

# IFT 6085

## Theoretical principles for deep learning

Winter 2020

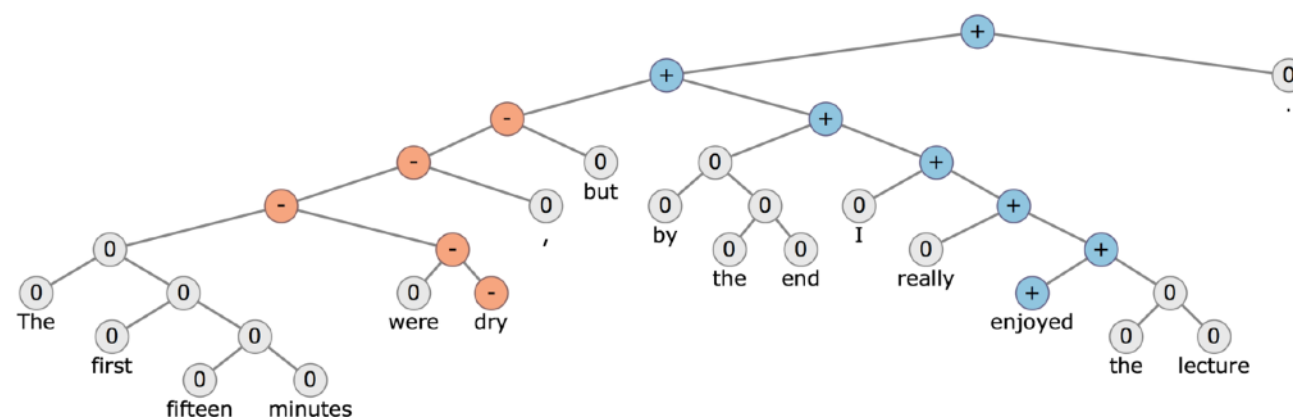
Instructor: Ioannis Mitliagkas

# Today [overview mostly]

- Why deep learning?
- How do we make it work?
- Why **does** it work? We don't fully know
- Class goals: learn tools, do research, present well
- Summary of content
- Class logistics
- Short quiz (not graded :) )
- Questions

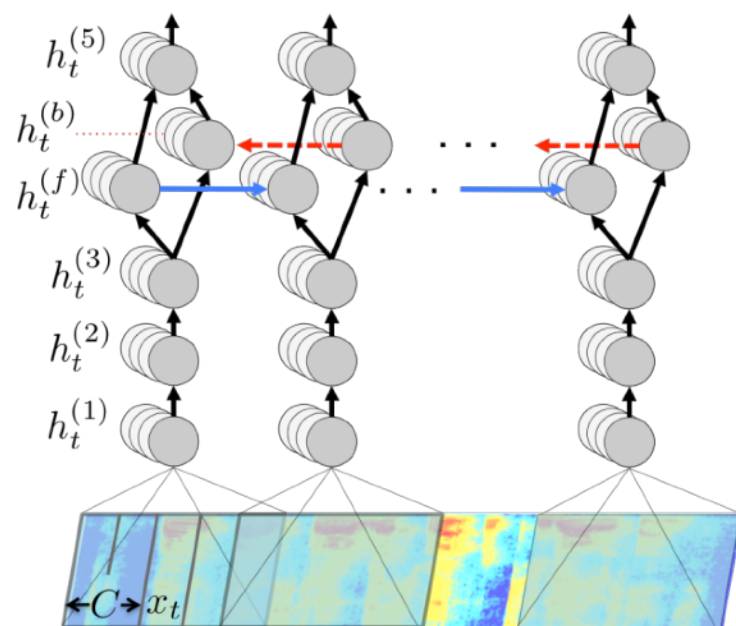
**Deep learning drives  
significant progress  
in modern ML/AI**

# NATURAL LANGUAGE PROCESSING



cs224d.stanford.edu

# SPEECH RECOGNITION



# COMPUTER VISION

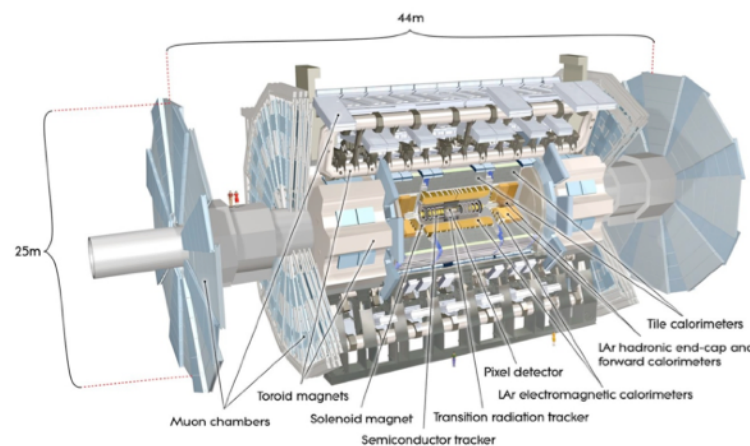


## CORI PHASE II

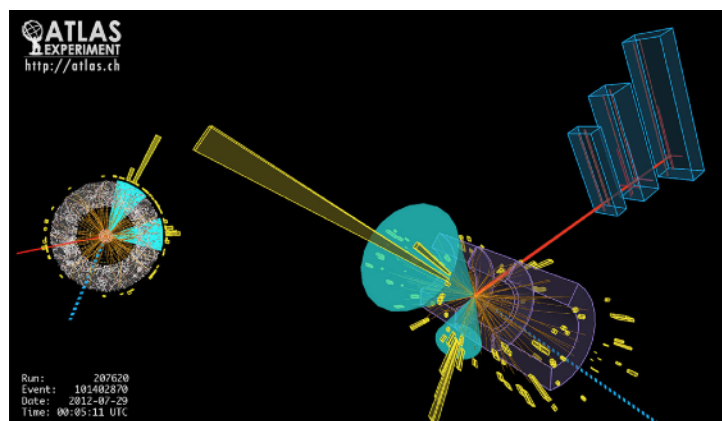
- ▶ 9600+ Knights Landing nodes



## HIGH-ENERGY PHYSICS (HEP)

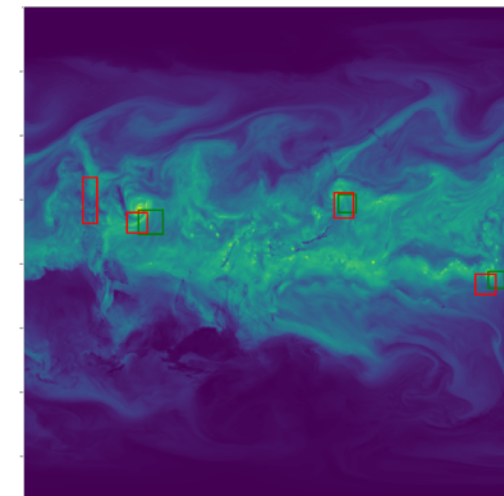


[atlas.ch](http://atlas.ch)

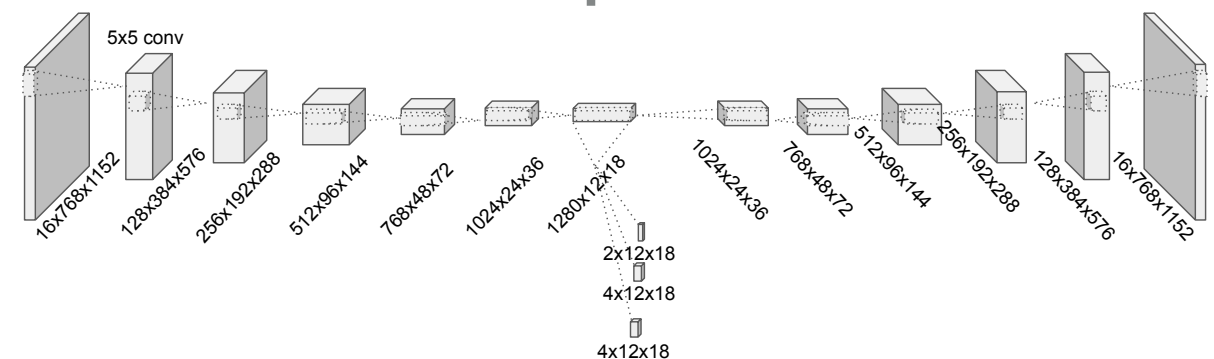


7.4TB data

## CLIMATE SCIENCE



15TB data  
semi-supervised

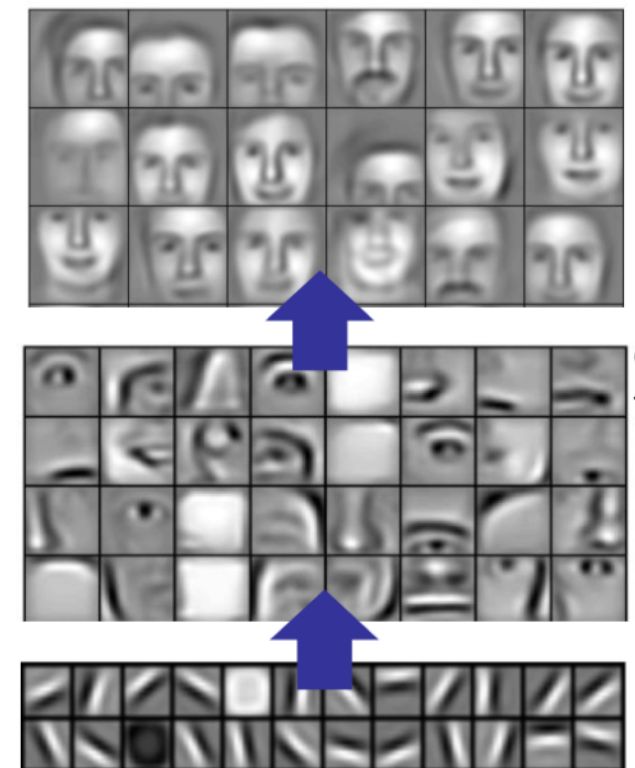
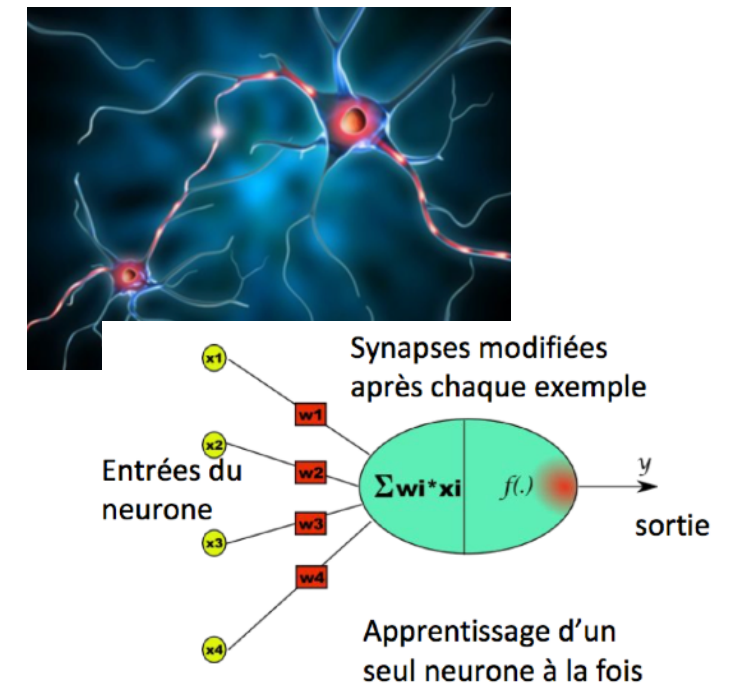


**Economy-altering  
potential**

**How do we achieve  
this great  
performance?**

# Knowledge from decades of research

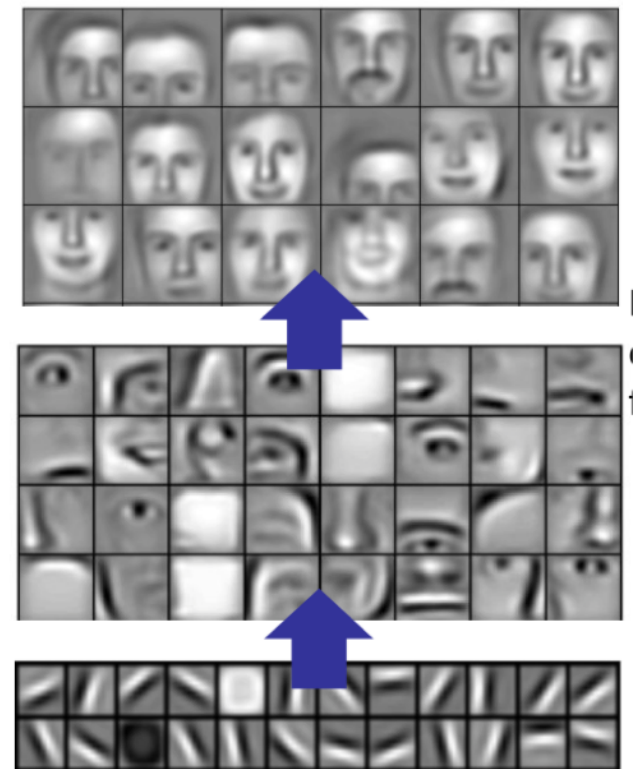
- Perceptron [Rosenblatt, 1957]
- [skipping people here!! not meant as a complete history]
- Progress in the 1980s-1990s
  - Bengio, Hinton, LeCun
  - Schmidhuber
- Took off again (seriously) in the 00's
- CIFAR-funded program gave new life to area





# Recent boom

- We have more data
- We have more computational power
- We have improved our techniques (though they're not brand-new)



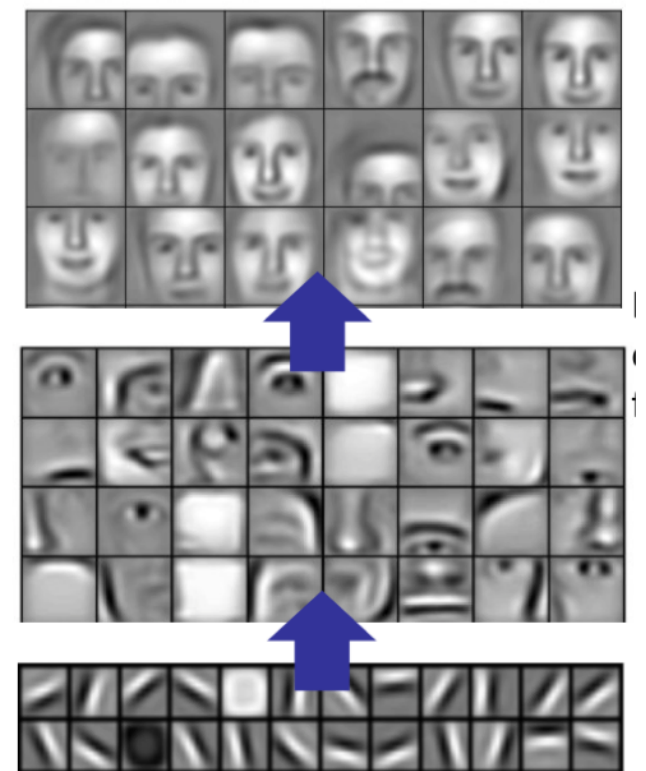
# Making things work

- Good research labs and big companies know how to make deep learning systems work
- MSc/PhD here is great way to pick up skills  
—> very valuable in industry
- Important announcement:  
**Professional MSc in ML:**
  - 2 extra classes instead of research project
  - MILA staff arranges/oversees internship on final semester
  - Can switch within 1-2 semesters. Email Linda Peinthière ([lpeinthiere.umontreal@gmail.com](mailto:lpeinthiere.umontreal@gmail.com)) if interested.



# Driven primarily by intuition and empirical success

- Good research and progress based on solid intuition
- **Practice leads** the way
- **Theory lags** dramatically
  - no guarantees
  - little understanding of limitations
  - limited interpretability
- More interestingly, classic theory suggests currently successful DL practices, wouldn't be likely to succeed.



**Why does deep  
learning work?**

**We do not fully  
understand**

**==**

**Research opportunity**

# This class

- Seminar-style: we go over recent papers
- We go over recent **theoretically-driven or theoretically-supported advances in deep learning**
- We cover different topics, but try to tie them under common themes
- With every opportunity we study some underlying theoretical tools.
- Students read and present papers, and work on a semester research project

# This class

- Will not teach you machine learning
  - You must have passed ML class first
- Will not teach you deep learning
  - You might be ok without DL class if you took ML
- Will not teach you math
  - You must have taken math classes before

# Goals of the class

- Exposure to useful theoretical tools
- Motivate you to apply your math skills
- Read many papers
- Engage in research
- Practice good presentation skills
- **ALL ARE VERY IMPORTANT**



# Main areas/topics of focus

- Optimization
- Information theory
- Statistics and Generalization
- Generative models
- Expressivity of deep architectures

**Who is this class  
for?**

# Advanced grad students

- If you are a **first/second-semester MSc** student this class is not good for you.
- Assumes solid knowledge of machine learning, and understanding of deep learning models
- Heavy focus on mathematics

# Prerequisites I

- Linear algebra
  - vector and matrix norms
  - singular value decomposition
  - eigen-decomposition, change of basis
  - spectral radius vs operator norm

# Prerequisites II

- Basic probability
  - Probability spaces
  - Basic distributions (Bernoulli, Gaussian, exponential...)
  - Basic statistics: mean, variance, ...
  - Basic concentration bounds:
    - union bound, Markov inequality, Chebyshev...
    - [We'll likely cover Chernoff bounds in class]

# Prerequisites III

- Machine learning/deep learning
  - Graduate class in ML/DL
  - the basic workflow of supervised learning (training/validation/test splits, evaluation ...)
  - composing and training basic ML models in PyTorch/TensorFlow/Keras...
  - **having read a few ML papers**

# “Should I take this class?”

- If you can't wait to start doing research do it!
- This class is not necessary if you want to:
  - Be a successful practitioner
  - Do more applied research
- **Quizzes** on every lecture will help us with assessment
- You can switch within the first couple of weeks to avoid fees (\*\*please double check)

**What are we going to  
achieve?**



# Calibrating expectations: tiny victories

- Deep learning theory is **hard**
- Researchers are extremely interested in it, but struggling to provide general results
- Many interesting results depend on strong assumptions
  - e.g. ‘for a class of objectives all local minima are global minima **if the data is Gaussian**’  
[Ma et al. 2017]
  - or a study of the expressivity of neural networks **with random weights** [Poole et al. 2016]
- Still, even this kind of theory is much-needed progress!

# Logistics

# Logistics

- Language of instruction
- Grade breakdown
- Class hours
- Office hours
- Auditing policy

# Language of instruction

- International group: many foreign students
- Lectures, notes and quizzes will be in english

# Grading

- Participation 5%
- Scribing 10%
- **Daily quizzes 15%**
- **Early in-class exam 5%**
- Midterm 15%
- Paper presentations 15%
- Research project 35%

# Scribing 10%

- Most lectures will be given by me on the board
- 1-2 students each time will be responsible for taking notes
- Deliverable: notes in Latex format one week after the lecture
  - 5% penalty for late notes
- Everyone has to scribe once

# Daily quizzes 15%

- On the reading material of the day
- 5-10 minutes at beginning of lecture, **on your laptop**
- 14-18 quizzes
- Grade computed as average of 8 best quizzes
- All material already available
- First graded quiz tomorrow morning!

# Midterm 15%

- Mostly on crash course material
  - Optimization
  - Statistical learning theory
- DATE TBA (mid-late February)



# Paper presentations 15%

- ~4 lectures will consist of paper presentations by students
  - they will only start after the fourth week (more later)
- Groups of 2 students:
  - read an agreed-upon paper from literature
  - prepare slides
  - present the work in class (20 minute talks)
- Graded based on quality of slides, and clarity of presentation

# Research project 35%

- Groups of 2 students
- Proposal due in the middle of the semester
- Short, in-class progress report (5 minute talk)
- Poster presentation (date TBD)
- End of semester report (date TBD)

# Research project 35%

- Topics (I will release list of suggested topics):
  - Optimization
  - Generalization
  - Representation
  - Generative models
- Chosen based on:
  - Your own research interests (as aligned with the class)
  - Lists of papers I will be making available as we're covering the topics

# Research project 35%

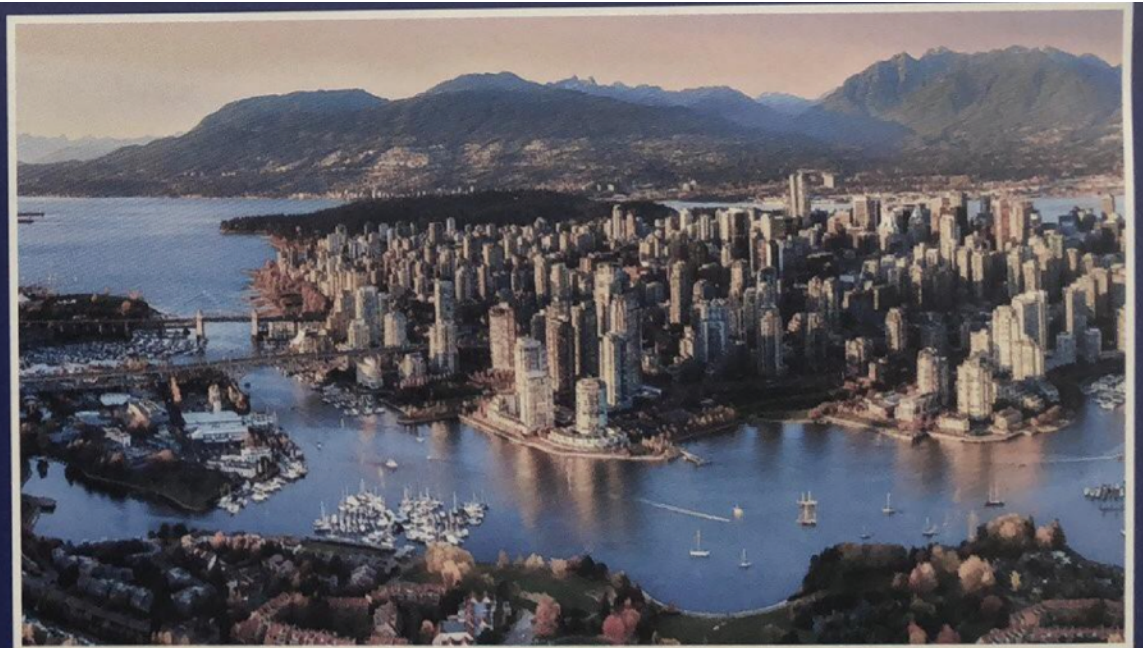
- Types of projects
  - Comprehensive literature review of selected topic, with careful presentation of relative merits of different methods and ideally performing simple experiments.
  - Application of ideas seen in class on your own research or other problem you find interesting.
  - Focusing on the math/analysis of existing or proposed methods.
  - Demonstrating limitations (via theory/experiments) of existing work
    - proposing solution
    - demonstrating simple prototype of new idea or analysis on a simplified setting (toy model)

# Research project 35%

- The ideal project ==> NeurIPS submission
  - ambitious and difficult goal but worth it
  - not required to do well in class!

**FUTURE  
CONFERENCES**

**Vancouver, Canada  
2019 & 2020**



# Class hours

- Wednesday 9:30-11:10
- Thursday **9:00**-10:40

# Office hours

- Talk to me after class
- If needed, we'll amend

# Communication

- Email is bad idea
  - Helps if you clearly label subject: “IFT6085: ... “
- Slack
- Studium for announcements



# Auditing policy

- You're free to sit in!
- As long as we have enough seating for registered students
- Interested in working on a project along with the class? Maybe we can accommodate. Come talk to me.
- Google group

**A theme:**

**When ML theory  
breaks down**

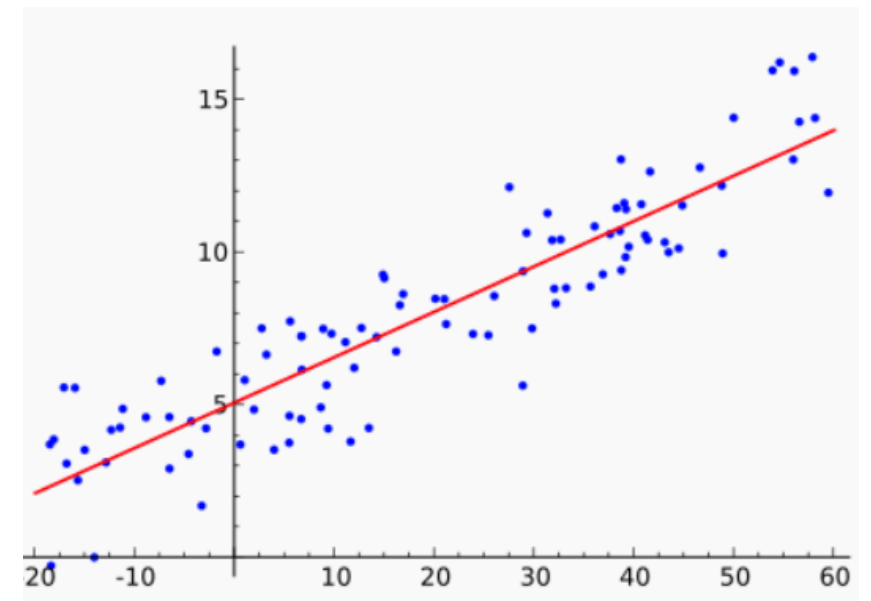
**Machine learning is  
rigorous**

# Logistic regression

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Classic and very successful tool from statistics
- If you're in this class you've at least heard of it
- Used for classification
- IMPORTANT BASELINE

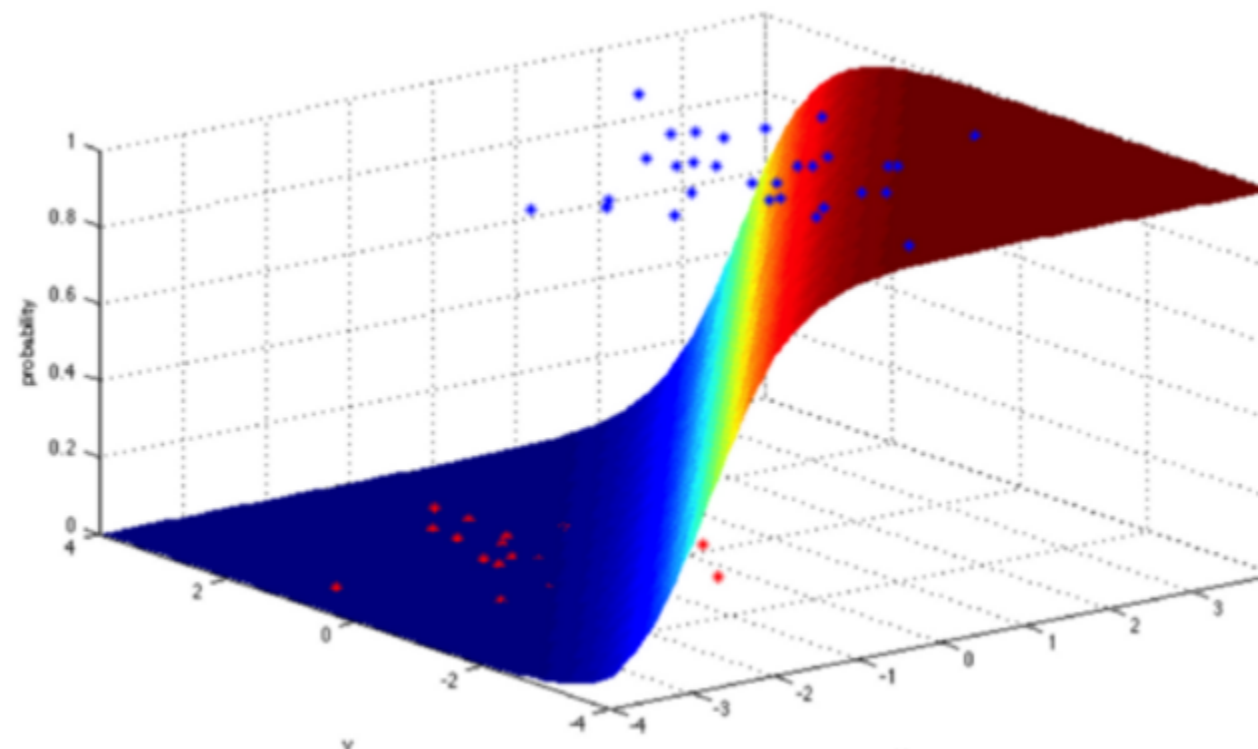


# Logistic regression: 'linear model'

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- **Hypothesis class:** What kind of functions do we represent when we vary the model parameters,  $\theta$ ?

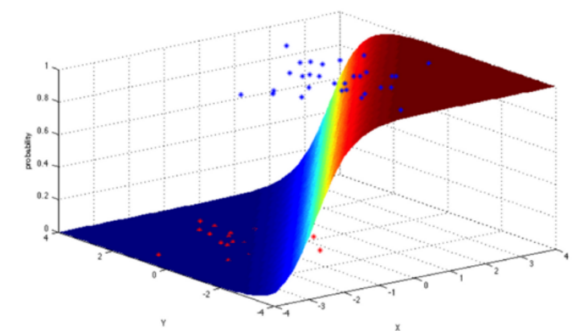


# Logistic regression is interpretable

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Predicted values can be interpreted as probabilities
- Learned coefficients have rigorous interpretation through log-odds

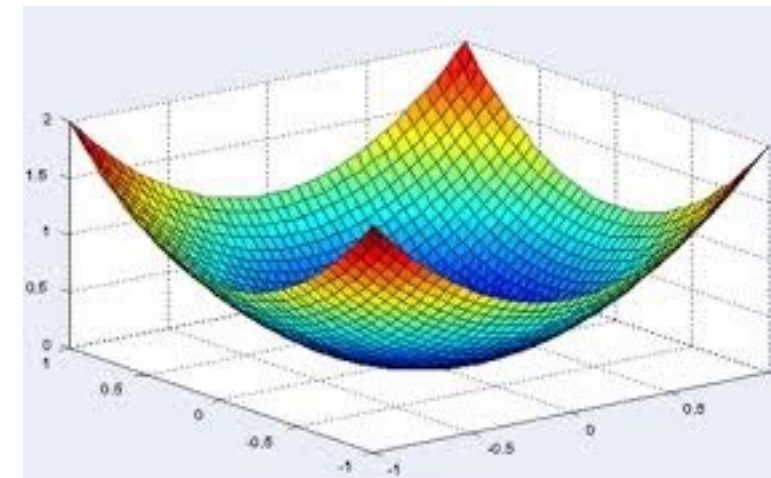


# Logistic regression: convex objective

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Convex objective! link to [proof](#). This means
- It is easy to optimize!
- (Stochastic) gradient descent works wonderfully
- We have convergence guarantees



# Logistic regression generalizes as expected

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- We fit models on the **training set**
- But in the real world they are used on unseen data
- How well do they do out there? (**Generalization**)
- Classic ML bounds are good at predicting the generalization error for logistic regression



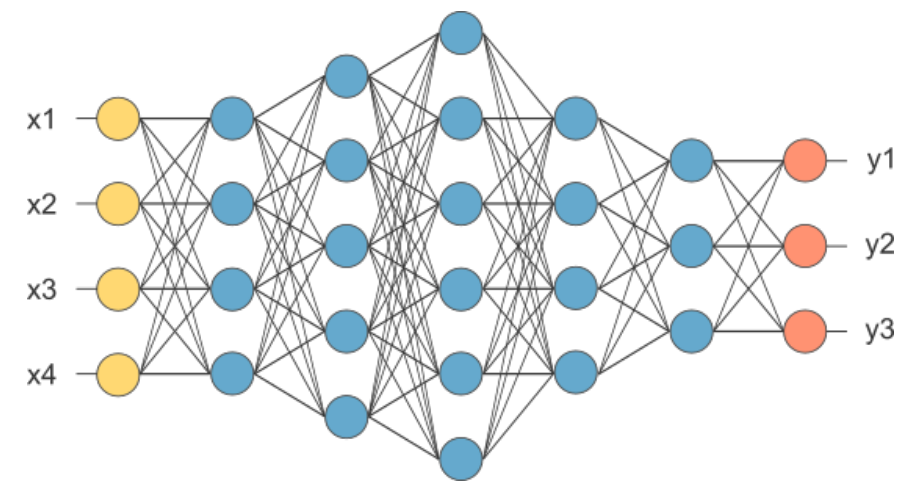
**Deep learning:  
magic?**

# Deep neural networks

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Softmax output (related to logistic regression's logit)
- Multiple layers of non-linearities!
- Very powerful
- State of the art



# Deep neural networks: hypothesis class?

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_\theta(x^i))] + (1 - y^i) [-\log(1 - h_\theta(x^i))] \quad (1)$$

$$\text{where } h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

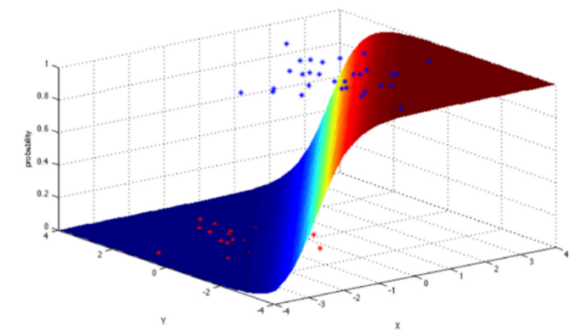
- **Hypothesis class:** What kind of functions do we represent when we vary the model parameters,  $\theta$ ?
- **Universal approximation:** single hidden layer with infinite neurons can approximate any function\*\*
- More generally, we don't exactly know.

# Deep neural networks: interpretability

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Why is our deep learning model coming up with this prediction?
- We don't exactly know how to attribute outputs to inputs like logistic regression
- With some notable exceptions
- Active area of research!

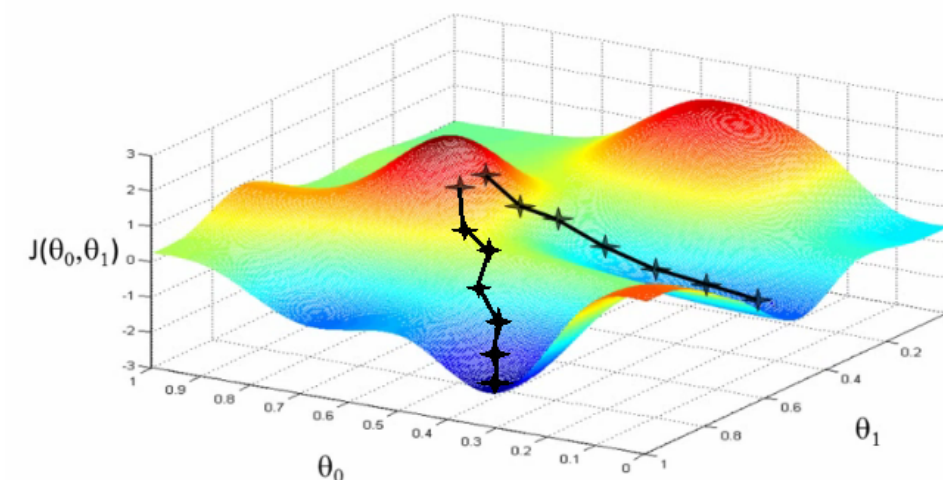
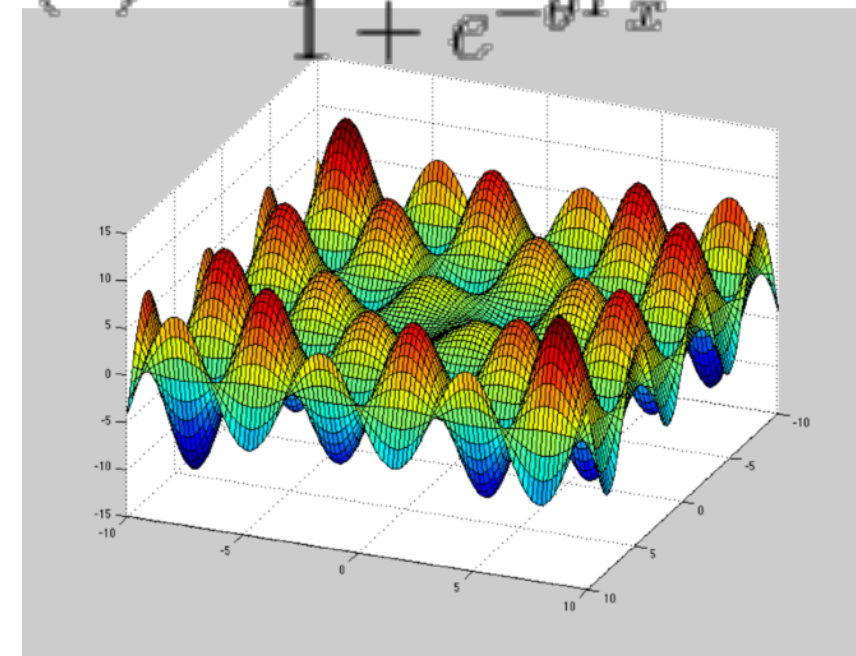


# Deep neural networks: non-convex objective

$$J(\theta) = \sum_{i=1}^m y^i [-\log(h_{\theta}(x^i))] + (1 - y^i) [-\log(1 - h_{\theta}(x^i))] \quad (1)$$

$$\text{where } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Non-convex objective!
  - All bets are off
  - We have no convergence guarantees
  - Different minima from different initializations
- However, gradient descent STILL WORKS!



# Deep neural networks: generalize against all odds

$$J(\theta) = \sum_{i=1}^n y^i [-\log(h_\theta(x^i))] + (1 - y^i) [-\log(1 - h_\theta(x^i))] \quad (1)$$

$$\text{where } h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Classic generalization bounds suggest that to get good generalization in DL, we should be using 10x or 100x the data we are actually using.
- What is going on?
- One of the most interesting questions right now in DL.

**First part of course:**

**Some classic results**

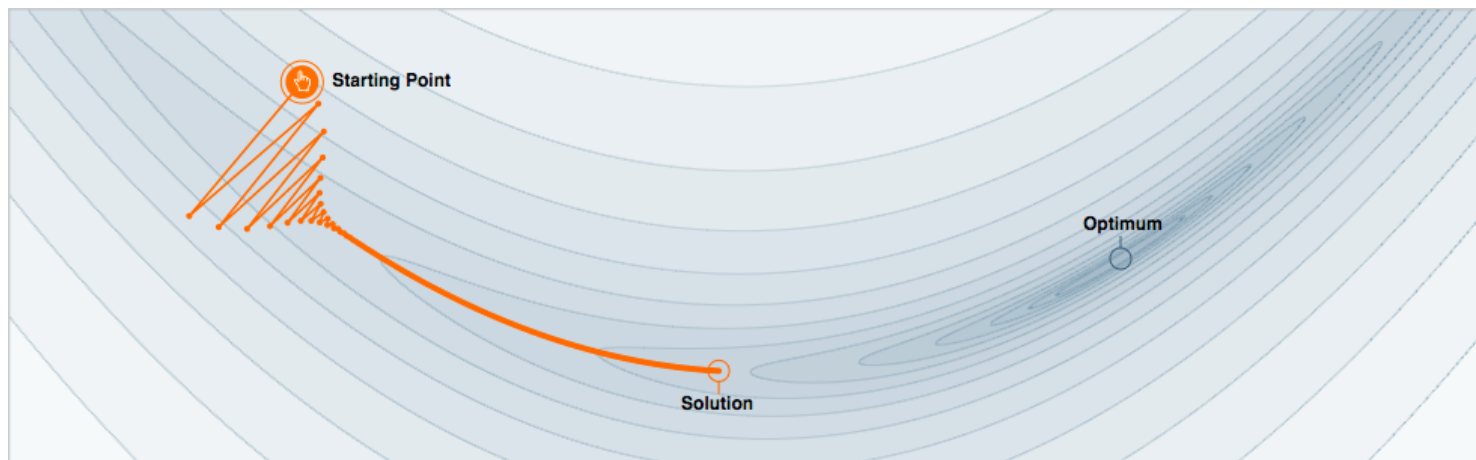
# Crash course in optimization



# GRADIENT DESCENT AND MOMENTUM ALGORITHMS

$$w_{t+1} = w_t - \alpha \nabla f(w_t)$$

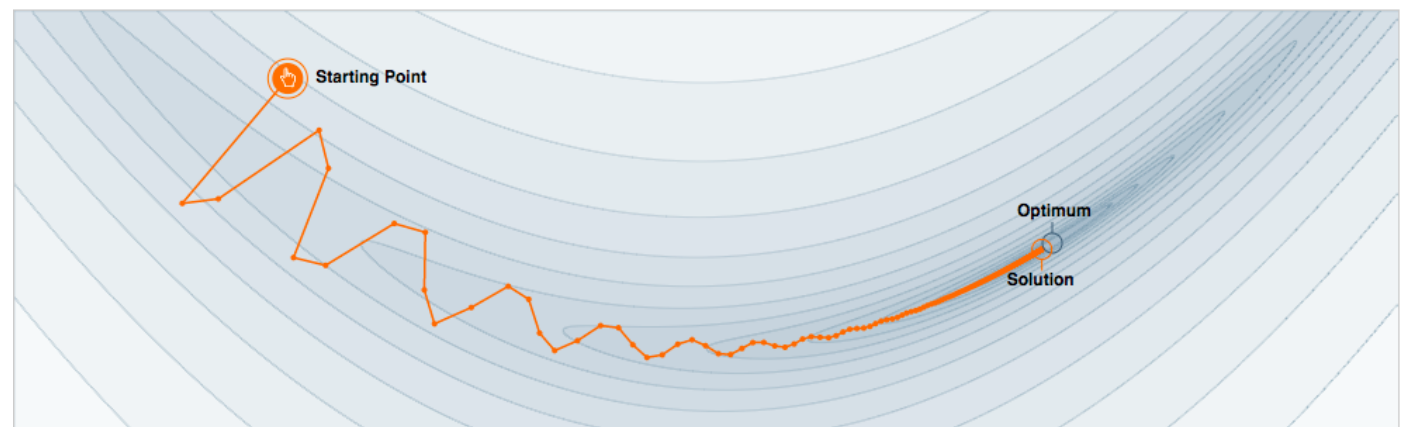
## Without momentum



## With momentum

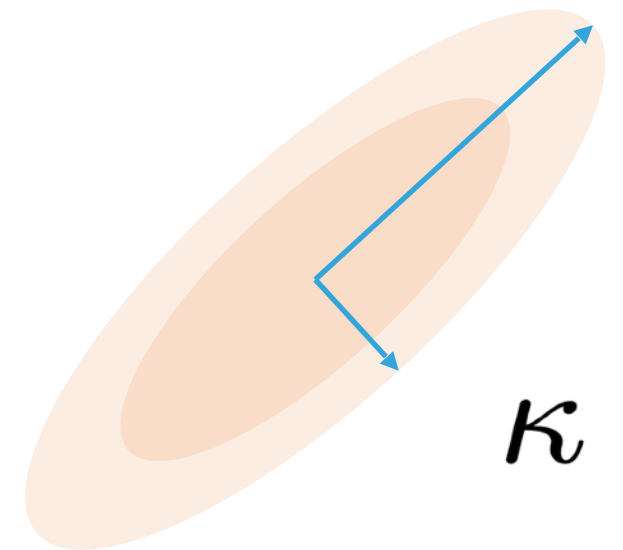
[Polyak, 1964]

[Distill blog]



## CONDITION NUMBER

Dynamic range of curvatures,  $\kappa$



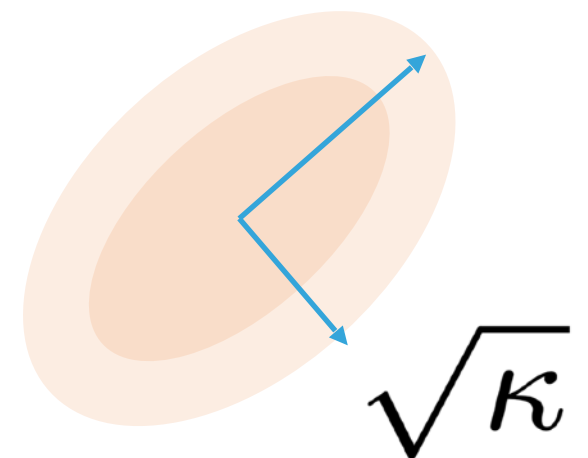
## GRADIENT DESCENT ON STRONGLY CONVEX

Convergence rate  $O\left(\frac{\kappa-1}{\kappa+1}\right)$

## GRADIENT DESCENT WITH MOMENTUM

Dependence on  $\kappa$  changes

$$O\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^*$$



**EFFECTIVELY IMPROVES THE CONDITION NUMBER**

## OBJECTIVE

$$f(w) = \frac{1}{n} \sum_{i=1}^n f(w; z_i) \quad z_i : \text{data point/batch}$$

## GOAL: MINIMIZE TRAINING LOSS

## STOCHASTIC GRADIENT DESCENT

$$w_{t+1} = w_t - \alpha_t \nabla_w f(w_t; z_{i_t})$$

$\alpha_t$  : step size

$i_t$  : batch used for step t

## MOMENTUM

$$w_{t+1} - w_t = \mu_L (w_t - w_{t-1}) - \alpha_t \nabla_w f(w_t; z_{i_t})$$

# **Quick review of basic elements of statistical learning**

# Statistical learning

- Real quick: supervised learning
- Concentration bounds  
     $\implies$  Classic generalization bounds
- VC dimension
- PAC-Bayes bounds

**Main part of course:**

**recent papers**

# Paper topics

- Generalization: theoretical analysis and practical bounds
- Information theory and its applications in ML (information bottleneck, lower bounds etc.)
- Generative models beyond the pretty pictures: a tool for traversing the data manifold, projections, completion, substitutions etc.
- Taming adversarial objectives: Wasserstein GANs, regularization approaches and controlling the dynamics
- The expressive power of deep networks

# Generative models



DISCRIMINATIVE

$$p(y|x; \theta)$$

GENERATIVE

$$p(x|y)$$

$$p(y)$$

Bayes rule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

MODELING ASSUMPTIONS

## DISCRIMINATIVE

$$p(y|x; \theta)$$

Logistic regression  $h_{\theta}(x) = g(\theta^T x)$

Sigmoid function

## GENERATIVE

$$p(x|y) \quad p(y)$$

Bayes rule: 
$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Gaussian discriminant analysis (GDA)

## GAUSSIAN DISCRIMINANT ANALYSIS

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$$

$$x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$$

## CONNECTION TO LOGISTIC REGRESSION

$$p(y = 1|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1 + \exp(-\theta^T x)},$$

Same form as logistic regression, though not exact same decision surface.

Converse not true:

Logistic regression form for  $y|x$  does not imply Gaussian distribution for  $x|y$

**GENERATIVE MODELS MAKE STRONGER ASSUMPTIONS**

## GENERATIVE MODELS

- ▶ Stronger assumptions
- ▶ Better/faster fit when assumptions are correct

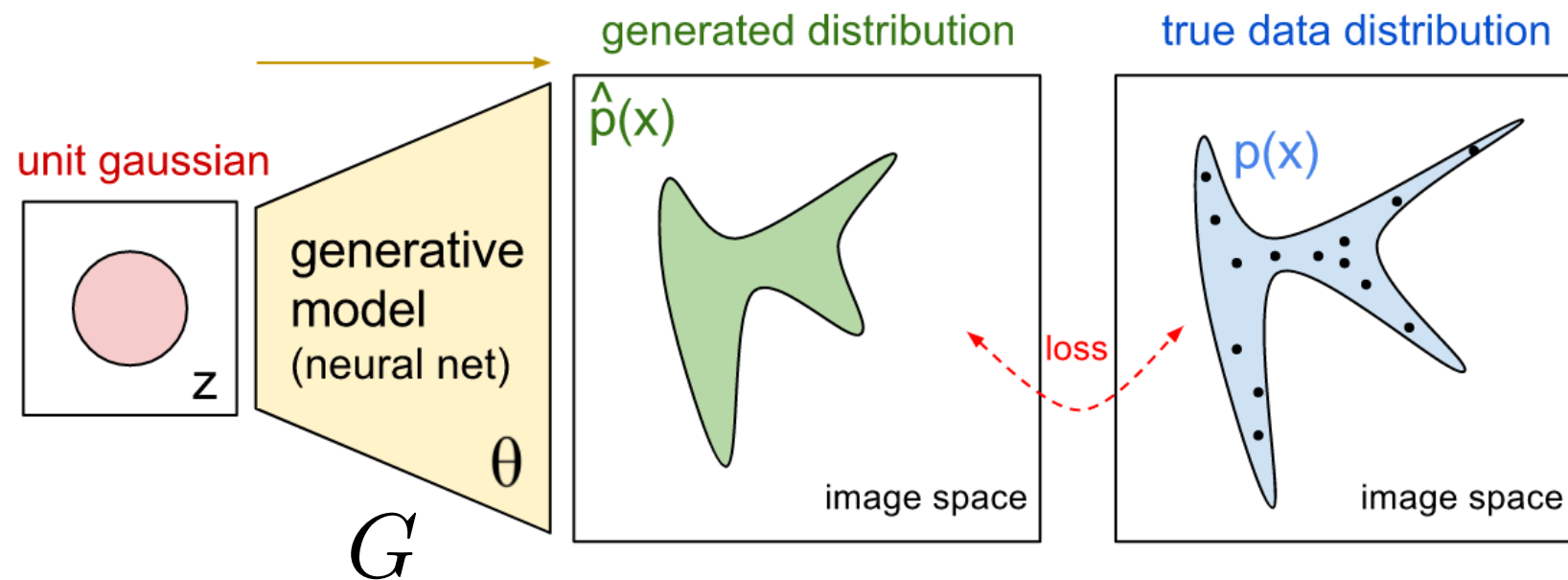
*Asymptotically efficient*

- ▶ Can perform badly when assumptions are bad

## DISCRIMINATIVE

- ▶ Weaker assumptions
- ▶ More robust!!
- ▶ More widely used for classification

# NO MODELING DECISIONS (\*RATHER, HIGHER LEVEL MODELING)



## A FEW APPROACHES TO TRAIN AND REGULARIZE

- ▶ Autoregressive models (PixelRNN)
- ▶ Variational AutoEncoders
- ▶ Generative moment matching networks

# GENERATIVE ADVERSARIAL NETWORKS [GOODFELLOW, 2014]

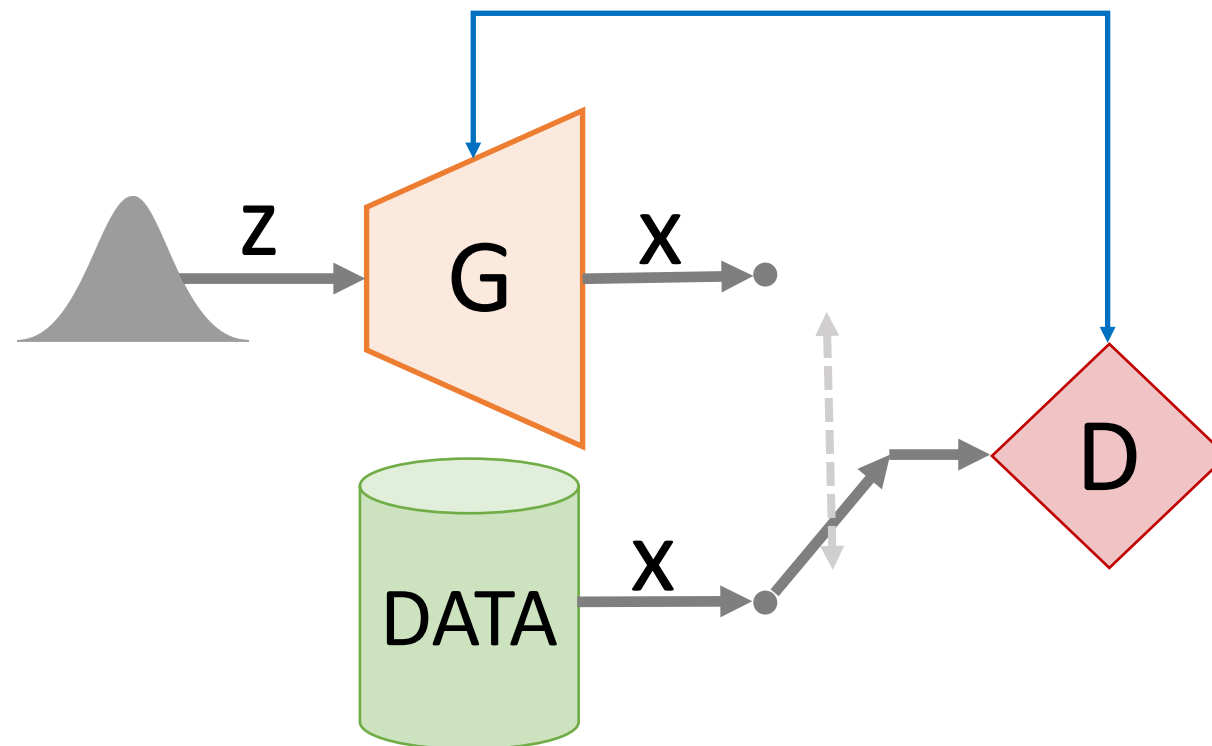
## Generator network, $G$

Given latent code,  $z$ , produces sample  $G(z)$

Both  
differentiable

## Discriminator network, $D$

Given sample  $x$  or  $G(z)$ , estimates probability it is real



# GENERATIVE ADVERSARIAL NETWORKS [GOODFELLOW, 2014]

## Generator network, $G$

Given latent code,  $z$ , produces sample  $G(z)$

Both  
differentiable

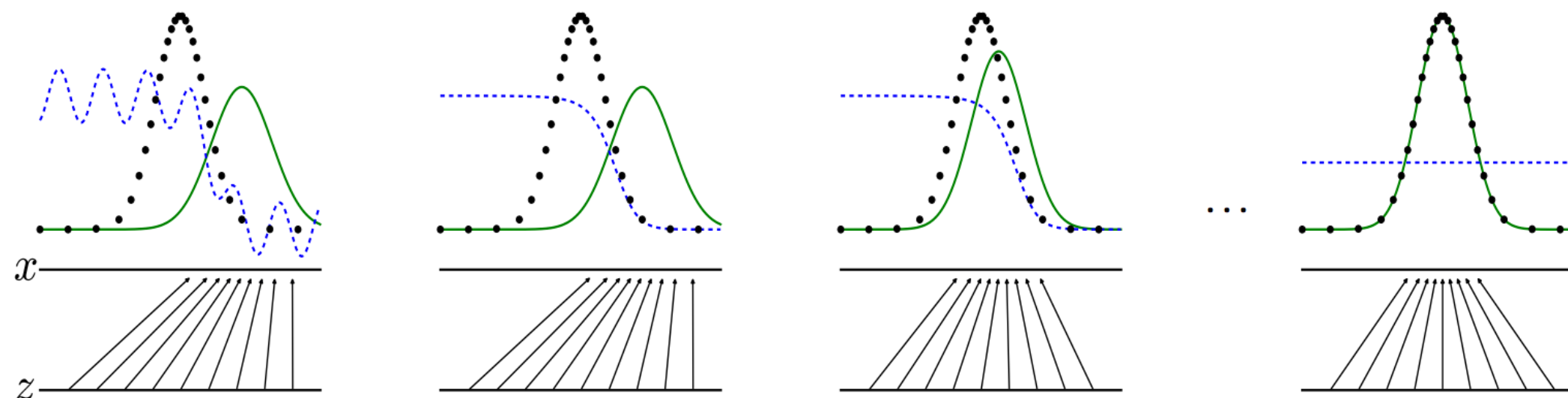
## Discriminator network, $D$

Given sample  $x$  or  $G(z)$ , estimates probability it is real

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

# GENERATIVE ADVERSARIAL NETWORKS [GOODFELLOW, 2014]

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$



## BENEFITS

- ▶ Easy to implement
- ▶ Computational (No approximate inference/no partition function estimation)



## DIFFICULT TO TRAIN

### SATURATION

Gradients become zero

### MODE COLLAPSE

Whole chunks of space can be dropped.

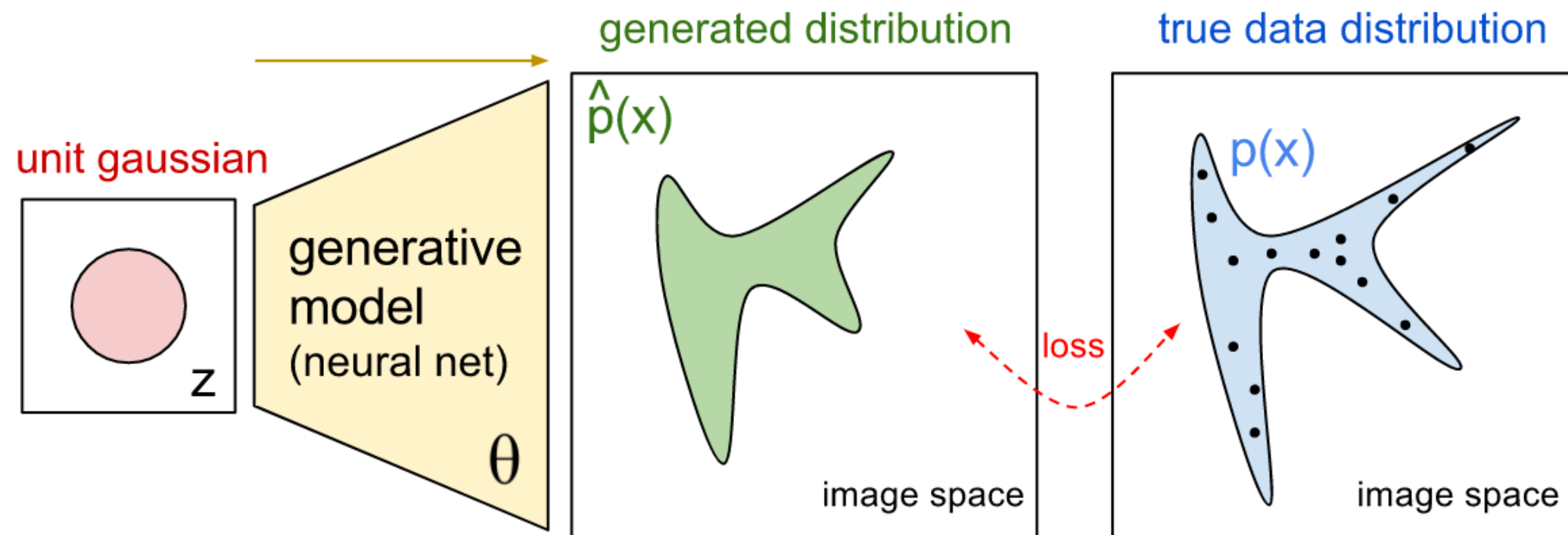
Wasserstein GANs deal with some of those issues

## DYNAMICS OF SADDLE POINT OPTIMIZATION!

Momentum dynamics play important role.

Negative momentum can help.

# GENERATIVE ADVERSARIAL NETWORKS [GOODFELLOW, 2014]



But why?

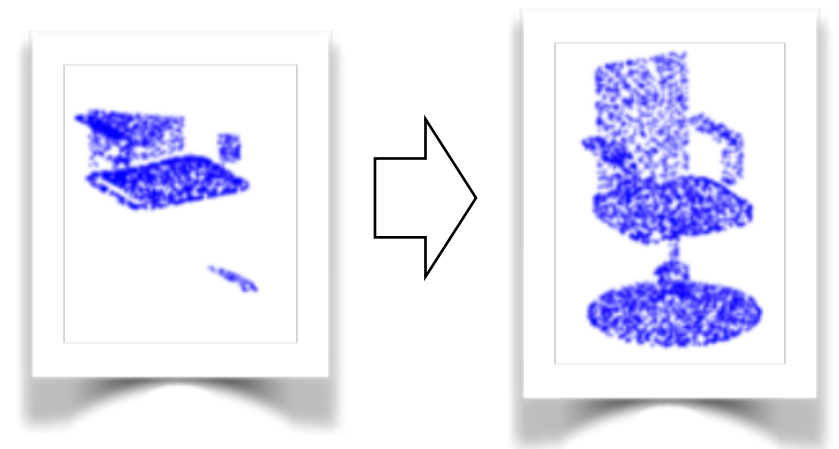
## DREAMING UP STUFF?



*Generated images*

## VERY USEFUL COMPONENT!!

- ▶ Data augmentation
- ▶ Semantic operations/analogies
- ▶ Completion
- ▶ Segmentation...



## BEYOND SPARSITY [BORA ET AL., 2017]

Generator G trained on desirable manifold (faces, 3D objects)

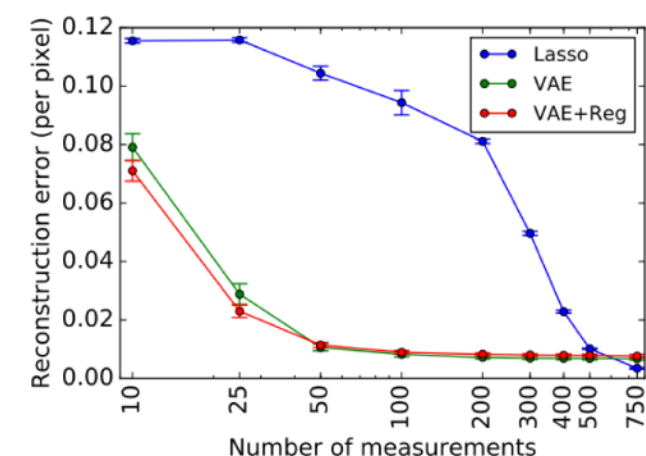
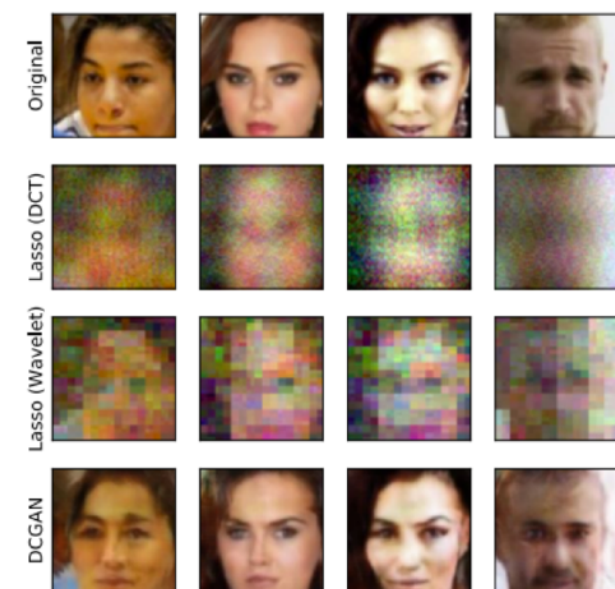
Given sample  $y$ ,  
potentially corrupted by function M

We can 'invert' the generator

$$\min_z \|M(G(z)) - y\|$$

and find a pre-image of  $y$  on the manifold

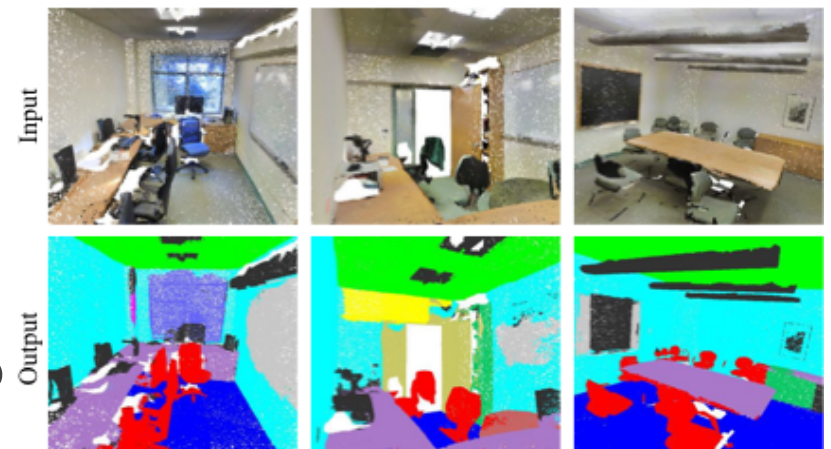
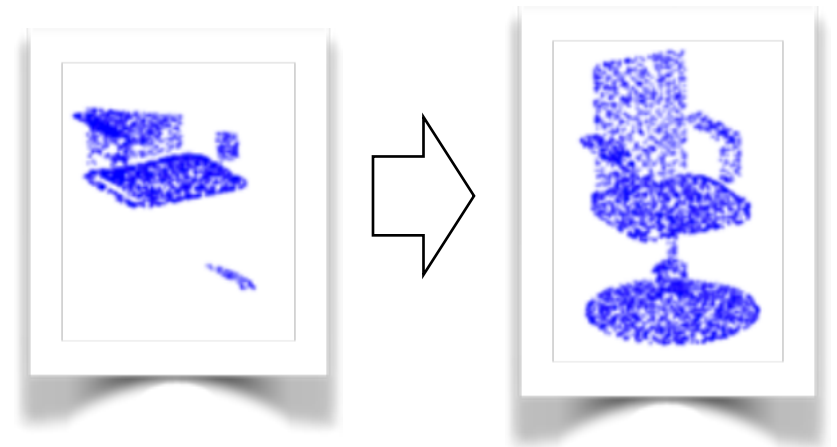
**The trained generative model is critical!**





# INTERESTING QUESTIONS

- ▶ How do we use generative models?
- ▶ How do we evaluate?
- ▶ How do we stabilize adversarial training?
- ▶ How do we reduce mode collapse?



# Resources on website

[mitliagkas.github.io/ift6085-dl-theory-class/](https://mitliagkas.github.io/ift6085-dl-theory-class/)

- Currently contains info about 2018, will be updated soon
- Most course material will remain the same
- First two monographs are a **great** sources of classic optimization and ML resource.
- I will be using them a lot throughout (and assigning some readings from there)

## Resources

1. [Convex Optimization: Algorithms and Complexity](#), Sebastien Bubeck.
2. [Understanding Machine Learning: From Theory to Algorithms](#), by Shai Shalev-Shwartz and Shai Ben-David.
3. [iPython notebook](#) demonstrating basic ideas of gradient descent and stochastic gradient descent, simple and complex models as well as generalization.

# Questions



**Quiz :)**

# First quiz

- **Not** part of grade
- Will allow us to assess the background of the class and adjust material accordingly

# Self assessment

- Did you feel like you knew what most of the quiz questions were talking about?
- Have you seen some of the “I hope you know this”- topics I mentioned in a previous class?
- Can you follow the code and ideas in the iPython notebook listed #3 under ‘Resources’ in the class website?
- Have you read 3 different machine learning papers?