Tina Memo No. 2018-002
Internal.

Deep Learning: Some Criticism for Discussion.

Neil Thacker.

Last updated
9 / 5 / 2018

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

Deep Learning: Some Criticism for Discussion.

N. A. Thacker 9/5/18

Abstract

This document has been motivated by the suggestion that there might be opportunities for research in the area of "deep learning". I intend to approach the task of constructing a research strategy in three steps. First identify key issues which require attention, then identify existing solutions and finally to target applications for funding at things which have not been addressed.

This document only begins the first of these stages and is intended as the basis for a discussion. It concentrates on trying to answer the following questions:

What do people mean when they refer to "Deep Learning", in particular how is it different to a conventional Artificial Neural Network (ANN)?

What is optimised when we train a neural network to minimise cross-entropy?

What are the consequences for model selection?

What are the consequences for ANN methods when interpreted as "state-of-the-art"?

What are the consequences for scientific use?

As this is intended to be a stand-alone logical criticism, I present all issues as I see them, there are therefore no references. If any of this proves to be correct, I don't imagine that I will have been the first to have identified these problems. I am therefore currently working to identify any published literature from the last 20 years which can be used to either clarify, contradict or support these arguments and would value any input with this task.

Introduction

Deep learning is the latest idea to be adopted by large industrial research laboratories, and it has met with considerable support for funding, largely driven by the "academic impact" agenda. We have observed that it also now poses a regular barrier to the publication of more thoughtful approaches to data analysis, with many reviewers demanding that anything new should be compared to it. It has connections with areas such as data mining and big data; often applied with some success (i.e. generating state of the art performance) to tasks such as image (for example face) recognition. Understanding if the methods have any value outside of the existing demonstrations requires a certain level of critical scrutiny. I will try to explain below how the quantitative assessments of uncertainty and statistical rigour needed for scientific tasks may not be met and that these issues have a long standing history.

If we wish to apply "deep learning" in order to achieve state of the art performance for some task (or even in order to conduct a shoot-out to satisfy a reviewer), then we must define what it is, and in what way it seems to generate performance benefits. This is quite a difficult question as many papers have been badged as "deep learning" and there are various claims made about how to use them. The original use of the term (Dechter 1986) was not as we see the idea today, but for adding second order terms into calculations. Later Le Cun (1989) adopted the description for a network with 5 layers. Although it has always been associated with some form of ANN. For purposes of discussion, we will show below how in general a trained ANN should be expected to estimate something relating to a probability of classification (c) given some input vector (X) following a training process, i.e. $P(c|X)$.

Deep learning has become synonymous with large scale tasks, for example training to recognise very large numbers of objects in very large numbers (millions) of images. The "deep" part of the name comes from the use of ANN's with very large numbers (even hundreds) of hidden layers, rather than any esoteric algorithmic capability. Additional algorithmic suggestions have been made which improve classification performance, but these extensions don't always have a "deep learning" scale of network or data-set. The two "deep learning" papers at a recent medical imaging conference had only three hidden layers and were trained with conventional algorithms on no more than a thousand training patterns, i.e. the work was largely indistinguishable from neural network research from the 1990's. When questioned about this the authors said they used the deep learning tool-kits. These tool-kits are in the form of a user interface which supports rapid prototyping of black-box classification systems, which allow people to try out some of the latest approaches to architecture construction and training.

A point to note here is that when people call something "deep learning" this may actually be nothing more than an ANN, and not be able to exploit the benefits of "deep learning" at all (more on this below), whatever we might

think they are. Most of conventional pattern recognition can be mapped into an ANN in one way or another and it is always possible to conceive of new training strategies or computational tricks which might be shoe-horned into a feed forward network architecture. In the absence of a meaningful definition, the large scope available for ANN's and Deep Learning, make any specific criticism difficult and at the same time any claim of superiority for one over the other potentially vacuous. For this reason I will try to focus here on fundamental problems with the basic ideas. I will not be addressing limitations of feed-forward computational systems or the distinction between causality and correlation, as these are long established. Instead I will discuss the kinds of tasks which proponents might suggest a pattern recognition (PR) system to be capable of. In particular; Are these systems really providing Bayes optimal solutions, and can we use them as the basis for quantitative scientific studies?

In image processing the concept of deep learning is seen as a hierarchical approach to the representation and recognition of image content. This is a very challenging problem which involves highly complex data variations in a large dimensional pattern space. With deep learning image features are extracted (perhaps following some pre-filtering) by initial processing layers, via computations very similar to more conventional feature detectors (e.g. edge detectors), whilst deeper layers compute ever increasing abstract levels of representation. The final layers perform the conventional overall mapping from variables to labels required for classification. Other identifiable contributions in this area involve the use of multiple learning constraints, which influence the solutions in a way which optimises several criteria, for example classification, and representation or the use of adversarial training. This integrated approach has also been taken in the area of speech recognition, and it is claimed that all current commercial systems now work in this way.

Linking an entire system together, as a continuous differentiable function, allows an overall roots and branches style of optimisation using ANN training, which directly addresses the issues of "optimally" for each level. For me, this is the core idea behind the success of deep learning.

An ANN has a large number of parameters. Unfortunately, even if the computational form of the network will support a meaningful computational construct, identifying the set of parameters corresponding to this construct is like searching for the proverbial needle in a haystack. The combination of non-linear calculation and architectural complexity results in many local optima, which can only be located by iterative optimisation. Starting from random initialisations of weights, we are much more likely to send the optimisation into a local minimum than the ideal global one. Conventional downhill training algorithms will use the available degrees of freedom and functional complexity to approximate any logical answer in ways which are unrecognisable upon inspection of the patterns of connective weights. This approximation may be finely balanced (like the subtraction of two very large but similar numbers), and only generate useful outputs around the areas close to the training data sets. This makes outputs generated for patterns well away from training examples effectively random. In neural network parlance, they are unlikely to generalise well.

When considering aspects of generalisation we must be careful not to get too carried away with our expectations. There are properties of data which can reasonably be expected to be discovered from input data. The simplest of these is variations due to input noise. More usefully, there are constrained low parameter variations, such as rotations or scalings, which a well designed statistical training algorithm may be able to discover. However, a training algorithm will be unable to infer all images of objects consistent with a label (e.g. "chair"), if there is no logically justifiable mathematical process which can map input image data into a consistent form. Generalisation over *random* discontinuous groups can only be achieved by linking together the disparate parts of a pattern space using other sources of information (i.e. the group labels)[1]. When using the term generalisation (below) I am therefore referring to the former, mathematically realisable category, rather than the latter.

For many years pattern recognition algorithms such as support vector machines and boosted decision trees were considered state-of-the art, but not ANNs. This was because the ANNs were difficult to train well, even though they were better in some respects from a theoretical point of view. Papers were even rejected from conferences for using them (then as now, this might be considered cheap reviewing practice). It is now claimed (despite all logical considerations to the contrary), that deep learning networks can not only be trained easily on large quantities of data[2], but also appear to be able to generalise well to new data. We can be generous here and put aside the obvious explanations of publication bias or poorly designed experiments, and just take these observations at face value. Even so, any modern paper now described as "Deep Learning" which is no more than an 1990's style ANN trained using a deep learning toolkit must have the same old problems. We can really only expect the much publicised benefits of deep-learning if we have an integrated system which combines multiple training constraints and very large training data sets. In other words, Deep Learning might be considered state of-the art, but conventional ANN's should not be. Unless, that is, we wish to ignore two decades of published work.

[1] Even if we were to ambitiously use the word "deep" to imply a sophisticated rationalisation, this is impossible for truly random label data.

[2] Though still not enough to estimate all of the free ANN parameters.

Some Criticisms of ANNs in General and Deep Learning Specifically.

It is my intention to criticise deep learning and ANN's in a very general way. To do this I must start by defining what these systems are expected to compute. Then I can explain what the issues are which limit the application of these calculations in real world. This will help shape the research issues which need to be addressed.

ANN's Estimate Conditional Probabilities

Artificial neural networks can be thought of as a high dimension interpolation function which maps an input vector $\mathbf{x_i}$ to an output $\mathbf{o}(\mathbf{x_i})$. Many supervised training algorithms are based upon optimising the difference between this output and some target t_i for N data samples. For feed forward networks this can be achieved by minimising a cost function such as the chi-square

$$\chi^2 \;=\; \sum_i^N \; |t_i \;-\; o(\mathbf{x_i})|^2$$

or a "cross-entropy" function of the form

$$E \;=\; \sum_i^N \; t_i \; log(o(\mathbf{x_i})) \;+\; (1-t_i) \; log(1-o(\mathbf{x_i})) \qquad\qquad -(1)$$

In a deep learning context, we might use the chi-square to tune up a regenerative representation (counter-propagation), whilst cross-entropy can be used to train the final classification process. The relationship of a chi-square function to Likelihood is well known, as are its problems with regard to model selection, but what about equation (1)? If you already know the answer to this you can skip to the last paragraph, but it is worth setting forth the mathematical argument so that we know exactly where our conclusions are coming from.

Generally the output function will be constrained by the network architecture to be smoothly varying over some volume of $\mathbf{x}$. Over this volume the output will be effectively constant o (or at least linear, see below) for some subset of data N_s [3]. A partial contribution to the overall cost function for this subset can be written as

$$E_s \;=\; \sum_i^{N_s} t_i \; log(o) \;+\; (1-t_i) \; log(1-o) \qquad\qquad (2)$$

We wish to understand how o is related to the original training data t_i and we can do this by determining the choice of o which minimises E_s.

$$\frac{\partial E_s}{\partial o} \;=\; \sum_i^{N_s} \frac{t_i}{o} \;-\; \frac{(1-t_i)}{(1-o)}$$

Setting, $\sum_i^{N_s} t_i \;=\; N_c$ i.e. the number of times the target class label was 1 in the selected volume, we have

$$\frac{\partial E_s}{\partial o} \;=\; \frac{N_c}{o} \;-\; \frac{(N_s - N_c)}{(1-o)}$$

$$=\; \frac{N_c \;-\; N_s o}{o(1-o)}$$

Assuming that $o(1-o) \;\neq\; 0$, this expression is zero (E_s is a minimum) when

$$o = N_c/N_s$$

so that the overall cost function is minimised when, in all locations in the space, the output from the network is the local fraction of the number of positive target labels; in other words the **conditional probability** $P(c|\mathbf{x_i})$. Interestingly, the same analysis can be made of a χ^2 cost function and the same conclusion is reached[4]. The binary

[3] Equally you can make this true in the limit of infinite data.

[4] Note however, this also illustrates that the appropriate selection of a cost function must be based on something other than the required output approximation. I would recommend following conventional Likelihood rules and using the noise properties of the training data.

targets t_i can also be relaxed to be real or "soft" labels, in accordance with our definition for N_c. The proof can also be generalised for a linearly varying output over the volume N_s with average o.

By minimising (1), assuming that the solution is reachable with the available degrees of freedom of the network, we are therefore ensuring

$$o(\mathbf{x_i}) \approx P(c|\mathbf{x_i}) \qquad -(3)$$

and if the output is used to attribute class assignment, it can attain Bayes optimal performance over a set of test data, provided the prior probabilities of classes are equivalent in both training and test data sets, and at all locations in the data space[5].

Finally, as $P(c|\mathbf{x_i}) \propto P(\mathbf{x_i}|c)$ we are also minimising the log likelihood of obtaining the data. We may wish to stay with an interpretation of MAP estimation, but the constant of proportionality, being fixed here, plays no role in controlling model complexity and so does not help to solve the key problem of model selection.

This assessment of feed forward training therefore embodies two key problems associated with neural network research. The first is;

1) The use of Likelihood for training does not make best use of data with regard to generalisation.

The second is that;

2) The Bayesian nature of training results in sub-optimal performance (non-Bayes optimal) when test data is not statistically equivalent to the training data in all parts of the data space.

As the specific construction of training and test data (not to mention that of the data seen during application) are never discussed, it is difficult to believe that recent "advances" in network design and training can be accepted as good solutions, even when taken on their own terms. We should also note here that simply having a lot of data will not remedy this situation, as there is always something fundamentally arbitrary about the selection of data-sets.

Hard Labelled Segmentations are Bad

For several years now the medical imaging community has run image segmentation shoot-out competitions at conferences. Given a large number of manually annotated images[6], PR researchers submit their algorithms, which attempt to replicate human decisions, e.g. grey/white matter segmentations in MR. This evaluation is performed by an independent group on a separate evaluation data set, never seen by the PR researchers. The wining methods are hailed "state-of-the-art".

It has been observed that when training a pattern recognition system, if there is a large imbalance between target and non-target categories, the system may completely fail to identify targets when labelling outputs according to the maximum output decision (hard label). The first time I can remember this being reported in the literature it was referred to as 'the signpost effect' because signpost detection systems trained on whole images would never conclude a signpost was the most likely interpretation of any image location.

Given the above analysis, it is easy to understand this problem. If $P(c|X) < P(\bar{c}|X)$ for all X, then we will never detect c. As these terms are proportional to the priors $P(c)$ and $P(\bar{c})$, this is a direct consequence of a prior imbalance in Bayes theorem, and completely expected. In order to get around this issue experts will change the prior balance of groups within training data. This can also be achieved by using weights on training data, and a common practice is to use weights to achieve the same result as having equal class priors. For any specific data-set, with a specific prior construction, this choice is however not only arbitrary but also inappropriate. It could be argued that given a set of alternative PR systems, all equally valid and Bayesian in nature, the researcher who will win any shoot-out competition will be the one who guesses the appropriate prior balance for the subsequent evaluation data. Also, if test and training data do not have the same demographic construction (at every location in data space), then by definition the best possible system must be sub-optimally trained[7]. In any of these cases, the latest 'state-of-the-art' method achieved its best performance status by accident and not design.

In addition, the practical application users want these segmentation algorithms for is generally volumetric or counting. Yet there is a difference between an attempt to replicate, on average, human subjectivity (using Bayesian labelling) and a quantitative assessment of specific regions in an image. It is easy to set up toy distributions and apply Bayes Theorem in order to show this. In particular, the quantity we are trying to estimate ($Q(c)$) is proportional to the prior and we can make a direct analogy to Expectation Maximisation (EM). EM is a coupled iterative scheme which converges on valid priors and data distributions. The priors are used during distribution

[5] Whilst this sounds impressive it is also quite a significant restriction on optimal use.

[6] Not large by big data or deep learning standards.

[7] The differences between alternative approaches may also be statistically insignificant if properly tested, though this is rarely checked.

estimation and visa-versa. We can simplify things here by assuming that the distributions ($P(X|c)$) are already known and do not require further update, but we must still iterate to estimate $Q(c)$. Specifically estimation (at time $t+1$) takes the form;

$$Q_{t+1}(c) \;=\; \sum_{i}^{N} P_t(c|X_i) \;=\; \sum_{i}^{N} \frac{P(X_i|c), P_t(c)}{P(X_i)} \qquad -(4)$$

with

$$P_t(c) \;=\; \frac{Q_t(c)}{Q_t(c) + Q_t(\bar{c})} \qquad -(5)$$

$Q_\infty(c)$ is the unbiased (Likelihood) estimate of quantity, which will be correct even when $P(c|X) < P(\bar{c}|X)$ for all X. Whereas, simply counting the number of class labels generated by an ANN will not generate $Q_\infty(c)$ (e.g. the signpost effect).

But our ANN's will give us conditional probabilities, so what happens if we use those in equation (4)? Even assuming that $P(X_i|c)$ is static, a Bayes approach, such as an ANN trained on cross-entropy, will generate $P'(c|X)$ estimated using *implicit priors*[8] ($P'(c)$) defined by training data, regardless of the actual data composition. It must then give biased estimations on specific cases, as

$$Q_\infty(c) \;\neq\; \sum_{i}^{N} \frac{P(X_i|c), P'(c)}{P'(X_i)} \qquad -(6)$$

for any case where $P'(c) \neq P(c)$. This is whenever wish to count something because we don't already know the answer! Put bluntly, the theories used to justify machine learning are enough to prove in a very general way that PR systems with fixed decision boundaries (established on a single data-set) cannot make valid estimates of quantity. I should point out here the link between this issue and the more general criticism of ANN's (or PR in general), that outputs are only meaningful in a stable world. This is embodied by the more stringent mathematical requirement that priors **never** change, which has to be considered less realistic than assuming, as here, that the priors can change but at least $P(X_i|c)$ is fixed.

The situation gets even worse than this. The above analysis demonstrates not only that any specific $P'(c|X_i)$ computed using a fixed implicit $P'(c)$ is not the optimal one for the data, but also the optimal value would change dependant upon data context ($P(c)$'s), i.e. it is not unique to X_i. The meaningful value arising from this analysis, and usable as a scientific summary, is $Q_\infty(c)$ and not the ANN output, $P'(c|X_i)$. Yet it is invariably the accuracy of $P'(c|X_i)$ and not $Q(c)$ which always gets evaluated.

This discussion therefore embodies two further problems with pattern recognition.

3) Evaluations to identify state of the art approaches do not address the known theoretical limitations of Bayesian methods.

4) Applications are evaluated on the wrong criteria and therefore incorrectly promote methods.

How important these issues are to you will depend upon the domain. Engineers might say these criticisms are points of detail which cannot detract from an observation that these systems "seem to work". However, we might also expect that scientists will object about any quantities estimated in this way being compared to a theory. Clinicians should also worry about loss of quantitative meaning, statistical accuracy and detection sensitivity.

Conventional Science and Medicine has Specific Challenges.

Computers continue to enjoy rapid advances in both speed and storage capacity. However, it has been the development of graphics cards and the support of large scale fine grain parallelism which has given a boost to ANN research. This has given rise to the popular topic of deep learning for big data. Deep learning uses large artificial neural networks with huge numbers of parameters, and both it and big data focus on extremely large datasets. It is believed that given sufficient numbers of examples, deep learning can learn the salient properties of a data-set, leading to data driven (bottom-up) solutions. Strong industrial promotion and their "state-of-the-art" position, combined with relatively small learning curve (black-box approach) for use, make these approaches highly popular.

In recent years we have investigated the problem of application of machine learning algorithms to biological, clinical and more general scientific tasks. We have found that uncertainty estimation (rather than transparency, see below) is key to using any estimated quantity, but that this is unavailable in the popular PR systems. They do not even

[8] By this I mean that even though the ANN may not explicitly use Bayes theorem we know that the required $P(c|X)$ must change if $P(c)$ or equivalently $Q(c)$ changes, in accordance with Bayes theorem.

signal when an input has been provided which is unlike any of the training examples. This is almost certainly one of the reasons why it is said to be difficult to build robust engineering solutions using ANNs (science and clinical applications are likely to be even more difficult). The conventional wisdom seems to be that any form of sophisticated uncertainty calculation, such as estimation errors, is either impossible or unnecessary. We have therefore sought to demonstrate that this view is wrong by providing working counter examples (Appendix C). Several papers are now in print which demonstrate how knowledge of both uncertainty due to the **finite amounts of training data**, and also that due to **input noise**, are needed to support scientific interpretation of data. Other authors have also identified the need for what they refer to as epistemic and aleatoric uncertainties (Appendix D).

Key to this work has been the observation that generally for science we need to apply PR to quite small datasets. Data may be complex but the numbers of examples are restricted by clinical or medical methodology, or financial realities. Making best use of relatively small sample sizes therefore becomes an important issue.

I emphasise that there are two sources of noise here (shown in bold above) and not just one. In more conventional statistical analyses used in science the sampling variations are absent, so that consideration of input noise using techniques such as error propagation or minimum variance bound (MVB) are enough. However, for pattern recognition systems we need to accept that a sample of one in a localised region ($N_s = 1$, see above) is insufficient to infer a label with 100% confidence. Under these circumstances it is clear that $o = N_c/N_s$ is not the conditional probability we need to make reliable decisions ($P(c|\mathbf{X}, training\ data) \neq P(c|\mathbf{X})$)[9].

Two more restrictions on the use of PR in science and medicine therefore follow;

5) Conventional science has small training datasets and this would inhibit application of highly parameterised systems.

6) Scientific interpretation of output requires a quantitative appreciation of uncertainty, which is missing in the field.

Although big data will throw up new and exciting problems suitable for deep learning, I believe that the long standing scientific problems associated with small data are not going to go away. Overdue emphasis on big data might well have very limited applicability and hinder any search for solutions to long standing problems. There is also a very real risk that excessive hype could sweep away resources for genuinely useful research when practical systems fail to deliver on outlandish claims [10].

Data Representation.

When considering problems in computer vision, the difficulties of generic image interpretation follow from the huge possible number of images which can be constructed from the same visual stimulus. Effects such as illumination, object orientation, grey level and object scale, not to mention occlusion and shadow, combine to multiply the possible combinations of object appearance exponentially. This is before we even begin to consider the often stochastic behaviour of textures. For many years researchers sought to try to remove these effects via the construction of appropriate data "representations", which computed explicit invariant descriptors. Prior to deep learning, conventional computer vision approaches would generally concentrate on reducing this variation, for example pre-processing, extracting features and re-representing the data to eliminate unwanted variation (exploit invariances). When performing classification, $P(c|X)$ is replaced with $P(c|f(X))$, where $f(X)$ is chosen to be invariant to the nuisance parameters. We can think the input data as a function $X(z)$ of the nuisance parameters z, and seek the $f(X(z))$ which makes $f(X(z')) = f(X(z))$. The pattern space associated with f is smaller in volume than X. By reducing the complexity of the data space we make the classification problem easier to learn. We also boost the sampling rates within the pattern space and so reduce the problems associated with low sampling frequencies described above. Ideally this should be done while maximising discrimination of c and we can consider this in terms of information loss. For example a "complete" representation supports the reconstruction of X up to the limits of invariance.

Although this topic was plagued with much repetition and not much genuine progress, it seems obvious that ignoring this issue and simply putting grey level image patches directly into a network architecture will generate more difficulty when learning to map data than would be the case with a good representation. Without learning algorithms which optimise generalisation, we cannot believe, even if an invariant representation is in principle computable, that the weights would be chosen appropriately to extract it. It has been noted previously that ANN architectures are not ideally suited for the incorporation of such prior knowledge, presumably because the required

[9] For a two class problem, using the same notation as above, the correct probability would appear to be $(N_c + 1/2)/(N_s + 1)$. We have to ask whether the success of deep learning is nothing more than reducing this problem by making $N_s >> 1$ over a larger volume of the pattern space.

[10] We should remember that ANN's are now in their third boom-bust cycle since 1965.

calculations are not efficient when mapped onto a feed-forward architecture. If we knew what was worth computing ($f(X)$), why use a training algorithm to tediously identify this calculation when we could of course just perform it in advance,(as a form of pre-filtering)? But this is not my point, if we think that our ANN's and training algorithms are capable of finding appropriate solutions we should see it as a problem if known mathematical approaches cannot be discovered. This is a particular issue with pre-filters based upon convolutions which are certainly computable exactly on an ANN.

On an associated issue, it has long been known both from psycho-physics and theory that the most informative parts of any image are at edges and object boundaries. For pattern recognition to work effectively the input data should have stationarity (i.e. the data density distribution must be compact). Yet a computational structure based on convolutions will have the problem that image patches at the boundaries of objects will be composed from both foreground and background objects, making the specific image patch unpredictable, except in the most trivial of circumstances. Additional problems are found when trying to predict depth close to edge boundaries (see Appendix D).

Putting these two observations together, we have another two criticisms of deep learning.

7) It is dumb to feed raw pixels into a pattern recognition system without first doing something to eliminate irrelevant data variation.

8) We need to believe that a large network is using its structure to compute invariants, but (if we cannot specify what these are) why should we believe this is mathematically possible, or that optimising Likelihood should find it.

In making these criticisms I am not suggesting that there are not pre-processing or training tricks which can be used to try to mitigate these problems (there are), what I am saying is that the general idea of feeding arbitrary data with complex behaviours into an ANN is unlikely to be the best approach. This is therefore a criticism of black boxes, and there are others.

Limitations of Black Box Methodologies.

Neural network approaches have always had their detractors, particularly from those who have worked in a more traditional way to understand aspects of human perception and statistical data processing. If our next generation of researchers in AI and PR are to cut their teeth on deep learning, with the expectation that all that is needed to generate useful academic output is to obtain a performance figure for the latest standard data-set, then who will be developing these methods ten years from now? Perhaps this is why UK funding bodies recently decided to award a large number of PhD's in deep learning to particle physics rather than the computer science community!

There are many possible areas in particle physics which may seem suitable for a deep learning solution. ANN's are particularly good at learning an inverse mapping, when a well defined forward mapping is known. Such examples include labelling the charge on jets or initial charge particle grouping. Any inadequacies (efficiencies) in performance can always be assessed via Monte-Carlo simulation, the staple tool of the field. However, it is also worth looking at more complex problems in order to try to identify limitations and also if the tools of deep learning, rather than just ANN's, would be of value. Scientific analysis has different requirements to engineering, and it would be wrong to think that any classification problem in science can be usefully solved by deep learning just because there are impressive engineering examples. We consider the problem of event recognition in the Appendix B below, not because it is easy but (as someone else once said) because it is hard and can serve to illustrate this point.

One long standing criticism is this; Although being able to construct state of the art performing systems using off the shelf software tools is useful for engineering;

9) Nobody really learns anything by building a black box system, other than a data-set specific performance figure (which is generally 85 %).

Although such work may have potential for industry, it is not of high academic value. It is for this reason that neural network research was effectively banned from many conferences in the 1990's, under the label of "mindless application of ANN's to X". Even today work, which investigates specific fundamental problems and solves them should be considered more valuable for the future of AI.

A further limitation of taking a "black-box" approach is that ANN's have repeatedly been shown to adopt solutions to problems which exploit undesirable properties of training data. For example face recognition systems trained on the bottom left hand corner of an image will produce quite good performance levels, even though they are not 'seeing' the face. One explanation for this is that specific patterns of noise can give information on the camera, and different cameras were used for different people. Another is bad data specification, for example images being

acquired with backgrounds (e.g. office, desk or chair) specific to the individual. So who knows how results are biased when trying to evaluate individual algorithms, presumably an algorithm which makes inappropriate use of the image to recognise a face will do better than one which correctly uses the region of the face alone. "It seems to work" is not an adequate response to this criticism. As a consequence of this, those attempting to use big data and deep learning will need to consider data quality, uniformity and nuisance issues. These problems generally do not exist in small data problems, where good data quality and consistency will be a key aspect of study design from the outset [11].

There are a multitude of other possible pitfalls when evaluating a training algorithm, including experimenter effects, publication bias and missing statistical significances (no one ever provides an uncertainty estimate for an algorithm ranking). So a good performance is not necessarily going to provide a gold plated conclusion. To counter this criticism,

10) We need to be able to understand how networks have solved a problem and why this solution would be expected to be valid on other data.

This issue has also been referred to as "transparency". However, it is probably unrealistic to ever expect a large ANN to be amenable to interpretive analysis. If the reason we want this property is to lend confidence to outputs then it might be better to consider this as another call for uncertainty assessment. Armed with honest estimates of uncertainty (and assurances that the pattern has been seen before) we might be quite satisfied that the output can be trusted, even though we do not know precisely how it was computed. In this case, point 10 is addressed by point 6.

Summary

Contrary to the views of some in our University, I do not believe that our role in scientific or medical research should be seen as training others to apply popular software tools to data, as I do not believe that these tools are ready for application to general problems. I have identified ten problems associated with PR in general, and the use of deep learning more specifically, to support this view. If we now look at these criticisms we can form some more general conclusions by understanding how they are related. In particular, points 1, 3 , 5, 8 and 10 are partly consequences of the long standing problem of model selection and optimal generalisation. The others, 2, 4, 6, 7 and 9 all relate to specific methodological failings. I would therefore argue that (almost 30 years after this was acknowledged by the ANN community) the most important question which still hangs over this area is that of optimal generalisation. **This is why it is a problem that training algorithms are equivalent to using Likelihood.**

The issue of neural network generalisation has been, and continues to be, an important research topic. If more recent training regimes, such as adversarial training, have merit then presumably it must be because they address this issue in some way. Indeed, such training methods can be interpreted as a form of regularisation, which constrains the best computational solutions and reduces the effective number of free parameters. Issues such as invariance are now solved not by pre-processing the input data but by demanding invariance of the classification output. In a similar notation to above we have $D(c|X(z)) = D(c|X(z'))$, where D is the equivalent network mapping function. This in turn may, or may not, configure the early layers to do something equivalent to pre-processing. I leave it to the reader to form their own opinions based upon personal experience with gradient based training algorithms.

For small architectures applied to simple mapping tasks, it is relatively meaningful to try to identify a best generalising architecture. However, for larger structures, it may be unrealistic to expect that simply training on vast quantities of data will address this issue. Adversarial training may discover $P(c|f(X))$, but as this would be the result of a regularised iterative optimisation we have to accept approximations which do not preserve all of the useful information (i.e. not complete).

There may be a certain element of wishful thinking involved in suggesting that deep learning systems can generate high level descriptions of data with a meaningful treatment of invariances (i.e. the mathematical functions discover some valid principle). It is a claim which should require solid evidence to back up. Others may argue that really we don't know what these systems are doing. Certainly everyone who comes to deep learning expects to be able to use the available tool-kits as a black box and does not want to think about the inner workings.

To be of real value, any future research project will need to address all of this whilst taking appropriate account of the other methodological issues.

Key research topics include;

[11] Historically, an author will have had real difficulty publishing a clinical imaging study when the control and response data were acquired on a random mixture of machines.

- Training algorithms which optimise generalisation (I can envisage at least two).
- A network architecture which gives quantitatively valid estimates of quantity (we already have one).
- Network architectures which explicitly extract known invariants. (I consider adversarial training only to be a partial solution.)
- Methods for the assessment of (scientifically useful) uncertainty on computed outputs (we have demonstrated this with LPM's).
- High impact applications in science and medicine of statistical machine learning, which make better use of data than would a T-test or ANOVA. (We will have far fewer applications to choose from if we stop doing fundamental imaging physics).

All of the above should operate on minimal sized architectures and small datasets. As such this would be a form of "deep learning" research which actually has very little to do with conventional deep learning. Appendix A contains a set of specific approaches which may address some of these areas.

Appendix A: Some Ideas and Challenges

Unsatisfactory Hacks

Adopt any training strategy which might improve generalisation and evaluate using an intermediate (validation) data set.

- Stop training early

  - there is no theory to determine when best to do this.
- Use previous network solutions to initialise new ones.

  - pre-supposes we want to solve more than one problem of a "similar" nature. - no quantitative theoretical understanding.
- Include a regulariser term

  - needs quantitative theory to say what is best
- Change the balance of example:counter example in order to fix the sign-post problem

  - no associated theory to determine how best to do this.

Good Ideas

- Increase the amount of data to improve generalisation by adding typical noise

  - training takes longer.
- Increase the amount of data by rotation or scaling.

  - training takes longer, no reduction in problem complexity.
- Train to constrain outputs to be equal for patterns which should have invariance.

  - does not actually guarantee invariance, but a useful smoothness constraint.
- Unsupervised trained reconstruction of input data for dimensional reduction

  - then all input data is used when it might not be needed, i.e. discrimination is not the same as compression. The network will require more data to train without apparent performance benefit.

The Bishop Variants

Things which are mentioned in passing in Bishop's book.

- Estimate missing variance using noisy data to correct the cost function to optimise generalisation.
  - then we either can't use back-prop, or must train on a lot of extra data.
- Compute a modification to the cost function derivative to incorporate generalisation directly into back-prop.
  - can this be done? and may not even work before reaching the minimum.
- Modify the final stages of a network to directly implement Bayes theorem, compute explicit density distributions for each class. This also fixes the problems with quantity estimation.
  - Learning densities requires much more data than learning decision boundaries.
- Estimate **statistical** uncertainties (propegated error) on outputs so that they can be used in science.
  - Very complex and may be numerically unstable.
  - May not even work with decision boundaries (Probabilities must be honest and they are not).

Other Ideas

- Correct outputs for **systematic** uncertainties (sampling error) so that they can be used in science.
  - Requires a local estimate of sample density.
- Compute invariant representation to eliminate rotation and scaling
  - cannot use CNN's without modification.
- Train on sets of sets to help determine variations in data densities. - Will require a significantly different training regime.
- Use RBF style layers to allow improved weight initialisation via leader clustering space mapping.
  - Network architecture must support RBF nodes. Less efficient than mapping only decision boundaries.
- Use the theory of Wavelets to construct invert-able representations and optimise the mother wavelet.
  - Wavelets are not easily rotate-able.
- Use a genetic programming approach over a set of network solutions to search for better minima.
  - requires a bolt and build network and far more training effort than back-prop.
- Use sensitivity to randomise training labels to select generalising architecture.
  - almost the same as testing generalisation on a separate data set except may give localised pattern information.

Informative Experiments

- Completely randomise target labels and see if generalisation performance is affected
  - if it is not then we cannot argue anything "deep" was ever extracted.
- Set up a network with with the mathematical capability needed to model a specific invariance property and see if training (either Likelihood or generalisation based) will find it.
  - if it cannot then we cannot expect deep learning to be finding good solutions.
- Generate artificial sample data from a known model where some network weights are set to zero and see if we can identify them.
  - if we cannot then generalisation training is not working.
- Generate multiple alternative artificial samples of data and see if adjustments to output probabilities for the uncertainties of sample statistics improve ROC performance.
  - if it does then this implies that all ANN's should have outputs adjusted depending upon local sample quantity.

- Evaluate the uncertainties in an output using a "what if?" approach (e.g. noise, rotation/reflection).

  - this may provide a sufficient assessment of uncertainty to support robust decision making.

- We can also use "what if?" approaches to test data and see how generalisation performance is affected.

  - if our generalisation assessment is significantly reduced then the original test data-set was not sufficient to reliably evaluate generalisation.

- Train up multiple ANNs on boot-strap samples and see how outputs vary.

  - this will tell us if the outputs are stable and if there are any mathematical problems (such as unrealistic extrapolations).

- Use regenerative input prediction to confirm knowledge of current input pattern.

  - helps with the problem of nuisance input variables/ inappropriate solutions.

- If previously trained regenerative ANN architectures are useful, they should be able to regenerate unseen data.

  - this could be used to help pick previous ANN solutions as the starting point for training.

Appendix B: Particle Physics

This abstract is a summary of a discussion between myself and two members of the Lancaster University Particle Physics group.

Conventional analysis of events operates by taking measured track descriptions which can be interpreted as consistent with the kinematics of a particle interaction or decay. In a conventional analysis the various measurements (momentum, energy, charge, position) are fitted to a vertex on the basis of the estimated measurement errors. Once a good reconstruction has been found decisions are made to select a subset of events based upon these values and other physically meaningful derived variables, (transverse momentum , impact parameter etc.) . A system of "cuts" is devised and evaluated on Monte-Carlo simulation data, in order to avoid issues such as invariant mass sculpting, which would bias subsequent estimation of mass and cross-section.

Let us consider the idea of training a deep learning system to recognise rare particle physics events, for purposes of finding new physics. This isn't an arbitrary choice, as the Particle Physics community have already run a competition to see if pattern recognition experts could provide useful ways of conducting Higgs searches, and we can if we wish generate very large amounts of Monte-Carlo simulation data in order to meet the requirements of deep learning.

Our first question might be; where do we start? Do we want to train a deep learning system to perform track reconstruction from the raw data? This seems a little excessive, but would be in keeping with image recognition applications. There might be some difficulties getting pre-existing tool-kits to take the data as it isn't an image. If we do this can we get the track parameter covariances? One objection here is that the event reconstruction software in particle physics already works very well, so there may be little or no need to attempt this. Though equally we could argue that edge detectors in computer vision also already worked well too.

There are many other problems in particle physics which might be tackled via a machine learning approach but have existing solutions. Here the correct mathematical approach is well defined so that we know at least which variables are required and how they should be used. For example, assuming we start from predefined track reconstructions, what do we do with the track parameter covariances? We can assume that most particle physicists are not going to be happy ignoring this information, but machine learning systems are not designed to make use of measurement errors as part of the input (computer scientists don't *do* errors). We could however just feed in the parameter covariances and hope that by training with enough data the deep learning architecture might figure out how to use it. In effect we would be expecting the free parameters in the ANN to configure themselves to achieve the equivalent of a vertex fit [12]. Why do this if it were unnecessary, instead given the mathematical and programming skills, could we re-design the input layers to make appropriate use of parameter error covariances?

Continuing with the example, our machine learning system would now be expected to compare kinematic distribution of tracks to the training data. But is the raw input measurements the best way to do this? The mathematics of particle decay is best described in the rest frame not the laboratory frame. We should at least be applying an appropriate Lorentz transformation. Once again, should we put this into the system by design or just let deep

[12] A vertex fit is an iterative optimisation procedure with a specific, and complex, optimisation function. It might be argued that it is not going to simply map onto a feed-forward network architecture.

learning use adversarial learning to rediscover this? The Higgs search competition provided some variables which had already been identified as important. This reduced the complexity of the problem but also reduced the chance that pattern recognition would do anything interesting. Certainly, it is not deep learning.

Once in the rest frame, should we just compare the kinematics here with example decay distributions. The theory of particle decay tells us that there are preferred co-ordinate systems, for example defined with respect to momentum transfer, which can be used to rotate the event in a way that generates predictable angular decay distributions. Do we wish to let deep learning figure this out from examples? Would it even be able to? Remember, it may need to propagate the error covariances too. We have statistical theories which are used currently in the course of data analysis, but does a feed-forward network architecture even have the structure required to emulate a kinematic fit?

The common factor here, and in contrast to our previous computer vision example, is the existence of a known general theory which we would always expect to be superior to a learned approximation valid only over the range of training data.

Assuming we have a well defined coordinate system in the rest frame, we can compare the sets of kinematic variables to training data using the errors propagated from the original measurements. We are now in a position to apply machine learning to categorise the event. But the theoretical optimal way to do this involves use of Bayes theorem. As we could fix the required calculations as part of the network architecture would we really try to learn it instead?

We now have a classification decision made in a way which is consistent with using the Bayes conditional probability of classification (as explained above this is what machine learning does), with probabilities defined according to the cohort of simulation data we used to train the system. But wait! What we want to do is fit a mass peak to determine the optimal significance over background noise. This is a different criteria to optimising the Bayes error rate and as a consequence rare event detections (new physics) will be suppressed, just as with the sign-post effect described above. The corresponding variations in phase space may even be considered as noise and removed from the decision process. Is this the correct way to train the system? For practical use, even an image recognition system should really be taking into account Bayes Risk, i.e. the consequences of getting a classification wrong.

What about our training data? What are the systematic errors associated with using a finite amount of Monte-Carlo data? Does training encompass the phase space needed for all new physics sought? Whatever the form of analysis, this is always a problem for particle physics. By definition, although we can make a few guesses, we do not know exactly what new physics will look like so this seems to be a significant problem. If we get the Monte-Carlo simulation wrong we could be very carefully rejecting the very data we need to keep. If the sample is too small we may be locking significant systematic errors into data interpretation. This is precisely why standard approaches use a simple cut based selection, as it only has a significant influence on identification of events for kinematic variables close to cut values. These can be individually evaluated in order to avoid highly sensitive (unstable) analyses.

Finally, given all of the degrees of freedom for the construction of mathematical functions, how can we be sure that the best event detection computed using the deep learning system didn't involve calculation (or approximation) of the invariant mass? A complete kinematic description of tracks would provide all of the raw information needed to do this, and it would provide a very powerful way of identifying the particles we seek. This goes to the core of the problems associated with use of a black box. No particle physicist would think it allowable to include the invariant mass of the sought particle as a constraint when performing a search, it is equivalent to our earlier example of making use of camera specific information for face recognition. We could try to use adversarial training here, provided data can be generated in a form which satisfies the required training regime. This would need significant testing (probably in simulation) in order to confirm that it had worked.

Particle Physics Conclusions

The average computer scientists view may well be that deep learning neural networks are capable of constructing state of the art classification systems for large scale complex data-sets. But on slightly closer inspection, the suggestion to use deep learning for detection of new physics has led to a list of troubling questions.

- What are these methods expected to do which can lead to performance beyond that of existing multi-variate analyses?
- How should we represent data, (w.r.t. pixels, momentum vectors, coordinates)?
- Can we incorporate knowledge of measurement covariances?
- How should we best train the network to optimise detection?

- What are the systematic errors resulting from training on finite samples?
- How do we guard against distribution sculpting and discovery bias?

We could observe that after the first, these questions are very unlikely to be impacted upon by training methods which improve generalisation in large scale recognition problems. They are a consequence of far broader considerations regarding utility of pattern recognition. They also look reminiscent of those posed by the EPSRC "Neural Networks the Key Questions" initiative in the 1990's.

Appendix C: Linear Poisson Models

Our group has already developed a pattern recognition system which addresses many of the issues raised in this document. It is designed to construct a linear decomposition of histograms (e.g. mass spectra) based upon Poisson sample statistics. The examples we have used to illustrate use so far include: texture quantification in Martian terrain images, quantitative analysis of RELAX and MALDI mass spectra and detection of diffusion changes in pre-clinical drug trials using mice.

The method constructs a regenerative model which can be used to confirm the consistency of incoming data, both in terms of signal and noise. It makes quantity estimates (by adjusting the quantitative prior) which are consistent with Likelihood estimation. We have developed an error theory which allows us to compute uncertainties for any derived results, suitable for use in scientific and clinical studies. Unless you have been involved in such an analysis it might be difficult to comprehend what is needed to do this. It is therefore worth listing the steps which were found necessary in order to achieve quantitative rigour.

- The input data has to be known to follow Poisson sample statistics. This can be checked using Bland-Altman plots. In order to achieve this property input data often has to be carefully pre-processed.
- In order to find a global minimum it is necessary to train (optimise) many instances of the model (a linear manifold) from multiple (100's) of random parameter starts and choose the best.
- In order to obtain the best generalising result (which is also closer to a physical interpretation), a process needs to be applied to identify model order and then modify the extracted linear components.
- The final estimates of quantity and uncertainty have to be checked to make sure that the overall computational method has worked sufficiently well for use in scientific analysis. In the absence of ground truth a leave-one-out approach can be applied.

Potential issues which would be expected to cause these various check to fail are; unstable noise processes, non-linear (rather than linear behaviour), local minima and noise correlations. Non-linear behaviour in particular, would result in a form of approximation noise which is not accounted for by the associated probability theory.

Making the step from the LPM to a conventional MLP architecture is a relatively small one. However, simply taking arbitrary input data and training an MLP with a gradient based algorithm would not involve the level of care which is found to be needed for LPM's to provide a quantitatively valid model. In particular, having the theory needed to estimate statistical and systematic errors is not enough without additional steps which confirm the required properties of data. This perhaps hints at the challenges which will be faced when trying to use an ANN for quantitative modelling.

Appendix D: Aleatoric and Epistemic Uncertanity

This appendix is a review of "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision." by Kendall and Gal, 2017.

The paper details an approach to estimate epistemic (systematic) and alegotoric (statistical) errors in CV applications. In particular depth maps are estimated from input image data along with these two forms of uncertainty. This attempt and stated motivation accord with my own understanding and fit in with my list of ANN challenges.

I would not necessarily expect the assessment of uncertainties to be soley the province of Bayesian methods. Here, despite the title, there is very little which is Bayesian about this paper. It includes a prior term in eq (1) for the purpose of motivating the use of a weight decay algorithm. No specific quantitative justification is provided for this, although we can observe that it is common to other papers. As usual for MAP, the methodology is arbitrary as the term is based upon a density over parameters, which is a specific user choice. For example we could also

have chosen to use the density over parameter cubed etc, and so obtained different results. For any construct in probability to be meaningful it must be independent of user choices. This is a hack which has been selected for its empirical behaviour (turning off weights to increase mapping smoothness) and retrospectively justified as a prior. This term could just as easily have been excluded, making the approach entirely consistent with Likelihood (see below). Although it is claimed that epistemic uncertainty can be estimated nothing is mentioned regarding optimising (reduction) of this estimate, which might seem to be the obvious way to derive a regulariser. Also, I did not see mention of how the overall cost function is to be optimised, but I assume it is gradient based.

Epistemic error is based upon $P(W|X,Y)$, which is "estimated" here based upon a dropout process (detailed in earlier work by the same author). This appears to have no link to more conventional statistical approaches to estimate parameter uncertainties (i.e. Minimum Variance Bound or boot-strap resampling). However, these alternative approaches are only valid when the mathematical model is valid, and we have no test here to confirm the adequacy of the final ANN mapping.

Estimation of aleatoric uncertainty (eq(4)) is a slight rewite of the sample covariance estimate and therefore can be accepted as valid Likelihood estimate.

Eq (5) is the conventional Likelihood expression for minimisation when including errors on input data (var(x)). This can also be taken as valid.

Eq(7) details the same idea as (5) but for the output (y), which is then rewritten as a function of $s = log(\sigma)$. The stated justification for this is the elimination of poles in the cost function. I believe that the real reason is to map the variance estimates into a homoscedatic space, in order to use the chi-square function (eq(8)), again as the basis for a valid Likelihood.

Aleotoric uncertainty (statistical error) on output is obtained by training an output from the network to predict s by minimising eq(8). This appears to be a mapped approximation to the alternative of applying error propagation to the input errors. As we have no check on the efficacy of this mapping this may fail and must therefore be considered less satisfactory than error propagation, which would be reliably provided for free by backprop.

Results demonstrate that depth estimates can be generated along with two estimates of uncertainty, in the form of output images. However, in these results, aleatoric errors are largest at edge boundaries. Indeed the images of estimated depth are clearly smoothed across object boundaries. The authors do not mention (or perhaps know) that these locations are where most (Fisher) information for calculation of depth is to be found, and so should have been the most precise. The aleatoric uncertainties therefore document the inability of the ANN to discover that differential discontinuities can be predicted from the image, i.e. the key CV principle of diffeomorphic equivalence. Instead the ANN just obtains, and then reports, very poor performance at these locations.

To put this result into some historical context, in previous CV work (e.g. Nelson's 3D vision system), object boundaries are represented as two co-located curves, one on the object and the other on the background. This allows depth estimation to adopt two depth estimates at the same spatial location in an image, and so represent differential discontinuities. Alternatively, in Poggio's sceme for interpolating surfaces between edges, any assumption of smooth continuity in the output was relaxed at the image edges. I would expect any method which claims to be a state-of-the-art solution to re-discover such "tricks" or similar. Clearly it has not.

There are two possible reasons for this, i) either the Bayesian prior has inappropriately smoothed the mapping function, or ii) (more likely) the ANN has no choice but to attempt to generate a smooth output. If it is the former then it is just an issue of inappropriate use of an arbitrary prior. This is then a criticism of MAP estimation and weight decay algorithms. If the problem is the latter this would mean that an ANN may be incapable of mapping well any output with differential discontinuities. As the real world is full of depth discontinuities, this limitation would apply to all shape-from-X solutions, and would pose a significant restriction on the valid use of black-box ANN's for CV.

Although I would also expect any assessment process which evaluates the average performance across all image locations to be insensitive to what is happenning at (the relatively infrequent) edge boundaries, this is something that researchers with some understanding of the nature of the data they are using, or informed of historical approaches, should surely have noticed.