Convolution

Kernel is a grid of weights
Kernel is applied to the image
Used in image processing

Image

4	5	6	9	2	4
3	**4**	6	8	8	1
2	4	7	9	5	0
7	9	0	4	3	6
0	7	2	4	6	8
7	8	3	5	2	9

Kernel

1	0	-1
0	3	2
-2	1	0

Kernel Applied to image
sum(image_i* K_i = 4 * 1+ 5 * 0 + 6 * (-1) + 3 * 0 + 4 * 3 + 6 * 2 + 2 * (-2) + 1 * 1 + 7 * 0 = 19
Resulting Matrix = is output

	19				

Kernels

Feature Detectors

Kernels are learned

Oblique line detector

1	**0**	**0**
0	**1**	**0**
0	**0**	**1**

Vertical Line detector

0	**1**	**0**
0	**1**	**0**
0	**1**	**0**

CNN learns from many Kernels different features.CNN learns the weight of the Kernel

Architectural decisions for convolution

Grid size

Number Of pixels for height/width

Odd Numbers used

Stride

Step size used for sliding kernel on image

Indicated in pixels

Depth

Kernel = 3 x 3 x 3, Number Of weight = 27

Number Of Kernels

A conv layer has multiple kernels

Each kernel outputs a single 2D array

Output from a layer Number of 2d arrays = number of kernels

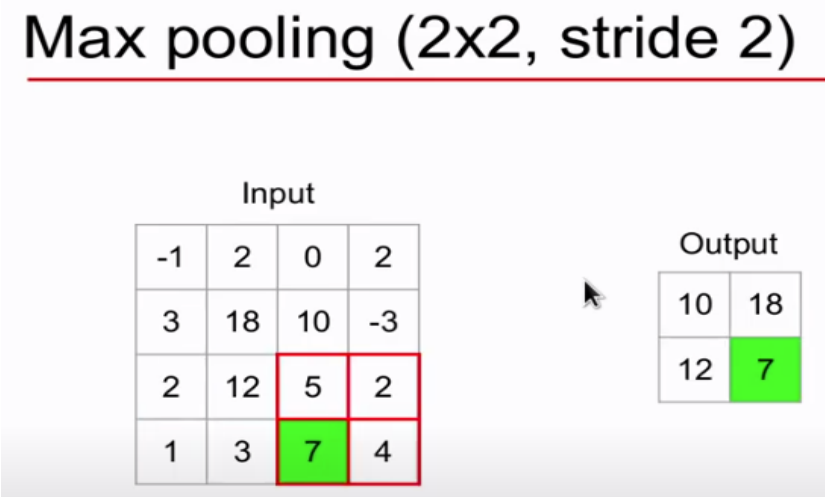
Pooling

- Downsample the image
- Overlaying grid on image
- Max/average pooling
- No parameters

Pooling Settings

- Grid Size
- Stride
- Type(max ,average)

Example

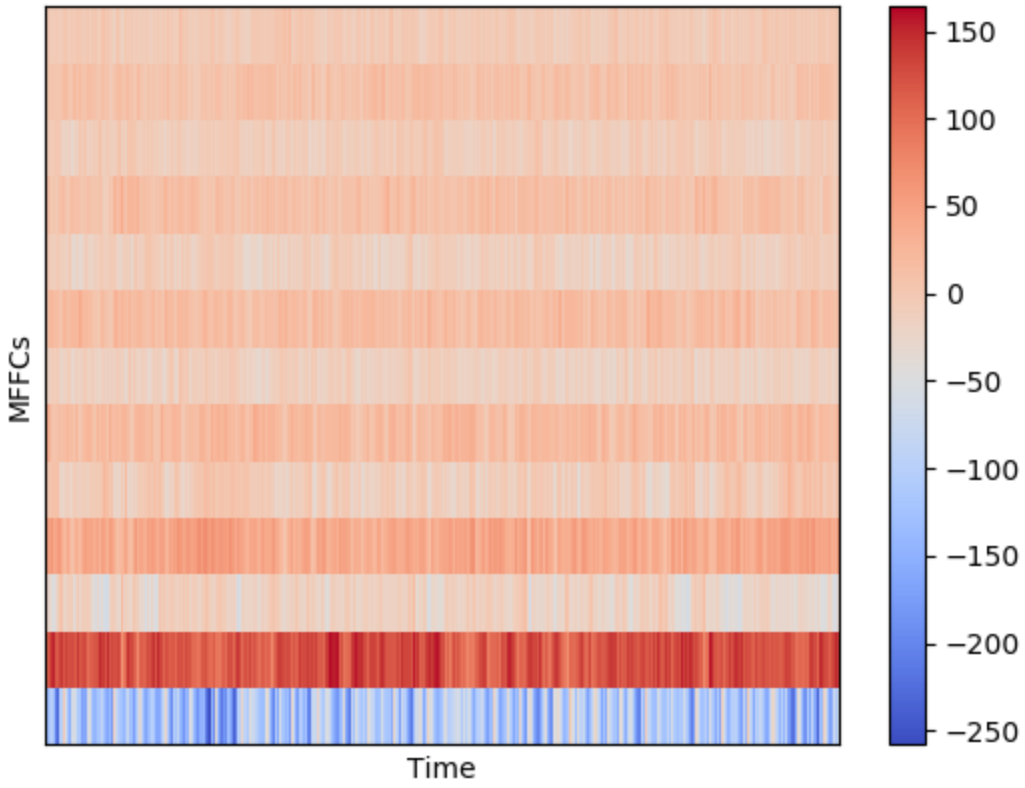


Input ->
Feature Learning =
[Convolution + relu -> pooling -> Convolution + relu -> pooling]->
Classification =[Flatten -> Fully Connected -> Softmax]

CNN apply to audio image

Spectrogram/MFCC = 2D image
Time, frequency = x, y
Amplitude = pixel value

Example 13 MFCC
Hop length = 512 sample
Number Of samples in audio file = 51200



Data shape = 100 x 13 x 1 , 1 means gray scale image => input to Tensorflow

Output from CNN

Acuracy on test set is: 0.7490909099578857
Expected index: 6, Predicted index: [6]
Expected Value: metal, Predicted Value: ['metal']