# Binary Cross Entropy / Log Loss (Classification)

$$J = \frac{1}{N}\sum_{i=0}^{N-1}\left(y_i \cdot \ln(aL_i) + (1-y_i)\cdot \ln(1-aL_i)\right)$$

over N batches

$$\frac{\partial J}{\partial aL} = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{y_i}{aL_i} + (1-y_i)\cdot\frac{1}{1-aL_i}\cdot(-1)\right)$$

$$= \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{y_i}{aL_i} - \frac{(1-y_i)}{(1-aL_i)}\right)$$

$$= \text{〃}\left(\frac{y_i - \cancel{y_i aL_i} - aL_i + \cancel{y_i aL_i}}{aL_i(1-aL_i)}\right)$$

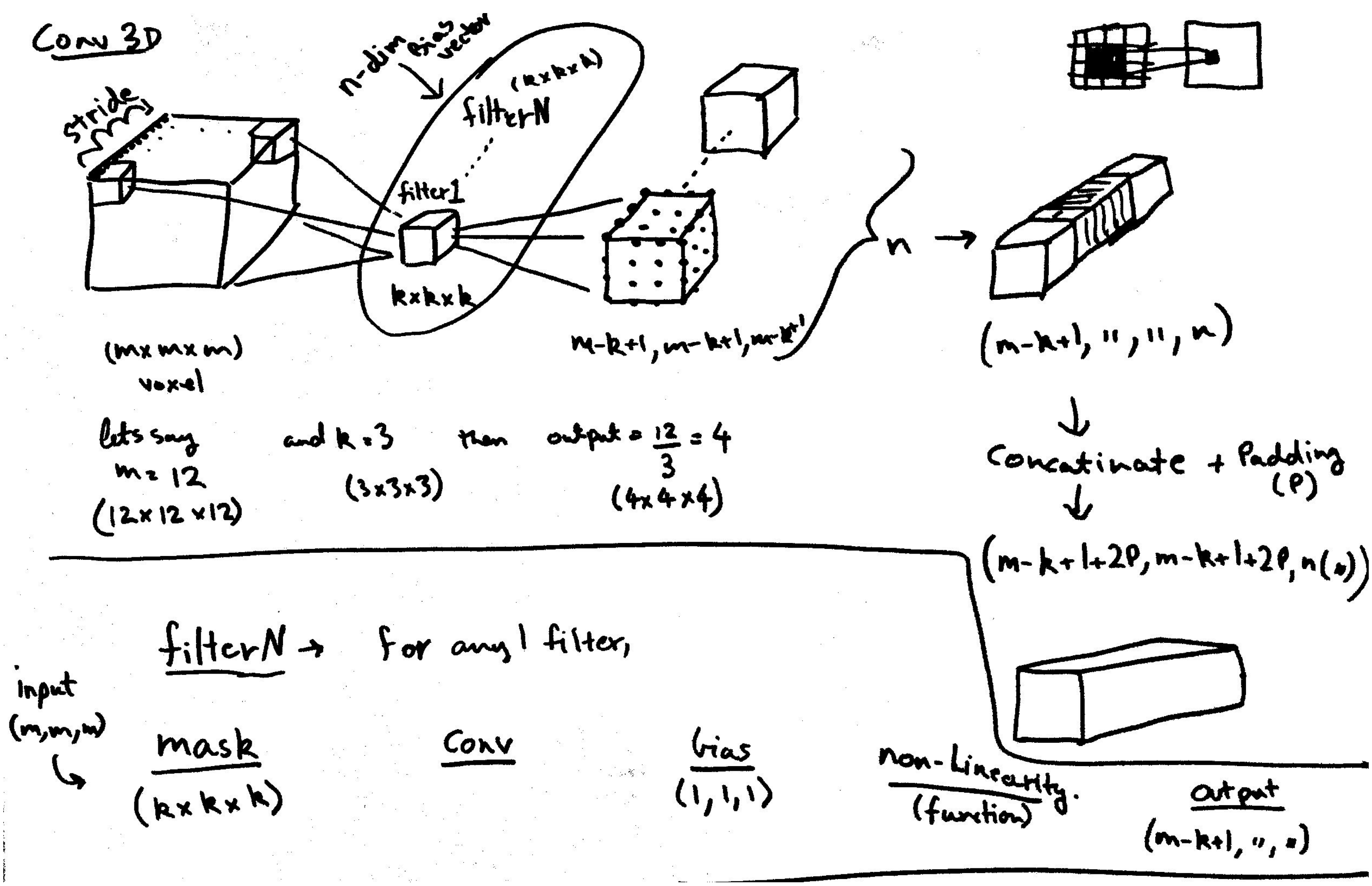
$$\Rightarrow \frac{\partial J}{\partial aL} = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{y_i - aL_i}{aL_i(1-aL_i)}\right)$$

name: model( $\underbrace{\text{layerinfo}}$, $\alpha$, decay, optimization, regularization, X, Y)

$\underbrace{L[\ ]}$

size = # of layers.

L[i] = "Activation", nL



Conv 3D
stride
n-dim bias vector
(k×k×k)
filterN
filter1
k×k×k
(m×m×m)
voxel
m-k+1, m-k+1, m-k+1
n →
(m-k+1, '', '', n)
lets say m=12 (12×12×12)
and k=3 (3×3×3)
then output = 12/3 = 4 (4×4×4)
Concatinate + Padding (P)
(m-k+1+2P, m-k+1+2P, n(*))
filterN → for any 1 filter,
input (m,m,m)
mask (k×k×k)
Conv
bias (1,1,1)
non-Linearity. (function)
output (m-k+1, '', '')