Deep Learning

Exercise 2: Gradient Descent

Room: **BIN-1-B.01**
Instructor: Manuel Günther
Email: guenther@ifi.uzh.ch
Office: AND 2.54

Friday, March 4, 2022

Outline

Outline

Gradient Descent

Goal of the Exercise

- Implement gradient descent for a pre-defined loss function
- Get to know the difficulties of gradient descent

Loss Function

The loss function is given as:

$$\mathcal{J}_{\vec{w}} = w_0^2 + w_1^2 + 20 \cdot \sin(w_0) \cdot \cos(w_1)$$

Task 1: Gradient

- Analytically compute the gradient for the loss.

Task 2: Loss Function

- Implement the loss function with `numpy`.

Gradient Descent

Task 3: Gradient Implementation

- Implement the gradient function from Task 1 with `numpy`.

Test 1: Sanity Check

- Test the implementations with $\vec{w} = (0, 0)^{\mathrm{T}}$.
  $\rightarrow$ What is the expected outcome?

Task 4: Stopping Criterion

- Define an appropriate stopping criterion for the gradient descent.

Task 5: Gradient Descent Implementation

- Implement gradient descent with that stopping criterion.

Gradient Descent

Task 6: Run Gradient Descent

- Run the above function 1000 times.
  → Use different initial weights $\vec{w} \in [-20, 20]^2$.
- Store the resulting optimized weight vectors.

Test 2: Count Minima (Difficult Task)

- Group the weight vectors and count the number of groups.
- Compare this with the theoretical maximum

Task 7: Global Minimum

- Find the weight vector that represents the global minimum.
- Compute the loss for this weight vector.

Gradient Descent

Task 8: Surface Plot

- Create 3D surface plot of loss.
- Limit range $\vec{w} \in [-10, 10]^2$.
- Plot minima as points.

  $\rightarrow$ The minima found in task 6.

Sample Surface Plot