

# Convolutional Neural Network

Boris Zubarev



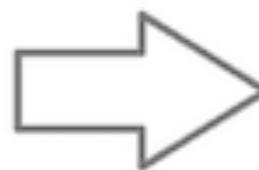
@bobazooba

# MLP Disadvantages

- A lot of parameters to solve task (neurons)
- Highly overfitting

# Convolutional Neural Network

|   |   |   |
|---|---|---|
| 1 | 1 | 0 |
| 4 | 2 | 1 |
| 0 | 2 | 1 |



|   |
|---|
| 1 |
| 1 |
| 0 |
| 4 |
| 2 |
| 1 |
| 0 |
| 2 |
| 1 |

# Convolutional Neural Network

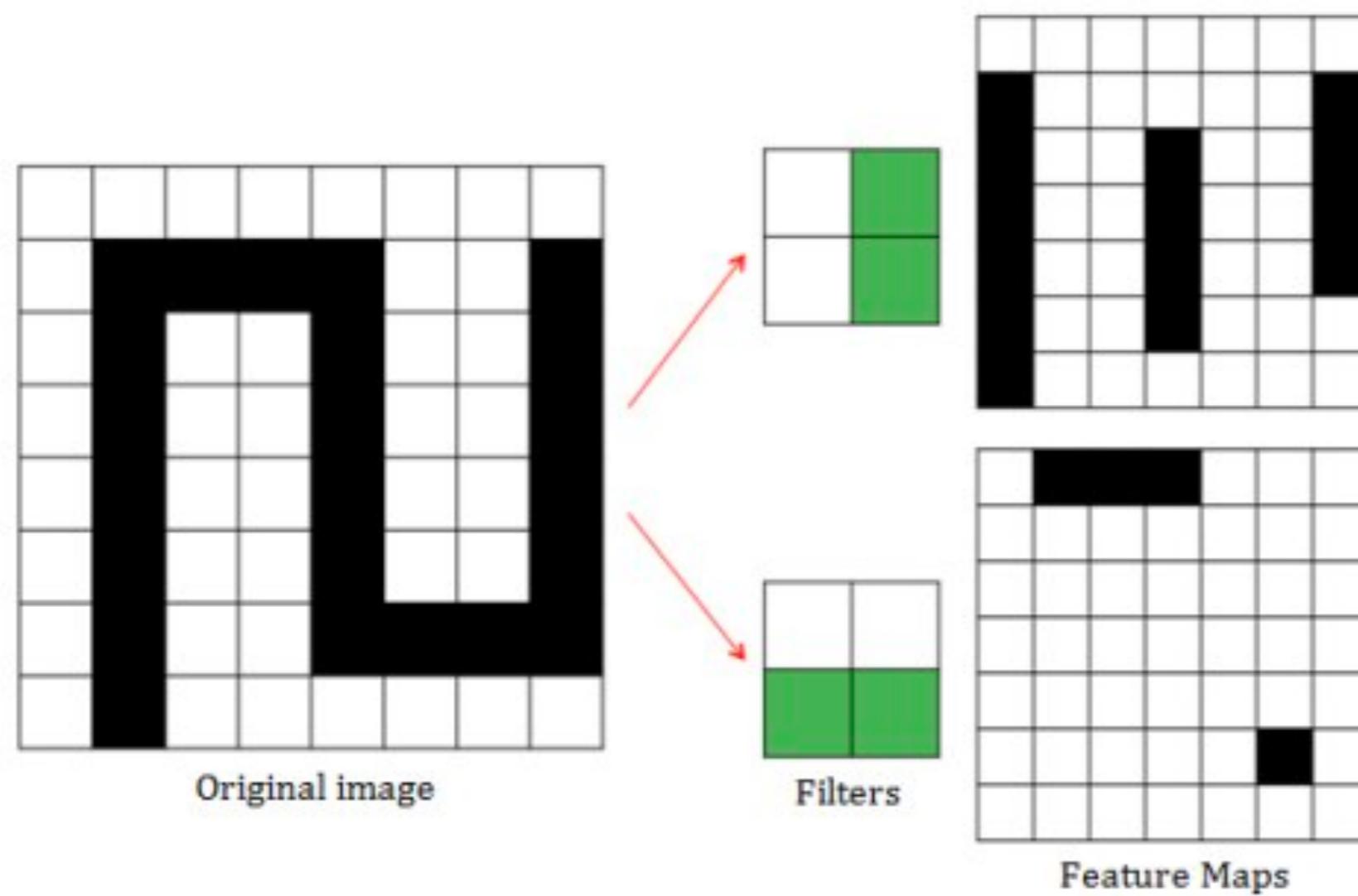
|                        |                        |                        |   |   |
|------------------------|------------------------|------------------------|---|---|
| 1<br><small>x1</small> | 1<br><small>x0</small> | 1<br><small>x1</small> | 0 | 0 |
| 0<br><small>x0</small> | 1<br><small>x1</small> | 1<br><small>x0</small> | 1 | 0 |
| 0<br><small>x1</small> | 0<br><small>x0</small> | 1<br><small>x1</small> | 1 | 1 |
| 0                      | 0                      | 1                      | 1 | 0 |
| 0                      | 1                      | 1                      | 0 | 0 |

Image

|   |  |  |
|---|--|--|
| 4 |  |  |
|   |  |  |
|   |  |  |

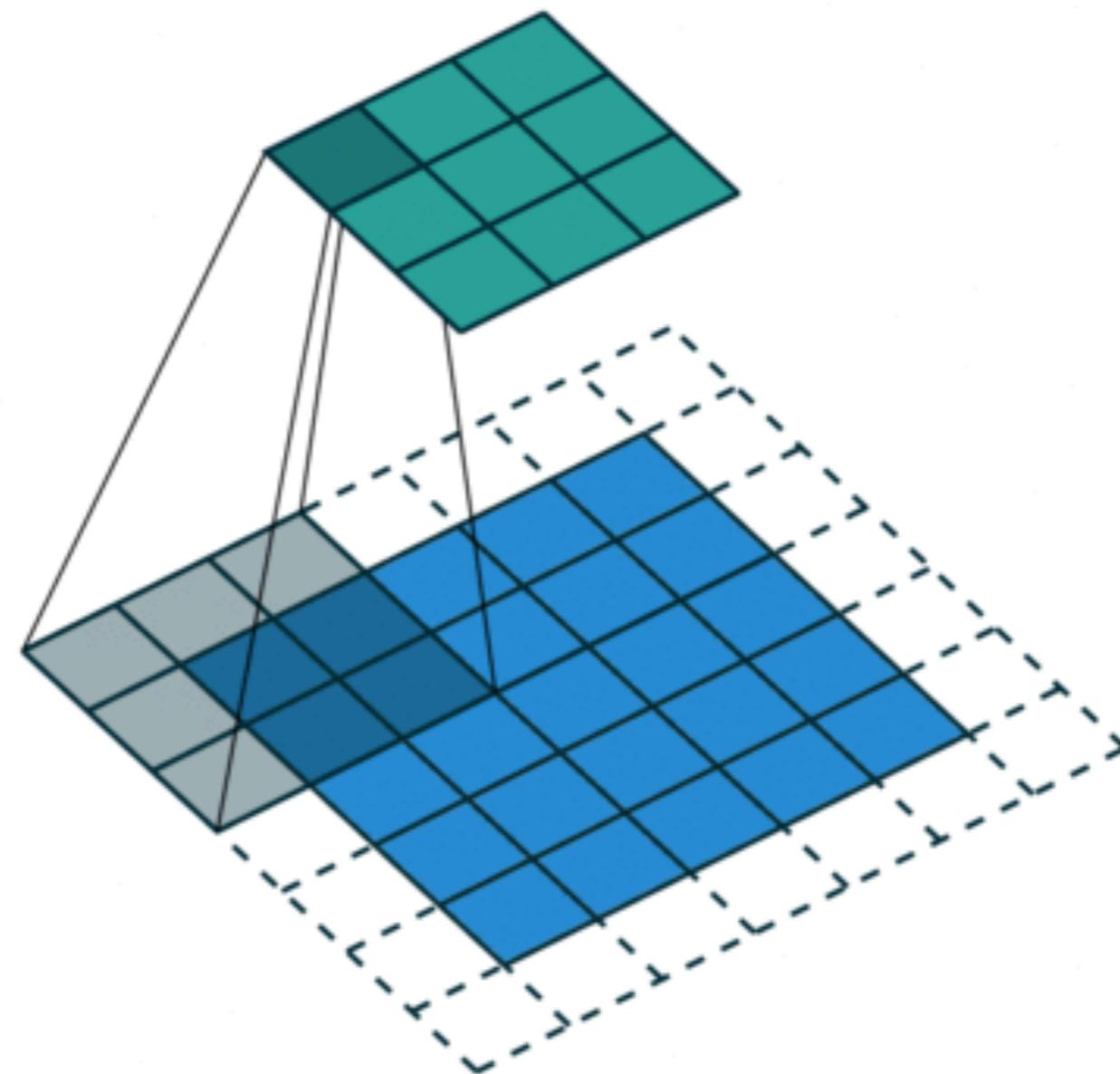
Convolved  
Feature

# Convolutional Neural Network

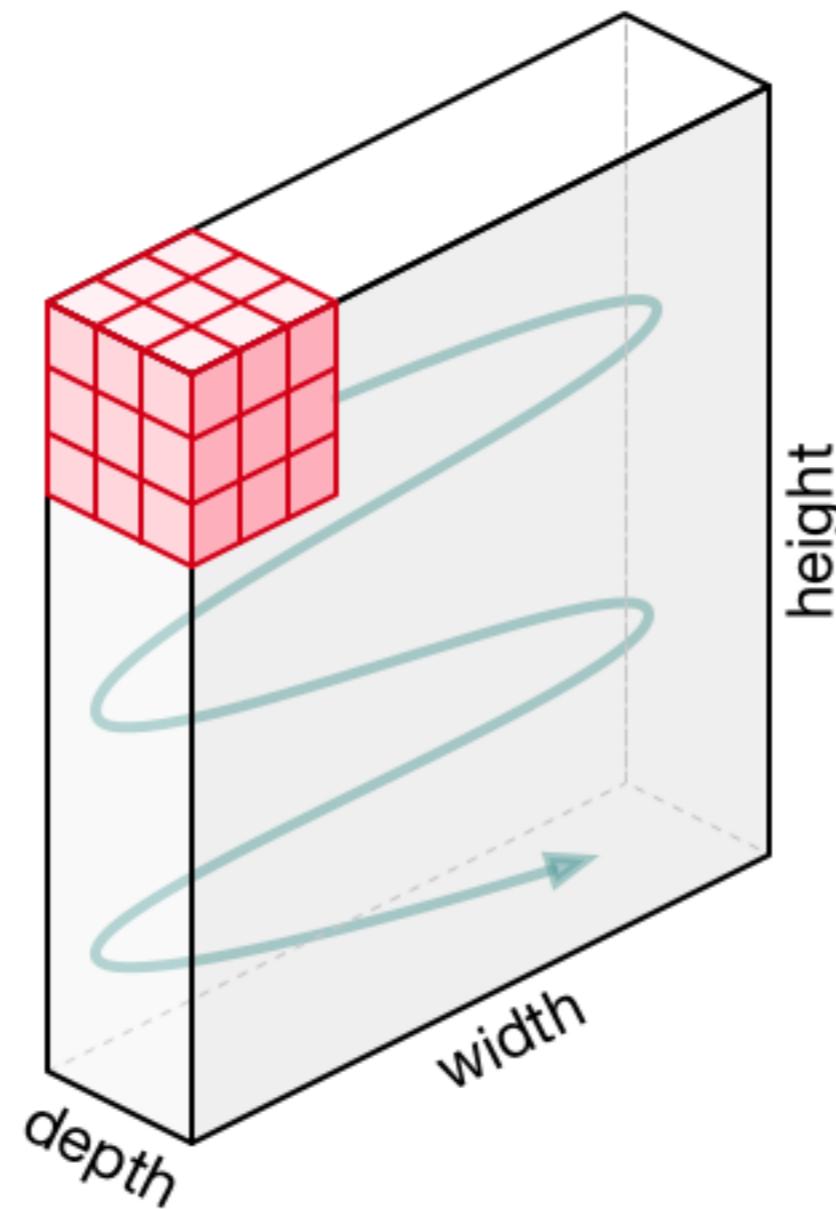


# Convolutional Neural Network

**Stride**



# Convolutional Neural Network



# Convolutional Neural Network

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 0   | 0   | 0   | 0   | 0   | 0   | ... |
| 0   | 156 | 155 | 156 | 158 | 158 | 158 | ... |
| 0   | 153 | 154 | 157 | 159 | 159 | 159 | ... |
| 0   | 149 | 151 | 155 | 158 | 159 | 159 | ... |
| 0   | 146 | 146 | 149 | 153 | 158 | 158 | ... |
| 0   | 145 | 143 | 143 | 148 | 158 | 158 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |

Input Channel #1 (Red)

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 0   | 0   | 0   | 0   | 0   | 0   | ... |
| 0   | 167 | 166 | 167 | 169 | 169 | 169 | ... |
| 0   | 164 | 165 | 168 | 170 | 170 | 170 | ... |
| 0   | 160 | 162 | 166 | 169 | 170 | 170 | ... |
| 0   | 156 | 156 | 159 | 163 | 168 | 168 | ... |
| 0   | 155 | 153 | 153 | 158 | 168 | 168 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |

Input Channel #2 (Green)

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 0   | 0   | 0   | 0   | 0   | 0   | ... |
| 0   | 163 | 162 | 163 | 165 | 165 | 165 | ... |
| 0   | 160 | 161 | 164 | 166 | 166 | 166 | ... |
| 0   | 156 | 158 | 162 | 165 | 166 | 166 | ... |
| 0   | 155 | 155 | 158 | 162 | 167 | 167 | ... |
| 0   | 154 | 152 | 152 | 157 | 167 | 167 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |

Input Channel #3 (Blue)

|    |    |    |
|----|----|----|
| -1 | -1 | 1  |
| 0  | 1  | -1 |
| 0  | 1  | 1  |

Kernel Channel #1



308

|   |    |    |
|---|----|----|
| 1 | 0  | 0  |
| 1 | -1 | -1 |
| 1 | 0  | -1 |

Kernel Channel #2



-498

|   |    |   |
|---|----|---|
| 0 | 1  | 1 |
| 0 | 1  | 0 |
| 1 | -1 | 1 |

Kernel Channel #3



164

$$+ 1 = -25$$

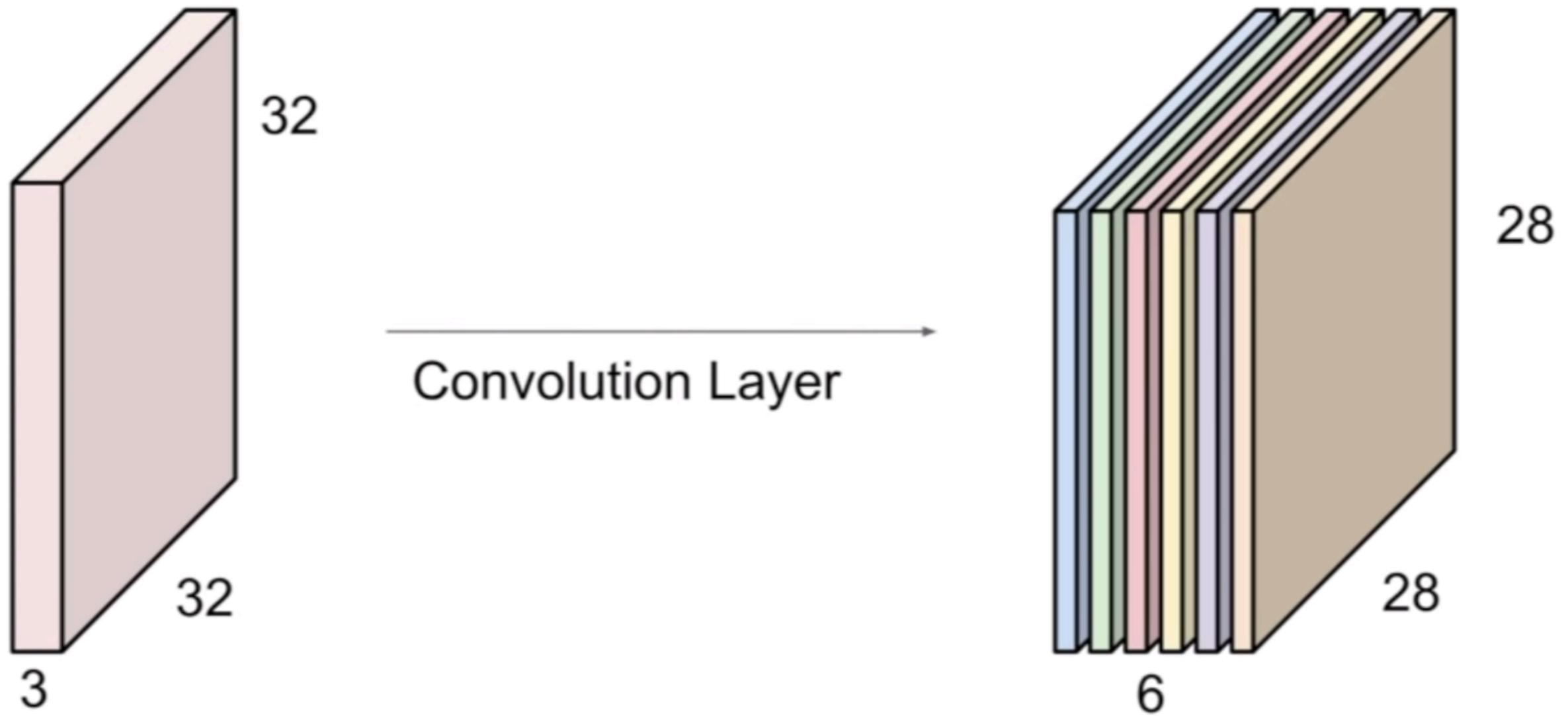


Bias = 1

|     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|
| -25 |     |     |     |     | ... |
|     |     |     |     |     | ... |
|     |     |     |     |     | ... |
|     |     |     |     |     | ... |
| ... | ... | ... | ... | ... | ... |

Output

# Convolutional Neural Network

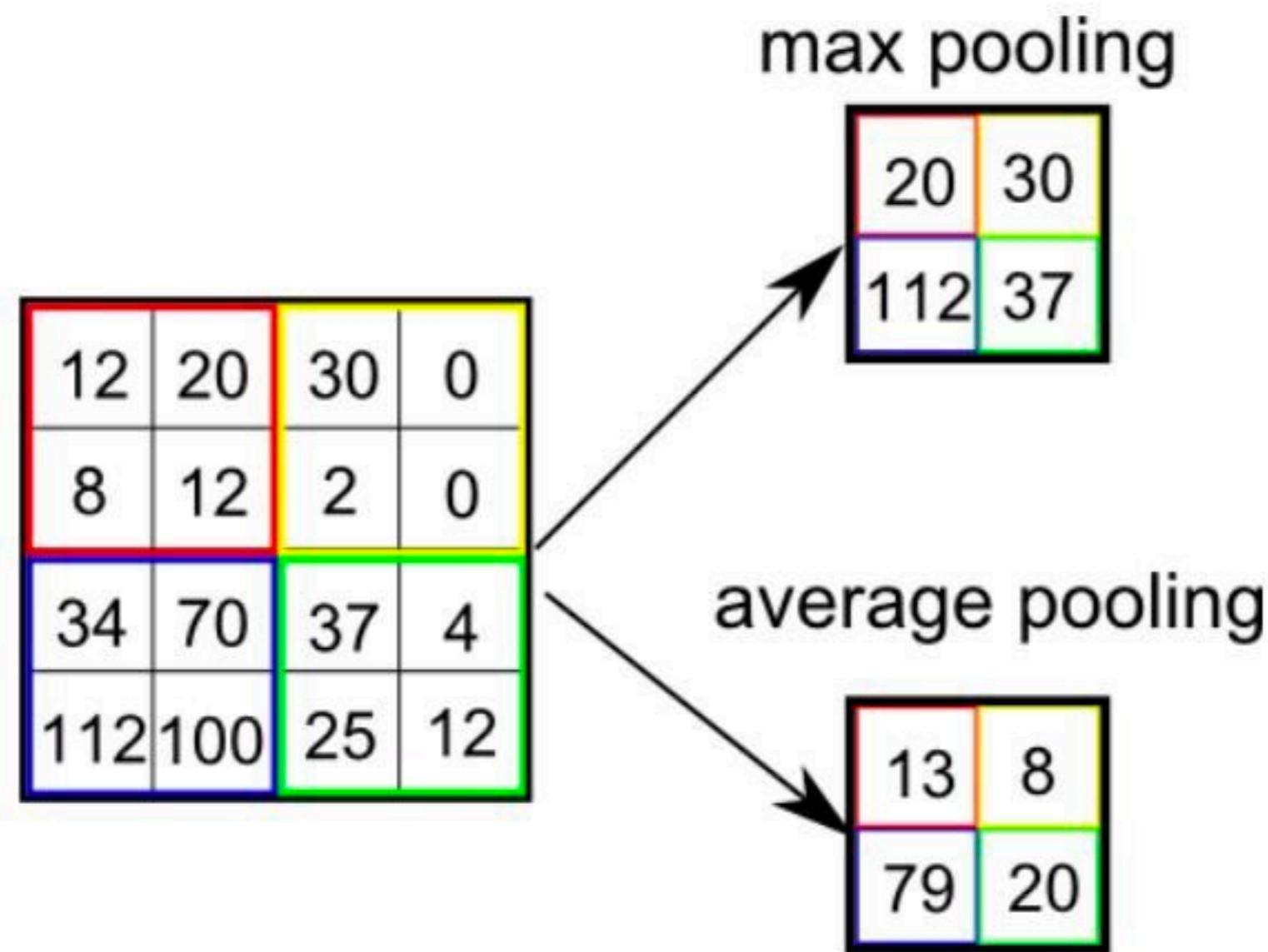


# Convolutional Neural Network

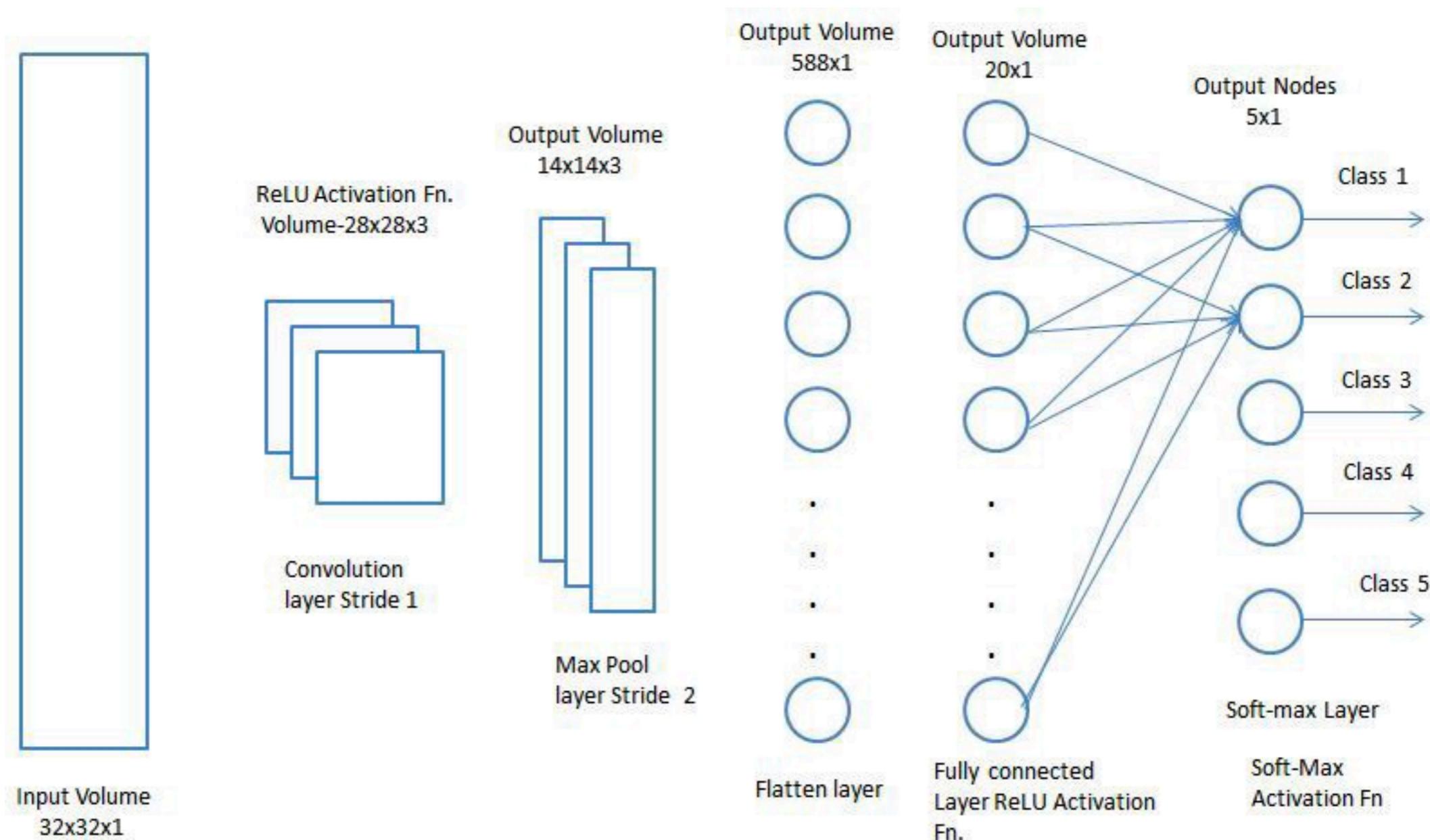
|     |     |     |
|-----|-----|-----|
| 3.0 | 3.0 | 3.0 |
| 3.0 | 3.0 | 3.0 |
| 3.0 | 2.0 | 3.0 |

|   |   |   |   |   |
|---|---|---|---|---|
| 3 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 3 | 1 |
| 3 | 1 | 2 | 2 | 3 |
| 2 | 0 | 0 | 2 | 2 |
| 2 | 0 | 0 | 0 | 1 |

# Convolutional Neural Network

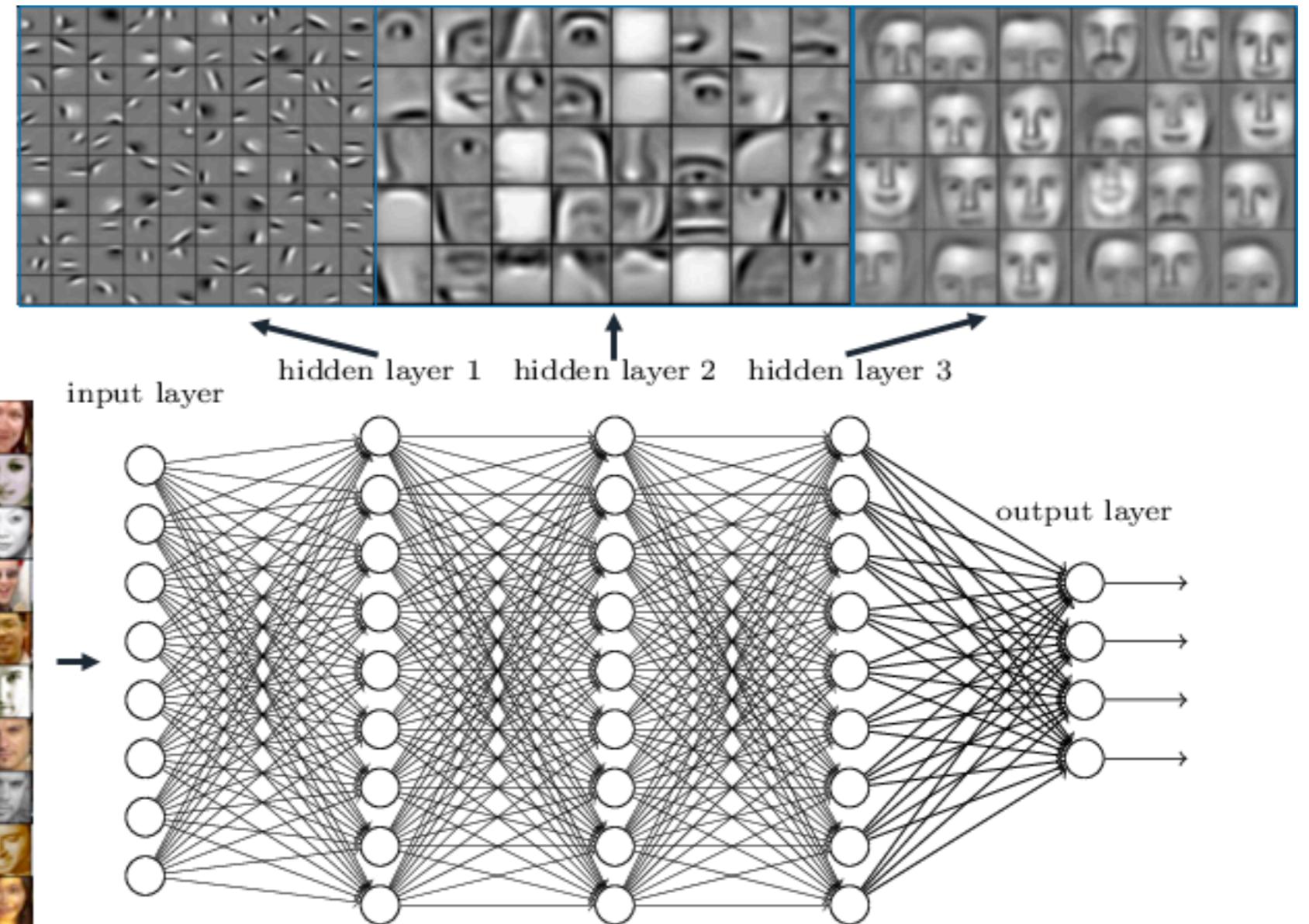


# Convolutional Neural Network



# Convolutional Neural Network

Deep neural networks learn hierarchical feature representations



# Convolutional Neural Network

embedding\_dim = 5

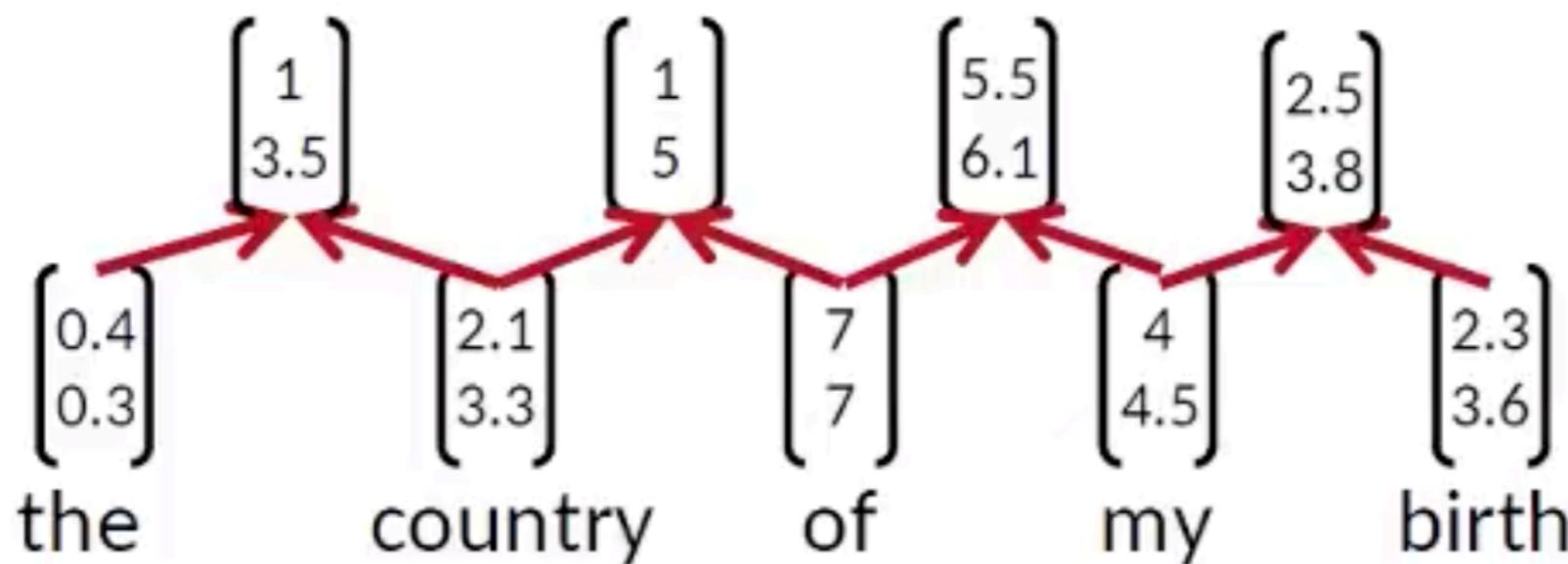
I  
like  
this  
movie  
very  
much  
!

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Convolutional Neural Network



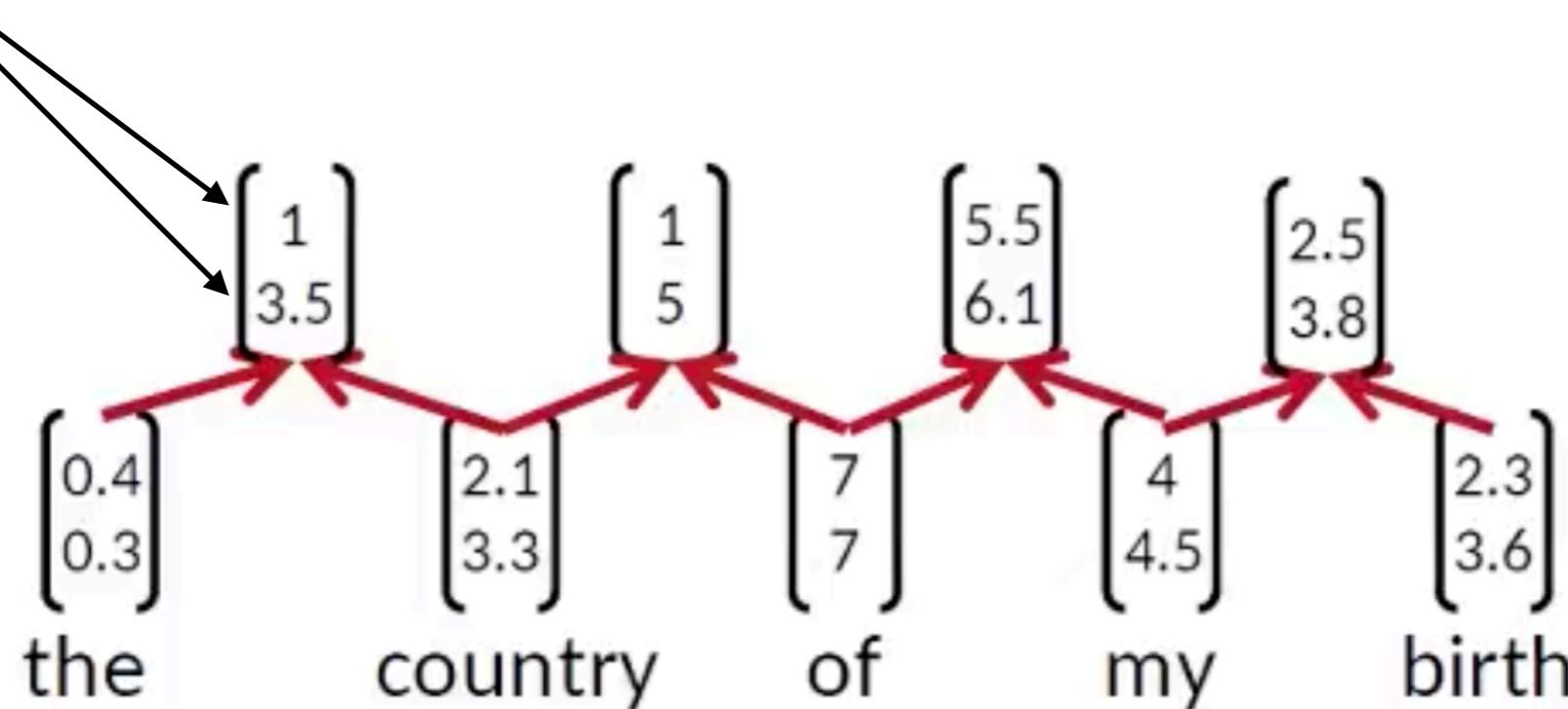
# Convolutional Neural Network



$$p = \tanh \left( W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b \right)$$

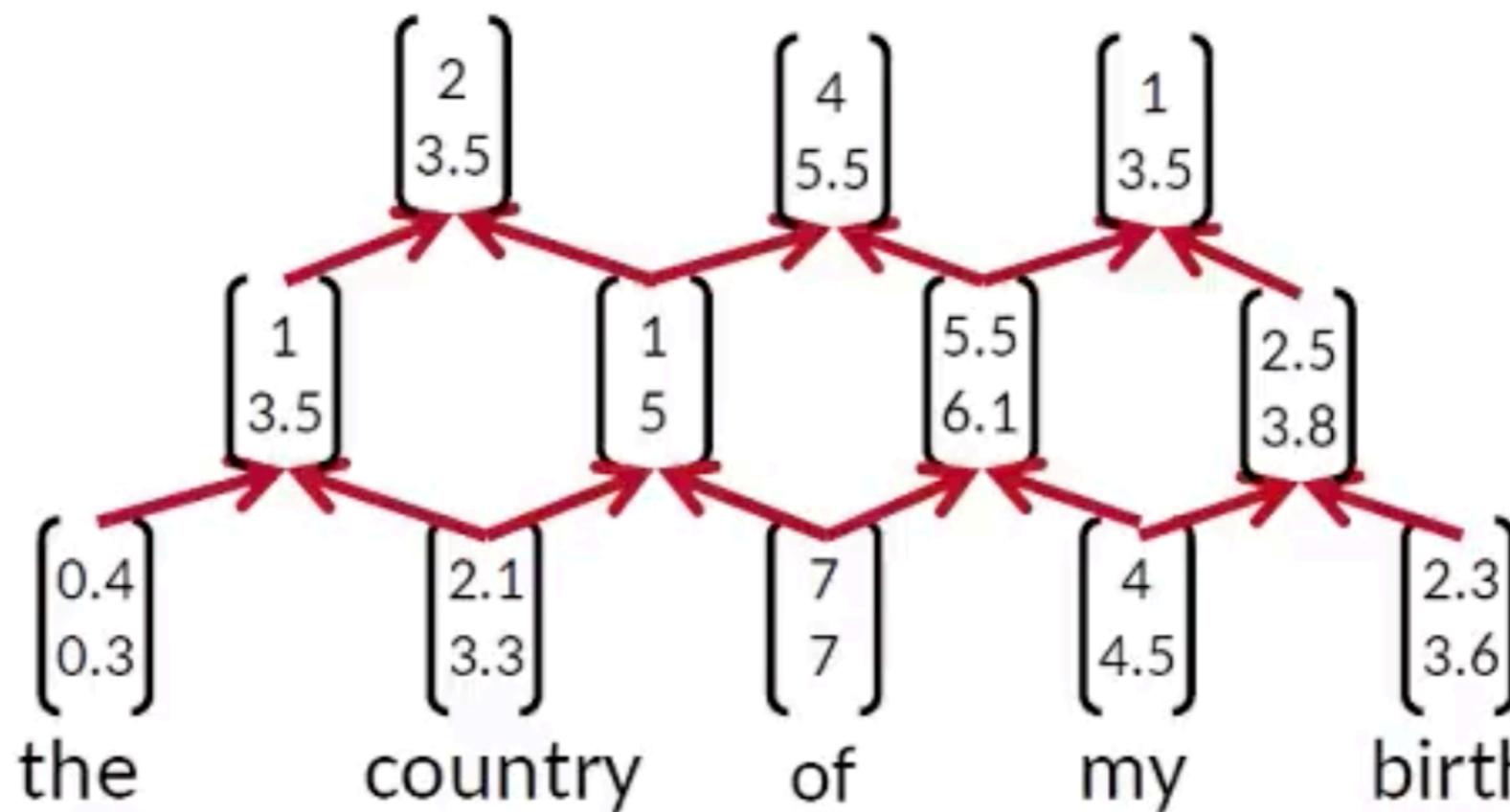
# Convolutional Neural Network

## Multiple filters



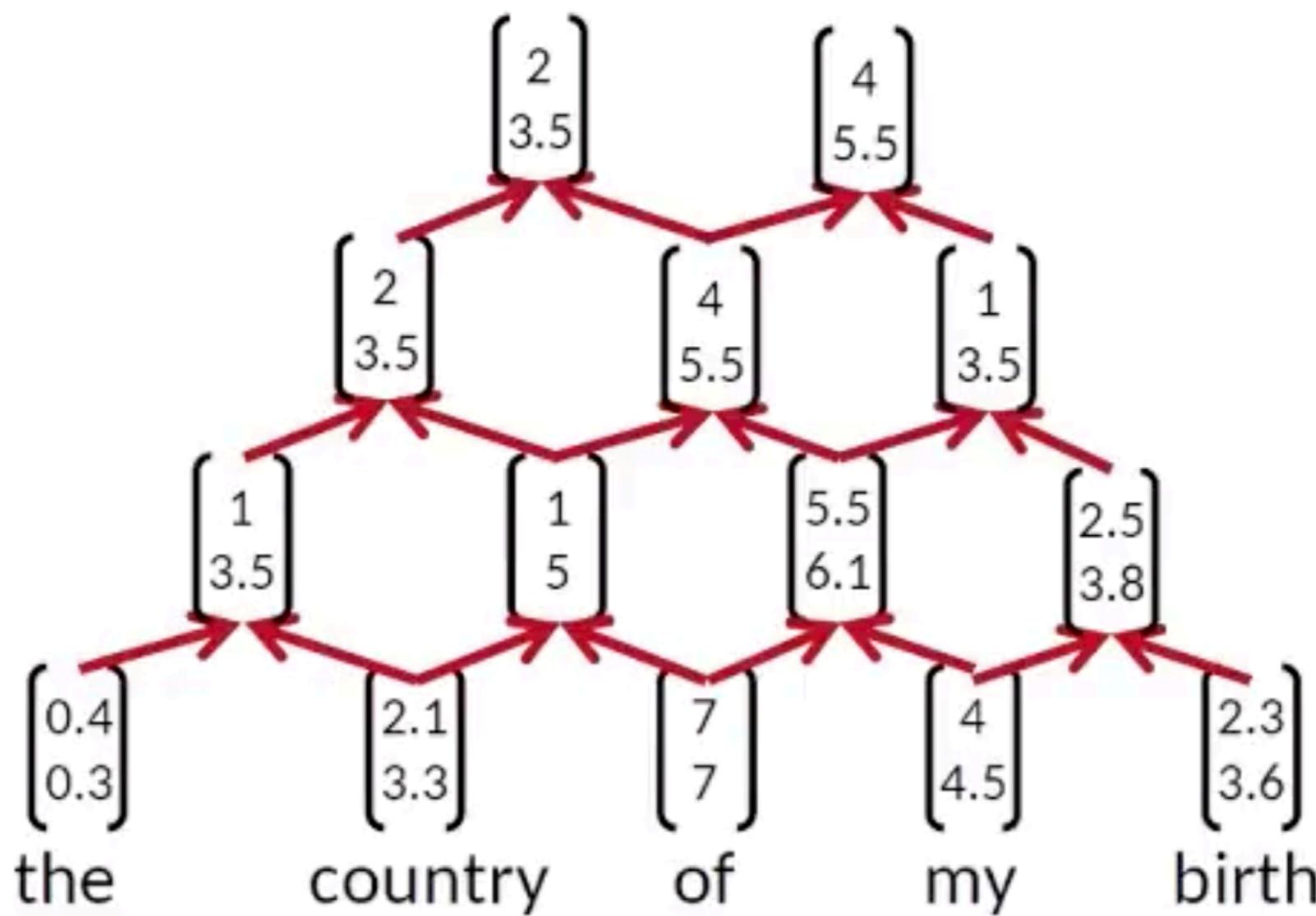
$$p = \tanh \left( W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b \right)$$

# Convolutional Neural Network

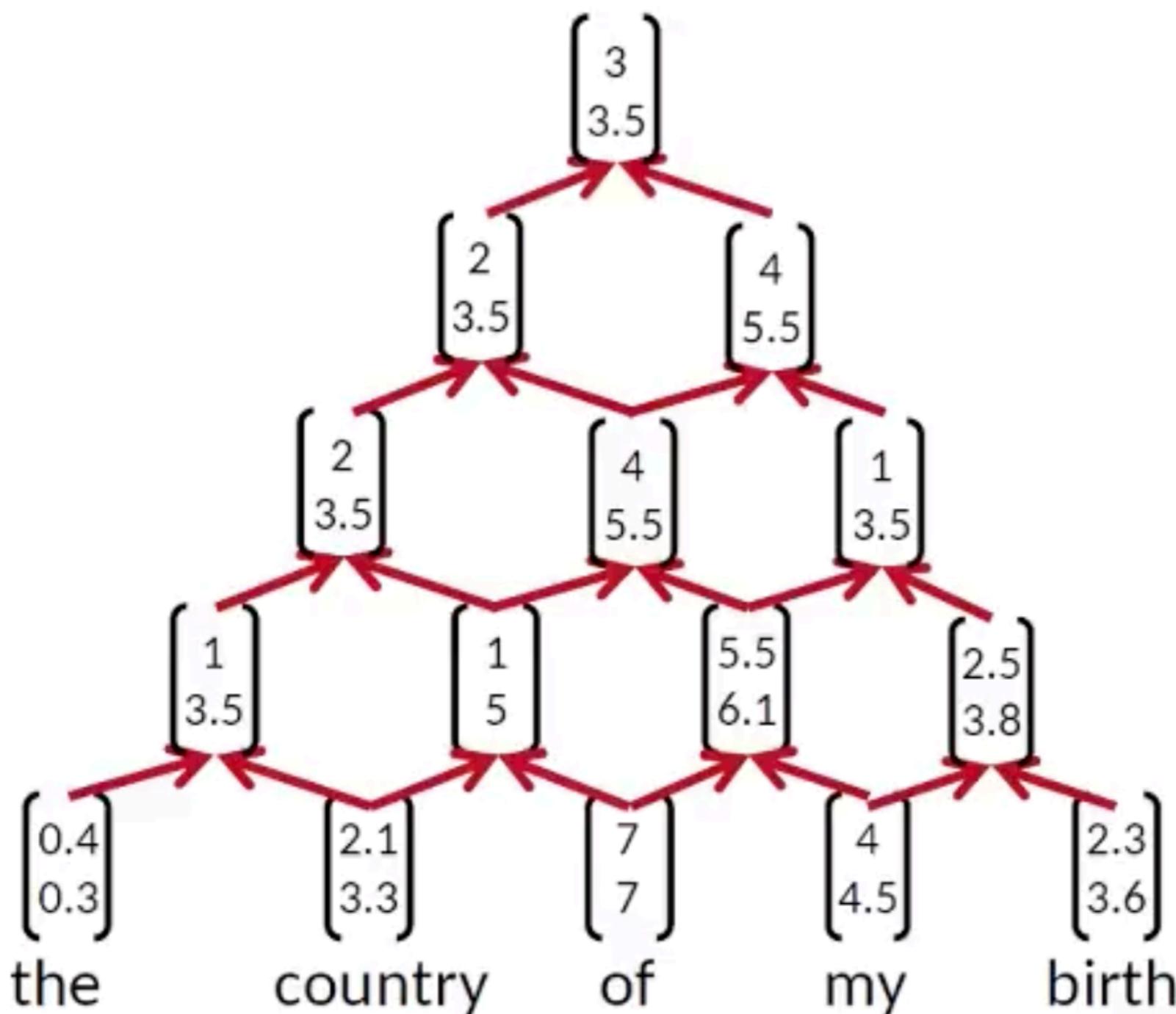


$$p = \tanh \left( W^{(2)} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b \right)$$

# Convolutional Neural Network

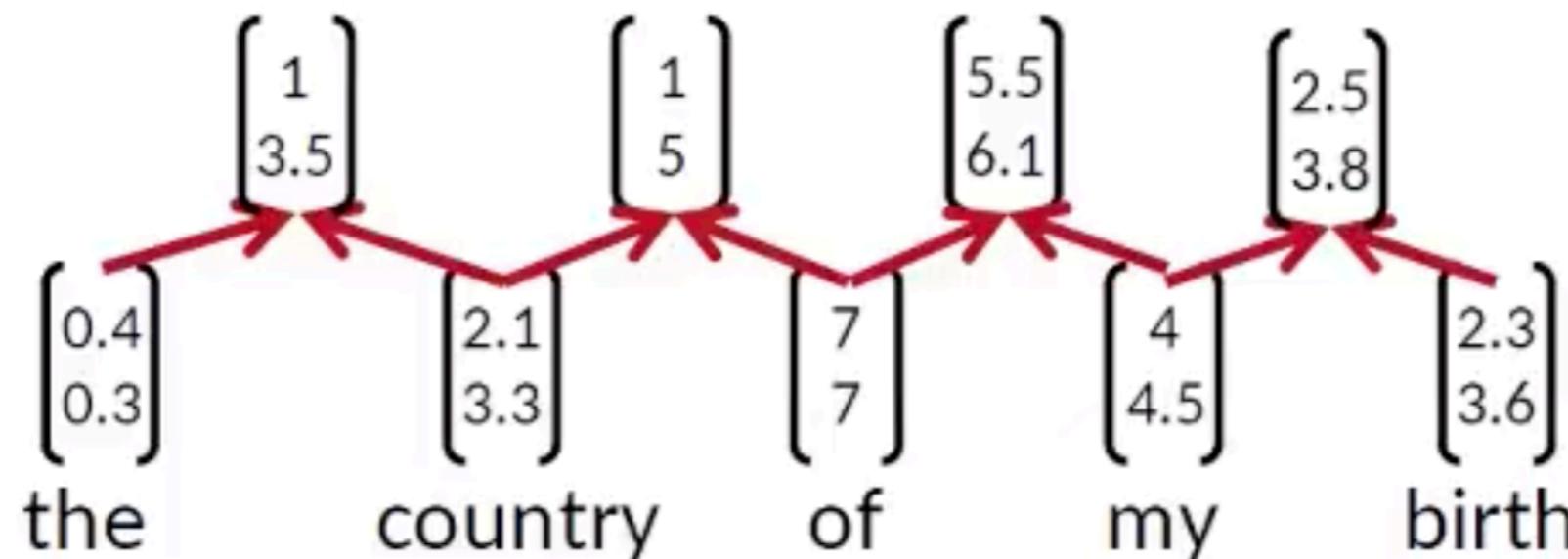


# Convolutional Neural Network

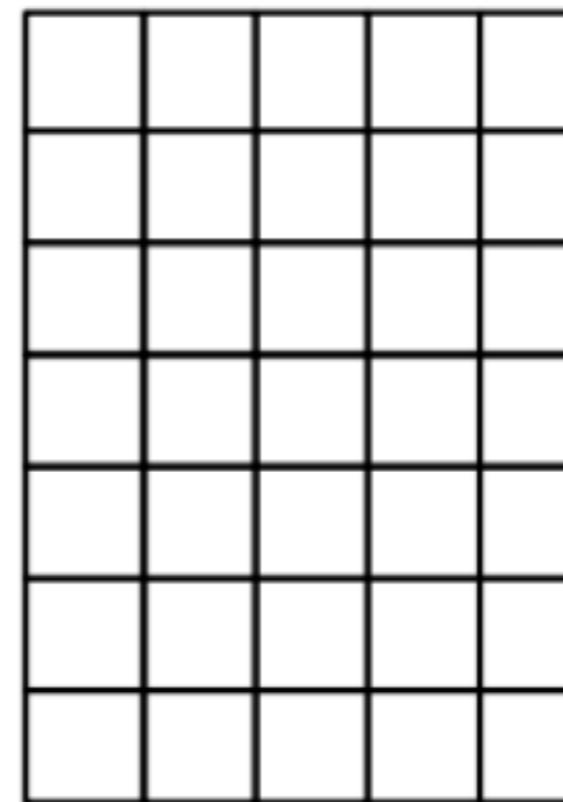


# Convolutional Neural Network

Max over time pooling



I  
like  
this  
movie  
very  
much  
!



# Convolutional Neural Network

Max over time pooling

**Мама**

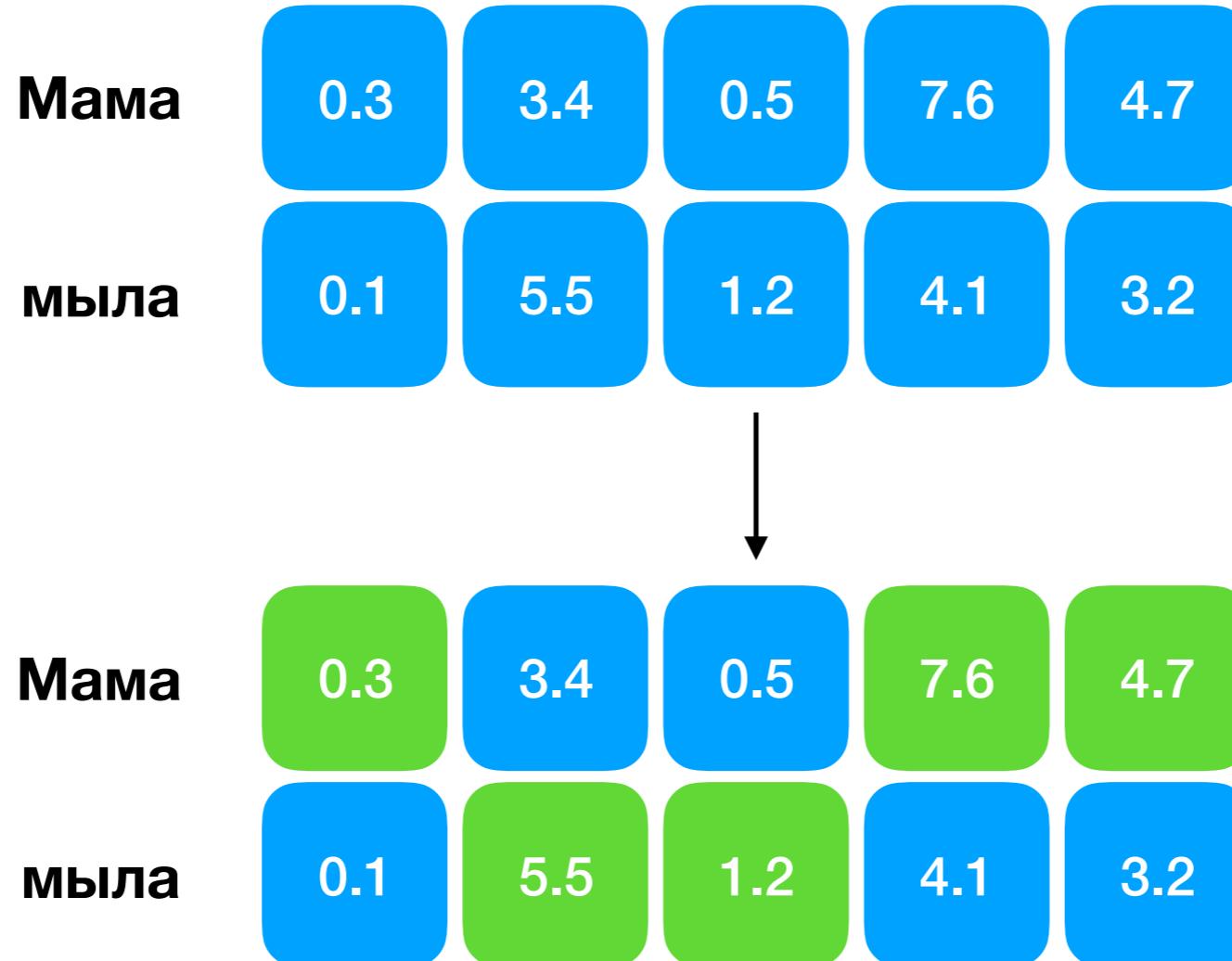
0.3 3.4 0.5 7.6 4.7

**мыла**

0.1 5.5 1.2 4.1 3.2

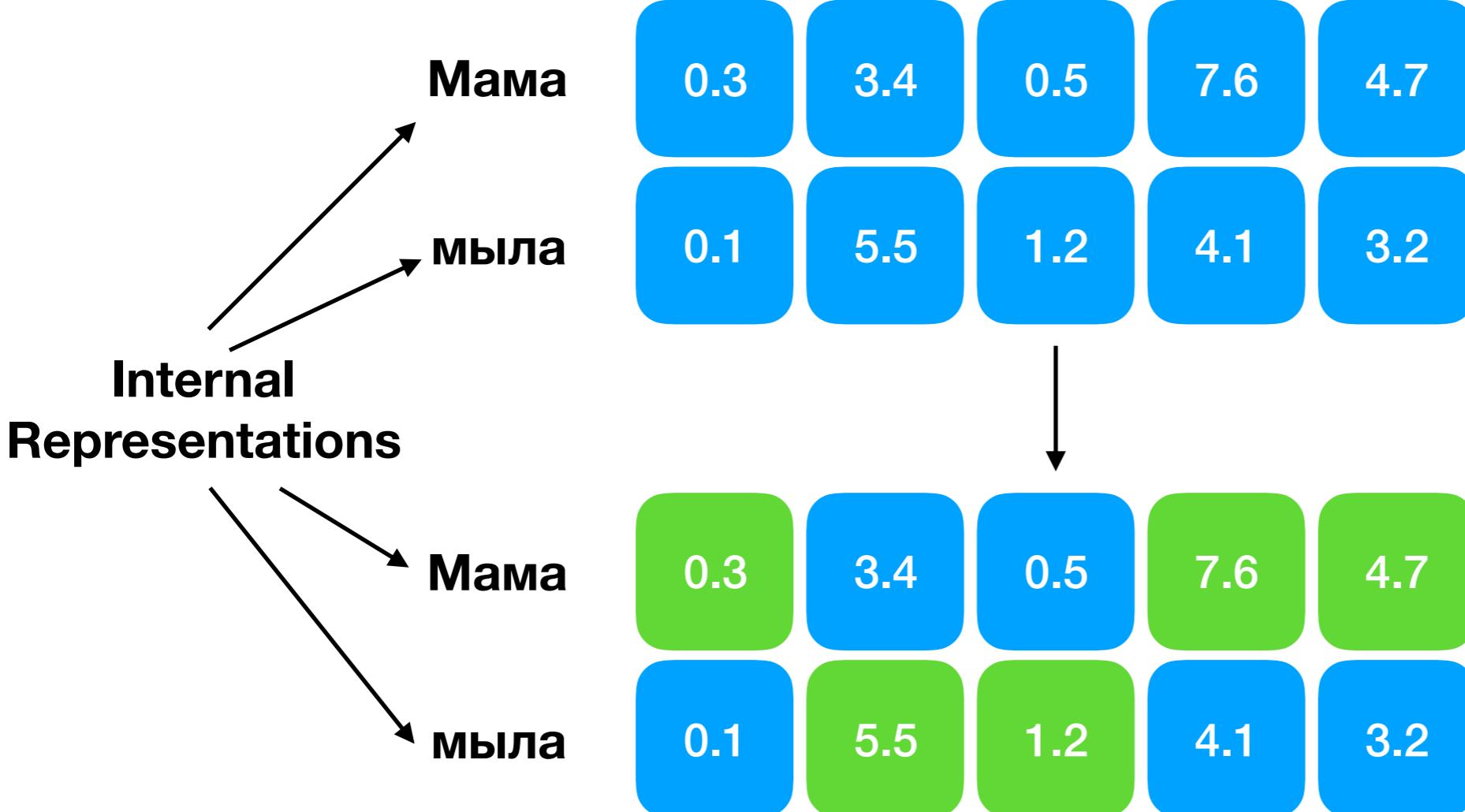
# Convolutional Neural Network

Max over time pooling



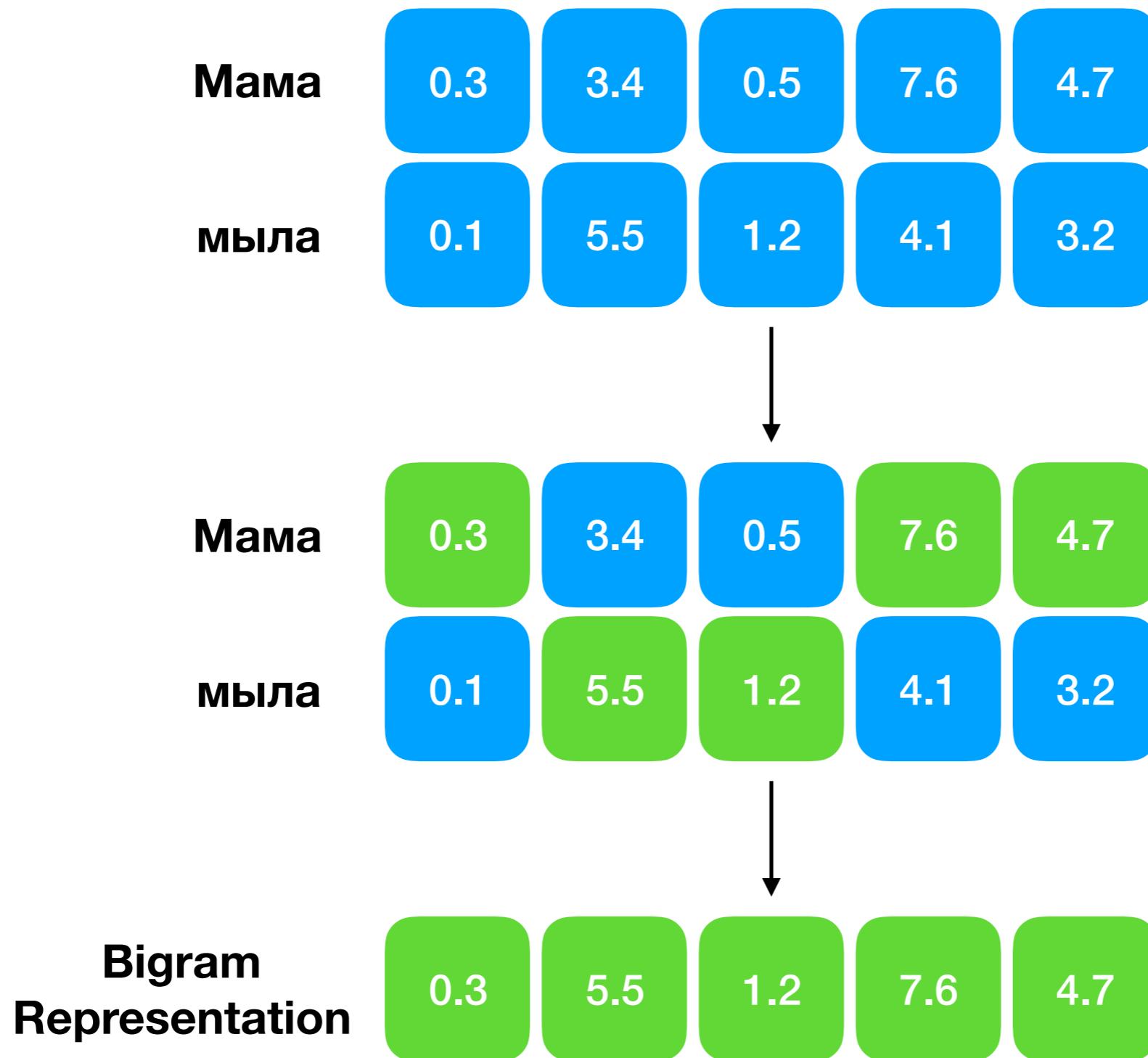
# Convolutional Neural Network

Max over time pooling



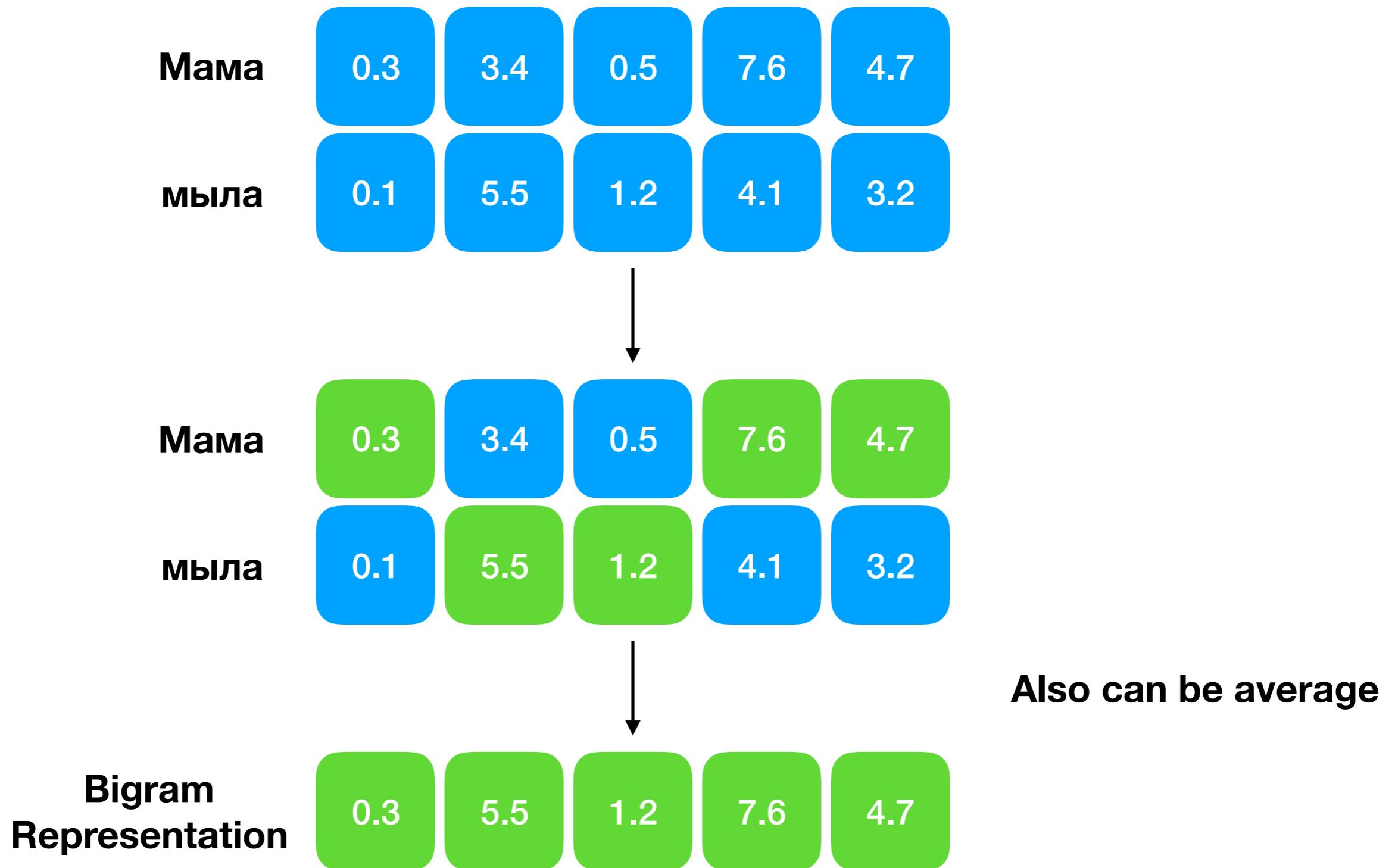
# Convolutional Neural Network

Max over time pooling

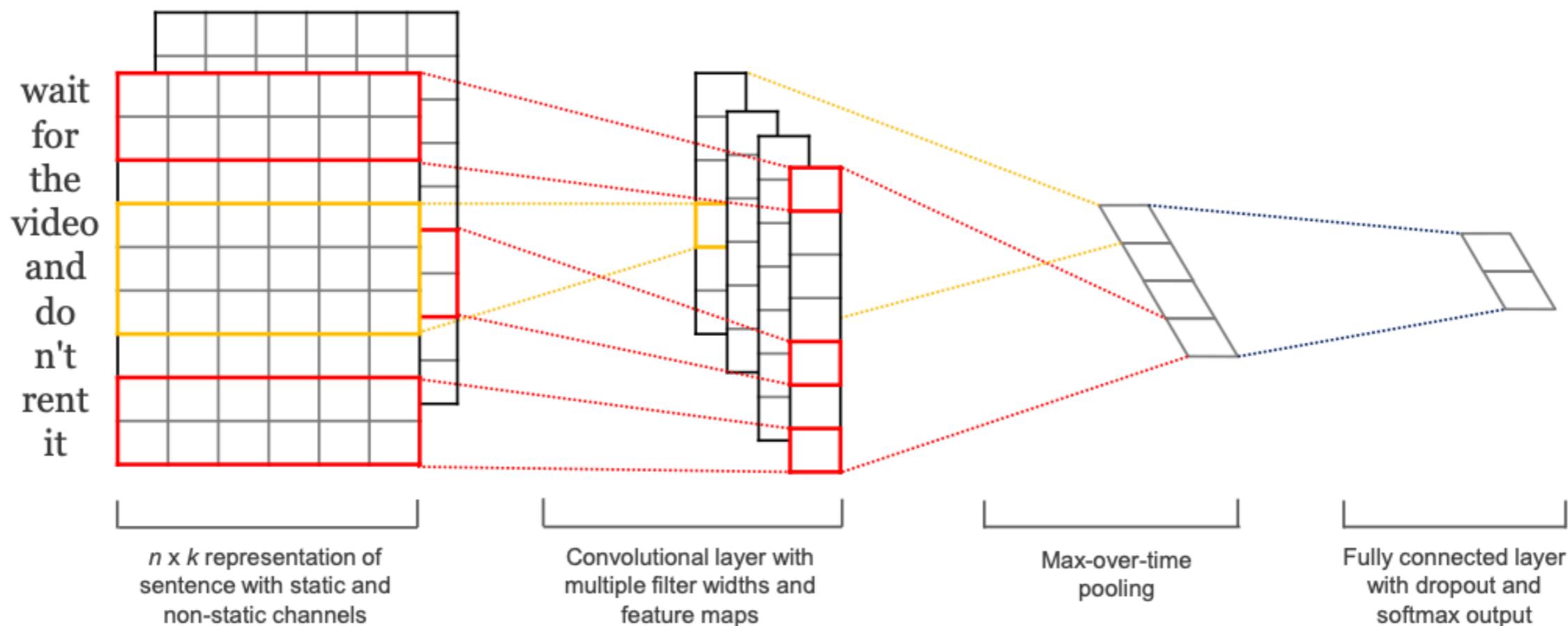


# Convolutional Neural Network

Max over time pooling

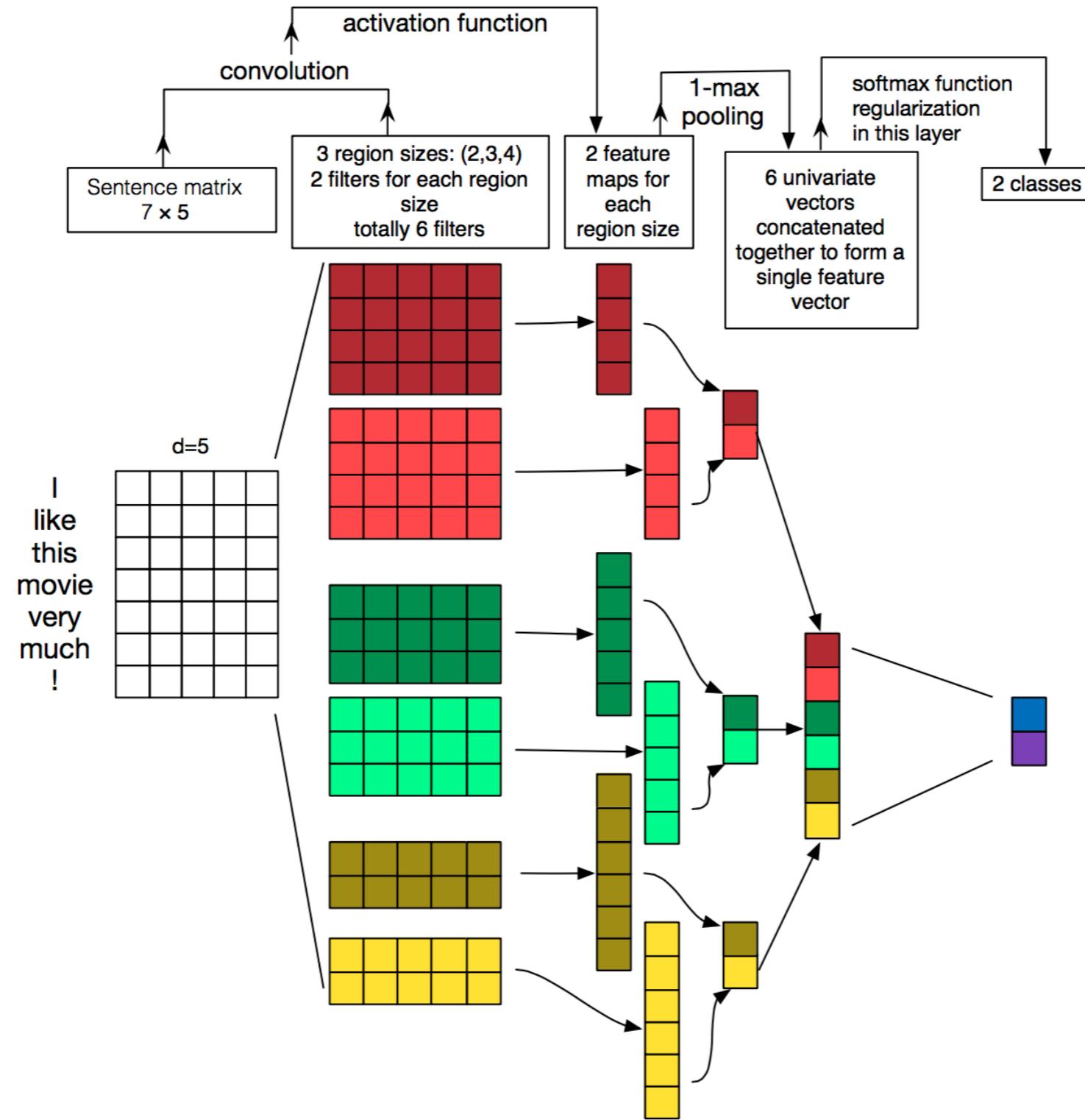


# Convolutional Neural Network

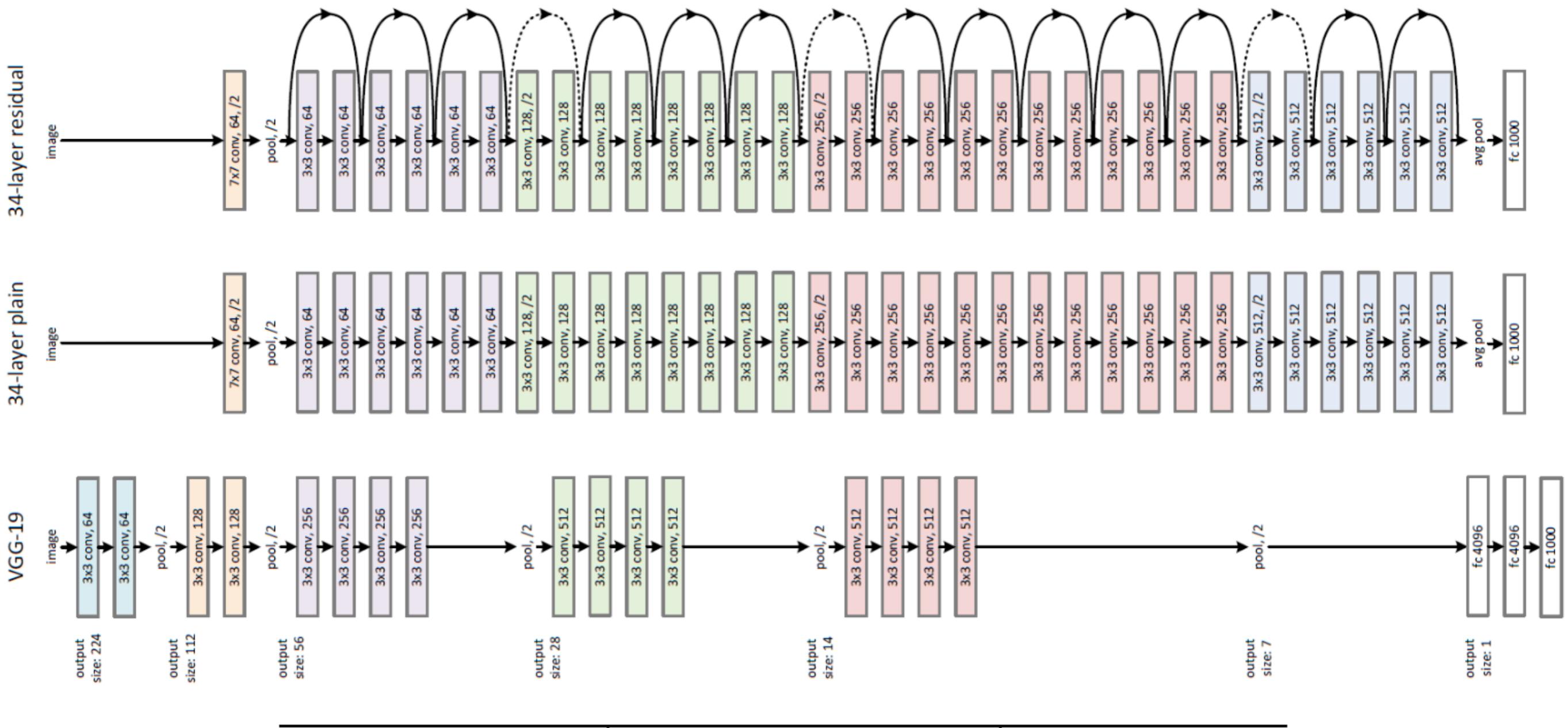


# **Kernel size = 1**

# Inception CNN



# Residual



## plain

## ResNet

## 18 layers

27.94

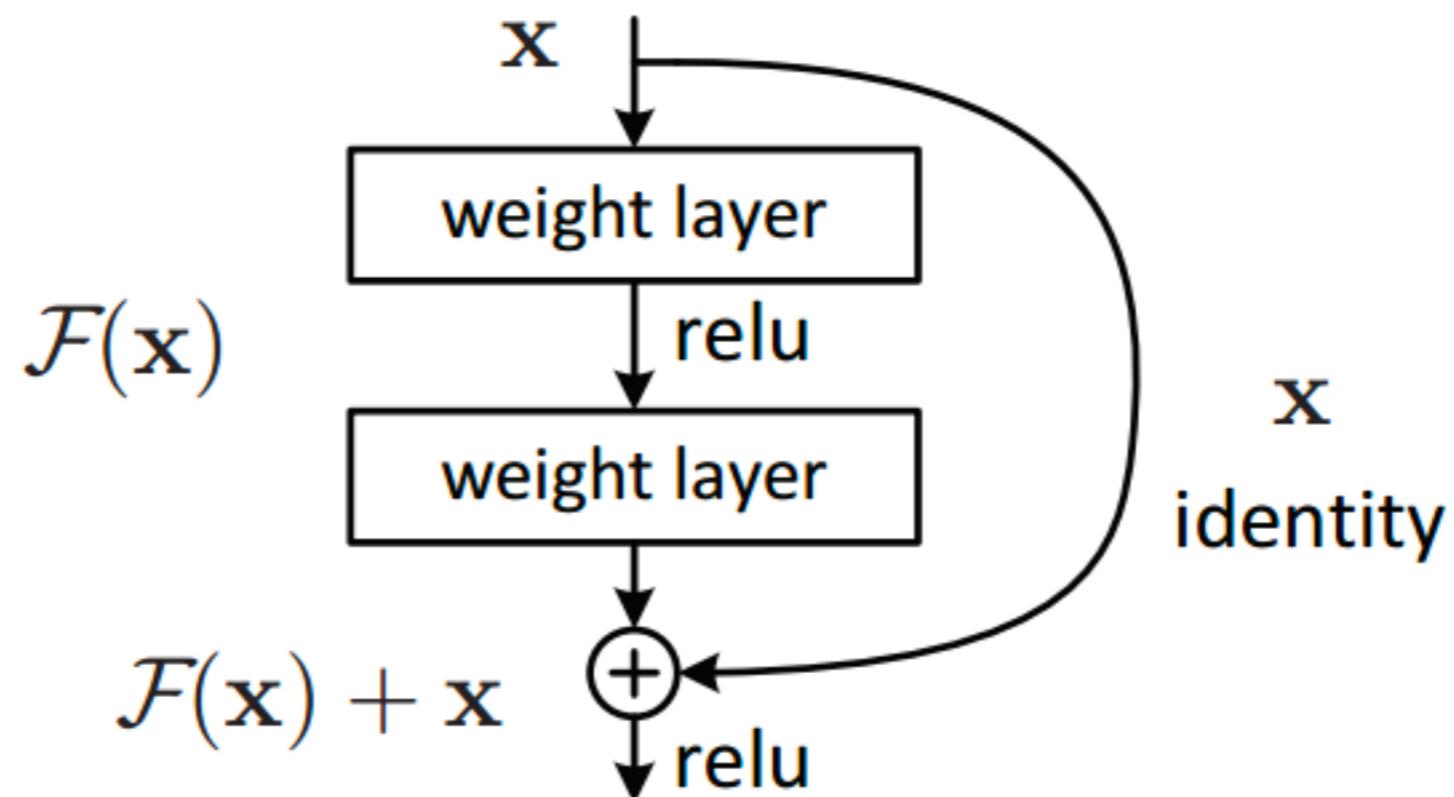
27.88

34 layers

28.54

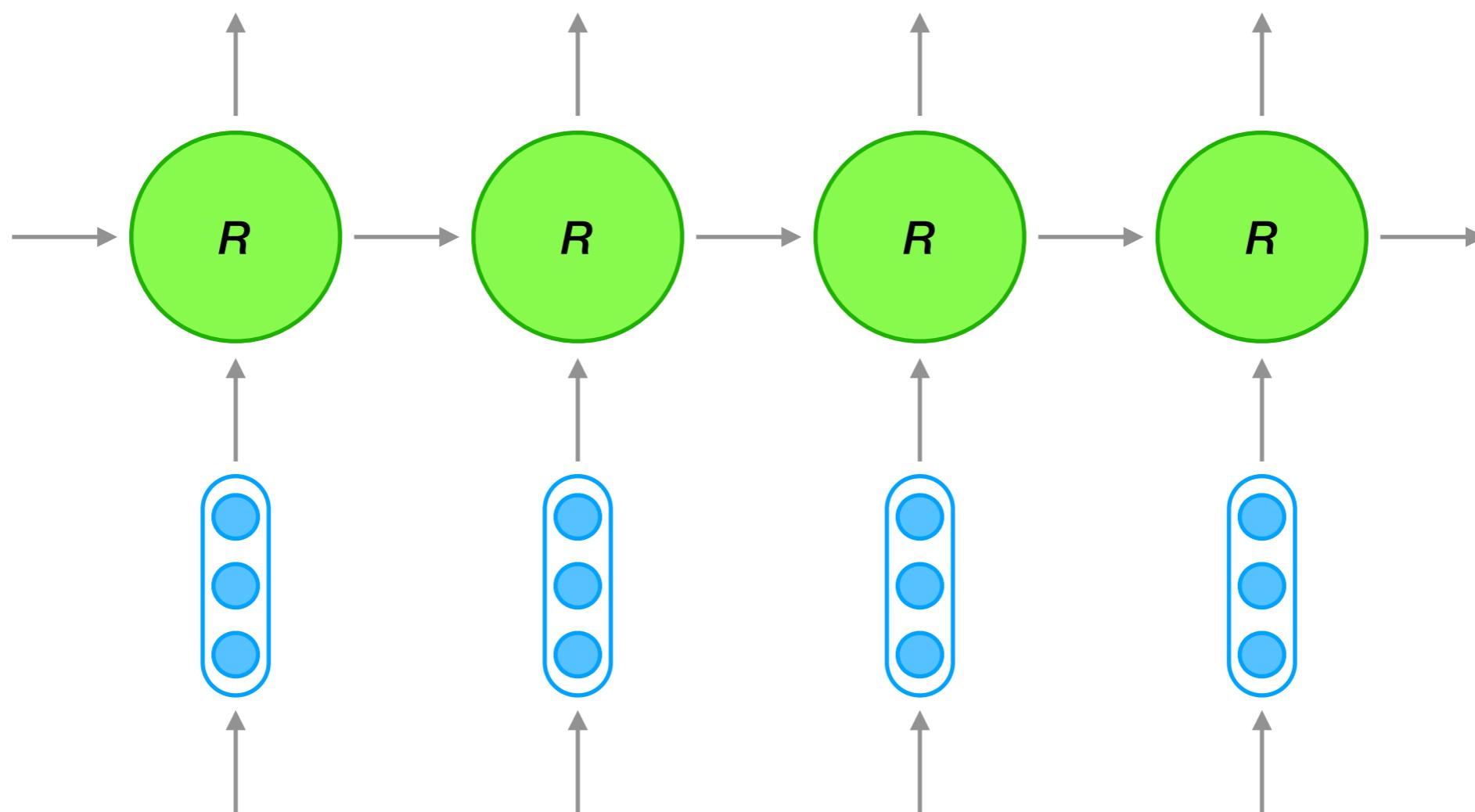
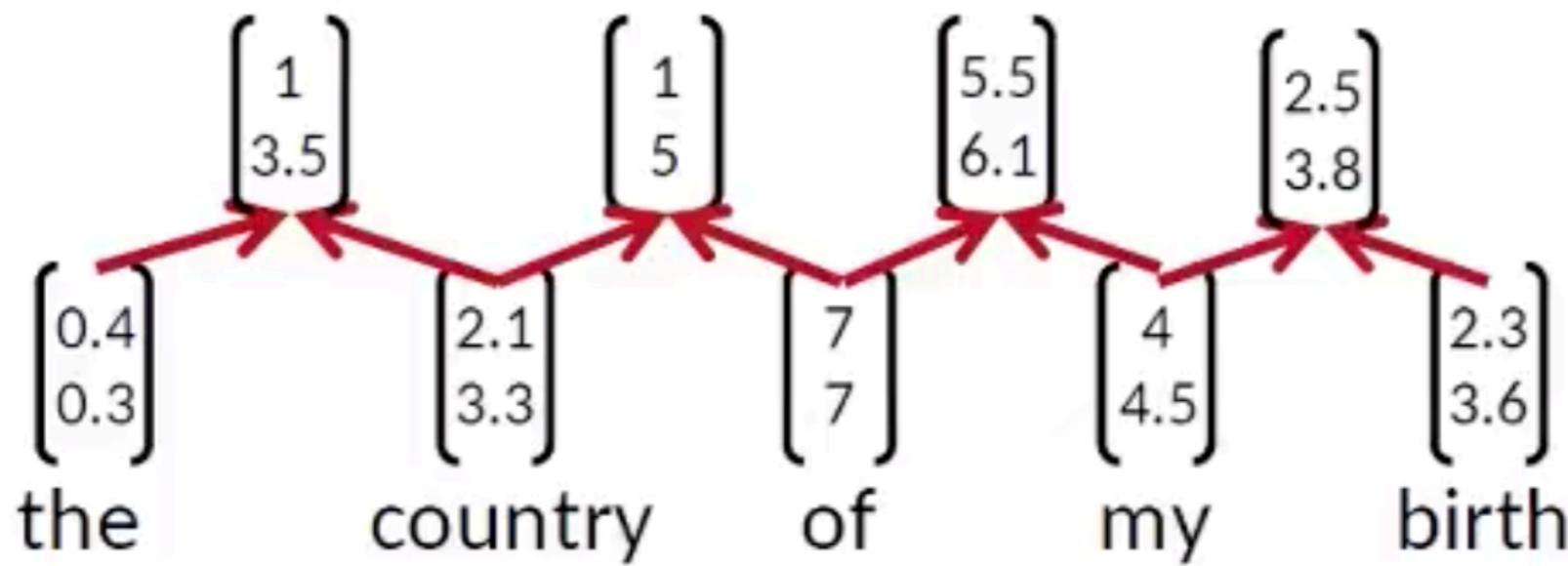
25.03

# Residual

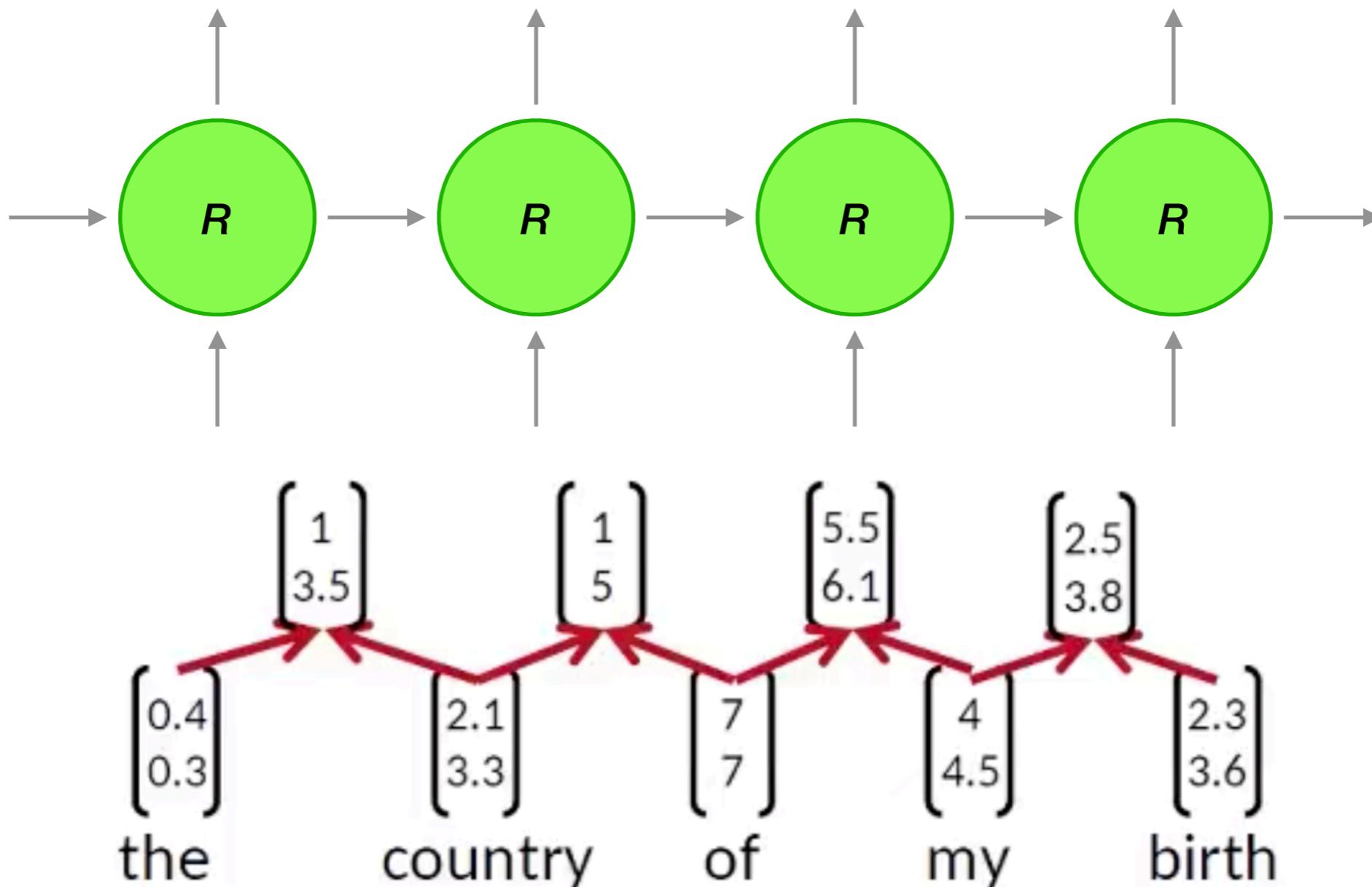


# Combination

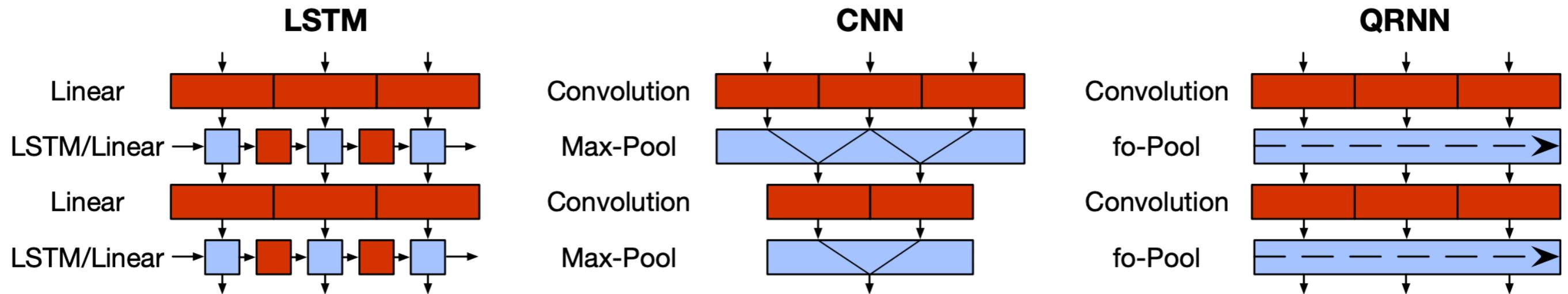
# Combination



# Combination



# Quasi-RNN



$$\mathbf{z}_t = \tanh(\mathbf{W}_z^1 \mathbf{x}_{t-1} + \mathbf{W}_z^2 \mathbf{x}_t)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f^1 \mathbf{x}_{t-1} + \mathbf{W}_f^2 \mathbf{x}_t)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o^1 \mathbf{x}_{t-1} + \mathbf{W}_o^2 \mathbf{x}_t).$$

$$\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{z}_t,$$

# Thanks for your Attention!

Boris Zubarev



@bobazooba