Deep Learning Refresher

Tirtharaj Dash

Based on:

https://atcold.github.io/pytorch-Deep-Learning/
(Week 1 – 6)

Inspiration of Deep Learning and its history

- Loosely: Brain :: Neural Nets [like, Bird :: Aeroplane]
- Historical names: Cybernetics (1940s–1960s), Connectionist Models (1980s–1990s), Deep Learning (2006–)
- Neural Network (NN, Neural Net) is the term used to refer to such an architecture

Week 1 II

- Started: McCulloch and Pitts Model of neuron (1943)
  - Idea: Neurons are threshold units (on/off states)
  - Purpose: Build Boolean circuit by connecting neurons
  - Outcome: Perform logical inference
  - How: (1) Neurons compute weighted sum of inputs; (2) Compare the sum to its threshold; (3) Neuron is turned 'on' if the sum is above the threshold; 'off' otherwise
  - A simplified view of how a neural network works

Week 1 III

- Donald Hebb: Hebb's rule or Hebbian Learning (1947)
  - Idea: Neurons in the brain learn by modifying the strength of the connections between neurons
  - How: If two neurons fire together, the connection linked between them increases; decreases otherwise
  - Also called hyper learning

Week 1 IV

- Norbert Wiener: Proposal for cybernetics (1948)
  - Idea: having systems with sensors and actuators, you have a feedback loop and a self-regulatory system
  - Result: The rules of the feedback mechanism of a car all come from this work.

- Frank Rosenblatt: Perceptron (1957)
  - Weight modification in a simple neural net
  - This was a big breakthrough in the field

- Towards late 1960s, the field started to die off. Reasons:
  - The researchers used neurons that were binary (not differentiable)
  - There was no idea of continuous neurons (or, activation functions)
  - Backpropagation requires continuous activation function
  - Before 1980: the multiplication of two floating-point numbers were extremely slow

Week 1 VII

- Restarted again: 1985 with emergence of backpropagation

Week 1 VIII

- 1995: the field died again and the machine learning community abandoned the idea of neural nets

Week 1 IX

- 2006-2010:
  - Huge performance improvement in speech recognition tasks using neural nets
  - Wide deployment in the commercial field

Week 1 X

- 2013: Computer Vision switched to neural nets
- 2016: Natural Language Processing switched to neural nets

- ..., and the rest is history!

Supervised Learning

- Majority of deep learning applications use supervised learning.
- Steps:
  - Collect a bunch of pairs of inputs and outputs
  - Inputs are feed into a machine to learn the correct output
  - When the output is correct, don't do anything
  - If the output is wrong, tweak the parameter of the machine and correct the output toward the one you want.
  - Change direction and amount of update requires gradient computation and backprogation

Pattern Recognition (before emergence of DL):

- Data $\rightarrow$ Feature Extraction $\rightarrow$ Trainable Classifier
- Issue: The feature extractor was designed by hand.

Pattern Recognition (in DL era):

- Sequence of modules (each module is a feature extractor)
- Each module has tunable parameters (and nonlinearity)
- Modules are stacked one after another (a "deep" stack)

A basic multi-layered neural net:

- The input is represented as a vector such as an image or audio.
- This input is multiplied by the weight matrix whose coefficient is a tunable parameter.
- Every component of the result vector is passed through a nonlinear function such as ReLU.
- Repeat for all modules to finally compute the outputs
- Compare the computed outputs and true outputs
- Optimise some objective function and tune the parameters of each module
- Repeat the above steps until some stopping condition

Computing gradients by backpropagation: Chain rule

$$\frac{\partial C}{\partial \boldsymbol{x}_{i-1}} = \frac{\partial C}{\partial \boldsymbol{x}_i} \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{x}_{i-1}}$$

$$\frac{\partial C}{\partial \boldsymbol{x}_{i-1}} = \frac{\partial C}{\partial \boldsymbol{x}_i} \frac{\partial f_i(\boldsymbol{x}_{i-1}, \boldsymbol{w}_i)}{\partial \boldsymbol{x}_{i-1}}$$

similarly:

$$\frac{\partial C}{\partial \boldsymbol{w}_i} = \frac{\partial C}{\partial \boldsymbol{x}_i} \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{w}_i}$$

$$\frac{\partial C}{\partial \boldsymbol{w}_i} = \frac{\partial C}{\partial \boldsymbol{x}_i} \frac{\partial f_i(\boldsymbol{x}_{i-1}, \boldsymbol{w}_i)}{\partial \boldsymbol{w}_i}$$

Convolutional Neural Net (CNN):

- Inspired from: Hubel and Wiesel's experiment with visual cortex of cat
  - Neurons react to edges that are at particular orientations.
  - Groups of neurons that react to the same orientations are replicated over all of the visual field.

- Fukushima (1982)
  - Neurons are replicated across the visual field
  - There are complex cells that pool the information from simple cells (orientation-selective units).
  - As a result, the shift of the picture will change the activation of simple cells, but will not influence the integrated activation of the complex cell (convolutional pooling).

- Breakthrough: Application of backprop to CNNs (LeCun, 1992)
  - Handwritten digit recognition (end-to-end) using neural network

Week 1 XIX

- Deep Learning emerged: AlexNet (2012)
  - Due to utilisation of general-purpose GPUs for computation

- ..., and the field is only growing.

Feature extraction (before DL):

- Random projections
- Radial-basis transformation
- Kernel tricks
- ...

Feature extraction (now): Exploit the compositional nature of data

- Images (pixels→edges→multi-edge shapes→motifs→. . .)
- Text (characters→words→word-groups→clauses→. . .)
- Speech (samples→bands→sounds→phones→phonemes→. . .)

DL attempts to learn the feature extractors in a hierarchical fashion.