Today: Efficient model serving.

→ Compression, Relational data (embeddings)

Prediction Serving

Deployment

Data → $f(\cdot)$ Model

Server $f(\cdot)$

Users Predictions

Typically Same

Serving Systems

Models: as a black box
(reuse the same code as training)

Standard tricks for efficient serving
for user requests.

Problem: Agnostic to ML

Focus only on prediction serving problem

1) User requests
   - batching
   - result caching
2) Increase user capacity
   - replicate the model (parallel evaluation)

External optimization

Q: What is the most expensive operation when serving machine learning models?
Model computation →
(sentence) Input → Featurization
(data point) from user (tokenize char n-gram)

prediction

LR layer (softmax)

Task: sentiment analysis: (🙂 or 😞)

Model 1: Logistic Regression
Model 2: LSTM (RNN)

Scenario: Multiple requests from users

Featureization: split tokens

Q: if we consider these ops as a black box?
You have two sentences that share 80% of the same words.

Saving 1: remove redundant lookups. (thick)

Batch → cache my ngrams and compute them once.

Opportunities for sharing resources if I open the black box?
Instead of having this setup feat.

How do you allocate CPUs to the models?

We need to open the model black box and look.
into individual components and processing?

→ No all input are equally hard.

Model Cascades ↔ Boosting

weak learners (p > 0.5)

Combine weak learners into an ensemble

the ensemble is a strong learner (p > \(1 - \varepsilon\))
A vertical line is a model that uses only 1 out of two features.

Weak learners are much cheaper to evaluate because they rely on fewer features.
Cascading classifiers

Viola/Jones Face Detector

Goal: real-time face detection

Start with simple classifiers that reject many on the negative sub-windows while they detect almost all positive sub-windows.
Result from the first classifier -> trigger anomaly from the second classifier.

Sub-window $\rightarrow$ class 1 $\rightarrow$ class 2 $\rightarrow$ class 3

class 1 $\bullet$
class 2 $\bullet$
class 3 $\bullet$

The negative outcome can come out at any point but the positive outcome has to go through all classifiers $\rightarrow$ class 1 $\rightarrow$ class 2 $\rightarrow$ class 3

Is Face? $\rightarrow$ False $\rightarrow$ Non-face $\rightarrow$ False $\rightarrow$ Non-face $\rightarrow$ False $\rightarrow$ Non-face
A classifier is "optimal" if it minimizes false negatives.
For each classifier, our training goal is to minimize false negatives instead of total prediction error.

Each classifier has a different training set (TR). What is the TR of class 2?

Class 1

\[ F \downarrow \text{ (goal min. Negative false negatives)} \]

TR \rightarrow Class 1

Initial.
This training is expensive because I change the TR set while I propagate data through the cascade.

Follow up questions:
- How do I learn the best cascade so that
I minimize latency → resource usage
maximize accuracy.

Willump (MLsys 20)

Statistical optimization to correlate features with different classes

Technical Idea:

Initial featureization Pipeline (Input to Willump)

- Compute all Features
  → Model
  → Prediction

- Use feature selection
  → Compute Features
  → Selected

⇒ Approximate model

Confidence ≥ 2

Y → Predicted

N → Compute remaining original features

Predicted

repeated.