CSE 446
Sequences, Conclusions

Administrative

- Final exam next week Wed Jun 8 8:30 am
- Last office hours after class today

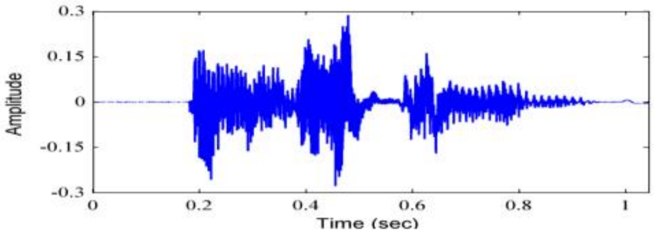
Sequence Models

- High level overview of *structured data*
- What kind of structure? Temporal structure:

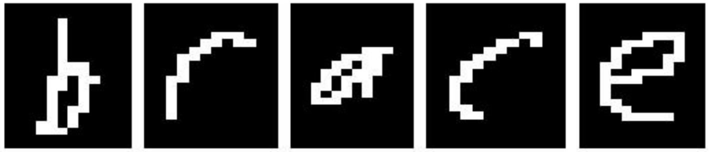
$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_{1,i} \\ \mathbf{x}_{2,i} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}_{T,i} \end{bmatrix}$$

- Sequential data
  - Time-series data

    E.g. Speech



  - Characters in a sentence



  - Base pairs along a DNA strand

Markov Model

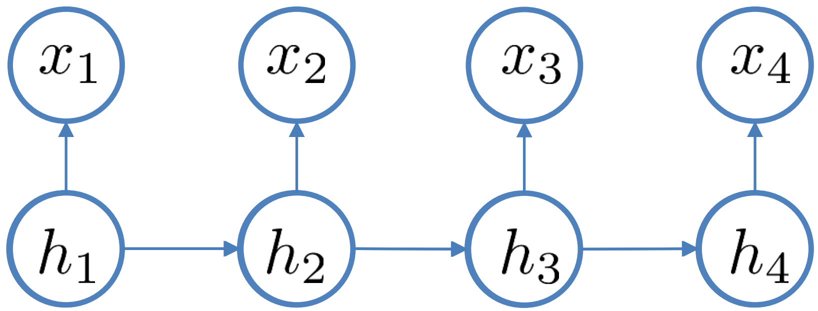
$$\mathbf{x}_i = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \cdot \\ \cdot \\ \cdot \\ x_{T,i} \end{bmatrix}$$

$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$$

$$x_{i,t} \in \{1, 2, \ldots, K\}$$

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2)\ldots p(x_T|x_{T-1})$$

Hidden Markov Model



$p(\mathbf{x}_t | h_t)$ $p(h_t | h_{t-1})$

Hidden Markov Model for Classification

- Condition transitions on label – different transition model for each label
- Use just like naïve Bayes: evaluate probability of a test sequence given every possible label
- Often label is left out of the math, but it's there…

$$p(\mathbf{x}_{1:T}|y=\ell) \propto \sum_{h_1,h_2,\ldots,h_T} p(h_{1:T},\mathbf{x}_{1:T}|y=\ell)$$

$\Updownarrow$

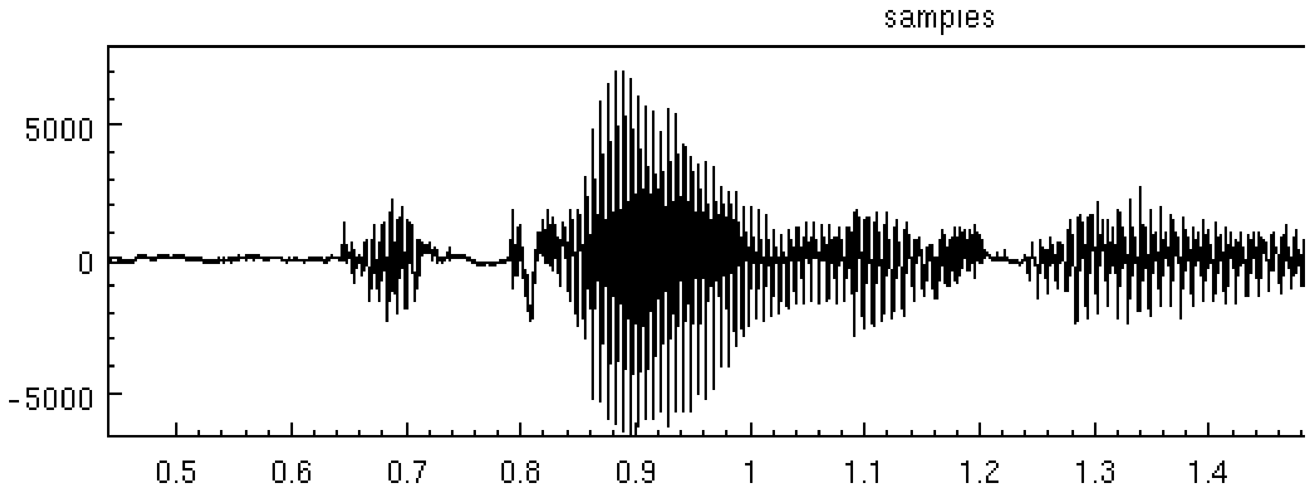
different model for each label

same thing

$$p(\mathbf{x}_{1:T}|y=\ell) \propto \sum_{h_1,h_2,\ldots,h_T} p_\ell(h_{1:T},\mathbf{x}_{1:T})$$

Hidden Markov Model Applications

- Extremely popular for speech recognition
- 1 HMM = 1 phoneme
- Given a segment of audio, figure out which HMM gives it highest probability



Continuous *and* Nonlinear?

- Nonlinear continuous sequence model:

recurrent neural network

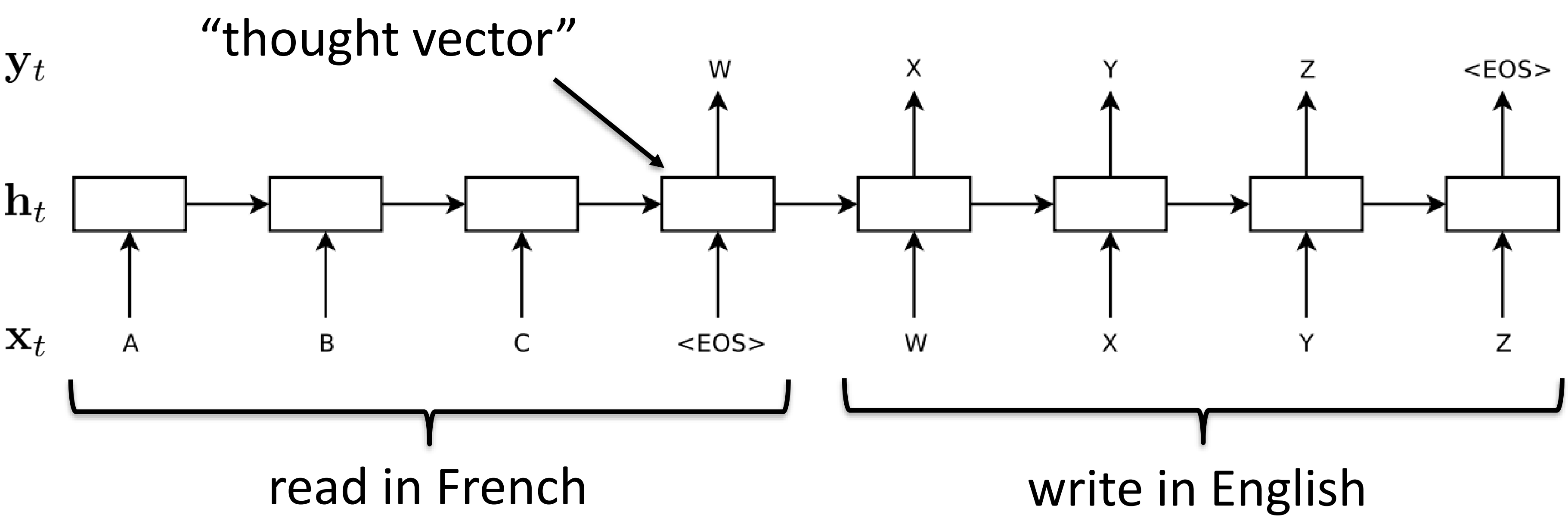
$$p(y_t = k|\mathbf{h}_t) = \frac{\exp(-\mathbf{W}_k \mathbf{h}_t)}{\sum_{k'=1}^{K} \exp(-\mathbf{W}_{k'} \mathbf{h}_t)}$$

$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}_h \mathbf{h}_t + \mathbf{b}_h)$$
$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}_h)$$
$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_x \mathbf{x}_t + \mathbf{W}_y \mathbf{y}_t + \mathbf{b}_h)$$
$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_y \mathbf{y}_t + \mathbf{b}_h)$$

RNN Application: Machine Translation

$$p(y_t = k|\mathbf{h}_t) = \frac{\exp(-\mathbf{W}_k \mathbf{h}_t)}{\sum_{k'=1}^{K} \exp(-\mathbf{W}_{k'} \mathbf{h}_t)}$$

$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_x \mathbf{x}_t + \mathbf{W}_y \mathbf{y}_t + \mathbf{b}_h)$$

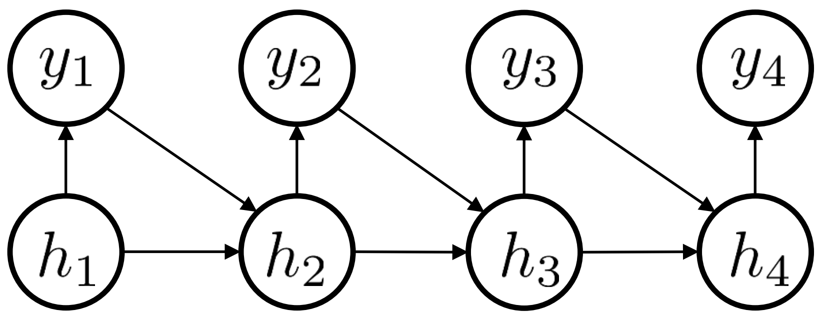


Sutskever et al. 2014

RNN Application: Language Modeling

$$p(y_t = k|\mathbf{h}_t) = \frac{\exp(-\mathbf{W}_k \mathbf{h}_t)}{\sum_{k'=1}^{K} \exp(-\mathbf{W}_{k'} \mathbf{h}_t)}$$

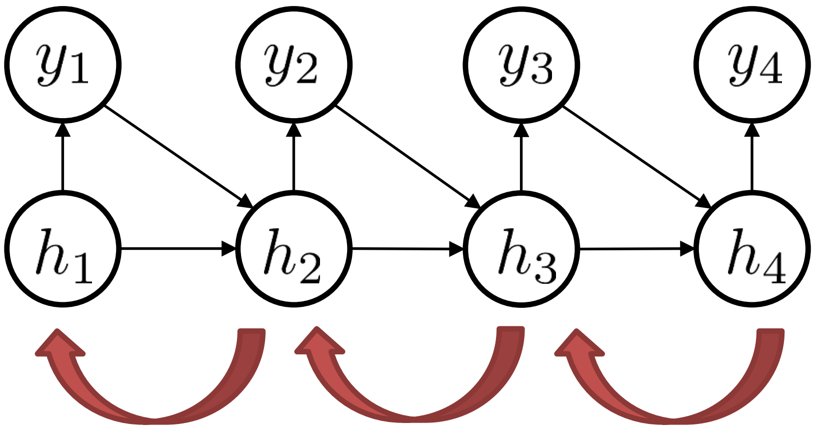
$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_y \mathbf{y}_t + \mathbf{b}_h)$$



RNN Training

- Almost always use backpropagation + stochastic gradient descent/gradient ascent
  - No different than any other neural network
  - Just have many outputs (and inputs)
  - Compute gradients and use chain rule
    - Per time step instead of per layer
    - Math is exactly the same
- But it's very hard to optimize…

Why RNN Training is Hard

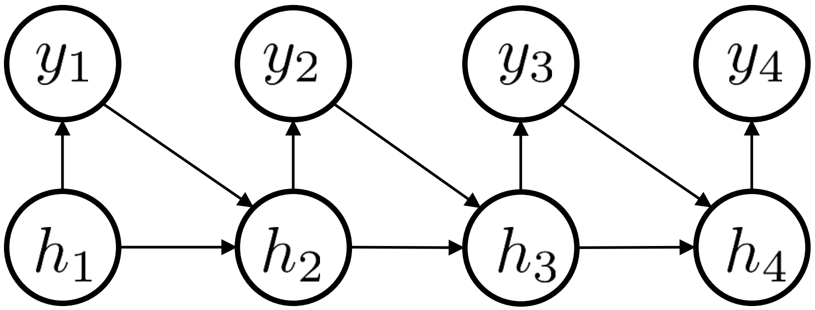


$$\frac{d\mathcal{L}(y_T)}{dh_2} = \underbrace{\frac{d\mathcal{L}(y_T)}{dh_T}\frac{dh_T}{dh_{T-1}}\cdots\frac{dh_3}{dh_2}}_{\substack{\text{lots of multiplication} \\ \text{very unstable numerically}}}$$

- Backpropagation = chain rule
- Derivative multiplied by new matrix at each time step (time step in RNN = layer in NN)
- Lots of multiplication by values less than 1 = gradients become tiny
- Lots of multiplication by values greater than 1 = gradients explode
- Many tricks for effective training
  - Clever nonlinearity (e.g. LSTM – special type of nonlinearity)
  - Better optimization algorithms (more advanced than gradient descent)

RNN Application: Text Generation

- http://www.cs.toronto.edu/~ilya/fourth.cgi



discrete character label

1500-dimensional state

The meaning of life is any older bird. Get into an hour performance, in the first time period in

RNN does Shakespeare

PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

From Andrej Karpathy

RNN does algebraic geometry (maybe it can write my lecture notes?)

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, **??** and the fact that any U affine, see Morphisms, Lemma **??**. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma **??** we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. $\square$

From Andrej Karpathy

RNN does operating system code

```
/*
 * If this error is set, we will need anything right after that BSD.
 */
static void action_new_function(struct s_stat_info *wb)
{
  unsigned long flags;
  int lel_idx_bit = e->edd, *sys & ~((unsigned long) *FIRST_COMPAT);
  buf[0] = 0xFFFFFFFF & (bit << 4);
  min(inc, slist->bytes);
  printk(KERN_WARNING "Memory allocated %02x/%02x, "
    "original MLL instead\n"),
    min(min(multi_run - s->len, max) * num_data_in),
    frame_pos, sz + first_seg);
  div_u64_w(val, inb_p);
  spin_unlock(&disk->queue_lock);
  mutex_unlock(&s->sock->mutex);
  mutex_unlock(&func->mutex);
  return disassemble(info->pending_bh);
}
```

From Andrej Karpathy

RNN does clickbait…

Romney Camp : ‘ I Think You Are A Bad President ‘
Here ‘ s What A Boy Is Really Doing To Women In Prison Is Amazing
L . A . ‘ S First Ever Man Review
Why Health Care System Is Still A Winner
Why Are The Kids On The Golf Team Changing The World ?
2 1 Of The Most Life – Changing Food Magazine Moments Of 2 0 1 3
More Problems For ‘ Breaking Bad ‘ And ‘ Real Truth ‘ Before Death
Raw : DC Helps In Storm Victims ‘ Homes
U . S . Students ‘ Latest Aid Problem
Beyonce Is A Major Woman To Right – To – Buy At The Same Time
Taylor Swift Becomes New Face Of Victim Of Peace Talks
Star Wars : The Old Force : Gameplay From A Picture With Dark Past (Part 2)
Sarah Palin : ‘ If I Don ‘ t Have To Stop Using ‘ Law , Doesn ‘ t Like His Brother ‘ s Talk On His ‘ Big Media ‘
Israeli Forces : Muslim – American Wife ‘ s Murder To Be Shot In The U . S .
And It ‘ s A ‘ Celebrity ‘
Mary J . Williams On Coming Out As A Woman
Wall Street Makes $ 1 Billion For America : Of Who ‘ s The Most Important Republican Girl ?
How To Get Your Kids To See The Light
Kate Middleton Looks Into Marriage Plans At Charity Event
Adorable High – Tech Phone Is Billion – Dollar Media

From Lars Eidnes

Concluding Remarks

- Summary: anatomy of a machine learning problem
- How to tackle a machine learning problem
- Where to go from here
- What we didn’t cover

Anatomy of a Machine Learning Problem

- Data
  - This is what we learn **from**
- Hypothesis space
  - Also called: model class, parameterization (though not all models are parametric…), etc.
  - This is **what** we learn
- Objective
  - Also called: loss function, cost function, etc.
  - This is the **goal** for our algorithm
  - Usually not the same as the overall goal of learning (training error vs generalization error)
- Algorithm
  - This is what optimizes the objective
  - Sometimes the optimization is not exact (e.g. k-means)
  - Sometimes the optimization is heuristic (e.g. decision trees)

How to Tackle a Machine Learning Problem

- Look at your data
  - What is its structure?
  - What domain knowledge do you have?
  - Plot something, cluster something, etc.
- Split into training and validation (remember, it's not a test set if you use it to tune hyperparameters…)
- Define the problem
  - What are the inputs and (if any) outputs?
  - What kind of objective should you use?
    - Usually either a probabilistic generative process, or a discriminative approach
- Choose a few possible hypothesis classes (including features…), experiment
- Troubleshoot & improve
  - Look for overfitting or underfitting
  - Look for overfitting or underfitting
  - Modify hypothesis class and features

Where to go From Here

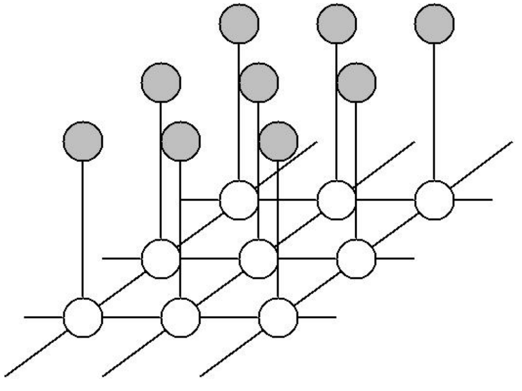
- This course provides a high-level sampling of various ML topics
  - Classification
  - Regression
  - Unsupervised learning
- There is much more depth behind each topic
- Here is a summary of modernized versions of some of the methods we covered

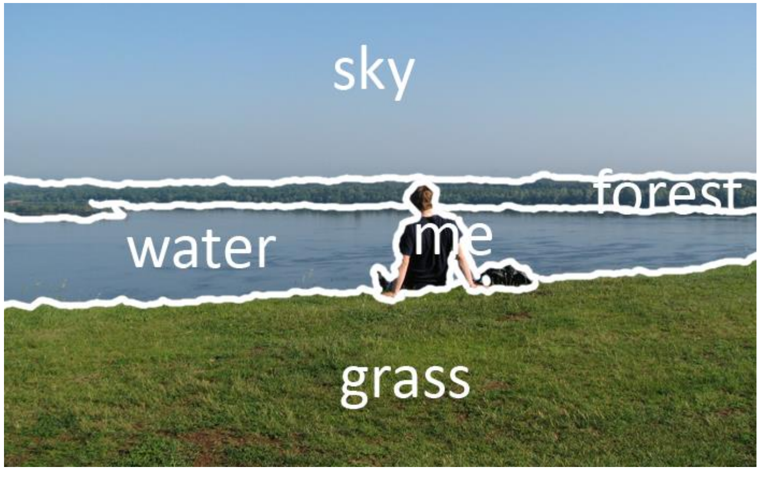
Decision Trees

- Almost never used individually
- Typically used with model ensembles
  - See bagging lecture and section on random forests
- Some of the most popular models in practice

Naïve Bayes

- Generalizes to Bayesian networks
  - Includes Markov models, hidden Markov models, Gaussian mixture models
- Generalizes to Markov random fields
  - Model dependencies on networks



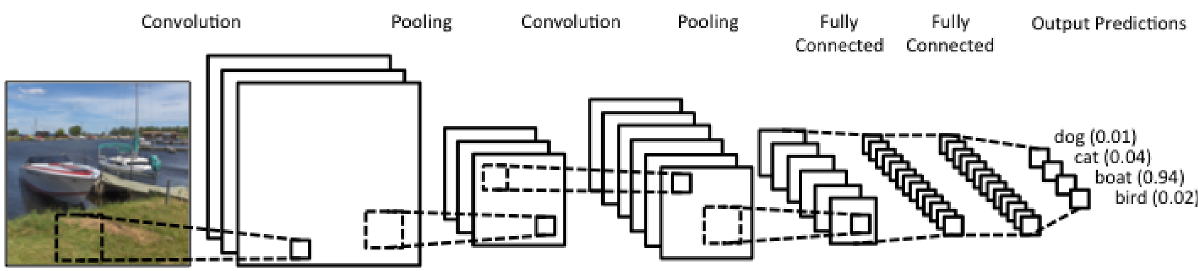


Logistic Regression

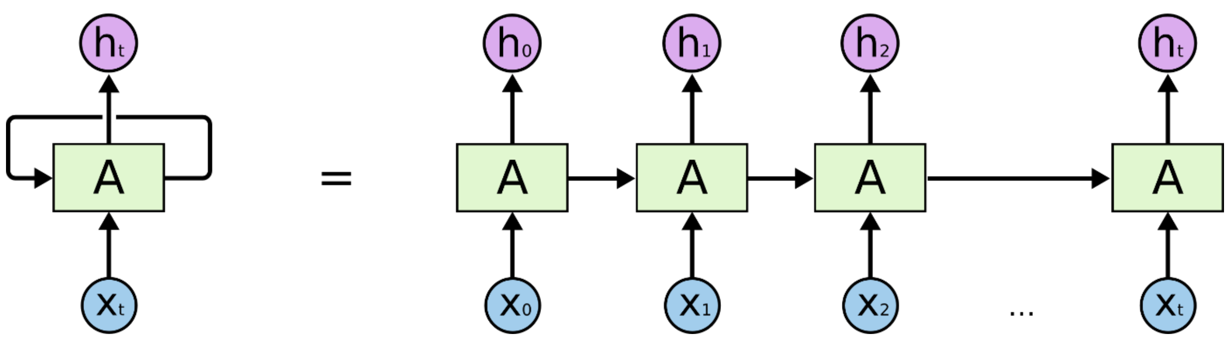
- Generalizes to neural networks
- Very flexible class of models
- Popular for a wide range of applications
  - Same tradeoff as naïve Bayes vs. logistic regression:
    - More data = neural network does well
    - Less data = neural network overfits, probabilistic Bayesian methods tend to do better

Neural Networks

- For image processing: convolutional neural networks



- For language, speech: recurrent neural networks



Neural Networks + Bayesian Networks

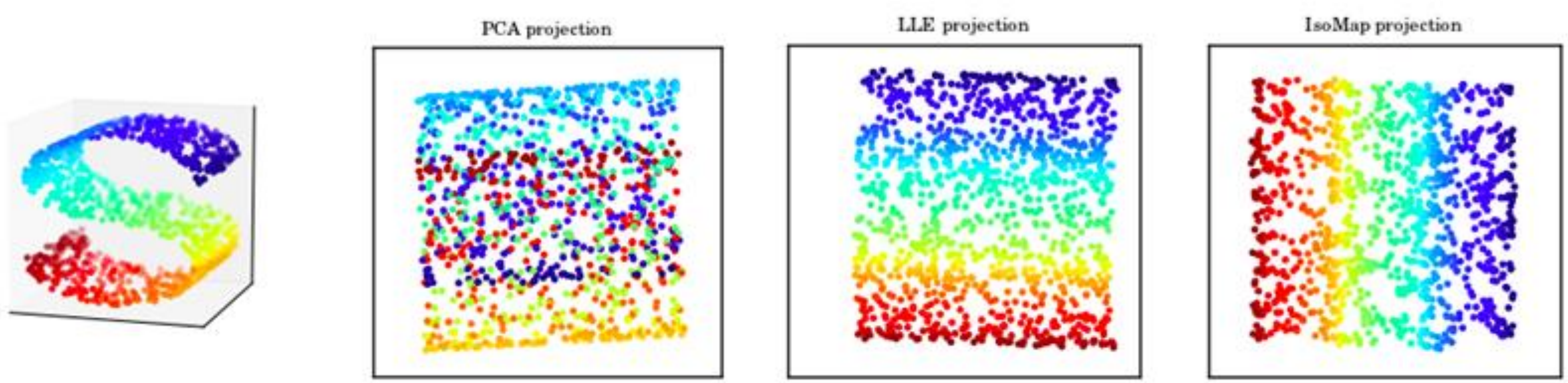
- Bayesian networks are typically generative
  - Can sample (generate) new data from the model
  - Can easily train on partial data (e.g. via EM)
- Neural networks are typically discriminative
  - Can predict label, but can't generate data
  - Hard to deal with partial data
- Generative neural networks?
  - Good for training with lots of unlabeled data and a little bit of labeled data
  - Can hallucinate some interesting images

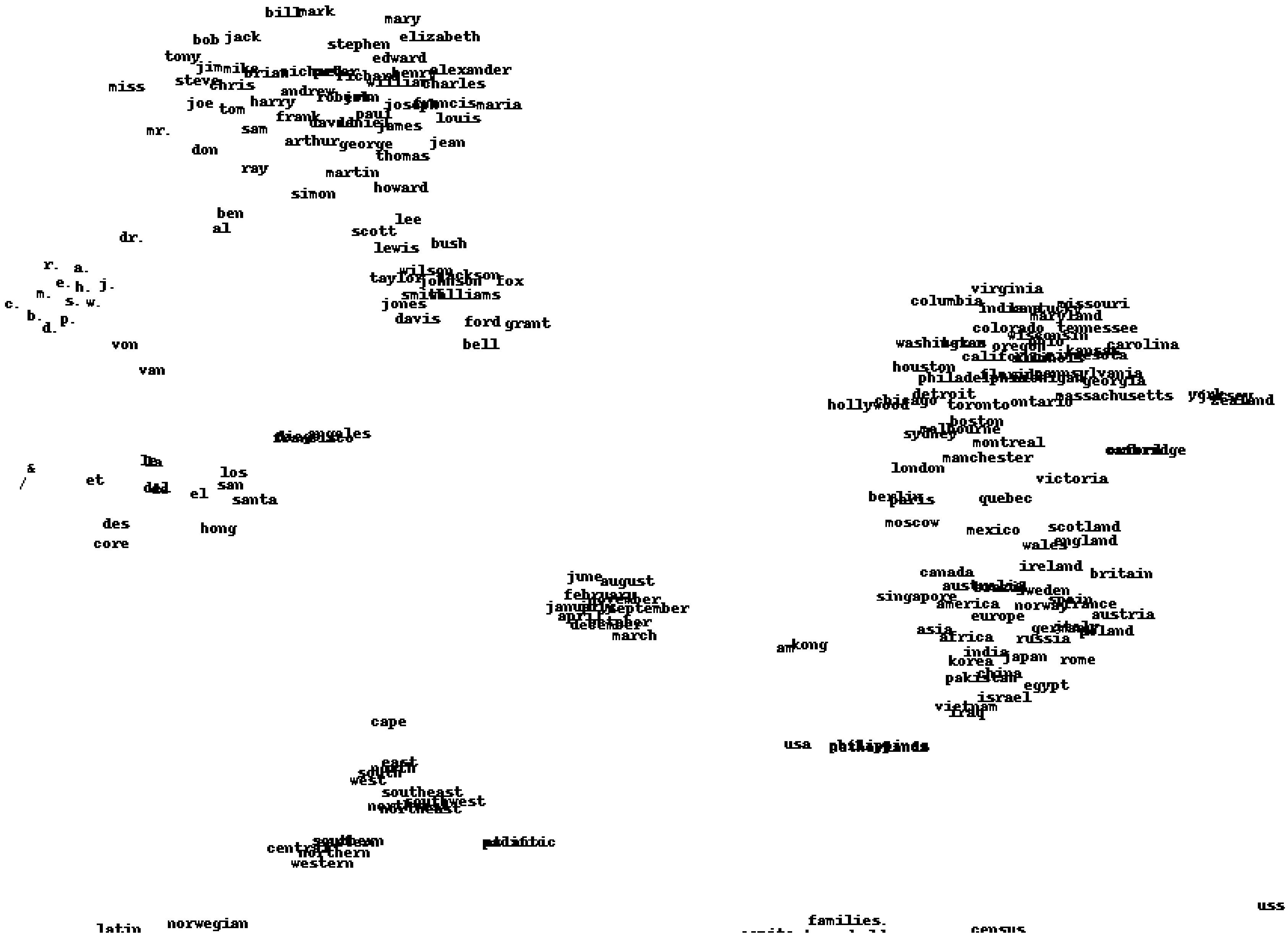
Support Vector Machines & Kernels

- Widely used with kernels
- Kernels allow for linear models to become extremely powerful nonlinear nonparametric models
  - Kernelized SVM
  - Kernelized linear regression (Gaussian process)
- Great when data is very limited

Unsupervised Learning

- Nonlinear dimensionality reduction
  - Reduce dimensionality much further while preserving more information
  - Intuition is to “unfold” nonlinear manifold into a low-dimensional space





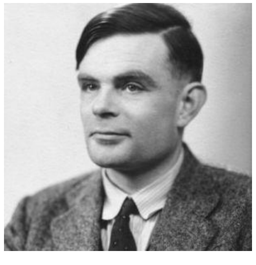
bill
mark
mary
elizabeth
bob
jack
stephen
edward
tony
alexander
charles
miss
steve
chris
andrew
maria
joe
tom
harry
francis
louis
mr.
sam
frank
paul
james
arthur
george
jean
don
thomas
ray
martin
simon
howard
ben
lee
al
scott
dr.
lewis
bush
r.
a.
e.
h.
j.
taylor
johnson
fox
m.
s.
w.
smith
williams
c.
b.
p.
jones
d.
davis
ford
grant
von
bell
van
angeles
&
et
los
/
sam
el
santa
des
hong
core
june
august
february
september
march
cape
southeast
west
northeast
central
northern
western
latin
norwegian
virginia
columbia
missouri
maryland
colorado
tennessee
carolina
houston
georgia
detroit
hollywood
toronto
ontario
massachusetts
boston
montreal
manchester
london
victoria
paris
quebec
moscow
mexico
scotland
england
wales
canada
ireland
britain
singapore
sweden
america
norway
france
europe
austria
asia
africa
russia
kong
india
japan
rome
korea
pakistan
china
egypt
israel
vietnam
iraq
usa
uss
families
census

Concluding Remarks

- Machine learning draws on several disciplines
  - Computer science
  - Statistics
  - Artificial intelligence
- Can be viewed as methods to process data
  - “data science”
- Can be viewed as methods to make machines more intelligent

- This is an engineering course
- Machine learning is engineering, but it is also science
- Scientific question: how to understand (and create) intelligence?
- (classic) artificial intelligence: design algorithms that act intelligently with common sense
  - Heuristic planning
  - Mixture of experts
- Learning: design algorithms that figure out on their own how to act intelligently, from experience

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain.



- Alan Turing