

AI Coworkers

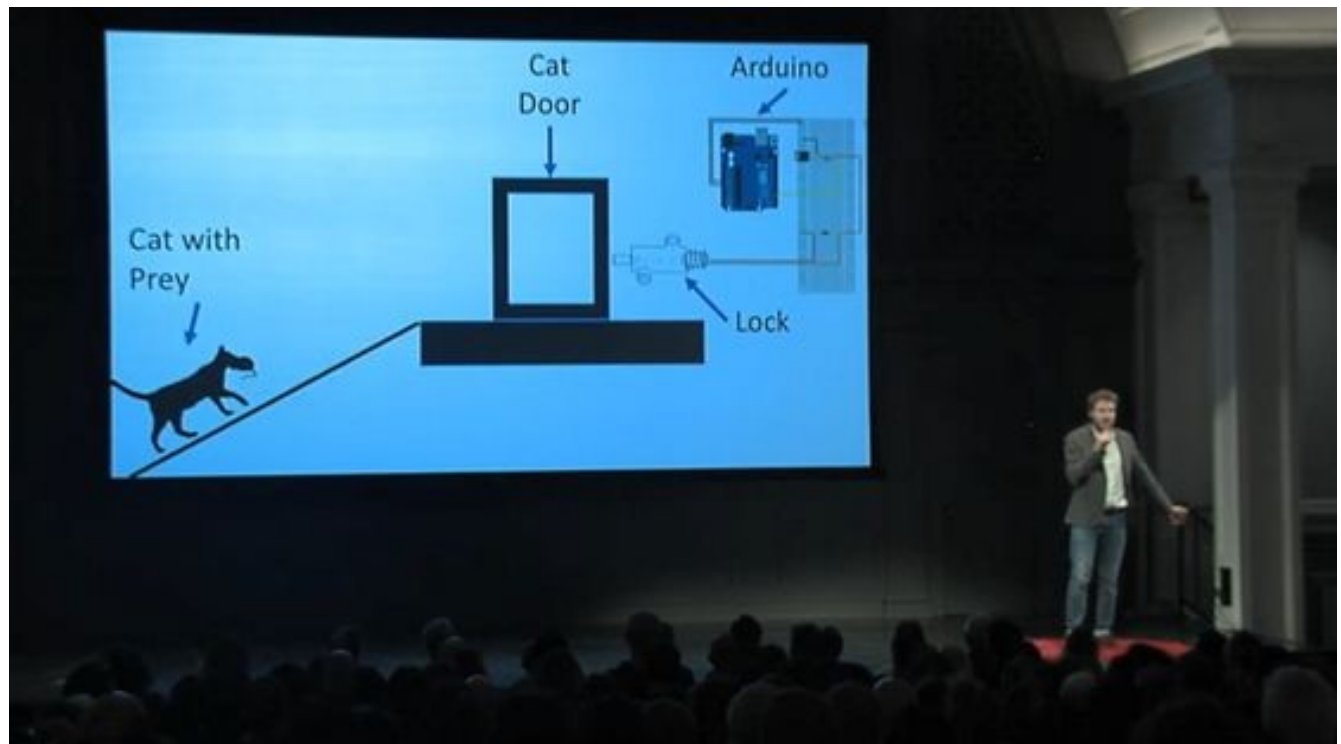
davidbarber

We argue for a more interactive approach in which AI systems function more like coworkers. For them to be effective in this role, they need to have reasonable estimates of confidence in their predictions, giving them an opportunity to learn and for humans to gain trust.

Building a better catflap

We start with a story about a man that wants to build a better catflap. The story highlights some of the amazing things one can do with machine learning, but also things that are wrong with the way AI systems are trained.

Ben Hamm wants to [build a better catflap](#):







His cat often catches prey during the night, and wants to bring them into the house:



His idea is to use a camera that can monitor the ramp up to the catflap

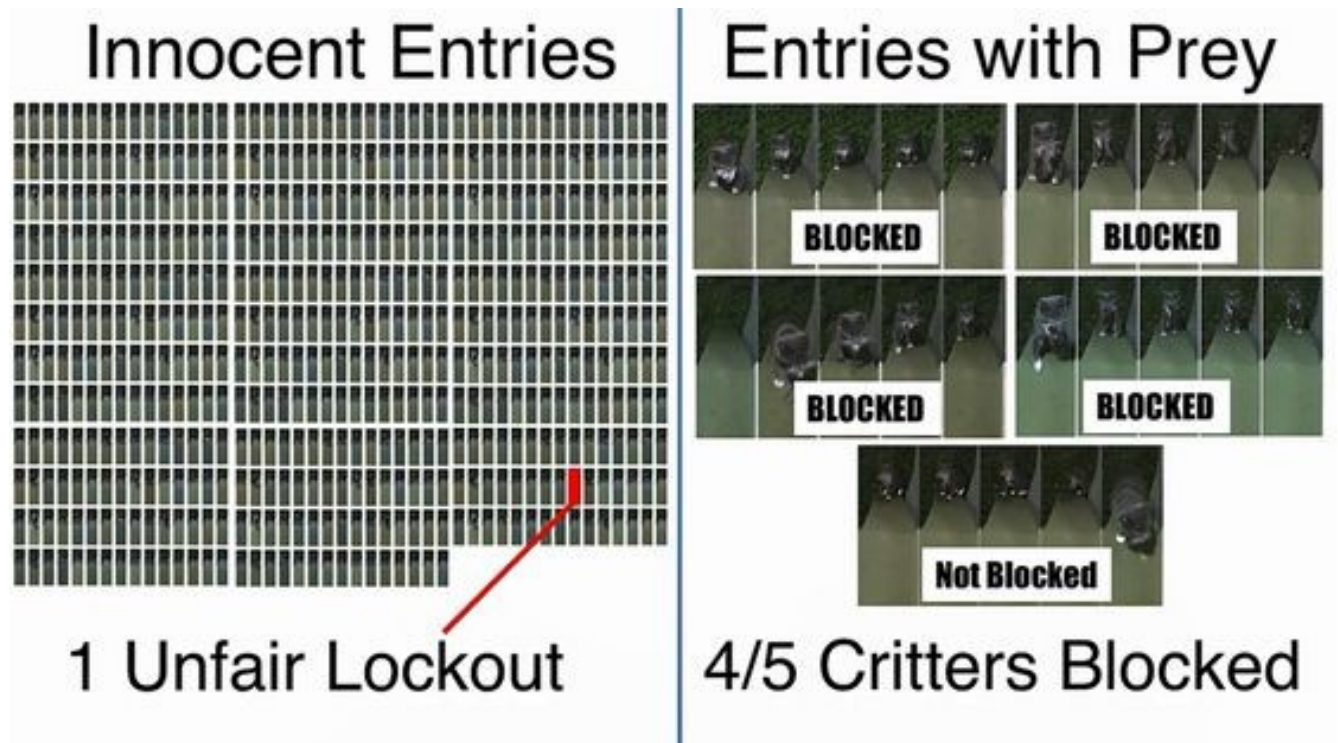


and then block the catflap if the cat is carrying prey. Over several months his camera captures 23,000 images of his cat. He laboriously hand labels each of these 23,000 images to provide training data ...

Image Type	No Cat	Cat not on approach	Cat on approach	Cat with prey
Count of Images	6,542	9,504	6,689	260
Example				

... and uses them to train deep neural networks to detect whether his cat is on the ramp with prey. Ben is not a machine learning expert, but is able to use the great free tools available.

The new catflap works pretty well during 5 weeks of testing. It correctly let the cat in 179 times (just one time unfairly locked out) and, of 5 times that the cat brought home prey, it was successfully blocked from entering 4 times:



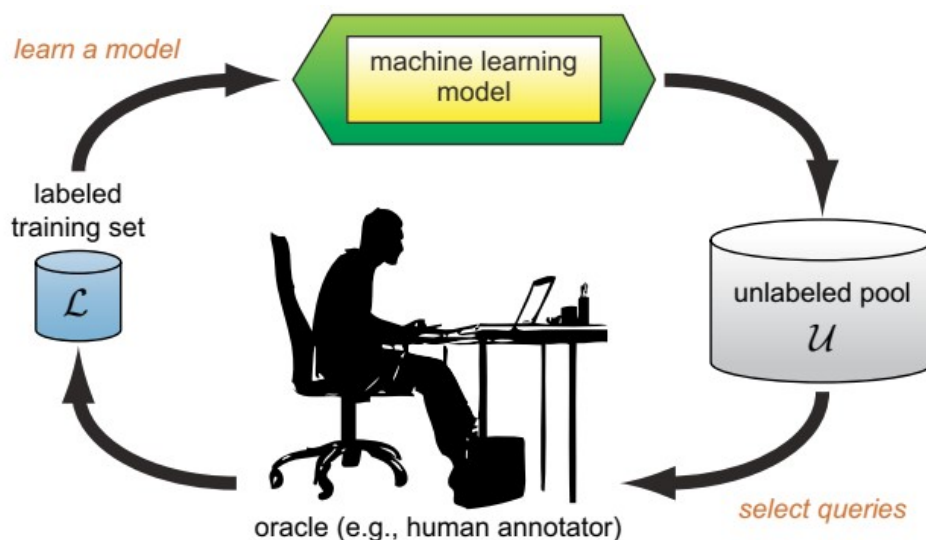


What's wrong with current Machine Learning?

Whilst the above story shows how far machine learning has come, it also highlights some of the current issues with machine learning:

1. Labelling training data by hand is very laborious and time-consuming. This is perhaps one of the biggest bottlenecks faced in industry in training machine learning models.
2. We need a lot of labelled training data to train a deep neural network.
3. Some problems are relatively easy for inexperienced humans to label (for example whether a cat has a bird in its mouth). However, others (for example medical diagnosis of CT scans) may not be and the number of human experts available to provide labels is scarce and labelling very expensive.

Active Learning



The standard paradigm in training machine learning systems is (like Ben did) to collect a set of data and then get people to label them, either in-house (if the data is sensitive) or externally (for example by using Amazon's Mechanical Turk).

Ben did the labelling 'in-house' (himself). However, did Ben really need to label all 23,000 of those images? An alternative approach is to use so-called Active Learning¹ that selects only a subset of the training data that needs to be labelled. Whilst this has been around for some time, it has still not permeated deeply into mainstream industry practice.

We will assume that, as in Ben's scenario, we have plenty of data (eg images), but we don't have labels for them.

1. In Active Learning, one starts with a small amount of labelled data to train an initial model.
2. The trained model then looks at the remaining unlabelled images. Some of the images will be similar to those that the model has already seen labelled data for and it will therefore be confident in its prediction. There is no need for these images to be labelled by the human. However, for images that are quite different from those that have been currently labelled by a human, the machine is likely to be less confident in its prediction. There are many different criteria that can be used by the machine to select which images it would like to label, but most rely on using the machine's estimate of its certainty in its prediction. For example, images for which the machine is least confident in its prediction are passed to the human for labelling.
3. The human labels these (machine chosen) datapoints.
4. After labelling, the model is retrained (on all labelled data).
5. The process (steps 2 to 4) repeats until convergence.

In this way the machine plays an active role in selecting which data it believes is useful for the human to label. This approach can be quite effective, to the point that only a small fraction of the data may need to be labelled in order to get predictive performance close to that which would be obtained from labelling the whole dataset.

Uncertainty : The potential Achilles Heel

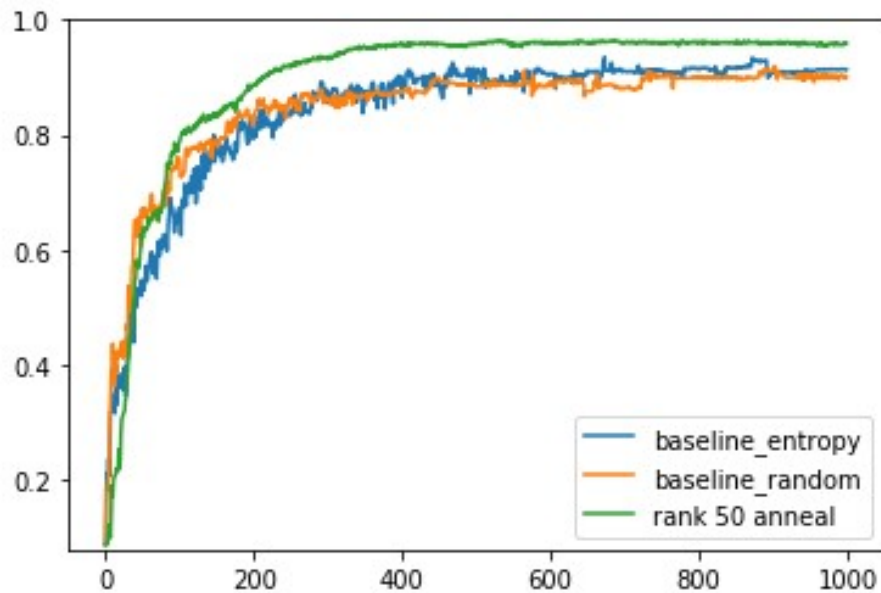
Whilst Active Learning holds great promise to drastically reduce the need to label a large amount of data, it does come with some risks. If the machine's estimate of its predictive uncertainty is poor, then the machine will select examples that are not appropriate and the machine will never see the labels it needs to generalise well.

Similarly, once trained and deployed, if we want to use our AI systems as smart coworkers, we need to trust their judgements, knowing when the AI system is not confident in its prediction. Imagine a human coworker that is arrogantly overconfident in their predictions. Sometimes they will be correct, but other times confidently predict the wrong answer. Being overconfident (when they shouldn't be) means that a vital opportunity to learn is lost. Humans in the workplace are highly sensitive to overconfident individuals since arrogance is rarely appreciated(!) and ignorance can result in valuable opportunities to learn and better understand being lost.

Many of the recent research trends in Deep Learning have not focussed on providing good estimates of uncertainty in the predictions. In the classification context, its standard for a deep network to output a class probability $p(c|x, \theta)$ where x is the input and θ are the weights of the network. For example, the network might output $p(c = 'cat' | \text{input image}, \theta) = 0.8$, and this can be taken as a measure of the prediction uncertainty. However, this uncertainty is based on *assuming that the network model is correct*. Since we may have only a small amount of data, then our confidence that the network weights θ are appropriate may itself be low (parameter uncertainty). Similarly, the network architecture itself may not be confidently determined (structural uncertainty). There have been attempts to incorporate such uncertainty into the predictions (for example Bayesian approaches or using a committee of networks trained on different datasets) and

these can be helpful in producing better estimates of prediction uncertainty.

Active Learning with an image classifier



The above figure shows the test accuracy of a model trained to predict the famous MNIST postcode digits. There are 60,000 images of handwritten digits (each image represents a digit from 0 to 9). When trained on the full dataset of labelled images, machines can reach prediction accuracies above 98%. The figure shows the progress of Active Learning, with the x-axis showing the number of labelled examples used so far; the y-axis is the prediction accuracy on a test set. There are three approaches used to select the next training datapoint, with each network being retained after receiving a new datapoint.

1. [orange] The random baseline simply selects datapoints at random to label. This can often be quite effective if the predictor does not provide a good estimate of its own uncertainty.
2. [blue] The entropy baseline uses a standard deep network, with datapoints for Active Learning selected on the basis of the uncertainty in the softmax neural network prediction. This is a limited estimate of the uncertainty in the prediction and means that the Active Learning process does not select good examples to be labelled.
3. [green] This is a Bayesian approach that takes parameter uncertainty into consideration. This gives a better estimate of the prediction uncertainty, meaning that Active Learning selects much better examples to be labelled. After training on only around 500 labelled datapoints, the test accuracy is comparable to standard training using 60,000 labelled examples in the full training set.

Whilst not yet commonplace, companies such as [re:infer](#) successfully use Active Learning and Natural Language Understanding to help customers rapidly train AI systems to derive insights from communications data and facilitate Robotic Process Automation. This is key to helping rapidly onboard new clients and get their systems up and running, without needing costly and lengthy data labelling sessions.

Weak Learning

The traditional way to provide training information to AI systems is through the simple labelling “cat with prey/cat without prey” style approach mentioned above. However, humans often have

much richer information about the problem. For example, it's clear that it would be useful for an image system to focus attention on the cat's mouth region to help determine whether the cat is carrying prey. Whilst this is probably a pretty good hint, it might not always be the case that the cat will be carrying prey in its mouth. These kind of hints can nevertheless be very useful in reducing the amount of labelled training data needed and can be thought of as weak rules, resulting in so-called weak learning².

More recently, related ideas called (perhaps somewhat unusefully) “self-supervised learning” can also help reduce the burden of data labelling, with humans only needing to place examples into some equivalence class — for example, “these two images are of the same thing”, or “these two images are different”, without needing to necessarily specify additional details. These can result in learning more robust features of the problem and improve generalisation³.

[Humanloop](#) is another noteworthy recent UCL spinout that provides a data labelling platform to more rapidly train models using Active and Weak Learning.

Summary

Whilst machine learning has come a long way, the way that machines interact and learn from people is still somewhat clunky. In order to help train these systems, particularly when data is scarce or private, it's important for machines to be able to learn effectively and interact in a more natural with human.

In a similar way, it's important that in deployment that AI systems also state when they are not confident in making predications, enabling humans to step in and, through natural interaction, explain potential solutions. This ability to have a measure of prediction confidence is vital for AI systems to “self-reflect” on what they know and appeal to humans for help. This is a natural trait for any useful coworker.

References