The Deep Learning Revolution

Geoffrey Hinton

Google Brain Team
&
Vector Institute

Two paradigms for Artificial Intelligence

The logic-inspired approach

The essence of intelligence is using symbolic rules to manipulate symbolic expressions.

We should focus on reasoning.

The biologically-inspired approach

The essence of intelligence is learning the strengths of the connections in a neural network.

We should focus on learning and perception.

Two views of internal representations

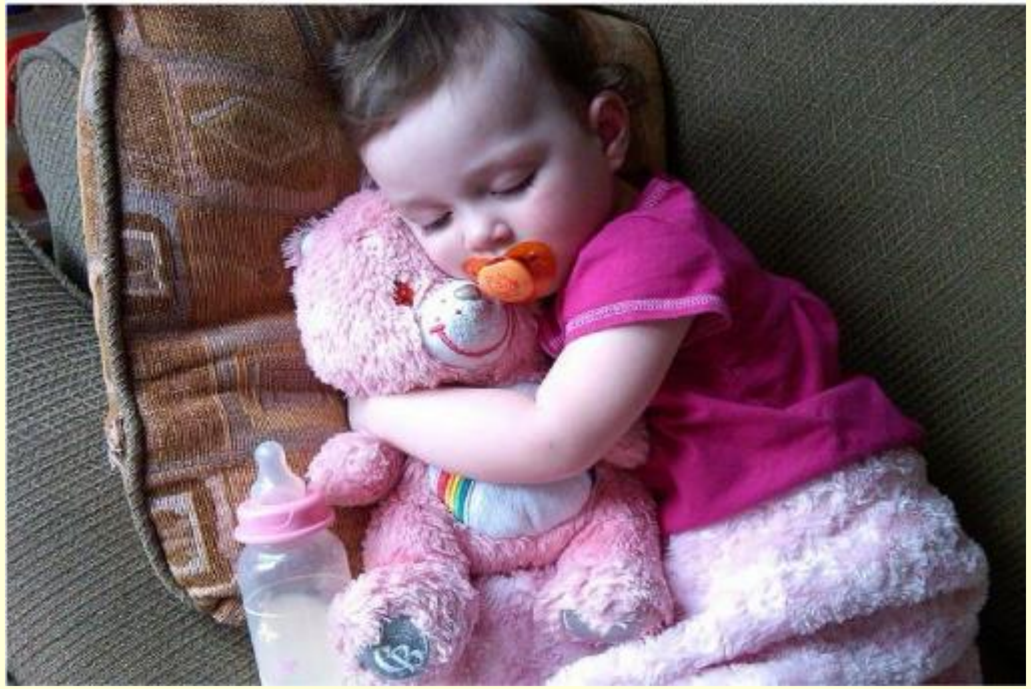
- Internal representations are symbolic expressions.
  - A programmer can give them to a computer using an unambiguous language.
  - New representations can be derived by applying rules to existing representations.

- Internal representations are nothing like language.
  - They are large vectors of neural activity.
  - They have direct causal effects on other vectors of neural activity.
  - These vectors are learned from data.

Two ways to make a computer do what you want

- Intelligent design: Figure out consciously exactly how you would manipulate symbolic representations to perform the task and then tell the computer, in excruciating detail, exactly what to do.

- Learning: Show the computer lots of examples of inputs together with the desired outputs. Let the computer learn how to map inputs to outputs using a general purpose, learning procedure.



A close-up of a child holding a stuffed animal.

Input is an image

Output is a caption

The central question

- Large neural networks containing millions of weights and many layers of non-linear neurons are very powerful computing devices.

- But can a neural network learn a difficult task (like object recognition or machine translation) by starting from random weights and acquiring all of its knowledge from the training data?

The obvious learning algorithm

- Early researchers like Turing and Selfridge proposed that neural networks with initially random connections could be trained by reinforcement learning.
  - This is extremely inefficient.

Perceptrons

- ~1960: Rosenblatt introduced a simple, efficient learning procedure that could figure out how to weight features of the input in order to classify inputs correctly.
  - But perceptrons could not learn the features.

- 1969: Minsky and Papert showed that perceptrons had some very strong limitations on what they could do.
  - Minsky and Papert also *implied* that having deeper networks would not help.

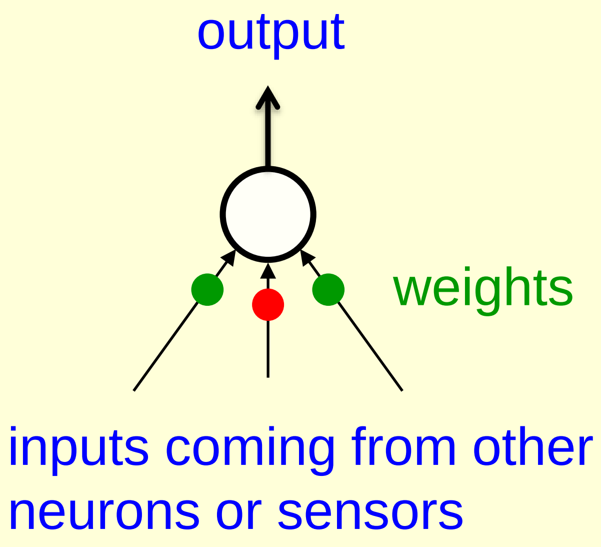
- 1970s: The first neural net winter

Back-propagation

- 1980s: The back-propagation procedure allows neural networks to design their own features and to have multiple layers of features.
  - Back-propagation created a lot of excitement.
  - It could learn vector embeddings that captured the meanings of words just by trying to predict the next word in a string.
  - It looked as if it would solve tough problems like speech recognition and shape recognition.

What is an artificial neuron?

- We make a gross idealization of a real neuron so that we can investigate how neurons can collaborate to do computations that are too difficult to program such as:
  - Convert the pixel intensity values of an image into a string of words that describe the image.

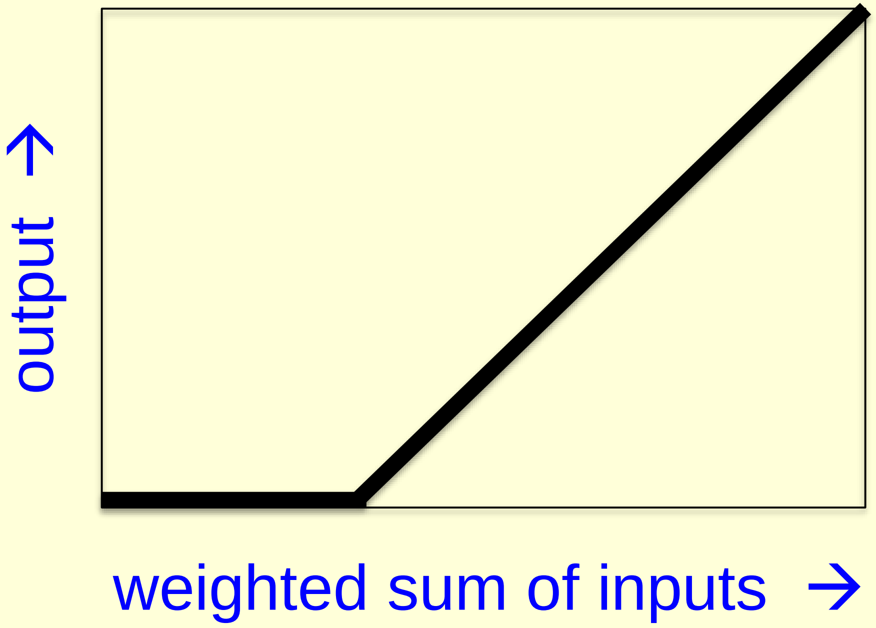


output

weights

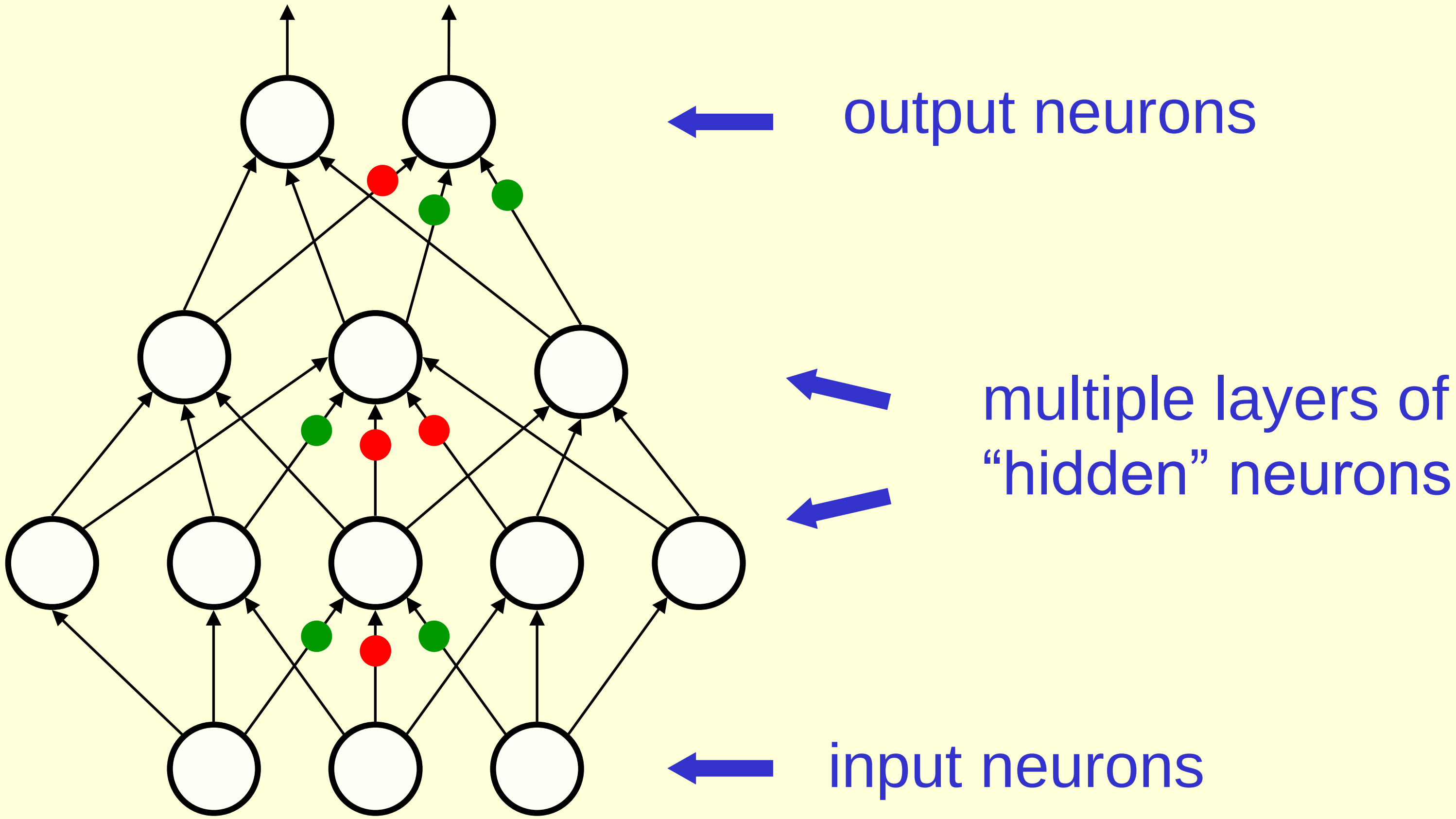
inputs coming from other neurons or sensors

A rectified linear neuron



What is an artificial neural network?

- If we connect the neurons in layers with no cycles we get a feed-forward neural net.



output neurons

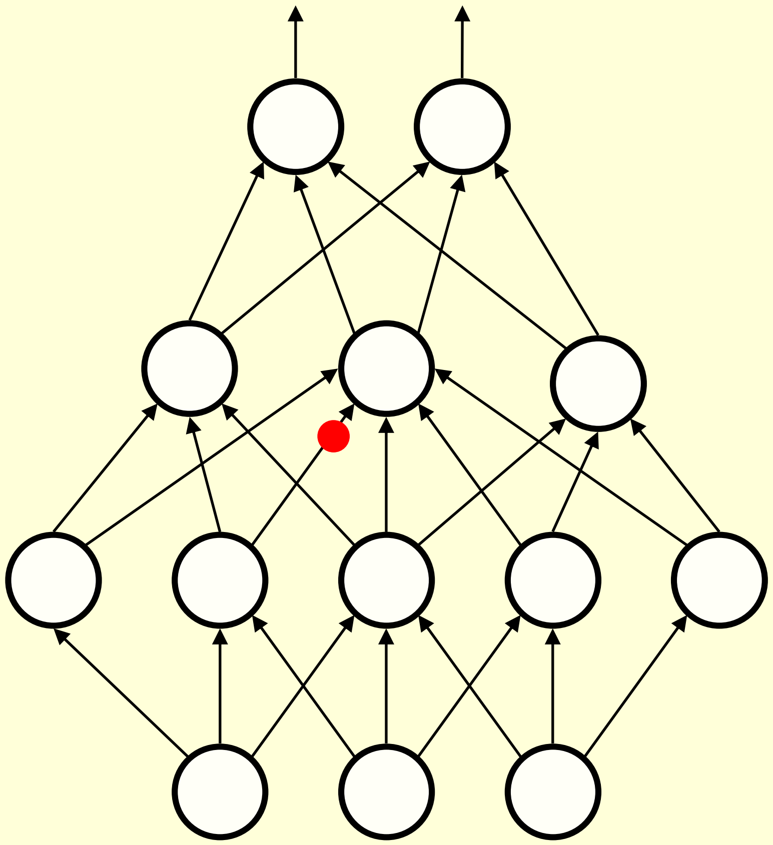
multiple layers of
“hidden” neurons

input neurons

How do we train artificial neural networks?

- Supervised training: Show the network an input vector and tell it the correct output.
  - Adjust the weights to reduce the discrepancy between the correct output and the actual output.
- Unsupervised training: Only show the network the input.
  - Adjust the weights to get better at reconstructing the input (or parts of the input) from the activities of the hidden neurons.

Supervised training: An inefficient “mutation” method that is easy to understand



- Take a small random sample of the training cases and measure how well the network does on this sample.
- Pick one of the weights.
- Increase or decrease the weight slightly and measure how well the network now does.
- If the change helped, keep it.

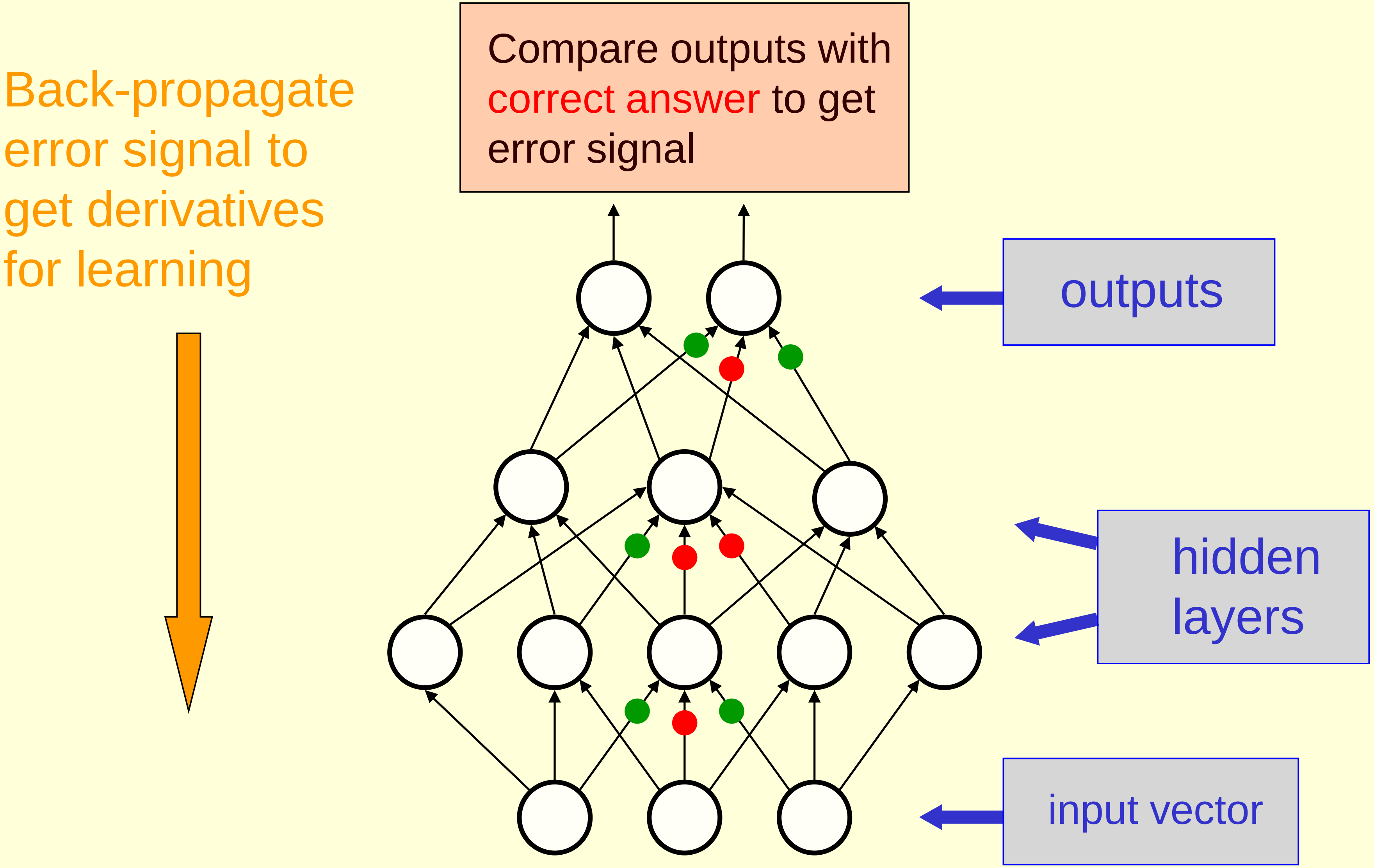
The backpropagation algorithm

- Backpropagation is just an efficient way of computing how a change in a weight will effect the output error.

- Instead of perturbing the weights one at a time and *measuring* the effect, use calculus to *compute* the error gradients for all of the weights at the same time. We can do this because we know how changing a weight will change the output.

  - With a million weights, this is more efficient than the mutation method by a factor of a million.

How to learn many layers of features (~1985)



Back-propagate error signal to get derivatives for learning

Compare outputs with correct answer to get error signal

outputs

hidden layers

input vector

Stochastic gradient descent

- The main discovery of neural nets so far is that dumb stochastic gradient descent (SGD) works much better than anyone expected.
  - You don’t need to get the gradient for a weight on the whole training set in order to make progress. A small “mini-batch” of data is sufficient.
  - Local optima are not a problem in practice.
  - Fancy second-order optimization methods are not needed.
    - Its sufficient to use momentum and to decrease the learning rate for weights that have large gradients.

A big disappointment

- 1990s: Backpropagation plus SGD works pretty well, but underperforms the expectations of its proponents.
  - It is hard to train deep neural networks. But why?

- On modest-sized datasets some other machine learning methods work better than backpropagation.
  - The second neural network winter begins
    (in the Machine Learning community)

- Symbolic AI researchers claim that it is silly to expect to learn difficult tasks in big deep neural nets that start with random connections and no prior knowledge.

Some really silly theories

- The continents used to be connected and drifted apart!
  (geologists spent 40 years laughing)
- Great big neural nets that start with random weights and no prior knowledge can learn to perform machine translation.
- The more you dilute a natural remedy, the more potent it gets.
  (this one really is silly)

“Further discussion of it merely incumbers the literature and befogs the mind of fellow students.”

- 2007: NIPS program committee rejects a paper on deep learning by *al. et.* Hinton because they already accepted a paper on deep learning and two papers on the same topic would be excessive.
- ~2009: A reviewer tells Yoshua Bengio that papers about neural nets have no place in ICML.
- ~2010: A CVPR reviewer rejects Yann LeCun’s paper even though it beats the state-of-the-art. The reviewer says that it tells us nothing about computer vision because everything is learned.

The return of backpropagation

- Between 2005 and 2009 researchers (in Canada!) made several technical advances that enabled backpropagation to work better in feed-forward nets.

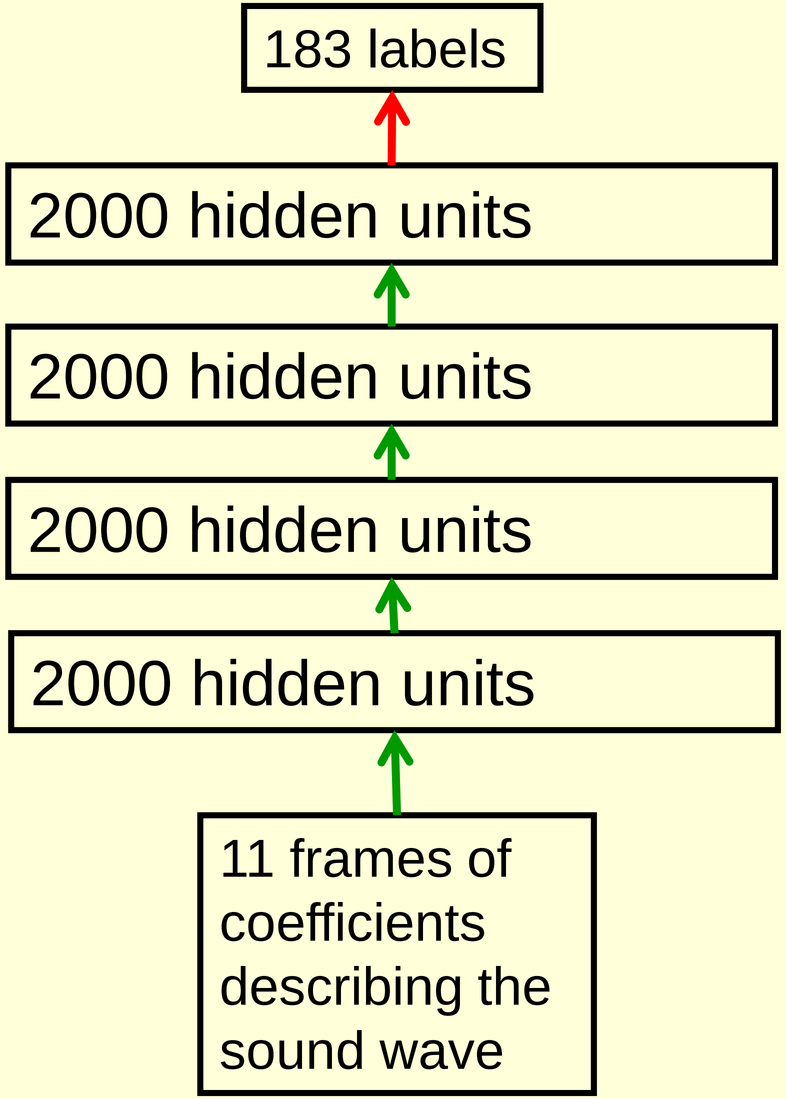
- The technical details of these advances are very important to the researchers but they are not the main message.

- The main message is that backpropagation now works amazingly well if you have two things:
  - a lot of labeled data
  - a lot of convenient compute power (*e.g.* GPUs)

Some of the technical tricks that made deep neural nets work better

- Unsupervised pre-training
  - Start by learning a layer of features that are good at reconstructing the input. Then learn a second layer that are good at reconstructing activities of the first layer etc.
  - After designing the features using unsupervised pre-training, use backpropagation to fine-tune them.
- Random dropout of units
  - Make each feature detector more robust by not allowing it to rely on other feature detectors to correct its mistakes.
- Rectified linear units
  - Rectified linear units are more powerful and easier to train than sigmoid units.

Acoustic modeling: The Killer App
(Mohamed, Dahl & Hinton 2009)



- After the standard post-processing this gets 23.0% phone error rate.
- The best previous result on TIMIT was 24.4%

What happened next

- As soon as Deep Neural Networks beat the previous technology, leading speech groups at MSR, IBM & Google developed them further.

- Navdeep Jaitly, a grad student from U of T, implemented our acoustic model at Google during an internship in 2011
  - By 2012, its was being used for voice search on the Android.
  - It gave a big decrease in the word error rate.

- Now the best speech recognition systems all use some form of neural net trained with backpropagation.

Object Recognition

- The 2012 ImageNet object recognition challenge has about a million high-resolution training images taken from the web.
  - There are 1000 different classes of object.
  - The task is to get the “correct” class in your top 5 bets.

- Some of the best existing computer vision methods were tried on this dataset by leading computer vision groups from all over the world.

Error rates on the ImageNet-2012 competition

2017 very deep neural nets (beats people!)	3%
University of Toronto (Krizhevsky *et al*, 2012)	16%
University of Tokyo	26%
Oxford University (Zisserman *et al*)	27%
INRIA (French national research institute in CS) + XRCE (Xerox Research Center Europe)	27%
University of Amsterdam	29%

A radically new way to do machine translation
(Suskever, Vinyals and Le, 2014)

- For each language we have an encoder neural network and a decoder neural network.

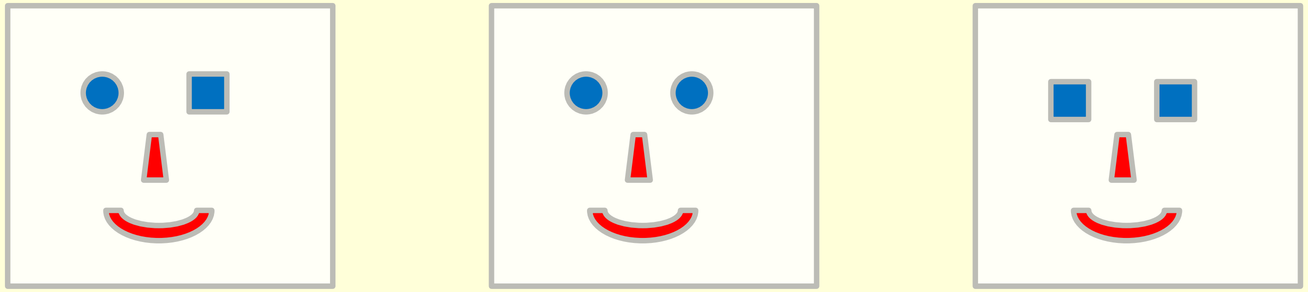
- The encoder reads in the sequence of words in the source sentence.
  - Its final hidden state represents the thought that the sentence expresses.
- The decoder expresses the thought in the target language.

Neural net machine translation

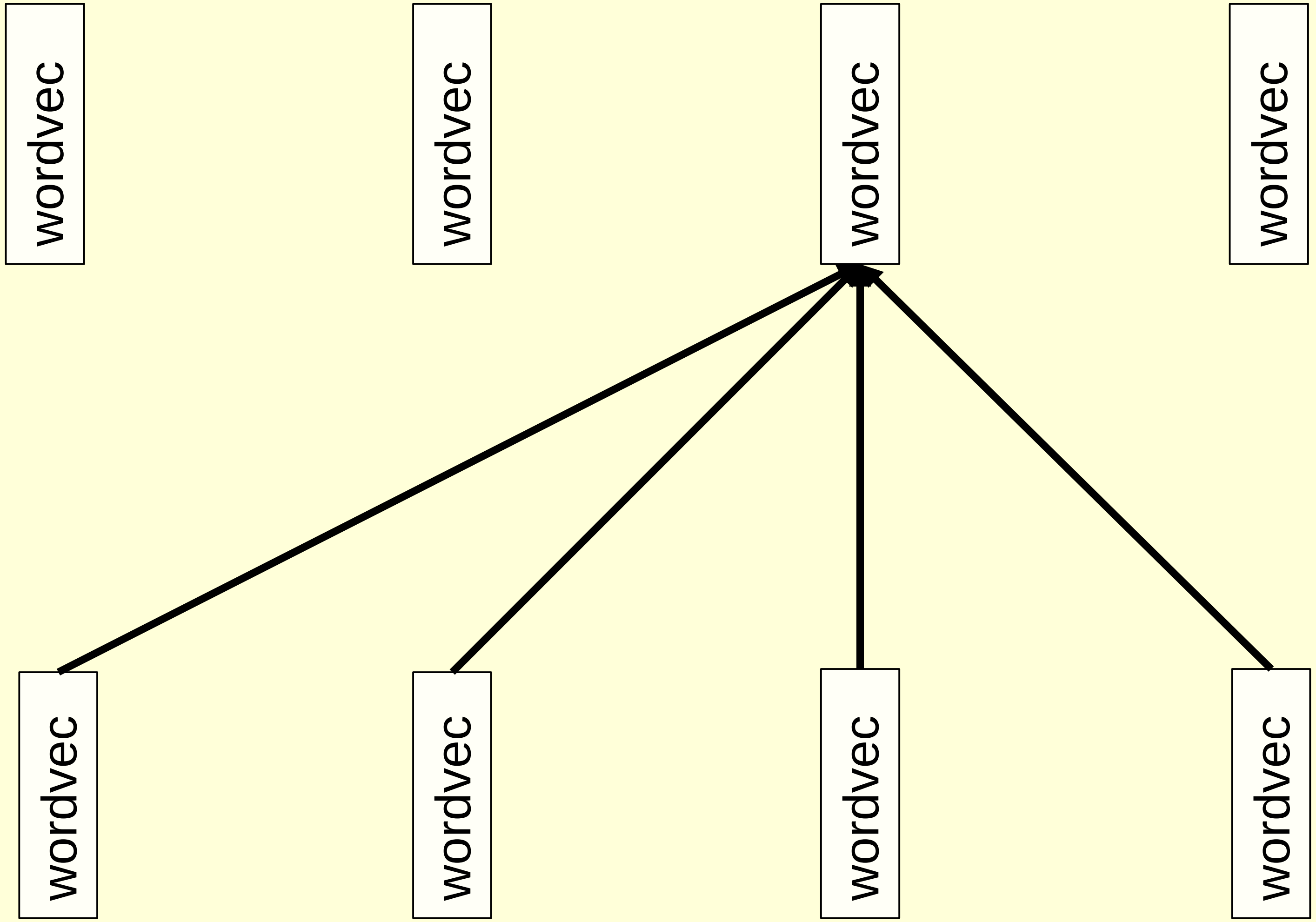
- It has evolved a lot since 2014.
  - Use soft attention to words in the source sentence when producing the target sentence.
  - Pre-train the word embeddings by trying to fill in the blanks using transformer networks. This unsupervised pre-training learns a lot of grammar.

Modeling covariances

- Standard neurons take the scalar product of a weight vector with an activity vector.
  - This is not good at modeling covariances between activities.

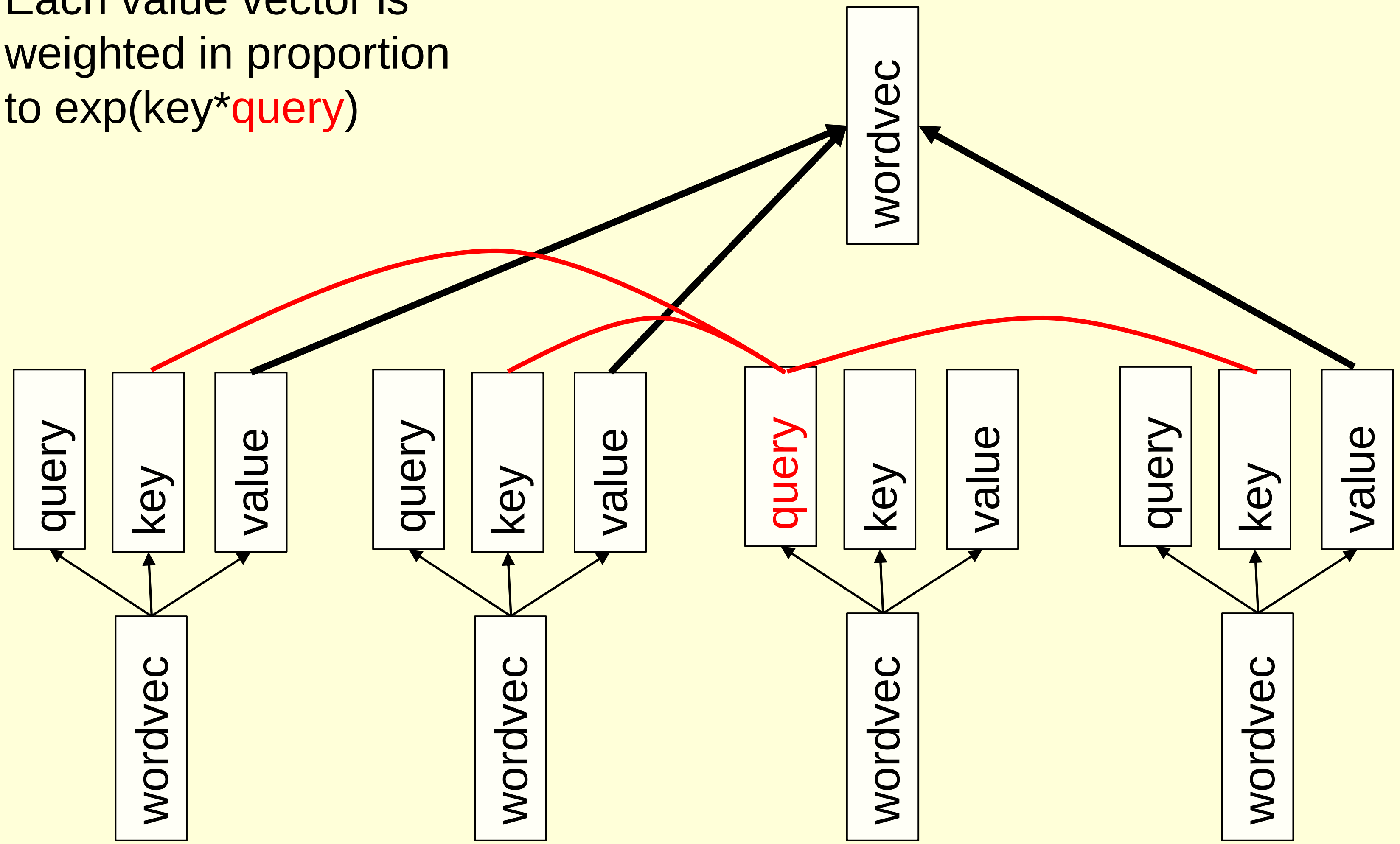


Standard convolutional neural network for refining word representations based on their context



How transformers work (roughly)

Each value vector is weighted in proportion to exp(key*query)



Neural net language modeling

- First pre-train word-fragment embeddings using transformers (BERT).

- Then, given a sequence of word fragments, train the net to predict the probability distribution over all possible next fragments.

- After training, see what it believes by giving it an initial word sequence and getting it to predict the probability distribution for the next fragment.
  - Pick from that distribution and tell it it guessed right.
  - Ask it to predict the next fragment.
  - Keep going until you have a story.

A huge neural net that predicts the next word fragment using a big temporal context

Initial text generated by hand:

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Continuation by Open AI's latest 1.5 billion weight neural network trained on a huge amount of text from the web:

The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Perez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes

Mountains when they found a small valley, with no other animals or humans. Perez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Perez and the others then ventured further into the valley. “By the time we reached the top of one peak, the water looked blue, with some crystals on top,” said Perez. Perez and his friends were astonished to see the unicorn herd.

These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns. While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Perez stated, “We can see, for example, that they have a common ‘language,’ something like a dialect or dialectic.” Dr. Perez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Perez, “In South America, such incidents seem to be quite common.”

However, Perez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. “But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization,” said the scientist.

How humans reason

- Almost all of our reasoning is intuitive rather than logical.
  - Intuitive reasoning makes use of big feature vectors that make analogies obvious with no intermediate steps of reasoning.
  - Which is better:
    - All cats are male and all dogs are female
    - All cats are female and all dogs are male
- Logical reasoning can be used to improve intuitive reasoning in much the same way as Monte-Carlo rollout is used to improve the intuitions that AlphaGo has about what is a good move or what is a good position.

The lesson of neural net machine translation

- Is this the final nail in the coffin of symbolic AI?
  - Machine translation is an ideal task for symbolic AI because the input is symbols and the output is symbols.

- But to make it work we need vectors inside.

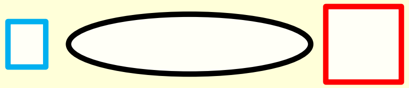
- The insights of symbolic AI researchers need to be used to design better architectures for neural nets.
  - Let stochastic gradient descent make it all work.

The future of neural network vision

- Convolutional neural nets get a huge win by wiring in the idea that if a feature is useful in one location it is useful everywhere.
  - But they do not recognize objects the same way as us, hence adversarial examples.
- People impose coordinate frames and recognize objects by using the viewpoint invariant geometrical relationships between the the coordinate frame of an object and the coordinate frames of its parts.
  - We can make neural networks do this by using transformer networks: arxiv.org/abs/1906.06818

Capsules 2019

- A capsule is a group of neurons that learns to represent a familiar shape fragment.
  - It has a logistic unit that represents whether the fragment exists in the current image.
  - It has a matrix that represents its "pose" *i.e.* the geometrical relationship between the fragment and the camera.
  - It represents other properties such as deformation, velocity color etc.

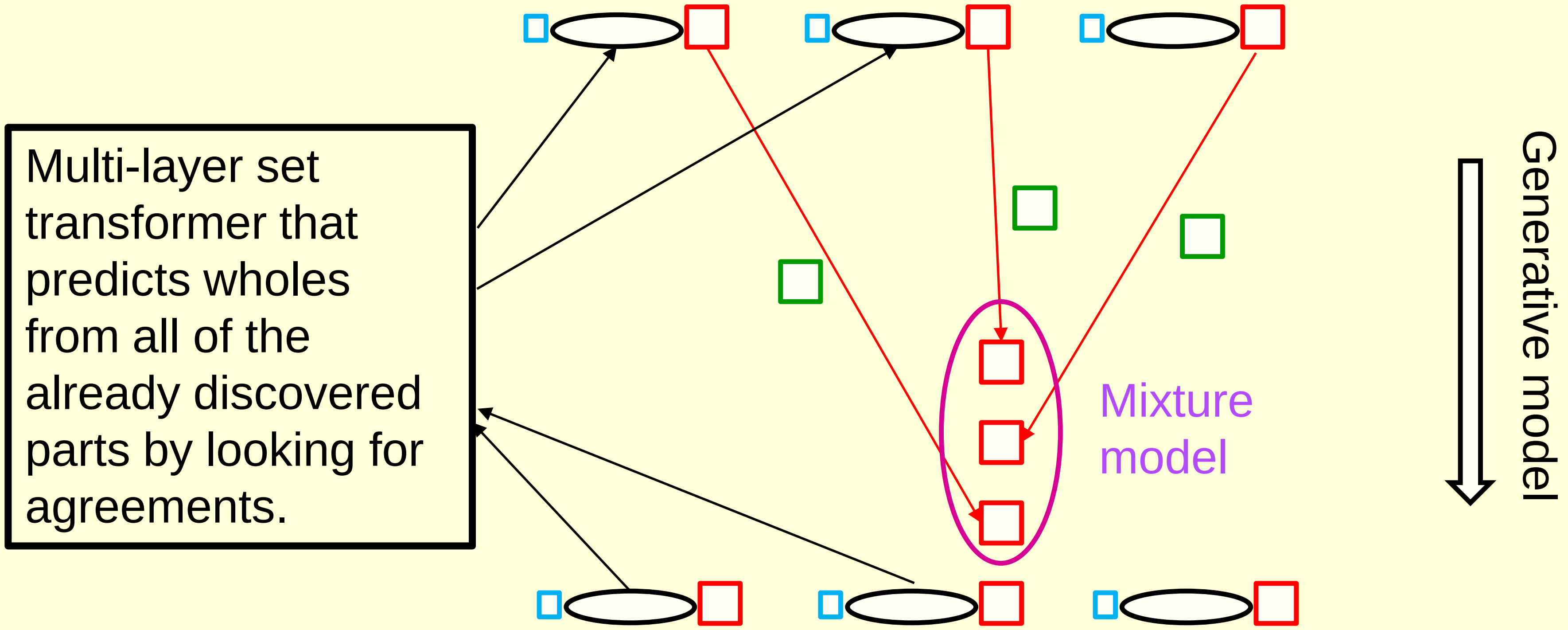


Capturing intrinsic geometry

- A capsule that represents an object can predict the poses of the parts of the object.
  - The pose of a part is the pose of the object times a matrix that represents the coordinate transform between the whole and the part.
    - This matrix does not depend on viewpoint so it is a statistically efficient way to represent shape.

A capsule auto-encoder

Red matrices are viewpoint equivariant activities.
Green matrices are viewpoint invariant weights.



Multi-layer set transformer that predicts wholes from all of the already discovered parts by looking for agreements.

The matrix multiplies deal with the effects of viewpoint.
The mixture models implement the single-parent constraint.

Summary

- Stochastic gradient descent can learn a billion weights.

- Unsupervised learning is making a comeback.

- We don’t know what primitive function should be computed by “neurons” or groups of neurons.

- Understanding the nature of the computations we want to perform should help us design better neural nets.
  - But leave the hard work to stochastic gradient descent.

THE END

The future of neural networks

- Nearly all artificial neural nets use only two time scales: Slow adaptation of weights and fast changes in neural activity.

- But synapses adapt at multiple different time scales.
  - Using fast weights for short-term memory will make neural networks different and better.
  - It can improve optimization.
  - It allows true recursion (1973, unpublished)