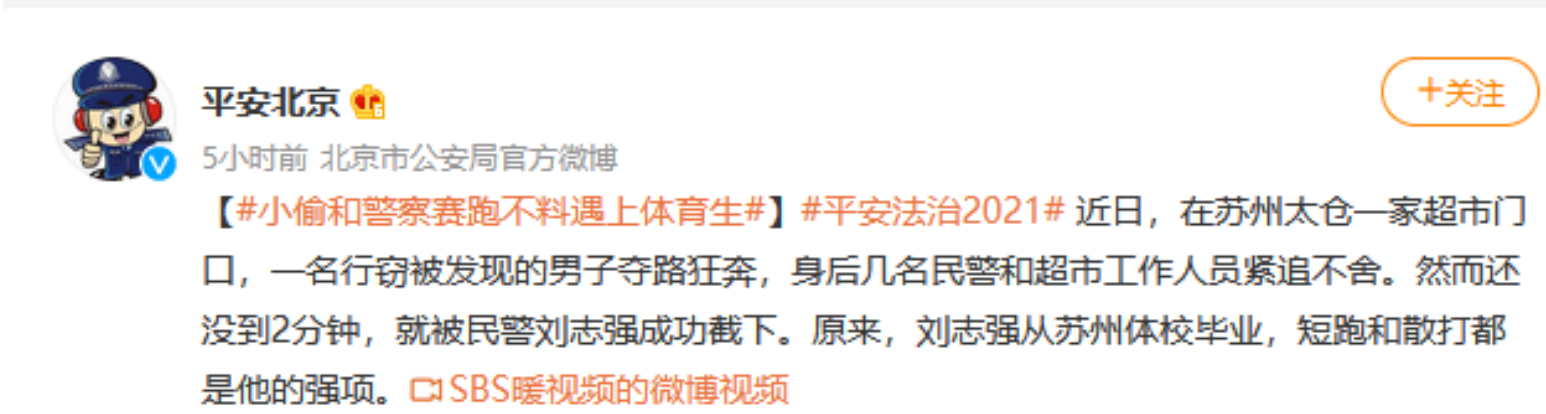
文本分类

任务定义、类型与应用

- 输入：
  - 视具体应用而定的不同类型、不同长度的文本
- 输出：
  - 预先定义的分类标签

- 文本分类可以依照输入的篇幅、分类标签体系的不同、以及应用场景等分类

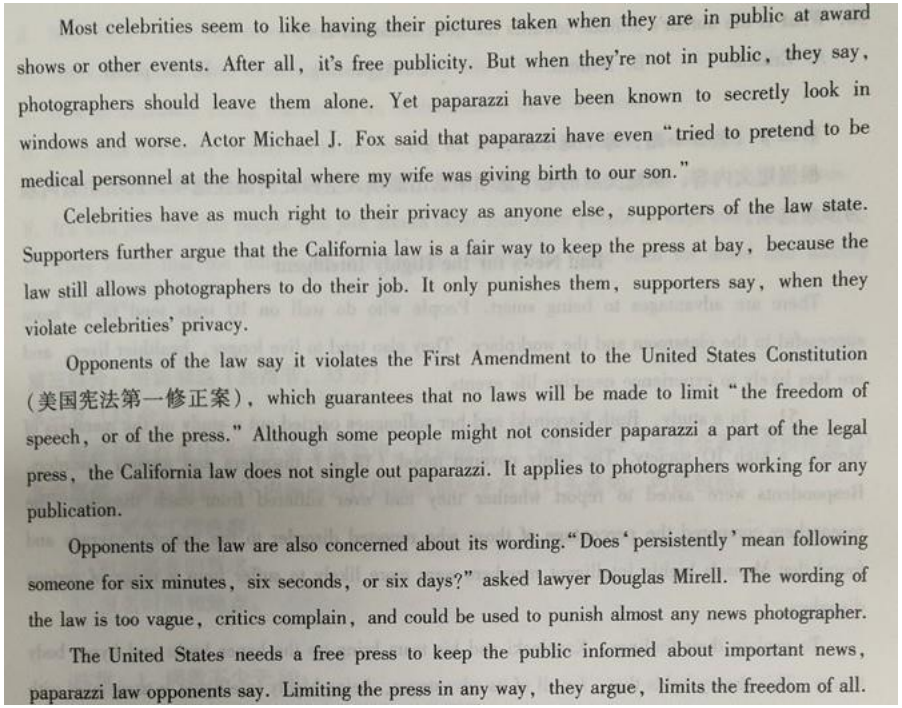
- 依照输入文段的篇幅，文本分类可以大致划分为短、中、长文段的分类
  - 短文段可能包含一句或几句话，如普通微博、推文等



文本分类

任务定义、类型与应用

- 依照输入文段的篇幅，文本分类可以大致划分为短、中、长文段的分类
  - 短文段可能包含一句或几句话，如普通微博、推文等
  - 中等的文段，如英文阅读理解的文章等



Most celebrities seem to like having their pictures taken when they are in public at award shows or other events. After all, it's free publicity. But when they're not in public, they say, photographers should leave them alone. Yet paparazzi have been known to secretly look in windows and worse. Actor Michael J. Fox said that paparazzi have even "tried to pretend to be medical personnel at the hospital where my wife was giving birth to our son."

Celebrities have as much right to their privacy as anyone else, supporters of the law state. Supporters further argue that the California law is a fair way to keep the press at bay, because the law still allows photographers to do their job. It only punishes them, supporters say, when they violate celebrities' privacy.

Opponents of the law say it violates the First Amendment to the United States Constitution (美国宪法第一修正案), which guarantees that no laws will be made to limit "the freedom of speech, or of the press." Although some people might not consider paparazzi a part of the legal press, the California law does not single out paparazzi. It applies to photographers working for any publication.

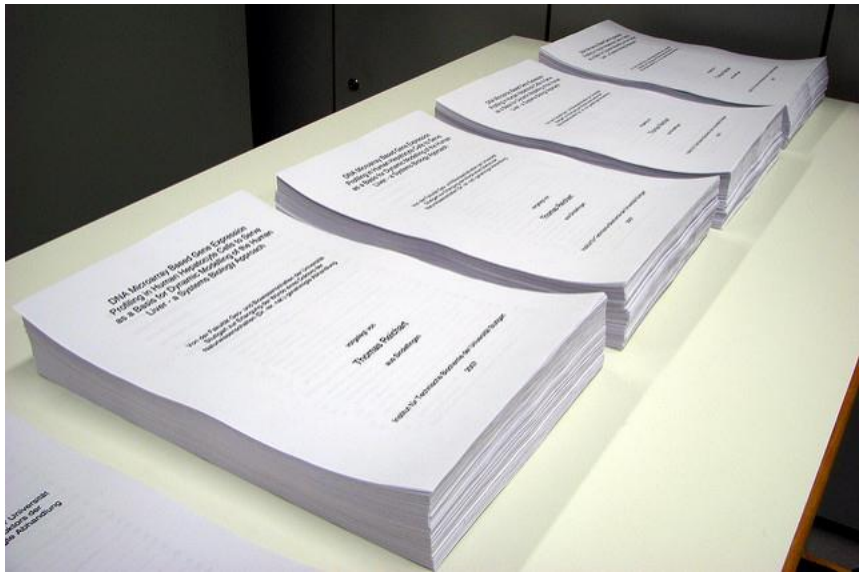
Opponents of the law are also concerned about its wording. "Does 'persistently' mean following someone for six minutes, six seconds, or six days?" asked lawyer Douglas Mirell. The wording of the law is too vague, critics complain, and could be used to punish almost any news photographer.

The United States needs a free press to keep the public informed about important news, paparazzi law opponents say. Limiting the press in any way, they argue, limits the freedom of all.

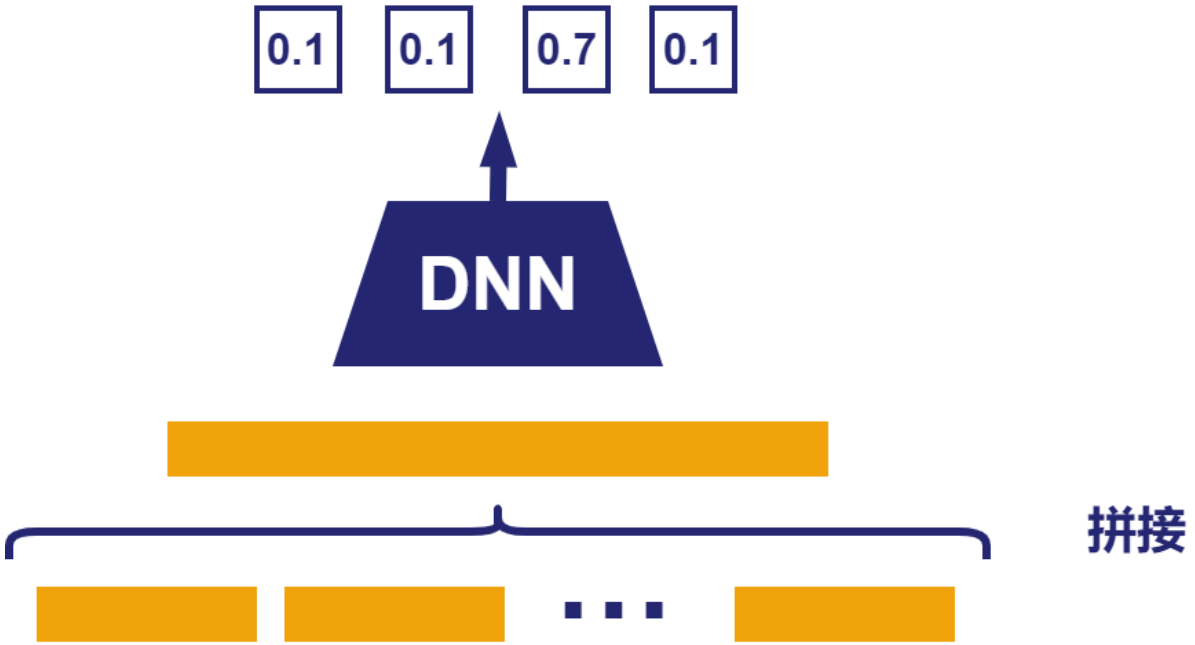
文本分类

任务定义、类型与应用

- 依照输入文段的篇幅，文本分类可以大致划分为短、中、长文段的分类
  - 短文段可能包含一句或几句话，如普通微博、推文等
  - 中等的文段，如英文阅读理解的文章等
  - 长篇的文段，如长篇的学术论文等



- 利用DNN做文本分类



$$\boldsymbol{p} = \text{softmax}(\boldsymbol{W}\,[\boldsymbol{e}_1^\intercal, \boldsymbol{e}_2^\intercal, \ldots, \boldsymbol{e}_n^\intercal]^\intercal + \boldsymbol{b})$$

利用拼接的方式从每个词的词向量生成句子的向量，再利用DNN分类；
代价是拼接得到的句子向量的维度会很高，模型参数量很大

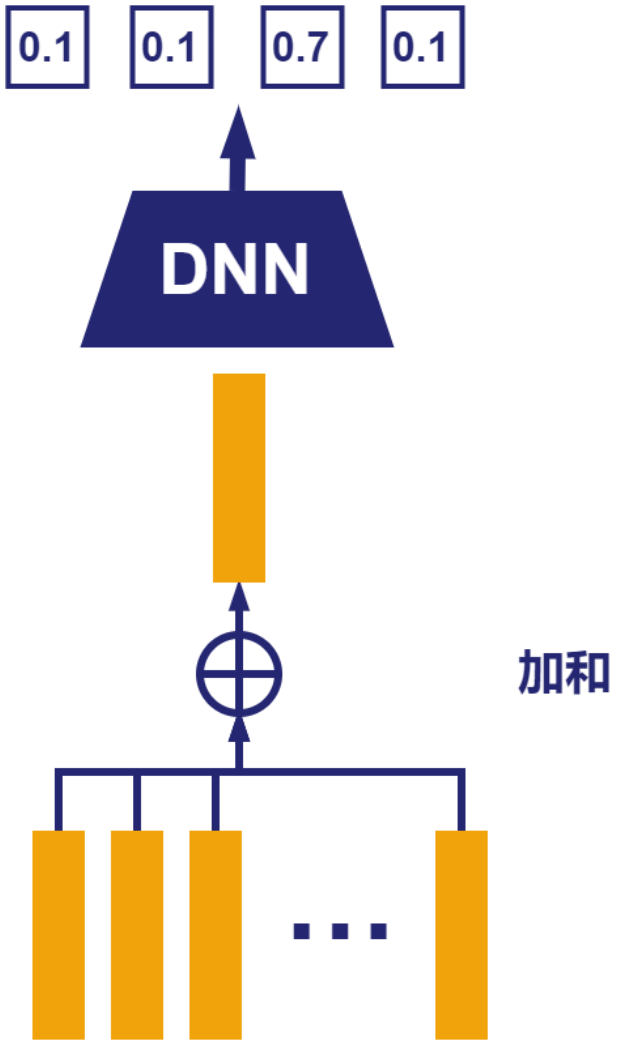
思考：这种结构如何处理不定长的序列?

- 利用DNN做文本分类

$$\boldsymbol{p} = \text{softmax}\left(\boldsymbol{W}\sum_i \boldsymbol{e}_i + \boldsymbol{b}\right)$$

  - 采用词袋模型，可以有效地减少输入维度和模型参数，容易建模各种长度的文本
  - 代价是损失了词序的信息

  - 一些语言中，词序很重要，如：
    - 猫吃鱼。
    - 鱼吃猫。



文本分类

典型模型结构 —— LSTM

- 利用LSTM做文本分类

$$\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n]$$
$$\boldsymbol{E}' = \mathrm{LSTM}(\boldsymbol{E})$$
$$\boldsymbol{E}' = [\boldsymbol{e}'_1, \boldsymbol{e}'_2, \ldots, \boldsymbol{e}'_n]$$
$$\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{e}'_n + \boldsymbol{b})$$

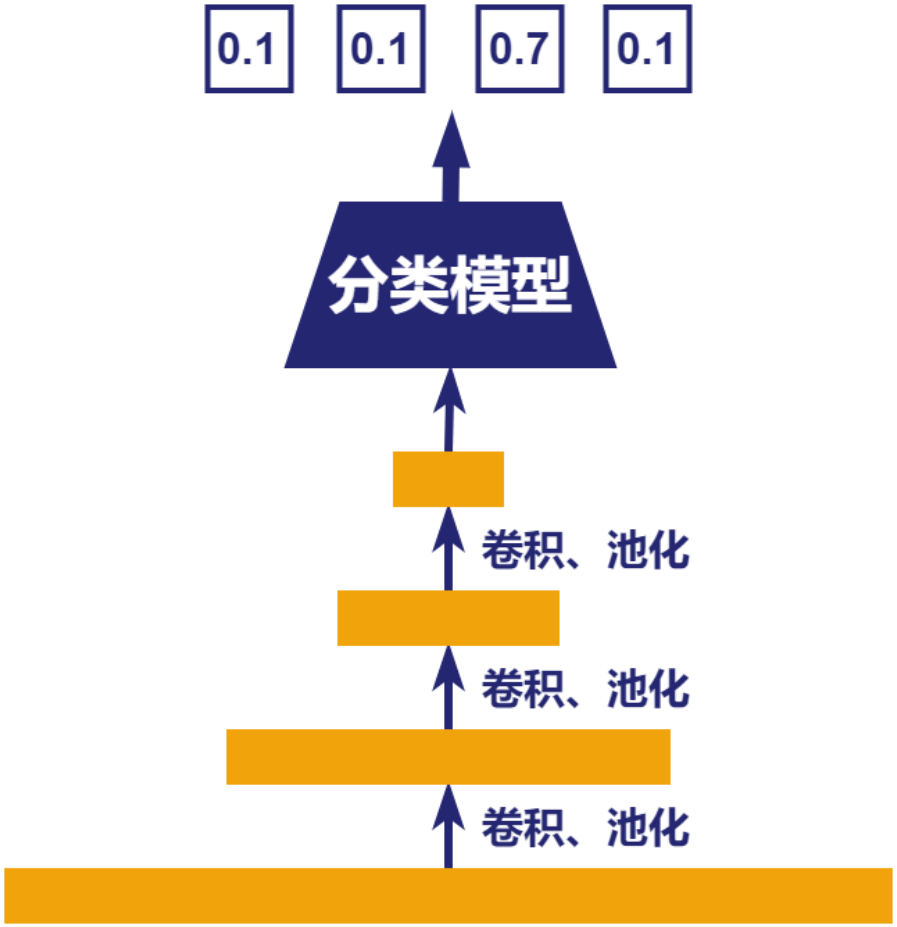
  - LSTM的最后一位输出向量编码了整个词序列的信息，可以用于整段文本的分类

  - LSTM可以在保留顺序信息的情况下，以相对较少的参数量编码各种长度的文段

文本分类

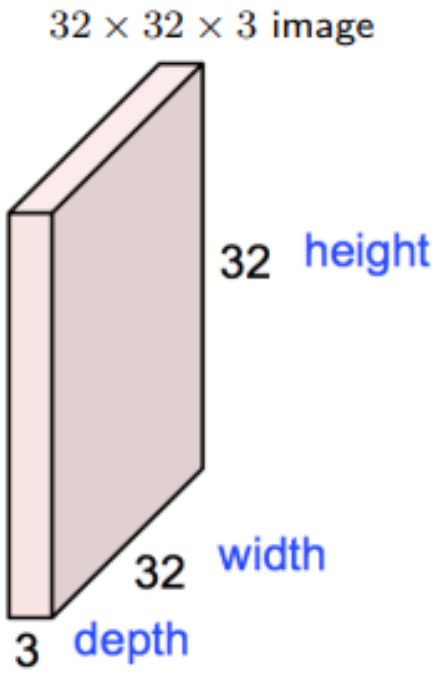
典型模型结构 —— CNN

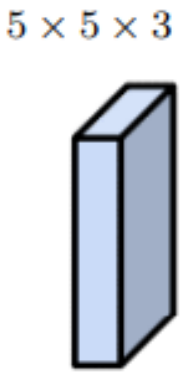
- 利用CNN（一维卷积）编码文本训练并做文本分类



CNN文本分类

卷积层

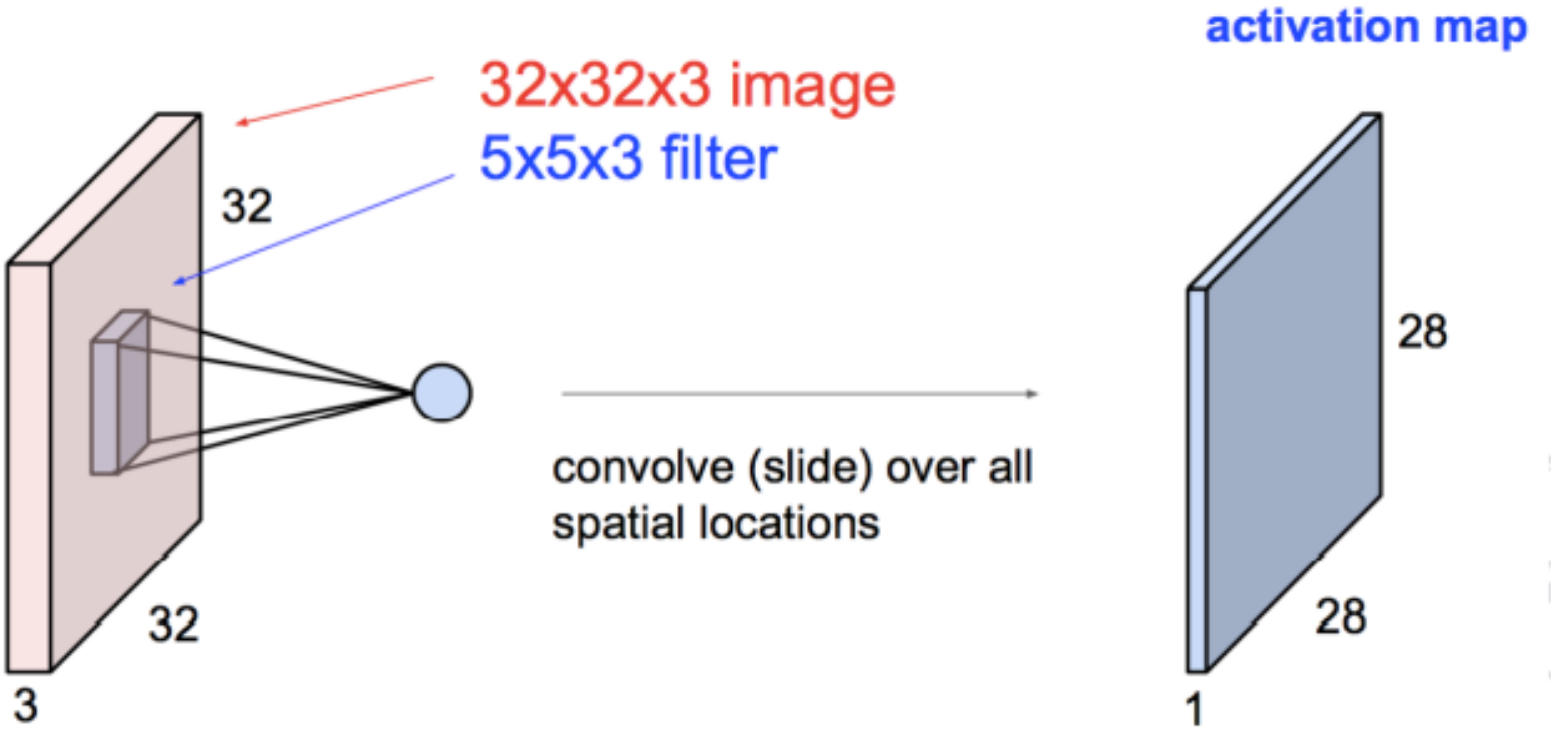




- 将filter与图像进行**卷积convolution**即“在图像上进行空间滑动，计算点积”

CNN文本分类

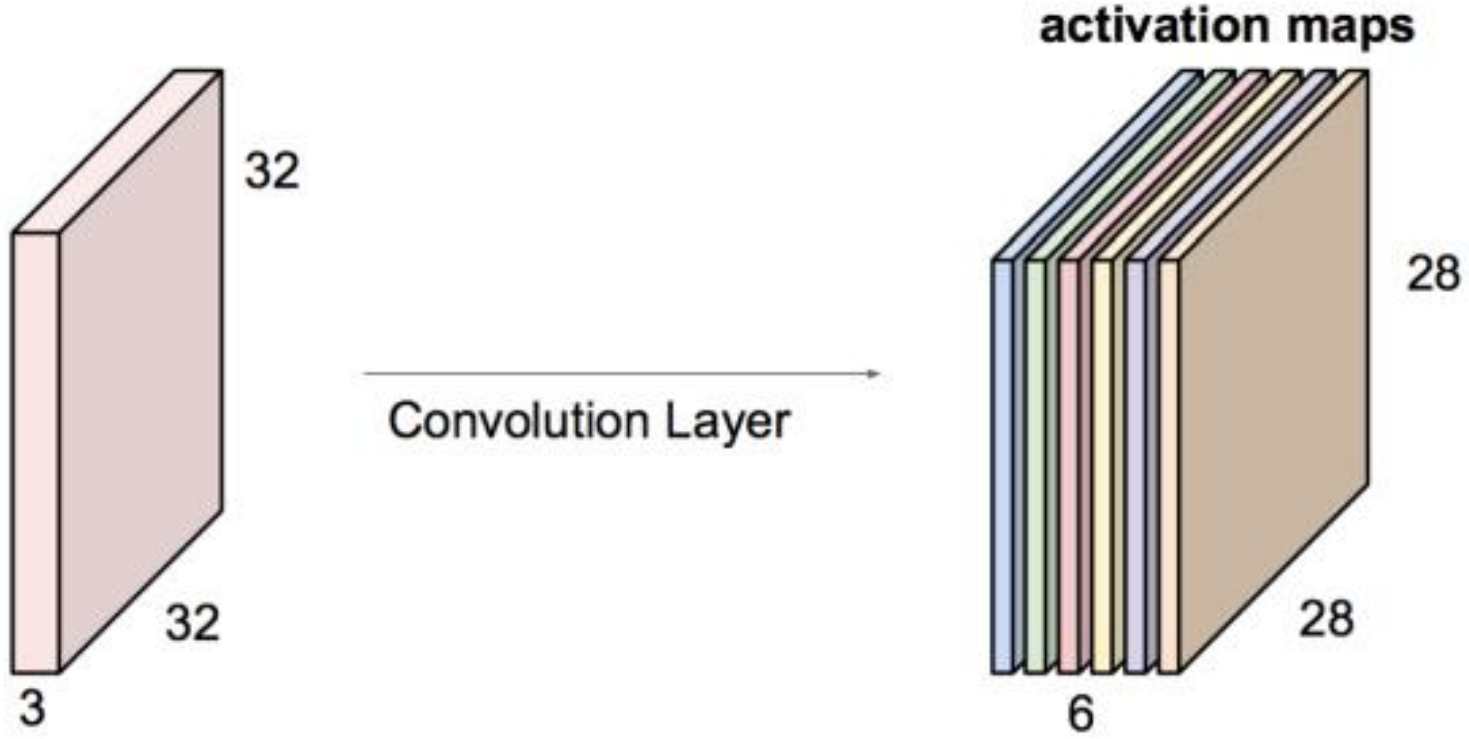
卷积层


activation map
32x32x3 image
5x5x3 filter
32
32
3
convolve (slide) over all
spatial locations
28
28
1

CNN文本分类

卷积层

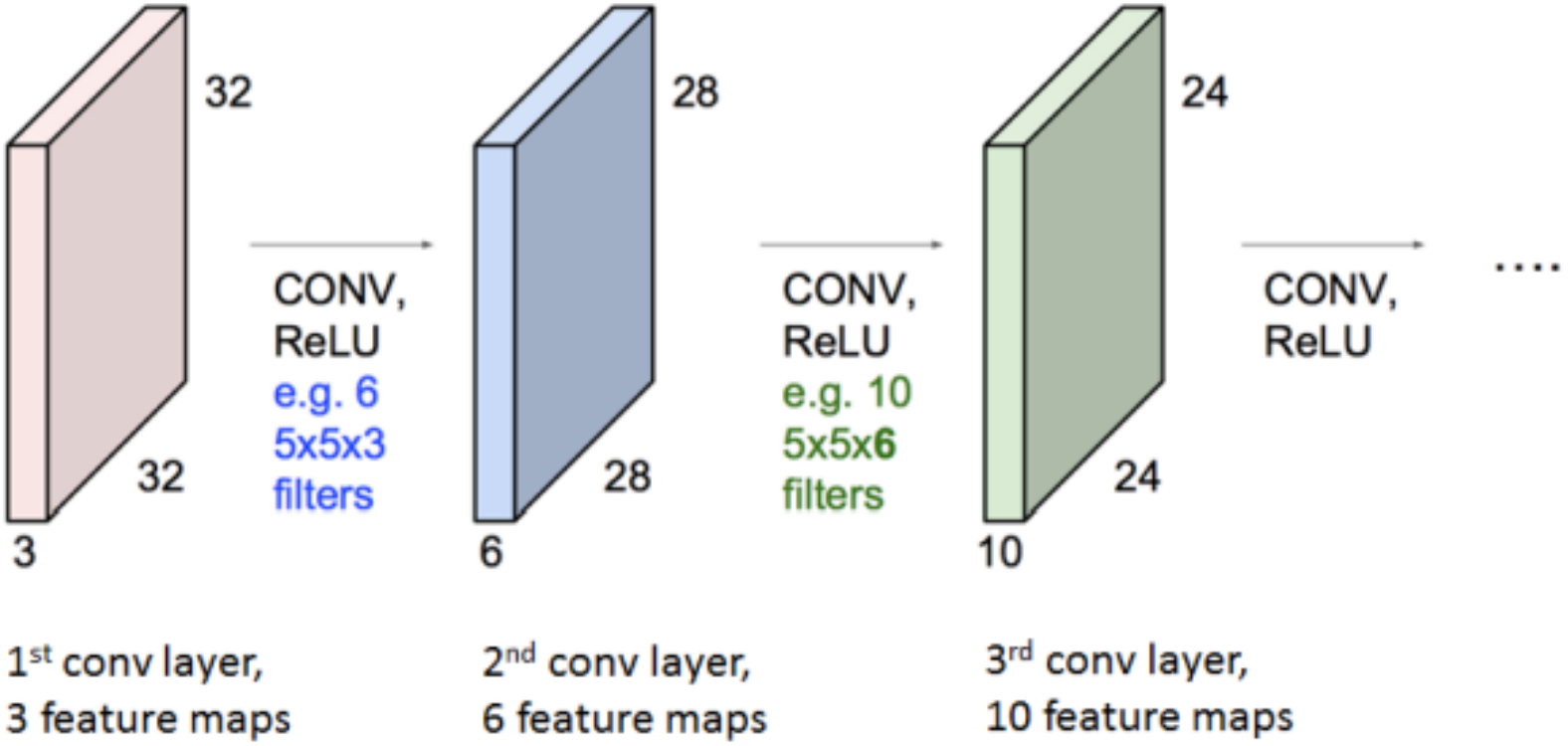
- 对于每个卷积层，最终的特征图数量是个**超参数**，这里就是6



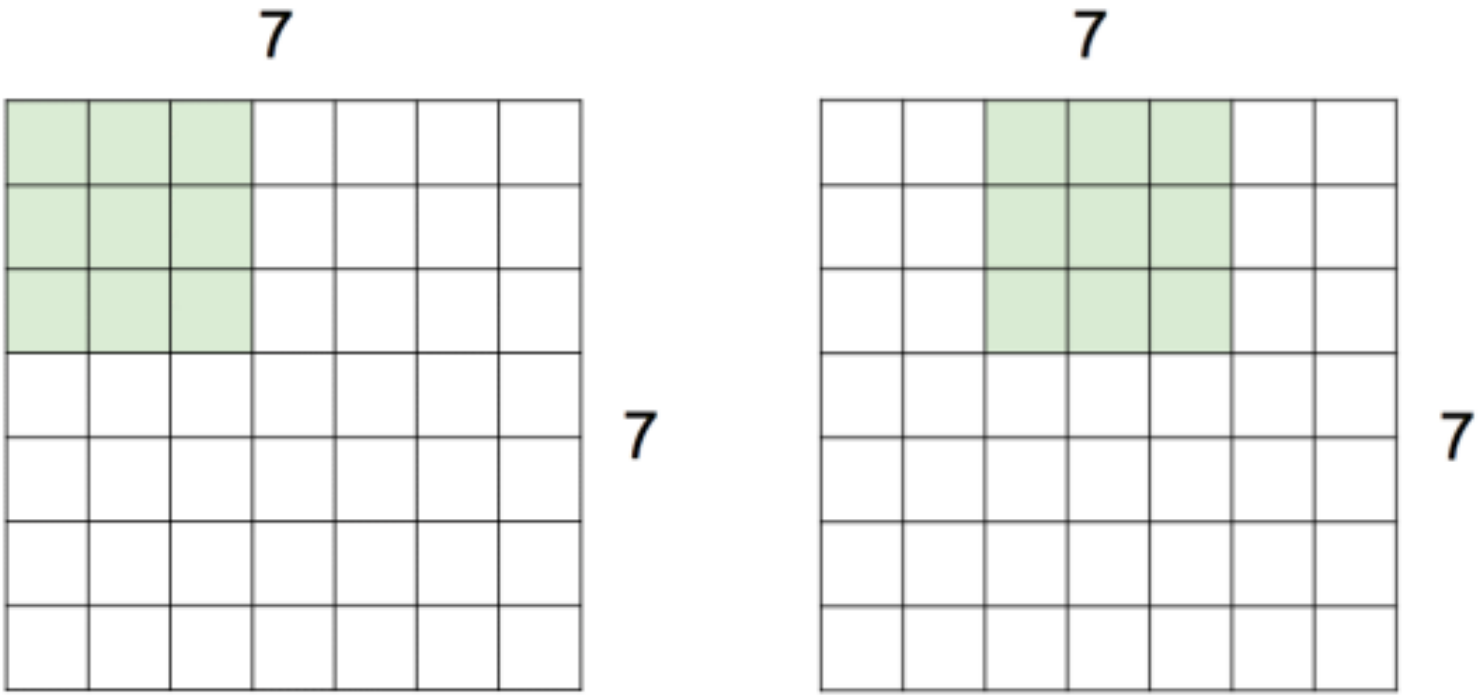
CNN文本分类

卷积层

- 一个卷积层序列，其中卷积后通常使用ReLU激活函数

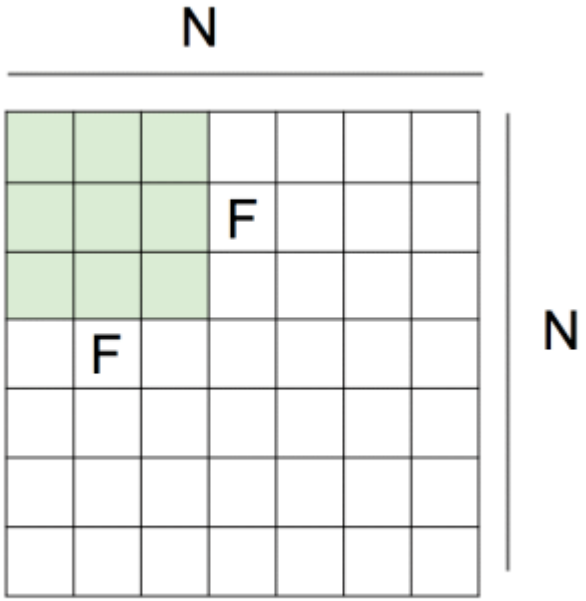


- 另一个**超参数：步长**
  - 下例中7×7的特征图，3×3 filter按步长 $stride = 2$ 与之进行卷积

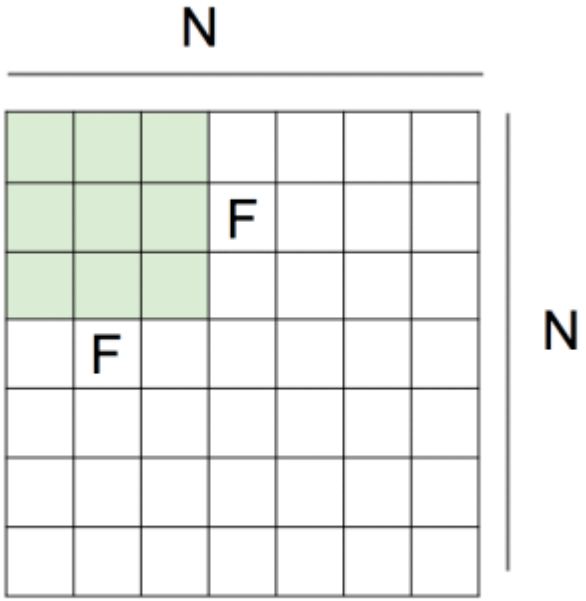


CNN文本分类

卷积层

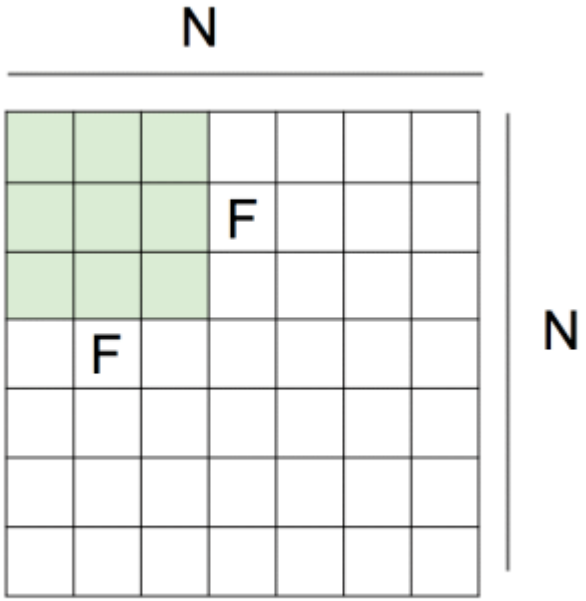


- **问题：**
  - 输出的特征图大小是？



- **问题：**
  - 输出的特征图大小是？

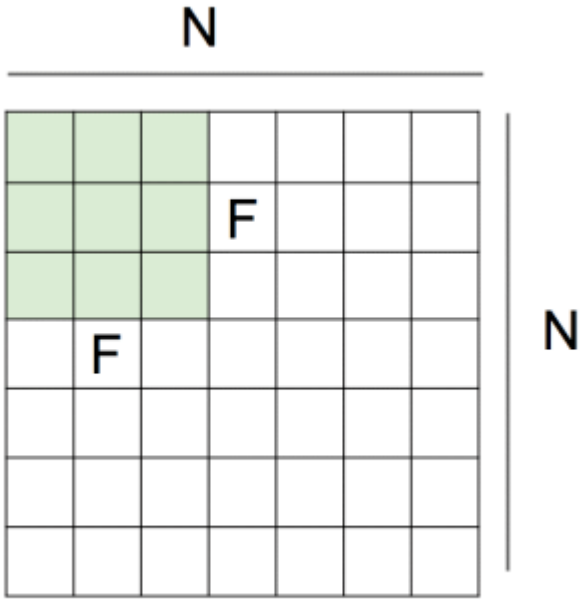
$$\frac{N-F}{stride}+1$$



- **问题：**
  - 输出的特征图大小是？

$$\frac{N-F}{stride}+1$$

  - 不能整除怎么办？



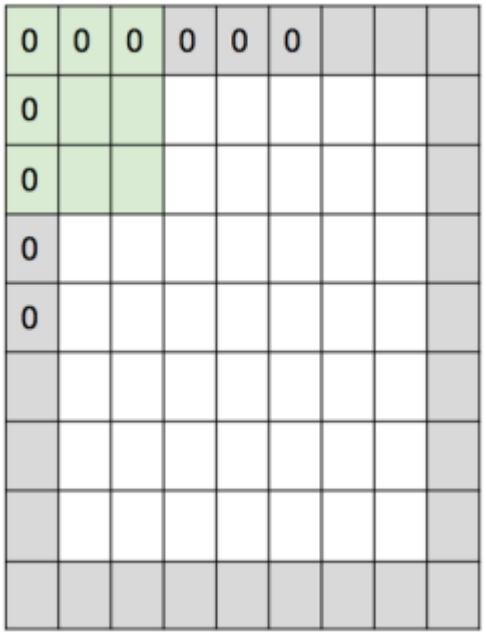
- **问题：**
  - 输出的特征图大小是？

$$\frac{N-F}{stride}+1$$

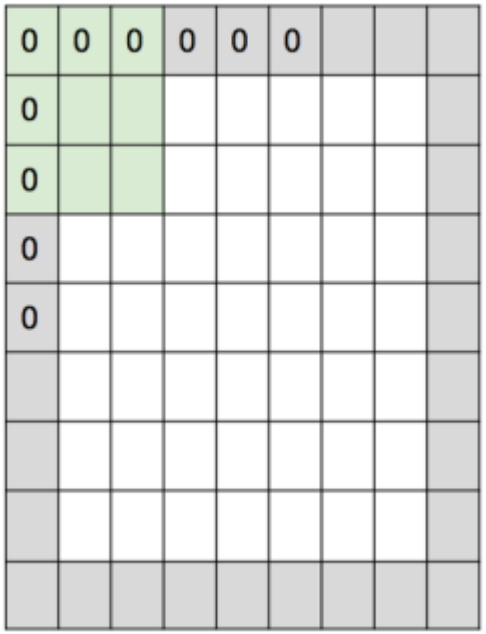
  - 不能整除怎么办？
  - 能否避免？

CNN文本分类

卷积层



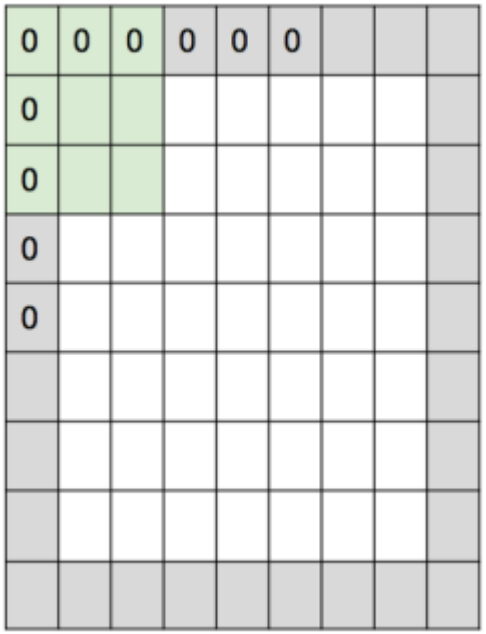
- **问题：**
  - $stride = 1$时，如果在边界处补一个0会发生什么？



- **问题：**
  - $stride = 1$时，如果在边界处补一个0会发生什么？
  - 输入输出大小相同了！

CNN文本分类

卷积层



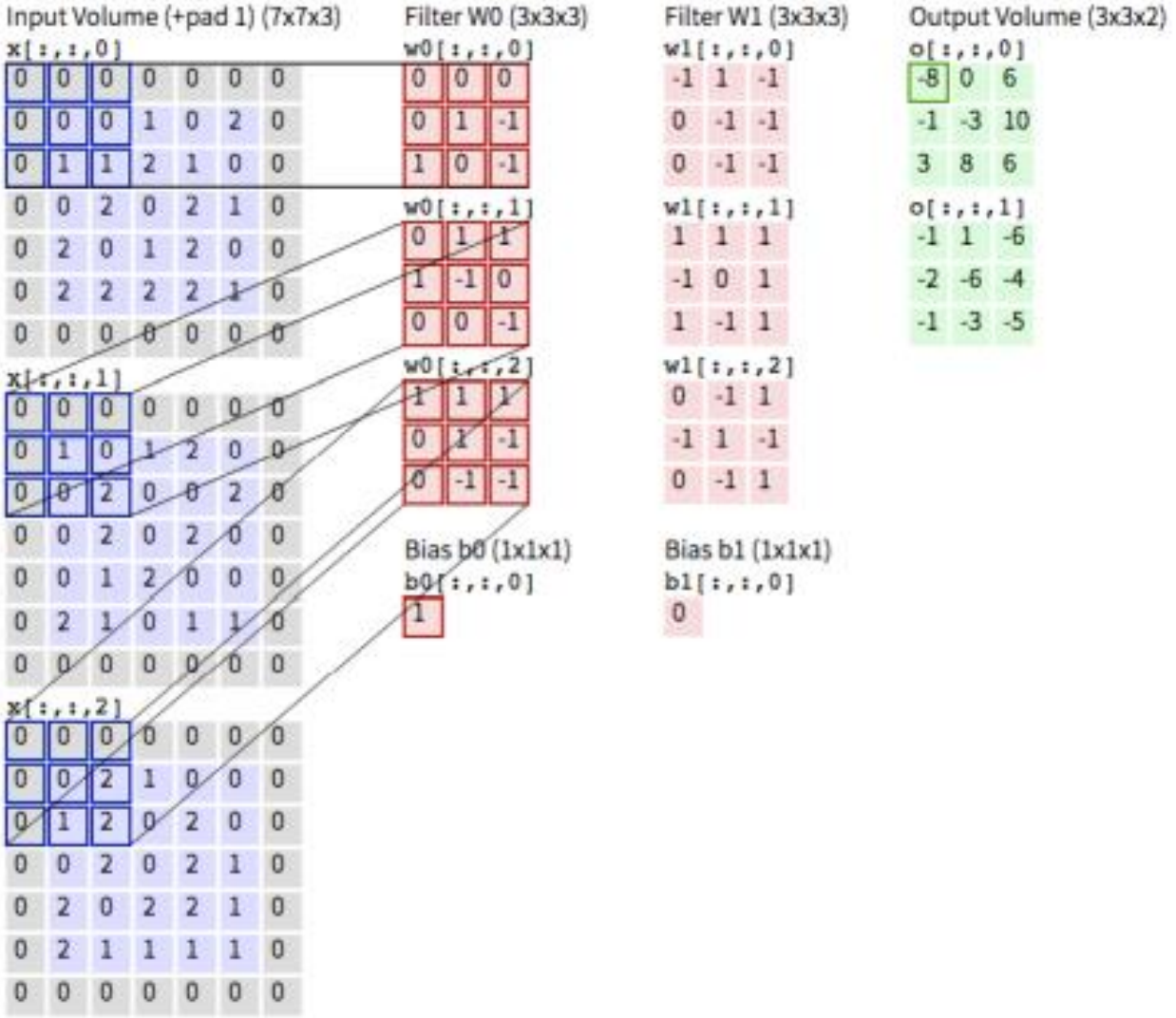
- **卷积前使用补0是一个常见操作。**
  - 保留了特征图的分辨率
  - 更好的利用了边界信息，从而提升性能

CNN文本分类

卷积层

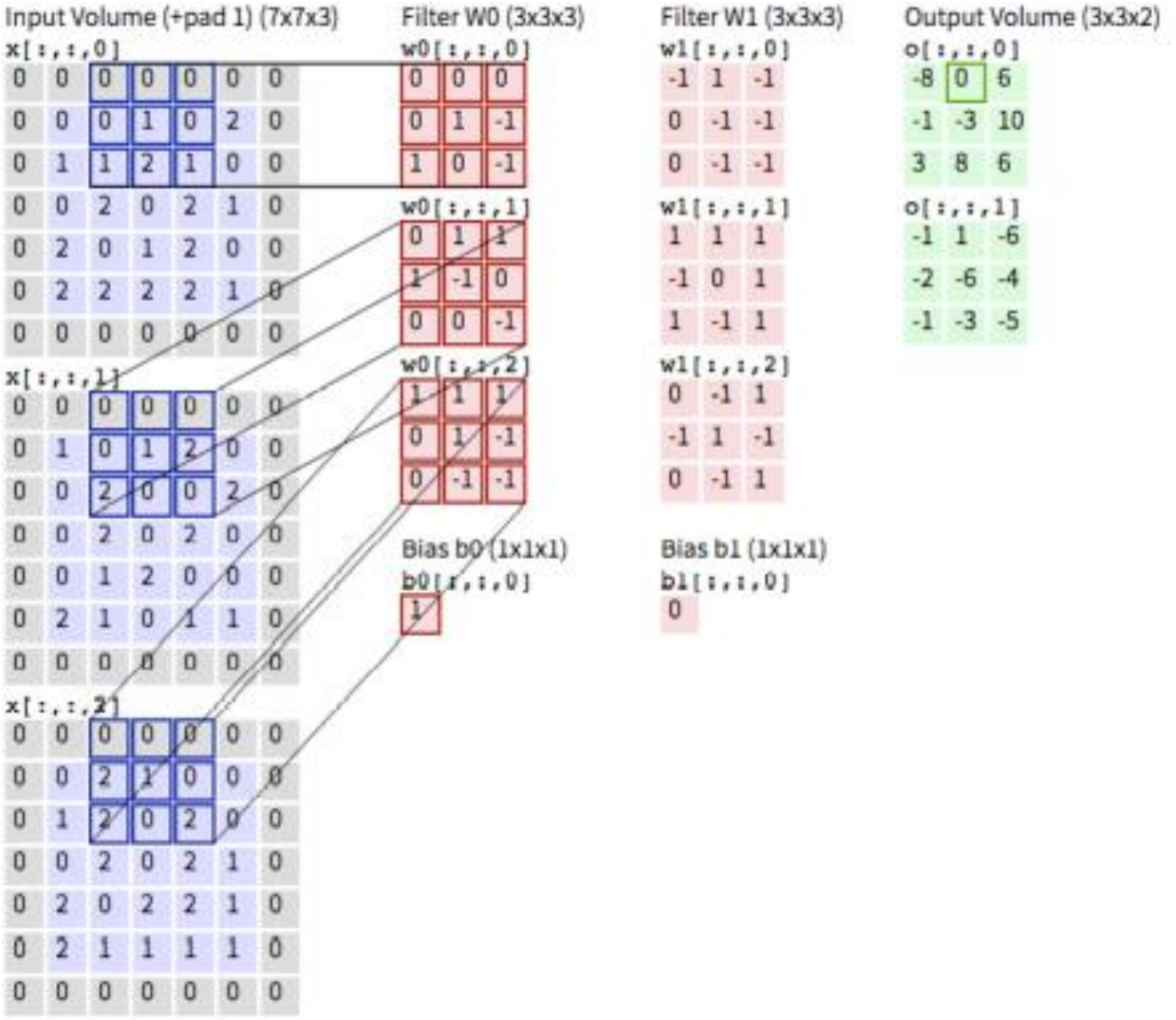
一个生动的 例子:

- $5 \times 5 \times 3$ 的输入图像 (3 个 5×5 的特征图)
- 和2 个 $3 \times 3 \times 3$ filter
- 那么得到2 个 3×3 的输出特征图
- $\text{Stride} = 2$
- $\text{pad} = 1$



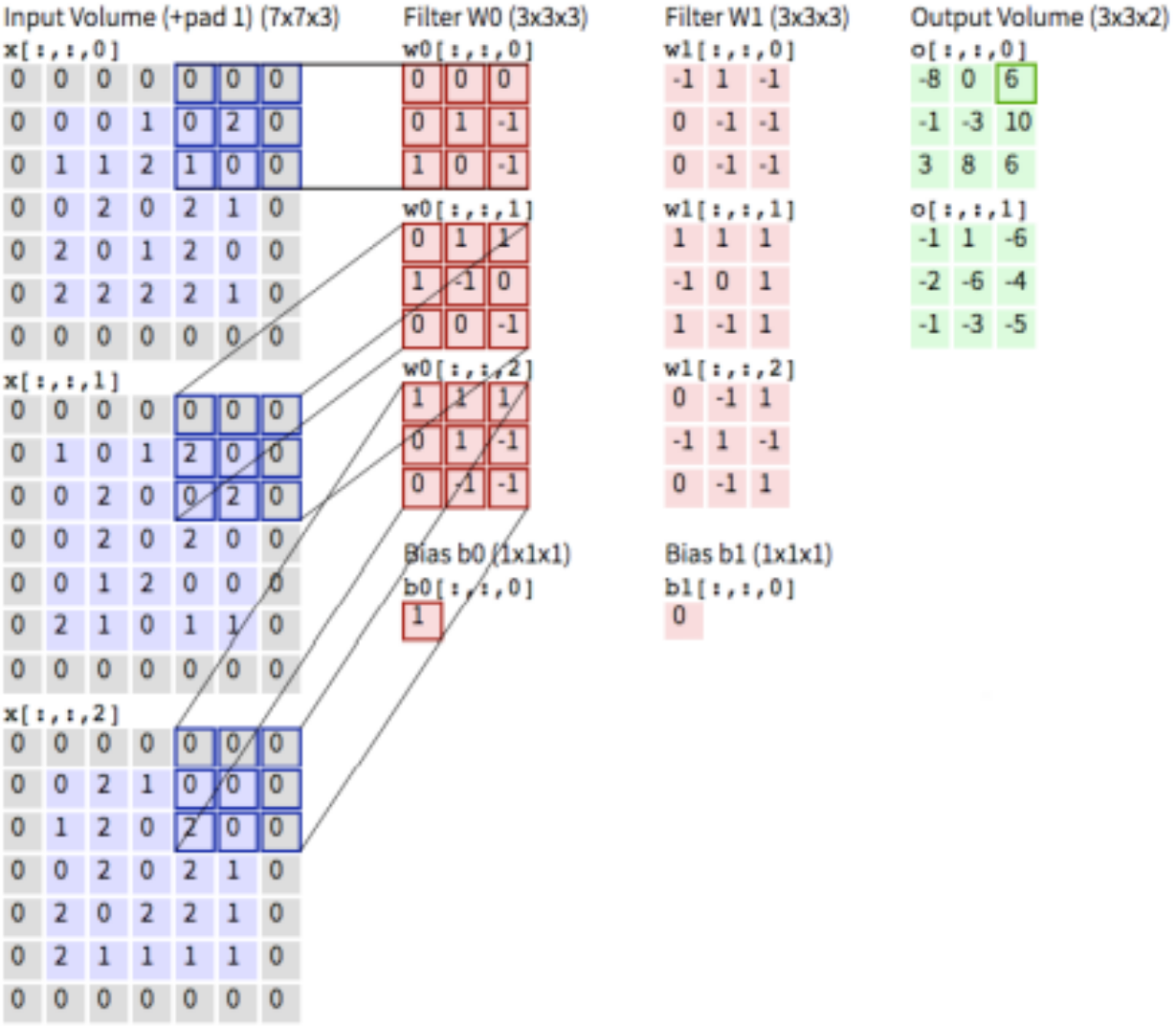
CNN文本分类

卷积层


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
x[:,:,1]
x[:,:,2]
Filter W0 (3x3x3)
w0[:,:,0]
w0[:,:,1]
w0[:,:,2]
Bias b0 (1x1x1)
b0[:,:,0]
Filter W1 (3x3x3)
w1[:,:,0]
w1[:,:,1]
w1[:,:,2]
Bias b1 (1x1x1)
b1[:,:,0]
Output Volume (3x3x2)
o[:,:,0]
o[:,:,1]

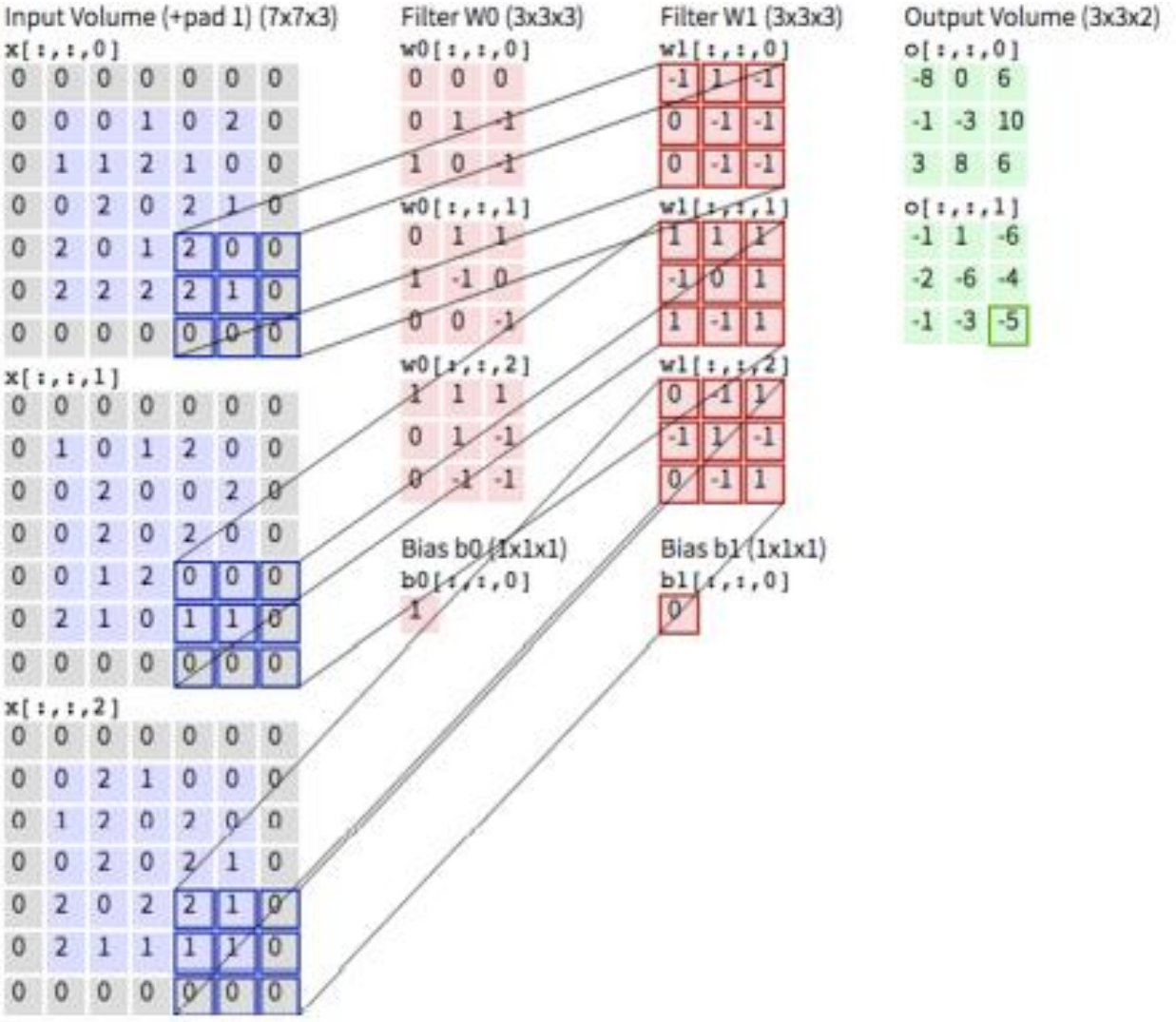
CNN文本分类

卷积层


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 0 1 0 2 0
0 1 1 2 1 0 0
0 0 2 0 2 1 0
0 2 0 1 2 0 0
0 2 2 2 2 1 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 0 1 2 0 0
0 0 2 0 0 2 0
0 0 2 0 2 0 0
0 0 1 2 0 0 0
0 2 1 0 1 1 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 0 2 1 0 0 0
0 1 2 0 2 0 0
0 0 2 0 2 1 0
0 2 0 2 2 1 0
0 2 1 1 1 1 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 0 0
0 1 -1
1 0 -1
w0[:,:,1]
0 1 1
1 -1 0
0 0 -1
w0[:,:,2]
1 1 1
0 1 -1
0 -1 -1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
-1 1 -1
0 -1 -1
0 -1 -1
w1[:,:,1]
1 1 1
-1 0 1
1 -1 1
w1[:,:,2]
0 -1 1
-1 1 -1
0 -1 1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
-8 0 6
-1 -3 10
3 8 6
o[:,:,1]
-1 1 -6
-2 -6 -4
-1 -3 -5

CNN文本分类

卷积层


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 0 1 0 2 0
0 1 1 2 1 0 0
0 0 2 0 2 1 0
0 2 0 1 2 0 0
0 2 2 2 2 1 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 0 1 2 0 0
0 0 2 0 0 2 0
0 0 2 0 2 0 0
0 0 1 2 0 0 0
0 2 1 0 1 1 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 0 2 1 0 0 0
0 1 2 0 2 0 0
0 0 2 0 2 1 0
0 2 0 2 2 1 0
0 2 1 1 1 1 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 0 0
0 1 1
1 0 1
w0[:,:,1]
0 1 1
1 -1 0
0 0 -1
w0[:,:,2]
1 1 1
0 1 -1
0 -1 -1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
-1 1 -1
0 -1 -1
0 -1 -1
w1[:,:,1]
1 1 1
-1 0 1
1 -1 1
w1[:,:,2]
0 -1 1
-1 1 -1
0 -1 1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
-8 0 6
-1 -3 10
3 8 6
o[:,:,1]
-1 1 -6
-2 -6 -4
-1 -3 -5

CNN文本分类

一维序列上的卷积

- 一维CNN卷积
  - 卷积核以小窗口的形式在输入序列上移动，每个位置产生一个输出
  - 对窗口内的向量序列

$$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n],$$

$$\boldsymbol{a}_i = [a_{1i}, a_{2i}, \ldots, a_{di}]^\mathsf{T} \in \mathbb{R}^d$$

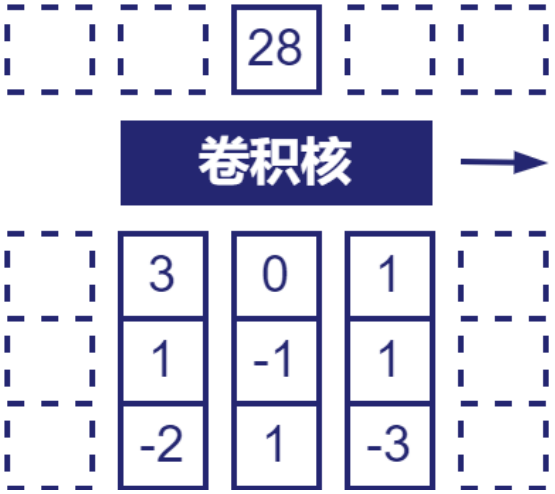
单输出通道的卷积核 $\boldsymbol{B} = (b_{ij})_{d \times n}$

卷积层运算为

$$\text{Conv1d}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{d} \sum_{j=1}^{n} a_{ij} b_{ij}$$

2	1	0
4	-2	2
-1	3	-3

单输出通道的卷积核



单输出通道的卷积层

CNN文本分类

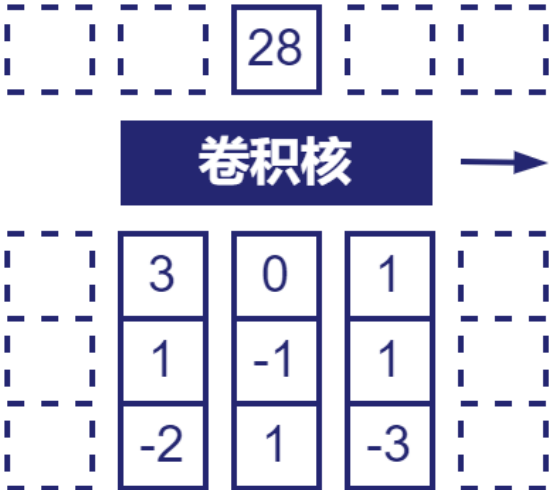
一维序列上的卷积

$$\text{Conv1d}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{d} \sum_{j=1}^{n} a_{ij} b_{ij}$$

2	1	0
4	-2	2
-1	3	-3

单输出通道的卷积核

$$\begin{aligned} 28 &= 2 \times 3 + 1 \times 0 + 0 \times 1 \\ &= +4 \times 1 + (-2) \times (-1) + 2 \times 1 \\ &= +(-1) \times (-2) + 3 \times 1 + (-3) \times (-3) \end{aligned}$$



单输出通道的卷积层

- 理解卷积层运算

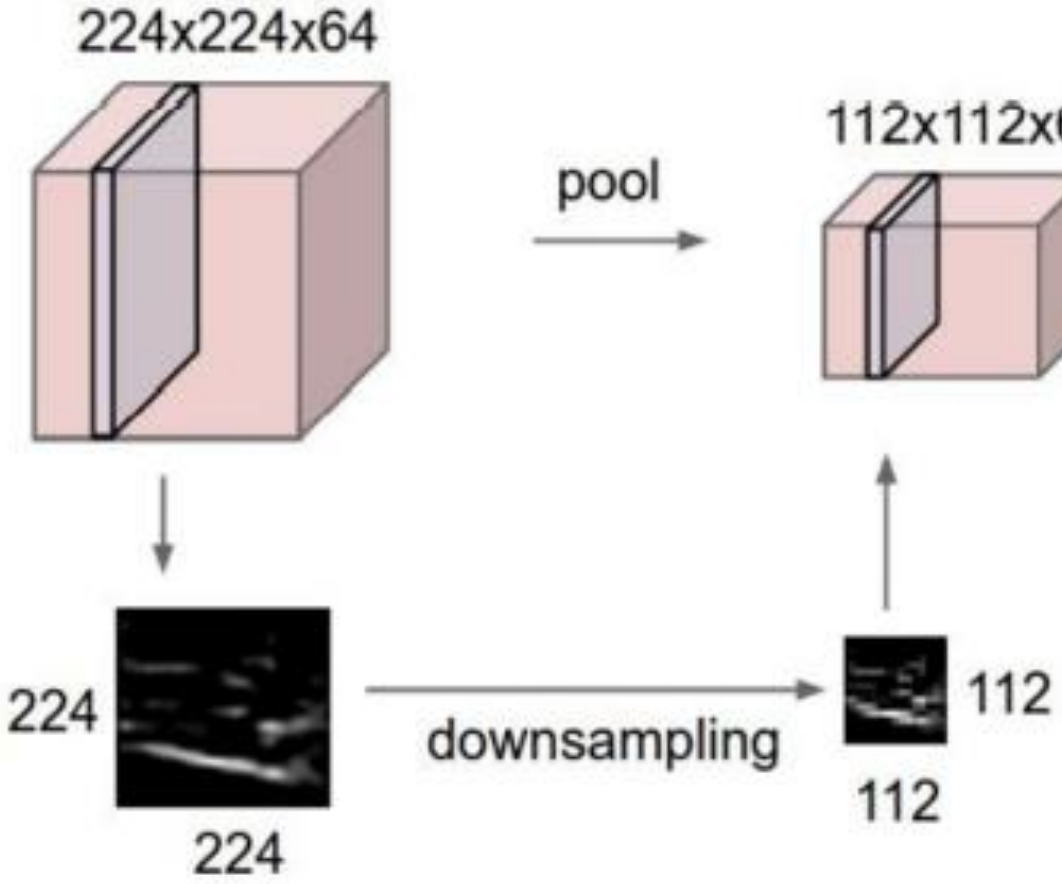
$$\text{Conv1d}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{d} \sum_{j=1}^{n} a_{ij} b_{ij}$$

  - 卷积层可以看作是两个向量序列的相关运算，或者理解成两个向量的内积运算
  - 余弦相似度：$\cos\theta = \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\| \|\boldsymbol{b}\|}$
  - 在相同的模长下，内积度量了两个向量的方向的相关性
  - 卷积层的输出值越大，说明窗口内的向量序列和卷积核越相似，每个通道的卷积核都可以看作特定的“模式提取器”，用来抽取输入序列中特定的局部模式，得到可以描述输入序列的某维度特征

CNN文本分类

池化层

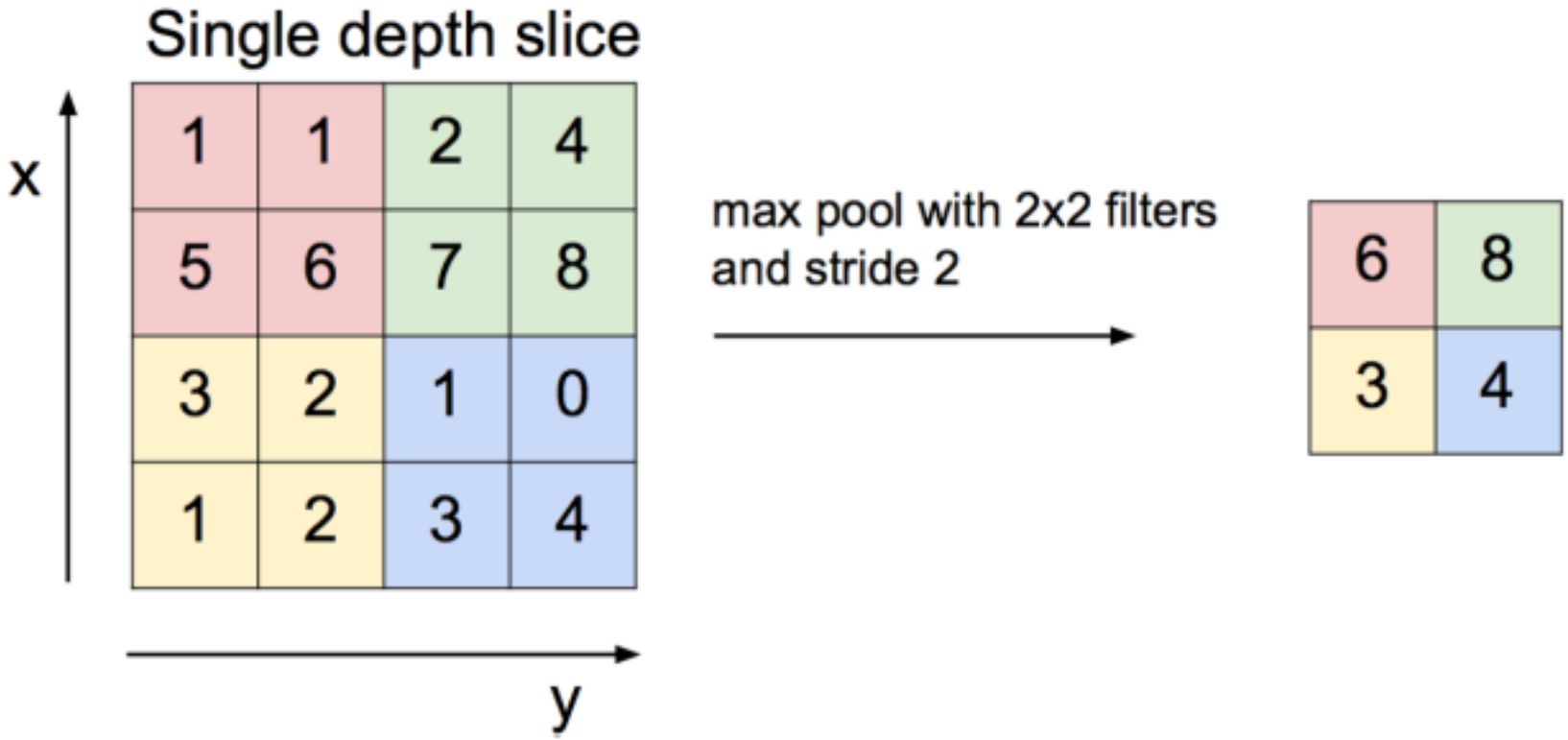
- 使特征表示（Representation）更小、更容易管理
- 在每个激活图上单独运算



CNN文本分类

池化层

- 这里，池化大小（filter size）为2×2，步长（stride）也是2×2，因此池化没有重叠（惯例）

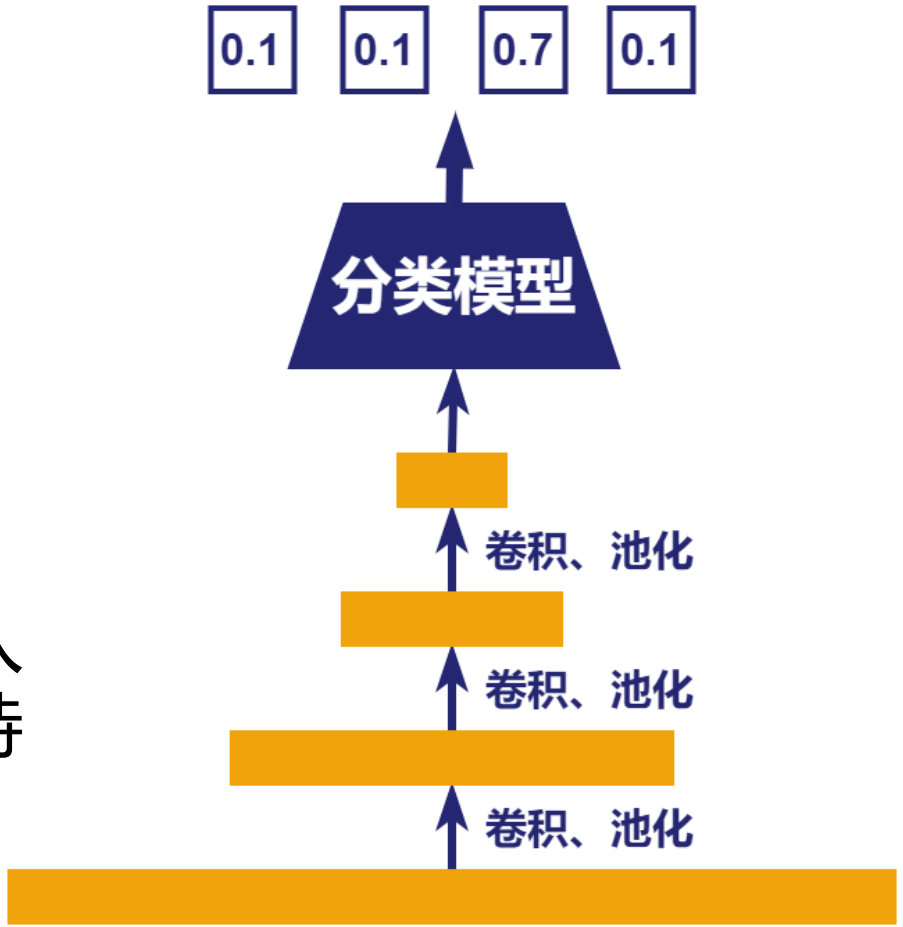


- 常见的池化操作：最大、平均、最小、随机、固定

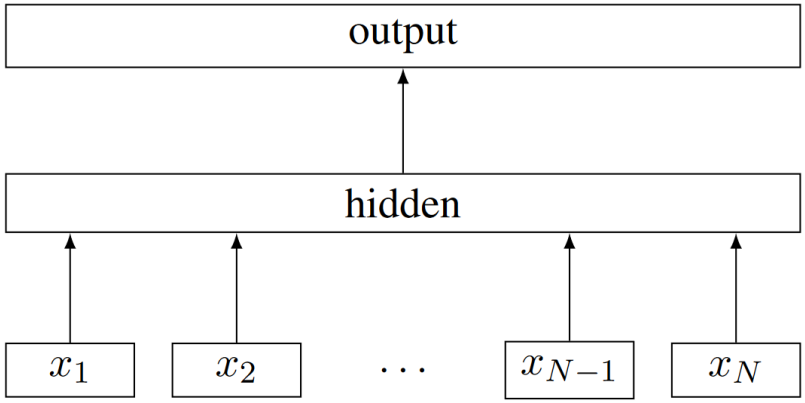
CNN文本分类

由二维到一维

- 卷积核为$1 \times F_c \times D$
  - F_c为卷积窗口大小
  - D为特征向量维度
- 池化大小为$1 \times F_p$
  - F_p为池化窗口大小
- 卷积、池化操作后，原输入文本被编码为长度缩减的特征序列，可用LSTM或者DNN继续处理



- FastText
  - 与我们介绍过的词袋DNN模型的主要区别在于模型的输入不是单个词的词向量，而是n-gram词组的向量表示，从而在一定程度上保留了局部的词序信息



图自[Armand Joulin et al., 2017]

  - 猫吃鱼。⟶ (“[BOS]猫”, “猫吃”, “吃鱼”, “鱼[EOS]”)

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, *Bag of Tricks for Efficient Text Classification*, EACL 2017.

情感分类

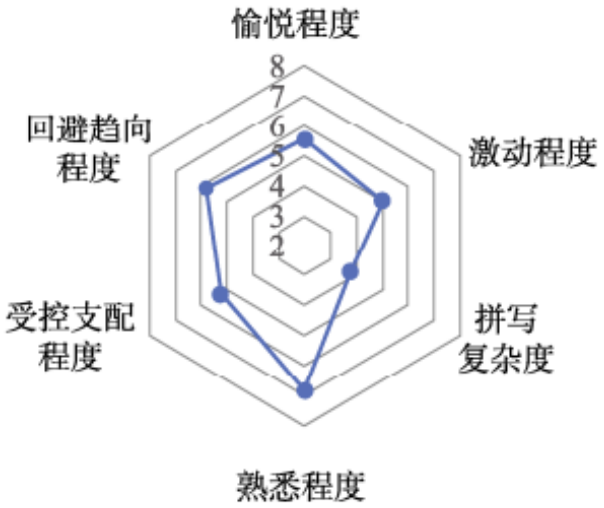
情感分类

- **极性分析用于判断一句话或一段言论的情感极性，多用于分析、筛选用户的评论、反馈以及网络舆论等**
  - 输入：用户的评论或社交媒体的发言等，通常比较短，仅包含若干句话
  - 模型：通常需要基于极性词库，提取相关特征，再使用分类器分类；构建合适的极性词库也是重要的任务
  - 输出：情感类别标签

情感分类

情感类别标签

- **极性分类：**
  - “正” “负”双极性分类
  - “正” “负” “中”三极性分类
- **更具体的情绪分类：**
  - 欢乐悲伤、愤怒恐惧、坚定怀疑、惊喜平静等
- **情绪强度分类：**
  - 情感标签+0-9强度标签
- **更综合的分析：**
  - 图自[袁加锦等. 2021]



袁加锦, 张祎程, 陈圣栋, 罗利, 茹怡珊. (2021). 中国情绪调节词语库的初步编制与试用. 心理学报, 53(5), 445-455.

情感分类

- **简单的模型**
  - 简单的情感分类可以利用词袋模型，通过统计文本中出现的正负极性词的比例来实现
  - This restaurant is fantastic. So gorgeous decoration and meticulous service! I felt I’m a true nobility and really like it.

情感分类

- **更精细的模型**
  - This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up.
  - 更精细的情感分类模型还应该建模文本中的其他的句式、结构等特征，如：否定词、连接词、反问、转折、让步、假设、虚拟语气等
  - 可直接利用LSTM、CNN等实现数据驱动的情感分类

文本匹配

文本匹配

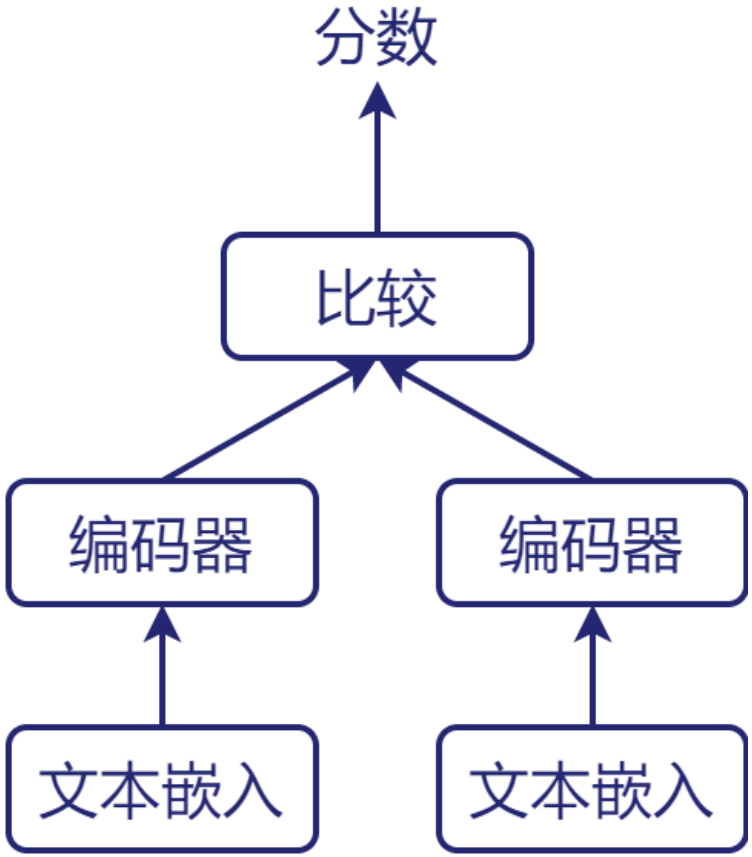
任务定义、类型与应用

- **文本匹配**
  - 输入：两段文本内容
  - 输出：两段文本是否匹配，或两段文本相似度的度量
- **应用**
  - **问答系统**
    - 可以利用文本匹配技术，将用户的问题映射到问答对库中的标准问题，从而构建问答系统
  - **机器阅读理解**
    - 利用模型计算答案选项和问题以及文本的匹配程度，选出正确的选项
  - **文书、工单的检索**
    - 从历史档案中检索相似的文书和工单，为处理人员提供参考

文本匹配

典型模型结构

- 基于表示的模型
  - 先利用编码器将输入映射到相同的向量空间，再比较得到的特征，计算相似度
  - 这种结构不要求采用相同的编码器：
    - 具有不同的语义身份，如一个是查询的关键词，一个是要匹配的文章
    - 具有不用的模态，如图像、视频、音频与文本做匹配



文本匹配

典型模型结构

- 编码器可以采用在文本分类中介绍过的各种编码器结构
- 比较阶段要计算两份输入的编码表示的相似度或距离：
  - 余弦相似度

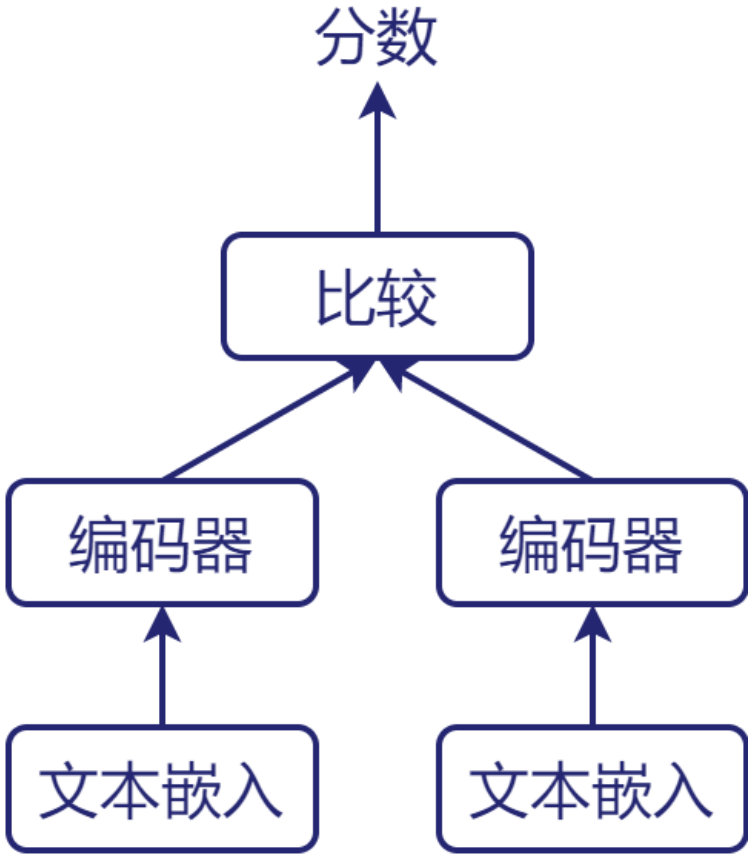
$$\cos\theta = \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$$

  - 二范数（欧几里得距离）

$$\|\boldsymbol{a} - \boldsymbol{b}\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

  - 一范数（曼哈顿距离）

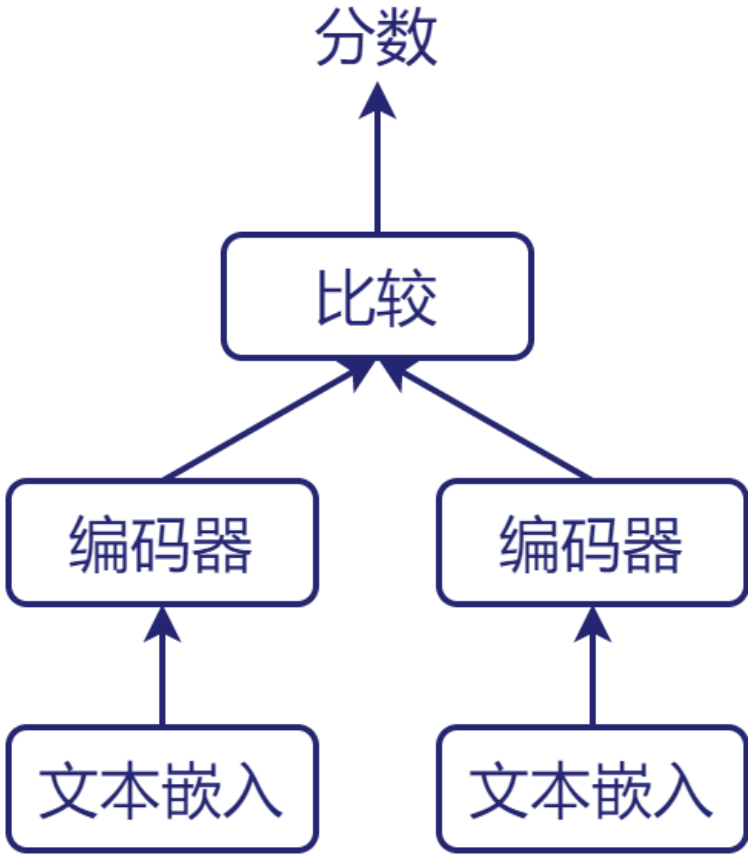
$$\|\boldsymbol{a} - \boldsymbol{b}\|_1 = \sum_i |a_i - b_i|$$



- 损失函数的一般形式

$$\mathcal{L} = \sum_{p^+} \left(l(p^+) - \sum_{p^- \in \mathcal{N}(p^+)} l(p^-) \right)$$

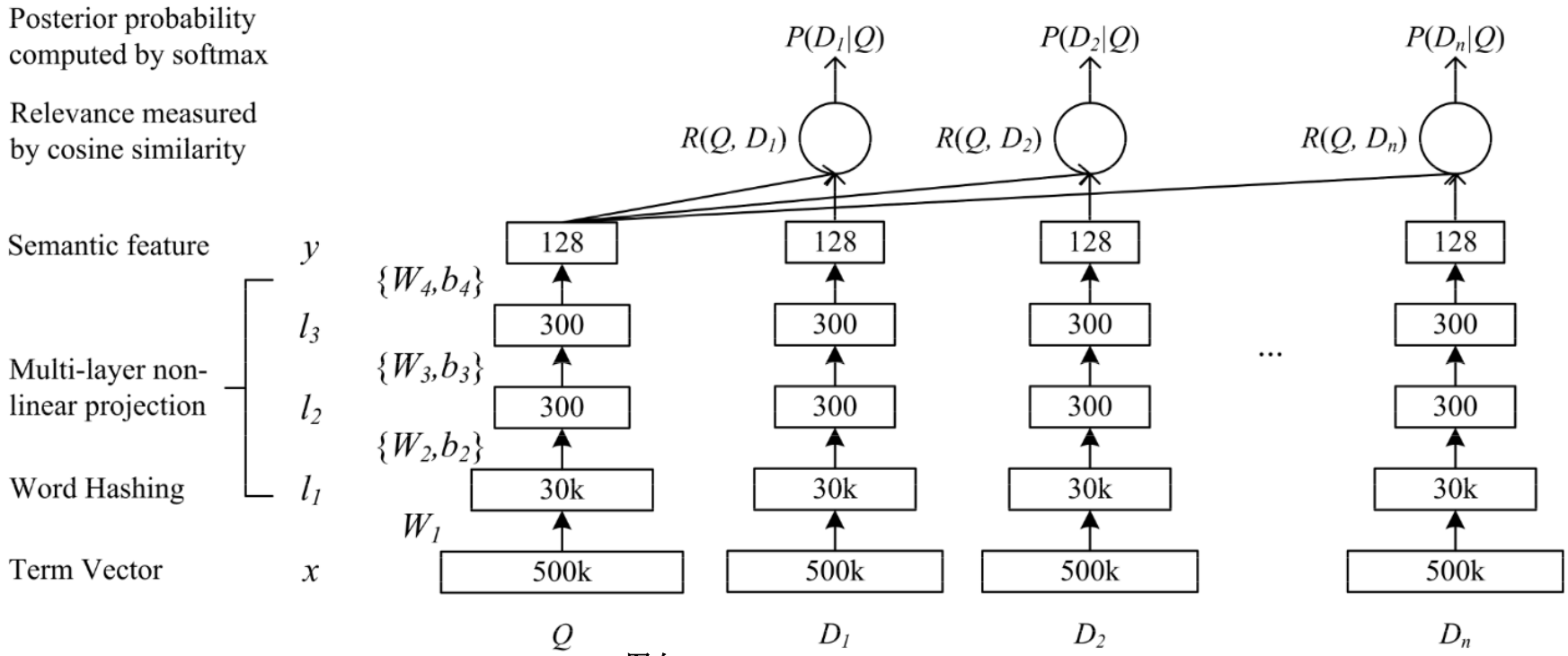
  - 其中p^+ 指数据集中的正样例，即匹配的样本对
  - p^- 为根据p^+构造的负样例，一般可以通过将匹配样本对中的某样本替换为随机样本得到
  - l 为单样本对上的损失，一般就是向量的距离或负的相似度



文本匹配

典型模型结构

- DSSM (Deep Structured Semantic Model)
  - 基于DNN提取的特征分别计算查询 Q 与各正负匹配样本的余弦相似度，然后根据余弦相似度计算后验概率
  - 通过这种方法可以学习到好的文本表示



图自[Po-Sen Huang et al., 2013]

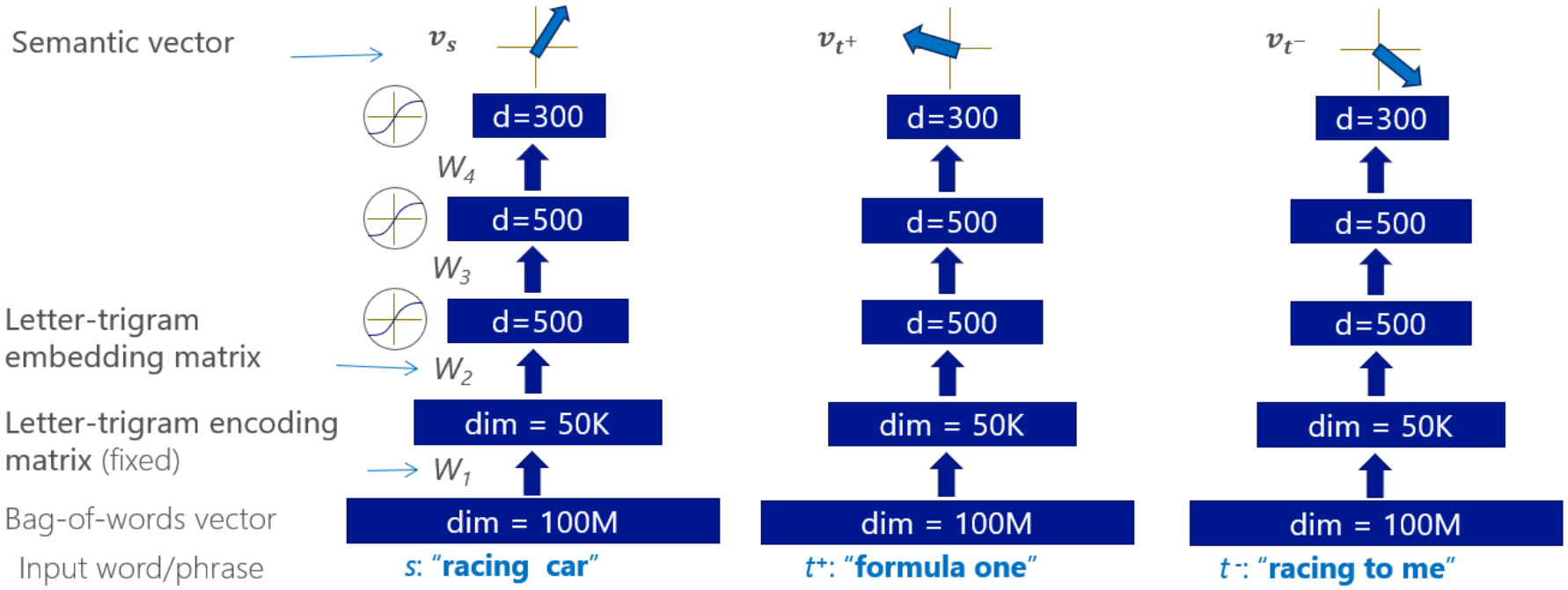
Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, Larry P. Heck, *Learning deep structured semantic models for web search using clickthrough data*, CIKM 2013

DSSM for semantic embedding Learning

Initialization:

Neural networks are initialized with random weights

Huang, He, Gao, Deng, Acero, Heck, "Learning deep structured semantic models for web search using clickthrough data," CIKM, 2013



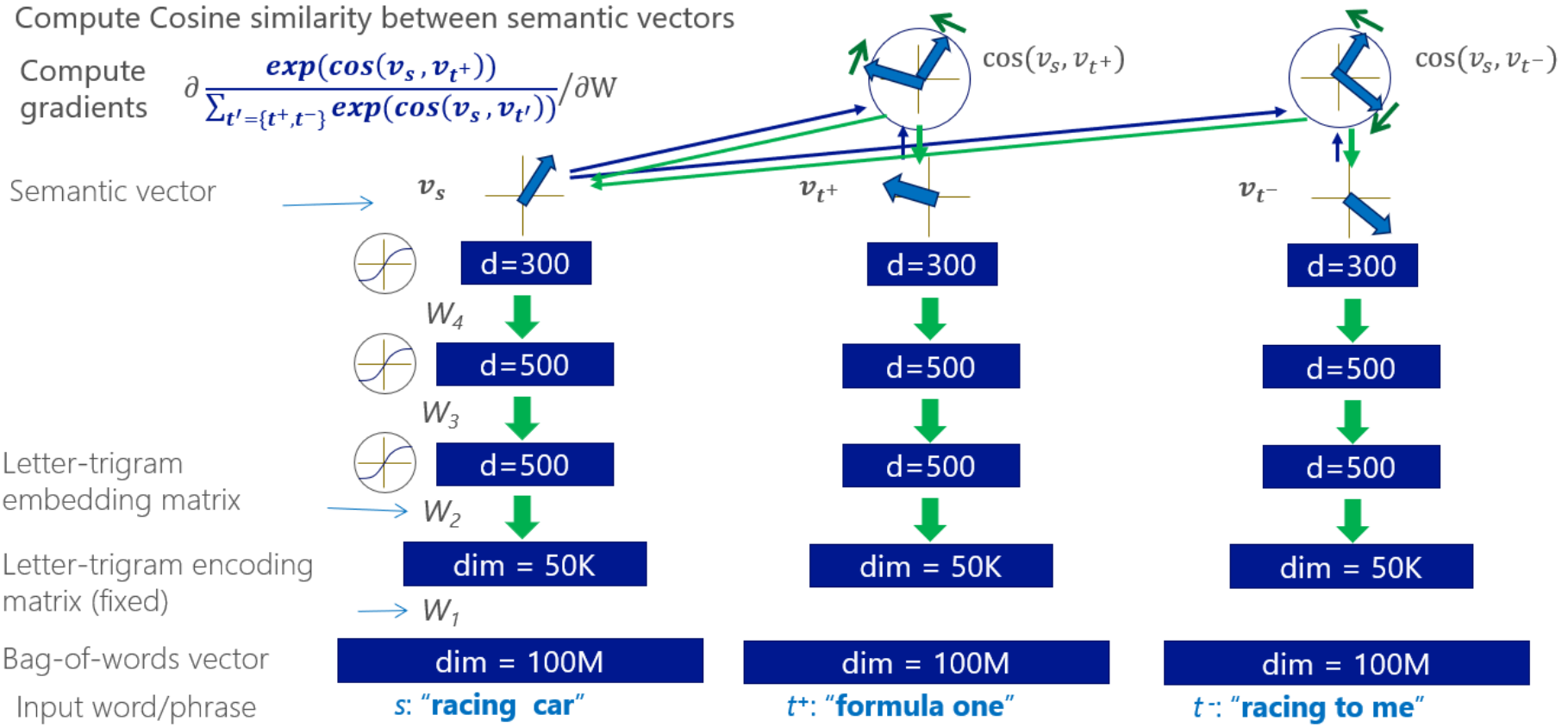
From Xiaodong He DSSM Introduction PPT

DSSM for Semantic Embedding Learning

Training:

Compute Cosine similarity between semantic vectors

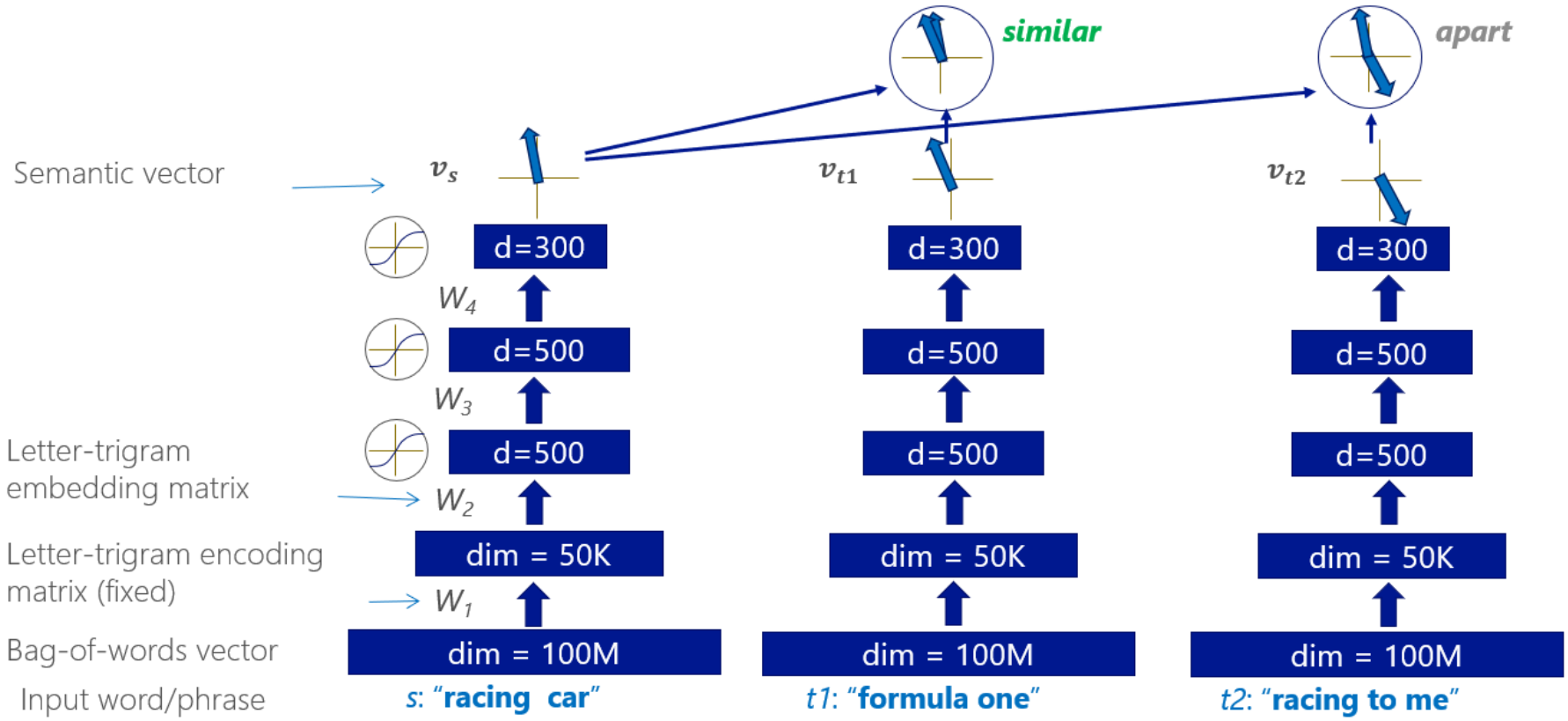
Compute gradients $\partial \frac{exp(cos(v_s, v_{t^+}))}{\sum_{t'=\{t^+,t^-\}} exp(cos(v_s, v_{t'}))} / \partial W$



From Xiaodong He DSSM Introduction PPT

DSSM for Semantic Embedding Learning

Runtime:



From Xiaodong He DSSM Introduction PPT