11-785 - Deep Learning 24/04/2020

L2 - Neural net as a universal approx.

- BR: - Neural networks can represent almost anything*
  - NN black box I/O device; content of boxes
  - NN originally conceived of emulations/caricatures of 'neurons'
  - ANN: perceptron as 'basic unit' - threshold unit
- Rewrite a linear combination of inputs with threshold → affine combination

from $z = \sum_i w_i x_i - T \longrightarrow y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{else} \end{cases}$ (?)

- Affine (think of as containing intercept)
- Replace threshold unit with sigmoid activation 'squashing function'
  - bounded on both sides
  - monotonic
- We continue assuming activation function is a threshold unit for this lecture.
- activation function - function that affine combination is put through
- original perceptron was formulated using threshold units

- MLP - network of perceptrons (units)
- directed I/O; layered ('hidden layers')
- layer - a way of referring to sub-networks of neurons/units that do not communicate
- "depth" - what does it mean for a graph to be 'deep'
  - sense of universal approximation in Hornik?

BR: - length of longest path in a directed graph from source → sink (graph theoretic specification)

- depth > 2 ⇒ 'deep'
  or ≥ 3
- one initial input layer, output layer → not 'deep'

- MLP: inputs: real or Boolean
  output: ...
- MLP can approximate any Boolean function; any real-valued function with limitations

4 topics → see slides (MLP)

- classify/organise lecture this way

1. MLPs as universal Boolean functions

LH: AND RH: NOT

Bottom: OR

universal AND gate

(W): (A1) = universal AND (logic) ✓

- values in circles - thresholds
- values edges - weights

- Perceptron also as universal OR (W): (A2) - logic
- generalised majority gate
- These 'nearly fooled' Rosenblatt
- Can single perceptron cannot compute XOR; but MLPs e.g. 3 perceptrons
- Emulation of XOR with 3 perceptrons (2 hidden perceptrons)
- Perceptrons → Boolean gates → Boolean functions.

(F1): MLPs are 'universal Boolean functions'

- (Q): How many layers needed? BR: only 1 hidden layer
- Boolean function is a truth table ultimately
- truth table - 6 conditions for which output is 1
- disjunctive normal formula for Boolean function → 7 disjunctive terms/clauses (see slides)
- ~~1 perceptron~~ Heuristic: Illustration of DNF - 1 hidden layer
- Any truth table
- (F1): A 1-hidden layer MLP is a universal Boolean function.

Karnaugh Map

- 7 blocks - 7 clauses
- Get reduced DNF $O = \bar{Y}\bar{Z} + \bar{W}X\bar{Y} + \bar{X}Y\bar{Z}$ (3 clauses) (reduced DNF)

∴ MLP - 3 layers

(W): (A3) - clarify logic of Karnaugh maps

- Largest irreducible DNF - see Karnaugh patterns
- largest no. of neurons for which we have irreducible DNF - 8 red squares: 8 'neurons'

- intermediate cube

- Generalisation: largest no. perceptrons required for a single layer MLP for an N-input variable function:-

(A11) 2^{N-1} (exponentially large)

- How about multiple hidden layers?

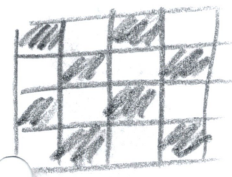
- 1 XOR - 3 perceptrons (in fact 2)

- checkerboard pattern is XOR $O = W \oplus X \oplus Y \oplus Z$ (check sqwe)

- check cube (see slide)

---

using this logic



$O = W \oplus X \oplus Y \oplus Z$

3 XORS ; 3 perceptron / XOR → 9 neurons / perceptrons.

(A4): more greatly, XOR of N variables requires $3(N-1)$ perceptrons (no deep)

(A11): for 1 hidden layer: $2^{N-1}+1$ perceptrons (including output): exponential in N

For deep-network: $3(N-1)$ perceptrons - linear in N!

arranged in $2\log_2(N)$ layers ?

(W)(A5): Relation between depth - hidden layers clarify

(W): (A4): depth-perceptrons logic

(A11): optimal depth is function of no. inputs

(A11): No. parameters → no. of connections 32 connections

(A6)

- this matters
- exponential no. neurons → superexponential no. of parameters / weights (A7)
- Read slides before lectures

- there are formal results here → XOR as parity problem (A8)
  Furst et al.
- Shannon et al.

- Network size summary → see course readings ✓
- (A11) points here (heuristic observations)

- caveats 2

- threshold gates substit. for Boolean circuits

---

2. MLPs as universal classifiers over real inputs

- decision boundary over input patterns in 784 dimensional space
- (N-1) dimensional hyperplane ? (~)
- OR, NOT, XOR
- composition of Boolean perceptrons → complex decision boundaries
- (W)(PG): Booleans over reals → pentagon
- and then further compositions (compositon of pentagons)
- via fragmentation; compose using subnets
- (PG): MLP can compose arbitrarily complex decision boundaries

  BR: with 1 hidden layer

- (Q): do you understand the heuristic arguments
- some kind of limit behaviour

(W): HOW quickly can you adapt your methods to couse

(W): I don't understand what he is trying to illustrate here

going from pentagon → ~~cylcle~~ cylinder

(PG): Illustration of arbitrarily approximating the two pentagons using limiting behaviour (universal approx.)

(BR): some shapes can be modelled with 1 hidden layer MLP exactly; others can use infinitely deep hided layers (double pentagon approx.) as approx.

(Q): need to clarify the claims precisely

---

optimal depth → see papers

(tradeoff of depth (hidden layers) vs no. of neurons)

- checkerboard → heuristic understanding → digest

(PG): the difference in size between the deeper optimal XOR net and shallower net increases (with more neurons) increases with the type of pattern complexity

(M): deeper networks are more expressive

---

3. MLPs as universal approximators

- use 'cylinders' with volume (some kind of calculus/analysis arguments)
- can exploit ranges of activations functions

(M): network as a universal map from entire domain of input values to entire range of output activation

---

- #1
- optimal depth vs width
- sufficiency of architecture

- there are some requisite depth/breadth constraints (capacity)
- not all architectures can represent any function

- After information has propagated through network for 8 (rather than 16) units in input layer; there is some loss in the sense that we can no longer make distinctions within the diamonds

(M): there are significant constraints on architecture:-

1. network must be sufficiently wide or deep at each layer

- (W) (AB) (?) - information propagation
- note his logic (*) - a lot missed; rewatch until you understand
- provided you have appropriate activations to capture info missed by lower layers?

---

width vs activations vs depth
- sufficiency of architecture
- capacity/expressive power has many definitions
  - info/storage capacity
  - VC dimension (complexity of patterns): bounded by square no. of weights

---

VC dimension (capacity): see papers (!) → [-]

- Highly abstract results

- The function we are trying to model :- remember :-
  1. width of individual layers } can tradeoff
  2. depth of network
  3. Activation functions used -

(*) For any given activation functn ; they limit extent of tradeoff.

NEXT CLASS: How do we set parametes, actually training an MLP