

# Deep Generative Networks in Single Cell Transcriptomics

Panagiotis Papasaikas

FMI Computational Biology



Friedrich Miescher Institute  
for Biomedical Research



Advanced scRNA course, May 2020



# Overview

1st break

- Introduction to Tensorflow & Keras
- Introduction to Deep Learning
- 1st Exercise (MLP in Keras)
- DGNs
  - VAEs
  - GANs
- Applications and Existing Tools
- 2nd break • 2nd Exercise (SC Variational Inference in R Keras)

Parts of the introduction heavily inspired by J.J. Allaire's keynote at rstudio::conf 2018  
and Franchoit Chollet's "**Deep Learning with R**"

# What is Tensorflow



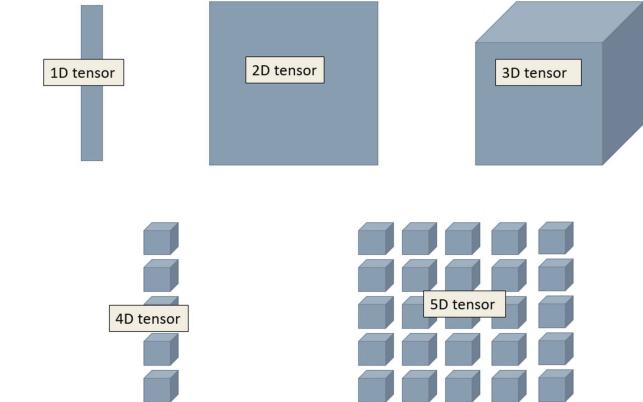
- TF is an open source general purpose numerical computing library (not only DL, e.g general optimization libraries).
- Originally developed by engineers in the Google Brain Team for conducting ML research
- Hardware independent (CPUs, GPUs, TPUs)
- Supports large datasets/distributed execution

# The model building blocks in Tensorflow/Keras

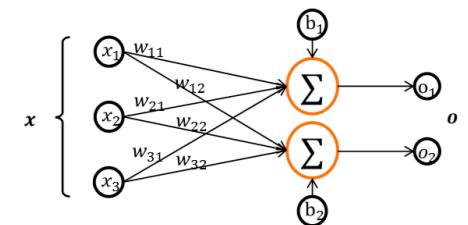
- Tensors are multidimensional arrays.

Data	Tensor dimension	R object
Cell label	1D (samples)	vector
Gene Count Matrix	2D (samples, genes)	matrix
Longitudinal data	3D (samples, genes, timestamp)	3d array
Microscopy Images	4D (samples, height, width, channels)	4d array
Video	5D (sample, height, width, channel, frame)	5d array

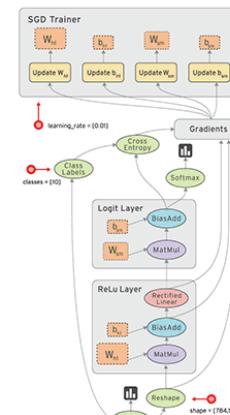
\*Notice the orientation convention is opposite to what bioinformaticians / R users are used to



- Layers are units of numerical computations (transformation functions) applied on tensors and **parameterized by weights**.  
e.g addition, matrix multiplication, sampling, taking gradients...



- Layers and Tensors are combined to construct computation graphs (DAGs).  
Nodes are layers (computations), edges are Tensors.  
Tensors “flow” through the computation graph and do smth useful (?).  
A fully specified graph from input to output is a Model.



TensorFlow graph CC  
by [Tensorflow.org](https://Tensorflow.org)

# Keras



- Keras is a high level API that provides convenient wrappers for commonly used layers or computation graphs

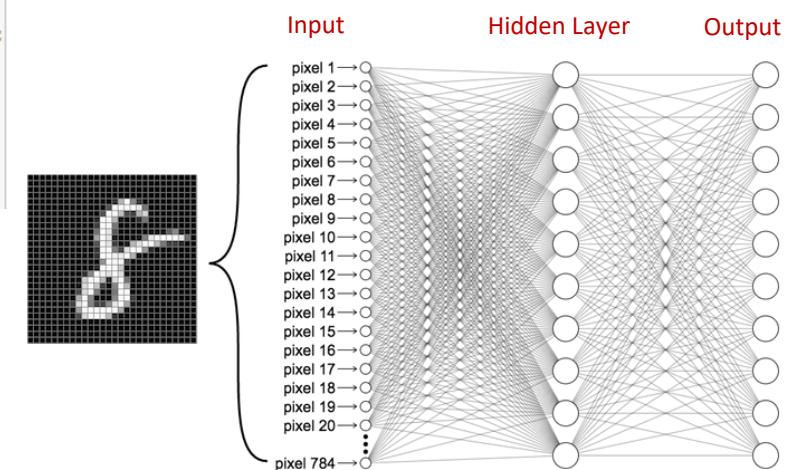
```
#defining a keras sequential model
model <- keras_model_sequential()

#defining the model with 1 input layer[784 neurons], 1 hidden layer[784 neurons] with dropout rate 0.4 and 1 output
#i.e digits from 0 to 9
model %>%
layer_dense(units = 784, input_shape = 784, activation = 'relu') %>%
layer_dropout(rate=0.4) %>%
layer_dense(units = 10,activation = 'softmax')

#defining model with one input layer[784 neurons], 1 hidden layer[784 neurons] with dropout rate 0.4 and 1 output 1.
model<-Sequential()

from keras.layers import Dense
model.add(Dense(784, input_dim=784, activation='relu'))
keras.layers.core.Dropout(rate=0.4)

model.add(Dense(10,input_dim=784,activation='softmax'))
```



MLP model for digit classification



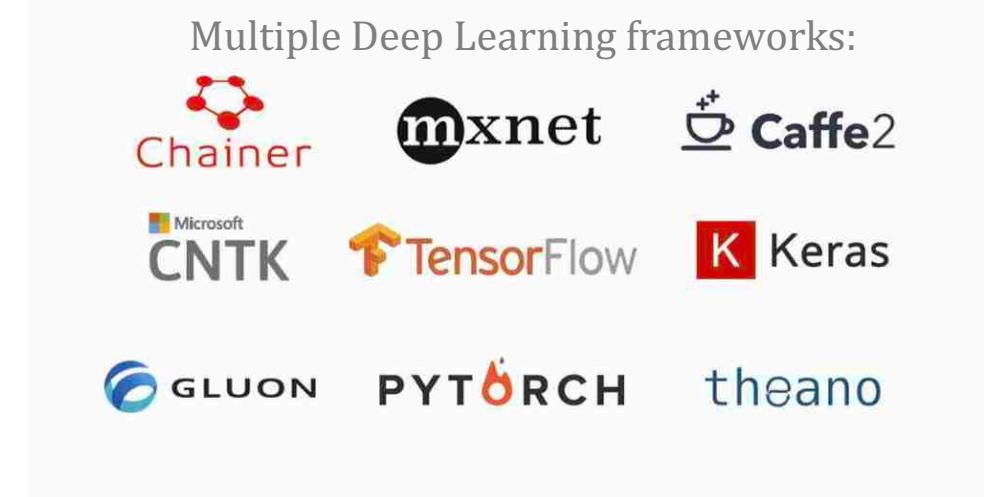
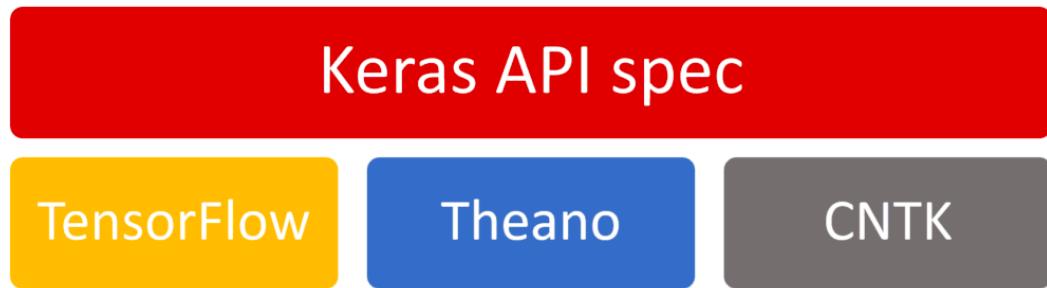
relu activation

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

softmax activation

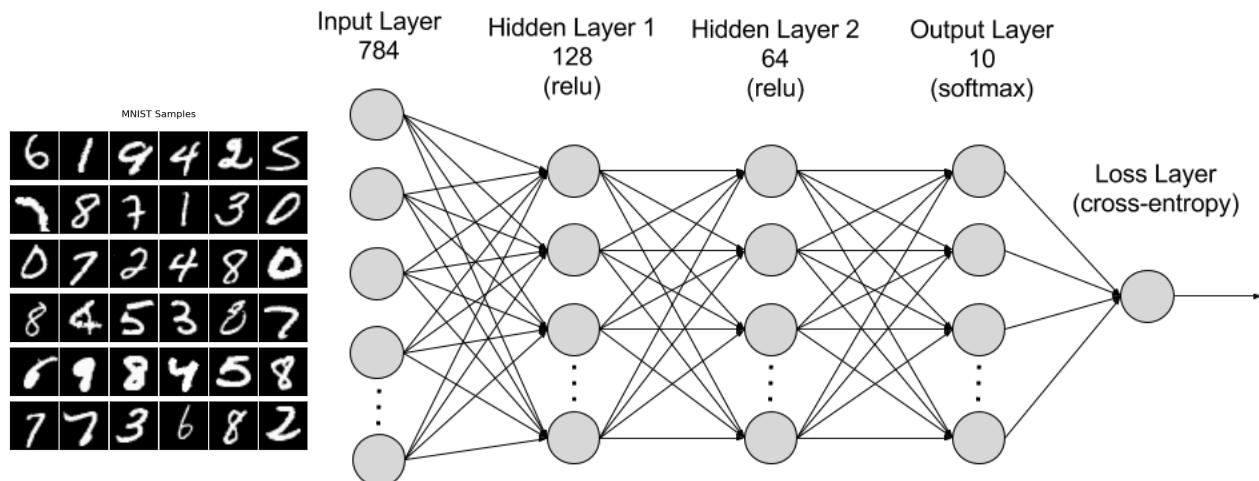
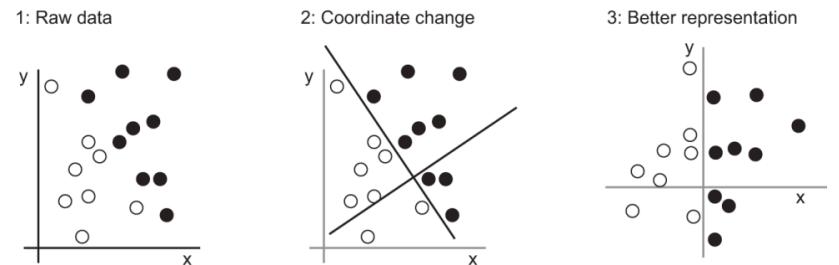
# Life beyond Tensorflow and Keras

Keras as a high level API supports multiple DL backends:



# What is Deep Learning

Deep Learning Models take an input and transform it to an output via successive layers of increasingly abstract and meaningful **representations**



Raw data

Extraneous information  
filtered, useful information extracted

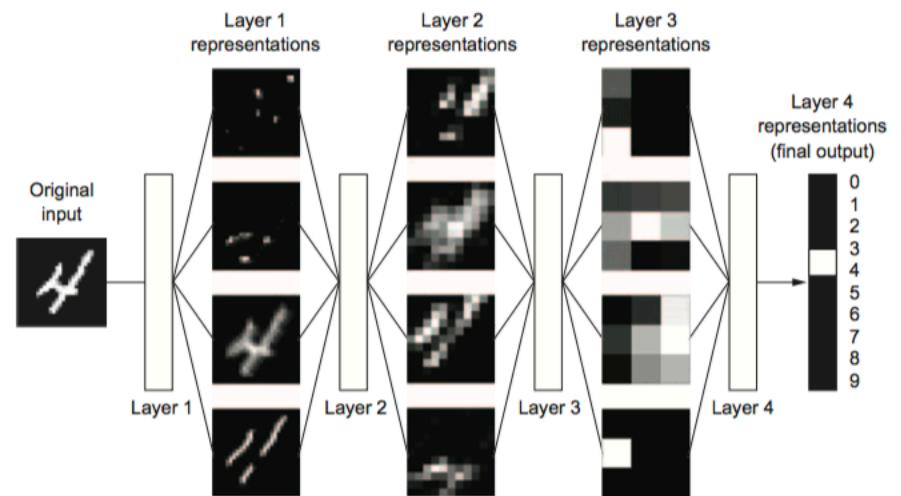


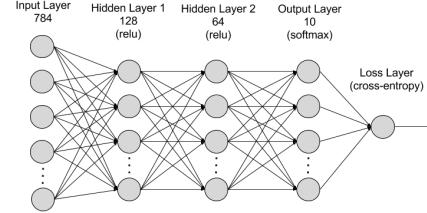
Figure 1.6 Deep representations learned by a digit-classification model

Image from F. Chollet's "Deep Learning with R"

!!! What is a “meaningful representation” is a relative concept that depends on the task at hand

Why Deep? -> Multi Layered Representation

# The mechanics of model training



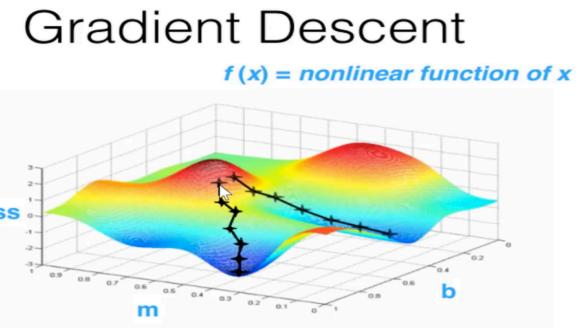
The **loss function** measures the success of the model for the task at hand.

The parameters (weights) of the model are updated towards a direction that provides an improvement

Updates are done using the **backpropagation** algorithm and the **chain rule** that traverses the model from the output towards the input

The direction towards which the parameters need to move is computed using **Stochastic Gradient Descent** variants

This loop is repeated many times using small splits of the data (batches)(epochs) until convergence



## What spurred the revolution?

Mainly advances on three fronts:

- Massively parallel computation hardware (GPUs, TPUs)
- Improved algorithms  
robust backprop, optimizers, regularization techniques
- High-quality (often labeled) datasets  
web usage, advances in tech/instrumentation in hard sciences



Improved architectures

User-friendly platforms

## Successes of Deep Learning

- Refined web-searching
- Spam/Fraud detection
- Near-human image classification (**MSRA**)
- Near-human machine translation (**DeepL**)
- Superhuman chess/GO playing (**AlphaZero, LC0**)
- Autonomous driving
- Natural language processing (e.g **IBM debater, GPT-2**)
- Protein Folding
- Medical Image Processing
- Drug design
- Diagnostics

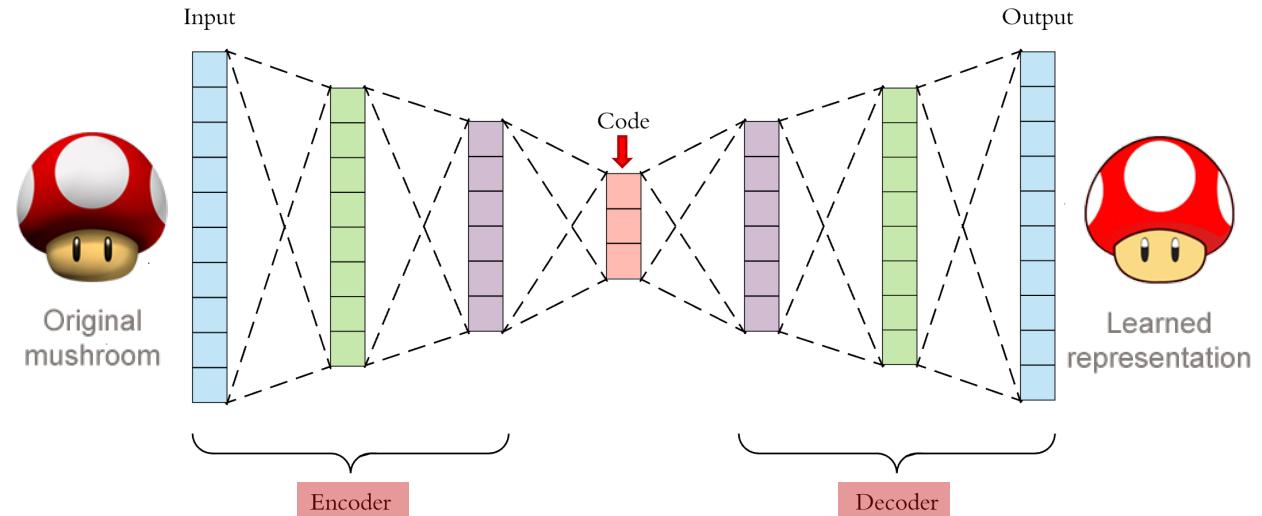
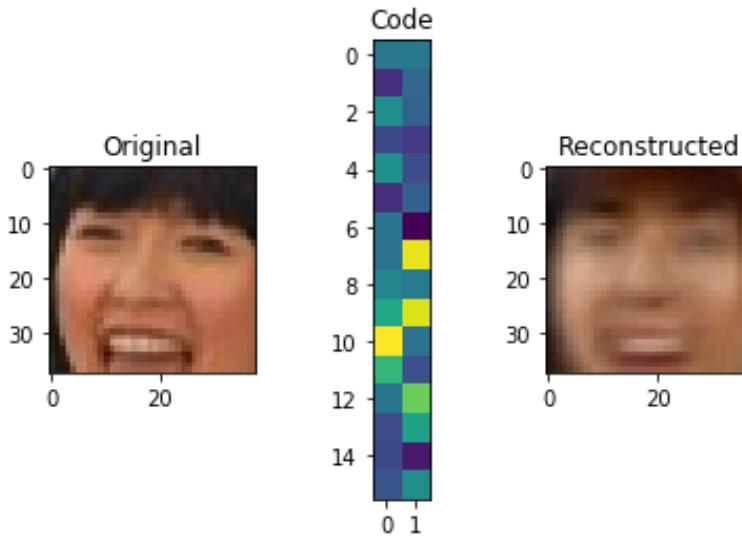
# 1st Exercise

## **What we will NOT talk about:**

- Distributed multi-GPU training
- Regularization techniques
- Custom layer construction with TF
- Batch Normalization
- Eager vs deferred execution
- CNNs
- Geometric Deep Learning
- Recurrent Layers (e.g LSTMs)
- Attention Models
- Reinforcement learning

# Autoencoders: architecture and latent codes

- Unsupervised (easy access to large training sets)
- Objective is to obtain an output that matches the input.
- Data are “squeezed” through successive layers of decreasing dimensions
- The middle hidden layer is a **code** (latent code) that **represents** the input:

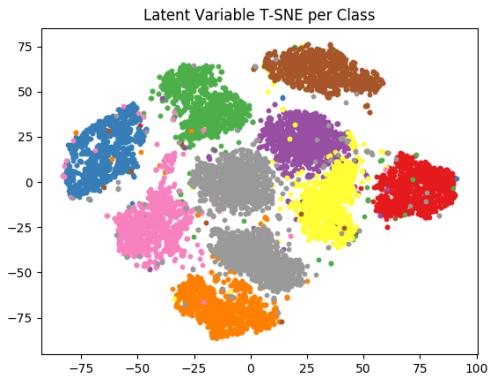


**Multiple AE flavors**  
Deep/Stacked, Sparse, Variational, Denoising,  
Adversarial, Disentangled...

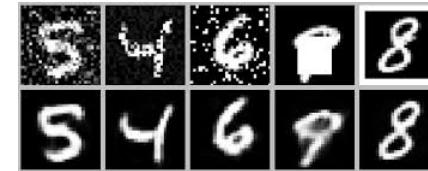
# Applications of AEs

## 1. Dimensionality reduction & visualization

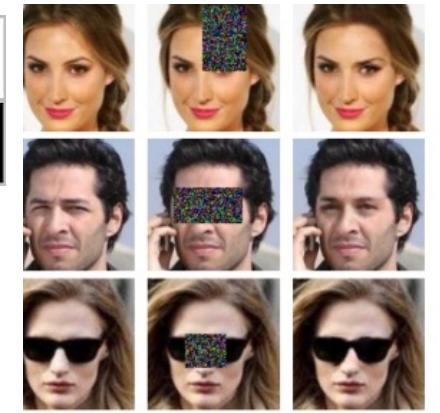
0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2 2 2 2 2 2 2 2  
3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4  
5 5 5 5 5 5 5 5 5 5 5 5 5  
6 6 6 6 6 6 6 6 6 6 6 6 6  
7 7 7 7 7 7 7 7 7 7 7 7 7  
8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9



## 2. Denoising & completion (imputation)

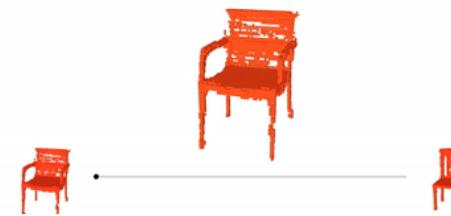
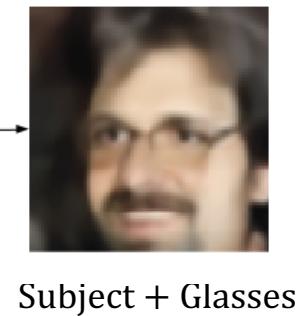
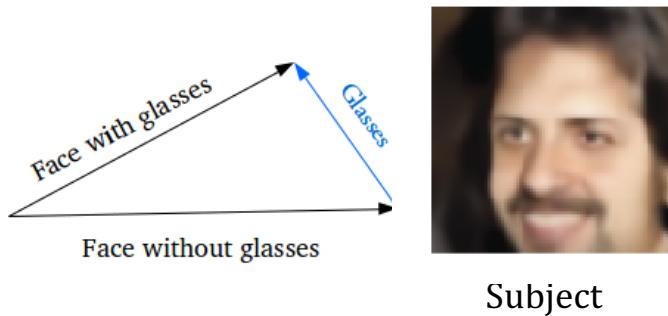


Digit Denoising



Face completion

## 3. Feature manipulation , interpolation and exploration



## Why AEs for SC transcriptomics?

Tx data: High dimensional    Noisy/corrupt →  
→ Visualization    Denoising

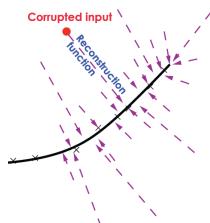
# AE extensions and “good” representation codes

## Multiple AE flavors

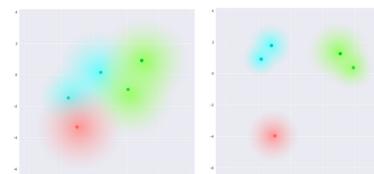
Deep/Stacked, Sparse, Variational, Denoising,  
Adversarial, Disentangling...

The common goal it to obtain **a good code representation** of the input data

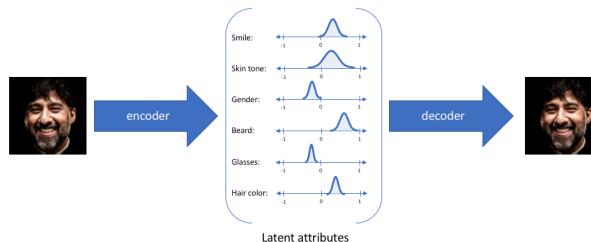
- Robust to “meaningless” input corruptions
- Generalizable  $\Rightarrow$  can transfer to multiple settings /related problems



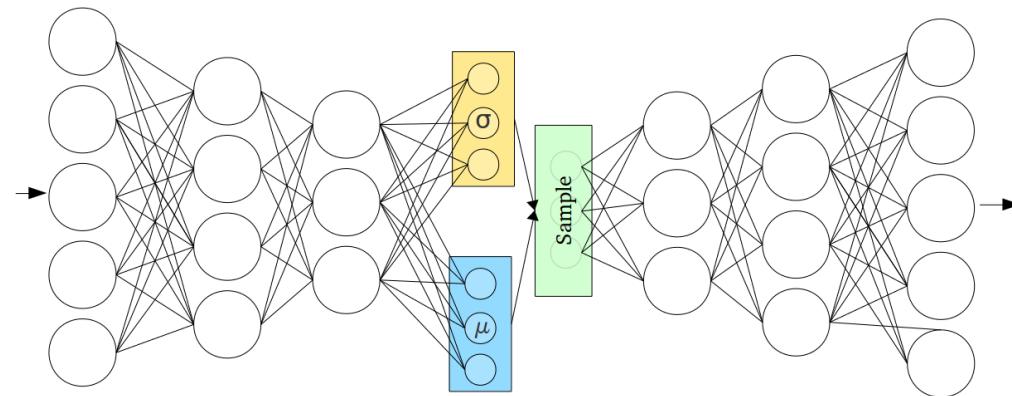
- Smooth / Coherent: similar inputs  $\mapsto$  similar codes.



- Explanatory



## Variational Autoencoders



- VAEs generalize AEs adding stochasticity
- Encourage a continuous latent manifold
- Robustness + valid decoding
- Allows interpolation and exploration

D. P. Kingma and M. Welling. "Auto-encoding variational Bayes". arXiv:1312.6114, 2013.

$$\mathcal{L}_\beta = \frac{1}{N} \sum_{n=1}^N (\mathbb{E}_q[\log p(x_n|z)] - \beta D_{KL}(q(z|x_n)||p(z)))$$

---

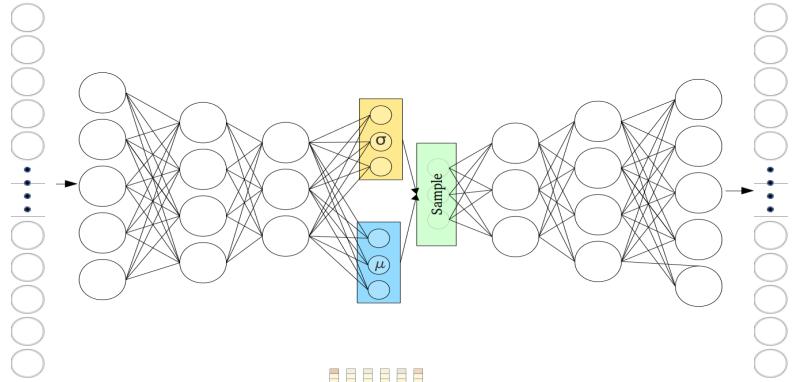
Reconstruction                          Distance to latent prior

The latent prior is multivariate normal with a unit covariance matrix

- $\beta = 1$  : ELBO (Evidence Lower Bound, standard VAE)
- $\beta < 1$  : Partially regularized VAE (Liang et al. 2018)
- $\beta > 1$  : Disentangling Autoencoders ( $\beta$ -VAE, Higgins et al. 2017)

# VAEs in single cell data

Gene Space

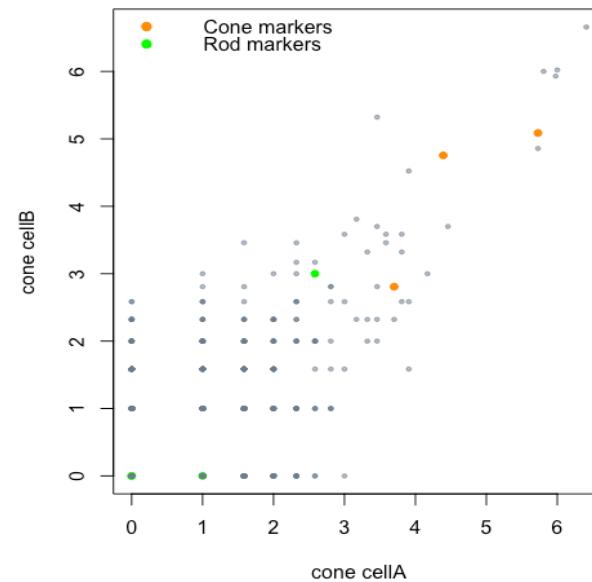


Gene Space

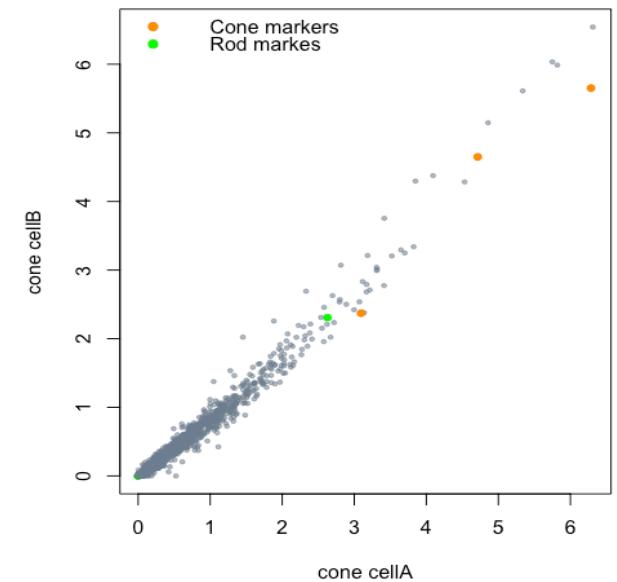
Latent Encoding

Dimensionality reduction and denoising are natural applications

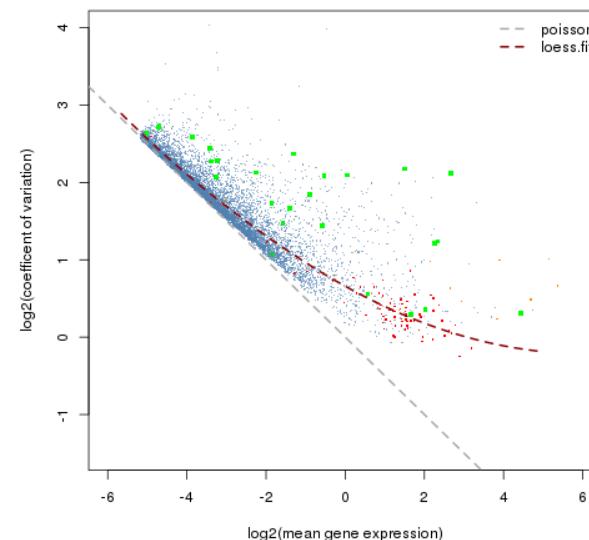
Observed



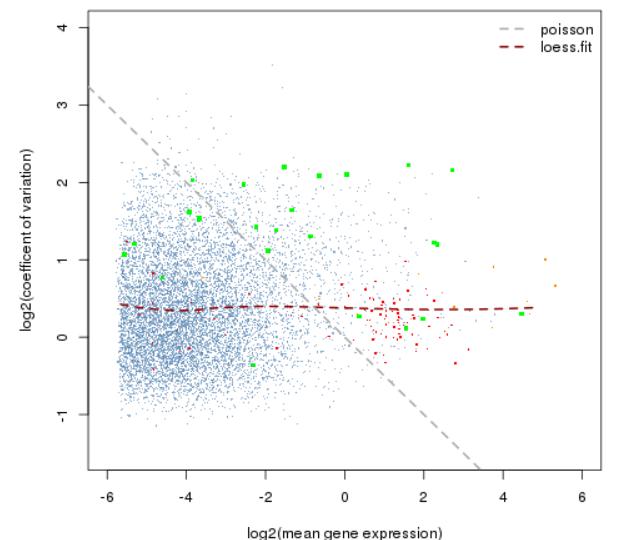
Denoised



mean-variance trend actual



mean-variance trend denoised



## Single-cell RNA-seq denoising using a deep count autoencoder

Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller & Fabian J. Theis

## Exploring single-cell data with deep multitasking neural networks

Matthew Amodio, David van Dijk, Krishnan Srinivasan, William S. Chen, Hussein Mohsen, Kevin R. Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, Anita Desai, V. Ravi, Priti Kumar, Ruth Montgomery, Guy Wolf & Smita Krishnaswamy 

*Nature Methods* **16**, 1139–1145(2019) | [Cite this article](#)

## Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P. Reilly, Gang Hu  & Mingyao Li 

*Nature Communications* **11**, Article number: 2338 (2020) | [Cite this article](#)

## Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P. Reilly, Gang Hu  & Mingyao Li 

*Nature Communications* **11**, Article number: 2338 (2020) | [Cite this article](#)

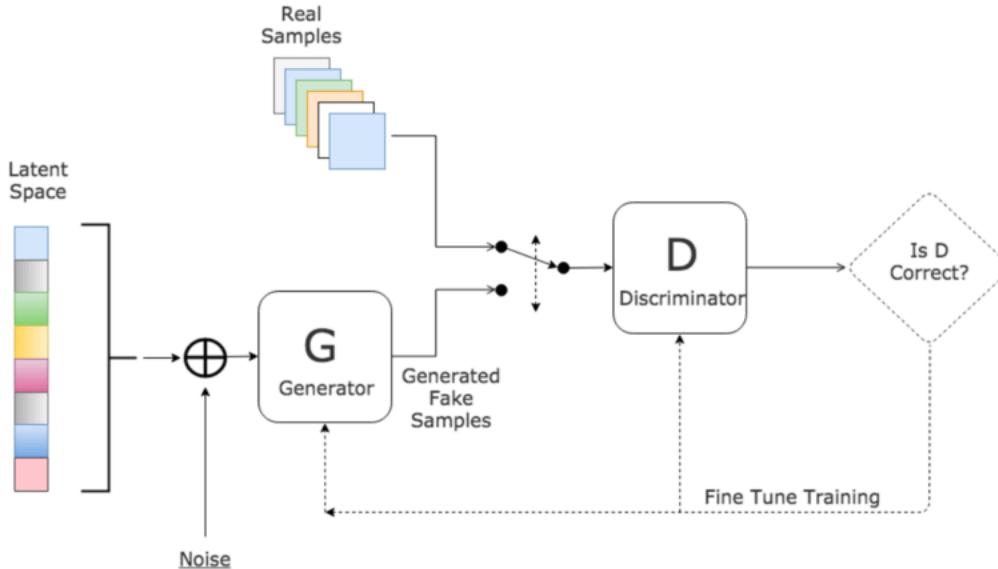
## Interpretable dimensionality reduction of single cell transcriptome data with deep generative models

Jiarui Ding , Anne Condon & Sohrab P. Shah 

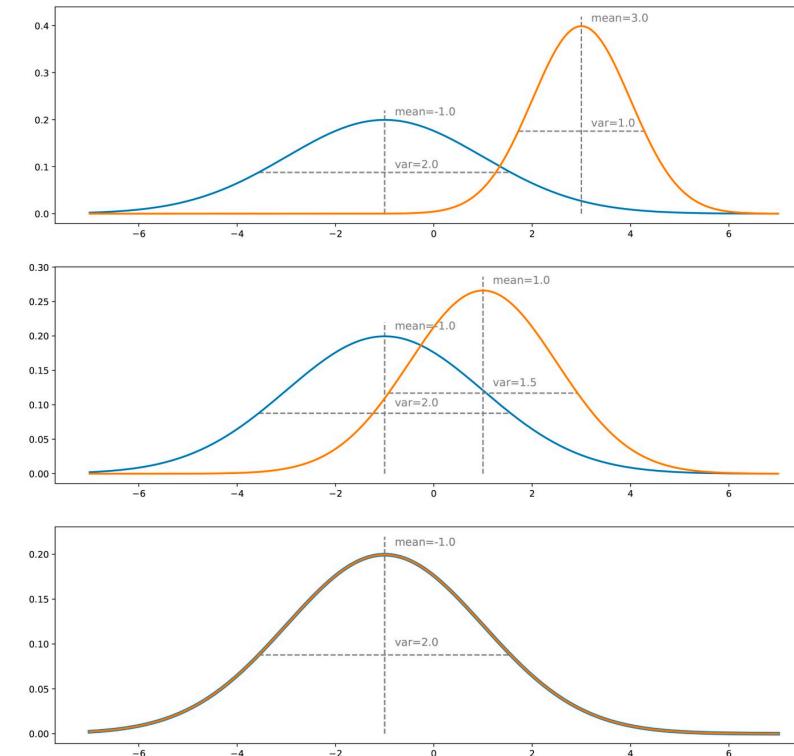
*Nature Communications* **9**, Article number: 2002 (2018) | [Cite this article](#)

# Generative Adversarial Networks (GANs)

## Generative Adversarial Network



I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.'Generative adversarial nets '. In Advances in neural information processing systems,2672-2680, 2014.



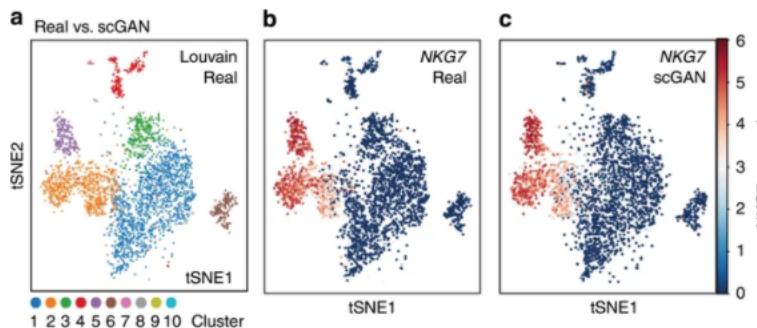
GANs have notoriously unstable training dynamics and suffer from what is known as “**mode collapse**”, which leads to some modes of the data being overrepresented and other missing.

However, they are able to generate highly realistic “fake” samples

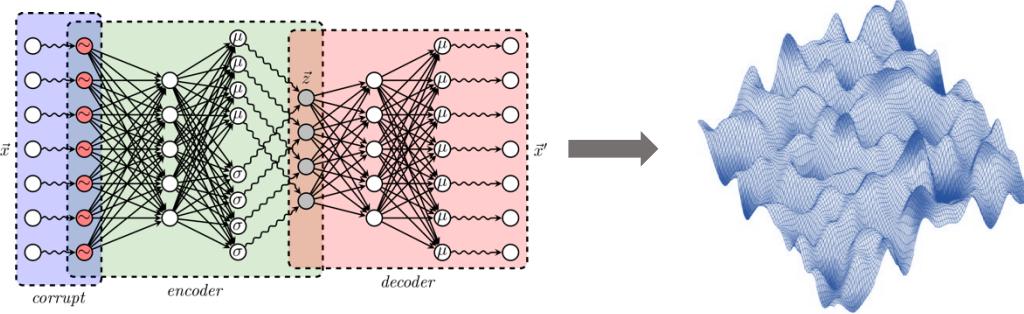
# Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks

Mohamed Marouf, Pierre Machart, Vikas Bansal, Christoph Kilian, Daniel S. Magruder, Christian F. Krebs & Stefan Bonn 

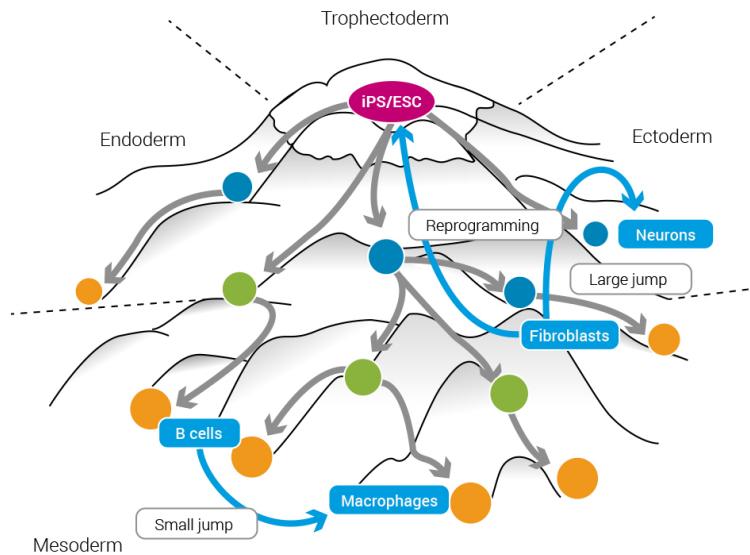
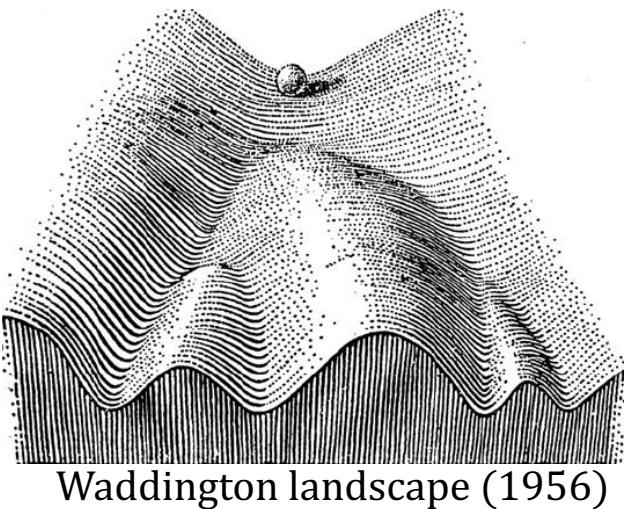
*Nature Communications* 11, Article number: 166 (2020) | [Cite this article](#)



The latent representation is an estimation of the underlying manifold that gives rise to the data

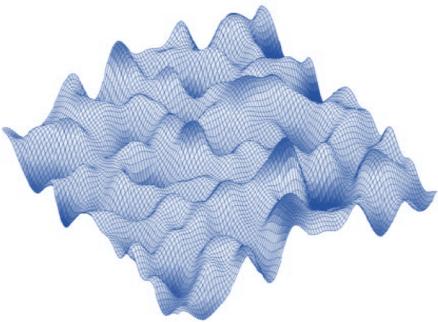


succinct, generative representations of complex Tx manifolds



Differentiation / trans-differentiation  
Pluripotency / Reprogramming  
Immunological maturation  
Cell ageing  
Cancer progression  
Response to perturbations

# DGN-based Inference on SC data



DGN based inference allows inspection of regions of the Tx landscape that have not been visited

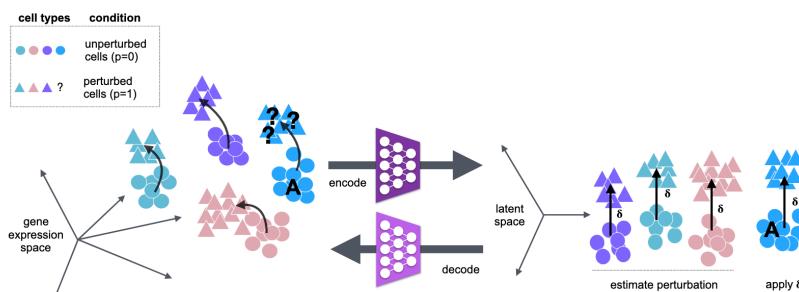
Some examples:

- Inferring transcriptomes upon biological perturbations (e.g in Silico KDS)
- Inferring effects of perturbations in different cell/tissue contexts (out-of-sample prediction)
- Inferring trajectories

## scGen predicts single-cell perturbation responses

Mohammad Lotfollahi, F. Alexander Wolf & Fabian J. Theis

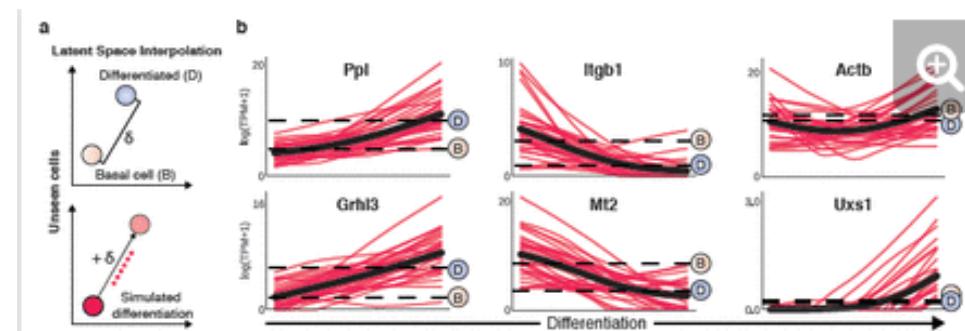
Nature Methods 16, 715–721(2019) | Cite this article



## Generative adversarial networks uncover epidermal regulators and predict single cell perturbations

Arsham Ghahramani, Fiona M. Watt, Nicholas M. Luscombe

doi: <https://doi.org/10.1101/262501>



## 2nd Exercise