Deep Learning

Exercise 3: Universal Function Approximator

Room: **BIN-1-B.01**

Instructor: Manuel Günther
Email: guenther@ifi.uzh.ch
Office: AND 2.54

Friday, March 11, 2022

Outline

1. Universal Function Approximator

Outline

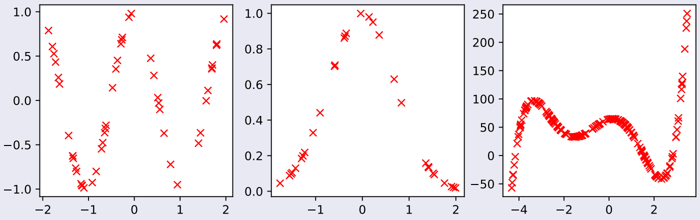
1 Universal Function Approximator

Universal Function Approximator

Goal of Exercise

- Implement a universal function approximator
  - → Two-layer network with logistic function as nonlinearity
- Try out how good it works for three functions
  - → Learn from samples of three different functions

Data Samples of Three Functions



Universal Function Approximator

Task 1: Network

- Implement 2-layer network with one input x and one output y
  - $\rightarrow$ Two fully-connected layers with weights $\Theta = (\mathbf{W}^{(1)}, \vec{w}^{(2)})$
  - $\rightarrow$ Variable number of hidden nodes K with logistic activation function

Test 1: Test Network Outcome

- What should be the output for $\mathbf{W}^{(1)} = \mathbf{0}$ and $\vec{w}^{(2)} = \vec{1}$
  - $\rightarrow$ For a given number of hidden neurons K
  - $\rightarrow$ For any data point $\vec{x}$

Universal Function Approximator

Task 2: Gradient Implementation

- Take loss $\mathcal{J}^{L_2}$ over dataset
- Implement function to return the gradient for a given dataset
  - $\rightarrow$ Split gradient into $\nabla_{\mathbf{W}^{(1)}}$ and $\nabla_{\vec{w}^{(2)}}$

Task 3: Gradient Descent

- Implement function to perform gradient descent
  - $\rightarrow$ Dataset is of form $\{(\vec{x}^{[n]}, t^{[n]}) \mid n \leq 1 \leq N\}$
  - $\rightarrow$ Return optimized parameters

Universal Function Approximator

Task 4: Datasets

- Create different training data
  - $\rightarrow$ N random values x in the proposed range
  - $\rightarrow$ Remember to add $x_0 = 1$ dimension
- Store as $X = \{(\vec{x}^{[n]}, t^{[n]}) \mid n \leq 1 \leq N\}$

Cosine

$$t = \cos(3x)$$

$$x \in [-2, 2]$$

Gaussian

$$t = e^{-x^2}$$

$$x \in [-2, 2]$$

Polynomial

$$t = x^5 + 3x^4 - 11x^3 - 27x^2 + 10x + 64$$

$$x \in [-4.5, 3.5]$$

Universal Function Approximator

Task 5: Define Parameters

- What is the appropriate number of hidden units for each function?

Task 6: Parameter Initialization

- Initialize Θ_1, Θ_2, Θ_3 for different functions/datasets
  $\rightarrow$ Take weights randomly from $[-1, 1]$

Task 7: Perform Gradient Descent

- Optimize the parameters Θ_i with according $X_i, i \in \{1, 2, 3\}$

Universal Function Approximator

Task 8/9: Implement and Call Plotting

- Plot data points with “x”
- Plot approximated functions in range $R = (x_{min}, x_{max})$
- Call plotting function three times

Example Results