What is machine learning and deep learning?

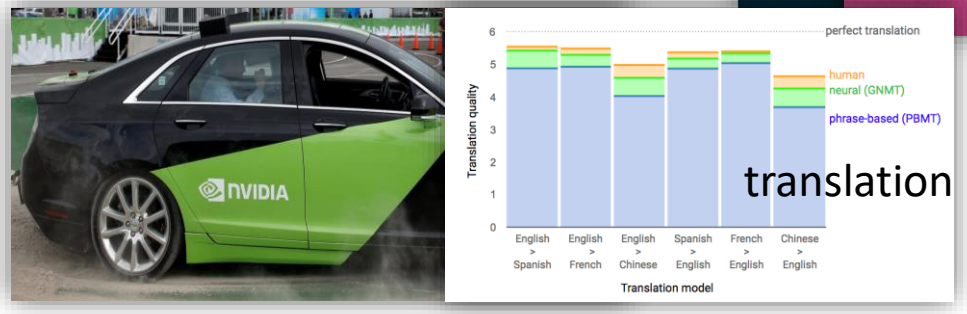
Sanjeev Arora

Princeton University and Institute for Advanced Study

http://www.cs.princeton.edu/~arora/
Group website: unsupervised.princeton.edu
Blog: www.offconvex.org
Twitter: @prfsanjeevarora

Support: NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Resarch, Mozilla Research. DARPA/SRC

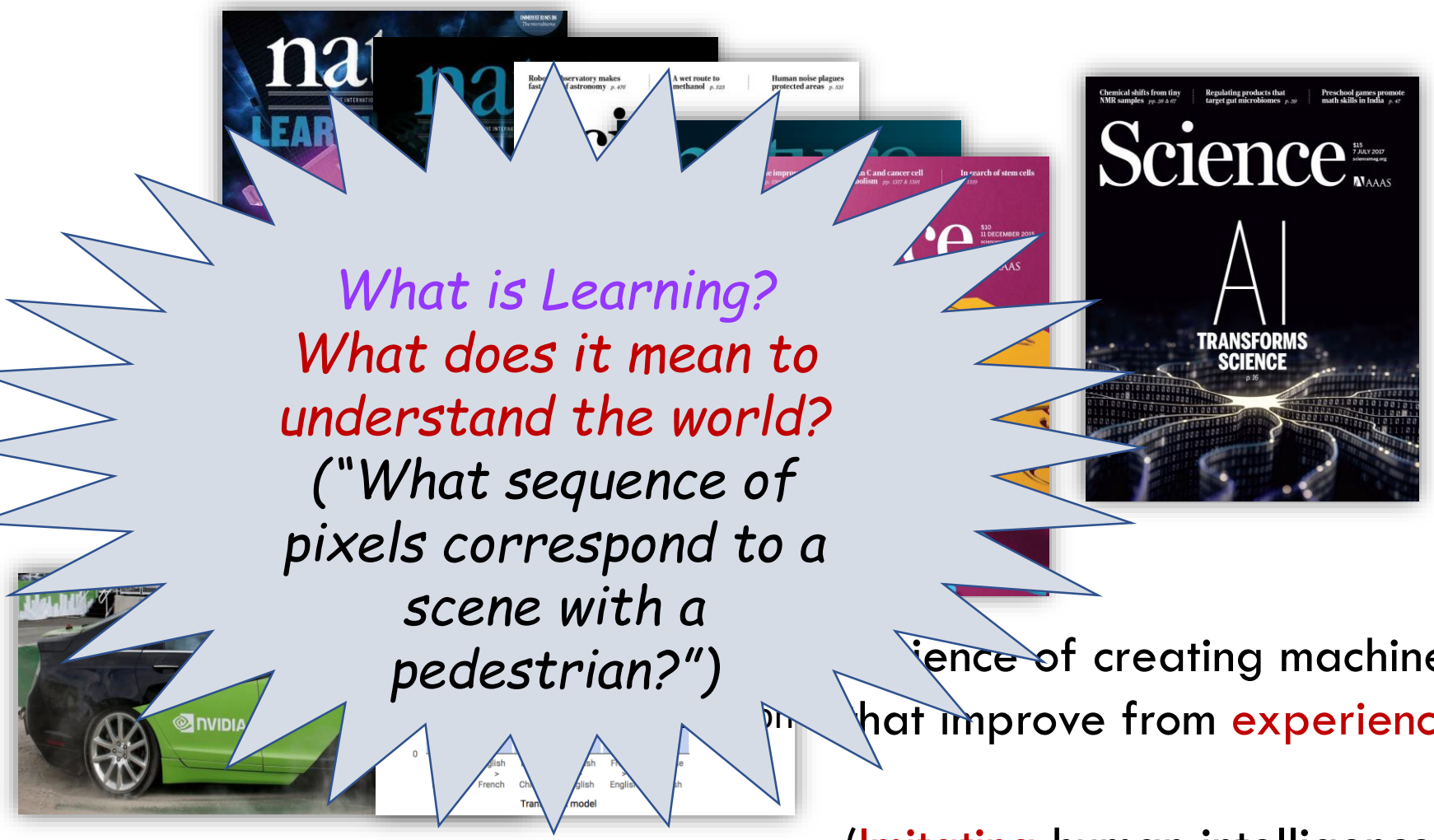
Machine learning (ML): A new kind of science



"Science of creating machines/programs that improve from experience and interaction."

(Imitating human intelligence not an explicit goal.)

Machine learning (ML): A new kind of science



What is Learning?
What does it mean to understand the world?
("What sequence of pixels correspond to a scene with a pedestrian?")

ience of creating machines/programs
hat improve from experience and interaction."

(Imitating human intelligence not an explicit goal.)

Talk overview

- Part 1: Mathematical formulation of Machine Learning (ML).
- Part 2: ML in action (unsupervised, sequential decision-making)
- Part 3: Toward mathematical understanding of deep learning
- Part 4: Taking Stock/Concluding thoughts

"A tour with many stops…"

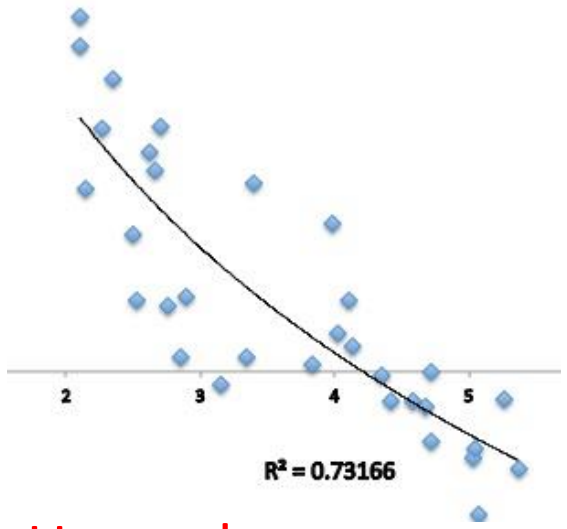
Part 1

Mathematical formalization of Machine Learning (ML)

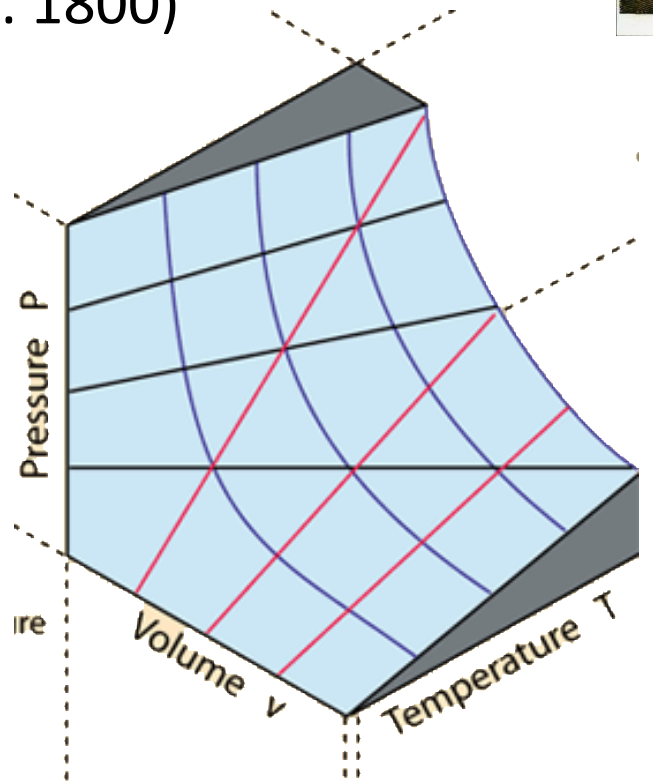
(Operationally speaking, boils down to learning patterns in data.)

Old Idea: Curve fitting (Legendre, Gauss, c. 1800)

Phillips curve (1958):



Gas Law (c. 1800)
PV = nRT



Machine Learning uses surface fitting, with many more variables (eg 20M)
("Learning patterns in data.")

Example: Learning to score reviews

Score = −1.5

"Slow moving. Couldn't get into this movie despite all the awards it has won."

Score = 2.5

"Wow! Did not know what to expect and was delighted! Loved the homage paid to the musicals of old. "

Given: Reviews for different movies and rating score (-3 to +3).

Task: Learn to predict rating score given text of a new review.

The "law" of movie review scores??

Example: Learning to rate reviews (contd)

"Slow moving. Couldn't get into this movie despite all the awards it has won."

100,000 words in English dictionary.

Hypothesized "law" (simple linear model):

- Each word w has sentiment score $\theta_w \in R$
  Review score ≈ total sentiment score of all words in it

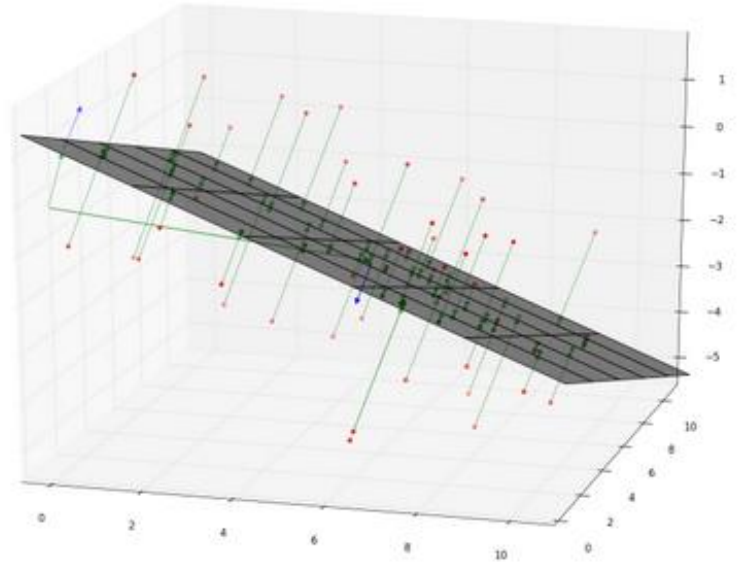
Finding sentiment score for 100,000 words ≡ Surface fitting with 100,000 variables!

R^{10000}

100,000 words in English dictionary.

Hypothesized "law" (simple linear model):

- Each word w has sentiment score $\boldsymbol{\theta}_w$
  Review rating $\approx$ total sentiment score of all words in it



$$Min \sum_{review} \left(Score - \sum_{w \in review} \boldsymbol{\theta}_w \right)^2$$

"Loss function"

Algorithm: "Gaussian Least Square Fit";
(Solvable in seconds given few million ranked reviews.)

Hypothesized law:

- Each word w has sentiment score θ_w

  Review rating ≈ total sentiment score of all words in it

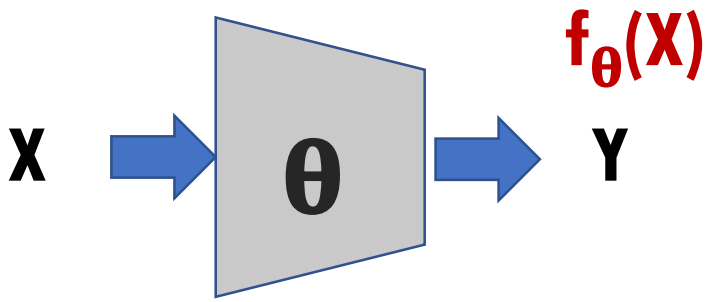
Word	Score
as	+0.1
good	+0.6
homage	0.0
loved	+1.1
musicals	+0.1
of	-0.1
old	-0.4
paid	-0.3
story	+0.3
the	+0.1
to	-0.1
was	-0.2
well	+1.4
⋮	⋮

"Loved the homage paid to the musicals of old. Story was good as well."

Loved	the	homage	paid	to	the	musicals	of	old.	Story	was	good	as	well.		
+1.1	+0.1	+0.0	-0.3	-0.1	+0.1	+0.1	-0.1	-0.4	+0.3	-0.2	+0.6	+0.1	+1.4	=	**2.7**

"Discovered Patterns" = How words correlate with positive score

ML ≈ finding suitable function ("model") given examples of desired input/output behavior

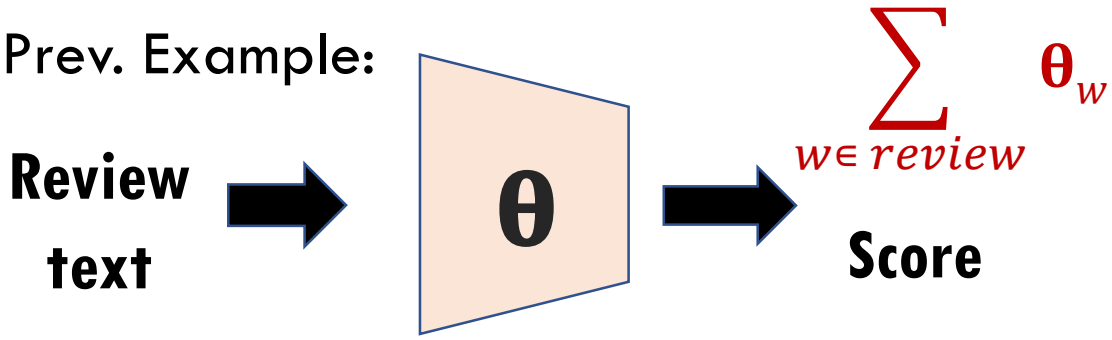


$\theta \in R^d$

(trainable parameters)

(X, Y) : (Input, Output) pair

Prev. Example:



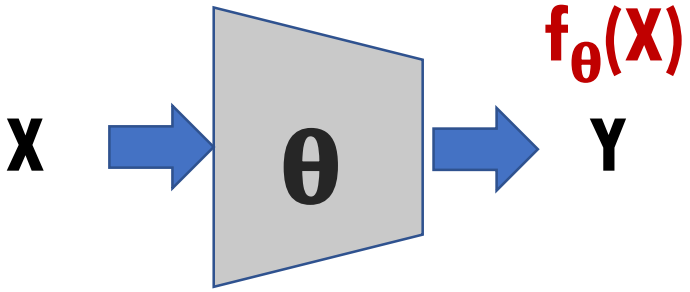
Training method:

$$Min \sum_{review} \left(Score - \sum_{w \in review} \theta_w \right)^2$$

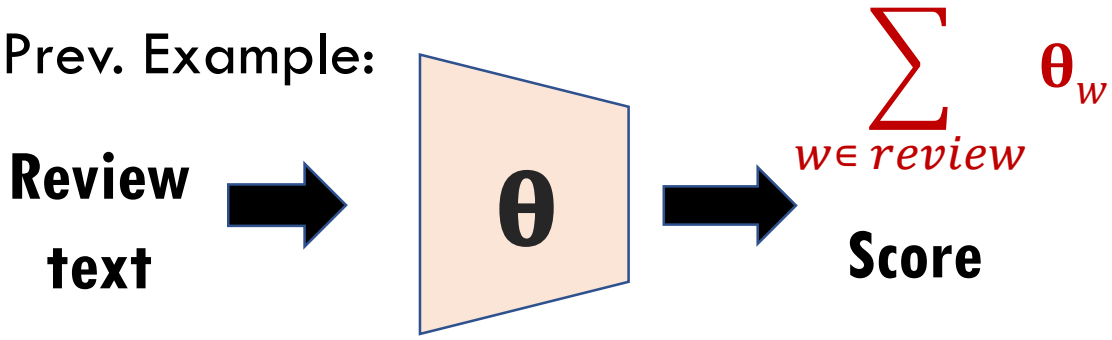
Desired output

Model's output

ML ≈ finding suitable function ("model") given examples of desired input/output behavior



Prev. Example:



$\theta \in R^d$

(trainable parameters)

Training data: $\{(X_1^{(i)}, Y^{(i)}): i=1,..,N\}$

"Loss" $\ell(\theta) = \sum_{i=1}^{N} (f_\theta(X_1^{(i)}) - Y^{(i)})^2$

(Many other loss formulations exist….)

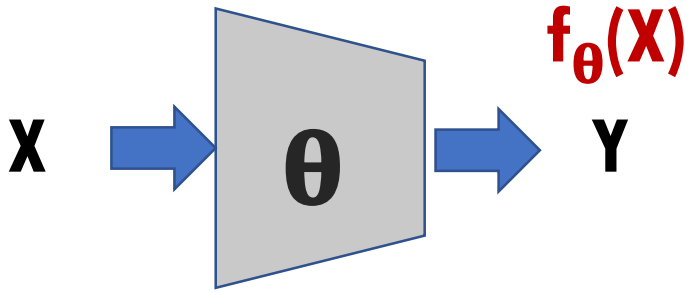
Training method:

$$Min \sum_{review} \left(Score - \sum_{w \in review} \theta_w \right)^2$$

Desired output

Model's output

Formal framework (inherited from classical statistics)



$\theta \in R^d$

(trainable parameters)

Training data: $\{(X_1^{(i)}, Y^{(i)}): i=1,..,N\}$

"Loss" $\ell(\theta) = \sum_{i=1}^{N} (f_\theta(X_1^{(i)}) - Y^{(i)})^2$

(Many other loss formulations exist….)

Assumption: Samples $(X^{(i)},Y^{(i)})$ drawn from a probability distribution D (e.g., distribution on pairs "(movie review, rating score)")

TRAINING: Optimize loss $\ell(\theta)$.

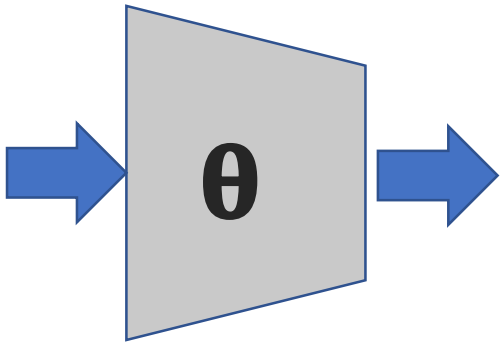
TESTING: Take new X's and see how well the trained model predicts the missing Y (e.g., "given review, predict rating score")

In practice: TRAIN on 80% of data;
TEST by evaluating loss on remaining 20%

Training via Gradient Descent ("natural algorithm")

$\theta \in \mathbb{R}^d$

trainable parameters

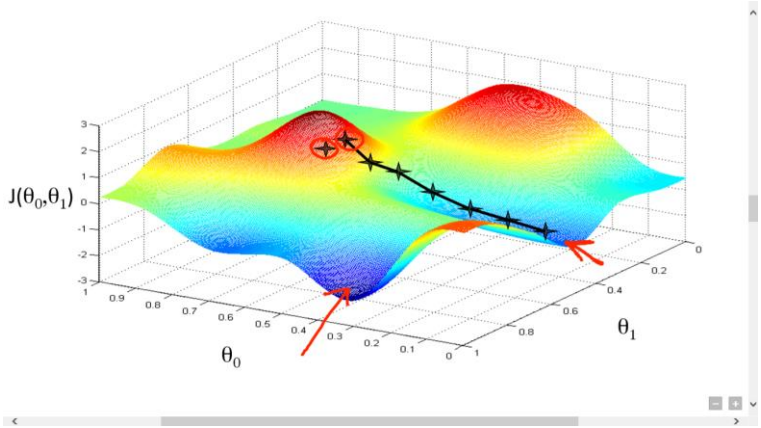


Loss $\ell(\theta)$

deficiency in desired fit

Often nonconvex, esp. in deep learning.

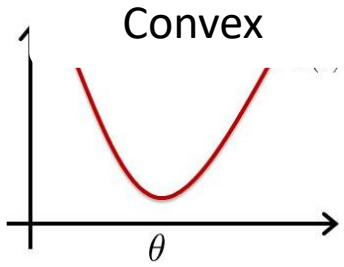
Starting with some $\theta^{(0)}$, compute for t=0,1,2,..

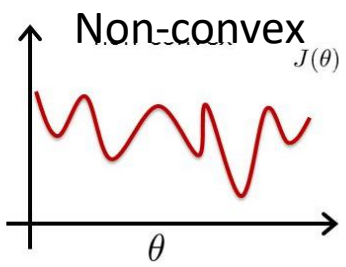


$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla(\ell)$$

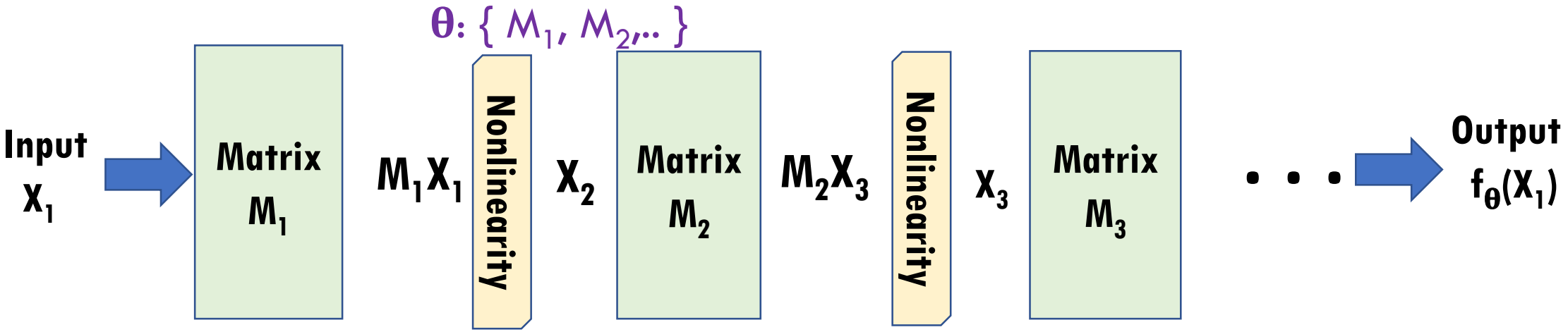
($\eta \in \mathbb{R}$ is "learning rate", say, 0.01)

In practice can improve GD with tricks: time-varying η, past gradients ("momentum"), "regularization", ..





Subcase: deep learning* (deep models = "multilayered")



"Nonlinearity": Given a vector, output same vector but negative entries turned to zero.

Training data: $\{(X_1^{(i)}, Y^{(i)}): i=1,..,N\}$ $\ell(\theta) = \sum_{i=1}^{N} (f_\theta(X_1^{(i)}) - Y^{(i)})^2$

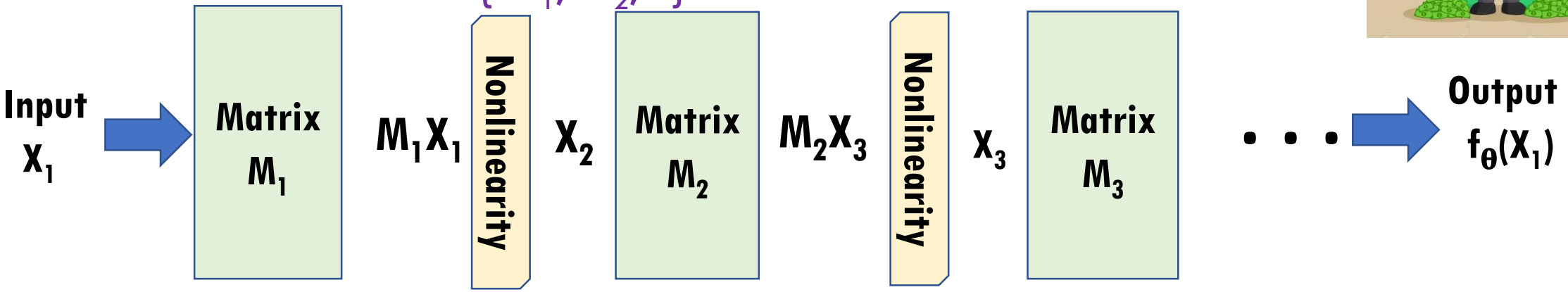
How to compute gradient of Loss??

Backpropagation Algorithm does it fast; clever application of chain rule [Werbos'77, Rumelhart et al'84]

(*Highly simplistic: could have "convolution", "bias", "skip connections", other loss fns etc.)

Subcase: deep learning* (deep models = "multilayered")

θ: { M_1, M_2,.. }



"Nonlinearity": Given a vector, output same vector but negative entries turned to zero.

Training data: {($X_1^{(i)}$, $Y^{(i)}$): i=

$$\sum_{i=1}^{N} \left(f_\theta(X_1^{(i)}) - Y^{(i)} \right)^2$$

Nonlinear enough to express many things; linear enough to allow quick optimization on today's computers

How to compute gradie

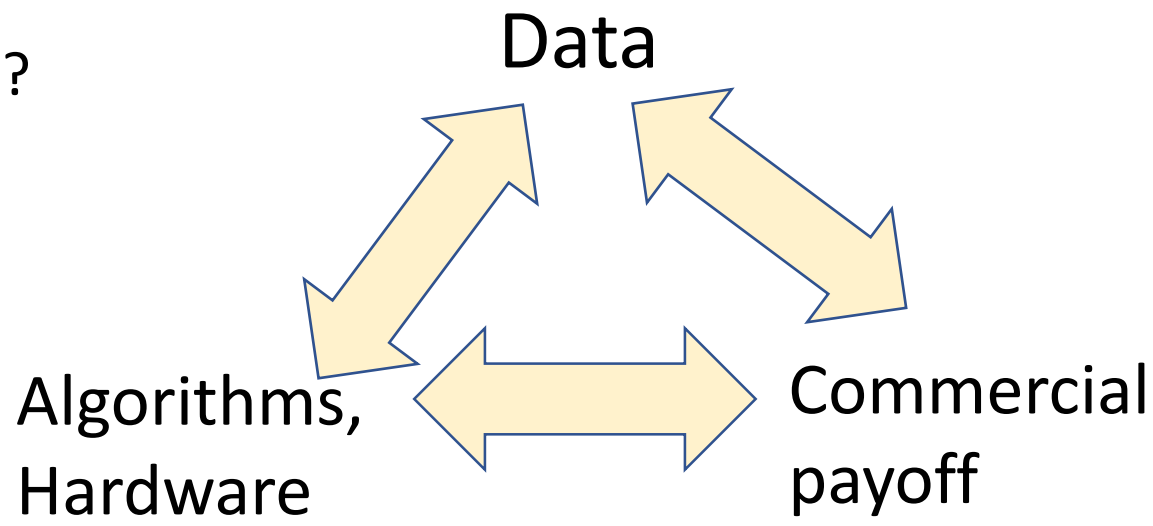
Backpropagation Al

et al'84]

(*Highly simplistic: could have "convolutio", ble skip nnections", other loss fns etc.)

Summary so far:

- Machine learning involves finding patterns in data.
- Finding patterns = some kind of surface fitting involving millions of variables.
- Why did it take off in past decade??



Part 2

Machine Learning in Action (Unsupervised, Interactive, etc.)

(Key idea: need to define “loss function” creatively….)

Review rating = total sentiment score of all words in it $\sum_{w \in review} \theta_w$

This is not how we humans do it!

(i) Word order in text matters. ("No good" vs "Good, no?")

(ii) Don't need millions of examples. (Because we understand language!)

Science has long tried to understand and formalize language + semantics
(using logic, grammars, model theory,…)

How can machines "understand" language?

(Arguably, more general-purpose than rating reviews!)

Unsupervised learning (no human-supplied labels)

- Key idea: Using large corpus (eg, Wikipedia), train model to predict part of text from adjacent text.

Example: "*I went to a café and ordered a….* "

Model learns to do such completions by training on huge amounts of text….

(In the process, implicitly picks up on grammar rules, common sense etc.)

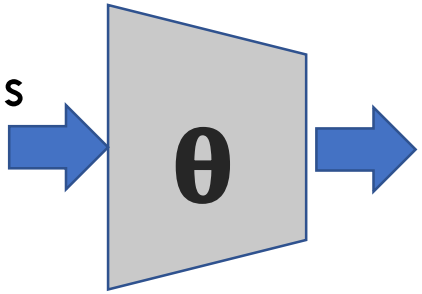
Used in: Machine Translation, Question-Answering (e.g. Siri, Alexa..)

"I went to a café and ordered a…. "

A Language model (baby "word2vec" [Mikolov et al'13])

Preceding words

$w_1, w_2, \ldots, w_5$



Distribution on all English words

$$\Pr[w|w_1 \ldots w_5] \propto \exp\left(\frac{1}{5}\sum_i \langle v_w, v_{w_i}\rangle\right)$$

(implicit normalizing term; will ignore for simplicity)

$\theta = \{ v_w \in R^{300} : w \text{ an English word} \}$

("semantic vectors")

	M	T	W	TH	F	S	S
Chance of rainfall	70%	80%	90%	80%	60%	20%	0%
							

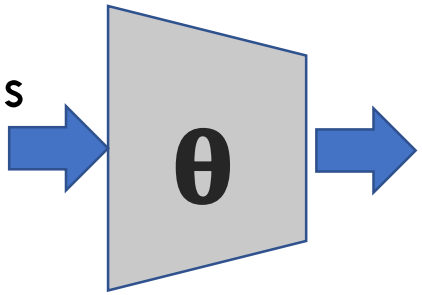
Measure of goodness of fit?

(i.e., loss function)?

A Language model (baby "word2vec" [mikolov et al'13])

Preceding words

$w_1, w_2, \ldots, w_5$



Distribution on all English words

$$\Pr[w|w_1 \ldots w_5] \propto \exp(\frac{1}{5}\sum_i \langle v_w, v_{w_i} \rangle)$$

$\theta = \{ v_w \in R^{300} : w \text{ an English word} \}$

("semantic vectors")

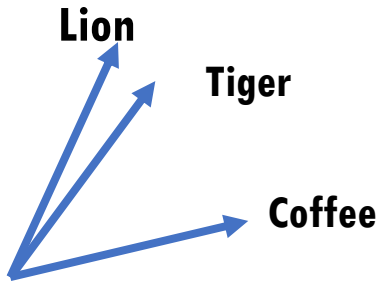
Loss $\ell(\theta)$: Reciprocal of Probability assigned by model to Wikipedia = $w_1 w_2 w_3 \ldots w_N$

$$\prod_{i=6}^{N} \Pr[w_i | w_{i-5}, \ldots, w_{i-1}]$$

$$= \exp(\sum_{i=6}^{N} \sum_{j=1}^{5} \frac{1}{5} \langle v_{w_i}, v_{w_{i-j}} \rangle)$$

(* Omitting negative sampling term…)

Properties of semantic word vectors



Cosine of angle captures human estimates of "similarity"[Deerwester et al'90]

(Possible to use word vectors to define sentence/paragraph vectors that capture sentence/paragraph similarity ["SIF embeddings" [A., Liang, Ma,'17])

Later!

Incorporating semantic vectors improves movie rating task …

Word vector space for different languages (e.g., English, French) can be meaningfully aligned via linear transformation [Lample et al'18, Arttextxe et al'18]

Hmm, predicting review scores, passively understanding word meaning…

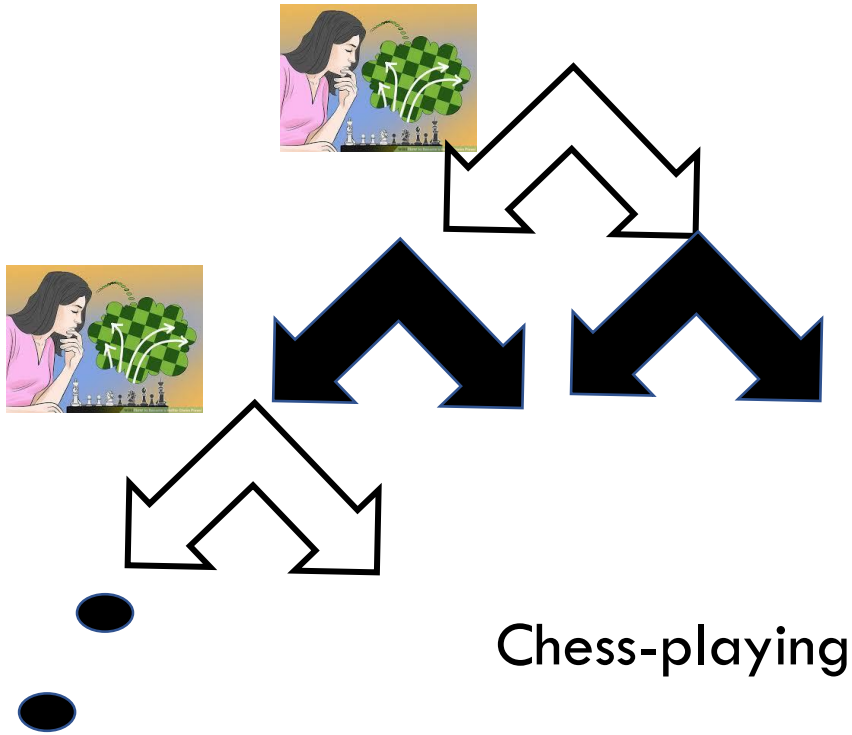
How can machines actively interact with the world, via sequence of intelligent decisions ….?

20th century antecedents

Probability Theory: Betting/gambling games (eg martingales)…
Economics: managing stock portfolios; playing repeated games,..
Control theory for power plants/machines,..

Sequential decision-making: framework

(uses: robotics, exploration..)



1=LOSE
0 = WIN

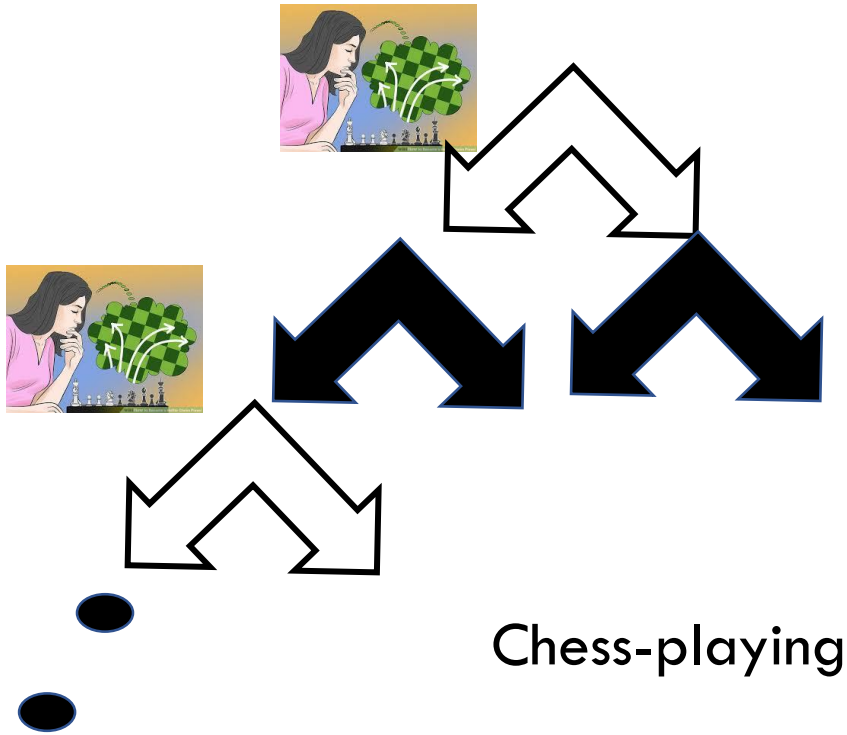
- Tree of all possible action/interactions. (responses can be stochastic)
- Optimum move = one that minimizes expected loss

Practical difficulty: tree too large to allow full evaluation!

Chess-playing software (circa 1990s): Decision-maker is

list of handcrafted rules
+
move evaluation algorithm
(approximate; limited lookahead)

Sequential decision-making (framework)



1=LOSE
0 = WIN

- Tree of possible action/interactions. (responses can be stochastic)
- Optimum move = one that minimizes expected loss

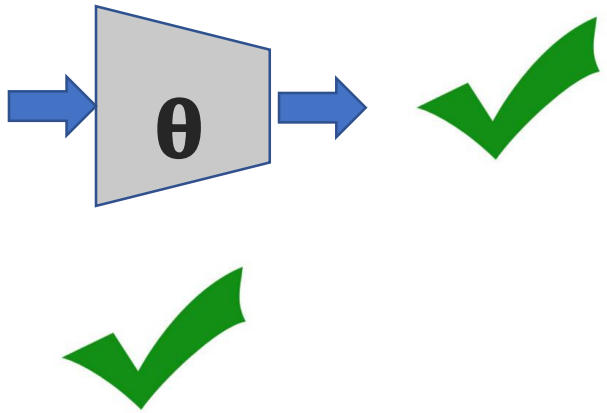
Practical difficulty: tree too large to allow full evaluation!

Chess-playing software (circa 1990s): Decision-maker is

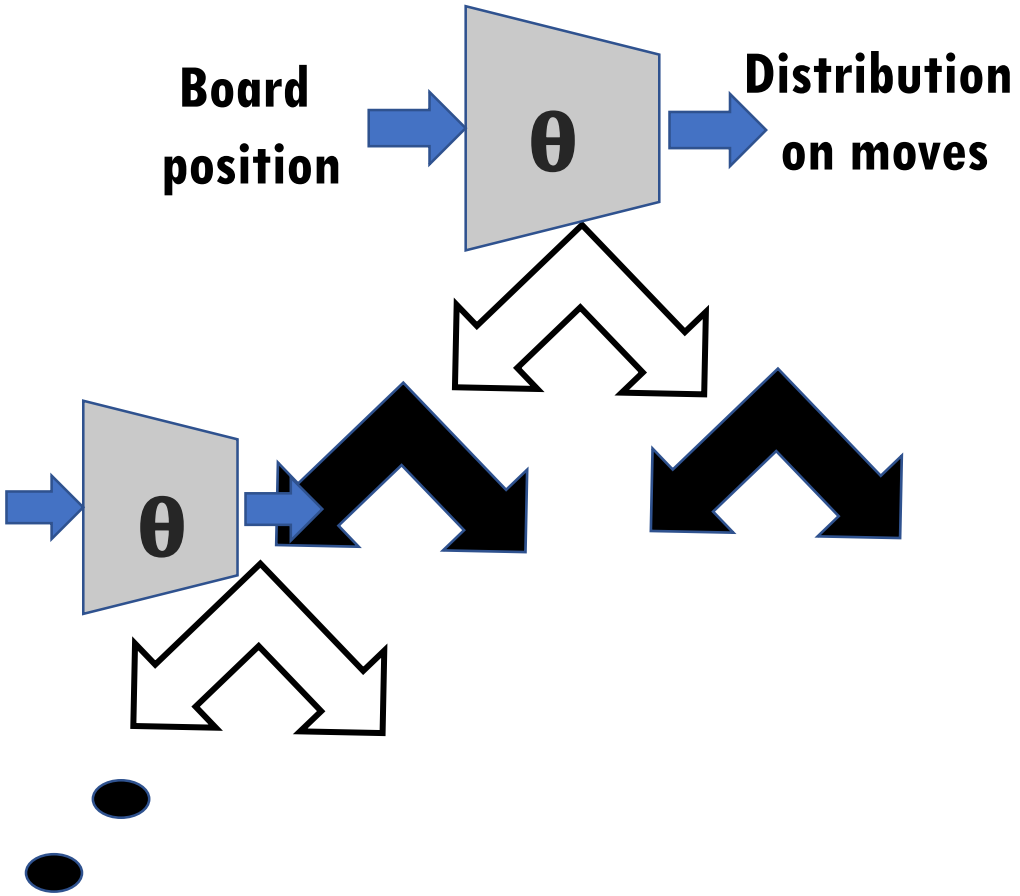
list of handcrafted rules

+

move evaluation algorithm
(approximate; limited lookahead)



Game-playing via Deep Learning (crude account of Alpha-Go Zero)



Use move evaluation algorithm
to evaluate $\ell(\theta) = \Pr[\text{loss}]$

Want: estimate of gradient of ℓ in order to improve θ !

Crude method:

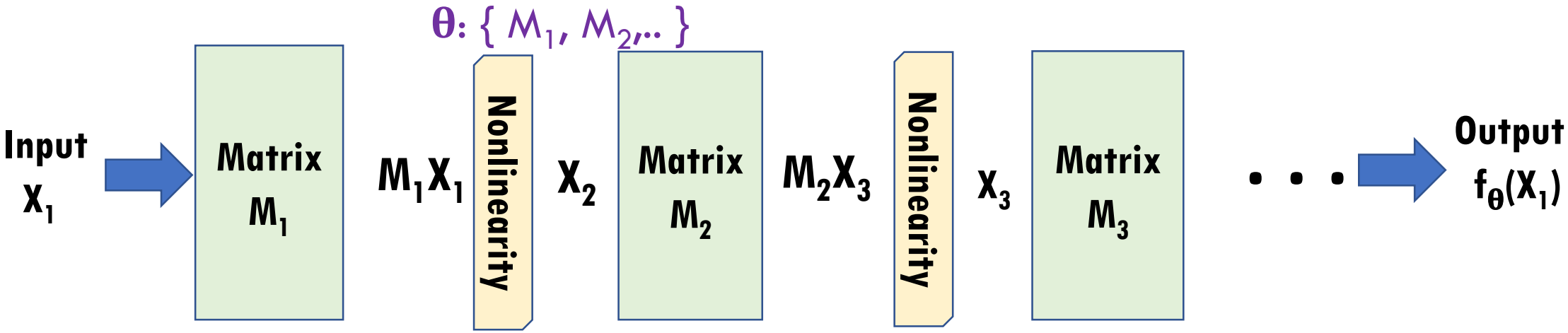
$$\ell(\theta + \eta) \approx \ell(\theta) + \eta \nabla \ell|_{\theta}$$

Can "read off" $\nabla \ell|_{\theta}$ using enough random η's

Part 3

Toward mathematical understanding of Deep Learning

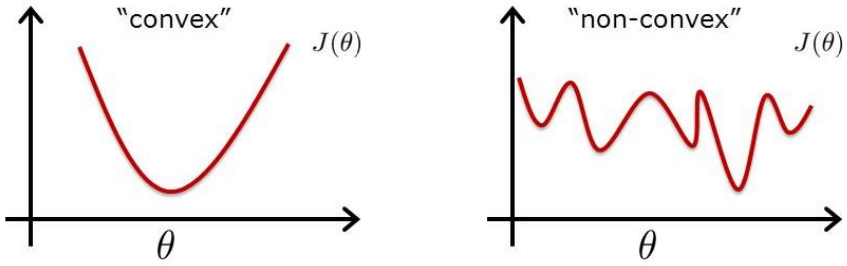
Special case: deep learning (deep = "multilayered")



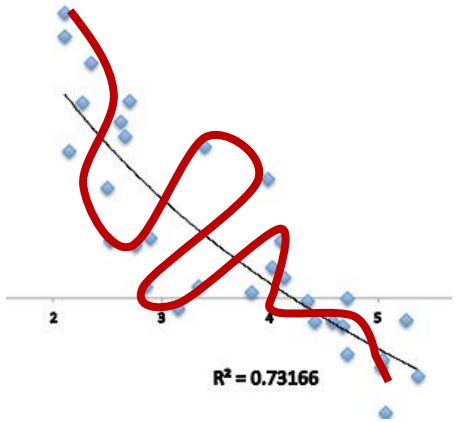
"Nonlinearity": Given a vector, output same vector but negative entries turned to zero.

Training data: $\{(X_1^{(i)}, Y^{(i)}): i=1,..,N\}$ $\ell(\theta) = \sum_{i=1}^{N} (f_\theta(X_1^{(i)}) - Y^{(i)})^2$

Some key questions



- Why/when does gradient descent work and how fast? (Nonconvex loss!)
- Why deep (and not shallow)?



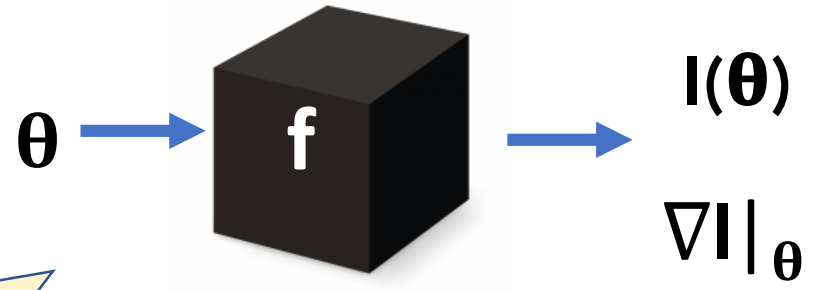
- Why doesn't training overfit to training data? (# parameters >> # training samples). Current deep models capable of achieving zero loss on random data [Zhang et al'17])
- How to interpret the trained model's inner workings?

Analysis of optimization

Hurdle 1: Nonconvex optimization is NP-hard.
No efficient algorithm if P ≠ NP.)

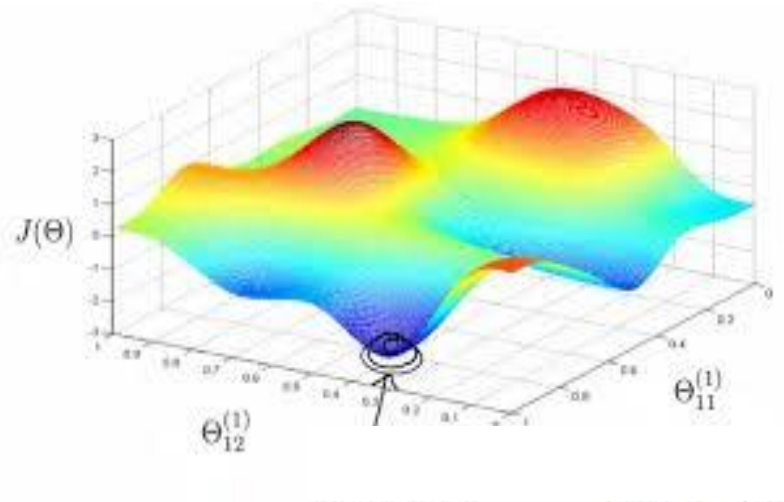
Hurdle 2: Loss $\ell(\theta)$ is essentially a black box to us.

$$\ell(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(X_1^{(i)}) - Y^{(i)})^2$$



Lack mathematical description of data $(X^{(i)}, Y^{(i)})$.
"What makes a bunch of pixels an image of a dog?"

Black box analysis (sketch)



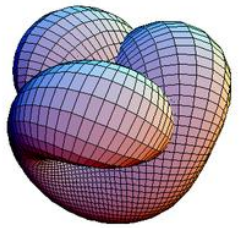
More complicated analyses shows Gradient Descent quickly finds "2nd order local minima" [Ge et al.'15]. (Bottom of valley)

- $\nabla \neq 0$ ➔ $\exists$ descent direction
- But if Hessian (∇^2) "large", allows ∇ to fluctuate a lot!

  ➔ To ensure descent, take small enough steps determined by smoothness (norm of $\nabla^{2)}$
- Guaranteed to get close to point with $\nabla \approx 0$ (speed determined by norms of ∇^2 and ∇). "Stationary point"

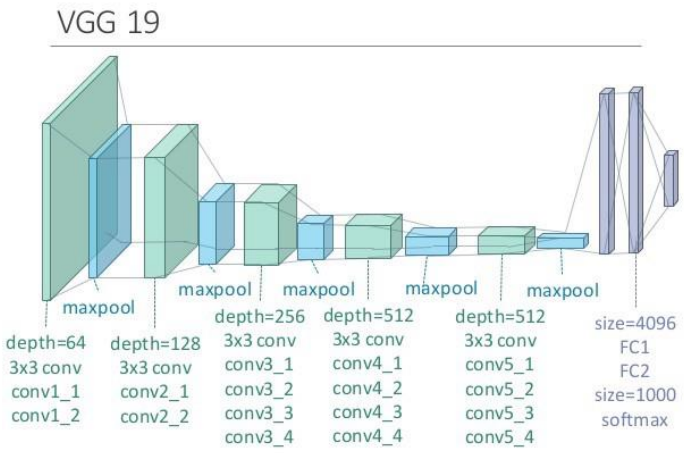
More about optimization in next talk, including recent works using **trajectory** analysis for gradient descent (somewhat non black box)

Why no overfitting? (Even for models with 20M parameters trained on 50k samples)



Popular conjecture: When trained on realistic data, the net's parameters are constrained ---by problem and/or training ---- to be highly interdependent (e.g., lie on manifold of much lower dimension)

Next talk: A partial realization of this suggestion



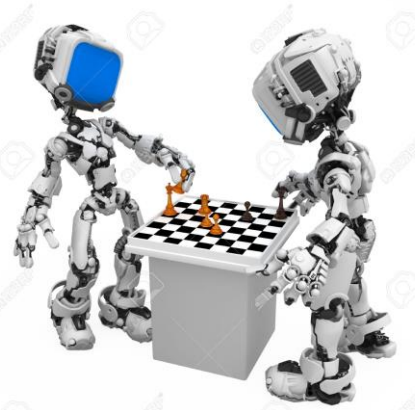
Properly trained nets have "noise stability" property. We prove theorem showing this implies their parameters lie in a lower-dimensional subspace

Part 4

Taking stock, wrapping up

“Mindless” model-fitting…
None of this is remotely how humans think!

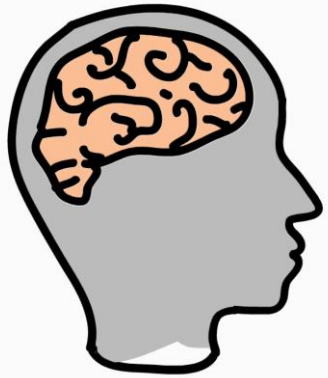
1. Imitation approach has not worked well in the past: airplanes, chess/go etc.



2. Machines’ advantage lies precisely in ability to crunch data.

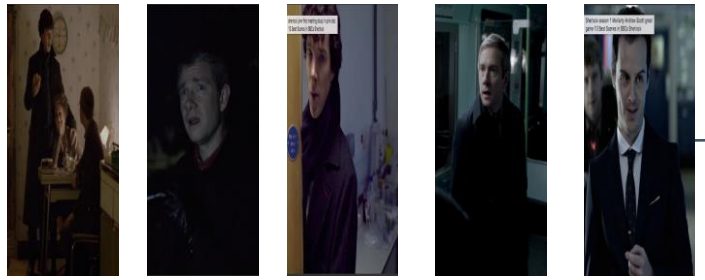
3. We have little idea at an operational level how humans think.

In fact, machine learning is currently the best hope for figuring out how humans think...



Sample Task: "Decoding" Brain fMRI [Vodrahalli et al, NeuroImage'17]

Movie scenes



Annotations of movie scenes

Sherlock and John talk about the murder in an old room with Mrs. Hudson.

John is worried as Sherlock runs off.

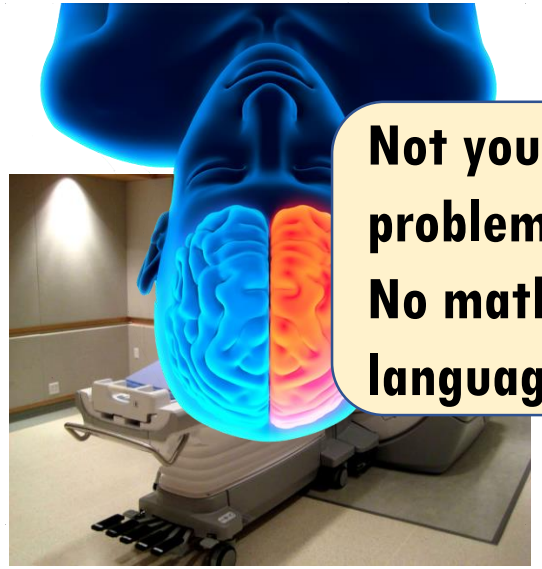
Sherlock enters the door to the chemistry lab saying "John, I was here the whole time."

Once they get on the subway, John exclaims, "No you weren't!"

Moriarty arrives and says, "Hello Sherlock, John."

"Decode??"

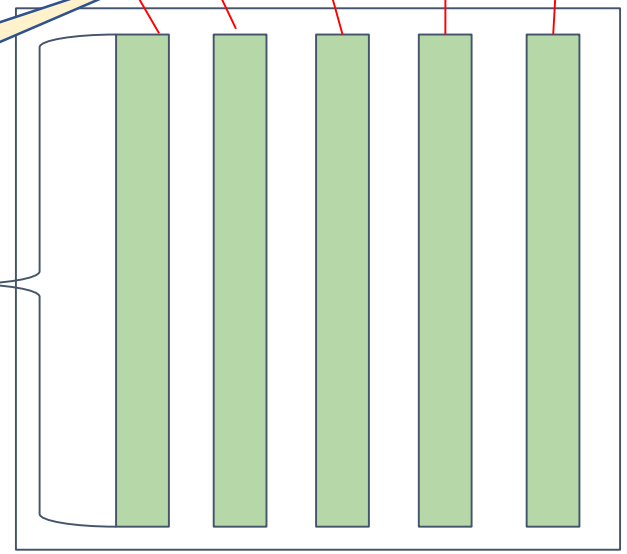
Each movie scene paired with text description fr external



Not your typical applied math problem (e.g., x-ray tomography). No mathematical model for language!!

fMRI Machine

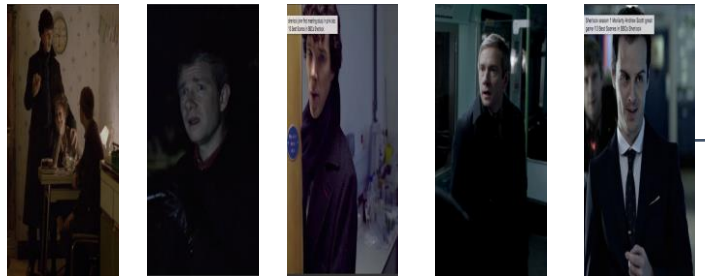
Voxel vector per time step $\in R^{50,000}$



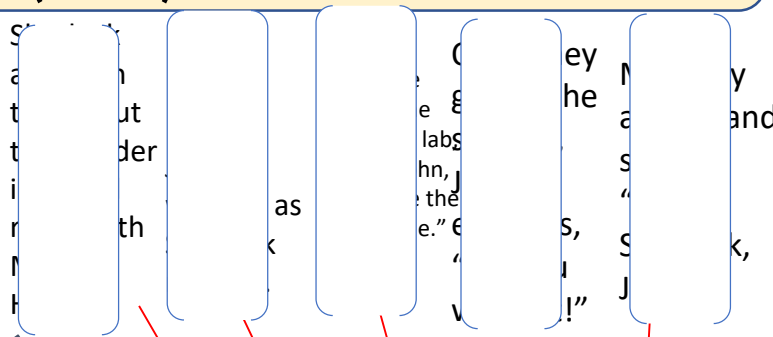
fMRI responses

Sample Task: "Decoding" Brain fMRI [Vodrahalli et al, NeuroImage'17]

Movie scenes

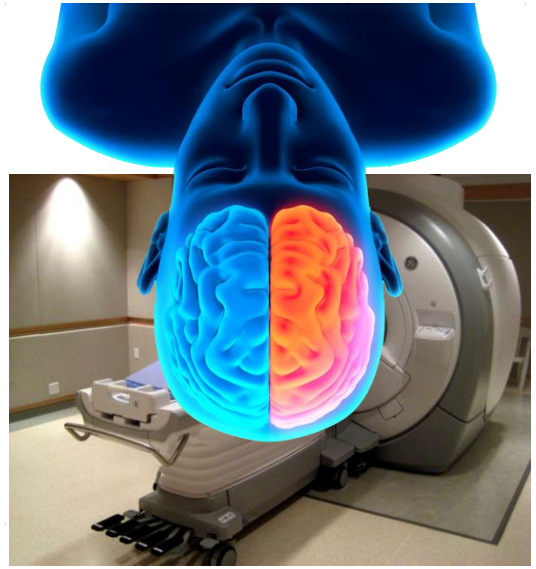


semantic embeddings of text ("SiF') from [Arora et al'17]



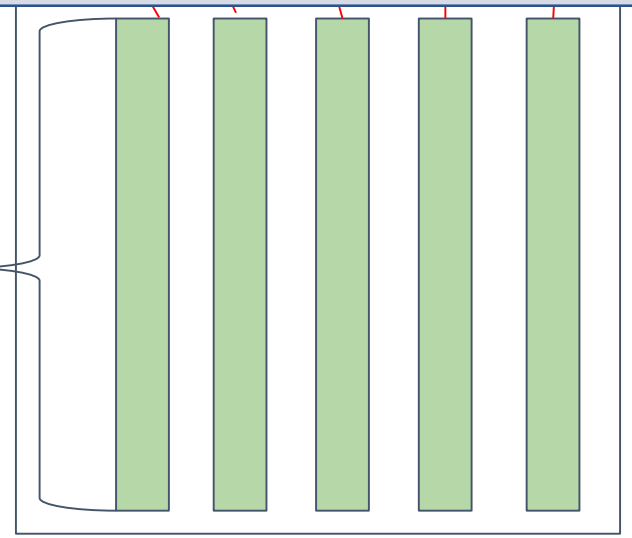
Learnt" Linear transformation"

Each movie scene paired with text description from external party.



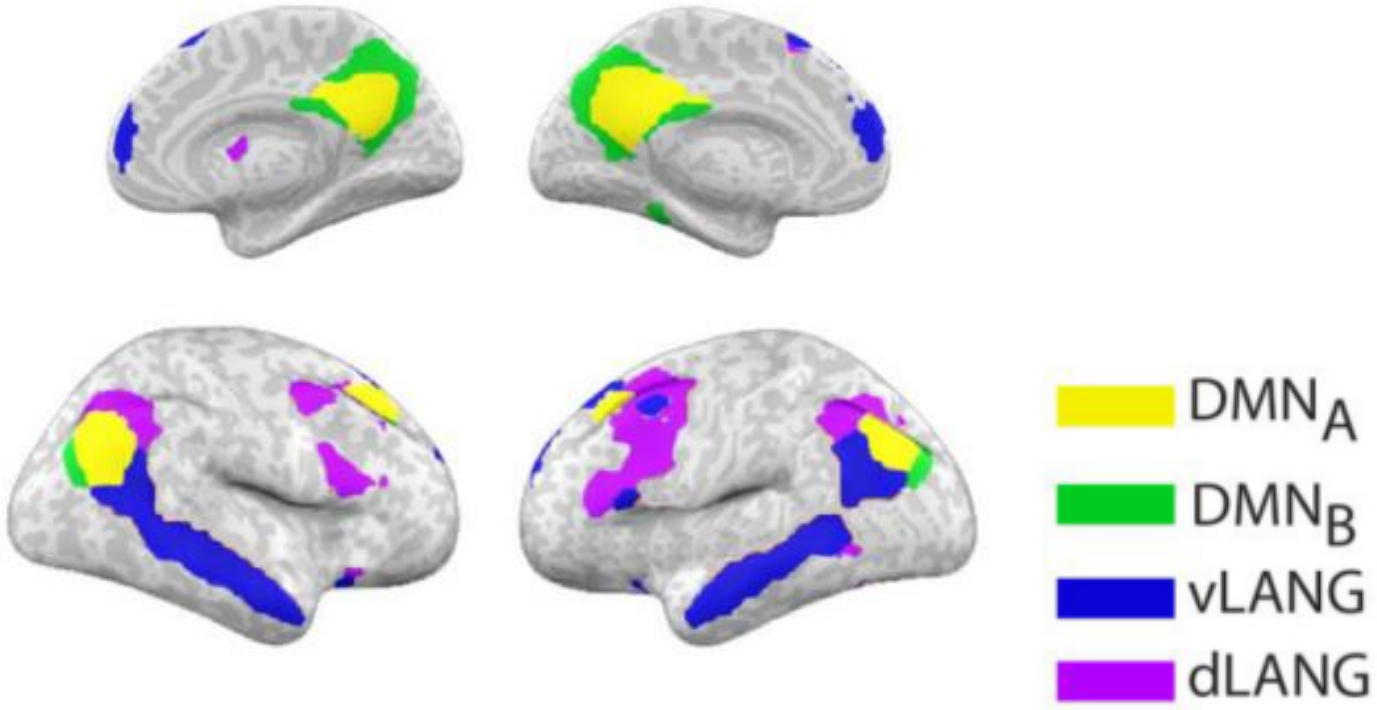
fMRI Machine

Voxel vector per time step $\in R^{50,000}$



fMRI responses

Brain regions useful for decoding



(Hottest trend in academia: ML + Field X; X = Neuroscience, Biology, Social Science,…)

Can Machine Learning thrive in India?

ML advances are driven ---including at leading tech companies---by armies of PhDs.

Top Indian ugrads don't seem to go for PhDs. (Dramatic difference compared to China as well as India itself in the past)..

Concluding thoughts on ML

What is Learning?
What does it mean to understand the world?

- Speaks to age-old wonders/mysteries
- New way of looking at the world (via complicated inexact descriptions)
- A new frontier for science and math
- I am optimistic that deep learning methods can be mathematically understood and/or simplified.

THANK YOU!!

(Thanks for feedback on slides: M. Goresky, A. Ionescu, J. Kollar, P. Sarnak, T. Spencer A. Wigderson.

Advertisements

http://unsupervised.cs.princeton.edu/deeplearningtutorial.html

Come join special year at Institute for Advanced Study 2019-20 http://www.math.ias.edu/sp/

Resources: www.offconvex.org

Grad seminar (hope to put all notes online soon) http://www.cs.princeton.edu/courses/archive/fall17/cos597A/ (another version Fall'18)