This is Not a Brain

Allison Wusterbarth
Western Washington University

This Is Not a Brain

A Technical Conversation with Allison Wusterbarth
March 16, 2018

[Show Slide 1]

In the past few years, I've been hearing about a lot of movies -- there's a genre being revived where killer robots conquer the Earth or, more recently, artificial intelligence decides it doesn't want a human master anymore. These screenwriters are just reflecting on the world around them and it's true that that world is changing.

These days, you may hear the term machine learning, which is a subset of artificial intelligence; like those AI movies we were talking about, it too has had a resurgence in the past decade. It's quite literally everywhere; even this early in the morning, I'm willing to bet that you've interacted with some technology that uses machine learning algorithms. Maybe you transcribed a text using a voice-to-text feature or scrolled through your Facebook feed. If you've deposited a check with an app or used a rideshare app recently, you've seen machine learning at work (Narula, 2018). It's new, mysterious, and exciting -- which can also make it frightening, hence its use in so many films.

A lot of the time, in an effort to simplify fairly technical concepts, people in industry (myself included) will abstract concepts to a point where they seem alarmingly anthropomorphized. We talk about machines "learning" or "making decisions" in such a way that can seem like these models are out of our control. Let me be clear: No model, no matter how complex or well-performing, is thinking in the way that we do as humans. No matter what, it is not a brain.

What machine learning *is* is an exciting application of math that's been around for a long time -- some of the concepts date back fifty years or more. The difference now is that we have the computational power and the access to data to start applying these algorithms on a number of tasks. Machine learning is showing state-of-the-art performance in healthcare, self-driving vehicles, and cybersecurity, sometimes surpassing human performance (He, Zhang, Ren, & Sun, 2015). So really, there's a lot to be excited about! But people are still bound to be a little apprehensive. What we don't understand can be intimidating, *especially* when we don't understand *why* it's doing so well.

I don't pretend to be able to make you all machine learning specialists in 45 minutes, but I think understanding the basics is well within anyone's ability. I know it may be a little early for math, so I'll try to keep it at a minimum. Once we cover a high-level overview, we can circle back to those sentient robot concerns and take a deeper look at the possible weaknesses of machine learning.

[Show Slide 2]

At its core, machine learning is a umbrella term for algorithms that use input data to better model a certain task. Here, an algorithm can be defined as a series of steps -- think of a recipe. The three main elements of a machine learning algorithm are the input, the model, and the output. The input, often referred to as 'x', is the data provided to the model. Our model, represented by the 'h', is a black box of sorts -- this is where the computations happen. The output of the model is 'y', which is the model's prediction based off the input provided.

[Show Slide 3]

In a simple case, where we can think of the data as being represented by points on a graph, our model would be some sort of line that approximates the shape created by the points. Depending on the relationship in the data, the model could be a straight line --

[Show Slide 4]

or a curve.

[Show Slide 5]

Curves will have higher polynomial orders, such as x^2, x^3, etc.

Training the model to become better at predicting the correct output is essentially an optimization problem. But what exactly are we optimizing? To understand this, we'll have to get a general idea of a few metrics. The first term to understand is the loss function: it compares our model's prediction to the truth and reports how "wrong" our model was for a single prediction. The loss can be zero if the model was correct or a positive number if the model's prediction was further from the truth. Here's an example to think about it: if I was hoping the model would output a prediction of "apple", and it gives me "banana", these are similar enough that the loss would be smaller than if the model predicted something completely unrelated, like "tractor".

When the individual losses over the entire training set of input data are averaged, we get the empirical risk. Thinking back to the question "What do we want to optimize?", the answer is now that we want to minimize empirical risk. This makes sense, since empirical risk is an average of how wrong our model's predictions were: we want the model to be wrong as little as possible.

We've talked about empirical risk as the average loss over the training set, but what exactly is a training set?

[Show Slide 6]

There is a standard practice in machine learning to divide your input data into three sections, or sets. Each set of data can help the model improve in different ways. The largest set is called the training set or just "train set" -- as a rule of thumb, about 60% of the total data should be put into the train set. This is the dataset that the model will train on in order to determine the best parameters. Parameters are values passed to a function and can be tuned, or changed, to impact the function's output. Some parameter values work better for certain models, so it is necessary to tune them using the train set.

The next set is the development set, also called the "dev set", which contains about 20% of the total data. This set contains data that the model hasn't seen yet. The model's success on the dev set helps us choose which polynomial order works best for the model. If you think back to our example of the graph, the dev set helps us determine if a linear line or a higher polynomial curve will be the best fit. A sign of a good model is when results are strong for both the train and the dev sets.

Finally, we have test set, which contains the final 20% of the total data. It's important that the data in this set is not included in the train and dev sets, since we don't want the model to have already learned how to predict this data exactly. The purpose of the test set is to see how the trained model will react to data that it didn't train on, thus mimicking how the model will be used in the future. The ability of a model to still perform well on unseen data means the model is good at generalizing.

If results from the test set look good, the parameters and polynomial orders determined from the train and dev sets will be stored. Then, if a model is deployed for general use, any new data can be fed into the model to produce an output without training all over again. If the results from the test set could be better, we start the process of training the model over again with different parameters.

This process seems simple enough, but there are a few issues to be aware of.

[Show Slide 7]

The first is the problem of overfitting. Overfitting occurs when the training set has a very low prediction error rate because we have made the model very complex. Thinking of the graph example, this would be equivalent to making our model, represented by a curve, touch every single datapoint. Technically, this gives good results on the train set, but it loses sight of the overall flow of the data. When the test set is run, the model will be far worse at predicting the points because of how tailored it is to the training set.

On the other hand, there is underfitting. This occurs when the prediction error rates for both the test and the train set are too high. Generally, this is because the model is too simple and not expressive enough. Think of trying to use a line to approximate a curve -- it just doesn’t do as well. Basically, overfitting involves a model that is too complex and underfitting involves a model that is too simple. A good model will be in between these two options, resulting in a lower prediction error rate for both the test and train sets.

[Show Slide 8]

Now that you have the basics down, we can look at a real life example. My team’s research over the past two years has included a series of machine learning algorithms -- today, we’ll take a look at some of the most recent work.

For some background, my group’s project began as a result of an increase in student enrollment in STEM fields ("Our Reports", 2018). With higher enrollment, we were also seeing higher dropout rates. This may be caused by the fact that the traditional lecture-based teaching style doesn’t work for all students. Instructors looking to address this issue have started trying out new teaching methodologies -- for instance, does a certain subject lend itself best to more group discussion? How about additional silent work time? To see how effective these new methodologies are, teachers can compare how long they spend on certain activities to student grades and evaluations. Currently, a common way to “annotate” a class -- a method used to see what activity was happening when -- is to bring in a trained professional. This annotator may sit in on the class or listen to a recording of the class session. Either way, this process can be expensive to the school. There is also the issue of a time delay, especially when sending off a recording and waiting to hear back. By bringing the annotator to sit in on the class in order to get faster feedback, teachers risk disrupting their students; people who know they’re being watched may act differently and skew the results.

My research group has teamed up with researchers at San Francisco State University to explore methods for cheaper, faster, and less disruptive classroom annotation; the goal is produce a tool where teachers can input a recording of their class and receive a breakdown of their class time automatically. We’ve implemented several machine learning models over the course of this project, but let’s take a look at the model I’ve done the most work with: our Deep Neural Network.

[Show Slide 9]

Neural networks are a subset of machine learning that are flexible and powerful. They’ve had many names over the years, such as Artificial Neural Networks or (my favorite) multi-layer perceptrons. If we look at a diagram of a neural network, you can see the same basic structure that we talked about at the beginning. There’s still an input layer ‘x’, an ‘h’ (this time called a “hidden layer”), and an output ‘y’. The different here is that they are vectors, which for the purpose of this conversation can be interpreted as a way to represent data.

You can think of information moving through this neural network, where it undergoes a transformation between each layer. There are weight matrices and weight vectors that can be multiplied and added to the data before it is processed by activation functions. These functions produce variations of the data at different points in the model. For instance, ‘h’ is what we call a “learned representation” because it’s essentially a modified version of the input data. What we often think of as “learning” in machines is really just a lot of linear algebra.

But hey, neural networks may be cool, but I promised you *Deep* Neural Networks -- so let’s take a look at those.

[Show Slide 10]

A Deep Neural Network, or DNN, looks a lot like the neural network we just saw. The “deep” part comes from having multiple hidden layers. The more hidden layers a model has, the “deeper” it’s considered. For the DNN used in my research, we have five hidden layers, so our model would look something more like this. Different problems require different model structures, so some projects may work better with fewer hidden layers.

In the specific case of my research, our input is a numerical representation of an audio wave from the recorded class. We also input the true label associated with that clip of audio. For example, if the sound clip is of a person lecturing, we include the “lecture” label. This helps the model to correlate patterns in that audio representation to that label. To keep things simple, the goal is to classify classroom audio into four activities, or labels: lecture, discussion, silent work, and other. The “lecture” label currently covers any time a single person is talking; this can be a professor lecturing, a question and answer session, or a video being played. Similarly, the “discussion” label includes times when there are multiple voices, such as during group work or transitioning between activities. “Silence” almost exclusively represents silent, individual work time and the “other” label catches any noises that don’t fit into the other categories.

After moving through the hidden layers, the output of our model is a predicted label associated with the audio representation provided as input. By adding up how many audio snippets are classified as each label, the system can tell the teacher how much of their class was spent on lecture, discussion, silent work, and other.

Now, I know what you’re thinking: you were hoping my research would be more like those rebellious, sentient AIs we mentioned at the start. The basic fact that many of the extremely “intelligent” systems in use today are using similar deep learning architectures to what my team is using to annotate classroom audio. Remember when I said they were versatile?

All jokes aside, walking through a DNN like we just did will hopefully put you a little more at ease. Machine learning models are not actually learning in the way humans do -- sure, they're permuting data, but they're doing so using same techniques as sophomores taking linear algebra. The only difference is that machines are much faster. John Pfeiffer said it best: "Man is a slow, sloppy, and brilliant thinker; computers are fast, accurate, and stupid" ("A quote by John Pfeiffer"). The computations they do are nowhere close to thoughts or feelings -- they are, quite simply, just doing the math.

So, we're safe, right? No issues or ethical concerns? What if I told you that the machines weren't the problem? What if we need to be worried about ourselves?

[Show Slide 11]

When someone brings up ethics in terms of computer science, it usually boils down to three things. The first isn't that specific to the field: it's the case of doing something unethical, like faking safety tests. The second is what most people think of: what should we be allowed to do with technology? Self-driving cars fall in this category, I should mention. No matter what party I end up at, everyone who hears that I do machine learning has some concerns about self-driving cars. But it also goes beyond that into facial recognition and surveillance. As what problems we can apply machine learning to expands, so does the relevance of the question "Should we?". Quite honestly, that topic could be another full presentation. What I'd like to talk more about today is the third branch, because it's the most subtle of the three. What is the ethical responsibility of a creator to make their creation accessible to everyone?

In recent years, there's been a notable rise in people pointing out serious flaws in popular machine learning algorithms. Google has been a company which has been called out several times, likely because of their prominence in the machine learning field; their research chapter, called DeepMind, is well known for some of the most cutting edge machine learning technology.

Google offers a service called Google Photos (you may have heard of it) which helps people store and organize their pictures. Since classification of images is a task well-suited to machine learning, a new algorithm was introduced to automatically tag and sort people's pictures as they were uploaded. In theory, this is a handy addition. You can see all of your selfies or all of your nature shots in one place. But in 2015, Google Photos classified a photo of an African-American woman as a gorilla (Guynn, 2015). Keep in mind, this technology had already been trained, tested, and distributed to the public -- all the while deemed fit for use.

In another case, Google Translate came under fire. For anyone who's tried to translate anything over the course of the years, you'll know that this Google product has improved leaps and bounds in the past few years -- this is largely due to machine learning advances. But as early as November of last year, Google Translate was reflecting some frustrating biases. The program was translating the Turkish gender neutral pronoun "o" into "he" or "she" based off the context of the sentence (Tousignant, 2017). The results were statements that aligned with traditional gender roles: "He is an engineer" or "She is a nanny".

These two cases are problematic for different reasons. The Google photos incident is quite literally dehumanizing and taps into a long history of systemic racism in both in America and other parts of the world. Meanwhile, while the Google Translate scenario isn't directly impacting a specific person, it affirms harmful gender stereotypes about what different genders can or "should" do. Both cases highlight a carelessness that may start showing up more and more in machine learning.

Not long ago, scientists had a bad habit of defending algorithms as perfectly impartial -- they were, after all, just a combination of math and facts. But really, all algorithms start with humans, who have all sorts of flaws. Machine learning is no exception; it, too, is just another type of algorithm. No one's hardcoding racism or sexism into their neural networks, but their bias can still affect a model. You may be asking how this is the case, since the explanations we've gone over have shown most "learning" done by models is just math -- there's no human hand in it. But what part of the process *do* developers and scientists have their hands all over?

Data.

Back before a single computation starts running, someone has to select what data to feed the model. What's included in that dataset -- and possibly more importantly, what's *not* included -- is up to a human decision. I want to be fair to the scientists working on these problems: data can be hard to come by. It can be expensive to obtain and hard to refine down to a representation useful to a model. But that's not an excuse to ignore significant populations.

For instance, based on how the Google Photo situation played out, I can make some educated guesses about their dataset. Their goal was to classify photographs, so I imagine they put together a dataset of as many types of photos as they could think of: people, animals, nature, things... you get the idea. They likely split their dataset into some sort of train, dev, and test structure (similar to what we talked about earlier) and proceeded to train their model. They evaluated on the dev and test sets until they were satisfied that the model could correctly classify data it hadn't seen before. Then this trained model, after a series of checks, was released. But, what about their photos? Something tells me that there were likely not many African-American people pictured in this dataset. When they ran their model on the test set, there were likely very few photos of people with darker skin. If they were included, then the scientists failed to check which images were being misclassified. Either way, their model didn't have enough data from this population in the train set to be able to recognize African American faces as just that -- human faces. If there had been more photos of people with dark skin in their test set, they likely could have caught this mistake. Instead, they had enough pictures of gorillas that their model had to fill in the blanks. And, well... we saw how that worked out.

Data is also the crux of the Google Translate issue. Likely, Google's model trained on sentences often used by people or seen in print. They also have the benefit of collecting more training data anytime a person uses their system. Likely, this data led the model to associate certain professions with certain pronouns. To a model, it's simple: if nine times out of ten a profession is attributed to a man, then there must be a correlation.

When technology permeates society to the level it has today, it becomes critical that this tech be usable by everyone. There's a long history of the forefront of innovation ignoring minoritized groups. Healthcare is notorious for this, as well as inventions as simple as color TV. Originally, both color television and Kodak photos optimized their equipment to best show off white skin tones, since white models were the standard for color correction (Gross, 2015). Human bias in machine learning is the next generation of this.

Virtually no project is immune to this. My team's project is a classic example of how difficult it can be to obtain data. The classroom recordings we have were shared with us by our collaborators at San Francisco State University, mainly because this data is difficult to collect; class sessions are protected by federal privacy laws like FERPA. Regardless, if our model is unable to function correctly in virtually *all* environments, it isn't at an acceptable level of generalizability. Currently, the way we address this is by using data from two colleges with recordings from seven instructors. Outside of the traditional train, dev, and test sets, our team added an additional test set. This extra dataset included instructors whose audio the model had not yet encountered. By checking the model's performance on both tests sets, we could be more confident that it would generalize to more classroom environments.

On a scale much larger than a university research team, there are even more options to address this issue. Companies like Google have an impressive reach when it comes to obtaining data -- they just need to be conscientious when selecting this data. Here are two ways that seem obvious to me:

The first is to bring different perspectives onto your team. In recent years, we've seen a big push for women in computer science, which is honestly fantastic. But we can't stop there. In this industry, we need more people of color, people with disabilities, people of different gender identities and sexual orientations, and so many more groups. Intersectionality is critical; if you're trying to make something that helps a wide range of people, it's important to work with a wide range of people.

The other piece is that bringing these people onto your team doesn't relieve you of responsibility. It's important to educate ourselves about our own biases and privileges so that we can start holding ourselves accountable. We are just as responsible for our subconscious lapses as we are for our conscious choices.

We're on the verge of an exciting future right now. Most of us are just in college, yet in our lifetimes we've seen the transition from dial-up to the internet as it is today. But as we build tomorrow's tools, it’s our responsibility not to bring along the baggage of today. Who knows where machine learning will take us in the next fifty years, but let’s make sure that we stick together. Technology should be universal in who it can help and we can’t be afraid to speak up when it fails to do so.

I know many of you aren’t computer scientists. This problem can seem out of your hands in many ways. But you can absolutely still help. First and foremost, start examining these biases in your own life. Do some research. What are your privileges? How do they create blind spots for you? Then reach out and have these conversations with the people around you. You never know who in the world will make what decisions; someone you talk to may grow up to, or already be, someone who works on these problems.

The other piece you can contribute is to encourage those around you to tackle these issues within machine learning and computer science as a whole. This doesn’t have to be limited to children who may grow up to scientists -- more and more, people later in life are finding ways to incorporate computer science into their work. I don’t think it’s too wild to say that in twenty years most professions will involve basic coding literacy. And I’ve become very suspicious of people who say that not everyone can code; I don’t think coding comes naturally to everyone, but I think to limit this valuable skill set to a very specific group of privileged people creates the sorts of problems we’ve seen here today.

So encourage everyone and anyone to try their hand. If they doubt you, tell them the story of a woman who started looking at colleges in pursuit of an English degree -- tell them about how she’d never coded in her life, but four years later was completing machine learning research before graduating. Tell them the stories of how women, people of color, queer people, and so many other groups founded the basis of computer science today. And do more than believe in them! Donate to organizations that are bringing computer science to schools like mine, who didn’t have the resources to include it in the curriculum without help. Donate to scholarships for different groups in STEM. Don’t give up because it isn’t easy.

I know this presentation is about technology. But to try and separate technology from humanity isn’t feasible. Machine learning has a lot to offer us in terms of innovation, but we need the human element. Machines aren’t brains -- they need *us* to make them the best they can be: high-performing, yes, but also accessible and universal. And that, to me, sounds like a job for a human brain.

[Show Slide 12]

Thank you for being a fantastic audience this morning. Before we start questions, I would like to thank Dr. Brian Hutchinson, my advisor for this project, and the other members of my research team. Their hard work pioneers new advances and it’s been my privilege to work with this group over the past two years. And, of course, a final shout out to the research team at San Francisco State University whose collaboration has made my team’s project possible.

Thank you.

Works Cited

A quote by John Pfeiffer. (n.d.). Retrieved March 3, 2018, from https://www.goodreads.com/quotes/488215-man-is-a-slow-sloppy-and-brilliant-thinker-computers-are

Gross, B. (2015, June 28). Living Test Patterns: The Models Who Calibrated Color TV. Retrieved March 12, 2018, from https://www.theatlantic.com/technology/archive/2015/06/miss-color-tv/396266/

Guynn, J. (2015, July 01). Google Photos labeled black people 'gorillas'. Retrieved March 14, 2018, from https://www.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/

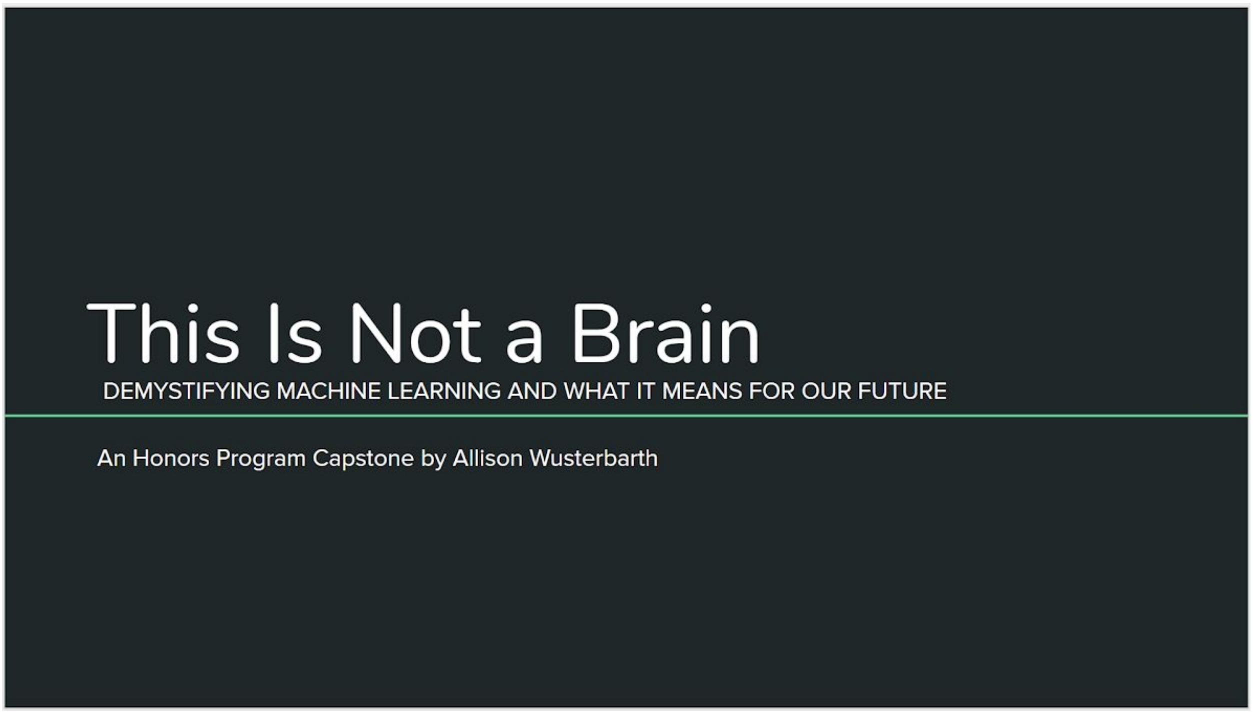
He, K., Zhang, X., Ren, S., & Sun, J. (2015, February 6). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[Scholarly project]. In Cornell University Library. Retrieved March 13, 2018, from https://arxiv.org/abs/1502.01852

Narula, G. (2018, March 02). Everyday Examples of Artificial Intelligence and Machine Learning. Retrieved March 15, 2018, from https://www.techemergence.com/everyday-examples-of-ai/

Our Reports. (2018, March 07). Retrieved March 14, 2018, from https://nscresearchcenter.org/ourreports/

Tousignant, L. (2017, December 01). Google Translate's algorithm has a gender bias. Retrieved March 15, 2018, from https://nypost.com/2017/11/30/google-translates-algorithm-has-a-gender-bias/

Appendix: Supplemental Slides

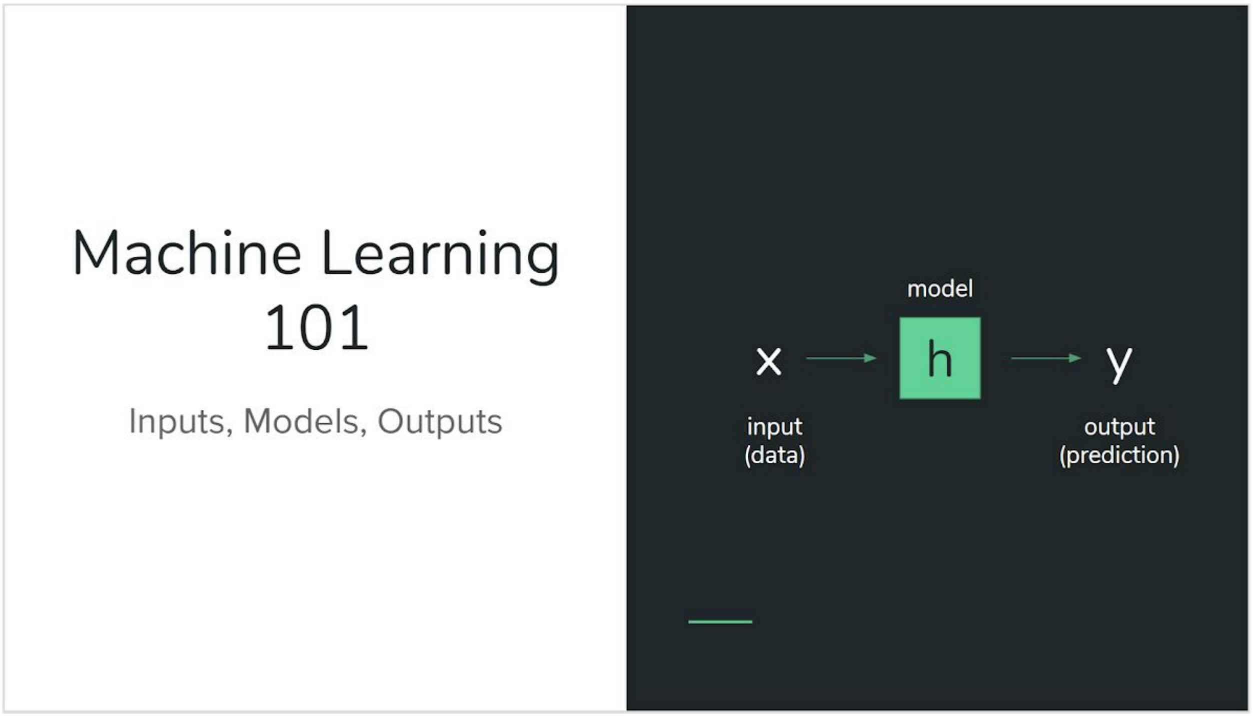


This Is Not a Brain

DEMYSTIFYING MACHINE LEARNING AND WHAT IT MEANS FOR OUR FUTURE

An Honors Program Capstone by Allison Wusterbarth

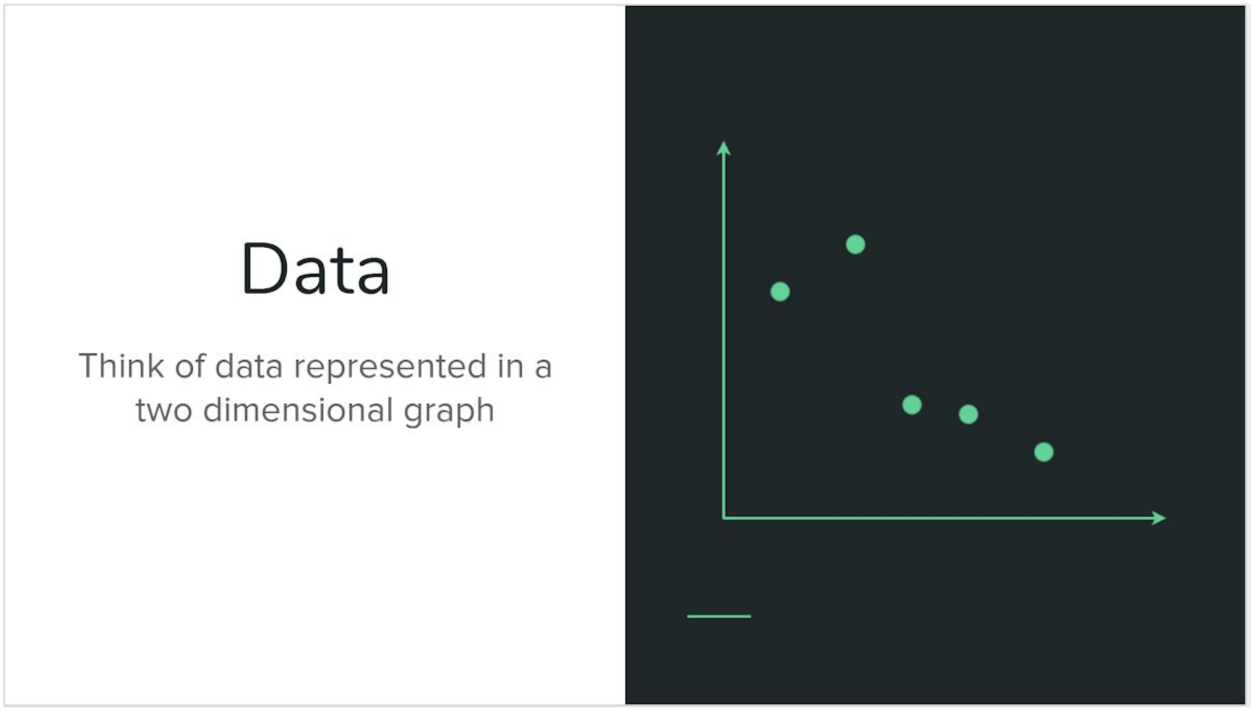
Slide 1



Machine Learning 101

Inputs, Models, Outputs

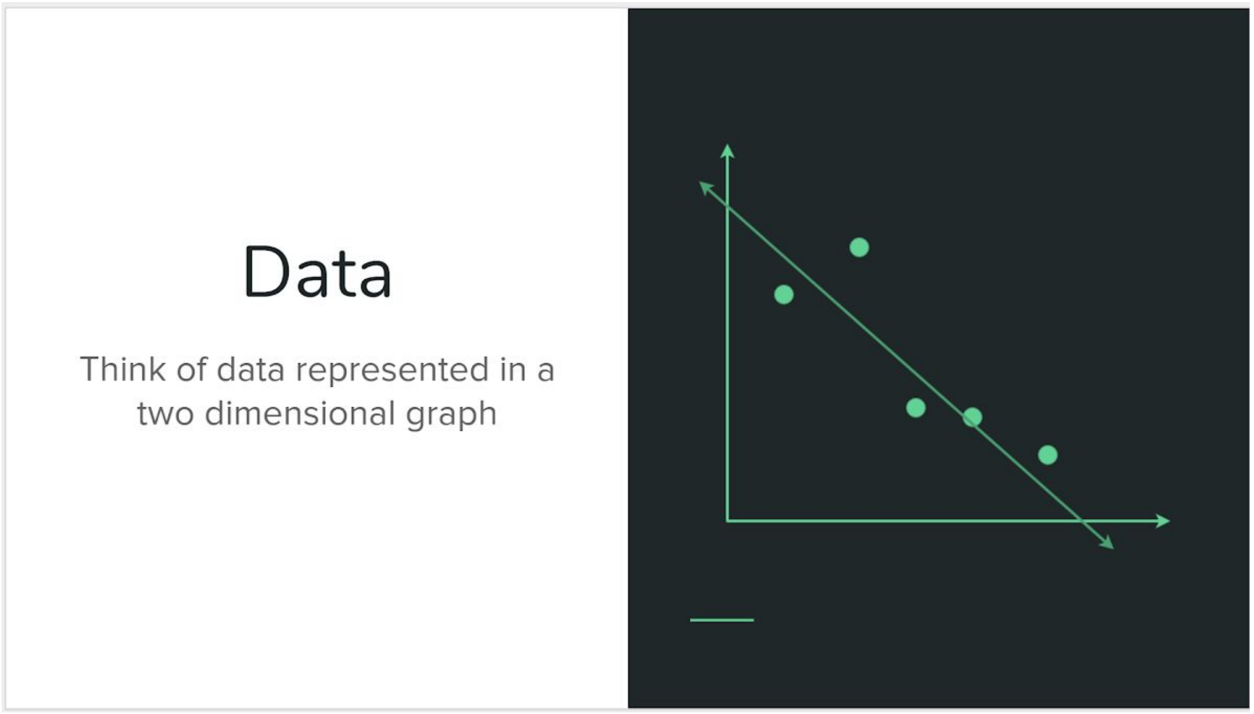
Slide 2



Data

Think of data represented in a two dimensional graph

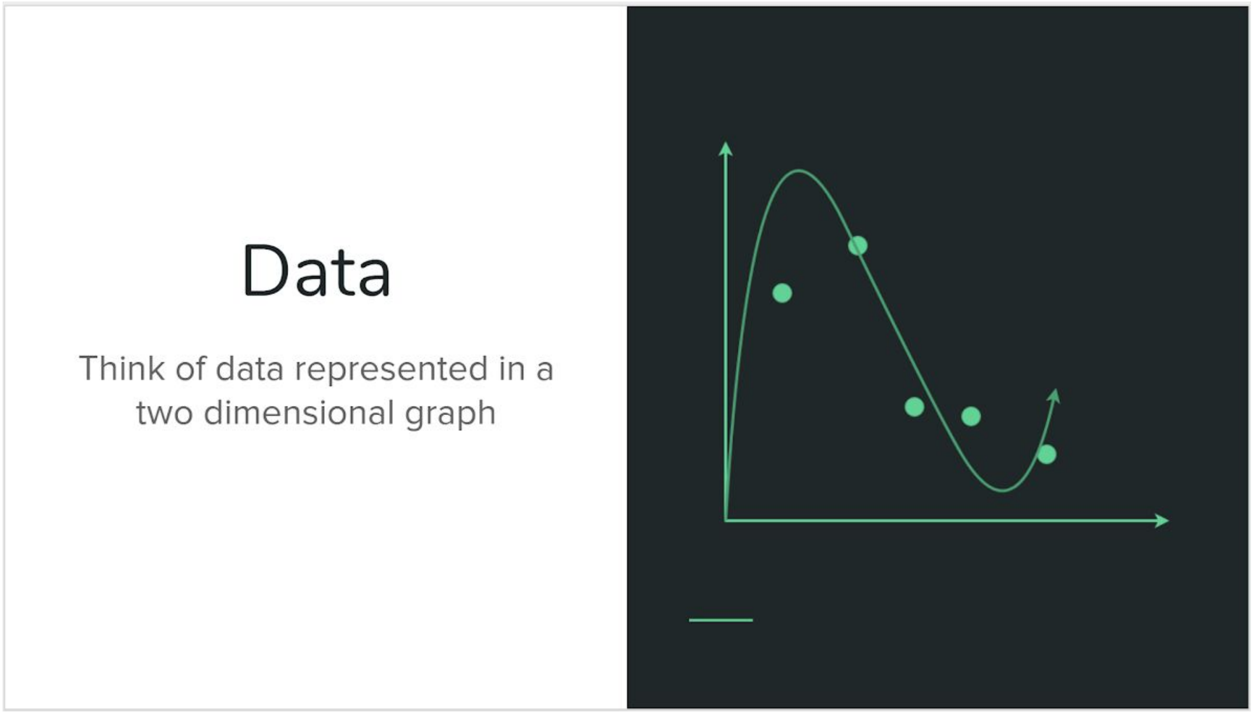
Slide 3



Data

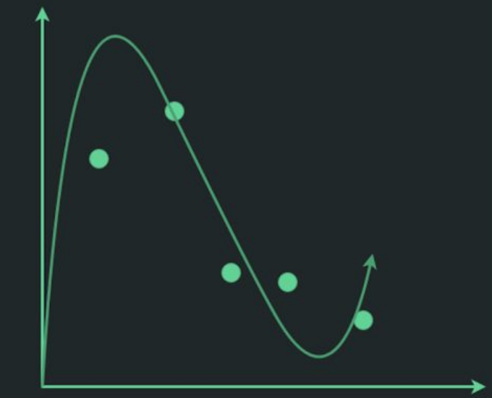
Think of data represented in a two dimensional graph

Slide 4

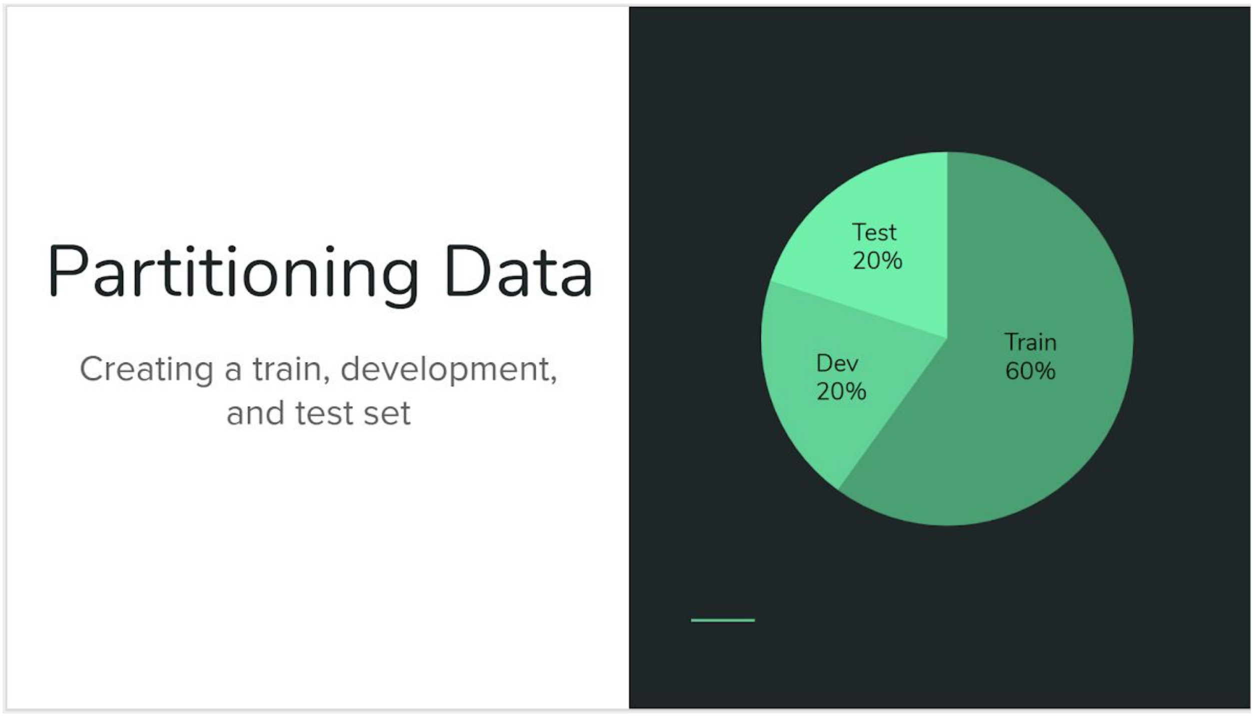


Data

Think of data represented in a two dimensional graph

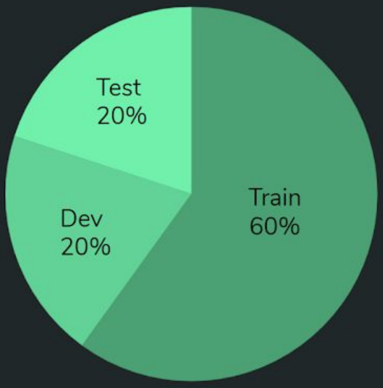


Slide 5



Partitioning Data

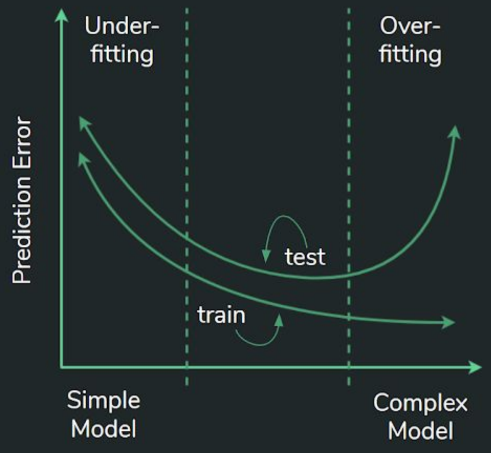
Creating a train, development, and test set



Slide 6

Training Risks

Overfitting vs. Underfitting



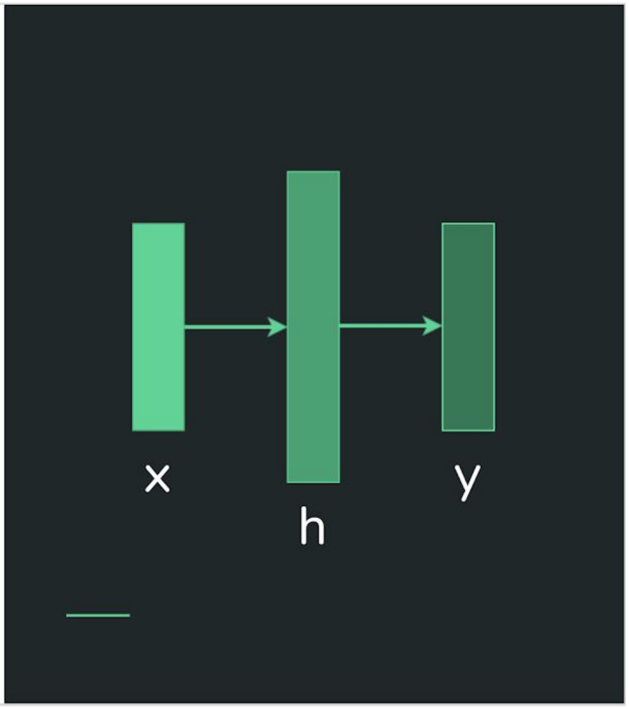
Slide 7

Research

Slide 8

Neural Networks

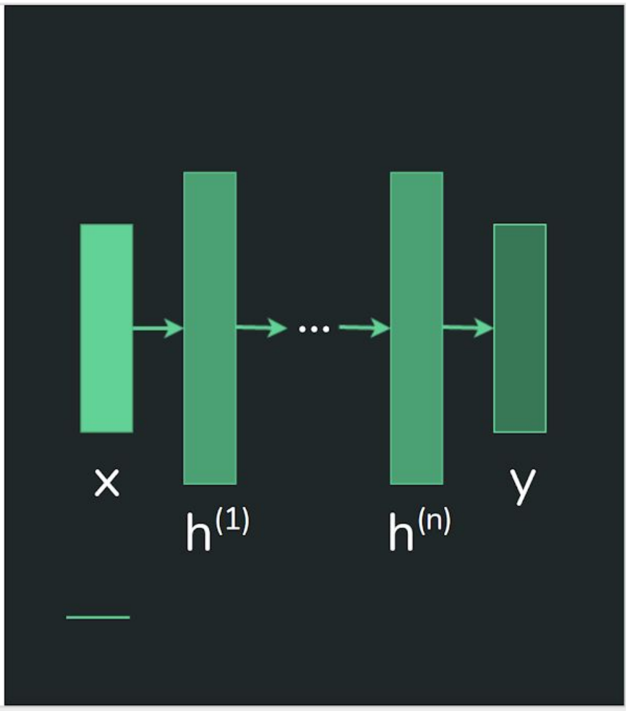
Still not a brain



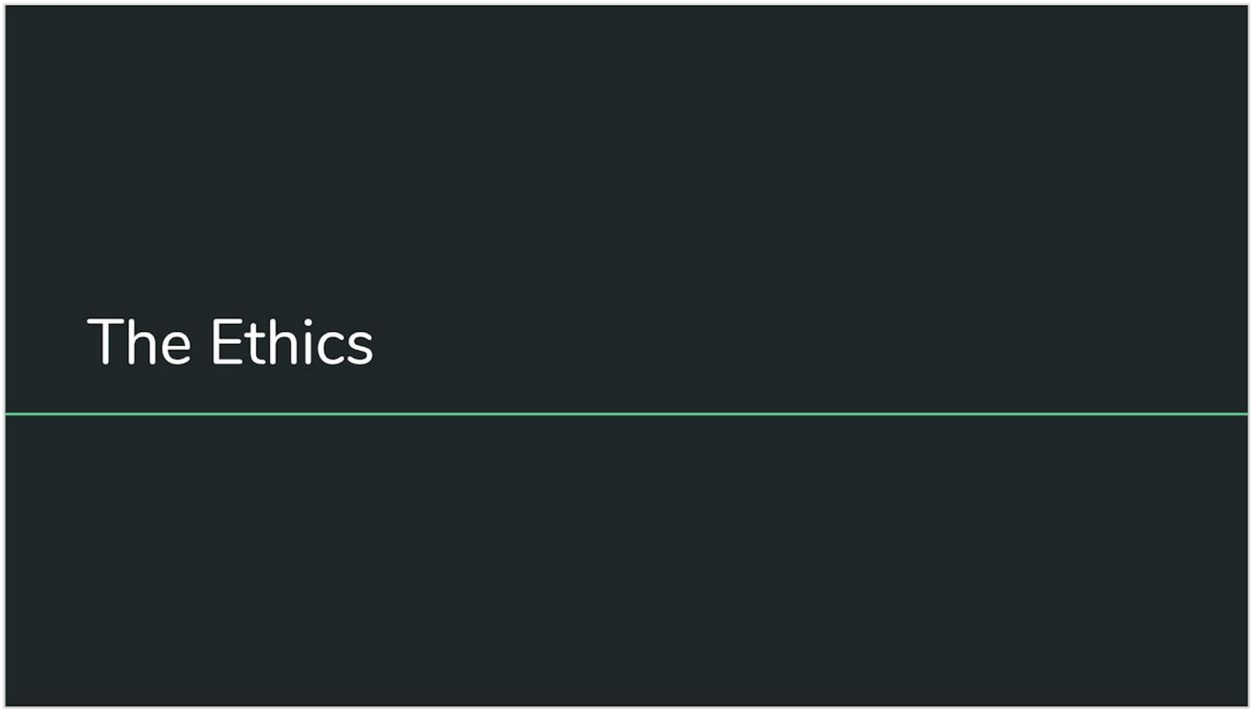
Slide 9

Deep Neural Networks

A type of neural network

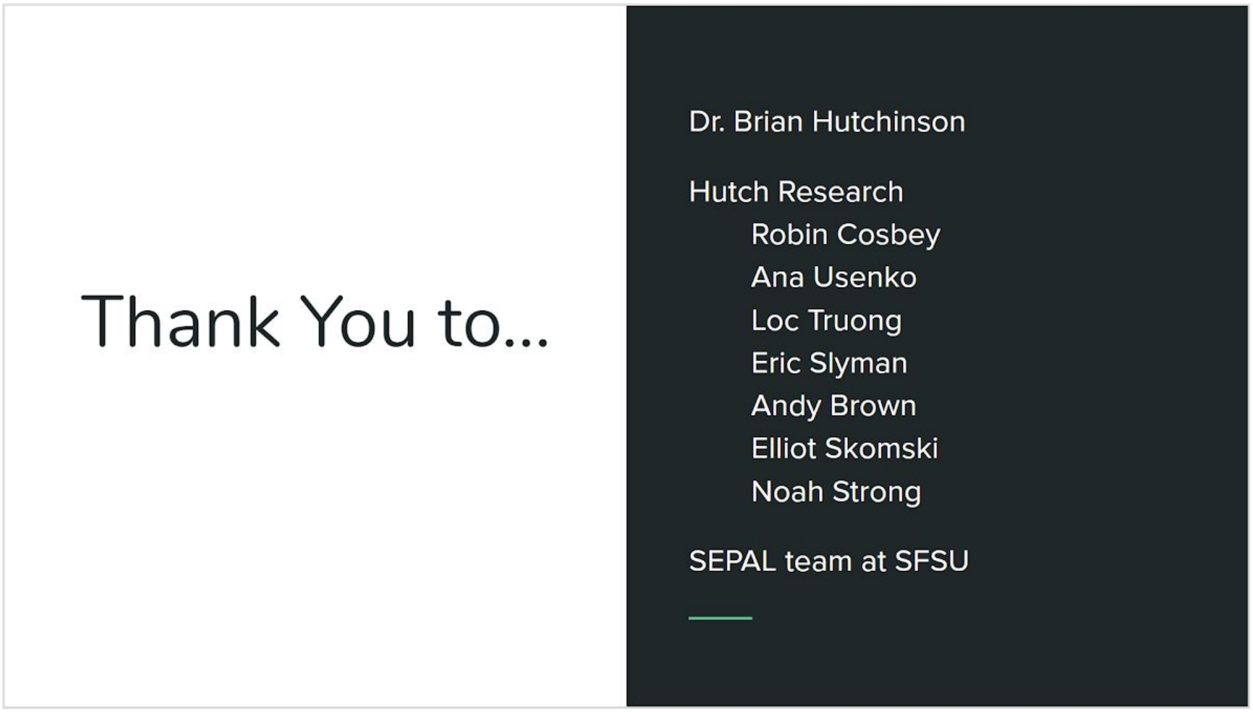


Slide 10



The Ethics

Slide 11



Thank You to...

Dr. Brian Hutchinson

Hutch Research
- Robin Cosbey
- Ana Usenko
- Loc Truong
- Eric Slyman
- Andy Brown
- Elliot Skomski
- Noah Strong

SEPAL team at SFSU

Slide 12