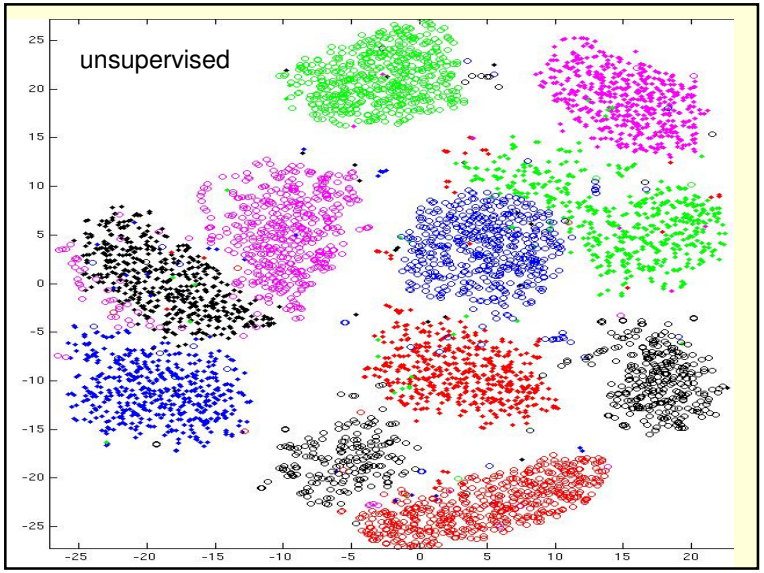
CSC321 Lecture 25:

More on deep autoencoders
&
Using stacked, conditional RBM's for modeling sequences

Geoffrey Hinton
University of Toronto

Do the 30-D codes found by the autoencoder preserve the class structure of the data?

- Take the activity patterns in the top layer and display them in 2-D using a new form of non-linear multidimensional scaling.
- Will the learning find the natural classes?

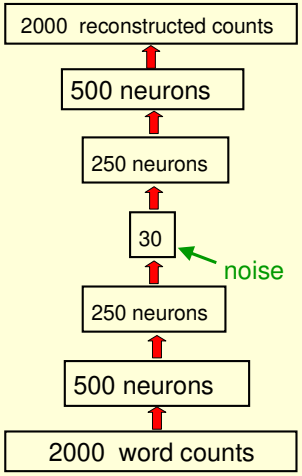


The fastest possible way to find similar documents

- Given a query document, how long does it take to find a shortlist of 10,000 similar documents in a set of one billion documents?
  - Would you be happy with one millesecond?

Finding binary codes for documents

- Train an auto-encoder using 30 logistic units for the code layer.
- During the fine-tuning stage, add noise to the inputs to the code units.
  - The "noise" vector for each training case is fixed. So we still get a deterministic gradient.
  - The noise forces their activities to become bimodal in order to resist the effects of the noise.
  - Then we simply round the activities of the 30 code units to 1 or 0.



Making address space semantic

- At each 30-bit address, put a pointer to all the documents that have that address.
- Given the 30-bit code of a query document, we can perform bit-operations to find all similar binary codes.
  - Then we can just look at those addresses to get the similar documents.
  - The "search" time is independent of the size of the document set and linear in the size of the shortlist.

Where did the search go?

- Many document retrieval methods rely on intersecting sorted lists of documents.
  - This is very efficient for exact matches, but less good for partial matches to a large number of descriptors.
- We are making use of the fact that a computer can intersect 30 lists each of which contains half a billion documents in a single machine instruction.
  - This is what the memory bus does.

How good is a shortlist found this way?

- We have only implemented it for a million documents with 20-bit codes --- but what could possibly go wrong?
  - A 20-D hypercube allows us to capture enough of the similarity structure of our document set.
- The shortlist found using binary codes actually improves the precision-recall curves of TF-IDF.
  - Locality sensitive hashing (the fastest other method) is 50 times slower and always performs worse than TF-IDF alone.

Time series models

- Inference is difficult in directed models of time series if we use distributed representations in the hidden units.
- So people tend to avoid distributed representations and use much weaker methods (e.g. HMM's) that are based on the idea that each visible frame of data has a single cause (e.g. it came from one hidden state of the HMM)

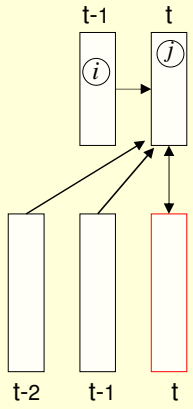
Time series models

- If we really need distributed representations (which we nearly always do), we can make inference much simpler by using three tricks:
  - Use an RBM for the interactions between hidden and visible variables. This ensures that the main source of information wants the posterior to be factorial.
  - Include short-range temporal information in each time-slice by concatenating several frames into one visible vector.
  - Treat the hidden variables in the previous time slice as additional fixed inputs (no smoothing).

The conditional RBM model

- Given the data and the previous hidden state, the hidden units at time t are conditionally independent.
  - So online inference is very easy if we do not need to propagate uncertainty about the hidden states.
- Learning can be done by using contrastive divergence.
  - Reconstruct the data at time t from the inferred states of the hidden units.
  - The temporal connections between hiddens can be learned as if they were additional biases

$$\Delta w_{ij} = s_i \left(< s_j >_{data} - < s_j >_{recon} \right)$$

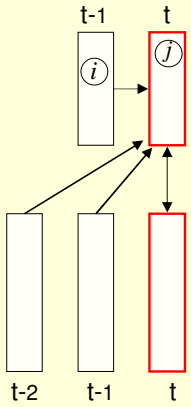


Comparison with hidden Markov models

- The inference procedure is incorrect because it ignores the future.
- The learning procedure is wrong because the inference is wrong and also because we use contrastive divergence.
- But the model is exponentially more powerful than an HMM because it uses distributed representations.
  - Given N hidden units, it can use N bits of information to constrain the future. An HMM only uses log N bits.
  - This is a huge difference if the data has any kind of componential structure. It means we need far fewer parameters than an HMM, so training is not much slower, even though we do not have an exact maximum likelihood algorithm.

Generating from a learned model

- Keep the previous hidden and visible states fixed
  - They provide a time-dependent bias for the hidden units.
- Perform alternating Gibbs sampling for a few iterations between the hidden units and the most recent visible units.
  - This picks new hidden and visible states that are compatible with each other and with the recent history.



Three applications

- Hierarchical non-linear filtering for video sequences (Sutskever and Hinton).
- Modeling motion capture data (Taylor, Hinton & Roweis).
- Predicting the next word in a sentence (Mnih and Hinton).

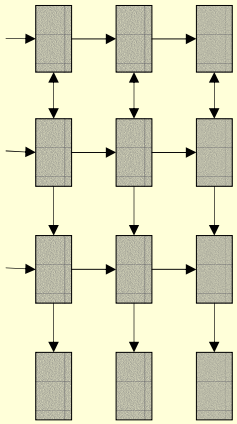
An early application (Sutskever)

- We first tried CRBM's for modeling images of two balls bouncing inside a box.
- There are 400 logistic pixels. The net is not told about objects or coordinates.
- It has to learn perceptual physics.
- It works better if we add "lateral" connections between the visible units. This does not mess up Contrastive Divergence learning.

Show Ilya Sutskever's movies

A hierarchical version

- We developed hierarchical versions that can be trained one layer at a time.
  - This is a major advantage of CRBM's.
- The hierarchical versions are directed at all but the top two layers.
- They worked well for filtering out nasty noise from image sequences.



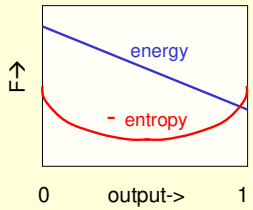
An application to modeling motion capture data

- Human motion can be captured by placing reflective markers on the joints and then using lots of infrared cameras to track the 3-D positions of the markers.
- Given a skeletal model, the 3-D positions of the markers can be converted into the joint angles plus 6 parameters that describe the 3-D position and the roll, pitch and yaw of the pelvis.
  - We only represent changes in yaw because physics doesn't care about its value and we want to avoid circular variables.

An RBM with real-valued visible units

(you don't have to understand this slide!)

- In a mean-field logistic unit, the total input provides a linear energy-gradient and the negative entropy provides a containment function with fixed curvature. So it is impossible for the value 0.7 to have much lower free energy than both 0.8 and 0.6. This is no good for modeling real-valued data.
- Using Gaussian visible units we can get much sharper predictions and alternating Gibbs sampling is still easy, though learning is slower.



$$E(\mathbf{v},\mathbf{h}) = \sum_{i \,\varepsilon\, vis} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j \,\varepsilon\, hid} b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j \, w_{ij}$$

Modeling multiple types of motion

- We can easily learn to model walking and running in a single model.
- This means we can share a lot of knowledge.
- It should also make it much easier to learn nice transitions between walking and running.

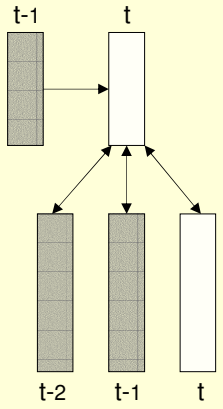
Show Graham Taylor's movies

Statistical language modelling

- Goal: Model the distribution of the next word in a sentence.
- N-grams are the most widely used statistical language models.
  - They are simply conditional probability tables estimated by counting n-tuples of words.
  - Curse of dimensionality: lots of data is needed if n is large.

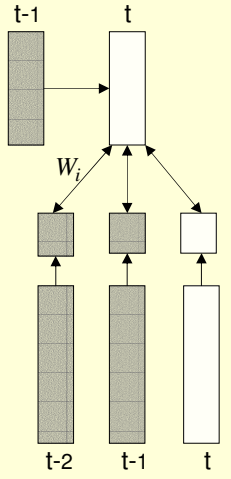
An application to language modeling

- Use the previous hidden state to transmit hundreds of bits of long range semantic information (don't try this with an HMM)
  - The hidden states are only trained to help model the current word, but this causes them to contain lots of useful semantic information.
- Optimize the CRBM to predict the conditional probability distribution for the most recent word.
  - With 17,000 words and 1000 hiddens this requires 52,000,000 parameters.
  - The corresponding autoregressive model requires 578,000,000 parameters.



Factoring the weight matrices

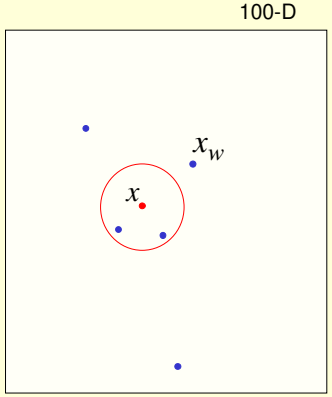
- Represent each word by a hundred-dimensional real-valued feature vector.
  - This only requires 1.7 million parameters.
- Inference is still very easy.
- Reconstruction is done by computing the posterior over the 17,000 real-valued points in feature space for the most recent word.
  - First use the hidden activities to predict a point in the space.
  - Then use a Gaussian around this point to determine the posterior probability of each word.



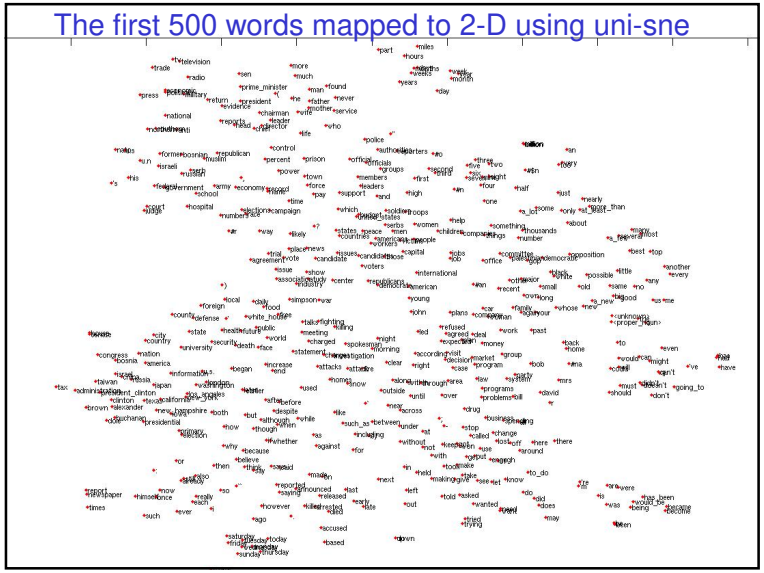
How to compute a predictive distribution across 17000 words.

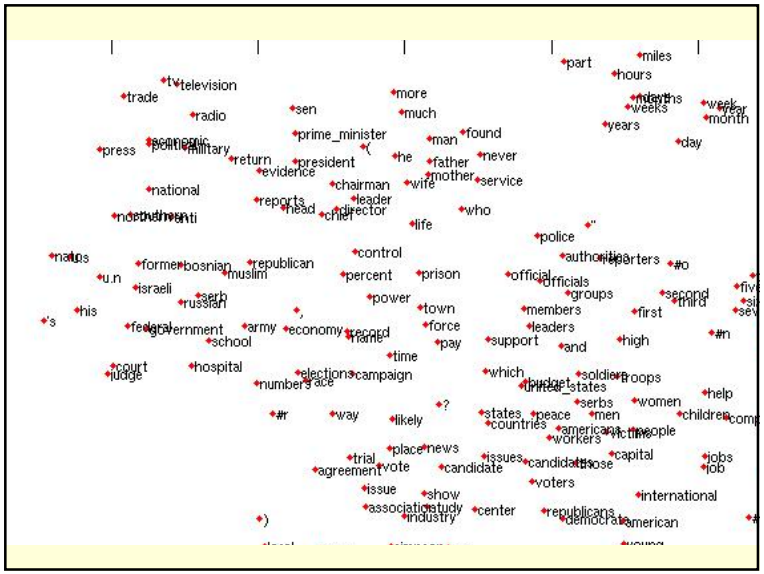
- The hidden units predict a point in the 100-dimensional feature space.
- The probability of each word then depends on how close its feature vector is to this predicted point.

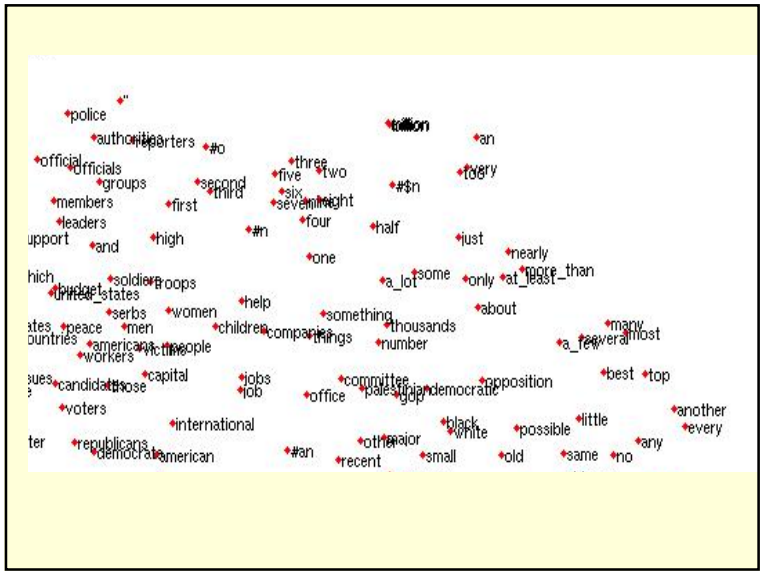
$$p(w) = \frac{\exp\left(\| x - x_w \|^2\right)}{Z}$$

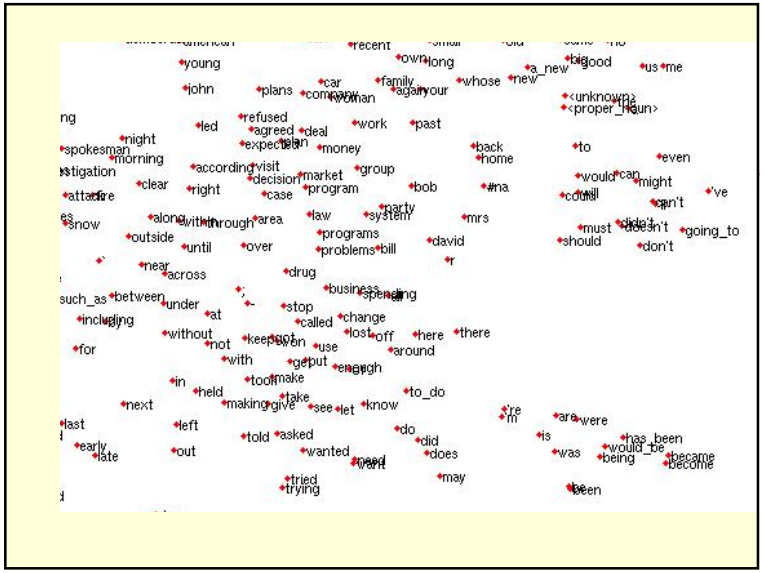


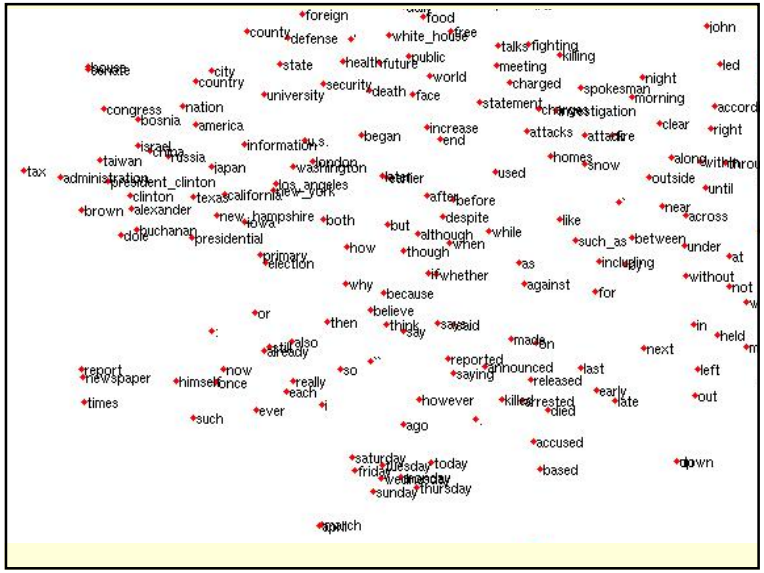
The first 500 words mapped to 2-D using uni-sne








recent
young
own
long
a_new
big
good
us
me
john
plans
car
family
whose
new
<unknown>
<proper_noun>
refused
led
agreed
deal
work
past
night
spokesman
morning
money
back
home
to
even
according
visit
decision
market
group
would
can
might
clear
right
case
program
bob
#na
could
can't
've
snow
area
law
system
party
mrs
must
doesn't
going_to
outside
until
over
programs
problems
bill
david
should
don't
near
across
drug
business
such_as
between
under
at
stop
called
change
including
without
not
keep
lost
off
here
there
for
with
use
around
in
held
took
make
to_do
next
making
give
take
see
let
know
're
'm
are
were
last
left
told
asked
do
did
is
has_been
would_be
early
late
out
wanted
does
was
being
became
become
tried
trying
may
been


foreign
food
john
county
defense
white_house
free
talks
fighting
killing
state
health
future
public
meeting
led
city
country
security
world
charged
spokesman
night
university
death
face
morning
congress
nation
statement
bosnia
america
increase
attacks
clear
right
began
end
information
homes
snow
taiwan
russia
japan
washington
tax
administration
president_clinton
used
outside
until
clinton
texas
california
after
before
brown
alexander
new_hampshire
iowa
both
despite
like
near
across
buchanan
presidential
but
although
while
such_as
between
under
dole
how
though
when
primary
election
as
including
at
if
whether
without
why
because
against
for
not
or
believe
then
think
say
said
in
held
also
already
made
on
next
report
newspaper
now
so
reported
announced
last
left
himself
once
really
saying
released
each
early
times
however
killed
arrested
late
out
ever
i
died
such
ago
accused
saturday
friday
tuesday
today
based
down
sunday
thursday
speech