### **Fault Detection and Diagnosis in AI (ECU-ITAI-3-02)**

#### **1. التعريفات الأساسية والمفاهيم المهمة**

قد يُطلب منك تعريف أو شرح بعض المصطلحات الأساسية، مثل:

* **Syntax Errors vs Logical Errors** (أخطاء بناء الكود مقابل الأخطاء المنطقية).
* **Overfitting vs Underfitting** (الإفراط في التعلّم مقابل نقص التعلّم).
* **Confusion Matrix** (مصفوفة الالتباس).
* **Cross-Validation** (التحقق المتقاطع).
* **Precision, Recall, and F1-score** (الدقة، الاسترجاع، ومعامل F1).
* **Data Preprocessing** (معالجة البيانات المسبقة).

#### **2. خطوات استكشاف الأخطاء وإصلاحها (Troubleshooting AI Systems)**

* **ما الخطوات العامة لاختبار وتصحيح أخطاء أنظمة الذكاء الاصطناعي؟**
  + **تحليل البيانات** والتأكد من جودتها.
  + **اختبار الوحدات البرمجية بشكل فردي (Unit Testing).**
  + **اختبار النظام ككل بعد دمج المكونات (Integration Testing).**
  + **تحليل الأخطاء وتصحيحها (Error Analysis & Debugging).**
  + **تحسين أداء النظام بناءً على التقييمات والاختبارات.**

✍ **سؤال متوقع:**

ما أهمية اختبار الأنظمة الذكية وما الخطوات الأساسية لاختبار وتصحيح الأخطاء في أنظمة الذكاء الاصطناعي؟

#### **3. أنواع الأخطاء في أنظمة الذكاء الاصطناعي**

* **Syntax Errors** → خطأ في كتابة الكود يمنع تشغيله.
* **Logical Errors** → خطأ في منطق البرنامج يؤدي إلى نتائج غير صحيحة.

✍ **سؤال متوقع:**

ما الفرق بين الأخطاء البرمجية Syntax Errors وLogical Errors؟ مع ذكر مثال لكل نوع.

#### **4. مشاكل اختلال توازن البيانات (Class Imbalance)**

* عندما يكون هناك فئة تمثل نسبة أكبر بكثير من باقي الفئات في مجموعة البيانات.
* الحلول تشمل:
  + **Oversampling**: زيادة العينات في الفئات الأقل تمثيلًا.
  + **Undersampling**: تقليل عدد العينات من الفئات الأكثر تمثيلًا.
  + **SMOTE**: إنشاء بيانات صناعية للفئة الأقل تمثيلًا.

✍ **سؤال متوقع:**

كيف يمكن معالجة مشكلة عدم توازن الفئات في بيانات التدريب؟

#### **5. تقنيات تحسين أداء أنظمة الذكاء الاصطناعي**

* **Data Augmentation**: زيادة حجم البيانات باستخدام تقنيات مثل التدوير والانعكاس.
* **Model Selection**: اختيار النموذج المناسب (مثل CNN بدلاً من Naïve Bayes).
* **Hyperparameter Tuning**: ضبط المعلمات للحصول على أداء أفضل.

✍ **سؤال متوقع:**

كيف يمكن تحسين أداء نموذج ذكاء اصطناعي يستخدم في تصنيف الصور؟

#### **6. تقييم أداء أنظمة الذكاء الاصطناعي (Model Evaluation)**

* **Accuracy** → النسبة المئوية للتوقعات الصحيحة.
* **Precision** → نسبة التوقعات الصحيحة من بين جميع التوقعات الإيجابية.
* **Recall** → نسبة التوقعات الصحيحة مقارنة بالحالات الفعلية.
* **F1-score** → مقياس يوازن بين Precision و Recall.
* **ROC AUC** → قياس أداء النموذج في تصنيف البيانات.

✍ **سؤال متوقع:**

لديك بيانات تقييم لنموذج ذكاء اصطناعي، كيف يمكنك تحديد ما إذا كان النموذج يعمل بشكل جيد؟

#### **7. خطوات تدريب نموذج الشبكات العصبية التلافيفية (CNN) من البداية إلى التقييم**

* **تحضير البيانات** (تنظيف، تحويل، تقسيم إلى تدريب واختبار).
* **بناء النموذج** (إضافة طبقات Convolution وPooling وDense).
* **تدريب النموذج** (باستخدام خوارزمية مثل Adam مع loss function مناسبة).
* **تقييم الأداء** باستخدام **Confusion Matrix** و **Precision, Recall, F1-score**.

✍ **سؤال متوقع:**

ما الخطوات الأساسية لتدريب نموذج ذكاء اصطناعي يعتمد على الشبكات العصبية التلافيفية (CNN)؟

### **📌 خلاصة: أهم الأسئلة المتوقعة في الامتحان**

1. **ما الفرق بين الأخطاء المنطقية والأخطاء البرمجية؟ مع مثال لكل نوع.**
2. **كيف يمكن اختبار أنظمة الذكاء الاصطناعي وتصحيح أخطائها؟**
3. **ما أهمية معالجة اختلال توازن الفئات في البيانات؟ وما الطرق المستخدمة لحلها؟**
4. **اذكر بعض الطرق المستخدمة لتحسين أداء أنظمة الذكاء الاصطناعي.**
5. **كيف يتم تقييم أداء نموذج تصنيف الصور؟ وما المقاييس المستخدمة؟**
6. **ما الفرق بين Precision و Recall و F1-score؟**
7. **ما الخطوات الأساسية لبناء وتدريب نموذج CNN لتصنيف الصور؟**  
     
     
     
     
     
     
     
     
     
     
   https://drive.google.com/drive/folders/1BbFBsFyxhILYG7ILdcUNfTmtQLcZiWI1

### 🔥 الاستعداد للامتحان: اكتشاف الأخطاء والتشخيص في الذكاء الاصطناعي 🔥

## **1️⃣ فهم الأخطاء البرمجية في الذكاء الاصطناعي**

### **🔷 الشرح بالعربي**

في أنظمة الذكاء الاصطناعي، تحدث الأخطاء عادة بسبب مشاكل في الكود البرمجي أو البيانات. هناك نوعان رئيسيان من الأخطاء:

✅ **1. الأخطاء التركيبية (Syntax Errors)**

* هذه الأخطاء تحدث عند كتابة كود غير صحيح وفقًا لقواعد اللغة البرمجية.
* تمنع البرنامج من التنفيذ حتى يتم إصلاحها.
* **مثال في Python:**
* print("Hello World" # خطأ: نسيان إغلاق القوس

✅ **2. الأخطاء المنطقية (Logical Errors)**

* هذه الأخطاء تحدث عندما يكون الكود صحيحًا من الناحية التركيبية لكنه يعطي نتائج غير صحيحة.
* يصعب اكتشافها لأنها لا تمنع الكود من التنفيذ، لكنها تؤدي إلى قرارات خاطئة.
* **مثال:**
* def circle\_area(radius):
* return radius / 3.14 # خطأ منطقي، يجب أن يكون الضرب (\*)

📌 **كيف يتم تصحيح هذه الأخطاء؟**

1. مراجعة الكود والتحقق من الأخطاء البرمجية باستخدام محرر الأكواد.
2. إجراء اختبارات على أجزاء صغيرة من الكود أولًا لتحديد المشكلة.
3. استخدام أدوات التصحيح (Debugging Tools) مثل print() أو debugger في بيئة التطوير.

### **🔷 Explanation in English**

In artificial intelligence systems, errors usually occur due to problems in the code or data. There are two main types of errors:

✅ **1. Syntax Errors**

* These errors occur when the code is written incorrectly according to the programming language rules.
* They prevent the program from running until fixed.
* **Example in Python:**
* print("Hello World" # Error: Missing closing parenthesis

✅ **2. Logical Errors**

* These errors occur when the code is syntactically correct but produces incorrect results.
* They are harder to detect because they don’t stop the program from running, but they lead to wrong decisions.
* **Example:**
* def circle\_area(radius):
* return radius / 3.14 # Logical error: Should be multiplication (\*)

📌 **How to fix these errors?**

1. Review the code and check for syntax errors using a code editor.
2. Test small parts of the code first to identify the problem.
3. Use debugging tools like print() or a debugger in the development environment.

## **2️⃣ تحسين أداء نماذج الذكاء الاصطناعي**

### **🔷 الشرح بالعربي**

بعد تصحيح الأخطاء، تحتاج النماذج إلى تحسين أدائها للحصول على دقة أعلى. هناك عدة طرق لتحقيق ذلك:

✅ **1. معالجة عدم توازن البيانات (Class Imbalance Handling)**

* عندما يكون هناك عدد كبير من البيانات في فئة معينة وعدد قليل في فئة أخرى، يؤدي ذلك إلى نتائج غير دقيقة.
* **الحلول تشمل:**
  + **Oversampling:** تكرار العينات من الفئة الأقل تمثيلًا.
  + **Undersampling:** حذف بعض العينات من الفئة الأكثر تمثيلًا.
  + **SMOTE (Synthetic Minority Over-sampling Technique):** إنشاء عينات جديدة اصطناعية.

✅ **2. اختيار النموذج المناسب (Model Selection)**

* بعض النماذج تعمل بشكل أفضل في مهام معينة.
* **مثال:**
  + **Naïve Bayes** يستخدم للبيانات النصية لكنه غير مناسب للصور.
  + **CNN (Convolutional Neural Network)** هو الأفضل لتصنيف الصور.

✅ **3. تحسين المعلمات (Hyperparameter Tuning)**

* ضبط القيم مثل **Learning Rate, Batch Size, Number of Layers** للحصول على نتائج أفضل.

### **🔷 Explanation in English**

After fixing errors, models need to be optimized for better accuracy. There are several ways to achieve this:

✅ **1. Handling Class Imbalance**

* When one class has significantly more data than another, the model may perform poorly.
* **Solutions include:**
  + **Oversampling:** Duplicating samples from the minority class.
  + **Undersampling:** Removing samples from the majority class.
  + **SMOTE:** Creating synthetic samples for the minority class.

✅ **2. Model Selection**

* Some models work better for specific tasks.
* **Example:**
  + **Naïve Bayes** is good for text data but not for images.
  + **CNN (Convolutional Neural Network)** is the best choice for image classification.

✅ **3. Hyperparameter Tuning**

* Adjusting parameters like **Learning Rate, Batch Size, Number of Layers** for better results.

## **3️⃣ تنفيذ الحلول باستخدام Python و Keras**

### **🔷 الشرح بالعربي**

📌 **كود لبناء نموذج CNN بسيط لتصنيف الصور:**

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

model = keras.Sequential([

layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 1)),

layers.MaxPooling2D(pool\_size=(2, 2)),

layers.Flatten(),

layers.Dense(128, activation='relu'),

layers.Dense(4, activation='softmax')

])

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.summary()

### **🔷 Explanation in English**

📌 **Code to build a simple CNN model for image classification:**

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

model = keras.Sequential([

layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 1)),

layers.MaxPooling2D(pool\_size=(2, 2)),

layers.Flatten(),

layers.Dense(128, activation='relu'),

layers.Dense(4, activation='softmax')

])

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.summary()

🚀 **باتباع هذه الخطوات، ستكون جاهزًا لأي سؤال في الامتحان!** 💯🔥