Machine learning is the core of artificial intelligence, involving statistics, system identification, approximation theory, neural networks, optimization theory, computer science, brain science and many other fields. It studies how computers can simulate or implement human learning behavior to acquire new knowledge or skills and recognize existing knowledge structures to continuously improve their own performance.

In contrast to traditional machine learning, which uses experience to improve the performance of the system, machine learning now uses data to improve the performance of the system. Data-based machine learning is one of the key approaches in modern intelligence technology, which looks for patterns from observing data and uses these pattens to make predictions about future data or unobservable data.

However, this technology can be unreliable when there are data doppelgangers which usually occur when independently derived data are very similar to each other. Under this condition, the models will perform well no matter how they are trained [1]. Given that unidentifiable duplicates may incorrectly inflate the predictive accuracy and confidence in differential expression, doppelganger checking should be part of the standard procedure for datasets [2].

According to [2], although some studies are published by groups from different regions, a careful reading of the papers reveals that their datasets share a set of samples. This kind of international collaboration is beneficial to research but presents a challenge to researchers developing independent validation [3]. In their detailed examination of the ovarian cancer database, they found that only 17% of the records were non-unique, including duplicate records from different datasets from the same institution [2].

In my perspective, doppelganger effects are not unique to biomedical data. This kind of phenomenon also appears in many cases. When the samples in training and validation are similar, there is no doubt the model will have a false accuracy level.

In machine learning algorithm, we usually divide the original data set into three parts.

* Training set: training the model
* Validation set: selecting the model
* Test set: evaluating the model

The dataset is firstly divided into a training set and a validation set. Since the model construction process also requires testing the configuration of the model and whether the training is overfitting or underfitting, the training data is further divided into two parts, a training set for training and a validation set for testing. The training set is used to train the neural network model, and then the validation set is used to verify the validity of the model and select the model that gives the best results. The validation set can be used repeatedly and is mainly used to assist in building the model. Finally, once the model has "passed" the validation set, the test set will be used to test the final results of the model, evaluating the accuracy of the model, the error, etc.

For the model fitting problems:

* If the correct rate is low on both the training set and test sets, then the model is in an underfitting state and the parameters need to be adjusted.
* If the correct rate is low on the training set and high on the test set, the data set is faulty.
* If the correct rate on the training set is high and the correct rate on the test set is low, the model is overfitted.
* If the correct rate is high on both the training and validation sets, but low on the test set, the model’s generalization ability is insufficient.
* If the correctness of the model is high on both the training and test sets (including the validation set), then that’s correct.

When optimizing a neural network using mini-batch gradient descent, a batch is randomly selected from the training data each time and the parameters are updated according to the average gradient value of all the samples in this batch. If a certain image or similar images presence many times in the training set, at the extreme, the selected batch will have a large proportion of that images. Therefore, when calculating the gradient value, this image will have a large weighting and then the neural network will have a strong discrimination of this image and a relatively weak discrimination of other images. This means that as long as this image is classified in the right category, it will have a small loss value. If the validation and test sets are a randomly selected portion of all the data, the validation and test sets will have a large portion of images that are similar to this image. As the network is updated, the network would have a good ability to classify this image, the accuracy achieved in the validation and test sets would be high. However, if the image does not occur with a high probability in reality, the accuracy should not be as high. To sum up, the it is better to have samples in the training set that are close to the actual occurrences. When this kind of images have a high occurrence in reality, they are acceptable. If these similar images are removed, it is important to ensure that the number of samples in this category is not less than that of other categories. Otherwise, similar to the analysis above, the recognition of that category will be weaker.

To avoid doppelganger effects, there are three general methods of dividing the datasets.

1. Hold-out method: If the data is relatively small, divide the training set, validation set and test set to: 60% training set, 20% validation set, 20% test set. If the data is abundant, Only a small portion of the data should be used as the test and validation sets and the rest as the training set.
2. Cross validation method: The original training dataset is split into K non-overlapping sub-datasets, and then K model training and validation sessions are done. Each time, one sub-dataset is used to validate the model, and the other K-1 sub-datasets are used to train the model. Finally, the training and validation errors are averaged for each of the K times.
3. Bootstrapping: Each time we take one sample from the data set D as an element of the training set, then put that sample back and repeat the act m times, so that we can get a training set of size m, in which some samples are repeated and some are not, and we use those that do not appear as the test set and the rest as the training set.

Comparing the accuracy levels of the training set and test set with the method mentioned above to see what is the problem could help to avoid data doppelganger.

References:

1. Li Rong Wang, Limsoon Wong and Wilson Wen Bin Goh, “How doppelgänger effects in biomedical data confound machine learning,” 2021, Elsevier Ltd
2. Levi Waldron, Markus Riester, Marcel Ramos, Giovanni Parmigiani, Michael Birrer, “The Doppelgänger Effect: Hidden Duplicates in Databases of Transcriptome Profiles,” Journal of the National Cancer Institute, 2016
3. Sandra Robinson, “Databases and Doppelgängers: New Articulations of Power,” Published by Johns Hopkins University Press, Volume 26, Number 4, Fall 2018, pp. 411-440