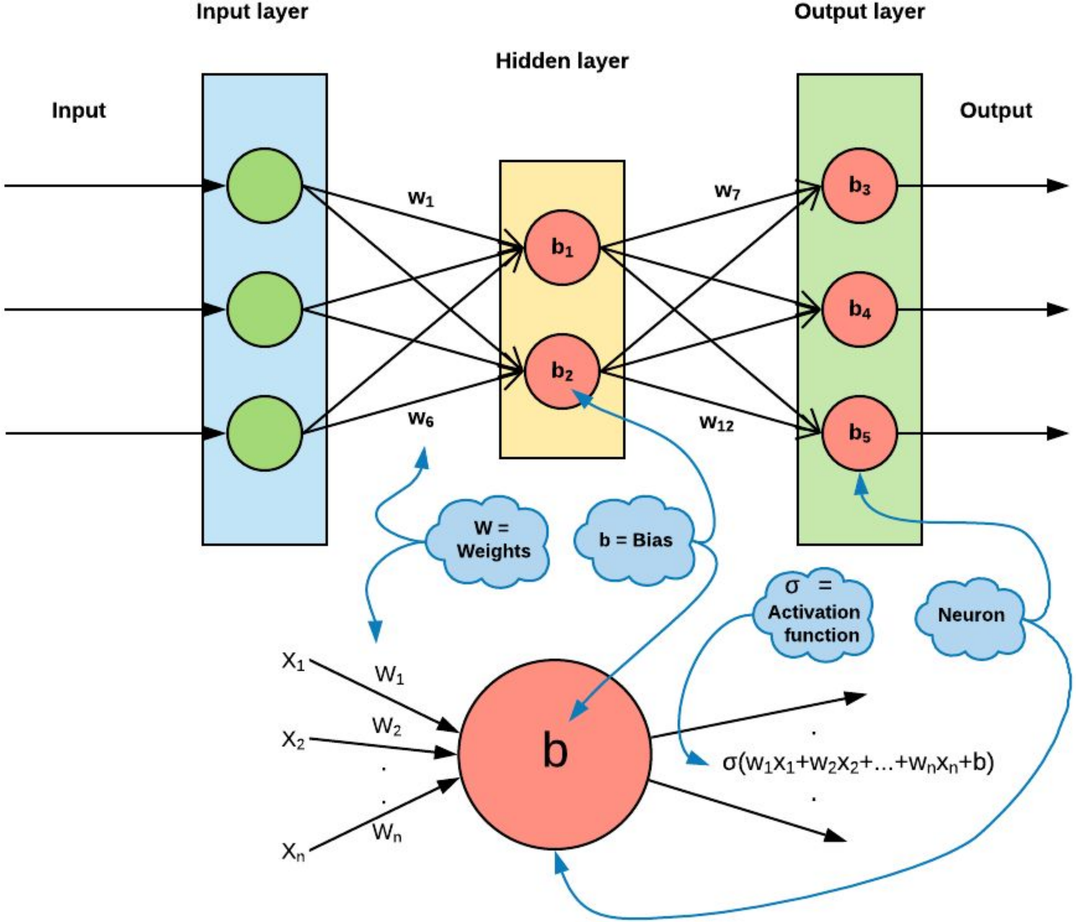
Attractors of autoencoders

- Memorization in neural networks

Jonas Strandqvist

Autoencoder neural networks

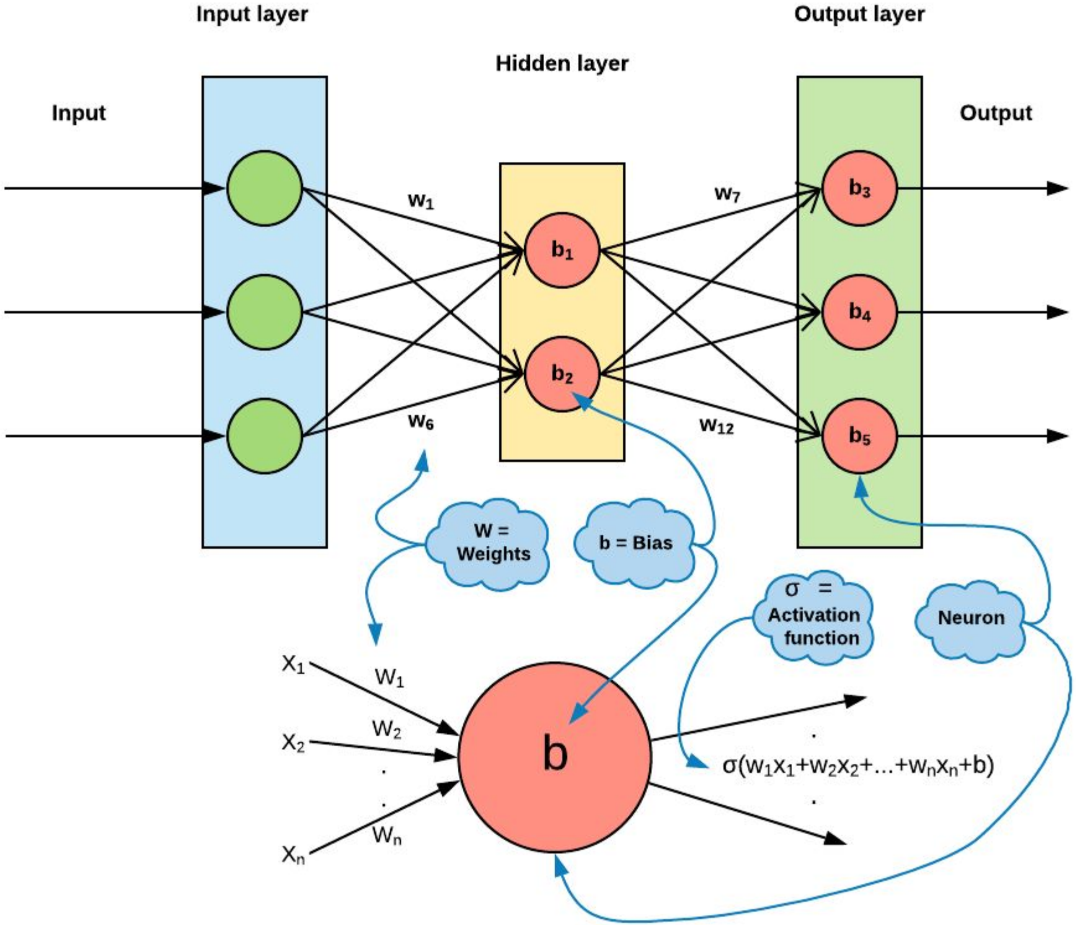
- Input & output layers: same number of neurons



Autoencoder neural networks

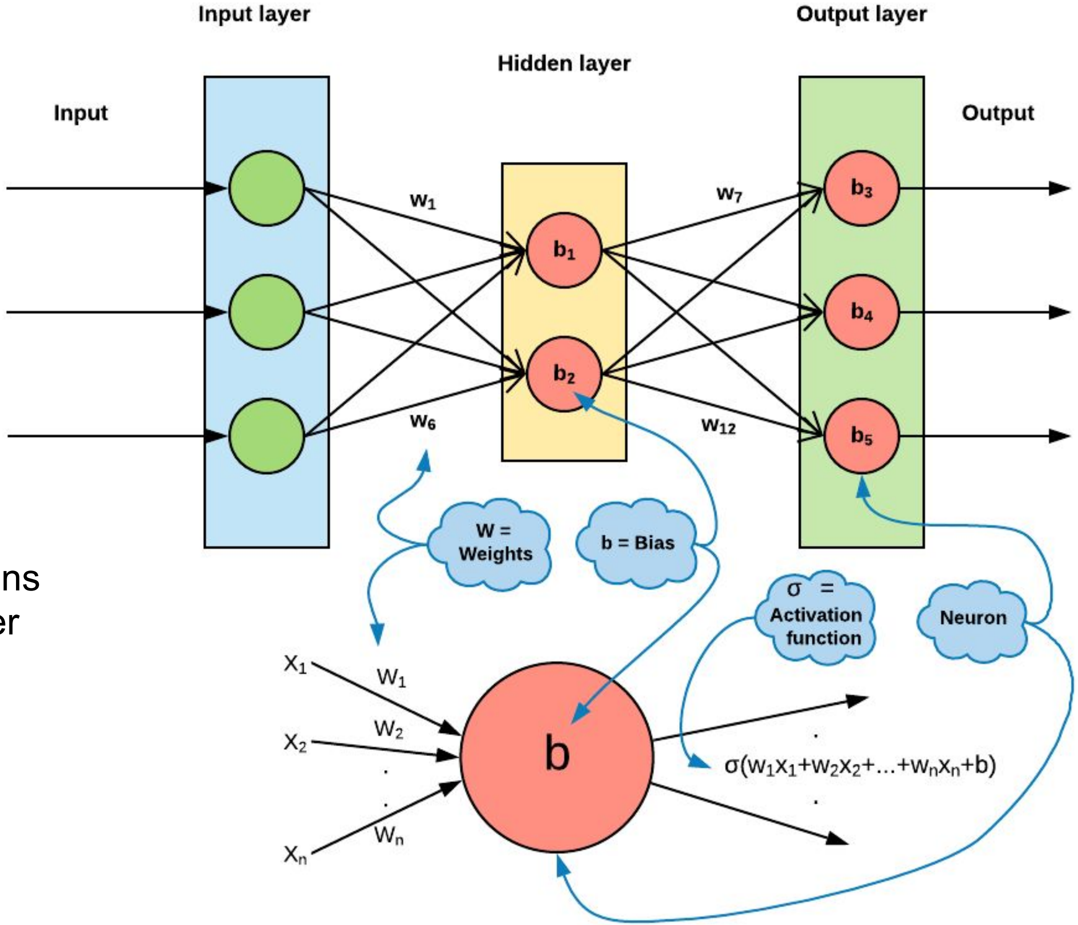
- Input & output layers:
  same number of neurons

- **Depth** = number of
  hidden layers in neural network
  - Example: depth = 1



Autoencoder neural networks

- Input & output layers:
  same number of neurons
- **Depth** = number of
  hidden layers in neural network
  - Example: depth = 1
- In this talk:
  - Every hidden layer has
    the same number of neurons
  - **Width** refers to this number
  - Example: width = 2



Machine learning with autoencoders

Training is the process that improves weights and biases such that our neural networks produces better results.

Goal of an autoencoder:

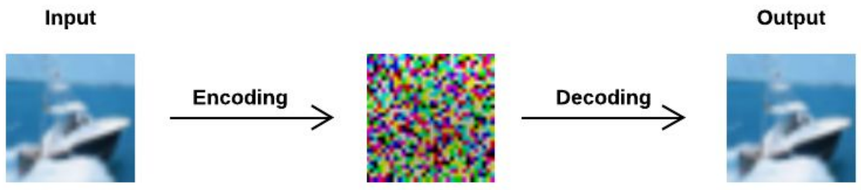
Machine learning with autoencoders

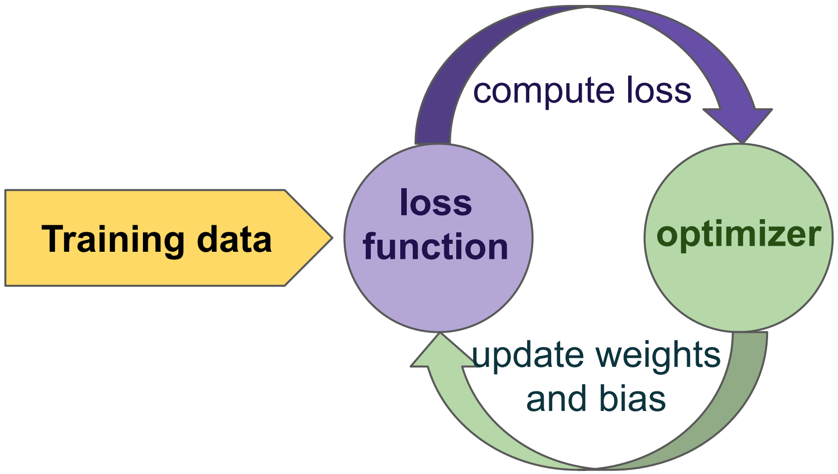
Training is the process that improves weights and biases such that our neural networks produces better results.

Requires three ingredients:

- a set of training data
- a loss function: measures how well the autoencoder achieves its task
- an optimizer: changes weights and biases to improve loss

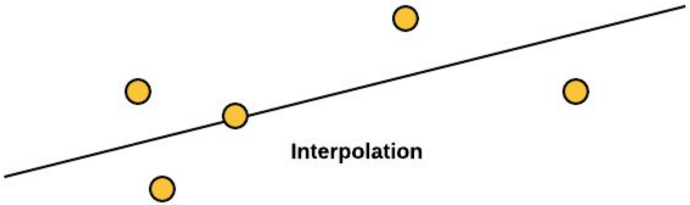
Goal of an autoencoder:



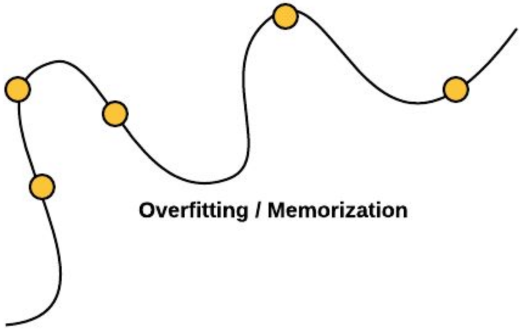


Memorization in autoencoders

Few parameters lead to interpolation.



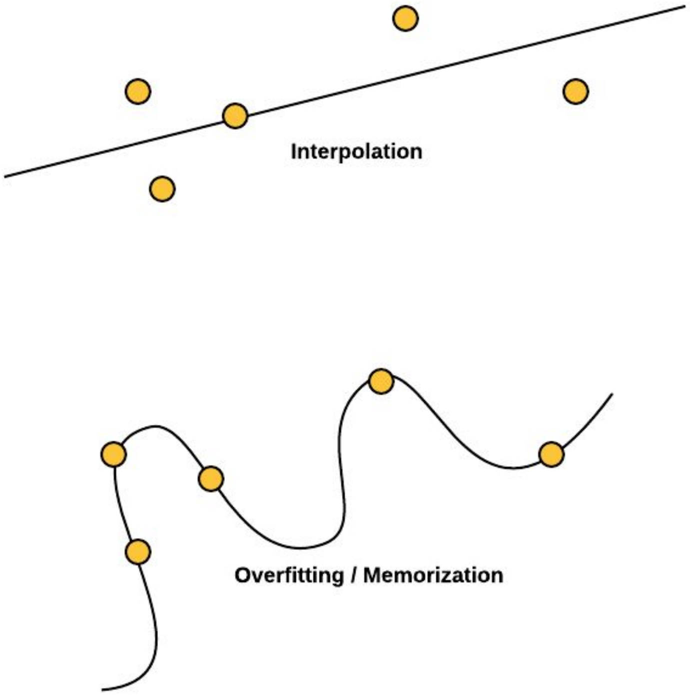
Too many parameters allow the network to learn the dataset.

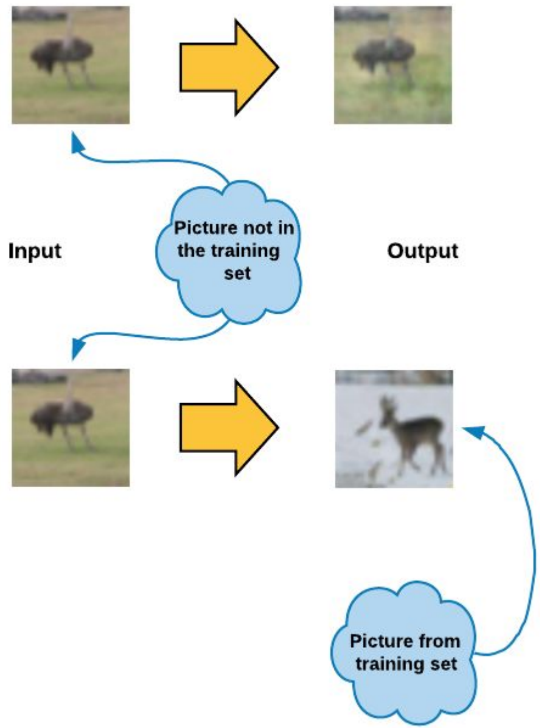


Memorization in autoencoders

Few parameters lead to interpolation.

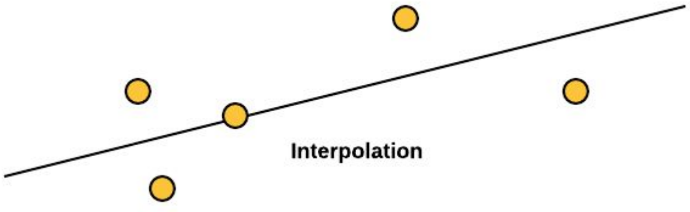
Too many parameters allow the network to learn the dataset.

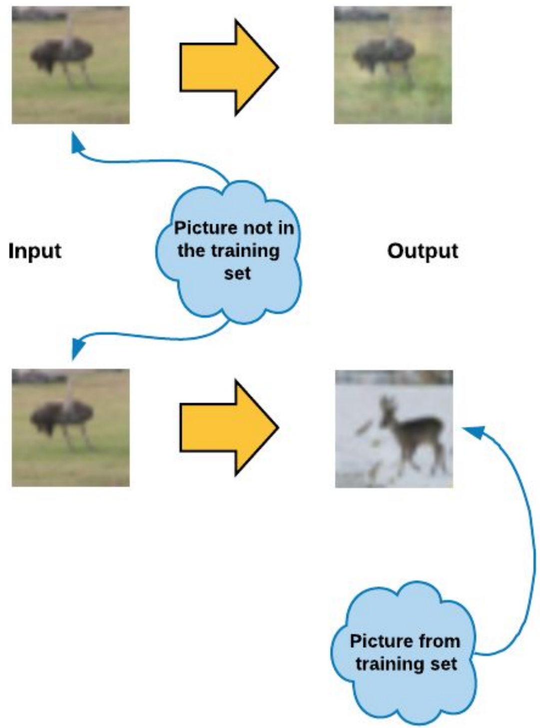




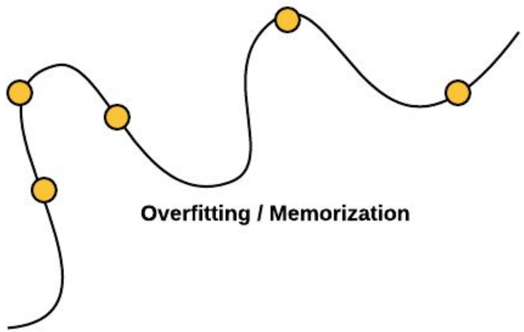
Memorization in autoencoders

Few parameters lead to interpolation.





Too many parameters allow the network to learn the dataset.

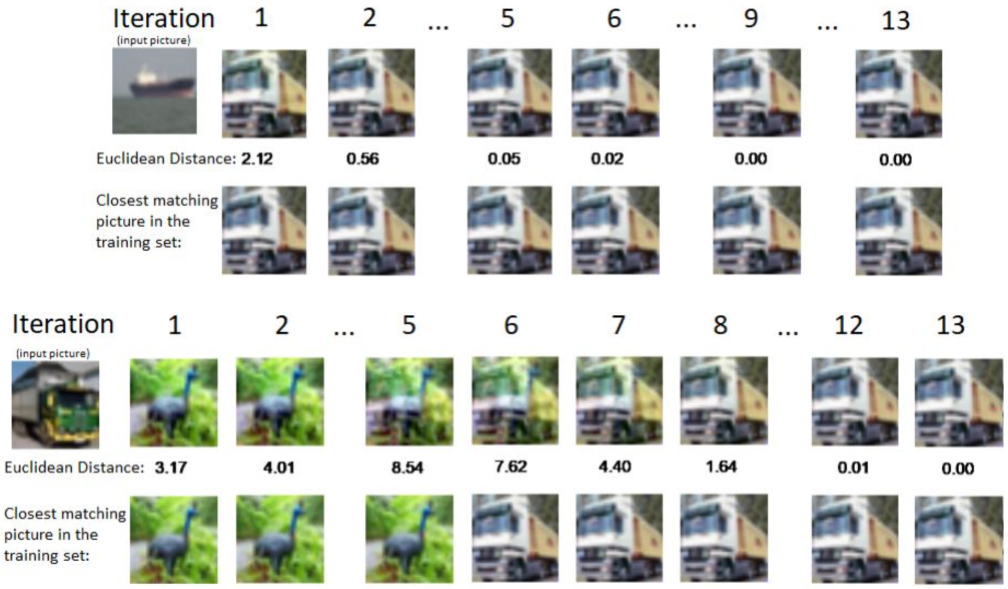


It is still an open question how neural networks memorize data.
For this, articles [1] and [2] suggest to study **attractors** of autoencoders.

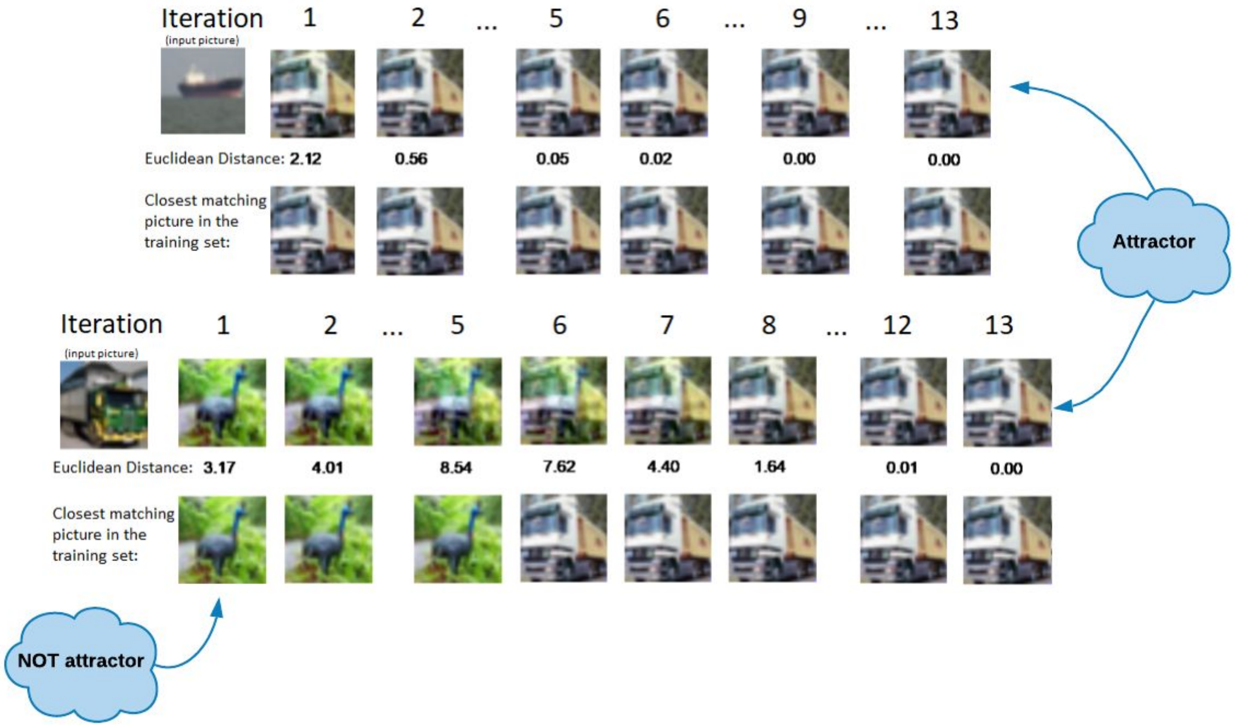
[1] “Memorization in overparameterized autoencoders” - A. Radhakrishnan, K.D. Yang, M. Belkin and C. Uhler
[2] “Overparameterized Neural Networks Can Implement Associative Memory” - A. Radhakrishnan, M. Belkin, C. Uhler

Attractors of autoencoders

Attractors of autoencoders



Attractors of autoencoders



Attractors of autoencoders

When is a training image an attractor?

Two things can go wrong:

- becomes another image when iterating autoencoder

  → we say that is not an **iterative fixed point**

Attractors of autoencoders

When is a training image an attractor?

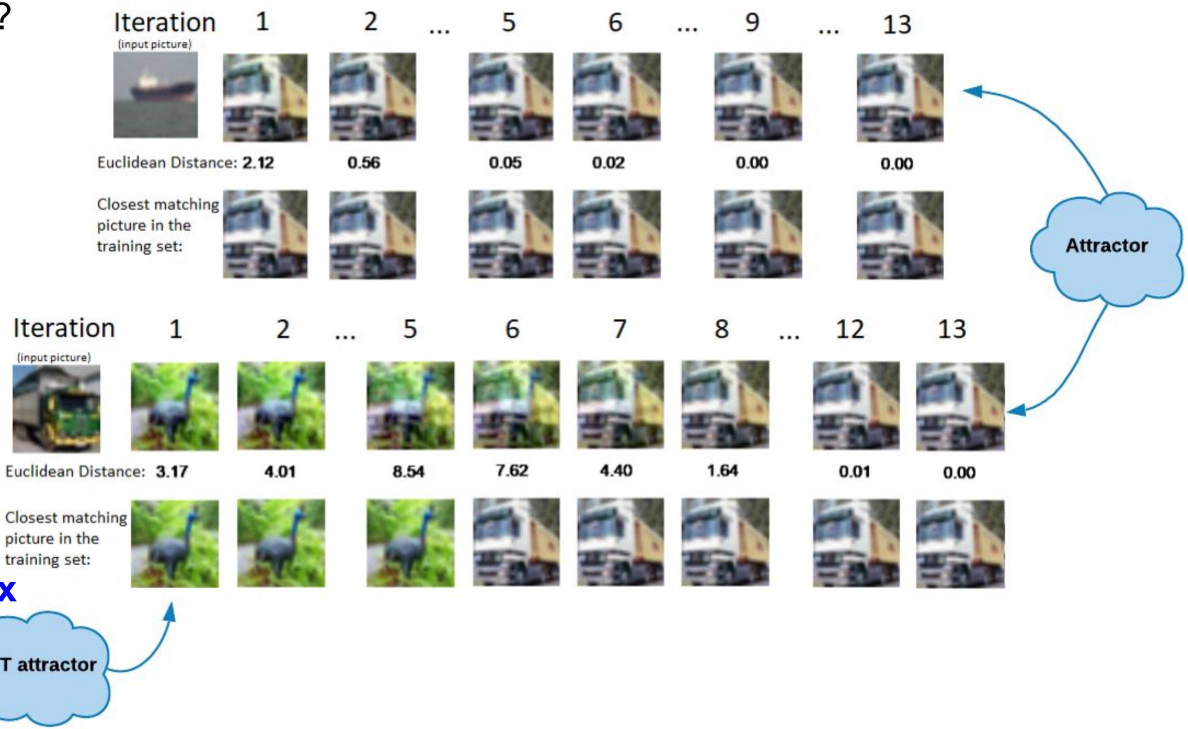
Two things can go wrong:

- becomes another image when iterating autoencoder

  → we say that is not an **iterative fixed point**

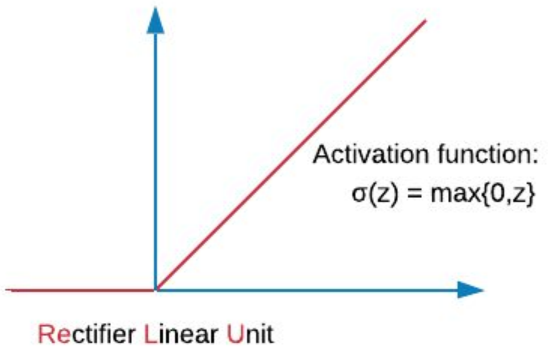
- **other** images never become

(can be checked with the **eigenvalues of the Jacobian matrix** of the autoencoder at)



Problem formulation

We want to extend the experiments in [2] to **ReLU** autoencoders.



[2] “Overparameterized Neural Networks Can Implement Associative Memory” - A. Radhakrishnan, M. Belkin, C. Uhler

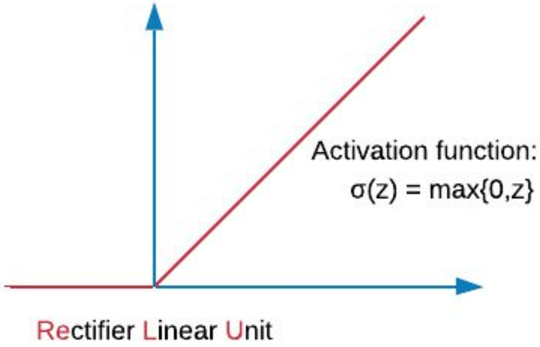
Problem formulation

We want to extend the experiments in [2] to **ReLU** autoencoders.

We investigate the impact on the amount of attractors:

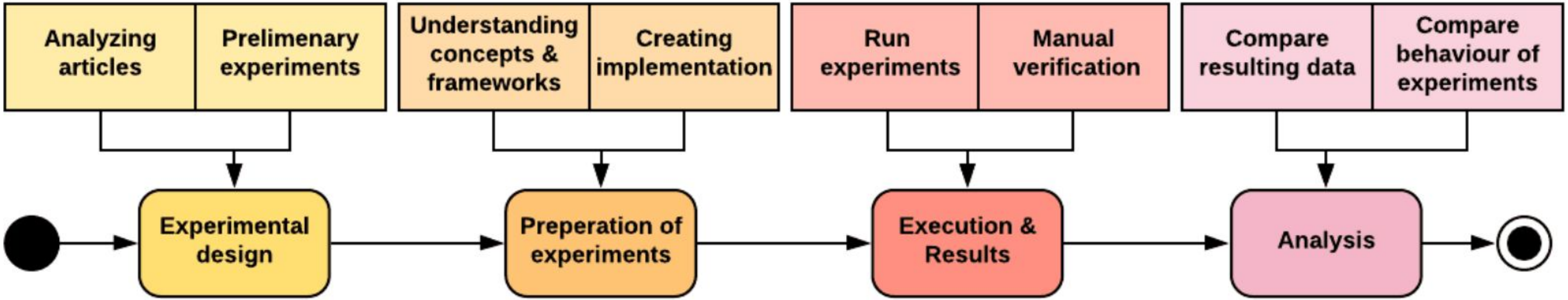
- changing **depth** and **width**
- with and without **bias**
- using **structured** and **random pictures**

From CIFAR10 dataset

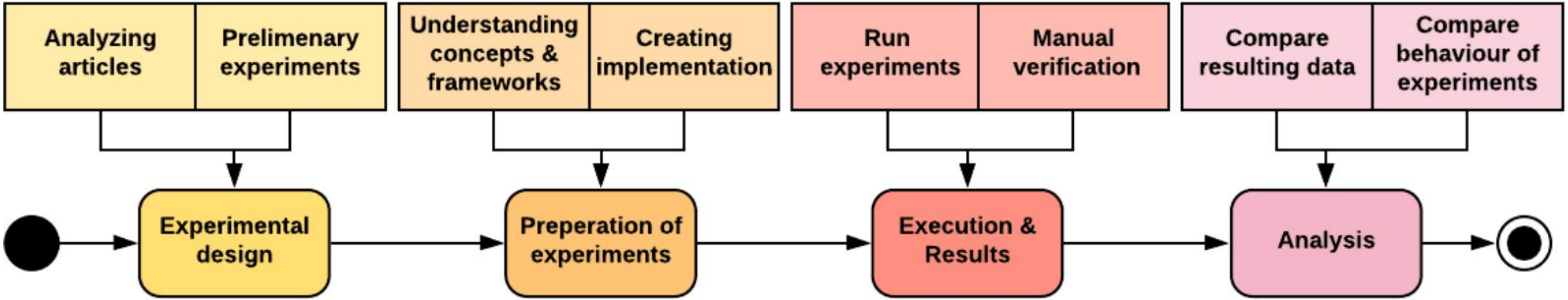


[2] “Overparameterized Neural Networks Can Implement Associative Memory” - A. Radhakrishnan, M. Belkin, C. Uhler

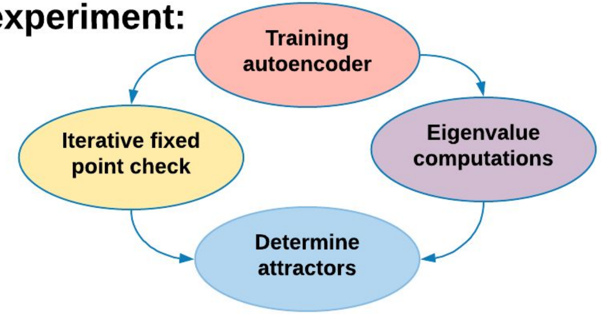
Method : Controlled experiment(s)



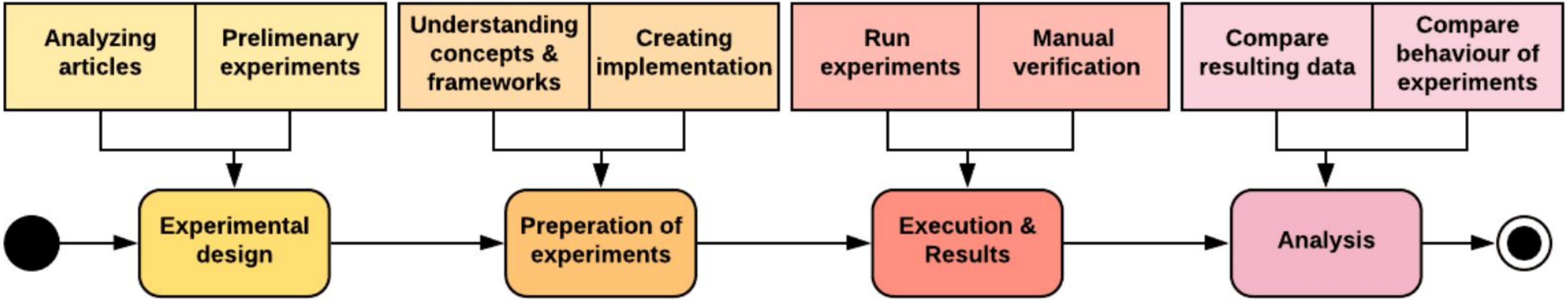
Method : Controlled experiment(s)



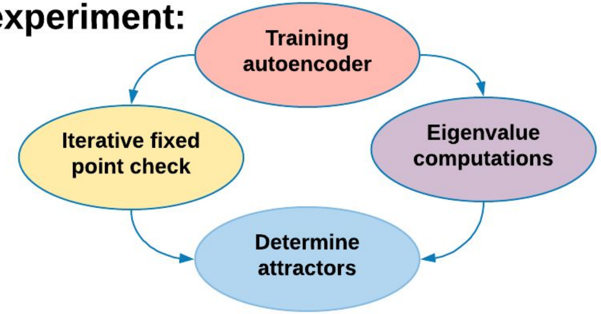
Single experiment:



Method : Controlled experiment(s)



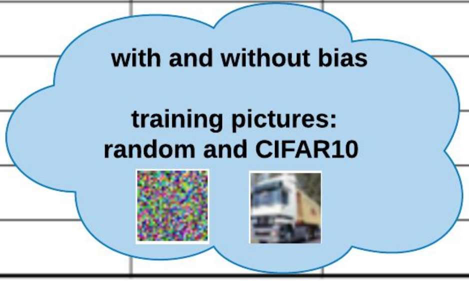
Single experiment:



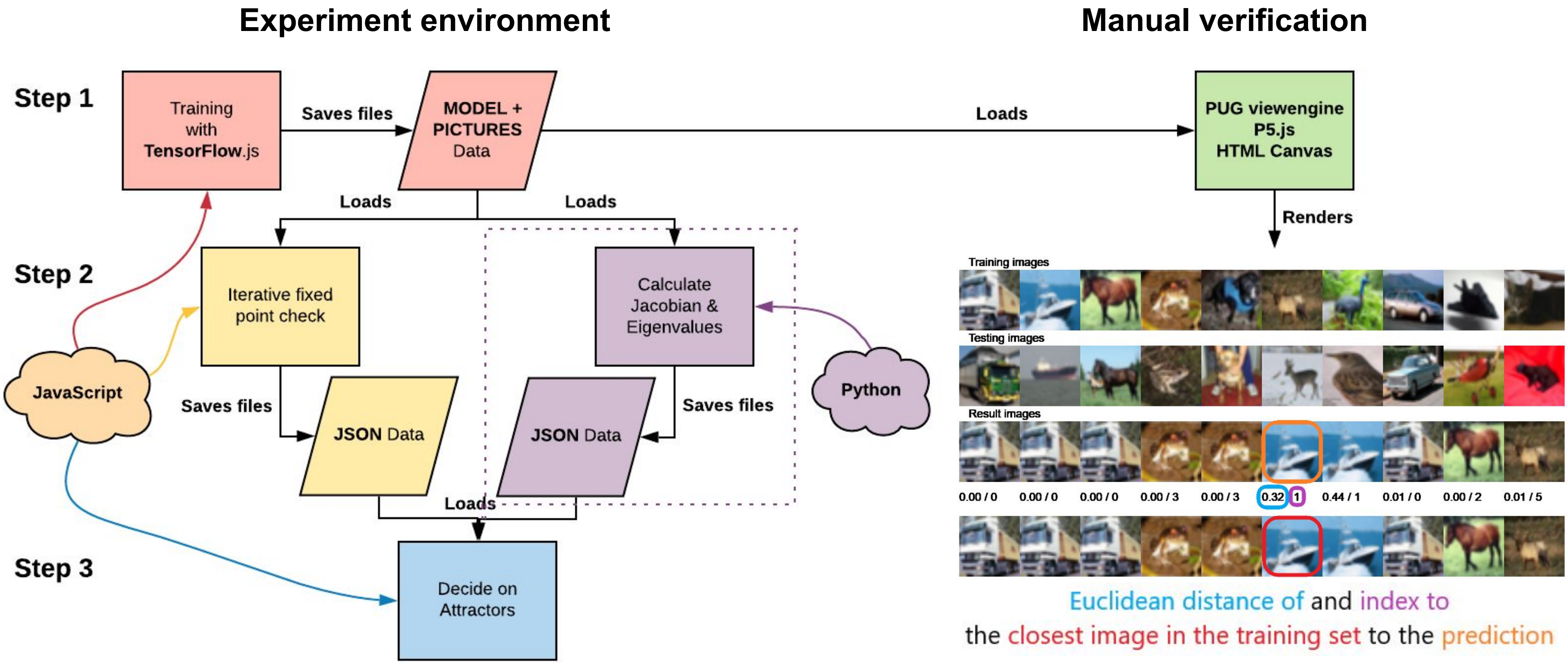
Each experiment repeats 4x, totalling 240 experiments

Scope:

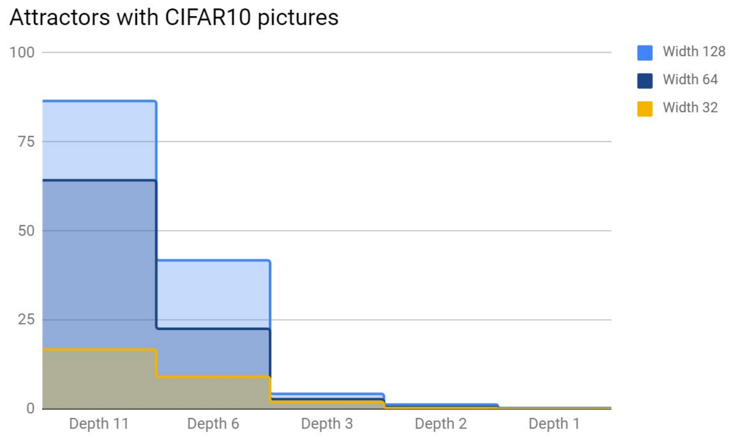
depth \ width	128	64	32
11			
6			
3			
2			
1			

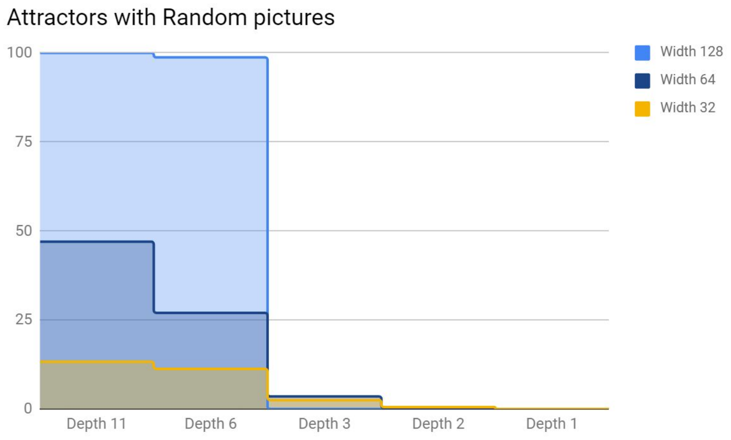


Implementation

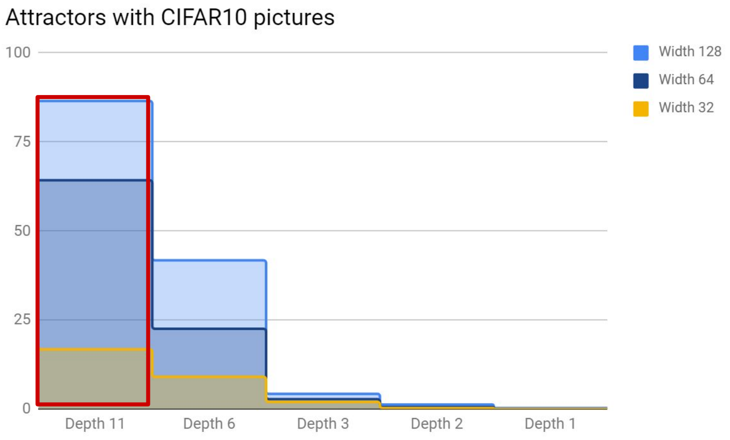


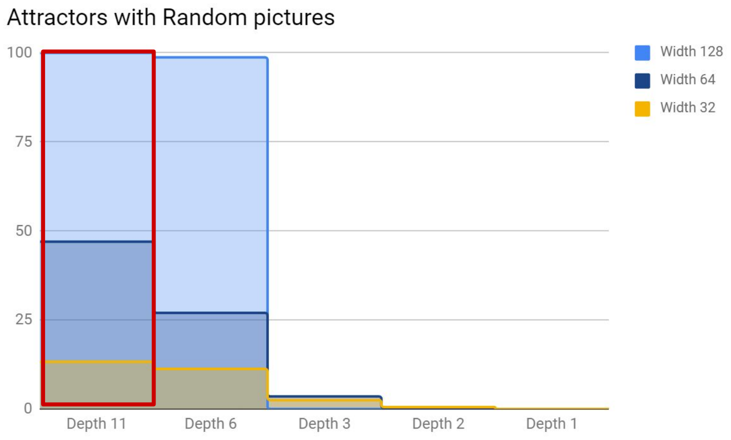
Results & Analysis (trained models with bias)



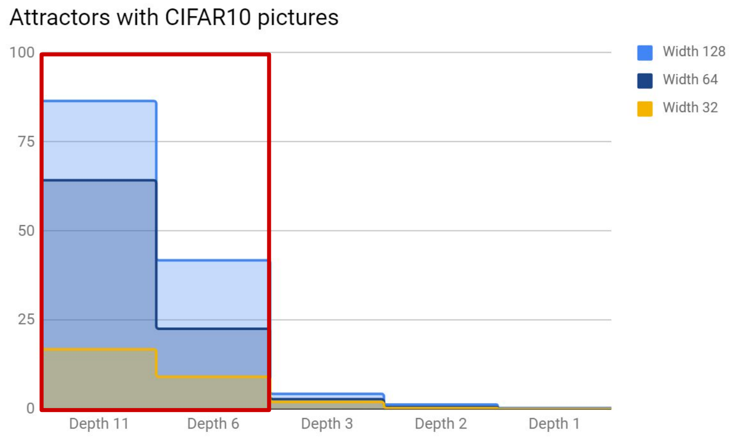


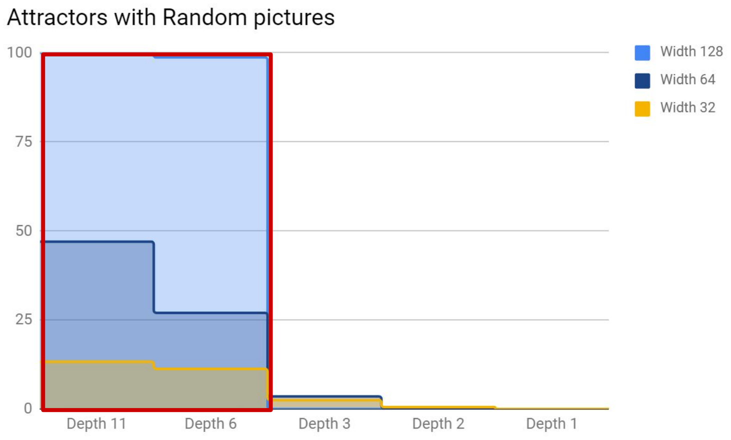
Results & Analysis
(trained models with bias)



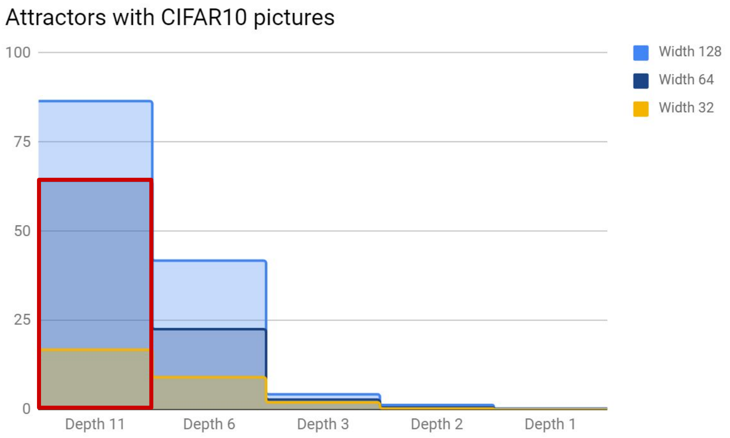


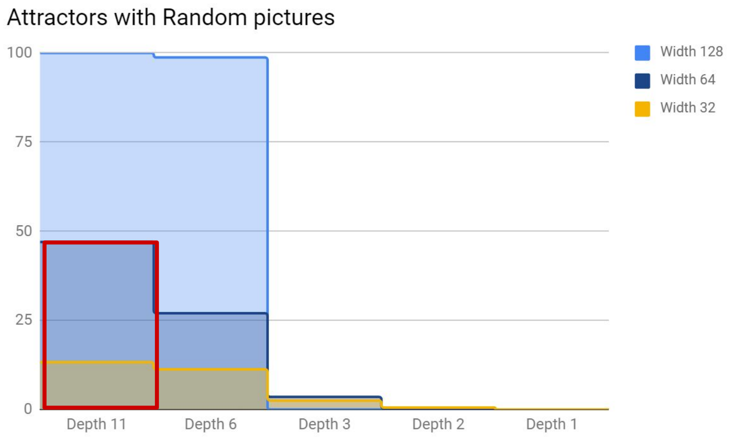
Results & Analysis
(trained models with bias)





Results & Analysis
(trained models with bias)

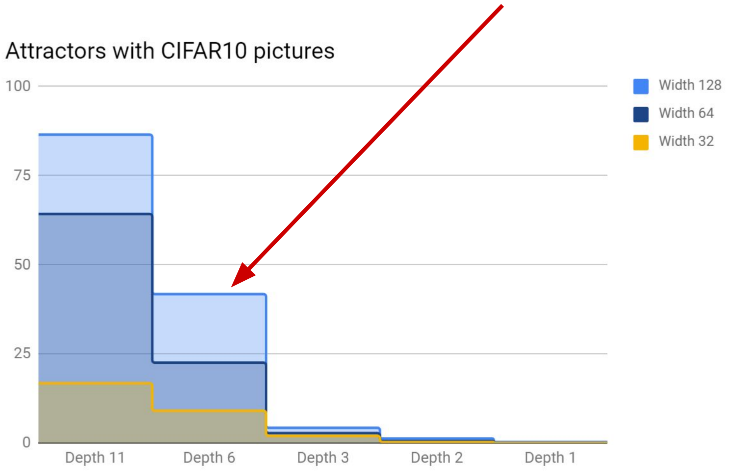


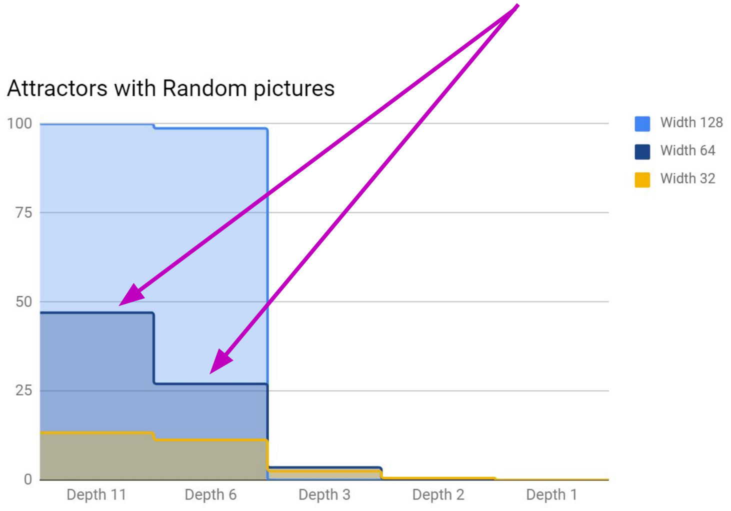


Results & Analysis
(trained models with bias)

- CIFAR10 pictures:
  greater impact when changing depth

- Random pictures:
  greater impact when changing width



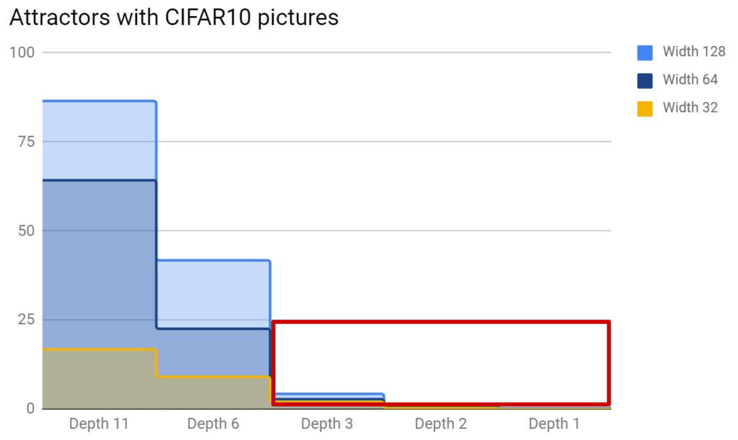


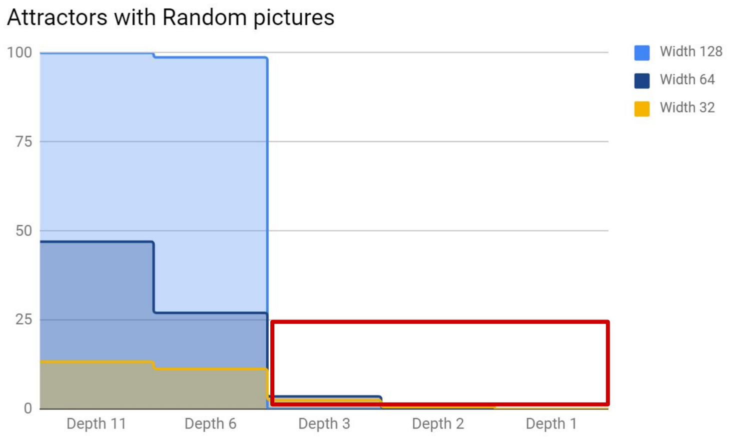
Results & Analysis (trained models with bias)

- CIFAR10 pictures:
  greater impact when changing depth

- Random pictures:
  greater impact when changing width

- Sufficient depth is required for creating attractors





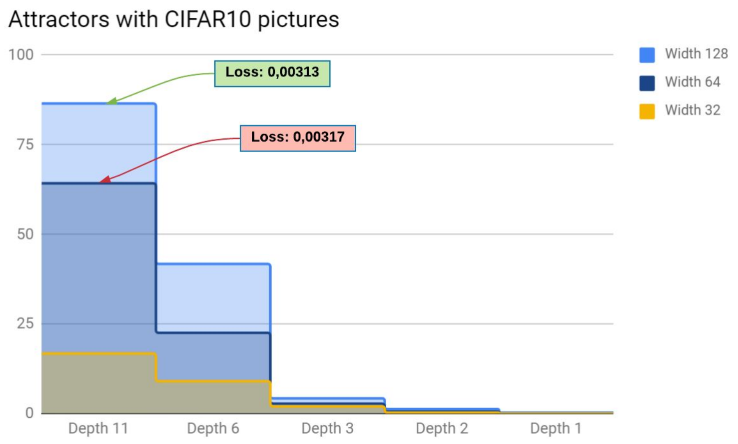
Results & Analysis
(trained models with bias)

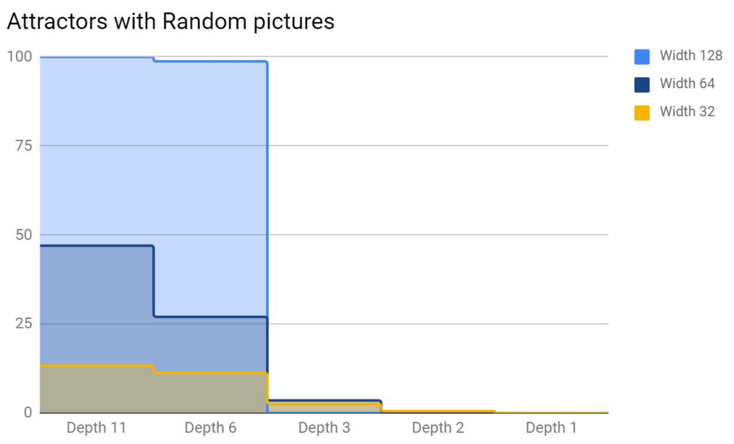
- CIFAR10 pictures:
  greater impact when changing depth

- Good loss and number of attractors
  are not necessarily related

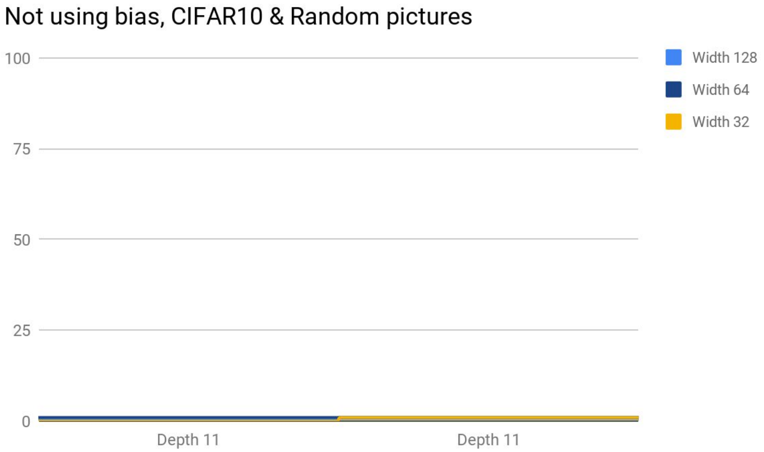
- Random pictures:
  greater impact when changing width

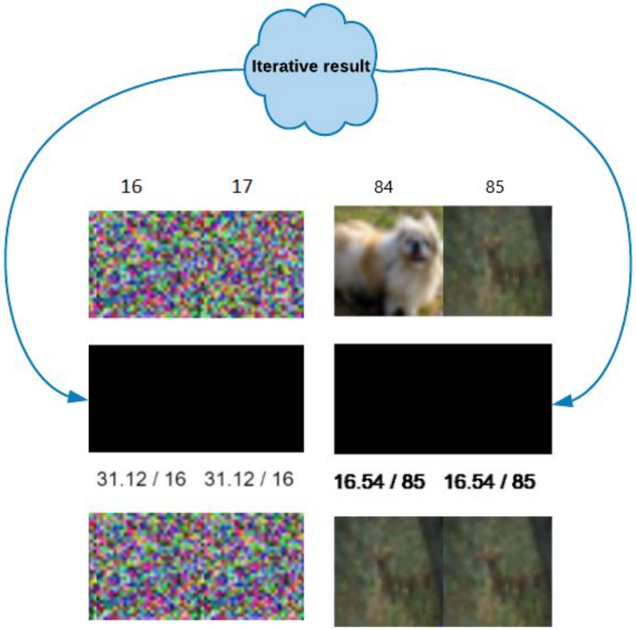
- Sufficient depth is required for
  creating attractors





Results & Analysis
(trained models without bias)





Not using bias, produces no attractors

Future work

- increase the scope (width and depth) of autoencoders
- the effect of epochs on trained autoencoders was not covered in the thesis (only trained 50k).
- training pictures was always 100, the impact of changing this amount would be interesting.
- training with pictures with a more specific structure could give interesting results.
- optimizing the thesis' notion of iterative fixed points, excluding the "false positives".
- a mathematical proof showing that every data point becomes an attractor for a sufficiently generic training set for a sufficiently large ReLU autoencoder

An autoencoder is a mathematical function; in this thesis:

$$\alpha : \mathbb{R}^{3072} \to \mathbb{R}^{3072}$$

At every picture, we can compute the Jacobian matrix $J_\alpha($$)$ $\rightarrow$ this is a 3072 x 3072 matrix.

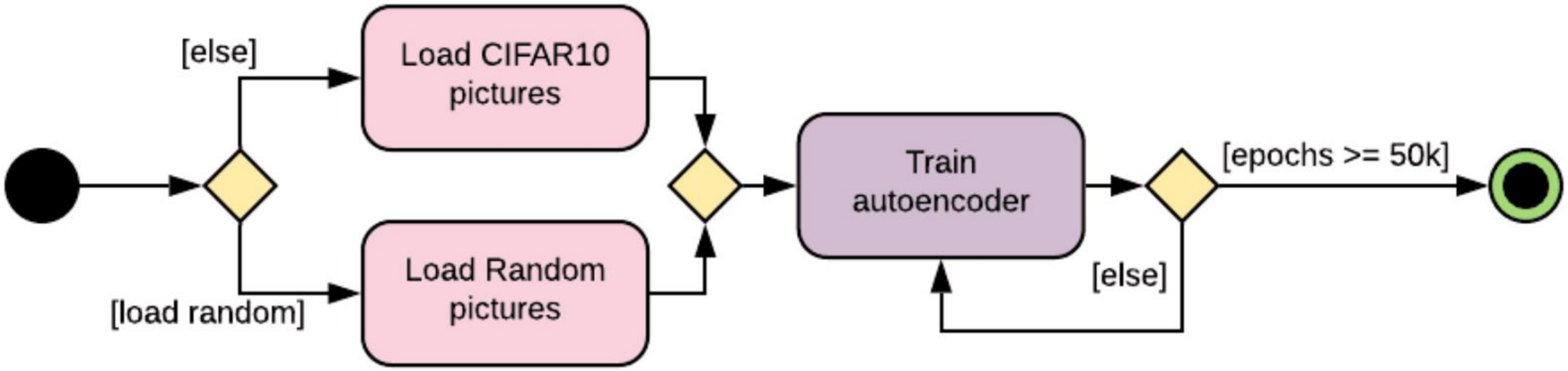
[1] and [2] show for a **perfectly trained autoencoder with loss 0**:

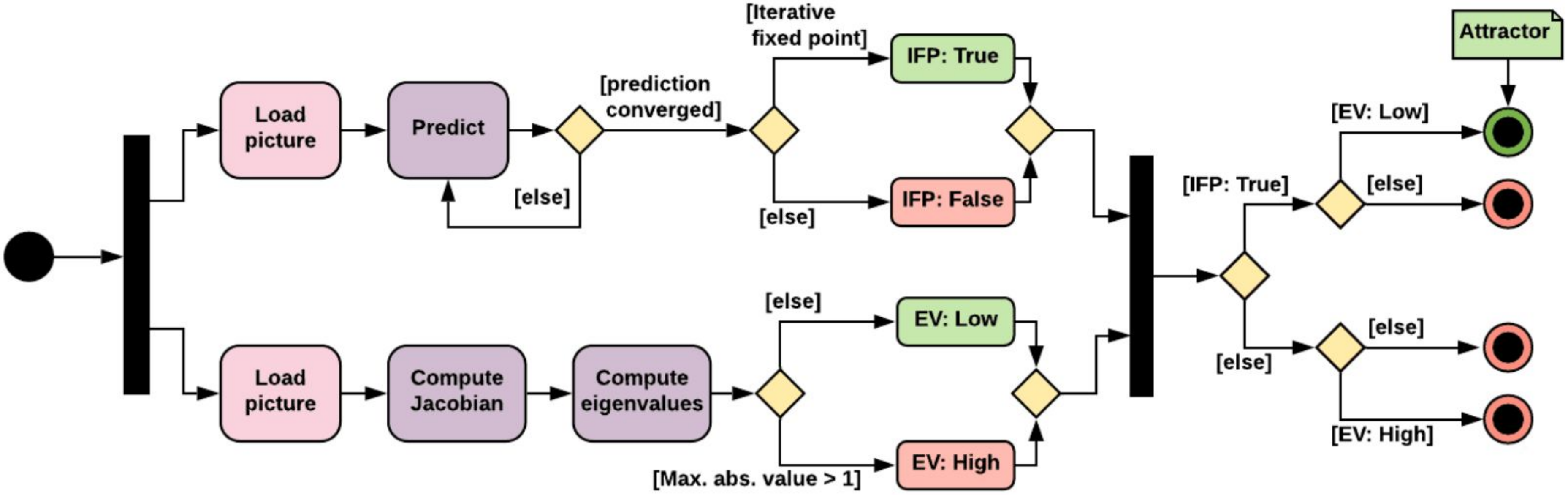
-  is an attractor if the highest absolute value of the eigenvalues of $J_\alpha($$)$ is smaller than 1.
-  is not an attractor if this highest absolute value is larger than 1.

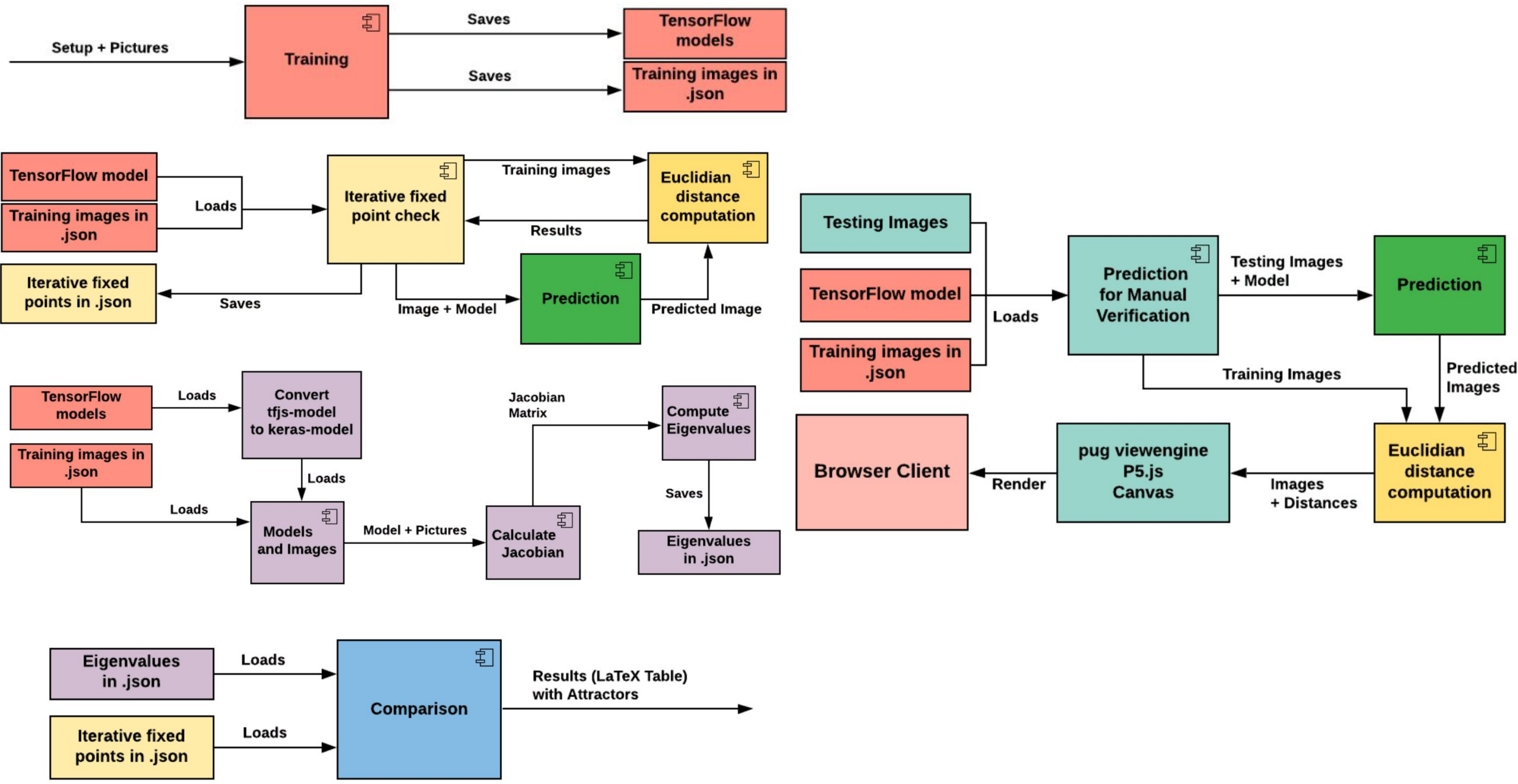
For an **autoencoder with loss > 0**,  is an attractor if this highest absolute value is smaller than 1 and  is an iterative fixed point.

		largest absolute value of all eigenvalues: < 1	largest absolute value of all eigenvalues: > 1
iterative fixed point?	yes	attractor	not attractor
	no	not attractor	not attractor

[1] “Memorization in overparameterized autoencoders” - A. Radhakrishnan, K.D. Yang, M. Belkin and C. Uhler
[2] “Overparameterized Neural Networks Can Implement Associative Memory” - A. Radhakrishnan, M. Belkin, C. Uhler


[else]
Load CIFAR10 pictures
Train autoencoder
[epochs >= 50k]
[else]
[load random]
Load Random pictures


[Iterative fixed point]
IFP: True
Attractor
Load picture
Predict
[prediction converged]
[EV: Low]
[else]
[IFP: True]
[else]
IFP: False
[else]
[else]
EV: Low
[else]
[else]
Load picture
Compute Jacobian
Compute eigenvalues
[EV: High]
[Max. abs. value > 1]
EV: High

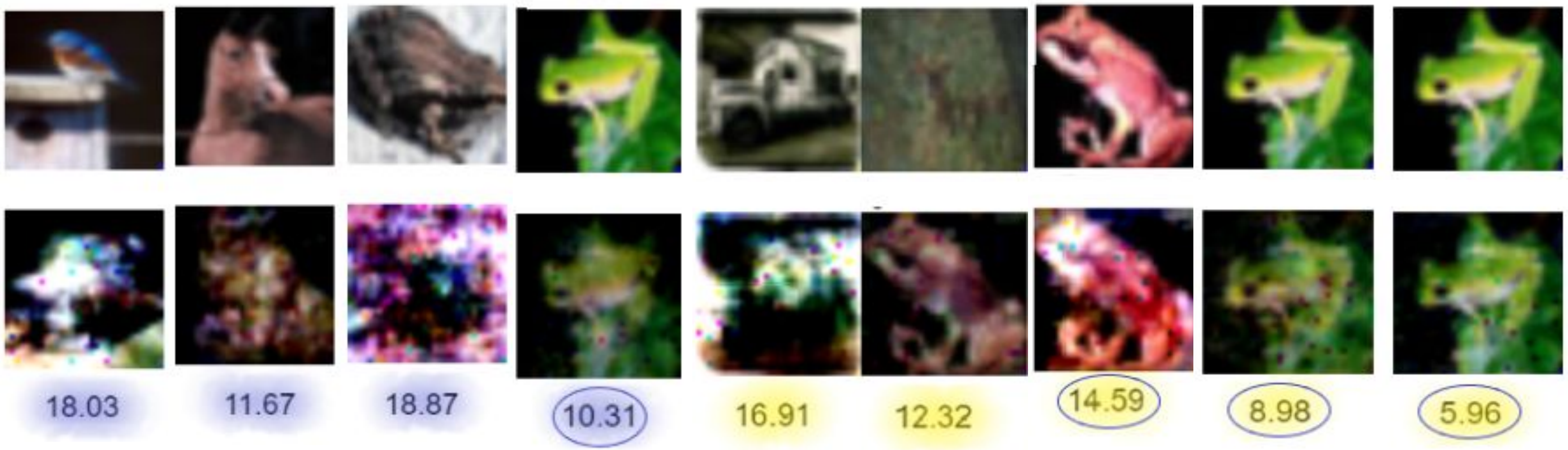


Setup + Pictures
Training
Saves
Saves
TensorFlow models
Training images in .json
TensorFlow model
Training images in .json
Loads
Iterative fixed point check
Training images
Euclidian distance computation
Results
Iterative fixed points in .json
Saves
Image + Model
Prediction
Predicted Image
Testing Images
TensorFlow model
Training images in .json
Loads
Prediction for Manual Verification
Testing Images + Model
Prediction
Training Images
Predicted Images
Browser Client
Render
pug viewengine P5.js Canvas
Images + Distances
Euclidian distance computation
TensorFlow models
Loads
Convert tfjs-model to keras-model
Training images in .json
Loads
Loads
Models and Images
Model + Pictures
Calculate Jacobian
Jacobian Matrix
Compute Eigenvalues
Saves
Eigenvalues in .json
Eigenvalues in .json
Loads
Iterative fixed points in .json
Loads
Comparison
Results (LaTeX Table) with Attractors

Results & Analysis
(iterative fixed points at lower depths
& manual verification)

'false predictions':

- 82% at depth 1
- 36% at depth 2
- 19% at depth 3



Examples depth 2

Examples depth 1