COLX-585: Trends in Computational Linguistics

Muhammad Abdul-Mageed

muhammad.mageed@ubc.ca

Natural Language Processing Lab

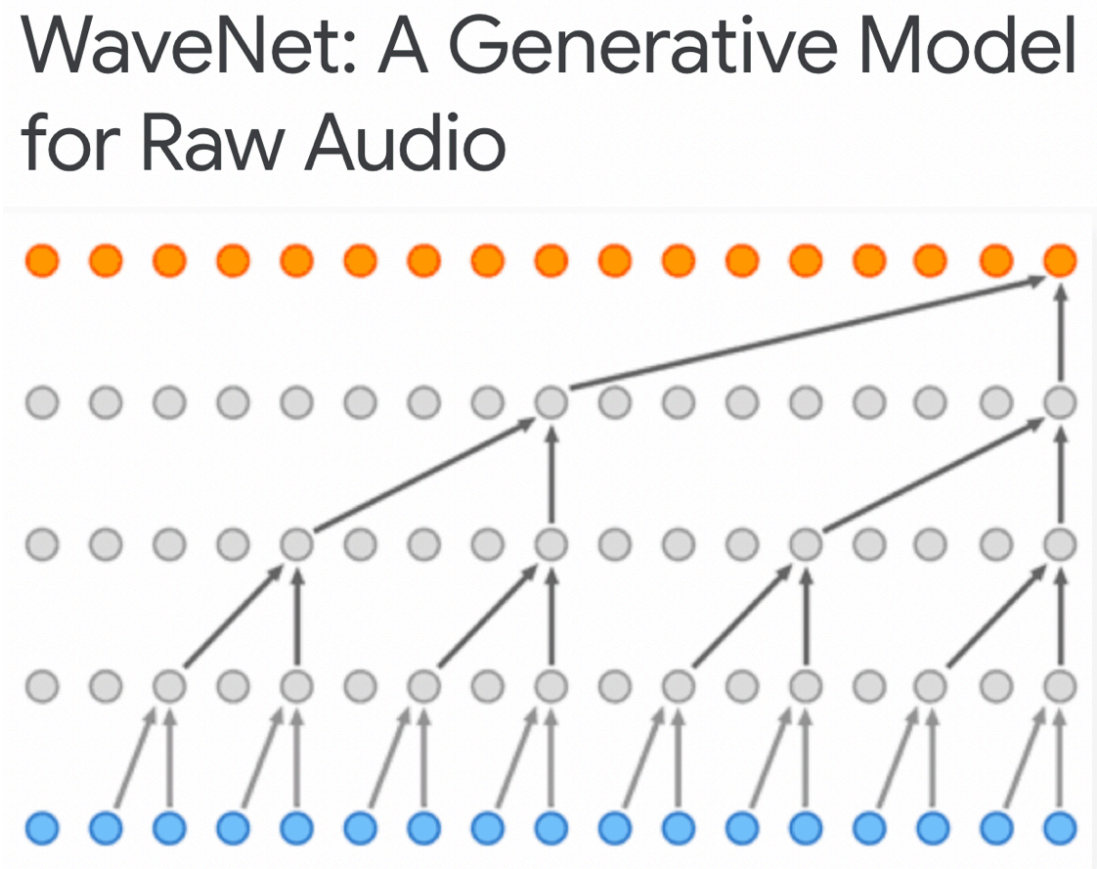
The University of British Columbia

Table of Contents

ConvNets: Pervasive Applications

Figure: Autonomous vehicles, art, and celebrity face-generation, etc.

ConvNets In Speech Synthesis



See: WaveNet.

Imagined Speech Decoding

SPEAK YOUR MIND!
Towards Imagined Speech Recognition With Hierarchical Deep Learning

Pramit Saha[1], *Muhammad Abdul-Mageed*[2], *Sidney Fels*[1]

[1]Human Communication Technologies Lab, University of British Columbia [2]Natural Language Processing Lab, University of British Columbia

pramit@ece.ubc.ca, muhammad.mageed@ubc.ca, ssfels@ece.ubc.ca

Abstract

Speech-related Brain Computer Interface (BCI) technologies provide effective vocal communication strategies for controlling devices through speech commands interpreted from brain signals. In order to infer imagined speech from active thoughts, we propose a novel hierarchical deep learning BCI system for subject-independent classification of 11 speech tokens including phonemes and words. Our novel approach exploits predicted articulatory information of six phonological categories (e.g., nasal, bilabial) as an intermediate step for classifying the phonemes and words, thereby finding discriminative signal responsible for natural speech synthesis. The proposed network is composed of hierarchical combination of spatial and temporal CNN cascaded with a deep autoencoder. Our best models on the KARA database achieve an average accuracy of 83.42% across the six different binary phonological classification tasks, and 53.36% for the individual token identification task, significantly outperforming our baselines. Ultimately, our work suggests the possible existence of a brain imagery footprint for the underlying articulatory movement related to different sounds that can be used to aid imagined speech decoding.

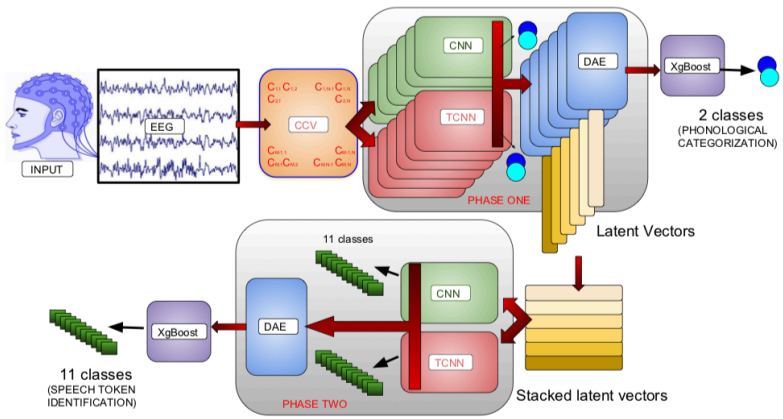


Figure 1: *Overall framework of the proposed approach*

Among the various brain activity-monitoring modalities in BCI, Electroencephalography (EEG) [3, 4] has been demonstrated as carrying promising signal for differentiating different brain activities (through measurement of related electric fields).

Model Example

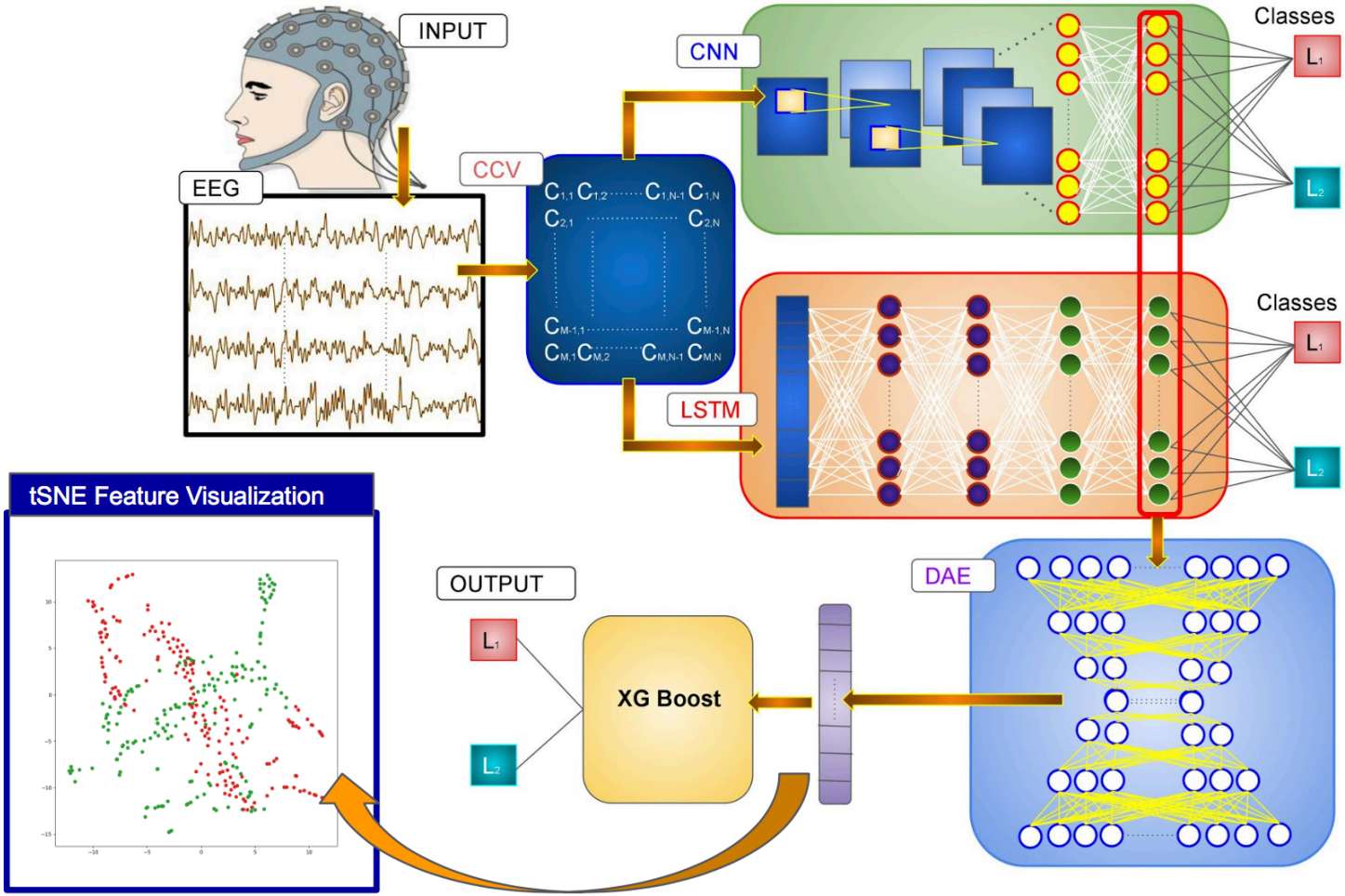


Figure: Decoding Imagined Speech from Brain Signal (EEG data).

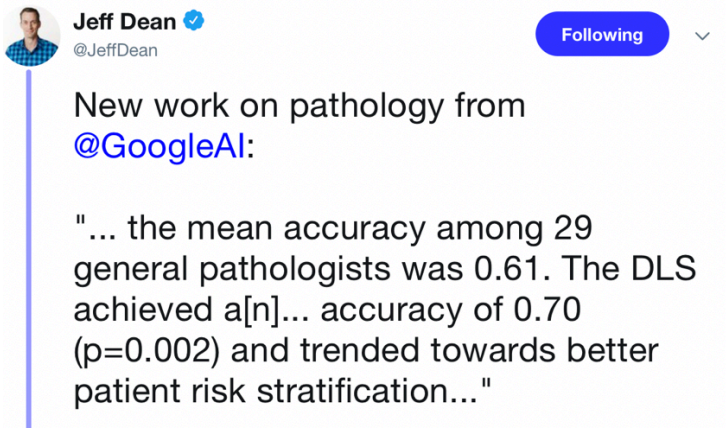
ConvNets In Medical Applications

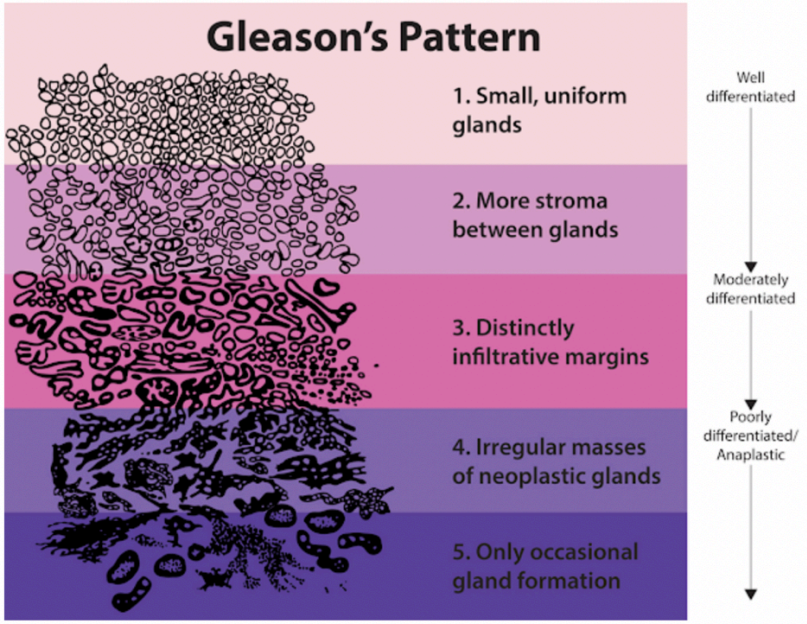
Improved Grading of Prostate Cancer Using Deep Learning

Friday, November 16, 2018

Posted by Martin Stumpe, Technical Lead and Craig Mermel, Product Manager, Healthcare, Google AI

Approximately 1 in 9 men in the United States will develop prostate cancer in their lifetime, making it the most common cancer in males. Despite being common, prostate cancers are frequently non-aggressive, making it challenging to determine if the cancer poses a significant enough risk to the patient to warrant treatment such as surgical removal of the prostate (prostatectomy) or radiation therapy. A key factor that helps in the "risk stratification" of prostate cancer patients is the Gleason grade, which classifies the cancer cells based on how closely they resemble normal prostate glands when viewed on a slide under a microscope.





See: Google AI Blog.

2D Convolution

5 x 5 matrix (e.g., input image)

1	1	0	0	0
0	0	1	1	1
0	0	0	0	0
1	1	0	0	1
1	1	0	1	1

3 x 3 kernel (*aka* filter)

1	0	0
0	1	1
1	0	0

Desired feature map

Figure: **Input**: 5 x 5 matrix/2D Tensor. We will convolve with a 3 x 3 kernel, with a `stride` of 1. Our goal is to acquire an feature map (filling in the 3 x 3 matrix on right-hand side).

Convolving 1

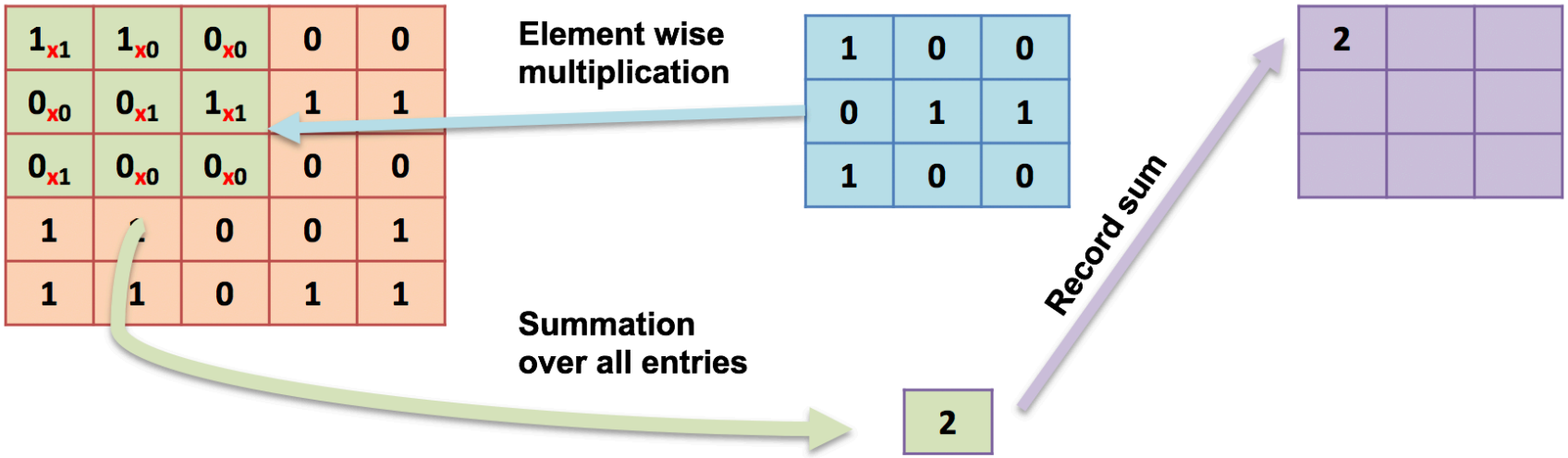


Figure: Filling in the first cell.

Convolving 2

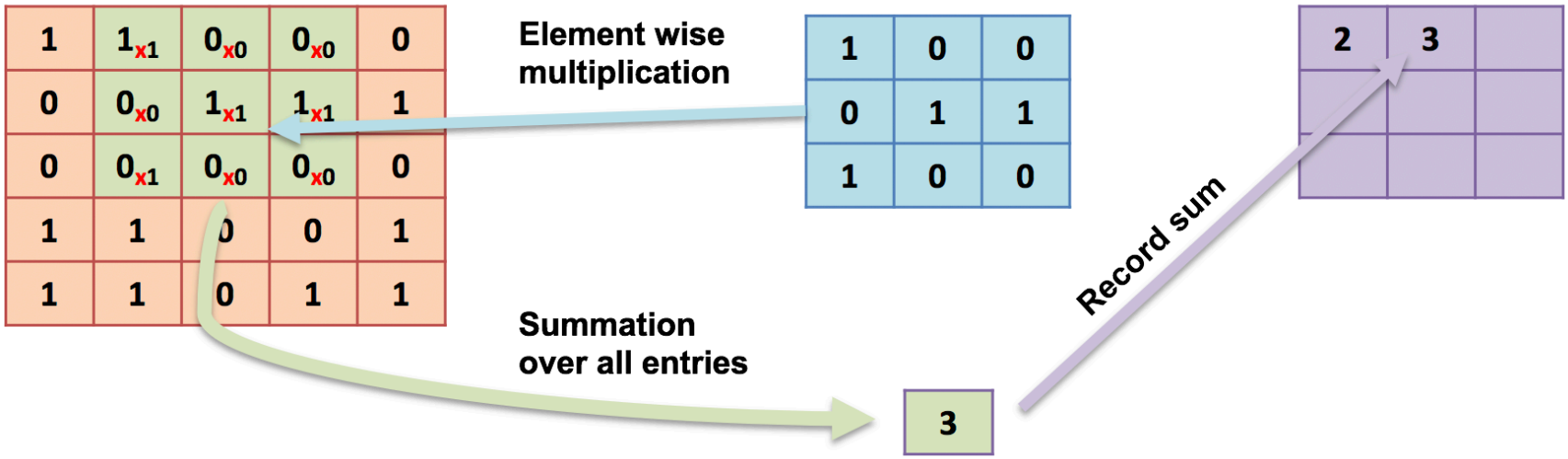


Figure: Filling in the second cell.

Convolving 3

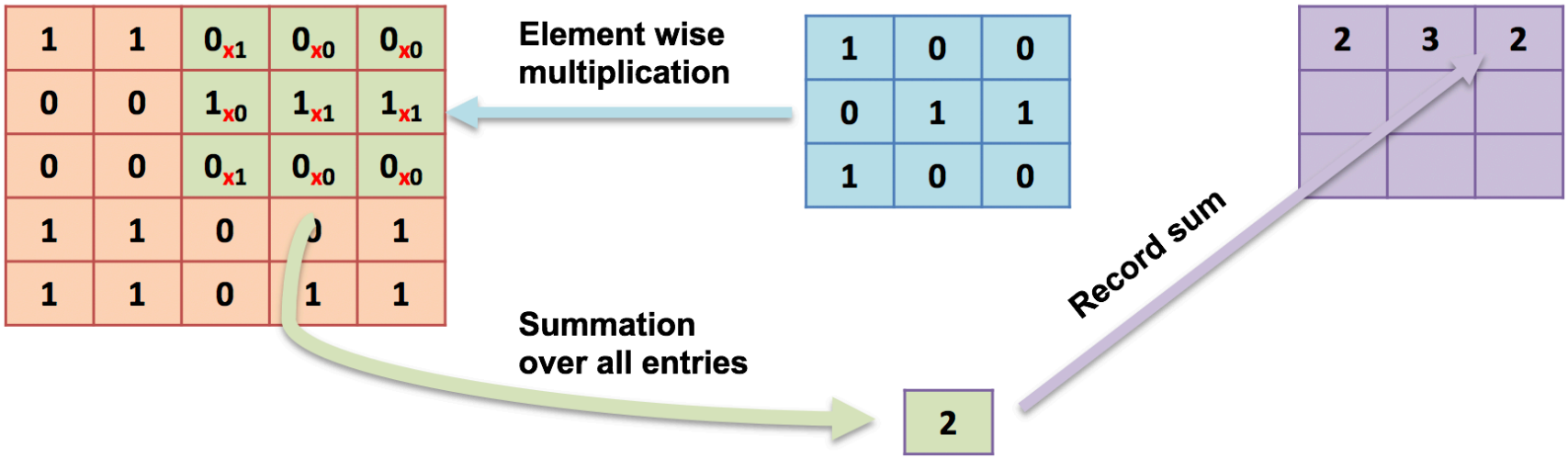


Figure: Filling in the third cell.

Convolving 4

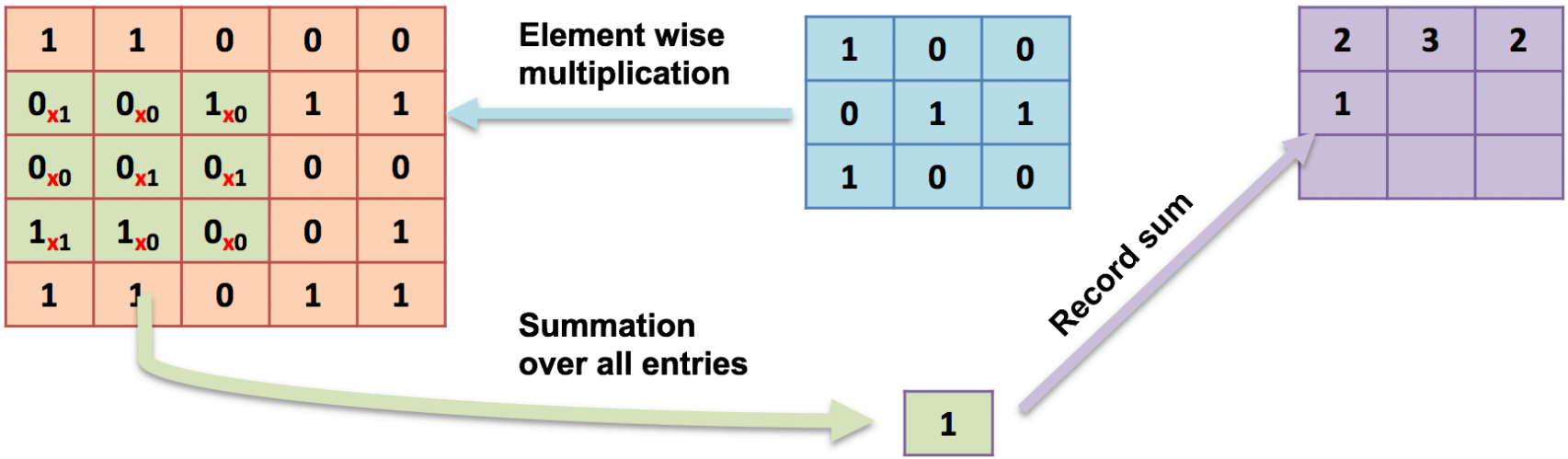


Figure: Filling in the fourth cell.

Convolving 5-9

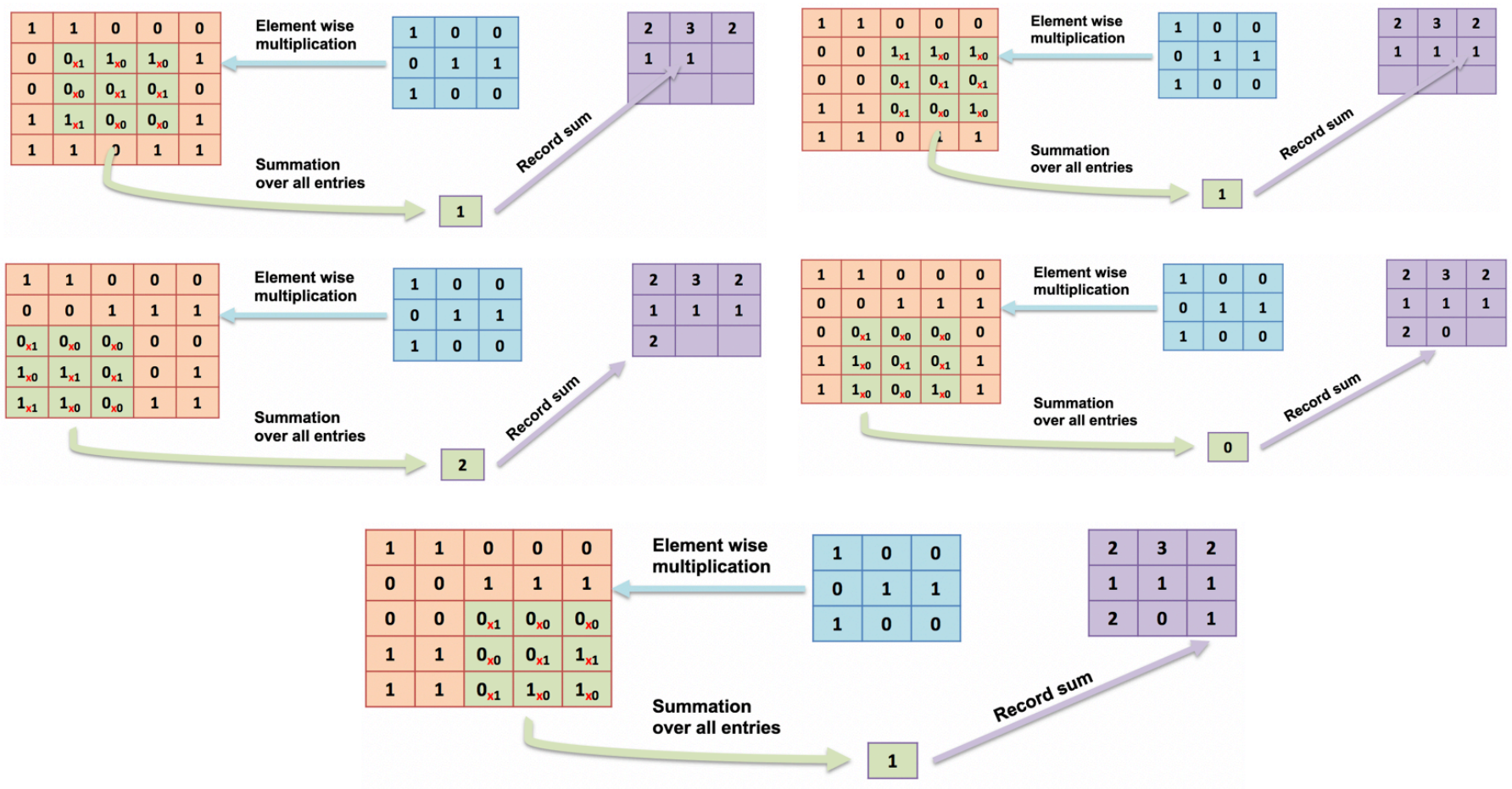


Figure: Filling in the fifth-to-ninth cells.

Convolving With 10 Kernels

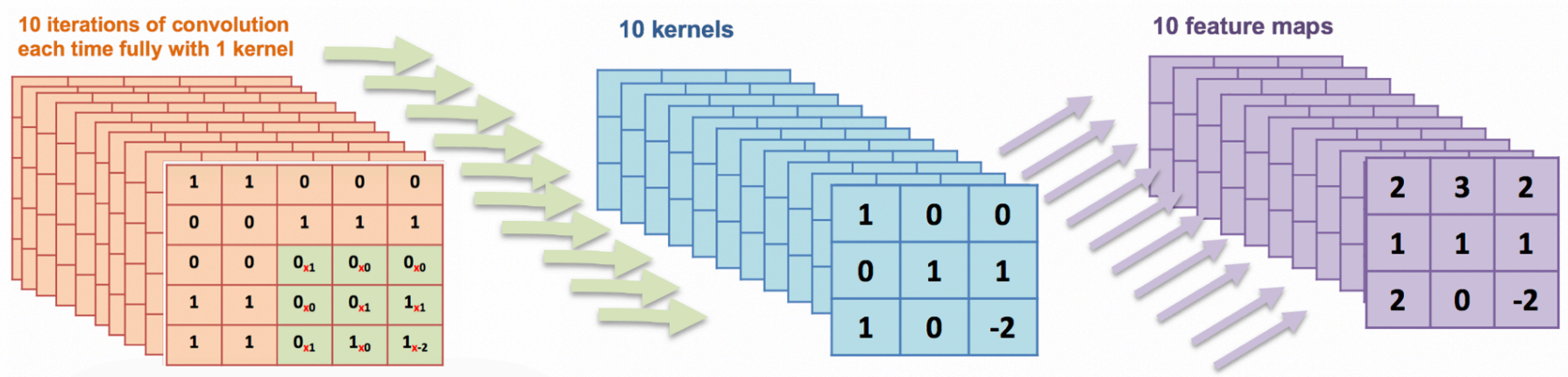


Figure: Convolving with 10 kernels, each with different values, we acquire 10 different feature maps.

Non-Linearity

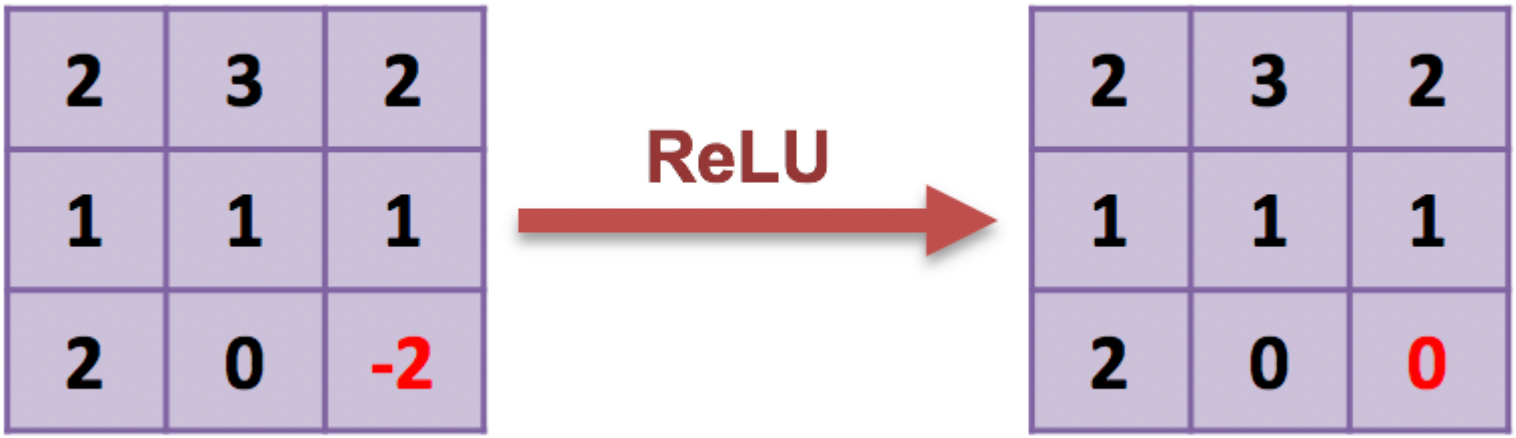


Figure: We actually apply a non-linear function like ReLU on the output of convolution operations such that we end up with meaningful feature maps.

Convolution With Non-Linearity

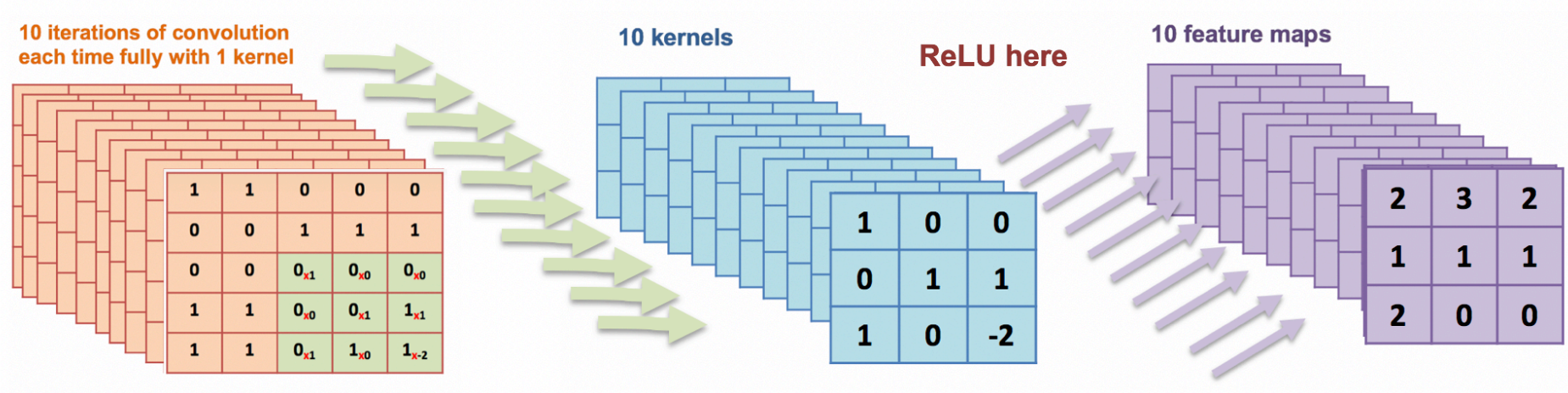


Figure: We actually apply a non-linear function like ReLU on the output of convolution operations such that we end up with meaningful feature maps.

A larger feature map

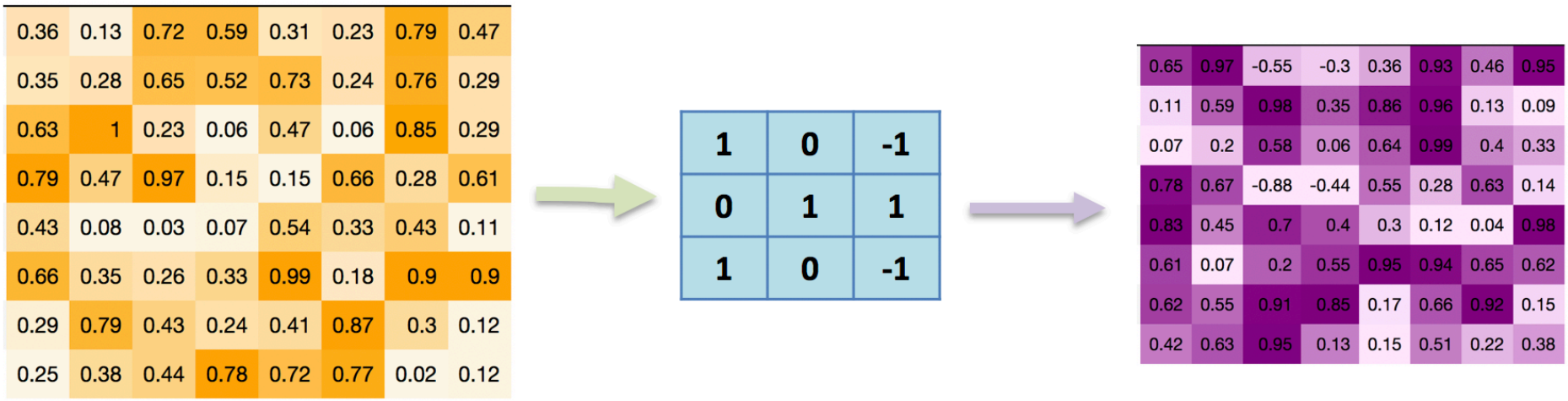


Figure: A larger feature map, still with negative values.

ReLU Applied to Feature Map

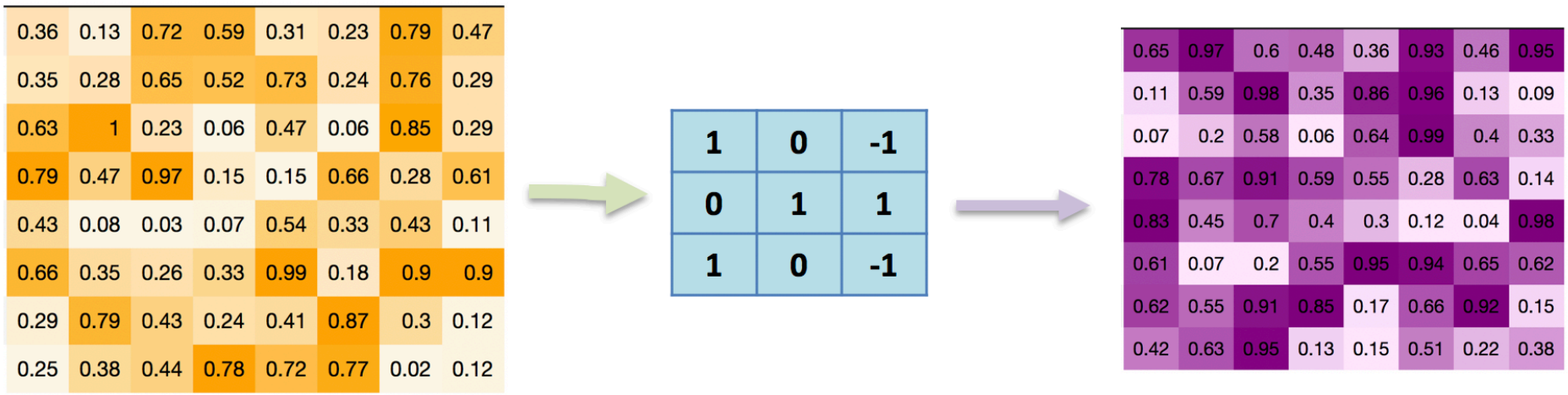


Figure: Feature map from previous slide, with ReLU applied.

Several Feature Maps

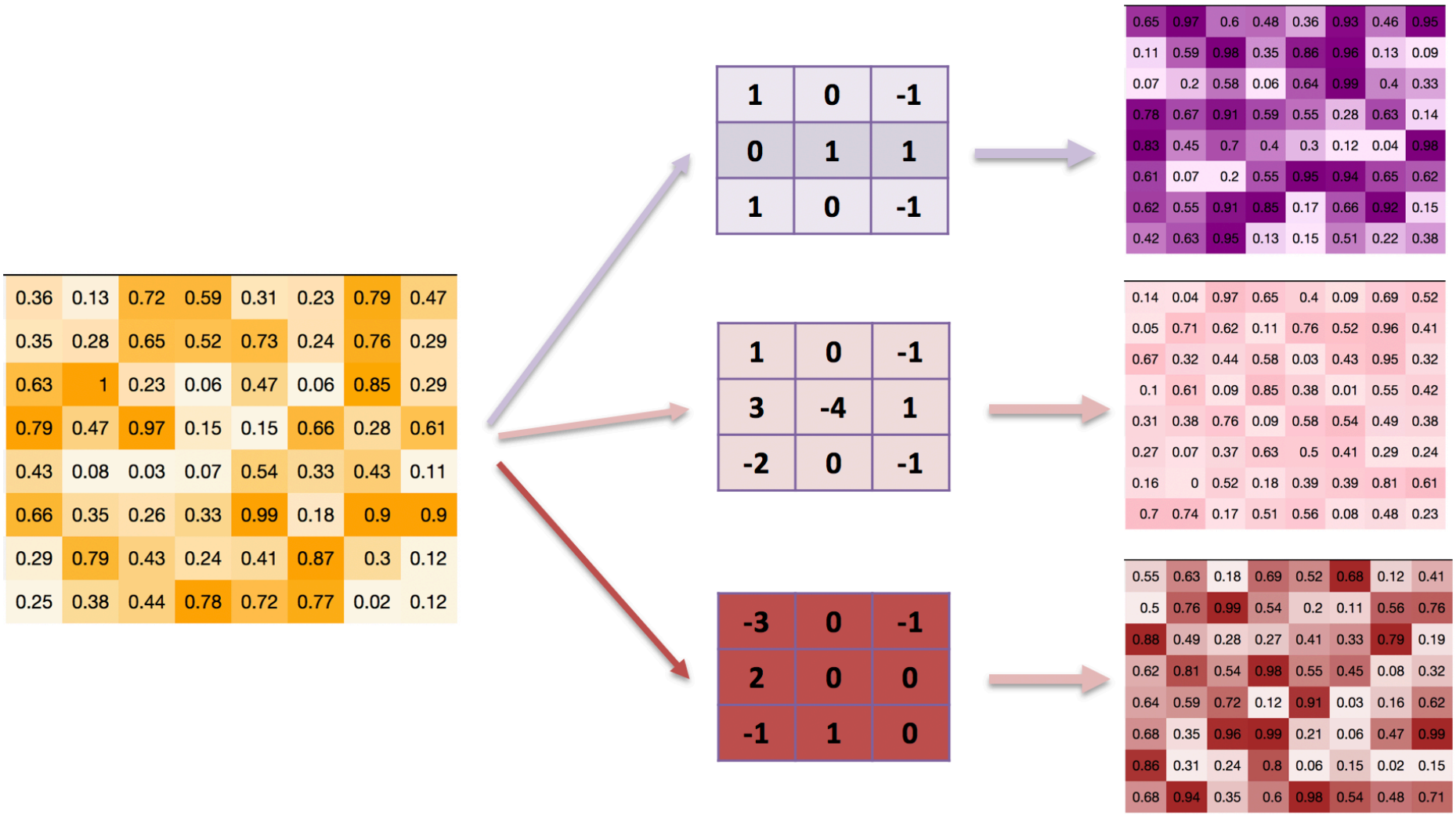


Figure: We acquire several feature maps (number is a hyperprameter) from input.

Feature Maps Stacked

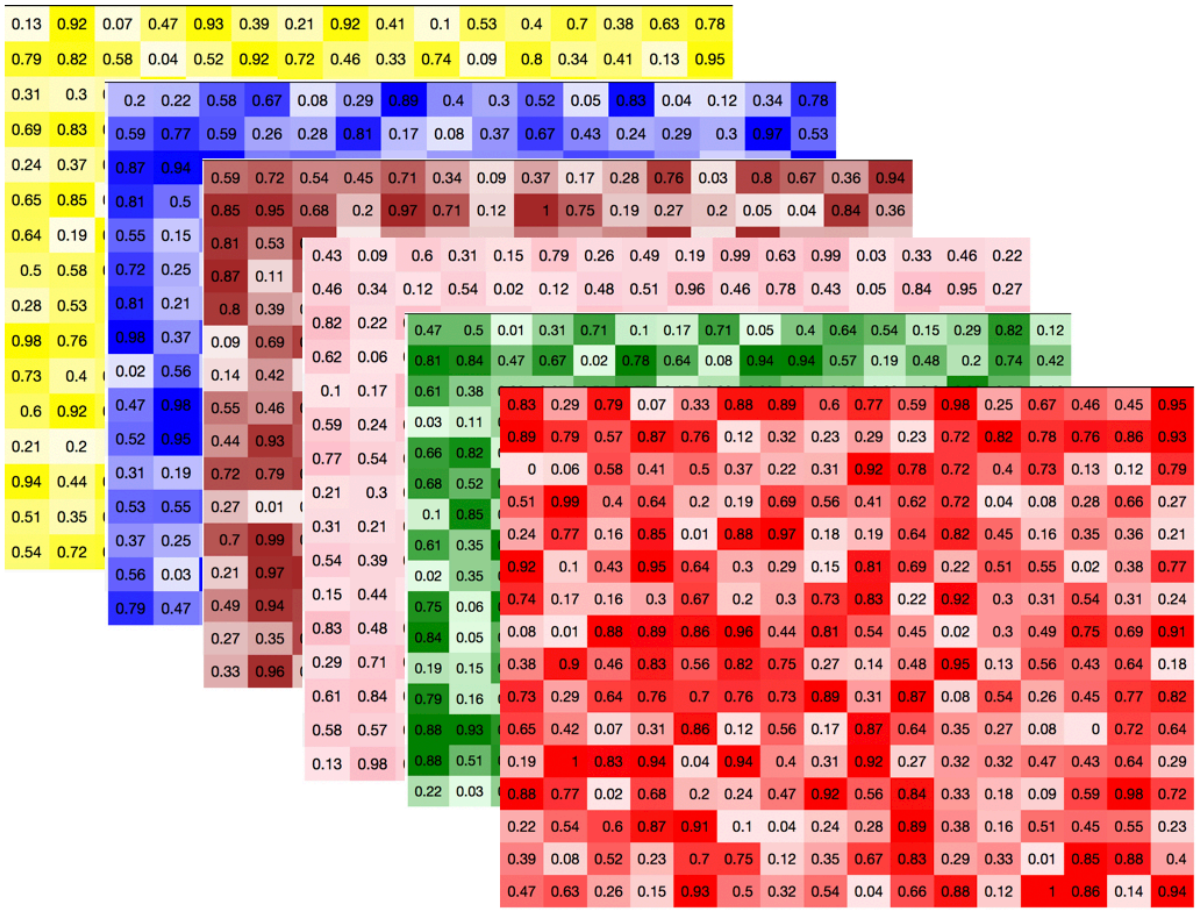


Figure: Several feature maps stacked

Max Pooling

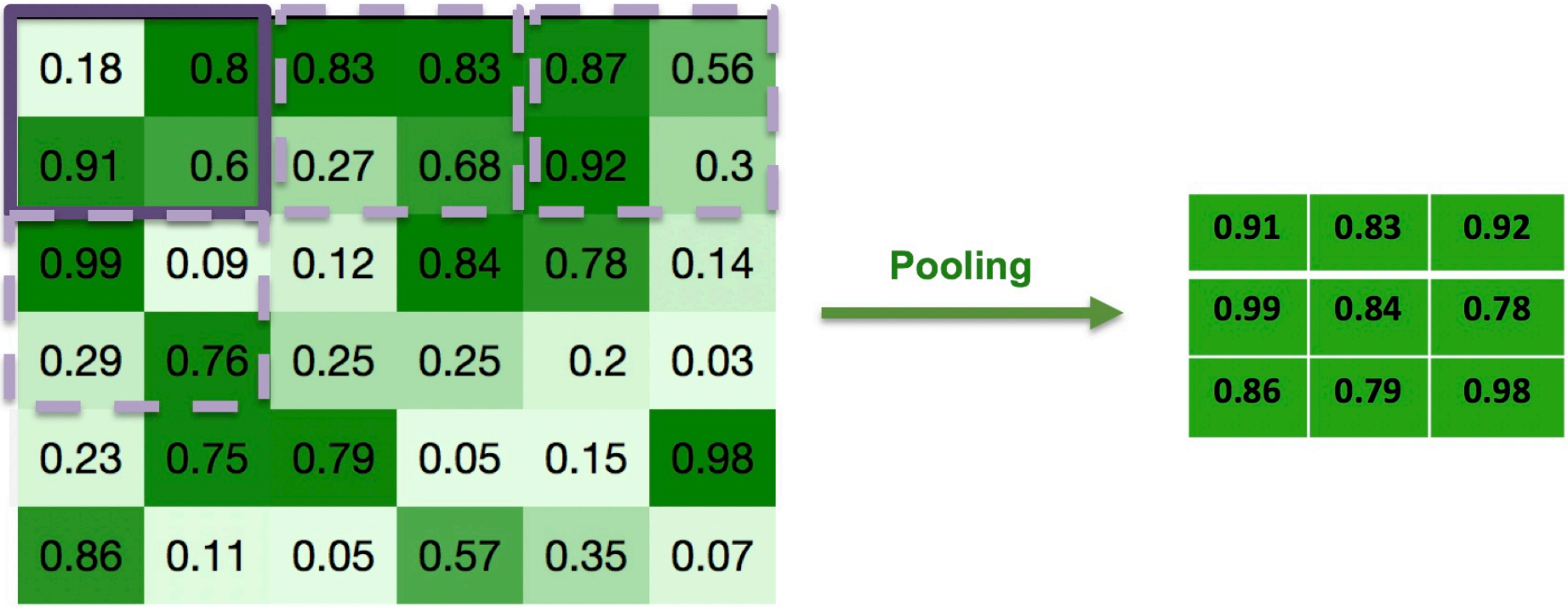


Figure: Max pooling (with a stride of 2) on one feature map

Bigger Picture

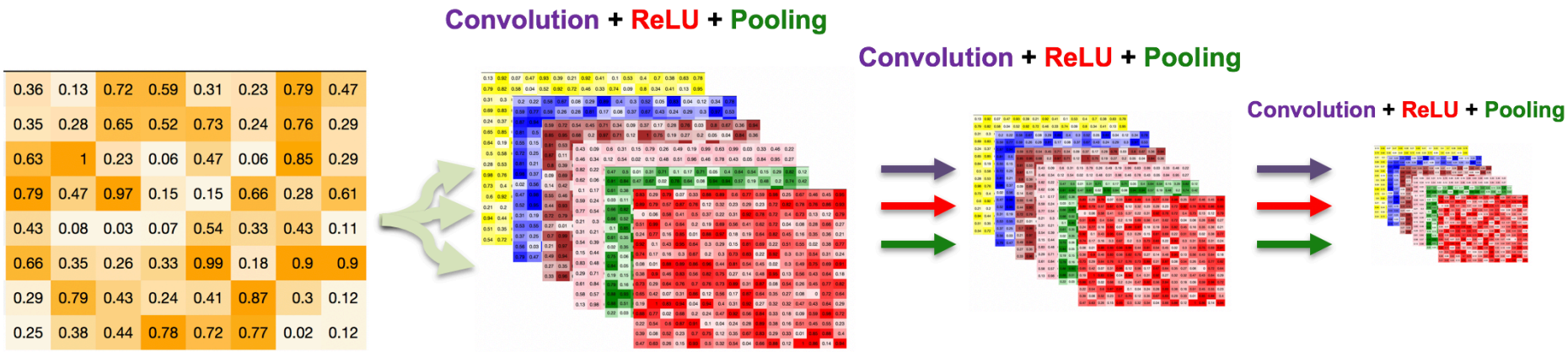


Figure: Convolution and pooling layers, performed on several feature maps

Fully Connected Layer

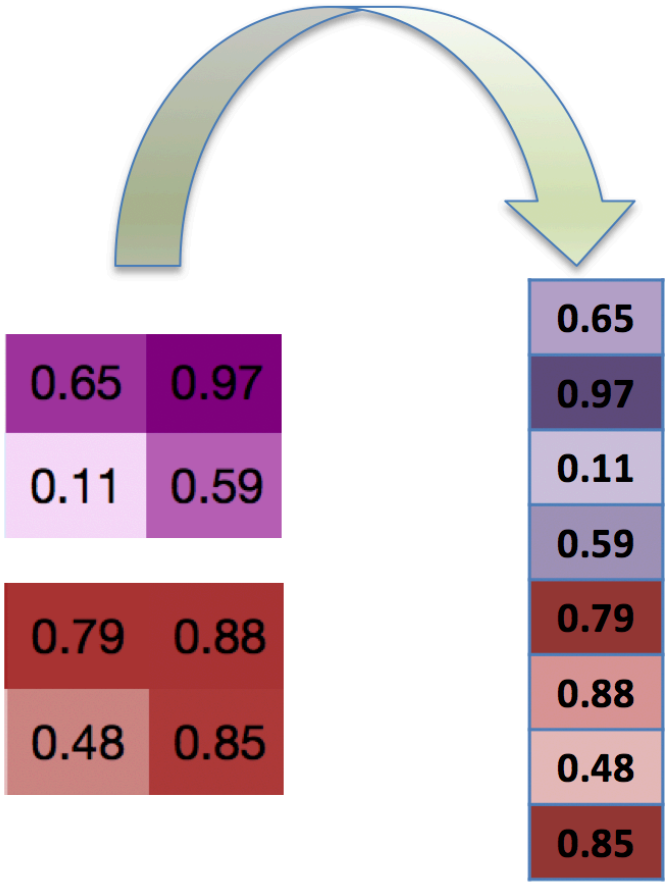


Figure: We collapse last layer of feature maps to one fully connected layer.

Network With Fully Connected Layer

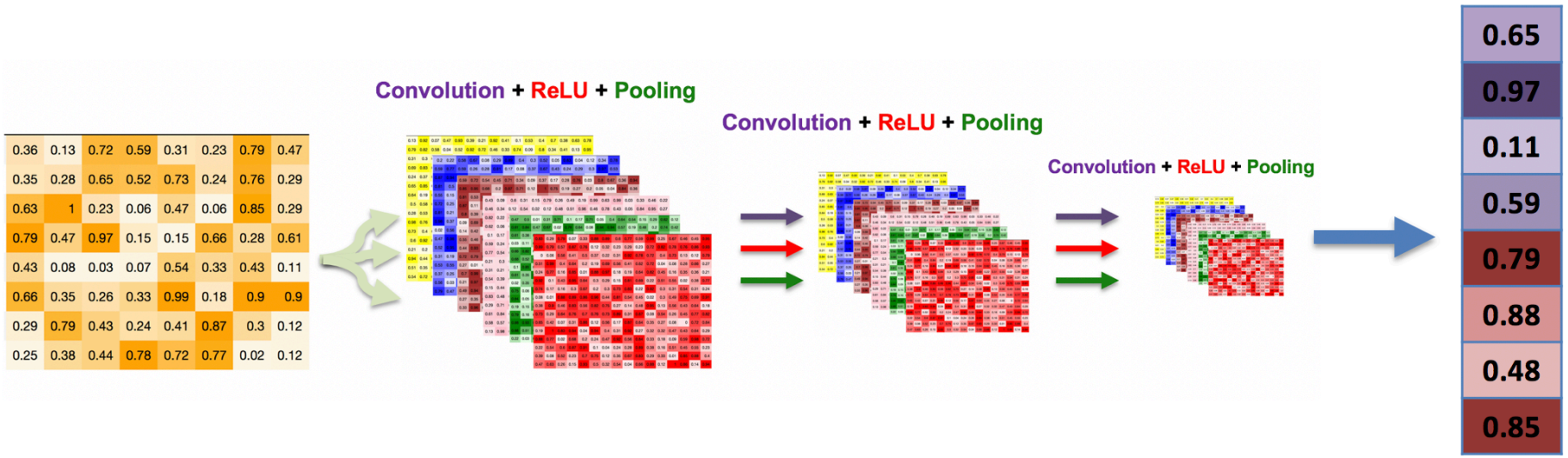


Figure: The network with a fully connected layer.

Motivations for Convolution

- In addition to it providing a means to work with inputs of variable size, convolution offers a number of advantages:

Advantages of Convolution

- **Sparse interactions**
- **Parameter sharing**
- **Equivariant representations**

Sparse Interactions

Sparse Interactions

- CNN uses a **kernel smaller** than its input units, giving **outputs that are not the result of an interaction with each input unit**.
- **Huge computational gains** (we need to store fewer parameters)

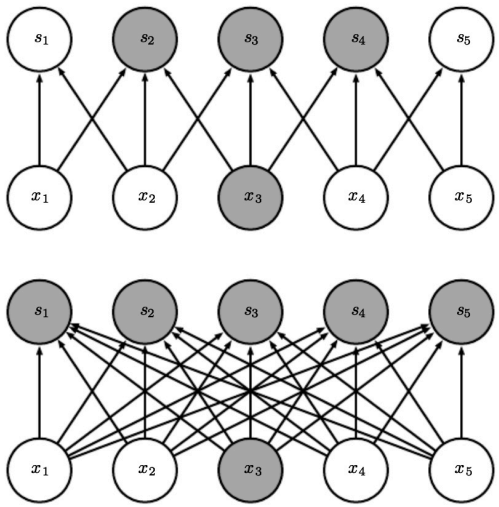


Figure: *(*Top): When s is formed by **convolution**, with a kernel of width 3, only 3 output units are affected by x. *(*Top): When s is formed by matrix multiplication, **connectivity is no longer sparse**. [Goodfellow et al. (2016, p. 325)].

Parameter Sharing

Parameter Sharing

- In a traditional neural net, each element of the weight matrix is multiplied by one element of the input and then **never revisited** (never used again).
- ConvNets have shared (or tied weights), where **each member of the kernel is used at every position of the input**.

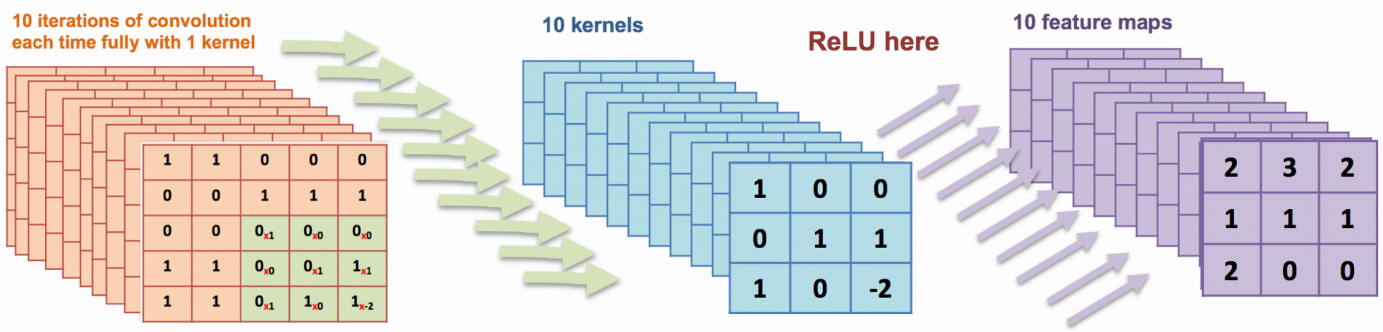


Figure: Each member of a given kernel is used at every position of the input

Equivariant Representations

Equivariance

- The form of parameter sharing in convolution causes the layer to have equivarience.
- Equivariant representations (or equivarience to translation) in a given function means that **if the input changes, the output changes the same way**.
- If a **feature moves later in input, its exact same representation will still appear in the output, just later**
- **Useful for image**, when we detect edges, since the same edges will appear in different parts.
- Can be useful in language, e.g., when we **move character sequences**.

Example CNN For Sentence Classification

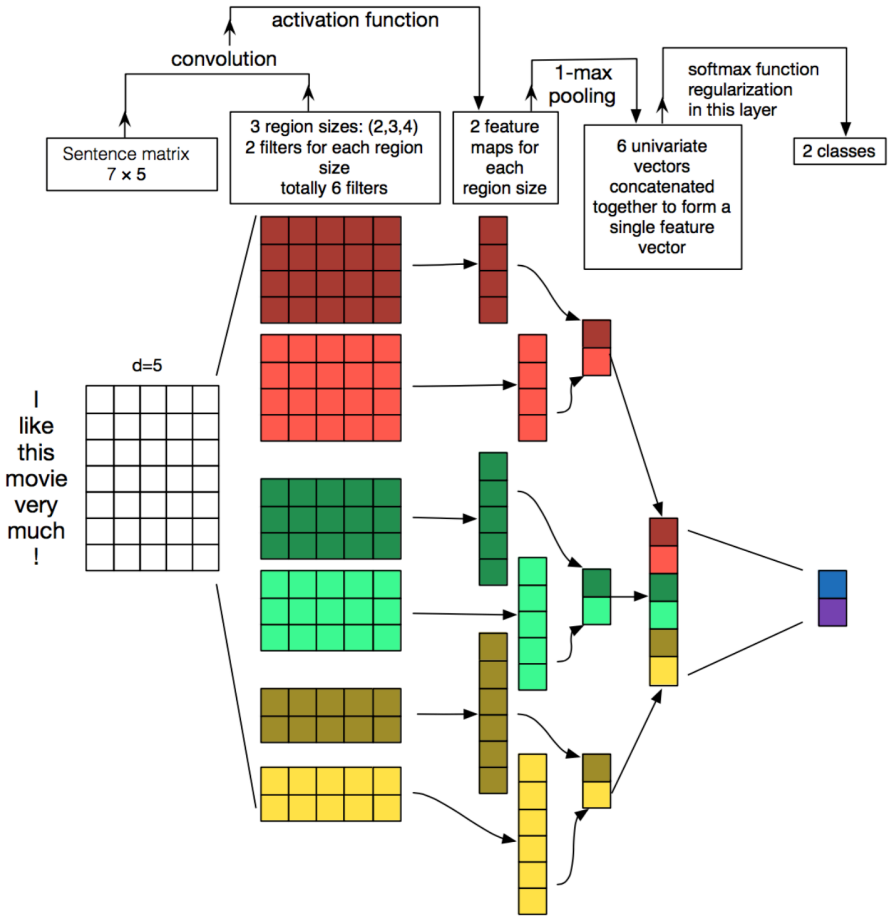


Figure: [From Zhang, Y., Wallace, B. (2015)]