SOLUTION TO MOST COMMON PROBLEMS IN ML

Portfolio Evidence U1

Professor: Victor Alejandro Ortiz Santiago
Universidad Politécnica de Yucatán

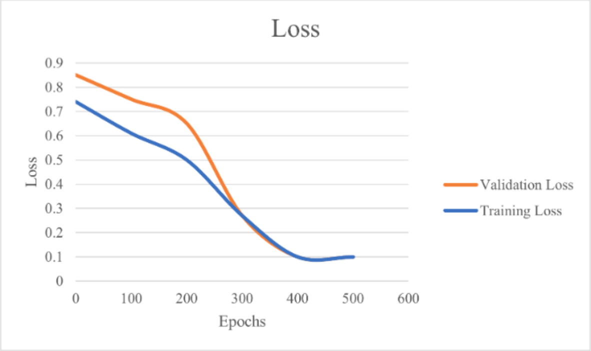
JESUS EDUARDO CASAS NAVARRO
2009024@upy.edu.mx

Overfitting:

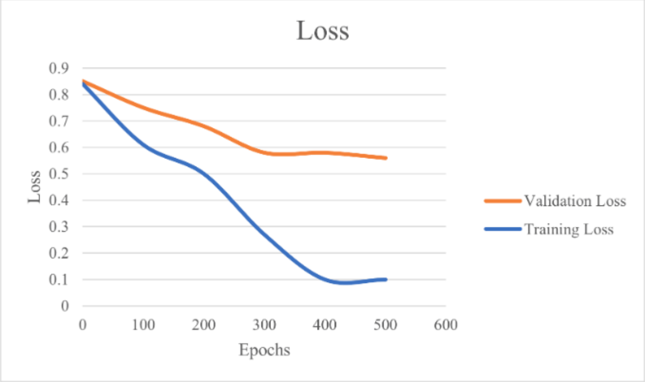
Overfitting occurs when a model is over trained, starting to learn different things within the dataset and making focus on noise or irrelevant information from the elements that should be learning. The overfitting produces the wrong functionality of the model while doing predictions. It is something common that a model which trains with a lot of epochs using a low quantity of elements overfits fast because it starts to memorize the elements as they are so it will not learn how to predict new data.

An Overfitted model, since it memorizes the training set, has almost a perfect value of accuracy while proving its functionality with repeated images, but fails constantly when it tries to predict the result with new data.

There are different parameters used to analyze the training of a model, the principal ones are related with the loss function while doing epochs of training with the elements taken from a dataset, training loss and validation loss. A model generalizes data in order to learn how to handle new and unseen data accurately and obtain a good prediction that results in classifying an element or getting a numerical value. Validation and training loss are both metrics; training loss is used to see how the model is fitting the training set, and the validation loss assesses the performance on the validation set. These metrics should be advancing near to the equality to determine a good train.



When the validation loss is greater to the training loss and never go near each other, then the model is overfitted.



Underfitting:

The opposite occurs when the loss is greater than the validation loss, then the model is underfitted. When underfitting happens then the model is not trained enough to generalize the data and then find relationships between input data and outputs, excluding important features or using not enough relevant elements to train. So, in place of memorizing the training data or learning unnecessary information, the model is unable to even generalize information from new or unseen data, therefore its performance while predicting results is poor or it cannot do it at all.

Outliers:

When an observation made is located beyond the datapoints at a pre-defined range of distribution or a means for the dataset, then it is called an outlier. It is separated from the common pattern of data and is used to be more visible when the analysis of compressed data is made. These are in simpler words extreme values that are far away from the general tendency of a dataset, being not consistent with the rest of observations. It can be present a situation when an outlier is the result of a measurement error, but it usually just abnormal data which should not be included in the selected data to work with the model.

The outliers can affect the performance of the model if they are not treated correctly while working the dataset, producing distortions in the metrics and measurements of the model while doing a training process, resulting in less accuracy values and directly affecting the results of the predictions made, also producing the necessity of longer periods of training.

Solutions for overfitting:

Overfitting has many useful solutions in order to be avoided while training models, these actions can vary depending on the model, but as part of the common there are some general points to take into account, however it also depends on the complexity of the model. Training a robust network takes a long time and also uses a high quantity of computational resources, so it is important to apply techniques and methods to try to have a good training from the first time:

- The training can be early stopped before the model starts to learn unnecessary data or noise from the dataset, but it is also quite complex to know when to stop it, so it can result in underfitting. There are some observation that can be performed during the training, principally the observation of the loss functions and the accuracy that firstly have greater changes at the beginning of the training, and start to be lower when it is almost trained completely, so a good idea is to stop the model training when there is not high differences from the accuracy of an epoch to another (by almost a thousandth. In addition, there should be also adjusted the number of epochs and batch size to find the most functional values or modify hyperparameters to have the best train.

- Avoid using noisy or too irrelevant/repeating data for the training, also it improves when a variety of elements is used, this because if there is only used a limited quantity of data and the epochs are more than the ones that can cover the number of elements then the model starts to memorize data in place of looking for relationships for future predictions. More data in the training data set can avoid this, or adjusting the number of epochs to one that can be covered with the actual data.
- Select the relevant data that can fit better the training dataset, noisy data is only used to make more stable the model by making it work with many kind of elements but having the same purpose.
- Feature selection and regularization, selecting the important/relevant features to avoid the irrelevant data, it is also suitable to regulate the number of features that are being used, reducing noise for complex models.
- There are also some activation functions that could optimize the training process, and regulation functions that help to avoid overfitting by establishing metrics to stop the process automatically when the right values are obtained. It is also common to use kind of check points, functioning as points to return to a right moment of the training when the model overfits, avoiding the need of train it from 0 again.

Solutions for underfitting:

While overfitting is the over train of a model, since the underfitting is the opposite concept, the solutions to handle and prevent this kind of situations are also similar to the contrary of overfitting solutions:

- To prevent overfitting is common to make the model less complex, for underfitting problems then it could be necessary to add complexity since it may not have enough resources to recognize the patterns and relationships among all data.
- In place of reducing the number of features, add more in order to avoid simple data, and reduce the regularization of the information values, so the patterns are easier to be detected and differentiated in order to have accurate predictions.
- Add more epochs for the training to increase the training of the model.
- Reduce noise from data to make it clearer for the model analysis.

Solutions for outliers in datasets:

It is important to treat outliers before they affect our model's performance and training processes.

- When outliers appear, it is normal to delete these abnormal observations since they are far away from our means and if they are result of errors while introducing or processing data.
- It is also a good option to use projection methods for dimensionality reduction in order to maintain the important information by having a new measurement unit from the features in the many dimensions and observations got.
- Having a standardization of the ranges for the values can also works, to apply specifically for outliers, these can be limited by using capping functions that will limit the outlier to a specific value.
- Outliers can be replaced by the imputation using the mean/media/random or can also be treated separately from the sample data that follows the common pattern, then it can be created a group if the number of outliers is high, to work them in each group and then combine the output values as a new mean to avoid its related problems.

Dimensionality problem:

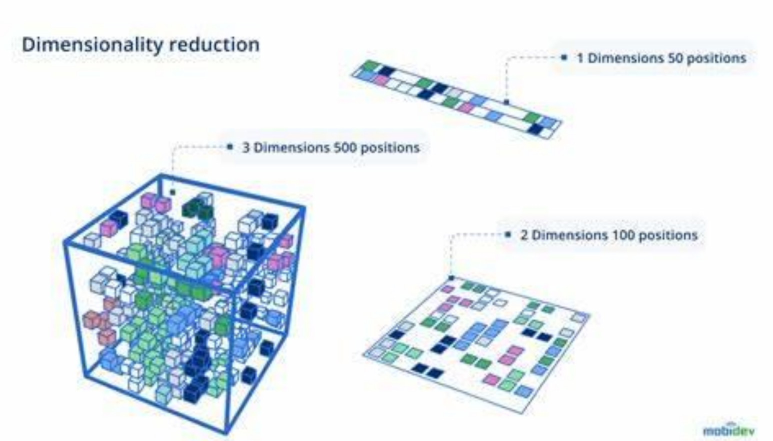
The dimensionality is defined as a phenomenon that increases the number of dimensions related with data, it can improve the quality of data but there is also an increment of noise and irrelevant information. The problem then is related principally with the cost computerization power that is needed to analyze data as more dimensions are present.

Data is described by features, and those features are each one a dimension to process, with more features are more characteristics but also more dimensions are more power needed, but also these features or dimensions are group together to create datapoints, finding a mean or common pattern for the data, while more datapoints then the information can get separated far away from the rest, producing outliers or clouds of datapoints that a function for the model can not fit at all or get really complex to work.

The curse of dimensionality also has a statement provided by a study realized from Hughes, the increased number of dimensions gives more characteristics/features so quality of data can be partially increased too, but at a certain point, many features/dimensions start to deteriorate the performance of the system/model.

Dimensionality reduction process:

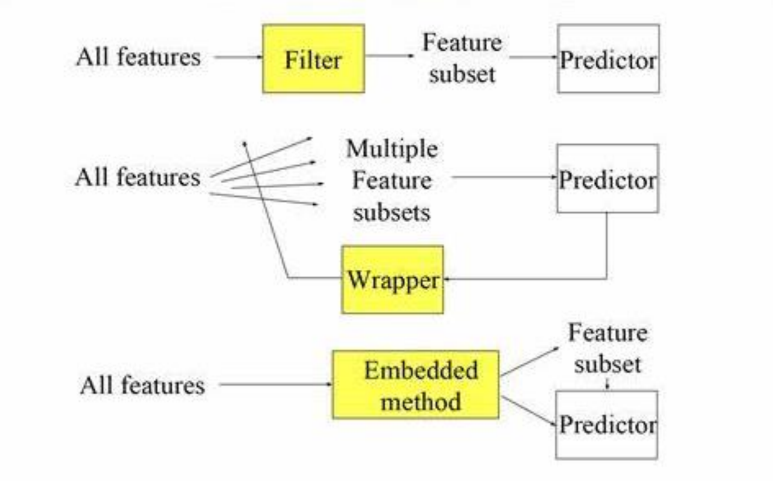
Dimensionality reduction is a technique used to reduce the dimensions/features of a dataset in order to have the possibility of analyzing better the data and make it easier for the model to work with it. Converting a dataset with a large quantity of features in a one that has a low number of dimensions but containing the most important information as possible.


Dimensionality reduction
1 Dimensions 50 positions
3 Dimensions 500 positions
2 Dimensions 100 positions
mobidev

The dimensionality reduction is a technic with 2 based factors in order to be performed:

Feature selection:

Selecting a subset of features from the whole dataset that are the most important and relevant for the problem solution. There are different methods which involve ranking the relevance of each feature, defining criteria to select them and combine them with the model training process. This factor has the purpose of having a dimensionality reduction with the relevant features retaining the important elements as mentioned.



Feature Extraction:

After selecting this subset of features, those are extracted by creating new features from the original ones, capturing the original data into a low dimension space. Applying different methods to obtain the result and then work.

Bias-Variance trade-off:

The bias and the variance in machine learning are considered as reducible errors, this is because help us to know also how is going the performance of the model. The bias then is a value given by the existent difference between the actual values (correct) and the prediction of the values. This value helps us to know in simple words the assumptions made of the model about data to be able to predict new input data, when the bias is going to high values then it means that the model is not learning from as much features as there are, so its assumptions about the data are simple, therefore the model is not learning effectively as it should. After that, the model will not have good results neither in the training nor predicting. The high values on the bias could be seen also as the model being underfitting, since it is not learning enough from important features and is making the model too simple to predict well new elements.

On the other hand, the variance is defined as the opposite of the bias, being the value of sensitivity in the model, is a data spread measure and helps the model to see the enough time the data to learn and find patterns, but it is important to avoid the model to pass more than the enough time with that data or it starts to learn only the necessary for the data principally used and will not know the recognition of patterns for new data.

Bias and Variance are directly related, for then it is important to find also an equilibrium or middle point where both can be reduced as more as possible without affecting or getting focused on one of them, this example is what also happens with the loss functions (validation and training loss.).

The bias-variance trade-off is about looking for the balance between the two reducible errors in order to have the most optimized model as possible in order to make it capture patterns while ignoring noise from data, so then the model can recognize well patterns and generalizing it to the new data. In addition, the model will have the adequate complexity to have good results for both, training and test processes.

References/Digital Resources

1. *What is overfitting?* (s/f). Ibm.com. Recuperado el 16 de septiembre de 2023, de https://www.ibm.com/topics/overfitting
2. Valdenegro-Toro, M., & Sabatelli, M. (s/f). Machine learning students overfit to overfitting. Arxiv.org. Recuperado el 15 de septiembre de 2023, de http://arxiv.org/abs/2209.03032
3. (S/f). Baeldung.com. Recuperado el 15 de septiembre de 2023, de https://www.baeldung.com/cs/training-validation-loss-deep-learning
4. Training loss and validation loss in deep learning. (s/f). Stack Overflow. Recuperado el 15 de septiembre de 2023, de https://stackoverflow.com/questions/48226086/training-loss-and-validation-loss-in-deep-learning
5. Overfitting and Underfitting in Machine Learning. (s/f). Www.javatpoint.com. Recuperado el 15 de septiembre de 2023, de https://www.javatpoint.com/overfitting-and-underfitting-in-machine-learning
6. databasecamp-blog. (s/f). What is the curse of dimensionality? Data Basecamp. Recuperado el 15 de septiembre de 2023, de https://databasecamp.de/en/ml/curse-of-dimensionality-en
7. Bonthu, H. (2021, mayo 21). Detecting and treating outliers. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/
8. Nichani, P. (2020, abril 22). OutLiers in machine learning. Analytics Vidhya. https://medium.com/analytics-vidhya/outliers-in-machine-learning-e830b2bd8660
9. Chemama, J. (2020, marzo 10). How to solve underfitting and overfitting data models. AllCloud. https://allcloud.io/blog/how-to-solve-underfitting-and-overfitting-data-models/
10. Follow, D. (2017, noviembre 23). ML. GeeksforGeeks. https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/
11. Introduction to dimensionality reduction. (2017, junio 1). GeeksforGeeks. https://www.geeksforgeeks.org/dimensionality-reduction/
12. Dimensionality reduction - introduction to machine learning. (s/f). Wolfram.com. Recuperado el 16 de septiembre de 2023, de https://www.wolfram.com/language/introduction-machine-learning/dimensionality-reduction/
13. Banoula, M. (2021, febrero 26). Bias and Variance in machine learning: An in depth explanation. Simplilearn.com; Simplilearn. https://www.simplilearn.com/tutorials/machine-learning-tutorial/bias-and-variance

14. Follow, P. (2020, febrero 3). Bias-variance trade off - machine learning. GeeksforGeeks. https://www.geeksforgeeks.org/ml-bias-variance-trade-off/