Introduction to the mathematics of deep learning.

Jonas Groschwitz, Antoine Venant

April 15, 2019

Approximating functions

The big picture

- Formalize problem as *learning* a function: $f : \mathbb{R}^n \mapsto \mathbb{R}^m$.
- Define a class of models. That, is a class of 'candidate' functions $g_\theta : \mathbb{R}^n \mapsto \mathbb{R}^m$ that we know how to compute.
  - $\theta \in \mathbb{R}^k$: parameters of the model.
- **Find** the model g_{θ^*} providing **the best approximation of** f **given available evidence.**

Questions

- Which class of models?
- What is the best approximation (given available evidence)?
- How do we find it?

POS-tagging example.

- Lexicon: $V = \{v_1, \ldots, v_L\}$, and POS-TAGS: $U = \{u_1, \ldots, u_M\}$.
- Observations: texts $x = (w_1, \ldots, w_N)$ with annotated POS-TAGs sequences $y = (t_1, \ldots, t_N)$.
- Hypothesis: observations (x, y) are instances of some function $f : x \mapsto y$.
- Goal: use observations to find a good computable approximation of f.

Example observations

$$V = \{\text{the}, \text{a}, \text{girls}, \text{walk}, \text{take}, \text{look}\}, U = \{DET, N, V\}$$

$$\text{Observations:} \quad \begin{array}{ll} x_1 = \text{(the, girls, walk)} & y_1 = (DET, N, V) \\ x_2 = \text{(the, girls, take, a walk)} & y_2 = (DET, N, V, DET, N) \end{array} .$$

Accounting for observations is not enough

The following function is obviously compatible with all observations:

$$f : \begin{cases} x_1 \mapsto y_1 \\ x_2 \mapsto y_2 \end{cases}$$

but we haven't learned anything new.

- Undefined for unobserved x's. We could take arbitrary definition.
- Won't generalize to new data.
- **Need to restrict to candidate functions capturing shared structures between observed and unobserved instances.**

A triple requirement

The considered class of candidate functions (or ‘models’) should be

- Expressive enough to account sufficiently for observations. (Example on blackboard)
- Constrained enough to allow generalizing from observations.
- Support appropriate learning algorithm so the best model can be found.

Example

A simple class of models

- POS-TAG only depends on word, not on context: $P(u \mid v)$.
- Candidate model: for each word v in lexicon, conditional distribution on POS-TAG given this word.
- *i.e.*, for each $v \in V$, a vector of $|U|$ real numbers summing to one.
- Best candidate model? Maximizing likelihood of observations: $\prod_{i=1}^{N} P(y_i \mid x_i)$.

Example

For a model with: $P(V \mid \text{walk}) = 1/2$, $P(N \mid \text{walk}) = 1/2$, $P(DET \mid \text{the}) = P(N \mid \text{girls}) = 1$:

$$P((DET, N, V) \mid (\text{the, girls, walk})) = 1 \times 1 \times 1/2 = 1/2$$

How does it fare with requirements?

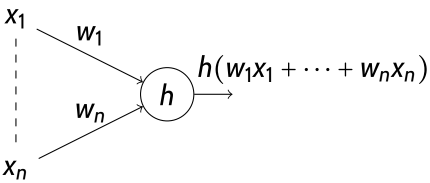
Observations: $$\begin{aligned} x_1 &= \text{(the, girls, walk)} & y_1 &= (DET, N, V) \\ x_2 &= \text{(the, girls, take, a walk)} & y_2 &= (DET, N, V, DET, N) \end{aligned}.$$

- Support appropriate learning algorithm? Model maximizing likelihood is obtained by setting $P(u \mid v) = \frac{\#\text{ occurences of word} v \text{ with POST-TAG } u}{\#\text{ occurences of v}}$. *e.g.*, $P(V \mid walk) = 1/2$.
- Constrained enough to allow generalizing from observation? To some extent. Can tag new sentences using same words, but no sentences using unseen words.
- Expressive enough to account sufficiently for observations? Not quite. Best model has only probability 1/2 for both instances, because lacks context dependence.

Neural networks

- In this seminar we are concerned with a specific class of models: neural networks.
- Roughly, a computing device represented as a directed graph where each node is a computation unit called a *neuron*.
- We will try to understand how to use them in compliance with the requirements we have mentionned.

A neuron



(Example network on whiteboard)

Expressivity of neural models.

Neural networks are very expressive. Here are for instance two known results (summarized here with some approximation):

- For any non-constant, bounded and continuous non-linearity h, any continuous function $f : [0,1]^n \mapsto \mathbb{R}^m$, we can find a neural network with a single hidden layer, using only h as non-linearity and approximating f as close as one wants. However the hidden layer could be extremely large w.r.t the input!
- If one allows deeper networks, such an f can be approximated arbitrarily close by a network with width $n + 4$, using RELU as non-linearity.

Note:

The super-simple probabilitic model from before *is* a neural network too (with a little work – illustration if time permits)!

Neural network as a class of models?

- Support appropriate learning algorithm? A very general one called backpropagation. **But computationally expensive if the network is large and 'gradient explosion' if the network is deep.**
- Constrained enough to allow generalizing from observation? In practice, yes but difficult question. The choice of architecture plays an important role here too.
- Expressive enough to account sufficiently for observations? Yes **if the network is sufficiently large or sufficiently deep.**

Topics for this course

Basics

- Mathematical basis for training and using (deep) neural networks.
- A little linear algebra, and a little functional analysis, and the backpropagation algorithm.
- Overview of (stochastic) gradient descent.

Topics for this course

Recurrent neural networks

- Gradient explosion problem and existing solutions.
- Long Short Term Memory Networks and how they solve the problem.
- Sequentially structured inputs.

Topics for this course

Tree structured inputs

- Tree lstm.
- Are they more efficient? On what kind of input? Why?

Topics for this course

Attention

- Learning which part of the input to pay attention to.

Topics for this course

Implementation

- Automatic differentiation: making everyone's life easier.