Perceptron.

By Frank Ronsenblatt, Cornell, 1957.

"Linear classifier", one of the first ANN.

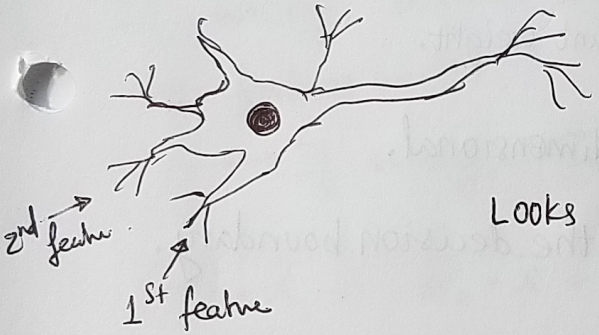
Linear classification:-

Feature vector $(\bar{x}_1, \ldots, \bar{x}_k)$, for each $\bar{x}$ (~~fe~~).

EX/Trng:- $(\bar{x}(1), f(\bar{x}(1))), \ldots$

↓ a k-dimensional vector.

Inspired by biology (very simplified).



Get a bunch of inputs.

if higher than a threshold, activate.

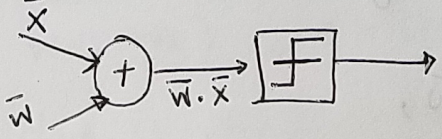
Looks like a thresholding circuit.

① Input:- feature values $(\bar{x}_1, \ldots, \bar{x}_k)$.

② Each feature has a weight. $w_j \leftrightarrow$ feature j's weight.

③ Sum the weighted summation is an activation function.

$$\text{activation}_w(\bar{x}) = \sum_{j=1}^{k} \bar{x}_j \cdot w_j = \bar{w} \cdot \bar{x}$$



$$\begin{cases} 1, \text{if } \bar{w} \cdot \bar{x} \geq 0 \\ -1 \text{ otherwise.} \end{cases}$$

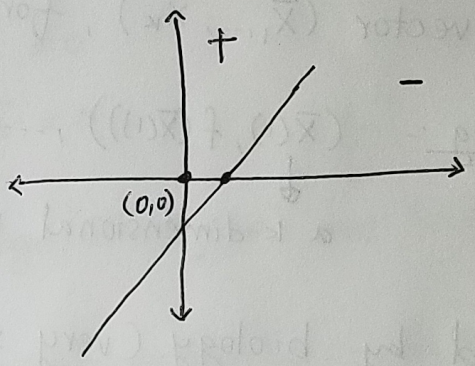
Intercept / bias, add a dummy feature $\bar{x}_0 = 1$ for every example, $\bar{w}_0$ is the weight of this feature

What does the perceptron classifier do?

1. Using the "training examples" learn a set of weight vectors $(w_0, w_1, \ldots, w_k)$

2. For the new test $\bar{x}$, output $\text{sign}(\bar{w} \cdot \bar{x})$.

In 2D :- $(k=2)$,



Is w_0 positive or negative?
Special feature, additional, for the intercepts.

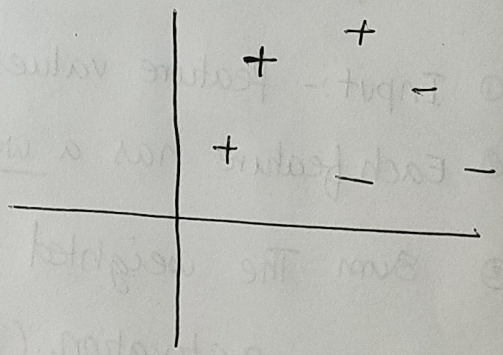
Large w_j → strong activation/important weight.

These are hyperplanes. very high dimensional.

↓

▷ Decision rule → $\bar{w} \cdot \bar{x} = 0$ is the decision boundary.

How to learn the weights?

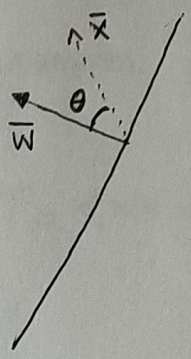


ALGORITHM :-

1. Start with weights = 0.
2. For iter = 1, ..., I
   2(a) For each training example,
   - Classify with the current weight
     - If correct, i.e., $f(\bar{x}(j)) = \text{sign}(\bar{w} \cdot \bar{x}(j))$
       DO NOTHING
     - If not correct, adjust $\bar{w}$,
       change the $\bar{w}$ toward the example.

θ = angle between $\bar{w}$ & $\bar{x}$

determines the sign of $\bar{w} \cdot \bar{x}$.



$\theta > 90^\circ \longrightarrow -1$

$\theta \leq 90^\circ \longrightarrow +1.$

$$\text{output} = \begin{cases} +1 & \text{if } \bar{w} \cdot \bar{x} \geq 0 \\ -1 & \text{if } \bar{w} \cdot \bar{x} < 0 \end{cases}$$

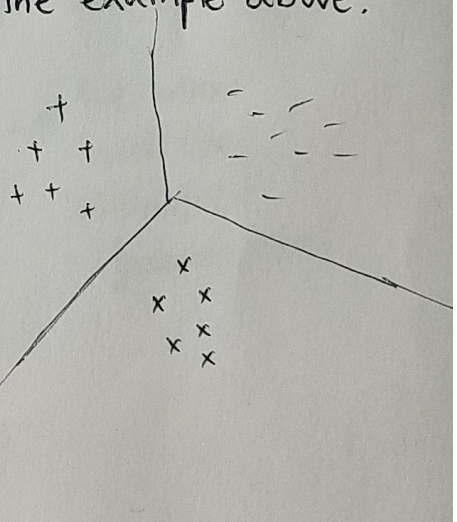
If correct, i.e., $f(\bar{x}) = \text{sign}(\bar{w} \cdot \bar{x})$, i.e, +ve here, do nothing.

If wrong, adjust the weight vector by subtracting/adding the feature vector.

$\longrightarrow$ If wrong, $\bar{w}_n = \bar{w}_o + f(\bar{x}) \cdot \bar{x}$.

$\bar{w}_n \cdot \bar{x} = \bar{w}_o \cdot \bar{x} + f(x) \cdot \|\bar{x}\|_2^2$, slowly make change towards $f(\bar{x})$.

⊖ Show the increase in the angle for the example above.



Multiclass Decision Rules.

For each class we have a ~~label~~ weight vector.

(w_+, w_-, w_x), C = set of all classes.

Score of a class is $w \cdot \bar{x}$

for feature vector $\bar{x}$,

$c = \underset{c \in C}{\text{argmax}}\ w_c \cdot \bar{x}$ ← select class with highest score.

Some notations, recap.

$\bar{X}$, $\bar{X}(1)$, $\bar{X}(2)$, ..., $\bar{X}(n)$, are the features of examples.

$\bar{X}(i)$ is example 'i', therefore a 'k-d' vector.

$\bar{X}_j(i)$ → j^{th} feature of example 'i'.

$T = \langle x(1), f(x(1)) \rangle$, ...

$\bar{X}$ → new example/ one new feature vector.

$\bar{X}_j$ → j^{th} feature of $\bar{X}$.

$f(\bar{X})$ → label of $\bar{X}$, eg. $f(\bar{X}(i))$ - label of example 'i'.

↳ a scalar.

- Linear classifiers :- A family number of classifiers with linear decision boundaries.

∴ decision boundary :- $w_1 \bar{x}_1 + \dots + \bar{w}\,\bar{x}_K \gtrless t$ (for example $\bar{x}$; scalar; threshold)

$\bar{x} = (\bar{x}_1, \dots, \bar{x}_K)$ ↳ scalars

$\bar{w} \cdot \bar{x} \gtrless t$ ← decision rule.

can be written as $\bar{x}^* \cdot \bar{w}^* \geq 0$,

by making

$\bar{x}^* = (-1, \bar{x}_1, \dots, \bar{x}_K)$

$\bar{w}^* = (t, w_1, \dots, w_K)$

→ We work with these 'modified feature vectors'

"how to choose good linear boundaries"?

Perceptron :-

Start with $\bar{w}^* = 0$

For iter = 1, ..., MAXITER,

For i = 1, ..., n,

if $sgn(\bar{x}^*(i) \cdot \bar{w}^*) \neq f(\bar{x}(i))$ (namely, $f(\bar{x}(i)) \times (\bar{w}^* \cdot \bar{x}(i)) < 0$)

$$\bar{w}^* \leftarrow \bar{w}^* + f(\bar{x}(i)) \cdot \bar{x}^*(i)$$

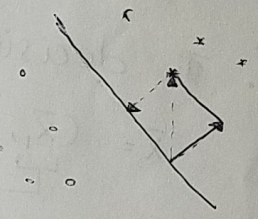
Output $\bar{w}^* = (w_0, w_1, \ldots, w_k)$,

Decision rule = $w_1 \bar{x}_1 + \ldots + w_k \bar{x}_k \gtrless w_0$

Example 'i', $sgn(\underbrace{\bar{w}^* \cdot \bar{x}(i)}_{+ve}) \neq \underbrace{f(\bar{x}(i))}_{-ve}$,

$$(\bar{w}^* + f(\bar{x}(i)) \cdot \bar{x}^*(i)) \cdot \bar{x}^*(i) = \underbrace{\bar{w}^* \cdot \bar{x}^*(i)}_{+ve} + \underbrace{f(\bar{x}(i)) \cdot |\bar{x}^*(i)|_2^2}_{-ve}$$

"makes it small."



Recall :-

For a plane $\bar{w}^* \cdot 0$, and any point $\bar{x}^*$,

$d(\bar{x}^*, \bar{w}^*) \left| \frac{\bar{w}^* \cdot \bar{x}^*}{|\bar{w}^*|} \right|$ is the distance of $\bar{x}^*$ from the plane $\perp^r$ to $\bar{w}^*$.

Margin :-

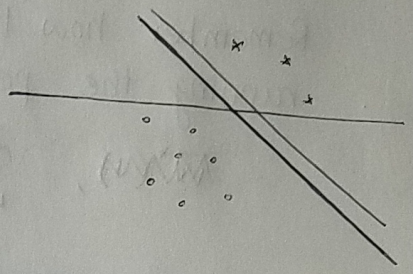
Separability :- $\bar{w}^*$ separates $\bar{x}(1), \ldots, \bar{x}(n)$ if it gets all the training examples correct.

$$\text{Margin}(\bar{w}^*, T) = \begin{cases} \min_{i=1,\ldots,n} d(\bar{x}^*(i), \bar{w}^*) & \text{if } \bar{w}^* \text{ separates } T \\ -\infty & \text{o/w} \end{cases}$$

Margin of the data, = maximum margin of all separating $\bar{w}^*$.

'γ' ⟶ denotes the margins.

$\gamma(T) = \max_{\bar{w}^*} \text{margin}(T, \bar{w}^*)$



* Larger margin, 'higher wiggle room'

Theorem :⇒ Suppose we run the perceptron algorithm, on a dataset 'T' with margin γ, and $|\bar{x}^*(i)|_1 \leq 1$. Then after at most $4/\gamma^2$ updates to $\bar{w}^*$ perceptron converges

$\bar{w}^*$ = all zeros initially, $\bar{w}^*_{OPT} \longrightarrow |\bar{w}^*_{opt} \cdot \bar{x}(i)| \geq \gamma$.

$$\bar{w}^*(j+1) \cdot \bar{w}^*_{OPT} = \bar{w}^*(j) \cdot \bar{w}_{OPT} + \underbrace{f(\bar{x}(i)) \cdot (\bar{w}^*_{OPT} \cdot \bar{x}(i))}_{>\gamma}$$

$$\geq (j+1) \cdot \gamma$$

$$|\bar{w}^*(j+1)|_2^2 \leq 2(j+1)$$

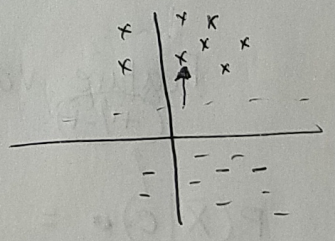
By Δ in $|\bar{w}^*(j+1)|_2^2 = |\bar{w}^*(j)|_2^2 + |\bar{x}(i)|_2^2 + 2\underbrace{f(\bar{x}(i)) \cdot (\quad)}_{<0}$

$$= |\bar{w}^*(j)|_2^2 + \sqrt{2}$$

$(j+1)\cdot\gamma < |\bar{w}^*(j+1)|_2 < \sqrt{2} \cdot \sqrt{(j+1)} \Rightarrow \boxed{j+1 < \frac{1}{\gamma^2}}$

---

* ~~NB~~ **Problems** (1) One nasty example.



Flip everything around

(2) weak generalization.

⟶ SVM later.

(3) Does not "learn from correct samples".

4. Multiclass even separable cannot be learnt

- Voted Perceptron.

Remember how long each hyperplane survives, as we keep running the perceptron algorithm.

$\bar{w}^{*(1)}, \ldots, \bar{w}^{*(c)}$

$t_1 \qquad t_c$

$$\text{sign}\left(\sum_{j=1}^{c} t_j \cdot \text{sign}(\bar{w}^{(j)} \cdot \bar{x})\right).$$

Works well, huge storage!, slow prediction.

- Averaged perceptron:- $\text{sign}\left(\sum_{j=1}^{c} t_j \cdot (\bar{w}^{(j)} \cdot \bar{x})\right)$

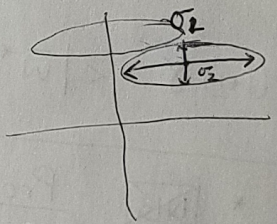
<Can maintain running averages> ⟶ Faster.

Gaussian NB as a linear model.

Two classes, $\{+1, -1\}$, / $\{c_1, \ldots, c_L\}$.

$\{\hat{\mu}_{j|+1}, \hat{\mu}_{j|-1}\}$ Each feature conditioned on class is Gaussian.

$\{\hat{\sigma}^2_{j|+1}, \hat{\sigma}^2_{j|-1}\}$ Suppose, $\hat{\sigma}^2_{j|+1} = \hat{\sigma}_{j|-1} = \hat{\sigma}_j^2$



Feature variances are same given class.

$$P(\bar{x}|c) = \prod_{j=1}^{K} P(\bar{x}_j|c) = \prod_{j=1}^{K} \frac{1}{\sqrt{2\pi\hat{\sigma}_j^2}} \exp\left(-\frac{(\bar{x}_j - \hat{\mu}_{j|c})^2}{2\hat{\sigma}_j^2}\right).$$

(+1/-1)

Decision Rule :-

$$P(C=+1) \cdot \prod_{j=1}^{K} \frac{1}{\sqrt{2\pi\sigma_j^2}} \cdot \exp\left(-\frac{(\bar{x}_j - \hat{\mu}_{j|+1})^2}{2\sigma_j^2}\right) \gtrless P(C=-1) \cdots$$

$$\Leftrightarrow \log(P(+1)) - \sum_{j=1}^{K} \frac{(\bar{x}_j^2 - 2\bar{x}_j \cdot \hat{\mu}_{j|+1} + \hat{\mu}_{j|+1}^2)}{2\sigma_j^2} \gtrless \log P(-1) - \sum_{j=1}^{K} (\times \quad)$$

$$\Rightarrow \quad 2 \times \sum_{j=1}^{K} \bar{x}_j \cdot \frac{(\hat{\mu}_{j|+1} - \hat{\mu}_{j|-1})}{2\sigma_j^2} + \boxed{\log \frac{P(+1)}{P(-1)} - \sum_{j=1}^{K} \frac{(\hat{\mu}_{j|+1}^2 - \hat{\mu}_{j|-1}^2)}{\sigma_j^2}} \gtrless 0.$$

A linear classifier.