Practical implementation of Lifecycle of Software Models

Brendan P. Bonner
School of Computing
Dublin City University
Dublin, Ireland
brendan.bonner2@mail.dcu.ie

***Abstract*—Artificial Intelligence is built on trustworthiness of the systems that provide the intelligence. The bedrock of these are models that have been trained in a particular manner to generate outcomes that replicate, or improve, the training inputs. To ensure that models used by AI practitioners are verifiable and traceable, we propose to investigate if we can validate models generated by machine learning systems, and generate a trace ability path between deployed models and the initial training, or pretrained models if transfer learned.**

***Index Terms*—Artificial Intelligence, Provenance, Explainability, Model Analysis**

I. Introduction

The topic covers the recording of CNN layer metrics during the training sequence. The hypothesis is to examine trust mechanisms for explaining AI, and providing an immutable record of the training sequences prior to a model being utilised or customised.

The practicum focuses on two approaches to this; recording and visualisation of the sequence of changes during the training of a model. During model fitting at configurable intervals, layer characteristics are measured and recorded, alongside mathematical verification, and stored. This can be visualised, supporting modern explainable AI measures, to verify model behaviour according to training, and can be trusted.

The proposal builds upon the attempts to provide additional metrics to support the establishment of the concept of fairness in AI. In comparison with human trust, which is established through verification and measurement of academic and professional credentials, Fairness in AI is an addendum to the model being examined, with attempts to quantify via measures such as Thiel scores for certain bias tests. This is further developed by IBMs AI360 Fairness toolkit paper which show how this is measured and corrected, but not being able to identify where that behaviour was acquired and why a model is trained in a certain way to develop these outcomes.

There is currently a gap in identifying if there is a provenance of trust when using deployed AI systems, and if sibling and inherited models retain these behaviours and if this can be visualised. While attempts to visualise the internals of decisions in the Simonyon and Bau papers, these focus on trying to visualise the pathways being activated in a 2 dimensional map. The dissection and visualisation papers provide a key how to measure and record this aspect of models, and our research will examine if there is a potential to securely record neural network snapshots.

The question being asked is to determine if we develop a novel mechanism and metric for representing the evolution of knowledge within a neural network during the training and development stages of creating a model. By analysing the state of the art in terms of AI fairness and methods of visualising the dissected layers common networks, the research will establish if creating an immutable chain of layer delta changes over time will provide an insight into the trustworthiness of the underlying model.

The second part of the research will ask if the information being stored can be visualised based on the elements of explainable AI to be able to identify which iteration and epoch of the training process is responsible for the manifestation of the behaviour. Ultimately, the outcome of the research should ask if providing verifiable additional data on how an artificial intelligence system evolves over time, if this can provide a foundation into further research towards comparisons of human trust and artificial intelligence trust.

A. Objectives

The overall objectives of the practicum are as follows:

- Discover a method to extract a unique signature from any Deep Neural Network software model.
- Create a local and global verification trail for
  - A Path of provenance to all ancestor models
  - A method to evaluate divergence of the child model from the parent
- Validation of Provenance via verification and illustration

II. Background

Models in CNNs are the primary reason for opacity of neural networks. Decision trees and algortihms have the benefit of being easily repeatable outside of the system. The complexity of even the simplest models makes this unviable. The introduction usually describes the background of the project with brief information on general knowledge of the subject. This sets the scene by stating the problem being tackled and what the aims of the project are.

A. *The Role of Models in CNNs*

B. *Identifying Differences in Models*

C. *Weights and Biases*

III. Method

Method should outline how the task/experiment was carried out, including rationale for any decisions made. Details of any equipment and subjects used should be also included. Basically, you should include enough information, so that the reader could duplicate as much of the experimental conditions or design details as possible.

A. *Reducing Model without Losing Character*

The first part of the delivery required that a model, in this case a CNN model in Keras, could be reduced to an information block that each layer contains a summary of the layer that can record any changes in the weights and biases. The structure of a model is an

1) input shape
2) series of dense layers
3) output layers

and within each of the layers, there are is a series of layers that, depending on type, will contain a series of tuples that have a single of multi-dimensional series of weights, plus a set of biases. it is within these layers that we will extract the information. As there is potentially a lot of information in the layer, we attempted to reduce this by extracting two identifying elements

TABLE I
Values generated from evaluated model

Mean	The average value of all layers across all dimension
StdDev	An indication of the deviation of values, to assist in seeing how the values are weighted
Histogram	A Histogram of the distribution of values across the layer. This will assist in giving a more informed distribution. As there are a large number of potential entries in the dense layers, from binary to minute changes in float values, the Histogram will be split between the values smallest non-zero granularity and the largest value across 10 bins.
Name.	Name of Layer
Angle of Data	This is a single value representing the weighting of a Hash of the Layer Values (weights and biases)

B. *Recording Changes*

As we now have a method of identifying the model in a broken down format, the next stage is to record and apply a signature to the model. There are a number of methods of ensuring that the model is recorded. The key part is to take the series of hashes on each layer, combined with the model shape, and generate a signature. The signature is what will be recorded as an immutable value that can be regenerated every time the **exact** same model is used. With this identification, we can create a list showing the parent and child of the model.

To ensure this is done correctly, we have two applications - local and global.

C. *Global Changes*

For global changes there will be a globa repository containing the signature and summary of published models.

TABLE II
Model Reference Storage

Model Identifier	This is the checksum generated by the model
Creation Date	This will be the timestamp that the model was first added to the repository
Parent	The model identification of the direct parent of the model.
Model Summary	A binary object that contains the items generated in the table above.

D. *Local Changes*

An additional benefit of the summarisation is to be able to view the model as it is being trained. This is a local filestore or database that contains a more regular record of the the summary changes. It is recommended that global model reporistory is only used for publishing models that require tracability, and not for regular changes.

In keeping with Git vocabulary, local changes are updated with the `model_add(model, [parent])` command, which takes a model and returns a signature and commits the information to a local repository.

The local repository stores the abstracted model information set that can be used for local comparisons. To commit the model to the public repository, the `model_add(model, [parent])` command is used, with this time the original model that already exists in the repository. The command will return an error if the parent does not exist.

E. *Comparing Models*

To ensure there is tracibility in between the models, if a parent model is defined, a *b-score* is generated with a degree of difference between the models. The score is based on two factors, the difference between the model structure, and the difference between the two primary parameters for each layer: *StdDev & Angle of Array*

F. *Collision Management*

In computer science, a collision or clash is a situation that occurs when two distinct pieces of data have the same hash value, checksum, fingerprint, or cryptographic digest.[1]

Due to the possible applications of hash functions in data management and computer security (in particular, cryptographic hash functions), collision avoidance has become a fundamental topic in computer science.

Collisions are unavoidable whenever members of a very large set (such as all possible person names, or all possible computer files) are mapped to a relatively short bit string. This is merely an instance of the pigeonhole principle.[1]

The impact of collisions depends on the application. When hash functions and fingerprints are used to identify similar data, such as homologous DNA sequences or similar audio *files, the functions are designed so as to maximize the probability of collision between distinct but similar data, using techniques like locality-sensitive hashing.[2] Checksums, on the other hand, are designed to minimize the probability of collisions between similar inputs, without regard for collisions between very different inputs.[3]*

IV. Results and Discussion

This section includes presentations of the data and results in graphical and/or tabular form. Statistical analysis of the variation in measurements/results can also be presented. The findings should be explicitly related to the stated objectives of the project. Details and tasks that that may distract the reader unnecessarily should be presented in Appendices.

This is the section where you have a chance to discuss the results and draw conclusions from them - i.e. where you show your ability to think analytically about your findings.

V. Conclusions and Recommendations

The conclusion of the practicum is there is currently no simple and efficient method of verifying the lifecycle of the development of the model. When models are released, either for production, competitions or uploaded to critical systems, they remain a blackbox. While developing a system for verifying the history of the model, it will allow us to both gain an insight into the development process, but also be able to check shared charactaristics between machine learning deep learning systems.

A. Recommendations

This section is usually left until the rest of the paper has been written. Conclusions are drawn in the context of the objectives of the project. They should be supported by data and results, and, if possible, compared with theory and data obtained by others in the literature (i.e. related published work). It is a chance to summarise what you have learnt from the project. Remember that human attention spans are extremely short, and the reader will appreciate a good summary, even if you feel that your conclusion is self-evident.

Recommendations are also extremely important, as they provide an opportunity to demonstrate the experience you have gained. The ability to self-assess one's work with a view to suggesting ways things could have been carried out differently is a valuable asset.

Recommendations can also include suggestions to how the work could be expanded or extended. This is often placed in a separate section entitled ”Future Work”. The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.