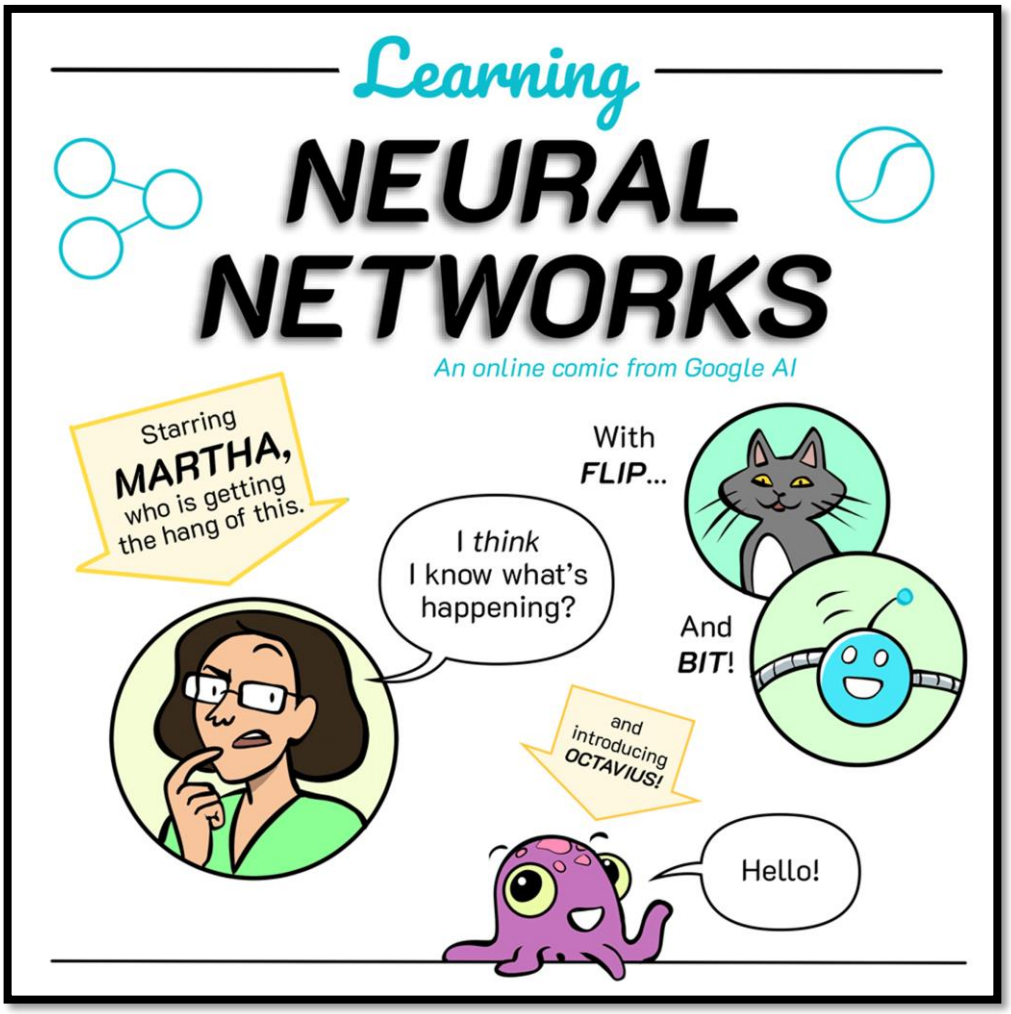
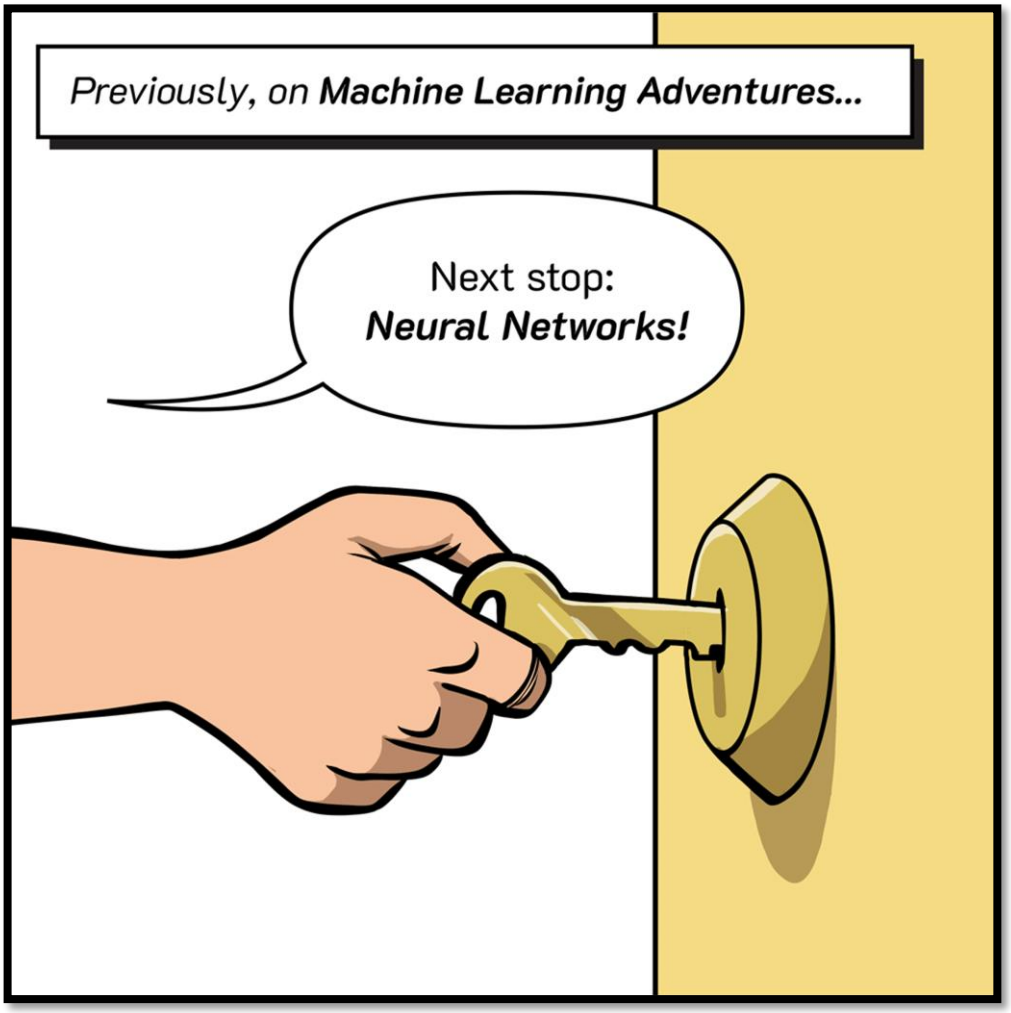
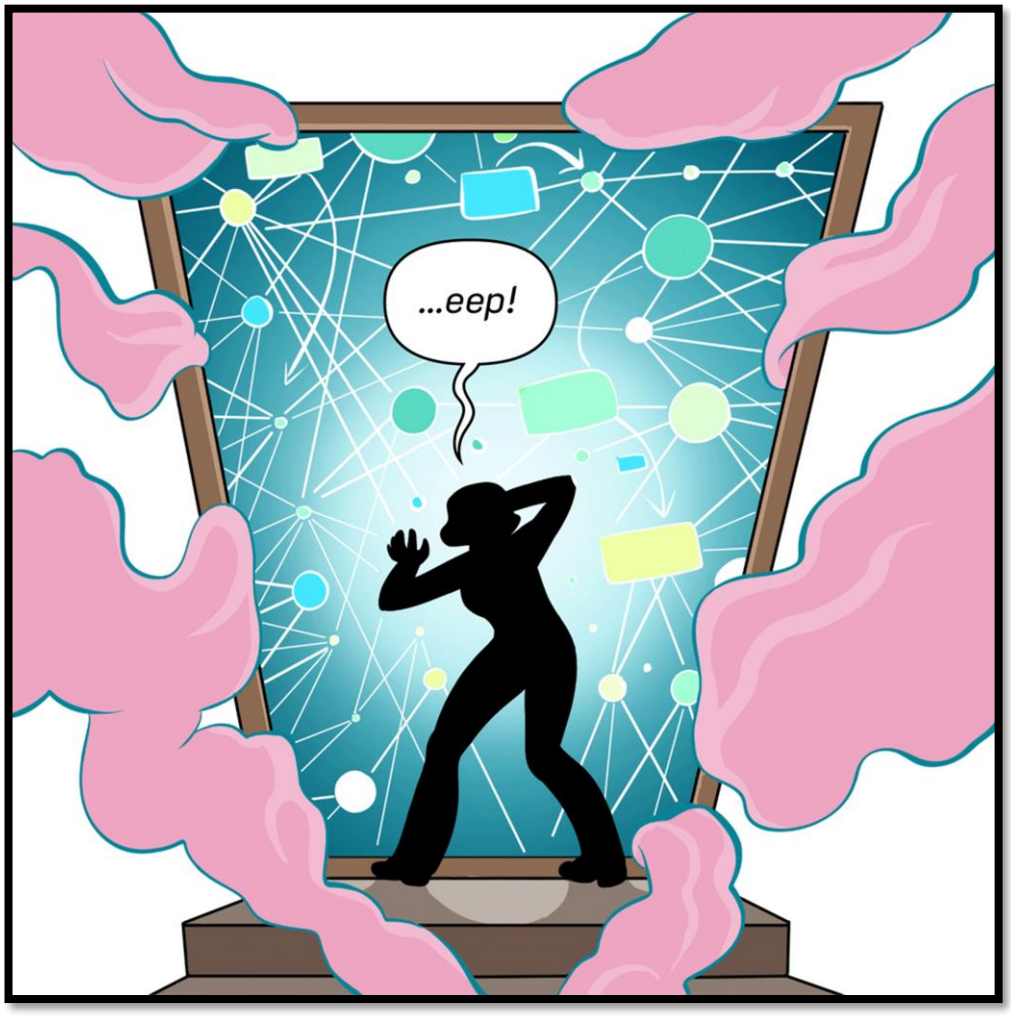
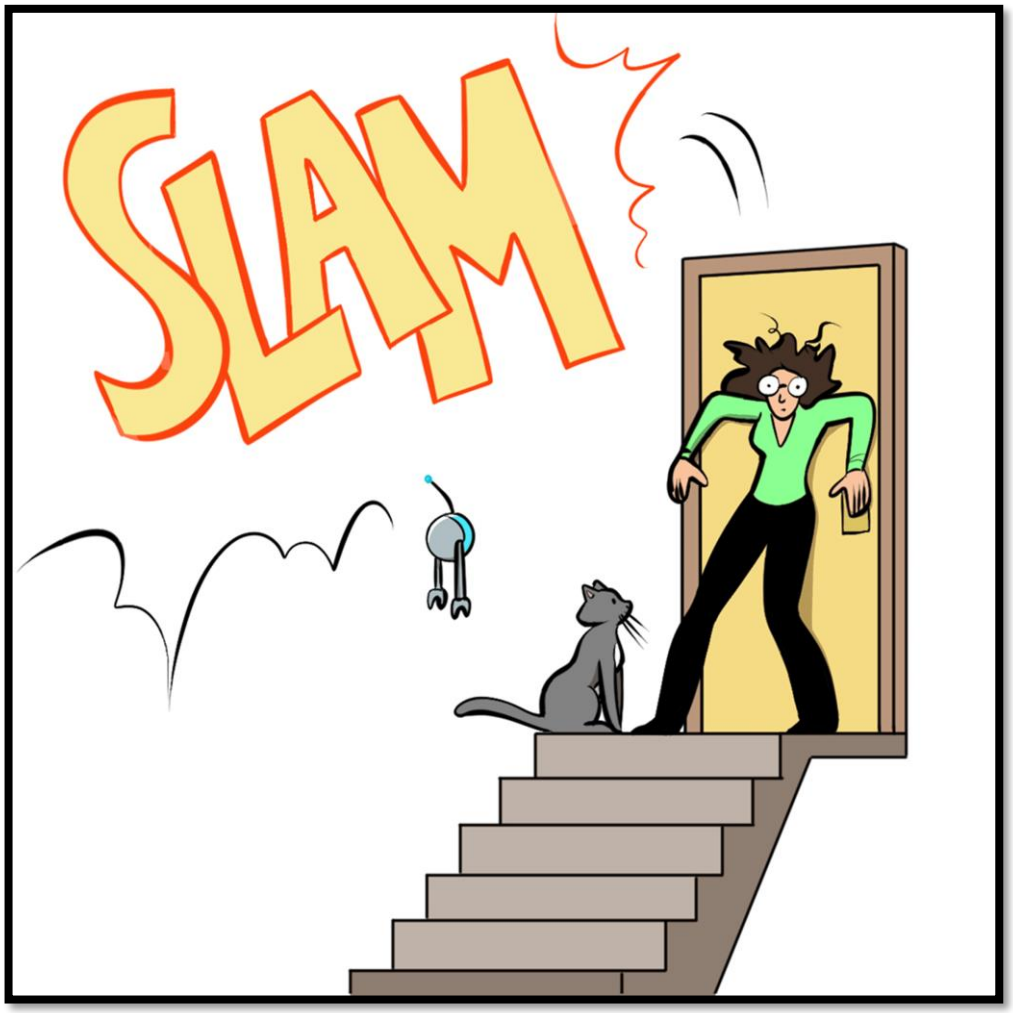

Learning
NEURAL NETWORKS
An online comic from Google AI
Starring MARTHA, who is getting the hang of this.
With FLIP...
And BIT!
I think I know what's happening?
and introducing OCTAVIUS!
Hello!

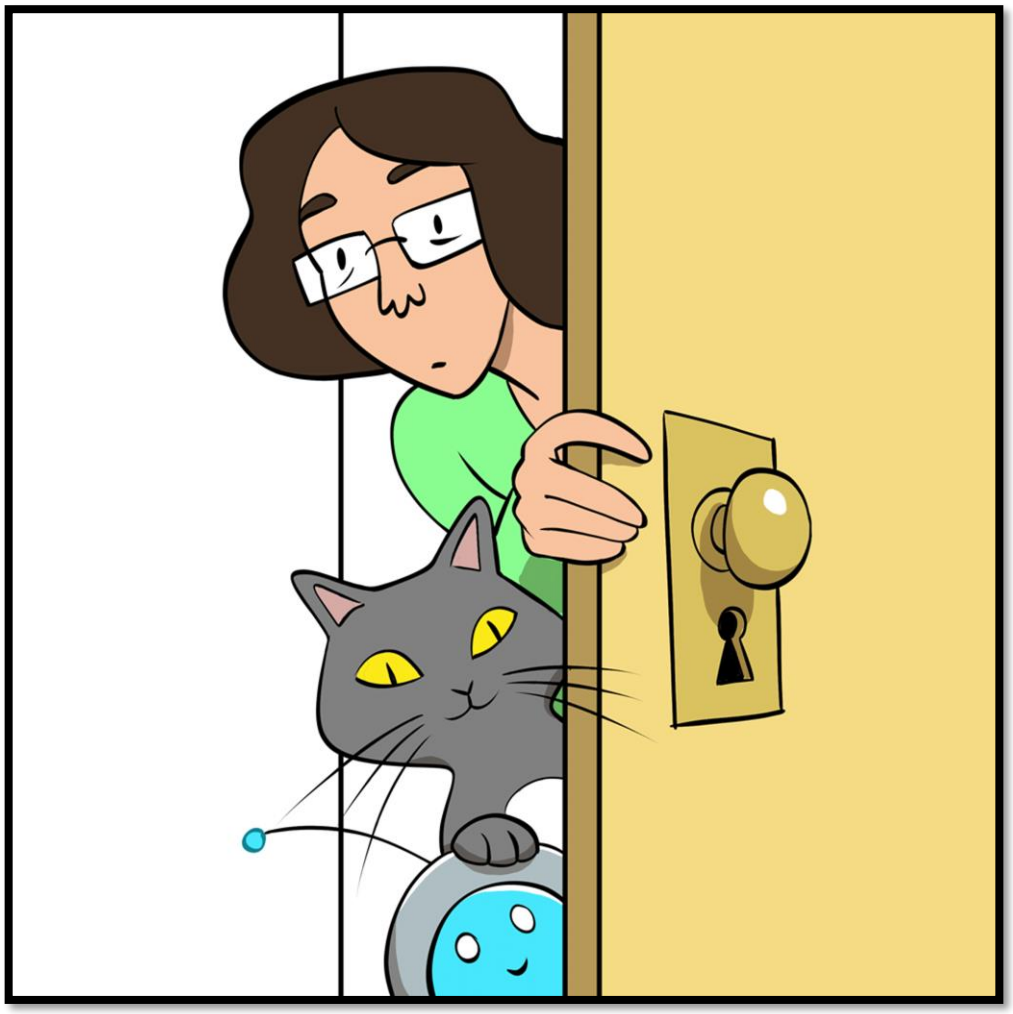

Previously, on Machine Learning Adventures...
Next stop: Neural Networks!

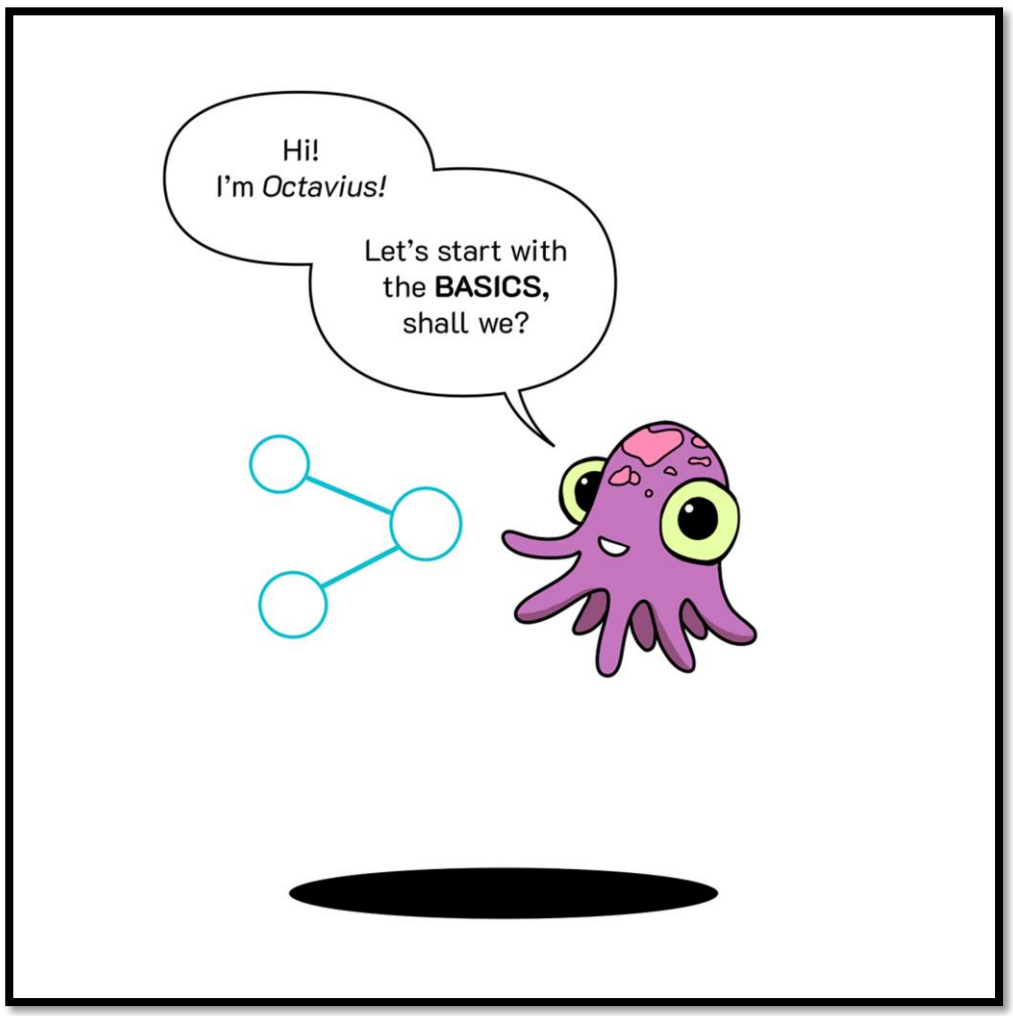

HELLO WORL—

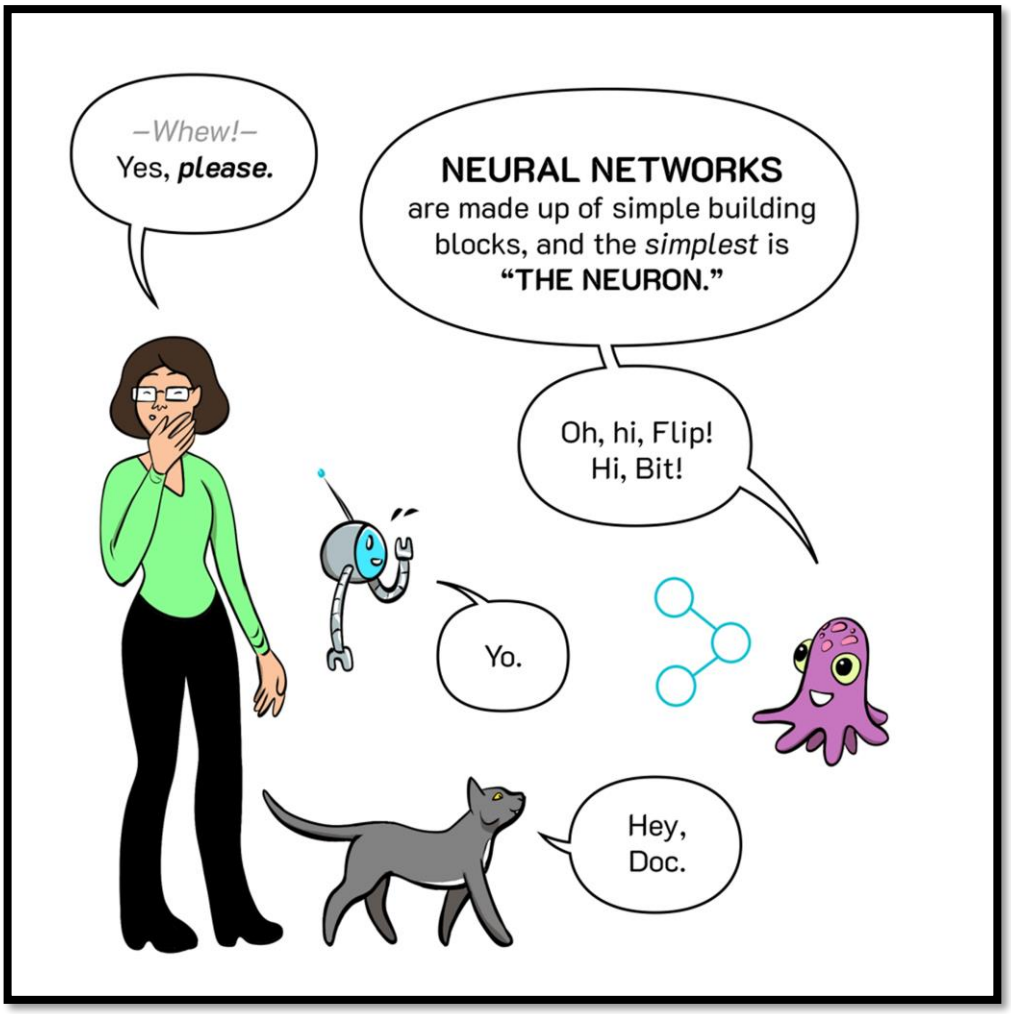

...eep!

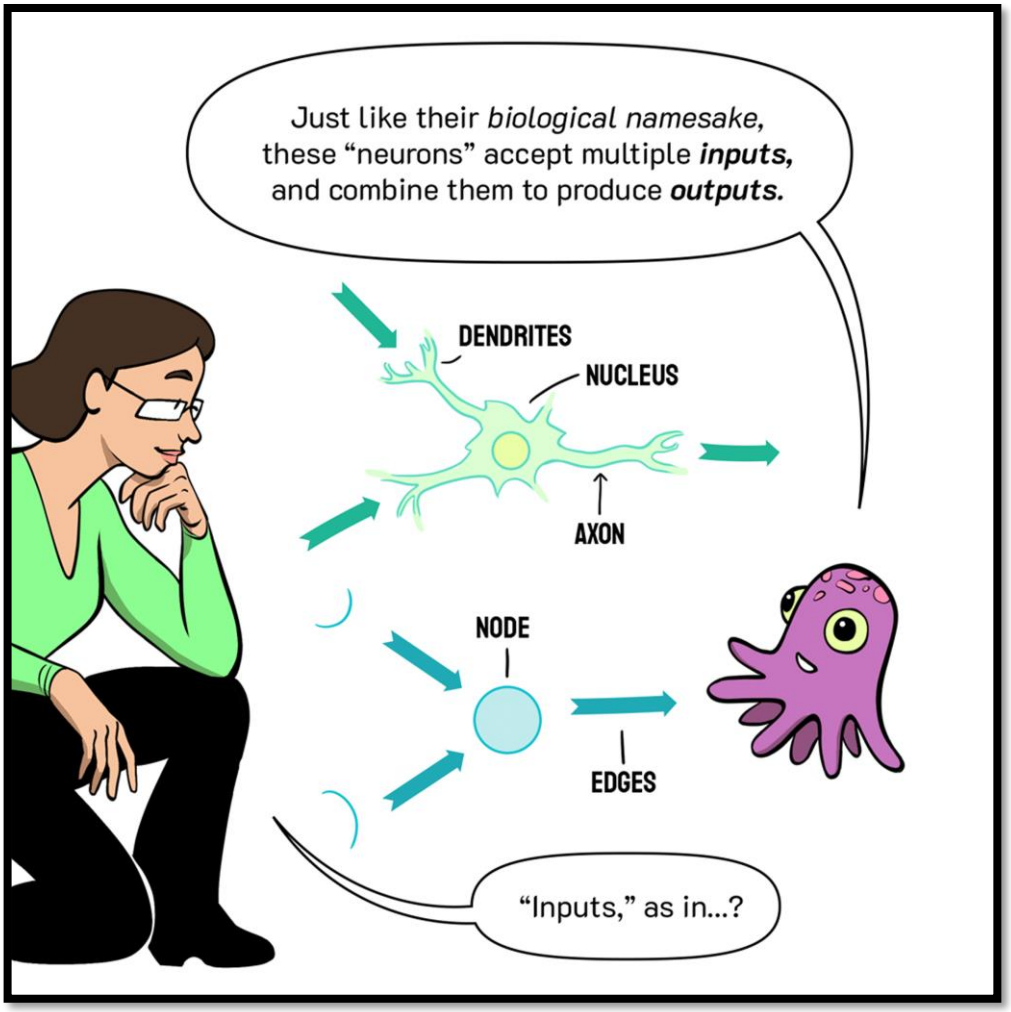

SLAM

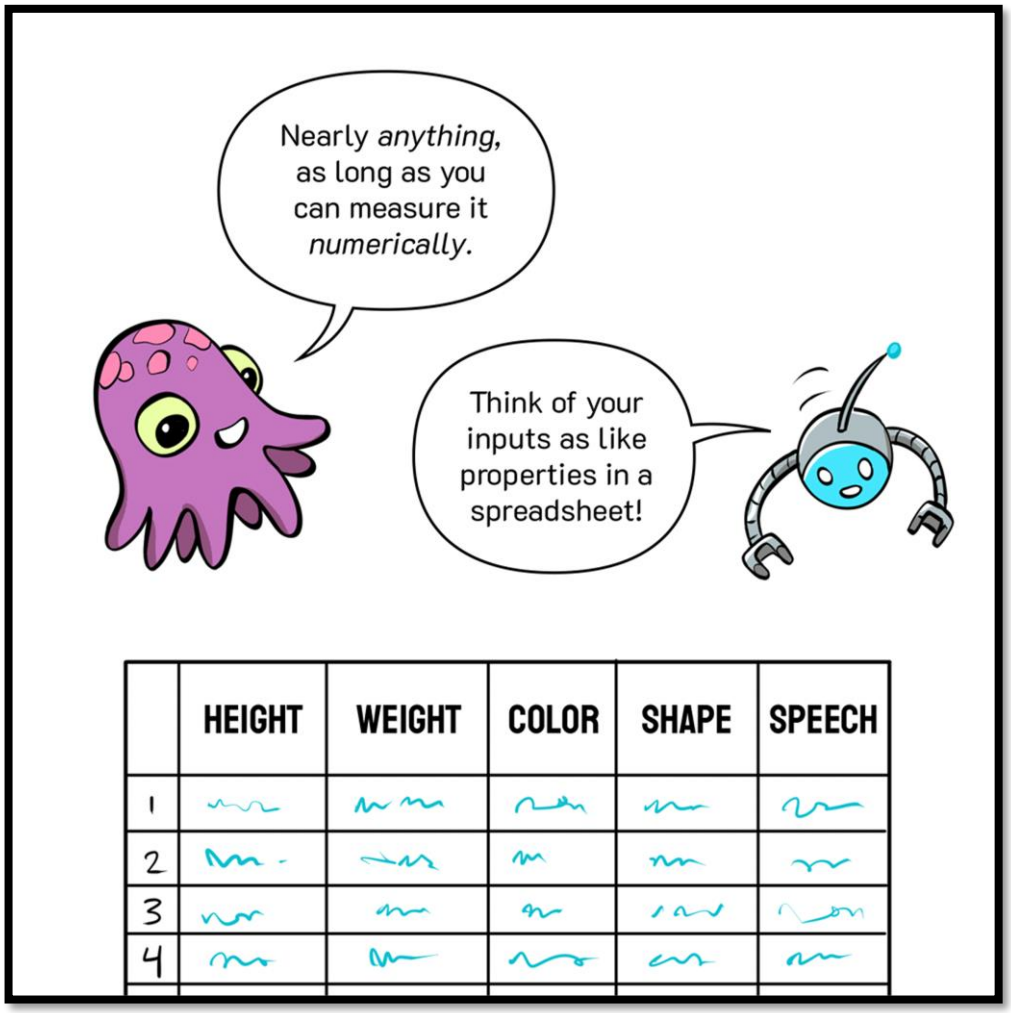

Sorry, *sorry!*
Wasn't ready yet.
Okay,
come in.

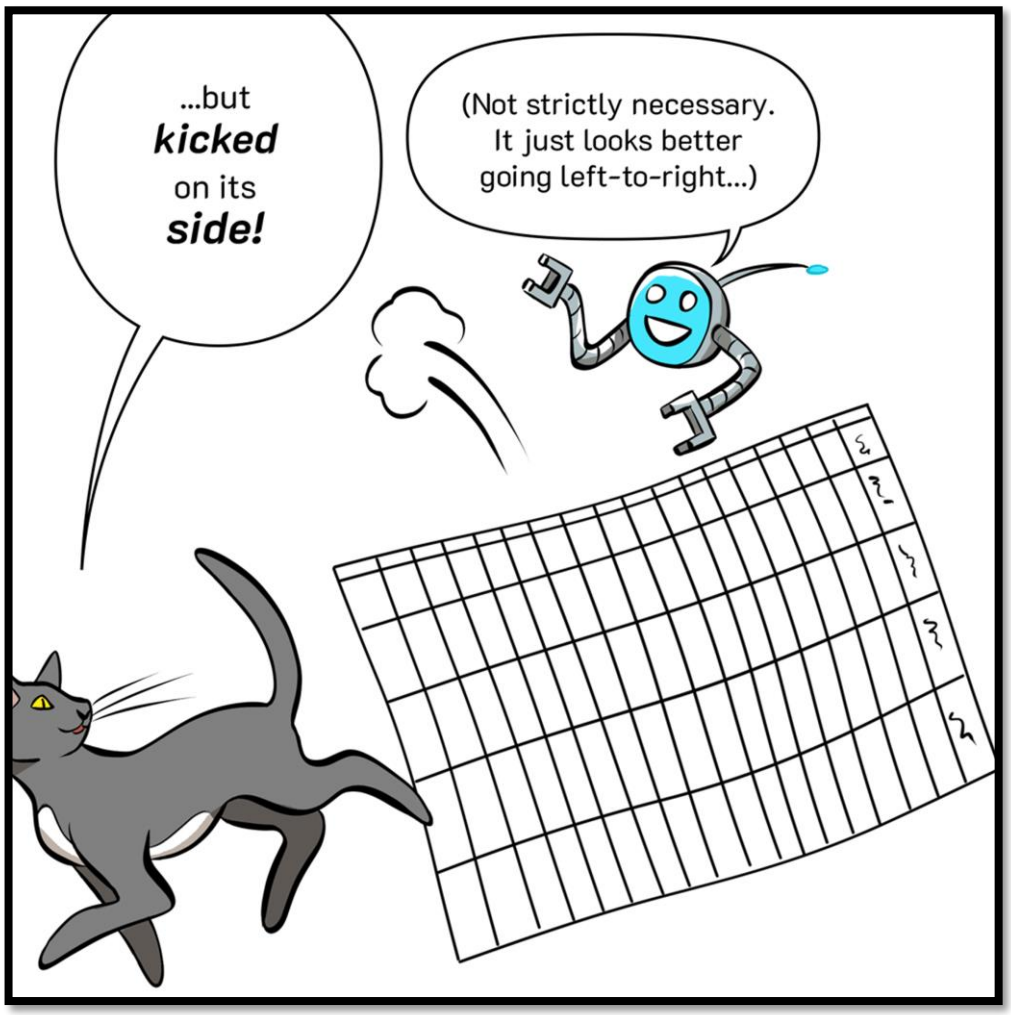


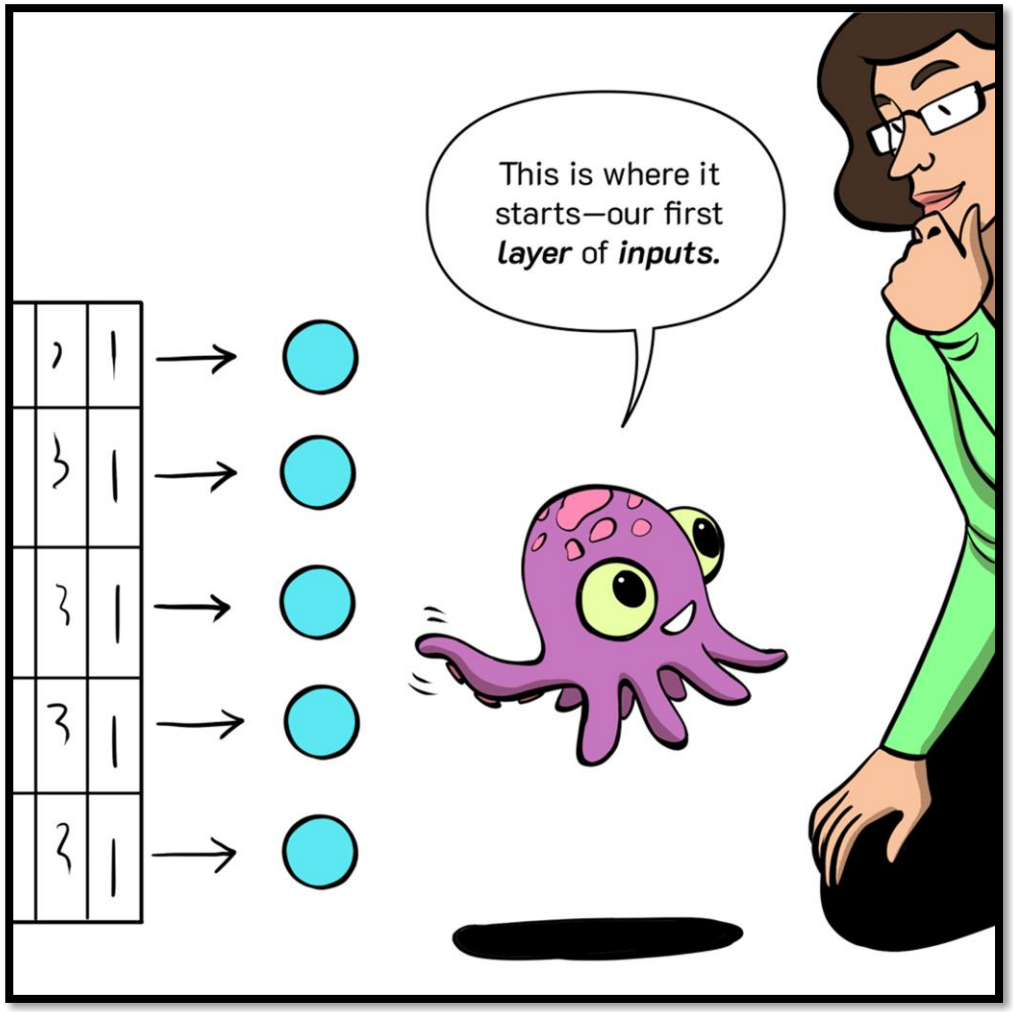

Hi!
I'm *Octavius!*
Let's start with the **BASICS,** shall we?

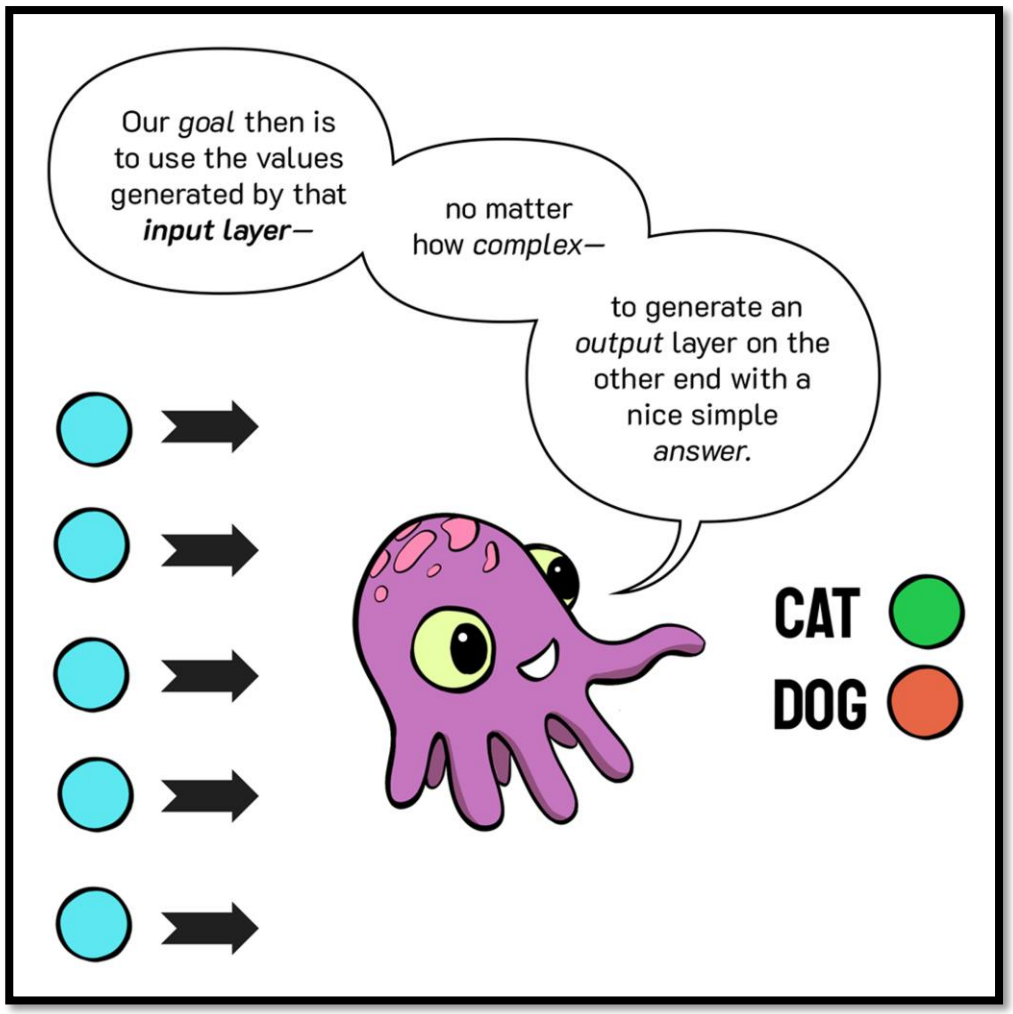

–Whew!–
Yes, please.
NEURAL NETWORKS
are made up of simple building blocks, and the simplest is "THE NEURON."
Oh, hi, Flip! Hi, Bit!
Yo.
Hey, Doc.

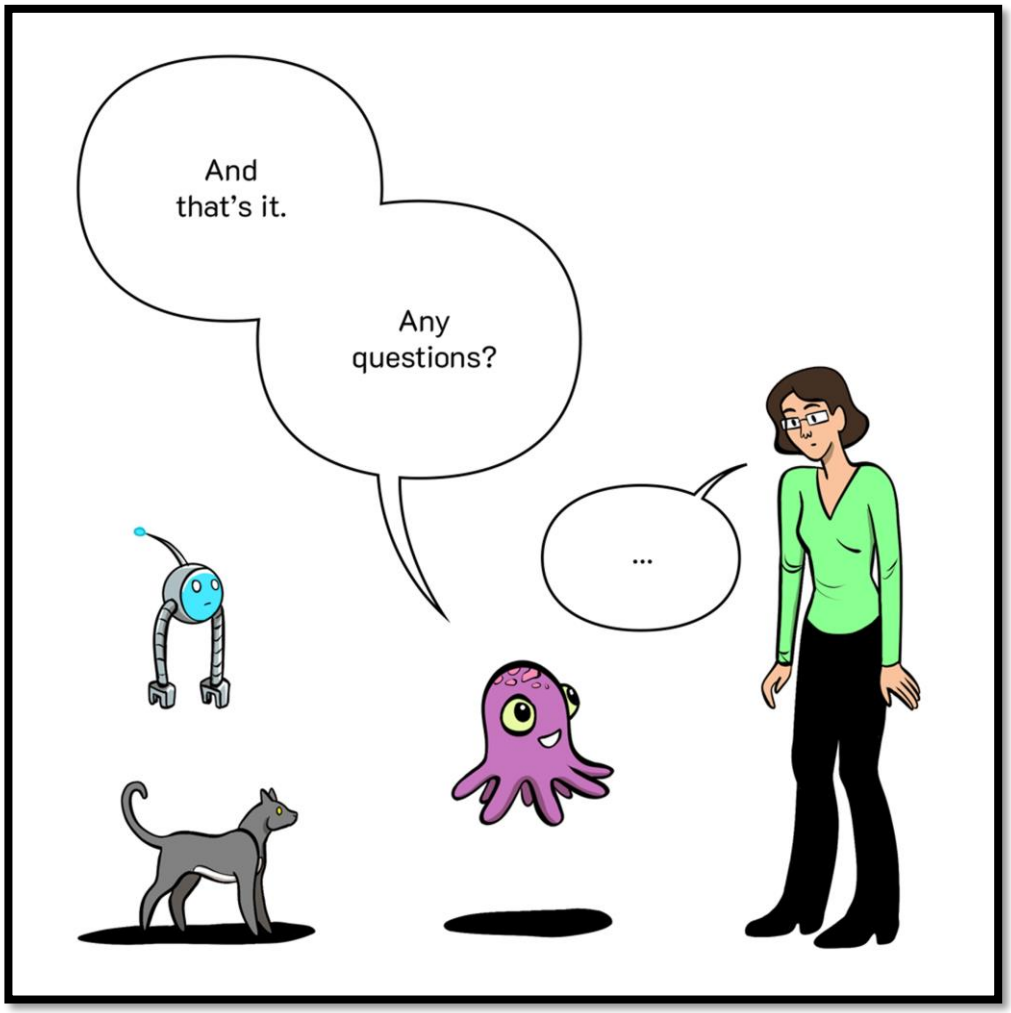

Just like their biological namesake, these "neurons" accept multiple inputs, and combine them to produce outputs.
DENDRITES
NUCLEUS
AXON
NODE
EDGES
"Inputs," as in...?

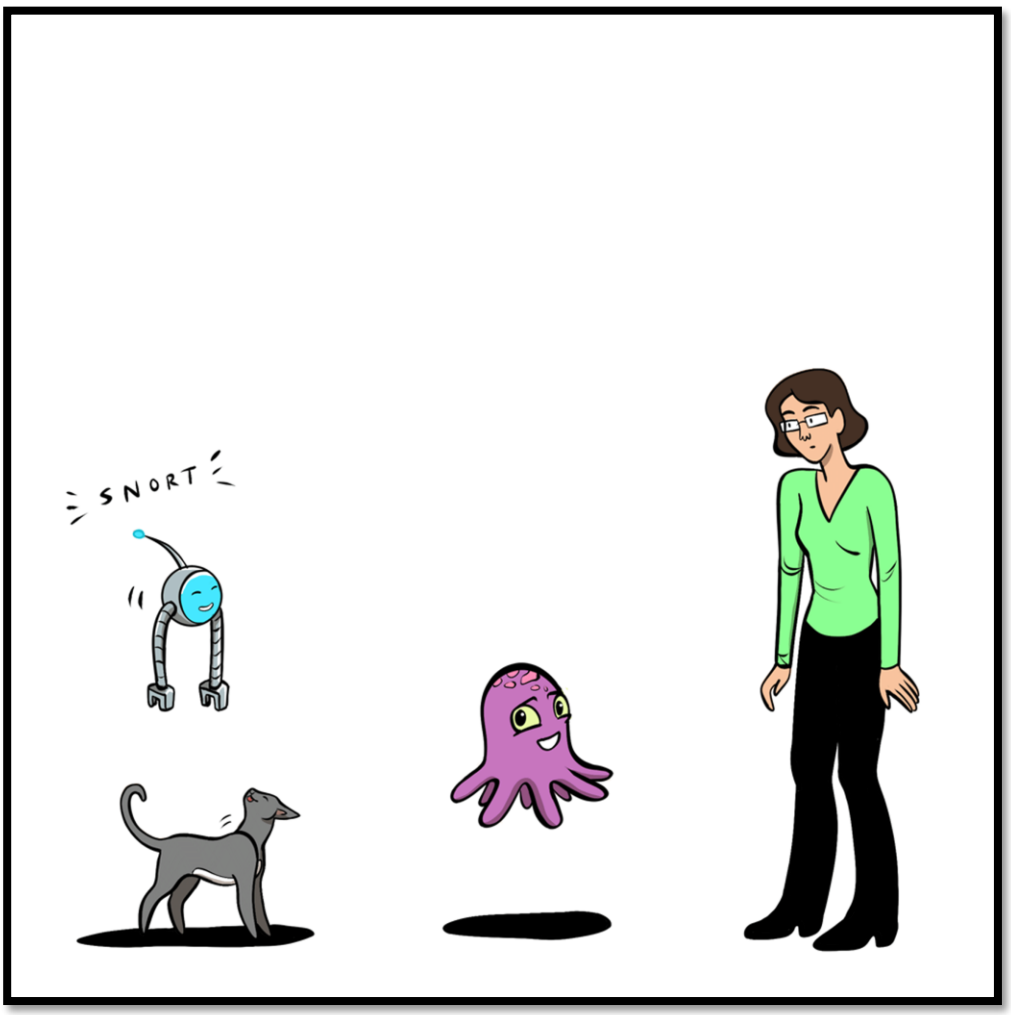

Nearly *anything*, as long as you can measure it *numerically*.
Think of your inputs as like properties in a spreadsheet!
HEIGHT
WEIGHT
COLOR
SHAPE
SPEECH
1
2
3
4


...but ***kicked*** on its ***side!***
(Not strictly necessary. It just looks better going left-to-right...)

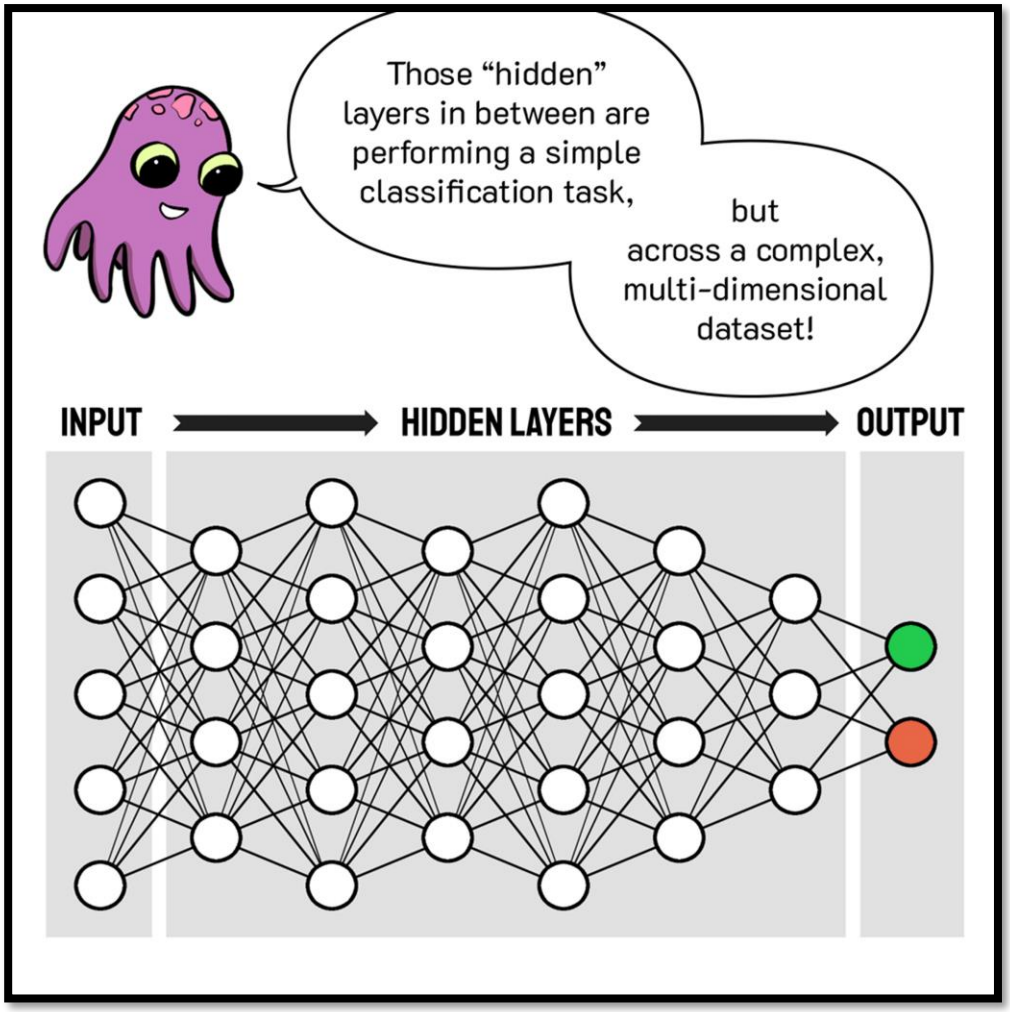

This is where it starts—our first *layer* of *inputs.*

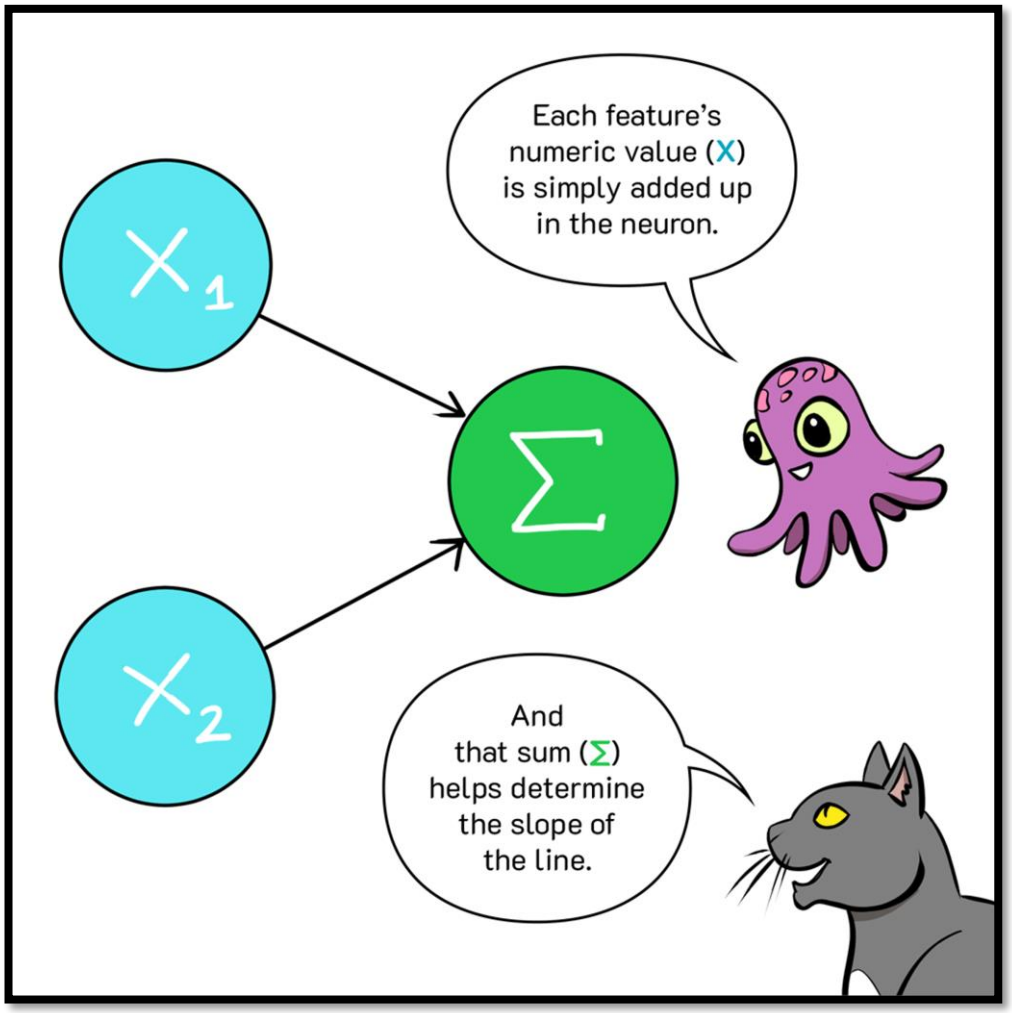

Our *goal* then is to use the values generated by that *input layer*—
no matter how *complex*—
to generate an *output* layer on the other end with a nice simple *answer.*
CAT
DOG

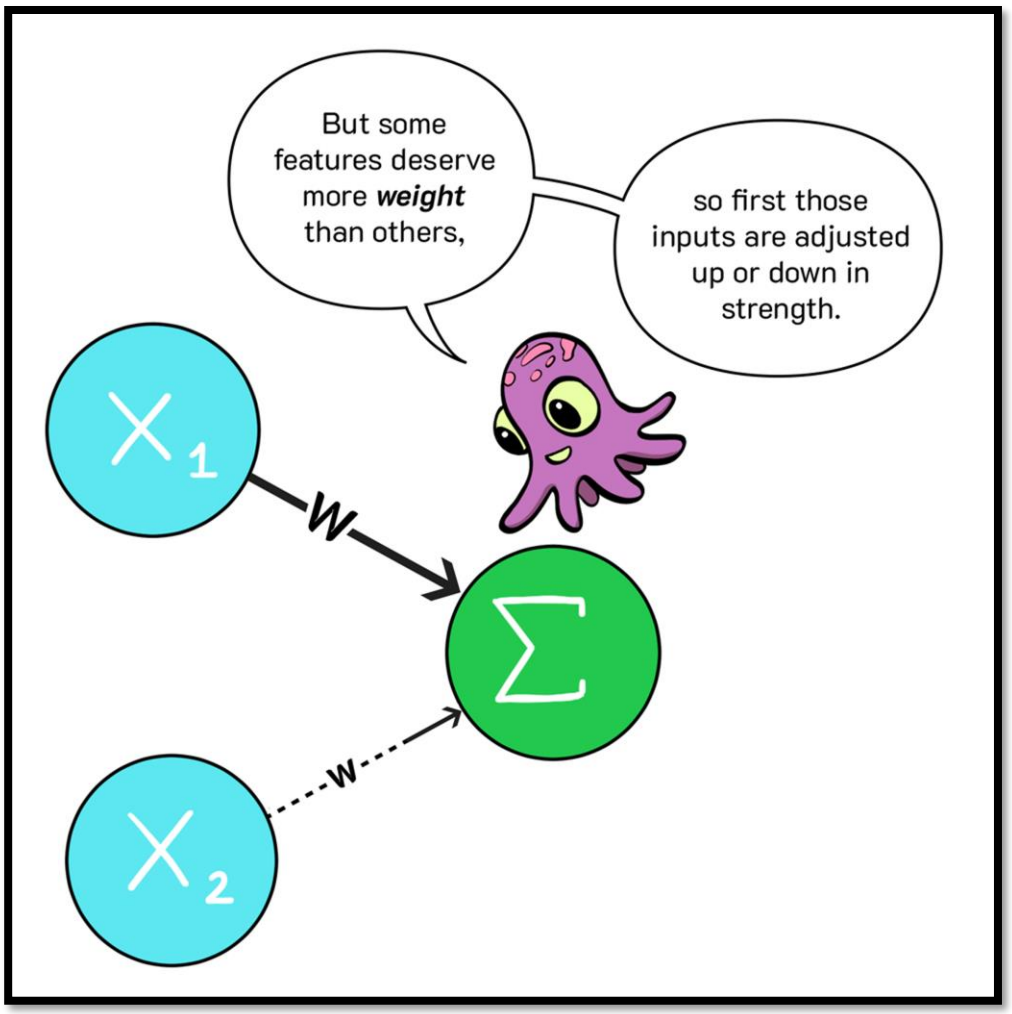

And
that's it.
Any
questions?
...

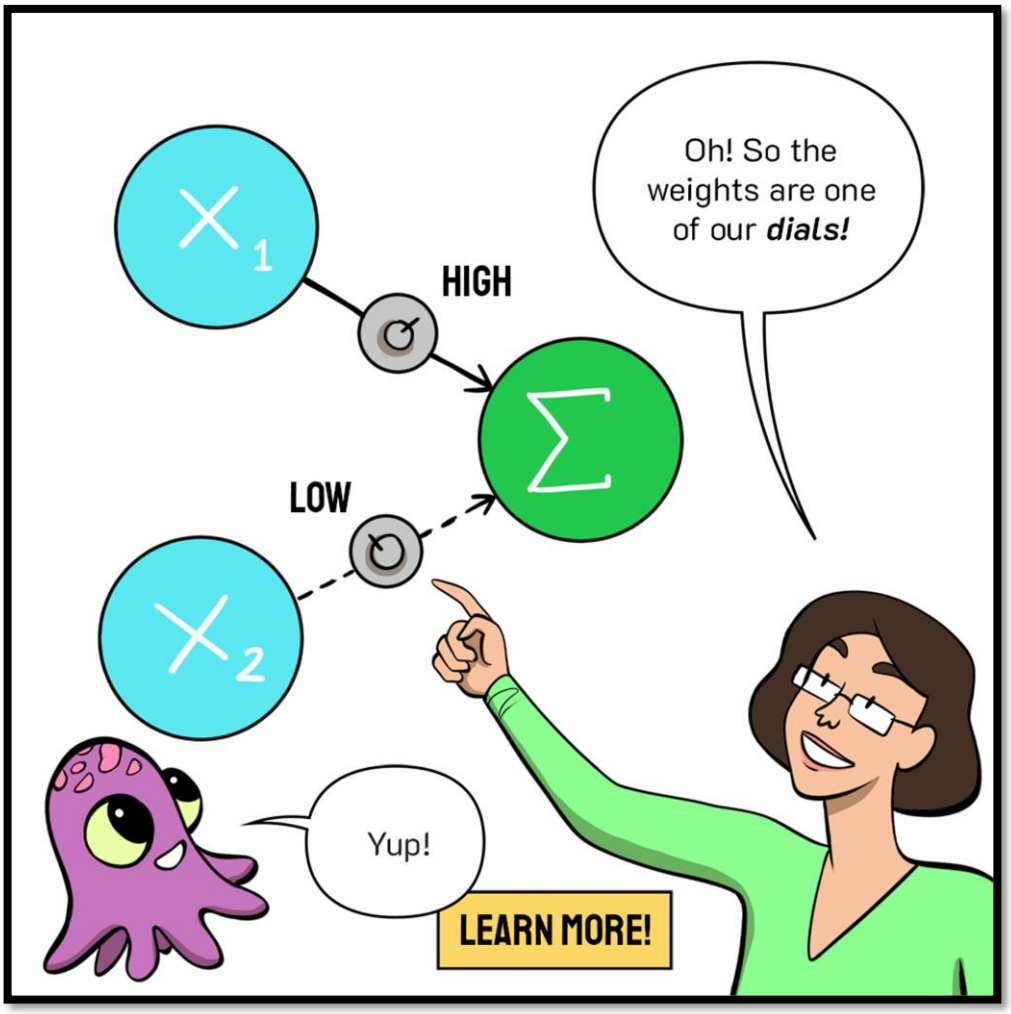

SNORT

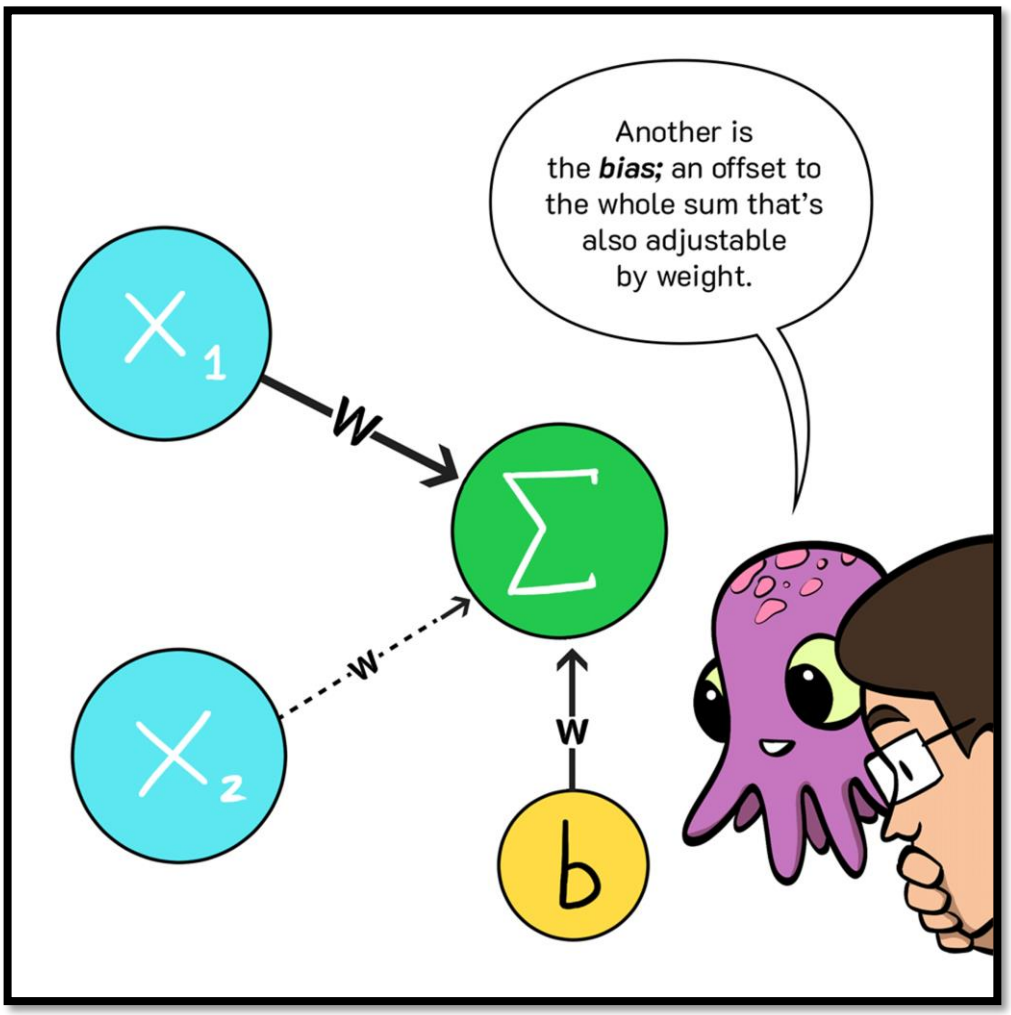

But–
WHAT HAPPENS IN BETWEEN?!
Exactly!

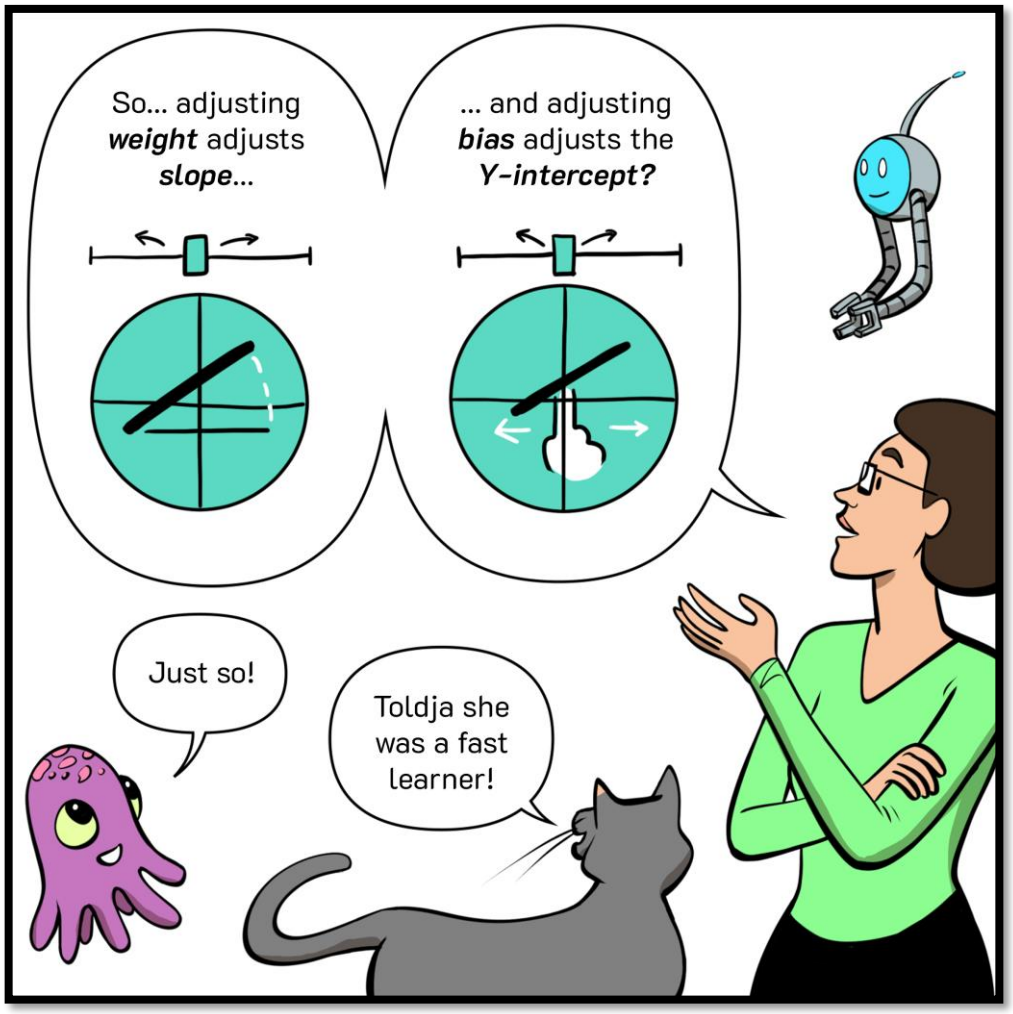

Those “hidden” layers in between are performing a simple classification task,
but across a complex, multi-dimensional dataset!
INPUT
HIDDEN LAYERS
OUTPUT

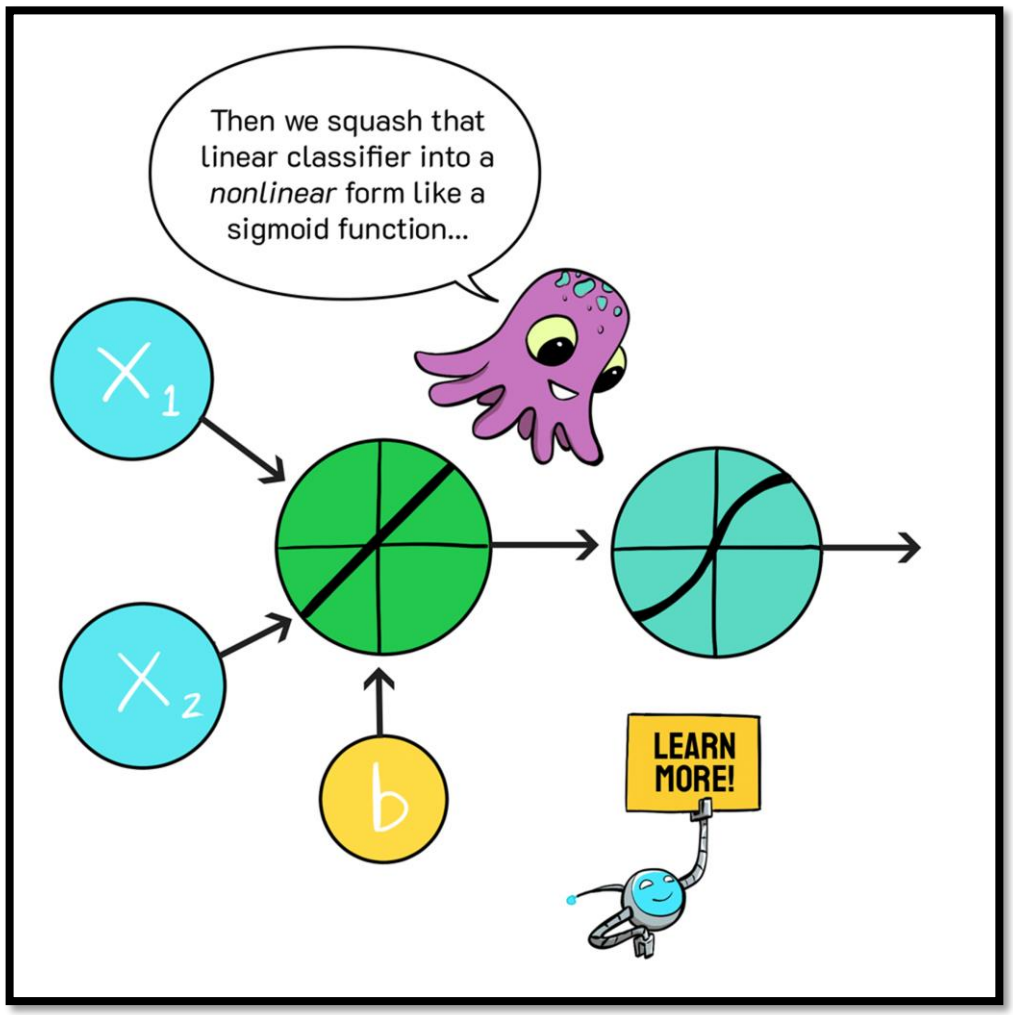

Each feature's numeric value (X) is simply added up in the neuron.
X 1
X 2
Σ
And that sum (Σ) helps determine the slope of the line.

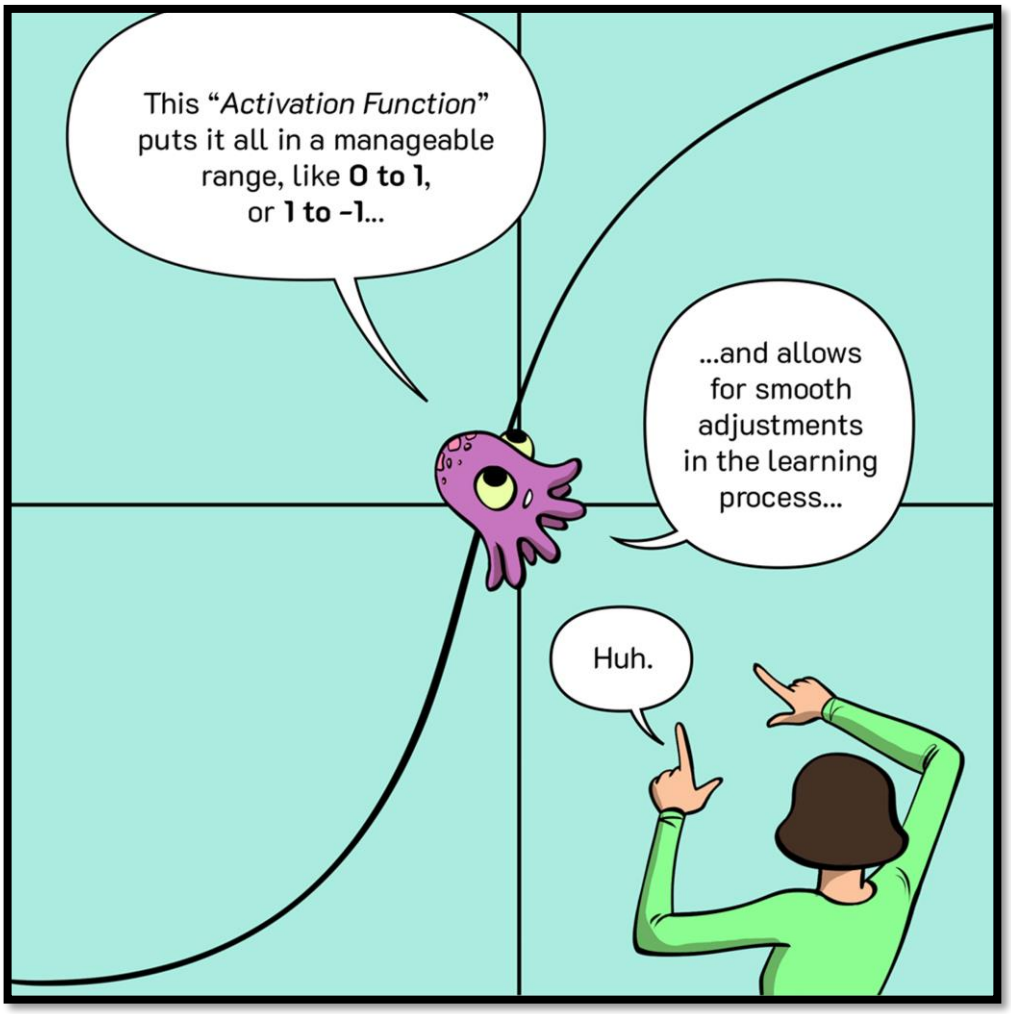

But some features deserve more *weight* than others,
so first those inputs are adjusted up or down in strength.
X 1
X 2
W
W
Σ

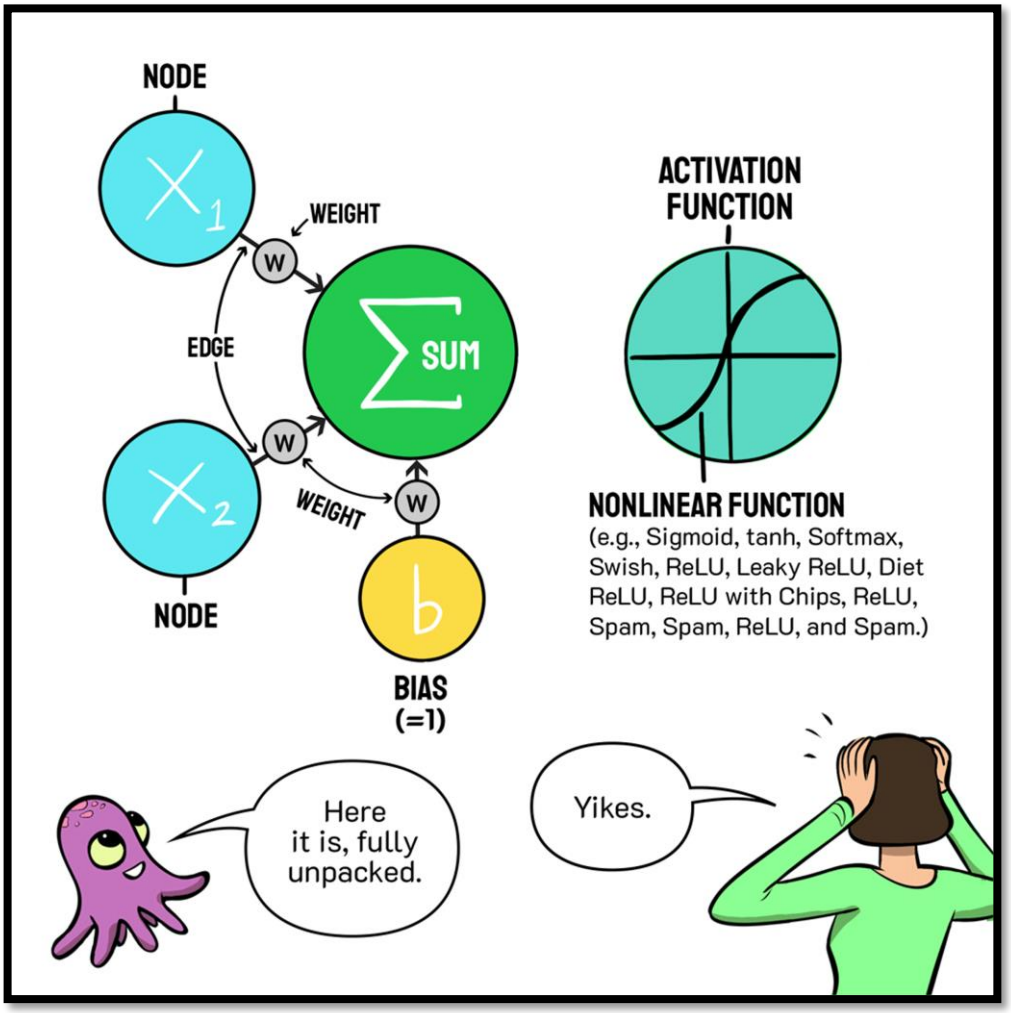

Oh! So the weights are one of our *dials!*
HIGH
LOW
Yup!
LEARN MORE!

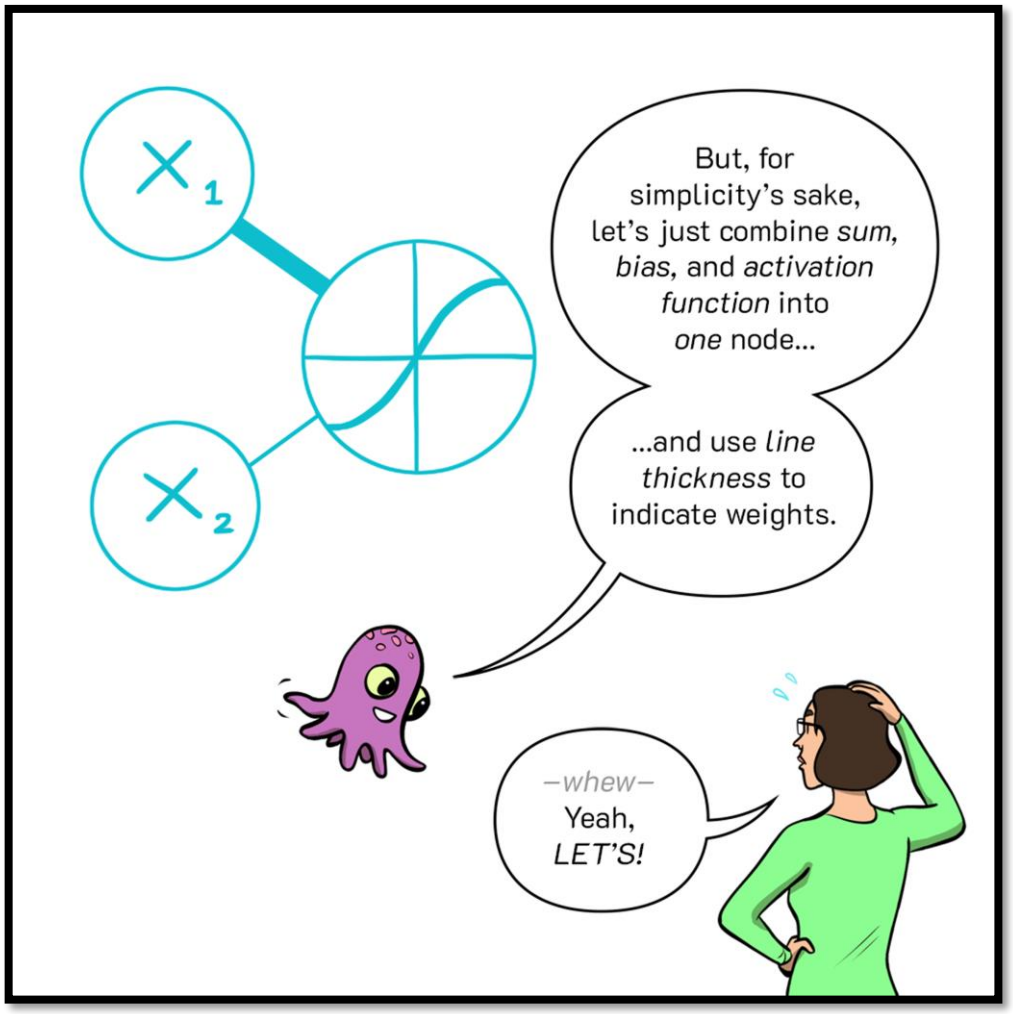

Another is the *bias;* an offset to the whole sum that's also adjustable by weight.
W
W
W
b

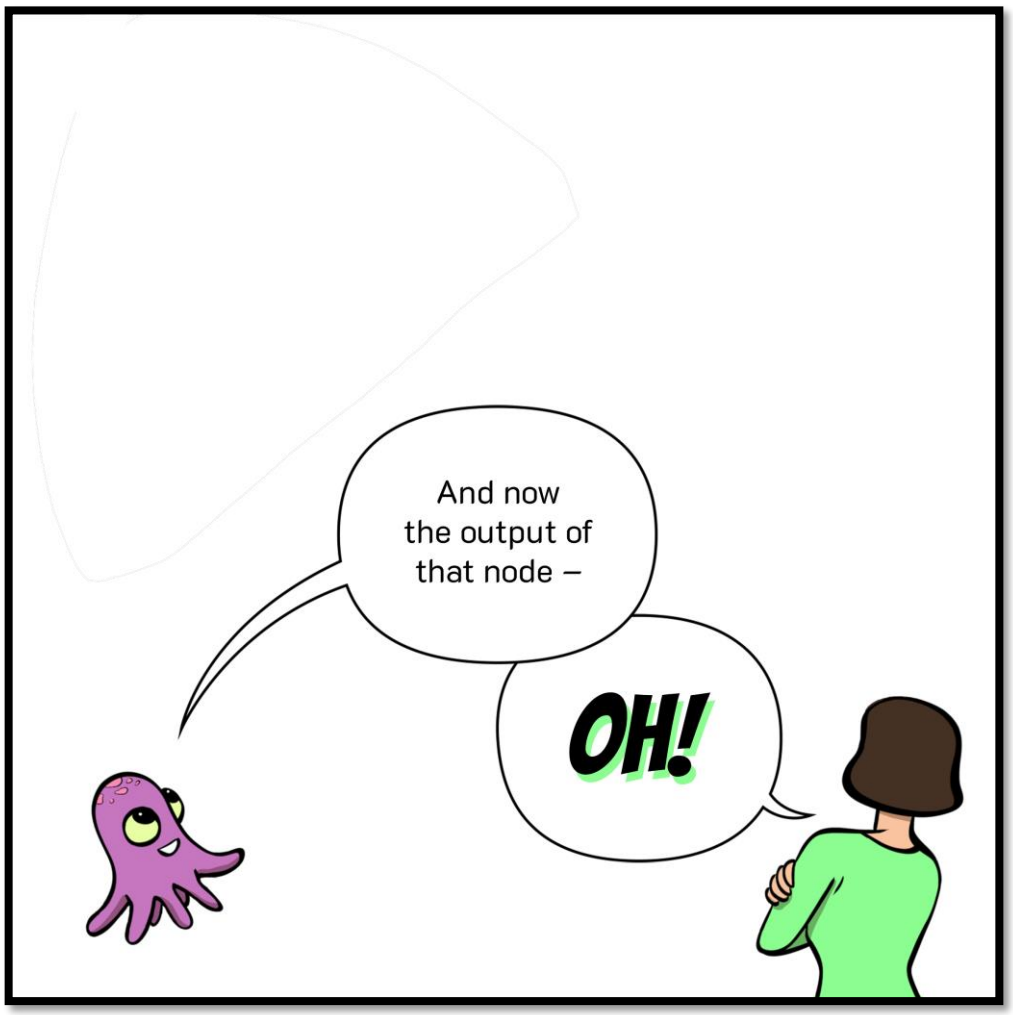

So... adjusting ***weight*** adjusts ***slope***...
... and adjusting ***bias*** adjusts the ***Y-intercept?***
Just so!
Toldja she was a fast learner!

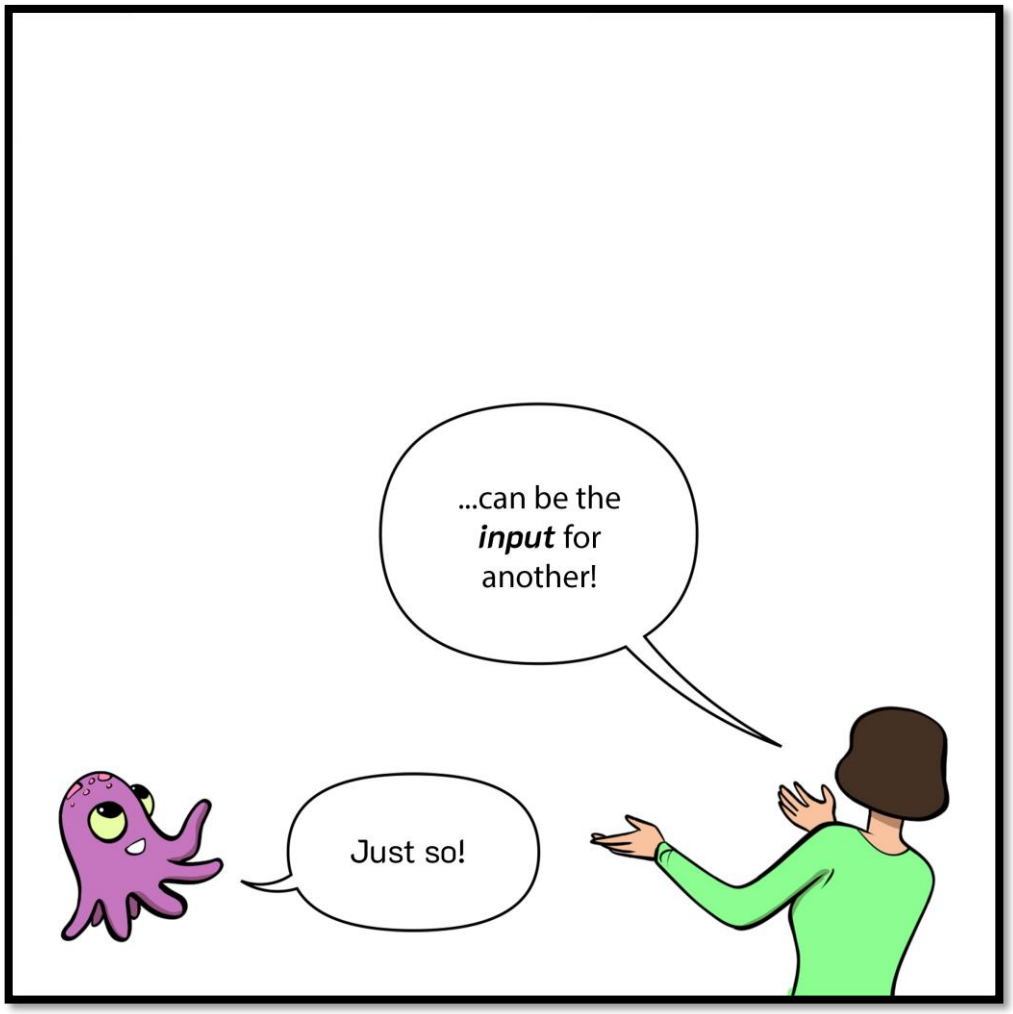

Then we squash that linear classifier into a *nonlinear* form like a sigmoid function...
X1
X2
b
LEARN MORE!

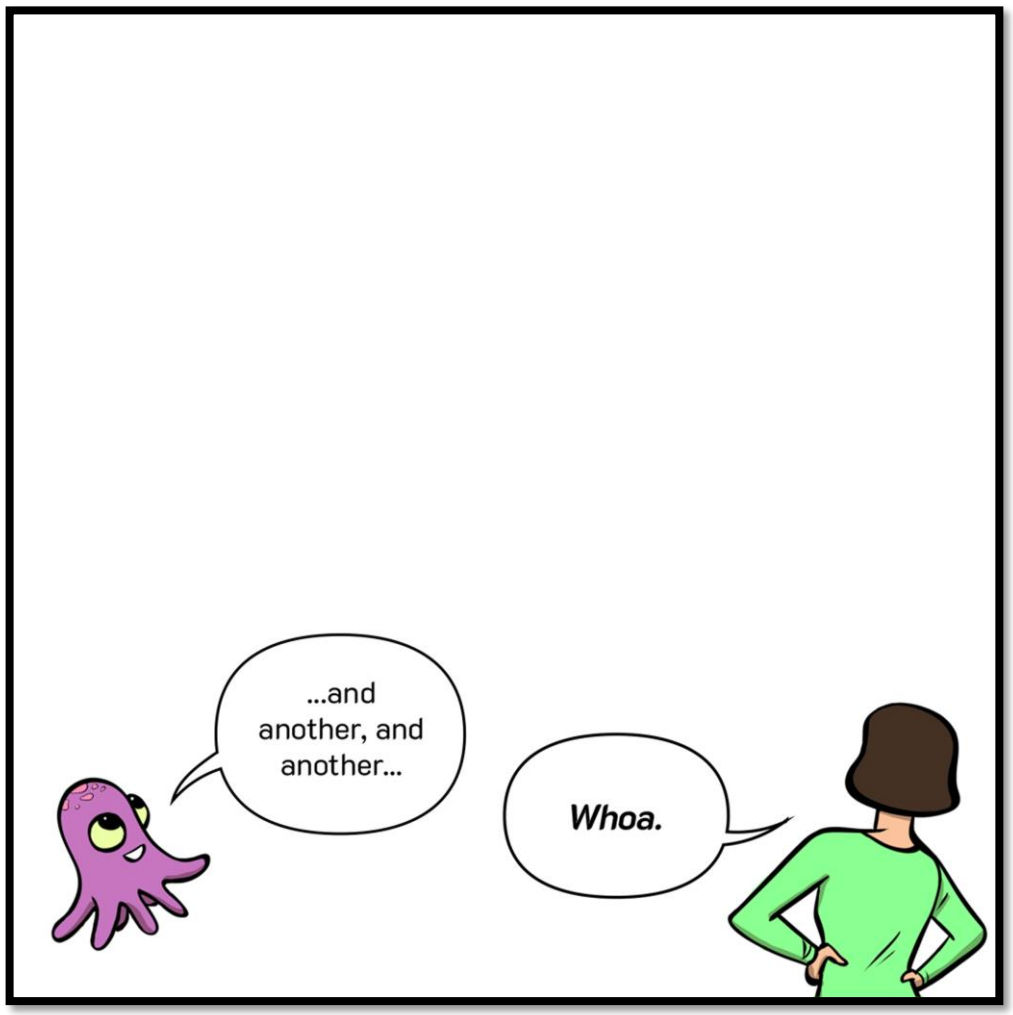

This "*Activation Function*" puts it all in a manageable range, like **0 to 1**, or **1 to -1**...
...and allows for smooth adjustments in the learning process...
Huh.

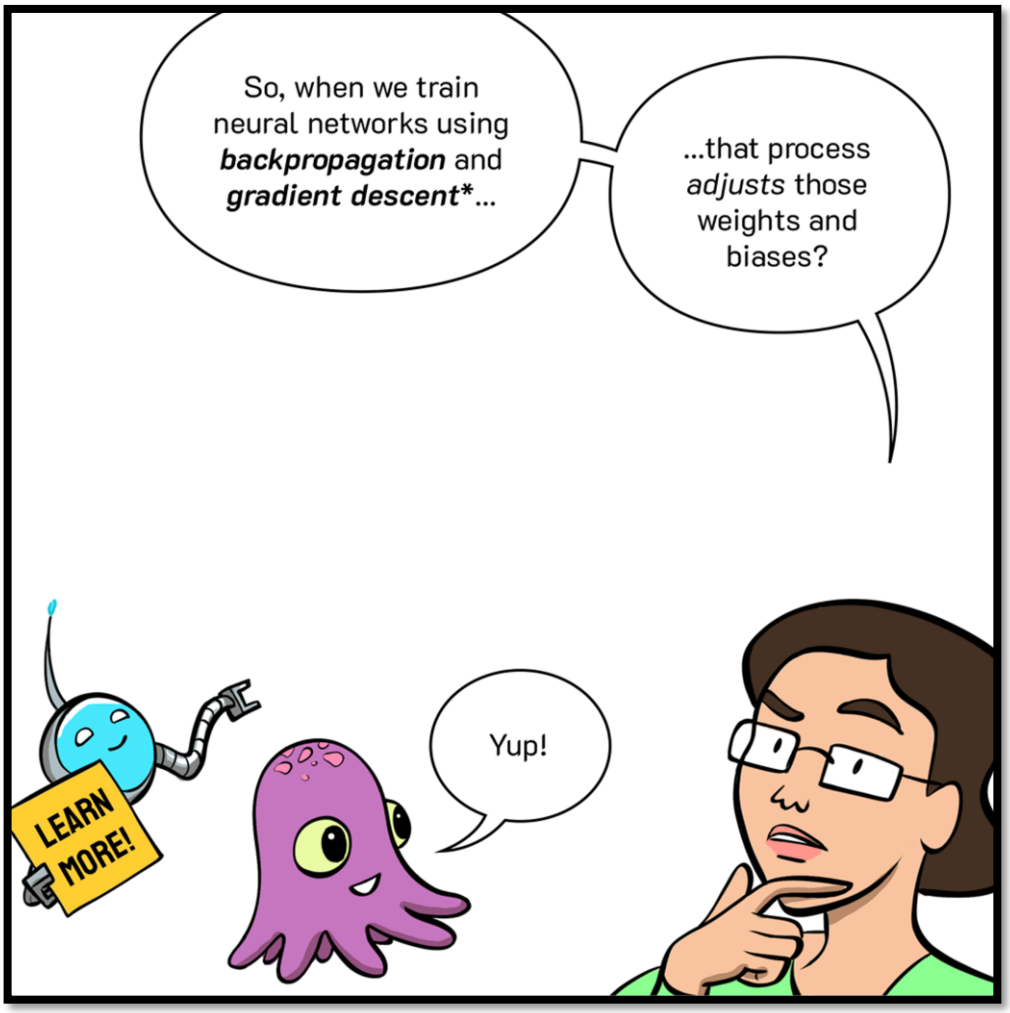

NODE
WEIGHT
W
EDGE
SUM
W
WEIGHT
W
NODE
BIAS
(=1)
ACTIVATION FUNCTION
NONLINEAR FUNCTION
(e.g., Sigmoid, tanh, Softmax, Swish, ReLU, Leaky ReLU, Diet ReLU, ReLU with Chips, ReLU, Spam, Spam, ReLU, and Spam.)
Here it is, fully unpacked.
Yikes.

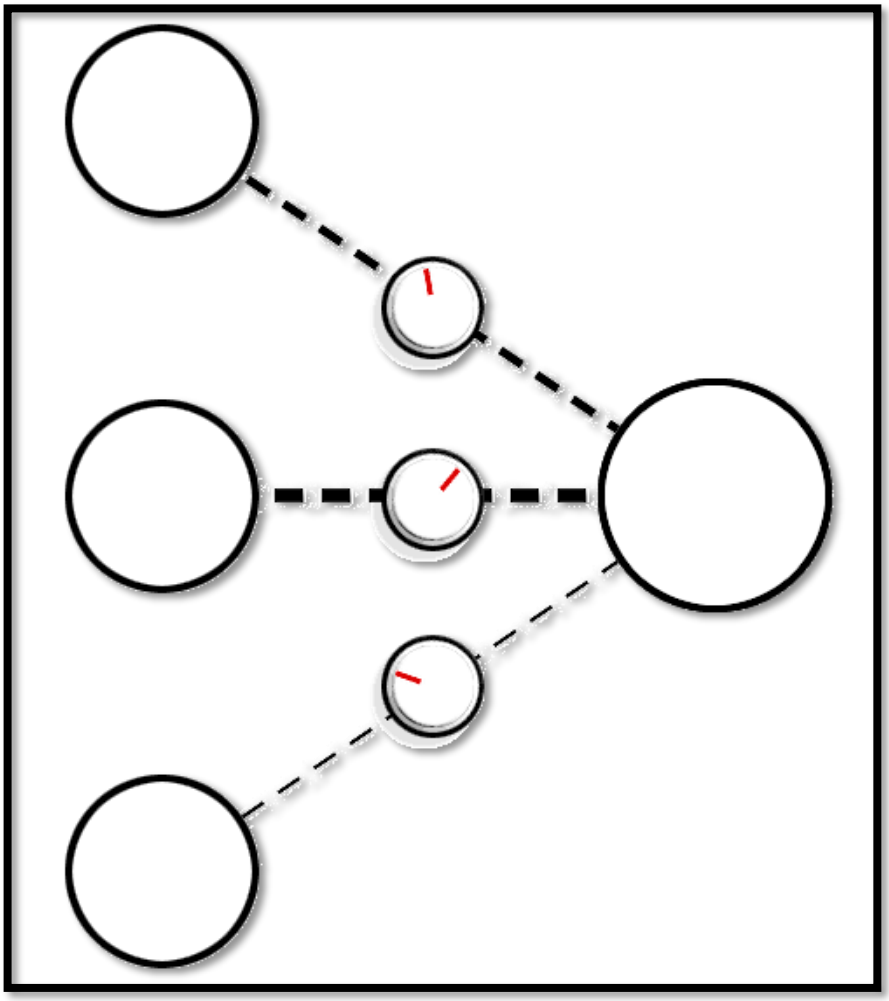

X 1
X 2
But, for simplicity's sake, let's just combine *sum*, *bias*, and *activation function* into *one* node...
...and use *line thickness* to indicate weights.
–whew–
Yeah, *LET'S!*

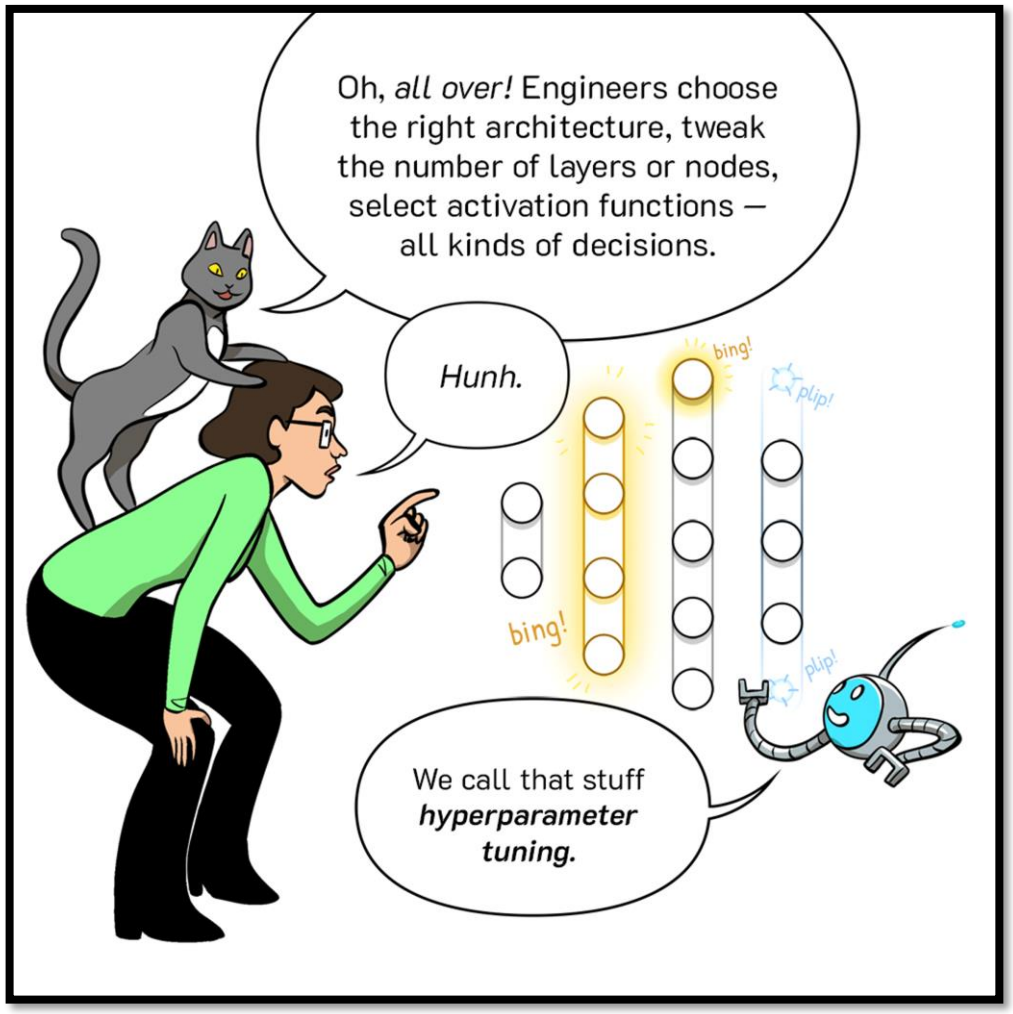

And now the output of that node –
OH!

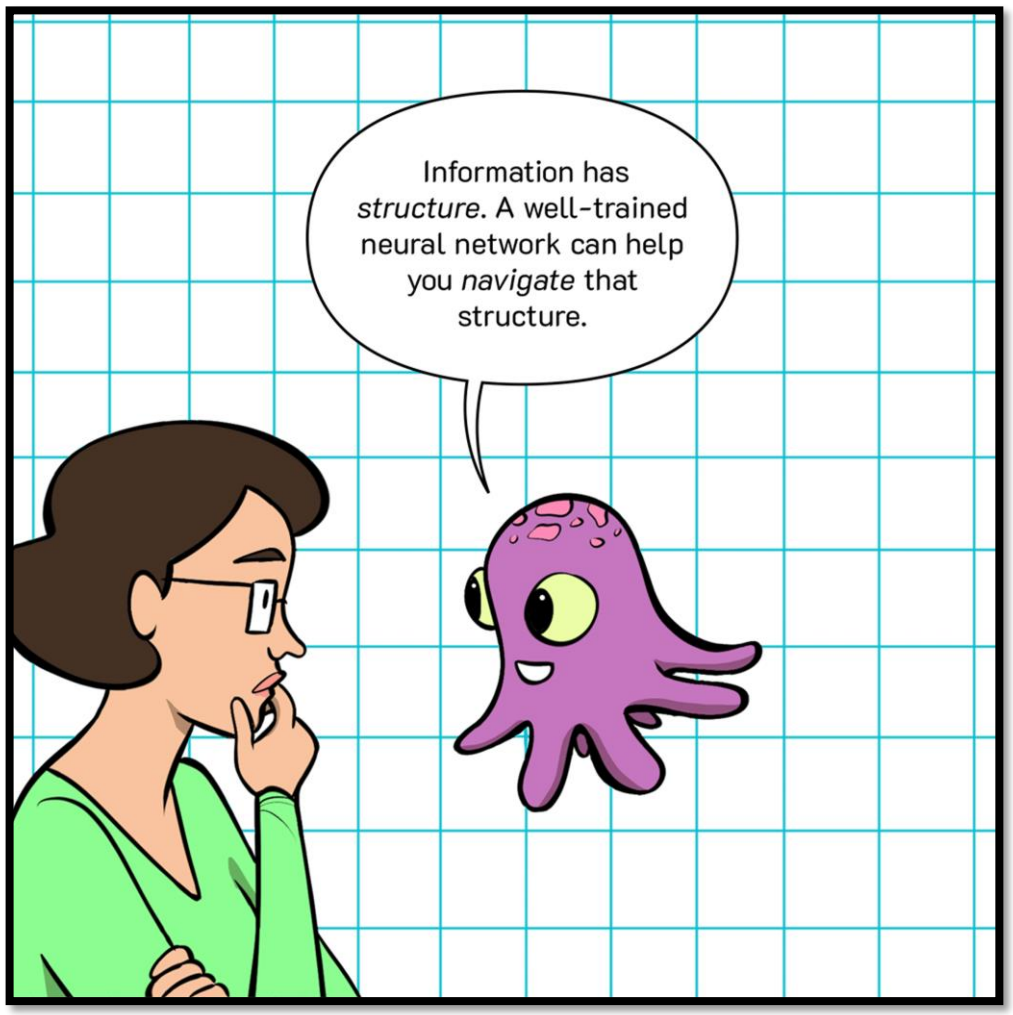

...can be the *input* for another!
Just so!

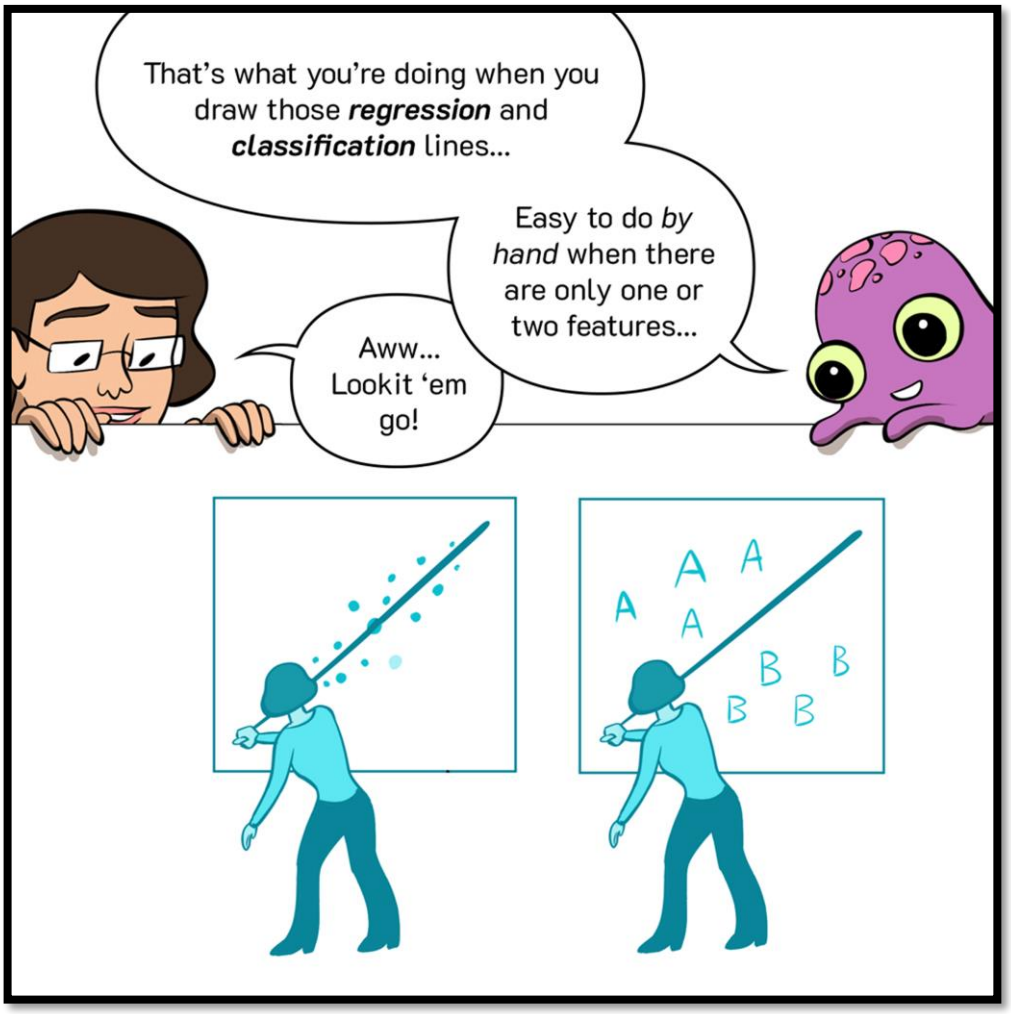

...and another, and another...
Whoa.

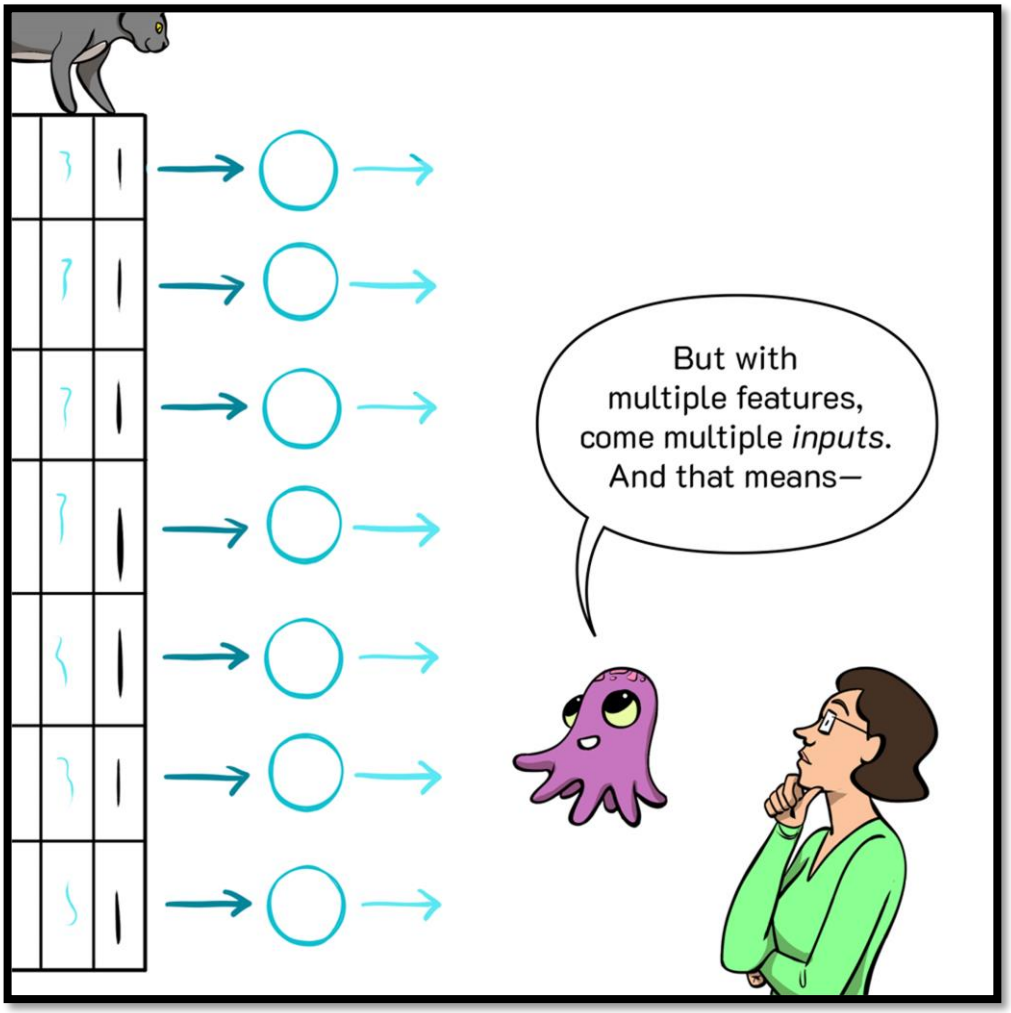

So, when we train neural networks using ***backpropagation*** and ***gradient descent****...
...that process *adjusts* those weights and biases?
LEARN MORE!
Yup!

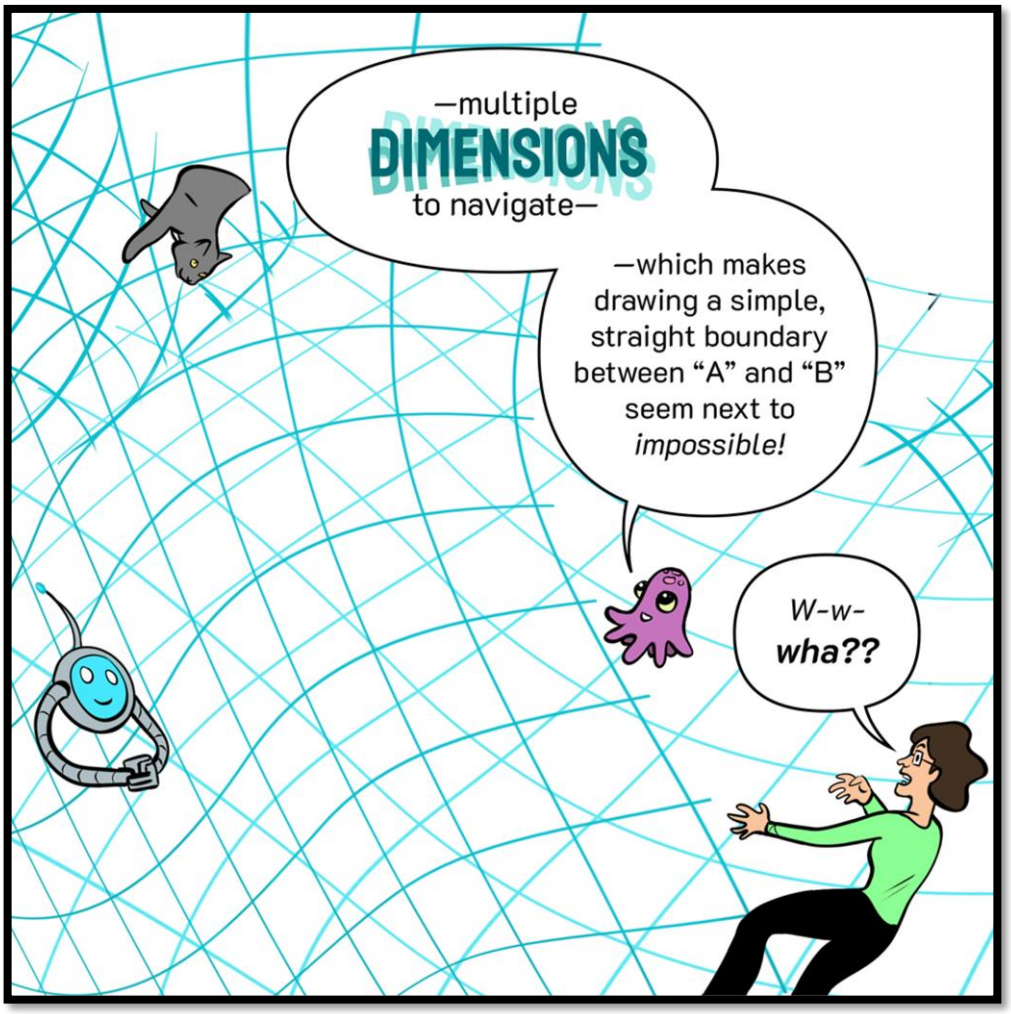


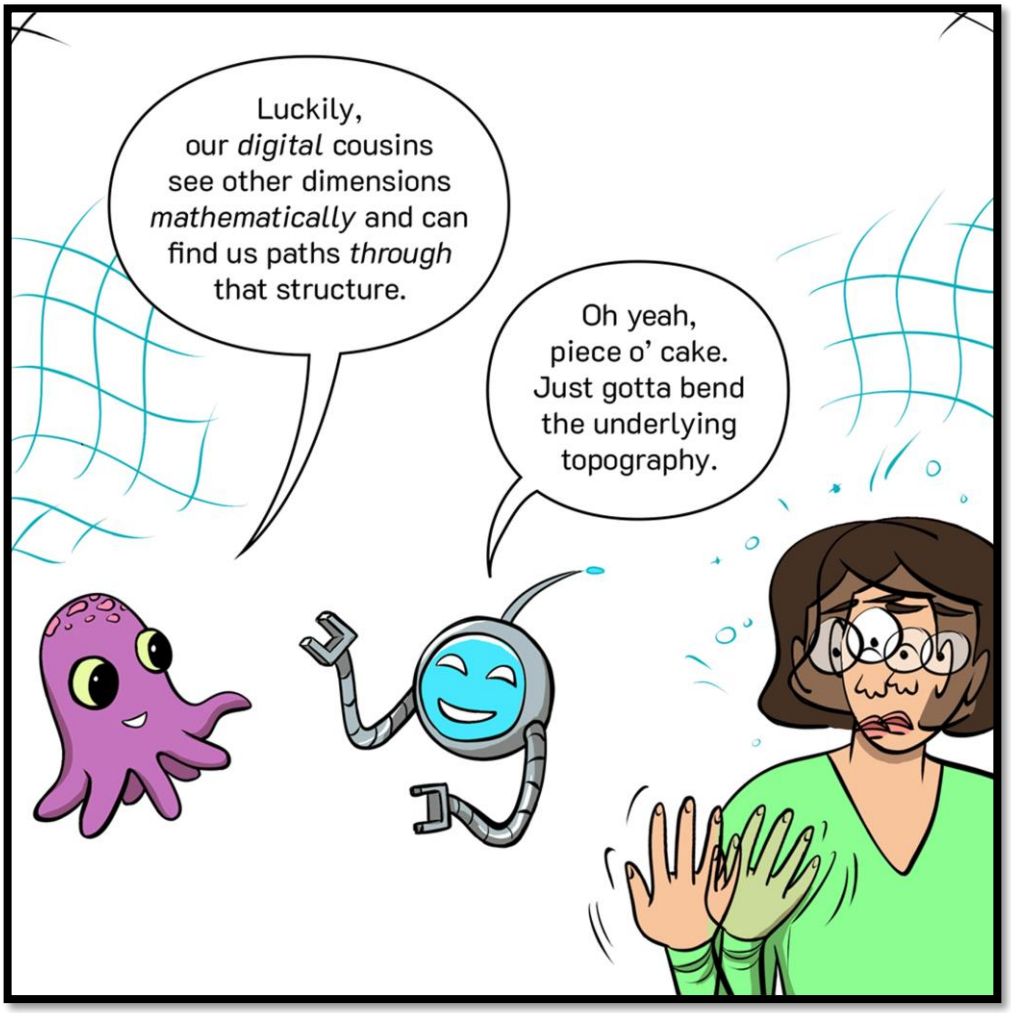

Oh, *all over!* Engineers choose the right architecture, tweak the number of layers or nodes, select activation functions — all kinds of decisions.
Hunh.
bing!
bing!
plip!
plip!
We call that stuff ***hyperparameter tuning.***

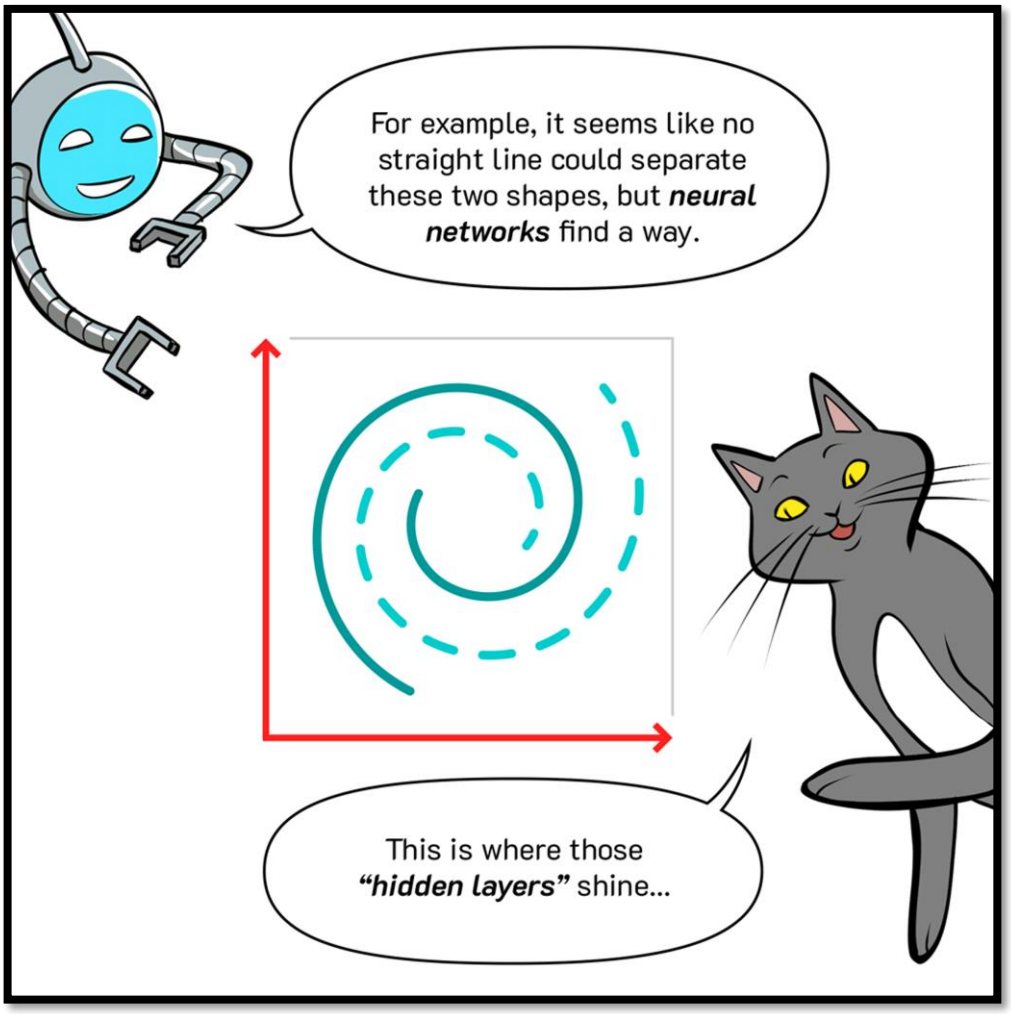

Information has *structure*. A well-trained neural network can help you *navigate* that structure.

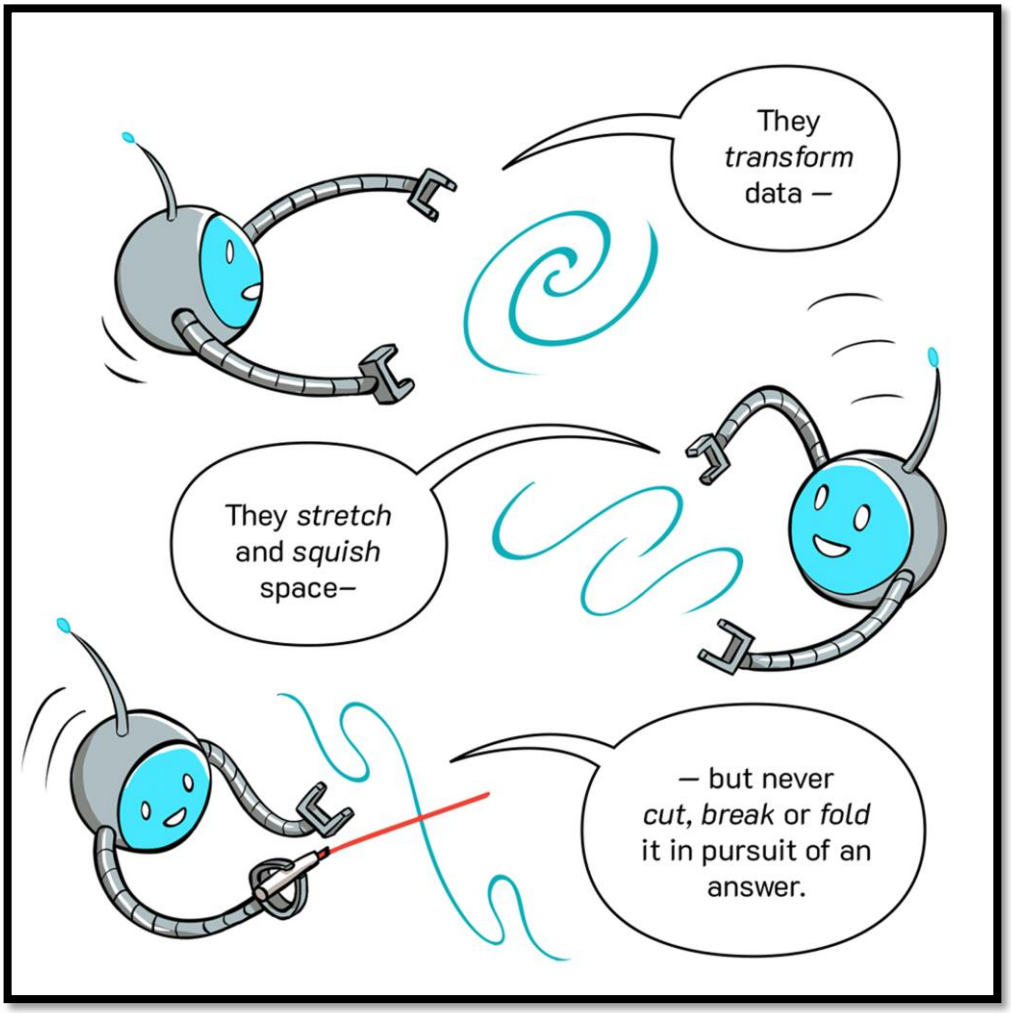

That's what you're doing when you draw those ***regression*** and ***classification*** lines...
Easy to do *by hand* when there are only one or two features...
Aww... Lookit 'em go!
A
A
A
A
B
B
B
B

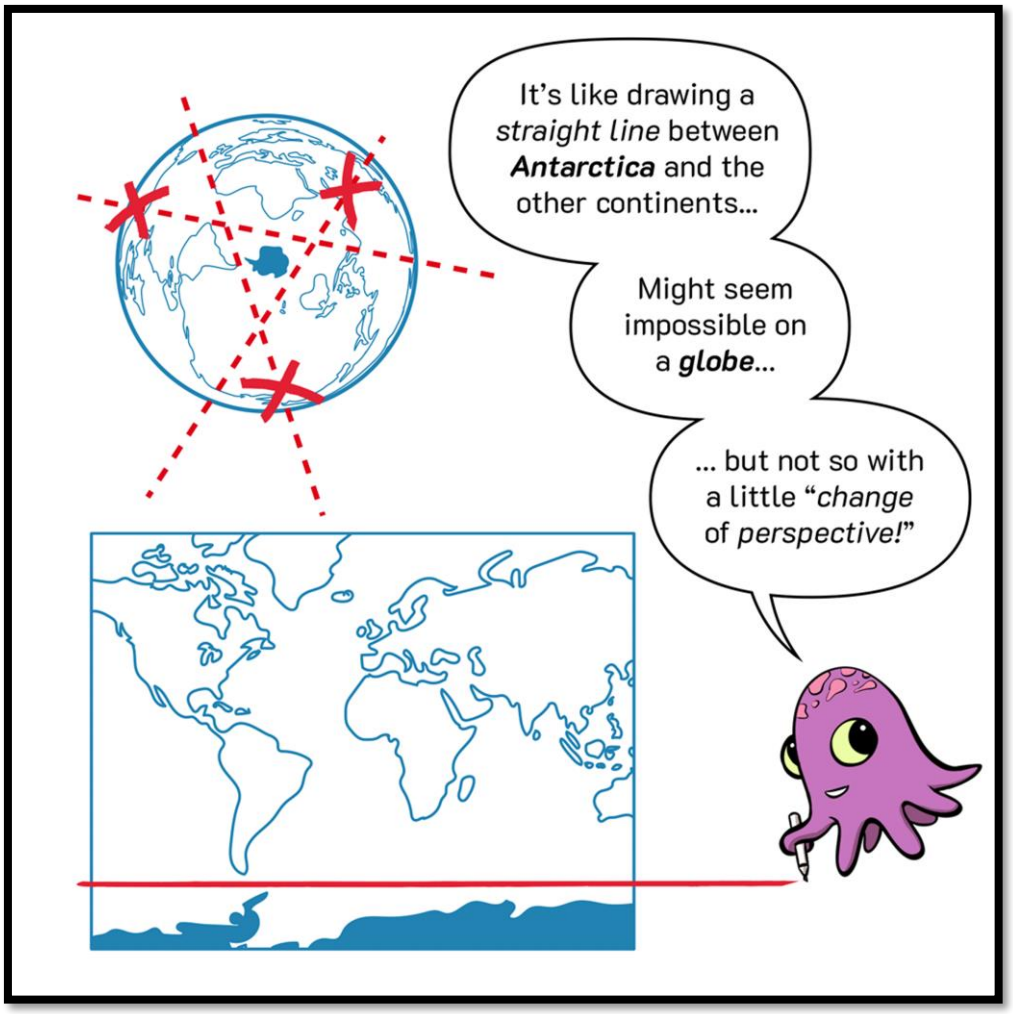

But with multiple features, come multiple *inputs*. And that means—

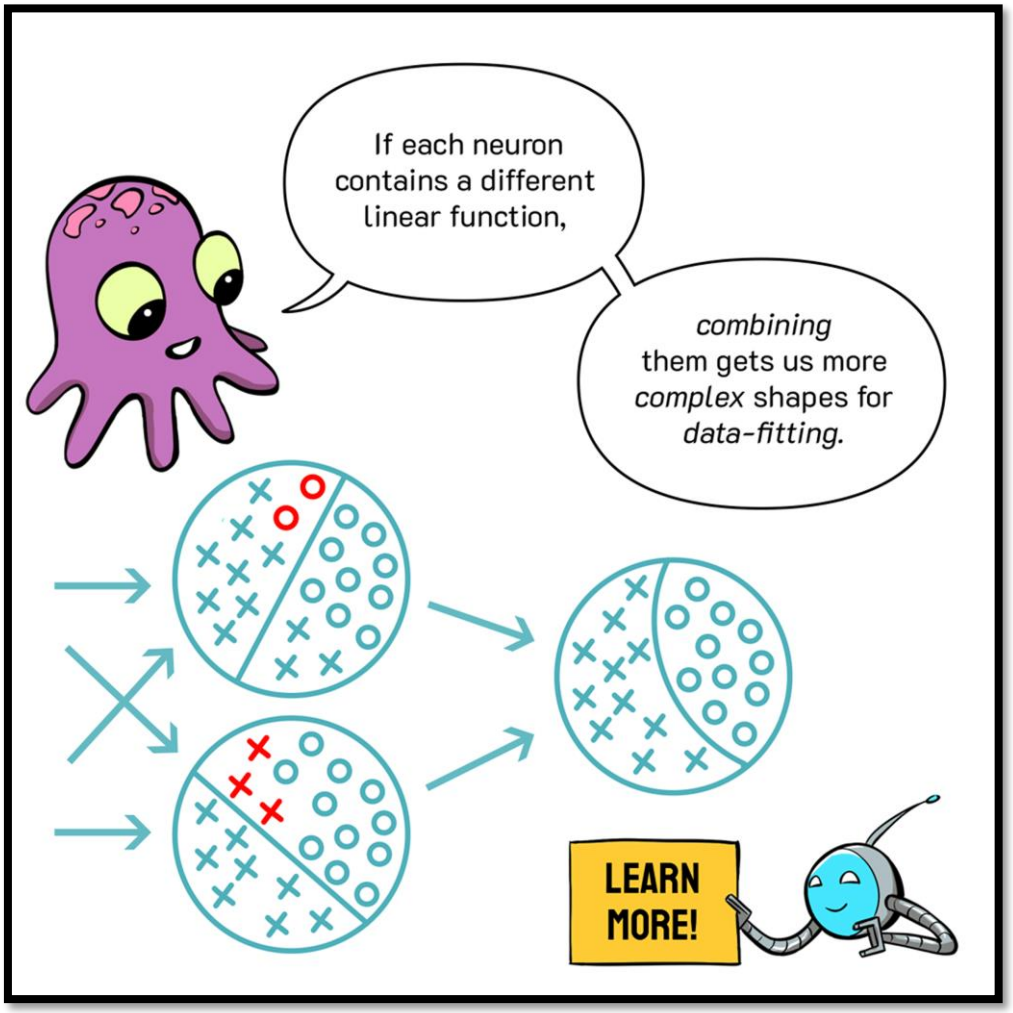

—multiple
DIMENSIONS
to navigate—
—which makes drawing a simple, straight boundary between “A” and “B” seem next to *impossible!*
W-w-
wha??


Luckily, our *digital* cousins see other dimensions *mathematically* and can find us paths *through* that structure.
Oh yeah, piece o’ cake. Just gotta bend the underlying topography.


For example, it seems like no straight line could separate these two shapes, but ***neural networks*** find a way.
This is where those ***"hidden layers"*** shine...


They *transform* data —
They *stretch* and *squish* space—
— but never *cut*, *break* or *fold* it in pursuit of an answer.

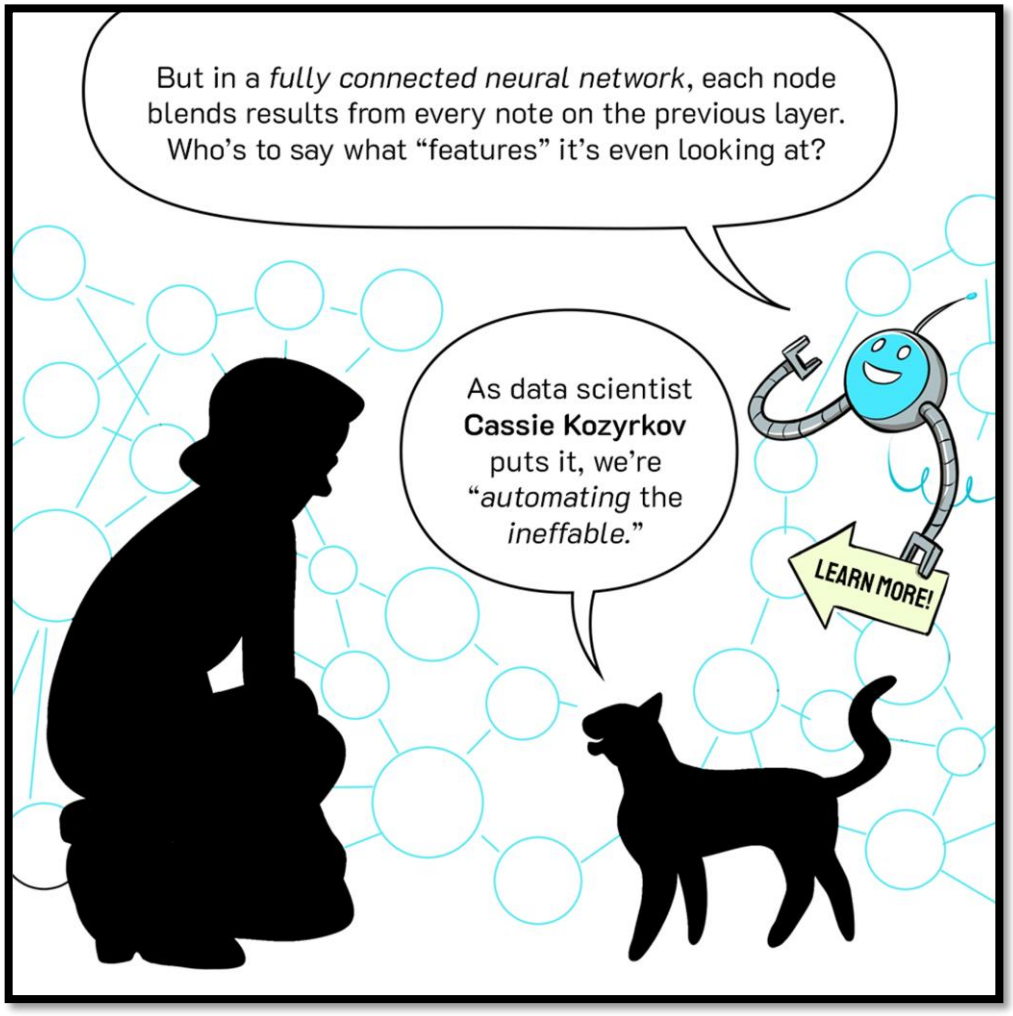

It's like drawing a *straight line* between ***Antarctica*** and the other continents...
Might seem impossible on a ***globe***...
... but not so with a little "*change* of *perspective!*"

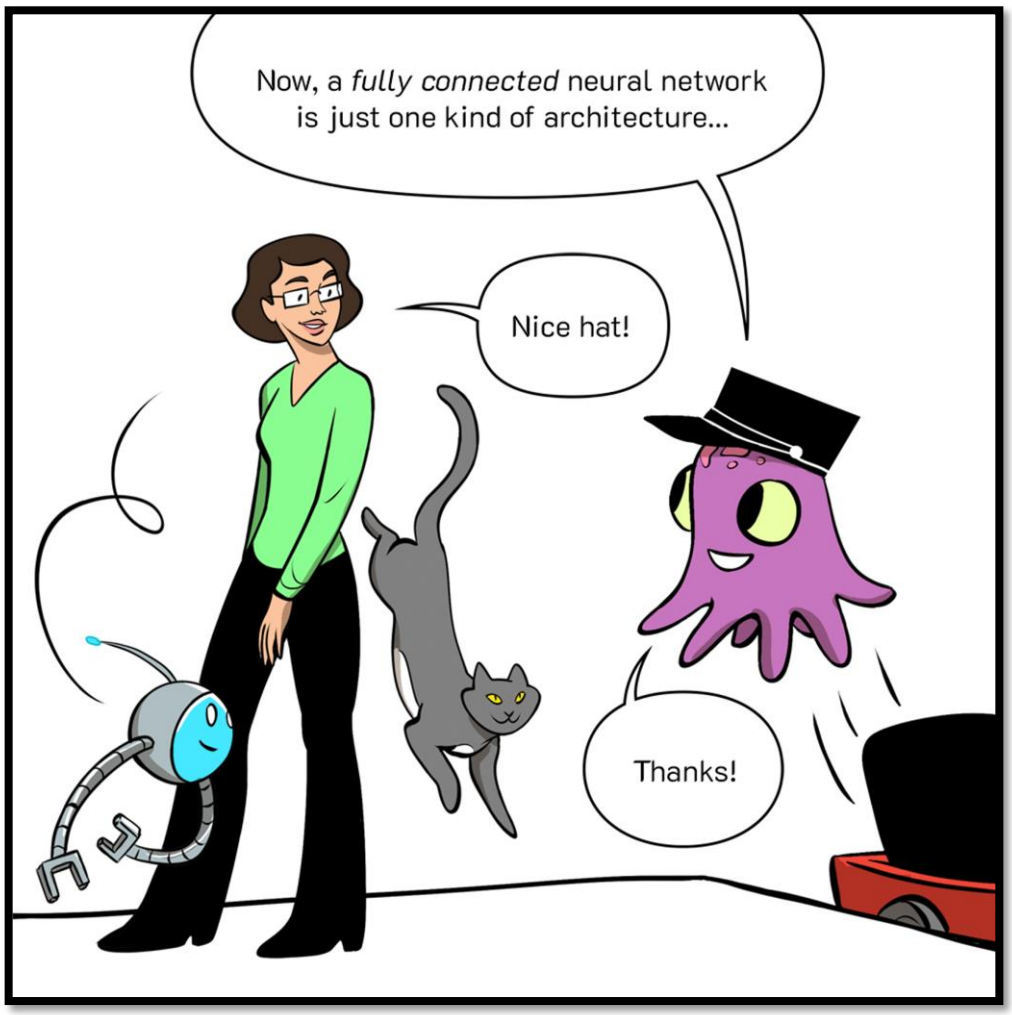

If each neuron contains a different linear function,
combining them gets us more *complex* shapes for *data-fitting.*
LEARN MORE!

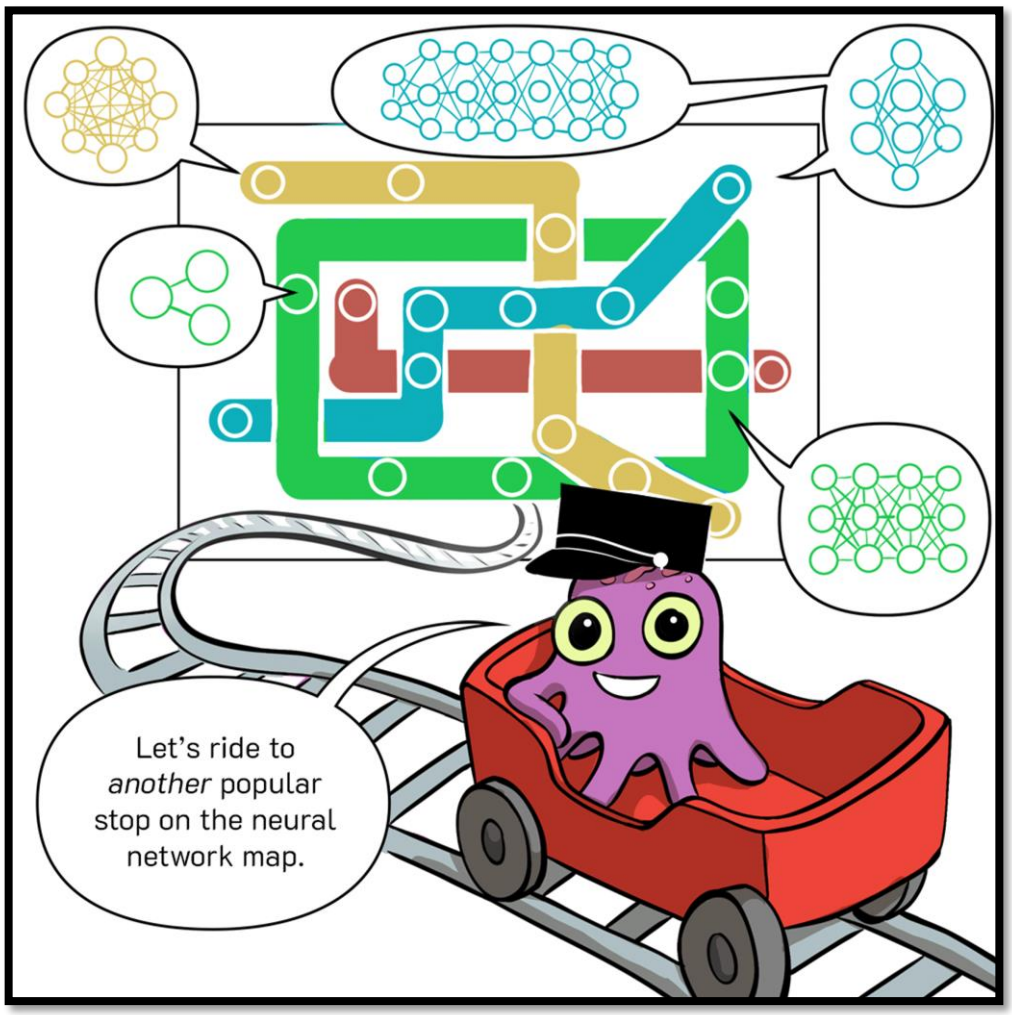

Ooh! So with enough neurons, I could fit ANY dataset, no matter how complex?? Mwa-ha-ha...
Not so fast! Too many neurons may lead to overfitting!

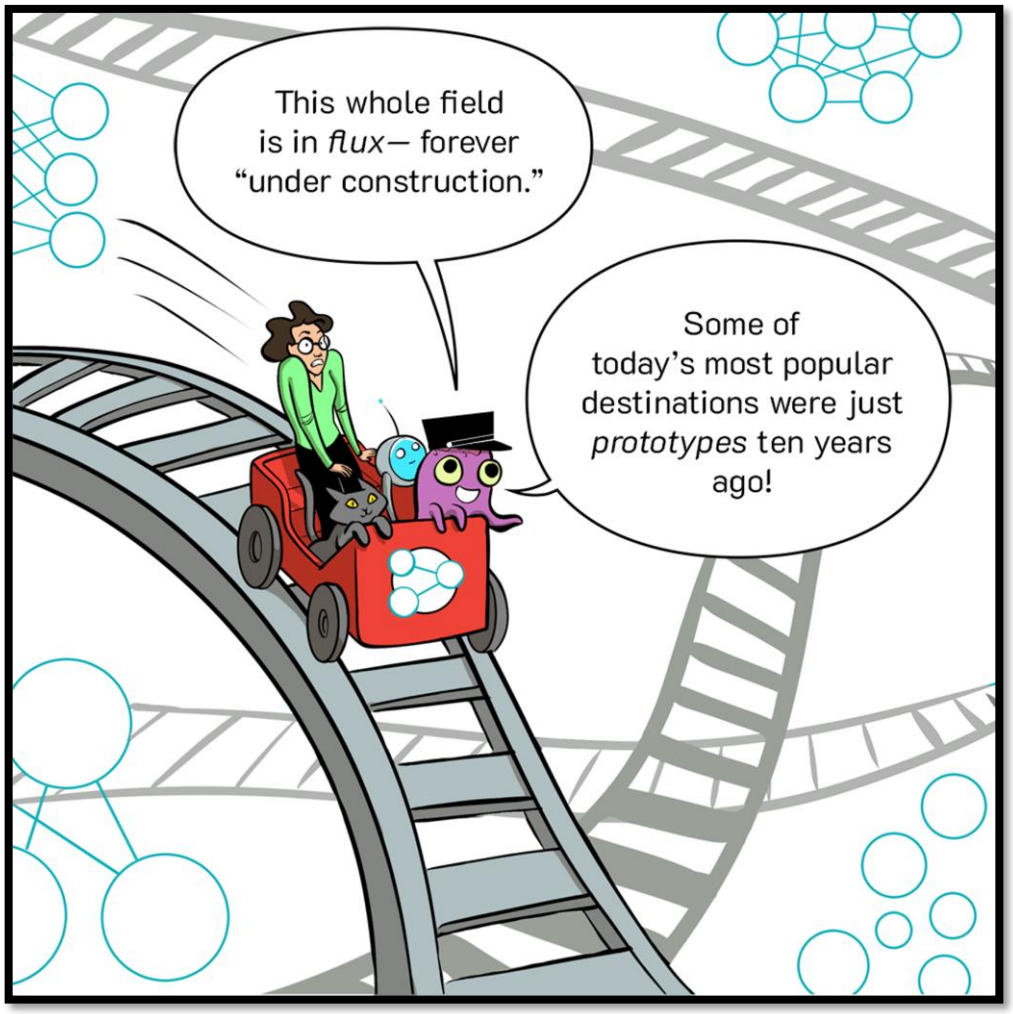

Not to mention inefficiency. Here's your latest energy bill.
Accuracy ain't the only marker of a good network.

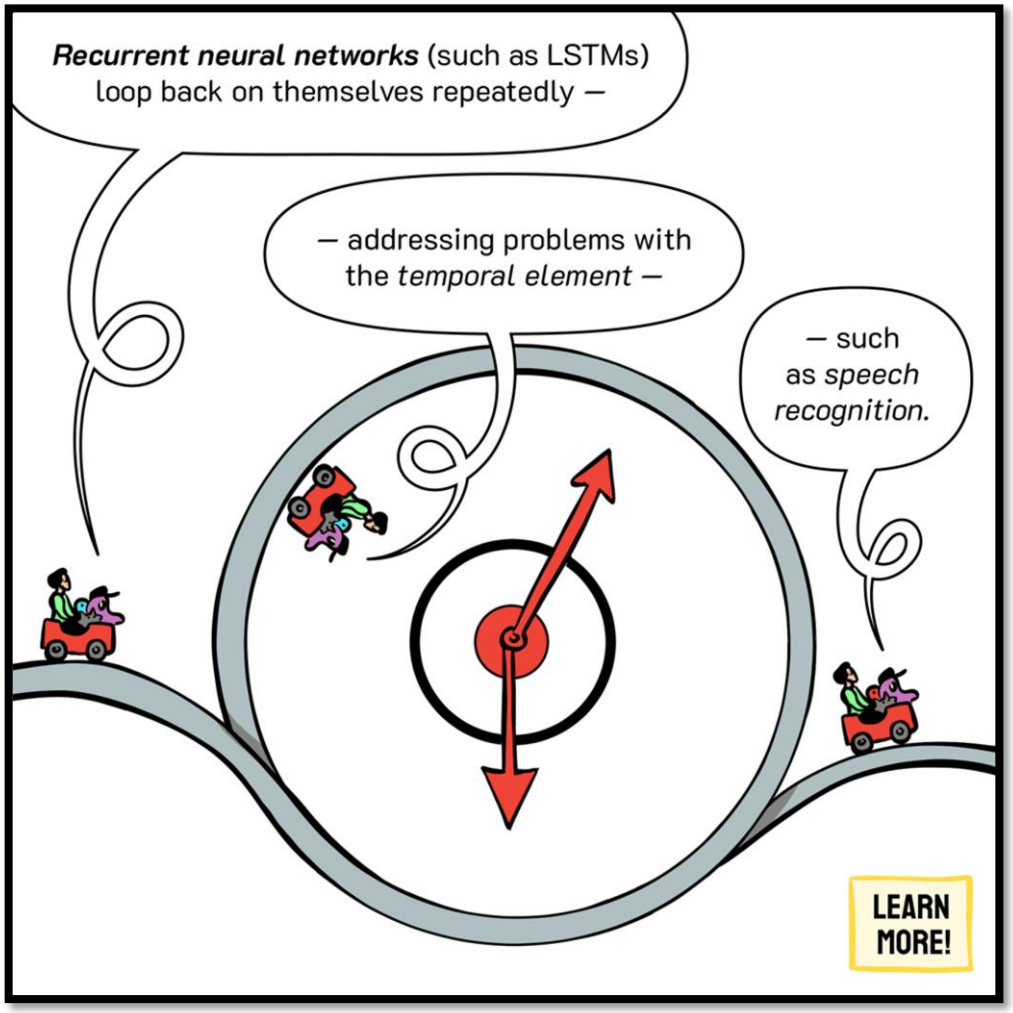

Awww...say, why *do* we call them **"hidden layers?"**
Hoo-boy. Good question.
Wellll... we know what features go *in*... we know what answers come *out*...
... and we can even describe *how* hidden layers work...

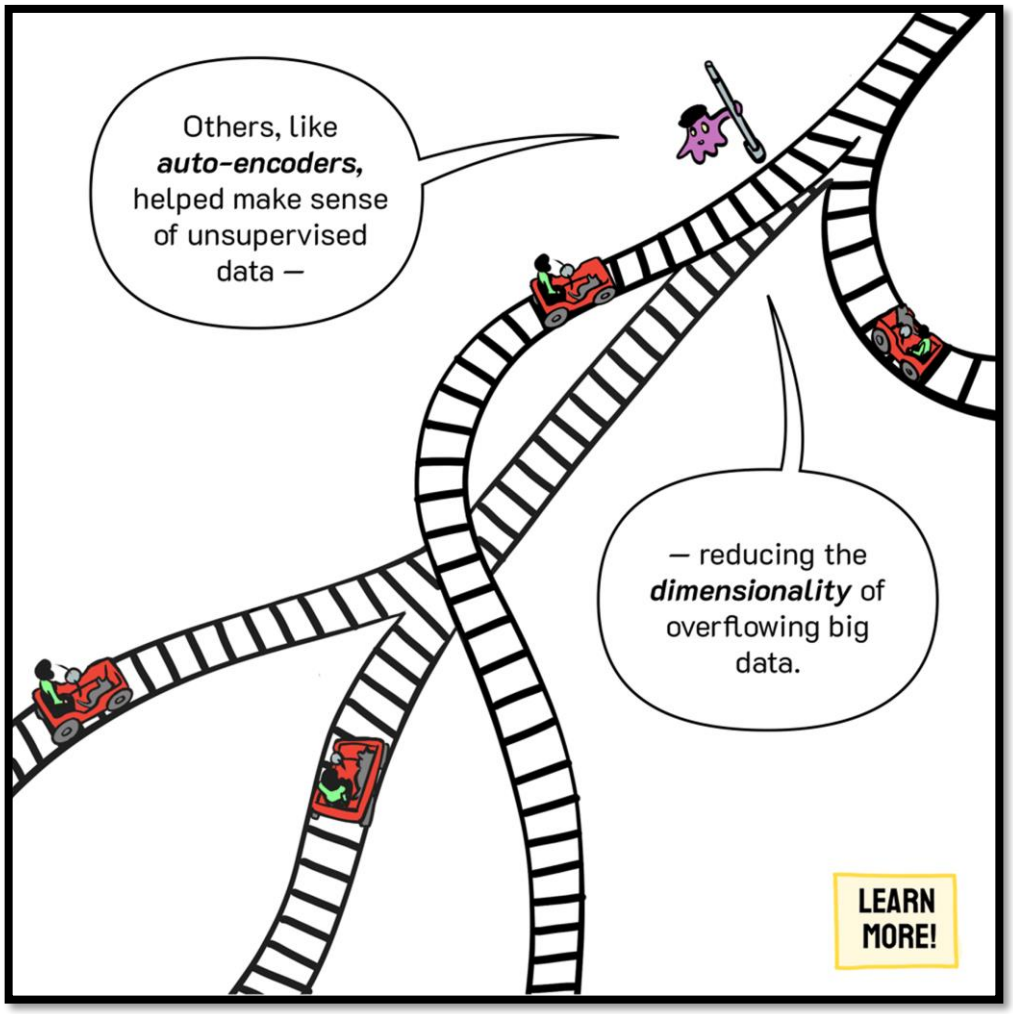

But in a *fully connected neural network*, each node blends results from every note on the previous layer. Who's to say what "features" it's even looking at?
As data scientist **Cassie Kozyrkov** puts it, we're "*automating* the *ineffable.*"
LEARN MORE!

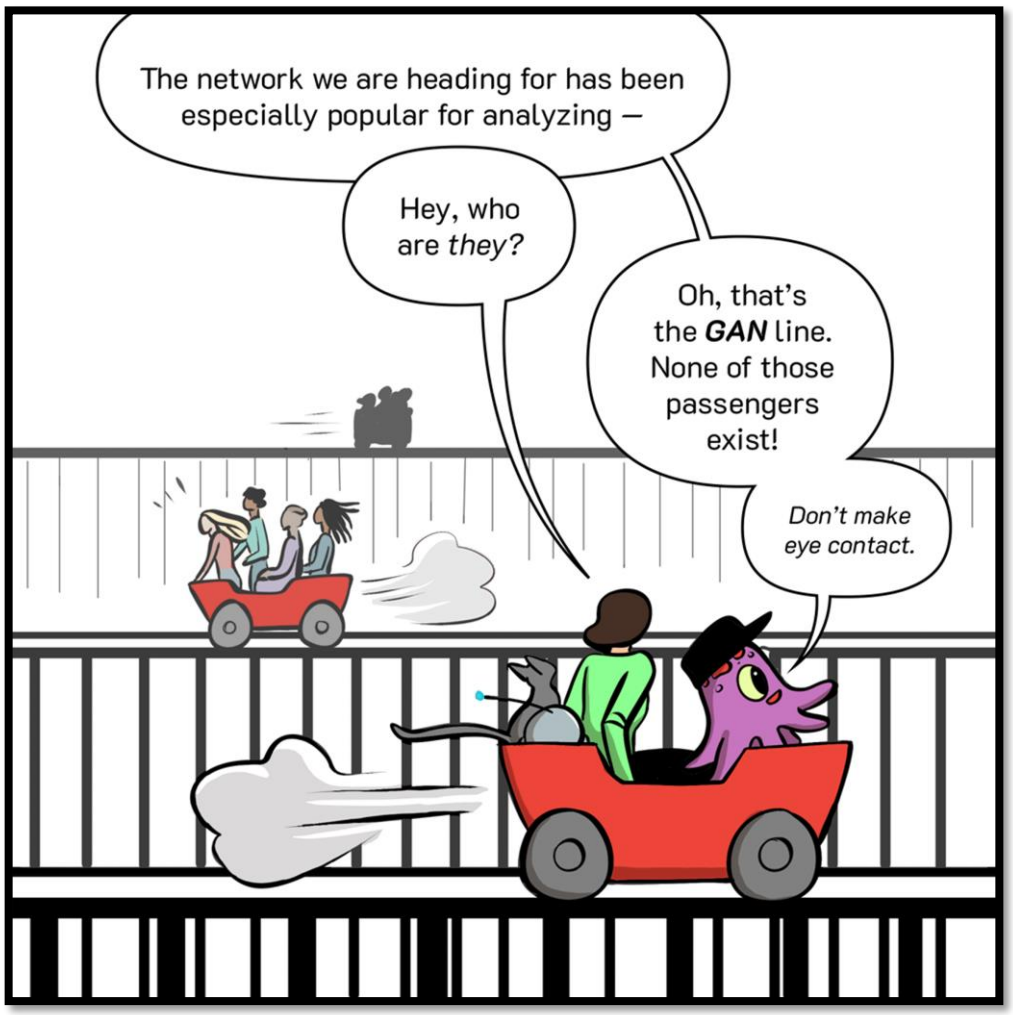

Now, a *fully connected* neural network is just one kind of architecture...
Nice hat!
Thanks!

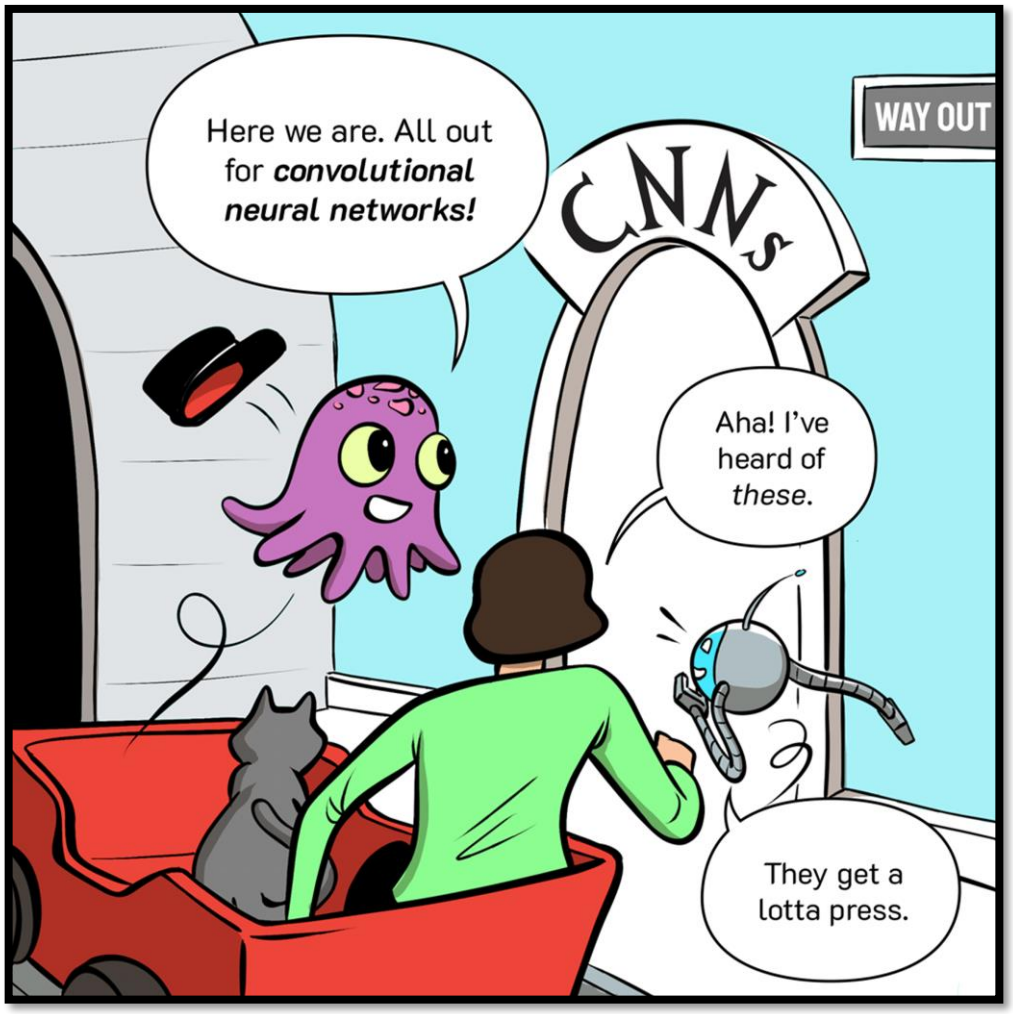

Let's ride to *another* popular stop on the neural network map.

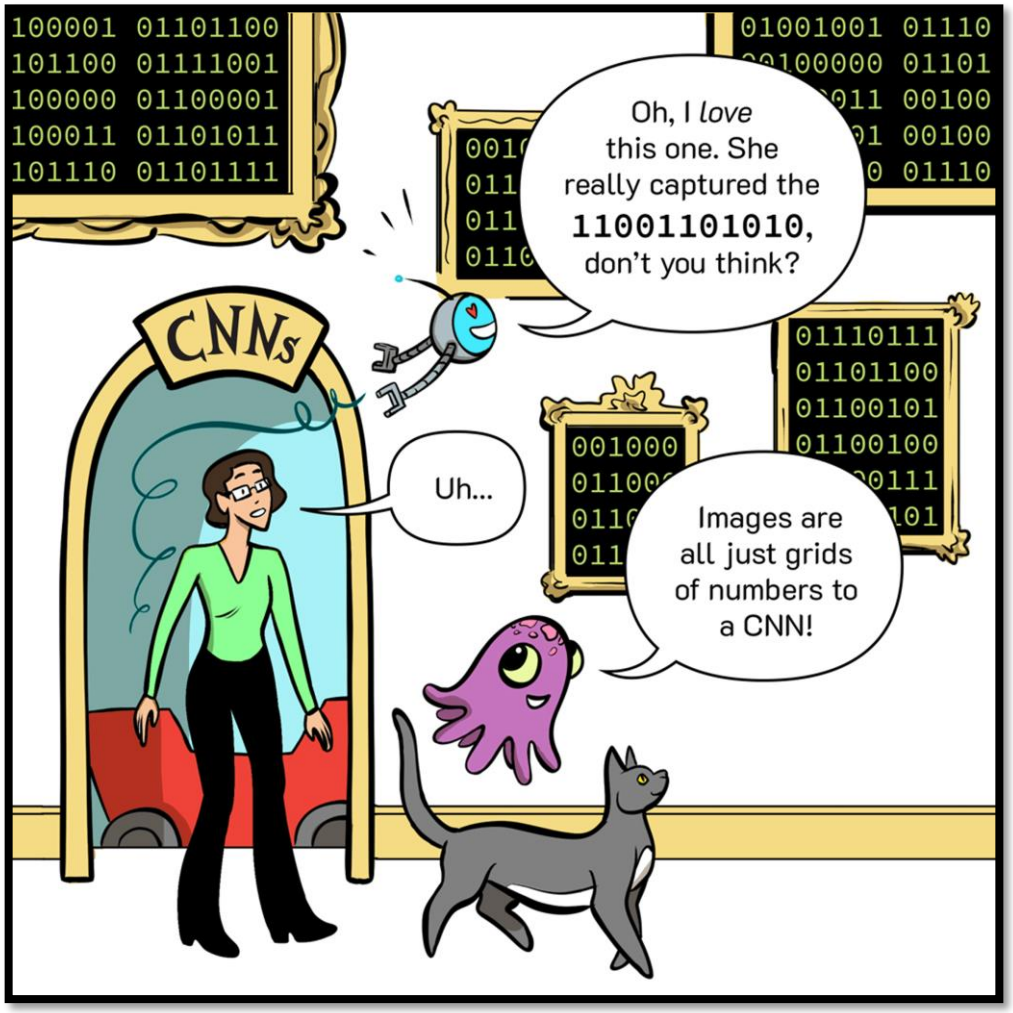

This whole field is in *flux*— forever "under construction."
Some of today's most popular destinations were just *prototypes* ten years ago!

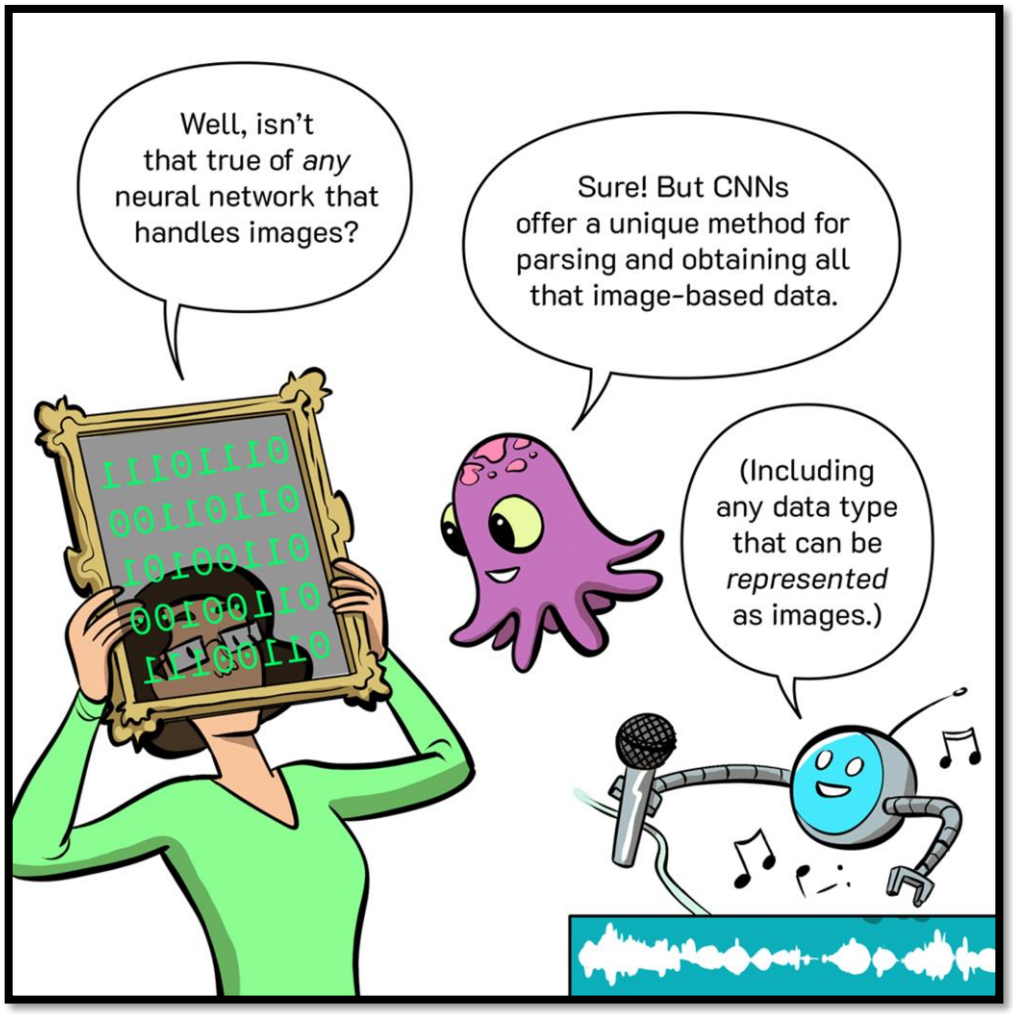

Recurrent neural networks (such as LSTMs) loop back on themselves repeatedly —
— addressing problems with the *temporal element* —
— such as *speech recognition.*
LEARN MORE!

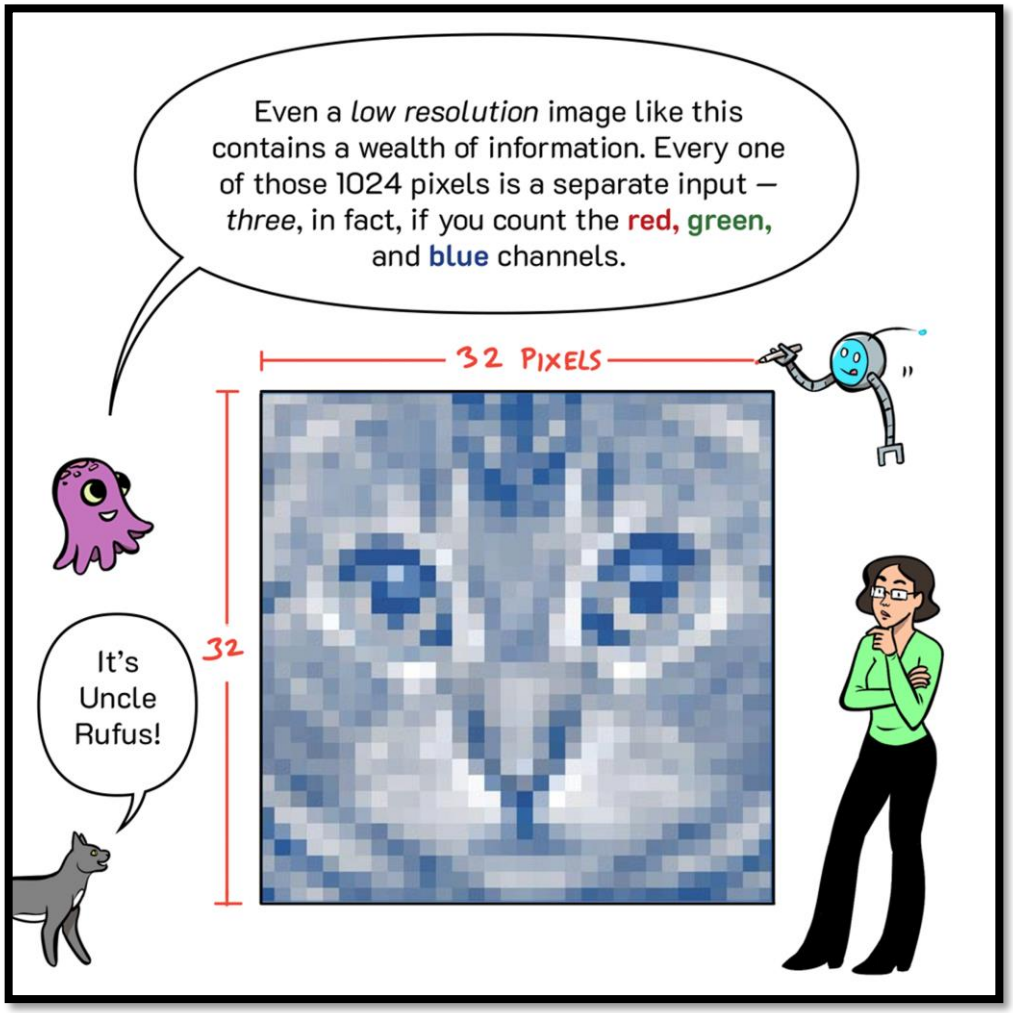

Others, like ***auto-encoders,*** helped make sense of unsupervised data —
— reducing the ***dimensionality*** of overflowing big data.
LEARN MORE!

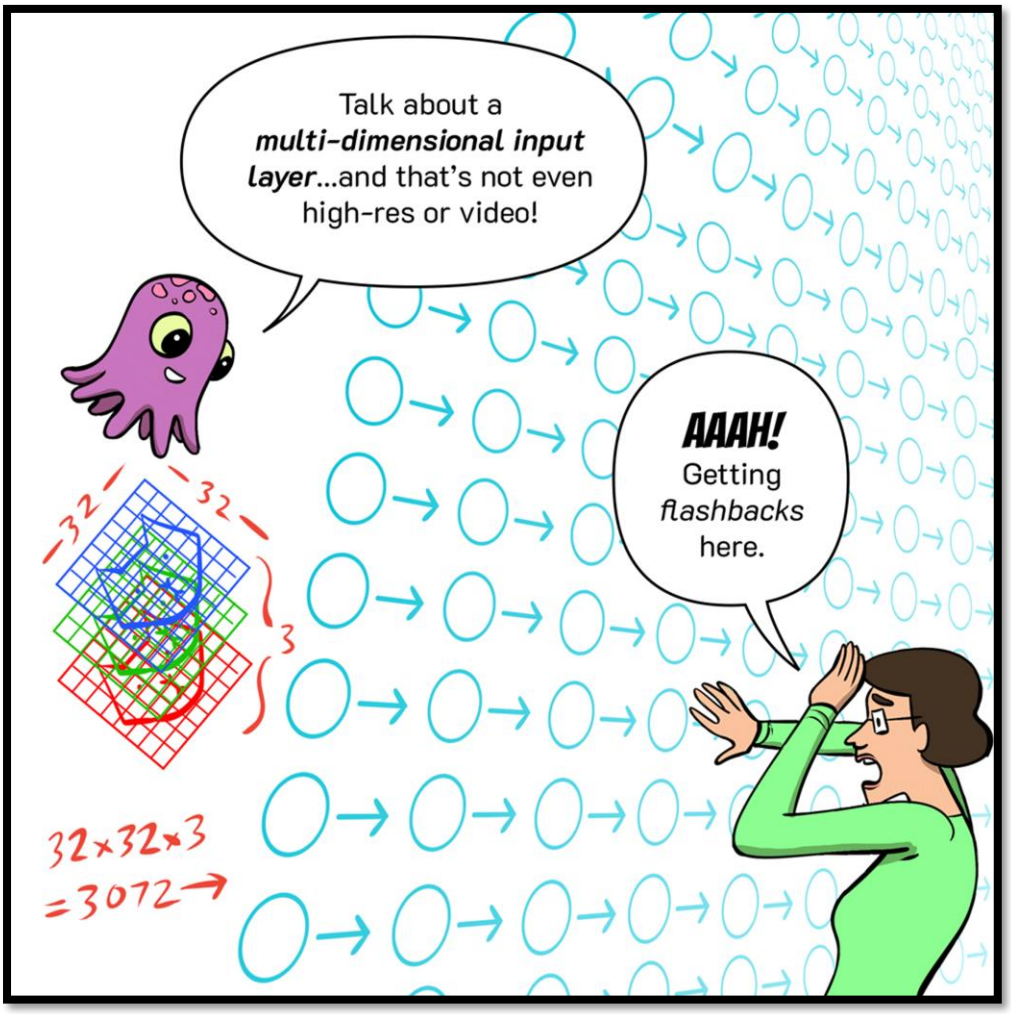

The network we are heading for has been especially popular for analyzing —
Hey, who are *they*?
Oh, that's the ***GAN*** line. None of those passengers exist!
Don't make eye contact.

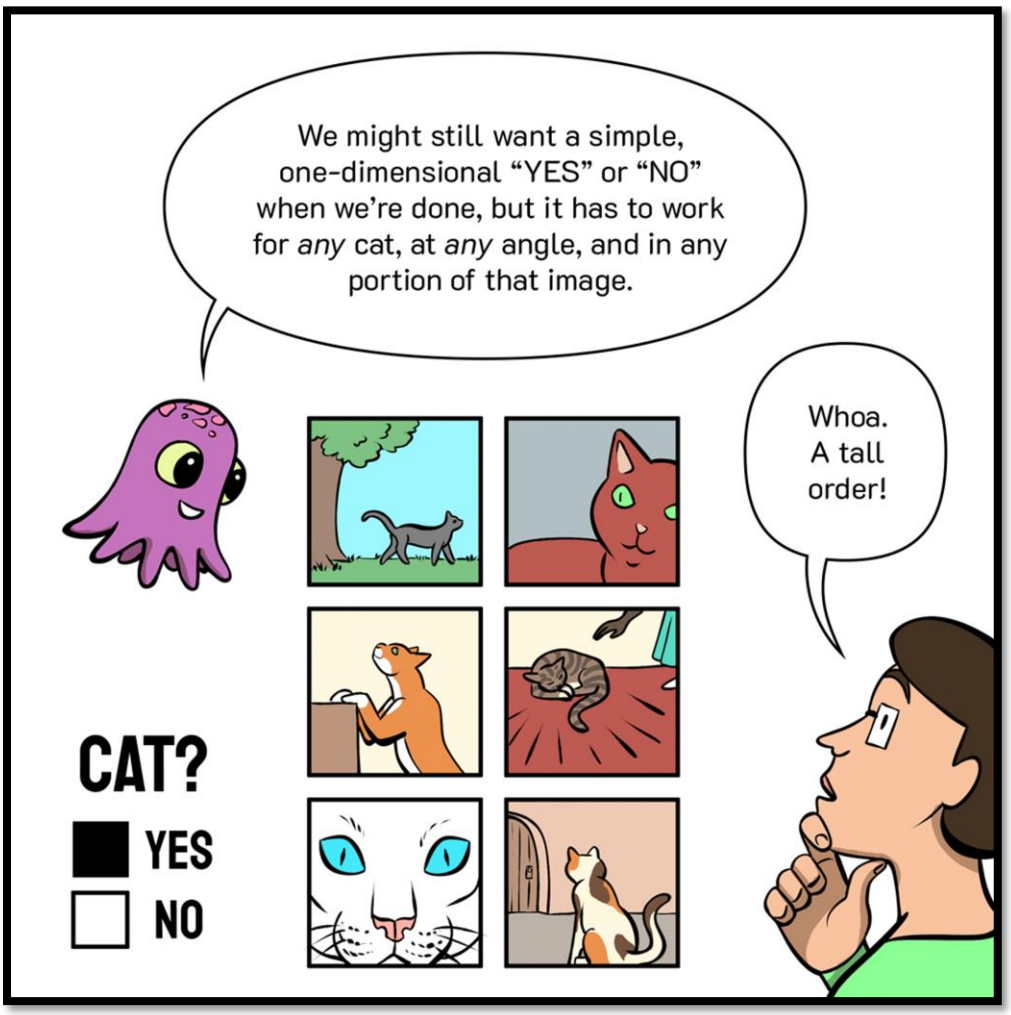

Here we are. All out for ***convolutional neural networks!***
CNNs
WAY OUT
Aha! I've heard of *these*.
They get a lotta press.

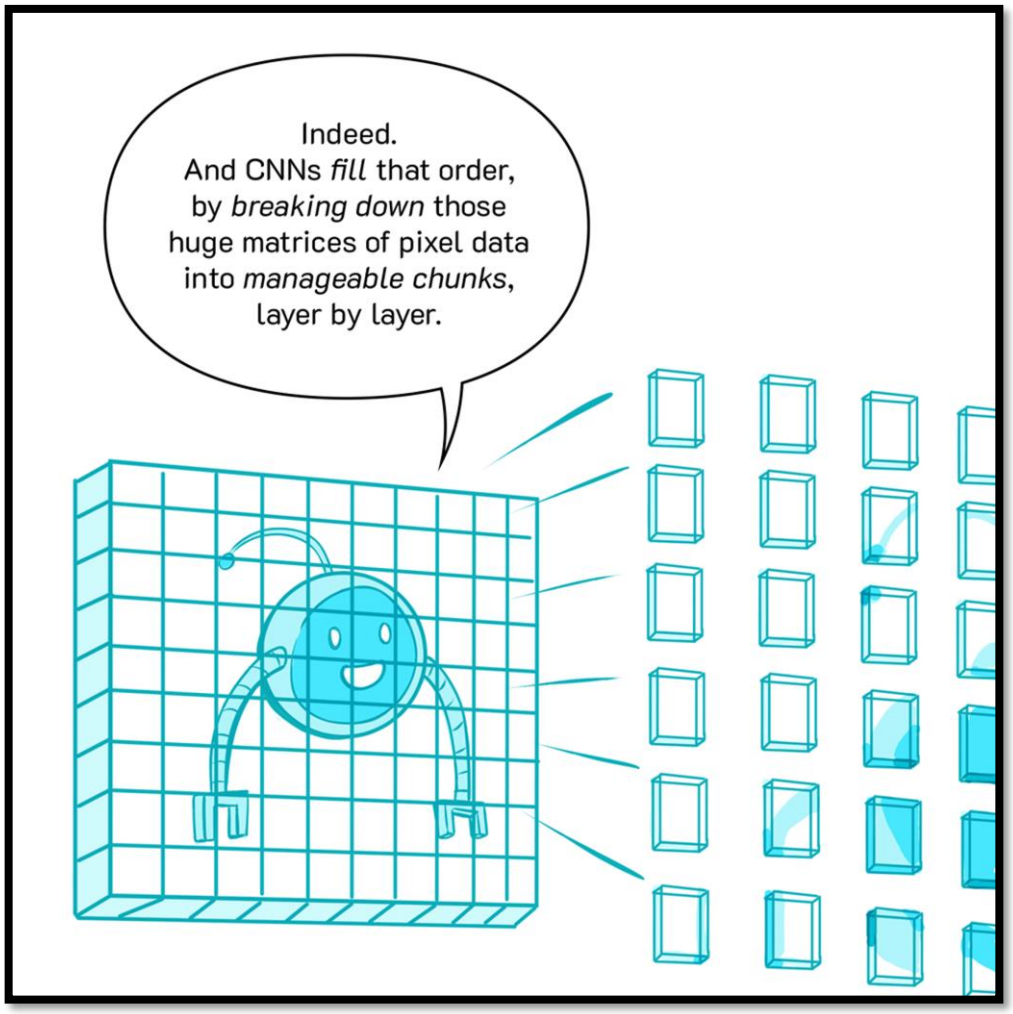

100001 01101100
101100 01111001
100000 01100001
100011 01101011
101110 01101111
Oh, I *love* this one. She really captured the **11001101010**, don't you think?
CNNs
Uh...
Images are all just grids of numbers to a CNN!

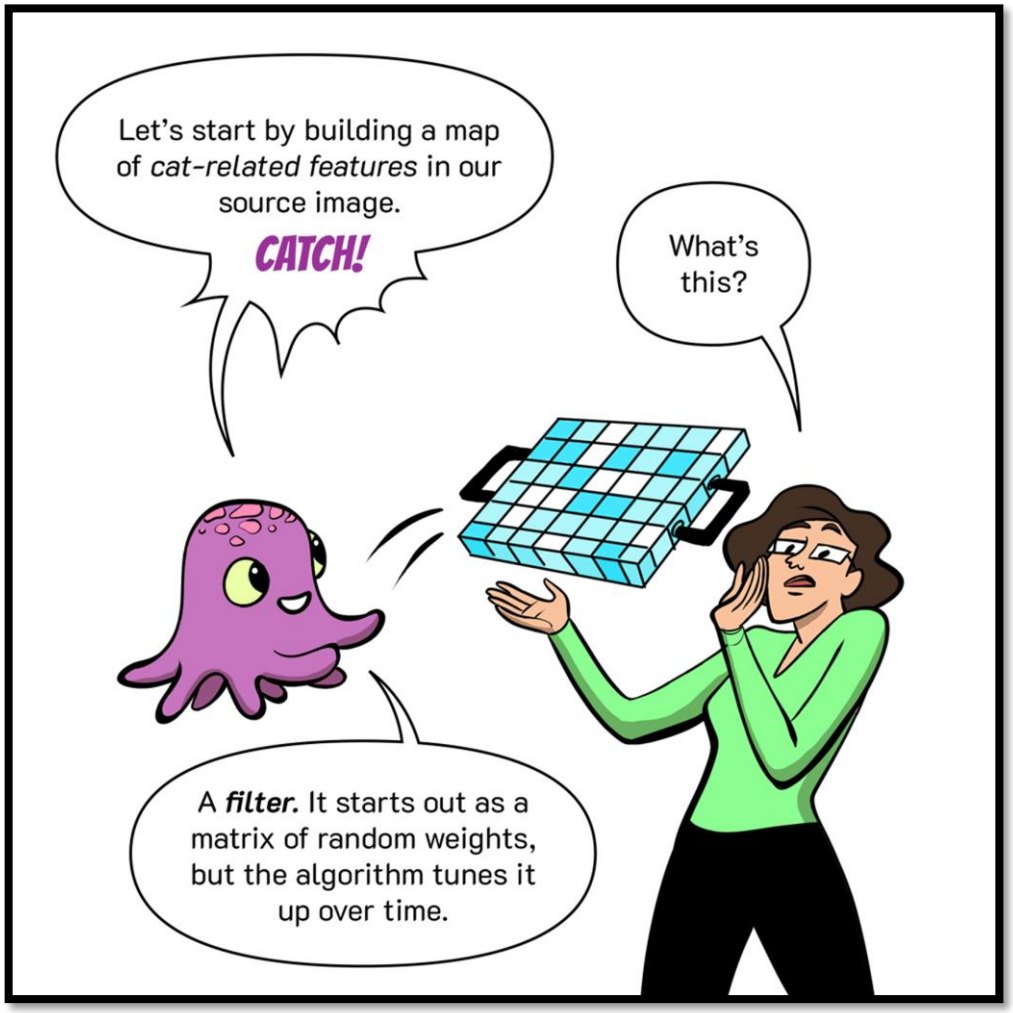

Well, isn't that true of *any* neural network that handles images?
Sure! But CNNs offer a unique method for parsing and obtaining all that image-based data.
(Including any data type that can be *represented* as images.)

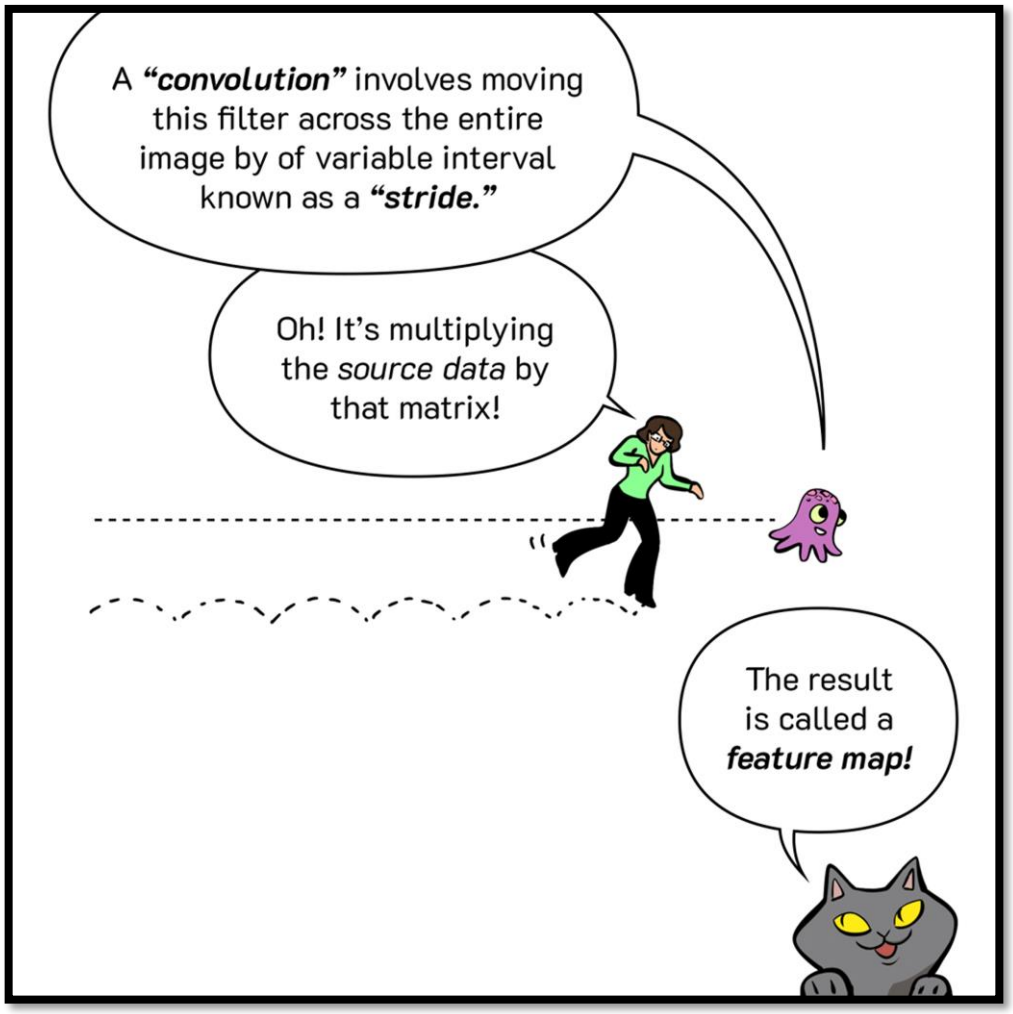

Even a *low resolution* image like this contains a wealth of information. Every one of those 1024 pixels is a separate input — *three*, in fact, if you count the **red**, **green**, and **blue** channels.
32 PIXELS
32
It's Uncle Rufus!

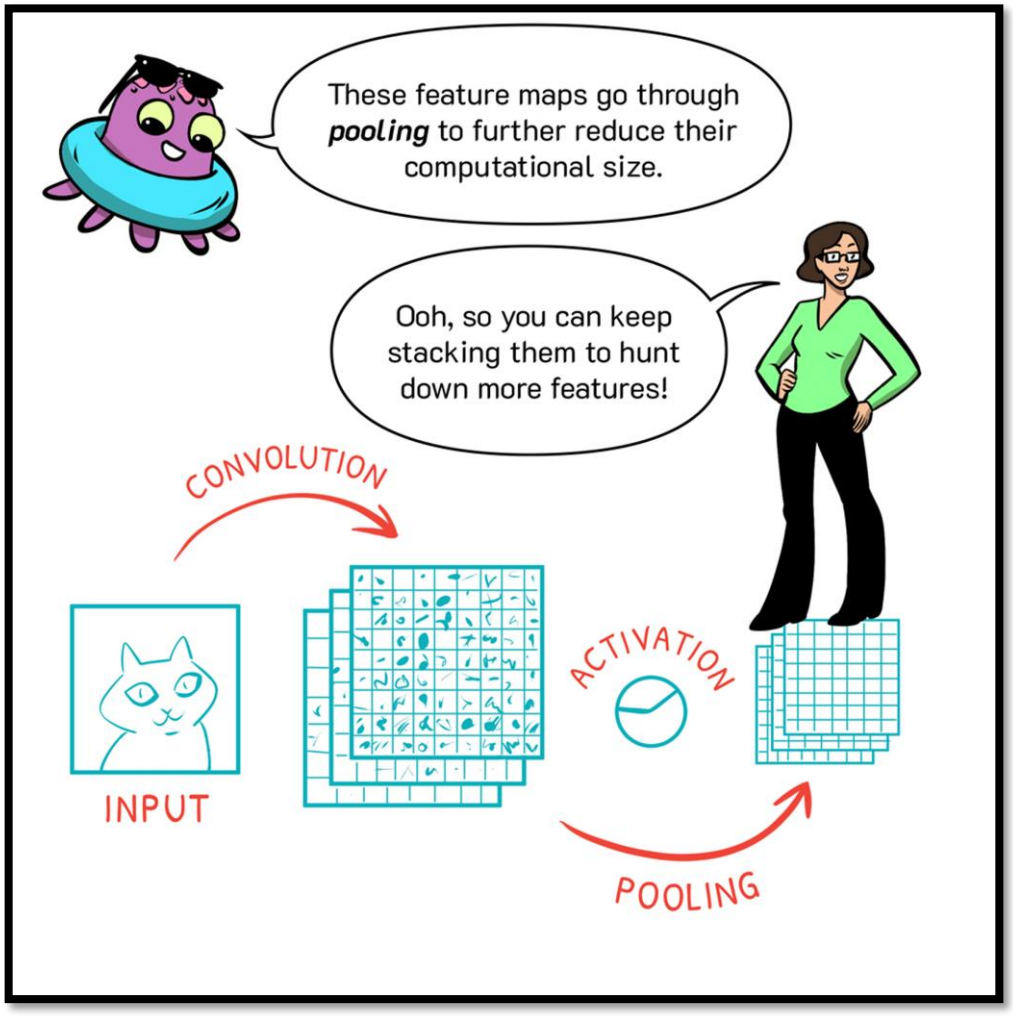

Talk about a ***multi-dimensional input layer***...and that's not even high-res or video!
AAAH! Getting *flashbacks* here.
32
32
3
32×32×3
=3072

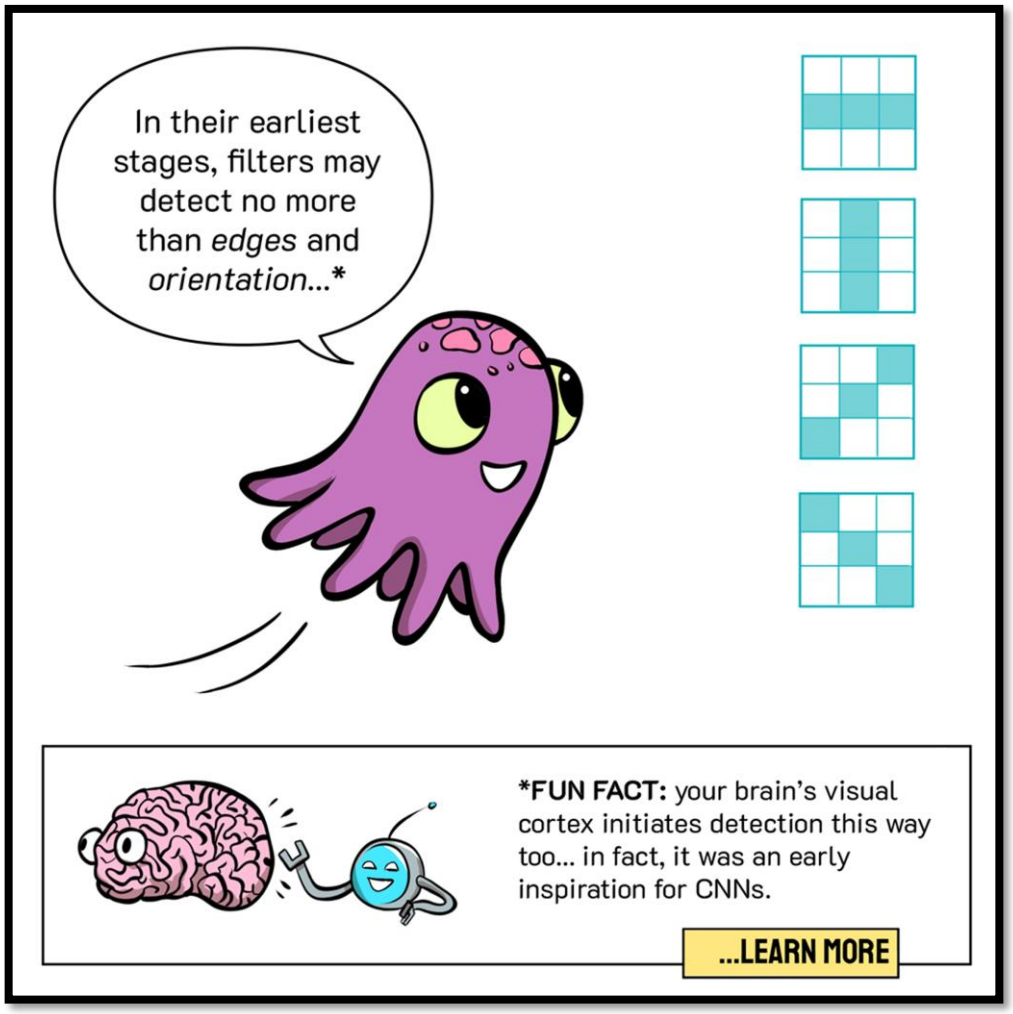

We might still want a simple, one-dimensional "YES" or "NO" when we're done, but it has to work for *any* cat, at *any* angle, and in any portion of that image.
Whoa. A tall order!
CAT?
YES
NO

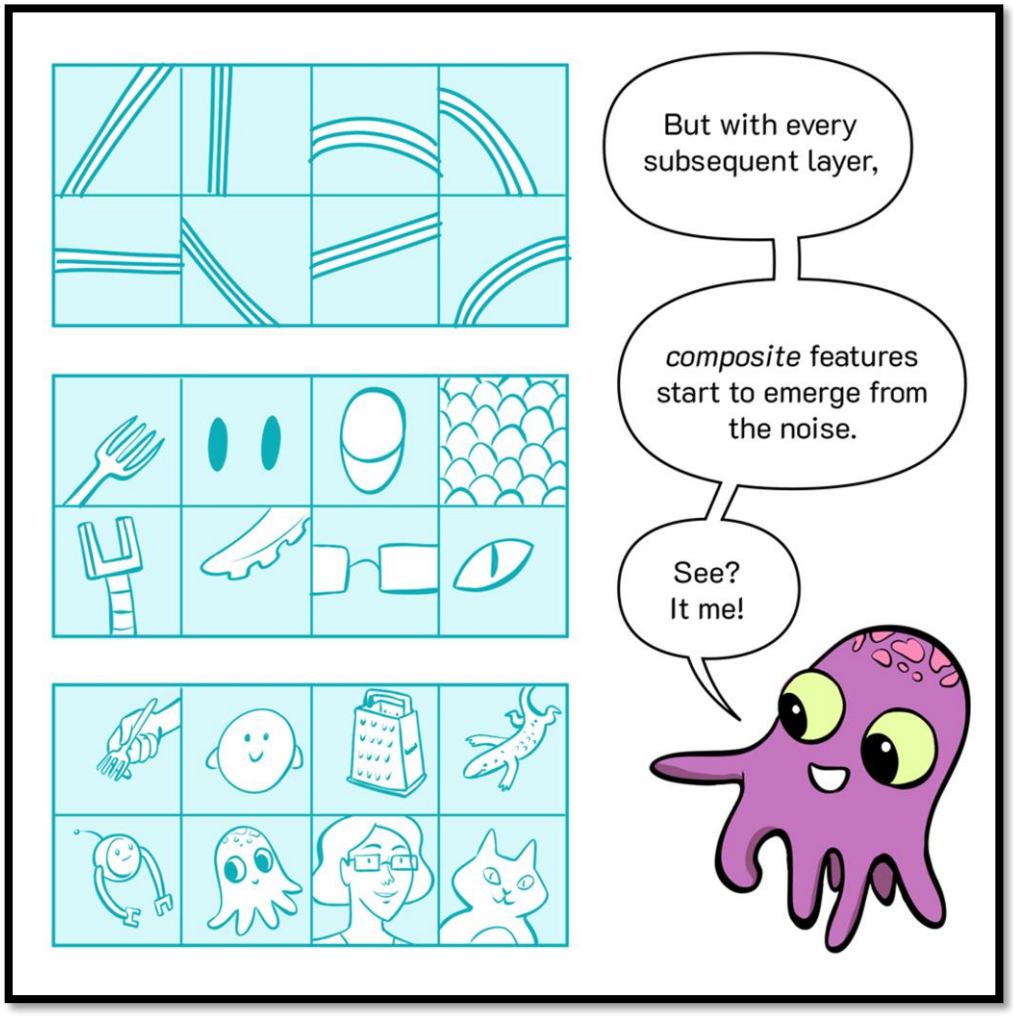

Indeed.
And CNNs *fill* that order, by *breaking down* those huge matrices of pixel data into *manageable chunks*, layer by layer.

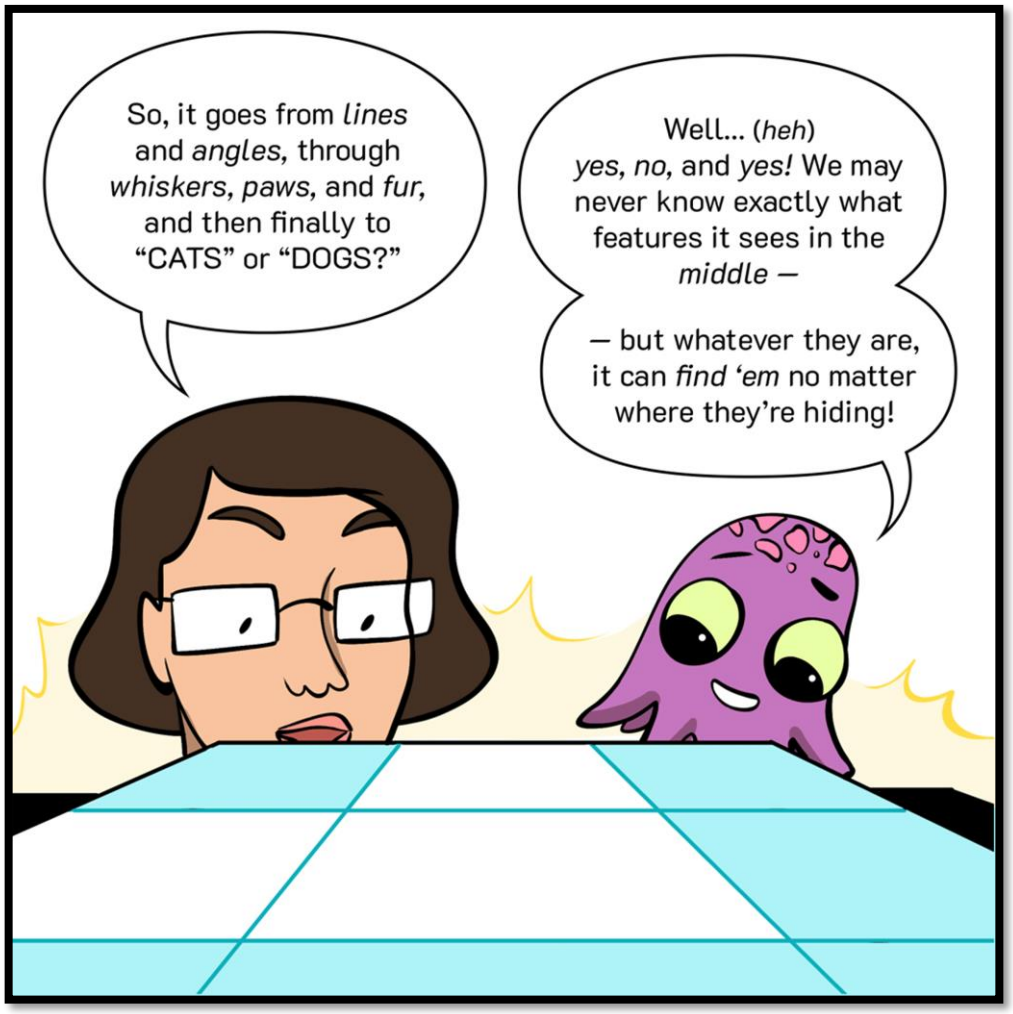

Let's start by building a map of *cat-related features* in our source image.
CATCH!
What's this?
A ***filter.*** It starts out as a matrix of random weights, but the algorithm tunes it up over time.

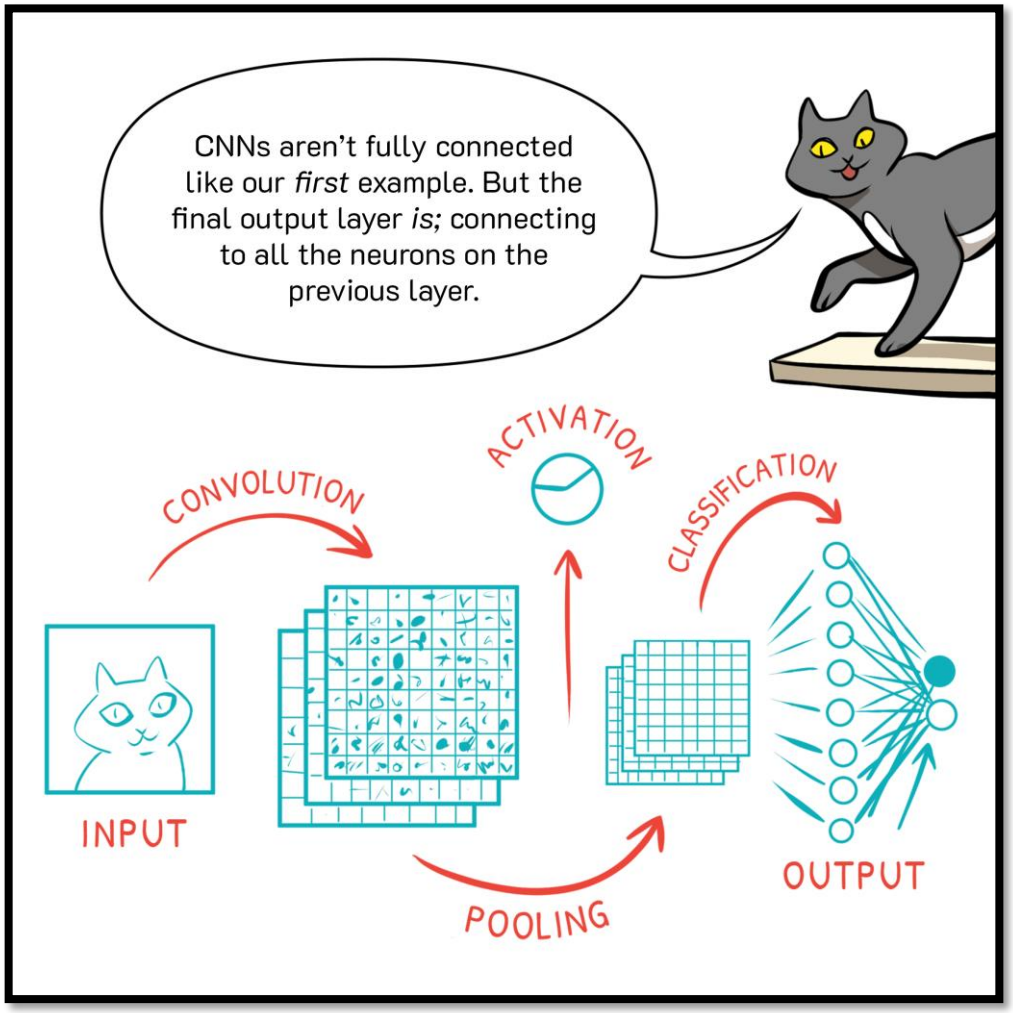

A "convolution" involves moving this filter across the entire image by of variable interval known as a "stride."
Oh! It's multiplying the source data by that matrix!
The result is called a feature map!

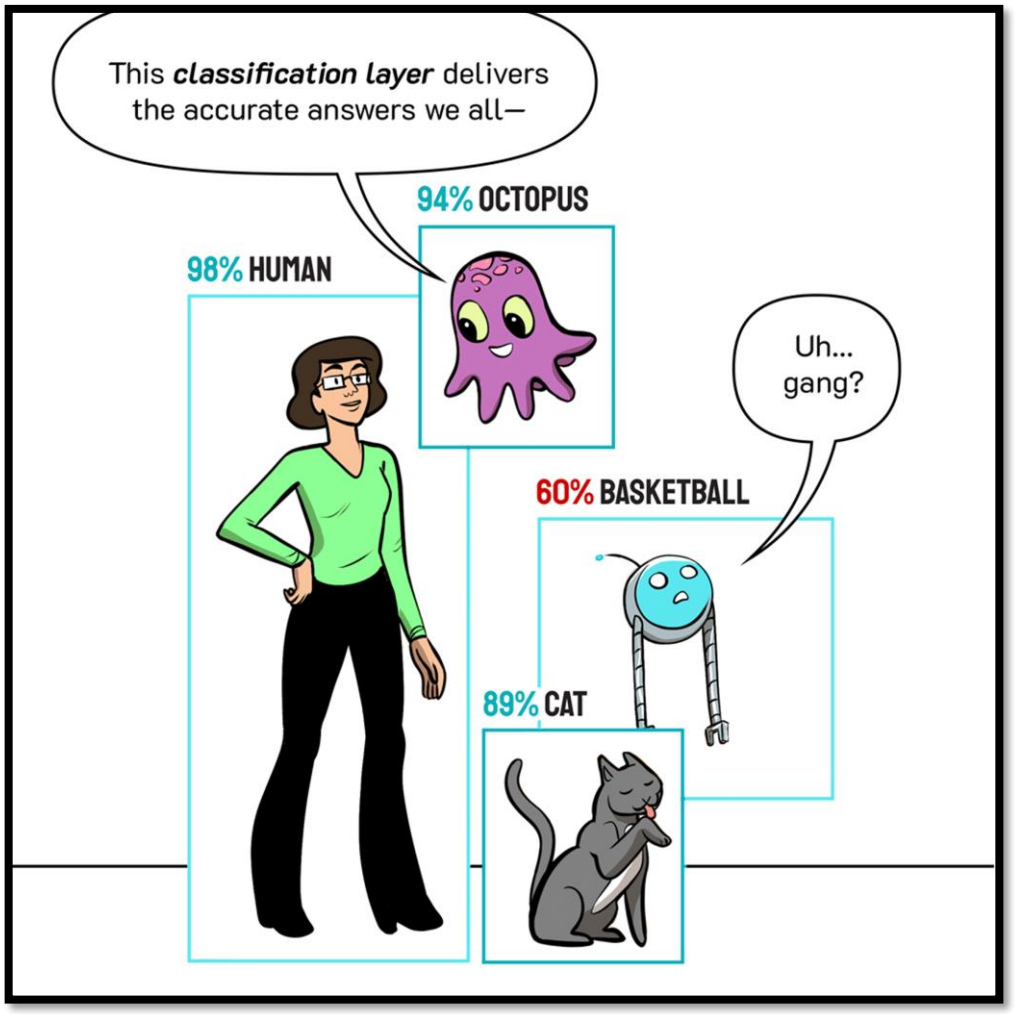

These feature maps go through pooling to further reduce their computational size.
Ooh, so you can keep stacking them to hunt down more features!
CONVOLUTION
ACTIVATION
INPUT
POOLING

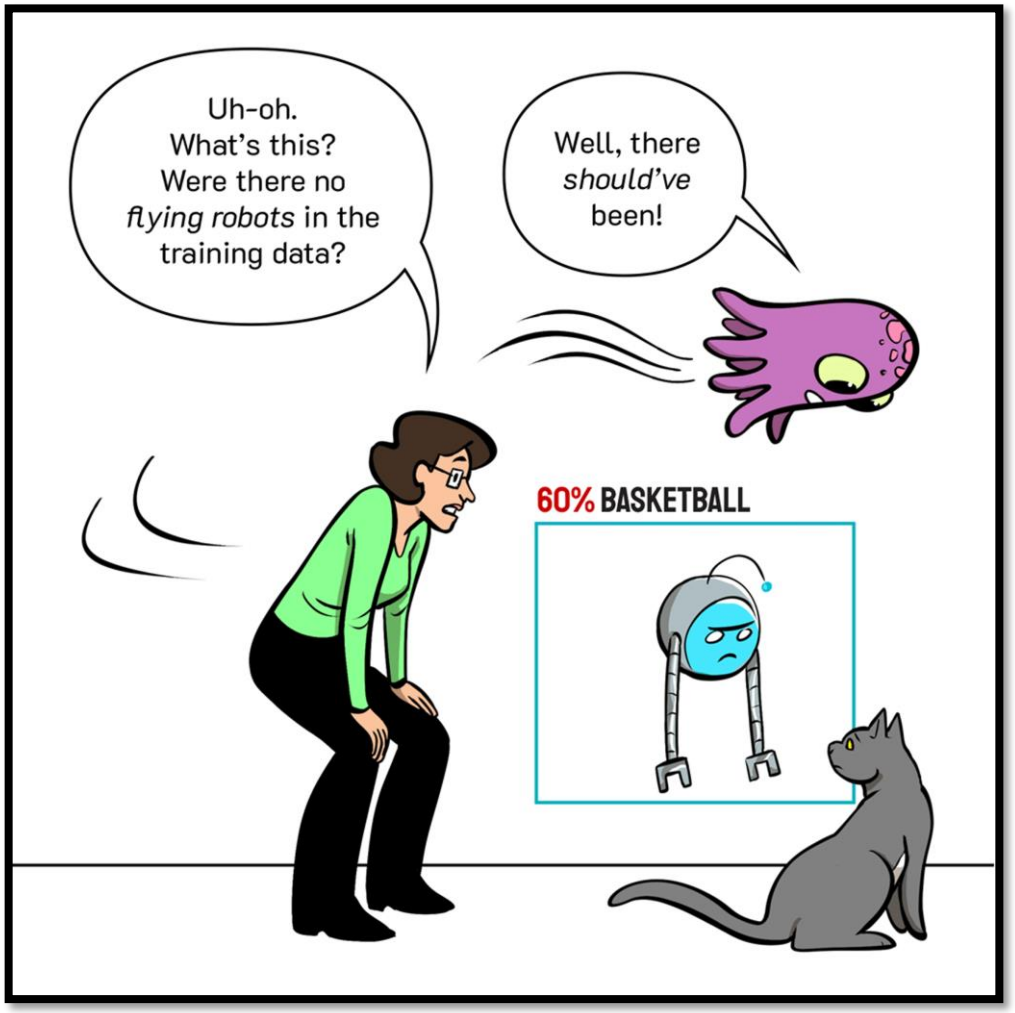

In their earliest stages, filters may detect no more than *edges* and *orientation*...*
*FUN FACT: your brain's visual cortex initiates detection this way too... in fact, it was an early inspiration for CNNs.
...LEARN MORE

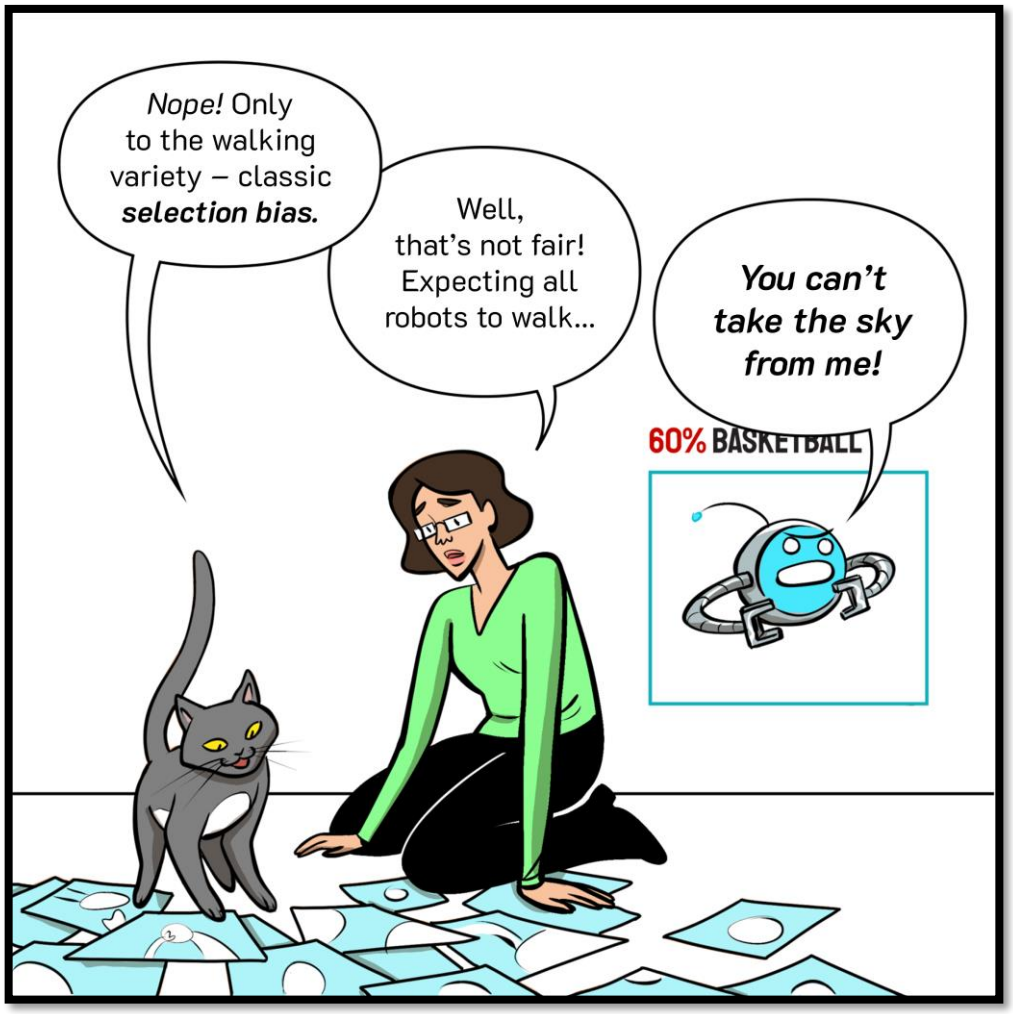

But with every subsequent layer,
composite features start to emerge from the noise.
See? It me!


So, it goes from *lines* and *angles*, through *whiskers*, *paws*, and *fur*, and then finally to "CATS" or "DOGS?"
Well... (*heh*) *yes*, *no*, and *yes!* We may never know exactly what features it sees in the *middle* —
— but whatever they are, it can *find* '*em* no matter where they're hiding!

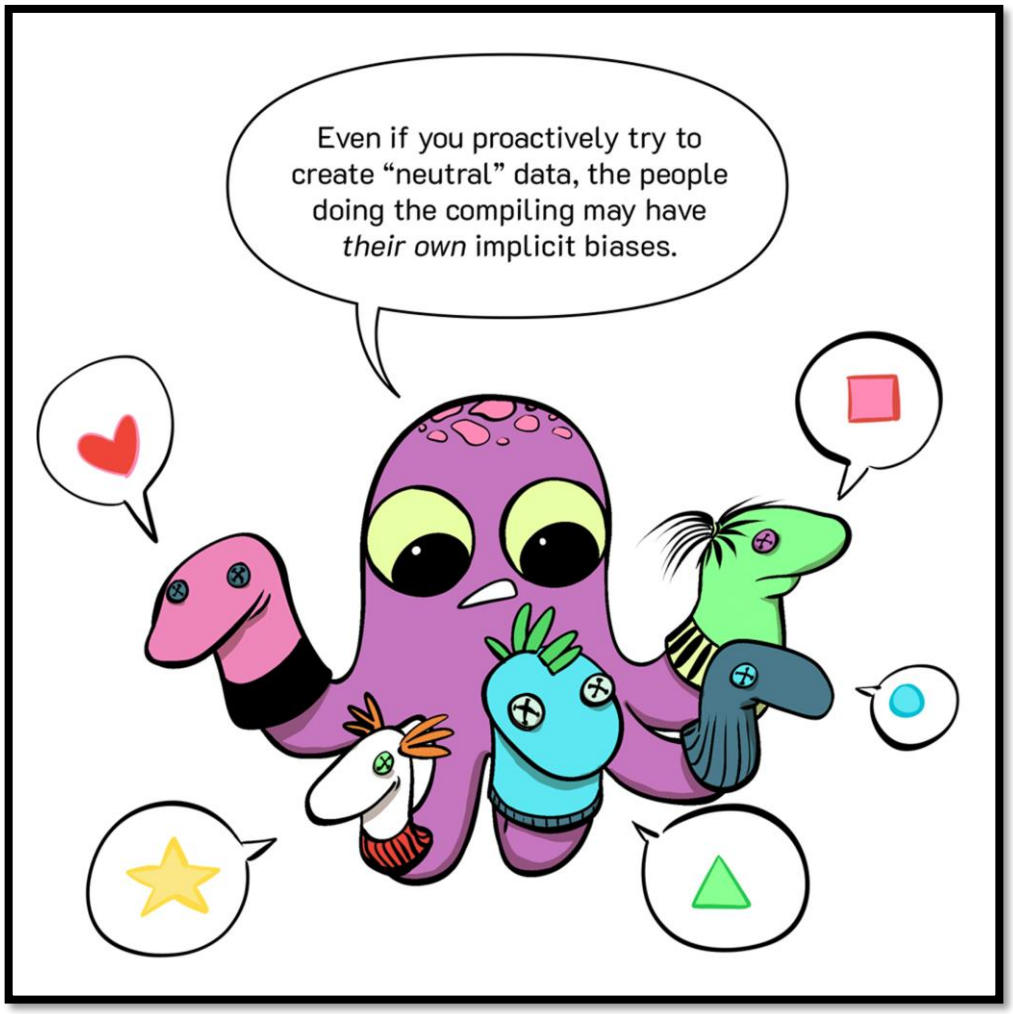

CNNs aren't fully connected like our *first* example. But the final output layer *is;* connecting to all the neurons on the previous layer.
CONVOLUTION
ACTIVATION
CLASSIFICATION
INPUT
POOLING
OUTPUT


This *classification layer* delivers the accurate answers we all—
94% OCTOPUS
98% HUMAN
Uh... gang?
60% BASKETBALL
89% CAT


Uh-oh. What's this? Were there no *flying robots* in the training data?
Well, there *should've* been!
60% BASKETBALL

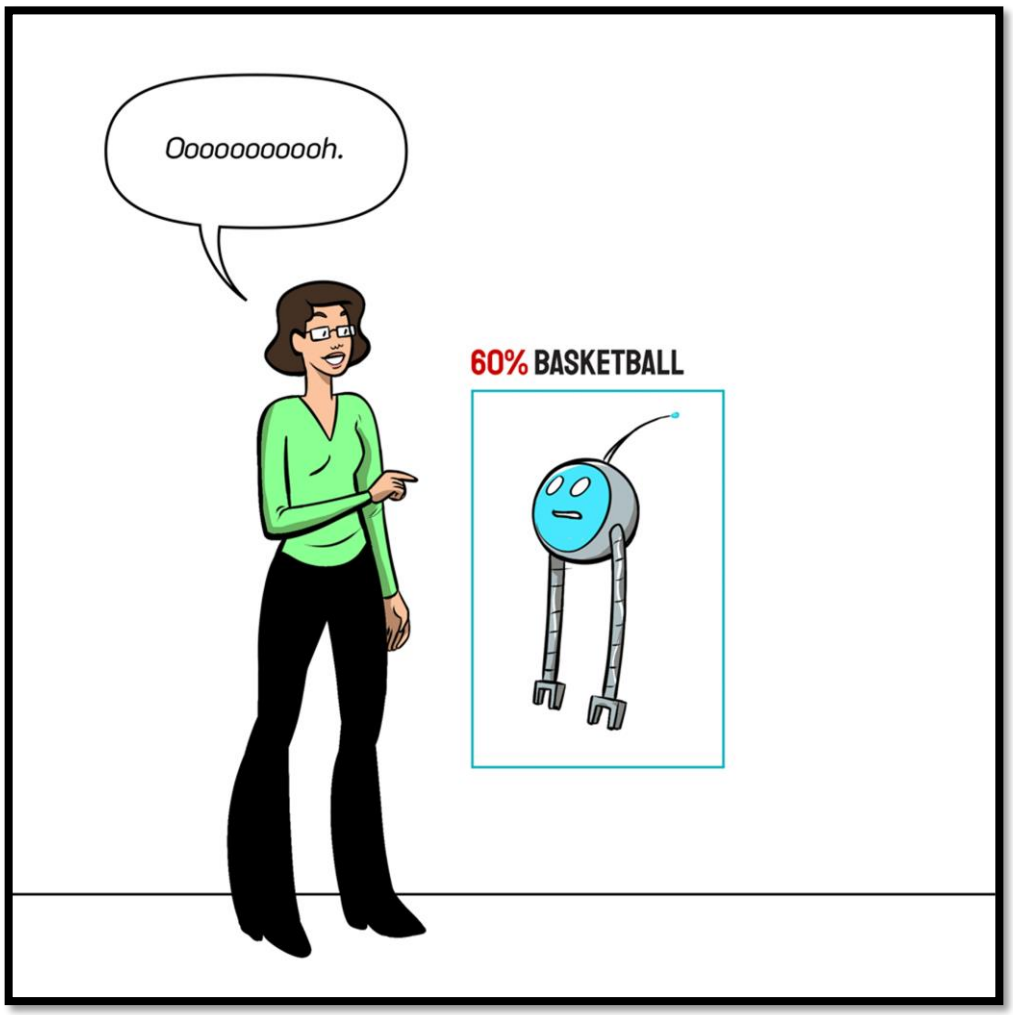

Nope! Only to the walking variety – classic selection bias.
Well, that's not fair! Expecting all robots to walk...
You can't take the sky from me!
60% BASKETBALL

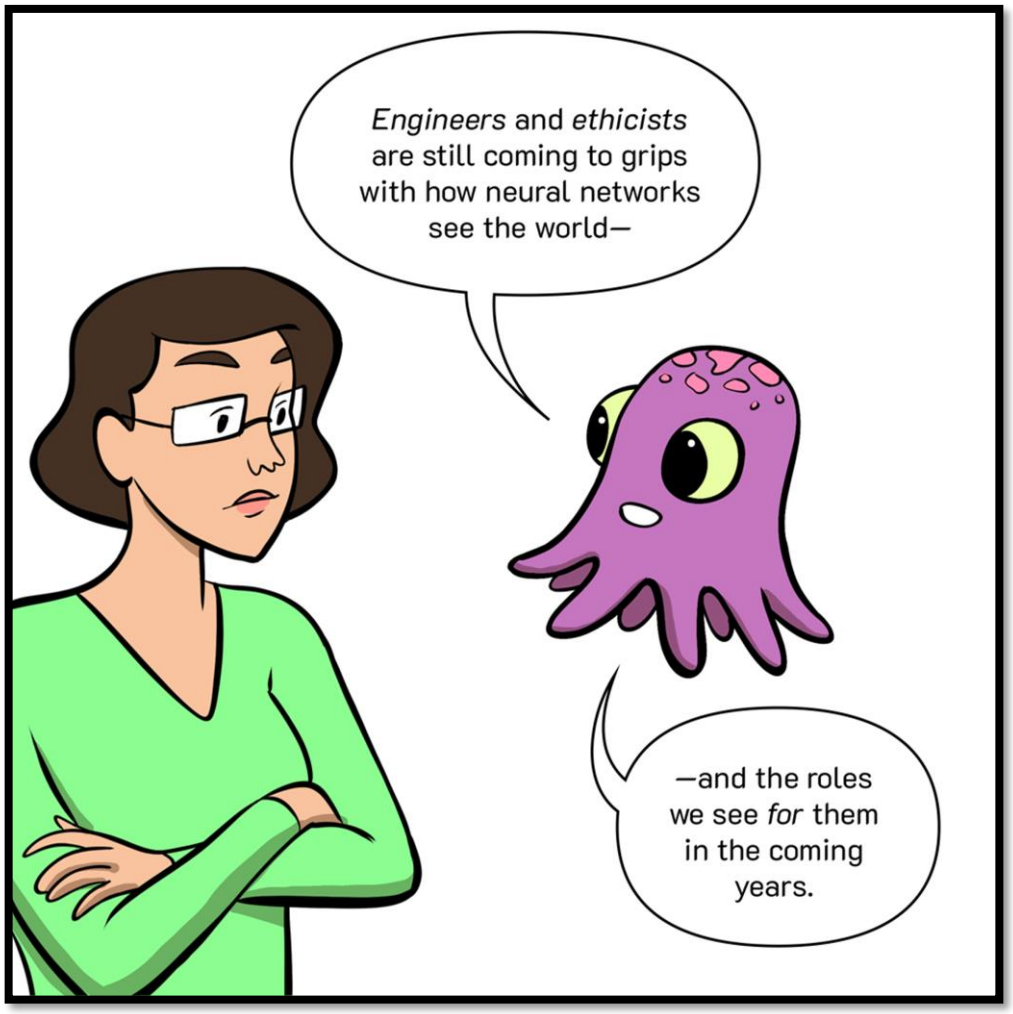

Guess I always thought of data as being unbiased. Just "numbers" and all...
Datasets are products of the time and place they're assembled.
60% BASKETBALL

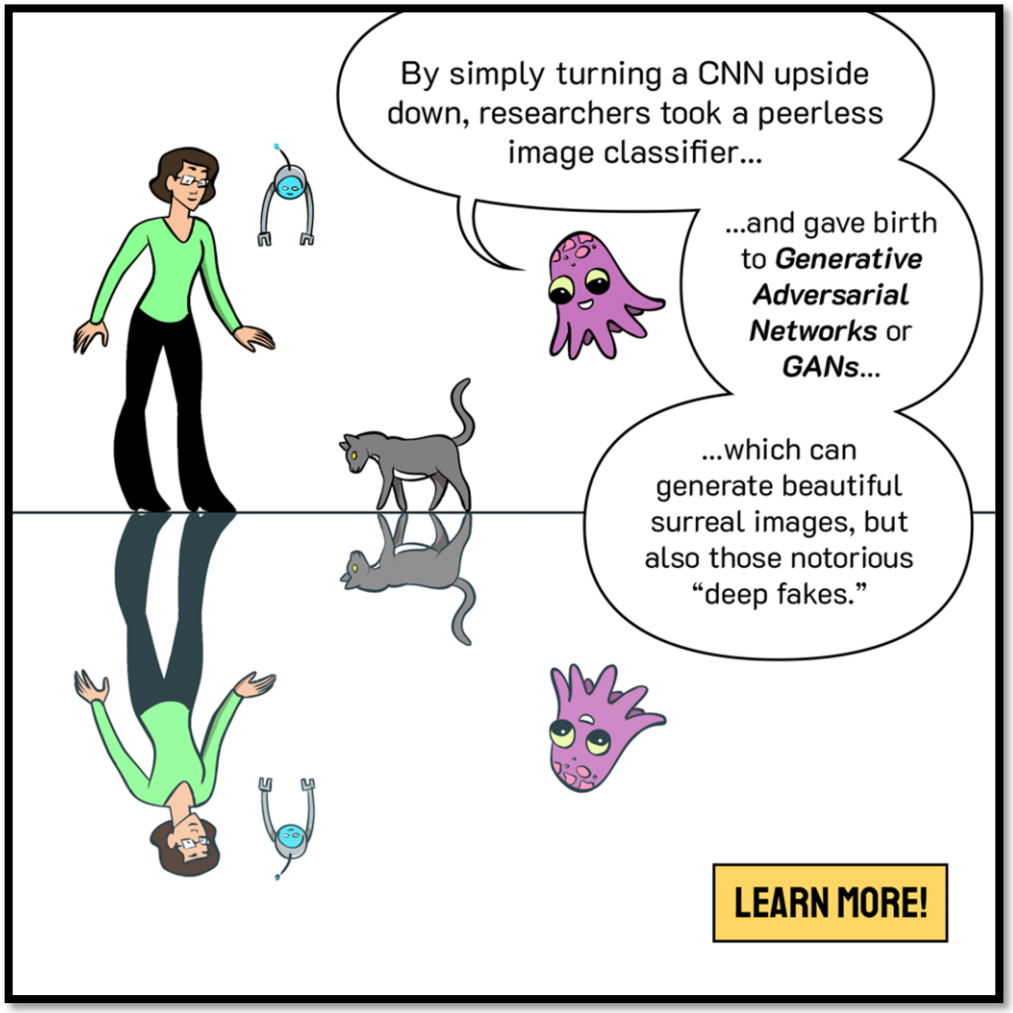

Even if you proactively try to create “neutral” data, the people doing the compiling may have *their own* implicit biases.


This is more complicated than I thought.
We might be dealing with something called ***latent bias,*** too.
Take a look at these basketball photos and see what they all have in common...
60% BASKETBALL

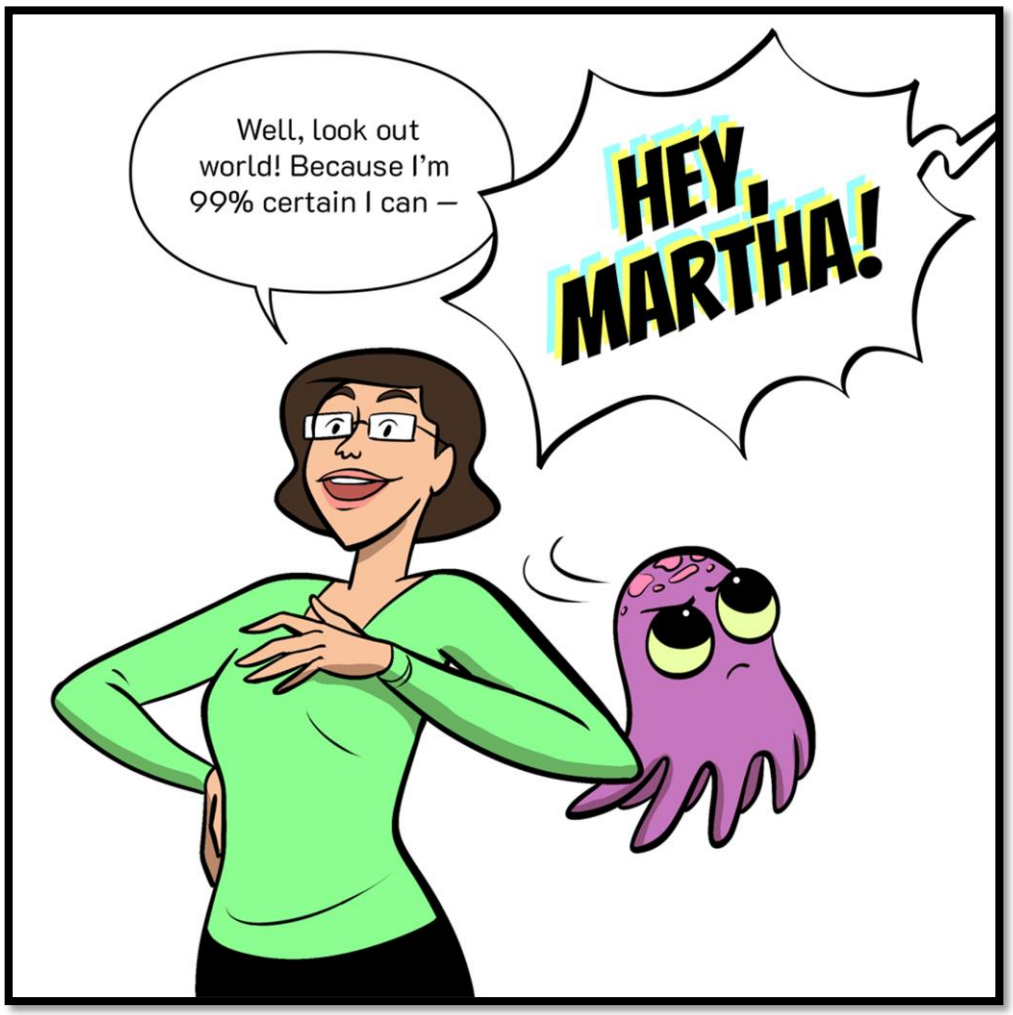

See anything *familiar?*
I dunno, just looks like a bunch of arms holding—


Oooooooooh.
60% BASKETBALL


Engineers and *ethicists* are still coming to grips with how neural networks see the world—
—and the roles we see *for* them in the coming years.

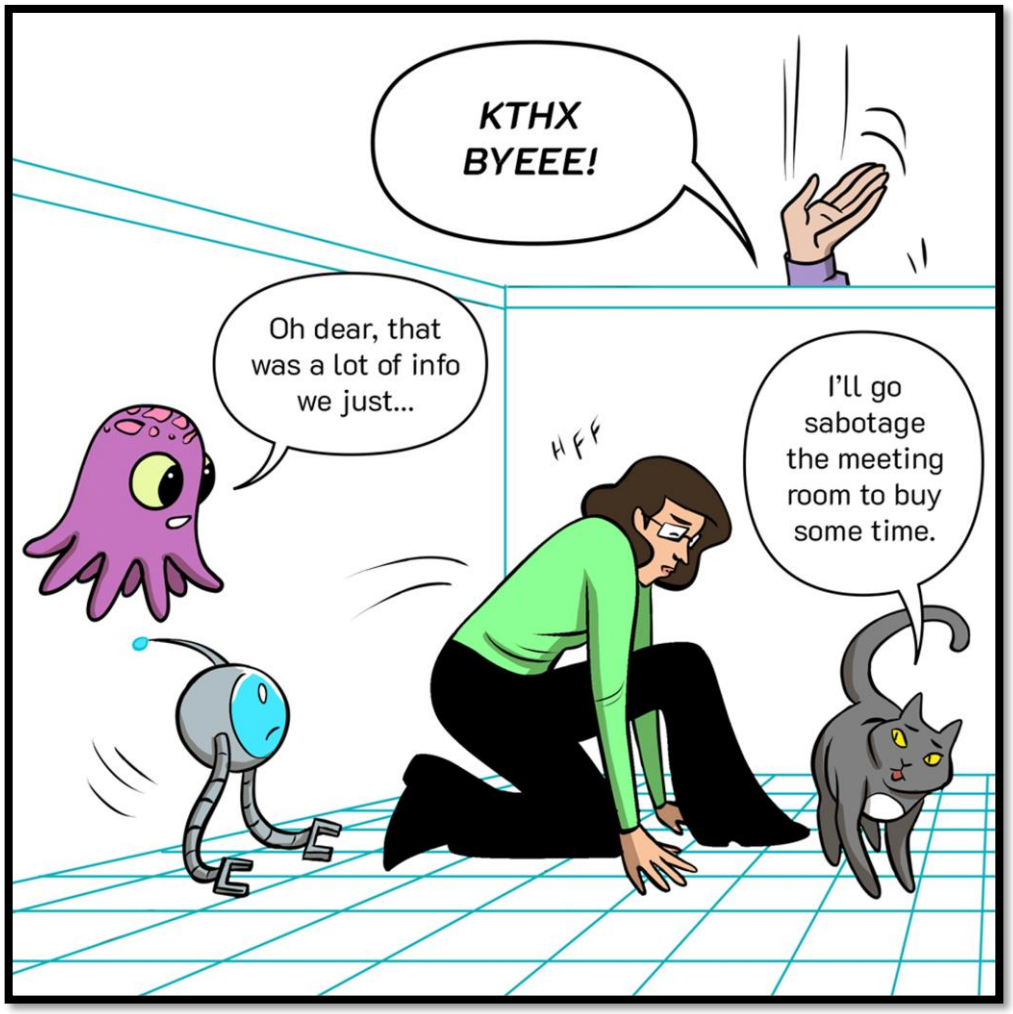

By simply turning a CNN upside down, researchers took a peerless image classifier...
...and gave birth to ***Generative Adversarial Networks*** or ***GANs***...
...which can generate beautiful surreal images, but also those notorious "deep fakes."
LEARN MORE!

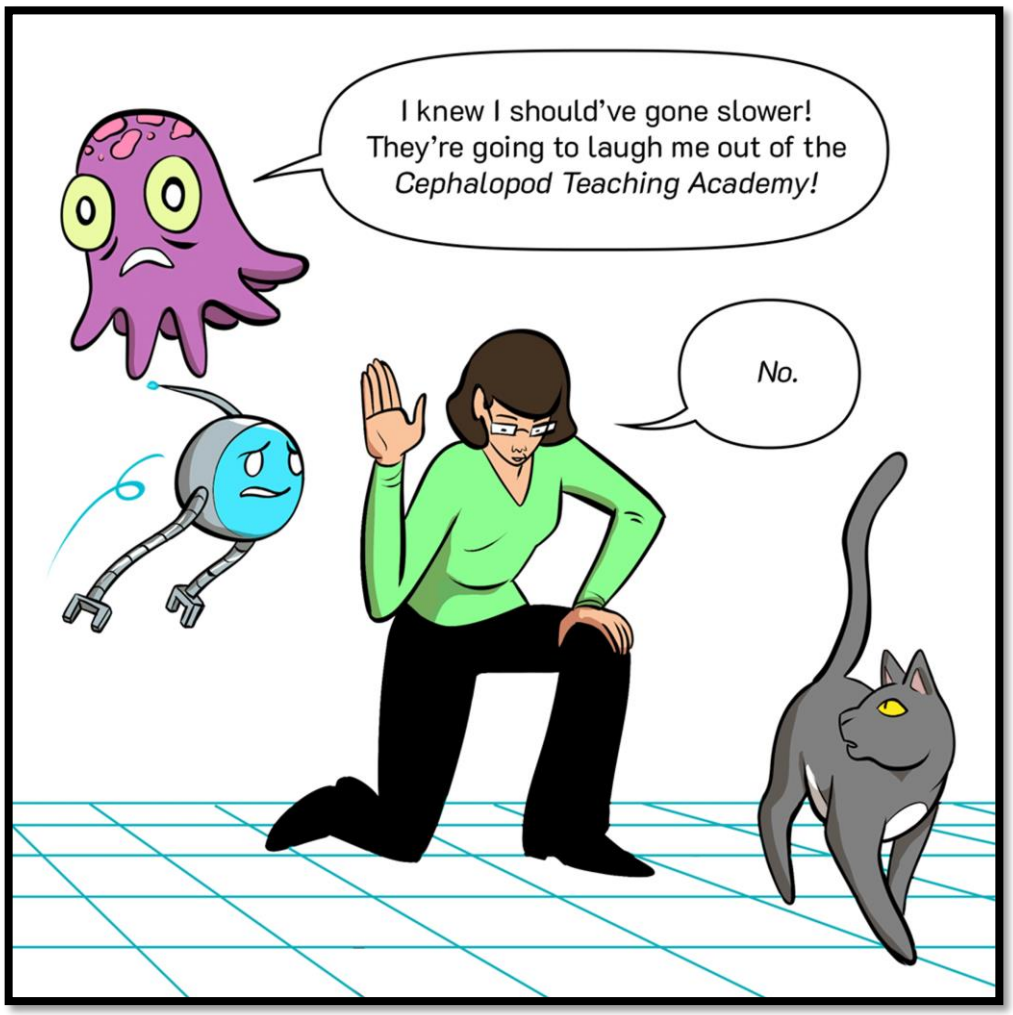

Yeah, I want to be part of the solution, not part of the problem... I just hope I can tell the *difference* when the time comes.
Hey, just learning how it all *works* is a great place to *start.*
Hear! Hear!


Well, look out world! Because I'm 99% certain I can —
HEY, MARTHA!


Ooh! Where am I? What's that? Who are *these* guys?
Uh, this is –
Actually, *I don't care!*
HISS!


I need you on deck in, like, *ten minutes* to present to the shareholders.
Is he always like this?
Yep.


KTHX BYEEE!
Oh dear, that was a lot of info we just...
HFF
I'll go sabotage the meeting room to buy some time.


I knew I should've gone slower! They're going to laugh me out of the *Cephalopod Teaching Academy!*
No.


I've got this.