Fragebogen der Fachschaft zu
mündlichen Prüfungen
im Informatikstudium

Dieser Fragebogen gibt den Studierenden, die nach dir die Prüfung ablegen wollen, einen Einblick in Ablauf und Inhalt der Prüfung. Das erleichtert die Vorbereitung.
Bitte verwende zum Ausfüllen einen schwarzen Stift. Das erleichtert das Einscannen.
Vielen Dank für deine Mitarbeit!

2 214100 126937

Dein Studiengang: Informatik

Prüfungsart:
- ○ Wahlpflichtfach
- ☒ Vertiefungsfach
- ○ Ergänzungsfach

Welches? Anthropomatik und Kognitive Systeme

Prüfungsdatum: 8. Oktober 2020

Prüfer(in): Prof. Friederich

Beisitzer(in): Prof. Friederich

Prüfungsfächer und Vorbereitung:

Veranstaltung	Dozent(in)	Jahr	regelmäßig besucht?
Maschinelles Lernen für die Naturwissenschaften	Prof. Friederich	SS 2020	nur Aufzeichnungen

Note: 1,0 Prüfungsdauer: 30 Minuten

War diese Note angemessen? Ich hätte mir eine 1,3 gegeben; Benotung war sehr großzügig

✍ **Wie war der Prüfungsstil des Prüfers / der Prüferin?**

Prüfungsatmosphäre, (un)klare Fragestellungen, Frage nach Einzelheiten oder eher größeren Zusammenhängen, kamen häufiger Zwischenfragen oder ließ er/sie dich erzählen, wurde dir weitergeholfen, wurde in Wissenslücken gebohrt?

Prof. Friederich hat zum Einstieg erst mal meinen Hintergrund (Studium, andere Kurse, geplante Masterarbeit) abgefragt. Auch dadurch war die Prüfungsatmosphäre sehr entspannt. Er hat dann mit Fragen angefangen, wo er sich bei meinem Hintergrund sicher sein konnte, dass ich sie gut beantworten kann. Er hat mich immer ausreden lassen und nicht in Wissenslücken gebohrt. Die Fragen waren anfangs sehr allgemein, sodass man erst mal alles, was einem zum Thema eingefallen ist, loswerden konnte. Erst später wurde er detaillierter, hat aber keine kleinen Einzelheiten verlangt. Bei Unsicherheiten hat er auch weiter geholfen.

↪ **Rückseite bitte nicht vergessen!**

✍ **Hat sich der Besuch / Nichtbesuch der Veranstaltung für dich gelohnt?**

Ich habe nur die Aufzeichnungen gesehen, nicht die Fragestunden besucht. Die Aufzeichnungen würde ich auf jeden Fall empfehlen, weil er schon auch viel erzählt, was nicht auf den Folien steht, und das alles verständlicher macht. Bei manchen Themen hätten die Folien allein nicht ausgereicht.

✍ **Kannst du ihn / sie weiterempfehlen? Warum?**

☒ ja
○ nein

Auf jeden Fall! Nicht so mathematisch wie die ML Grundlagen VL bei Neumann. Hier werden sogar mehr Modelle vorgestellt, aber das ganze zielt mehr auf das allgemeine Verständnis und die Anwendung ab. Gerade die Anwendungen waren auch sehr interessant und alle damit verbundenen vorgestellten Paper hochaktuell.

✍ **Wie lange und wie hast du dich alleine oder mit anderen auf die Prüfung vorbereitet?**

alleine ca. 12 volle Tage für Aufzeichnungen anschauen, Zusammenfassung und auswendig lernen

✍ **Fanden vor der Prüfung Absprachen zu Form oder Inhalt statt? Wurden sie eingehalten?**

nein, alles ist relevant

✍ **Welche Tipps zur Vorbereitung kannst du geben?**

Wichtige / unwichtige Teile des Stoffes, gute Bücher / Skripten, Lernstil, ...

Auch Geschichte des Machine Learning ist relevant.
Die Mathe kann man größtenteils weglassen für die Prüfung.
Aufzeichnungen anschauen.

✍ **Kannst du Ratschläge für das Verhalten in der Prüfung geben?**

das Übliche halt: offen, freundlich, anständig anziehen, keine Panik ;)

Inhalte der Prüfung (bitte auf weitere Blätter):

- Schreibe bitte möglichst viele Fragen und Antworten auf.
- Wo wurde nach Herleitungen oder Beweisen gefragt oder sonstwie nachgehakt?
- Worauf wollte der Prüfer / die Prüferin hinaus?
- Welche Fragen gehörten nicht zum eigentlichen Stoff?

Fangen wir an. Was können Sie denn zur Geschichte des Maschinellen Lernens sagen?
Ich jetzt schon völlig panisch, weil ich den Einführungsteil wie üblich nur überflogen habe und nicht explizit gelernt habe
Puh, ähm... Da gab es verschiedene Teile. Angefangen hat das glaube ich mit dem Perzeptron. Dann gab es da verschiedene Winter des Maschinellen Lernens. In den 2010ern kam dann der große Durchbruch mit Deep Learning, weil wir endlich genug Rechenpower hatten.

Und warum kam es zu den Wintern zwischendurch?
Weil wir nicht genug Rechenpower hatten und große Versprechen gemacht haben, die nicht eingehalten wurden. *sinnvoller bzw. auch wichtig wäre gewesen, dass es die Datenmengen ja früher auch noch nicht gab*

Ja, wenn Sie sich gerne mit Sprachverarbeitung bzw. Übersetzung beschäftigen: welches Machine Learning Modell benutzt man denn dafür?
Beim Übersetzen benutzt man sogenannte Sequence-to-sequence Modelle oder ganz allgemein Rekurrente Neuronale Netze. Dieses Modell kann nämlich Input variabler Länge verarbeiten. Das Modell arbeitet mit einem hidden state und bekommt in jedem Schritt einen Input. Aus dem alten hidden state und dem Input wird dann der neue hidden state berechnet. Dann gibt es da zwei Arten von Output: entweder am Ende für z.B. Klassifizierung oder in jedem Zeitschritt wie bei Übersetzung. Da kann dann der Output auch noch in die Berechnung des neuen hidden state mit eingehen. Aber dann hat man das Problem, dass zum Beispiel Ï want to goßu Ïch will gehen"übersetzt wird und dann die Anzahl der Wörter nicht übereinstimmt und nicht für jeden Schritt des RNN ein Output zur Verfügung steht. Da hatten wir dann noch das Encoder-Decoder Modell. Da verarbeitet der Encoder erst mal den ganzen Input und updatet damit immer seinen hidden state. Der letzte hidden state des encoders wird dann als Kontextvektor an den Decoder übergeben. Der Decoder hat dann seinen eigenen hidden state, in den der Kontextvektor immer wieder einfließt, und der berechnet dann basierend darauf und dem späteren Output den neuen Output, also die Übersetzung.

Gut, da hatten wir aber noch eine Erweiterung.
Ja, wir haben dann noch Attention verwendet. Da wird dann jeder hidden state vom encoder nochmal mit einer Matrix A gewichtet und geht dann in den aktuellen hidden state vom Decoder mit ein. Wir hatten da in der VL das Beispiel, dass man ein Buch liest und dazu Fragen beantworten muss. Mit Attention ist das so, als kenne man die Frage schon vorher und man kann im Buch nachschlagen. So lernen wir die Gewichtung der hidden states mit den Inputs vom Decoder. Wir lernen quasi, wann welcher Input für den Output wichtig ist über die Gewichte dieser A Matrix. Die kann man dann auch visualisieren. Das ist die sogenannte Attention Matrix und da sieht man dann, für welchen Output, welcher Input angeschaut wurde. Beim Übersetzen ist das meist grob eine Diagonale. Aber dann hat man in einigen Sprachen die Adjektive und Nomen vertauscht von der Reihenfolge und dann sieht man das auch in der Attention Matrix.

Was ist denn mit so bildgebenden Daten?
hab die Frage erst nicht gecheckt, weil mich das gebendirritiert hat. Er meinte aber einfach Bilddaten
Dafür kann man dann sogenannte Convolutional Neural Networks verwenden.

Und was machen die? Wie funktionieren die?
CNNs sind besonders gut für grid Daten wie eben zum Beispiel Bilder. Man hat also 2D input - bei den Bildern zumindest und wenn es Farben hat, dann sind die als Channel in der 3. Dimension.
In einem CNN hat man dann so convolutional layer. Die bestehen zunächst mal aus convolution also Faltung. Dabei werden Filter auf das Bild angewendet, zum Beispiel Kantendetektion und mit diesen Filtern "faltet"man das Bild dann. Dann hat man noch Nichtlinearität, also eine Aktivierungsfunktion, zum Beispiel ReLU. Am Ende kommt dann noch ein Pooling, zum Beispiel Max Pooling oder Average Pooling. *Dann sollte ich nochmal genauer erklären, was beim Pooling passiert. Ich hab einfach so lange irgendwas weitererklärt wie ich was wusste.* Dabei werden mehrere Pixel zu einem zusammengefasst. Also der Input ist am Anfang vielleicht 256x256 Pixel groß und mit jedem Layer wird er dann kleiner durch Convolution und Pooling. Beim Max Pooling wählt man dann einfach das Pixel mit dem maximalen Wert aus und beim average Pooling mittelt man über alle. Mit den Filtern der Convolution kommen dann immer neue Channel dazu. Sodass aus 2D Bild dann quasi eine kleinere 3D innere Repräsentation wird. Wenn man nicht will, dass sich das Bild bei Convolution verkleinert, kann man da auch noch padding hinzufügen, also je nach Filtergröße einfach entsprechend viele 0en als Rahmen um das Bild setzen, sodass das Bild danach wieder Originalgröße hat.

Wir hatten da so eine Anwendung aus der Quantenchemie, wo wir Elektronendichte auf Elektronendichte gemappt haben. Wie haben wir denn dabei CNNs angewendet?
Da hatten wir das U-Net verwendet. Das sieht schematisch halt wie ein U aus. Die rechte Seite ist dabei ein ganz normales CNN. Also wir wenden Faltung an, um eine innere Repräsentation zu lernen und dabei wird der Input entsprechend immer kleiner. Auf der rechten Seite wenden wir dann Deconvolution an. Wir nehmen also die kleine gelernte Repräsentation und vergrößern also entfalten sie quasi wieder, sodass wir wieder ein Bild produzieren nur halt ein anderes. Wir hatten da noch dieses Beispiel mit den Zellen, die eingefärbt werden sollten. *Das mit der Elektronendichte hab ich nämlich nicht so gut kapiert, dass ich damit anschaulich was hätte erklären können.* Dabei wollten wir ja auch das Bild der Zellen wieder haben, nur halt eingefärbt. Dafür haben wir dann auch die Zwischenschritte der gelernten Repräsentation für die Entfaltung wieder verwendet.

Was ist denn das Schöne oder Besondere am U-Net?
Ähm ja, diese Skip Connections, dass man die Zwischenrepräsentationen des Bilds bei der Deconvolution wieder verwendet und diese verschiedenen Repräsentationen kombiniert und verrechnet werden.

Wir hatten die Skip Connections ja noch woanders, nämlich beim ResNet. Wie funktioniert denn das?
Das ResNet ist so ein vortrainiertes Modell für Bilderkennung, das man sich einfach runterladen kann und dann mit Transfer Learning verwenden kann, indem man darauf aufbauend nochmal trainiert. Das ResNet hat diese Skip Connections, wo nach 1 oder 2 Layern, also Convolution, Nichtlinearität und Pooling, der Input x einfach nochmal drauf addiert wird. Dadurch dass wir die Identität nochmal addieren lernen wir eine Korrektur vom Input x und nicht einfach nur ein Mapping. Dadurch kann man dann tiefere Netze mit mehr Layern trainieren, weil der Fehler besser durch die Ebenen zurück propagiert wird.

Das ResNet wurde ja auch für Objektklassifizierung verwendet. Wie funktioniert das dann, dass man klassifiziert, wenn man ein CNN hat?
Da hat man dann am Ende einfach nochmal ein densely connected layer, also quasi ein kleines Neuronales Netz hintendran, das dann die Klassifizierung macht. Also zum Beispiel mit Softmax dann die Wahrscheinlichkeiten für die einzelnen Klassen berechnet.

Machen wir mal neues Thema. Hmm, wie wäre es mit Entscheidungsbäumen. Wie funktionieren die denn?
Entscheidungsbäume sind ein nicht-parametrisches Modell. Dabei werden einfach die Daten gesplittet, indem man so Abfragen macht, wie ist $x < x_1$? Wenn ja, dann geht man eben in den linken Unterbaum und da gibt es dann weitere Aufteilungen bis man dann in einem Blatt ankommt, wo nur noch ein Datenpunkt ist und dann eine Klasse vorhergesagt wird.

Und wie genau geht das mit dem Training?
Also da gab es den Greedy Algorithmus. Wir haben ja verschiedene Features also Dimensionen in unseren Daten. Anhand eines Features wird ja dann eine Entscheidung getroffen, wo der Datensatz geteilt wird. Bei k Features und n Datenpunkten kann man dann $k \cdot n$ verschiedene Splits machen, um die Daten zu teilen. Man wählt dann den Split aus, für den die Unreinheit in den Subsets am geringsten ist. Die Unreinheit gibt quasi an, wie viele Datenpunkte falsch klassifiziert würden. Die kann man zum Beispiel über Entropie, den Gini-Koeffizienten oder den Klassifizierungsfehler berechnen.

Wenn Sie jetzt schon $k \cdot n$ verschiedene Splits erwähnen. Wie viele Splits muss man denn insgesamt berechnen?
$O(k \cdot n \log(n)^2)$

Und wo kommen die logs her?
Also das eine log kommt von der Baumtiefe. Wenn die Splits einigermaßen gut gewählt sind bekommt man idealerweise einen halbwegs ausbalancierten Baum, sodass man nicht auf einer Seite viele Entscheidungskriterien anschauen muss und auf der anderen Seite kaum welche. Das zweite log kommt vom Sortieren der Samples nach ihren Features.

Ist das Trainieren eines einzelnen Baums denn gut?
Nein, nicht so besonders. Also der produziert eine sehr kantige Entscheidungslinie.

Wie ist das da mit Bias und Varianz? Also welches davon ist bei einem einzelnen Baum denn hoch?
Ich hatte keine Ahnung, jetzt kam der Teil, den ich beim Lernen später nochmal anschauen wollte und dann vergessen habe. Also habe ich geraten:
Der Bias ist dann hoch? Also weil es ja stark von dem Baum abhängt.

Nein, der Bias ist da nicht hoch. Wenn ich mit neuen Daten trainiere, dann bekomme ich ja einen ganz anderen Baum. Wir haben also eine hohe Varianz. Die wollen wir verringern.
Sowas hat er grob gesagt, um mich dahin zu leiten, dass man viele Bäume verwenden sollte, da sie die Varianz verringern, weil wir bei Mittelung den Faktor 1/m bei der Varianz beibehalten. Hat aber etwas gedauert, bis ich gecheckt habe, dass er dahin wollte.
Ahh! Ja, da hatten wir das mit der Varianz, die durch mehrere Bäume verringert wird. Da mitteln wir über mehrere Bäume und dann bleibt der Faktor 1/m erhalten. Wobei beim Erwartungswert der Faktor nicht bleibt, also der Bias gleich bleibt.

Wieso jetzt mehrere Bäume?
Achso ja, also wir können da Ensemble Methoden verwenden, zum Beispiel Random Forests. Damit wir mehrere unabhängige Bäume haben, um dann über die zu mitteln und so eine bessere Vorhersage und eine glattere Entscheidungslinie zu bekommen.

Ja genau. Wie kriegt man denn unabhängige Bäume?
Für Random Forests verwenden wir Bagging. Das steht für Bootstrap Aggregation. Da erstellen wir Subsets mit Zurücklegen, also es kann auch mal vorkommen, dass ein Datenpunkt mehrfach in einem Subset vorkommt. Und mit diesen unabhängigen Subsets können wir dann unabhängige Bäume trainieren. Da gab es auch noch eine zweite Methode, um Unabhängigkeit zu erreichen. Der Name fällt mir grad nicht ein, aber ich beschreib mal, was man da macht.

Gut. Haben wir denn noch Zeit? Ist ja erst 22 nach.
Wie noch 8min - das war doch schon ne Ewigkeit?

Wieso werden Entscheidungsbäume denn gerne in den Naturwissenschaften verwendet?
Weil man bei Entscheidungsbäumen durch die Entscheidungskriterien eindeutig erklären kann, wie es zu einer Entscheidung kam und welche Features für die Entscheidung besonders wichtig waren. Das ist für die Naturwissenschaften besonders wichtig, weil man immer am Warum interessiert ist. Außerdem sind Entscheidungsbäume einfach zu trainieren und schnell und einfach in den Vorhersagen

Ja, also hauptsächlich sind sie einfach zu trainieren. Warum denn?
??? *wusste nicht, worauf er hinaus wollte*

Also gerade im Vergleich zu neuronalen Netzen, wo man Parameter und noch sowas optimieren muss.
Ach ja, man hat keine Hyperparameter bis auf die Anzahl der Bäume und damit muss man nicht viel herumprobieren und das Training ist quasi ein Selbstläufer.

Gut. Wir haben immer noch Zeit. Wir hatten ja noch das Thema Bayes'sche Optimierung. Wie können wir uns das in den Naturwissenschaften nutzbar machen?
Für autonome Experimente. Wir hatten da das Project Ada als Beispiel aus Vancouver. Da wurden automatisch Experimente durchgeführt mit variablen Mischverhältnissen von Stoffen, um das ideale Mischverhältnis für organische Solarzellen zu bestimmen. In diesem Falle ging es um die Leitfähigkeit, die optimiert werden sollte. Je nach Ausgang des Experiments wurden mit Hilfe von Bayes'scher Optimierung automatisch die neuen Mischverhältnisse (also Parameter) bestimmt, die als nächstes getestet werden sollen, um die Leitfähigkeit zu verbessern.

Und wie genau bestimmen wir das?
Wir bestimmen eine sogenannte acquisition function, die uns sagt, welcher Punkt (also welche Kombination von Parameterwerten) am nützlichsten zu wissen wäre auf der Suche nach dem Optimum.

Was heißt denn am nützlichsten
Am nützlichsten heißt, wenn wir nach dem globalen Maximum suchen einerseits eine Umgebung, wo schon hohe Werte vorhanden sind, aber gleichzeitig genug Unsicherheit, dass es wahrscheinlich wird, dass man noch höhere Werte in der Gegend findet.

Und geht das auch mit zum Beispiel neuronalen Netzen mit der Unsicherheit?
Nein, dafür nutzen wir Gaußprozesse mit Unsicherheit. Gaußprozesse haben als Output eine Wahrscheinlichkeitsverteilung unter der man dann einen Wert samplen kann. Diese Wahrscheinlichkeitsverteilung modelliert die Unsicherheit, die bezüglich des Wertes in dieser Gegend besteht.

Ok und wie ging das jetzt mit der acquisition function?
Er wollte da auf den tradeoff zwischen Exploration und Exploitation hinaus und was das heißt. Das hat mehr schlecht als recht funktioniert und er musste mir mal wieder helfen.
Ok, das war jetzt sehr verschwurbelt erklärt. Soll ich das nochmal aufzeichnen?

Nein, danke. Ich glaube wir haben es jetzt. Sie können kurz rausgehen. Wir besprechen uns und ich hole Sie dann wieder.
Ich habe also nervös die Plakate auf dem Flur gelesen und wurde recht schnell auch schon wieder reingebeten.

Ja sehr schön. Es freut mich immer, wenn ich so eine eindeutige Prüfung habe und alles so glatt läuft. Wir waren uns dann auch schnell einig, dass Sie eine 1,0 bekommen. Das hat mich sehr gefreut. :)