Deep Networks
Convolution: Sobel Edge Detector

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>0</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gx</td>
<td>-2</td>
<td>0</td>
<td>+2</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>0</td>
<td>+1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>+1</th>
<th>+2</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gy</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>
AlexNet Impressive Results for the 2012 ImageNet Challenge

Eight layers with weights: five convolutional and three fully connected; 60,000,000 weights to be estimated

ImageNet Challenge
- 200 categories, labeled training images
- 1000 categories, object localization
- 150,000 labeled photographs
First Layer Convolutional Kernels

M. Zeiler and R. Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014
Deepening Imagenet Networks

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, arXiv 2015
The Bad News: Adversarial Examples

“panda” 57.7% confidence

+ .007 ×

“nematode” 8.2% confidence

= “gibbon” 99.3% confidence
The Bad News: Rubbish Examples

Each of these is classified by an ImageNet winner with > 99.5% confidence

Evaluating Classifiers

• Accuracy
  – Not smooth
  – Misleading in unbalanced situations
    • e.g., Achalasia, a rare disease
    • approximately 10 cases per 100,000 population
    • “No, you’re fine” yields 99.99% accuracy

• Loss
  – Classifier parameters often tradeoff false positives with false negatives
  – Difficult to compare classifiers with different strengths
Recall and Precision

• Search / Information retrieval, Classification generally
• Distinguish false positive errors from false negatives
• Recall:
  – The fraction of the desired items that were returned
  – True Positives / (True Positives + False Negatives)
• Precision:
  – The fraction returned that were desired
  – True Positives / (True Positives + False Positives)
F Score

• Parameterized by $\beta$

$$F_\beta = (1 + \beta^2) \times \text{Precision} \times \text{Recall} / ((\beta^2 \times \text{Precision}) + \text{Recall})$$

• Common $\beta$
  – $F_1$ weighs recall and precision equally (harmonic mean)
  – $F_{0.5}$ weighs recall lower than precision
  – $F_2$ weighs recall higher than precision

• PR Curve:
  Plot
    Precision (dependent Y) vs. Recall (independent X)
  As relevant classification parameters are varied
ROC: Receiver Operating Characteristic from old-time radio

True Positive vs. False Positives
As relevant classification parameters are varied

Single number: AUC are under the curve 0.5 – 1.0
Unsupervised Learning

• AKA Clustering
• Probably most popular form of “data mining”
• Find frequent patterns in data
• Need a metric space for similarity / distance
• Unlabeled examples
• Often very high dimensional representations
• Group “close” examples together
  – Small intra-cluster distances
  – Large inter-cluster distances
Clustering: No Training Labels
Popular Metrics

distance between example A and example B

• Cosine  $\mathbf{A} \cdot \mathbf{B}$  (perhaps normalized by length)
• Euclidean or $L_2$  $\sqrt{\left(\mathbf{A} - \mathbf{B}\right) \cdot \left(\mathbf{A} - \mathbf{B}\right)}$
  Squared distance is cheaper and often used
• Manhattan or $L_1$
• Mahalanobis:  $\sqrt{\left(\mathbf{A} - \mathbf{B}\right)^\top \mathbf{\Sigma}^{-1} \left(\mathbf{A} - \mathbf{B}\right)}$
  where $\mathbf{\Sigma}^{-1}$ is the inverse of the covariance matrix
  like scale-invariant Euclidean (measured in $\sigma$’s)
• Many others
Hierarchical Clustering

• Form dendrograms:

  - Shows affinities among examples / clusters
  - Agglomerative / Divisive
Hierarchical Clustering

• Linkage criterion
  (in addition to distance metric)
• How clusters are compared
• Closest elements of each cluster (single)
• Farthest elements (complete)
• Median elements
• Centroid (may be non-elements)
• Many others
Dendrogram

• Clusters with increasing dissimilarity
• Agglomerative: start with individuals, merge closest, left to right
• Divisive start with one big cluster, find best split, right to left in figure
• Coherence of clusters shown as dendrite length
• Typically one uses a package
K-Means

• Not hierarchical...partition-based
• Simple, useful, popular – easy to implement & adapt (HW8)
• Efficient
  – time & space
  – can be configured to make sequential passes through the data – efficient disk access
• Need to specify number of clusters
  – guess
  – try a succession w/ just a linear penalty
  – adaptive...
K-Means

• Try to find the K best clusters
• Greedy
• Often very sensitive to initialization
• A cluster is represented as the centroid (mean) of its elements
• Generally, not an element of the set
• The centroids specify a clustering
k-Means Algorithm

- Choose k and a distance metric.
- Select k initial locations for the cluster centroids. This induces a Voronoi tessellation over the space.
Voronoi Tessellation

• Red dots are centroids
• Blue lines are boundaries
• Examples (not shown) are fixed
k-Means Algorithm

- Choose k and a distance metric.
- Select k initial locations for the cluster centroids. This induces a Voronoi tessellation over the space.
- Assign each example to the nearest centroid. This partitions the examples into clusters.
- Re-compute each cluster centroid as the mean of its constituent examples.
- Repeat steps 3) and 4) until either:
  - no change
  - some pre-specified number of iterations has been reached
  - the maximum distance moved by any centroid is less than some pre-specified tolerance value
Voronoi Tessellation

- The dots move to the centroid of their examples
- This changes the boundaries
- Some examples may then be reassigned to new clusters
- The centroids are again re-computed and the process repeats
K-Means

• Algorithm alternates between
  – Assignment
  – Update

• Assignment:
  – keep the K means
  – forget old cluster assignments
  – assign each example to the closest mean
  – forms Voronoi tessellation

• Update
  – re-compute the K cluster means by averaging the assigned examples
K-Means

• Both Assignment and Update reduce the within-cluster distances
• Eventually a minimum is reached
K-Means

- Finds a local optimum
- Very sensitive to initialization
- Random restarts, keeping the best
- How to measure best?
  - depends on task
  - Easy: Minimum sum of squared distances to centroids
  - Clusters can disappear
    - continue with k-1
    - split the largest or most spread out cluster
    - choose a current outlier as a replacement centroid
    - ...
Clustering Application
Codewords: BOW for Vision

Here classifying scenes
Also works for detecting (but not localizing) objects

Beach  City  Kitchen
Statistical Image Classification w/ Bag of Words

- Classify by “beach-ness” or “kitchen-ness”
  - Robust & reasonably confident
  - But (like humans) can be difficult to say why
- Build (very rough) generative models over some image features
- Generative model:
  - Estimate \( \Pr(\text{class} \land \text{image-features}) \)
  - Given a new image, calculate the image features
  - Classify as \( \text{argmax} \Pr(\text{class} \mid \text{image-features}) \)
  - For example, using a Bayesian Network (naïve Bayes is popular)
- Need “good” features
  - Good = adequate for the task
  - Reliable and easy to compute
  - Informative about the class
  - Little else matters – need not be interpretable
  - Consider NLP bag-of-words model: build big joint \( \Pr(\text{topic} \land \text{features}) \)
  - Perceptron and logistic regression embody a discriminative model
What is a Good Feature?

• Not “palm tree” or “toaster” (why?)
• Not individual pixels (why?)
• Not blocks or patches of pixels (why?)
• Successful systems can (often)
  – say “car” (vs. face or airplane) without necessarily finding the car
  – say “leopard” (vs. dog or elephant) even though the leopard has been excised from the image