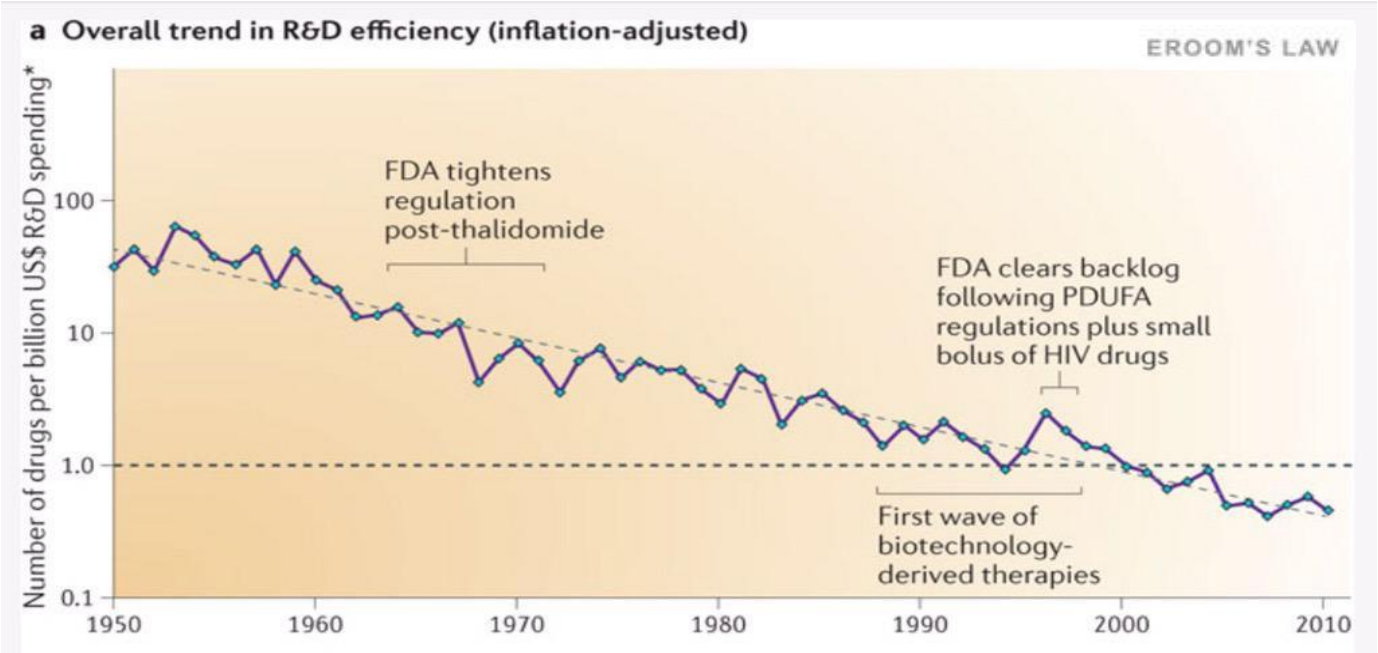
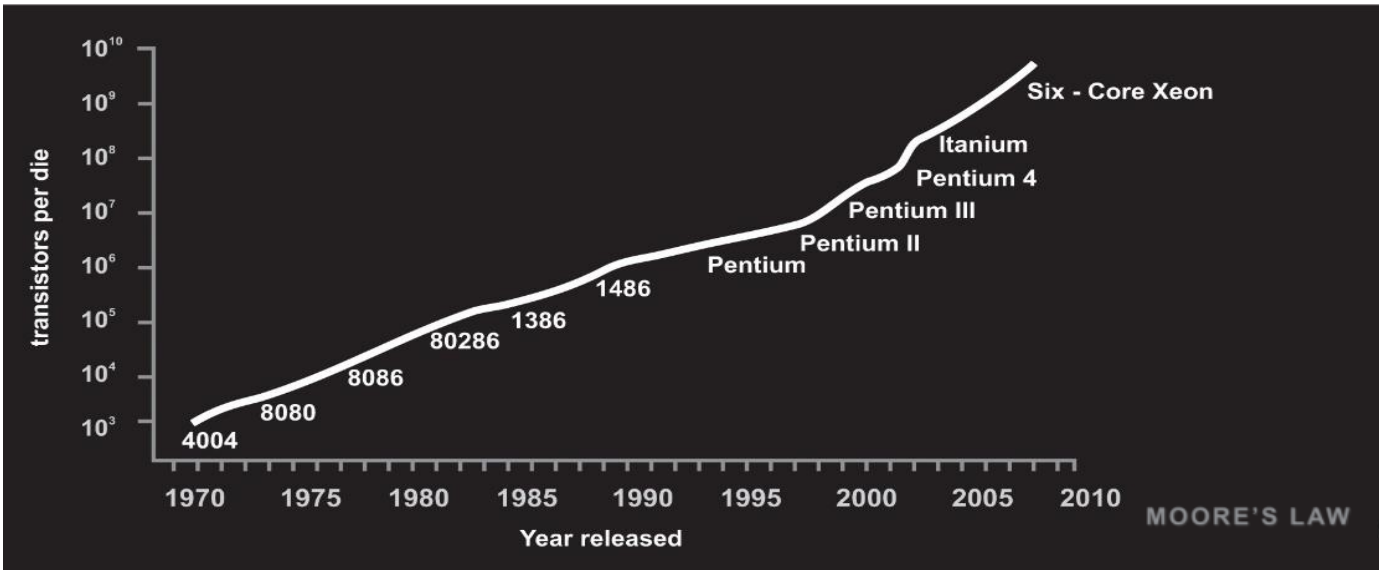
Deep Learning in Pharmaceuticals: Answering the R&D Epidemic

Lucas Rappette

Artificial Intelligence (CS 452)





Research and development for new drugs in the pharmaceutical drug industry has become increasingly more expensive and difficult since the 1950's. [8]. Drug manufacturers are experiencing a stall in the amount of new drugs brought to market each year accompanied by a nearly 100-fold increase in the average cost of development for a new drug between 1950 and 2010 [1]. The phenomenon is known as "Eroom's Law", Moore's law spelled backwards [8]. Coined in 2012, the law states that the constantly increasing problem complexity of discovering novel therapeutic drugs previously not known about that outperform the ones we already have will cause the average cost associated with bringing a new drug to market to double every 9 years [1,8]. In an effort to reverse this trend described by the graph above (and consequently to "escape the Eroom"), drug manufacturers are turning to artificial intelligence teams utilizing deep learning networks to not only reduce the time it takes to discover and bring a new drug to market, but to also potentially find and patent alternative therapeutic uses for drugs previously used to treat another ailment or disease [2, 5]. In this paper I will attempt to explain just how researchers are managing to encapsulate biochemical innovation in a deep neural network.

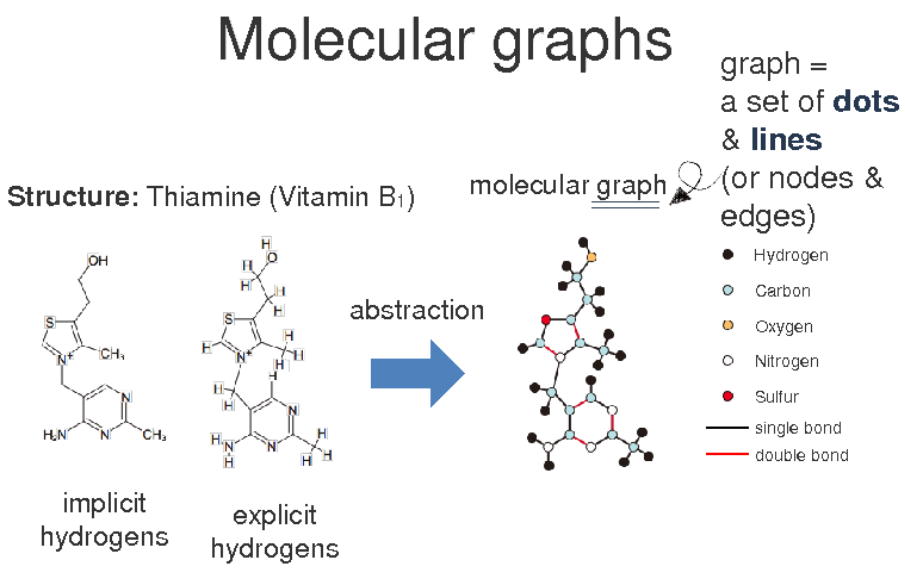
It wasn't until just recently that deep learning has been considered a feasible option to aid in pharmaceutical research; it is an immensely complex problem that requires training on networks in size that are well beyond what could ever fit on the memory of a CPU (or a single GPU for that matter) [2]. As technology and algorithm efficiency improved for machine learning in drug discovery, a hope of true deep neural networks for medicine grew. However, it wasn't until the maturation of GPU-accelerated processing, the availability of open source software training packages such as Google's

release of Tensorflow, and the subsequent release of stable open source Tensorflow API's such as *DeepChem,* along with its included molecular benchmark program *MoleculeNet* that has made true deep learning on small molecules achievable [2, 6].

In a standard *feed-forward* neural network, one or more *hidden layers* of neurons are connected to an input layer comprised of a set of numerical inputs known as *features*. Features from the input layer are then passed on to the first hidden layer's corresponding neurons which produces an output value that is mapped on to the corresponding neurons in the next layer [2]. A path from one neuron's function to another neuron in the next layer is considered an edge with a weight associated with it [2]. The "weight" of the edge is an adjustment parameter to minimize the error function associated with the output [2]. The last layer of the neural net (called the output layer) corresponds to the object we are trying to predict, which then undergoes the process of backpropagation through the neural net to update edge weights of each neuron until we are back at the input layer. Terminal output is reached when either the network's error function converges or calculated errors stop decreasing [2]. Such a network like this requires large-massive amounts of data in the form of datasets to "learn" properly.

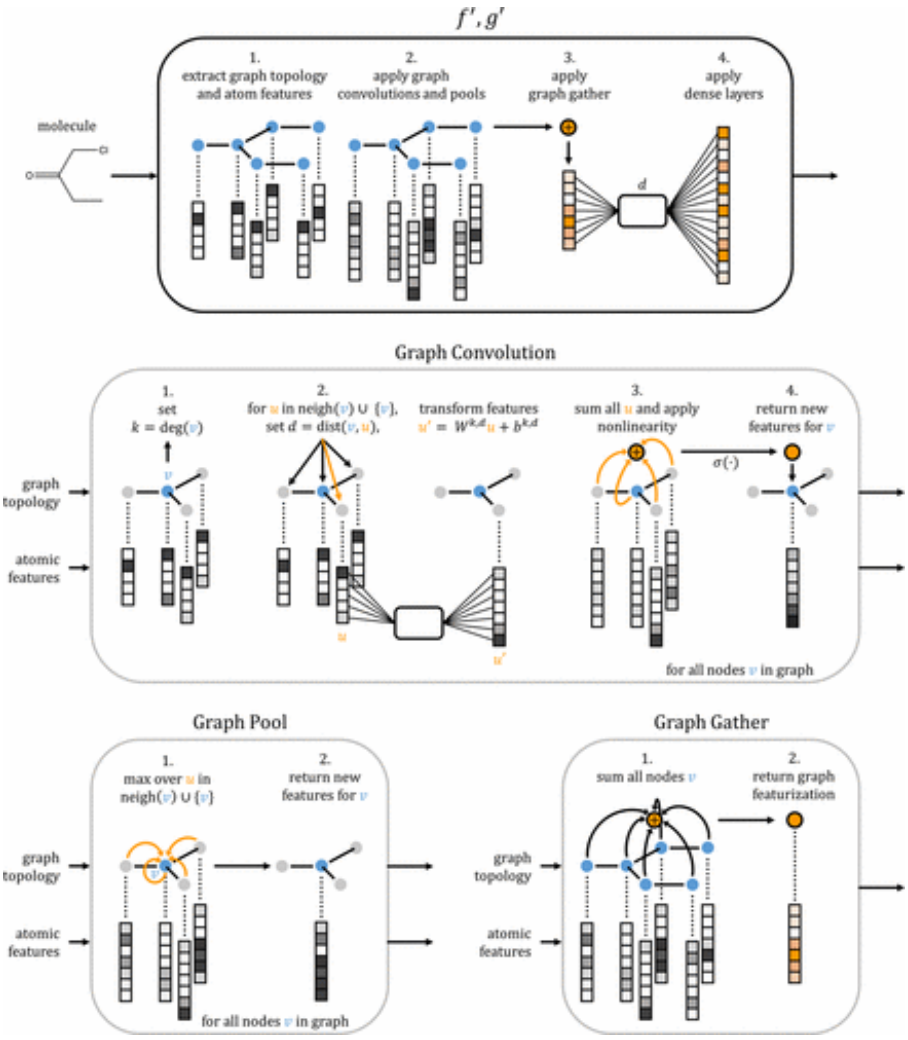
In order to train a neural network, all edge weights between neurons are originally set to some random number [7]. The network then takes in feature values given by a specific training example from the input layer and steps through the layers of the network until the output layer is reached. If the output is not terminal then backpropagation will occur, minimizing the error associated with a single specific training example by adjusting the weights and passing through the network again [2]. Convolutional neural networks are like feed-forward networks except they are

comprised of multiple layers of neural networks which characterize a hierarchical order of abstraction (and thus features) on an input image or graph before passing the vector of features to the feed-forward neural network [2, 6]. Deep learning is the name given to these such networks because they have the ability to automatically extract features from a set of inputs rather than explicitly defining them.



In comparison to other deep neural networks, neural networks used in DeepChem (and networks for all molecules in general) break the traditional approach to neural network modelling in a few key ways [6]. For starters, the nature of a molecule is inherently unsuitable for explicitly defined features [6, 7]. Molecules for a deep learning network such as DeepChem come in all sorts of shapes, sizes, and connectivity [7]. To transform molecules into a form suitable for conventional machine learning algorithms (that usually accept fixed length input), we must extract relevant and related

information from a molecule into a fixed dimensional representation [7]. The most elegant solution to this problem is to use convolutional networks on a graph representing a molecule to create a "neural fingerprint" of that molecule [7]. An example of a molecular graph is shown above.



By defining a convolutional neural network as a circular fingerprint that applies the same operation locally everywhere and that is differentiable at each layer where each atom is described as vector of features, then it's possible to transform molecular inputs

and reconstruct them into a graph feature vector of fixed length by pooling [7]. The image above summarizes this general idea.

Another paradigm that is completely reversed in deep learning for pharmaceutical drugs relative to other deep networks is the *lack* of large data sets for training [6]. It's been hypothesized that the difficulty in finding large data sets for molecule-based machine learning is because obtaining precise and accurate results for chemical properties typically requires specialized and very expensive instruments as well as expert level supervision [6]. In response to this, a DeepChem group committed a "one-shot learner" which functions on a very low number of data points instead [9]. The one-shot learner is a way to quickly map out sufficient representations of features with only a couple examples [9]. Based around the hierarchy that is innately part of deep learning and provided by the graph feature vector, these methods work by using related data to learn a meaningful distance metric over the space of possible inputs [9]. The drawbacks to one shot learning (at least right now) is that it struggles to generalize on patterns it has never seen before, most likely because the state space grows exponentially fast [9].

The real gold mine of deep learning for drug discovery is the multitask network, which allows you to predict multiple outputs of interest simultaneously on a network [2]. The implication of simultaneous execution is that the input featurizations for every task running simultaneously are all from a shared feature extraction pipeline which the deep network built itself [2]. By having neurons run processes on features linked by reference label means the weights of shared features will reflect an even stronger statistical model then a single task due to better generalizations from the increase in

state space [2]. Furthermore, it's possible that with the first low data one shot learner coming to fruition that these shared statistical models can be established *almost right away*, providing a ramp up in performance.

The analysis of methods on how deep neural networks grow to more complex relationships is both an exciting and confusing time. The community is building *and discovering* the power of the DNN through new network architectures, big test data, and multitasking but there is no sense in which we can see what the deep neural network is truly learning. Whatever features it identifies are hidden from us, which isn't desirable for long term monitoring of a Turing-like machine. Nevertheless drug discovery with deep learning is looking optimistic as ever with powerful opensource libraries already being used in high through put screening and re-evaluation of known therapeutic agents.

References

[1]. Jack Scannell and Jim Bosley. When Quality Beats Quantity: Decision Theory, Drug Discovery, and the Reproducibility Crisis. *PLoS ONE,* 11(2): e0147215, 2016.

[2]. Garrett B. Goh and Nathan O. Hodas. *Deep Learning for Computational Chemistry*. *arXiv*, 1701.04503, 2017.

[3]. Artur Kadurin, et al. The Cornucopia of Meaningful Leads: Applying Deep Adversarial Autoencoders for New Molecule Development in Oncology. *Oncotarget,* 8:10883-10890, 2017.

[4]. Ching Travers, et al. *Opportunities and Obstacles for Deep Learning in Biology and Medicine*. Cold Spring Harbor Laboratory, 2017.

[5]. inSilico Medicine. "Pharma.AI Launches to Apply Artificial Intelligence to Drug Discovery and Development". www.eurekalert.org/pub_releases/2016-03/imi-plt031516.php, 2016. Originally posted as a AAAS blog post. Last retrieved from Internet Archive: 20 Dec. 2017.

[6]. Zhenqin Wu, et al. *MoleculeNet: A Benchmark for Molecular Machine Learning*. *arXiv,* 1703.00564, 2017.

[7]. David K. Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, Ryan P. Adams. *Convolutional Networks on Graphs for Learning Molecular Fingerprints*, 2017.

[8]. Jack W. Scannell, Alex Blanckley, Helen Boldon and Brian Warrington. Diagnosing the decline in pharmaceutical R&D efficiency. *Nature Reviews Drug Discovery,* volume 11, pages 191–200, 2012.

[9]. Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. Low Data Drug Discovery with One-Shot Learning, *ACS Central Science,* 3 (4), 283-293, 2017