Natural Language Processing (CS-472) Spring-2023

Muhammad Naseer Bajwa

Assistant Professor,
Department of Computing, SEECS
Co-Principal Investigator,
Deep Learning Lab, NCAI
NUST, Islamabad
naseer.bajwa@seecs.edu.pk

NUST
Defining futures
School of Electrical Engineering & Computer Science

Overview of this week’s lecture

Convolutional Neural Networks

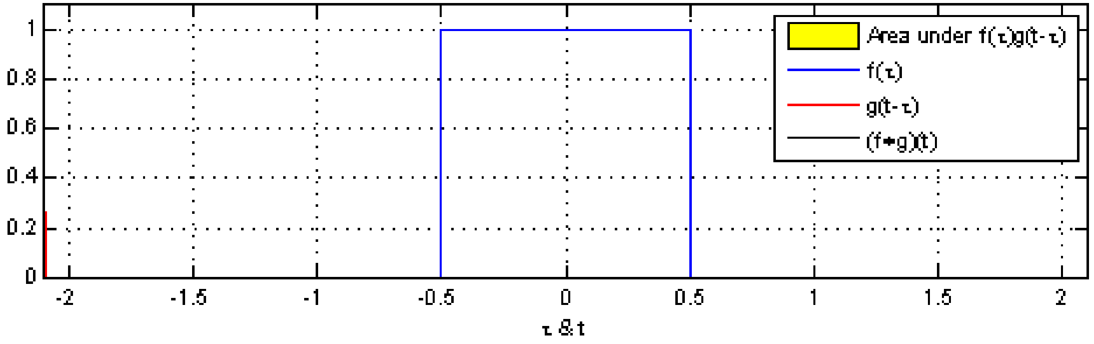
- Fundamentals of CNNs

- Applications of CNNs in NLP

What is convolution?

- Mathematically, convolution is an operation on two functions that shows how one function is modified by the other.

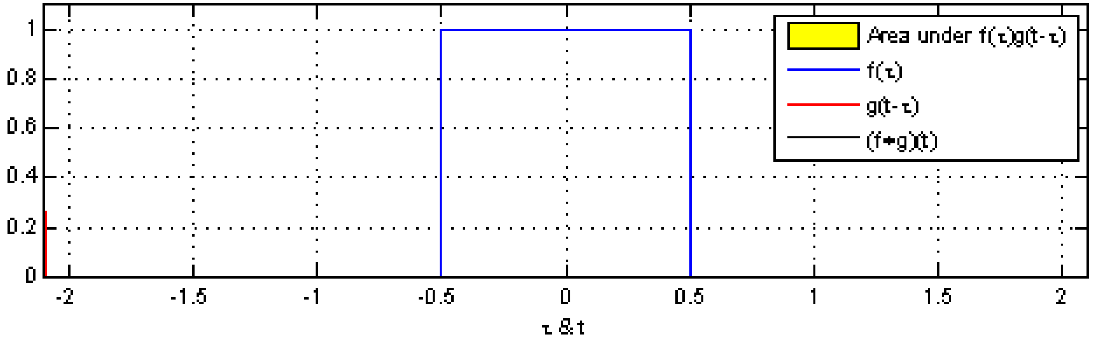
$$f[n] * g[n] = \sum_{m=-M}^{M} f[n]\, g[n-m]$$



NUST
Defining futures
School of Electrical Engineering & Computer Science

What is convolution?

- Mathematically, convolution is an operation on two functions that shows how one function is modified by the other.



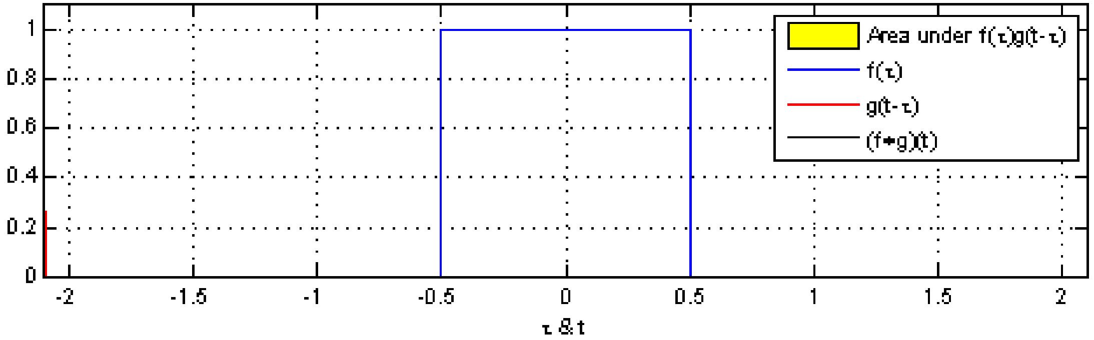
$$f[n] * g[n] = \sum_{m=-M}^{M} f[n]\, g[n-m]$$

- In computer vision, convolution finds features that are translation invariant.

What is convolution?

- Mathematically, convolution is an operation on two functions that shows how one function is modified by the other.

$$f[n] * g[n] = \sum_{m=-M}^{M} f[n]\, g[n-m]$$



- In computer vision, convolution finds features that are translation invariant.

- In AI, the operation normally referred to as convolution is actually cross-correlation.

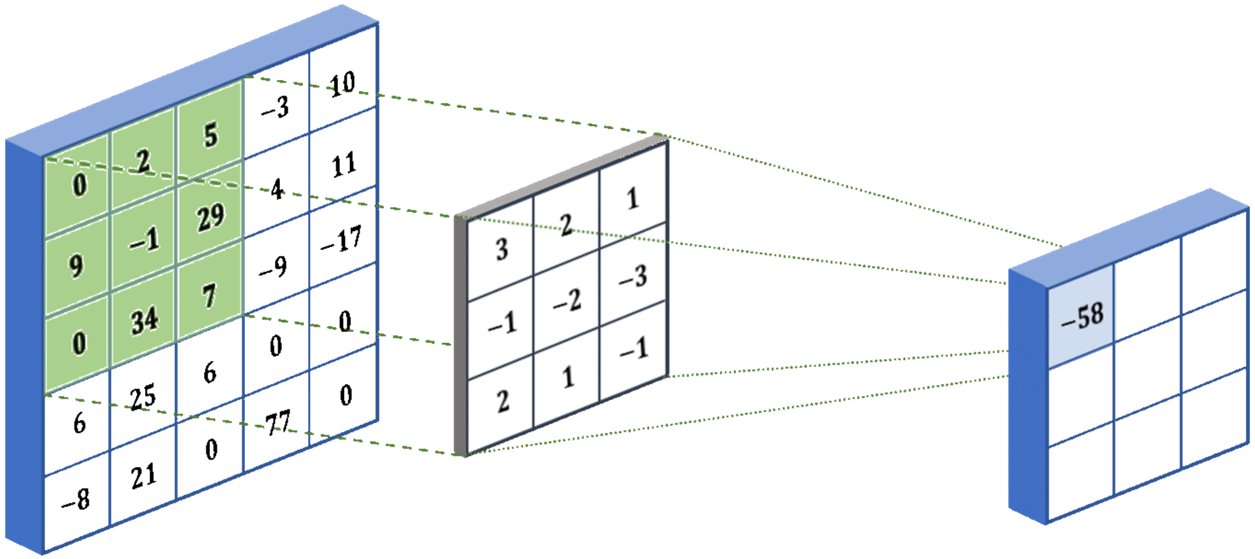
$$f[n] \star g[n] = \sum_{m=-M}^{M} f[n]\, g[n+m]$$

  - Correlation is a measure of similarity.

NUST
Defining futures
School of Electrical Engineering
& Computer Science

2D Convolution is inspired from visual cortex

- Take dot product between part of an image (receptive field) and a kernel or filter.

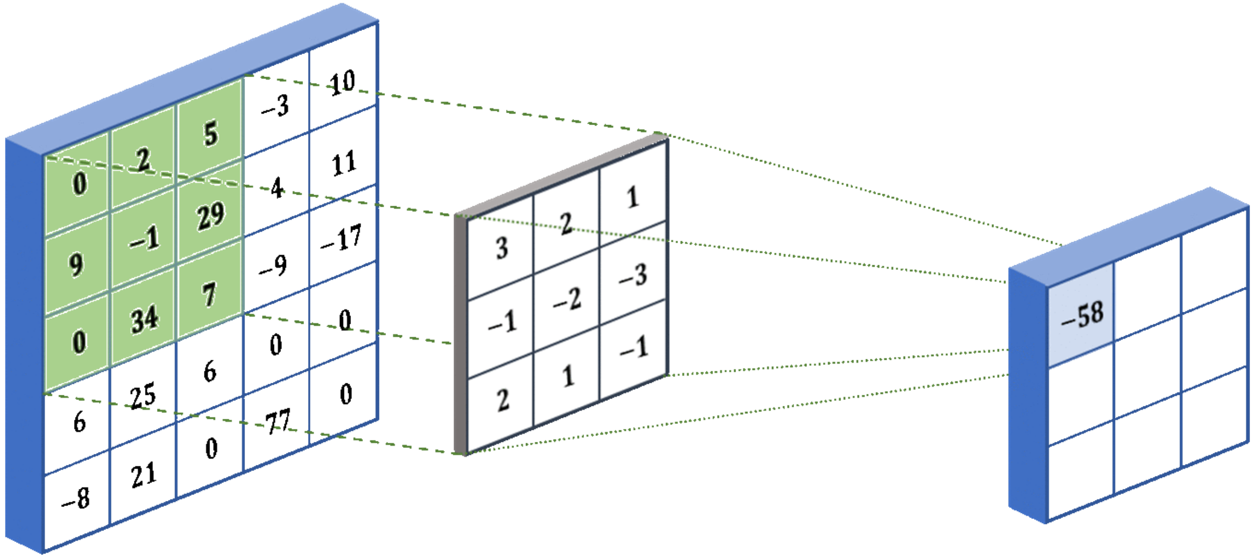


NUST
Defining futures
School of Electrical Engineering
& Computer Science

2D Convolution is inspired from visual cortex

- Take dot product between part of an image (receptive field) and a kernel or filter.

- The output of this operation (activation map) is, most of the times, a smaller image which finds a particular feature in the input.



NUST
Defining futures
School of Electrical Engineering
& Computer Science

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.

Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.

- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.

Let’s	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

NUST
Defining futures
School of Electrical Engineering
& Computer Science

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.
- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.
- Let we have a filter of size 3 (arbitrary number). The filter must have the same number of channels as the input (word vectors).

Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

NUST
Defining futures
School of Electrical Engineering
& Computer Science

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.
- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.
- Let we have a filter of size 3 (arbitrary number). The filter must have the same number of channels as the input (word vectors).

Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Let's put a	-1.0

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.
- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.
- Let we have a filter of size 3 (arbitrary number). The filter must have the same number of channels as the input (word vectors).

Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Let's put a	-1.0
Put a smile	-0.5

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.

- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.

- Let we have a filter of size 3 (arbitrary number). The filter must have the same number of channels as the input (word vectors).

Let’s	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Let’s put a	-1.0
Put a smile	-0.5
A smile on	-3.6

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.
- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.
- Let we have a filter of size 3 (arbitrary number). The filter must have the same number of channels as the input (word vectors).

Let’s	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Let’s put a	-1.0
Put a smile	-0.5
A smile on	-3.6
Smile on that	-0.2

1D convolution is used for text processing

- Let we have a sequence of words, each represented by a 4-dimentional dense vector.
- In CNN, 4-dimensions of these word vectors may be referred to as 4-channels.
- Let we have a filter of size 3 (arbitrary number). The filter must have the same number of channels as the input (word vectors).

Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Let's put a	-1.0
Put a smile	-0.5
A smile on	-3.6
Smile on that	-0.2
On that face	0.3

How to preserve the dimension of output?

Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Let's put a	-1.0
Put a smile	-0.5
A smile on	-3.6
Smile on that	-0.2
On that face	0.3

How to preserve the dimension of output?

- Use zero-padding on one or both ends of the signal.

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Ø Let's put	-0.6
Let's put a	-1.0
Put a smile	-0.5
A smile on	-3.6
Smile on that	-0.2
On that face	0.3
That face Ø	-0.5

How to preserve the dimension of output?

- Use zero-padding on one or both ends of the signal.

- Apply multiple filters.

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

How to preserve the dimension of output?

- Use zero-padding on one or both ends of the signal.
- Apply multiple filters.
  - Each filter learns a specific aspect of the text.

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

Pooling over time helps summarise the text using features

- Max-pooling checks whether a feature learnt by a filter is present anywhere in the text.

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Max-pool	**0.3**	**1.6**
Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

Pooling over time helps summarise the text using features

- Max-pooling checks whether a feature learnt by a filter is present anywhere in the text.

- Average-pooling estimates the average strength of a feature in a piece of text.

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Avg-pool	**-0.87**	**0.26**
Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

Pooling over time helps summarise the text using features

- Max-pooling checks whether a feature learnt by a filter is present anywhere in the text.
- Average-pooling estimates the average strength of a feature in a piece of text.
  - Average pooling is not widely used.

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Avg-pool	-0.87	0.26
Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Put a smile	-0.5	-0.1
Smile on that	-0.2	-0.1
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Put a smile	-0.5	-0.1
Smile on that	-0.2	-0.1
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Put a smile	-0.5	-0.1
Smile on that	-0.2	-0.1
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Put a smile	-0.5	-0.1
Smile on that	-0.2	-0.1
That face Ø	-0.5	-0.9

NUST
Defining futures
School of Electrical Engineering
& Computer Science

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

How to compress the feature map?

Ø Let's put a	-0.6	1.6

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling

Ø Let's put a	-0.6	1.6
Let's Put a smile	-0.5	1.6

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put a	-0.6	1.6
Let's Put a smile	-0.5	1.6
Put a smile on	-0.5	0.3
A smile on that	-0.2	0.3
Smile on that face	0.3	0.6
On that face Ø	0.3	0.6

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

NUST
Defining futures
School of Electrical Engineering
& Computer Science

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling
  - k-max pooling

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

2-Max-pool	-0.2	1.6
	0.3	0.6

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

NUST
Defining futures
School of Electrical Engineering
& Computer Science

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling
  - k-max pooling
  - Dilated Convolution

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

How to compress the feature map?

- Some of the following may be used to compress the feature map.
  - Kernel stride of greater than 1
  - Local pooling
  - k-max pooling
  - Dilated Convolution

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

NUST
Defining futures
School of Electrical Engineering
& Computer Science

How to compress the feature map?

- Some of the following may be used to compress the feature map.

  - Kernel stride of greater than 1
  - Local pooling
  - k-max pooling
  - Dilated Convolution

Ø	0.0	0.0	0.0	0.0
Let's	0.2	0.1	-0.3	0.4
Put	0.5	0.2	-0.3	-0.1
A	-0.1	-0.3	-0.2	0.4
Smile	0.3	-0.3	0.1	0.1
On	0.2	-0.3	0.4	0.2
That	0.1	0.2	-0.1	-0.1
Face	-0.4	0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Politeness

-3	1	2	-3
-1	2	1	-3
1	1	-1	1

Question

1	0	1	1
1	0	-1	-1
0	1	0	1

Ø Let's put	-0.6	0.2
Let's put a	-1.0	1.6
Put a smile	-0.5	-0.1
A smile on	-3.6	0.3
Smile on that	-0.2	-0.1
On that face	0.3	0.6
That face Ø	-0.5	-0.9

NUST Defining futures
School of Electrical Engineering & Computer Science

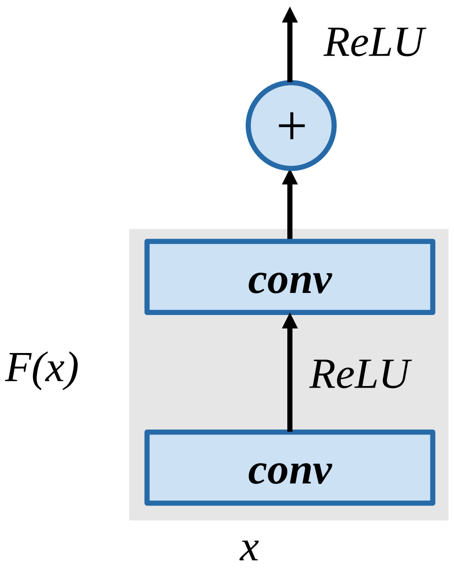
Training deeper CNNs requires some tricks to avoid vanishing gradients

- The idea of gating/skipping an input as seen in LSTMs can also be used in CNNs.

Training deeper CNNs requires some tricks to avoid vanishing gradients

- The idea of gating/skipping an input as seen in LSTMs can also be used in CNNs.

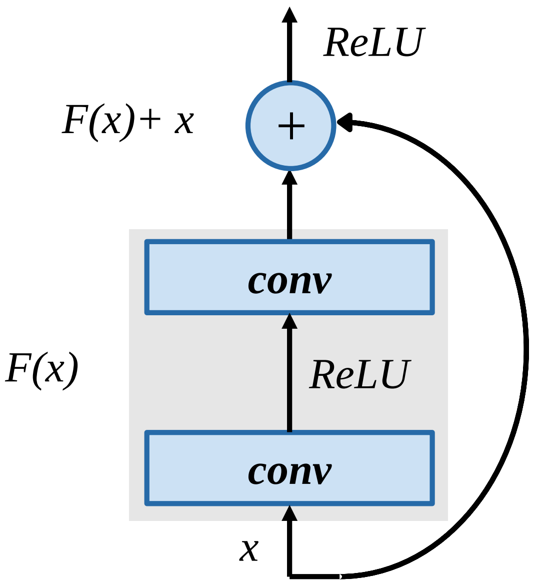
- Summing candidate updates with shortcut connections is required for deeper networks to train.



Residual Block

Training deeper CNNs requires some tricks to avoid vanishing gradients

- The idea of gating/skipping an input as seen in LSTMs can also be used in CNNs.
- Summing candidate updates with shortcut connections is required for deeper networks to train.

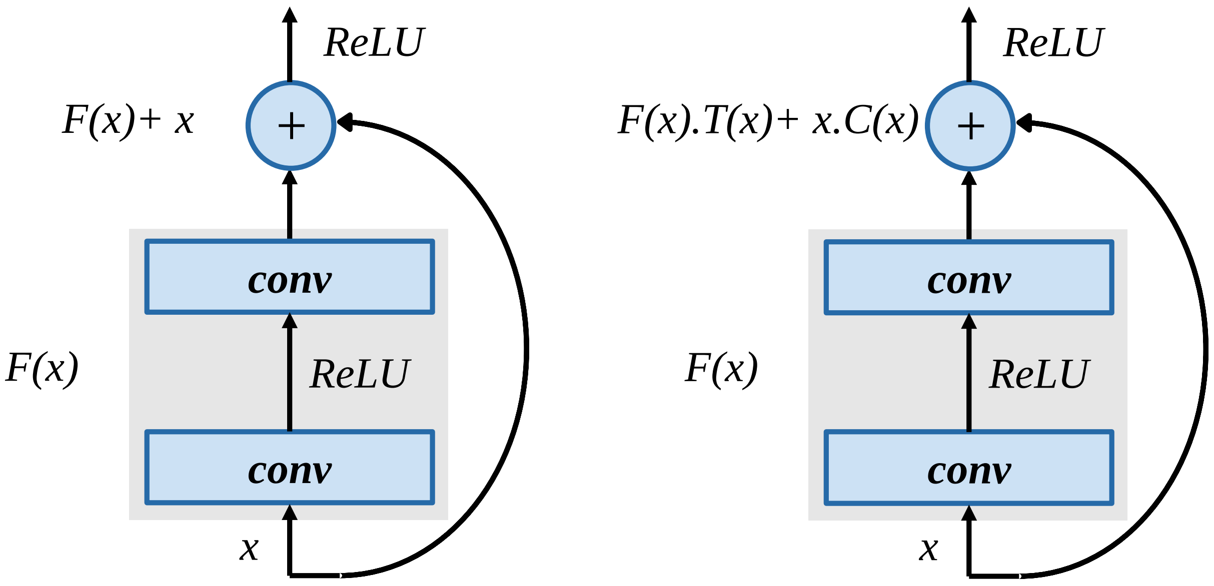


Residual Block

Training deeper CNNs requires some tricks to avoid vanishing gradients

- The idea of gating/skipping an input as seen in LSTMs can also be used in CNNs.

- Summing candidate updates with shortcut connections is required for deeper networks to train.

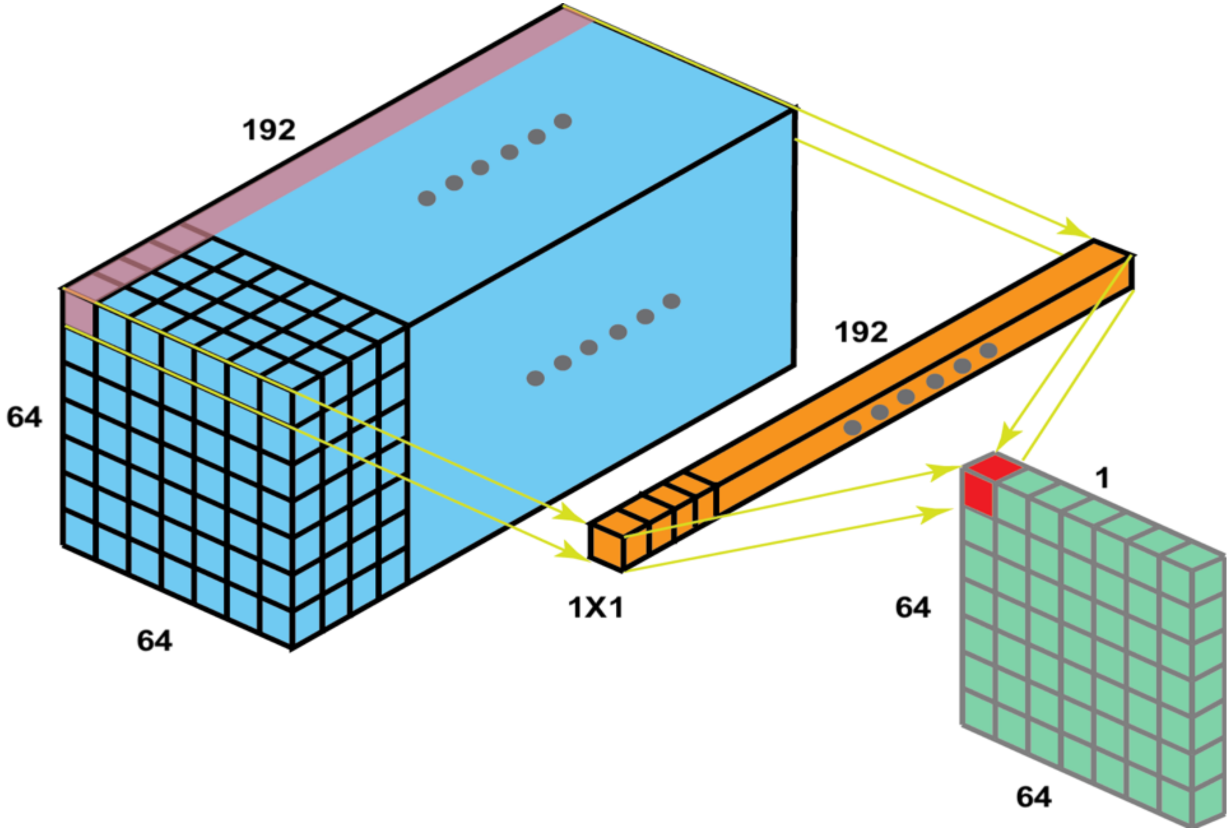


Residual Block **Highway Block**

Does 1 × 1 convolution actually make sense?

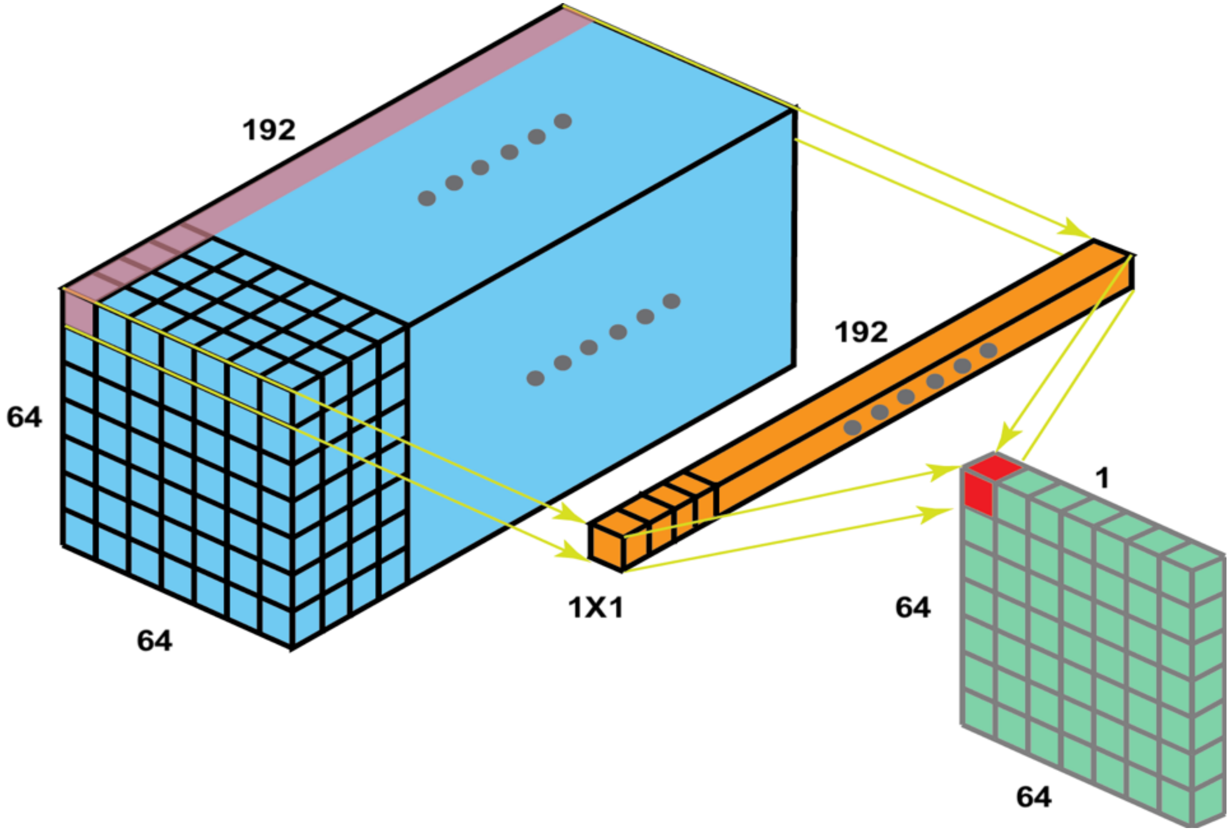
Does 1×1 convolution actually make sense?

- Kernels have size 1 but have depth equal to the input channels.



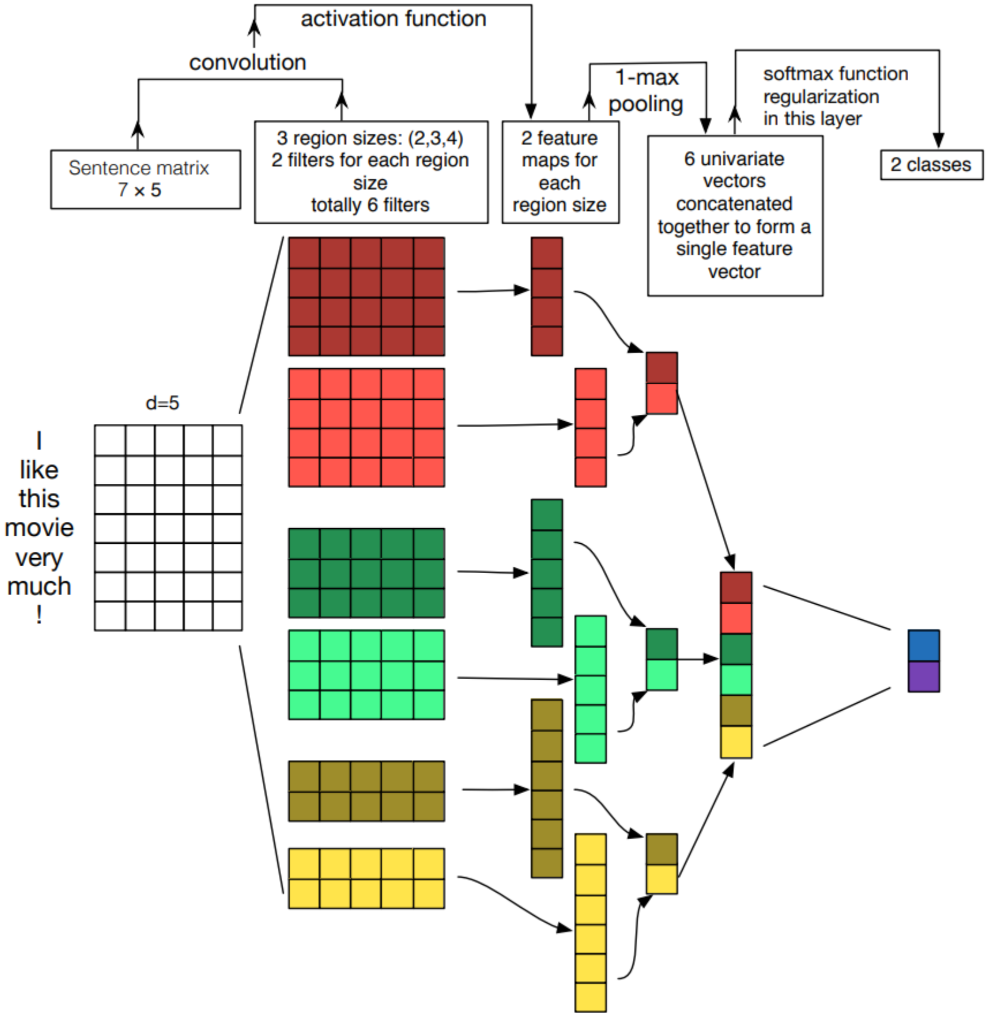
Does 1×1 convolution actually make sense?

- Kernels have size 1 but have depth equal to the input channels.
    - Also known as a Network-in-Network connections.
    - Acts as a fully connected layer across channels.
    - Adds additional NN layer with only a few parameters.
    - May be used to compress channels.



NUST
Defining futures
School of Electrical Engineering
& Computer Science

A simple application of CNN for text classification



Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

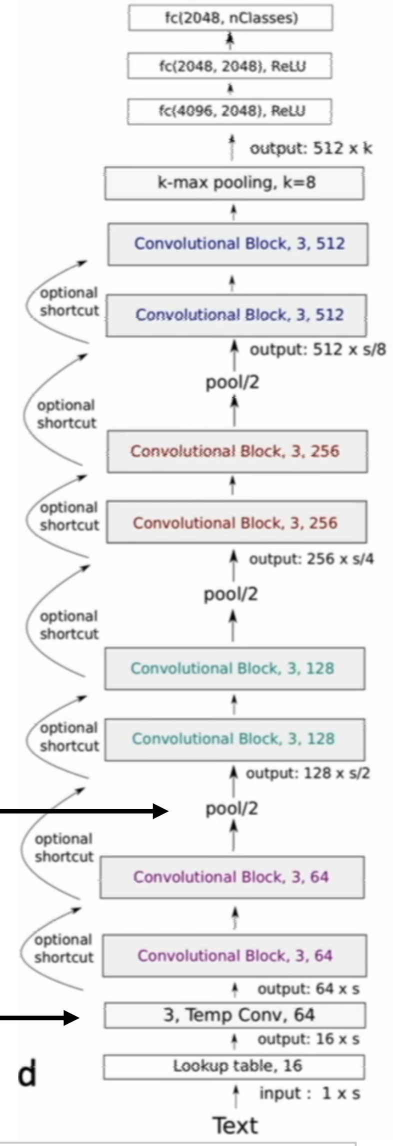
NUST Defining futures
School of Electrical Engineering & Computer Science

A very deep CNN architecture

- A CNN based DNN for text classification built over characters.
  - Input is taken as a fixed sequence of 1024 characters.
  - Each character is represented by an embedding of size 16.
  - k-max pooling is used after convolutional backbone ($k = 8$).

Local pooling at each stage halves temporal resolution and doubles number of features

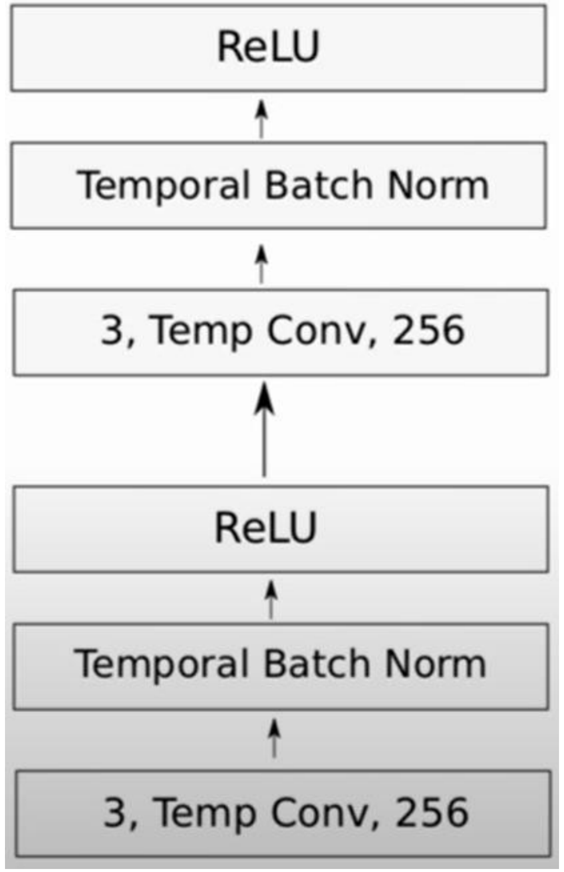
64 kernels of size 3



Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781.

A very deep CNN architecture

- Each convolutional block consists of
  - Two $1d$ convolutional layers
  - Each $conv$ layer is followed by a batch norm and a $ReLU$.
  - Kernel size is 3.
  - Number of kernels differ in different blocks.
  - Padding is used to set the dimensions as per requirement.



Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781.

Do you have any problem?

Some material (images, tables, text etc.) in this presentation has been borrowed from different books, lecture notes, and the web. The original contents solely belong to their owners, and are used in this presentation only for clarifying various educational concepts. Any copyright infringement is ***not at all*** intended.