Neural Networks

Declan Groves

June 30 2016

Outline

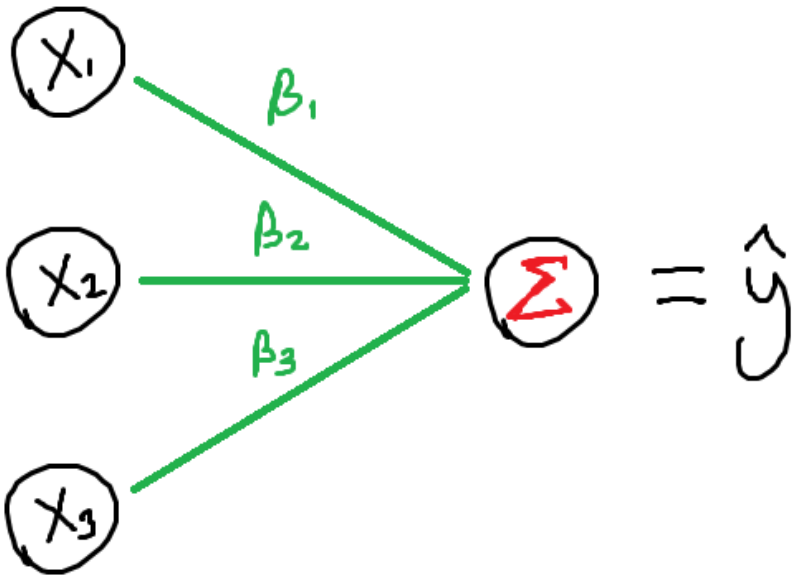
Overview

- Hypest ML
- Good at unstructured problems
- Suboptimal at structured problems

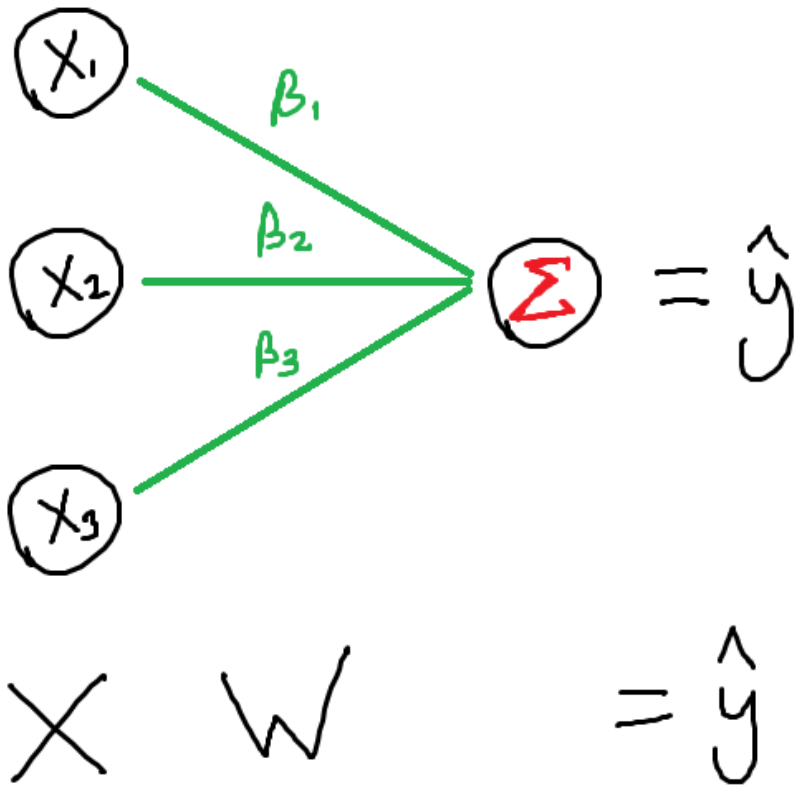
History

- Around since the 1950s
- Resurgence in 1970s
- Resurgence in late 2000s

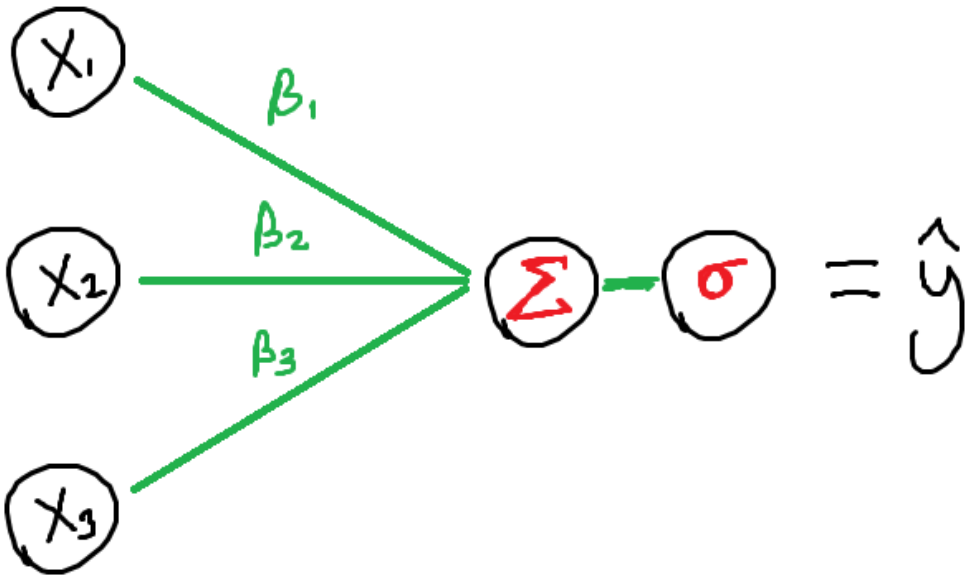
A graphical linear model



A graphical linear model

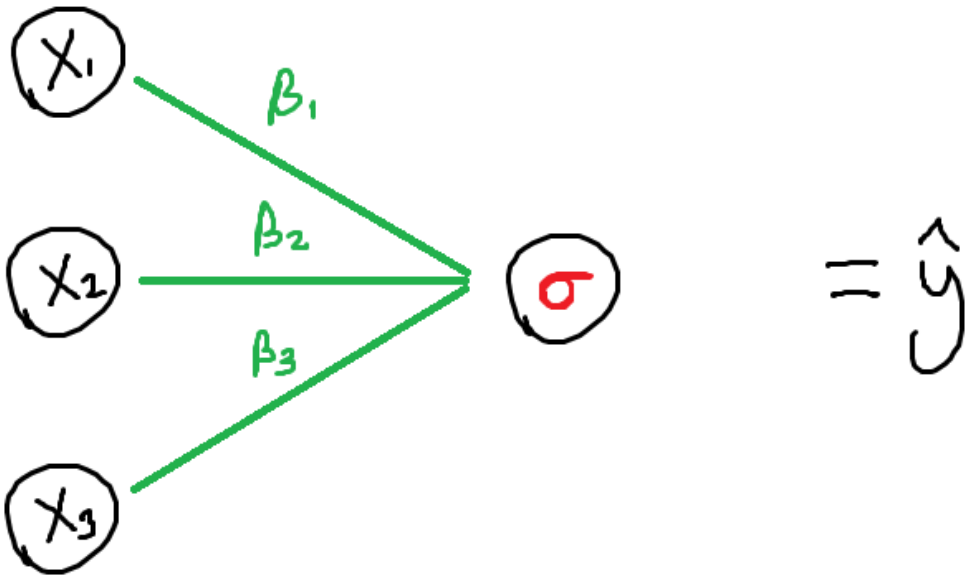


A graphical linear model



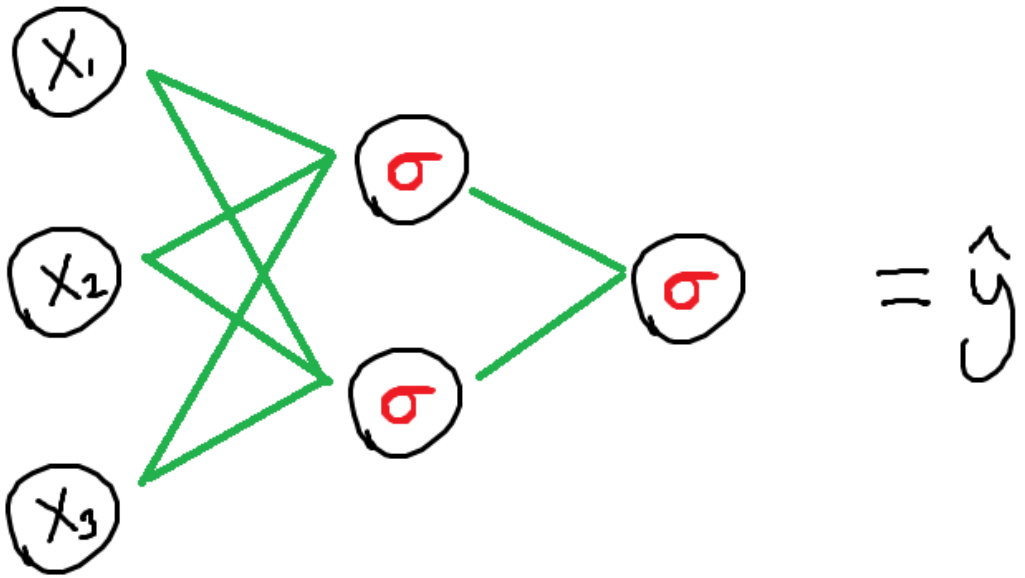
$$\sigma(X \quad W) \quad = \hat{y}$$

A graphical linear model

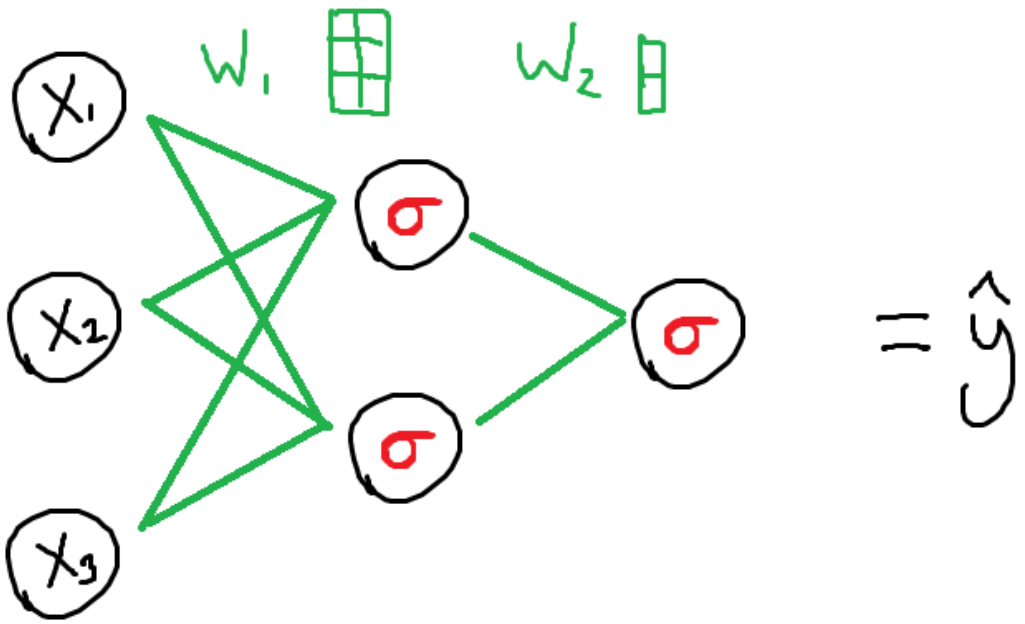


$$\sigma(X \quad W) \quad = \hat{y}$$

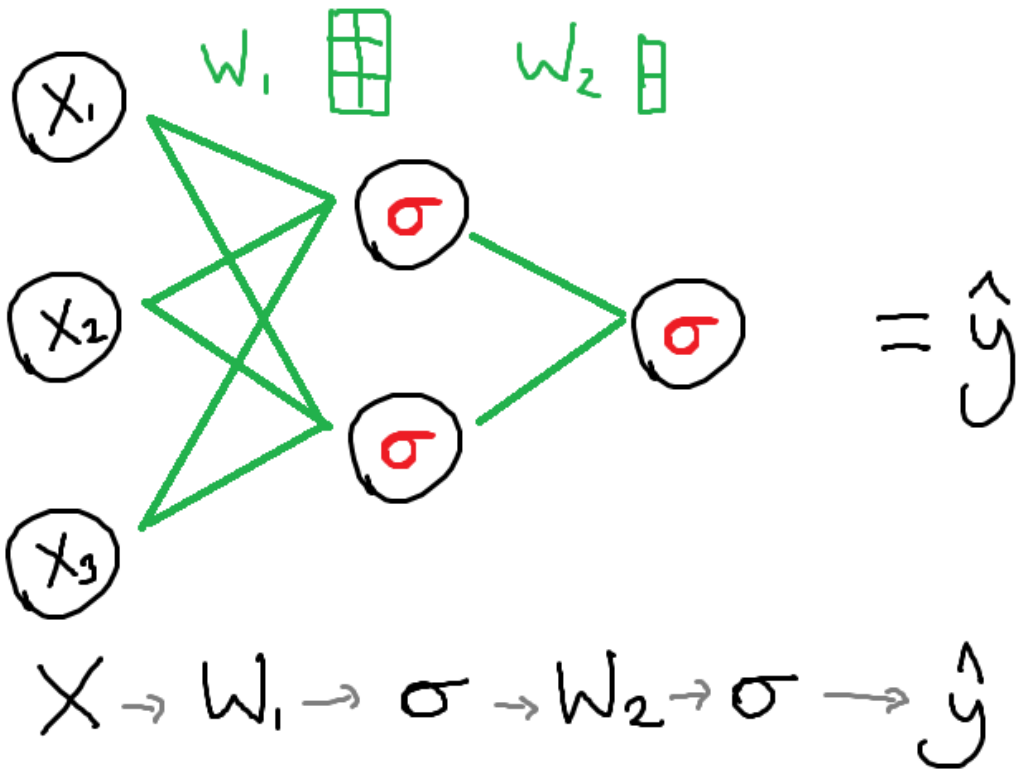
A simple neural network



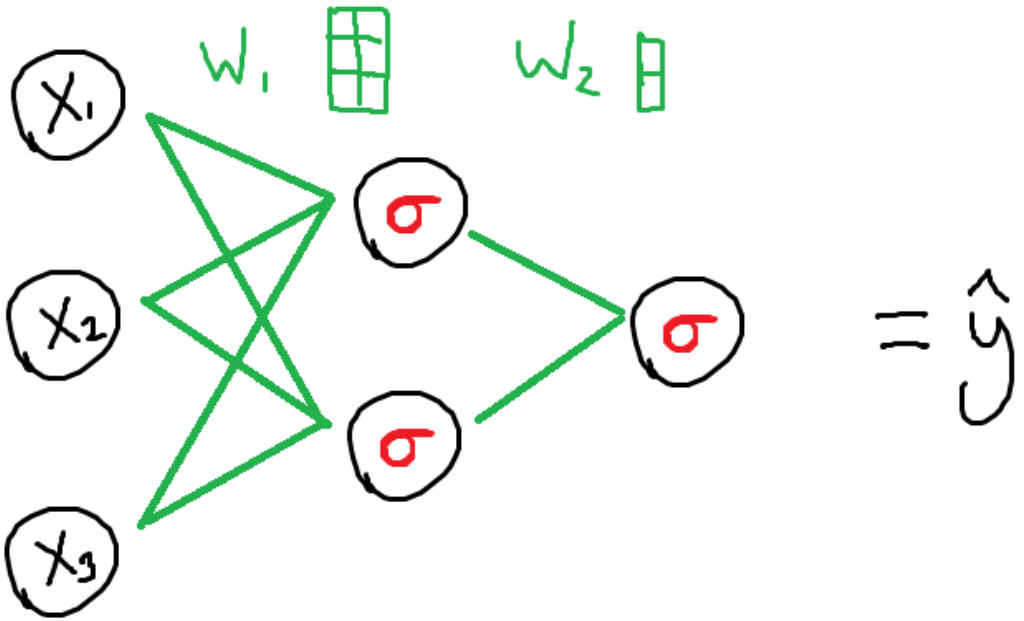
A simple neural network



A simple neural network



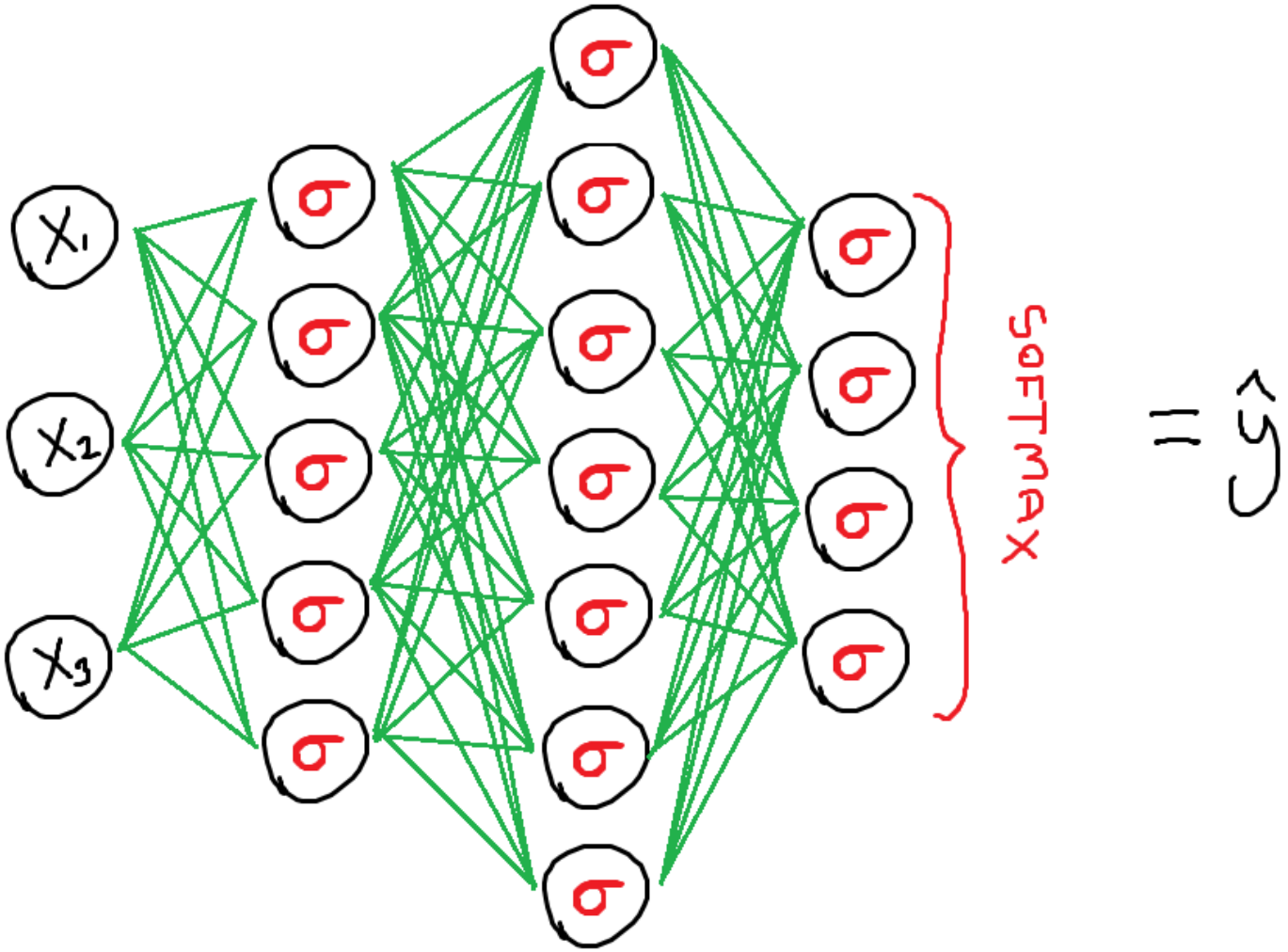
A simple neural network



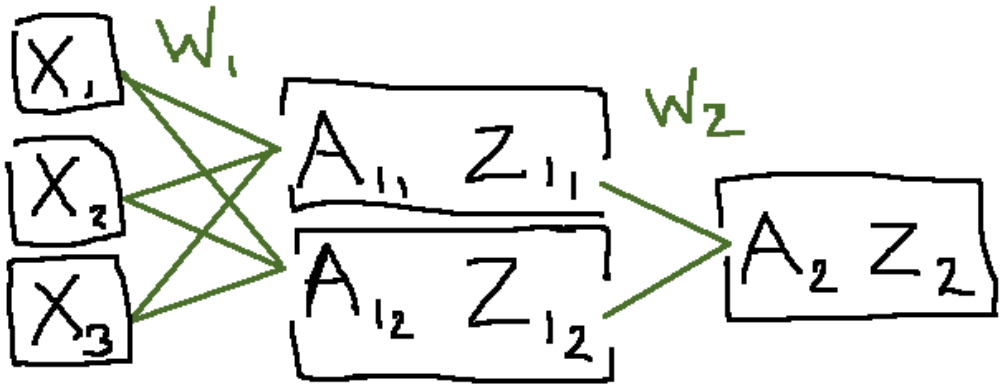
$$X \rightarrow W_1 \rightarrow \sigma \rightarrow W_2 \rightarrow \sigma \rightarrow \hat{y}$$

$$\sigma(\sigma(XW_1)W_2) = \hat{y}$$

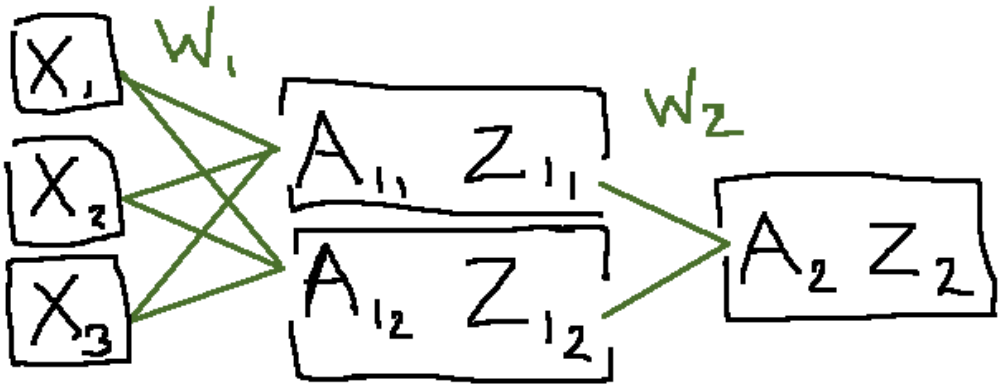
A multilayer multinomial classifier



Forward pass



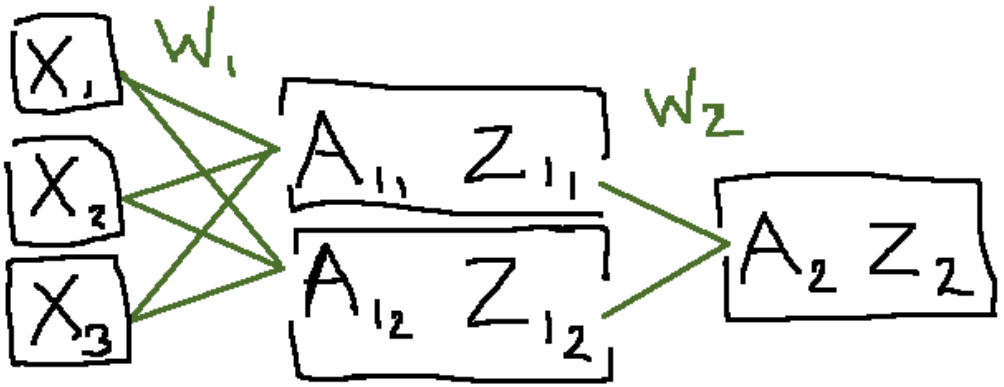
Forward pass



$$A_1 = XW_1$$
$$Z_1 = \sigma(A_1)$$
$$A_2 = Z_1 W_2$$
$$Z_2 = \sigma(A_2) = \hat{y}$$

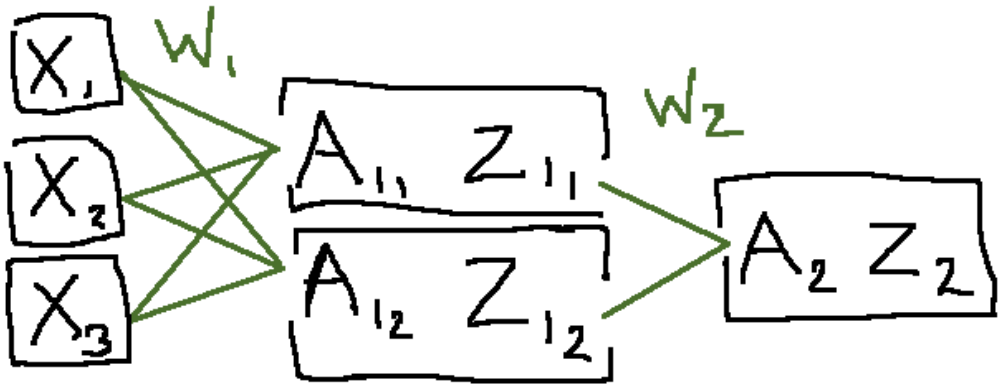
Forward pass

Backwards pass



$$J = \text{err}(y, \hat{y}) \; ; \; J \leftarrow \frac{1}{2}(y - \hat{y})^2$$

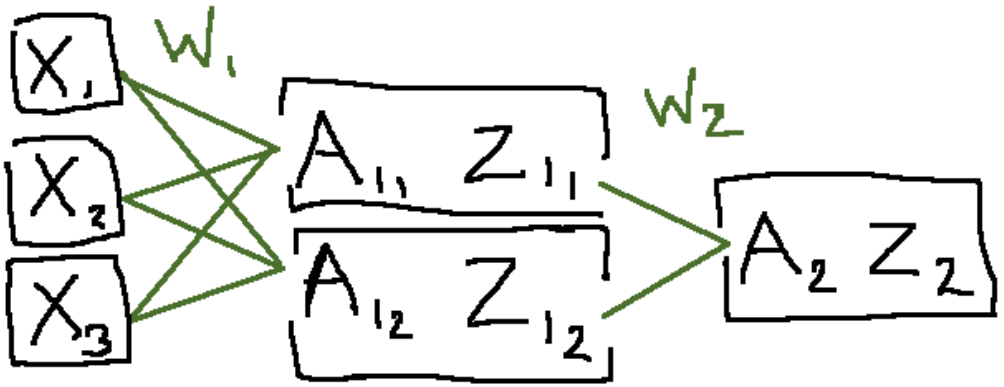
Backwards pass



$J = err(y, \hat{y})$; $J \leftarrow \frac{1}{2}(y - \hat{y})^2$

$\partial J / \partial W_2 =$

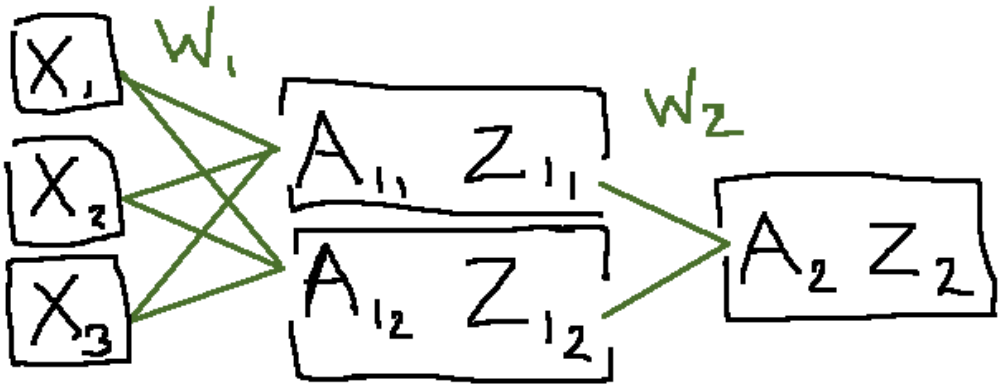
Backwards pass



$$J = \text{err}(y, \hat{y}) \; ; \; J \leftarrow \frac{1}{2}(y - \hat{y})^2$$

$$\partial J / \partial W_2 = \partial J / \partial \hat{y} \; \partial \hat{y} / \partial W_2$$

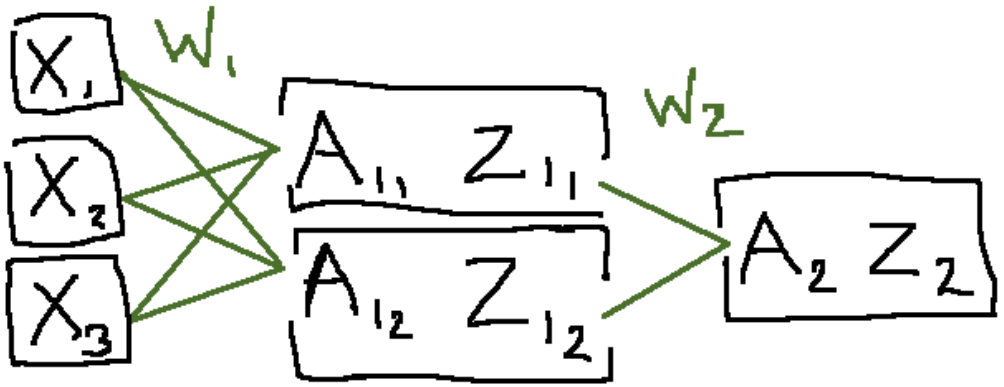
Backwards pass



$$J = \text{err}(y, \hat{y}) \; ; \; J \leftarrow \frac{1}{2}(y - \hat{y})^2$$

$$\partial J / \partial W_2 = \partial J / \partial \hat{y} \; \partial \hat{y} / \partial A_2 \; \partial A_2 / \partial W_2$$

Backwards pass

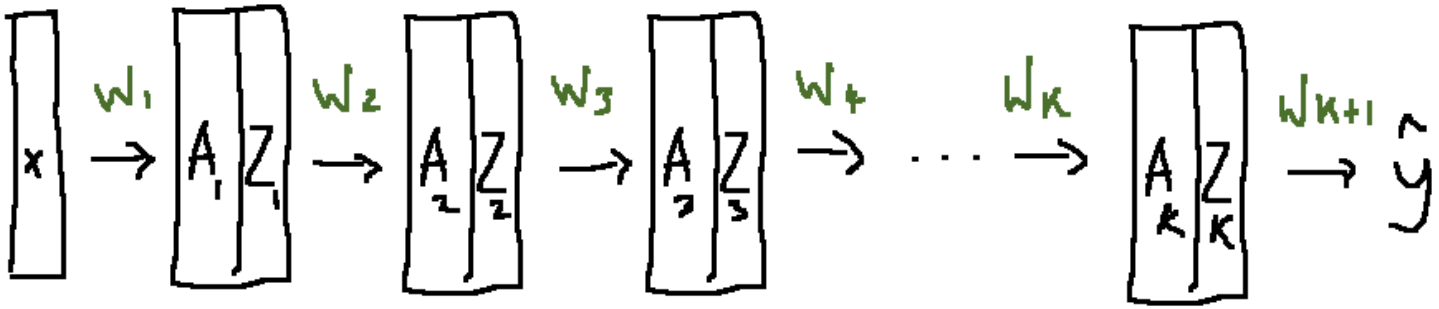


$$J = \text{err}(y, \hat{y}) \; ; \; J \leftarrow \frac{1}{2}(y - \hat{y})^2$$

$$\partial J / \partial W_2 = \partial J / \partial \hat{y} \; \partial \hat{y} / \partial A_2 \; \partial A_2 / \partial W_2$$

$$= \underbrace{(y - \hat{y}) \; \sigma'(A_2)}_{\delta} \; Z_1$$

Backwards pass

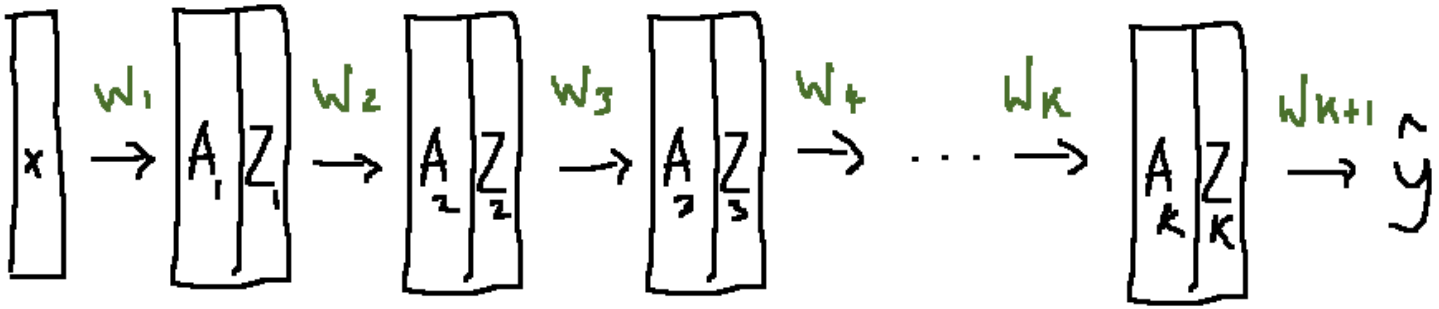


$$\partial J / \partial W_i =$$

$$(\partial J / \partial \hat{y}) \sigma'(A_K) W_K \sigma'(A_{K-1}) W_{K-1} \cdots$$

$$\cdots W_{i+1} \sigma'(A_i) Z_{i-1}$$

Backwards pass



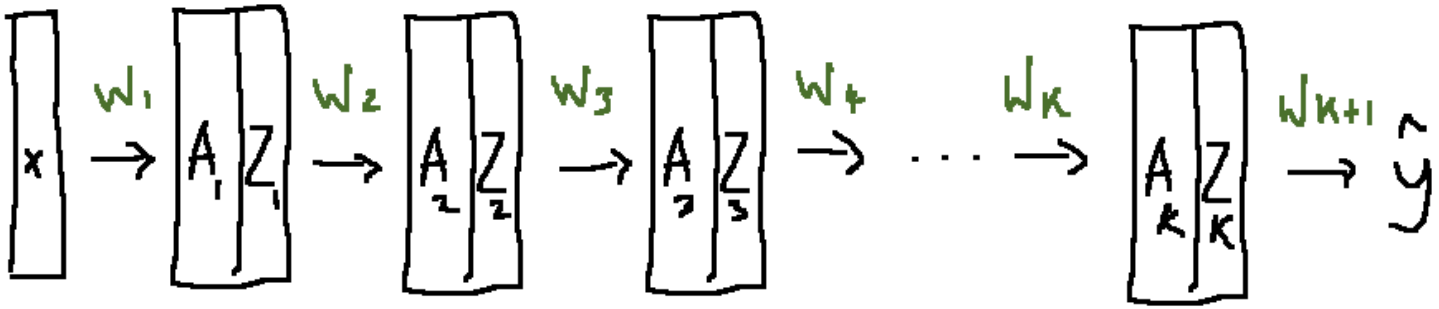
$$\partial J / \partial W_i =$$

$$(\partial J / \partial \hat{y}) \sigma'(A_K) W_K \sigma'(A_{K-1}) W_{K-1} \cdots$$

$$\cdots W_{i+1} \sigma'(A_i) Z_{i-1}$$

!!

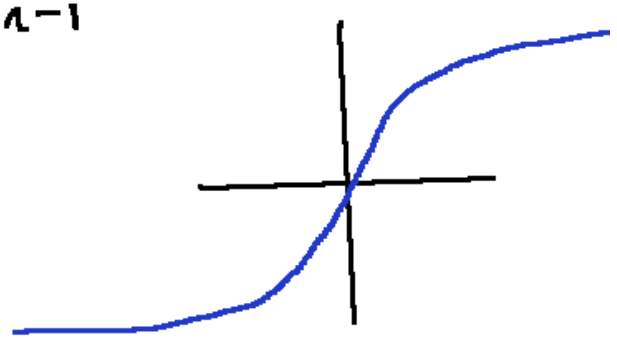
Backwards pass



$$\partial J / \partial W_i =$$

$$(\partial J / \partial \hat{y}) \sigma'(A_K) W_K \sigma'(A_{K-1}) W_{K-1} \cdots$$

$$\cdots W_{i+1} \sigma'(A_i) Z_{i-1}$$

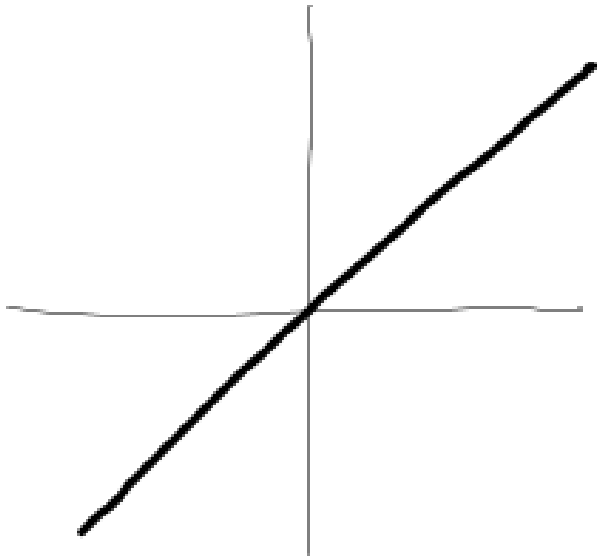


Resources

- Cool youtube series: https://www.youtube.com/watch?v=bxe2T-V8XRs
- Pain and toil: Elements of Statistical Learning ch11

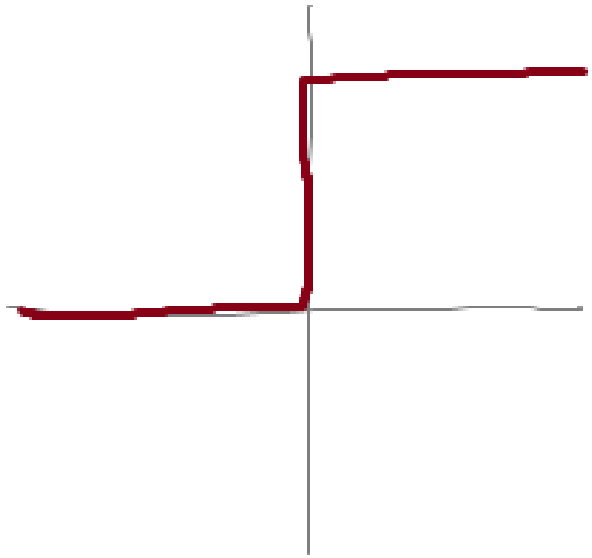
Linear

- Useless



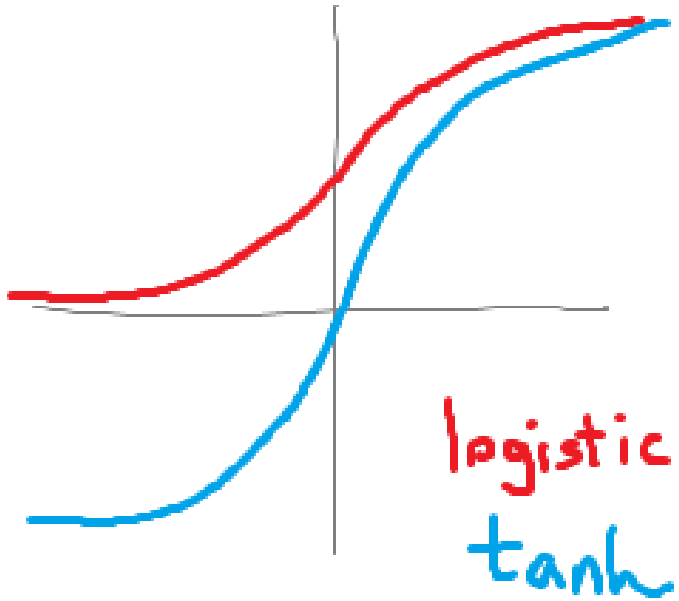
Threshold

- Similar to biological neuron
- No gradient



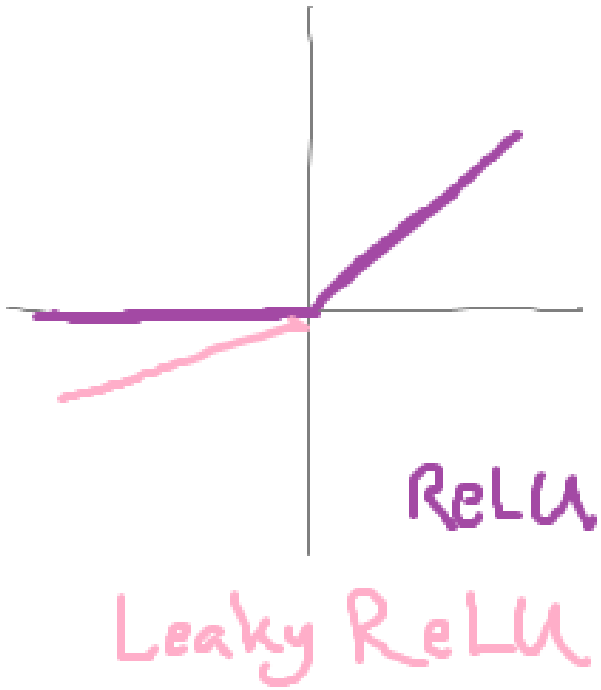
Sigmoid

- *tanh* preferred
- Gradients can vanish



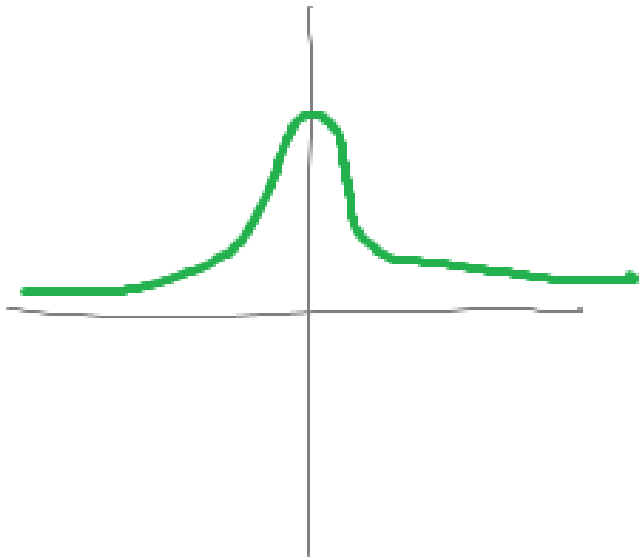
ReLU

- No vanishing gradient
- Cheap to compute
- Can explode and die
- Popular with CNNs



Radial basis functions

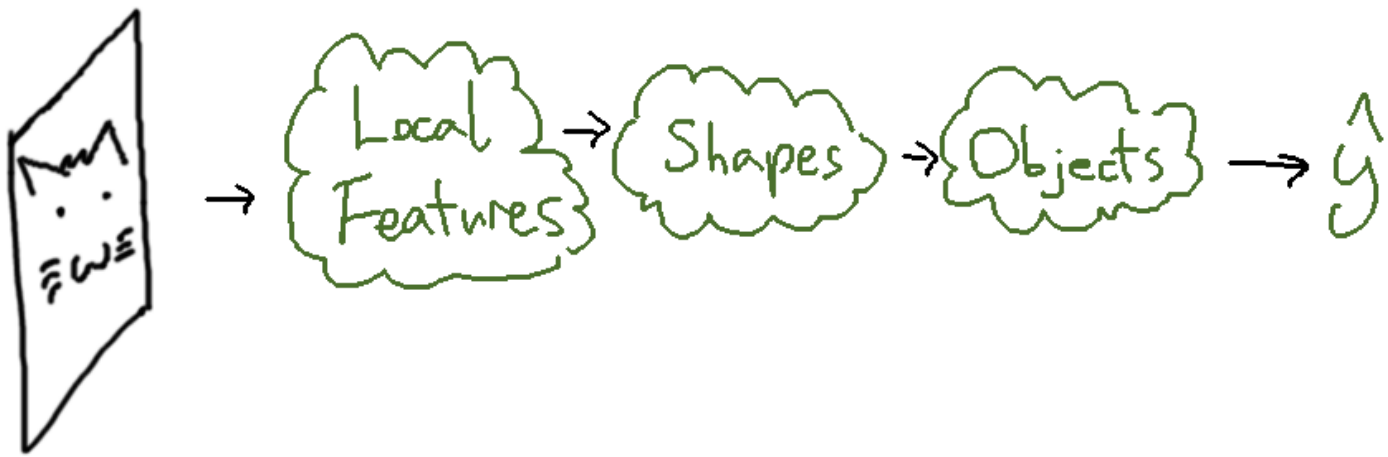
- Gaussian + others
- Train very quickly
- Good at interpolation



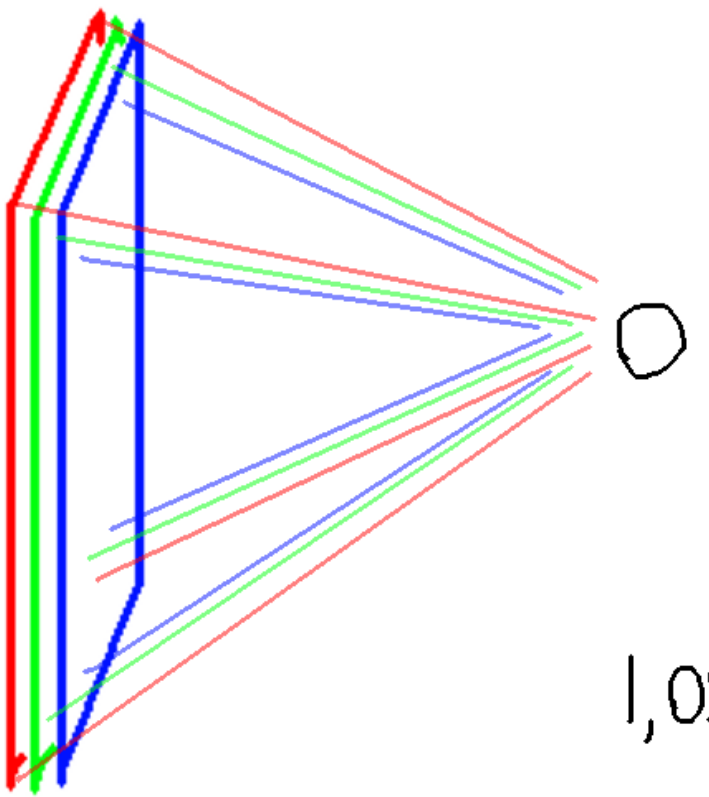
Convolutional Neural Networks

- Image recognition killer

Conceptual structure



MLP $\rightarrow$ too many weights!

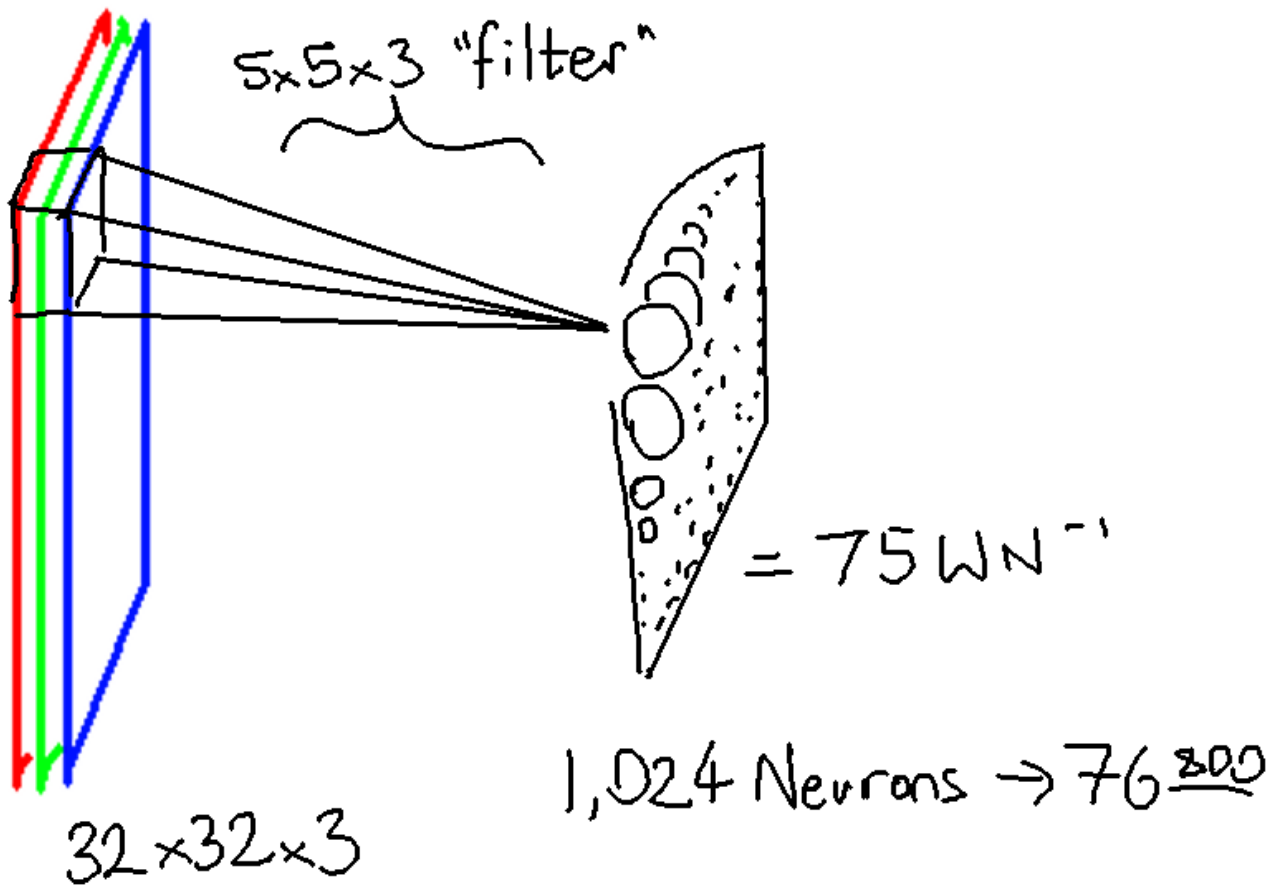


$= 3,072$ weights N^{-1}

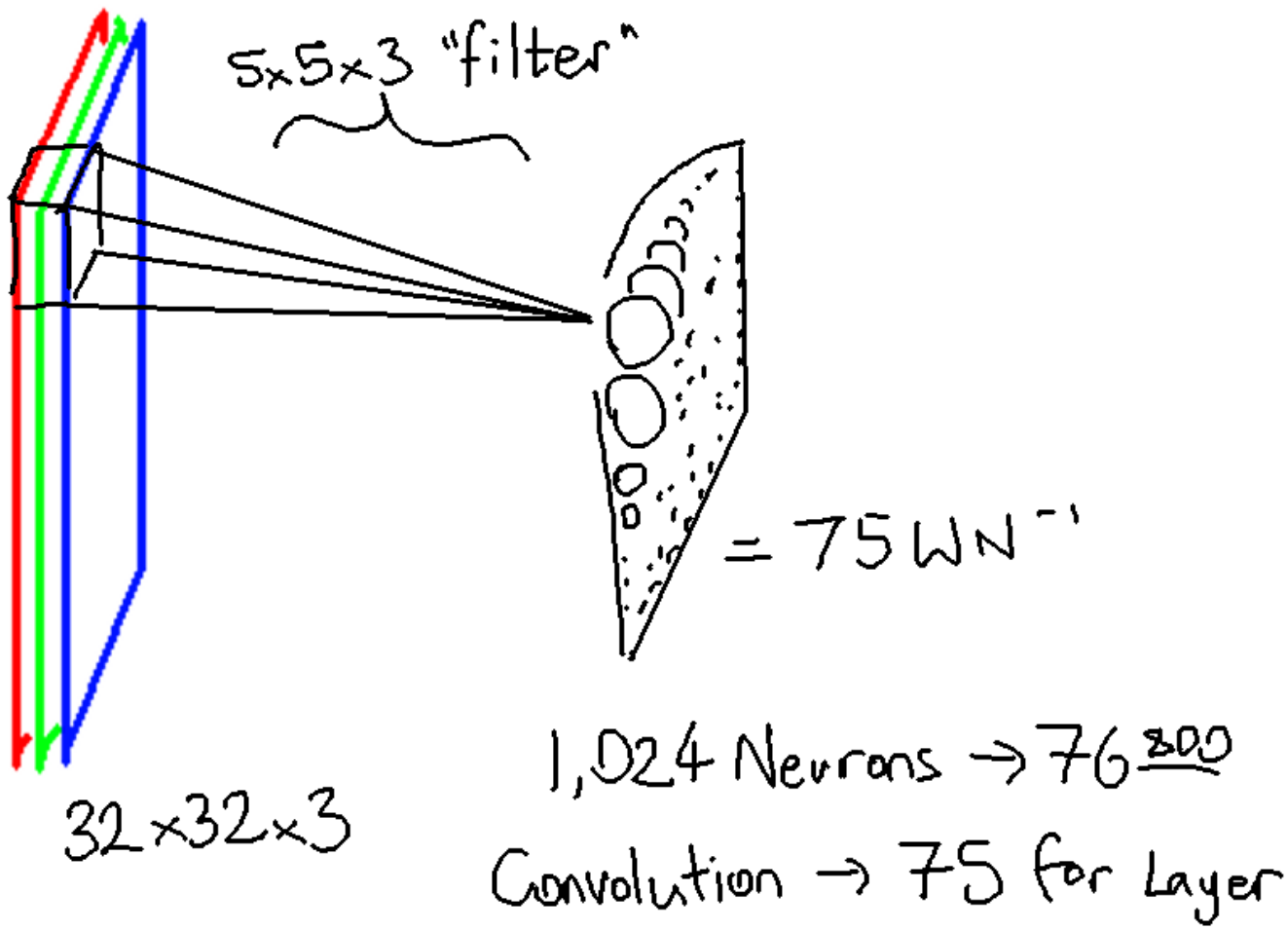
1,024 Neurons

$\rightarrow$ >3M weights

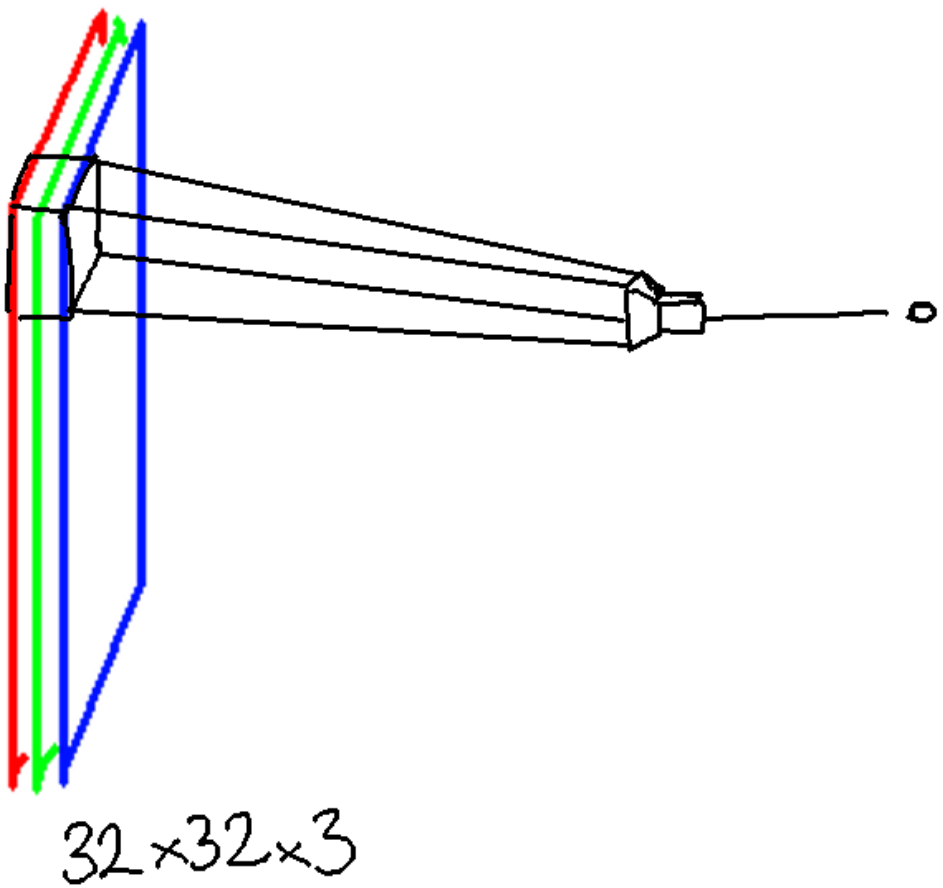
Local connectivity



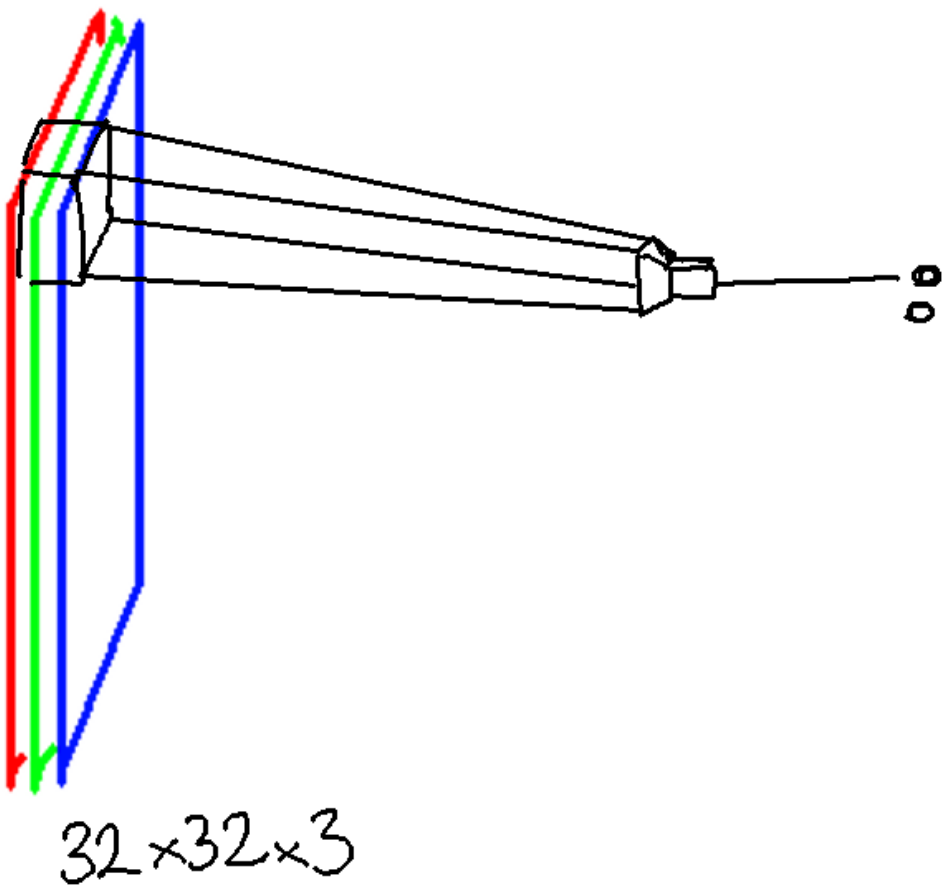
Local connectivity + convolution



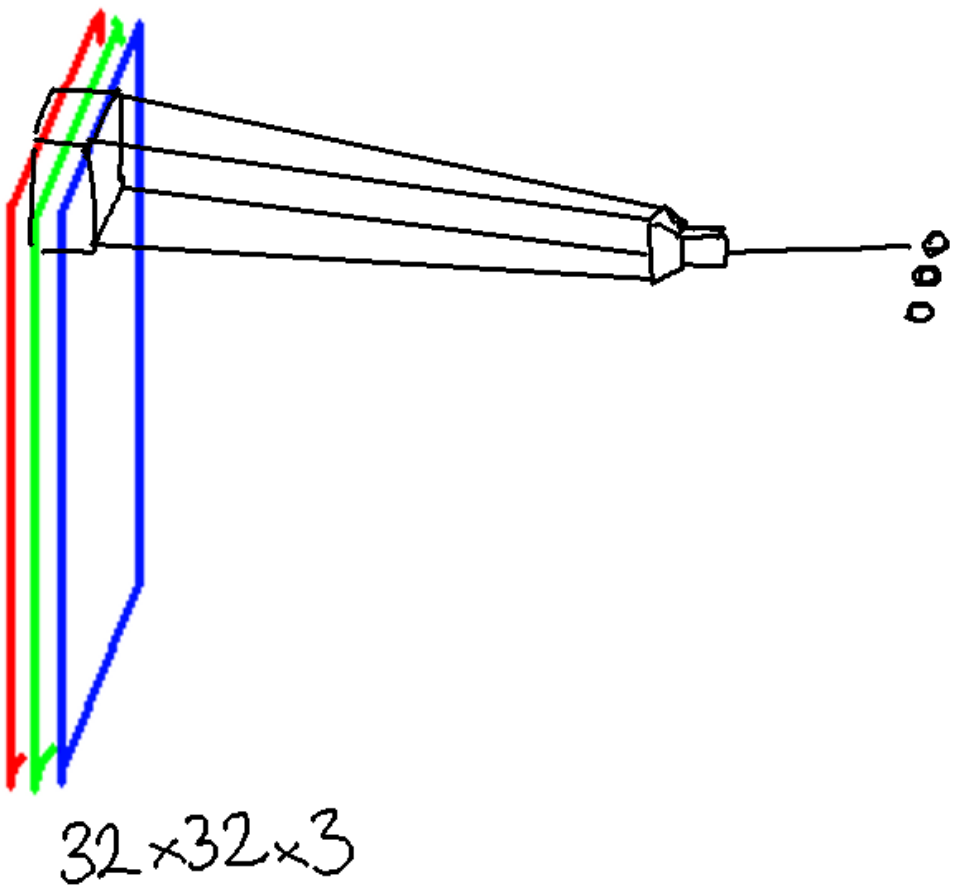
Local connectivity + convolution



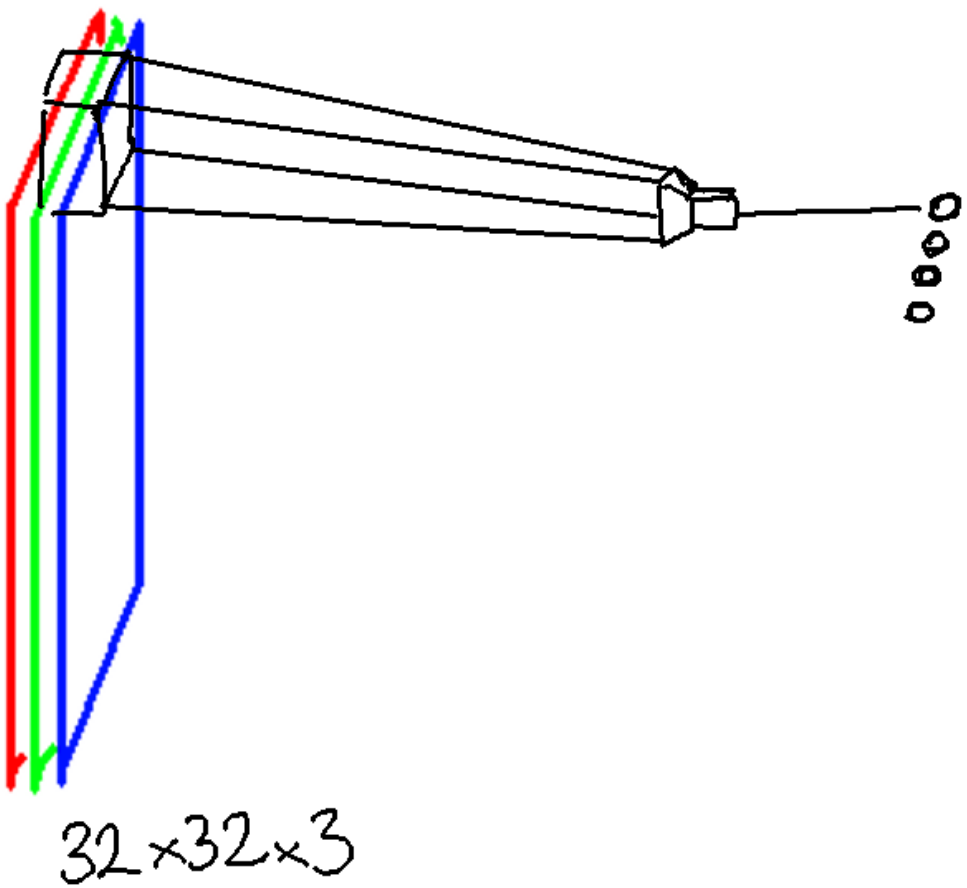
Local connectivity + convolution



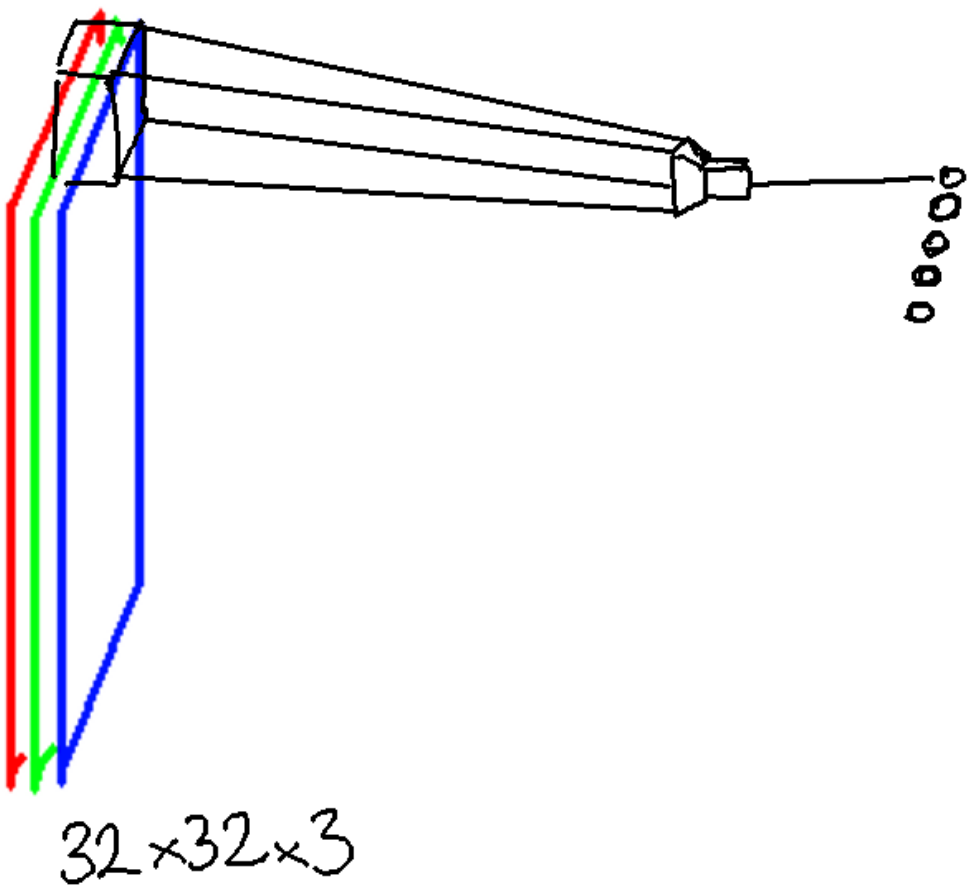
Local connectivity + convolution



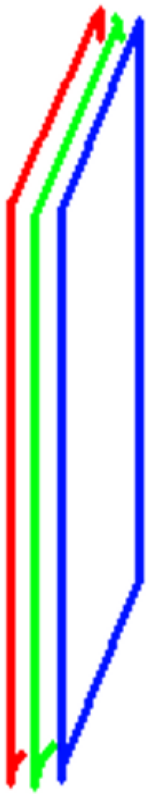
Local connectivity + convolution



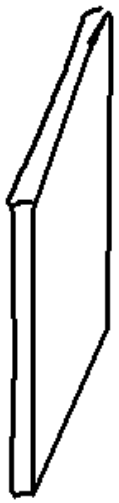
Local connectivity + convolution



One filter forms a surface

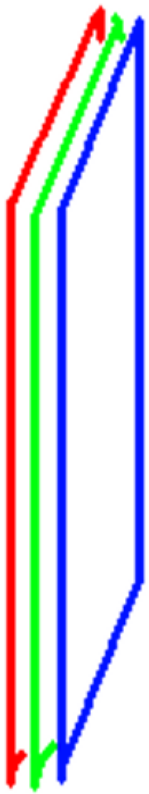


32×32×3

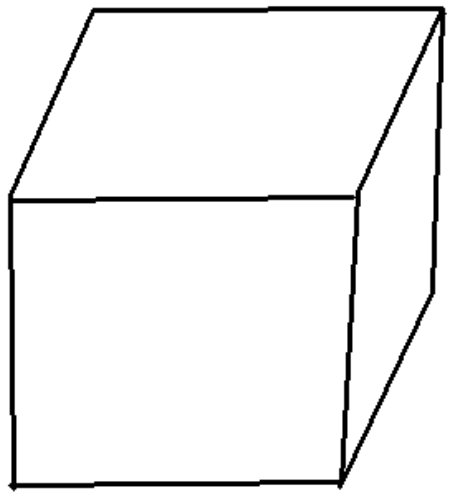


32×32×1

Many filters form a volume

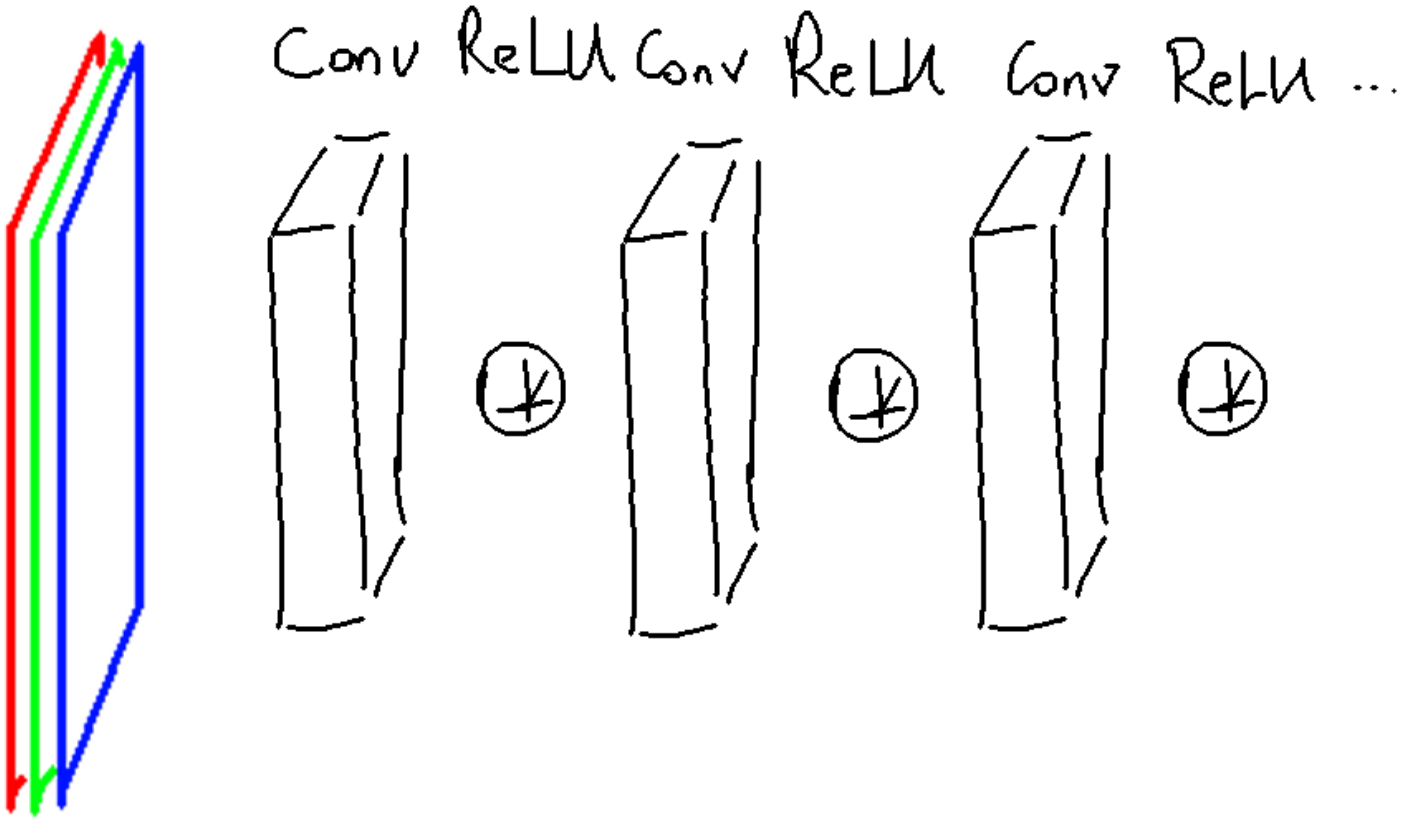


$32 \times 32 \times 3$

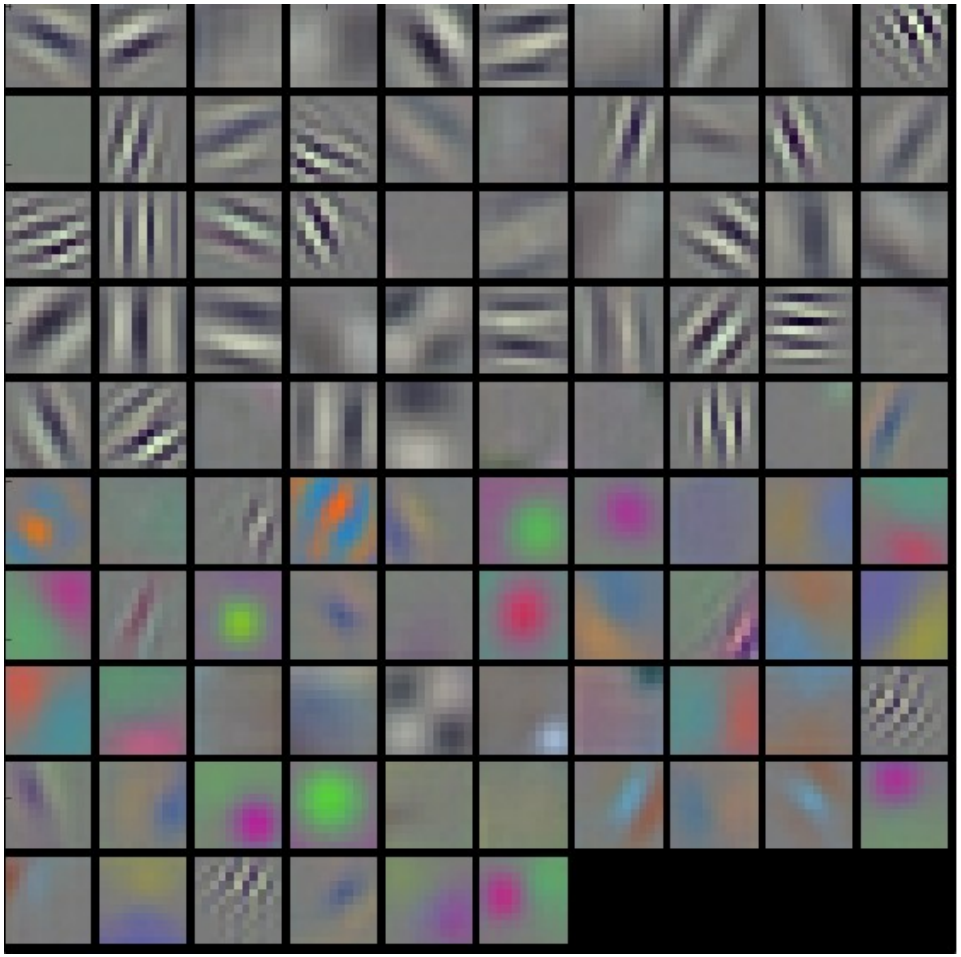


$32 \times 32 \times N_F$

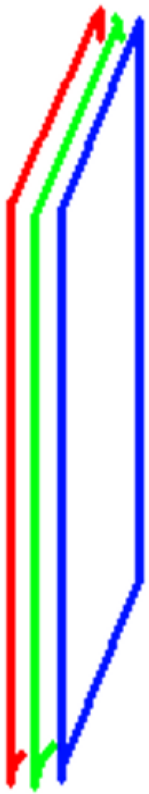
Typical processing structure



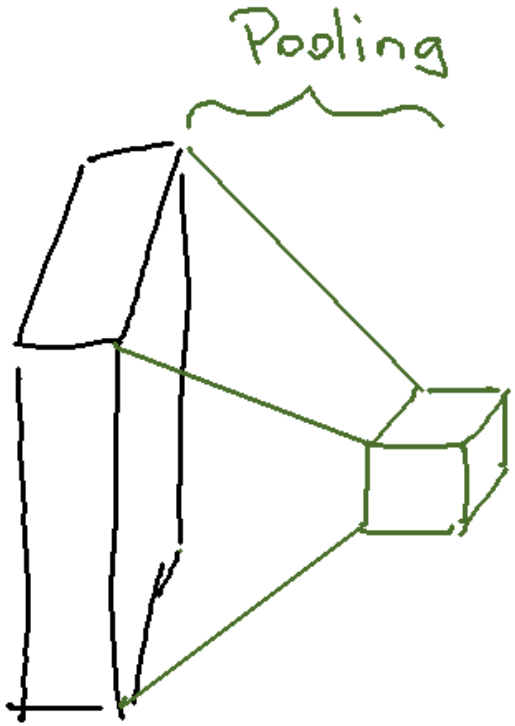
AlexNet's filters



Pooling

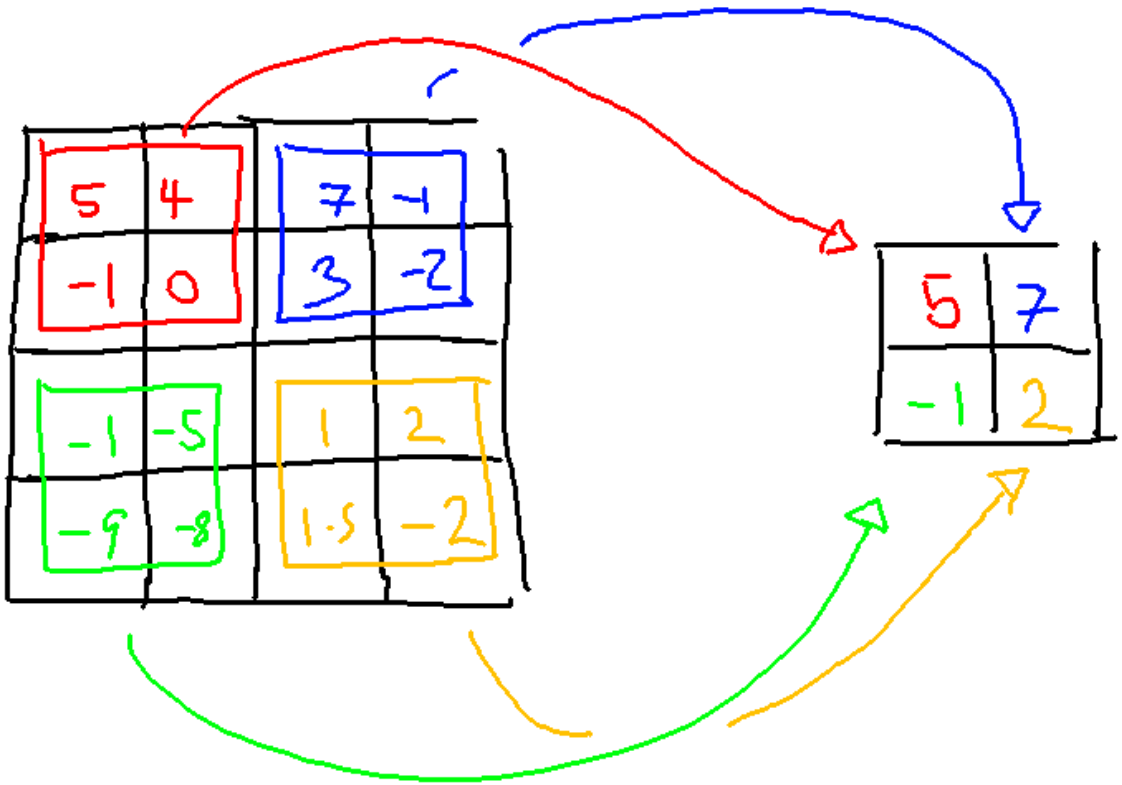


$\{Conv, ReLU\}_M$



$\{X, Y, 3\} \rightarrow \{X, Y, K\} \rightarrow \{<X, <Y, K\}$

Pooling



Typical structure of a CNN