Machine Learning (ML)

Alex Zhou, Ollin Boer Bohan

Some slides (barely) adapted from Noah Smith

Some quotes on ML

- “A breakthrough in machine learning would be worth ten Microsofts”
  - -Bill Gates
- “Machine learning is the next Internet”
  - Tony Tether (DARPA director)
- “Machine learning is going to result in a real revolution”
  - Greg Papadopoulos (Sun CTO)
- “Machine learning is today’s discontinuity”
  - Jerry Yang (Yahoo founder)
- “So hot right now”
  - Alex Zhou (CSE 160 TA)

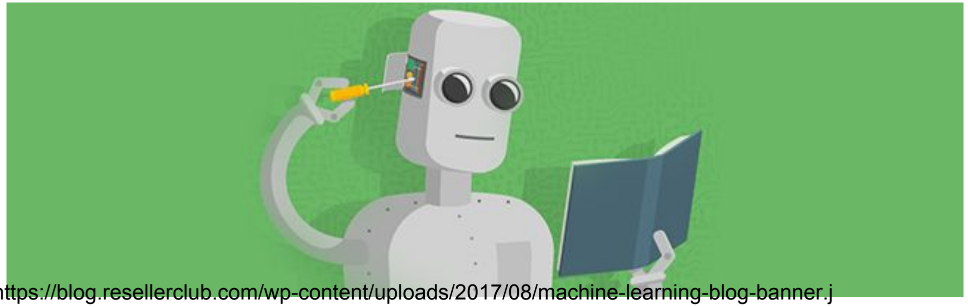
How do we define learning?

- Given the past, guess the future
- Generalize and adapt to new situations and information
- Get better with practice

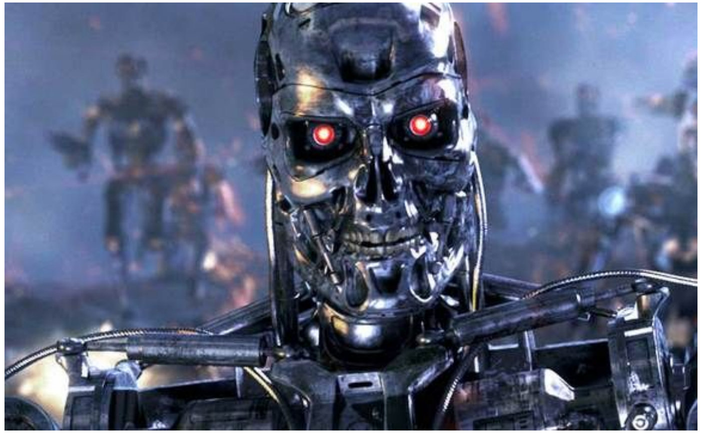
How do we measure if someone or something has learned?

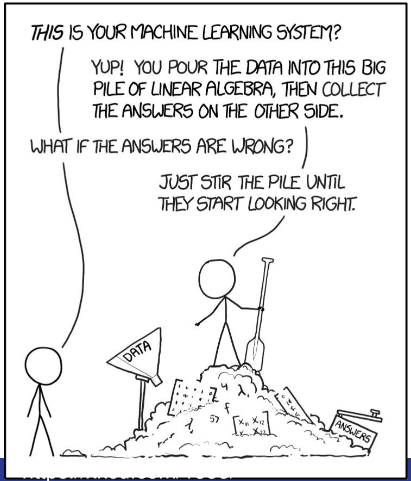
Make it take a test!

What is Machine Learning?

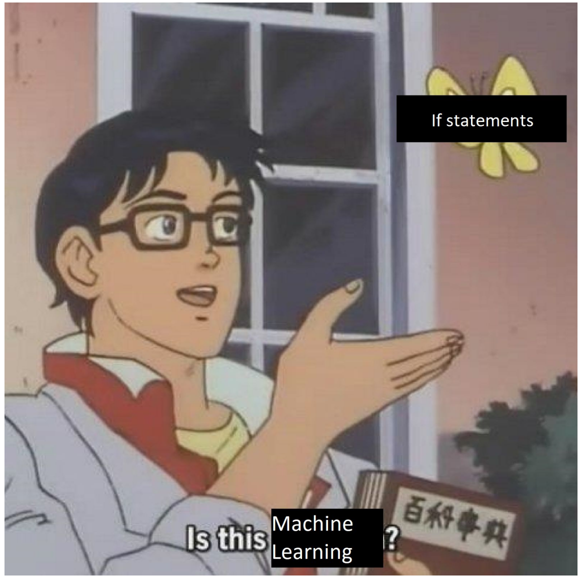


https://blog.resellerclub.com/wp-content/uploads/2017/08/machine-learning-blog-banner.jpg

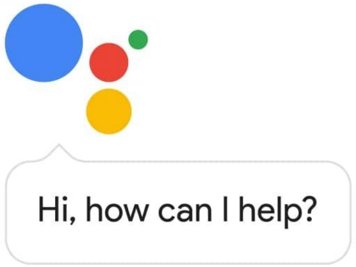


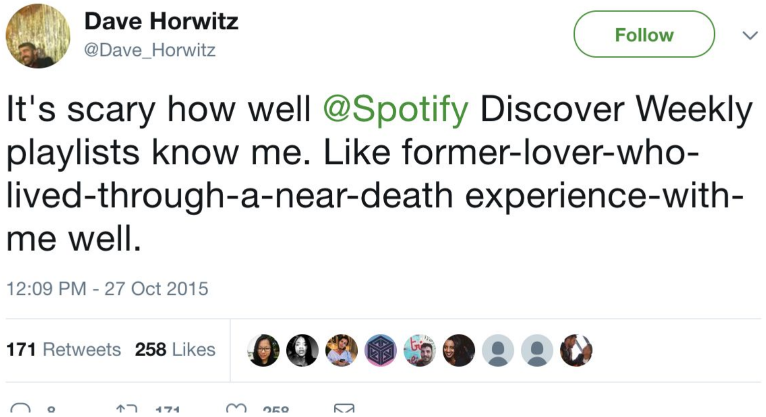

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

[illegible]



What is Machine Learning?




Dave Horwitz
@Dave_Horwitz

Follow

It's scary how well @Spotify Discover Weekly playlists know me. Like former-lover-who-lived-through-a-near-death experience-with-me well.

12:09 PM - 27 Oct 2015

171 Retweets **258** Likes

https://xkcd.com/1838/

Common Applications

- Classifying documents (e.g., “is this email spam?”)
- Labeling/captioning images (e.g., “who’s in this picture?”, “what’s happening in this picture?)
- Predicting the future: weather, finance, medical outcomes
- Recommending content (e.g., movies, books, music)
- Fraud detection (e.g. detecting money laundering)
- Self driving cars (a combination of a number of these and other problems)

Today

ML a normal part of our lives:

- Video/image processing
- Speech/language processing
- Controlling robots
- Computational biology
- Healthcare analysis
- Annoying recommendation systems

When people say “AI” these days, they usually mean “ML”

Many of these problems are too complex to solve “by hand”

Is ML Magic?

More like gardening

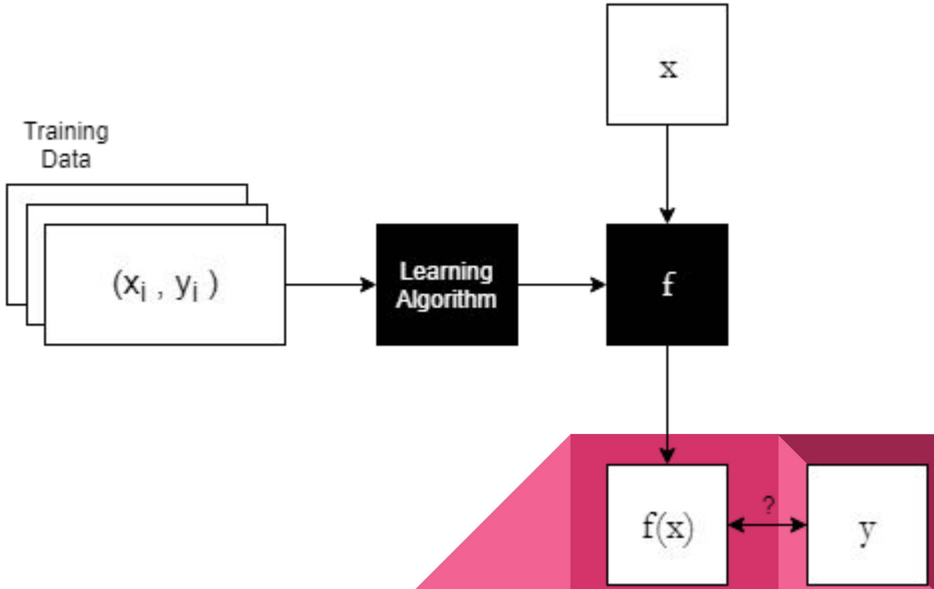
Successful plants (programs) require:

- Seeds (algorithms)
- Nutrients (data)
- A gardener (ML expert)

We can kind of predict how gardens will grow, but can’t be certain, and our understanding is still evolving!

What does (Trained, Supervised) ML Look like?

- **Training**: Given a **training set** of input/output pairs (x, y), train a function f where f(x) reproduces y.
- **Testing**: Take a **test set** of input/output pairs (x, y) and see how accurately f(x) matches with y



Input/Output? (x,y)?

Our x is usually a vector consisting of the values of different characteristics (**features**), can be anything.:

- Age
- Height
- Rainfall
- Time spent on a screen
- Color values of a pixel

Our y is some sort of output associated with our features:

- Some real value (regression)
- A **label** (classification)
- An ordering (ranking)
- A vector (multivariate regression/classification)
- ...

Decision Trees (or, 20 Questions!)

- Classification algorithm
- Common day-to-day decision-making process
- 20 Questions
  - Start general
  - Get more specific
  - Settle on an educated guess
- Decision tree
  - Examine training data
  - Decide how to separate it at different levels
  - Settle on an educated guess

Decision Tree Example

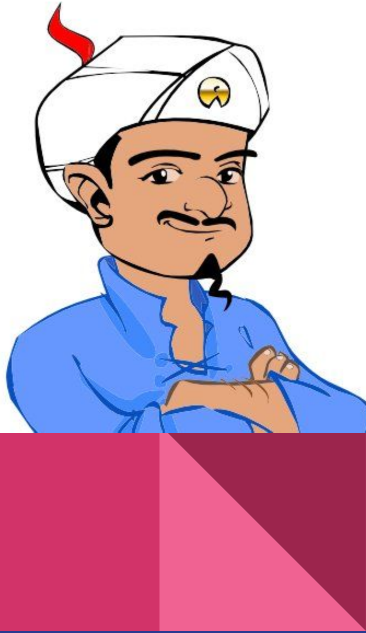
Not sure if you need to act like you know someone

You know (**xs**, **features**):

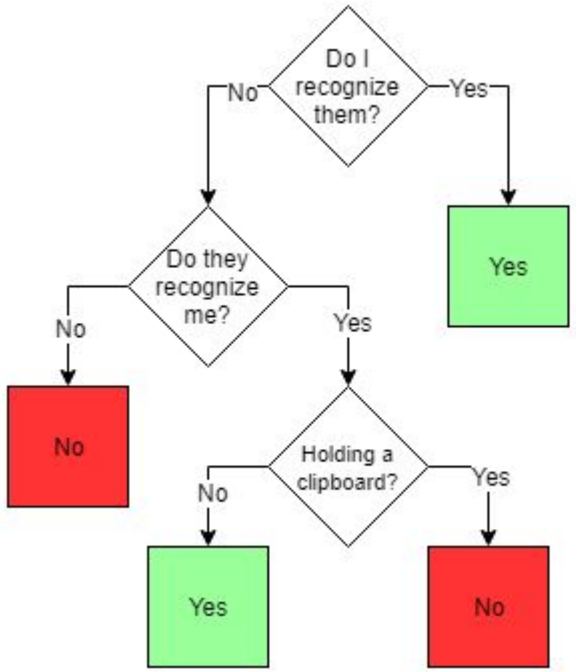
- If you recognize them
- If they recognize you
- They know your name
- They're holding a clipboard

Want to decide (**y**, **label**):

- Do I act like I know them? (Y/N, 1/0)



Decision Tree Example



Decision Trees in Python

Implemented as part of `sklearn`. The code to use it fits on a slide:

```
from sklearn import tree
X = [[20], [20], [20], [30], [30], [30]]
Y = [1, 1, 1, 0, 0, 0]
model = tree.DecisionTreeClassifier()
model.fit(X, Y)
```

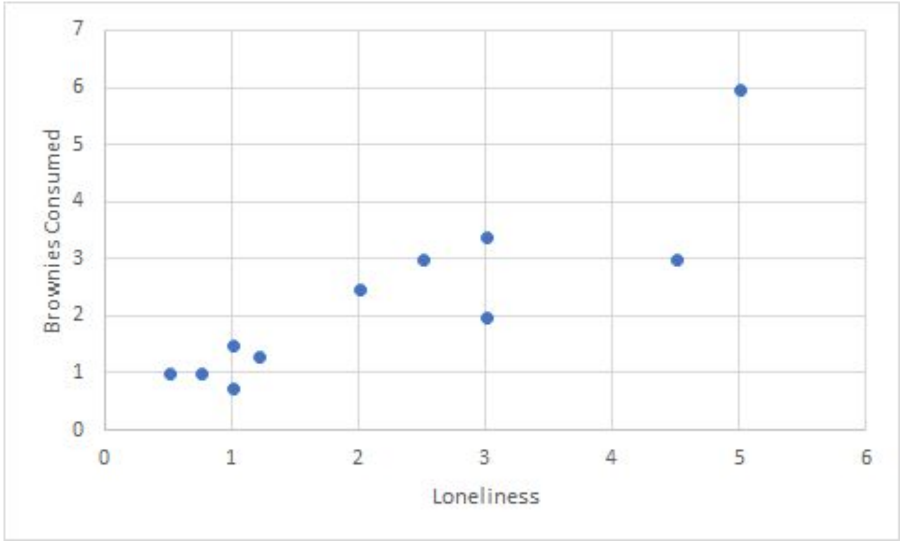
Regression

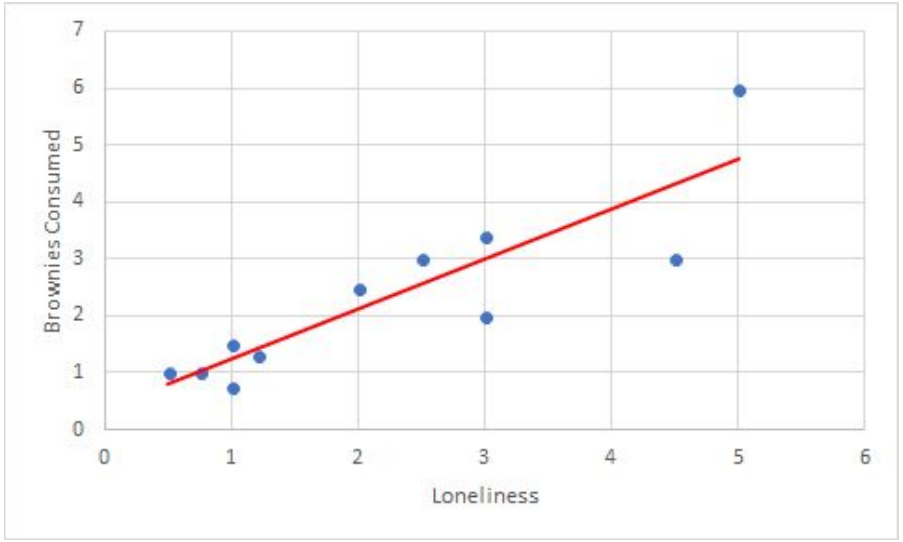
Imagine now we want to predict how many brownies I can eat, right now.

We might consider as features:

- Time since I last ate
- Calories I’ve consumed today
- How many calories I’ve burned today
- **How lonely I’m feeling?**

Linear Regression





A model **y = mx + b**, with two parameters (**m**, **b**).

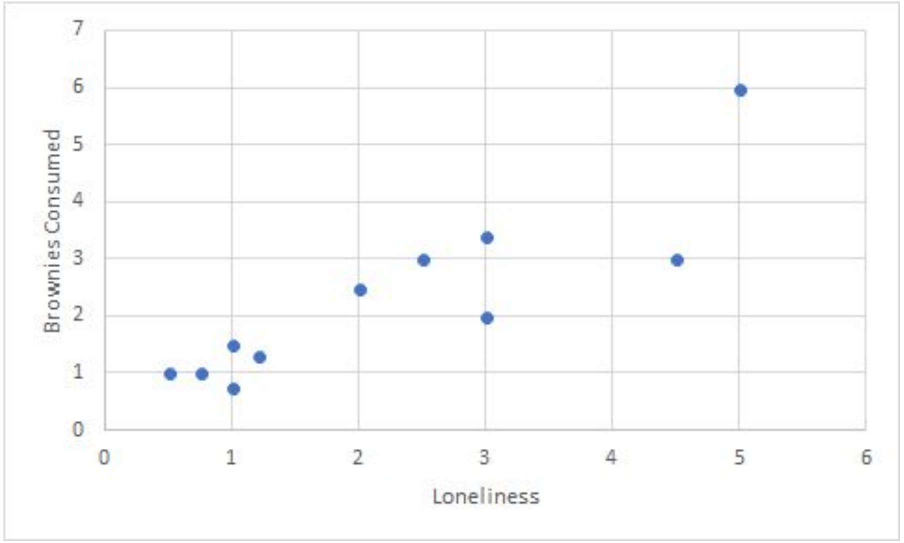
Draw the "best" line.

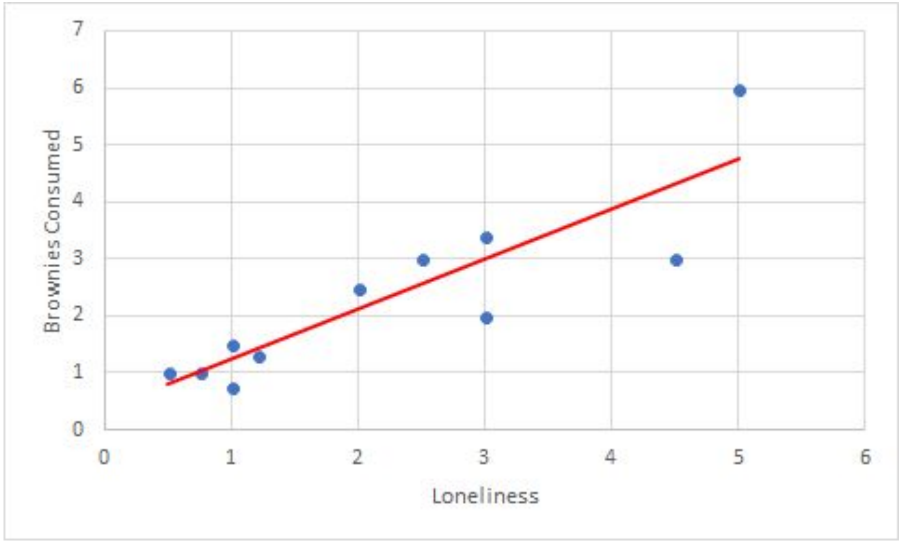
Linear Regression in Python

Implemented as part of `sklearn`. The code to use it fits on a slide:

```
from sklearn import linear_model
X = [[1], [2], [3], [4]]
Y = [2, 4, 6, 8]
model = linear_model.LinearRegression()
model.fit(X, Y)
```

Linear Regression





A model **y = mx + b**, with two parameters (m, b).

Draw the "best" line. **What does that mean?**

Squared Error

Remember Homework 6?

- The difference between predicted and actual, squared, summed across all examples
- Find the line that minimizes squared error (in 2 dimensions, plane in 3 dimensions...)
  - Does this by choosing weights (**m**, **b**); changing the weights moves the line.
- How do we find the best weights?
  - Derivatives of the loss: see MATH 12X series
  - Tons of features (high dimensional) data?
  - Expensive/difficult to just compute the whole gradient
  - Loss functions weird in high dimensions

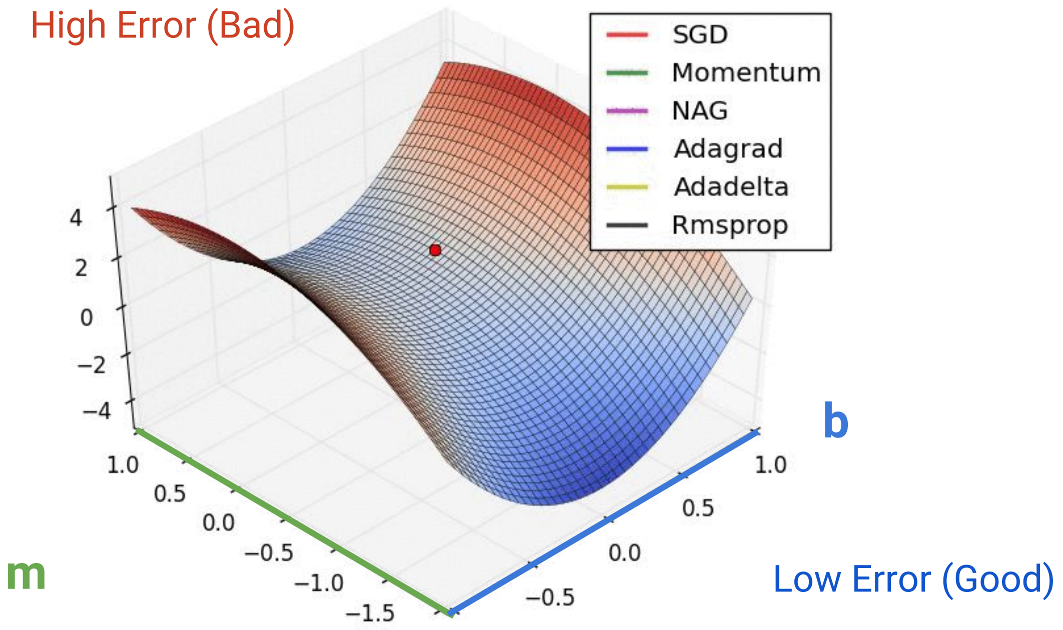
How does training work/Gradient Descent

Let’s say we’re playing a fun game: Blindfolded Mountain Descent!

Find a valley before you freeze to death!

How does training work/Gradient Descent

- Intuitively: if you want to find a low-altitude point, try to move downhill.



- "Movement" is adjustments to model parameters (**m**, **b**).
- "Altitude" is the error (how bad is that **m**, **b** at predicting the data?).

Neural Networks

- Not the same as biological neural networks (much more limited!)
- Still the most heavily-parameterized models money can buy.
- Can learn a mathematical function mapping from one set of points → another set of points.
  - Chess/Go boards → good moves to play (AlphaZero, beat world champion)
  - Text to spoken audio (WaveNet, in Google Assistant)
  - Random noise to ramen noodles? (Progressive GANs, kinda pointless?)
- Trained by gradient descent (sorta)
  - Also backpropagation

How do Neural Networks work?

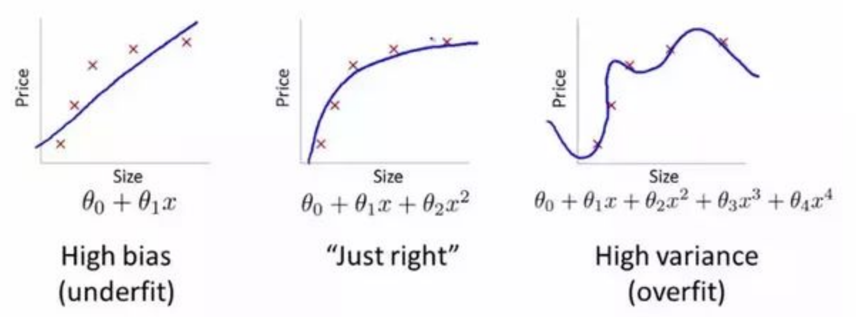
- Think linear regression from before, but grown up...
- Instead of **y = mx + b**, it's

  $Y_1 = \sigma(W_1 X + B_1)$,
  $Y_2 = \sigma(W_2 Y_1 + B_2)$ ...
  $Y_n = \sigma(W_n Y_{n-1} + B_n)$
- What's different?
  - Everything has more dimensions (so B_1 might be **[0.1, 0.5, -0.1, 0.5, 10, ...]**).
  - More layers! More parameters! Lots of **W**s. Lots of **B**s.
  - There's a "nonlinearity" **σ**, which squishes things before passing them to next layer. Without this, there'd be no point in stacking layers...

More parameters: Overfitting, Bias, and Variance

When a model does much better on the training data than on the testing data, we say it is **overfitting** (high **variance**). This is a student who memorizes examples instead of learning general rules.

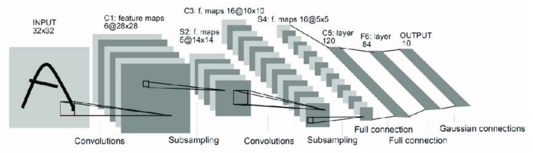
When a model does poorly on all data, we say it is **underfitting** (high **bias**). This is a student who ignores the specifics of each problem and just tries to answer them all the same (incorrect) way.



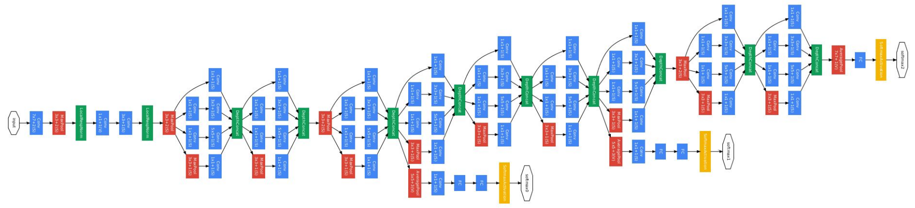
Solving the Bias-Variance Tradeoff

If your model **underfits**, add more parameters!

1998

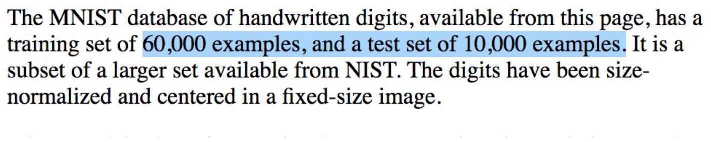


2014



If your model **overfits**, get more data!

1998



The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

2010-

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

Neural Networks in Python

Unless you're dealing with very high-dimensional data (images, audio, text), it's probably overkill. But it's really powerful!

Neural net starter kit:

- Google Colab (online Jupyter notebooks with GPUs)
- Keras (simple-ish framework for writing neural nets in Python)
  - Other options: Torch, Tensorflow
- Patience; getting everything set up can be painful and neural nets are finicky :P

Things we didn’t cover

- Train/Dev/Test set splits
- Backpropagation
- Cross-validation
- Unsupervised learning
- Regularization
- Ensembling
- Other neural net forms (CNNs, LSTMs...)

Learning More

- **Decision Trees:** A visual introduction to machine learning: http://www.r2d3.us
- **Neural Nets:** How Machines *Really* Learn and But What *is* a Neural Network?
- **More Neural Nets:** Tensorflow playground: https://playground.tensorflow.org/, CS231n Lectures, course.fast.ai
- Hal Daume’s free machine learning textbook: http://ciml.info/
- **CSE 416/446**