



Lecture 18: Deep Learning



CS221 / Spring 2020 / Finn & Anari

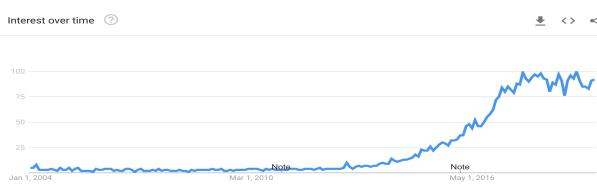
A brief history

- 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)

CS221 / Spring 2020 / Finn & Anari

Google Trends

Query: deep learning



CS221 / Spring 2020 / Finn & Anari



Reminders

- Projects due on Wednesday at noon PT (no late days allowed, to give time for grading).
- Exam regrade requests due Wednesday at midnight PT.

CS221 / Spring 2020 / Finn & Anari

1

A brief history

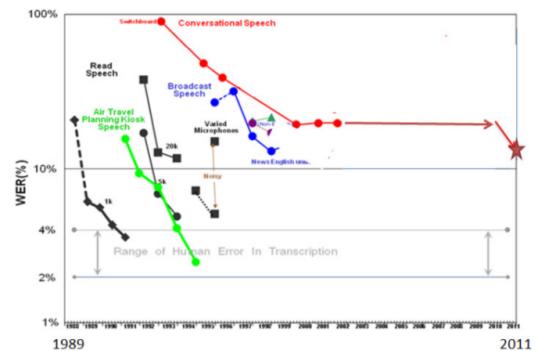
- 1980: Neocognitron, a.k.a. convolutional neural networks (Fukushima)
- 1989: backpropagation on convolutional neural networks (LeCun)
- 1990: recurrent neural networks (Elman)
- 1997: Long Short-Term Memory networks (Hochreiter/Schmidhuber)
- 2006: unsupervised layerwise training of deep networks (Hinton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

2 CS221 / Spring 2020 / Finn & Anari

3

[figure from Li Deng]

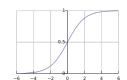
Speech recognition (2009-2011)



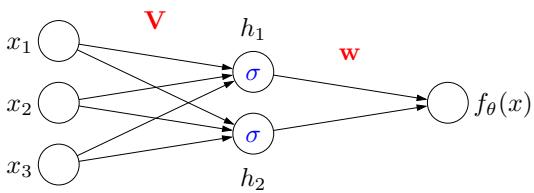
4 CS221 / Spring 2020 / Finn & Anari

5

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



Review: neural networks



Intermediate hidden units:

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

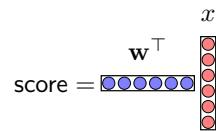
Output:

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

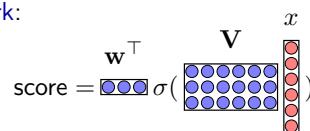
Parameters: $\theta = (\mathbf{V}, \mathbf{w})$

Deep neural networks

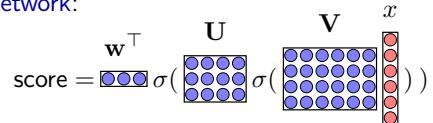
1-layer neural network:



2-layer neural network:

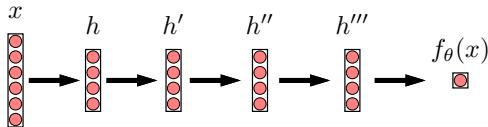


3-layer neural network:



...

Depth

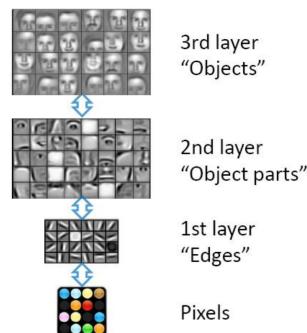


Intuitions:

- 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)
- Formal theory/understanding is still incomplete
- Some hypotheses emerging: double descent, lottery ticket hypothesis

[figure from Honglak Lee]

What's learned?



Review: optimization

Regression:

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

Key idea: minimize training loss

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

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



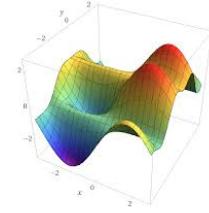
Algorithm: stochastic gradient descent

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

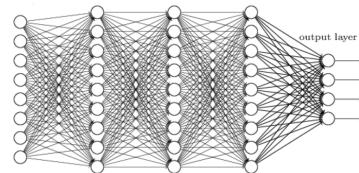
Computation (time/memory)



Information (data)



How to make it work



- More hidden units (over-parameterization)
- Adaptive step sizes (AdaGrad, Adam)
- Dropout to guard against overfitting
- Careful initialization (pre-training)
- Batch normalization

Model and optimization are tightly coupled

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



Roadmap

Feedforward neural networks

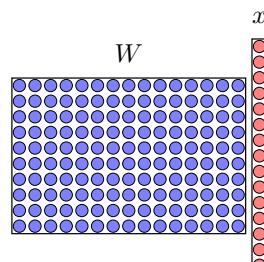
Convolutional networks

Sequence models

Unsupervised learning

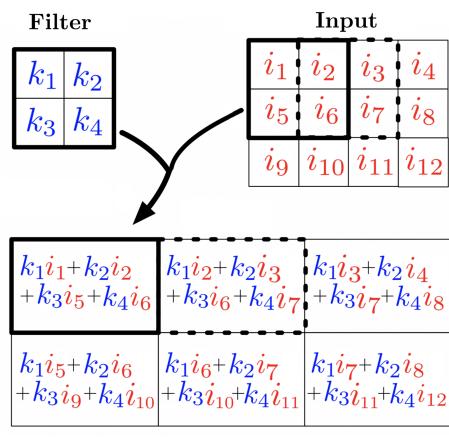
Final remarks

Motivation

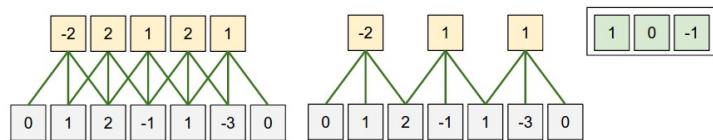


- **Observation:** images are not arbitrary vectors
- **Goal:** leverage spatial structure of images (translation equivariance)

Idea: Convolutions

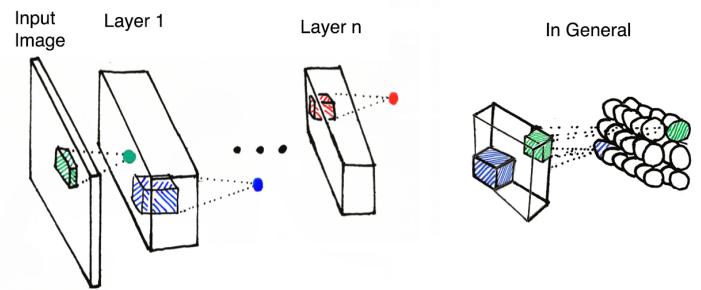


Prior knowledge



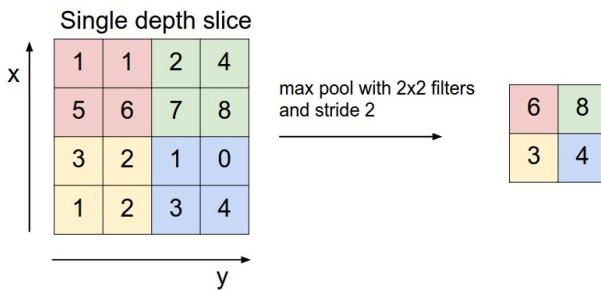
- **Local connectivity:** each hidden unit operates on a local image patch (3 instead of 7 connections per hidden unit)
- **Parameter sharing:** processing of each image patch is same (3 parameters instead of $3 \cdot 5$)
- **Intuition:** try to match a pattern in image

Convolutional layers



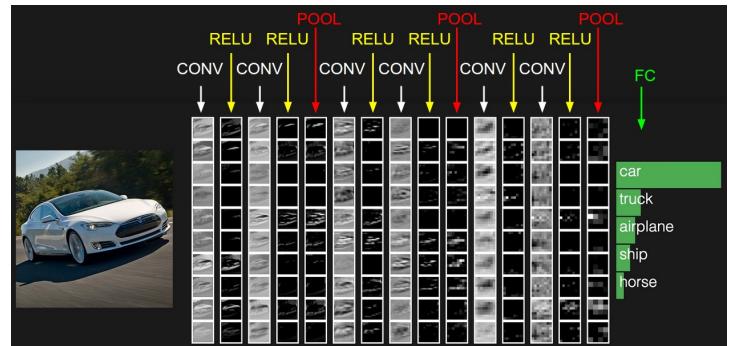
- Instead of vector to vector, we do volume to volume
- [Andrej Karpathy's demo]

Max-pooling



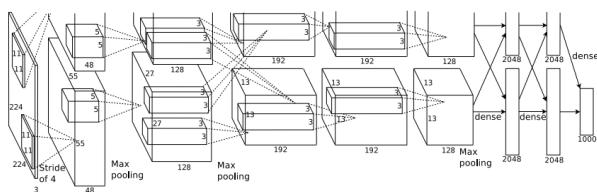
- Intuition: test if there exists a pattern in neighborhood
- Reduce computation, prevent overfitting

Example of function evaluation



[Andrej Karpathy's demo]

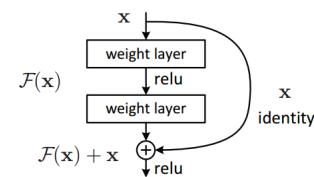
AlexNet



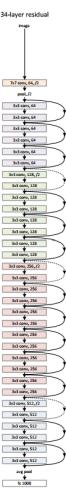
- **Non-linearity:** use ReLU ($\max(z, 0)$) instead of logistic
- **Data augmentation:** translate, horizontal reflection, vary intensity, dropout (guard against overfitting)
- **Computation:** parallelize across two GPUs (6 days)
- **Results on ImageNet:** 16.4% error (next best was 25.8%)

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 1: locality of connections, capture spatial structure
- Key idea 2: Filters share parameters, capture translational equivariance
- Depth matters
- Applications to images, text, Go, drug design, etc.



Roadmap

Feedforward neural networks

Convolutional networks

Sequence models

Unsupervised learning

Final remarks

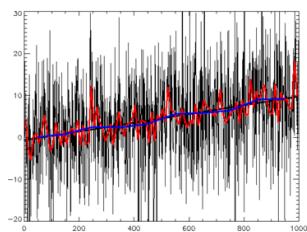
Motivation: modeling sequences

Sentences:

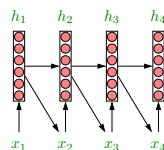
$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}$

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

Time series:



Language models with RNNs

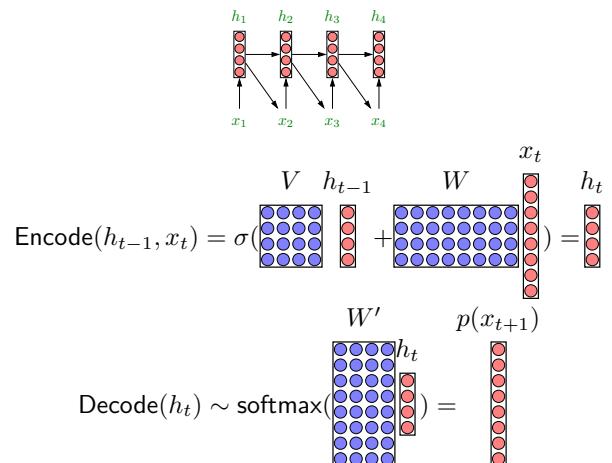


$h_1 = \text{Encode}(x_1)$
 $x_2 \sim \text{Decode}(h_1)$
 $h_2 = \text{Encode}(h_1, x_2)$
 $x_3 \sim \text{Decode}(h_2)$
 $h_3 = \text{Encode}(h_2, x_3)$
 $x_4 \sim \text{Decode}(h_3)$
 $h_4 = \text{Encode}(h_3, x_4)$

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

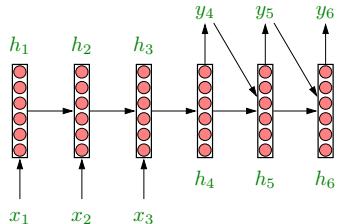


Sequence-to-sequence model

Motivation: machine translation

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

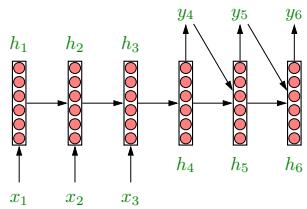
y: Fear the man of one book.



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



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!

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.

Attention-based models

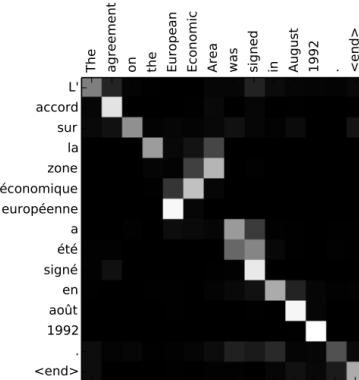
Motivation: long sentences — compress to finite dimensional vector?

Ein 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.

Machine translation



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 networks

Sequence models

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:

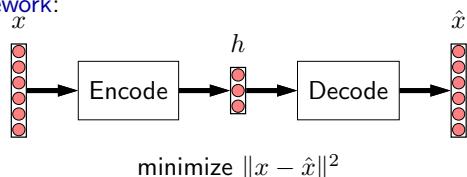
A A A A B B B B B B → 4 A's, 5 B's → A A A A B B B B B B



Key idea: autoencoders

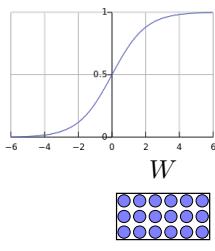
If we can compress a data point and still reconstruct it, then we have learned something generally useful.

General framework:



Non-linear autoencoders

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



$$\text{Encode}(x) = \sigma(Wx + b)$$

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

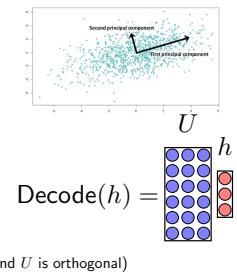
Loss function:

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

Principal component analysis

Input: points x_1, \dots, x_n

$$\text{Encode}(x) = U^\top x$$



$$\text{Decode}(h) = Uh$$

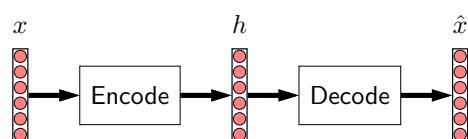
(assume x_i 's are mean zero and U is orthogonal!)

PCA objective:

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

Autoencoders

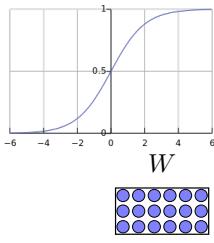
Increase dimensionality of hidden dimension:



- Problem: learning nothing — just set Encode, Decode to identity function!

Non-linear autoencoders

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



$$\text{Encode}(x) = \sigma(Wx + b)$$

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

W b W' b'

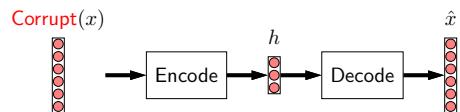
Loss function:

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

Key: Compressing h (e.g. low-dim, sparse) prevents identity

CS221 / Spring 2020 / Finn & Anari

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, identity function no longer good

CS221 / Spring 2020 / Finn & Anari

54 CS221 / Spring 2020 / Finn & Anari

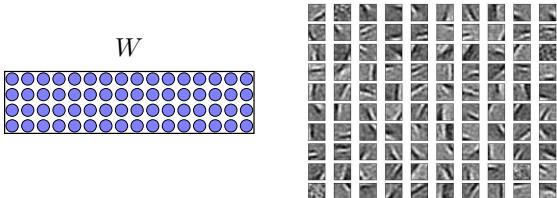
55

Denoising autoencoders

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



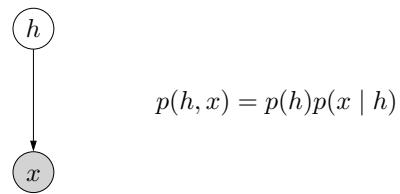
200 learned filters (rows of W):



CS221 / Spring 2020 / Finn & Anari

Variational autoencoders

Motivation: learn a latent-variable model



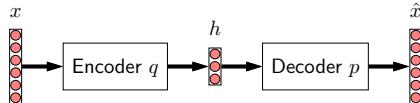
E-step in EM: computing $p(h | x)$ is intractable

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

56 CS221 / Spring 2020 / Finn & Anari

57

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

CS221 / Spring 2020 / Finn & Anari

58 CS221 / Spring 2020 / Finn & Anari

59

Raw text

Stanford University (officially Leland Stanford Junior University)^[10] (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.^{[11][12][13][14][15]}
 The university was founded in 1895 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,^[23] 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.^[16] Following World War II, Provost Frederic Terman supported faculty and graduates' entrepreneurship to build self-sufficient local industry in what would later be known as Silicon Valley.^[17] 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.^[18]

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.^{[19][20][21][22][23]} 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,^[24] the most for a university. Stanford athletes have won 512 individual championships,^[25] and Stanford has won the NACDA Directors' Cup for 23 consecutive years, beginning in 1994–1995.^[26] In addition, Stanford students and alumni have won 270 Olympic medals including 139 gold medals.^[27]

...

3.3 billion words

Unsupervised pre-training

labeled

unlabeled



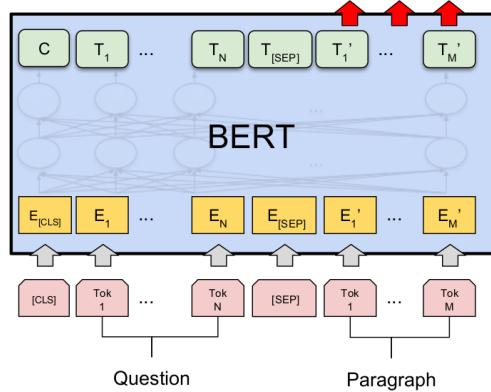
BERT

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

Start/End Span



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

Sequence models

Unsupervised learning

Final remarks

Getting things to work

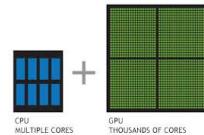
Better optimization algorithms: SGD, SGD+momentum, AdaGrad, AdaDelta, momentum, Nesterov, Adam

Tricks: initialization, gradient clipping, batch normalization, dropout

More hyperparameter tuning: step sizes, architectures

More data: larger, broader datasets

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

