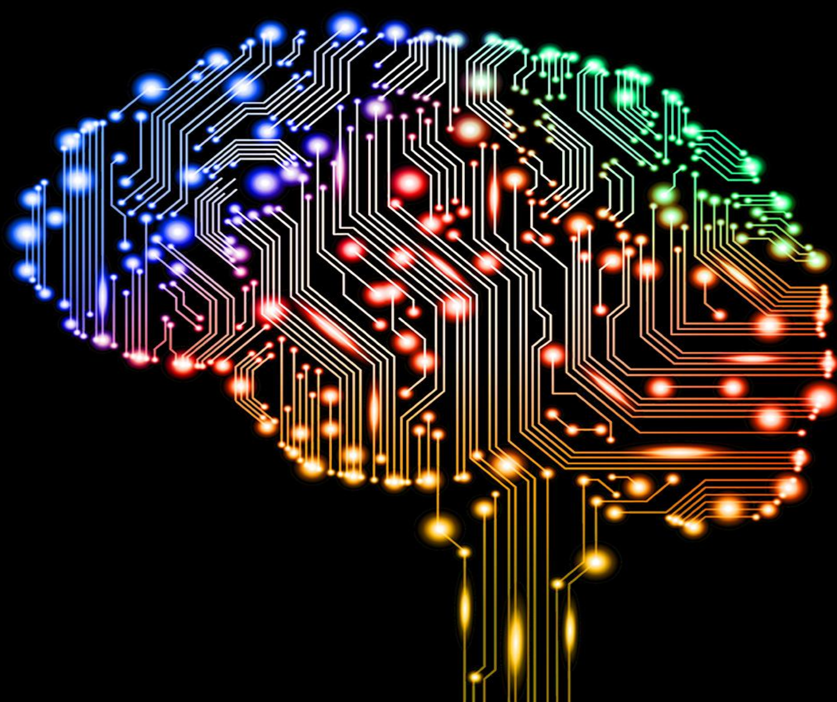
数据驱动的人工智能（6b）基于能量的神经网络

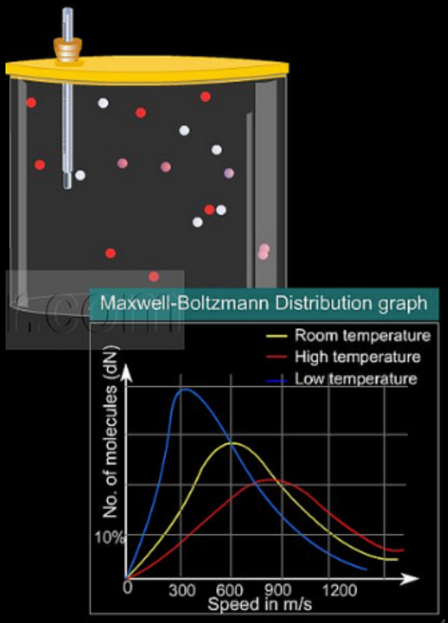
Data Driven Artificial Intelligence

郚学宁 SAP硅谷创新中心

2017 / 03

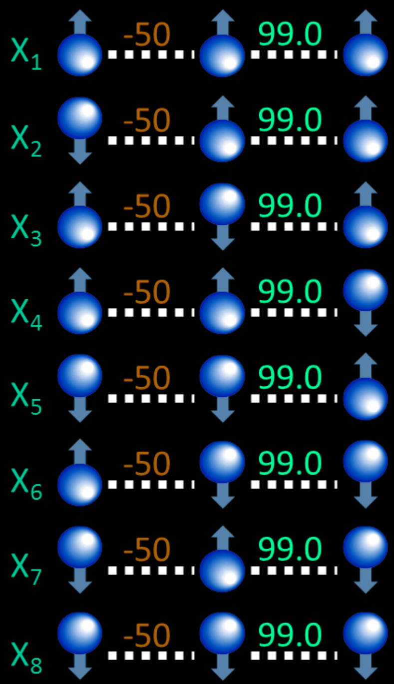


日程: EBN



- Bayesian Belief Network
- Auto-encoder / PCA
- Hopfield Network
- Boltzmann Machine
- Restricted Boltzmann Machine
- Deep Boltzmann Machine
- Deep Belief Network
- Sparse Coding
- Game Theory & Generative Adversarial Network

Ising Model Probabilities



$E(x_1,w) = -49$

$E(x_2,w) = -149$

$E(x_3,w) = +49$

$E(x_4,w) = +149$

$E(x_5,w) = +149$

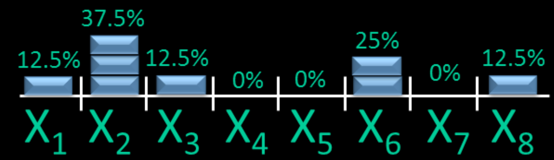
$E(x_6,w) = -149$

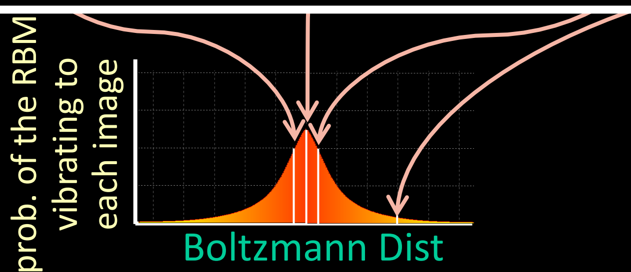
$E(x_7,w) = +49$

$E(x_8,w) = -49$

The probability of vibrating to each configuration x_i (given fixed weights w) can also be predicted by this formula:

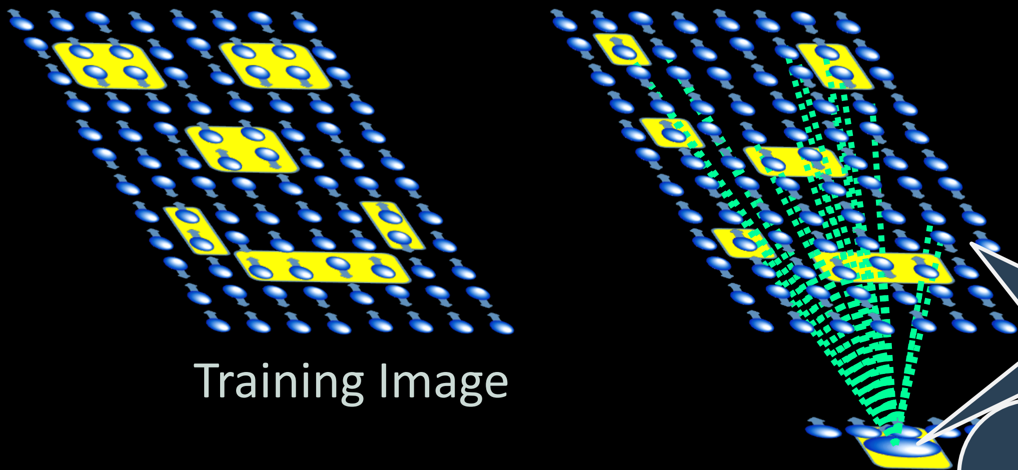
$$p(x_i|w) = \frac{e^{-E(x_i,w)}}{\sum_{k=1}^{n} e^{-E(x_k,w)}}$$





Learning an Image - Intuition

So what is actually happening to the wei
Gradient Descent algorithn



This will bias "vibration" of the system to configs resembling the trained image.

Why? Because in those random cases where our hidden atom activates…

it will generate our image on the visible plane.

And, any time the visible plane vibrates into a config that (remotely) resembles our training image…

this will likely result in the activation of our hidden atom…

which will then increase the odds of the trained image showing up, etc…

Let's imagine for a moment that we o

…our Gradient Descent algorithm would
atoms that need to be c

and negative weights (not depicte

“OK, so we know how to
train an RBM o
image… But ca
it to learn (to da
multiple images

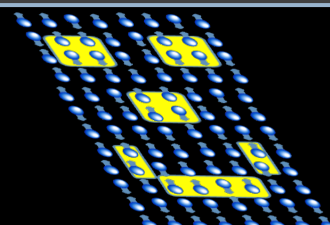
“You’re saying it’s possible to find a single set of weights that will create multiple such low-energy points?”

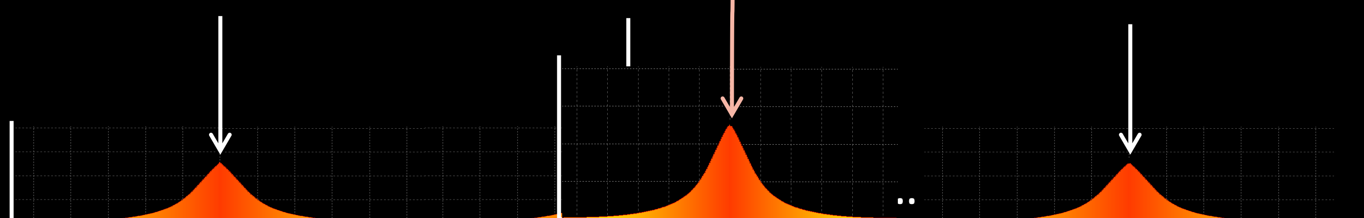
“Why not? Instead of using our Gradient Descent algorithm to learn weights to create a single low-energy state…

w

m

“That’s correct – if you have enough atoms/weights, the system can represent many low-energy states.”



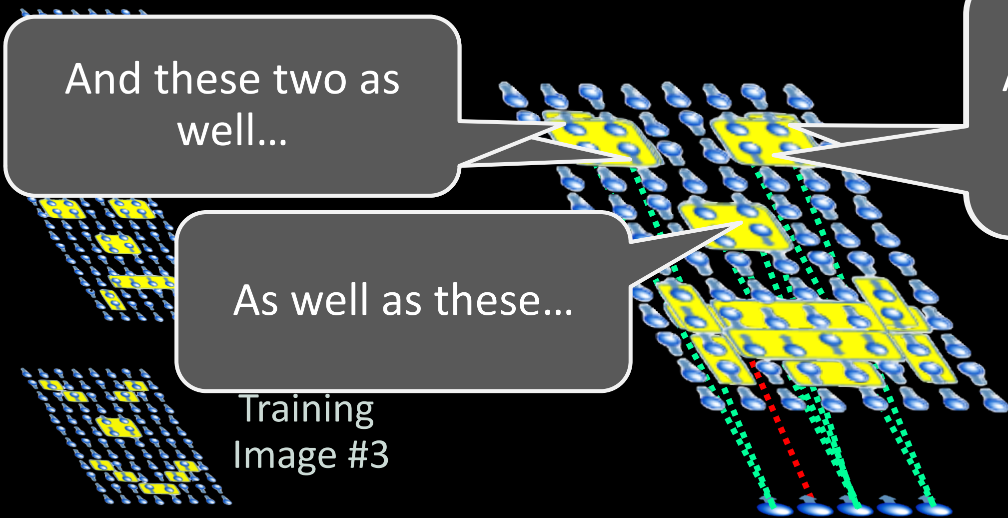


Boltzmann Dist

Boltzmann Distribution for multiple images

Learning Multiple Images - Intuition

So intuitively, how does the Gradient Descent algorithm discover a combination of weights that creates low energy states for multiple images?



Why?

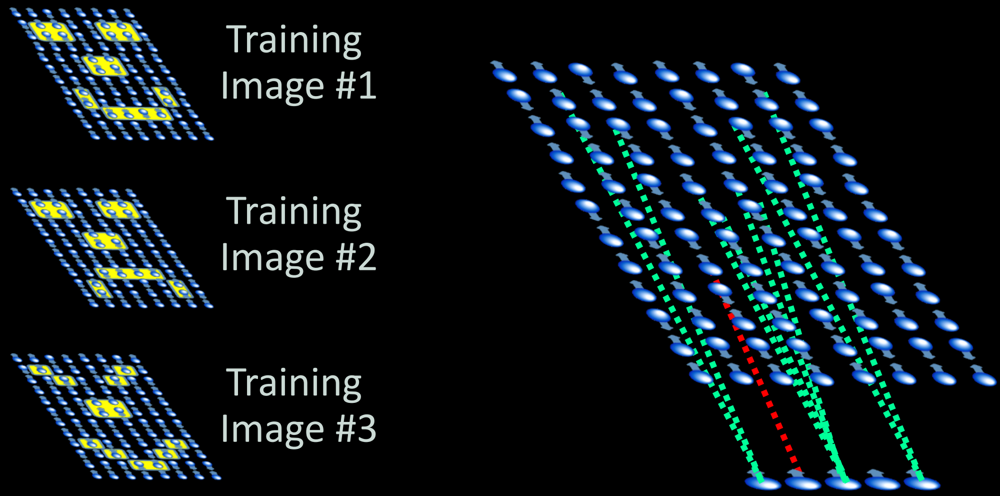
Notice that some atoms are actually consistently on across multiple images…

Our Gradient Descent algorithm identifies these "regularities" and ties them to hidden atoms with strong positive weights!

It'll similarly tie visible atoms that are consistently off across multiple training images to hidden atoms with strong negative weights.

Learning Multiple Images - Intuition

So intuitively, how does the Gradient Descent algorithm discover a combination of weights that creates low energy states for multiple images?



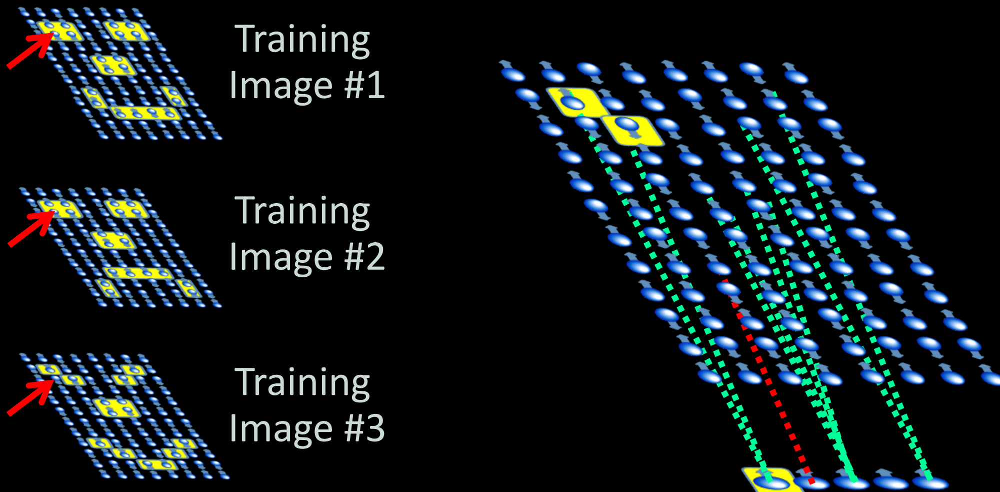
Why?

Notice that some atoms are actually consistently on across multiple images…

Our Gradient Descent algorithm identifies these "regularities" and ties them to hidden atoms with strong positive weights!

It'll similarly tie visible atoms that are consistently off across multiple training images to hidden atoms with strong negative weights.

Learning Multiple Images - Intuition



Associating a positive weight between a hidden atom

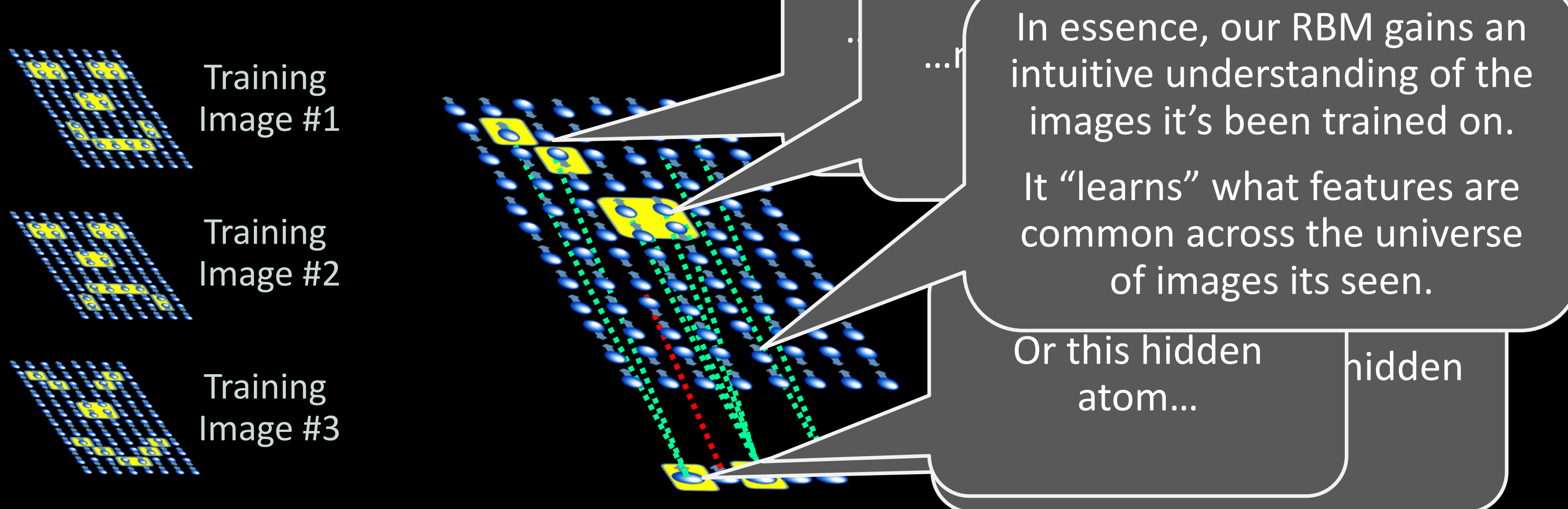
and one or more visible atom(s) that are frequently on across multiple training images…

means that any time that hidden atom activates,

there's a good chance it'll also activate visible atom(s) that are useful in reconstructing many of our training images …

this increases the odds that one of those multiple trained images will vibrate up on the RBM's visible surface!

Learning Multiple Images - Int



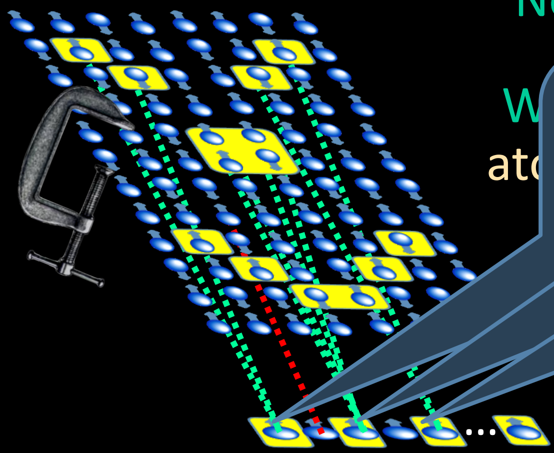
So the gradient descent process identifies "regularities" – common clusters of activated (or deactivated) visible atoms – across multiple training images…

And it associates these via stronger weights with various hidden atoms.

In practice, the identified regularities will be very simple – not complete eyes but rather simple edges and curves.

It's this exploitation of regularities that causes our RBM to display its learned images more frequently than arbitrary/random images.

An Interesting Intuition



Now once we've trained our RBM on a bunch of images…

W… ally flip our visible

at… hold them steady…

T…

to…

And it appears to have

part of a right eye…

…s they like?

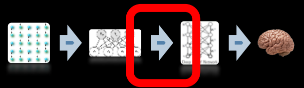
As expected, this would cause our hidden atoms to adjust based on our learned weights…

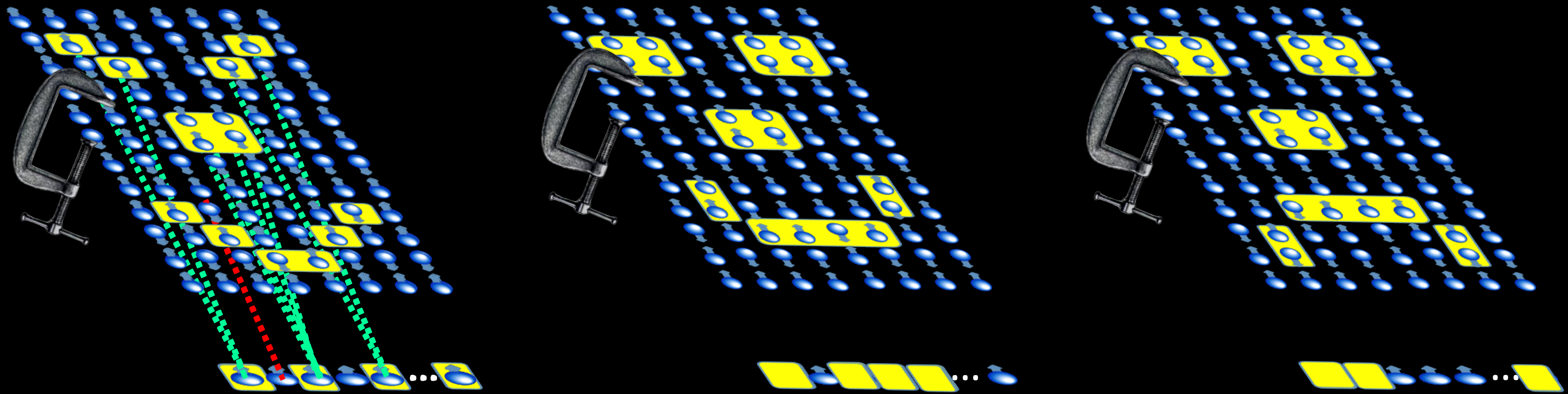
In essence, the resulting hidden atom configuration is a "code" that describes the set of regularities that were found within the visible image.

The hidden atom configuration thus represents a "high-level description" of the original image.

An Interesting Intuition

OK, let's take our trained RBM and use it to compute codes for each training image.





So now we have high-level descriptions for each of our original images:

Let's do something wacky with them!

An Interesting Intuition

... to these specific configurations of atoms.

…reate a s… …mann Machine

…us…

…his RB…

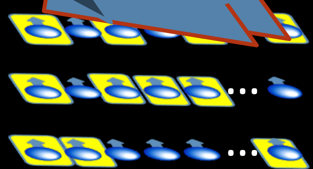
…ut trai…

Recall tha… associate…

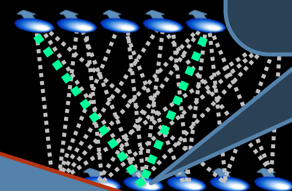
And, as before, to do so, we'll need to find regularities across our inputs…

So we'll learn the … …is

… and associate them with various hidden atoms.



Gradient Descent



So this atom might activate when an image has two eyes!

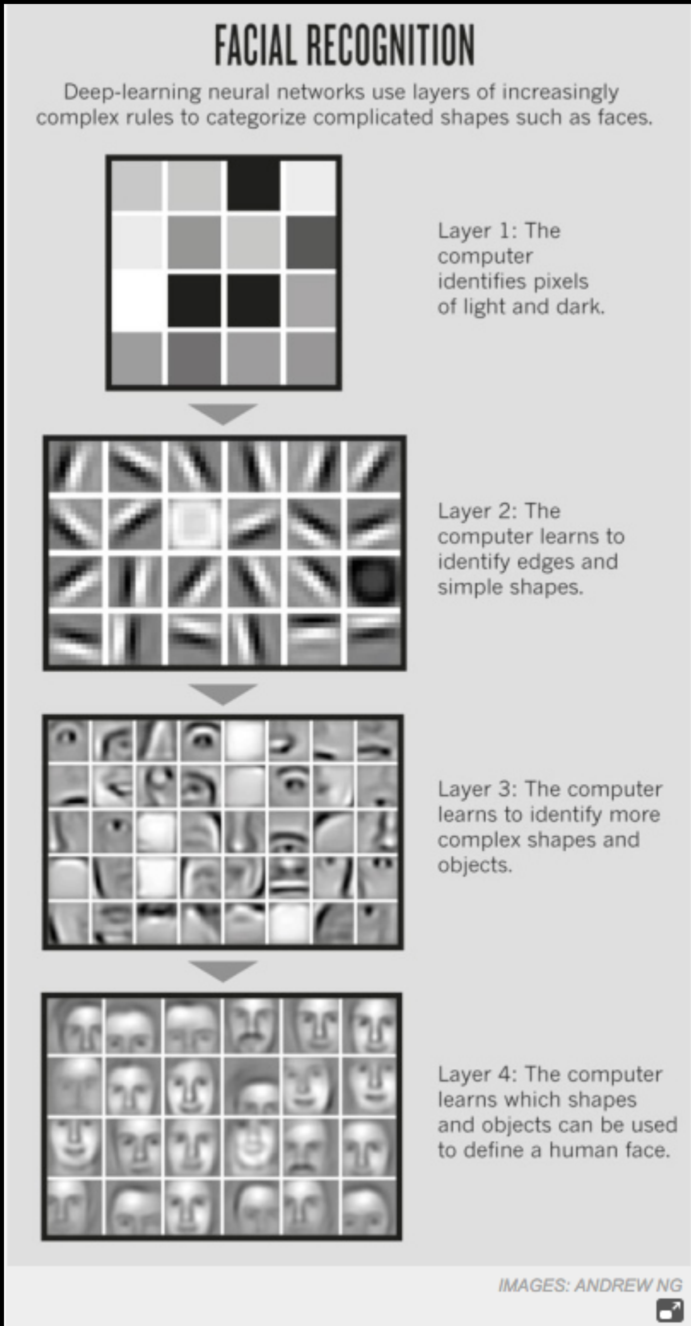
But wait – when we train o… …ts visible atoms don't represent raw dots i… …n our previous RBM…

Instead, as we learne… …his represent more abstract concepts like parts… …yes, curves of the mouth, etc.

As such, when our second RBM discovers regularities in its inputs, these will be even higher-order regularities - for instance that every face has two eyes.

ition

- n repeat this process over and over!
- n RBM layer, its hidden atoms “learn” successive bstraction about the original images!


FACIAL RECOGNITION

Deep-learning neural networks use layers of increasingly complex rules to categorize complicated shapes such as faces.

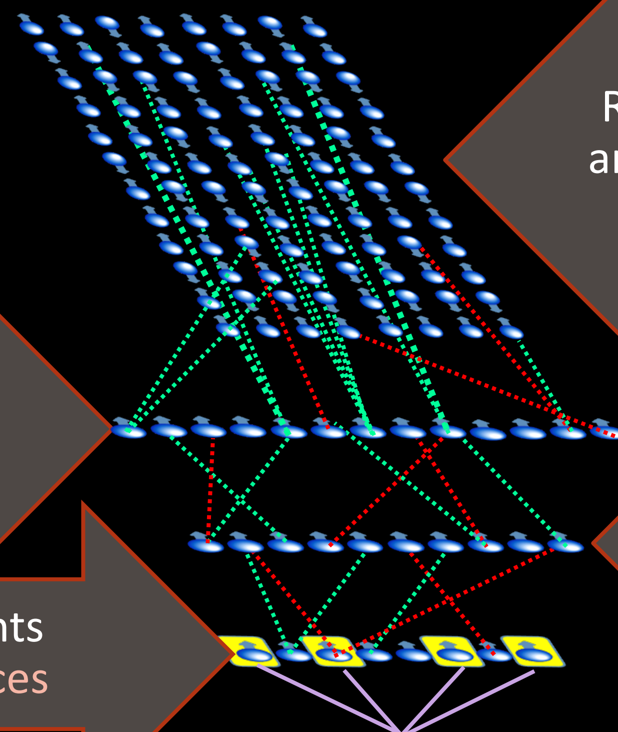
Layer 1: The computer identifies pixels of light and dark.

Layer 2: The computer learns to identify edges and simple shapes.

Layer 3: The computer learns to identify more complex shapes and objects.

Layer 4: The computer learns which shapes and objects can be used to define a human face.

IMAGES: ANDREW NG



Represents black and white “pixels” or dots...

le
ts
s

Represents combinations of facial components, like pairs of eyes

ents
faces

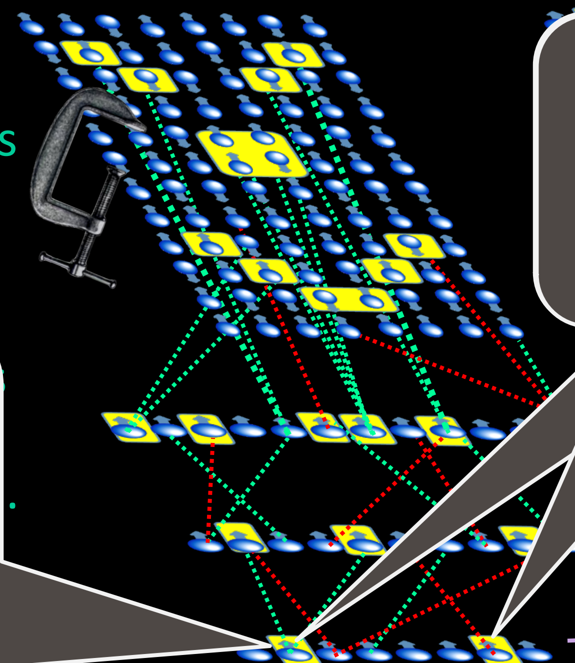
“Happy face with hair.”
“Sad girl's face.”

The Deep Belief Network

This network is called a "Deep Belief Network."

Once we have such a network, we can present an image to it at the top visible layer and fix it in place…

And then see how the atoms in the lower levels activate!

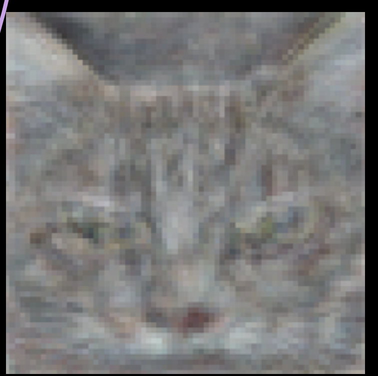


"I was just presented with a picture of a happy person!"

If trained with enough input images…

the DBN can actually learn to recognize common elements in those images (e.g., cats)…

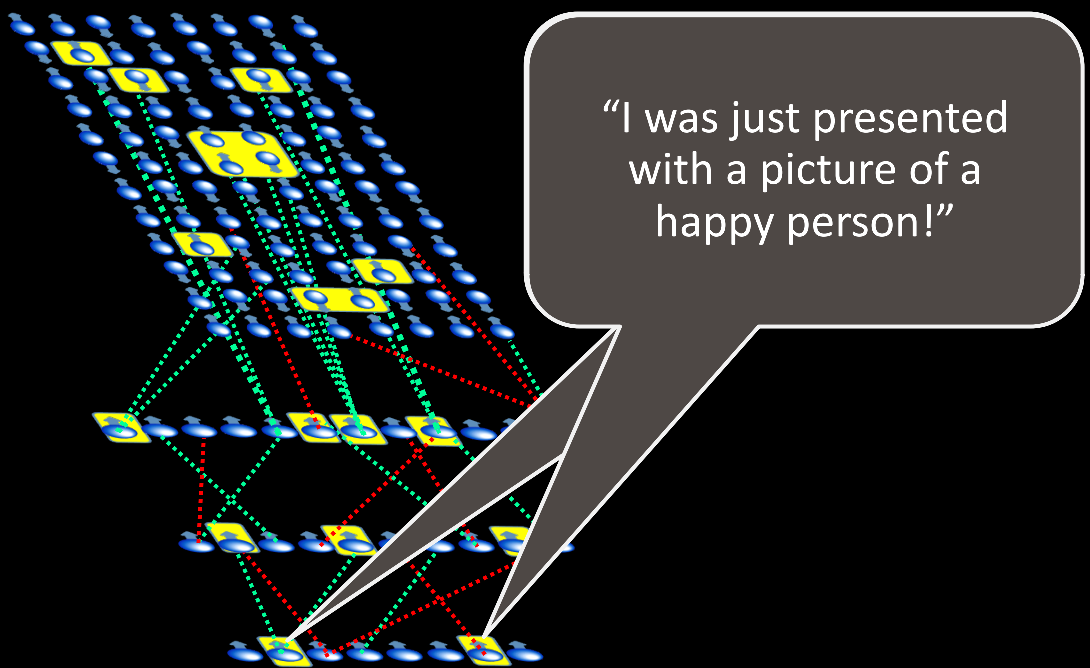
without ever being told what they are!



The Deep Belief Network: Recognition of Noisy Inputs

And the network is resilient to noise!

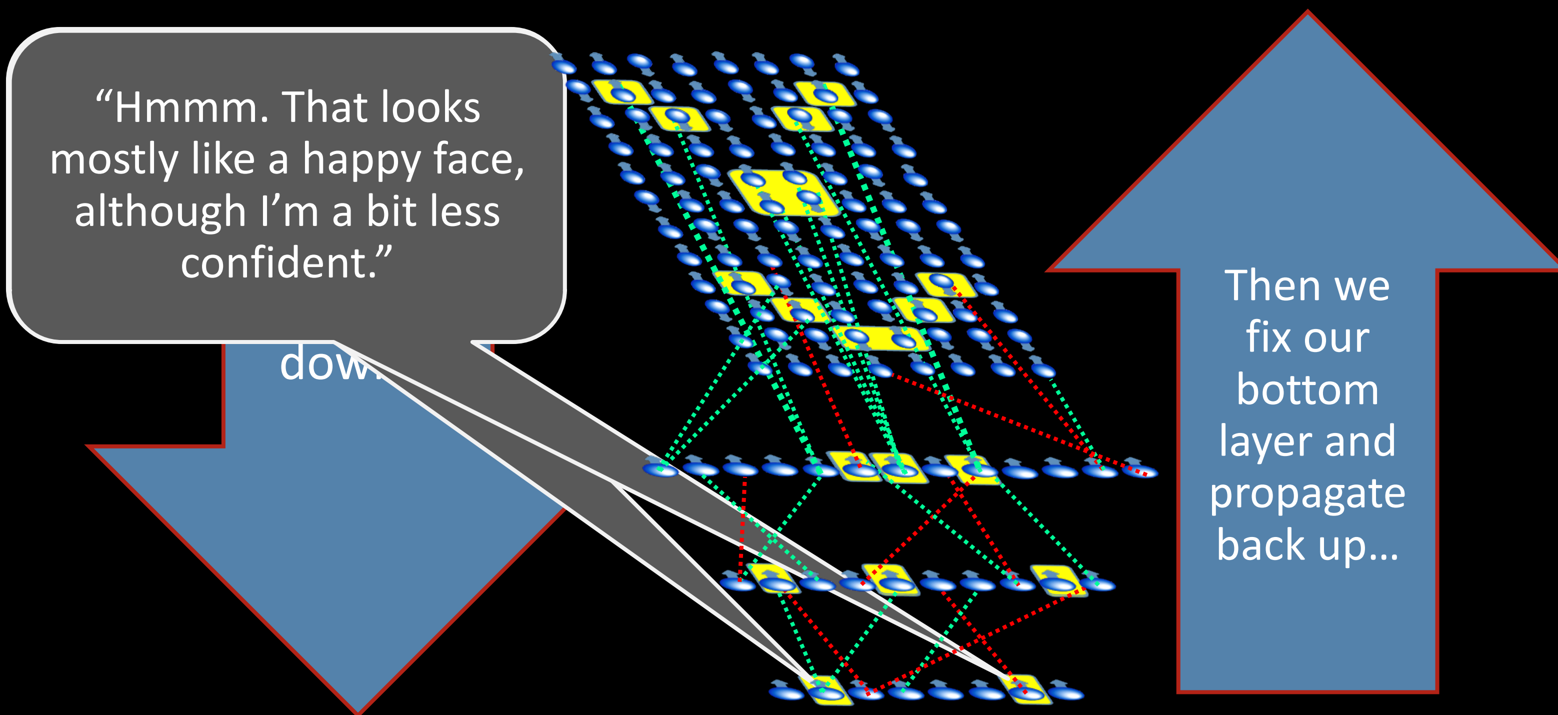
Even when presented with a partial image or a similar (but not identical) image, the network generally activates as expected.



The Deep Belief Network: Reconstruction

Even more exciting, the network can reproduce the original image from an incomplete or partial input!

All you have to do is propagate down and back up!

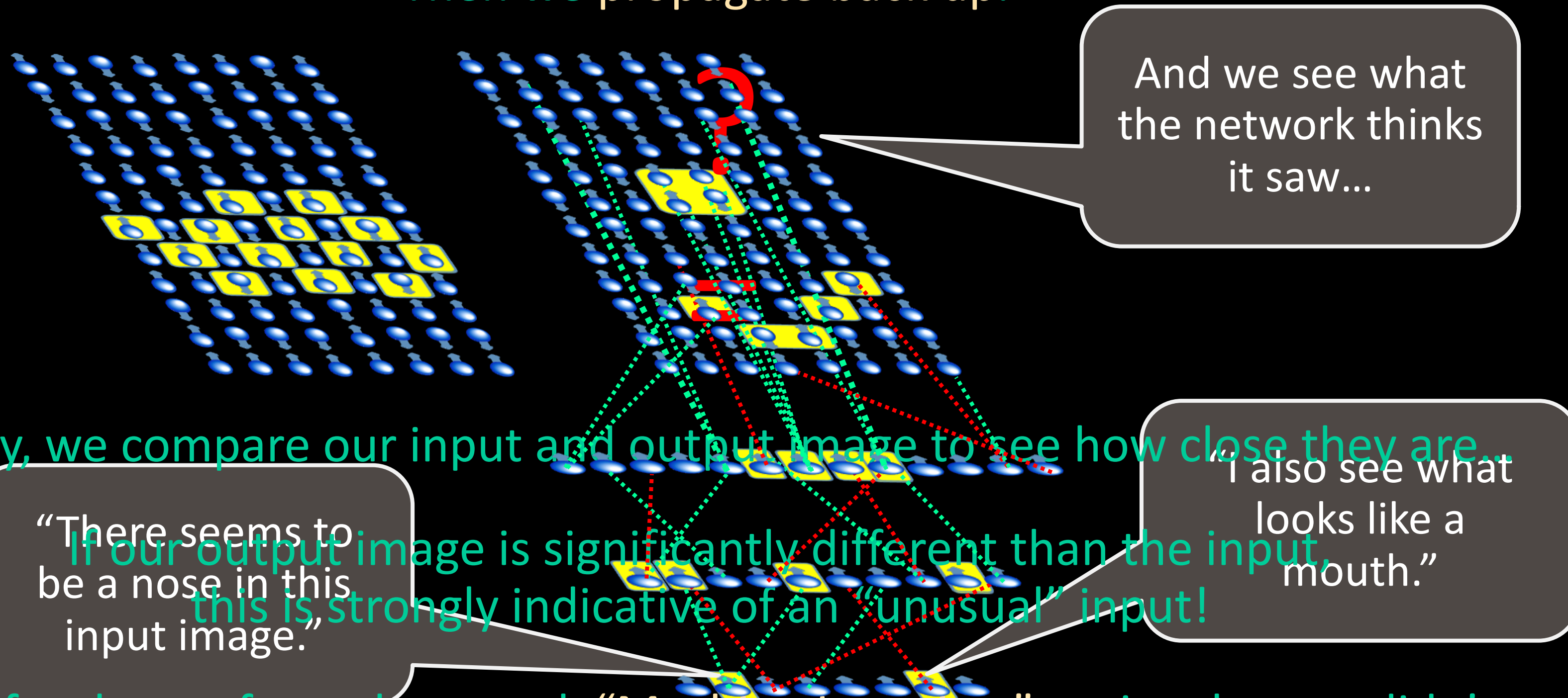


The Deep Belief Network: Anomaly Detection

Finally, we can use our DBN to detect anomalous inputs that don't match what our network has been trained on!

We take a new input image and propagate down…

Then we propagate back up.



Finally, we compare our input and output image to see how close they are…

If our output image is significantly different than the input, this is strongly indicative of an "unusual" input!

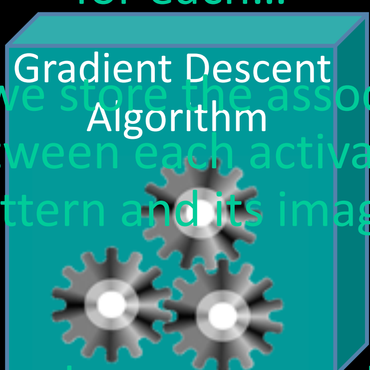
And for those of you that speak "Machine Learning," notice that we didn't need any ground truth data to perform this operation!

The Deep Belief Network: Fuzzy Searching

We can also use the network to search for similar items!

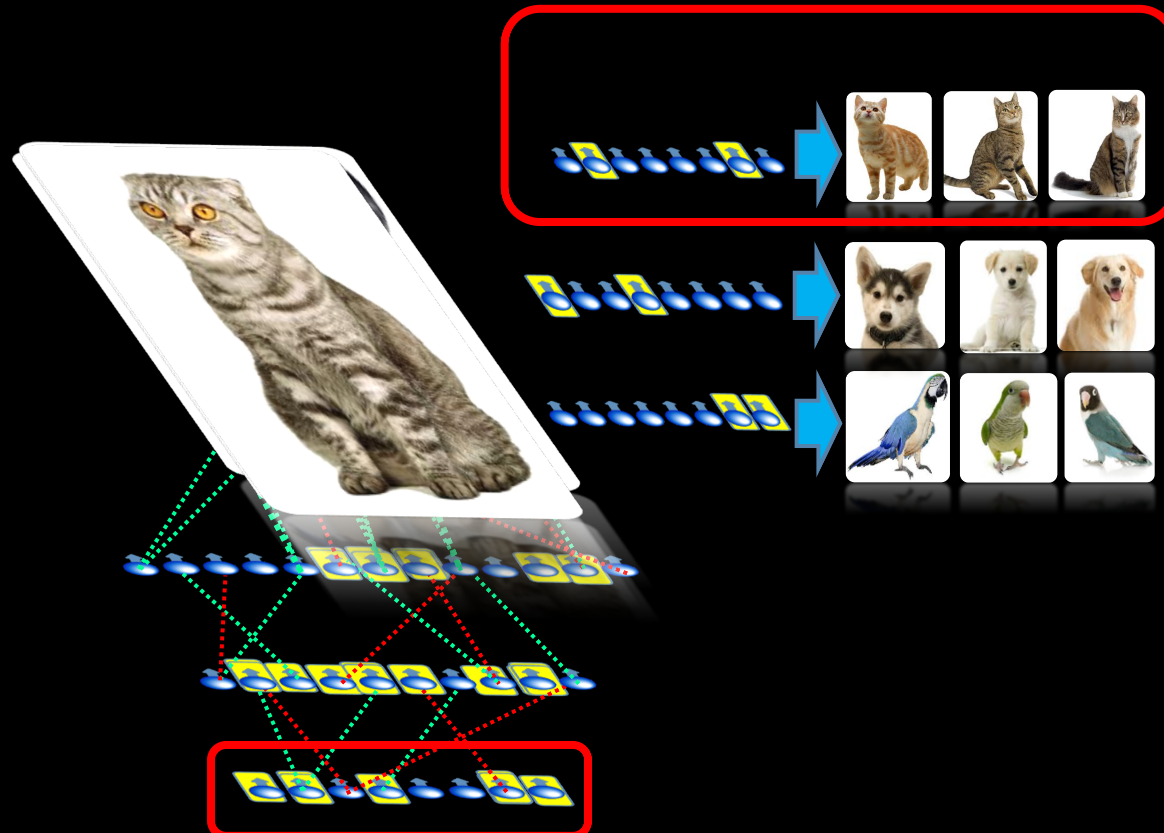
First we train a Deep Belief Network on millions of images to compute its weights...

Then we pump all of our images through our network to compute the deepest layer activations for each...



And we store the association between each activation pattern and its image...

Later, when presented with a new image, we can then identify similar matches!



And Here's the Kicker

Notice that we literally "pumped in" raw input data into our Restricted Boltzmann Machines!

They figured out the rest!

Most traditional Machine Learning approaches require the engineer to do extensive manual feature extraction…

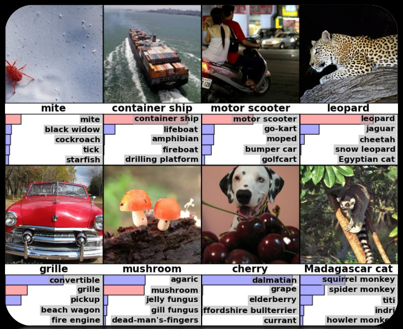
prior to applying Machine Learning on the input.

Deep Learning: Recent Wins

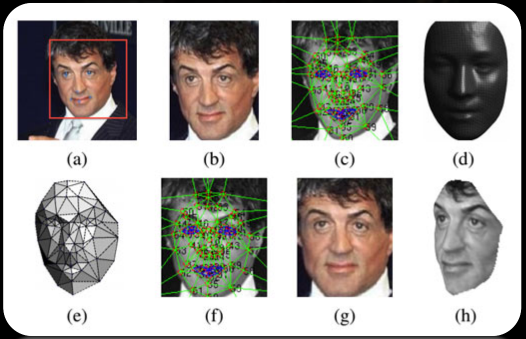
Baidu's Deep Speech project uses Recurrent Neural Networks and massive amounts of speech data (100x) to achieve a 29% improvement in speech recognition in 2 weeks!

System	Clean (94)	Noisy (82)	Combined (176)
Apple Dictation	14.24	43.76	26.73
Bing Speech	11.73	36.12	22.05
Google API	6.64	30.47	16.72
wit.ai	7.94	35.06	19.41
Deep Speech	**6.56**	**19.06**	**11.85**

Microsoft Research recently published image recognition results of 4.94% (top-5 error rate) on the 2012 ImageNet corpus… besting the established human error rate of 5.1%!



Facebook's DeepFace is capable of facial recognition rates of 97.25% - roughly .28% less than human-level accuracy!



Deep Learning: The Catalysts

2006 – Geoffrey Hinton discovers Contrastive Divergence and greedy layer-wise pre-training.

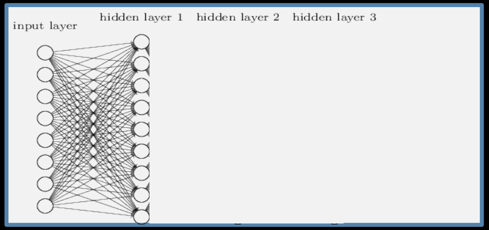
LETTER — Communicated by Yann Le Cun

A Fast Learning Algorithm for Deep Belief Nets

Geoffrey E. Hinton
hinton@cs.toronto.edu
Simon Osindero
osindero@cs.toronto.edu
Department of Computer Science, University of Toronto, Toronto, Canada M5S 3G4

Yee-Whye Teh
tehyw@comp.nus.edu.sg
Department of Computer Science, National University of Singapore, Singapore 117543

We show how to use "complementary priors" to eliminate the explaining-away effects that make inference difficult in densely connected belief nets that have many hidden layers. Using complementary priors, we derive a fast, greedy algorithm that can learn deep, directed belief networks one layer at a time, provided the top two layers form an undirected associative memory. The fast, greedy algorithm is used to initialize a slower learning procedure that fine-tunes the weights using a contrastive version of the wake-sleep algorithm. After fine-tuning, a network with three hidden layers forms a very good generative model of the joint distribution of handwritten digit images and their labels. This generative model gives better digit classification than the best discriminative learning algorithms. The low-dimensional manifolds on which the digits lie are modeled by long ravines in the free-energy landscape of the top-level associative memory, and it is easy to explore these ravines by using the directed connections to display what the associative memory has in mind.



Huge increases in computing power from GPUs and cloud computing.

The Internet yields orders of magnitude more data to train on.

A series of advances in machine learning algorithms: weight initialization, regularization, new activation functions

One Last Thought…

Have you ever thought about how amazing your visual memory is?

Do you think your brain literally stores away high-resolution snapshots of everything it sees?

Well maybe not!

This could explain why…

Given a memory cue you can suddenly remember a whole thought.

When viewing a partial image, your brain can visualize the rest…

When you see something unusual your brain knows it immediately…

After seeing hundreds of cats as a kid, but not knowing what they were called…

the moment your mom told you “that’s called a ‘cat’” you could instantly classify every cat you’d ever seen.

Conclusion

Deep Learning is a Machine Learning technique that works by learning multiple levels of abstraction about its inputs.

Once a Deep Learning system learns the inherent structure of its inputs, you can use it to…

classify items,

reconstruct partial inputs,

detect anomalies,

search for related items,

just daydream!

And finally, the engineer doesn't have to do lots of pre-processing to apply Deep Learning to many real-world problems!

Thank you！

As we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns—the ones we don't know we don't know.

Rumsfeld

Contact information:

Wu Xuening (邬学宁)
SAP硅谷创新中心 首席科学家
A:上海市浦东新区晨辉路1001号
E: x.wu@sap.com
T:021-61085287