

# *Statistical learning and deep artificial neural networks: selected applications in proteomics*



Toni Giorgino

[toni.giorgino@cnr.it](mailto:toni.giorgino@cnr.it)

[www.giorginolab.it](http://www.giorginolab.it)

@giorginolab

[@giorginolab@mstdn.science](https://@giorginolab@mstdn.science)



# Summary

- Part I: basics of artificial neural networks
  - Statistical learning basics
  - The artificial neuron
  - Multi-layer neural networks
  - Training, test, validation and pitfalls
- Part II: advanced ANN architectures
  - Convolutional NNs
  - Recurrent NNs
  - Autoencoders
  - Adversarial neural networks
  - Transformers

# Summary

- Part III: applications
  - Pocket finding
  - Structure-based affinity ranking
  - Protein structure prediction
  - General-purpose sequence-based transformers
- Appendix
  - Merck's challenge
  - Datasets
  - Resources
  - Literature

# Course objectives

1. Know what an artificial NN is
    - What problem does it solve?
    - Advantages and pitfalls
  2. What DNN variations are there and what for?
  3. How is the proteomics community using them
- Grasp the essence of current research  
(For the computationally-oriented: experiment in autonomy)

# **Part I**

**Basics of artificial  
neural networks**

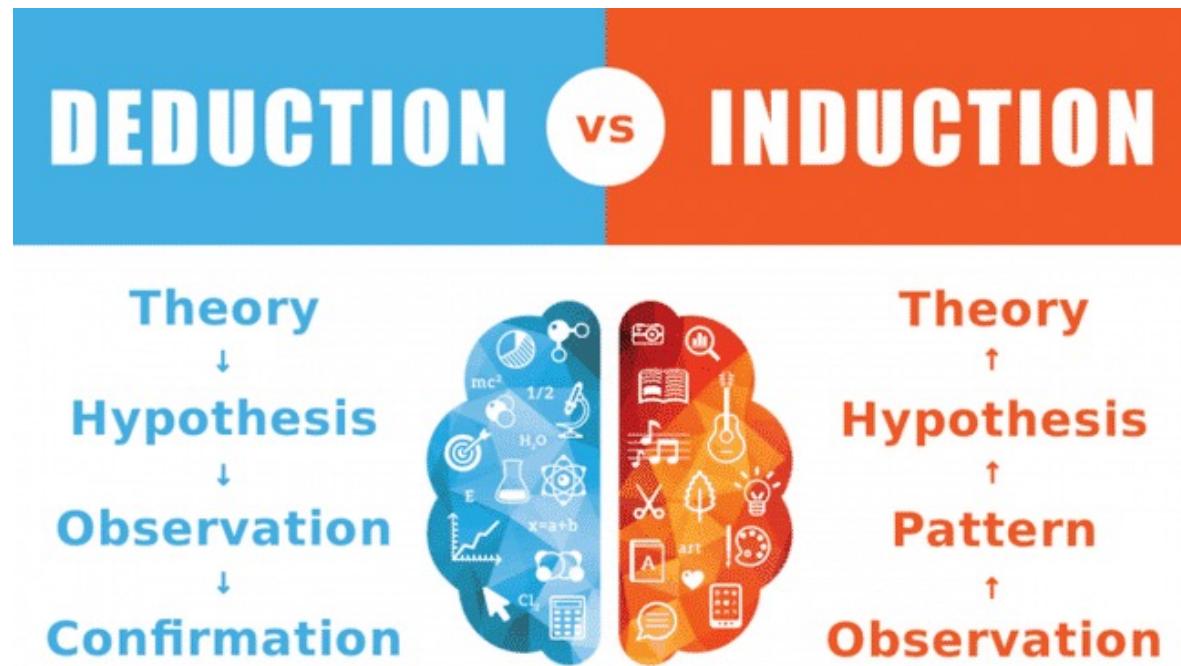
# Inference = reasoning

Given the theory

“All humans are mortal.  
Greeks are humans.  
*Therefore, Greeks are mortal.*”

Given the data

“I’ve never seen any donkey fly.  
I’ve observed 10 donkeys.  
*Therefore, donkeys don’t fly.*”

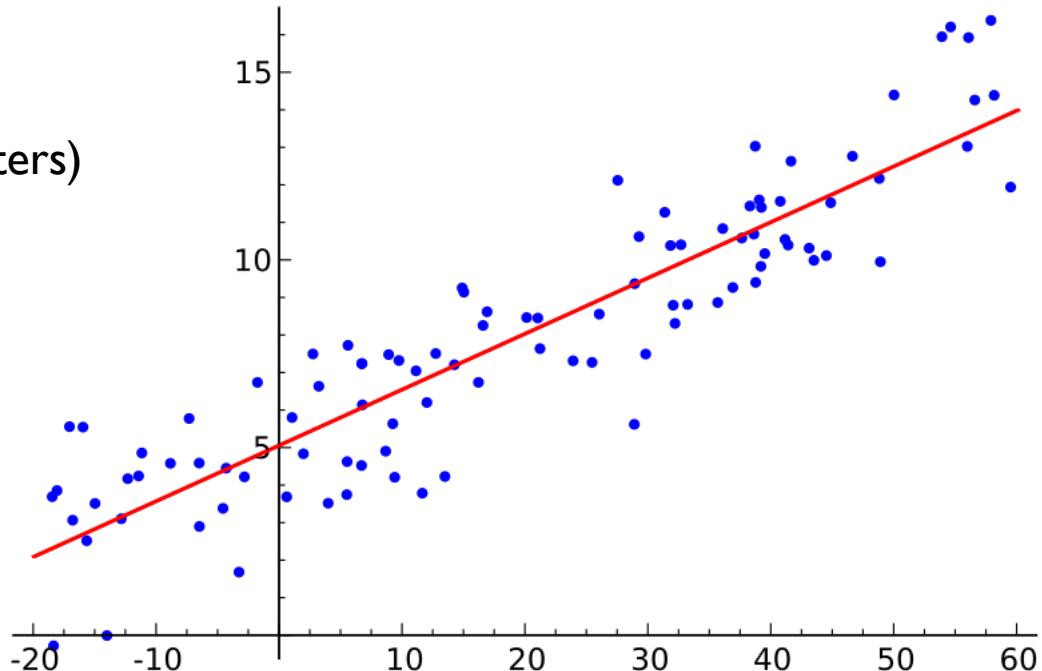


# However

- For deductive reasoning:
  - Are the premises correct?
  - If so, the conclusion is **certain**
  - Upon observing a counter-example,  
we'd need to revise the premises/theory
- For inductive reasoning:
  - The induction is only **probabilistic**
  - In presence of noise: *statistical inference*

# Machine learning

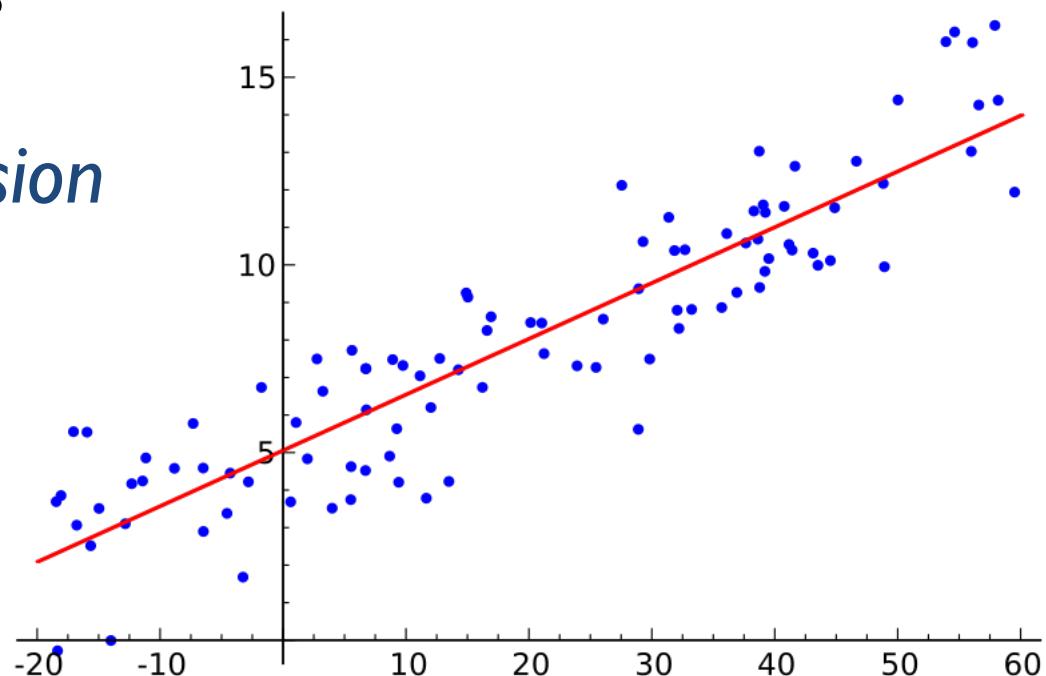
- Collect large amounts of data
- Let computers do the induction\*
- Research focuses on algorithms to do so
- Example: regression
  - Note how (1) we are inferring a rule, i.e. the specific relationship (parameters) but we assumed a functional form → power
  - (2) Once rule obtained, we make **predictions**

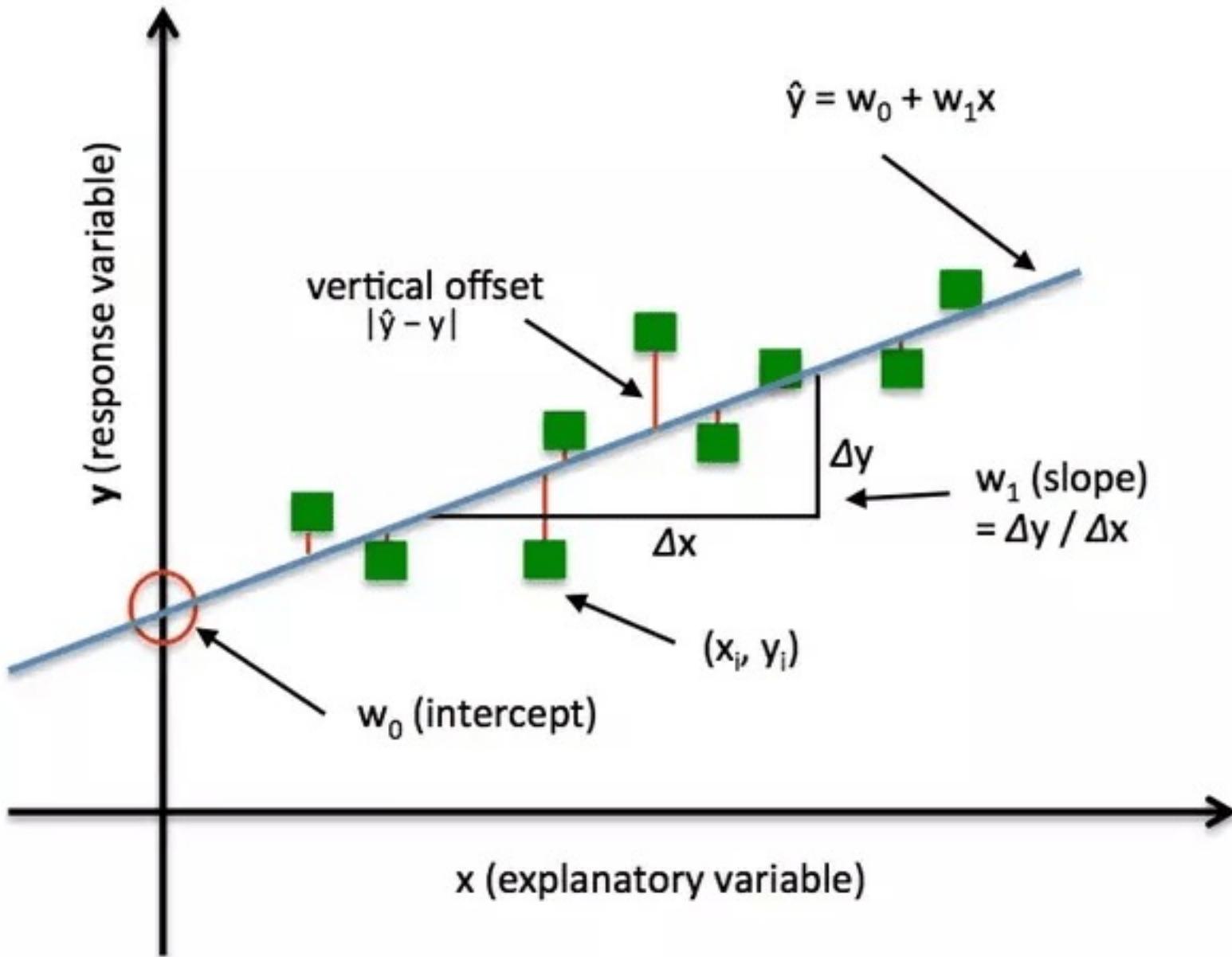


\* dangerous!

# Regression

- Regression (e.g. linear regression) is a form of learning.
- One assumes that data is generated (with noise) by a certain functional relation, and *induces* the missing parameters
- Real-valued: *regression*
- If discrete: called *classification*



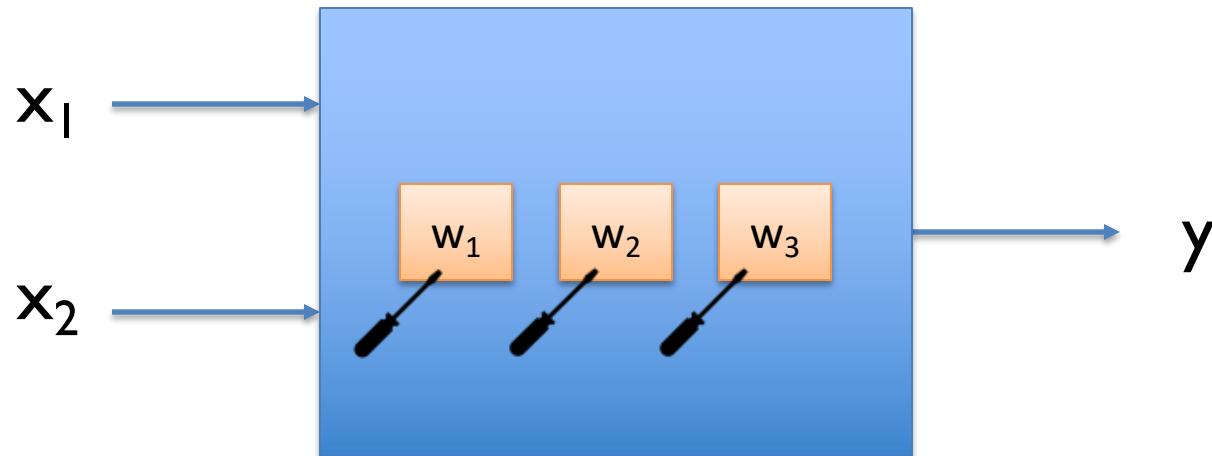


# Interpretability

- Assumptions are dangerous as they tend to be hidden
  - Is linearity the “best” model? What if up/down?
- The rules found by induction may be precious *per se!*
  - Example: CV risk increases with age and smoking
  - How much *per year?*
- **Interpretability** is important
  - One may want to sacrifice accuracy for it

# Regression as a box

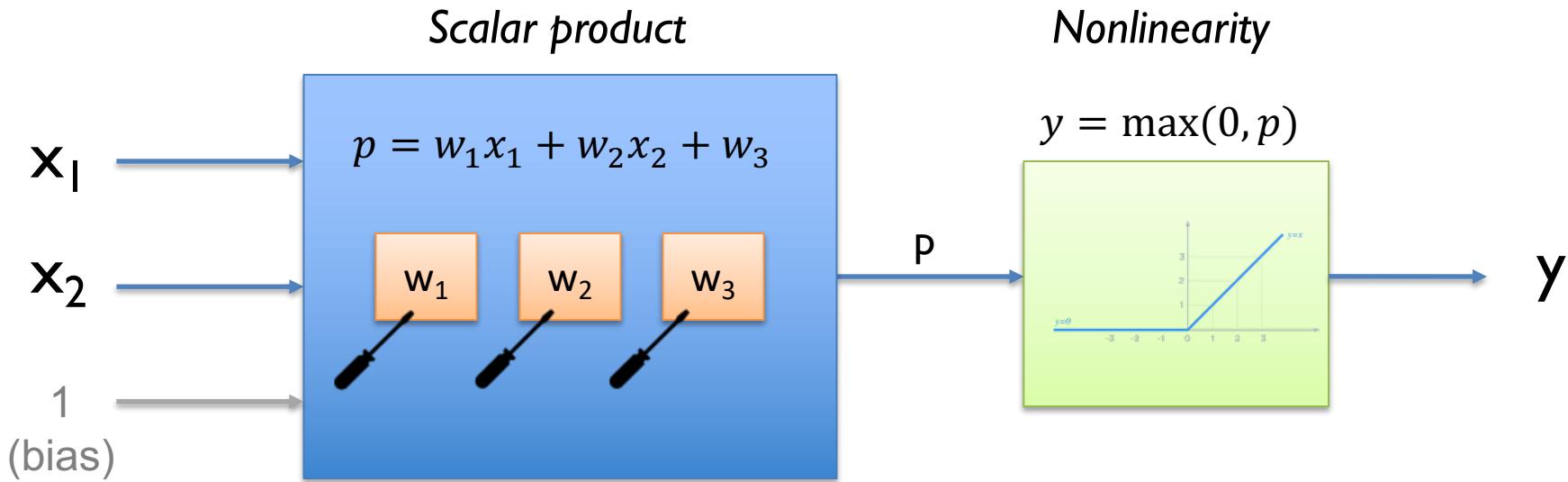
$$y(x_1, x_2 | w_1 \dots 3) = \underbrace{w_1 x_1 + w_2 x_2 + w_3}_{\text{A blue bracket under the equation}}$$



For now, *real numbers* in input and output

# An artificial neuron

- Adding a *nonlinear activation function* gives an *artificial neuron*
- It is a *mathematical model!*
- It is **not** a biological model!



# In other words...

Linear regression, 1 D →

$$y = wx + b$$

Linear regression,  $n$  D →

$$y = \mathbf{w} \cdot \mathbf{x} + b$$

One artificial neuron →

$$y = f(\mathbf{w} \cdot \mathbf{x} + b)$$

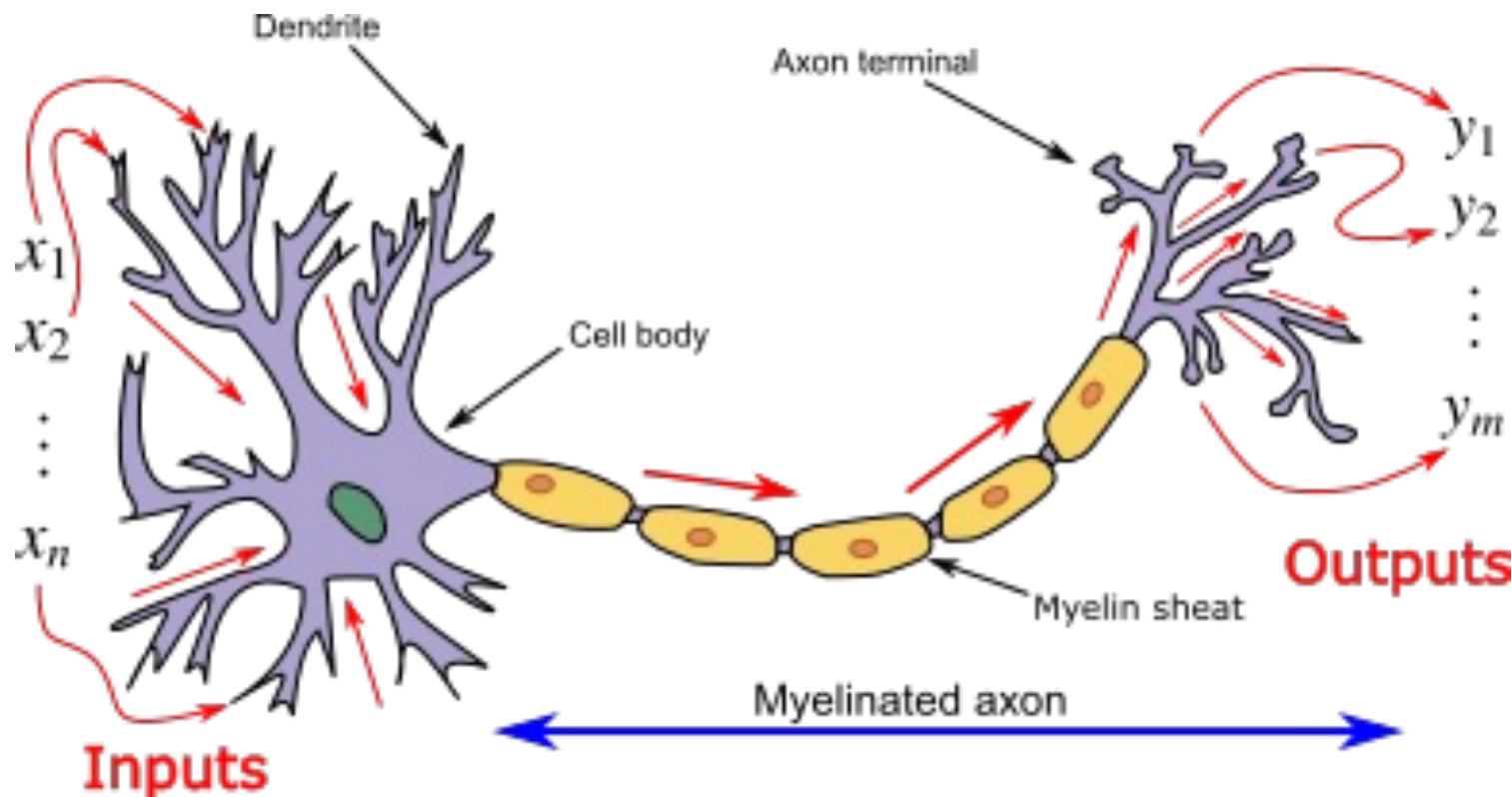


Activation function

Repeat: it is a **mathematical model**.

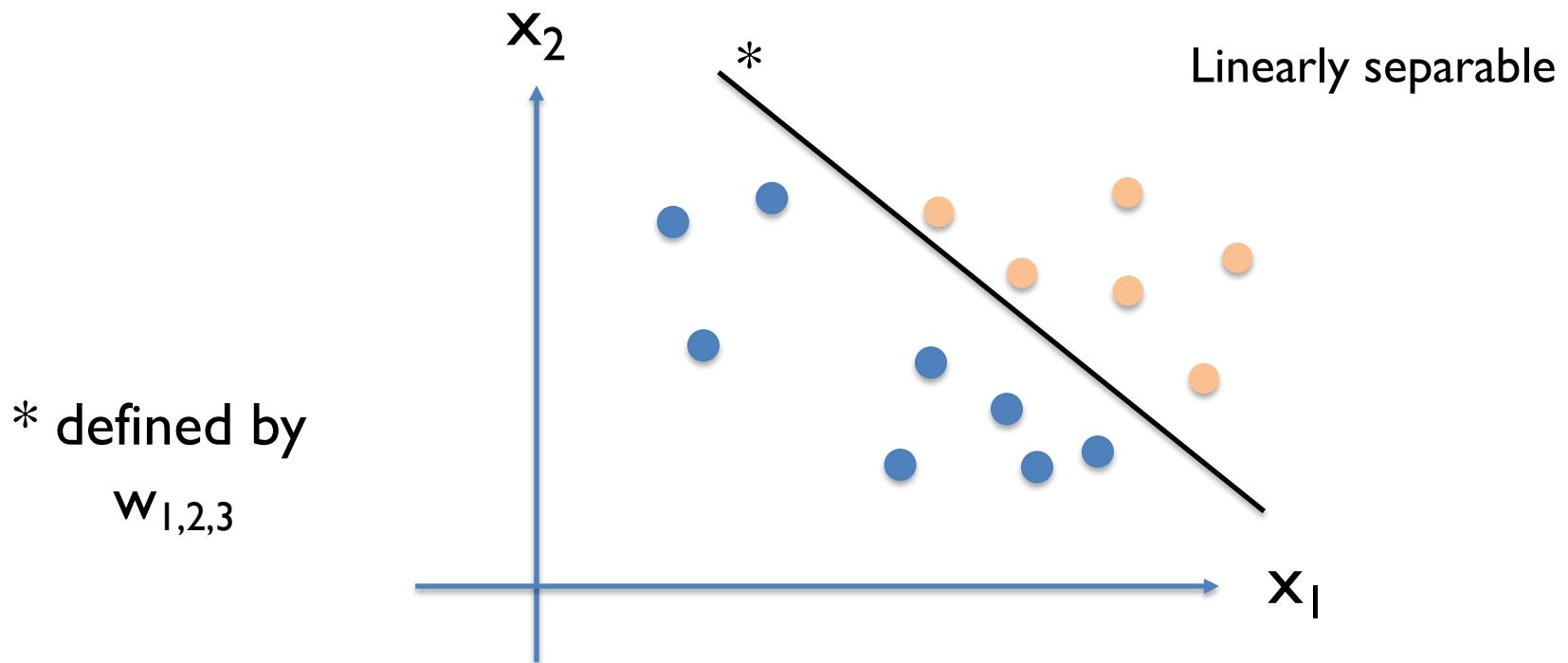
Not a model of biological significance.

(This parallel with actual neurons is a curiosity.)



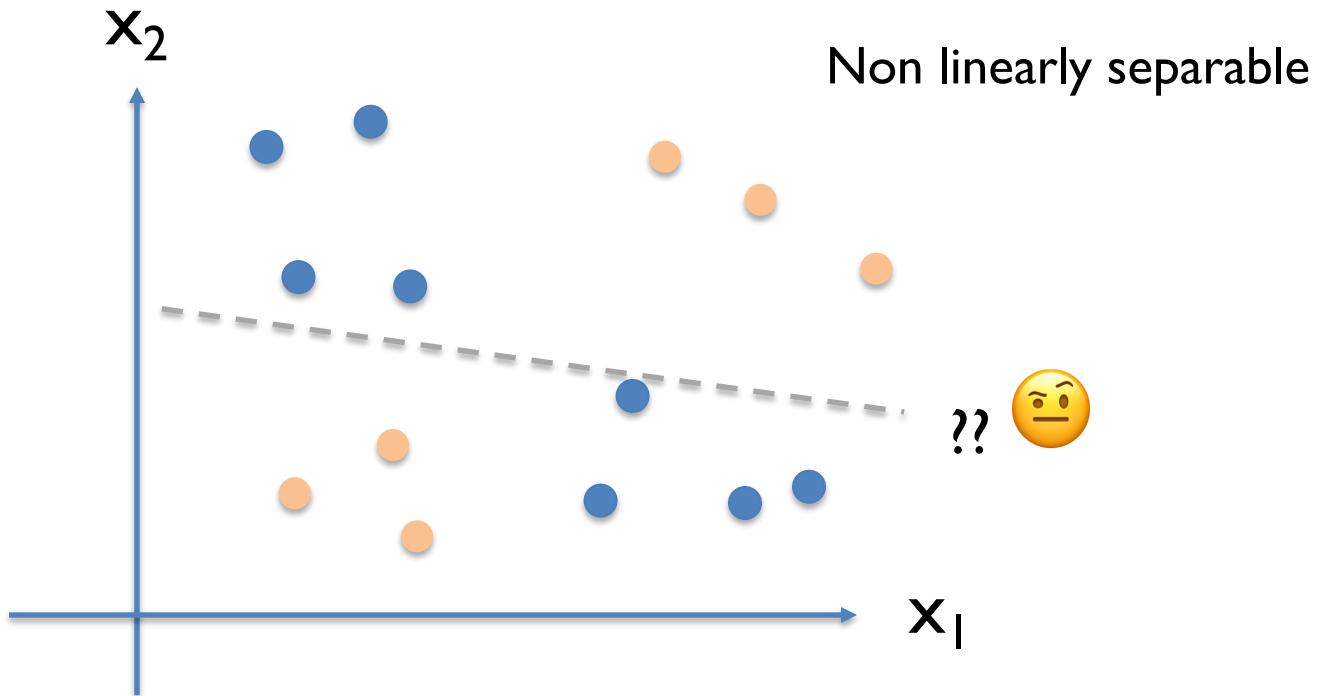
# Linear separability

A **single-neuron** model would only solve *linearly separable* problems.



# Linear separability

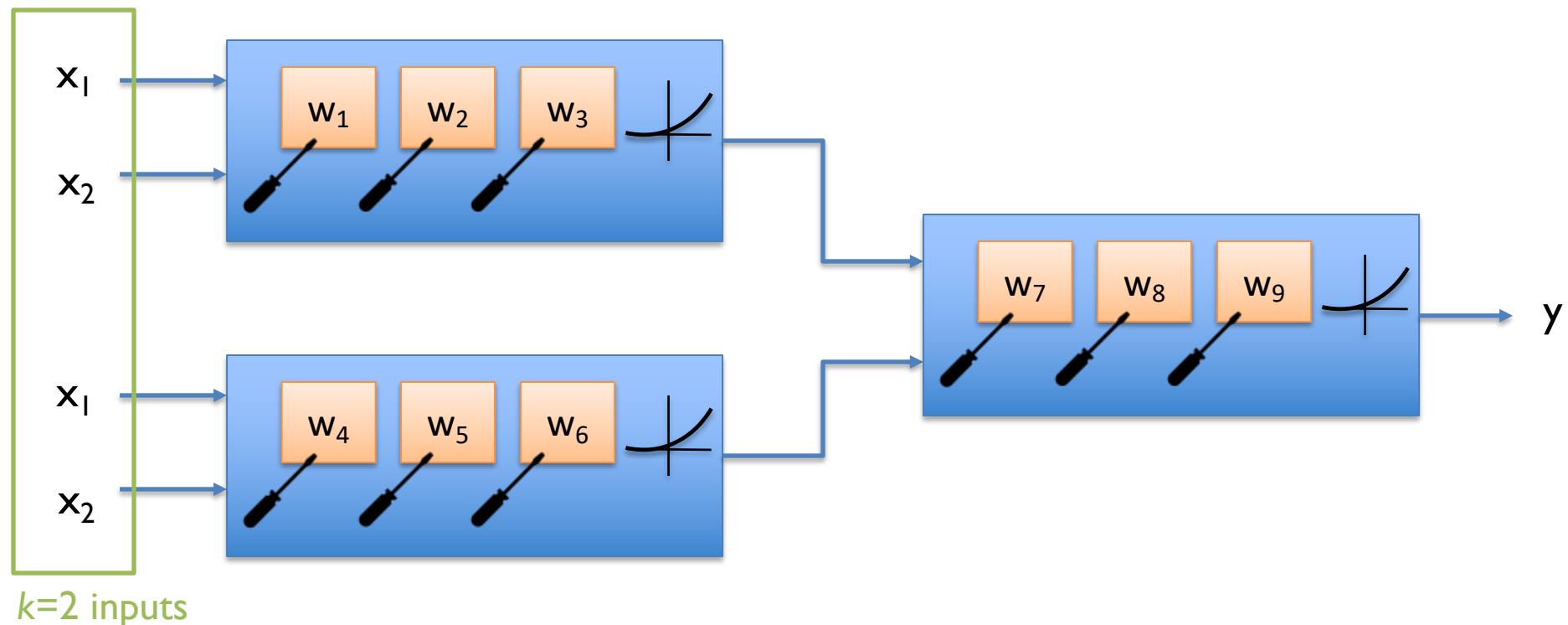
A **single-neuron** model would only solve *linearly separable* problems.



# Artificial neural networks (ANNs)

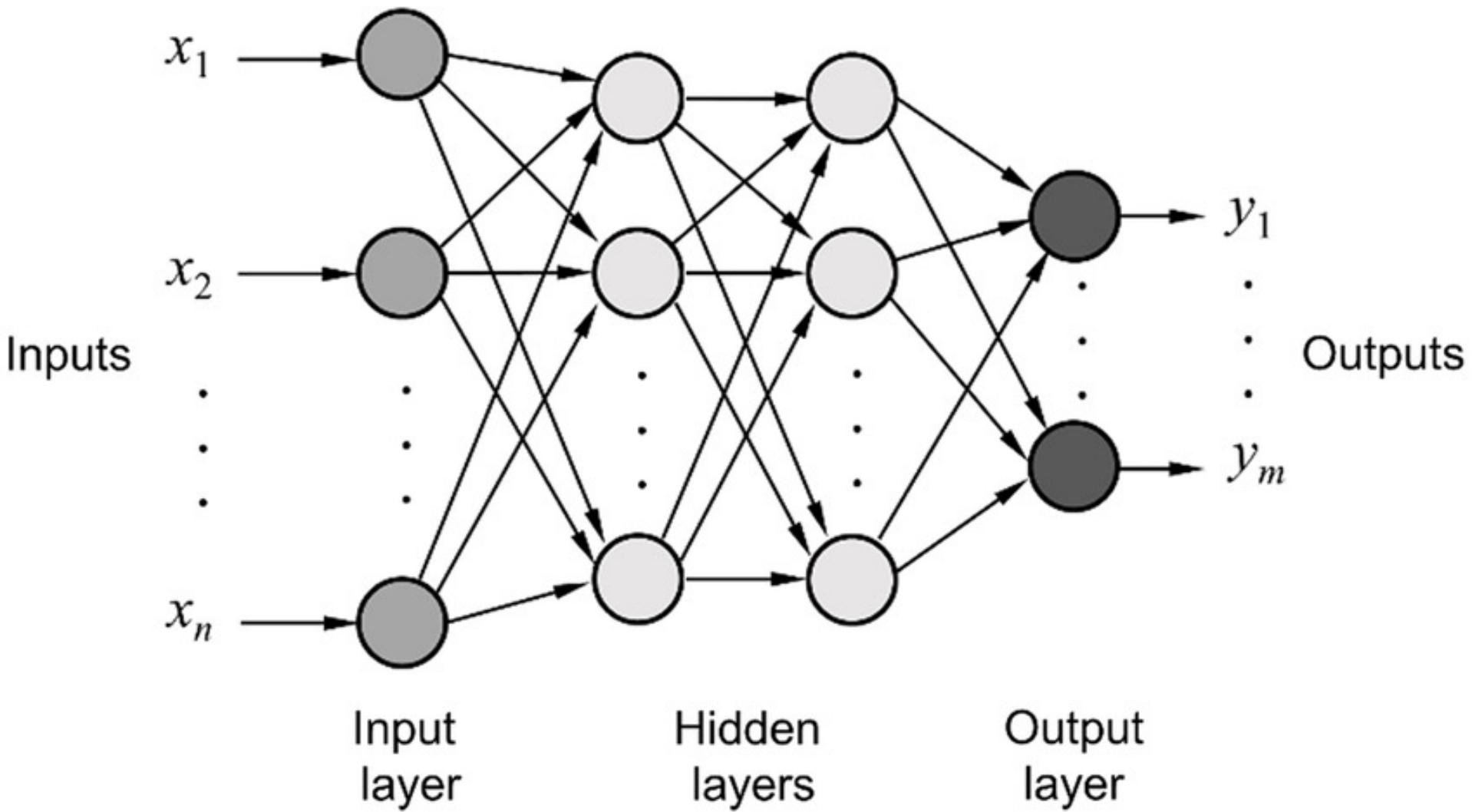
- The power of this model comes from its modularity.
- A **repeated arrangement** of neurons as a network can solve much more complex problems.
- Addition of more neurons will require tuning many more parameters → 🤪 computing cost, data hunger, overfitting

# An artificial neural network



$N=3$  neurons    $k=2$  inputs    $u=1$  output  
 $n_w = N(k+1) = 9$  weights   (in this case)

- An artificial neural network with sufficient complexity *and the correct weights* may approximate arbitrary functions
- The interest lies in the fact that one can find the “correct” weights with an automatic procedure...
- ...*given enough data*
- Engineering problem → data collection problem (at least in principle!)



(Note: here showing *fully connected* layers. They needn't be.)

# Training: loss function

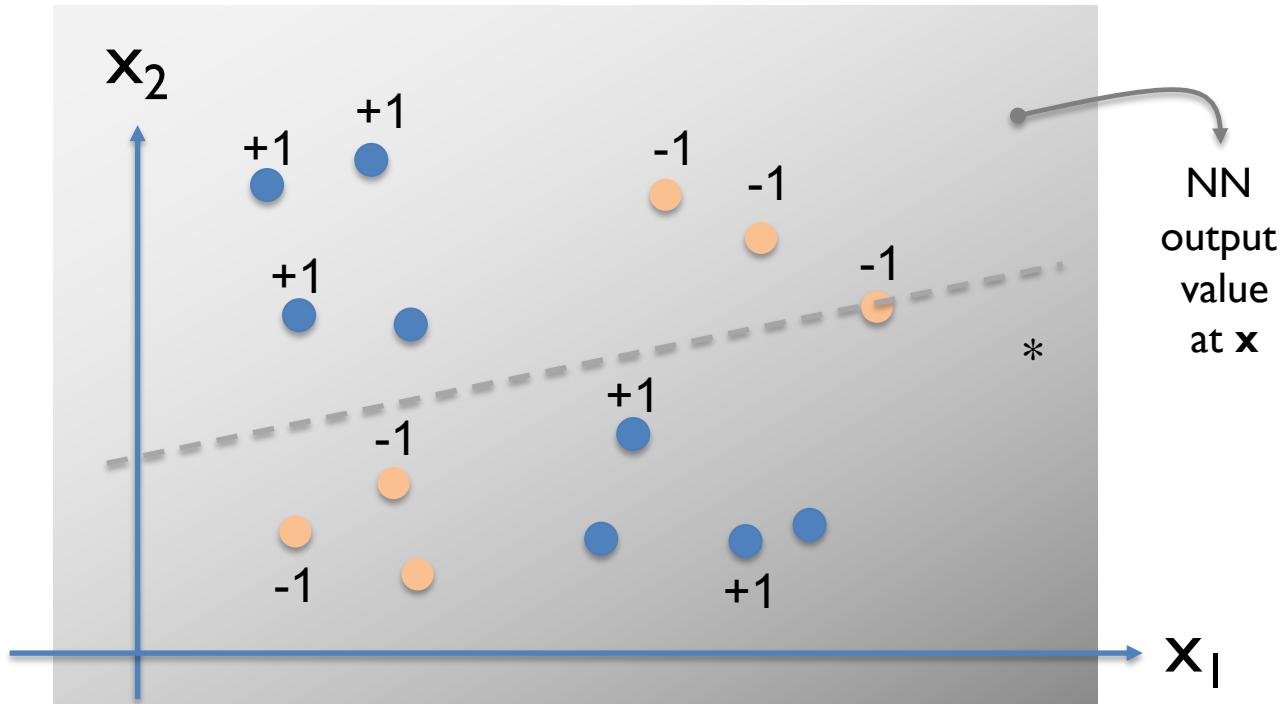
Labels

+1      -1

1. An ANN is trained on a (large) set of *labeled* examples
2. They are used as inputs.
3. They are compared with expected outputs (labels)
4. The squared difference is the *loss*

\* defined by

$w_{1,2,\dots,n}$



# Training: loss function

Red screwdriver icon pointing down.

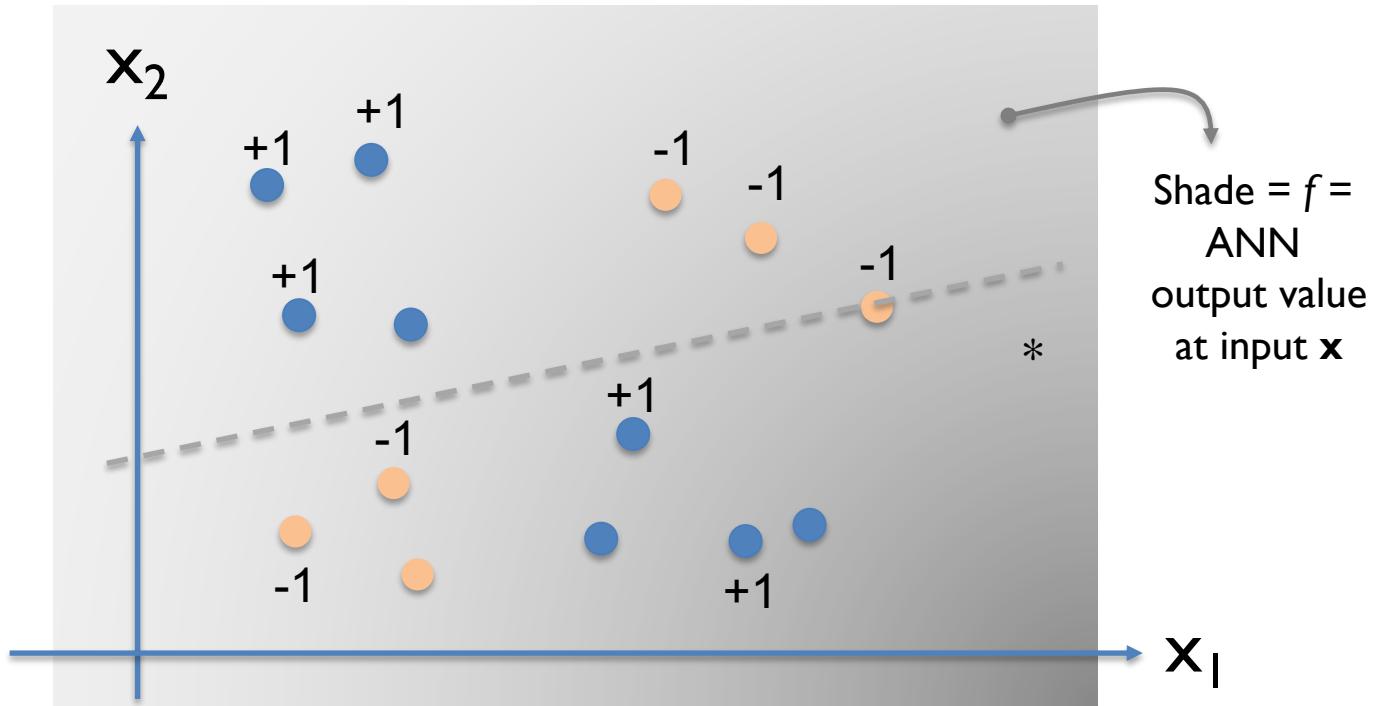
$$L(\mathbf{w}; \underline{\mathbf{x}_i, y_i}) = \sum_{i \in \text{examples}} [f(\mathbf{x}_i, \mathbf{w}) - y_i]^2$$

Labeled examples

ANN output

\* defined by

$\mathbf{w}_{1,2,\dots,n}$



# An optimization problem

$$\min L(\mathbf{w}; \mathbf{x}_i, y_i)$$



- That is, vary the set of weights  $\mathbf{w}$  until a “good enough” solution (=low loss) is found
- How good?

Available examples

Labeled examples to use as a *training set*

Labeled examples to use as a *test set*

# Overfitting

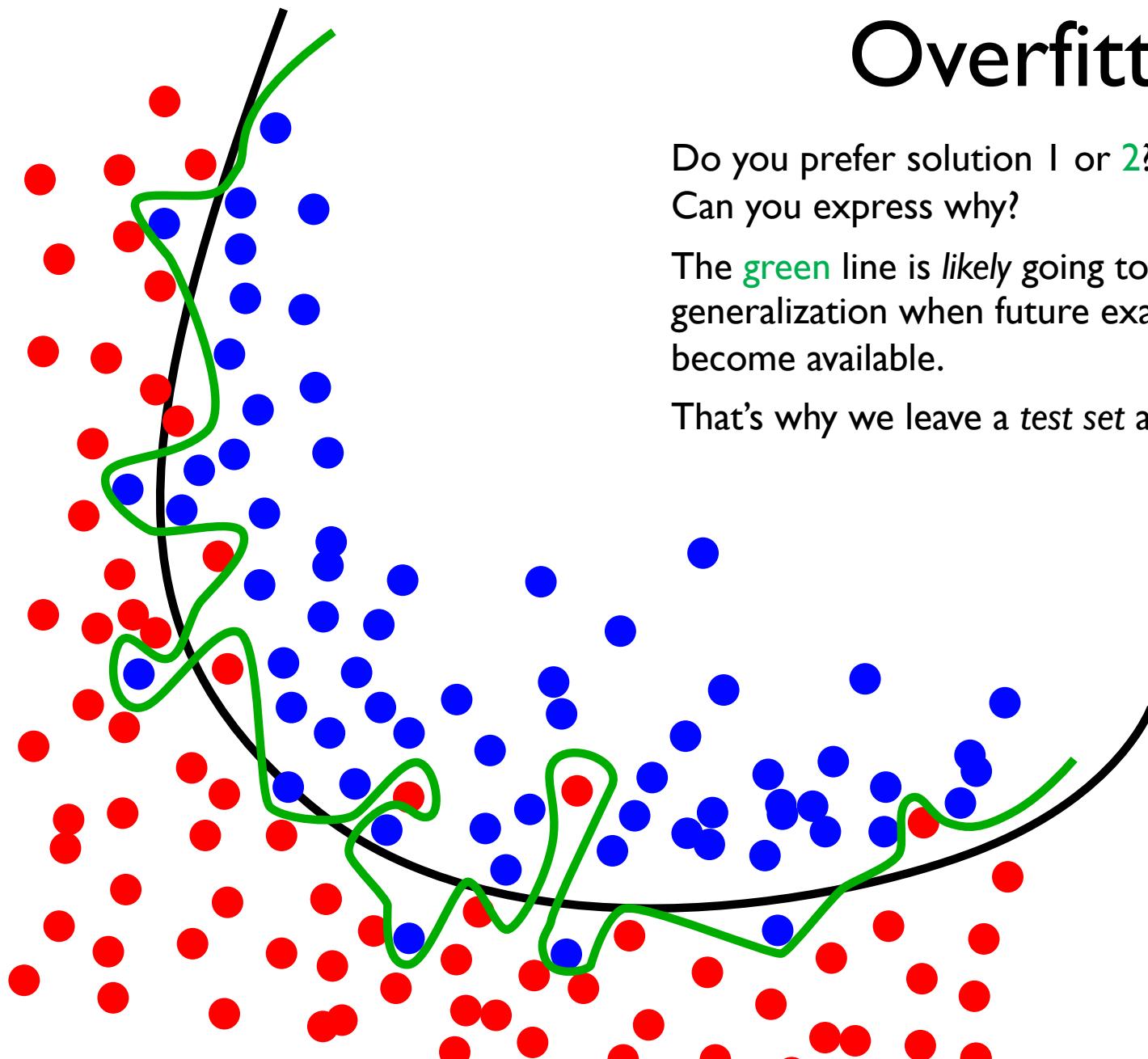
Do you prefer solution 1 or 2?  
Can you express why?

The green line is *likely* going to be a bad generalization when future examples become available.

That's why we leave a test set aside.

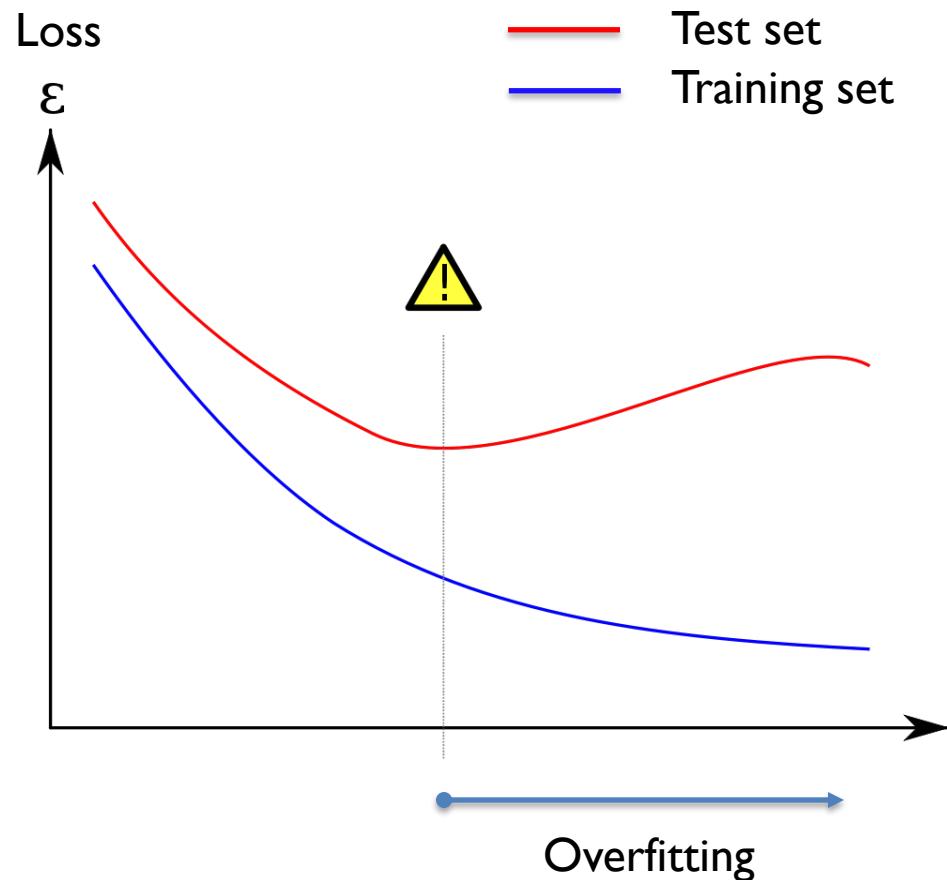
Property 1 (input 1)

Property 2 (input 2)



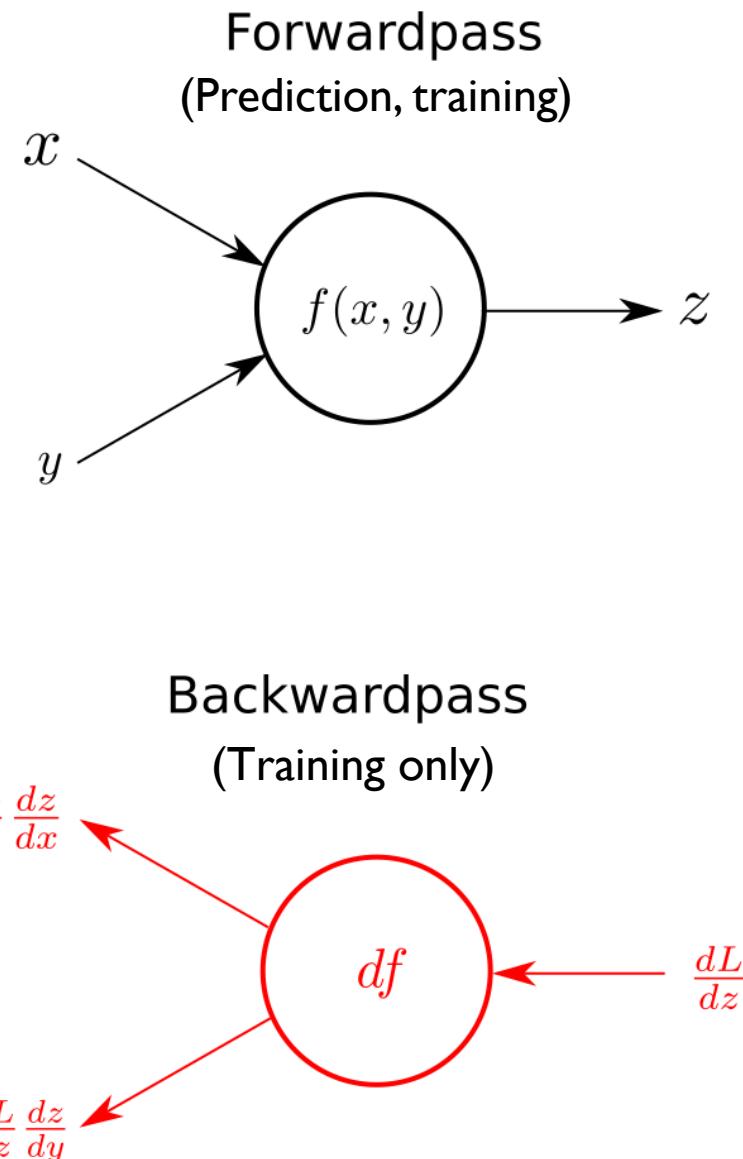
# Training and stopping

- In practice, one computes the loss function on both the training set and the testing set, separately
- Uses the error on the **training set** to optimize parameters
- Training stops when loss on **test set** is minimized



# Backpropagation

- The weights are adjusted in the direction that decreases the error on the test set
- BP is an algorithm determining how each training example would nudge the weights towards the most rapid decrease of the loss function
- Gradient/stochastic gradient descent (SGD) algorithms
- Illustration:  
[www.3blue1brown.com/nn3-thanks](https://www.3blue1brown.com/nn3-thanks)

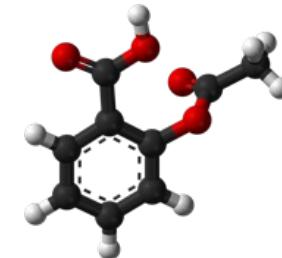


# Interpretability

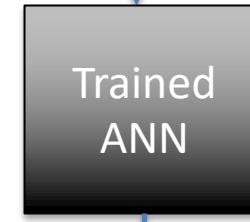


i.e. try to recover the theory/logic behind the rules

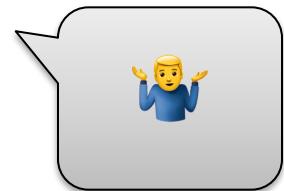
- A simple *regression*, once trained, may hold useful information in its parameters
  - E.g.: predict toxicity from  $K_D$ 's on various targets. What's  $\Delta LD50$  due to  $\Delta K_D$  of each target?
- ANNs (esp. if large) behave as “black boxes”.
  - Will provide (hopefully) useful predictions, but you won't learn easily from them.
- Interpretability of ML models is problematic (with ethical implications too!).



Input ( $x_1, x_2 \dots x_k$ )



But why?



Prediction  $y(x)$

# Featurization

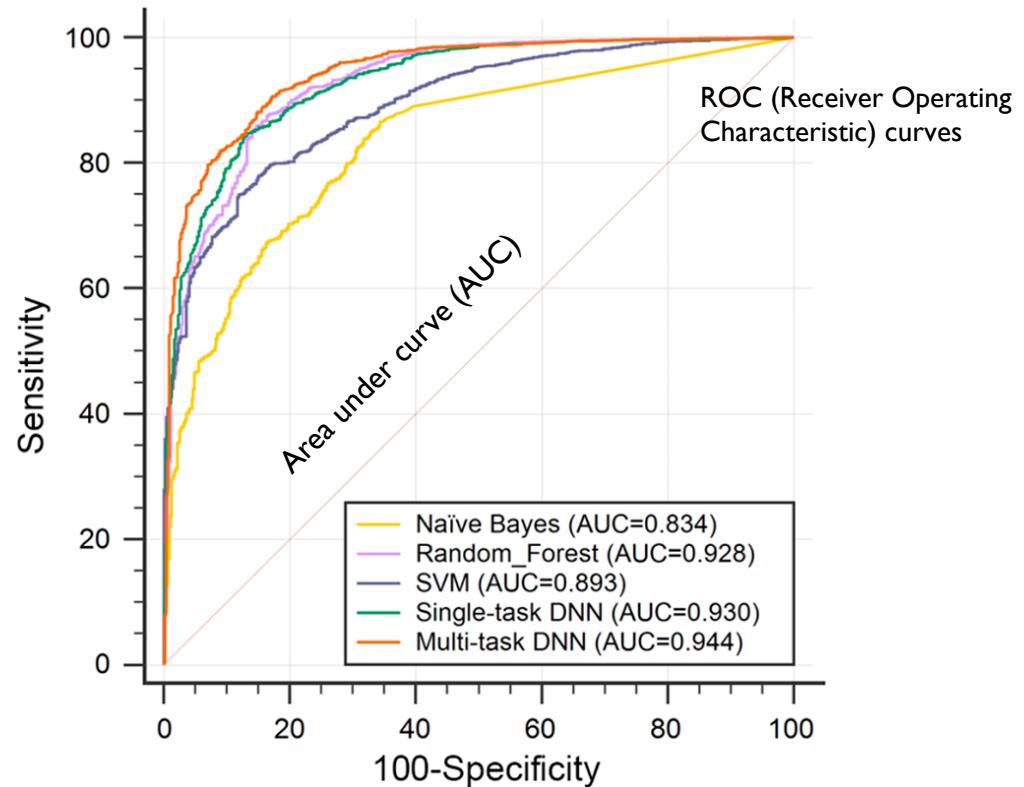
- What if the inputs are not a set of real numbers?
- We need to devise a *featurization* to convert arbitrary objects to (fixed length) vectors
  - Chemical topologies
  - Structures
  - Protein surfaces
  - Images...

*Commonly-used features in proteomics / chemoinformatics:*

- Sequences
- Grids (voxels)
- Molecular descriptors, fingerprints (binary or not)
- SMILES strings
- Graph-based models

# Evaluation of (binary) classifiers

Reality			
		P	N
Prediction	P	TP	FP
	N	FN	TN



$$SE = \frac{TP}{TP + FN}$$

*Sensitivity*: fraction (%) real P's I caught

$$SP = \frac{TN}{TN + FP}$$

*Specificity*: fraction real N's I caught

$$Q+ = \frac{TP}{TP + FP}$$

*Pos. pred. accuracy*: fraction of my P's were correct

$$Q- = \frac{TN}{TN + FN}$$

*Neg. pred. accuracy*: fraction of my N's were correct

$$Q = \frac{TP + TN}{TP + FN + FP + TN}$$

*Overall pred. acc.*: fraction of my preds. were correct

# Software

- The concepts are relatively easy...
- ...implementation (e.g. gradients) is not
- Many software packages
  - PyTorch
  - Keras
  - TensorFlow
  - ...
- Essentially you
  - describe the network topology (in code!)
  - Supply the examples
  - Run training
- Serious training requires GPUs



# **Part II**

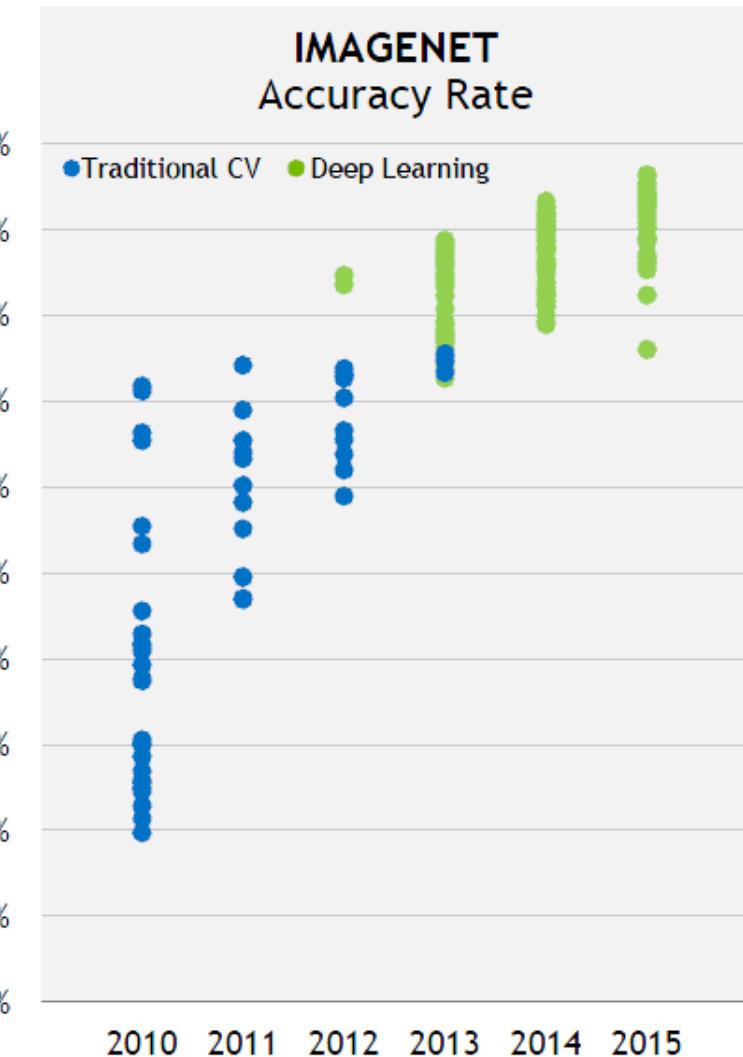
**Advanced ANN architectures**

# Modularity and training

- ANNs with > 2-3 hidden layers are called *deep*.
  - In principle they may learn more “abstract” concepts
  - However, lots of data and computing power is needed.
- Architectures have become very varied...
  - ...but all of them rely on the concept of *finding the weights that minimize the loss function*
  - Let's review some recent “famous” topologies (because they were successful).

# Convolutional NNs (CNNs)

- Featurization is a problem. How to be sure to have the “best” features?
- → *Learn them*
- CNNs work “naturally” with spatially-correlated and translation-invariant problems, e.g. *images*.



# Convolution operation

1 x1	1 x0	1 x1	0	0
0 x0	1 x1	1 x0	1	0
0 x1	0 x0	1 x1	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

Result for filter F1,  
for the given channel.

Will be processed with further convolutions!

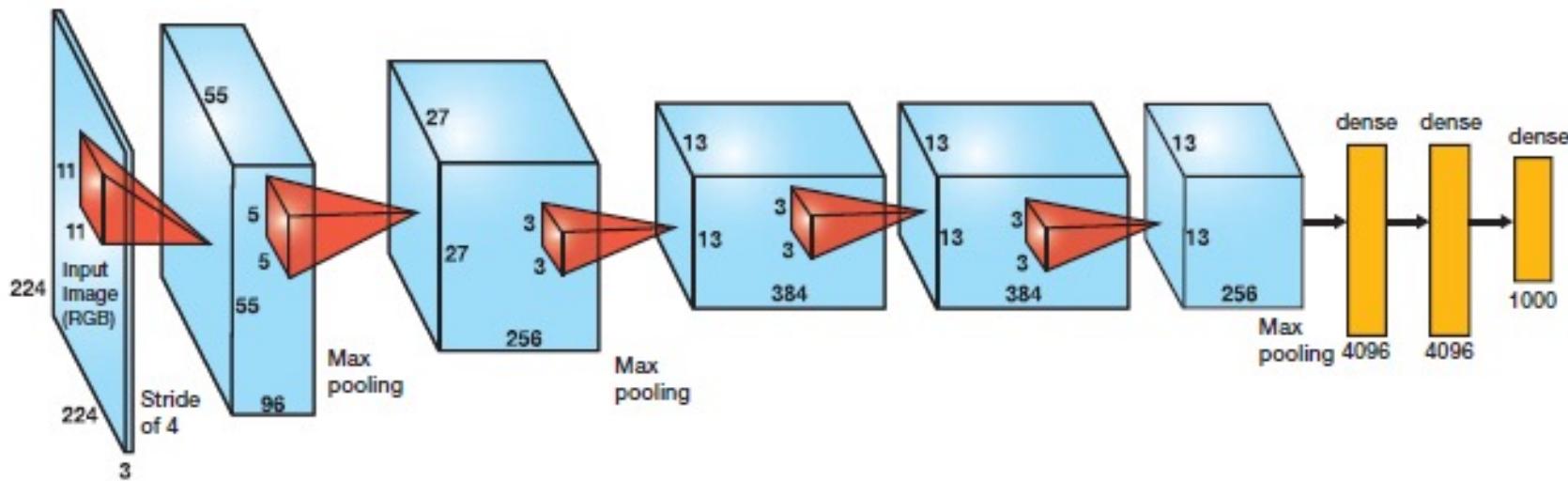
1	0	1
0	1	0
1	0	1

Convolution filter weights  
(will be trained)

- Convolution weights are learned during the training process
- The most “useful” preprocessings for the given task (see aside) will emerge from the training
- Automatic feature selection

	Operation	Filter	Convolved Image
Identity		$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection		$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Sharpen		$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
Box blur (normalized)		$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)		$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

# Example



AlexNet, 2012 ImageNet classification winner.

[PDF] [Imagenet classification with deep convolutional neural networks](#)

A Krizhevsky, I Sutskever... - Advances in neural ..., 2012 - proceedings.neurips.cc

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is ...  
☆ ⚡ Citato da 83853 Articoli correlati Importa in BibTeX ☰

Hinton et al., NIPS 2012 paper #4824



## LeNet, 1998

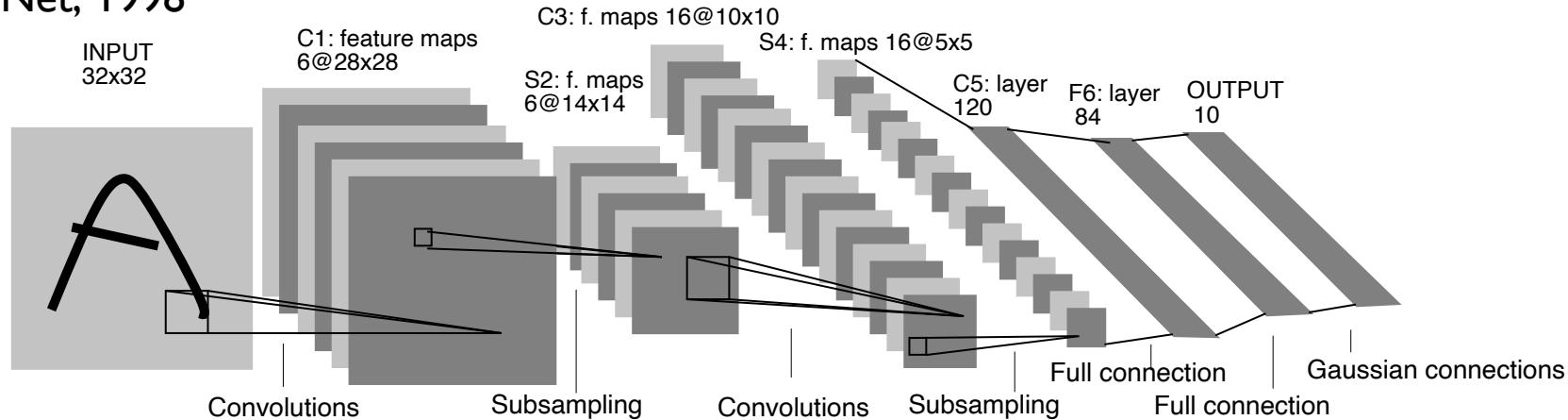
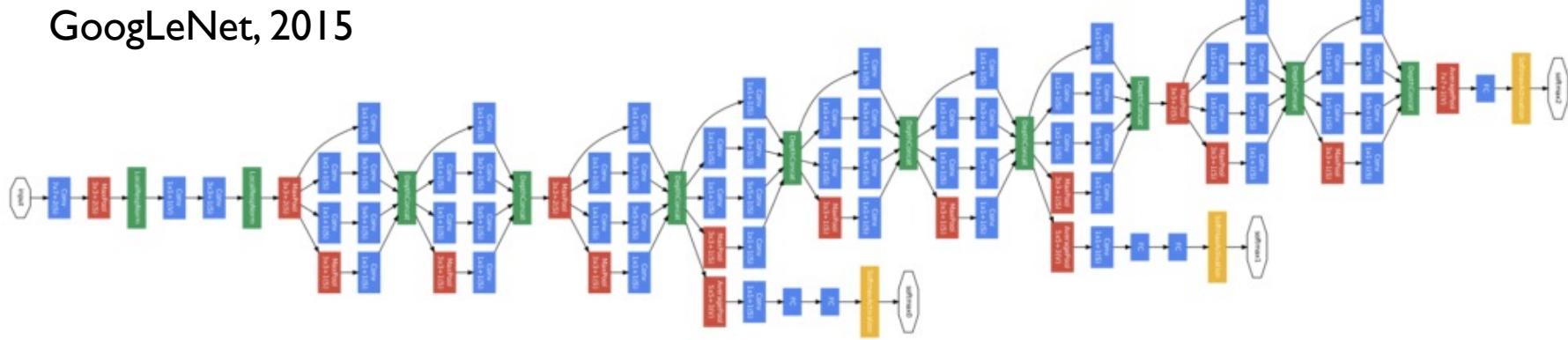


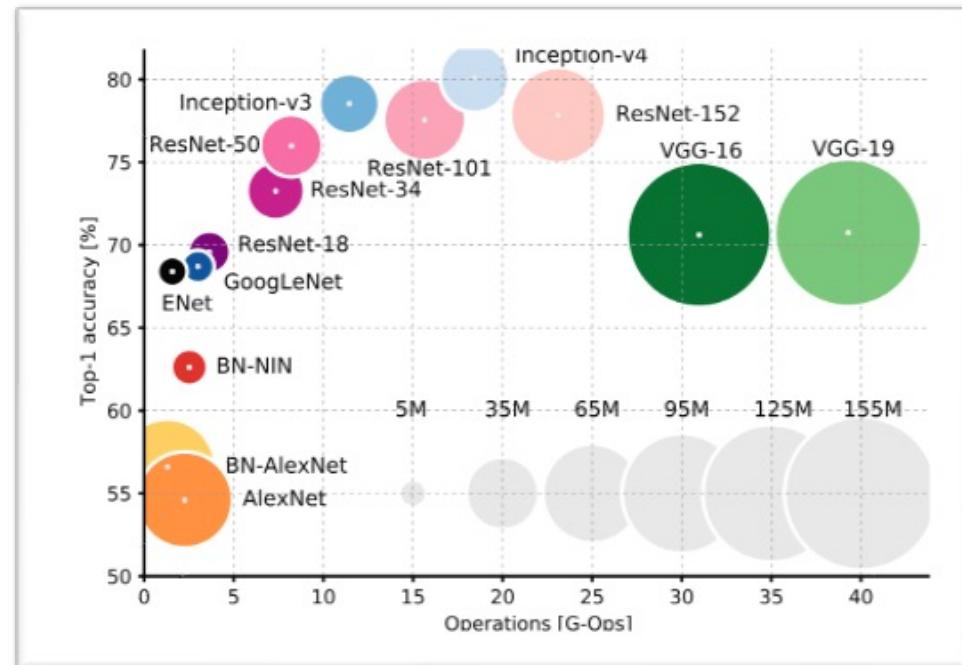
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

## GoogLeNet, 2015



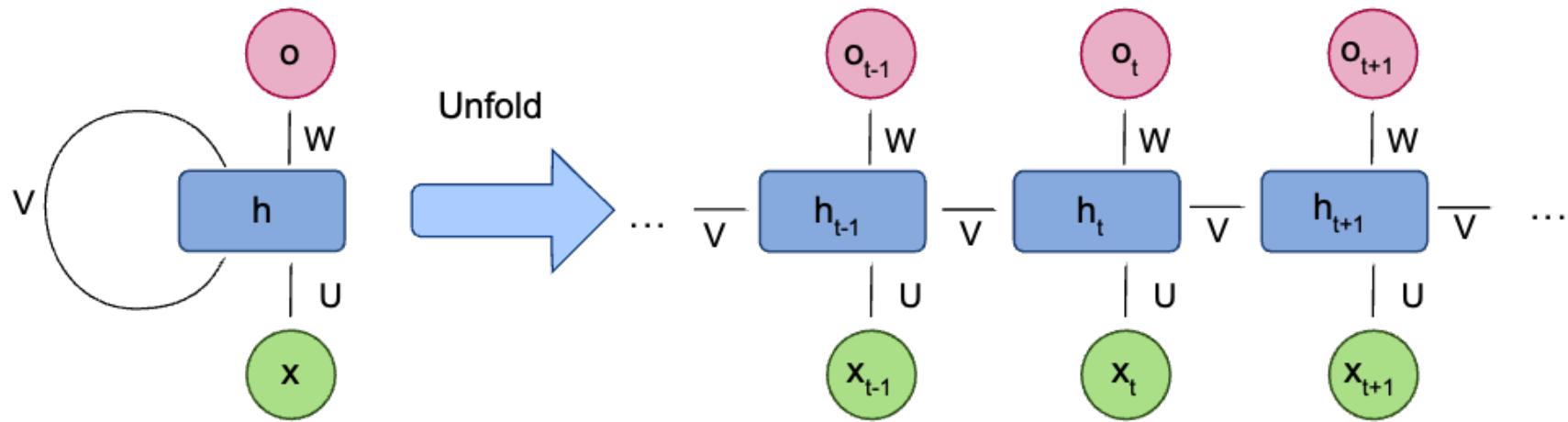
# How complex can things become?

Year	CNN	Developed By	Error rates	No. of parameters
1998	LeNet	Yann LeCun et al		60 thousand
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	15.3%	60 million
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8%	
2014	GoogLeNet	Google	6.67%	4 million
2014	VGGNet	Simonyan, Zisserman	7.3%	138 million
2015	ResNet	Kaiming He	3.6%	



# Recurrent NNs

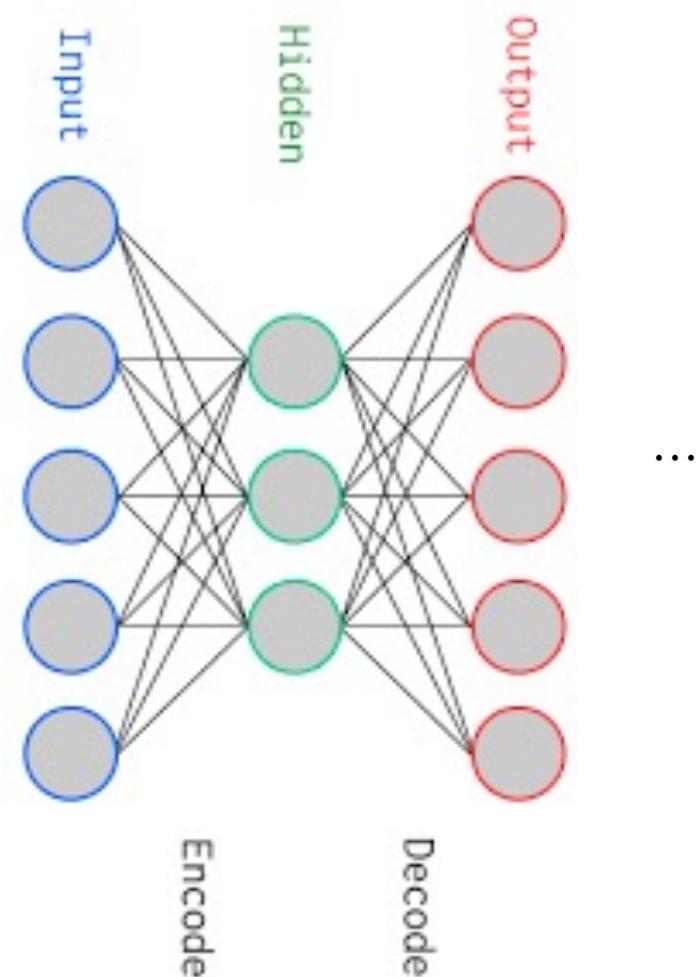
- To address *time-varying* data, or sequences, we may feed the output of a neuron to itself in the next step.
- Inputs and outputs are sequences.
- Good for translation-like tasks.



# Autoencoders

- “Force” the network to cut down information by creating a “funnel”
- Input is the same as output (loss is reproduction error)
- A dimensionality reduction technique

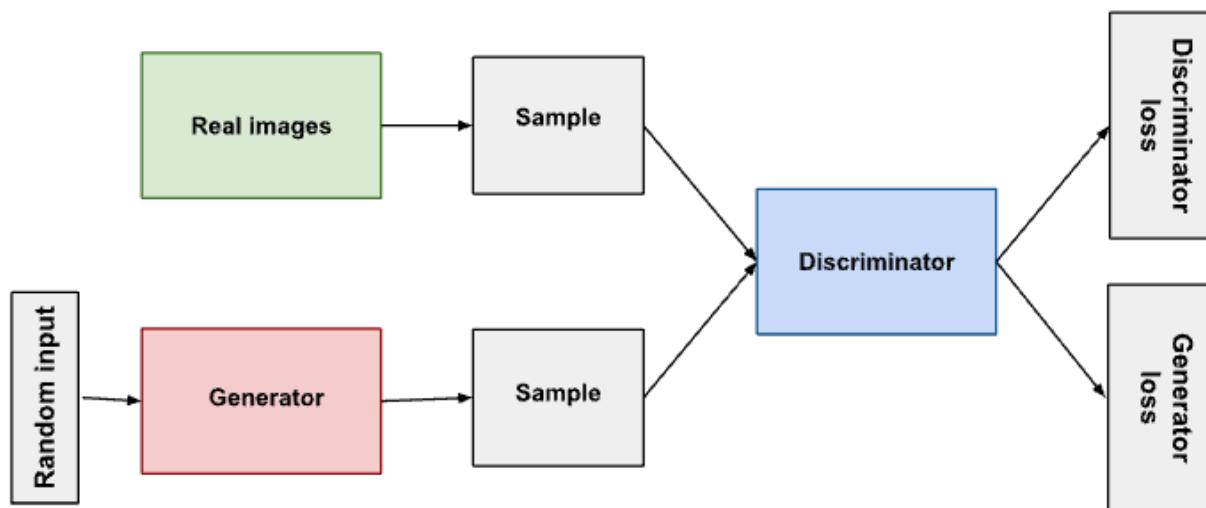
“Bottleneck”



# Generative Adversarial Networks

Goodfellow et al., 2014    16,000 citations

- Train two networks
  - the *generator* produces “fake” cases *from noise*
  - the *discriminator* must learn to tell fakes from actual examples
- Train simultaneously until convergence
- Used for “inventing” realistic molecules with given properties



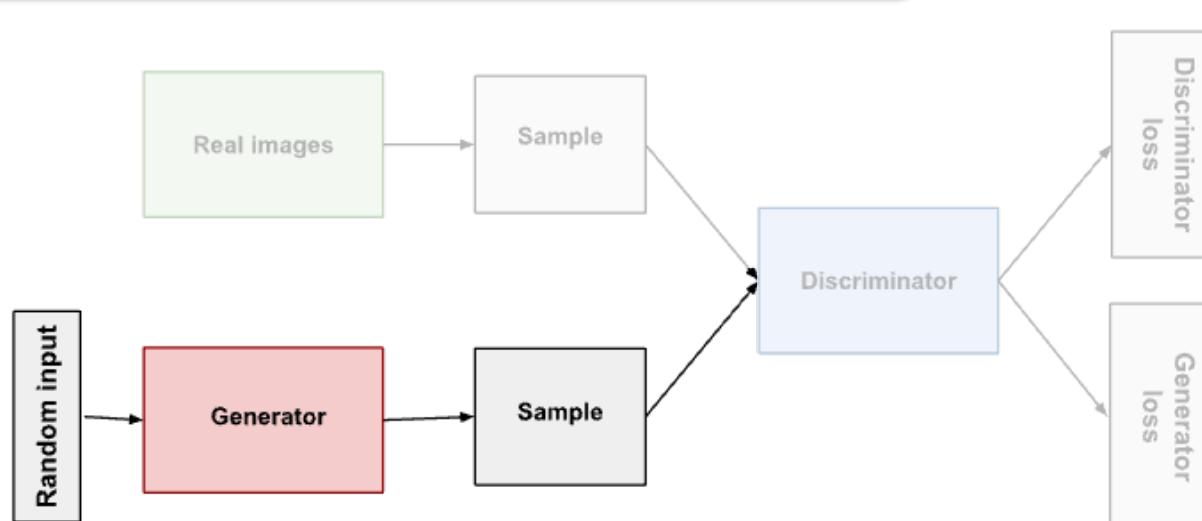
# Generative Adversarial Networks

- Example: StyleGAN2
  - [www.thispersondoesnotexist.com](http://www.thispersondoesnotexist.com)
  - [www.whichfaceisreal.com](http://www.whichfaceisreal.com)

Idea (one of the many possible):  
realistic faces → molecules with given properties



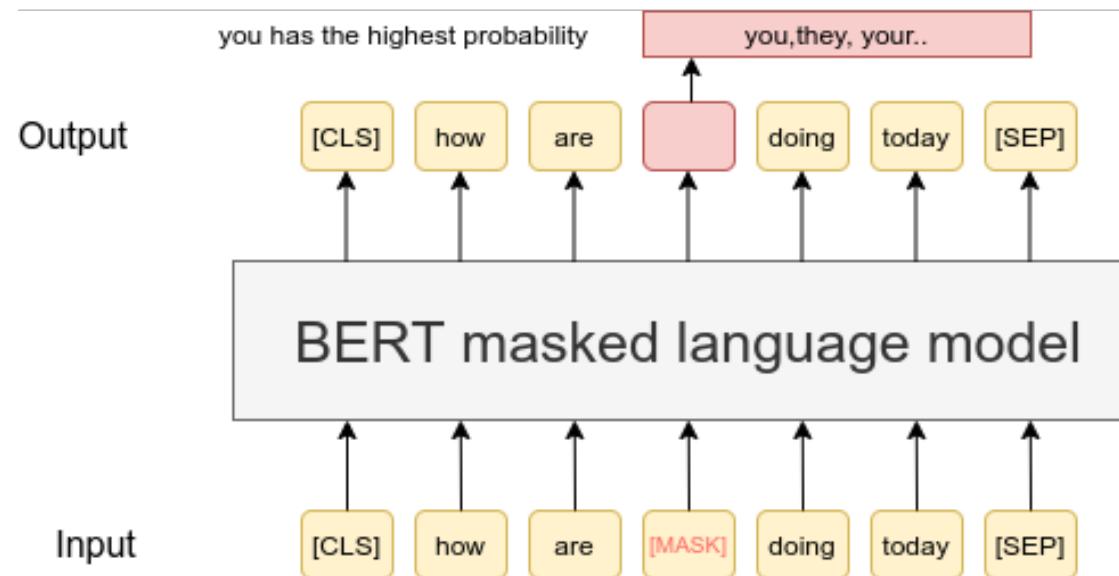
Sampled from generator.



# **Transformers and self-attention**

# Bidirectional Encoder Representations from Transformers (BERT)

- In 2018 broke records in *many* natural-language tasks
  - Generation, understanding, summarization, translation...
  - Try @ [huggingface.co/bert-base-multilingual-cased](https://huggingface.co/bert-base-multilingual-cased)



- Google's - see <https://arxiv.org/abs/1810.04805v2>
- <https://jalammar.github.io/illustrated-bert/>

# Attention heads

- Given enough examples, the network learns sequence and position-dependent features
- E.g. will learn to associate pronouns with the referenced object (~ language structure)
- “Attention mechanism”

The defining characteristic of the Transformer is its **attention** mechanism, which we'll illustrate with an example from natural language. Suppose we train a Transformer to predict the next word in a sentence – a task known as **language modeling**:

| *The girl kicked the ball into the woods. It \_\_\_\_\_*

For the model to guess a reasonable next word, e.g. “*rolled*”, it must understand that “it” refers to “*ball*” (a relationship known as **coreference**). The Transformer does this by looking back at, or *attending to*, the earlier mention of “*ball*” when processing “it”:

| *The girl kicked the ball into the woods. It \_\_\_\_\_*



# **Transformer-based models for protein sequences**

(a)

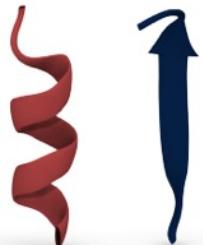
PRIMARY

MKLPIYTRF  
GVNCWSWQA  
AIPLMN...

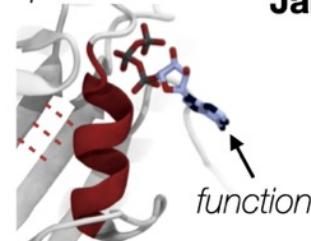
*amino acids*

ABCDEFGHIJKLMNO  
PQRSTUVWXYZ

*alphabet*



*protein*



Jane  
bicycle  
ride

Jane rides her bicycle



*meaning*

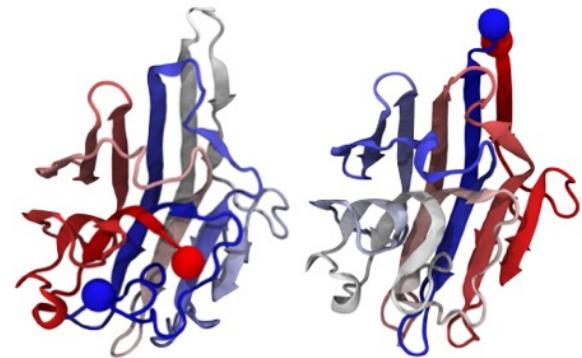
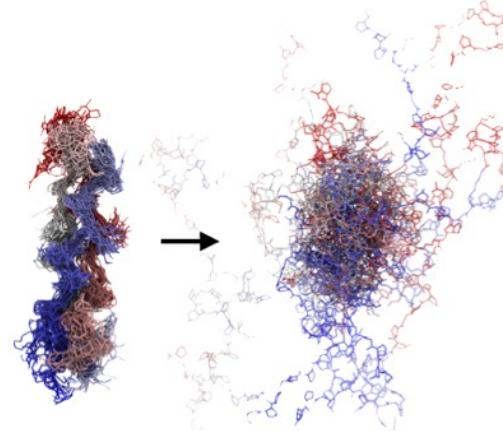
SECONDARY

TERTIARY

QUATERNARY

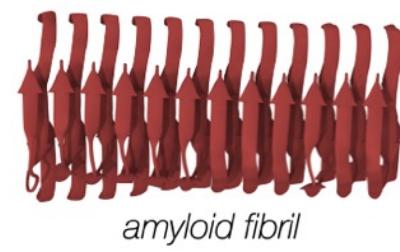
(b)

Your  
manuscript is  
now **t**  
publishable



I am happy  
when it rains.  
When it rains, I  
am happy.

Colourless  
green ideas  
sleep furiously.



*amyloid fibril*

# Transformer-based sequence analysis

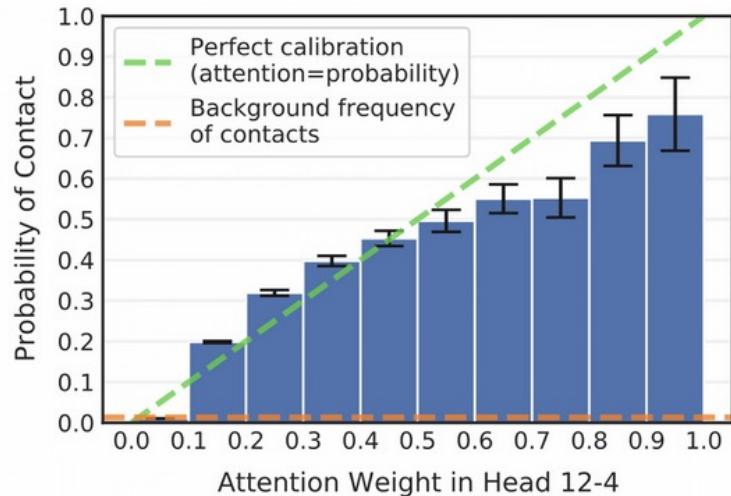
- **ProteinBERT**
- **BERTology** (Salesforce Research)
- UDSMProt
- ESM\* (Facebook)
- UniRep
- TAPE
- ProtTrans (ProtBERT)

\* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences

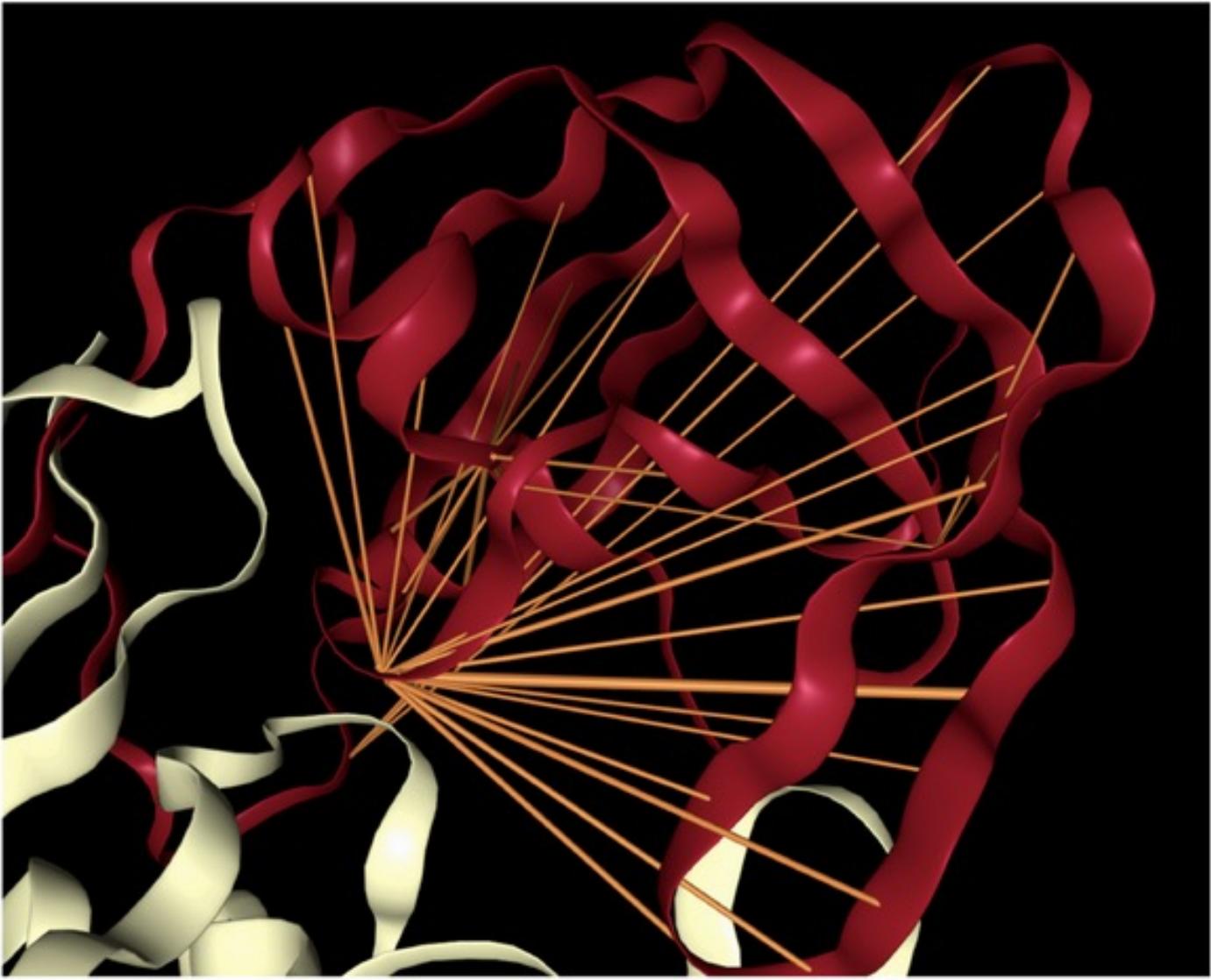
# BERTology



Head 12-4 has learned to focus attention (indicated by orange lines) between amino acids that are spatially close in the folded protein structure (see inset subsequence 117D-157I) but lie apart in the sequence, based solely on language model pre-training. The example is a *de novo* designed TIM-barrel. 76% of high-confidence ( $>0.9$ ) attention from this head aligns with ground-truth contact maps on average over a dataset.



Probability that two amino acids are in contact [95% confidence intervals], as a function of attention between the amino acids in Head 12-4, showing attention approximates a perfectly-calibrated estimator



Head 7-1 has learned to focus attention (indicated by orange lines) on binding sites, a key functional component of proteins. Example is HIV-1 protease (7HVP). The primary location receiving attention is 27G, a binding site for protease inhibitor small-molecule drugs. 44% of high-confidence ( $>0.9$ ) attention from this head focuses on binding sites on average over a dataset.

# ProteinBERT: A universal deep-learning model of protein sequence and function

doi:10.1101/2021.05.24.  
445464

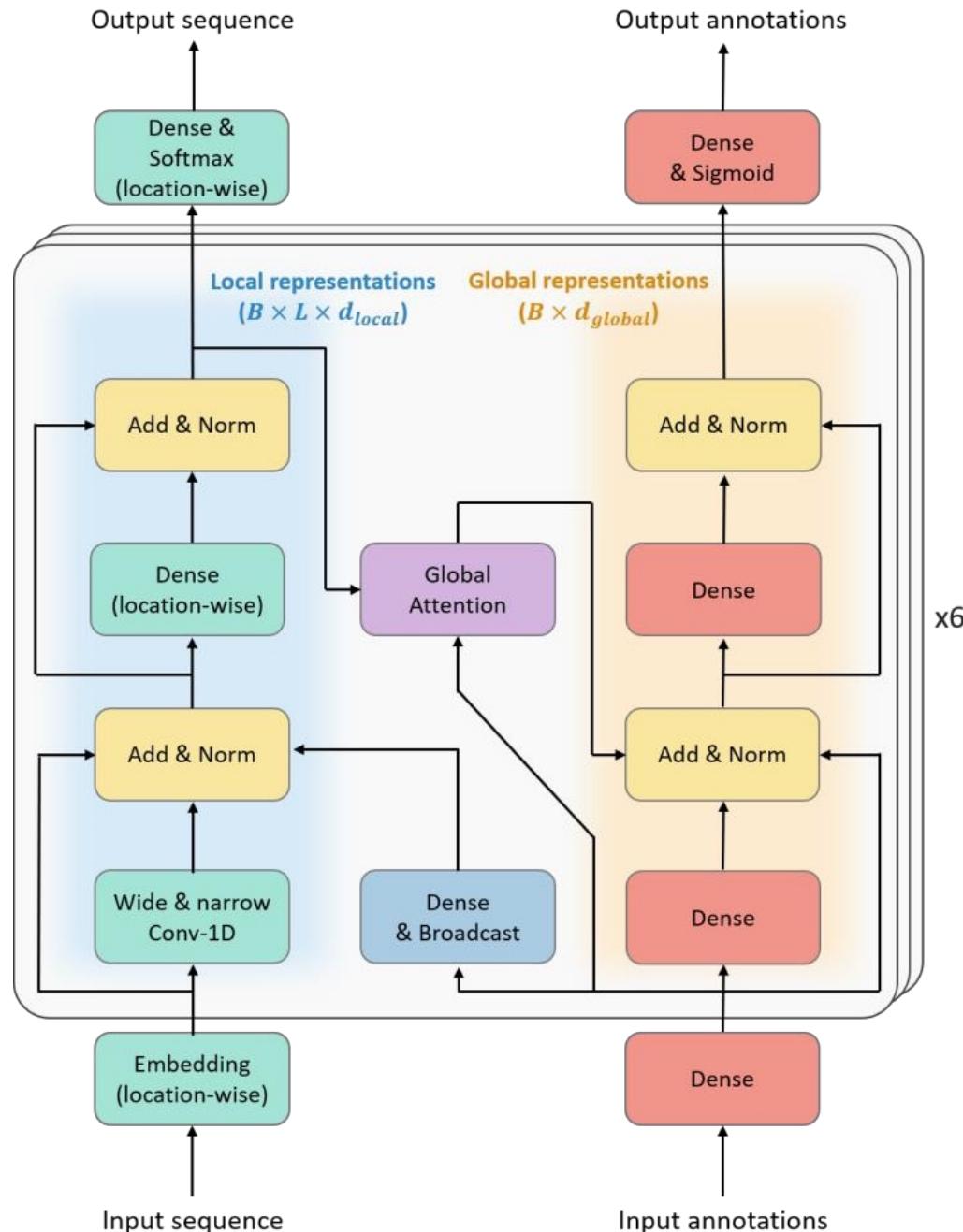
[https://github.com/nadavbra/protein\\_bert](https://github.com/nadavbra/protein_bert)

Self-supervised deep language modeling has shown unprecedented success across natural language tasks, and has recently been repurposed to biological sequences. However, existing models and pretraining methods are designed and optimized for text analysis. We introduce ProteinBERT, a deep language model specifically designed for proteins. Our pretraining scheme consists of masked language modeling combined with a novel task of Gene Ontology (GO) annotation prediction. We introduce novel architectural elements that make the model highly efficient and flexible to very large sequence lengths. The architecture of ProteinBERT consists of both local and global representations, allowing end-to-end processing of these types of inputs and outputs. ProteinBERT obtains state-of-the-art performance on multiple benchmarks covering diverse protein properties (including protein structure, post translational modifications and biophysical attributes), despite using a far smaller model than competing deep-learning methods. Overall, ProteinBERT provides an efficient framework for rapidly training protein predictors, even with limited labeled data. Code and pretrained model weights are available at [https://github.com/nadavbra/protein\\_bert](https://github.com/nadavbra/protein_bert).

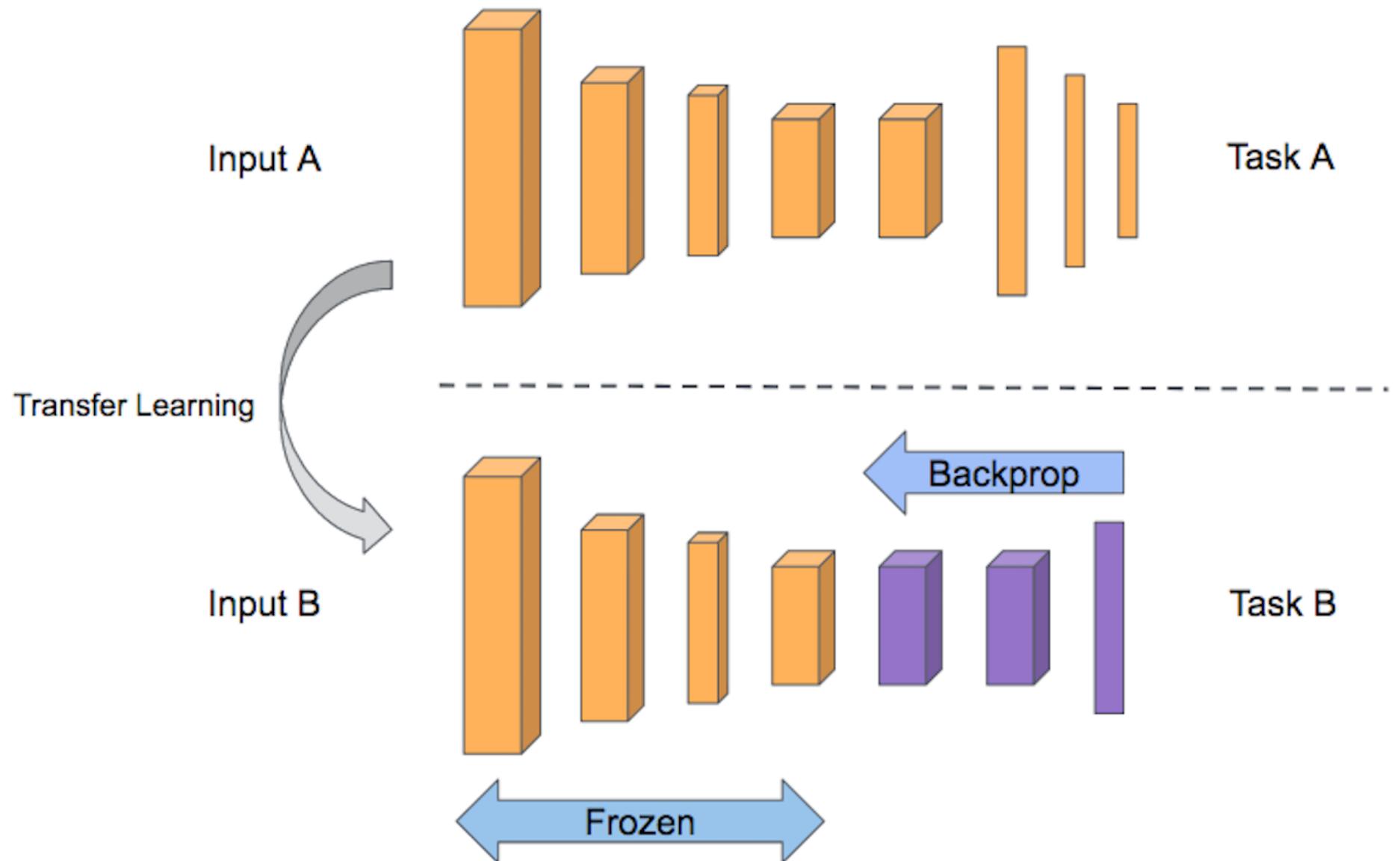
# “Exemplary” as it provides:

- Paper
- Code
- Model
- Weights
- Data!

Easy to reproduce and repurpose



# Transfer learning



Topic	Benchmark	Target type <sup>a</sup>	Resolution	# Training sequences	Source
Protein structure	Secondary structure	Categorical (3)	Local	8,678	[21, 30]
	Disorder	Binary	Local	8,678	[30]
	Remote homology	Categorical (1,195)	Global	12,312	[21, 31, 32]
	Fold classes	Categorical (7)	Global	15,680	[31, 32]
Post-translational modifications	Signal peptide	Binary	Global	16,606	[33]
Biophysical properties	Major PTMs	Binary	Local	43,356	[34]
	Neuropeptide cleavage	Binary	Local	2,727	[35–37]
Biophysical properties	Fluorescence	Continuous	Global	21,446	[21, 38]
	Stability	Continuous	Global	53,679	[21, 39]

<sup>a</sup>For categorical targets, the number of classes appears in parentheses.

# Sequence examples

ftp://ftp.cs.huji.ac.il/users/nadavb/protein\_bert/  
protein\_benchmarks/\*

- Global (signal peptide)

label, seq

0, MLGMIRNSLFGSVETWPWQVLSTGGKEDVSYEERACEGGKFATVEVTDPVDEALREAMPKIMKYVGGTN  
1, MQPAKNLLFSSLLFSSLLFSSAARAASEDGGRGPYVQADLAYAAERITHDYPKPTGTGKNKISTVSDYFR  
0, MDKGEGLRLAATLRQWTRLYGGCHLLLGAVVCSLLAACSSSPGGVKVDRNGSAPAAARRTPVTSGQYI  
...

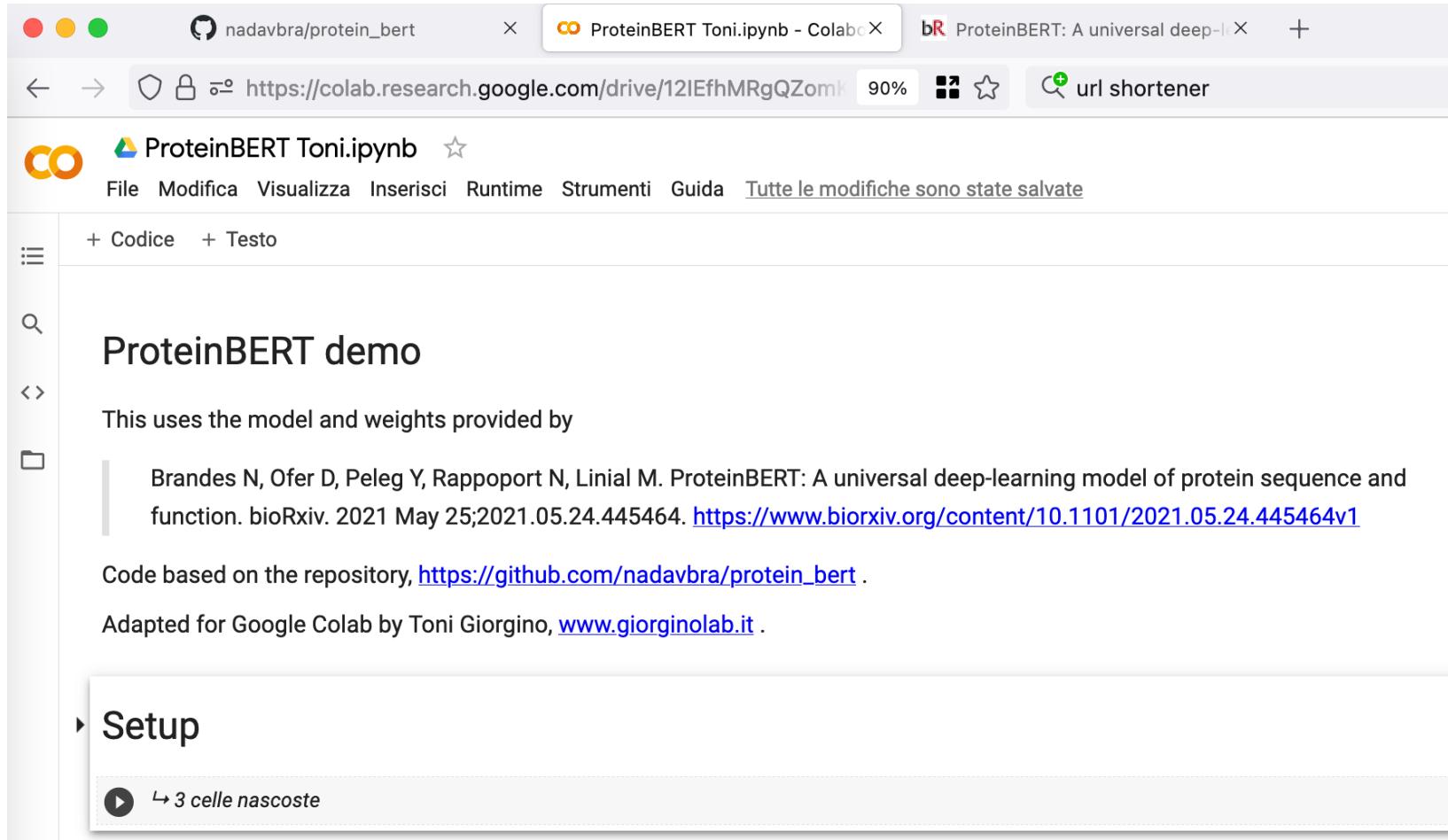
- Local (secondary structure)

seq, label

MNDKRLQFNETLSKLILGGLRLRGISNSITDYQKLYKITFDAEFTHRDELKRISMGSGEEVSFESLQETVETLL  
KLFTKSLEHHHHHH, 222220000000000000000022222220000000000000002000000002222  
22220000000000000002222222222

...

[github.com/giorginolab/protein\\_bert](https://github.com/giorginolab/protein_bert)  
<https://tinyurl.com/4xfswksu>



The screenshot shows a Google Colab notebook titled "ProteinBERT Toni.ipynb". The notebook interface includes a toolbar with file operations like File, Modifica, Visualizza, Inserisci, Runtime, Strumenti, Guida, and a message indicating "Tutte le modifiche sono state salvate" (All changes have been saved). Below the toolbar, there are two tabs: "+ Codice" and "+ Testo". The main content area is titled "ProteinBERT demo" and contains the following text:

This uses the model and weights provided by

Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M. ProteinBERT: A universal deep-learning model of protein sequence and function. bioRxiv. 2021 May 25;2021.05.24.445464. <https://www.biorxiv.org/content/10.1101/2021.05.24.445464v1>

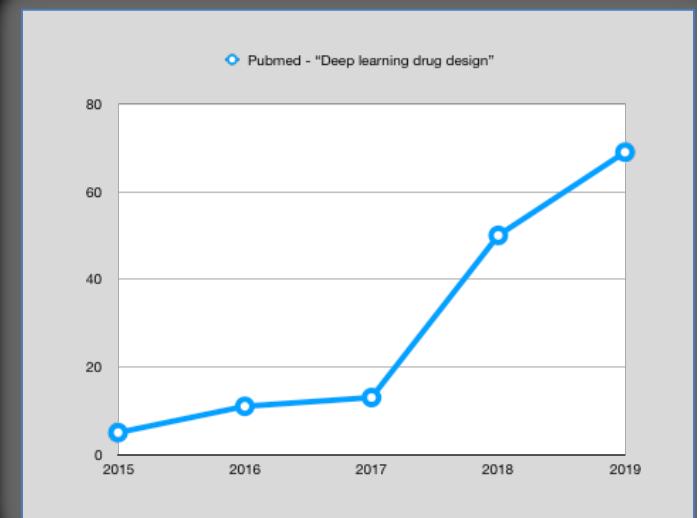
Code based on the repository, [https://github.com/nadavbra/protein\\_bert](https://github.com/nadavbra/protein_bert).

Adapted for Google Colab by Toni Giorgino, [www.giorginolab.it](http://www.giorginolab.it).

A section titled "Setup" is expanded, indicated by a right-pointing arrow icon. A button at the bottom left of this section says "▶ 3 celle nascoste" (▶ 3 hidden cells).

# Part III

## Applications from the literature

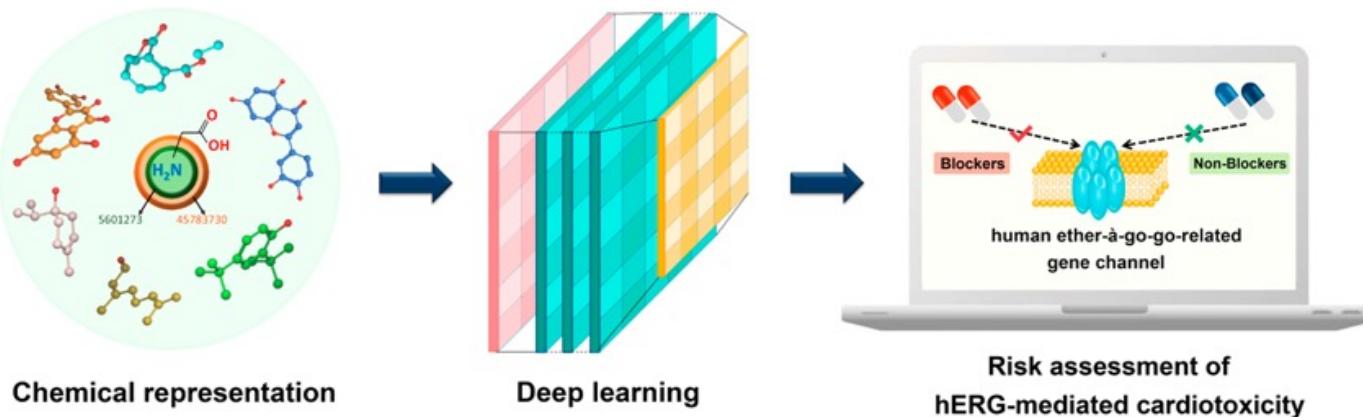
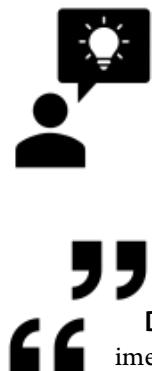


# Motivation for DNN in DD

- Can handle thousands of descriptors without selection
  - Promise of increased classification accuracy
    - (Hope to reproduce the *exploit* achieved in image recognition)
  - Will automatically trained features surpass “hand-crafted” ones?
  - Tackle open problems such as precise  $K_D$  prediction
- Virtual screening
  - Affinity prediction
  - Affinity ranking
  - Pose generation
  - Pocket finding
  - *De novo* generation
  - Reaction plans

## Deep Learning-Based Prediction of Drug-Induced Cardiotoxicity

Chuipu Cai,<sup>†,‡,§,¶</sup> Pengfei Guo,<sup>†</sup> Yadi Zhou,<sup>§</sup> Jingwei Zhou,<sup>†</sup> Qi Wang,<sup>†</sup> Fengxue Zhang,<sup>‡</sup> Jiansong Fang,<sup>\*,†,||,‡,¶</sup> and Feixiong Cheng<sup>\*,†,||,‡,¶</sup>



**Data Preparation.** The original compounds with experimental hERG blockage bioactivities were assembled from various well-defined experimental assays: (i) patch-clamp measurements from ChEMBL bioactivity database; (ii) radioligand binding measurements on mammalian and non-mammalian cell lines; (iii) hERG K<sup>+</sup> channel binding affinity, and (iv) literature-derived data (Supporting Information, Table S1).<sup>25,28,29,34</sup> We then implemented three criteria: (i) compounds without well-defined experimental hERG blocking bioactivities were eliminated; (ii) incompatible measuring units were converted to unified IC<sub>50</sub> value ( $\mu\text{M}$ ); (iii) only compounds with IC<sub>50</sub> value  $\leq 10 \mu\text{M}$  were considered as hERG blockers, while the rest were regarded as “decoy pool” for further screening according to different threshold settings.

validating single 3D conformers. Finally, a comprehensive collection consisted of 7,889 compounds with well-defined experimental hERG blocking bioactivities was obtained, and 4,355 of the compounds whose experimental values were less than or equal to 10  $\mu\text{M}$  were regarded as hERG blockers (Supporting Information, Table S2 and Table S3).

Subsequently, all compounds were split into three sets—training set, test set, and validation set—with ratios of 8:1:1 via chemical diversity analysis performed by Tanimoto Coefficient measure based on MACCS fingerprint in MOE 2010. Such a split would assign the chemical structures uniformly and avoid potential data bias. Following standard practice, the training

## Deep Learning-Based Prediction of Drug-Induced Cardiotoxicity

Chuipu Cai,<sup>†,‡,§,¶</sup> Pengfei Guo,<sup>†</sup> Yadi Zhou,<sup>§</sup> Jingwei Zhou,<sup>†</sup> Qi Wang,<sup>†</sup> Fengxue Zhang,<sup>‡</sup>  
Jiansong Fang,<sup>\*,†,§</sup> and Feixiong Cheng<sup>\*,||,†,‡,¶</sup>

### Descriptors:

- MOE (x 185)
- Mol2vec (x 100, “embeddings”)

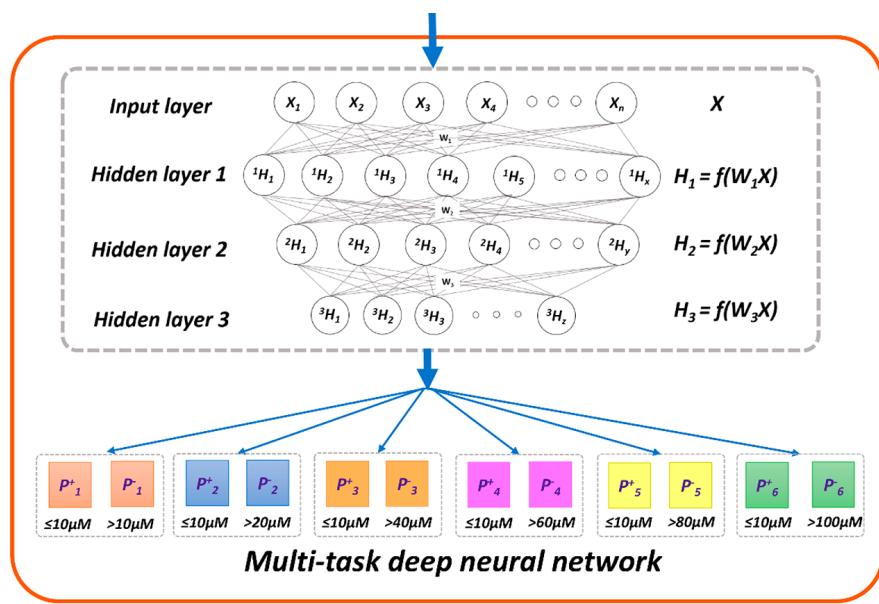


Table 1. Detailed Description of All Data Sets Used in This Stud

threshold value of decoy ( $\mu M$ )	training set		
	positive	negative	total
10	3,485	2,826	6,311
20	3,485	1,755	5,240
40	3,485	863	4,348
60	3,485	644	4,129
80	3,485	469	3,954
100	3,485	380	3,865

# DeepSite: protein-binding site predictor using 3D-convolutional neural networks

J. Jiménez<sup>1</sup>, S. Doerr<sup>1</sup>, G. Martínez-Rosell<sup>1</sup>, A. S. Rose<sup>2</sup> and  
G. De Fabritiis<sup>1,3,\*</sup>

<sup>1</sup>Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), 08003 Barcelona, Spain, <sup>2</sup>San Diego Supercomputer Center, UC San Diego, MC 0505, 9500 Gilman Drive, La Jolla, CA 92093-0505, USA and <sup>3</sup>ICREA, 08010 Barcelona, Spain

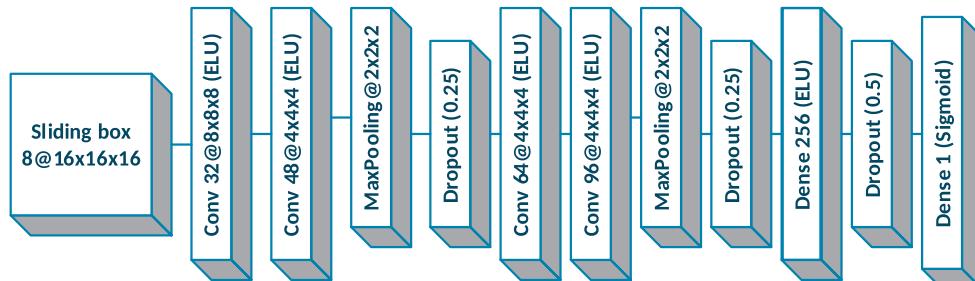
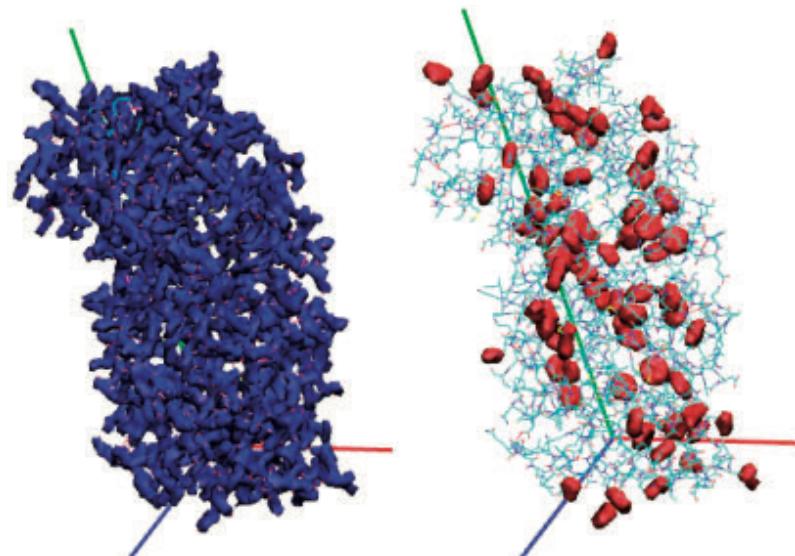
\*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on January 10, 2017; revised on May 9, 2017; editorial decision on May 29, 2017; accepted on May 30, 2017

**Table 2.** Property-atom type (AutoDock 4) correspondence used for Deepsite's 3D descriptor computation

Property	Rule
Hydrophobic	atom type C or A
Aromatic	atom type A
Hydrogen bond acceptor	atom type NA or NS or OA or OS or SA
Hydrogen bond donor	atom type HD or HS with O or N partner
Positive ionizable	atom with positive charge
Negative ionizable	atom with negative charge
Metal	atom type MG or ZN or MN or CA or FE
Excluded volume	all atom types



[www.playmolecule.org](http://www.playmolecule.org)

Jimenez et al., Bioinformatics 2017



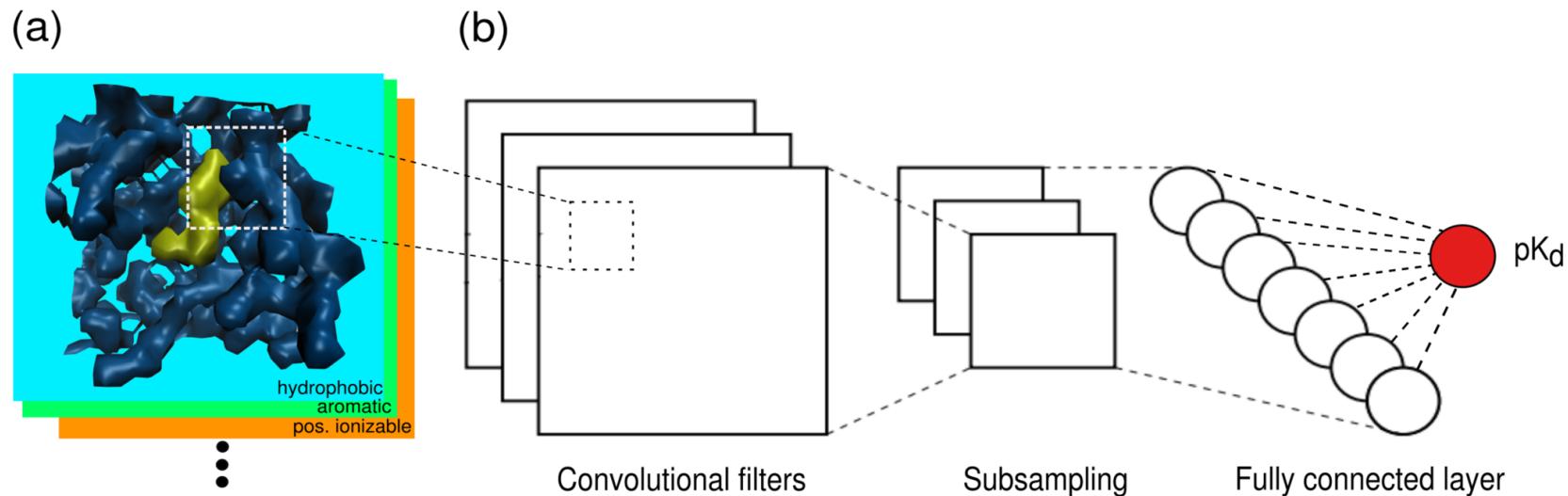
# $K_{\text{DEEP}}$ : Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks

José Jiménez,<sup>†</sup> Miha Škalič,<sup>†</sup> Gerard Martínez-Rosell,<sup>†</sup> and Gianni De Fabritiis<sup>\*,†,‡</sup>

<sup>†</sup>Computational Biophysics Laboratory, Universitat Pompeu Fabra, Parc de Recerca Biomèdica de Barcelona, Carrer del Dr. Aiguader 88, Barcelona 08003, Spain

<sup>‡</sup>Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluís Companys 23, 08010 Barcelona, Spain

Supporting Information



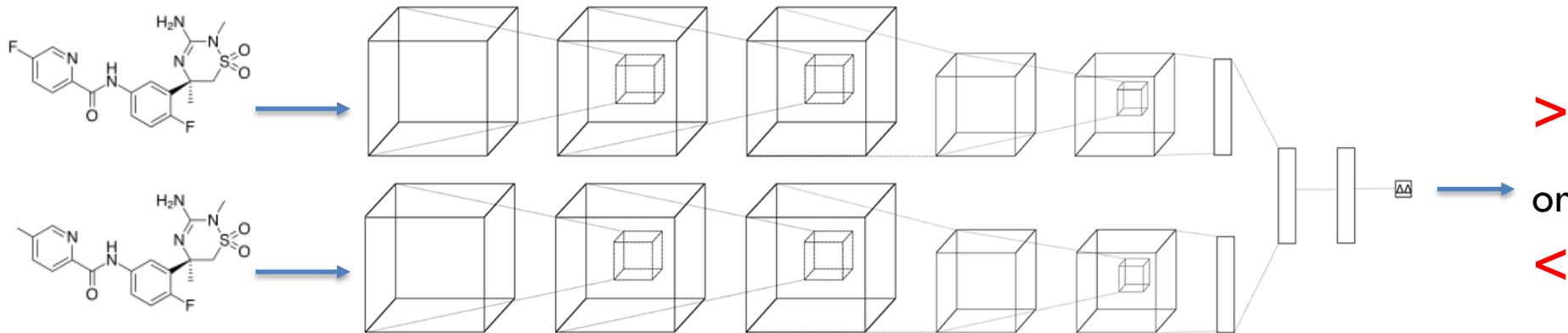


Cite this: *Chem. Sci.*, 2019, 10, 10911

All publication charges for this article have been paid for by the Royal Society of Chemistry

# DeltaDelta neural networks for lead optimization of small molecule potency†

José Jiménez-Luna, <sup>a</sup> Laura Pérez-Benito, <sup>bc</sup> Gerard Martínez-Rosell, <sup>f</sup>  
Simone Scialbola, <sup>d</sup> Rubben Torella, <sup>e</sup> Gary Tresadern <sup>c</sup> and Gianni De Fabritiis \*<sup>afg</sup>



**Fig. 1** Architecture of the proposed model. A two-legged neural network with tied weights was constructed, and a pair of protein-ligand voxelization is feed-forwarded through it to later perform a latent space difference.

# Peptide generation via RNN



JOURNAL OF  
CHEMICAL INFORMATION  
AND MODELING

Article

Cite This: *J. Chem. Inf. Model.* 2018, 58, 472–479

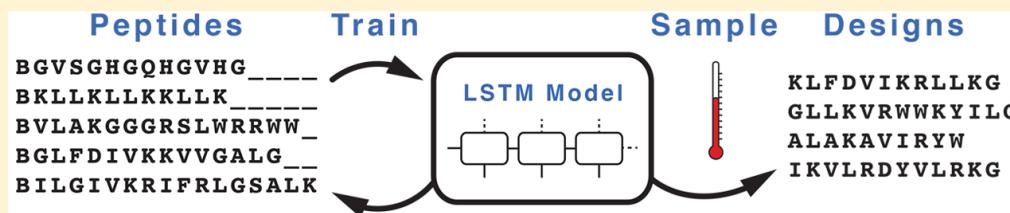
[pubs.acs.org/jcim](https://pubs.acs.org/jcim)

## Recurrent Neural Network Model for Constructive Peptide Design

Alex T. Müller, Jan A. Hiss, and Gisbert Schneider\*<sup>ID</sup>

Swiss Federal Institute of Technology (ETH), Department of Chemistry and Applied Biosciences, Vladimir-Prelog-Weg 4, CH-8093 Zurich, Switzerland

### Supporting Information



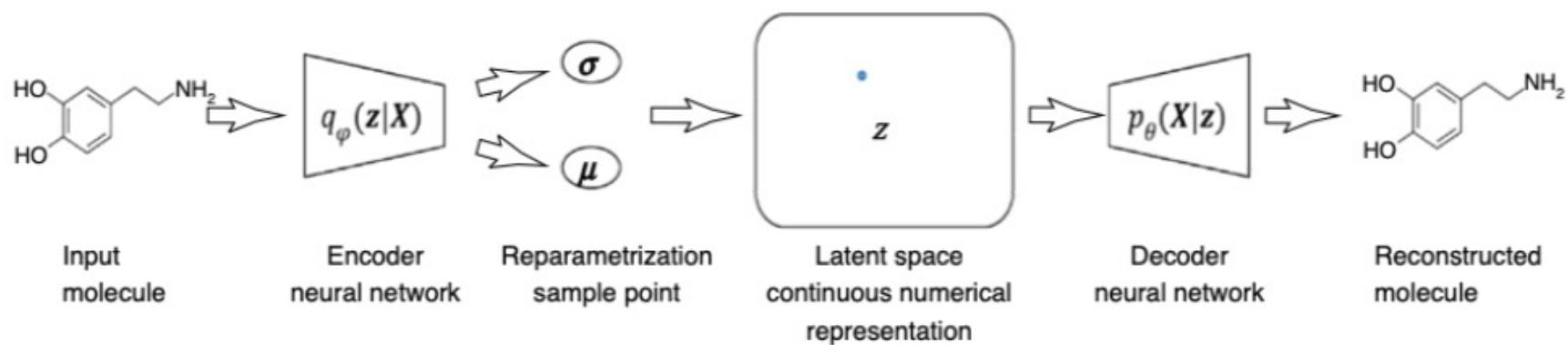
**Training Data.** We collected peptide sequences from the following three publicly accessible sources: “a database for antimicrobial peptides” (ADAM),<sup>28</sup> “antimicrobial peptide database” (APD),<sup>29</sup> and “database of Anuran defense peptides” (DADP).<sup>30</sup> The ADAM database contains sets of sequences for different secondary structures from which we extracted the helical AMPs (cluster ID: AC\_003). From the APD, we retrieved all sequences annotated as “helical”. We included the entire DADP peptide database. The majority of all of these

Also:

Grisoni et al., *Chem Med Chem* 2018

# Structure generation via Variational Autoencoder (VAE)

1. Train an autoencoder
2. It will learn to project molecules in a continuous  $\mathbb{R}^n$  (low  $n$ ) space
3. Project a “lead” molecule
4. Perturb the projection
5. Decode

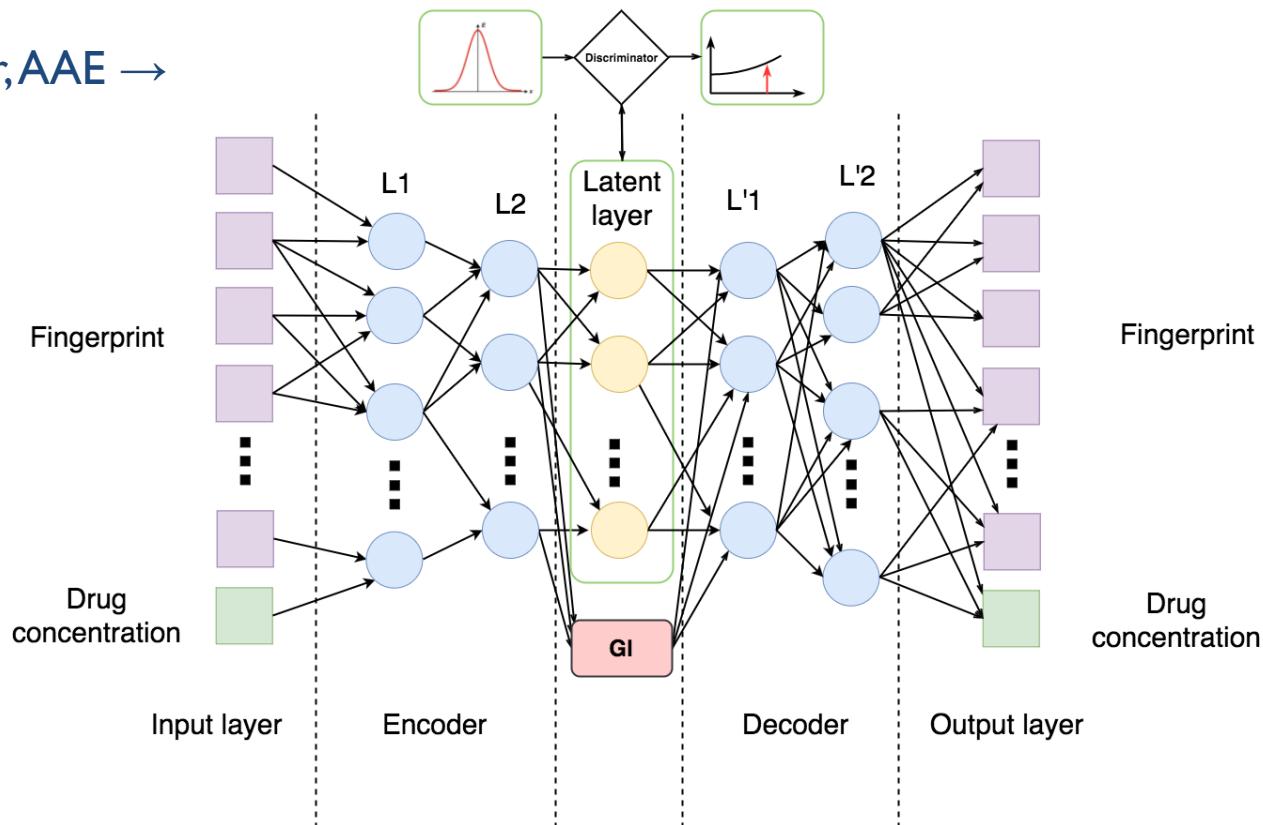


# The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology

Artur Kadurin<sup>1,2,3,4</sup>, Alexander Aliper<sup>2</sup>, Andrey Kazennov<sup>2,7</sup>, Polina Mamoshina<sup>2,5</sup>, Quentin Vanhaelen<sup>2</sup>, Kuzma Khrabrov<sup>1</sup>, Alex Zhavoronkov<sup>2,6,7</sup>



## Adversarial Autoencoder, AAE →



**Figure 1: Architecture of Adversarial Autoencoder (AAE) used in this study.** Encoder consists of two consequent layers L1 and L2 with 128 and 64 neurons, respectively. In turn, decoder consists of layers L'1 and L'2 comprising 64 and 128 neurons. Latent layer consists of 5 neurons one of which is Growth Inhibition percentage (GI) and the other 4 are discriminated with normal distribution.

## Analogy:

molecular fingerprints

→ pixels

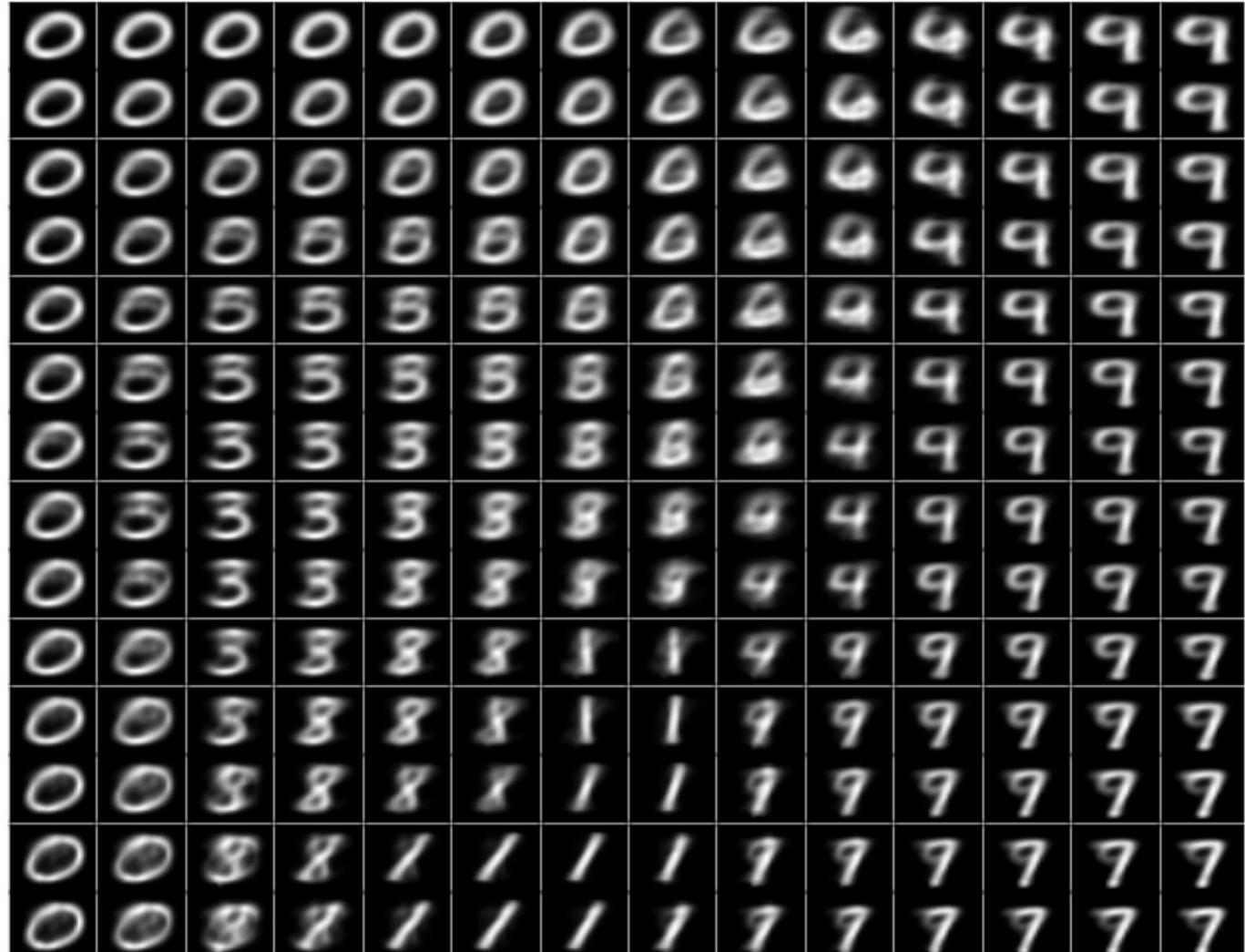
active candidates

→ handwritten digits

latent space

→ 2D plane

Latent axes  
may be made  
*meaningful*  
(Faceapp-like)\*.



\* See e.g. StyleGAN and

"Face editing with GANs". [Link](#).

<https://www.youtube.com/watch?v=dCKbRCUyop8>

<https://towardsdatascience.com/a-wizards-guide-to-adversarial-autoencoders-part-2-exploring-latent-space-with-adversarial-2d53a6f8a4f9>

# Role of simulation

- Time-consuming (compute intensive) methods may serve as inputs
- Examples:
  - QM to create classical FF
  - Classical all-atom FF to create coarse-grained potentials

SCIENTIFIC DATA

OPEN

Data Descriptor: ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules

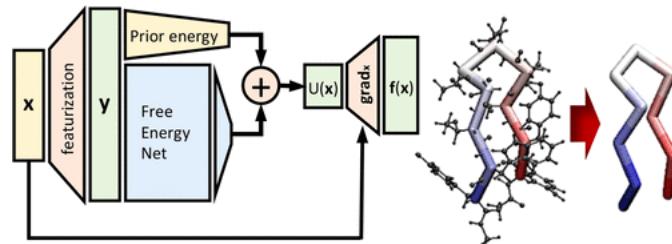
Received: 10 August 2017

Accepted: 27 October 2017

Justin S. Smith<sup>1</sup>, Olexandr Isayev<sup>2</sup> & Adrian E. Roitberg<sup>1</sup>

TorchMD: A Deep Learning Framework for Molecular Simulations

Stefan Doerr, Maciej Majewski, Adrià Pérez, Andreas Krämer, Cecilia Clementi, Frank Noe, Toni Giorgino, and Gianni De Fabritiis\*



**Real-world example:  
Merck's  
activity prediction  
competition**

[doi:10.1021/ci500747n](https://doi.org/10.1021/ci500747n)

[www.kaggle.com/c/MerckActivity](https://www.kaggle.com/c/MerckActivity)

- The **Training** and **Test Sets** each consist of 15 biological activity data sets in comma separated value (CSV) format. Each row of data corresponds to a chemical structure represented by molecular descriptors.
- The **training files** are of the form
  - Column 1: Molecule ID
  - Column 2: Activity. Note that these are raw activity values and different data sets can have activity measured in different units.
  - Column 3-end: Molecular descriptors/features
- The **test files** are in the same format with *Column 2 removed*.
- Molecule IDs and descriptor names are global to all data sets. Thus some molecules will appear in multiple data sets, as will some descriptors.
- The challenge is to predict the activity value for each molecule/data set combination in the test set. To keep predictions for molecules unique to each data set, a data set identifier has been prepended to each molecule ID (e.g., "ACT1\_" or "ACT8\_").

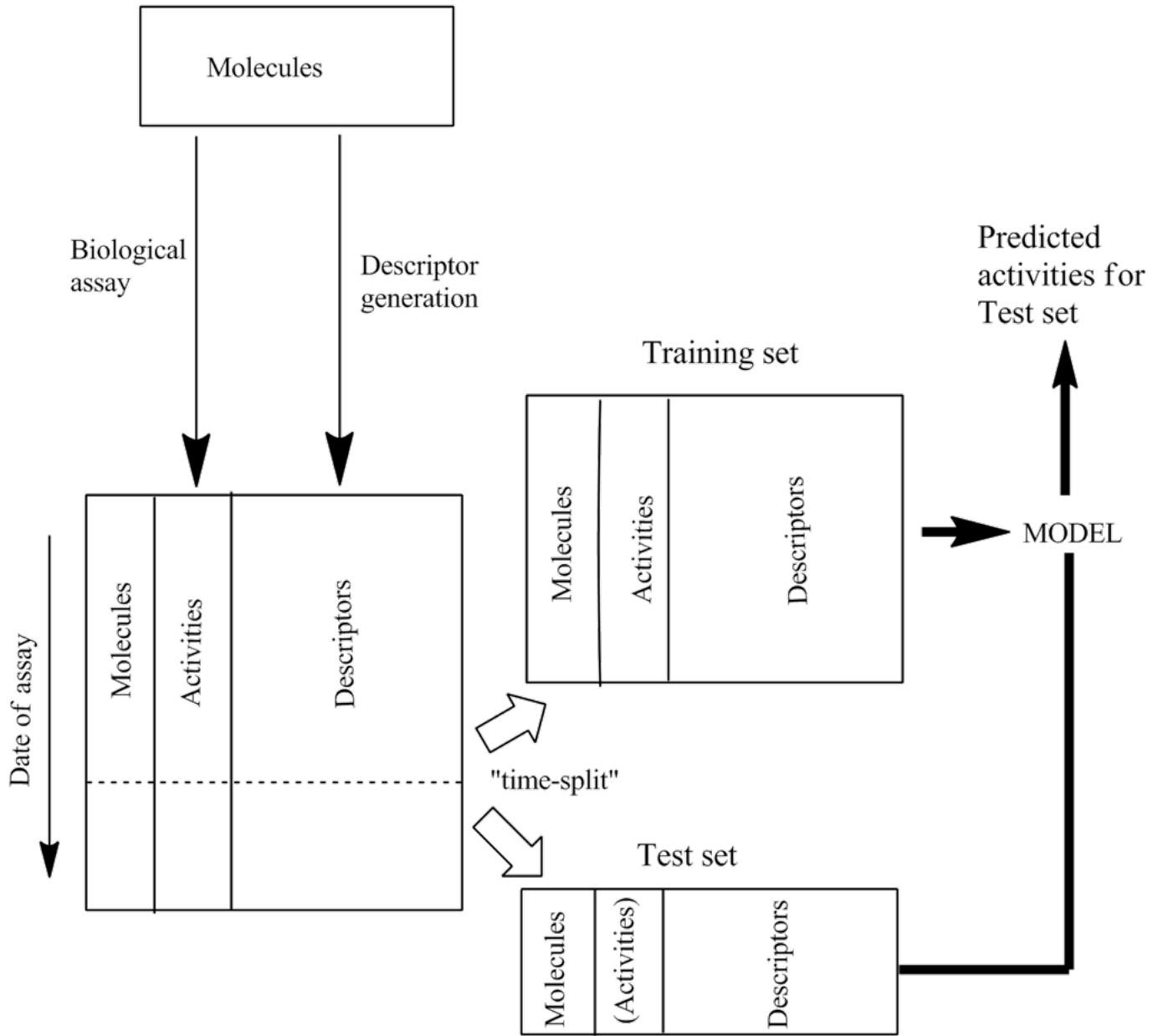
- For each activity, the training/test set split is done by dates of testing. That is, the training set consists of compounds assayed by a certain date, and the test set consists of compounds tested after that date. Therefore it is expected that the distribution of descriptors will not necessarily be the same between the training and test sets.
- We find "time-split" validation is a much more realistic simulation of true prospective prediction in terms of  $R^2$ . We find construction of test sets by random sampling gives  $R^2$ 's that are much too optimistic. That is what makes these data sets more of a challenge.

# Example:



Merck  
molecular  
activity  
challenge  
on Kaggle

Approx 11k  
descriptors per  
molecule  
2k-37k activities  
per target



Goodness of prediction is measured by  $R^2$  between predicted activities and observed activities (withheld from the contestant) of the test set.

# **Final words**

# Take-home messages

- ANN are *mathematical models* that can be trained via examples to reproduce arbitrary functions
- Training is however not trivial: *many* examples are required, and independent test sets must be observed
- Failure to do so yields *overfitting*: models which predict only cases “already seen” in training. They don’t generalize or predict.
- Deeper networks can be trained for a variety of tasks, including *de-novo* generation

# Ethical issues



- Where do the data come from?
- Database errors contain biases. They reflect on results, yet they are hidden
- Cognitive biases
  - Hope in “magic boxes”
  - The machine told me and I don’t want responsibility
- Optimization vs. understanding.
  - Black boxes do not empower discovery
  - Is this “epistemologically” desirable for society?
- SOTA publishing selection
- Immense hype, AI winter.

# Resources (self-study)

- Mathematics: [3Blue1Brown](#) (YouTube)
- Prof. White's *Deep Learning for Molecules & Materials* - [dmol.pub](#)
- You should learn Python
- Compute power (“notebooks”):  
[Google Colaboratory](#), [Kaggle](#)

# Resources (datasets)

- Datasets, competitions: Kaggle (e.g. Merck's)
- Toxicology data: Tox21.gov (NIH)
- Many benchmark sets: [moleculenet.ai](https://moleculenet.ai) →

- **BBBP**: Binary labels of blood-brain barrier penetration(permeability).

Classification

- **Tox21**: Qualitative toxicity measurements on 12 biological targets, including nuclear receptors and stress response pathways.

Classification

- **ToxCast**: Toxicology data for a large library of compounds based on *in vitro* high-throughput screening, including experiments on over 600 tasks.

Classification

- **SIDER**: Database of marketed drugs and adverse drug reactions (ADR), grouped into 27 system organ classes.

Classification

- **ClinTox**: Qualitative data of drugs approved by the FDA and those that have failed clinical trials for toxicity reasons.

Classification

# References: reviews (many!)

- Gawehn, E., Hiss, J.A. & Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **35**, 3–14 (2016).
- Colwell, L. J. Statistical and machine learning approaches to predicting protein–ligand interactions. *Current Opinion in Structural Biology* **49**, 123–128 (2018).
- Pérez, A., Martínez-Rosell, G. & De Fabritiis, G. Simulations meet machine learning in structural biology. *Current Opinion in Structural Biology* **49**, 139–144 (2018).
- Chen, H. The rise of deep learning in drug discovery. *Drug Discovery Today* **23**, 10 (2018).
- Lavecchia, A. Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug Discovery Today* **24**, 2017–2032 (2019).
- Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P.W. Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.* **4**, 828–849 (2019).
- Li, H., Sze, K., Lu, G. & Ballester, P. J. Machine-learning scoring functions for structure-based drug lead optimization. *WIREs Comput Mol Sci* (2020) doi:[10.1002/wcms.1465](https://doi.org/10.1002/wcms.1465).

Drug Discovery Today • Volume 23, Number 6 • June 2018

REVIEWS

Check for updates



The rise of deep learning in drug discovery

Hongming Chen<sup>1</sup>, Ola Engkvist<sup>1</sup>, Yinhai Wang<sup>2</sup>, Marcus Olivercrona<sup>1</sup> and Thomas Blaschke<sup>1</sup>

<sup>1</sup>Hit Discovery, Discovery Sciences, Innovative Medicines and Early Development Biotech Unit, AstraZeneca R&D Gothenburg, Mölndal 43183, Sweden  
<sup>2</sup>Quantitative Biology, Discovery Sciences, Innovative Medicines and Early Development Biotech Unit, AstraZeneca, Unit 310, Cambridge Science Park, Milton Road, Cambridge CB4 0WG, UK

Drug Discovery Today • Volume 24, Number 10 • October 2019

REVIEWS

Check for updates



Teaser This paper focuses on deep learning approaches in the context of drug discovery for designing new effective molecules, predicting for the desired molecular property profiles and planning synthesis.

Deep learning in drug discovery: opportunities, challenges and future prospects

Antonio Lavecchia

Department of Pharmacy, "Drug Discovery" Laboratory, University of Napoli "Federico II", via D. Montesano 49, I-80131 Napoli, Italy

Artificial Intelligence (AI) is an area of computer science that simulates the

Antonio Lavecchia is Professor of Medicinal Chemistry and Head of Drug Discovery Laboratory at the Department of Pharmacy of the University of Napoli Federico II, where he is also Vice-dean Research.



# References

- Jiménez, J., Doerr, S., Martínez-Rosell, G., Rose, A. S. & De Fabritiis, G. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* **33**, 3036–3042 (2017).
- Kadurin, A. et al. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* **8**, (2017).
- Yuan, W. et al. Chemical Space Mimicry for Drug Discovery. *J. Chem. Inf. Model.* **57**, 875–882 (2017).
- Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **4**, 120–131 (2018).
- Müller, A. T., Hiss, J. A. & Schneider, G. Recurrent Neural Network Model for Constructive Peptide Design. *J. Chem. Inf. Model.* (2018) doi:[10.1021/acs.jcim.7b00414](https://doi.org/10.1021/acs.jcim.7b00414).
- Merk, D., Friedrich, L., Grisoni, F. & Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Mol. Inf.* **37**, 1700153 (2018).
- Jiménez Luna, J., Skalic, M., Martinez-Rosell, G. & De Fabritiis, G. KDEEP: Protein-ligand absolute binding affinity prediction via 3D-convolutional neural networks. *J. Chem. Inf. Model.* (2018) doi:[10.1021/acs.jcim.7b00650](https://doi.org/10.1021/acs.jcim.7b00650).
- Zhavoronkov, A. et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat Biotechnol* **37**, 1038–1040 (2019).
- Skalic, M., Varela-Rial, A., Jiménez, J., Martínez-Rosell, G. & De Fabritiis, G. LigVoxel: inpainting binding pockets using 3D-convolutional neural networks. *Bioinformatics* **35**, 243–250 (2019).
- Skalic, M., Jiménez, J., Sabbadin, D. & De Fabritiis, G. Shape-Based Generative Modeling for de Novo Drug Design. *J. Chem. Inf. Model.* **59**, 1205–1214 (2019).

**End**