Deep Learning for Biological Data Analysis

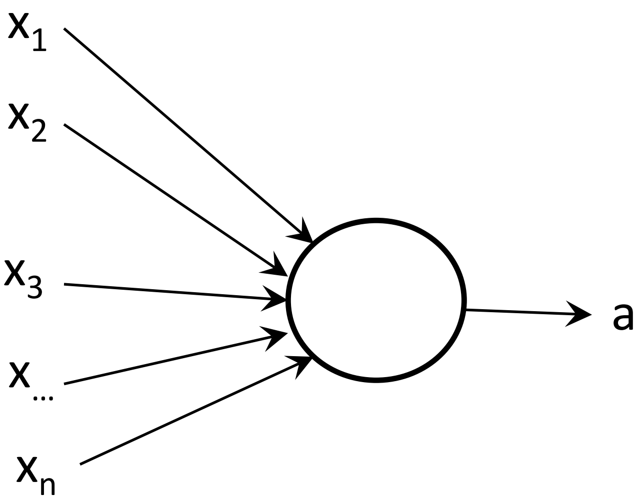
Oleg Moskvin

BioComP meeting, Oct. 23, 2017

Outline

- **Deep Learning demystified**
  – An artificial neuron
  – Main architecture types and applications of deep neural networks
- **Trends in DNN application to *omics data**
- **Development platforms**

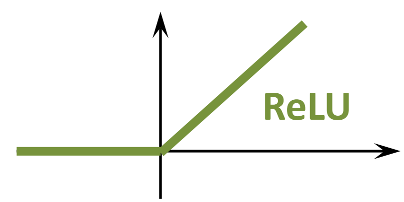
An Artificial Neuron



weights
bias
activation

Activation functions:

ReLU – Rectified Linear Unit
LeakyReLU
ThresholdedReLU
Sigmoid
Softmax (normalized exponential; final layer for multiple output classes)
Tanh
ELU – Exponential Linear Unit
SELU – Scaled ELU
…



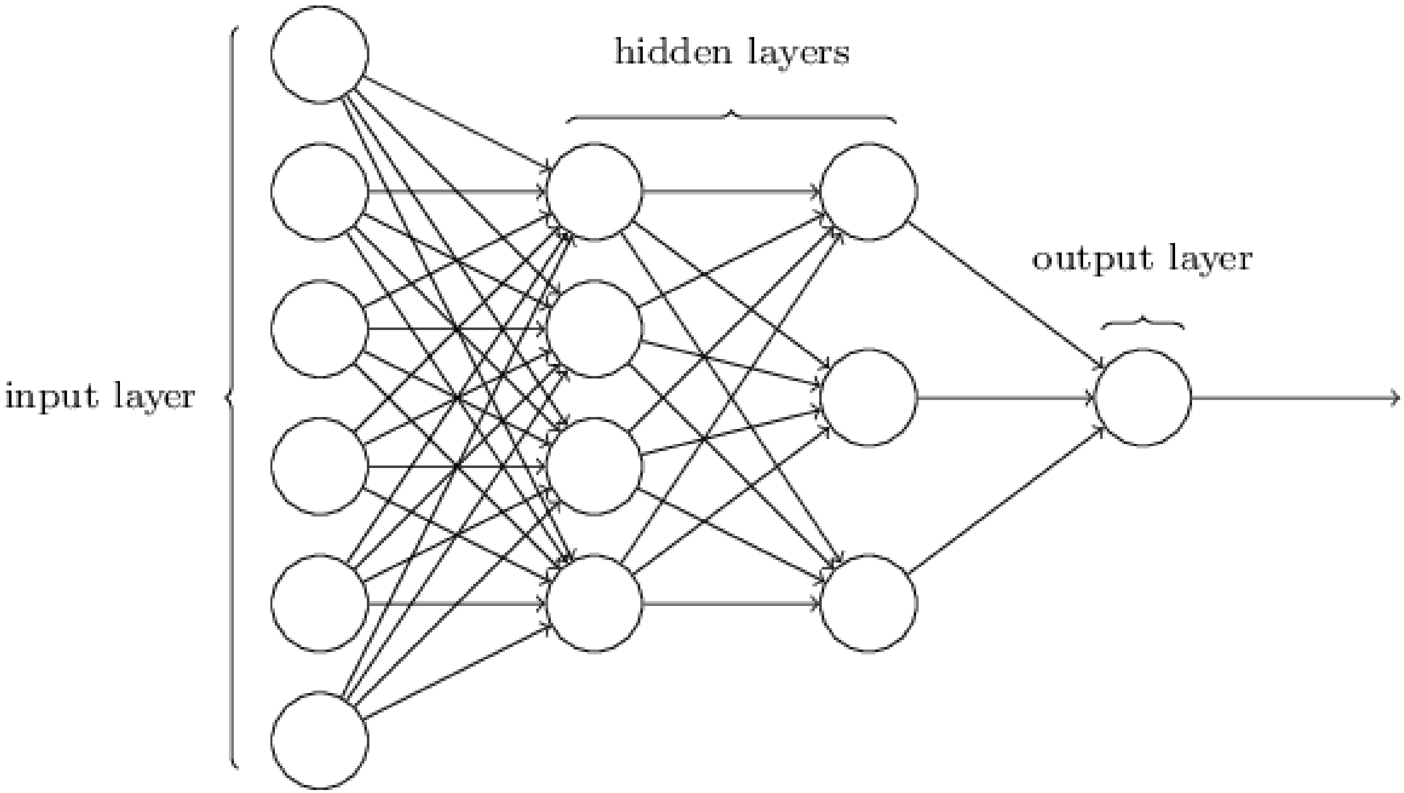
A neuron does 2 things:

- **Linear summary of the inputs** $z = w_1 * x_1 + w_1 * x_1 + \ldots + w_n * x_n + b$
- **Non-linear transformation of the result** $a = f(z)$

Architecture types of the DNNs

- Fully Connected (multilayer perception)
- Convolutional
- Recurrent

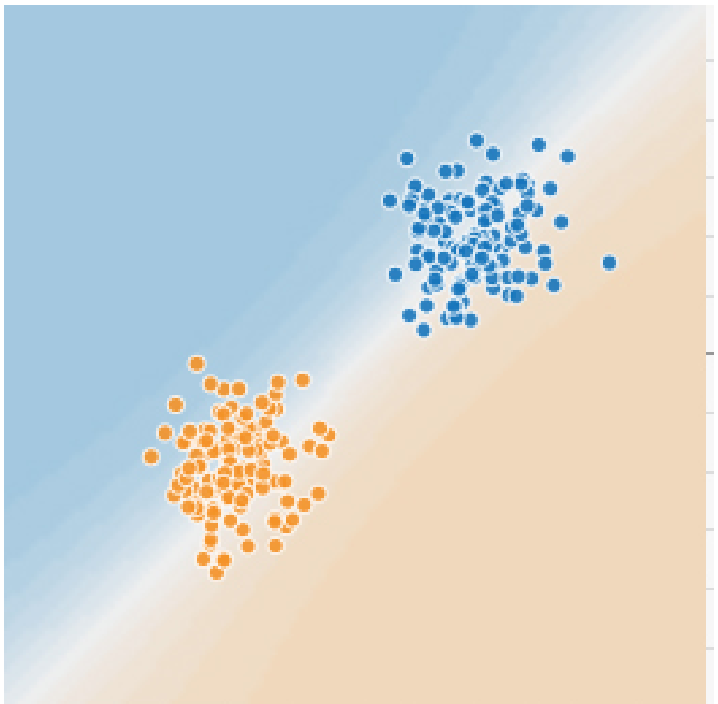
Fully Connected Networks



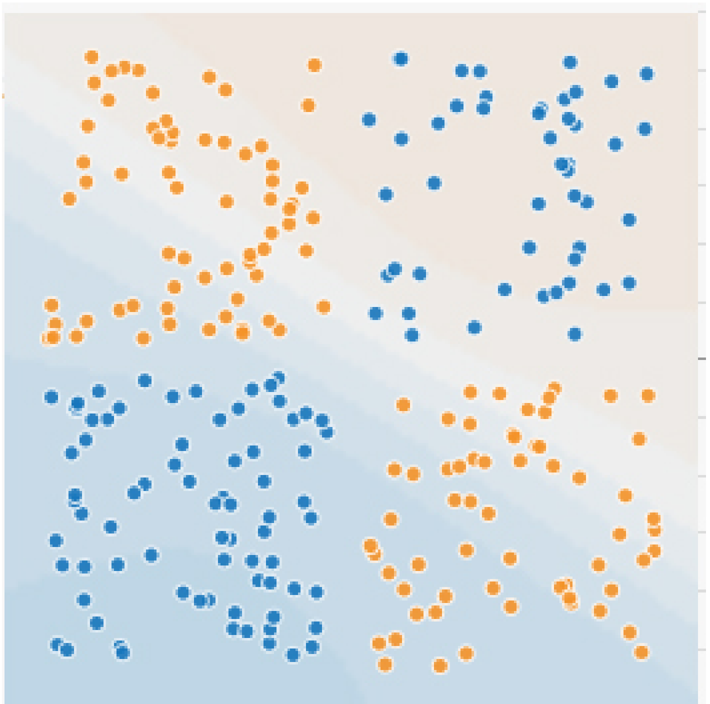
http://blog.gadgetfactory.net/2016/12/free-ebook-neural-networks-and-deep-learning/

Applications: general task of finding non-linear relationships and complex patterns in the data

Fully Connected Networks: Playground Examples

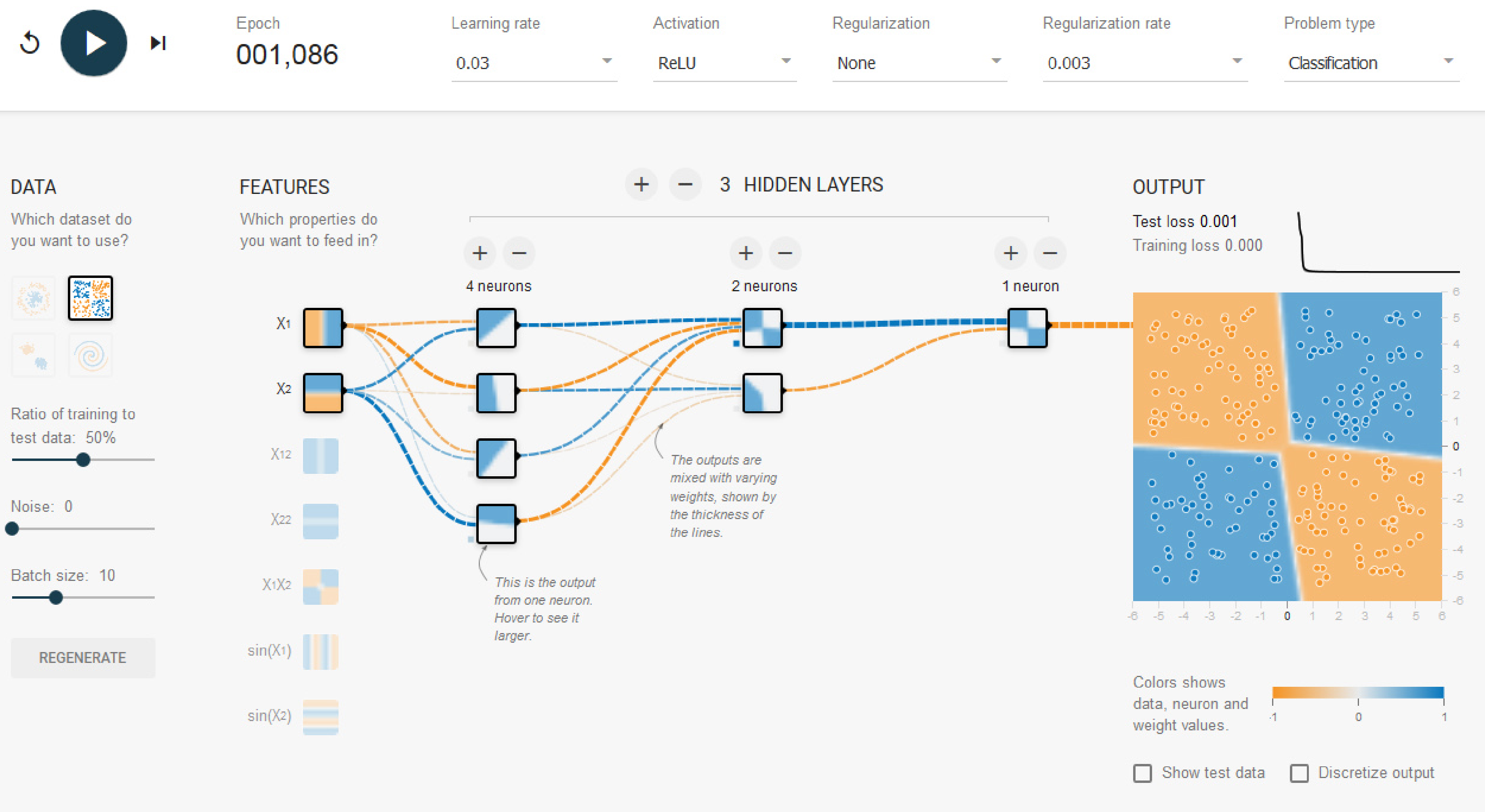


Classification problem easily solvable using a linear function

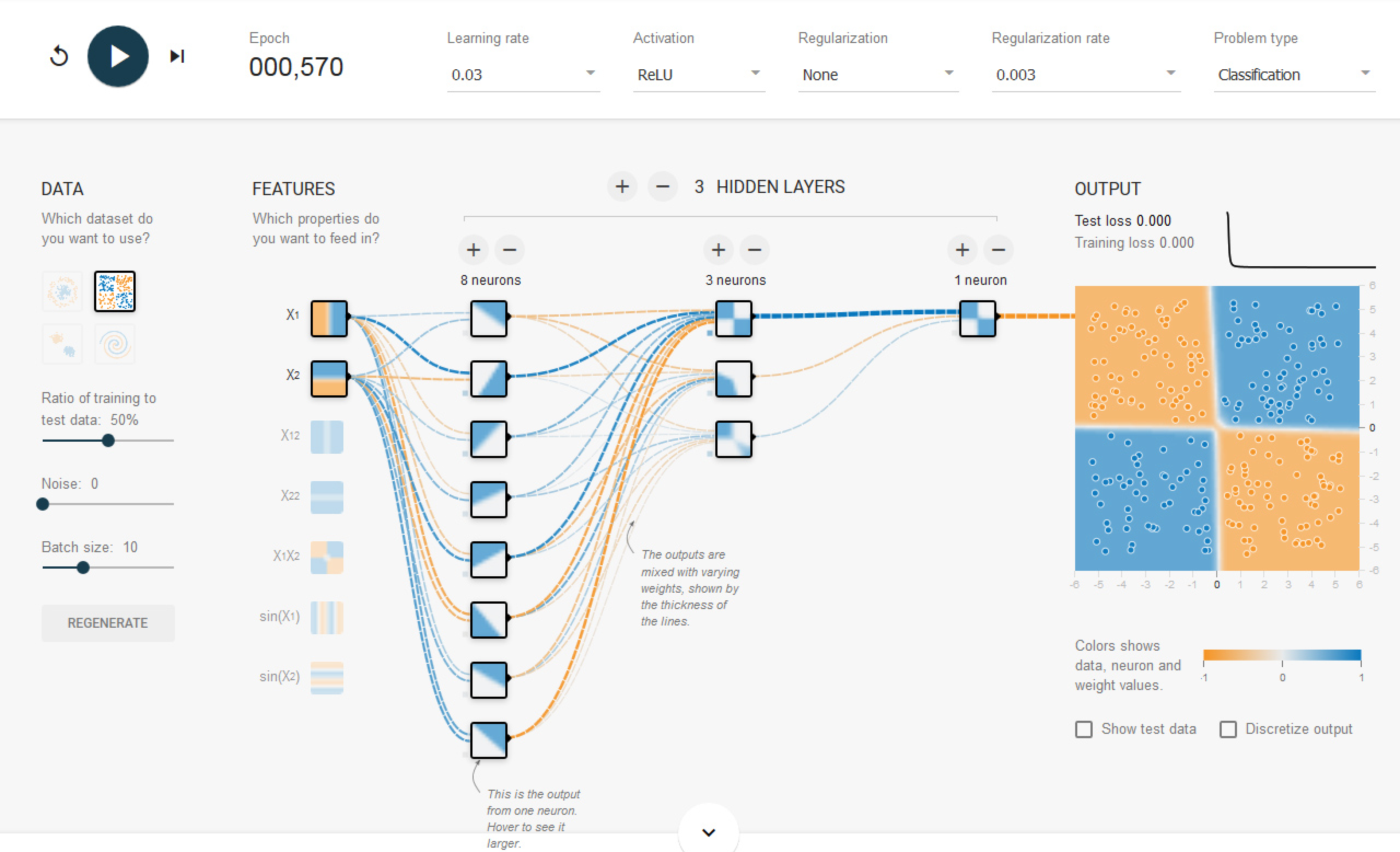


This is something more interesting…

Fully Connected Networks: Playground Examples

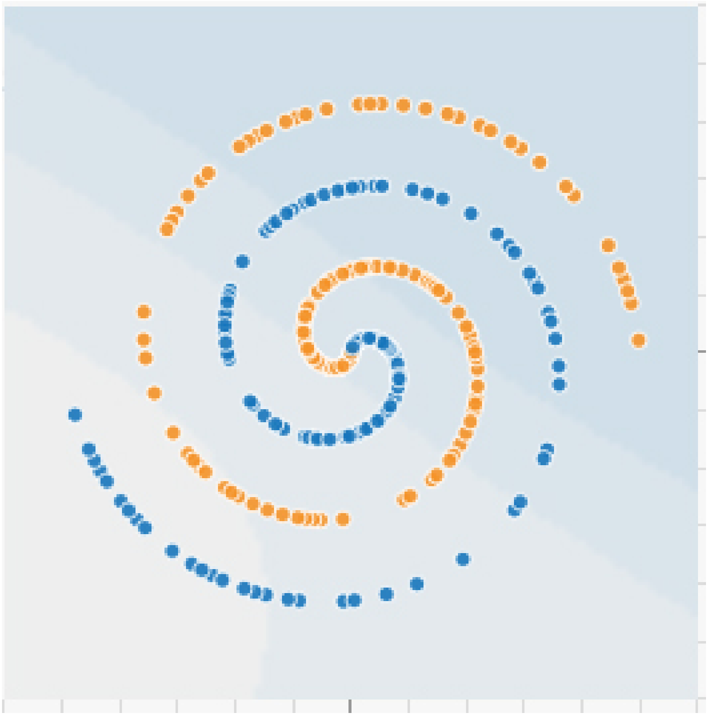


Fully Connected Networks: Playground Examples

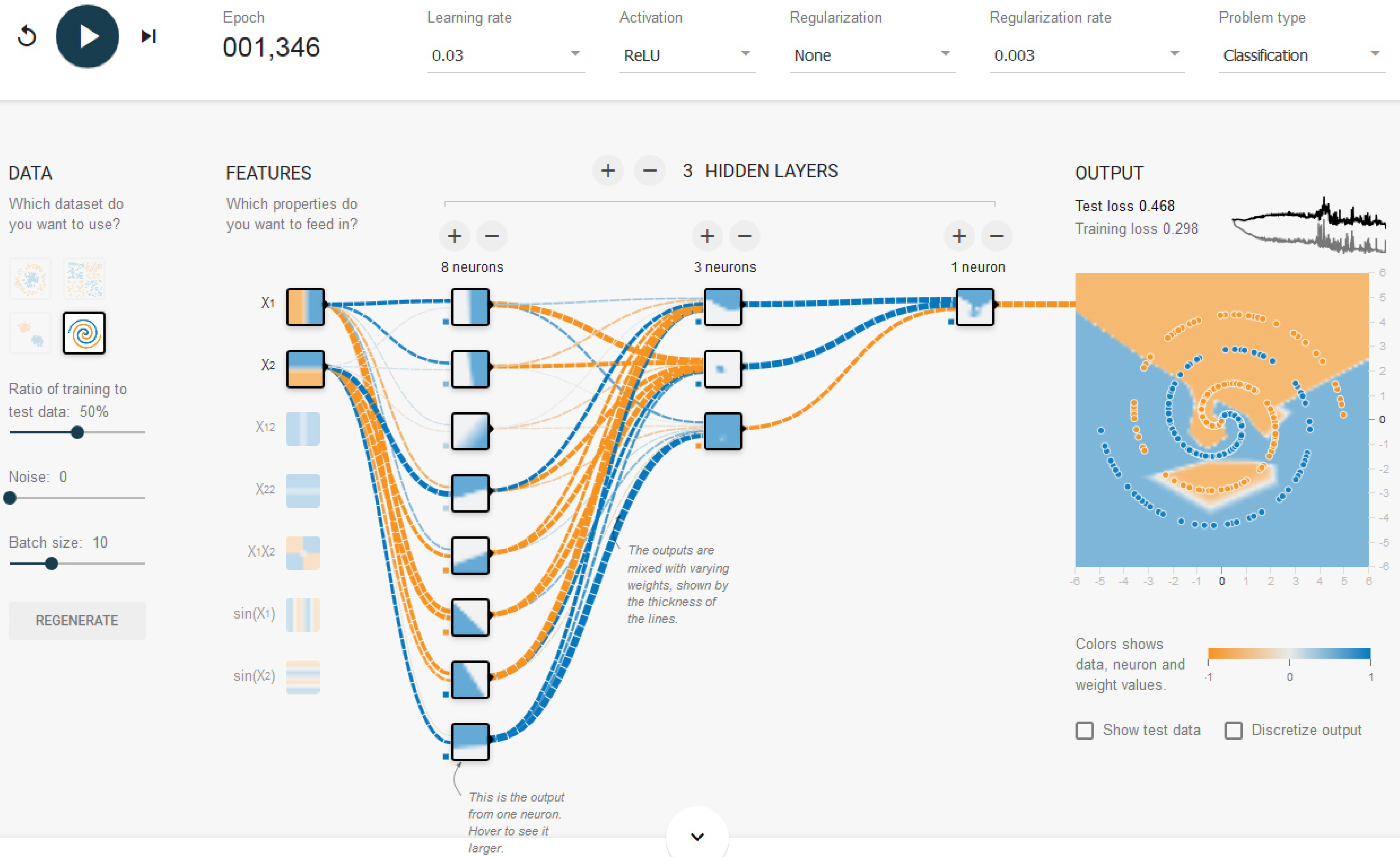


P.S. one may argue this is a stupid solution because using feature engineering (X1 * X2) you may get away with a single neuron…

More challenging problem

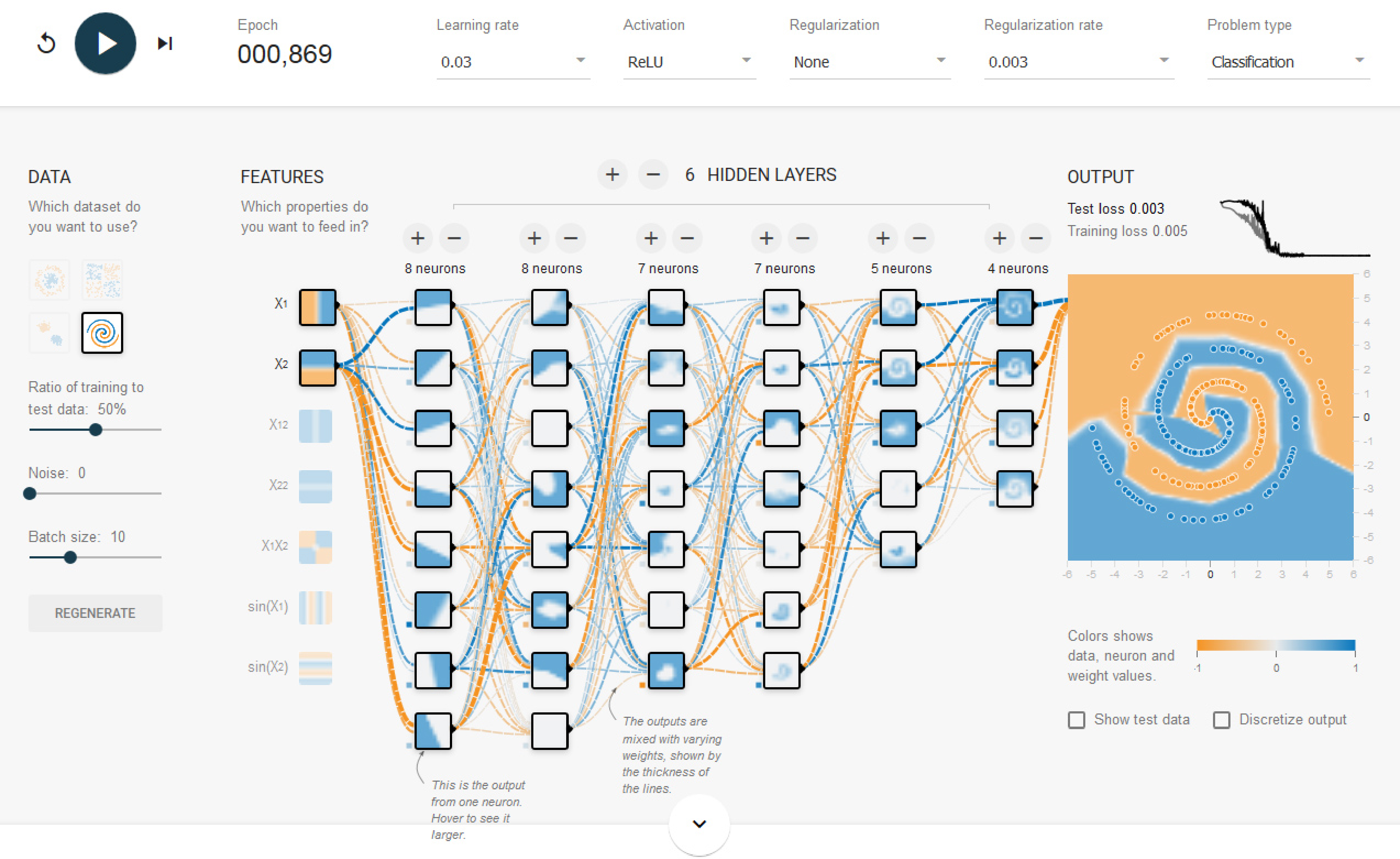


Fully Connected Networks: Playground Examples



http://DNA.today

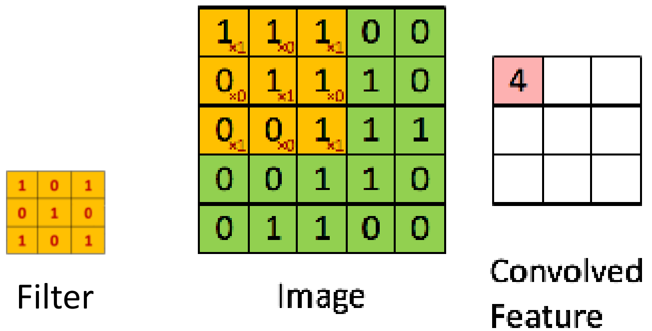
Fully Connected Networks: Playground Examples



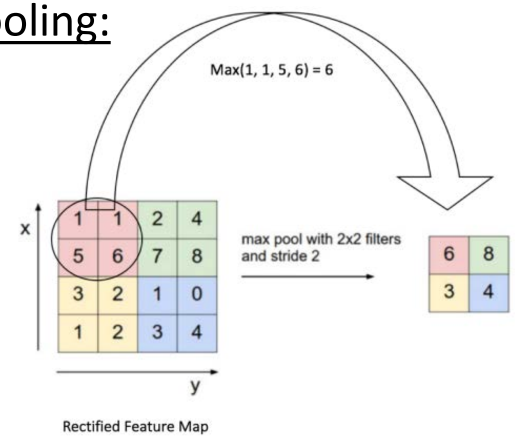
Convolutional Neural Networks

- <u>The issues</u>: 1) exploding number of pixel-level parameters for larger images; 2) for a 2-dimensional input, information on the neighbors / 2D context of each value needs to be accounted for
- <u>The solution</u>: Combination of convolutional layers, pooling layers and fully connected layers

<u>Convolution:</u>



<u>Pooling:</u>

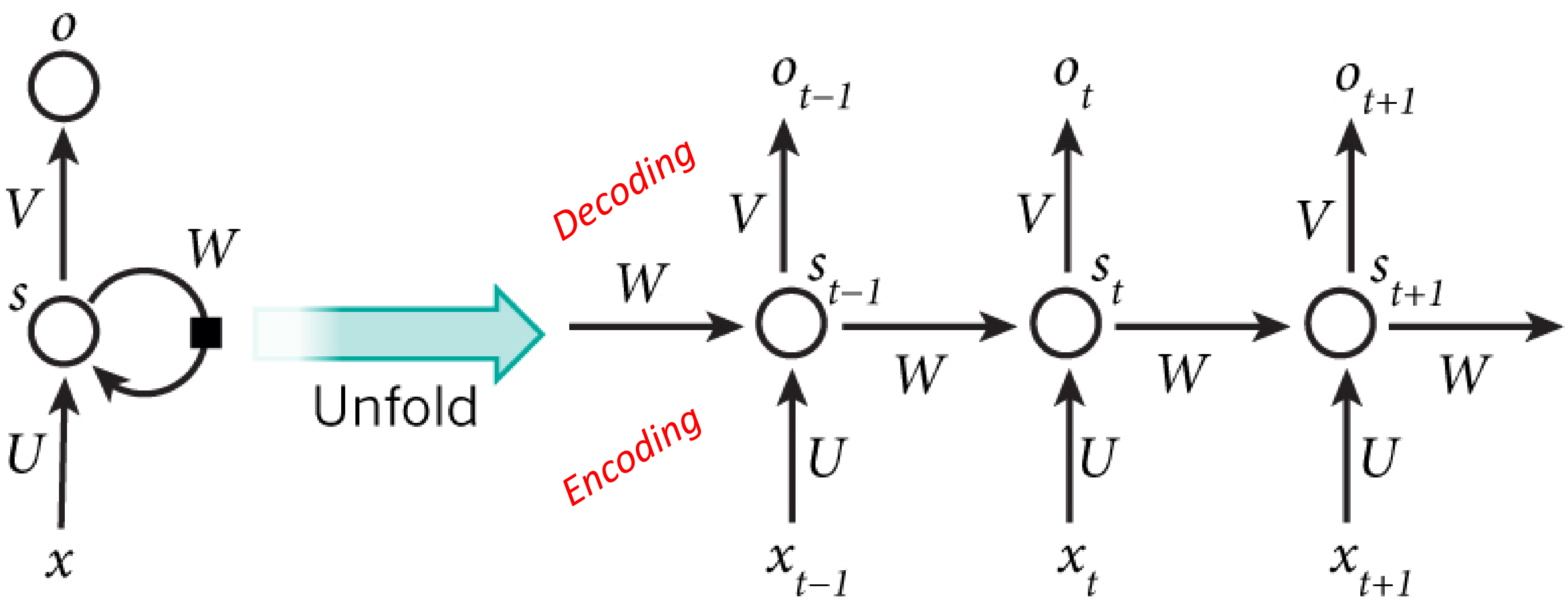


https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Typical application: image recognition

Recurrent Neural Networks

- <u>The issue</u>: for a sequential input, information on the previous state needs to be accounted for
- <u>The solution</u>: loops in the network

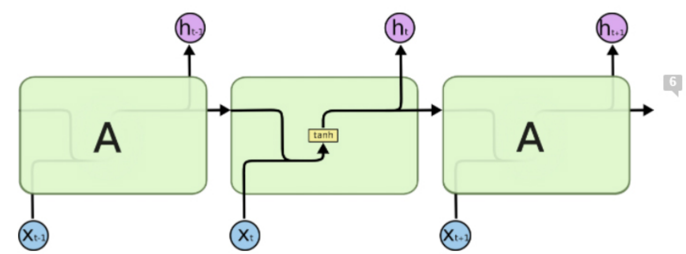


http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

Applications: speech recognition, translation, games

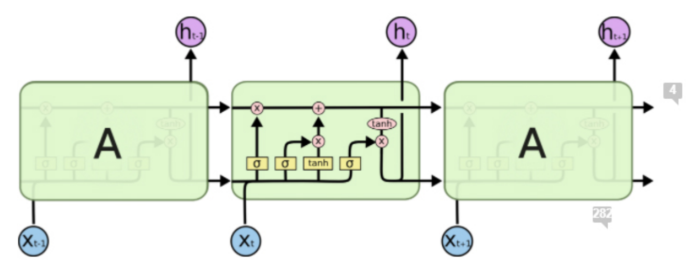
Further developments in RNN

- Explicit separation of encoding and decoding stages into separate networks (whole string digestion -> then start output generation)
- Facilitating capture of long-range dependencies by replacing a single neuron with a complex “cell” as a recurring unit -> Long Short Term Memory (LSTM) architecture



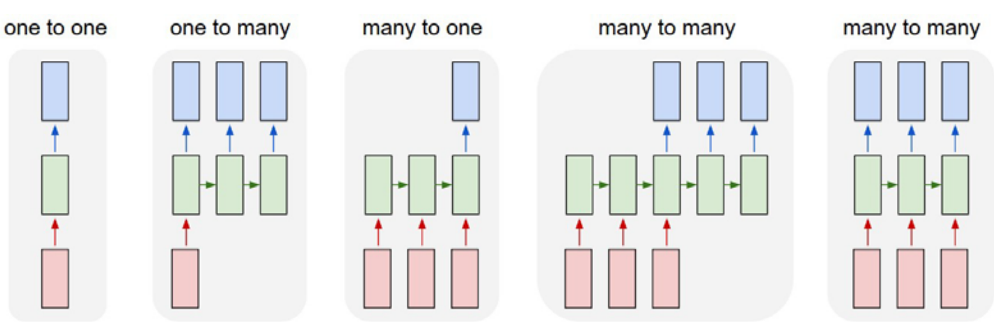
The repeating module in a standard RNN contains a single layer.

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



The repeating module in an LSTM contains four interacting layers.

Flexibility of RNN architectures

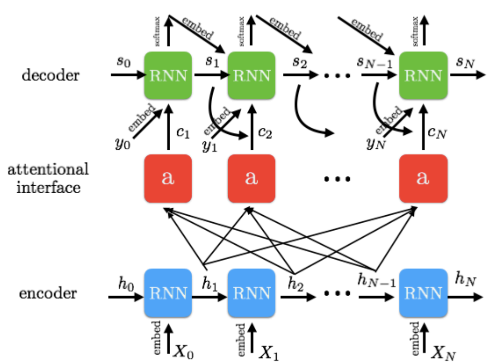


http://karpathy.github.io/2015/05/21/rnn-effectiveness

http://colah.github.io

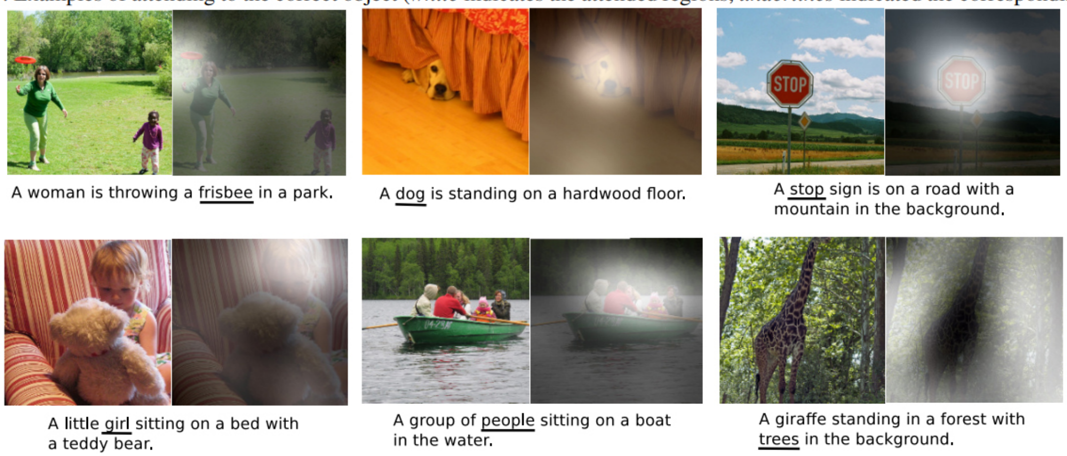
RNN Twist: Attention Interface

- The issue: RNN's internal representation has fixed length which results in degrading performance for longer input strings.
- The solution: allow the decoder to pick the most relevant set of encodings by adding "attentional interface" that generates a context vector for the decoder; encoder inputs are weighted to accurately predict the decoder target



theneuralperspective.com

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



Xu et.al. (2015) Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
https://arxiv.org/abs/1502.03044

What Recurrent Neural Networks can do

- Potential basis for development of General Purpose AI
- Demonstrated potential to bypass the need of any human expert to aid the learning

What Recurrent Neural Networks can do

- Potential basis for development of General Purpose AI
- Demonstrated potential to bypass the need of any human expert to aid the learning
  - AI Stage 1: approaching human intelligence
  - AI Stage 2: human intellectual traits become an impediment on the road of AI-based learning

Near future vs. remote future?

What Recurrent Neural Networks can do

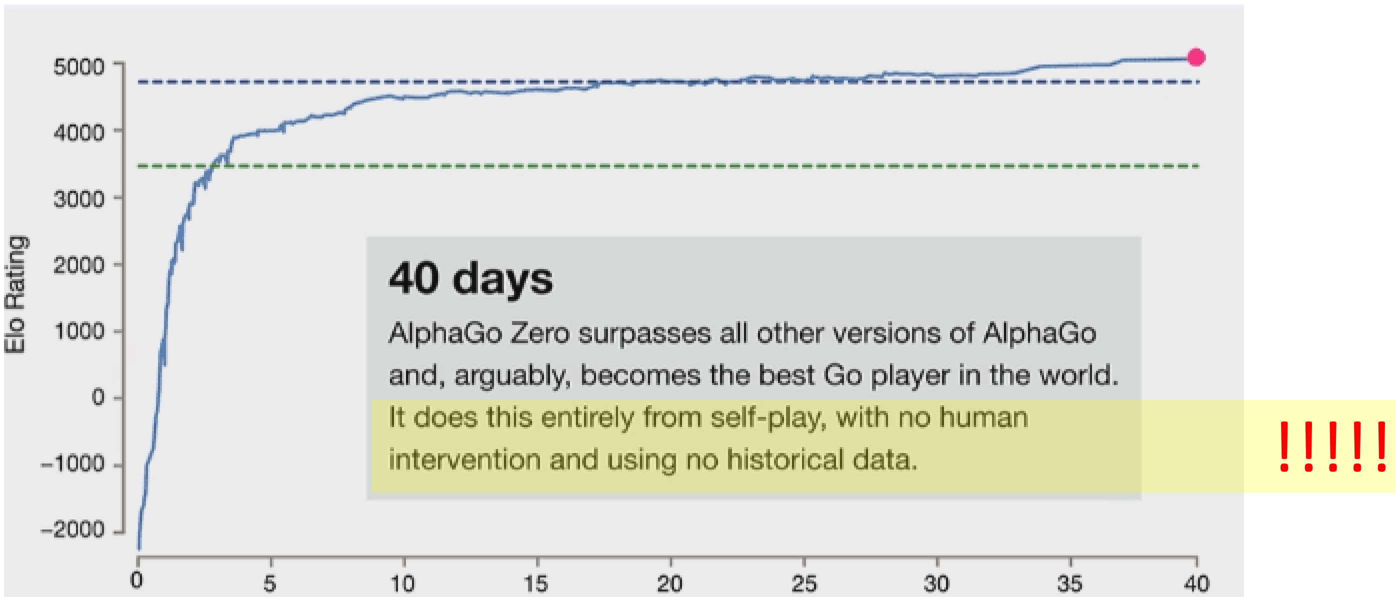
- Potential basis for development of General Purpose AI
- Demonstrated potential to bypass the need of any human expert to aid the learning
  - AI Stage 1: approaching human intelligence
  - AI Stage 2: human intellectual traits become an impediment on the road of AI-based learning
  - The reality: we are already at Stage 2!

An Example of the “Stage 2”: Game of GO

- In 2015, Alpha Go defeated the world Go champion; what happened 2 years later?
- Now we have **Alpha Go Zero** that reached this level in 3 days of training… What happened after those 3 days?

https://www.theguardian.com/science/2017/oct/18/its-able-to-create-knowledge-itself-google-unveils-ai-learns-all-on-its-own

An Example of the “Stage 2”: Game of GO



- Based on Recurrent Neural Networks
- “It discovered one common play, called a joseki, in the first 10 hours. Other moves, with names such as ‘small avalanche’ and ‘knight’s move pincer’ soon followed. **After three days, the program had discovered brand new moves that human experts are now studying”**.
- “You can see it rediscovering thousands of years of human knowledge.”
- “[this accomplishment] opens a new book, which is where computers teach humans…”

https://www.theguardian.com/science/2017/oct/18/its-able-to-create-knowledge-itself-google-unveils-ai-learns-all-on-its-own

Trends in application to *omics data

General idea

- Biological systems are no less non-linear than images!
- Our models of their operation are limited
- Both **a)** toolsets for auto-defining the models from the data via deep neural networks and **b)** abundant data (*omics data flood) are already there
- Let’s marry the two!

Dynamics of the application to bioinformatics

- PubMed search for words "*deep, learning, bioinformatics*" in the abstract (Sept 19, 2017) – 54 results

- 2009 – 1 paper
- 2010 – 1 paper
- 2011 – 0 papers
- 2012 – 2 papers
- 2013 – 3 papers
- **2014** – 2 papers
- 2015 – **6** papers
- 2016 – **13** papers
- 2017 (8+ months) – **26** papers

Lag phase

Exponential growth

This search is not supposed to capture significant part of the publications. Still, it shows the trend.

The "conference excitement" of 2014 marked the start of the exponential phase in publications (2015)

*omics (gene expression, sequence analysis) and the DNN architectures

“Gene expression inference with deep learning”

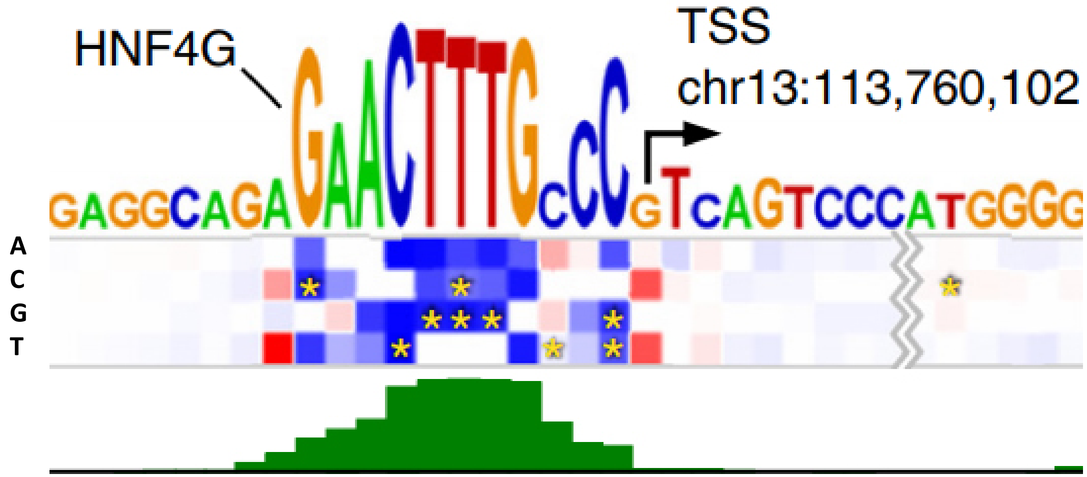
- *Bioinformatics (2016) 32(12):1832-1839*
- 111,009 Affy + (2,921+462) Illumina RNA-seq expression profiles as training data
- 943 landmark genes and 9,520 target genes
- Task: <u>model expression of the target genes given the landmark gene expression</u>
- Architectures: Multilayer Perception (1-3 hidden layers, 3K-9K neurons / layer)

Sequence analysis: Predicting <u>promoters in a sequence</u>

- *"Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks" (PLoS One 2017, 12:*e0171410*)*
- Why CNN for sequence analysis??? - *"convolution filters can capture information on functional sequence motifs"*
  - 200 convolution filters of length 21
  - Max pooling layer
  - 128-neuron fully connected layer (ReLU activation)
  - Output layer (Sigmoid activation); classification (promoter vs. non-promoter sequence)

Sequence analysis: DeepBind

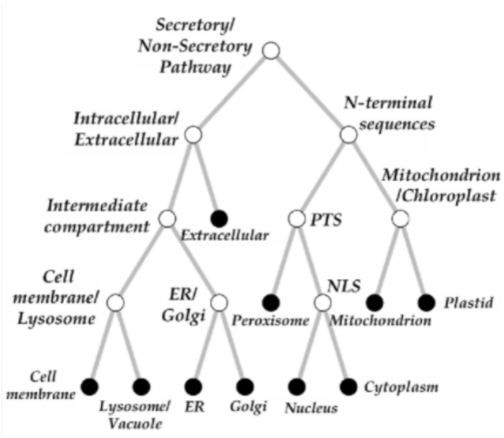
- *"Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning" (Nat. Biotechnol. 33(2015): 831-838)*
- Uses CNN for sequence pattern detection
- Nice byproduct is the ability to predict both the importance of each nucleotide and effect of all possible mutations on the binding site functionality. The predictions are highly coherent with known effects of disruptive mutations



Sequence analysis: Predicting protein subcellular localization from sequence

- *"DeepLoc: prediction of protein subcellular localization using deep learning" (Bioinformatics, **July 2017**)*
- **recurrent neural network** that processes the entire protein sequence and an **attention mechanism** identifying protein regions important for the subcellular localization
- Sequential use of CNN (motif extraction) -> RNN (extracting spatial dependencies between aminoacids) -> attention mechanism ranks aminoacids by their predictive value -> fully connected dense layer

All the 3 NN types were employed!



Bioinformatics, btx431

Slicing publications by architecture: range of problems covered

More titles involving fully connected DNN

- Deep learning of the tissue-regulated splicing code (24931975)
- DANN: a deep learning approach for annotating the pathogenicity of genetic variants (25338716)
- DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations (28155641)
- Opening up the blackbox: an interpretable deep neural network-based classifier for cell-type specific enhancer predictions (27490187)
- Deep Learning in Label-free Cell Classification (26975219)
- Unsupervised deep learning reveals prognostically relevant subtypes of glioblastoma (28984190)

More titles involving CNN

- Classification of breast cancer histology images using Convolutional Neural Networks (28570557)
- Convolutional Neural Networks for Biomedical Text Classification: Application in Indexing Biomedical Articles
- nRC: non-coding RNA Classifier based on structural features (28785313)
- DeepSite: protein-binding site predictor using 3D-convolutional neural networks (28575181)
- Denoising genome-wide histone ChIP-seq with convolutional neural networks (28881977)
- DeepChrome: deep-learning for predicting gene expression from histone modifications (27587684)
- Convolutional neural network architectures for predicting DNA-protein binding (27307608)
- A neural network multi-task learning approach to biomedical named entity recognition (28810903)
- 3D deep convolutional neural networks for amino acid environment similarity analysis (28615003)
- Deep convolutional neural networks for annotating gene expression patterns in the mouse brain (25948335) [image analysis, ISH data]
- DeepPep: Deep proteome inference from peptide profiles (28873403)
- Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks (27197224)

More titles involving RNN

- Sequence-specific bias correction for RNA-seq data using recurrent neural networks (28198674)
- Recurrent neural network based hybrid model for reconstructing gene regulatory network (27570069)
- Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks (28011771)
- Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility (28430949)
- Reconstructing Genetic Regulatory Networks Using Two-Step Algorithms with the Differential Equation Models of Neural Networks (28748400)
- Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts (28339747)
- Recurrent neural network-based modeling of gene regulatory network using elephant swarm water search algorithm (28659000)
- Doctor AI: Predicting Clinical Events via Recurrent Neural Networks (28286600)
- DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads (28582401)

More titles involving combinations of architectures

- DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences (27084946)
- De novo peptide sequencing by deep learning (28720701) [CNN + RNN]
- Low Data Drug Discovery with One-Shot Learning (28470045) [CNN + LSTM]
- MusiteDeep: a deep-learning framework for general and kinase-specific phosphorylation site prediction (29036382) [CNN + novel 2D attention]
- Drug drug interaction extraction from biomedical literature using syntax convolutional neural network (27466626) [CNN with novel syntax embedding + autoencoder]
- RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach (28245811) [CNN + Deep Belief]
- De novo identification of replication-timing domains in the human genome by deep learning (26545821) [DNN + HMM]
- TITER: predicting translation initiation sites by deep learning (28881981) [LSTM + CNN]
- BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone (28334114) [CNN + BRNN]

What do we see?

- All the 3 architecture classes of DNNs have made their way to help solving computational biology problems, including architecture combinations that initially look specific to "foreign" domains (such as CNN + BLSTM with attention)
- In particular, CNN + RNN combination grows in popularity for sequence analysis applications

Development Platforms

Data scientist's "Python vs. R" dilemma

- **R** has the richest toolbox of statistical packages and is deeply "rooted" in many existing analytical workflows but is often referred to as being slow and not really scalable; deep learning extensions for R were rudimentary compared to Python
- **Python** is now often being referred to as The Big Data Language and the leading language for Deep Learning

A solution for an R-friendly scalable ML: H2O

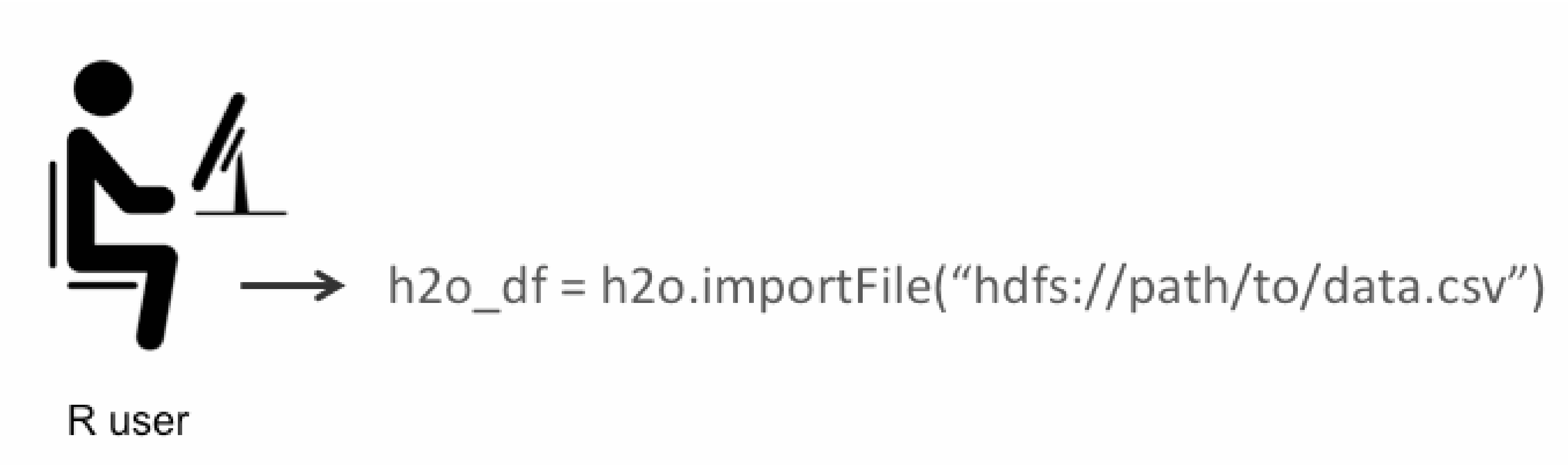
- **H2O is Big Data-friendly ML platform**
  - Scalability: built on Spark / Hadoop
  - Interfaces to R, Python, Scala, Java; has web UI
  - Handy ML toolset; transparent support for common data types
  - Grid search for hyperparameter optimization and model selection
  - Open source (Apache 2.0 license)

"With H2O, enterprises like PayPal, Nielsen Catalina, Cisco, and others can use all their data without sampling to get accurate predictions faster"

(R H2O booklet, page 6)

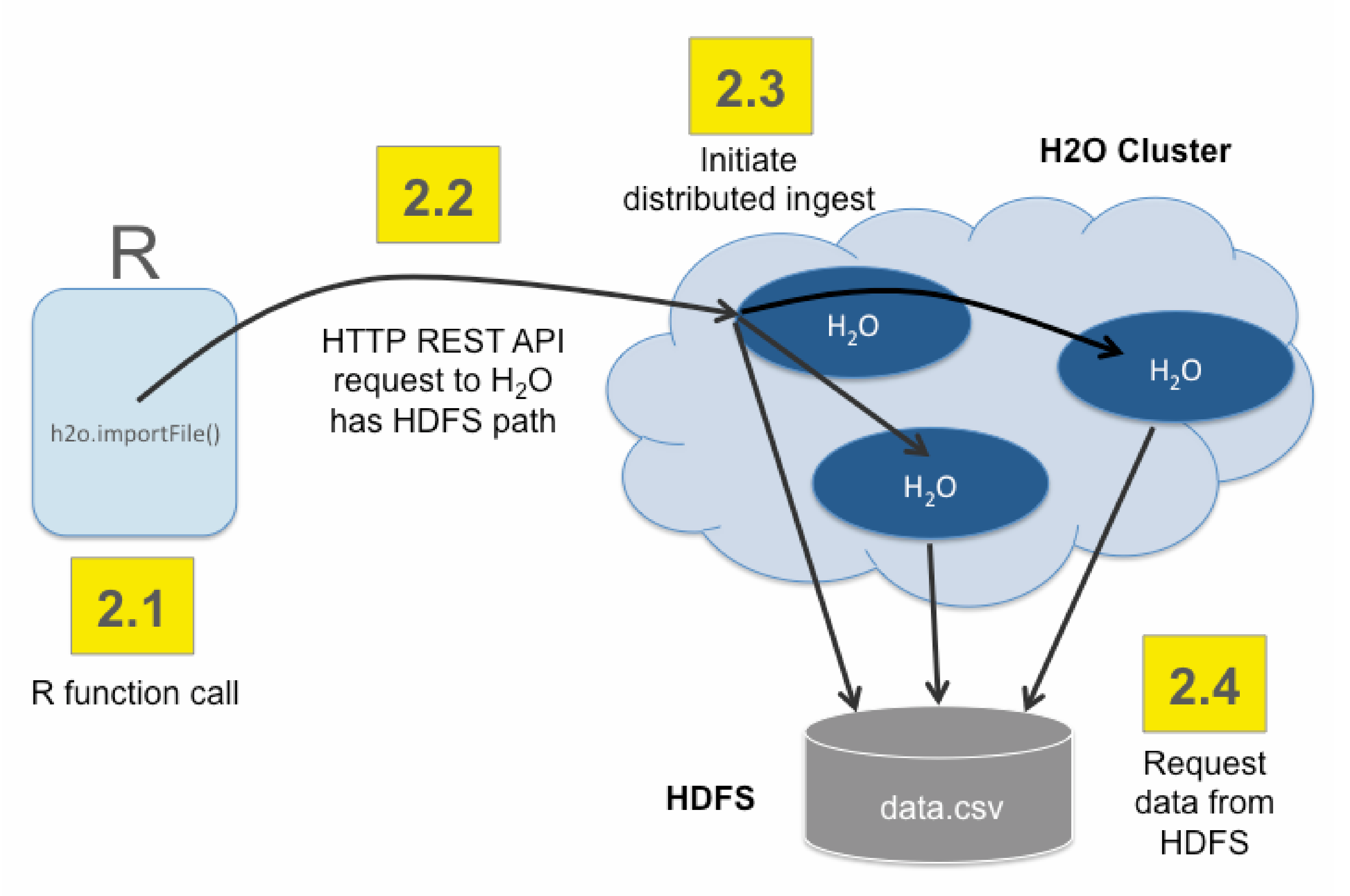
- **Advantages for an R user:**
  - Continue to be an R user and expand the existing workflows
  - Leverage H2O's Machine Learning (distributed random forest, deep learning, gradient boosting etc.) and model benchmarking tools
  - Transparently upscale the analysis with Hadoop, if needed

R - H2O interaction: Step1



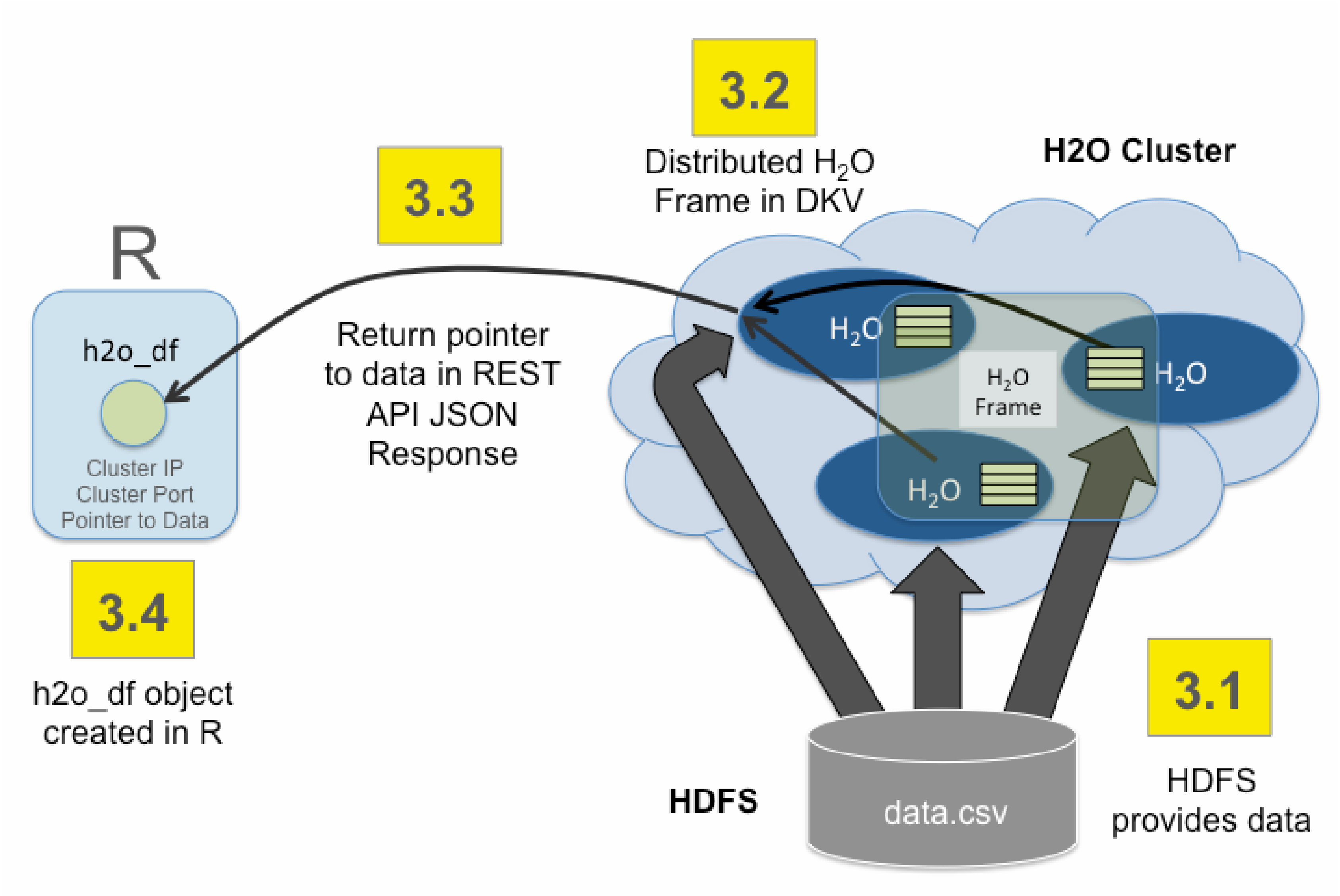
Source: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/architecture.html

R - H2O interaction: Step2



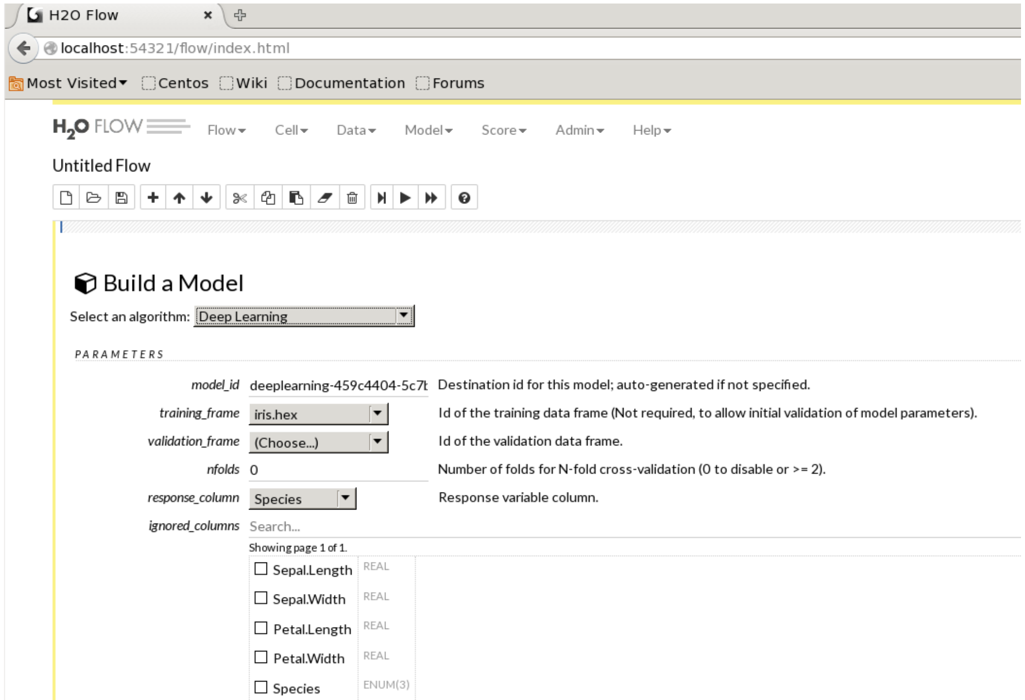
Source: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/architecture.html

R - H2O interaction: Step3



Source: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/architecture.html

Why we were looking at H2O?



(Screenshot of an H2O Flow instance)

- Early (long before the TensorFlow Revolution!) effort to make deep learning user-friendly
- Still, the DL options there are extremely limited, compared to the current mainstream frameworks (next slide)

DL Frameworks Backed by Giants

The Giant	DL framework	Highlights	Higher-level interface
Google (developer)	**TensorFlow** (Python) [C++ backend]	- The fastest-growing framework (TF itself was forked 36,580 times on GitHub) - TensorBoard network visualization - Extensive documentation, community - Slow, harder to distribute the load	**Keras** (Python)
Amazon (user)	**MXNet** (Python, R, Scala, Julia)	- Focus on lightweight and scalability (easy distribution across GPUs etc.) - Fast, memory-efficient - Support for multiple languages, including "the data language of the future" Julia	**Module API** (Python)

P.S. Many past and present frameworks, such as Amazon DSSTNE, Torch, Theano, PyBrain, etc., etc., etc. are not mentioned here.

Keras for R is now also available

```
> library(keras)
> mnist <- dataset_mnist()
Using TensorFlow backend.

> x_train <- mnist$train$x
> y_train <- mnist$train$y
> x_test <- mnist$test$x
> y_test <- mnist$test$y
> # reshape
> dim(x_train) <- c(nrow(x_train), 784)
> dim(x_test) <- c(nrow(x_test), 784)
> # rescale
> x_train <- x_train / 255
> x_test <- x_test / 255
>
> y_train <- to_categorical(y_train, 10)
> y_test <- to_categorical(y_test, 10)
> model <- keras_model_sequential()
> model %>%
+   layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
+   layer_dropout(rate = 0.4) %>%
+   layer_dense(units = 128, activation = 'relu') %>%
+   layer_dropout(rate = 0.3) %>%
+   layer_dense(units = 10, activation = 'softmax')
```

Conclusions

- Computational biology started to actively leverage the modern AI advances by adopting all the architecture types of the neural networks to solve problems in its domain
- Development tools for neural network construction and training are becoming increasingly accessible to work at higher coding level