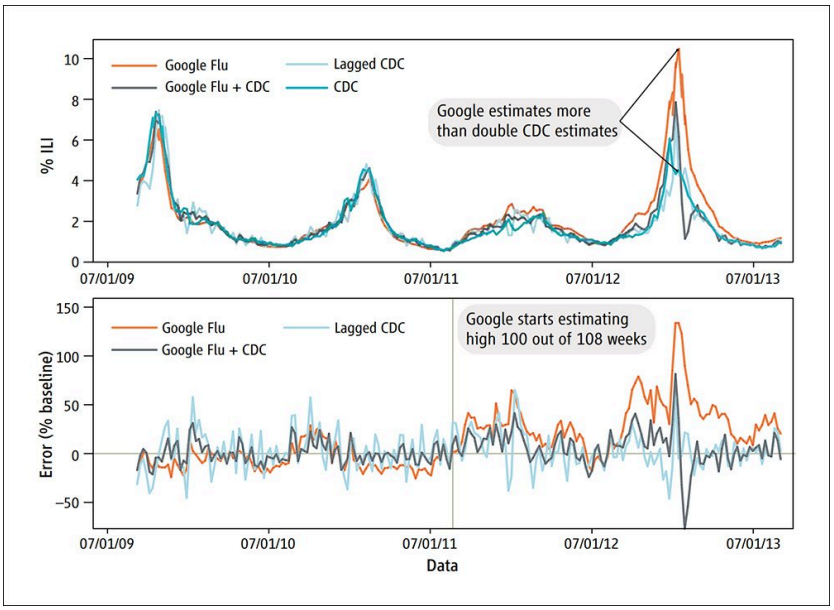
Overfitting in Machine Learning

Originally published at https://blog.metaphysic.ai/overfitting-in-machine-learning/

January 26, 2023

Overfitting is the tendency of a trained machine learning model to only work efficiently on the *exact data on which it was trained*. Since the objective of training a model is usually to ensure that it will operate efficiently on similar (but not identical) data when it's deployed, this is generally an undesirable state for the model.



Google's 'Flu Trends' platform came under scientific criticism as a textbook case of overfitting, where it came to light that the system may have been trained on too few instances of data, and that a very specific trend in low-volume data was being applied at scale, thereby overshooting the CDC's own predictions, and the actual instances of flu which occurred in the predicted period. Source: https://gking.harvard.edu/files/gking/files/0314policyforumff.pdf

During training, most machine learning models are intended to gain a broad and applicable understanding of a *domain* (which could be cats, churches, cars, or a class in a much larger model), so that they'll be able to recognize a class on which they were trained, so long as it falls within the learned parameters of the model's trained weights.
This more desirable state is known as generalization, with 'general' indicating that the model is capable of recognizing 'unseen' examples of a trained class or label.
Sometimes a model can also be well-generalized within the scope of a too-limited amount of data, and will only work well on *similar scales of data* to that on which it was trained. A common pitfall in analytics is when researchers apply the predictions of such 'small scale' sets to much higher volumes of data, assigning small>large probabilities that are plausible but inaccurate.

Causes of Overfitting

Overfitting can occur when the data is either too constricted and repetitive (it all looks the same, even if there is a lot of it) or inadequate in volume (it's relatively diverse, but there's hardly any of it).
It can also occur when training is not interrupted in time, allowing the model to 'overstudy' the data.
Additionally, a badly-configured model, such as one that is over-complex, can also 'over-weight' data so that it obtains specificity instead of generalization.
In any case, the likely result is *memorization*, where the trained model becomes 'obsessed' with a particular example or stance of data, and is likely to spit it out in responses to any query at inference time, like a person with dementia or monomania.
The training process iterates at scale over the dataset supplied to it, and most pipelines are designed to extract multiple and varied features from a great number of data examples, looking for common features that could define entities and classes. Either feeding excessively similar data at scale to the system, or forcing the system to obsess over particular data, can lead to overfitting.

Hazards of Overfitting

One possible consequence of an overfitted model is that it may be capable of exactly reproducing the data it was trained on. In the case of language models that may have been trained on sensitive data with personally identifiable information (PII), this constitutes a security risk.
In the case of generative models, depending on how the current crop of lawsuits unfold, data memorization could constitute the difference between learning a style and reproducing an original (and perhaps copyrighted) artwork.

Above, generated images from Stable Diffusion, below, very similar source images from the LAION subset on which Stable Diffusion was trained. Source: https://arxiv.org/pdf/2212.03860.pdf

In the image above, we can see that Stable Diffusion, trained on the LAION dataset, can effectively reproduce the semantic layout of a training image, even if the generation is not pixel-perfect, raising the question as to whether a diffusion model of this type is prone to violate copyright.
Stable Diffusion is a multi-domain model that seeks to be able to represent any concept visually. It's possible that incidences like those above, which are uncommon, occur because the original dataset had very little material for a term that the user has referenced in a text-prompt; therefore the model is, perhaps, presenting the only material that it has for that term, constituting a kind of 'inner overfitting', where the broader model itself is well-generalized, but contains instances of classes or labels that have very little associated visual material.
The primary traditional consequence of overfitting, however, is that the model is simply ineffective at making predictions for similar data. In practice, this could lead to a face recognition model that 'sees' one particular face that it was trained on, even when it is looking at a different face; a generative model that produces repetitive and near-identical images, or in which the central concepts are brittle and not easy to constrain into unlearned configurations; or a weather prediction model that's waiting for the exact circumstances of the data it was trained on before issuing a hurricane warning.

Detecting Overfitting

Usually a trained machine learning practitioner will be able to understand that a model has become overfitted simply by looking at the output. However, there are systematic methods to test for overfitting, which can be useful when the overfitted quality is not likely to emerge at a casual inference, or in predictive models, where an accurate prediction can't be realistically simulated (i.e., you have to wait for real events to unfold in order to know if the model is working well).
In a supervised learning scenario, K-fold cross-validation involves holding back one part of the data as a test set, and testing the model on the data that was held back for this purpose. If the tests produce diverse results, the model has generalized successfully; if, on the other hand, the model fails to make valid predictions on the test data, this *can* be a sign of overfitting, where the model fails to recognize and act on what it now considers out-of-distribution (OOD) data.

Is Overfitting Ever Desirable?

Technically, the term 'overfitting' is pejorative, and indicative of an undesirable training result. However, there are cases where only small variations in the output are desired at inference time, and where the user may want to make only very minor changes to the original source data, and reflect these in the output from the model.
In image synthesis, such workflows generally result in full-size checkpoints (or models) that are single-task and 'disposable', not intended either for archiving or general deployment; but if we're going to be semantically accurate, we could call such constrained models 'close-fitted' rather than over-fitted, since the latter term indicates a non-usable model.

Underfitting

Underfitting occurs for almost the inverse reasons to overfitting: for instance, the data may be so extraordinarily varied that it does not easily yield discrete features; the training session may be badly-configured, with an excessively high learning rate that 'skates' so quickly over each data input that it does not have time to examine it for common traits; or the model may not have been given adequate complexity or time to train properly.