Applied Machine Learning
W207

datascience@berkeley

Week 1 Agenda

- How this class works: orga, syllabus, etc
- Introductions - Get to know each other
- What is AI/ML?
- Intro to SciKitLearn

How this class works: orga, syllabus, etc

A word on COVID-19

Communication

- Email: dschib@berkeley.edu → for **personal** non-tech questions only
- Slack channel for this class: **#w207_schioberg** → for **code/content questions**, highest chance for quick help
- OR: te big ML channel: **#w207**
- Office hours: Wednesday 8:05 PM PST or by appointment
  - You can go to other instructors' office hours. Should be shared in ISVC
- Announcements: Slack only! **#w207_schioberg_announce** - Double check your slack notification settings. DO NOT post there

Syllabus (in Github) - general approach

- One algorithm each week, will dig deeper in some.
- Typical class outline (may vary based on topic):
    - Review async material
    - Walk through notebooks or small group work on notebooks (or both)
    - Dig deeper: Use case, examples, questions
    - Sometimes: Discuss reading/paper
- Find readings here: https://github.com/MIDS-W207/coursework/tree/master/Readings
- Find Syllabus here: https://github.com/MIDS-W207/coursework/blob/master/Schioberg/datasci-w207_syllabus.pdf

I need to give you access to the github repo!!!
Fill out the survey pinned to slack

Syllabus (in github) - details

Week 1: Welcome!
Week 2: Nearest Neighbors
Week 3: Naive bayes, Spam Classification

Week 4: Decision Trees, Bagging, Boosting
Week 5: Linear Regression, Logistic Regression.
Week 6: Gradient Descent, Regularization (Deep Learning)
Week 7: Neural Networks (Deep Learning)
Week 8: Algorithm Comparison (Deep Learning)

Week 9: K-Means
Week 10: Gaussian Mixture Models
Week 11: PCA

Week 12: Graph Analysis
Week 13: Recommender Systems

Week 14: Class presentations (project 4)

I need to give you access to the github repo!!!
Fill out the surcey pinned to slack

Projects, Grades, Rules, Hints (1)

- **ALL TIMES (due dates, office hours, etc) IN THIS CLASS ARE PST**
- Grades are based on 4 projects. See syllabus (in github, folder Schioberg) for grading scheme
- 3 individual (Jupyter) Python projects - work in groups, submit individually
- 1 group **kaggle competition** presented in the last lecture as a group. Details will follow
- Due dates: Sunday night (23:59 PST) after around weeks 5, 9, 12 (exact dates to follow).
- Late submission = -10% on the grade! General rule for all 207 sections
- Advanced ML/programmer? It might still take you longer than expected :)

Projects, Grades, Rules, Hints (2)

- **All projects available now in Github.** You can start any of them any time. (I need to add you!)
- Use Git whenever possible! You will thank yourself when your laptop decides to give up two hours before the deadline.
- Projects are graded by a TA/me by hand: We **read** and **run** everyone's **code**. Be ready to explain your setup to me so I can recreate it and see if your code actually runs.
- Upload your finished notebook to ISVC (or the link to your git repo)!
- ISVC does not allow re-uploads :(Contact me if you submitted too early.
- **COMMENT** your code! Explain your train of thoughts to me -> extra points even if code looks wonky and result is off

Your questions?

"This doesn't run… HELP!"
How to ask for help

In case of code bugs!

1. Send code as a code snippet in slack. Please do NOT screenshot your notebook/shell/script!
2. Say what you were trying to do with some details.
3. Where are you trying to run this? Local Jupyter installation, colab, GCloud etc
4. Copy paste the whole error message into a code snippet in Slack

Why all this? I want to understand what you wanted to do, reproduce the error, and give you a helpful answer quickly

No screenshots! Really!

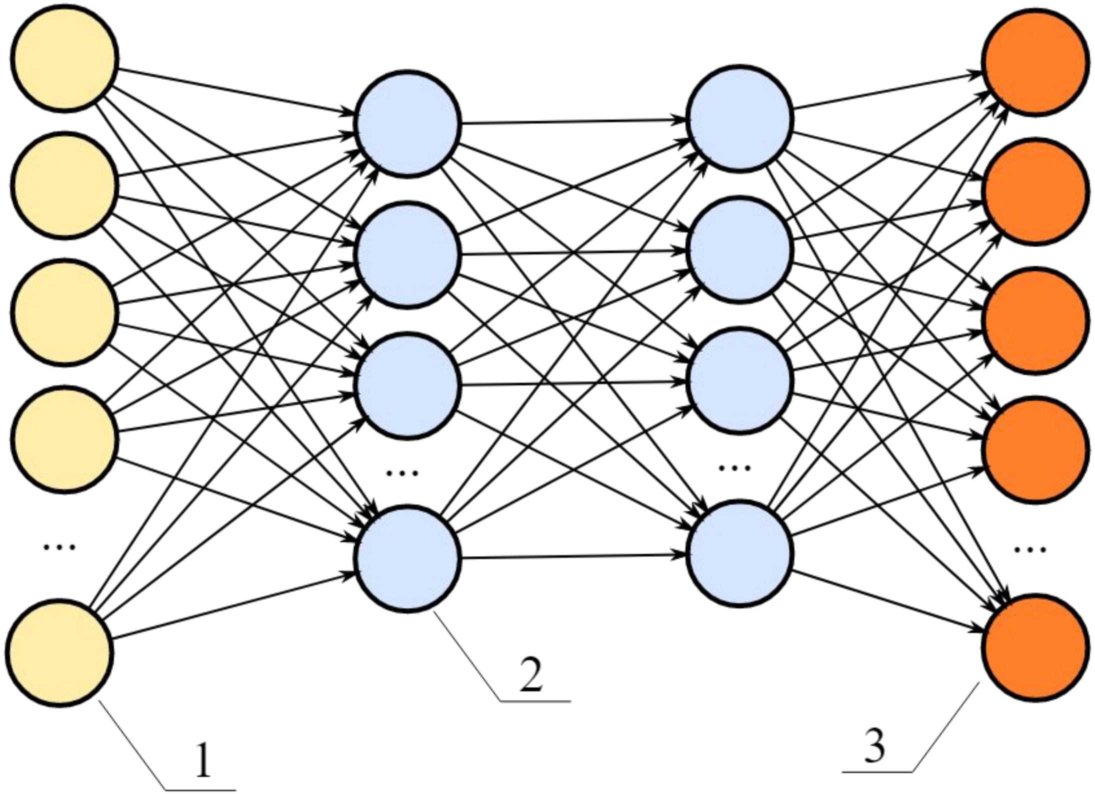
All other questions: simply explain where you are stuck! (if possible without screenshots)

Introductions

Introductions

- Tell us about yourself - your background, location, etc
- What (topic) are you most excited about in this class?
- Fun fact about yourself?

What’s Artificial Intelligence?


...
...
...
...
1
2
3

Different fields and/or names

- Artificial Intelligence
- Machine Learning
- Deep Learning
- Optimization
- Data Mining
- Statistical learning theory
- Pattern Recognition
- “Big Data”
- Natural Language processing
- Distributed computing
- GPUs vs. CPUs

What do we need for machine learning? What are the ingredients?

Supervised Learning

Where do the labels come from??

Input X:

- Document, e-mail, or social media post
- Audio
- Image
- Video
- Demographic profile; user log

Output Y:

- Topic of document
- Text of audio
- Object in image
- Action in video
- Interests with user log

Goal: Predict Y from X.

Do: Collect labeled examples (X, Y).

Getting the data

- It likely won't be clean
- Pieces of data in lots of places - talk to your data engineer from w205 ;)
- You likely can't use it as is as input for the ML algorithm:
    - Feature extraction may still be needed.
    - Representation / vectorization.
    - Quantization, etc

An example process could look like this:

Collect data ⇒ stitch it together ⇒ clean it ⇒ represent/vectorize for ML

Unsupervised learning

Unsupervised Setting

Same input (X).
No labeled output (Y).
- Supervision can be added by collecting labels for data.
- We generally want to know what we can learn from the data without labels.

Goal: Discover hidden structure in X.
- A clustering algorithm should reveal that groups of data points are separate.

Unsupervised learning

Unsupervised Methods

- Clustering
- Outlier detection
  - Have a model for expected data.
  - Look for anomalies.
- Dimensionality reduction
  - Set of features is large.
  - Important information can be expressed in a few dimensions.
- Signal separation
  - Can be used to separate sources of Data
  - Word embedding (Arguably self-supervised)

How to code in this class

- **Tutorial.ipynb** is found in github under "notebooks"
- SciKitLearn
- Let's have a look at some specifics