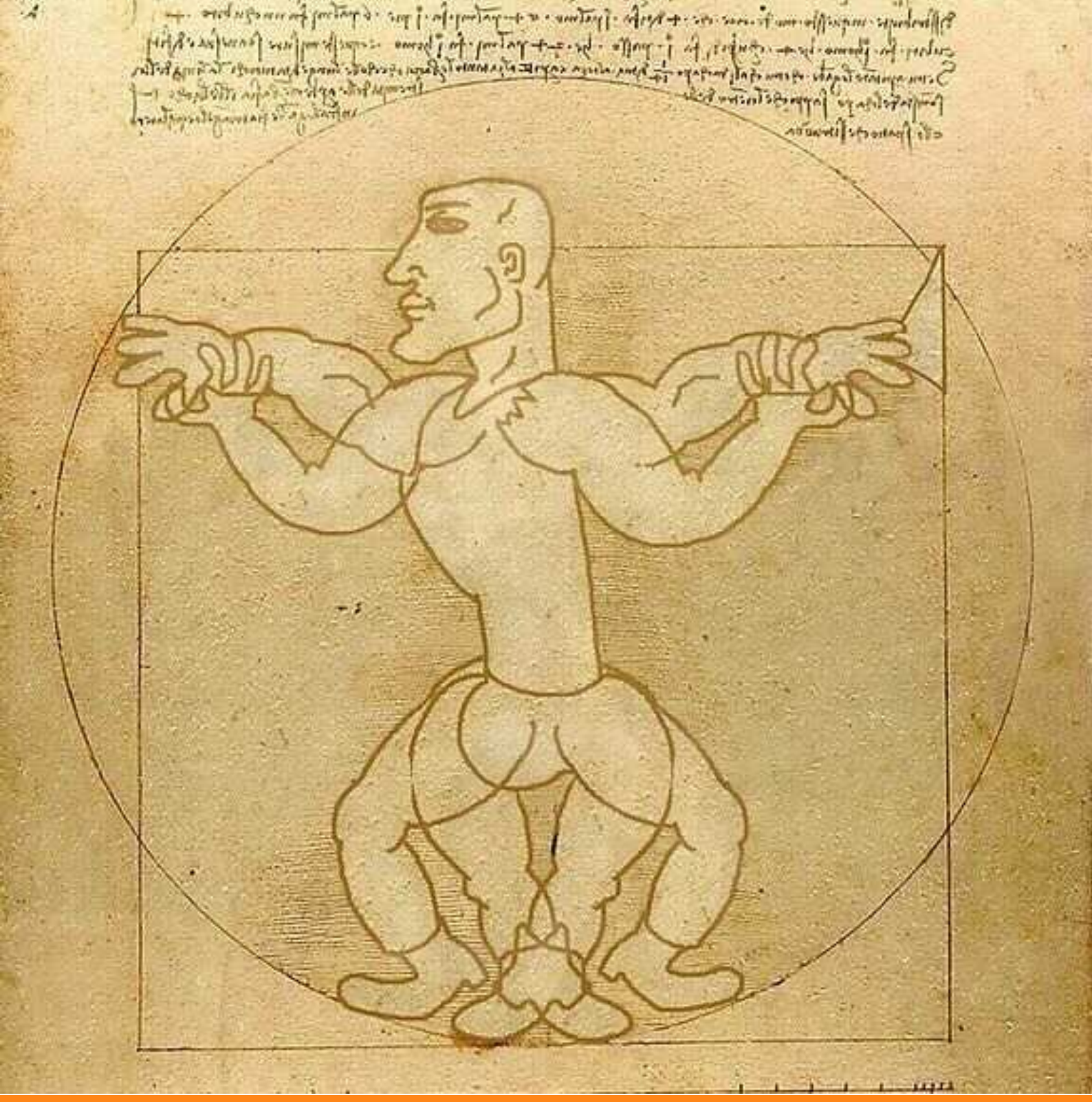


Commonplace Book - Unique Divine

Author: Unique Divine

Institute: Columbia University

Date: May, 2020 - present

Bio: http://uniquedivine.xyz/

Victory won't come to us unless we go to it.

Contents

Part I

Artificial Intelligence & Deep Learning

Chapter 1 Deep Reinforcement Learning

1.1 Getting Started

From Lex Fridman's intro lecture on deep reinforcement learning [video].

Deep reinforcement learning (RL) is perhaps one of the most exciting fields in artificial intelligence (AI). It marries the beauty and power of deep neural networks to represent and comprehend the world with the ability to act on that understanding. That's basically what the creation of intelligent beings is. Recent breakthroughs in RL captivate our imagination and inspire us.

RL at high level: An agent has to make a sequence of decisions.

Types of Learning: 4 learning domains within machine learning: Supervised, semi-supervised, unsupervised, and reinforcement. People often think that supervised learning is the only domain that requires manual labeling, however in reality, all types of machine learning are "supervised" in some way by a loss function. Every type of machine learning is supervised learning to a degree. These names for the domains tell us about the cost of human labor required to obtain that supervision.

- Supervised learning: "teach by example"; RL: "teach by experience"

RL in humans: Humans appear to learn to walk (and do many other activities) through "very few examples" of trial and error. **How** is an open question. Possible answers:

- Hardware: 230 million years of bipedal movement data
- Imitation learning: Observation of other humans walking
- Algorithms: Better than backpropagation and stochastic gradient descent

Current spot: https://youtu.be/zR11FLZ-O9M?t=757

3 Types of RL

There are countless ways to taxonomize all of the reinforcement learning algorithms. At the highest level, there are model-based algos and model-free algos. We generally put algorithms under 3 categories: model-based, value-based, and policy-based.

Model-based:

- Learn a model of the world, then plan using the model. As an agent interacts with the environment, it constructs a model for what the dynamics of the world might be.
- Better sample efficiency. Once you have a model, you can do all kinds of reasoning that doesn't require experiencing each scenario.

- In chess and in Go, the model is given to you. The agent knows the rules of the game.

Q: What is meant by "model" is model-based learning?

In this context, a model of the environment would be a function which predicts state transitions and rewards.

(cloze) AlphaZero is the reinforcement learning algorithm that learned to play Chess and Shogi.

(cloze) AlphaZero is a famous example of a model-based algorithm.

[AlphaZero paper, 2017]

Q: If model-based methods are more sample efficient when they work, what is the advantage of model-free methods?

Q: (cloze) Q-functions are also called action-value functions.

Q: And what's te difference between action-value function and state-value function?

THe state-value function returns the value of achieving a certain state, whereas the action-value function returns the value for choosing an action in a state.

Q: Why are Q-functions sometimes called action-value functions?

Because the Q-function gives us the value for taking an action in some state.

Note that, as of 2018, model-free methods are more popular and have been more extensively developed and tested than model-based methods.

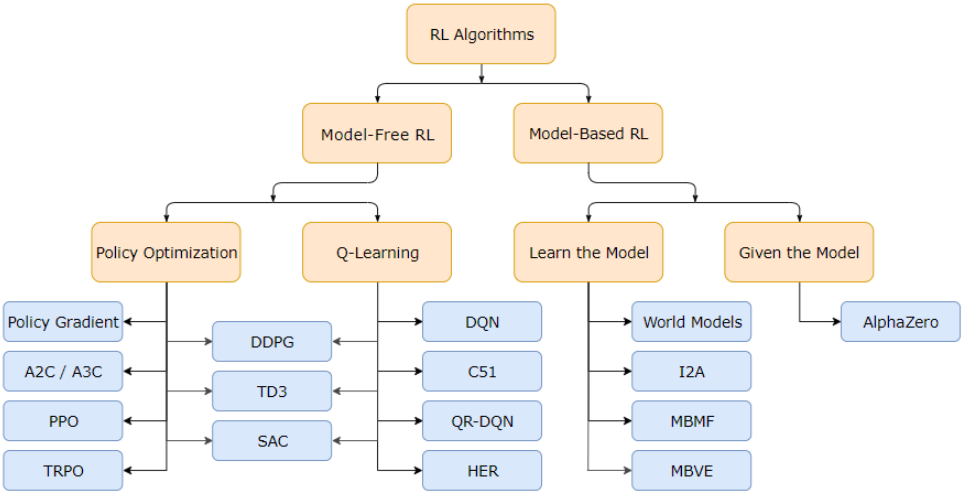
Value-based:

- Learn the state or state-action value. Value-based methods look to estimate the quality of states and the possible actions performed in them. This quality of the state is then used to pick a "best" action.
- Act by choosing the best action in a state. This is considered indirect learning of a "policy".

Policy-based:

- Learn the stochastic policy function that maps state to action.
- Directly learn a policy function. In other words, take as input the representation (or representation of that world) and output an action. This action will be stochastic.
- Act by sampling policy.
- Exploration is baked in. Why? The output action is stochastic, meaning it's a r.v.

Great resource for learning about deep reinforcement learning: Open AI Spinning Up [link]



1.2 Review Paper:

?

Deep learning is enabling reinforcement learning (RL) to scale to problems that were previously intractable, such as learning to play video games directly from pixels. Deep reinforcement learning algorithms are also applied to robotics, allowing control policies for robots to eb learned directly from camera inputs in the real world.

Topics covered: Value-based methods, policy-based methods, central algorithms in deep reinforcement learning such as the deep Q-network, trust region policy otpimisation, and asnchronous advantage actor-critic. Conclude with several current areas of research within the field.

Primary goal of the field of artificial intelligence (AI) is to produce fully autonomous agents that interact with their environments to learn optimal behaviours, improving over time through trial and error. Reinforcement learning is a principled mathematical framework for experience-driven autonomous learning. RL is a general way of approaching optimisation problems by trial and error.

The most important property of deep learning is that deep neural networks can automatically find compact low-dimensional representations (features) of high-dimensional data (e.g., images, text and audio)

the use of deep learning algorithms within RL defining the field of “deep reinforcement learning” (DRL)

For a more comprehensive survey of recent efforts in DRL, including applications of DRL to areas such as natural language processing [106, 5], we refer readers to the overview by Li [78]

The first, kickstarting the revolution in DRL, was the development of an algorithm that could learn to play a range of Atari 2600 video games at a superhuman level, directly from image pixels [84]. Providing solutions for the instability of function approximation techniques in RL, this work was the first to convincingly demonstrate that RL agents could be trained on raw,

high-dimensional observations, solely based on a reward signal. The second standout success was the development of a hybrid DRL system, AlphaGo, that defeated a human world champion in Go [128], paralleling the historic achievement of IBM's Deep Blue in chess two decades earlier [19] and IBM's Watson DeepQA system that beat the best human Jeopardy! players [31].

AlphaGo was composed of neural networks that were trained using supervised and reinforcement learning, in combination with a traditional heuristic search algorithm

In a step towards even more capable agents, DRL has been used to create agents that can meta-learn ("learn to learn") [29, 156], allowing them to generalise to complex visual environments they have never seen before

One of the driving forces behind DRL is the vision of creating systems that are capable of learning how to adapt in the real world.

1.3 Mine

Q-learning

Q: **(cloze) Q-learning is a model-free RL algorithm.**
(cloze) Q-learning gets is name because the agent learns the quality of actions to know how to act.
(cloze) In Q-learning, "Q" function is computed to maximize expected rewards for an action taken in a given state.
(cloze) "Model-free" means that an agent does not require a model of the environment.
(cloze) For set of states S and set of actions A, the Q-function is a map s.t. $Q : S \times A \to \mathbb{R}$.

Q: **In Q-learning, Q is updated according to**

$$Q_{\textbf{new}}(s,a) := Q(s,a) + \alpha \cdot \left(r + \gamma \max_{a_b \in A} Q(s', a_b) - Q(s,a) \right)$$

.

- **$Q(s,a)$: The "quality" fn. at the current state s**
- **α: The learning rate. $0 \leq \alpha \leq 1$. Values close to 1 make faster changes to Q.**
- **r: Reward received when moving from $s \to s'$**
- **γ: The discount factor. Quantifies how much to "discount" the**
- **$\max_{a_b \in A} Q(s', a_b)$: Estimate of optimal future Q-value. This would be the highest $Q(s'|a_b)$, where a_b is the "best" action and s' is the next state.**

1.4 DQN

?

1.5 Collection of Refs

Deep Attention Recurrent Q-Network. 2015. **?**

https://causalai.net/r26.pdf

Papers with Code - RL

Papers with Code - Representation Learning

Near-Optimal Representation Learning for Hierarchical Reinforcement Learning

Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model, 2020

"... deep RL algorithm based on the maximum entropy reinforcement learning framework. In this framework, the actor aims to maximize expected reward while also maximizing entropy. That is, to succeed at the task while acting as randomly as possible" - Soft Actor-Critic: Off-Policy Maximum ENtropy Deep RL w/ a Stochastic Actor

Chapter 2 Attention Mechanism

Attention is All You Need, 2016

Rethinking Attention with Performers

2.1 Image + Attention

2.1.1 Facebook AI Research applies Transformer architecture to streamline object detection models. 2020.

- Tags:
- Affiliations: Facebook AI
- [paper]
- [article]

2.1.2 Attention Agent: Neuroevolution of Self-Interpretable Agents. Tang, Ngyuen, and Ha. 2020.

- Tags: interpretability, evolutionary algorithm, transformer, self-attention, deep RL, image input
- Affiliations: Google Brain, Google Japan

2.1.3 CURL: Contrastive Unsupervised Representations for Reinforcement Learning. Srinivas et al., 2020.

- Tags: unsupervised, representation learning, deep RL, CNN, image input
- Affiliations

2.1.4 M-CURL: Masked Contrastive Representation Learning for Reinforcement Learning. Zhu et al., 2020.

- Tags: masked training, representation learning, sample efficiency, self-supervised, CNN, transformer, deep RL, BERT, contrastive learning, image input
- Affiliations: 1. Uniersity of Science an Technology of China. 2. Microsoft Research

Improving sample efficiency is a key research problem in reinforcement learning (RL). Contrastive Unsupervised representations for Reinforcement Learning (CURL).

Q: Besides the involvement masked training, what's the key difference between M-CURL and CURL?

M-CURL deals with videos (seqs of images) rather than individual images. Although consecutive frames are highly correlated, CURL handles them independently. M-CURL's

main improvement, outside of getting improved performance on several benchmarks, is that it takes into consideration the correlation between sequential frames.

This is where the transformer comes in. The Transformer, together with a CNN encoder, leverages the correlation of consecutive input frames to reproduce missing features in masked frames.

Q: Why use Transformers, specifically?

The input in this paper was a sequence of images rather than a single image. Transformers (Waswani et al., 2017) are the current state-of-the-art module for modeling sequences and capturing their interdependencies.

Q: The authors call the Transformer module an "auxilliary Transformer". What makes it auxiliary?

Q: CNN encoder of what? What's being encoded? And what is meant by "encode" here?

Q: Why discard the Transformer during action selection?

Q: Policy network? What does it do? What is it made up of? What are its inputs?

Q: Contrastive learning?

Chapter 3 Generative Adversarial Networks

3.0.1 GANs ?

Abstract & Introduction

Abstract: Goodfellow et al. propose a new framework for estimating generative models via an adversarial process, in which two models are simultaneously trained:

- G: a generative model that captures the data distribution, and
- D: a discriminative model that estimates the probability that a sample came from the training data rather than G

The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples.

Introduction: In the proposed adversarial nets framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution.

- **Counterfeiters Analogy:** The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

In this article, we explore the special case when the generative model generates samples by passing random noise through a multilayer perceptron, and the discriminative model is also a multilayer perceptron. We refer to this special case as adversarial nets. In this case, we can **train both models using only the highly successful backpropagation and dropout algorithms** [17] and sample from the generative model using only forward propagation.

Adversarial Net Algorithm

The adversarial modeling framework is most straightforward to apply when the models are both multilayer perceptrons. To learn the generator's distribution p_g over data $\boldsymbol{x}$, $p_g(\boldsymbol{x})$, we define a prior on input noise variables, $p_{\boldsymbol{z}}(\boldsymbol{z})$, then represent a mapping to data space as $G(\boldsymbol{z}|\theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. We also define a second multilayer perceptron $D(\boldsymbol{x}|\theta_d)$ that outputs a single scalar. $D(\boldsymbol{x})$ represents the

probability that $\boldsymbol{x}$ came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1 - D(G(\boldsymbol{z})))$. In other words, D and G play the following two-player minimax game with value function, $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} [\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} [\log (1 - D(G(\boldsymbol{z})))] .$$

Goodfellow et al. present a theoretical analysis of adversarial nets, essentially showing that the training criterion allows one to recover the data generating distribution as G and D are given enough capacity, i.e., in the non-parametric limit.

Figure 1 explains this approach: GANs are trained by simultaneously updating the discriminative distribution, D, so that it discriminates between samples from the data generating distribution, $p_{\boldsymbol{x}}$, and those of the generative distribution p_g (G).

After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate b/w the two distributions, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$.

Algorithm 1: Minibatch SGD training of GAN. The number of steps to apply the discriminator, k, is a hyperparameter. $k = 1$ is the least expensive option.

for *number of training iterations* **do**

  for k *steps* **do**

    # Apply discriminator

    - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior, $p_g(\boldsymbol{z})$.
    - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution, $p_{\text{data}}(\boldsymbol{x})$.
    - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log \left(D(\boldsymbol{x}^{(i)}) \right) + \log \left(1 - D(G(\boldsymbol{z}^{(i)})) \right) \right] .$$

  end

  # Update generator

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior, $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(G(\boldsymbol{z}^{(i)})) \right) .$$

end

The gradient-based updates can use any standard gradient-based learning rule.

Note:

- $k = 1$ was used in the experiments.
- Momentum was the gradient-based learning rule used in the experiments.

Related Work

The adversarial nets framework does not require a Markov chain for sampling. Because adversarial nets do not require feedback loops during generation, they are better able to leverage piece-wise linear units [19, 9, 10], which improve the performance of backpropagation but have problems with unbounded activation when used in a feedback loop.

?

Chapter 4 ML Finance Project

example w/ multivariate time series in PyTorch

Q: (cloze) Neural networks can be constructed using the `torch.nn` package.

Q: Import the package for constructing neural networks in PyTorch.

```
import torch.nn as nn
```

Q: (cloze) Seaborn comes with built-in datasets.

Q: Load seaborn's flights dataset.

```
flight_data = sns.load_dataset("flights")
```

Q: Why must time series data be scaled for sequence predictions?

When a network is fit on unscaled data, it is possible for large inputs to slow down the learning and convergence of your network and in some cases prevent the network from effectively learning your problem.

Q: sklearn import for scaling data?

```
from sklearn.preprocessing import MinMaxScaler
```

We know the field is fast moving. If the reader looking for more recent free reading resources, there are some good introductory/tutorial/survey papers on Arxiv; I happen to be compiling a list of them.

One of said review papers **?**

(**?**, hello)

Chapter 5 Bioinformatics

5.1 Deep Learning for Genomic Prediction

original title: DL for Genomic Risk Scores

> " A central aim of computational genomics is to identify variants (SNPs) in the genome which increase risks for diseases. Current analyses apply linear regression to identify SNPs with large associations, which are collected into a function called a Polygenic Risk Score (PRS) to predict disease for newly genotyped individuals. This project is broadly interested in whether we can improve performance of genomic risk scores using modern machine learning techniques.
>
> A recent study assessed the disease prediction performance of neural networks in comparison to conventional PRSs, but did not find evidence of improvement. This project will explore whether neural networks can improve performance by incorporating gene expression data to the training process. Gene expression is often integrated with SNP data in Transcriptome-Wide Association Studies (TWAS), which bear some resemblance to neural network architectures with SNPs as input nodes, genes as intermediate nodes, and disease status as the output node. Modeling this process as a neural network however will require defining a more unconventional architecture in which a small subset of hidden nodes is anchored to observed values.
>
> This project is designed for students with experience in machine learning topics and preferably with deep learning tools such as tensorflow or pytorch. Students should also be interested in applying machine learning and statistics to genomics applications." - Jie Yuan

Terms to know: Computational genomics, variants, single-nucleotide polymorphism (SNP), genome, Polygenic Risk Score (PRS), Transcriptome-Wide Association Studies (TWAS), gene(s), genomics, genotype, intermediate node (NN), hidden node (NN)

Q: Explain at a high level how PRSs tells us disease risk.

Q: "In what task were neural networks outperformed by conventional PRSs?"

Q: What is meant by "improve performance of genomic risk scores?"

genotype : An individual's collection of genes. Also can refer to the two alleles inherited for a particular gene.

node (NN) : An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input

of another[1].

hidden node (NN) : A node in a hidden layer.

hidden layer (NN) :

5.1.1 Papers

Polygenic Risk Scores (paper) ?

Abstract (mining)

recurrence risks : In genetics, the likelihood that a hereditary trait or disorder present in one family member will occur again in other family members[2].

"Evidence for genetic contribution to complex diseases is described by recurrence risks to relatives of diseased individuals."

This is distinguished from recurrence risk for cancer, which is the chance that a cancer that has been treated will recur.

gene : a sequence of DNA that codes for a specific peptide or RNA molecule; the physical and functional unit of heredity.

locus : the position of a gene on a chromosomes

somatic cell : any cell of the body except sperm and egg cells. A non-germline cell. any biological cell forming the body of an organism (except gametes).

sôma (Ancient Greek): body

genome : An organism's complete set of DNA, including all of its genes. Each genome contains all of the information needed to build and maintain that organism. In humans, a copy of the entire genome—more than 3 billion DNA base pairs—is contained in all cells that have a nucleus [3].

"genome-wide association"

allosome : (1) A sex chromosome such as the X and Y human sex chromosomes. (2) An atypical chromosome [4].

allo- (Greek): other, differnt

autosome : Any chromosome that is not a sex chromosome. The numbered chromosomes.

auto (Greek): self, one's own, by oneself, of oneself

-some, soma (Greek): body

allele : (genetics) One of a number of alternative forms of the same gene occupying a given position, or locus, on a chromosome.

Borrowed from German Allel, shortened from English allelomorph. Ultimately from the Ancient Greek prefix allēl- from állos ("other").

[1] https://en.wikipedia.org/wiki/Artificial_neural_network

[2] https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/recurrence-risk

[3] https://ghr.nlm.nih.gov/primer/hgp/genome

[4] https://www.merriam-webster.com/medical/allosome

"their effects and allele frequencies"

allelomorph: another term for allele.

risk loci :

"genome-wide association studies allow a description of the genetics of the same diseases in terms of risk loci..."

haploid : the quality of a cell or organism having a single set of chromosomes.

diploid : the quality of having two sets of chromosomes.

"Sexually reproducing organisms are diploid" (having two sets of chromosomes, one from each parent)

eukaryotes : Organisms whose cells have a nucleus enclosed within a nuclear envelope.

gamete : A mature sexual reproductive cell, as a sperm or egg, that unites with another cell to form a new organism. A haploid cell that fuses with another haploid cell during fertilization in organisms that sexually reproduce. A mature haploid male or female germ cell which is able to unite with another of the opposite sex in sexual reproduction to form a zygote.

gamete (Ancient Greek): to marry

zygote : A eukaryotic cell formed by a fertilization event between two gametes.

zygōtos (Greek): joined. yoked.

monozygotic : Monozygotic (MZ) or identical twins occur when a single egg is fertilized to form one zygote (hence, "monozygotic") which then divides into two separate embryos.

"monozygotic twins"

empirical :

"generate results more consistent with empirical estimates"

genetic variants :

Q: A human cell containing 22 autosomes and a Y chromosome is a sperm.

Neural Networks for Genomic Prediction (paper) ?

Transcriptome Wide Association

5.1.2 Meetings

Jie Yuan meeting #2 (Sep 10)

- There's a dot product and its output is passed through some activation function like sigmoid or ReLU. It's still linear in the sense that there's some sort of function that takes in a dot product (which is the definition of a linear operation). You can think of [the betas] as the weights in the neural network because the betas are the coefficients of a linear model.
- Liability is discussed in the multi-locus models paper.
- In the genomics context, this is called the **liability threshold model**. If you google that, you'll find some papers. In a broader machine learning sense, this is called the **probit regression model**. WIkipedia probably has a sufficient article on it.

- The gist of it is that it's a model to map continuous sums (the dot products basically) into binary labels: cases and controls. This can be any sort of logistic regression type thing where you have 0s and 1s. Basically, the **probit model is an alternative to the logistic regression model**.
- The liability is related to what's called the "link function". The link function in the probit model is the normal CDF function. There's a term that goes into the link function, and that term is what we're calling liability.
- The liability is basically the product of the vector of genotypes and the vector of betas. It's the linear part of the generalized linear model.
- Once you have the liability and plug it into the link function. Here, the link function is the probit regression. In a neural network, this would be the linear activation function (sigmoid, ReLU, etc.).
- You know how a sigmoid has a domain that's $(-\infty, \infty)$? It's range is (0,1), so what that function does is map a real value into being a probability. It gives you the probability of being a case. If you look at the normal CDF function, you find that it looks almost the same.

Why is the liability threshold model better than or different from the logistic reg model?

- One reason people use log reg more often is that the betas from log reg are interpretable. The have a meaning in terms of odds ratios. In a log reg model, the effect sizes are the natural log of the odds ratios.
- One disadvantage of log reg compared to liability-threshold is that log reg doesn't have a concept of the underlying distribution of the liability. In log reg, you get the linear $X\beta$ term, but there's no sort of distribution around it. In the liability-thresh model, we say that the liability, itself, is standard normally distributed. By taking the normal CDF, what you're doing is mapping that $X\beta$ value, the liability value, onto this distribution and asking, "what's the probability that it's larger than some threshold?"
- larger than threhsold $\implies$ label as 1, smaller $\implies$ 0.
- Because liability-threshold has that normal distribution, it gives you more concepts to play with such as variance explained, which the logistic reg model doesn't have.
- liability threshold model in simple terms
  from wikipedia: Liability threshold model

schizophrenia (SZ) example:

- Whether or not someone has SZ comes from a combination of factors in both their genes and environment.
- There are all sorts of variables. Imagine hypothetically that you could collect all of them and produce a score from that (that determines whether or not you have SZ). That score

would be the liability.

- The liability is a $\mathcal{N}(0, 1)$ distribution that, if higher than the threshold, predicts/indicates the patient has SZ.
- The problem with that is that we don't actually observe everything. We don't truly observe every factor that goes into whether you have SZ b/c (1) we can't measure one's environment and (2) we can't know all of the genomic risk factors from GWAS. We can only identify the largest risk factors.

  This means we are identifying only a small fraction of what goes into genetic risk for SZ.
- Essentially, what we have is a small subset of the factors that actually determine your SZ risk (which we're reasonably confident indicate this relationship). In this example, some subset of the risk factors like the genomic variance which increase your risk for SZ.
- What that means is that inside this standard normal liability distribution/function, what we actually know is a small subset of the factors that go into that. If we score people on that subset of factors, what we get is a smaller distribution, a distribution that has a variance that's a small fraction of the total liability variance, which we can't observe. The variance of that small fraction that we observe, that's the **variance explained**.
-

5.2 Computational Genomics (course, Rob Edwards)

This section details my notes from Rob Edwards's open source course at San Diego State University (Copyright 2018).

What will be learned in this course?

We'll use cutting-edge tools to analyze microbial genomes.

By the time we're done with the course, you'll have the ability to:

- use Amazon Web Services to analyze genomes
- use existing bioinformatics applications
- describe how algorithms used in bioinformatics work
- download, identify, and analyze data from public repositories
- critically analyze genomes and metagenomes

5.2.1 Central Dogma of Biology

[lecture video]

Dogma (def):

1. a principle or set of principles laid down by an authority as incontrovertibly true."the Christian dogma of the Trinity". "the rejection of political dogma." "the classic dogma of objectivity in scientific observation". "the difficulty of resisting political dogma".

2. Characterized by assertion of unproved or unprovable principles. A doctrine that is proclaimed as true without proof. "he believed all the Marxist dogma". "dogmatic writings".

In Essence: DNA is the genetic code. DNA is converted to messenger RNA (mRNA) through a process called transcription. There are two other types of RNA: tRNA and rRNA. mRNA is converted into proteins. Proteins are comprised of amino acids. In mRNA, nitrogenous bases consitute what's called a codon. A codon encodes for one amino acid. mRNA is read sequentially, one codon at a time, to give sequences of amino acids that make up a protein. [bonus] There's also a "reverse" transcription that certain viruses can do, where mRNA is converted back into DNA. Humans don't normally do that. Bacteria don't normally do that. Viruses do. **Bottom Line:** DNA is essentially the standard code for all living organisms as far as we know.

DNA's alphabet: A C G T

RNA's alphabet: A C G U

DNA gets transcribed into RNA in a sequence-depended fashion.

DNA has a direction. One side of a strand is 5', "the 5 prime band", and one is 3'. This direction is 5' to 3'.

DNA lines up in pairs of sequences with bases aligned in complementary pairs. These pairs are called "reverse compliments"

RNA is made with what's called a template strand.

5.2.2 What does it mean to sequence a genome?

Q: What's a genome?

Humans have 23 pairs of chromosomes. Each pair consists of a chromosome from each parent.

All of the DNA contained in one cell is called the genome. We have one copy of the genome in nearly every cell in our body. Human genomes are $\approx 99.8\%$ identical to that of every other human being. The other 0.2% of the genome is what is of high interest to healthcare professionals as understanding it can help in the prediction, prevention, diagnosis, and treatment of disease.

Q: e

- Human genome 3.1 billion bp (base pairs), i.e. 3.1 Gbp
- bacteria 100 kbp - 2Mbp
- This means that the computational overhead of studying bacterial genomes is much smaller and can be done on a standard personal computer.

Part II

Math & Code

Chapter 6 Java

Let's dissect the following code block that sums two numbers.

```
import java.util.*;
public class Solution {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int a, b;
        a = in.nextInt();
        b = in.nextInt();
        System.out.println(a + b);
    }
}
```

Q: What does `String[] args` mean in the hello world program?

`args` is an array of strings.

Arrays

Q: Declare an array of strings.

```
String[] arr;
```

Q: Initialize an array of integers containing 1, and 2.

```
int[] arr = {1, 2};
```

Q: Initialize an array of strings containing "BMW" and "Ford".

```
String[] carBrands = {"BMW", "Ford"};
```

Q: Given an array, `String[] carBrands = {"BMW", "Ford"};` print the first element.

```
System.out.println(carBrands[0]);
```

Q: `in.nextInt()` ?

content

It may be a good idea to go through all of this: https://www.w3schools.com/java/default.asp

6.1 Gradle

https://youtu.be/aYu994I8Z6I?list=PLMxpKvJf0K0QUyvmkKZu7WpwTVzdePGTl

- Gradle is an pen-source build atomation tool.
- Gradle is the official android build tool; maintained by Android SDK Tools team
- Gradle build scripts are written using Groovy

WHy Gradle?

- highly customizable and extensible
- used for multiple languages
- it's fast: reuses outputs from previous executions, processing only inputs that changed; executes tasks in parallel
- higher performance than its peers (such as Maven)

Gradle project tasks

Chapter 7 Git

Git (cookbook)

7.1 Fundamental Concepts

7.1.1 Local branch vs. remote branch

- **local branch**: a branch only the local user can see. It exists only on your local machine.
  - Ex. Create local branch named "myNewBranch":
    `git branch myNewBranch`
- **remote branch**: a branch on a remote location (in most cases 'origin'). Local branches can be pushed to 'origin' (a remote branch), where other users can track it.
  - Ex. Push local branch, "myNewBranch", to the remote, "origin" so that a new branch named "myNewBranch" is created on the remote machine ("origin"):
    `git push -u origin myNewBranch`
- **remote tracking branch**: A local copy of a remote branch.
  When 'myNewBranch' is pushed to 'origin' using the command above, a remote tracking branch named 'origin/myNewBranch' is created on your local machine.

local tracking branch: a local branch that is tracking another branch.

source(s): SNce & Brian Webster. stackoverflow.com

7.1.2 HEAD, master, and origin

I highly recommend the book "Pro Git" by Scott Chacon. Take time and really read it, while exploring an actual git repo as you do.

- **HEAD**: the current commit your repo is on. Most of the time HEAD points to the latest commit in your current branch, but that doesn't have to be the case. HEAD really just means "what is my repo currently pointing at".
  In the event that the commit HEAD refers to is not the tip of any branch, this is called a "detached head".
- **master**: the name of the default branch that git creates for you when first creating a repo. In most cases, "master" means "the main branch". Most shops have everyone pushing to master, and master is considered the definitive view of the repo. But it's also common for release branches to be made off of master for releasing. Your local repo has its own master branch, that almost always follows the master of a remote repo.
- **origin**: the default name that git gives to your main remote repo. Your box has its own repo, and you most likely push out to some remote repo that you and all your coworkers push to. That remote repo is almost always called origin, but it doesn't have to be.

- `HEAD` is an official notion in git. `HEAD` always has a well-defined meaning. `master` and `origin` are common names usually used in git, but they don't have to be.

source: HEAD, master, and origin. Matt Greer & Jacqueline P. via stackoverflow.com

7.1.3 Large File Storage

`https://git-lfs.github.com/`

7.2 Git§2

7.2.1 SSH keys

An SSH key is an alternative to username/password authorization on GitHub. This will allow you to bypass entering your username and password for future GitHub commands.

SSH keys come in pairs, a public key that gets shared with services like GitHub, and a private key that is stored only on your computer. If the keys match, you're granted access.

The cryptography behind SSH keys ensures that no one can reverse engineer your private key from the public one.

SSH Keys for GitHub [article]

Generating a new SSH key: Follow Generating a new SSH key and adding it to the ssh-agent [article]

7.3 Permanently removing files from commit history

WHy do this? You may have commited a password, some other sensitive information, or a large file that you want to remove from github. If the change is only a few commits back, you can "rebase" changes out of the history. I actually need to do this for a much older set of files and accidentally uploaded a textbook that takes up almost a GB of space.

`git filter-branch -force -index-filter "git rm -cached -ignore-unmatch PathToSensitiveFile" -prune-empty -tag-name-filter cat -- -all`

All you need to change is the `PathToSensitiveFile` item. Once you've used this command for all of the files you'd like to get rid of, update the origin by typing `git push origin -force -all` .

7.4 Branching

Suppose your application is stable. Later, you discover a gigantic bug that was passing silently. You want to write some tests, fix the bug, and eventually have a stable, passing application once again. To do this, you'd create a branch for the fix and push the branch to the remote so that all of the developers on your team can collaborate and make the fix.

Once all of the necessary changes have been made and the application is stable, someone from the team would commit merge the commits from the other branch into master. Since the commit history from the branch will have been saved to master, the new branch could be deleted without loss of information (if you no longer wanted to work on this branch).

View branches: `git branch`

Switch branches: `git checkout [branch-name]`

7.4.1 Merging

Merge the specified branch's history into the current one. `git merge [branch]`

7.4.2 Deleting branches

Delete local branch `git branch -d [branch-name]`

Delete remote branch `git push origin –delete [branch-name]`

Chapter 8 C++

C++ source code files end with a .cpp extension.

Hello world program: Run these and find out which one works.

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

Compiling and executing the C++ program:

1. Step 1 is to install the gcc compiler.

   `hi`

2. Verify the install of g++ and gdb with `whereis g++` and `whereis gdb`

   To install gdb (linux or WSL), use `sudo apt-get install build-essential gdb`

Open MP To import:

```
#include <omp.h>
```

People use OpenMP for shared memory parallelization.

Header files

C++ programs consist of more than just .cpp files. They also use **header files**, which can have a .h extension, .hpp extension, or even none at all.

Q: What is a `.h` file?

header file

Q: What is the purpose of a header file?

Header files allow us to put declarations in one location and then import them wherever we need them. This can save a lot of typing in multi-file programs.

Q: What's contained in a `.h` file?

..

References & Further Reading

- C header files
- learncpp.com/.../header-files

Chapter 9 Web development

9.1 Hugo Web Design

Start Sep 16 (Mike Dane Tutorial Series)

9.1.1 Intro to Hugo -(video)

- Hugo is a static site generator.
- Static website generators allow you to compromise between writing a bunch of static html pages and using a heavy, and potentially expensive, content management system.
- Why Hugo? It's extremely fast.
- 2 kinds of websites, dynamic and static. Dyanmic ex. Facebook. Facebook pages are dynamically generated for each user. For static websites, what you see is what you get.
- Static websites are notoriously harder to maintain b/c you lack some of the flexibility of things on a dynamic site. Usually you can't use much conditional logic, functions, or variables.
- However, static pages are extremely fast.
- Hugo is great for a blog, portfolio website, etc.
- Hugo doesn't explicity require you to write a single line of HTML code.
- Flexibility | With that said, if you want to go in and change every little detail of the layout of the site, you cna do that. You can write as much of the HTML and have as much control as you'd like.
- Hugo is 100% free and open-source.

9.1.2 Intalling Hugo on Windows - (video)

- Mine's already installed. I'll skip this for now.
- a
- a

9.1.3 Creating a new site - (video)

- skip, a bit too easy

9.1.4 Installing & Using Themes - (video)

- my themes are already installed

9.1.5 Creating & Organizing Content - (video)

- Hugo has 2 types of content: single pages and list pages
- List content lists other content on the site. You can call this a list page.
- Individual blog posts are single pages.
- Your posts should not just be in the content directory. They should be in directories inside the content directory.
- A list page is automatically created for directories inside the content folder. Hugo automatically does this. Note, this only occurs for directories at the “root" level of the content directory. For example: `content/post/` would generate a list page at `site.com/post/`, but `content/post/dir0` would not.
- If you want a list page to be generated for a dir that is not at the root level of the content dir, you have to create an **index filed**, `_index.md`. For a convenient and efficient way to do this from the cmd, use `hugo new post/dir0/_index.md` (above above example), then there will be a list page for dir0. Content can also be added to`_index.md` and it should show up on the page.
- Additionally, for list pages that are aautomatically generate by hugo, you can edit the content by adding an index.md to those as well. Ex. `content/post/_index.md`.

9.1.6 Front Matter- (video)

- Front matter in Hugo is what is commonly called meta data.
- Front matter is data about our content files.
- The metadata automatically generated by Hugo at the top of md files when using `hugo new` is front matter
- Front matter is stored in key-value pairs
- Front matter can be written in 3 different languages: YAML, TOML, and JSON
- The defualt lang for front matter in Hugo is YAML
- YAML - indicated by “–”,
- TOML - indicated by "+++" and uses "=" instead of ":",
- JSON - indicated
- You can create your own custom front matter variables.
- Front matter is super powerful in its utility.

9.1.7 Archetypes - (video)

- How does the default front matter from using `hugo new .md` get selected? Short answer: archetypes
- An archetype is basically the default front matter template for when you create a new content file.

- Archetypes are modified under `static/themes/archetypes/default.md`
- Suppose your content dir has a subdirectory, `content/dir0`. If you wanted to create an archetype for the files in dir0, you'd simply create `dir0.md` inside the archetypes dir.

9.1.8 Shortcodes - (video)

- Shortcodes are predefined chunks of HTML that you can insert into your markdown files.
- Let's say you have a md file that you want to spice up by adding in some custom HTML. For instance, maybe you'd like to embed a YouTube video. Normally this would require lots of HTML that you'd have to paste it. Shortcodes can allow you to sidestep this. Hugo comes with a YouTube video shortcode predefined.
- General shortcode syntax `< shortcode_name param0 >`
- Youtube shortcode | For a YouTube video with url, "youtube.com/watch?v=random-text", the shortcode we'd use to embed would be `< youtube random-text >` because "random-text" is the id of the youtube video and the only parameter for that shortcode.

9.1.9 Taxonomies - (video)

- Taxonomies in hugo are basically ways that you can logically group different pieces of content together in order to organize it in a more efficient way.
- Hugo provids 2 defualt taxonomies: tags & categories
- All taxonomy information is declared in front matter. In YAML, tags has the syntax `tags: ["tag0", "tag1", ...]`

9.1.10 Templates - (video)

- Templates here mostly refers to HTML templates. If you're not comfortable writing HTML, CSS, and coding for the web, templates might be a little bit above your head.
- A hugo theme is actually made up of hugo templates.
- Any template that you use in Hugo is going to be inside `themes/theme-name/layouts`. This is where all the templates live.
- `/layouts/default` usually contains a default style for list and single pages by use of `list.html` and `single.html`.

9.1.11 List Templates - (video)

- List templates give default HTML layout to list content files.
- .

Resources

- A clear and concise beginner hugo tutorial

- CSS Crash Course for Absolute Beginners

9.2 HTTP, TCP, IP

9.2.1 HTTP

Q: (cloze) HTTP = Hypertext Transfer Protocol

Q: Give me a brief overview of HTTP.

HTTP is a protocol that allows the fetching of resources such as HTML documents. It allows web-based apps to communicate and exchange data.

HTTP is a client-server protocol. This means requests are initiated by a recipient, usually the web browser, and a complete document is constructed from the files fetched such as text, layout description, images, videos, scripts, etc.

- The client is the one making the request.
- The server responds to this request.

Three important things about HTTP

1. HTTP is connectionless: After making the request, the client disconnects from the server. Then when the response is ready, the server re-establishes the connection again and delivers the response.
2. HTTP can deliver any sort of data.
3. HTTP is stateless: The client and server know about each other only during the current request. If the current request closes and the two computers want to connect again, they need to provide information to each other anew. In other words, statelessness means that the connection between the browser and the server is lost once the transaction ends.

To learn about requests and repsonses: `https://youtu.be/eesqK59rhGA?t=275`

9.2.2 TCP/IP

The internet protocol suite is the conceptual model and set of communications protocols used in the internet and similar computer networks. It is commonly known as TCP/IP because the foundational protocols in the suite are the Transmission Control Protocol (TCP) and Internet Protocol (IP).

Communication prootocl: A system of rules that allow two or more entitites of a communications system to transmit information via any kind of variation of a physical quantity.

Internet Protocol (IP): IP has the task of delivering packets from from the source host to the destination host solely based on the IP addresses in the packet headers. For this purpose, IP defines packet structures that encapsulate the data to be delivered.It also defines addressing methods that are used to label the datagrame with source and destination information.

- (network) packets: Formatted units of data carried by a packet-switched network. A packet consists of control information and user data; the latter is also known as the payload. Control information provides data for delivering the payload (e.g. source and destination network addresses, error detection codes, or sequencing information).

Q: What's an IP address?

An Internet Protocol address, or IP address, is a numerical label assigned to each device connected to a computer network tha tuses the Internet Protocol for communication.

Bandwitdth (computing): (Computing) bandwitdth is the maximum rate of data transfer across a given path. Bandwidth can be characterized as network bandwidth, data bandwidth, or digital bandwidth.

Computing bandwidth is different from the bandwidth defined in the field of signal processing, signal bandwidth. Signal bandwidth is the frequency range between lowest and highest attainable frequency while meeting a well-defined impairment level in signal power. It's measured in hertz.

In relation to internet protocol, we often talk about consume dbandwidth in bit/s, which corresponds to achieved throughput or goodput, i.e. the avg rate of successful data transfer through a communication path.

Q: Why is network bandwidth an average rate instead of a current rate?

A channel with v_β may not necessarily transmit data at v_t rate since protocols, encryption, and other factors can add overhead. For instance, internet traffic often uses TCP, which requires a three-way handshake for each transaction. TCP is efficient, but it does add significant overhead compared to simpler protocols. Additionally, data packets may be lost, further reducing the data throughput.

Q: Packet loss?

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is either caused by errors in data transmission, typically across wireless networks, or network congestion. Packet loss is measured as a percentage of packets lost with respect to packets sent.

Q: Packet switching?

Packet switching is a method of grouping data that is transmitted over a digital netowrk into packets. Packets are made of a header and a payload. Data in the header is used by networking hardware to direct the packet to its destination, where the payload is extracted and used by application software. Packet switching is the primary basis for data communications in computer networks worldwide.

Q:

Resources

- https://en.wikipedia.org/wiki/Internet_protocol_suite
- .

Chapter 10 Algorithms

10.1 Algorithms

"4.5 years of learning programming and working as fullstack software engineer ... had interview with one of the FAANG companies this summer in Hong Kong but failed it due to the fact that I suck in DSA (Data Structures & Algorithms)."

"I'm using to leetcode.com to learn data structures and algorithms since I got a rejection from FAANG after interviewing with them onsite."

Role of DSA in Programming (July, 2020)

10.2 Design Patterns

Q: Why use "design patterns"?

- Design patterns let your write better code more quickly by providing a clearer picture of how to implement the design
- Design patterns encourage code reuse and accomodate change by supplying well-tested mechanisms for delegation, composition, and other non-inheritance based reuse techniques
- Design patterns encourage more legible and maintainable code

Q: Delegation? Composition?

- delegation: a pattern where a given object provides an interface to a set of operations. However, the actual work for those operations is performed by one or more other objects.
- composition: Creating objects with other objects as members. Should be used when a "has-a" relationship appears.

Q: What are design patterns?

content

Q: Which resources will you use to start learning about design patterns?

GOF patterns (C++). Then, potentially Head First Design Patterns (Java)/

10.2.1 References & Further Reading

Introduction to Design Patterns Course

Chapter 11 Miscellaneous

11.1 HTTP, REST, & Spring Boot

Q: (cloze) REST has become the de-facto standard for building web services. Why? Easy to build, easy to consume.

Q: What is a micro service and why do we use them?

Microservice is an architecture that allows the developers to develop and deploy services independently. Each service running has its own process and this achieves the lightweight model to support business applications.

Microservices, a.k.a. the microservice architecture, structure an application as a collection of services that are

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team

Q: What is Spring Boot?

An open source Java-based framework used to create microservices.

Q: Why use Spring Boot?

- manages REST endpoints
- flexible way to configure database transactions
- auto configured as opposed to manually configured
- eases dependency management

Q: How does Spring Boot easy dependency management?

Spring Boot automatically configures your application based on the dependencies you have added to the project by using @EnableAutoConfiguration annotation. For example, if MySQL database is on your classpath, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

The entry point of the spring boot application is the class contains @SpringBootApplication annotation and the main method.

11.1.1 References & Further Reading

Building REST services with Spring.

What are microservices?

Spring Boot - Introduction