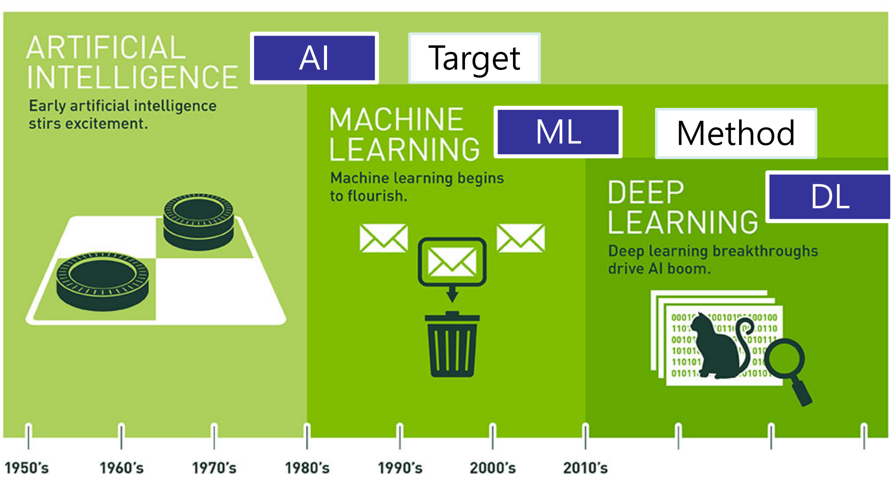
ECE 590-10/11
COMP ENG ML & DEEP NEURAL NETS

1. INTRODUCTION

HAI LI & YIRAN CHEN, FALL 2019

1

AI ↔ ML ↔ DL



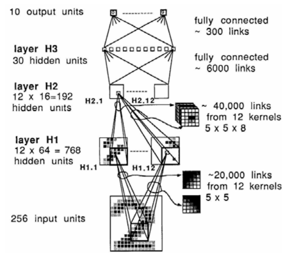
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Source of image: https://blogs.nvidia.com.tw/2016/07/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

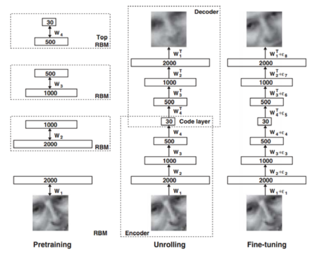
2

Historical View (Overview)

Convolutional Network (1980s)	Dark period (1990s)	Renaissance (2006 ~ Present)



- Serious problem: Vanishing gradient
- No benefits observed by adding more layers
- No high-performance computing devices



pentium III

Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition.1989.

J. Schmidhuber. Deep Learning in Neural Networks: An Overview. arxiv, 2014.

G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. Science, 2006.

3

3

Historical View (Milestones)

Timeline

1943 First Neural Network Model 

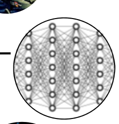
1941 First Electronic Computer 

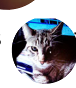
1950 Turing Test 

1970 Expert Systems 

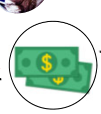
1990 AI in the Military 

1997 Deep Blue Defeats Kasparov 

2006 The Concept of Deep Learning 

2012 Google X learns to Find Cat Videos 

2016 Alpha Go Defeated Lee Sedol 

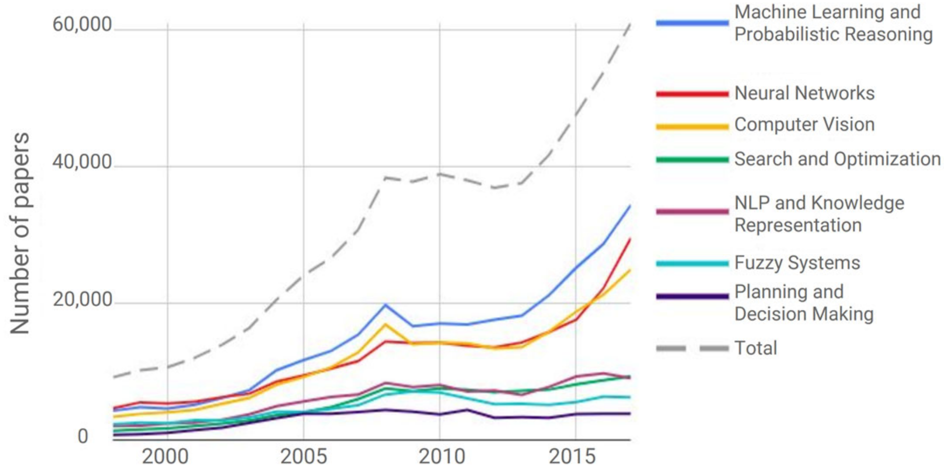
~~Now 180K $/year for AI Engineer~~ 

4

4

Historical View (Papers)

The number of AI papers on Scopus



Source of image: http://cdn.aiindex.org/2018/AI%20Index%202018%20Annual%20Report.pdf

5

5

Overview

- For: MS/MEng students who want to learn computer engineering methods commonly performed in developing and using machine learning and deep neural network models.
- ***Practice*** will be the focus of this course, while ***theoretical understanding*** is essentially important.

6

6

Objectives

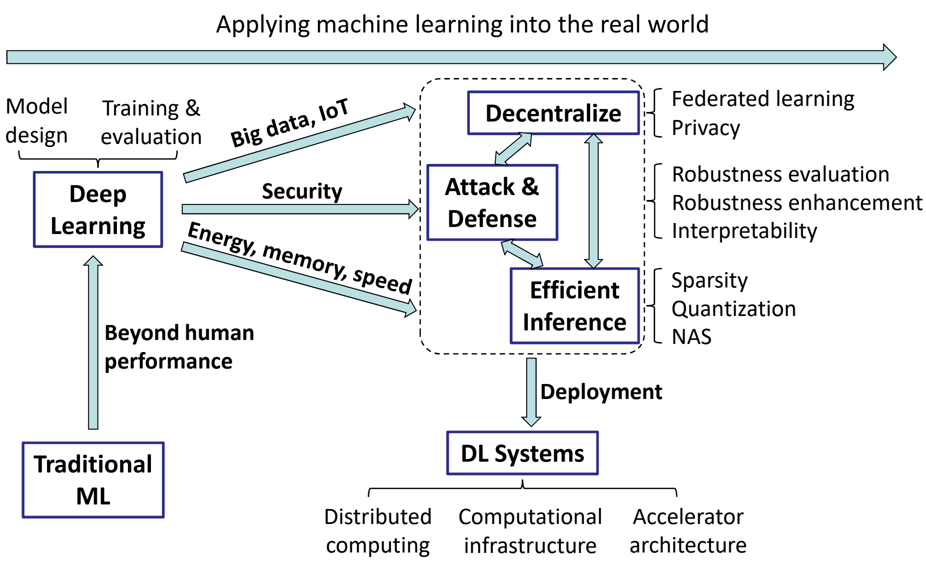
This course is designed to improve your ability to:

1. **Comprehend** the mechanisms, applications, and limitations of techniques commonly used in training and inference of machine learning and deep neural networks algorithms;
2. **Formulate** hypotheses and conduct experiments employing these techniques;
3. **Analyze** experimental results obtained by these techniques and your own practices and **derive** the conclusions that are supported or not supported by your data;
4. **Synthesize** and **communicate** the experimental results and data through oral narrative, graphs, figure legends, and result narratives;
5. **Utilize** proper engineering techniques for novel machine learning algorithms and deep neural network models;
6. **Propose** new engineering approaches and techniques to further enhance machine learning and deep neural network training and inference execution.

7

7

Roadmap of the course



8

8

What we will learn

- DNN fundamentals
  - Convolutional neural network (CNN), recurrent neural network (RNN), forward/backward propagation, training, network architecture, …
- DNN acceleration
  - Compact neural architecture, model compression, pruning, quantization, sparsification, …
- Machine learning security
- Hardware systems
  - GPUs, CPUs, cloud servers, accelerators, etc.
- Advanced topics
  - Distributed computing, neural architecture search (NAS), generative adversarial network (GAN), decentralization and privacy

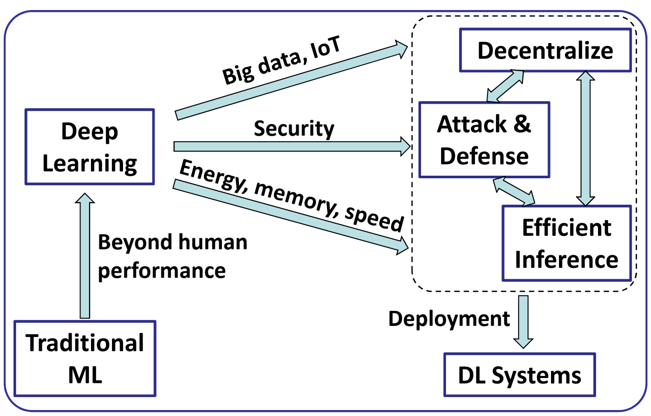
9

9

Related topics and courses

Deep learning: **ECE 590-06**

Cloud computing: **ECE 563**; Smart sensor: **ECE 590-04**
Security: **ECE 590-03**; Image processing & denoise: **ECE 588**
Information theory: **ECE 587**; Compressed sensing: **ECE 741**



Math basics: **ECE 581, 586**
Machine learning: **ECE 681, 682**

Implementation: **ECE 550, 551, 650**
Architecture design: **ECE 552, 590-24**
System optimization: **ECE 558, 563, 565**
Hardware: **ECE 538, 539, 559**

10

10

Prerequisite

- We expect that students to have basic object-oriented programming experience (e.g. C++, Python) and be familiar with linear algebra and computer hardware fundamentals prior to taking this course, such as
  - For graduate students: ECE 550 + ECE 551
  - For undergraduate students: ECE 381 + CS 308 + ECE/CS 250.
- If you are familiar with a topic that we are covering …
  - You may learn something new
  - I may present it slightly differently than you are used to
  - You may be able to help other students learn it
- If you do not have these pre-requisites and are unfamiliar with these topics
  - We will **NOT** be slowing down to cover them
  - Please come talk to me or a TA sooner rather than later!

11

11

"Learn by doing"

- 3-4 homework assignments
- 4 lab assignments
  - Lab 1: Environment setup and CNN visualization
  - Lab 2: Build and train your own CNN
  - Lab 3: Deep compression
  - Lab 4: Adversarial attack and adversarial training

12

12

Logistics

	ECE 590-10	ECE 590-11
Faculty:	Dr. Hai "Helen" Li Room 209B Hudson Hall Hai.li@duke.edu	Dr. Yiran Chen Room 209B Hudson Hall Yiran.chen@duke.edu
Lectures:	Tuesday/Thursday 10:05-11:20am Hudson Hall 208	Tuesday/Thursday 1:25-2:40pm Hudson Hall 207
Office Hours:	By appointment	By appointment
Teaching Assistants:	Tunhou Zhang tunhou.zhang@duke.edu Qing Yang qing.yang21@duke.edu	Huanrui Yang huanrui.yang@duke.edu Meng Xia mx41@duke.edu

TAs are NOT under obligation to bail you out at 3am or debug your code. Your best bet is to get help in a timely and reasonable manner!

13

13

Getting Info

- **Sakai:**
  - Syllabus, schedule, slides, assignments, rules/policies, prof/TA info, office hour info
  - Links to useful resources
  - Just assignment submission and gradebook
- **Piazza:** questions/answers
  - Signup link: piazza.com/duke/fall2019/ece5901011f19
  - Post all your questions here
  - Questions must be "public" unless good reason otherwise
  - No code in public posts!

14

14

Getting Answers to Questions

- What do you do if you have a question?
  - Check Sakai
  - Check Piazza
    - If you have questions about homework, use Piazza – then everyone can see the answer(s) posted there by me, a TA, or your fellow classmate
  - Contact TA directly if need additional background materials for prerequisite knowledge
  - Contact professor directly if issue that is specific to you and that can't be posted on Piazza (e.g., regrade)

15

15

Textbook & Software

- There are no designated textbooks for this course.
- The related reading materials (e.g., papers, webpages, etc.) will be distributed through Sakai before the classes.

- We recommend downloading Pytorch (https://pytorch.org/)

16

16

Homework and Lab Submission

- Homework assignments and lab reports will be submitted as PDF files through the Assignments tool in Sakai. The code of lab assignments will be submitted to our servers. The details will be given during class.
- Late policy
  - 5 days per individual total for the semester
    - Days, not classes
  - Used in entire days: 10 min late = on next day
  - After used up: must turn in on time
  - No credit for late work after this

17

17

Grading

Assignment	%
Lab assignments (4)	65%
Homework	30%
In-class assignments/discussion	5%

- Completion of all assignments is required in order to earn a passing grade of D- or better in this course.
- Course grades are determined using an absolute, but adjustable scale (i.e., there is no curve). A final course average (rounded to the nearest 0.1 point) of at least 93.3 = A, 90.0 = A-, 86.7 = B+, 83.3 = B, 80.0 = B-, etc.
- Note: the professors reserve the rights to scale the grades.

18

18

Grade Appeals

- All re-grade requests must be in writing
  - your assignment in question, with
  - a brief written description of the error, and
  - your Duke NetID.
- I will respond to your regrade request by email and make arrangement to return your work to you.
- As a matter of policy, when you request a regrade you are agreeing that the grading of the entire assignment may be re-evaluated.
- All regrade requests must be submitted no later than 1 week after the assignment was returned to you.

19

19

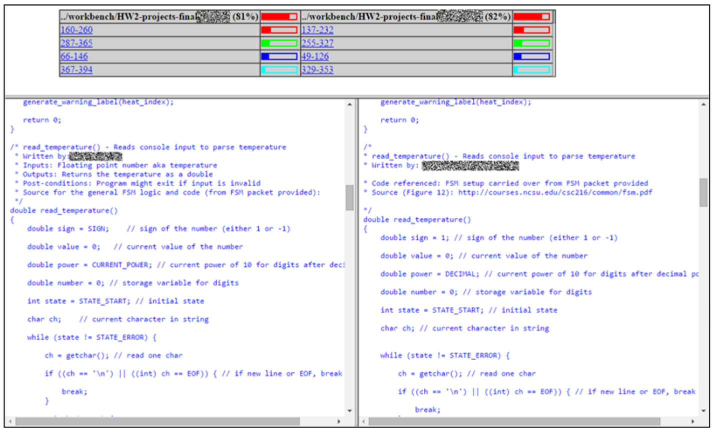
Academic Misconduct

- Academic Misconduct
  - Refer to Duke Community Standard
  - Homework/lab is individual – you do your own work
  - Common examples of cheating:
    - Running out of time and using someone else's output
    - Borrowing code from someone who took course before
    - Using solutions found on the Web
    - Having a friend help you to debug your program
- We will not tolerate any academic misconduct!
  - We use software for detecting cheating
- "But I didn't know that was cheating" is not a valid excuse

20

20

MOSS: Measure of Software Similarity



Doesn't care about:

- Comments
- Whitespace
- Naming
- Values

Only cares about code structure.

How to beat it?

Write your own code

21

21

Academic Integrity: General

Some general guidelines

- If you don't know if something is OK, please ask me.
- If you think "I don't want to ask, you will probably say no" that is a good sign its NOT acceptable.
- If you do something wrong, and regret it, please come forward—I recognize the value and learning benefit of admitting your mistakes. (Note: this does NOT mean there will be no consequences if you come forward).
- If you are aware of someone else's misconduct, you should report it to me or another appropriate authority.

22

22

Course Problems

- Struggling in course
  - Come to see me/TAs: We are here to help
- Other problems:
  - Feel free to talk to the instructor, who generally understands and will try to work with you
  - Some problems may extend well beyond my course
    - Academic Advisor
    - DGS Team

23

23

Our Responsibilities

- The instructor and TAs will…
  - Provide lectures/recitations at the stated times
  - Set clear policies on grading
  - Provide timely feedback on assignments
  - Be available out of class to provide reasonable assistance
  - Respond to comments or complaints about the instruction provided
- Students are expected to…
  - Receive lectures/recitations at the stated times
  - Turn in assignments on time
  - Seek out of class assistance in a timely manner if needed
  - Provide frank comments about the instruction or grading as soon as possible if there are issues
  - Assist each other within the bounds of academic integrity

24

24

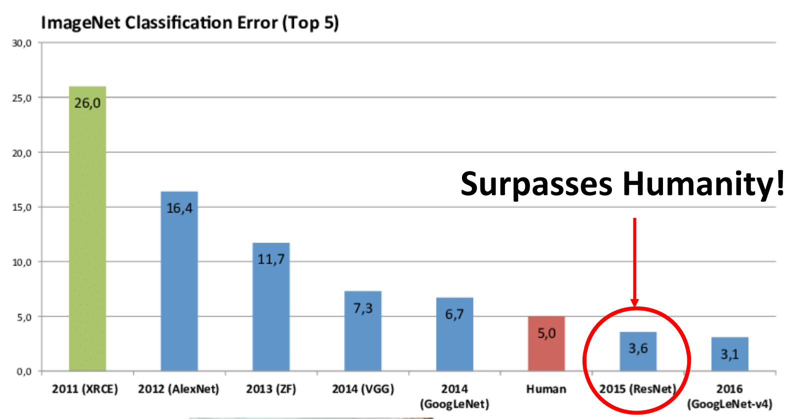
Outline

- Course introduction
- Machine learning & deep neural networks
  - Applications
  - Categories
  - Important metrics
  - Platforms & frameworks

25

25

Applications: Images



Can you tell what kind of turtle this is?

A. Dermochelys coriacea
B. Caretta caretta
C. Lepidochelys kempii
D. Lepidochelys olivacea

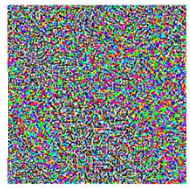
From Hung-yi Lee's introduction v8. Source of image: https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

26

26

Applications: Images

- However, surprisingly weak…

 + 0.007 ×  = 

Panda — noise — gibbon

57.7% confidence — 99.3% confidence

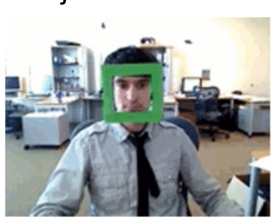
Source of this Pic: https://arxiv.org/pdf/1412.6572.pdf

27

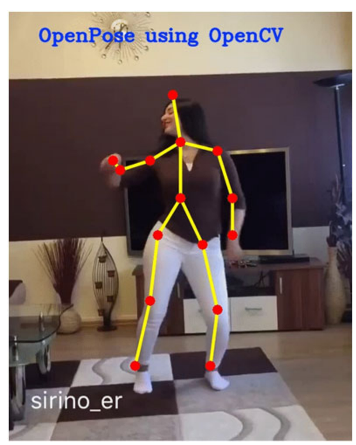
27

Applications: Videos

Object Detection



~~The Perfect Real Time Face Tracking~~



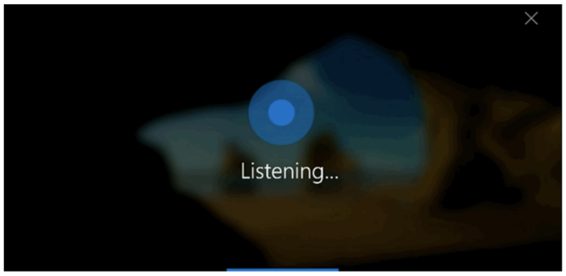
Human Pose Estimation

Source of Gifs: https://towardsdatascience.com/real-time-and-video-processing-object-detection-using-tensorflow-opencv-and-docker-2be1694726e5; https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/; https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/

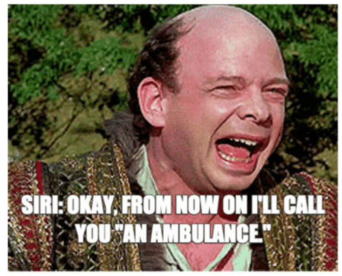
28

28

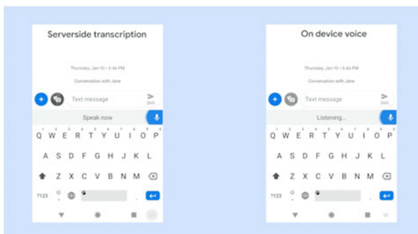
Applications: Speech



Cortana



“Siri, Call me an ambulance”



Speech To Text

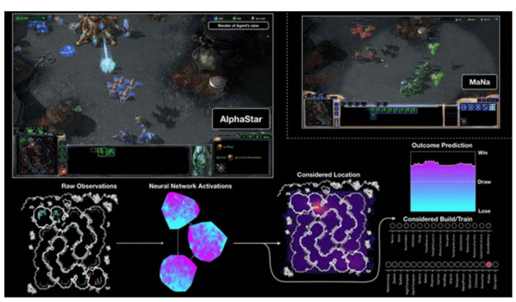
“Remember when people typed with two fingers? My voice is faster.”

Source of Gifs: http://www.impactlab.net/2016/10/26/microsofts-speech-recognition-is-now-as-accurate-as-a-humans/; https://www.nextbigfuture.com/2019/04/google-has-80-mb-speech-recognizer-that-can-work-offline-on-your-smartphone.html ; https://media.giphy.com/media/LYIgUG96n20h2/giphy.gif

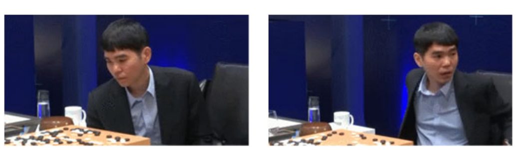
29

29

Applications: Game; Strategy



AlphaStar : StarCraft II



Alpha Go

Source of Gifs: https://www.techspot.com/news/78431-human-player-finally-beat-deepmind-alphastar-ai-starcraft.html; https://imgur.com/gallery/aPw2D; Matthew Inkawhich’s RT meeting Slides

30

30

Outline

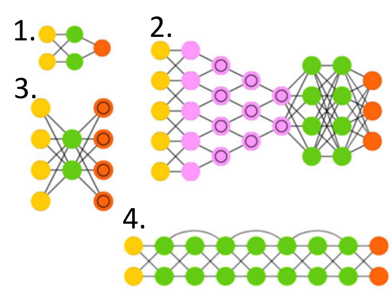
- Course introduction
- Machine learning & deep neural networks
  - Applications
  - Categories
  - Important metrics
  - Platforms & frameworks

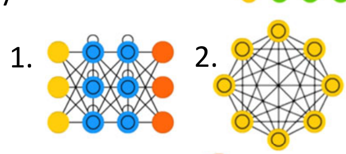
31

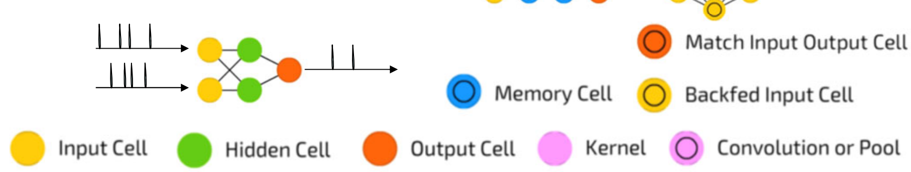
31

Structures

- Feedforward neural network:
  1. Multilayer perceptron
  2. Convolutional neural network
  3. Autoencoder
  4. Deep residual network
- Recurrent neural network:
  1. Long short-term memory
  2. Hopfield
  3. …
- Spiking neural network







Source of image: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

32

32

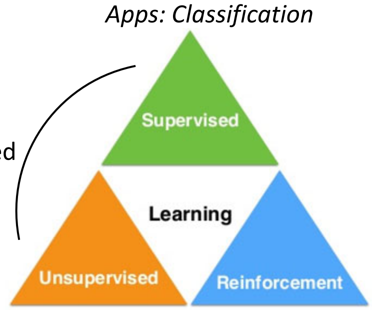
Learning types of NN

- Supervised
- Semi-supervised
- Unsupervised
- Reinforcement

Labeled data
Direct feedback
Predict outcome/future
Apps: Classification

Semi-supervised

Weakly-supervised



Details will be discussed in the next lecture

No labels
No feedback
Find hidden representations
Apps: Reconstruction

Decision process
Reward system
Learn series of actions
Apps: Decision-making

33

33

Outline

- Course introduction
- Machine learning & deep neural networks
  - Applications
  - Categories
  - Important metrics (**LASER**)
    - **L**atency
    - **A**ccuracy
    - **S**ize of model
    - **E**nergy efficiency
    - **R**obustness
  - Platforms & frameworks

34

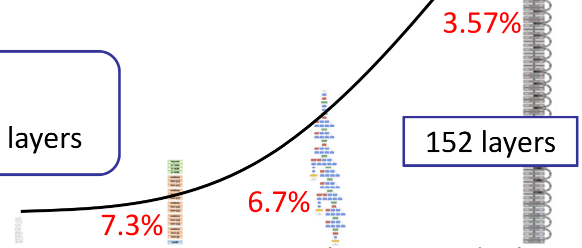
34

Latency

- Latency is a measure of delay.
  - The length of time it takes for the data that you feed into one end of your network to emerge at the other end.
- Better accuracy? Longer latency!
- VGG-16 needs ~3s to process a single image on your smart phone , which is unacceptable.

Going Deeper!

Deep = Many hidden layers

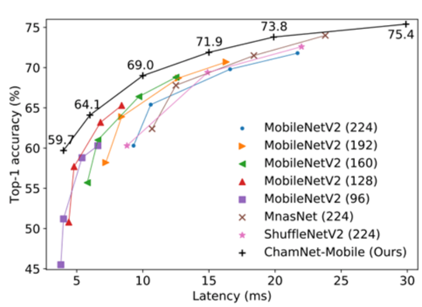


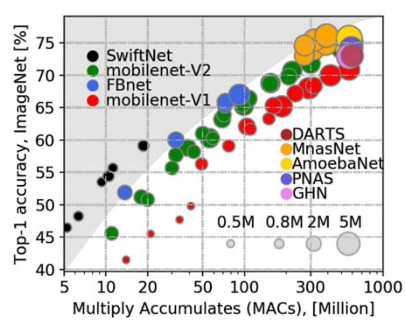
35

35

Accuracy

- Accuracy is a metric for classification problem
- We call it: “Top-*K* Accuracy”
- Higher accuracy is good, but we need to pay for it
  - Everything is a trade-off.





Source:
1. Dai, Xiaoliang, et al. "Chamnet: Towards efficient network design through platform-aware model adaptation." (2019)
2. Cheng, Hsin-Pai et al. "SwiftNet: Using Graph Propagation as Meta-knowledge to Search Highly Representative Neural Architectures" (2019)

36

36

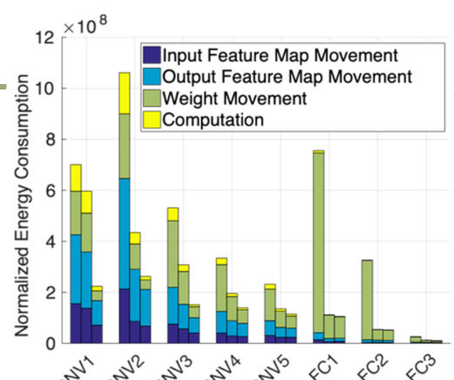
Size of model

- # FLOP: Number of floating point operations.
- # MAC: Number of multiply-and-accumulate operations
  - Usually, 1 floating-point multiply-and-accumulate is considered equivalent to 2 FLOPs.
- # Parameters
- Area [mm^2]

37

37

Energy efficiency



- Power consumption [mW]
- Energy is mainly used for
  - Calculation
  - Data movement
- Energy is a different thing:
  - A lower number of MACs does not necessarily lead to lower energy consumption.
  - Convolutional layers consume more energy than fully-connected layers.
  - Deeper CNNs with fewer weights do not necessarily consume less energy than shallower CNNs with more weights.

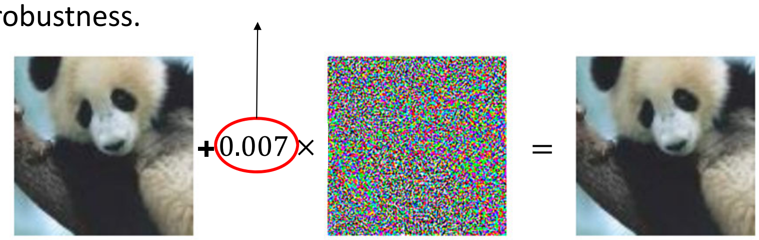
Source of image: http://eyeriss.mit.edu/

38

38

Robustness

- This parameter, is used to evaluate a neural network's robustness.



Panda noise gibbon

57.7% confidence 99.3% confidence

- Usually, a high accuracy model is not robust.
- Compare to the size of a neural network, the structure has more impact towards robustness.

Everything is a trade-off

Source of image: Explaining and harnessing adversarial examples

39

39

Outline

- Course introduction
- Machine learning & deep neural networks
  - Applications
  - Categories
  - Important metrics
  - Platforms & frameworks
    - Software platforms
    - Hardware computing devices

40

40

Software Platform

Chainer

PyTorch

TensorFlow

mxnet

Microsoft CNTK

Caffe2

41

41

Comparison of Machine Learning Framework

	PyTorch	mxnet	Chainer	TensorFlow
Supported interfaces	**C++, Python, Java, Rust, Go**	**C++, Python, Scala, Julia, Perl, MATLAB**	**C++, Python**	**C++, Python, Go, Java, Swift, JavaScript**
Multi-GPU: Data parallel	**Yes**	**Yes**	**Yes**	**Yes**
Multi-GPU: Model parallel	**Yes**	**Yes**	**Yes**	**Yes**
Developed by	facebook **AI research group**	The Apache Software Foundation	Preferred Networks, NVIDIA, intel, IBM	Google **Brain team**

S. Bahrampour, et al. Comparative Study of Deep Learning Software Frameworks. *arxiv*, 2016.

42

42

Tensorflow Pros/Cons

Pros

- **TensorBoard for visualization**
- **Data AND model parallelism; best of all frameworks**
- **Bigger developer community**

Cons

- Steep learning curve
- Usually slower than other frameworks right now
- Much "fatter" than Pytorch

Example TensorFlow operation types.

Category	Examples
Element-wise mathematical operations	Add, Sub, Mul, Div, Exp, Log, Greater, Less, Equal, ...
Array operations	Concat, Slice, Split, Constant, Rank, Shape, Shuffle, ...
Matrix operations	MatMul, MatrixInverse, MatrixDeterminant, ...
Stateful operations	Variable, Assign, AssignAdd, ...
Neural-net building blocks	SoftMax, Sigmoid, ReLU, Convolution2D, MaxPool, ...
Checkpointing operations	Save, Restore
Queue and synchronization operations	Enqueue, Dequeue, MutexAcquire, MutexRelease, ...
Control flow operations	Merge, Switch, Enter, Leave, NextIteration

43

43

Pytorch Pros/Cons

Pros

- **Dynamic approach to graph computation**
- **Usually faster than other DNN toolkits**
- **More convenient than Tensorflow**

Cons

- **Relatively smaller developer community compared to Tensorflow**
- **Less product oriented compared to Tensorflow or MXNet**

44

44

Pytorch vs. Tensorflow: MNIST

```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 20, 5, 1)
        self.conv2 = nn.Conv2d(20, 50, 5, 1)
        self.fc1 = nn.Linear(4*4*50, 500)
        self.fc2 = nn.Linear(500, 10)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.max_pool2d(x, 2, 2)
        x = F.relu(self.conv2(x))
        x = F.max_pool2d(x, 2, 2)
        x = x.view(-1, 4*4*50)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)
```

https://github.com/pytorch/examples/blob/master/mnist/main.py

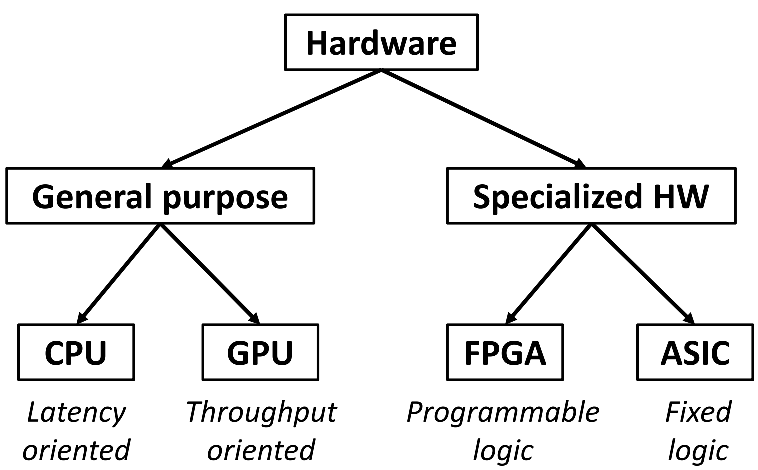
```python
def cnn_model_fn(features, labels, mode):
  """Model function for CNN."""
  input_layer = tf.reshape(features["x"], [-1, 28, 28, 1])
  conv1 = tf.layers.conv2d(
      inputs=input_layer,
      filters=32,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)
  pool1 = tf.layers.max_pooling2d(inputs=conv1, pool_size=[2, 2], strides=2)
  conv2 = tf.layers.conv2d(
      inputs=pool1,
      filters=64,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)
  pool2 = tf.layers.max_pooling2d(inputs=conv2, pool_size=[2, 2], strides=2)
  pool2_flat = tf.reshape(pool2, [-1, 7 * 7 * 64])
  dense = tf.layers.dense(inputs=pool2_flat, units=1024, activation=tf.nn.relu)
  dropout = tf.layers.dropout(
      inputs=dense, rate=0.4, training=mode == tf.estimator.ModeKeys.TRAIN)
  logits = tf.layers.dense(inputs=dropout, units=10)

  predictions = {
      # Generate predictions (for PREDICT and EVAL mode)
      "classes": tf.argmax(input=logits, axis=1),
      # Add `softmax_tensor` to the graph. It is used for PREDICT and by the
      # `logging_hook`.
      "probabilities": tf.nn.softmax(logits, name="softmax_tensor")
  }
```

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/exampl
es/tutorials/layers/cnn_mnist.py

45

45

Hardware computing devices



46

46

Hardware computing devices

Cost and speed are critical for both training and inference

GPU
- High power consumption
- Higher demand on data center cooling, power supply, and space utilization

CPU
- Medium cost
- Medium power consumption
- Low speed

ASIC
- High Non-recurring engineering
- Long design period, not suitable for fast iteration in NN development

FPGA
- Low power
- Low cost, Hundreds of dollars
- Hardware reconfigurable

47

47

ImageNet-1K Classification Performance

Platform	Inference Throughput	Peak TFLOPs	Effective TFLOPs	Power	Power Efficiency GOPs/J
Intel Xeon E5-2450	53 images/s	0.27T	0.074T (27%)	~225W	~0.3
Altera Arria 10 GX1150	369 images/s	1.366T	0.51T (38%)	~40W	~12.8
NVIDIA Titan X	4129 images/s	6.1T	5.75T (94%)	~250W	~23.0

Neural network is usually trained in back-end GPU clusters, while FPGA is very suitable for low-power real-time inference job

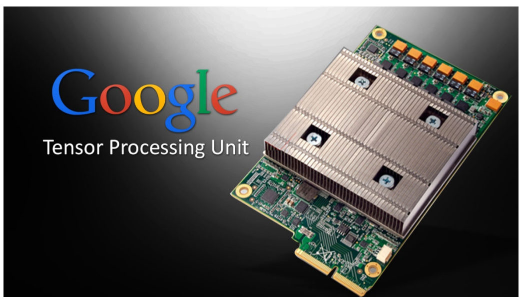
K. Ovtcharov, et al, Hot Chips Symposium (HCS), 2015

48

48

Tensor Processing Unit (TPU)

Unveiled during Google I/O Conference, Mountain View, CA (May 2016).
Tensor Processing Unit (TPU): a custom ASIC built specifically for machine learning — and tailored for TensorFlow.
This unit is designed for dense matrices, sparsity will have higher priority in the future.



(Google, https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html)

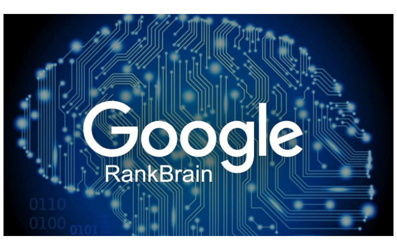
49

49

Tensor Processing Unit (TPU)

Applications:
1. RankBrain: improve the relevancy of search results.
2. Street View: improve the accuracy and quality of our maps and navigation.
3. AlphaGo: "think" much faster and look farther ahead between moves.

Server racks with TPUs used in the AlphaGo matches with Lee Sedol



TensorFlow

(Google, https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html)

50

50

Future

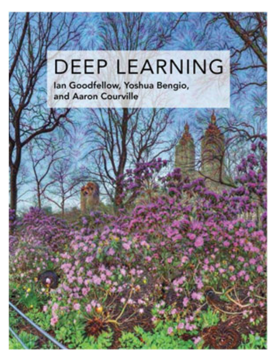
Smart **Low latency** **Privacy** **Mobility** **Energy efficient**

51

51

Reading material

- Deep Learning (2016), Ian Goodfellow and Yoshua Bengio and Aaron Courville http://www.deeplearningbook.org/
  - Chapter “Introduction”



52

52