CPSC 340:
Machine Learning and Data Mining

Neural Networks: the model (“predict”)

Original version of these slides by Mark Schmidt, with modifications by Mike Gelbart.

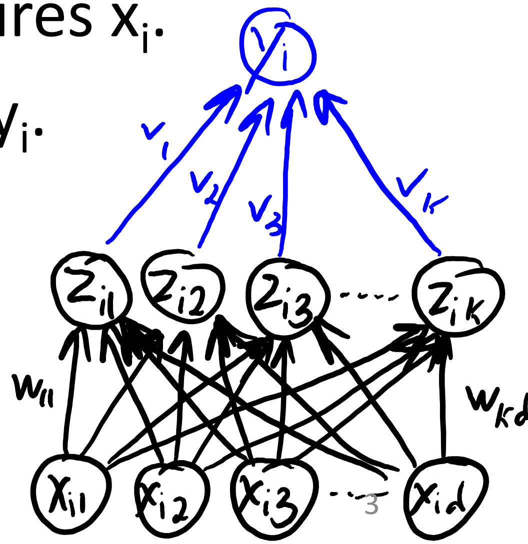
Admin

- Assignment 5:
  - Due tonight.

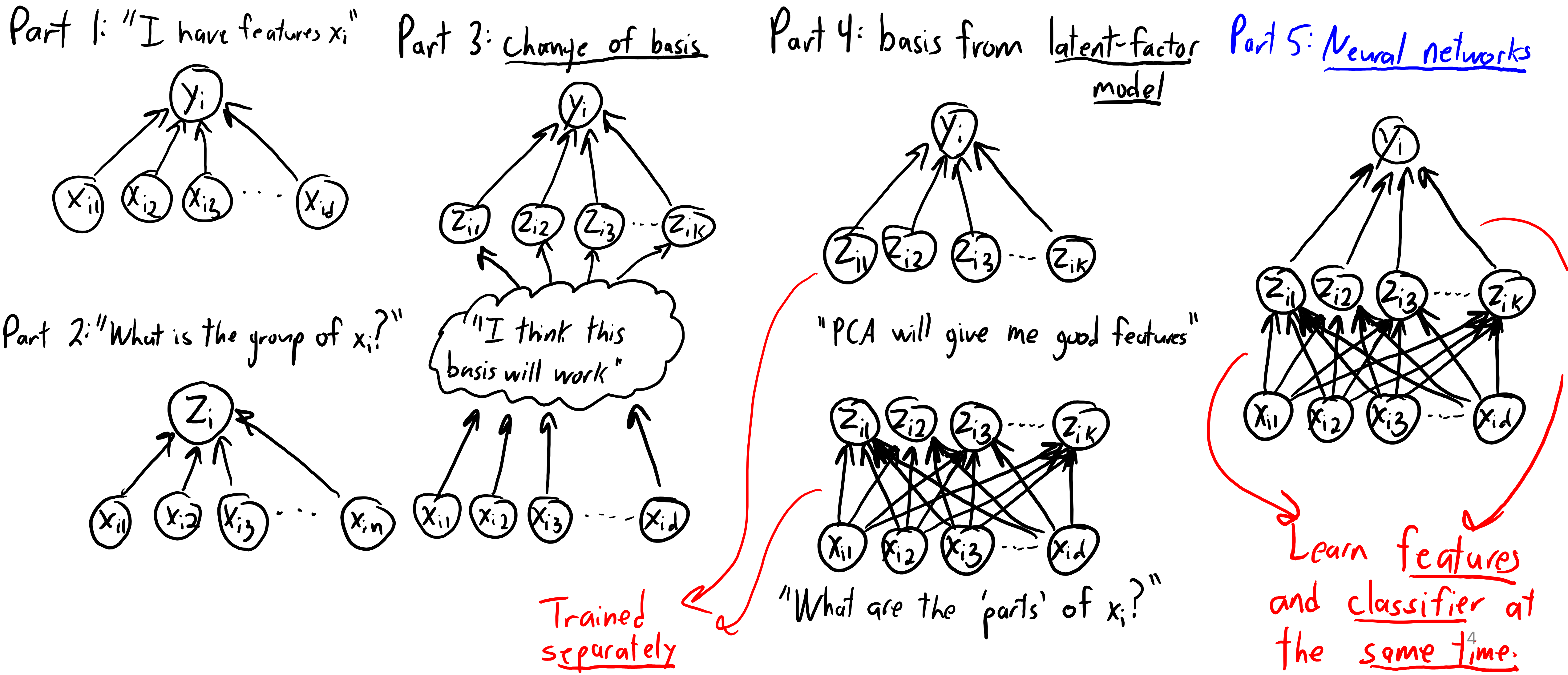
- Assignment 6:
  - Will be released very soon.
  - Due in 13 days (Thursday April 5).

Supervised Learning Roadmap

- Part 1: "Direct" Supervised Learning.
  - We learned parameters 'w' based on the original features x_i and target y_i.
- Part 3: Change of Basis.
  - We learned parameters 'w' based on a change of basis z_i and target y_i.
- Part 4: Latent-Factor Models.
  - We learned parameters 'W' for basis z_i based on only on features x_i.
  - You can then learn 'w' based on change of basis z_i and target y_i.
- Part 5: Neural Networks.
  - Jointly learn 'W' and 'w' based on x_i and y_i.
  - **Learn basis z_i that is good for supervised learning.**



A Graphical Summary of CPSC 340 Parts 1-5



Notation for Neural Networks

We have our usual supervised learning notation:

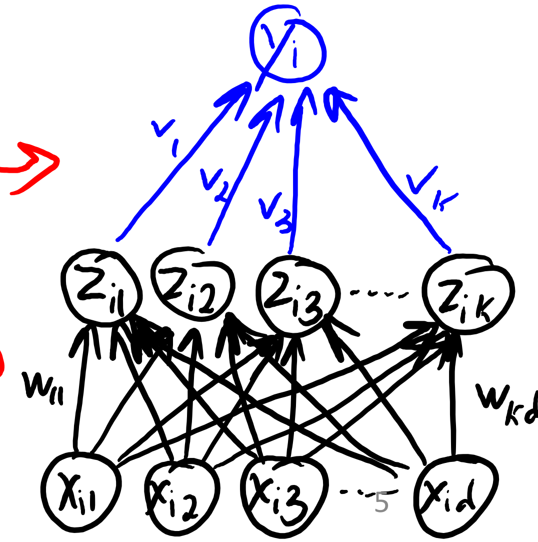
$$X = \underset{n \times d}{\begin{bmatrix} \text{---} & x_1^T & \text{---} \\ \text{---} & x_2^T & \text{---} \\ & \vdots & \\ \text{---} & x_n & \text{---} \end{bmatrix}} \qquad y = \underset{n \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}$$

We have our latent features:

$$Z = \underset{n \times K}{\begin{bmatrix} \text{---} & z_1^T & \text{---} \\ \text{---} & z_2^T & \text{---} \\ & \vdots & \\ \text{---} & z_n^T & \text{---} \end{bmatrix}}$$

We have <u>two</u> sets of parameters:

$$v = \underset{k \times 1}{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_K \end{bmatrix}} \qquad W = \underset{k \times d}{\begin{bmatrix} \text{---} & w_1 & \text{---} \\ \text{---} & w_2 & \text{---} \\ & \vdots & \\ \text{---} & w_K & \text{---} \end{bmatrix}}$$



Linear-Linear Model

- Obvious choice: linear latent-factor model with linear regression.

  Use features from latent-factor model: $z_i = Wx_i$

  Make predictions using a linear model: $y_i = v^T z_i$

- We want to train 'W' and 'v' jointly, so we could minimize:

$$f(W,v) = \frac{1}{2}\sum_{i=1}^{n} \underbrace{(v^T z_i - y_i)^2}_{\text{linear regression with } z_i \text{ as features}} = \frac{1}{2}\sum_{i=1}^{n}(v^T \underbrace{(Wx_i)}_{z_i \text{ come from latent-factor model}} - y_i)^2$$

- But this is just a linear model:

$$y_i = v^T z_i = v^T(Wx_i) = \underbrace{(\overbrace{v^T W}^{1 \times d})}_{\text{some vector 'w'}} x_i = w^T x_i$$

Introducing Non-Linearity

- To increase flexibility, something needs to be non-linear.
- Typical choice: transform z_i by non-linear function 'h'.

$$z_i = Wx_i \qquad\qquad y_i = v^T h(z_i)$$

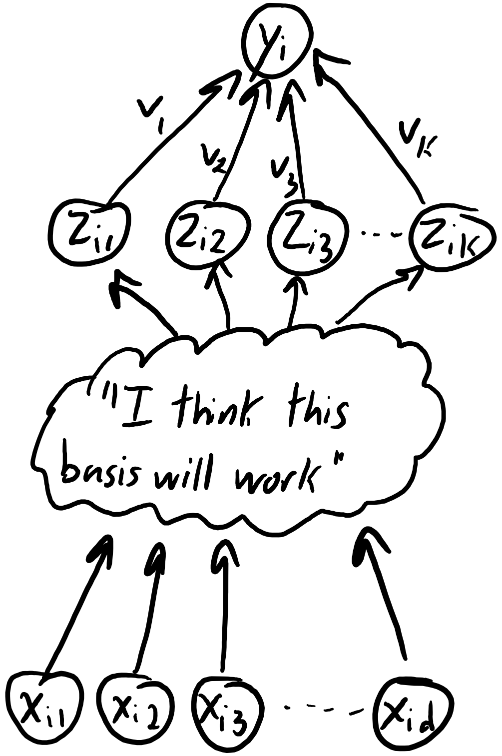
  – Here the function 'h' transforms 'k' inputs to 'k' outputs.
- Common choice for 'h': applying sigmoid function element-wise:

$$h(z_{ic}) = \frac{1}{1 + \exp(-z_{ic})}$$

  – So this takes the z_{ic} in $(-\infty,\infty)$ and maps it to $(0,1)$.
  – We'll see another activation function next class.
- This is called a "multi-layer perceptron" or a "neural network".

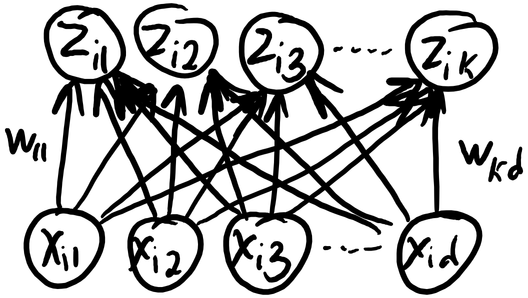
Supervised Learning Roadmap

Hand-engineered features:

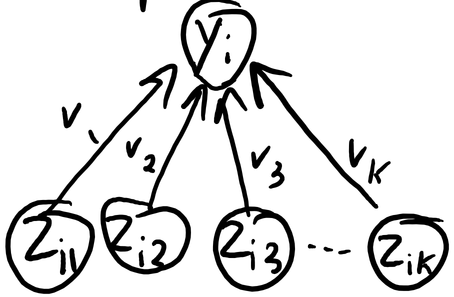


Requires domain knowledge and can be time-consuming

Learn a latent-factor model:

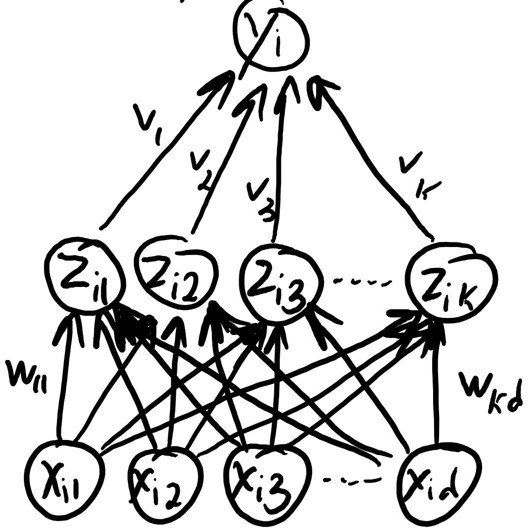


Use latent features in supervized model:



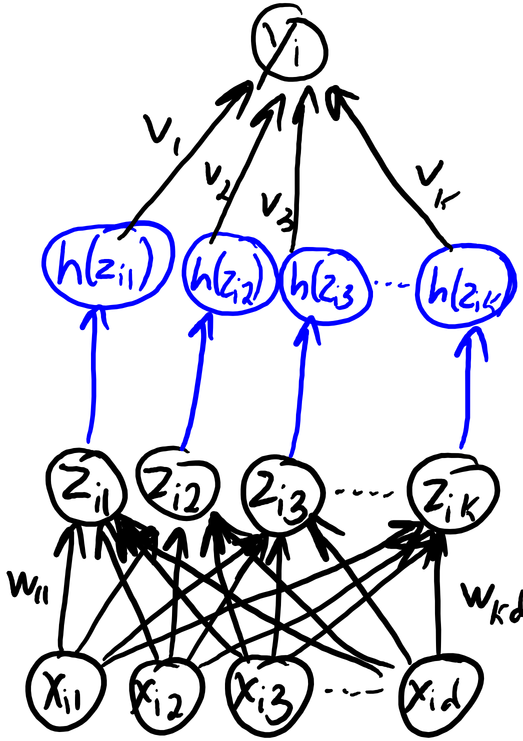
Good representation of x_i might be bad for predicting y_i

Learn 'w' and 'W' together:



But still gives a linear model.

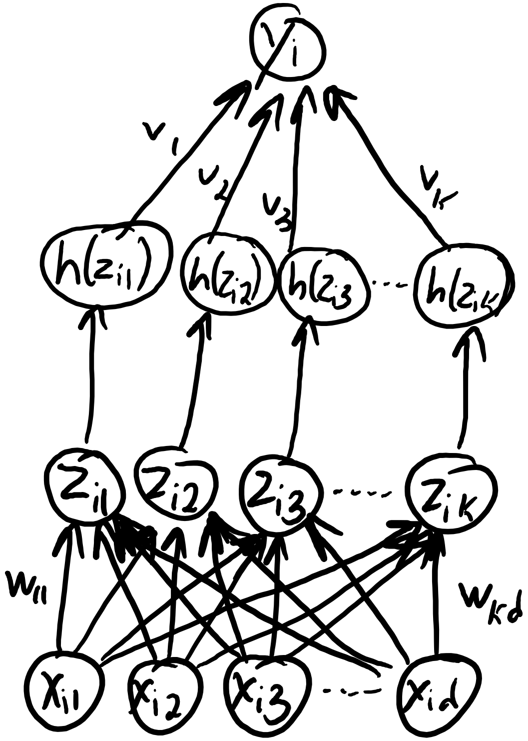
Neural network:



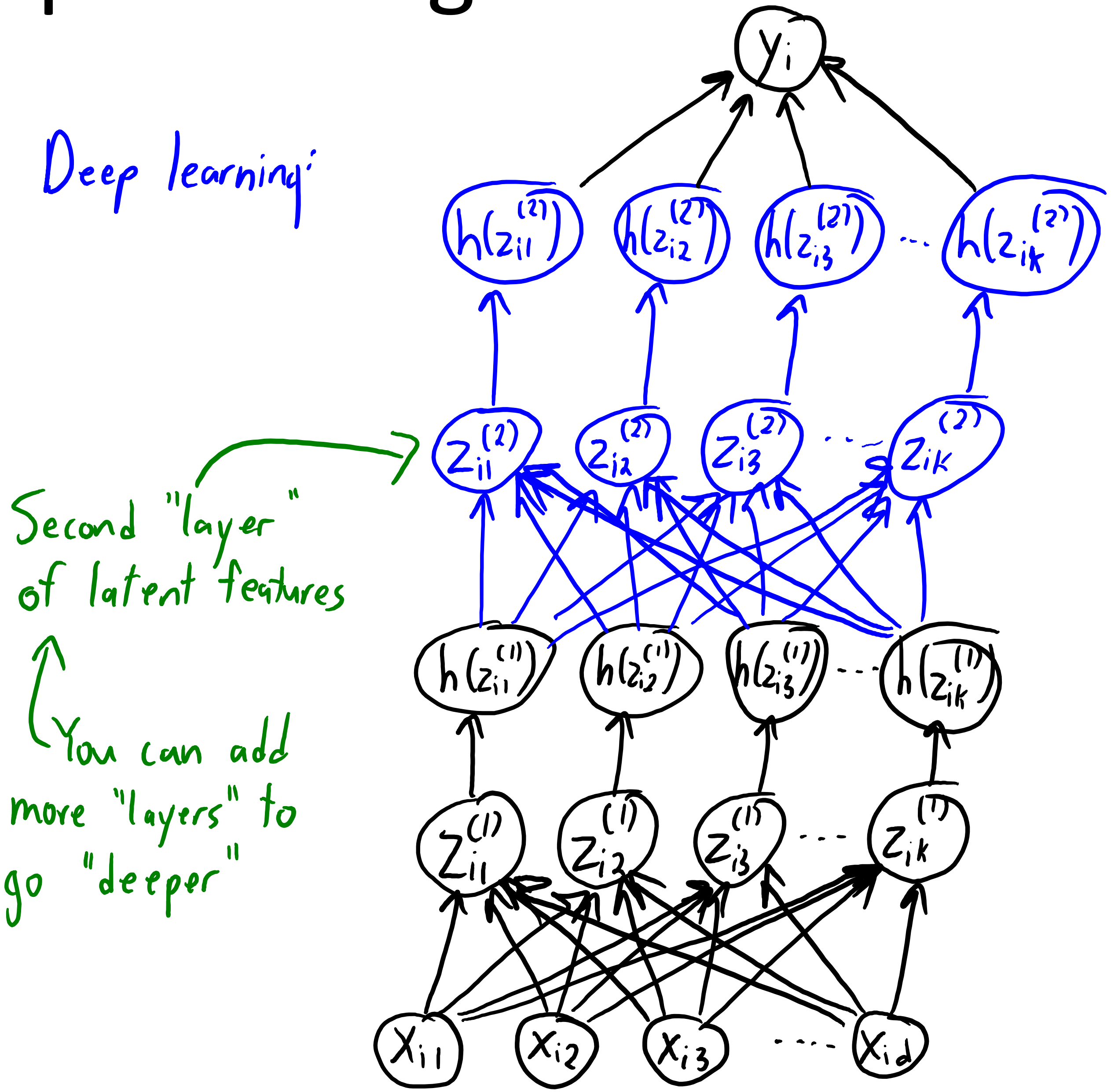
Extra non-linear transformation 'h'.

Deep Learning

Neural network:



Deep learning:



Deep Learning

Linear model:

$$\hat{y}_i = w^T x_i$$

Neural network with 1 hidden layer:

$$\hat{y}_i = v^T h(\underbrace{Wx_i}_{z_i})$$

Neural network with 2 hidden layers:

$$\hat{y}_i = v^T h(\underbrace{W^{(2)} h(\underbrace{W^{(1)} x_i}_{z_i^{(1)}})}_{z_i^{(2)}})$$

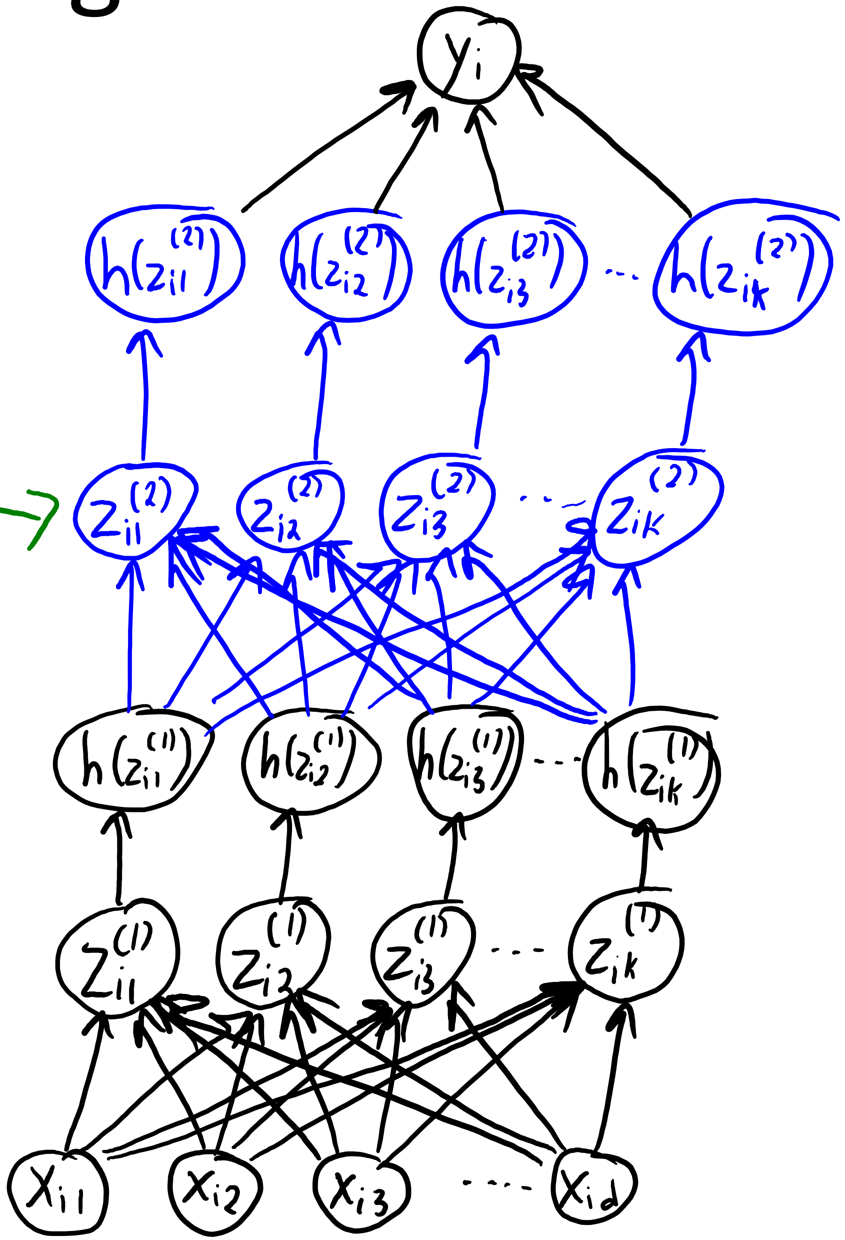
Neural network with 3 hidden layers

$$\hat{y}_i = v^T h(\underbrace{W^{(3)} h(\underbrace{W^{(2)} h(\underbrace{W^{(1)} x_i}_{z_i^{(1)}})}_{z_i^{(2)}})}_{z_i^{(3)}})$$

Deep learning:

Second "layer" of latent features

You can add more "layers" to go "deeper"



Adding Bias Variables

- Recall fitting line regression with a bias:

$$\hat{y}_i = \sum_{j=1}^{d} w_j x_{ij} + \beta$$

  – We avoided this by adding a column of ones to X.

- In neural networks we often want a bias on the output:

$$\hat{y}_i = \sum_{c=1}^{k} v_c h(w_c^T x_i) + \beta$$

- But we also often also include biases on each z_{ic}:

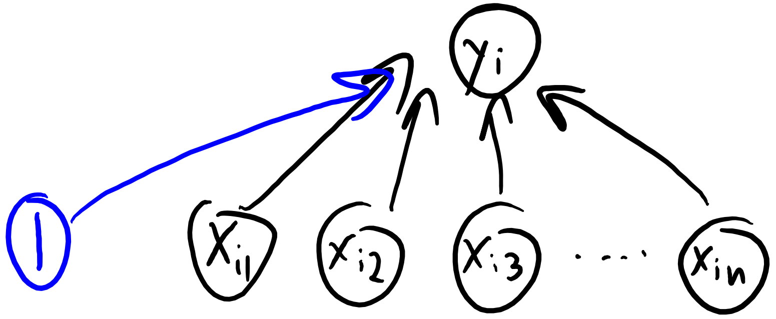
$$\hat{y}_i = \sum_{c=1}^{k} v_c h(w_c^T x_i + \beta_c) + \beta$$

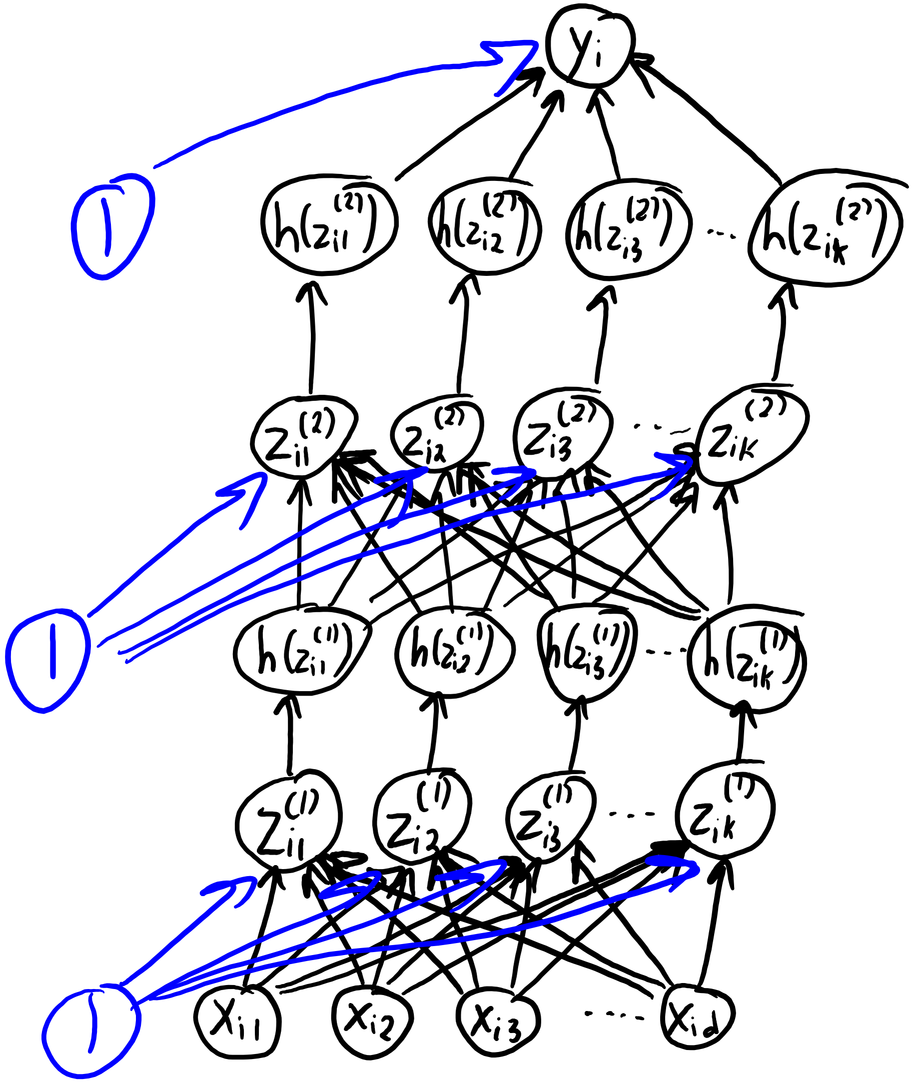
  – A bias towards this $h(z_{ic})$ being either 0 or 1.

- Equivalent to adding to vector $h(z_i)$ an extra value that is always 1.

Adding Bias Variables

Linear model with bias:





Jupyter notebook demo

(Very Abridged) Deep Learning History

- 1950s and 1960s: initial excitement
  - MIT students assigned to solve object recognition over the summer
- 1970s-2000s: progress but also disappointment, “AI winter”
  - SVMs very popular in 1990s & 2000s
- Late 2000s-2010s: the return of deep learning
  - Similar models but new tricks, bigger data, more processing power, GPUs
  - Huge improvements in automatic speech recognition (2009).
    - All phones now have deep learning.
  - Huge improvements in computer vision (2012).
    - This is now finding its way into products

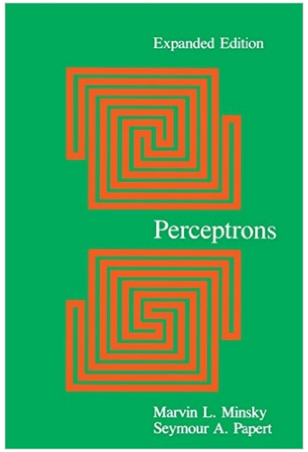
Vocabulary

- deep learning
- (artificial) neural net(work)
- NN, ANN, CNN
- layers
- units, neurons, activations
- hidden, visible
- activation function, nonlinearity

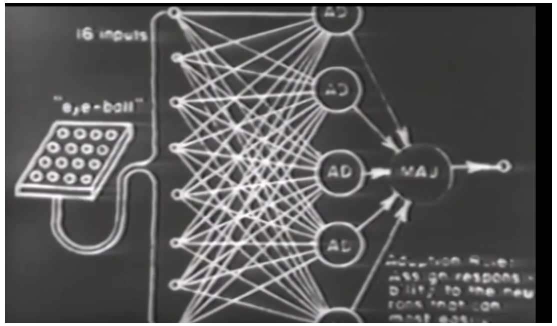
Summary

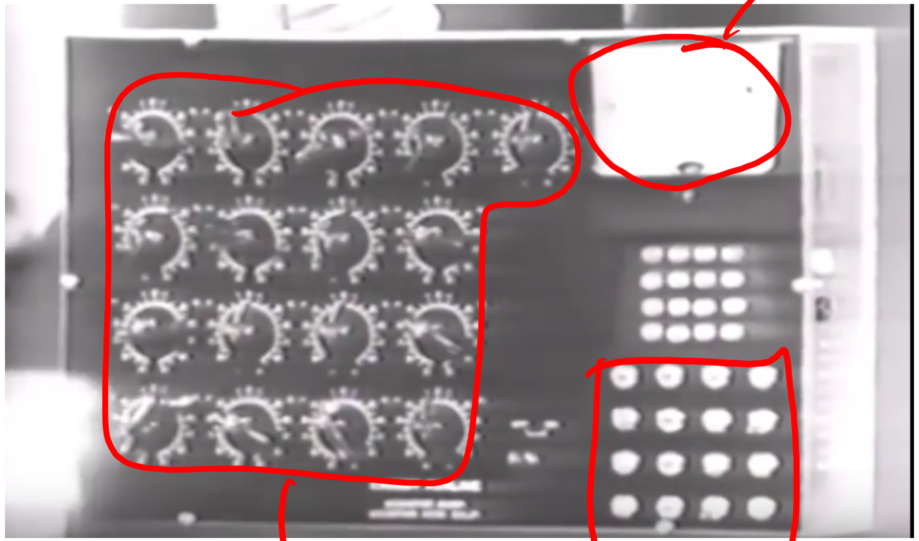
- Neural networks: simultaneously learn features and regression coefficients for supervised learning.
- Sigmoid function: avoids degeneracy by introducing non-linearity.
- Deep learning: neural networks with many hidden layers.
- Unprecedented performance on difficult pattern recognition tasks.

ML and Deep Learning History



- 1950 and 1960s: Initial excitement.
  - Perceptron: linear classifier and stochastic gradient (roughly).
  - "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." New York Times (1958).
    - https://www.youtube.com/watch?v=IEFRtz68m-8
  - Marvin Minsky assigns object recognition to his students as a summer project





$w^T x_i$

w

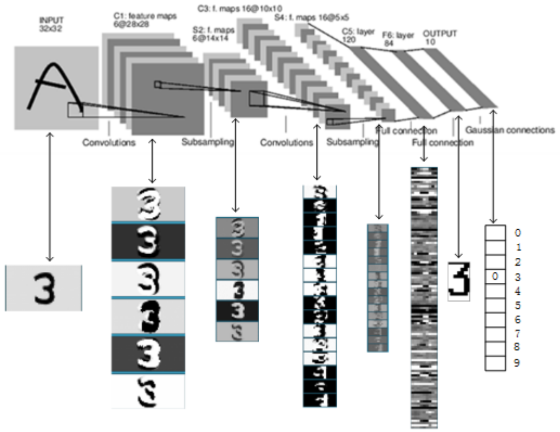
x_i

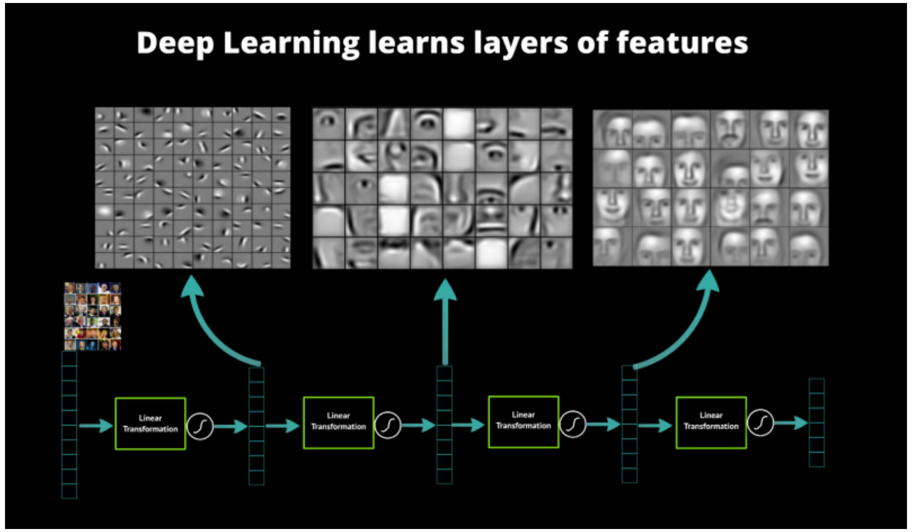
- Then drop in popularity:
  - Quickly realized limitations of linear models.

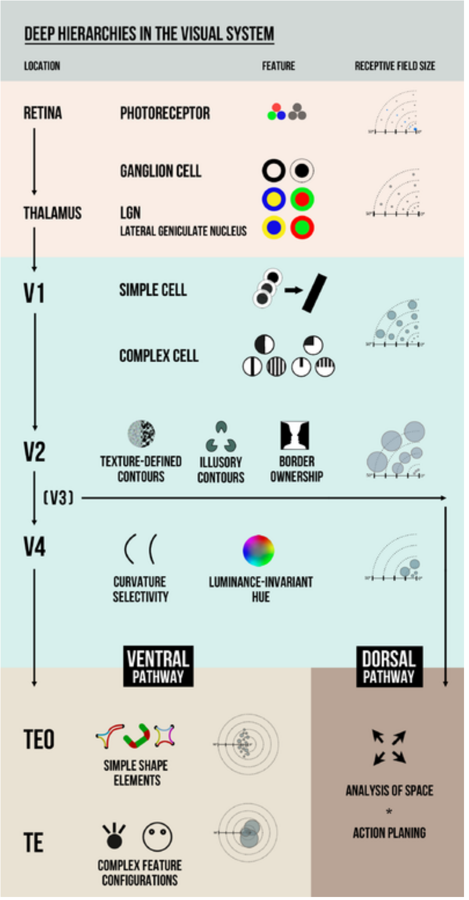
https://mitpress.mit.edu/books/perceptrons/

ML and Deep Learning History

- 1970 and 1980s: Connectionism (brain-inspired ML)
  – Want “connected networks of simple units”.
    - Use parallel computation and distributed representations.
  – Adding hidden layers z_i increases expressive power.
    - With 1 layer and enough sigmoid units, a universal approximator.
  – Success in optical character recognition.







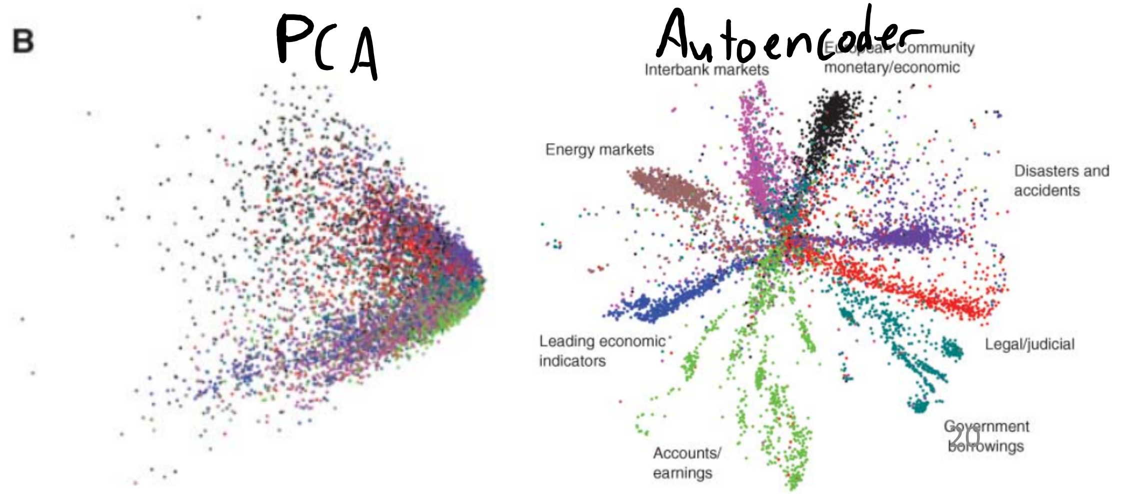
https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing
http://www.datarobot.com/blog/a-primer-on-deep-learning/
http://blog.csdn.net/strint/article/details/44163869

ML and Deep Learning History

- 1990s and early-2000s: drop in popularity.
  - It proved really difficult to get multi-layer models working robustly.
  - We obtained similar performance with simpler models:
    - Rise in popularity of logistic regression and SVMs with regularization and kernels.
  - ML moved closer to other fields (CPSC 540):
    - Numerical optimization.
    - Probabilistic graphical models.
    - Bayesian methods.

ML and Deep Learning History

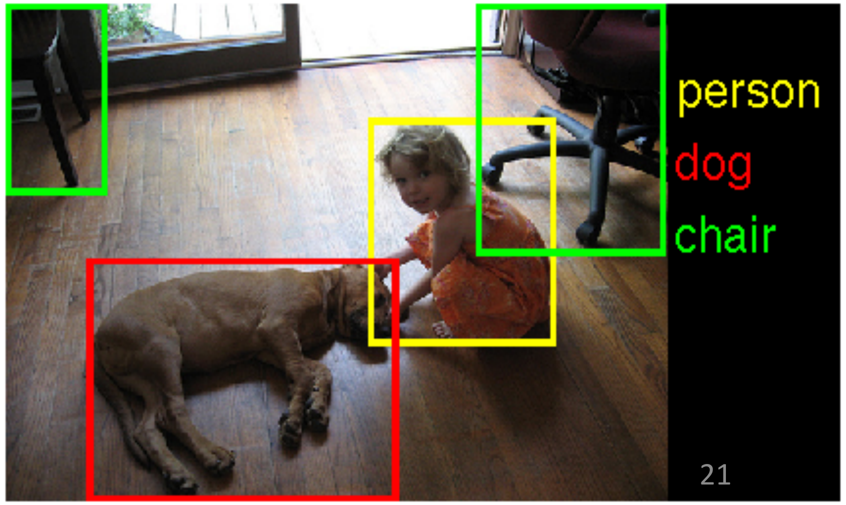
- Late 2000s: push to revive connectionism as “deep learning”.
  - Canadian Institute For Advanced Research (CIFAR) NCAP program:
    - “Neural Computation and Adaptive Perception”.
    - Led by Geoff Hinton, Yann LeCun, and Yoshua Bengio (“Canadian mafia”).
  - Unsupervised successes: “deep belief networks” and “autoencoders”.
    - Could be used to initialize deep neural networks.
    - https://www.youtube.com/watch?v=KuPai0ogiHk



https://www.cs.toronto.edu/~hinton/science.pdf

2010s: DEEP LEARNING!!!

- Bigger datasets, bigger models, parallel computing (GPUs/clusters).
  - And some tweaks to the models from the 1980s.
- Huge improvements in automatic speech recognition (2009).
  - All phones now have deep learning.
- Huge improvements in computer vision (2012).
  - Changed computer vision field almost instantly.
  - This is now finding its way into products.



http://www.image-net.org/challenges/LSVRC/2014/

2010s: DEEP LEARNING!!!

- Media hype:
  - “How many computers to identify a cat? 16,000”

    New York Times (2012).
  - “Why Facebook is teaching its machines to think like humans”

    Wired (2013).
  - “What is ‘deep learning’ and why should businesses care?”

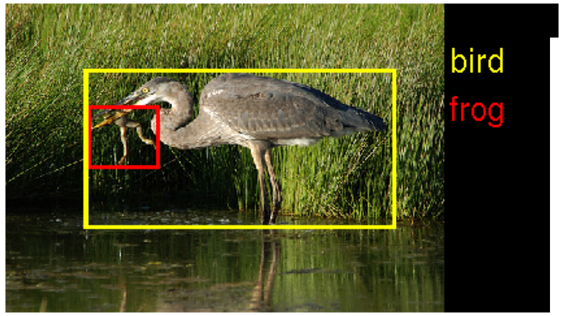
    Forbes (2013).
  - “Computer eyesight gets a lot more accurate”

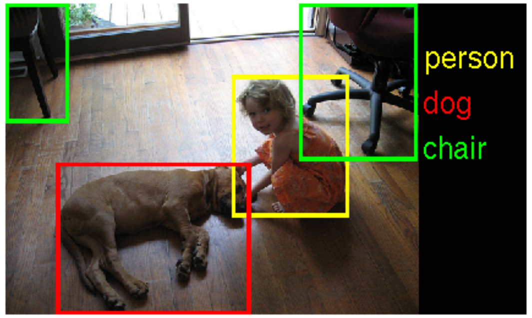
    New York Times (2014).

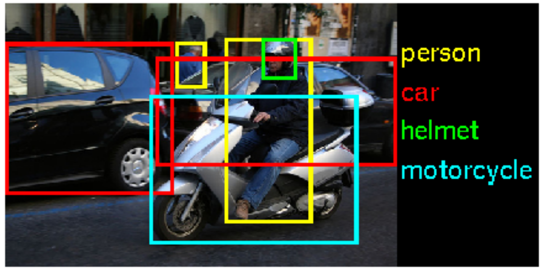
- 2015: huge improvement in language understanding.

ImageNet Challenge

- Millions of labeled images, 1000 object classes.







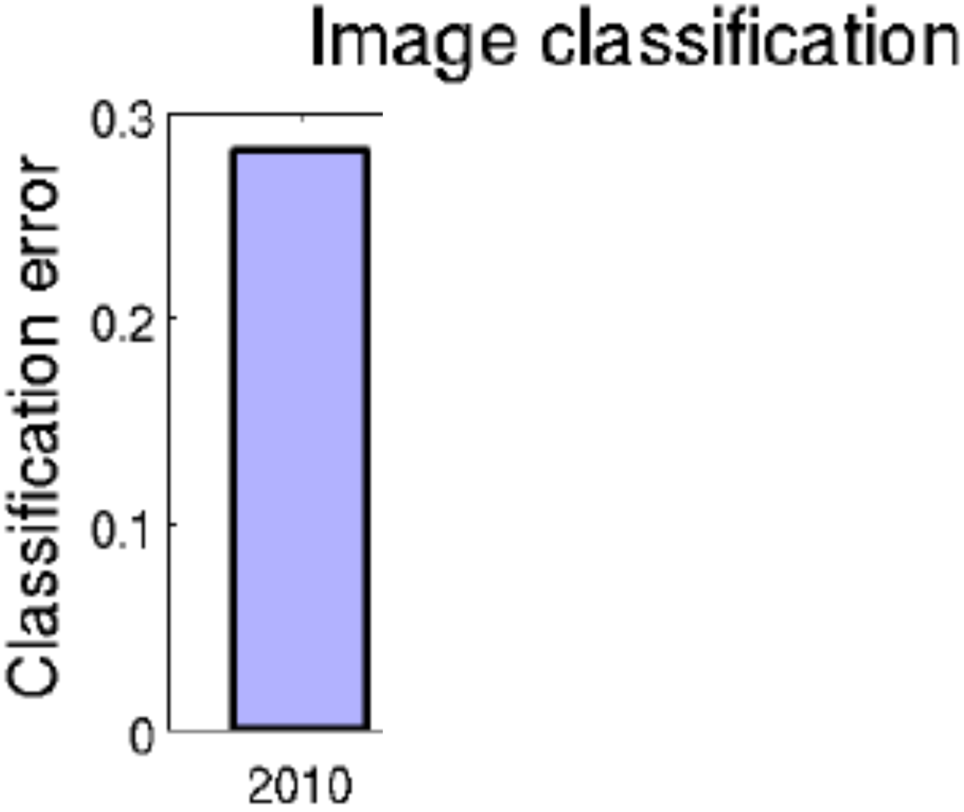
Easy for humans but <u>hard</u> for computers.

http://www.image-net.org/challenges/LSVRC/2014/

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

(a) Siberian husky (b) Eskimo dog



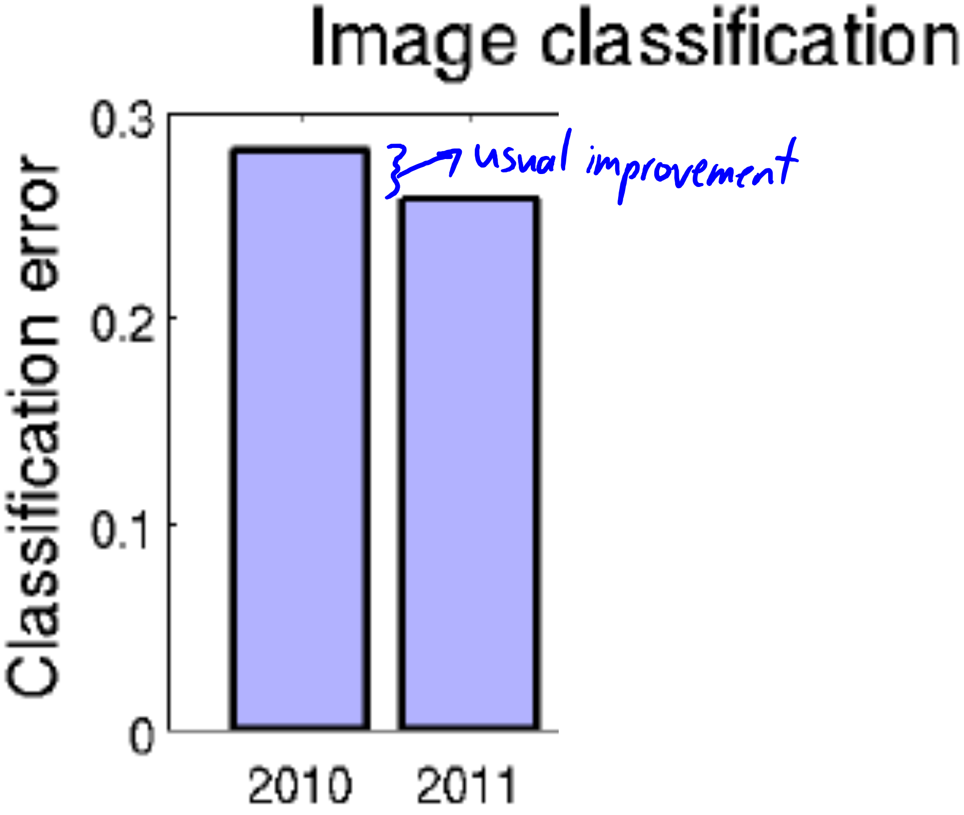
https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

(a) Siberian husky

(b) Eskimo dog



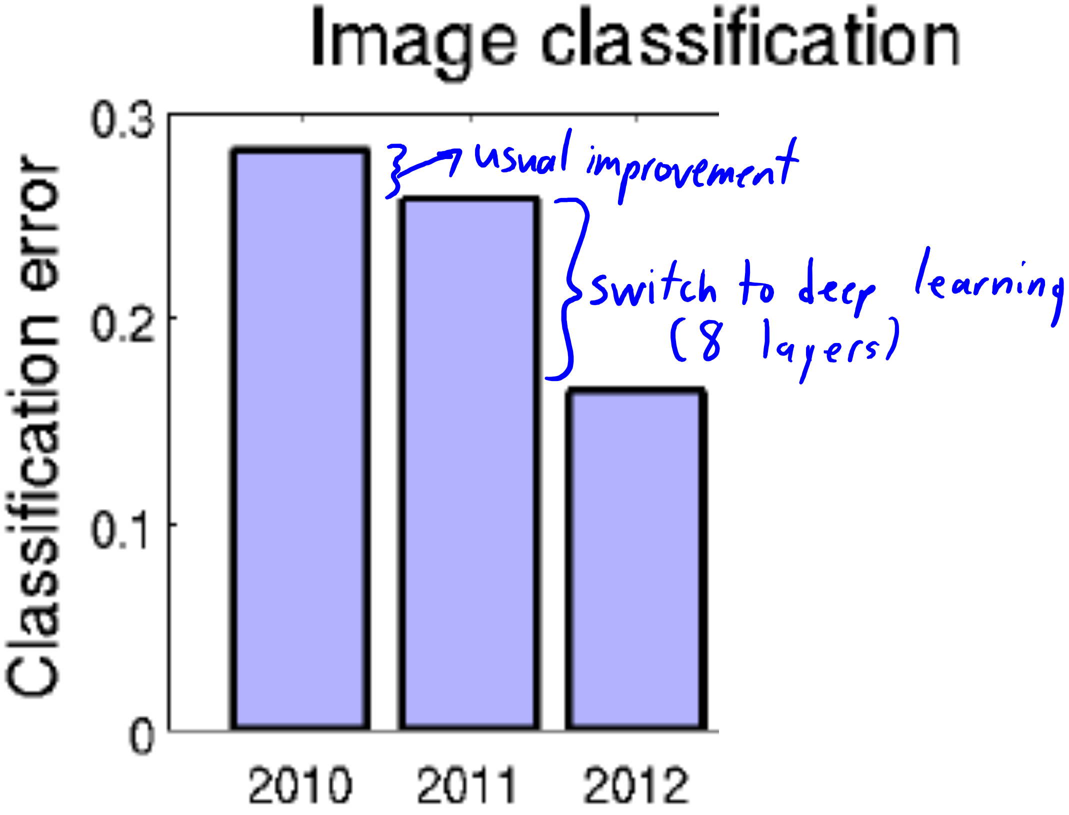
https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

(a) Siberian husky

(b) Eskimo dog

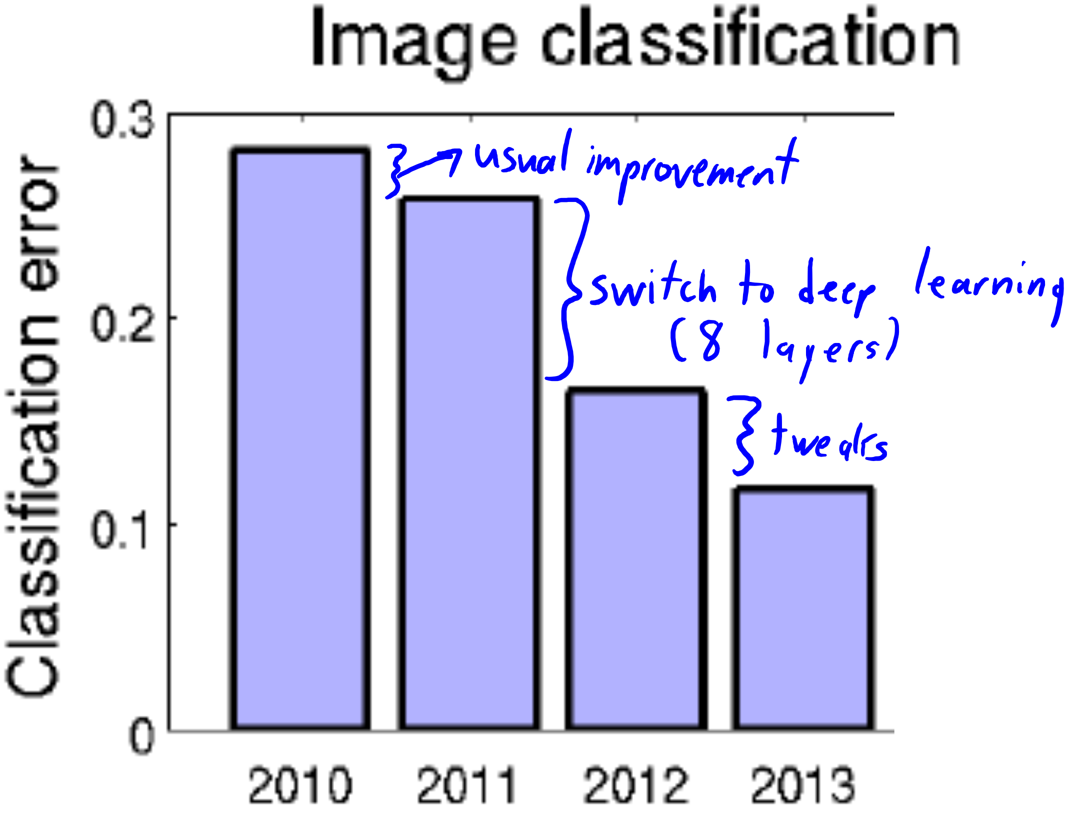


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

(a) Siberian husky (b) Eskimo dog

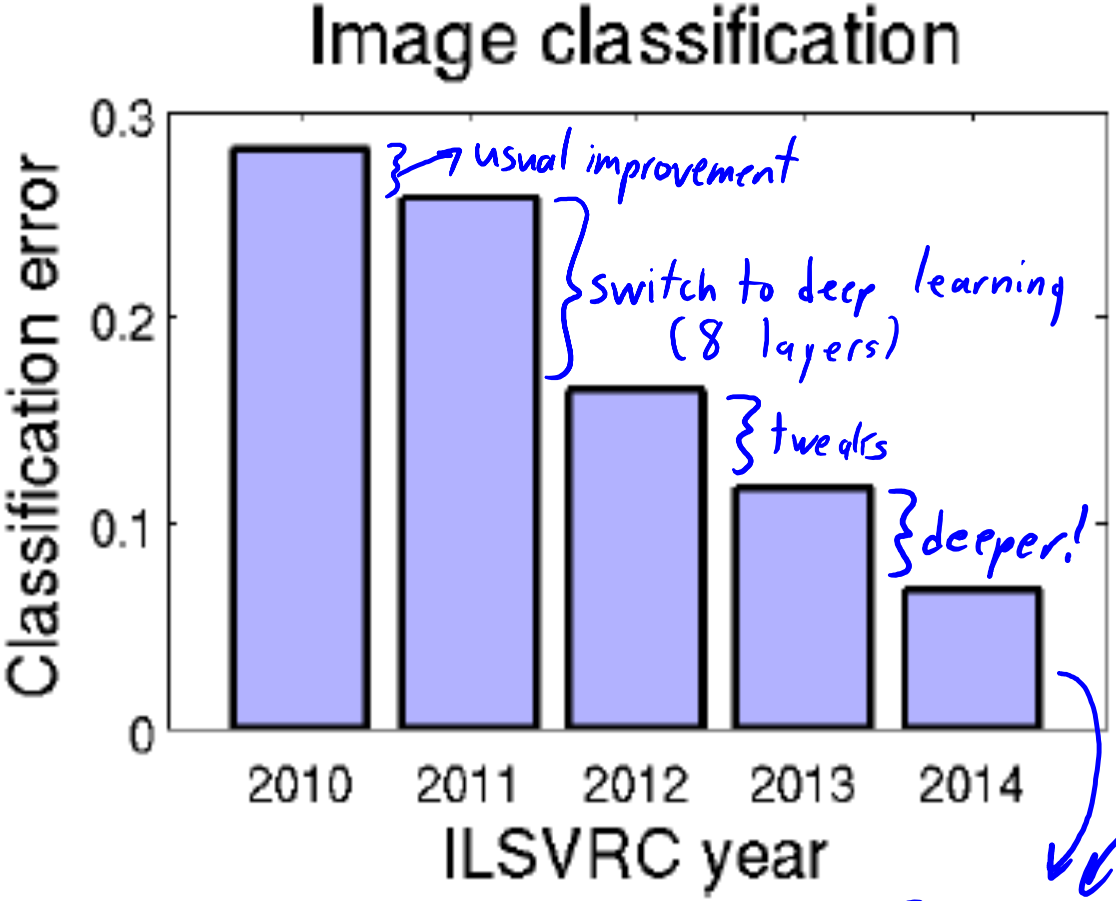


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

(a) Siberian husky (b) Eskimo dog



GoogLeNet:
6.7% error
22 layers

https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

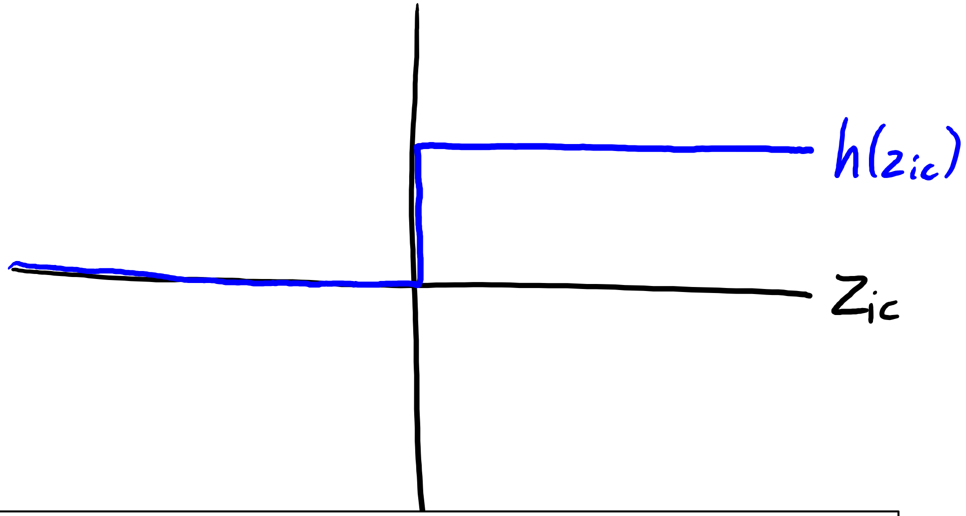
ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.
- 2015: Won by Microsoft Research Asia
  - 3.6% error.
  - 152 layers.
- 2016: Chinese University of Hong Kong:
  - Ensembles of existing methods.
- 2017: fewer entries, organizers decided this would be last year.

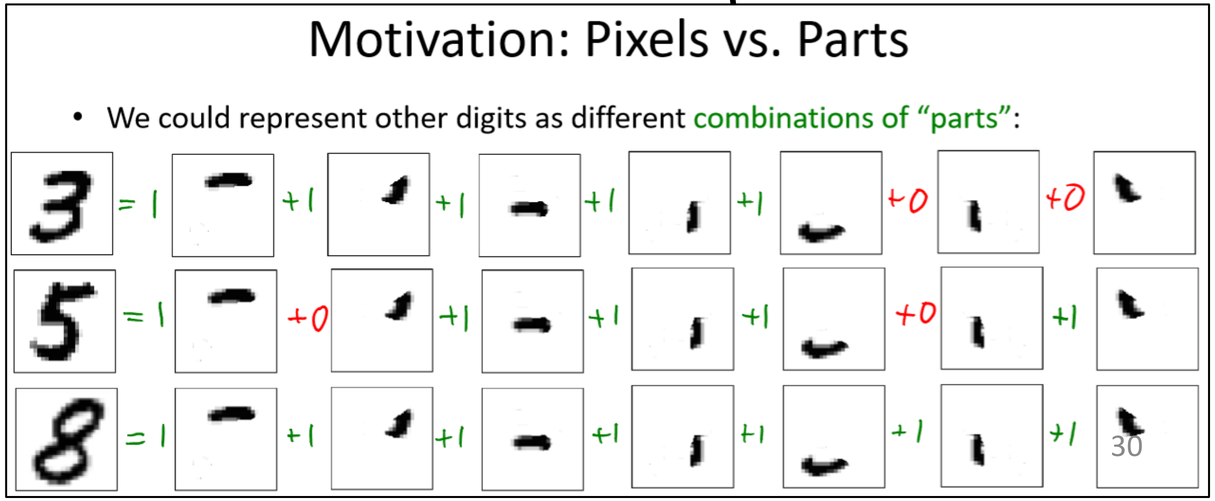
Why Sigmoid?

- Consider setting 'h' to define binary features z_i using:

$$h(z_{ic}) = \begin{cases} 1 & \text{if } z_{ic} \geq 0 \\ 0 & \text{if } z_{ic} < 0 \end{cases}$$



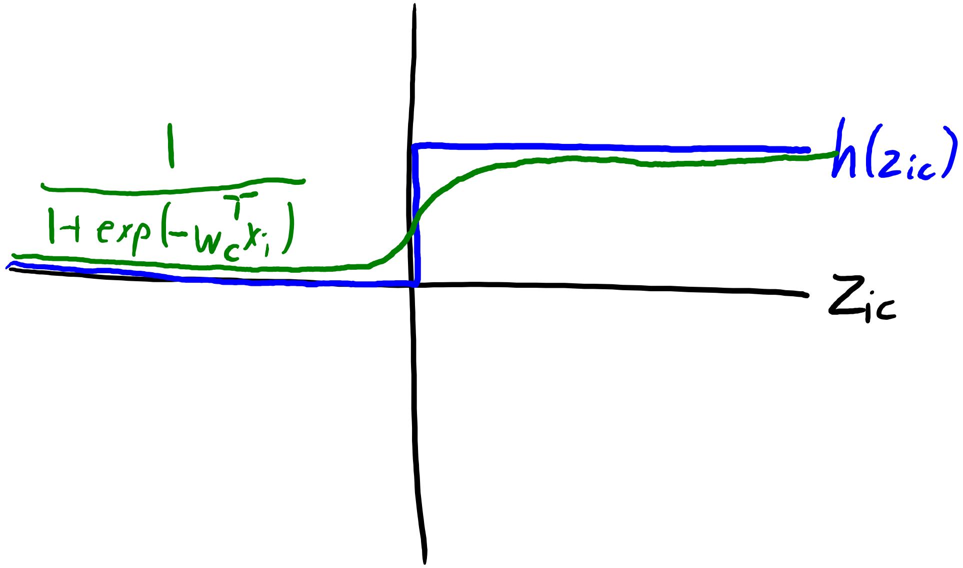
  - Each h(zi) can be viewed as binary feature.
    - "You either have this 'part' or you don't have it."
  - We can make 2^k objects by all the possible "part combinations".



Why Sigmoid?

- Consider setting 'h' to define binary features z_i using:

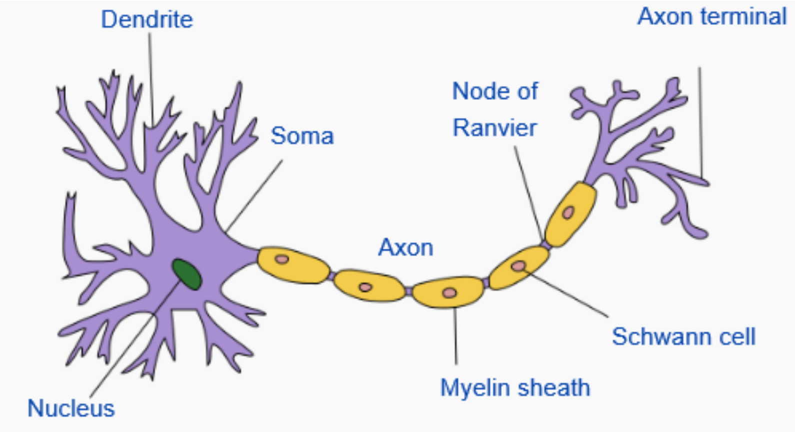
$$h(z_{ic}) = \begin{cases} 1 & \text{if } z_{ic} \geq 0 \\ 0 & \text{if } z_{ic} < 0 \end{cases}$$



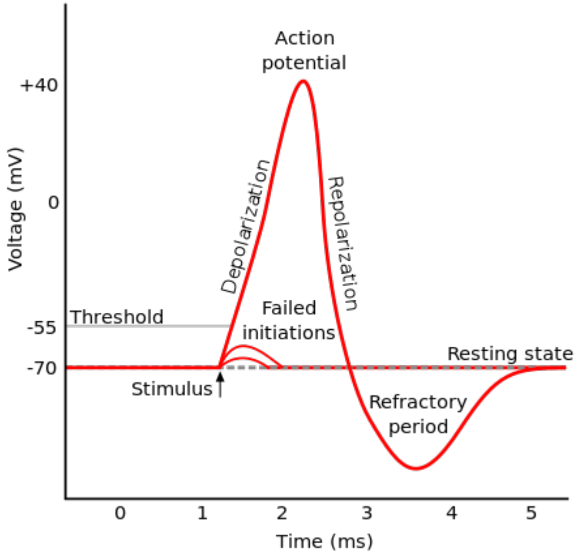
  - Each h(zi) can be viewed as binary feature.
    - "You either have this 'part' or you don't have it."
  - We can make 2^k objects by all the possible "part combinations".
- But this is hard to optimize (non-differentiable/discontinuous).
- Sigmoid is a smooth approximation to these binary features.

Why "Neural Network"?

- Cartoon of "typical" neuron:

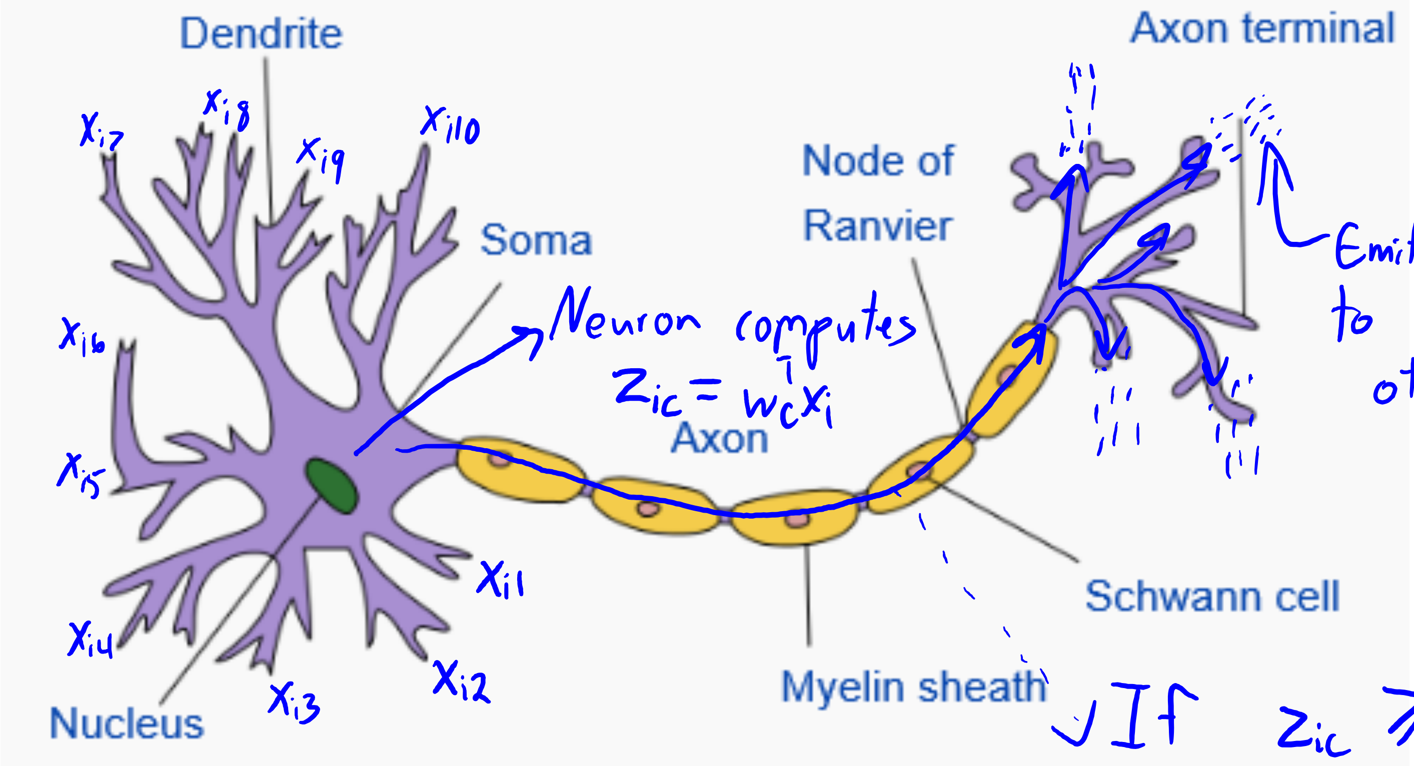


- Neuron has many "dendrites", which take an input signal.
- Neuron has a single "axon", which sends an output signal.
- With the right input to dendrites:
  – "Action potential" along axon (like a binary signal):



https://en.wikipedia.org/wiki/Neuron
https://en.wikipedia.org/wiki/Action_potential

Why "Neural Network"?



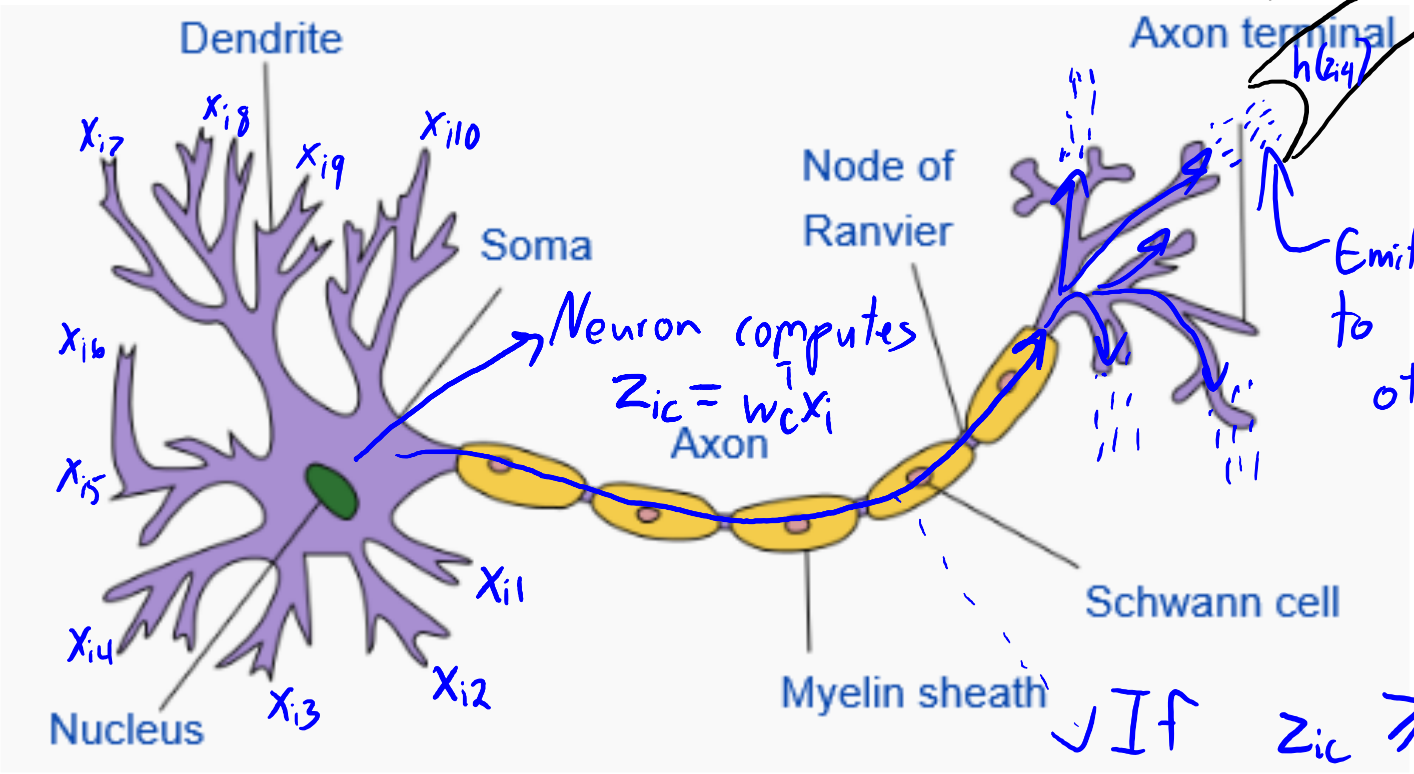
Emits "neurotransmitter" to send signal to other neurons.

If $z_{ic} \geq 0$ neuron sends signal along axon.

We approximate binary signal with $\frac{1}{1+\exp(-z_{ic})}$

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

Why "Neural Network"?

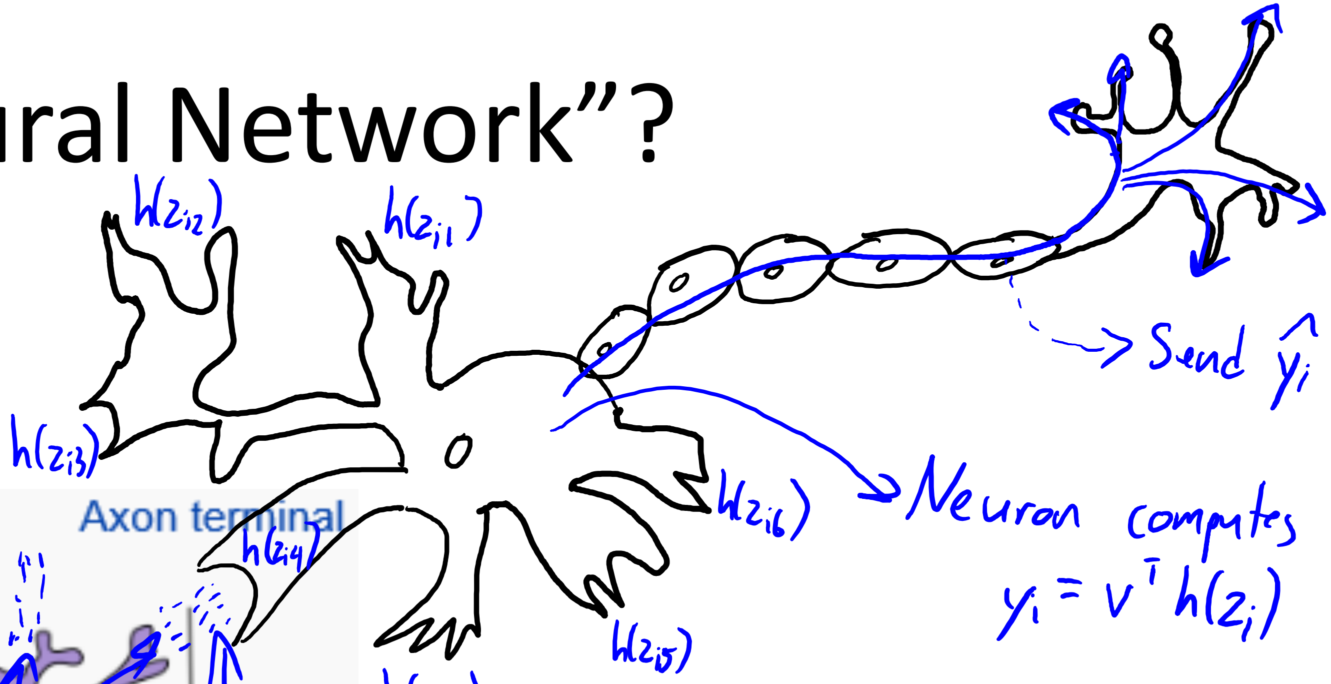


Neuron computes $z_{ic} = w_c^T x_i$

Emits "neurotransmitter" to send signal to other neurons.

If $z_{ic} \geq 0$ neuron sends signal along axon.

We approximate binary signal with $\frac{1}{1+\exp(-z_{ic})}$

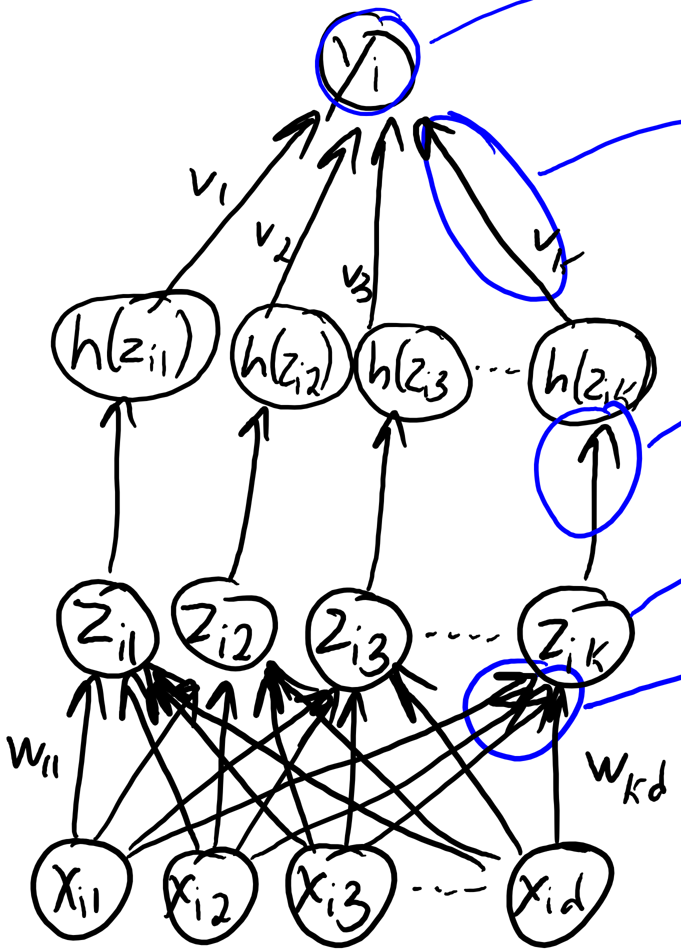


Send $\hat{y}_i$

Neuron computes $y_i = v^T h(z_i)$

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

Why "Neural Network"?



Predictions based on aggregation $v^T h(Wx_i)$ at y_i "neuron"

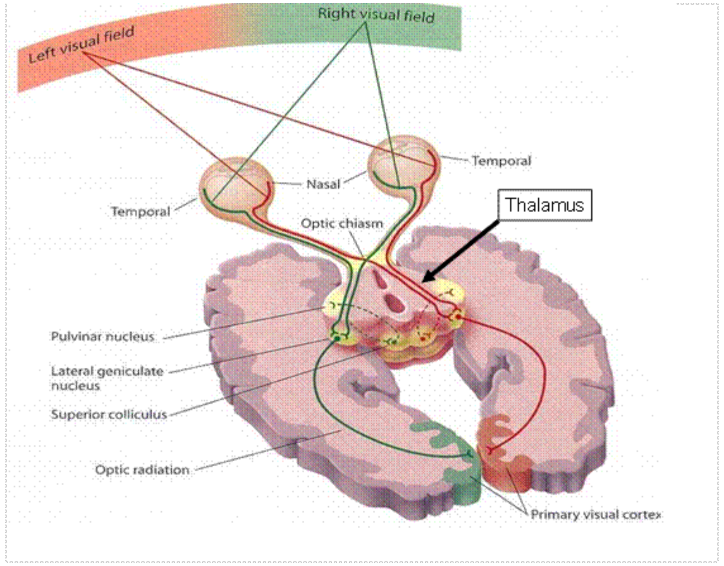
Synapse between z_{ik} and y_i "neuron"

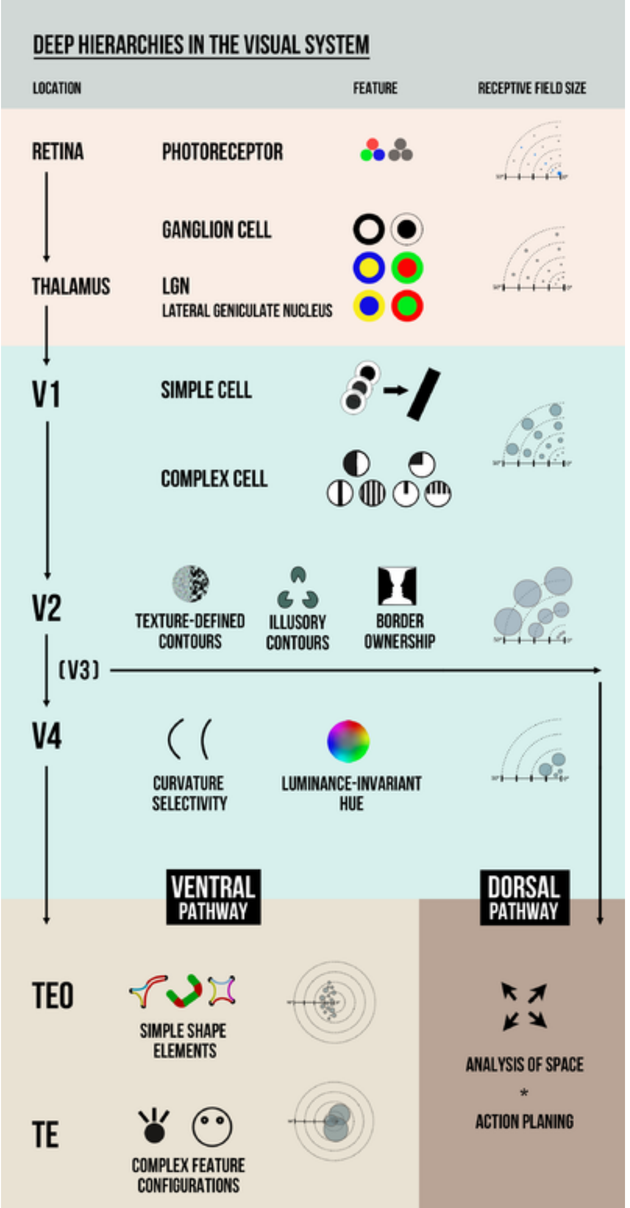
binary signal $h(w_c^T x_i)$ sent along "axon"

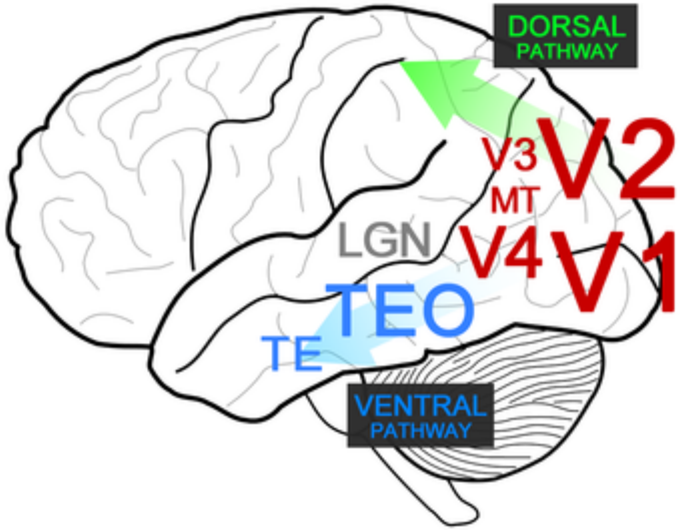
Neuron aggregates signals: $w_c^T x_i$

"dendrites" for z_{ik} "neuron" are reciving x_{ij} values

Deep Hierarchies in the Brain



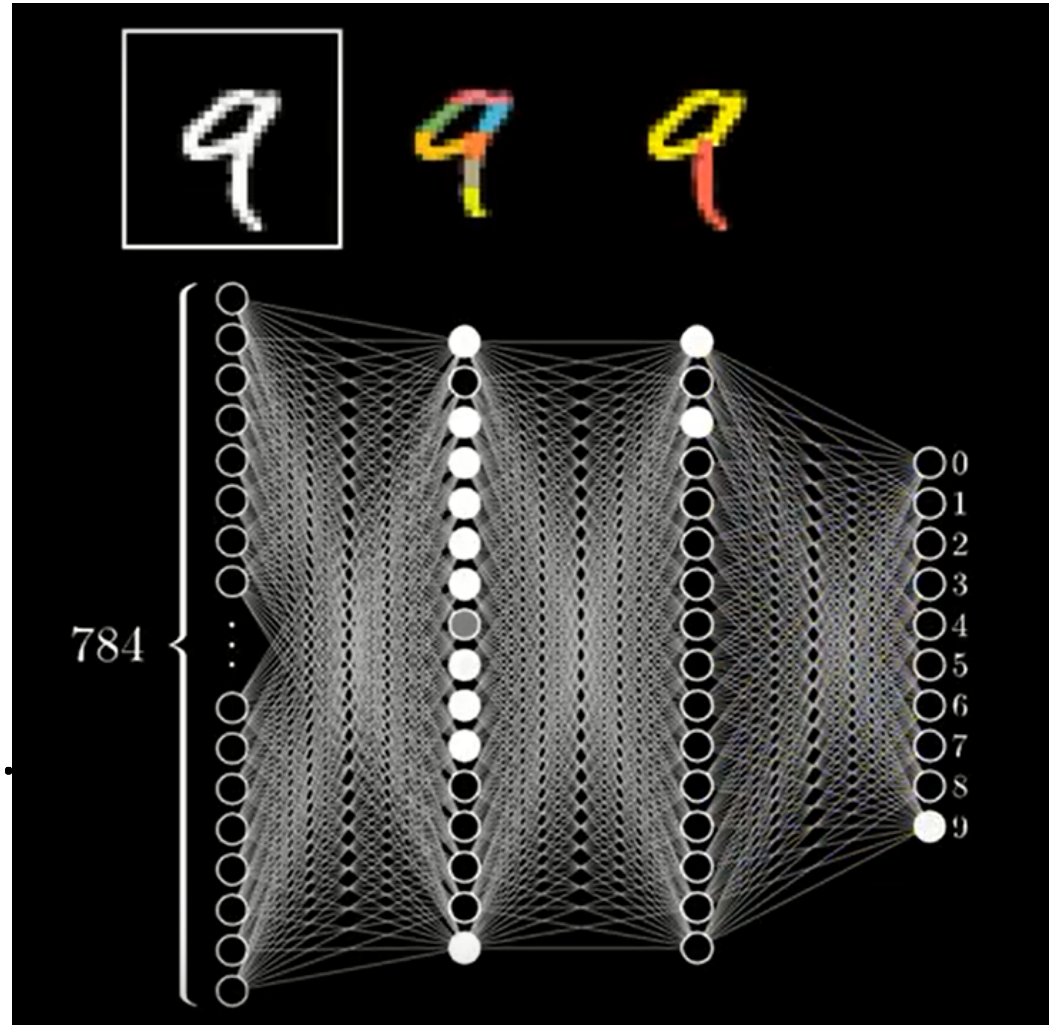




http://www.strokenetwork.org/newsletter/articles/vision.htm
https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing

"Hierarchies of Parts" Motivation for Deep Learning

- Each "neuron" might recognize a "part" of a digit.
  - "Deeper" neurons might recognize combinations of parts.
  - Represent complex objects as hierarchical combinations of re-useable parts (a simple "grammar").



- Watch the full video here:
  - https://www.youtube.com/watch?v=aircAruvnKk

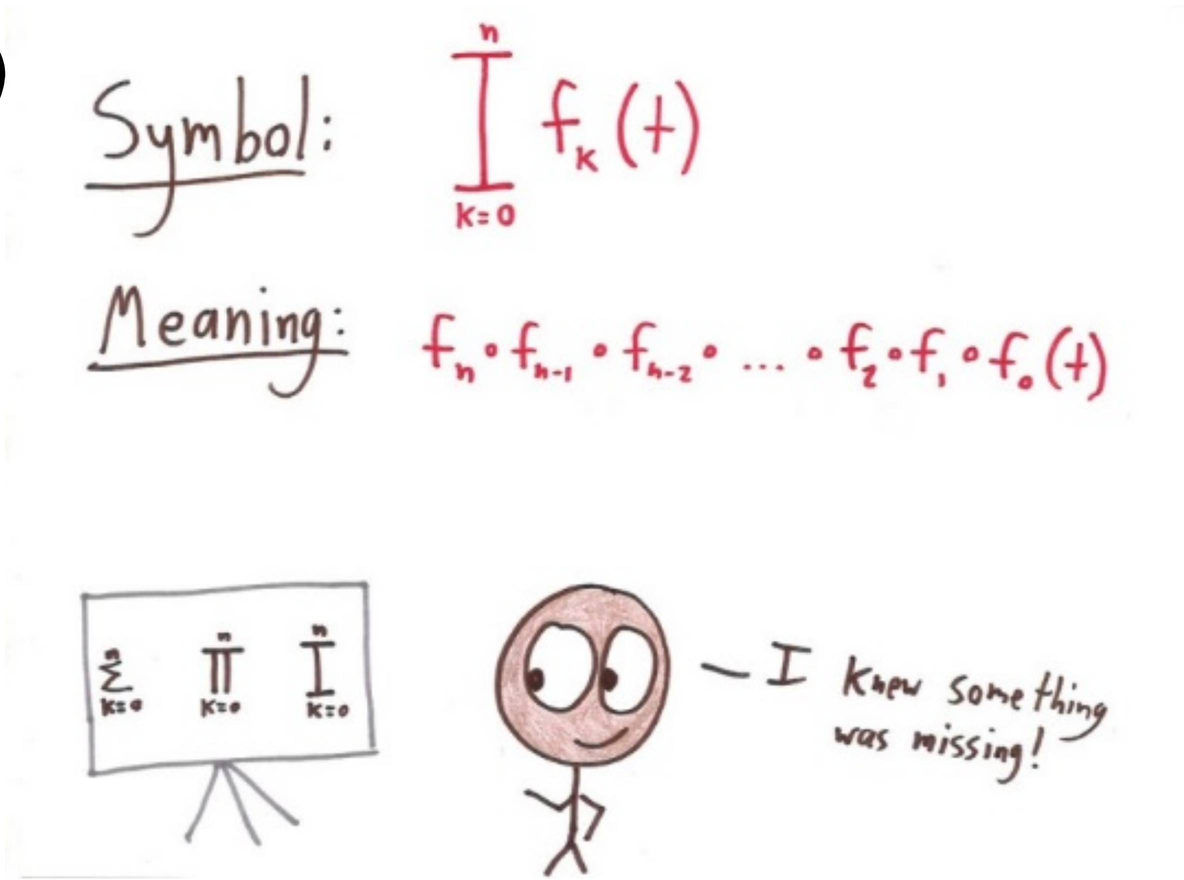
Deep Learning

- For 4 layers, we could write the prediction as:

$$\hat{y}_i = v^T h(W^{(4)} h(W^{(3)} h(W^{(2)} h(W^{(1)} x_i))))$$

- For 'm' layers, we could use:

$$\hat{y}_i = w^T \left(\mathop{\mathrm{I}}_{\ell=1}^{m} h(W^{(\ell)} x_i) \right)$$



https://mathwithbaddrawings.com/2016/04/27/symbols-that-math-urgently-needs-to-adopt

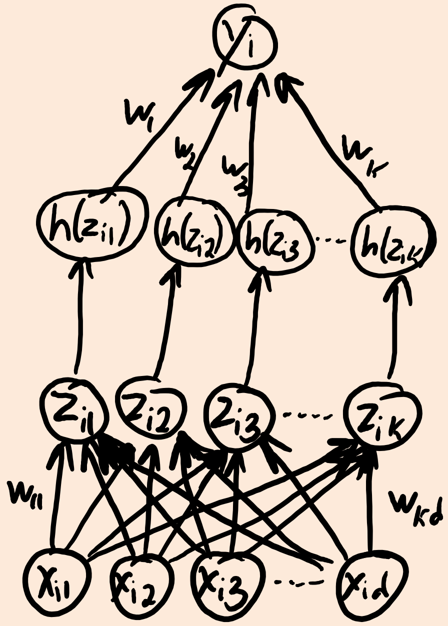
Why $z_i = Wx_i$?

- In PCA we had that the optimal $Z = XW^T(WW^T)^{-1}$.
- If W had normalized+orthogonal rows, $Z = XW^T$ (since $WW^T = I$).
  - So $z_i = Wx_i$ in this normalized+orthogonal case.

- Why we would use $z_i = Wx_i$ in neural networks?
  - We didn't enforce normalization or orthogonality.

- The value $W^T(WW^T)^{-1}$ is just "some matrix".
  - You can think of neural networks as just directly learning this matrix.

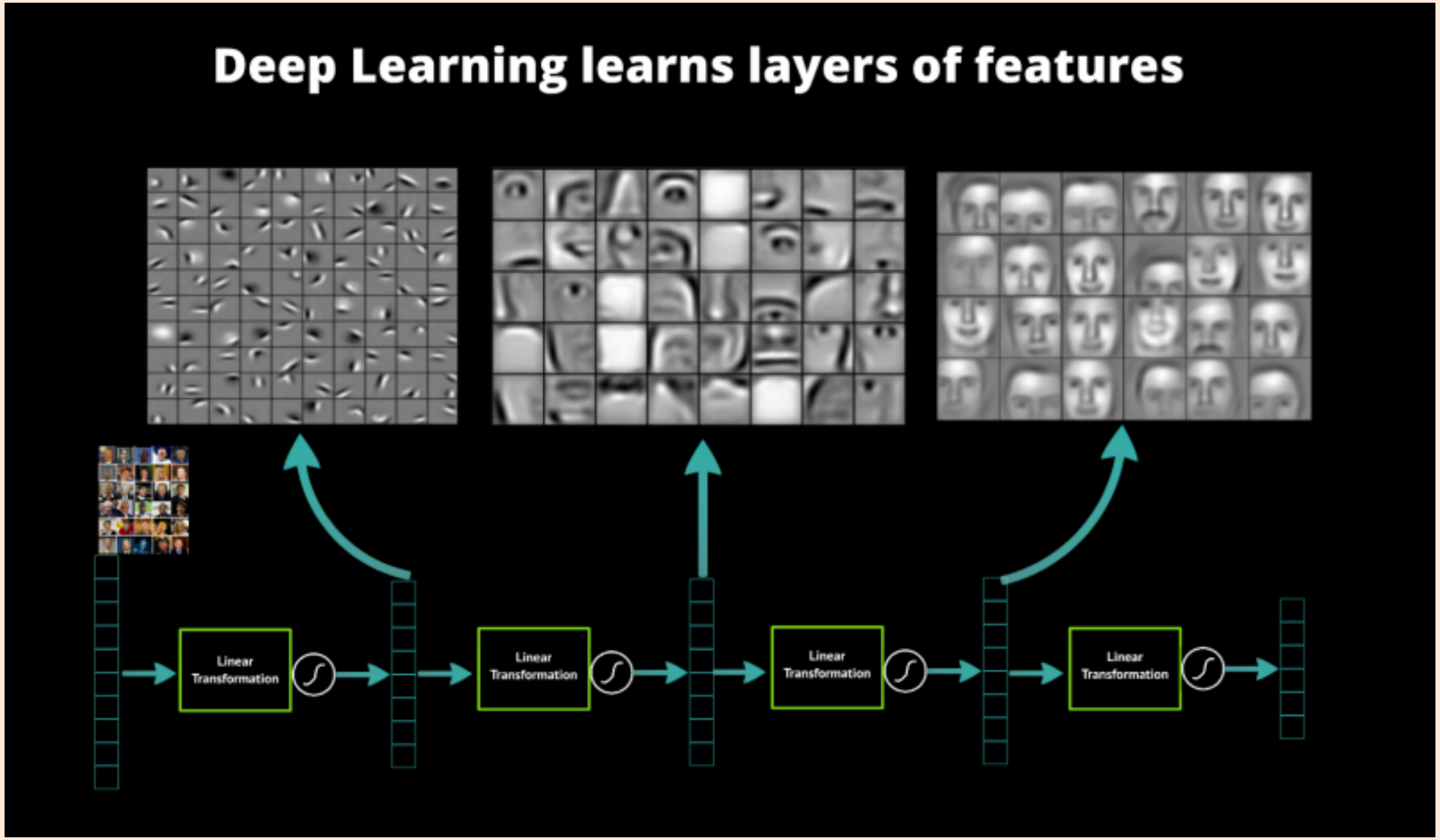
“Artificial” Neural Nets vs. “Real” Networks Nets



- Artificial neural network:
  - x_i is measurement of the world.
  - z_i is internal representation of world.
  - y_i is output of neuron for classification/regression.
- Real neural networks are more complicated:
  - Timing of action potentials seems to be important.
    - “Rate coding”: frequency of action potentials simulates continuous output.
  - Neural networks don’t reflect sparsity of action potentials.
  - How much computation is done inside neuron?
  - Brain is highly organized (e.g., substructures and cortical columns).
  - Connection structure changes.
  - Different types of neurotransmitters.

Cool Picture Motivation for Deep Learning

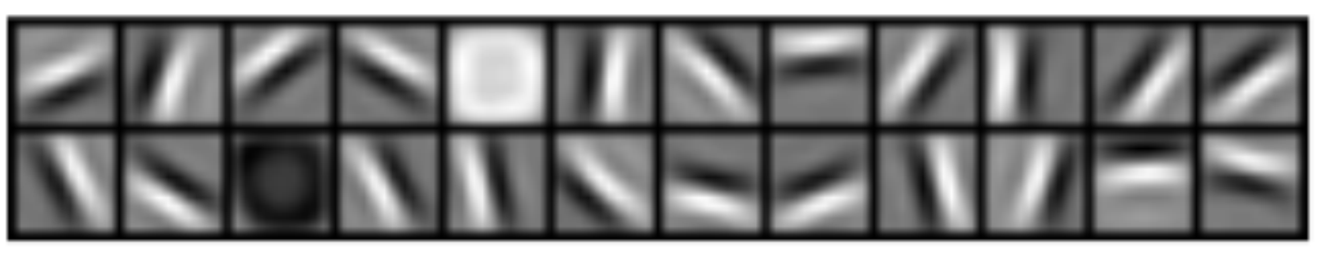
- Faces might be composed of different “parts”:



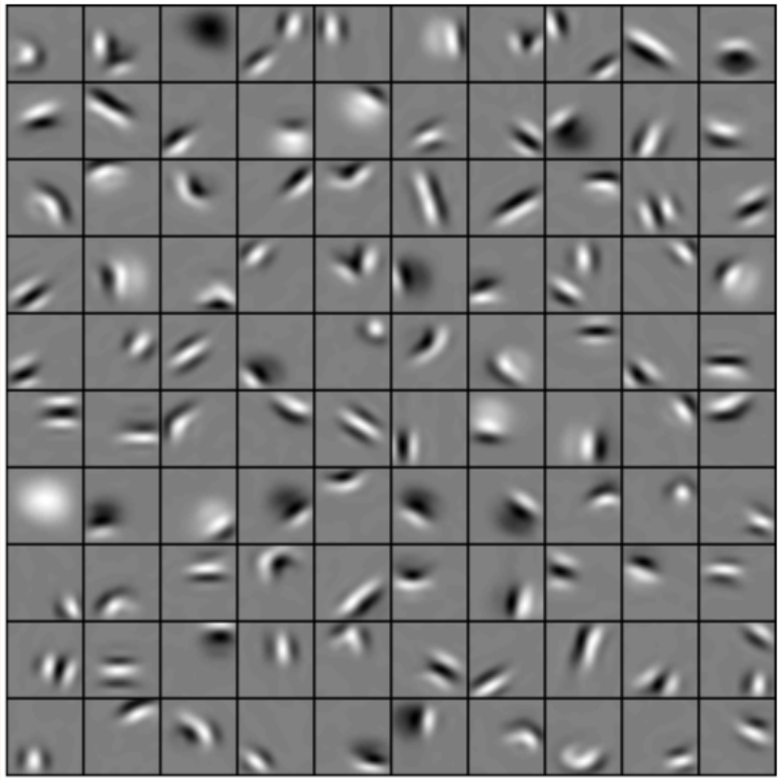
http://www.datarobot.com/blog/a-primer-on-deep-learning/

Cool Picture Motivation for Deep Learning

- First layer of z_i trained on 10 by 10 image patches:



"Gabor filters"

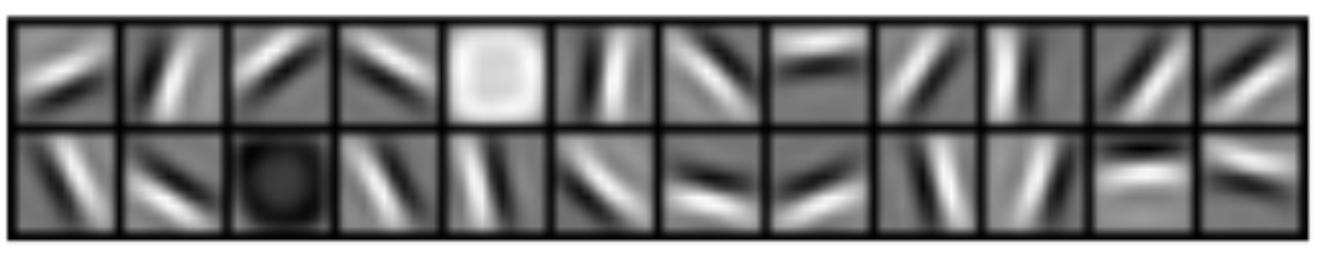


- Attempt to visualize second layer:
  - Corners, angles, surface boundaries?
- Models require many tricks to work.
  - We'll discuss these next time.

http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf

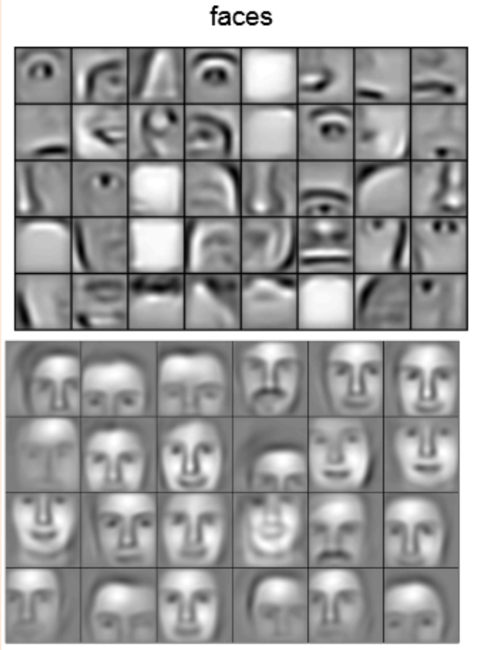
Cool Picture Motivation for Deep Learning

- First layer of z_i trained on 10 by 10 image patches:



"Gabor filters"

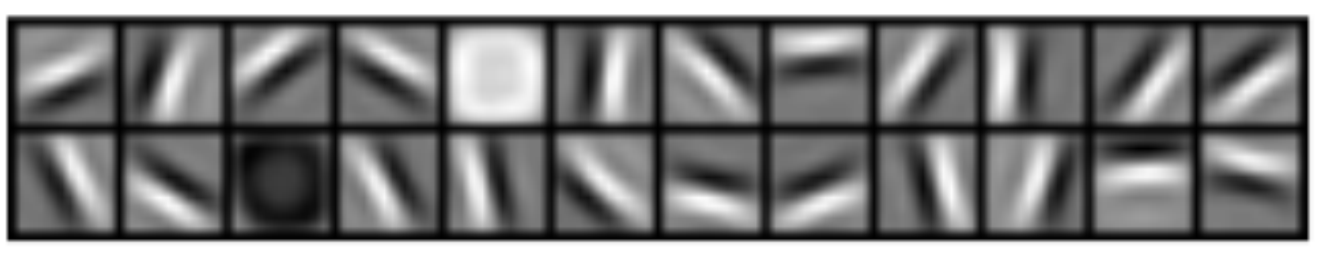
- Visualization of second and third layers trained on specific objects:



http://www.cs.toronto.edu/~rgrosse

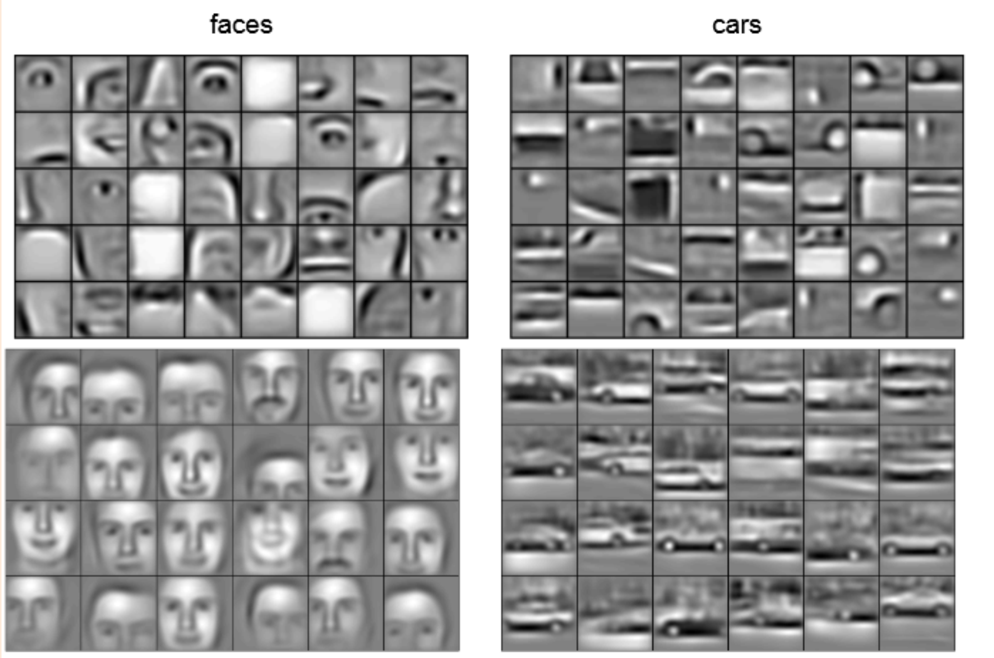
Cool Picture Motivation for Deep Learning

- First layer of z_i trained on 10 by 10 image patches:



"Gabor filters"

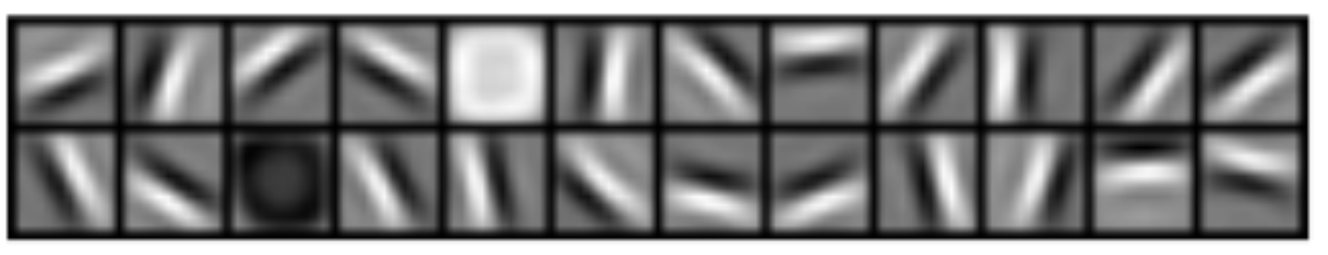
- Visualization of second and third layers trained on specific objects:



http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf

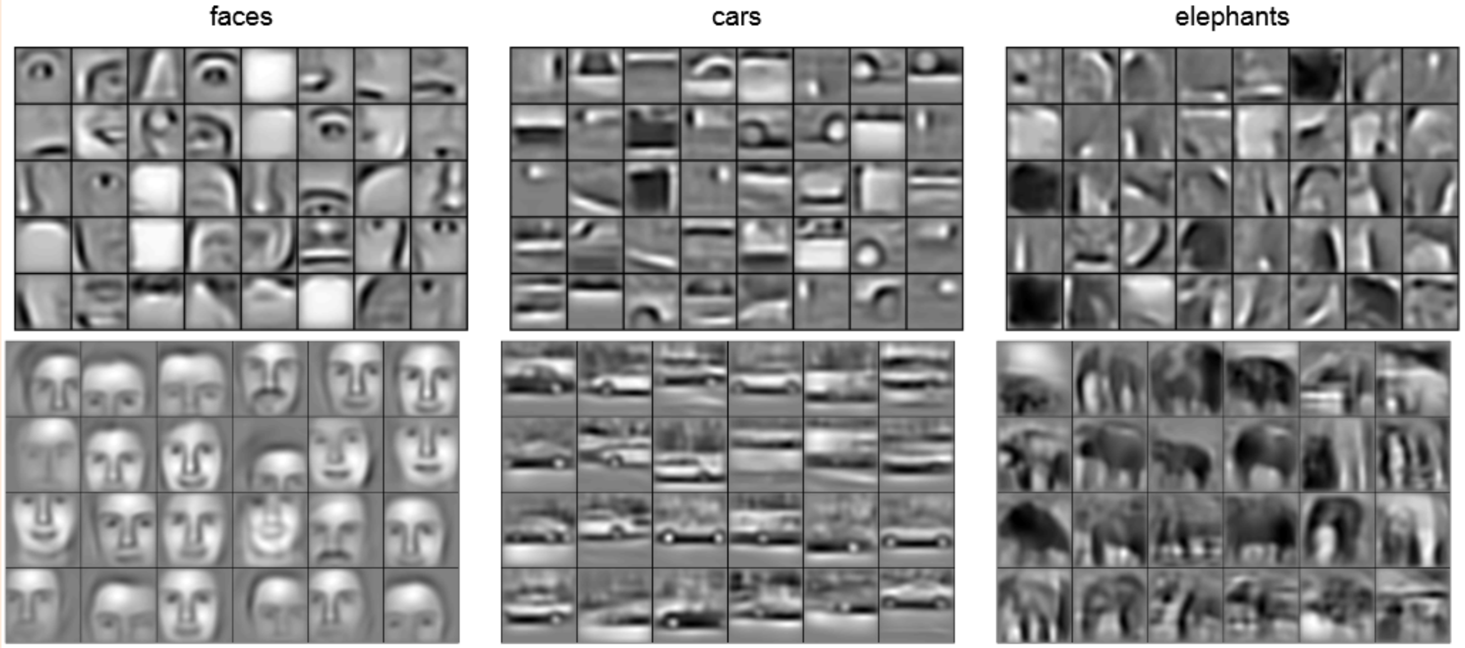
Cool Picture Motivation for Deep Learning

- First layer of z_i trained on 10 by 10 image patches:



"Gabor filters"

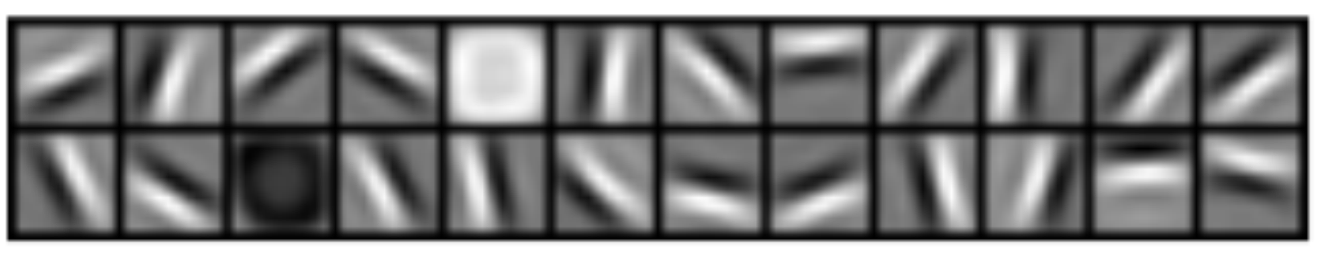
- Visualization of second and third layers trained on specific objects:



http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf

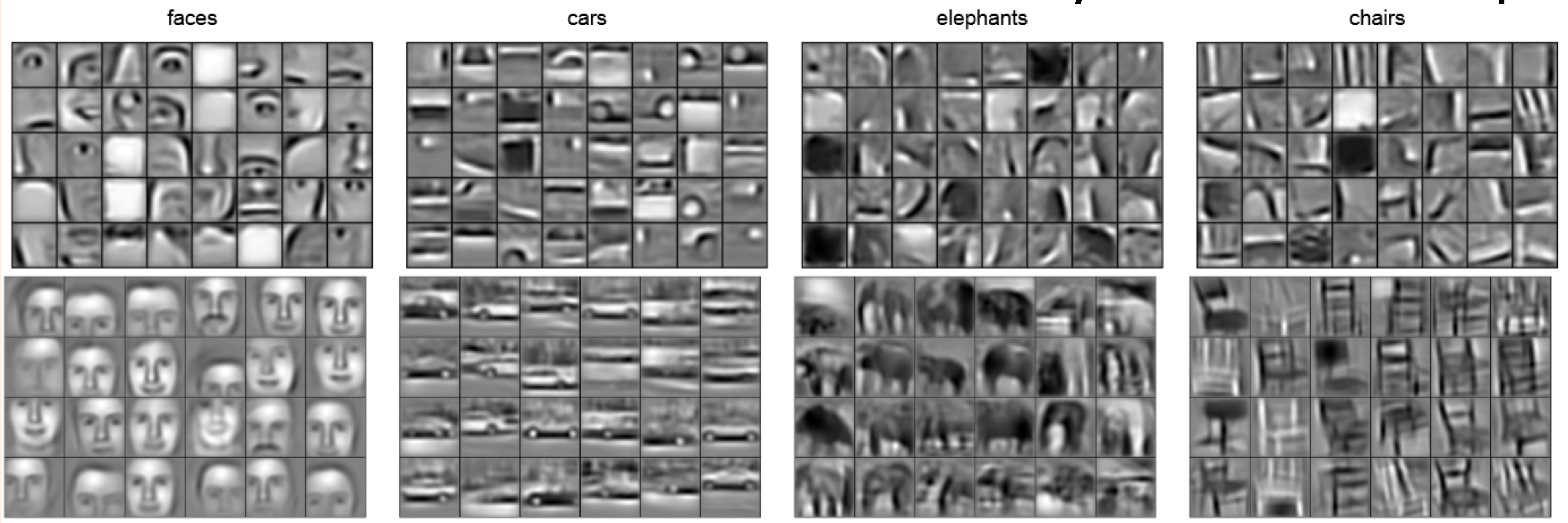
Cool Picture Motivation for Deep Learning

- First layer of z_i trained on 10 by 10 image patches:



"Gabor filters"

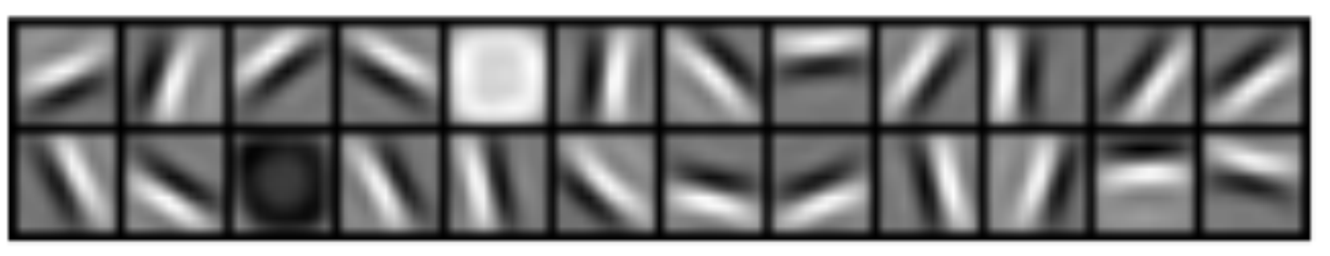
- Visualization of second and third layers trained on specific objects:



http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf

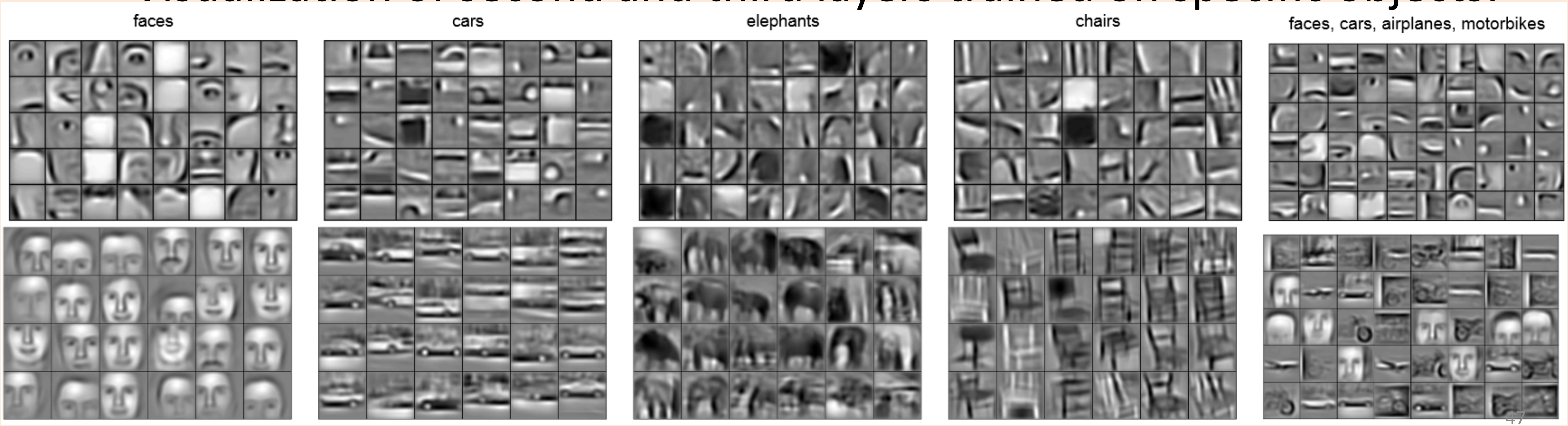
Cool Picture Motivation for Deep Learning

- First layer of z_i trained on 10 by 10 image patches:



"Gabor filters"

- Visualization of second and third layers trained on specific objects:



http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf