

KI1: Einführung in die künstliche Intelligenz

Wintersemester 2023/24

Klausur



Die Codeteile liegen im bekannten Gitlab unter https://git-kik.hs-ansbach.de/2023_wise_ki1/2023_ki1_klausur. Der Projektname darf beim Fork in die Gruppe <benutzername>_and_dozent2023 nicht geändert werden.

Zur Beantwortung der Theoriefragen ist die Datei *abgabe.py* zu verwenden. Die Programmieraufgaben sind im jeweiligen Code zu bearbeiten.

Aufgaben:

1. (1 Punkt) Fülle deinen Namen und deine Matrikelnummer in *abgabe.py* korrekt aus.
2. (2 Punkte) Nenne zwei Beispiele von KI in der Medizin! [2 Stichpunkte]
3. (1 Punkt) Was ist der Unterschied zwischen Klassifikation und Regression? [1-2 Sätze]
4. (2 Punkte) Was unterscheidet parametrische und nicht-parametrische Algorithmen? [2-3 Sätze]
5. (2 Punkte) Nenne jeweils ein Beispiel für einen parametrischen und einen nicht-parametrischen Algorithmus! [2 Stichpunkte]
6. (2 Punkte) Nenne zwei typische Anzeichen dafür, dass ein maschinelles Lernmodell overfitted ist. [2 Stichpunkte]
7. (1 Punkt) Nenne eine Methode zur Verhinderung von Overfitting in maschinellen Lernmodellen. [1 Stichpunkt]
8. (2 Punkte) Nenne zwei Methoden der Merkmalsselektion. [2 Stichpunkte]
9. (1 Punkt) Nenne ein Vorteil von Segmentierung? [1 Stichpunkt]
10. (3 Punkte) Welche Probleme können sich ergeben, wenn bei der Evaluation eines Klassifikators nur auf die Genauigkeit (Accuracy) als Metrik geachtet wird? Welche Alternativen gibt es zusätzlich? [2-3 Sätze]
11. (3 Punkte) Aus welchen Bestandteilen besteht ein künstliches Neuron? [5 Begriffe]
12. (3 Punkte) Beschreibe die Berechnungen, die in einem einzelnen Neuron (lineare Schwellwerteinheit) stattfinden. [1-2 Sätze + Formel]

13. (2 Punkte) Ein Entwickler plant die Erstellung eines neuronalen Netzes zur Klassifikation von 20 verschiedenen Obstsorten anhand von Bildern. Die Bilder haben nach den Vorverarbeitungsschritten eine Größe von 100 * 100 Pixel und sind als Grauwertbilder abgespeichert. Als Auswertungsmetrik soll die top5 Accuracy verwendet werden, und die Labels sind im one-hot encoded Format vorliegend. Bestimme die Anzahl der Eingangsneuronen und Ausgangsneuronen für dieses neuronale Netz. [2 Zahlen oder ganze Rechnung]

14. (1 Punkt) Welche Größe hat das "gefilterte Bild 1" nach der Filterung von "Bild 1" mit dem angegebenen Filter (ohne Randbehandlung)?

Filter:

1	2
3	4

X

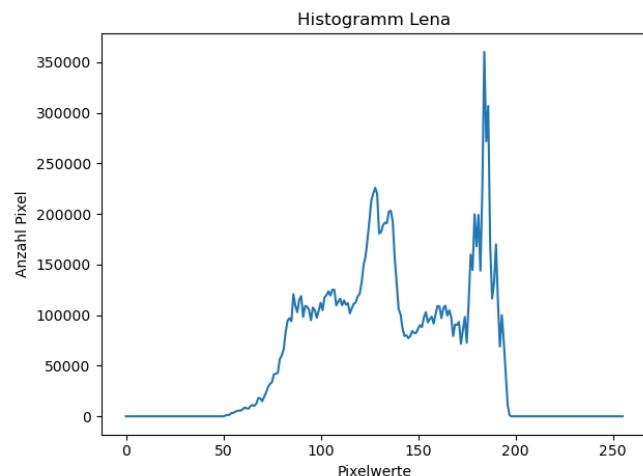
1	3	2
4	2	2
3	1	3

15. (4 Punkte) Filtere das "Bild 1" mit dem angegebenen Filter (ohne Randbehandlung). [Ergebnis oder ganze Rechnung]

16. (2 Punkte) Ein Thermometer an der Außenwand eines Zuges misst mit einer Frequenz von 2 Hz die Temperatur. Das Thermometer wird auf einer Fahrt von 10 min eingesetzt. Aus wie vielen Werten besteht die Aufnahme? [Ergebnis mit Rechenweg]

17. (3 Punkte) Erläutere den Unterschied und die versch. Einsatzgebiete des Validierungs- und Testdatensatzes. Erläutere dabei die Wörter "Validierung" und "Testen" des ML-Modells. [2-3 Sätze]

Abbildung 18: Einzelfoto Lena



18. (2 Punkte) Abbildung 18 zeigt ein Foto von Lena und das dazugehörige Histogramm (des Graubildes). Du wirst als Bildverarbeitungsexperte gefragt, ob man an diesem Foto noch etwas verbessern könnte. Benenne und begründe eine geeignete Methode zur Vorverarbeitung von Bildern, welche auf das gegebene Beispiel angewendet werden kann. [1-2 Sätze]

19. (2 Punkte) Welches Ziel verfolgt die Hauptkomponentenanalyse (PCA) bei der Verarbeitung von Daten? [1-2 Sätze]
20. (2 Punkte) Ein Mailclient möchte Spammails erkennen und bekommt, mithilfe eines ML-Algorithmus, die in Tabelle 20 gezeigten Ergebnisse. Die "positive" Klasse ist als Spam definiert.

Berechne die Sensitivity und die Specificity. [Zwei Zahlen mit Rechnung]

Tabelle 20: Confusion Matrix

		Wahre Ergebnisse	
		Spam	kein Spam
		Positiv	Negativ
Vorhersage	Spam / Positiv	100	45
	kein Spam / Negativ	5	350
		105	395
			500

21. (1 Punkt) Berechne die Accuracy von Tabelle 20. [Eine Zahl mit Rechnung]

Programmieren

Ziel der Programmieraufgaben ist es, ein Modell zu trainieren, welches die Bilder des MNIST-Datensatzes klassifiziert. Der MNIST-Datensatz besteht aus handgeschriebenen Zahlen von 0 bis 9. Die Datei *prog_given.py* und die Variablennamen müssen und dürfen nicht verändert werden.

22. (2 Punkte) Wähle zwei Modelle aus, welches du für die Klassifizierung der MNIST-Bilder verwenden möchtest. Zur Auswahl stehen die im Ordner docs/Modelle zu findenden Algorithmen. Begründe deine Wahl (in *abgabe.py*). [1-2 Sätze]
23. (4 Punkte) Trainiere die beiden ausgewählten Modelle mit den Trainingsdaten. Dazu den Code in der *prog_main.py* erweitern.
24. (2 Punkte) Wende die beiden trainierten Modelle auf die Testdaten an. Dazu den Code in der *prog_main.py* erweitern.
25. (6 Punkte) Implementiere die Funktion *conf_matrix2accuracy()* in der Datei *prog_own.py*. Die Funktion bekommt eine Confusion Matrix (Multiclass) übergeben und soll die zugehörige Accuracy berechnen. Die Confusion Matrix für das gegebene Problem hat die Klassen 0 - 9 (insg. 10 Klassen). Für diese Funktion darf die Bibliotheksfunktion *sklearn.metrics.accuracy_score* **nicht** genutzt werden.
26. (2 Punkte) Bewerte die beiden Modelle anhand der Accuracy und der Confusion Matrix. Welches Modell ist besser? Begründe deine Antwort (in *abgabe.py*). [1-2 Sätze]

27. (6 Punkte) Implementiere die Normierung der Bilddaten auf den Wertebereich [0,1]. Dazu steht die Funktion *normalize_images()* in der Datei *prog_own.py* zur Verfügung. Für diese Funktion darf die Bibliotheksfunktion *sklearn.preprocessing.MinMaxScaler* nicht genutzt werden.

Hinweis: keine der vorhergehenden Programmieraufgaben wird benötigt.

28. (1 Punkt) Code ist ohne Fehler importier- und ausführbar.

29. (1 Bonuspunkt) Abgabe des Source Codes sowie der Datei abgabe.py als Commit im geforkten Gitlabprojekt.

Viel Erfolg!

Bearbeitungszeit 90 Minuten