Neuronale Netze

Proseminar Data Mining

Lukas Krenz
Fakult t f r Informatik
Technische Universit t M nchen
Email: lukas@krenz.land

***Zusammenfassung*— Das Paper ist eine kurze Einführung in Neuronale Netze, insbesondere in Multilayer Perceptrons inklusive Analyse ihrer praktischen Anwendung.**

***Schl sselworte*— Neuronale Netze, Multilayer Perceptrons, Backpropagation,**

INHALTSVERZEICHNIS

I. EINLEITUNG

Die Familie der Künstlichen Neuronalen Netze(ANN) ist wohl aktuell eins der spanndendsten Forschungsgebiete im Bereich des machinellen Lernens. In der letzen Zeit ist es die Methode, die in vielen Bereichen die besten Ergebnisse erzielt:

- Clustering
- Computer Vision
- Natural Language Processing.

Auch, wenn sie auf den ersten Eindruck kompliziert und unverständlich wirken, ist die mathematische und statistische Grundlage simpel.
Im folgenden wird der Weg vom linearen Modell zum Perceptron zum ANN gezeigt - mit jedem Schritt wird dabei die Komplexität, aber auch die Flexibilität erweitert. Nur die sogenannten Feed-Forward NNs werden hier beschrieben, wie sich später zeigen wird, sind diese auch mächtig genug, um viele Probleme zu lösen - auch wenn andere ANNs bessere Ergebnisse liefern können, sind die Grundlagen meistens übertragbar.

II. VOM LINEAREN MODELL ZUM NEURONALEN NETZ

Eines der klassischsten Modelle der Regressionsanalyse sind lineare Regression und die logistic Regression.

Es sind mathematisch simple Verfahren, die sogar analytisch lösbar sind - aber eben nicht immer ideale Ergebnisse erzielen: es gibt Probleme, die mit den Verfahren nicht lösbar sind.

Sehr simpel; lineares Modell als Input -> Output layer, dann lineare Funktion als Hidden Layer -> Lineares Modell
(Vergleich; Bild Logistic Regression vs Lineares Perceptron")

A. Das Perceptron als additatives lineares Modell

Modell perceptron.

Es gibt viele Probleme, die von einem Perceptron nicht gelöst werden können. Es können nur Funktionen von linearen Modellen approximiert werden, die linear seperierbar sind. Da das Perceptron als Verknüfung von linearen Funktionen auch linear ist, kann auch es diese Klasse von Funktionen nicht lösen.

B. Das Perceptron als einfaches ANN

Verallgemeinerung zum ANN.
blabla mit drei Quellenangaben [1]–[3]

C. Das XOR-Problem: nicht linear separierbare Probleme

Theorem: Single Layer kann alle Funktionen annähern.

D. Die Lösung: Neuronale Netze

Neuronale Netze sind eine verallgemeinerung von Perceptrons. Sie besitzen noch eine (oder mehrere) weitere Schicht(en) Neuronen, die jedoch nicht nur lineare Transformationen durchführen, sondern zusätzlich noch eine nicht lineare.

Diese nicht-linearität ermöglicht es, auch nicht linear seperierbare Probleme zu lösen.

Durch die Nicht-linearität und durch die komplexere Verknüpfung resultiert ein nicht lineares Optimierungsproblem, das meistens nicht mehr analytisch zu lösen ist: es werden numerische Verfahren benutzt.

III. Neuronale Netze

A. Die Aktivierungsfunktion

Die Funktion, die pro Knoten die nicht-lineare Transformation durchführt, heißt Aktivierungsfunktion.
Häufig wird eine Funktion mit sigmoiden Erscheinungsbild gewählt, das heißt eine Funktion, die die Ergebnisse in ein bestimmtes Interval "zusammenquetscht"(Quelle) und ein an ein "S"erinnerndes Erscheinungsbild hat.
Die einfachste Funktion diesen Typs ist die so genannte logistische Funktion

$$\sigma_1(x) = \frac{1}{1+e^{-x}} \tag{1}$$

oder der hyperbolische Tangens

$$\sigma_2(x) = \tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{2}$$

.

Der Hyperbolische Tangens hat in der Praxis eine bessere Laufzeit (vlg. LeCunn: Efficient Backprop)

Alternativ ReLu:

$$\sigma_3(x) = \max(0, x) \tag{3}$$

bis zu 6 mal schneller (TODO: Quelle)
...

B. Die Kostenfunktion

Mean squared Errors(MSE) (euklidische Norm)

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2 \tag{4}$$

oder negative Cross entropy (TODO: Quelle, richtige Formel)

$$-\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right] \tag{5}$$

Bei Klassifikationsproblem ist Cross entropy fast immer die bessere Wahl.

C. Die Minimierung der Kostenfunktion - Backpropagation

Backpropagation ist der bei Neuronalen Netzen benutze Trainingsalgorithmus.

Er wird benutzt, um den Gradient der Lossfunction in Bezug auf alle Gewichte zu berechnen.

Dieser wird dann mit Hilfe einer geeigneten numerischen Methode benutzt, um die Gewichte zu optimieren.

Am Ende wird das Ergebniss der Optimierungsmethode benutzt, um die Gewichte wieder zu updaten.

D. Die numerische Methode - Gradient Descent und verwandte

Der Gradient wird oft mit der bekannten numerischen Methode Gradient Descent(eingedeutscht: Verfahren des steilen Abstiegs) benutzt.

Er ist relativ simpel:

Formel

Eine andere oft benutze Form ist Stochastic Gradient Descent. Bei diesem Optimierungsalgorithmus ist die Auswahl der zu updatenden Gewichte nicht deterministisch. Dadurch werden in der Praxis oft bessere Ergebnisse erziel.

Gradient Descent with momentum

IV. Neuronale Netze in der Praxis

Weigth Decay, Anfangwert

A. Problematik von lokalen Minima

Cost function hat mehrere Minima.
Lösung: Neu starten
Oft nicht globales minimum erwünscht (siehe Overfitting)

B. Anzahl Hidden Layer

C. Initialisierung der Gewichte

Randomisiert - oft nicht gut

D. Vermeidung von Overfitting

Ein Problem, dass in der Praxis auftreten kann, ist das so genannte Overfitting, das ist die übermäßige Anpassung an die Trainingsmenge. Oft ist das Modell zu stark angepasst und daher nicht mehr allgemein effizient.

Eine mögliche Lösung ist die L2-Regularisierung. Bei ihr wird an die Lossfunction ein weiterer Term angehängt:

$$\text{Term einbauen + Quelle!} \tag{6}$$

.

Er skaliert mit der Summe aller Gewichte, es werden also komplexe Modelle bestraft
Bayesian View of L2-Reg. (wtf?)

Eine weitere Lösung ist das so genannte early stopping, bei dem man nicht mit der Backpropagation aufhört, wenn ein (lokales) Minimum gefunden wurde, sondern dann, wenn die Performance des Modells bei dem Validierungsset optimal ist. Dabei wird oft die Iteration oft beendet, wenn bei der Fehlerfunktion gar kein Minimum vorliegt.

E. Regression

F. Klassifizierung

Cross entropy.
Soft-Max-F unction

V. Zusammenfassung und Ausblick

blabla

Literatur

[1] B. Claise, “IPFIX protocol specifications,” Internet-Draft, draft-ietf-ipfix-protocol-07, December 2004.

[2] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, “Hash-based IP traceback,” in *ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2001.

[3] A. Belenky and N. Ansari, “IP traceback with deterministic packet marking,” *IEEE Communications Letters*, vol. 7, no. 4, pp. 162–164, 2003.