

Review of machine learning.

Purpose of machine learning:

First, let the machine learn a lot of data ( data1, data2, data3, ..., dataN ). then tell the machine the type of the all of the data: ( type1, type2, type3, ..., typeN ).

Second, give a new data: data N+1, machine will automatically tell the type of the data N+1.

For example: give you N banana pictures. each picture has the same width and height. Each picture is actually a long vector of values from 0 to 255. the size of the vector = width \* height \* 3 . 3 is rgb for each pixel. So each banana image becomes something like: vector = { 23, 123, 245, 4, 234, 234, 42, 40, 134, 4, 45, 92, ..., 56, 54, 23, 15, 30, 120, ... } this is also called "**features**" of the image.

suppose you have vector1, vector2, ..., vectorN-1, vectorN. are N images. and you know the type of them are type1 = apple, type2 = banana, ... typeN-1 = apple, typeN = pear. Then you map all of the type to a unique number. like apple = 1. banana = 2, pear = 3. then you have:

vector1 ===> 1  
vector2 ===> 2  
...  
vectorN-1 ===>1  
vectorN ===>3

You let computer know this as we discussed previously. then computer's task is to find one way can compute vectors(features) to integers(types). The procedure to find how to convert vector to type is called "**training**".

Now introduce a data type named "matrix" and how to use it. "matrix" basiclly is a 2D array. ". "training" is to find the matrix.

Give an example:

suppose you have very small pictures:

vec1 = { 12, 15, 30, 93, 203, 19 } { matrix\_row1 } = type1  
vec2 = { 41, 2, 201, 191, 84, 39 } { matrix\_row2 } = type2  
vec3 = { 7, 192, 30, 42, 94, 30 } { matrix \_row3 } = type3

The rule might be different of how to use matrix. but here, we suppose the rule of how to use matrix is:

vec1 \* row1 + vec1 \* row2 + vec1 \* row3 = type1;  
vec2 \* row1 + vec2 \* row2 + vec2 \* row3 = type2;  
vec3 \* row1 + vec3 \* row2 + vec3 \* row3 = type3;  
each row contrains 6 numbers. so I let vec1 \* row1 = vec1\_1 \* row1\_1 + vec1\_2 \* row1\_2 + ... + vec1\_6 \* row1\_6. which means. vec1 \* row1 is let each element of vec1 multiply row1. then add together.

It is like solving equations. You know vec1, vec2, vec3, and type1, type2, type3. Let computer solve row1, row2, row3. suppose the computer found matrix\_row1 and row2 and row3 after training. You don't need to know the detail of training at this moment. You just know matrix is some number array 2D. For example. computer found it.

Then if you have a vec4. a new image. And you use the matrix row you just computed by training. You use the rule to compute: vec4 \* row1 + vec4 \* row2 + vec4 \* row3 = some value4. This value4 虽然不是一定是type1， type2 或者type3. 但是可能是离他们比较近的一个。 那么就把这个新image， 就可以归到某个type里面了。这个过程就是"**inference**" . Inference means using the matrix computed by training. to compute the type. 这就是machine learning。通过上一步traning算出来的matrix.根据规则算type。不一定准， 但是肯定比胡乱猜要好很多。 上面rule也很关键。 有的好几步rule。就是vec1， vec2用matrix1 算出来一个中间层， 再用中间层和matrix2再算下一层， 再用matrix3 算下一层， 最后再算type。 中间折腾好几次， rule比较复杂 然后用training出来的matrix1, matrix2, matrix3..和上面的复杂rule作用在新的vector上 那么inference效果就更好。