



# Bonus Lecture: Deep Learning





# A brief history

- 1943: neural networks  $\Leftrightarrow$  logical circuits (McCulloch/Pitts)
- 1949: "cells that fire together wire together" learning rule (Hebb)
- 1969: theoretical limitations of neural networks (Minsky/Papert)
- 1974: backpropagation for training multi-layer networks (Werbos)
- 1986: popularization of backpropagation (Rumelhardt, Hinton, Williams)



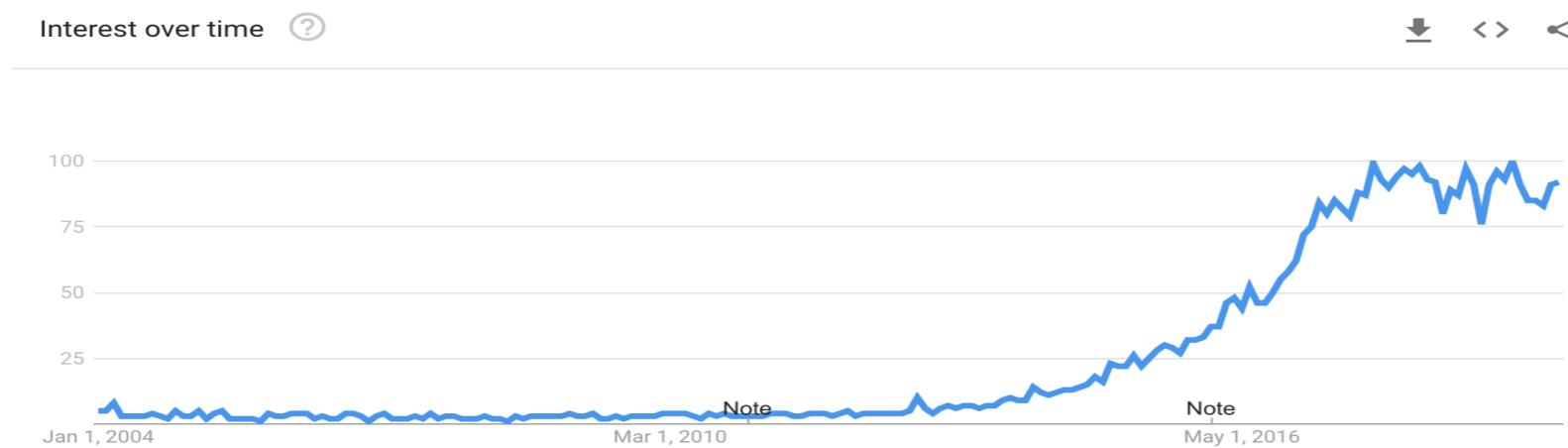
# A brief history

- 1980: Neocognitron, a.k.a. convolutional neural networks (Fukushima)
- 1989: backpropagation on convolutional neural networks (LeCun)
- 1990: recurrent neural networks (Elman)
- 1997: Long Short-Term Memory networks (Hochreiter/Schmidhuber)
- 2006: unsupervised layerwise training of deep networks (Hinton et al.)



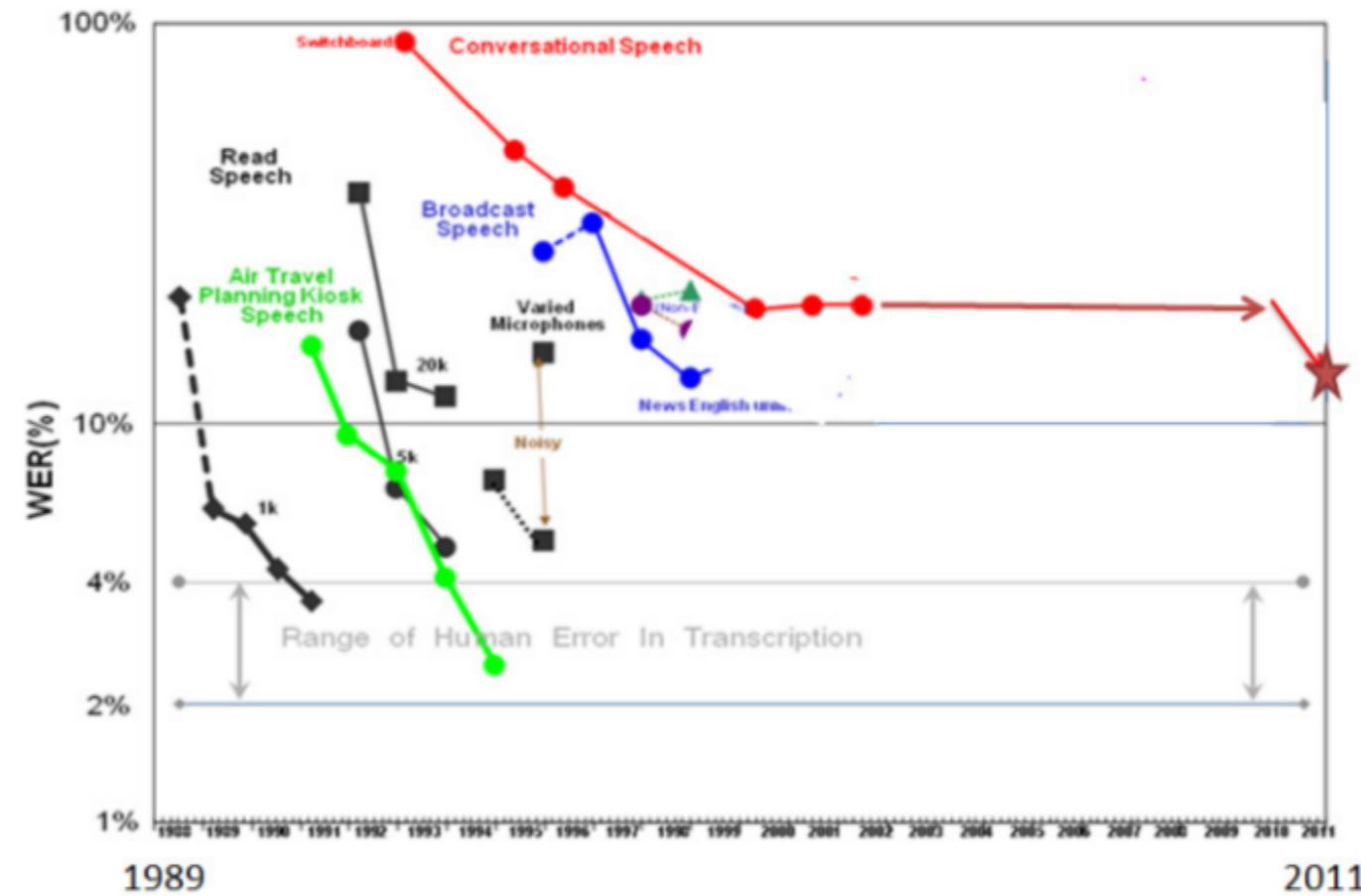
# Google Trends

Query: deep learning





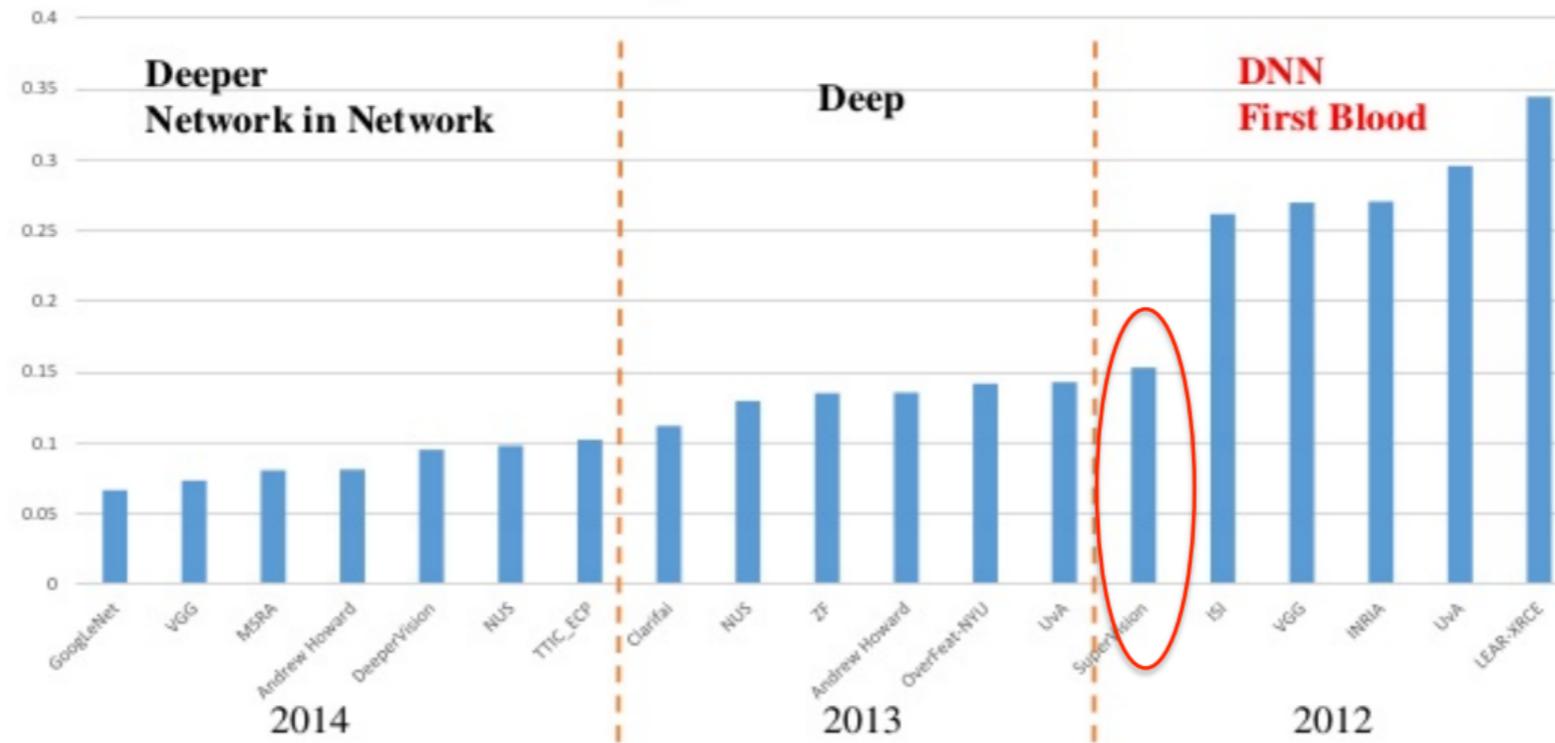
# Speech recognition (2009-2011)



- Steep drop in WER due to deep learning
- IBM, Google, Microsoft all switched over from GMM-HMM



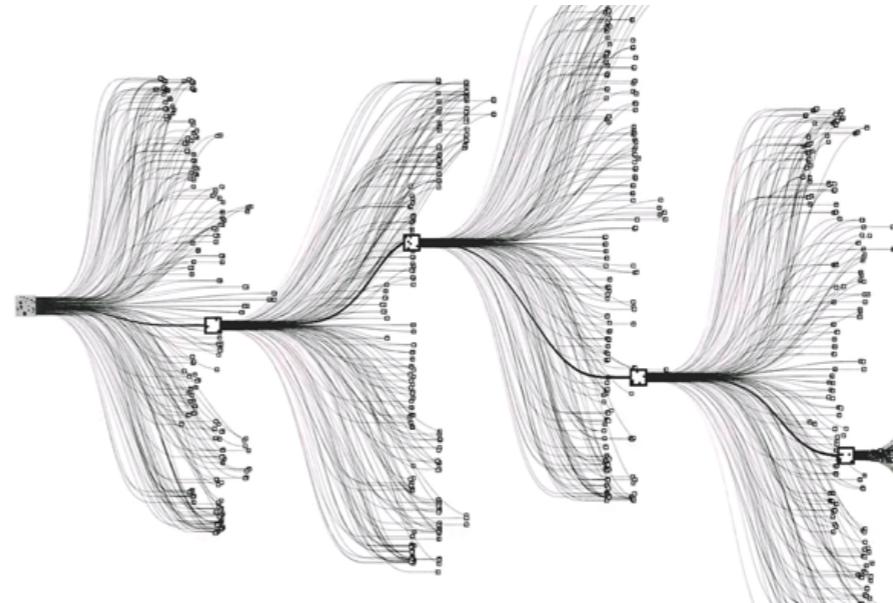
# Object recognition (2012)



- Landslide win in ILSVRC object recognition competition
- Computer vision community switched to CNNs
- Simpler than hand-engineered features (SIFT)



# Go (2016)

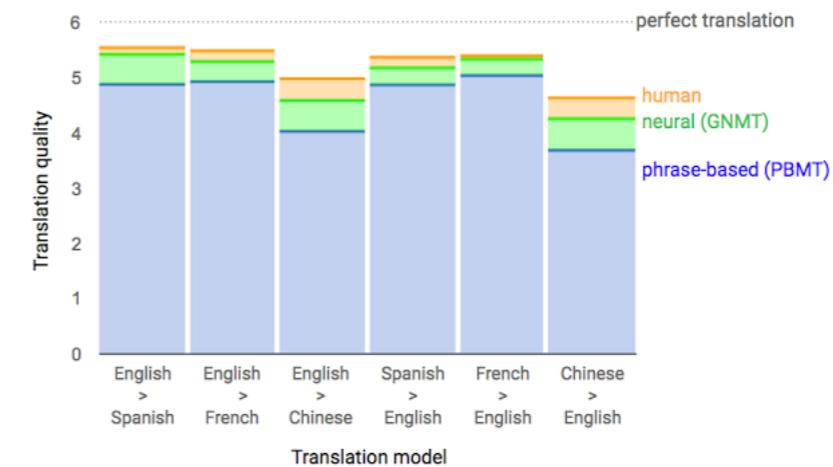


- Defeated world champion Le Sedol 4-1
- Simple architecture (in contrast, DeepBlue was search + hand-crafted heuristics)
- 2017: AlphaGoZero does not require human expert games as supervision



# Machine translation (2016)

<i>Input sentence:</i>	<i>Translation (PBMT):</i>	<i>Translation (GNMT):</i>	<i>Translation (human):</i>
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

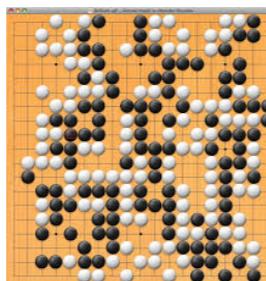
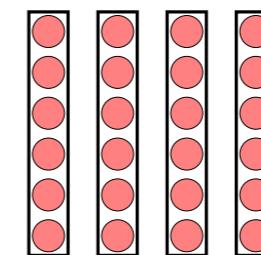


- Decisive wins have taken longer to achieve in NLP (words are meaningful in a way that pixels are not)
- Current state-of-the-art in machine translation
- Simpler architecture (throw out word alignment, phrases tables, language models)



# What is deep learning?

*A family of techniques for learning compositional vector representations of complex data.*







# Roadmap

**Feedforward neural networks**

Convolutional neural networks

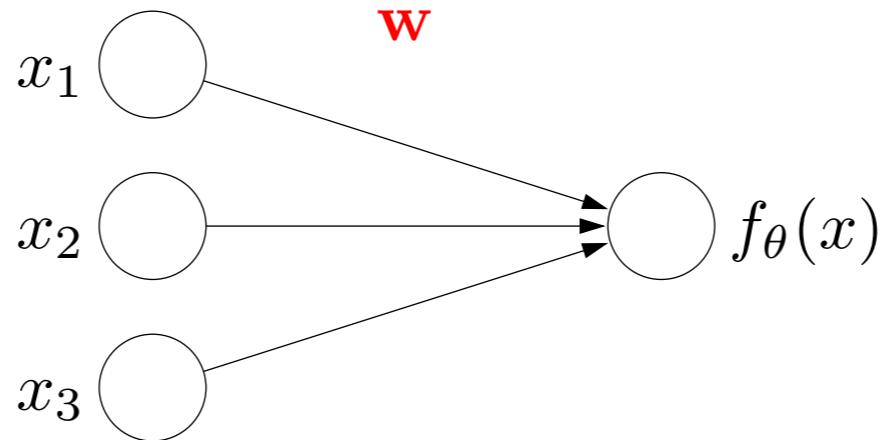
Recurrent neural networks

Unsupervised learning

Final remarks



# Review: linear predictors

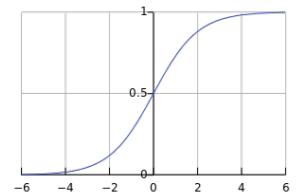


Output:

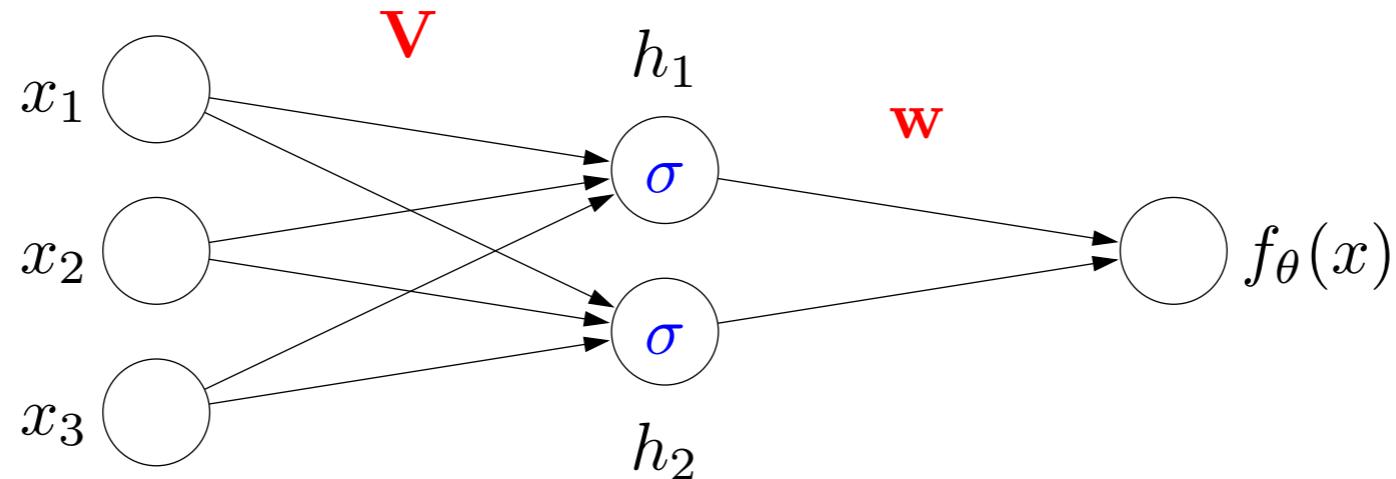
$$f_{\theta}(x) = \mathbf{w} \cdot x$$

Parameters:  $\theta = \mathbf{w}$





# Review: neural networks



Intermediate hidden units:

$$h_j(x) = \sigma(\mathbf{v}_j \cdot x) \quad \sigma(z) = (1 + e^{-z})^{-1}$$

Output:

$$f_\theta(x) = \mathbf{w} \cdot \mathbf{h}(x)$$

Parameters:  $\theta = (\mathbf{V}, \mathbf{w})$



## Deep neural networks

1-layer neural network:

$$\text{score} = \mathbf{w}^\top \mathbf{x}$$

2-layer neural network:

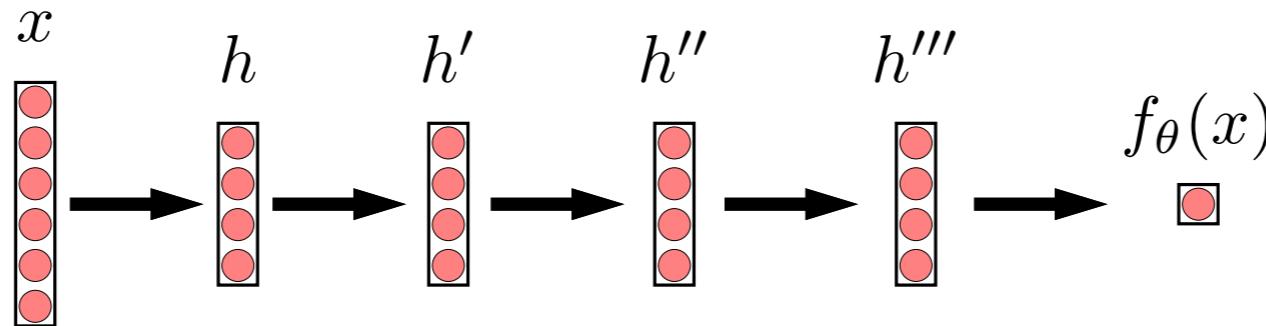
$$\text{score} = \mathbf{w}^\top \mathbf{V} \sigma(\mathbf{U}^\top \mathbf{x})$$

3-layer neural network:

$$\text{score} = \mathbf{w}^\top \sigma(\mathbf{U}^\top \sigma(\mathbf{V}^\top \mathbf{x}))$$



# Depth

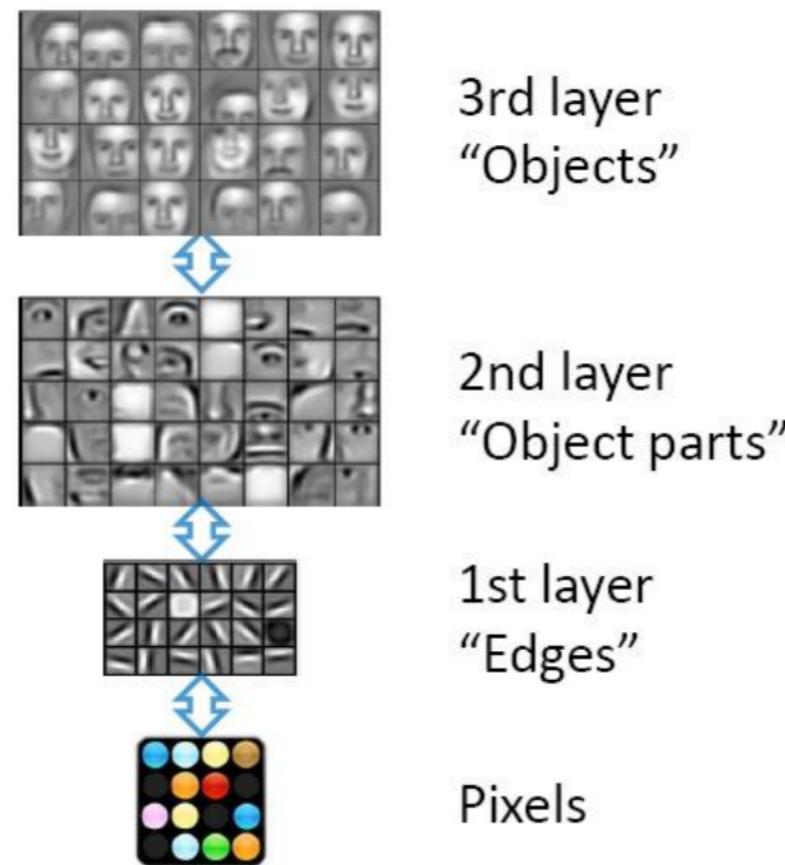


## Intuitions:

- Hierarchical feature representations
- Can simulate a bounded computation logic circuit (original motivation from McCulloch/Pitts, 1943)
- Learn this computation (and potentially more because networks are real-valued)
- Depth  $k + 1$  logic circuits can represent more than depth  $k$  (counting argument)
- Formal theory/understanding is still incomplete



# What's learned?







# Summary

- Deep networks learn hierarchical representations of data
- Train via SGD, use backpropagation to compute gradients
- Non-convex optimization, but works empirically given enough compute and data



# Review: optimization

Regression:

$$\text{Loss}(x, y, \theta) = (f_\theta(x) - y)^2$$



**Key idea: minimize training loss**

$$\text{TrainLoss}(\theta) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \text{Loss}(x, y, \theta)$$

$$\min_{\theta \in \mathbb{R}^d} \text{TrainLoss}(\theta)$$



**Algorithm: stochastic gradient descent**

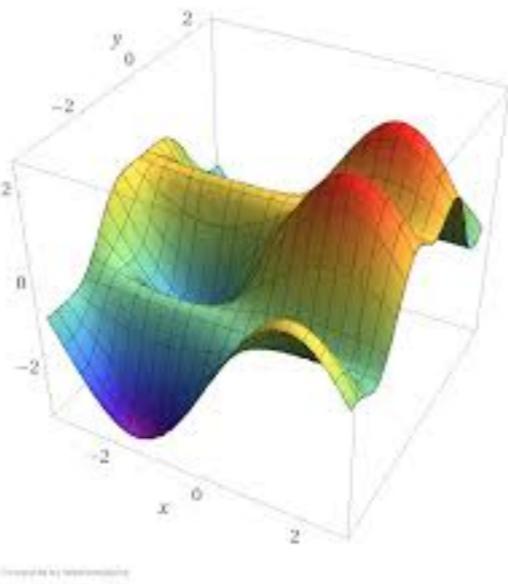
For  $t = 1, \dots, T$ :

    For  $(x, y) \in \mathcal{D}_{\text{train}}$ :

$$\theta \leftarrow \theta - \eta_t \nabla_{\theta} \text{Loss}(x, y, \theta)$$



# Training

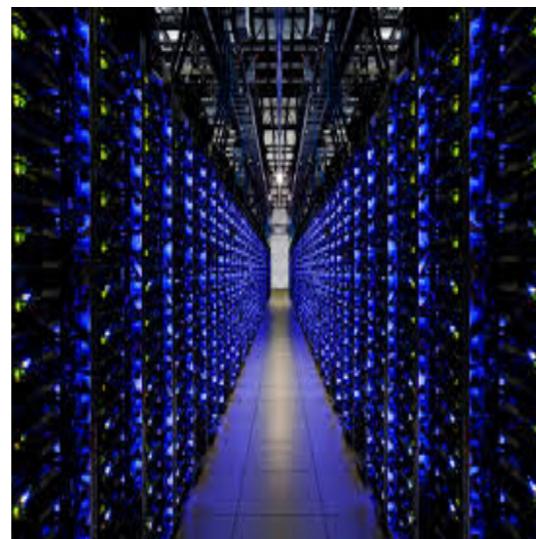


- Non-convex optimization
- No theoretical guarantees that it works
- Before 2000s, empirically very difficult to get working



# What's different today

Computation (time/memory)

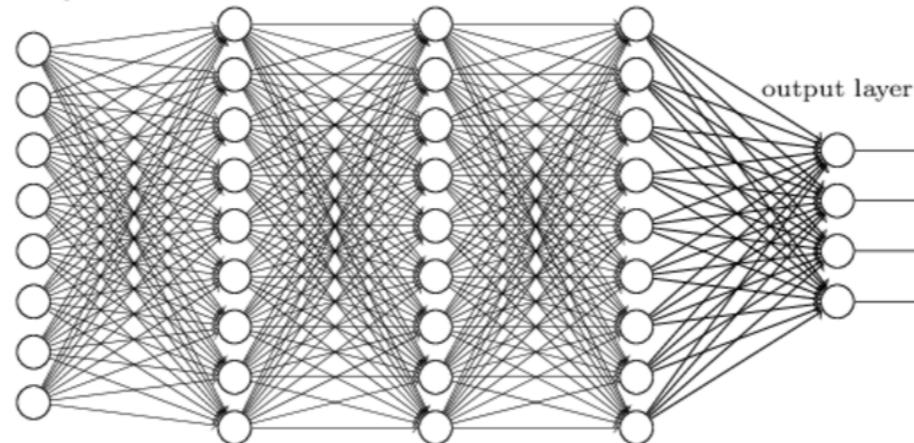


Information (data)





# How to make it work



- More hidden units (over-provisioning)
- Adaptive step sizes (AdaGrad, ADAM)
- Dropout to guard against overfitting
- Careful initialization (pre-training)
- Batch normalization
- Model and optimization are tightly coupled





# Roadmap

Feedforward neural networks

**Convolutional neural networks**

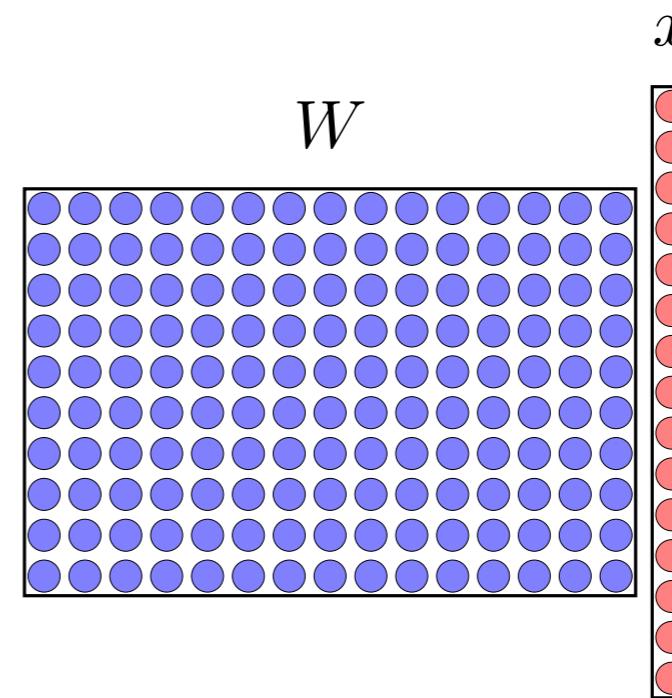
Recurrent neural networks

Unsupervised learning

Final remarks



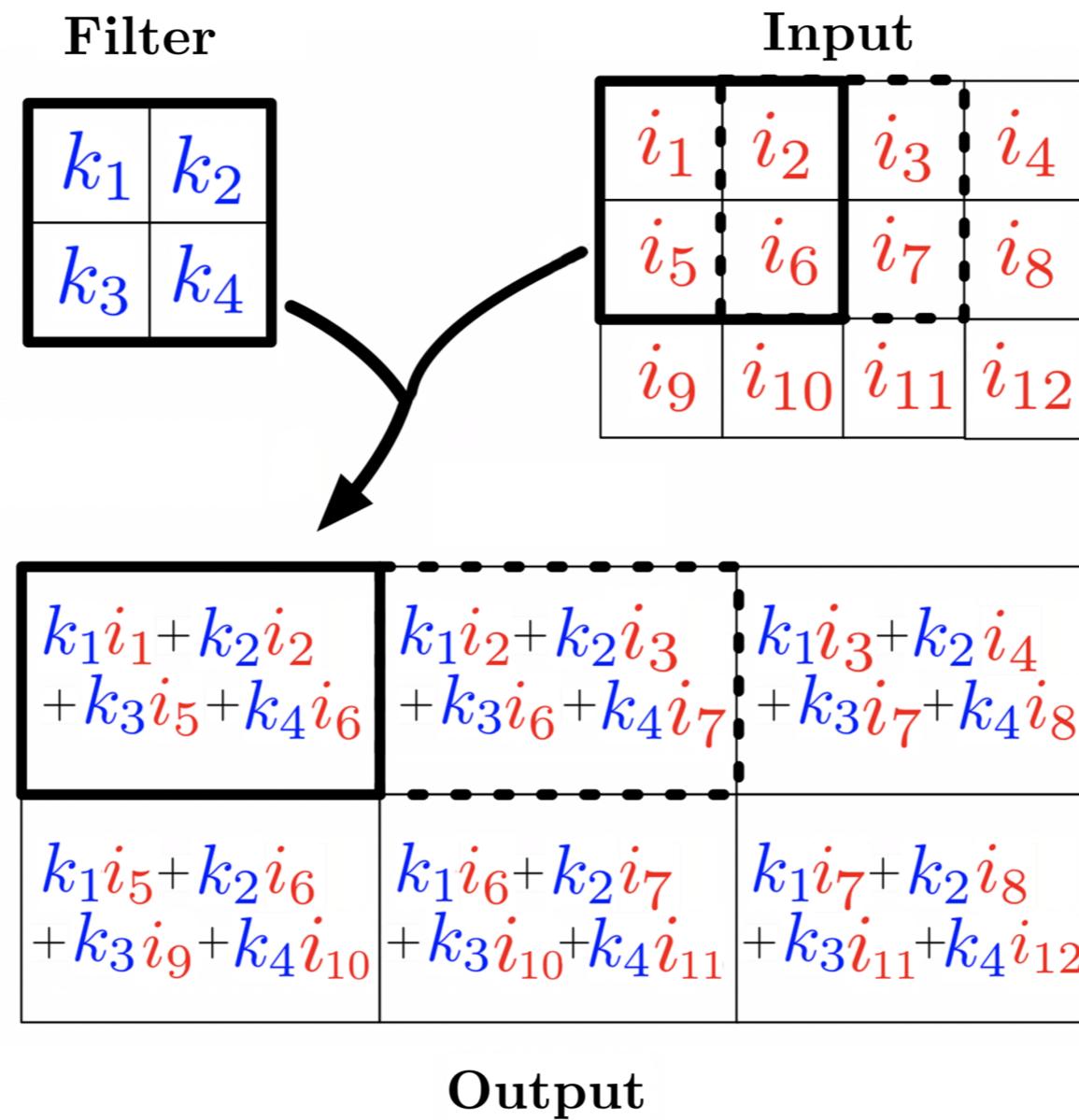
# Motivation



- **Observation:** images are not arbitrary vectors
- **Goal:** leverage spatial structure of images (translation invariance)

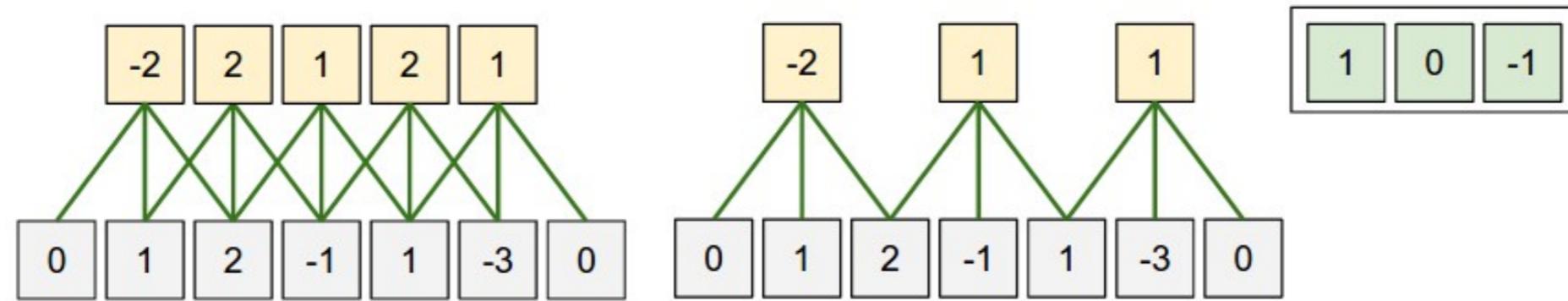


# Idea: Convolutions





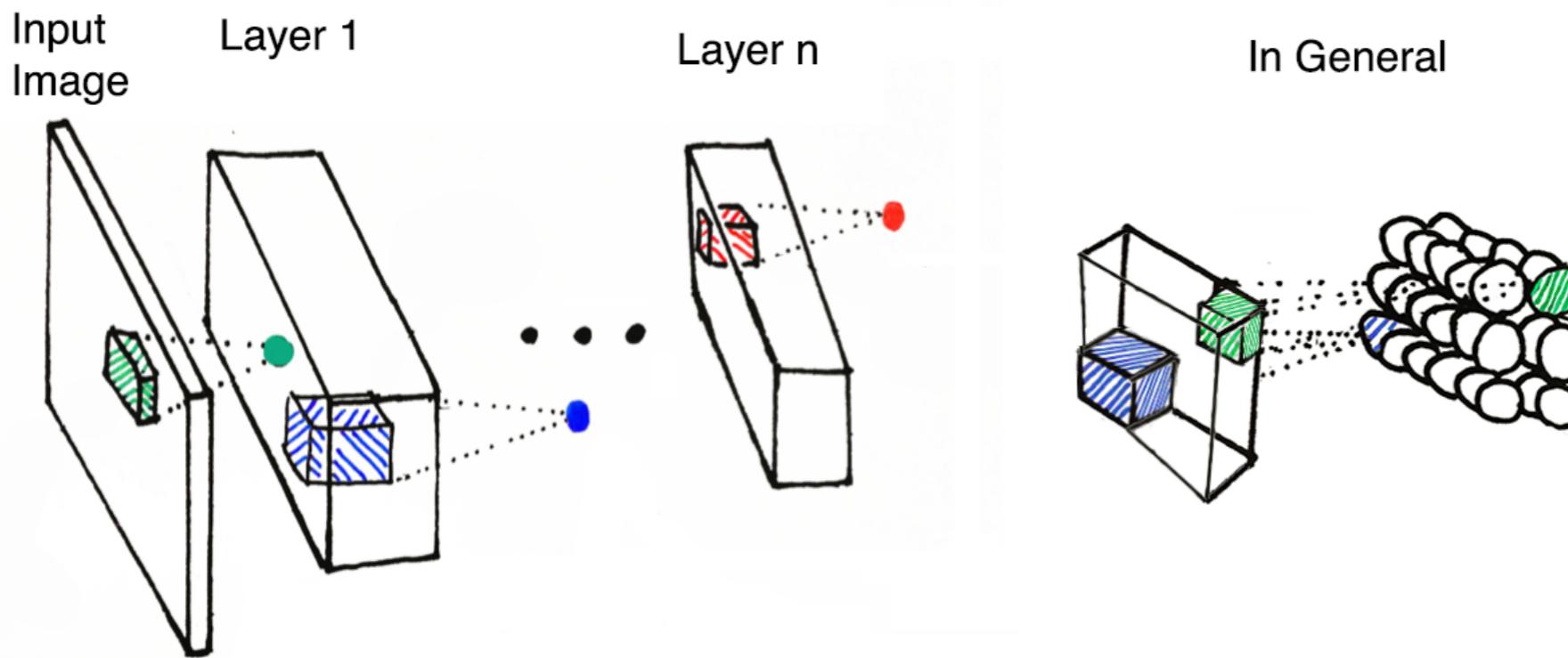
# Prior knowledge



- **Local connectivity:** each hidden unit operates on a local image patch (3 instead of 7 connections per hidden unit)
- **Parameter sharing:** processing of each image patch is same (3 parameters instead of  $3 \cdot 5$ )
- **Intuition:** try to match a pattern in image



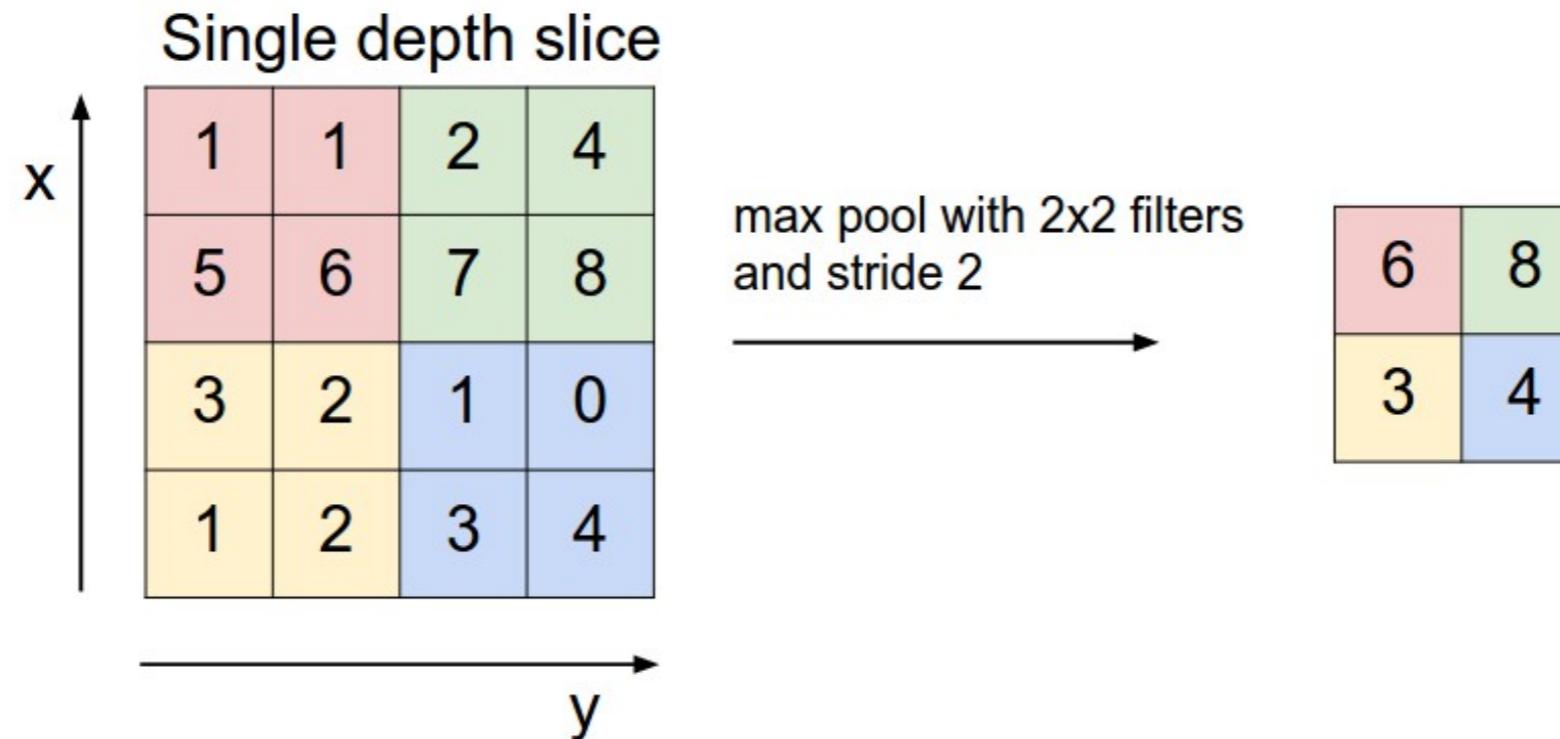
# Convolutional layers



- Instead of vector to vector, we do volume to volume



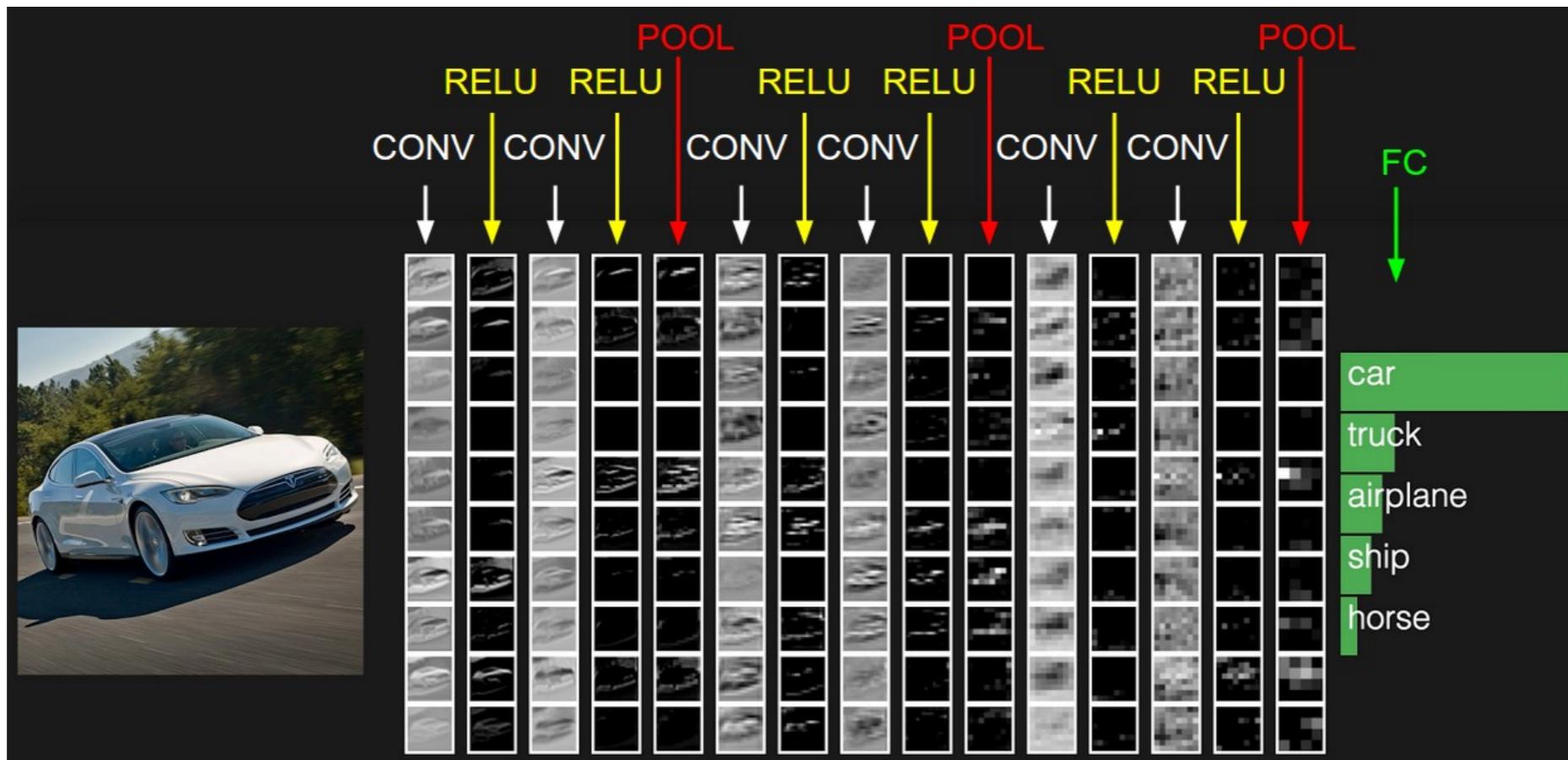
# Max-pooling



- Intuition: test if there exists a pattern in neighborhood
- Reduce computation, prevent overfitting

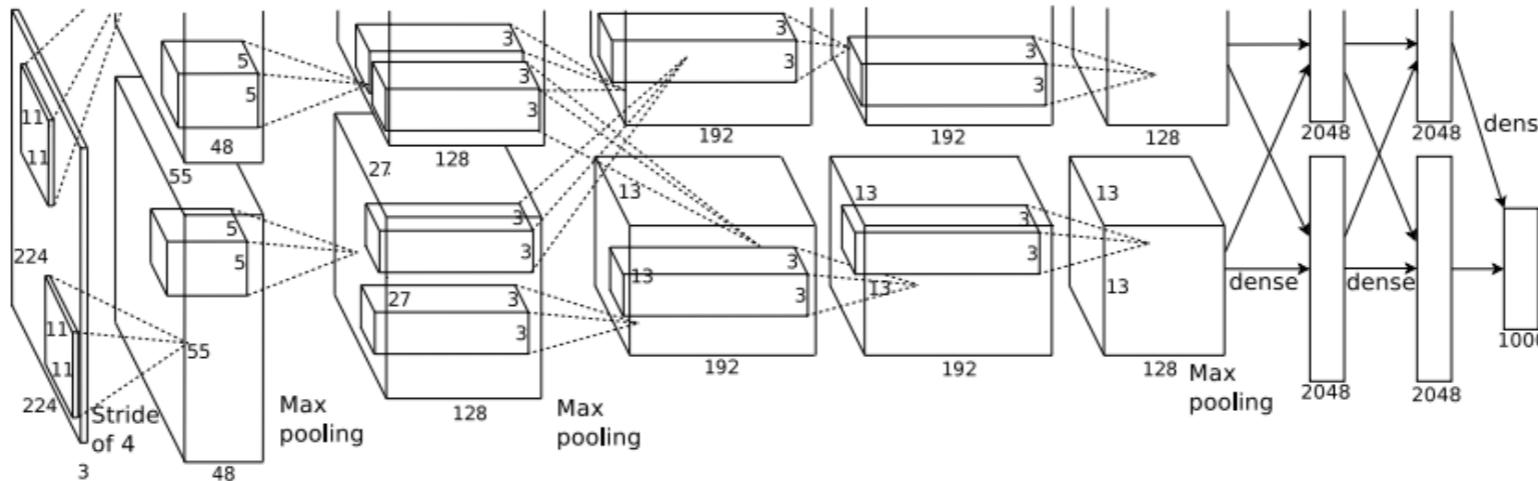


# Example of function evaluation





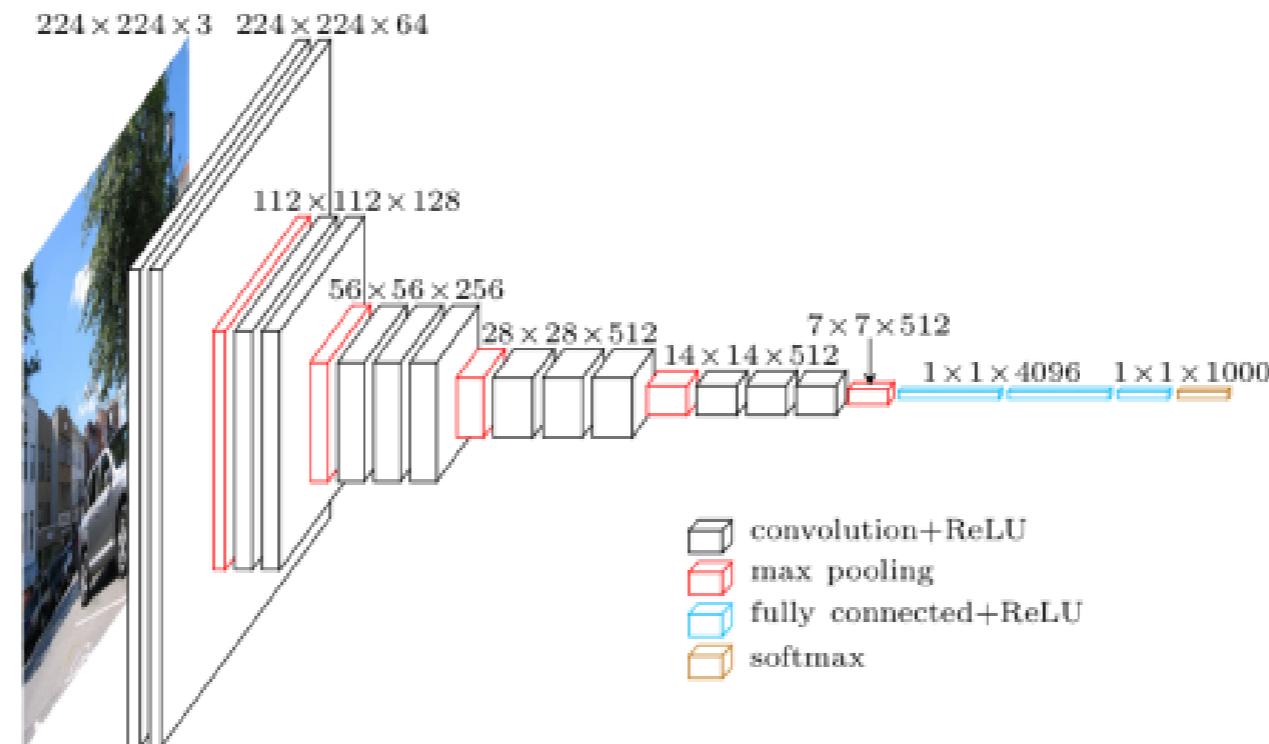
# AlexNet



- **Non-linearity:** use ReLU ( $\max(z, 0)$ ) instead of logistic
- **Data augmentation:** translate, horizontal reflection, vary intensity, dropout (guard against overfitting)
- **Computation:** parallelize across two GPUs (6 days)
- **Results on ImageNet:** 16.4% error (next best was 25.8%)



# VGGNet

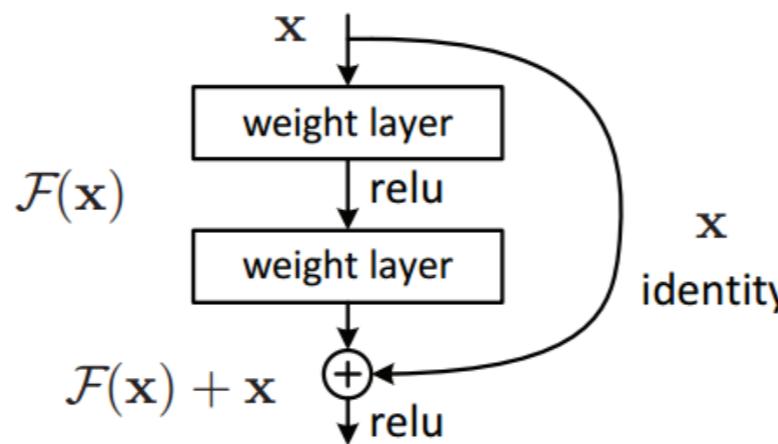


- **Architecture:** deeper but smaller filters; uniform
- **Computation:** 4 GPUs for 2-3 weeks
- **Results on ImageNet:** 7.3% error (AlexNet: 16.4%)

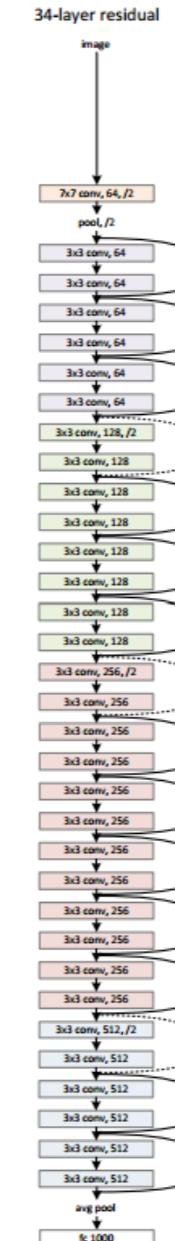


# Residual networks

$$x \mapsto \sigma(Wx) + x$$



- Key idea: make it easy to learn the identity (good inductive bias)
- Enables training 152 layer networks
- Results on ImageNet: 3.6% error







# Summary

- Key idea: locality of connections, capture spatial structure
- Filters have parameter sharing; most parameters in last fully connected layers
- Depth really matters
- Applications to text, Go, drug design, etc.





# Roadmap

Feedforward neural networks

Convolutional neural networks

**Recurrent neural networks**

Unsupervised learning

Final remarks

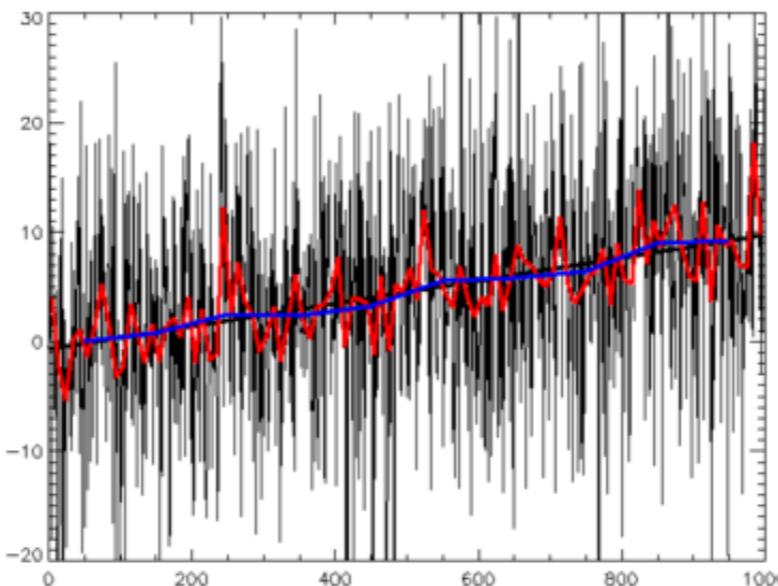


# Motivation: modeling sequences

Sentences:

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$
<i>Paris</i>	<i>Talks</i>	<i>Set</i>	<i>Stage</i>	<i>for</i>	<i>Action</i>	<i>as</i>	<i>Risks</i>	<i>to</i>	<i>the</i>	<i>Climate</i>	<i>Rise</i>

Time series:



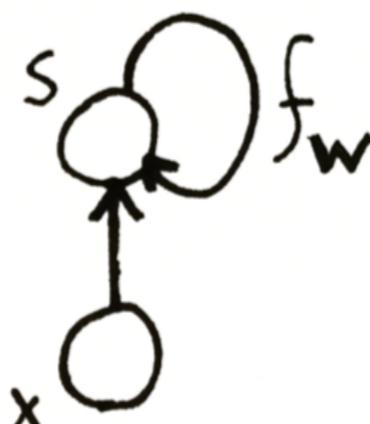


# Recurrent neural networks

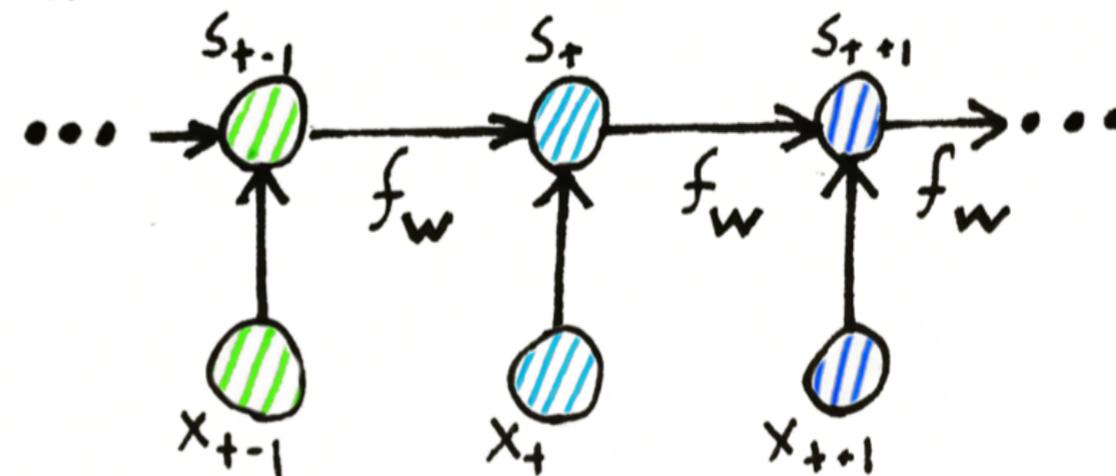
Formula

$$s_t = f_W(s_{t-1}, x_t)$$

Network

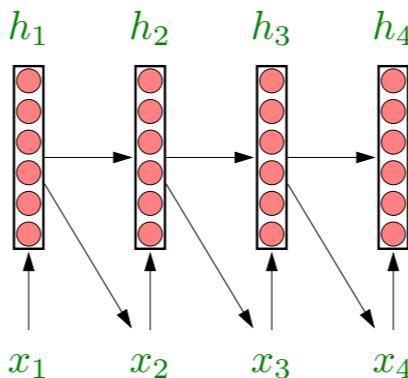


Computation  
Graph





# Recurrent neural networks



$h_1 = \text{Encode}(x_1)$

$x_2 \sim \text{Decode}(h_1)$

$h_2 = \text{Encode}(h_1, x_2)$

$x_3 \sim \text{Decode}(h_2)$

$h_3 = \text{Encode}(h_2, x_3)$

$x_4 \sim \text{Decode}(h_3)$

$h_4 = \text{Encode}(h_3, x_4)$

Update context vector:

$h_t = \text{Encode}(h_{t-1}, x_t)$

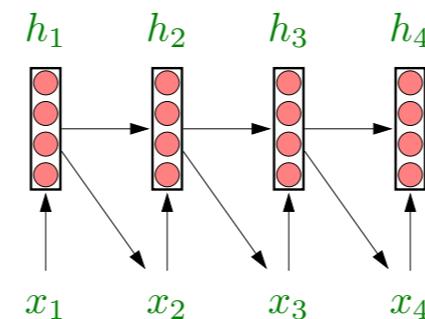
Predict next character:

$x_{t+1} = \text{Decode}(h_t)$

context  $h_t$  compresses  $x_1, \dots, x_t$



# Simple recurrent network



$$\text{Encode}(h_{t-1}, x_t) = \sigma(V h_{t-1} + W x_t) = h_t$$

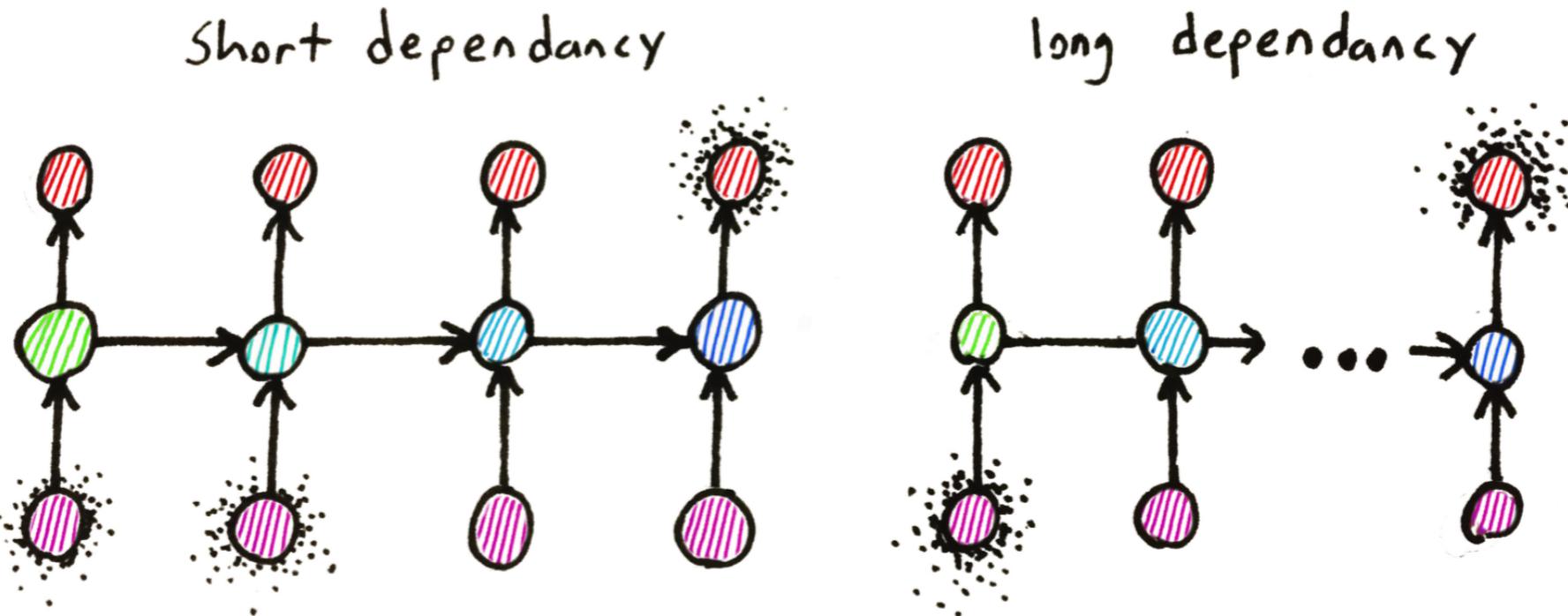
Diagram illustrating the forward pass of a simple recurrent network. The hidden state  $h_{t-1}$  (represented by a vertical column of four red circles) is multiplied by weight matrix  $V$  (represented by a 4x4 grid of blue circles). The input  $x_t$  (represented by a vertical column of four red circles) is multiplied by weight matrix  $W$  (represented by a 4x4 grid of blue circles). The results are summed and passed through a sigmoid activation function  $\sigma$  to produce the hidden state  $h_t$  (represented by a vertical column of four red circles).

$$\text{Decode}(h_t) \sim \text{softmax}(W' h_t) = p(x_{t+1})$$

Diagram illustrating the backward pass of a simple recurrent network. The hidden state  $h_t$  (represented by a vertical column of four red circles) is multiplied by weight matrix  $W'$  (represented by a 4x4 grid of blue circles). The result is passed through a softmax function to produce the probability distribution  $p(x_{t+1})$  (represented by a vertical column of four red circles).



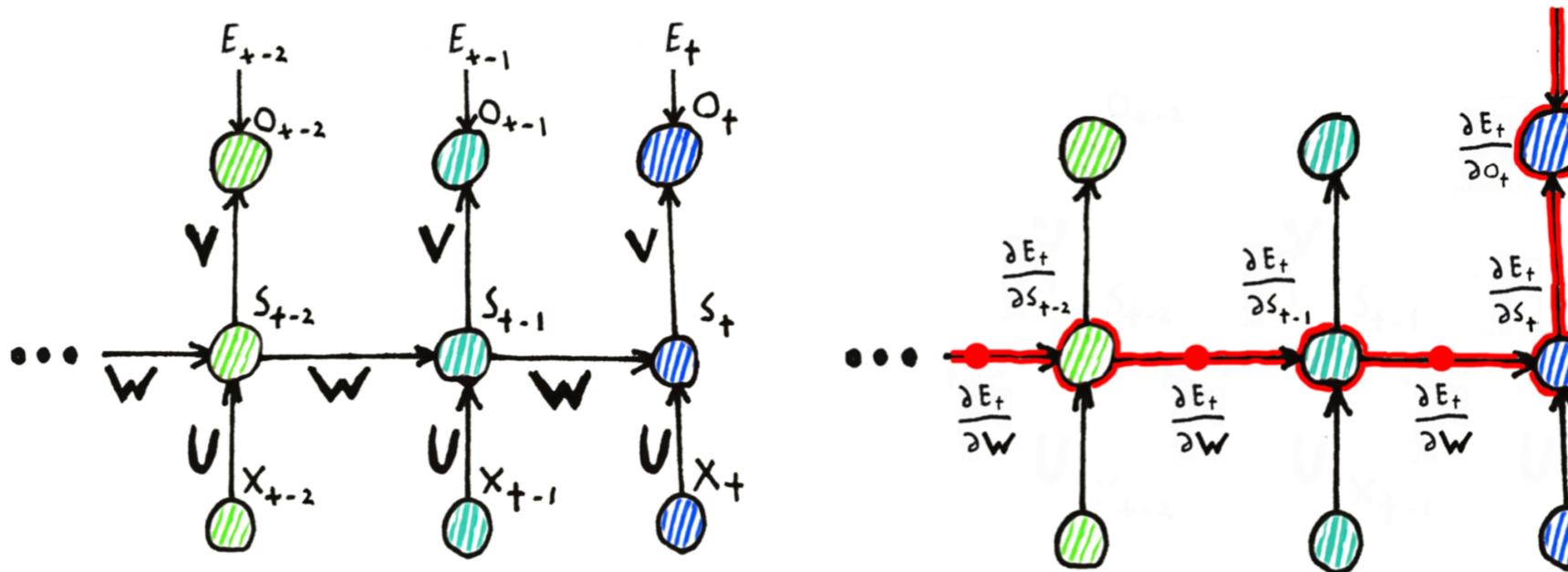
# Vanishing gradient problem



- RNNs can have long or short dependancies
- When there are long dependancies, gradients have trouble backpropagating through



# Vanishing gradient problem



Chain rule => multiplications

Can explode or shrink!

$$\frac{\partial E_t}{\partial \mathbf{W}} = \sum_{k=0}^t \frac{\partial E_t}{\partial o_t} \frac{\partial o_t}{\partial s_t} \frac{\partial s_t}{\partial s_k} \frac{\partial s_k}{\partial \mathbf{W}}$$

$$\frac{\partial s_t}{\partial s_k} = \prod_{j=k+1}^t \frac{\partial s_j}{\partial s_{j-1}}$$



# Long Short Term Memory (LSTM)

API:

$$(h_t, c_t) = \text{LSTM}(h_{t-1}, c_{t-1}, x_t)$$

Input gate:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i)$$

Forget gate (initialize with  $b_f$  large, so close to 1):

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f)$$

Cell: additive combination of RNN update with previous cell

$$c_t = i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) + f_t \odot c_{t-1}$$

Output gate:

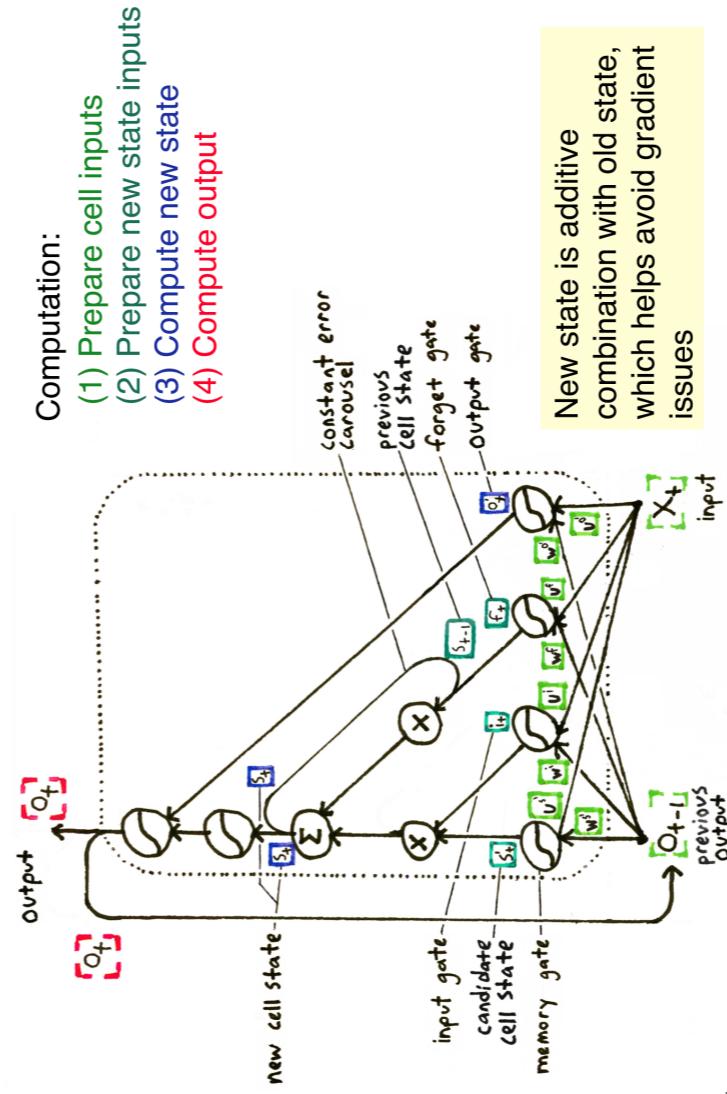
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t + b_o)$$

Hidden state:

$$h_t = o_t \odot \tanh(c_t)$$



## Long Short Term Memory (LSTM)





# Character-level language modeling

Sampled output:

*Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25—21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict.*



Cell sensitive to position in line:

```
The sole importance of the crossing of the Berezina lies in the fact
that it plainly and indubitably proved the fallacy of all the plans for
cutting off the enemy's retreat and the soundness of the only possible
line of action--the one Kutuzov and the general mass of the army
demanded--namely, simply to follow the enemy up. The French crowd fled
at a continually increasing speed and all its energy was directed to
reaching its goal. It fled like a wounded animal and it was impossible
to block its path. This was shown not so much by the arrangements it
made for crossing as by what took place at the bridges. When the bridges
broke down, unarmed soldiers, people from Moscow and women with children
who were with the French transport, all--carried on by vis inertiae--
pressed forward into boats and into the ice-covered water and did not,
surrender.
```

Cell that turns on inside quotes:

```
"You mean to imply that I have nothing to eat out of.... On the
contrary, I can supply you with everything even if you want to give
dinner parties," warmly replied Chichagov, who tried by every word he
spoke to prove his own rectitude and therefore imagined Kutuzov to be
animated by the same desire.
```

```
Kutuzov, shrugging his shoulders, replied with his subtle penetrating
smile: "I meant merely to say what I said."
```

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (!!(current->notifier)(current->notifier_data)) {
                    clear_thread_flag(TIF_SIGPENDING);
                    return 0;
                }
            }
            collect_signal(sig, pending, info);
        }
    }
    return sig;
}
```

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
    char *str;
    if (!*bufp || (len == 0) || (len > *remain))
        return ERR_PTR(-EINVAL);
    /* Of the currently implemented string fields, PATH_MAX
     * defines the longest valid length.
    */
```

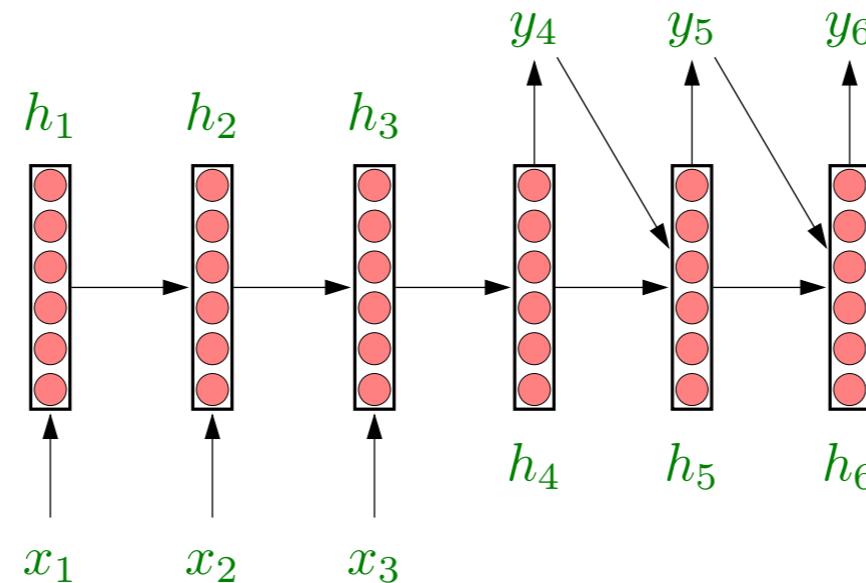


# Sequence-to-sequence model

Motivation: machine translation

$x: Je \ crains \ l'homme \ de \ un \ seul \ livre.$

$y: Fear \ the \ man \ of \ one \ book.$



Read in a sentence first, output according to RNN:

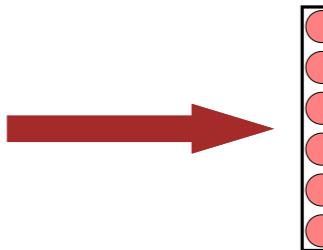
$$h_t = \text{Encode}(h_{t-1}, x_t \text{ or } y_{t-1}), \quad y_t = \text{Decode}(h_t)$$



# Attention-based models

Motivation: long sentences — compress to finite dimensional vector?

*Eine Folge von Ereignissen bewirkte, dass aus Beethovens Studienreise nach Wien ein dauerhafter und endgültiger Aufenthalt wurde. Kurz nach Beethovens Ankunft, am 18. Dezember 1792, starb sein Vater. 1794 besetzten französische Truppen das Rheinland, und der kurfürstliche Hof musste fliehen.*

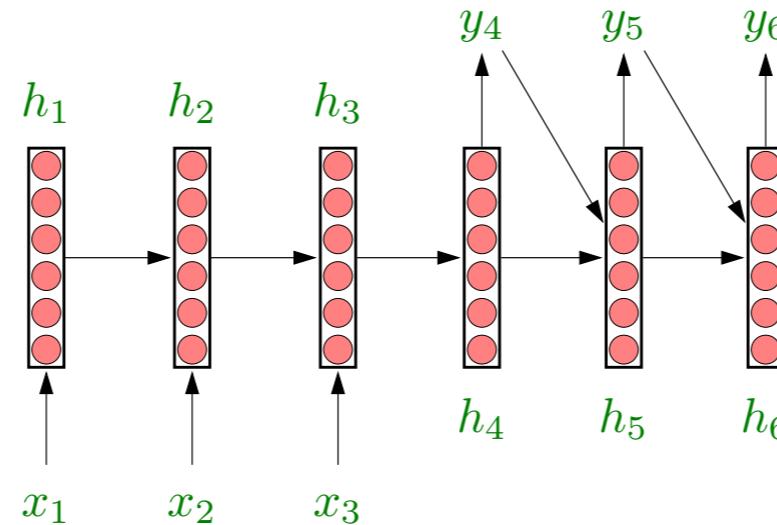


**Key idea: attention**

Learn to look back at your notes.



# Attention-based models



Distribution over input positions:

$$\alpha_t = \text{softmax}([\text{Attend}(h_1, h_{t-1}), \dots, \text{Attend}(h_L, h_{t-1})])$$

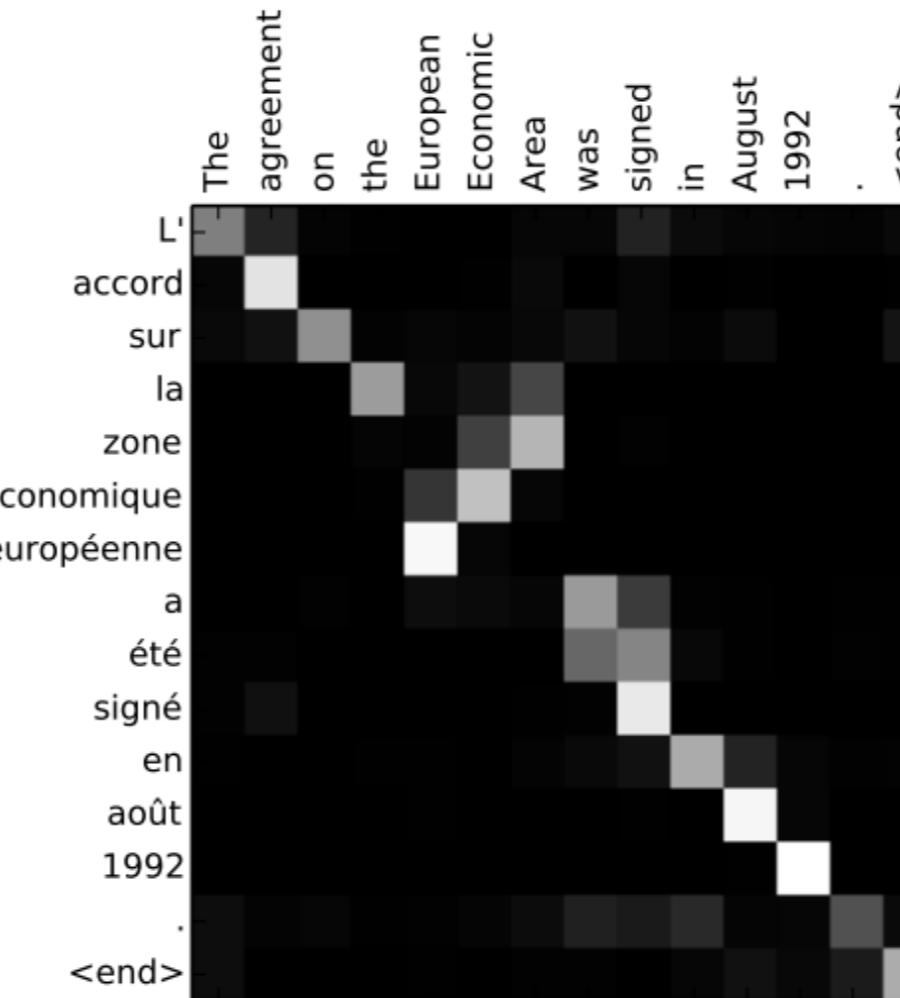
Generate with attended input:

$$h_t = \text{Encode}(h_{t-1}, y_{t-1}, \sum_{j=1}^L \alpha_t h_j)$$

Transformer models: attention only – no RNN!



# Machine translation





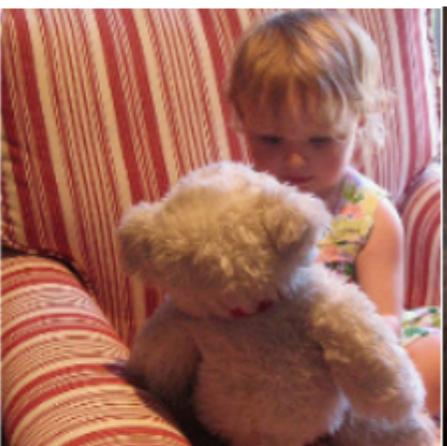
# Image captioning



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.





# Summary

- Recurrent neural networks: model sequences (non-linear version of Kalman filter or HMM)
- Logic intuition: learning a program with a for loop (reduce)
- LSTMs mitigate the vanishing gradient problem
- Attention-based models: when only part of input is relevant at a time
- Newer models with "external memory": memory networks, neural Turing machines





# Roadmap

Feedforward neural networks

Convolutional neural networks

Recurrent neural networks

**Unsupervised learning**

Final remarks



# Motivation

- Deep neural networks require lot of data
- Sometimes not very much labeled data, but plenty of unlabeled data (text, images, videos)
- Humans rarely get direct supervision; can learn from raw sensory information?



# Autoencoders

Analogy:

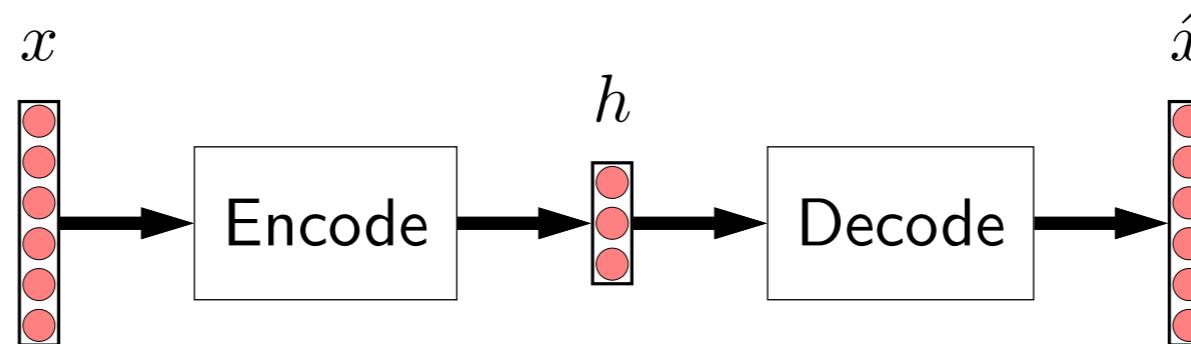
A A A A B B B B B → 4 A's, 5 B's → A A A A B B B B B



**Key idea: autoencoders**

If we can compress a data point and still reconstruct it, then we have learned something generally useful.

General framework:

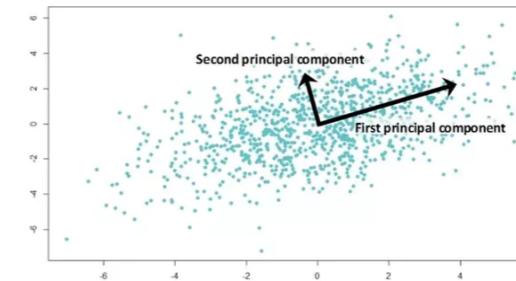


$$\text{minimize } \|x - \hat{x}\|^2$$



# Principal component analysis

Input: points  $x_1, \dots, x_n$



$$\text{Encode}(x) = U^\top \begin{matrix} x \\ \vdots \\ x \end{matrix}$$
$$\text{Decode}(h) = \begin{matrix} h \\ \vdots \\ h \end{matrix} U$$

(assume  $x_i$ 's are mean zero and  $U$  is orthogonal)

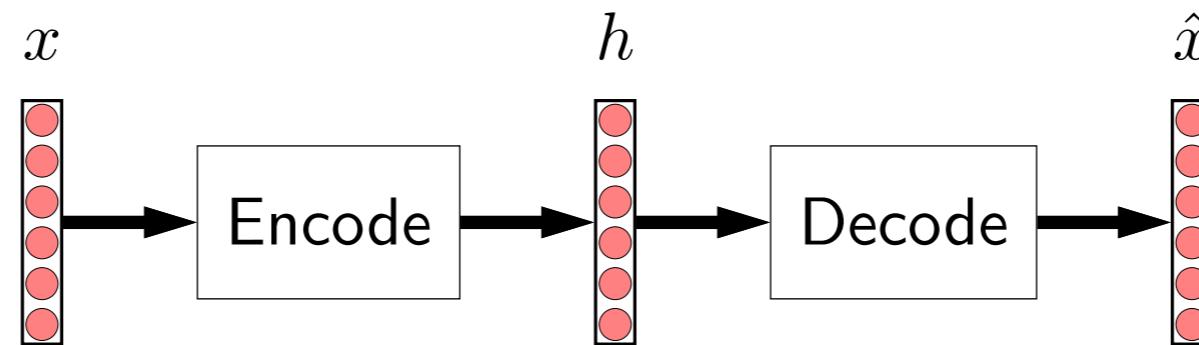
PCA objective:

$$\text{minimize} \sum_{i=1}^n \|x_i - \text{Decode}(\text{Encode}(x_i))\|^2$$



# Autoencoders

Increase dimensionality of hidden dimension:

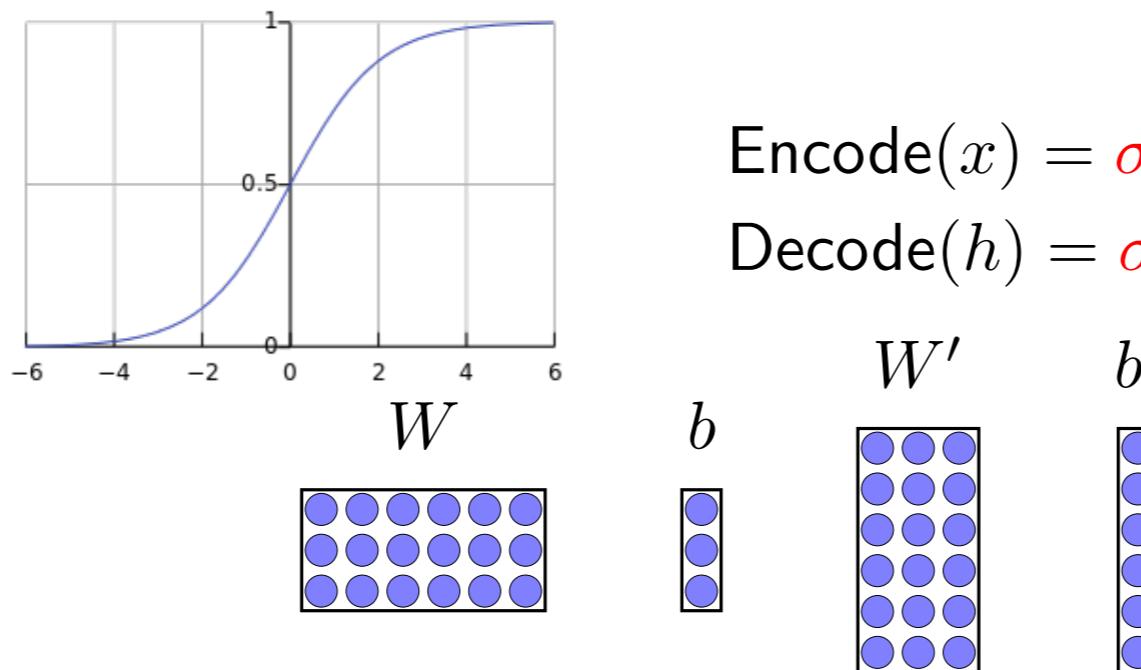


- Problem: learning nothing — just set Encode, Decode to identity function!
- Need to control complexity of Encode and Decode somehow...



# Non-linear autoencoders

Non-linear transformation (e.g., logistic function):



$$\text{Encode}(x) = \sigma(Wx + b)$$
$$\text{Decode}(h) = \sigma(W'h + b')$$

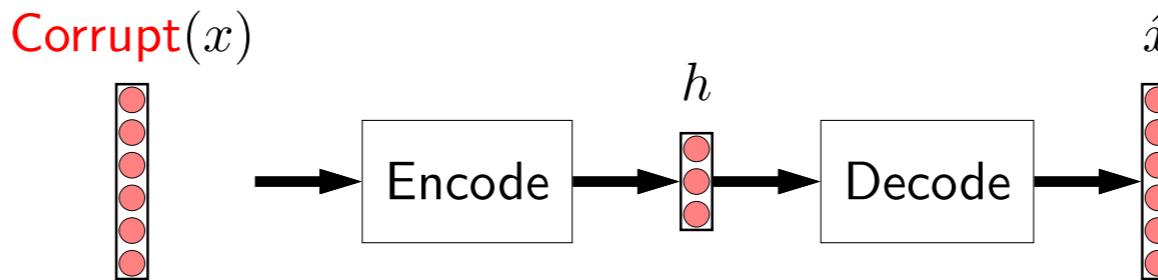
Loss function:

$$\text{minimize } \|x - \text{Decode}(\text{Encode}(x))\|^2$$

Key: non-linearity makes life harder, prevents degeneracy



# Denoising autoencoders



Types of noise:

- Blankout:  $\text{Corrupt}([1, 2, 3, 4]) = [0, 2, 3, 0]$
- Gaussian:  $\text{Corrupt}([1, 2, 3, 4]) = [1.1, 1.9, 3.3, 4.2]$

Objective:

$$\text{minimize } \|x - \text{Decode}(\text{Encode}(\text{Corrupt}(x)))\|^2$$

Algorithm: pick example, add fresh noise, SGD update

Key: noise makes life harder, prevents degeneracy

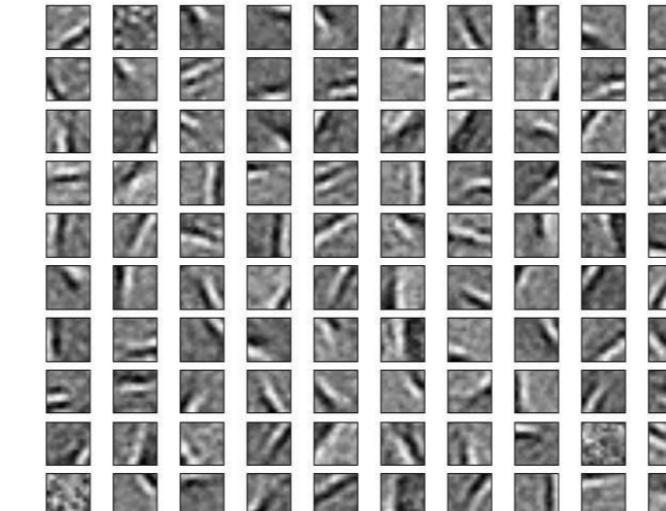
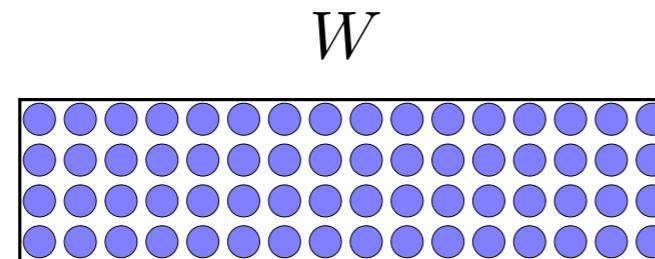


# Denoising autoencoders

MNIST: 60,000 images of digits (784 dimensions)



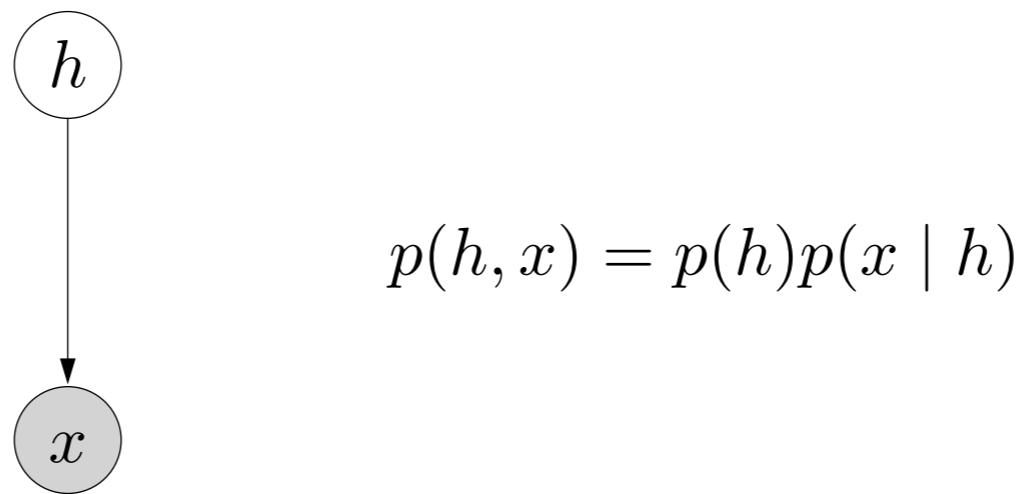
200 learned filters (rows of  $W$ ):





# Variational autoencoders

Motivation: learn a latent-variable model

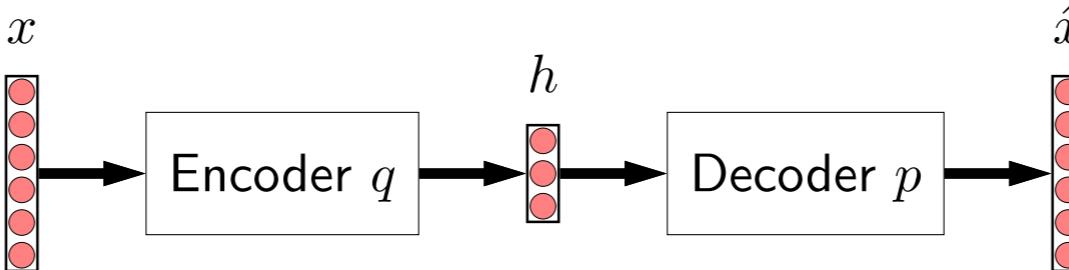


E-step in EM: computing  $p(h \mid x)$  is intractable

Solution: approximate using a neural network  $q(h \mid x)$



# Variational autoencoders



Objective: maximize

$$\log p(x) \geq \mathbb{E}_{q(h|x)}[\log p(x | h)] - \text{KL}(q(h | x) || p(h))$$

Algorithm:

- Sample  $h$  from encoder  $q$ , gradient update on  $q$  and  $p$
- Reparametrization trick [Kingma/Welling, 2014]



# Reading comprehension (SQuAD)

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?  
**gravity**

100K examples

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?  
**graupel**

Where do water droplets collide with ice crystals to form precipitation?  
**within a cloud**



# Raw text

Stanford University (officially Leland Stanford Junior University,<sup>[10]</sup> colloquially "the Farm") is a private research university in Stanford, California. Stanford is known for its academic strength, wealth, proximity to Silicon Valley, and ranking as one of the world's top universities.<sup>[11][12][13][14][15]</sup>

The university was founded in 1885 by Leland and Jane Stanford in memory of their only child, Leland Stanford Jr., who had died of typhoid fever at age 15 the previous year. Stanford was a U.S. Senator and former Governor of California who made his fortune as a railroad tycoon. The school admitted its first students on October 1, 1891,<sup>[2][3]</sup> as a coeducational and non-denominational institution.

Stanford University struggled financially after the death of Leland Stanford in 1893 and again after much of the campus was damaged by the 1906 San Francisco earthquake.<sup>[16]</sup> Following World War II, Provost Frederick Terman supported faculty and graduates' entrepreneurialism to build self-sufficient local industry in what would later be known as Silicon Valley.<sup>[17]</sup> The university is also one of the top fundraising institutions in the country, becoming the first school to raise more than a billion dollars in a year.<sup>[18]</sup>

The university is organized around three traditional schools consisting of 40 academic departments at the undergraduate and graduate level and four professional schools that focus on graduate programs in Law, Medicine, Education and Business. Stanford's undergraduate program is one of the top three most selective in the United States by acceptance rate.<sup>[19][20][21][22][23]</sup> Students compete in 36 varsity sports, and the university is one of two private institutions in the Division I FBS Pac-12 Conference. It has gained 117 NCAA team championships,<sup>[24]</sup> the most for a university. Stanford athletes have won 512 individual championships,<sup>[25]</sup> and Stanford has won the NACDA Directors' Cup for 23 consecutive years, beginning in 1994–1995.<sup>[26]</sup> In addition, Stanford students and alumni have won 270 Olympic medals including 139 gold medals.<sup>[27]</sup>

...

3.3 billion words



# Unsupervised pre-training







# BERT

Paris Talks \_\_\_ Stage for \_\_\_\_ as Risks to \_\_\_ Climate Rise

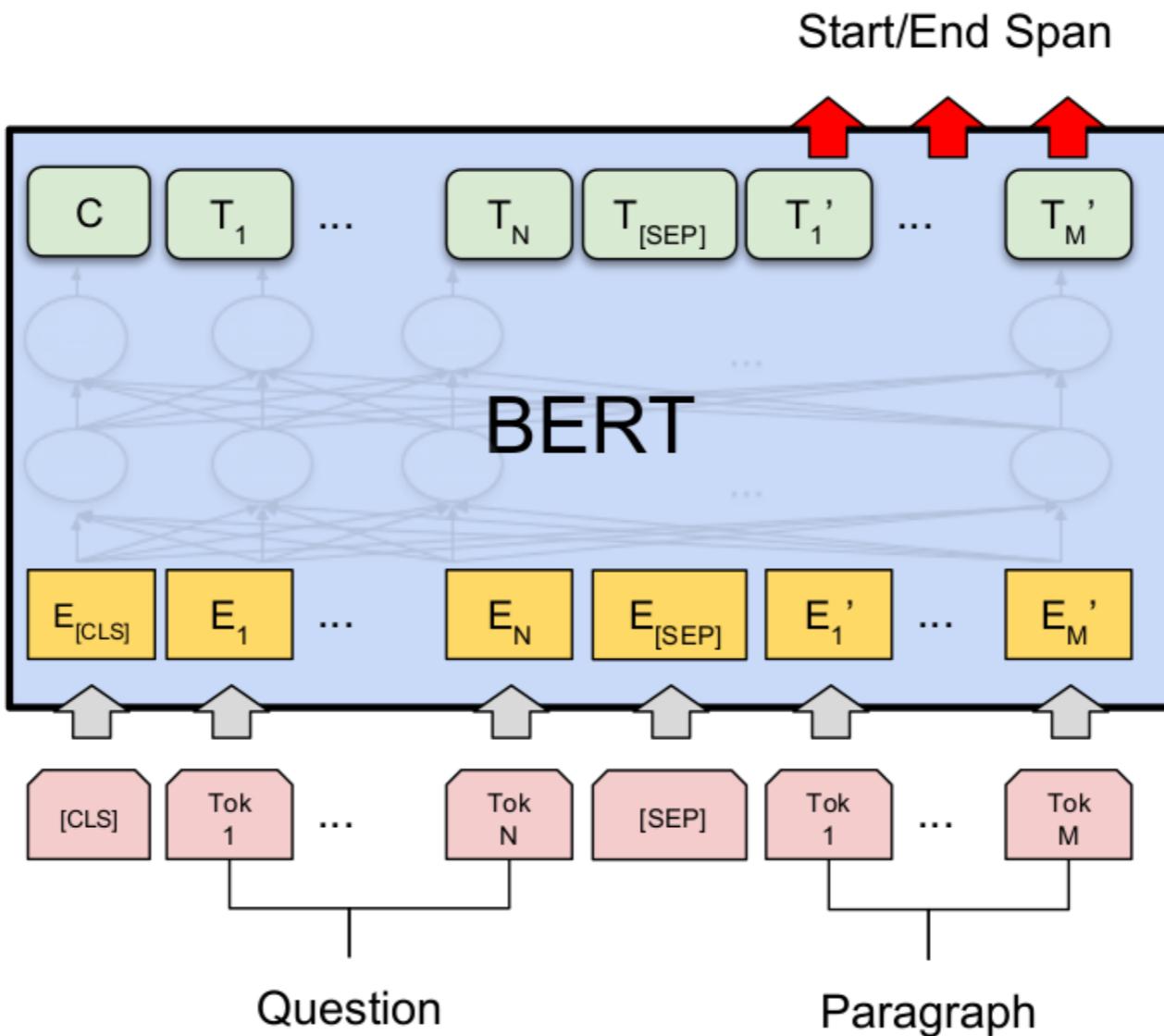


Paris Talks Set Stage for Action as Risks to the Climate Rise

- Tasks: fill in words, predict whether is next sentence
- Trained on 3.3B words, 4 days on 64 TPUs



# BERT





Rank	Model	EM	F1
	Human Performance <i>Stanford University</i> (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) <i>Google A.I.</i>	87.433	93.160
2 Oct 05, 2018	BERT (single model) <i>Google A.I.</i>	85.083	91.835
2 Sep 09, 2018	nInet (ensemble) <i>Microsoft Research Asia</i>	85.356	91.202
2 Sep 26, 2018	nInet (ensemble) <i>Microsoft Research Asia</i>	85.954	91.677
3 Jul 11, 2018	QANet (ensemble) <i>Google Brain &amp; CMU</i>	84.454	90.490
4 Jul 08, 2018	r-net (ensemble) <i>Microsoft Research Asia</i>	84.003	90.147
5 Mar 19, 2018	QANet (ensemble) <i>Google Brain &amp; CMU</i>	83.877	89.737
5 Sep 09, 2018	nInet (single model) <i>Microsoft Research Asia</i>	83.468	90.133
5 Jun 20, 2018	MARS (ensemble) <i>YUANFUDAO research NLP</i>	83.982	89.796
6 Sep 01, 2018	MARS (single model) <i>YUANFUDAO research NLP</i>	83.185	89.547





# Unsupervised learning

- Principle: make up prediction tasks (e.g.,  $x$  given  $x$  or context)
- Hard task → pressure to learn something
- Loss minimization using SGD
- Discriminatively fine tune: initialize feedforward neural network and backpropagate to optimize task accuracy
- How far can one push this?





# Roadmap

Feedforward neural networks

Convolutional neural networks

Recurrent neural networks

Unsupervised learning

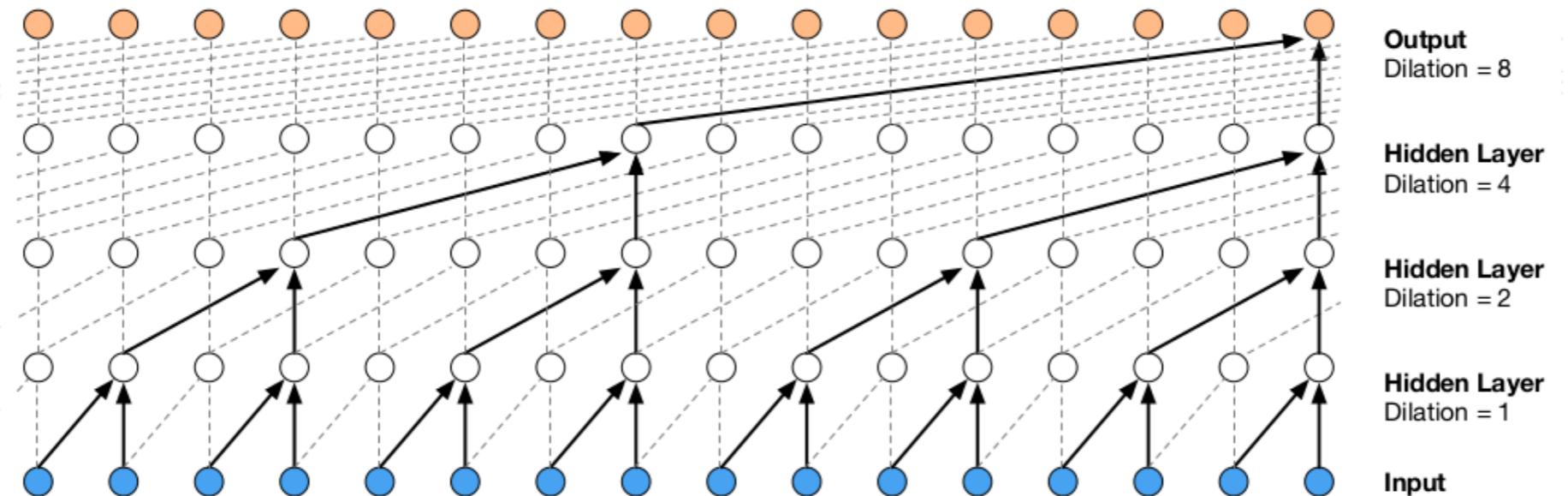
**Final remarks**



# WaveNet for audio generation



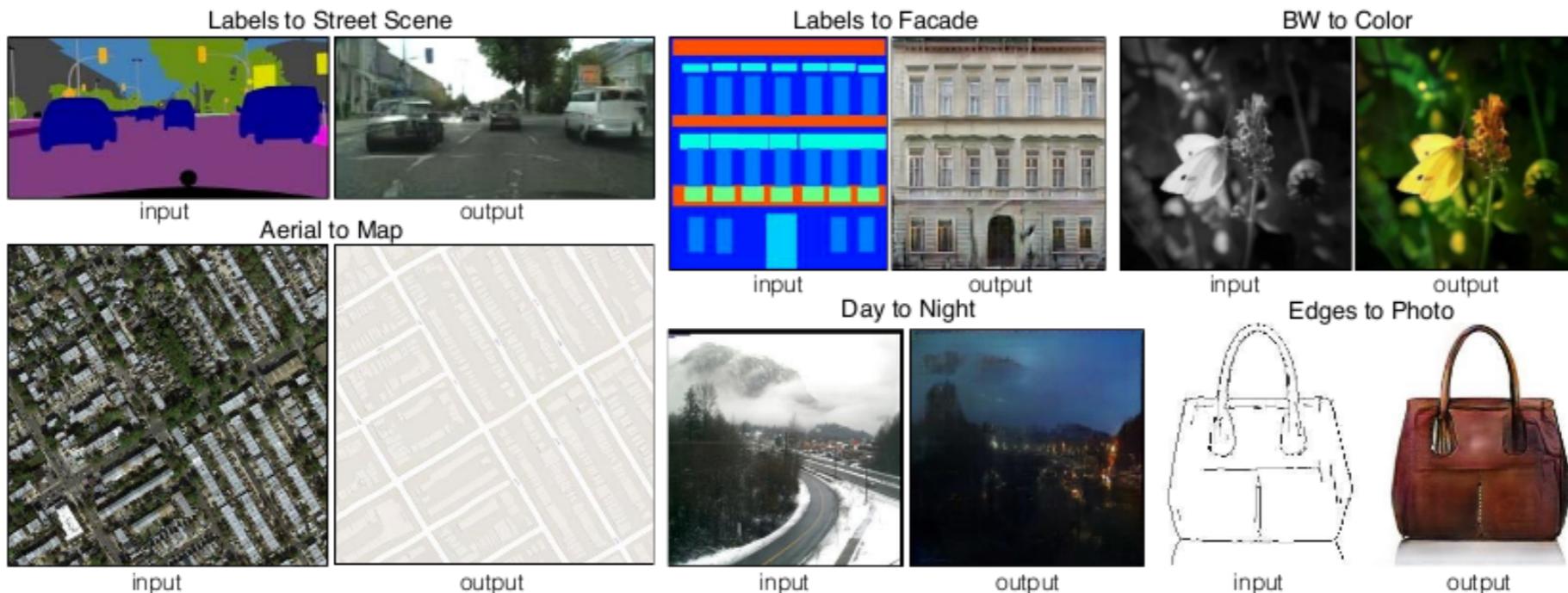
- Work with **raw** audio (16K observations / second)



- Key idea: **dilated convolutions** captures multiple scales of resolution, not recurrent



# Conditional adversarial networks



Key idea: game between

- **Generator:** generates fake images
- **Discriminator:** distinguishes between fake/real images



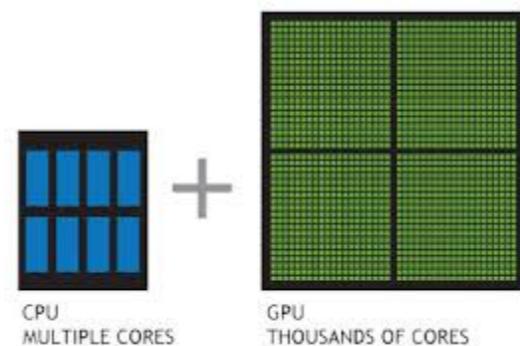
# Getting things to work

Better optimization algorithms: SGD, SGD+momentum, AdaGrad, AdaDelta, momentum, Nesterov, Adam

Tricks: initialization, gradient clipping, batch normalization, dropout

More hyperparameter tuning: step sizes, architectures

Better hardware: GPUs, TPUs



...wait for a long time...



# Theory: why does it work?

Two questions:

- Approximation: why are neural networks good hypothesis classes?
- Optimization: why can SGD optimize a high-dimensional non-convex problem?

Partial answers:

- 1-layer neural networks can approximate any continuous function on compact set [Cybenko, 1989; Barron, 1993]
- Generate random features works too [Rahimi/Recht, 2009; Andoni et. al, 2014]
- Use statistical physics to analyze loss surfaces [Choromanska et al., 2014]





# Summary

## Phenomena

Fixed vectors

Spatial structure

Sequence

Sequence-to-sequence

Unsupervised

## Ideas

Feedforward NNs

convolutional NNs

recurrent NNs

LSTMs

encoder-decoder

attention-based models

autoencoders

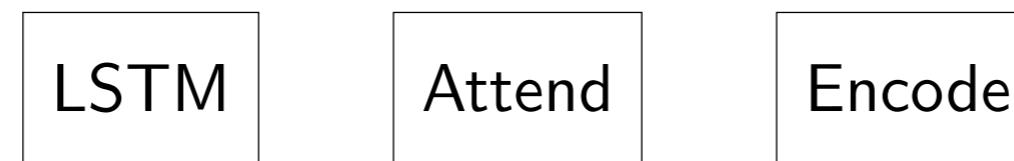
variational autoencoders

any auxiliary task



# Outlook

Extensibility: able to compose modules



Learning programs: think about analogy with a computer

