Presentation on Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

– Ning Wang, 502B, 02/06/2025

Outline:

- Background on neural networks
- Figures from Yamins et al., 2016
- Discussion

Outline:

- **Background on neural networks**
- Figures from Yamins et al., 2016
- Discussion

Convolutional Neural Network, not ~~Cable News Network~~

- Not this type of CNN —>
- This type of CNN:

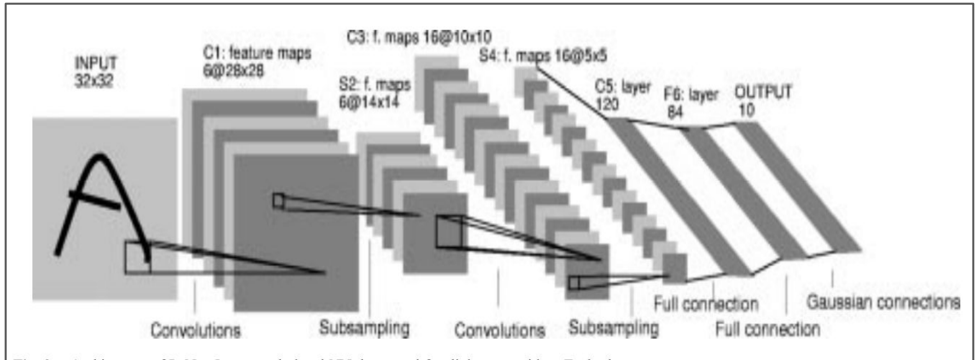
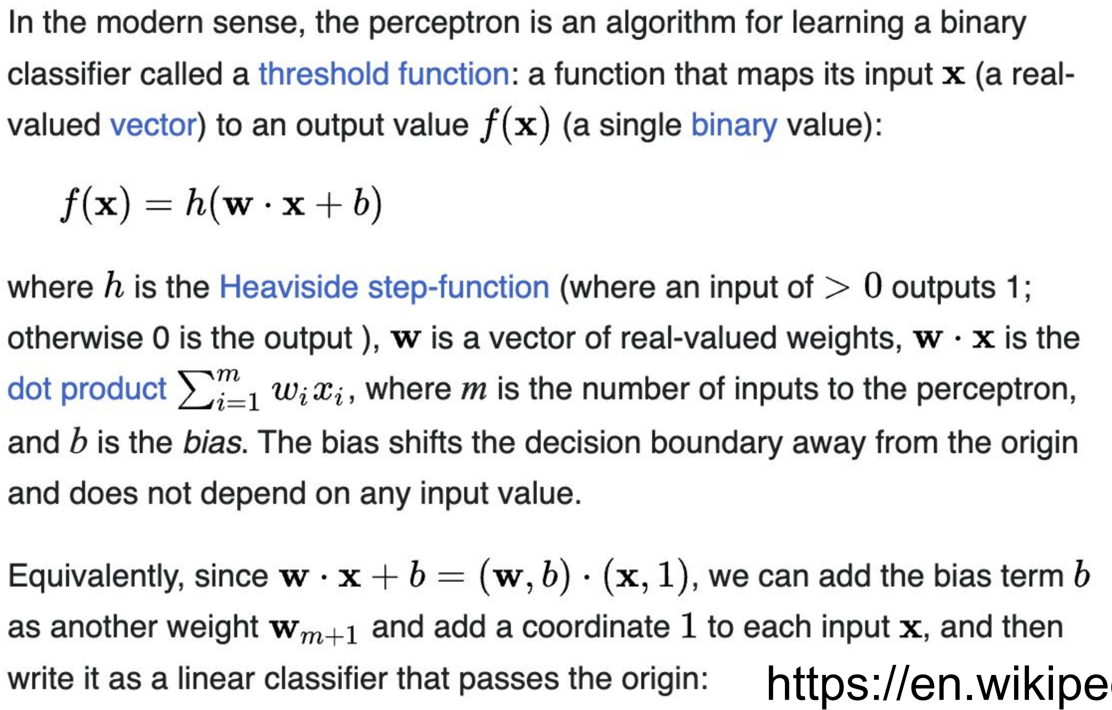


Fig. 2. Architecture of LeNet-5, a convolutional NN, here used for digits recognition. Each plane is a feature map, i.e., a set of units whose weights are constrained to be identical.

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

Perceptron is a "GLM" and a one-layer neural network

- Now implemented with code instead of a physical machine

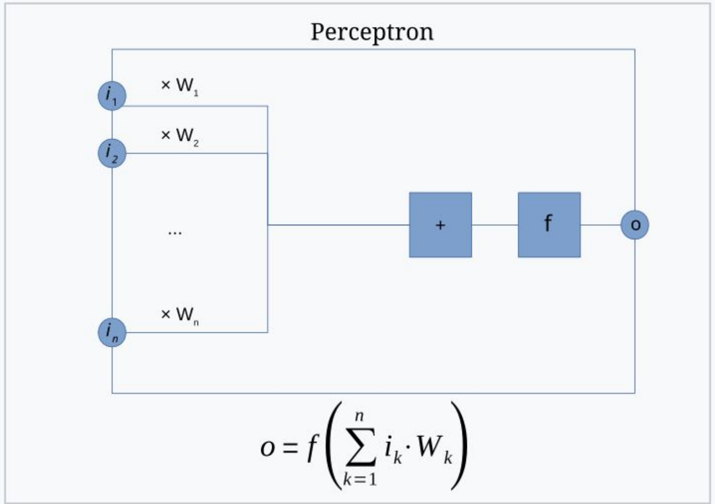


In the modern sense, the perceptron is an algorithm for learning a binary classifier called a threshold function: a function that maps its input $\mathbf{x}$ (a real-valued vector) to an output value $f(\mathbf{x})$ (a single binary value):

$$f(\mathbf{x}) = h(\mathbf{w} \cdot \mathbf{x} + b)$$

where h is the Heaviside step-function (where an input of > 0 outputs 1; otherwise 0 is the output), $\mathbf{w}$ is a vector of real-valued weights, $\mathbf{w} \cdot \mathbf{x}$ is the dot product $\sum_{i=1}^{m} w_i x_i$, where m is the number of inputs to the perceptron, and b is the *bias*. The bias shifts the decision boundary away from the origin and does not depend on any input value.

Equivalently, since $\mathbf{w} \cdot \mathbf{x} + b = (\mathbf{w}, b) \cdot (\mathbf{x}, 1)$, we can add the bias term b as another weight $\mathbf{w}_{m+1}$ and add a coordinate 1 to each input $\mathbf{x}$, and then write it as a linear classifier that passes the origin:

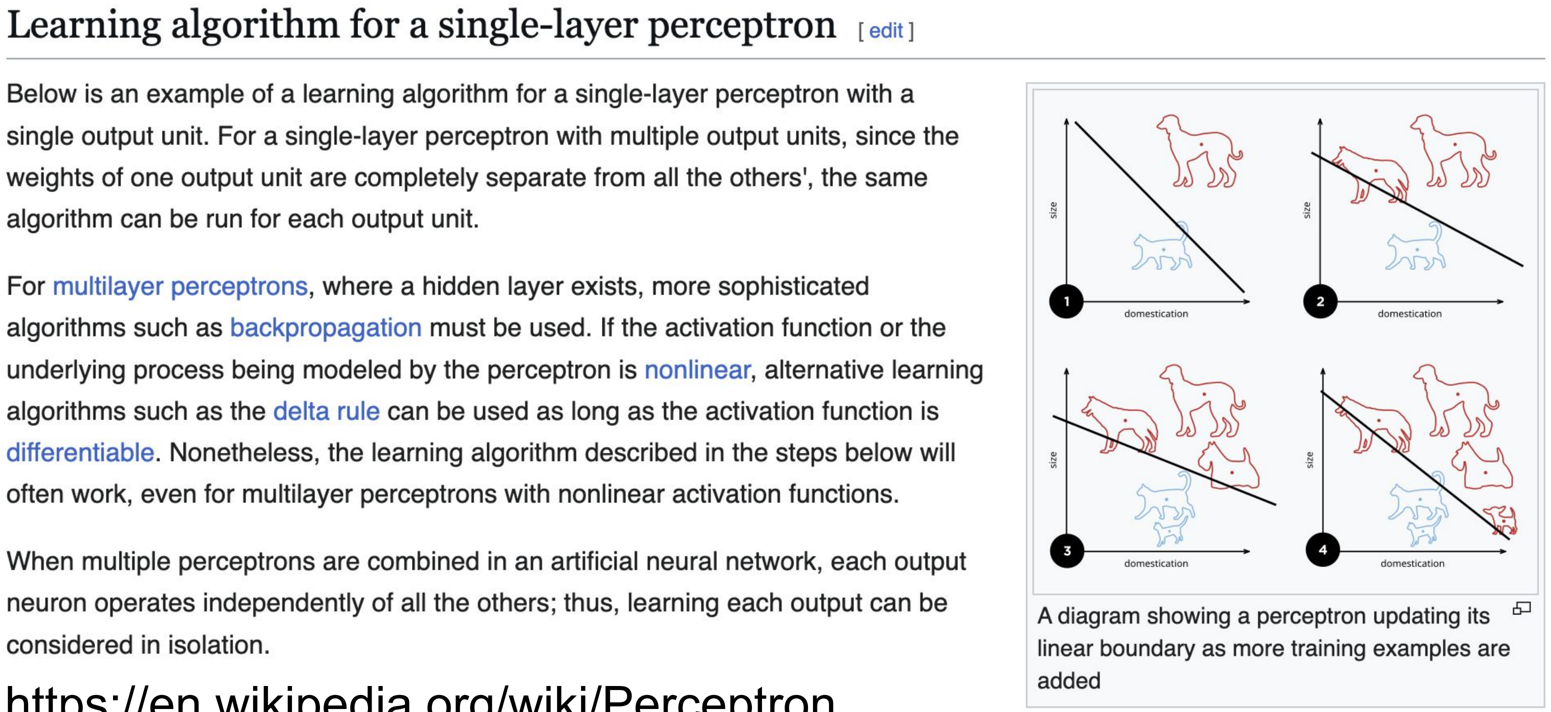


The appropriate weights are applied to the inputs, and the resulting weighted sum passed to a function that produces the output o.

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

Perceptron learns from making mistakes. No pain = no gain

- TLDR: update weights based on misclassified data

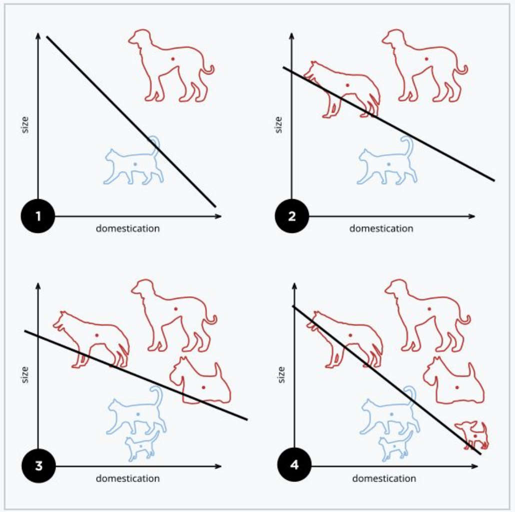


Learning algorithm for a single-layer perceptron [edit]

Below is an example of a learning algorithm for a single-layer perceptron with a single output unit. For a single-layer perceptron with multiple output units, since the weights of one output unit are completely separate from all the others', the same algorithm can be run for each output unit.

For multilayer perceptrons, where a hidden layer exists, more sophisticated algorithms such as backpropagation must be used. If the activation function or the underlying process being modeled by the perceptron is nonlinear, alternative learning algorithms such as the delta rule can be used as long as the activation function is differentiable. Nonetheless, the learning algorithm described in the steps below will often work, even for multilayer perceptrons with nonlinear activation functions.

When multiple perceptrons are combined in an artificial neural network, each output neuron operates independently of all the others; thus, learning each output can be considered in isolation.

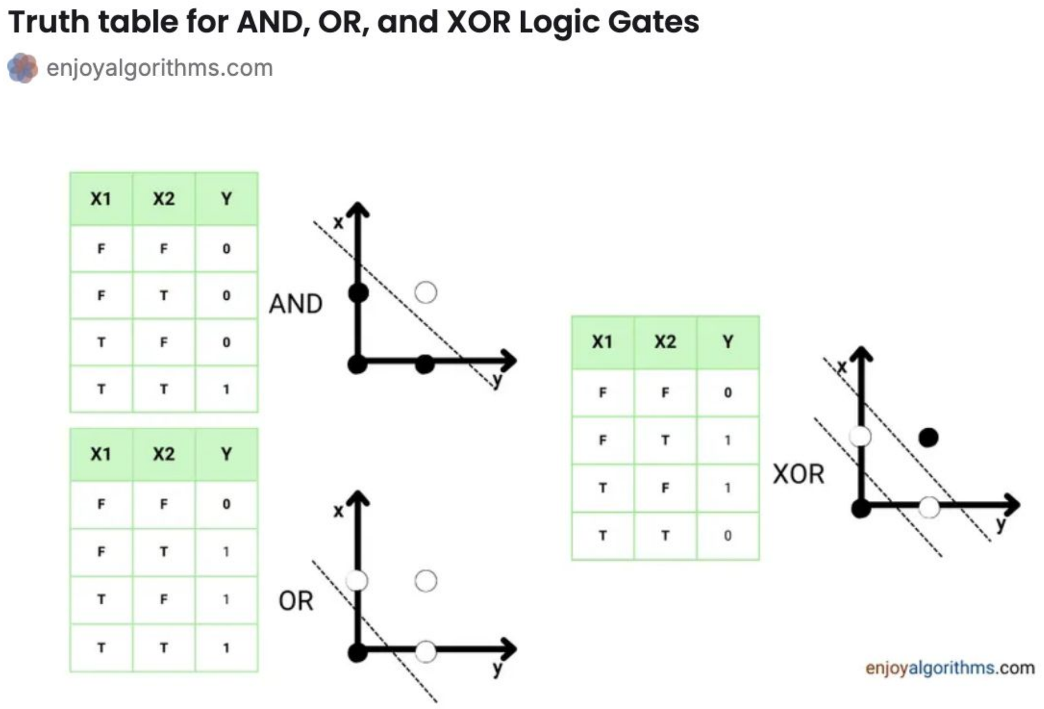


A diagram showing a perceptron updating its linear boundary as more training examples are added

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

Perceptron cannot learn "simple" operations like XOR

- Requires having linearly separable data to "learn" correctly
- Multiple lines needed for XOR
- Draw a really long line?
- Give up?


Truth table for AND, OR, and XOR Logic Gates

enjoyalgorithms.com

X1	X2	Y
F	F	0
F	T	0
T	F	0
T	T	1

X1	X2	Y
F	F	0
F	T	1
T	F	1
T	T	1

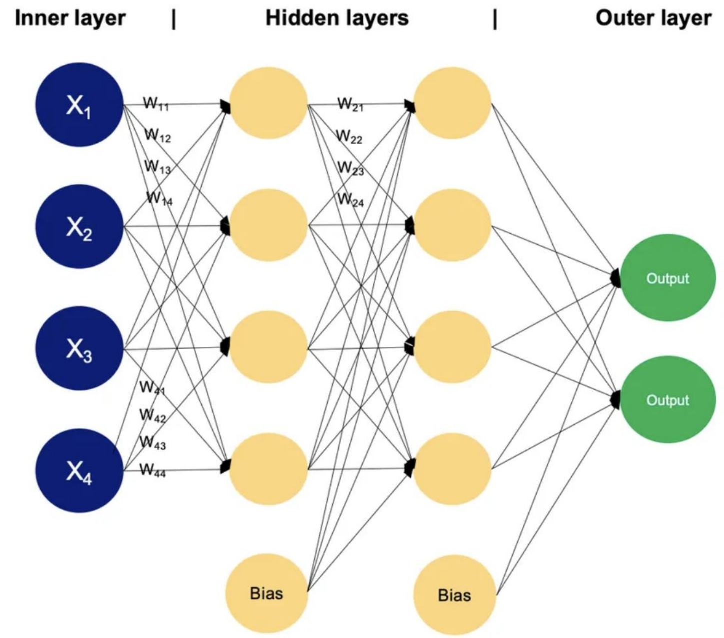
X1	X2	Y
F	F	0
F	T	1
T	F	1
T	T	0

Feed-forward neural network is a multi-layered perceptron

- Each hidden node ~= 1 perceptron
- Repeat across many hidden layers
- Retrieve one or more output nodes
- Can learn more complex patterns

Example of a MLP having two hidden layers

datacamp.com



CNNs are biologically inspired machine learning

- LeNet uses convolutions to learn features from 2D input for image classification

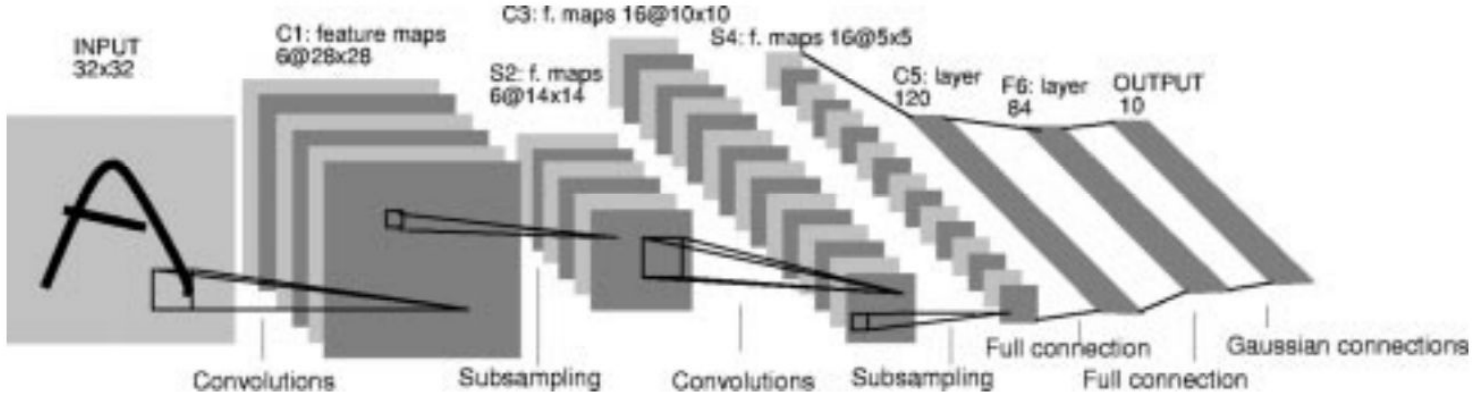


Fig. 2. Architecture of LeNet-5, a convolutional NN, here used for digits recognition. Each plane is a feature map, i.e., a set of units whose weights are constrained to be identical.

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

"Kernels" are essentially fancier, 2D dot products

- Slide kernel along an input image to combine information across shapes

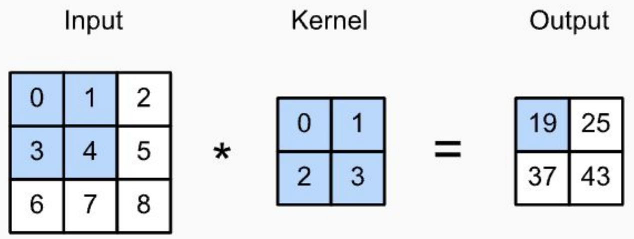


Fig. 7.2.1 Two-dimensional cross-correlation operation. The shaded portions are the first output element as well as the input and kernel tensor elements used for the output computation: $0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$.

D2L AI textbook

Padding allows "data" to change or keep same shape

- Another example, with padding on boundaries

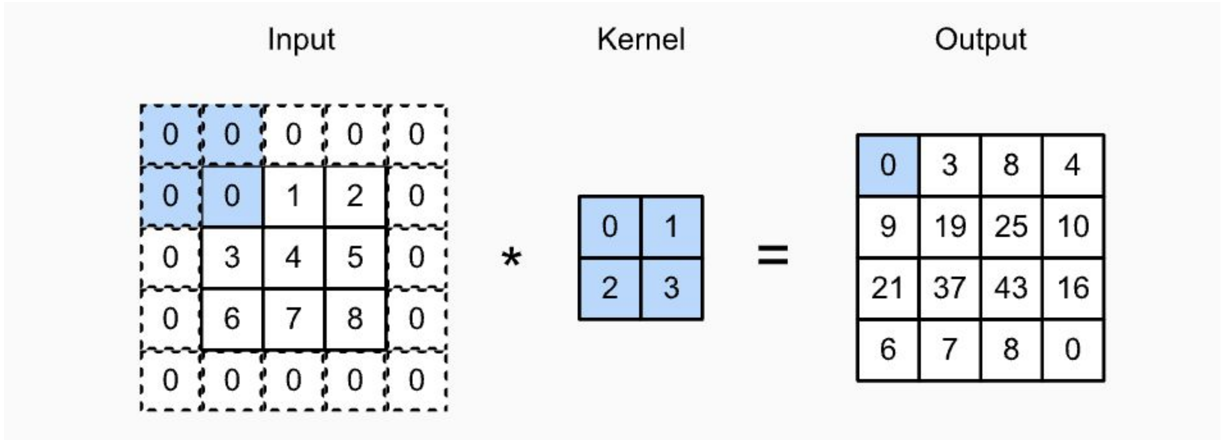


Fig. 7.3.2 Two-dimensional cross-correlation with padding.

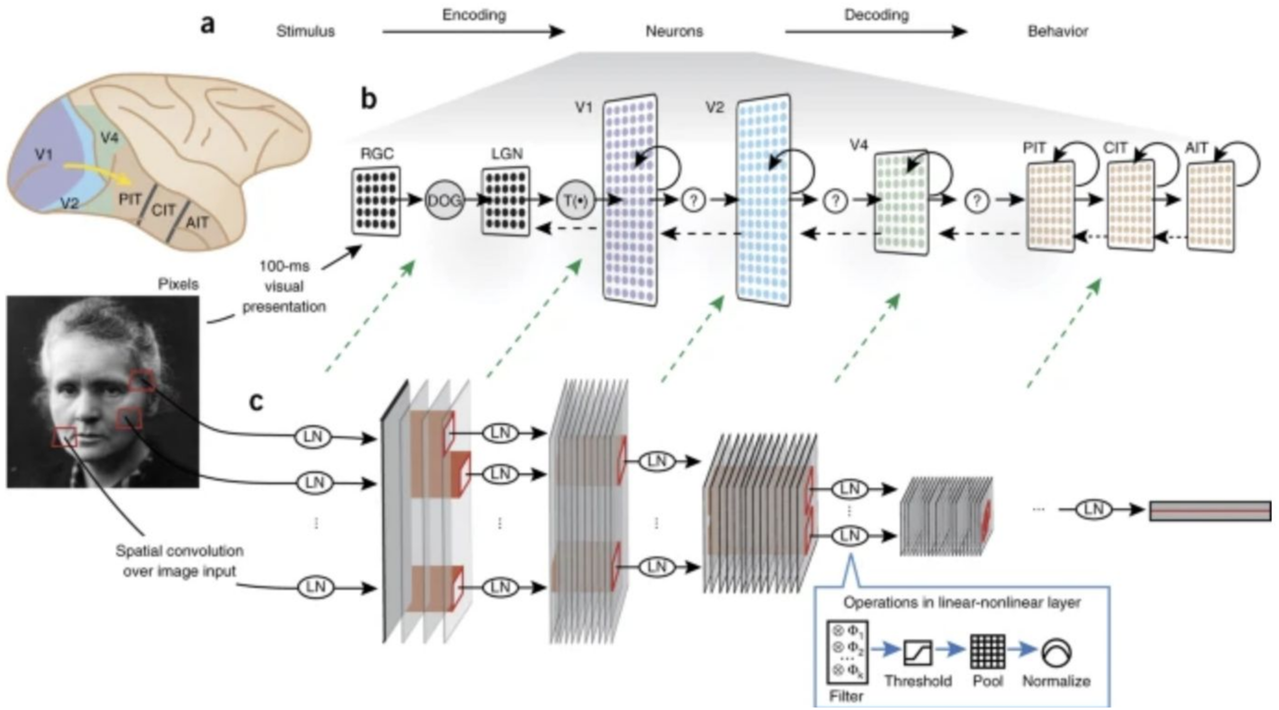
D2L AI textbook

Outline:

- Background on neural networks
- **Figures from Yamins et al., 2016**
- Discussion

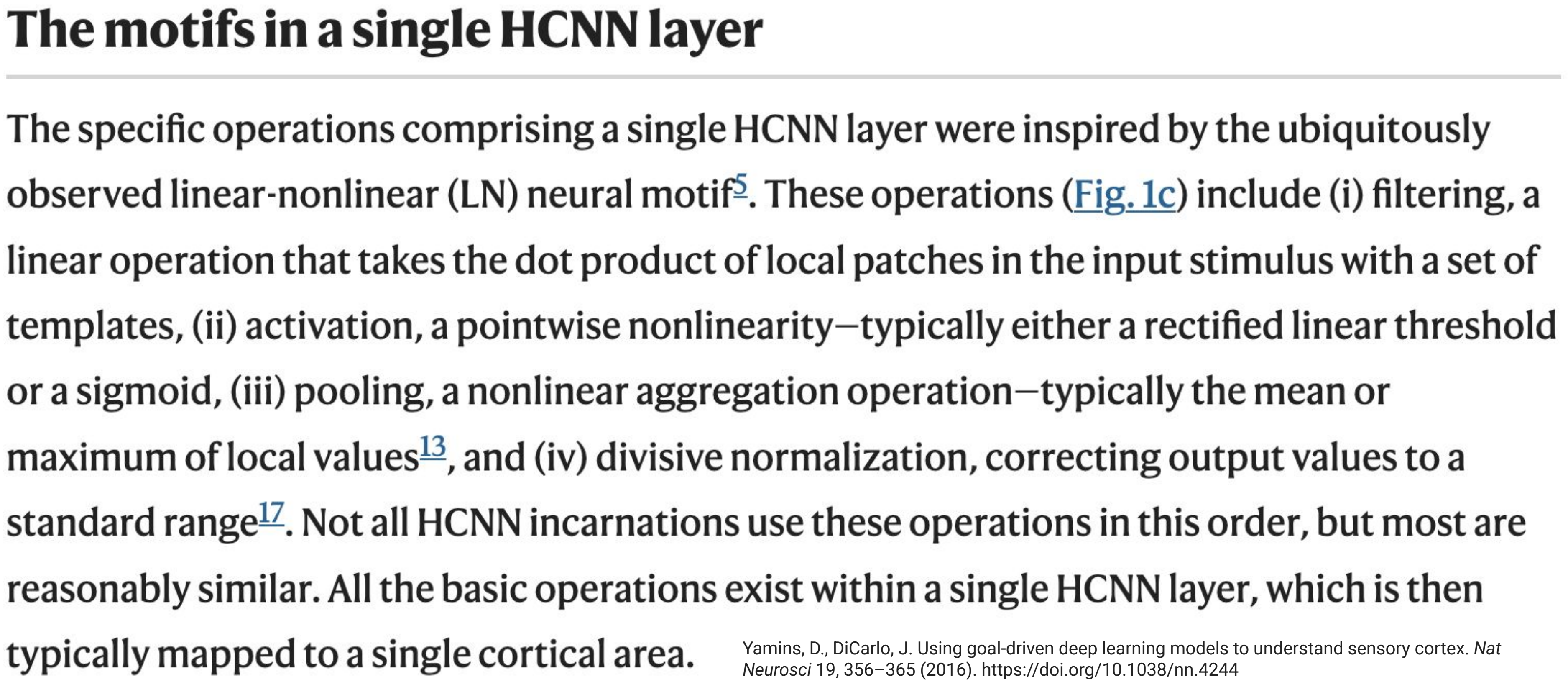
Figure 1: HCNNs are more biologically realistic CNNs

Figure 1: HCNNs as models of sensory cortex.



Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Various combinations available for each layer



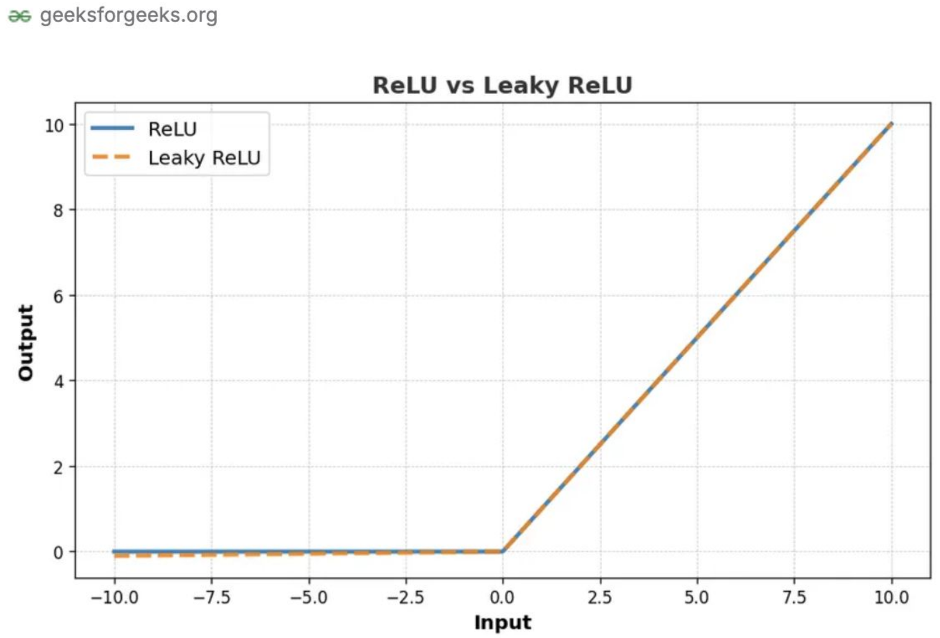
The motifs in a single HCNN layer

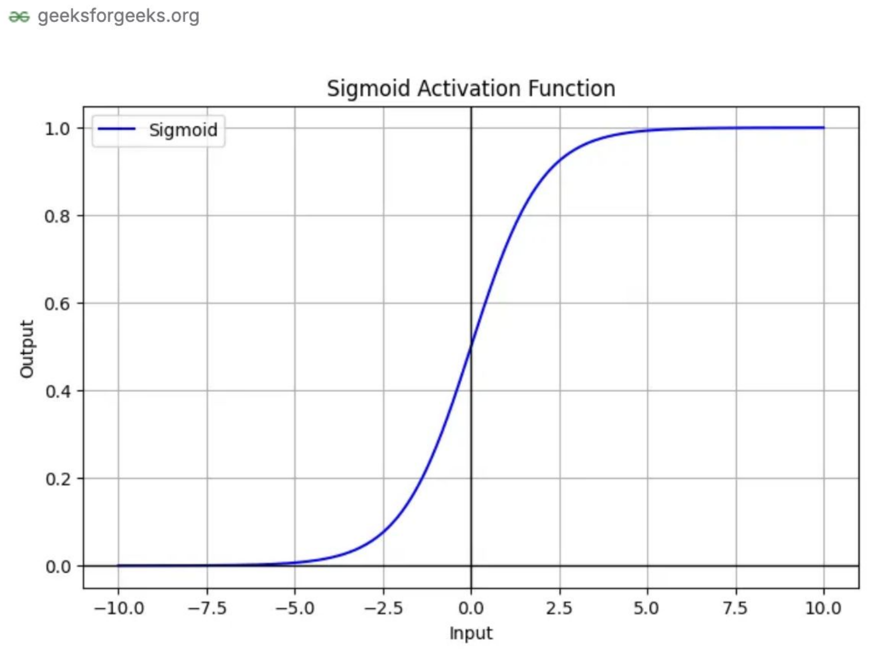
The specific operations comprising a single HCNN layer were inspired by the ubiquitously observed linear-nonlinear (LN) neural motif[5]. These operations (Fig. 1c) include (i) filtering, a linear operation that takes the dot product of local patches in the input stimulus with a set of templates, (ii) activation, a pointwise nonlinearity—typically either a rectified linear threshold or a sigmoid, (iii) pooling, a nonlinear aggregation operation—typically the mean or maximum of local values[13], and (iv) divisive normalization, correcting output values to a standard range[17]. Not all HCNN incarnations use these operations in this order, but most are reasonably similar. All the basic operations exist within a single HCNN layer, which is then typically mapped to a single cortical area.

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Activation functions enable non-linear transformations

- Relu [0, inf) and Sigmoid (0, 1) are common activation functions

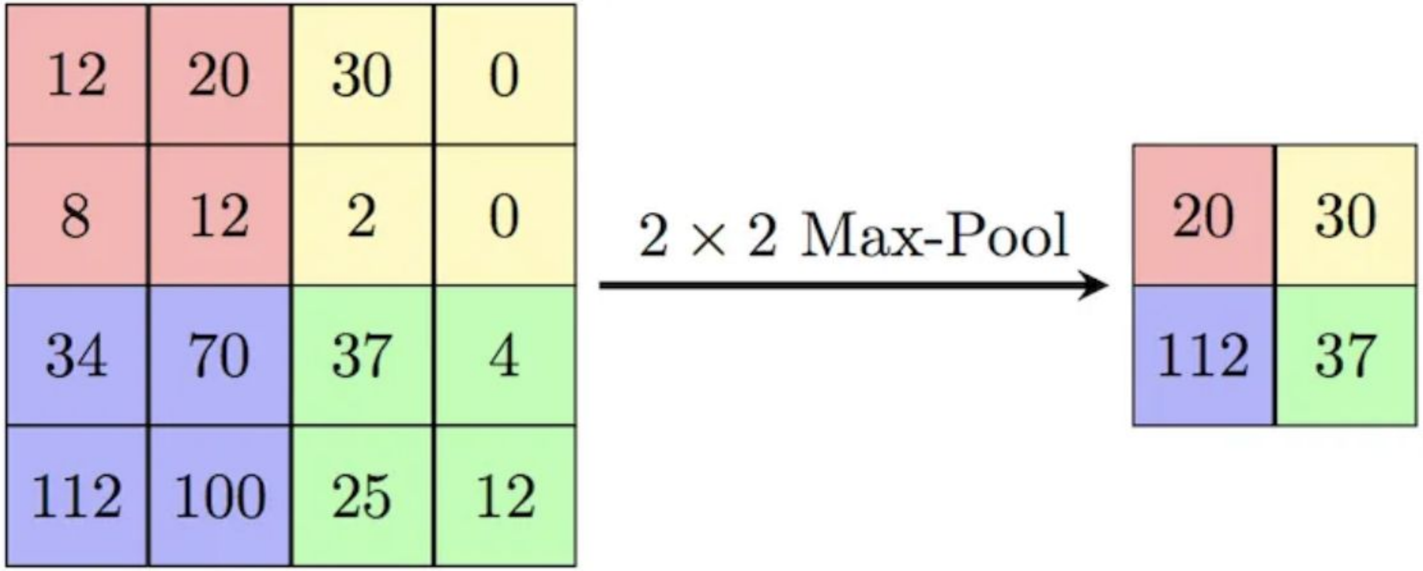




Max pool "downsamples" dimensions by some size factor

- Commonly seen in U-net architectures (e.g. used in SLEAP)

paperswithcode.com



Strides also change the "receptive field" of later layers

- Example with stride 2, as opposed to stride 1 (seen previously)

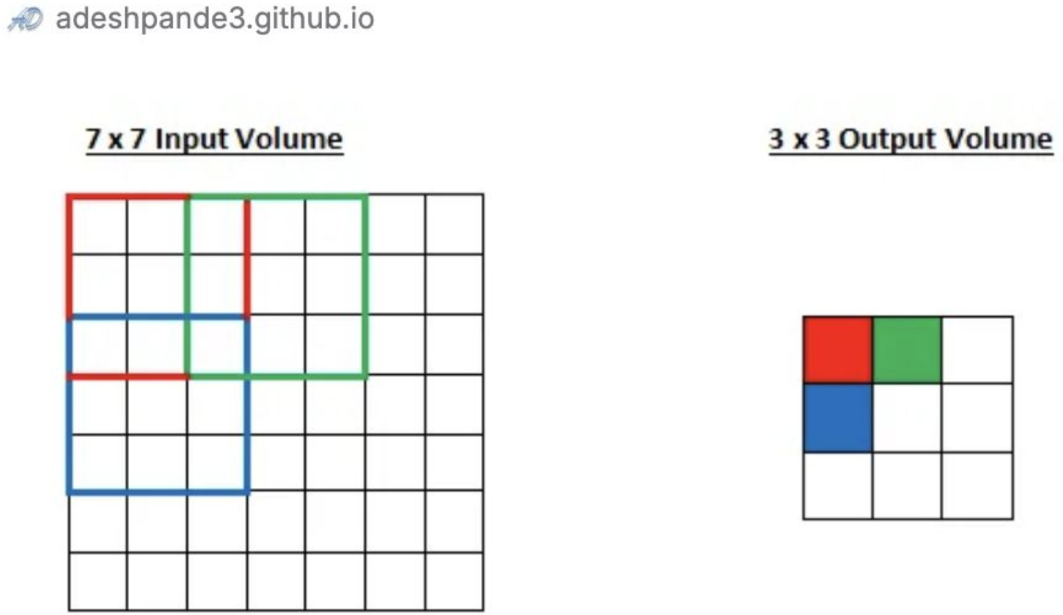
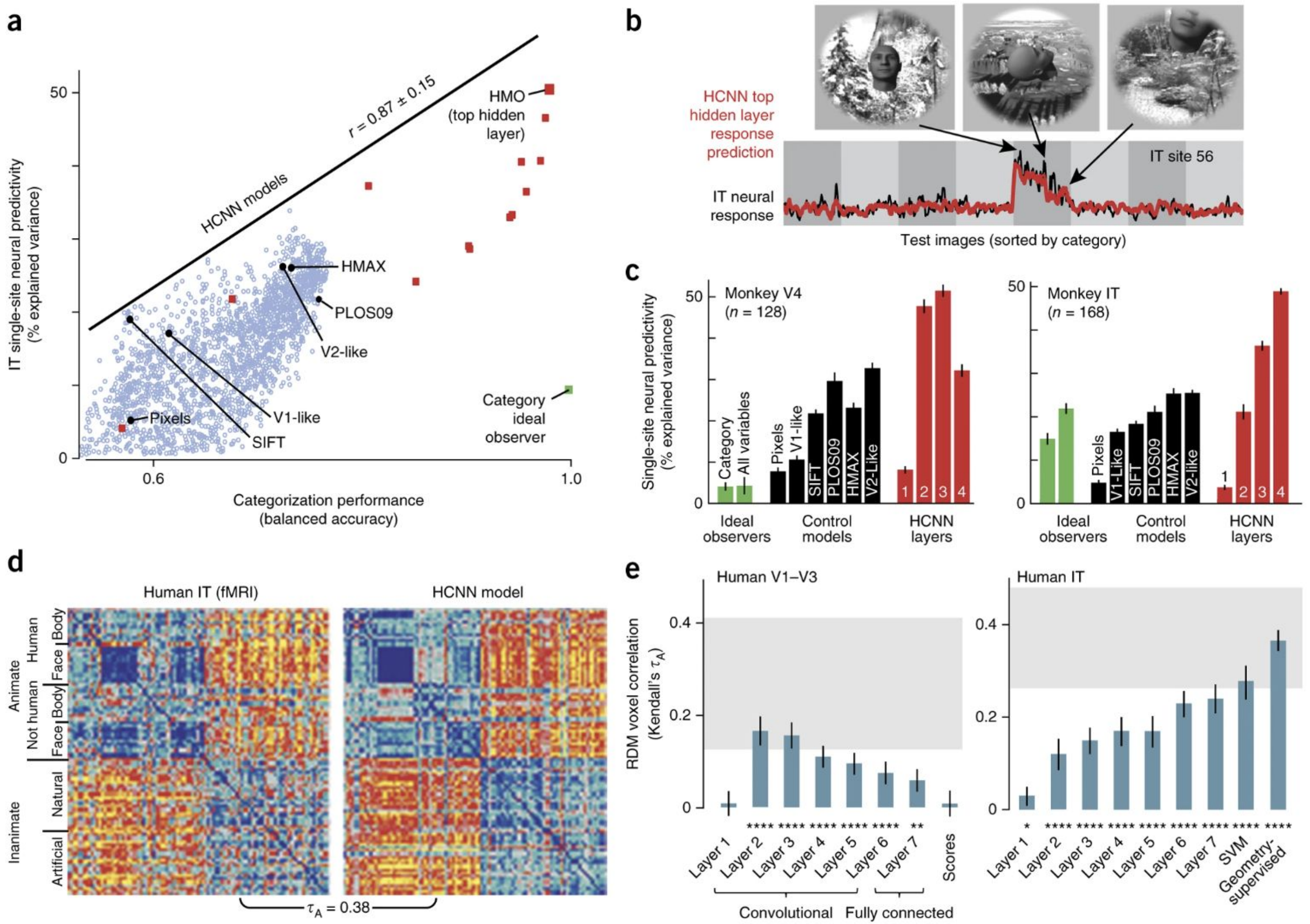


Figure 2:

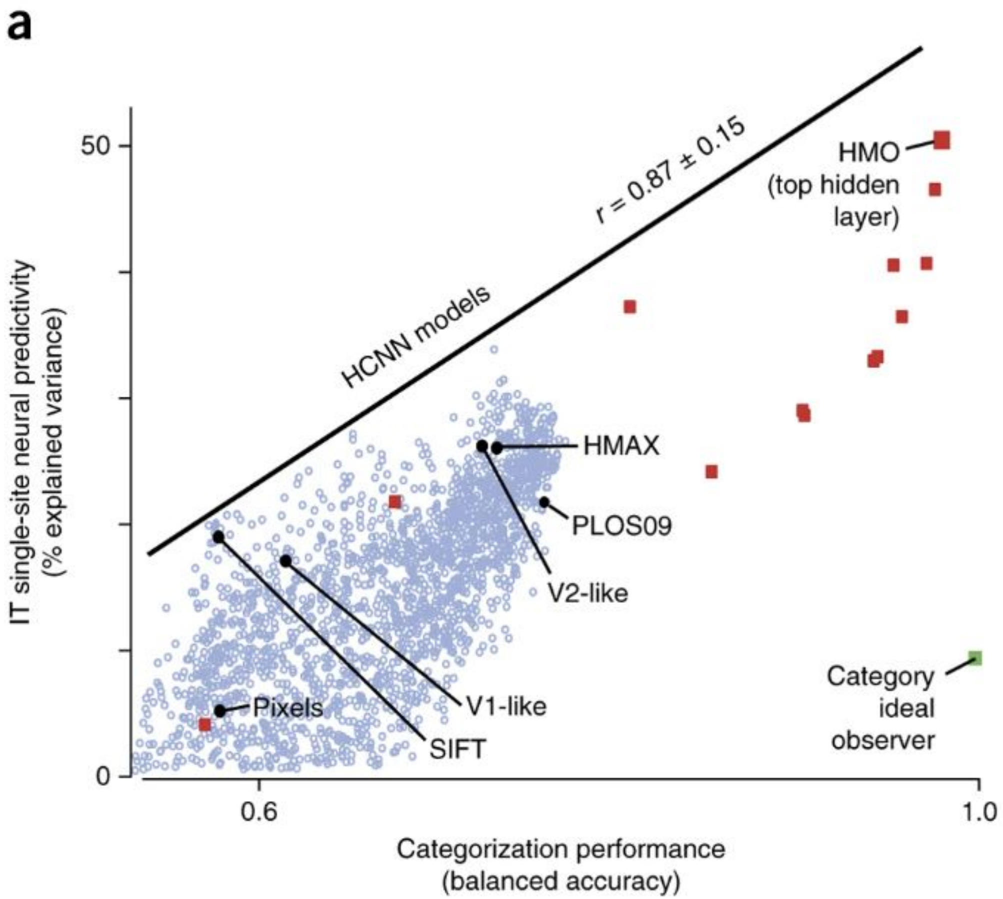
- Comparisons of HCNN spiking predictions against actual



Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Figure 2:

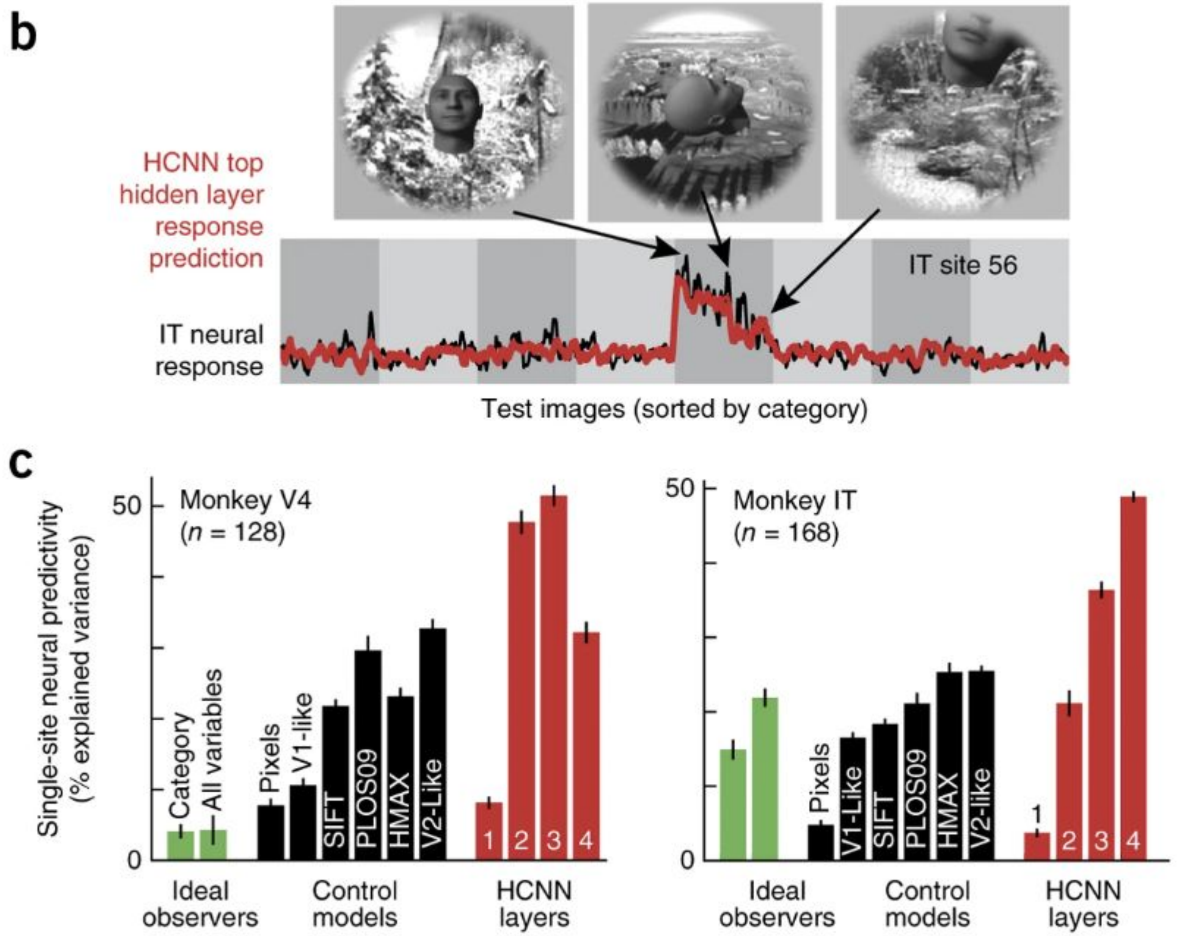
- Positive correlation between object classification and neural predictivity



Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Figure 2:

- HCNN are also predictive of neural spiking data.
- Last layer predicts IT
- 2nd to last layer predicts V4

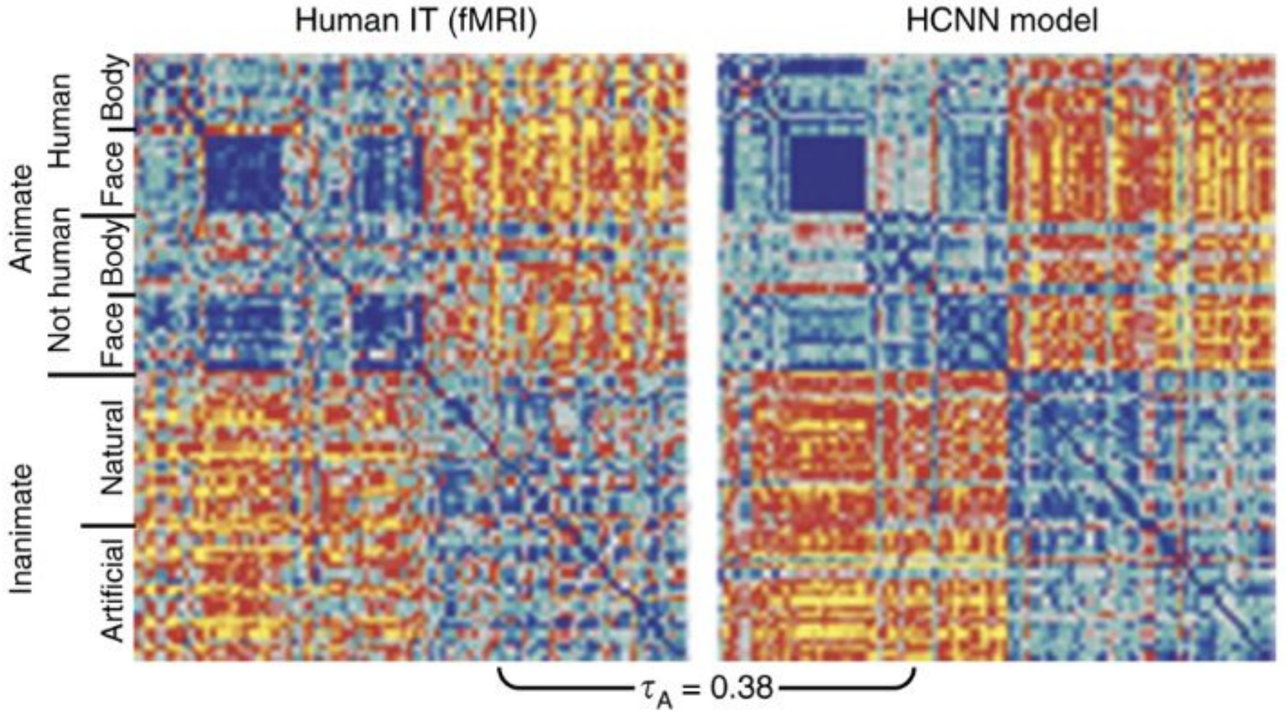


Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Figure 2:

- Also reflective of fMRI patterns

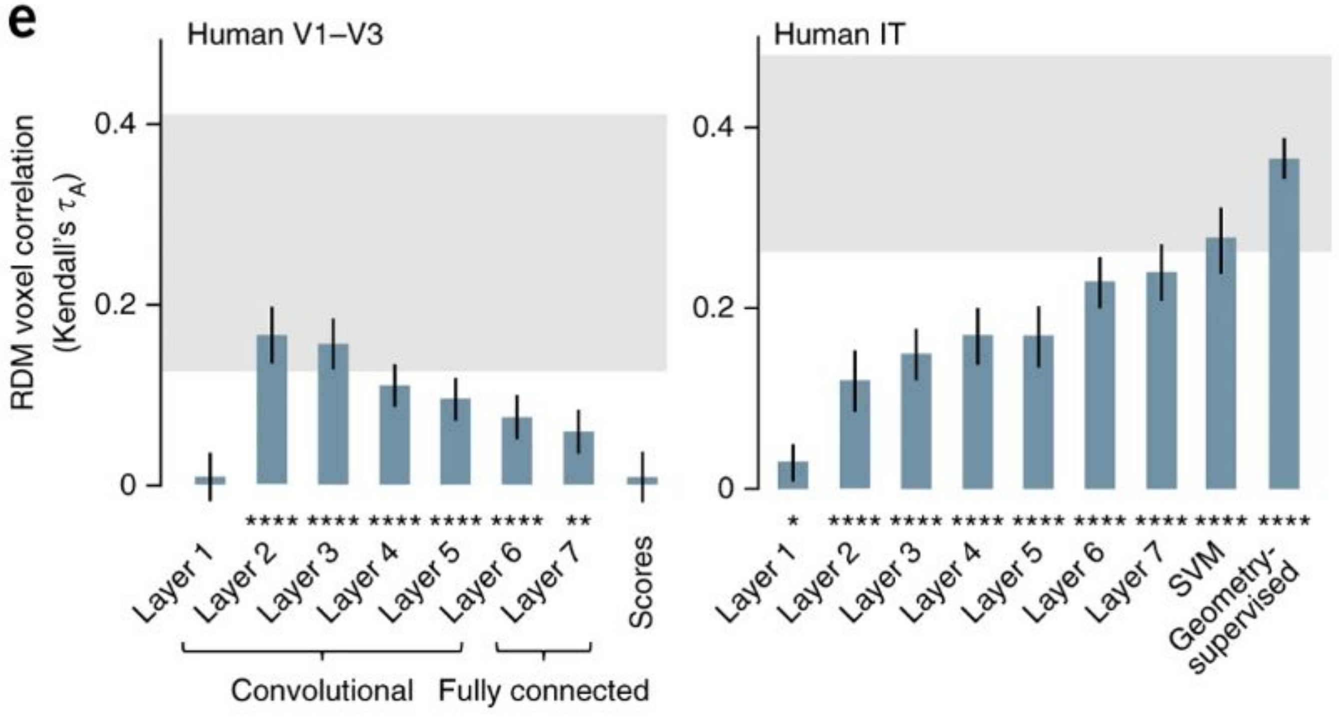
d



Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Figure 2:

- Earlier layers correlate with earlier brain regions, and vice versa
- "RDM similarity, measured with Kendall's τ_A, between HCNN model layer features and human V1–V3 (left) or human IT (right). "



Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Figure 3: Goal-driven learning follows 3 "requirements"

- TLDR: try to integrate knowledge about functional and anatomical brain structure into machine learning to answers questions

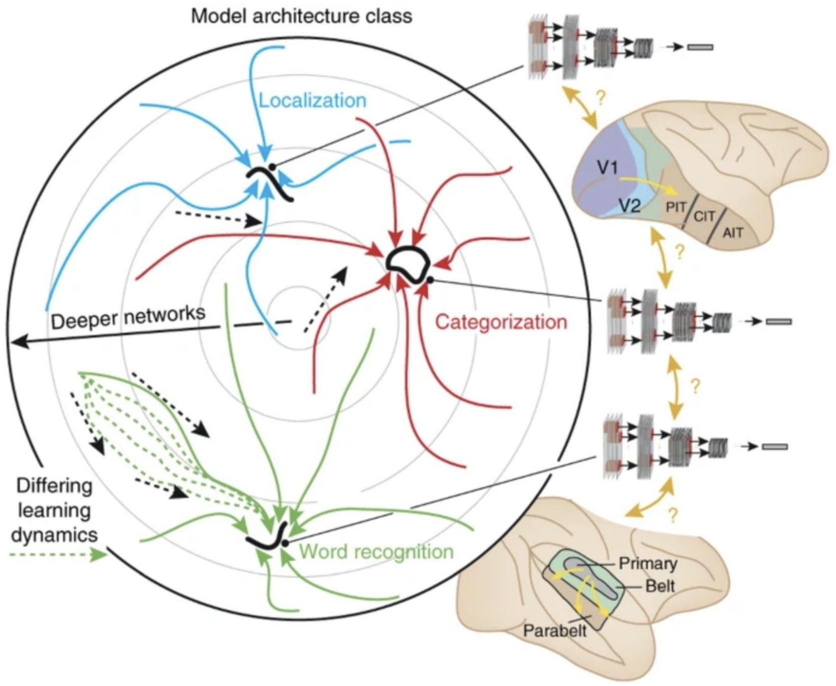
Goal-driven deep neural network models are built from three basic components (Fig. 3):

1 a model architecture class from which the system is built, formalizing knowledge about the brain's anatomical and functional connectivity;

2 a behavioral goal that the system must accomplish, such as object categorization; and

3 a learning rule that optimizes parameters within the model class to achieve the behavioral goal.

Figure 3: The components of goal-driven modeling.



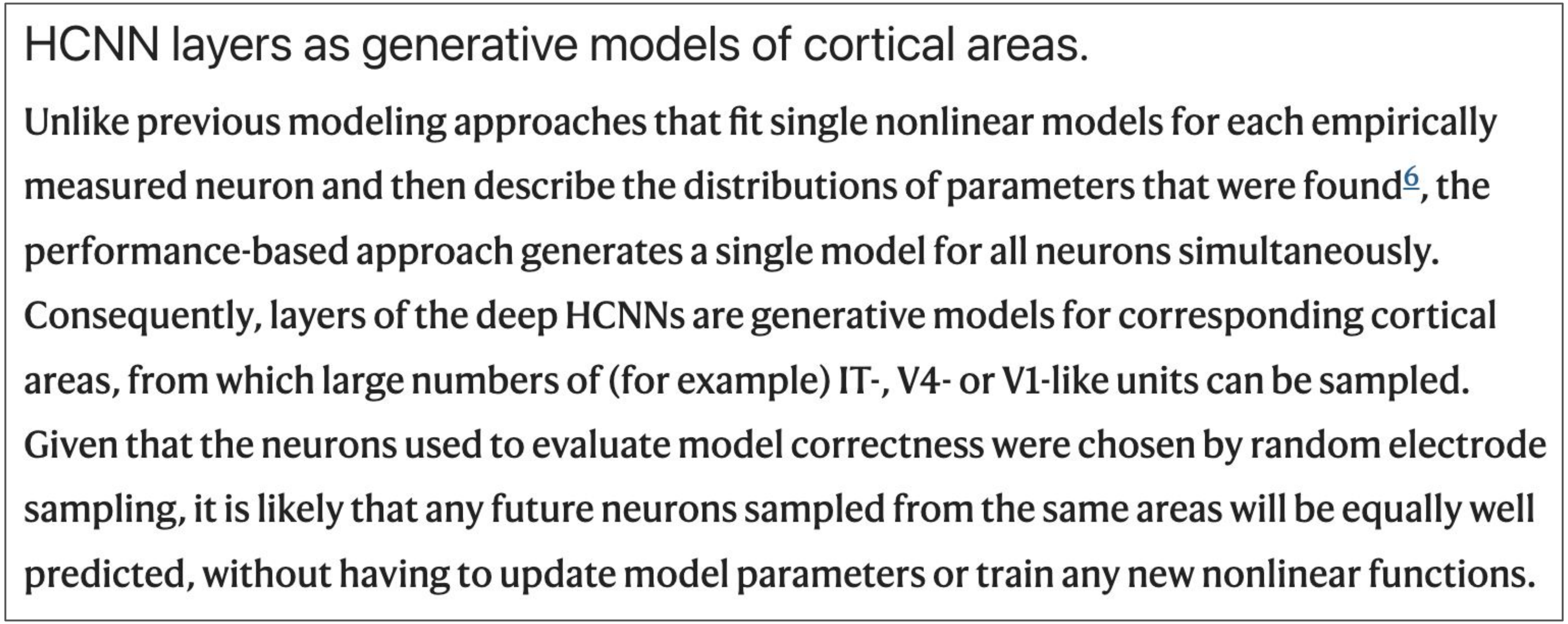
Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Outline:

- Background on neural networks
- Figures from Yamins et al., 2016
- **Discussion**

Bold claims: HCNNs as generative models

- Can be potentially used to reflect neurons (not just the one sampled in fig 2)



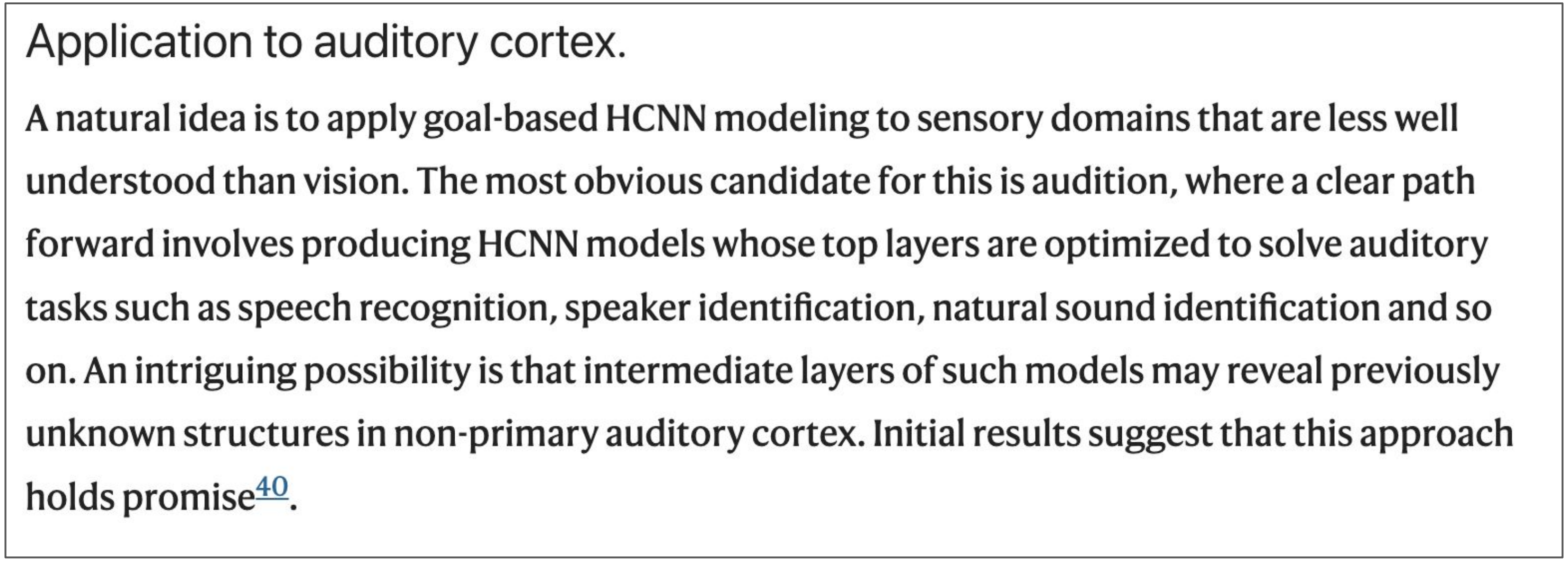
HCNN layers as generative models of cortical areas.

Unlike previous modeling approaches that fit single nonlinear models for each empirically measured neuron and then describe the distributions of parameters that were found[6], the performance-based approach generates a single model for all neurons simultaneously. Consequently, layers of the deep HCNNs are generative models for corresponding cortical areas, from which large numbers of (for example) IT-, V4- or V1-like units can be sampled. Given that the neurons used to evaluate model correctness were chosen by random electrode sampling, it is likely that any future neurons sampled from the same areas will be equally well predicted, without having to update model parameters or train any new nonlinear functions.

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Bold claims: HCNNs to predict unknown function

- Extrapolate to predict other sensory brain regions despite showing non-specific modeling in V4 / IT



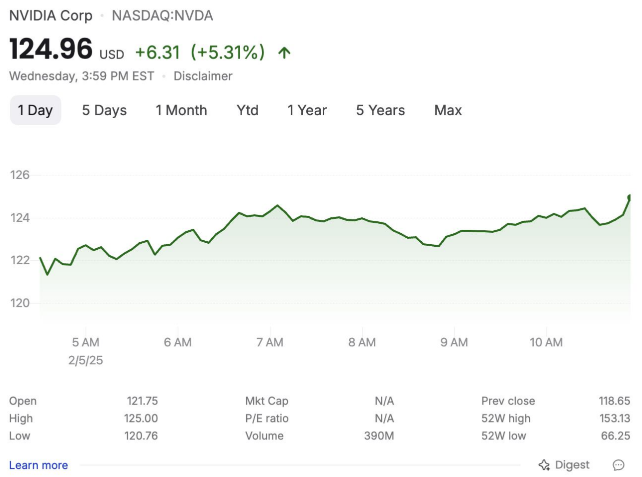
Application to auditory cortex.

A natural idea is to apply goal-based HCNN modeling to sensory domains that are less well understood than vision. The most obvious candidate for this is audition, where a clear path forward involves producing HCNN models whose top layers are optimized to solve auditory tasks such as speech recognition, speaker identification, natural sound identification and so on. An intriguing possibility is that intermediate layers of such models may reveal previously unknown structures in non-primary auditory cortex. Initial results suggest that this approach holds promise[40].

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Predicted prevalence of GPUs for deep learning

- Nvidia stocks go up



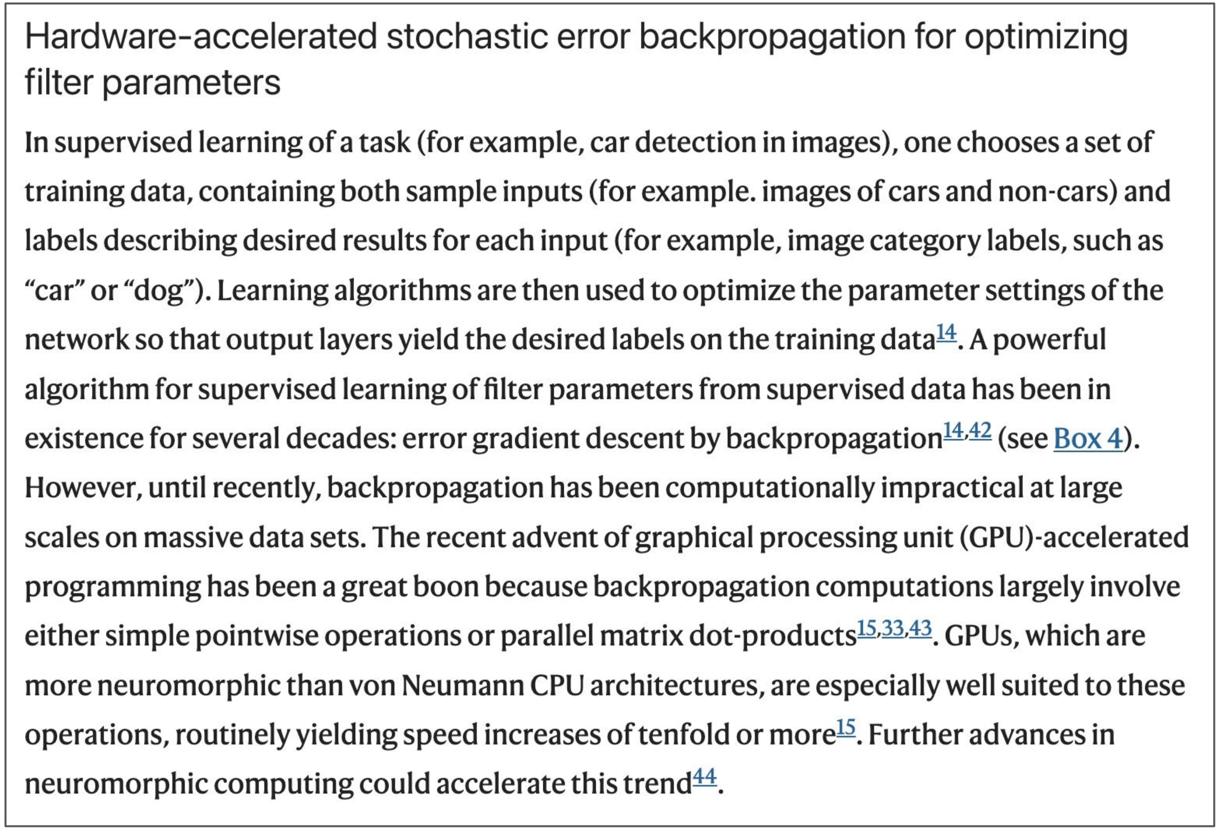
NVIDIA Corp · NASDAQ:NVDA

124.96 USD +6.31 (+5.31%) ↑

Wednesday, 3:59 PM EST · Disclaimer

Open	121.75	Mkt Cap	N/A	Prev close	118.65
High	125.00	P/E ratio	N/A	52W high	153.13
Low	120.76	Volume	390M	52W low	66.25

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

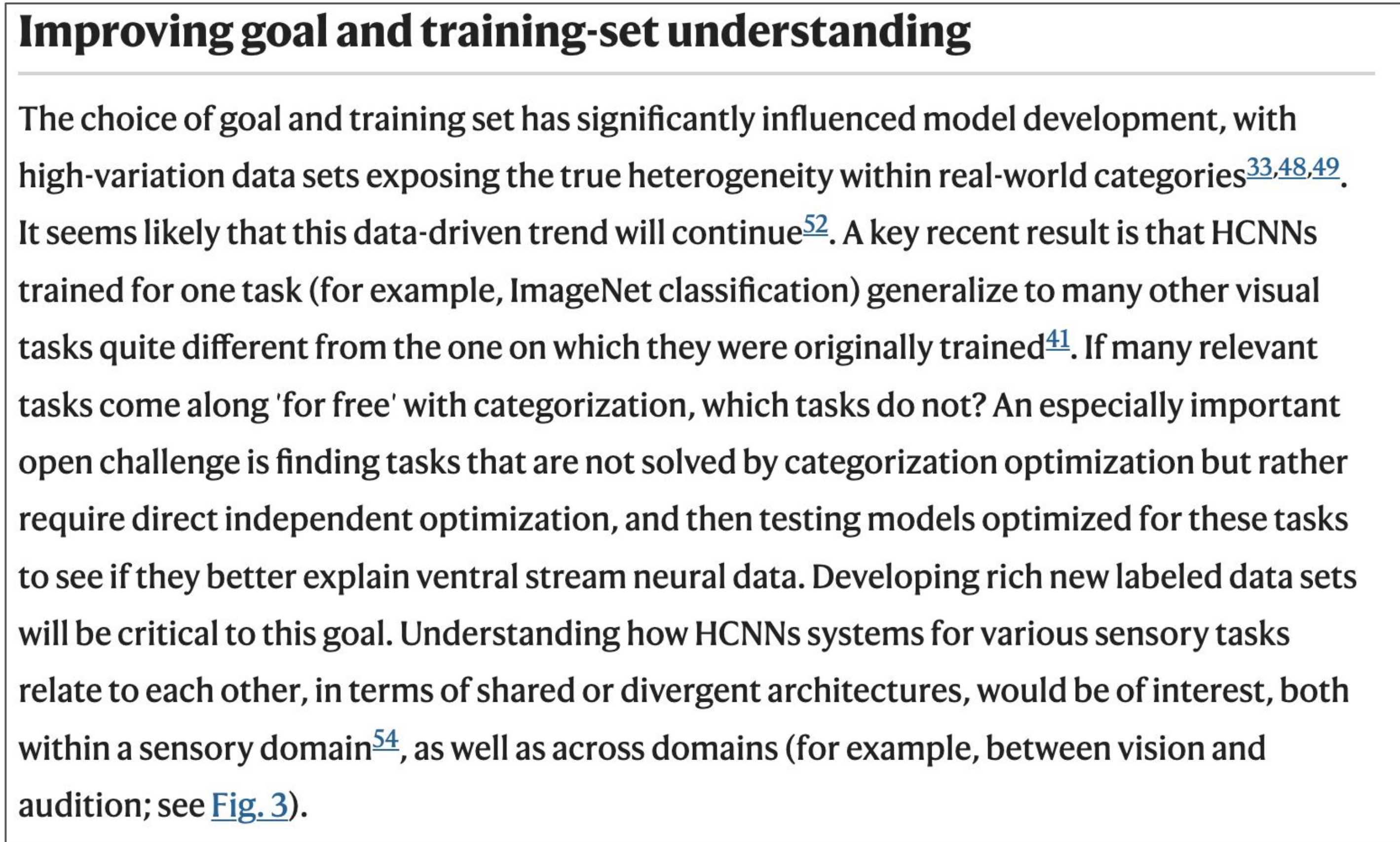


Hardware-accelerated stochastic error backpropagation for optimizing filter parameters

In supervised learning of a task (for example, car detection in images), one chooses a set of training data, containing both sample inputs (for example. images of cars and non-cars) and labels describing desired results for each input (for example, image category labels, such as "car" or "dog"). Learning algorithms are then used to optimize the parameter settings of the network so that output layers yield the desired labels on the training data[14]. A powerful algorithm for supervised learning of filter parameters from supervised data has been in existence for several decades: error gradient descent by backpropagation[14,42] (see Box 4). However, until recently, backpropagation has been computationally impractical at large scales on massive data sets. The recent advent of graphical processing unit (GPU)-accelerated programming has been a great boon because backpropagation computations largely involve either simple pointwise operations or parallel matrix dot-products[15,33,43]. GPUs, which are more neuromorphic than von Neumann CPU architectures, are especially well suited to these operations, routinely yielding speed increases of tenfold or more[15]. Further advances in neuromorphic computing could accelerate this trend[44].

"Transfer learning" from one domain to another

- Free training
- Why and how?



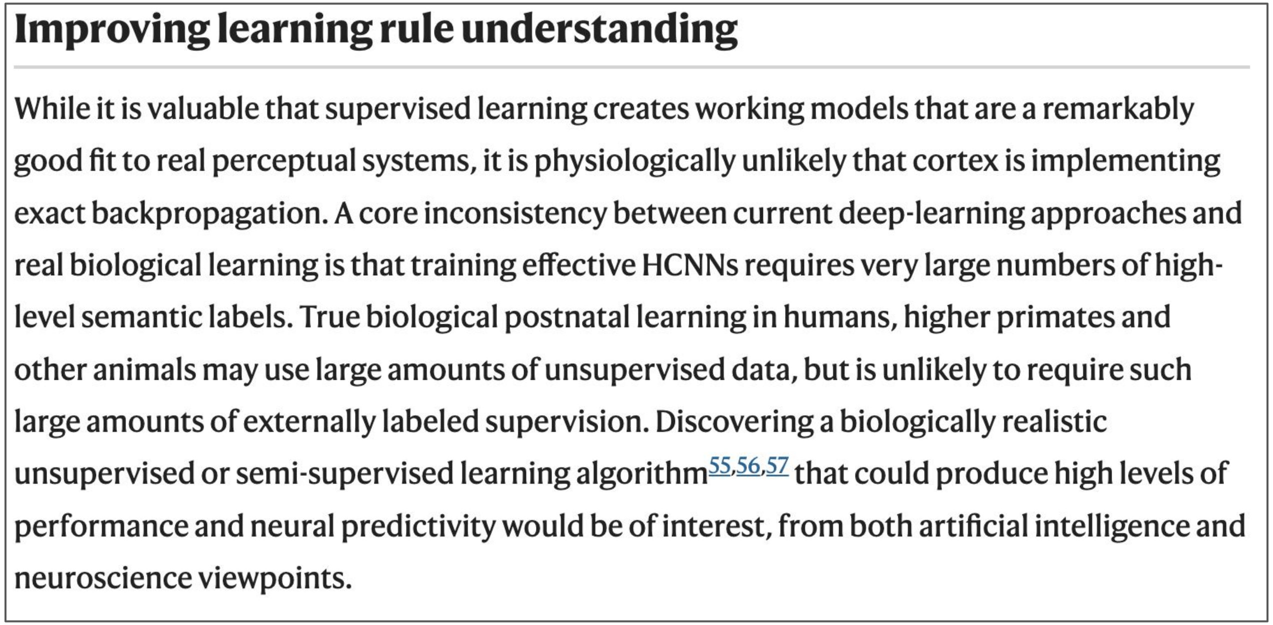
Improving goal and training-set understanding

The choice of goal and training set has significantly influenced model development, with high-variation data sets exposing the true heterogeneity within real-world categories[33,48,49]. It seems likely that this data-driven trend will continue[52]. A key recent result is that HCNNs trained for one task (for example, ImageNet classification) generalize to many other visual tasks quite different from the one on which they were originally trained[41]. If many relevant tasks come along 'for free' with categorization, which tasks do not? An especially important open challenge is finding tasks that are not solved by categorization optimization but rather require direct independent optimization, and then testing models optimized for these tasks to see if they better explain ventral stream neural data. Developing rich new labeled data sets will be critical to this goal. Understanding how HCNNs systems for various sensory tasks relate to each other, in terms of shared or divergent architectures, would be of interest, both within a sensory domain[54], as well as across domains (for example, between vision and audition; see Fig. 3).

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Disparities between real learning data and ML data size

- Babies learn "faster" than computers and yet are incapable of contributing to my research



Improving learning rule understanding

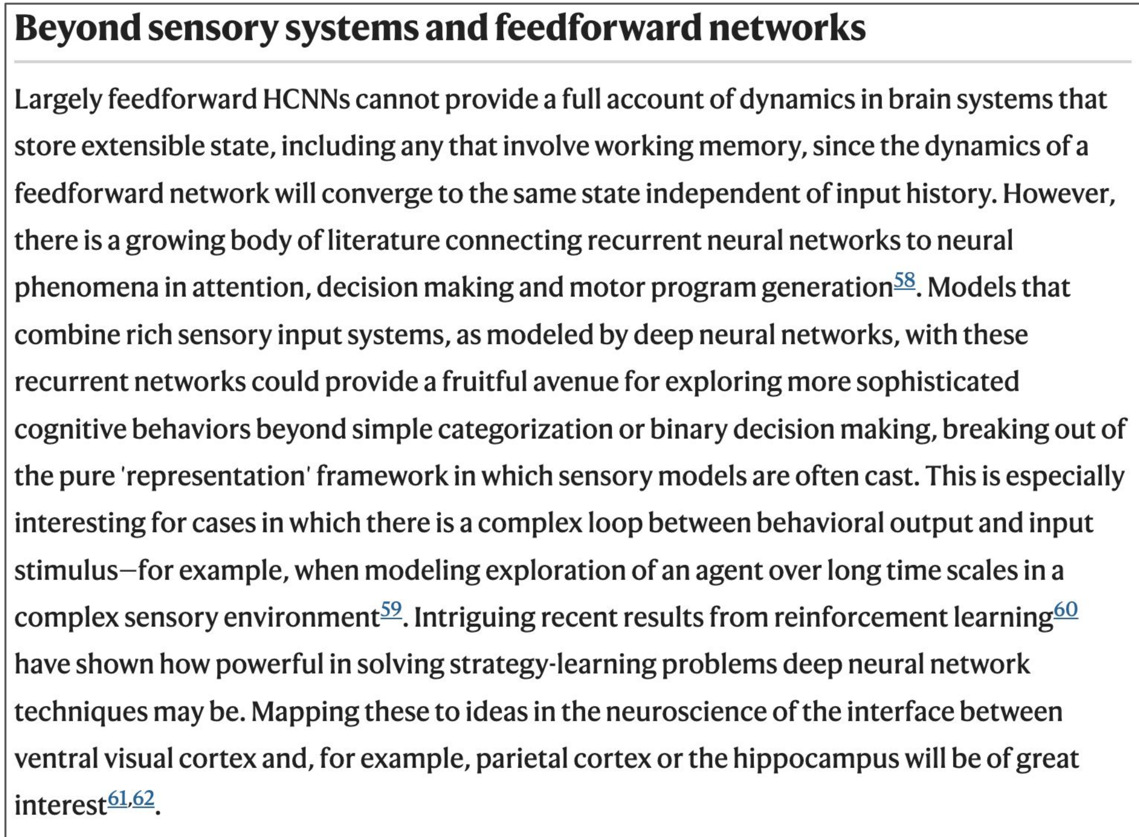
While it is valuable that supervised learning creates working models that are a remarkably good fit to real perceptual systems, it is physiologically unlikely that cortex is implementing exact backpropagation. A core inconsistency between current deep-learning approaches and real biological learning is that training effective HCNNs requires very large numbers of high-level semantic labels. True biological postnatal learning in humans, higher primates and other animals may use large amounts of unsupervised data, but is unlikely to require such large amounts of externally labeled supervision. Discovering a biologically realistic unsupervised or semi-supervised learning algorithm[55,56,57] that could produce high levels of performance and neural predictivity would be of interest, from both artificial intelligence and neuroscience viewpoints.

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244

Feed forward neural networks have no "memory"

- May benefit from memory cells, recurrence, or attention mechanisms

Yamins, D., DiCarlo, J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 19, 356–365 (2016). https://doi.org/10.1038/nn.4244



Beyond sensory systems and feedforward networks

Largely feedforward HCNNs cannot provide a full account of dynamics in brain systems that store extensible state, including any that involve working memory, since the dynamics of a feedforward network will converge to the same state independent of input history. However, there is a growing body of literature connecting recurrent neural networks to neural phenomena in attention, decision making and motor program generation[58]. Models that combine rich sensory input systems, as modeled by deep neural networks, with these recurrent networks could provide a fruitful avenue for exploring more sophisticated cognitive behaviors beyond simple categorization or binary decision making, breaking out of the pure 'representation' framework in which sensory models are often cast. This is especially interesting for cases in which there is a complex loop between behavioral output and input stimulus—for example, when modeling exploration of an agent over long time scales in a complex sensory environment[59]. Intriguing recent results from reinforcement learning[60] have shown how powerful in solving strategy-learning problems deep neural network techniques may be. Mapping these to ideas in the neuroscience of the interface between ventral visual cortex and, for example, parietal cortex or the hippocampus will be of great interest[61,62].

Pop quiz!

- What is the output of this convolution?

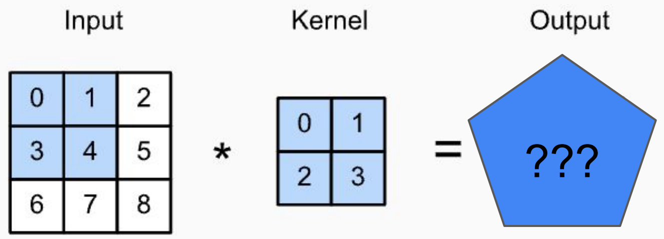


Fig. 7.2.1 Two-dimensional cross-correlation operation. The shaded portions are the first output element as well as the input and kernel tensor elements used for the output computation: (

Hint: It's a "dot product"!

D2L AI textbook

Pop quiz!

- What is the output of this convolution?

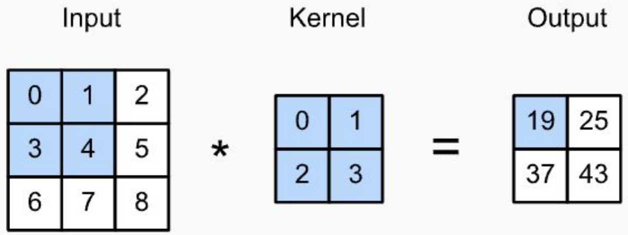


Fig. 7.2.1 Two-dimensional cross-correlation operation. The shaded portions are the first output element as well as the input and kernel tensor elements used for the output computation: $0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$.

D2L AI textbook