



Fragebogen der Fachschaft zu
mündlichen Prüfungen
im Informatikstudium

Dieser Fragebogen gibt den KommilitonInnen, 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.

Dein Studiengang: ... Informatik Master.....

Prüfungsart:

- Wahlpflichtfach
- Vertiefungsfach
- Ergänzungsfach

Welches ?

Barcode:



Prüfungsdatum: 25.03.2021

Prüfer/-in: Pascal Friederich

Beisitzer/-in: Matthias Schniewind

Prüfungsfächer und Vorbereitung:

| Veranstaltung | Dozent/-in | Jahr | regelmäßig besucht? |
|----------------------------|-------------------|------|---------------------|
| ML für Naturwissenschaften | Pascal Friederich | 2021 | Alle VL gesehen |
| | | | |
| | | | |
| | | | |

Prüfungsablauf:

Prüfungsdauer:35... Minuten

Note:1.0....

War diese Note angemessen?

Ja

• 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?)

Sehr nett und angenehm. Er hat kaum Fragen gestellt und ich hab halt einfach alles Erzählt was ich zu dem Thema weiß. Prüfung war online mit Kamera

↳ Rückseite bitte nicht vergessen

• Hat sich der Besuch / Nichtbesuch der Veranstaltung für dich gelohnt?

• Wie lange und wie hast Du Dich alleine bzw. mit anderen vorbereitet?

5 Tage, aber sehr viel Vorwissen

War alles Online, Videos anschauen war gut und wichtig

• Welche Tipps zur Vorbereitung kannst Du geben? (Wichtige / Unwichtige Teile des Stoffes, gute Bücher / Skripten, Lernstil) Warum?

• Kannst Du ihn/sie weiterempfehlen? x Ja / Nein

Der Youtubekanal DeepLizard hat eine klasse Serie zum Reinforcement Learning

Ja ist interessant

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

• Kannst Du Ratschläge für das Verhalten in der Prüfung geben?

Ich habe gefragt ob ich Papier bereithalten soll (für Formeln) was verneint wurde

Nö

Inhalte der Prüfung: → Bitte auf die Rückseite und weitere Blätter!

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

Zuerst haben wir über meine Masterarbeit und mein Vorwissen geredet.

P: Was ist denn das simpleste ML Verfahren das Sie kennen

I: Lineare Regression, Linie wird auf Punkte so gefittet, dass der Abstand der Punkte zu der Linie minimal ist

P: Wenn ich jetzt klassifizieren will auf binärklassen wie mach ich das?

I: Logistische Regression, Wahrscheinlichkeit von zweiter Klasse ist $(1 - px)$, Neuron mit sigmoid Aktivierungsfunktion, die mappt auf $(0,1)$

P: Und wenn man mehrere Klassen hat?

I: Neuronen in Anzahl an Klassen, Softmax normiert ausgabewerte auf 100%, Ergebnisse können jetzt als Prozentzahl interpretiert werden

P: Dann kommen wir zu Bildgebenden Verfahren da hatten wir in der Vorlesung ein Modell zur Brustkrebserkennung (er hat nicht zur Funktionsweise von CNNs gefragt, da ich ihm gesagt hatte dass ich da Vorwissen hab),

I: Ensemble Modell, 3 Modelle, 2 Stage Objektdetektor der Retinanet (obwohl das 1 stage ist?) als BB Prädiktor nimmt

und dann mit MobileNet die Boxen klassifiziert. 2tes Modell macht Resnet auf linke und rechte Brust einzeln und nimmt die resultierende Featuremap und gibt sie weiteren residualen Blöcken, dann Klassifikation ob Krebs rechts oder links. Letztes Modell nimmt ResNet und klassifiziert nur ob Krebs ja oder nein

P: Wie trainiert man das jetzt, man hat ja nur wenige Daten.

I: Normalerweise initialisiert man ResNet mit ImageNet Gewichten. Entweder trainiert man nur Fully Connected Layer am Ende neu oder man trainiert das ganze ResNet (mit vlt kleinerer Learning Rate). Hatten dazu eine Diskussion wie sinnvoll es ist auch die Conv Gewichte zu trainieren, ich hab gesagt, dass die Scans schon echt anders als die ImageNet Bilder sind und dass die higher level features sehr unterschiedlich sind. Er war der Meinung, dass man beim CNN aufpassen muss um nicht zu overfitten und lieber nur den FC trainiert.

P: Dann kommen wir mal zu generativen Modellen was hatten wir denn da

I: Hab zuerst erwähnt dass wir meist SMILES Codes generiert haben

P: Warum SMILES und nicht Fingerprint?

I: Gut lesbar und so aber konnte keine richtige Antwort geben

P: Der Fingerprint ist nicht eindeutig wegen hash collisions, man kann einen Fingerprint möglicherweise nicht nur einem Molekül zuweisen. Ich hab dann noch gesagt dass in dem einen Paper der Vektor 47000 Dimensionen hatte was hash Kollisionen weniger wahrscheinlich macht und er hat gesagt, dass das trotzdem nicht reicht. Zurück zu generativen Modellen

I: Autoencoder, transformiere Input zu Latent Space mit Encoder und transformiere den Latent Space Vektor mit Decoder zum Output, Output soll gleich wie input sein, Lossfunktion vergleicht input mit target.

P: Können wir im latent space interpolieren?

I: Nein, latent space wird nicht kompakt genutzt, viele Punkte haben keine Bedeutung daher Variational Autoencoder. Statt Vektor im Latent Space erzeugt Generator eine Wahrscheinlichkeitsverteilung mit Mittelwert und Varianz. Aus der wird gesampelt und das Sample dem Decoder gegeben

P: Wie macht er die Verteilung?

I: Einfach zwei Neuronen, die mean und varianz ausgeben, training ist Ende zu Ende.

P: Komische Frage, aber er wollte das mit der Regularisierung hören.

I: Zusätzlich zu dem Input Output Unterschied loss kommt ein Regularisierungsterm hinzu, man nimmt die Kullback-Leibler Divergence um die Prädizierten Wahrscheinlichkeitsverteilungen mit der Normalverteilung zu vergleichen. (hab nicht mehr zur KL gesagt, keine Formel). Das zieht die Prädiktionen zu $mean=0$ und $var=1$ und gibt uns einen kompakten latent space.

P: Wenn ich jetzt den Latent Space nach Eigenschaft sortieren will, wie mach ich das?

I: Zusätzlich zu Dekoder ein FeedForward Network, dass aus dem latent space eigenschaften prädiziert, das sortiert den Latent Space dann?

P: Das tut es nur wenn was vorliegt?

I: KA

P: Das Feedforward Network muss linear sein, sonst ist der Latent space nicht linear sortiert. Kommen wir noch zu Reinforcement Learning, was waren da so die Begriffe?

I: Agent in Environment, States mit Markov Eigenschaft, Aktionen die in States möglich sind und den agenten in einen anderen state bringen, Rewards die man bekommt weil man Aktion im State gemacht hat und die Policy, die mir sagt in welchem State ich welche Aktion machen soll

P: Brauch ich ne Policy?

I: Hmm, 100% Exploration ist auch ne Policy mMn und nur den Q values folgen eig auch.

P Exploration exploitation?

I: e greedy, Zahl mit der ne Zufallszahl jeden iterationsschritt verglichen wird, wenn kleiner als e nimmt man eine zufällige Aktion, sonst die mit dem höchsten Q Value. E wird über das Training hin angepasst.

P: Was sind Q values?

I: $Q(s,a)^* = R(s,a) + \gamma(\max_{a'} Q(s',a'))$, γ ist discount über zukünftige Rewards, Q gibt mit erwartungswert des Rewards wenn ich im state s aktion a mache.

P: Wie lernt man jetzt?

I: Q-Learning, große Tabelle mit allen state und actions. Initialisiert mit entweder Zufallswerten oder 0. Vergleiche aktuellen $Q(s,a)$ Value aus der Tabelle mit dem erhaltenen Reward und dem discounteten maximalen Q Value im state

s' (den nimmt man auch aus der Tabelle). Dann nimm ne learning Rate α und vergleiche initialen Q wert mit dem Target. $Q(s,a) = Q(s,a) + \alpha (R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a))$.

P: Die Tabelle hat ja bestimmte Limitationen. Welche?

I: Wenn States kontinuierlich sind oder es sehr viele diskrete gibt nimmt man ein DeepQ Network und macht DeepQ-Learning. Ein Netzwerk bekommt den State als input und prädiziert die Q Values für jede mögliche Aktion. Target Q value ist wieder Reward + max q value im nächsten state. Den Max q value im nächsten State bekommen wir vom gleichen netz aber instabil. Deswegen Aufteilung in 2 Netze, identische Architektur und identische initiale Gewichte, aber Target Network Gewichte bleiben erstmal fest und die Gewichte vom QNetwork werden alle X Iterationsschritte zum Targetnetwork rüberkopiert. Verhindert „Hund jagt eigenen Schwanz“, nach einem Backprop vom QNetwork würden sich sonst auch sofort die Targets die es lernen soll ändern. Dann noch Replay Memory, enthält $(s, a, s', R(s,a))$ wird aufgebaut und Netz wird nicht direkt nach jeder Aktion trainiert. Stattdessen wird das im replay memory gespeichert und wenn das voll ist wird daraus gesampled um das Netz zu trainieren. Das macht man weil nacheinanderfolgende states oft sehr stark korrelieren.

P: Wir hatten ein Beispiel für RL in der Vorlesung. Was war das?

I: Es gab 3, (nehmen Sie einfach eins davon, ich hab das erste genommen). 2 NNs, das erste erzeugt SMILES Codes, das zweite prädiziert anhand des SMILES Codes Eigenschaften. Beide sind vortrainiert und bereits gut. Jetzt in RL rein: State ist der Aktuelle SMILES Code, actions ist das hinzufügen eines Characters oder Symbols, Reward ist hoch wenn Prädiktor gute Eigenschaften prädiziert. Mit RL lernt der Generator jetzt gewünschte Moleküle zu prädizieren. Benutzt kein DeepQ Learning sondern den Reinforce algorithmus. Prädiktor wird natürlich nicht weiter trainiert.