

المحاضره رقم 6 هنتكلم فيها عن ال perceptron algorithm .. بيفك ان ال HW3 هيبقا written .. و HW2 قرب يتسلم .. شد حيلك و فنش الهوموورك .. نهبط شوية تعريفات .. بس من ناحيتين ال statistics و احنا . ال statistics .. هيقول ان الموديل بيعرف ال data generation process .. ده كإنو ال a set or a family of parametric probability distributions .. انما في الماشين هتقول ان الموديل بيعرف ال hypothesis space اللي ال learning هيدور فيها ... مش بتقول ان هيبقا في ال probabilistic model .. الموديل بارمترز في الستاتستكس برضو تعريفها بيختلف ..

## Model Selection

| Statistics  | Machine Learning   |
|---|--|
| <ul style="list-style-type: none"> <li>Def: a <b>model</b> defines the data generation process (i.e. a set or family of parametric probability distributions)</li> <li>Def: <b>model parameters</b> are the values that give rise to a particular probability distribution in the model family</li> <li>Def: <b>learning</b> (aka. estimation) is the process of finding the parameters that best fit the data</li> <li>Def: <b>hyperparameters</b> are the parameters of a prior distribution over parameters</li> </ul> | <ul style="list-style-type: none"> <li>Def: (loosely) a <b>model</b> defines the hypothesis space over which learning performs its search</li> <li>Def: <b>model parameters</b> are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis</li> <li>Def: the <b>learning algorithm</b> defines the data-driven search over the hypothesis space (i.e. search for good parameters)</li> <li>Def: <b>hyperparameters</b> are the tunable aspects of the model, that the learning algorithm does not select</li> </ul> |

احنا هناخد ال statistical machine learning view .. بس مش دلوقت .. هيبقا في النص الثاني .. الدكتور عاوز يأجل الربط ما بين الناحيتين لبعدين .. الدكتور تقريباً قرا السلايدز مع شوية شرح .. اقراهم هتفهم اللي مكتوب .. بص ع المثال اللي جاي ده:

## Model Selection

### Example: Decision Tree

- model = set of all possible trees, possibly restricted by some hyperparameters (e.g. max depth)
- parameters = structure of a specific decision tree
- learning algorithm = ID3, CART, etc.
- hyperparameters = max-depth, threshold for splitting criterion, etc.

### Machine Learning

- Def: (loosely) a **model** defines the hypothesis space over which learning performs its search
- Def: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- Def: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- Def: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does not select

9

المهم الدكتور بيقول ان لو خدنا ال KNN كمثال: ..

## Model Selection

### Example: k-Nearest Neighbors

- model = set of all possible nearest neighbors classifiers
- parameters = none (KNN is an instance-based or non-parametric method)
- learning algorithm = for naïve setting, just storing the data
- hyperparameters =  $k$ , the number of neighbors to consider

### Machine Learning

- Def: (loosely) a **model** defines the hypothesis space over which learning performs its search
- Def: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- Def: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- Def: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does not select

10

هتلاقي ان مالوش بارمترز .. لأن التريننج عبارته عن انك حرفياً بتخزن الداتا .. و التستنج هنا هو عبارته عن انك بتدور في الداتا اللي عندك ..

الآلجورزم الجديد بقا ... perceptron ... الموديل نفسو هبيقا عبارته عن شوية linear separators .. ال parameters هبيقو a vector of weights .. واحد لكل فيتشر .. ال learning algorithm هبيقا iterative ... ومفيش hyperparameters

## Model Selection

### Example: Perceptron

- model = set of all linear separators
- parameters = vector of weights (one for each feature)
- learning algorithm = mistake based updates to the parameters
- hyperparameters = none (unless using some variant such as averaged perceptron)

### Machine Learning

- Def: (loosely) a **model** defines the hypothesis space over which learning performs its search
- Def: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- Def: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- Def: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does not select

11

نيجي للموديل سليكشن .. هو انك تختار احسن موديل .. الهايبر بارمترز أوبتيمايزيشن هو انك تختار قيمه محدده للهايبر بارمترز ..

## Model Selection

- Two very similar definitions:
  - Def: **model selection** is the process by which we choose the “best” model from among a set of candidates
  - Def: **hyperparameter optimization** is the process by which we choose the “best” hyperparameters from among a set of candidates (**could be called a special case of model selection**)
- **Both** assume access to a function capable of measuring the quality of a model
- **Both** are typically done “outside” the main training algorithm --- typically training is treated as a black box

12

تعال نفكر في اللي احنا عملناه .. بص ع السلايد الجايه ديه وشوف كل واحد بتاخذ ايه ..

## Experimental Design

|                             | Input  | Output  | Notes  |
|-----------------------------|--|---|--|
| Training                    | <ul style="list-style-type: none"> <li>• training dataset</li> <li>• hyperparameters</li> </ul>                      | <ul style="list-style-type: none"> <li>• best model parameters</li> </ul> | We pick the best model parameters by learning on the training dataset for a fixed set of hyperparameters                     |
| Hyperparameter Optimization | <ul style="list-style-type: none"> <li>• training dataset</li> <li>• validation dataset</li> </ul>                   | <ul style="list-style-type: none"> <li>• best hyperparameters</li> </ul>  | We pick the best hyperparameters by learning on the training data and evaluating error on the validation error               |
| Testing                     | <ul style="list-style-type: none"> <li>• test dataset</li> <li>• hypothesis (i.e. fixed model parameters)</li> </ul> | <ul style="list-style-type: none"> <li>• test error</li> </ul>            | We evaluate a hypothesis corresponding to a decision rule with fixed model parameters on a test dataset to obtain test error |

تعال نشوف ازاى نختار قيمة ال  $k$  عن طريق اننا نرسم ال train and test error .. تعال نرسم علي الإكس إكسيس  $K$  . و الإيروز علي الواي أكسيس .. قيم ال  $k$  عادي 1 و 2 و 3 و  $N$  اللي هو من أول 1 لحد عدد ال examples اللي عندنا .. تعال برضو نفترض ان الداتا سيت بتاعتنا بتتكون من 40% امثله من ال  $y=0$  و 60% من ال  $y=1$  ..

عند ال  $k=1$  .. لما بتيجي ت evaluate التريننج سيت علي KNN .. كأنك بتعمل copy of the training set .. لما واحده من ال new copied training point هتيجيك .. هيبقا اقرب حاجة ليها هو نفسها .. لو التريننج سيت جواها أكثر من نقطة ليهم نفس المكان بس اختلاف ال labels .. بس هيبقا عندك تعادل .. بس لو ده محصلش .. بيبقا التريننج إيروز هيبقا صفر .. والتريننج إيروز هيبدا يزد كل ما ال  $k$  عماله تزيد .. هتلاقي ان الإيروز عند ال  $K=N$  .. هيبقا ال majority votes .. اللي هو هيبقا ال 60% لكل الليبلز اللي عندنا .. فالإيروز هيقف عند ال 40% ..

بالنسبة للتستنج إيروز هيبدا أعلي حاجة و بعدين يبدأ يبق .. بص ع الجراف اللي الدكتور رسمه ... الدكتور سأل سؤال .. نختار أنهي قيمة لل  $K$  .. حد اقترح نختار اقل تست إيروز .. بس الدكتور طعن ف الفكره ديه .. قال انو معندناش held out data عشان تقدر نتست الموديل عليها .. لو انت عملت الاقتراح ده .. فانت كدا عملت training لقيمة ال  $K$  علي التست داتا .. ده معناه انك دلوقت خلاص ممعكش داتا فعلياً .. فانت مينفعش تدخل التست داتا في أي حاجة نهائي نهائي .. مش فاكركان مين اللي قال الجملة ديه برضو ..

في حل انك تقسم التريننج داتا سيت لتريننج و فاليديشن داتا .. واستخدم فاليديشن انك تختار ال  $k$  اللي انت عاوزها .. أول ما هتختار ال  $K$  .. روح اعمل تست علي التست داتا سيت ..

....

دلوقت انت في حاجة مش واخد بالك منها .. انت عامل تريننج علي 900 مثال و بعدين اخترت ال  $K$  علي اساس 300 مثال .. طب ما قبل ما تعمل evaluation علي التست إيروز ... روح اعمل retraining علي ال 1200 مثال .. وغالباً انت فعلياً هتبقا احسن بدل ما تعمل تريننج علي 900 بس ..

دلوقت انت ممكن مثلاً تقول ان ال estimation علي ال validation error مش robust .. ايه رأيك نعملو أكثر من مره .. يعني انت ممكن تاخذ ال 300 مثال دول بس من أكثر من حته في التريننج داتا .. فالدكتور اختار فاليديشن سيت مختلفه و قام راسم كيرف ثاني .. و بعدين اختار داتا اللي هي متلونه بال grey و بعدين يرسم ... دلوقت انت عندك 4 كيرفات ... دلوقت ازاى اختار قيمة ال  $K$  من ال 4 كيرفات بتوع ال validation set ... الدكتور اقترح مثلاً انك تاخذ ال average .. وهو ده ال cross validation ..

cross validation

KNN Train & Test Error



في حد سأل سؤال .. ليه ال fold ده اللي جواه 300 مثال بس .. هي perform differently .. الدكتور رد قالو في 300 مثال بس في كل fold .. لو انت كان عندك infinite امثله في كل fold .. في الحاله ديه كل الكيرفات بتاعت ال fold هتبقا زي بعض .. انما هنا عشان احنا عندنا small sample كدا عندنا شوية noise في الكيرف .. سؤال جامد الحقيقه ...

المهم لو جينا نعرف ال cross validation هنا .. نقول ان احنا هنقسم لشوية folds ... و اقرا السلايد الجايه ..

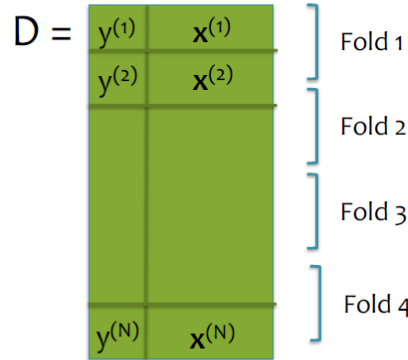
# Cross-Validation

**Cross validation** is a method of estimating loss on held out data

**Input:** training data, learning algorithm, loss function (e.g. 0/1 error)

**Output:** an estimate of loss function on held-out data

**Key idea:** rather than just a single “validation” set, use many!  
(Error is more stable. Slower computation.)



Definition:  
N-fold cross validation = cross validation with N folds

## Algorithm:

Divide data into folds (e.g. 4)

1. Train on folds  $\{1,2,3\}$  and predict on  $\{4\}$
2. Train on folds  $\{1,2,4\}$  and predict on  $\{3\}$
3. Train on folds  $\{1,3,4\}$  and predict on  $\{2\}$
4. Train on folds  $\{2,3,4\}$  and predict on  $\{1\}$

Concatenate all the predictions and evaluate loss (almost equivalent to averaging loss over the folds)

18

احنا هنقسم لشوية فولدز .. فِكر في الأول انك predicting on all of the training data .. ازاي هتعمل كذا؟ ...

أول حاجه train علي فولدز 1 2 3 و اعمل ال prediction علي 4

بعدين train علي فولدز 1 2 4 و اعمل ال prediction علي 3

وهكذا لحد ما توصل لخطوه رقم 4 و تخلصها .. المهم بعد خطوه 4 خذ كل ال predictions اللي انت عملتهم .. حطهم في فيكتور واحد .. وبعدين evaluate ال loss function بتاعتك علي كل ال predictions دول .. المهم كذا ان ولا single prediction انت عملتو كان training point في ال hypothesis اللي طلعتها ..

احنا بنقول K-folds cross validation فبيبقى عندنا K folds .. انما لو قلنا N-fold cross validation ده معناه ان عدد ال folds هو قد عدد الأمثله اللي عندك .. ف كل fold هيبقا جواه مثال واحد بس .. و ده اللي هو اسمو leave one out cross validation ..

# Experimental Design

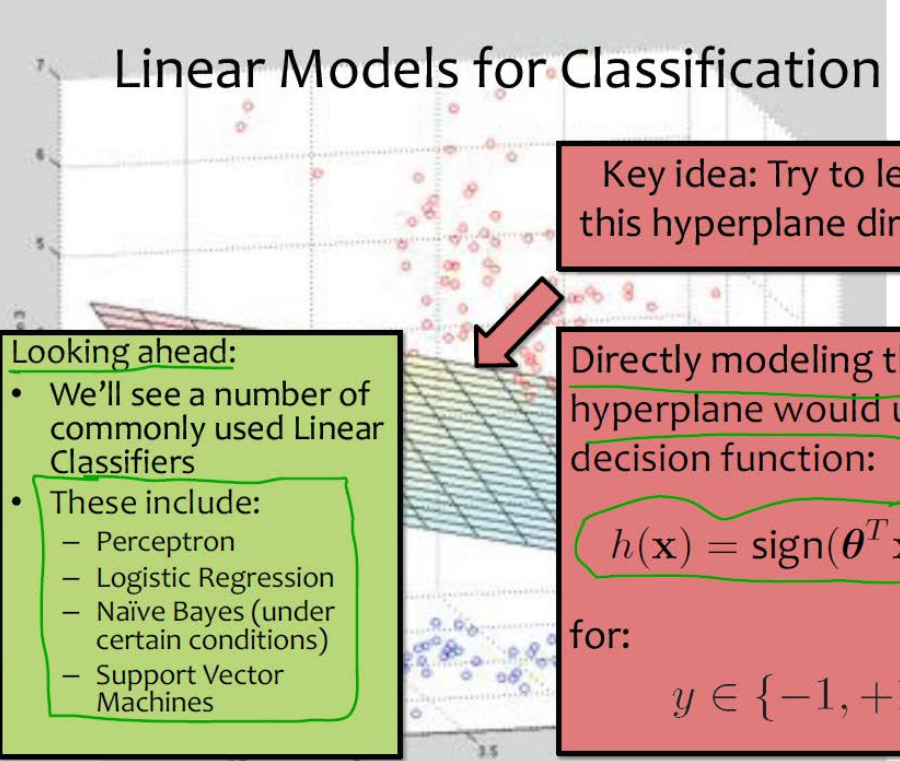
|                             | Input  | Output   | Notes  |
|-----------------------------|--|--|--|
| Training                    | <ul style="list-style-type: none"> <li>training dataset</li> <li>hyperparameters</li> </ul>                      | <ul style="list-style-type: none"> <li>best model parameters</li> </ul>  | We pick the best model parameters by learning on the training dataset for a fixed set of hyperparameters                     |
| Hyperparameter Optimization | <ul style="list-style-type: none"> <li>training dataset</li> <li>validation dataset</li> </ul>                   | <ul style="list-style-type: none"> <li>best hyperparameters</li> </ul>   | We pick the best hyperparameters by learning on the training data and evaluating error on the validation error               |
| Cross-Validation            | <ul style="list-style-type: none"> <li>training dataset</li> <li>validation dataset</li> </ul>                   | <ul style="list-style-type: none"> <li>cross-validation error</li> </ul> | We estimate the error on held out data by repeatedly training on N-1 folds and predicting on the held-out fold               |
| Testing                     | <ul style="list-style-type: none"> <li>test dataset</li> <li>hypothesis (i.e. fixed model parameters)</li> </ul> | <ul style="list-style-type: none"> <li>test error</li> </ul>             | We evaluate a hypothesis corresponding to a decision rule with fixed model parameters on a test dataset to obtain test error |

استخدمه عشان تختار قيمه ال hyperparameter اللي انت عاوزو

19

يلا نخش في ال perceptron algorithm .. تخيل انت عايش 1957 .. وانت عاوز تبني new machine learning technique .. فقام قايل انا هبني physical machine ... فقام بني دايره electric circuit و هي ديه ال perceptron .. تعال نشوف شكلو عامل ازاى ... تعال نبص علي ال linear models الاول بتعمل ايه .. بص ع الصوره:

## Linear Models for Classification



**Key idea: Try to learn this hyperplane directly**

**Looking ahead:**

- We'll see a number of commonly used Linear Classifiers
- These include:
  - Perceptron
  - Logistic Regression
  - Naïve Bayes (under certain conditions)
  - Support Vector Machines

**Directly modeling the hyperplane would use a decision function:**

$$h(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$$

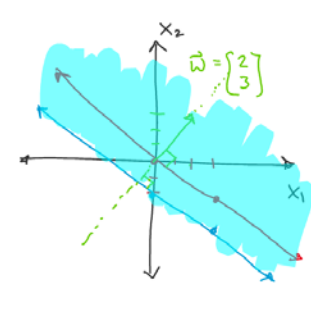
for:

$$y \in \{-1, +1\}$$

تعال ناخذ exercise ...

**Ex: Linear Classifier**

$w_1 x_1 + w_2 x_2 + b > 0$  where  $w_1 = 2, w_2 = 3, b = 6$



$\vec{w} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$w_1 x_1 + w_2 x_2 = 0$

$x_2 = \left(-\frac{w_1}{w_2}\right) x_1$

$w_1 x_1 + w_2 x_2 + b = 0$

$x_2 = \left(-\frac{w_1}{w_2}\right) x_1 - \frac{b}{w_2}$

**Linear Classifier**

$\hat{y} = h(\vec{x})$

$= \text{sign}(w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)$

$= \text{sign}(\vec{w}^T \vec{x} + b)$



## Definitions

Def: a vector  $\vec{a}$  is orthogonal to vector  $\vec{b}$   
iff  $\vec{a} \cdot \vec{b} = 0$

Def: dot product  $\vec{a} \cdot \vec{b} = \vec{a}^T \vec{b} = \sum_i a_i b_i$

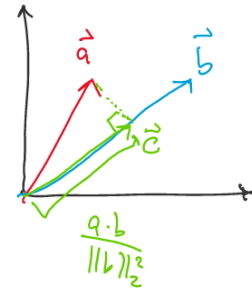
Def: the  $\ell_2$  norm of vector  $\vec{u}$  is  $\|\vec{u}\|_2 = \sqrt{\sum_{m=1}^M u_m^2}$

Def: vector projection of  $\vec{a}$  onto  $\vec{b}$   
where  $\|\vec{b}\|_2 = 1$

$$\vec{c} = (\vec{a} \cdot \vec{b}) \vec{b}$$

Def: vector projection of  $\vec{a}$  onto  $\vec{b}$

$$\vec{c} = \left( \frac{\vec{a} \cdot \vec{b}}{\|\vec{b}\|_2} \right) \frac{\vec{b}}{\|\vec{b}\|_2} = \frac{\vec{a} \cdot \vec{b}}{(\|\vec{b}\|_2)^2} \vec{b}$$



## Hypersplane

in 2D  $\equiv$  line

3D  $\equiv$  plane

4D  $\equiv$  hypersplane



Def: hypersplane  $S = \{ \vec{x} : \vec{w}^T \vec{x} + b = 0 \}$

Def: half spaces  $S_+ = \{ \vec{x} : \vec{w}^T \vec{x} + b > 0 \}$

$S_- = \{ \vec{x} : \vec{w}^T \vec{x} + b < 0 \}$

تعال نروح ن define ايه هو ال perceptron algorithm ونشوفه بيعمل ايه .... بيقلك ان ال perceptron algorithm هيعتمد علي الفانكشن بتاعت ال sign ... لحد دلوقت احنا كنا مفترضين اننا بنعمل batch learning .. بتتعلم من كل الأمثلة مره واحده .. في ال Online learning .. مفيش تريننج سبت ومفيش تست سبت .. كل مثال بيجيك ورا الثاني .. عندك ستريم من الأمثلة جايك ورا بعضها .. حاجه مثلا زي ال stock market prediction . عندك برضو ال email classification و برضو ال recommendation .. ال distribution بتاع الحاجات ديه بيتغير مع الوقت .. كل دول بيتغيرو مع الوقت ..

فال paradigm اللي شغالين عليها دلوقت هي اننا هنلف علي ال time steps من 1 لحد انفتي .... و عند تايم ستيب i .. هتستقبل unlabeled instance  $X_i$  .. ده اللي هو الفيتشر فيكتور .. وبعدين تاخذ ال hypothesis بتاعتك ناو .. اللي هي اسمها h theta .... وبعدين ت apply h theta ل ال  $X_i$  اللي جالك .. ويطالعك  $\hat{Y}_i$  . و بعدها هيجيك ال true label اللي اسمو  $Y_i$  وهنا بس انت بتعرف هل انت عملت ال  $\hat{Y}_i$  صح ولا لا .. في ال stock market انت بتخسر فلوس فعلاً ... فبعد ما خسرت هتعمل ايه .. هتروح ال update البارمتر بتاعك .. والهدف هنا هو انك تقلل عدد الغلطات ..

معدكش test ولا train ... منين ما يجيلك مثال .. أياً كان ال hypothesis اللي عندنا .. و نطبقها علي المثال اللي عندنا .. ونطلع ناتج .. بعد شوية هيدبك الرد وفي اللحظة ديه و اللحظة ديه بس .. انت هتعرف انت جبت الناتج صح ولا لا ... فانت بالاضافه لإن عندك sequence of examples .. عندك كمان sequence of classifier over time .. دلوقت بقا نقدر نعرف ال Online version من ال perceptron algorithm ..

الالجورزم هيبدا عن طريق انوي initialize 2 بارمترز .. واحد هو الفيكتور  $W$  بيتكون من  $w_1, w_2, w_3, \dots, w_m$  .. وكمان هنعرف intercept term اسمو  $B$  .. في الالجورزم ده هنقول ان الاتنين دول ليهم قيمه صفر .. اللي هو  $W = \text{zero vector}$  .. فيكتور كلو اصفار .. و ال  $b$  هيبقا ليه قيمه واحده اللي هي صفر ..

ملحظه: كل فيكتور بيبقا المقصود منو column vector ... فعشان كذا في الصوره هتلاقية حط ال Transpose ... ال  $W$  هي الويتس .. و ال  $b$  هي ال باياس .. الباياس هنا ليه اسم ثاني وهو ال intercept term .. الالجورزم نفسو هيبقا عامل ازاي ..

For  $i = 1, 2, 3, \dots$

1. Receive an unlabeled instance called:  $X_i$
2. Predict a label  $\hat{y} \rightarrow \hat{y} = h(x) = \text{sign}(w^T x + b)$ , where  $\text{sign}(a) = +1$  if  $a \geq 0$ ,  $-1$  otherwise
3. Now you will Receive a true,  $Y_i$ , where you will have an opportunity to learn
4. If positive mistake ( $\hat{y} \neq y_i$  and  $y_i = (+1)$ )
  - a.  $W \leftarrow W + X_i$
  - b.  $B \leftarrow B + 1$
5. If negative mistake ( $\hat{y} \neq y_i$  and  $y_i = (-1)$ )
  - a.  $W \leftarrow W - X_i$
  - b.  $B \leftarrow B - 1$

ده كذا هو ال learning algorithm .. سهل فشاخه .. قبل ما نتحرك لقدام أكثر تعال نفكر ايه هو ال hypothesis class لل perceptron ده .. ايه هو اللي بيعرف ال set  $H$  بتاع ال possible hypotheses اللي احنا بنشتغل عليها هنا ..

Hypothesis class: ?????? what is the set of all possible hypothesis, what is the space to search for a good classifier?

- ➔ Set of all hyperplanes ( aka linear decision boundaries )
- ➔ In  $M$  dimensional space where  $M = \text{number of features}$
- ➔  $H$ : is a function that maps to  $+1$  or  $-1$
- ➔ Hypothesis space  $H = \{\text{set of all functions } h \text{ such that there exists a vector } W \text{ in } R^m \text{ and a scalar } b \text{ such that we can define } h(x) \text{ as } \text{sign}(w^T x + b)\}$

➔ فبالتالي كذا هو فعلاً الفيكتور  $W$  و ال  $B$  ترم .. هم اللي بيعرفو ال hypothesis .. كل hypothesis بتبقا عبارته عن setting من البارمترز دول ..

for  $i = 1, 2, 3, \dots$

- ① receive instance  $\vec{x}^{(i)}$
- ② predict  $\hat{y} = h(\vec{x}) = \text{sign}(\vec{w}^T \vec{x} + b)$  where  $\text{sign}(a) = \begin{cases} +1 & \text{if } a \geq 0 \\ -1 & \text{otherwise} \end{cases}$
- ③ receive true label  $y^{(i)}$
- ④ if positive mistake ( $\hat{y} \neq y^{(i)}$  and  $y^{(i)} = +1$ )
 
$$\vec{w} \leftarrow \vec{w} + \vec{x}^{(i)} \quad \text{"add } \vec{x}^{(i)} \text{"}$$

$$b \leftarrow b + 1$$
- if negative mistake ( $\hat{y} \neq y^{(i)}$  and  $y^{(i)} = -1$ )
 
$$\vec{w} \leftarrow \vec{w} - \vec{x}^{(i)} \quad \text{"subtract } \vec{x}^{(i)} \text{"}$$

$$b \leftarrow b - 1$$

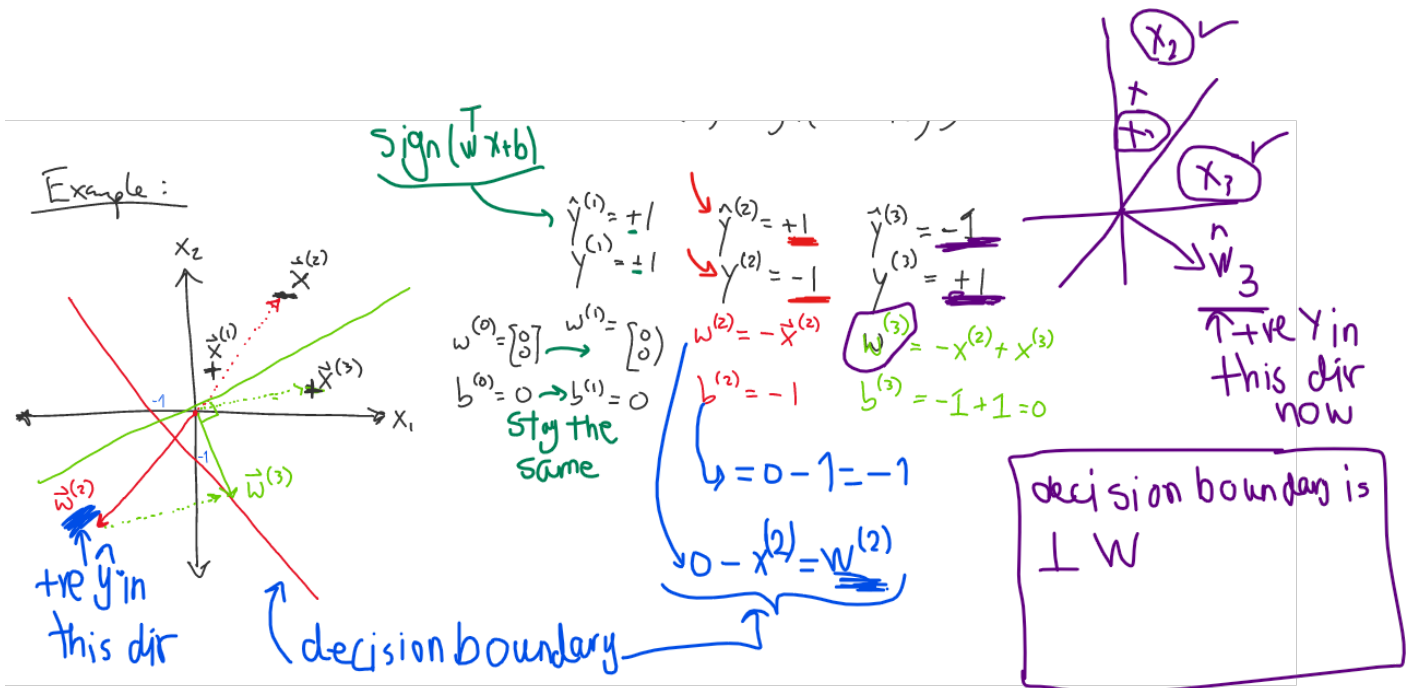
## Hypothesis Class:

set of all hyperplanes (aka. linear decision boundaries)  
in  $M$ -dimensional space (where  $M = \# \text{ features}$ )

$$h: \mathbb{R}^M \rightarrow \{+1, -1\}$$

$$\mathcal{H} = \{h(\cdot) : \exists \vec{w} \in \mathbb{R}^M, b \in \mathbb{R} \text{ s.t. } h(\vec{x}) = \text{sign}(\vec{w}^T \vec{x} + b)\}$$

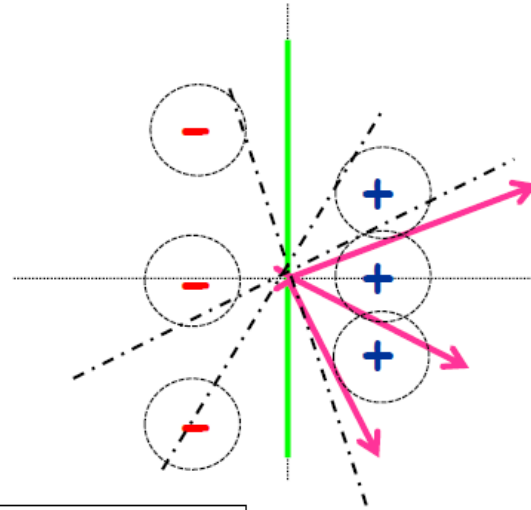
تعال نفكر ازاى نخط ال perceptron algorithm في مثال .. حاجه كذا في ال 2D ...





# Perceptron Algorithm: Example

Example:  $(-1, 2) -$  ✗  
           $(1, 0) +$  ✓  
           $(1, 1) +$  ✗  
           $(-1, 0) -$  ✓  
           $(-1, -2) -$  ✗  
           $(1, -1) +$  ✓



## Perceptron Algorithm: (without the bias term)

- Set  $t=1$ , start with all-zeroes weight vector  $w_1$ .
- Given example  $x$ , predict positive iff  $w_t \cdot x \geq 0$ .
- On a mistake, update as follows:
  - Mistake on positive, update  $w_{t+1} \leftarrow w_t + x$
  - Mistake on negative, update  $w_{t+1} \leftarrow w_t - x$

$$w_1 = (0, 0)$$

$$w_2 = w_1 - (-1, 2) = (1, -2)$$

$$w_3 = w_2 + (1, 1) = (2, -1)$$

$$w_4 = w_3 - (-1, -2) = (3, 1)$$