

Abdelrahman Rezk

Web Developer

NLP & ML student

AOU University

# بسم الله الرحمن الرحيم

الملخص ده بإذن الله لازم تكون شوفت قبله تالت إسبوع من Andrew Ng

<https://www.coursera.org/learn/machine-learning>

وفيه مقابل ليه بالعربى مهندس هشام وده بصراحة ربنا يباركله ويجازيه خير على العلم الى بيقدمه والمساعدته الجميله وده برده تالت اسبوع ممثل فى الجزء الرابع فى الكورس فى البلاى ليست بتاعته.

[https://www.youtube.com/watch?v=e1lAgiFsobc&list=PL6-3IRz2XF5Uq7Pkl\\_PWOm\\_DLC2CFvSzU](https://www.youtube.com/watch?v=e1lAgiFsobc&list=PL6-3IRz2XF5Uq7Pkl_PWOm_DLC2CFvSzU)

المذكور هنا هايكون جهد شخصى ومحاولة إنى اساعد فى تبسيط المصطلحات والمعادلات مع حل امثله عملية تربط جميع الاجزاء المختلفه خلال الإسبوع التالت بإذن الله فالمفروض إنك تكون شوفته.

"وهاكون شاكر لو كان هناك دعوة بظهر الغيب"

# Classification

ببساطة التصنيف هو إني بدل ما كنت بحاول اتوقع فى ال linear regression continuous value لا هنا انا بحاول اتوقع discrete value يعنى value محدده ممكن يكون في اكثر من value بحاول اتوقعها زى الارقام من 0 ل 9 فى الارقام ولكن فى الغالب بيكون معظم التصنيف binary بمعنى توقع الميل المرسل spam or not.

عشان ققدر أحقق حاجه زى كده واني اخلى ال output بتاعى يكون قيم محدده بستخدم ال sigmoid function وهى بتعمل mapping للأرقام بين  $0 \leq h\theta(x) \leq 1$  بمعنى انى فى الآخر ال value الى هاتطلع هاتكون فى الرنج بين 0 و 1 وانا من هنا ببندى احدد بقا لو كان مثلا اكبر من او يساوى 5. ققول انه spam اقل من كده ققول انه not-spam.

input x	output y
0	4
1	7
2	7
3	8

$m$  = training examples

عدد الامثله الى هاقدر استخدمها عشان اعمل training للموديل بتاعى فى المثال الى فوق 4

$n$  = number of features

الحاجات المميزه الى بتساعدنى اعمل predict كويس وهنا معنديش الا 1 بس الى هو  $x$

$y$  = actual value or output value

انا فى الآخر بتعامل مع supervised learning فبحاول انى اعمل mapping from input to output معين وده الى انا بخلى الموديل بتاعى يدرب عليه ويحاول يقلل الفرق قدر الامكان بين القيم المتوقعة والقيم الحقيقية.

$y^{\wedge} = h(x)$  = predicted value لكن هنا على عكس ال linear regression انا هنا هاستخدم معادله ال sigmoid.

ديه القيمه المتوقعة

$\Sigma$  summation sign

$\Theta$  thetas parameters

قيم ال thetas الاخير هى الى بتستخدم عشان احاول اعمل predict لحاجه جديدة مشفتهاش قبل كده عن طريق قيم ال thetas الى انا طلعتها من ال training examples.

$\alpha$  alpha called learning rate

ديه الى بتاثر فى تغير تقليل نسبه الخطأ بصورة كبيره او صغيره على حسب قيمتها.

المعادلات المستخدمة

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

$$h_{\theta}(x) = g(\theta^T X)$$

$$z = \theta^T X$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

إننا بنحاول إننا نشتغل بنفس معادله ال linear عد إننا بنيجى بقا مع ال  $h(x)$  ونستخدم ال sigmoid function والى هى بتخلينى اعمل mapping لل real number فى  $\text{range}(0,1)$  بس هنا ال  $h(x)$  هاتدينا ارقام بين ال  $(0,1)$  كانها probability وإننا المفروض نعمل mapping بقا لو كان مثلا  $h(x) = 0.7$  فده اقرب لل 1 وهكذا.

Cost function of logistic regression

$$J(\theta) = -\frac{1}{m} * \sum_{i=1}^m [y^{(i)} * \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) * \log(1 - h_{\theta}(x^{(i)}))]$$

Vectorized version of cost function of logistic regression

$$h = g(X\theta)$$

$$J(\theta) = -\frac{1}{m} * [(y^T * \log(h) + (1 - y)^T * \log(1 - h))]$$

Gradient Descent of logistic regression

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

نفس ال linear regression لكن ال  $h(x)$  هنا بتاعت ال sigmoid function

Vectorized version of Gradient Descent of logistic regression

$$\theta := \theta - \frac{\alpha}{m} * X^T * (g(X\theta) - \vec{y})$$

Another vectorize version

$$grad = \frac{1}{m} * X^T * (g(X\theta) - \vec{y})$$

## Binary Classification examples with 2 features

### Dataset

$X_1$	$X_2$	$Y_0$
34	78	0
30	43	0
35	72	0
60	86	1
79	75	1

افتراض انى  $\theta_0 = -25$ ,  $\theta_1 = .20$ ,  $\theta_2 = .20$

والمفروض  $X$  بدل ما تبقا  $5 \times 2$  هاتبقا  $5 \times 3$  وهافتراض  $X_0 = 1$  فالبتالى  $x$  هاتكون

$X_0$	$X_1$	$X_2$
1	34	78
1	30	43
1	35	72
1	60	86
1	79	75

اولا cost function of vectorized version

$$h = g(X\theta)$$

$$J(\theta) = -\frac{1}{m} * [(y^T * \log(h) + (1 - y)^T * \log(1 - h))]$$

M=5 training examples

الاول نجيب ال g الى ال sigmoid function والى كانت بتساوى

$$g(z) \text{ or } h = \frac{1}{1 + e^{-z}}$$

$$z = X\theta$$

بما ان  $X = 5 \times 3$  و  $\theta$  بتساوى  $3 \times 1$  فينفع يحصل بينهم multiplication والناتج هايكون  $5 \times 1$  vector.

$$\begin{array}{rrrr} 1 & 34 & 78 & -2.6 \\ 1 & 30 & 43 & -25 \\ z = 1 & 35 & 72 * .20 & = -3.6 \\ 1 & 60 & 86 & .20 \\ 1 & 79 & 75 & 5.8 \end{array}$$

$$\begin{array}{r} 0.069138 \\ 0.000030432 \\ h = 0.026597 \\ 0.98523 \\ 0.99698 \end{array}$$

$$J(\theta) = -\frac{1}{5} * \begin{array}{rrrr} 0.069138 & 0 & 0.069138 \\ 0.000030432 & 0 & 0.000030432 \\ 0.026597 & 1 & 0.026597 \\ 0.98523 & 1 & 0.98523 \\ 0.99698 & 1 & 0.99698 \end{array} * \log( ) + (1 - 0)^T * \log(1 - )]$$

ضرب ال  $y = 1 \times 5$  vector فى ال  $h = 5 \times 1$  vector الناتج هايكون  $5 \times 1$ ,  $1 \times 5$  هايكون  $1 \times 1$  الى هو -0.0179046.

$$J(\theta) = -\frac{1}{5} * -0.0179046 + -0.098570968$$

هنا لما كانت ال  $y = 0$  هاتلاحظ فى اول جزء من المعادله محسبناش الى اخر رقمين الى كانت فيهما ال  $y = 1$  والباقي كله طار عشان ضربنا فى صفر والجزء الثانى العكس لما كانت ال  $y = 0$  بقا  $1 - 0$  بتساوى 1 والواحد الى طارت.

$$J(\theta) = 0.023295$$

ثانيا Gradient Descent of vectorized version

$$grad = \frac{1}{m} * X^T * (g(X\theta) - \bar{y})$$

ال  $sigmoid\ function\ g(X\theta)$  الى إحنا كنا جايينا فوق الى هي  $h$

بما إن لما عملنا  $transpose$  لل  $X$  هاتبقا  $3*5$  لما تضرب في  $5*1$  الى هو ناتج طرح ال  $(g(X\theta) - \bar{y})$  فهيكون عندي  $3 * 1$  vector يضرب في  $1/5$  ويكون ده قيم ال  $theta$  الجديدة الى هي هاتسعدني بعد كده اعمل  $predict$  عن طريق ال  $sigmoid\ function$  لما اخلص  $training$  بعد طبعا ما اكون حاولت اني قفل ال  $cost\ function$  قدر الإمكان.

$$= \frac{1}{5} * \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 34 & 30 & 35 & 60 & 79 \\ 78 & 43 & 72 & 86 & 75 \end{pmatrix} * \begin{pmatrix} 0.069138 & 0 \\ 0.000030432 & 0 \\ 0.026597 & -0 \\ 0.98523 & 1 \\ 0.99698 & 1 \end{pmatrix} = \begin{pmatrix} 0.013245 \\ 0.245894 \\ 0.986217 \end{pmatrix}$$

يتم التكرار حتى تقليل قيمه ال error لل global minimum.

**بإذن الله الجاي هايكون التطبيق على Neuron Network واتمنى من الله اكون قدرت اساعد بحاجه بسيطة وإن شاء الله ققدر اكمل باقى الكورس على خير.**