In recent times, it has been found that advances in deep learning have solved many complex problems such as object detection, image classification, face recognition etc. These problems were considered almost as "NP Hard" before the rise of neural networks. Additionally, the concept of transfer learning along with the availability of huge data has made the deep learning models unbelievably more precise and accurate as these models, more often than not, give accuracy over 90 percent. However, does it imply that these models have reached human level intelligence and do we have cracked the code for Artificial General Intelligence? The simple and straight forward answer to this question is NO. We will try to discuss this in further details in this article (with minimum mathematics, if possible!).

For any task such as image recognition or face detection, we train our AI system or deep learning model with data which is considered to be relevant for given task. The basic and most controversial assumption behind using this data is: The given sample of data is true representation of population under consideration and this is the point where dataset bias comes in. It has been found that there is very small probability that the sample of data which we have chosen to train our deep learning model will be representative of population and therefore, our deep learning models often miss out on capturing ALL the variance or features of population data. Yes, it is possible that our deep learning model might achieve astonishing accuracy on held out test data; yes, it is possible that our deep learning model might capture interesting objects that even human eye cannot see; however, this definitely doesn’t imply that our model is perfect in all senses. At the most, we can just state that our deep learning model has correctly learnt the underlying data distribution of training data. The great performance of our model on held out test data can be attributed to the fact that test data comes from the same data distribution as that of training data and since our model has correctly learnt the data distribution of training data, our model is performing well on test data. There is high probability that population data might come from data distribution which is different from training data distribution and therefore, our model might perform poorly on such data which is coming from different data distribution than training data. For example, suppose we have developed a deep learning model for classifying between Ferrari and BMW. Let us assume that we have trained our model with images of Ferrari having red colour only. Hence, it is possible that our deep learning model might interpret the red colour as a feature of Ferrari and this is definitely, not a relevant feature for Ferrari as there are many Ferraris which have colour other than red. Hence, there is clear distinction between training data distribution and actual population data distribution and therefore, our model might not detect Ferraris which are having colour other than red. This is the simplest intuition behind the difference in data distribution of training and actual population data.

Therefore, to test AI system for checking whether we have covered all the variance of population using training data, we can create **adversial** examples. Adversial examples are training instances which are very similar to data instances or examples present in training data distribution and a normal human can hardly see any difference in image from training data and corresponding adversial example; however, these adversial examples are created by small, intentional changes in features of training data. Therefore, these adversial examples come from a data distribution which is different from training data distribution. Therefore, our deep learning model would perform poorly if training data that we have provided is not representative of actual population.

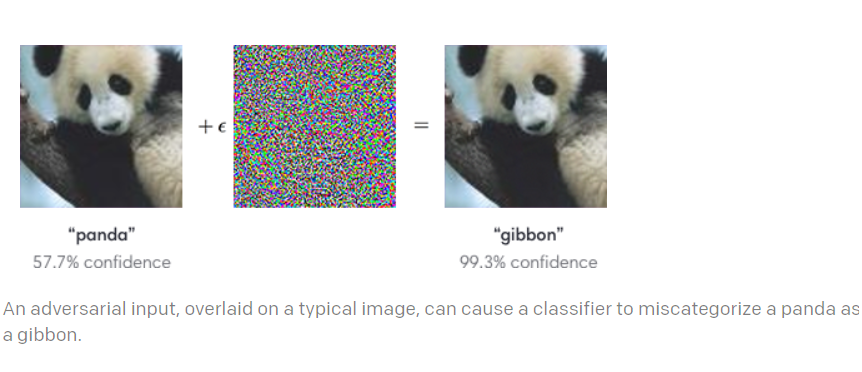
Since we now have an intuitive understanding of what adversial example is, we can focus on how can we create such adversial examples. To create adversial examples, we have to dig into the gradient descent algorithm followed during the training of our neural network. Normal gradient descent algorithm often tries to minimize the loss function defined. For example, suppose we are performing a classification task and task is to detect whether an image contains Ferrari or BMW. Normal gradient descent algorithm trains the network in such a way that we change the weights associated with each pixel in an image so that loss is minimized. Therefore, we often change weights in a direction which is opposite to gradient, resulting in the movement towards minima of cost function defined and thus, we ultimately minimize the loss. Mathematically, it can be represented as follows:

w=w+α\*(-∂L/∂w),

where, minus sign in partial derivative of loss function with respect to weight denotes that we are moving in a direction opposite to gradient.

The idea behind generating adversial examples is exactly opposite to that of gradient descent. Instead of going in the direction opposite to gradient, we go in the direction of gradient with respect to its input (each pixel of the image), for a small amount, so small that human eye cannot distinguish between original image and adversial image. However, this small perturbation in each pixel of the image creates image which is statistically very different from original image and therefore, our deep learning model might get fooled and can give erroneous output. So, long story short, for creating adversial examples, we look at the sign of gradient (positive or negative) used to train our model and we simply follow that to create small changes in input image; however, these small changes are considered as “catastrophic” by our model and thus, it might provide wrong outputs with high confidence.

Following image depicts perfectly what adversial image is and how it might fool our deep learning model:



In original image of panda, our model predicts perfectly that we have panda in the image. However, if we add small amount of noise (€) to input image, deep learning model miscategorises panda s gibbon, even though we, as humans, hardly see any differences between image on LHS and RHS. This is a definite indication that our model has not been yet exposed to sample of data which can be considered as true representation of population data. The image on the RHS is generated by following the exact same procedure discussed before, that is, look at the sign of gradient for each pixel in the image and go in that direction of gradient instead of going opposite, for a small amount. In this way, deep learning model might get fooled even with small changes in input that are not noticeable to humans.

**Conclusion:**

Deep Learning models developed are still far away from realizing the dream of Artificial General Intelligence. However, techniques such as generative networks and Explainable Artificial Intelligence are definitely a way to go forward for realizing this dream!