How can machines learn without supervision?

Sanjeev Arora

Princeton University
Institute for Advanced Study

www.unsupervised.princeton.edu
www.offconvex.org (blog)

**Disclaimer: A quirky, personal view…

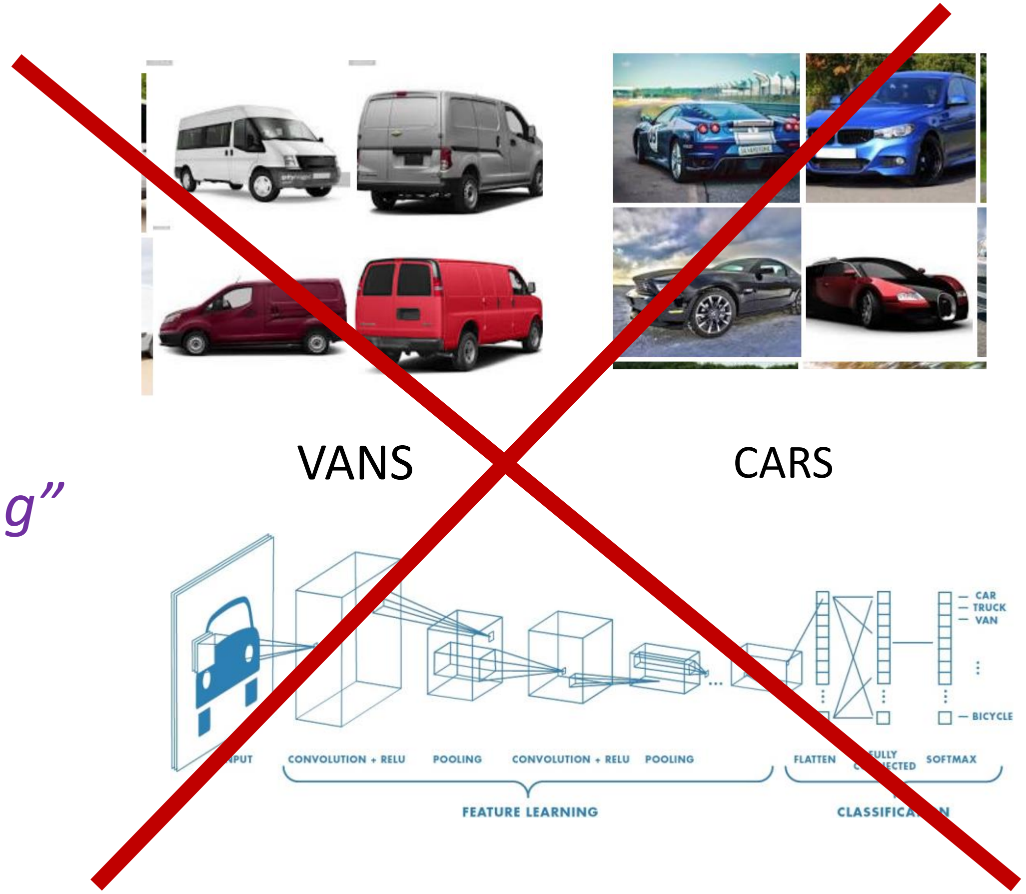
Unsupervised Learning

(arguably, dominant part of human learning)

Learning latent structure in unlabeled data.
(Example: “Meaning” of a paragraph)

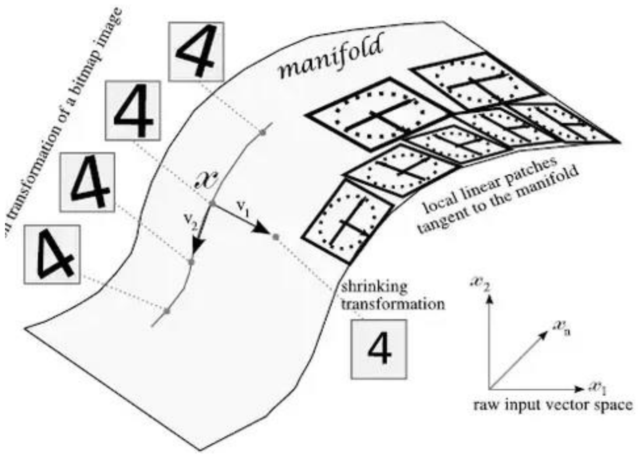
To be contrasted with supervised learning,
which uses labeled data…

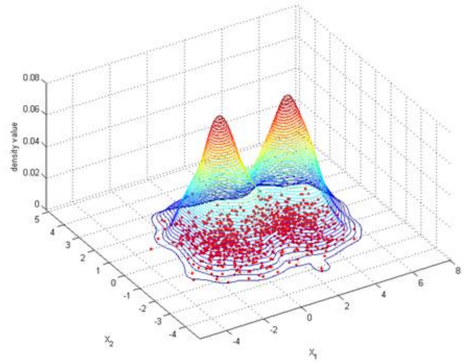
If task is “label a given paragraph with its meaning”
Then # of possible labels is too large!!



Unsupervised learning: varied interpretations

- Low dimensional embeddings:
  PCA, Clustering, Manifold Learning,..
- Find probability distribution for data. $p_\theta(x, h)$
  (e.g x = image in pixel description
  h = high level description ("boy in pajamas")
  inference problem: given x infer most likely h
- Use one part of x to predict rest of x (e.g., using deep net)





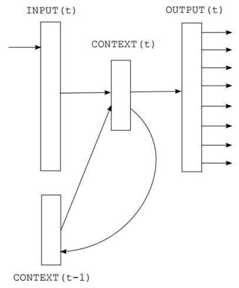
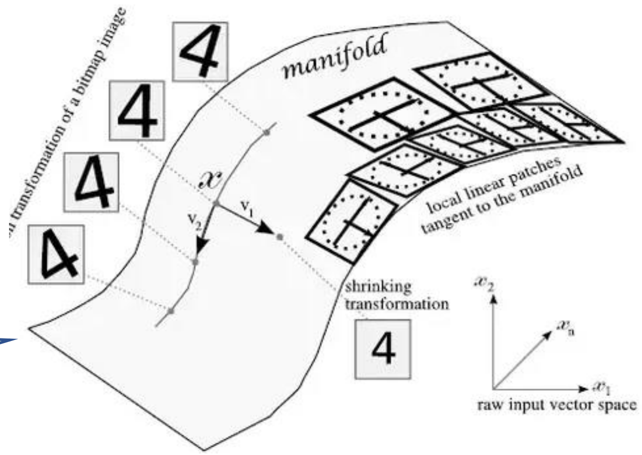


Figure 1: *Simple recurrent neural network.*

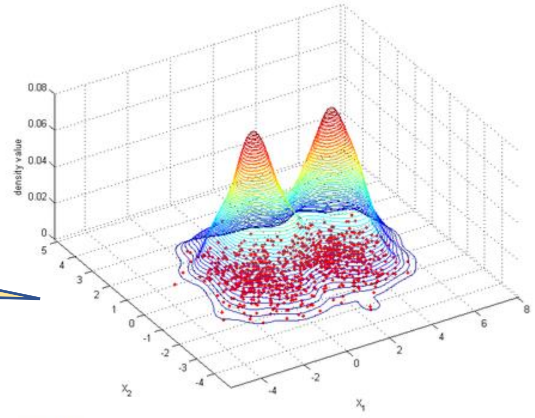
Unsupervised learning: difficulties

PCA, Clustering etc. are too simplistic.

Manifold learning seems v. general, but takes exp(d) time (d = latent dimension)



(i) Ground truth distribution probably has no clean functional form.
(ii) Multilayer neural nets tried, but haven't been as successful
(ii) Computationally difficult (NP-hard)



Usually unclear how to reconcile the three views

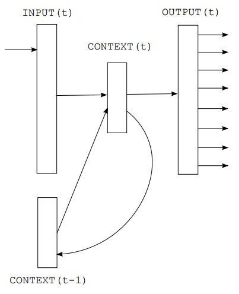
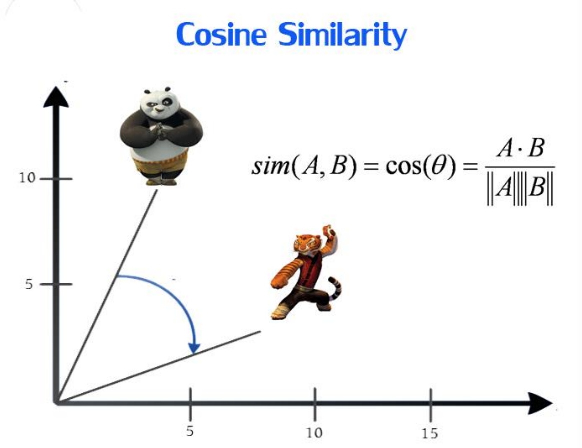


Figure 1: *Simple recurrent neural network.*

Next few slides: Vector representations of "meaning" of text

Simplest subcase: Word embeddings. (used several decades in info retrieval; have interesting properties) . Dimension 100 to 500.

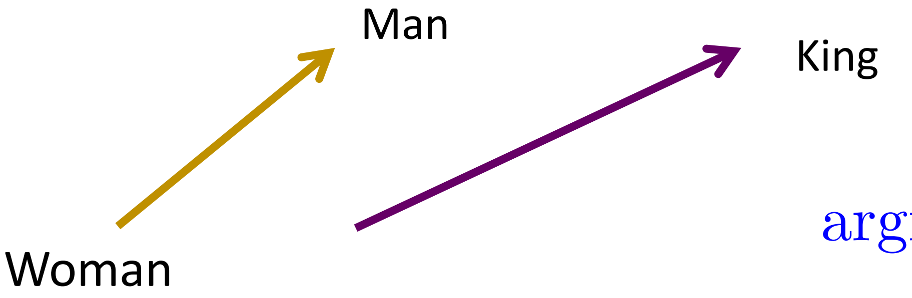


Inner product ≈ human notions of similarity

[Mikolov et al'13]: Solving word analogy tasks via linear algebra on embeddings.

man: woman:: king: ??

France:Paris:: Japan: ??



$$\text{argmin}_w \quad ||v_w - v_{\text{king}} - v_{\text{woman}} + v_{\text{man}}||^2$$

Methods to compute word embeddings

(Distributional hypothesis : “A word is known by what it cooccurs with.” [Firth, Harris’50s])

Train a neural net to predict next word using last few words [Bengio’03; Collobert-Weston’08]

“He stopped at a café and got a …..”

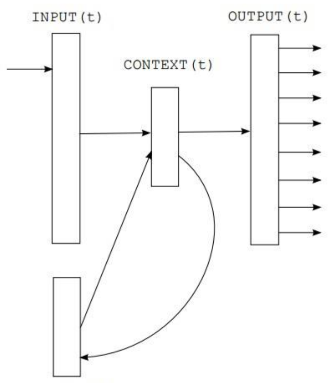
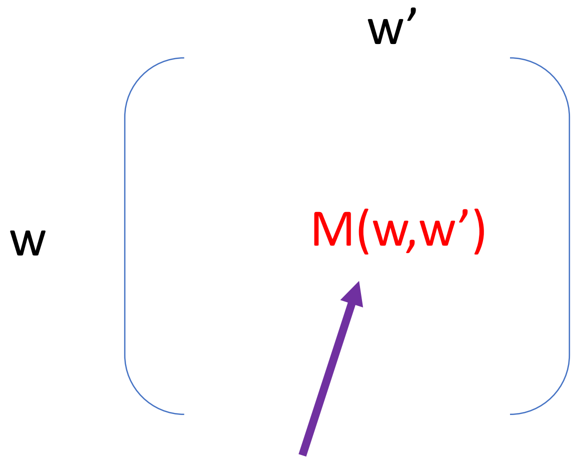


Figure 1: *Simple recurrent neural network.*

$$Pr[w \mid w_1 w_2 .. w_5] \propto \exp(v_w \cdot (\frac{1}{5} \sum_i v_{w_i}))$$

Word2vec [Mikolov et al’13]

Low rank approximation to some matrix (”Log PCA”)

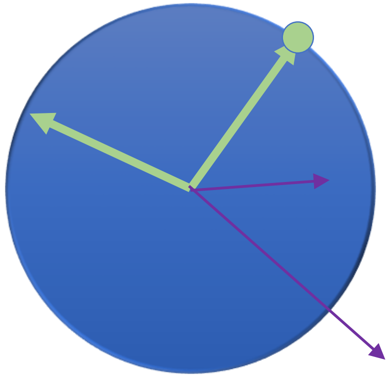


log Pr[words w, w’ co-occur in window of size 5 in corpus]

Our work unifies various viewpoints via a generative model….

Generative model for language [A., Li, Liang, Ma, Risteski TACL'16]

(dynamic version of loglinear topic model, [Mnih-Hinton06])



"Semantic space" inside writer's head; each direction in R^d associated with a discourse (narrow "topic")

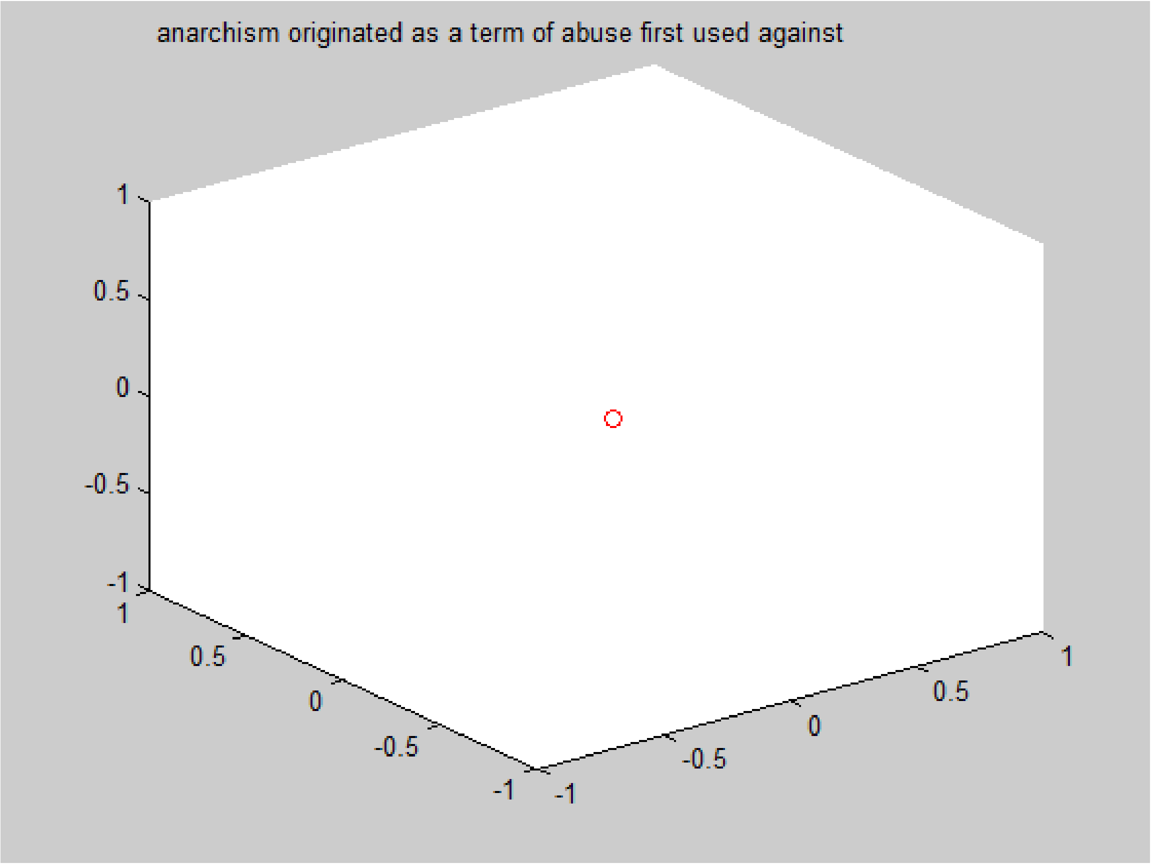
Each word w also associated with a vector v_w in R^d (aka "word embedding"))

Corpus generated by random walk of a discourse vector c on unit sphere in R^d $\Pr[w \text{ is output} \mid c_t] \propto \exp(v_w \cdot c_t)$

[Related: [Hashimoto, Alvarez-Melis, Jaakola TACL'16]

Generative model (illustration)

Discourse vector c_t does random walk;
if context vector at time t is c_t, $\Pr[w \text{ is output} \mid c_t] \propto \exp(v_w \cdot c_t)$



NB: Locally bag of words;

No syntax modeling.

Theoretical results from this model

- Main idea: Integrate out the random walk.
- Thm 1: *Max Likelihood estimation of the v_w 's → training objectives for popular methods for word embeddings (word2vec and GloVe).*
- Thm 2: Relations (eg masculine-feminine) correspond to linear structure as exhibited by analogy solving
- Thm 3: (To appear in TACL'18) An explanation of how different meanings of a polysemous word reside within its word embedding.
  (Short answer: $v_{tie} = \alpha\, v_{tie1} + \beta\, v_{tie2}$)

Side product: Sentence Embeddings

1) The tiger rules this jungle.

2) Milk flowed out from the bottle.

3) Carnegie is a generous man.

4) A lion hunts in a forest.

5) Pittsburgh has great restaurants, does it not?

[A., Liang, Ma ICLR'17] sentence embedding: take weighted combination of its word subtract component along a certain direction c_0

In many tasks, does as well as embeddings from neural nets. (And v. fast to compute!)

New word production model [A., Liang, Ma ICLR'17]

$$\Pr[w \text{ emitted in sentence } s \mid c_s] = \alpha p(w) + (1-\alpha)\frac{\exp\left(\langle \tilde{c}_s, v_w \rangle\right)}{Z_{\tilde{c}_s}},$$

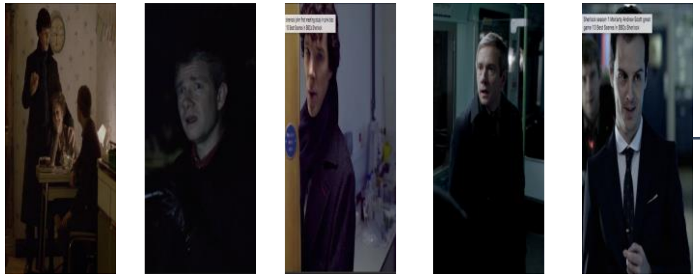
$$\text{where } \tilde{c}_s = \beta c_0 + (1-\beta) c_s, \quad c_0 \perp c_s$$

Each word has a small background probability of being produced out of context…

Some words (eg "just" "when" "there") have a much higher probability of being produced in all contexts. Their vectors will have higher components on c_0

Sample Task: "Decoding" Brain fMRI [Vodrahalli et al, NeuroImage'17]

Movie scenes



Annotations of movie scenes

Sherlock and John talk about the murder in an old room with Mrs. Hudson.

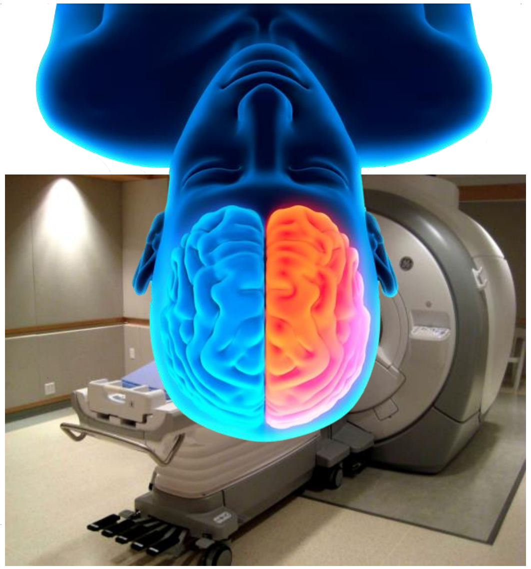
John is worried as Sherlock runs off.

Sherlock enters the door to the chemistry lab saying "John, I was here the whole time."

Once they get on the subway, John exclaims, "No you weren't!"

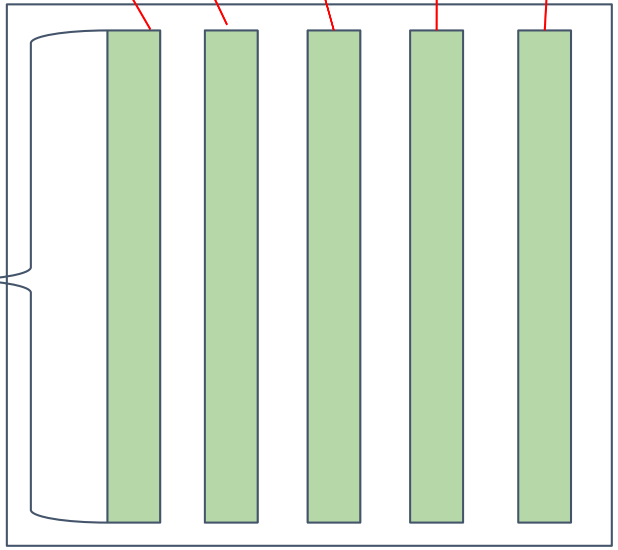
Moriarty arrives and says, "Hello Sherlock, John."

Each movie scene paired with text description from external party.



fMRI Machine

Voxel vector per time step



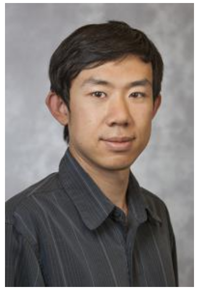
fMRI responses

(Funding: NSF, Simons Foundation, ONR)

Do GANs actually learn the distribution? Some theory and empirics.

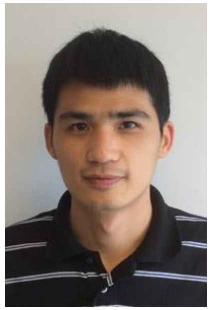
Sanjeev Arora
Princeton University

Rong Ge



Duke

Yingyu Liang



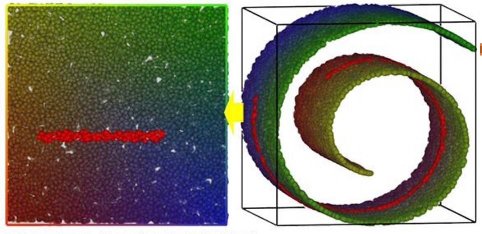
UW (Madison)

Tengyu Ma

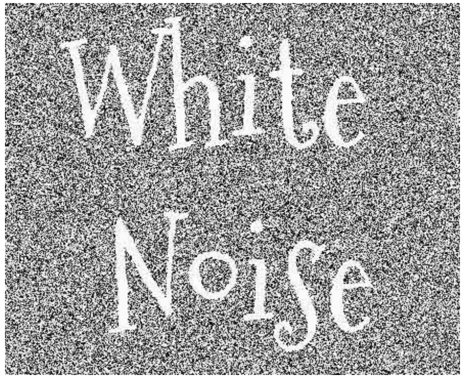
Soon: FAIR + Stanford

Yi Zhang

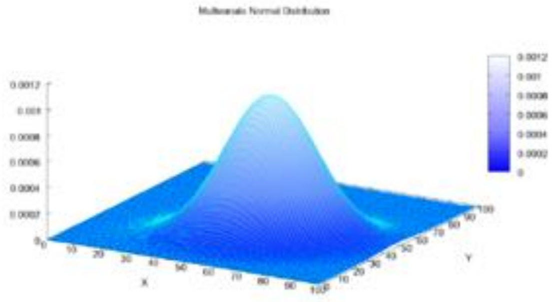
Deep generative models



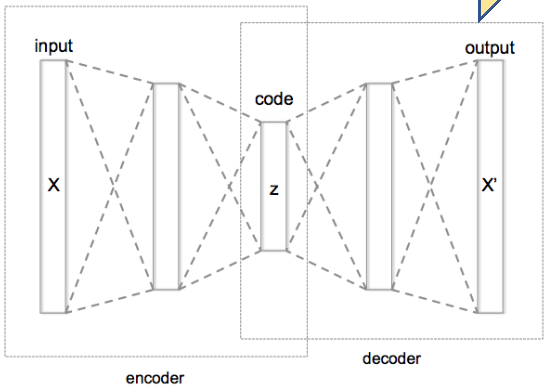
"Manifold"



N(O, I)



Generative Model

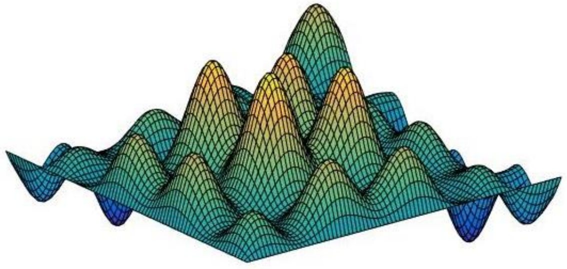


Denoising Auto encoders (Vincent et al'08)
Variational Autoencoder (Kingma-Welling'14)
GANs (Goodfellow et al'14)

Traditional training:

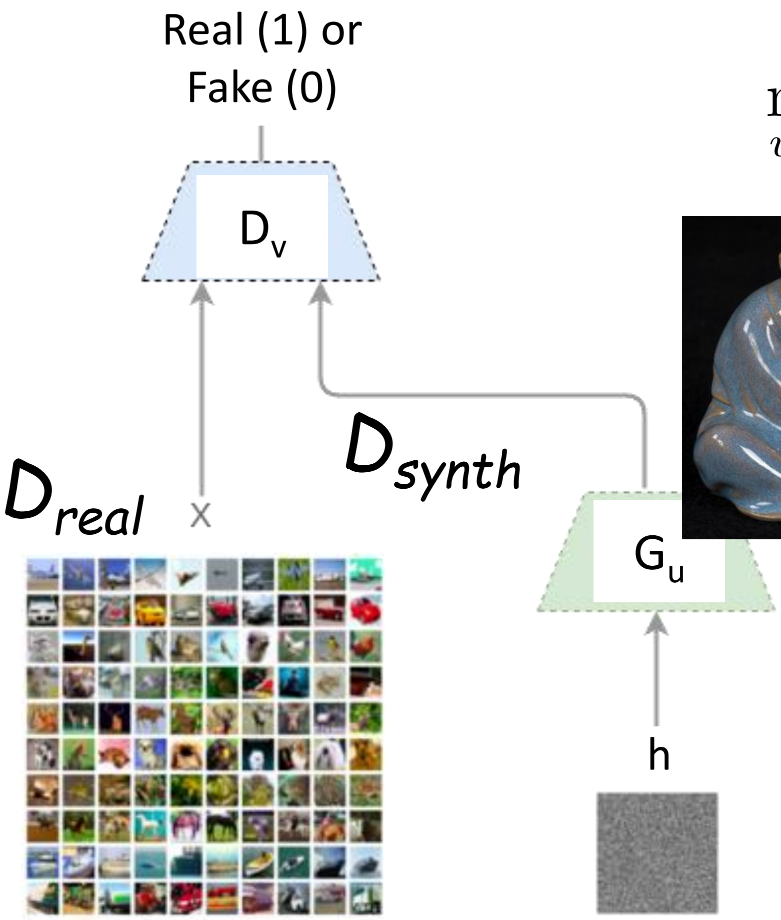
Max $E_x[\log p(x)]$

("perplexity" objective; Favors fuzziness in samples)



Data Distribution D_{real}

Generative Adversarial Nets (GANs) [Goodfellow et al. 2014]



$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}_{real}} [\log D_v(x)] + \mathop{\mathbb{E}}_{h \sim \mathcal{D}_h} [\log(1 - D_v(G_u(h)))].$$

Discriminator trained to output 1 on real inputs, and 0 on synthetic inputs.

Generator trained to produce synthetic inputs that make discriminator output 1.

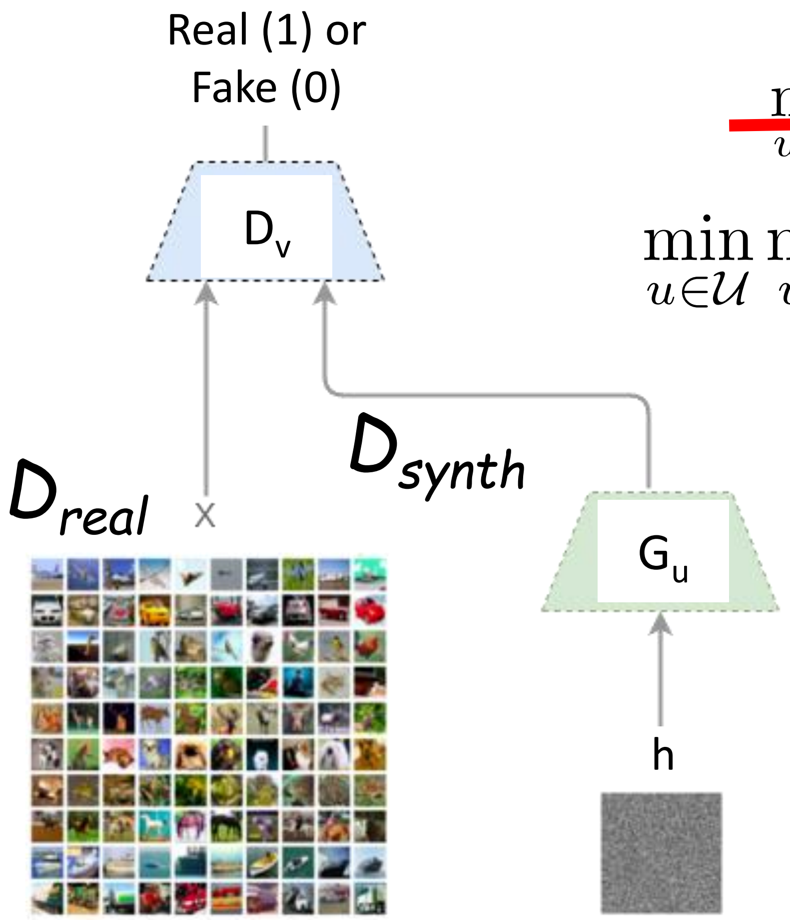
[Excellent resource: [Goodfellow's survey]

u= trainable parameters of Generator net

v = trainable parameters of Discriminator net

GANs (W-GAN variant)

“Difference in expected output on real vs synthetic images”
Wasserstein GAN [Arjovsky et al’17] **



$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}}[\log D_v(x)] + \mathbb{E}_{h \sim \mathcal{D}_h}[\log(1 - D_v(G_u(h)))].$$

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbf{E}_{x \sim \mathcal{D}_{real}}[D_v(x)] - \mathbf{E}_h[D_v(G_u(h))].$$

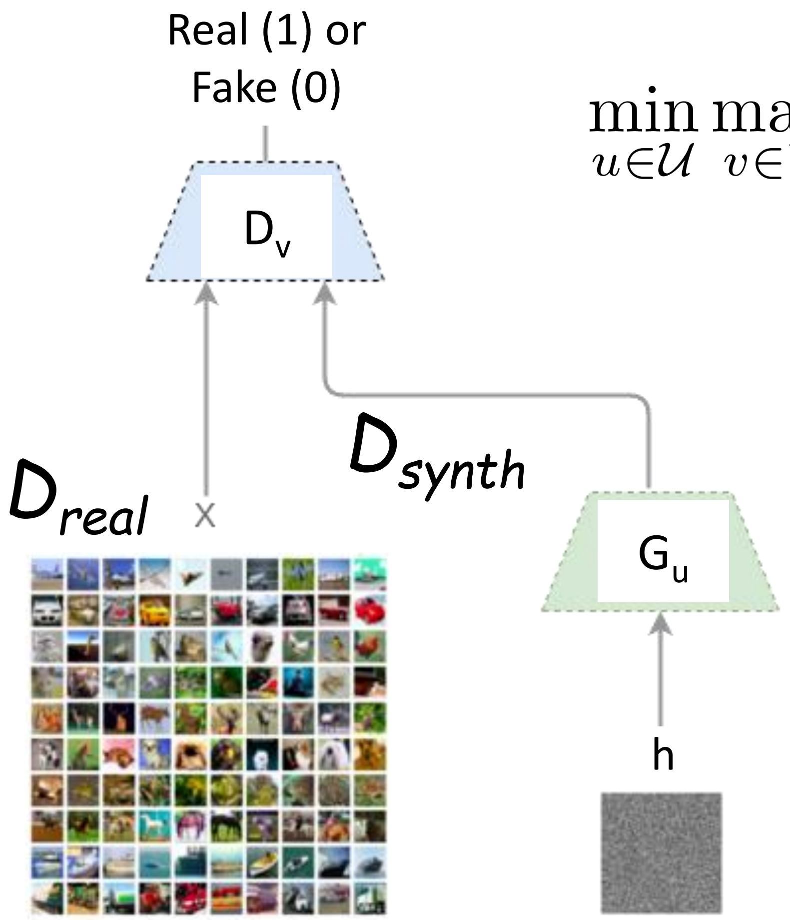
- Discriminator trained to output 1 on real inputs, and 0 on synthetic inputs.
- Generator trained to produce synthetic inputs that make discriminator output 1.

(** NB: Our results apply to most standard GANs objectives.)

u= trainable parameters of Generator net
v = trainable parameters of Discriminator net

GANs: "Winning"/Equilibrium

"Difference in expected output on real vs synthetic images"
Wasserstein GAN [Arjovsky et al'17]



$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \quad \mathbf{E}_{x \sim \mathcal{D}_{real}}[D_v(x)] - \mathbf{E}_h[D_v(G_u(h))].$$

- Discriminator trained to output 1 on real inputs, and 0 on synthetic inputs.
- Generator trained to produce synthetic inputs that make discriminator output 1.

Generator "wins" if objective ≈ 0 and further training of discriminator doesn't help. (An "Equilibrium.")

u= trainable parameters of Generator net

v = trainable parameters of Discriminator net

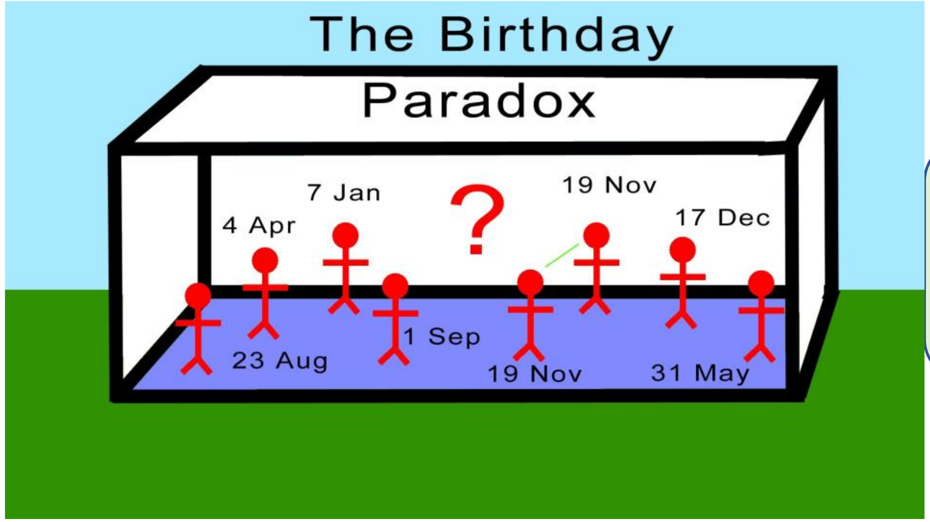
How to quantify success of training algorithm?

(Needed: ML theory for such implicit models.)

Main message here : Signs of trouble

Theorem 1 (A, Ge, Liang, Ma, Zhang ICML'17): There exist approx. equilibria where generator appears to win and yet the learnt distribution is very far from D_{real} . (In fact, distribution has v. low support.)

Supporting empirical finding (A., Risteski, Zhang '17): Learnt distributions by popular GANs trainings actually have low support (quantified using "birthday paradox" test.)



If you put 23 random people in a room, the chance is > 1/2 that two of them share a birthday.

Suppose a distribution is supported on N images.
Then Pr[sample of size $\sqrt{N}$ has a duplicate image] > ½.

Birthday paradox test* [A, Zhang; arxiv] : If a sample of size s has near-duplicate images with prob. > 1/2, then distribution has only s^2 distinct images.

Conclusions/further work

More exposition at www.offconvex.org

- Develop better (and "interpretable") type of of text embeddings than our simple linear embeddings. Capture grammar and semantics?
- Provably poly time algorithms?
- GANS with provably correct algorithms that learn the distribution?
- Provable convergence of neural net training algorithms?

THANK YOU!