

# HackCU: Intro to AI

Matt Strong

# Outline

- Self-Introduction
- Intro and Motivation to AI
- History and What is Being Done Now
- Basic AI techniques
- Real World Example

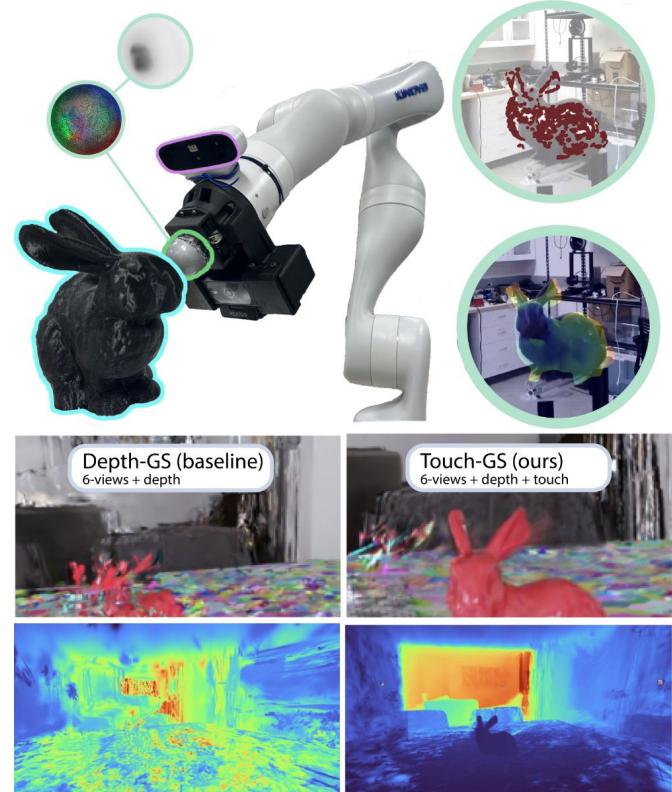
# Self Introduction

- Born and Raised in Colorado
- Attending CU Boulder for all 4 years of undergrad
- 2X SWE Intern at Microsoft
- Worked for a couple years at Microsoft
- Now: **Stanford CS PhD**



# My Research Blurb

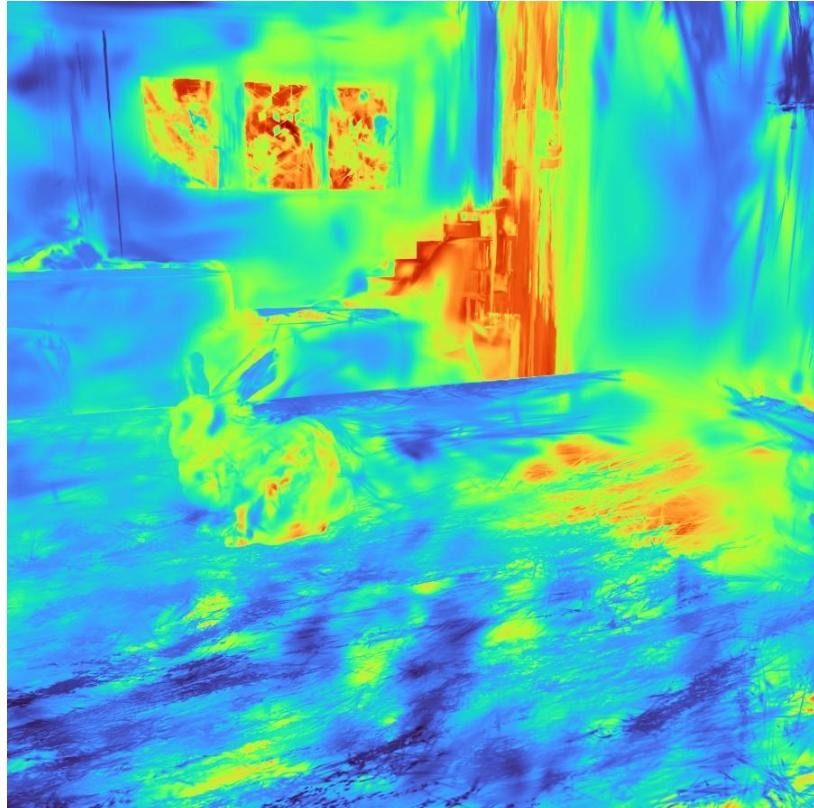
- I work on the fusion of vision and touch for dexterous manipulation
- Humans use their sense of vision and touch to understand a scene, with the right sensing, robots can roughly attempt this.
- Robots can touch what will fool vision



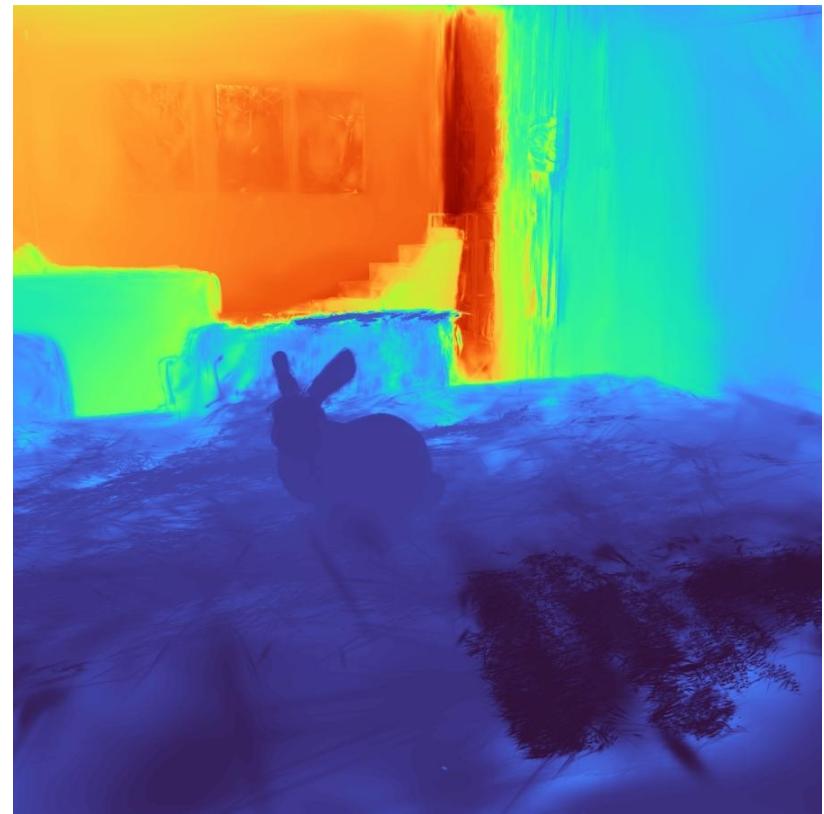
# Example where Vision is Fooled



## 3DGS Vanilla



## Our Method



# Introduction and Motivation to AI

- AI = Artificial Intelligence
- Algorithms that perform “human-like” tasks
- AI is all about **generalizing**
- You train an AI model on some stuff, then see how it does on stuff it’s never seen
- Simplest example:
  - Sentiment analysis
  - Train model on “**angry**” tweets, tell the model they are angry
  - Do the same on “**happy**” tweets
  - Show it new tweets and it should do well.

I hate LeBron. He is so overrated. Kobe is better

LeBron sucks

LeBum

LeBron is sooo good. The GOAT fr

King James!

I think LeBron is excellent  
<3

# Examples of AI

The screenshot shows the ChatGPT interface. On the left is a sidebar with the following items:

- ChatGPT
- DALL-E
- Explore GPTs

Below these are sections for "Today" and "Previous 7 Days" containing links to various AI-related topics like "MSE Loss Equation.", "Debugging PyTorch Gradient Error", etc.

The main area is titled "ChatGPT 4" and contains a message from "ChatGPT" about the greatest basketball player of all time.

## ChatGPT 4



### ChatGPT

The question of who is the greatest basketball player of all time (GOAT) is highly subjective and often leads to extensive debate among fans, analysts, and players themselves. Several legendary players are frequently mentioned in discussions about the GOAT in basketball, each with their own unique achievements and impact on the game. Here are a few of the most commonly cited candidates:

1. **Michael Jordan:** Often the first name mentioned in discussions about the GOAT, Jordan's impact on the NBA and basketball worldwide is unparalleled. He won six NBA championships with the Chicago Bulls and earned five regular-season MVP awards, among numerous other accolades. His competitiveness, skill, and influence make him a top pick for many.
2. **LeBron James:** LeBron is frequently compared to Jordan in GOAT debates. His longevity at the top of the game, combined with his versatility, has led to four NBA championships with three different teams, four MVP awards, and numerous other honors. James is known for his all-around game, leadership, and off-court activism.
3. **Kareem Abdul-Jabbar:** Holding the record for the most points scored in NBA history, Abdul-Jabbar's dominance is undeniable. With six MVP awards and six NBA championships, his skyhook shot became one of the most unstoppable moves in basketball history.

# Examples of AI





How it looks like  
snowboarding with your mates



15.9K



86



...



artem.kogut • Follow

io artem.kogut · Ori 3 people



≡ restaurant 🔍 X

Price

\$	\$\$	\$\$\$	\$\$\$\$
----	------	--------	----------

Rating at least

Any rating ▾

Sort by ▾ MORE OPTIONS

---

**Loló**  
4.5 ★★★★★ (849)  
\$· Mexican · 974 Valencia St  
Creative Mexican chow in colorful space  
Open until 11:00 PM



---

**Eric's Restaurant**  
4.3 ★★★★★ (254)  
\$· Chinese · 1500 Church St  
Chinese food in a casual atmosphere  
Open until 9:30 PM



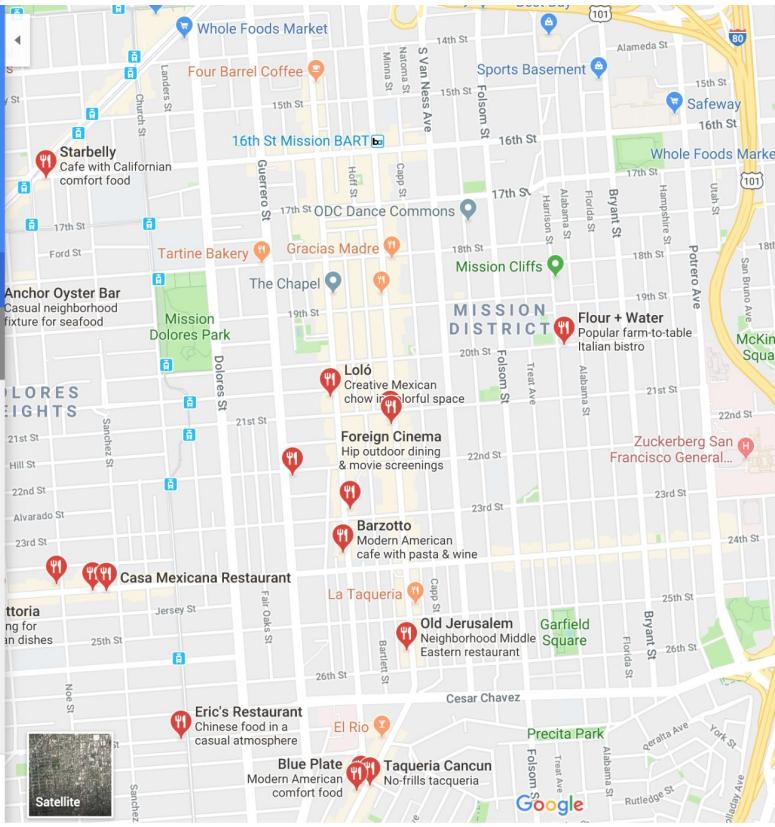
---

**Firefly Restaurant**  
4.5 ★★★★★ (364)  
\$\$\$ · American · 4288 24th St  
Modern comfort food  
Closing soon: 9:00 PM



---

**Blue Plate**



# Robot-Parkour-Learning

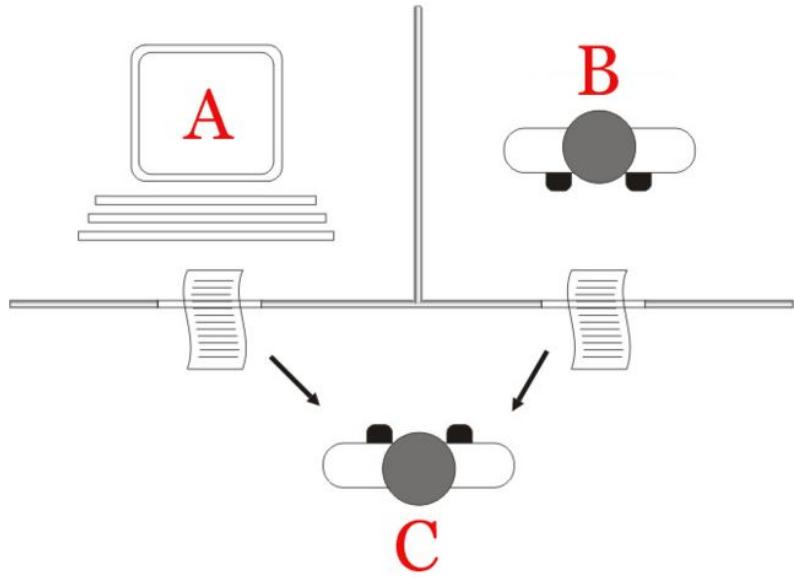


# Main Theme

- **Predict** what you want
- **Classify** restaurants, tweets, etc
- **Regress** driving time in certain areas vs driving skill
- For all: generalize to new topics!!
- Diverse examples to train on
- How to we get better at certain **skills**
  - **Diversified Practice**

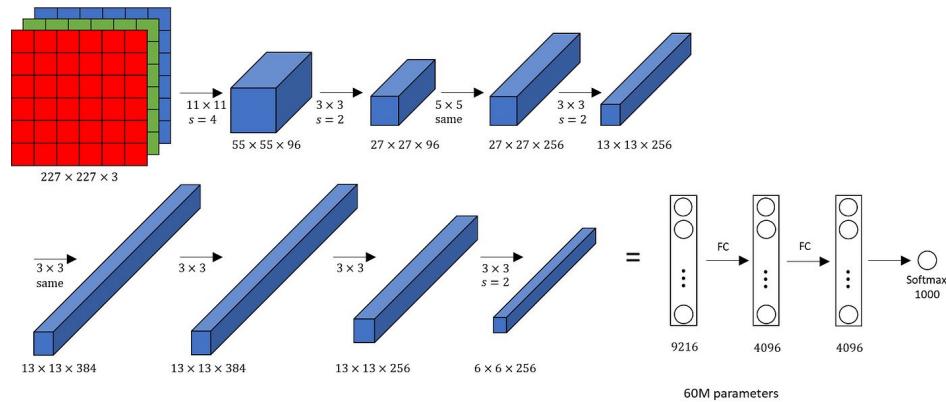
# History of AI

- Turing Test
- Term Coined 1956
- 90's: Internet and hardware
- Geoffrey Hinton Yoshua Bengio, Yann Lecun



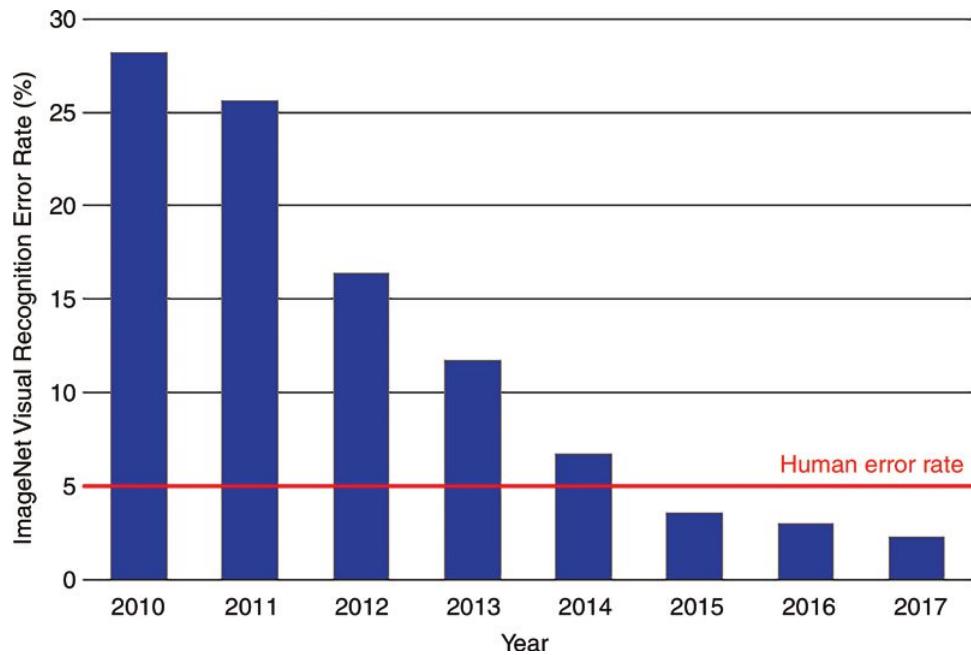
# The Explosion

- **Great hardware** to match the grand requirements by large models: training is possible!!
- Image recognition hits the scene
- Object detection works without noob feature extractors



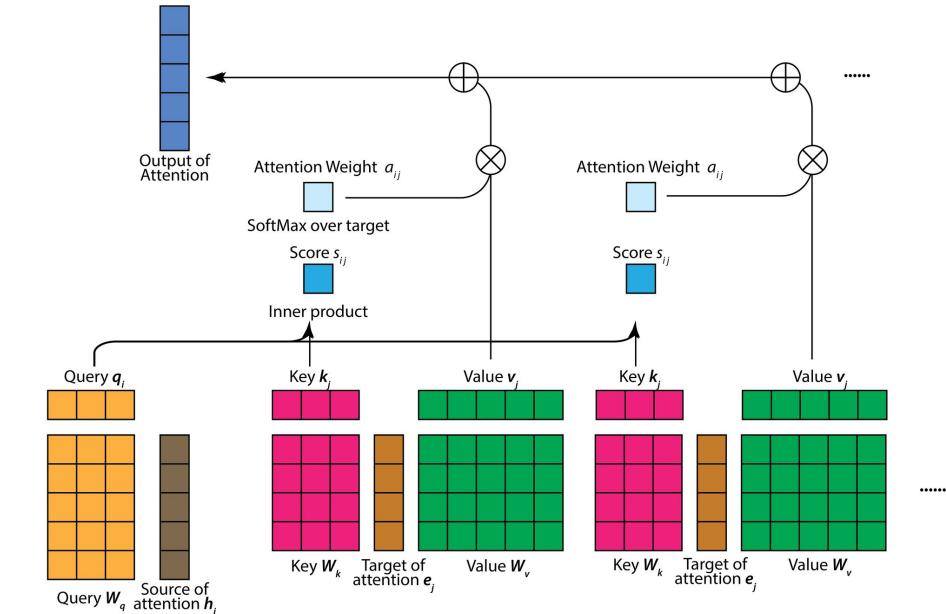
# ImageNet Results

- Insane results



# Transformers

- New architecture with new hardware
- Even better results!
- LLMs
  - Unreal hardware
  - Massive amount of parameters and data
  - The capacity to learn a lot
  - It is a massive **autocomplete**
  - **The GOAT of hockey is:**
    - Wayne Gretzky
    - This is free data to label with





**Yann LeCun** ✅ 💡 @ylecun · Feb 24

...

Ask not what LLMs can do for planning, ask what planning can do for LLMs.

59

119

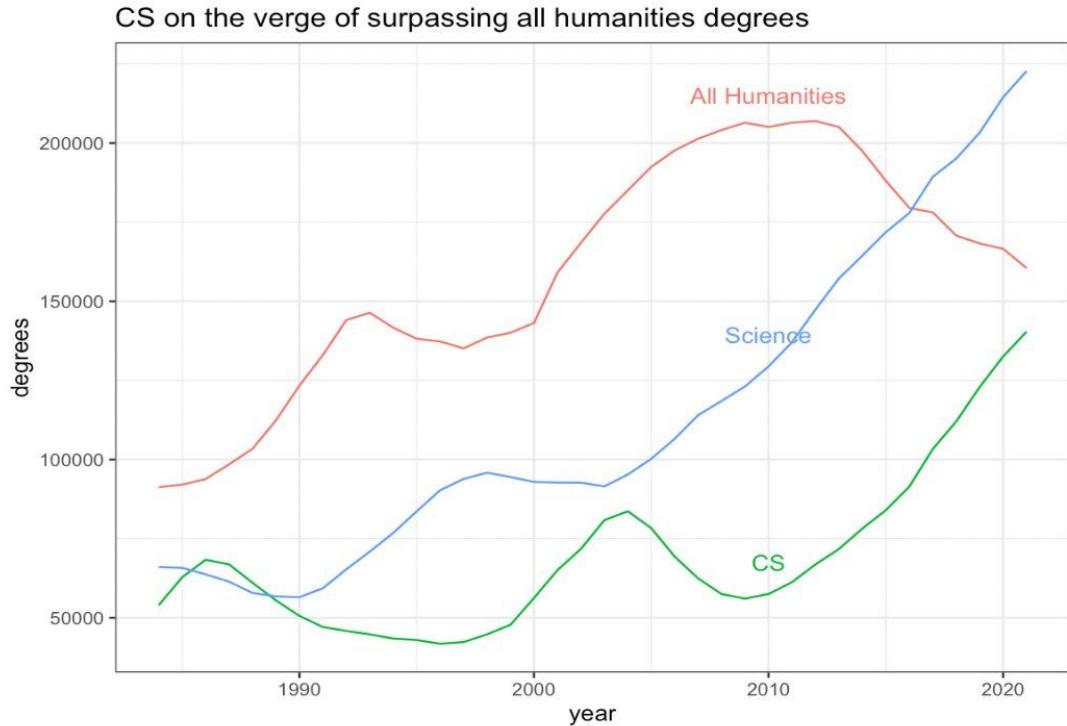
1K

138K



# Recent Years

- CS is becoming one of the most popular degrees alone

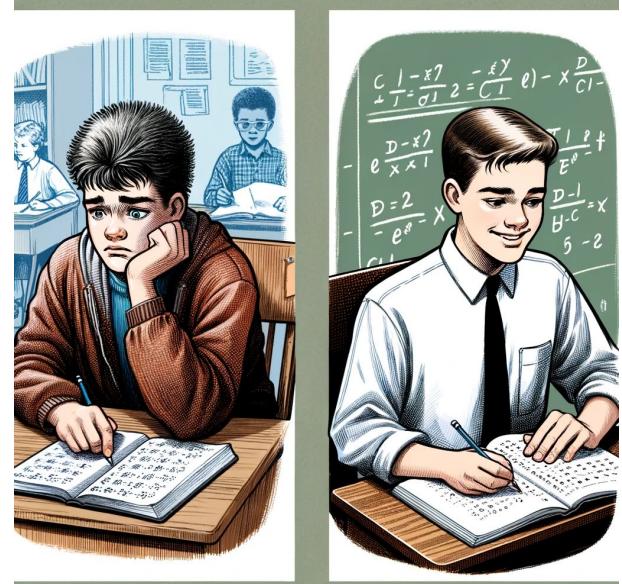


# Basic AI Techniques

- Regression
- Classification
- Reinforcement Learning

# Regression

- Create a function that predicts a continuous value given inputs.
- Predict future salary of high schoolers based on:
  - Family situation
  - Socioeconomic status
  - Test scores
  - Intangibles
- Putting things along a spectrum



25k

350k

# Classification

- Predict which “group” something belongs to
- Putting things in “buckets”
- Do this if it absolutely makes sense



# Reinforcement Learning

- Learning as you interact with the environment



# Model Components

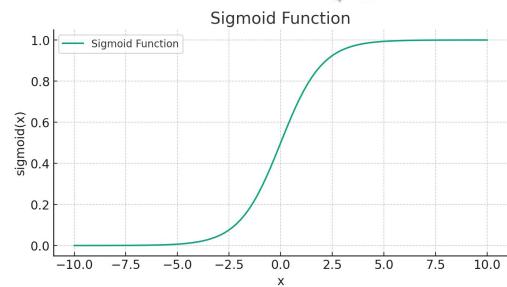
- After you know the task:
- Model architecture
- Model training parameters (this is super empirical)
- Loss function and making it work
- How to validate

# Model Architecture

- How you construct a function to predict or classify something
- Common “non-linear” function is to take your input, multiply it by weights, add a “bias”, then transform that function with a nice “non-linear” activation that works well

$$y = f(z) = f(w^T x + b)$$

$$f(z) = \frac{1}{1+e^{-z}}$$



# Model Components

- Combine a bunch of these together as one massive function to predict or classify.
- The problem formulation is much more important than the nitty gritty of the architecture

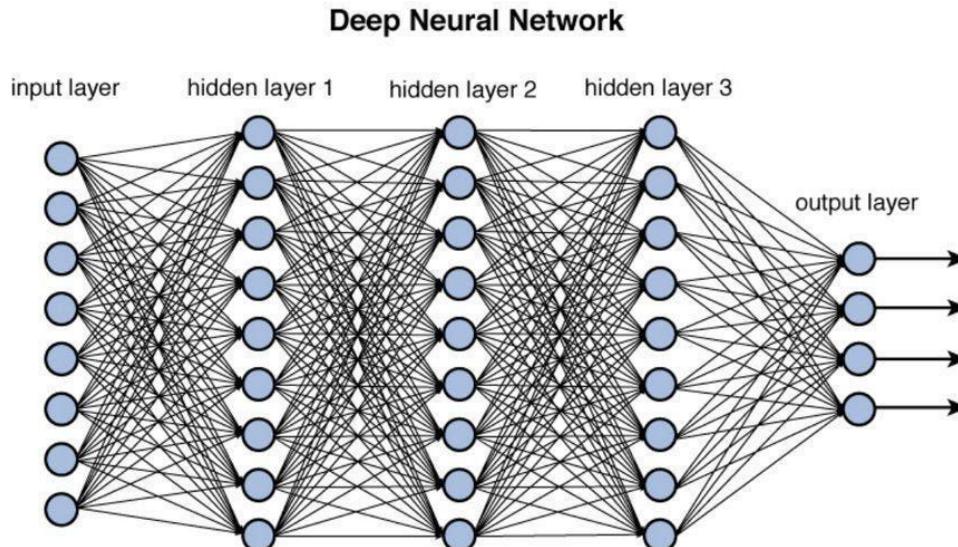


Figure 12.2 Deep network architecture with multiple layers.

## Model Training Parameters

- Answer: follow what is done in the literature

```
@dataclass
class GaussianSplattingModelConfig(ModelConfig):
    """Gaussian Splatting Model Config"""

    target: Type = field(default_factory=lambda: GaussianSplattingModel)
    warmup_length: int = 500
    """period of steps where refinement is turned off"""
    refine_every: int = 100
    """period of steps where gaussians are culled and densified"""
    resolution_schedule: int = 250
    """training starts at 1/d resolution, every n steps this is doubled"""
    num_downscales: int = 0
    """at the beginning, resolution is 1/2^d, where d is this number"""
    cull_alpha_thresh: float = 0.1
    """threshold of opacity for culling gaussians. One can set it to a lower value (e.g. 0.005) for higher quality."""
    cull_scale_thresh: float = 0.5
    """threshold of scale for culling huge gaussians"""
    continue_cull_post_densification: bool = True
    """If True, continue to cull gaussians post refinement"""
    reset_alpha_every: int = 30
    """Every this many refinement steps, reset the alpha"""
    densify_grad_thresh: float = 0.0002
    """threshold of positional gradient norm for densifying gaussians"""
    densify_size_thresh: float = 0.01
    """below this size, gaussians are "duplicated", otherwise split"""
    n_split_samples: int = 2
    """number of samples to split gaussians into"""
    sh_degree_interval: int = 1000
    """every n intervals turn on another sh degree"""
    cull_screen_size: float = 0.15
    """if a gaussian is more than this percent of screen space, cull it"""
    split_screen_size: float = 0.05
    """if a gaussian is more than this percent of screen space, split it"""
    stop_screen_size_at: int = 4000
    """stop culling/splitting at this step WRT screen size of gaussians"""
    random_init: bool = True
    """whether to initialize the positions uniformly randomly (not SFM points)"""
    ssim_lambda: float = 0.2
    """weight of ssim loss"""
    stop_split_at: int = 15000
    """stop splitting at this step"""
    sh_degree: int = 3
    """maximum degree of spherical harmonics to use"""
    camera_optimizer: CameraOptimizerConfig = CameraOptimizerConfig(mode="off")
    """camera optimizer config"""
    ...
```

# Loss Function

- A metric to define how “wrong” the model is
- Regression: you are more wrong if the predicted number is off from the real one
- Classification: if the class is wrong, fix the probability distribution

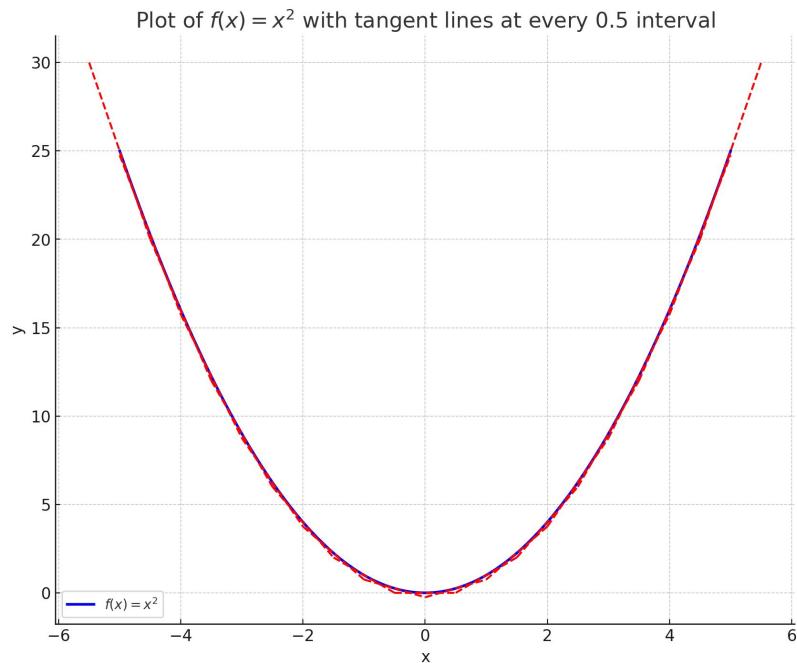
# Loss Functions

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log(\hat{y}_{ik})$$

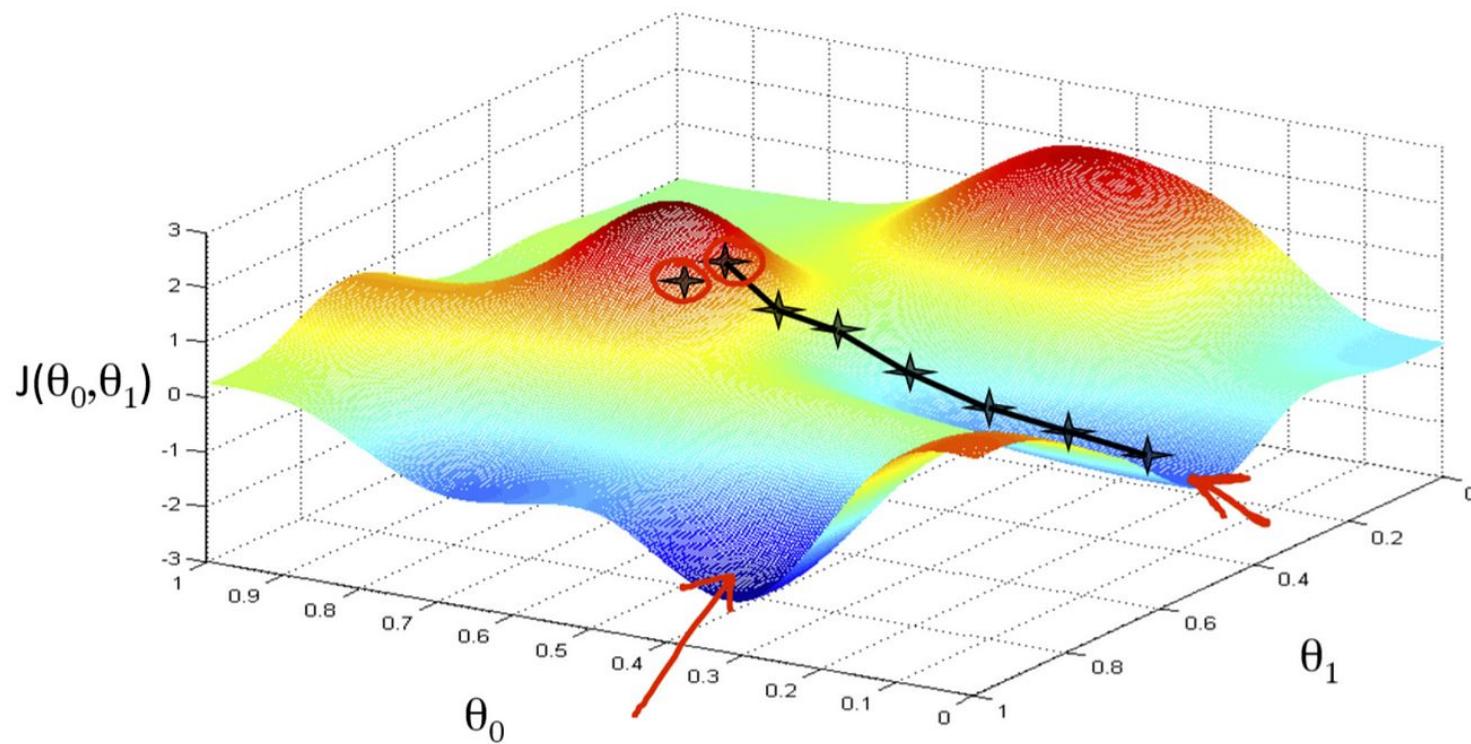
# Why Would a Loss Function Work?

- AI comes down to a minimization or maximization problem
- If we have a differentiable function, how can we get close to the min or max?
- **Set derivative to 0**
- This is called the gradient, and want to go in the opposite direction of it for minimization



# Gradient Descent

$$\theta_{\text{new}} = \theta_{\text{old}} - \alpha \cdot \nabla_{\theta} J(\theta)$$



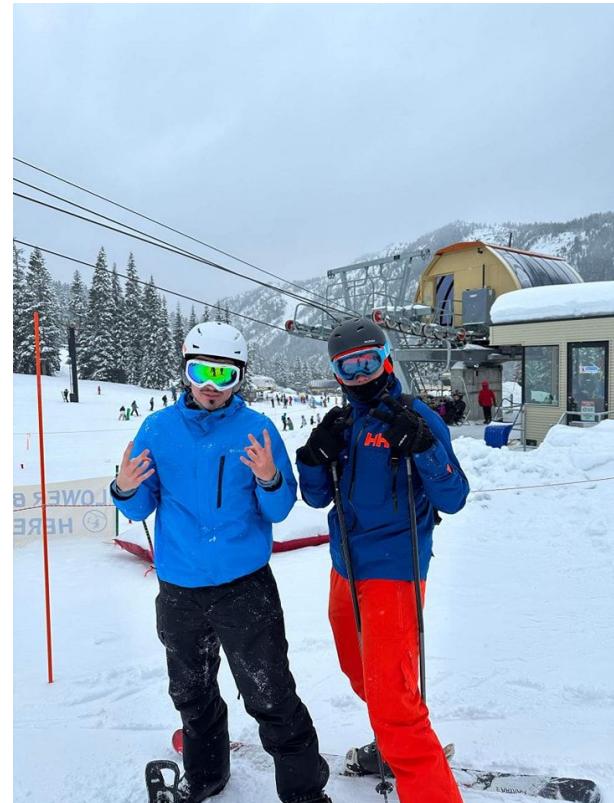
# How to Validate

- Once you have trained the model
- Validate on a **test set**
- The test set is unseen data that the model should generalize to



# My Snowboarding Journey

- Goal: Learn snowboarding and be good
- Specific goal: Go down a slope fast and while not falling
- More specific goal: have the angle w.r.t to the slope as perpendicular to the slope and not fall. My outputs are my body configuration
- **Regression of my joints, and then deploy to environment.**



# Model Architecture

- Massive neural network with a vision transformer
- State of the art
- Model needs to account for outside environment (snow, snow quality, slope) and myself (my balance of weight, etc)



# Model Training Parameters

- Train over a very, very long time
- Slowly update model based on the loss

# Loss Function

- When did I do something wrong? What would the “ground truth” do?
- **Falls:** extremely high loss. My speed effectively becomes 0
- **Low speed w/o falls:** average loss
- **Angle of board:** low loss
- **Edge contact with snow:** more loss when no edge is contacting the snow
- *If I am going fast, not falling, making an edge contact, have a good angle: I am good*



# Loss Function

$$L(v, f, \theta, e) = L_{fall}(f) + L_{speed}(v, f) + L_{angle}(\theta) + L_{edge}(e)$$

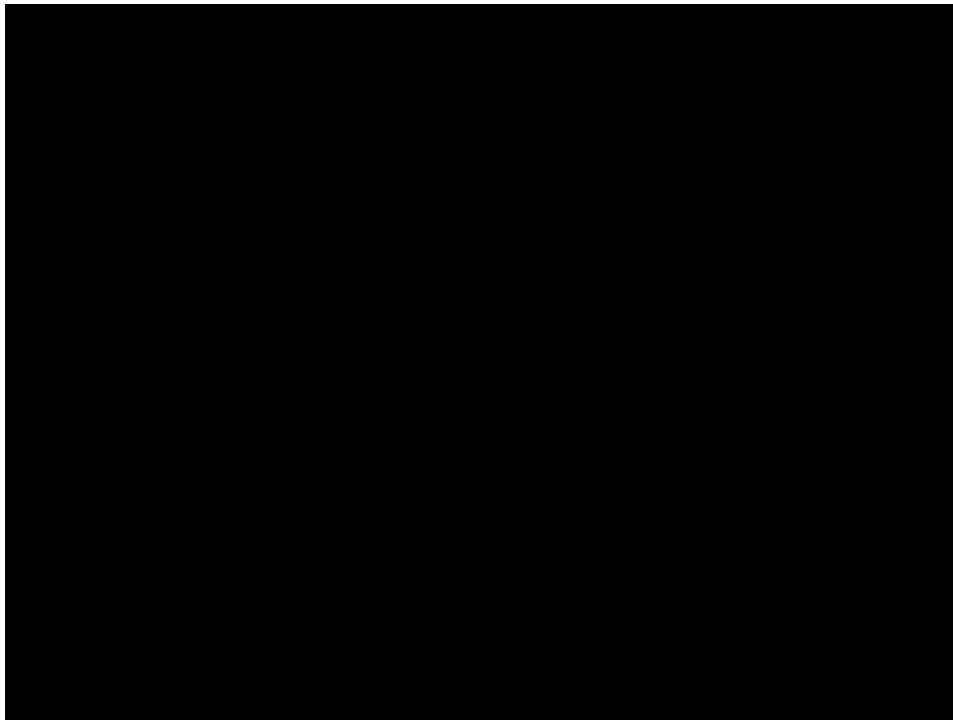


# Validation

- Test on an environment and see how I do
- Equivalent to sending it on an unseen slope
- How will I do?



# Validation



# Validation



# How You Can Get Started

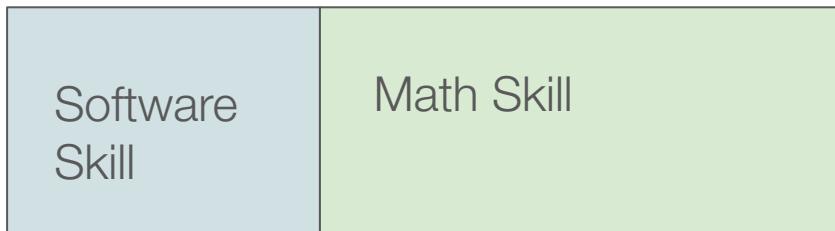
- [Andrew Ng AI courses](#)
- [3B1B NN](#)
- Install PyTorch
- Try out the tutorials
- Make a simple problem of your own!
  - Mine was digit recognition



# The Two Skills

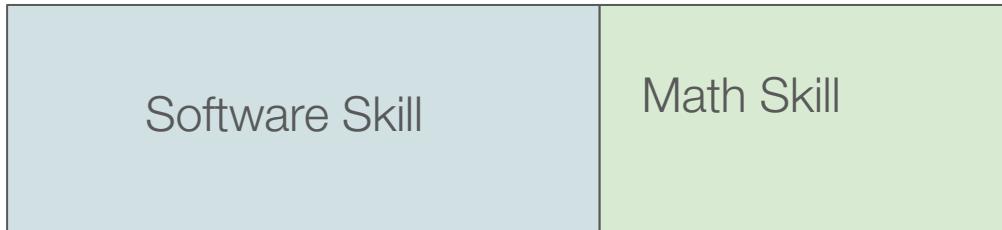


CS Major



Math Major

# What is a Worthwhile Skill?



**ML Engineer**

Most worth it



ML Researcher

Math is the  
killer

# Quick Tips About Industry and Research

- Try out research in undergrad. Find a younger prof with cool mentors and cool research
- Figure out the math
- When you have to install a new codebase, get it working as soon as possible, otherwise you enter installation hell
- Push early, push often
- Be kind

Questions?