Perceptron Learning: Ch 19.1-19.3

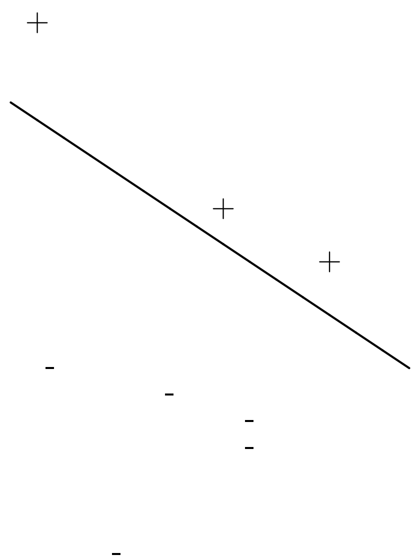
Goals

- How to Represent?
- How to Use?
- How to Learn?
- What can they do?
- Linearly separable vectors

Motivation

- Model of Neuron: Rosenblatt (58)
- Minsky & Papert (69): formal capability
- Brain has 10^{11} neurons with 10^{3} synapses per neuron
- Brain gives rise to mind.
- Model of neuron simple: add inputs, fire if greater than a threshold
- Model of learning simple (Hebbian): change with use.

Linearly Separable Concept



Basics

- Representation
- input: inhibitory and excitory attributes
- prediction: real or boolean
- output: combination of inputs via
  step (LTU), sign, or sigmoid.
- Processing
  - Let I_i be input features with weights W_i
  - Perceptron fires if $\sum(W_i * I_i) > threshold$
  - Trick: add feature $I_{(n+1)}$ with value = -1
  - With this notion, neuron fires if $(W \cdot I)$ >0.
  - The weight $-W_{n+1}$ is the old threshold.
  - Now we can use the same procedure to learn the threshold as well as the weights.

Vectors review

- $< 12.3, 32.1, -14.0 >$ is an example of a 3-dimension vector.
- Let V, V1, and V2 be vectors of size n.
- Vector addition: V=V1+V2 means V[i]=V1[i]+V2[i] for each i.
- Scalar product: If r is a real number, V = r*V1 means V[i]=r*V1[i] for each i.
- Vector dot product: if V1 and V2 are vectors of size n, then

$$V1 \cdot V2 = \sum_{i=1}^{i=n} (V1_i * V2_i)$$

- If $V1 \cdot V2 = 0$ then V1 and V2 are perpendicular.
- If $V1 \cdot V1$ is the square of length of V1.
  Check: if $V =< x, y, z >$ then $V \cdot V$ is?.
- Usual rules of algebra, e.g.

$$(V + V1) \cdot V2 = V \cdot V2 + V1 \cdot V2$$

  etc.
- Geometrically, $V1 \cdot V2$ is length of projection of V1 onto V2 (or vice versa).

More Basics

Learning

- incremental (sort of)
- epoch/cycle : single pass through data
- typically pass throught same data several times.
- fuzzy (can derive typicality value)
- failure-driven
- When to stop cycling?

Algorithm: Simplest Version

- Let W be weight vector of perceptron.
- Let I be vector for instance (with the magic 1)
- Repeat for each instance I,
  - If $W \cdot I > 0$ and I positive, do nothing.
  - If $W \cdot I < 0$ and I negative, do nothing.
  - If $W \cdot I > 0$ and I neg, W = W - I.
  - If $W \cdot I < 0$ and I pos, W = W + I.
- Until W stops changing

More general version

- When neuron wrong, fix weights
- If neuron should fire and doesn't,
- let W' be $W + \alpha x$. Note $W' \cdot I = W \cdot I + \alpha I \cdot I$.
- By choosing α can ensure that neuron fires.
- If neuron should not fire and does, let W' be $W - \alpha I$.
- Weights can grow very large
- A usual value for α is 0.1, but perhaps it should vary with n.

Extensions

- knowledge → features
- Gaussian nodes (radial basis functions)
- add new features, eg. $I_i * I_j$ and treat as LTU
- move faster, momemtum term or doubling/halving

Properties

- A set of points is linearly separable if there is a line (hyper-plane) which separates them.
- Theorem: Perceptron learning is guaranteed to work iff the concept is linear separable
- Perceptron may give a useful approximation even if concept is not linearly separable.
- If not linear separable, hyperplane keeps moving.
- Value of α determines how much you believe new instance relative to old values.
- Slow: cycle or epoch is complete pass of all data
- Incremental as opposed to batch

Strengths

- And of $I_1, ...I_n$ is realized by $W_i = 1$, threshold = n
- Or of $I_1, ...I_n$ is realized by $W_i = 1$, threshold = 1
- K-of-N of $I_1, ...I_n$ is realized by $W_i = 1$, threshold = k
- Natural concepts may be of k-of-n variety
- Prototype theory

Performance

- guaranteed convergence
- perfect characterization
- often best accuracy
- Effective procedure, i.e. can tell when to stop, but it's a long long time.
- Effective on some medical domains
  if patient has 5 of following 7 conditions then ...