Abdelrahman Rezk

Web Developer

NLP & ML student

AOU University

فى week 3 هنبتدى بعد شغلنا على ال linear regression والى هو كان بيحاول يتوقع حاجه معينه بتختلف وملهاش قيمه او قيمه محدده بل هى توقع لقيمة ممكن تزيد او تقل على حسب ال features ولكنه فى الاخر نوع من انواع ال supervised learning واللى بيشتغل على حاجه زي توقع مثلا سعر شقه ما … نبتدى بقا نتكلم عن تانى حاجه بعد ال linear regression وهى ال classification problem وديه ليها ال algorithm معروف اسمه logistic regression وده بيتشغل على الحاجات الى بيكون قيمه محدده يعنى انا معايا برضه features ولكن ال y-actual هى قيم محدده ممكن تكون binary classification بمعنى انى تكون قيم ال y كلها اما 0 او 1 او اكتر من كده ولكنها فى الاخر بتكون قيم محددة بظبط زى انى احاول اعمل classification for mail is it a spam or not لذلك ديه تعتبر binary classification problem.
لازم اكون عارف انى ضرب الثيتات فى ال xs فى الاخر المفروض يطلع قيمه واحده.
اخلى بالى من موضوع ال data requlrization وهو انى بظبط البيانات الى عندى او اعرف انى هاستخدمها ازاى وده بيساعدنى فيه شخص متخصص فى المجال الى انا بحاول اعمله مودل او ماشين ليرننج

محتاج تركز فى ال sigmoid function وفي القيم الى بتخرجها والرسمة بتاعتها

In all of these problems the variable that we're trying to predict is a variable
y that we can think of as taking on two values either zero or one,
either spam or not spam, fraudulent or not fraudulent, related malignant or benign.

Classification

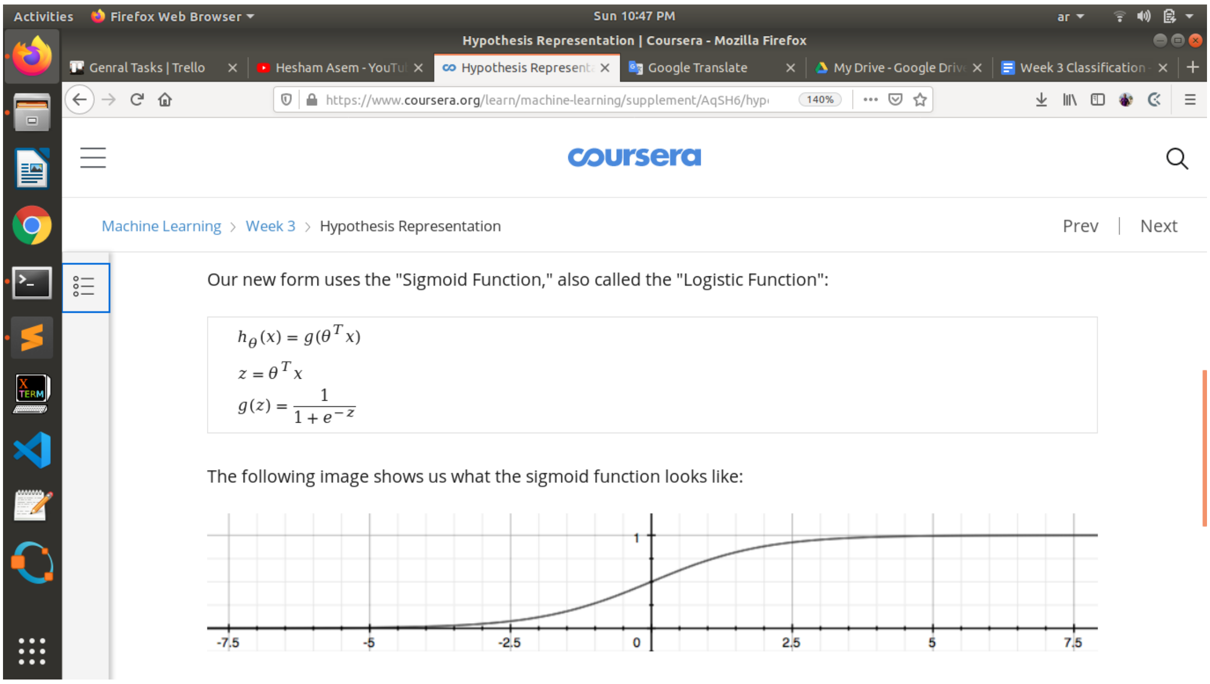
To attempt classification, one method is to use linear regression and map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0. However, this method doesn't work well because classification is not actually a linear function.

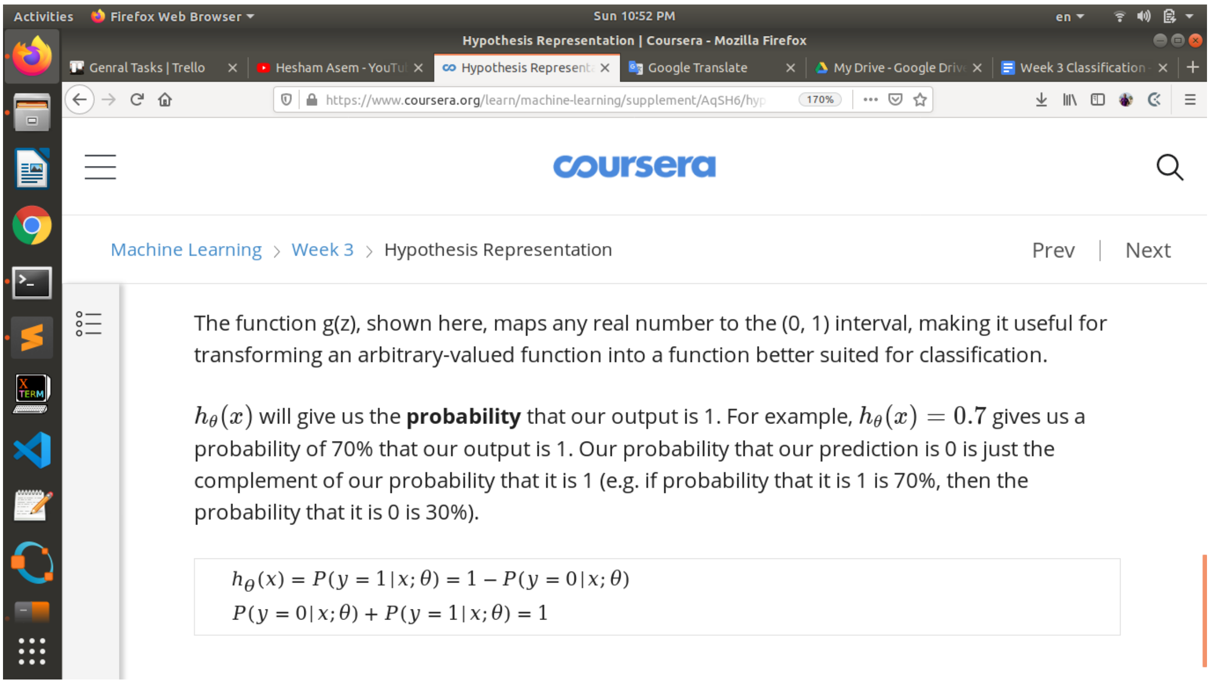
The classification problem is just like the regression problem, except that the values we now want to predict take on only a small number of discrete values. For now, we will focus on the **binary classification problem** in which y can take on only two values, 0 and 1. (Most of what we say here will also generalize to the multiple-class case.) For instance, if we are trying to build a spam classifier for email, then x(i)x^{(i)}x(i) may be some features of a piece of email, and y may be 1 if it is a piece of spam mail, and 0 otherwise. Hence, y∈{0,1}. 0 is also called the negative class, and 1 the positive class, and they are sometimes also denoted by the symbols “-” and “+.” Given x(i)x^{(i)}x(i), the corresponding y(i)y^{(i)}y(i) is also called the label for the training example.

Hypothesis Representation

We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given x. However, it is easy to construct examples where this method performs very poorly. Intuitively, it also doesn't make sense for hθ(x)h_\theta (x)hθ(x) to take values larger than 1 or smaller than 0 when we know that y ∈ {0, 1}. To fix this, let's change the form for our hypotheses hθ(x)h_\theta (x)hθ(x) to satisfy 0≤hθ(x)≤10 \leq h_\theta (x) \leq 10≤hθ(x)≤1. This is accomplished by plugging θTx\theta^Txθ Tx into the Logistic Function.

Our new form uses the "Sigmoid Function," also called the "Logistic Function":





https://www.coursera.org/learn/machine-learning/supplement/N8qsm/decision-boundary

https://www.coursera.org/learn/machine-learning/supplement/bgEt4/cost-function

انا بعد ما اعمل training لل data بتاعى وجبت قيم ال theta المناسبه الى هى بتطلع predication تمام وكمان ظبط ال cost function بنبتدى ناخد قيم ال theta ديه او نحفظها عشان لما نروح نعملها على test set بنشتغل بالقيم بتاعت الثيتات الى انا جبتها من ال training data

https://www.coursera.org/learn/machine-learning/supplement/0hpMl/simplified-cost-function-and-gradient-descent

https://www.coursera.org/learn/machine-learning/supplement/cmjIc/advanced-optimization

https://www.coursera.org/learn/machine-learning/supplement/HuE6M/multiclass-classification-one-vs-all

https://www.coursera.org/learn/machine-learning/supplement/VTe37/the-problem-of-overfitting

https://www.coursera.org/learn/machine-learning/supplement/1tJlY/cost-function

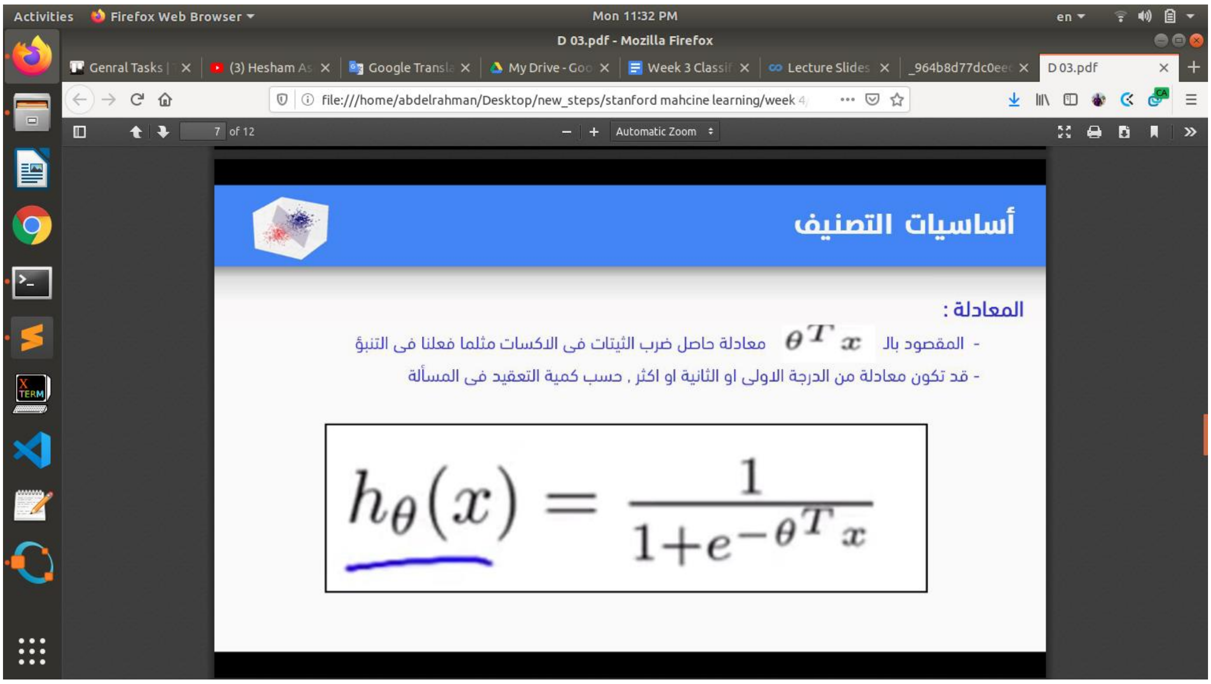
ملخص

ال classification هو نوع من انواع ال supervised learning الى بيكون مهتم اكتر بالحاجات الى ليها قيم محددة تسمى discreated values والى هى عباره عن مدخلات ومخرجات محددة غالبا ما بتكون binary classification عنى المخرجات اما حاجه من الاتنين ممكن نقول 0 و 1 وعن طريق ده ببتدى اشتغل وافصل الاتنين عن بعض عن طريق ال logistic regression.

Ln is standard for logarithm natural number

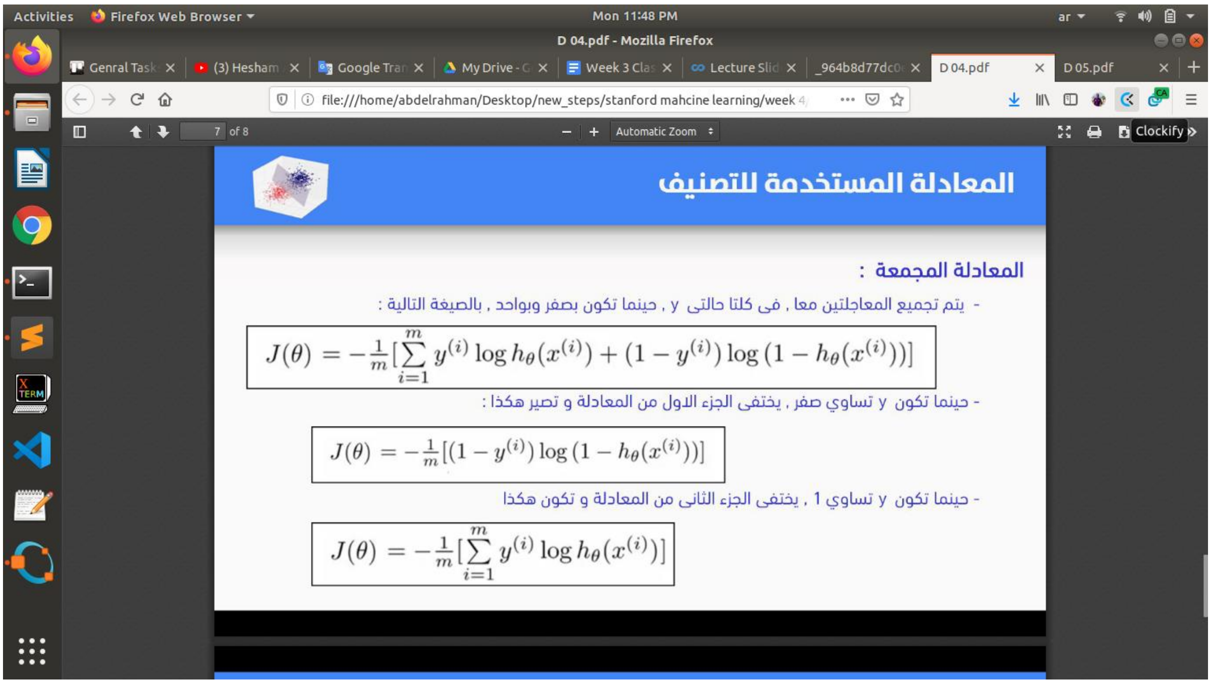
ال natural number هو ال e =2.718

عن طريق ال sigmoid function قدر اخلى قيم التوقعات عندى دايما بين ال 0 و 1 وده بيساعدنى انى اعمل classification وبناء عليها بشتغل على فكره ال probabilty انى لو كان مثلا الرقم 7. فبقربه مثلا ل 1 و هكذا

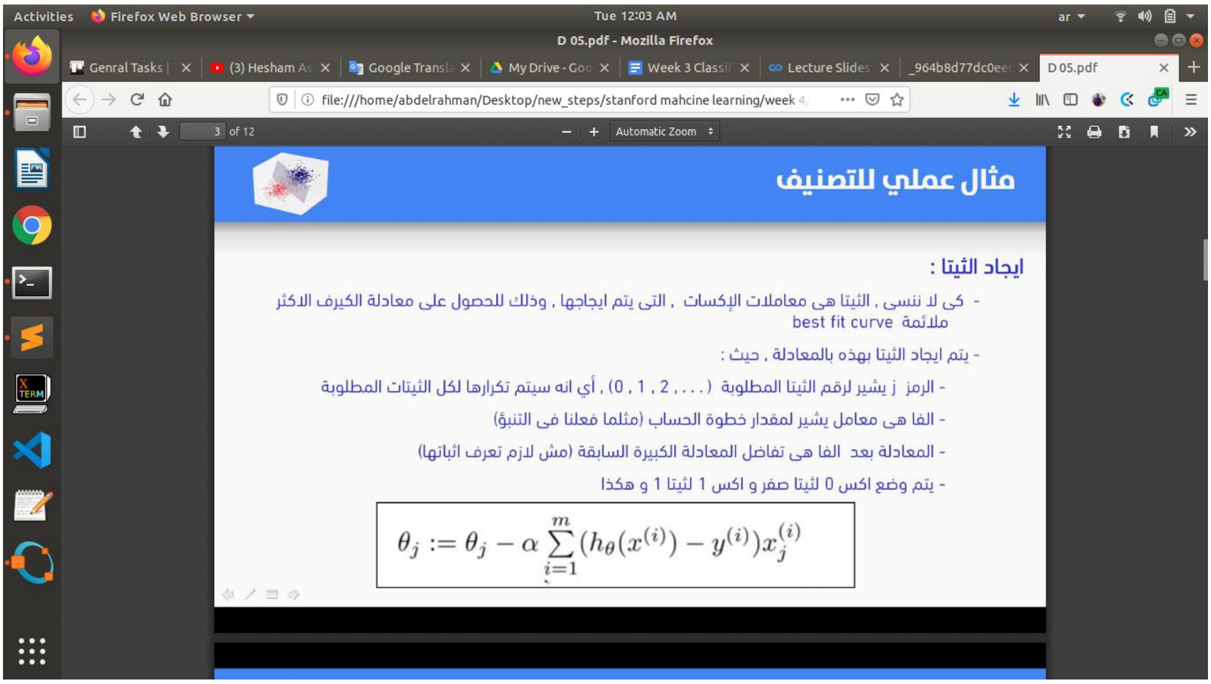


بعد كده بيتدى احسب ال cost function الى عن طريقها هاقدر اعرف الدنيا كويسه عندى ولا لا وده عن طريق المعادله المجمعه.

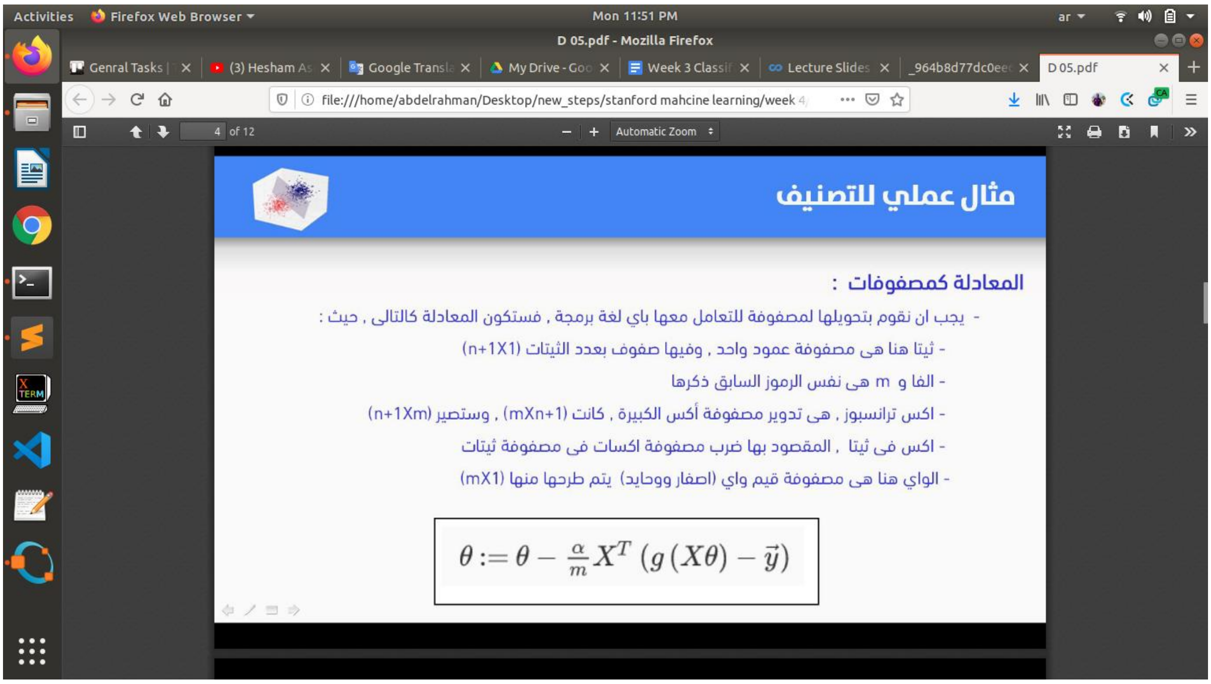
اخلى بالى من انى معادله التوقع الى فوق ديه هى الى بعد اما اعمل training للداتا هابتدى استخدمها عشان اتوقع اى داتا جديده بعد اما دربتها وجبت قيم الثيتات الى انا عايز ها.



بعد كده بيتدى اجيب قيم الثيتات الى بتعمل fit للمعادله عن طريق ال graident descent نفس المعادله يعتبر ولكن التعبير عنها مختلف فبستخدم فى القيمه المتوقعه ال sigmoid function



ولكنها بعد اما عملت اشتقاق ليها بقت بالمنظر ده

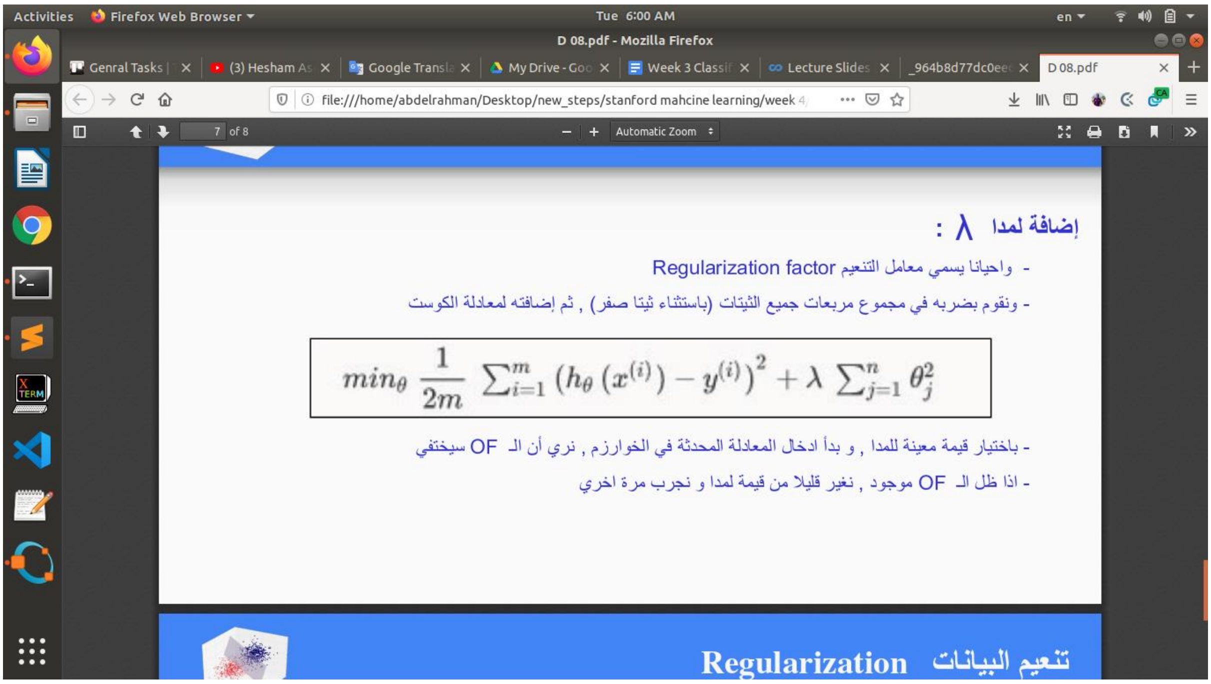


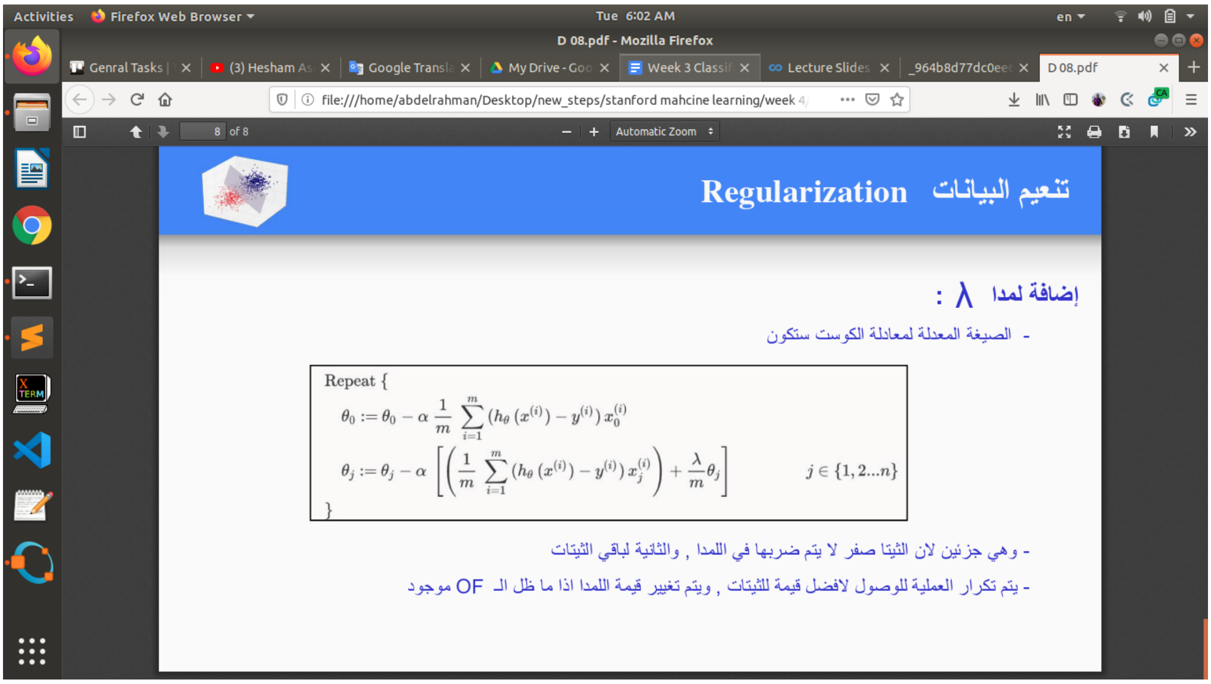
اخلى بالى من حاجتين مهمين هما ال over fitting and under fitting.

واحاول اعما regularization للداتا بالاستعانه بمتخصصين فى الحاجه الى انا شغال فيها لان ممكن يكون في بعض ال features الى مش مهمه واستخدمها انا فتضيع الدنيا.

ال regularization بيكون عن طريق واحد او اكتر من الحاجات ديه

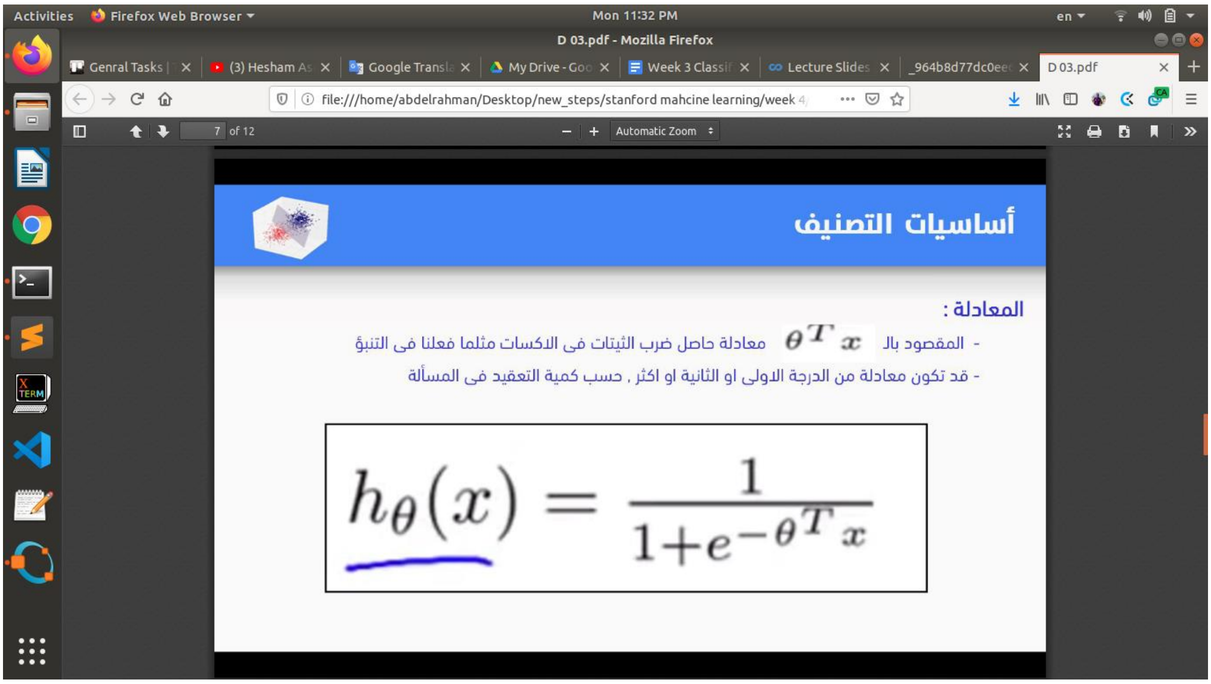

تنعيم البيانات Regularization
المفهوم :
- عملية تنقية البيانات بهدف التخلص من الضبط الزائد OF
- يتم عبر 3 طرق هامة, قد نقوم باحداها فقط او باثنين فقط او بالثلاثة معا :
* انتقاء البيانات
* تغيير المعاملات
* اضافة (لمدا)


إضافة لمدا λ :
- واحيانا يسمي معامل التنعيم Regularization factor
- ونقوم بضربه في مجموع مربعات جميع الثيتات (باستثناء ثيتا صفر) , ثم إضافته لمعادلة الكوست
$min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta (x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2$
- باختيار قيمة معينة للمدا , و بدأ ادخال المعادلة المحدثة في الخوارزم , نري أن الـ OF سيختفي
- اذا ظل الـ OF موجود , نغير قليلا من قيمة لمدا و نجرب مرة اخري
تنعيم البيانات Regularization



ال classification فى الحقيقة عامل زى ال regression عدا انى هو بيحاول يتوقع الحاجة ديه فى قيم صغيره ومنفصله ومحدده except that the values we now want to predict take on only a small number of discrete values.

لما بجى اشتغل انا هاتجاهل خالص انى اصلا قيم ال y ى قيم محدده وهاستخدم ال linear regression شان يعمل predict for y given x and parametrized by thetas ولكن انا بغير طريقه ال predication function الى كنت بستخدمها مع ال linear regression عن طرق ال logistic function or sigmoid function.



وده خلانى انى احول function بتطلع القيم بطريقة عشوائية ل function هاتطلع قيم محصوره بين قيمتين محددتين وهما 0 و 1 وبناء على ده هابتدى اشتغل كانها مسألة probability عن طريق لو كانت اكبر من قيم معينة تبقا 1 مثلا والعكس تبقا 0.

hθ(x) will give us the **probability** that our output is 1. For example, hθ(x)=0.7h_\theta(x)=0.7hθ(x)=0.7 gives us a probability of 70% that our output is 1. Our probability that our prediction is 0 is just the complement of our probability that it is 1 (e.g. if probability that it is 1 is 70%, then the probability that it is 0 is 30%).

Decision Boundary

عشان اقدر اخلى قيم ال hypothesis function تكون فعليا discrete values هانتحتاج نقول انى لو كان ال hypothesis بتاعنا اكبر من حاجه معينه يبقا 1 والعكس يبقا صفر

$h_\theta(x) \geq 0.5 \rightarrow y=1$

$h_\theta(x) < 0.5 \rightarrow y=0$

$g(z) \geq 0.5$

$When z \geq 0$

Facts

$z=0, e^0=1 \Rightarrow g(z)=1/2$

$z \rightarrow \infty, e^{-\infty} \rightarrow 0 \Rightarrow g(z)=1$

$z \rightarrow -\infty, e^{\infty} \rightarrow \infty \Rightarrow g(z)=0$

So if our input to g is $\theta^T X$ then that means:

$h_\theta(x)=g(\theta^T x) \geq 0.5$

$when \theta^T x \geq 0$

عشان فى ال sigmoid function القيمه ديه بتكون مرفوعه ك اس سالب فبتصغر جدا وبيتبقا 1 + القيمه الصغيره

From these statements we can now say:

$\theta^T x \geq 0 \Rightarrow y=1$

$\theta^T x < 0 \Rightarrow y=0$

The **decision boundary** is the line that separates the area where y = 0 and where y = 1. It is created by our hypothesis function.

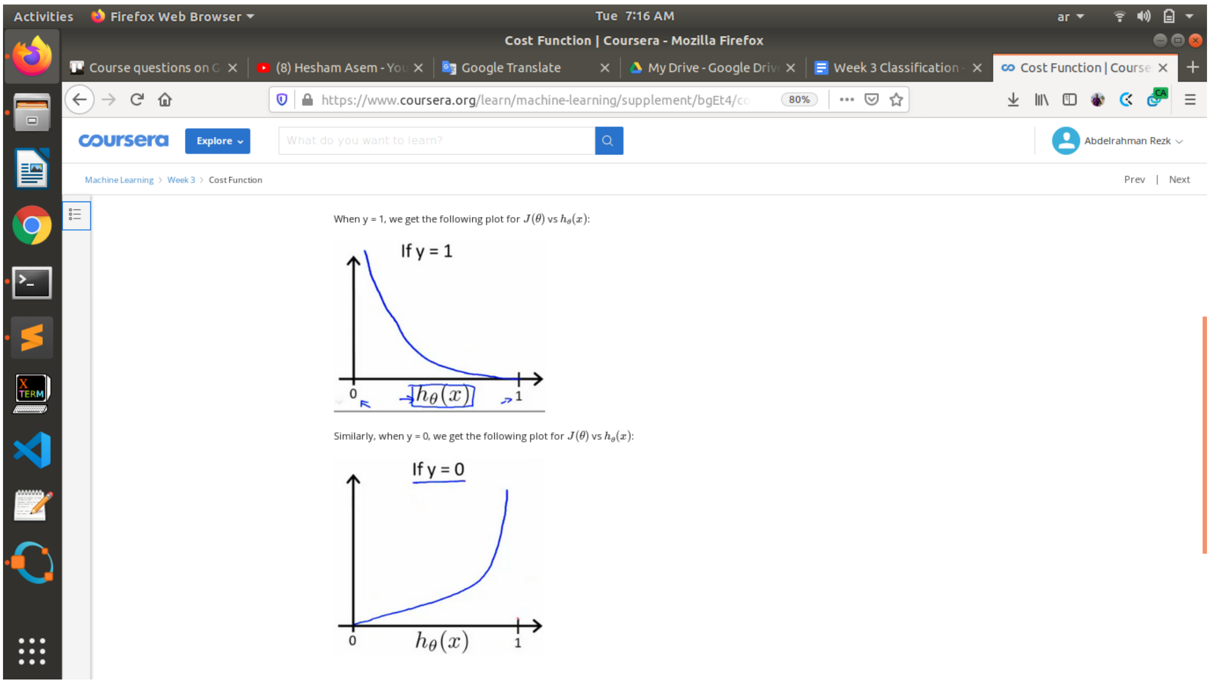
Cost Function

We cannot use the same cost function that we use for linear regression because the Logistic Function will cause the output to be wavy, causing many local optima. In other words, it will not be a convex function.

$Cost(h_\theta(x),y)=0 \text{ if } h_\theta(x)=y$

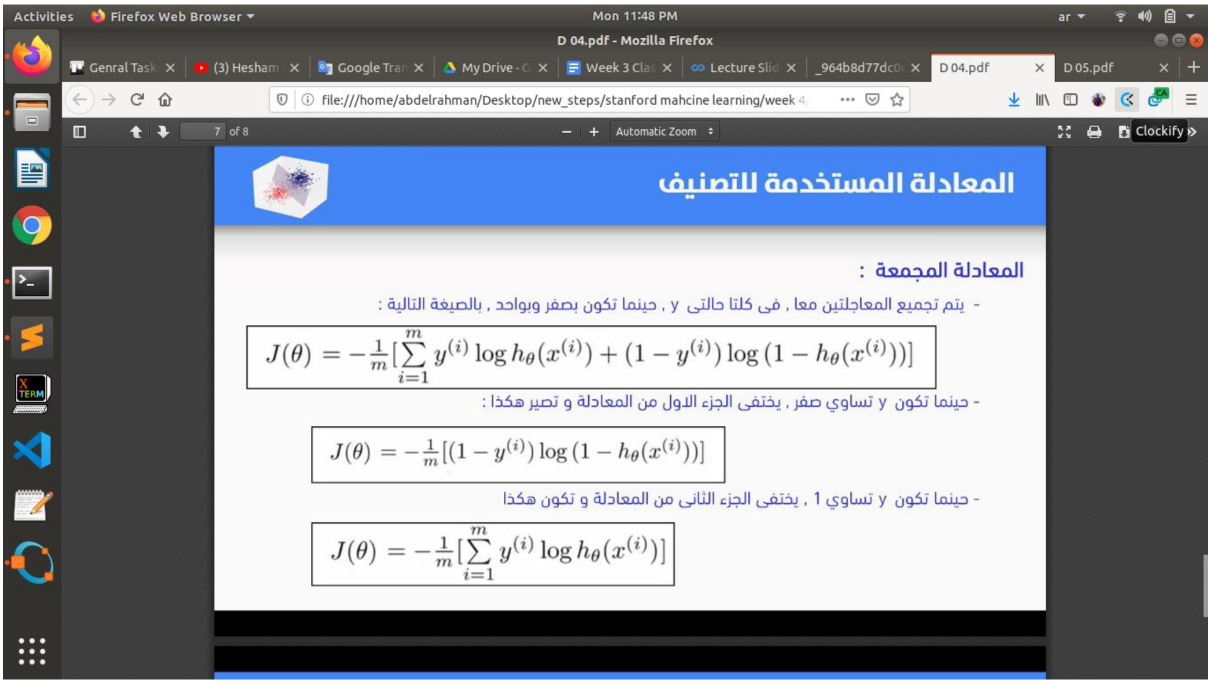
$Cost(h_\theta(x),y) \rightarrow \infty \text{ if } y=0 \text{ and } h_\theta(x) \rightarrow 1$

$Cost(h_\theta(x),y) \rightarrow \infty \text{ if } y=1 \text{ and } h_\theta(x) \rightarrow 0$



Simplified Cost Function and Gradient Descent

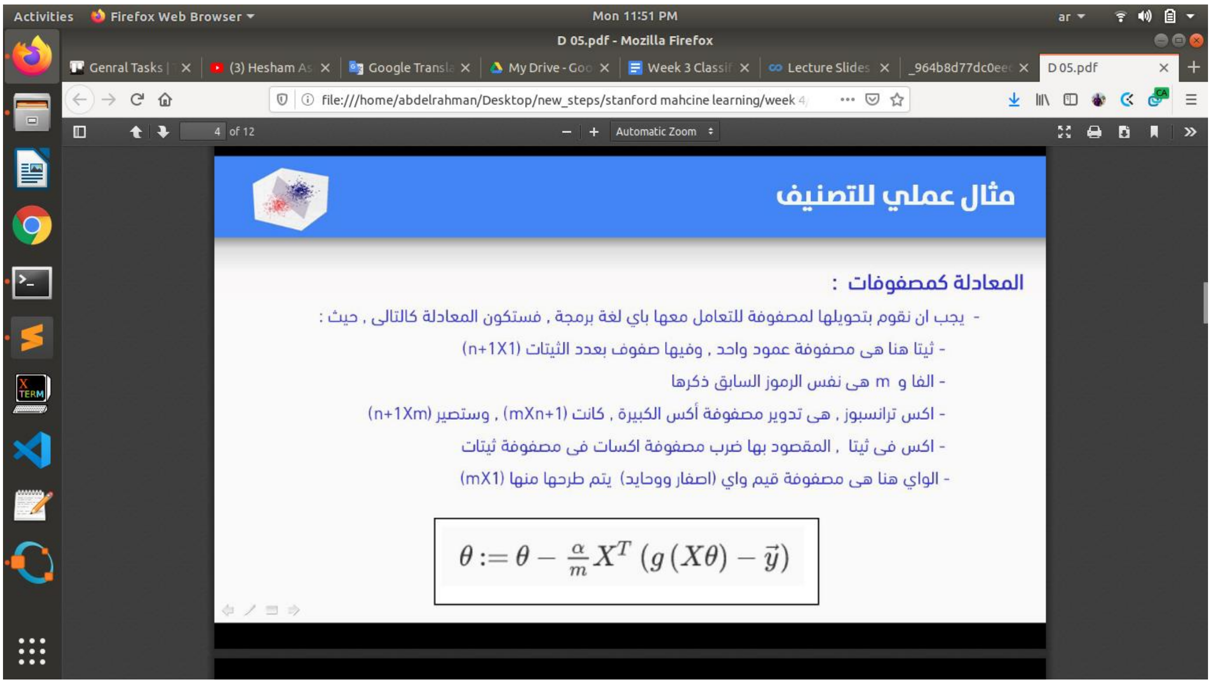
We can compress our cost function's two conditional cases into one case:



Instead of this form of gradient descent
Repeat{

$\theta_j := \theta_j - \alpha/m \sum (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

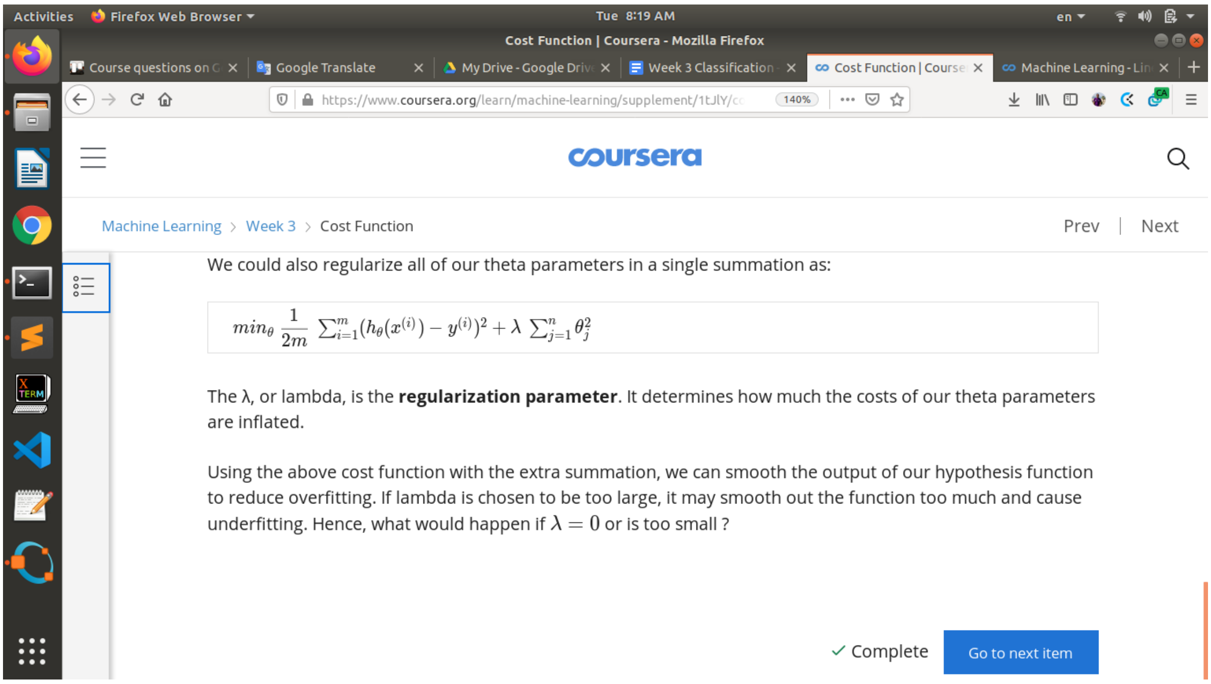
}
We use A vectorized implementation of gradient descent is



Multiclass Classification: One-vs-all

https://www.coursera.org/learn/machine-learning/supplement/HuE6M/multiclass-classification-one-vs-all

هنا انا بشتغل على فكره انها binary انى اخد مثلا كلاس معين اسميه 0 والباقى كله اسمه 1 وافصل اول كلاس ثم التانى مثلا اخد اسمه 1 واسمى الباقى كله 0 بلس الى انا عملته وافصل التانى وهكذا لحد n-classes



Machine Learning > Week 3 > Cost Function

We could also regularize all of our theta parameters in a single summation as:

$$min_\theta\ \frac{1}{2m}\ \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\ \sum_{j=1}^n \theta_j^2$$

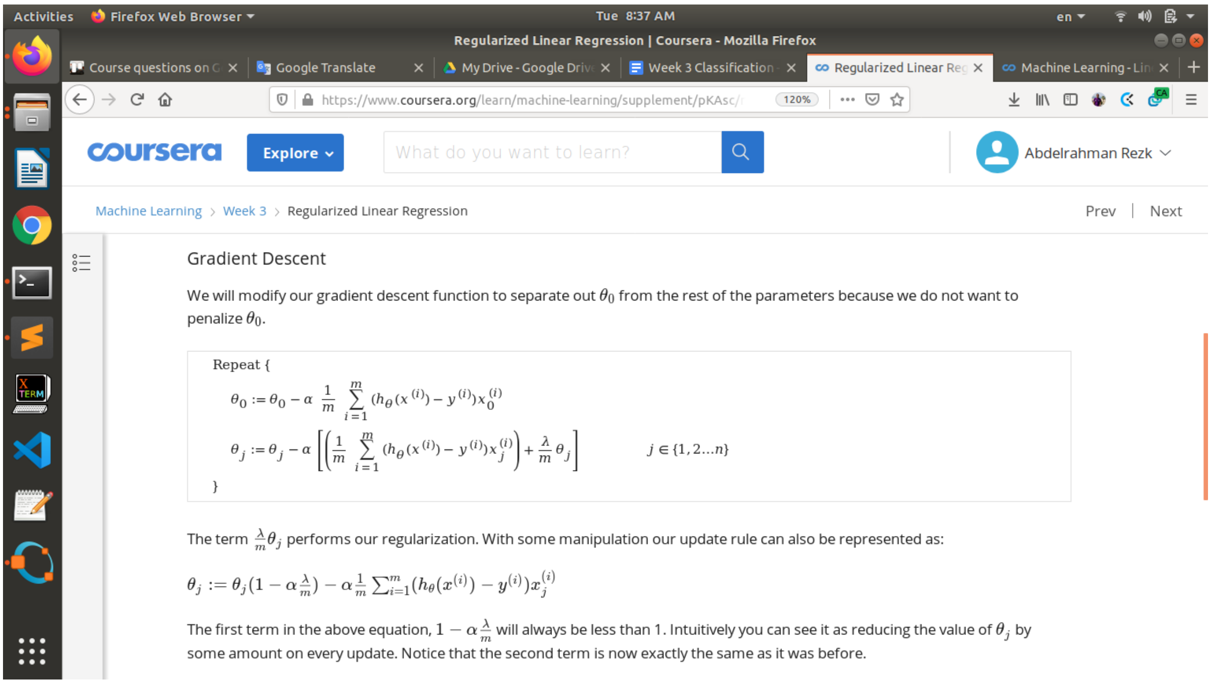
The λ, or lambda, is the **regularization parameter**. It determines how much the costs of our theta parameters are inflated.

Using the above cost function with the extra summation, we can smooth the output of our hypothesis function to reduce overfitting. If lambda is chosen to be too large, it may smooth out the function too much and cause underfitting. Hence, what would happen if $\lambda = 0$ or is too small ?

Regularized Linear Regression

Note: [8:43 - It is said that X is non-invertible if m ≤\leq≤ n. The correct statement should be that X is non-invertible if m < n, and may be non-invertible if m = n.

We can apply regularization to both linear regression and logistic regression. We will approach linear regression first.



Machine Learning > Week 3 > Regularized Linear Regression

Gradient Descent

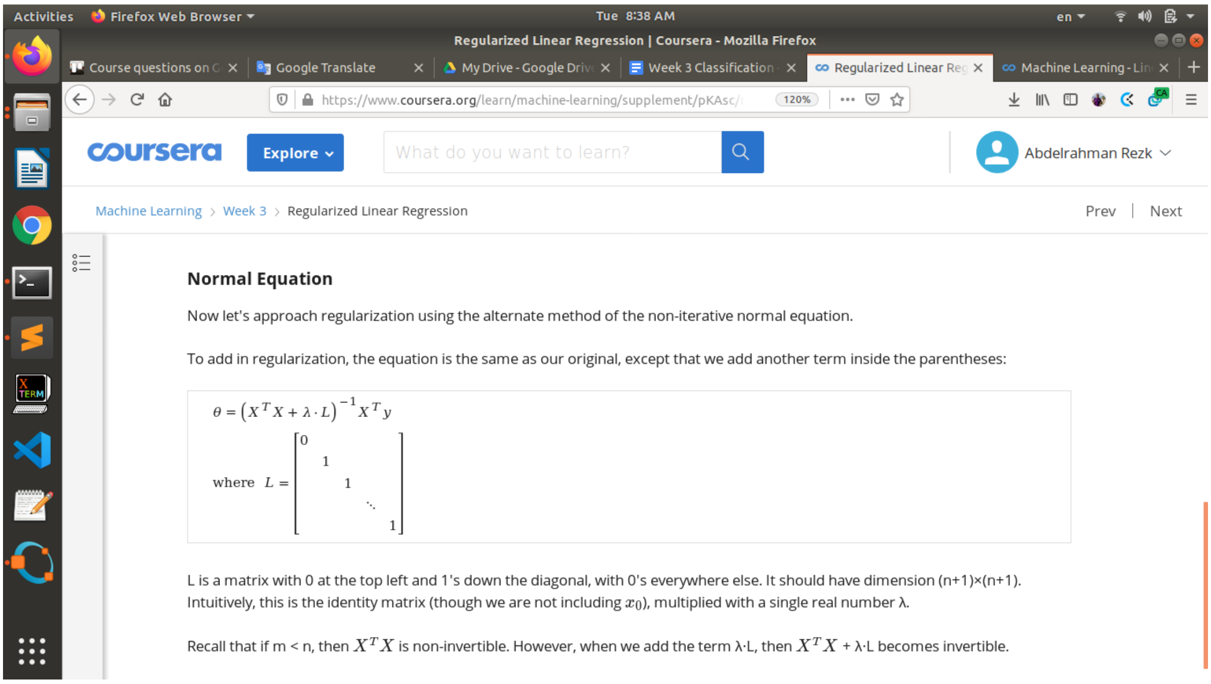
We will modify our gradient descent function to separate out θ_0 from the rest of the parameters because we do not want to penalize θ_0.

$$\begin{aligned}
&\text{Repeat } \{\\
&\quad \theta_0 := \theta_0 - \alpha\ \frac{1}{m}\ \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}\\
&\quad \theta_j := \theta_j - \alpha\ \left[\left(\frac{1}{m}\ \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j\right] \qquad j \in \{1,2...n\}\\
&\}
\end{aligned}$$

The term $\frac{\lambda}{m}\theta_j$ performs our regularization. With some manipulation our update rule can also be represented as:

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^m(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

The first term in the above equation, $1 - \alpha\frac{\lambda}{m}$ will always be less than 1. Intuitively you can see it as reducing the value of θ_j by some amount on every update. Notice that the second term is now exactly the same as it was before.



Regularized Logistic Regression

We can regularize logistic regression in a similar way that we regularize linear regression. As a
result, we can avoid overfitting.