

Contents

- 1 Machine Learning 7**
 - 1 Machine Learning 7
 - 1.1 Support Vector Machines 7
 - 2 Neuronale Netze 8
 - 2.1 Einführung 8
 - 2.2 SVMs als Neuronale Netze 9
 - 2.3 Lineare Regression 9
 - 2.4 Deep Learning 9
 - 2.5 Convolutional Neuronal Networks 9

Machine Learning

SVMs und Neuronale Netzwerke

1 Machine Learning

Machine Learning ist ein Konzept, bei dem der Computer durch Algorithmen in der Lage ist Probleme selbstständig zu lösen. Dabei erkennt der Computer nach einer gewissen Lernphase Gesetzmässigkeiten, durch die er dann in der Lage ist, neu, ihm unbekannte Datensätze bezüglich eines bestimmten Problems zu lösen.

Dass der Computer die Gesetzmässigkeiten erkennen kann, werden verschiedene Systeme genutzt.

1.1 Support Vector Machines

Support Vector Machines oder kurz SVMs sind ein Konzept zum Lösen von Machine Learning Problemen. Dabei werden für alle Datensätze zuerst gewisse Features festgelegt. Diese Features entsprechen den Attributen des Datensatzes. Je nach Problem kann es unterschiedlich viele Features geben, doch allgemein kann man sagen, dass, je mehr Features es sind die Genauigkeit des Programms genauer wird.

Diese Features werden normalerweise in einem Vektor für den entsprechenden Datensatz gespeichert. Dieser Vektor wird x_i genannt wobei $i = 1, \dots, n$ den Datensatz beschreibt mit $x \in \mathbb{R}^m$. Zusätzlich hat jeder Datensatz noch einen weiteren Wert, welcher das "Ergebnis" des Datensatzes beschreibt, der $y_i \in \mathbb{R}$ oder auch "Label" genannt wird.

Die Features und das Label zusammen beschreiben einen Punkt im n-dimensionalen Koordinatensystem, wobei $n = m + 1$, da zu den m Dimensionen von x noch die eine Dimension von y dazukommt.

Durch die Features ist es dann möglich den Datensätzen einen bestimmten Ort im n-dimensionalen Koordinatensystem zuzuordnen. Diese können dann auf Grund ihrer räumlichen Anordnung klassifiziert werden. Dabei soll der Abstand von beiden Gruppen von Punkten zu dem Trennobjekt maximal sein. Dadurch kann die Klassifizierung optimal durchgeführt werden. Andernfalls würden schon geringe Abweichungen von den Test-Datensätzen reichen, das ein Datensatz falsch klassifiziert würde. (??)

Um die Trennlinie zu optimieren, gibt es auch eine mathematische Beschreibung. Da sich Ebenen

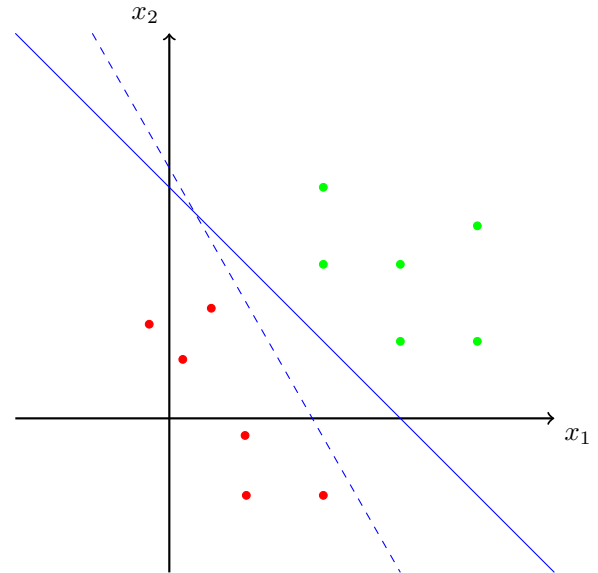


Abb. 1.1: Da der Abstand zwischen den unterschiedlich klassifizierten Datensätzen maximiert werden soll, gilt die durchgezogene und nicht die gestrichelte Linie als Trennelement.

auch in der Form $\langle x, w \rangle = b$ beschreiben lassen, folgt, dass wir die Datensätze auch etwas anders betrachten können (??). Das daraus folgende mathematische Programm sieht wie folgt aus:

$$\begin{aligned} &\text{maximiere}_{w,b} \quad \frac{2}{\|w\|} \\ &\text{sodass} \quad y_i(\langle w, x_i \rangle) \geq 1 \quad \forall i = 1, \dots, n; \quad x \in \mathbb{R}^n; \quad y \in \{1, -1\} \end{aligned}$$

Unter der Voraussetzung, dass alle vorhandenen Datensätze richtig klassifiziert sind. Dieses Problem kann man aber genauso schreiben als

$$\min_{w,b} \|w\| \quad \text{sodass} y_i(\langle w, x_i \rangle) \geq 1 \quad \forall i = 1, \dots, n; \quad x \in \mathbb{R}^n; \quad y \in \{1, -1\}$$

Klassifikation:

$$y_i = \{1 \text{ falls } \langle w, x_i \rangle - b \geq 1; -1 \text{ falls } \langle w, x_i \rangle - b \leq -1\}$$

Nachdem das Problem optimiert wurde, ist der Computer in der Lage, weitere Daten einzuordnen, vorausgesetzt, es ist richtig optimiert. Dennoch kann es bei dieser Art von SVMs zu einigen Problemen kommen.

1. Wenn die Datensätze nicht linear separierbar sind, das heisst, es ist nicht möglich die beiden Datensätze mit einem linearen Element zu trennen
2. Wenn die Daten nicht alle korrekt klassifiziert sind oder es Daten gibt, die in dem "Bereich" der anderen Seite liegen.

1.1.1 Soft Margin SVMs

Die Soft Margin SVMs können mit dem zweiten Problem umgehen. Sie sind im Prinzip wie herkömmliche SVMs aufgebaut, nur dass sie eine gewisse Fehlertoleranz besitzen. Diese kommt durch eine Veränderung an der Formel zustande:

$$\min_{w,b} \|w\| + C \sum_i z_i \forall i = 1, \dots, n$$

sodass

$$y_i(\langle w, x_i \rangle) \geq 1 \forall i = 1, \dots, n; x \in \mathbb{R}^n; y \in \{1, -1\} \text{ und } z_i \geq 0$$

Dabei steht z_i für die Grösse des Fehlers des Punktes x_i und C ist eine Konstante, deren Grösse bestimmt, wie stark die Fehler gewichtet werden, da die Konstante mit der Summe der Fehler multipliziert wird. Da die gesamte Gleichung aber minimiert werden soll sorgt ein hohes C dafür, dass das Problem schwerer optimiert werden kann. Die Konstante muss vom Mensch selbst gewählt werden, je nachdem, wie schwer Fehler gewichtet werden sollen.

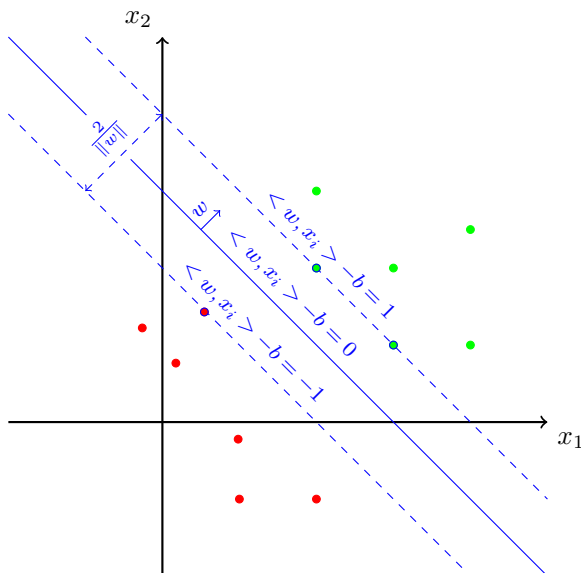


Abb. 1.2: Nun wurden zu der Trenngerade zwischen den Datensätzen noch zwei parallele Geraden hinzugefügt, die als Stützvektoren (Support vectors) einen der klassifizierten Datenpunkte nutzen. Dann lässt sich das mathematische Programm optimieren.

2 Neuronale Netze

2.1 Einführung

Inspiziert von unserem Verständnis, wie das menschliche Gehirn lernt, benutzen Neuronale Netze Lernalgorithmen, welche besonders für praktische Anwendungen geeignet sind. Dazu zählen Spracherkennung, Objekterkennung in Bildern und die Fähigkeit individuell passende Produkte vorzuschlagen, die dem Kunden gefallen könnten. Ein Neuronales Netz wird von mehreren Schichten aufgebaut. Ausgangsschicht ist dabei, die Datenschicht, auf die ein oder mehrere hidden layer folgen. Als Ausgabewert erhält man schließlich einen Vektor, welcher die Wahrscheinlichkeitsverteilung darstellt. Mit anderen Worten, wie wahrscheinlich es ist, dass das Ausgangsobjekt zu einer bestimmten Klasse gehört.

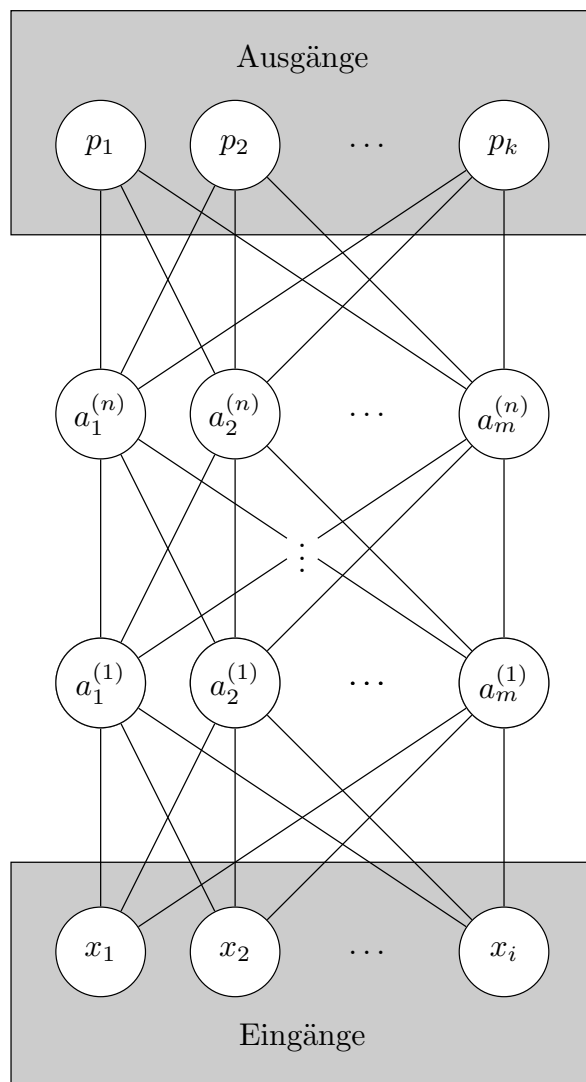


Abb. 1.3: Lorem ipsum dolor sit amet, consetetur sadipscing elitr

2.2 SVMs als Neuronale Netze

Wir haben bereits SVMs kennengelernt. Im folgenden wird verdeutlicht, wie man diese auch als Neuronales Netz ausdrücken kann. Charakteristisch für dieses Neuronale Netz ist, dass es nur einen Layer besitzt:

$$\min_{b,w,z} \|w\| + C \cdot \sum z$$

sodass:

$$y_i(\langle w, x \rangle - b) \geq 1 - z; z_i \geq 0 \Rightarrow z_i \geq 1 - y_i(\langle w, x \rangle - b); z_i \geq 0$$

$$\min \|w\| + C \cdot \sum \max(0, 1 - y_i(\langle w, x_i \rangle - b)) = \|w\| + C \sum \ell(1 - y_i(\langle w, x_i \rangle - b)) = \|w\| + C \sum \ell(a_i)$$

2.3 Lineare Regression

Auch die lineare Regression kann auch in Form von Neuronalen Netzen ausgedrückt werden. Der wesentliche Unterschied dabei ist, dass keine Klassifikation stattfindet, sondern den Datensätzen konkrete Werte zugeordnet werden.

$$\min_{w,b} \sum_i ((\langle w, x_i \rangle - b) - y_i)^2$$

2.4 Deep Learning

Mit Deep Learning beschreibt man die Neuronalen Netze, die über mehr als einen hidden layer verfügen. Dabei wird jeweils eine nicht lineare Funktion auf die linearen Ergebnisse der einzelnen layer angewendet, um die Anzahl der Parameter zu vergrößern.

2.5 Convolutional Neuronal Networks

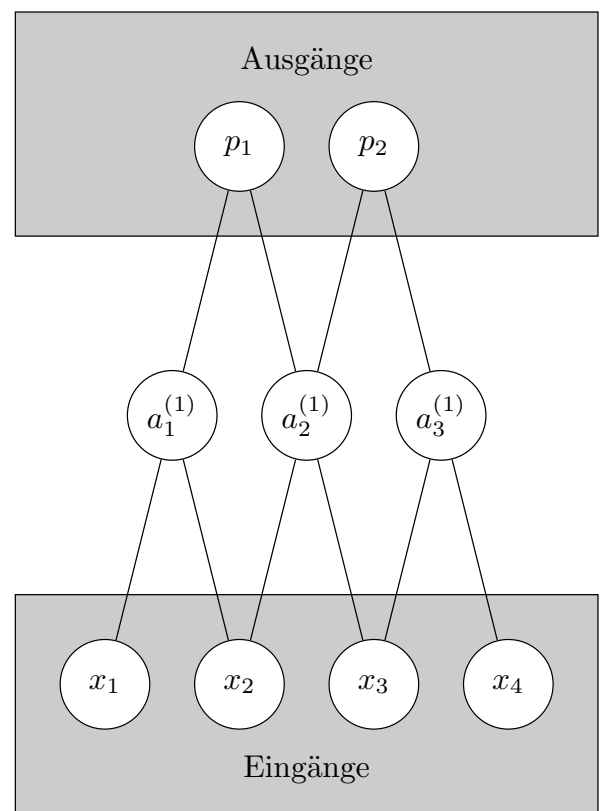


Abb. 1.4: Lorem ipsum dolor sit amet, consetetur sadipscing elitr