Karlsruher Institut für Technologie

Fakultät für Informatik
Institut für Anthropomatik und Robotik
Lehrstuhl für Intelligente Sensor-Aktor-Systeme (ISAS)
Prof. Dr.-Ing. Uwe D. Hanebeck

Diplomarbeit

Title

Author

31. Dezember 2020

Referent: Prof. Dr.-Ing. Uwe D. Hanebeck

Betreuer: Dipl.-Inform. X

Zusammenfassung

Abstract.

Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Diplomarbeit selbstständig angefertigt zu haben. Die verwendeten Quellen sind im Text gekennzeichnet und im Literaturverzeichnis aufgeführt.

Karlsruhe, 31. Dezember 2020

Author

Contents

List of Figures

List of Algorithms

List of Examples

List of Review Material

Notation

Conventions

x	Scalar
$\boldsymbol{x}$	Random variable
$\hat{x}$	Mean of random variable $\boldsymbol{x}$.
$\underline{x}$	Column vector
$\underline{\boldsymbol{x}}$	Random vector
$\hat{\underline{x}}$	Mean of random vector $\underline{\boldsymbol{x}}$.
$\mathbf{A}$	Matrix
$(.)_k$	Quantity at time step k.
$\mathbb{R}$	Set of real numbers.
$\sim$	Distribution operator. E.g., $\boldsymbol{x} \sim \mathcal{U}$ means $\boldsymbol{x}$ is distributed according to $\mathcal{U}$.
■	End of example.
□	End of proof.

Abbreviations

KF	Kalman Filter
LRKF	Linear Regression Kalman Filter
RMSE	Root Mean Square Error

CHAPTER 1

Grundlagen

Grundlagen:

- TrackSort Schüttgutsortierung - Kalman Filter - NN - RNN - LSTM

Das Kalman-Filter Als Kalman-Filter bezeichnet man ein mathematisches Verfahren mit dem Messfehler in realen Messwerten reduziert werden können und nicht messbare Systemgrößen geschätzt werden können.

[vergangene, aktuelle und zukünftige Systemzustände schätzen] [Einschränkung Linearität (Extended Kalman) und Gauß rauschen]

Der Zustand des Systems zum Zeitschritt t wird als y_t und die Messung im Zeitschritt t als z_t bezeichnet.

$$y_t = Ay_{t-1} + w, w \sim N(0, Q)$$

$$z_t = Hy_t + v, v \sim N(0, R)$$

Dabei ist A die Zustandsübergangsmatrix, die den Übergang von einem Zustand in den nächsten beschreibt. H ist die Messmatrix, die beschreibt wie Messungen aus dem Zustand entstehen und Q und R sind die Kovarianzmatrizen des Systemrauschens beziehungsweise des Messrauschens.

Das Kalman-Filter funktioniert mittels abwechselnd ausgeführter *predict* und *update* Schritte.

$$\hat{y}'_t = A\hat{y}'_{t-1}$$

$$\hat{P}'_t = A\hat{P}'_{t-1}A^T + Q$$

1.1 Neuronale Netze

Die Grundsteine des Feldes wurde 1943 von Warren McCulloch und Walter Pitts gelegt, die in ihrem Paper ein Neuronenmodell vorschlugen, mit dem sich logische arithmetische Funktionen berechnen lassen. Infolge dessen gab verschiedene Forschungsbestrebungen in dem Feld, wie [Examples: TODO!].

[Viele der Begrifflichkeiten, die wir heute noch verwenden wurden 1956 auf der Dartmouth Conference festlegt.]

Nachdem jedoch Marvin Minsky und Seymour Papert zeigten, dass einzelne Perceptrons nicht in der Lage sind linear nicht separierbare Probleme zu lösen sank das Interesse an dem Feld.

1.1.1 Perzeptron

Die kleinste Einheit eines neuronalen Netzes ist das Perzeptron. Es ist eine Art künstliches Neuron, dass eine Reihe an Eingaben entgegen nimmt und einen einzelnen Wert o ausgibt. Die einzelnen Eingaben x_i haben jeweils eine Gewichtung w_i. Es existiert ein sogenannter Schwellwert oder *bias*, der normalerweise durch eine zusätzliche Eingabe x_{m+1} mit dem Wert $+1$ und dem dazugehörigen Gewicht w_{m+1} modelliert wird. Den Ausgabewert y erhält man dadurch, dass man die gewichteten Eingaben aufsummiert und in die Aktivierungsfunktion des Perzeptrons gibt. Ein Überblick über verschiedene Aktivierungsfunktionen ist unter 1.1.2 zu finden.

Mathematisch ist die Ausgabe eines Perzeptrons also wie folgt definiert:

$$y = \phi(\sum_{i=0}^{m} w_i x_i)$$

1.1.2 Aktivierungsfunktionen

Es gibt verschiedene Aktivierungsfunktionen, die für den Einsatz in neuronalen Netzen in Frage kommen. Sie sind von essenzieller Wichtigkeit, da ohne eine Nicht-Linearität das Netz in eine einfache Regression kollabiert.

Eine Aktivierungsfunktion sollte leicht abzuleiten sein, da dies im Rahmen des Backpropagation Algorithmus häufig geschieht und sonst beträchtlicher Rechenaufwand entsteht.

Sigmoid-Funktion

$$f(x) = \frac{1}{1+e^x} = \frac{e^x}{e^{x+1}},$$

$$f'(x) = f(x) * (1 - f(x))$$

TanH

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
$$f'(x) = 1 - f(x)^2$$

ReLu

$$f(x) = max(0, x)$$
$$f'(x) = \begin{cases} 0 & \text{, falls } x < 0 \\ 1 & \text{, falls } x > 0 \end{cases}$$

1.2 Related Work

Notizen:

Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization von Huseyin et al. - Motion Model und Measurement Model mittels LSTM Netzen lernen. - Am Beispiel Pose-Estimation. - 3 Separate LSTMs, je eins für Motionmodel, System und Messrauschen. (A = Transition Matrix, H = Measurement Matrix, Q = Covariance of process noise, R = Covariance of sensor noise)