

Post 10

معظم الشايتز كان الكلام بشكل عام ومنتجة اكثر ناحية ال Linear Regression وهو اني اتوقع قيمة بتسمى continuous value يعنى فى الآخر مقدرش احصر القيمة ديه فى range معين ، لكن الجزء الآخر من ال Regression هو اني احاول اتوقع قيمة من مجموعة من القيم ، زى اني مثلا اتوقع هل الى فى الصور ده قطة ولا لا ، هنا ان يهمنى فقط اني الصورة قطة ولا لا (معنى كده اى صورة كانت اى حاجة تانية هتمثل بالنسبة ليا class واحد) وال class التانى هو القطة ، وده الى احنا بنسمية Binary Classification ، لكن ازاى انا هقدر اعمل map من القيم الى بيخرجها الموديل ل قيم فى الآخر عبارة عن 0 او 1 او قيمة بينهم وبناء على threshold معين زى اني لو طلع النسبة فوق 50% مثلا هقول اني ديه صورة قطة ولو اقل بيقا لا ، فهنا زى ال prediction الى كان بيحصل فى ال linear regression بظبط بس هخد ال output منه واديه ل function اخرى اسمها Sigmoid function وظيفتها انها تعمل map لل output ده يكون من 0 ل 1 ، بعد ما عملت ال prediction الى انا عايزه عن طريق ال Sigmoid function هبتدى احتاج حاجة تانية هو اني اعمل training يكون الهدف منه هو توقع كبير لل positive class لما تكون الصورة قطة يعنى اكبر من 50% ويكون توقع قليل لما يكون العكس ، وده بيحصل من خلال اني بعمل tweak لل paramters عشان تقدر تساعدنى فى التوقع ده .

Post 11

انا حببت يكون الكلام عن ال cost function الخاصة بال Logistic Regression يكون فى بوست لوحده عشان نوضح فيه ايه الى بيحصل لو كان ال target ب 0 او 1 بناء على ال function الى معانا ، خلينا نطبقها لما يكون ال target بتاعنا الى هو $y=1$ ونشوف هنوصل لايه هنلاقى اني ال term الاخير بقا كله ب 0 لان $(y-1)$ هتساوى 0 بما انا $y=1$ وبكده ده يوضح اني سواء كانت ال $y=1$ or 0 نص المعادلة بيطير ، طيب جميل دلوقت ال $y=1$ المفروض اني ده كده positive class ومعنى كده اني المفروض يكون ال output الخاص بال instance ده اكبر من ال threshold عشان تقدر تقول ده positiv ، فانا كده محتاج اشوف ال $\log(p_i)$ الى هو ال prediction بتاعى مضروب فى ال negative الى بره وده مهم جدا لان ال mapping من 0 ل 1 وبعد كده تاخد ال log هيدى negative value لذلك انا باخد negative كمان طيب معنى كده كل ما كان ال prediction بتاعى بيقترب من 1 ال error هيكون اقل ما يمكن وكل اما بيقرب ناحية ال 0 هيكون اكبر ما يمكن وده الى احنا عايزينه لان ال target فى ال instance ده هو 1 ، زى ما هو واضح فى المثال الى فى الصورة ، والعكس هيجصل لما تكون ال $y=0$. بعد كده طبعا هنبتدى اننا نجيب التفاضل بتاع ال cost function عشان نعمل updates لل Weights .

```
In [38]: test = np.linspace(.01,1,10)
          print(test)
          test = [-np.log(i) for i in test]
          test

Out[38]: [4.605170185988091,
          2.120263536200091,
          1.4696759700589417,
          1.0788096613719298,
          0.7985076962177716,
          0.579818495252942,
          0.40047756659712525,
          0.2484613592984996,
          0.11653381625595151,
          -0.0]
```

Equation 4-17. Logistic Regression cost function (log loss)

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

Post 12

معظم ال classification problem يكون multinomial يعنى بيكون مش فقط 2-class لا بيبقا اكثر من كده ممكن مثلا اتوقع رقم من 0 ل 9 فهنا عندى 10-classes وهكذا وده يختلف عن ال multi output الى فيه بتوقع اكثر من object مثلا زى الفيس بوك لما بيعمل tag لصحابك الى معاك فى الصورة ، ال multiclassification التعامل معاه ممكن يكون برضه من خلال ال Binary model ولكن هحتاج models كثير جدا عشان افرق مثلا 0 عن 1 و 0 عن 2 لحد 9 وبعد كده ال 1 عن ال 2 وبرضه نفس الكلام فهلاقى عندنا عدد كبير جدا من ال models وفى نفس الوقت كل ما ال Number of classes بيزيد ال models هتزيد وكل ما يكون عندى instance جديد لازم امرة على كل ال models ديه ، لكن ده مش practical ولذلك بنتعامل مع ال multi classification من خلال function اسمها Softmax وفيها انا الأول بشوف ال score بتاع كل class بالنسبة ل instance x اد ايه بعدين بعمل normalization عشان اوزع ال probability وتكون بين ال 0 و 1 من خلال ال softmax ، يعنى مثلا لو عندى 150 مثال فى الداتا وكل واحد معاه 2 features هنا انا هحتاج weights بس هتكون خاصة بكل class على حد يعنى بدل ما فى العادى كان عندى 2 * 1 عشان عندى 2 features لا هيبقا عندى بقا 2 * matrix 10 ليه عشان 2-classes * 10-features ومن خلال ال prediction function العادية بتاعت ال linear regression هقدر اجيب ال score الخاصة بكل class بالنسبة لكل instance فى الداتا بمعنى

$$X = 150 * 2$$

$$\text{Weights} = 2 * 10$$

Prediction = $X * \text{weights} = (150 * 2) * (2 * 10) = 150 * 10$ which means 10-score for each instance.

بعد كده بقا ببتدى ادى الكلام ده لل softmax بالمعادلة الى فى الصورة عشان يكون فى الاخر مجموع ال scores بالنسبة لكل instance يكون 1 عن طريق انى بقسم score of each class بمجموع ال scores بتاعت ال instance ، ونفس الكلام هروح اشوف ال cross entropy function وبعد كده ال partial derivative بتاعها عشان اوصل فى الاخر انى اعمل updates لل weights.

Equation 4-19. Softmax score for class k

$$s_k(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\theta}^{(k)}$$

Equation 4-20. Softmax function

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp(s_k(\mathbf{x}))}{\sum_{j=1}^K \exp(s_j(\mathbf{x}))}$$

- K is the number of classes.
- $\mathbf{s}(\mathbf{x})$ is a vector containing the scores of each class for the instance \mathbf{x} .
- $\sigma(\mathbf{s}(\mathbf{x}))_k$ is the estimated probability that the instance \mathbf{x} belongs to class k given the scores of each class for that instance.