الأسبوع الأول

Introduction

التعليم المشرف Supervised:

- التقسيم Classification
  - نظام الـ Linear Regression
- التوقع Regression

- **التعليم غير المشرف Unsupervised :**
  - المجموعات Clustering
    - نظام الـ K Mean

- مجالات تعلم الالة :
  - الطب
  - الطائرات
  - مواقع مثل الفيس و امازون
  - الايميلات
- تعريفه :
  - هو جعل الالة تتعلم بنفسها دون توجيه بشري
  - هو جعل الالة تتعلم عبر مشاهدتها خبرة سابقة للانسان (E) للمهمة المطلوبة (T) ثم نقيم ادائها لاحقا (P)
    - فلو عايزين الايميل يفلتر الايميلات السبام اوتوماتيك ، فكون انه يشوفك بتعمل سبام للايميلات E
    - انه هو يقوم بعمل مهمة الفلترة بنفسه T
    - اننا نقيس مهارته في فلترة الايميلات و عدم اللغبطة P
  - نوعيه:
    - المشرف وغير المشرف

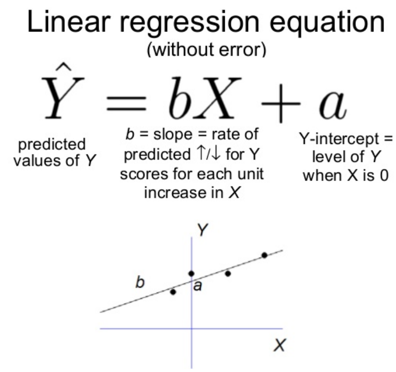
  - أهمية الخوارزميات
    - النجار اللي راح اتعلم ان ده شاكوش و ده منشار ، وشكرا ، هو نفسه اللي راح اتعلم ادوات تعلم الالة من غير ما يفهم الخوارزميات ، لان الخوارزميات هي نفسها طريقة استخدام ادوات تعلم الالة ، و من غيرها مفيش انجاز

- التعليم باشراف Supervised ML
  - لعمل مثال بسيط عليه . عندك اسعار بيوت و مساحاتها ، و عندك بيانات ، فممكن تعمل جراف بينهم ، و تعمل خط مستقيم او دالة تربيعية لتوقع البيت الفلاني سعره كام (المثال ده ريجريشين)
  - لو فيه مقارنه بيه حجم الورم و هل هو خبيث ولا حميد ، فبرضه نعمل جراف و نتوقع (كلاسيفيكاشن)

- التعليم غير المشرف
  - كمية ضخمة من البيانات و يتم تقسيمها الي مجموعات , كل مجموعة بسلوك محدد , ولها امثلة كثيرة , مثلما يقوم جوجل بتقسيم الاخبار كل علي حدة بشكل تلقائي , ومن امثلتها
    - تقسيم منتجات امازون
    - اقتراح الاصدقاء في الفيسبوك
    - تقسيم المركت لسيجمنتات
    - تقسيم اصوات متداخل (اتنين بيتكلمو في نفس الوقت) لصوتين منفصلين
    - والغريب ان هذا الامر يتم بسطر واحد في البرنامج

- الادوات المستخدمة
  - ماتلاب
  - اوكتيف :
    - الاستخدام الامثل ان تتعلم اكتيف اولا , ثم تركبه علي سي بلس بلس او جافا

- **الاستنتاج الخطي Linear Regression**
  - و يسمي أيضا (One Varilable Regression) او (Unvariate Regression)
  - يستخدم مع الامثلة البسيطة نوعا , و مع عامل واحد
  - يتم استخدم معادلة خطية من الدرجة الاولي :



- و هنا المعادلة الخاصة بيها
- الهدف تقليل الفارق بين قيمة h(x) و هي القيمة المتوقعة من المعادلة الخطية و قيمة y و هي القيمة الحقيقية بمعادلة تشابه الانحراف المعياري
- ويتم القسمة علي 2m لربط قيمة الخطا بعدد القيم بالعينة
- الهدف ايجاد قيم ثيتا 1 و ثيتا 2 , والتي تجعل من J (نسبة الخطا) اقل ما يمكن
- تسمي احيانا Cost error function

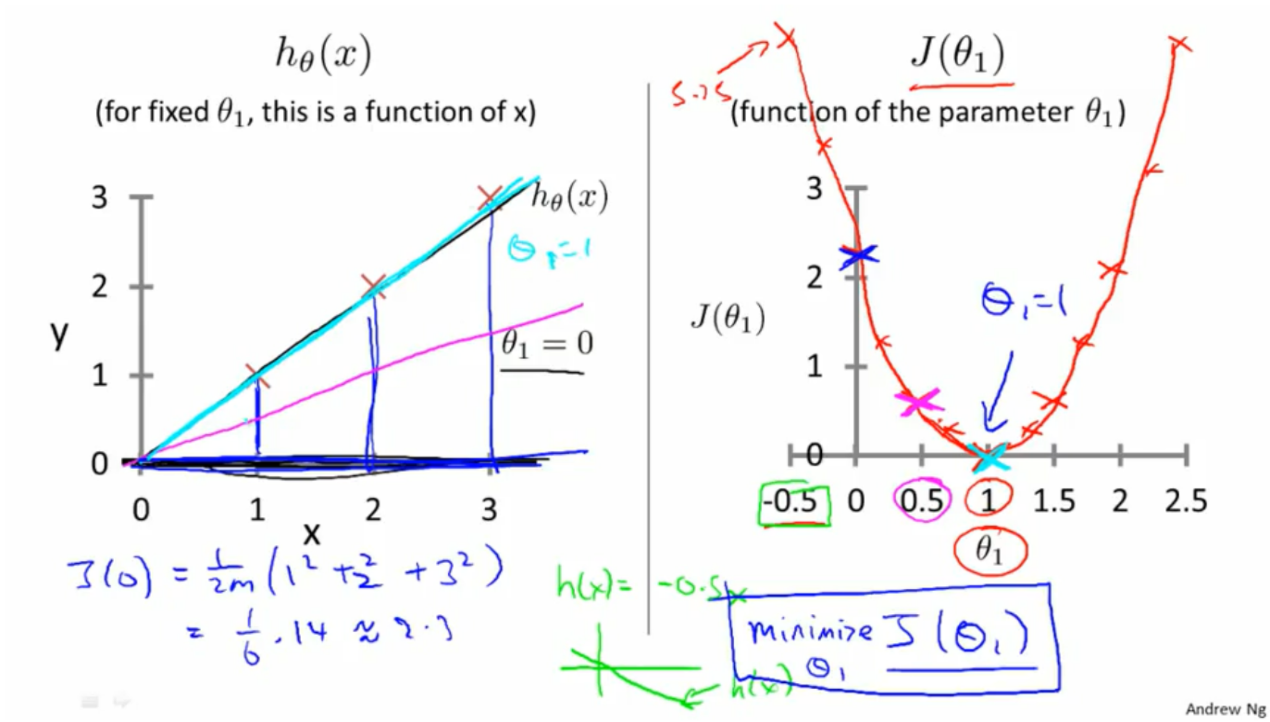
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

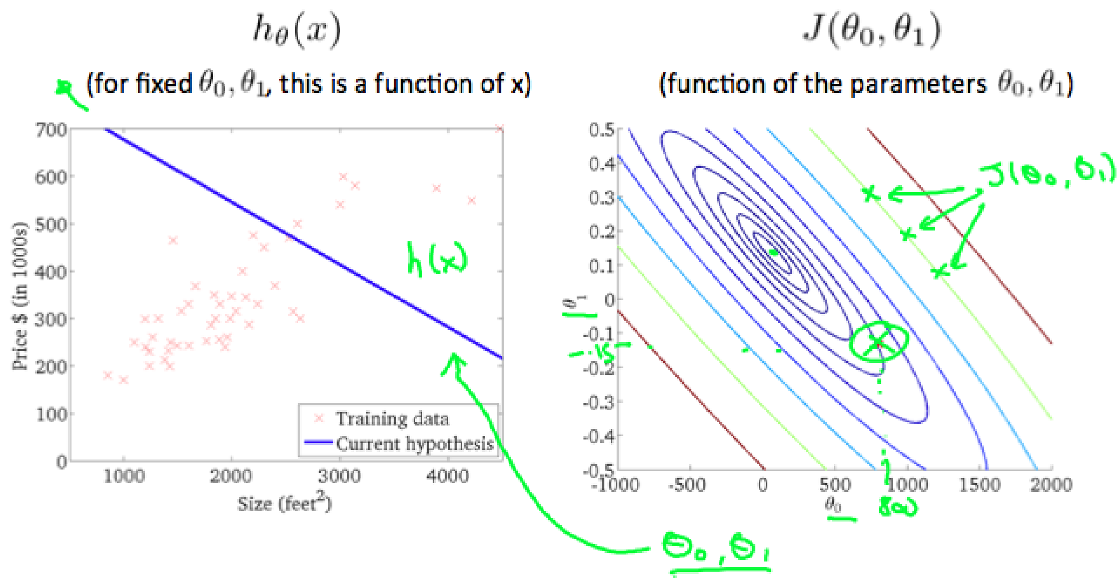
Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

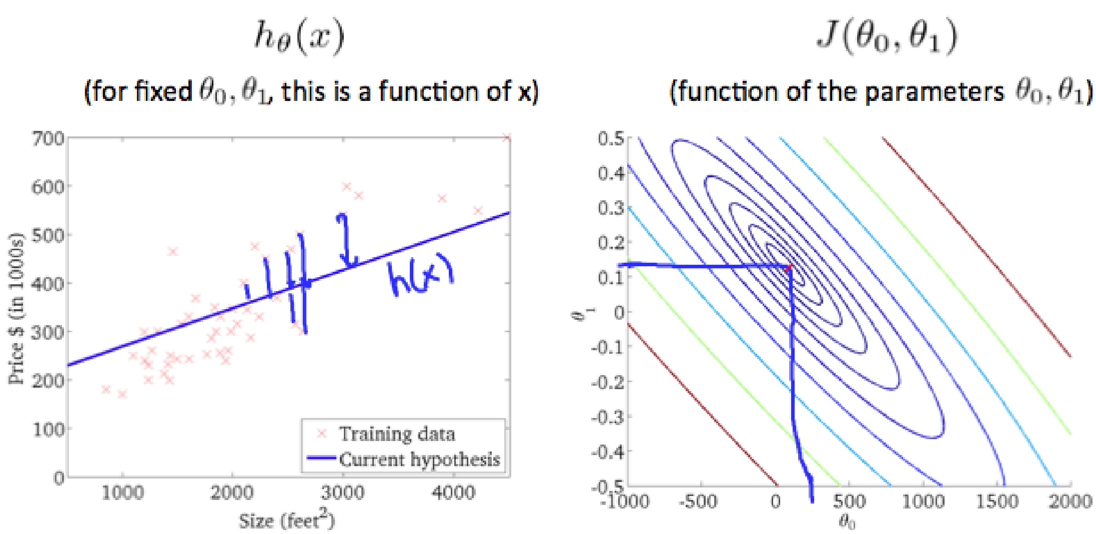
- نري علي اليسار قيم ثيتا 1 (معامل اكس و بفرض ان ثيتا صفر الحد المطلق يساوي صفر) مع hx وكلما زادت قيمة ثيتا 1 زاد السلوب , بينما نري علي اليمين قيمة الخطا الناشئ من زيادة او تقليل قيمة ثيتا , فنجد ان القيمة المثلي الان هي 1 (تختلف بالطبع في قيم اخري)

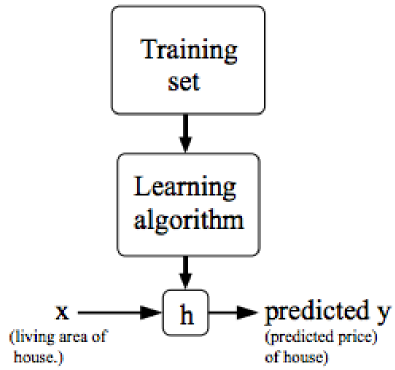


- المثال السابق كان مع تصفير ثيتا صفر و التلاعب فقط بثيتا 1 , الان اذا لعبنا بكلا من ثيتا 1 و ثيتا 0 , فسيكون لدينا عاملين , وبالتالي يجب ان تكون معادلة **J** مع الثيتتين , يجب ان تكون كونتور لنتمكن من قرائتها
- مع معلومية ان كل خط و كيرف من اليمين , يمثل قيمة ثابتة للـ **J** حتي لو اختلفت الثيتتين
- فمثلا الدايرة الحمرا البعيدة خالص , تمثل قيمة كبيرة للـ **J** (مثلا 20) و اي نقطة تقع عليها بقيم ثيتا 1 و ثيتا 2 هتجيب قيمة **J** بـ 20 والدايرة الاصغر شوية 19 , وهكذا
- من المنطقي اننا لو عايزين اقل قيمة للـ **J** (الـ **J** هي قيمة الفروق بين القيم الحقيقية و المتوقعة) يبقي نروح للدايرة الاصغر وهي الصفر او قيمة قريبة للصفر
- ناخد النقطة الحمراء كمثال , وهي تمثل ثيتا 0 تساوي 800 و ثيتا 1 تساوي سالب 1.5 , وبالتالي نرسمها مخط مستقيم علي اليمين بالمعادلة : **h(x) = 800-1.5x**



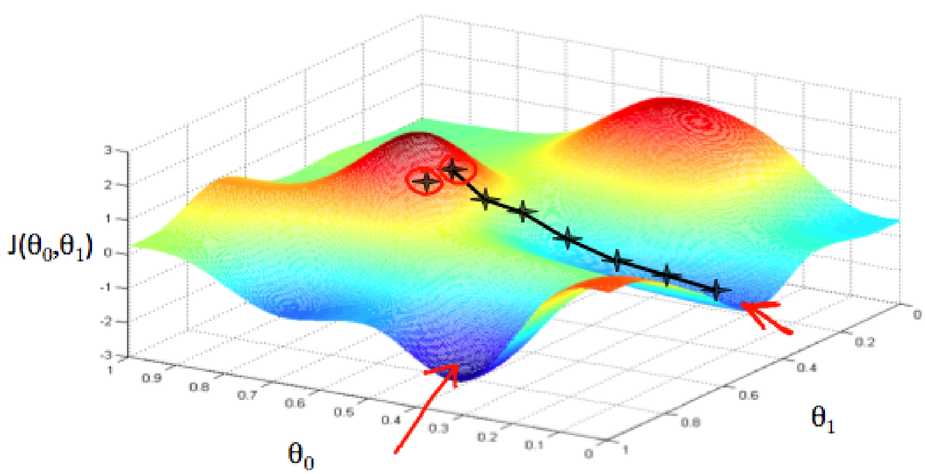
- ونفس الموضوع لو اخذنا كذا نقطة من اليمين و اللي تمثلها خط علي الشمال , لغاية لما نلاقي ان افضل حل هو النقطة اللي في نص الاشكال البيضاوية علي اليمين , اللي تمثل بالظبط افضل بيست فيت لاين





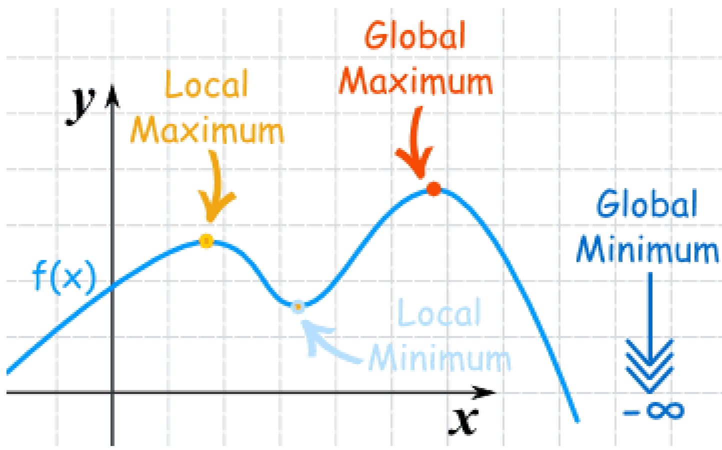
- الطريقة المستخدمة عموما في تعليم الالة كالتالي :
  - بيانات للتدريب
  - الالة تتعرف علي الموديل
  - يتم تكوين النظرية او المعادلة في الالة
  - الان اذا اعطيناها اكس (المعطي) , تعطينا واي (المطلوب)

- فكرة الـ **Gradient Descent**
  - طالما نحن نبحث عن قيم ثيتا 1 و 2 التي ستقلل قيمة J باقصي قدر , فسنفرض قيم لثيتا 1 و 2 , ثم نقوم بتقليلها تدريجيا حتي نصل لاقل قيمة للـ J
  - لكن حاذر اننا قد نقل لقيم مختلفة قليلة للـ J حسب من اين سنبدا



  - فهناك ما يسمي local minimum يعني قيمة دنيا , لكن محلية (علي اليمين) و لا نري جوارها اي قيم دنيا اخري , لكن في الحقيقة هناك قيم اقل منها لكن ابعد

- و القيمة الأقل جميعا اسمها global minimum وهي المطلوبة , مثل هذا الشكل



- و معادلة الوصول لافضل قيمة هي :

repeat until convergence:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

- اولا فكرة ان =: معناها ان يا برنامج , اعمل overwrite للقيمة اليسري بقيمة اليمني
- الفا هو معامل , ان زاد , ستكون الخطوات اسرع (و غالبا اقل دقة) و ان قل ستكون الخطوات ابطئ و ادق
- ما هو علي يمين الفا , هو اشتقاق جزئي للدالة , بالنسبة لثيتا
- لاحظ ان هذه المعادلة تكون مرة لثيتا صفر , ومرة لثيتا 1
- كلا المعادلتين يمشيان بالتوازي معا , ويتم تكرارهما معا
- ولاحظ ان الابديت لازم يتم زي اللي علي الشمال , مش اليمين

Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Incorrect:

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \boxed{\alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)}$$
$$\theta_1 := \text{temp1}$$

- لان اللي علي الشمال , بتدخل قيم ثيتا 0 و 1 الحالية مع بعض , بينما علي اليمين الغلط , بتدخل في المعادلة التانية قيمة ثيتا 0 المعدلة , وده هيلغبط الدنيا
- لاحظ ان في حالة القيمة الدنيا قيمة التفاضل بصفر (لأن وقتها هيكون الخط شبه مستقيم فالميل هيكون تقريبا 0)

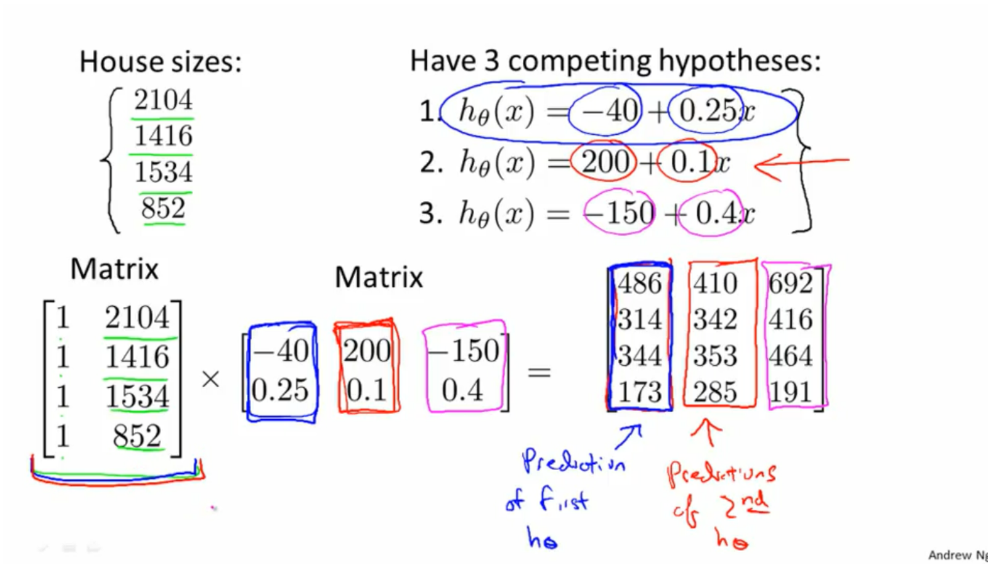
- لاحظ ان قيمة التفاضل تقل كلما قل الميل (التفاضل هو ميل الخط المستقيم ,فتدريجيا هيقل قيمة التفاضل لتغير الميل) , وكلما اقترب من القيمة الدنيا , فلا داعي لتقليل الالفا , فالقيمة نفسها ستقل تدريجيا
- نقوم بالتفاضل , نجد انها ستكون :

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$

- رقم 2 راح لانه اتضرب في الاس , ولاحظ اننا بنفاضل بالنسبة لثيتا 0 في الاول و ثيتا 1 في التانية فبقت كدة

- الـ Vectors
  - مش المقصود بيها المتجهات , لكن يقصد بها انها عمود في مصفوفة , او كانها مصفوفة , ذات عمود واحد , وبمقدار مختلف من الصفوف
  - و حينما يشار الي y2 مثلا , فيقصد بها القيمة الثانية (الصف الثاني) في الفيكتور
- الـ Matrix
  - يستخدم ضرب الماتركس في شئ مهم هنا , لو عندي مساحات 4 بيوت علي اليسار , وعندي 3 معادلات للتقييم علي اليمين , فممكن اعمل لهم تقييم كامل سريع عبر تكوين ماتركس عمودها الاول وحايد (لانها ستضرب في ثيتا 0) , عمودها الثاني مساحات البيوت , والماتركس التاني عدد عواميدها هي عدد معادلات التقييم , بها صفين , فيها قيمة ثيتا 0 و ثيتا 1 , لكل معادلة
  - ستجد ان ضرب الماتركسين في بعضهما , ستكون ماتركس جديدة , قيمة العمود الأول فيها هو اسعار البيت الاول للتقييمات الثلاث , وهكذا



- فمن المهارات المطلوبة لمبرمج الـ ML ان يكون لديه القدرة علي صياغة مصفوفة و تشكيلها بطرق عديدة , حتي يتمكن من استخدام ضربها و جمعها الترانزبوس , ليصل للنتيجة المطلوبة
