Deep Learning

Exercise 3: Non-Linear Regression via Gradient Descent

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

Friday, March 12, 2020

Outline

Outline

Non-Linear Regression via Gradient Descent

Task

1. Implement 2-layer network with one input x and one output y
   $\rightarrow$ Variable number of hidden nodes K with logistic activation function
2. Implement a function to compute the loss for a given training set
3. Implement a function to compute the gradient for given training set
   $\rightarrow$ Split gradient into $\nabla_{\mathbf{W}^{(1)}}$ and $\nabla_{\vec{w}^{(2)}}$
4. Implement a function for iterative gradient descent
5. Create different training data (next page)
6. Run gradient descent with appropriate K, η and number of epochs
7. Plot training data and approximated function together in one plot
8. Plot loss progression into another plot

Non-Linear Regression via Gradient Descent

Cosine

$$t = \frac{\cos(3x) + 1}{2}$$

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

Gaussian

$$t = e^{-\frac{1}{4}x^2}$$

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

Polynomial

$$t = \frac{x^5 + 3x^4 - 11x^3 - 27x^2 + 10x + 64}{100}$$

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

Sub-Tasks

- Create random x values inside the proposed range
  $\rightarrow$ Vary number of samples between 20 and 1000
- Run each training several times. What can you discover?
- What is the appropriate number of hidden units for each task?

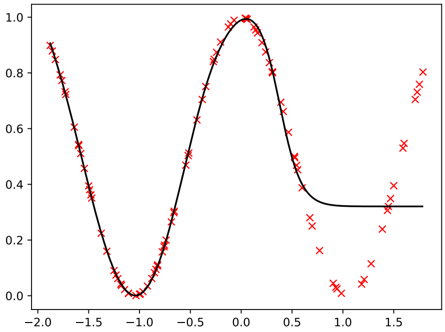
Non-Linear Regression via Gradient Descent

Recall: How to Compute the Gradient

$$\nabla_{\Theta}(\mathcal{J}) = \left(\begin{array}{ccl} \frac{\partial \mathcal{J}}{\partial w^{(1)}_{1,0}} & = & \frac{2}{N}\sum_{n=1}^{N}\left(y^{[n]} - t^{[n]}\right) w^{(2)}_1\, h^{[n]}_1 (1 - h^{[n]}_1)\, x^{[n]}_0 \\ \vdots & & \vdots \\ \frac{\partial \mathcal{J}}{\partial w^{(1)}_{D,K}} & = & \frac{2}{N}\sum_{n=1}^{N}\left(y^{[n]} - t^{[n]}\right) w^{(2)}_K\, h^{[n]}_K (1 - h^{[n]}_K)\, x^{[n]}_D \\ \frac{\partial \mathcal{J}}{\partial w^{(2)}_{0}} & = & \frac{2}{N}\sum_{n=1}^{N}\left(y^{[n]} - t^{[n]}\right) h_0 \\ \vdots & & \vdots \\ \frac{\partial \mathcal{J}}{\partial w^{(2)}_{K}} & = & \frac{2}{N}\sum_{n=1}^{N}\left(y^{[n]} - t^{[n]}\right) h_K \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} \nabla_{\mathbf{W}^{(1)}} \\ \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} \nabla_{\vec{w}^{(2)}} \end{array}$$

Non-Linear Regression via Gradient Descent

Approximation of Cosine



Loss Progression