

What is machine learning
Supervised learning
Unsupervised learning
Cost function
Gradient Descent
Linear Algebra

Machine Learning

ال machine learning هو ازاى تخلقى الاله الى عندك تتعلم من نفسها من غير ما تديها برامج او حاجات محددة يعنى بمعنى اصح انت بتدربها على بعض الحاجات وهى المفروض تطور نفسها بنفسها من خلال التجارب التي بتمر عليها وتتعلم من الماضى زى الانسان كده معنا مبنتلعلمش بس قسطه.

By Tom Mitchell:

He says, a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E

انى الكومبيوتر يتعلم من خلال التجارب التي مر عليها زى مثلا لو بنعمل clarify emails as spam or not الاول الكومبيوتر بيتعلم انتا ازاى بتصنيف ال emails اما كده او كده بعدها تبتدي تجرب بعض ال tasks الجديدة الى هو مشفهاش ومن خلال ال performance بتاعه الى هو الميلات الى صنفها صح واللى مصنفهاش صح.

.Example: playing checkers

E = the experience of playing many games of checkers

.T = the task of playing checkers

.P = the probability that the program will win the next game

:In general, any machine learning problem can be assigned to one of two broad classifications

.Supervised learning and Unsupervised learning

ال machine learning بيتقسم ل تلت حاجات مهمه
1- ال supervised learning وده ببساطة انى بديله داتا وبديله بعض ال labels الخاصة بالداتا ديه عشان يبتدى انه يتعلم منها ويقدر ياخذ اكشن بعد كده وبرضه انا بتابع الكلام ده مع ال model

● Linear Regression

- The term Supervised Learning refers to the fact that we gave the algorithm a data set in which the, called, "right answers" were given.

عشان فى الاول انتا بتديله داتا بقيم حقيقية زى اسعار الشقق ك output مقابل مميزات فى الشقق وهو بعدها بيبتدى يتعلم مع نفسه من الحاجات ديه عشان لما تديله داتا جديده يقدر يفيدك ويقولك سعر شقة ما كام.

اولا كده بالنسبه لل machine learning الجزء الاول فيه الا وهو ال supervised learning يعتمد فى التوقع بتاعه على جزئين مهمين الا وهما ال input and output ويحصل فى ال supervised حاجتين مهمين اما انى محتاج اعمل classification or regression ال regression الا وهو التوقع وده ببساطة يشتغل على فكره انه بياخد بيانات ليها علاقه ببعض البيانات ديه عباره عن features وهو بيتدى يشتغل على الكلام ده كويس جدا وذاكره ويتعلم منه على سبيل المثال مثلا اسعار الشقق وعندى حوالى 10 الاف raw من ال data الخاصه بالشقق متسجل فيها مكان الشقه والمساحه وغير بقا من الميزات وفى المقابل ال input بتاع ال features ديه بيكون فيه output لل model برضه الى هو اسعار كل شقه منهم اد ايه فالموديل بيشتغل على الكلام ده وفى الآخر بيحصل عملية test انك بتشوف بقا شقه جديده بمواصفات جديده وتديها للموديل فيبدأ انه يعمل predict ليها انى سعرها اد كذا.

ال regression بيتوقع النتائج ديه بطرق مختلفه ممكن تكون linear او curved وغيره

By regression problem, I mean we're trying to predict a continuous valued output.

بعض التطبيقات المختلفه لل regression حاجه زى اسعار المنازل والأرصاء الجوية والمشتريات بالنسبة لعميل جديد بناء على مواصفات العملاء الى من النوع ده.

بينما بقا ال classification هو انه بيحاول يقسم الحاجه لجروبات مختلفه زى الصور مثلا وانه يحاول يعمل تصنيف لصور القطط او الكلاب فلما تيجى تديله صورته جديده يقولك لا ديه صورته كلب او صورته قطه وهكذا.

The term classification refers to the fact, that here, we're trying to predict a discrete value output zero or one, malignant or benign.

يعنى فى الآخر انا بحاول انى اتوقع قيم منفصله عن بعضها لكن ال regression هو عباره عن انى بحاول اتوقع سعر حاجه ممكن تتغير مع تغير ال data الى داخله ليا انما ال classification هو توقع حاجه معينه مثلا من set او مثلا التوقع ده بكون binary اما يحصل او لا.

So this is an example of a supervised learning algorithm. And it's supervised learning because we're given the, quotes, "right answer" for each of our examples. Namely we're told what was the actual house, what was the actual price of each of the houses in our data set were sold for and moreover, this is an example of a regression problem where the term regression refers to the fact that we are predicting a real-valued output namely the price.

2- ال unsupervised learning وده انا بديله داتا وهو المفروض يتعلم منها من غير ما اديله اى label او حاجه يتعلم منها هو بيتدى يقسم الداتا ويتعلم مع نفسه.

انتا بتديله الداتا وهو بيحاول يلاقي structure مختلفه فى الداتا ديه بمعنى بيحاول انه يقسم الداتا لجروبات بناء على الحاجات الى ليها علاقه ببعض.

So this is Unsupervised Learning because we're not telling the algorithm in advance that these are type 1 people, those are type 2 persons, those are type 3 persons and so on and instead what were saying is yeah here's a bunch of data. I don't know what's in this data. I don't know who's and what type. I don't even know what the different types of people are, but can you automatically find structure in the data from the you automatically cluster the individuals into

these types that I don't know in advance? Because we're not giving the algorithm the right answer for the examples in my data set, this is Unsupervised Learning.

Unsupervised Learning

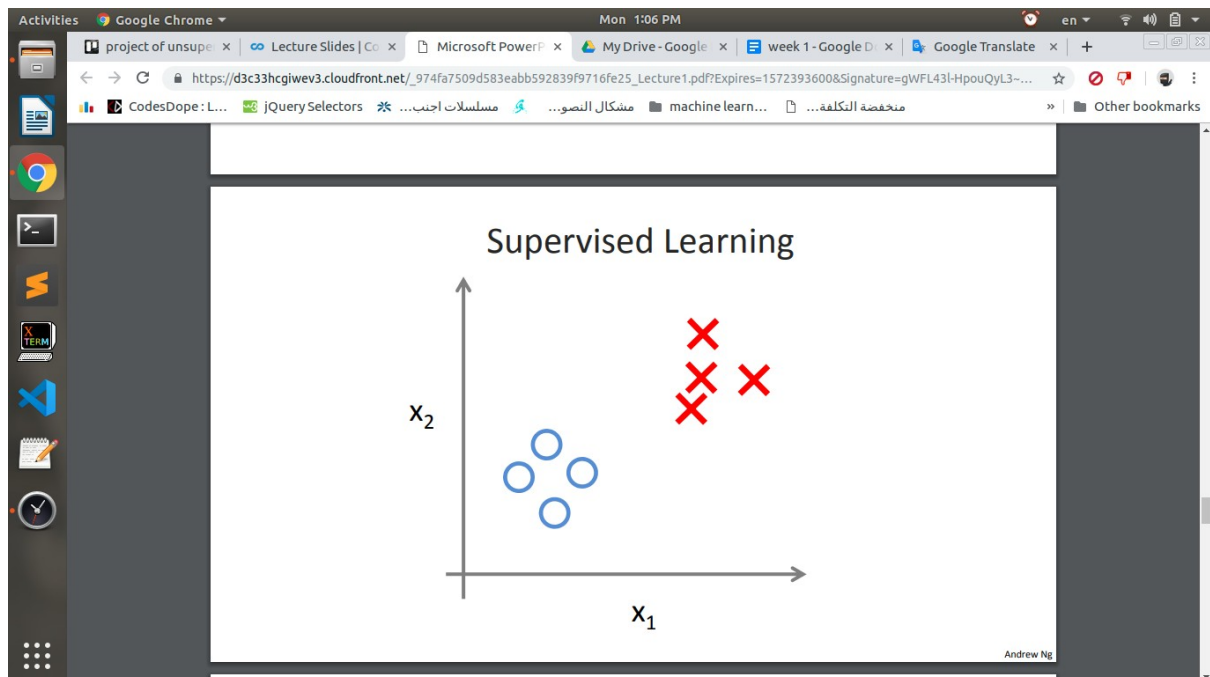
Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.

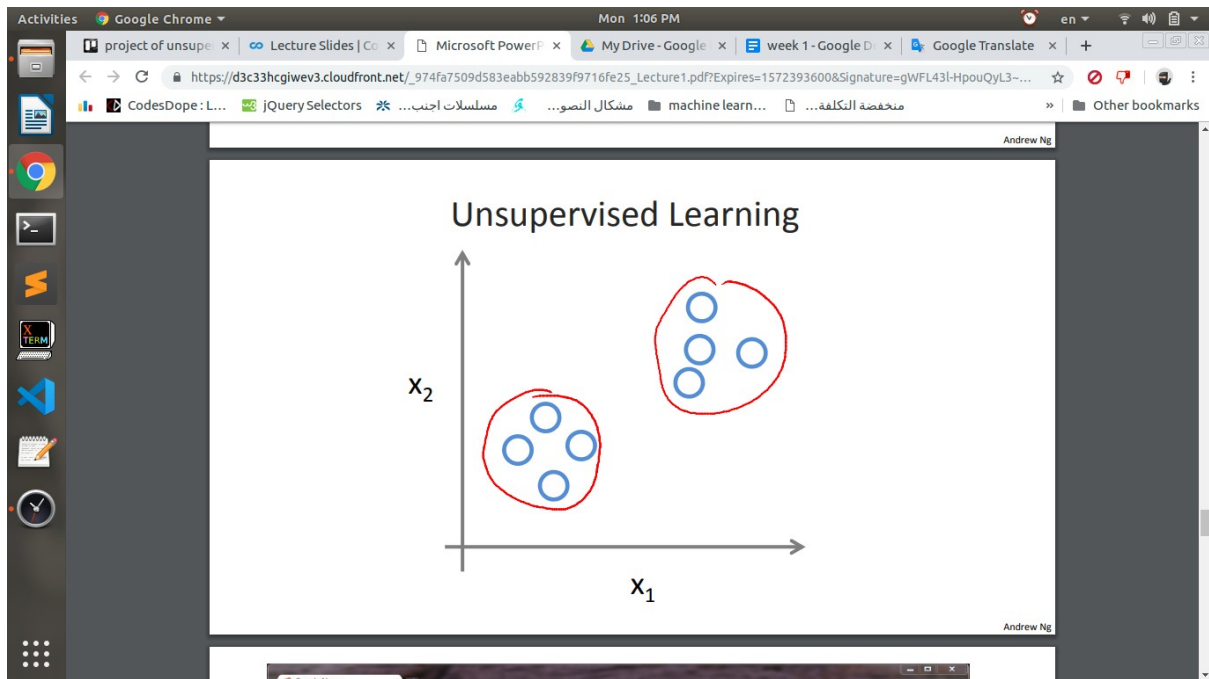
We can derive this structure by clustering the data based on relationships among the variables in the data.

With unsupervised learning there is no feedback based on the prediction results.

Clustering: Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on.

Non-clustering: The "Cocktail Party Algorithm", allows you to find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a [cocktail party](#)).





Training set of housing prices (Portland, OR)

Size in feet² (x)

Price (\$) in 1000's (y)

→ 2104

1416

→ 1534

852

...

460

232

315

178

...

$m = 47$

Notation:

→ m = Number of training examples

→ x 's = "input" variable / features

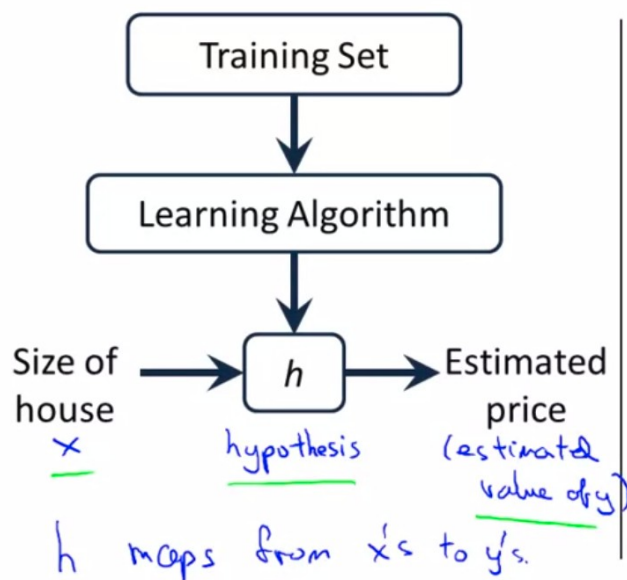
→ y 's = "output" variable / "target" variable

(x, y) - one training example

$(x^{(i)}, y^{(i)})$ - i th training example

$$\begin{aligned} x^{(1)} &= 2104 \\ x^{(2)} &= 1416 \\ y^{(1)} &= 460 \end{aligned}$$

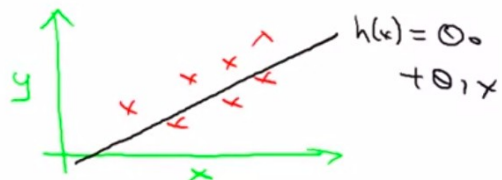
Andrew Ng



How do we represent h ?

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Shorthand: $h(x)$



Linear regression with one variable. (x)
Univariate linear regression.
 ↳ one variable

Activities Google Chrome Mon 8:33 PM en

Machine Le x Genral Tas x Microsoft x My Drive - x week 1 - Go x Google Tra x Lecture Sli x _ec21cea3 x

https://d3c33hcgivew3.cloudfront.net/_ec21cea314b2ac7d9e627706501b5baa_Lecture2.pdf?Expires=1572393600&Signature=dkMaCWz1kiPLS-6cz...

CodesDope: L... JQuery Selectors ... مسلسلات اجنب... مشكالات النصوص... machine learn... ... منخفضة التكلفة

Other bookmarks

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

Parameters:
 θ_0, θ_1

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

Goal: minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1

Simplified

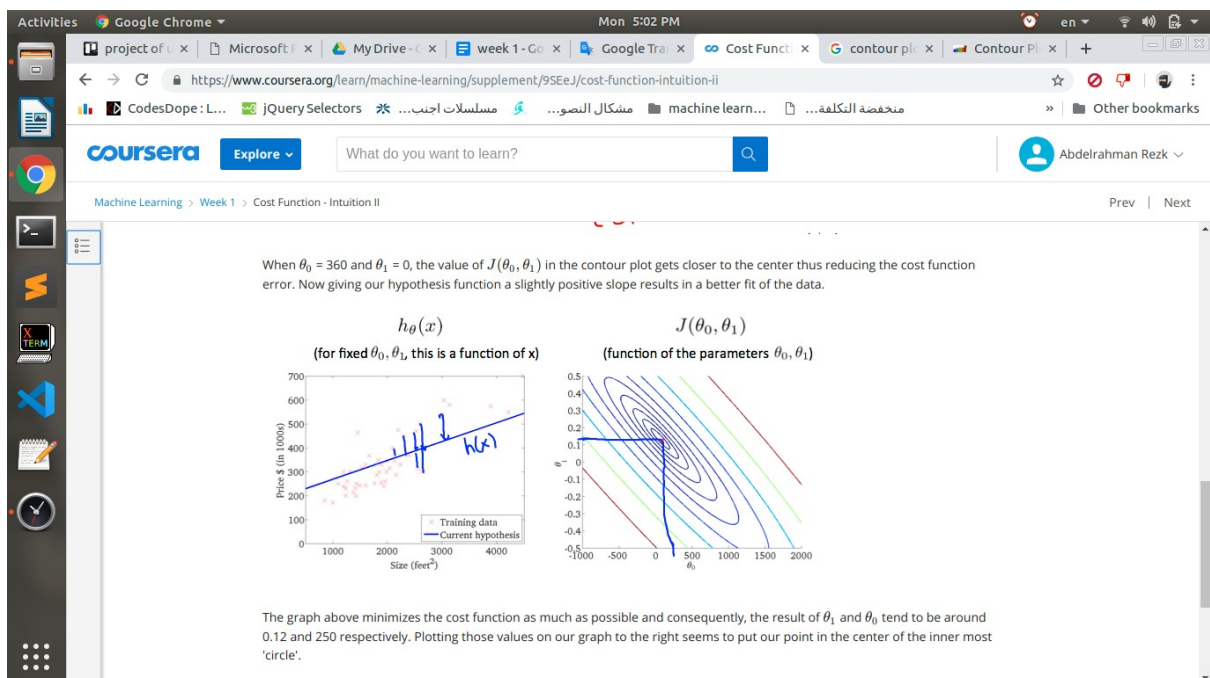
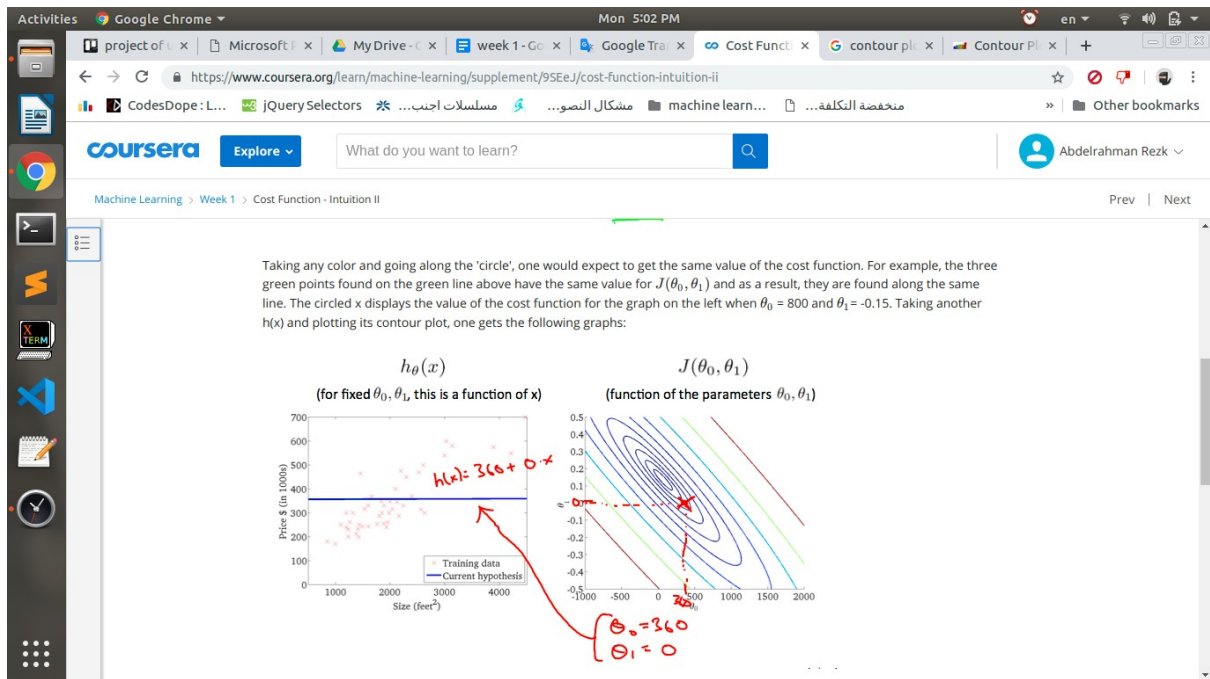
$h_{\theta}(x) = \theta_1 x$
 $\theta_0 = 0$

θ_1

$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$
 θ_1
 $\theta_1 x^{(i)}$

Andrew Ng

Lecture2.pdf Show all



طيب دلوقتي انا عرفت ازاى بحسب ال cost function بعد اما اختار قيم θ_0 و θ_1 واشتغل على المعادله بتاعت ال $J(\theta_0, \theta_1)$ ولكن انا لو عندي داتا كثيره مش هقعد اشتغل manually انى اجرب كل نقطه و اشوفها بتعمل ايه معايا بناء على قيم ال θ الى اختارتها وهنا بيحى دور Algorithm called gradient descent وهو عبارة عن خوارزمية بتحاول انها تقلل نسبة الخطأ على قدر ما تقدر.

Activities Google Chrome Mon 6:06 PM

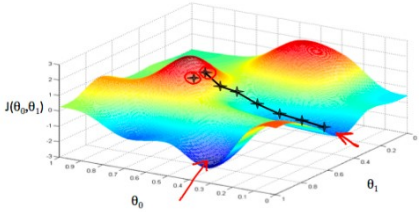
Machine Learning > Week 1 > Gradient Descent

Gradient Descent

So we have our hypothesis function and we have a way of measuring how well it fits into the data. Now we need to estimate the parameters in the hypothesis function. That's where gradient descent comes in.

Imagine that we graph our hypothesis function based on its fields θ_0 and θ_1 (actually we are graphing the cost function as a function of the parameter estimates). We are not graphing x and y itself, but the parameter range of our hypothesis function and the cost resulting from selecting a particular set of parameters.

We put θ_0 on the x axis and θ_1 on the y axis, with the cost function on the vertical z axis. The points on our graph will be the result of the cost function using our hypothesis with those specific theta parameters. The graph below depicts such a setup.



Activities Google Chrome Mon 6:11 PM

Machine Learning > Week 1 > Gradient Descent

We will know that we have succeeded when our cost function is at the very bottom of the pits in our graph, i.e. when its value is the minimum. The red arrows show the minimum points in the graph.

The way we do this is by taking the derivative (the tangential line to a function) of our cost function. The slope of the tangent is the derivative at that point and it will give us a direction to move towards. We make steps down the cost function in the direction with the steepest descent. The size of each step is determined by the parameter α , which is called the learning rate.

For example, the distance between each 'star' in the graph above represents a step determined by our parameter α . A smaller α would result in a smaller step and a larger α results in a larger step. The direction in which the step is taken is determined by the partial derivative of $J(\theta_0, \theta_1)$. Depending on where one starts on the graph, one could end up at different points. The image above shows us two different starting points that end up in two different places.

The gradient descent algorithm is:

repeat until convergence:

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

where

$j=0,1$ represents the feature index number.

Activities Google Chrome Mon 6:11 PM

Machine Learning > Week 1 > Gradient Descent

repeat until convergence:

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

where

$j=0,1$ represents the feature index number.

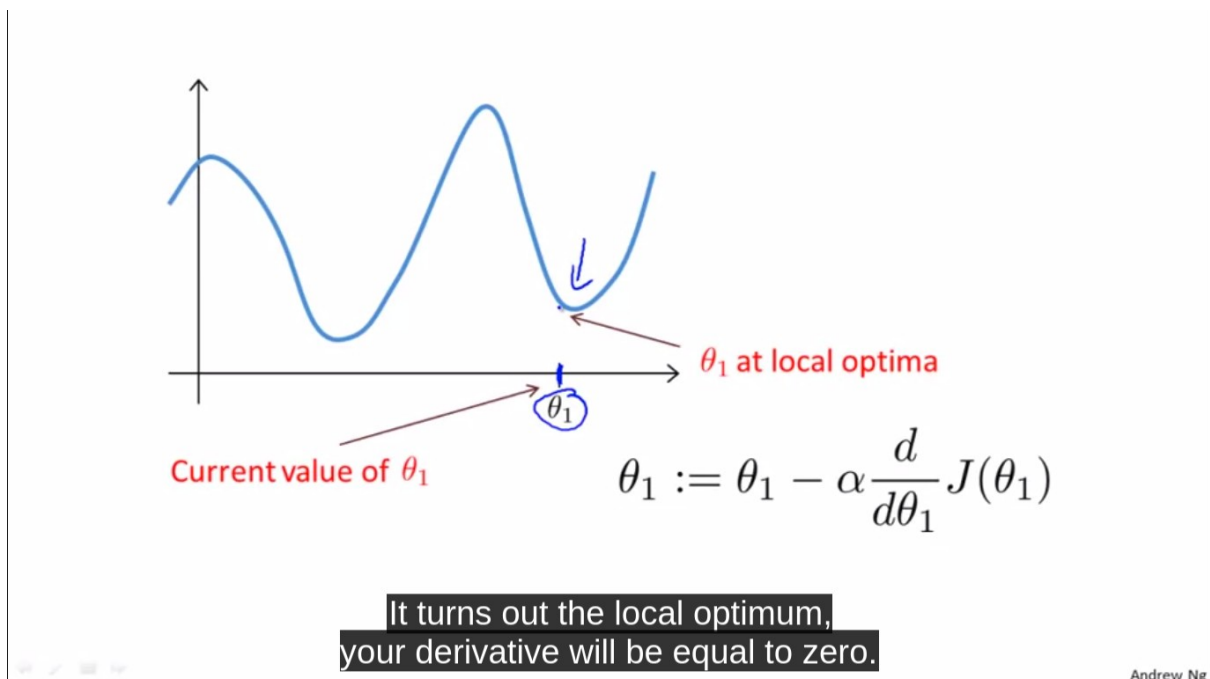
At each iteration j , one should simultaneously update the parameters $\theta_1, \theta_2, \dots, \theta_n$. Updating a specific parameter prior to calculating another one on the j^{th} iteration would yield to a wrong implementation.

Correct: Simultaneous update	Incorrect:
$\rightarrow \text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$	$\rightarrow \text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
$\rightarrow \text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$	$\rightarrow \theta_0 := \text{temp0}$
$\rightarrow \theta_0 := \text{temp0}$	$\rightarrow \text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
$\rightarrow \theta_1 := \text{temp1}$	$\rightarrow \theta_1 := \text{temp1}$

ال derivative part الى فى المعادلة ده هو اللى يساعدنى ويبخلىنى اعرف انا هامشى ازاي عشان اوصل لل minimum error وده من خلال انى لما بختار نقطة ما بروج اشوفه بتمس الخط ازاي واشوف ال slope بتاعها اما positive or negative ومن خلاله بعرف انا هامشى ازاي.

لما ال gradient descent و ال theta بتاعتى بتوصل لل minupmum optimal bottom الى هو بتقلل ال error لأكثر حاجه ممكنه ساعتها ال slope الناتج من ال derivative بيكون ب 0 وبعد كده قيمه ال theta مش هتتغير.

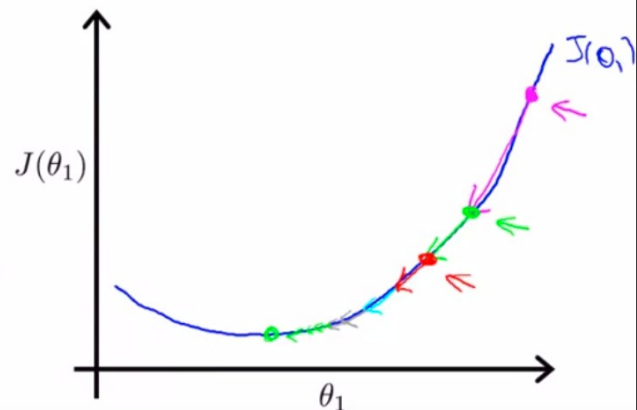
فى ال gradient descent انتا كل اما بتاخذ خطوة فى اتجاه انك تقلل ال error الخطوط نفسها بتقل لانك فى كل مرة بتضرب ال learning rate فى ال derivative الجديدة الناتج عن ال slope الجديد وهكذا.



Gradient descent can converge to a local minimum, even with the learning rate α fixed.

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.



انتا هنا مش محتاج انك تقلل ال learning rate لانك كده كده فى كل مره الخطوة نفسها بتقل بسبب انى قيمة ال derivative بتقل لذلك ال learning rate can be fixed

Activities Google Chrome Mon 8:47 PM

Machine Learning x Genral Tas x Microsoft x My Drive - x week 1 - Go x Google Tra x Lecture Sli x _ec21cea3 x

https://d3c33hcgivew3.cloudfront.net/_ec21cea314b2ac7d9e627706501b5baa_Lecture2.pdf?Expires=1572393600&Signature=dkMaCWz1kiPIS-6cz... Customise and control Google Chrome

Gradient descent algorithm

repeat until convergence {

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

(for $j = 1$ and $j = 0$)

}

Linear Regression Model

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

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

Andrew Ng

Activities Google Chrome Mon 7:23 PM

Machine Learning > Week 1 > Gradient Descent Intuition

Regardless of the slope's sign for $\frac{\partial}{\partial \theta_1} J(\theta_1)$, θ_1 eventually converges to its minimum value. The following graph shows that when the slope is negative, the value of θ_1 increases and when it is positive, the value of θ_1 decreases.

Handwritten notes for the top graph:

$$\theta_1 := \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} J(\theta_1) \geq 0$$

$$\theta_1 := \theta_1 - \alpha \cdot (\text{positive number})$$

Handwritten notes for the bottom graph:

$$\theta_1 := \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} J(\theta_1) \leq 0$$

$$\theta_1 := \theta_1 - \alpha \cdot (\text{negative number})$$

Activities Google Chrome Mon 7:24 PM

Machine Learning > Week 1 > Gradient Descent Intuition

On a side note, we should adjust our parameter α to ensure that the gradient descent algorithm converges in a reasonable time. Failure to converge or too much time to obtain the minimum value imply that our step size is wrong.

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

If α is too small, gradient descent can be slow.

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

Activities Google Chrome Mon 7:24 PM

Machine Learning > Week 1 > Gradient Descent Intuition

Parameter Learning

- Video: Gradient Descent 11 min
- Reading: Gradient Descent 3 min
- Video: Gradient Descent Intuition 11 min
- Reading: Gradient Descent Intuition 3 min
- Video: Gradient Descent For Linear Regression 10 min
- Reading: Gradient Descent For Linear Regression 6 min
- Review
- Reading: Lecture Slides 20 min
- Quiz: Linear Regression with One Variable

How does gradient descent converge with a fixed step size α ?

The intuition behind the convergence is that $\frac{d}{d\theta_1} J(\theta_1)$ approaches 0 as we approach the bottom of our convex function. At the minimum, the derivative will always be 0 and thus we get:

$$\theta_1 := \theta_1 - \alpha * 0$$

Gradient descent can converge to a local minimum, even with the learning rate α fixed.

$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.

ازای بقا انی احط ال gradient descent with cost function to fit best line in your data

Activities Google Chrome Mon 8:19 PM

Machine Learning > Week 1 > Gradient Descent For Linear Regression

When specifically applied to the case of linear regression, a new form of the gradient descent equation can be derived. We can substitute our actual cost function and our actual hypothesis function and modify the equation to:

repeat until convergence: {

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

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

}

where m is the size of the training set, θ_0 a constant that will be changing simultaneously with θ_1 and x_i, y_i are values of the given training set (data).

Note that we have separated out the two cases for θ_j into separate equations for θ_0 and θ_1 ; and that for θ_1 we are multiplying x_i at the end due to the derivative. The following is a derivation of $\frac{\partial}{\partial \theta_j} J(\theta)$ for a single example:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_{\theta}(x) - y)^2 \\ &= 2 \cdot \frac{1}{2} (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_{\theta}(x) - y) \\ &= (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^n \theta_i x_i - y \right) \\ &= (h_{\theta}(x) - y) x_j \end{aligned}$$

الجزء الكبير فى ال linear regression هو الرياضه او المعادلة الخطيمومعناها هو العلاقه بين متغيريين حين يكون كلا من هذان المتغيران لهما اوس = 1

استخدام الدوال فى الرياضه مع الماشين ليرننج بكون على حسب انا شغال على ايه فمثلا ممكن استخدم المعادله الخطيه لو هى دائما فى زياده لكن لو قدام مثلا الحاجه لما بتزيد بتقل بعد شويه فممكن تكون معادله من الدرجه الثانيه

وهكذا.

The screenshot shows the Coursera interface for the Machine Learning course, Week 2, Multiple Features. The left sidebar lists the course content, including 'Multivariate Linear Regression' and 'Computing Parameters Analytically'. The main content area displays the following text:

The multivariable form of the hypothesis function accommodating these multiple features is as follows:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

In order to develop intuition about this function, we can think about θ_0 as the basic price of a house, θ_1 as the price per square meter, θ_2 as the price per floor, etc. x_1 will be the number of square meters in the house, x_2 the number of floors, etc.

Using the definition of matrix multiplication, our multivariable hypothesis function can be concisely represented as:

$$h_{\theta}(x) = [\theta_0 \quad \theta_1 \quad \dots \quad \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

This is a vectorization of our hypothesis function for one training example; see the lessons on vectorization to learn more.

Remark: Note that for convenience reasons in this course we assume $x_0^{(i)} = 1$ for $(i \in 1, \dots, m)$. This allows us to do matrix operations with theta and x . Hence making the two vectors θ and $x^{(i)}$ match each other element-wise (that is, have the same number of elements: $n+1$).

Data Rescaling

هو انى احاول اخل جميع الداتا الى عندى وفيها ارقام ما بين ال -1 و 1 وده بيساعدنى انى الجراف نفسه بكون معقول وليه علاقه ببعضه على عكس اما يكون عندى قيم صغيره جدا وقيم كبيره جدا والصورتين الى تحت دول بيوضحوا قبل وبعد ال scaling.

The screenshot shows the Coursera interface for the Machine Learning course, Week 2, Gradient Descent in Practice I - Feature Scaling. The left sidebar lists the course content, including 'Gradient Descent in Practice I - Feature Scaling' and 'Gradient Descent in Practice II - Learning Rate'. The main content area displays the following text:

Two techniques to help with this are **feature scaling** and **mean normalization**. Feature scaling involves dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable, resulting in a new range of just 1. Mean normalization involves subtracting the average value for an input variable from the values for that input variable resulting in a new average value for the input variable of just zero. To implement both of these techniques, adjust your input values as shown in this formula:

$$x_i := \frac{x_i - \mu_i}{s_i}$$

Where μ_i is the **average** of all the values for feature (i) and s_i is the range of values (max - min), or s_i is the standard deviation.

Note that dividing by the range, or dividing by the standard deviation, give different results. The quizzes in this course use range - the programming exercises use standard deviation.

For example, if x_i represents housing prices with a range of 100 to 2000 and a mean value of 1000, then,

$$x_i := \frac{\text{price} - 1000}{1900}$$

Mark as completed

Normal Equation

ديه بتحل مشكله انى افترض قيم للثبات ومن خلالها بقدر اوصل لقيم الثبات الى انا عايزها عشان اشتغل على ال gradient descent واجيبه فى خطوه واحده بدل المشكله الى بتواجهنى فى تحديده ال learning rate ولكن المشكله هنا بتكون فى ال inverse العمليه بتاعته نفسها بتاخذ وقت كبير جدا كل اما الماتركس تكبر على عكس انى احاول مع

ال learning rate رغم انى معرفش عدد الخطوات الى هاخدھا قد ايه ولكن ممكن فى range ال 10 الاف وده الى هو عدد ال features اشتغل علطول بال normal question .
ولازم اخلى بالى لما اجى اجيب ال normal question ممكن لما اجى اجيب ال inverse الاقى الماتركس نفسها singular الى هو ملهاش inverse

تدريج البيانات Data Rescaling

$$x_i \leftarrow \frac{x_i - \mu_i}{s_i}$$

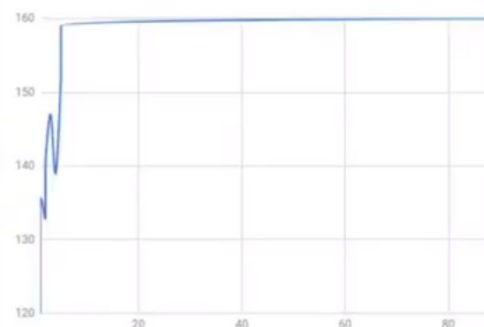
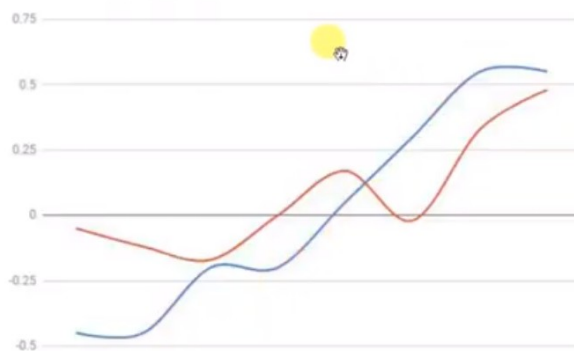
● فالحل اننا نخلى سكيل جميع القيم علي 1 بس بحيث تكون القيمة القصوي ليها واحد

									المتوسط	المدي
X	عدد الغرف	1	1	2	2	3	4	5	2.8	4
Y	السعر	120	135	133	140	147	139	153	140	39

X	عدد الغرف	-0.45	-0.45	-0.2	-0.2	0.05	0.3	0.55	0.55	0.55
Y	السعر	-0.05	-0.12	-0.17	0	0.17	-0.02	0.33	0.33	0.48

13

تدريج البيانات Data Rescaling



14

Gradient Descent in Practice II - Learning Rate

Activities Google Chrome Fri 8:12 AM

T227 MTA Review x Google Translate x Gradient Descent x Features and Poly x My Drive - Google x week 1 & 2 - Goog x + -

← → ↻ coursera.org/learn/machine-learning/supplement/TnHvV/gradient-descent-in-practice-ii-learning-rate

CodesDope: L... JQuery Selectors مسلسلات اجنب... مشكالات النصوص... machine learn... منخفضة التكلفة... Other bookmarks

coursera Explore What do you want to learn? Abdelrahman Rezk

Machine Learning > Week 2 > Gradient Descent in Practice II - Learning Rate Prev Next

Debugging gradient descent. Make a plot with *number of iterations* on the x-axis. Now plot the cost function, $J(\theta)$ over the number of iterations of gradient descent. If $J(\theta)$ ever increases, then you probably need to decrease α .

Automatic convergence test. Declare convergence if $J(\theta)$ decreases by less than E in one iteration, where E is some small value such as 10^{-3} . However in practice it's difficult to choose this threshold value.

Making sure gradient descent is working correctly.

It has been proven that if learning rate α is sufficiently small, then $J(\theta)$ will decrease on every iteration.

Activities Google Chrome Fri 8:13 AM

T227 MTA Review x Google Translate x Gradient Descent x Features and Poly x My Drive - Google x week 1 & 2 - Goog x + -

← → ↻ coursera.org/learn/machine-learning/supplement/TnHvV/gradient-descent-in-practice-ii-learning-rate

CodesDope: L... JQuery Selectors مسلسلات اجنب... مشكالات النصوص... machine learn... منخفضة التكلفة... Other bookmarks

coursera Explore What do you want to learn? Abdelrahman Rezk

Machine Learning > Week 2 > Gradient Descent in Practice II - Learning Rate Prev Next

It has been proven that if learning rate α is sufficiently small, then $J(\theta)$ will decrease on every iteration.

Making sure gradient descent is working correctly.

To summarize:

If α is too small: slow convergence.

If α is too large: $J(\theta)$ may not decrease on every iteration and thus may not converge.

إننا ممكن واننا شغال لما تشوف ال features تكرر انك تعمل features جديدة ممكن تكون تجميع ل لاكثر من feature زي مثلا الطول والعرض بتاع المنزل تخليهم الاتنين فى features واحد وتسميه المساحة وممكن انك ت create feature مش موجود بس يكون ليه علاقه بالحاجة ومؤثر فيها.

Activities Google Chrome FRI 8:31 AM

T227 MTA Review Sum... Google Translate Features and Polynomi My Drive - Google Drive week 1 & 2 - Google Do... + -

← → ↻ coursera.org/learn/machine-learning/supplement/ITznZ/features-and-polynomial-regression

CodesDope: L... JQuery Selectors مسلسلات اجنب... مشكالات النصوص machine learn... منخفضة التكلفة... Other bookmarks

coursera Explore What do you want to learn? Abdelrahman Rezk

Machine Learning > Week 2 > Features and Polynomial Regression Prev Next

- Descent in Practice I - Feature Scaling 3 min
- Video: Gradient Descent in Practice II - Learning Rate 8 min
- Reading: Gradient Descent in Practice II - Learning Rate 4 min
- Video: Features and Polynomial Regression 7 min
- Reading: Features and Polynomial Regression 3 min

Computing Parameters Analytically

Submitting Programming Assignments

Our hypothesis function need not be linear (a straight line) if that does not fit the data well.

We can **change the behavior or curve** of our hypothesis function by making it a quadratic, cubic or square root function (or any other form).

For example, if our hypothesis function is $h_{\theta}(x) = \theta_0 + \theta_1 x_1$ then we can create additional features based on x_1 , to get the quadratic function $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$ or the cubic function $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3$

In the cubic version, we have created new features x_2 and x_3 where $x_2 = x_1^2$ and $x_3 = x_1^3$.

To make it a square root function, we could do: $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 \sqrt{x_1}$

One important thing to keep in mind is, if you choose your features this way then feature scaling becomes very important.

eg. if x_1 has range 1 - 1000 then range of x_1^2 becomes 1 - 1000000 and that of x_1^3 becomes 1 - 1000000000

Mark as completed

ال features الى بتكرر معايا بطريقة متناسبة مع بعضها زي لما قسم حاجه على حاجه او الاتنين يكونوا بيتقسموا على رقم معين غالبا ال features ديه ممكن امسحها كلها واخلي واحد منهم فقط لان يعتبر مغيث غير feature واحد منهم هو الى هياثر معايا لانهم في تناسب مع بعضهم وده مش هياساعدنى بحاجه غير وقت و تكلفة على الفاضى يعنى محتاج اخلى بالى وانا شغال من كل حاجه