The ML Nitty Grittys

Details on general ML implementations

Today’s Content

- Getting your data right
- Training Data vs. Testing Data
- Overfitting
- Error Functions
- Applications of machine learning

We need a little more context before talking about Deep Learning

- What are machine learning systems trying to accomplish?
- How do we set the data up most effectively?
- What problems might arise?

Getting your data right - Preprocessing

- All machine learning systems try to learn the function

$$f(X) = Y$$

- s.t. X is a matrix of data and Y is a vector (usually)

	X_1 = Roses	X_2 = Violets	Y=Poem Quality
Sample 1	Red	Blue	Good
Sample 2	Not Red	Not Blue	Bad

Getting your data right - Preprocessing

- Our ML system here is learning **Poem Quality** as a function of **Word Choice**
  - Our data here is **Categorical**, not continuous
  - Each column in the X-set is called a **feature**
- But this data isn't good enough, we haven't preprocessed it yet.

	X_1 = Roses	X_2 = Violets	Y=Poem Quality
Sample 1	Red	Blue	Good
Sample 2	Not Red	Not Blue	Bad

Getting your data right - Preprocessing

- First, we need to bound our sets. We need to establish the space that we are working in.
  - What values can each feature take on?
  - Will each feature always appear in each sample?

	X_1 = Roses (Red/Not Red)	X_2 = Violets (Blue/Not Blue)	Y=Poem Quality (Good/Bad)
Sample 1	Red	Blue	Good
Sample 2	Not Red	Not Blue	Bad

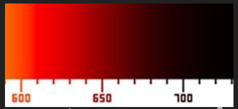
Getting your data right - Preprocessing

- Then, we introduce a numeric scale
  - S.t. it levels each variable's importance
    - [0,1] scale, normalize each column vector, etc.
  - Why is a numerical scale necessary?

	X_1 = Roses (1/0)	X_2 = Violets (1/0)	Y=Poem Quality (1/0)
Sample 1	1	1	1
Sample 2	0	0	0

Getting your data right - Preprocessing

- Numeric scales are useful when
  - We are working with continuous functions
    - Roses are red...how red?



  - We want to graphically represent our data
  - We want to see patterns in the data easier by eye
- If you have categorical data, choose a numerical scale so that it makes sense!
  - Lower class = 0, upper class = 1, middle class = 2 doesn't make sense

Getting your data right - Preprocessing

- Ex: bird sanctuary website records data on user page visits
- **Feature Selection**
  - More data = Longer run time
  - Some data may be superfluous
- **Decomposition**
  - What if time is recorded in seconds but the patterns are hourly?
- **Aggregation**
  - What if you get a list of all login attempts but don't care about them as single data points?

Why do we pre-process the data?

- As we’ll see, ML systems only learn to make decisions by minimizing their error from data

Why do we pre-process the data?

- As we’ll see, ML systems only learn to make decisions by minimizing their error from data
- Inconsistent data = Inconsistent decisions
- Poorly scaled data = Poorly scaled decisions

Why do we pre-process the data?

- As we'll see, ML systems only learn to make decisions by minimizing their error from data
- Inconsistent data = Inconsistent decisions
- Poorly scaled data = Poorly scaled decisions
- Good data = Good decisions?

Training Data vs. Testing Data

- Once our data is processed, we need to split it into training data vs. testing data
  - ???????
  - ???????

Training Data vs. Testing Data

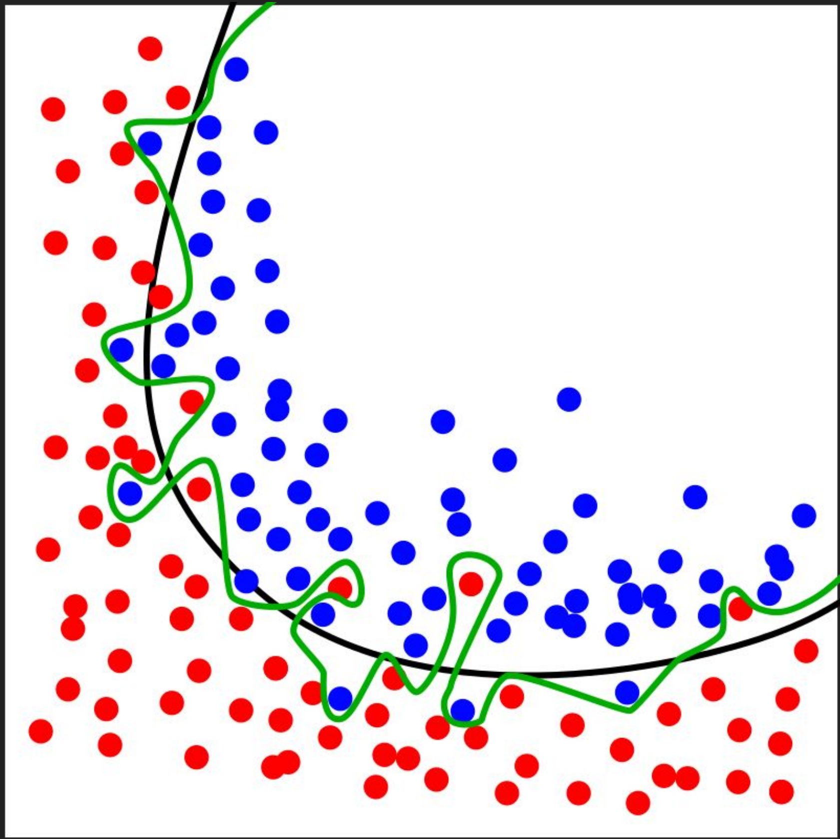
- Once our data is set up, we need to split it into training data vs. testing data
- Training Data - What your machine learning system learns on
- Testing Data - What your machine learning system checks its learning on
- Why do we need this?

Overfitting

- Any ML system conforms itself as best it can to your exact data
- In a perfect world, they will learn to generalize
- ML systems instead overfit to whatever data they are given
  - We did this last week with the buses

Overfitting

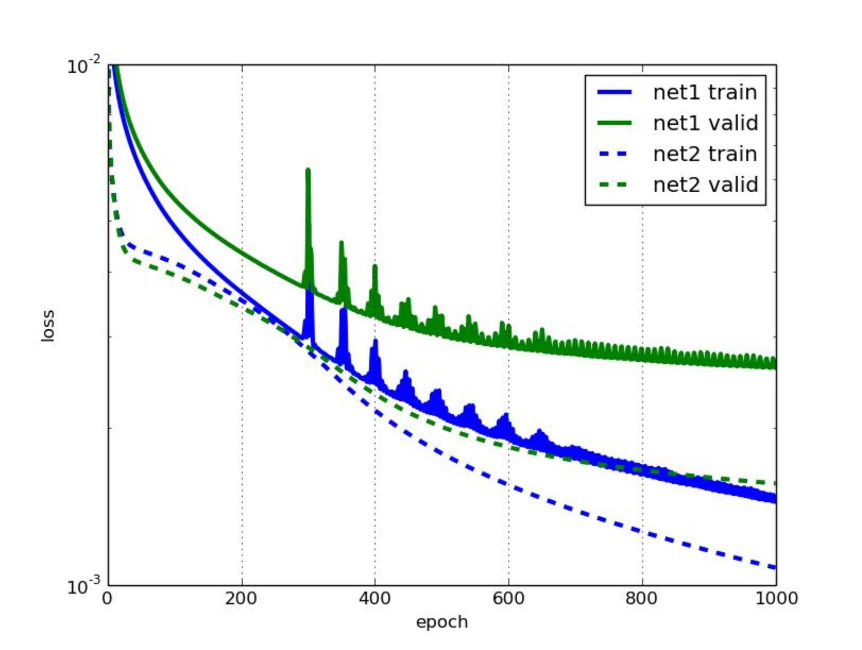
- More complex ML systems are more likely to overfit
- Less complex ML systems may not have the means to learn
- Training data that repeats specifics is more likely to be overfit on
  - Roses = red ➡ Good Poem every time in training set?



Back to Training vs. Testing

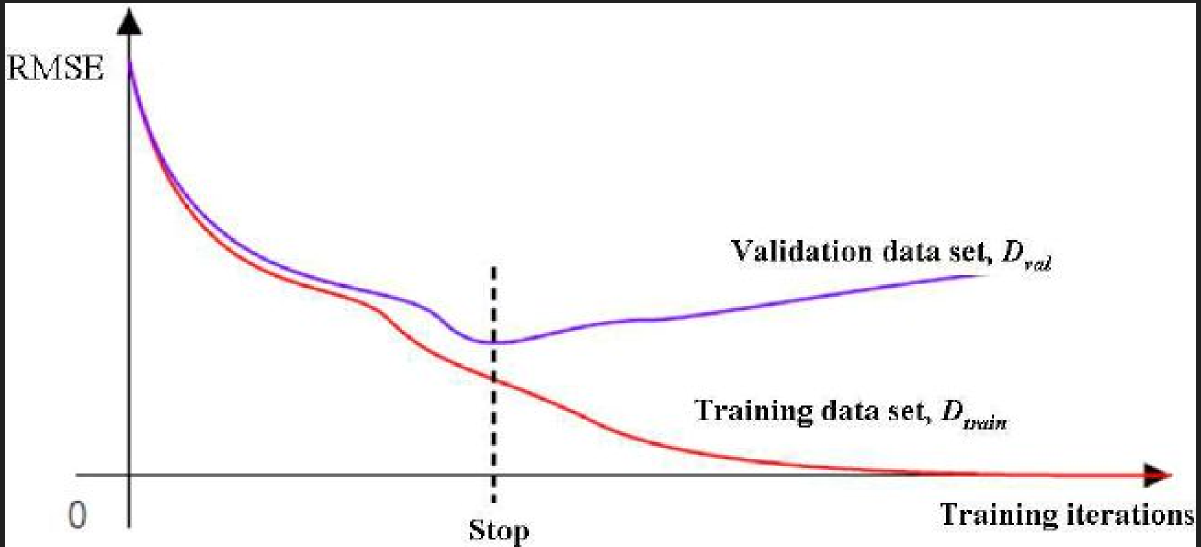
- So we train on one set of data **hundreds** of times over
- Hope that this will carry over to test data (which is a separate set!)

- Overfit == Something's wrong?
- No overfit == Everything's fine?



Early Stopping to avoid overfitting

- Often, the overfit will end up **worsening** the test results with enough iterations on the training data
- We'll talk about deep learning strategies for this later



Okay, but how does an ML system actually learn?

- Error functions!
  - Aka cost functions, loss functions
- We want to minimize the following I s.t.
  - f is our machine learning system
  - V is the error function
  - p is the probability distribution of the data

$$I[f] = \int_{X\times Y} V(f(\vec{x}), y)p(\vec{x}, y)\, d\vec{x}\, dy$$

Error Functions (in theory)

- Let’s do an example error analysis given a **classification task**
  - X is the vector space of all possible inputs
  - Y is only from set [-1, 1]
  - Our error function V is the 0-1 Indicator Function
- Remember, we want to minimize I

$$I[f] = \int_{X \times Y} V(f(\vec{x}), y) p(\vec{x}, y) \, d\vec{x} \, dy$$

So that's our I, how do we minimize it?

- We take the gradient of I (usually of V actually) and go towards the local minimum
  - We take the gradient of V rather than I since we may not know p
- More on this in the next lecture

$$I[f] = \int_{X \times Y} V(f(\vec{x}), y)p(\vec{x}, y)\, d\vec{x}\, dy$$

Back to the real world

- Hopefully you can see how powerful this is
- We can minimize our prediction error on *any* type of data
  - Language can be made into data
  - Videos are data
  - Your logins are data
  - Motion is data
  - Blood tests are data
- Our error equation is **general**
  - Our data can be anything and these techniques will work

Machine Learning Applications today

- Self driving car
- Recommendations (Netflix, Youtube, etc.)
- Weather
- Fraud detection
- Looking at X-Rays
- Autonomous grocery store
- Crime prediction
- Autonomous military vehicles
- Automated investing (every thousandth of a second)