"Equations are just the boring part of mathematics. I attempt to see things in terms of geometry."
—Stephen Hawking

Week 2: Linear Classification: Perceptron

Professor Liang Huang

some slides from Alex Smola (CMU/Amazon)
Roadmap for Weeks 2-3

- Week 2: Linear Classifier and Perceptron
  - Part I: Brief History of the Perceptron
  - Part II: Linear Classifier and Geometry (testing time)
  - Part III: Perceptron Learning Algorithm (training time)
  - Part IV: Convergence Theorem and Geometric Proof
  - Part V: Limitations of Linear Classifiers, Non-Linearity, and Feature Maps

- Week 3: Extensions of Perceptron and Practical Issues
  - Part I: My Perceptron Demo in Python
  - Part II: Voted and Averaged Perceptrons
  - Part III: MIRA and Aggressive MIRA
  - Part IV: Practical Issues and HW1
  - Part V: Perceptron vs. Logistic Regression (hard vs. soft); Gradient Descent
Brief History of the Perceptron
Perceptron
(1959-now)

Frank Rosenblatt
deep learning
~1986; 2006-now

multilayer perceptron

logistic regression
1958

perceptron
1959

SVM
1964; 1995

kernels
1964

cond. random fields
2001

structured perceptron
2002

structured SVM
2003
Neurons

- **Soma (CPU)**
  Cell body - combines signals

- **Dendrite (input bus)**
  Combines the inputs from several other nerve cells

- **Synapse (interface)**
  Interface and parameter store between neurons

- **Axon (output cable)**
  May be up to 1m long and will transport the activation signal to neurons at different locations
Frank Rosenblatt’s Perceptron
Multilayer Perceptron (Neural Net)
**Brief History of Perceptron**

- **1959**: Rosenblatt invention
- **1962**: Novikoff proof
- **1969**: Minsky/Papert book killed it
- **1999**: Freund/Schapire voted/avg: revived
- **2002**: Collins structured
- **2003**: Crammer/Singer MIRA
- **2005**: McDonald/Crammer/Pereira structured MIRA
- **2006**: Singer group aggressive
- **2007--2010**: Singer group Pegasos
- **1997**: Cortes/Vapnik SVM
- **1999**: McConkey/Crammer/Pereira structured MIRA

**Online vs. Batch**
- Online: conservative updates, inseparable case
- Batch: max margin, kernels, soft-margin

**AT&T Research**
- ex-AT&T and students

*mentioned in lectures but optional (others papers all covered in detail)*
Part II

- Linear Classifier and Geometry (testing time)
  - decision boundary and normal vector $\mathbf{w}$
  - not separable through the origin: add bias $b$
  - geometric review of linear algebra
  - augmented space (no explicit bias; implicit as $w_0=b$)

Test Time

Input $\mathbf{x}$  \rightarrow  \text{Linear Classifier}  \rightarrow  \text{Prediction } \sigma(\mathbf{w} \cdot \mathbf{x})

Model $\mathbf{w}$

Training Time

Input $\mathbf{x}$  \rightarrow  \text{Perceptron Learner}  \rightarrow  \text{Model } \mathbf{w}

Output $y$
Linear Classifier and Geometry

Linear classifiers: perceptron, logistic regression, (linear) SVMs, etc.

\[ f(x) = (w \cdot x) \]

weights

weights \( w \): “prototype” of positive examples

it’s also the normal vector of the decision boundary

meaning of \( w \cdot x \): agreement with positive direction

test: input: \( x, w \); output: 1 if \( w \cdot x > 0 \) else -1

training: input: \((x, y)\) pairs; output: \( w \)

separating hyperplane (decision boundary)

\[ w \cdot x = 0 \]
What if not separable through origin?

solution: add bias $b$

weights

output $f(x) = \sigma(w \cdot x + b)$

negative $w \cdot x + b < 0$

positive $w \cdot x + b > 0$

$||x|| \cos \theta = \frac{w \cdot x}{||w||}$
Geometric Review of Linear Algebra

**line in 2D**

\[ w_1 x_1 + w_2 x_2 + b = 0 \]

\[ \frac{|b|}{\|w_1, w_2\|} \]

\( (x_1^*, x_2^*) \)

\( (w_1, w_2) \)

**\((n-1)\)-dim hyperplane in \(n\)-dim**

\[ w \cdot x + b = 0 \]

\[ \frac{|w \cdot x + b|}{\|w\|} \]

**point-to-line distance**

\[ \frac{|w_1 x_1^* + w_2 x_2^* + b|}{\sqrt{w_1^2 + w_2^2}} \]

\[ \frac{|(w_1, w_2) \cdot (x_1, x_2) + b|}{\|w_1, w_2\|} \]

**point-to-hyperplane distance**

\[ \frac{|w \cdot x + b|}{\|w\|} \]

required: algebraic and geometric meanings of dot product

[link to PDF](http://classes.engr.oregonstate.edu/eecs/fall2017/cs534/extra/LA-geometry.pdf)
Augmented Space: dimensionality+1

weights

output

explicit bias

\[ f(x) = \sigma(w \cdot x + b) \]

augmented space

\[ f(x) = \sigma((b; w) \cdot (1; x)) \]

can’t separate in 1D from the origin

can separate in 2D from the origin

weights

output

x_0 = 1 \quad x_1 \quad x_2 \quad x_3 \quad \cdots \quad x_n

w_0 = 0 \quad w_1 \quad w_2 \quad w_3 \quad \cdots \quad w_n

Output

1

0
Augmented Space: dimensionality+1

$$f(x) = \sigma(w \cdot x + b)$$

weights

output

can’t separate in 2D from the origin

explicit bias

weights

output

can separate in 3D from the origin

augmented space

$$f(x) = \sigma((b; w) \cdot (1; x))$$
Part III

- The Perceptron Learning Algorithm (training time)
  - the version without bias (augmented space)
  - side note on mathematical notations
  - mini-demo

\[
\sigma(w \cdot x)
\]

![Diagram of Perceptron Learning Algorithm](attachment:image.png)
The Perceptron Algorithm

**input:** training data $D$
**output:** weights $w$
**initialize** $w \leftarrow 0$
**while** not converged
  **for** $(x, y) \in D$
    **if** $y(w \cdot x) \leq 0$
      $w \leftarrow w + yx$

- the simplest machine learning algorithm
- keep cycling through the training data
  - update $w$ if there is a mistake on example $(x, y)$
- until all examples are classified correctly
Side Note on Mathematical Notations

- I’ll try my best to be consistent in notations
- e.g., bold-face for vectors, italic for scalars, etc.
- avoid unnecessary superscripts and subscripts by using a “Pythonic” rather than a “C” notational style
- most textbooks have consistent but bad notations

initialize \( w \leftarrow 0 \)
while not converged
  for \((x, y) \in D\)
    if \( y(w \cdot x) \leq 0 \)
      \( w \leftarrow w + yx \)
  good notations:
  consistent, Pythonic style

initialize \( w = 0 \) and \( b = 0 \)
repeat
  if \( y_i [\langle w, x_i \rangle + b] \leq 0 \) then
    \( w \leftarrow w + y_i x_i \) and \( b \leftarrow b + y_i \)
  end if
until all classified correctly
bad notations:
inconsistent, unnecessary \( i \) and \( b \)
Demo

while not converged
for (x, y) ∈ D
  if y(w · x) ≤ 0
    w ← w + yx
Demo

\[
\text{while not converged} \\
\text{for } (x, y) \in D \\
\text{if } y(w \cdot x) \leq 0 \\
w \leftarrow w + yx
\]
Demo

\[
\begin{align*}
\text{while not converged} & \\
\text{for } (x, y) \in D & \\
\text{if } y(w \cdot x) \leq 0 & \\
w & \leftarrow w + yx
\end{align*}
\]
while not converged

for \((x, y) \in D\)

if \(y(w \cdot x) \leq 0\)

\[ w \leftarrow w + yx \]
while not converged

for $(x, y) \in D$

if $y(w \cdot x) \leq 0$

$w \leftarrow w + yx$
while not converged

for \((x, y) \in D\)

if \(y(w \cdot x) \leq 0\)

\[w \leftarrow w + yx\]
while not converged
for \((x, y) \in D\)
if \(y(w \cdot x) \leq 0\)
\[
w \leftarrow w + yx
\]
Part IV

- Linear Separation, Convergence Theorem and Proof
  - formal definition of linear separation
  - perceptron convergence theorem
  - geometric proof
  - what variables affect convergence bound?
Linear Separation; Convergence Theorem

- dataset $D$ is said to be “linearly separable” if there exists some unit oracle vector $u: ||u|| = 1$ which correctly classifies every example $(x, y)$ with a margin at least $\delta$:
  \[ y(u \cdot x) \geq \delta \text{ for all } (x, y) \in D \]

- then the perceptron must converge to a linear separator after at most $R^2/\delta^2$ mistakes (updates) where
  \[ R = \max_{(x, y) \in D} ||x|| \]

- convergence rate $R^2/\delta^2$

  - Dimensionality independent
  - Dataset size independent
  - Order independent (but order matters in output)
  - Scales with ‘difficulty’ of problem
part 1: progress (alignment) on oracle projection

assume \( w^{(0)} = 0 \), and \( w^{(i)} \) is the weight before the \( i \)th update (on \( (x, y) \))

\[
\begin{align*}
  w^{(i+1)} &= w^{(i)} + yx \\
  u \cdot w^{(i+1)} &= u \cdot w^{(i)} + y(u \cdot x) \\
  u \cdot w^{(i+1)} &\geq u \cdot w^{(i)} + \delta \\
  u \cdot w^{(i+1)} &\geq i\delta
\end{align*}
\]

projection on \( u \) increases!

(more agreement w/ oracle direction)

\[
\|w^{(i+1)}\| = \|u\| \|w^{(i+1)}\| \geq u \cdot w^{(i+1)} \geq i\delta
\]
part 2: upperbound of the norm of the weight vector

\[ \| \mathbf{w}^{(i+1)} \|^2 = \| \mathbf{w}^{(i)} + y \mathbf{x} \|^2 \]

\[ = \| \mathbf{w}^{(i)} \|^2 + \| \mathbf{x} \|^2 + 2y(\mathbf{w}^{(i)} \cdot \mathbf{x}) \]

\[ \leq \| \mathbf{w}^{(i)} \|^2 + R^2 \]

\[ \leq iR^2 \]

\[ R = \max_{(\mathbf{x},y) \in D} \| \mathbf{x} \| \]

\[ \theta \geq 90^\circ \]

\[ \cos \theta \leq 0 \]

\[ \mathbf{w}^{(i)} \cdot \mathbf{x} \leq 0 \]
Geometric Proof, part 2

• part 2: upper bound of the norm of the weight vector

\[ \mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} + y\mathbf{x} \]

\[ \left\| \mathbf{w}^{(i+1)} \right\|^2 = \left\| \mathbf{w}^{(i)} + y\mathbf{x} \right\|^2 \]

\[ = \left\| \mathbf{w}^{(i)} \right\|^2 + \left\| \mathbf{x} \right\|^2 + 2y(\mathbf{w}^{(i)} \cdot \mathbf{x}) \]

\[ \leq \left\| \mathbf{w}^{(i)} \right\|^2 + R^2 \]

\[ \leq iR^2 \]

Combine with part 1:

\[ \left\| \mathbf{w}^{(i+1)} \right\| = \left\| \mathbf{u} \right\| \left\| \mathbf{w}^{(i+1)} \right\| \geq \mathbf{u} \cdot \mathbf{w}^{(i+1)} \geq i\delta \]

\[ i \leq R^2 / \delta^2 \]
Convergence Bound

- is independent of:
  - dimensionality
  - number of examples
  - order of examples
  - constant learning rate
- and is dependent of:
  - separation difficulty
  - initial weight $w^{(0)}$
    - changes how fast it converges, but not whether it’ll converge
  - feature scale

$R^2 / \delta^2$
• Limitations of Linear Classifiers and Feature Maps
  • XOR: not linearly separable
  • perceptron cycling theorem
  • solving XOR: non-linear feature map
  • “preview demo”: SVM with non-linear kernel
  • redefining “linear” separation under feature map
XOR

- XOR - not linearly separable
- Nonlinear separation is trivial
- Caveat from “Perceptrons” (Minsky & Papert, 1969) Finding the minimum error linear separator is NP hard (this killed Neural Networks in the 70s).
Brief History of Perceptron

1959
Rosenblatt invention

1962
Novikoff proof

1969*
Minsky/Papert book killed it

1997
Cortes/Vapnik SVM

1999
Freund/Schapire voted/avg: revived

2002
Collins structured

2003
Crammer/Singer MIRA

2005*
McDonald/Crammer/Pereira structured MIRA

2006
Singer group aggressive

2007--2010*
Singer group Pegasos

DEAD

*mentioned in lectures but optional
(others papers all covered in detail)

batch

minibatch

online

+max margin
+kernels
+soft-margin

1997

Cortes/Vapnik SVM

online approx.
max margin

subgradient descent

2007--2010*
Singer group Pegasos

minibatch

2003
Crammer/Singer MIRA

2006
Singer group aggressive

2005*
McDonald/Crammer/Pereira structured MIRA

batch

online

conservative updates

inseparable case

AT&T Research

ex-AT&T and students
What if data is not separable

- in practice, data is almost always inseparable
  - wait, what exactly does that mean?
- perceptron cycling theorem (1970)
  - weights will remain bounded and will not diverge
- use dev set for early stopping (prevents overfitting)
- non-linearity (inseparable in low-dim => separable in high-dim)
  - higher-order features by combining atomic ones (cf. XOR)
  - a more systematic way: kernels (more details in week 5)

---

ON THE BOUNDEDNESS OF AN ITERATIVE PROCEDURE FOR SOLVING A SYSTEM OF LINEAR INEQUALITIES

H. D. BLOCK AND S. A. LEVIN
Solving XOR: Non-Linear Feature Map

- XOR not linearly separable
- Mapping into 3D makes it easily linearly separable
  - this mapping is actually non-linear (quadratic feature $x_1x_2$)
  - a special case of “polynomial kernels” (week 5)
- Linear decision boundary in 3D => non-linear boundaries in 2D
Low-dimension <=> High-dimension
Low-dimension $\iff$ High-dimension

not linearly separable in 2D
Low-dimension $\iff$ High-dimension

not linearly separable in 2D $\implies$ linearly separable in 3D
Low-dimension $\iff$ High-dimension

not linearly separable in 2D $\quad\longrightarrow\quad$ linearly separable in 3D

linear decision boundary in 3D
Low-dimension $\iff$ High-dimension

- not linearly separable in 2D $\iff$ linearly separable in 3D
- non-linear boundaries in 2D $\iff$ linear decision boundary in 3D
SVM with a polynomial Kernel visualization

Created by:
Udi Aharoni
we have to redefine separation and convergence theorem

dataset $D$ is said to be **linearly separable under feature map** $\phi$ if there exists some unit oracle vector $u$: $||u|| = 1$ which correctly classifies every example $(x, y)$ with a margin at least $\delta$:

$$y(u \cdot \Phi(x)) \geq \delta \text{ for all } (x, y) \in D$$

then the perceptron must converge to a linear separator after at most $R^2/\delta^2$ mistakes (updates) where

$$R = \max_{(x, y) \in D} ||\Phi(x)||$$

in practice, the choice of feature map ("feature engineering") is often more important than the choice of learning algorithms

- the first step of any machine learning project is data preprocessing: transform each $(x, y)$ to $(\Phi(x), y)$

- at testing time, also transform each $x$ to $\Phi(x)$

- deep learning aims to automate feature engineering