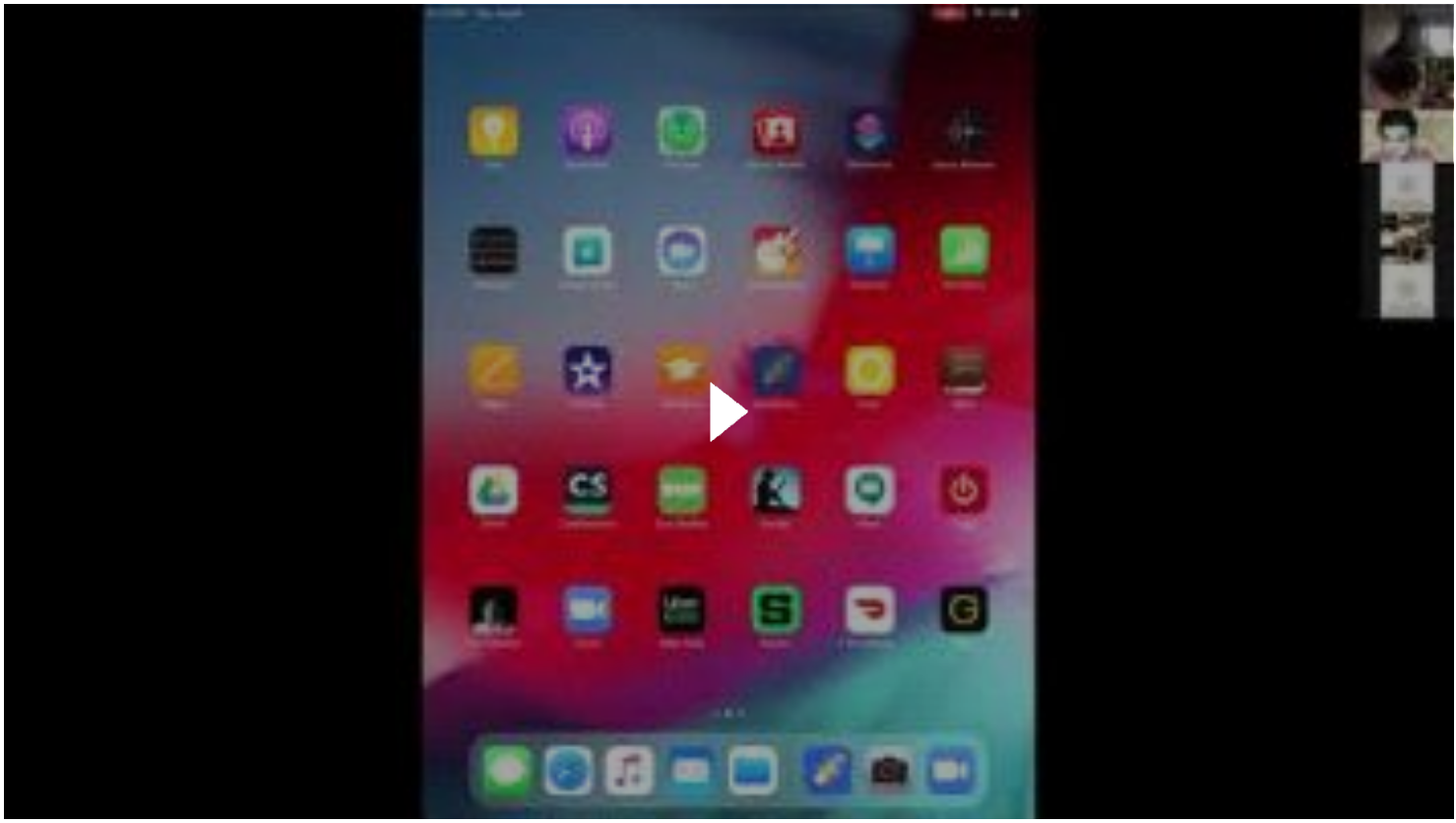
Live Lecture (Week 7)

Sunday, January 8, 2023 12:00 PM

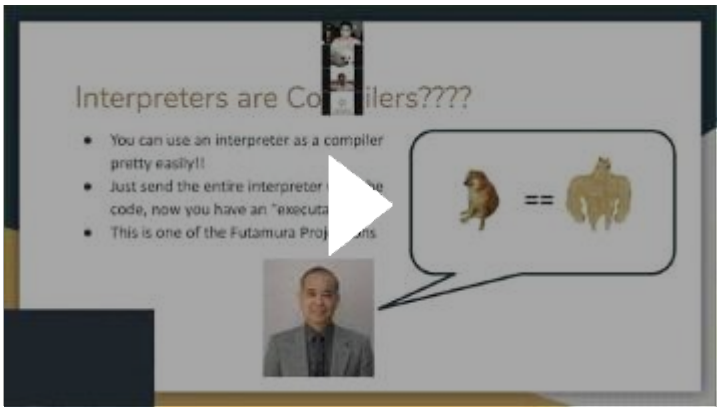
CS61A Summer 2020: Live Lecture (Week 7)



Special Topics Lecture I: Programming Languages

Sunday, January 8, 2023 12:04 PM

[CS61A Summer 2020: Special Topics Lecture I: Programming Languages]



Interpreters and Compilers

Interpreters
Some underlying language X e.g., python
Used to execute code in a language Y e.g., scheme
Why?
More languages = better
Lets you interpret new languages

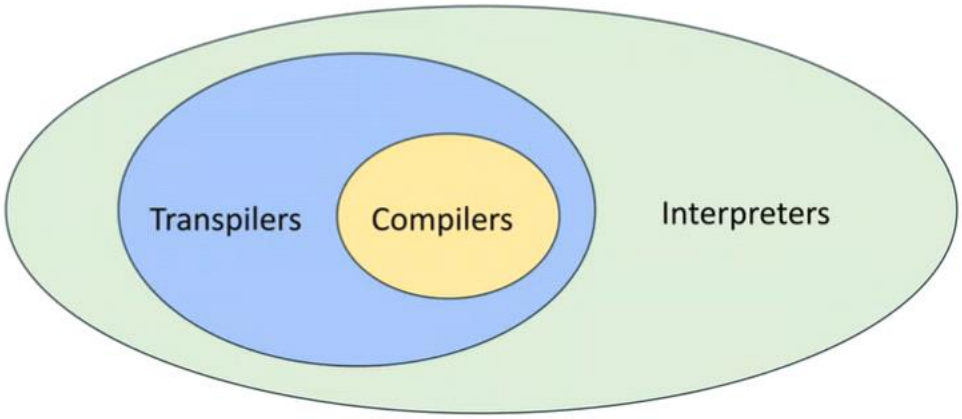
Compilers
Take code in language Y and convert it to machine code
This is called an "executable"
Can be run stand-alone from that point on
How all languages eventually work

Compiler Pros and Cons

Compiler Pros
Faster to run the code
Don't need to send an interpreter to your users
    Just the executable that the compiler produces
Interpreter Pros
    Easier to implement

Transpilers
What about a transpiler
Converts code from one language to another
e.g., convert something like "define x (= x 2)" "x = 2"
Is this a compiler or interpreter

Transpilers / Compilers / Interpreters
Compilers are obviously Transpilers
Transpilers are Interpreters



Hierarchy

Interpreters are Compilers???
You can use an interpreter as a compiler pretty easily!!
Just send the entire interpreter with the code, new you have an "executable"
This is one of the Futamura Projections

How does python work?
It's interpreted !
Python3 is an interpreter written in the programming language C
C is a compiled language
Python frames are allocated as objects in C
    It is handled like a dictionary
There is a python_eval function written in C

```
PyObject *
_PyEval_EvalCode(PyThreadState *tstate,
           PyObject *_co, PyObject *globals, PyObject *locals,
           PyObject *const *args, Py_ssize_t argcount,
           PyObject *const *kwnames, PyObject *const *kwargs,
           Py_ssize_t kwcount, int kwstep,
           PyObject *const *defs, Py_ssize_t defcount,
           PyObject *kwdefs, PyObject *closure,
           PyObject *name, PyObject *qualname)
{
    assert(is_tstate_valid(tstate));

    PyCodeObject *co = (PyCodeObject*)_co;

    if (!name) {
        name = co->co_name;
    }
    assert(name != NULL);
```

How does is work?
C has an is operator, confusingly called ==
Thus python is can be directly implemented in terms of C's ==

How does + work?
C an operator called +
But it's bad
In C, the integer "overoflows" so values that are too big become negative
SO python's + is implemented as a function
    Basically it does the addition algorithm you learned in elementary school

HOW does if work?
In terms of C's if !

Programming language Features

Variables
More or less all languages have variables
But not all do!
In MIPS here's how to write the featured_distance function
    MIPS is an assembler language, very close to what the machine reads

CS61C will introduce the assembler

Functions
Not all languages have function
These are really a convenience
Technically you can write anything in python without any function

Recursion
Technically, you can have functions without recursion!!
This hasn't been a thing since 19 70's
The original FORTRAN always reuses function frames, so it cannot handle recursion at all
    No concept of multiple frames of the same function being open at the same time.

Memory Management
Python and Scheme both have what is known as "memory management"
Code on the right doesn't "load memory"
    When the hailstone_sum frame loses elements can be deallocated
    They can be removed from memory
In languages like C you would have to do that manually

```
def hailstone_sum(x):
    elements = [x]
    while x > 1:
        if x % 2 == 0:
            x //= 2
        else:
            x = 3 * x + 1
        elements.append(x)
    return sum(elements)
```

Type safety
Python and scheme both have what is known as type safety
"2" + 3 gives a really big number
C, on the other hand, adds to the memory location "2" is at to 3
    This is called type unsafe

Exceptions
Control flow that can break the bounds of a function

Usually for error handling
Python uses raise

Macros
Scheme has macros, but as an alternative to functions
TeX only has macros!!
    Kinda an HTML for PDF
C has macros but they are just basic text substitution
    #define until(x) while (not (x))

Coroutines
When two functions call trade control off to each other
Python does this with generator which can actually send data one ways

Lazy Evaluation
Tail recursion is kinda a (very limited) version of this
Streams are a more fleshed-out version of this

Lazy errors

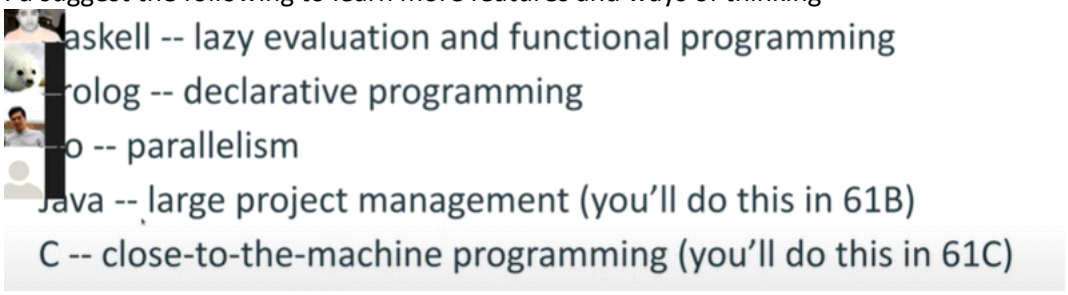
Lazy infinite data
This is like streams in scheme, but everything is lazy
Can't handle effects very well

Parallelism
You can run multiple pieces of code at the same time
Saves time
Allows multiple programs to happen at the same time!

Infinite Diversity in Infinite Combinations

Python and scheme are pretty advanced

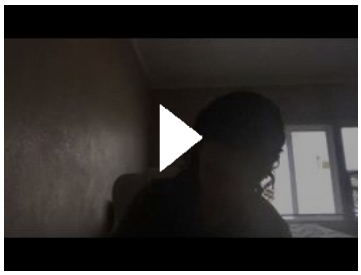
I'd suggest the following to learn more features and ways of thinking



CS61A Summer 2020: Special Topics Lecture II: Machine Learning

Friday, January 13, 2023 10:23 PM

CS61A Summer 2020: Special Topics Lecture II: Machine Learning



Machine Learning
Focus on supervised ML

Motivation Regression
Overview of Neural Networks

We will heavily rely on abstraction to avoid the math behind these concepts

How we talk about computing in this class

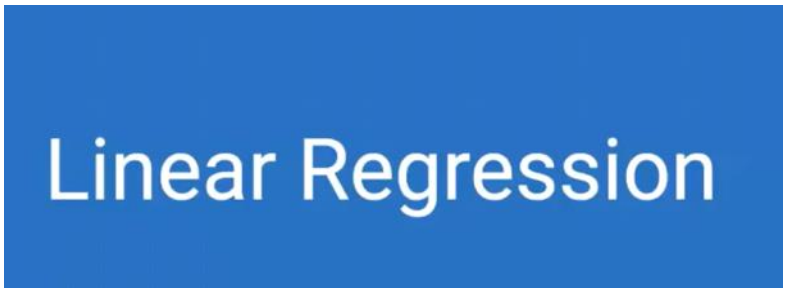
Traditional Arithmetic vs. ML

	x	f	f(x)
Arithmetic	3	f = lambda x: x**2	**9**
Machine Learning	3	**?**	9

What is “Machine Learning?”

- A very big buzzword
- Ultimately just a different problem we are trying to solve
- Instead of trying to find f(x), we are trying to find f
- A Statistician would call this an **inference problem** - given a set o data (x, f(x)), we are trying to infer what f is from that data

But how do we do this??



Like algebra
Y = mx+b

$$final\_score = a \times diagnostic\_score + b \times midterm\_score + c$$

If we can determine what a b and c are then we have our function because our function is just an input of the diagnostic whic h I'll abbreviate with diag and then midterm which I'll abbreviate with mid

$$f(diag, mid) = a \times diag + b \times mid + c$$

So in order to find out function f we have to find the values of a b and c
And once we done that we have our linear function

Geometry
First term, range of values

$$diagnostic\_score \in [0, 100]$$

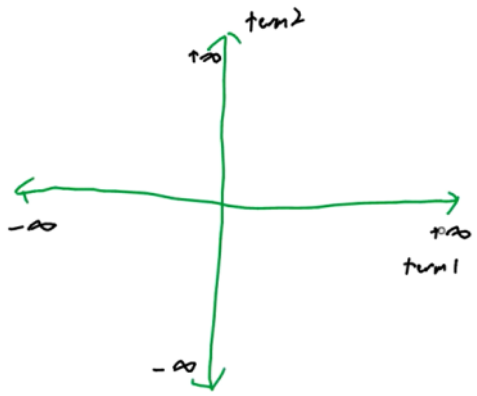
Since Should be percentage

$$a \in (-\infty, \infty)$$

$$a \times diagnostic\_score \in [-\infty, \infty]$$

so the second term

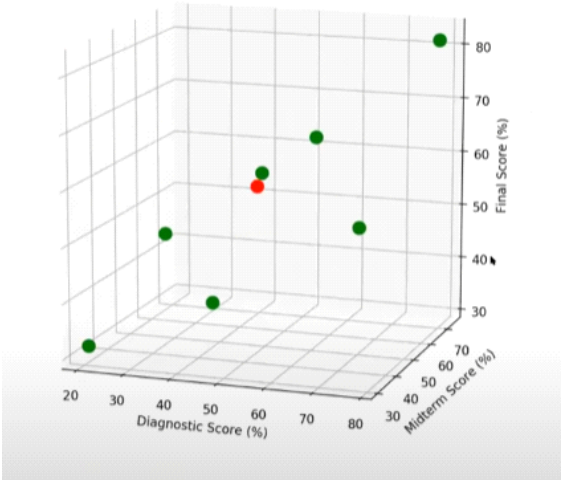
Draw that out



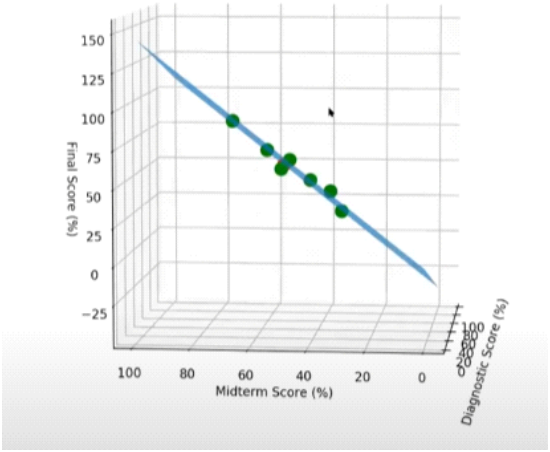
Two-dimensional

For the third term C
Allow us to move up and down within a third axis a third dimension

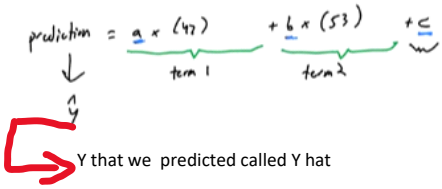
First two terms each one is its own axis and each one allows us to kind of move around on an infinite claim



Roughly decent positive linear relationship
So generally do better on the diagnostic and midterm will generally do better on the final



How can we write some type of mathematical equation that will have it so that we have this plane that represents our predictions being very close to the data



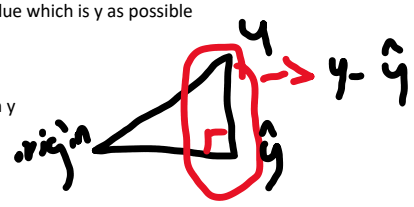
Y that we predicted called Y hat

As it can also hypothesize that there's a light in my room and if you want to get close to it then should firstly stand underneath it, standing directly underneath it and while maintaining being on the floor

If we want to try and get our prediction
Y hat which represents the floor in this case as close to the actual value which is y as possible

We need to have y hat be directly underneath

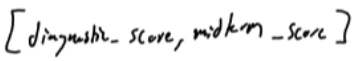
That say we want prediction that equation to be directly underneath y

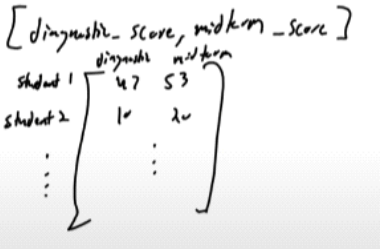


Write it Mathematically
So that we should know what variables we need to be as close to our actual value

All have to be in this plane that we defined up here
The entire prediction has to be somewhere within the plane

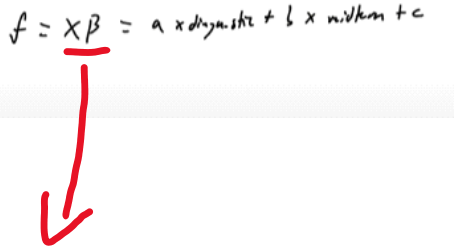
X will be matrix, think about it as a list of lists





The way that go about determining regression in statistics will be very different it won't really focus too much on the geometric interpretation of the line

Y will be a vector as a list



to represent our function
Just this x beta we know our prediction

Solve this question is for what beta should be

Multiply the matrices essentially
And the outer list represents the two columns so it's like diagnostic

com·pact

adjective: **compact**

1. closely and neatly packed together; dense.
   - having all the necessary components or features neatly fitted into a small space.
     synonyms: small, little, petite, miniature, mini, small-scale, neat, economic of space, fun-size, wee, teeny, teeny-weeny, teensy-weensy, dinky, little-bitty
   - (of a person or animal) small, solid, and well-proportioned.
     synonyms: dense, packed close, close-packed, tightly packed, pressed together, thick, tight, firm, solid
   - (of speech or writing) concise in expression.

Solve this question is for what beta should be

Multiply the matrices essentially

And the outer list represents the two columns so it's like diagnostic and midterm score then each list within that outer list represents all of the roads, so diagnostics for would be one entire column and then midterm score would be one entire column And when you wrap another list around that you can think about it like having two different dimensions that you can go over but again, I wouldn't worry about that too much. It's just a nice compact way to write it

A nice property in linear algebra is that we know that two vectors are going to be orthogonal or perpendicular form a right angle when their dot product is zero

"dot product"

$$[-1\ \ 2] \cdot [4, 5] = -1 \times 4 + 5 \times 2 = 6$$

Underneath it means the dot product is always zero

Main goal here that we talked about previously is to try and be as close the actual value as we can while maintaining kind of while being in our plane of predicted values

x orthogonal to $y - \hat{y}$

$$X^T(Y - X\beta) = 0$$

$$X^T Y - X^T X \beta = 0$$

$$\beta = (X^T X)^{-1} X^T y$$

And also since

$$f = X\beta$$

We can get f

But since there's noises and errors so the prediction value data won't be perfect and so there will actually be if only a little there will be a little difference between your predicted value and then the actual value

The noise will be in the third dimension

The key is to find the actual value and get close to it as possible
In order to do that we have to be orthogonal to it

$$f(x) = w \cdot x + b$$

Linear equation

Take something type of linear equation make it non-linear we can always just add a non-linear function in front of it

$$f(x) = (w \cdot x + b)^2$$

log

Ask some way to represent from x to y

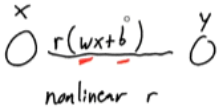
x — wx+b — y

proportioned.
synonyms: dense, packed close, close-packed, tightly packed, pressed together, thick, tight, firm, solid
- (of speech or writing) concise in expression.

or·thog·o·nal /ôr THägən(ə)l/

adjective: **orthogonal**

1. of or involving right angles; at right angles.
2. (of variates) statistically independent.
   - (of an experiment) having variates which can be treated as statistically independent.

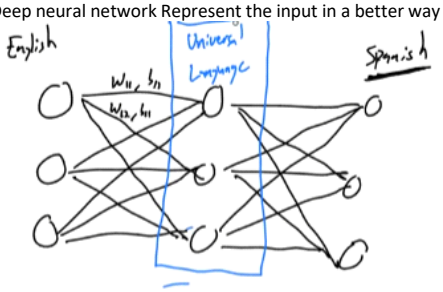


cal·cu·lus /ˈkalkyələs/

noun: **calculus**; noun: **infinitesimal calculus**; plural noun: **calculuses**; plural noun: **calculi**

1. the branch of mathematics that deals with the finding and properties of derivatives and integrals of functions, by methods originally based on the summation of infinitesimal differences. The two main types are differential calculus and integral calculus.
2. a particular method or system of calculation or reasoning.
3. a concretion of minerals formed within the body, especially in the kidney or gallbladder.
   - another term for tartar.

Deep neural network Represent the input in a better way



What neural networks are good at is finding ways to represent data

is that it's very hard for a human to come up with this metal man ourselves in order to input that into like regression. For example and so when the computer can come up with itself. It is much much better at doing that. Basically So someone asked some machine learning top-level research is mostly about math but not CS. There's a lot of different answers that I would argue largely. Yes, However, there's like the rise lab at Berkeley is the one I can think of where they work on kind of distributed machine

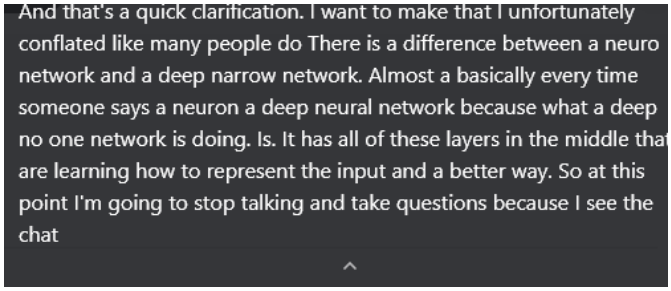
Statistical learning

So the real power just to reiterate this again of neural networks. Is that they're finding ways to represent the input in a way that makes sense to computers And the way that they're doing that is something called back complication which we talked about is basically the chain rule on steroids.

Back propagation

steroids. The power that you're getting with a deep neural network is the fact that for regression and for something called regression trees or decision trees are in general. There's a class of machine learning algorithms called trees. You have to figure out as a human what the input should be to the model right? I have to determine I'm trying to regress on all of these inputs over here. The issue is that humans are very bad at doing this. We're very bad at coming up with ways that the computer should be able to

input should be to the model right? I have to determine I'm trying to regress on all of these inputs over here. The issue is that humans are very bad at doing this. We're very bad at coming up with ways that the computer should be able to understand for example, English or an image or anything And so the power that you get with a deep neural network is the fact that it learns to represent the input in a better way And that's a quick clarification. I want to make that I unfortunately conflated like many people do There is a difference between a neur


And that's a quick clarification. I want to make that I unfortunately
conflated like many people do There is a difference between a neuro
network and a deep narrow network. Almost a basically every time
someone says a neuron a deep neural network because what a deep
no one network is doing. Is. It has all of these layers in the middle that
are learning how to represent the input and a better way. So at this
point I'm going to stop talking and take questions because I see the
chat