A Brief Introduction to Deep Learning

--Yangyan Li
How would you crack it?
How to avoid being cracked?
Seam Carving!
Labradoodle or fried chicken
Puppy or bagel
Sheepdog or mop
Chihuahua or muffin
Barn owl or apple
Parrot or guacamole
Raw chicken or Donald Trump
But, we human actually lose!

• A demo that shows **we, human, lose**, on the classification task, we are proud of, we have been **trained** for millions of years!

• If we want to make it hard for bots, it has to be hard for human as well.
How would you crack it?
We human lose on Go!
Chess: $10^{47}$
Deep Blue, Feb 10, 1996

Go: $10^{170}$
AlphaGo, March, 2016
We (will) lose on many **specific** tasks!

- Speech recognition
- Translation
- Self-driving
- ...

- BUT, they are not AI yet...
- Don’t worry until it dates with your girl/boy friend...
Deep learning is so cool for so many problems...
OLD CROW

MODERN CROW

Update Yourself - It saves a lot of extra effort
A Brief Introduction to Deep Learning

- Artificial Neural Network
- Back-propagation
- Fully Connected Layer
- Convolutional Layer
- Overfitting
Artificial Neural Network

1. Activation function
2. Weights
3. Cost function
4. Learning algorithm

Live Demo
Neurons are functions

• Let’s start with a complex one!
  \[ f(x, y) = x + y \]

• Given \( x = a, y = b \), how to update \( x \) and \( y \) to make \( f(x, y) \) larger?

• Follow gradient directions!

\[
f(x, y) = x + y \quad \rightarrow \quad \frac{\partial f}{\partial x} = 1 \quad \frac{\partial f}{\partial y} = 1
\]

\[
x = a + 0.01 \times 1, \quad y = b + 0.01 \times 1
\]

\[ f(x, y): a + b \rightarrow a + b + 0.02 \]
Neurons are functions

- A more complex one!
  \[ f(x, y) = x \cdot y \]

- Given \( x = a, y = b \), how to update \( x \) and \( y \) to make \( f(x, y) \) larger?

- Follow gradient directions!

  \[ f(x, y) = xy \quad \rightarrow \quad \frac{\partial f}{\partial x} = y \quad \frac{\partial f}{\partial y} = x \]

  \[
  x = a + 0.01 \cdot b, \\
  y = b + 0.01 \cdot a
  \]

  \[
  f(x, y): a \cdot b \rightarrow (a + 0.01 \cdot b)(b + 0.01 \cdot a) \\
  f(x, y): 4 \cdot (-3) \rightarrow 3.97 \cdot (-2.96)
  \]
Back-propagation

- An extremely complex one!
  
  \[ f(x, y, z) = (x + y) * z \]

- Let \( q(x, y) = (x + y) \), then \( f(x, y, z) = q(x, y) * z \)

- Chain rule:
  \[
  \frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}
  \]
Now, serious stuff, a bit...
Fully Connected Layers
“When in doubt, use brute force.”
--Ken Thompson
“If brute force is possible...”
--Yangyan Li
Convolutional Layers
Convolutional Layers

Input Volume (+pad 1) (7x7x3)  Filter W0 (3x3x3)  Filter W1 (3x3x3)  Output Volume (3x3x2)

- x[:,:,0]
- 0 0 0 0 0 0 0
- 0 0 2 0 1 0 0
- 0 0 2 0 2 0 0
- 0 0 2 0 2 0 0
- 0 0 2 1 2 0 0
- 0 0 2 1 2 0 0
- 0 0 0 0 0 0 0

- w0[:,:,0]
- 1 1 -1
- 0 1 0
- 0 -1 -1

- w1[:,:,0]
- 1 1 -1
- -1 1 0
- -1 0 1

- b0 (1x1x1)
- Bias b0
- 1

- x[:,:,1]
- 0 0 0 0 0 0 0
- 0 1 0 2 2 0 0
- 0 1 1 0 2 2 0
- 0 0 0 0 2 2 0
- 0 0 1 2 2 2 0
- 0 1 1 2 2 2 0
- 0 2 1 2 2 2 0
- 0 0 0 0 0 0 0

- w0[:,:,1]
- -1 1 -1
- -1 1 0
- 1 0 -1

- w1[:,:,1]
- 0 0 -1
- 0 0 0
- 1 1 0

- b1 (1x1x1)
- Bias b1
- b0

- x[:,:,2]
- 0 0 0 0 0 0 0
- 0 1 0 2 2 2 0
- 0 2 2 1 0 2 0
- 0 2 2 2 2 0 0
- 0 0 1 1 0 1 0
- 0 0 1 1 0 1 0
- 0 0 0 0 0 0 0

- w0[:,:,2]
- -1 1 0
- -1 1 0
- 0 1 0

- w1[:,:,2]
- 1 0 -1
- 1 -1 0
- 0 -1 0

- b1[:,:,0]
- 0

- toggle movement

- o[:,:,0]
- 5 1 -1 4
- 7 3 2
- 8 5 11
- 2 0 5
- -4 8 7
- 1 0 6
Convolution Filters
Computer vision features

SIFT

Spin image

HoG

Textons

and many others:

SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, .....
Traditional Recognition Approach

Features are not learned

Input data (pixels) -> feature representation (hand-crafted) -> Learning Algorithm (e.g., SVM)

Image -> Low-level vision features (edges, SIFT, HOG, etc.) -> Object detection / classification
Feature Engineering vs. Learning

• Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work.

• “When working on a machine learning problem, feature engineering is manually designing what the input x's should be.”

  -- Shayne Miel

• “Coming up with features is difficult, time-consuming, requires expert knowledge.”

  -- Andrew Ng
With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

— John von Neumann
Under- and Over-fitting examples

Regression:
- Predictor too inflexible: cannot capture pattern
- Predictor too flexible: fits noise in the data

Classification:
- x_1 vs. x_2
How to detect it in training process?
Dropout
Sigmod $\rightarrow$ ReLU

\[
\frac{1}{1 + e^{-x}}
\]
Sigmod $\rightarrow$ ReLU
Compute, connect, evaluate, correct, train madly...

Non-linearity, distributed representation, parallel computation, adaptive, self-organizing...
A brief history


• 1993: Nvidia started...


• 2010: “GPUS ARE ONLY UP TO 14 TIMES FASTER THAN CPUS” SAYS INTEL –Nvidia


“Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.”

--Winston Churchill
Is Deep Learning Taking Over the World?

- What applications are likely/unlikely to benefit from DL? Why?
Deep learning, yay or nay?

A piece of cake, elementary math...

It eats, a lot!