DISCUSSION Convolutions

2	0
0	2

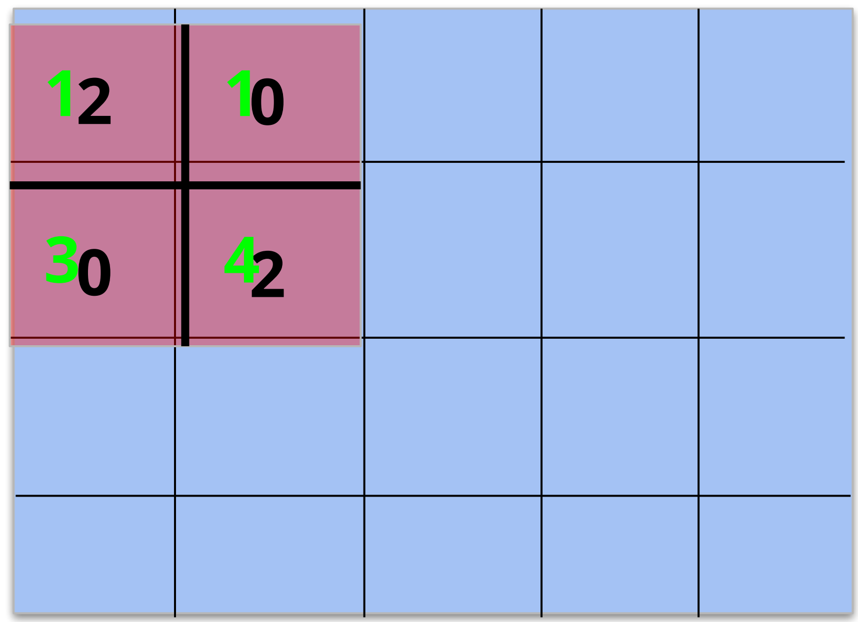
2x2 convolution matrix

1	1			
3	4			

100x100 image

DISCUSSION Convolutions

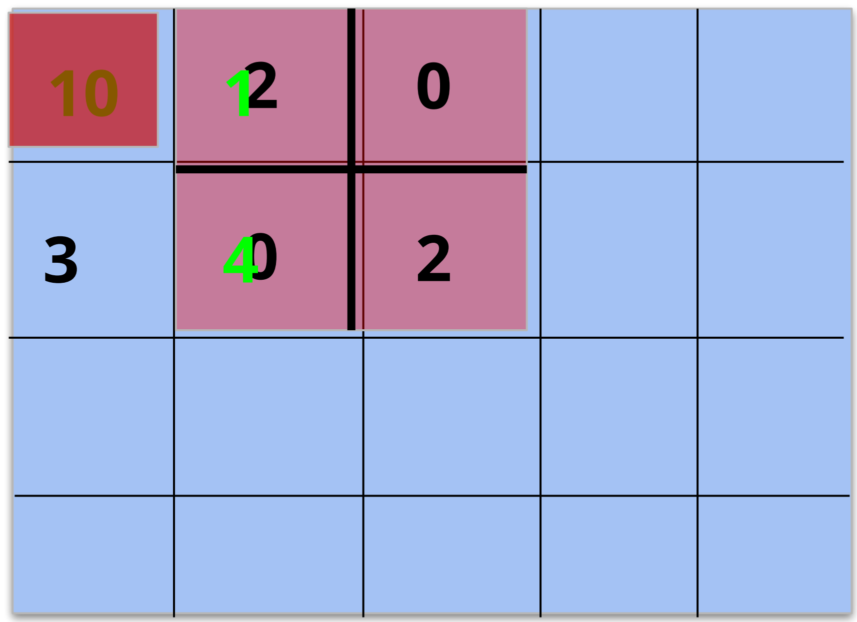
Result=1x2 + 1x0 + 3x0+ 4x2 =10



100x100 image

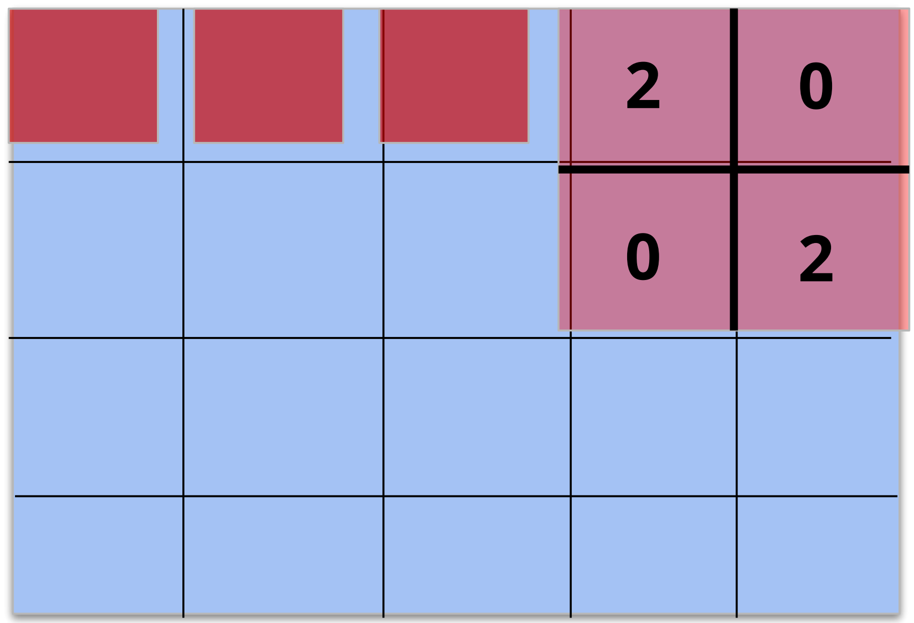
DISCUSSION Convolutions

Result=1x2 + 1x0 +
3x0+ 4x2
=10



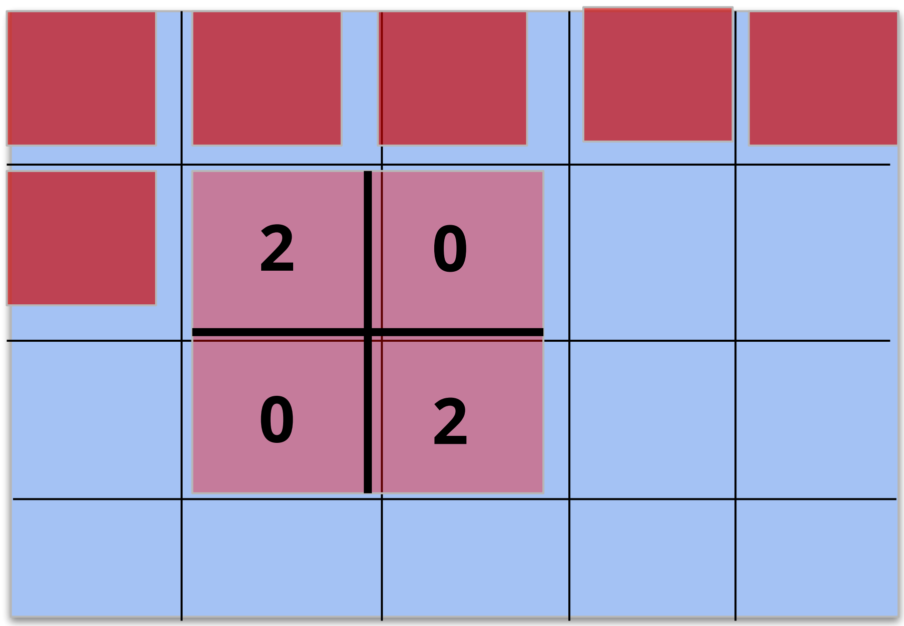
100x100
image

DISCUSSION Convolutions



100x100
image

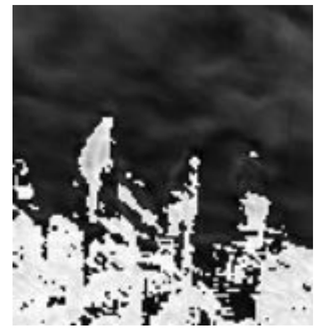
DISCUSSION Convolutions



100x100
image

DISCUSSION Convolutions

Convolution

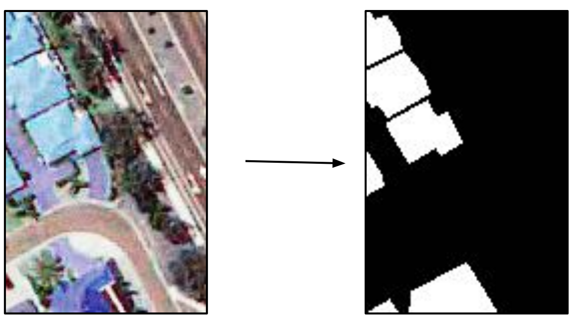


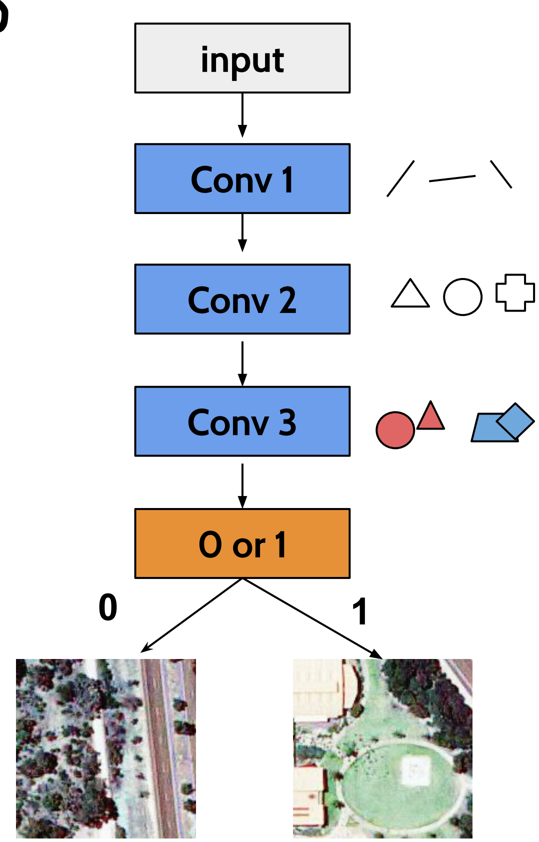
Scale and Rotational invariance

Try building a 3 layer CNN in Ex. 3.1

DISCUSSION Convolutions

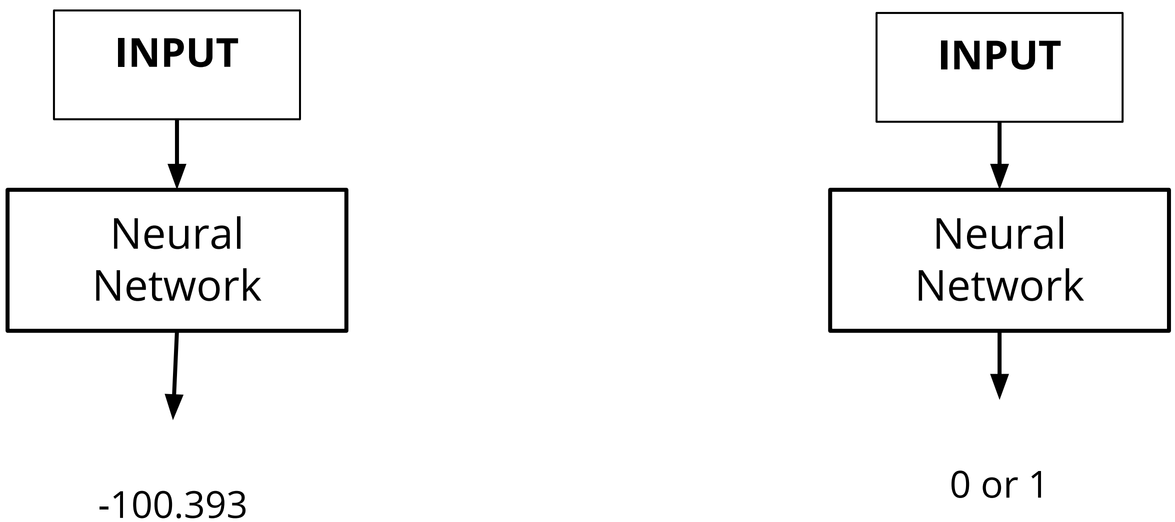
- Convolution matrix
- Role of max pooling
- Image based problem types:
  - Classification
  - Semantic Segmentation
  - Object detection



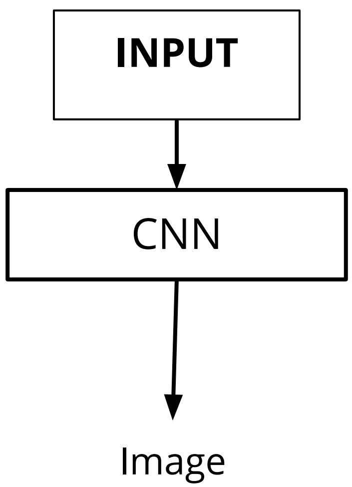


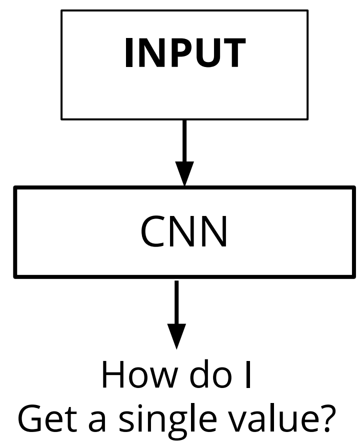
DISCUSSION Convolutions

Supervised Learning problem types



DISCUSSION Convolutions





Try Ex. 3.3

Teaser

How do I solve yes/no problems where the output is constrained to lie between **zero** and **1** ?

Can I even have such an output?

Let us, using pyplot, plot this function:

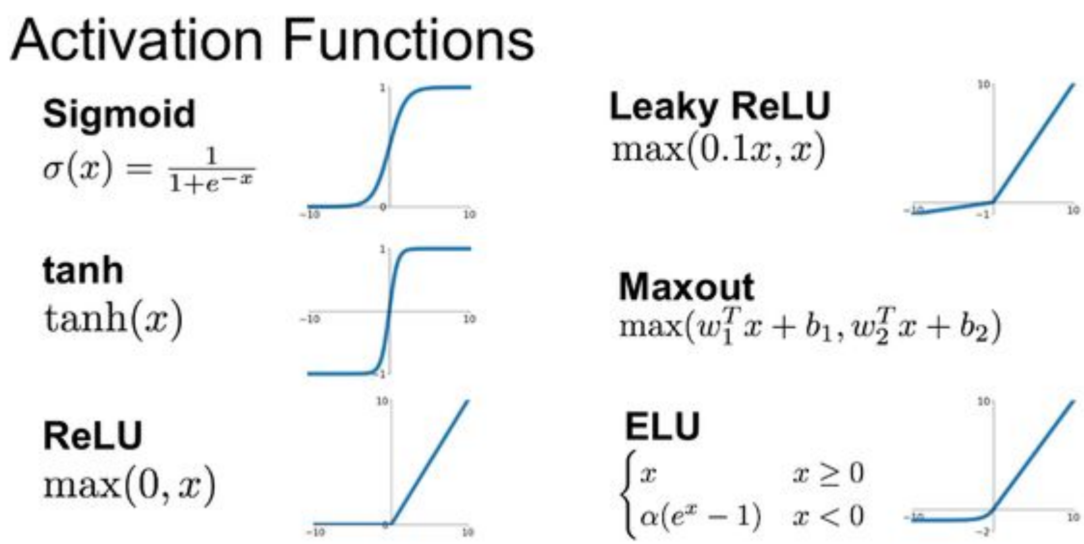
1/(1+exp(-x))

Can we, computationally, succeed getting a zero (not 0.0000*something*) or even a perfect 1(not 0.99999something) from the above expression

Why is having something close to one/zero (and not one/zero!) so important?
We'll revisit this question later...

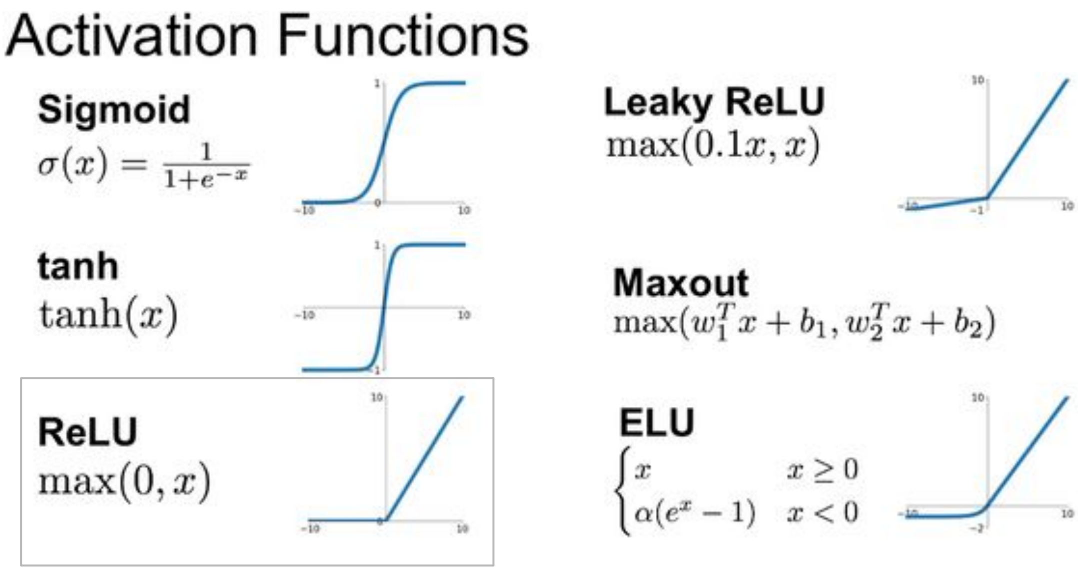
Curve Fitting - Activations

A word about activations. Why neural networks work even for highly non-linear predictions?



Curve Fitting - Activations

A word about activations. Why neural networks work for even highly non-linear predictions?



Curve Fitting - Logistic regression

How does the *traditional* loss - mse work for yes/no problems?

Ground truth	Predictor	MSE
1	0.02	0.96
1	0.002	0.996 ??
1	0.5	0.25
1	0.95	0.0025
1	0.4	0.36

Curve Fitting - Logistic regression

Which loss do you think is better?

Ground truth	Predictor	MSE	New loss
1	0.02	0.96	3.9
1	0.002	0.996	6.21
1	0.5	0.25	1.38
1	0.95	0.0025	0.05
1	0.4	0.36	0.91
1	1E-10	1	23

Curve Fitting - Logistic regression

Such a loss is termed as cross-entropy

Mathematically, loss =

-1*{prediction * log(ground_truth) + (1-prediction)*log(1-ground_truth)}

Don’t worry about getting a log(0) (1E-10 is still not zero!).