# THEORY OF MACHINE LEARNING

# LECTURE 23

REPRESENTATION LEARNING

# ANNOUNCEMENTS

- **Homework 3 due on Monday April 11**
- **<u>Project presentations:</u>** starting in two weeks! (~18 projects)
  - <u>Dates</u>: April 19, 22, 26 (5 projects /class), couple online

    → Watch out for a signup link.
- This week and next: representation learning, robustness

→ Self-supervision

→ Word embeddings

→ Node embeddings for graphs.

→ adversarial training (corruptions, training ...).

→ mean estimation (basic problem).

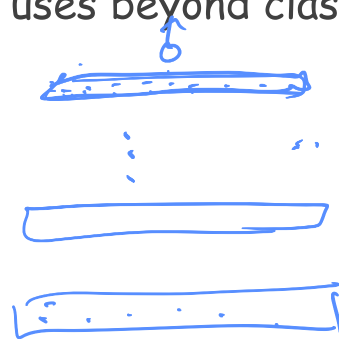
# LAST CLASS

- Neural Tangent Kernels
  - Can one analyze dynamics of gradient descent? [can view as Kernel regression for a time-varying kernel]
  - [Jacot et al.] for infinitely wide nets, kernel remains "fixed" – neural tangent kernel; so NN learning == kernel regression with Neural Tangent Kernel.
- NNs as Feature Learning or Representation Learning
  - NN transforms inputs -> "feature space embeddings", i.e., new representation
  - **Why?** Representations can have uses beyond classification (e.g., image captions, transfer learning, …)

2010-2016

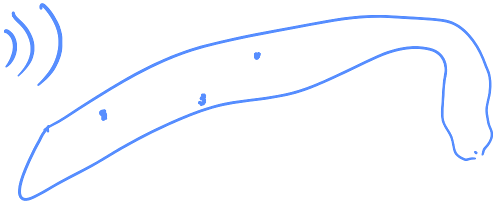
Solve new task for which we don't have enough data.

$F_{L-1}$



$x$

$F_3(F_2(F_1(x)))$



# REPRESENTATION LEARNING

- What makes a good representation?
  - Contrastive (for classification)
  - "Disentangled" or orthogonal
  - Sparse "explanations" for phenomena
  - Hierarchically organized, explanatory
  - Leverage domain knowledge
- Supervised vs Unsupervised

"Auto-encoder" view:

→ representation should allow you to "approximately recover" input.

→ Pervading view in unsupervised rep. learning.

→ Information "bottleneck"

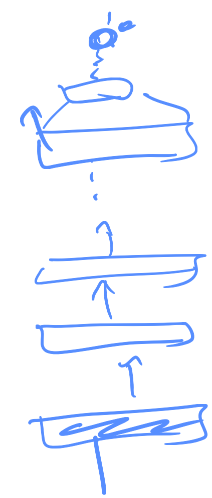
"calibration"

Supervised

- Pros: contrastive, better accuracy for *given task*, no special training
- Cons: may not generalize

Unsupervised

- Pros: generalizes to many tasks, no careful data collection needed
- Cons: unclear how to learn!



# UNSUPERVISED REPRESENTATION LEARNING

- Classic approaches: manual feature engineering, autoencoders and "sparse coding"

recovering input is key.

inputs: $x_1, x_2, \dots, x_N \in \mathbb{R}^d$.

"basis" or "dictionary": $v_1, \dots, v_m \in \mathbb{R}^d$ such that each $x_i \approx \sum_{j=1}^{m} \alpha_j^{(i)} v_j$, where $\alpha^{(i)}$ is a "sparse" vector (only $k$ nonzero)

useful in "early layers", not beyond (in practice).

- More modern: self-supervised learning, invariances and data augmentation

  - Example: NLP tasks

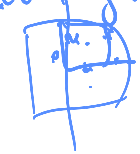
High level idea:

→ Can you do unsupervised rep. learning using supervised learning?

Idea: This is a cat; .. can do it for images..

pretext tasks.

auxilliary tasks.



An image and its slightly rotated version have the same rep.

# A Neural Probabilistic Language Model (2003)

**Yoshua Bengio** BENGIOY@IRO.UMONTREAL.CA
**Réjean Ducharme** DUCHARME@IRO.UMONTREAL.CA
**Pascal Vincent** VINCENTP@IRO.UMONTREAL.CA
**Christian Jauvin** JAUVINC@IRO.UMONTREAL.CA
*Département d'Informatique et Recherche Opérationnelle*
*Centre de Recherche Mathématiques*
*Université de Montréal, Montréal, Québec, Canada*

**Editors:** Jaz Kandola, Thomas Hofmann, Tomaso Poggio and John Shawe-Taylor

## Abstract

A goal of statistical language modeling is to learn the joint probability function of sequences of words in a language. This is intrinsically difficult because of the **curse of dimensionality**: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training. Traditional but very successful approaches based on n-grams obtain generalization by concatenating very short overlapping sequences seen in the training set. We propose to fight the curse of dimensionality by **learning a distributed representation for words** which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. The model learns simultaneously (1) a distributed representation for each word along with (2) the probability function for word sequences, expressed in terms of these representations. Generalization is obtained because a sequence of words that has never been seen before gets high probability if it is made of words that are similar (in the sense of having a nearby representation) to words forming an already seen sentence. Training such large models (with millions of parameters) within a reasonable time is itself a significant challenge. We report on experiments using neural networks for the probability function, showing on two text corpora that the proposed approach

# REPRESENTATION LEARNING IN NLP

word2vec. [Mikolov et al.]

Latent Semantic Indexing (90s).

- Approaches based on Firth's hypothesis (1950s).

"a word is defined by the company it keeps".

Look at blocks of text; look at n-grams, continuous n-sequences of words.

Bob ate an apple.

$w_1$
$w_2$
$\vdots$
$w_N$
set of all words.

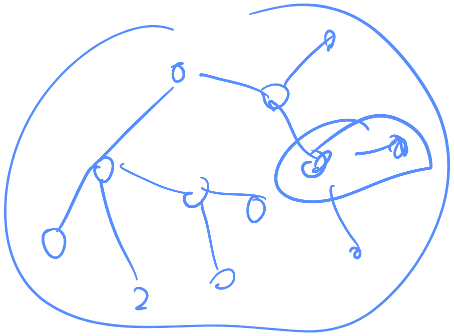
for each $w_i$, how frequently does it co-occur with $w_j$ (in say a 5-gram)

$a_{ij}$


If we want a vector rep. for words,

→ For every word $i$, we have vector $u_i$ with the prop. that $a_{ij} \approx \langle u_i, u_j \rangle$.

# REPRESENTATION LEARNING IN GRAPHS




→ Graph as a matrix is useful.

→ Common nbrs etc. not exactly captured.

→ Graphs produce n-grams by random walks.