Master Thesis Proposal

Elias Arnold

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Biomedical Data Analysis

Examiner: Prof. Dr. Volker Roth
Supervision: Mario Wieser and Vitali Nesterov

1 Introduction

The need of our society for new materials is steadily increasing. As the demand for it grows, the research should be as resource-efficient as possible [(Kim et al. 2018)]. Conventional approaches to molecule generation are no longer sufficient, especially for the more complex and specialised materials. The methods do not cope with the enormous variety of possible molecules. Even for small organic molecules, the number of stable compounds is estimated to be bigger than 10^{60} (Kirkpatrick & Ellis 2004). To find new molecules, conventional methods scan a database of already known compounds and try to find molecules which best match the desired properties. Such conventional search methods are computationally very exhausting (Nesterov et al. 2019). In the next step, the selected molecules are further tested and evaluated, until the most suitable is determined. This can be achieved by solving Schrödingers equation or by making very expensive and time-consuming experiments. Solving the problem algorithmically would take a lot of time and money because such algorithms would scale very poorly.

However, the availability of large databases of known molecules has led to a new approach. Instead of finding new compounds with exhaustive search approaches, machine learning models take over the task. In recent years, a large number of publications have been made in this area. A very popular model architecture for molecule generation is the Variational Autoencoder (VAE) (Kingma & Welling 2013). The main idea of a VAE is to encode the molecule into a low-dimensional representation and decode a corresponding sample from the latent distribution to the original molecule. Once both, the encoder and the decoder, are trained, one can explore the latent distribution by sampling and decoding molecules from it. However, there are also significant challenges with this approach. For example, the representation or the chemical validity of generated molecules are two very important aspects of any work that deals with VAE architecture.

At first glance, a handy choice would be the widely used and text-based Simplified Molecular-Input Line-Entry Specification (SMILES) representation of molecules (Gómez-Bombarelli et al. 2018). In fact, SMILES-strings are less suitable, since one SMILES string can produce various molecule configurations, and it is a challenge to produce semantically valid SMILES strings. Also, SMILES do not contain important 3-D information. Therefore, newer approaches use tensor based representations of the molecular graph as an input. These representations better reflects the networked structure of graphs. However, there are several drawbacks. Although graph-structured data is very common, it is not so easy to learn from such data. The challenge is to handle the arbitrary connectivity of the nodes. Unlike images, where all neighbouring pixels have a certain relation, there is no clear way how to represent a graph (Simonovsky & Komodakis 2018).

Another difficulty when generating new molecules is the chemical validity. In order to ensure a valid outcome, some chemical constraints must be applied (Samanta et al. 2018). Since a model is not capable of learning the underlying rules reliably, those are enforced during the sampling process. The research and testing of automatic methods for molecule generation are still young. However, there is tremendous potential and a lot of uses for such methods.

Molecule generation is not an easy task. Conventional methods are no longer able to cope with the growing demand for new materials. That is why people have begun to search for new molecules with machine learning methods. This is associated with difficulties in the representation of new molecules. In the next section, we will present the VAE, which is the core concept for these machine learning methods, and CGVAE (Liu et al. 2018), a real-world implementation of a VAE.

2 Methods

In this section, we introduce two main concepts that we will need in our thesis. First, we will present the generative model called variational autoencoder. Second, we will show an actual implementation of it.

2.1 Variational Autoencoder

Unfortunately, an autoencoder can not be used as a data generator. We might want to sample random latent representations and feed them into the decoder, to produce custom, unobserved x'. This is not possible, since the distribution from which we would like to sample from, is unknown in an autoencoder. Therefore, we introduce the concept of variational autoencoders.

To generate unseen data, we have to deal with inference in graphical models. Let us assume that the visible output $\mathbf{x}'$ depends on a latent variable $\mathbf{z}$, which is hidden from our view ($\mathbf{z}$ corresponds to the low-dimensional representation of the input $\mathbf{x}$). In order to get the distribution from which we want to sample, we need to compute $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$. This is intractable, because the marginal density $p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz$ would require us to evaluate all possible configurations of z (the latent variable). There exist two main strategies to approximate $p_\theta(z|x)$:

- Monte-Carlo methods
- Variational inference

For our setting, the first approach would be too costly and time expensive, especially if the parameters get updated for each single data point on a large dataset (Kingma & Welling 2013). Therefore, the variational inference is the key to solve this problem. Variational inference approximates $p_\theta(z|x)$ by a tractable distribution $q_\phi(z|x)$, where ϕ denotes the parameters of q, that have to be learned. The graphical model can be seen in Figure 1.

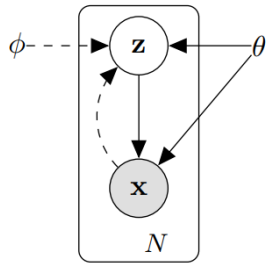


Fig. 1: The graphical model where x represents the observed variable and z is the latent variable, on which x depends on. The ground truth generative model parameters θ and the variable z are hidden to us. With variational inference, the variational parameters ϕ are searched to approximate a known distribution $q_\phi(z|x)$ of the sought distribution $p_\theta(z|x)$. [1]

Our goal is to find $q^*_\phi(z|x)$, which is as similar as possible to $p_\theta(z|x)$. The distance between two probability distributions can be determined with the Kullback-Leibler divergence (KL). So, our goal is to compute the following term:

$$q^*_\phi(z|x) = \arg\min_{q_\phi(z|x)} KL(q_\phi(z|x)||p_\theta(z|x)).$$

[1] Image source: Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

After substituting the KL divergence with its definition and some term transformations, we end up with the following objective function:

$$log(p_\theta(x)) - KL(q_\phi(z|x)||p_\theta(z|x)) = \mathbb{E}_{z\sim q_\phi(z|x)}[log(p_\theta(x|z))] - KL(q_\phi(z|x)||p_\theta(z)).$$

The goal is to find the distribution $q_\phi(z|x)$ that minimises $KL(q_\phi(z|x)||p_\theta(z|x))$. However, this unknown term also appears in the objective function. We can not compute the KL term, because both the distribution $p_\theta(z|x)$ is unknown to us, and it is not necessarily of the same type as $q_\phi(z|x)$. Accordingly, equality can not be guaranteed. Therefore, the objective function can be rewritten as an inequality:

$$log(p_\theta(x)) \geq \underbrace{\mathbb{E}_{q_\phi(z|x)}[log(p_\theta(x|z))] - KL(q_\phi(z|x)||p_\theta(z))}_{\text{lower bound } \mathcal{L}}.$$

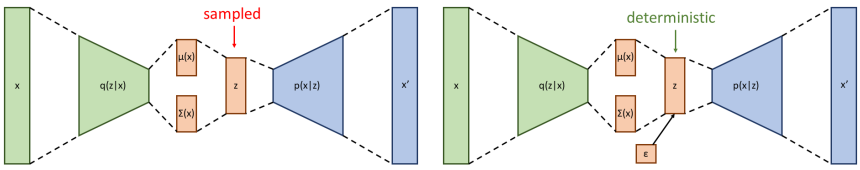
Since the KL-divergence will never be negative (by Jensen's inequality), maximizing of the lower bound corresponds to minimizing the KL-divergence. We have now found a solvable sub-problem whose maximum is at the same time the minimum of the actual problem. In the literature, the lower bound is often called $\mathcal{L}(\theta, \phi, x)$ or ELBO, short for Evidence Lower Bound. To optimize the lower bound, the most commonly used algorithms are based on a mean-field approximation such as CAVI, short for coordinate ascent variational inference [Blei et al. (2017)]. Such algorithms assume that $q_\phi(z|x)$ is a product of mutually independent factors, where each can be optimized independently.

The task of maximizing the first term (expectation) could be considered as a maximum likelihood estimation. The second term acts as a regulariser and requires $p_\theta(z|x)$ to be close to $q_\phi(z|x)$ [Doersch (2016)]. On the one hand, we want to maximize the reconstruction probability of x, on the other hand, we want the distribution $q_\phi(z|x)$ to be tractable.

To train a variational autoencoder, the negative of the lower bound (-ELBO) acts as the loss function. The encoder ($q_\phi(z|x)$) should be designed in such a way, that it learns the parameters θ of the latent space distribution (in the case of a Gaussian, these are μ and σ). In order to bring $q_\phi(z|x)$ in a tractable form, we substitute $p_\theta(z)$ with a simple Gaussian prior (Kingma & Welling 2013).

In order to get a z in the latent space, it has to be sampled (see Figure 2). The problem is, that we cannot run backpropagation on a stochastic node. The solution is called reparameterisation trick, where we decouple the sample from the parameterisation of the normal distribution whose parameters we are learning. Instead of sampling the z directly, we first compute a $\epsilon \sim \mathcal{N}(0, 1)$ and then

compute $z = \mu(x) + \Sigma^{\frac{1}{2}}(x) * \epsilon$. Now, given a fixed x and a sample ϵ this function is deterministic and continuous in the parameters ϕ and θ (Doersch 2016).



(a) without reparametrization trick **(b)** with reparametrization trick

Fig. 2: Illustration of the reparametrisation trick. **(a)** If the encoder of the VAE is forced to learn parameters for $q_\phi(z|x)$, it requires to draw a sample from that distribution to get a latent representation z. The issue is, that such a stochastic layer is not-differentiable and the VAE can not be trained with backpropagation. **(b)** Therefore, we reparametrize the sample z, such that the noise is independent of the parameters. In both fubfigures, backpropagation takes only place between the dashed lines.

The reparameterisation trick allows us to propagate the error back from x' to x without going through ϵ.

2.2 Constrained Graph Variational Autoencoder

The Constrained Graph Variational Autoencoder (CGVAE) (Liu et al. 2018) uses a Gated Graph Neural Netowork (GGNN) (Li et al. 2015) for encoding and decoding purpose. The decoder assembles new molecules in a sequential way. They use a pool of N nodes , which correspond to atoms, in the latent space and sequentially add a node to the molecule. Some rules for correlation of two atoms are learned by the VAE, the remaining chemical rules are enforced by masking. Every time a new atom is added to the molecule, they perform a full decoding step on the partial graph, which is the key to their good results. So, to make a graph $\mathcal{G}$ in $t+1$ steps, t partial graphs are generated $(\mathcal{G}^{(0)}, \mathcal{G}^{(1)}, ..., \mathcal{G}^{(t-1)})$.

However, by choosing to build the molecule sequentially, they lose permutation symmetry. To compute the likelihood of the graph, the full generation history has to be marginalized out. The sequential molecule generation model of (Li et al. 2018) conditions on the full history of the generation trace $log(\mathcal{G}|\mathcal{G}^{(0)}, \mathcal{G}^{(1)}, ..., \mathcal{G}^{(t-1)})$. Nonetheless, they mention a variety of problems like overfitting, stability, and scalability with that approach. CGVAE sidesteps these issues by condition only on the current partial graph, and not on the full generation sequence.

New molecules should be optimized with respect to desired target properties. To train the VAE, they construct the loss function in such a way, that it concentrates the target property areas in the latent space's high probability area. This ensures that new molecules can be optimized for a chemical property by gradient ascent in the latent space.

3 Thesis Objectives

Our work will be split up in three milestones. In the first milestone, we aim to predict the chemical properties of molecules. The generation of the molecular graph is the goal of the second milestone. Milestone three combines both approaches and aims to generate molecules with given properties.

3.1 First Milestone

In the first step, we seek to predict molecule properties from from the corresponding latent representation. To achieve this, we will train a regression model. It should be checked whether the encoder of CGVAE can represent a graph as a latent vector, from which the molecule properties can be learned. The promising results of CGVAE got us to choose their encoder to map molecular graphs to latent vectors.

3.2 Second Milestone

The second goal is to build and train a VAE model for molecule generation. The representation of the molecule in latent space should be exactly the same as in CGVAE. The goal is that we can then sample a new latent vector with CGVAE and turn it into a molecule with our decoder. The main challenge will be to implement CGVAE's sequential decoder.

3.3 Third Milestone

Once the prediction of molecule properties and the generation of new graphs is implemented, we will combine both ideas and generate molecular graphs with some desired properties. This requires a structuring of the latent space. This can be achieved by adjusting the loss function with an additional term, which depends on the molecule's properties. Molecules with similar properties are then forced to be close together in latent space.

Bibliography

Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. (2017), 'Variational inference: A review for statisticians', *Journal of the American Statistical Association* **112**(518), 859–877.

Doersch, C. (2016), 'Tutorial on variational autoencoders', *arXiv preprint arXiv:1606.05908* .

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P. & Aspuru-Guzik, A. (2018), 'Automatic chemical design using a data-driven continuous representation of molecules', *ACS central science* **4**(2), 268–276.

Kim, K., Kang, S., Yoo, J., Kwon, Y., Nam, Y., Lee, D., Kim, I., Choi, Y.-S., Jung, Y., Kim, S. et al. (2018), 'Deep-learning-based inverse design model for intelligent discovery of organic molecules', *npj Computational Materials* **4**(1), 67.

Kingma, D. P. & Welling, M. (2013), 'Auto-encoding variational bayes', *arXiv preprint arXiv:1312.6114* .

Kirkpatrick, P. & Ellis, C. (2004), 'Chemical space'.

Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. (2015), 'Gated graph sequence neural networks', *arXiv preprint arXiv:1511.05493* .

Li, Y., Vinyals, O., Dyer, C., Pascanu, R. & Battaglia, P. (2018), 'Learning deep generative models of graphs', *arXiv preprint arXiv:1803.03324* .

Liu, Q., Allamanis, M., Brockschmidt, M. & Gaunt, A. L. (2018), 'Constrained graph variational autoencoders for molecule design', *CoRR* **abs/1805.09076**.
URL: *http://arxiv.org/abs/1805.09076*

Nesterov, V., Wieser, M., Wieczorek, A. & Roth, V. (2019), Learning deep representationsfor molecule design, Master's thesis, University of Basel, Department of Mathematics and Computer Science - Biomedical Data Analysis.

Samanta, B., De, A., Ganguly, N. & Gomez-Rodriguez, M. (2018), 'Designing random graph models using variational autoencoders with applications to chemical design', *CoRR* **abs/1802.05283**.
URL: *http://arxiv.org/abs/1802.05283*

Simonovsky, M. & Komodakis, N. (2018), 'Graphvae: Towards generation of small graphs using variational autoencoders', *CoRR* **abs/1802.03480**.
URL: *http://arxiv.org/abs/1802.03480*