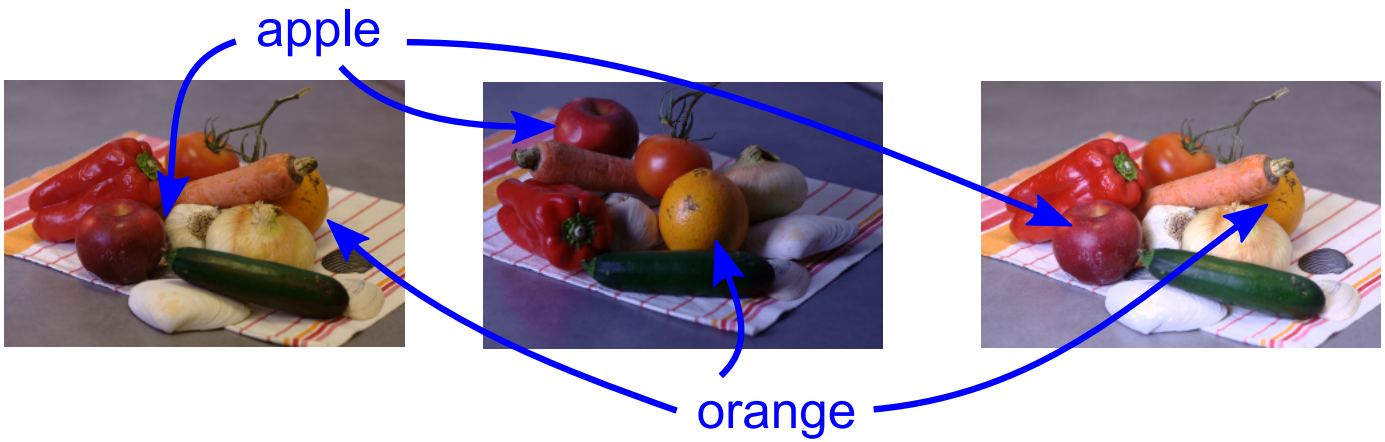
# Lecture 14: Convolutional neural networks for computer vision

Dr. Richard E. Turner (ret26@cam.ac.uk)

November 20, 2014

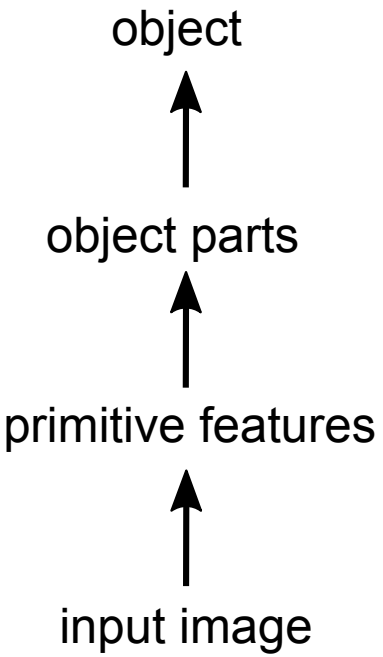
# Big picture

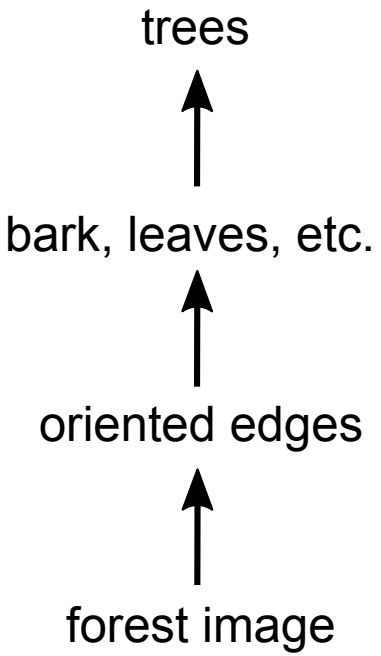
- **Goal**: how to produce good internal representations of the visual world to support recognition...
  - detect and classify objects into categories, independently of pose, scale, illumination, conformation, occlusion and clutter
- how could an artificial vision system learn appropriate internal representations automatically, the way humans seem to by simply looking at the world?
- **previously in CV and the course**: hand-crafted feature extractor
- **now in CV and the course**: learn suitable representations of images

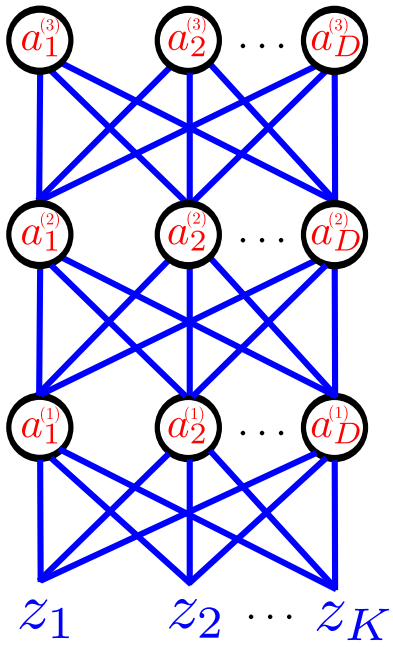


# Why use hierarchical multi-layered models?

Argument 1: visual scenes are hierachically organised

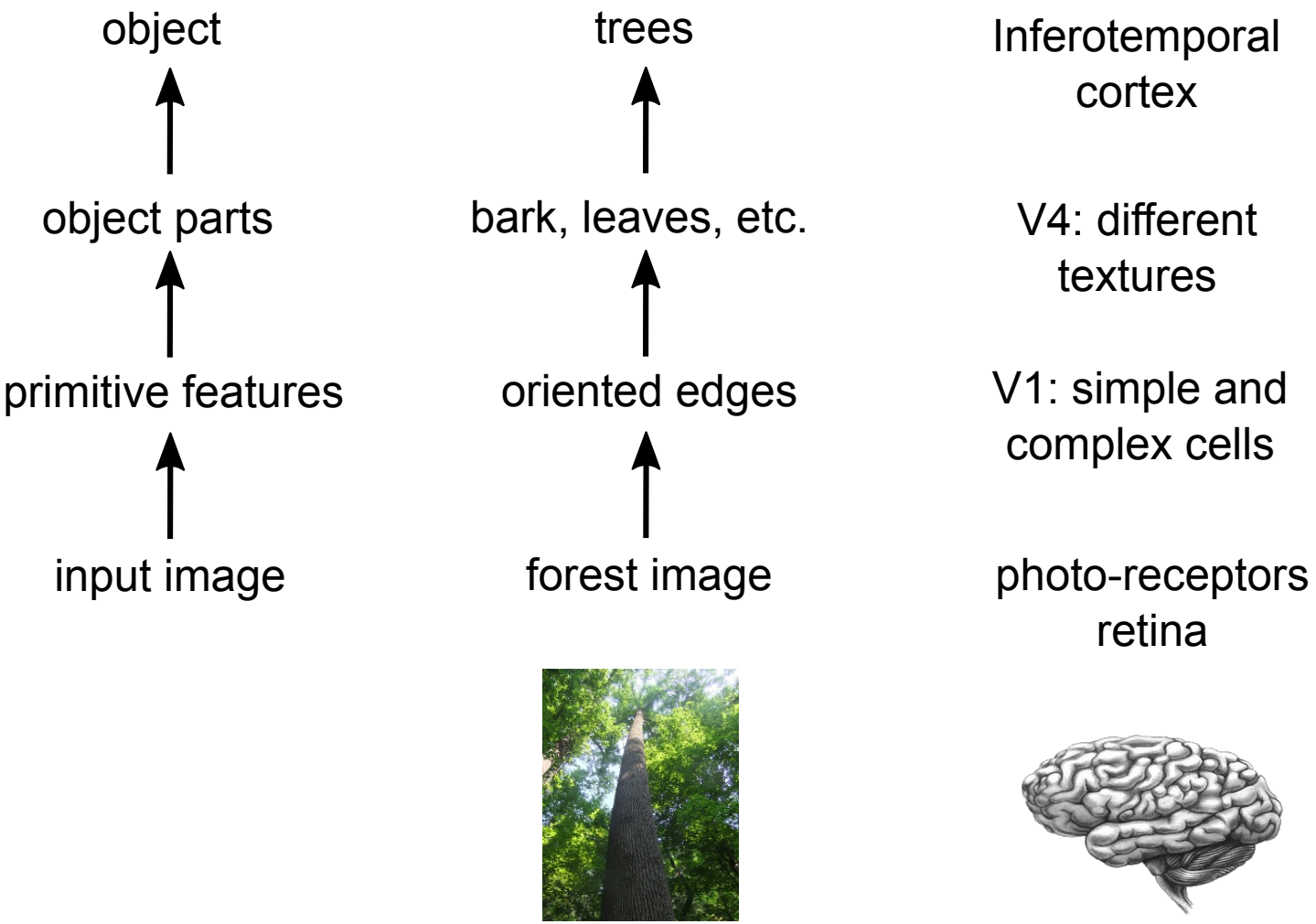



# Why use hierarchical multi-layered models?

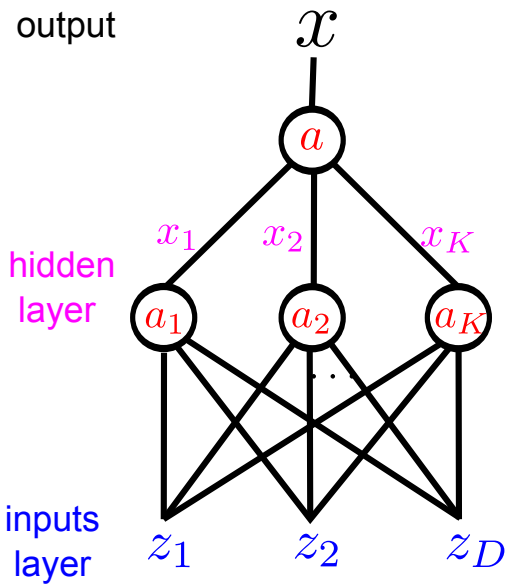
Argument 2: biological vision is hierachically organised



# Why use hierarchical multi-layered models?

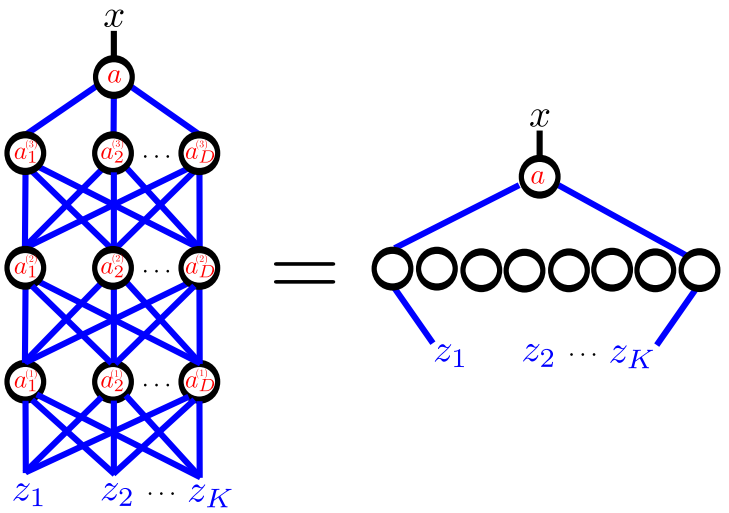
## Argument 3: shallow architectures are inefficient at representing deep functions

single layer neural network
implements: $x = f_\theta(\mathbf{z})$




networks we met last lecture
with large enough single hidden layer
can implement **any** function
'**universal approximator**'

shallow networks can be computationally inefficient




however, if the function is 'deep'
a very large hidden layer may
be required

# What's wrong with standard neural networks?

How many parameters does this neural network have?

$$|\theta| = 3D^2 + D$$

For a small 32 by 32 image:

$$|\theta| = 3 \times 32^4 + 32^2 \approx 3 \times 10^6$$

**Hard to train**

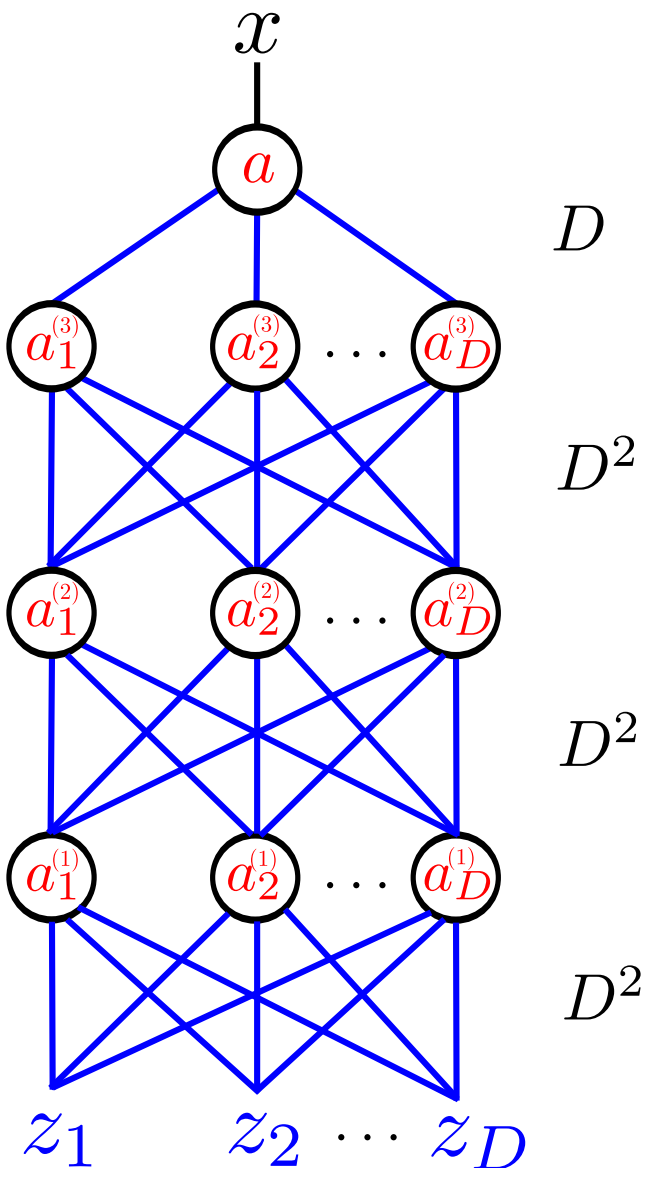
over-fitting and local optima

**Need to initialise carefully**

layer wise training

unsupervised schemes

**Convolutional nets reduce the number of parameters**



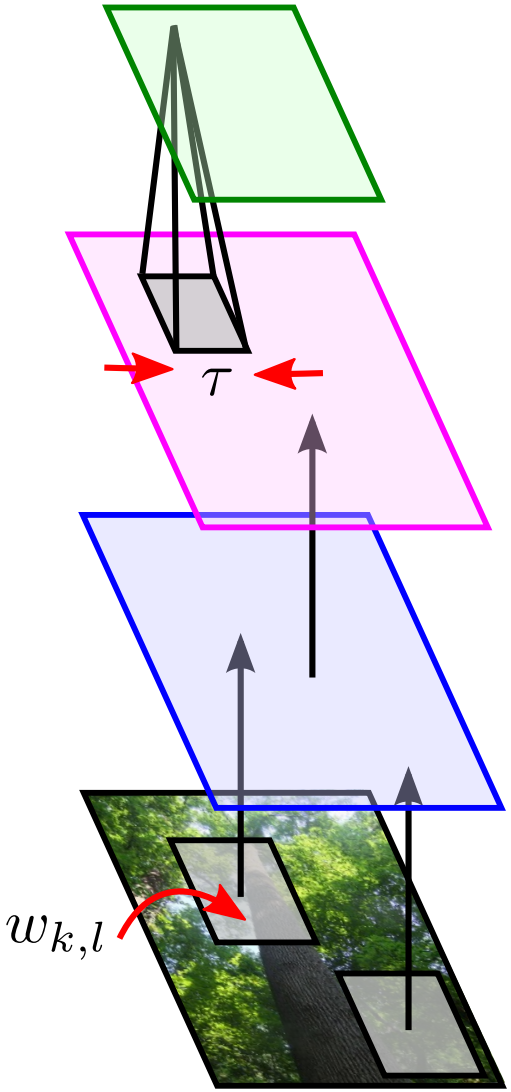
# The key ideas behind convolutional neural networks

- **image statistics are translation invariant** (objects and viewpoint translates)
  - build this translation invariance into the model (rather than learning it)
  - tie lots of the weights together in the network
  - reduces number of parameters

- **expect learned low-level features to be local** (e.g. edge detector)
  - build this into the model by allowing only local connectivity
  - reduces the numbers of parameters further

- **expect high-level features learned to be coarser** (c.f. biology)
  - build this into the model by subsampling more and more up the hierarchy
  - reduces the number of parameters again

# Building block of a convolutional neural network




$$x_{i,j} = \max_{|k|<\tau,|l|<\tau} y_{i-k,j-l}$$

mean or subsample also used

pooling stage

$$y_{i,j} = f(a_{i,j})$$

e.g. $f(a) = [a]_+$

$f(a) = \text{sigmoid}(a)$

non-linear stage

$$a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l}$$

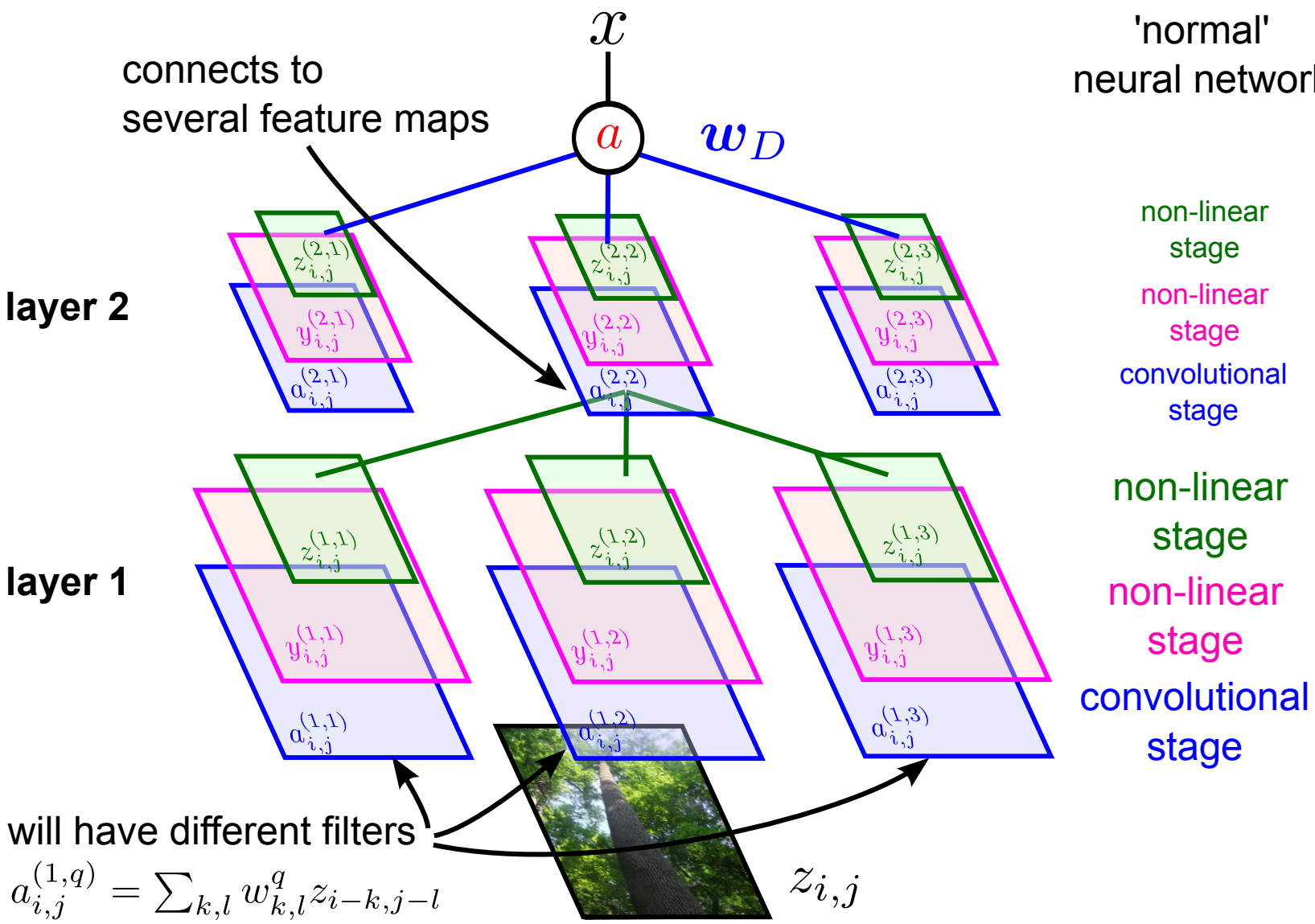
only parameters

convolutional stage

$z_{i,j}$

input image

# Full convolutional neural network



# How many parameters does a convolutional network have?

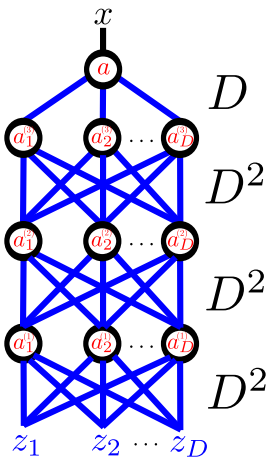
How many parameters does this neural network have?

$$|\theta| = 3K^2 + 9K^2 + 9K^2 + 3(D/S)^2$$
$$= 21K^2 + D$$

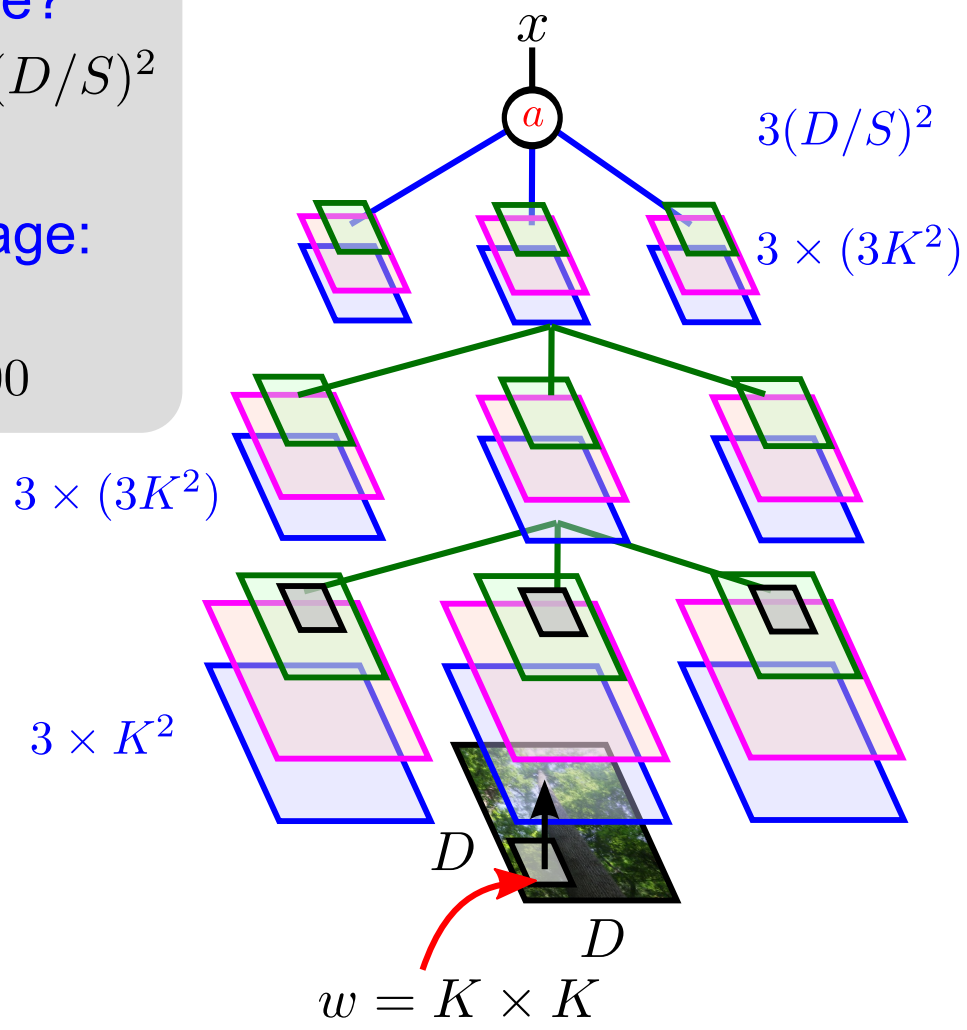
For a small 32 by 32 image:

$$K = 5 \qquad S = 2$$

$$|\theta| = 21 \times 5^2 + 4^2 \approx 600$$




$$|\theta| = 3D^2 + D \approx 3 \times 10^6$$



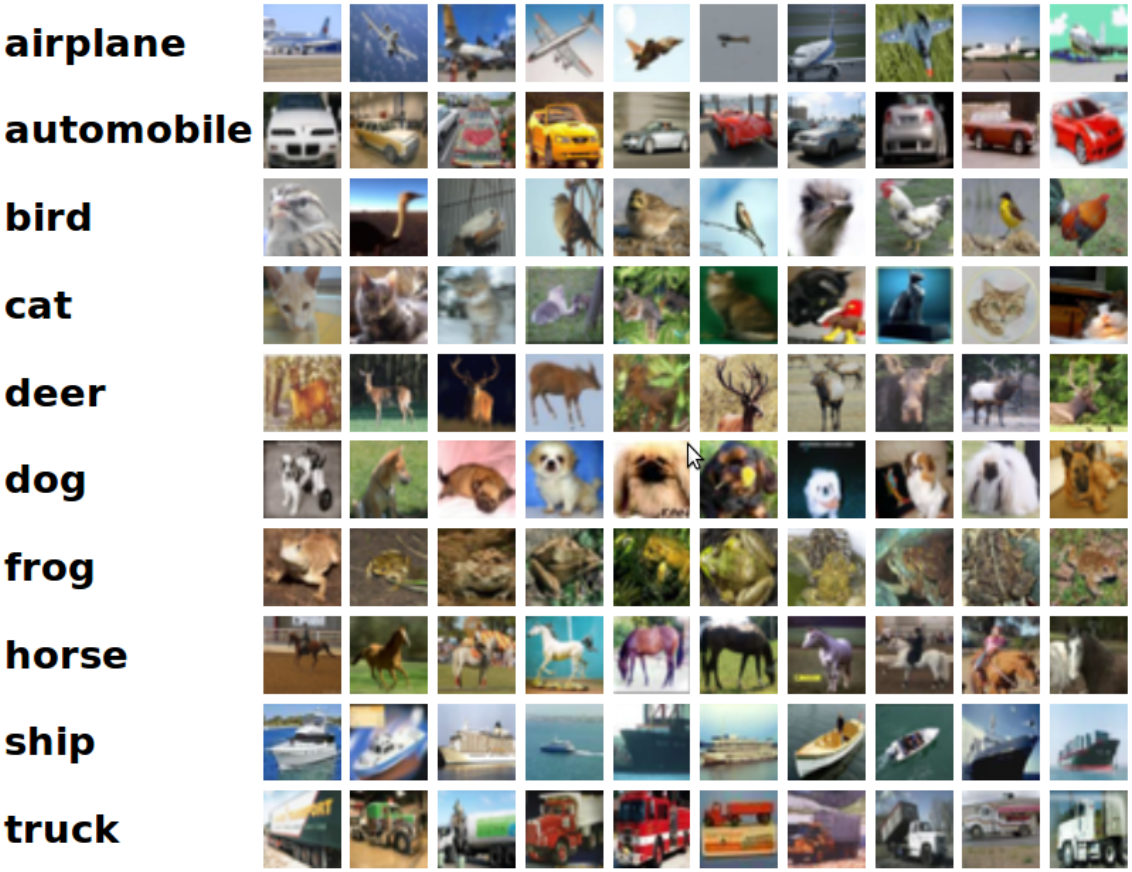
# Training

- **back-propagation for training**: stochastic gradient ascent
  - like last lecture output interpreted as a class label probability, $x = p(t = 1|\boldsymbol{z})$
  - now $x$ is a more complex function of the inputs $\boldsymbol{z}$
  - can optimise same objective function computed over a mini-batch of datapoints

- **data-augmentation**: always improves performance substantially (include shifted, rotations, mirroring, locally distorted versions of the training data)

- **typical numbers**:
  - 5 convolutional layers, 3 layers in top neural network
  - 500,000 neurons
  - 50,000,000 parameters
  - 1 week to train (GPUs)

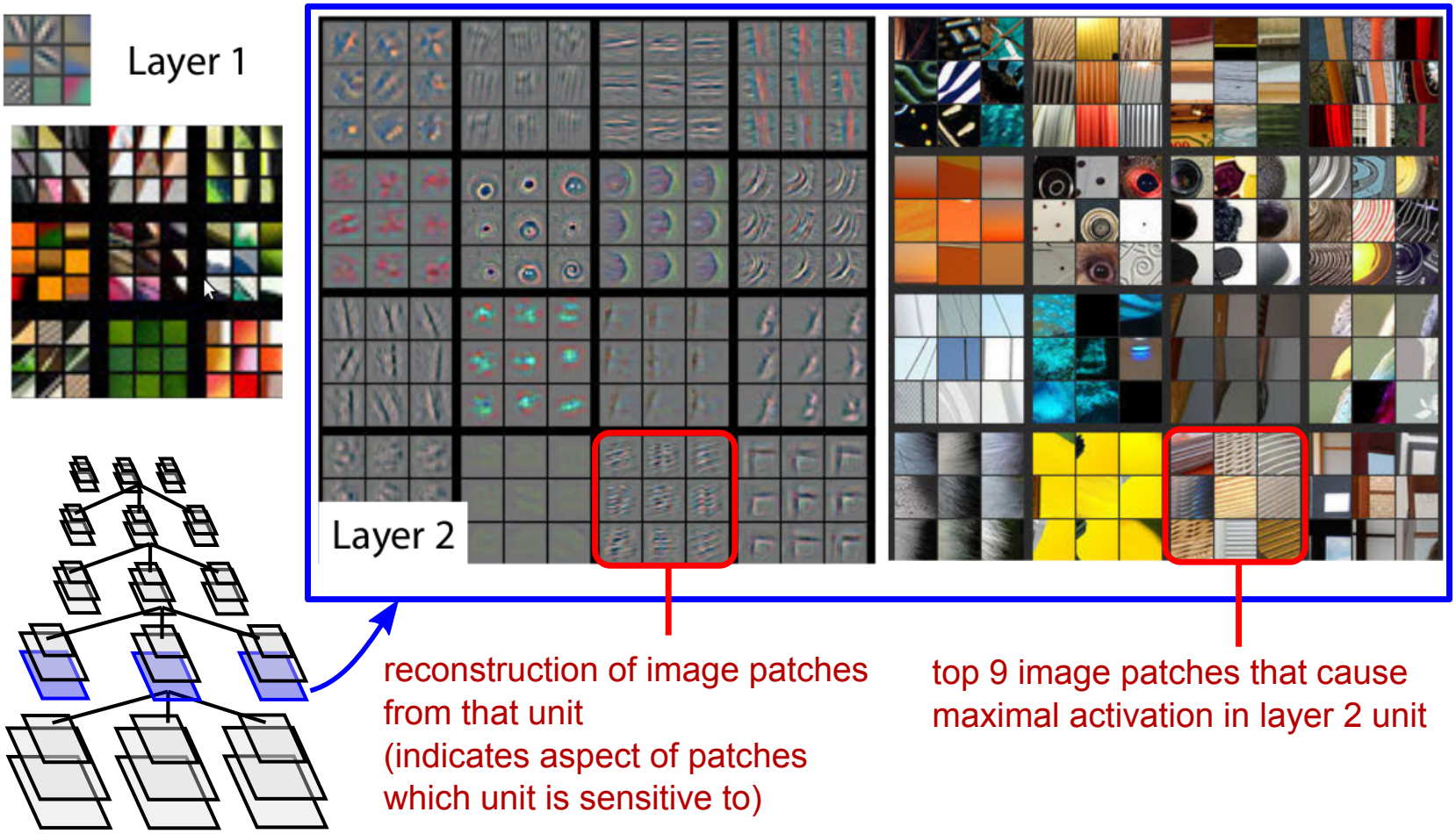
# Demo



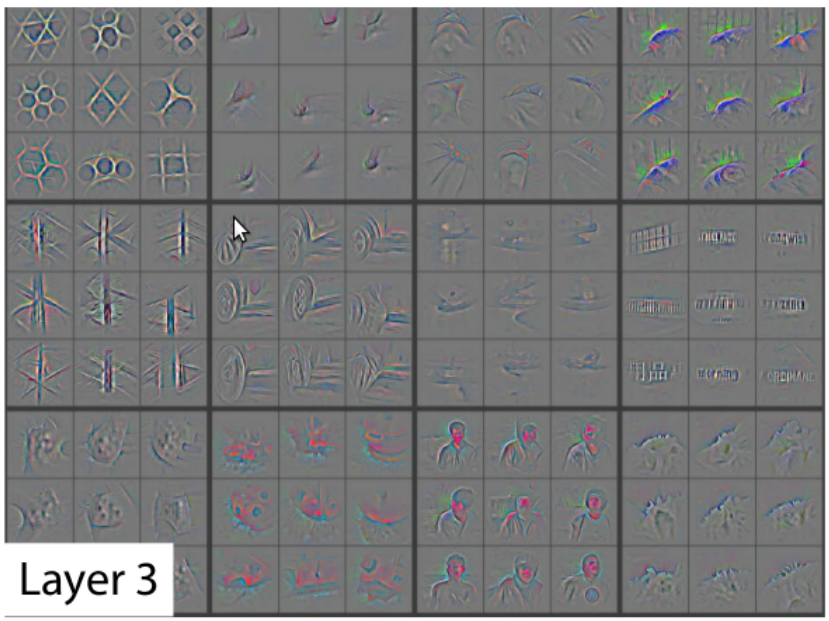

CIFAR 10 dataset: 50,000 training images, 10,000 test images

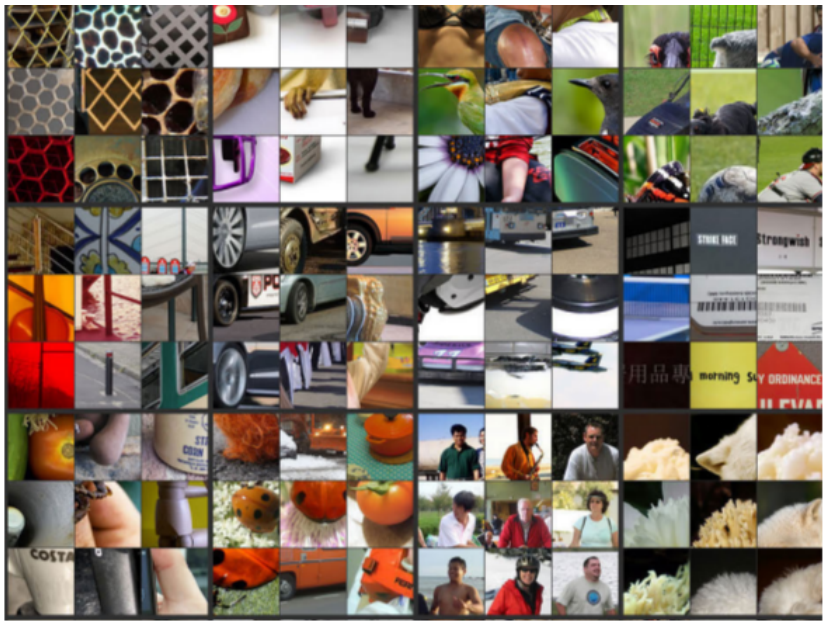
http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

# Looking into a convolutional neural network's brain

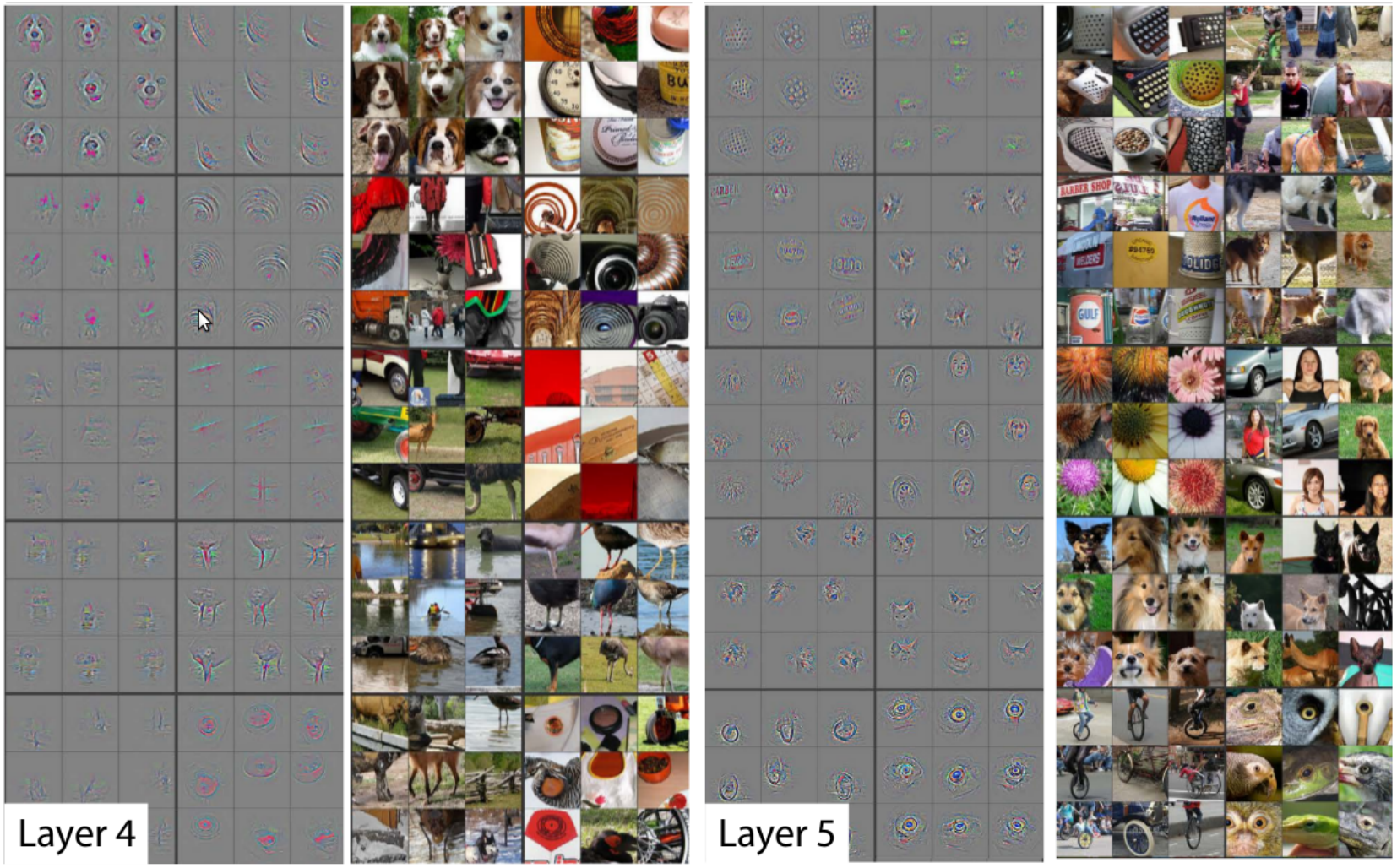


# Looking into a convolutional neural network's brain



# Looking into a convolutional neural network's brain



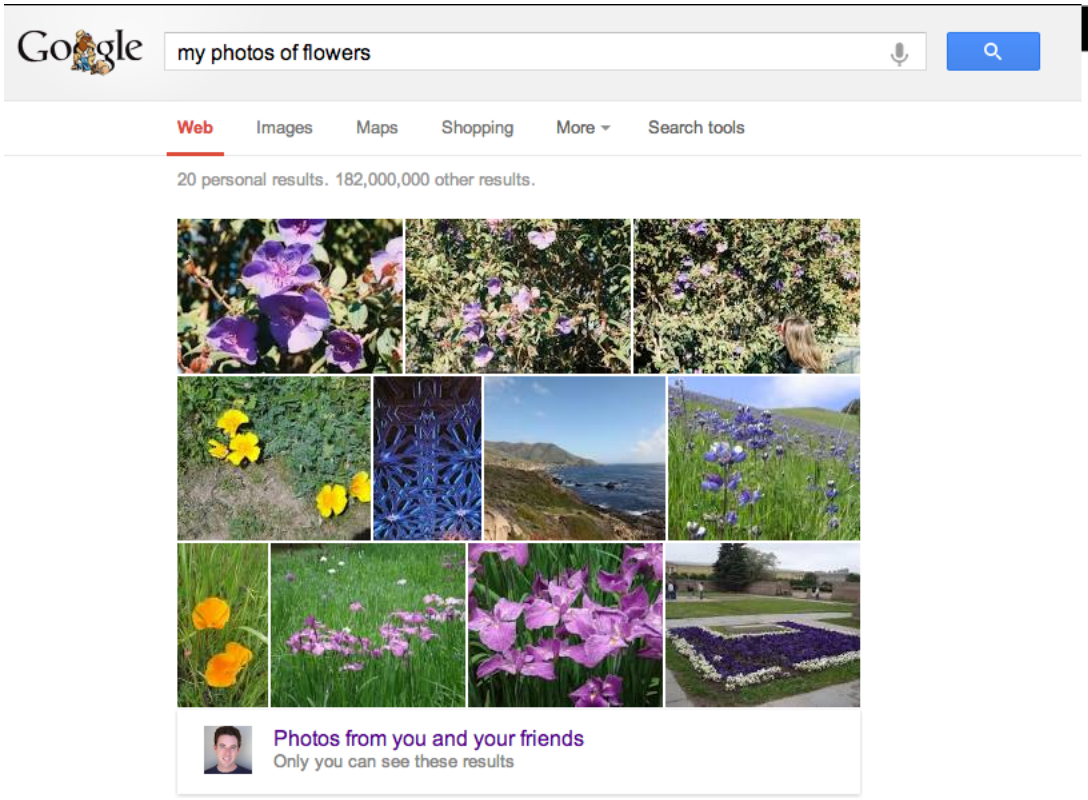
# Summary

- higher level layers encode more **abstract features**
- higher level layers show more **invariance to instantiation parameters**
  - translation
  - rotation
  - lighting changes
- a method for **learning feature detectors**
  - first layer learns edge detectors
  - subsequent layers more complex
  - integrates training of the classifier with training of the featural representation

# Convolutional neural networks in the news

- convolutional neural networks are the go-to model for many computer vision classification problems

- form of neural network with an architecture suited to vision problems




# Finally some cautionary words

- hierarchical modelling is a **very old idea** and not new
- the ‘deep learning’ revolution has come about mainly due to new methods for initialising learning of neural networks
- current methods aim at invariance, but this is far from all there is to computer and biological vision: **e.g. instantiation parameters should also be represented**
- **classification can only go so far**: "tell us a story about what happened in this picture"