How to calculate number of features based on image resolution?

Just covered Andrew Ng's Non-linear Hypothesis of Neural Netowrks, and we had a multiple choice question for determining **number of features** for an image of resolution **100x100** of **grescale** intensities.

And the answer was 50 million, 5 x 10^7

However, earlier for a 50 x 50 pixel, grey scale image. the number of features is 50x50 (2500)

Why would it be 5 x 10^7 instead of $10,000$?

He does however say including all quadratic terms ($x_i x_j$) as features

> Suppose you are learning to recognize cars from 100×100 pixel images (grayscale, not RGB). Let the features be pixel intensity values. If you train logistic regression including all the quadratic terms ($x_i x_j$) as features, about how many features will you have?

and in the earlier slide regarding the 100x100, that the quadratic features (x_i x x_j) = 3 million features, but I still can't put a finger on the connection.

feature-selection image-processing

edited Dec 29 '13 at 0:46

asked Dec 29 '13 at 0:13
lancovici
255 1 3 12

2 Answers

If you are using all the linear and quadratic features, the total number is supposed to be:

```
100*100 + 100*100 + C(100*100,2) = 50015000
10000   + 10000   + 49995000     = 50015000
xi        xi^2      xixj
```

edited Dec 30 '13 at 1:40
lancovici
255 1 3 12

answered Dec 29 '13 at 2:56
lennon310
2,142 1 10 26

1 Can you explain it a little bit further? are you saying xi + xi^2 + xixi? Is xi = 100, and xj = 100? why is xi and xi^2 both are 100*100? What is C(100*100,2)? – lancovici Dec 29 '13 at 18:52

2 (1)there are totally 100*100 pixels, if you are using intensity as features, there will be 100*100 features in total, that's xi; and (ii) you may also use the power density as a feature, that's (xi,xi) or xi.^2, still 100*100 in total; finally(iii) if you use the correlations between two pixels, there will be C pairs of pixels in total, that's (xi,xj), C is combination in math (mathworld.wolfram.com/Combination.html) – lennon310 Dec 29 '13 at 19:20

Thanks, one last question is why does xi = xi^2 in this context? – lancovici Dec 30 '13 at 1:39

I used xi to represent single pixel, and xi^2 means use pairs of the same pixel (xi,xi). The number of single pixel is the same of pairs of the same pixel. It has nothing to do with the pixel intensity. Sorry for the confusion. – lennon310 Dec 30 '13 at 1:49

Perhaps a simpler case will make things clearer. Lets say we choose a 1x2 sample of pixels instead of 100x100.

Sample Pixels From the Image

```
+----+----+
| x1 | x2 |
+----+----+
```

Imagine when plotting our training set, we noticed that it can't be separated easily with a linear model, so we choose to add polynomial terms to better fit the data.

Let's say, we decide to construct our polynomials by including all of the pixel intensities, and all possible multiples that can be formed from them.

Since our matrix is small, let's enumerate them:

$$x_1,\ x_2,\ x_1^2,\ x_2^2,\ x_1 \times x_2,\ x_2 \times x_1$$

Interpreting the above sequence of features can see that there is a pattern. The first two terms, group 1, are features consisting only of their pixel intensity. The following two terms after that, group 2, are features consisting of the square of their intensity. The last two terms, group 3, are the product of all the combinations of pairwise (two) pixel intensities.

group 1: $x_1,\ x_2$

group 2: $x_1^2,\ x_2^2$

group 3: $x_1 \times x_2,\ x_2 \times x_1$

But wait, there is a problem. If you look at the group 3 terms in the sequence ($x_1 \times x_2$ and $x_2 \times x_1$) you'll notice that they are equal. Remember our housing example. Imagine having two features x1 = square footage, and x2 = square footage, for the same house... That doesn't make any sense! Ok, so we need to get rid of the duplicate feature, lets say arbitrarily $x_2 \times x_1$. Now we can rewrite the list of group three features as:

group 3: $x_1 \times x_2$

We count the features in all three groups and get 5.

But this is a toy example. Lets derive a generic formula for calculating the number of features. Let's use our original groups of features as a starting point.

$sizegroup1 + sizegroup2 + sizegroup3 = m \times n + m \times n + m \times n = 3 \times m \times n$

Ah! But we had to get rid of the duplicate product in group 3.

So to properly count the features for group 3 we will need a way to count all unique pairwise products in the matrix. Which can be done with the binomial coefficient, which is a method for counting all possible unique subgroups of size k from an equal or larger group of size n. So to properly count the features in group 3 calculate $C(m \times n, 2)$.

So our generic formula would be:

$$m \times n + m \times n + C(m \times n, 2) = 2m \times n + C(m \times n, 2)$$

Lets use it to calculate the number of features in our toy example:

$$2 \times 1 \times 2 + C(1 \times 2, 2) = 4 + 1 = 5$$

Thats it!

edited Jul 1 '15 at 0:30

answered Jul 1 '15 at 0:04

1 Beautiful answer! Thanks – Kevin Zakka Jan 5 '16 at 13:29

Wish this explanation had been given in the lecture! – Ian Walker-Sperber Jan 16 at 14:00