Deep learning, Brains and the Evolution of Culture

+Bonus: how the brain could do credit assignment without backprop

Yoshua Bengio

23 July 2014

COGSCI 2014, Quebec City

Workshop on Deep Learning and the Brain

Main Theory

- Optimization difficulty for deeper nets, more abstract concepts
- Humans manage to bypass this difficulty thanks to culture, guidance from other humans
- The evolution of memes & culture is an effective way to explore the space of brain configurations, by divide-and-conquer

Hypothesis 1

- *When the brain of a single biological agent learns, it performs an approximate optimization with respect to some endogenous objective.*

Almost all modern ML training procedures are justified by approximately optimizing some training objective.

Hypothesis 2

- *When the brain of a single biological agent learns, it relies on approximate local descent in order to gradually improve itself.*

Most ML training procedures proceed by gradual changes, and humans appear on the surface to generally change only a little at a time.

Theoretical and experimental results on deep learning suggest:

Hypothesis 3

- *Higher-level abstractions in brains are represented by deeper computations (going through more areas or more computational steps in sequence over the same areas).*

Deep net = composition of functions.

Examples:

- Gulcehre & Bengio ICLR'2013, learning a composition of functions
- Current SOTA in object recognition on ImageNet requires deeper nets (8 layers) than SOTA on MNIST (digit recognition, 3 layers)
- Parts-based visual hierarchies are deeper for more abstract and complex objects.

Effective Local Minima

- It is not clear that **actual** local minima are a real issue in training deep nets
  - But initial conditions can sometimes matter a lot!
  - see evidence suggesting instead that saddle points create plateaus that act as obstacles:

    Pascanu et al, On the saddle point problem for non-convex optimization, arXiv 2014

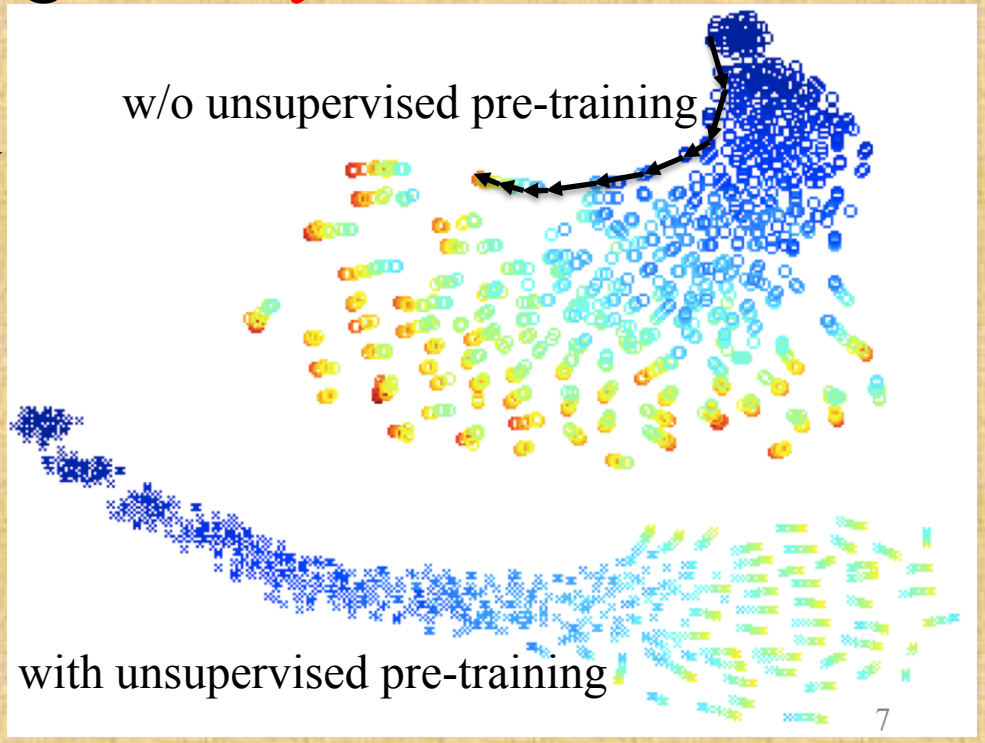
- An optimizer like the one in brains may get stuck → **effective local minima**

Effect of Initial Conditions in Deep Nets

- *(Erhan et al 2009, JMLR)*
- Supervised deep net with vs w/o unsupervised pre-training → very different minima

Neural net trajectories in function space, visualized by t-SNE

No two training trajectories end up in the same place → huge number of effective local minima



Hypothesis 4

- *Learning of a single human learner is limited by **effective** local minima.*

In spite of seeing contradictory evidence, humans sometimes stick to wrong beliefs…

Hypothesis 5

- *A single human learner is unlikely to discover high-level abstractions by chance because these are represented by a deep sub-network in the brain.*

- ML methods do not fare as well when trying to learn more abstract higher-level concepts.
- Deeper neural networks are more difficult to train (often faring worse than sufficiently deep but shallower ones)

Experimental Evidence from Deep Learning Research

- In *(Gulcehre & Bengio ICLR'2013)* we set up a task that seems almost impossible to learn by shallow nets, deep nets, SVMs, trees, boosting etc

Algorithm	20k dataset		40k dataset		80k dataset	
	Training Error	Test Error	Training Error	Test Error	Training Error	Test Error
SVM RBF	26.2	50.2	28.2	50.2	30.2	49.6
KNN	24.7	50.0	25.3	49.5	25.6	49.0
Decision Tree	5.8	48.6	6.3	49.4	6.9	49.9
Randomized Trees	3.2	49.8	3.4	50.5	3.5	49.1
MLP	26.5	49.3	33.2	49.9	27.2	50.1
Convnet/Lenet5	50.6	49.8	49.4	49.8	50.2	49.8
Maxout Convnet	14.5	49.5	0.0	50.1	0.0	44.6
2 layer sDA	49.4	50.3	50.2	50.3	49.7	50.3
Struct. Supervised MLP w/o hints	0.0	48.6	0.0	36.0	0.0	12.4
Struct. MLP+CAE Supervised Finetuning	50.5	49.7	49.8	49.7	50.3	49.7
Struct. MLP+CAE+DAE Supervised Finetuning	49.1	49.7	49.4	49.7	50.1	49.7
Struct. MLP+DAE+DAE Supervised Finetuning	49.5	50.3	49.7	49.8	50.3	49.7

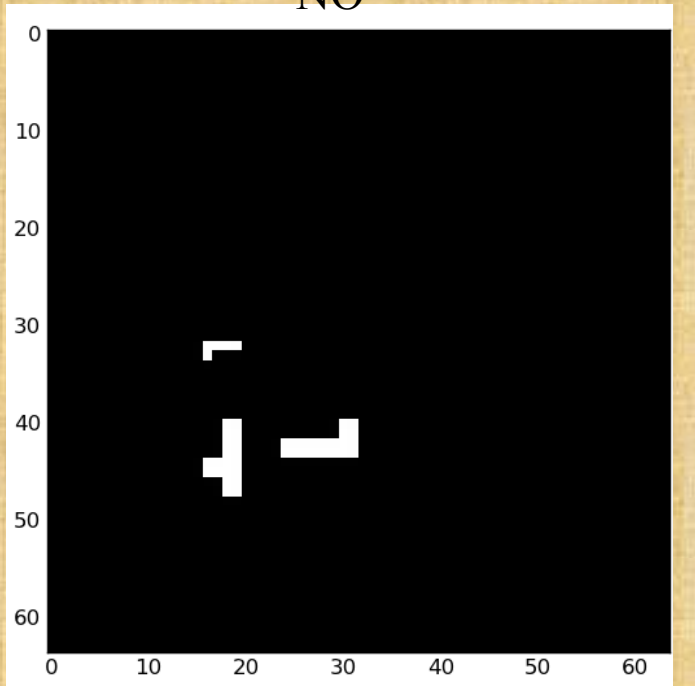
CHANCE PREDICTIONS

The composed task: Pentominoes

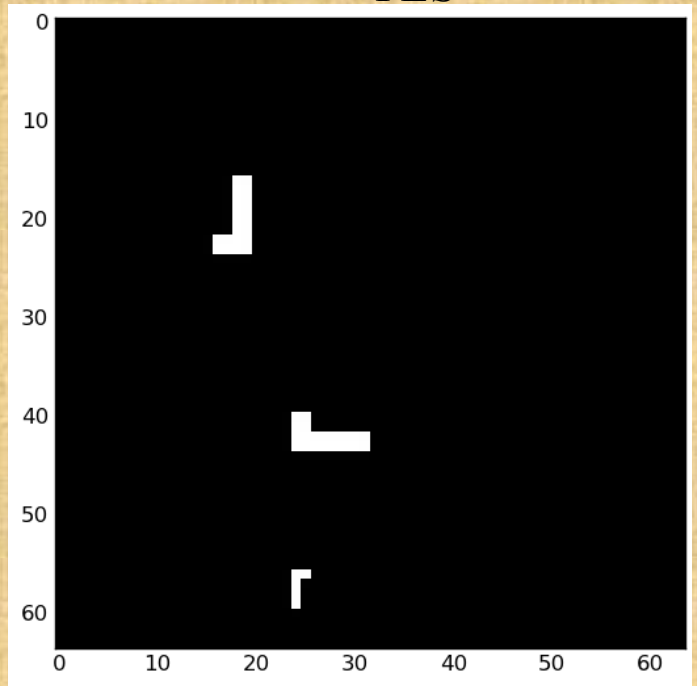
- Input = 64x64 binary pixels with 3 shapes (rotated, scaled, translated) from 10 categories

- Target = are the 3 shapes of the same category?

NO



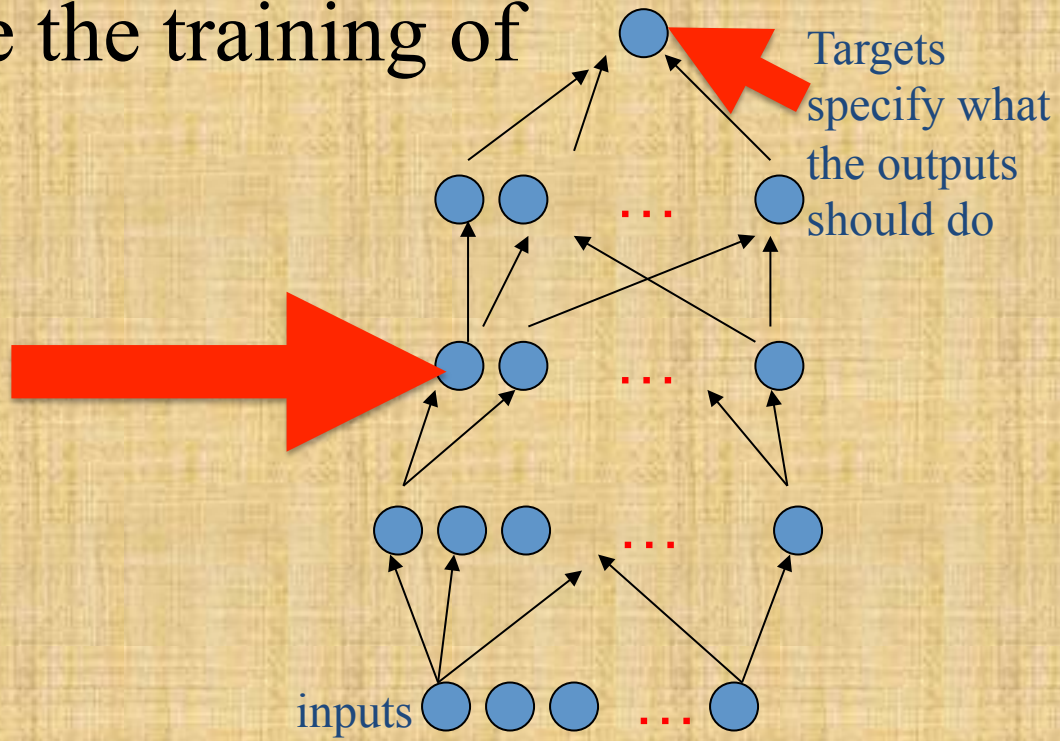
YES



So... how do humans manage to learn high-level abstractions?

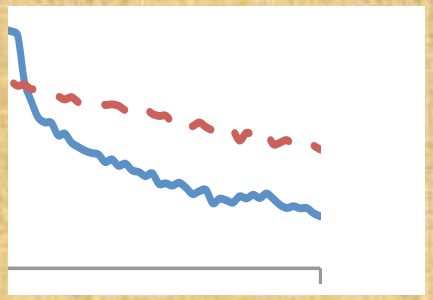
Hints about intermediate concepts

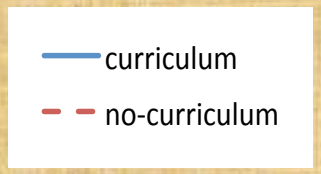
- Training a deep net from end-to-end is a difficult optimization problem
- But it gets much easier if some training signal can be used to guide the training of intermediate layers



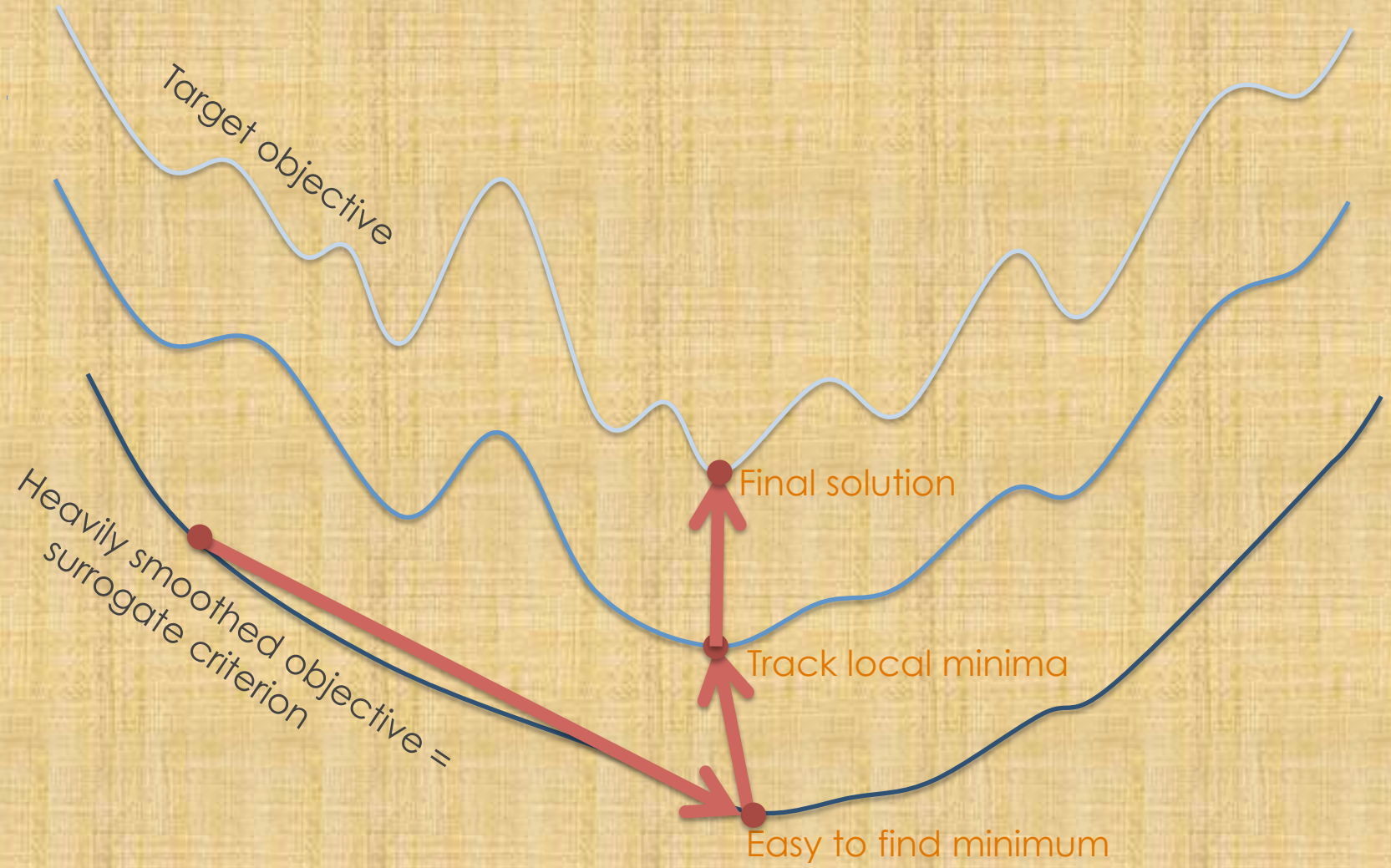
Curriculum Learning

- Start with easier examples and build new concepts on top of previously acquired ones
- *(Bengio et al, ICML 2009)*

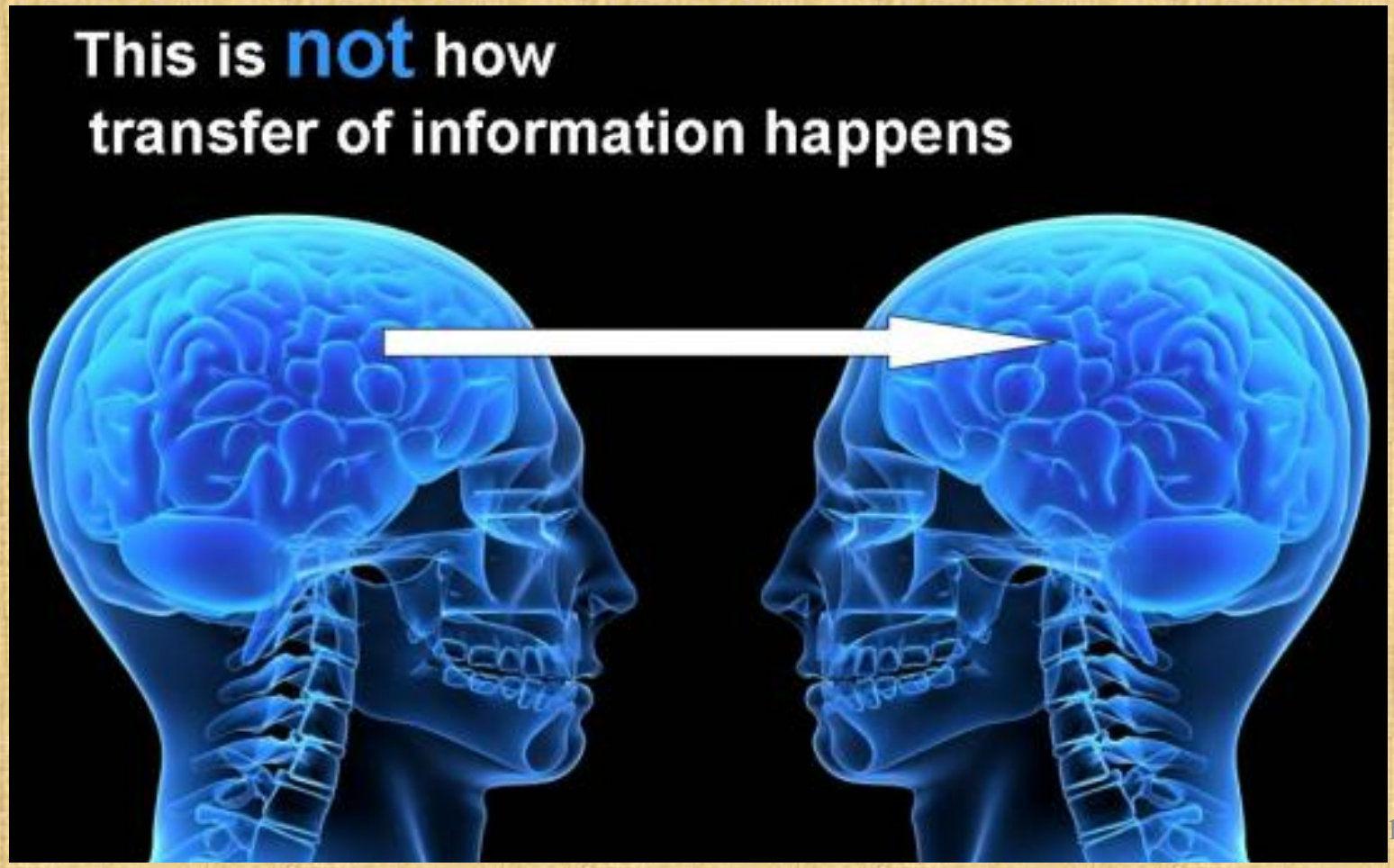




Curriculum Learning as a Continuation Method to Defeat Effective Local Minima

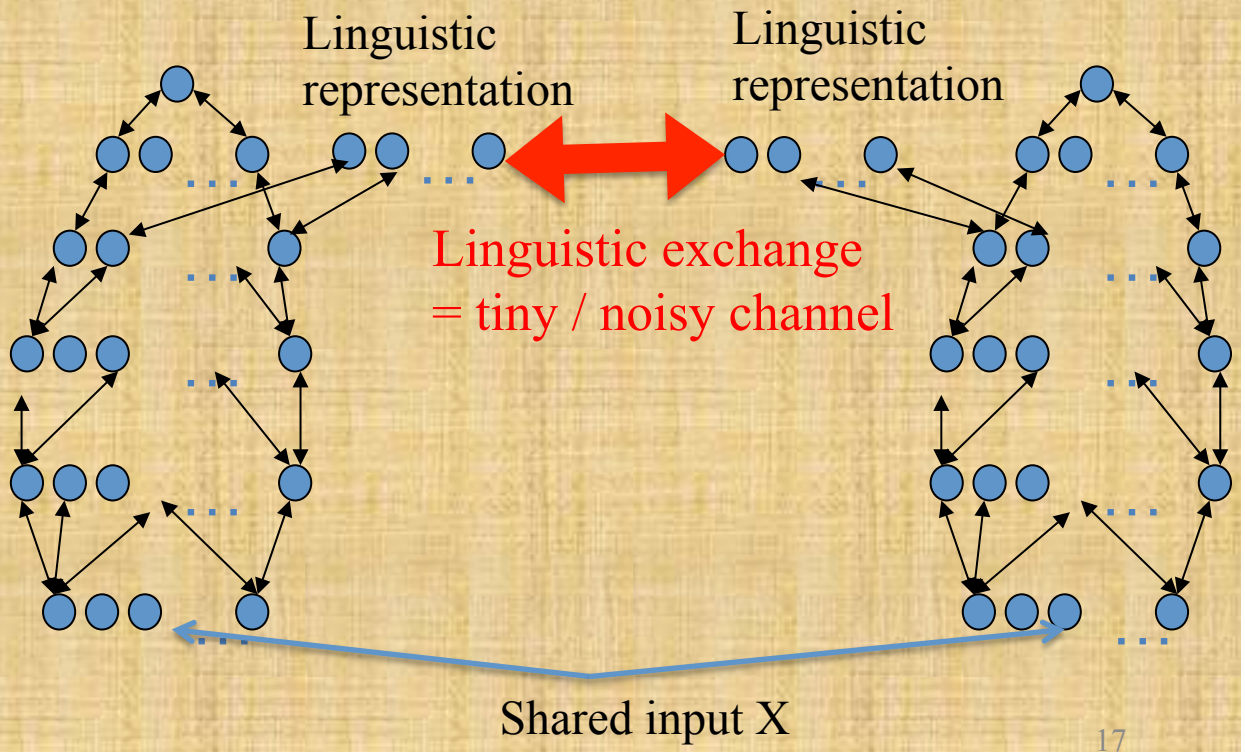


Guided learning: How is one brain transferring abstractions to another brain? From synapses to synapses? No!



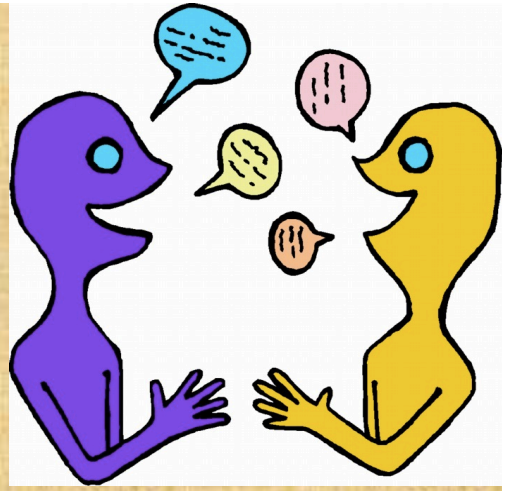
How is one brain transferring abstractions to another brain?

Two individuals sharing a similar visual input, the teacher gives hints to the student about high-level abstractions



The linguistic output of one individual is modeled by the other one, jointly with X.

What it says about language

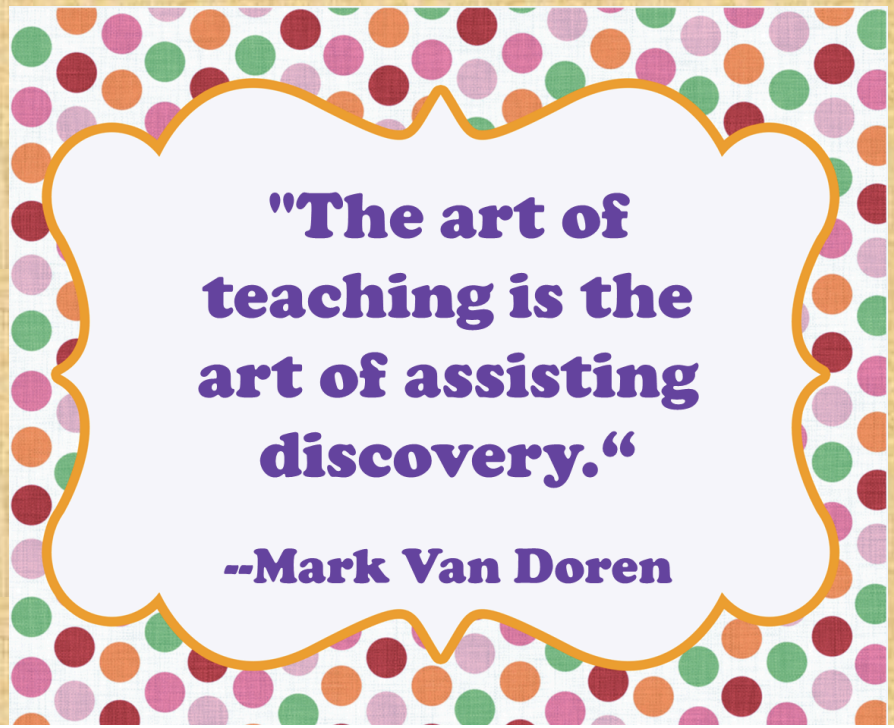


- Each individual has a different ‘language’, a 2-way map between internal representations and linguistic symbols
- We learn language by trying to predict other humans’ language output (in some context)
- Individual languages tend to converge to collective conventions shared by many individuals for expressing thoughts (but never perfectly, there is still a lot of miscommunication)
- Different languages = different attractors

Hypothesis 6

Curriculum learning

- *A human brain can learn high-level abstractions if guided by the signals produced by other humans, which act as hints or indirect supervision for these high-level abstractions.*



How do we escape effective local minima?

- linguistic inputs → virtual examples (stories told by other humans), summarize knowledge
  - teacher/student roles can change
  - credibility of teacher (and how well its theories match data) matter in how much weight the student gives it
- criterion landscape becomes easier to optimize
  e.g. via curriculum learning
- turn difficult unsupervised learning into easy supervised learning of **intermediate abstractions**

Guided Training, Intermediate Concepts

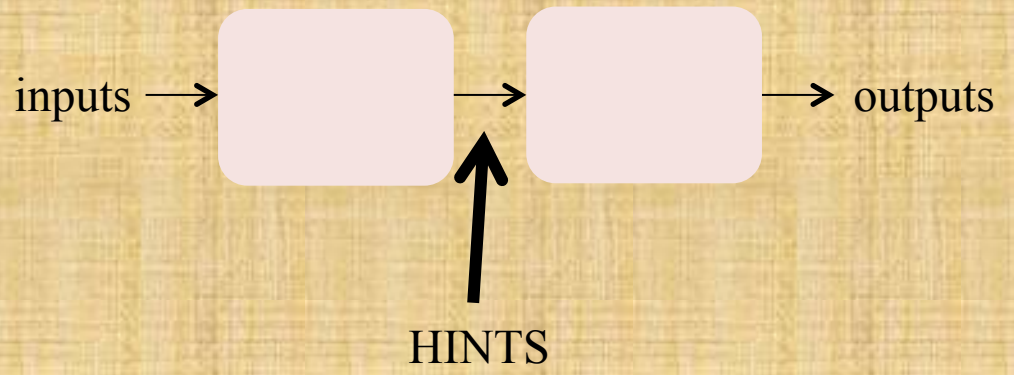
- In *(Gulcehre & Bengio ICLR'2013)* we set up a task that seems almost impossible to learn by shallow nets, deep nets, SVMs, trees, boosting, etc.
- Yet, sucessful learning is possible…

Algorithm	20k dataset		40k dataset		80k dataset	
	Training Error	Test Error	Training Error	Test Error	Training Error	Test Error
SVM RBF	26.2	50.2	28.2	50.2	30.2	49.6
KNN	24.7	50.0	25.3	49.5	25.6	49.0
Decision Tree	5.8	48.6	6.3	49.4	6.9	49.9
Randomized Trees	3.2	49.8	3.4	50.5	3.5	49.1
MLP	26.5	49.3	33.2	49.9	27.2	50.1
Convnet/Lenet5	50.6	49.8	49.4	49.8	50.2	49.8
Maxout Convnet	14.5	49.5	0.0	50.1	0.0	44.6
2 layer sDA	49.4	50.3	50.2	50.3	49.7	50.3
Struct. Supervised MLP w/o hints	0.0	48.6	0.0	36.0	0.0	12.4
Struct. MLP+CAE Supervised Finetuning	50.5	49.7	49.8	49.7	50.3	49.7
Struct. MLP+CAE+DAE Supervised Finetuning	49.1	49.7	49.4	49.7	50.1	49.7
Struct. MLP+DAE+DAE Supervised Finetuning	49.5	50.3	49.7	49.8	50.3	49.7
Struct. MLP with Hints	**0.21**	**30.7**	**0**	**3.1**	**0**	**0.01**

PERFECT PREDICTIONS

Guided Training, Intermediate Concepts

- Breaking the problem in two sub-problems and pre-training each module separately, then fine-tuning, nails it
- *Need prior knowledge to decompose the task*
- Guided pre-training allows to find much better solutions, escape effective local minima



Where did the knowledge used to guide a learner come from in the first place?

How could language/education/culture possibly help humanity find the better synaptic configurations associated with more useful abstractions?

MEME

Anything that can be copied from one mind to another.

More than random search:
Potentially an exponential speed-up by divide-and-conquer

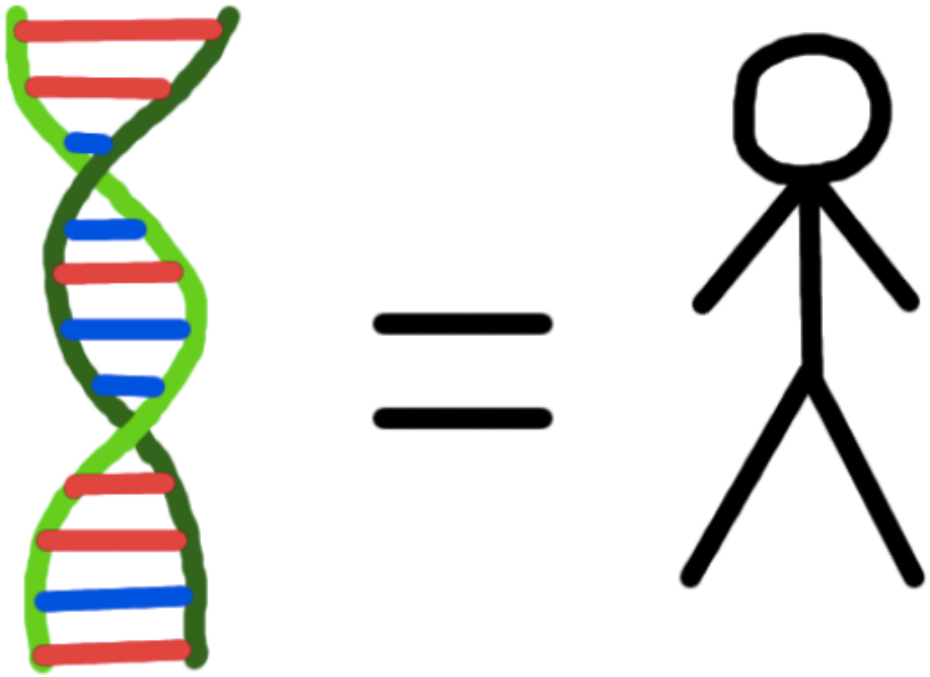
Combinatorial advantage: can combine solutions to independently solved sub-problems

Hypothesis 7

- *Language and meme recombination provide an efficient evolutionary operator, allowing rapid search in the space of **memes**, that helps humans build up better high-level internal representations of their world.*

From Genes to Memes: a Revolution in Search Efficiency

2 principles combined:
- Noisy copy of meme:
  = teaching by example
- Recombination of sub-solutions
  = creativity



Ideas as efficient memes

Selective Pressure on Memes

- Better ideas dominate by being shown to be useful
- Diffusion of information (about ideas, and about their value), crucial for efficiency of this process
- Premium given to novelty and diversity: to promote and evaluate potentially good novel ideas, avoid losing them
- **Credibility can be assigned to an idea and not just to its author, making the selective pressure more efficient than in the genetic case.**

From where do **new** ideas emerge?

3 time scales:

- Seconds: **inference** (novel explanations for current x)
- Minutes, hours: **learning** (local descent, like current DL)
- Years, centuries: **cultural evolution** (global optimization, recombination of ideas from other humans)

Consequences of the Theory

More efficient cultural evolution with

- **Better exploration of new ideas**
  - Scientific research
  - Spreading the investment across many high-risk explorations
  - Encouraging diversity
- **Better rate of spread of good ideas**
  - Open & free access to information & open research
  - Education for the whole planet
  - Open Internet where everyone can publish
  - Multiple non-centralized rating systems

Conclusions

- Deep learning research suggests that cultural evolution helps to collectively deal with a difficult optimization problem that single humans could not solve
- Social and political implications for organizing our societies towards maximum efficiency of growth of cultural wealth: brains that better understand the world around us
- Implications for AI research:
  - Collections of learning agents building on each other's discoveries to build up towards higher-level abstractions
  - Guiding computers just like we guide children

Reference papers

- *Yoshua Bengio, Evolving culture vs local minima, ArXiv 1203.2990, chapter in 'Growing Adaptive Machines'. 2013.*

- *Caglar Gulcehre and Yoshua Bengio, Knowledge matters: importance of prior information for optimization. ICLR'2013.*

+Bonus

How the brain could do credit assignment without back-prop

An immature but very exciting theory!

Yoshua Bengio

July 23rd, 2014

Preliminaries

- Regularized auto-encoders implicitly learn a distribution *P(x)* that estimates the data generating distribution *(ICLR'2013, NIPS'2013, ICML'2014)*, from which one can sample by MCMC (encode/decode/add noise)

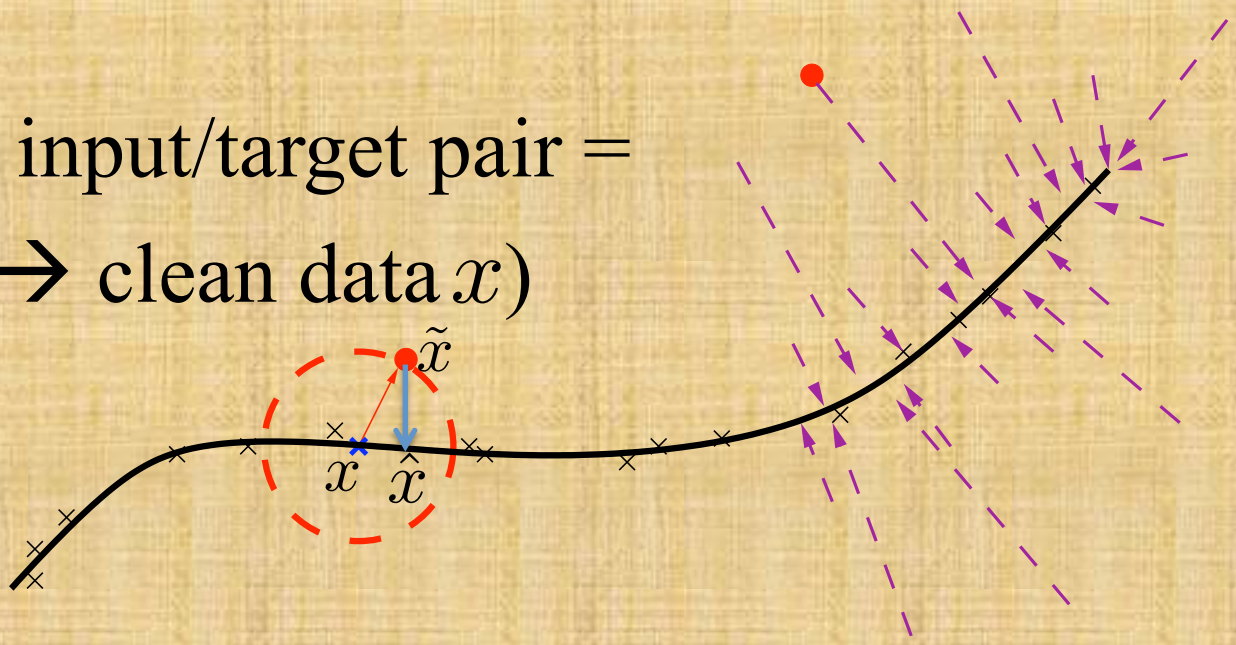
Denoising Auto-Encoders Learn a Small Move Towards Higher Probability

- Reconstruction $\hat{x}$ points in direction of higher probability

$$\hat{x} - x \propto \frac{\partial \log P(x)}{\partial x}$$

gradient

- Trained with input/target pair = (corrupted $\tilde{x}$ → clean data x)



Reconstruction = how to change some activations so as to be more consistent with the others

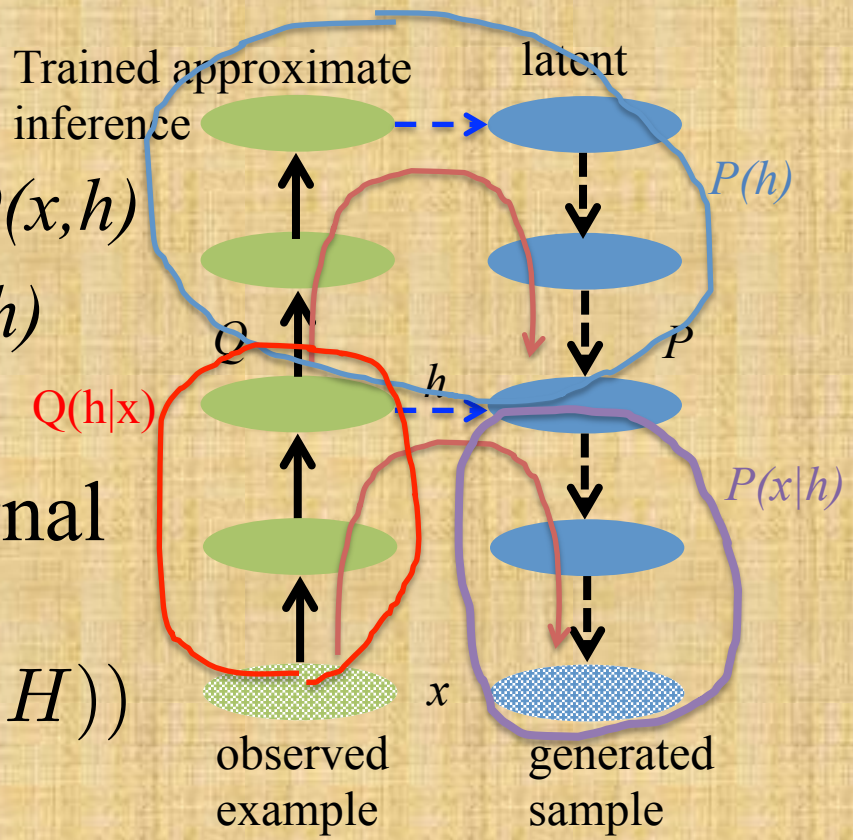
- Consider two 'parts' x_1 and x_2, and the reconstruction on x_1, given (x_1,x_2):

$$\frac{\partial \log P(x_1, x_2)}{\partial x_1} = \frac{\partial \log P(x_1|x_2)}{\partial x_1}$$

- Thus **reconstruction tells a unit how it should change to agree more with the others**

Training Objective

- Two distributions:
  - data $Q(x) \rightarrow h \sim Q(h|x)$: $Q(x,h)$
  - model $P(h) \times P(x|h)$: $P(x,h)$



- Objective to provide a signal at any layer h

$$\min KL(Q(X,H)||P(X|H))$$

can be decomposed into:

- reconstruction error of x through $h \sim Q$
- log-likelihood of $h \sim Q$ according to $P(h)$
- entropy of $Q(h|x)$

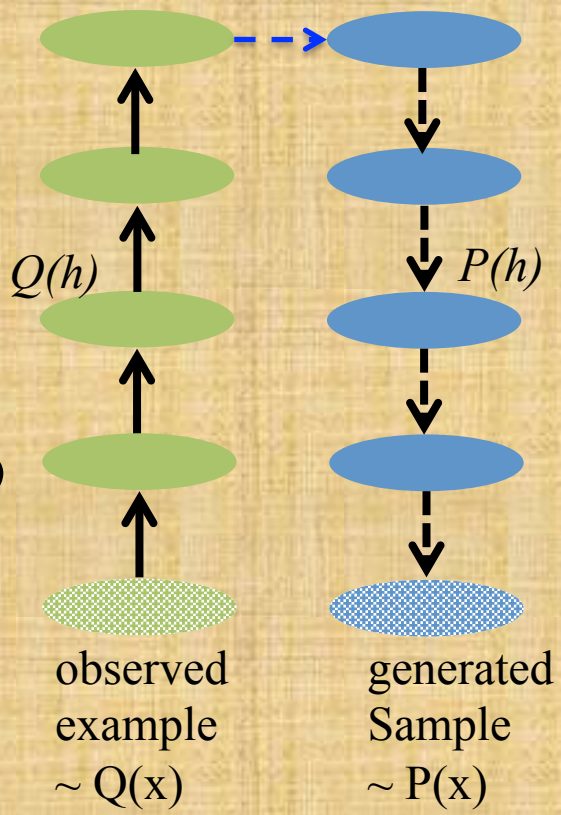
Why Q(h|x) and P(x|h) should be information preserving

- Most other deep generative models have the property that $Q(h|x)$ and/or $P(x|h)$ are « noisy »
- Injecting "noise" at low levels when generating downward creates high-frequency iid noise in generated images our sounds, unlike real data

➔ Noise must be added only at the high levels

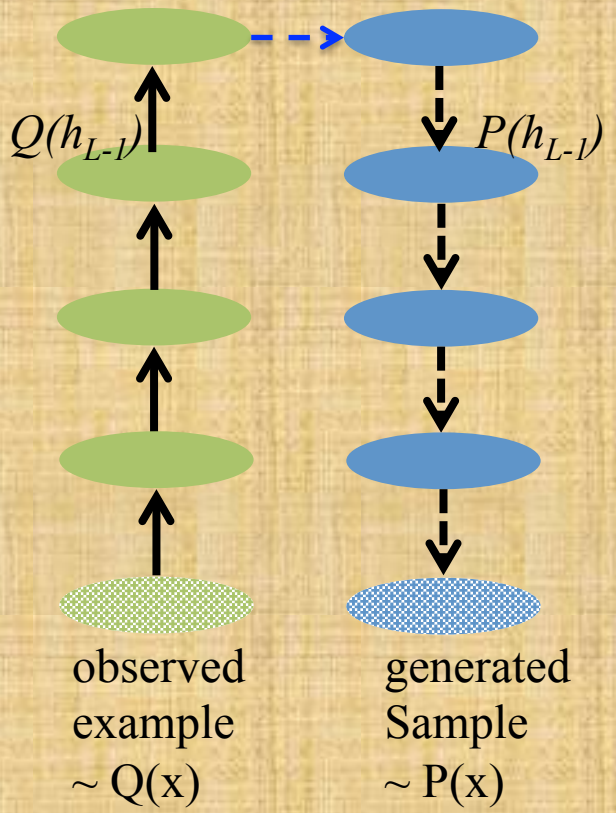
Beyond Learning an Invertible Mapping

- There is an infinite number of invertible mappings that would minimize reconstruction error
- We want one that maps a complicated distribution *Q(x)* into a simpler one *Q(h)* that can be modeled by *P(h)*



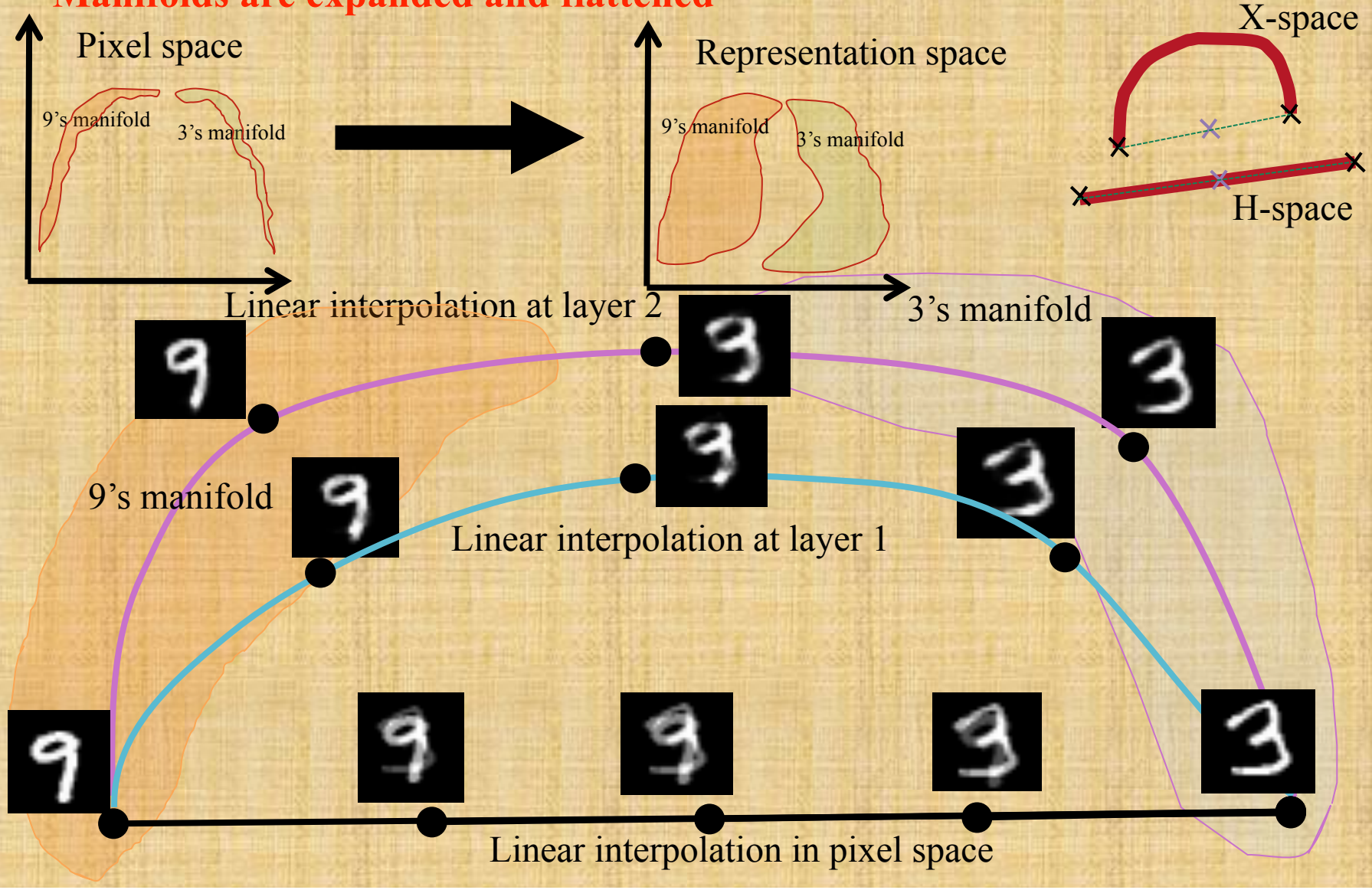
Gradual Transformation of a Twisted Distribution into a Flat One

- What the successive layers do (going up from x) is to transform their input distribution into one that is less twisted, and more disentangled (where the features are more nearly independent, with a flat or factorial joint marginal)
- The top auto-encoder wants $Q(h_{L-1})$ to match what a shallow auto-encoder can capture in its implicity $P(h_{L-1})$: Gaussian (in the linear case) or more generally, factorizable.



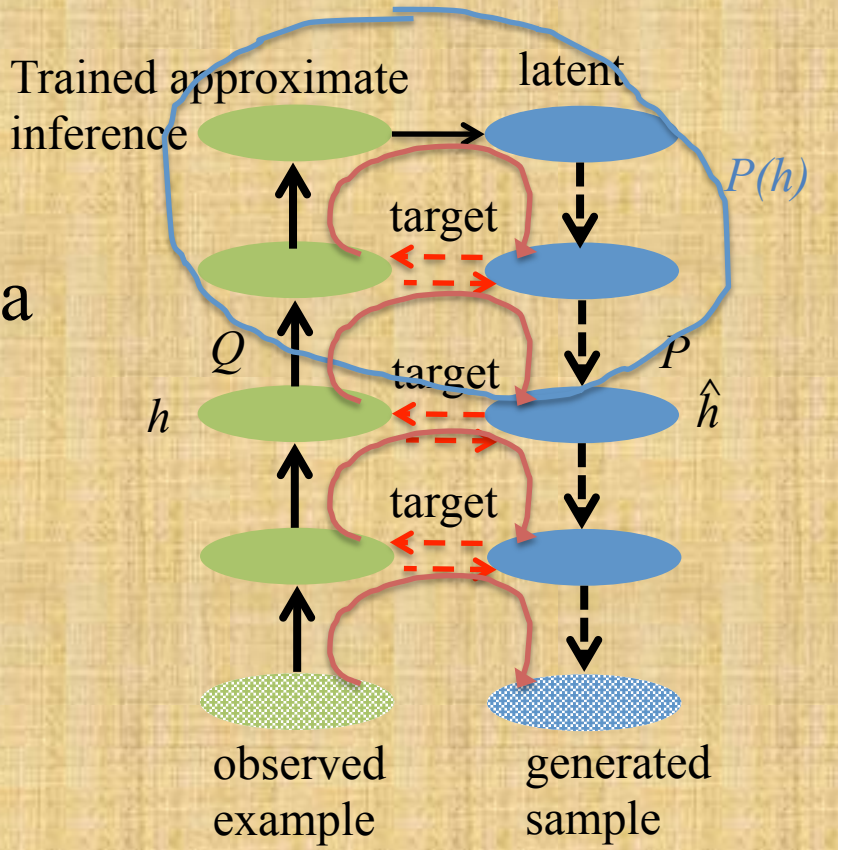
Space-Filling in Representation-Space

- **Deeper representations → abstractions → disentangling**
- **Manifolds are expanded and flattened**



Purely Local Training Signals

- Each layer tries to be a good denoising auto-encoder while transforming the lower-level data into a form *h* easier to model by higher levels: higher *P(h)*
- This basically makes the long-path reconstructions (going all the way up) a target $\hat{h}$ for the original *h*, and vice-versa, while the long-path auto-encoder is trained with *h* as data



How to avoid back-prop altogether

- If each layer of a deep auto-encoder has small reconstruction error and is contractive, so is the deep auto-encoder, i.e., it is a good denoising auto-encoder.

define

$$\tilde{f}_l(x) = f_l(f_{l-1}(\dots f_2(f_1(x)))) \tag{10}$$

and

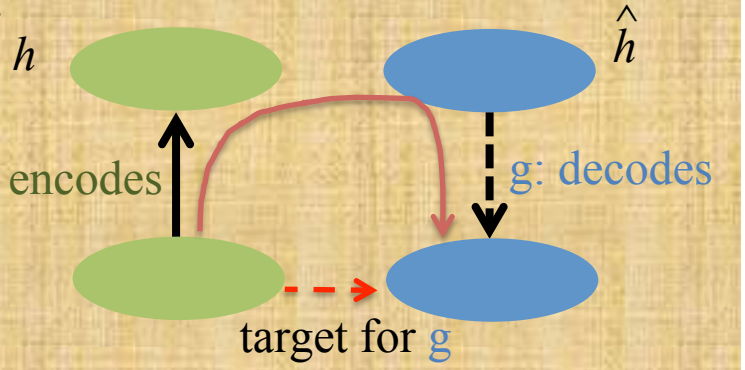
$$\tilde{g}_l(h_l) = g_1(g_2(\dots g_{l-1}(g_l(h_l)))). \tag{11}$$

Then we have that

$$g_i(f_i(h_{i-1})) = h_{i-1}, \ \ \forall h_{i-1} \sim Q(H_{i-1}), \ \forall i \quad \Rightarrow \quad \tilde{g}_i(\tilde{f}_i(x)) = x, \ \ \forall x \sim Q(X), \ \forall i$$

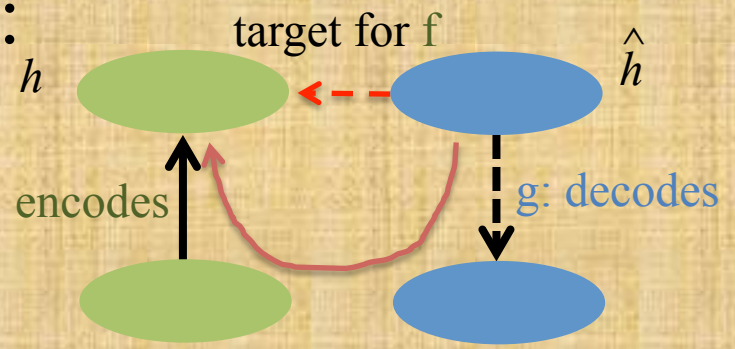
How to avoid back-prop altogether

- Long paths provide top-down signal for encoders to produce easy-to-model distributions
- How do we make layer-wise encoders and decoders good inverses of each other on data?
- Auto-encoding BOTH ways + future-matches-past (reconstruction) principle:
  - encode/decode/update decoder
  - decode/encode/update encoder



How to avoid back-prop altogether

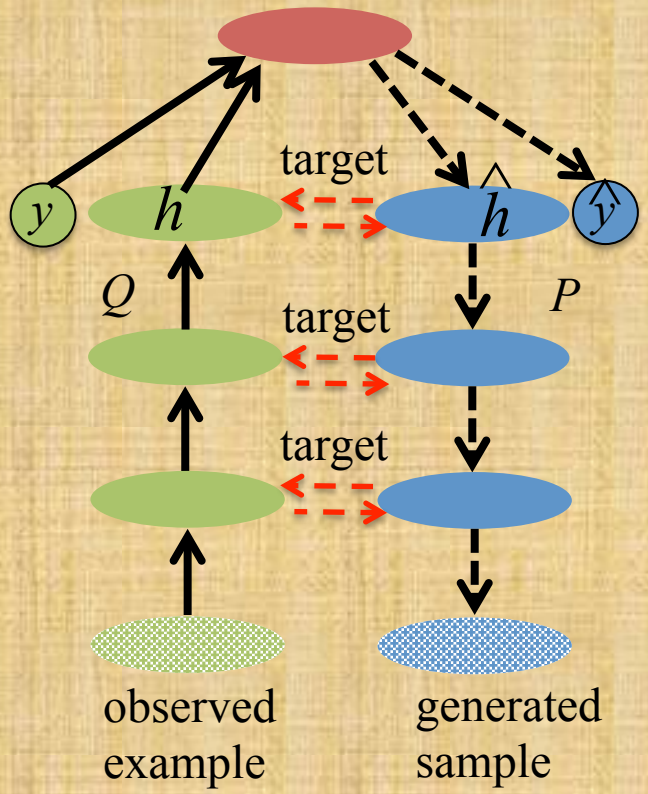
- Long paths provide top-down signal for encoders to produce easy-to-model distributions
- How do we make layer-wise encoders and decoders good inverses of each other on data?
- Auto-encoding BOTH ways + future-matches-past (reconstruction) principle:
  - encode/decode/update decoder
  - decode/encode/update encoder



➔ *Similar to the Recirculation algorithm (Hinton & McClelland 1988)*

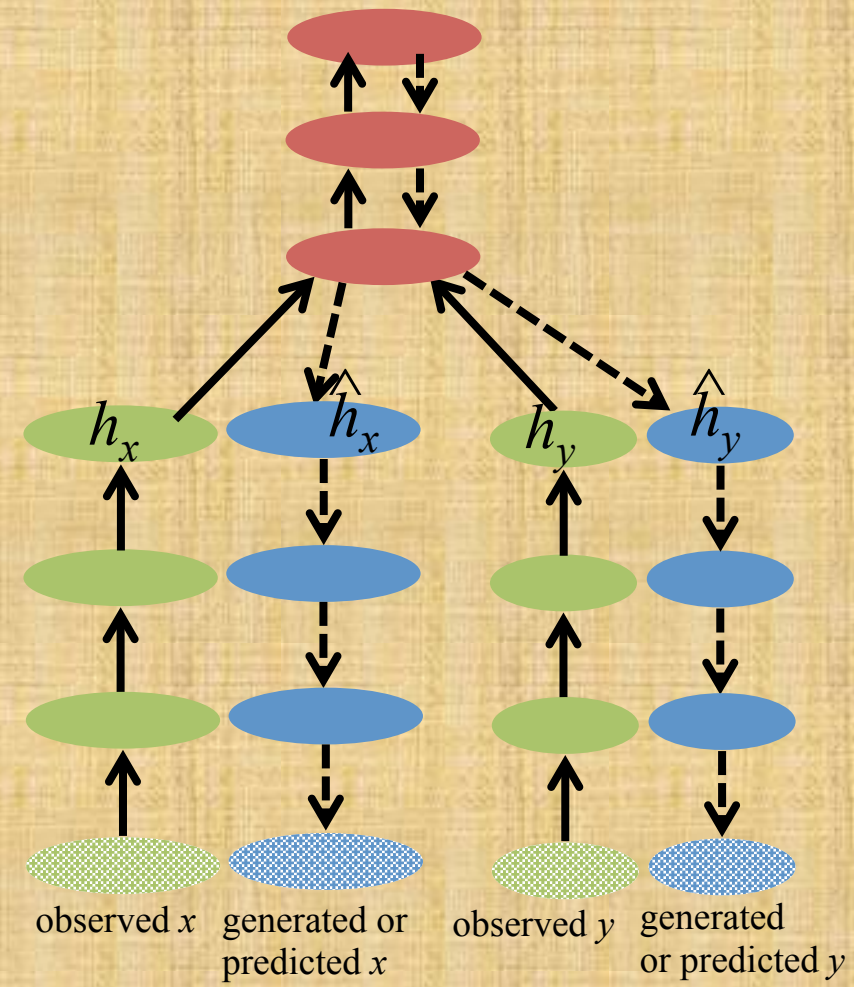
Supervised Learning by Target Propagation

- Simple classification: same principle except that the top-level auto-encoder models the joint of h and y
- $\hat{h} - h$ is now indicating $\frac{\partial \log P(h,y)}{\partial h}$
- A discriminant version compares the reconstruction with and without y clamped, and computes $\frac{\partial \log P(y|h(x))}{\partial h(x)}$



Multi-Modal / Structured Output

- y is complex and needs its own $P(y)$ (modeled by it's own stack of auto-encoders) and non-trivial $P(y|x)$. Model joint of $h_x(x)$ and $h_y(y)$ with another stack on top.
- Inference (MAP or MCMC) is done with the top stack, then projected back in the x or y space.

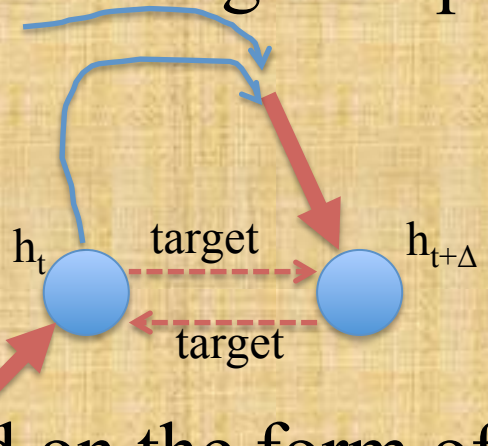


How the Brain Might Learn

- Two principles:
  - The past tries to match the future: prediction
  - The future tries to match the past: reconstruction

  Not clear if these should be on same or different units.
- Plus: observations being clamped (not always)

Different loops =
Different lengths =
Different Δ



No need to store past activations: just average pre-synaptic contributions with a temporal kernel

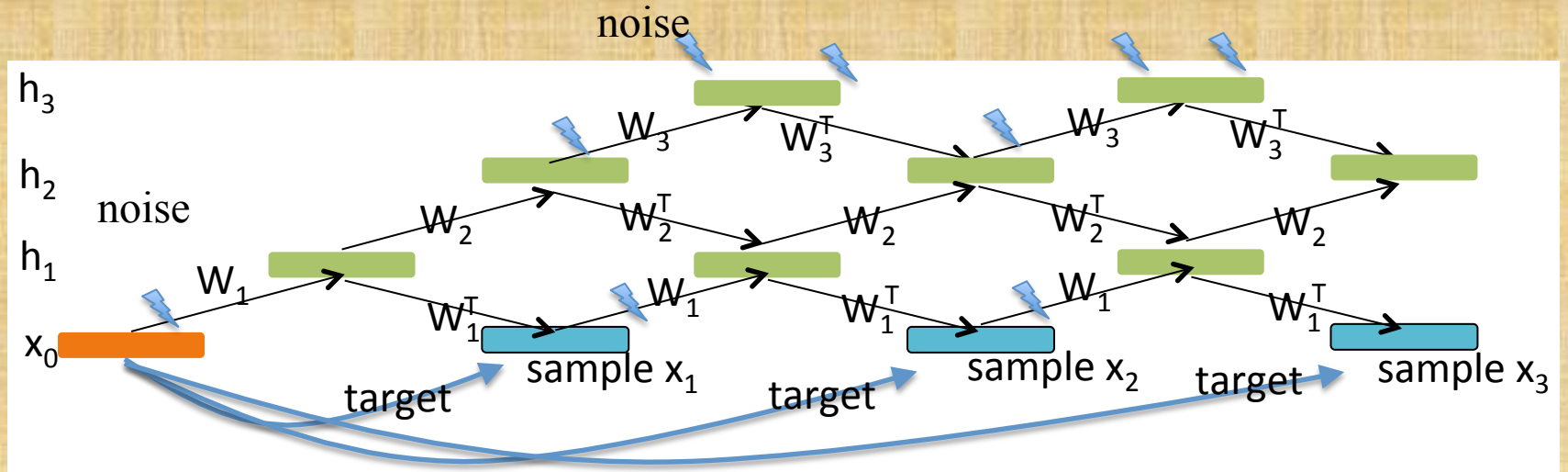
- Does not depend on the form of activation function, tied symmetric weights, differentiability of anything, using rates vs spikes, etc.

Automatic Buildup of Higher-Level Representations

- Imagine an initial empty slate with small random weights: intermediate layers are happy outputting 0, but the neurons h_1 projecting into the clamped sensory x get a signal that trains them to predict x, given whatever they get.
- When the senses are unclamped, this makes the x-to-h_1 fibers learn to invert the h_1-to-x fibers.
- h_1 becomes a boundary condition for h_2, etc.
- Layers above h_1 now model it and provide a signal for the x-to-h_1 fibers to learn a mapping that is easy to reconstruct by upper layers, etc.

The Maths of Denoising Auto-Encoders Extend to Stochastic Recurrent Networks

- Bengio et al, ICML 2014, on Deep **Generative Stochastic Networks (GSNs)**
- Same criterion but now the reconstruction can be obtained through an arbitrary noise recurrent network
- Running the net = MCMC sampling from the model. If clamped: conditional sampling.

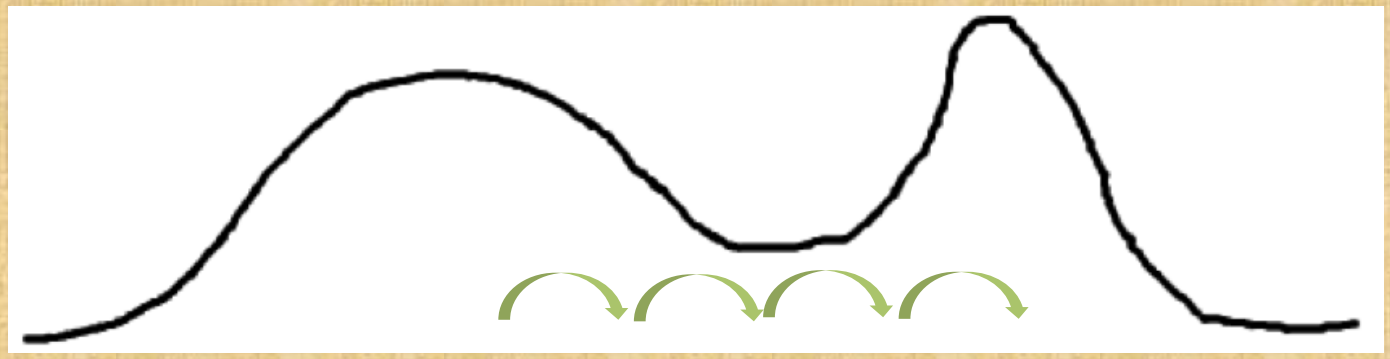


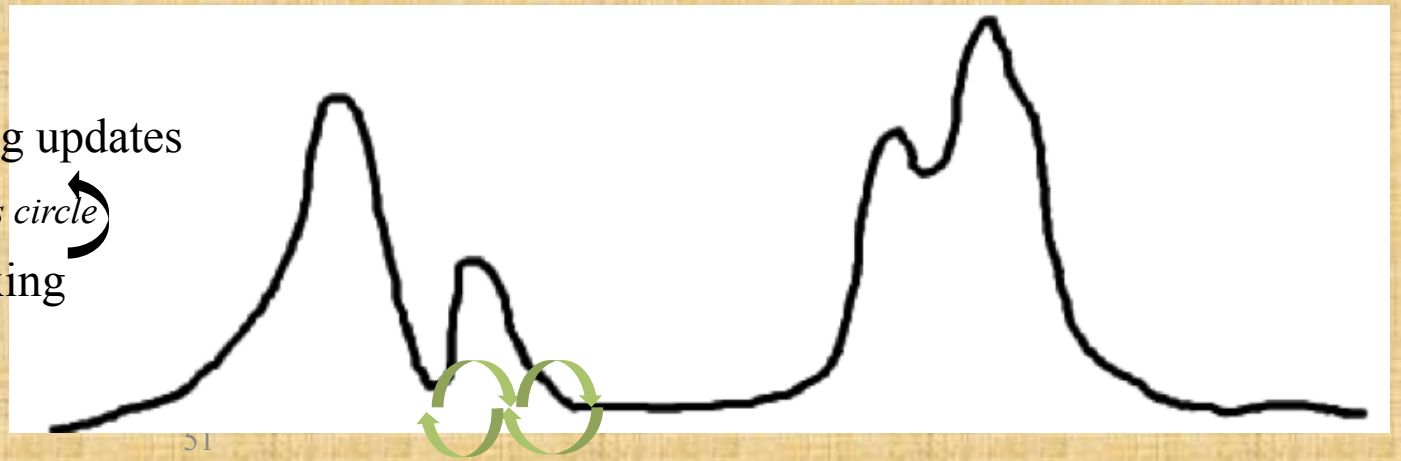
(Conditional) Sampling & MAP

- Two things we want from our models:
  - Probabilistic inference:
    - Sample some variables given others (or none)
  - MAP inference:
    - Choose likely values for some variables given others
- Both can be done here:
  - Unconditional sampling by ancestral sampling from P
  - Conditional sampling by GSN-like MCMC, clamping the given variables and resampling others:
    - Iteratively encode/decode with noise injected (top level stack)
  - Local ascent for approximate MAP:
    - Iteratively encode/decode with no noise injected (top level stack)

Issues with Boltzmann Machines

- Sampling from the MCMC of the model is required in the inner loop of training
- As the model gets sharper, mixing between well-separated modes stalls





Issues with Back-Prop

- Over very deep nets or recurrent nets with many steps, non-linearities compose and yield sharp non-linearity → gradients vanish or explode
- Training deeper nets: harder optimization
- In the extreme of non-linearity: discrete functions, can't use back-prop