**Projektbeschreibung: Zweistufige stochastische Programmierung für die Produktionsprogrammplanung**

Im Rahmen meines Hiwi-Jobs werde ich an einem Projekt arbeiten, das sich mit der Produktionsprogrammplanung unter unsicherer Verfügbarkeit von Repetierfaktoren beschäftigt. Dabei geht es darum, ein zweistufiges stochastisches Programm zu entwickeln, das Unsicherheiten in der Verfügbarkeit von Ressourcen wie Rohstoffen oder Maschinen in die Planung einbezieht.

**Ziele des Projekts:**

* Ich möchte ein Modell erstellen, das in zwei Phasen funktioniert: Zuerst werden Entscheidungen getroffen, bevor die Unsicherheiten bekannt sind, und dann können Anpassungen vorgenommen werden, wenn die Unsicherheiten eintreten.
* Ich werde eine einfache Sampling-Approximation implementieren, die ohne komplizierte Verfahren zur Varianzreduzierung auskommt. Das hilft mir, ein gutes Verständnis für Sampling-Methoden zu entwickeln.
* Außerdem plane ich, das stochastische Modell mit einem deterministischen Modell zu vergleichen, um zu sehen, wie sich die Berücksichtigung von Unsicherheiten auf die Entscheidungen auswirkt.
* Wenn die Zeit reicht, möchte ich auch eine rollierende Planung einführen, bei der Entscheidungen regelmäßig an die aktuellen Gegebenheiten angepasst werden.

**1. Was ist ein Feedforward Neural Network (FNN), und wie funktioniert es?**

**Antwort:**  
Ein **Feedforward Neural Network (FNN)** ist die einfachste Form eines neuronalen Netzwerks, bei dem Informationen nur in eine Richtung fließen: von den Eingabeneuronen über versteckte Schichten hin zu den Ausgabeneuronen. Es gibt keine Zyklen oder Rückkopplungen. Jede Schicht ist vollständig mit der nächsten Schicht verbunden, was es einem FNN ermöglicht, eine Vielzahl von Mustern zu erkennen und Funktionen zu approximieren.

* **Vorteile**: Einfach und effektiv für viele Aufgaben. FNNs sind gut für Aufgaben geeignet, die keine zeitliche oder sequentielle Abhängigkeit erfordern, wie Bildklassifikation und einfache Regressionen.
* **Nachteile**: Da FNNs keine Informationen aus der Vergangenheit speichern können, sind sie weniger geeignet für Aufgaben, bei denen zeitliche oder sequentielle Zusammenhänge wichtig sind, wie z. B. Sprach- oder Textverarbeitung.

**2. Wie unterscheiden sich Convolutional Neural Networks (CNNs) von Feedforward Neural Networks (FNNs), und warum sind CNNs besonders nützlich für Bildverarbeitungsaufgaben?**

**Antwort:**  
**Convolutional Neural Networks (CNNs)** sind neuronale Netzwerke, die speziell für die Verarbeitung und Klassifikation von Bilddaten entwickelt wurden. CNNs verwenden **Faltungsschichten** (Convolutional Layers), die kleine Filter (Kerne) auf Bilddaten anwenden, um lokale Merkmale wie Kanten, Ecken und Texturen zu extrahieren.

* **Unterschiede zu FNNs**: Während FNNs vollständig verbunden sind und alle Neuronen miteinander verknüpfen, nutzen CNNs Konvolutions- und Pooling-Schichten, die die Anzahl der Parameter drastisch reduzieren und die Berechnungseffizienz erhöhen. Dies ermöglicht CNNs, komplexe visuelle Muster zu erkennen.
* **Vorteile**: Durch die Extraktion von lokalen Merkmalen sind CNNs besonders robust gegenüber Verschiebungen und Verformungen im Bild und benötigen weniger Rechenleistung als vollständig verbundene Netzwerke.
* **Anwendung**: CNNs werden für Bildklassifikation, Objekterkennung und Segmentierungsaufgaben verwendet, aber auch für andere strukturierte Daten wie Audio oder Text.

**3. Was sind Recurrent Neural Networks (RNNs), und wie funktionieren sie?**

**Antwort:**  
**Recurrent Neural Networks (RNNs)** sind eine Art neuronales Netzwerk, das für die Verarbeitung von Sequenzdaten entwickelt wurde. RNNs besitzen **Rückkopplungsverbindungen**, die es ihnen ermöglichen, Informationen von vorherigen Zeitschritten zu speichern und zukünftige Schritte zu beeinflussen. In einem RNN fließen die Informationen durch "Zustände" von einem Zeitschritt zum nächsten.

* **Vorteile**: Die Rückkopplung macht RNNs ideal für Aufgaben, die zeitliche Abhängigkeiten haben, wie Sprachverarbeitung, Zeitreihenanalyse und maschinelles Übersetzen.
* **Nachteile**: RNNs können das **Vanishing Gradient Problem** haben, was das Training erschwert und ihre Fähigkeit einschränkt, langfristige Abhängigkeiten zu lernen.

**4. Warum wurden LSTM- und GRU-Architekturen entwickelt, und wie verbessern sie RNNs?**

**Antwort:**  
**LSTM (Long Short-Term Memory)** und **GRU (Gated Recurrent Unit)** sind spezielle RNN-Architekturen, die entwickelt wurden, um das Vanishing Gradient Problem in Standard-RNNs zu beheben. Beide nutzen **Gating-Mechanismen**, die steuern, wie viel Information von einem Zeitschritt zum nächsten weitergegeben wird.

* **LSTM**: LSTMs verwenden drei verschiedene "Gates" (Input, Forget und Output), um Informationen zu speichern, zu vergessen oder weiterzugeben, was sie besser für das Lernen langfristiger Abhängigkeiten macht.
* **GRU**: GRUs sind eine vereinfachte Variante von LSTMs mit weniger Gates (Update und Reset), was sie recheneffizienter macht, jedoch weniger flexibel als LSTMs.
* **Verbesserung**: Durch diese Mechanismen können LSTM und GRU Modelle effektiver lernen, welche Informationen beibehalten und welche verworfen werden sollen, was die Leistung bei der Modellierung langanhaltender Abhängigkeiten deutlich verbessert.

**5. Was ist der Unterschied zwischen RNNs und CNNs, und wann sollte man welche Architektur verwenden?**

**Antwort:**  
Der Hauptunterschied zwischen **RNNs** und **CNNs** liegt in ihrem Aufbau und ihren Anwendungsbereichen.

* **RNNs**: RNNs sind für sequentielle und zeitabhängige Daten geeignet. Sie sind gut für natürliche Sprachverarbeitung, Zeitreihen und Aufgaben, bei denen die Reihenfolge der Daten wichtig ist.
* **CNNs**: CNNs sind für räumlich strukturierte Daten (z. B. Bilder) entwickelt. Sie sind hervorragend in der Lage, visuelle Muster wie Formen und Texturen zu erkennen, unabhängig von ihrer Position im Bild.

**Wann verwenden?**

* Verwenden Sie **RNNs** für zeitabhängige Aufgaben wie maschinelle Übersetzung, Spracherkennung oder Analyse von Sensorik-Daten.
* Verwenden Sie **CNNs** für Aufgaben, die eine räumliche Datenstruktur haben, wie Bildklassifikation, Objekterkennung oder medizinische Bildverarbeitung.

**6. Wie funktionieren Activation Maps und Feature Maps in einem CNN?**

**Antwort:**  
**Activation Maps** und **Feature Maps** sind die Ausgaben, die durch die Anwendung von Filtern auf die Eingabedaten in einem CNN entstehen.

* **Feature Maps**: Eine Feature Map ist das Ergebnis einer Faltungsschicht, das die Positionen und Stärke bestimmter Merkmale in den Eingabedaten darstellt. Jedes Filter extrahiert spezifische Merkmale (z. B. Kanten, Texturen).
* **Activation Maps**: Nach der Faltung wird häufig eine Aktivierungsfunktion (wie ReLU) angewendet, um die Nichtlinearität zu erhöhen. Die Aktivierungen auf der Feature Map stellen dann die Aktivationskarte dar, die anzeigt, wie „aktiv“ jedes Merkmal ist.

**7. Welche Rolle spielt die Pooling-Schicht in einem CNN, und welche Arten von Pooling gibt es?**

**Antwort:**  
Pooling-Schichten reduzieren die räumliche Größe der Feature Maps, um die Anzahl der Parameter und Berechnungen im Netzwerk zu verringern.

* **Max-Pooling**: Wählt den maximalen Wert in einem bestimmten Bereich aus. Es betont die wichtigsten Merkmale und reduziert das Rauschen.
* **Average-Pooling**: Berechnet den Durchschnittswert in einem Bereich und sorgt für eine gleichmäßigere Verkleinerung der Bilddaten.

Pooling macht das Netzwerk robuster gegen kleine Verschiebungen in den Eingabedaten und verringert die Wahrscheinlichkeit, dass es überanpasst wird.

**8. Warum ist das Vanishing-Gradient-Problem ein großes Problem in tiefen RNNs, und wie wird es in modernen Netzwerken gelöst?**

**Antwort:**  
Das **Vanishing-Gradient-Problem** tritt auf, wenn die Gradienten in tieferen Schichten eines Netzwerks extrem klein werden und somit die Aktualisierung der Gewichte stoppt. In RNNs bedeutet dies, dass das Netzwerk Schwierigkeiten hat, Informationen über lange Zeiträume zu lernen und zu speichern.

* **Lösungen**: Um dieses Problem zu lösen, wurden LSTM- und GRU-Zellen entwickelt, die durch ihre Gating-Mechanismen den Gradientenerhalt über längere Sequenzen verbessern. Eine weitere Lösung ist die Verwendung von Residual Connections (wie in Residual Networks), die die Gradienten durch das Netzwerk leiten und verhindern, dass sie verschwinden.

**9. Wie unterscheiden sich die Einsatzgebiete von TensorFlow und PyTorch, und welche Frameworks eignen sich am besten für bestimmte Anwendungen?**

**Antwort:**

* **TensorFlow**: Bekannt für seine Produktionsreife und unterstützende Tools wie TensorFlow Serving. TensorFlow ist besonders gut geeignet für den Einsatz in Unternehmensanwendungen und mobile Implementierungen.
* **PyTorch**: Beliebt für Forschung und Entwicklung, da es eine dynamische Berechnungsgrafik verwendet und das Debuggen erleichtert. PyTorch wird oft in der akademischen Forschung und in schnell entwickelnden Projekten verwendet.

Beide Frameworks unterstützen neuronale Netzwerke gut, wobei TensorFlow in der Industrie häufiger anzutreffen ist und PyTorch mehr in der Forschung verwendet wird.