Deep Neural Networks

CS 486/686
University of Waterloo
Lecture 21: July 14, 2015

Outline

- Deep Neural Networks
  - Gradient Vanishing
    - Rectified linear units
  - Overfitting
    - Dropout
- Breakthroughs
  - Acoustic modeling in speech recognition
  - Image recognition

Deep Neural Network

- Definition: neural network with many hidden layers
- Advantage: high expressivity
- Challenges:
  - How should we train a deep neural network?
  - How can we avoid overfitting?

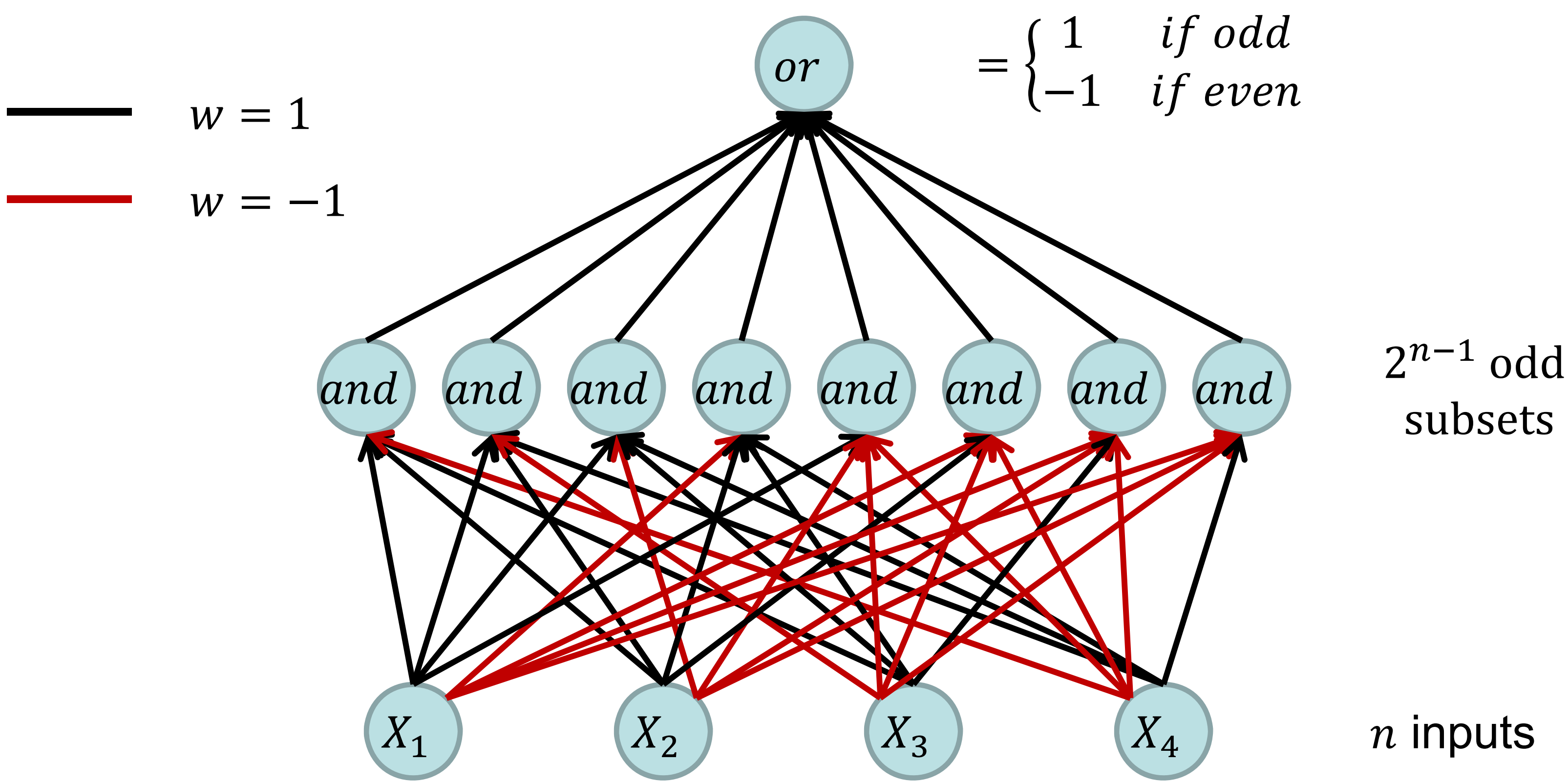
Expressivity

- Neural networks with one hidden layer of sigmoid/hyperbolic units can approximate arbitrarily closely neural networks with several layers of sigmoid/hyperbolic units

- However as we increase the number of layers, the number of units needed may decrease exponentially (with the number of layers)

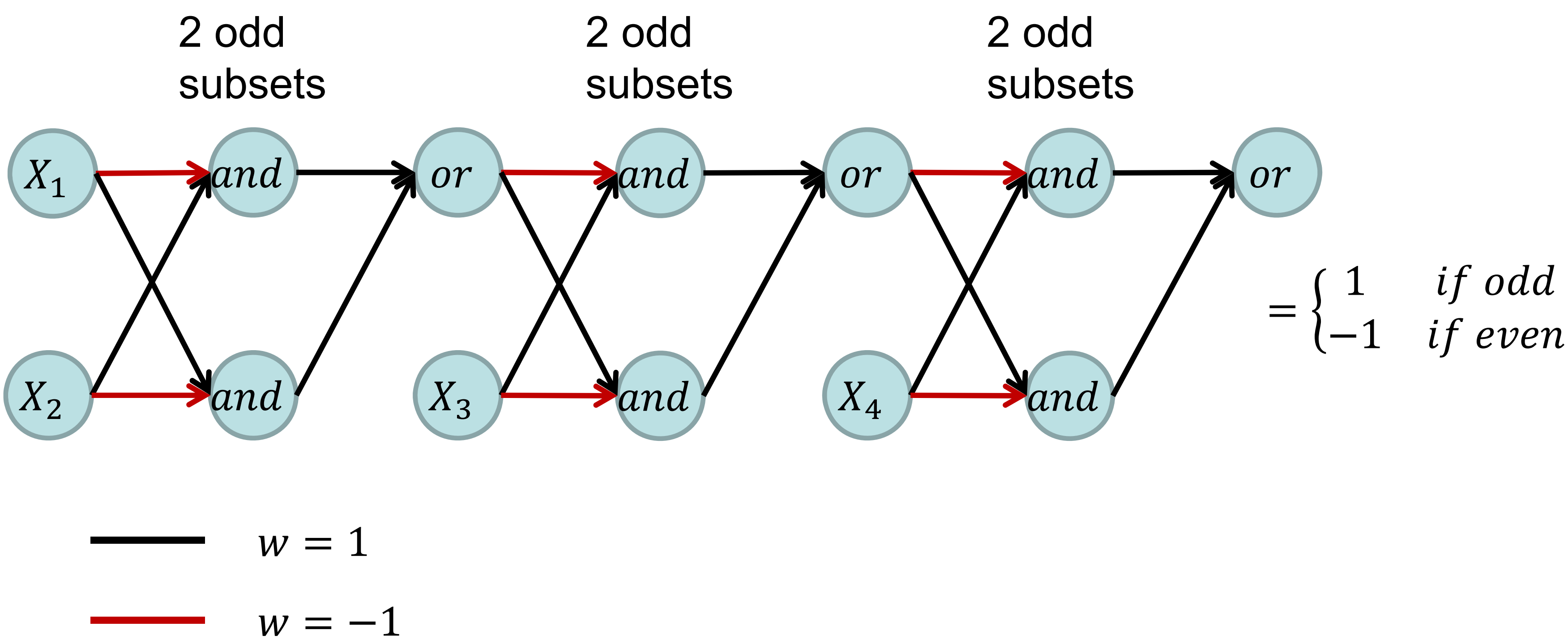
Example - Parity Function

- Single layer of hidden nodes



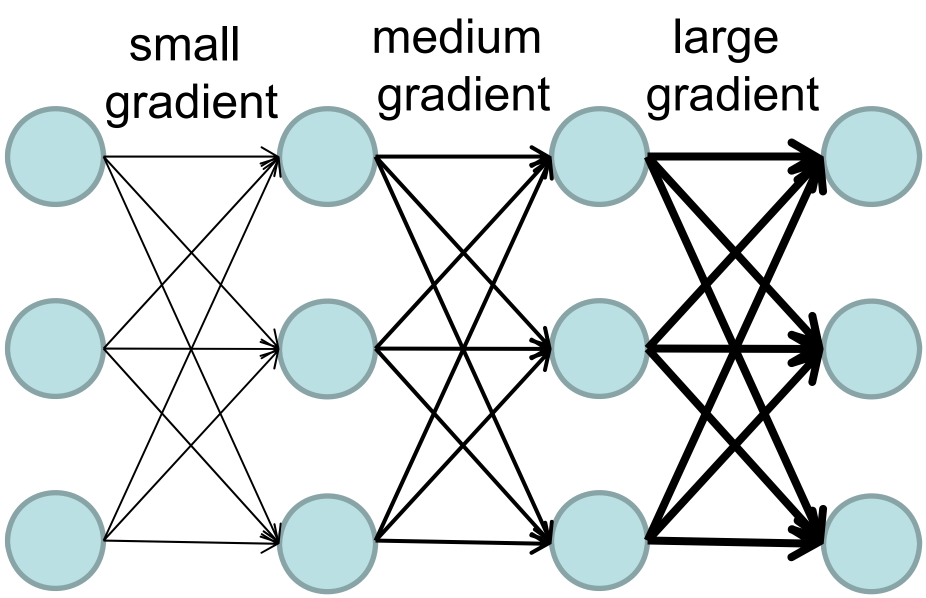
Example - Parity Function

- $2n - 1$ layers of hidden nodes



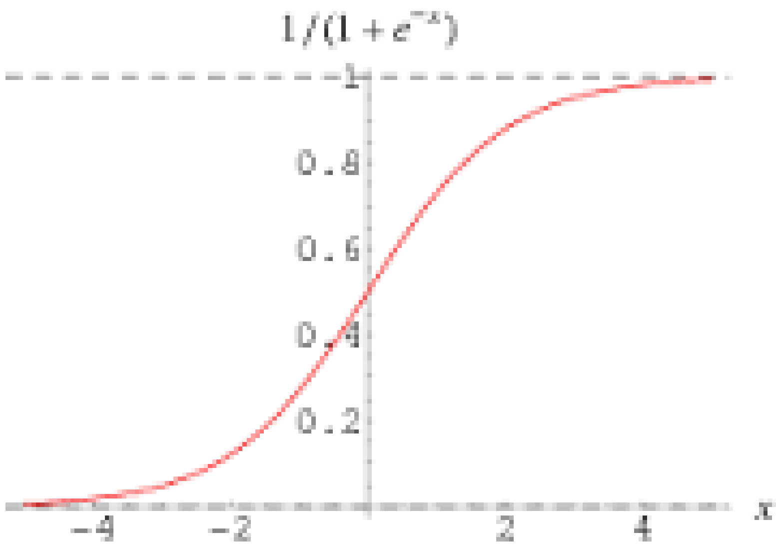
Vanishing Gradients

- Deep neural networks of sigmoid and hyperbolic units often suffer from vanishing gradients

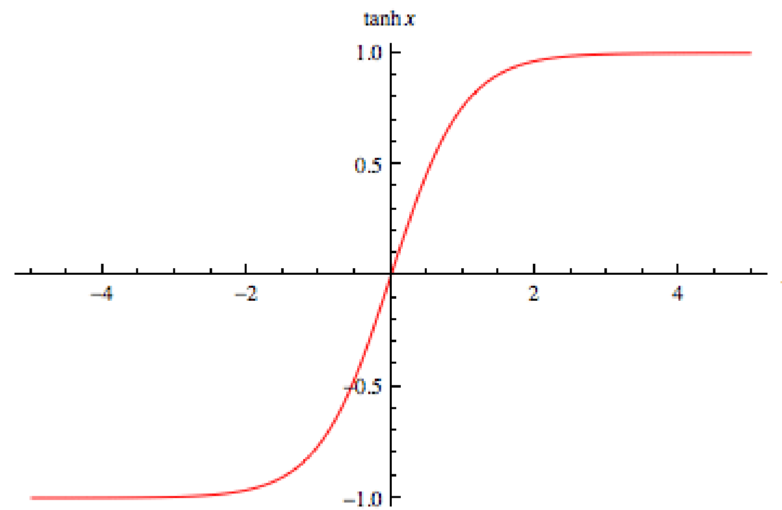


Sigmoid and hyperbolic units

- Derivative is always less than 1



sigmoid



hyperbolic

Simple Example

- $Y = \sigma(W_4\, \sigma(W_3\, \sigma(W_2\, \sigma(W_1\, X))))$

$$X \xrightarrow{W_1} H_1 \xrightarrow{W_2} H_2 \xrightarrow{W_3} H_3 \xrightarrow{W_4} Y$$

- Common weight initialization in (-1,1)
- Sigmoid function and its derivative always less than 1
- This leads to vanishing gradients:

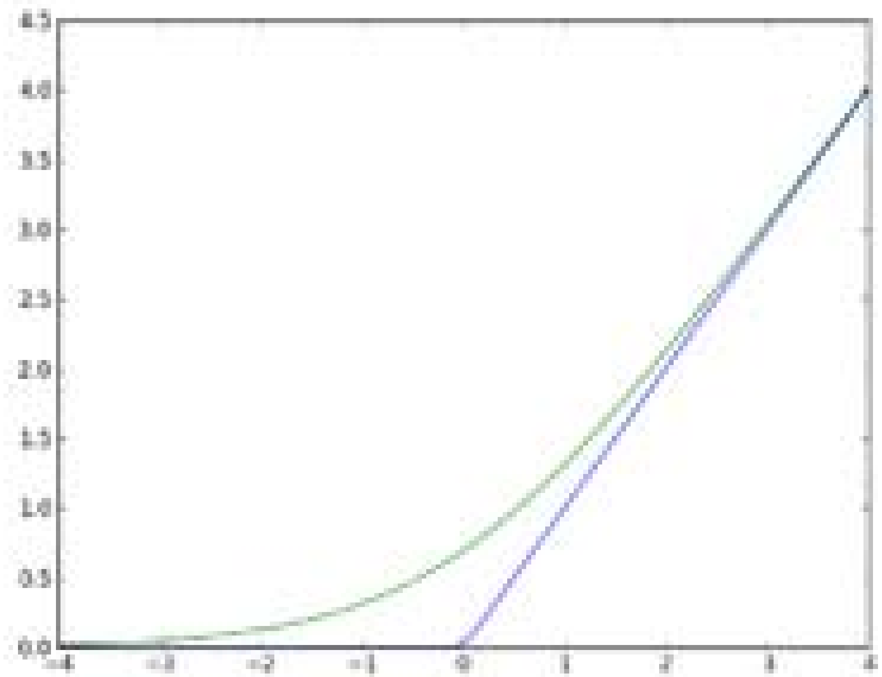
$$\frac{\partial Y}{\partial W_4} = \sigma'^{(in_4)}\sigma(in_3)$$
$$\frac{\partial Y}{\partial W_3} = \sigma'(in_4)W_4\sigma'(in_3)\sigma(in_2) \leq \frac{\partial Y}{\partial W_4}$$
$$\frac{\partial Y}{\partial W_2} = \sigma'(in_4)W_4\sigma'(in_3)W_3\sigma'(in_2)\sigma(in_1) \leq \frac{\partial Y}{\partial W_3}$$
$$\frac{\partial Y}{\partial W_1} = \sigma'(in_4)W_4\sigma'(in_3)W_3\sigma'(in_2)W_2\sigma'(in_1)X \leq \frac{\partial Y}{\partial W_2}$$

Avoiding Vanishing Gradients

- Two popular solutions:
  - Pre-training
  - **Rectified linear units**

Rectified Linear Units

- Rectified linear: $g(x) = \max(0, x)$
  - Gradient is 0 or 1
  - Sparse computation



- Soft version ("Softplus") :
  $g(x) = \log(1 + e^x)$

Overfitting

- High expressivity increases the risk of overfitting
  - # of parameters is often larger than the amount of data

- Solution: **dropout**

Dropout

- Idea: randomly "drop" some units from the network when training
- Training: at each iteration of gradient descent
  - Each hidden unit is dropped with prob. 0.5
  - Each input unit is dropped with prob. 0.2
- Prediction (testing):
  - Multiply the output of each unit by its drop probability

Intuition

- Dropout can be viewed as an approximate form of ensemble learning

- In each training iteration, a different subnetwork is trained
- At test time, these subnetworks are "merged" by averaging their weights

Robustness

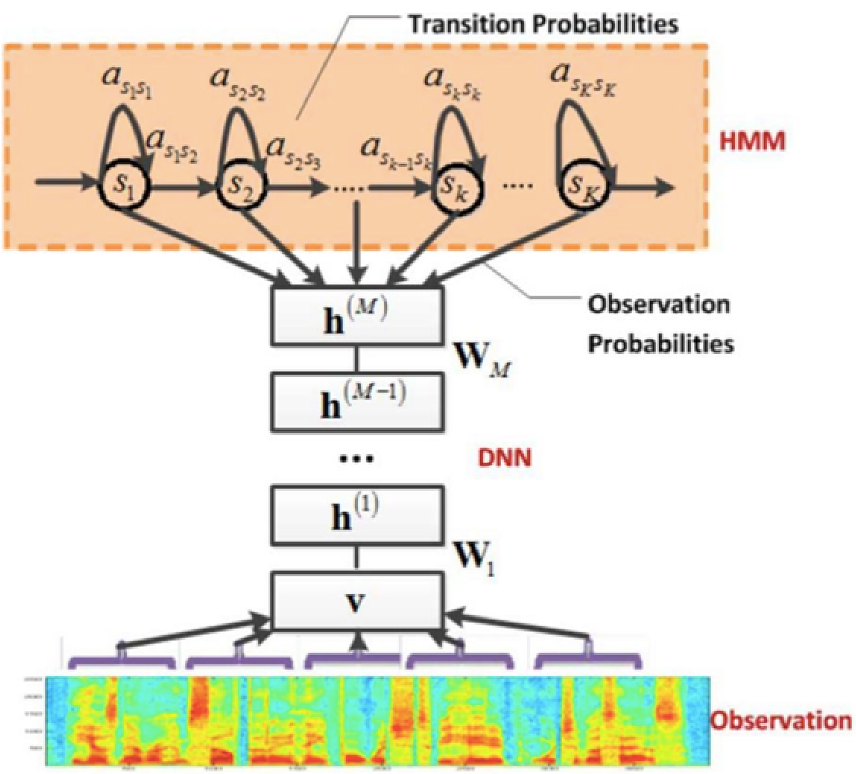
- In sexual reproduction, half of the genes of two individuals are dropped and the remaining genes are merged to produce a new individual
- Genes are forced to evolve independently so that most combinations yield functional individuals
- Similarly, units in a neural net are forced to capture features that are largely independent of other units

Applications of Deep Neural Networks

- Speech Recognition
- Image recognition
- Machine translation
- Control
- Any application of shallow neural networks

Acoustic Modeling in Speech Recognition

Architecture of a DNN-HMM hybrid system



Acoustic Modeling in Speech Recognition

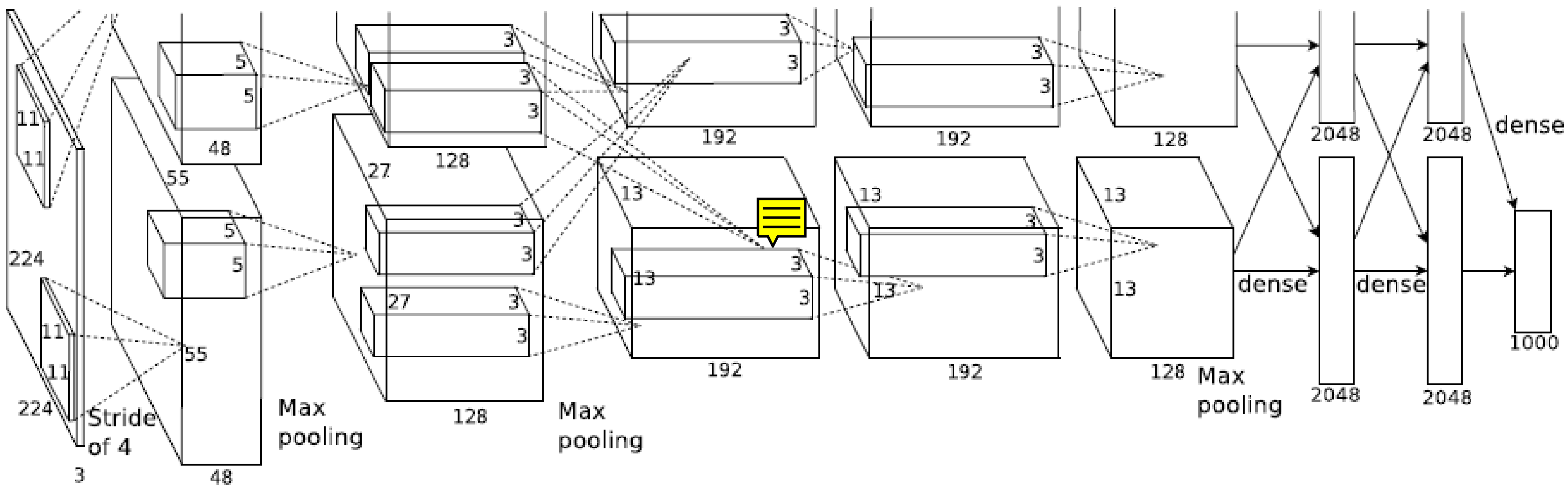
TABLE III

A comparison of the Percentage Word Error Rates using DNN-HMMs and GMM-HMMs on five different large vocabulary tasks.

task	hours of training data	DNN-HMM	GMM-HMM with same data	GMM-HMM with more data
Switchboard (test set 1)	309	18.[illegible]	27.4	18.6 (2000 hrs)
Switchboard (test set 2)	309	16.1	23.6	17.1 (2000 hrs)
English Broadcast News	50	17.5	18.8	
Bing Voice Search (Sentence error rates)	24	30.4	36.2	
Google Voice Input	5,870	12.3		16.0 (>>5,870hrs)
Youtube	1,400	47.6	52.3	

Image Recognition

- Convolutional Neural Network
  - With rectified linear units and dropout
  - Data augmentation for transformation invariance



ImageNet Breakthrough

- Results: ILSVRC-2012
- From Krizhevsky, Sutskever, Hinton

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

ImageNet Breakthrough

- From Krizhevsky, Sutskever, Hinton

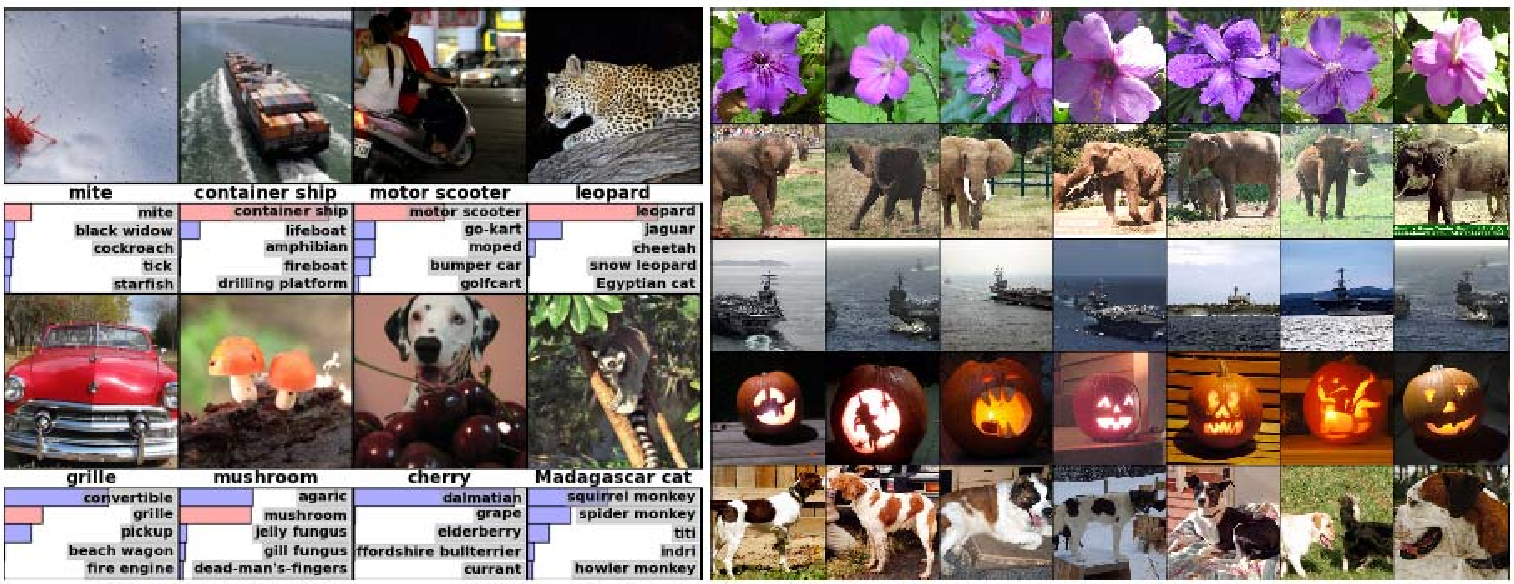


Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.