Optimization Challenges for Deep Learning

Yoshua Bengio

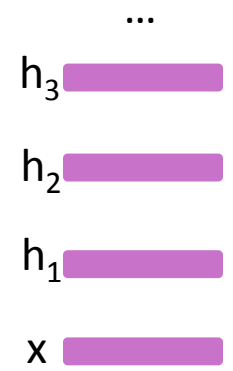
U. Montreal

December 12th, 2014

OPT'2014: NIPS Workshop on Optimization for Machine Learning

Deep Representation Learning

Learn multiple levels of representation of increasing complexity/abstraction



- theory: exponential gain
- brains are deep
- cognition is compositional
- Better mixing (Bengio et al, ICML 2013)

- **They work! SOTA on industrial-scale AI tasks (object recognition, speech recognition, language modeling, music modeling)**

Deep Learning Challenges (Bengio, arxiv 1305.0445 Deep learning of representations: Looking forward)

- Computational Scaling
- Optimization & Underfitting
- Intractable Marginalization, Approximate Inference & Sampling
- Disentangling Factors of Variation
- *Reasoning & One-Shot Learning of Facts*

Deep Learning Challenges
(Bengio, arxiv 1305.0445 Deep learning of representations: looking forward)

- Computational Scaling
- Optimization & Underfitting
- Intractable Marginalization, Approximate Inference & Sampling
- Disentangling Factors of Variation
- Reasoning & One-Shot Learning of Facts

Challenge: Computational Scaling

- Recent breakthroughs in speech, object recognition and NLP hinged on faster computing, GPUs, and large datasets

- In speech, vision and NLP applications we tend to find that

*as **Ilya Sutskever** would say*

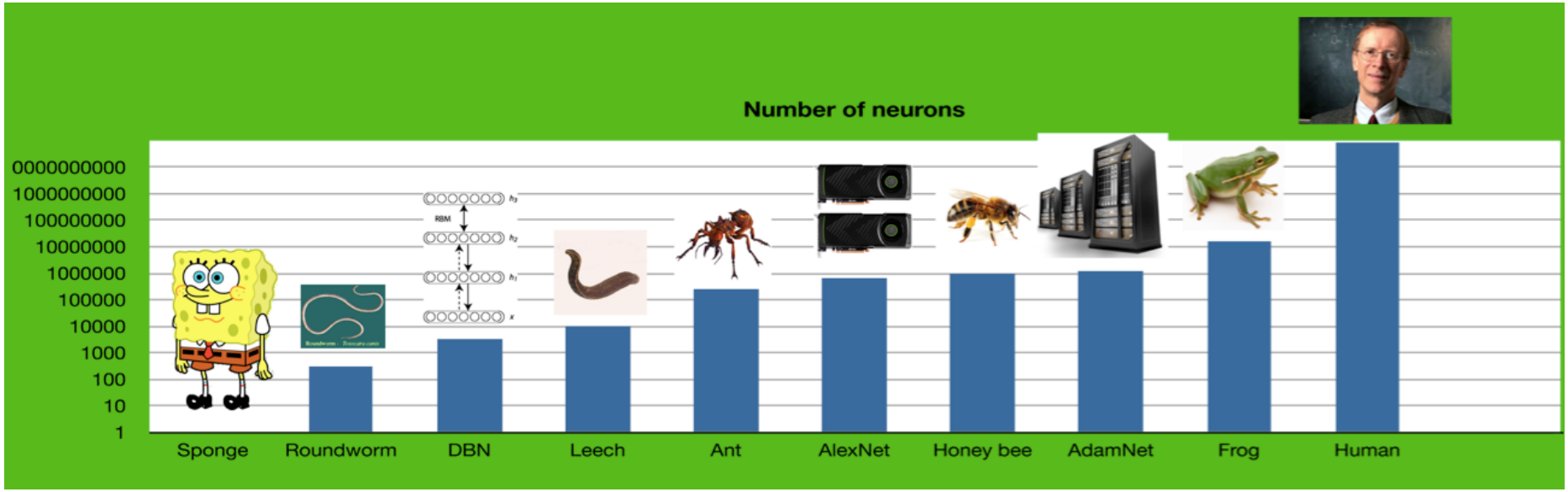
BIGGER IS BETTER

Because deep learning is

EASY TO REGULARIZE while

it is MORE DIFFICULT TO AVOID UNDERFITTING

We still have a long way to go in raw computational power



Computation / Capacity Ratio

- N-grams, decision trees, etc.: poor generalization but capacity (and memory) can grow a lot while computation remains constant or grows as log(capacity).

- Neural nets / deep learning: very good generalization, but computation grows linearly with capacity (number of parameters). Each parameter is used for every example.

- To build much higher-capacity models, we need to break that linear relationship while keeping the compositional structure that makes deep learning generalize so well.

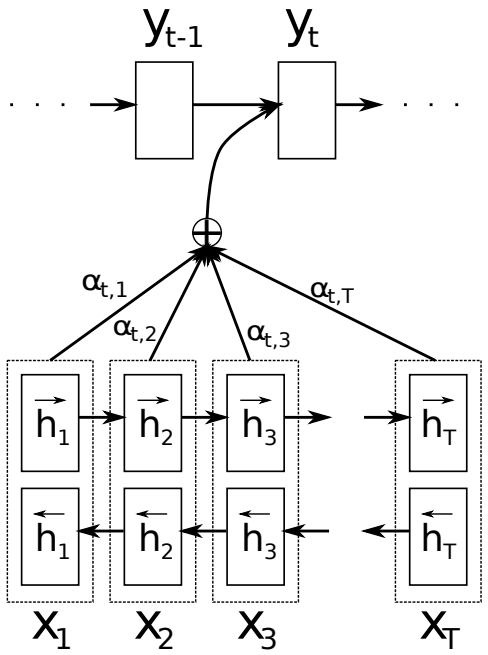
Machine Translation Examples

- n-gram based English-French MT: ~ 26 Gbytes (zipped), 80 G unzipped?
  - Moses phrase-based baseline: 33.3 BLEU
  - Edinburgh: 37 BLEU (using very large LM dataset)

- SOTA deep-learning based English-French MT:
  - Montreal:
    - Single model, 285M (unzipped): published 28.5 BLEU, latest 33.2 BLEU
  - Google:
    - Single large model, 1.7G: 32.7 BLEU
    - Ensemble of 8 models, 13.5G: 36.9 BLEU

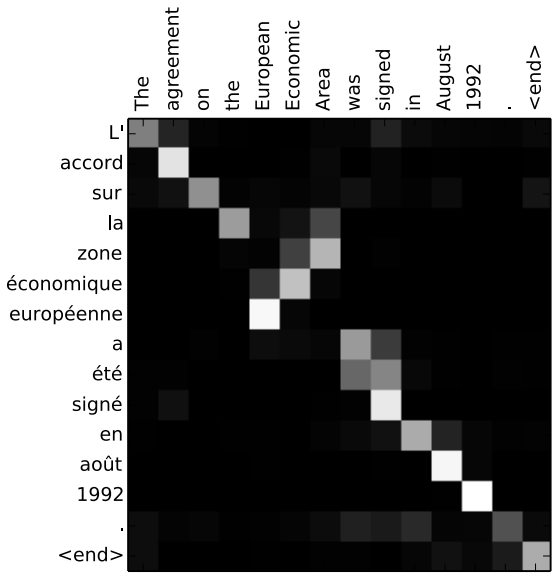
New Results on Deep Machine Translation

- Handles long sentences by introducing an attention mechanism
- Learns to choose which part of the input sentence to pay most attention to when predicting the next output word, as a function of the output RNN state and input bi-RNN state

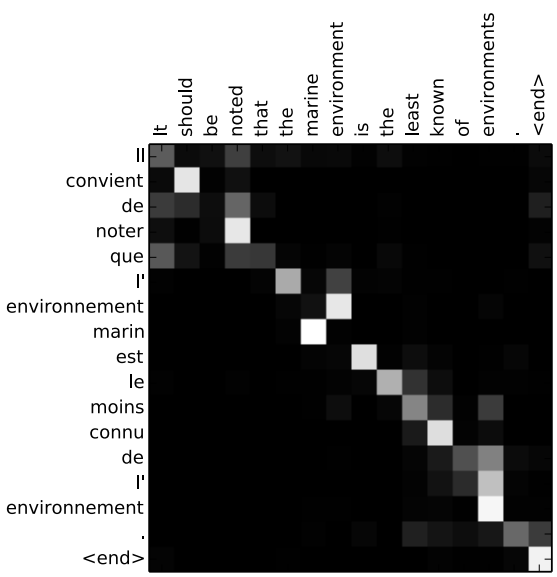


- Single GPU trained over 2 weeks

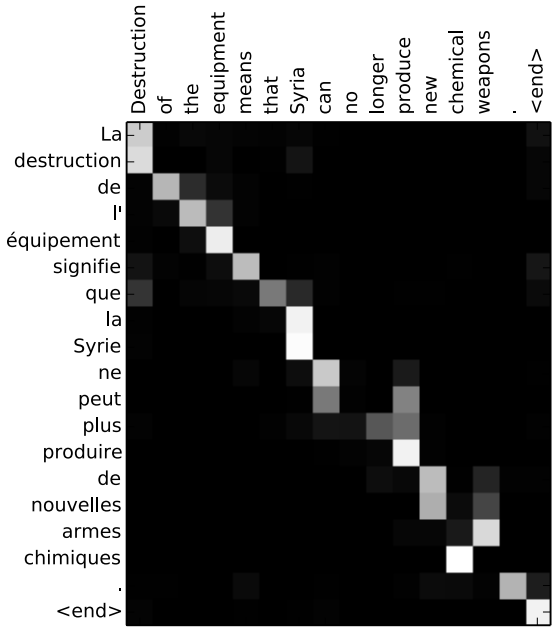
Predicted Alignments



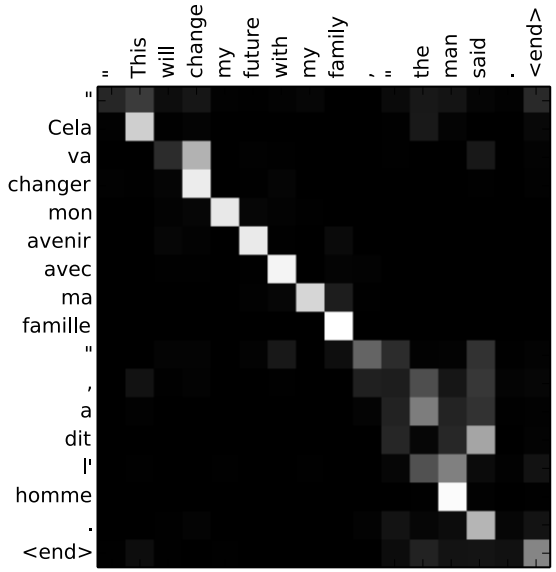
(a)



(b)

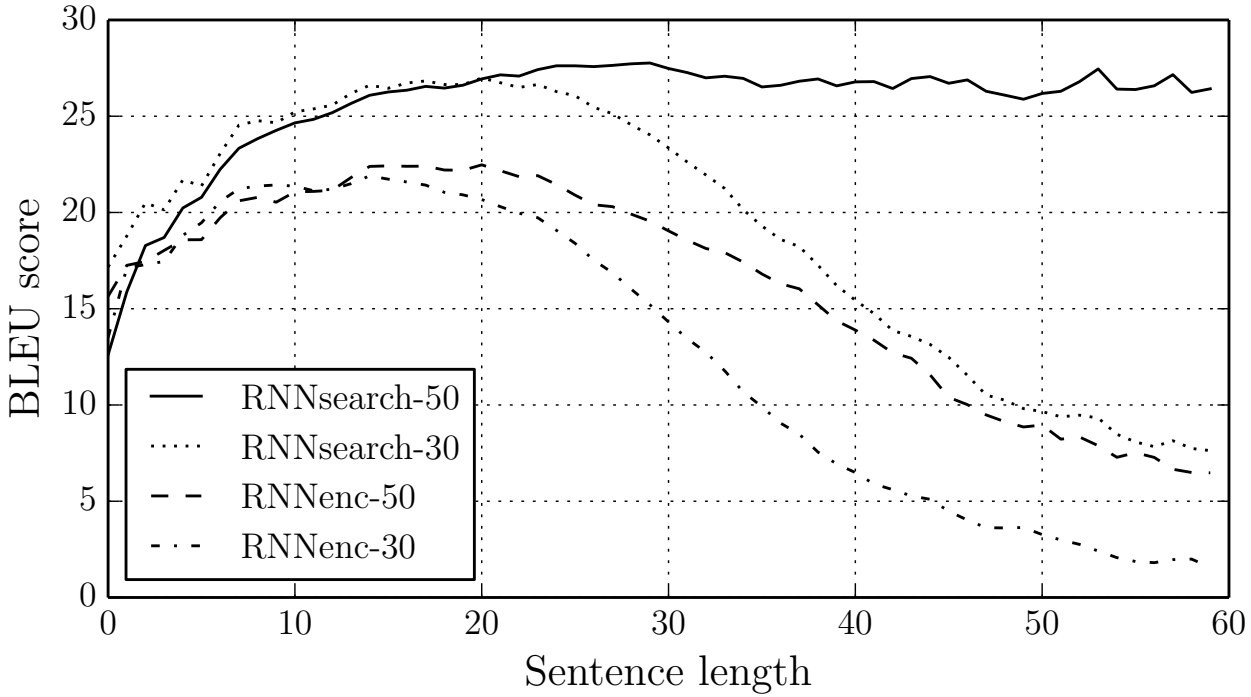


(c)



(d)

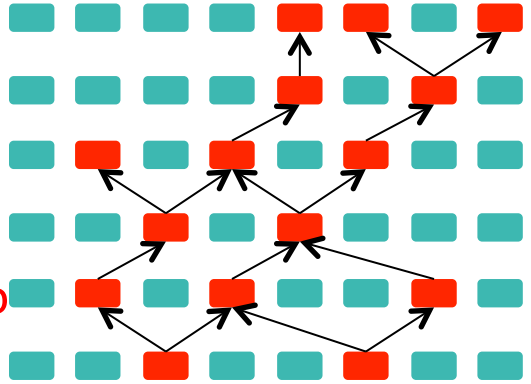
Improvements over Pure AE Model



- RNNenc: encode whole sentence
- RNNsearch: predict alignment
- BLEU score on full test set (including UNK)
- We now reached SOTA on En-Fr (37 BLEU) and En-Ge (21 BLEU)

Conditional Computation: only visit a small fraction of parameters / example

Bengio, Leonard & Courville
arXiv 1305.2982



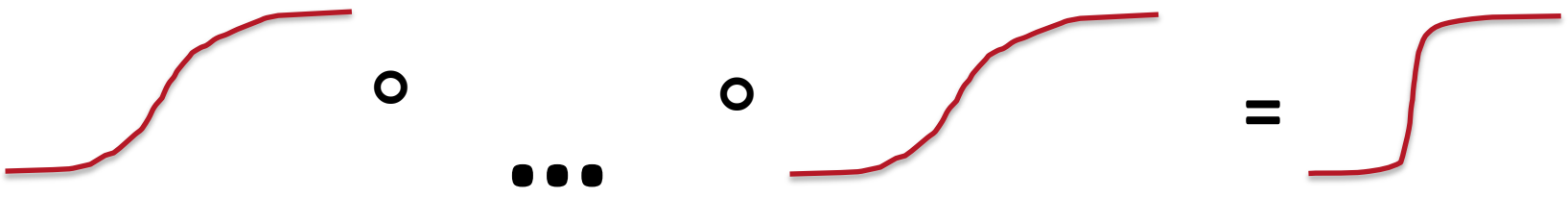
- Deep nets vs decision trees
- Hard mixtures of experts (Collobert, Bengio & Bengio 2002)
- Conditional computation for deep nets: sparse distributed gaters selecting combinatorial subsets of a deep net
- Challenges:
  - Credit assignment for hard decisions
  - Gated architectures exploration

Deep Learning Challenges
(Bengio, arxiv 1305.0445 Deep learning of representations: Looking forward)

- Computational Scaling
- Optimization & Underfitting
- Intractable Marginalization, Approximate Inference & Sampling
- Disentangling Factors of Variation
- Reasoning & One-Shot Learning of Facts

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



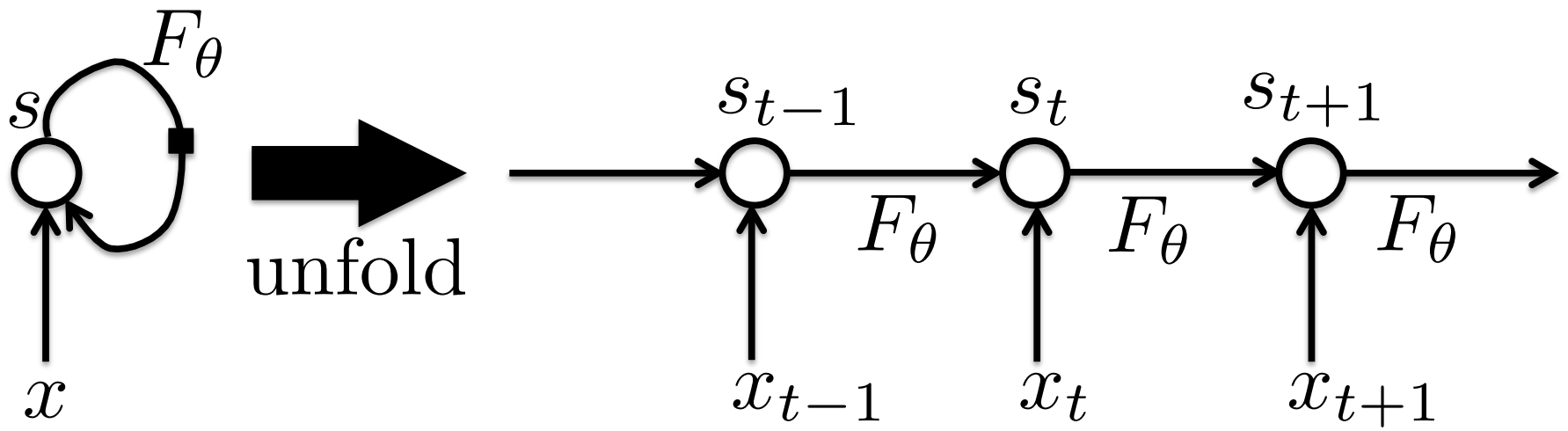
Issues with Undirected Graphical Models & Boltzmann Machines

- 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

Recurrent Neural Networks

- Selectively summarize an input sequence in a fixed-size state vector via a recursive update

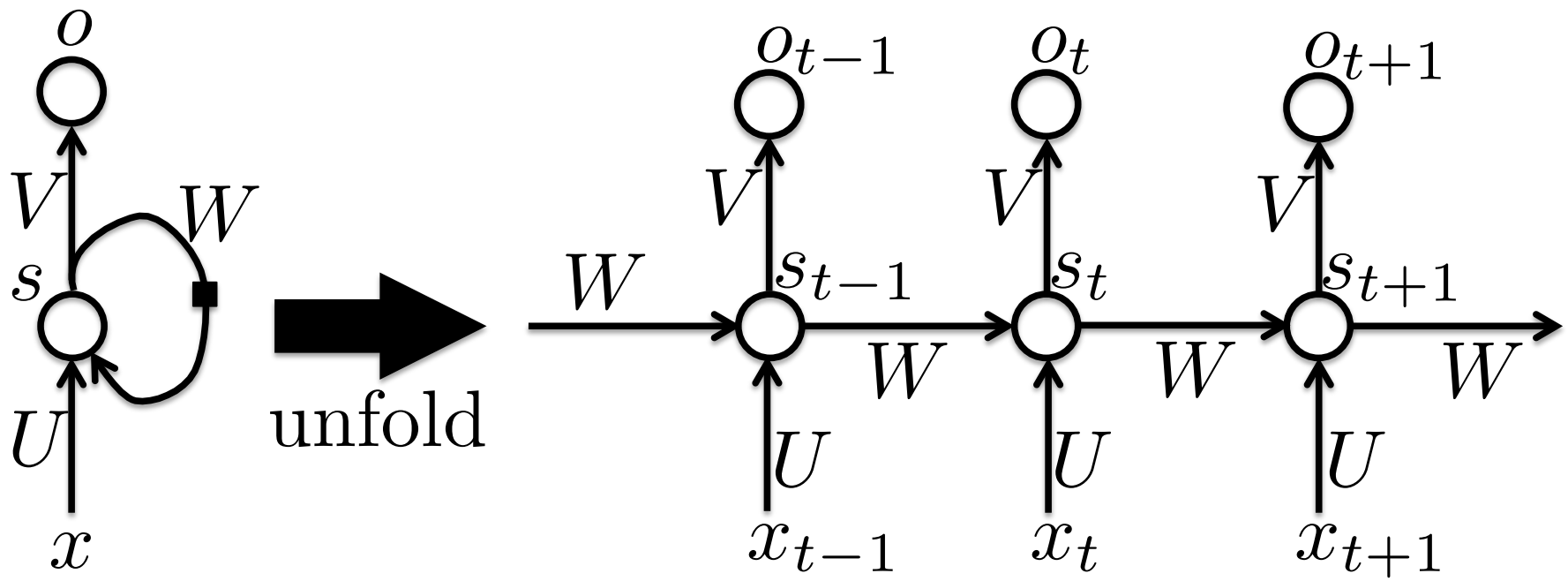
$$s_t = F_\theta(s_{t-1}, x_t)$$



$$s_t = G_t(x_t, x_{t-1}, x_{t-2}, \ldots, x_2, x_1)$$

Recurrent Neural Networks

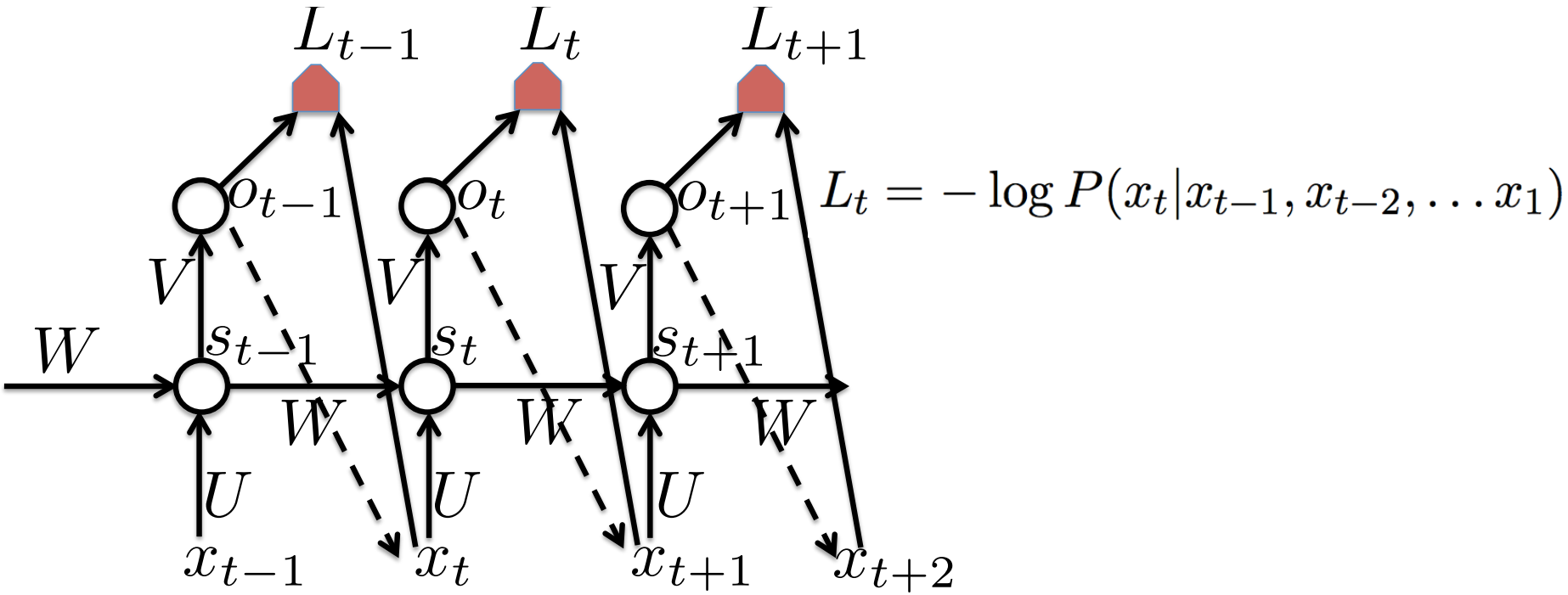
- Can produce an output at each time step: unfolding the graph tells us how to back-prop through time.



Generative RNNs

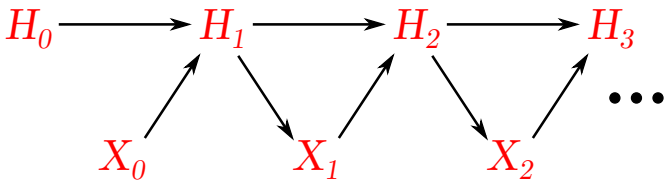
- An RNN can represent a fully-connected directed generative model: every variable predicted from all previous ones.

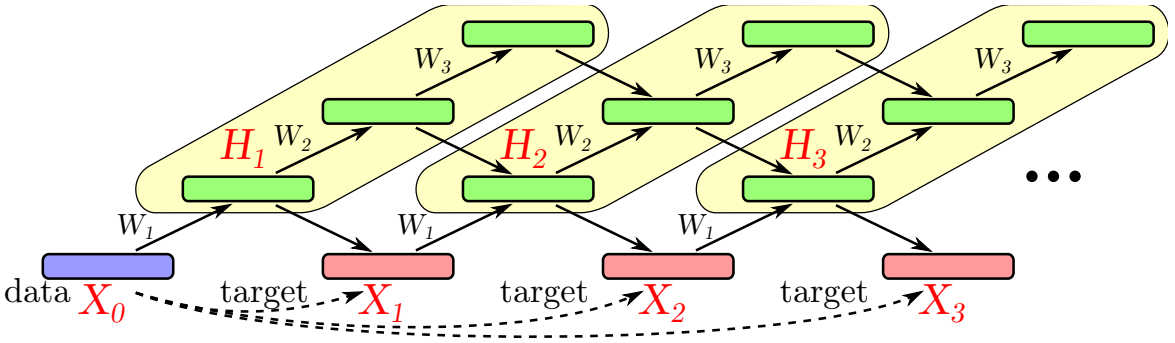
$$P(\mathbf{x}) = P(x_1, \ldots x_T) = \prod_{t=1}^{T} P(x_t | x_{t-1}, x_{t-2}, \ldots x_1)$$



Generative Stochastic Nets

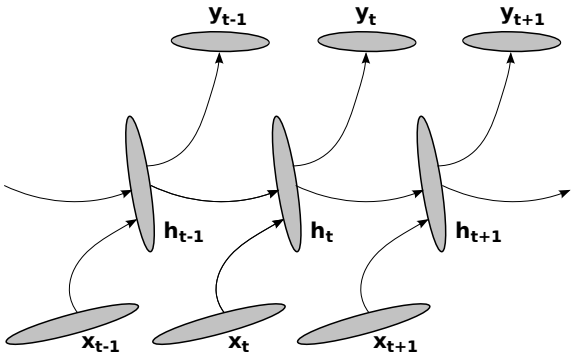
- Recurrent nets with noise injected and trained to reconstruct the visible variables (inputs, targets) are called GSNs
- ICML 2014 paper: they estimate the joint distribution of the visible variables via the stationary distribution of the Markov chain
- Can be trained via back-prop, no need to get reliable samples from the chain as part of training





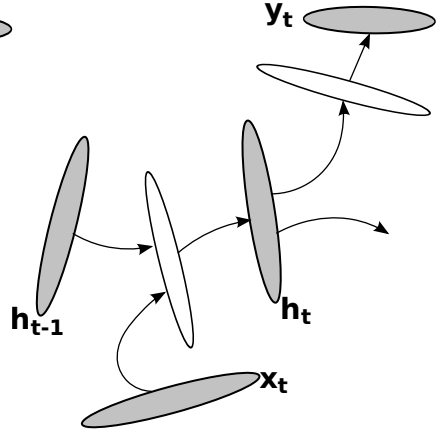
Increasing the Expressive Power of RNNs with more Depth

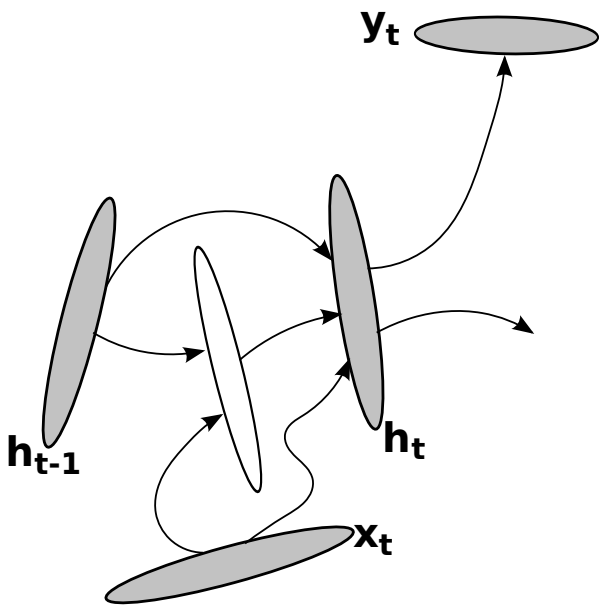
- ICLR 2014, *How to construct deep recurrent neural networks*



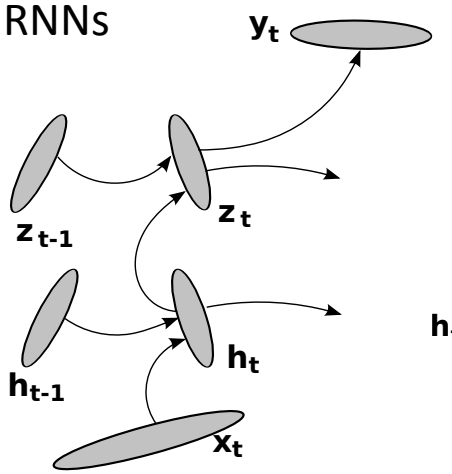
Ordinary RNNs

+ deep hid-to-out
+ deep hid-to-hid
+deep in-to-hid





+ stacking



+ skip connections for creating shorter paths

Long-Term Dependencies

- In very deep networks such as **recurrent networks**, the gradient is a product of Jacobian matrices, each associated with a step in the forward computation. It can become very small or very large quickly [Bengio et al 1994], and the locality assumption of gradient descent breaks down.

$$L = L(s_T(s_{T-1}(\ldots s_{t+1}(s_t, \ldots))))$$

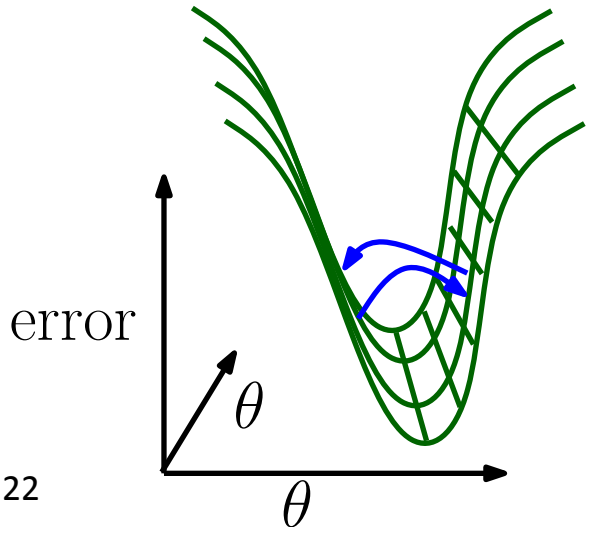
$$\frac{\partial L}{\partial s_t} = \frac{\partial L}{\partial s_T}\frac{\partial s_T}{\partial s_{T-1}}\cdots\frac{\partial s_{t+1}}{\partial s_t}$$

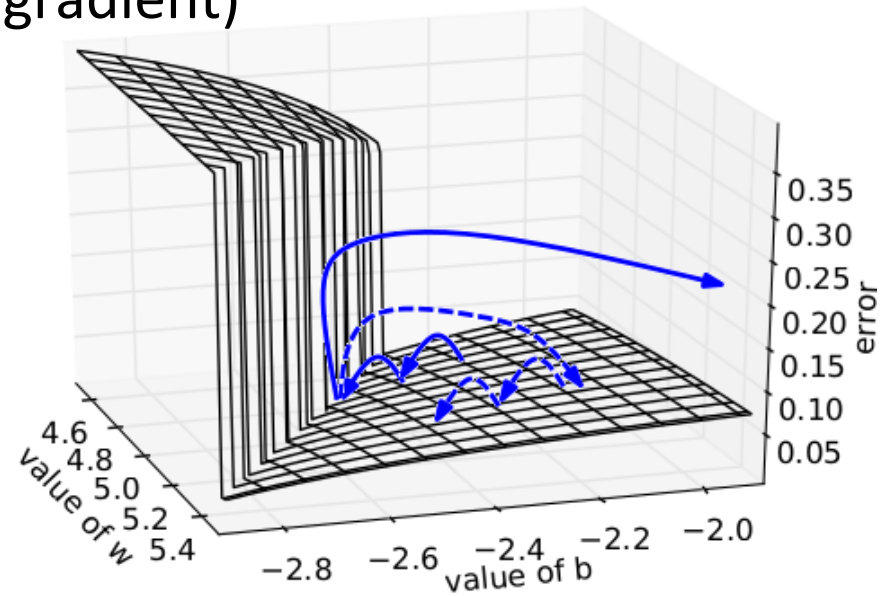
- Two kinds of problems:
  - sing. values of Jacobians > 1 → *gradients explode*
  - or sing. values < 1 → *gradients shrink & vanish*
  - or random → variance grows exponentially

RNN Tricks

(Pascanu, Mikolov, Bengio, ICML 2013; Bengio, Boulanger & Pascanu, ICASSP 2013)

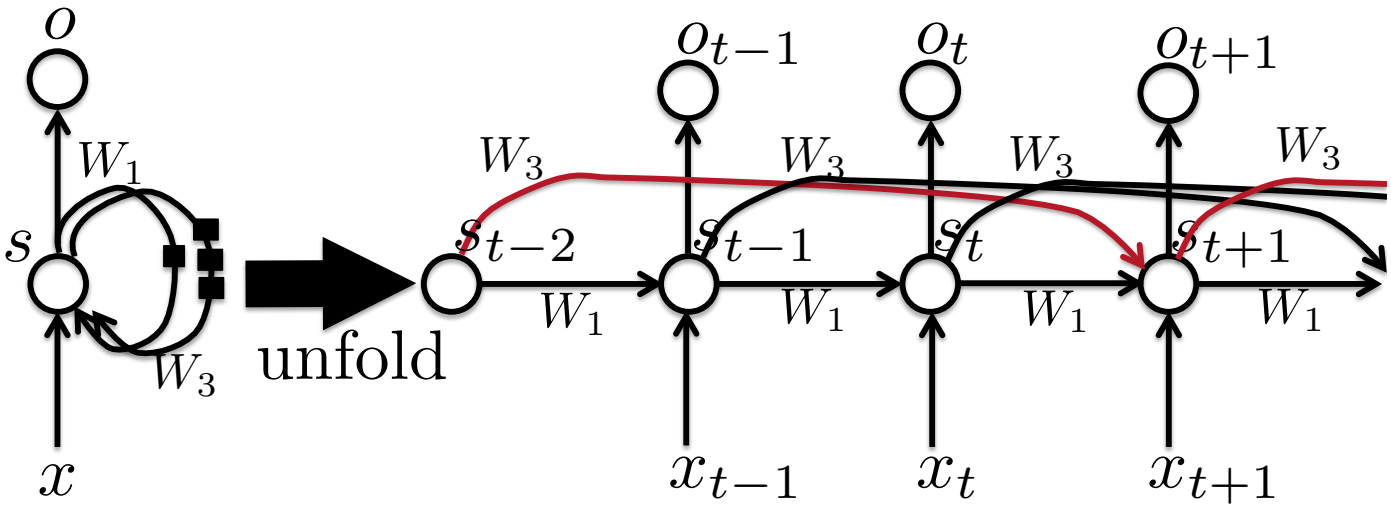
- Clipping gradients (avoid exploding gradients)
- Leaky integration (propagate long-term dependencies)
- Momentum (cheap 2nd order)
- Initialization (start in right ballpark avoids exploding/vanishing)
- Sparse Gradients (symmetry breaking)
- Gradient propagation regularizer (avoid vanishing gradient)
- LSTM self-loops (avoid vanishing gradient)

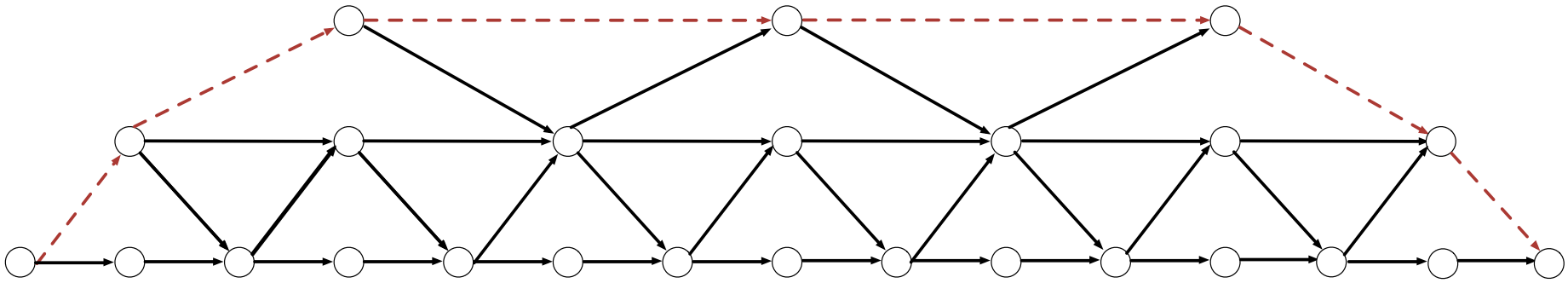




RNN Tricks

- Delays and multiple time scales, Elhihi & Bengio NIPS 1996

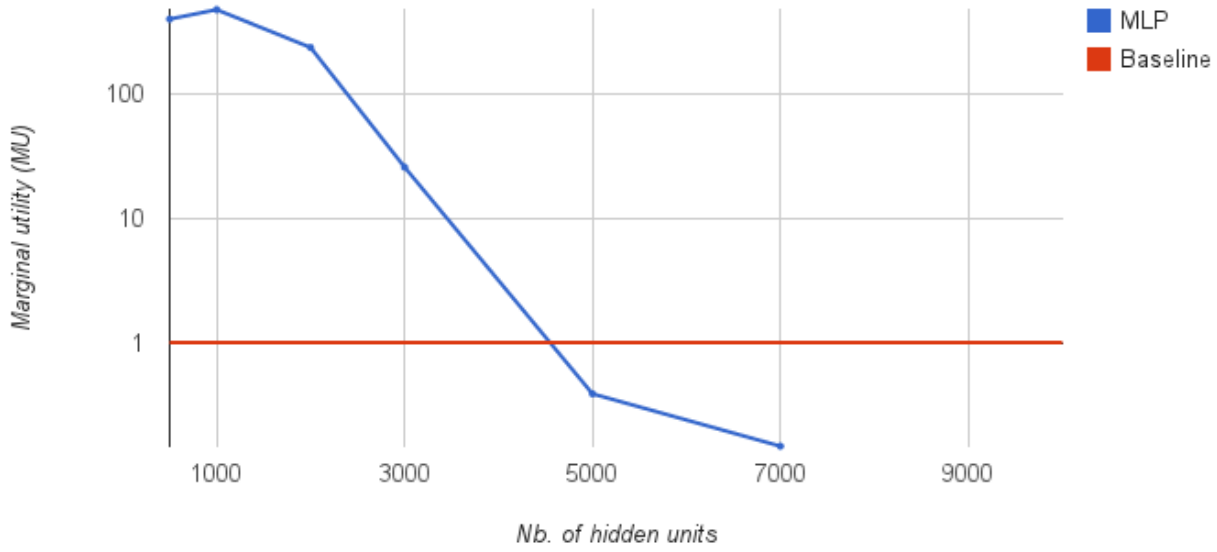




Optimization & Underfitting

- On large datasets, major obstacle is underfitting
- **Marginal utility** of wider tanh MLPs decreases quickly below memorization baseline

(Dauphin & Bengio, ICLR'2013)

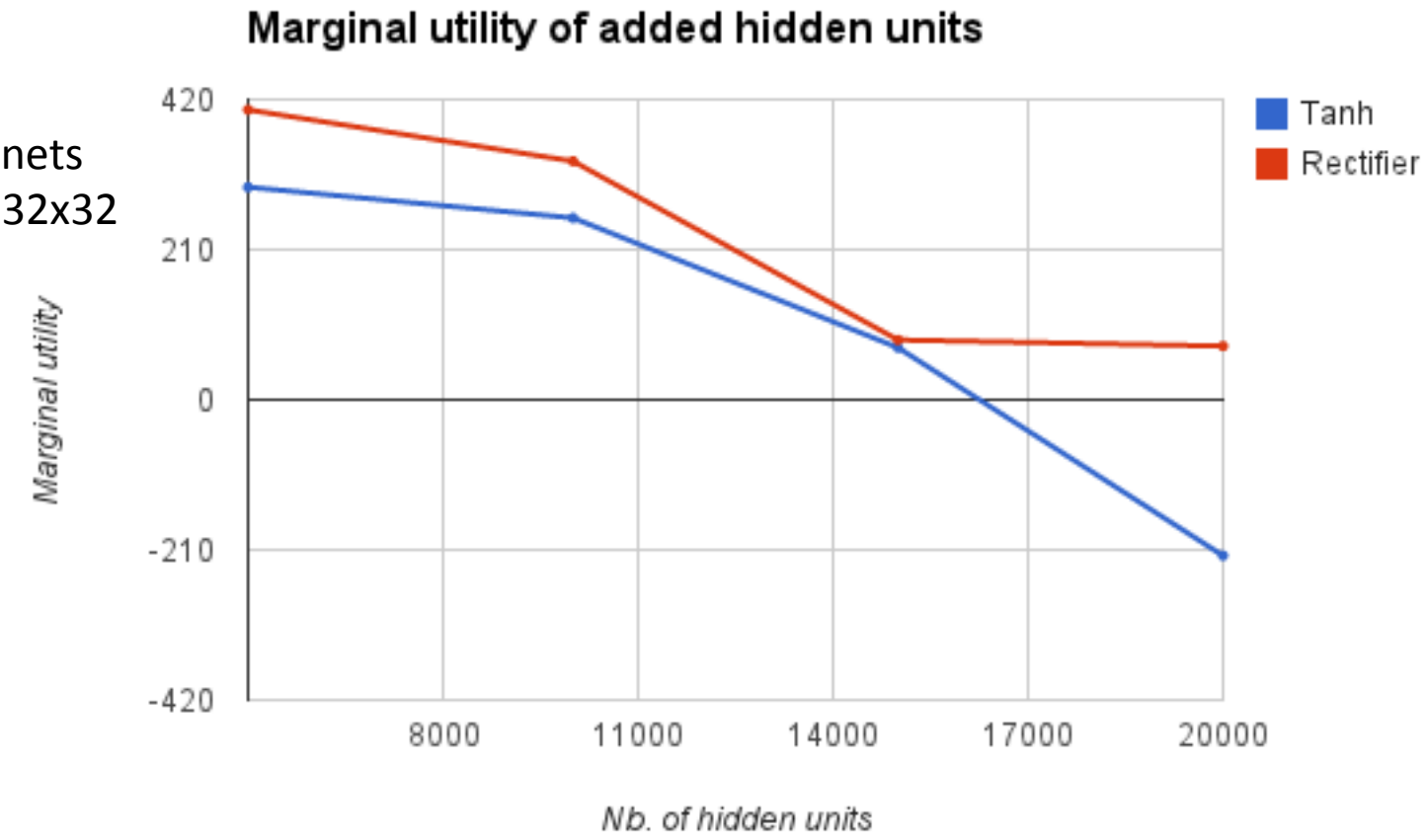


- Current limitations: local minima, ill-conditioning or else?

Easier Optimization with Rectifiers

- Why? Conjecture: Symmetry-breaking due to sparse gradients

Feedforward nets
on ImageNet 32x32

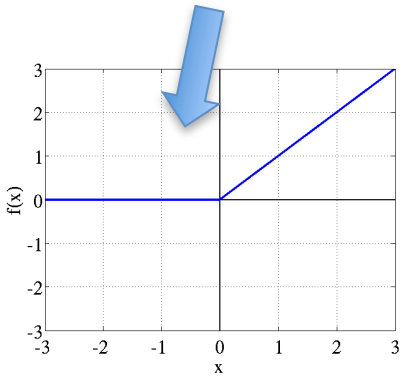


Deep Sparse Rectifier Neural Networks

(Glorot,Bordes and Bengio AISTATS 2011), following up on (Nair & Hinton 2010) softplus RBMs

Neuroscience motivations

Leaky integrate-and-fire model

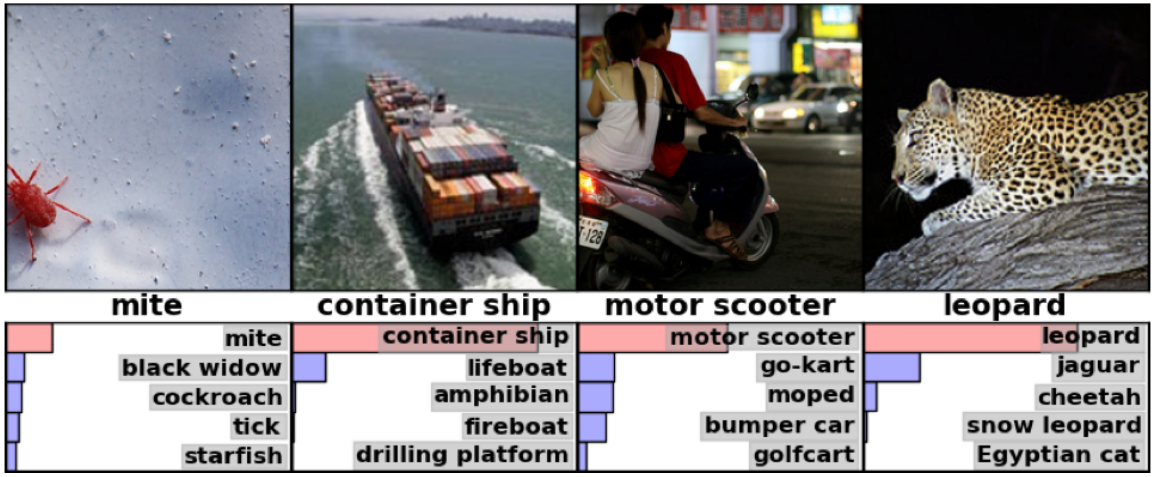


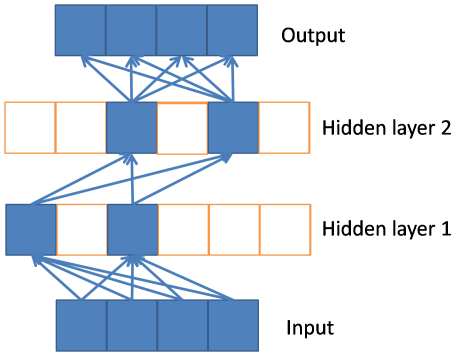
Rectifier

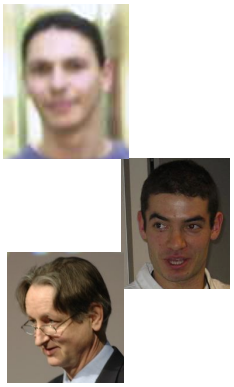
f(x)=max(0,x)

Machine learning motivations

- Sparse representations
- Sparse gradients
- *Trains deep nets even w/o pretraining*







Outstanding results by Krizhevsky et al 2012 killing the state-of-the-art on ImageNet 1000:

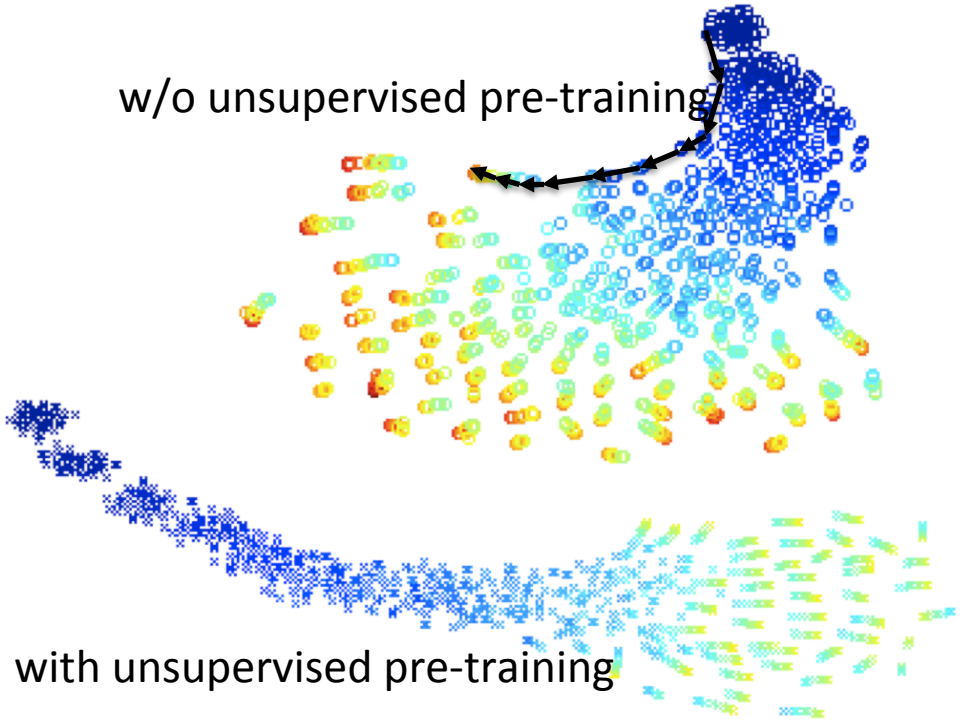
	1st choice	Top-5
2nd best		27% err
Previous SOTA	45% err	26% err
Krizhevsky et al	37% err	15% err

Effect of Initial Conditions in Deep Nets

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

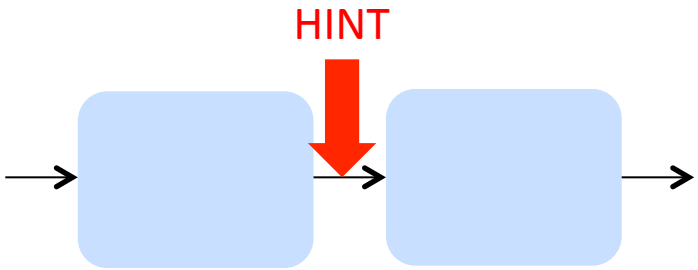
Neural net trajectories in function space, visualized by t-SNE

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



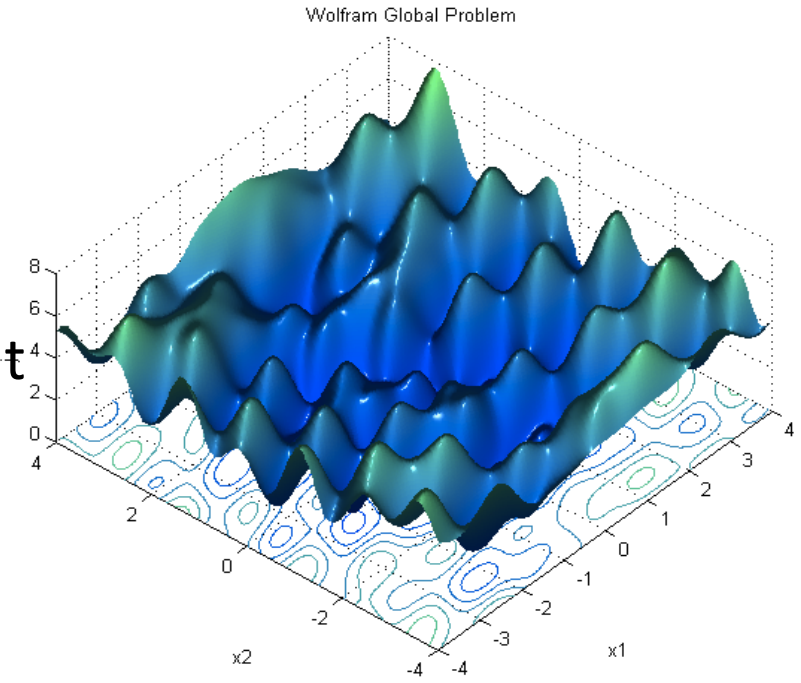
Guided Training, Intermediate Concepts

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

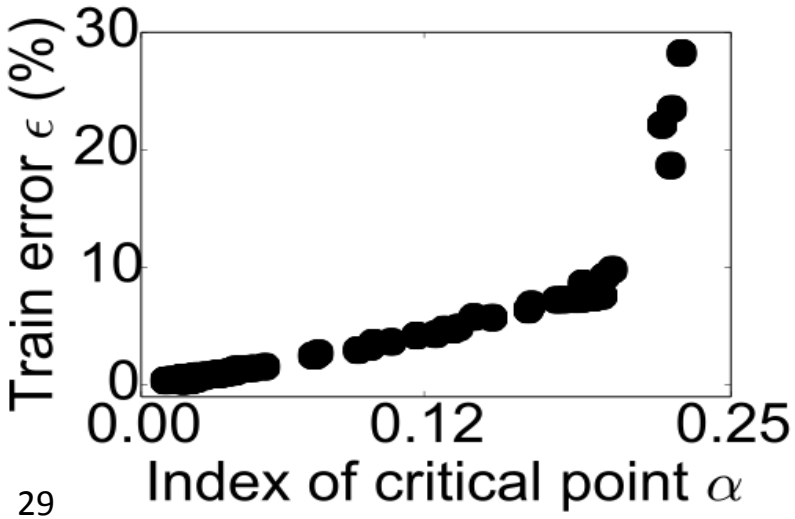


Saddle Points

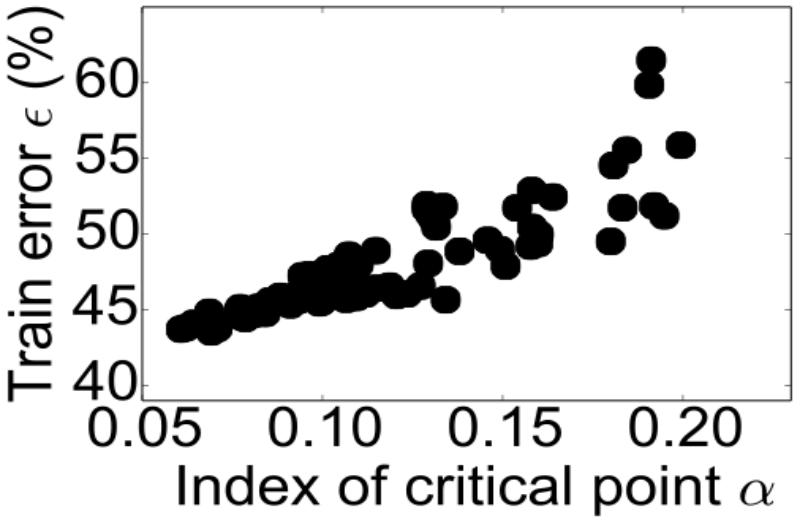
- Local minima dominate in low-D, but saddle points dominate in high-D
- Most local minima are close to the bottom (global minimum error)



MNIST



CIFAR

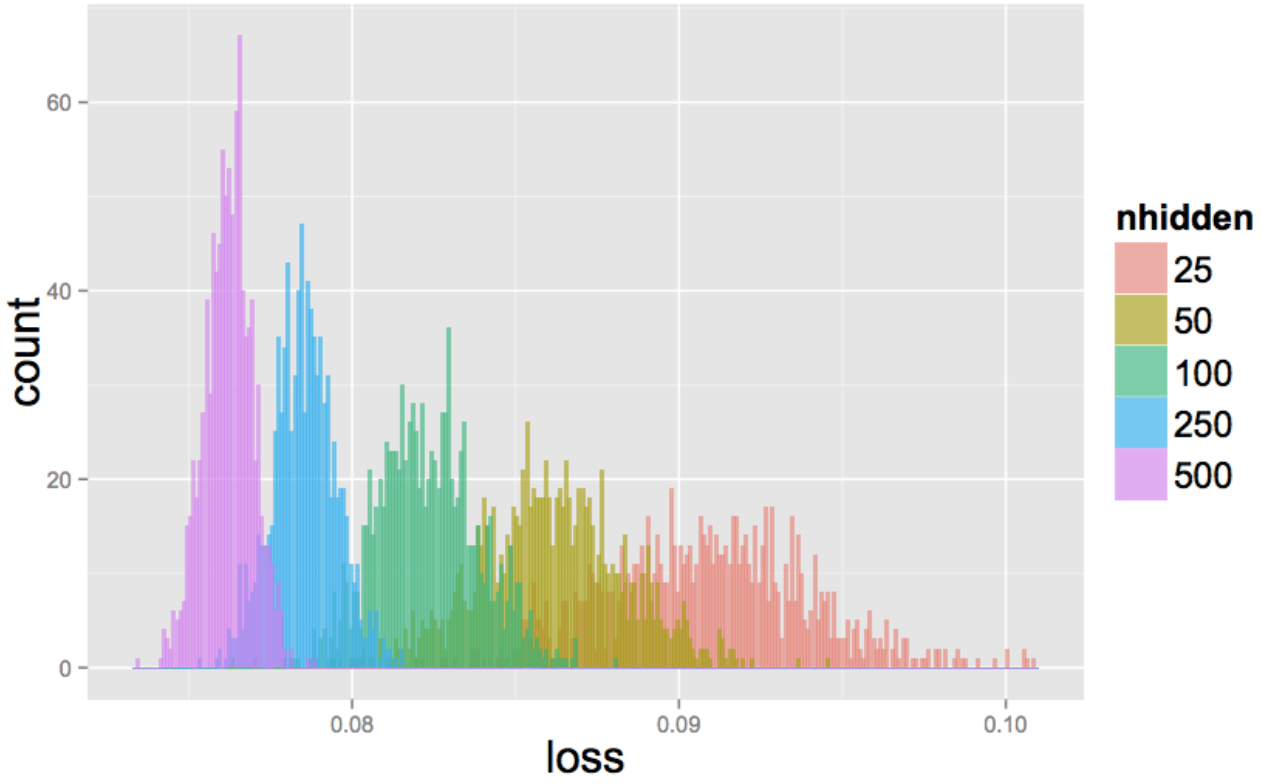


Low Index Critical Points

Choromanska et al & LeCun 2014, 'The Loss Surface of Multilayer Nets'

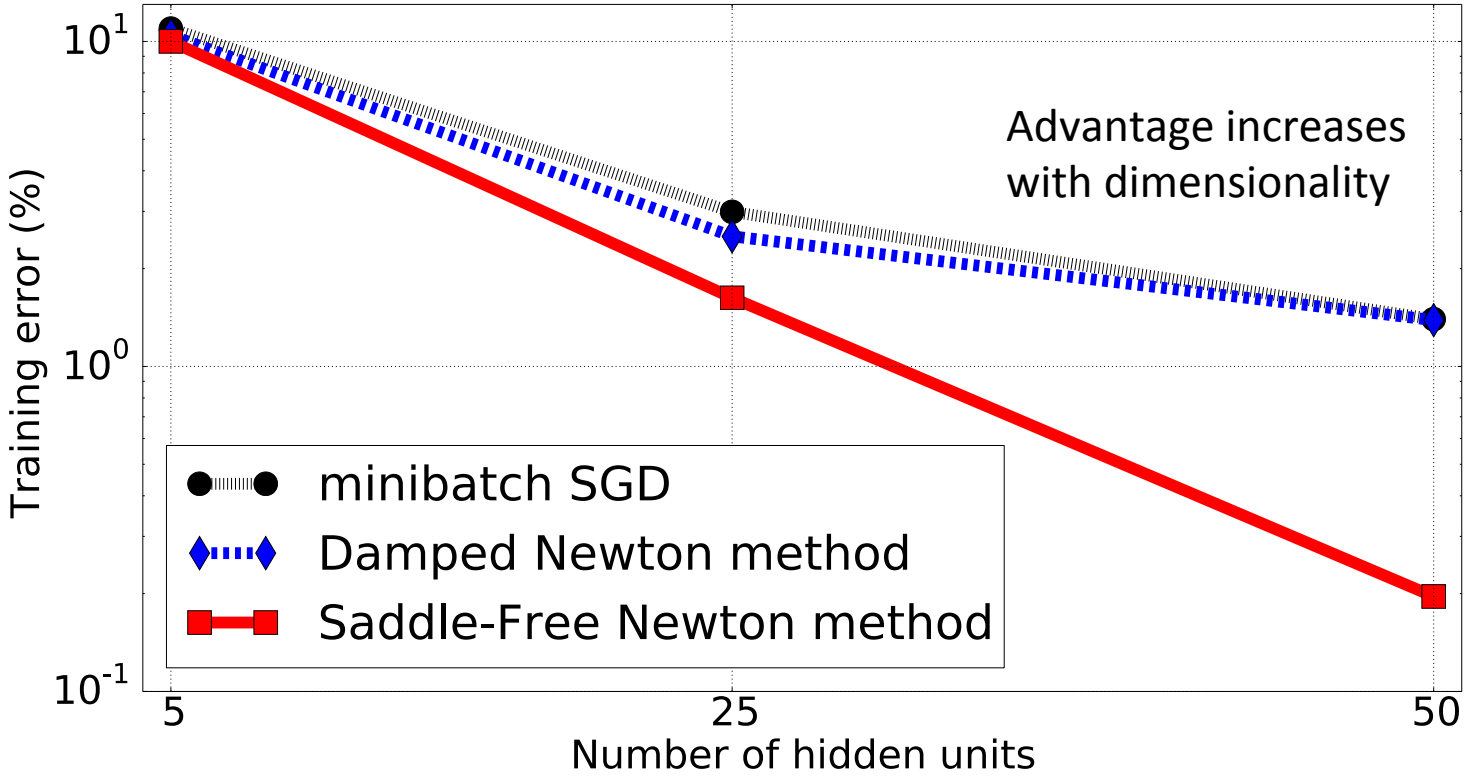
Shows that deep rectifier nets are analogous to spherical spin-glass models

The low-index critical points of large models concentrate in a band just above the global minimum



Saddle-Free Optimization (Pascanu, Dauphin, Ganguli, Bengio 2014)

- Saddle points are ATTRACTIVE for Newton's method
- Replace eigenvalues λ of Hessian by |λ|
- Justified as a particular trust region method

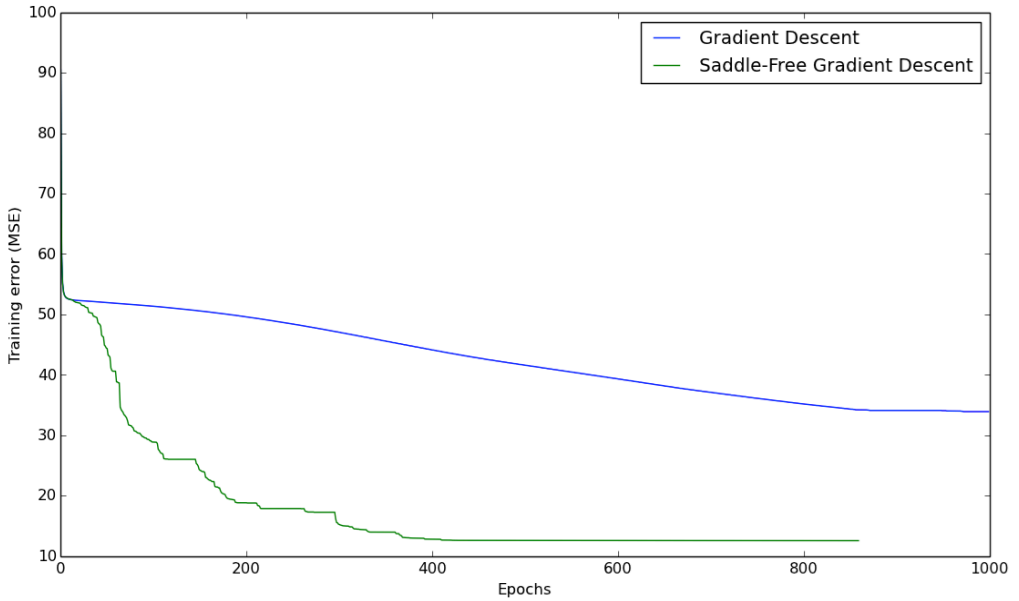


It is possible to escape saddle points!

- **NIPS'2014, *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,* Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio.**
- More work is ongoing to make it online
- Challenge: track the most negative eigenvector, which is easy in batch mode with power method, if we also track most positive, e.g.

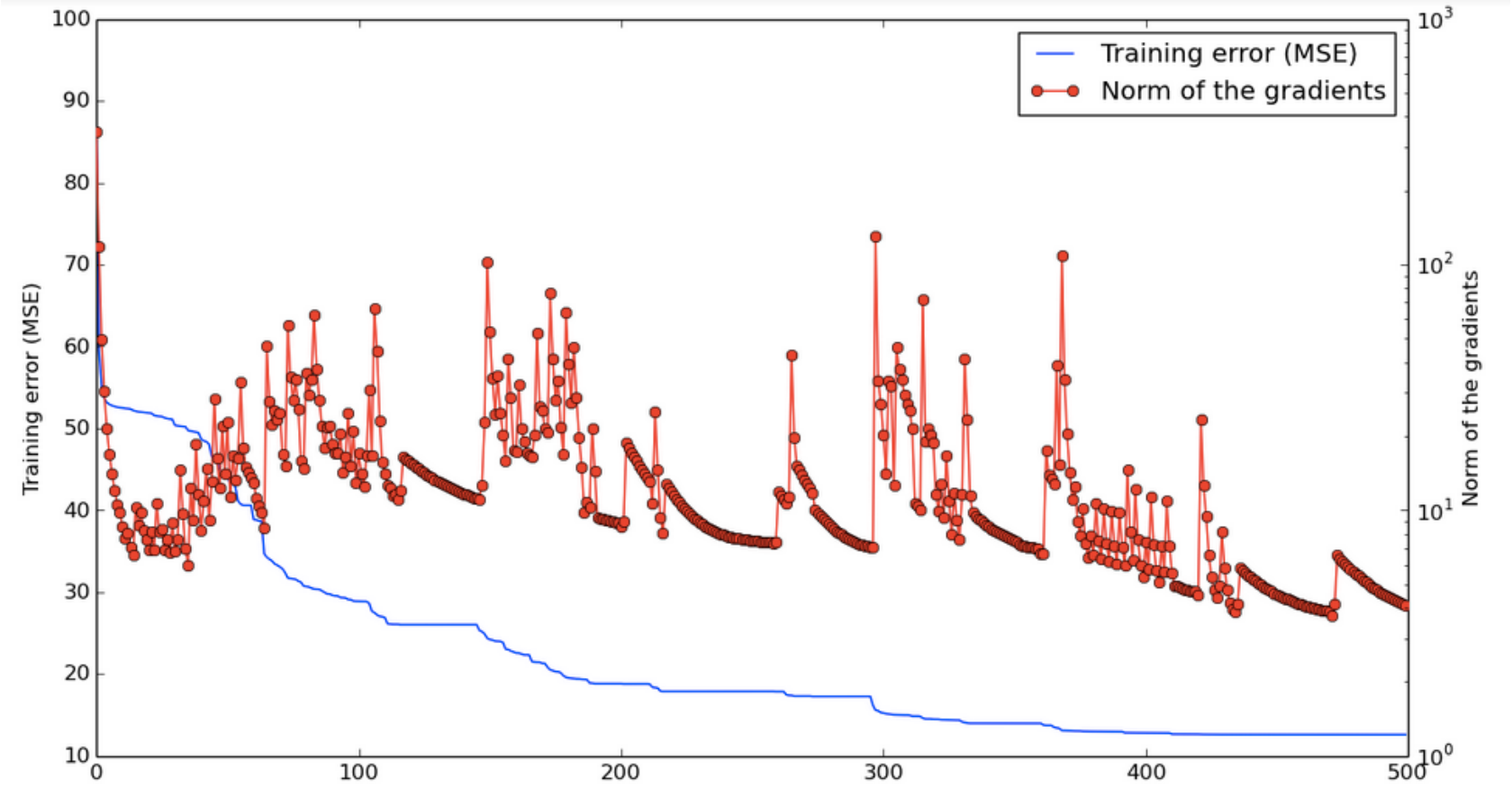
$$v \leftarrow (H - \lambda I)v$$

- The paper used a Krylov subspace method.

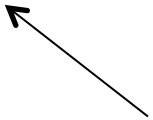


Saddle Points During Training

- Oscillating between two behaviors:
  - Slowly approaching a saddle point
  - Escaping it



Issue: underfitting due to combinatorially many poor *effective* local minima, most likely to be flat saddle points

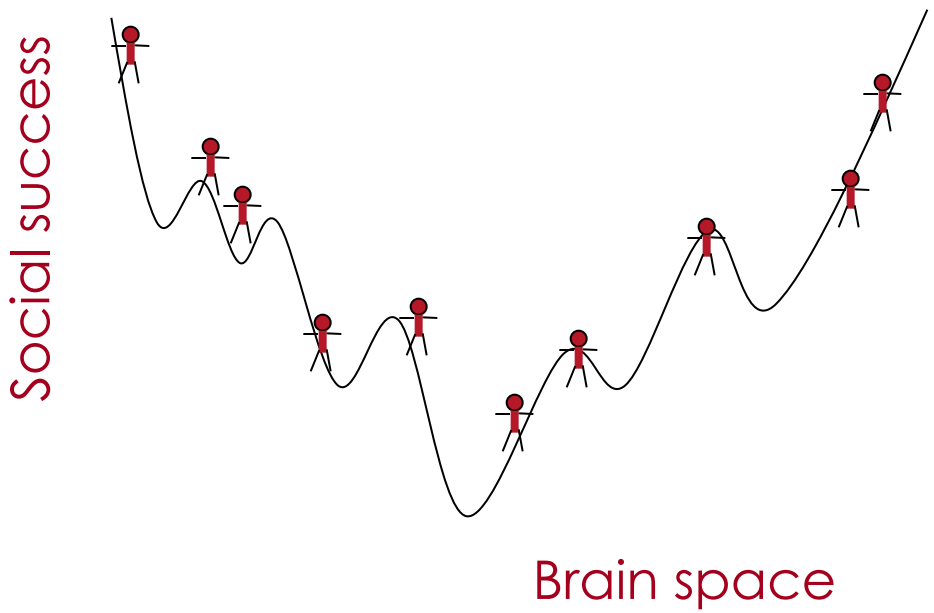


where the optimizer gets stuck

Culture vs Effective Local Minima

Bengio 2013 (also arXiv 2012)

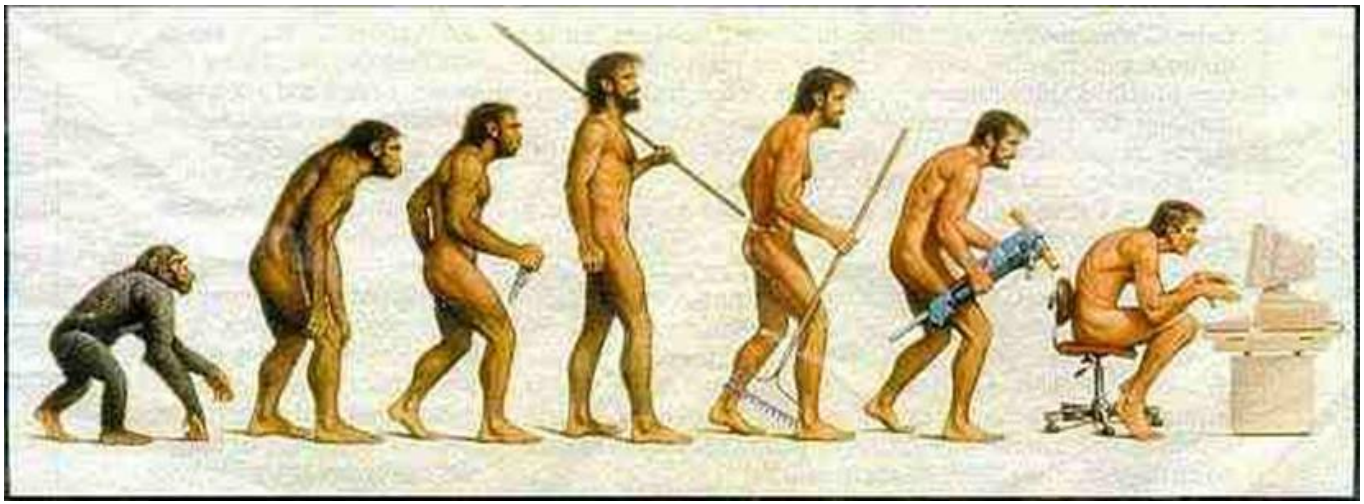
Parallelized exploration in brain space



- Each brain explores a potential solution
- Instead of exchanging synaptic configurations, exchange ideas through language

Memes

Genetic Algorithms	Evolution of ideas
Population of individuals	Population of brains
Recombination mechanism	Culture and language
Unit = Gene	Unit = Meme = idea



Hypothesis 1

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

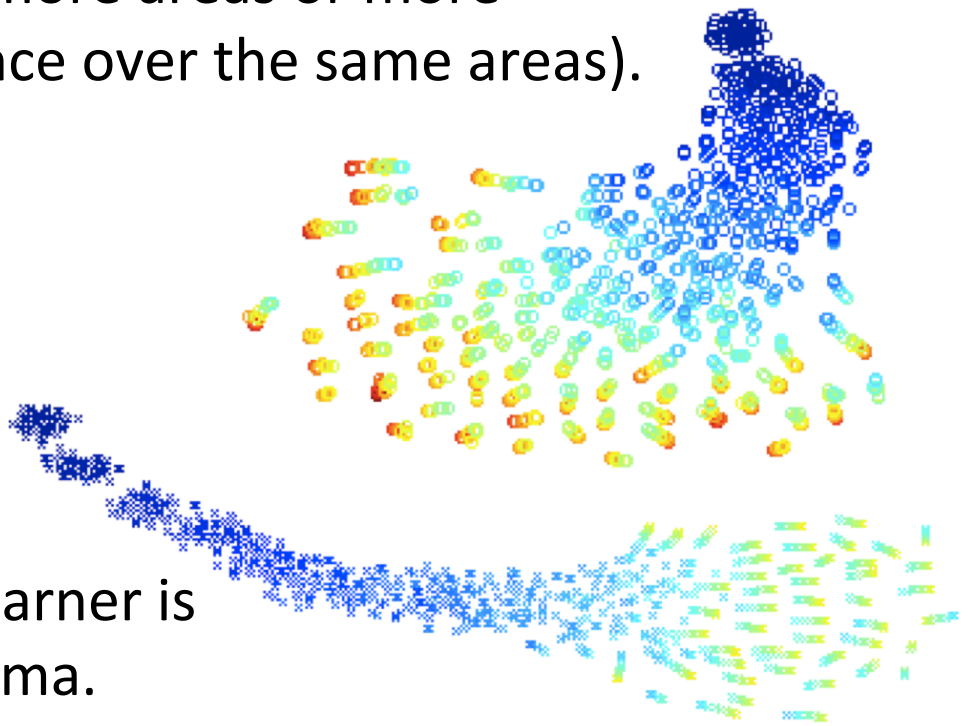
Hypothesis 2

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

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).



Hypothesis 4

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

Possibly due to ill-conditioning and flat saddle points but behaves like local min

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.

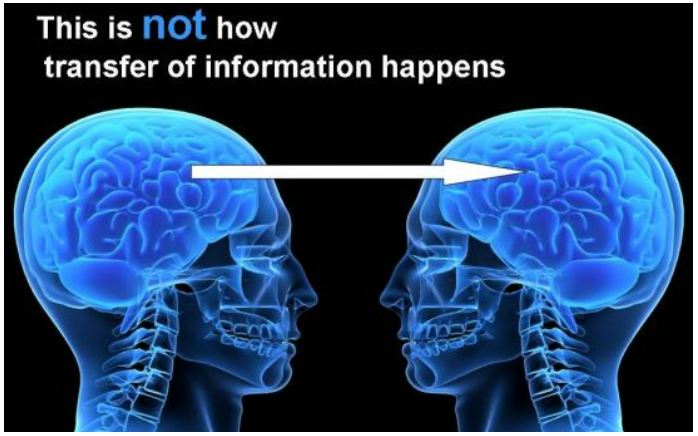
Hypothesis 6

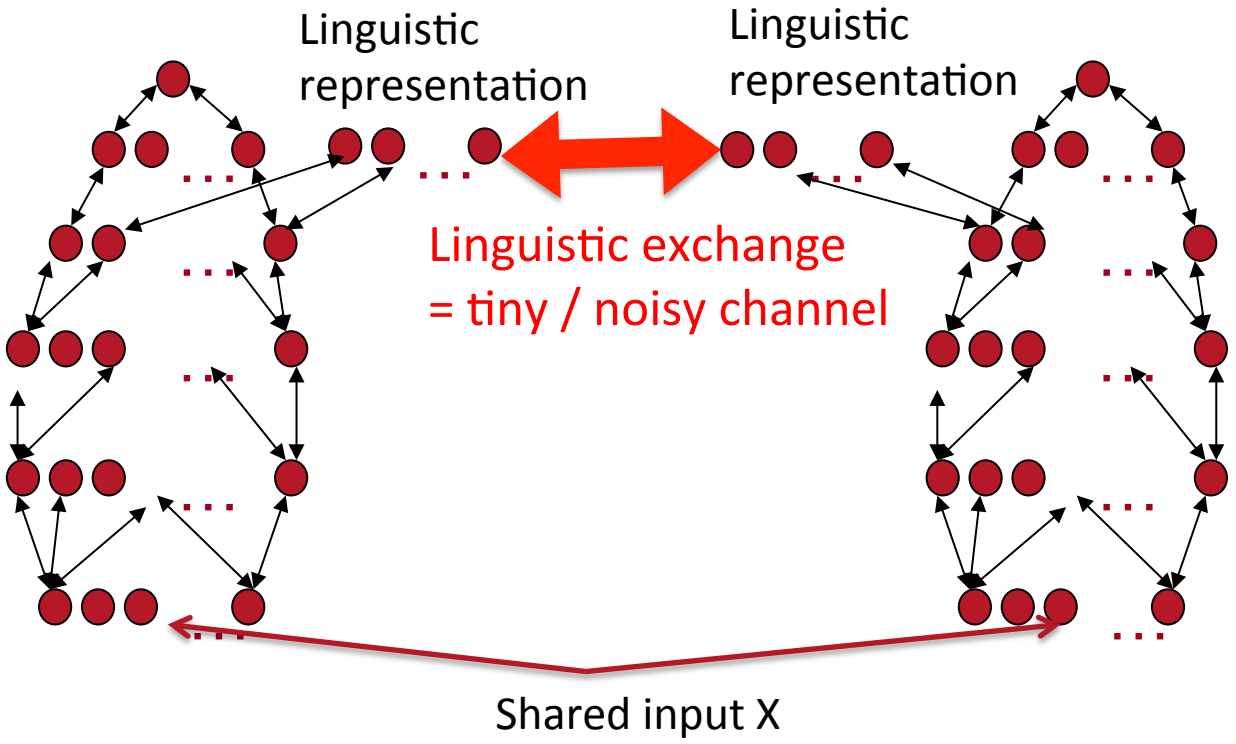
Curriculum learning (Bengio et al ICML 2009)

- 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.

Supporting evidence: (Gulcehre & Bengio ICLR 2013)

How is one brain transferring abstractions to another brain?





How do we escape local minima?

- linguistic inputs = extra examples, summarize knowledge
- criterion landscape easier to optimize (e.g. curriculum learning)
- turn difficult unsupervised learning into easy supervised learning of intermediate abstractions

How could language/education/culture possibly help find the better local minima associated with more useful abstractions?

More than random search: potential 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 where do new ideas emerge?

- 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)

Conclusions

- Deep learning involves a powerful prior but optimization can be difficult because we are trying to learn a highly non-linear function that has compositional structure.
- Very long-term dependencies remain a challenge for RNNs but much progress has been made, yielding SOTA in MT
- The myth that local minima are an issue for big deep nets is blown away by recent evidence, both theoretical and experimental
- Dealing with flat saddle points opens new avenues for research on optimization for deep learning.

MILA: Montreal Institute for Learning Algorithms