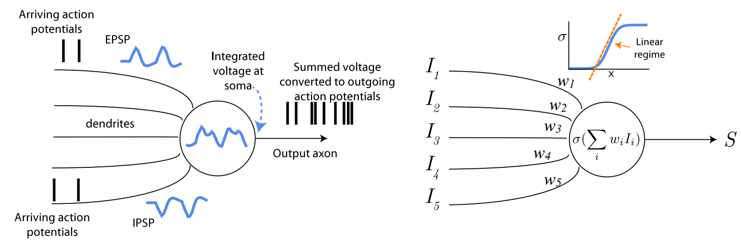
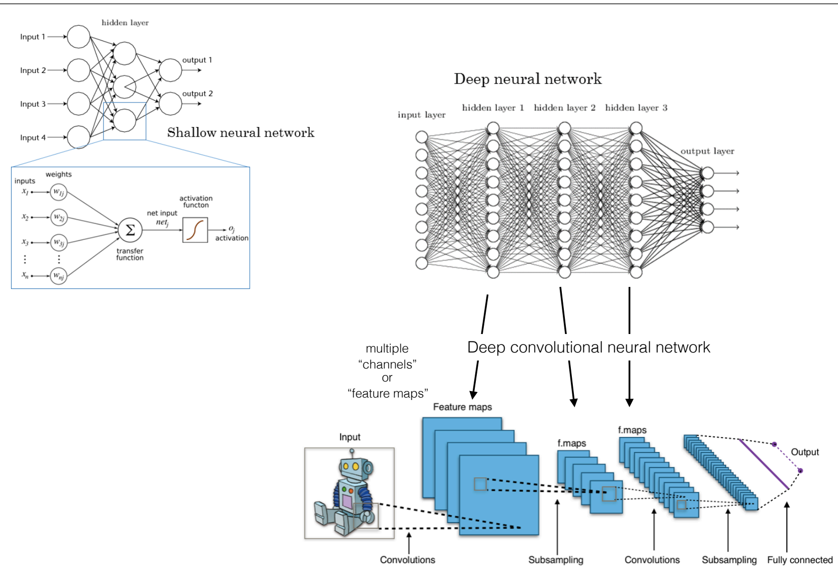
Deep learning and human vision

Mini lecture 3: review so far, and learning the weights

local building blocks

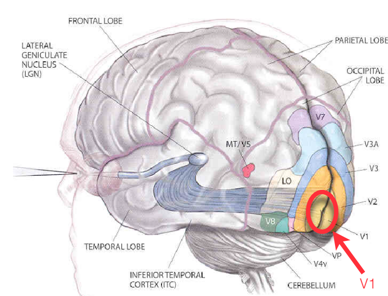


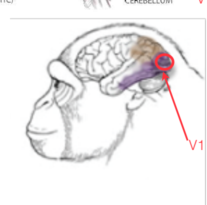
continuous valued inputs and outputs representing frequency of action potentials (spikes"

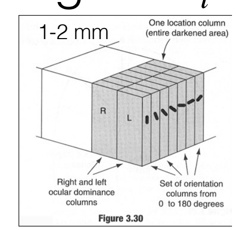


https://en.wikipedia.org/wiki/Convolutional_neural_network

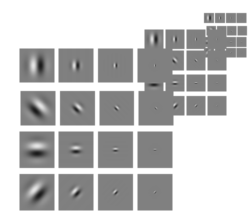
what determines the weights w_i?,







Hubel & Wiesel, 1960s

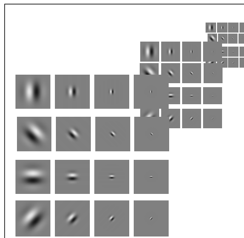


receptive fields

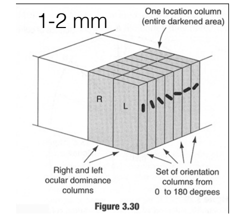
a dictionary of image features?

hand-wired
shallow models

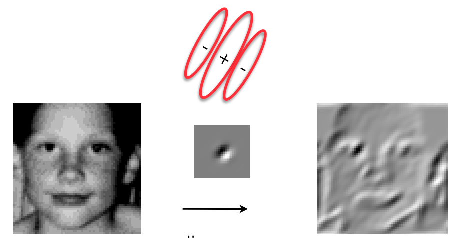
Models of weights based on a large body of empirical measurements characterizing the spatial filtering properties of neurons particularly in V1



linear spatial filter V1 model



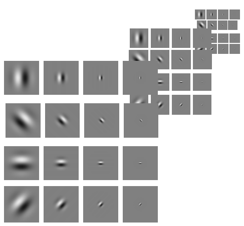
Receptive field



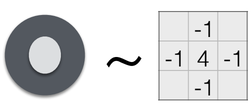
linear

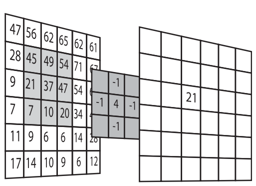
convolution

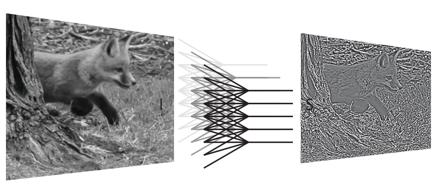
just one of a set of spatial frequency “channels”



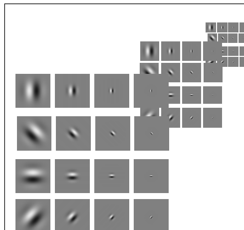
What is convolution?





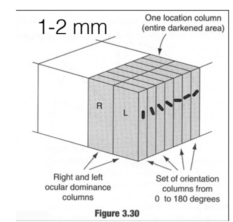


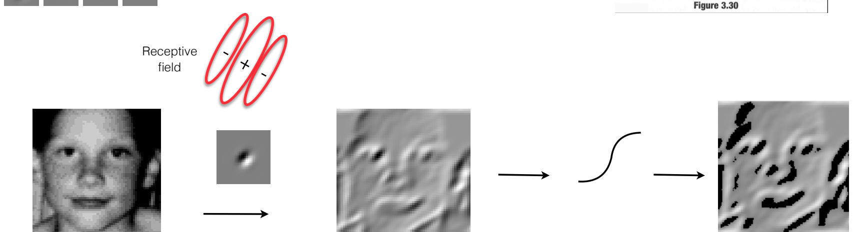
center-surround filter replaces each output cell with a weighted sum of the input levels



standard V1 model of “simple” cells:

including point-wise non-linearity

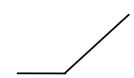




linear

point-wise non-linearity

convolution



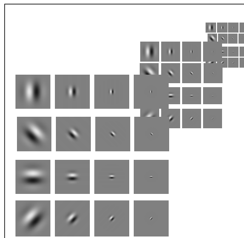
reLU

Other non-linearities to model the output of an otherwise linear model of a neuron?

Table 1: Potential Computations That Can Be Performed by the Neural Circuits in Figure 1 at Their Steady States.

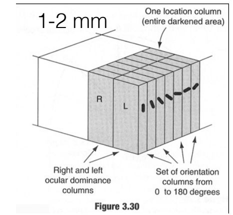
Operation	(Steady-State) Output		
Canonical	$y = \frac{\sum_{i=1}^{n} w_i \, x_i^{\,p}}{k + \left(\sum_{i=1}^{n} x_i^{\,q}\right)^{r}}$	(2.1)	generalized model of divisive normalization
Energy model	$y = \sum_{i=1}^{2} x_i^{\,2}$	(2.2)	
Sigmoid-like	$y = \frac{\sum_{i=1}^{n} x_i^{\,2}}{k + \sum_{i=1}^{n} x_i^{\,2}}$	(2.3)	
Gaussian-like	$y = \frac{\sum_{i=1}^{n} w_i \, x_i}{k + \sum_{i=1}^{n} x_i^{\,2}}$	(2.4)	
Max-like	$y = \frac{\sum_{i=1}^{n} x_i^{\,3}}{k + \sum_{i=1}^{n} x_i^{\,2}}$	(2.5)	

Kouh, M., & Poggio, T. (2008). A canonical neural circuit for cortical nonlinear operations. Neural Computation, 20(6), 1427–1451.

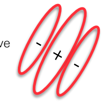


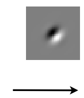
V1 model of “simple” cells:

including divisive normalization



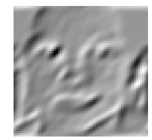
Receptive field

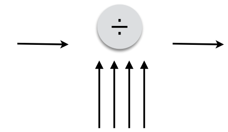




linear

convolution

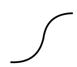




inputs from other nearby neurons—divisive normalization

an example of what DiCarlo et al. called a “normalized LN” mode, or NLN

Table 1: Potential Computations That Can Be Performed by the Neural Circuits in Figure 1 at Their Steady States.

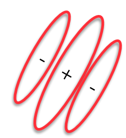
Operation	(Steady-State) Output		
Canonical	$y = \frac{\sum_{i=1}^{n} w_i \, x_i^{\,p}}{k + \left(\sum_{i=1}^{n} x_i^{\,q}\right)^{r}}$	(2.1)	generalized model of divisive normalization
Energy model	$y = \sum_{i=1}^{2} x_i^{\,2}$	(2.2)	quadrature phase pairs
Sigmoid-like	$y = \frac{\sum_{i=1}^{n} x_i^{\,2}}{k + \sum_{i=1}^{n} x_i^{\,2}}$	(2.3)	
Gaussian-like	$y = \frac{\sum_{i=1}^{n} w_i \, x_i}{k + \sum_{i=1}^{n} x_i^{\,2}}$	(2.4)	
Max-like	$y = \frac{\sum_{i=1}^{n} x_i^{\,3}}{k + \sum_{i=1}^{n} x_i^{\,2}}$	(2.5)	

Kouh, M., & Poggio, T. (2008). A canonical neural circuit for cortical nonlinear operations. Neural Computation, 20(6), 1427–1451.

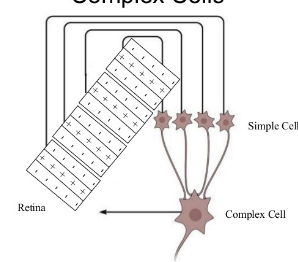
simple and complex cells in V1

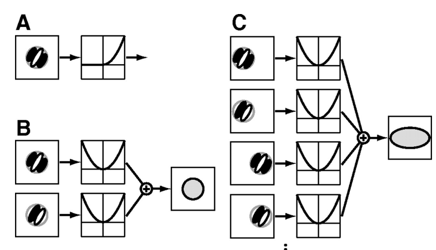
Simple Cell

receptive field



Complex Cells



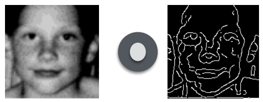


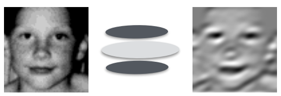
circuits for building translational invariance

shallow convolutional networks

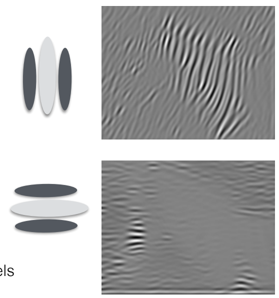
what can they do?

detect edges





detect faces?
(but not reliably)

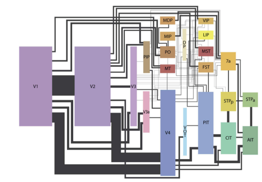


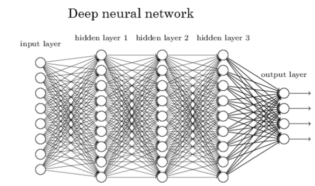
textures?
(with multiple channels
or “features”)

…but despite much research, few of these shallow networks work very well as models of human perception, except for simple stimuli and tasks

e.g. they do work well as predictors of contrast detection, discrimination of fairly large family of textures.

Deep networks
what determines the weights w_{ij} as one proceeds up levels (j) of the hierarchy?,





the tasks of vision,
e.g. “core” recognition

the regularities in images,
e.g. high correlations between nearby pixels

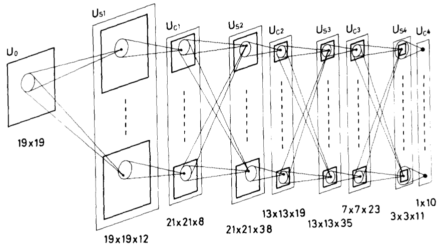
?

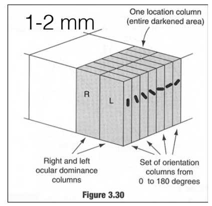
hierarchical models for feature extraction given task constraints, e.g. core recognition

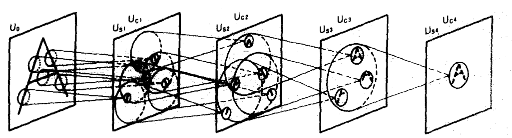
- Local features progressively grouped into more structured representations
    - edges => contours => fragments => parts => objects
- Selectivity/invariance trade-off
    - Increased selectivity for object/pattern type
    - Decreased sensitivity to view-dependent variations of translation, scale and illumination

deep solutions to the translational invariance problem

Fukushima 1988



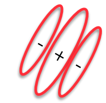




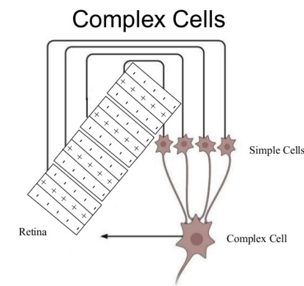
Fukushima, K. (1988). Neocognitron - a Hierarchical Neural Network Capable of Visual-Pattern Recognition. *Neural Networks, 1*(2), 119–130.

simple and complex cells in V1

simple cell receptive field



"AND-ing"



one model illustrating local translation invariance

"OR-ing"

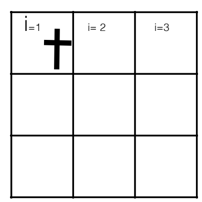
simple & complex cells in V1

- Simple cells
  - "template matching", i.e. detect conjunctions, logical "AND"
- Complex cells
  - insensitivity to small changes in position, detect disjunctions, logical "OR"
- Recognition as the hierarchical detection of "disjunctions of conjunctions"

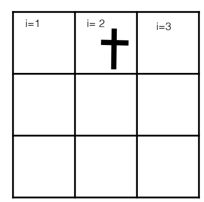
Recognize the letter "t"

"t" is represented by the conjunction of a vertical and horizontal bar

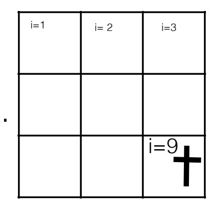
| AND — = t



OR

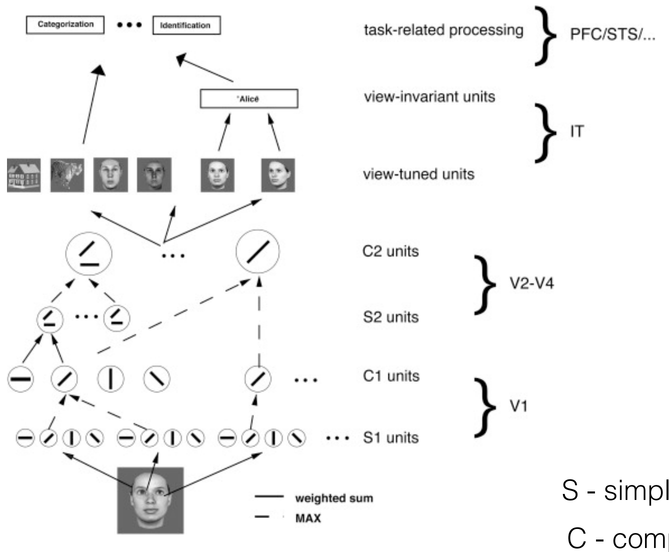


OR ...



which can occur at any one of many locations i

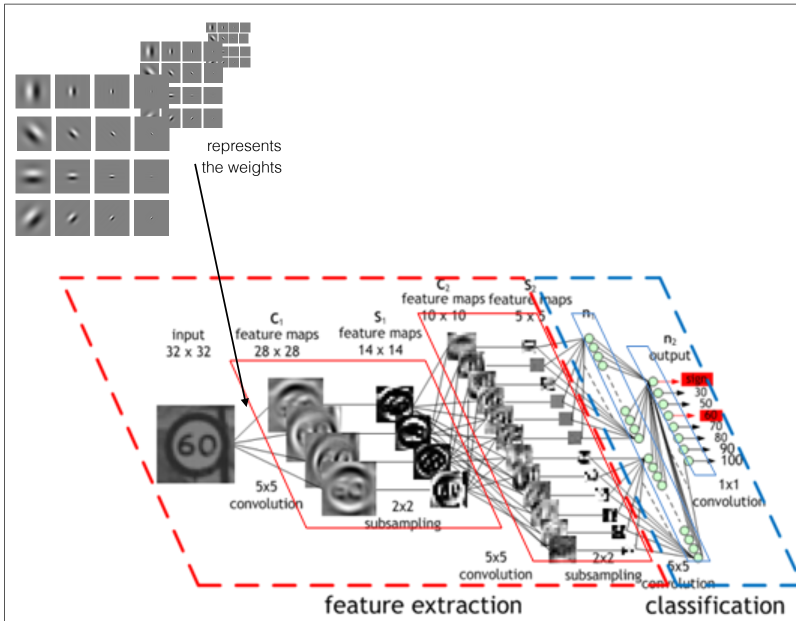
"t": $h_1 \;\&\&\; v_1 \;||\; h_2 \;\&\&\; v_2 \;||\; h_3 \;\&\&\; v_3 \ldots$



S - simple cell like

C - complex cell like

Riesenhuber & Poggio, 1999



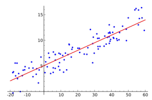
preview of the following weeks

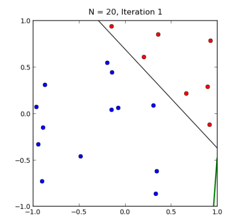
learning the weights?

- instead of “hand wiring”, can the weights—the parameters required for subsequent inferences—be learned?
  - “machine learning”
- two main approaches
  - unsupervised learning
  - supervised learning

shallow supervised learning

- Lots of algorithms for regression and classification
  - Linear regression
  - Simple perceptron
  - Fisher linear discriminant
  - Support vector machines
    - linear & non-linear



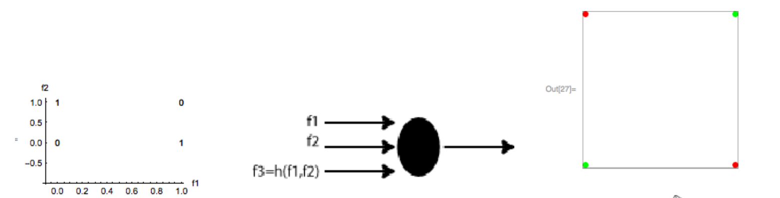


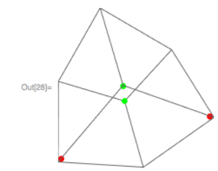
https://datasciencelab.wordpress.com/2014/01/10/machine-learning-classics-the-perceptron/

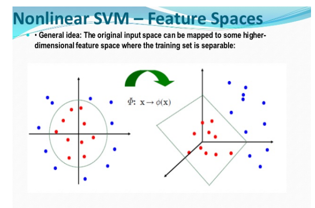
online training: iterate through the training pairs

non-linear support vector machines

Use a “kernel” to transform the input data into a higher dimensional space where the classes may be linearly separable.







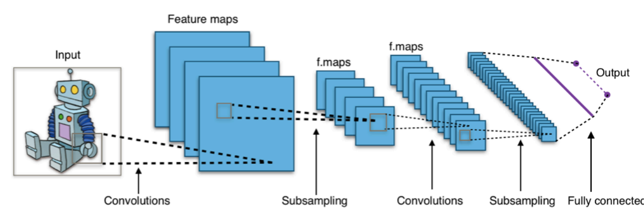
http://www.slideshare.net/m80m07/support-vector-machine-15142742

general problems

- non-linear classification requires much trial and error pre-processing to find feature representations that enable linear separation of the classes
- over-fitting
  - too many free parameters relative to the number of training examples. Good fits to what has been learned, but poor prediction given new sample inputs
  - the bias/variance trade-off
  - solutions?
    - keep the model simple—e.g. “shallow”, fewer parameters to learn, but bigger errors
    - go for more parameters but then need more data AND better algorithms

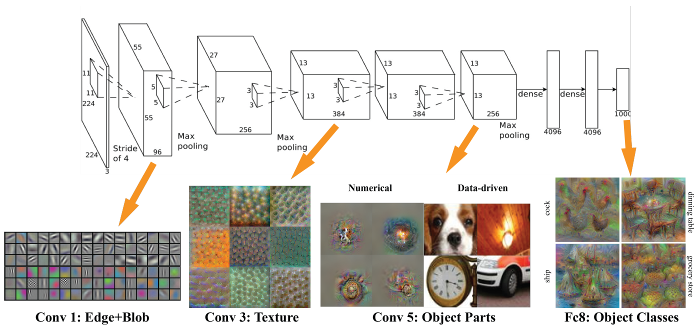
deep supervised learning

training set: annotated datasets, with error back-propagation learning



deep supervised learning

training set: annotated datasets, with error back-propagation learning



http://www.slideshare.net/ckmarkohchang/applied-deep-learning-1103-convolutional-neural-networks

next week

- Epshtein, B., Lifshitz, I., & Ullman, S. (2008). Image interpretation by a single bottom-up top-down cycle. Proceedings of the National Academy of Sciences, 105(38), 14298–14303. http://doi.org/10.1073/pnas.0800968105
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, 1097–1105.