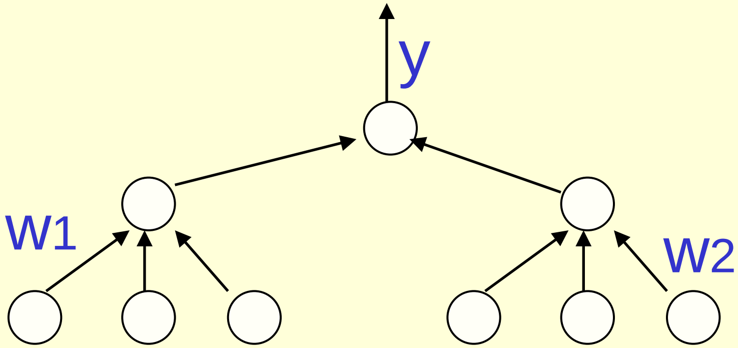
CSC321: Neural Networks

Lecture 13: Learning without a teacher: Autoencoders and Principal Components Analysis

Geoffrey Hinton

Three problems with backpropagation

- Where does the supervision come from?
  - Most data is unlabelled
    - The vestibular-ocular reflex is an exception.
- How well does the learning time scale?
  - Its is impossible to learn features for different parts of an image independently if they all use the same error signal.



- Can neurons implement backpropagation?
  - Not in the obvious way.
    - but getting derivatives from later layers is so important that evolution may have found a way.

Three kinds of learning

- Supervised Learning: this models p(y|x)
  - Learn to predict a real valued output or a class label from an input.
- Reinforcement learning: this just tries to have a good time
  - Choose actions that maximize payoff
- Unsupervised Learning: this models p(x)
  - Build a causal generative model that explains why some data vectors occur and not others

    or

  - Learn an energy function that gives low energy to data and high energy to non-data

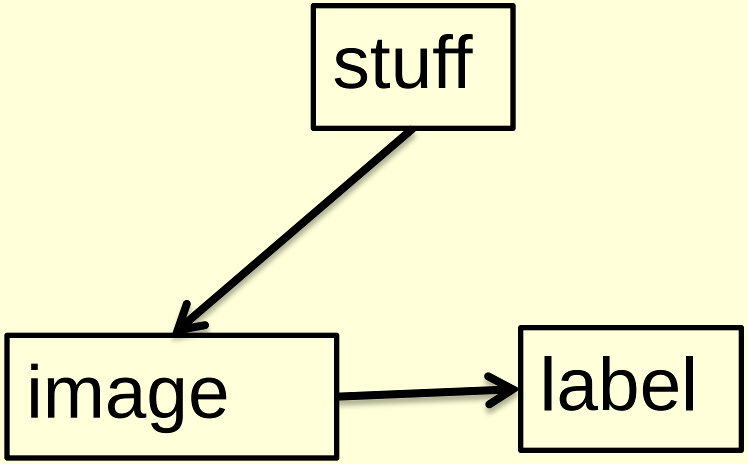
    or

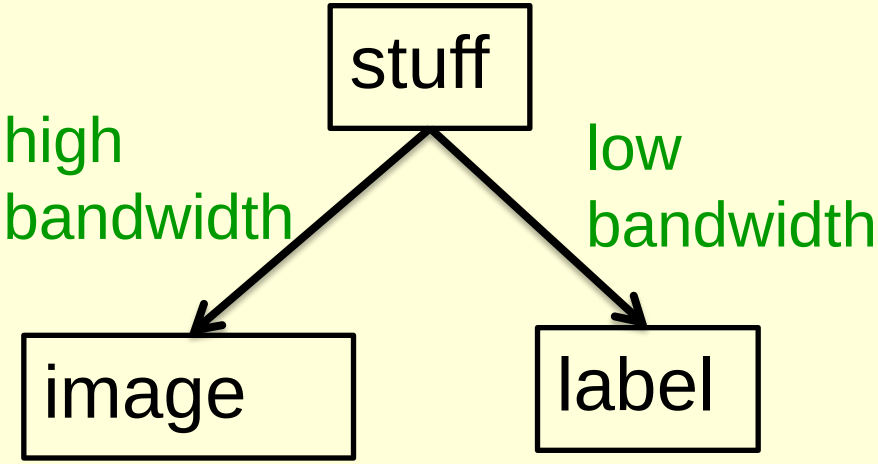
  - Discover interesting features; separate sources that have been mixed together; find temporal invariants etc. etc.

The Goals of Unsupervised Learning

- The general goal of unsupervised learning is to discover useful structure in large data sets without requiring a target desired output or reinforcement signal.
  - It is not obvious how to turn this general goal into a specific objective function that can be used to drive the learning.
- A more specific goal:
  - Create representations that are better for subsequent supervised or reinforcement learning.

Why unsupervised pre-training makes sense





If image-label pairs were generated this way, it would make sense to try to go straight from images to labels.

For example, do the pixels have even parity?

If image-label pairs are generated this way, it makes sense to first learn to recover the stuff that caused the image by inverting the high bandwidth pathway.

Another Goal of Unsupervised Learning

- Improve learning speed for high-dimensional inputs
  - Allow features within a layer to learn independently
  - Allow multiple layers to be learned greedily.
  - Improve the speed of supervised learning by removing directions with high curvature of the error surface.
    - This can be done by using Principal Components Analysis to pre-process the input vectors.

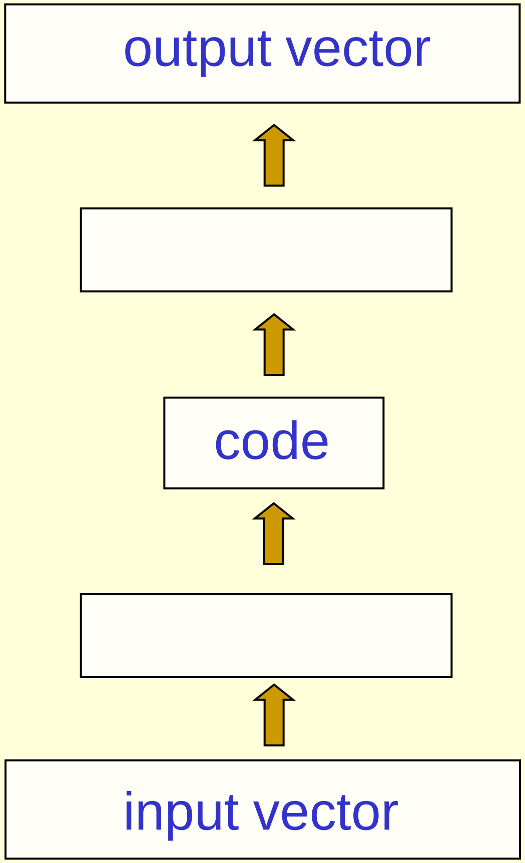
Another Goal of Unsupervised Learning

- Build a density model of the data vectors.
  - This assigns a “score” or probability to each possible datavector.

- There are many ways to use a good density model.
  - Classify by seeing which model likes the test case data most
  - Monitor a complex system by noticing improbable states.
  - Extract interpretable factors (causes or constraints).

Using backprop for unsupervised learning

- Try to make the output be the same as the input in a network with a central bottleneck.
  - The activities of the hidden units in the bottleneck form an efficient code.
    - The bottleneck does not have room for redundant features.
  - Good for extracting independent features (as in the family trees)



Self-supervised backprop in a linear network

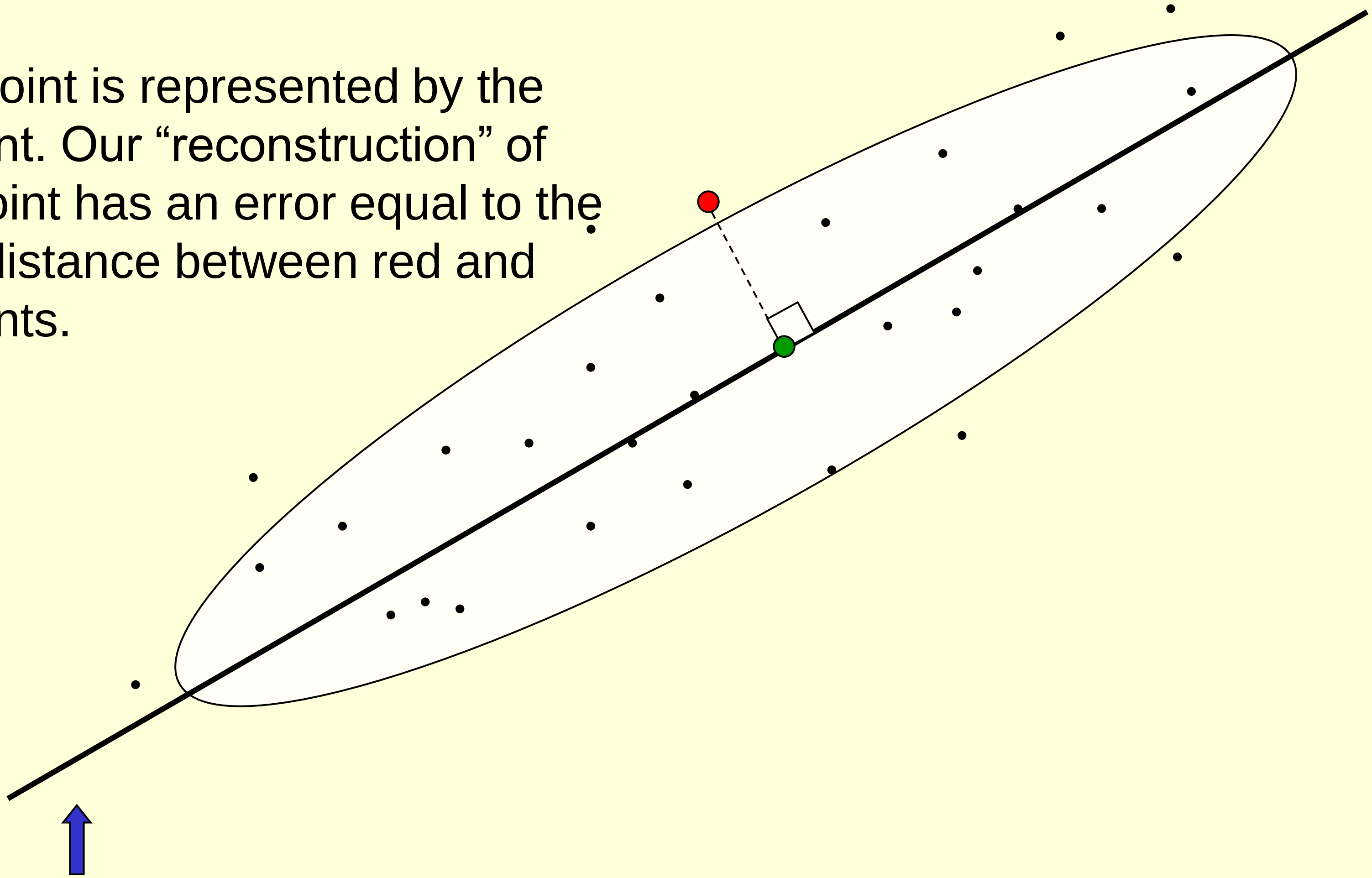
- If the hidden and output layers are linear, it will learn hidden units that are a linear function of the data and minimize the squared reconstruction error.
  - This is exactly what Principal Components Analysis does.
- The M hidden units will span the same space as the first M principal components found by PCA
  - Their weight vectors may not be orthogonal
  - They will tend to have equal variances

Principal Components Analysis

- This takes N-dimensional data and finds the M orthogonal directions in which the data have the most variance
  - These M principal directions form a subspace.
  - We can represent an N-dimensional datapoint by its projections onto the M principal directions
    - This loses all information about where the datapoint is located in the remaining orthogonal directions.
  - We reconstruct by using the mean value (over all the data) on the N-M directions that are not represented.
    - The reconstruction error is the sum over all these unrepresented directions of the squared differences from the mean.

A picture of PCA with N=2 and M=1

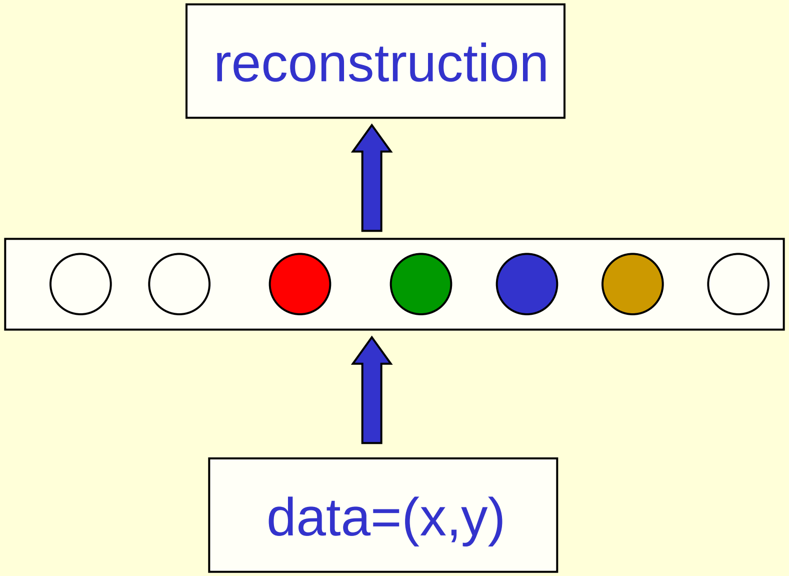
The red point is represented by the green point. Our “reconstruction” of the red point has an error equal to the squared distance between red and green points.

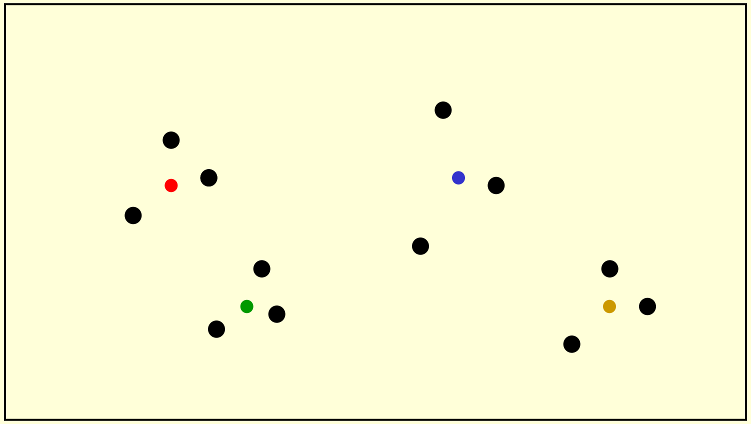


First principal component:
Direction of greatest variance

Self-supervised backprop and clustering

- If we force the hidden unit whose weight vector is closest to the input vector to have an activity of 1 and the rest to have activities of 0, we get clustering.
  - The weight vector of each hidden unit represents the center of a cluster.
  - Input vectors are reconstructed as the nearest cluster center.





Clustering and backpropagation

- We need to tie the input->hidden weights to be the same as the hidden->output weights.
  - Usually, we cannot backpropagate through binary hidden units, but in this case the derivatives for the input->hidden weights all become zero!
    - If the winner doesn't change – no derivative
    - The winner changes when two hidden units give exactly the same error – no derivative
- So the only error-derivative is for the output weights. This derivative pulls the weight vector of the winning cluster towards the data point. When the weight vector is at the center of gravity of a cluster, the derivatives all balance out because the c. of g. minimizes squared error.

A spectrum of representations

- PCA is powerful because it uses distributed representations but limited because its representations are linearly related to the data
  - Autoencoders with more hidden layers are not limited this way.
- Clustering is powerful because it uses very non-linear representations but limited because its representations are local (not componential).
- We need representations that are both distributed and non-linear
  - Unfortunately, these are typically very hard to learn.

	Local	Distributed
Linear		PCA
non-linear	clustering	What we need