SOLUTION TO MOST COMMON PROBLEMS IN MACHINE LEARNING

UNIT 1

JOSE DE LA CRUZ PAT RAMIREZ

2009104@upy.edu.mx

Robotics 9°B

15/09/2023

VICTOR ALEJANDRO ORTIZ SANTIAGO

Overfitting

It is quite common that when starting to learn machine learning we fall into the problem of Overfitting. What will happen is that our machine will only adjust to learning the cases that we teach it and will be unable to recognize new input data. In our input data set we often introduce atypical (or anomalous) samples or samples with “noise/distortion” in some of their dimensions, or samples that may not be completely representative. When we “over-train” our model and fall into overfitting, our algorithm will consider as valid only data identical to that of our training set – including its defects – and will be unable to distinguish good entries as reliable if they are a little off. of the pre-established ranges.

Underfitting

It is caused by the excessive generalization of the input data that is introduced into the model, which affects the precision of the data that is presented.

A module on which underfitting occurs is a model that cannot model the data set nor generalize to a new data set. Likewise, the model cannot create a mapping between data or input variables, as well as target variables.

Meanwhile, the situation of underfitting within the model can be easy to detect, so if a solution is not found to put an end to it, it is much better and more useful to try with other types of algorithms.

In general, and a little more everyday terms, underfitting happens when, as human beings, we cannot give a machine the same level of conceptualization that we can have outside the computing sector. We could give an example with the Keep Coding bootcamps, where, if we called them all bootcamps or intensive training, we would not be able to differentiate one from the other.

Define and distinguish the characteristics of outliers.

Outliers are data points that significantly deviate from the average or typical values within a dataset. These observations, though rare, can influence statistical analyses and machine learning models if not properly addressed.

Outliers are data points that lie far away from most of the data, either above or below the expected range. They can arise due to several reasons, such as measurement errors, experimental anomalies, or truly exceptional observations. Outliers can distort statistical analyses, affecting the accuracy and reliability of the results.

When are Outliers Dangerous?

Outliers can be particularly dangerous when they exert a disproportionate influence on the analysis or modeling results. They can skew the statistical measures of central tendency, such as the mean and median, leading to biased estimates. In regression analysis, outliers can significantly affect the slope and intercept of the regression line, distorting the relationship between variables. Outliers can also impact clustering algorithms by affecting the distance metrics and the formation of clusters.

How do you solve these problems?

overfitting

- Adding more data

Your model is overfitting when it fails to generalize to new data. That means the data it was trained on is not representative of the data it is meeting in production. So, retraining your algorithm on a bigger, richer, and more diverse data set should improve its performance. Unfortunately, getting more data can prove to be difficult; either because collecting it is expensive or because very few samples are regularly generated. In that case, it might be a good idea to use data augmentation.

- Data augmentation

This is a set of techniques used to artificially increase the size of a dataset by applying transformations to the existing data. For instance, in the case of images, you can flip images horizontally or vertically, crop them or rotate them. You can also turn them into grayscale or change the color saturation. As far as the algorithm is concerned, new data has been created. Of course, not all transformations are useful in every case. And in some cases, your algorithm will not be fooled in short, data augmentation can be a powerful tool, but it requires careful examination and understanding of your data.

- Regularization

Regularization refers to a large range of techniques and we won't list them all or go into details here. The main idea you need to remember is that these techniques introduce a "complexity penalty" to your model. If the model wants to avoid incurring that penalty, it needs to focus on the most prominent patterns which have a better chance of generalizing well. Regularization techniques are powerful and all the models you build will use them in some way.

- Removing features from data

Sometimes, your model may fail to generalize simply because the data was trained on was too complex and the model missed the patterns it should have detected. Removing some features and making your data simpler can help reduce overfitting.
It is important to understand that overfitting is a complex problem. You will almost systematically face it when you develop a deep learning model, and you should not get discouraged if you are struggling to address it. Even the most experienced ML engineers spend a lot of time trying to solve it.

underfitting.

- Increasing the model complexity

Your model may be underfitting simply because it is not complex enough to capture patterns in the data. Using a more complex model, for instance by switching from a linear to a non-linear model or by adding hidden layers to your neural network, will very often help solve underfitting.

- Reducing regularization

The algorithms you use include by default regularization parameters meant to prevent overfitting. Sometimes, they prevent the algorithm from learning. Reducing their values helps.

- Adding features to training data

In contrast to overfitting, your model may be underfitting because the training data is too simple. It may lack the features that will make the model detect the relevant patterns to make accurate predictions. Adding features and complexity to your data can help overcome underfitting.

Did you notice? That is right! Adding more data is not included in the techniques to solve underfitting. Indeed, if your data is lacking the decisive features to allow your model to detect patterns, you can multiply your training set size by 2, 5 or even 10, it won’t make your algorithm better!

Unfortunately, it has become a reflex in the industry. No matter what the problem their model is facing, a lot of engineers think that throwing more data at it will solve the problem. When you know how time-consuming and expensive it can be to collect data, this is a mistake that can seriously harm or even jeopardize a project.

Being able to diagnose and tackle underfitting/overfitting is an essential part of the process of developing a good model. Of course, there are lots of other techniques to solve these problems on top of the ones we just listed. But these are the most important ones and if you manage to master them, you will be well equipped to start your machine learning journey!

Outliers

- **Trimming:**

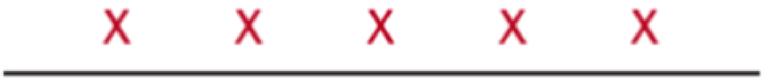
Trimming involves removing a certain percentage of extreme values from both ends of the data distribution. This approach discards the outliers entirely, which can reduce their influence on statistical analyses. By trimming the dataset, the extreme values are eliminated, and the analysis focuses on most of the data. However, it is essential to carefully consider the percentage to be trimmed to avoid removing too much data.

- **Capping:**

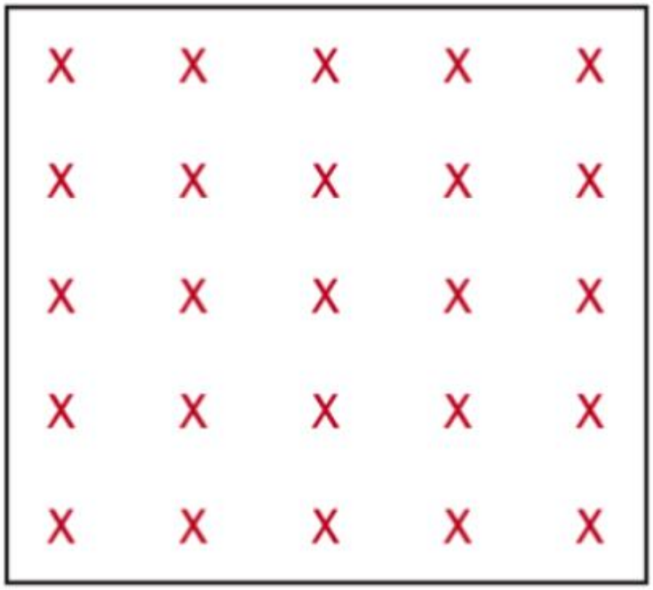
Capping, also known as Winsorization, sets a predefined threshold beyond which the outlier values are replaced with the nearest acceptable value within that threshold. Capping prevents the complete removal of outliers and instead modifies their values to align them with the nearby observations. This approach helps control the impact of outliers while retaining their presence in the dataset. Capping can be done symmetrically, by capping both lower and upper extremes, or asymmetrically if there is a specific directionality to the outliers.

The dimensionality problem.

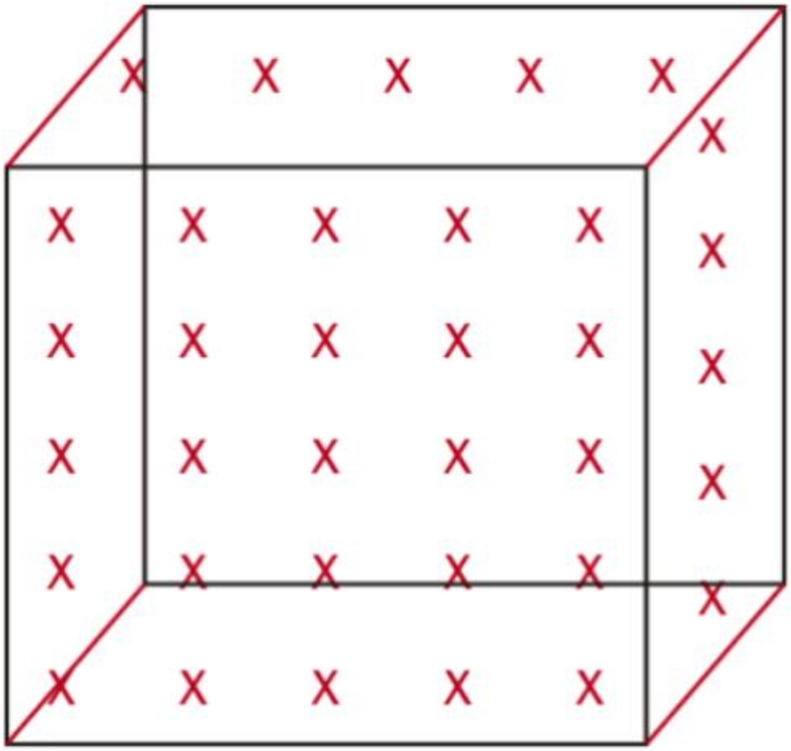
The dimensionality, also known as the Hughes Phenomenon — there are two things to consider. On the one hand, ML excels at analyzing data with many dimensions. Humans are not good at finding patterns that may be spread out across so many dimensions, especially if those dimensions are interrelated in counter-intuitive ways. On the other hand, as we add more dimensions we also increase the processing power we need to analyze the data, and we also increase the amount of training data required to make meaningful data models.



A one-dimensional features space with five data points



A two-dimensional features space with 25 data points



A three-dimensional features space with 125 data points

What problems arise with many dimensions?

When working with data, it is desirable to have a lot of attributes and information in the data set to give the model many possibilities to recognize structures in the data. However, it can also lead to serious problems, as the name Curse of Dimensionality already suggests.

Data Sparsity

The example shown illustrates a problem that occurs with many attributes. Due to many dimensions, the so-called data space, i.e., the number of values that a data set can assume, also grows. This can lead to so-called data sparsity. This means that the training data set used to train the model does not contain certain characteristics at all or only very rarely. As a result, the model delivers only poor results for these edge cases.

Assume that we examine 1,000 customers in our example, as it would be too time-consuming to survey even more customers, or this data is simply not available. It is possible that all age groups from young to old are well represented among these customers. However, if the additional dimension of income is added, it is unlikely that all possibilities of age and income combinations are simultaneously represented among the 1,000 customers.

Distance Concentration

In Machine Learning, when one wants to evaluate the similarity of different datasets, distance functions are often used for this purpose. The most common clustering algorithms, such as k-means clustering, rely on calculating the distance between points and assigning them to a cluster or not depending on the size. In multidimensional spaces, however, it can quickly come to the point that all points are similarly far apart, so that separation seems almost impossible.

We know this phenomenon to some extent from everyday life as well. If you take a photo of two objects, such as two trees, then they can look very close to each other in the picture, since it is only a two-dimensional image. In real life, however, it can happen that the trees are several meters apart, which only becomes clear in three dimensions.

These problems, which can arise in connection with many dimensions, are summarized under the term Curse of Dimensionality.

There are two different approaches to this:

Feature Selection

The first solution to combat the Curse of Dimensionality may be very single-minded: we select only those features that contribute particularly well to solving the problem. This means we limit ourselves only to dimensions that have some importance to the problem. There are several ways to do this as well, among the most common are these:

Correlation analysis: To select suitable attributes, all correlations between two attributes are calculated. If two attributes have a high correlation, it may be possible to remove one of the attributes without losing much information content.

Multicollinearity: This procedure is like the correlation analysis with the difference that not only pairwise attributes are examined, but also whether individual attributes can be represented as a sum, as a linear regression of other attributes. If this is possible, all attributes that are part of the regression line can be omitted.

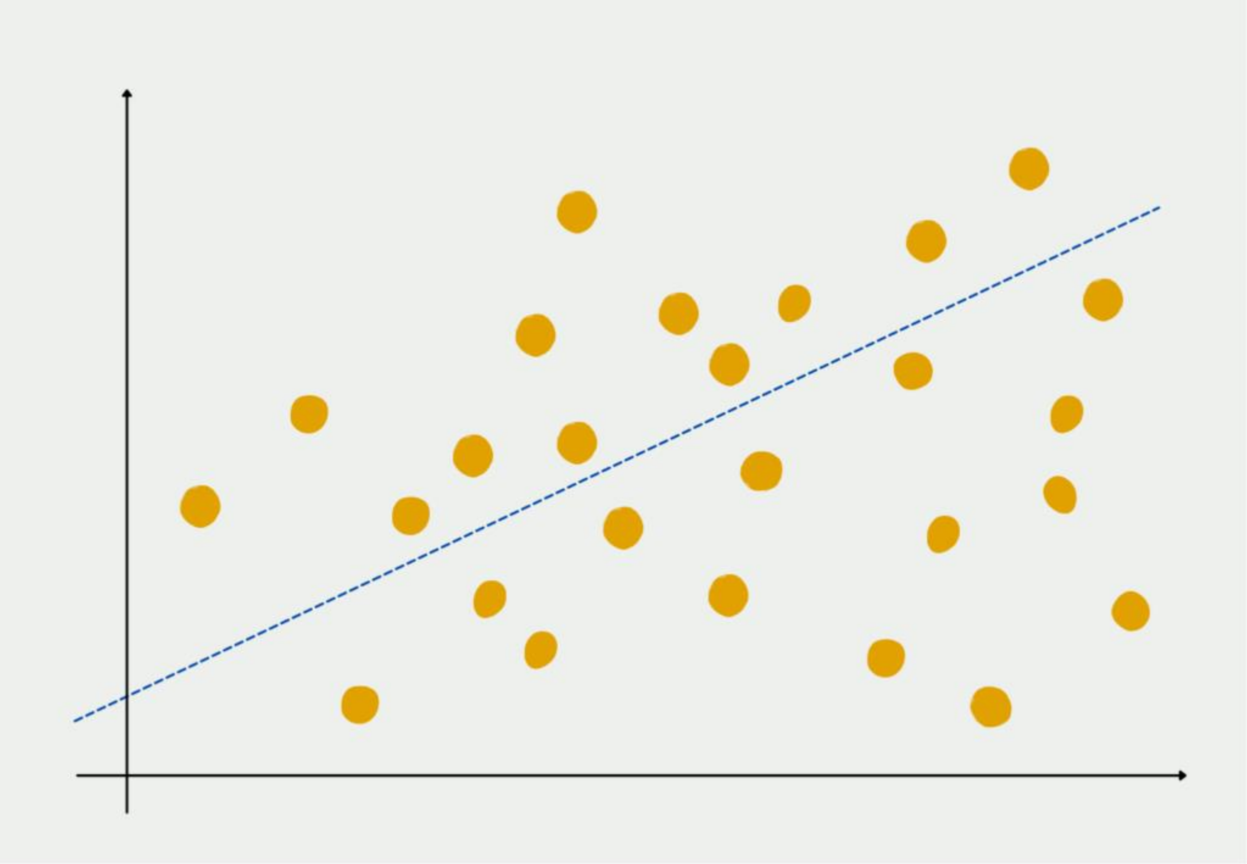
The advantage of feature selection is that the attributes are preserved in their original form, which also makes them easier to interpret.

Feature Extraction

The second approach to combat the curse of dimensionality takes a different approach and attempts to form new, summarized dimensions from the many dimensions so that the number of dimensions is reduced while still retaining as much information content as possible.

A frequently used algorithm for this is the so-called Principal Component Analysis. The core idea of Principal Component Analysis is that several variables in a data set may measure the same thing, i.e. they are correlated. Thus, the different dimensions can be combined into fewer so-called principal components without compromising the validity of the data set. Body size, for example, has a high correlation with shoe size, since in many cases tall people also have a larger shoe size and vice versa. So, if we remove shoe size as a variable from our data set, the information content does not really decrease.

In statistics, the information content of a data set is determined by the variance. This indicates how far the data points are from the center. The smaller the variance, the closer the data points are to their mean value and vice versa. A small variance thus indicates that the mean value is already a good estimate for the data set.



In the first step, PCA tries to find the variable that maximizes the explained variance of the data set. Then, step by step, more variables are added to explain the remaining part of the variance, because the variance, i.e., the deviation from the mean, contains the most information. This should be preserved if we want to train a model based on it.

This approach can in many cases lead to better results than feature selection, but it creates new, artificial attributes that are not so easy to interpret. The first main component, for example, is a combination of various attributes. In individual cases, these can then be combined to form a larger supergroup. For example, attributes such as height, weight, and shoe size can be combined and interpreted as the group “physical characteristics”.

Bias-variance trade-off

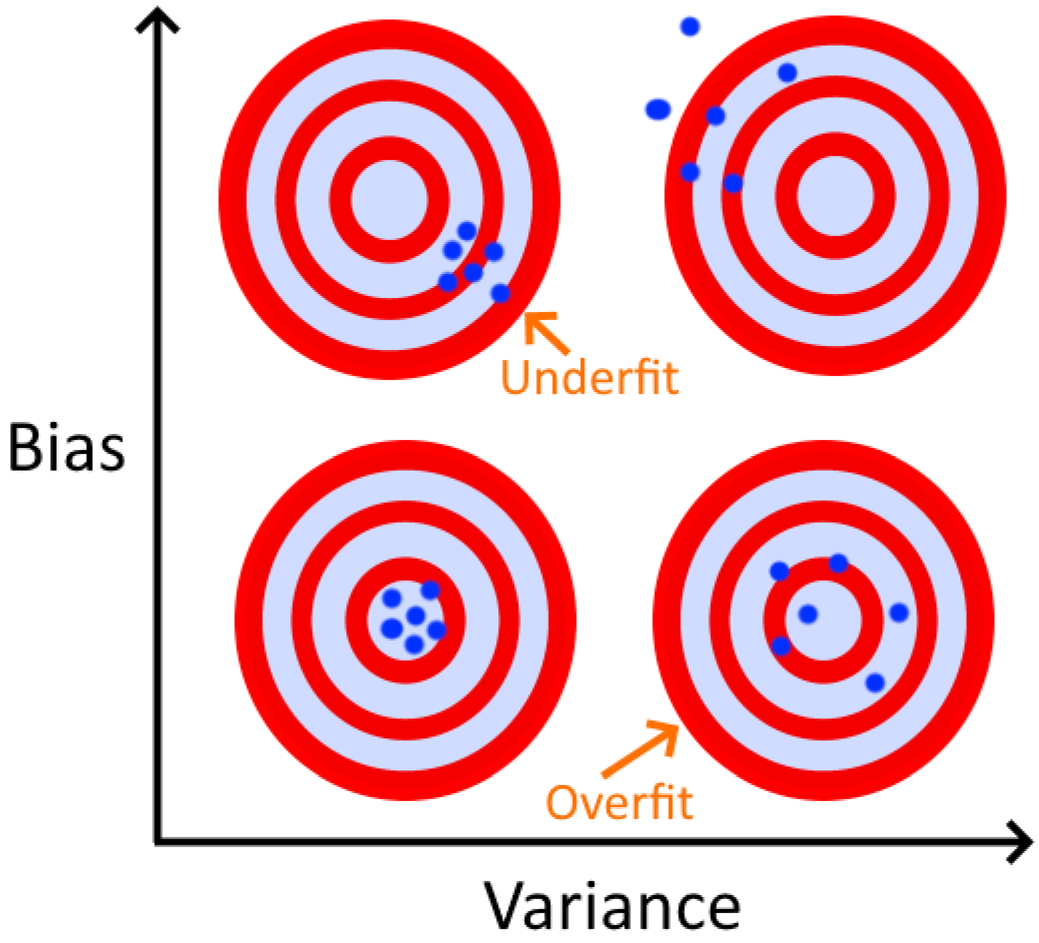
When we create a machine learning model, we want it to be as efficient as possible to obtain the best results, but to achieve this we must achieve a balance between Bias and Variance. This brings us to Bias-Variance compensation.

Both Bias and Variance are errors in our model, and we seek to reduce them, but it is not that simple. When we manage to reduce Bias, we will notice that Variance increases, if we reduce variance then Bias increases. This leads us to a balanced situation where both must be as small as possible.

Bias or bias in a machine learning model is very easy to understand, it is simply a type of error that indicates the difference that exists between the model's prediction and the current value. If we look at it from the perspective of statistics, it is the tendency to overestimate or underestimate a parameter. If the model has a high Bias, it means that it pays little attention to the data and oversimplifies the model. This leads us to have a high error in both training and testing.

Variance or variance is also considered an error in the machine learning model we have. We can understand this error as how sensitive our model is to the data. If we have excess sensitivity, the model may believe it sees patterns that do not really exist there. In the context of statistics, it is a measure of dispersion of the data, it is the distance of each variable from the mean of all the variables. If we have a high Variance value this means that the model pays a lot of attention to the training data and will not generalize well to data, it has not seen. We can see this when the model works very well in training but has a lot of errors in the test.

It is not possible to select one over another to reduce, since if we concentrate on one the model can generate overfit or underfit. In both cases our model is going to have problems. The only option we have is to find the point where both can be as small as possible.



By observing the image, we can understand the behavior of each type of error. If the Bias is small, we are close to the center, but if it increases, we move away from it. With the variance we can observe that if the value is small the points are close to each other, but when increasing the points are dispersed. Only if we lower the error in both can we have the best results.

By lowering the Bias and Variance we reduce the total error of the model, which brings us closer to having an optimal model. By achieving this we have a model that is not complex or very simple. Why is this important? If we have a complex model, we tend to have overfitting, but if we have a simple model what we will achieve is underfitting.

References

Cosio, N. A. L. (2022, 16 abril). Guía definitiva A Bias-Variance Tradeoff - Nicolás Arrioja Landa Cosio - Medium. *Medium*. https://medium.com/@nicolasarrioja/gu%C3%ADa-definitiva-a-bias-variance-tradeoff-94fb5c118d0f

KeepCoding, R. (2023, 27 abril). Diferencias: Underfitting vs overfitting. *KeepCoding Bootcamps*. https://keepcoding.io/blog/underfitting-vs-overfitting/

Chemama, J. (2020, 29 mayo). *How to solve underfitting and overfitting data models | AllCloud*. AllCloud. https://allcloud.io/blog/how-to-solve-underfitting-and-overfitting-data-models/

Shetty, B. (2022). What is the curse of dimensionality? *Built In*. https://builtin.com/data-science/curse-dimensionality

What is the curse of dimensionality? | Data Basecamp. (2023, 15 marzo). Data Basecamp. https://databasecamp.de/en/ml/curse-of-dimensionality-en