

# Machine Learning and its application to Protein Engineering

Mohsen Sadeghi  
[mohsen.sadeghi@fu-berlin.de](mailto:mohsen.sadeghi@fu-berlin.de)

Department of Mathematics and Computer Science,  
Freie Universität Berlin

University of Groningen  
December 2022

[https://github.com/MohsenSadeghi/ml\\_course](https://github.com/MohsenSadeghi/ml_course)

*“There's method in the madness”*

This quote, from *Hamlet*, suggests that there is a logical reason or purpose behind seemingly irrational or chaotic behavior. In the context of protein engineering using machine learning, this could be interpreted as meaning that although the process of designing and optimizing proteins may seem complex and unpredictable, there is a underlying structure and organization to the data and algorithms involved. Machine learning algorithms can help to uncover these patterns and relationships, enabling the design of proteins with specific properties and functions. The use of machine learning in protein engineering can therefore provide a systematic and rational approach to a field that might otherwise seem chaotic or unpredictable.

*ChatGPT, OpenAI*

# Introduction

## Proteins

- ↳ Macromolecules formed from single strands of **amino acids**
- ↳ Can form highly complex **3D structures**
- ↳ Responsible for most of **vital function** in living systems

## Wide range of protein functions

- ↳ Digesting food and metabolism, muscle movement, vision
- ↳ Interesting that all this function stems from rather simple chemistry
  - ~**20** building blocks (amino acids), **1D** polymers

## Maybe we can design proteins for our own set of purposes

- ↳ Protein **engineering**
- ↳ Drugs, antibodies, carbon capture (RuBisCO), plastic recycling (PETase)
- ↳ **Evolution** did it, probably there is something to it
  - Maybe do it faster?
- ↳ We can use readily available protein factories, i.e. living cells

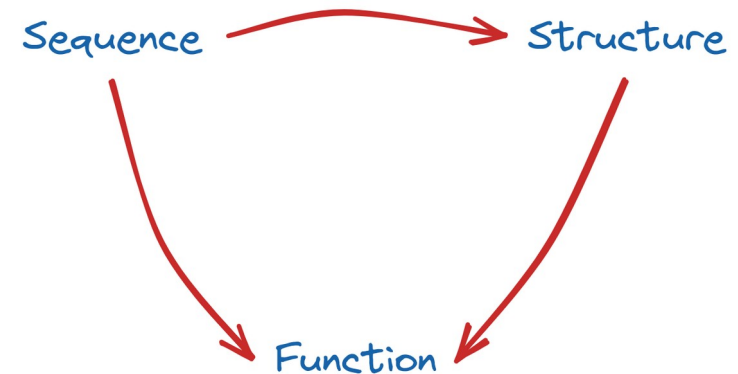
# Introduction

Why do we need the help of computers? i.e. **machine learning**

- ↳ Consider the **combinatorial** possibilities in the **sequence space** ( $20^N$ )
  - Every written text in English belongs to a set of  $26^N$  possibilities
- ↳ How are we going to come up with proteins that
  - Pass some structural criterion: e.g. fold in the first place
  - Follow evolutionary fitness
  - Have a certain function, without unwanted effects
  - Are top-performers

The art of using data, to come up with new data

- ↳ “Modeling” data
- ↳ Making sense of large amounts of data
  - Picking up patterns, generating new data



# Machine learning (ML)

Automatically find representations of large amounts of data

- ↳ For proteins: encoding function, structure, and evolutionary role

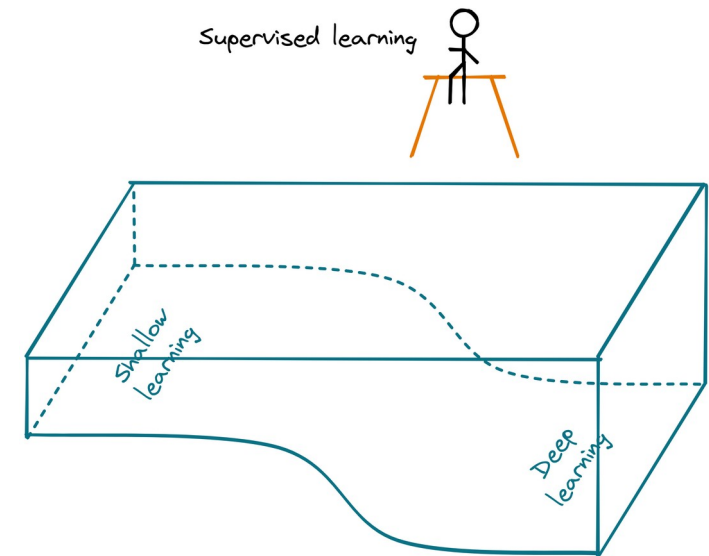
Different ways of categorizing ML models

- ↳ **Shallow** vs. **deep** learning
- ↳ **Supervised** vs. **unsupervised** (self-supervised) learning

Deep learning has taken the lead.

- ↳ Well-established software frameworks for easy implementation and application
  - **PyTorch**
  - TensorFlow, JAX

**GPUs** have helped a lot with the recent advances in ML



# Case 1: regression

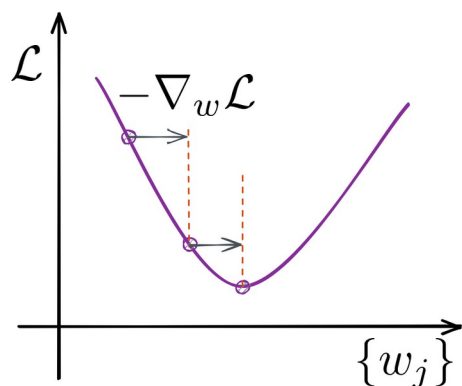
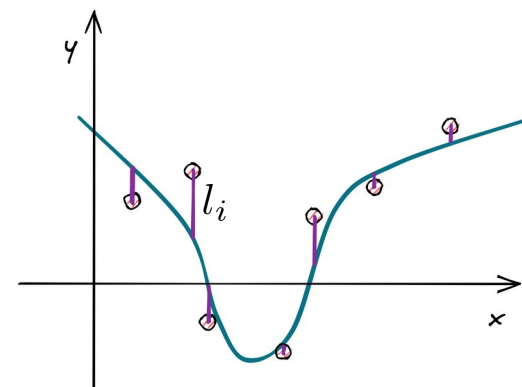
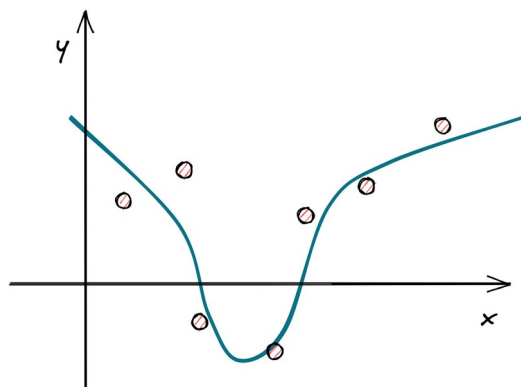
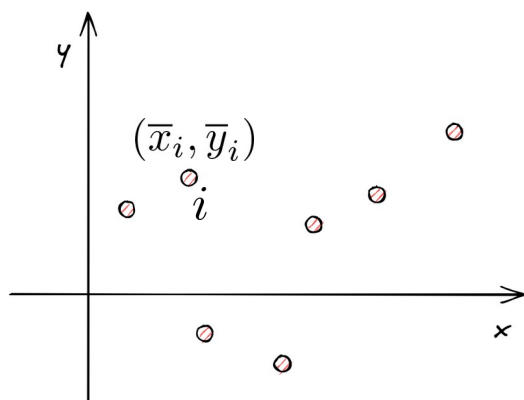
Given some data points, which inherently include noise, find a best-fitting function

- ↳ Learning functions in a **supervised** manner
- ↳ Translate the **learning** problem into an **optimization** in parameter-space

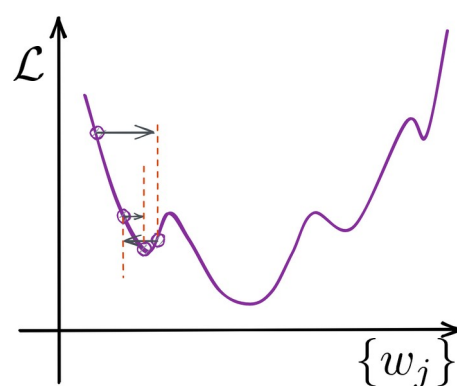
$$f(x; \{w_j\}) = w_0 + w_1x + w_2x^2 + \dots$$

Training dataset

$$D = \{(\bar{x}_1, \bar{y}_1), (\bar{x}_2, \bar{y}_2), \dots\}$$



Gradient descent



Stochastic gradient descent

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N l_i^2 = \frac{1}{N} \sum_i (f(\bar{x}_i) - \bar{y}_i)^2$$

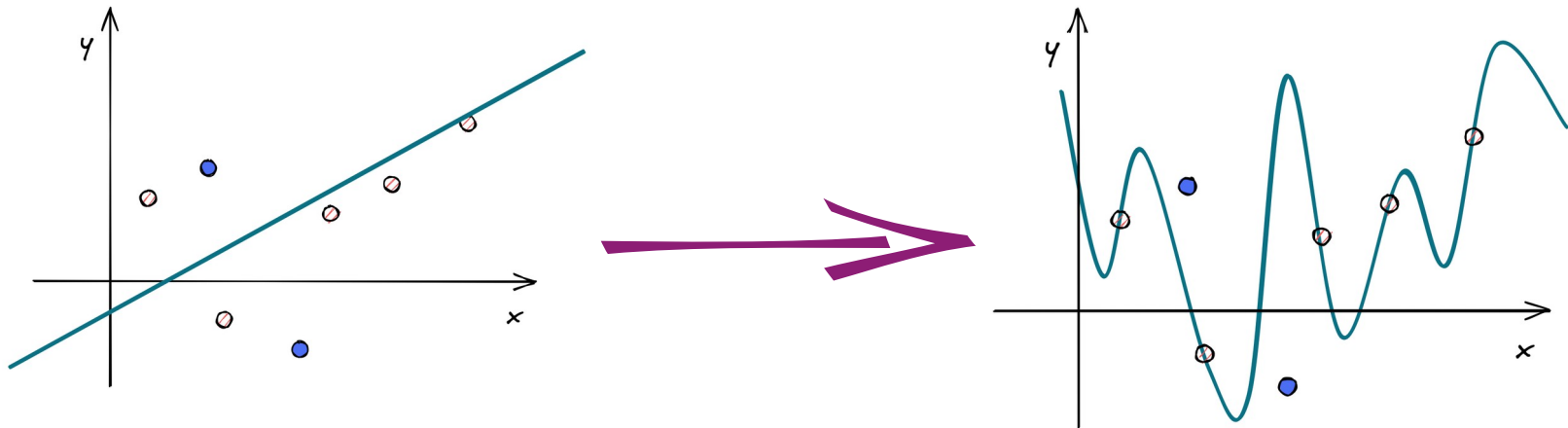
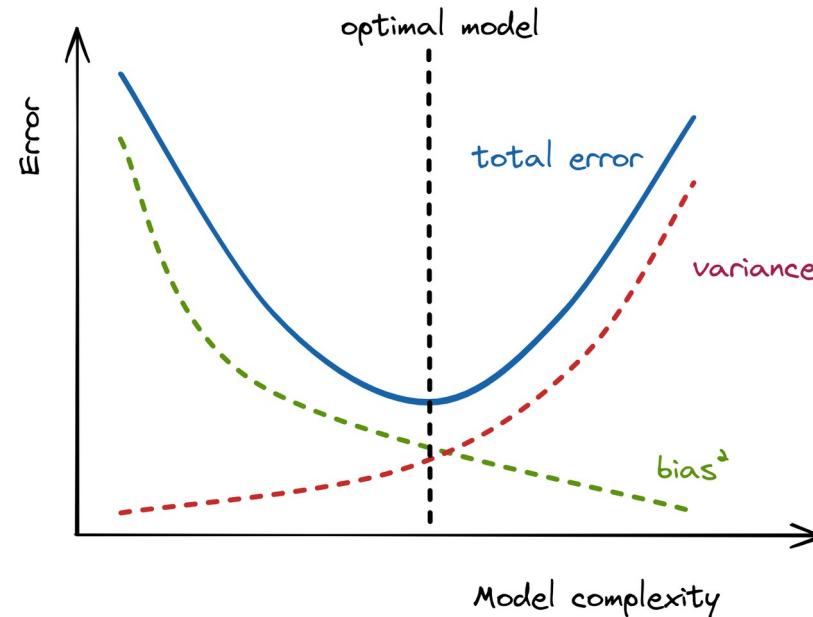
The **loss** function  
Mean Squared Error

# Case 1: regression

The question of how complex the model needs to be

How to check if the model learned anything or memorized everything?

- ↳ The concept of **overfitting**
- ↳ Cross-validation
  - Leave some data points out
  - Check the performance on them
  - **train/validation split**



# Case 1: regression

## **Code time**

### **Task:**

fitting a polynomial to a bunch of random data points, using  
stochastic gradient-descent



# Case 2: classification

Let's make “learning the function” a bit more generic and interesting.

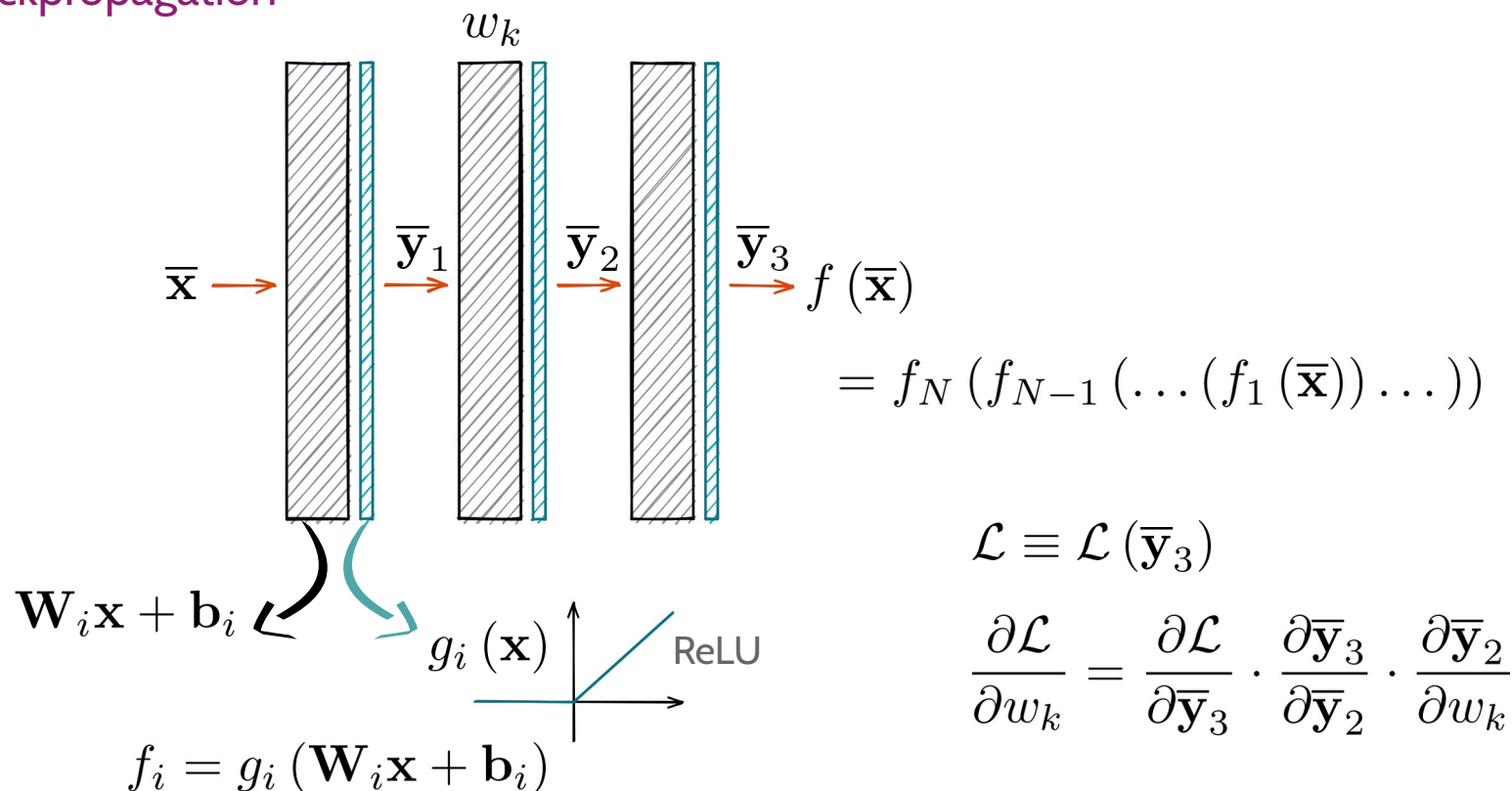
Stacking up “layers” in a **feed-forward** network

↳ Moving **from a shallow model** with a hand-crafted parametric function **to a deep model**

We just need the gradients to train similarly

↳ Trainable parameters are now matrix elements (network weights)

↳ **Backpropagation**

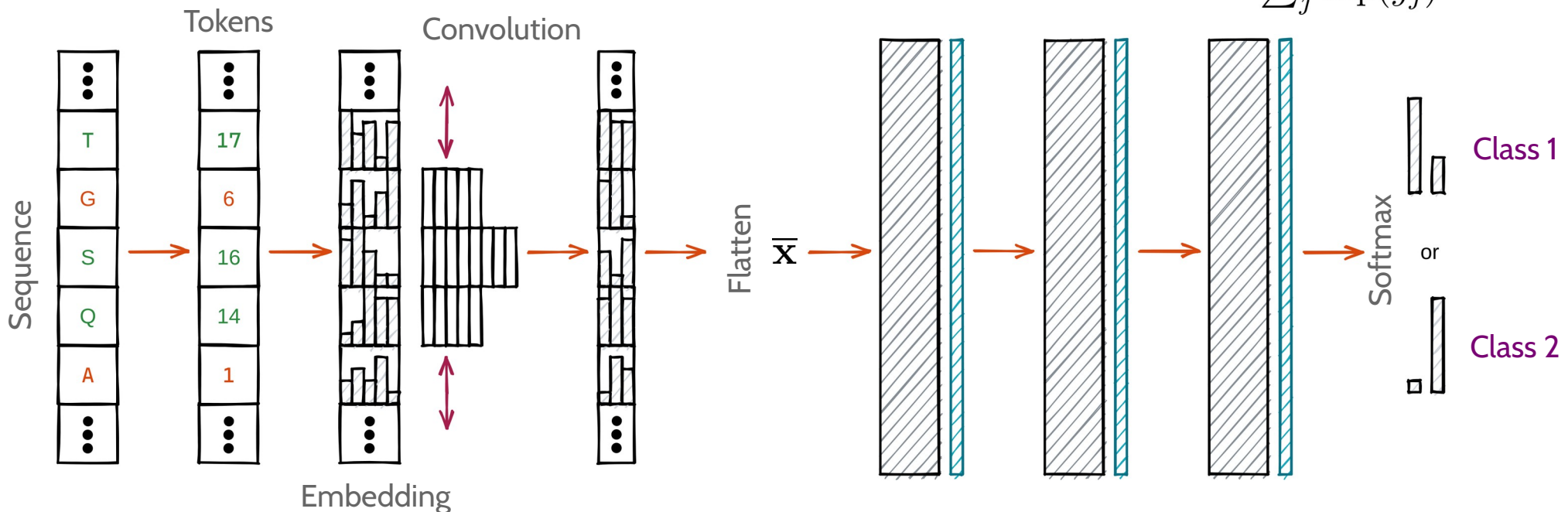


# Case 2: classification

*In silico* screening: training the model to decide on the family, function, etc. based on the sequence

- ↳ Turning the alphabet of protein sequence into numerical **tokens**
- ↳ Use a combination of **convolutional** and dense layers
- ↳ Make predictions on **class probabilities**
- ↳ **Cross-entropy** loss function  $\mathcal{L} = - \sum_i \bar{p}_i \log p_i$

$$p_i = \text{softmax}(\mathbf{y})_j = \frac{\exp(y_j)}{\sum_j \exp(y_j)}$$



## Case 2: classification

### Code time

#### Task:

Train a deep network to distinguish between two protein classes based on sequence data

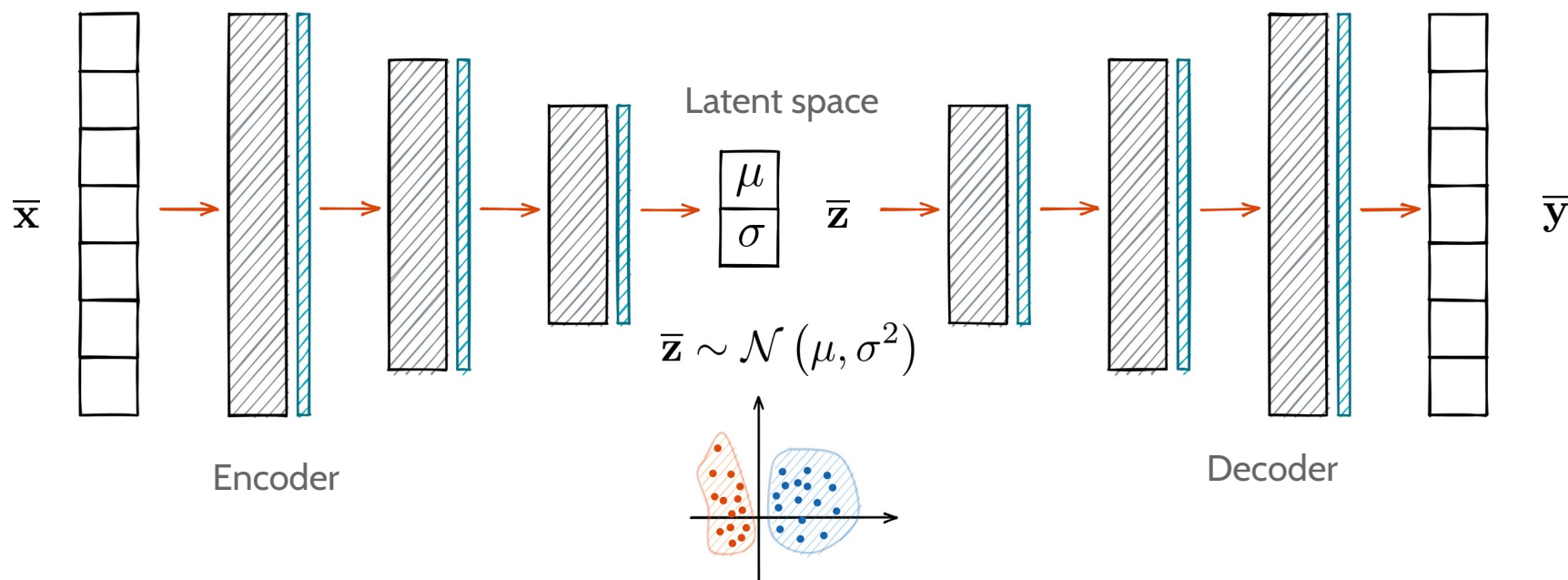
(Apolipoprotein or not)

# Case 3: representation

## Learning distributions versus learning functions

- ↳ Can the computer pick on similarities in the data on its own (i.e. without labels)?
  - Self-supervised learning
- ↳ Moving from learning the function to learning from which **distribution** the data points come from.
  - The exciting possibility of generating new and unseen samples (**Generative learning**).

Behold, the **Variational Autoencoder (VAE)**



$$\mathcal{L} = \|\bar{\mathbf{y}} - \bar{\mathbf{x}}\|^2 + \lambda D_{KL} [\mathcal{N}(\mu, \sigma^2) \parallel \mathcal{N}(0, 1)]$$

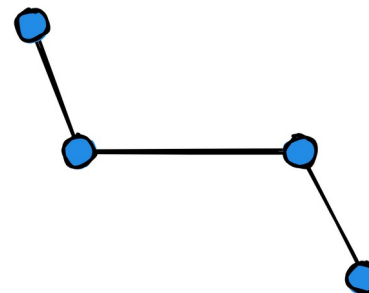
$$D_{KL} [P \parallel Q] = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

# Case 3: representation

## Code time

### Task:

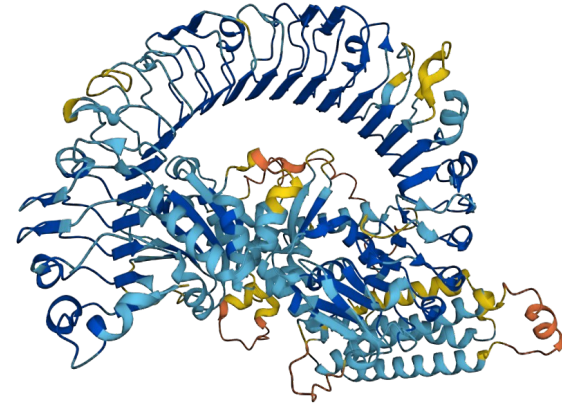
Train a variational autoencoder to learn the distinction between **cis** and **trans** conformations of random molecules from the position of their 4 atoms



# State-of-the-art

## AlphaFold2

- ↳ Google's DeepMind
- ↳ **sequence** → **structure**
- ↳ Blew the competition out of the water at **CASP14 (Critical Assessment of Structure Prediction)**
- ↳ Multiple sequence alignment (MSA) + residue-residue pairwise distance prediction



<https://alphafold.ebi.ac.uk/entry/Q8W3K0>

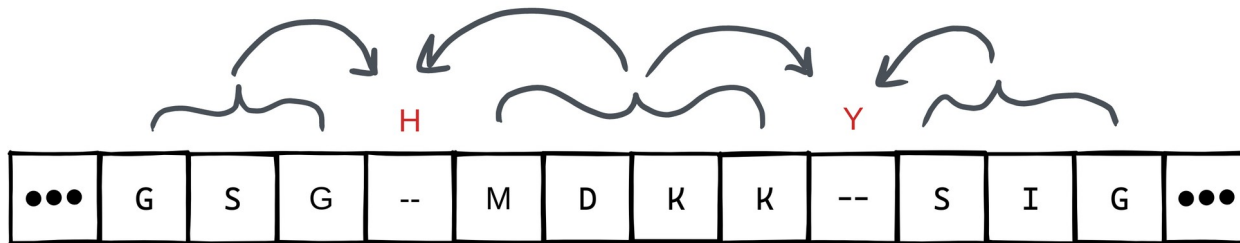
Pereira, J., et al. (2021). High-accuracy protein structure prediction in CASP14. *Proteins: Structure, Function and Bioinformatics*, **89**(12), 1687–1699.

Jumper, J., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**(7873), 583–589.

# State-of-the-art

Using **language models** to process sequence data

- ↳ Shallow learning: Hidden Markov Models
- ↳ Deep Language Models
  - Using Recurrent Neural Networks with attention-based memory (transformers)
- ↳ MetaAI Evolutionary Scale Modeling (ESM) Atlas <https://esmatlas.com/explore>
  - 617 million proteins in one representation map
  - **Structural information** has been internally learned by the network!



Credit: ESM Metagenomic Atlas

Callaway, E. (2022). AlphaFold's new rival? Meta AI predicts shape of 600 million proteins. *Nature*, 611(7935), 211–212.

# Conclusion

You have (hopefully) learned

- ↳ The importance of protein design and engineering and the main challenge involved
- ↳ How machine learning can help make sense out of the universe of possibilities
- ↳ The wide range of ML models
  - Learning functions
  - Classification in a supervised manner
  - Learning representations and probability distributions
- ↳ State-of-the-art in ML for protein engineering
  - Structure prediction
  - Large language models trained on sequences



# References and further reading

- (1) Goodfellow, I. J., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.  
<http://www.deeplearningbook.org>
- (2) White, A. D. (2022). Deep learning for molecules and materials. Living Journal of Computational Molecular Science, 3(1). <https://doi.org/10.33011/livecoms.3.1.1499>
- (3) Nilsson, N. J. (2005). Introduction to Machine Learning. In Department of Computer Science, Stanford University. <https://ai.stanford.edu/~nilsson/mlbook.html>
- (4) Bepler, T., & Berger, B. (2021). Learning the protein language: Evolution, structure, and function. Cell Systems, 12(6), 654–669.e3. <https://doi.org/10.1016/j.cels.2021.05.017>
- (5) Akdel, M., Pires, et al. (2022). A structural biology community assessment of AlphaFold2 applications. Nature Structural and Molecular Biology, 5, 2021.09.26.461876.  
<https://doi.org/10.1038/s41594-022-00849-w>
- (6) Watson, J. L., et al. (2022). Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. bioRxiv 519842.  
<https://doi.org/10.1101/2022.12.09.519842>