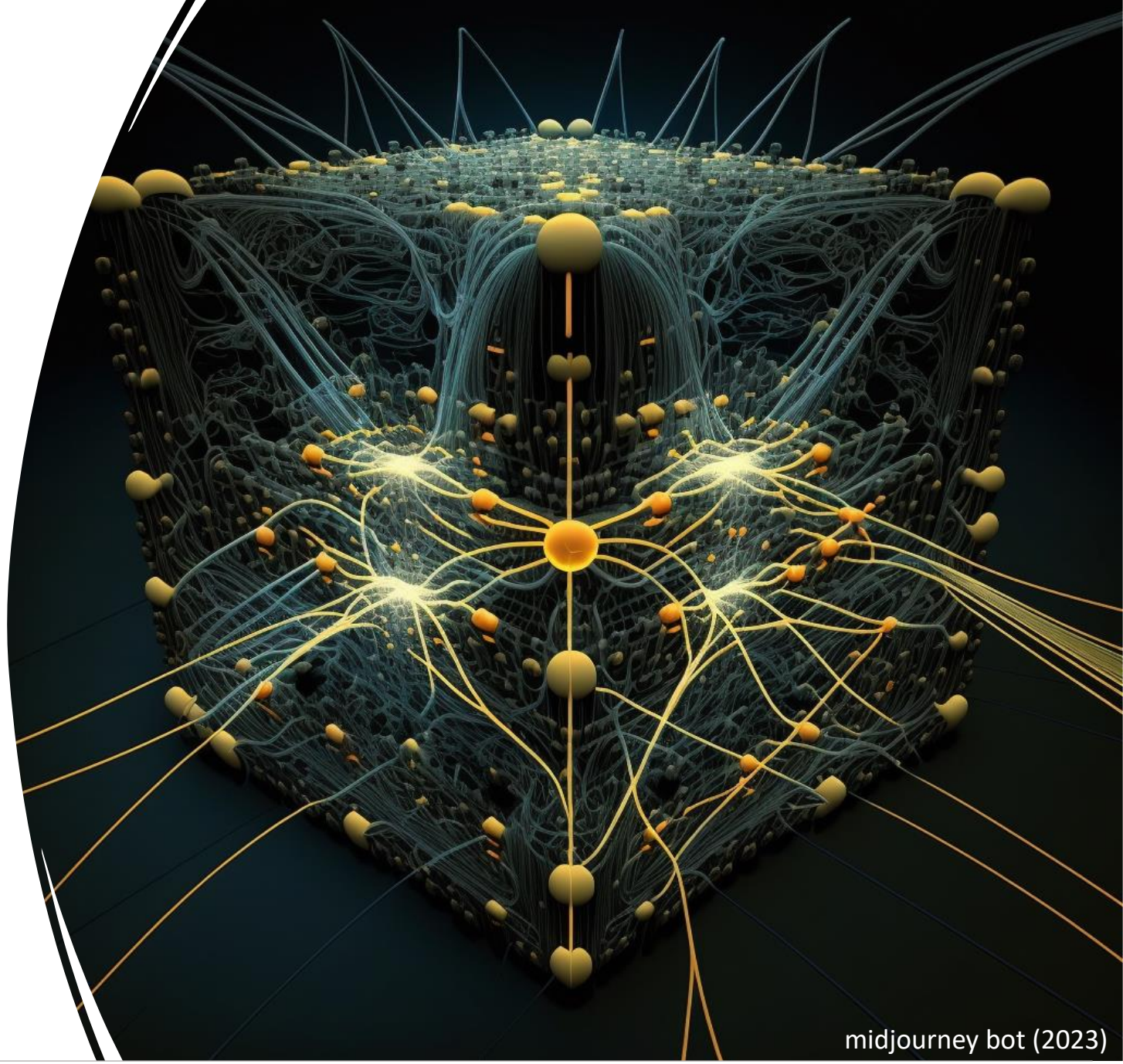
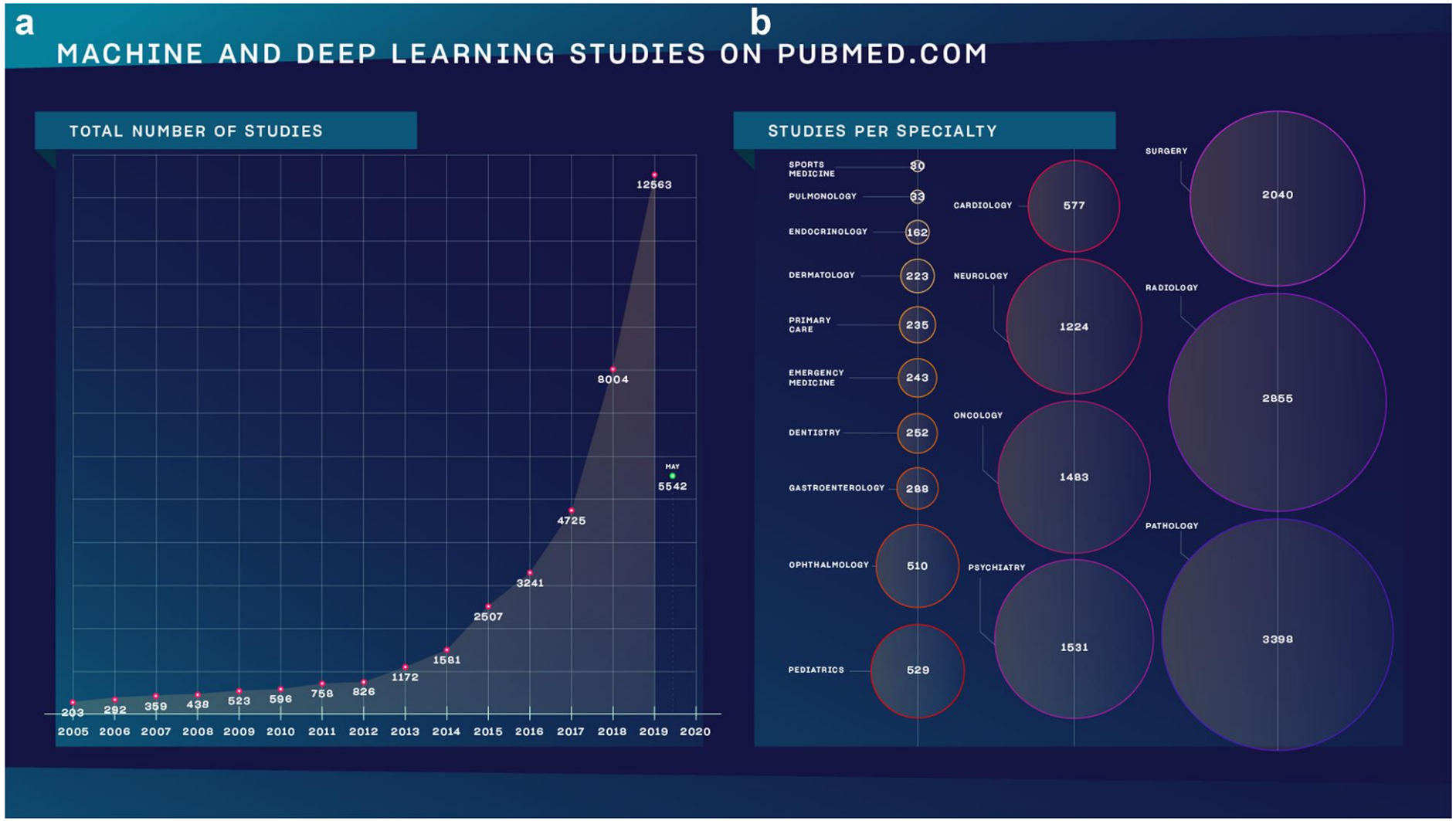
deep neural networks



deep learning



Meskó, B., Görög, M. A short guide for medical professionals in the era of artificial intelligence. npj Digit. Med. 3, 126 (2020)

logistic regression

Fit a logistic model

$$f(x,\theta) = g(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m) = g(\theta' x)$$

such that the cost function

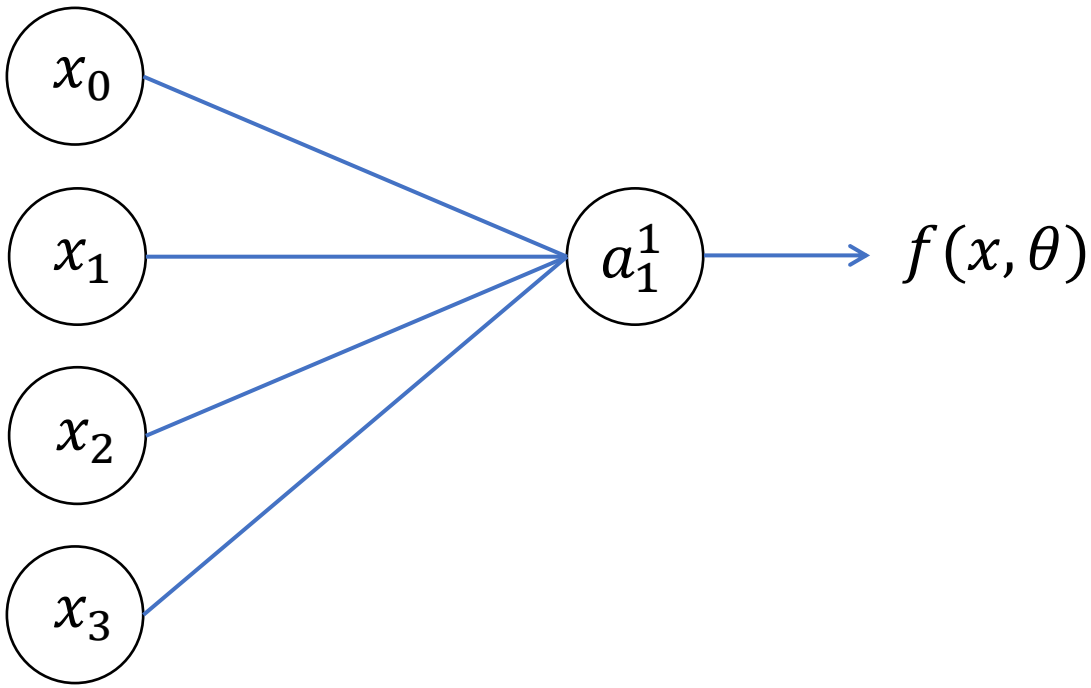
$$J(\theta) = -[\frac{1}{n}\sum_{i=1}^{n} y^{(i)} log(f(x^{(i)},\theta)) + (1 - y^{(i)}) log(1 - f(x^{(i)},\theta))]$$

is minimised with the gradient descent algorithm:

$$\theta_j := \theta_j - \alpha \frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)},\theta) - y^{(i)})x_j^{(i)}$$

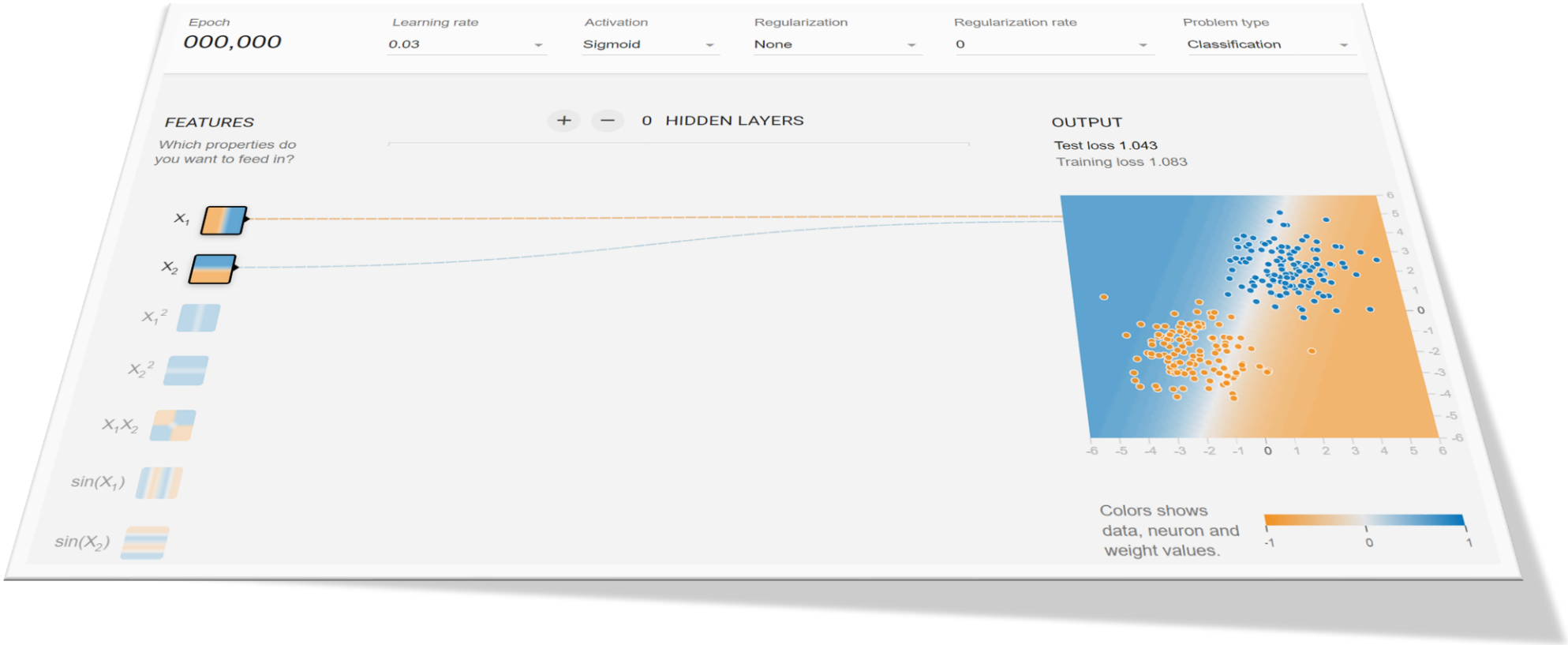
neural network

We can visualize a logistic regression model as follows:



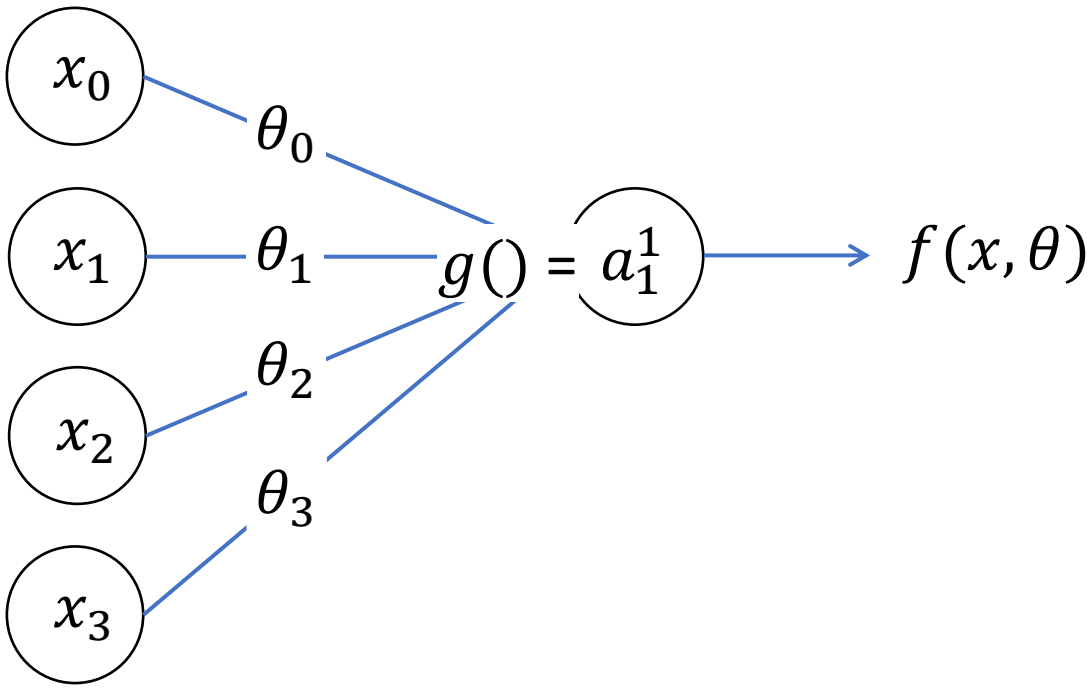
neural network

We can visualize a logistic regression model as follows (ctrl-click image):



neural network

We can visualize a logistic regression model as follows:



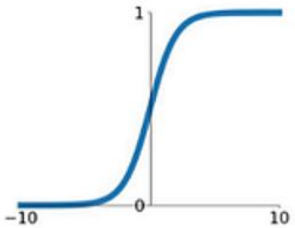
non-linearity: activation functions g()

We can employ different types of activation functions.

They need to be non-linear (and ‘differentiable…..ReLU?...’).

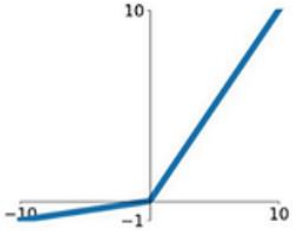
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



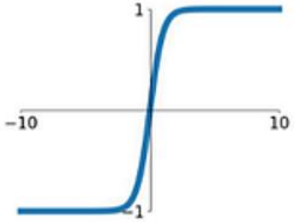
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

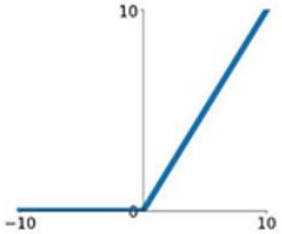


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

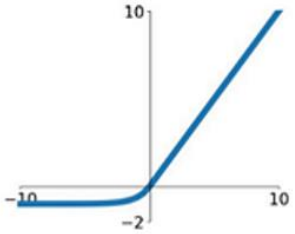
ReLU

$$\max(0, x)$$



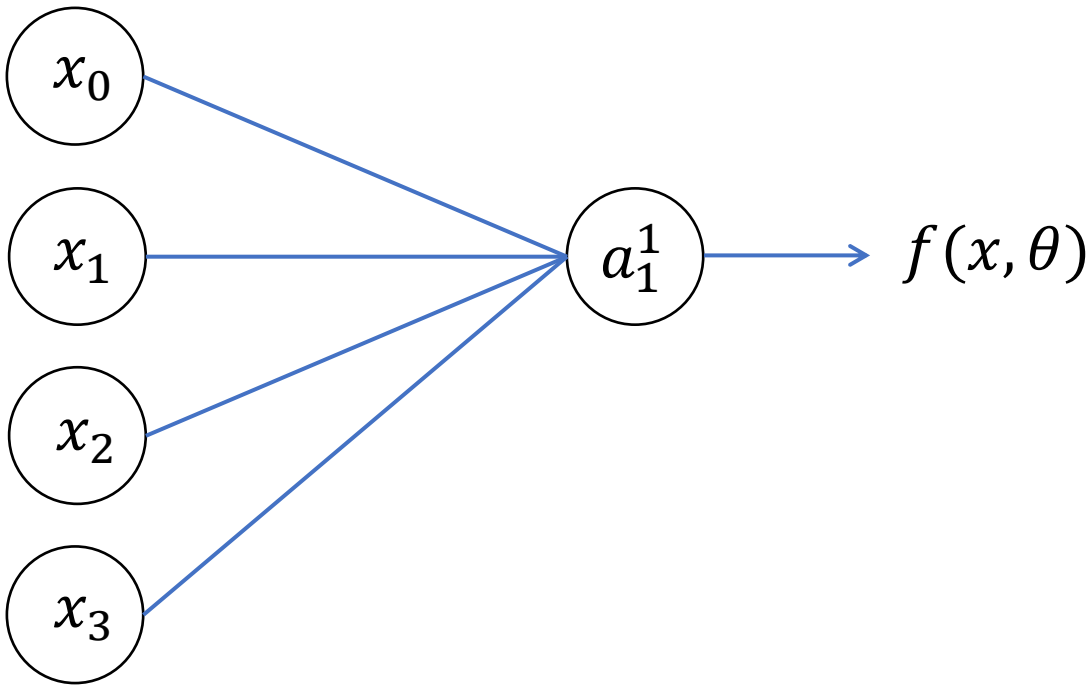
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



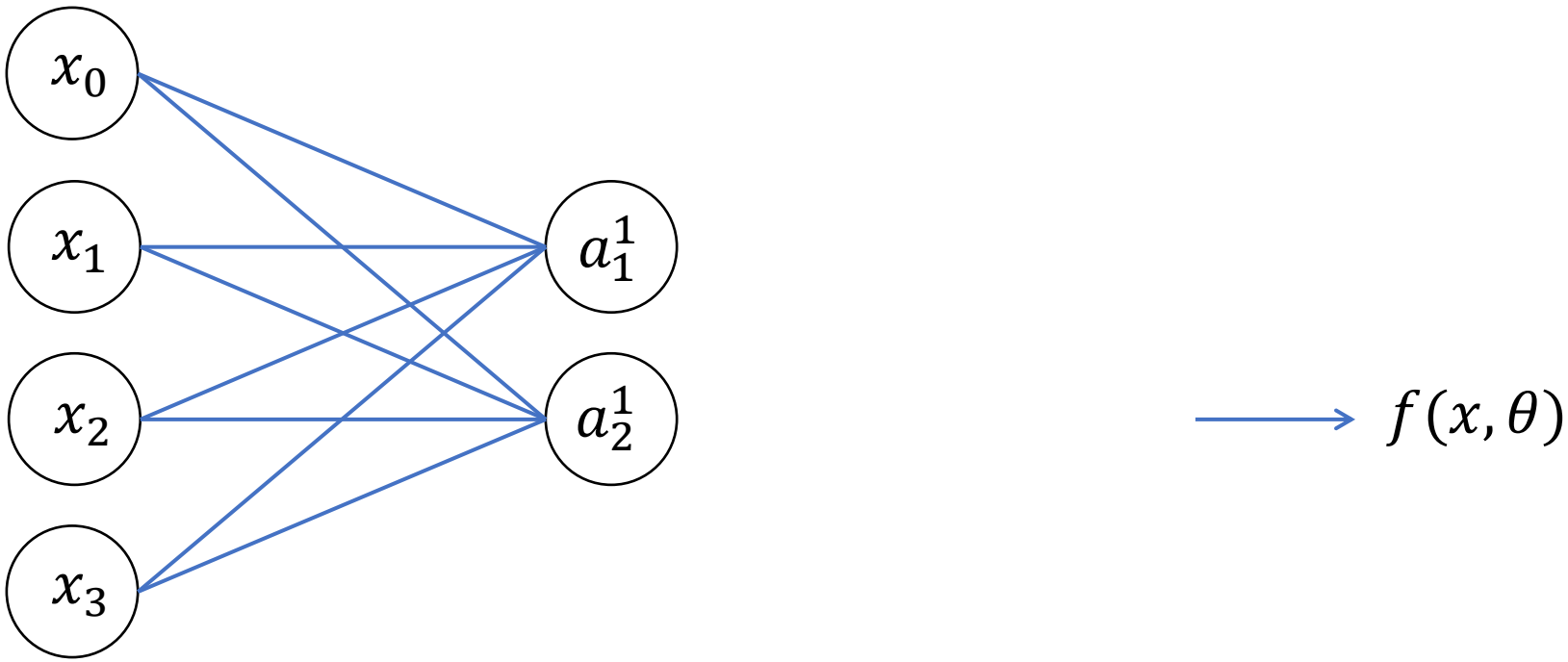
neural network

We can visualize a logistic regression model as follows:



neural network

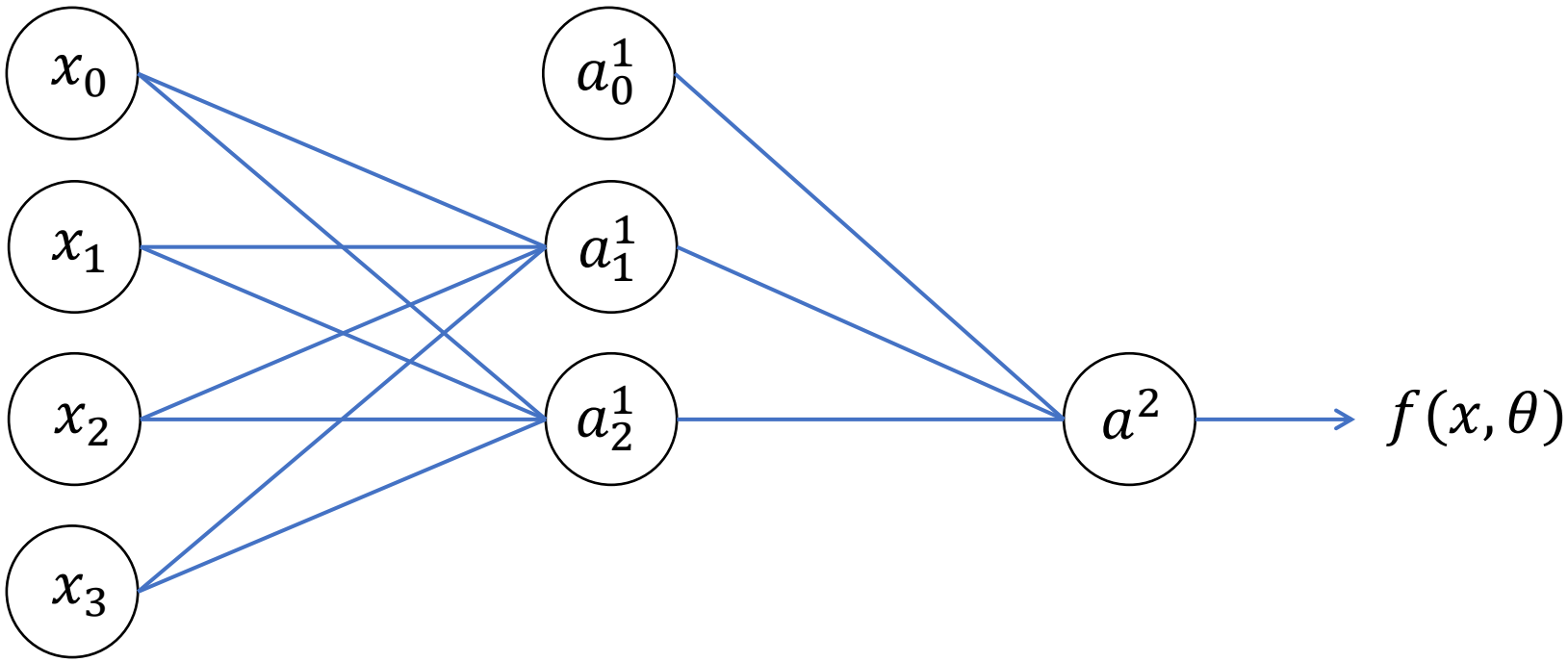
A neural network is an **architecture** that consists of logistic models:



An additional logistic model is added as a **node** in the **model's first (hidden) layer**.

neural network

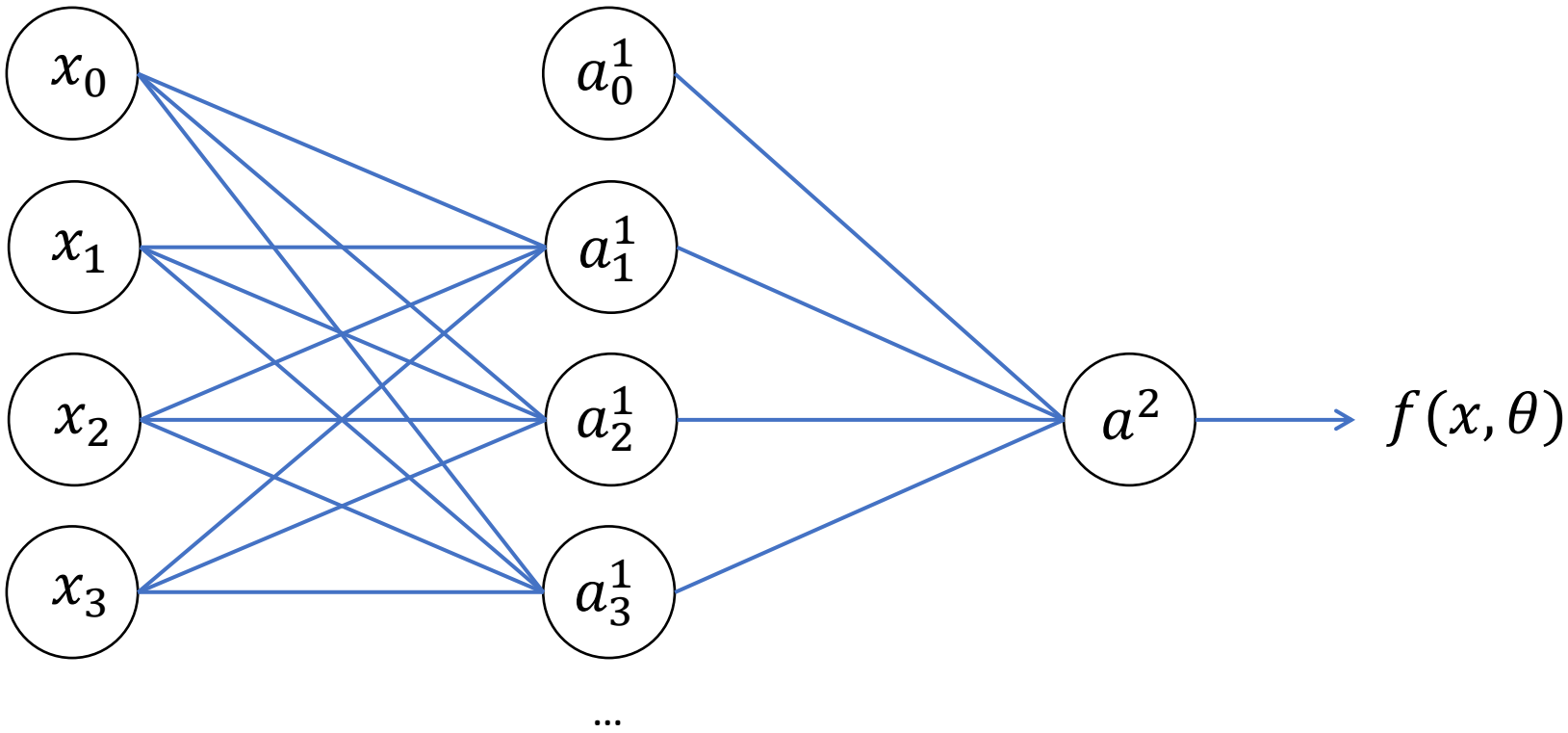
A neural network is an **architecture** that consists of logistic models:



An additional logistic model is added as a **node** in the **model's first (hidden) layer**, followed by a logistic regression model in the **output layer**.

neural network

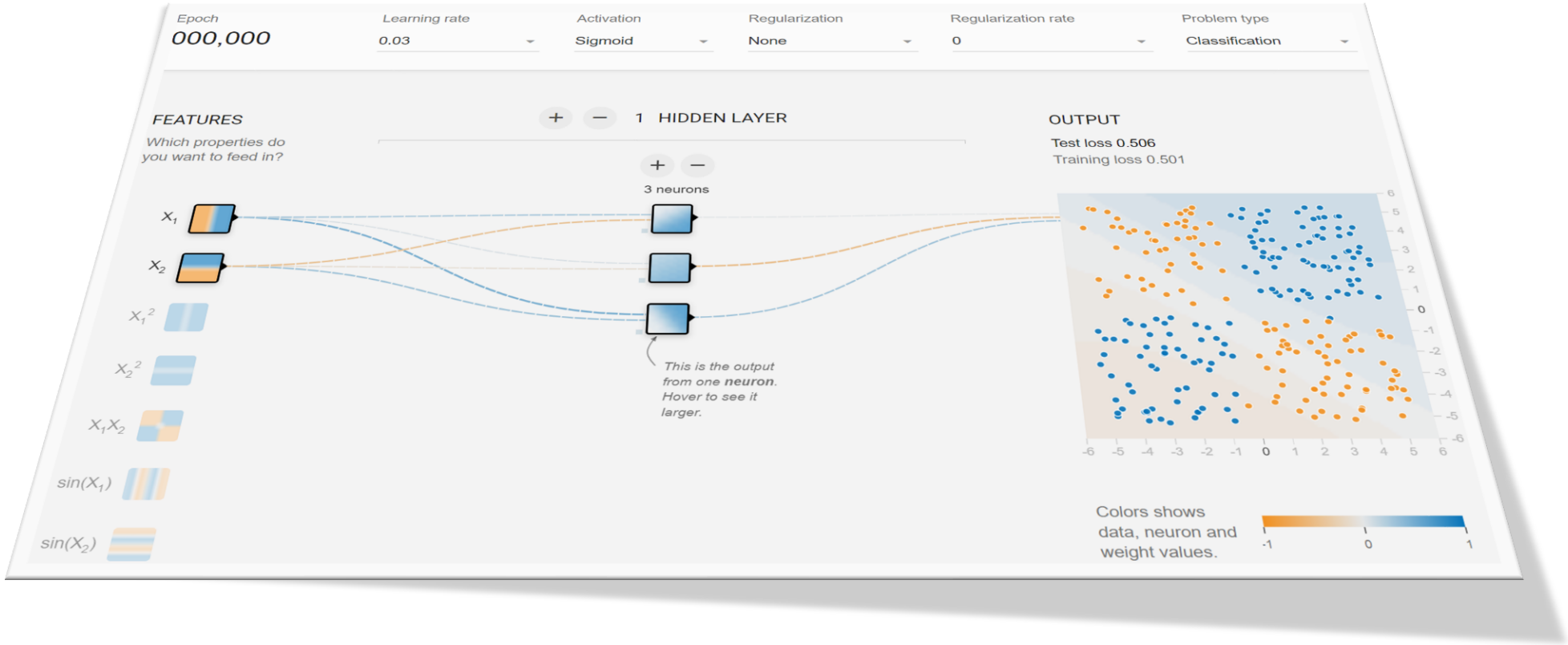
A neural network is an **architecture** that consists of logistic models:



More additional logistic models can be added as **nodes** in the **model's first (hidden) layer**, followed by a logistic regression model in the **output layer**.

neural network

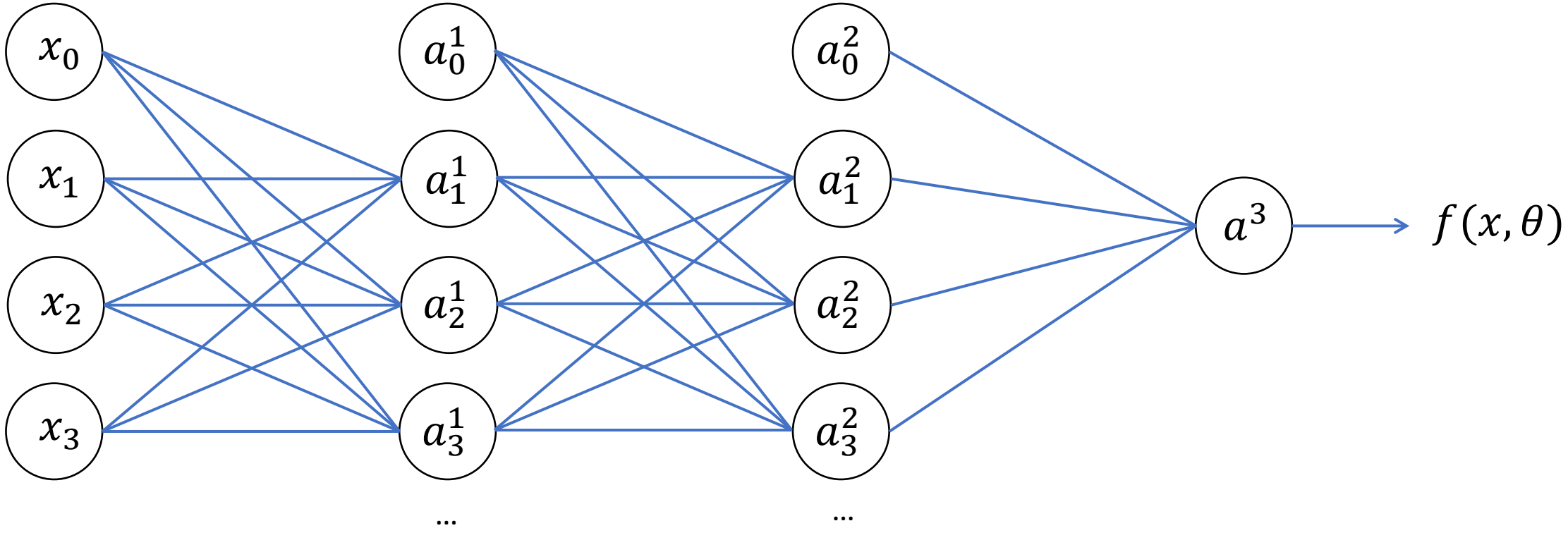
A neural network is an **architecture** that consists of logistic models (ctrl-click image):



More additional logistic models can be added as **nodes** in the **model's first (hidden) layer**, followed by a logistic regression model in the **output layer**.

(deep) neural network

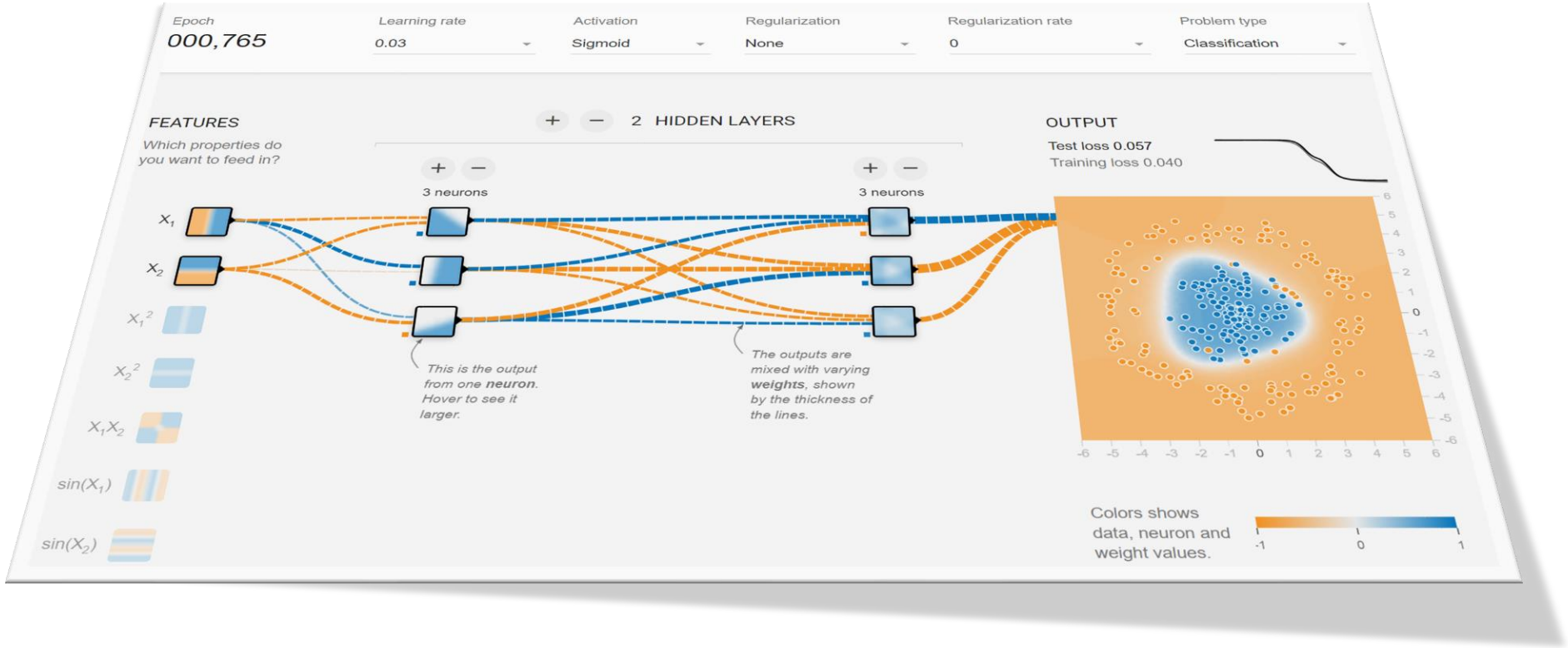
A neural network is an **architecture** that consists of logistic models:



We can also add a **second (or more) layer** of nodes to the model, followed by an **output layer**.

(deep) neural network

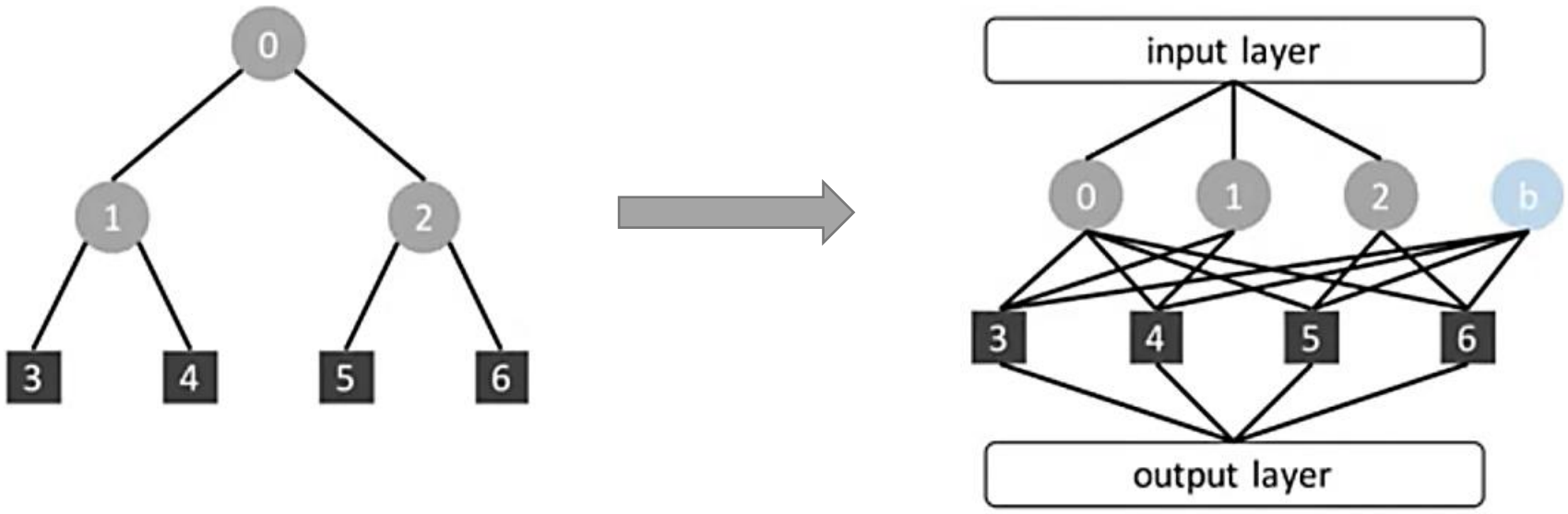
A neural network is an **architecture** that consists of logistic models (ctrl-click image):



We can also add a **second (or more) layer** of nodes to the model, followed by an **output layer**.

(deep) neural network

A neural network is somewhat similar to a decision tree.



hidden layer 1

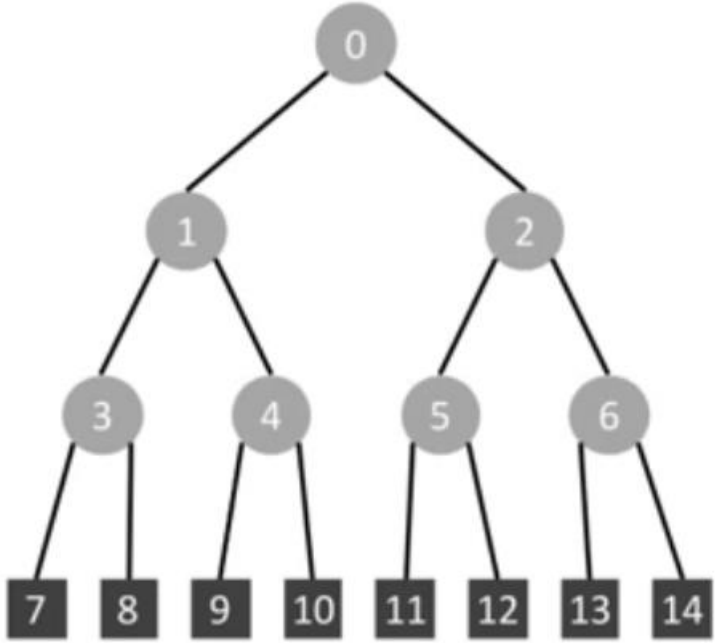
hidden layer 2

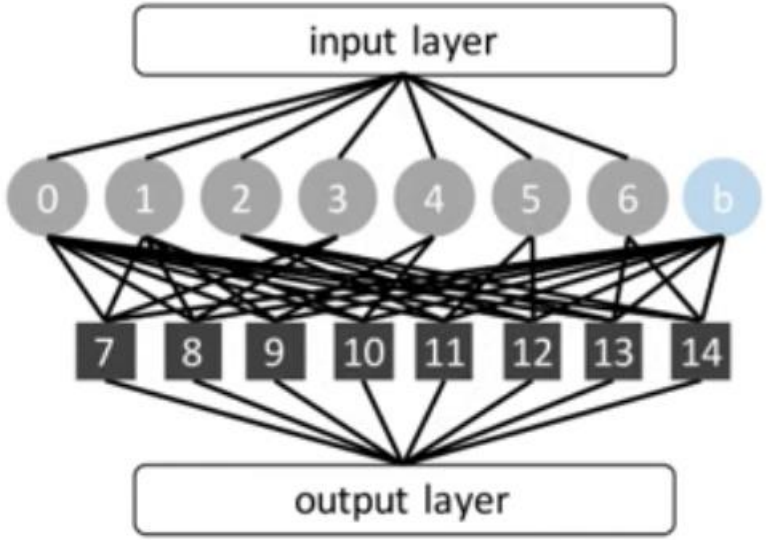
Sethi I.K., Proceedings of the IEEE (1990)

(deep) neural network

A neural network is somewhat similar to a decision tree.

Except that adding a level in a decision tree is the same as expanding the first hidden layer in a neural network. This means that neural networks can learn much more complex functions by adding hidden layers.



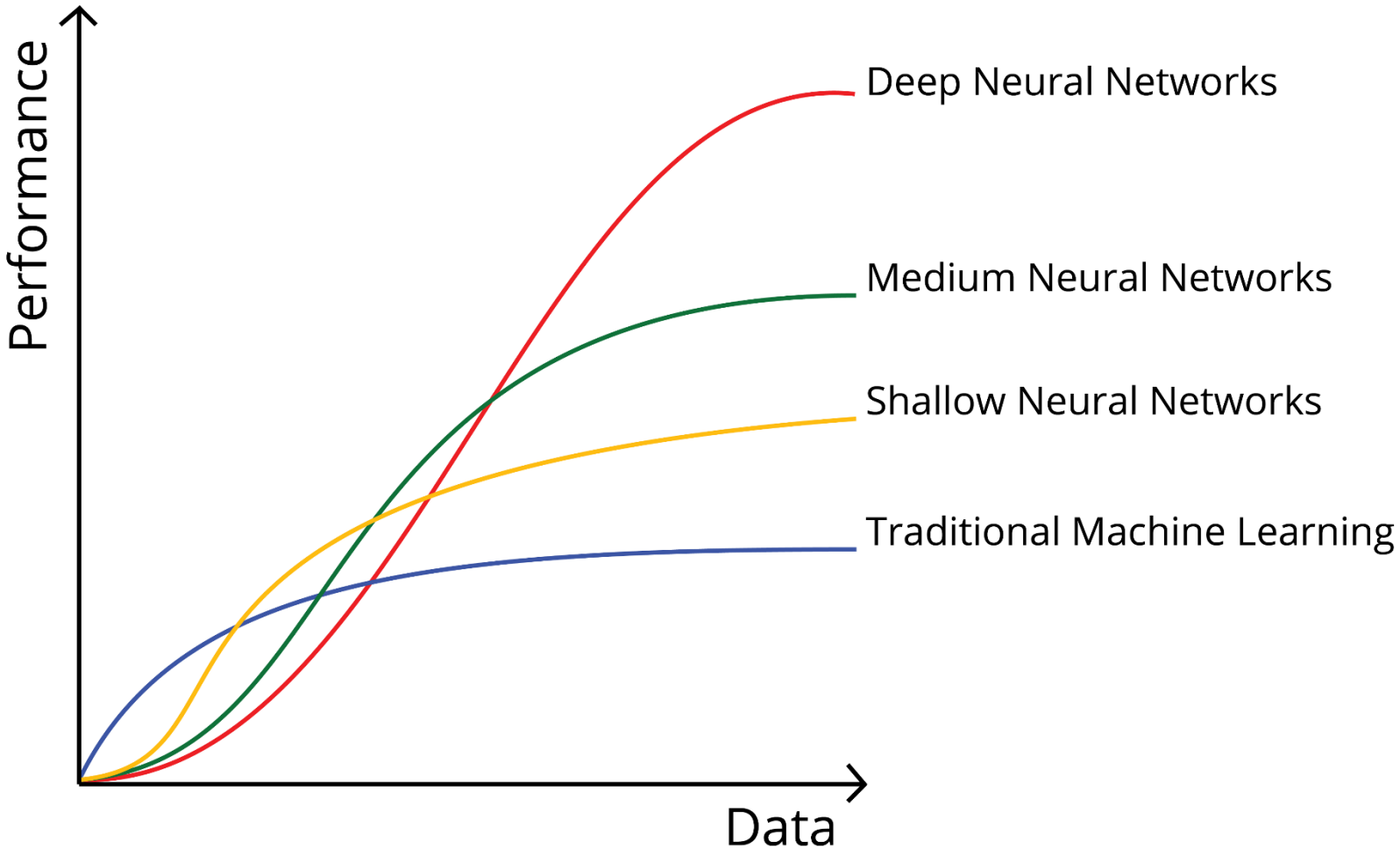


hidden layer 1

hidden layer 2

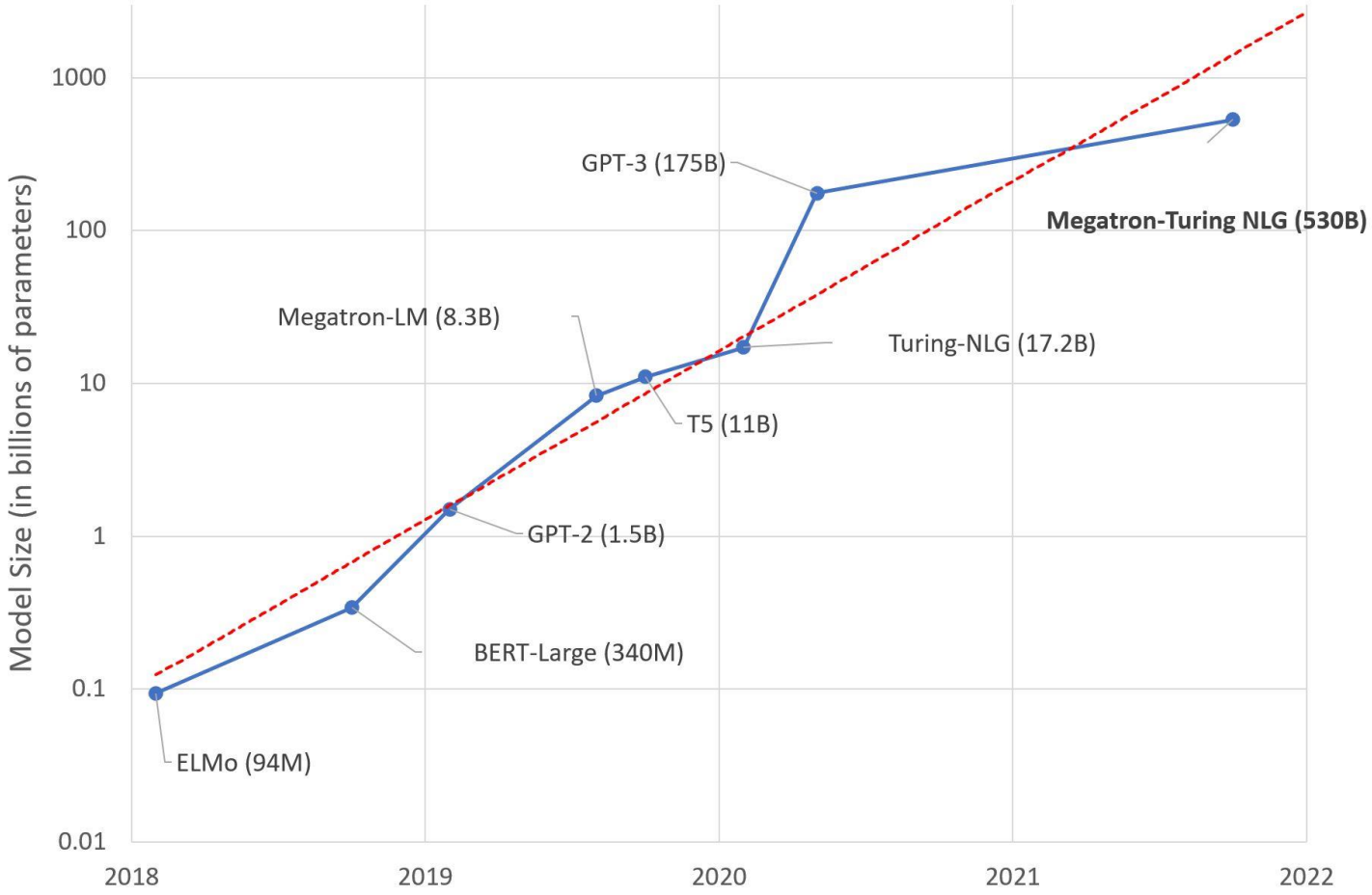
Sethi I.K., Proceedings of the IEEE (1990)

deep neural networks



deep neural networks

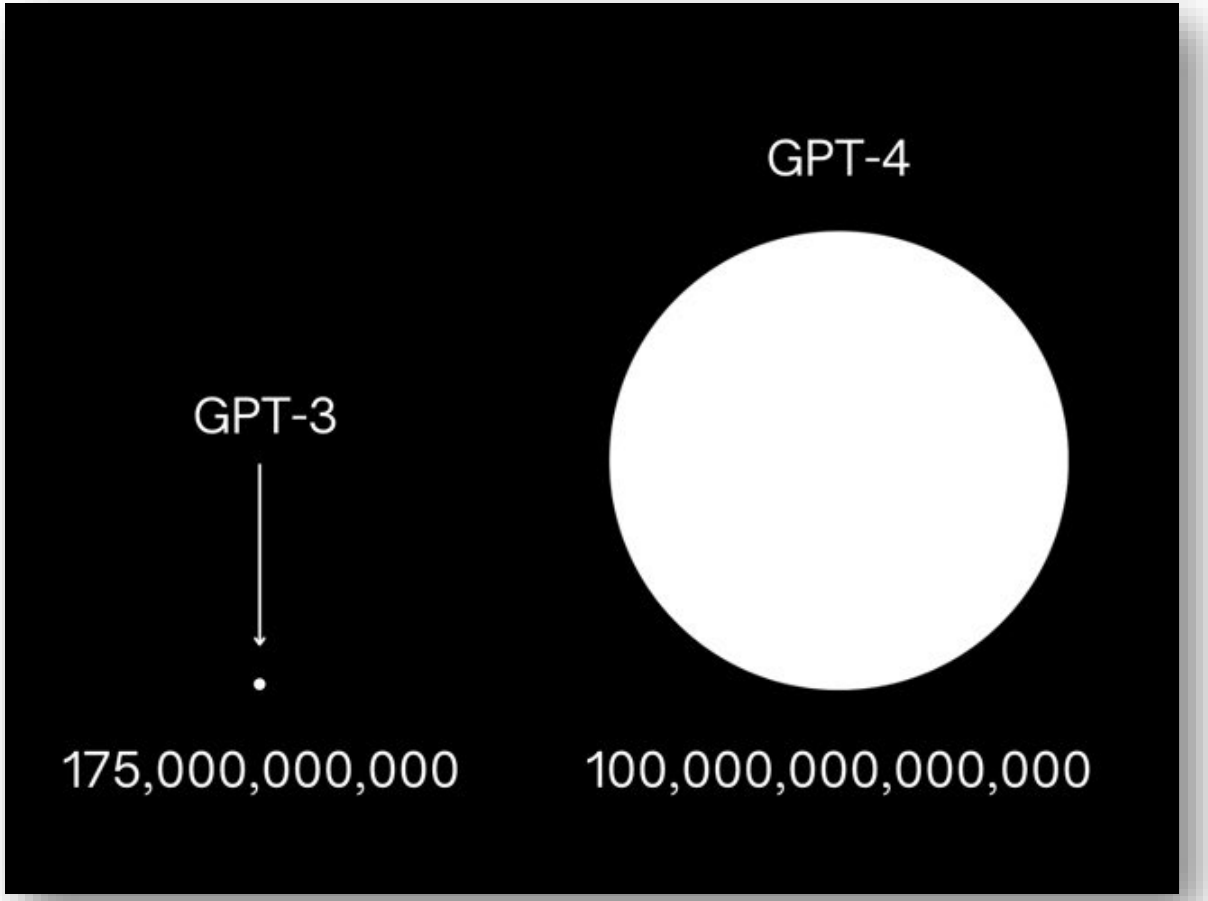
AI is (for a large part) about scaling things up *(logistic regression on steroids).*



https://huggingface.co/blog/large-language-models

deep neural networks

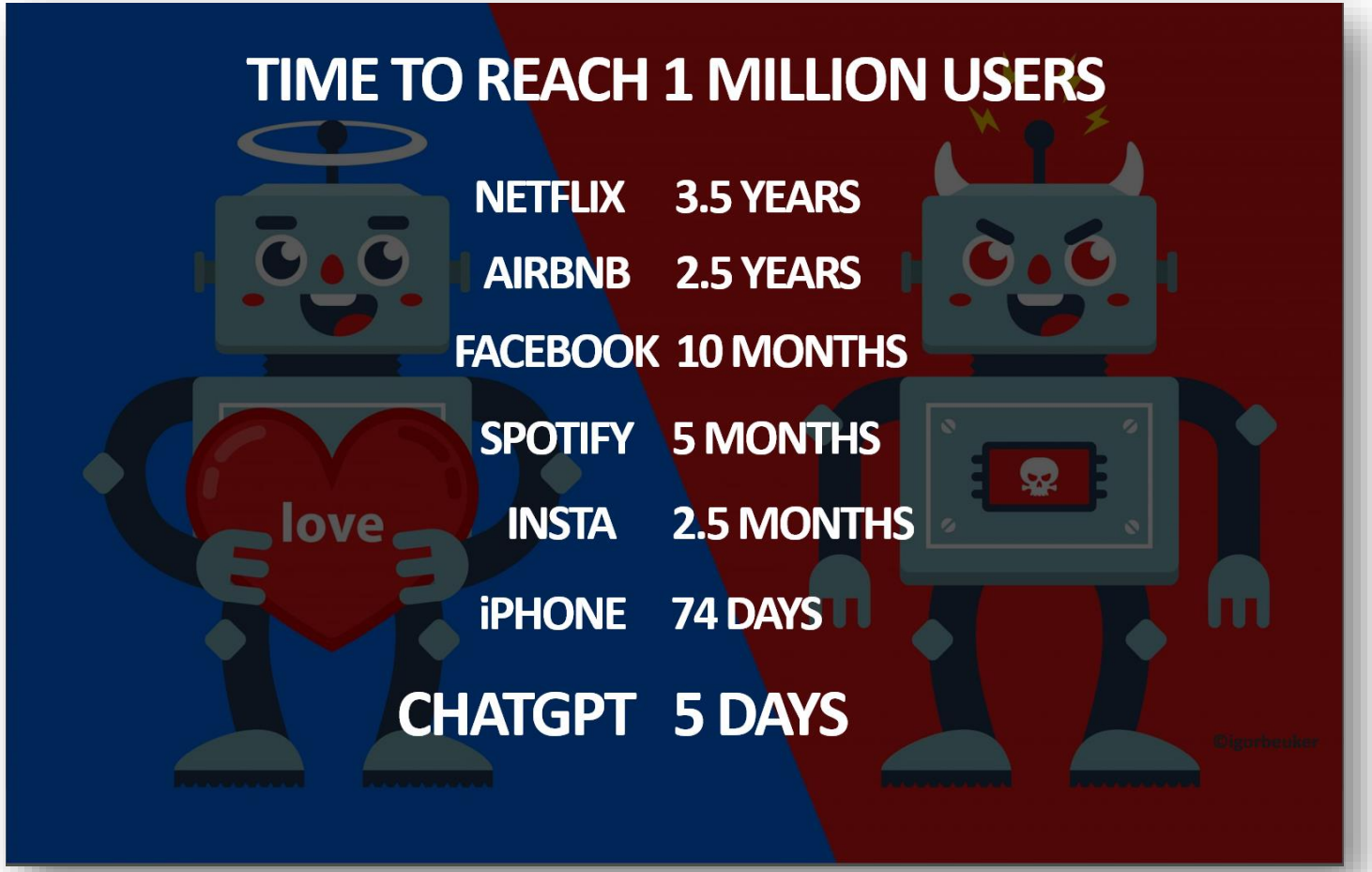
AI is (for a large part) about scaling things up *(logistic regression on steroids).*



https://openai.com/product/gpt-4

deep neural networks

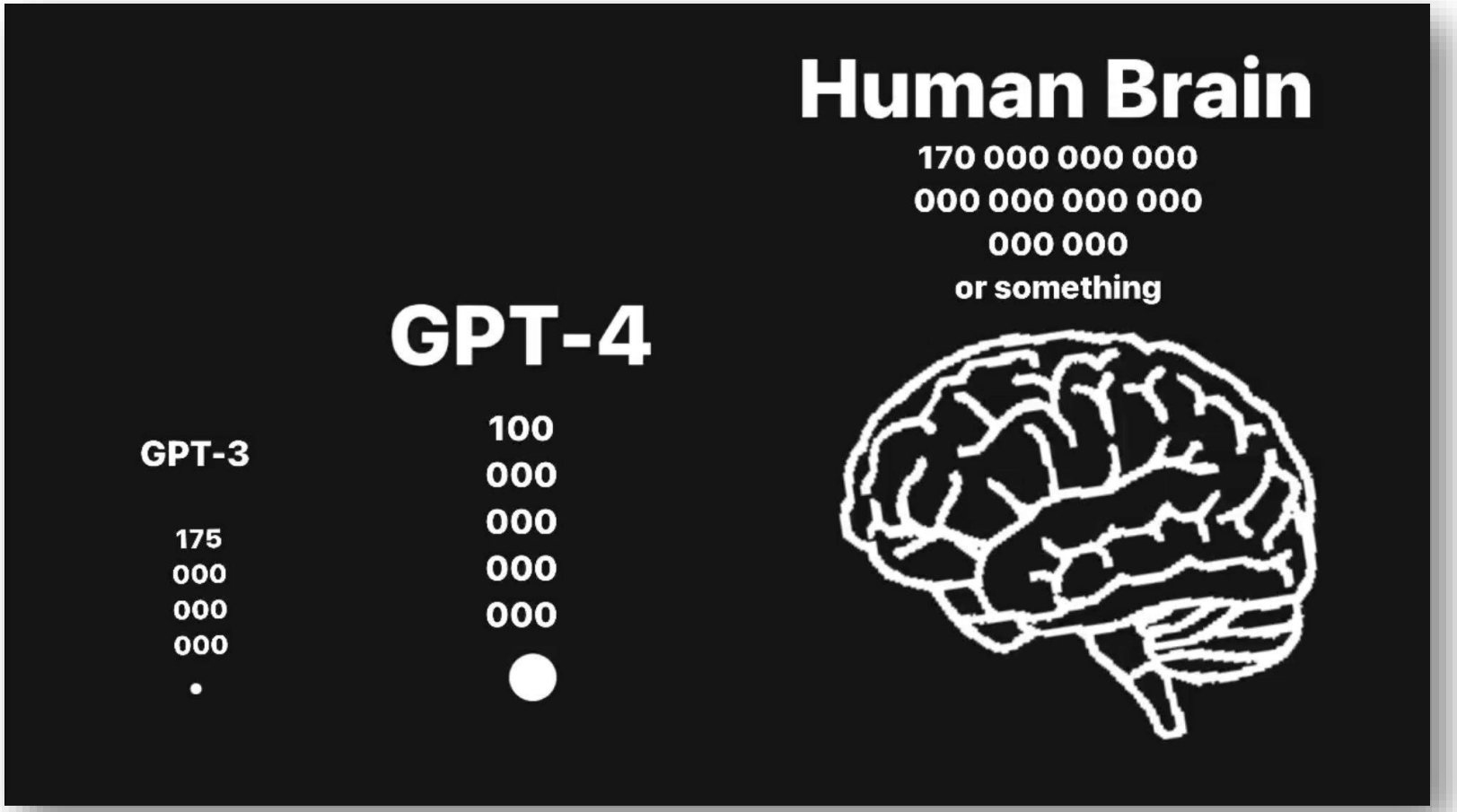
AI is (for a large part) about scaling things up *(logistic regression on steroids)*.



https://igorbeuker.com/marketing-innovation-blog

deep neural networks

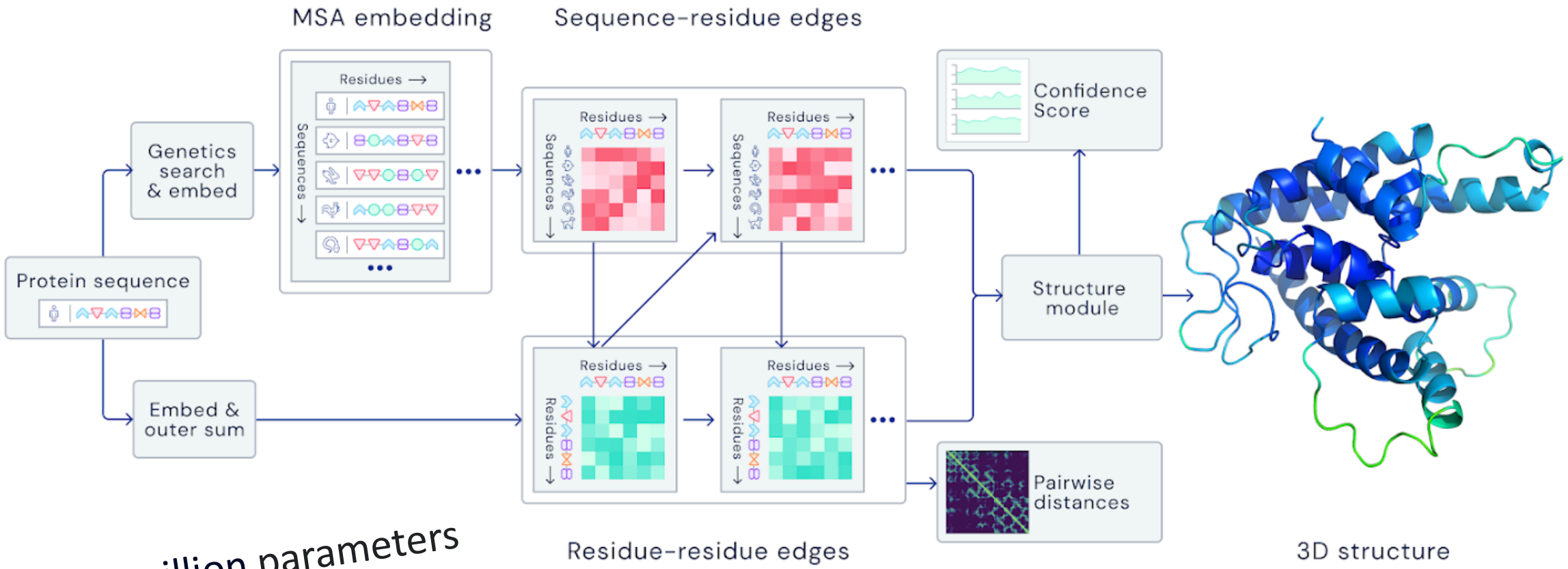
AI is (for a large part) about scaling things up *(logistic regression on steroids).*



https://openai.com/product/gpt-4

deep neural networks

AI is (for a large part) about scaling things up *(logistic regression on steroids).*



21 million parameters

https://www.nature.com/articles/s41586-021-03819-2

modules

AI is (for a large part) about scaling things up *(logistic regression on steroids).*

modules == building block == transformation == function

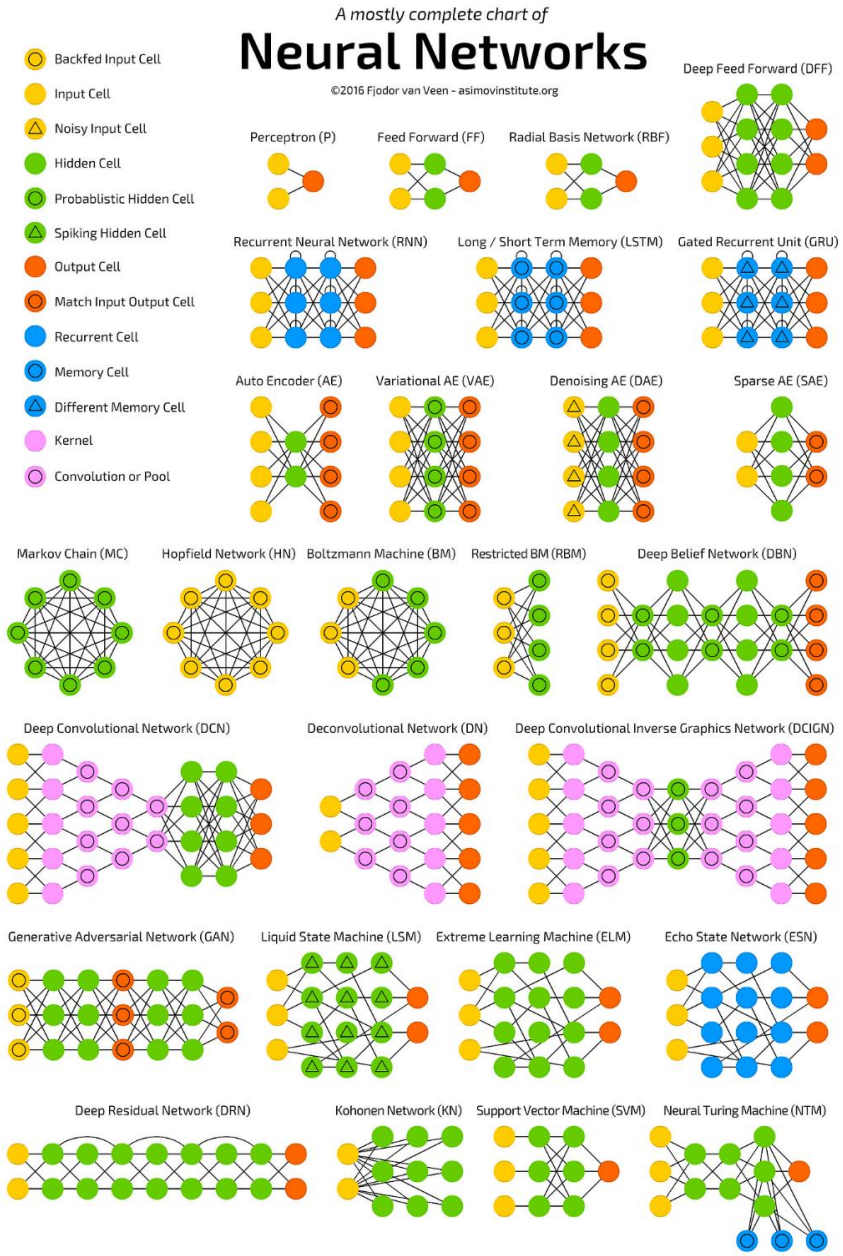
A module receives as input either data $x$or another module's output.

A module returns an output based on its activation function.

A module may or may not have trainable modelparameters.

modules

Complex hierarchies of modules.

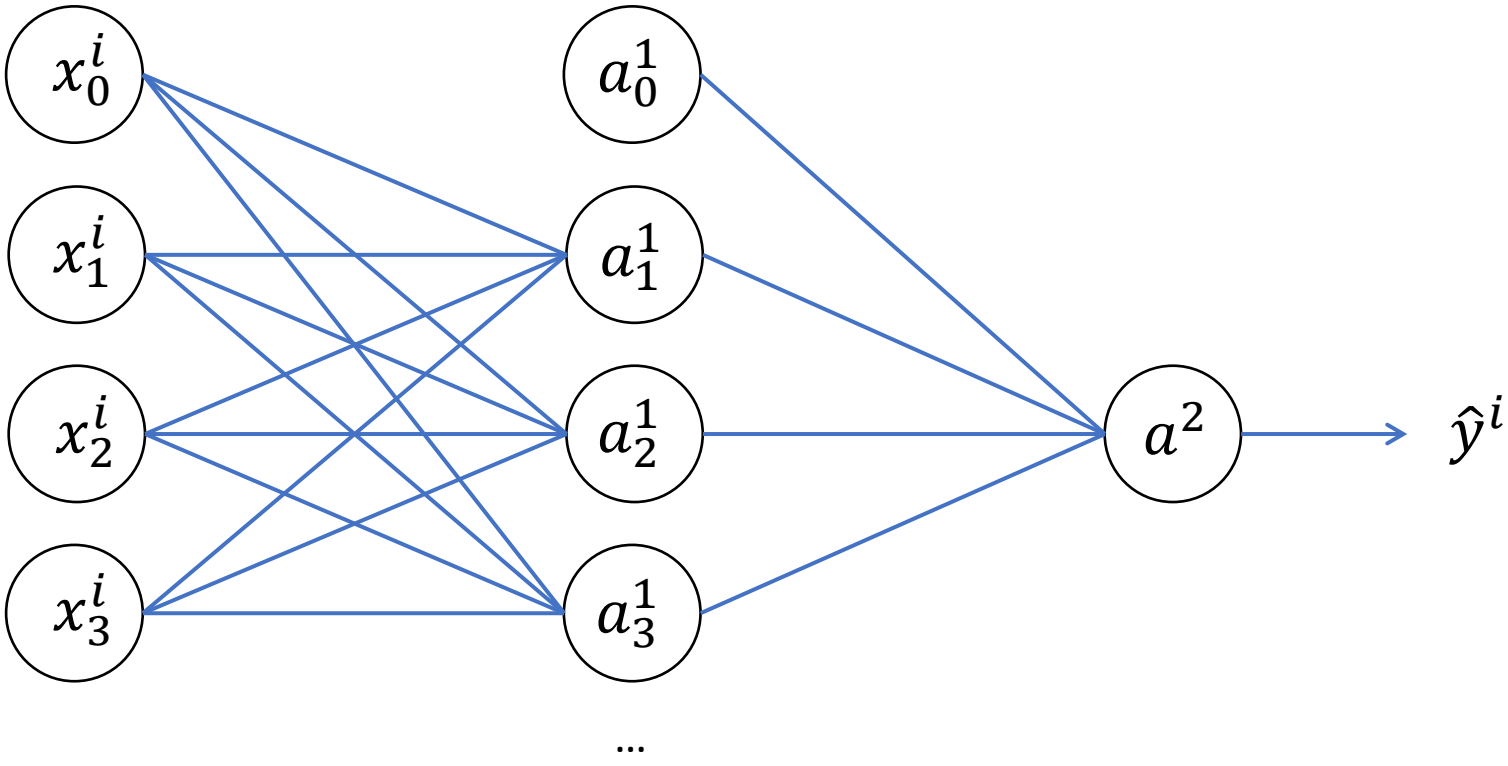


Fjodor van Veen from Asimov institute

loss function

This function computes the “distance” between the output of the model and the expected output.

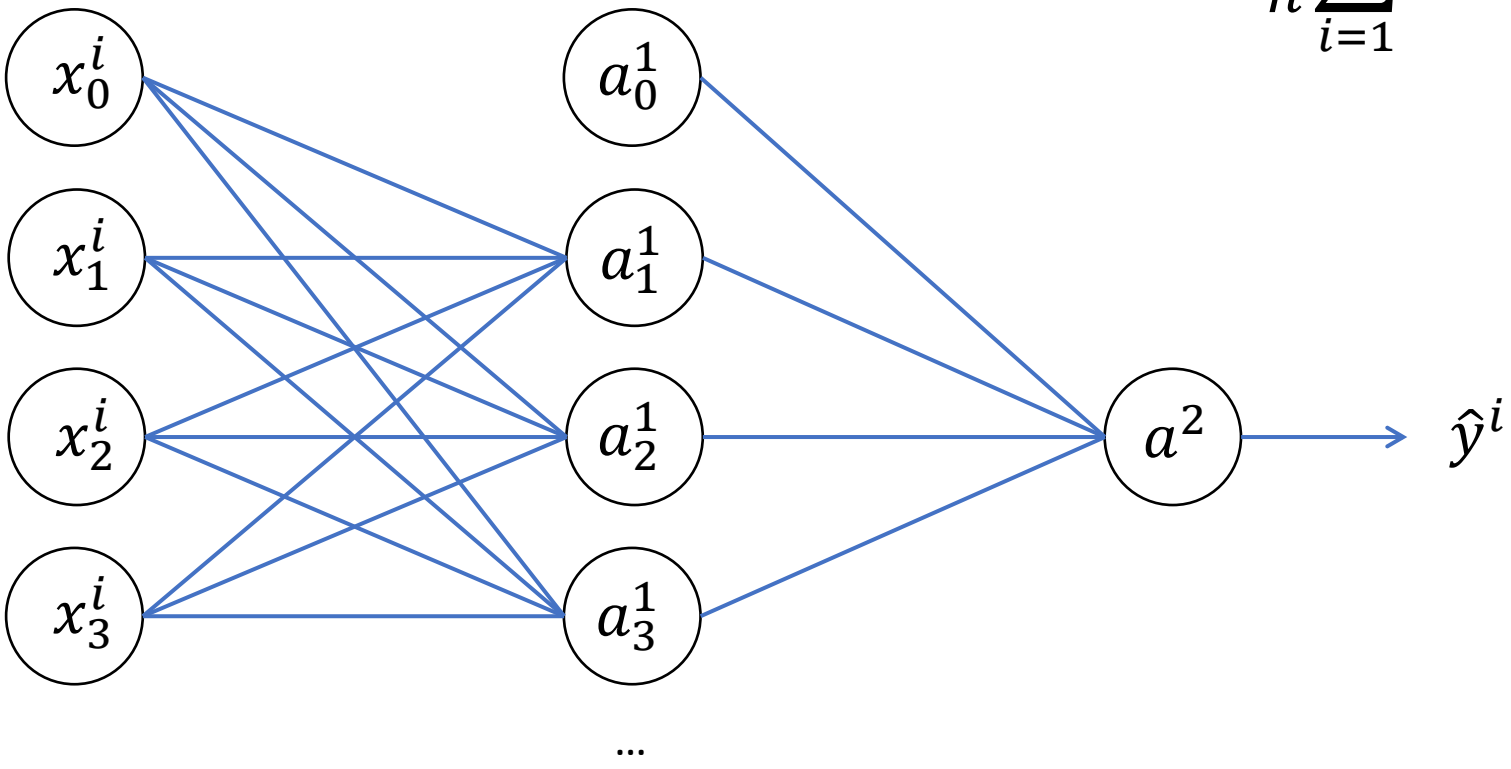
It defines the error the model makes.



loss function

For a regression task we use the **Mean Squared Error (MSE)** also known as L2 loss.

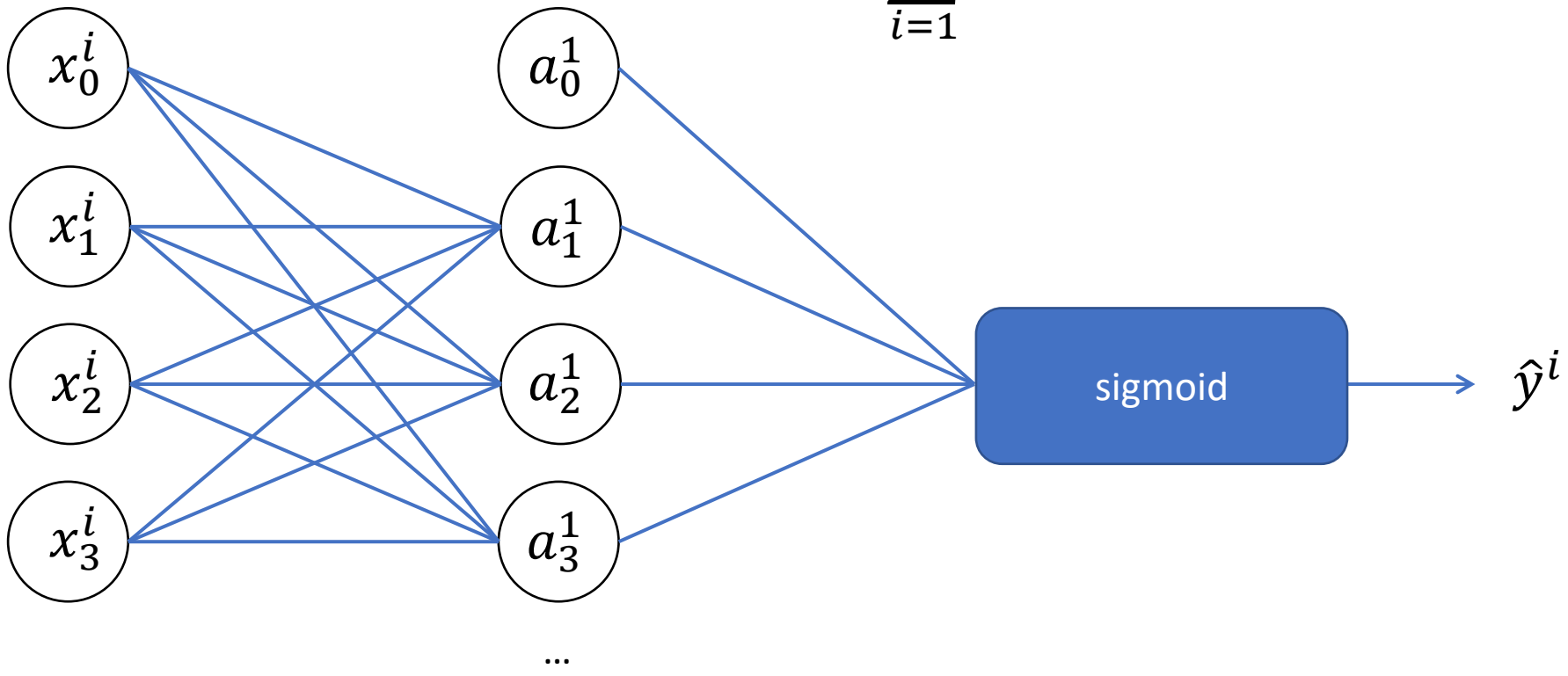
$$\frac{1}{n}\sum_{i=1}^{n}(y^i - \hat{y}^i)^2$$



loss function

For a two-class (or binary) classification task we use the **Binary Cross Entropy (BCE)** loss.

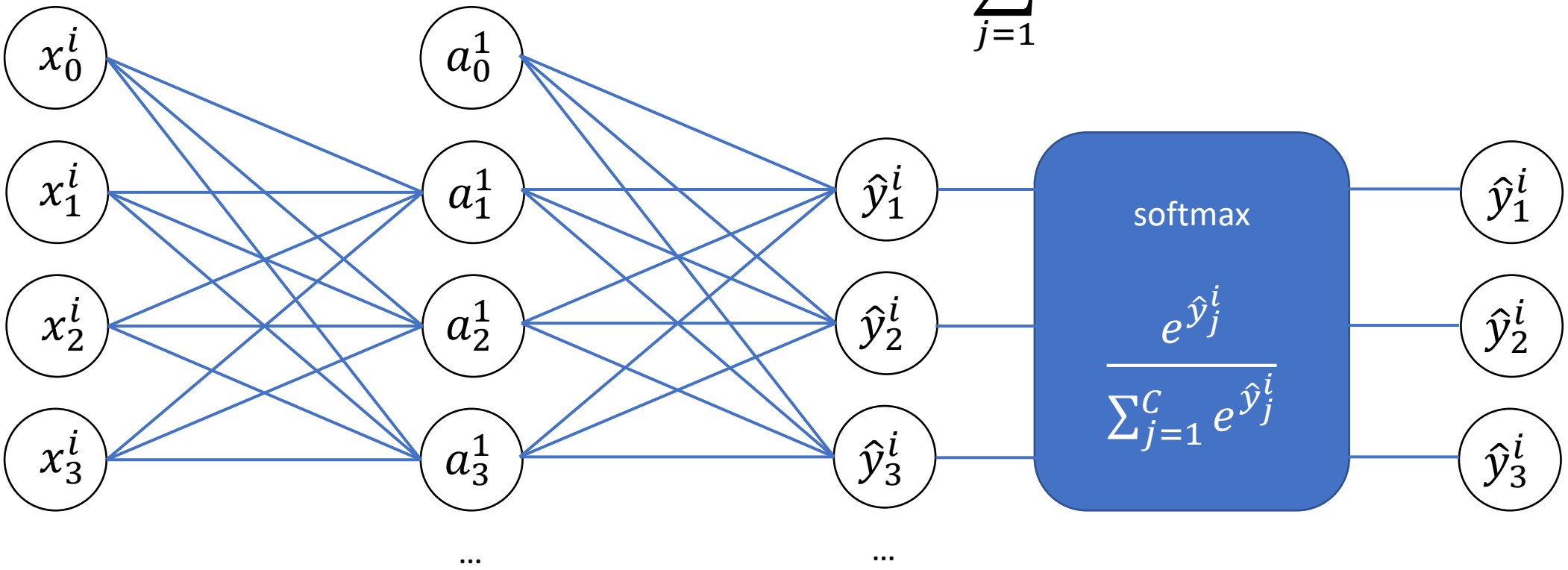
$$-\sum_{i=1}^{n}\left(y^i \log\left(\hat{y}^i\right) + \left(1 - y^i\right)\log\left(1 - \hat{y}^i\right)\right)$$



loss function

For a multi-class classification task we use the **Categorical Cross Entropy (CE)** loss.

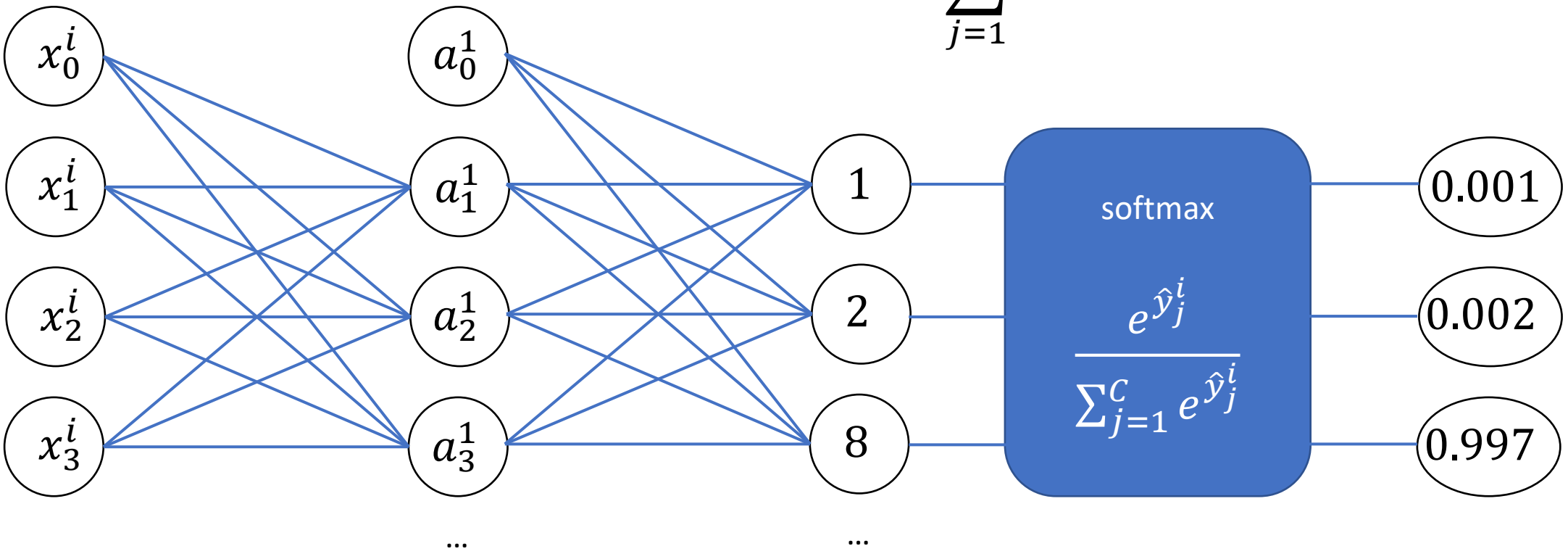
$$-\sum_{j=1}^{C}(y^i \log(\hat{y}^i)$$



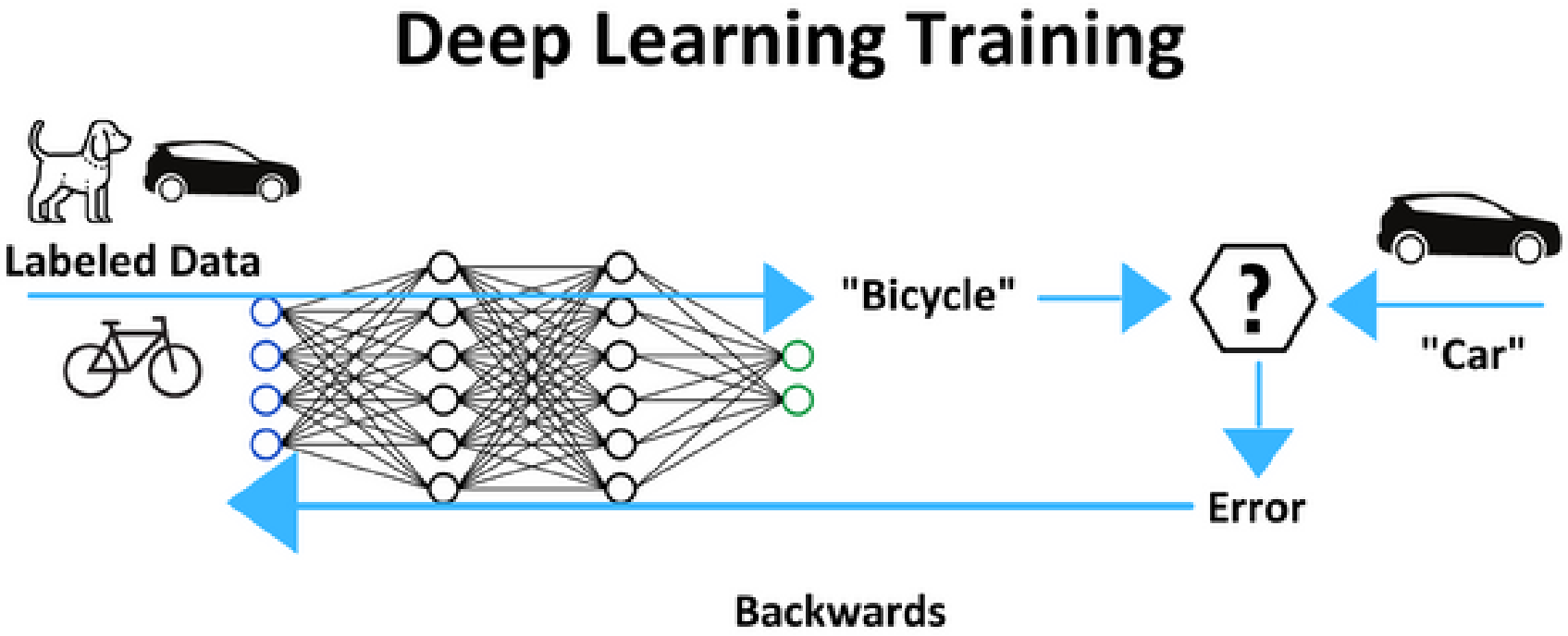
loss function

For a multi-class classification task we use the **Categorical Cross Entropy (CE)** loss.

$$-\sum_{j=1}^{C}(y^i \log(\hat{y}^i))$$



neural network training



The algorithm used to train a neural network is called **backpropagation**.

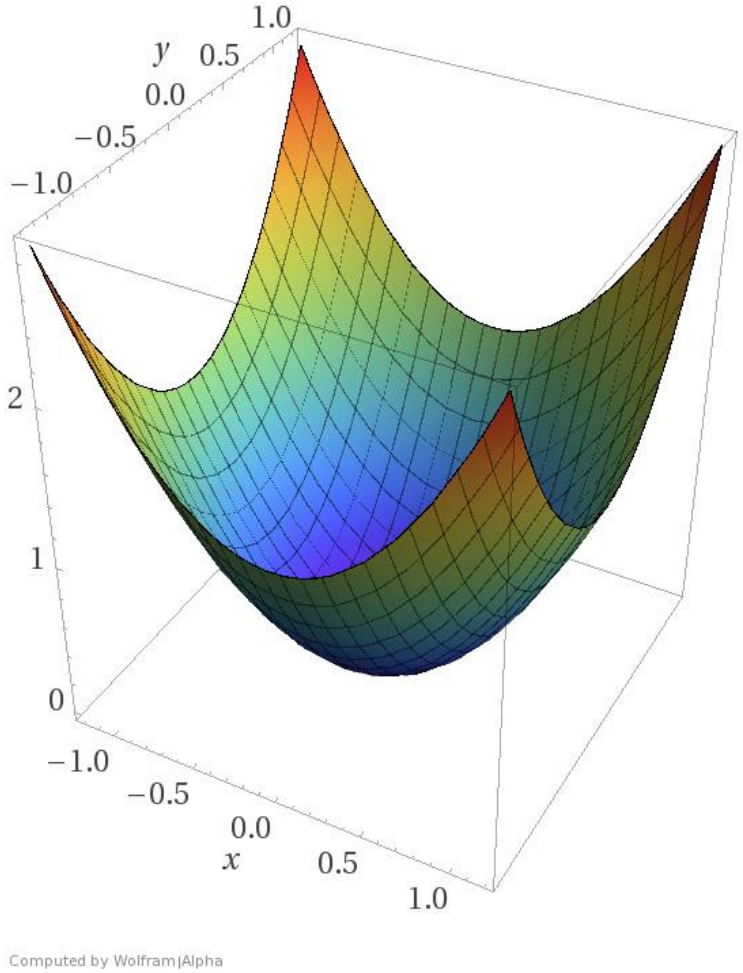
It fine-tunes the weights of a neural net based on the error rate (i.e. loss) obtained in the previous **epoch** (i.e. **iteration**).

backpropagation

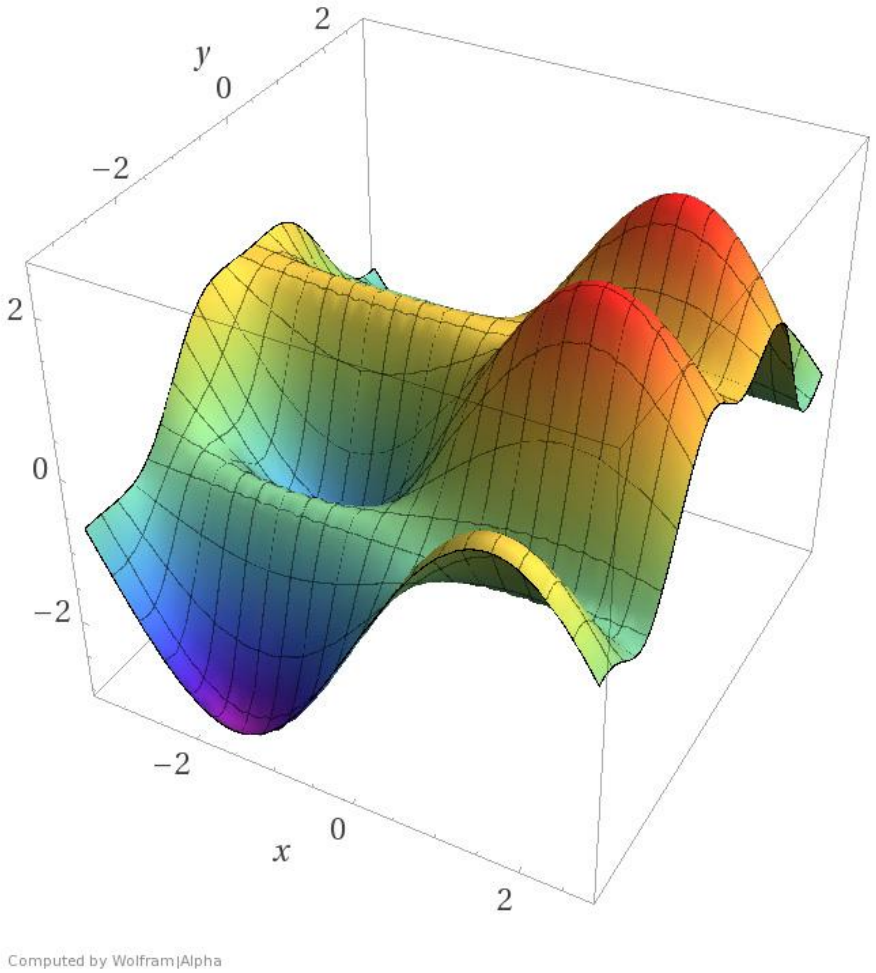
https://www.youtube.com/watch?v=Ilg3gGewQ5U&t=548s&ab_channel=3Blue1Brown

loss function

Training is
a challenge.



logistic regression

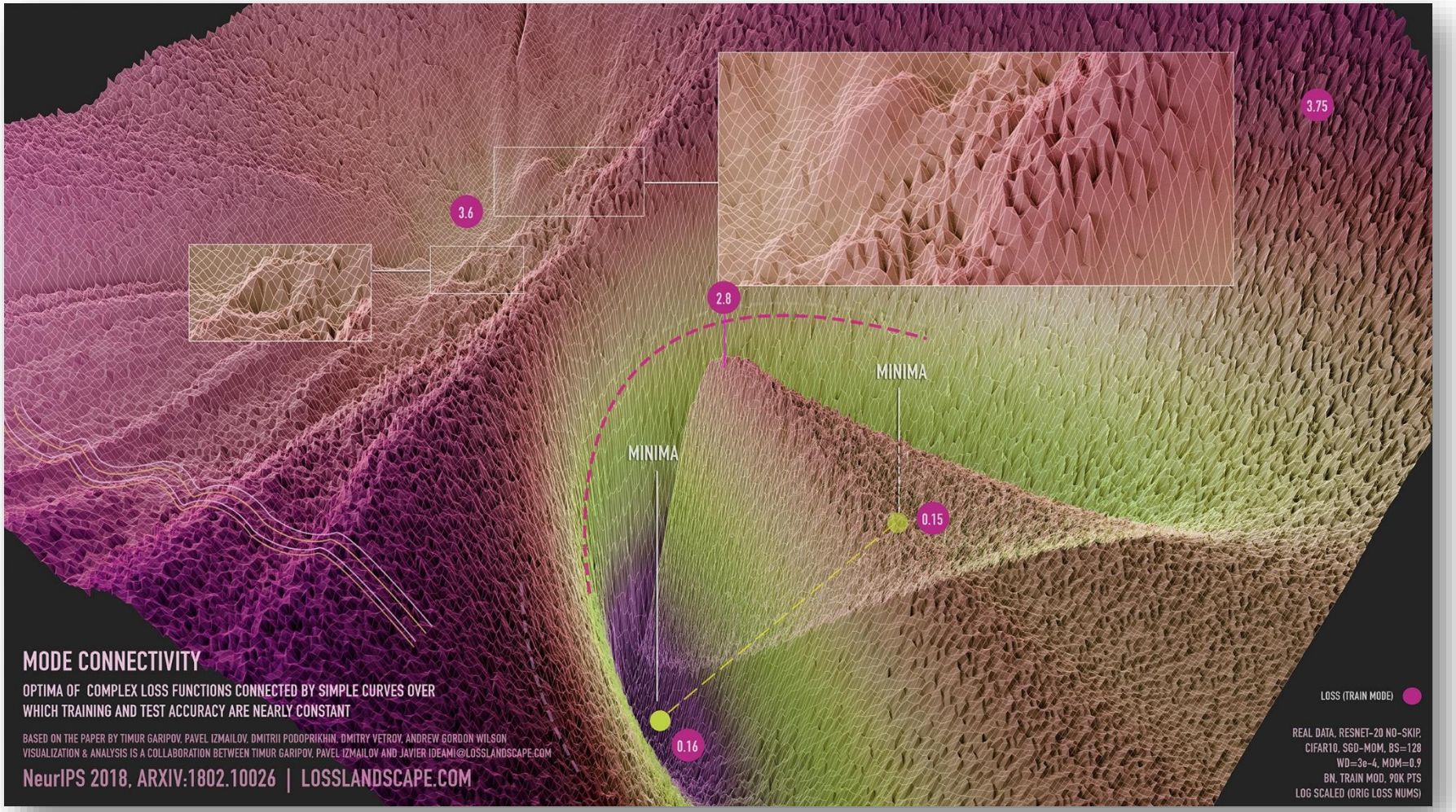


(small) neural network

loss function

Training is
a challenge.

Loss surface can
be highly
non-convex.



(deep neural network)

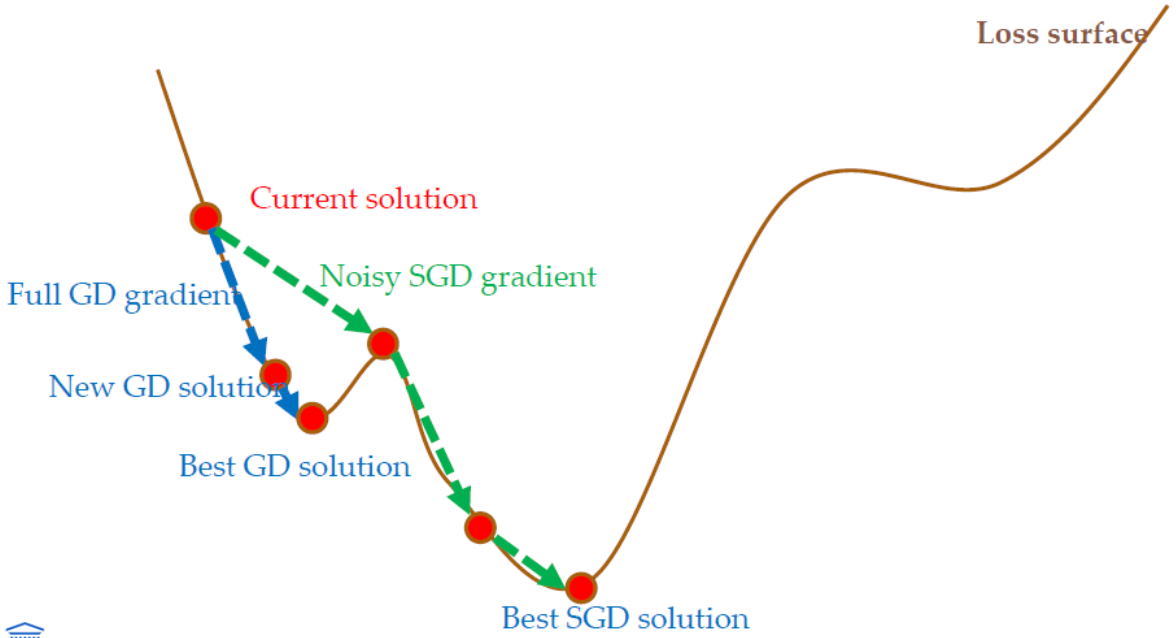
https://izmailovpavel.github.io/curves_blogpost/

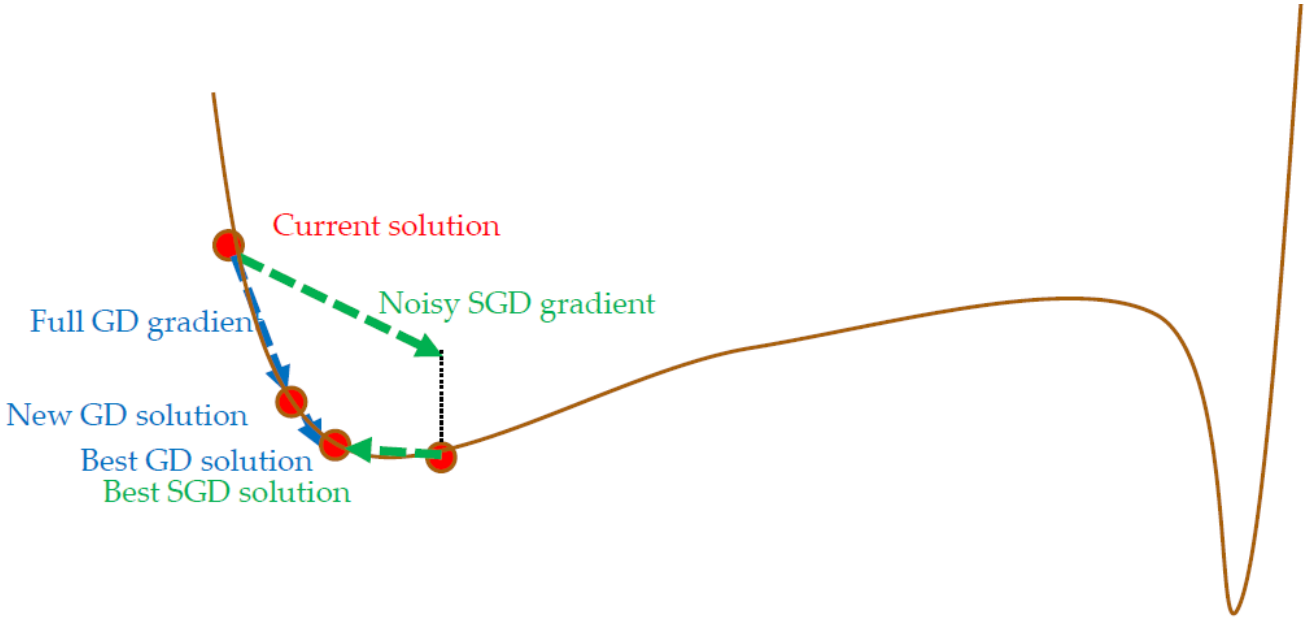
stochastic gradient descent (SGD)

Gradient descent (GD) takes into consideration the complete training dataset through every iteration, which may become time-consuming and computationally expensive.

GD may converge to a local minimum and may not be able to get away from it.

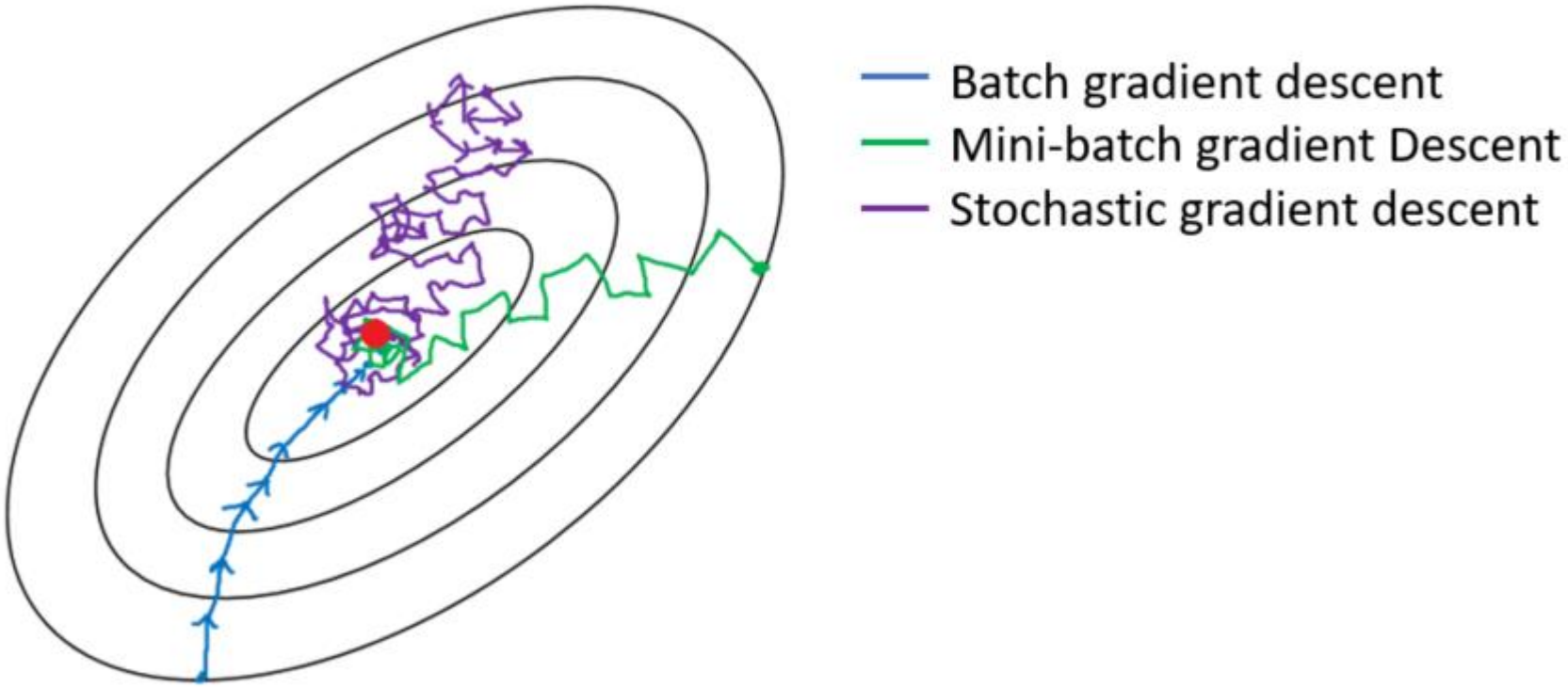
SGD takes into consideration one random data point through every iteration, which may lead to noisier results than the GD and may allow escaping local minima.





mini batch gradient descent

Mini batch gradient descent is a cross-over between GD and SGD. Instead of iterating through the entire dataset (GD) or one observation (SGD), we split the dataset into small subsets (**batches**) and compute the gradients for each batch.



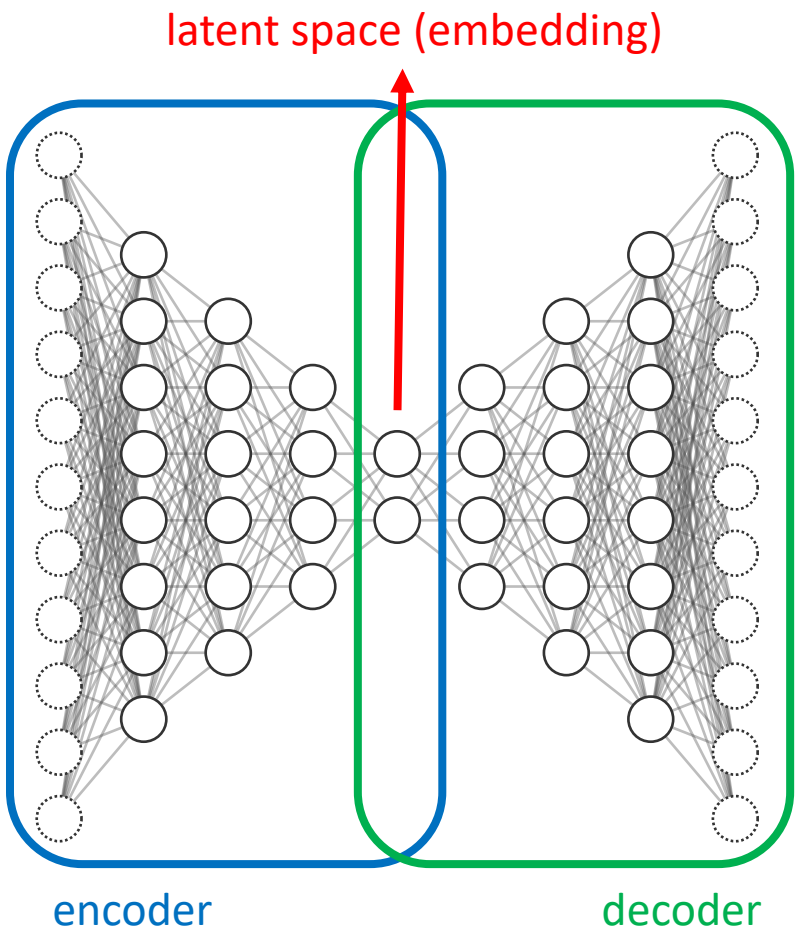
encoder – decoder (autoencoder)

An **autoencoder** architecture learns an efficient embeddings of unlabeled data.

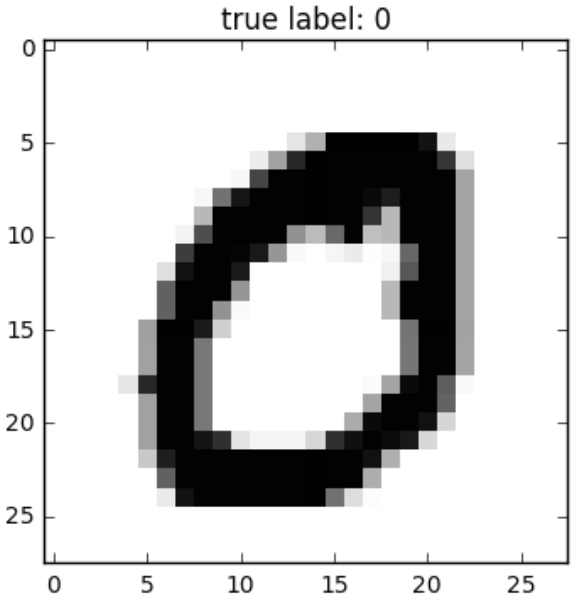
It consists of two parts, an **encoder**, and a **decoder**.

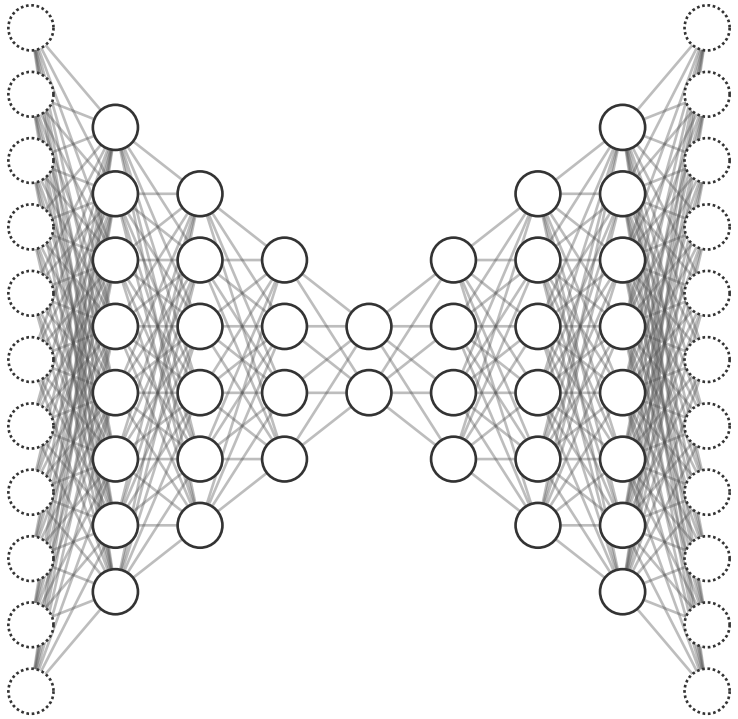
The encoder compresses the data from a higher-dimensional space to a lower-dimensional space (also called the **latent space**)

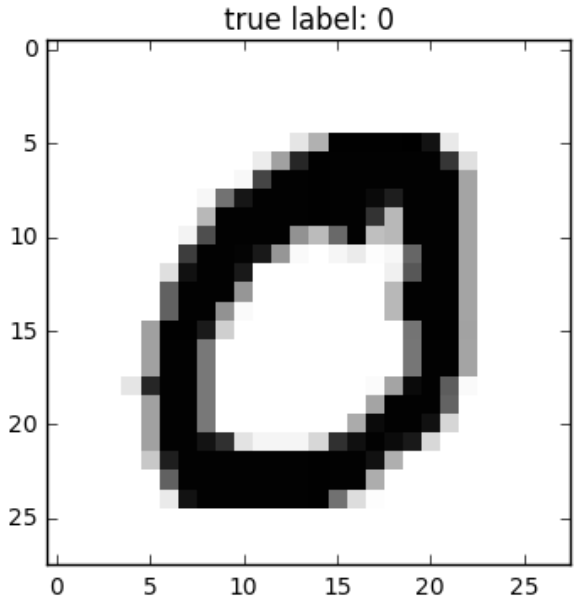
The decoder does the opposite i.e., convert the latent space back to higher-dimensional space.



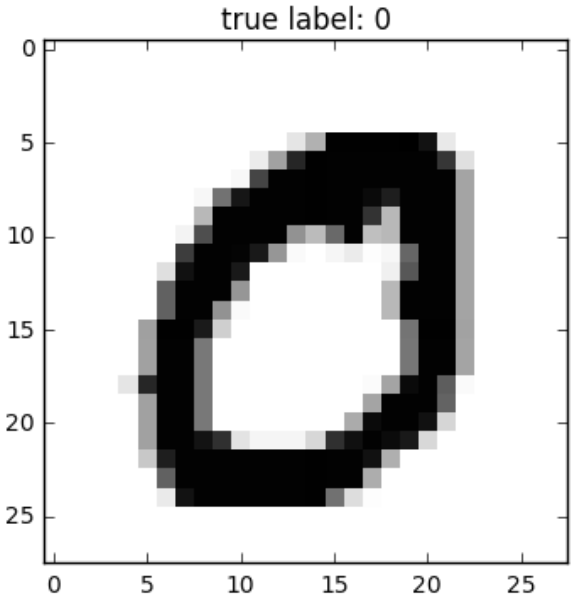
encoder – decoder (autoencoder)

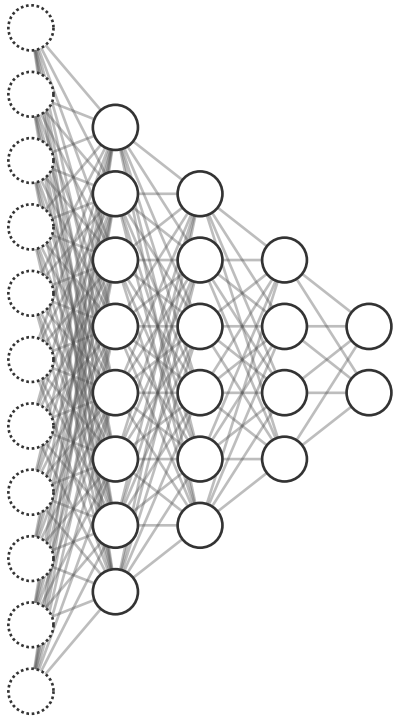


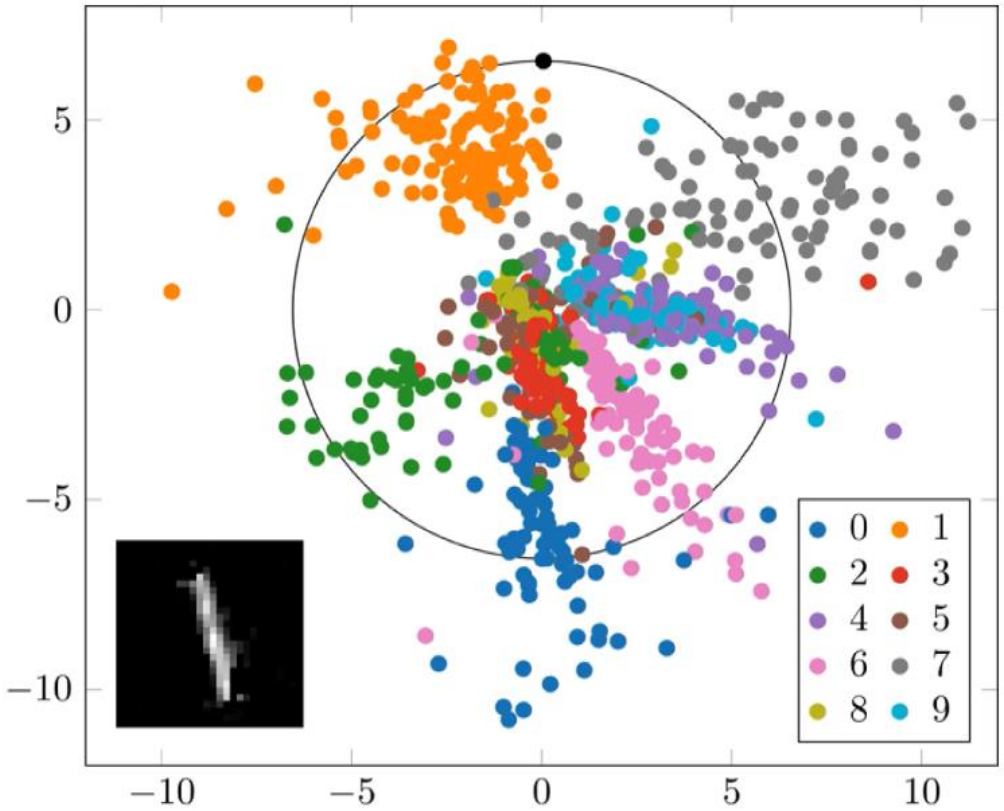




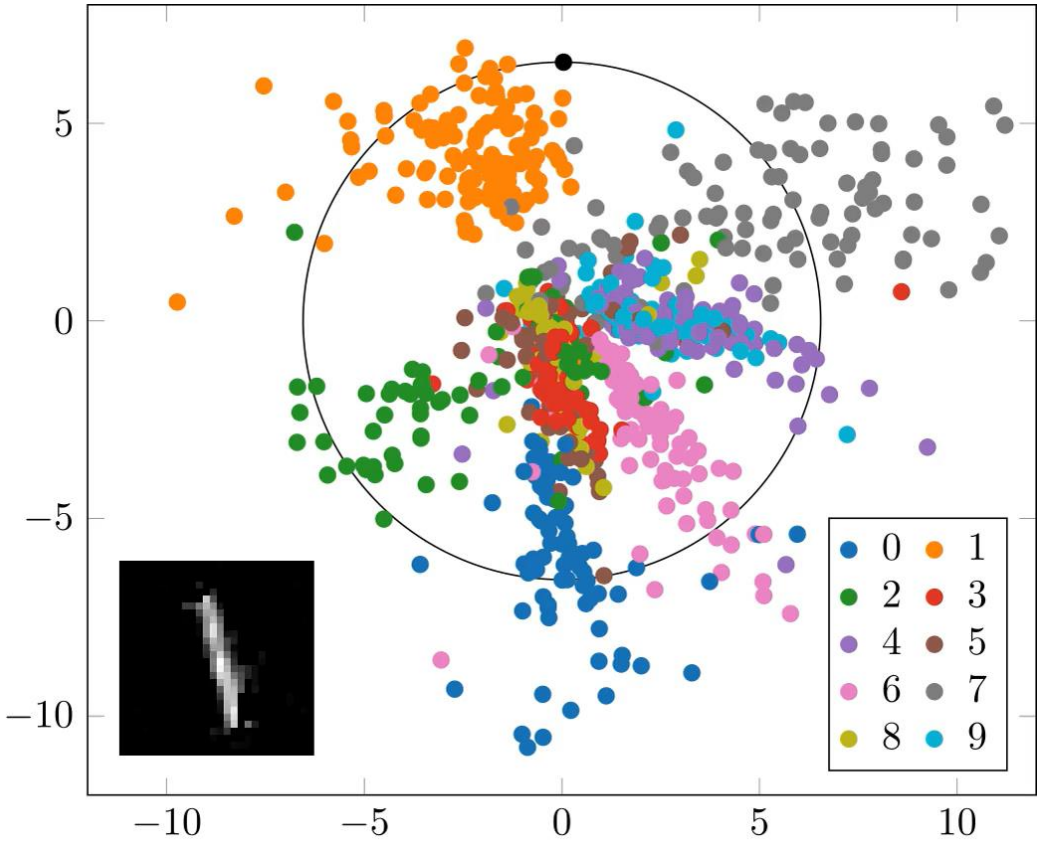
encoder – decoder (autoencoder)

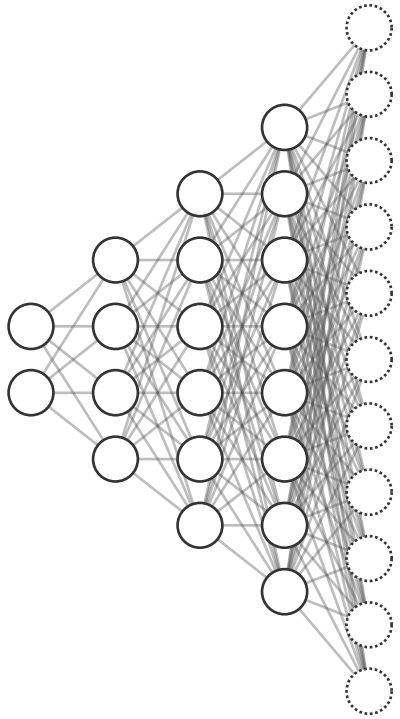


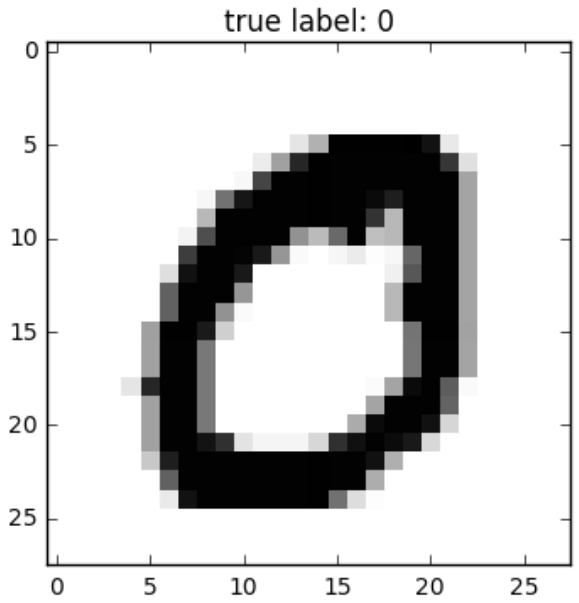




encoder – decoder (autoencoder)



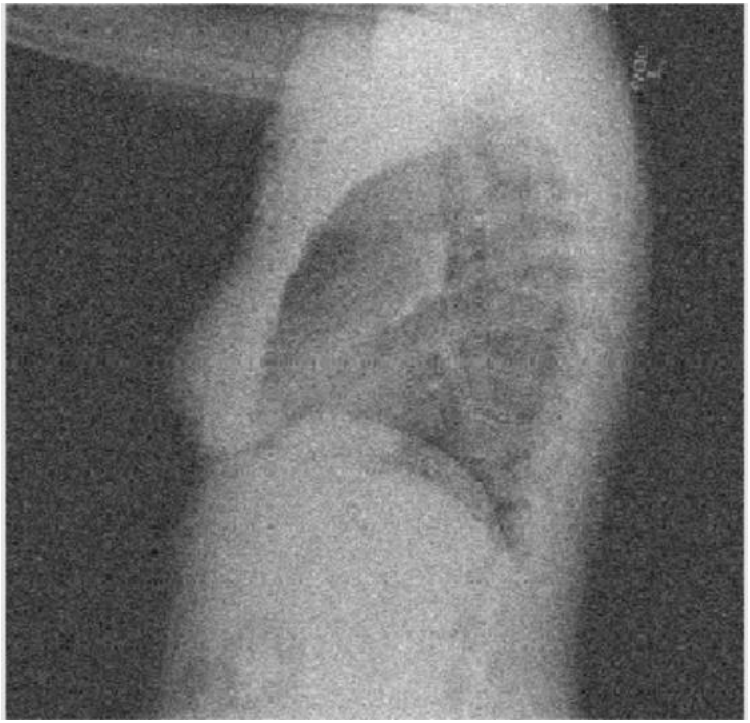


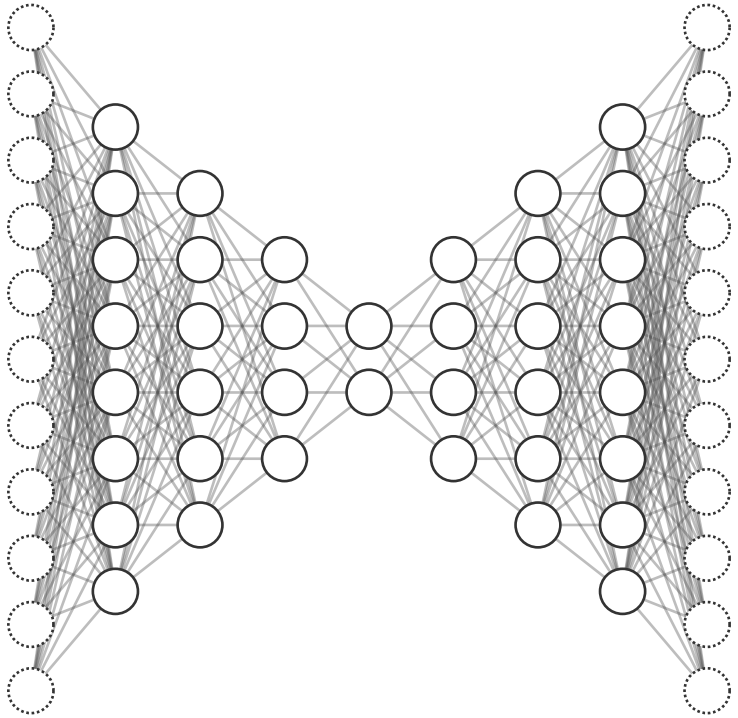


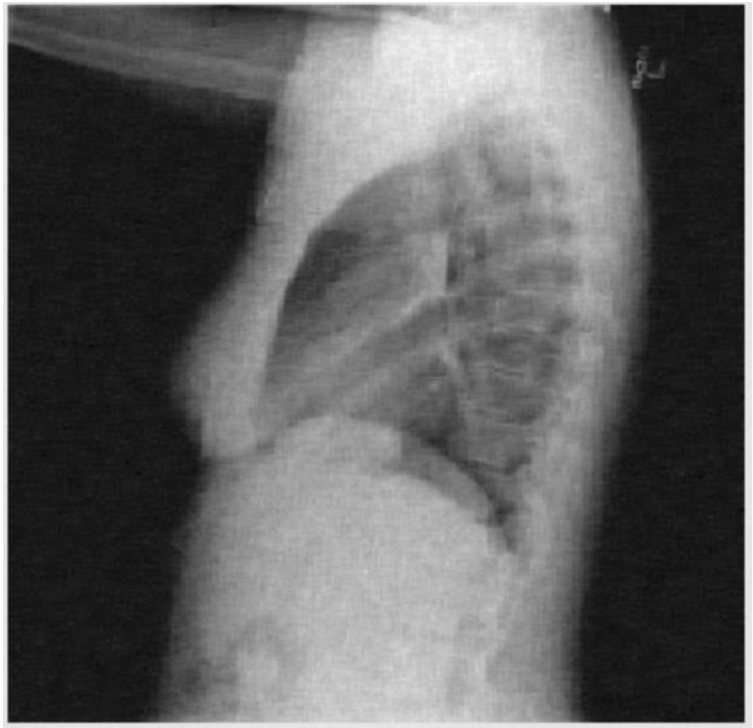
encoder – decoder (denoising autoencoder)

Autoencoders can easily be trained to **denoise** image data, expression data, …

The principle is to force the latent space layer capture the most robust features by reconstructing a clean input from a corrupted one.



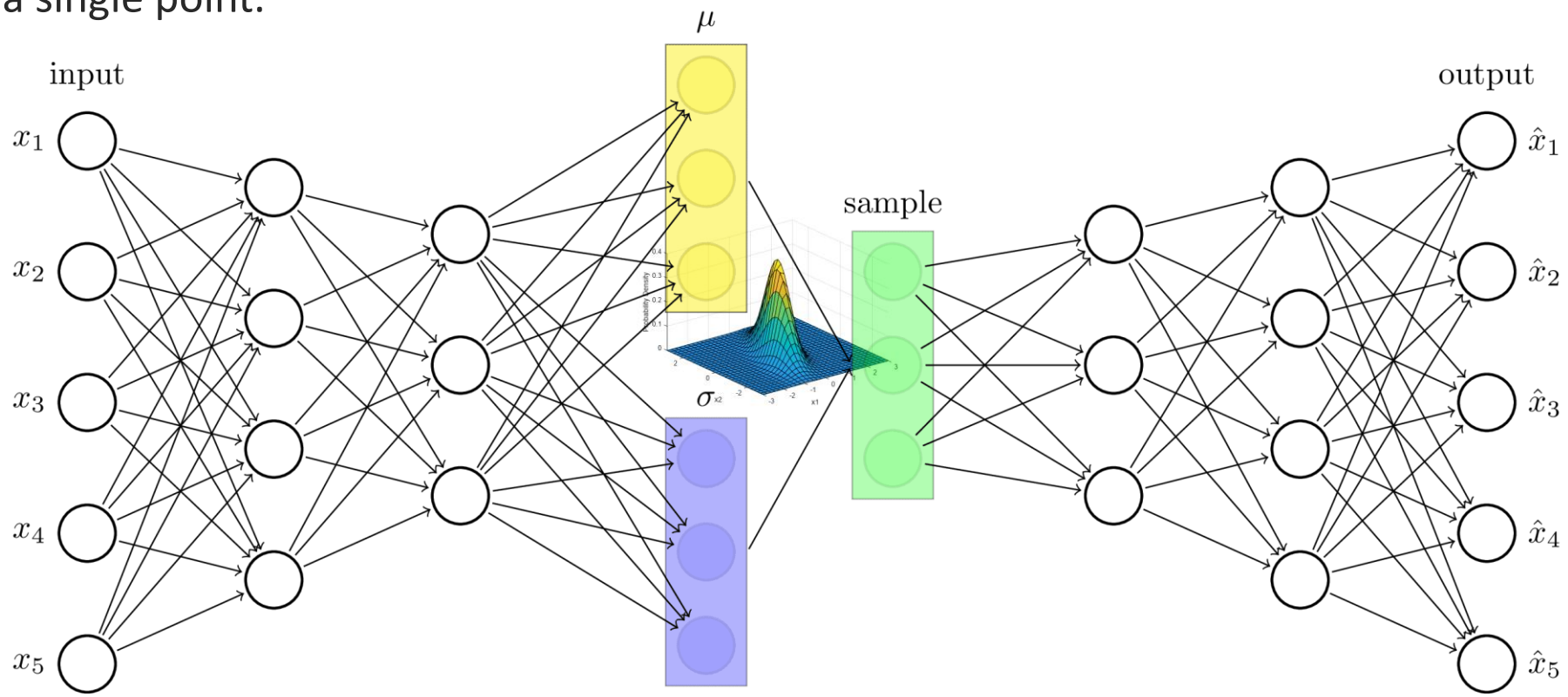




variational autoencoder

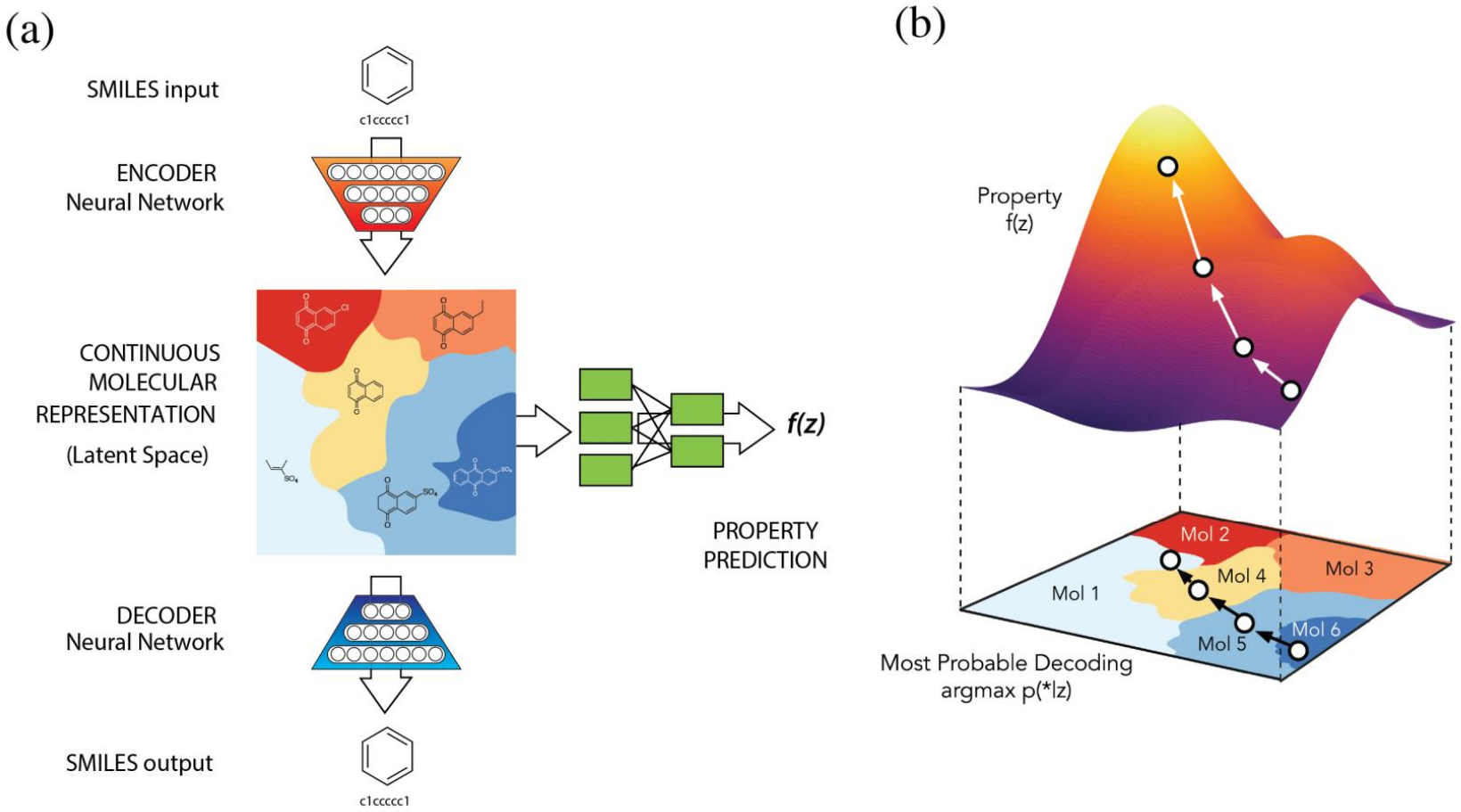
In an **autoencoder** the latent variable is not regularized, picking a random latent variable can generate garbage output.

A **variational autoencoder** solves this by mapping the input image to a distribution in the latent space, rather than a single point.



variational autoencoder

Representing chemicals in latent space to search for chemicals that optimize some function (e.g. maximize activity, or minimize toxicity).



Gómez-Bombarelli et al. ACS Central Science (2019)