The deep in deep learning means that there are multiple layers in then neural network.

The last layer is the classifier/regression. You can actually replace the last layer and stick the result in an SVM/Random forest/whatever if you want to.

What deep learning is really about is feature extraction. Supervised or unsupervised.

The difference between traditional ML approach and DL approach is that in traditional ML you'll spend most of your time feature engineering and doing all kinds of magic tricks to the data before you even model it. That's where all the work happens.

The deep learning approach is to throw data at the algorithm and adjust the dropout and crunch some numbers on your quad 2080TI rig for a week until you get the results.

This approach will beat any other approach given enough data and computational resources. You can use tricks like transfer learning where you learn some of the feature extraction steps on some other data (supervised or unsupervised) and then use the embeddings on your new data. So your "100 pictures" dataset becomes "100 pictures and 14 million of imagenet pictures".

Similarly in IoT, audio, raw signals, video etc. you can use your unlabeled raw data to pre-train your networks to an extent and use a smaller labeled dataset to finish the training.

If you don't have something extremely "raw" and information rich that you need to extract features out of, then deep learning is a wrong approach.

Let's say I have a 1000x1000 JPEG. Thats 1 million pixels so 3 million features (3 colors). Let's say I have a raw sensor signal at 100Hz. So 1 minute of that raw sensor data would be 6000 features.

Now we can extract meaningful features out of that mess of data, but you need to do it manually. Features extracted from a heartbeat signal will be different from features extracted from a thigh muscle signal. Similarly features extracted for facial recognition will be different from features extracted for cucumber size classification.

Deep learning makes "familiarize yourself with the data", "understand the domain to extract meaningful features" etc. irrelevant.

It truly is "put on a cowboy hat and scream YOLO" of data science requiring 0 understanding of the domain and 0 understanding of pretty much anything else such as statistics. All you need is to understand deep learning. It leads to unreasonably good results and it's infuriating when you've worked for 20 years and became an expert in this niche and then some 19 year old comes with an keras based LSTM and beats your state of the art algorithm you've developed for 20 years by studying the domain thoroughly. And that 19 year old doesn't even know anything about the domain.

Because the feature extraction is fully automatic, deep learning is unreasonably good at finding features from seemingly thin air. Things people thought are noise or relationships too complicated can get picked up by neural networks no problem. This means you can get completely new knowledge because it's truly a machine approach that will be different from the human approach. Good luck finding out how/why exactly it works though.

Finally because of the automatic nature, it works great with dynamic data. Where things, trends etc. keep changing and you can't build a model and forget about it. You might be able to pump out a new model once a year by analyzing the trends and the changes in the domain, but you won't be able to do it every month, every week or even every day.

In the real world you rarely get to use deep learning because there isn't enough data, the data is aggregated/calculated/derived and features are already meaningful and there isn't much to extract, the patterns in the data are fairly static and so on.

In my experience deep learning beats everything else when you start with raw data since it usually extracts features better than anything else (including humans).

I've never seen raw data in the wild. Even when someone says "raw sensor data" they really mean summaries such as minimums, averages, maximums, peaks and other engineered features etc. I've never seen someone hand me raw data because they can't comprehend why someone would want the actual raw samples.

tl;dr deep learning is feature extraction on steroids and isn't that great if you've thrown away 99.99% of the information by feature engineering it first.

In many natural sciences, where you need to fit multi-dimensional data with a neural network, 1-2 layers is sufficient. Read this article.

<https://www.heatonresearch.com/2017/06/01/hidden-layers.html>

I've been looking myself some published articles were very deep neural nets are claimed to be successful. I do not think I've seen a very convincing one. They often miss a fair comparison with shallow networks where you actually understand the input data, and design correctly feature space taken into account the underlying dynamics (in many cases, we know it, or at least, we have a good guess about it.). And this is all science about - understanding problems. The modern trend is to throw raw data into a very deep neural net (assuming a lot of CPU and a huge amount of data), and get prediction, without much thinking what the predictions are, and are they trustful? I would appreciate if somebody point me some paper in a peer-reviewed journal (do not care about blogs) where deep learning engineering (I do not call it science!) shows better performance than the standard ML with 1-2 layers. Especially I'm interested in any article with this claim: we tried 1 hidden layer, we tried 2 hidden layers, etc etc. and, finally, we tried X hidden layers (aka deep learning), and here is the magic - it over-performs every simple ML we tried before.