Analysis of Neural Networks. Applications to Interpretability and Uncertainty

Gabriel Marín Sánchez

Directors: Antoni Benseny Ardiaca Alberto Rubio Muñoz

July 15, 2020

UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques i Informàtica

Contents

1. Introduction
2. Feed-Forward Neural Networks
3. Supervised Learning
4. Unsupervised Learning
5. Interpretability
6. Uncertainty
7. Conclusions

Introduction

- Beginning of neural networks:
  - McCulloch-Pitts neuron (1943)
  - Perceptron (1958)
- Second wave: Universal approximation theorems (Cybenko Hornik, 1990s)
- Neural Networks have revolutionized areas such as computer vision or text processing.
- In general, this technology is treated as a black-box.
- Goals:
  - Give NNs a mathematical treatment.
  - Present and develop methods to understand why NNs perform so well.

Contents

1 Introduction

2 Feed-Forward Neural Networks

3 Supervised Learning

4 Unsupervised Learning

5 Interpretability

6 Uncertainty

7 Conclusions

Description

- Given f', the goal of a NN is to find a function f that approximates f'.
- Linear function, where w is the weight and b the bias:

$$\phi(x) = w^T x + b$$

- Component-wise composition:

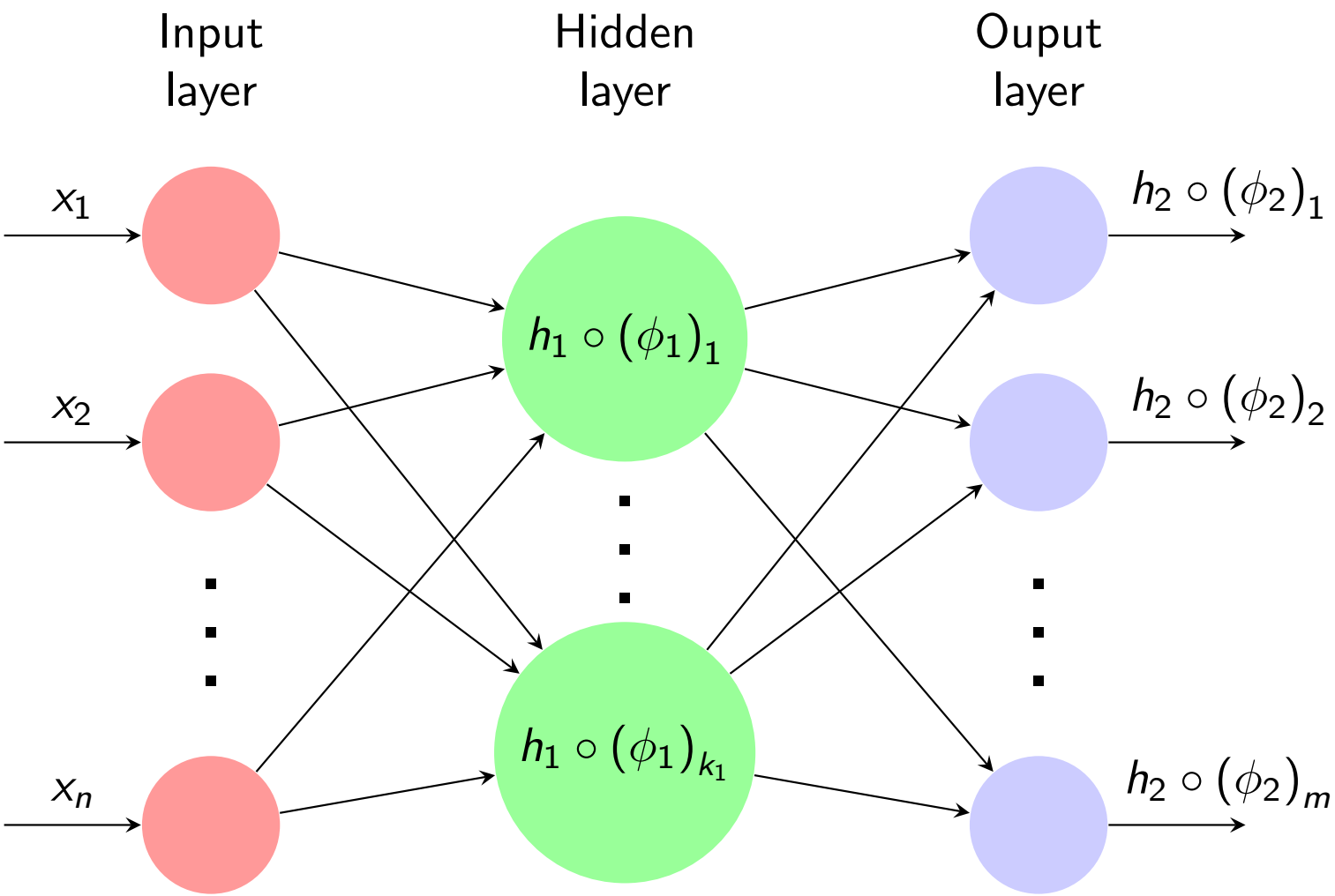
$$(g \bullet f)(x) = ((g \circ f_1)(x), \ldots, (g \circ f_m)(x))$$

- Feed-forward neural network with $L + 1$ layers:

$$f_\theta = h_L \bullet \phi_L \circ \cdots \circ h_1 \bullet \phi_1$$

h_i activation functions

Representation



Learning Process

- Loss Function: Computes the predictions' error.
- Optimization of the loss function: gradient descent.

Loss Function

Let $x = \{x_i\}_{i=1,\dots,N}$ be the input values, $y' = \{y'_i\}_{i=1,\dots,N}$ the target and f_θ NN function.

- Regression: Mean squared error loss.

$$J_\theta^{MSE}(x, y') = \frac{1}{N}\sum_{i=1}^{N} \|y' - f_\theta(x)\|^2$$

- Classification: Cross-entropy loss.
  - Multiple classes:

$$J_\theta^{MCE}(x, y') = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c\in C} y'_c \log(f_{\theta,c}(x))$$

  - Binary case:

$$J_\theta^{MCE}(x, y') = -\frac{1}{N}\sum_{i=1}^{N}\left[(y'\log(f_\theta(x)) + (1-y')\log(1-f_\theta(x))\right]$$

Optimization

- Gradient descent method.

$$w_{ij}^{l} = w_{ij}^{l} - \eta \frac{\partial J_\theta}{\partial w_{ij}^{l}}$$

- Others: Adam, Adagrad, etc.
- Computing the gradients: Backpropagation.

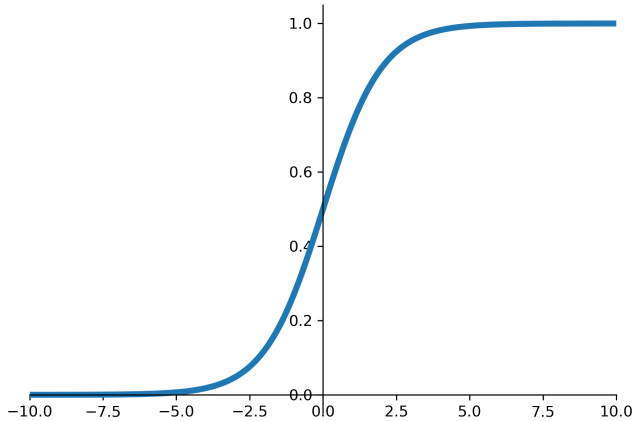
$$\frac{\partial J_\theta}{\partial w_{ij}^{(l)}} = \beta_j^{(l)} z_i^{(l-1)} \quad ; \quad \frac{\partial J_\theta}{\partial b_j^{(l)}} = \beta_j^{(l)}$$

$$\beta_j^{(l)} = \begin{cases} \frac{\partial J_\theta}{\partial z_j^{(l)}} \frac{\partial h_l\left(u_j^{(l)}\right)}{\partial u_j^{(l)}}, & \text{if } l = L \\ \left(\sum_{a \in A} w_{ja}^{(l+1)} \beta_a^{(l+1)}\right) \frac{\partial h_l\left(u_j^{(l)}\right)}{\partial u_j^{(l)}}, & \text{otherwise} \end{cases}$$

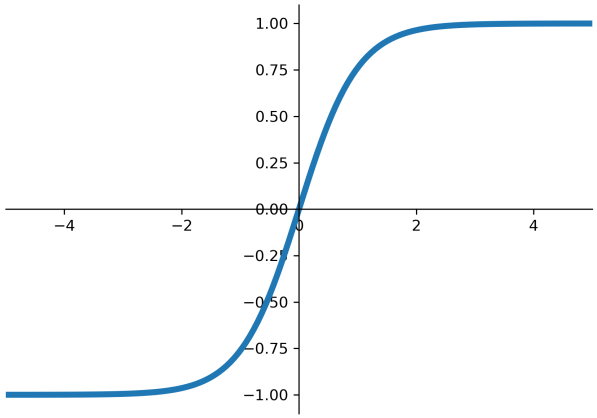
Activation Function

- Non-linearity: Universal approximation theorems.
- Differentiability: Errors when computing the gradients.
- Rang: Stability in the optimisation process.

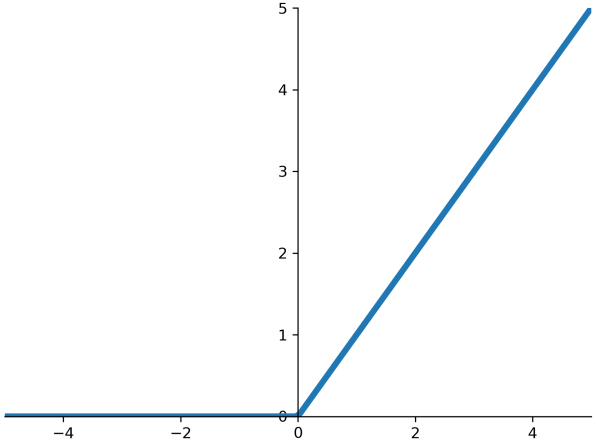
Activation Function



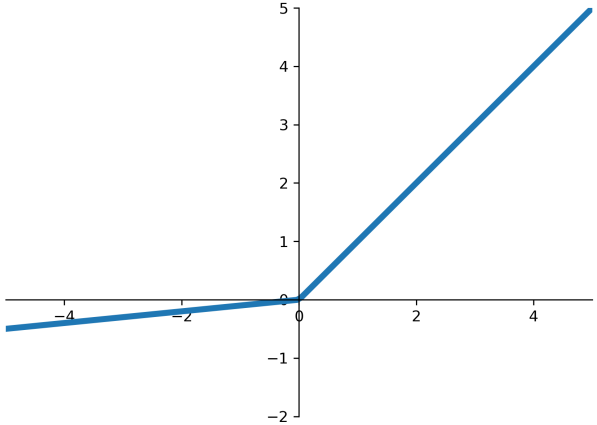
(a) Sigmoid.



(b) Tanh.



(c) ReLU.



(d) Leaky ReLU.

Contents

1 Introduction

2 Feed-Forward Neural Networks

3 Supervised Learning

4 Unsupervised Learning

5 Interpretability

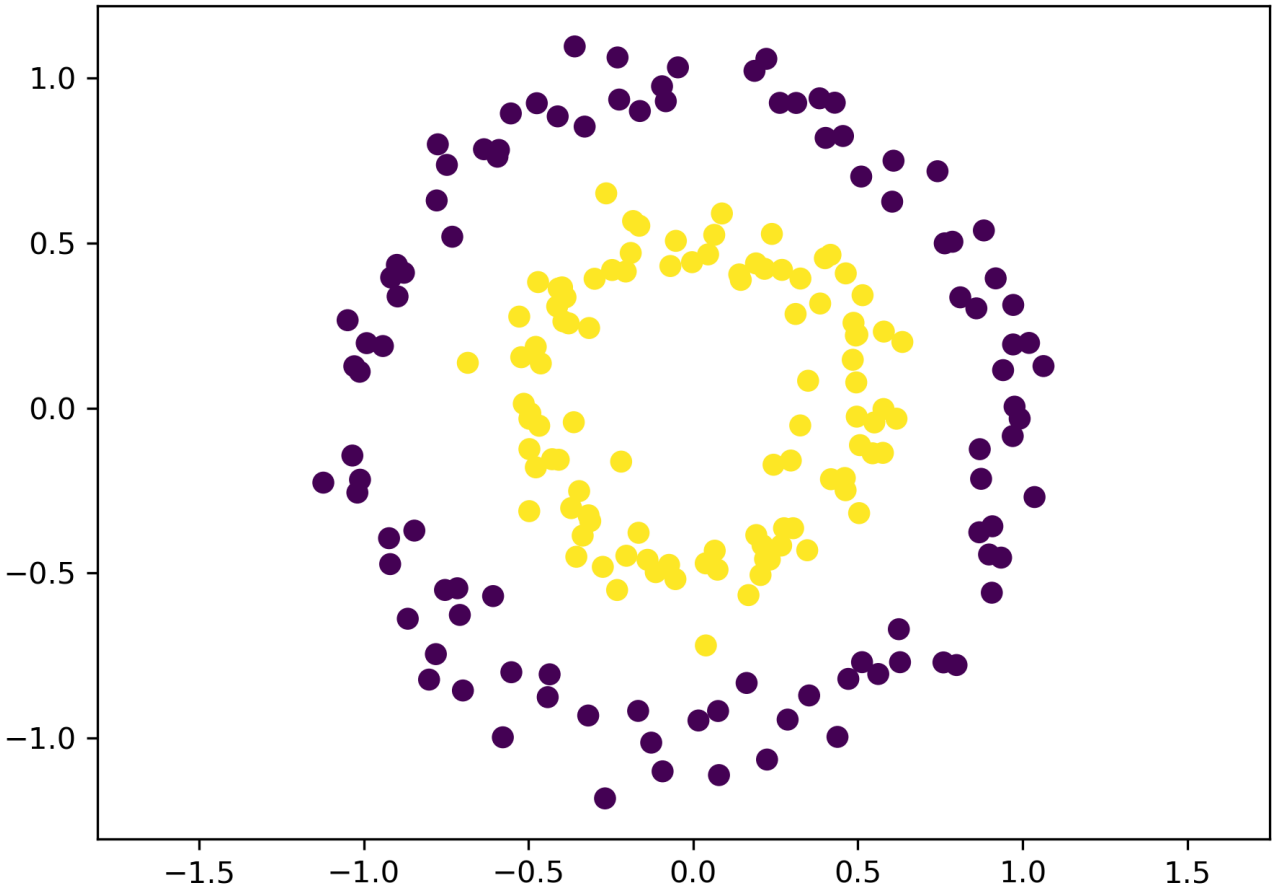
6 Uncertainty

7 Conclusions

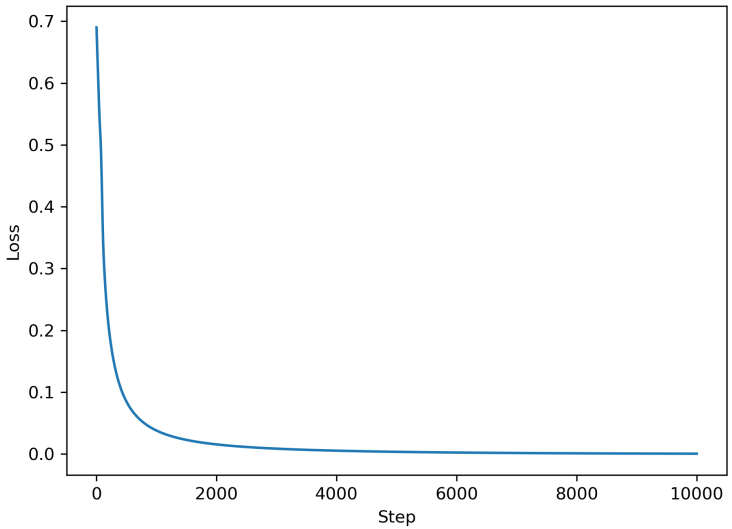
Supervised Learning

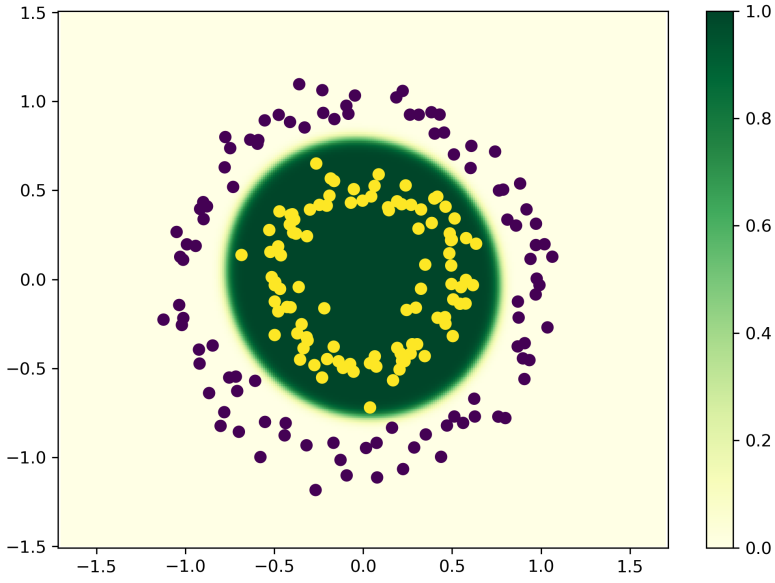
- The network has a specific target for each input.
- The goal is to approximate the function that maps the input to the target.
- Two classification problems:
  - Binary classification of two concentric circles.
  - Binary classification of points in a 1D segment.

Two Concentric Circles

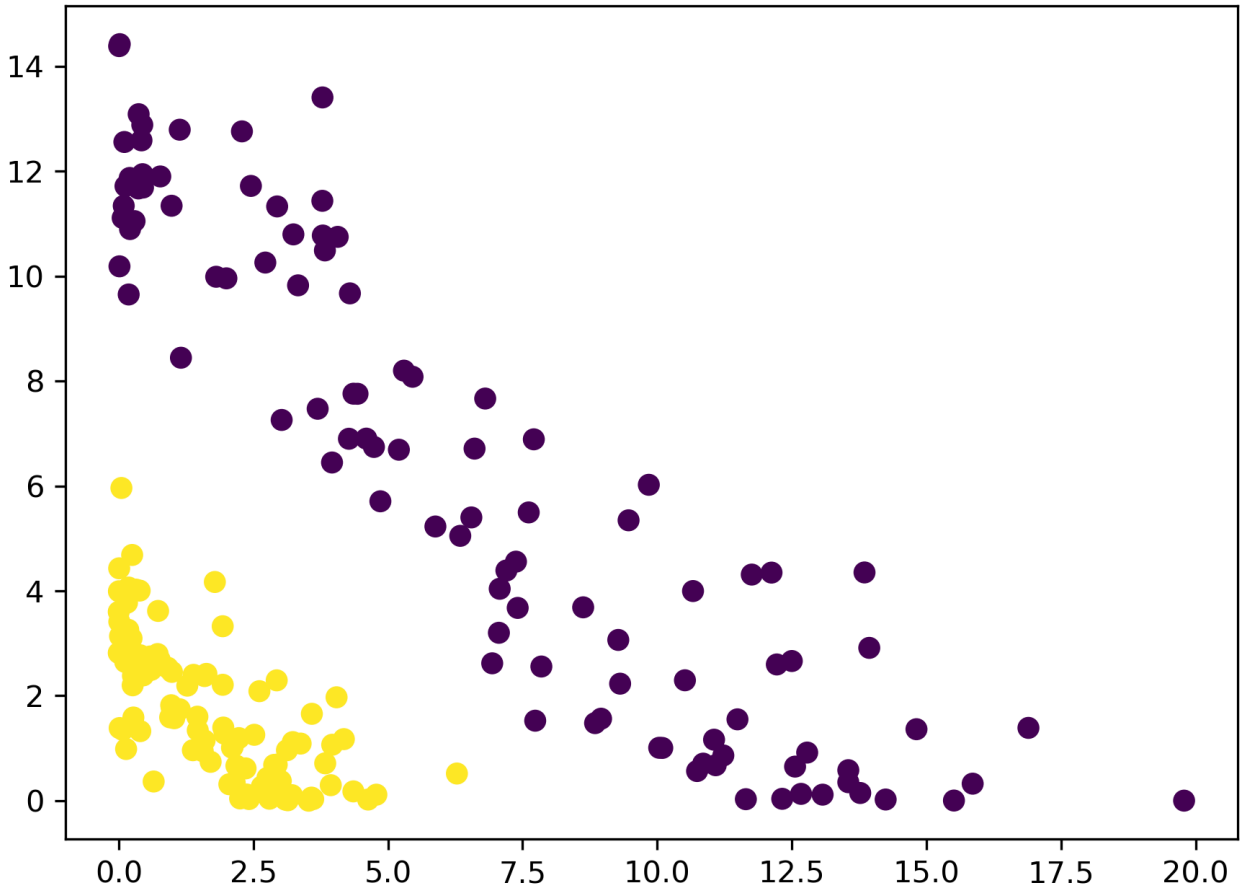


Activation Function: x^2

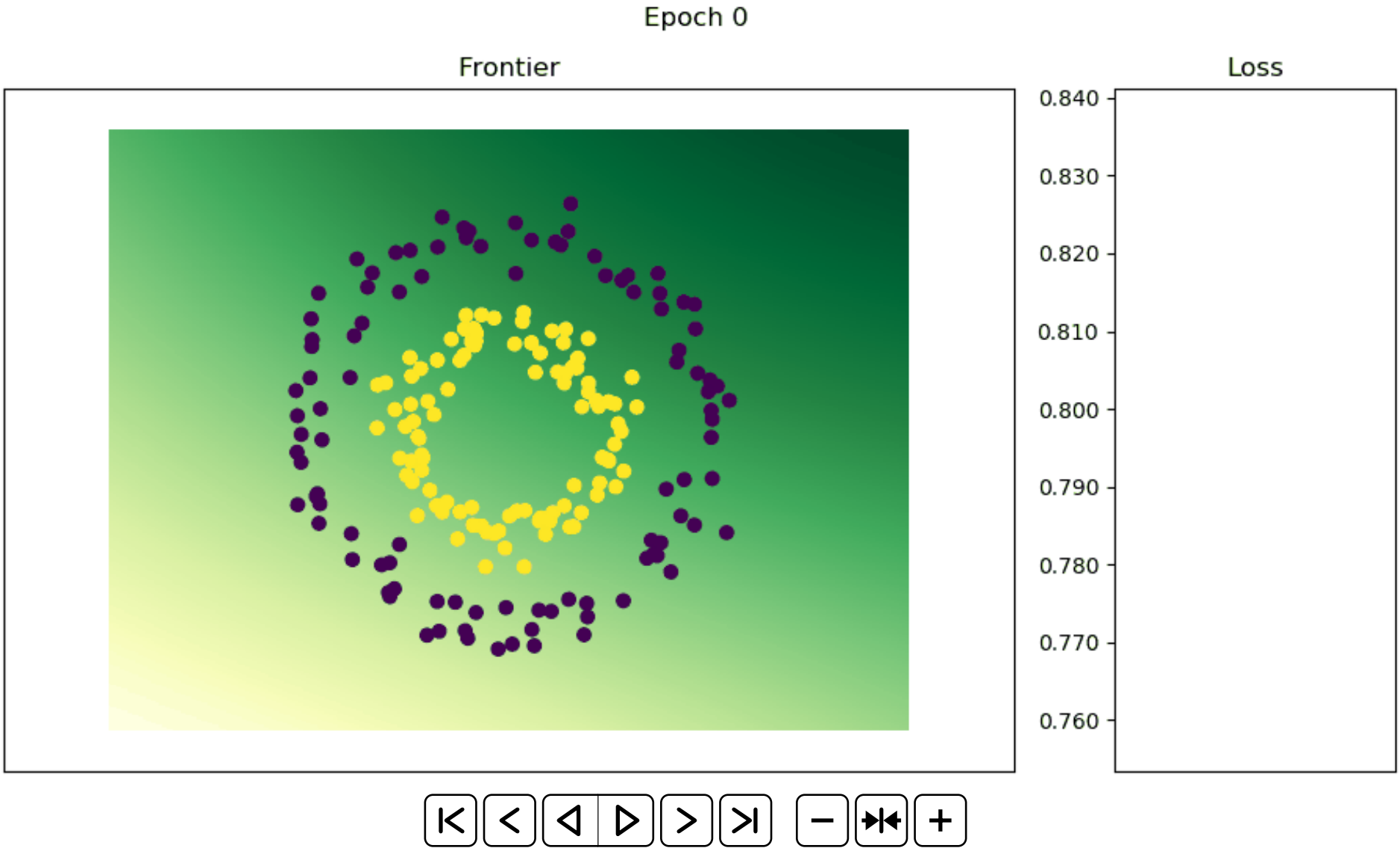




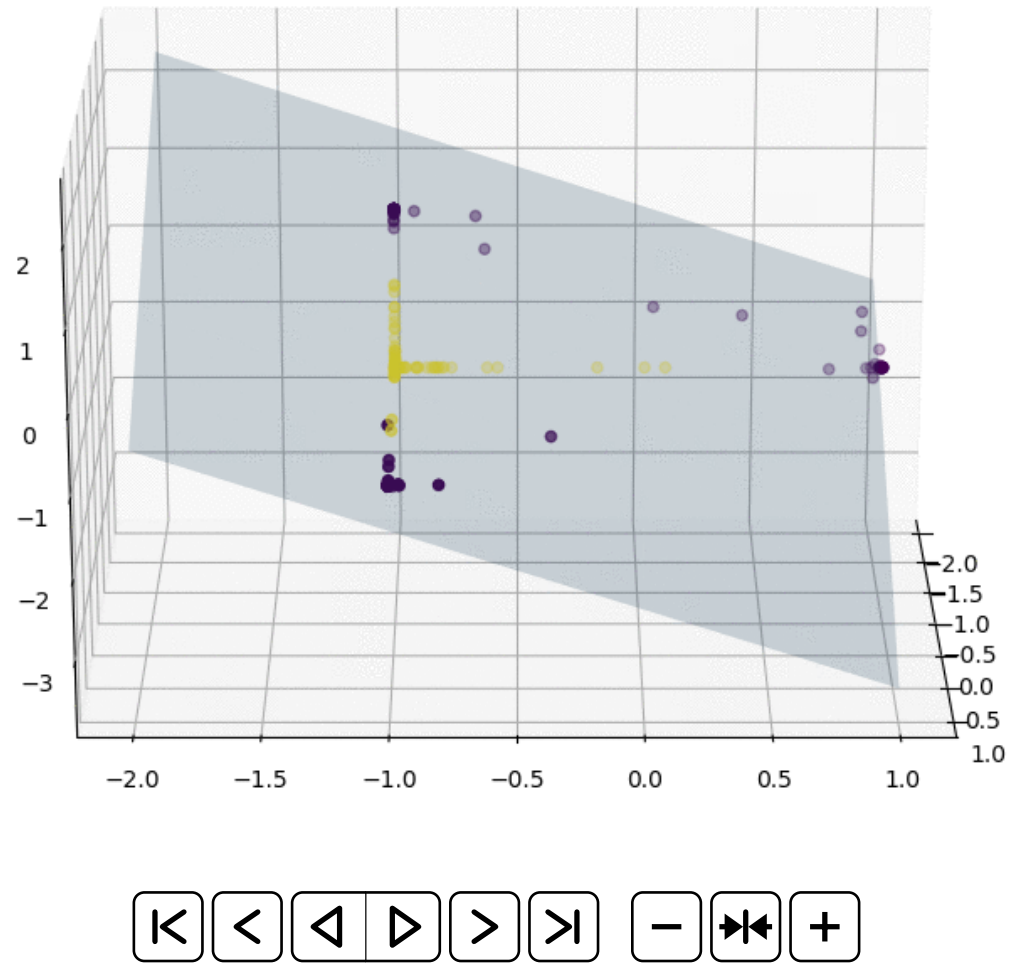
Activation Function: x^2



Activation Function: tanh



Activation Function: tanh



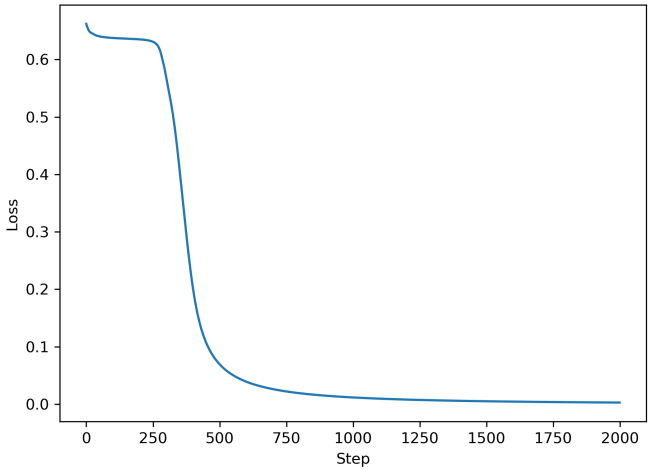
1D Segment

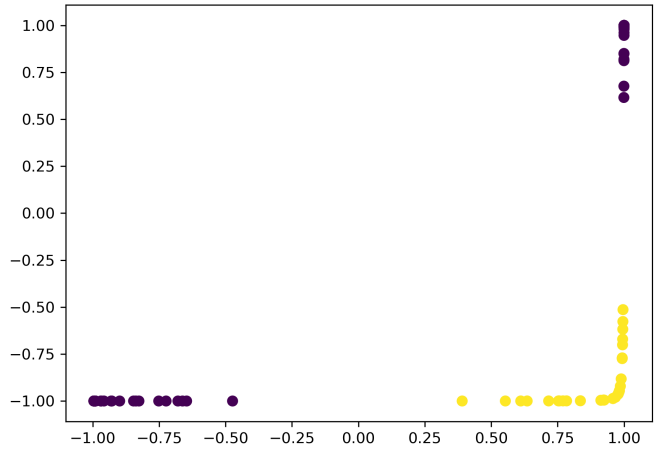
- We divide a 1D segment in s subgroups of 30 points each.
- We assign the label 0 to subgroups with odd index and 1 to the rest.

- Two scenarios:
  - Injective activation function.
  - Non-injective activation function.

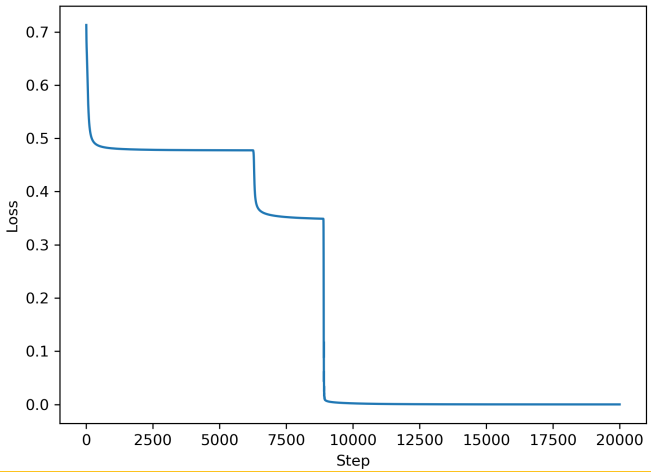
Injective activation function: tanh

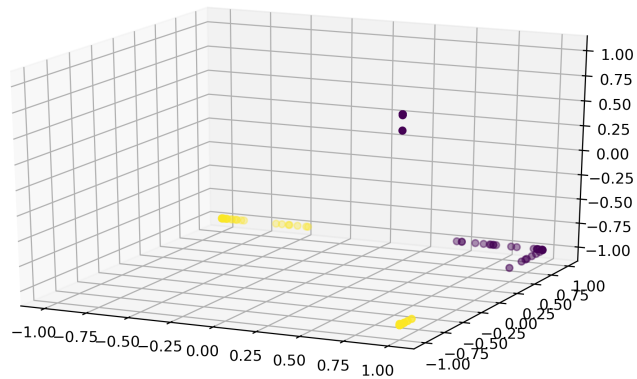
$s = 3$





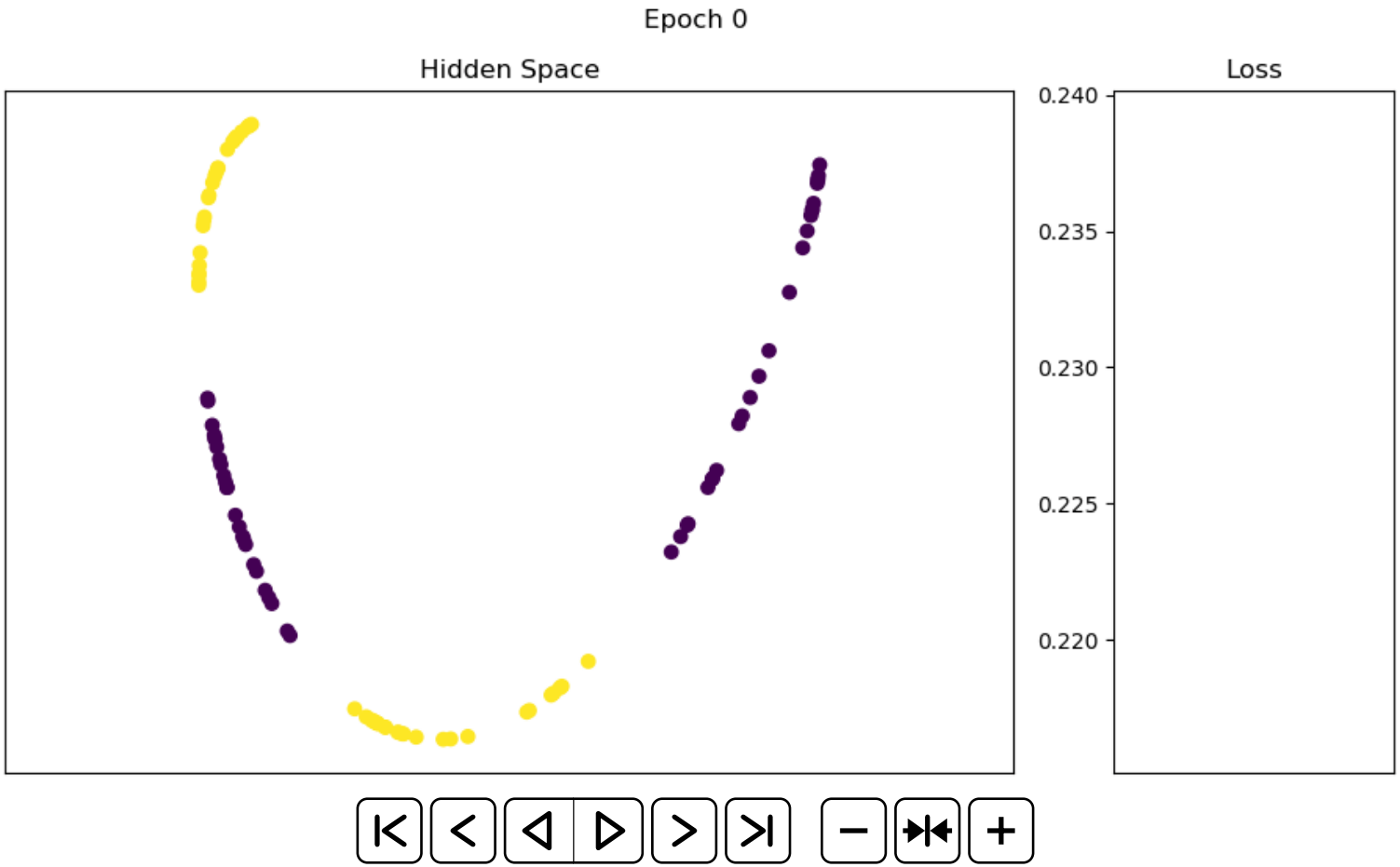
$s = 4$





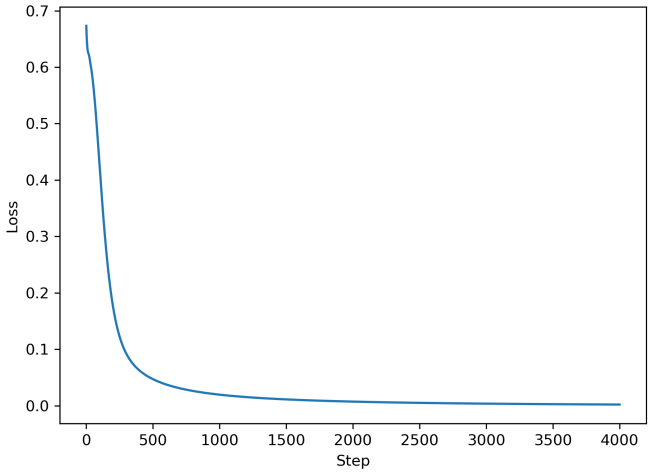
Non-injective activation function: sin

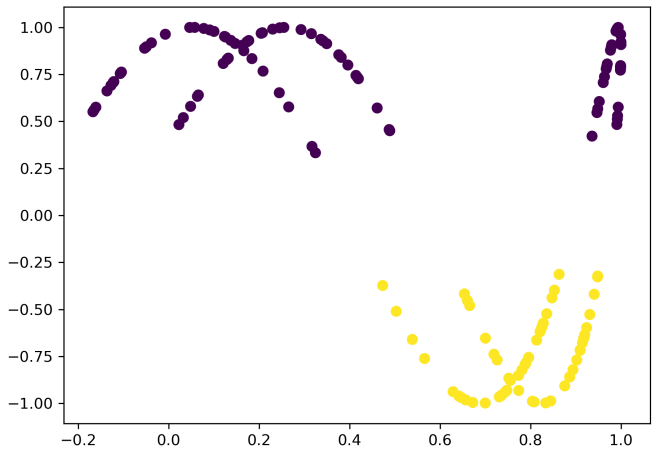
$s = 4$



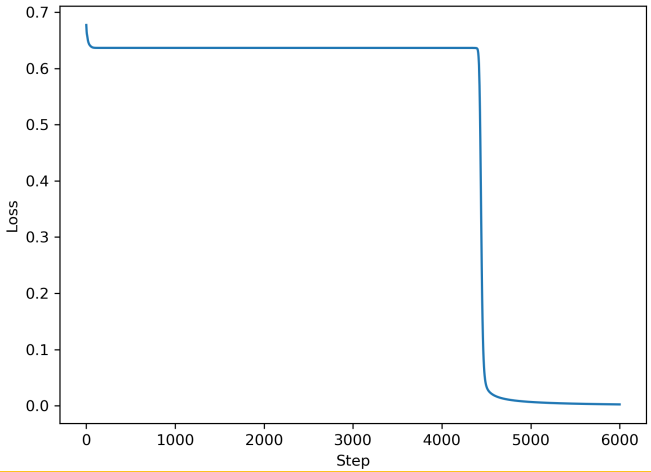
Non-injective activation function: sin

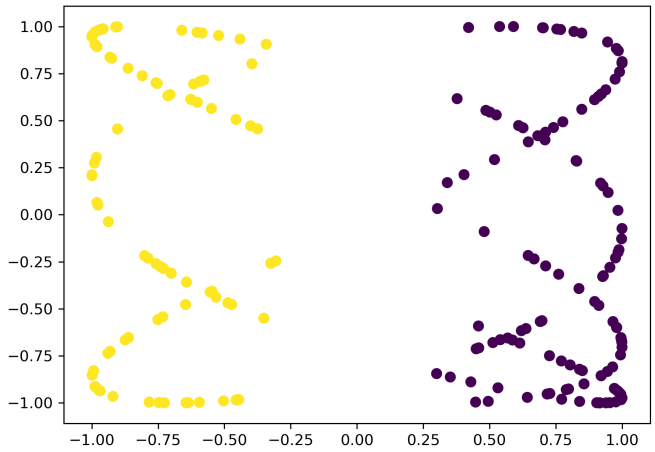
$s = 5$





$s = 7$





Periodic activation functions

Implicit Neural Representations with Periodic Activation Functions

Vincent Sitzmann* sitzmann@cs.stanford.edu
Julien N. P. Martel* jnmartel@stanford.edu
Alexander W. Bergman awb@stanford.edu
David B. Lindell lindell@stanford.edu
Gordon Wetzstein gordon.wetzstein@stanford.edu
Stanford University
vsitzmann.github.io/siren/

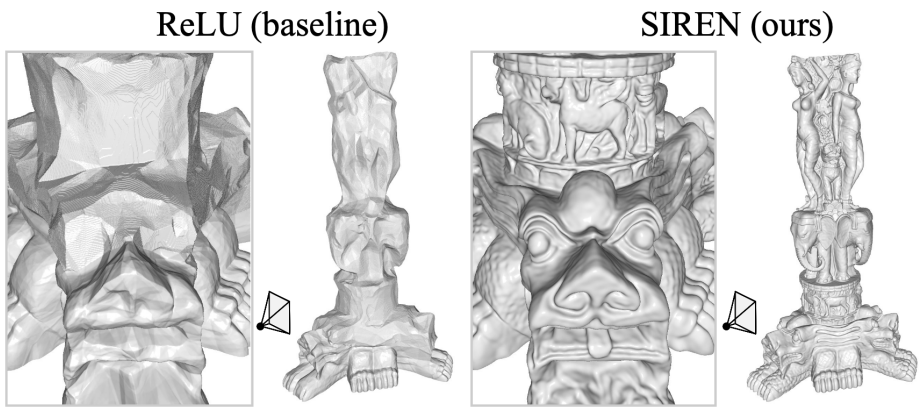


Figure: June 2020.

Contents

1 Introduction

2 Feed-Forward Neural Networks

3 Supervised Learning

4 Unsupervised Learning

5 Interpretability

6 Uncertainty

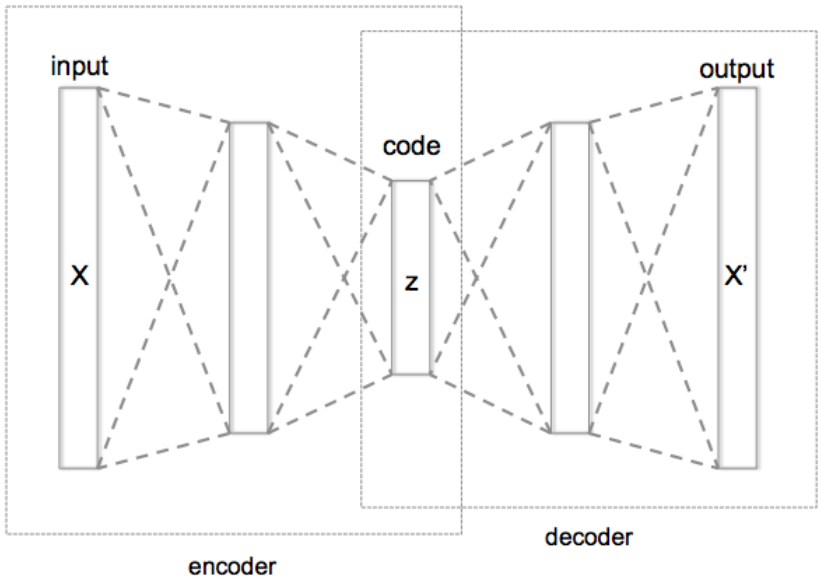
7 Conclusions

Unsupervised Learning

- The network does not have a specific target for each input.
- The NN attempts to learn properties from the input data.
- Representation problem using time series and Autoencoders.

Autoencoder

- The goal is to get an approximation of the input data.
- This model encodes the data into a space with lower dimension and then decodes it.
- Acts as an information bottleneck.



Time Series

- Daily historic market data from 500 random companies, obtained using *Yahoo! finance*.

- A total of 3017 time series of length 365.

Time Series: Data Transformation

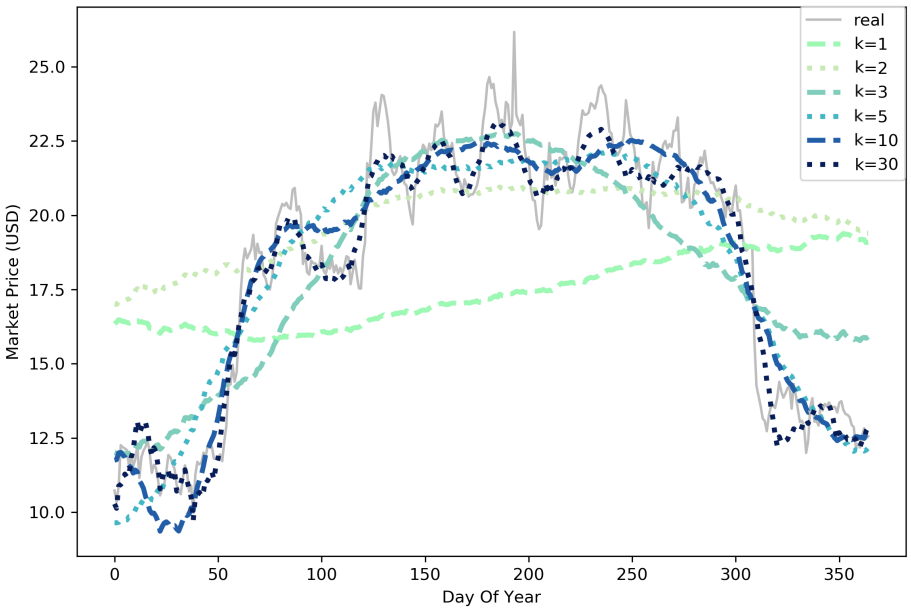
- The time series might have different values and then the range may be too wide.
- Two transformations:
  - Normalization: Values between 0 and 1.

$$x' = \frac{x - \min}{\max - \min}$$

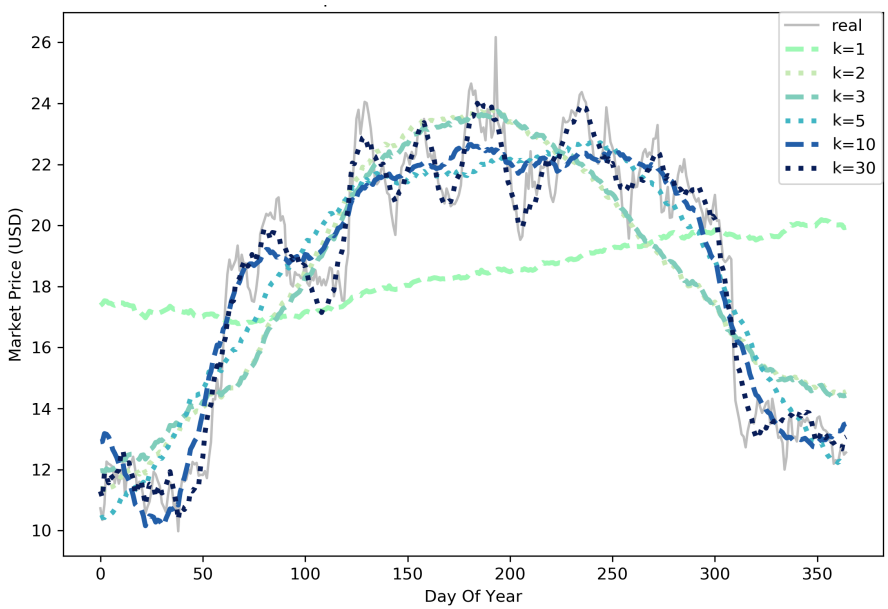
  - Standardization: Mean 0 i variance 1.

$$x' = \frac{x - \mu}{\sigma}$$

Time Series: Results

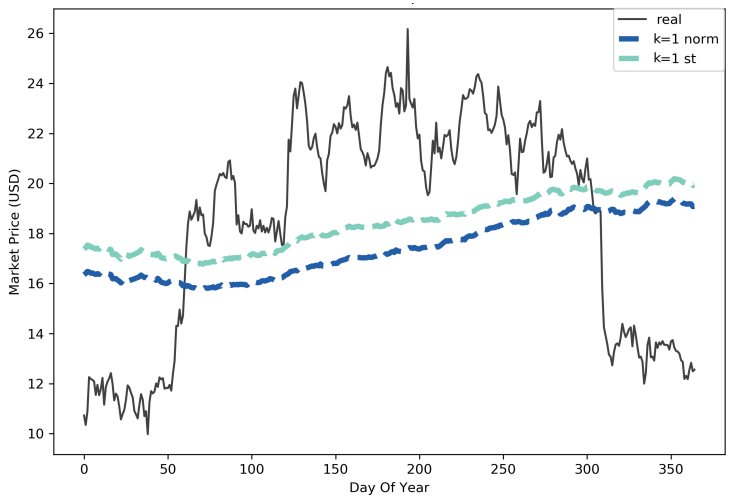


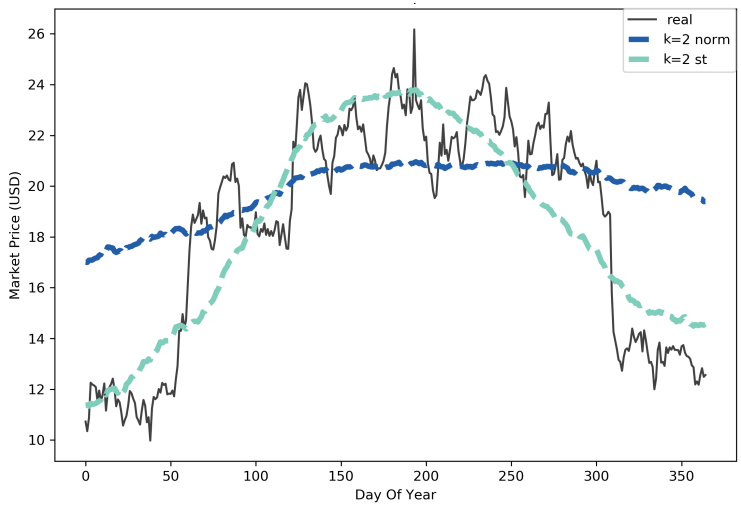
(a) Normalization.

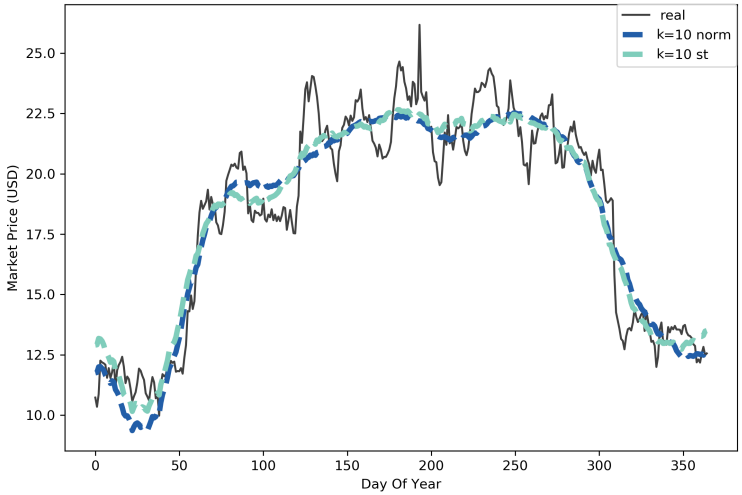


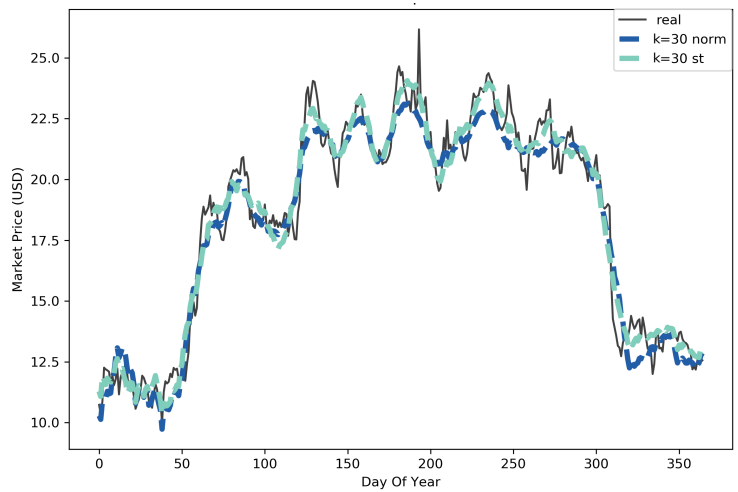
(b) Standardization.

Time Series: Comparison

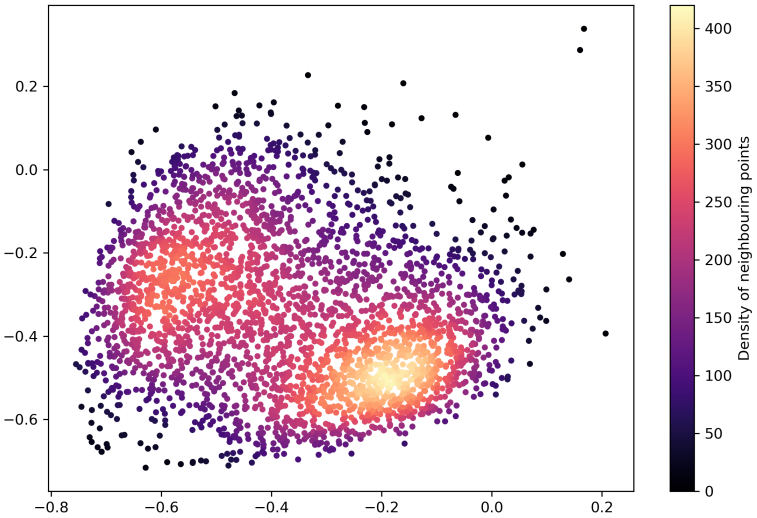




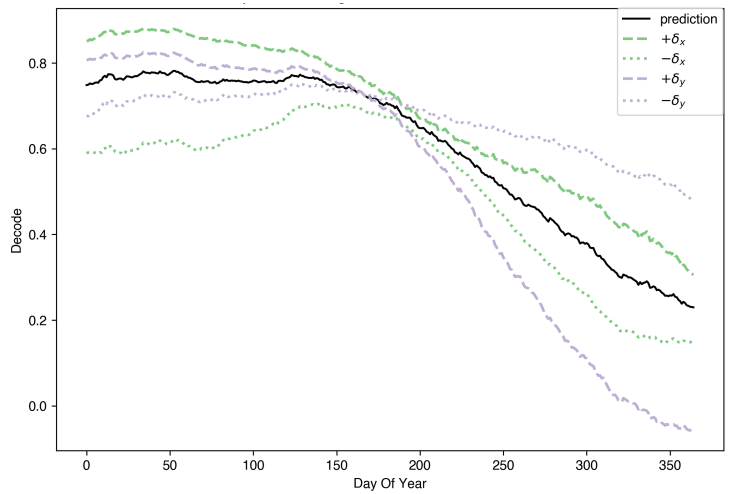




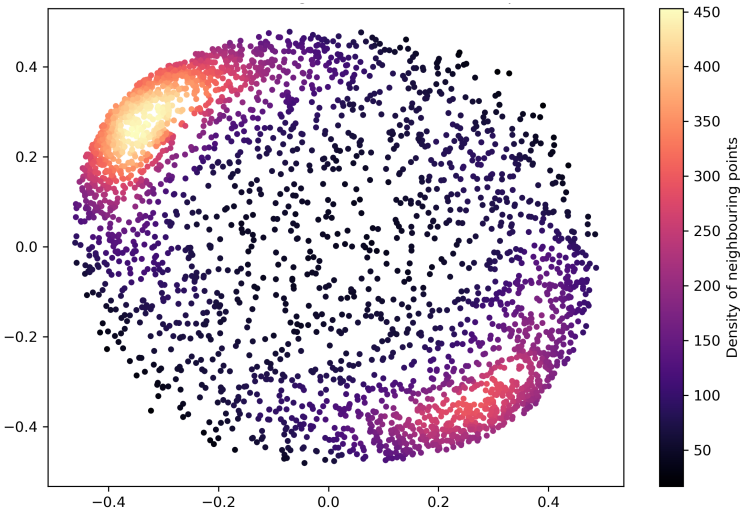
Time Series: Latent Space — $k = 2$



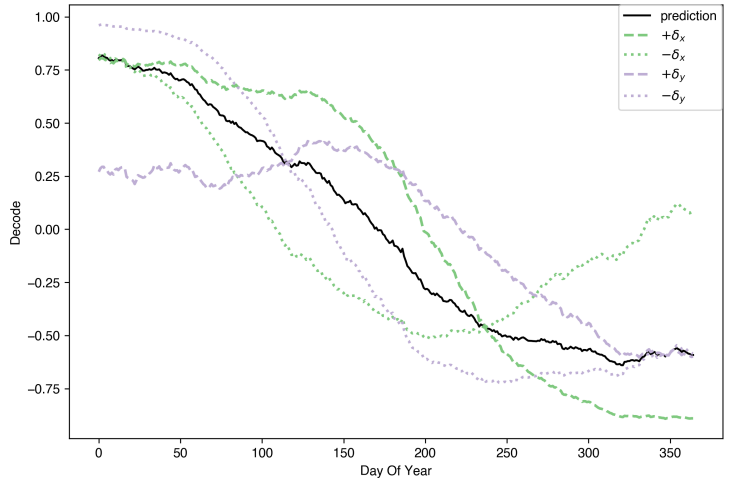
(a) Latent Space Normalization.



(b) Decoding Normalization.

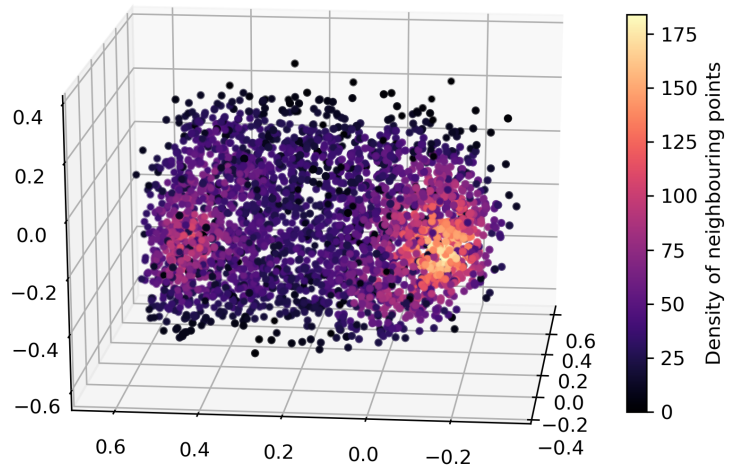


(c) Latent Space Standardization.

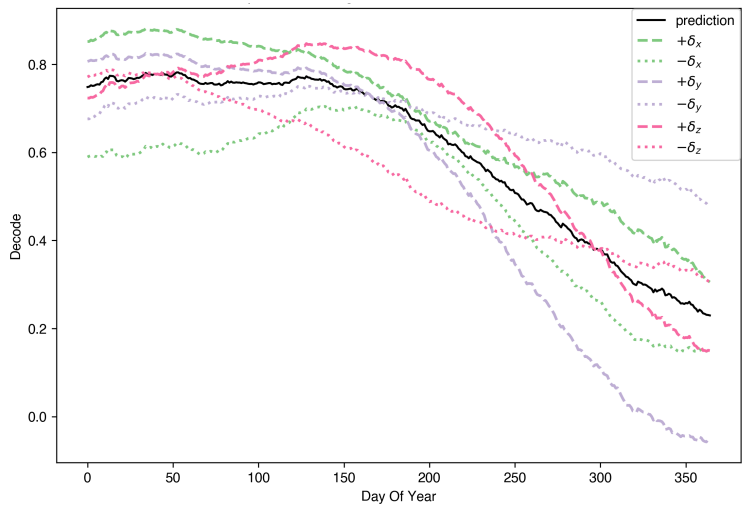


(d) Decoding Standardization.

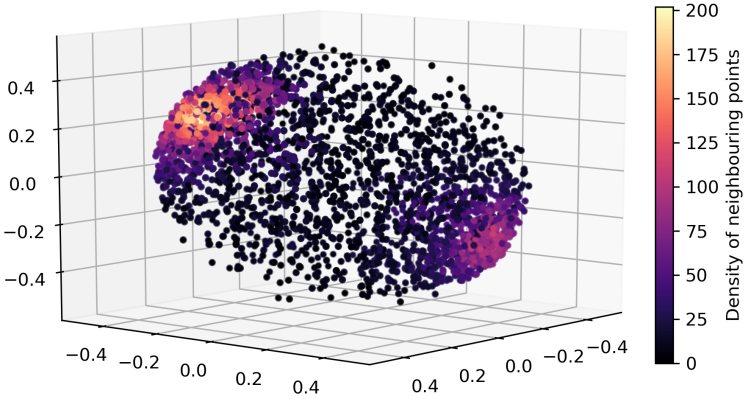
Time Series: Latent Space — $k = 3$



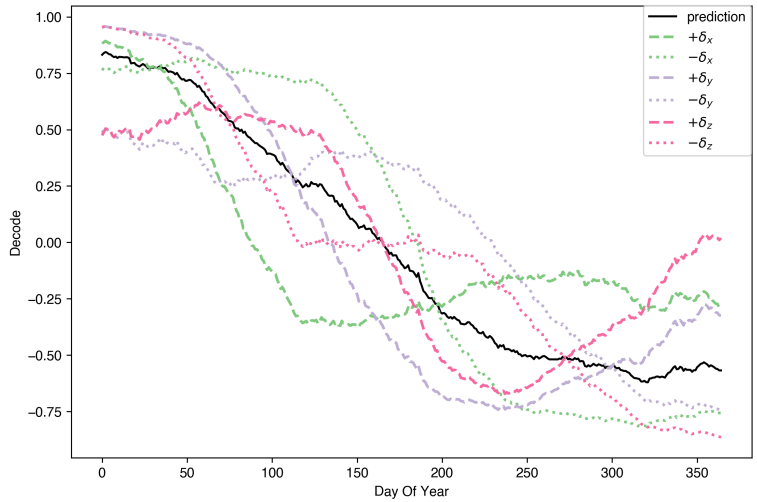
(a) Latent Space Normalization.



(b) Decoding Normalization.



(c) Latent Space Standardization.



(d) Decoding Standardization.

Contents

1 Introduction

2 Feed-Forward Neural Networks

3 Supervised Learning

4 Unsupervised Learning

5 Interpretability

6 Uncertainty

7 Conclusions

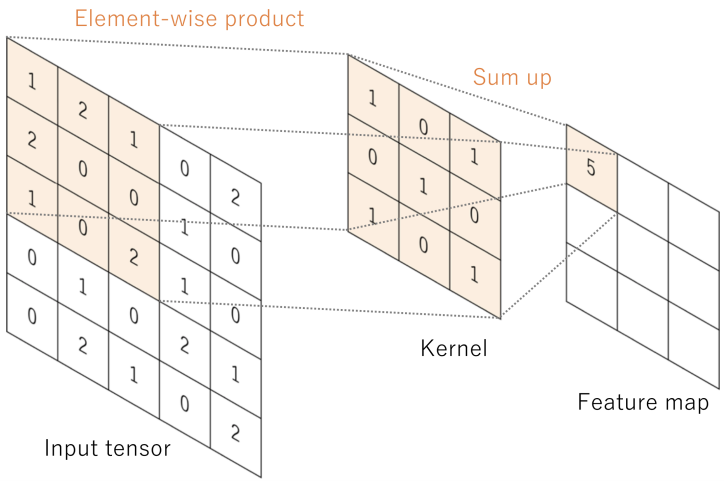
Interpretability

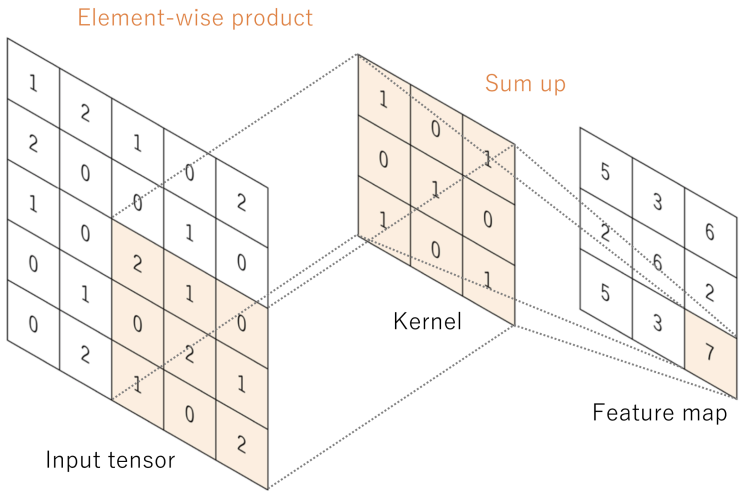
- Interpretability is the degree to which a human can comprehend the cause of a decision.
- The specific reason why a neural network makes a decision is unknown.
- This lack of knowledge turns this technology into a black-box.
- Interpretability algorithms attempt to solve this problem.

Convolutional Neural Networks

- For processing data that has a grid-like topology, like images.
- Based on the convolution operation. For discrete values:

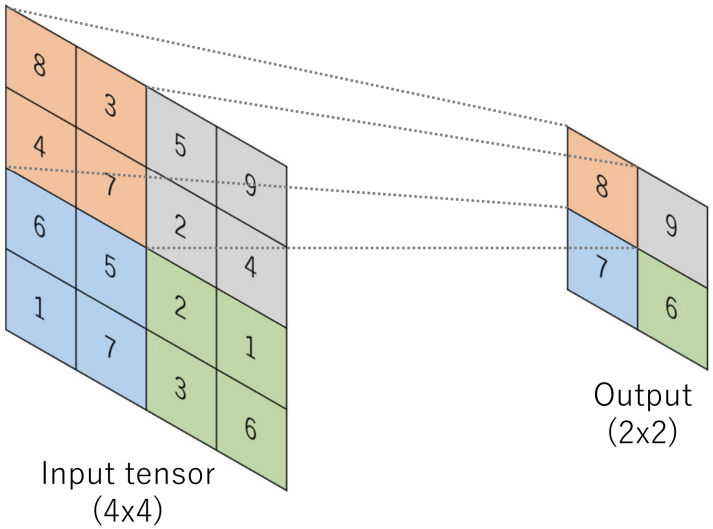
$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$





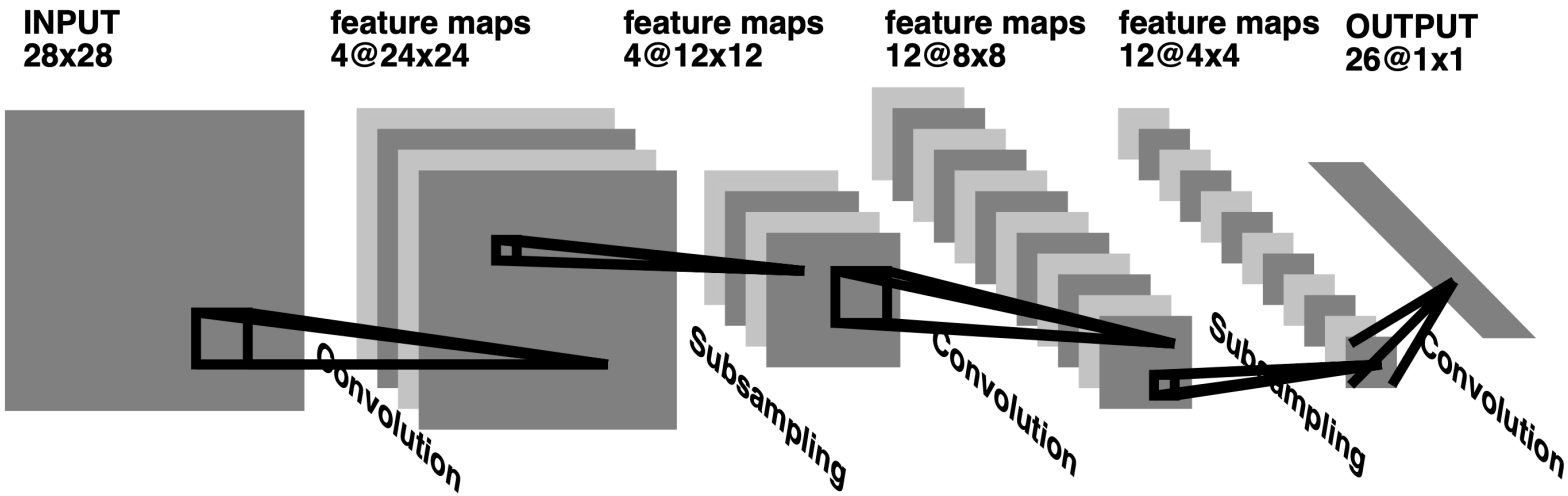
Convolutional Neural Networks

- After a convolutional layer, we have a pooling layer.
- Max-pooling and global average-pooling.



Convolutional Neural Networks

- The model learns the parameters of the kernels of each layer.
- Good at finding patrons and edges in pictures.



Interpretability Algorithms

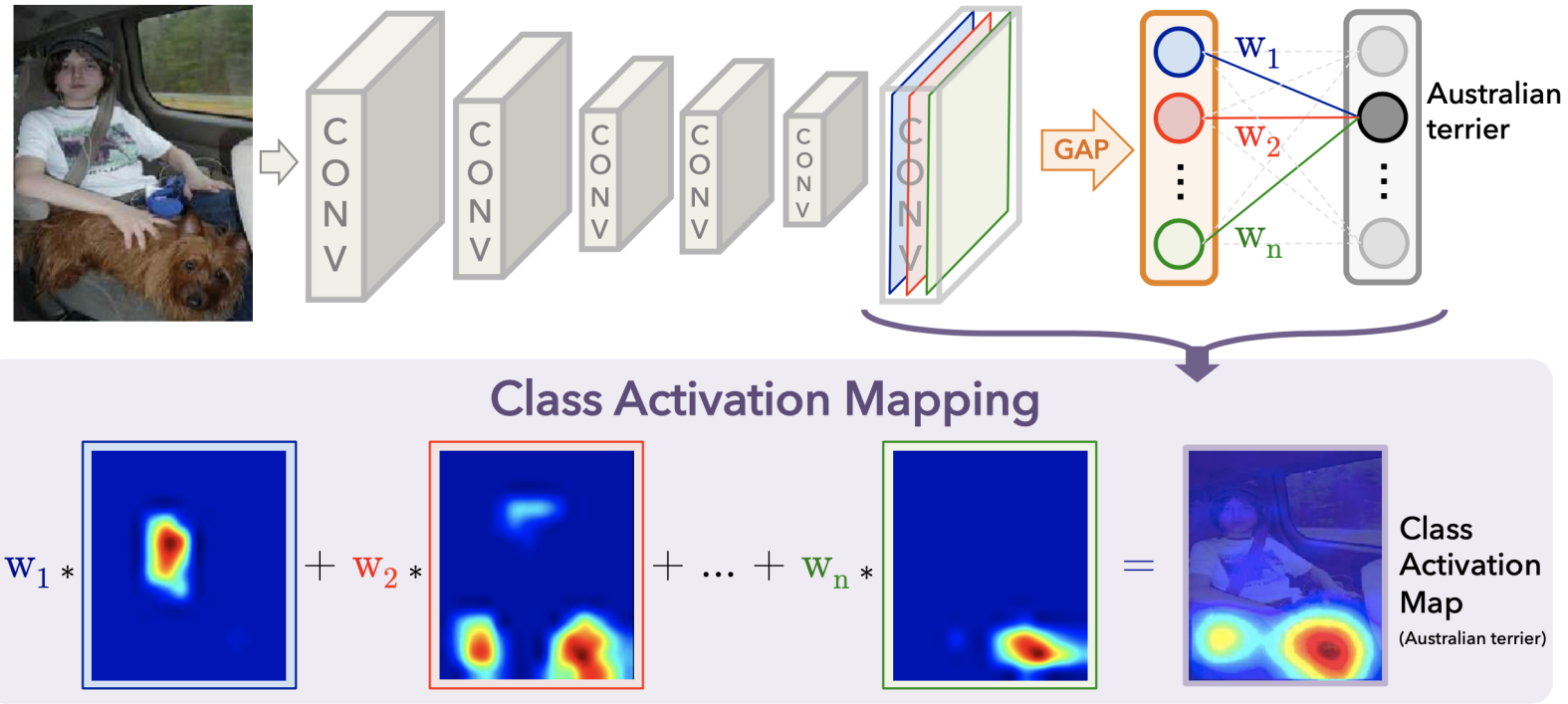
- Different contributions: features and input data, neuron i de layer.
- Baseline: Represents lack of information.
- Two axioms:
  - Implementation Invariance.
  - Sensitivity.
- Backpropagation. Integrated Gradients:

$$(x_i - x_i') \times \int_0^1 \frac{\partial F\left(x' + \alpha(x - x')\right)}{\partial x_i} d\alpha$$

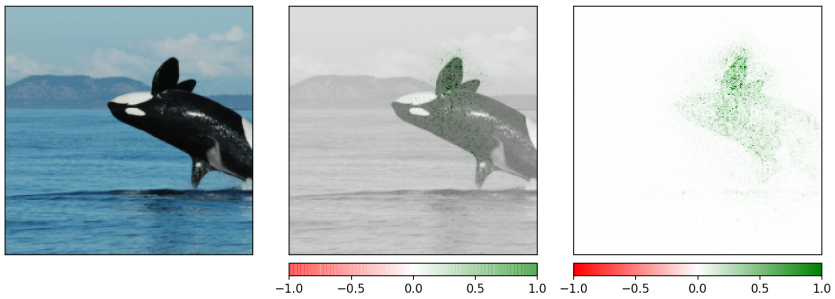
- Input Modification: Occlusion.

Interpretability Algorithms

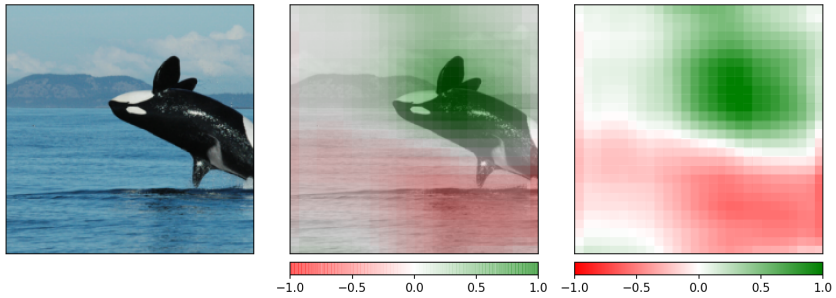
- Attention: Class Activation Mapping.



Interpretability Algorithms



(a) Integrated Gradients.



(b) Occlusion.

(c) Class Activation Mapping.

Contents

1 Introduction

2 Feed-Forward Neural Networks

3 Supervised Learning

4 Unsupervised Learning

5 Interpretability

6 Uncertainty

7 Conclusions

Uncertainty

- A prediction without the uncertaintyó might become useless.
- Example: sales forecasting.
- We present two methods for adding error to regression predictions:
  - NN returns both the regression and the uncertainty.
  - NN that adds uncertainty to a given regression.

NormalLoss

- Given a normal distribution $\mathcal{N}(\mu, \sigma^2)$, the probability density function is:

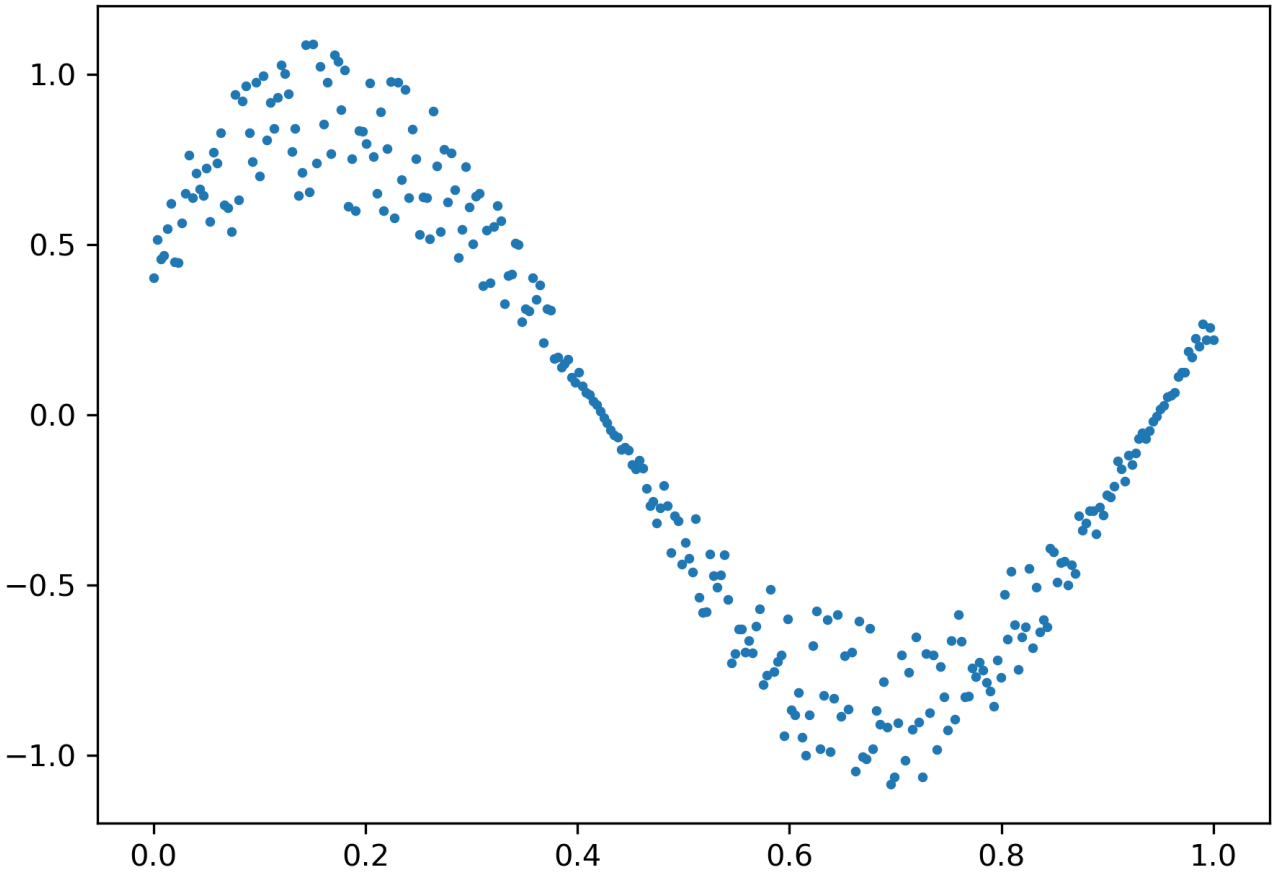
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Indicates how likely a point is to be in this distribution.
- It can be transformed into a loss function:

$$J(x, \mu, \sigma) = \frac{1}{N} \sum_{i=1}^{N} \left(log(\sigma_i) + \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right)$$

on $x = \{x_i\}_{i=1,\ldots,N}$, $\mu = \{\mu_i\}_{i=1,\ldots,N}$ i $\sigma = \{\sigma_i\}_{i=1,\ldots,N}$.

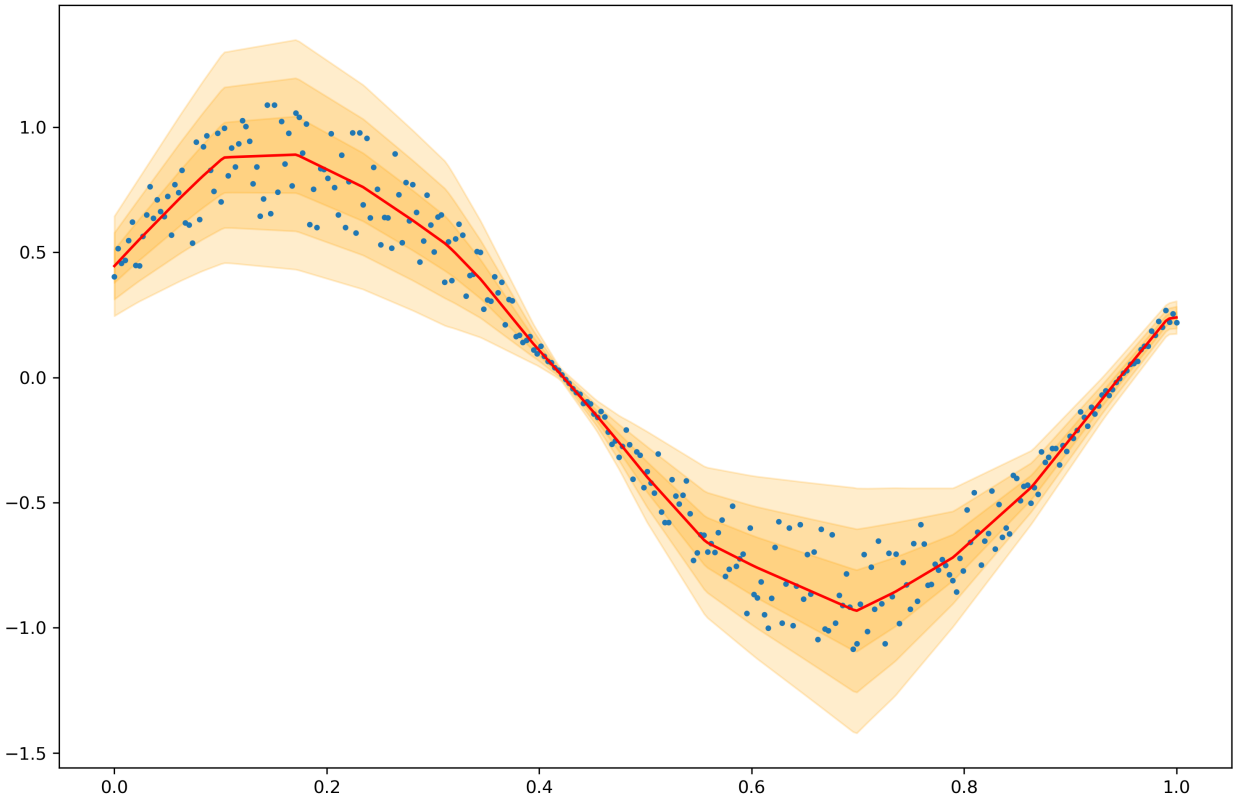
Regression Problem



Case 1: Regression Error

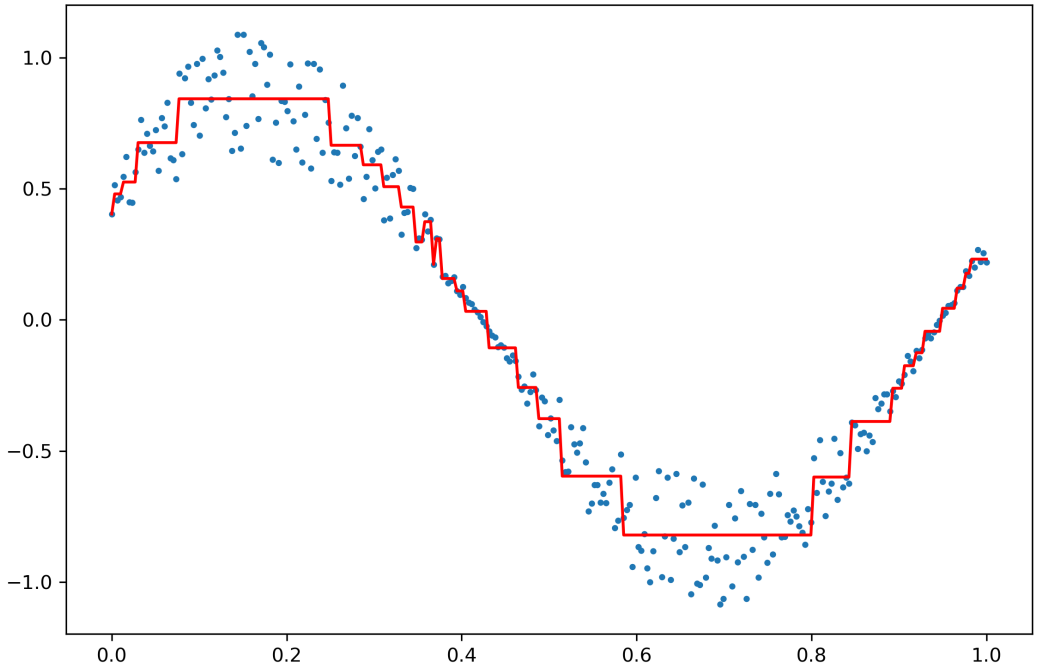
- NN with two hidden layers and 128 neurons each.
- NormalLoss as the loss function.
- Given x, we want to fit the best distribution $\mathcal{N}(\mu, \sigma^2)$.

Case 1: Regression Error



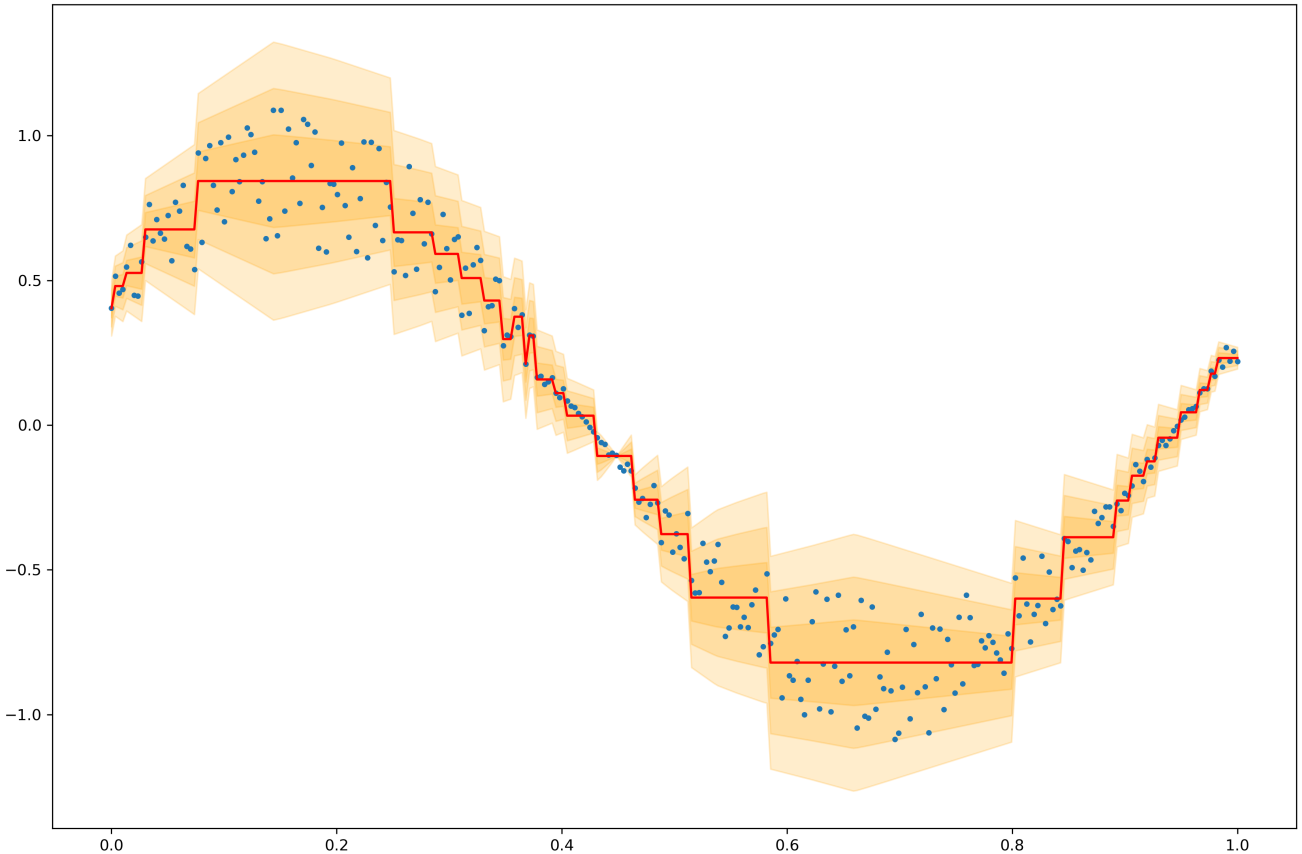
Case 2: Regressor

- Considere a decision tree with max depth 5.
- We will have, at most, $2^5 = 32$ different values.

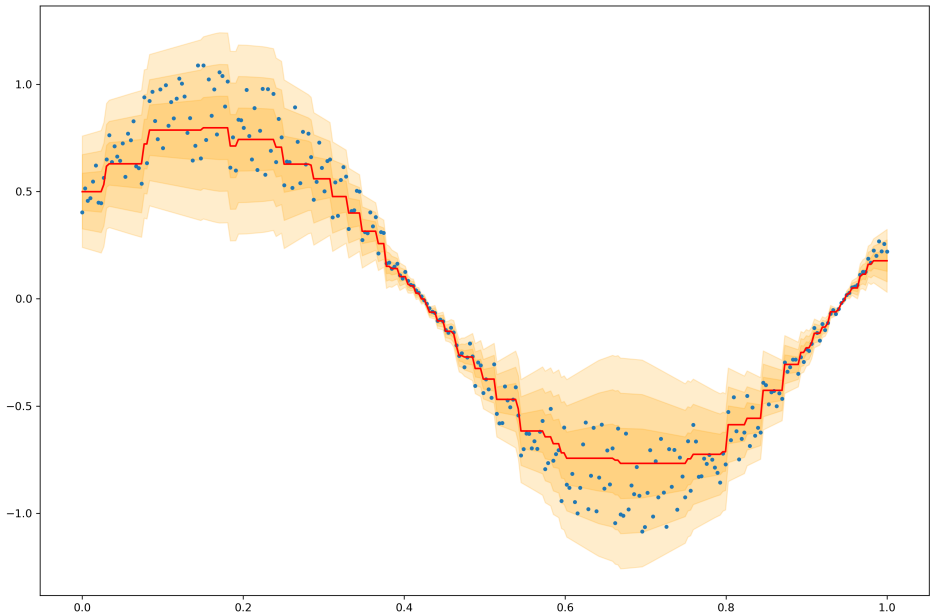


Case 2: Neural Network

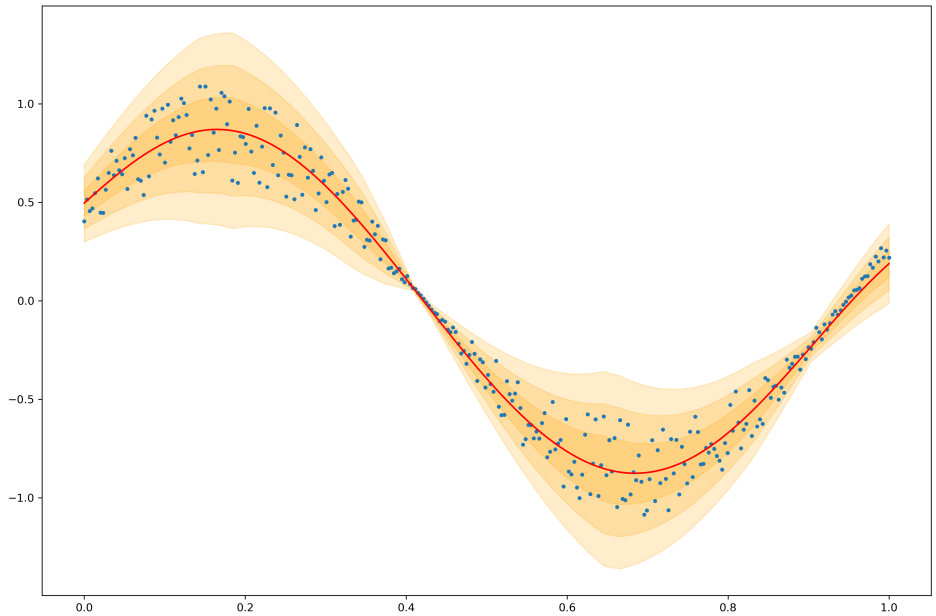
- NN with two hidden layers and 64 neurons each.
- NormalLoss as the loss function. Now the μ values are computed using the regressor.



Case 2: Different Regressors



(a) Gradient Boosting with 25 estimadors.



(b) SVR with a RBF kernel.

Uncertainty: Conclusions

Uncertainty Estimation for Black-Box Classification Models: A Use Case for Sentiment Analysis

José Mena[1,3(✉)], Axel Brando[2,3], Oriol Pujol[3], and Jordi Vitrià[3]

[1] Eurecat, Centre Tecnològic de Catalunya, Barcelona, Spain
jose.mena@eurecat.org
[2] BBVA Analytics Data & Analytics, Madrid, Spain
axel.brando@bbvadata.com
[3] Universitat de Barcelona, Barcelona, Spain
{axelbrando,oriol_pujol,jordi.vitria}@ub.edu

(a) September 2019.

Building uncertainty models on top of black-box predictive APIs

AXEL BRANDO[1,2], DAMIÀ TORRES[2], JOSE A. RODRÍGUEZ-SERRANO[2] AND JORDI VITRIÀ.[1]
[1]Departament de Matemàtiques i Informàtica, Universitat de Barcelona (UB) (e-mails: axelbrando@ub.edu, jordi.vitria@ub.edu)
[2]BBVA Data and Analytics, Spain, (e-mails: axel.brando@bbvadata.com, joseantonio.rodriguez.serrano@bbvadata.com)
Corresponding author:Axel Brando (e-mail: axelbrando@ub.edu).

(b) July 2020.

Contents

1. Introduction
2. Feed-Forward Neural Networks
3. Supervised Learning
4. Unsupervised Learning
5. Interpretability
6. Uncertainty
7. Conclusions

Conclusions

- Developed a mathematical description for feed-forward neural networks.
- Presented the supervised and unsupervised tyeps of learning.
- Analyzed applications to the fields of Interpretability and Uncertainty.
- Introduced methods to understand the good performance of NNs.
- Future research:
  - New interpretability methods.
  - Development of the field of uncertainty.
  - Analyze different architectures (GANs, reinforcement learning, etc.)
  - Applications in specific research fields.