

---

# Embedding and generation of indoor climbing routes with variational autoencoder

Kin Ho Lo<sup>1</sup>

**1** [kin.ho.lo@cern.ch](mailto:kin.ho.lo@cern.ch)

## Abstract

Recent increase in popularity of indoor climbing allows possible applications of deep learning algorithms to classify and generate climbing routes. In this work, we employ a variational autoencoder to climbing routes in a standardized training apparatus MoonBoard, a well-known training tool within the climbing community. By sampling the encoded latent space, it is observed that the algorithm can generate high quality climbing routes. 22 generated problems are uploaded to the Moonboard app for user review. This algorithm could serve as a first step to facilitate indoor climbing route setting.

## 1 Introduction

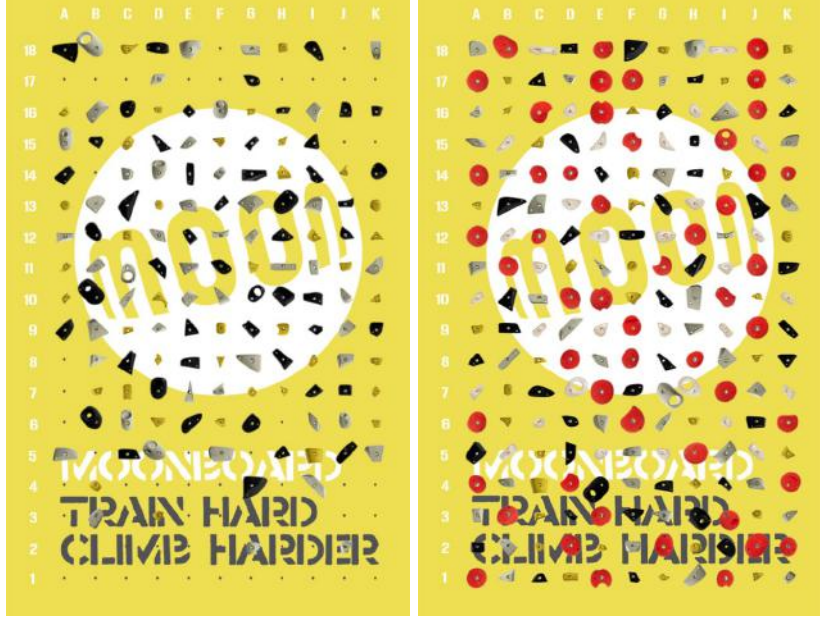
Given increasing popularity of indoor climbing, we created a deep autoencoder to generate climbing routes (route setting) with a worldwide training apparatus Moonboard. One of the mottos of the apparatus is to set the standard for indoor training and over the years it has become globally recognised as one of the most effective strength training tools for rock climbing. Unlike many indoor climbing gyms, its standardized setup and well-developed user platform provide an unique opportunity for prototyping and testing various supervised machine learning algorithms for route setting. Route setting is an important area in indoor climbing, and very often it challenges route settings' imaginations. Although fulfilling, route setting could be tedious at times. This work serves as a first step for wide applications of machine learning (ML) to facilitate or speed up the route setting.

Existing works for indoor climbing focus primarily on classification of route difficulties [1, 2, 11, 13], with or without using deep learning algorithms. Inspired by techniques from natural language processing [6], just as words constitute sentences, indoor climbing problems are composed of holds, route setting has been recently investigated with long short-term memory (LSTM) with sequential predictions based on previous climbing holds [3]. Here we adopted a different approach of embedding each climbing problem into a latent space to allow maximal freedom for the algorithm to learn correlations between each climbing hold.

The paper is organized as follows. Section 3 gives an overview of the algorithm. Section 4 describes the results obtained from the training of the algorithm.

## 2 Moonboard

The Moonboard website hosts a database of Moonboard climbing routes for each Moonboard configuration. Routes are represented graphically by a Moonboard image with holds selected by circles. Various Moonboard setups are shown in Figure 1. We consider the 2017 configuration, which consists of 198 holds.



**Figure 1.** Moonboard setup 2016 (left), Moonboard setup 2017 (right).

### 3 Method

Autoencoders, originally introduced in [12], aims to transform inputs to outputs with least amount of deformations. More recently, autoencoders have been used in combination with deep learning algorithms [5, 7–9]. We formulated our problem in terms of a variational autoencoder [10].

An autoencoder [4, 14] is a mapping  $f$  that has been optimized to approximate the identity on a set of training data  $x$  with the dimension of input features as  $N_d$ . It consists of a dimensional reducing mapping (encoder)  $g$  followed by a dimension increasing mapping (decoder)  $h$ . Each problem is represented as a binary vector  $x \in \mathbb{Z}_2^{N_d}$ ,  $N_d = 198$ , with each occupied hold represented as 1 and 0 otherwise.

$$f = h \circ g: x \in \mathbb{Z}_2^{N_d} \rightarrow x_{\text{out}} \in \mathbb{R}^{N_d} \quad (1)$$

where  $g$  and  $h$  are defined respectively as

$$g: x \in \mathbb{Z}_2^{N_d} \rightarrow v \in \mathbb{R}^{N_\ell} \quad (2)$$

$$h: v \in \mathbb{R}^{N_\ell} \rightarrow x_{\text{out}} \in \mathbb{R}^{N_d} \quad (3)$$

where  $N_\ell$  is the dimension of the latent space and  $N_\ell = 16$  in actual training steps. Parameters of  $g$  and  $h$  are obtained by solving the optimization problem

$$\min \sum_k^n \left( \sum_i^{N_d} L_{\text{binary}}(x_{i,k}, f_{i,k}(x; \theta_g, \theta_h)) + L_{\text{MSE}} \left( \sum_i^{N_d} x_{i,k}, \sum_i^{N_d} f_{i,k}(x; \theta_g, \theta_h) \right) + \sum_i^{N_\ell} L_{\text{KL}}(v_{i,k}) \right) \quad (4)$$

where  $n$  is the total number of input data. The first term encodes the reconstruction loss of the algorithm. The second term constrains the number of occupied holds in the problem and the third term is the Kullback–Leibler divergence.

---

## 4 Result

In this work, we adopted the 2017 Moonboard setup with 198 holds. The dimension of the latent space is 16. The variational autoencoder is trained with a single GPU (GeForce RTX 2080) for 2000 epochs with batch size of 512, using a training sample with 16979 Moonboard problems and a testing sample with 1886 problems.

Each sample is formed by randomly selecting the existing Moonboard problems to ensure that problems with various style and difficulty could be considered in the algorithm. After training, 50 routes are generated by randomly sampling the latent space and selecting the holds with the highest probabilities according to the length indicated. All problems generated are required to have at least 6 holds. We observed that this requirement selects 27 out of 50 problems. 22 moonboard problems are generated and uploaded to the moonboard mobile application for user review. Out of the remaining 5 problems, 1 problem coincides with an existing problem in the Moonboard database and 4 problems are failed attempts, as described below. Some interesting features could be observed in the generated problems suggesting that the algorithm have picked up special features of the board and the climbing routes:

- Every Moonboard climbing route has one or two finishing holds at the top row of the board. Most of generated problems exhibit this feature.
- Some generated problems used only a specific type of holds in the board, as shown in Figure 2.
- There are much more problems in lower grade of difficulty. Therefore, many generated problems have relatively good holds in the board, as shown in Figure 3.

Despite the fact that many problems are generated with good quality, 4 generated problems, shown in Figure 4 remain "unclimbable". These observations are highlighted below:

- 1 problem is generated such that the finishing hold is almost unreachable from the rest of the holds.
- 2 problem are generated with missing finishing holds.
- 1 problem is generated with missing starting holds. Starting holds are required to be below row 7 on the board. Despite this caveat, the problem can still be climbed.

## 5 Conclusions and future work

In this work, we presented a simple application of variational autoencoder to the generation of indoor climbing routes. We found that the simple and well-established algorithm achieved reasonable performance and can generate climbing routes with high quality. Given more time, this approach could be applied to the problems with other board setup, e.g. Moonboard 2016 and Kilterboard. In the future, various improvements could be incorporated by, but not limited to, conditioning the algorithm with the user-specified route difficulty, climbing style, and enhancing the generation efficiency.

## References

1. Graph neural networks in classifying rock climbing difficulties. [http://cs230.stanford.edu/projects\\_winter\\_2020/reports/32175834.pdf](http://cs230.stanford.edu/projects_winter_2020/reports/32175834.pdf).

- 
2. Machine learning methods for climbing route classification.  
<http://cs229.stanford.edu/proj2017/final-reports/5232206.pdf>.
  3. Moonboard climbing.  
<https://github.com/andrew-houghton/moon-board-climbing>.
  4. Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, and U. Montreal. Greedy layer-wise training of deep networks. volume 19, 01 2007.
  5. Y. Bengio and Y. Lecun. *Scaling learning algorithms towards AI*. MIT Press, 2007.
  6. D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder, 2018.
  7. D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, 11:625–660, Mar. 2010.
  8. G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.
  9. G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
  10. D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2013.
  11. C. Phillips, L. Becker, and E. Bradley. strange beta: An assistance system for indoor rock climbing route setting. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013130, Mar 2012.
  12. D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, USA, 1986.
  13. D. Scarff. Estimation of climbing route difficulty using whole-history rating, 2020.
  14. M. Scholz, M. Fraunholz, and J. Selbig. Nonlinear principal component analysis: Neural network models and applications. In A. N. Gorban, B. Kégl, D. C. Wunsch, and A. Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 44–67, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

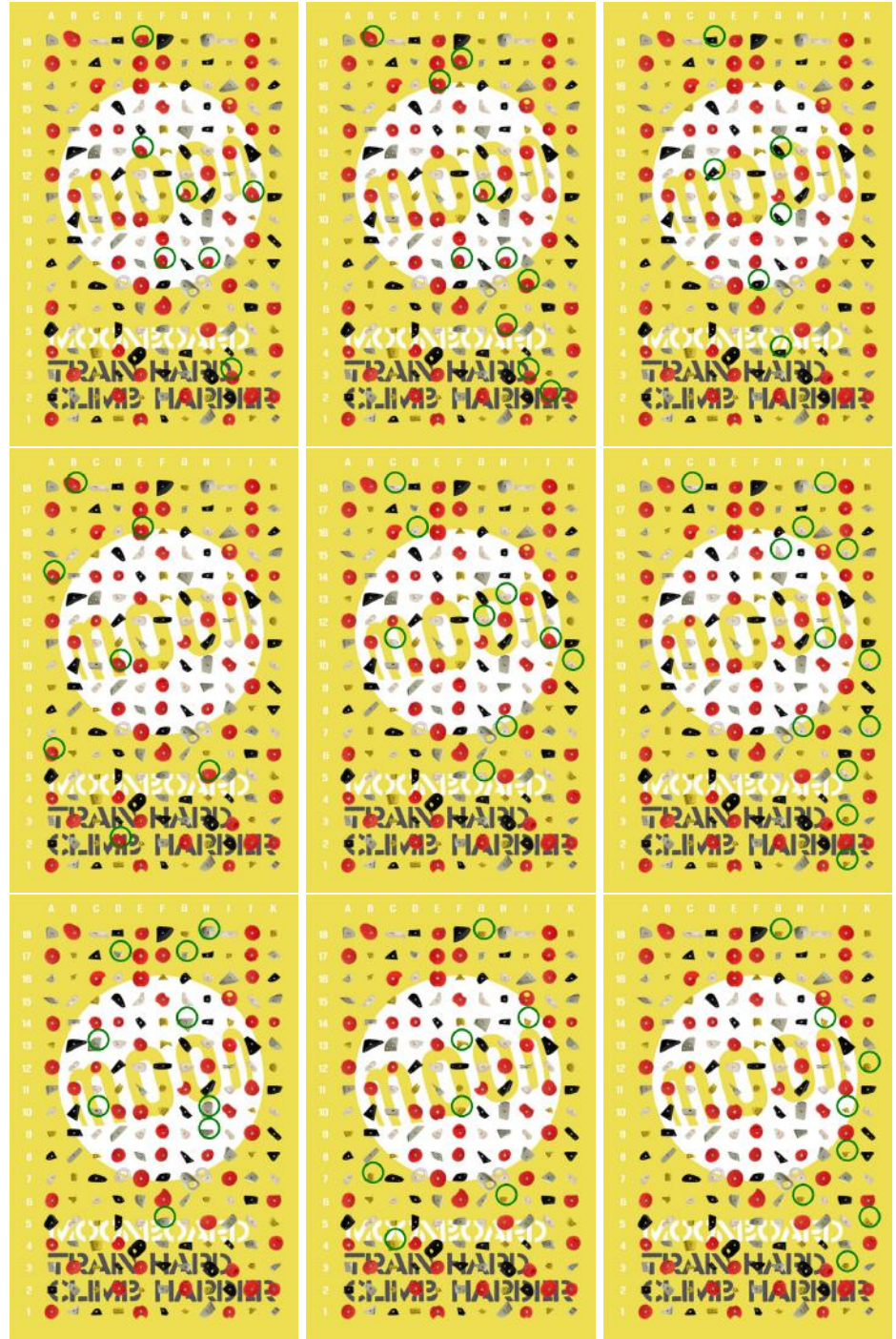


Figure 2. Examples of generated problems using only certain type of holds in the board.



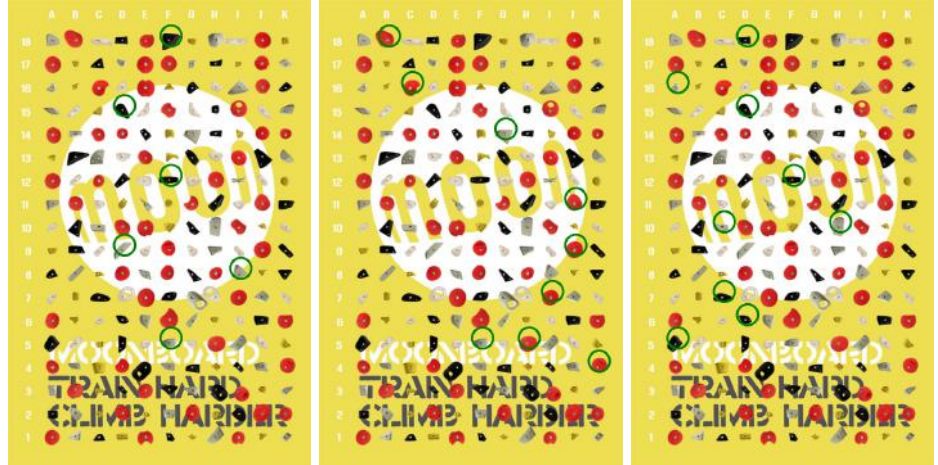


Figure 3. Examples of generated problems using relatively good holds in the board.

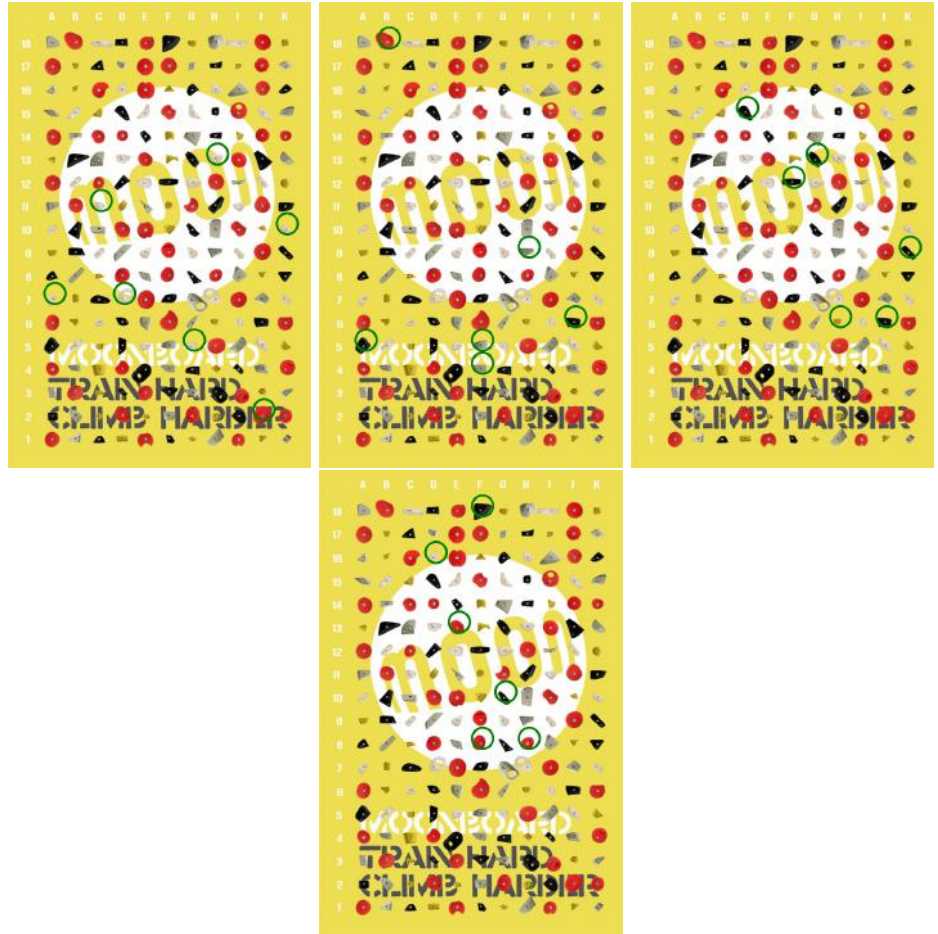


Figure 4. Examples of generated problems with bad quality.