Natural Language Processing
#L9

# Sentence Representation via CNN

袁彩霞

yuancx@bupt.edu.cn

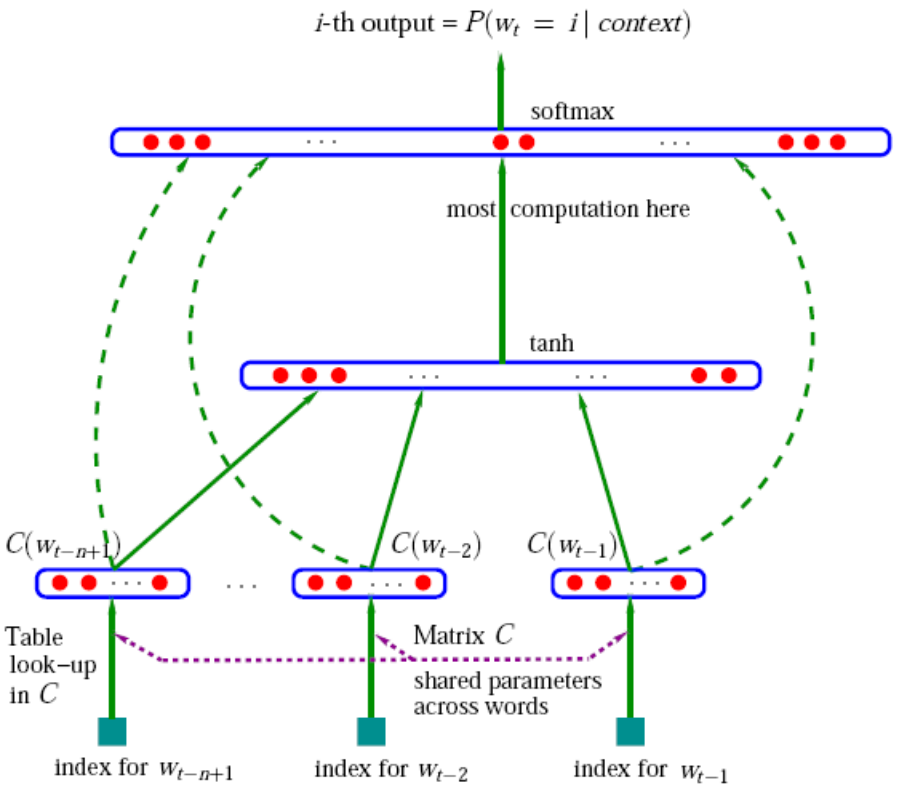
智能科学与技术中心

# Word Embeddings (or Word Vectors)

- 传统NLP技术中：词被表示为在词典中的索引 ($R^V$空间中的"one-hot" 向量)
- 每个词彼此正交
  - 因此：$w_{mother} \cdot w_{father} = 0$
- 问题：是否可以将词表示在一个低维空间$R^D$（ D<V ），使得在$R^V$中语义相似的词在$R^D$中依然相似? (i.e. $w_{mother} \cdot w_{father} > 0$)
- Yes!
  - Latent Semantic Analysis, Latent Dirichlet location, 或基于简单的上下文矩阵共现都可以得到词的压缩表示
  - 深度学习自主学习得到一种词的分布式表示→词嵌入(word embedding)

# Neural Language Models (NLM)

- 一种得到词嵌入的方法
- 通过一个隐藏层将词从$R^V$映射到$R^D$
- $D$：待调节的模型参数
- 有多种不同的神经网络语言模型的结构



# Linguistic regularities in the obtained embeddings

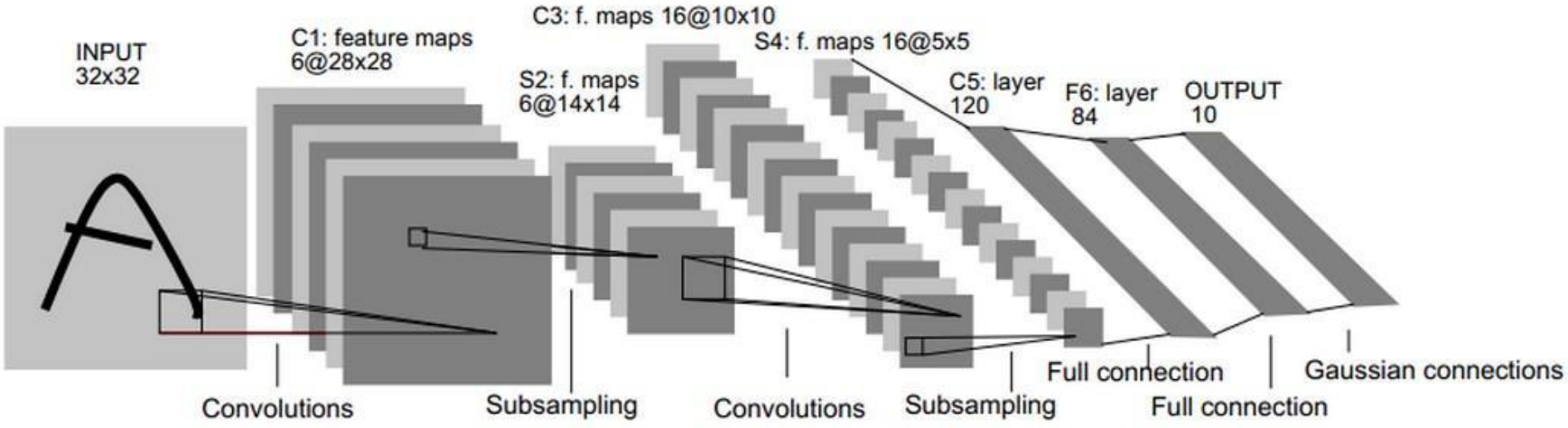
- 学习得到的embeddings可以编码词的语义和语法规则：
  - $w_{big} - w_{bigger} = w_{slow} - w_{slower}$
  - $w_{france} - w_{paris} = w_{China} - w_{Peking}$
- 基于神经网络的词的分布式表示保留了词-上下文共现矩阵的隐藏模式
- 尽管不是神经网络语言模型的训练目标，但得到这种运算非常有用

# Using word embeddings as features in classification

- 词的embeddings可以用作分类器的特征（也可以和其它的传统特征融合）
- 将每个词的embeddings求平均或求和可以得到句子或短语的embeddings
- 然而，这种方法太粗鲁...
  - 词的组合是有序的
  - 词的组合隐藏着某种结构（词法、语法、语义约束）
- 是否可以直接得到句子或短语的embeddings?

# Convolutional Neural Networks (CNN)

- 最早应用于计算机视觉 (Lecun et al, 1989).
- 现在几乎所有的计算机视觉系统都使用 CNNs




LeCun et al., "Gradient-based learning applied to document Recognition", IEEE 1998

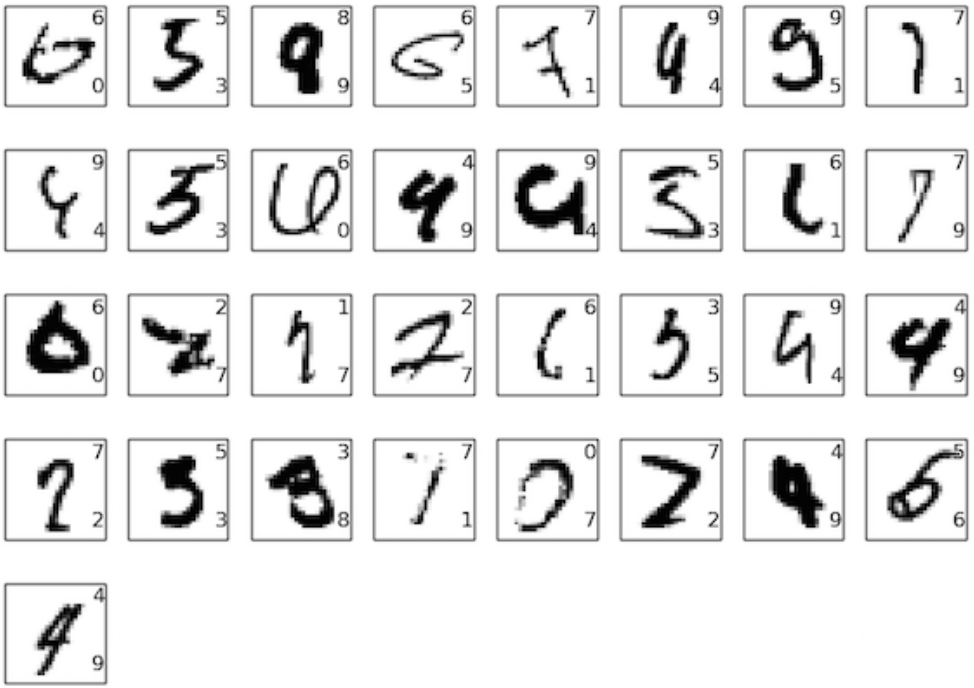
# 动机

- 实践验证：
  - 比起浅层神经网络，深度神经网络会更难训练
  - 然而，如果训练好一个深度网络，它会比浅层网络强大得多

- 因此，
  - 有必要开发一种可以训练深度网络的技术
  - 有必要开发一种能够训练的深度网络结构

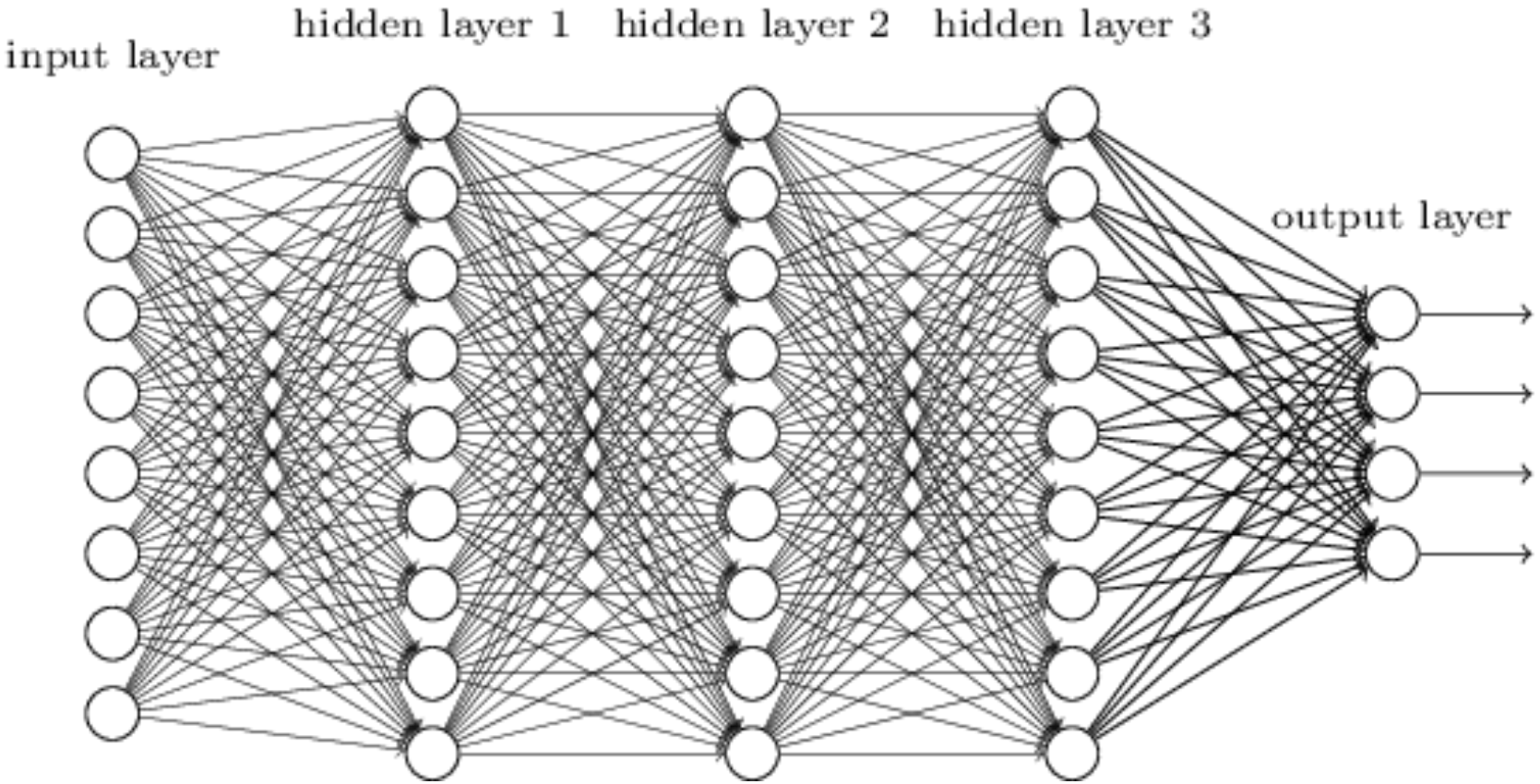
# 动机

- 深度卷积神经网络(CNN)是应用最广的深度神经网络之一
- 对于MNIST中10,000张测试图片，CNN正确识别出了9,967张图片，仅错了33张



# 卷积神经网络

- 一个用于识别手写数字体的全连接网络:



# 卷积神经网络

- 全连接网络存在的问题：
  - 结构过于复杂，参数规模过于庞大
  - 没有考虑到图像的空间结构信息
    - 采用同样的方式处理相距较远的输入和临近的输入像素

What We See

What Computers See

- 卷积神经网络可以简化网络结构，同时利用空间结构信息

# 卷积神经网络

- 卷积运算：
  - 卷积经常用在信号处理中，用于计算信号的延迟累积
  - 假设一个信号发生器每个时刻$t$产生一个信号$x_t$，其信息的衰减率为$w_k$，即在$k-1$个时间步长后，信息衰减为原来的$w_k$倍
    - 例如设$w_1 = 1, w_2 = 1/2, w_3 = 1/4$
  - 则时刻$t$收到的信号$y_t$为当前时刻产生的信息和以前时刻延迟信息的叠加

$$
\begin{aligned}
y_t &= 1 \times x_t + 1/2 \times x_{t-1} + 1/4 \times x_{t-2} \\
&= w_1 \times x_t + w_2 \times x_{t-1} + w_3 \times x_{t-2} \\
&= \sum_{k=1}^{3} w_k \cdot x_{t-k+1}.
\end{aligned}
$$

滤波器（filter）或卷积核（convolution kernel）

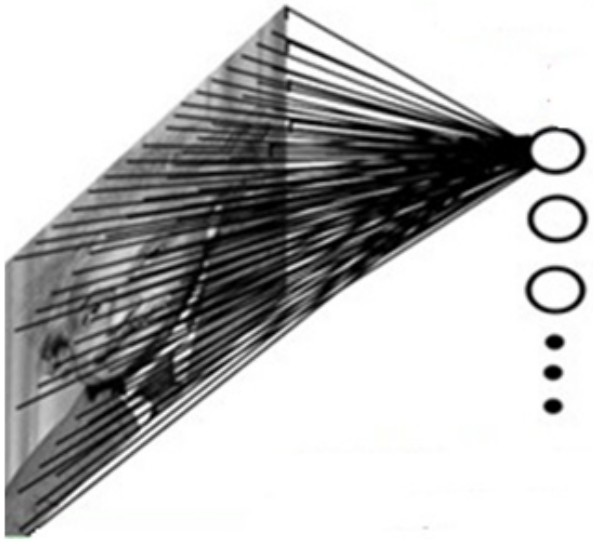
# 卷积神经网络

- 三个基础概念
  - 局部感受野(Local receptive fields)
  - 权值共享(Shared weights)
  - 池化(Pooling)

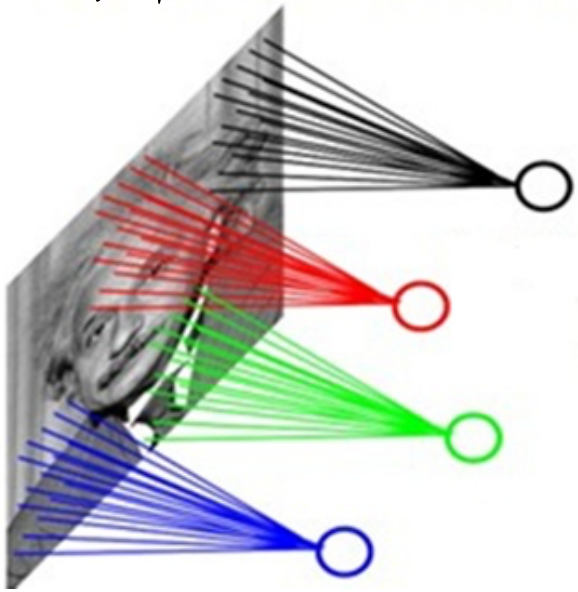
# 局部感受野

- 局部感受(Local reception)：对外部世界由局部到全局的感知

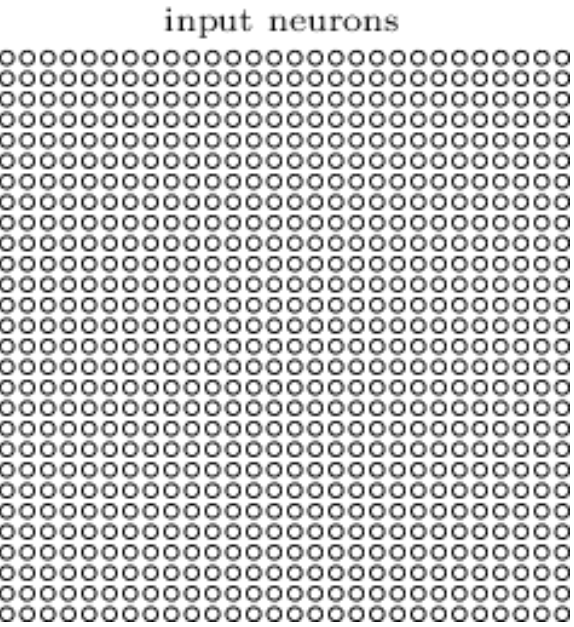
全连接网络



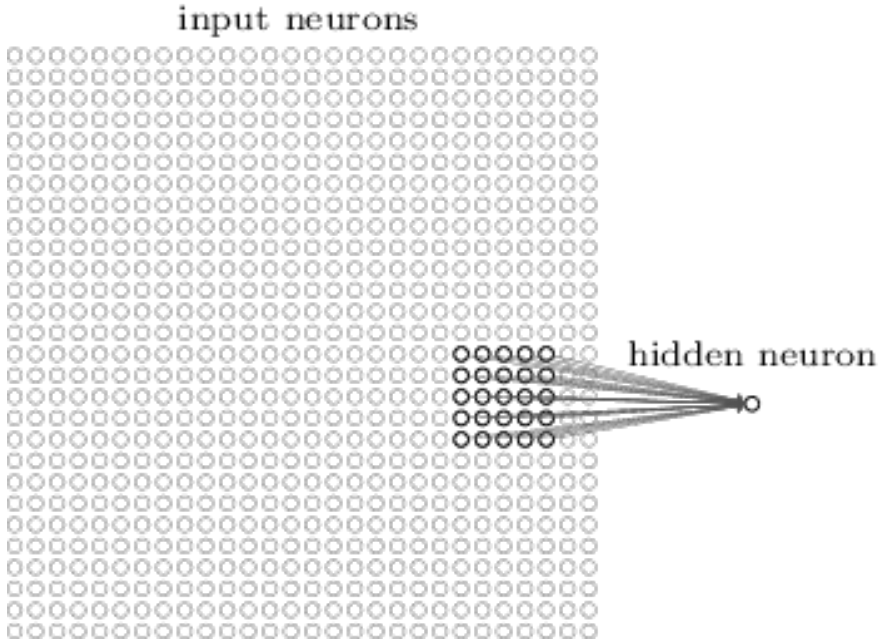
局部连接网络



# 局部感受野



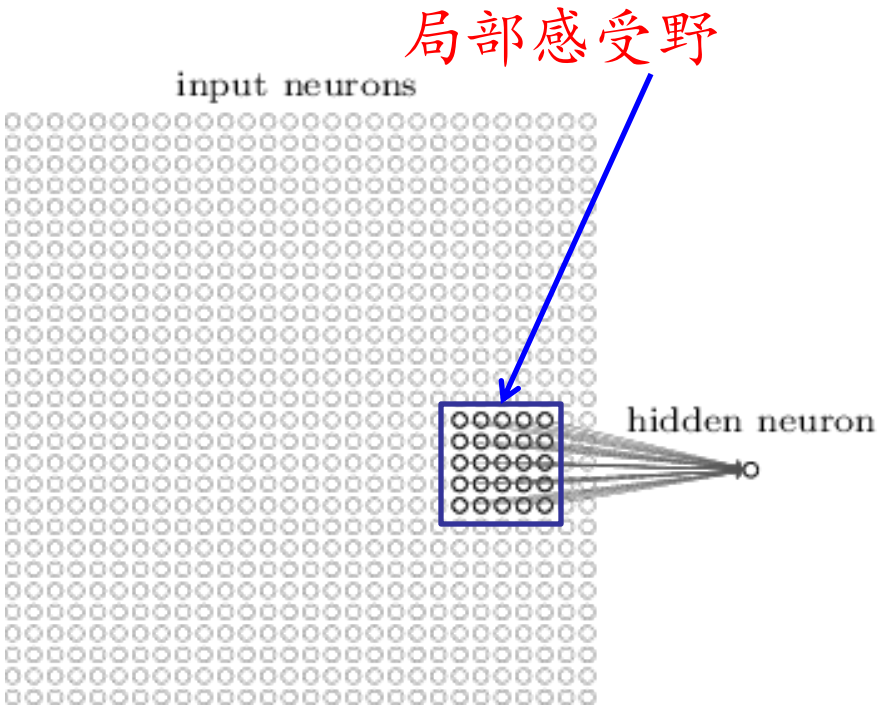

每个输入像素连接到隐藏层的每个神经元




一个小区域的输入像素连接到隐藏层的一个神经元

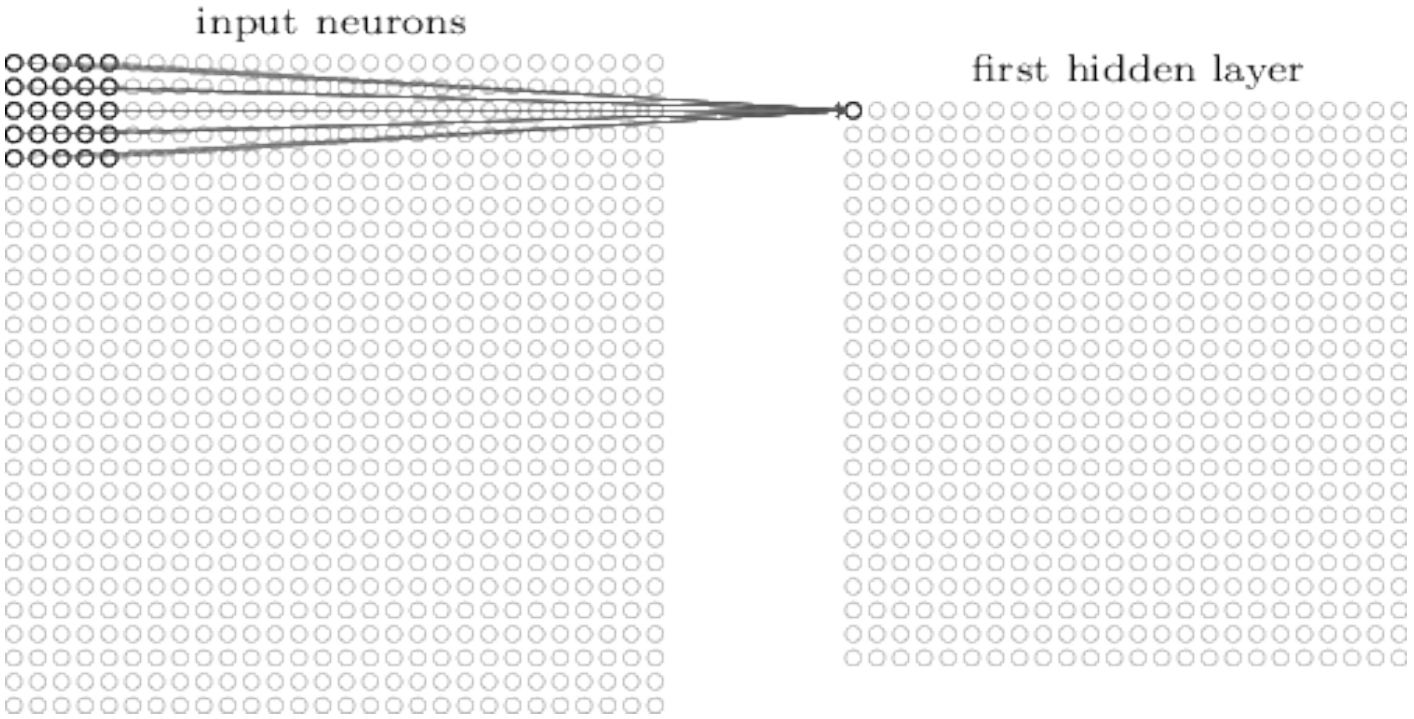
# 局部感受野

- 输入图像中的此类区域称为隐藏神经元的局部感受野 (region)
- 例如，一个 5×5 感受野，对应了25个输入像素



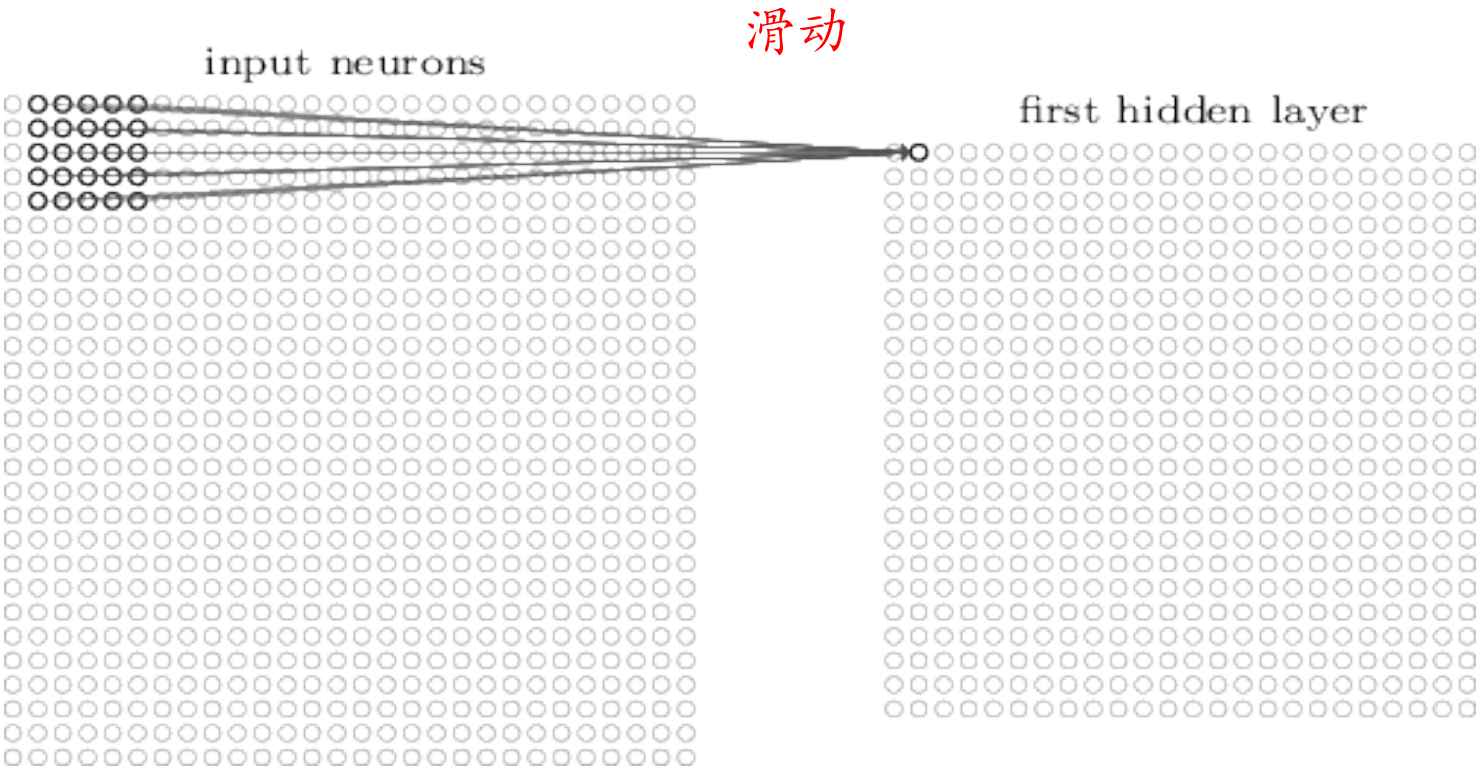
# 局部感受野

- 然后，在整张图片上滑动这个局部感受野
- 每个不同的局部感受野，对应于隐藏层的一个不同的神经元



# 局部感受野

- 然后，在整张图片上滑动这个局部感受野
- 每个不同的局部感受野，对应于隐藏层的一个不同的神经元



# 局部感受野

- 步长(Stride length)：每次局部感受野移动的幅度

- 对于一张28×28的输入图片，使用一个5×5 局部感受野，设步长为1，那么一次卷积后得到的隐藏层的尺寸是多少？
  - 24×24 (28-5+1)

# 权值共享

- 隐藏层中第j, k个神经元的输出:

$$\sigma(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m} a_{j+l,k+m})$$

$\sigma$: 表示激活函数，例如sigmoid函数
$b$: 共享的偏置
$w_{l,m}$: 一个5×5的共享权值矩阵
$a_{j,k}$: 神经元$(j, k)$的激活值

- 24×24隐藏层神经元使用相同权值与偏差

# 权值共享

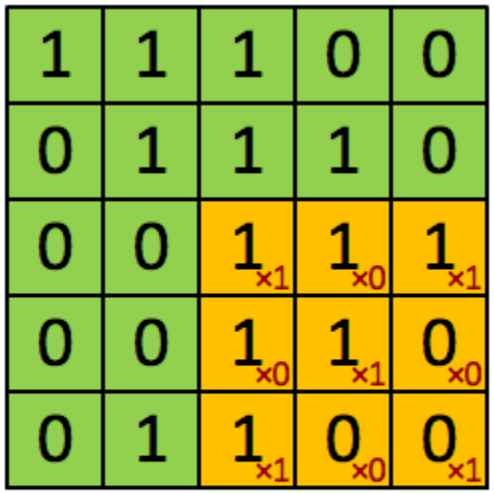
- 更具体地：

$$a^1 = \sigma(b + w * a^0)$$

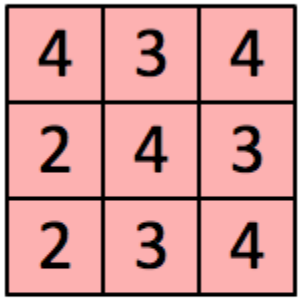
$a^1$ 表示来自一个隐藏神经元的输出激活值
$a^0$ 表示输入激活值
$*$ 表示卷积操作

- 顾名思义：卷积神经网络

# 权值共享




输入像素




卷积后的结果

# 权值共享

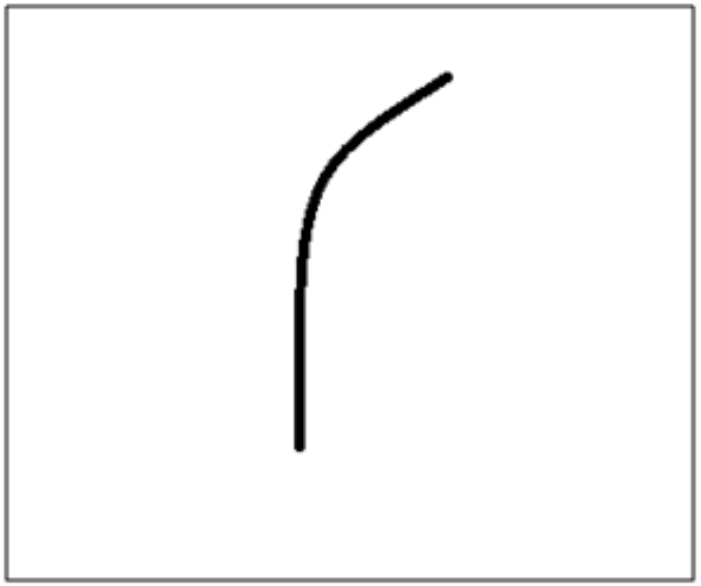
- 卷积操作在高层空间实际上做了什么?
  - 第一个隐藏层中所有神经元都检测到了相同特征，只是在不同的输入图像的位置
- 因此，从输入层到隐藏层的映射通常称之为特征映射 (feature map)
- 共享的权值和偏置被称为卷积核(kernel)或者滤波器(filter)

# 权值共享

- 例1:
  - 一个7 x 7的曲线检测滤波器

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 30 | 0 |
| 0 | 0 | 0 | 0 | 30 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

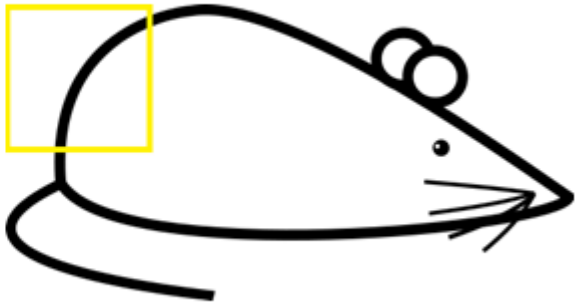
Pixel representation of filter



Visualization of a curve detector filter

# 权值共享

Original image



Visualization of the filter on the image

Visualization of the receptive field

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 30 |
| 0 | 0 | 0 | 0 | 50 | 50 | 50 |
| 0 | 0 | 0 | 20 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |

Pixel representation of the receptive field

*

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 30 | 0 |
| 0 | 0 | 0 | 0 | 30 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

# 权值共享

Visualization of the filter on the image

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 40 | 0 | 0 | 0 | 0 | 0 |
| 40 | 0 | 40 | 0 | 0 | 0 | 0 |
| 40 | 20 | 0 | 0 | 0 | 0 | 0 |
| 0 | 50 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 50 | 0 | 0 | 0 | 0 |
| 25 | 25 | 0 | 50 | 0 | 0 | 0 |

Pixel representation of receptive field

*

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 30 | 0 |
| 0 | 0 | 0 | 0 | 30 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Pixel representation of filter

Multiplication and Summation = 0

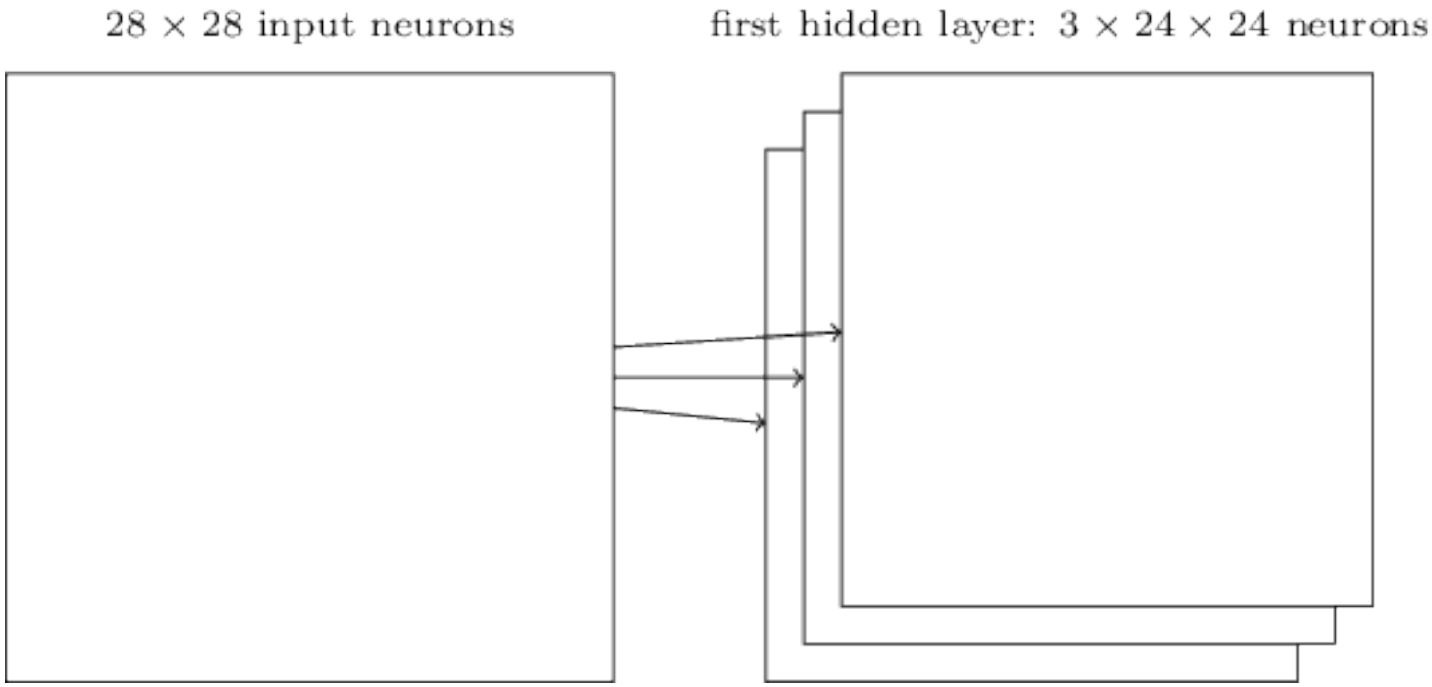
- Feature map显示了图片中哪些区域有感兴趣的特征

# 权值共享

- 例2: 一个边缘检测滤波器

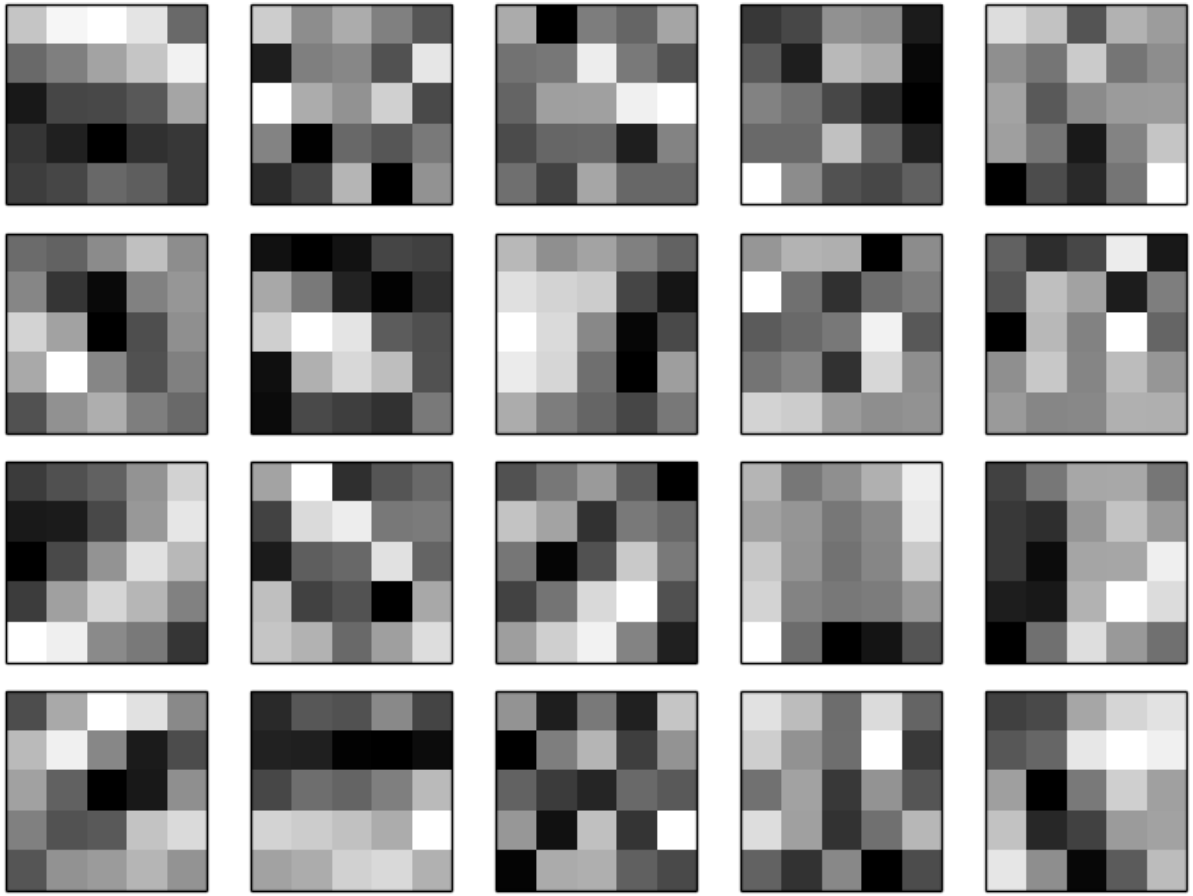
# 权值共享

- 实际中，往往需要不止一个feature map
- 如下图设计了3个不同的feature map



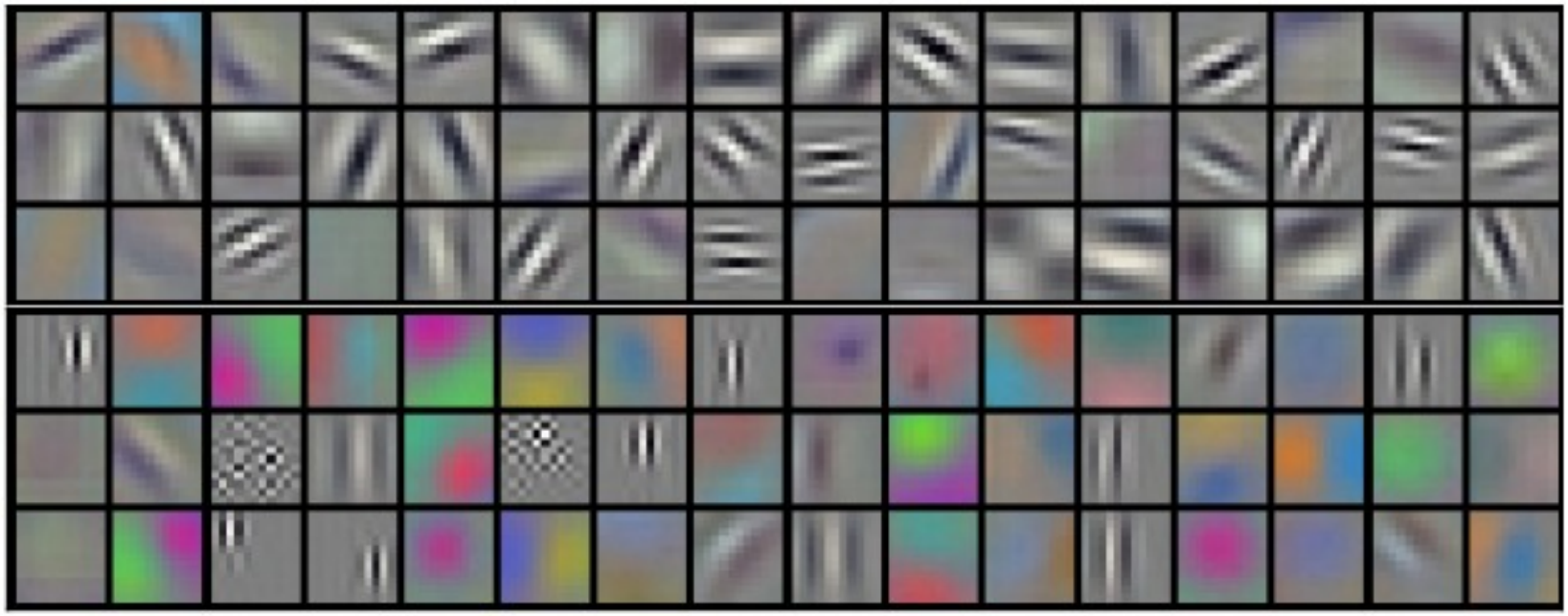
# 权值共享

- 每一个这样的滤波器都是一种特殊的特征标识符
- 卷积核的可视化
  - 下图为20 个不同的feature maps



# 权值共享

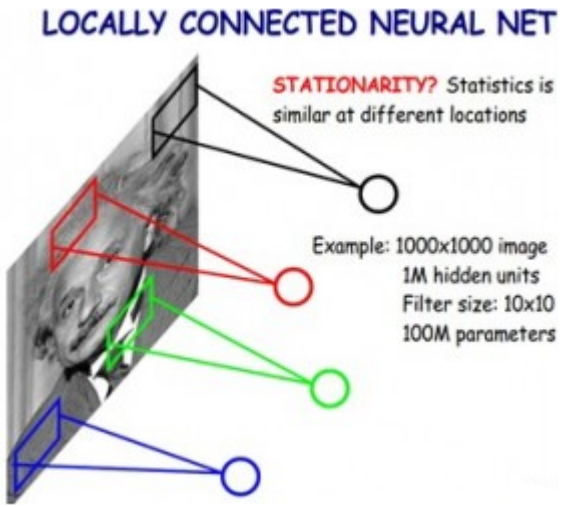
- 卷积核的可视化
  - 下图为AlexNet中的96个滤波器

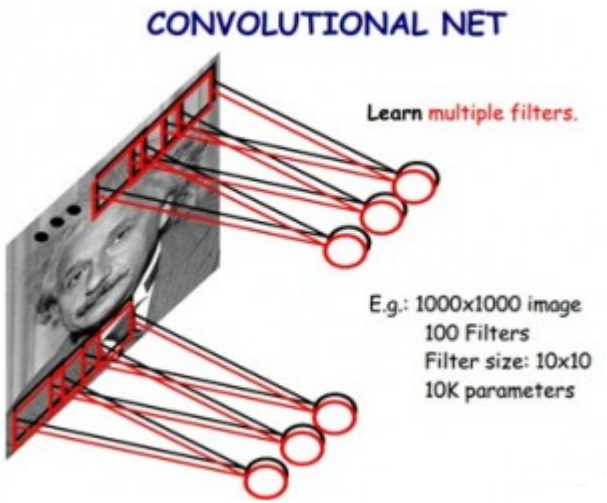


Alex Krizhevsky, et al. (2012) ImageNet Classification with Deep Convolutional Neural Networks.

# 权值共享

- 网络会学习与空间结构相关的特征
  - 例如，许多特征都有清晰的明亮子区域
- 滤波器越多，特征映射的深度越大，得到的关于输入的信息就越多
- 共享权值极大降低了CNN的参数规模

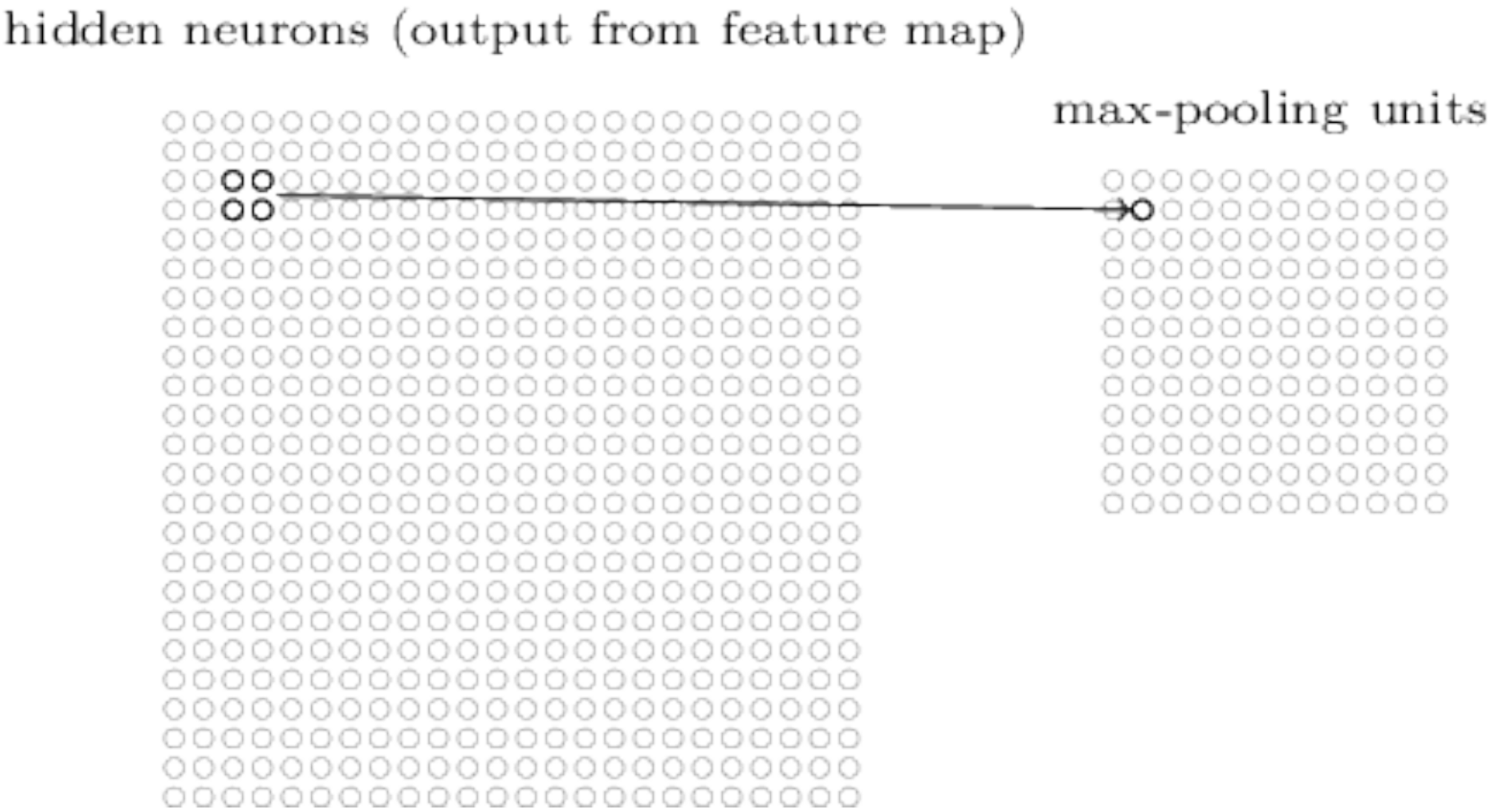


# 池化

- 一旦知道原始输入中的特定特征，其确切位置就不如其与其它特征的相对位置一样重要了
- 池化层通常用在卷积层之后，以简化卷积层输出的信息
  - Max-pooling
  - Mean-pooling
  - L2 pooling
  - …

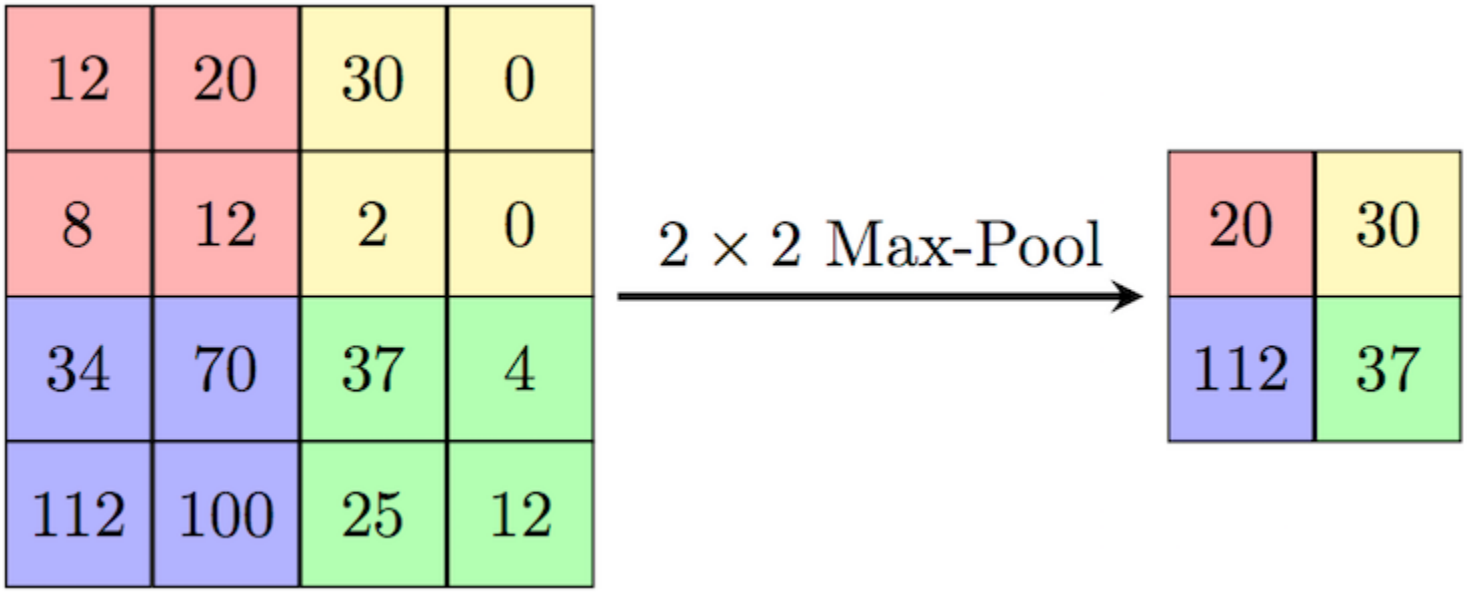
# 池化




卷积层输出的24×24个神经元，在使用2×2的池化后，减少为12×12个神经元

# 池化

- 在最大池化中，池化层输出池化矩阵中最大激活值

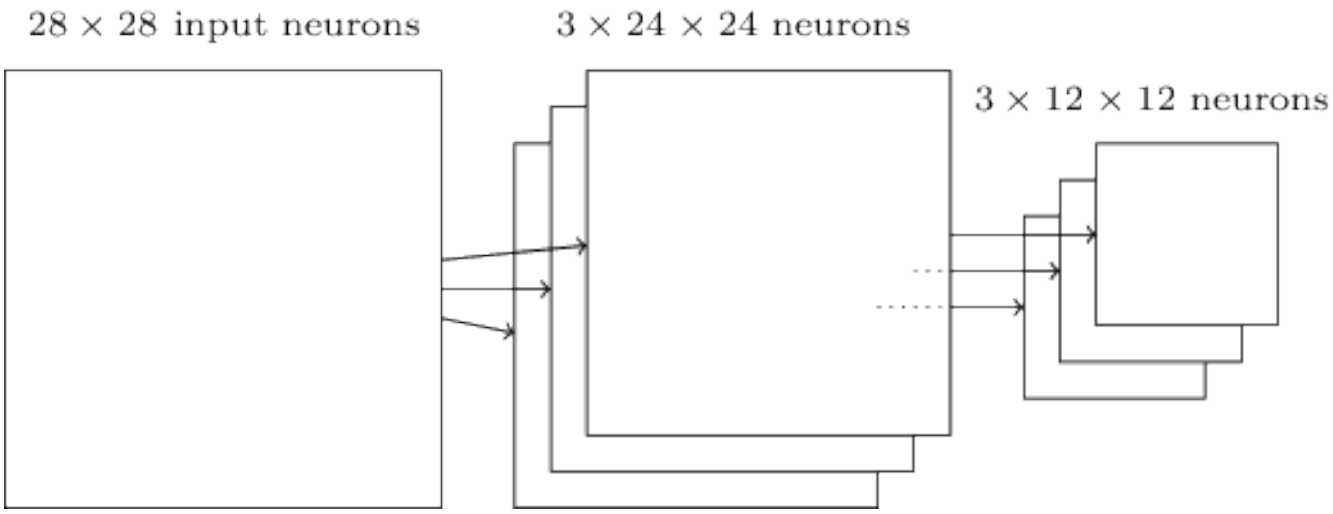


# 池化

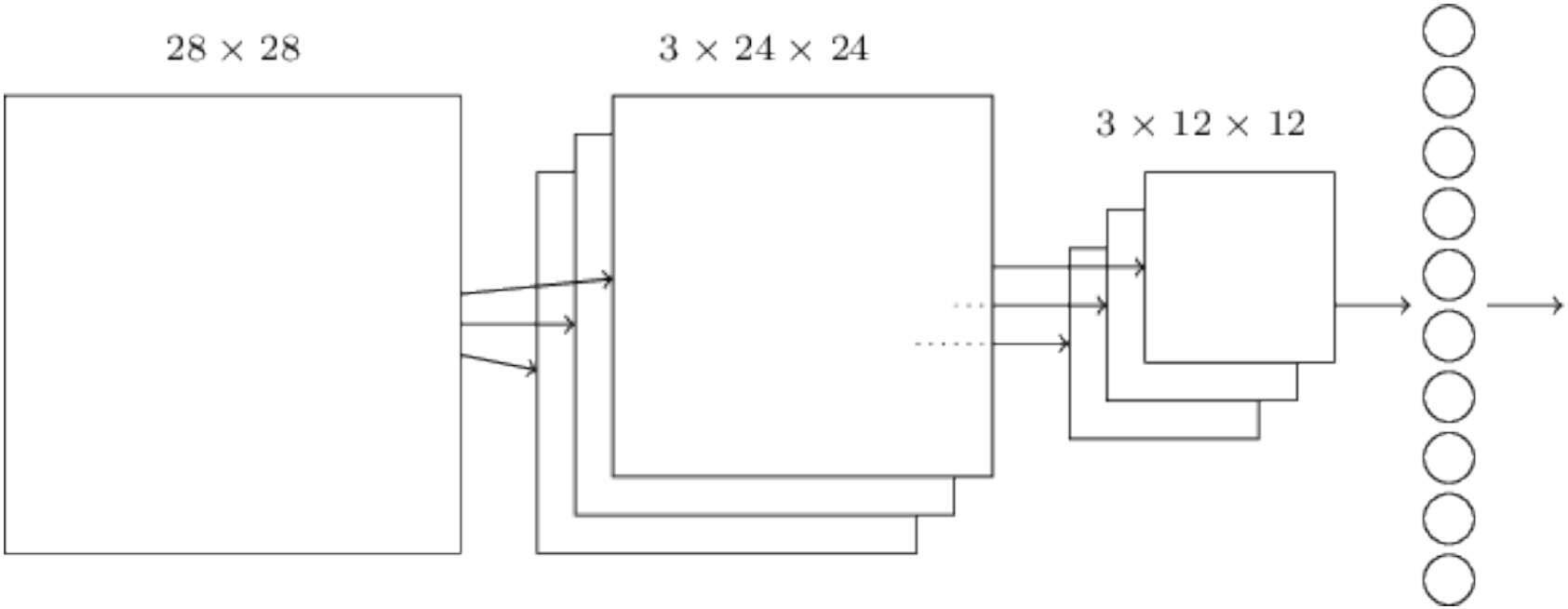
- 池化的两个主要目的:
  - 减少参数(e.g.，上面的例子降低了75%)，因此可以降低计算量
    - 控制过拟合
  - 使得特征具有局部的转移和扭曲不变性（微小形变的鲁棒性）

# 池化

- 池化分别应用于每一个feature map

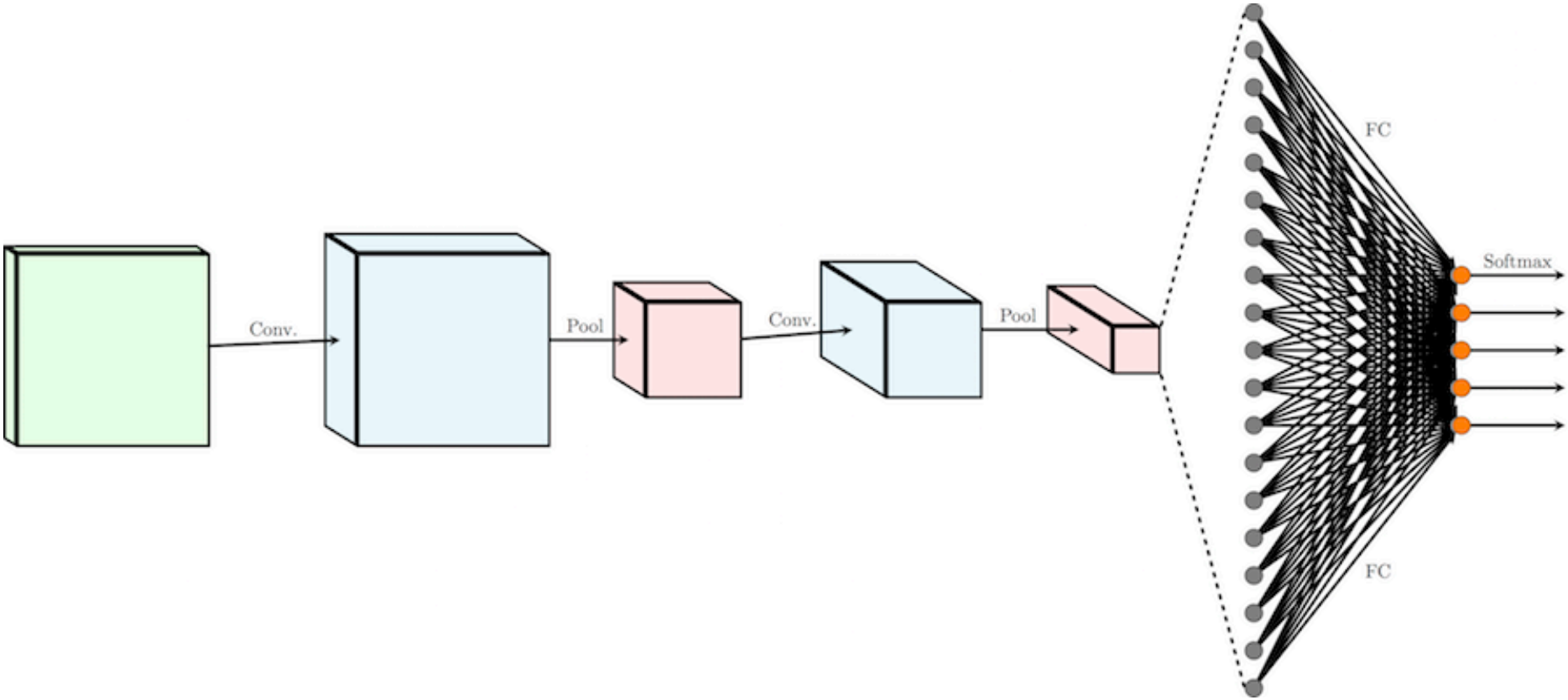


# 一个一般的CNN结构



# 一个一般的CNN结构

- 更深层的卷积网络结构:

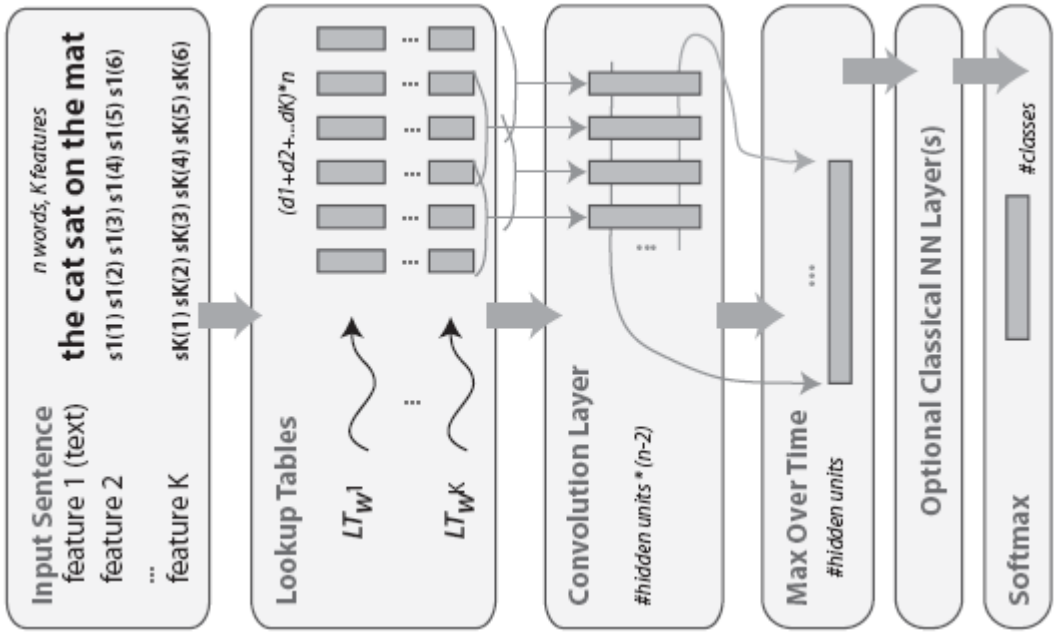


# CNNs in NLP

- 研究者们使用CNNs在诸多NLP任务上获得了非常好的性能，例如 POS tagging, SRL, etc.
  - Ref. Collobert et al., “Natural Language Processing (almost) from scratch”, JLMR 2011.
  - Semantic parsing (Yih et al., “Semantic Parsing for Single-Relation Question Answering”, ACL 2014)
  - Search query retrieval (Shen et al., “Learning Semantic Representations Using Convolutional Neural Networks for Web Search”, WWW 2014)
  - Sentiment analysis (Kalchbrenner et al., “A Convolutional Neural Network for Modelling Sentences”, ACL 2014; dos Santos and Gatti, “Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts”, COLING 2014)
  - Machine Translation (Gehring Jonas et al., “A Convolutional Encoder Model for Neural Machine Translation”, ACL 2017)
  - ……
- 实际中，往往采用具有多个卷机层的的CNN

# CNNs in NLP

- 一般架构：
  - CNN + softmax 分类器
  - CNN + CRF标注器
- Collobert-Weston 使用了预训练的词向量+CNN+Softmax




Ref. Collobert et al., “Natural Language Processing (almost) from scratch”, JLMR 2011.

# CNN architecture

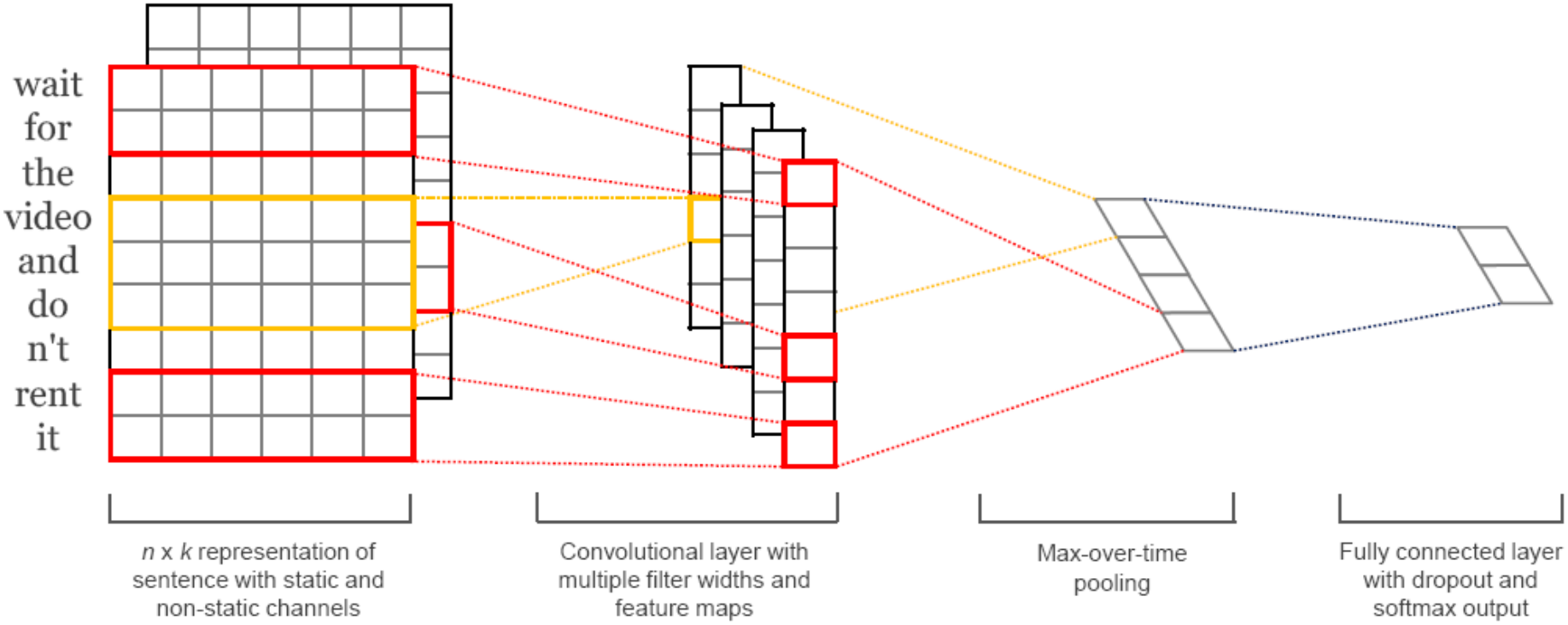
- One layer of convolution with ReLU ($f(x) = x_+$) non-linearity.
- Multiple feature maps and multiple filter widths.
- Filter widths of 3, 4, 5 with 100 feature maps each, so 300 units in the penultimate layer.
- Words not in word2vec are initialized randomly from U[-a, a] where $a$ is chosen such that the unknown words have the same variance as words already in word2vec.
- Regularization: Dropout on the penultimate layer with a constraint on $L_2$-norms of the weight vectors.
- These hyperparameters were chosen via some light tuning on one of the datasets.

# Dropout

- Proposed by Hinton et al. (2012) to prevent co-adaptation of hidden units.
- During forward propagation, randomly "mask" (set to zero) each unit with probability p.
- Backpropagate only through unmasked units.
- At test time, do not use dropout, but scale the weights by p.
- Like taking the geometric average of different models.
- Rescale weights to have $L_2$-norm = s whenever $L_2$-norm > s after a gradient step.

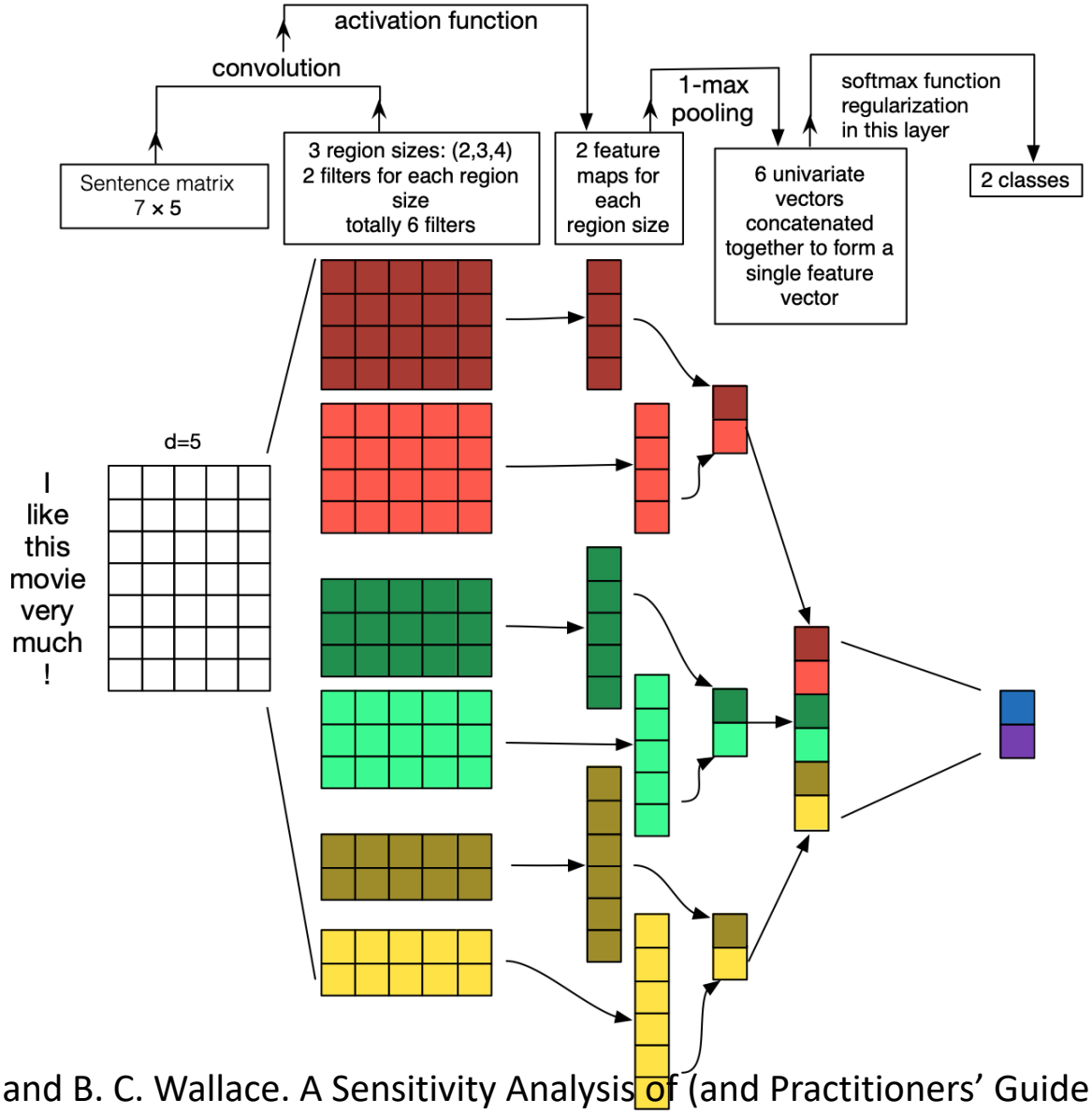
# Multi-channel CNN

- Two “channels” of embeddings (i.e. look-up tables).
- One is allowed to change, while one is kept fixed.
- Both initialized with word2vec.



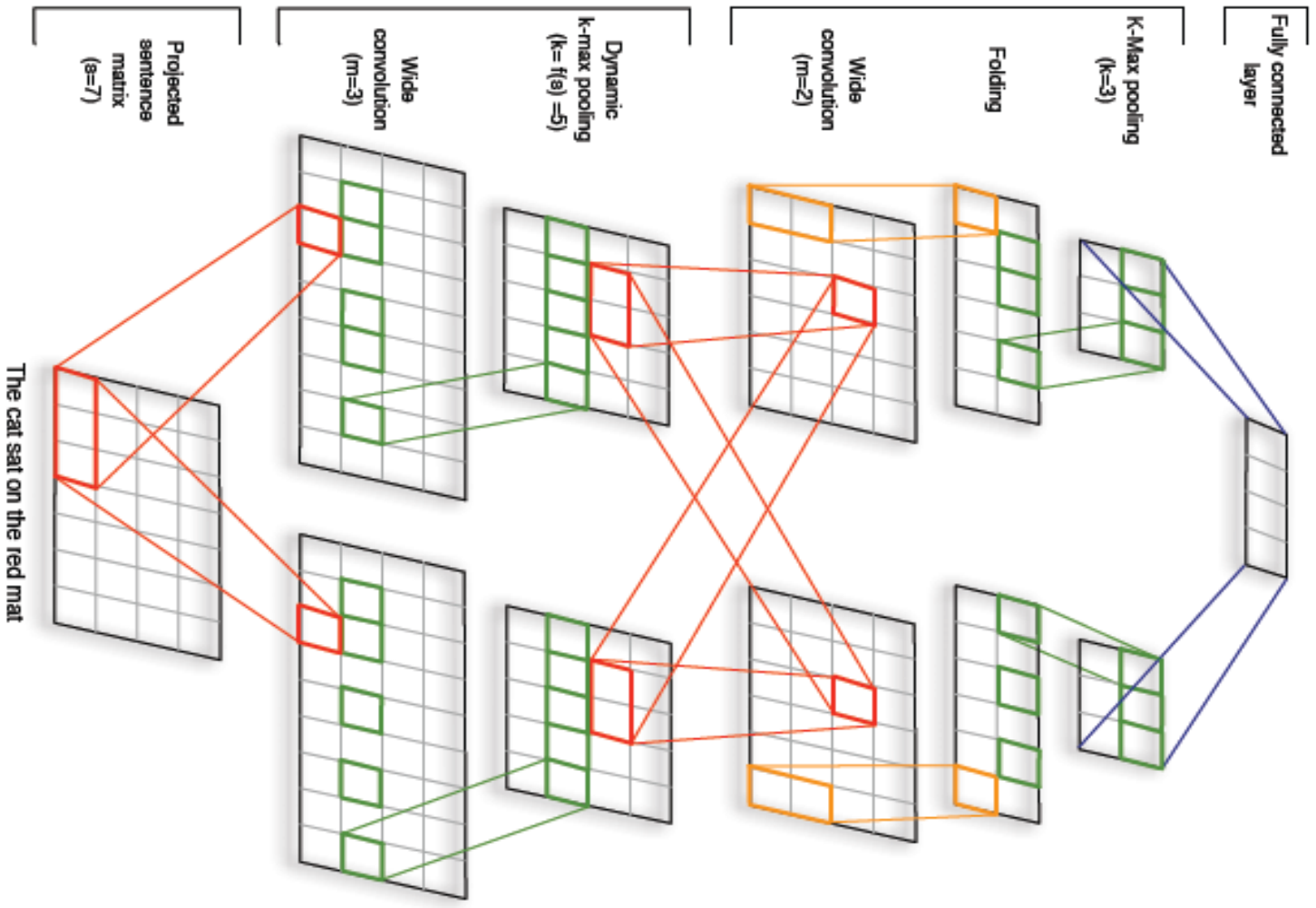

Yoon Kim, Convolutional Neural Networks for Sentence Classification, 2014.

# One-layer CNNs




Y. Zhang and B. C. Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. IJCNLP 2016.

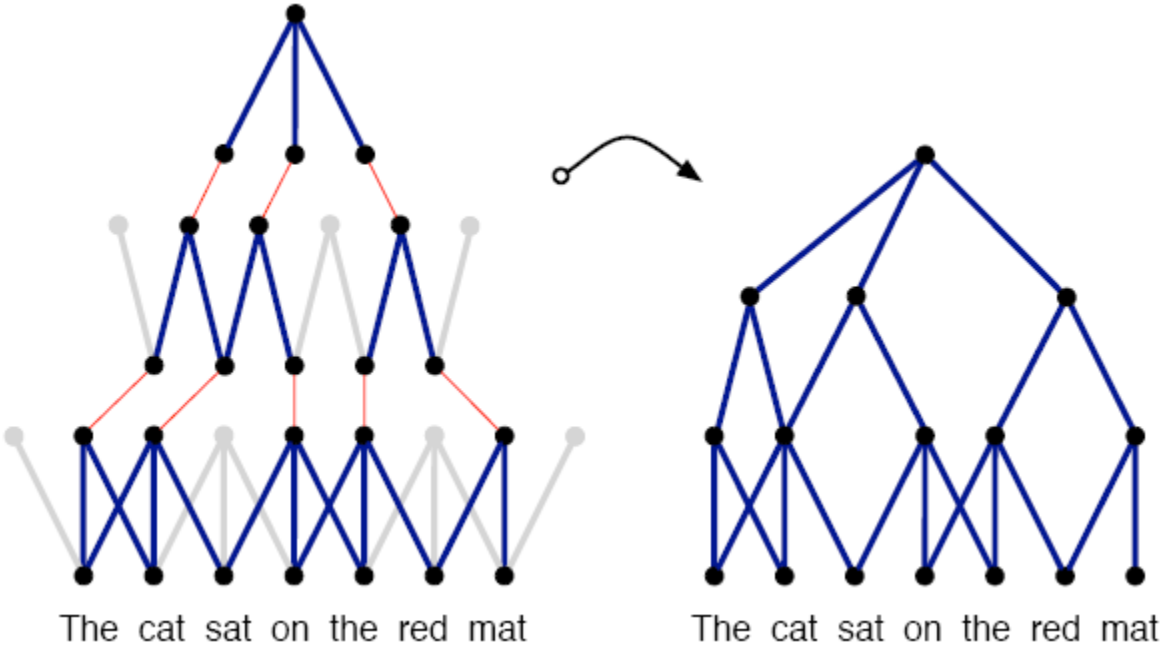
# Dynamic CNN



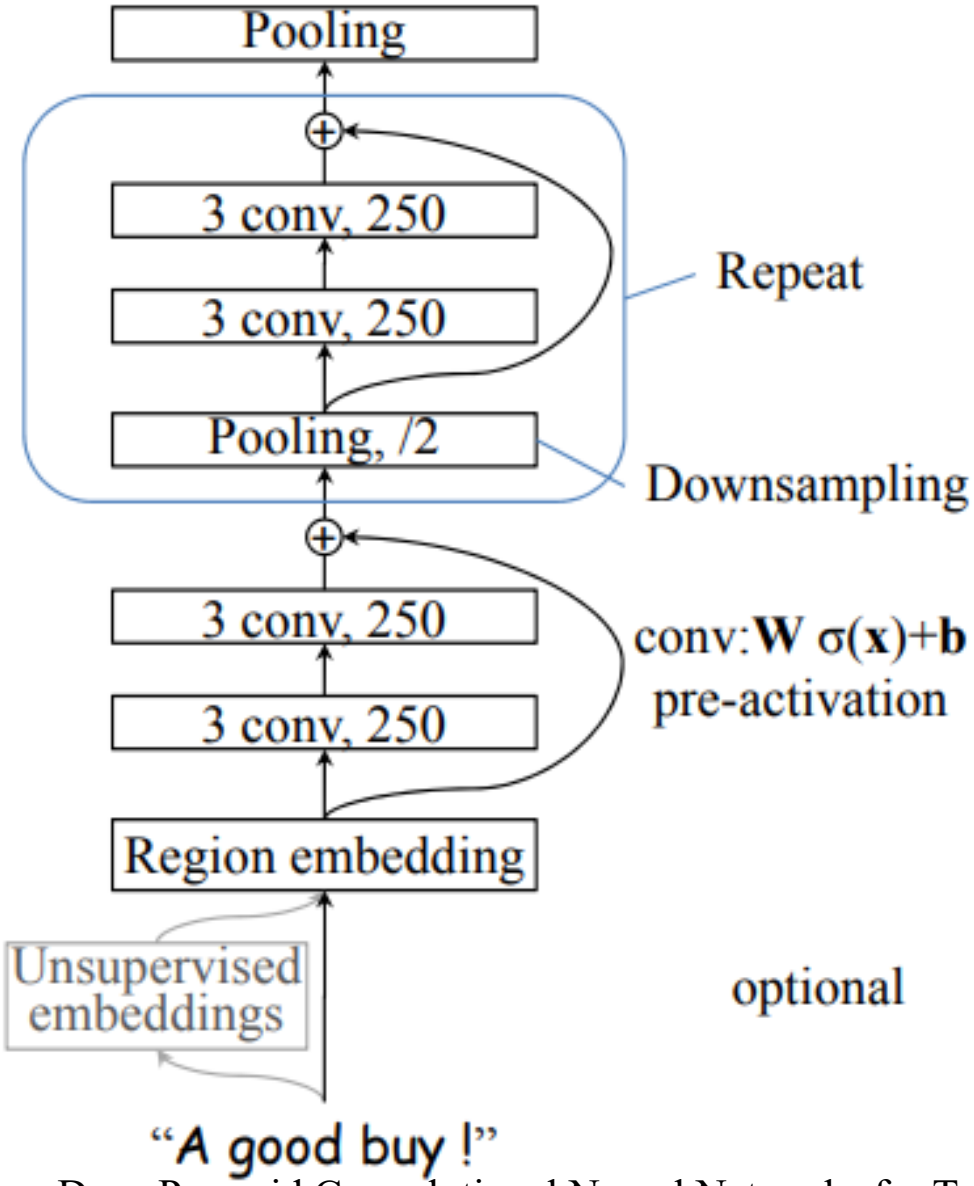

Kalchbrenner et al., "A Convolutional Neural Network for Modeling Sentences", ACL 2014

# Further Observations

- Subgraph of a feature graph induced over an input sentence in a Dynamic CNN.

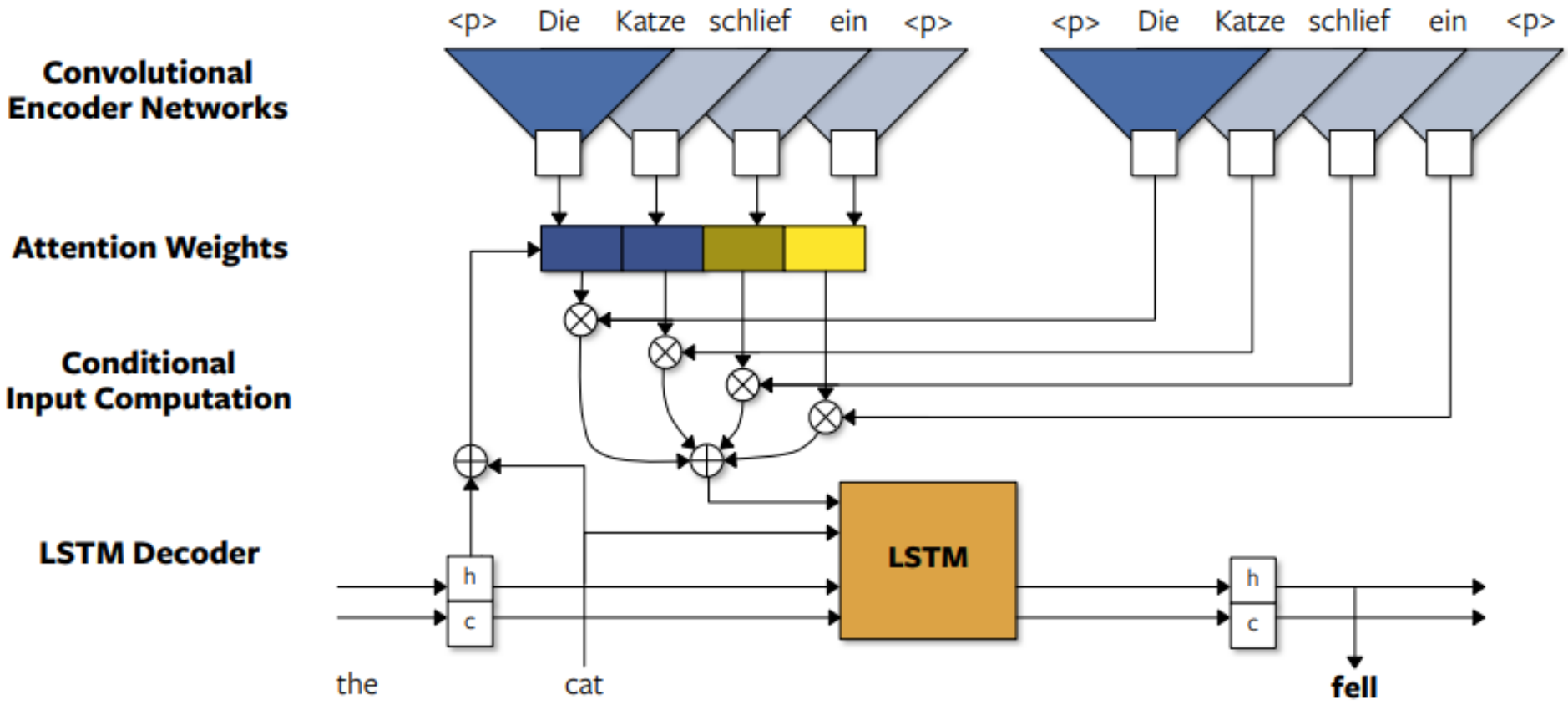


# Recent work




R. Johnson and T. Zhang. Deep Pyramid Convolutional Neural Networks for Text Categorization. ACL 2017

# Recent work




J. Gehring, et al. A Convolutional Encoder Model for Neural Machine Translation. ACL 2017.

# Overview

- Text representation
- CNN and Its Pros/Cons

- Next lecture: Text Representation via RNN