Cross Validated

What is “unit” standard deviation?

Asked 1 year, 11 months ago Active 1 year, 10 months ago Viewed 2k times

3

2

It is a common practice to normalize data for training a neural network to zero mean and "unit" standard deviation to reduce exploding and vanishing gradient effects etc.

What does "unit" std mean here? An example would be really helpful.

neural-networks normalization data-preprocessing

edited Sep 30 '17 at 12:29

asked Sep 30 '17 at 12:07

Bob

135 7

5 It means equal to 1. To get that you divide values by the standard deviation of the original data. (The zero mean comes from subtracting the mean.) – Nick Cox Sep 30 '17 at 12:32

1 Answer

Supose you want to predict house prices from two features: number of bedrooms (integer unit) and size (in squared meters unit), like the fictitious data bellow:

```
import numpy as np

X = np.array([[1, 65],[3, 130],[2, 80],[2, 70],[1, 50]])
```

Notice that each feature have very different mean and standard deviation

```
print("mean={}, std{}".format(X.mean(axis=0), X.std(axis=0))
```

Outputs: `mean=[  1.83333333,  78.33333333]), std=[  0.68718427,  24.94438258])`

Noticed that the feature size has mean and std more than 30x bigger than number of bedroom, this produces distortions in some algorithms calculation (like neural nets, svm, knn, etc) where somes feature with larger values dominates completely the others with smaller values. To solve that an common and very effective practice is to transform the data to units of standard deviation with zero mean, that is, you subtract the mean and divides by the standard deviation, like bellow:

```
X_t = (X - X.mean(axis=0))/X.std(axis=0)
```

The variable `X_t` (X transformed) contains your features in unit standard deviations with zero mean, printing `X_t` you get:

```
array([[-1.21267813, -0.53452248],
       [ 1.69774938,  2.07127462],
       [ 0.24253563,  0.06681531],
       [ 0.24253563, -0.33407655],
       [-1.21267813, -1.13586028],
       [ 0.24253563, -0.13363062]])
```

Look how the numbers in both features have all the same magnitude. If you print `X_t` mean and std now you get

```
mean=[  1.11022302e-16   2.08166817e-16], std=[ 1.  1.]
```

as expected.

answered Oct 5 '17 at 17:47

xboard

505 5 13