

5

REMOVING UNDESIRABLE INFERENCES FROM A MACHINE LEARNING MODEL**FIELD OF THE INVENTION**

10 The invention relates to a system and method for removing undesirable inferences from a machine learning model.

BACKGROUND OF THE INVENTION

15 The lack of insight provided concerning the ‘reasoning’ of machine learning (ML) models does not instill trust in end-users and limits the acceptance of ML in industry, even when the end-users are domain experts. Systems have been proposed to explain as well as contextualize ML outputs in technical systems in industrial applications, which fosters acceptance of the ML decision support. However, a domain expert may
20 still lose trust in the ML model if either or both of the model output and its explanation are perceived to be incorrect. ML models are typically trained using historical data and it is not uncommon that, with the passage of time, rules previously inferred may no longer be valid. Rules learned by the ML model at any time may contradict the expertise of the domain expert. Machine Unlearning (MuL) has been proposed as a
25 way of “unlearning” or removing undesirable inferences, as described for example in L. BOURTOULE ET AL. Machine Unlearning. arXiv:1912.03817, 2020.

SUMMARY OF THE INVENTION

30 There is a need to identify the source of undesirable inferences which are to be removed from an ML model. This need is met by the subject-matter of the independent claims. Optional features are set forth by the dependent claims and by the following description.

- 2 -

According to a first aspect, there is provided a system for removing undesirable inferences from a machine learning model. The system comprises a search component configured to: receive a rejected explanation of model output provided by the machine learning model; identify data samples to unlearn by selecting training samples from training data that were used to train the machine learning model, the selected training samples being associated with explanations that are similar to the rejected explanation (e.g. according to a calculated similarity measure, with the calculated similarity measure satisfying (e.g. exceeding) a predetermined threshold); and pass the data samples to unlearn to a machine unlearning unit.

10

In this way, a machine learning pipeline may be provided that includes not only an explainer path (to explain the model output) but also a feedback path (to correct or improve the model), providing domain experts with the capability to tune ML models by leveraging model output explanations. The system may be applied in an industrial setting to improve quality of ML model outputs. The system promises high potential in the area of transfer learning, where a trained machine learning model is used as a starting point for training on a new industrial process, because it offers the possibility to unlearn predictions and explanations that do not apply to the new process.

20 In one implementation, the search component may be configured to identify the data samples to unlearn by: obtaining a training explanation associated with a respective candidate training sample selected from the training data; calculating an explanation similarity measure between the rejected explanation and the training explanation; comparing the calculated explanation similarity measure to an explanation similarity threshold; and selecting the training sample for inclusion in the data samples to unlearn in response to the calculated explanation similarity measure satisfying the explanation similarity threshold. The search component may be further configured to calculate a sample similarity measure between the rejected sample and the training sample, to compare the calculated sample similarity measure to a sample similarity threshold, and to identify as data samples to unlearn only those selected training samples whose calculated sample similarity measures also satisfy the sample similarity threshold. The search component may be further configured to repeat the obtaining, calculating, comparing, and selecting for one or more further training samples in the training data.

25

30

By "similarity measure" is meant any suitable metric for representing similarity. In some implementations, the search component may be configured to calculate the explanation similarity measure as either (i) a distance metric between the rejected explanation and the training explanation, the distance metric comprising one or more of a Euclidian distance, a City-Block distance, a Mahalanobis distance, a Huber distance, or an elastic distance measure such as Dynamic Time Warping or a Levenshtein distance, or (ii) a Jaccard similarity coefficient. The sample similarity measure described herein may be calculated in the same way.

The explanations may be rule-based, in which case the search component may be configured to identify the explanations that are associated with training samples as being similar to the rejected explanation in response to one or more of (i) premises, (ii) structure, and (iii) decision values of the rule-based explanations being the same or similar (e.g. within a predetermined similarity threshold).

In the case that the explanations comprise prototypes, the search component may be configured to identify the explanations that are associated with training samples as being similar to the rejected explanation in response to the same prototype being produced.

The machine unlearning unit may be configured to remove undesirable inferences from the machine learning model by assigning, to the data samples to unlearn, weightings for a training loss function before retraining the machine learning model using an updated set of the training data. The weightings may be set at 0 to effect a complete removal of the unwanted data samples or any weighting between 0 and 1 may be used for advantageous partial removal, if this is desired.

The weightings assigned to the data samples to unlearn may be based on the similarity measures calculated for those data samples.

- 4 -

The weightings may be assigned to some but not all of a plurality of signals in the data samples to unlearn, allowing useful signals in the data samples to unlearn to be retained for further training of the model.

5 The machine unlearning unit may be configured to incrementally change the weightings based on the Implicit Function Theorem applied to optimality conditions of a machine model training loss minimization problem, in order to avoid complete retraining in favour of incremental training.

10 To readily implement complete removal of the undesirable inferences, the machine unlearning unit may be configured to remove undesirable inferences from the machine learning model by retraining the machine learning model using an updated set of the training data from which the data samples to unlearn have been removed.

15 Instead of removing the unwanted samples completely or reducing their weighting, the machine unlearning unit may be configured to alter the data samples to unlearn by finding perturbations to the data samples that result in an explanation derived from the perturbed data samples matching a non-rejected explanation. This provides an expert with the possibility to generate artificial training data from existing training data in order
20 to change the model's decision boundary in such a way that the yielded explanations comply better with the expert's domain knowledge.

According to a second aspect, there is provided a method for removing undesirable inferences from a machine learning model. The method comprises: receiving a rejected
25 explanation of model output provided by the machine learning model; identifying data samples to unlearn by selecting training samples from training data that were used to train the machine learning model, the selected training samples being associated with explanations that are similar to the rejected explanation according to a calculated similarity measure; and passing the data samples to unlearn to a machine unlearning
30 unit.

- 5 -

According to a third aspect, there is provided a computer program product comprising instructions which, when executed by a computer, cause the computer to perform the method of the second aspect.

5 According to a fourth aspect, there is provided a non-transitory computer storage medium having stored thereon the computer program product of the third aspect.

Any optional features or sub-aspects of any aspect apply as appropriate to any other aspect.

10

The above aspects and examples will become apparent from and be elucidated with reference to the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

15

Illustrative examples will be described with reference to the following drawings, in which:-

Figure 1 shows a machine learning system including not only an explainer path but also a feedback path;

20

Figure 2 shows the feedback path in the system of figure 1 in more detail;

Figures 3 and 4 illustrate methods of identifying samples to be unlearned; and

Figure 5 illustrates a use case of the system of figure 1.

DETAILED DESCRIPTION OF EMBODIMENTS

25

Figure 1 shows a high-level view of a machine learning system 100 including not only an explainer path but also a feedback path. The system 100 may be applicable wherever an ML model is used in an industrial application and independently of the type of ML model. The ML model 102 is trained on the basis of training data 104 to provide ML model output 106 to an end-user 112 when a sample 105 is input to the model 102. The model output 106 may comprise a recommendation to the end-user 112 to take a predetermined action. An explainer 108 also receives the model output 106 and provides to the end-user 112 an explanation 110 of the model output 106. The

30

- 6 -

path model-explainer-end-user constitutes the explainer path. The explanation 110 may be produced in any known way. One example of providing explanation of ML model output is described for example in related European patent application no. 20 196 232.1. According to the present disclosure, the path end-user-model constitutes the feedback path, whereby the end-user 112 is able to provide feedback 114 to the ML model 102.

Figure 2 shows the feedback path in the ML system 100 in more detail. In this example, the end-user 112 is a control room operator as the domain expert for whom the ML model 102 was created. The operator 112, using his own expertise, can evaluate the explanation 110 provided to him by the explainer 108 and consequently provide the feedback 114. The operator 112 may deem the model output 106 and/or the explanation 110 to be inaccurate and thus use the feedback path to have this undesirable inference removed from the system 100. The operator 112 may for example choose whether the MuL unit 122 is to remove this undesirable inference from the system 100 completely or to consider it less important (in the latter case, the affected training data 104 may not be removed but rather assigned less weight in the ML model 102), as described further below.

An example of an inference which the operator may like to remove could comprise for example an erroneous inference which the ML model 102 made based on training data 104 gathered as a result of apprentice operator actions. Another example concerns changes in the plant equipment or production process which have been implemented over a period of time, so that older rules inferred by the ML model 102 no longer reflect the new reality. Yet another example involves rules inferred by the ML model 102 which are false from a domain perspective, such as rules which have been falsely learned on the basis of training data 104 provided when the plant equipment was in maintenance mode.

Following the decision to remove the undesirable inference, the operator 112 initiates a removal process 116 to remove the undesirable inference, designating the explanation 110 as a rejected explanation.

- 7 -

Since the operator 112 sees only the model output 106 and explanation 110 when calling for the undesirable inference to be removed, a search component 118 is provided to consolidate all raw data points that contributed to the undesirable inference as one or more data samples to unlearn 120.

5

Referring to figures 3 and 4, with figure 4 illustrating step 302 of figure 3 in more detail, the search component 118 is configured automatically to identify the data samples to unlearn 120 by selecting training samples from the training data 104 associated with explanations that are similar to the rejected explanation.

10

For example, the search component 118 is configured to:

1. Receive (301) the rejected explanation identified by the operator 112 when initiating the removal process 116, the rejected explanation being derived from or associated with at least one rejected sample;
- 15 2. Identify (302) the data samples to unlearn 120 by selecting from the training data 104 one or more training samples that are associated with explanations similar to the rejected explanation (and which therefore contributed to the undesirable inference), for example by:
 - i. Obtaining (401) a training explanation associated with a respective
20 candidate training sample selected from the training data 104, for example by computing for the training sample an explanation (e.g. by inputting the training sample to the ML model 102 to generate model output and inputting that model output to the explainer 108), and/or by retrieving a precomputed explanation associated with the training
25 sample, where the precomputed explanation was computed upfront for one or more versions of the ML model 102 and stored for later retrieval;
 - ii. Calculating (402) an explanation similarity measure between the rejected explanation and the training explanation, and optionally also calculating a sample similarity measure between the rejected sample
30 and the training sample;
 - iii. Comparing (403) the calculated explanation similarity measure to an explanation similarity threshold and optionally also comparing the calculated sample similarity measure to a sample similarity threshold;

- 8 -

- iv. Selecting (404) the training sample for inclusion in the data samples to unlearn 120 in response to the calculated explanation similarity measure satisfying the explanation similarity threshold, the selected data samples to unlearn 120 being optionally further limited to include only those training samples whose calculated sample similarity measures also satisfy the sample similarity threshold; and
- v. Optionally repeating (405) steps 401-404 for one or more further training samples, for example by traversing all training samples in the training data 104, or by traversing select representative samples (e.g. cluster centroids derived by grouping all training samples into clusters using an unsupervised clustering algorithm); and
3. Pass (303) the data samples to unlearn 120 to the MuL unit 122.

In the case that the sample 105 is used also as training data for further development of the ML model 102, the sample 105 may be incorporated in the data samples to unlearn 120.

The process of selecting training samples to unlearn as shown in figures 3 and 4 is thus based on similarity measures. Respective similarity measures may be used for both the explanations and the samples. Generally speaking, if a training sample similar to the rejected sample with the rejected explanation yields an explanation which is similar to the rejected explanation, it may be a good candidate for unlearning. So, two similarity measures may be defined: the sample similarity measure and the explanation similarity measure. In the context of the present disclosure, the term "similarity measure" may be taken to include also distance measures as a measure of the difference between two samples or explanations.

Concerning the sample similarity measure, a sample in machine learning can be described as a vector or matrix s and the dimension s_i ($x_{i,j}$ in case of a matrix) often share the same numerical data type (e.g. $U = \mathbb{R}^N$ or $U = \mathbb{N}^N$) and all data samples share the same dimension N . Examples comprise a series of a temperature measurements sampled at equidistant times (say, for instance once per minute), or the pixel values of an image. Further examples comprising vectors of different length and

non-homogenous data types are described below. The similarity of two samples s^1 and s^2 can for instance be measured by the Euclidian distance: $d(s^1, s^2) = \sqrt{\sum_{i=1}^n (s^1_i - s^2_i)^2}$.

If s^1 and s^2 are equal, then $d(s^1, s^2)$ will be zero. The less s^1 and s^2 are alike, the larger $d(s^1, s^2)$ will become. The Euclidian distance can be replaced or supplemented with other distance measures such as the City-Block distance, Mahalanobis distance, or distances that are more robust to outliers like the Huber distance. Additionally or alternatively, similarity measures like the Jaccard coefficient could be used. If the dimension of the vector varies, elastic distance measures like Dynamic Time Warping (DTW) or the Levenshtein distance can be used. When applied to matrices, these distance measures can be applied row- or column-wise.

Regarding the explanation similarity measure, the question of how to measure the similarity of two explanations may depend on the structure of the explanation. Using the same notation, the similarity (e.g. distance) measure between two explanations e^1 and e^2 may be expressed as $d(e^1, e^2)$.

The explanations may comprise feature explanations. For example, feature explanation methods such as LIME (M. T. RIBEIRO ET AL. "Why Should I Trust You?" Explaining the Predictions of Any Classifier) or SHAP (Shapley sampling values as described in E. ŠTRUMBELJ ET AL. Explaining prediction models and individual predictions with feature contributions. Knowledge and Information Systems 41.3, 2014, pages 647-665) may be used to assign to each feature x_{ij} an importance weight w_{ij} . In this case, the similarity of features can be measured in the same way as the similarity of samples, as described above.

Another way of providing the explanations is by employing rules, for instance generated by the ANCHORS method (M. T. RIBEIRO ET AL. Anchors: High-precision model-agnostic explanations. Thirty-Second AAAI Conference on Artificial Intelligence, 2018) or by a decision tree surrogate model. One example of a rule-based explanation could be:

Rule 1: IF temperature $t > 100^\circ$ AND flow $f < 1 \text{ m}^3/\text{s}$ THEN product quality is good.

Two rule-based explanations may be considered to be similar if they have the same premises in the same structure (considered as features and comparison operator) and similar decision values. For instance, the following rule can be considered similar to Rule 1:

5 Rule 2: IF temperature $t > 105^\circ$ AND flow $f < 0.8 \text{ m}^3/\text{s}$ THEN product quality is good.

The decision regarding similarity can be extended by allowing additional or missing premises. For instance, the following rule might be considered similar to Rule 1 and Rule 2:

10 Rule 3 IF temperature $t > 105^\circ$ THEN product quality is good.

Methods like ANCHORs try to find rules that apply to a large subset of the training data 104 and thus will produce a small number of rules so that the above-outlined rule-based process will be sufficient in most practical cases.

15 Another way of providing the explanations 110 is by prototype. In this case, the explainer 108 presents a prototype that represents how machine learning represents the sample internally and optionally how similar the internal representation of the sample is to the prototype. If the explainer 108 only presents the prototype, two explanations 110 are similar or identical if the same prototype is produced. If the
20 explainer 108 also provides a measure of similarity to the prototype, this similarity to the prototype can be used. If the similarity or distance to all prototypes is delivered, a measure like the Euclidian distance can be used.

As described above in relation to figures 3 and 4, the training explanations for the
25 training samples may be generated on demand or generated upfront. With the above-described methods of measuring the similarity of a training sample s^t and the rejected sample s^r and the similarity of the corresponding training and rejected explanations e^t and e^r , one way of selecting training samples for inclusion in the data samples to
30 unlearn 120 (as described above with reference to steps 403 and 404) is by applying the two thresholds using the rule:

IF $d(s^r, s^t) < t_s$ and $d(e^r, e^t) < t_e$ then unlearn s^t ,

where t_s is the sample similarity threshold and t_e is the explanation similarity threshold. As described below, the values of $d(s', s^t)$ and $d(e', e^t)$ can be also used as weights (or to calculate weights) in the unlearning process.

5 Referring again to figure 2, the consolidated data samples to unlearn 120 are passed on to an MuL unit 122 employing an MuL algorithm (such as that described in BOURTOULE ET AL. Machine Unlearning) to be unlearned. The MuL unit 122 may employ any one or more of a number techniques for unlearning samples, as will now be described.

10 One readily implementable form of unlearning the unwanted samples 120 comprises the MuL unit 122 retraining the ML model 102 using an updated set of training data 104 from which the unwanted samples 120 have been removed.

15 In order to increase the flexibility of the forgetting mechanism, and as an alternative to the complete removal of the unwanted samples 120, the MuL unit 122 may apply remembrance factors to the unwanted samples 120, wherein each remembrance factor multiplies the term of its corresponding unwanted sample 120 in the training loss function. Choosing a remembrance factor of 0 is equivalent to removing the unwanted
20 sample 120 from the training data 104, while choosing a remembrance factor of 1 is equivalent to retaining the sample as before. Extra flexibility can be gained by choosing remembrance factors between 0 and 1, for instance based on the similarity measures, which means that the unwanted samples 120 may also be half-forgotten, for example. For the mapping of a similarity measure to the remembrance factor an inverted
25 sigmoidal function with upfront specified scale and slope parameters or a thresholding step function can be employed.

Additional flexibility can be gained by employing even more fine-grained remembrance factors that can vary for different components of the labels of the unwanted samples
30 120. These remembrance factors can be used partially to mask out the unwanted samples 120. In the exemplary case of an industrial plant in which the position of a temperature sensor was modified, instead of discarding the unwanted samples 120 in their entirety, fine-grained remembrance factors enable masking out only the now-

invalid temperature signal and save all the remaining information of the unwanted samples 120 for training. This can be achieved by setting the remembrance factors corresponding to the temperature signal data to 0 while keeping all other signals' remembrance factors at 1.

5

Besides the increased flexibility that remembrance factors deliver, they can also be leveraged by the MuL unit 122 to reduce the computational ML training effort by using mathematical sensitivity analysis of the optimal ML weights (where "weight" refers here to the degrees of freedom in the ML model 102, not the feature weights of an explanation 110) with respect to the continuous remembrance factors, based for example on the Implicit Function Theorem applied to the optimality conditions of the ML training loss minimization problem. The MuL unit 122 may thus change the optimal weights in correspondence to changes in the remembrance factors, in order to avoid complete retraining in favour of incremental training. The application of the Implicit

10

Function Theorem yields that the inverse of the so-called Hessian matrix of the training objective function yields the correct scaling of increments in the remembrance factors to the optimal ML weights increments. Hence, approximately optimal ML weight increments can be computed by iterative linear algebra methods, such as the conjugate gradients method, which only require Hessian matrix-vector products that

15

modern ML frameworks can compute efficiently for mini-batch approximation of the Hessian.

20

Additionally or alternatively, an approach using learning and unlearning using summations may be used by the MuL unit 122, as presented in Y. CAO ET AL.

25

Towards Making Systems Forget with Machine Unlearning. The approach comprises transforming learning algorithms into a summation form. To forget an unwanted sample 120, the approach only needs to update a small number of summations, which is asymptotically faster than retraining from scratch. This approach is general in that the summation form is from statistical query learning in which many machine learning

30

algorithms can be implemented.

Figure 5 illustrates how the ML system 100 as depicted in figure 2 may be applied in an industrial setting. The training data 104 comprises two features, temperature and

pressure, which determine whether a valve in a monitored system (e.g. industrial plant equipment) should be opened or remain closed (for the sake of simplicity, only an excerpt of the training data 104 is shown in figure 5). Considering one example in which the monitored system provides as the sample 105 a temperature of 80,0 °C and a pressure of 3,1 bar, the ML model 102 provides as model output 106 a recommendation to open the valve. The corresponding inference presented to the control room operator 112 as the explanation 110 is the rule "Temp. $\geq$ 80,0 and Pressure $>$ 3,0". Based on his expertise, the operator 112 may believe this explanation 110 to be incorrect (or no longer valid or applicable) and thus may want this undesirable inference to be removed from the system 100 completely. The operator 112 thus initiates the removal process 116, whereupon the search component 118 consequently retrieves all raw data points that contributed to this undesirable inference, in the manner described above. Figure 5 shows that 3 data samples to unlearn 120 were selected from the underlying training data 104 that fulfilled the criteria provided by the explanation 110. These data samples 120 are passed on to the MuL unit 122 to be unlearned.

In addition to unlearning the samples 120 which produced the rejected explanation, the MuL unit 122 may alter the samples 120 to produce an explanation that complies with the opinion of the end-user 112. To do so, the end-user 112 is given the option to alter the explanation 110 or annotate the associated sample 105 to capture domain insight. The system 100 then perturbs the data samples in such a way that the output of the explainer 108 matches the altered explanation. Sample perturbation (such as that described in M. T. RIBEIRO ET AL. "Why Should I Trust You?" Explaining the Predictions of Any Classifier) combined with an optimization or meta-learning process (such as that described in European patent application no. 20 196 232.1) can be used to alter the samples 120 resulting in the rejected explanation. If the process succeeds in producing useful new data samples, the original samples 120 can be preserved: The new data sample will impact the decision border of the ML model 102 in such a way that the explanation 110 provided by the explainer 108 also changes.

The end-user 112 as domain expert may be replaced by a system (e.g. an expert system) that encapsulates domain knowledge and provides the feedback 114 instead

of the human domain expert. The expert system can be another ML model. Thus, the decision on whether to remove the unwanted samples 120 completely or to remove their weighting (as described above) may be taken automatically by the expert system based on predetermined rules, based on a threshold of the proximity of the explanation to an acceptable explanation, or based on additional information concerning samples in the affected training data 104. The predetermined rules can be based on e.g. data/actions/knowledge of human domain experts which has/have already been modeled.

From the above, it is clear that one or more computer programs comprising machine-readable instructions can be provided which, when executed on one or more computers, cause the one or more computers to perform the described and claimed methods. Also, from the above it is clear that a non-transitory computer storage medium, and/or a download product, can have stored thereon the one or more computer programs. One or more computers can then operate with the one or more computer programs. One or more computers can then comprise the non-transitory computer storage medium and/or the download product.

While the invention has been illustrated and described in detail in the drawing and foregoing description, such illustration and description are to be considered illustrative or exemplary and not restrictive. The invention is not limited to the disclosed embodiments. Other variations to the disclosed embodiments can be understood and effected by those skilled in the art in practicing a claimed invention, from a study of the drawings, the disclosure, and the dependent claims.

CLAIMS

1. A system (100) for removing undesirable inferences from a machine learning model (102), the system comprising a search component (118) configured to:
5 receive a rejected explanation (110) of model output (106) provided by the machine learning model (102);
identify data samples to unlearn (120) by selecting training samples from training data (104) that were used to train the machine learning model (102), the selected training samples being associated with explanations that are similar to the
10 rejected explanation according to a calculated similarity measure; and
pass the data samples to unlearn (120) to a machine unlearning unit (122).
2. The system (100) of the claim 1, wherein the search component (118) is configured to identify the data samples to unlearn (120) by:
15 obtaining (401) a training explanation associated with a respective candidate training sample selected from the training data (104);
calculating (402) an explanation similarity measure between the rejected explanation and the training explanation;
comparing (403) the calculated explanation similarity measure to an explanation
20 similarity threshold; and
selecting (404) the training sample for inclusion in the data samples to unlearn (120) in response to the calculated explanation similarity measure satisfying the explanation similarity threshold.
3. The system of claim 2, wherein the search component (118) is further configured to calculate a sample similarity measure between the rejected sample and the training sample, to compare the calculated sample similarity measure to a sample similarity threshold, and to identify as data samples to unlearn (120) only those
25 selected training samples whose calculated sample similarity measures also satisfy the sample similarity threshold.
30
4. The system of claim 2 or 3, wherein the search component (118) is further configured to repeat (705) the obtaining (401), calculating (402), comparing

- 16 -

(403), and selecting (404) for one or more further training samples in the training data (104).

5 5. The system of any of claims 2-4, wherein the search component (118) is configured to calculate (402) the explanation similarity measure as either (i) a distance metric between the rejected explanation and the training explanation, the distance metric comprising one or more of a Euclidian distance, a City-Block distance, a Mahalanobis distance, a Huber distance, or an elastic distance measure such as Dynamic Time Warping or a Levenshtein distance, or (ii) a Jaccard similarity
10 coefficient.

 6. The system of any preceding claim, wherein the explanations are rule-based, and wherein the search component (118) is configured to identify the explanations that are associated with training samples as being similar to the rejected
15 explanation in response to one or more of (i) premises, (ii) structure, and (iii) decision values of the rule-based explanations being the same or similar.

 7. The system of any preceding claim, wherein the explanations comprise prototypes, and wherein the search component (118) is configured to identify the explanations that are associated with training samples as being similar to the rejected
20 explanation in response to the same prototype being produced.

 8. The system of any preceding claim further comprising the machine unlearning unit (122), wherein the machine unlearning unit is configured to remove
25 undesirable inferences from the machine learning model (102) by assigning, to the data samples to unlearn (120), weightings for a training loss function before retraining the machine learning model (102) using an updated set of the training data (104).

 9. The system of claim 8, wherein the weightings assigned to the data
30 samples to unlearn (120) are based on the similarity measures calculated for those data samples.

- 17 -

10. The system of claim 8 or 9, wherein the weightings are assigned to some but not all of a plurality of signals in the data samples to unlearn (120).

5 11. The system of any of claims 8-10, wherein the machine unlearning unit (122) is configured incrementally to change the weightings based on the Implicit Function Theorem applied to optimality conditions of a machine model training loss minimization problem.

10 12. The system of any of claims 1-7 further comprising the machine unlearning unit (122), wherein the machine unlearning unit is configured to remove undesirable inferences from the machine learning model (102) by retraining the machine learning model (102) using an updated set of the training data (104) from which the data samples to unlearn (120) have been removed.

15 13. The system of any of claims 1-7 further comprising the machine unlearning unit (122), wherein the machine unlearning unit is configured to alter the data samples to unlearn (120) by finding perturbations to the data samples (120) that result in an explanation derived from the perturbed data samples matching a non-rejected explanation.

20

14. A method for removing undesirable inferences from a machine learning model (102), the method comprising:

receiving (301) a rejected explanation (110) of model output (106) provided by the machine learning model (102);

25 identifying (302) data samples to unlearn (120) by selecting training samples from training data (104) that were used to train the machine learning model (102), the selected training samples being associated with explanations that are similar to the rejected explanation according to a calculated similarity measure; and

30 passing (303) the data samples to unlearn (120) to a machine unlearning unit (122).

15. A computer program product comprising instructions which, when executed by a computer, cause the computer to perform the method of claim 14.