# Conclusion

Ravid Shwart Ziv

CDS, NYU

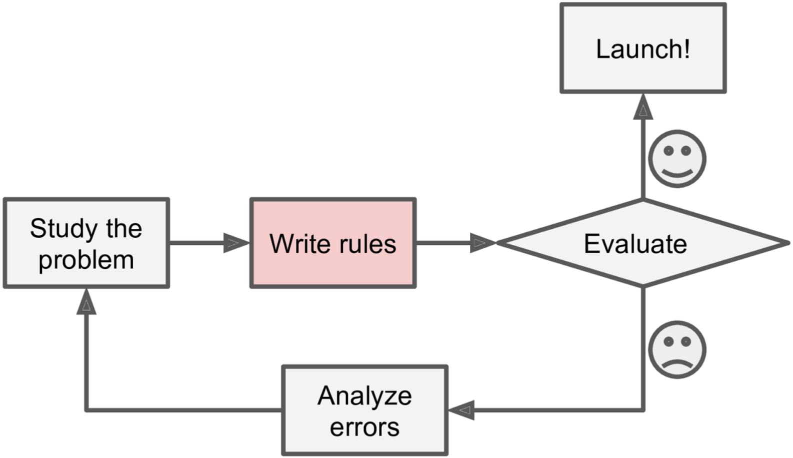
May 2, 2023

# Contents

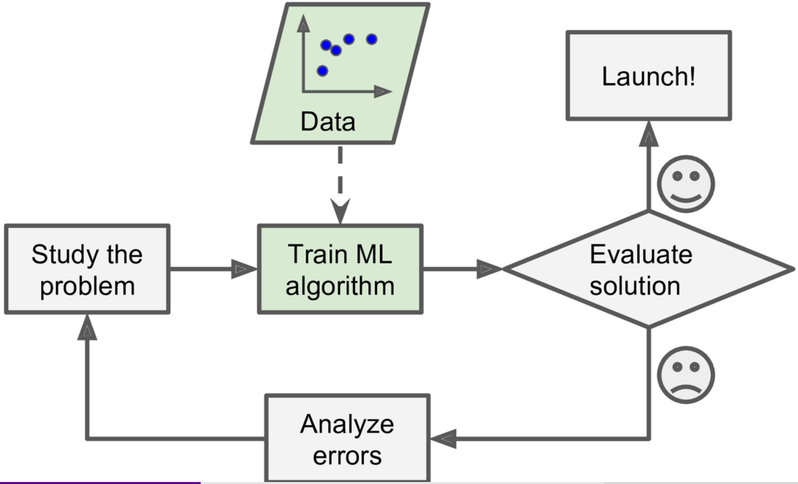
# Summary of This Course

# Rule-based approach

Code an algorithm



Learn the algorithm from data



# Machine learning vs rule-based approach

$$\text{data} \xrightarrow{\text{learning algorithm}} \text{model}$$

- Machine learning is the main driving force of modern AI.
- Shift from specifying *how to* solve the problem to *what* the solution looks like.
- Promise: *generalization* to unseen examples.

# Typical machine learning recipe

**Model** A family of predictors that map from the input space to the action space:

$$f(x;\theta) = a. \tag{1}$$

**Objective** Empirical risk minimization:

$$\min_{\theta} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} L(x,y,\theta). \tag{2}$$

**Algorithm** Stochastic gradient descent:

$$\theta \leftarrow \theta - \alpha_t \nabla_{\theta} L(x,y,\theta). \tag{3}$$

# Models

| | |
|---|---|
| Linear | Perceptron, conditional probability models, SVMs |
| Non-linear | Kernelized models, trees, basis function models, neural nets |

# Models

**Linear** Perceptron, conditional probability models, SVMs

**Non-linear** Kernelized models, trees, basis function models, neural nets

How to choose the model family?

- Trade-offs:
  - approximation error and estimation error (bias and variance),
  - accuracy and efficiency (during both training and inference).
- Start from the task requirements, e.g. amount of data, computation resource

# Objectives

**Loss functions** How far off a prediction is from the target, e.g. 0-1 loss, margin-based loss, squared loss.

# Objectives

**Loss functions** How far off a prediction is from the target, e.g. 0-1 loss, margin-based loss, squared loss.

**Risk** Expected loss - but expectation over what?

- Frequentist approach: expectation over data.
  - Empirical risk minimization, i.e. average loss on the training data.
  - Regularization: balance estimation error and generalization error.
- Bayesian approach: expectation over parameters.
  - Posterior: prior belief updated by observed data.
  - Bayes action minimizes the posterior risk.

# Algorithms

**Learning** Find model parameters—often an optimization problem.

- (Stocahstic) (sub)gradient descent
- Functional gradient descent (gradient boosting)
- Convex vs non-convex objectives

**Inference** Answer questions given a learned model.

- Bayesian inference: compute various quantities given the posterior.
- Dynamic programming: compute $\arg\max$ in structured prediction.

# Debugging Machine Learning Algorithms

# Get it to work

- This course provides you a *toolbox*.
- Motivate a tool from the problem.
- Keep the solution as simple as possible.

# How to start on a problem

Given a task, e.g., spam classification, how should we get started on the problem?

- Approach 1: Analyze the problem/dataset, design the right features/models/objectives carefully, then implement the algorithm.
- Approach 2: Implement a simple baseline quickly, see what's wrong with it and fix its problems.

Reference: Andrew Ng's advice on applying ML.

# How to start on a problem

Given a task, e.g., spam classification, how should we get started on the problem?

- Approach 1: Analyze the problem/dataset, design the right features/models/objectives carefully, then implement the algorithm.
- Approach 2: Implement a simple baseline quickly, see what's wrong with it and fix its problems.

In practice, often an *iterative process* with a mix of both approaches.

Reference: Andrew Ng's advice on applying ML.

# Practical scenario

You build a logistic regression model with L2 regularization using bag-of-words features for the spam classification task. After running SGD for some iterations, validation error is 20%. What do you do now?

# Practical scenario

You build a logistic regression model with L2 regularization using bag-of-words features for the spam classification task. After running SGD for some iterations, validation error is 20%. What do you do now?

- Get more data.
- Better feature selection.
- Try another optimizer.
- Run SGD for longer.
- Tune weights on the regularizer.
- Try another model.
- ...

# Setting expectations on performance

**Lower bound** What's the dumbest/simplest predictor you can build?

# Setting expectations on performance

Lower bound What's the dumbest/simplest predictor you can build?

- Always predict the majority class / target mean.
- Decision/regression stump, i.e. use only one feature.
- Linear model (with regularization).

# Setting expectations on performance

Lower bound What's the dumbest/simplest predictor you can build?

- Always predict the majority class / target mean.
- Decision/regression stump, i.e. use only one feature.
- Linear model (with regularization).

Upper bound What's the best performance you can get given the hypothesis space?

# Setting expectations on performance

Lower bound What's the dumbest/simplest predictor you can build?

- Always predict the majority class / target mean.
- Decision/regression stump, i.e. use only one feature.
- Linear model (with regularization).

Upper bound What's the best performance you can get given the hypothesis space?

- Fit your model on the validation set without regularization, i.e. cheating.

**Overfitting** Low bias, high variance

**Underfitting** High bias, low variance

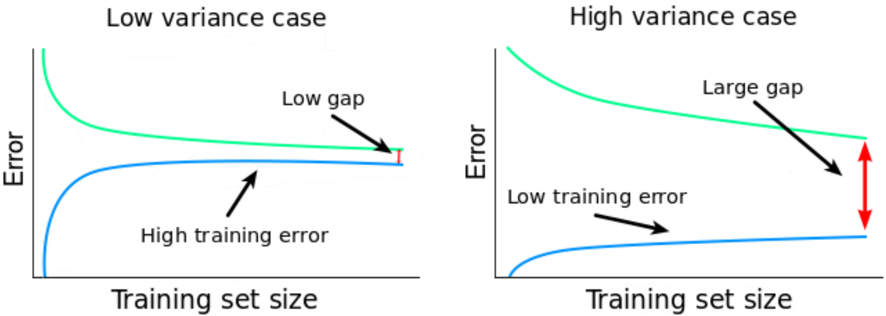



Figure: https://www.dataquest.io/blog/learning-curves-machine-learning/.

# Fixes for high bias/variance

| | |
|---|---|
| High variance | • More data, bagging<br>• Simpler feature, e.g., remove rare words, L1 regularization<br>• Simpler model, increase regularization strength |
| High bias | • Better feature, e.g., n-grams<br>• Increase model complexity, e.g., neural nets<br>• Reduce regularization strength |

# Optimization error

Is there a problem with optimization?

# Optimization error

Is there a problem with optimization?

- Look at the learning curve. Is the loss decreasing?
- Verify initial loss, e.g., $p(y \mid x)$ is a uniform distribution.
- Add a cheating feature (i.e. the label), does it achieve zero training error?

# Optimization error

Is there a problem with optimization?

- Look at the learning curve. Is the loss decreasing?
- Verify initial loss, e.g., $p(y \mid x)$ is a uniform distribution.
- Add a cheating feature (i.e. the label), does it achieve zero training error?

Fixes

- Tune the hyperparameters, e.g., the learning rate.
- Try another optimizer.

# Error analysis

Data **Look at your data, but only the training/validation data!**

- Preprocessing errors, data corruption.
- Label imbalance, noise.
- Look at misclassified examples.
- In general, print out as much information about the data as possible.

# Error analysis

**Data** **Look at your data, but only the training/validation data!**

- Preprocessing errors, data corruption.
- Label imbalance, noise.
- Look at misclassified examples.
- In general, print out as much information about the data as possible.

**Model** How much does each component help?

- **Ablation study**: change one thing at a time, e.g., how much does the performance change if you add/remove the component?
- **Oracle experiment**: use groundtruth information, e.g., how much gain do you get if the output of the component is perfect?
- Similar to how you would debug your code.

# Summary

- Many things can go wrong: data, model, learning algorithm etc.
- Fixes should be motivated by a problem identified from your analysis.
- Verify that the fix actually fixes the problem it's intended to fix!
- Be creative.

# Next Steps

# Other courses offered by CDS

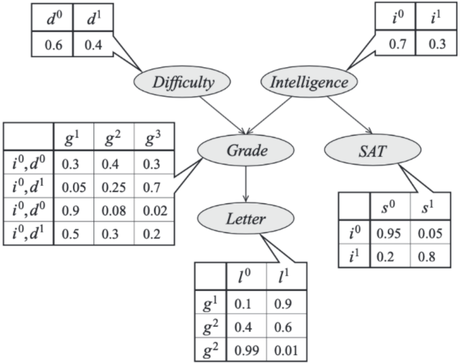
**Foundations**

- DS-GA 1005 Inference and Representation
- DS-GA 1008 Deep Learning
- DS-GA 1013 Mathematical Tools for Data Science
- DS-GA 3001 Special Topics in Data Science: Responsible Data Science

**Applications**

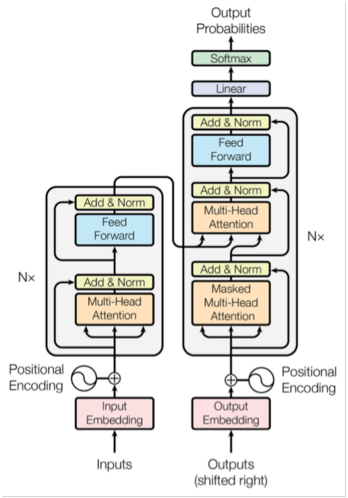
- DS-GA 1011 Natural Language Processing with Representation Learning
- DS-GA 1012 Natural Language Understanding and Computational Semantics
- DS-GA 3001 Special Topics in Data Science: Intro to Computer Vision
- DS-GA 3001 Special Topics in Data Science: Computational Cognitive Modeling

# Probabilistic graphical models




- DS-GA 1005
- Model: represent the world as a joint distribution of observed and unobserved variables.
- Learning: estimate parameters of the distribution from data, e.g., MLE.
- Inference: compute posterior distribution of the latent variables.

# Deep learning




- DS-GA 1008
- Advanced neural network architectures: CNN, RNN, Seq2Seq, memory networks, Transformers etc.
- Deep generative models: auto-encoders, GANs, energy-based models.
- Representation learning: self-supervised learning.

# Ethics

- DS-GA 3001.009
- Remember that the model we build will eventually impact *people*.
- Privacy: will the model/data leak user information?
- Fairness: does the model "discriminate" a group of people?
- Bias: does the model inherit bias in our society which generates the data?

# Language

- DS-GA-10011, DS-GA-1012
- Goal: teach computers to understand human languages.
- Properties: discrete, compositional, ambiguous
- Representation: how do we represent words, sentences, and documents?
- Conditional sequence modeling: machine translation, summarization, dialogue etc.
- Language understanding tasks: information extraction, entailment, question answering.

# Vision

- DS-GA 3001.005
- Goal: teach computers to see the world as we see it.
- Challenges:
  - Low-level input: large amounts of raw pixels.
  - Variations of the same object: illumination, view point, occlusion, scale etc.
- Tasks: object detection, video understanding, 3D vision etc.
- Many related areas: medical imaging, robotics.

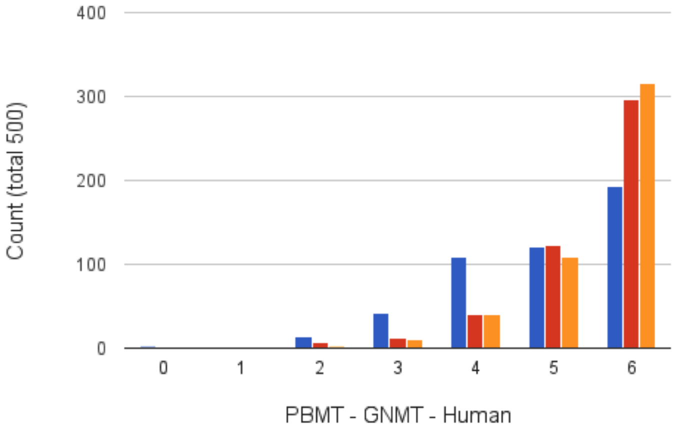
# Cognitive modeling

- DS-GA 1016
- Goal: computational approaches to understanding human intelligence.
- Why should machine learning care?
  - Humans learn from a few examples—few shot learning.
  - Humans adapt to new concepts/environments quickly—transfer learning.
  - AI and cognitive science inform each other at both conceptual and technical levels.
    - Reinforcement learning, neural nets, Bayesian modeling etc.

# Machine Learning in the Wild

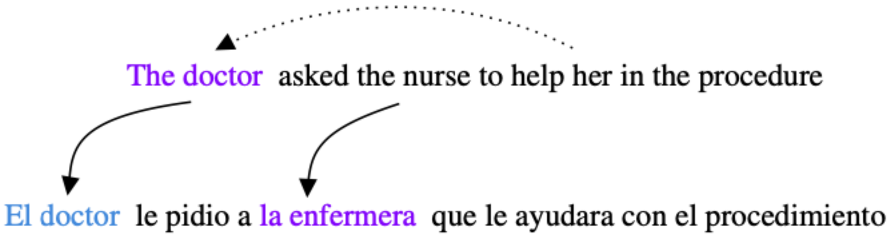
# Google neural machine translation

Approaching human performance [Wu+ 2016].



# Gender bias in MT

MT systems are prone to gender-biased translation errors [Stanovsky+ 2019].



# Object recognition

Lower error rate than humans on ImageNet.

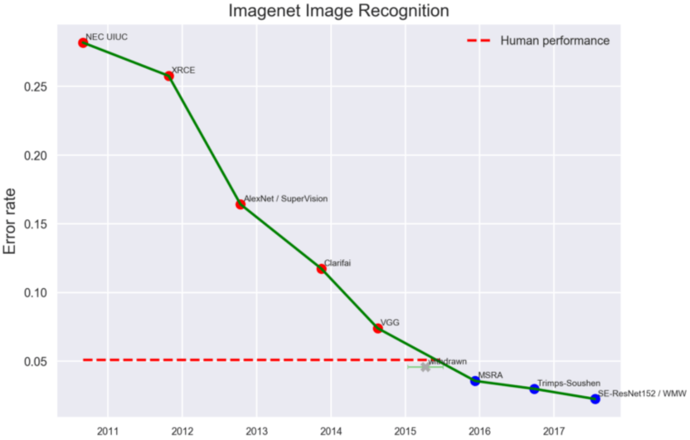
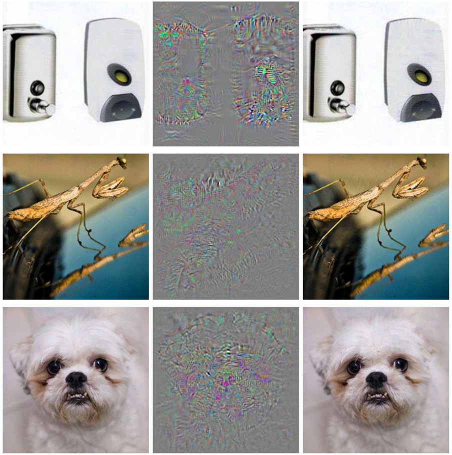



Chart from Measuring the Progress of AI Research by EFF (CC BY-SA).

# Adversarial examples

Left: correct. Middle: added noise. Right: ostrich. [Szegedy+, 2013]



# Adversarial examples

Left: real graffiti. Right: advesarial patches. [Eykholt+, 2018]

# Reinforcement learning

Deepmind's AlphaGo beats the world's best human Go player.

# Reward hacking

Instead of finishing the course, agent learns to circle and attack target repeated despite bumping into other boats.

https://openai.com/blog/faulty-reward-functions/

# Pretrained Models, Large Language Models

- Finish!

# Scaling of Parameters

- Do we still need ML?

# Ethics of Machine Learning

Ensure that the behavior of machines towards human users is ethically acceptable.

- Bias and accountability in high-stake decisions
  - Hiring, banking, legal, medical decisions etc.
- Unintended, long-term influences
  - Chat bots, recommender systems etc.
  - Addiction, mental health etc.
- Fairness: the system works well for a specific group of users
- Privacy: access to (private) data generated by users
- AI safety: Potential risks of AI taking over humanity?

# Food for thought

- Should the learning objective be maximizing accuracy?
- Who is responsible and what to do when ML systems go wrong?
- How do we factor in humans when designing models?