Neural Networks

Some of these slides based on content from Seyong Kim and Nando de Freitas
Nando's youtube lectures are really great! See last slide for link

Video from last class

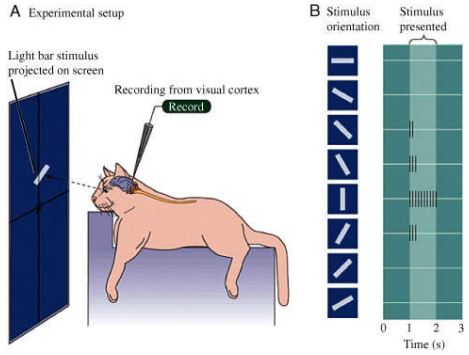
- https://www.youtube.com/watch?v=0qVOUD76JOg

More fun

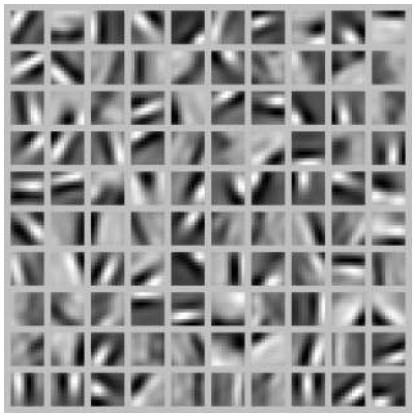
- https://aiexperiments.withgoogle.com/quick-draw

An Aside

- In this video, the static noise you hear is a representation of the neurons firing in response to the visual stimulus
  - https://www.youtube.com/watch?v=jw6nBWo21Zk

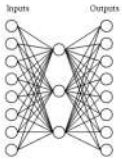


NNs learn something similar



http://deeplearning.stanford.edu/wiki/index.php/Visualizing_a_Trained_Autoencoder

The Autoencoder



When input = output the net is called an autoencoder

A target function:

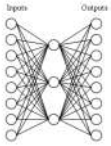
Input		Output
10000000	→	10000000
01000000	→	01000000
00100000	→	00100000
00010000	→	00010000
00001000	→	00001000
00000100	→	00000100
00000010	→	00000010
00000001	→	00000001

Can this be learned??

Wait, is this even interesting? It is learning the identity function…?

The Autoencoder

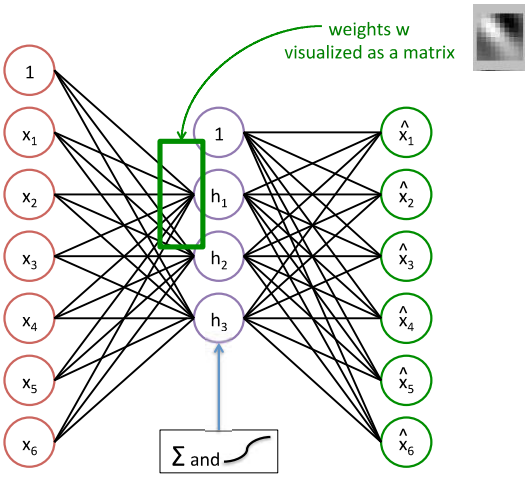
A network:



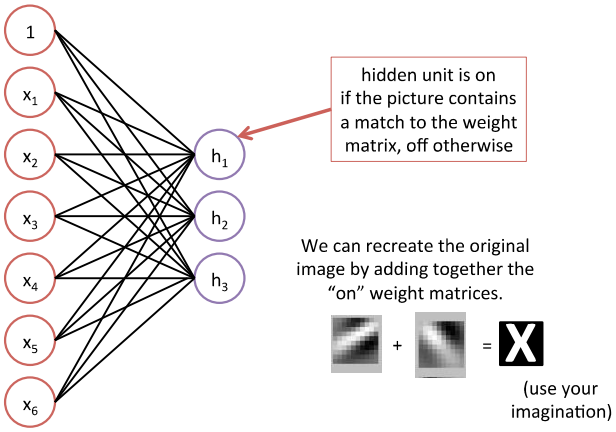
Learned hidden layer representation:

Input	Hidden Values	Output
10000000	→ .89 .04 .08 →	10000000
01000000	→ .01 .11 .88 →	01000000
00100000	→ .01 .97 .27 →	00100000
00010000	→ .99 .97 .71 →	00010000
00001000	→ .03 .05 .02 →	00001000
00000100	→ .22 .99 .99 →	00000100
00000010	→ .80 .01 .98 →	00000010
00000001	→ .60 .94 .01 →	00000001

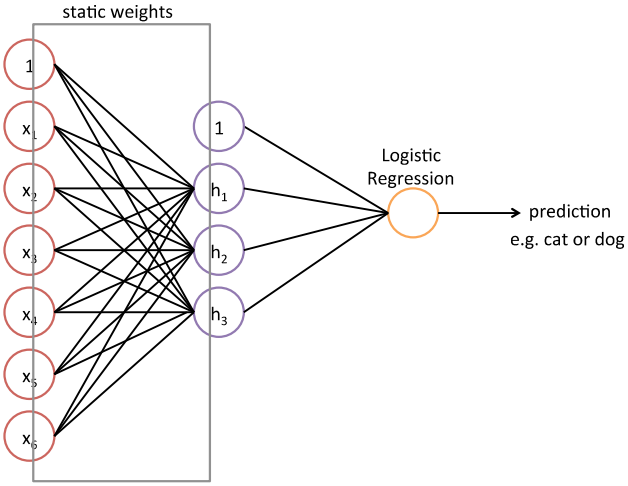
The Autoencoder



The Autoencoder



The Autoencoder



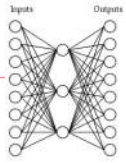
The Autoencoder

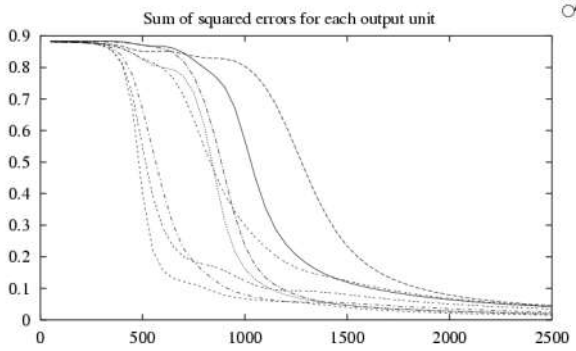
The error function

$$\sum_{i=1}^{N}\sum_{j=1}^{p}(x_i - \hat{x}_{i,j})^2 + \sum_{i=1}^{N}\sum_{j=1}^{k} g(W_j x_i)$$

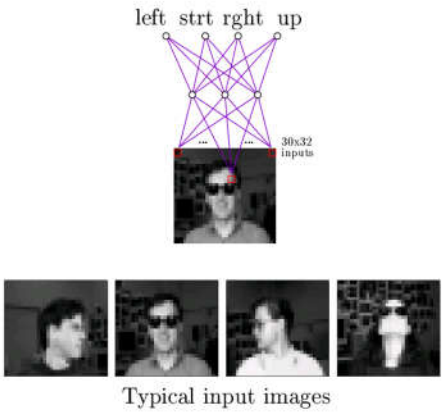
g is a regularizer that wants only a few neurons to fire for each picture, i.e. only a few of the hidden nodes (h) to be non-zero

Training



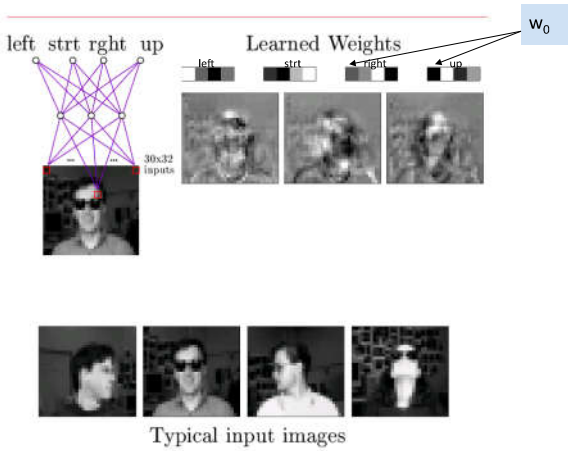


Neural Nets for Face Recognition



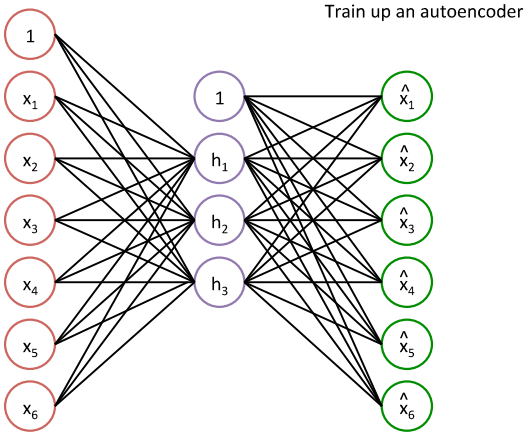
90% accurate learning head pose, and recognizing 1-of-20 faces

Learned Hidden Unit Weights



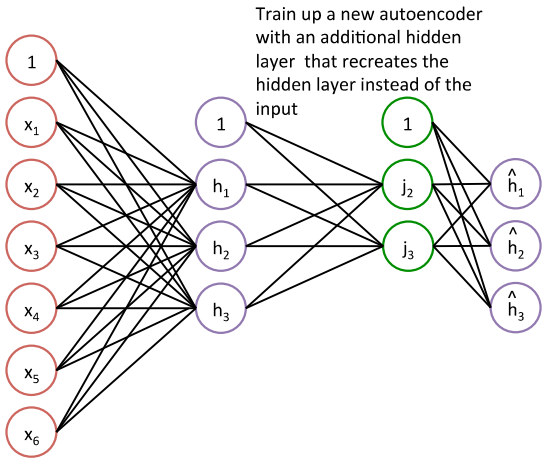
http://www.cs.cmu.edu/~tom/faces.html

The Deep Autoencoder



[Hinton & Salakhutdinov, 2006]

The Deep Autoencoder

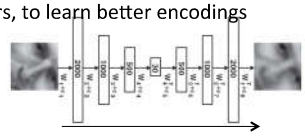


[Hinton & Salakhutdinov, 2006]

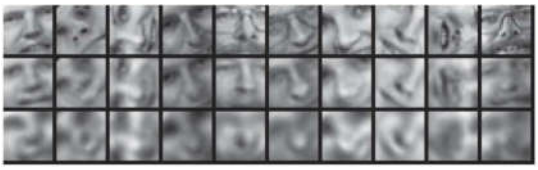
Deep Belief Networks

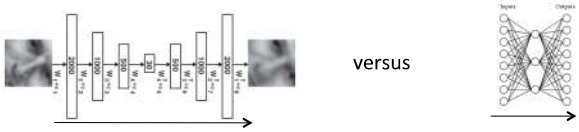
- Problem: training networks with many hidden layers doesn't work very well
  - local minima, very slow training if initialize with zero weights
- Deep belief networks
  - autoencoder networks to learn low dimensional encodings

  - but more layers, to learn better encodings



Deep Belief Networks



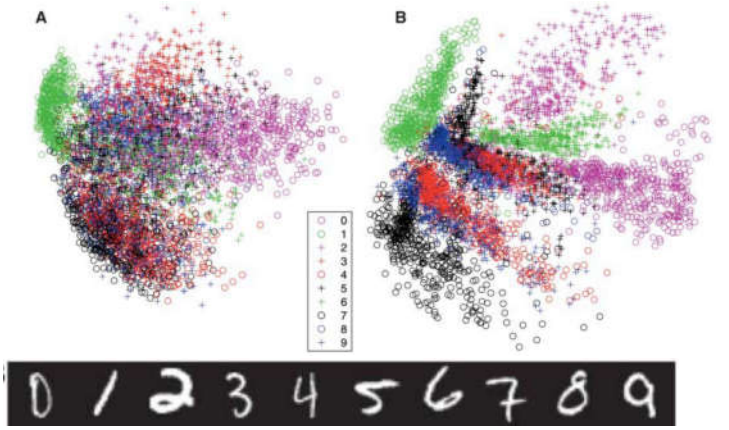


Encoding of digit images in two dimensions

[Hinton & Salakhutdinov, 2006]

784 pixel image -> 2 dimension linear encoding (PCA)

784-1000-500-250-2 DBNet



Resources

- Programming resources for training your own NNs
  - Theano http://deeplearning.net/software/theano/tutorial/index.html#tutorial
  - Tensorflow https://www.tensorflow.org/
- Short course on deep learning (Nando De Freitas)
  - https://www.youtube.com/playlist?list=PLjK8ddCbDMphIMSXn-w1IjyYpHU3DaUYw
- Commentary on AlphaGo
  - https://www.youtube.com/watch?v=UMm0XaCFTJQ
  - https://www.youtube.com/watch?v=g-dKXOlsf98
- Other fun videos
  - Geoff Hinton is in this one! Neural Net stuff is towards the end
    - https://www.youtube.com/watch?v=yxxRAHVtafI
  - Fei Fei Li's Ted Talk
    - https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures?language=en