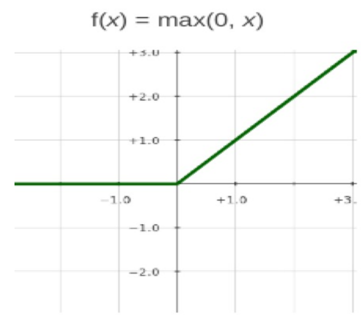
Neural Networks:
Deep Learning (2)

Lyle Ungar

Multilevel network: architecture, link functions
CNNs: local receptive fields, max pooling

Regularization: L_2, early stopping, dropout
Gradient Descent (again + Adgrad)
Semi-supervised and **transfer learning**
Visualization

Modern deep nets



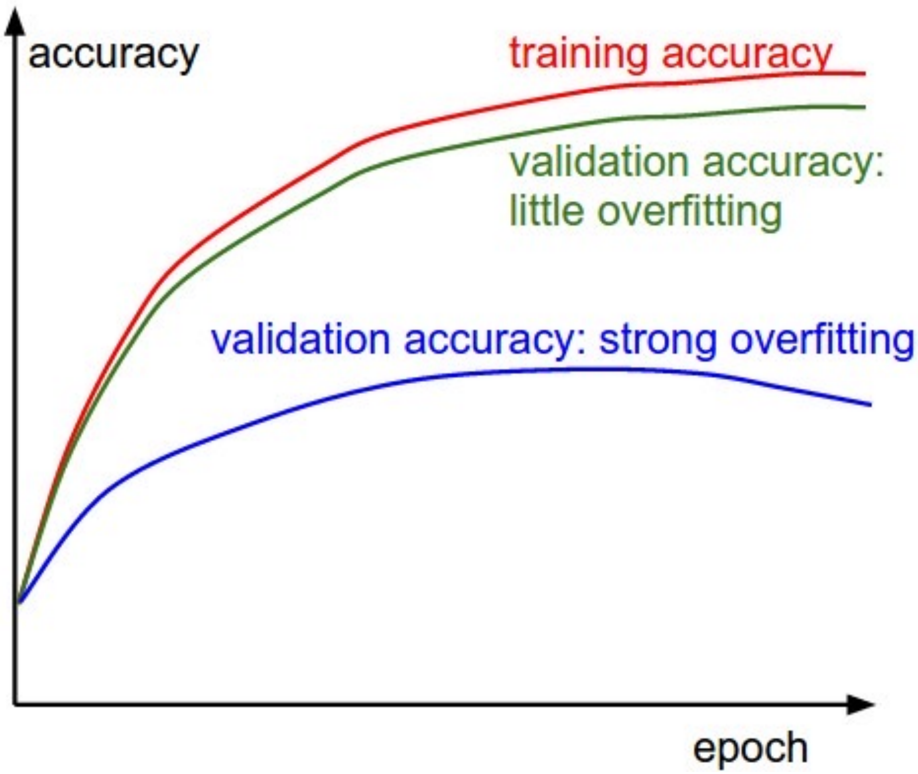
- **Often use rectified linear units (ReLUs)**
  - Faster, less problems of saturation than logistic
- **Use a variety of loss functions**
  - Cross-entropy with *softmax*

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

- **Can be very deep**
- **Solved with mini-batch gradient descent**
- **Regularized using L_1 and L_2 penalty plus "dropout"**
  - and partial convergence and ..

Regularization

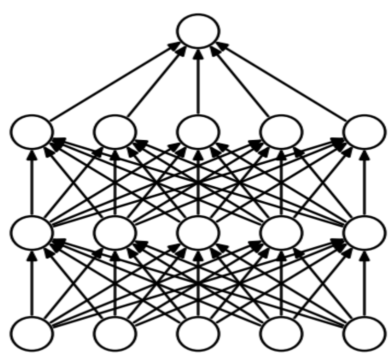
- L_2 and/or L_1
- **Early stopping**
- **Max norm (L_∞)**
  - Weight clipping
  - Gradient clipping
- **Dropout**



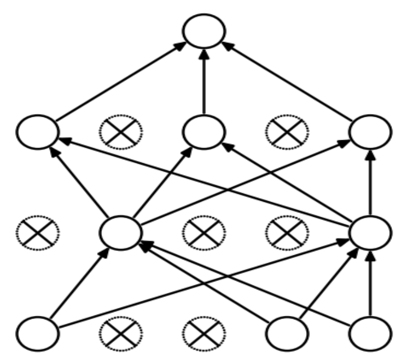
http://cs231n.github.io/neural-networks-3/

Dropout

- Randomly (temporarily) remove a fraction *p* of the nodes (with replacement)
  - Usually, p = 1/2
- Repeatedly doing this samples (in theory) over exponentially many networks
  - Bounces the network out of local minima
- For the final network use all the weights, but shrink them by *p*



(a) Standard Neural Net



(b) After applying dropout.

Gradient descent

- **Gradient descent**
  - Minibatch
  - Gradient clipping
- **Momentum**
- **Learning rate adaptation**
  - Adagrad and friends

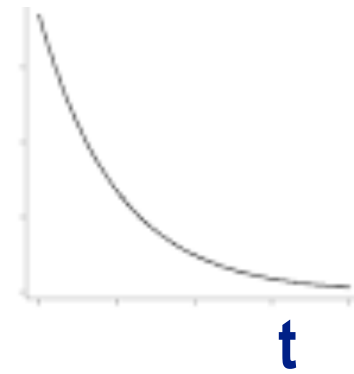
$$\frac{\delta Err}{\delta w} = \frac{Err(w+h) - Err(w-h)}{2h}$$

$$\Delta w^t = \eta \frac{\delta Err}{\delta w} + m \Delta w^{t-1}$$

Learning rate adaption

- **Adjust the learning rate over time**

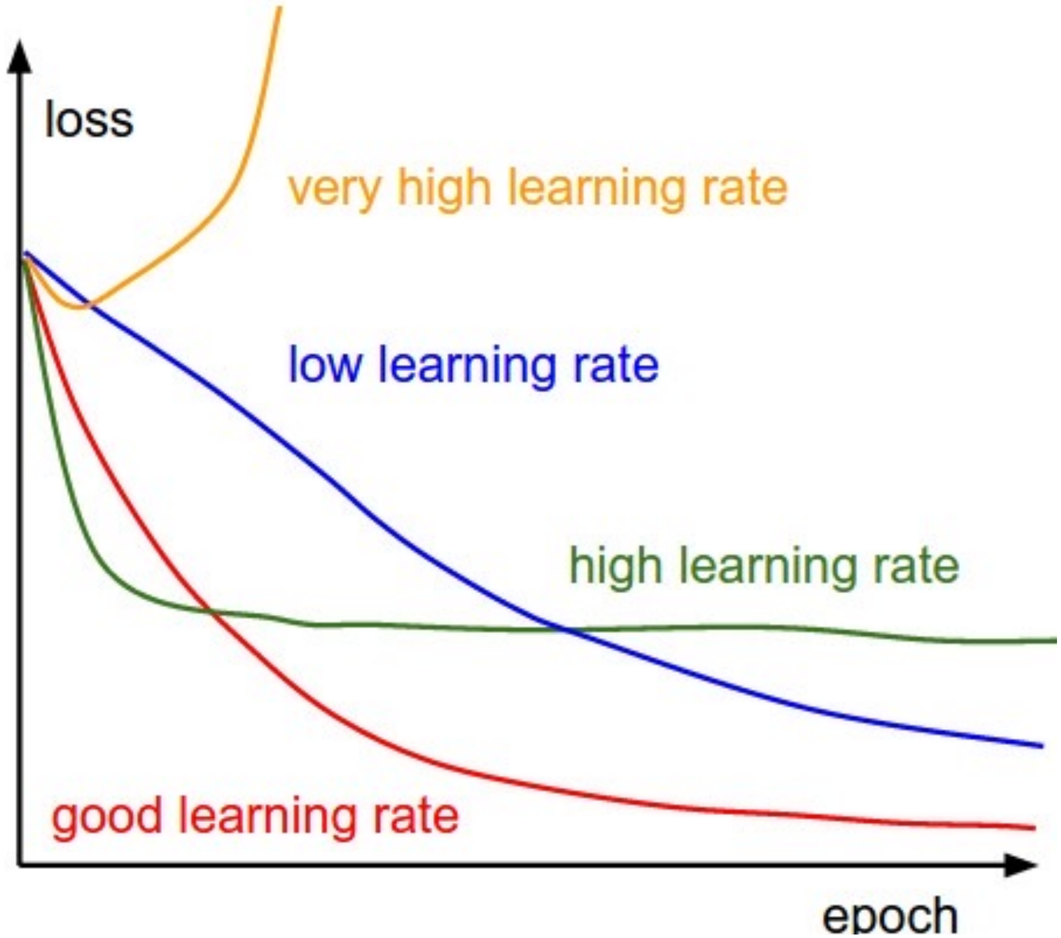
$$\Delta w^t = \eta(t) \frac{\delta Err}{\delta w}$$

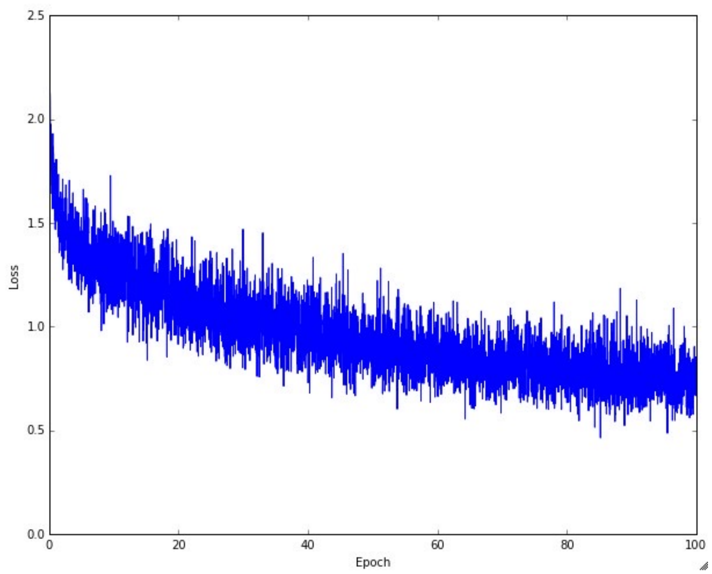


- **Adagrad: make the learning rate depend on previous changes in *each* weight**
  - increases the learning rate for more sparse parameters and decreases the learning rate for less sparse ones.

$$\Delta w_j^t = \frac{\eta}{||\delta w_j^\tau||_2} \frac{\delta Err}{\delta w_j}$$

Learning rate

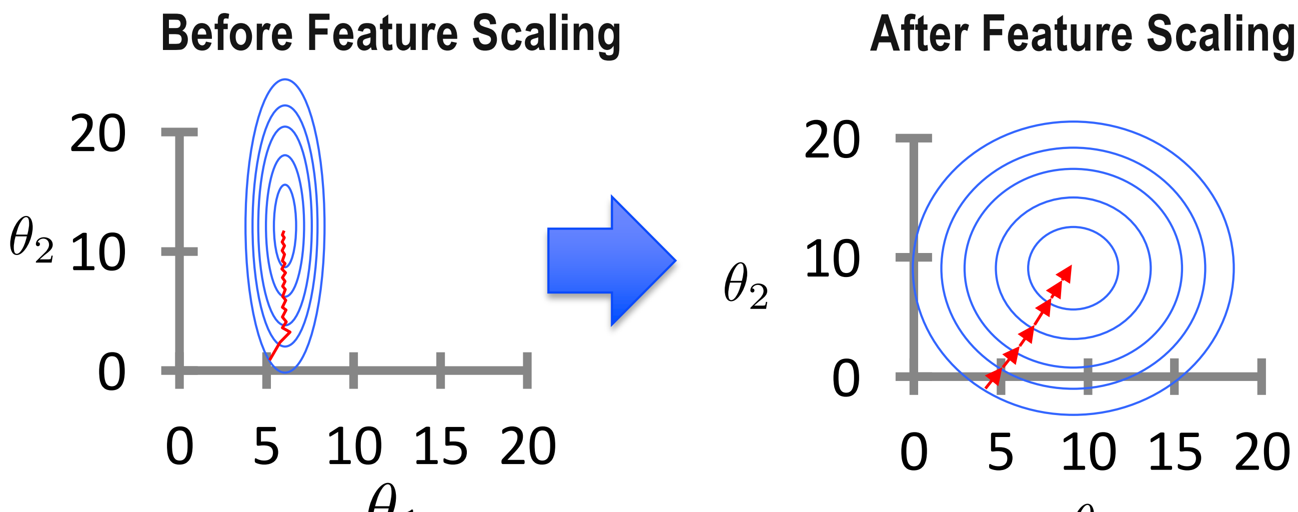




http://cs231n.github.io/neural-networks-3/

Feature Scaling (standardizing)

- Idea: Ensure that features have similar scales



- Can do this for hidden layer outputs as well for each minibatch
- Makes gradient descent converge much faster

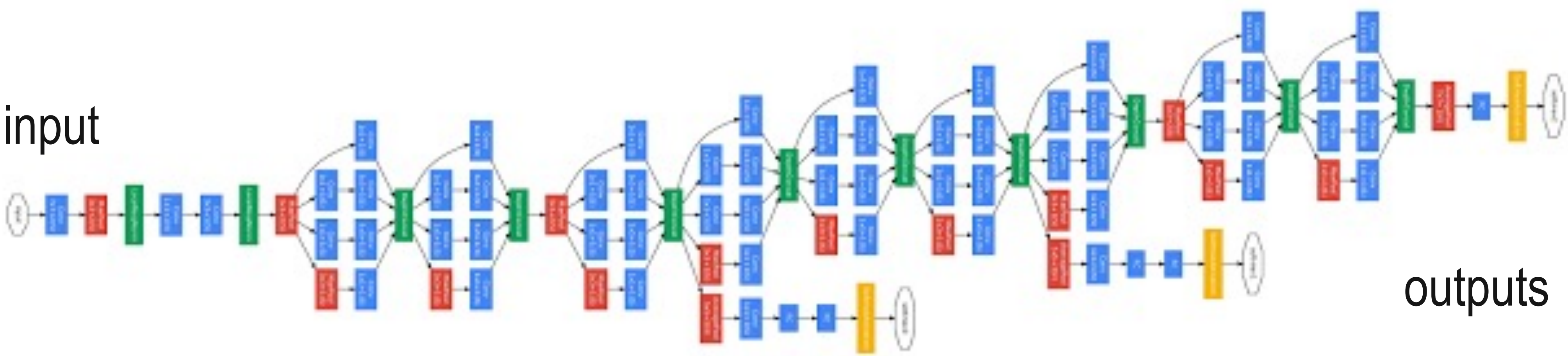
Is deep learning scale invariant?

A word on hyperparameters

- **Regularization**
  - L1, L2
  - Dropout
  - Early stopping
  - Learning rate
- **Architecture**
  - Number of layers, and nodes/layer
  - Filter, maxpool, fully connected

t

Lots of fancy network structures

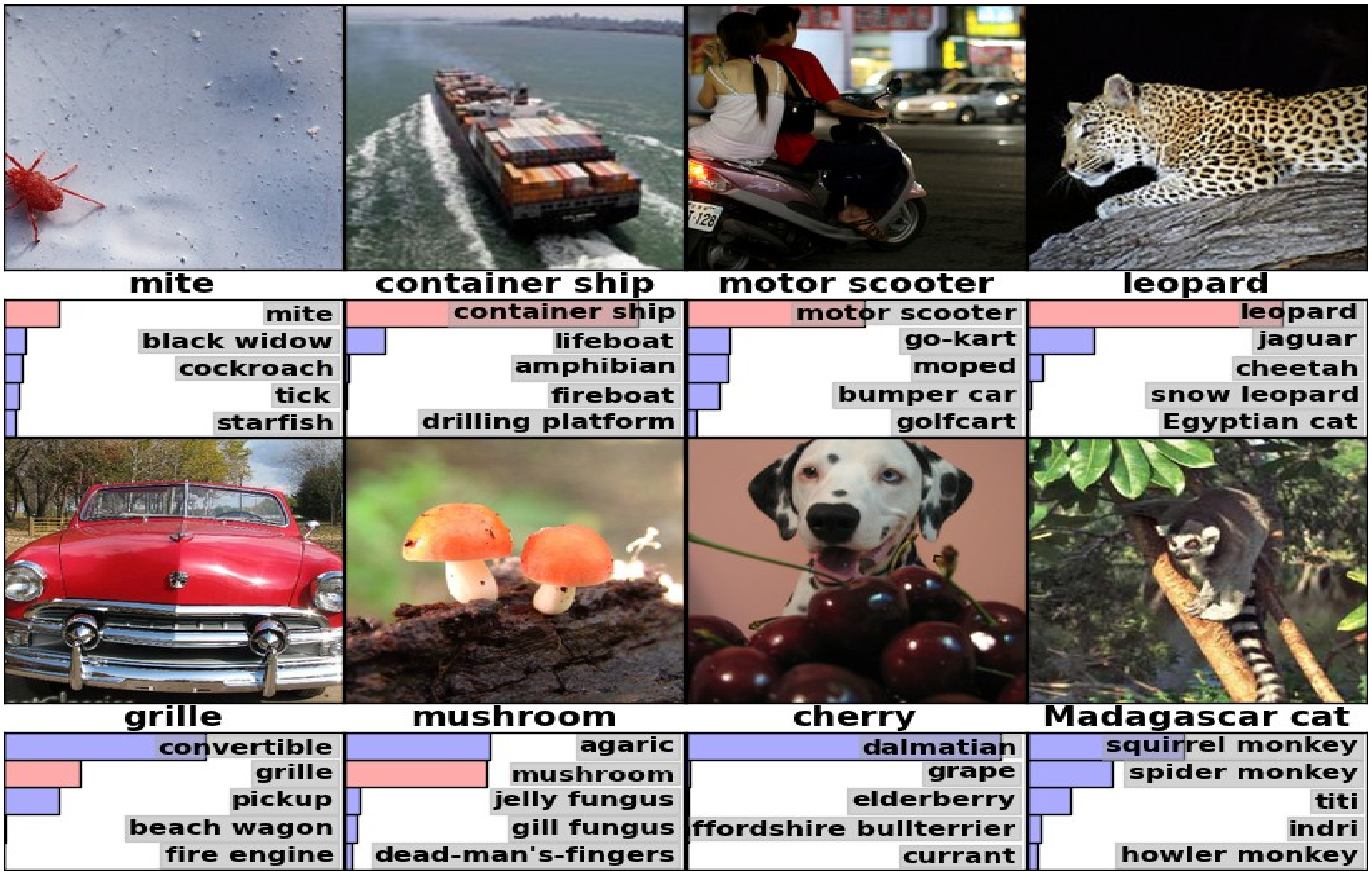


Convolutional (different sizes)
Or fully connected
Maxpool
Concatenation
Softmax

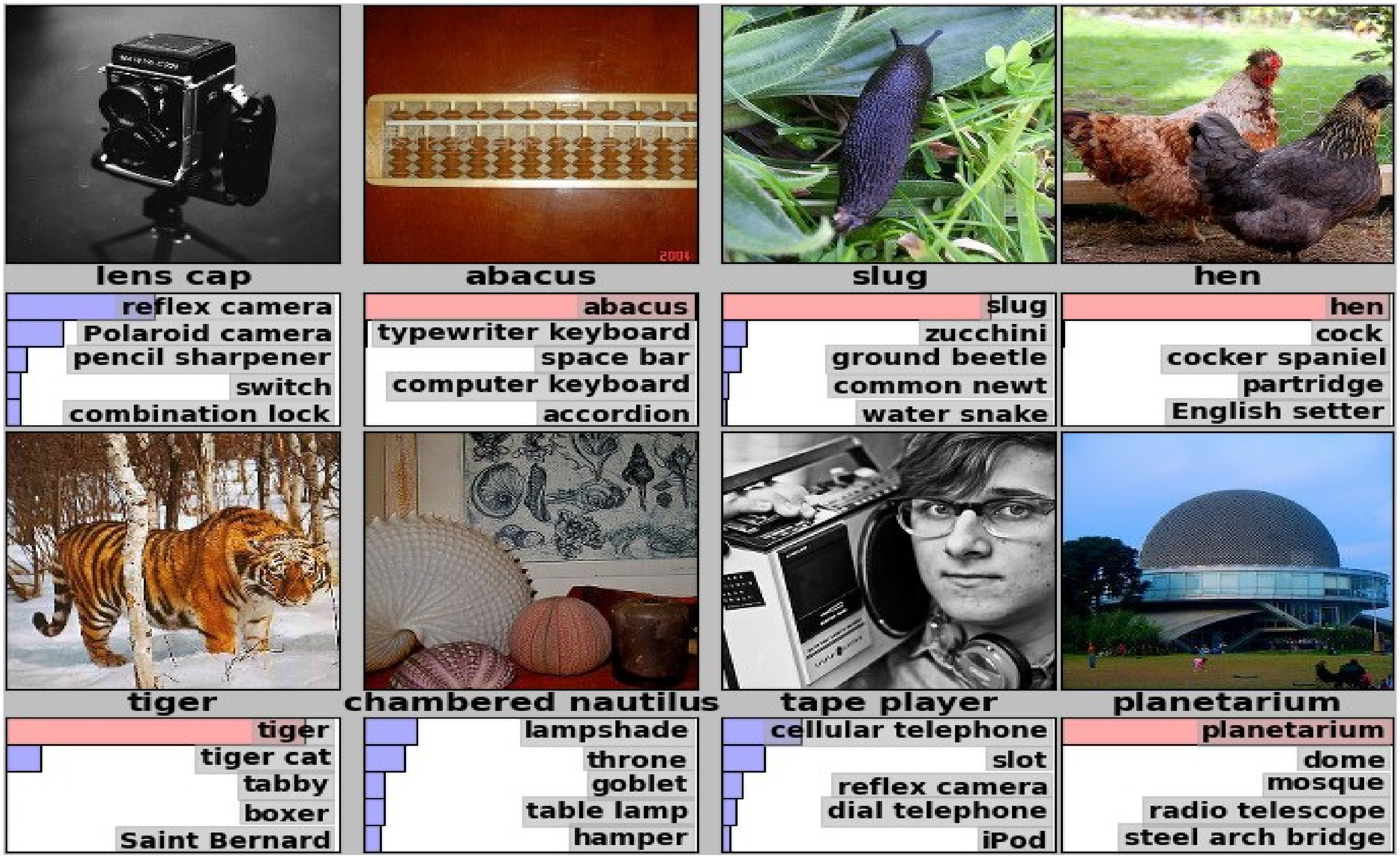
Some layers
use dropout

googlenet

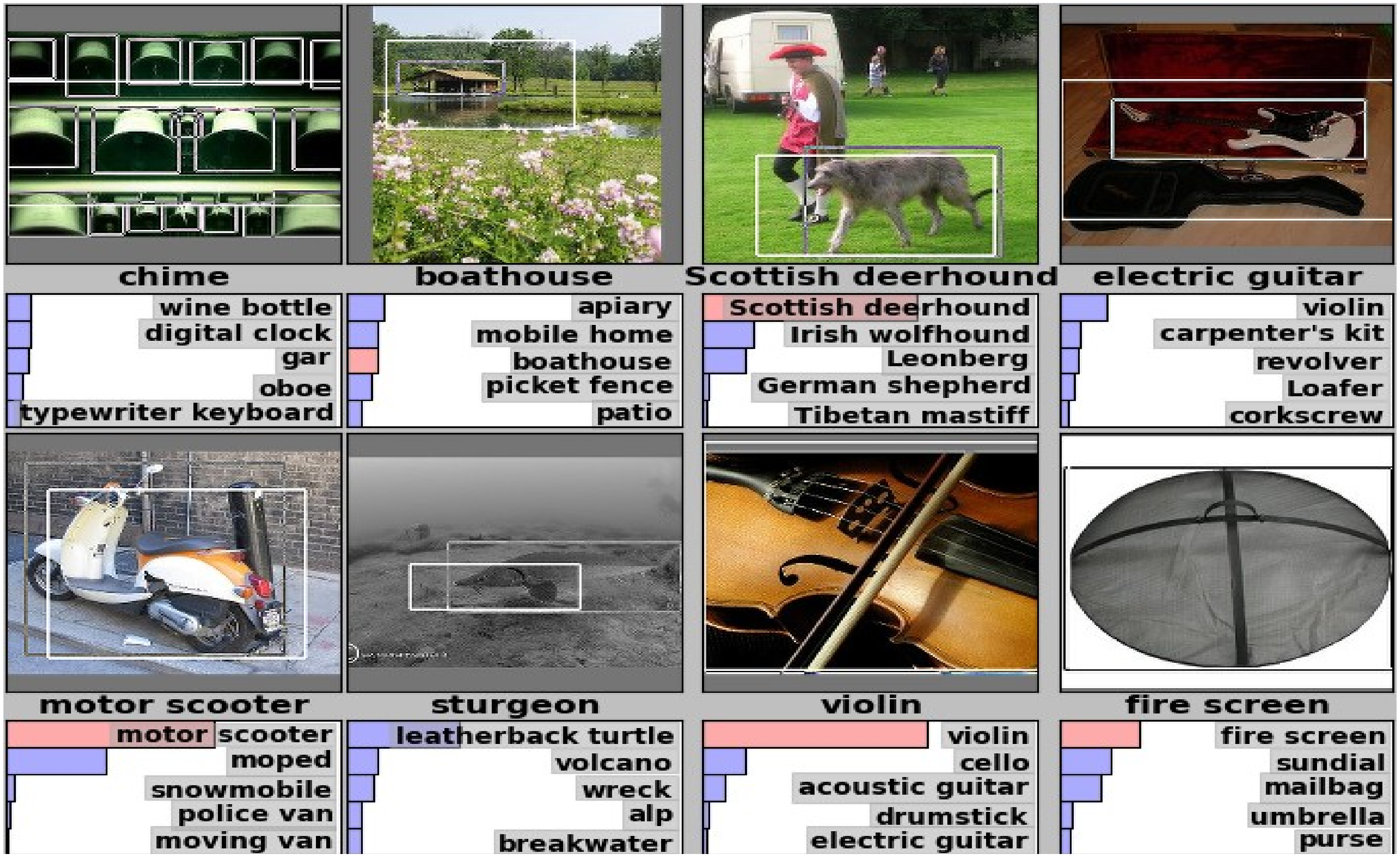
Validation classification



Validation classification

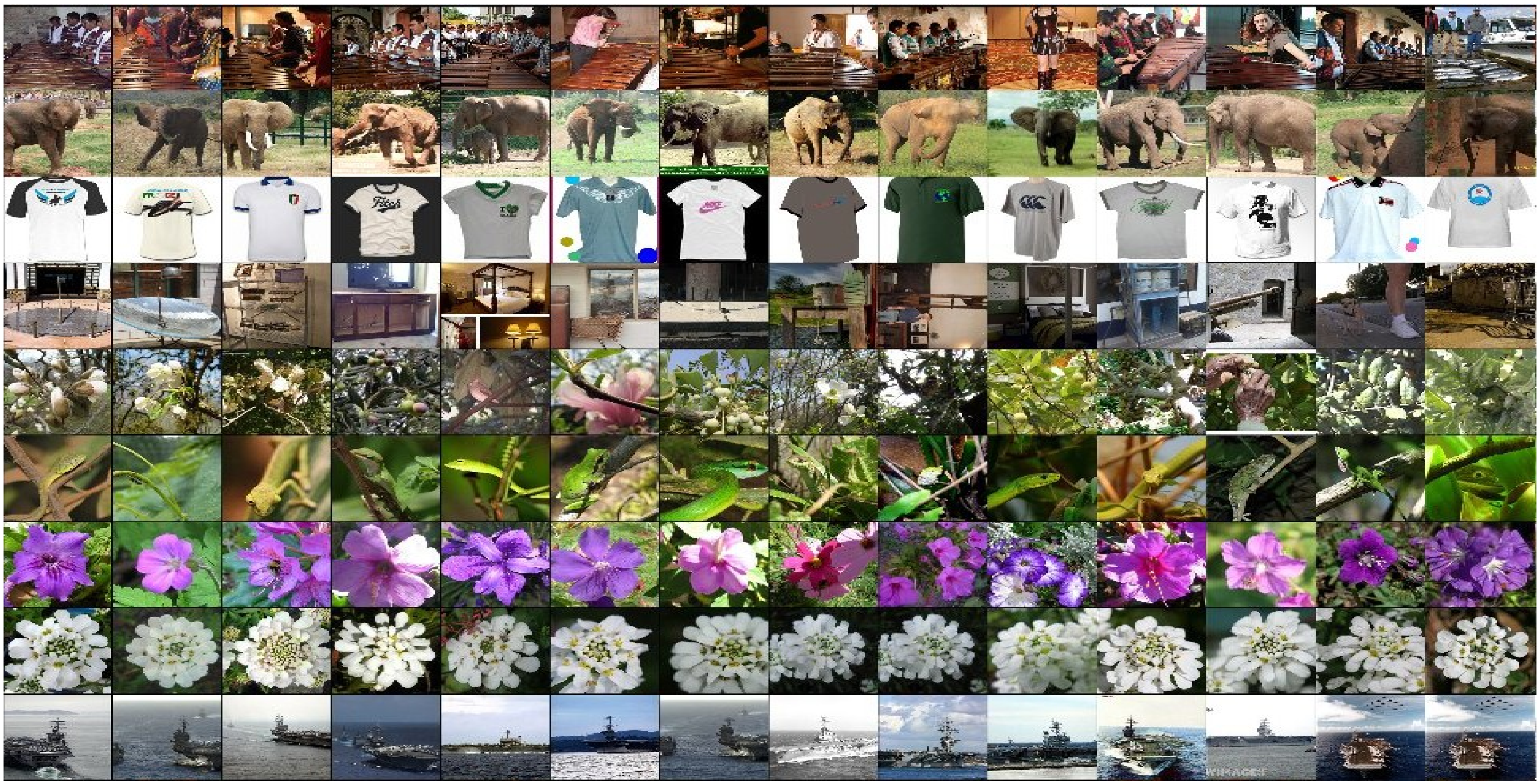


Validation localizations



Retrieval experiments

First column contains query images from ILSVRC-2010 test set, remaining columns contain retrieved images from training set.



Now used for image search; Benefit: Good Generalization

Both recognized as "meal"

Jeff Dean, google

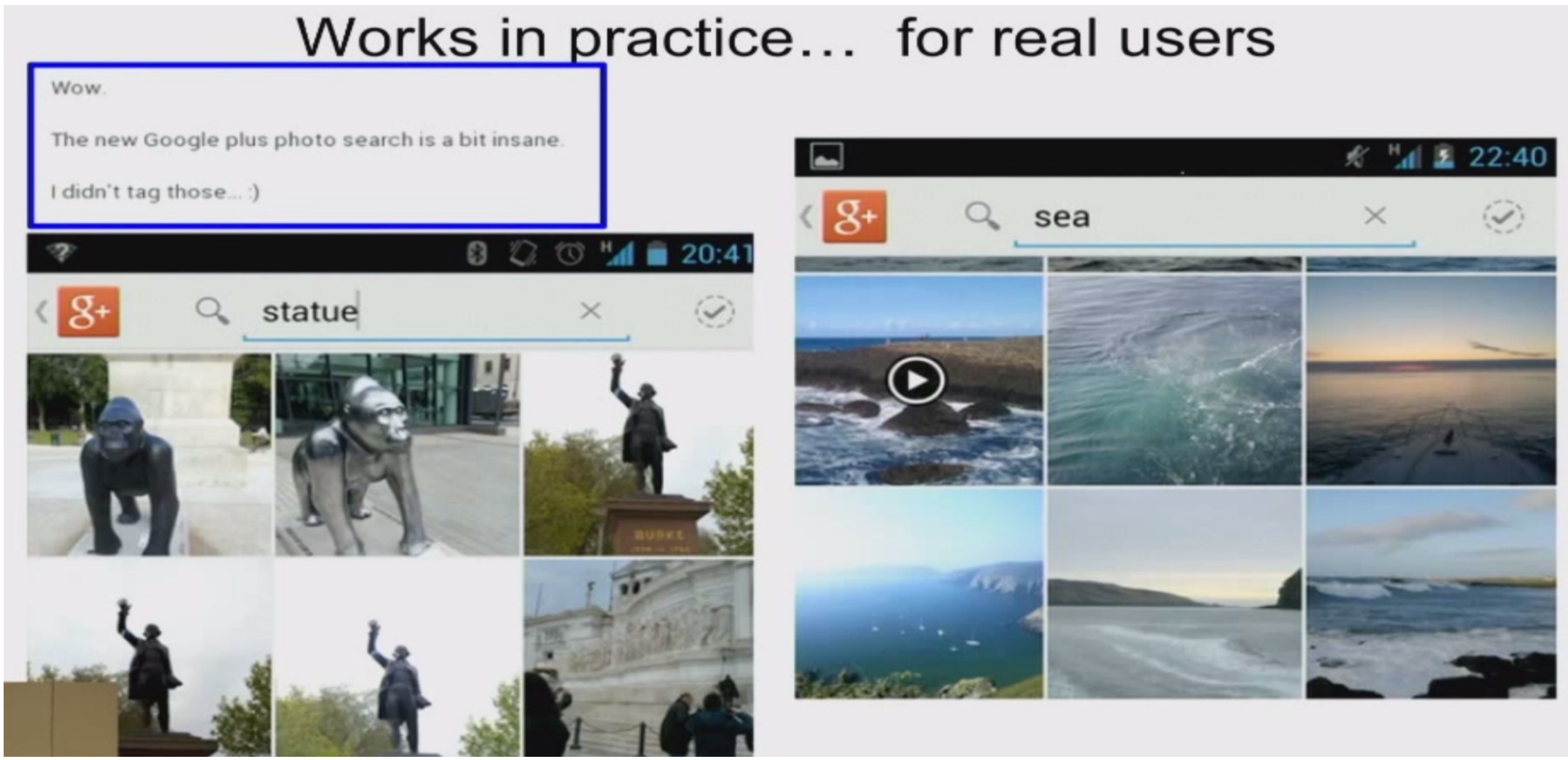
Sensible Errors (sometimes)

“snake”

“dog”

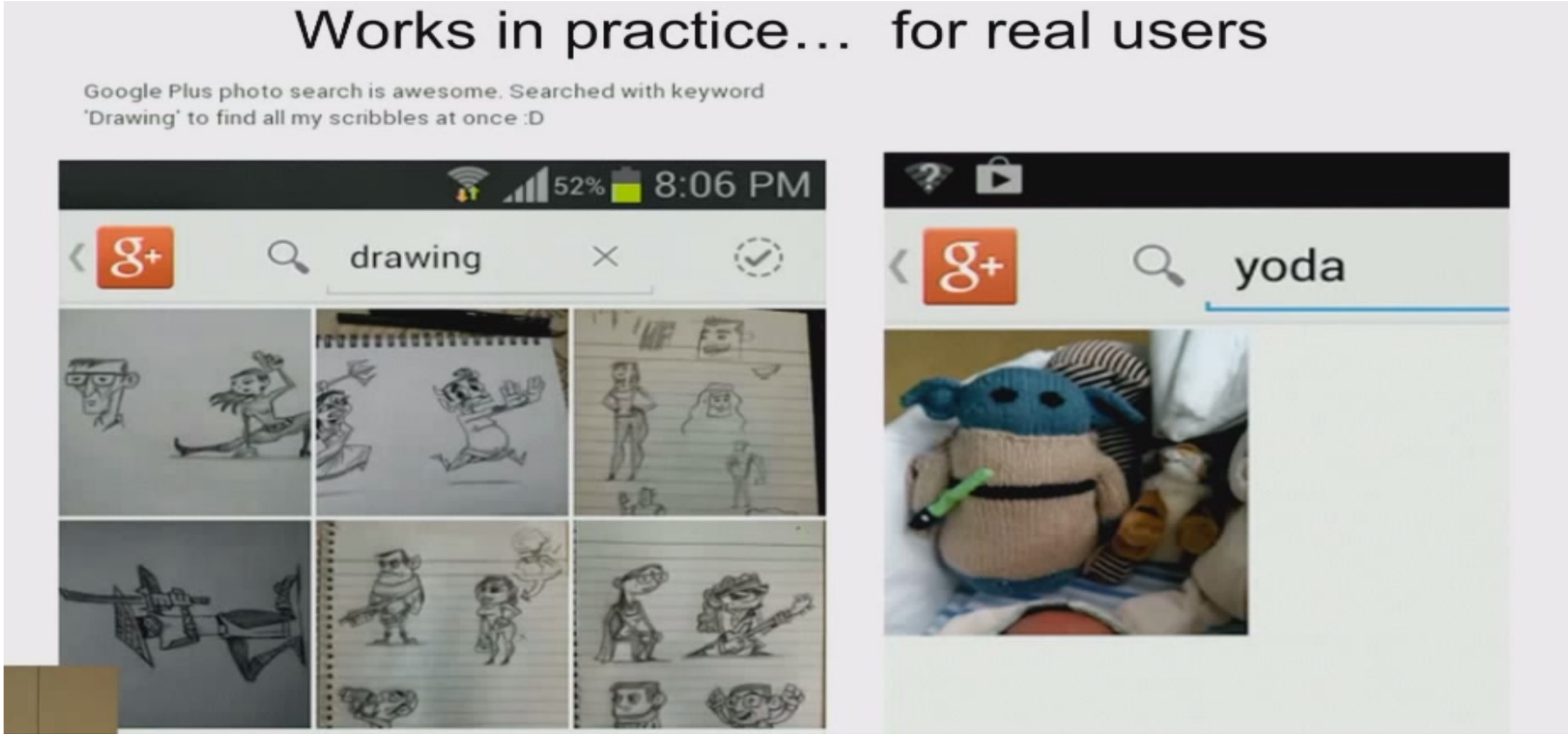
Jeff Dean, google

Now used for image search



Jeff Dean, google

Now used for image search



Jeff Dean, google

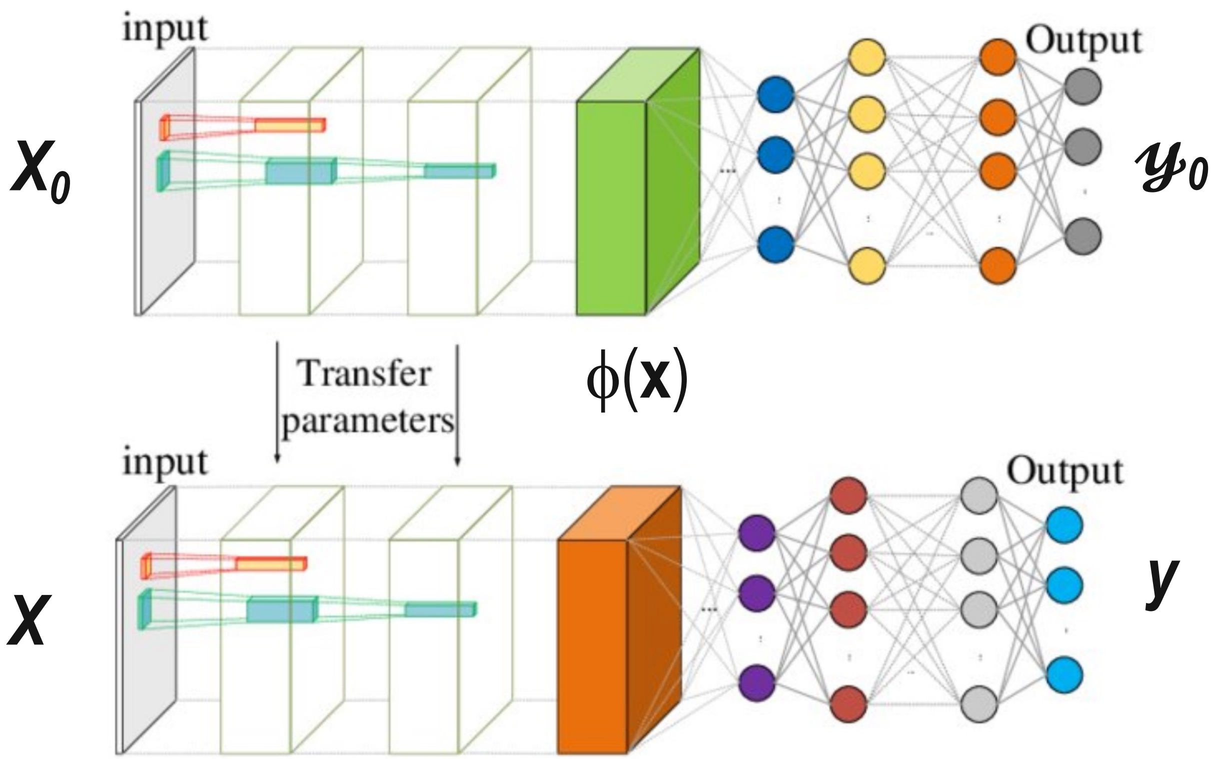
Transfer learning

- Use one data set (X_0, y_0) to train a model
- Find feature transformations $\phi(x)$
- Use those transformations $\phi(x)$ on data from data set with a different label, y.

Data set for learning $\phi(x)$

Data set with target y.

Transfer learning for NNets

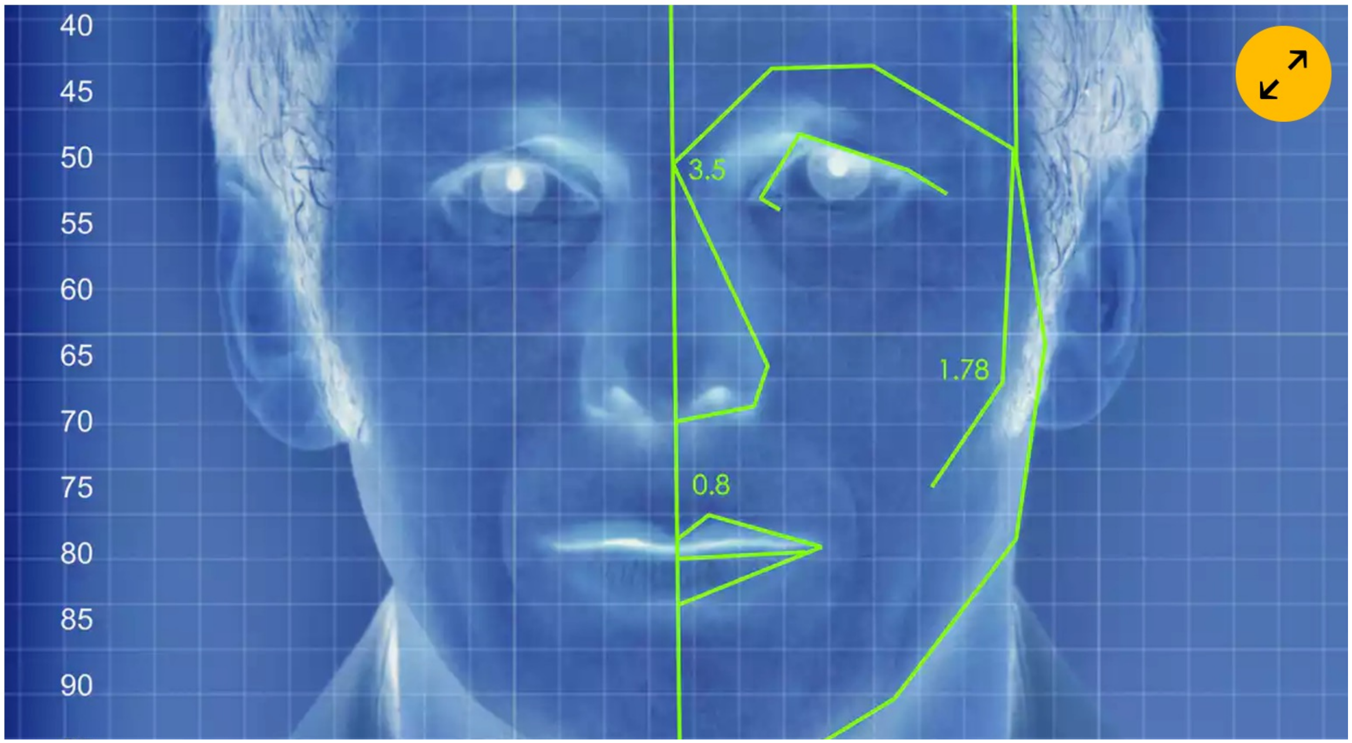


Data set for learning $\phi(x)$

Data set with target *y*.

New AI can guess whether you're gay or straight from a photograph

An algorithm deduced the sexuality of people on a dating site with up to 91% accuracy, raising tricky ethical questions



https://www.theguardian.com/technology/2017/sep/07/new-artificial-intelligence-can-tell-whether-youre-gay-or-straight-from-a-photograph

Deep neural networks are more accurate than humans at detecting sexual orientation from facial images.

Michal Kosinski & Yilun Wang

We show that faces contain much more information about sexual orientation than can be perceived and interpreted by the human brain. We used deep neural networks to extract features from 35,326 facial images. These features were entered into a logistic regression aimed at classifying sexual orientation. Given a single facial image, a classifier could correctly distinguish between gay and heterosexual men in 81% of cases, and in 74% of cases for women. Human judges achieved much lower accuracy: 61% for men and 54% for women. The accuracy of the algorithm increased to 91% and 83%, respectively, given five facial images per person. Facial features employed by the classifier included both fixed (e.g., nose shape) and transient facial features (e.g., grooming style). Consistent with the prenatal hormone theory of sexual orientation, gay men and women tended to have gender-atypical facial morphology, expression, and grooming styles. …. Additionally, given that companies and governments are increasingly using computer vision algorithms to detect people's intimate traits, our findings expose a threat to the privacy and safety of gay men and women.

https://osf.io/fk3xr/ **2017**

Deep learning case study

- **Download images and labels from a dating site**
  - where people declare their sexual orientation
- **Only keep images with a single “good” face**
  - Use Face++ to identify faces -- yielded 35,000 faces
- **Use M-turkers to QC & restrict to Caucasians**
- **Use pretrained CNN to compute ~ 4,000 ‘scores’/image**
  - VGG-Face was trained on 2.6 million faces
- **Use logistic regression on SVD of the 4,000 scores**
  - report cross-validation error predicting gay/straight

Limitations of NNs

- For many problems, tree ensemble methods are better than NNs. Why?

Visualizing networks

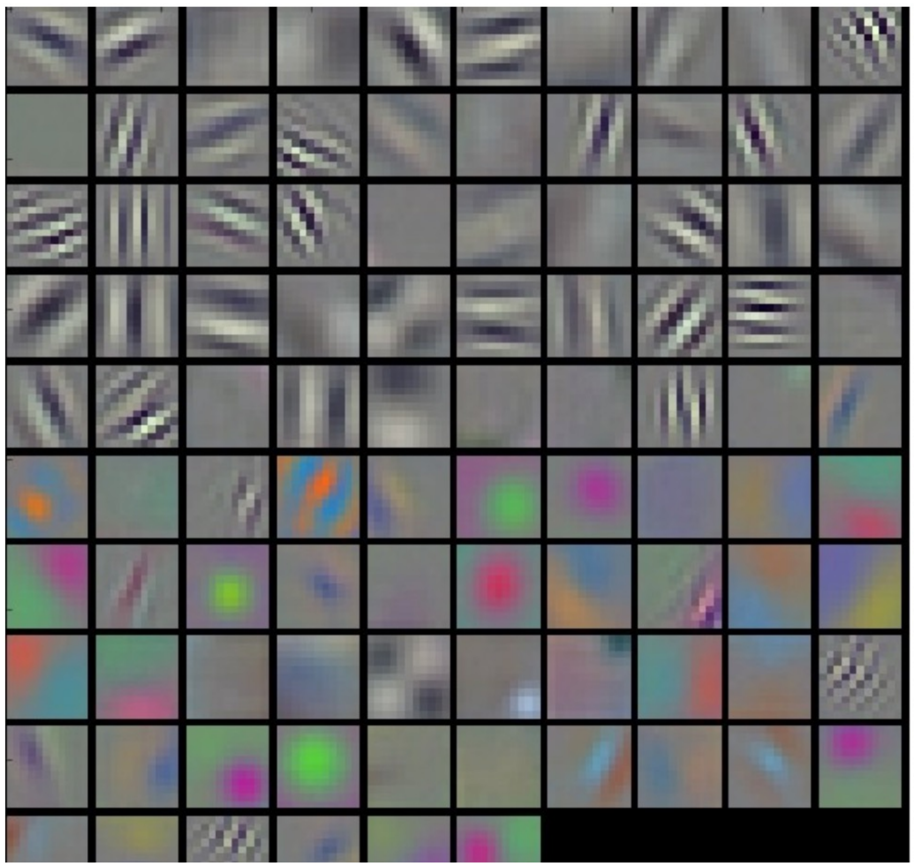
- **Display pattern of hidden unit activations**
  - Just shows they are distributed and sparse
  - Better: feed activations into a linear model to predict something
- **Show input that maximizes a node's output**
  - Over all inputs in the training set
  - Over the space of possible inputs
- **Show effect of occluding parts of an image on classification accuracy**

http://cs231n.github.io/understanding-cnn/

What happens where

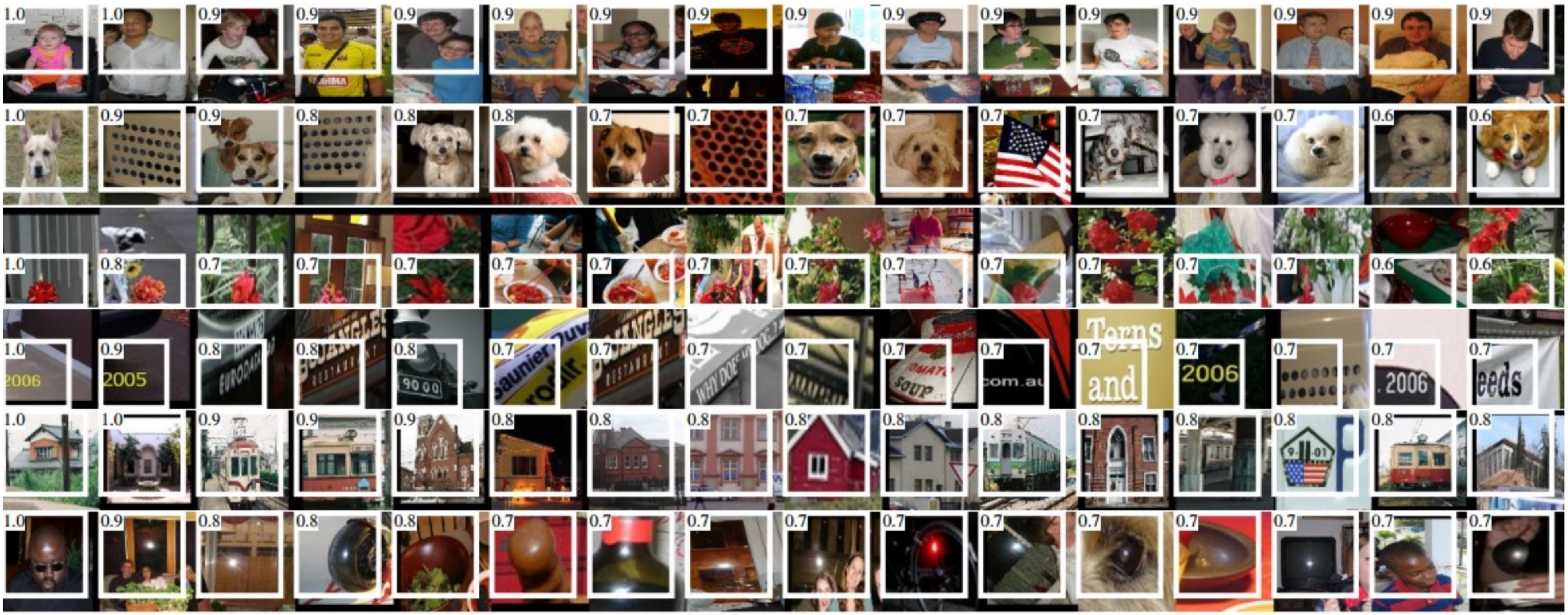
- **In CNN's for images**
  - Early layers do feature detection
  - Later layers do object detection
- **In Neural nets for language**
  - Early layers do Part of Speech detection
  - Later layers do co-references …

Maximally activating inputs for the first CONV layer of an AlexNet

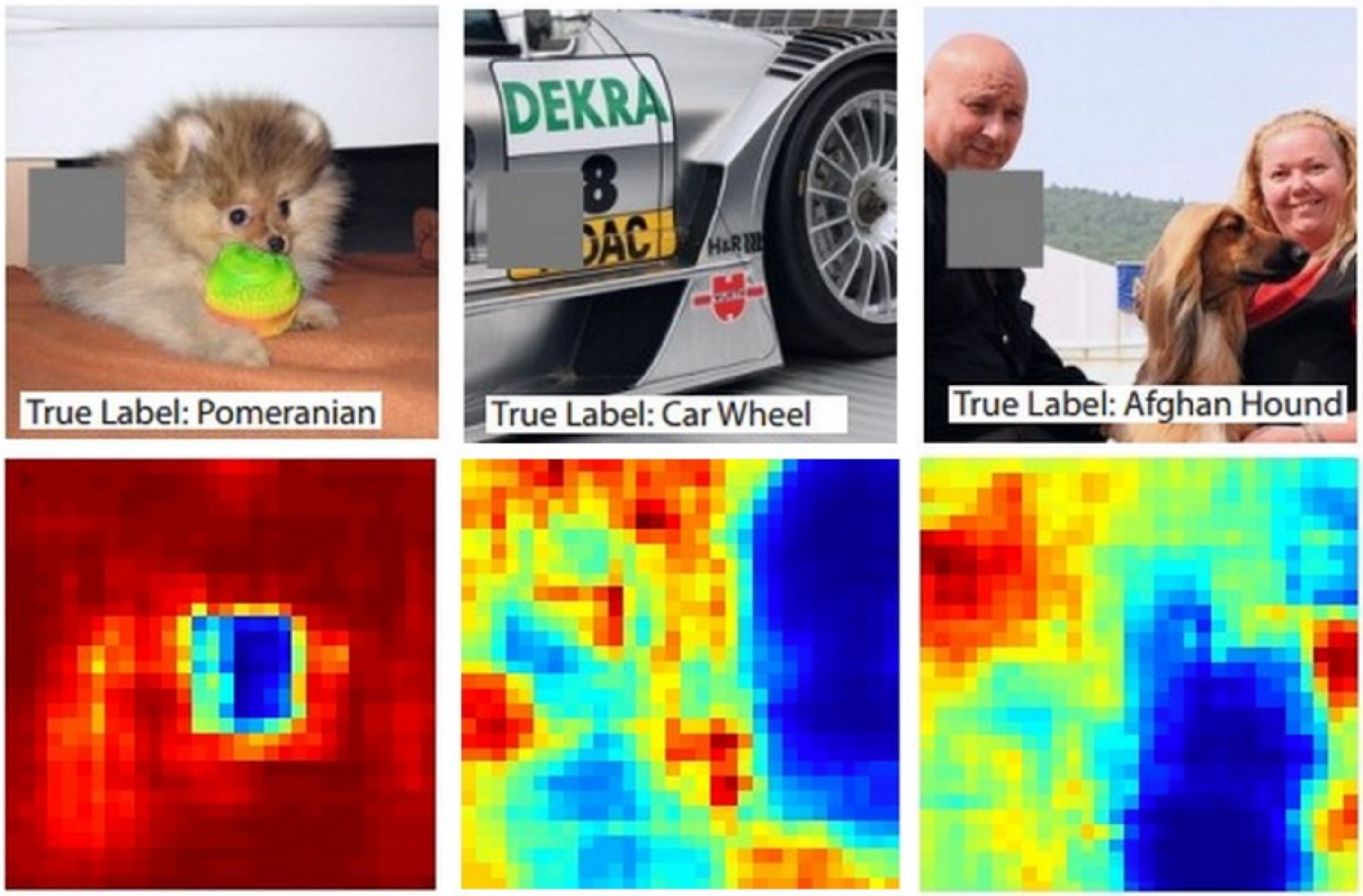


http://cs231n.github.io/understanding-cnn/

Maximally activating images for some 5th maxpool layer neurons of an AlexNet.



P(correct label) after occlusion



Matthew Zeiler's [Visualizing and Understanding Convolutional Networks](#):

What you should know

- **CNN**
  - local receptive field, max pooling
- **Rectified Linear Unit (ReLU)**
- **At least four kinds of regularization**
  - Dropout
- **Back-propagation, momentum, Adagrad, mini-batch**
- **Transfer learning**