COS324: Introduction to Machine Learning

Prof. Elad Hazan & Prof. Yoram Singer

Near term agenda

- Administrative details
- Course outline, goals, topics
- Review of concepts in probability (recitation)
- Introduction to introduction

COS 324: Administrative details

- Teaching assistants:
  Wei Hu, Nikunj Saunshi, Karan Singh, Cyril Zhang, Yi Zhang

- Course webpage: `http://iml.cs.princeton.edu`

- Coding exercises: solely in Python

- Due in class one week after announcement. 5 late days.

- Whiteboard use, slides, written material

Course workload and expectations

- Rigorous treatment of elements of machine learning
- Formal-theoretical approach and some proofs
- Algorithmic perspective
- Programming exercises:
  aesthetic & functionality over PL-formalism
- Roughly equally division between programming, applications, and theoretical exercises and exam

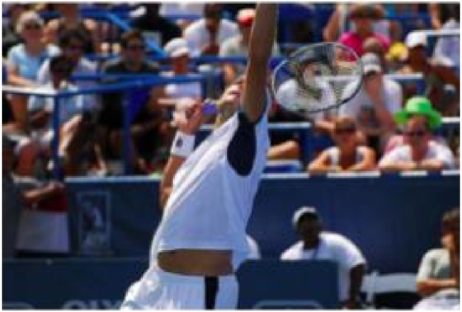
Grade structure

- Final exam: 40%
- Exercises: 40%
- Midterm exam: 20%
- Class attendance is mandatory
- Bonus questions in class (A+ grade)
- Last lecture: movie & discussion on AI (TBA)

Machine learning for image captioning

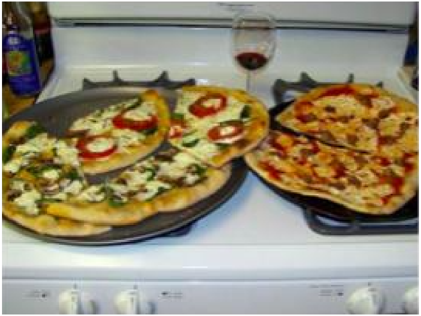
- Input: images containing multiple objects
- Output: written description of objects & relations
- Uses:
  - automatically generated subtitles
  - visualization aid for sight impaired
- ML:
  - Collect many pairs of images and captions
  - “Learn” a mapping from image-space to sequences of characters

Automatic image captioning: example I



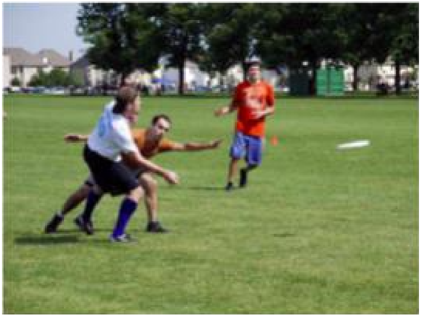
A man holding a tennis racquet on a tennis court.

Automatic image captioning: real example II



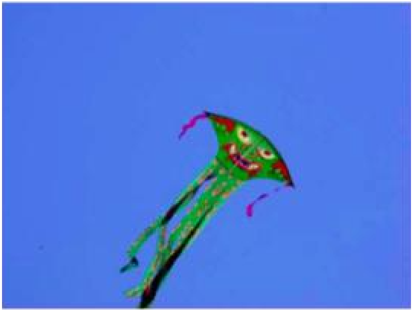
Two pizzas sitting on top of a stove top.

Automatic image captioning: real example III



A group of young people playing a game of Frisbee.

Automatic image captioning: real example IV



A man flying through the air while riding a snowboard.

Show, Attend and Tell: Neural Image Caption Generation
Xu, Ba, Kiros, Cho, Courville, Salakhutdinov, Zemel, Y. Bengio

Where is machine learning used?

Eclectically devised list. . .

- Genomics, reconstruction of 3D folding of proteins
- Finance, media's sentiment to predict markets' trend
- Autonomous vehicles: cars, drones, legged robots
- Natural language processing, translation, dialogue systems
- Fraud detection, network security
- Automatic surveillance systems
- Healthcare, early detection of disease outbreaks
- Ranking and filtering, Web search, web & email spam
- Virtual reality and gaming
- Brain-computer interfaces, brain-cntl typing, artificial retina
- Grading of COS324 . . .

Learning from experts' advice

- Run a NY tour company
- Need to know whether it'll rain next day
- Consult with "local" experts who are supposedly good

No

Yes!

No

Yes

Yes

…

Learning from experts' advice

Eve of next tour day . . .

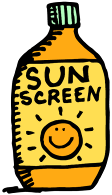
XX

Yes!

XX

Yes

No



•••

After two rounds

One expert is left to consult with . . .

. . . and she seems to be always correct

Is it a coincidence?

Notation

- Scalars $a, b, c, \cdots, i, j, k, \cdots$ (lowercase)

- Vectors $\mathbf{w}, \mathbf{u}, \mathbf{v}, \cdots$ (boldface)

- Inner products $\mathbf{w} \cdot \mathbf{u} = \sum_{j=1}^{n} w_j u_j$

- Matrices $A, B, C, \cdots$ (uppercase)

- Indicator operator $\mathbb{1}[\pi]$

$$\mathbb{1}[\texttt{false}] = 0 \;;\; \mathbb{1}[\texttt{true}] = 1$$

$$\mathbb{1}[x \leq |x|] = 1$$

$$\mathbb{1}[\forall x : log(x) > 0] = 0$$

Ingredients

- Input space $\mathcal{X} = \{-1, +1\}^n$
- Instance, input, point $\mathbf{x} \in \mathcal{X}$
- Target space $\mathcal{Y} = \{-1, +1\}$
- Target, label $y \in \mathcal{Y}$
- Example, instance-label pair $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$

Prediction

- Predictor, classifier, hypothesis $h : \mathcal{X} \to \mathcal{Y}$
- For now assume that,

$$h(\mathbf{x}; \mathbf{w}) = \text{sign}(\mathbf{w} \cdot \mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} w_j x_j\right)$$

where

$$\text{sign}(z) = \begin{cases} +1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

and $\mathbf{w}$ is an n-dimensional weight vector s.t. $w_j \geq 0$

Online classification-learning

- Initialize $\mathbf{w}^1$; $\mathcal{L}^1 = 0$

- For $t = 1, 2, \ldots, T, \ldots$

  1. Predict $\hat{y}^t = \mathrm{sign}(\mathbf{w}^t \cdot \mathbf{x}^t)$

  2. Observe true outcome y^t

  3. Endure loss: $\ell^t = \mathbb{1}[\hat{y}^t \neq y^t] \quad ; \quad \mathcal{L}^{t+1} = \mathcal{L}^t + \ell^t$

  4. Update $\mathbf{w}^{t+1} := F(\mathbf{w}^t, \mathbf{x}^t, y^t)$

Remarks:

- We can work in-place and use a single vector $\mathbf{w}$ for all t
- F is called update-rule or learning-rule

The Consistent Algorithm

In words: Pick an expert. So long as it makes correct predictions, stay put. Otherwise, discard and replace with un-discarded expert.

- Initialize $\mathbf{w}^1 = (1, \ldots, 0)$
- Add a "0-coordinate" $\mathbf{w}^1 \Rightarrow (0, 1, 0, \ldots, 0)$
- For $t = 1, 2, \ldots, T, \ldots$

  1. Predict $\hat{y}^t = \mathrm{sign}(\mathbf{w}^t \cdot \mathbf{x}^t)$
  2. Observe true outcome y^t
  3. Endure loss $\ell^t = \mathbb{1}[\hat{y}^t \neq y^t]$
  4. Update for $j = 1, \ldots, n$:

$$w_j^{t+1} := (1 - \ell^t) w_j^t + \ell^t w_{j-1}^t$$

The Halving Algorithm

In words: Take a majority vote of consistent experts. Discard all inconsistent experts on each round.

- Initialize $\mathbf{w}^1 = (1, 1, \dots, 1)$
- For $t = 1, 2, \dots, T, \dots$
  1. Predict $\hat{y}^t = \text{sign}(\mathbf{w}^t \cdot \mathbf{x}^t)$
  2. Observe true outcome y^t
  3. Endure loss $\ell^t = \mathbb{1}[[] \, \hat{y}^t \neq y^t]$
  4. Update $w_j^{t+1} := w_j^t(1 - (|x_j^t - y^t|)/2)$

Simple Analysis

Claim

Halving is going to make at most $\lfloor \log_2(n) \rfloor$ mistakes.

Proof

Whenever Halving makes a prediction error on round t, at least half of the experts made the wrong prediction.

Therefore, on the next round at most half the number of experts from the previous round are still being consulted with.

Rounds without a prediction mistake may or may not reduce the number of consistent experts.