

Abdelrahman Rezk
AOU University
NLP & ML student
&
Web Developer

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 والمساعدده الجميله وده برده تالت اسبوع ممثل فى الجزء الرابع فى الكورس فى البلاى ليست بتاعته.

https://www.youtube.com/watch?v=e1lAqjFsobc&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 في ال linear regression continuous value لا هنا انا بحاول اتوقع discrete value يعني value محدده ممكن يكون في اكثر من value بحاول اتوقعها زي الارقام من 0 ل 9 في الارقام ولكن في الغالب بيكون معظم التصنيف binary بمعنى توقع الميل المرسل spam or not .

عشان ققدر أحقق حاجة زي كده واني اخلى ال output بتاعى يكون قيم محدده بستخدم ال sigmoid function وهي بتعمل mapping للأرقام بين $h\theta(x) \leq 1 \geq 0$ بمعنى اني في الاخر ال value الى هاتطلع هاتكون في الرنج بين 0 و 1 وانا من هنا ببتدى احدد بقا لو كان مثلا اكبر من او يساوي 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 في المثال الى فوق 4

n = number of features

الحاجات المميزه الى بتساعدني اعمل predict كوبس وهنا معنديش الا 1 بس الى هو x

y = actual value or output value

انا في الاخر بتعامل مع supervised learning فبحاول اني 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.

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

ديه القيمه المتوقعة

summation sign Σ

Θ thetas parameters

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

α 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

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

إحنا 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 $(0,1)$ كانه probability وإحنا المفروض نعمل mapping بقا لو كان مثلا $h(x) = 0.7$ فده اقرب لل 1 وهكذا.

Cost function of logistic regression

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

Vectorized version of cost function of logistic regression

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

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

Gradient Descent of logistic regression

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

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

Vectorized version of Gradient Descent of logistic regression

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

Another vectorize version

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - g(\theta^T X_i))^2$$

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

افترض اني $\theta_0 = -25$, $\theta_1 = .20$, $\theta_2 = .20$

والمفروض X بدل ما تبقا 2×5 هاتبقا 3×5 وهافترض $X_0 = 1$ فالتالى 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

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - g(\theta^T X_i))^2$$

$$\theta = \theta - \frac{1}{m} \sum_{i=1}^m (y_i - g(\theta^T X_i)) X_i$$

M=5 training examples

الاول نجيب ال g الى هي ال sigmoid function والى كانت بتساوى

$$h = \frac{1}{1 + e^{-z}}$$

$$z = X\theta$$

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

$$\begin{bmatrix} 5.2 \\ 0.1 \\ 5.3 \\ .4 \\ .5 \end{bmatrix} = \begin{bmatrix} 43 & 87 \\ 03 & 34 & -52 \\ 53 & 27 & 02. \\ 06 & 68 & 02. \\ 97 & 57 \end{bmatrix} \theta = z$$

$$\begin{bmatrix} .0 \\ .0 \\ .0 \\ .0 \end{bmatrix} = h$$

$$= \begin{bmatrix} 0.069138 \\ 0.000030432 \\ 0.026597 \\ 0.98523 \\ 0.99698 \end{bmatrix} + \begin{bmatrix} 0^T \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \log(1 - \dots) = \begin{bmatrix} 0.069138 \\ 0.000030432 \\ 0.026597 \\ 0.98523 \\ 0.99698 \end{bmatrix} \theta = -\frac{1}{5}$$

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

$$-0.098570968 + 0.0179046 \theta = -\frac{1}{5}$$

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 0-1 بتساوى 1 والواحد هي الى طارت.

$$J(\theta) = 0.023295$$

ثانيا Gradient Descent of vectorized version

$$X^T * (y - \hat{y}) = \frac{1}{m}$$

ال **sigmoid function** (θX الى إحنا كنا جايينا فوق الى هي **h**

بما إن لما عملنا **transpose** لل X هاتبقا 5×3 لما تضرب في 1×5 الى هو ناتج طرح ال $\hat{y} - y$ فهيكون عندي **vector 3×1** 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 ققل ال **cost function** قدر الإمكان.

$$J = \frac{1}{2} \sum_{i=1}^m (y_i - \hat{y}_i)^2 = \frac{1}{2} (43 - 03)^2 + \frac{1}{2} (53 - 06)^2 + \frac{1}{2} (97 - 87)^2 + \frac{1}{2} (34 - 27)^2 + \frac{1}{2} (68 - 57)^2$$

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 فقدر اكمل باقى الكورس على خير.