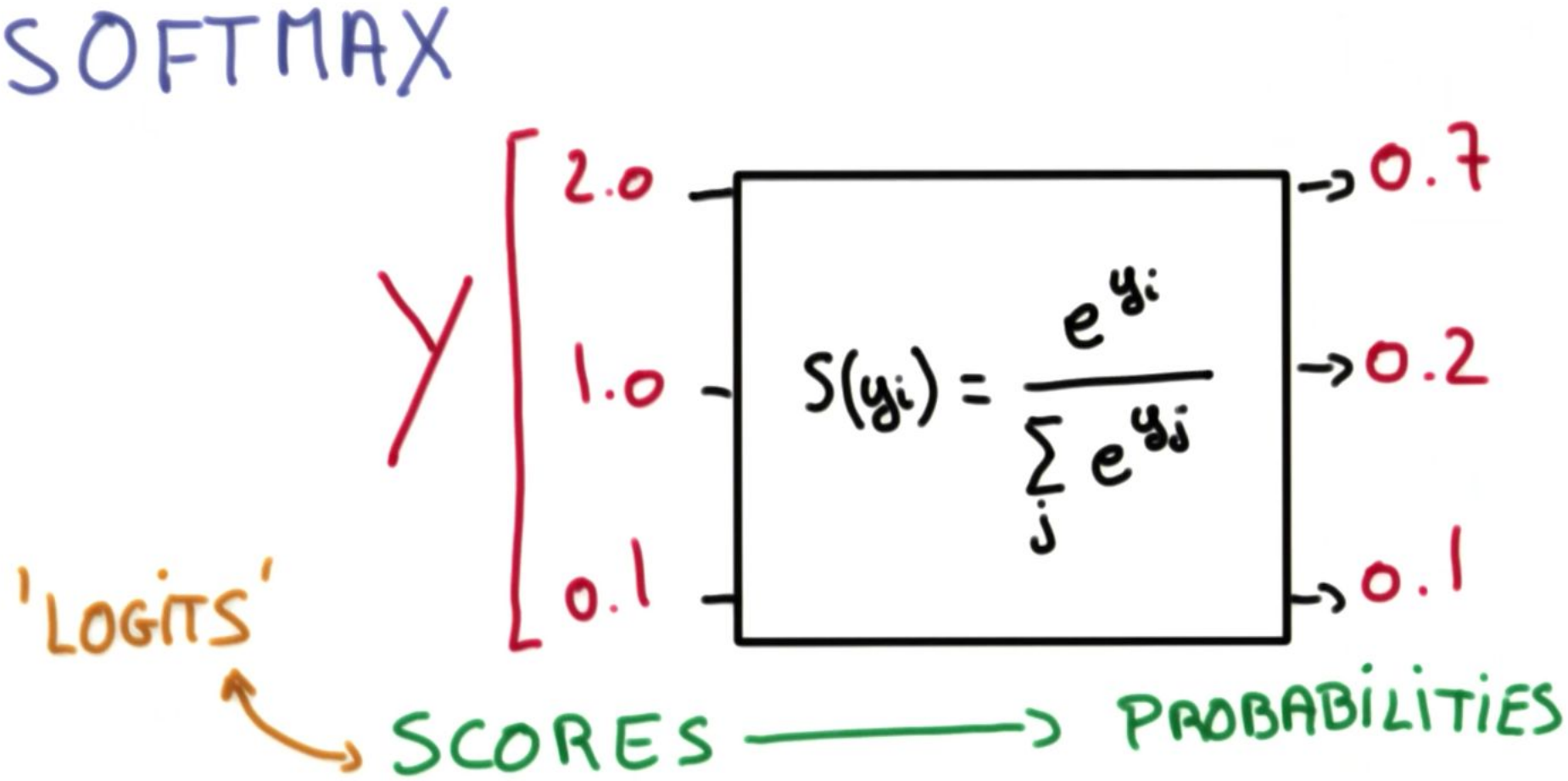
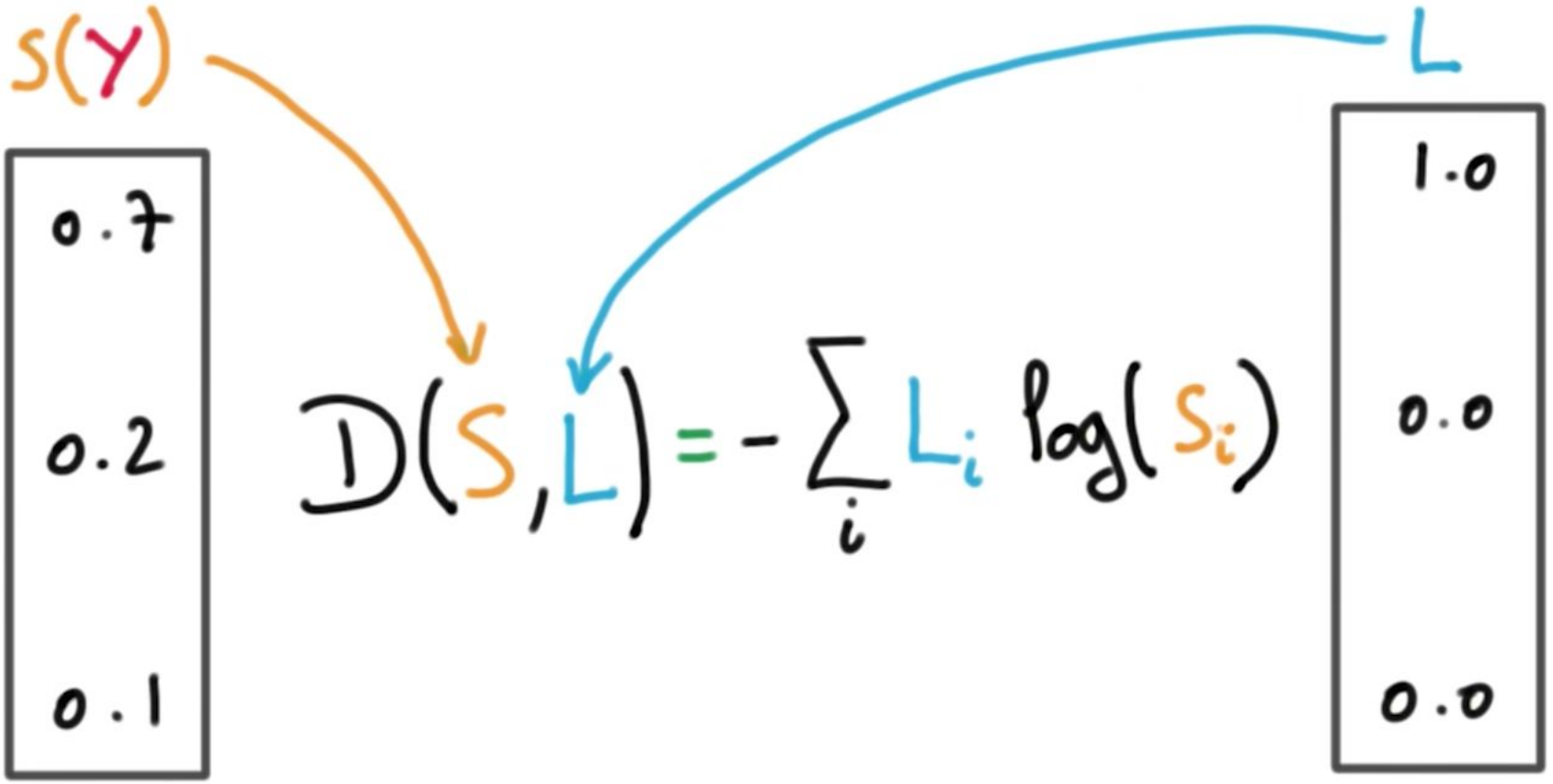
My first Neural Nets Lasagne: MNIST

Data Minings in Action: Trends

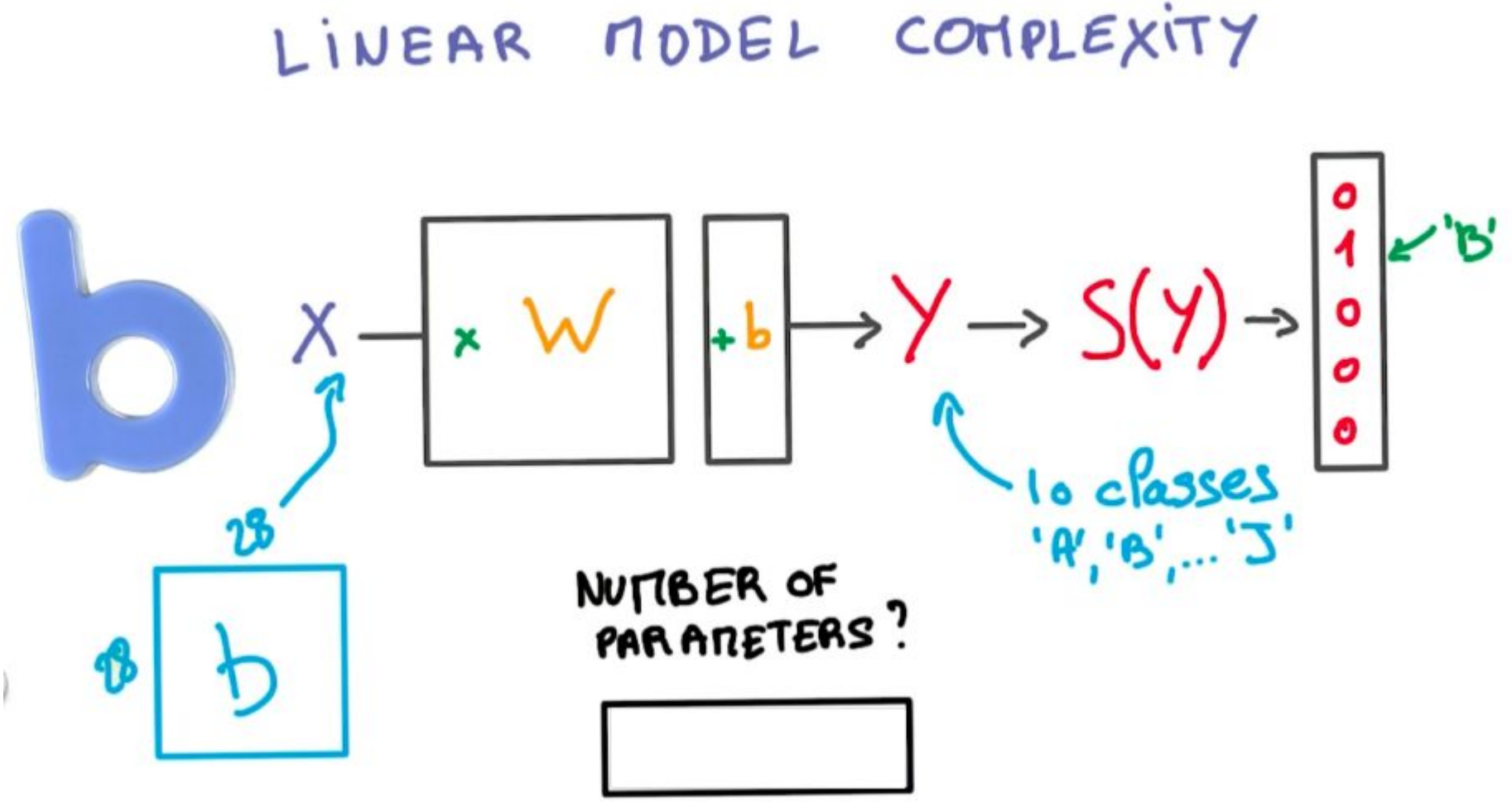



```
from lasagne.nonlinearities import softmax
```

CROSS-ENTROPY

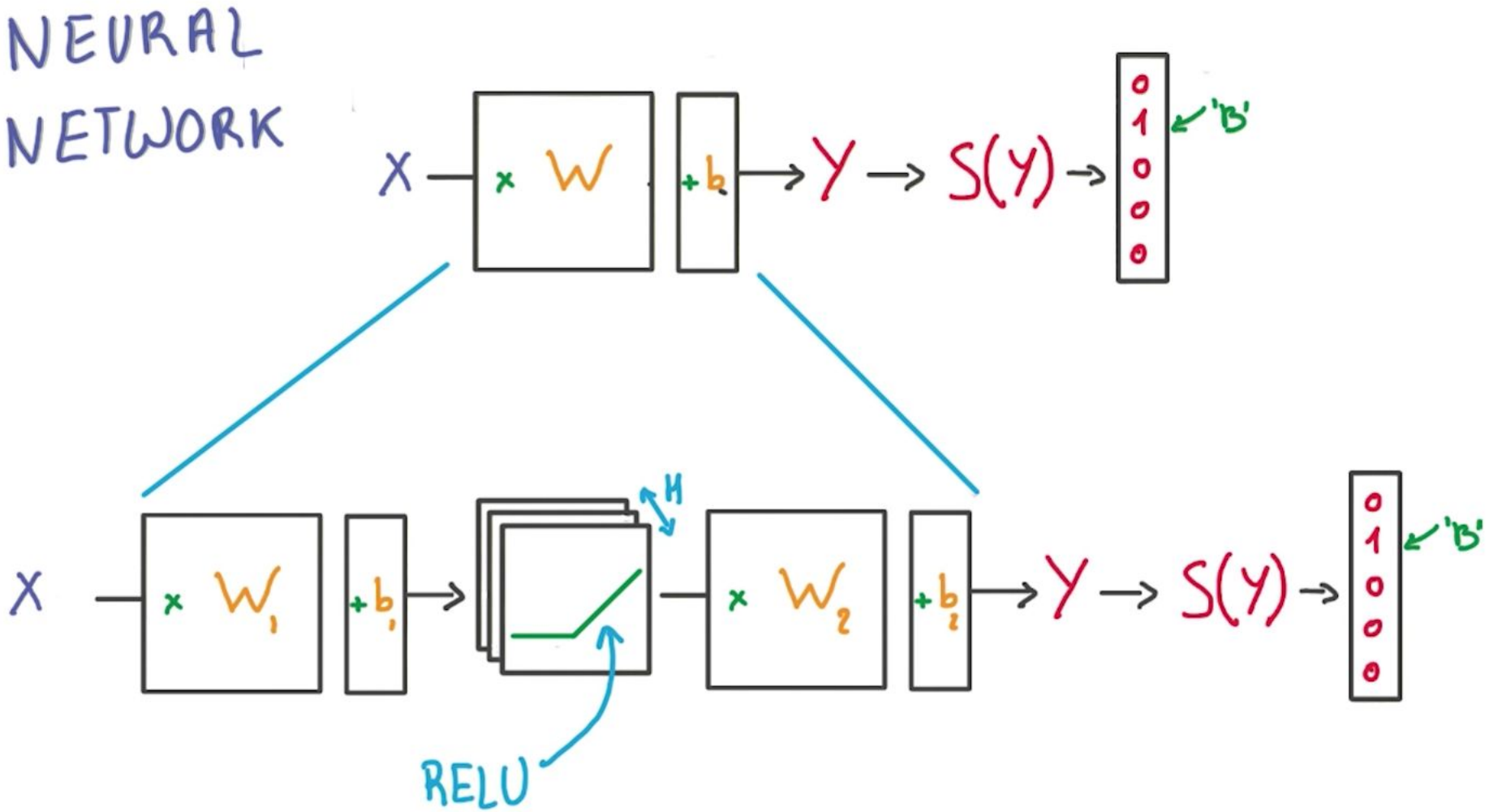



```
from lasagne.objectives import categorical_crossentropy
```




```
from lasagne.layers import InputLayer, DenseLayer

l_in  = InputLayer(shape=(None, 1, 28, 28), input_var=input_var)
l_out = DenseLayer(l_in, num_units=10, nonlinearity=softmax)
```

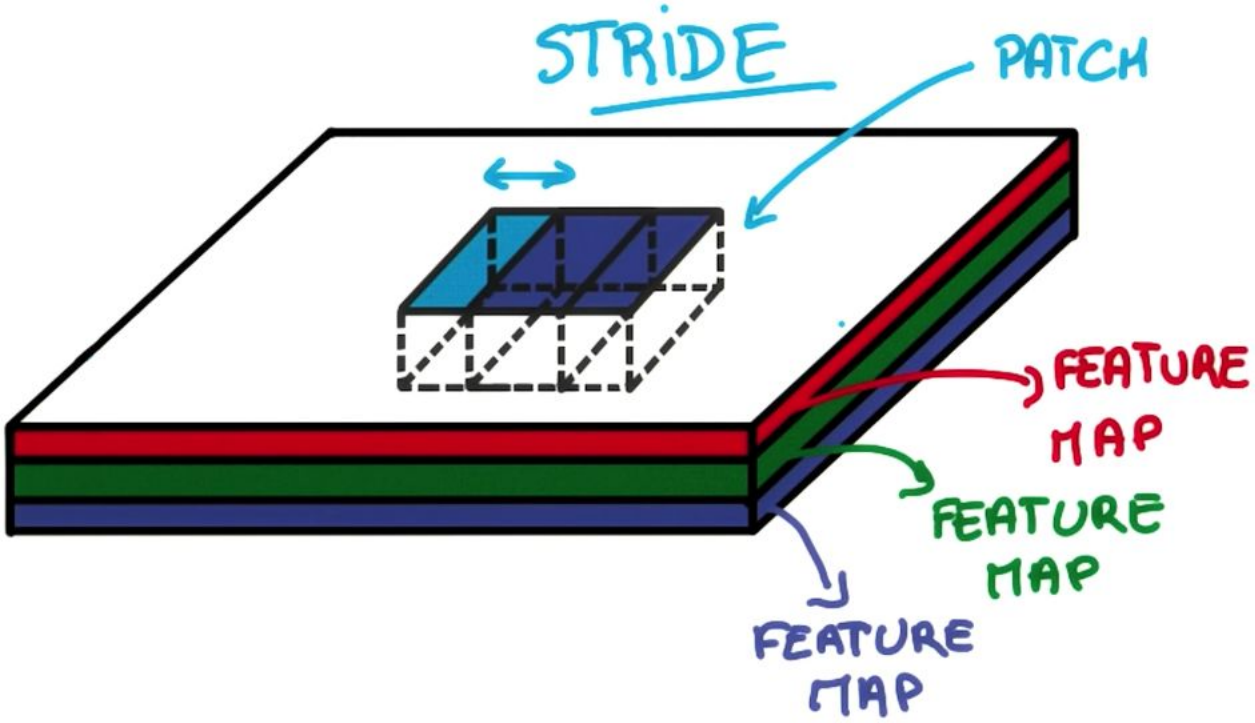



```
from lasagne layers import InputLayer, DenseLayer


l_in  = InputLayer(shape=(None, 1, 28, 28), input_var=input_var)
l_mid = DenseLayer(l_in,  num_units=100, nonlinearity=rectyfy)
l_out = DenseLayer(l_mid, num_units=10, nonlinearity=softmax)
```

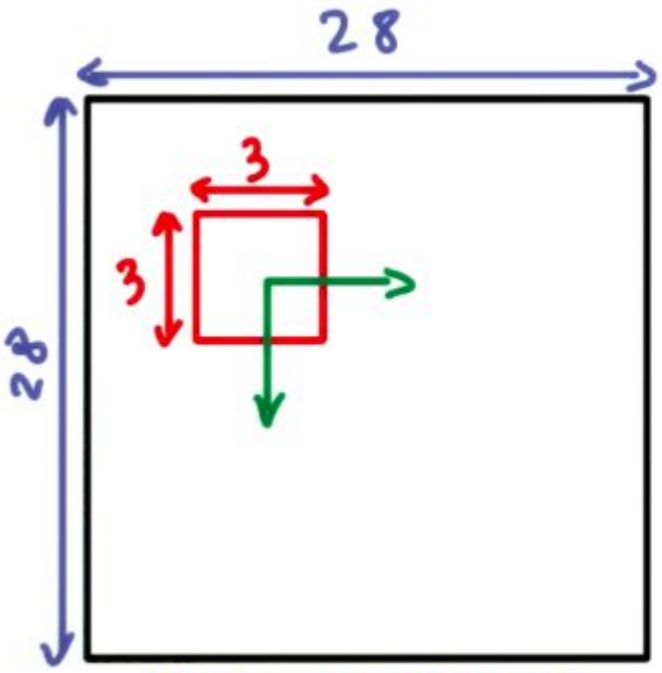
CONVOLUTIONS

CONVOLUTIONAL LINGO



https://youtu.be/jajksuQW4mc?t=150

STRIDES, DEPTH & PADDING

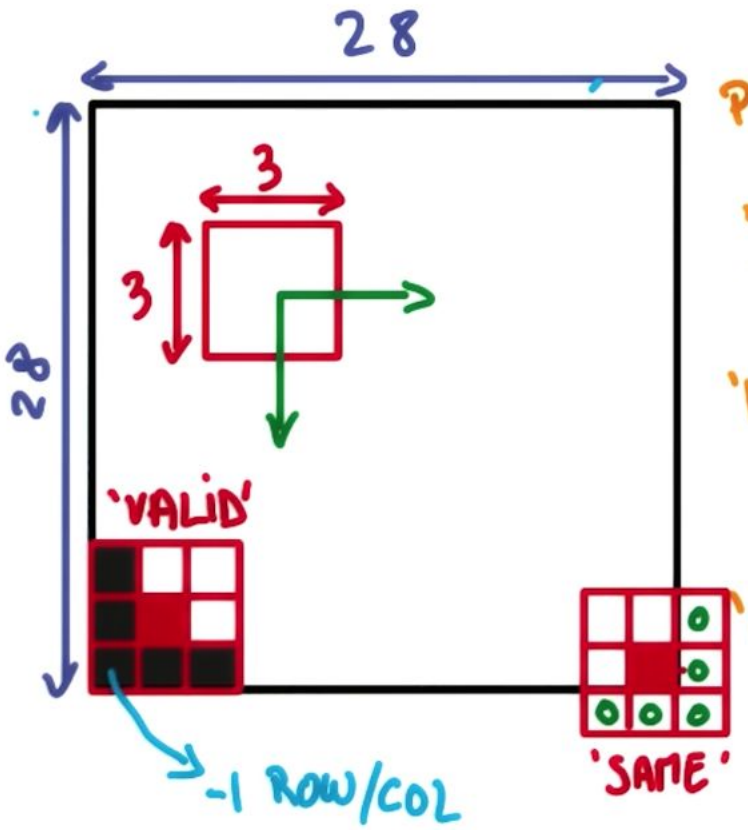


INPUT DEPTH = 3

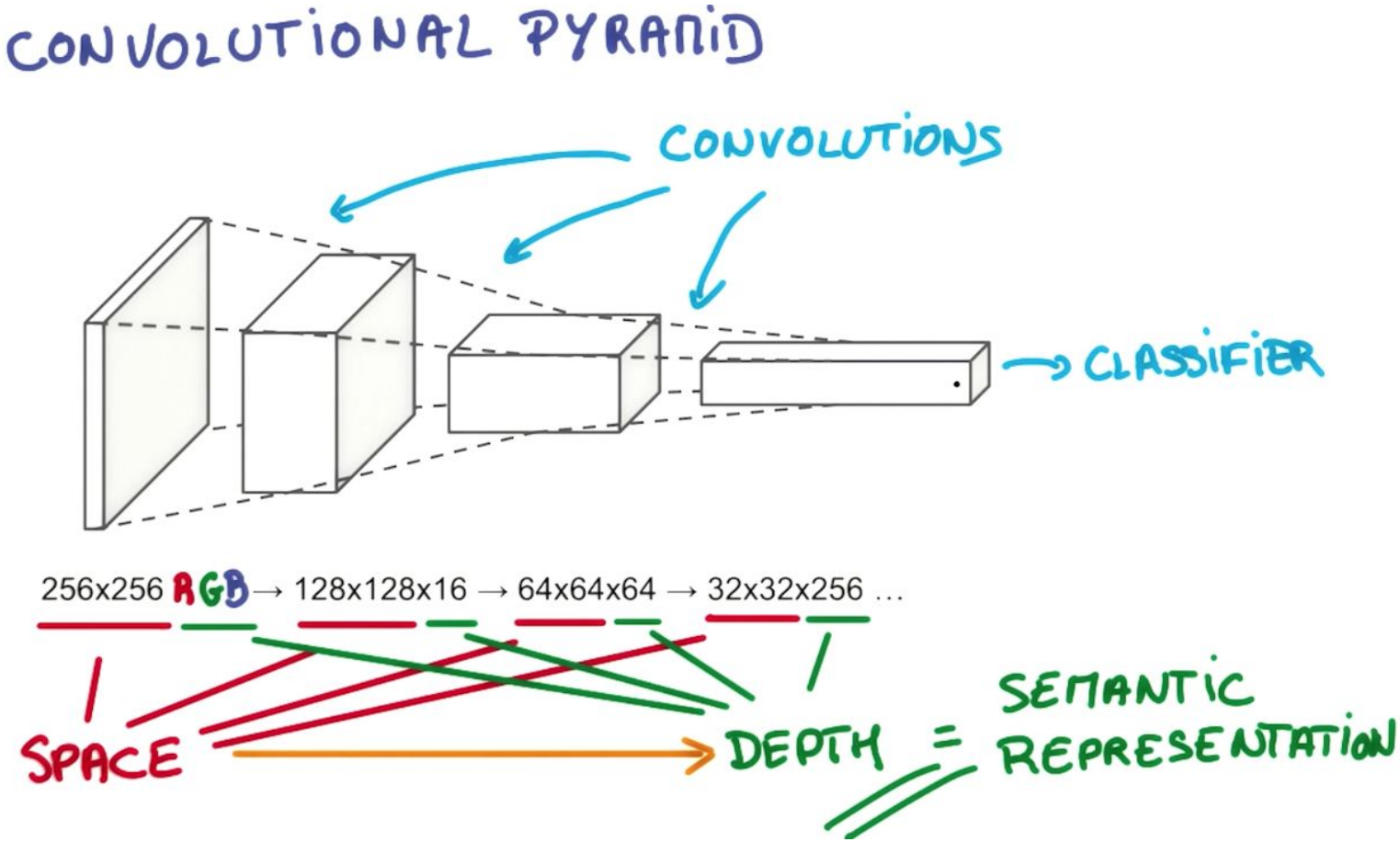
OUTPUT DEPTH = 8

		OUTPUT		
PADDING	STRIDE	WIDTH	HEIGHT	DEPTH
'SAME'	1			
'VALID'	1			
'VALID'	2			

STRIDES, DEPTH & PADDING



		OUTPUT		
PADDING	STRIDE	WIDTH	HEIGHT	DEPTH
'SAME'	1	28	28	8
'VALID'	1	26	26	8
'VALID'	2	13	13	8

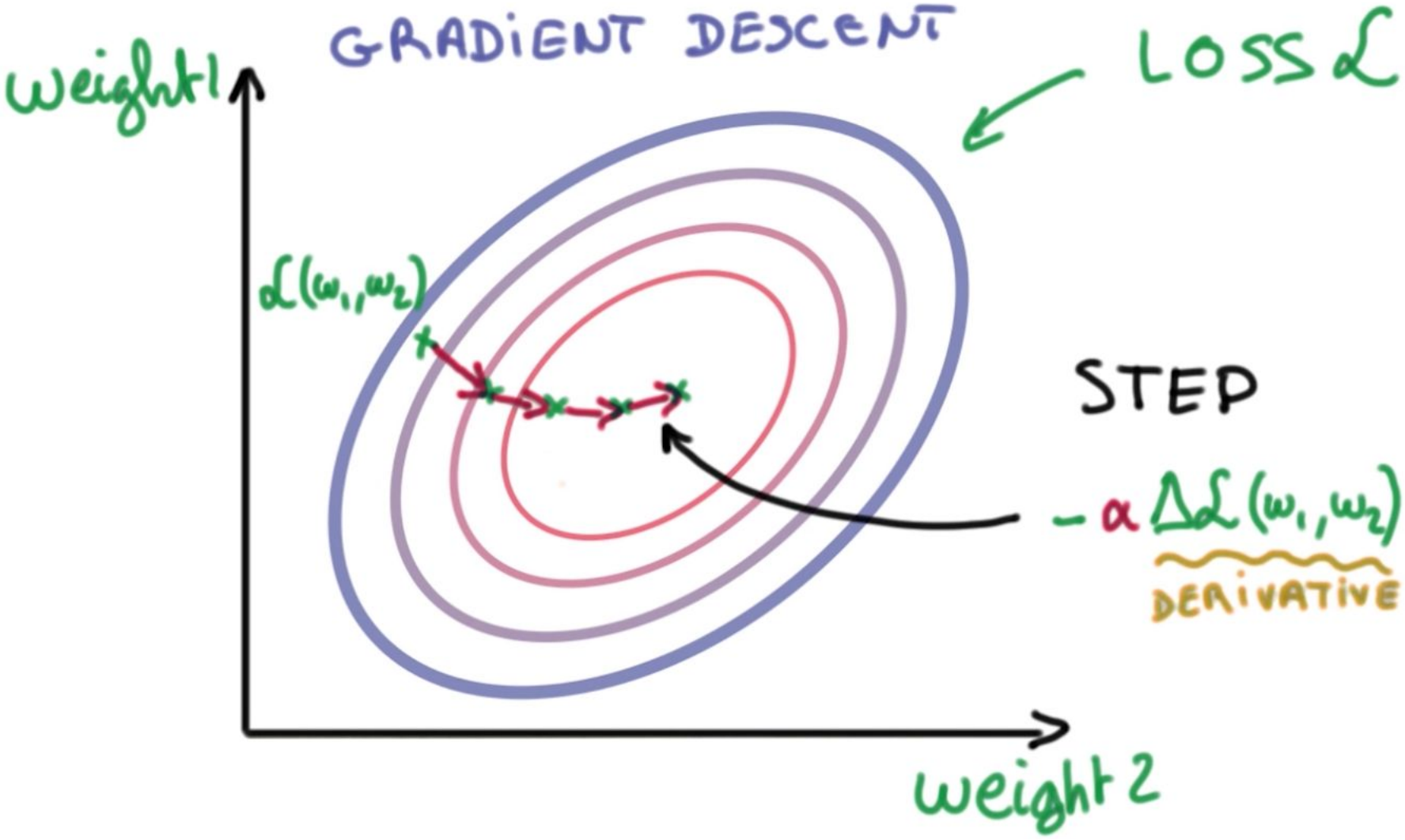



```
from lasagne layers import Conv2DLayer, DenseLayer
l_in  = InputLayer(shape=(None, 3, 256, 2256), input_var=input_var)
l_mid = Conv2DLayer(l_in,  num_units=16, nonlinearity=rectyfy)

l_mid = Conv2DLayer(l_in,  num_units=64, nonlinearity=rectyfy)

l_mid = Conv2DLayer(l_in,  num_units=256, nonlinearity=rectyfy)

...
```


GRADIENT DESCENT
LOSS $\mathcal{L}$
weight 1
$\mathcal{L}(w_1, w_2)$
STEP
$-\alpha \Delta \mathcal{L}(w_1, w_2)$
DERIVATIVE
weight 2