CNN – An approach

Create Each Filter

* Max pool filter (better in back propagation). Average? L2 pool?
* ReLU Filter (Activation). Leaky?
* Convolution weight kernel
* Sharpen? Noise?

Sequence types

(Conv -> ReLU -> Conv -> ReLU -> Pool) X 3

Forward pass -> calculate loss -> back pass -> gradient descent

calculate loss step: we will batch the losses of multiple convolution agents and propagate the same values throughout many “cultures” they learn at the same time.

A thousand brains vote on a solution and come to an agreement after traditional neural network processing and population, but vote like a mini jury.

Gradient for Conv : Derivative of multiplying a kernel is just dev(axw) where a is a pixel value and w is a weight = a

Gradient for Max: pass greatest value backwards

Gradient for ReLU: 1 if above zero and 0 otherwise

These will be used as the gradient for the backward pass. They will be multiplied by the previous gradient in the case of “chain rule”