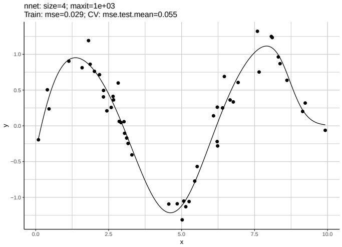
Introduction to Machine Learning

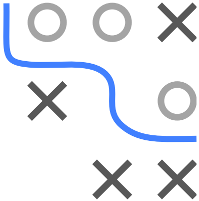
Neural Networks
Universal Approximation



Learning goals

- Universal approximation theorem for one-hidden-layer neural networks
- The pros and cons of a low approximation error

UNIVERSAL APPROXIMATION PROPERTY



Theorem. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous, non-constant, bounded, and monotonically increasing function. Let $C \subset \mathbb{R}^p$ be compact, and let $\mathcal{C}(C)$ denote the space of continuous functions $C \to \mathbb{R}$. Then, given a function $g \in \mathcal{C}(C)$ and an accuracy $\varepsilon > 0$, there exists a hidden layer size $m \in \mathbb{N}$ and a set of coefficients $W_j \in \mathbb{R}^p$, $u_j, b_j \in \mathbb{R}$ (for $j \in \{1, \ldots, m\}$), such that

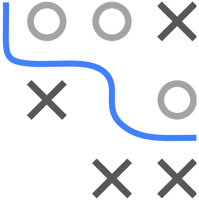
$$f : C \to \mathbb{R}\,; \quad f(\mathbf{x}) = \sum_{j=1}^{m} u_j \cdot \sigma\left(W_j^T \mathbf{x} + b_j\right)$$

is an ε-approximation of g, that is,

$$\|f - g\|_\infty := \max_{x \in C} |f(\mathbf{x}) - g(\mathbf{x})| < \varepsilon \ .$$

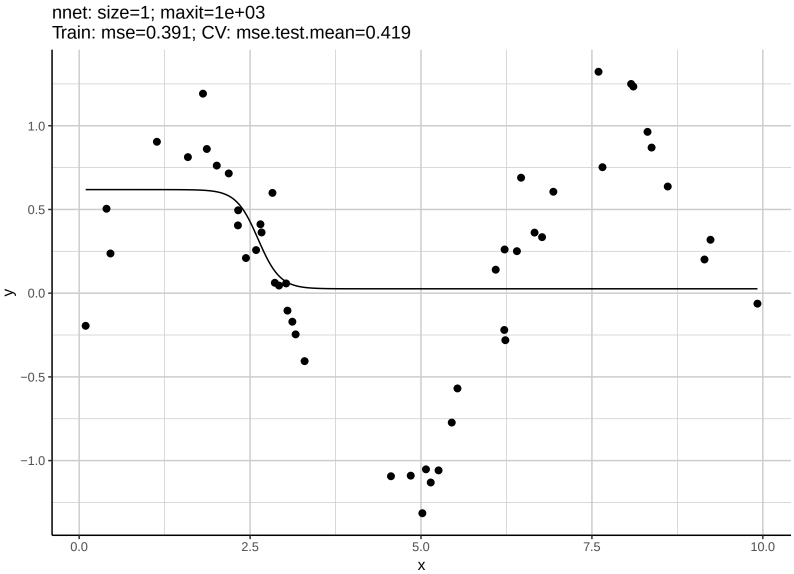
The theorem extends trivially to multiple outputs.

EXAMPLE : REGRESSION/CLASSIFICATION



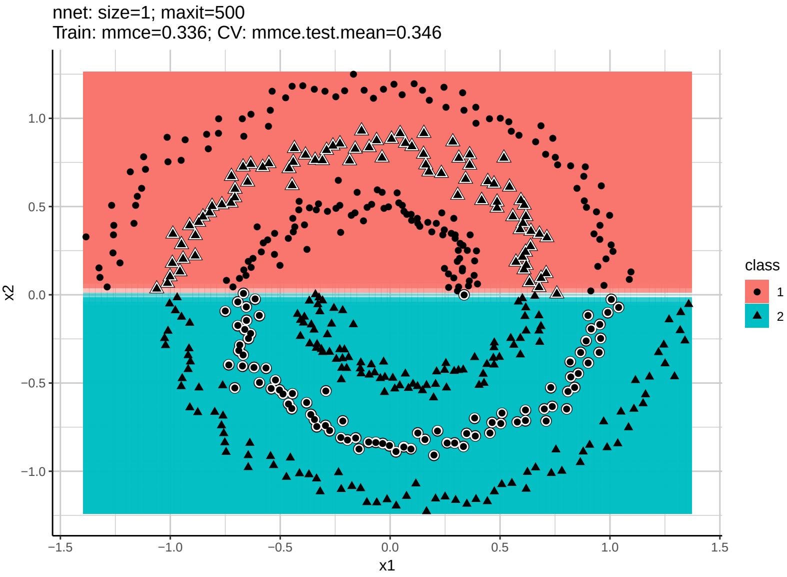
- Let’s look at a few examples of the types of functions and decisions boundaries learnt by neural networks (with a **single** hidden layer) of various sizes.
- "size" here refers to the number of neurons in the hidden layer.
- The number of "iterations" in the following slides corresponds to the number of steps of the applied iterative optimization algorithm (stochastic gradient descent).

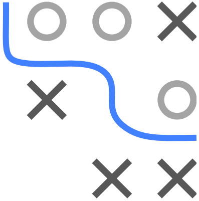
REGRESSION EX.: 1000 TRAINING ITERATIONS



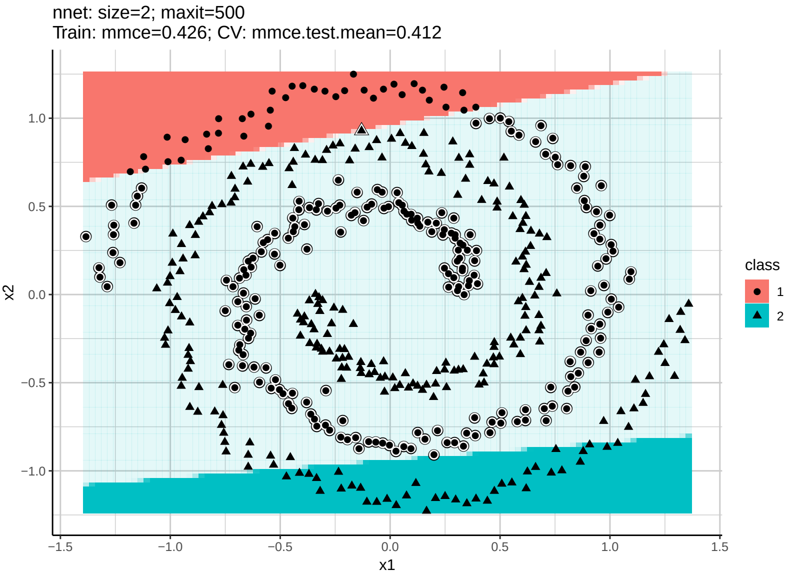
nnet: size=1; maxit=1e+03
Train: mse=0.391; CV: mse.test.mean=0.419

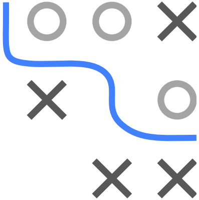
CLASSIFICATION: 500 TRAINING ITERATIONS



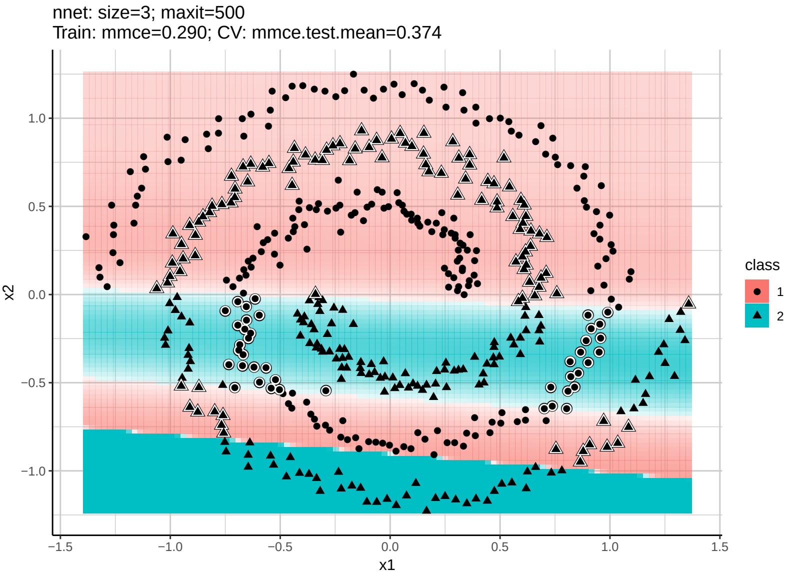


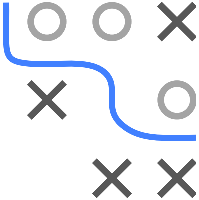
CLASSIFICATION: 500 TRAINING ITERATIONS



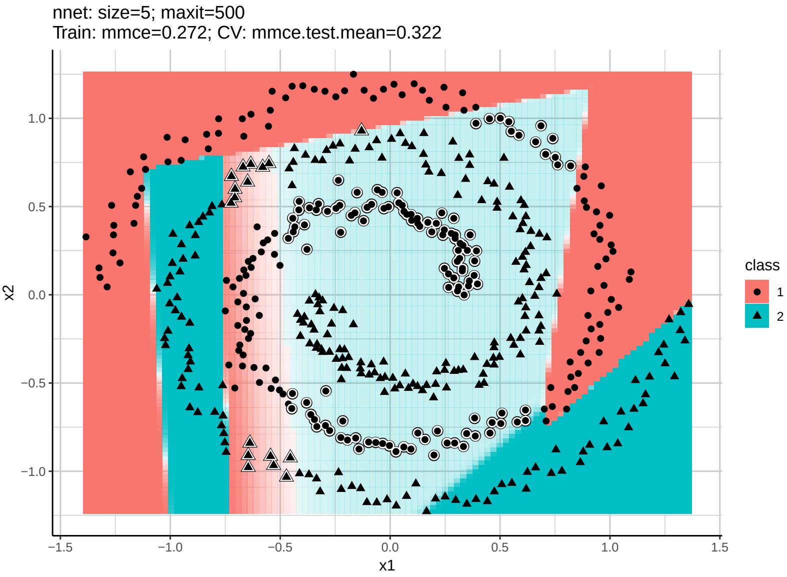


CLASSIFICATION: 500 TRAINING ITERATIONS

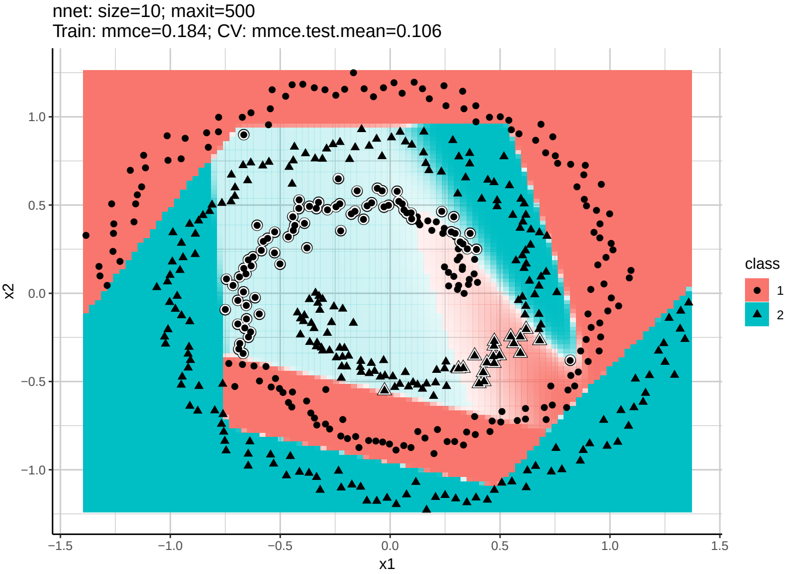




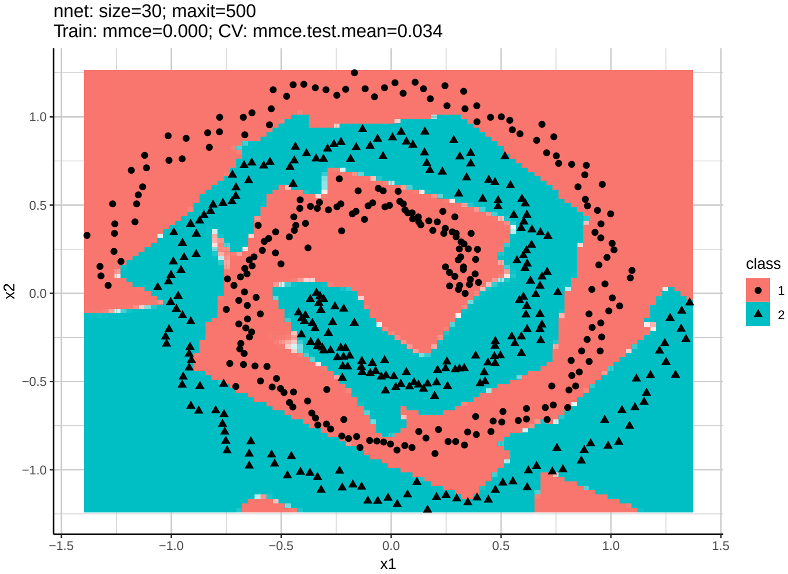
CLASSIFICATION: 500 TRAINING ITERATIONS

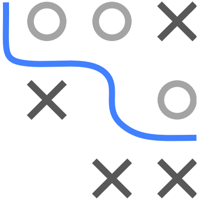


CLASSIFICATION: 500 TRAINING ITERATIONS

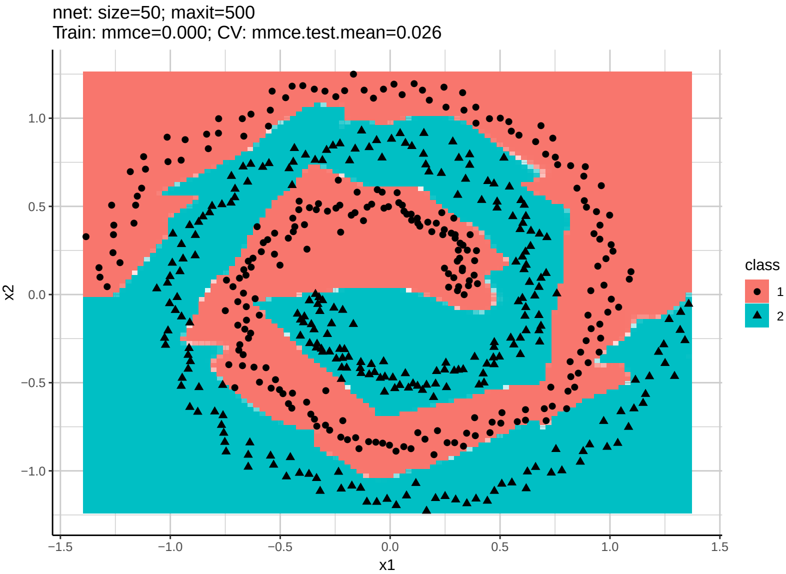


CLASSIFICATION: 500 TRAINING ITERATIONS





CLASSIFICATION: 500 TRAINING ITERATIONS


nnet: size=50; maxit=500
Train: mmce=0.000; CV: mmce.test.mean=0.026
x2
x1
class