

Praktisches Maschinelles Lernen am Beispiel des Gradientenverfahrens

Billy Joe Franks

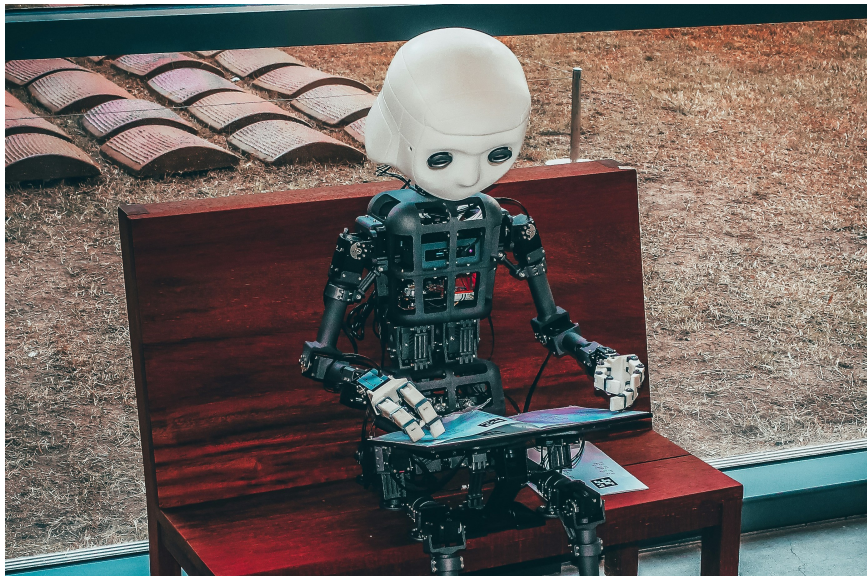
Neustadt, 07.06.2024

Über mich.

Karriere

- ▶ Bachelor Informatik (Schwerpunkt Maschinelles Lernen)
- ▶ Master Informatik (Schwerpunkt Maschinelles Lernen)
- ▶ PhD Student (Schwerpunkt Maschinelles Lernen)

Machinelles Lernen





Section Navigation

An introduction to machine learning with
scikit-learn

A tutorial on statistical-learning for
scientific data processing

Working With Text Data

Choosing the right estimator

External Resources, Videos and Talks

> scikit-learn...

scikit-learn Tutorials

An introduction to machine learning with scikit-learn

[Machine learning: the problem setting](#)

[Loading an example dataset](#)

[Learning and predicting](#)

[Conventions](#)

A tutorial on statistical-learning for scientific data processing

[Statistical learning: the setting and the estimator object in scikit-learn](#)

[Supervised learning: predicting an output variable from high-dimensional observations](#)

[Model selection: choosing estimators and their parameters](#)

[Unsupervised learning: seeking representations of the data](#)

[Putting it all together](#)

Working With Text Data

Tutorial setup

Machine Learning

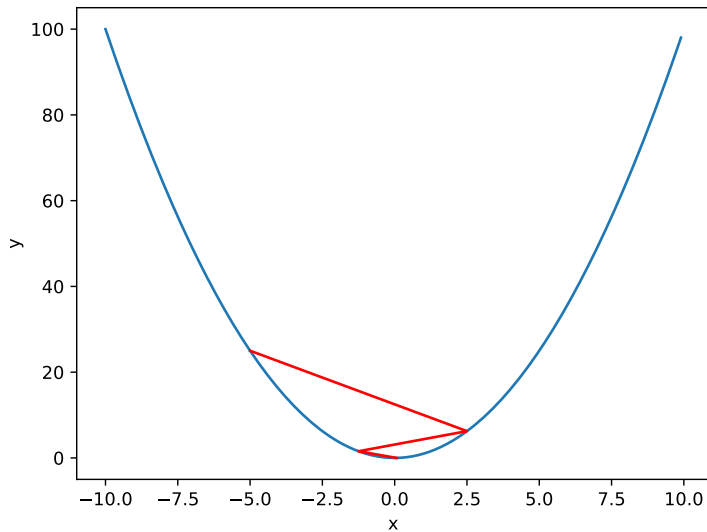
Linear Soft-Margin Support Vector Machine

$$\begin{aligned} \max_{\gamma, b \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^d \setminus \{0\}, \xi_1, \dots, \xi_n \geq 0} \quad & \gamma - C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i \frac{(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|} \geq \gamma - \xi_i, \quad \forall i = 1, \dots, n \end{aligned}$$

Kernel Soft-Margin Support Vector Machine

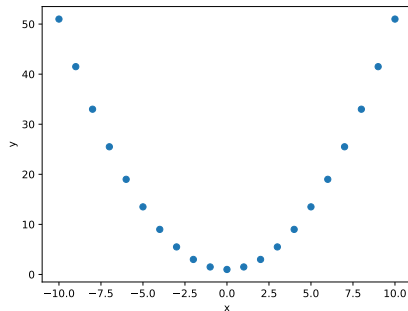
$$\min_{b \in \mathbb{R}, \alpha \in \mathbb{R}^n} \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j) + C \sum_{i=1}^n \max(0, 1 - y_i (\sum_{j=1}^n \alpha_j k(x_i, x_j) + b))$$

Gradientenverfahren

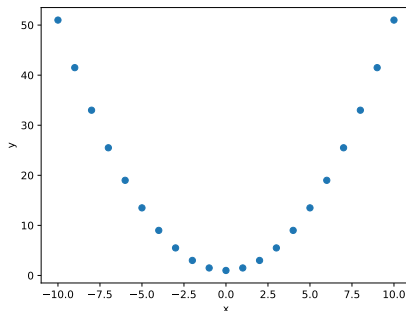


Fehlerminimierung

Fehlerminimierung



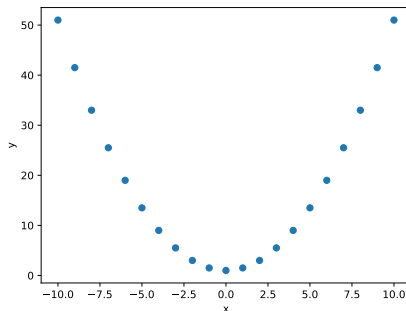
Fehlerminimierung



Methode der kleinsten Quadrate (Least Squares)

Angenommen $D := \{(x_1, y_1), \dots, (x_n, y_n)\}$ mit $\forall i \in [n] : x_i, y_i \in \mathbb{R}$
und $g_\theta : \mathbb{R} \rightarrow \mathbb{R}$ mit Parametern θ , $g_\theta(x) := \theta_1 x^2 + \theta_2$.

Fehlerminimierung

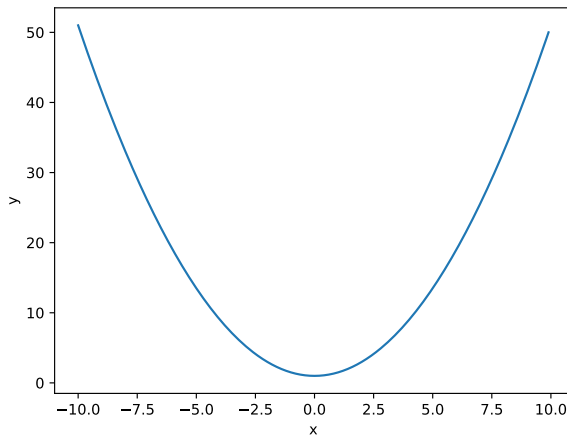


Methode der kleinsten Quadrate (Least Squares)

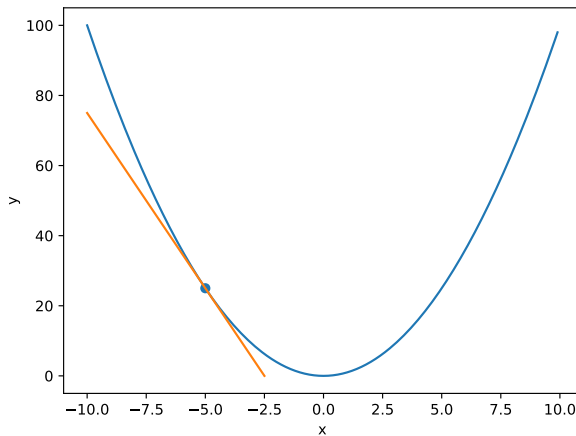
Angenommen $D := \{(x_1, y_1), \dots, (x_n, y_n)\}$ mit $\forall i \in [n] : x_i, y_i \in \mathbb{R}$ und $g_\theta : \mathbb{R} \rightarrow \mathbb{R}$ mit Parametern θ , $g_\theta(x) := \theta_1 x^2 + \theta_2$.

$$\theta^* := \arg \min_{\theta} \sum_{(x,y) \in D} (g_\theta(x) - y)^2$$

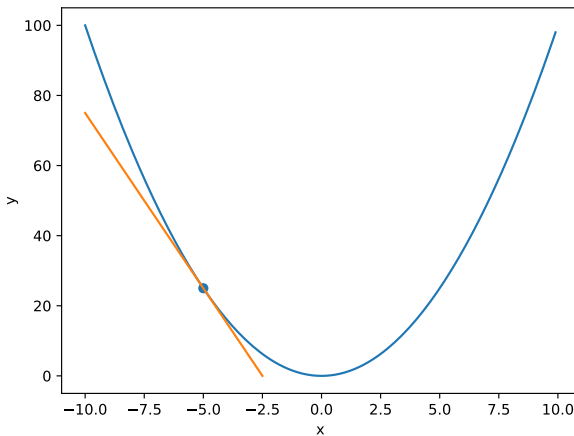
Gradientenverfahren



Gradientenverfahren

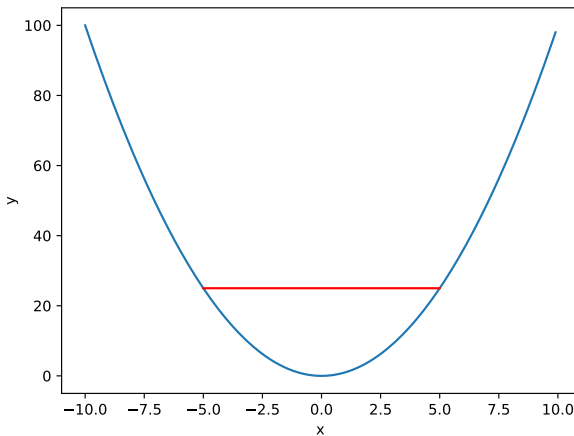


Gradientenverfahren



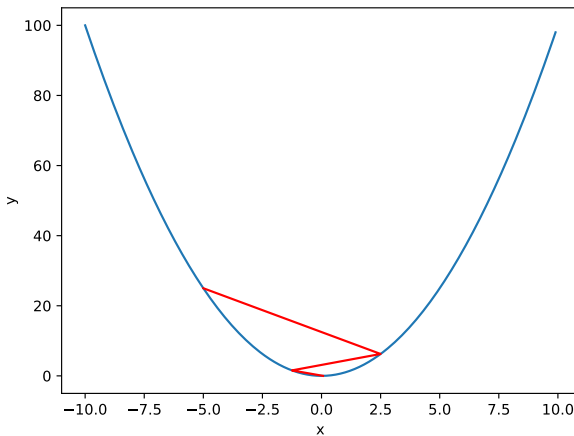
$$x \leftarrow x - \frac{\partial x^2}{\partial x}$$

Gradientenverfahren



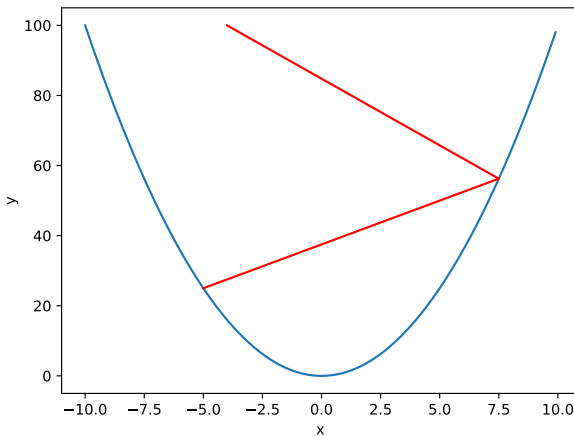
$$x \leftarrow x - \frac{\partial x^2}{\partial x}$$

Gradientenverfahren



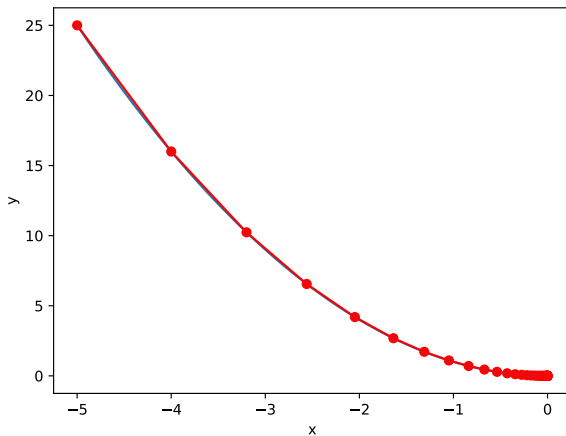
$$x \leftarrow x - \frac{3}{4} \frac{\partial x^2}{\partial x}$$

Gradientenverfahren



$$x \leftarrow x - \frac{5}{4} \frac{\partial x^2}{\partial x}$$

Gradientenverfahren



$$x \leftarrow x - \lambda \frac{\partial x^2}{\partial x}, \lambda = 0.1$$

Gradientenverfahren

Gradient Descent

Input: Initialization θ_0 , function f , iterations T , learning rate λ

$\theta \leftarrow \theta_0$

for $i \in [T]$ **do**

$\theta \leftarrow \theta - \lambda \frac{\partial f(\theta)}{\partial \theta}$

end for

return θ

Gradientenverfahren

Gradient Descent

Input: Initialization θ_0 , function f , iterations T , learning rate λ

$\theta \leftarrow \theta_0$

for $i \in [T]$ **do**

$\theta \leftarrow \theta - \lambda \frac{\partial f(\theta)}{\partial \theta}$

end for

return θ

Least Squares

$$\theta^* := \arg \min_{\theta} \sum_{(x,y) \in D} (g_{\theta}(x) - y)^2$$

$$\implies f(\theta) := \sum_{(x,y) \in D} (g_{\theta}(x) - y)^2$$

Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

```

1: given  $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$ 
2: initialize time step  $t \leftarrow 0$ , parameter vector  $\theta_{t=0} \in \mathbb{R}^n$ , first moment vector  $\mathbf{m}_{t=0} \leftarrow \mathbf{0}$ , second moment
   vector  $\mathbf{v}_{t=0} \leftarrow \mathbf{0}$ , schedule multiplier  $\eta_{t=0} \in \mathbb{R}$ 
3: repeat
4:    $t \leftarrow t + 1$ 
5:    $\nabla f_t(\theta_{t-1}) \leftarrow \text{SelectBatch}(\theta_{t-1})$  ▷ select batch and return the corresponding gradient
6:    $\mathbf{g}_t \leftarrow \nabla f_t(\theta_{t-1}) + \lambda \theta_{t-1}$ 
7:    $\mathbf{m}_t \leftarrow \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t$  ▷ here and below all operations are element-wise
8:    $\mathbf{v}_t \leftarrow \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2$ 
9:    $\hat{\mathbf{m}}_t \leftarrow \mathbf{m}_t / (1 - \beta_1^t)$  ▷  $\beta_1$  is taken to the power of  $t$ 
10:   $\hat{\mathbf{v}}_t \leftarrow \mathbf{v}_t / (1 - \beta_2^t)$  ▷  $\beta_2$  is taken to the power of  $t$ 
11:   $\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$  ▷ can be fixed, decay, or also be used for warm restarts
12:   $\theta_t \leftarrow \theta_{t-1} - \eta_t \left( \alpha \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \epsilon) + \lambda \theta_{t-1} \right)$ 
13: until stopping criterion is met
14: return optimized parameters  $\theta_t$ 

```

Ihr seid dran!

Gradient Descent

Input: Initialization θ_0 , function f , iterations T , learning rate λ

```
 $\theta \leftarrow \theta_0$   
for  $i \in [T]$  do  
     $\theta \leftarrow \theta - \lambda \frac{\partial f(\theta)}{\partial \theta}$   
end for  
return  $\theta$ 
```

Linear Least Squares

$$\theta^* := \arg \min_{\theta} \sum_{(x,y) \in D} (ax - y)^2 \implies f(a) := \sum_{(x,y) \in D} (ax - y)^2$$

$$\frac{\partial f(a)}{\partial a} = \sum_{(x,y) \in D} 2(ax - y)a, \quad a \leftarrow a - \lambda \sum_{(x,y) \in D} 2(ax - y)a$$