# Reinforcement learning based drug research & discovery for the COVID-19

## Accelerating drug discovery by leveraging machine learning to generate and create retro-synthesis pathways for molecule

## 

The way we discover drugs is EXTREMELY inefficient. *Something needs to be done.*

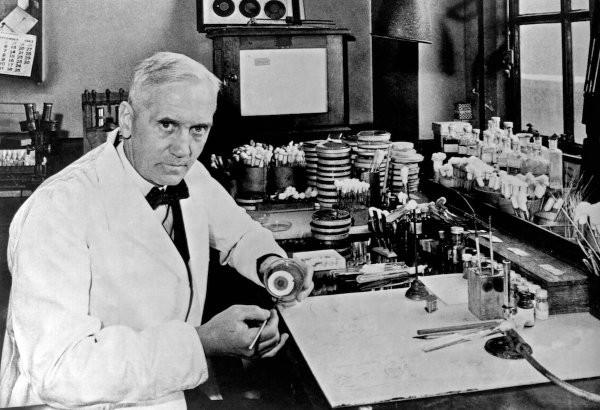
Despite all the innovation that is happening in the pharmaceutical industry recently, especially in the cancer research space, there’s still a huge gap for improvement!

Our current approach to drug discovery hasn’t changed much since the 1920s.

**This is the story of how drugs were once discovered:**

In 1928, Alexander Fleming, a pathologist, often described as being “care-free” accidentally left his petri dish uncovered beside a window before leaving for his month-long vacation.

Arriving back from his lovely vacation, something even *lovelier* happened. \****To his surprise\****, the discarded culture dish led Fleming to the **remarkable discovery of the world’s first antibiotic, Penicillin, which disrupted the pharmaceutical industry** 🤯**.**

****

Just one of the many examples of how a petty mistake manifested itself into a therapeutic breakthrough.

**Fast forward 8-ish decades later, this is how drugs are currently being discovered:**

Recently, New Zealand researchers \****unexpectedly\**** discovered that the vaccine previously used to combat the meningitis epidemic in the early 2000s also subsequently **lowers the risk against gonorrhoea.**

*The secret sauce to drug discovery has never seemed to escape the magic of serendipity despite all the progression that has occurred.*

Over the past 8 decades, technological advancement has taken an exponential growth trajectory, but progress in the field of drug discovery has remained relatively stagnant.

Cell phones weren’t even a thing, let alone Apple in 1928 (if you’re not an apple fan, we’re not friends — just kidding 😁), and now we not only have Apple, but we also have brain-computer interfaces, artificial intelligence bringing innovation into every industry, quantum computing (recently, *Google announced quantum supremacy*) and the emergence of tons more exponential technologies.But, in comparison, there is little rogression made in improving the process of drug discovery.

# **What does the current process of drug discovery look like?**

There’s tons of evidence which indicates that the current drug discovery process is just ***inadequate***, and ***shortcoming***for those in the most need — people living with chronic and deadly diseases.

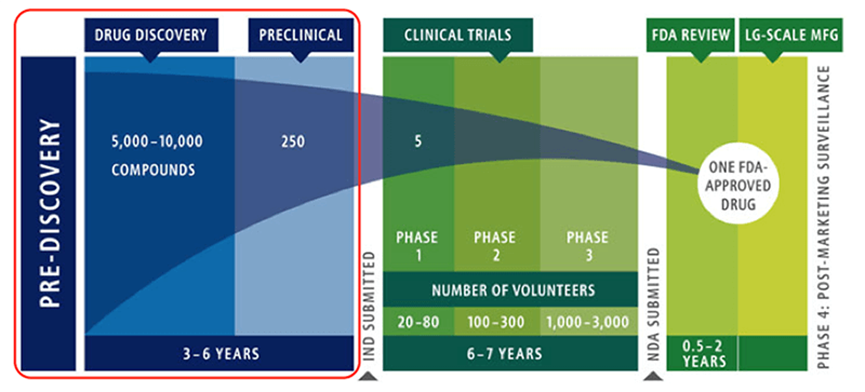
Taking a drug from research to market is estimated to cost on average **2.6 billion dollars, and more than 10+ years** 😱**.** Since drug discovery is so resource-consuming, most of the advancement lately has been focused on high-return markets like cancer research.

This partly explains **the lack of effective treatment for more than 90% of rare diseases.**

But why is this even the case? Let’s take a look at the drug discovery pipeline (aka the multifaceted process of drug development).

We all want these life-saving drugs to be in the hands of patients in need *quicker and cheaper.*

The simple answer as to why this process is so lengthy and costly is the sheer *complexity*.

The drug discovery pipeline looks something like this

Drug discovery pipeline! There are a total of 7 phases.

Complicated right? Let’s break it down…

There are seven phases in drug discovery:

## **1. Target identification: Discovery (2+ years)**

The first step isn’t even about the drug, it’s all about understanding the targets that are responsible for the disease. These targets often consist of DNA mutations, misfolded proteins, and other potential disease biomarkers. However, this is obviously not the case in the discovery of Penicillin, which was purely driven by serendipity.

**Issue #1:** Though the desired route is to identify targets, then develop drugs that specifically combat the disease targets, there is no guarantee in this process. The discovery of Penicillin clearly was an exception to this process, and there are many more exceptions. There are just so many possible compounds, and so many possible targets, it’s extremely difficult for human beings to make sense of all of these possible combinations.

## **2. Lead Discovery: Pre-Clinical (1–2+ years)**

This is the process of screening thousands of compounds designed to interfere with the disease targets. The objective is the significantly narrow the domain of the variety of potential compounds.

## **3. Medicinal Chemistry: Pre-Clinical (1–2+ years)**

This phase involves the process of further testing the compounds to analyze their interaction with disease targets. Some analysis that may be conducted includes, for instance, investigating the interaction of compounds taking into account their 3D configuration. In accordance with the results derived from analysis, compounds are further optimized towards the intended targets.

## **4. In Vitro Studies: Pre-Clinical (1-2+ years)**

As proof of concept, compounds that make it to this stage are tested in a cell system, an in-vitro model of the disease. This is the phase where the petri dish studies occur. In vitro studies attempt to scrutinize the effectiveness of the compound is at interfering with the target.

***Issue #2:*** *Results from in-vitro studies often don’t reflect results from animal or clinical studies, causing a* ***high failure rate.*** *According to an MIT study,* ***clinical trial success rate hovers just about14%, that’s insane!*** *This is because our in-vitro models of the cell system is often a gross simplification of our complex human system. Biology just doesn’t operate in 2D mode, but petri dish studies are 2D models of cells and is a widely used in-vitro study method.*

## **5. In Vivo Studies: Animal Studies (1–2+ years)**

After success from in-vitro studies (YAY, but the hard part has yet to come), the compound is typically tested in animal models like rat or mouse models. The results from animal studies are usually more representative compared to the 2D in-vitro cell culture models. However, it is significantly more expensive than in-vitro studies. The failure rate at this stage is also higher since the results from in-vitro studies don’t necessarily correlate with animal studies due to the disparity in the architecture of the cell model.

***Note to keep in mind:*** *What if we can fail earlier, instead of failing later? It doesn’t make sense to fail at such expensive phases, what if we fail during the phase of in-vitro studies?*

## **6. Finally, we arrive at Clinical Trials (6+ years)**

If the results from all the above phases showed promise in the compound, it then proceeds to clinical trials. There are three phases in clinical trials and each with a different objective. The main goal of clinical trials is to validate the efficacy and safety of the compound or potential drug in a human setting.

***Issue #3:*** *There are many regulatory aspects of clinical trials, which is why it’s the most lengthy and expensive phase. Proving the efficacy of drugs is often not an easy task; specifically, the long-term side effects of compounds usually remained unknown after a period of time. Testing new drugs on human beings always exists a significant inherent risk. The average cost of phase 3 clinical trial is estimated to be $19 million dollars.*

## **7. There’s one last Phase: FDA Approval and Commercialization (1+ years)**

Once all testing is completed, the compound can be submitted to be reviewed by the FDA for approval. Once approved, the drug can finally be able to be commercially available in the hands of patients to improve lives and treat diseases!!!!!! — The most exciting part of all the time, but this was an extremely long and costly journey!

***Issue #4:*** *Newly available drugs are often extremely expensive because of how costly research and development is. Companies often have a 20-year patent to protect their drug/product from the competition. This means that they can price their drugs at whatever price they think seems sound. Sometimes, this price can be at hundreds of thousands of dollars, making it unaffordable for the general public.*

**Note to keep in mind:** How can we accelerate the process of drug discovery?

Wow, that was a journey 🌄!

But what if there was a way to significantly accelerate the process of early phase drug discovery (all phases except clinical trials + commercialization) from 6 years to 6 days? I’m optimistic, and I don’t think this is sci-fi at all! In fact, this is POSSIBLE! We don’t lack the tools needed to realize this, **we just lack optimism and the right people.**

## **Current innovation: accelerating drug discovery**

* [Insilico Medicine](https://www.technologyreview.com/f/614251/an-ai-system-identified-a-potential-new-drug-in-just-46-days/) (AI + pharmaceutical startup) was able to design, synthesize and validate new drug candidates in just 46 days.
* [AlphaFold](https://deepmind.com/blog/article/alphafold) (Google’s AI algorithm) was able to predict the 3D structure of proteins (crucial assessment in drug discovery) at an unprecedented speed and accuracy, topping some of the world’s best biologists and researchers in this field.

People are the single most important factor in driving growth, not anything else. It doesn’t matter if you’re a 16-year-old/high schooler (spoiler: that’s me), or if you’re a renowned AI researcher, I believe there’s something for all of us to do in order to create impact. *And that’s also fundamentally why I decided to do something, as a curious 16-year-old.*

# **Introducing Synbiolic: Leveraging Variational Autoencoders for Drug Discovery**

We’ve been stuck with the same process of doing tons of wet-lab experiments and going through a bunch of trial and error for drug discovery for way too long! The manual process of drug discovery MUST change!

It’s time for something new,introducing ***Synbiolic*: extrapolating the potential of artificial intelligence, specifically variational auto-encoders (VAE) to generate novel & valid molecules in the format of SMILES** (Simplified molecular-input line-entry system).

## **How does Synbiolic’s magic work?**

1. *Uses the magic wand of* ***VAE*** *to generate molecules*
2. *Uses more magic to filter the list of generated molecules down to a few really good ones*

Here’s an *inside* look of what the magic is all about!

## **Beyond being an AI + nerdy buzzword, what is a variational autoencoder?**

Variational autoencoder (VAE) is a type of ***generative*** machine learning algorithm. **A generative model is a type of AI architecture that *generates new data* that is similar characteristically to the training data.**

It consists of two components which are neural networks: **(1) the encoder, and (2) the decoder.***Note: different neural networks can be used for the encoder and the decoder.*

**The encoder network is responsible for reducing the dimensions of the data that is inputted into the model, and the decoder network reverses the process by reconstructing the compact representation of the data back into its original dimensions/input.**

The compact representation of the data is termed the **latent space representation** (aka the bottleneck).

Autoencoder 💻

Synbiolic leverages VAE to generate new molecules that have drug-like properties similar to that of the molecules used to train the model. **The generated molecules are not exactly the same as the training molecules but are rather variations of the training molecules.**

To more intuitively understanding what the encoder and decoder models do, let’s play a game of charades!

**Let’s enter into programming mode and start by declaring some variables 🤖:**

* Lady on the right = encoder
* Gentlemen on the left = decoder
* Finding Nemo = latent space/compact representation of molecules
* The action = generated molecules

The lady guessing the phrase “Finding Nemo” *(\*compact representation of molecules\*)* is similar to the encoder since she condenses the “action” *(\*molecules\*)* into a phrase *(\*compact representation of molecules\*).* In terms of AI language, she is essentially constructing the latent representation of the “actions” *(\*molecules\*)*. This is an analogy to how the encoder reduces the dimensions used to represent the molecules into a more compact form.

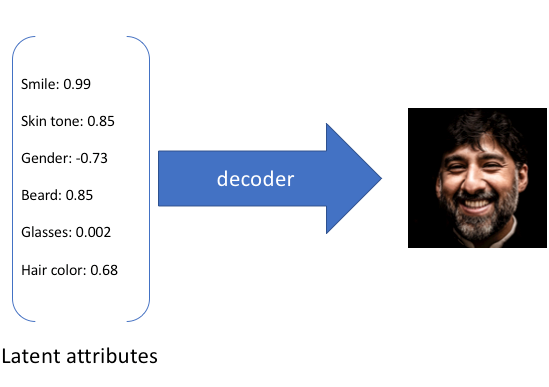
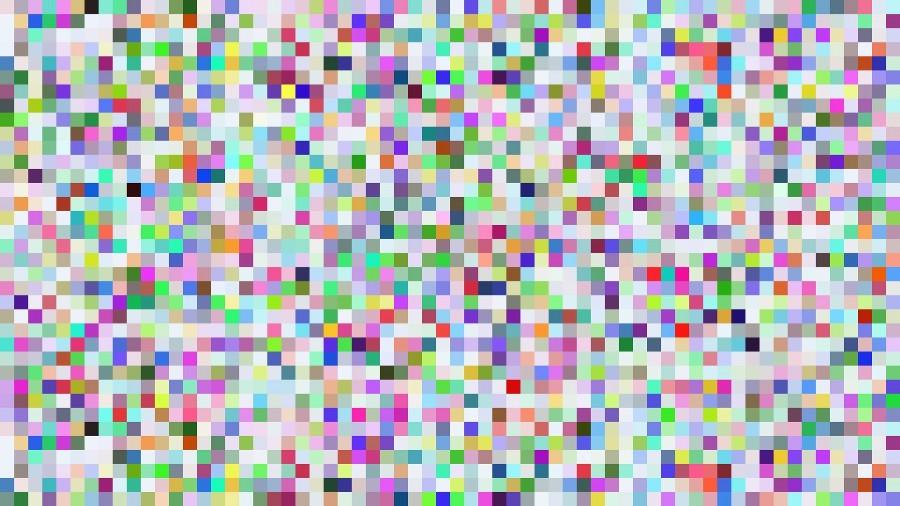
The gentlemen acting “Finding Nemo” *(\*compact representation of molecules\*)* is similar to the decoder since he attempts to reinterpret the phrase into actions *(\*molecules\*)*, which is the expanded representation. In terms of generating molecules, the decoder does this by reconstructing the molecule based on its condensed features, or the latent space representation.

## **But wait, what makes the variational autoencoder different from the vanilla autoencoder?**

A variational autoencoder has an encoder and decoder network just like a regular vanilla autoencoder pictured in the image above (the one before the finding Nemo). However, it’s not just a regular machine learning model, **VAE uses a probabilistic approach for its encoder network.**

The reason why VAE was used instead of autoencoders is mainly because **vanilla autoencoders can’t generate data**. This is the case since vanilla autoencoders (specifically the encoder) can’t seem to figure out a good approach that creates “usable” latent space representation to feed into the decoder network in order to generate “good” molecules.

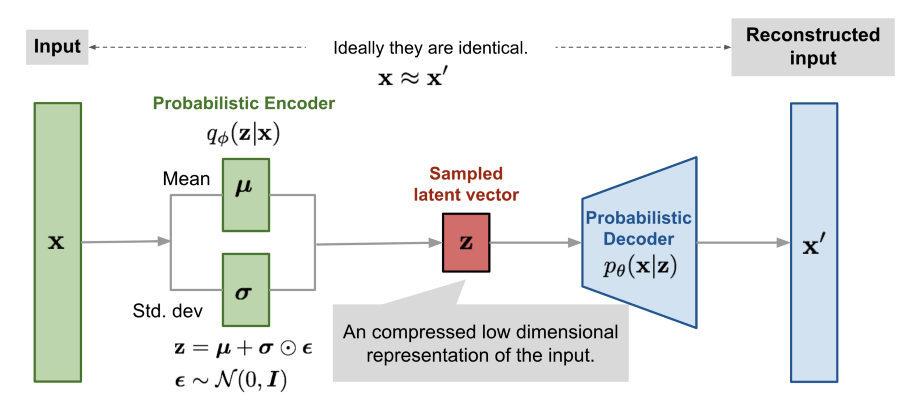
What ends up happening when autoencoders try to generate new data is that a random latent representation/vector is fed into the decoder model, which in turn generates *really funky and useless data*. This is because **vanilla autoencoders don’t know how to derive “good” latent vectors apart from forcing the encoder to create latent vectors from the training data.**

**The latent space of vanilla autoencoders is not continuous, which is why they’re no good for generating data** This means that taking the latent representation in the above picture, the decoder is able to generate a decent looking picture, but if one of the latent parameters like skin tone is altered just a bit from 0.85 to 0.84, then the decoder network gets completely screwed over and might end up generating something like this:

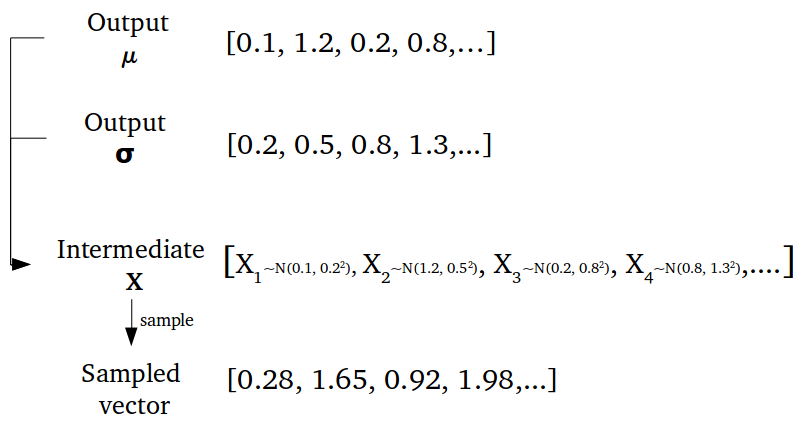
**This is all because the latent space representation isn’t continuous.**

It’s also not all so magical generating *random* data. What’s magical about VAE that autoencoders can’t achieve is that it’s one of the best generative models that can be used to **apply variations to the data you already have in *desired directions.***

The encoder network is denoted as q(z|x) and the decoder network is denoted as p(x|z) for x represents the input into the network and z is the latent representation (which is a vector).



What’s different about variational autoencoders is that its encoder network leverages a *probabilistic model*.This allows for the **latent space to be continuous.**

VAE does this by making its encoder assemble two vectors: **a vector which takes the mean, and another vector which takes the standard deviation of the input data.** Then the latent vector z is constructed by *sampling* from the vector of means and the vector of standard deviation.

By sampling, this allows the latent space to be continuous, instead of discrete. This is the reason why VAEs can generate synthetic data which essentially variations of the training data. (We can make apply these “variations” intentionally by optimizing the latent space to generate new data in with desired properties.)

Let’s move on to the *secret sauce* behind all of this, aka the models used for Synbiolic’s VAE.

**For the encoder and decoder models, the following neural networks are used:**

* Encoder network = convolutional neural network (CNN)
* Decoder network = type of recurrent neural network (RNN) known as Gated Recurrent Unit (GRU)

# **Understanding Convolutional Neural Networks: The Encoder**

Before we dive into CNNs, let’s first explore what dimensionality reduction actually is.

**Dimensionality reduction is the technique in machine learning used to downside the data by *reducing its features.***

There are **two techniques the encoder model can use to perform dimensionality reduction:**

1. By feature selection — selects only some features and excludes the rest
2. By feature extraction — transforms input data into a latent representation

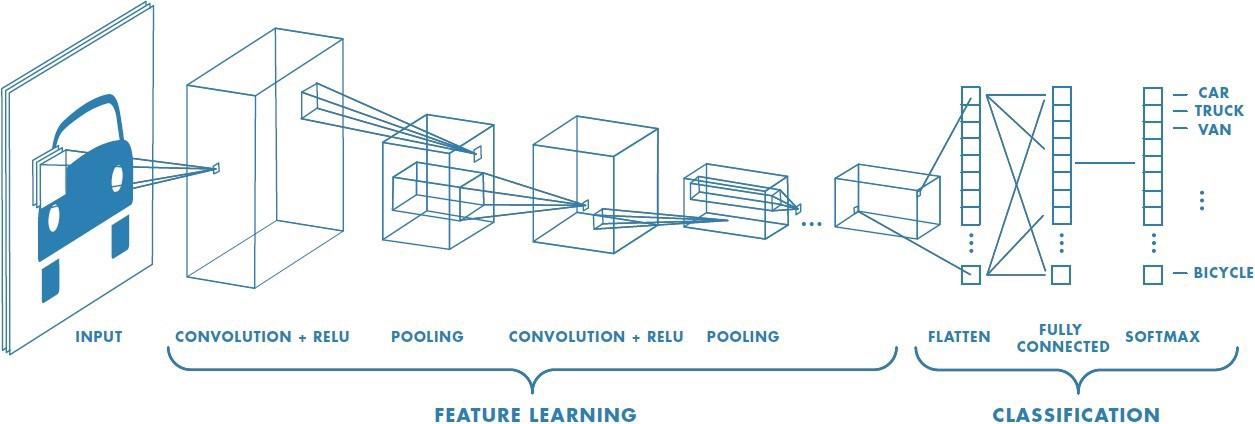
The difference between the two techniques is that feature selection *doesn’t* alter the features, while feature extraction does since it applies transformations to “input features/data” to arrive at the latent space representation.

Convolutional neural network (CNN) is a type of deep learning algorithm that is widely used in image classification because of its ability to detect features.

**They do this by implementing feature extraction, and therefore subsequently reducing the dimensions of the input data.**

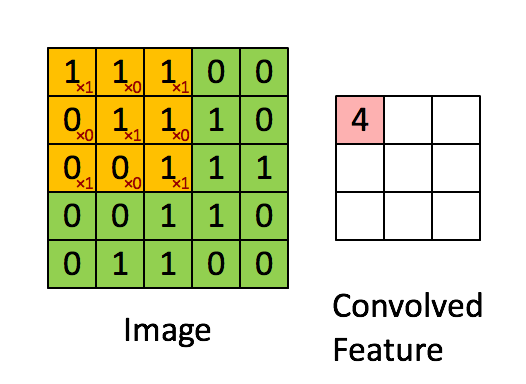
Like other deep learning algorithms, CNN are composed of **[1] input layers, [2] hidden layers, and [3] output layers.**

The hidden layers are made up of several different types of layers like convolutional layers, pooling layers, and fully connected (aka vanilla) layers.



Convolutional Neural Networks

What makes a “convolutional neural network” a “convolutional neural network” is the convolutional layer(s). Yup, it’s that simple, actually, maybe not that simple…

The convolutional layers essentially perform the dot product between the input and the filter to arrive at a lower-dimensional representation of the data (aka, the convolutional layers are powered by math).

**The yellow matrix is the filter**, the green matrix is the input image, and the pink matrix is the convolved feature or latent representation.

As you can see, the convolved feature is *dimensionally more compact* than the original input.

To learn more about convolutional neural networks and its applications in skin cancer detection, check [this article](https://medium.com/analytics-vidhya/teaching-machines-to-detect-skin-cancer-bd165566f0fe) out!

Instead of using CNN for image classification like for detecting skin cancer from images of moles, I trained the CNN on a dataset of molecules.

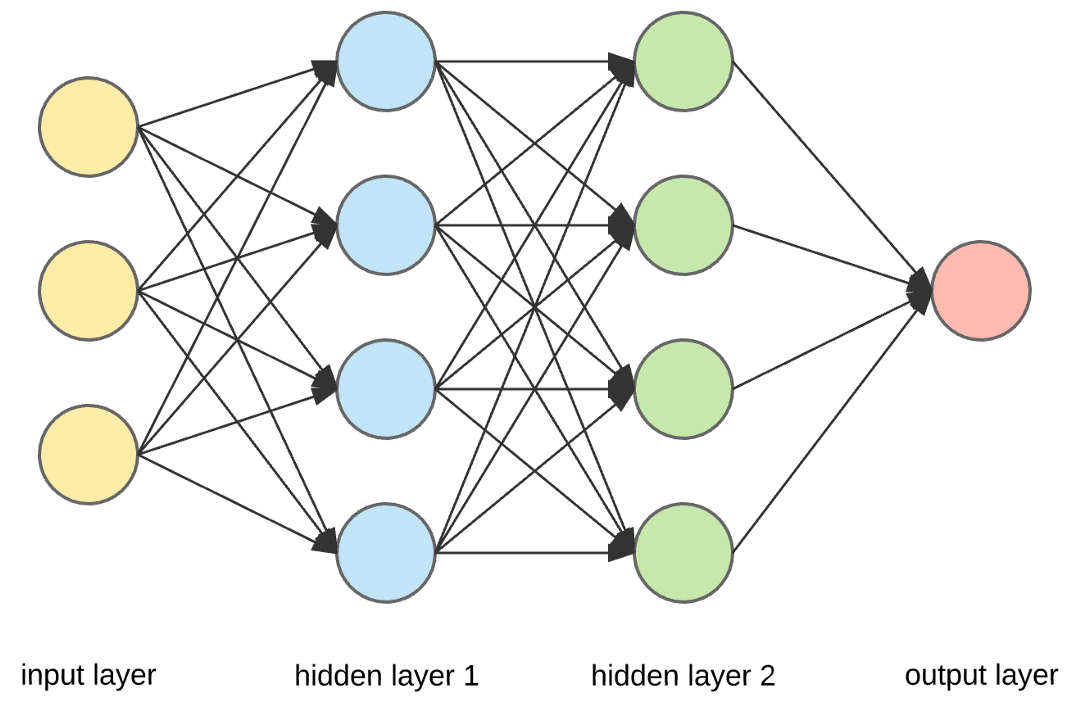
The VAE is trained on the [zinc dataset](https://zinc.docking.org/) containing 250 000 molecules. The encoder (CNN) transmutes the representation of the features of the molecules into the more compact representation.

# **What about the Decoder?**

For the decoder, a specific type of recurrent neural network (RNN) known as gated recurrent unit (GRU) is used.

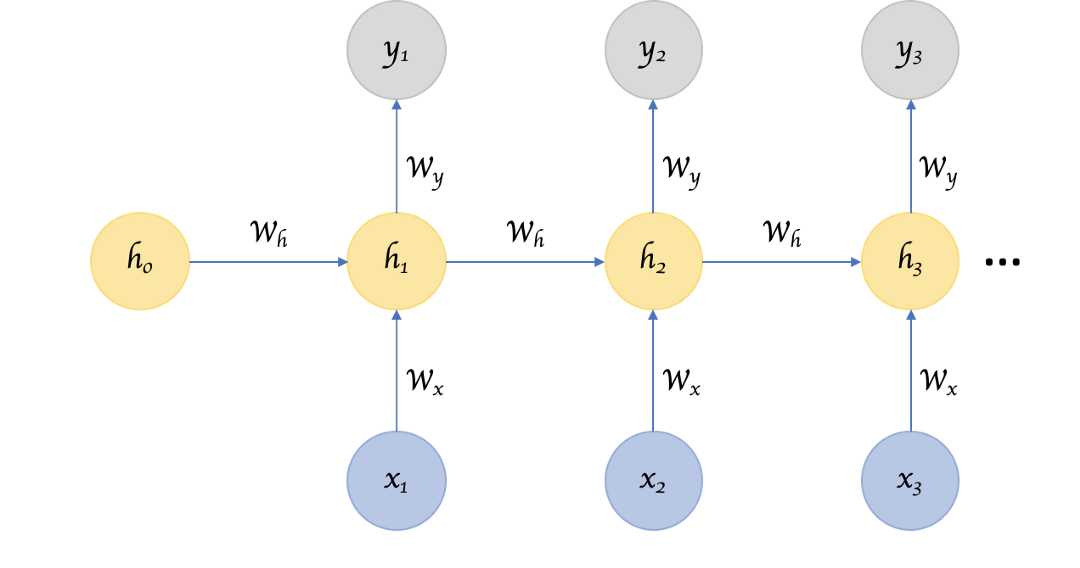
Unlike other types of neural networks, **RNNs have an internal memory capacity.** RNNs are typically applied to **sequential data** since their special property is that their decisions are influenced by the previous inputs and outputs ***within* the network.**

Vanilla neural networks also have the ability to “remember” things, but they are limited to remembering things from ***previous* training iterations.**

****

“Vanilla” Neural Network

In contrast, RNNs remember things within each training iteration and make their decisions based on the inputs and generated outputs from the previous hidden state.



x nodes are the inputs, y nodes are the outputs, and h nodes are the hidden nodes

**Think of it like this:** remember how your math teacher used to tell you how if you don’t know how to add, subtract, multiply and divide numbers, you will never be able to master calculus! That’s because as human beings, **we understand concepts based on the previous concepts we’ve learned.**

This is similar to how as you are reading this sentence, you understand it based on your understanding of the previous words. If I only wrote “previous words” as the sentence, you wouldn’t be able to understand what I was trying to say. Essentially, this is what RNNs are really good at doing: dissecting the meaning of sequential data like sentence structures.

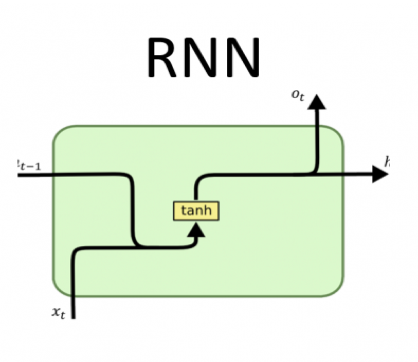
RNNs internal memory capacity refers to its ability to remember previous information within the network to help it make decisions. They have the ability to predict the next value in sequential data by understanding the connection between the previous values in the sequence.

All machine learning is powered by math, and the math of RNNs consists of hidden states (h), which are the hidden nodes.

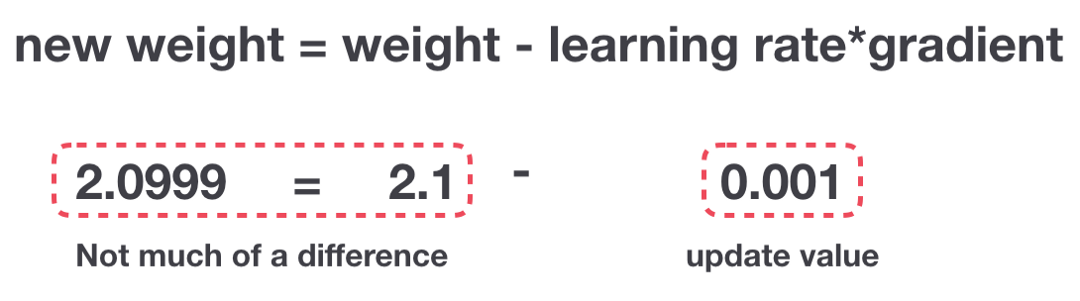
The hidden states are connected to each other. In each hidden states, there are gates, which are neural networks that control the transfer of information.

For the hidden state to compute the information, it has to be pre-processed and transformed into a matrix/vector representation.

The hidden state of “vanilla” RNNs look something like this:

* The output from the previous hidden state combines with the current input to form a vector which includes information about the current and previous inputs.
* This vector is passed through a tanh activation function which restricts the range to values between -1 and 1 to produce the new hidden state.

However, traditional RNN suffers from an issue known as vanishing gradient descent.



What happens when training a traditional RNN is that as the weight is being updated, the gradient becomes so small that it renders the updated weight insignificant. If the original weight has practically remained unchanged, this means that there is *little to no learning* occurring within this model.

A small gradient update doesn’t result in any significant learning, thus preventing the model from doing its job.

In response to this issue, I implemented the gated recurrent unit (RNN, but with a twist) for the decoder instead of using “traditional” RNN.

There are two gates involved in the gated recurrent unit (GRU): **the update gate and the reset gate.** In essence, gates are just *neural networks* within the hidden nodes.

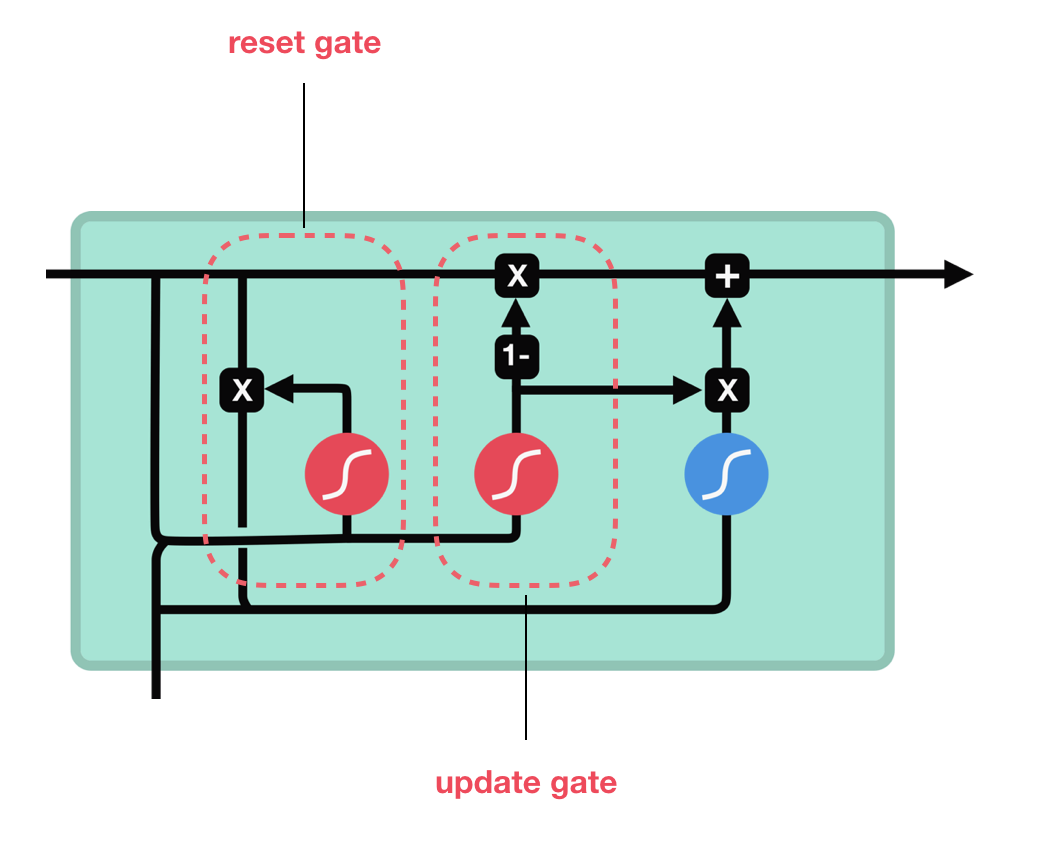
Instead of remembering every single word in a sentence, unlike RNN, **GRU only remembers significant phrases and words.**

For example, let’s take a look at this sentence:

“To examine the question of how to best accelerate rare diseases research, we will review some of the challenges in rare diseases research and opportunities for therapy development that recent scientific advances are presenting.”

Words like “to”, “the”, “of”, “that”, and “for” are not really relevant to the meaning of the sentence, and thus the GRU would forget those words.

“To examine the question of how to best **accelerate rare diseases research**, we will review some of **the challenges** in **rare diseases research** and **opportunities** for **therapy development** that recent **scientific advances** are presenting.”

While the RNN may attempt to memorize the entire sentence, GRU only memorizes the significant terms (which are bolded above)

Gated Recurrent Unit

## **Update Gate**

The update gate’s responsibility is to determine how much of the information from the previous hidden states needs to be kept. **Vanishing gradient descent problem is resolved by passing through this gate since the model gets to decide how much of the past information it needs to pass to the future hidden states.**

## **Reset Gate**

The output from the previous hidden state and the current input combine to form a vector which includes information about the current input and the previous inputs. Each vector contains not only information in the current hidden state, but also previous hidden states.

Reset gate’s job is to decide what past information should the model forget. Essentially, the functionality of this gate is opposite that of the update gate.

In the scenario of generating molecules, the input for the GRU or the decoder is a vector representing the compressed form or latent space representation of the molecules. The final output of the GRU is the reconstruction of the molecule in its original/expanded representation.

The quantity which measures the drug-likeness of a molecule is called QED (quantitative estimation of drug-likeness). I obtained the QED of the generated molecules to validate that they are drug-like molecules.

All generated molecules with a QED below 0.5 are disregarded, this ensures the quality of the generated molecules. The molecules with the highest QED values can be synthesized to further validate their properties through in-vitro and in-vivo studies!

YAY! We were able to successfully design and validate molecules! This same process which is classified under early phase drug design would’ve taken at least 3 years of traditional research to arrive at!

If you’re interested, feel free to check out the source code from [Github here](https://github.com/joeym-09/Leveraging-VAE-to-generate-molecules).

This was possible thanks to my wonderful team: [Aryan Misra](https://medium.com/u/bd6ad9886982?source=post_page-----8b2a64333e07----------------------) & [Elias Kountouris](https://medium.com/u/b295dac7f56?source=post_page-----8b2a64333e07----------------------)!

## **Key Takeaways:**

* Drug discovery is crazy expensive and lengthy
* VAE is a type of generative neural network that can be used to generate molecules for drug discovery
* Artificial intelligence is a powerful tool that can be used to revolutionize healthcare