



جامعة الكويت
مركز البحوث والدراسات والنشر



ISBN: 978-9922-726-71-7

موسوعة أساسيات الحاسوب الحاسوب والرياضيات لعلوم البيانات والذكاء الاصطناعي دليل شامل

تأليف

م. دعاء طالب زيدان الموسوي
م.م. حسنين فليح حسن الزركاني
م.م. زين العابدين عباس ناصر الصافي
م.م. حسنين علي ذويب العجيلي

منشورات

مركز البحوث والدراسات والنشر
جامعة الكوت



٠٠٤

م ٩٥٩ الموسوي ، دعاء طالب زيدان .

موسوعة اساسيات الحاسوب : الحاسوب والرياضيات / دعاء طالب زيدان الموسوي،

زين العابدين عباس، حسنين علي . - ط١. - بغداد: مطبعة جامعة الكوت،

لسنة ٢٠٢٦ .

١١٤ ص ؛ ٢٤ سم .

١- الحاسوب ، علم - أ - عباس ، زين العابدين (م.م)

ب - علي، حسنين (م.م) ج - العنوان.

رقم الايداع

٢٠٢٦ / ١٣٤٨

المكتبة الوطنية/الفهرسة اثناء النشر

رقم الايداع في دار الكتب والوثائق ببغداد

١٣٤٨ لسنة ٢٠٢٦ م

ISBN: 978-9922-726-71-7

ملاحظة

مركز البحوث والدراسات والنشر في كلية الكوت الجامعة
غير مسؤول عن الافكار والرؤى التي يتضمنها الكتاب
والمسؤول عن ذلك الكاتب او الباحث فقط.



الإهداء

بسم الله الرحمن الرحيم

إلى كل من قدّم دمه وروحه فداءً للوطن،
وإلى كل شهيدٍ سقى الوطن بدمه،
يا من رسمتم ببطولاتكم ملامح الشجاعة
والإيمان، وكنتم مثلاً للتضحية والفداء.

(شهداء الوطن العربي)

إلى الذين واجهوا الظلم والطغيان
وسطروا بدمانهم أروع الملاحم البطولية.

(شهداء العراق)

إلى الأرواح الطاهرة التي عانقت السماء
وخلدت في قلوب الأجيال.

(شهداء غزة ولبنان)

نهدي هذا الكتاب وفاءً لذكراكم العطرة،
وسوف تبقون مصدر إلهام لنا
في طريق الحق والعدالة.

رحمكم الله وأسكنكم مع أصحاب الحسين (عليه السلام).

تقديم الكتاب

موسوعة أساسيات الحاسوب والرياضيات

في ظل التحولات العلمية والتقنية المتسارعة التي يشهدها العالم المعاصر، برزت علوم الحاسوب والرياضيات بوصفها الدعامة الأساسية التي تستند إليها مختلف التخصصات العلمية والتطبيقية، لما لهما من دور محوري في بناء النظم الحديثة، وتحليل البيانات، وتطوير الخوارزميات، ودعم الابتكار في مجالات الذكاء الاصطناعي وعلوم البيانات والتحول الرقمي. وانطلاقاً من هذه الأهمية الاستراتيجية، تبرز الحاجة الملحة إلى مراجع علمية أكاديمية رصينة تُسهم في ترسيخ المفاهيم الأساسية وتقديمها ضمن إطار منهجي متكامل.

تأتي موسوعة أساسيات الحاسوب والرياضيات استجابةً لهذه الحاجة، بوصفها عملاً علمياً توثيقياً شاملاً يهدف إلى تقديم الأسس النظرية والعملية لعلوم الحاسوب والبنية الرياضية الداعمة لها، وفق منهج علمي منظم ومتدرج، يراعي المعايير الأكاديمية المعتمدة في التعليم الجامعي. وقد تم إعداد هذه الموسوعة لتكون مرجعاً موحدًا يجمع بين الدقة العلمية، والوضوح المنهجي، والتكامل المعرفي، بما يضمن انسجام المحتوى مع متطلبات الخطط الدراسية الحديثة.

تغطي هذه الموسوعة طيفاً واسعاً من الموضوعات الأساسية، بدءاً من الإطار المفاهيمي والفلسفي لعلم الحوسبة، والتطور التاريخي للحاسوب، مروراً بالبنية المادية والبرمجية للحاسوب، ونظم التشغيل، والشبكات الحاسوبية، وأمن المعلومات، وانتهاءً بالمجالات المعاصرة المتقدمة مثل الحوسبة المتوازية والسحابية، والحوسبة الطرفية، والحوسبة الكمية، والذكاء الاصطناعي وتطبيقاته. وفي موازاة ذلك، تقدم الموسوعة الأساس الرياضي الضروري لفهم هذه المجالات، من خلال عرض علمي منضبط لموضوعات المنطق الرياضي، والجبر الخطي، والتفاضل والتكامل، ونظرية الاحتمالات، والإحصاء، وتحليل الخوارزميات.

وقد روعي في بناء فصول الموسوعة التسلسل المنطقي والترابط الموضوعي بين المفاهيم، بما يحقق تكاملاً معرفياً يربط بين الجانب النظري والتطبيقي، ويسهم في تنمية التفكير التحليلي والمنهجي لدى القارئ. كما تم اعتماد المصطلحات العلمية الدقيقة المتداولة أكاديمياً، مع تقديمها بلغة عربية علمية واضحة، تراعي سلامة التعبير ودقة الصياغة، وتواكب التطور العلمي المعاصر.

وتستهدف هذه الموسوعة طلبة الجامعات في المراحل الأولى والمتقدمة، وطلبة الدراسات العليا، وأعضاء الهيئات التدريسية، والباحثين، فضلاً عن المختصين في مجالات الحاسوب والرياضيات وعلوم البيانات،

بوصفها مرجعًا أكاديميًا يمكن الاعتماد عليه في التدريس والدراسة والتأهيل العلمي. كما تم إعدادها بما ينسجم مع متطلبات الاعتماد الأكاديمي وضمان الجودة في المؤسسات التعليمية.

إن موسوعة أساسيات الحاسوب والرياضيات تمثل إضافة نوعية للمكتبة العربية، وتسهم في دعم مسيرة التعليم العالي، وتعزيز البنية المعرفية العلمية، وتوفير مرجع موثوق يُعتمد عليه في إعداد المناهج وتطويرها، بما يواكب متطلبات العصر الرقمي ويخدم أهداف التنمية العلمية المستدامة.

المحتويات

- 1..... الفصل الأول: الإطار المفاهيمي لعلم الحاسوب وتطور الآلة
- 1..... 1.1 مقدمة نظرية: فلسفة الحوسبة ونظرية المعلومات
- 1..... 1.1.1 التعريف الأكاديمي: أطروحة تشيرش-تورينج (Church-Turing Thesis)
- 2..... 1.1.3 خصائص النظام الحاسوبي الحديث
- 3..... 1.1.4 دورة معالجة المعلومات (IPOS Cycle)
- 4..... 1.2 الجذور التاريخية للحوسبة: رحلة 5000 عام لأتمتة العقل
- 4..... 1.2.1 العصور القديمة: الحساب الميكانيكي والتمثيل الموضعي
- 5..... 1.2.3 عصر النهضة الميكانيكية (القرن 17-19)
- 6..... 1.3 ولادة الحاسوب الحديث: سنوات الحرب والتشفير (1936-1945)
- 7..... 1.4 أجيال الحاسوب الإلكترونية: التصنيف الشامل والمفصل
- 8..... 1.5 التصنيف المنهجي للحواسيب وأنواعها
- 8..... 1.5.1 حسب مبدأ معالجة البيانات
- 9..... 1.5.2 حسب الحجم والقدرة (Taxonomy of Computer Systems)
- 10..... 1.6 المماريات الحاسوبية (Computer Architectures)
- 10..... 1.6.1 معمارية فون نيومان (Von Neumann Architecture)
- 11..... 1.6.2 معمارية هارفارد (Harvard Architecture)
- 11..... 1.6.3 تصميم مجموعة التعليمات (ISA: CISC vs RISC)
- 12..... 1.7 الأساس المنطقي: البوابات الرقمية والجبر البوليني
- 13..... 1.7 الأساس المنطقي: البوابات الرقمية والجبر البوليني
- 14..... 1.8 أنظمة العد وتمثيل البيانات (Data Representation)
- 15..... 1.8.4 تمثيل النصوص (Text Encoding)

16	1.9 مقاييس الأداء وقوانين الحوسبة (Performance Laws)
16	1.9.1 معادلة أداء المعالج (The Iron Law of Performance)
16	1.9.2 قانون أمدال - (Amdahl's Law) قانون التشاؤم
16	1.9.3 قانون جوستافسون - (Gustafson's Law) قانون التفاؤل
17	الفصل الثاني: البنية المادية – (Hardware) التشريح التفصيلي للحاسوب
17	2.1 مقدمة في العتاد وتكامل المنظومة الحاسوبية
17	2.2 اللوحة الأم – (Motherboard) العمود الفقري للنظام
17	2.2.1 المكونات الرئيسية
18	2.2.2 عوامل الشكل (Form Factors)
18	2.3 وحدة المعالجة المركزية – (CPU) العقل المدبر
18	2.3.1 دورة التعليمات (Machine Cycle)
19	2.3.2 المكونات الداخلية
19	2.3.3 مفاهيم تقنية متقدمة
19	2.4 الذاكرة الرئيسية (RAM)
20	2.4.1 أجيال الذاكرة (DDR SDRAM)
20	2.4.2 معمارية القنوات (Channel Architecture)
20	2.4.3 الكمون والتوقيتات (Latency & Timings)
20	2.5 وحدات التخزين الثانوية
20	2.5.1 الأقراص الصلبة الميكانيكية (HDD)
20	2.5.2 أقراص الحالة الصلبة (SSD)
21	2.5.3 بروتوكولات النقل SATA vs NVMe
21	2.6 وحدة معالجة الرسوميات (GPU)
21	2.7 مزود الطاقة (PSU) وأنظمة التبريد

21	2.7.1 مزود الطاقة (PSU)
21	2.7.2 التبريد (Cooling)
22	2.8 وحدات الإدخال والإخراج (I/O)
23	الفصل الثالث: المعماريات المتقدمة والحوسبة الحديثة
23	3.1 المعالجات متعددة الأنوية ومتعددة الخيوط (Multicore and Multithreading)
23	3.1.1 قانون أمداال (Amdahl's Law)
23	3.1.2 المعالجة المتعددة الـمتماثلة (SMP)
23	3.2 الحوسبة المتوازية (Parallel Computing) وحوسبة الأداء العالي (HPC)
23	3.2.1 تصنيف فليين (Flynn's Taxonomy)
24	3.2.2 الحوسبة العنقودية (Cluster Computing)
24	3.3 وحدات التسريع GPU, TPU, NPU (Accelerator Units)
24	3.3.1 وحدة معالجة الرسومات (GPU) كحاسوب عام (GPGPU)
24	3.3.2 وحدة المعالجة الموترية (TPU - Tensor Processing Unit)
25	3.4 الحوسبة السحابية (Cloud Computing): IaaS, PaaS, SaaS
25	3.4.1 نماذج الخدمة (Service Models)
25	3.4.2 نماذج النشر (Deployment Models)
25	3.5 الحوسبة الطرفية (Edge Computing) وإنترنت الأشياء (IoT)
26	3.6 لمحة تمهيدية عن الحوسبة الكمية (Quantum Computing)
26	3.6.1 الكيوبت (Qubit) مقابل البت (Bit)
26	3.6.2 التشابك الكمي (Entanglement)
26	3.6.3 التأثير المستقبلي
27	الفصل الرابع: المنظومات البرمجية (Software) وبيئات التشغيل
27	4.1 مفهوم البرمجيات وتصنيفاتها

27	4.1.1 برمجيات النظام (System Software)
28	4.1.3 لغات البرمجة – (Programming Languages) لمحة تصنيفية
28	4.2 أنظمة التشغيل: المفهوم والوظائف الأساسية
29	4.4 نظم الملفات (File Systems): FAT32, NTFS, exFAT, ext4, APFS
30	4.5 بيئات العمل الافتراضية والحاويات (Virtualization & Containers)
30	4.5.1 المحاكاة الافتراضية (Virtualization)
30	4.5.2 الحاويات (Containers)
32	الفصل الخامس: الدليل التشغيلي لنظامي Windows 10 و Windows 11
32	5.1 التعرف على واجهة المستخدم الرسومية (GUI) وسطح المكتب
32	5.2 إدارة الملفات والمجلدات وتصنيفها
33	5.3 لوحة التحكم وتطبيق Settings وإدارة الأجهزة
33	5.4 إدارة المستخدمين والصلاحيات
34	5.5 أدوات الصيانة واستعادة النظام
34	5.6 الأوامر الأساسية عبر CMD و PowerShell
35	5.7 استكشاف الأخطاء ومعالجتها (Troubleshooting)
36	الفصل السادس: شبكات الحاسوب والبنية التحتية للإنترنت
36	6.1 مقدمة في علم الشبكات وأهميتها في العصر الرقمي
36	6.2 تصنيف الشبكات حسب النطاق الجغرافي
36	6.2.1 شبكة المنطقة الشخصية (PAN - Personal Area Network)
36	6.2.2 شبكة المنطقة المحلية (LAN - Local Area Network)
37	6.2.3 شبكة الحرم الجامعي (CAN - Campus Area Network)
37	6.2.4 شبكة المنطقة الإقليمية (MAN - Metropolitan Area Network)
37	6.2.5 شبكة المنطقة الواسعة (WAN - Wide Area Network)

37	6.3 طوبولوجيا الشبكات (Network Topology)
38	6.4 وسائط الإرسال (Transmission Media)
38	6.5 النماذج المرجعية OSI: TCP/IP
39	6.6 التجهيزات الشبكية (Network Equipment)
40	6.7 بروتوكولات الإنترنت الأساسية
41	الفصل السابع: الأمن السيبراني – الحصن الرقمي في عصر 2026
41	7.1 مبادئ أمن المعلومات (CIA) ثالوث
41	7.2 تصنيف التهديدات والبرمجيات الخبيثة (Malware)
42	7.2.1 الفيروسات (Viruses)
42	7.2.2 الديدان (Worms)
42	7.2.3 أحصنة طروادة (Trojans)
42	7.2.4 برامج الفدية (Ransomware)
42	7.2.5 برامج التجسس (Spyware) والـ Rootkits
44	7.5 علم التشفير (Cryptography)
44	7.6 استراتيجيات الدفاع
45	7.7 القوانين والتشريعات السيبرانية وحماية البيانات
46	8.1 مقدمة في الحزم المكتتبية وأدوارها المهنية والأكاديمية
46	8.2 معالجة النصوص (Word Processing)
46	8.2.1 التنسيق المتقدم وبناء المستندات الأكاديمية
47	8.2.2 أدوات البحث العلمي والمراجع
47	8.2.3 دمج المراسلات (Mail Merge)
47	8.3 الجداول الإلكترونية (Spreadsheets)
47	8.3.1 أنواع البيانات وتنسيق الخلايا

48	8.3.2 الدوال والصيغ المتقدمة
48	8.3.3 الجداول المحورية (Pivot Tables)
48	8.3.4 أدوات تحليل البيانات (Data Analysis Toolpak)
48	8.4 العروض التقديمية (Presentations)
49	8.4.1 مبادئ التصميم الجيد للعروض
49	8.4.2 الشرائح الرئيسية (Slide Master)
49	8.4.3 الحركات والانتقالات (Morph & Zoom)
49	8.5 أدوات التعاون السحابي (Cloud Collaboration)
50	الفصل التاسع: مدخل إلى البرمجة والخوارزميات (النسخة المرجعية الشاملة)
50	9.1 ما هي الخوارزمية؟ الفلسفة والنشأة والتكوين
50	9.1.1 التأصيل التاريخي واللغوي
50	9.1.2 الخصائص الخمس للخوارزمية الصحيحة (Knuth's 5 Properties)
51	9.1.3 الحتمية مقابل العشوائية (Determinism vs. Non-determinism)
51	9.2 مفاهيم البرمجة الأساسية: البنية التحتية للكود
53	9.3 التصميم الهندسي: الكود الزائف ومخططات التدفق
53	9.4 لغة Python: سكين الجيش السويسري لعلوم البيانات
55	9.5 هياكل البيانات: (Data Structures) تنظيم المعلومات
57	9.6 التفكير الخوارزمي (Computational Thinking)
57	9.7 تحليل تعقيد الخوارزميات (Algorithm Complexity Analysis)
59	9.8 أنماط البرمجة (Programming Paradigms)
60	الفصل العاشر: المنطق الرياضي ونظرية المجموعات
60	10.1 الجمل الرياضية والقضايا المنطقية (Propositions)
62	10.3 قوانين المنطق (Logical Laws)

62	10.4 نظرية المجموعات (Set Theory)
63	10.5 العلاقات والدوال
64	الفصل الحادي عشر: الرياضيات المتقطعة وهياكل البيانات الرياضية
64	11.1 المتتاليات والمتسلسلات البسيطة (Sequences and Series)
64	11.2 التوافيق والتباديل (Combinatorics)
65	11.3 الرسوم البيانية (Graphs) والأشجار (Trees)
66	11.5 تطبيقات الرياضيات المتقطعة في الشبكات والخوارزميات
68	الفصل الثاني عشر: الجبر الخطي والتحليل المتجهي
68	12.1 المتجهات (Vectors) في بعدين وثلاثة أبعاد
68	12.2 المحددات (Determinants) والرتبة (Rank)
69	12.4 القيم والمتجهات الذاتية (Eigenvalues & Eigenvectors)
69	12.5 تطبيقات الجبر الخطي في الرسوميات الحاسوبية والذكاء الاصطناعي
71	الفصل الثالث عشر: حساب التفاضل والتكامل لعلوم الحاسوب
71	13.1 الدوال الحقيقية ورسم المنحنيات (Real Functions)
71	13.2 النهايات والاستمرارية (Limits and Continuity)
74	الفصل الرابع عشر: الاحتمالات والإحصاء وتحليل البيانات
74	14.1 مفاهيم أساسية في الاحتمال (Probability Concepts)
75	14.2 المتغيرات العشوائية والتوزيعات الشهيرة
75	14.3 المقاييس الإحصائية (Statistical Measures)
76	14.4 الاستدلال الإحصائي (Statistical Inference)
77	14.5 تطبيقات الاحتمالات والإحصاء في أمن المعلومات والذكاء الاصطناعي
78	الفصل الخامس عشر: الرياضيات التطبيقية في الذكاء الاصطناعي وعلوم البيانات
78	15.1 نظرة عامة على نماذج التعلم الآلي (Machine Learning Models)

79	15.3 دور الاحتمالات والإحصاء في نماذج التنبؤ
79	15.4 أمثلة تطبيقية مبسطة لربط الرياضيات ببرامج عملية
81	الفصل السادس عشر: مدخل إلى علوم البيانات ودورة حياة مشروع البيانات
81	16.1 تعريف علوم البيانات وتطورها
82	16.3 أنواع البيانات ومصادرها
83	16.4 دورة حياة مشروع علوم البيانات (Data Science Lifecycle)
83	16.5 تنظيف البيانات ومعالجتها (Data Cleaning)
85	الفصل السابع عشر: أساسيات تعلم الآلة (Machine Learning)
85	17.1 مقدمة في تعلم الآلة وأنواعه
86	17.2 بيانات التدريب والتحقق والاختبار (Train/Validation/Test Split)
87	17.4 نماذج التعلم غير الخاضع للإشراف
88	17.5 مقاييس تقييم النماذج (Evaluation Metrics)
88	17.6 ضبط المعاملات وتقنيات Cross-Validation
92	الفصل الثامن عشر: الشبكات العصبية العميقة والتعلم العميق (Deep Learning)
92	18.1 البنية العامة للشبكات العصبية الاصطناعية (ANN)
92	18.2 خوارزمية الانتشار العكسي (Backpropagation) والهبوط التدريجي
93	18.3 الشبكات الالتفافية (CNN) للرؤية الحاسوبية
93	18.4 الشبكات المتتابعة (RNN, LSTM) للسلاسل الزمنية والنصوص
94	18.5 لمحة عن Transformers والنماذج اللغوية الكبيرة (LLMs)
94	18.6 متطلبات التدريب GPU/TPU
95	18.7 تحديات عملية Overfitting ، Dropout ، Regularization
96	الفصل التاسع عشر: هندسة البيانات وبنى الذكاء الاصطناعي في البيئات الواقعية
96	19.1 قواعد البيانات العلائقية SQL وغير العلائقية NoSQL

96	19.2	مستودعات البيانات (Data Warehouse) وبحيرات البيانات (Data Lake)
97	19.3	أنابيب البيانات (ETL/ELT)
97	19.4	المعالجة الموزعة (Hadoop, Spark)
98	19.5	أساسيات MLOps من النموذج التجريبي إلى الإنتاج
98	19.6	نشر نماذج التعلم الآلي كخدمات
98	19.7	مراقبة النماذج (Model Monitoring)
99		الفصل العشرون: أخلاقيات الذكاء الاصطناعي ومستقبل الحوسبة والبحث العلمي
99	20.1	مبادئ أخلاقيات الذكاء الاصطناعي
99	20.2	التحيز الخوارزمي (Algorithmic Bias)
100	20.3	الخصوصية وحماية البيانات
101	20.4	الذكاء الاصطناعي وسوق العمل
101	20.5	الاستدامة والآثار البيئية
102	20.6	الأطر التشريعية والتنظيمية العالمية
103	20.7	آفاق البحث المستقبلية بعد 2026

مقدمة الكتاب

في ظل التطور المتسارع في مجالات المعرفة والتقنية، أصبحت الحاجة إلى مصادر علمية منظمة وموثوقة أمرًا بالغ الأهمية، سواء للطلبة أو للباحثين أو للمهتمين بتنمية مهاراتهم الفكرية والعملية. ومن هذا المنطلق، يأتي هذا الكتاب ليقدم محتواه بأسلوب علمي واضح، يجمع بين الدقة المعرفية وسهولة العرض، بما يراعي الفروق الفردية في الفهم ويعزز القدرة على الاستيعاب والتطبيق.

يهدف هذا الكتاب إلى تسليط الضوء على المفاهيم الأساسية والمتقدمة في مجاله، مع تقديم شروح مبسطة مدعومة بالأمثلة والتوضيحات التي تساعد القارئ على الربط بين الجانب النظري والتطبيقي. وقد تم تنظيم فصول الكتاب بطريقة منهجية متسلسلة، تبدأ بالمبادئ التمهيدية وتنتقل تدريجيًا إلى الموضوعات الأكثر عمقًا، بما يضمن بناء معرفة متكاملة ومتينة.

حرصنا في إعداد هذا العمل على اعتماد لغة علمية رصينة، والالتزام بالموضوعية والدقة، مع مراعاة أن يكون المحتوى مواكبًا للتطورات الحديثة، وقابلًا للاستخدام كمرجع أكاديمي أو دليل تدريبي عملي. ونأمل أن يحقق هذا الكتاب الفائدة المرجوة، وأن يسهم في إثراء المكتبة العلمية ودعم مسيرة التعلم والبحث لدى القارئ.

وفي الختام، فإن هذا الجهد المتواضع هو محاولة للإسهام في نشر المعرفة، وبيقى الكمال هدفًا نسعى إليه، آملين أن يجد القارئ في صفحات هذا الكتاب ما يلبي احتياجاته العلمية ويثري خبرته المعرفية.

الفصل الأول

الإطار المفاهيمي لعلم الحاسوب وتطور الآلة

1.1 مقدمة نظرية: فلسفة الحوسبة ونظرية المعلومات

علم الحاسوب (Computer Science) ليس مجرد دراسة للألات أو البرمجيات، بل هو علم دراسة "التعقيد (Complexity)" وكيفية ترويضه. إنه العلم الذي يقع عند تقاطع الرياضيات (المنطق)، والهندسة (التصميم)، وعلم النفس (الإدراك)، والفيزياء (الإلكترونيات). في هذا القسم، سنؤسس الفهم الفلسفي والرياضي الذي يسبق وجود الآلة نفسها.

1.1.1 التعريف الأكاديمي: أطروحة تشيرش-تورينج (Church-Turing Thesis)

في ثلاثينيات القرن العشرين، وقبل بناء أي حاسوب إلكتروني، وضع عالما الرياضيات ألان تورينج (Alan Turing) و ألونزو تشيرش (Alonzo Church) الإطار النظري لما يمكن "حسابه".

- مفهوم القابلية للحساب (Computability): تنص الأطروحة على أن أي دالة رياضية أو منطقية يمكن حلها بواسطة خوارزمية (سلسلة خطوات محددة)، يمكن تنفيذها بواسطة آلة نظرية بسيطة تسمى "آلة تورينج".
- آلة تورينج (The Turing Machine): تتكون نظرياً من:

1. شريط لا نهائي مقسم إلى خلايا. (Memory Tape)
2. رأس قراءة وكتابة (Read/Write Head) يتحرك يميناً ويساراً.
3. سجل حالة (State Register) يخزن الوضع الحالي للآلة.
4. جدول تعليمات (Transition Table) يحدد ما تفعله الآلة بناءً على الرمز المقروء والحالة الحالية.

- **الاستنتاج الفلسفي العميق:** لا يوجد حاسوب في الكون (سواء كان دماغاً بشرياً، أو حاسوباً كمياً، أو حاسوباً فانقاً من عام 2050) أقوى نظرياً من آلة تورينج. الفروق تكمن فقط في **السرعة و الكفاءة**، وليس في "القدرة" على الحل. هذا يعني أن حدود الحوسبة هي حدود رياضية مطلقة.

1.1.2 نظرية المعلومات (Information Theory) والإنتروبيا

في عام 1948، نشر **كلود شانون (Claude Shannon)** ورقته البحثية الثورية "نظرية رياضية للاتصال".

- **المعلومة كـ "تقليل للشك":** عرّف شانون المعلومة بأنها الشيء الذي يزيل عدم اليقين. إذا أخبرتك أن "الشمس ستشرق غداً"، فهذه ليست معلومة (لأن الاحتمال 100%). لكن إذا أخبرتك بنتيجة رمية نرد، فهذه معلومة (لأنها أزلت احتمالاً من 6 احتمالات).
- **البت (Bit) كوحدة قياس:** قدم شانون مصطلح "Binary Digit" لقياس المعلومات. البت الواحد هو الجواب على سؤال نعم/لا باحتمال متساوي (50/50).
- **الإنتروبيا (Entropy):** مقياس لعشوائية البيانات. الملفات المضغوطة (Compressed Files) هي ملفات ذات إنتروبيا عالية (معلومات كثيفة)، بينما النصوص العادية ذات إنتروبيا منخفضة (تكرار كثير).

1.1.3 خصائص النظام الحاسوبي الحديث

يتسم الحاسوب الرقمي الحديث بأربع ركائز تميزه عن الآلات الحاسبة الميكانيكية:

1. **البرمجة (Programmability):** القدرة على تغيير وظيفة الآلة جذرياً بتغيير التعليمات البرمجية المخزنة في الذاكرة، دون الحاجة لتعديل الأسلاك أو التروس (مبدأ الآلة ذات الغرض العام General Purpose Machine).
2. **السرعة الهائلة (High Speed):** القدرة على إجراء مليارات العمليات الحسابية في الثانية (GigaFLOPS)، مما يسمح بحل المعادلات التفاضلية المعقدة، ومحاكاة الطقس، وتشفير البيانات في الزمن الحقيقي.
3. **الدقة المتناهية (Accuracy):** الحواسيب الرقمية لا تعاني من "الضوضاء (Noise)" التي تعاني منها الأنظمة التناظرية. الخطأ في الحاسوب هو إما خطأ في التصميم (Hardware Bug) أو في البرمجة (Software Bug)، أو في البيانات المدخلة (GIGO).

4. التخزين الهائل (Mass Storage): القدرة على أرشفة تاريخ البشرية الرقمي واسترجاعه دون فقدان بت واحد.

1.1.4 دورة معالجة المعلومات (IPOS Cycle)

تعمل جميع الأنظمة الرقمية وفق دورة حياة صارمة للبيانات:

المرحلة	الوصف التقني العميق	المكونات الفيزيائية	أمثلة تطبيقية
الإدخال (Input)	عملية التحويل (Transduction) والرقمنة (Digitization). تحويل الكميات الفيزيائية المستمرة (ضغط صوت، طيف ضوئي) إلى تيار من البتات المنفصلة (Discrete Bits) باستخدام محولات ADC.	لوحة المفاتيح، الميكروفون، الماسح الضوئي، مستشعرات IoT.	تحويل صوتك إلى ملف WAV.
المعالجة (Processing)	التلاعب بالرموز وفق قواعد المنطق البولياني والحساب الثنائي. تشمل العمليات الحسابية، المقارنة المنطقية، ونقل البيانات.	وحدة المعالجة المركزية (CPU)، وحدة معالجة الرسومات (GPU).	تطبيق فلتر على صورة، ضغط ملف، حساب مسار صاروخ.
التخزين (Storage)	الحفاظ على حالة البتات (State Preservation) عبر الزمن. ينقسم إلى تخزين متطاير (Volatile) وسريع، وتخزين دائم (Non-Volatile) وبطيء.	RAM, Cache, SSD, HDD, Magnetic Tapes.	حفظ مستند وورد، تخزين متغيرات اللعبة في الرام.

المرحلة	الوصف التقني العميق	المكونات الفيزيائية	أمثلة تطبيقية
Output (الإخراج)	عملية التصيير (Rendering) وإعادة التشكيل (Reconstruction). تحويل البيانات الرقمية إلى إشارات تناظرية يدركها الإنسان باستخدام محولات DAC.	الشاشات (Monitors)، الطابعات، مكبرات الصوت، المحركات (Motors).	عرض فيديو K4، طباعة تقرير ثلاثي الأبعاد.

1.2 الجذور التاريخية للحوسبة: رحلة 5000 عام لأتمتة العقل

لفهم الحاضر، يجب الغوص في ماضي محاولات الإنسان لتقليل العبء الذهني للحساب.

1.2.1 العصور القديمة: الحساب الميكانيكي والتمثيل الموضعي

- **المعداد 2400 - (Abacus) ق.م:** اخترع في بابل وتطور في الصين. الأهمية العلمية للمعداد تكمن في فصله بين "العدد" و "الخانة". خرزة في العمود الأول قيمتها 1، ونفس الخرزة في العمود الثاني قيمتها 10. هذا هو التمثيل الموضعي (Positional Notation) الذي تعتمد عليه حواسيبنا اليوم.
- **آلية أنتيكثيرا 100 - (Antikythera Mechanism) ق.م:** اكتشفت في حطام سفينة يونانية. تعتبر "أول حاسوب تناظري". استخدمت نظاماً معقداً من التروس التفاضلية (Differential Gears) - التي أعيد اكتشافها في القرن العشرين- لحساب مواقع الشمس والقمر والكسوف لسنوات قادمة بدقة مذهلة.

1.2.2 العصر الذهبي الإسلامي: تأسيس اللوغاريتمات والأتمتة

- **الخوارزمي (780-850م):** في كتابه "المختصر في حساب الجبر والمقابلة"، وضع منهجية لحل المعادلات خطوة بخطوة، وهو ما عُرف لاحقاً بـ "الخوارزمية (Algorithm)". كما ساهم في نقل الأرقام الهندية (بما فيها الصفر) إلى الغرب، مما مهد الطريق للحساب الرقمي.

- الكندي (801-873م): مؤسس علم تحليل الشفرات. (Cryptanalysis) استخدم التحليل الترددي (Frequency Analysis) لكسر الشفرات، وهو أساس علم أمن المعلومات الحديث.
- الجزري (1136-1206م): مهندس عبقري صمم آلات ذاتية الحركة. (Automata) ساعته المائية الشهيرة "ساعة الفيل" وآلاته الموسيقية المبرمجة باستخدام أوتاد على أسطوانة دوارة، تعتبر أولى محاولات "البرمجة الميكانيكية".

1.2.3 عصر النهضة الميكانيكية (القرن 17-19)

- جون نابير: 1614 - (John Napier) اخترع اللوغاريتمات وعيدان نابير (Napier's Bones)، التي حولت عمليات الضرب المعقدة إلى عمليات جمع بسيطة.
- بليز باسكال: 1642 - (Pascaline) أول آلة حاسبة ميكانيكية عملية. قدمت حلاً هندسياً لمشكلة "نقل الفائض (Carry)" باستخدام الجاذبية والتروس.
- جوتفريد لايبنتز: 1673 - (Leibniz) صمم "العجلة المتدرجة (Stepped Reckoner)" القادرة على الضرب والقسمة. الأهم من آله هو فلسفته؛ فقد كان مهووساً بـ النظام الثنائي (Binary System)، ورأى فيه لغة المنطق النقي (1 للوجود، 0 للعدم).
- جوزيف ماري جاكارد: 1804 - (Jacquard Loom) استخدم البطاقات المثقبة (Punched Cards) للتحكم في نمط نسيج القماش. الثقوب كانت تعليمات (1/0) للإبر. كانت هذه اللحظة التي انتقلت فيها الصناعة بالبرمجة.

1.2.4 الحلم الفيكتوري: بابيج ولوفلايس

- تشارلز بابيج: 1837 - (Charles Babbage) صمم (على الورق) المحرك التحليلي (Analytical Engine) كانت آلة بخارية بحجم قاطرة، لكنها احتوت المخطط الكامل للحاسوب الحديث:

1. الطاحونة (The Mill): وحدة المعالجة المركزية (ALU).

2. المخزن (The Store): الذاكرة (Memory).

3. القارئ والطابعة: وحدات I/O.

4. التفرع الشرطي: القدرة على اتخاذ قرارات.

- **آدا لوفلايس (Ada Lovelace)** ابنة الشاعر اللورد بايرون. ترجمت أعمال بابيج وأضافت عليها ملاحظات تفوق الأصل. كتبت أول برنامج في التاريخ (لحساب أعداد برنولي). تنبأت بأن الحاسوب سيتمكن يوماً ما من تأليف الموسيقى والرسم (الحوسبة الرمزية)، متجاوزة نظرة بابيج التي انحصرت في الأرقام.

1.3 ولادة الحاسوب الحديث: سنوات الحرب والتشفير(1936-1945)

الحاجة لفك شفرات العدو وحساب مسارات القذائف الباليستية كانت الوقود الذي سرّع التطور التكنولوجي بقرون خلال عقد واحد.

- **Zuse Z3 ألمانيا، 1941** : بناه المهندس كونراد زوس في عزلة تامة عن العالم. كان أول حاسوب كهروميكانيكي رقمي قابل للبرمجة بالكامل، واستخدم النظام الثنائي. دمرته غارة جوية للحلفاء عام 1943.

- **Colossus بريطانيا، 1943** : في "بليتشلي بارك"، صمم تومي فلاورز هذا العملاق لفك شفرة "لورنز" الألمانية المعقدة (وليس إنجما التي فكها تورينج بألة بومب). كان Colossus أول حاسوب إلكتروني بالكامل (استخدم الصمامات المفرغة بدلاً من التروس)، مما منحه سرعة هائلة. ظل وجوده سراً عسكرياً حتى السبعينيات.

- **Harvard Mark I أمريكا، 1944** : حاسوب كهروميكانيكي بطول 15 متراً تم بناؤه بتمويل من IBM. استخدمته البحرية الأمريكية لحساب الجداول. اشتهر بأن **غريس هوبر (Grace Hopper)** وجدت حشرة (عثة Moth) حقيقية عالقة في أحد مفاتيحه، ومن هنا سُنَّ مصطلح "**Debugging**" تصحيح الأخطاء.

- **ENIAC أمريكا، 1945** : الحاسوب والمكامل الرقمي الإلكتروني". يعتبر أول حاسوب إلكتروني عام الأغراض. صممه إيكيرت وموكلي. كان وزنه 30 طناً، ويحتوي على 18,000 صمام مفرغ. برمجته كانت تتطلب أياماً من إعادة توصيل الكابلات يدوياً.

1.4 أجيال الحاسوب الإلكترونية: التصنيف الشامل والمفصل

يتم تصنيف حقب الحاسوب بناءً على "المفتاح الإلكتروني (Switching Element)" المستخدم لبناء البوابات المنطقية.

جدول المقارنة الشامل للأجيال

الجيل	الفترة الزمنية	التقنية الأساسية (Hardware)	البرمجيات واللغات	خصائص الأداء والمشاكل	مثل للأجهزة
الأول	1940 - 1956	الصمامات المفرغة (Vacuum Tubes): أنابيب زجاجية بحجم المصباح تتحكم في تدفق الإلكترونات.	لغة الآلة (Machine Code): برمجة بالأرقام الثنائية المباشرة.	حرارة هائلة، استهلاك طاقة عالي، أعطال يومية (احتراق الصمامات). سرعة: ~40,000 عملية/ثانية.	ENIAC, EDVAC, UNIVAC I
الثاني	1956 - 1963	الترانزستور (Transistor): اختراع مختبرات بيل. قطعة صلبة من الجرمانيوم أو السيليكون.	لغة التجميع (Assembly): ولغات عالية المستوى الأولى (FORTRAN, COBOL).	حجم أصغر بـ 100 مرة، موثوقية أعلى، لا تحتاج تسخين. سرعة: ~200,000 عملية/ثانية.	IBM 7094, PDP-1
الثالث	1964 - 1971	الدوائر المتكاملة (IC): دمج مئات الترانزستورات والمقاومات على شريحة سيليكون واحدة.	أنظمة التشغيل (OS)، البرمجة المهيكلية، المشاركة الزمنية	ظهور "عائلة الحواسيب" المتوافقة. انخفاض التكلفة وبداية الانتشار	IBM System/360, DEC PDP-8

الجيل	الفترة الزمنية	التقنية الأساسية (Hardware)	البرمجيات واللغات	خصائص الأداء والمشاكل	مثال للأجهزة
			Time-) (sharing).	التجاري. سرعة: ~1 MIPS.	
الرابع	1971-الآن	المعالج الدقيق (Microprocessor): تقنية VLSI و ULSI. ملايين/مليارات الترانزستورات في شريحة واحدة.	البرمجة كائنية التوجه (OOP)، واجهات رسومية (GUI)، الإنترنت، الذكاء الاصطناعي.	الحوسبة الشخصية، الهواتف الذكية، الحوسبة السحابية. سرعة: Giga/TeraFLOP S.	Intel 4004, Apple M-Series, AMD Ryzen
الخامس	المستقبل	ما بعد السيليكون: الحوسبة الكمية (Quantum)، العصبية (Neuromorphic)، الضوئية (Optical).	معالجة اللغة الطبيعية (NLP)، التعلم العميق الذاتي.	كسر حاجز قانون مور. حل مشاكل مستعصية (التشفير، طبي البروتين) في ثوانٍ.	Google Sycamore, IBM Q System One

1.5 التصنيف المنهجي للحواسيب وأنواعها

1.5.1 حسب مبدأ معالجة البيانات

1. الحاسوب التناظري: (Analog Computer)

- يمثل البيانات بجهد كهربائي مستمر. لا يوجد 0 و 1، بل قيم متصلة. (1.5V, 1.51V...)
- الاستخدام: حل المعادلات التفاضلية في الزمن الحقيقي (مثل أنظمة توجيه الصواريخ القديمة، ومحاكاة التعليق في السيارات).
- الميزة: سرعة فورية. (Instantaneous) العيب: الدقة محدودة بضوضاء المكونات، وصعوبة إعادة البرمجة.

2. الحاسوب الرقمي: (Digital Computer)

- يمثل البيانات بشكل متقطع (Discrete) باستخدام النظام الثنائي.
- *الميزة*: دقة مطلقة (يمكن زيادتها بزيادة عدد البتات)، مرونة كاملة في البرمجة.

3. الحاسوب الهجين: (Hybrid Computer)

- يجمع السرعة التناظرية والدقة الرقمية. يُستخدم في وحدات العناية المركزة (لقياس إشارات القلب التناظرية وتحويلها رقمياً للتحليل).

1.5.2 حسب الحجم والقدرة (Taxonomy of Computer Systems)

1. الحواسيب العملاقة: (Supercomputers)

- أجهزة فريدة ومكلفة جداً. تتكون من عشرات الآلاف من المعالجات (CPUs & GPUs) المتصلة بشبكة فائقة السرعة.
- *الغرض*: المهام التي تتطلب حسابات مكثفة (CPU-Bound) مثل نمذجة المناخ، محاكاة الأسلحة النووية، وتدريب نماذج الذكاء الاصطناعي الضخمة.
- *نظام التشغيل*: نسخ مخصصة من Linux.

2. الحواسيب المركزية: (Mainframes)

- وحوش المعالجة التجارية. تركز على عمليات الإدخال والإخراج (I/O Bound).
- *الغرض*: معالجة المعاملات (Transaction Processing) للبنوك وشركات التأمين. قادرة على التعامل مع ملايين الطلبات المترابطة.
- *الميزة*: الموثوقية الخرافية (99.99999%)، وقابلية صيانة الأجزاء دون إيقاف التشغيل.

3. الحواسيب الصغيرة - (Minicomputers) تاريخي:

- ظهرت في الستينيات كبديل أرخص للمركزية. تطورت لتصبح الخوادم (Servers) الحديثة.

4. الحواسيب الدقيقة: (Microcomputers)

- كل جهاز يعتمد على معالج دقيق واحد. تشمل: الحواسيب المكتبية (Desktop) ، المحمولة (Laptop)، اللوحية (Tablet) ، والهواتف الذكية.

5. الأنظمة المدمجة: (Embedded Systems)

- حواسيب "غير مرئية" مدمجة في أجهزة أخرى. لا يراها المستخدم كحاسوب.
- أمثلة: نظام ABS في السيارة، وحدة التحكم في الميكروويف، أجهزة إنترنت الأشياء. (IoT)
- الخصائص: موارد محدودة (ذاكرة وكهرباء)، وتعمل غالباً في الزمن الحقيقي (Real-time Constraints).

1.6 المعماريات الحاسوبية (Computer Architectures)

هندسة الحاسوب (Computer Architecture) هي العلم الذي يربط بين البرمجيات والعتاد.

1.6.1 معمارية فون نيومان (Von Neumann Architecture)

نموذج اقترحه جون فون نيومان عام 1945، وهو التصميم السائد في 99% من الأجهزة اليوم.

- المفهوم الجوهري: البرنامج المخزن. (Stored-Program) "الذاكرة الرئيسية تحتوي على خليط من التعليمات (Code) و البيانات (Data)."
- الهيكلية:

1. وحدة المعالجة المركزية (CPU) تحتوي وحدة التحكم (CU) ووحدة الحساب (ALU).

2. ذاكرة واحدة مشتركة.

3. نظام نواقل (Buses) مشترك لنقل العناوين والبيانات.

- عنق زجاجة فون نيومان (Von Neumann Bottleneck): بما أن هناك مساراً واحداً فقط للذاكرة، فإن المعالج لا يستطيع جلب تعليمة وجلب بيانات في نفس الوقت. وسرعة المعالجات تضاعفت آلاف المرات بينما سرعة الذاكرة تطورت ببطء، مما جعل المعالج يقضي معظم وقته "منتظراً" وصول البيانات.

1.6.2 معمارية هارفارد (Harvard Architecture)

- **المفهوم:** فصل فيزيائي تام. ذاكرة خاصة للتعليمات، وذاكرة خاصة للبيانات.
- **الميزة:** سرعة مضاعفة في الوصول للذاكرة (نظرياً). يمكن جلب تعليمة وكتابة نتيجة في نفس النبضة.
- **التطبيق:** المتحكمات الدقيقة (Microcontrollers) ومعالجات الإشارات الرقمية (DSP).
- **الحل الهجين (Modified Harvard):** الحواسيب الحديثة (x86) تستخدم معمارية هجينة؛ فهي تبدو كفون نيومان من الخارج (RAM واحدة)، لكنها من الداخل) في الـ (CPU تمتلك ذاكرة كاش منفصلة (L1 Instruction Cache & L1 Data Cache) لتحقيق سرعة هارفارد.

1.6.3 تصميم مجموعة التعليمات (ISA: CISC vs RISC)

هذا هو "عقد التفاهم" بين البرمجيات والعتاد.

1. CISC (Complex Instruction Set Computer):

- الفلسفة: اجعل التعليمة الواحدة قوية ومعقدة (تقوم بمهام متعددة مثل الجلب والضرب والحفظ في خطوة واحدة).
- الهدف: تقليل حجم الكود البرمجي (لأن الذاكرة كانت غالية).
- المثال: معالجات Intel x86 و AMD64 المسيطرة على الحواسيب الشخصية والخوادم.

2. RISC (Reduced Instruction Set Computer):

- الفلسفة: اجعل التعليمات بسيطة جداً وسريعة وموحدة الطول. كل تعليمة تقوم بعمل واحد بسيط وتنفذ في دورة ساعة واحدة.
- الهدف: تبسيط تصميم المعالج وزيادة كفاءة الطاقة.
- المثال: معالجات ARM المسيطرة على الهواتف المحمولة وأجهزة Apple Silicon M-series

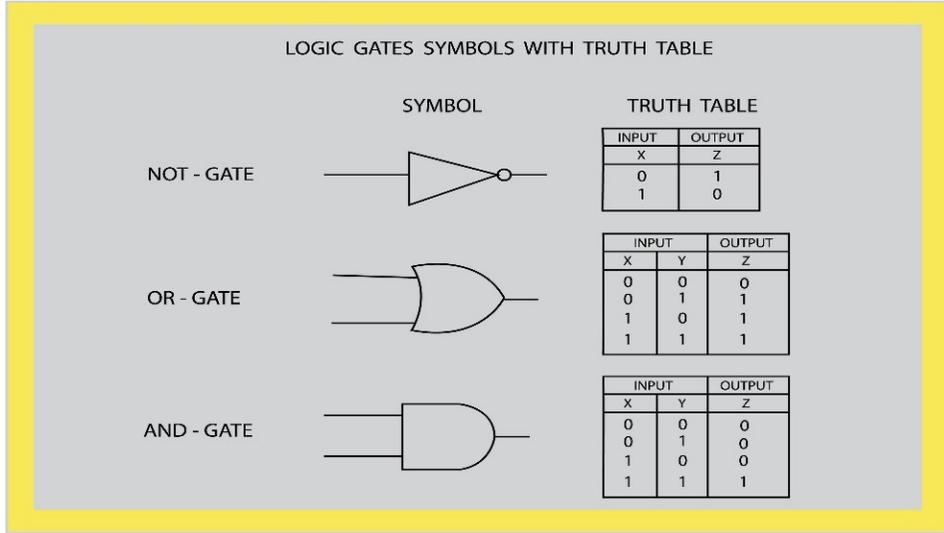
1.7 الأساس المنطقي: البوابات الرقمية والجبر البوليني

في المستوى الأدنى، الحاسوب هو شبكة ضخمة من المفاتيح المنطقية. الرياضيات التي تحكم هذه المفاتيح هي الجبر البوليني (Boolean Algebra) الذي طوره جورج بول في القرن التاسع عشر.

البوابات المنطقية الأساسية (Logic Gates)

البوابة هي دائرة إلكترونية (مبنية من ترانزستورات) لها مدخل أو أكثر ومخرج واحد، وتعمل وفق منطق محدد.

اسم البوابة	الرمز الجبري	الوصف الوظيفي	جدول الحقيقة	(1,1)	(1,0)	(0,1)
AND	$A \cdot B$	الضرب المنطقي. تعطي 1 فقط إذا كان كل المدخلات 1.	0	1	0	0
OR	$A + B$	الجمع المنطقي. تعطي 1 إذا كان أي من المدخلات 1.	0	1	1	1
NOT	\bar{A}	العكس. تقلب القيمة (0 يصبح 1 والعكس).	1	0	-	-
XOR	$A \oplus B$	الجمع الحصري. تعطي 1 إذا كانت المدخلات مختلفة. وهي أساس عملية الجمع الحسابي في المعالج.	0	0	1	1
NAND	$\overline{A \cdot B}$	نفي الـ AND. تسمى البوابة الشاملة لأنه يمكن بناء أي بوابة أخرى باستخدام NAND فقط.	1	0	1	1



1.7 الأساس المنطقي: البوابات الرقمية والجبر البوليني

في المستوى الأدنى، الحاسوب هو شبكة ضخمة من المفاتيح المنطقية. الرياضيات التي تحكم هذه المفاتيح هي الجبر البوليني (**Boolean Algebra**) الذي طوره جورج بول في القرن التاسع عشر.

البوابات المنطقية الأساسية (Logic Gates)

البوابة هي دائرة إلكترونية (مبنية من ترانزستورات) لها مدخل أو أكثر ومخرج واحد، وتعمل وفق منطق محدد.

(1,1)	(1,0)	(0,1)	جدول الحقيقة (0,0)	الوصف الوظيفي	الرمز الجبري	اسم البوابة
1	0	0	0	الضرب المنطقي. تعطي 1 فقط إذا كان كل المدخلات 1.	$A \cdot B$	AND

0	1	1	1	نفي الـ AND. تسمى البوابة الشاملة لأنه يمكن بناء أي بوابة أخرى باستخدام NAND فقط.	$\overline{A \cdot B}$	NAND
0	-	-	1	العاكس. تقلب القيمة (0 يصبح 1 والعكس).	\bar{A}	NOT
1	1	1	0	الجمع المنطقي. تعطي 1 إذا كان أي من المدخلات 1.	$A + B$	OR
0	1	1	0	الجمع الحصري. تعطي 1 إذا كانت المدخلات مختلفة . وهي أساس عملية الجمع الحسابي في المعالج.	$A \oplus B$	XOR

1.8 أنظمة العد وتمثيل البيانات (Data Representation)

الحاسوب "أعمى" عن المعاني البشرية. هو يرى فقط حالات الجهد الكهربائي (High/Low) التي نترجمها نحن إلى (1/0).

1.8.1 الأنظمة العددية

- النظام العشري: (Decimal) أساس 10. (رموز: 0-9). هو نظامنا الطبيعي.
- النظام الثنائي: (Binary) أساس 2. (رموز: 0، 1). هو لغة الهاردوير لسهولة تمثيله بالترانزستور (On/Off).
- النظام الست عشري: (Hexadecimal) أساس (رموز: 0-A-F) يستخدم المبرمجون كاختصار لقراءة البيانات الثنائية. (كل 4 بتات = 1 خانة هيكس).

◦ مثال: اللون الأزرق الصافي 0000FF أسهل للقراءة من

000000000000000011111111

1.8.2 تمثيل الأعداد الصحيحة (Integers)

- الأعداد الموجبة: (Unsigned) تحويل مباشر من ثنائي لعشري.
- الأعداد السالبة: (Signed) نستخدم طريقة "المكمل الثنائي" (Two's Complement) الفكرة هي جعل الخانة الأخيرة (MSB) للإشارة، وقلب البتات وإضافة 1. هذه الطريقة العبقورية تجعل عملية الطرح هي نفسها عملية الجمع، مما يبسط تصميم المعالج.

1.8.3 الأعداد العائمة (Floating Point - IEEE 754)

تمثيل الأعداد الكسرية والضخمة جداً (مثل ثابت بلانك أو المسافة للمجرات) يتطلب صيغة علمية :

$$Value = (-1)^{Sign} \times (1.Mantissa) \times 2^{(Exponent-Bias)}$$

تقسم الـ 32 بت (Single Precision) إلى:

1. Sign (1 bit) : إشارة الرقم.
 2. Exponent (8 bits) : الأس (يحدد مكان الفاصلة العائمة).
 3. Mantissa (23 bits) : الكسور (تحدد دقة الرقم).
- مشكلة الدقة: الحاسوب لا يستطيع تمثيل \$0.1\$ بدقة متناهية بالنظام الثنائي، مما يؤدي لأخطاء تقريبية تتراكم في الحسابات العلمية.

1.8.4 تمثيل النصوص (Text Encoding)

- ASCII (1963) معيار 7-بت. يمثل الحروف الإنجليزية والأرقام والرموز الأساسية (128 رمزاً). كان كافياً لأمريكا، لكنه فشل عالمياً.
- Unicode (1991) المعيار العالمي الموحد. يعطي رقماً فريداً (Code Point) لكل حرف في كل لغة بشرية (عربي، صيني، هيروغليفي) وحتى الرموز التعبيرية (Emojis).

- **UTF-8**: طريقة تشفير الـ Unicode الأكثر شيوعاً. تستخدم طولاً متغيراً (1 بايت للإنجليزية، 2 العربية، 3 للصينية..). هذا الذكاء في التصميم جعلها العمود الفقري للإنترنت.

1.9 مقاييس الأداء وقوانين الحوسبة (Performance Laws)

كيف نقيس سرعة الحاسوب؟ وهل زيادة المعالجات تعني زيادة السرعة دائماً؟

1.9.1 معادلة أداء المعالج (The Iron Law of Performance)

$$\text{Time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

لتحسين الأداء، أمام المهندسين ثلاث طرق:

1. تقليل عدد التعليمات ← (Instruction Count) عبر مترجمات (Compilers) أذكى أو خوارزميات أفضل.
2. تقليل عدد الدورات لكل تعليمة ← (CPI) عبر تحسين معمارية المعالج (Pipelining, Superscalar).
3. زيادة التردد ← (Clock Rate) عبر تحسين تقنية تصنيع الترانزستور (تكنولوجيا النانو).

1.9.2 قانون أمداال - (Amdahl's Law) قانون التشاؤم

يحكم حدود السرعة القصوى عند استخدام المعالجة المتوازية. ينص على أن تسريع النظام محدود بالجزء الذي لا يمكن موازاته (الجزء المتسلسل).

- مثال: إذا كان 10% من البرنامج يجب أن ينفذ بالتسلسل، فلا يمكن تسريع البرنامج أكثر من 10 أضعاف، حتى لو استخدمت مليون معالج!

1.9.3 قانون جوستافسون - (Gustafson's Law) قانون التفاؤل

يرد على أمداال قائلاً: "عندما نمتلك قوة حاسوبية أكبر، نحن لا نحل نفس المشكلة الصغيرة بسرعة أكبر، بل نحل مشكلة أكبر وأقعد في نفس الوقت". هذا القانون هو الذي يبرر الجدوى الاقتصادية لمراكز البيانات العملاقة في عصر الذكاء الاصطناعي.

الفصل الثاني

البنية المادية – (Hardware) التشريح التفصيلي للحاسوب

2.1 مقدمة في العتاد وتكامل المنظومة الحاسوبية

البنية المادية (Hardware) هي الجسد الفيزيائي للحاسوب، وهي المكونات الملموسة التي تنفذ التعليمات البرمجية. من منظور هندسي، لا تعمل هذه المكونات كجزر منعزلة، بل كنظام متكامل (Integrated System) يعتمد على التوافقية (Compatibility) وتزامن نقل البيانات. (Synchronization) الفهم العميق للعتاد ليس ضرورياً فقط لمهندسي الحاسوب، بل هو حاسم لعلماء البيانات ومطوري الذكاء الاصطناعي لفهم قيود الأداء وكيفية تحسين الخوارزميات (Optimization) لتناسب مع قدرات الآلة.

2.2 اللوحة الأم – (Motherboard) العمود الفقري للنظام

اللوحة الأم هي لوحة الدوائر المطبوعة الرئيسية (Main Printed Circuit Board - PCB) التي تحتضن وتوصل جميع مكونات الحاسوب. دورها يتعدى كونها "حاملة" للمكونات؛ فهي تدير توزيع الطاقة (Power Distribution) وتنظم حركة المرور للبيانات بين المعالج والذاكرة والملحقات.

2.2.1 المكونات الرئيسية

1. **المقبس (CPU Socket):** هو الحاضنة الميكانيكية والكهربائية للمعالج. يختلف تصميمه باختلاف الشركة المصنعة) مثل LGA من Intel حيث الدبابيس في اللوحة، و PGA من AMD حيث الدبابيس في المعالج قديماً، و LGA في الأجيال الحديثة مثل (AM5).
2. **مجموعة الشرائح (Chipset):** هي "شرطي المرور" في اللوحة الأم. تتكون قديماً من الجسر الشمالي (Northbridge) المتصل بالمعالج والذاكرة، والجسر الجنوبي (Southbridge) المتصل بالمنافذ البطيئة. في المعماريات الحديثة، تم دمج وظائف الجسر الشمالي داخل المعالج نفسه لتقليل الكمون (Latency)، وبقيت الشريحة (PCH) في Intel لإدارة المنافذ والاتصالات الطرفية.

3. شقوق التوسعة: (Expansion Slots - PCIe) تعتمد معيار "ربط المكونات الطرفية السريع" (PCI Express) تأتي بسرعات وأحجام مختلفة. (x1, x4, x8, x16) تُستخدم لبطاقات الرسومات، بطاقات الشبكة، ووحدات تخزين NVMe.

4. نظام توصيل الطاقة: (VRM - Voltage Regulator Module) مكون حاسم لضمان استقرار النظام، يقوم بتحويل 12V القادمة من مزود الطاقة إلى فولتية منخفضة ودقيقة جداً مثل 1.2V يحتاجها المعالج للعمل باستقرار تحت الضغط العالي.

2.2.2 عوامل الشكل (Form Factors)

تحدد المعايير القياسية لأبعاد اللوحة وأماكن التثبيت:

- **ATX (Advanced Technology eXtended)**: القياس القياسي (305×244 مم)، يوفر أكبر عدد من شقوق التوسعة ومنافذ الذاكرة. مثالي لمحطات العمل والخوادم المصغرة.
- **Micro-ATX (mATX)**: أصغر حجماً (244×244 مم)، حل وسط يوفر قابلية توسعة جيدة مع حجم مدمج.
- **Mini-ITX**: صغير جداً (170×170 مم)، مخصص للأجهزة المدمجة (SFF)، وعادة ما يحتوي على شق توسعة واحد فقط.

2.3 وحدة المعالجة المركزية – العقل المدبر (CPU)

المعالج الدقيق هو الدائرة الإلكترونية الأكثر تعقيداً التي صنعها الإنسان، ووظيفته الأساسية هي جلب التعليمات وتنفيذها.

2.3.1 دورة التعليمات (Machine Cycle)

تعمل وحدة المعالجة وفق دورة متكررة تعرف بـ "دورة الجلب والتنفيذ: (Fetch-Execute Cycle)"

1. **الجلب (Fetch)**: نقل التعليمات التالية من الذاكرة (RAM) إلى مسجل التعليمات (Instruction Register) في المعالج.
2. **فك التشفير (Decode)**: تقوم وحدة التحكم (CU) بترجمة التعليمات الثنائية إلى إشارات تحكم تفهمها دوائر المعالج الداخلية.

3. **التنفيذ (Execute):** تقوم وحدة الحساب والمنطق (ALU) بإجراء العملية المطلوبة (جمع، مقارنة، نقل بيانات).

4. **التخزين (Store/Write Back):** كتابة النتيجة في الذاكرة أو في سجل داخلي.

2.3.2 المكونات الداخلية

- **وحدة الحساب والمنطق (ALU - Arithmetic Logic Unit):** الآلة الحاسبة" داخل المعالج، مسؤولة عن العمليات الرياضية (Integer Arithmetic) والعمليات المنطقية (AND, OR, NOT).
- **وحدة التحكم (CU - Control Unit):** المايسترو الذي يوجه تدفق البيانات بين المعالج والأجهزة الأخرى.
- **السجلات (Registers):** وحدات ذاكرة صغيرة جداً ولكنها فائقة السرعة، تقع داخل قلب المعالج وتستخدم لتخزين البيانات التي يتم العمل عليها فوراً (مثل Accumulator, Program Counter).

2.3.3 مفاهيم تقنية متقدمة

- **تعدد الأنوية (Multi-core):** دمج عدة وحدات معالجة مستقلة (Cores) في شريحة واحدة، مما يسمح بتنفيذ عدة مهام فيزيائياً في وقت واحد.
- **تعدد الخيوط (Hyper-Threading / SMT):** تقنية تسمح للنواة الواحدة بمعالجة خيطين (Threads) من التعليمات في وقت واحد، مما يحسن استغلال موارد المعالج.
- **الذاكرة المخبئية (Cache Memory - L1, L2, L3):** ذاكرة SRAM فائقة السرعة تعمل كمناطق عازلة بين المعالج السريع والذاكرة الرئيسية الأبطأ، لتقليل وقت انتظار البيانات.

2.4 الذاكرة الرئيسية (RAM)

ذاكرة الوصول العشوائي (Random Access Memory) هي مساحة العمل المؤقتة للنظام. سميت "عشوائية" لأن الوصول لأي خلية ذاكرة يستغرق نفس الوقت بغض النظر عن موقعها (خلافاً للأشرطة المغناطيسية المتسلسلة).

2.4.1 أجيال الذاكرة (DDR SDRAM)

تطورت الذاكرة لزيادة السرعة وتقليل استهلاك الطاقة:

- **DDR (Double Data Rate):** التقنية التي تسمح بنقل البيانات مرتين في كل نبضة ساعة (عند الحافة الصاعدة والهابطة للإشارة).
- **DDR4:** الجيل السائد لفترة طويلة، يعمل بفولتية 1.2V وسرعات تصل لـ 3200 MT/s.
- **DDR5:** الجيل الأحدث (عام 2026 وما قبله بقليل)، يوفر سرعات مضاعفة تبدأ من 4800 MT/s، وإدارة طاقة مدمجة (PMIC) على شريحة الذاكرة نفسها، وتصحيح أخطاء (ECC) داخلي.

2.4.2 معمارية القنوات (Channel Architecture)

- **قناة مزدوجة (Dual Channel):** تقنية تضاعف عرض نطاق نقل البيانات (Bandwidth) عبر استخدام مسارين متوازيين للوصول للذاكرة (128-bit) بدلاً من 64-bit.
- **قناة رباعية (Quad Channel):** تستخدم في محطات العمل (HEDT) والخوادم لتوفير نطاق ترددي هائل للتطبيقات العلمية.

2.4.3 الكمون والتوقيت (Latency & Timings)

سرعة الذاكرة ليست فقط بالتردد (MHz)، بل بالتوقيت (CL - CAS Latency). التوقيت هو عدد نبضات الساعة التي يجب أن ينتظرها المعالج بين طلب البيانات ووصولها. الذاكرة المثالية هي ذات تردد مرتفع وتوقيت منخفض.

2.5 وحدات التخزين الثانوية

2.5.1 الأقراص الصلبة الميكانيكية (HDD)

تعتمد على أقراص ممغنطة دوارة ورأس قراءة/كتابة ميكانيكي. تتميز بالسعة العالية والسعر المنخفض، لكنها بطيئة وعرضة للتلف الميكانيكي. تستخدم حالياً للأرشيف وتخزين البيانات الضخمة (Cold Data).

2.5.2 أقراص الحالة الصلبة (SSD)

تستخدم ذواكر فلاش (NAND Flash) ولا تحتوي أجزاء متحركة. هي أسرع بمئات المرات من HDD، وأكثر تحملاً للصدمات.

2.5.3 SATA vs NVMe بروتوكولات النقل

- **SATA** : بروتوكول قديم صمم للأقراص الميكانيكية، يحد من سرعة SSD بحد أقصى نظري 600 MB/s.
- **NVMe (Non-Volatile Memory express)** : بروتوكول صمم خصيصاً للذاكر الوميضية، يعمل عبر مسارات PCIe المباشرة للمعالج. تصل سرعات الجيل الرابع (Gen4) إلى 7500 MB/s ، والجيل الخامس (Gen5) يتجاوز 12000 MB/s ، وهو أمر حيوي لتدريب نماذج الذكاء الاصطناعي الضخمة التي تتطلب تدفق بيانات هائل.

2.6 وحدة معالجة الرسومات (GPU)

لم تعد الـ GPU مجرد وحدة لعرض الصور والألعاب. بفضل معماريتها المتوازية (التي تحتوي آلاف الأنوية الصغيرة)، أصبحت المحرك الأساسي لـ **الذكاء الاصطناعي (AI)** و **التعلم العميق (Deep Learning)**. بينما يبرع الـ CPU في المهام التسلسلية المعقدة، تبرع الـ GPU في إجراء ملايين العمليات الحسابية البسيطة (مثل ضرب المصفوفات) في وقت واحد، وهو جوهر عمل الشبكات العصبية.

2.7 مزود الطاقة (PSU) وأنظمة التبريد

2.7.1 مزود الطاقة (PSU)

هو القلب الذي يضخ الدماء (الكهرباء) للنظام. معايير **80 PLUS (Bronze, Gold, Platinum, Titanium)** تشير إلى كفاءة تحويل الطاقة من المقبس الحائطي إلى الحاسوب، مما يقلل من الفاقد الحراري وقاتورة الكهرباء.

2.7.2 التبريد (Cooling)

- **التبريد الهوائي**: استخدام مشتتات حرارية معدنية ومراوح. فعال واقتصادي.
- **التبريد المائي (Liquid Cooling)**: استخدام سائل لنقل الحرارة من المكونات إلى مشتت (Radiator) كبير. ضروري في الحواسيب عالية الأداء (HPC) ومحطات تدريب الذكاء الاصطناعي للحفاظ على درجات حرارة منخفضة تمنع الاختناق الحراري (Thermal Throttling).

2.8 وحدات الإدخال والإخراج (I/O)

هي حواس النظام التي يتفاعل بها مع العالم الخارجي. تشمل التطورات الحديثة شاشات ذات معدلات تحديث عالية (High Refresh Rate) ودقة 4K/8K لتحليل البيانات البصرية، ولوحات مفاتيح ميكانيكية لسرعة الإدخال، وطابعات ثلاثية الأبعاد (3D Printers) التي نقلت الإخراج من الورق إلى الواقع الملموس.

الفصل الثالث

المعماريات المتقدمة والحوسبة الحديثة

3.1 المعالجات متعددة الأنوية ومتعددة الخيوط (Multicore and Multithreading)

شهد العقد الأول من القرن الحادي والعشرين تحولاً جذرياً في فلسفة تصنيع المعالجات. فبعد أن كان التركيز ينصب على زيادة تردد الساعة (Clock Speed)، اصطدم المصممون بـ "جدار الطاقة (Power Wall)" والقيود الحرارية. الحل كان الانتقال من التوسع الرأسي (زيادة السرعة) إلى التوسع الأفقي (زيادة عدد الأنوية).

3.1.1 قانون أمدال (Amdahl's Law)

قبل الخوض في التقنية، يجب فهم الأساس النظري. ينص قانون أمدال على أن التحسن في أداء النظام نتيجة لتوازي العمليات محكوم بالجزء المتسلسل (Serial Part) من البرنامج الذي لا يمكن موازاته.

- **المفهوم:** إذا كان 50% من البرنامج يجب أن ينفذ بتسلسل، فإننا لن نحصل على أكثر من ضعف السرعة مهما أضفنا من أنوية.
- **الأهمية لعلوم البيانات:** هذا يفسر لماذا تحتاج بعض خوارزميات تعلم الآلة إلى إعادة هيكلة جذرية لتستفيد من المعالجات الحديثة.

3.1.2 المعالجة المتعددة المتماثلة (SMP)

في هذه المعمارية، تتشارك جميع الأنوية الذاكرة الرئيسية نفسها وتعمل تحت إشراف نظام تشغيل واحد. هذا النموذج هو السائد في الحواسيب الشخصية والخوادم التقليدية.

3.2 الحوسبة المتوازية (Parallel Computing) وحوسبة الأداء العالي (HPC)

الحوسبة المتوازية هي الاستخدام المتزامن لعدة موارد حاسوبية لحل مشكلة حسابية واحدة. يتم تقسيم المشكلة الكبيرة إلى أجزاء صغيرة يمكن حلها في وقت واحد.

3.2.1 تصنيف فلين (Flynn's Taxonomy)

هو التصنيف الأكاديمي المعتمد للمعماريات الحاسوبية:

1. **SISD (Single Instruction, Single Data):** الحاسوب التقليدي القديم (تعليمية واحدة لبيانات واحدة).
2. **SIMD (Single Instruction, Multiple Data):** تعليمية واحدة تطبق على بيانات متعددة في آن واحد. هذا هو أساس عمل وحدات معالجة الرسومات (GPUs) ومعالجات المتجهات، وهو جوهري في عمليات المصفوفات في الذكاء الاصطناعي.
3. **MIMD (Multiple Instruction, Multiple Data):** تعليمات مختلفة لبيانات مختلفة (مثل الحواسيب العملاقة الموزعة).

3.2.2 الحوسبة العنقودية (Cluster Computing)

ربط مجموعة من الحواسيب المستقلة (Nodes) عبر شبكة محلية سريعة لتعمل كحاسوب فائق واحد. تُستخدم هذه التقنية في بناء "مزارع الخوادم (Server Farms)" ومراكز البيانات.

3.3 وحدات التسريع (Accelerator Units) GPU, TPU, NPU

لم يعد المعالج المركزي (CPU) كافياً لمتطلبات العصر، مما أدى لظهور "المسرعات (Accelerators)"

3.3.1 وحدة معالجة الرسومات (GPU) كحاسوب عام (GPGPU)

- البنية: تعتمد على آلاف الأنوية الصغيرة المصممة للموازاة الهائلة (Massive Parallelism).
- **CUDA & OpenCL:** هي أطر برمجية سمحت للمبرمجين باستخدام GPU في غير الرسومات، مثل المحاكاة العلمية وتدريب الشبكات العصبية.

3.3.2 وحدة المعالجة المتوترية (TPU - Tensor Processing Unit)

- **التخصص:** صممها Google خصيصاً لتسريع عمليات جبر المصفوفات (Matrix Operations) التي يعتمد عليها التعلم العميق (Deep Learning) باستخدام مكتبة TensorFlow.
- **المصفوفات الانقباضية (Systolic Arrays):** معمارية تسمح بتدفق البيانات عبر مصفوفة من وحدات المعالجة دون الحاجة للعودة للذاكرة مراراً، مما يوفر سرعة وكفاءة طاقة هائلة.

3.3.3 وحدة المعالجة العصبية (NPU)

- دمجت مؤخراً في الهواتف الذكية والحواسيب الشخصية (AI PC) لمعالجة مهام الذكاء الاصطناعي محلياً (On-device AI) مثل التعرف على الوجه ومعالجة الصوت، لتخفيف العبء عن المعالج الرئيسي والحفاظ على الخصوصية.

3.4 الحوسبة السحابية (Cloud Computing): IaaS, PaaS, SaaS

الحوسبة السحابية ليست مجرد "تخزين على الإنترنت"، بل هي نموذج لتقديم الموارد الحاسوبية كخدمة (Utility) مثل الكهرباء والماء.

3.4.1 نماذج الخدمة (Service Models)

1. البنية التحتية كخدمة (IaaS): استئجار عتاد افتراضي (خوادم، تخزين، شبكات). مثال AWS : EC2, Google Compute Engine. تمنح أقصى درجات التحكم.
2. المنصة كخدمة (PaaS): توفير بيئة لتطوير ونشر التطبيقات دون القلق بشأن إدارة الخوادم ونظم التشغيل. مثال. Google App Engine, Heroku.
3. البرمجيات كخدمة (SaaS): استخدام التطبيقات مباشرة عبر الإنترنت. مثال Gmail, Office : 365, ChatGPT.

3.4.2 نماذج النشر (Deployment Models)

- السحابة العامة (Public Cloud): موارد مشتركة بين الجميع (مثل Azure).
- السحابة الخاصة (Private Cloud): بنية تحتية مخصصة لمؤسسة واحدة (للأمان العالي).
- السحابة الهجينة (Hybrid Cloud): دمج النموذجين، مما يسمح بمرونة نقل البيانات والتطبيقات بينهما.

3.5 الحوسبة الطرفية (Edge Computing) وإنترنت الأشياء (IoT)

- مع تضخم حجم البيانات، أصبح إرسال كل شيء للسحابة مكلفاً وبطيئاً.
- المفهوم: الحوسبة الطرفية تعني معالجة البيانات بالقرب من موقع إنشائها (عند "طرف" الشبكة) بدلاً من إرسالها لمركز بيانات بعيد.

- الأهمية في إنترنت الأشياء (IoT) في السيارات ذاتية القيادة، لا يمكن انتظار استجابة السحابة للتوقف المفاجئ. يجب أن يتخذ المعالج المحلي القرار في أجزاء من الثانية. الحوسبة الطرفية تقلل من الكمون (Latency) وتوفر استهلاك النطاق الترددي (Bandwidth).

3.6 لمحة تمهيدية عن الحوسبة الكمية (Quantum Computing)

نحن نقف على أعتاب ثورة فيزياء الكم التي ستغير مفهوم الحوسبة للأبد.

3.6.1 الكيوبت (Qubit) مقابل البت (Bit)

- الحاسوب التقليدي يستخدم "البت" (0 أو 1).
- الحاسوب الكمي يستخدم "الكيوبت" الذي يمكن أن يكون 0 و 1 في نفس الوقت بفضل ظاهرة

التراكب الكمي (Superposition)

3.6.2 التشابك الكمي (Entanglement)

ظاهرة فيزيائية يرتبط فيها جسيमान ببعضهما بحيث يؤثر تغيير حالة أحدهما فوراً على الآخر مهما بعدت المسافة. هذا يسمح بإجراء عمليات منطقية معقدة بسرعة تفوق الحواسيب التقليدية بملايين المرات في مسائل محددة.

3.6.3 التأثير المستقبلي

- كسر التشفير: خوارزمية شور (Shor's Algorithm) تهدد بكسر تشفير RSA الحالي.
- اكتشاف الأدوية والمواد: القدرة على محاكاة التفاعلات الجزيئية بدقة متناهية.
- تحسين الذكاء الاصطناعي: تسريع خوارزميات البحث والتحسين (Optimization) بشكل هائل.

الفصل الرابع

المنظومات البرمجية (Software) وبيئات التشغيل

4.1 مفهوم البرمجيات وتصنيفاتها

إذا كان العتاد (Hardware) يمثل جسد الحاسوب، فإن البرمجيات (Software) هي الروح التي تمنحه القدرة على العمل. البرمجيات هي مجموعة من التعليمات والبيانات المنظمة التي تخبر العتاد ماذا يفعل وكيف يفعله. من الناحية الأكاديمية، البرمجيات هي تجريد (Abstraction) يعزل المستخدم عن التعقيد الفيزيائي للدارات الإلكترونية.

4.1.1 برمجيات النظام (System Software)

هي الطبقة الأساسية التي تتعامل مباشرة مع العتاد وتوفر منصة لتشغيل البرامج الأخرى.

- **نظام التشغيل (OS):** المدير التنفيذي للحاسوب (سيتم تفصيله لاحقاً).
- **برامج تشغيل الأجهزة (Device Drivers):** هي "الترجمات" التي تسمح لنظام التشغيل بالتحدث مع قطع العتاد المحددة) مثل تعريف بطاقة الشاشة الذي يسمح للعبة باستخدام قدرات الـ GPU.
- **أدوات النظام (Utilities):** برامج لصيانة وتحليل النظام (مثل برامج إلغاء التجزئة، ومكافحة الفيروسات).

4.1.2 البرمجيات التطبيقية (Application Software)

هي البرامج المصممة لمساعدة المستخدم النهائي (End-User) على إنجاز مهام محددة.

- **تطبيقات عامة:** مثل معالجات النصوص ومتصفحات الويب.
- **تطبيقات متخصصة:** مثل برامج المحاسبة، أو بيئات تطوير الذكاء الاصطناعي) مثل Jupyter Notebooks.

4.1.3 لغات البرمجة – (Programming Languages) لمحة تصنيفية

هي الأدوات التي تُكتب بها البرمجيات. تنقسم حسب مستواها من العتاد:

- **لغات منخفضة المستوى (Low-Level):** قريبة جداً من لغة الآلة، مثل Assembly. توفر تحكماً كاملاً بالذاكرة والمعالج ولكنها صعبة التعلم.
- **لغات عالية المستوى (High-Level):** أقرب للغة البشر، مثل Python و Java. تعتمد على "مترجم (Compiler)" أو "مفسر (Interpreter)" لتحويل الكود إلى لغة الآلة. في علوم البيانات، يفضل اللغات عالية المستوى لسهولة بناء النماذج الرياضية المعقدة بها.

4.2 أنظمة التشغيل: المفهوم والوظائف الأساسية

نظام التشغيل (Operating System) هو أهم برنامج في الحاسوب. هو وسيط برمجي (Interface) بين المستخدم/التطبيقات وبين العتاد الصلب. بدونه، سيضطر المبرمج لكتابة كود خاص للتحكم في دوران القرص الصلب لكل عملية حفظ ملف.

نواة النظام (The Kernel): هي القلب النابض لنظام التشغيل، وهو الجزء الذي يُحمل في الذاكرة عند الإقلاع ويبقى فيها حتى إيقاف التشغيل. النواة مسؤولة عن إدارة الموارد الحرجة (الذاكرة، المعالج، الأجهزة). تعمل النواة في وضع محمي يسمى **Kernel Mode** يمنع التطبيقات العادية (التي تعمل في **User Mode**) من العبث بمكونات النظام الحساسة.

الوظائف الجوهرية:

1. إدارة الموارد: توزيع وقت المعالج ومساحة الذاكرة.
2. توفير واجهة المستخدم: سواء كانت سطر أوامر (CLI) أو واجهة رسومية (GUI).
3. إدارة الملفات: تنظيم البيانات في وحدات تخزين.

4.3 إدارة العمليات والذاكرة والتخزين

هذه المفاهيم هي المعيار الذي يميز المبرمج المحترف عن الهاوي، وفهمها ضروري لتحسين أداء خوارزميات الذكاء الاصطناعي.

أولاً: إدارة العمليات (Process Management)

- **العملية (Process):** هي برنامج قيد التنفيذ. (Instance of a running program).
- **الجدولة (Scheduling):** بما أن المعالج (في الأغلب) لا يمكنه تنفيذ إلا تعليمة واحدة لكل نواة في اللحظة الواحدة، يقوم نظام التشغيل بتبديل التنفيذ بين العمليات بسرعة هائلة (أجزاء من الملي ثانية) ليعطي وهماً بالتزامن. تُعرف هذه العملية بـ **تبديل السياق (Context Switching)**.
- **تعدد المهام (Multitasking):** قدرة النظام على التعامل مع عدة عمليات "في نفس الوقت" عن طريق الجدولة الذكية.

ثانياً: إدارة الذاكرة (Memory Management)

- **الذاكرة الافتراضية (Virtual Memory):** تقنية حيوية تسمح بتشغيل برامج حجمها أكبر من الذاكرة الفعلية (RAM). يقوم النظام بحجز مساحة على القرص الصلب) تسمى Paging File أو Swap واستخدامها كامتداد للذاكرة. عندما تمتلئ الرام، تُنقل البيانات الأقل استخداماً إلى القرص (Swap out) لإفراح المجال للبيانات النشطة.
- *أهميتها للذكاء الاصطناعي: نماذج البيانات الكبيرة غالباً ما تستهلك الرام بالكامل، وفهم الذاكرة الافتراضية يفسر سبب البطء الشديد عند تجاوز حد الذاكرة الفيزيائية.*

4.4 نظم الملفات (File Systems): FAT32, NTFS, exFAT, ext4, APFS

نظام الملفات هو الطريقة التي ينظم بها نظام التشغيل البيانات على القرص؛ فهو بمثابة "الفهرس" أو "الخريطة" التي تحدد موقع كل بايت. بدون نظام ملفات، القرص الصلب هو مجرد مساحة خام من البيانات غير المفهومة.

1. FAT32 (File Allocation Table)

- قديم ولكنه مدعوم من جميع الأجهزة تقريباً (Windows, Mac, Linux, Game Consoles).
- **العيب القاتل:** لا يدعم تخزين ملف واحد أكبر من 4 جيجابايت.

2. NTFS (New Technology File System)

- النظام الافتراضي لـ Windows.
- يدعم الملفات الضخمة، والأمان (الأذونات)، والتدوين (**Journaling**) الذي يحمي البيانات عند انقطاع الكهرباء المفاجئ.

3. exFAT (Extensible FAT)

- صمم ليكون جسراً بين FAT32 و NTFS. يدعم الملفات الكبيرة ومتوافق مع Windows و Mac ، مما يجعله مثالياً لوحدة التخزين الخارجية (Flash Drives).

4. ext4 (Fourth Extended Filesystem)

- النظام القياسي لتوزيعات Linux يتميز بالكفاءة العالية والموثوقية.
- ملاحظة: معظم خوادم الويب وبيئات التدريب السحابية للذكاء الاصطناعي تعمل على Linux وتستخدم ext4 أو XFS.

5. APFS (Apple File System)

- صمم خصيصاً لأقرص الحالة الصلبة (SSD) في أجهزة Mac. يتميز بسرعة التشفير ونسخ الملفات الفوري.

4.5 بينات العمل الافتراضية والحاويات (Virtualization & Containers)

هذا القسم يمثل حجر الزاوية للحوسبة السحابية الحديثة ونشر تطبيقات علوم البيانات.

4.5.1 المحاكاة الافتراضية (Virtualization)

هي تقنية تسمح بإنشاء نسخ افتراضية من موارد الحاسوب.

- الآلة الافتراضية: (**Virtual Machine - VM**) هي حاسوب كامل "وهمي" يعمل داخل حاسوبك الحقيقي. تتطلب وجود **Hypervisor** (مدير الأجهزة الافتراضية) لتشغيلها.
- الميزة: عزل كامل (نظام تشغيل كامل مستقل).
- العيب: تستهلك موارد كبيرة) لأن كل VM تحتاج تحميل نظام تشغيل كامل خاص بها.

4.5.2 الحاويات (Containers)

هي تطور للمحاكاة الافتراضية، وأشهر تقنياتها **Docker**.

- **الفكرة:** بدلاً من تثبيت نظام تشغيل كامل لكل تطبيق، تتشارك الحاويات نواة نظام التشغيل المضيف (Host Kernel)، ولكنها تعزل التطبيق ومكتباته فقط.
- **الأهمية القصوى في علوم البيانات:** تحل مشكلة "إنه يعمل على جهازي فقط (It works on my machine) باستخدام Docker، يمكنك تغليف كود البايثون الخاص بك مع جميع المكتبات والإصدارات المحددة في حاوية، وضمان أنها ستعمل بنفس الطريقة تماماً على أي خادم سحابي أو جهاز آخر.

الفصل الخامس

الدليل التشغيلي لنظامي Windows 10 و Windows 11

5.1 التعرف على واجهة المستخدم الرسومية (GUI) وسطح المكتب

نظام التشغيل Windows (الذي تبلغ حصته السوقية أكثر من 70% من الحواسيب المكتبية عالمياً) يعتمد بشكل جوهري على واجهة المستخدم الرسومية (Graphical User Interface - GUI) التي تستبدل الأوامر النصية بعناصر مرئية (نوافذ، أيقونات، قوائم).

الفروقات الجوهرية بين Windows 10 و Windows 11 رغم تشابه النواة (Kernel) ، إلا أن Windows 11 تقدم إعادة تصميم بصرية شاملة تُعرف بـ "Mica Design" و "Fluent Design System".

- شريط المهام (Taskbar) في Windows 11 ، أصبحت الأيقونات وسيطية (Centered) افتراضياً لمحاكاة تجربة الأجهزة اللوحية، مع قائمة "ابدأ" مبسطة تتخلى عن "Live Tiles".
- تخطيطات المحاذاة (Snap Layouts) ميزة متقدمة في Windows 11 تسمح بتقسيم الشاشة تلقائياً لنوافذ متعددة بضغطة زر، وهو أمر حيوي للمبرمجين الذين يحتاجون لرؤية الكود والمتصفح والتوثيق في آن واحد.

5.2 إدارة الملفات والمجلدات وتصنيفها

إدارة الملفات ليست مجرد نسخ ولصق، بل هي فهم لهيكلية البيانات على القرص.

مستكشف الملفات (File Explorer) هو الأداة الرئيسية لتصفح نظام الملفات. يعتمد الهيكل الهرمي (Hierarchical Structure) الذي يبدأ من الدليل الجذري (Root Directory) للقرص (غالباً C:\).

مفاهيم متقدمة للمطورين:

1. امتدادات الملفات (File Extensions): هي اللاحقة التي تلي النقطة في اسم الملف (مثل .py). ملفات بايثون، .csv للبيانات). يجب دائماً تفعيل خيار "Show file name extensions" لتجنب الوقوع في فخ الملفات الخبيثة (مثل virus.pdf.exe).

2. المسارات:(Paths)

- **المسار المطلق (Absolute Path):** العنوان الكامل للملف (مثال : C:\Users\Admin\Projects\data.csv).
- **المسار النسبي (Relative Path):** العنوان بالنسبة للمجلد الحالي (مثال ..\data.csv) :
فهم هذا الفرق حاسم عند كتابة أكواد برمجية لضمان عملها على أجهزة مختلفة.

5.3 لوحة التحكم وتطبيق Settings وإدارة الأجهزة

يشهد ويندوز انتقالاً تدريجياً من لوحة التحكم الكلاسيكية (Control Panel) المبنية على معمارية Win32 القديمة، إلى تطبيق الإعدادات (Settings App) الحديث المبني على منصة UWP لإدارة الأجهزة: (Device Manager) أداة حيوية للمهندسين، تعرض شجرة بجميع المكونات المادية المتصلة بالحاسوب.

- **تحديث التعريفات (Drivers):** عند توصيل بطاقة رسومات (GPU) جديدة لتدريب الذكاء الاصطناعي، يتم التحقق من تعريفها هنا.
- **استكشاف التعارضات:** ظهور علامة تعجب صفراء يعني وجود خلل في التعريف أو تعارض في الموارد. (IRQ Conflict).
- **متغيرات البيئة (Environment Variables):** من أهم الإعدادات للمبرمجين (موجودة تحت System Properties > Advanced).
- **متغير PATH:** قائمة بالمجلدات التي يبحث فيها النظام عن الأوامر التنفيذية. لكي تعمل أوامر مثل python أو java في أي مكان في سطر الأوامر، يجب إضافة مسار تثبيتها إلى هذا المتغير.

5.4 إدارة المستخدمين والصلاحيات

يعتمد ويندوز نظاماً أمنياً متعدد المستخدمين. (Multi-user System)

1. أنواع الحسابات:

- **المسؤول (Administrator):** يملك صلاحيات كاملة لتثبيت البرامج وتعديل إعدادات النظام.

○ **المستخدم القياسي (Standard User):** يمكنه استخدام البرامج وتغيير إعداداته الشخصية، لكن لا يمكنه إجراء تغييرات تؤثر على النظام أو المستخدمين الآخرين. أمنياً، يُنصح باستخدام حساب قياسي للعمل اليومي.

2. **التحكم في حساب المستخدم (UAC - User Account Control):** هي تلك الشاشة المعتمدة التي تظهر لتطلب الموافقة عند محاولة برنامج إجراء تغييرات جذرية. هي خط الدفاع الأول ضد البرمجيات التي تحاول تثبيت نفسها خلسة في الخلفية.

5.5 أدوات الصيانة واستعادة النظام

لضمان استمرارية العمل (Business Continuity)، يوفر ويندوز أدوات صيانة مدمجة:

1. **تنظيف القرص (Disk Cleanup / Storage Sense):** إزالة الملفات المؤقتة ومخلفات التحديثات لتحرير المساحة.

2. **إلغاء التجزئة (Defragmentation) والتحسين:**

○ للأقرص الميكانيكية (HDD) إعادة ترتيب البيانات لتكون متجاورة، مما يسرع القراءة.

○ لأقرص الحالة الصلبة (SSD) يتم استخدام أمر **TRIM** بدلاً من إلغاء التجزئة، لإخبار القرص بالخلايا الفارغة للحفاظ على عمره الافتراضي.

3. **نقطة استعادة النظام (System Restore Point):** لقطة (Snapshot) لملفات النظام والريجستري في لحظة زمنية معينة. تسمح بالعودة بالزمن للوراء في حال تسبب تحديث أو برنامج جديد بانهيار النظام.

5.6 الأوامر الأساسية عبر CMD وPowerShell

بالنسبة لعالم البيانات، الواجهة الرسومية بطيئة وغير قابلة للأتمتة. لذا، إتقان سطر الأوامر ضرورة قصوى.

موجه الأوامر (Command Prompt - CMD)

هو المفسر التقليدي للأوامر في ويندوز) موروث من نظام (DOS.

• **cd:** لتغيير المجلد الحالي. (Change Directory)

- dir: لعرض محتويات المجلد. (Directory)
- mkdir: لإنشاء مجلد جديد. (Make Directory)
- ipconfig: لعرض إعدادات الشبكة وعنوان IP.
- ping: لاختبار الاتصال بخادم معين.

باورشيل (PowerShell)

هو بيئة أتمتة مهام (Task Automation) متقدمة جداً، مبنية على إطار عمل .NET.

- الفرق الجوهرية: في CMD ، المخرجات هي نصوص (Text) في PowerShell ، المخرجات هي كائنات (Objects) برمجية لها خصائص وطرق (Methods) ، مما يسمح بمعالجة بيانات معقدة جداً.
- مثال: يمكن باستخدام أمر واحد في PowerShell البحث في آلاف الملفات النصية، استخراج بيانات محددة، وتصديرها لملف Excel ، وهو ما يصعب جداً فعله بـ CMD.

5.7 استكشاف الأخطاء ومعالجتها (Troubleshooting)

القدرة على التشخيص هي مهارة أساسية.

1. إدارة المهام: (Task Manager) اختصار (Ctrl+Shift+Esc) :
 - مراقبة استهلاك الموارد (CPU, RAM, Disk) إذا كان النظام بطيئاً، هنا تعرف السبب.
 - إنهاء العمليات العالقة. (End Task)
2. عارض الأحداث: (Event Viewer)
 - السجل الأسود للنظام. يسجل كل خطأ، تحذير، أو معلومة. عند حدوث "شاشة الموت الزرقاء (BSOD)" ، نلجأ لهذا السجل لمعرفة كود الخطأ والسبب الجذري.
3. الوضع الآمن: (Safe Mode)
 - تشغيل ويندوز بالحد الأدنى من التعريفات والخدمات. يستخدم لعزل المشاكل؛ فإذا اختفت المشكلة في الوضع الآمن، فغالباً سببها برنامج مثبت أو تعريف جهاز غير متوافق.

الفصل السادس

شبكات الحاسوب والبنية التحتية للإنترنت

6.1 مقدمة في علم الشبكات وأهميتها في العصر الرقمي

شبكة الحاسوب (Computer Network) هي منظومة تتكون من جهازين أو أكثر، مرتبطة ببعضها عبر وسائط اتصال (Communication Media) لغرض تبادل البيانات والمشاركة في الموارد (مثل الطابعات، الملفات، التطبيقات).

في سياق علوم البيانات، الشبكات ليست مجرد وسيلة تواصل، بل هي "خطوط الأنابيب (Pipelines)" التي تتدفق عبرها البيانات الخام من المصادر (IoT Sensors, Web Logs) إلى مراكز المعالجة (Data Centers)، ومن ثم إلى المستخدم النهائي. كفاءة النموذج التحليلي تعتمد بشكل مباشر على زمن الانتقال (Latency) وعرض النطاق الترددي (Bandwidth) للشبكة.

6.2 تصنيف الشبكات حسب النطاق الجغرافي

يُعد النطاق الجغرافي المعيار الأساسي لتصنيف الشبكات، حيث يحدد البروتوكولات والأجهزة المستخدمة.

6.2.1 شبكة المنطقة الشخصية (PAN - Personal Area Network)

- المجال: أمتار قليلة حول الفرد.
- التقنية: تعتمد غالباً على Bluetooth أو Zigbee أو NFC.
- المثال: ربط الهاتف بالساعة الذكية وسماعات الرأس. في إنترنت الأشياء، تُستخدم لجمع البيانات الحيوية من المرضى.

6.2.2 شبكة المنطقة المحلية (LAN - Local Area Network)

- المجال: مبنى واحد، طابق، أو منزل.
- التقنية: تعتمد بشكل كلي تقريباً على تقنيات Ethernet (IEEE 802.3) و Wi-Fi (IEEE 802.11).
- السرعة: عالية جداً (تصل إلى 10 Gbps أو أكثر)، مع معدل خطأ منخفض جداً.

6.2.3 شبكة الحرم الجامعي (CAN - Campus Area Network)

- **المجال:** مجموعة مباني مترابطة (حرم جامعي، قاعدة عسكرية، مدينة طبية).
- **الخصائص:** شبكة خاصة تربط عدة شبكات LAN ببعضها، وتدار بالكامل من قبل المؤسسة دون الاعتماد على شركات الاتصالات العامة.

6.2.4 شبكة المنطقة الإقليمية (MAN - Metropolitan Area Network)

- **المجال:** مدينة كاملة.
- **التقنية:** تستخدم الألياف الضوئية أو تقنيات الميكروويف (WiMAX) سابقاً (لربط فروع المؤسسات داخل المدينة).

6.2.5 شبكة المنطقة الواسعة (WAN - Wide Area Network)

- **المجال:** دول، قارات، أو العالم بأسره.
- **التقنية:** تعتمد على الكابلات البحرية، الأقمار الصناعية، وشبكات الهواتف المحمولة.
- **أشهر مثال:** شبكة الإنترنت (The Internet) هي أكبر شبكة WAN في الوجود.

6.3 طوبولوجيا الشبكات (Network Topology)

تشير الطوبولوجيا إلى الهيكلية الهندسية لربط الأجهزة (Nodes) والروابط (Links).

1. **الخطية (Bus Topology):** كابل رئيسي واحد يربط جميع الأجهزة. (قديمة، تعاني من توقف الشبكة بالكامل عند قطع الكابل).
2. **الحلقية (Ring Topology):** البيانات تدور في اتجاه واحد). تستخدم في شبكات الألياف FDDI لزيادة الموثوقية).
3. **النجمية (Star Topology):** جميع الأجهزة تتصل بجهاز مركزي (Switch). هي الأكثر شيوعاً اليوم؛ فإذا تعطل كابل جهاز واحد، لا تتأثر بقية الشبكة.

4. الشبكية: (Mesh Topology) كل جهاز يتصل بكل الأجهزة الأخرى (Full Mesh) توفر أعلى درجات الموثوقية (Redundancy) وتستخدم في البنية التحتية للإنترنت ومراكز البيانات لضمان عدم انقطاع الخدمة.

6.4 وسائط الإرسال (Transmission Media)

الوسائط السلكية (Guided Media)

1. الأسلاك النحاسية المجدولة: (Twisted Pair - UTP/STP) الأكثر انتشاراً في شبكات LAN. الالتواء يقلل من التداخل الكهرومغناطيسي (EMI). تصنف إلى فئات (Cat5e, Cat6, Cat7) حسب السرعة والتردد.
2. الألياف الضوئية: (Fiber Optics) تستخدم الضوء لنقل البيانات عبر شعيرات زجاجية.
 - مزايها: سرعة هائلة (تيرابت/ثانية)، مناعة تامة ضد التشويش الكهربي، تغطي مسافات طويلة.
 - Single-mode: للمسافات البعيدة (بين المدن) باستخدام الليزر.
 - Multi-mode: للمسافات القصيرة (داخل مراكز البيانات) باستخدام LED.

الوسائط اللاسلكية (Unguided Media)

تعتمد على الطيف الكهرومغناطيسي (موجات الراديو، الميكروويف، الأشعة تحت الحمراء) لنقل البيانات عبر الهواء.

6.5 النماذج المرجعية OSI و TCP/IP

لفهم كيفية انتقال البيانات من تطبيق على جهازك إلى خادم في قارة أخرى، نستخدم نماذج طبقية (Layered Models).

نموذج - OSI (Open Systems Interconnection) المرجع النظري (7 طبقات)

1. الفيزيائية: (Physical) البتات (Bits) والإشارات الكهربائية/الضوئية.

2. ربط البيانات: **(Data Link)** العنوان الفيزيائية (MAC Address) والتحكم في الوصول للوسيط. الوحدة هنا تسمى "إطار. (Frame) "
3. الشبكة: **(Network)** العنوان المنطقية (IP Address) وتوجيه المسارات. (Routing) الوحدة تسمى "حزمة. (Packet) "
4. النقل: **(Transport)** ضمان وصول البيانات (Reliability) والتحكم بالتدفق. الوحدة تسمى "قطعة. (Segment) "
5. الجلسة: **(Session)** إدارة جلسات الاتصال بين التطبيقات.
6. التقديم: **(Presentation)** تشفير وضغط البيانات وتنسيقها. (JPEG, ASCII)
7. التطبيق: **(Application)** الواجهة التي يتعامل معها المستخدم. (HTTP, FTP)

نموذج - TCP/IP النموذج العملي للإنترنت (4 طبقات)

يدمج الطبقات العليا (5,6,7) في طبقة التطبيق، ويدمج السفلى (1,2) في طبقة الوصول للشبكة. هو اللغة الفعلية التي تتحدث بها الإنترنت اليوم.

6.6 التجهيزات الشبكية (Network Equipment)

1. **المبدل: (Switch)** جهاز ذكي يعمل في الطبقة الثانية. (Data Link) يربط الأجهزة في شبكة LAN ويحول البيانات فقط للجهاز المعني باستخدام عنوان MAC ، مما يقلل الازدحام.
2. **الموجه: (Router)** جهاز يعمل في الطبقة الثالثة. (Network) يربط شبكات مختلفة ببعضها) مثل ربط شبكة منزلك بشبكة مزود الخدمة. (ISP) وظيفته الأساسية اختيار أفضل مسار للحزمة (Packet) عبر الإنترنت.
3. **نقطة الوصول: (Access Point - WAP)** تحول الشبكة السلكية إلى لاسلكية. (Wi-Fi)
4. **الجدار الناري: (Firewall)** جهاز (أو برنامج) أمني يراقب حركة المرور ويسمح أو يمنع الاتصالات بناءً على قواعد محددة.

6.7 بروتوكولات الإنترنت الأساسية

البروتوكول هو مجموعة من القواعد التي تحكم كيفية التنسيق والاتصال.

- **IP (Internet Protocol):** نظام العنونة IPv4 و IPv6 الذي يعطي هوية رقمية لكل جهاز.
- **TCP (Transmission Control Protocol):** بروتوكول "موثوق" يضمن وصول البيانات كاملة وبترتيب صحيح (يستخدم في الويب والبريد الإلكتروني).
- **UDP (User Datagram Protocol):** بروتوكول "سريع" لا يضمن الوصول، يضحى بالموثوقية مقابل السرعة (يستخدم في البث المباشر والألعاب أونلاين).
- **DNS (Domain Name System):** دليل الهاتف للإنترنت؛ يترجم أسماء النطاقات (www.google.com) إلى عناوين IP رقمية.
- **DHCP (Dynamic Host Configuration Protocol):** يوزع عناوين IP تلقائياً للأجهزة عند اتصالها بالشبكة.

6.8 الشبكات اللاسلكية و 5G وإنترنت الأشياء (IoT)

في عصر 2026، أصبح الاتصال اللاسلكي هو المعيار.

- **Wi-Fi 6 (802.11ax) وما بعده:** يركز على الكفاءة في الأماكن المزدحمة (مثل الملاعب) وليس فقط السرعة القصوى، باستخدام تقنيات مثل OFDMA.
- **الجيل الخامس (5G):** ثورة في الاتصالات الخلوية. لا تقدم فقط سرعة عالية، بل الأهم هو **الكمون المنخفض جداً (Ultra-low Latency)** والموثوقية العالية، مما يجعلها العمود الفقري للسيارات ذاتية القيادة والجراحة عن بعد.
- **إنترنت الأشياء (IoT):** ربط مليارات الأجهزة "الغبية" (ثلاجات، مصابيح، حساسات صناعية) بالإنترنت. تتطلب بروتوكولات خفيفة الوزن وموفرة للطاقة (مثل MQTT) لأن أجهزة الاستشعار قد تعمل لسنوات على بطارية صغيرة.

الفصل السابع

الأمن السيبراني – الحصن الرقمي في عصر 2026

7.1 مبادئ أمن المعلومات (ثالوث CIA)

يعتمد الأمن السيبراني (Cybersecurity) على ركيزة أساسية تُعرف بـ "مثلث الأمن (CIA Triad)" أي نظام أمني ناجح يجب أن يوازن بين هذه الأضلاع الثلاثة، وأي خلل في أحدها يعني اختراقاً للنظام.

1. **السرية (Confidentiality):** ضمان أن المعلومات لا يتم كشفها إلا للمصرح لهم.

○ مثال: تشفير بيانات المرضى بحيث لا يراها إلا الطبيب المعالج.

○ آليات التحقيق: التشفير (Encryption)، قوائم التحكم بالوصول (ACLs)، والمصادقة الثنائية (2FA).

2. **النزاهة أو السلامة (Integrity):** ضمان أن البيانات لم يتم العبث بها أو تعديلها من قبل جهة غير مصرح لها (سواء كان التعديل بشرياً أو بسبب خطأ تقني).

○ مثال: التأكد من أن رصيد البنك لم يتغير أثناء عملية التحويل.

○ آليات التحقيق: دوال التجزئة (Hashing)، التوقيع الرقمي (Digital Signature).

3. **التوافر (Availability):** ضمان أن البيانات والأنظمة متاحة للمستخدمين المصرح لهم عند الحاجة إليها.

○ مثال: موقع التجارة الإلكترونية يجب أن يعمل وقت الذروة دون انقطاع.

○ آليات التحقيق: النسخ الاحتياطي (Backups)، وتوزيع الأحمال (Load Balancing)، وخطط التعافي من الكوارث (Disaster Recovery).

7.2 تصنيف التهديدات والبرمجيات الخبيثة (Malware)

البرمجيات الخبيثة (Malicious Software) هي أي كود برمجي صمم لإلحاق الضرر أو السرقة.

7.2.1 الفيروسات (Viruses)

برامج تحتاج إلى "حاضن (Host File) "التعمل. ترفق نفسها بملفات تنفيذية (مثل .exe) وتنتشر عند تشغيل الملف المصاب. تشبه الفيروس البيولوجي في حاجتها لخلية حية لتتكاثر.

7.2.2 الديدان (Worms)

أخطر من الفيروسات لأنها ذاتية الانتشار (Self-replicating) ولا تحتاج لتدخل بشري أو ملف حاضن. تستغل الثغرات في الشبكات لتنسخ نفسها من جهاز لآخر بسرعة هائلة، مما يسبب اختناقاً في الشبكة). مثال شهير: دودة (WannaCry).

7.2.3 أحصنة طروادة (Trojans)

برامج تبدو شرعية ومفيدة (مثل لعبة مجانية أو أداة تفعيل)، لكنها تخفي بداخلها كوداً خبيثاً يفتح "باباً خلفياً" (Backdoor) للمخترق للتحكم بالجهاز عن بعد.

7.2.4 برامج الفدية (Ransomware)

الكابوس الأكبر في العصر الحديث. تقوم بتشفير جميع ملفات الضحية باستخدام خوارزميات معقدة، وتطلب فدية مالية) غالباً بعملة Bitcoin مقابل مفتاح فك التشفير.

- التأثير: شلل تام للمستشفيات، البنوك، والشركات.

7.2.5 برامج التجسس (Spyware) والـ Rootkits

- Spyware : تجمع المعلومات (كلمات المرور، تاريخ التصفح) وترسلها للمهاجم.
- Rootkits : أدوات متقدمة جداً تخفي نفسها في عمق نظام التشغيل) أحياناً في النواة (Kernel ، مما يجعل اكتشافها بواسطة مضادات الفيروسات التقليدية شبه مستحيل.

7.3 الهجمات الشبكية (Network Attacks)

هجوم حجب الخدمة (DoS/DDoS)

- DoS (Denial of Service) : إغراق الخادم بطلبات وهمية تفوق قدرته على المعالجة، مما يجعله يتوقف عن خدمة المستخدمين الحقيقيين.

- **DDoS (Distributed DoS)** : استخدام جيش من الأجهزة المخترقة (Botnet) لشن الهجوم من آلاف المصادر في وقت واحد، مما يجعل صده صعباً جداً.

هجوم الرجل في المنتصف (MITM - Man-in-the-Middle)

يقوم المهاجم باعتراض الاتصال بين الضحية والخادم) مثلاً عبر شبكة Wi-Fi عامة غير آمنة(، فيقرأ الرسائل أو يعدلها دون علم الطرفين.

حقن (SQL Injection)

هجوم يستهدف قواعد البيانات. يقوم المهاجم بكتابة أوامر SQL خبيثة في حقول الإدخال (مثل حقول تسجيل الدخول)، مما يخدع قاعدة البيانات لتكشف عن محتوياتها السرية أو تحذفها.

- **الحل**: استخدام الاستعلامات المجهزة مسبقاً (Prepared Statements) في البرمجة.

البرمجة عبر المواقع (XSS - Cross-Site Scripting)

حقن كود JavaScript خبيث في صفحات ويب موثوقة. عندما يزور الضحية الصفحة، يتم تنفيذ الكود في متصفحه وسرقة ملفات تعريف الارتباط (Cookies) الخاصة بجلسته دخوله.

7.4 الهندسة الاجتماعية واختراق العقول (Social Engineering)

أضعف حلقة في السلسلة الأمنية هي الإنسان. الهندسة الاجتماعية هي فن التلاعب النفسي لدفع الناس للكشف عن معلومات سرية.

- **التصيد (Phishing)**: رسائل بريد إلكتروني تنتحل صفة جهات رسمية (بنك، شركة شحن) تحتوي روابط لصفحات مزورة.
- **Pretexting**: اختلاق سيناريو (مثال: "أنا من قسم الدعم الفني وأحتاج كلمة مرورك لتحديث النظام").
- **Baiting**: ترك فلاش ميموري (USB) مصاب في مكان عام، بانتظار فضول الضحية ليوصله بجهازه.

7.5 علم التشفير (Cryptography)

هو العلم الذي يحول البيانات الواضحة (Plaintext) إلى رموز غير مفهومة (Ciphertext).

التشفير المتماثل (Symmetric Encryption)

يستخدم مفتاحاً واحداً للتشفير وفك التشفير.

- الميزة: سريع جداً (مناسب لتشفير القرص الصلب أو ملفات الفيديو).
- العيب: صعوبة تبادل المفتاح بأمان بين الطرفين.
- أمثلة: خوارزمية AES (Advanced Encryption Standard).

التشفير غير المتماثل (Asymmetric Encryption)

يستخدم زوجاً من المفاتيح:

1. مفتاح عام (Public Key): متاح للجميع، يستخدم للتشفير.
 2. مفتاح خاص (Private Key): سري للغاية، يستخدم لفك التشفير.
- السيناريو: لكي ترسل رسالة سرية لأحمد، تشفرها بمفتاح أحمد "العام". لا يمكن لأحد فكها إلا أحمد باستخدام مفتاحه "الخاص".
 - أمثلة: خوارزمية RSA و ECC. هذا هو أساس أمان الويب (HTTPS).

7.6 استراتيجيات الدفاع

جدران الحماية (Firewalls)

جهاز أو برنامج يراقب حركة المرور ويطبق قواعد صارمة (مثال: منع أي اتصال قادم من الخارج على المنفذ 3389).

- (Deep Packet Inspection) (Next-Generation Firewall (NGFW): يفحص محتوى الحزم (Deep Packet Inspection) وليس فقط العناوين).

أنظمة كشف ومنع التسلل (IDS/IPS)

- "IDS (Intrusion Detection System): نظام إنذار" يراقب الشبكة وينبه المسؤول عند اكتشاف نشاط مشبوه.
- "IPS (Intrusion Prevention System): نظام دفاعي" يتدخل تلقائياً ويقطع الاتصال عند اكتشاف هجوم.

التحكم بالوصول (Access Control)

مبدأ "الامتيازات الأقل: (Least Privilege)" يجب منح المستخدم الحد الأدنى فقط من الصلاحيات اللازمة لإنجاز عمله، وليس أكثر.

7.7 القوانين والتشريعات السيبرانية وحماية البيانات

في عالم العولمة الرقمية، ظهرت الحاجة لتنظيم قانوني.

- (GDPR) اللائحة العامة لحماية البيانات - الاتحاد الأوروبي: (التشريع الأشد صرامة عالمياً. يفرض غرامات هائلة على الشركات التي تسيء استخدام بيانات المستخدمين أو تفشل في حمايتها. يمنح المستخدم "الحق في النسيان" (Right to be Forgotten)."
- الامتثال (Compliance): يجب على علماء البيانات التأكد من أن نماذج الذكاء الاصطناعي لا تنتهك خصوصية الأفراد ولا تستخدم بيانات حساسة دون موافقة صريحة.

نهاية الفصل السابع

موسوعة أساسيات الحاسوب والرياضيات لعلوم البيانات والذكاء الاصطناعي – 2026

المحور الثاني: البرمجيات، الشبكات، الأمن، والإنتاجية الرقمية

الفصل الثامن

الإنتاجية الرقمية وتطبيقات المكتب (Office & Cloud)

8.1 مقدمة في الحزم المكتبية وأدوارها المهنية والأكاديمية

لم تعد الحزم المكتبية (Office Suites) مجرد أدوات للكتابة والحساب البسيط، بل تطورت لتصبح بيئات عمل متكاملة تدعم الأتمتة (Automation) والتحليل الإحصائي والتعاون السحابي. في سياق علوم البيانات، غالباً ما يكون برنامج الجداول الإلكترونية هو "المحطة الأولى" لتنظيف البيانات واستكشافها قبل نقلها إلى بيئات برمجية معقدة مثل Python أو R.

تتنافس في هذا المجال منظومتان رئيسيتان:

1. (Microsoft 365 سابقاً): **Office** المعيار الصناعي للشركات الكبرى، يتميز بقوة الميزات وعمقها.

2. **Google Workspace**: المعيار للشركات الناشئة والعمل التعاوني المرن، يتميز بالعمل السحابي الأصيل (Cloud Native).

8.2 معالجة النصوص (Word Processing)

برنامج معالجة النصوص (مثل Microsoft Word) هو أداة الباحث الأساسية لصياغة الأطروحات والتقارير التقنية. الاحترافية هنا تعني الانتقال من التنسيق اليدوي (تلوين كل عنوان يدوياً) إلى التنسيق الهيكلي الآلي.

8.2.1 التنسيق المتقدم وبناء المستندات الأكاديمية

- **الأنماط (Styles)**: هي حجر الزاوية في المستندات الطويلة. بدلاً من تغيير حجم الخط لكل عنوان يدوياً، نستخدم أنماط (Heading 1, Heading 2, Normal).

○ **الفائدة:** توليد جدول المحتويات (Table of Contents) تلقائياً بنقرة زر، وضمان اتساق المستند كاملاً.

• **فواصل المقاطع (Section Breaks):** تسمح بتغيير تخطيط الصفحة في جزء محدد من المستند (مثلاً: جعل صفحة واحدة "أفقية" لعرض جدول عريض داخل مستند "عمودي"، أو تغيير ترقيم الصفحات من أرقام رومانية إلى عربية).

8.2.2 أدوات البحث العلمي والمراجع

إدارة المراجع يدوياً هي خطأ فادح في العصر الرقمي.

- **إدارة المصادر (Citation & Bibliography):** يوفر Word أدوات مدمجة لإضافة الاقتباسات وتوليد قائمة المراجع تلقائياً وفق معايير عالمية (APA, IEEE, MLA).
- **التسميات التوضيحية (Captions):** ترقيم الصور والجدول تلقائياً (مثال: "شكل 1-1")، وإنشاء فهرس للأشكال (Table of Figures) يتم تحديثه ديناميكياً عند إضافة أو حذف صورة.

8.2.3 دمج المراسلات (Mail Merge)

أداة قوية للأتمتة، تسمح بإنشاء آلاف المستندات المخصصة (رسائل، شهادات، تقارير أداء) انطلاقاً من قالب واحد وقاعدة بيانات) ملف Excel).

- **تطبيق عملي:** إرسال تقارير درجات مخصصة لـ 500 طالب بضغطة زر واحدة.

8.3 الجداول الإلكترونية (Spreadsheets)

يُعد Excel أو "Google Sheets" سكين الجيش السويسري "لبيانات الأعمال. بالنسبة لعالم البيانات، هو أداة التحليل الاستكشافي الأولى (EDA).

8.3.1 أنواع البيانات وتنسيق الخلايا

من الضروري التمييز بين القيمة المخزنة (Stored Value) والقيمة المعروضة (Displayed Value).

- **مثال:** التاريخ 45300 قد يظهر كـ 12/01/2024. فهم هذا التخزين الرقمي للتواريخ ضروري عند إجراء عمليات حسابية عليها (مثل حساب الفرق بين تاريخين).

8.3.2 الدوال والصيغ المتقدمة

• دوال البحث: (Lookup Functions)

- VLOOKUP: الدالة الكلاسيكية لربط الجداول (بطيئة وعرضة للأخطاء).
- XLOOKUP: البديل الحديث القوي، يسمح بالبحث في أي اتجاه ولا يتأثر بإضافة أعمدة جديدة.

• الدوال المنطقية والشرطية IFS, COUNTIFS, SUMIFS: لاستخراج إحصائيات دقيقة بناءً على معايير متعددة.

• معالجة النصوص TEXTSPLIT, TRIM, CONCAT: لتنظيف البيانات (Data Cleaning) قبل التحليل.

8.3.3 الجداول المحورية (Pivot Tables)

هي الأداة الأقوى في Excel لتلخيص البيانات الضخمة. (Summarization) تسمح بتحويل نصف مليون سطر من بيانات المبيعات الخام إلى تقرير ملخص يظهر المبيعات حسب "المنطقة" و"المنتج" في ثوانٍ معدودة، مع إمكانية الفرز والتصفية التفاعلية. (Slicers)

8.3.4 أدوات تحليل البيانات (Data Analysis Toolpak)

إضافة برمجية (Add-in) تحول Excel إلى برنامج إحصائي مصغر. يمكنها إجراء اختبارات متقدمة مثل:

- تحليل الانحدار (Regression Analysis) للتنبؤ.
- اختبارات (t-Test) وتحليل التباين. (ANOVA)
- تحليل الارتباط (Correlation) لدراسة العلاقات بين المتغيرات.

8.4 العروض التقديمية (Presentations)

في علوم البيانات، المهارة التقنية وحدها لا تكفي؛ يجب أن تمتلك مهارة سرد القصص بالبيانات (Data Storytelling).

8.4.1 مبادئ التصميم الجيد للعرض

- قاعدة 30/20/10 (غاي كاواساكي): لا تزيد عن 10 شرائح، لا تتجاوز 20 دقيقة، ولا تستخدم خطأ أصغر من 30 نقطة.
- التصور البصري: تجنب جداول البيانات المكدسة؛ استخدم الرسوم البيانية المبسطة (Charts) والإنفوجرافيك لإيصال الفكرة فوراً.

8.4.2 الشرائح الرئيسية (Slide Master)

أداة للمحترفين تضمن توحيد هوية العرض. بدلاً من وضع شعار الشركة يدوياً على 50 شريحة، يتم وضعه مرة واحدة في "الشريحة الرئيسية"، ليظهر تلقائياً في كل الشرائح وبنفس الموقع الدقيق، مما يسهل التعديل اللاحق.

8.4.3 الحركات والانتقالات (Morph & Zoom)

- **Morph Transition:** ميزة حديثة ثورية تسمح بتحريك العناصر بسلسلة سينمائية بين الشرائح، وهي ممتازة لتوضيح التغيرات في الرسوم البيانية أو تطور المخططات الزمنية.

8.5 أدوات التعاون السحابي (Cloud Collaboration)

انتقل العمل في 2026 من الملفات المرفقة عبر البريد الإلكتروني (v1, v2, final_v3.docx) إلى العمل المتزامن. (Simultaneous Co-authoring)

- **التحرير في الوقت الفعلي (Real-time Editing):** إمكانية عمل فريق كامل على نفس المستند أو ورقة العمل في نفس اللحظة، مع رؤية مؤشرات الكتابة لكل عضو.
- **سجل الإصدارات (Version History):** شبكة أمان تسمح بالعودة لأي نسخة سابقة من الملف، ومعرفة من قام بأي تعديل وتاريخه بالدقيقة والثانية.
- **Teams / Slack:** منصات دمجت المحادثة الفورية مع إدارة الملفات وعقد الاجتماعات المرئية، لتصبح "مكتباً افتراضياً" بدلاً عن البريد الإلكتروني الداخلي.

الفصل التاسع

مدخل إلى البرمجة والخوارزميات (النسخة المرجعية الشاملة)

9.1 ما هي الخوارزمية؟ الفلسفة والنشأة والتكوين

في قلب الثورة الرقمية، لا يقبع "الكود" بحد ذاته، بل الخوارزمية. (Algorithm) الكود هو مجرد وسيلة تعبير، بينما الخوارزمية هي الفكرة والمنطق. إنها الجسر الفكري الذي يربط بين المشكلة الإنسانية والحل الآلي.

9.1.1 التأصيل التاريخي واللغوي

يعود أصل المصطلح إلى العالم المسلم محمد بن موسى الخوارزمي (القرن التاسع الميلادي)، الذي وضع في كتابه "الجبر والمقابلة" أسس حل المعادلات بخطوات منهجية. حُرّف اسمه باللاتينية إلى *Algoritmi*، ليصبح لاحقاً *Algorithm*. أكاديمياً، الخوارزمية ليست مجرد وصفة، بل هي إجراء حسابي محدد جيداً يأخذ قيمة أو مجموعة قيم كمدخلات (Input) وينتج قيمة أو مجموعة قيم كمخرجات (Output).

9.1.2 الخصائص الخمس للخوارزمية الصحيحة (Knuth's 5 Properties)

حدد دونالد كانوث (أب علم تحليل الخوارزميات) خمسة شروط لكي تسمى سلسلة الخطوات خوارزمية:

1. **المحدودية (Finiteness):** يجب أن تنتهي الخوارزمية بعد عدد محدد من الخطوات. (الإجراء الذي لا ينتهي يسمى "طريقة حسابية" وليس خوارزمية).
2. **التحديد (Definiteness):** كل خطوة يجب أن تكون واضحة تماماً ولا تحتمل التأويل (لا غموض).
3. **المدخلات (Input):** للخوارزمية صفر أو أكثر من المدخلات.
4. **المخرجات (Output):** للخوارزمية مخرج واحد أو أكثر له علاقة بالمدخلات.
5. **الفاعلية (Effectiveness):** يجب أن تكون الخطوات بسيطة بما يكفي لتنفيذها نظرياً بواسطة إنسان يستخدم الورقة والقلم في زمن محدود.

9.1.3 الحتمية مقابل العشوائية (Determinism vs. Non-determinism)

- الخوارزمية الحتمية: إذا أعطيت نفس المدخلات، ستمر دائماً بنفس تسلسل الخطوات وتعطي نفس النتيجة (مثل العمليات الحسابية).
- الخوارزمية العشوائية: تستخدم أرقاماً عشوائية كجزء من منطقتها، وقد تعطي نتائج مختلفة أو تسلك مسارات مختلفة في كل مرة (مثل خوارزميات التشفير أو محاكاة مونت كارلو).

9.2 مفاهيم البرمجة الأساسية: البنية التحتية للكود

البرمجة هي فن التخاطب مع الذاكرة والمعالج. لفهمها، يجب أن نفهم كيف يتم تمثيل البيانات والتحكم فيها.

1. المتغيرات وإدارة الذاكرة (Variables & Memory Management)

المتغير ليس مجرد "صندوق"، بل هو اسم رمزي (Symbolic Name) يرتبط بموقع في ذاكرة الحاسوب (Address).

- **التعيين (Assignment):** عندما نقول $x = 5$ ، نحن نطلب من نظام التشغيل حجز مساحة في الذاكرة لتخزين الرقم 5، وربط الاسم x بعنوان تلك المساحة.
- **نطاق المتغير (Scope):** يحدد أين يمكن رؤية المتغير واستخدامه.
 - **نطاق محلي (Local):** متغير معرف داخل دالة، يختفي بمجرد انتهاء الدالة (يُحفظ في الـ Stack).
 - **نطاق عام (Global):** متغير متاح لكل البرنامج (يُحفظ غالباً في الـ Heap أو Data Segment).

2. أنواع البيانات (Data Types) ونظم النوع (Type Systems)

الحاسوب يتعامل مع الأعداد والأحاد فقط. "نوع البيانات" هو القناع الذي يخبر المعالج كيف يفسر هذه الأعداد والأحاد.

- **أنواع بدائية (Primitive Types):**
 - **Integer:** أعداد صحيحة. يتم تمثيلها بنظام المكمل الثنائي. (Two's Complement).

- **Float (IEEE 754)** أعداد عشرية. تتكون من (إشارة، أس، وكسر). دقتها محدودة وقد تسبب أخطاء تقريب.
- **Boolean** بت واحد منطقياً (0 أو 1)، لكنه غالباً يخزن في بايت كامل لسهولة العنوان.
- **Char** حرف واحد (ASCII) أو (Unicode).
- أنواع مركبة: **(Composite Types)** المصفوفات، السلاسل النصية، والكائنات.
- **التمييز: (Typing)**
 - *Static Typing* مثل *C++*, *Java*: يجب تحديد نوع المتغير قبل التشغيل. يساعد في اكتشاف الأخطاء مبكراً.
 - *Dynamic Typing* مثل *Python*: يتم استنتاج النوع أثناء التشغيل. يوفر مرونة عالية ولكن قد يخفي أخطاء.

3. العبارات الشرطية والمنطق البوليني

القدرة على "اتخاذ القرار" تعتمد على البوابات المنطقية.

- **Short-circuit Evaluation**: في عبارة $if(A \text{ and } B)$ ، إذا كانت A خاطئة، فإن الحاسوب لا يكلف نفسه عناء فحص B، لأن النتيجة محسومة سلفاً. هذا مهم للأداء ولتجنب أخطاء (مثل القسمة على صفر).

4. الحلقات التكرارية (Iterative Structures)

- **حلقة العداد (For Loop)**: مثالية عندما نعرف عدد التكرارات مسبقاً. تستخدم بكثرة في معالجة المصفوفات.
- **حلقة الشرط (While Loop)**: مثالية عندما لا نعرف متى نتوقف (مثل: "انتظر حتى يضغط المستخدم زر الخروج").
- **الحلقات اللانهائية (Infinite Loops)**: خطأ قاتل يستهلك موارد المعالج، إلا في حالات خاصة (مثل الحلقة الرئيسية لنظام التشغيل أو خادم الويب).

9.3 التصميم الهندسي: الكود الزائف ومخططات التدفق

قبل البناء، نحتاج إلى مخطط أزرق. (Blueprint)

الكود الزائف (Pseudo-code)

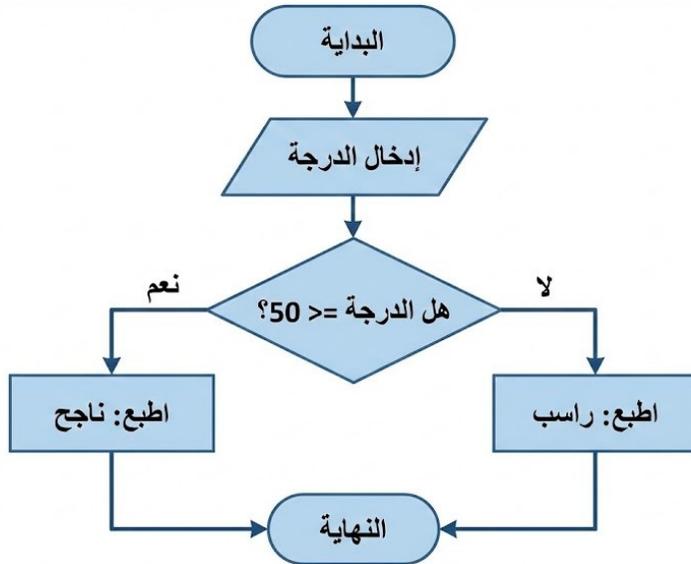
لغة بسيطة بين لغة البشر ولغة البرمجة. يركز على المنطق ويهمل قواعد النحو. (Syntax)

- الفائدة: يسمح للمبرمجين الذين يستخدمون لغات مختلفة (Python, C++, Java) بفهم نفس الخوارزمية.

مخططات التدفق (Flowcharts)

هي المعيار البصري لتمثيل الخوارزميات. (ISO 5807)

تمثيل بصري لخوارزمية اتخاذ قرار (مثال: قبول طالب):



9.4 لغة Python: سكين الجيش السويسري لعلوم البيانات

لماذا هيمنت بايثون على مجال الذكاء الاصطناعي؟

1. لغة مفسرة: **(Interpreted)** يتم تنفيذ الكود سطرًا تلو الآخر، مما يسهل التجريب وتصحيح الأخطاء. (Rapid Prototyping)
 2. إدارة الذاكرة التلقائية: **(Garbage Collection)** لا يحتاج المبرمج لتحرير الذاكرة يدويًا كما في C++ ، مما يقلل من أخطاء تسرب الذاكرة. (Memory Leaks)
 3. نظام الأنواع الديناميكي القوي: **(Strong Dynamic Typing)** لا تحتاج لتعريف الأنواع، لكن اللغة لا تسمح بجمع نص مع رقم (Strong) ، مما يوفر توازنًا بين الأمان والمرونة.
- مثال متقدم: دالة مع معالجة الأخطاء

```
import math

def calculate_circle_area(radius):
    """
    دالة لحساب مساحة الدائرة مع التحقق من المدخلات
    """
    try:
        # التحقق من النوع
        r = float(radius)
        if r < 0:
            raise ValueError("نصف القطر لا يمكن أن يكون سالباً")

        area = math.pi * (r ** 2)
        return round(area, 2)
```

```
except ValueError as e:
```

```
    return f"خطأ في المدخلات {e}"
```

```
# الاستخدام
```

```
user_input = input("أدخل نصف القطر: ")
```

```
result = calculate_circle_area(user_input)
```

```
print(f"النتيجة: {result}")
```

9.5 هياكل البيانات: تنظيم المعلومات (Data Structures)

قال لينوس تورفالدس (مبتكر Linux) المبرمجون السيئون يقلقون بشأن الكود. المبرمجون الجيدون يقلقون بشأن هياكل البيانات وعلاقاتها.

1. المصفوفات والقوائم (Arrays & Lists)

- **Array:** مساحة متصلة في الذاكرة. الوصول لأي عنصر يتم في زمن ثابت $O(1)$ لأننا نعرف العنوان بحساب رياضي بسيط. عيبها: الحجم الثابت) في لغات مثل C).
- **List (Python):** مصفوفة ديناميكية يمكن أن يتغير حجمها وتخزن أنواعاً مختلفة.

2. القائمة المترابطة (Linked List)

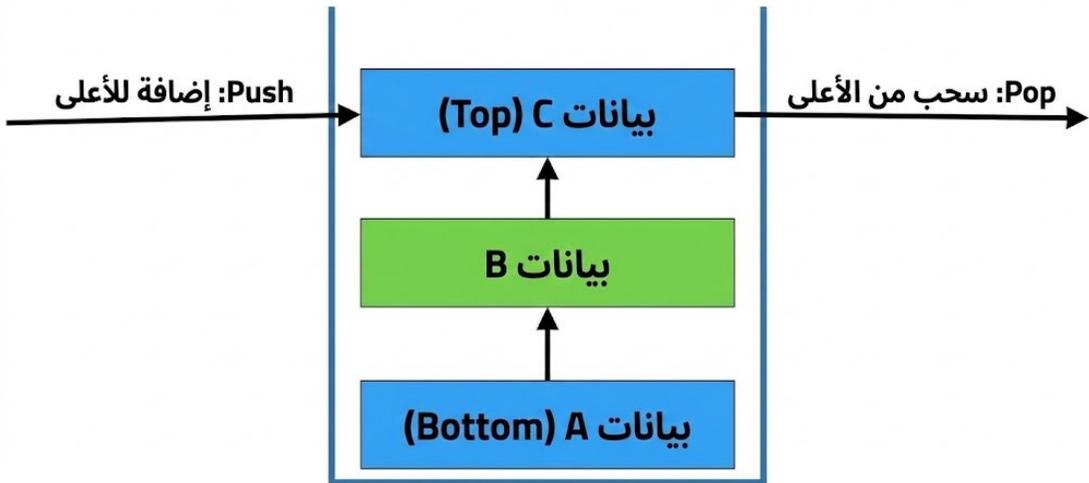
- سلسلة من العقد (Nodes) ، كل عقدة تحتوي على بيانات ومؤشر (Pointer) للعقدة التالية.
- **الميزة:** سهولة إضافة وحذف عناصر في المنتصف دون الحاجة لإزاحة باقي البيانات.
 - **العيب:** الوصول لعنصر يتطلب المرور على كل العناصر السابقة. $O(n)$.

3. المكس (Stack) - LIFO

- يعمل بمبدأ "آخر من يدخل، أول من يخرج".

- **التطبيقات:** إدارة استدعاء الدوال (Call Stack) في المعالج، التراجع (Undo) في المحررات، والتحقق من توازن الأقواس في الكود. { () }

Stack Operations



4. الطابور (Queue) - FIFO

- يعمل بمبدأ "أول من يدخل، أول من يخرج".
- **التطبيقات:** جدولة مهام الطباعة، إدارة طلبات خادم الويب، خوارزمية البحث بالعرض (BFS) في الشبكات.

5. جداول التجزئة (Hash Tables / Dictionaries)

هيكل البيانات الأهم في علوم البيانات.

- **الفكرة:** استخدام "دالة تجزئة (Hash Function)" لتحويل المفتاح (مثل اسم الطالب) إلى رقم يمثل فهرس تخزينه في الذاكرة.
- **الميزة:** الوصول للبيانات يتم بسرعة فائقة $O(1)$ في المتوسط، بغض النظر عن حجم البيانات.

- تطبيق: البحث عن كلمة في قاموس، قواعد البيانات المفتاحية (Key-Value Stores like Redis).

9.6 التفكير الخوارزمي (Computational Thinking)

هو عملية ذهنية لحل المشكلات بطريقة تمكن الحاسوب من تنفيذ الحل.

1. التحليل (Decomposition): كسر المشكلة.

- مثال: تطبيق "Uber" ليس كتلة واحدة، بل هو: (نظام خرائط + نظام دفع + نظام تتبع سائقين + نظام تقييم).

2. التعرف على الأنماط: (Pattern Recognition)

- مثال: ملاحظة أن حركة المرور تزداد دائماً بين 7-9 صباحاً، واستخدام هذا النمط لتعديل تسعير الرحلة.

3. التجريد: (Abstraction)

- إخفاء التعقيد. عند استخدام دالة $\text{sort}()$ ، لا يهتمك كيف تقوم بالترتيب (Bubble sort) أو (Quick sort)، يهتمك فقط أنها ترتب القائمة.

4. تصميم الخوارزمية: (Algorithmic Design)

- كتابة الخطوات الفعلية للحل.

9.7 تحليل تعقيد الخوارزميات (Algorithm Complexity Analysis)

كيف نعرف أن خوارزمية "أفضل" من أخرى؟ ليس بقياس الزمن بالثواني (لأنه يعتمد على سرعة الجهاز)، بل بقياس معدل نمو الخطوات بالنسبة لحجم المدخلات. ($\$n\$$) نستخدم لذلك ترميز **Big-O**.

أشهر رتب التعقيد الزمني: (Time Complexity)

1. $O(1)$ - زمن ثابت (Constant): الوقت لا يتغير مهما زادت البيانات.
 - مثال: الوصول لعنصر في مصفوفة عبر الفهرس.
2. $O(\log n)$ - زمن لوغاريتمي (Logarithmic): الوقت يزيد ببطء شديد.

○ مثال: البحث الثنائي (Binary Search) في دليل هاتف مرتب. البحث في مليون اسم يتطلب فقط 20 خطوة!

3. $O(n)$ - زمن خطي (Linear) الوقت يزيد بنفس نسبة البيانات.

○ مثال: قراءة كتاب صفحة بصفحة، البحث في قائمة غير مرتبة.

4. $O(n \log n)$ - زمن شبه خطي:

○ مثال: أفضل خوارزميات الترتيب (Merge Sort, Quick Sort).

5. $O(n^2)$ - زمن تربيعي (Quadratic) الوقت يتضاعف بشكل مربع.

○ مثال: حلقات متداخلة (Nested Loops)، خوارزمية ترتيب الفقاعة (Bubble Sort).
سينة للبيانات الكبيرة.

6. $O(2^n)$ - زمن أسّي (Exponential) كارثي. كل عنصر إضافي يضاعف الوقت.

○ مثال: كسر كلمات المرور بالتجربة الشاملة (Brute Force)، مشكلة البائع المتجول.

xychart-beta

" title مقارنة معدل نمو الوقت بالنسبة لحجم البيانات"

x-axis [10, 100, 1000]

y-axis 10000 --> عدد العمليات "0"

line [1, 1, 1] text "O(1)"

line [10, 100, 1000] text "O(n)"

line [100, 10000, 1000000] text "O(n^2)"

9.8 أنماط البرمجة (Programming Paradigms)

تختلف لغات البرمجة في "فلسفتها" وطريقة تنظيم الكود.

1. البرمجة الإجرائية: (Procedural Programming)

- البرنامج هو سلسلة من التعليمات والدوال.
- أمثلة (C, Pascal, Python): يمكن استخدامها كإجرائية.
- مناسبة لـ: البرامج البسيطة، السكريبتات، برمجة النظم.

2. البرمجة كائنية التوجه: (Object-Oriented Programming - OOP)

- البرنامج هو مجموعة من "الكائنات (Objects)" تتفاعل مع بعضها. كل كائن يحتوي ببياناته (Attributes) ووظائفه (Methods).
- المفاهيم الأربعة: التغليف (Encapsulation)، الوراثة (Inheritance)، تعدد الأشكال (Polymorphism)، التجريد (Abstraction).
- أمثلة: Java, Python, C++.
- مناسبة لـ: الأنظمة الكبيرة والمعقدة، الألعاب، واجهات المستخدم.

3. البرمجة الوظيفية: (Functional Programming)

- تعامل البرنامج كمعادلات رياضية بحتة. تتجنب تغيير حالة البيانات (Immutability).
- أمثلة (Haskell, Scala): وميزات داخل Python وJavaScript.
- مناسبة لـ: الأنظمة الموزعة، المعالجة المتوازية، الأنظمة المالية التي تتطلب دقة عالية.

الفصل العاشر

المنطق الرياضي ونظرية المجموعات

10.1 الجمل الرياضية والقضايا المنطقية (Propositions)

المنطق الرياضي هو اللغة الرسمية للتفكير الدقيق. كل شرط iff يكتبه، وكل استعلام قاعدة بيانات، يخضع لهذه القوانين.

القضية (Proposition)

هي جملة خبرية يمكن الحكم عليها بأنها صائبة (T) أو خاطئة (F). نرمز لها بمتغيرات مثل p, q .

$$v(p) = T \text{ (صائبة).}$$

$$v(p) = F \text{ (خاطئة).}$$

10.2 الروابط المنطقية وجداول الصدق

1. النفي $\neg p$ (Negation)

p	$\neg p$
T	F
F	T

2. الوصل $p \wedge q$ (AND)

تكون صائبة فقط إذا كان كلاهما صائباً.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

3. الفصل $p \vee q$ (OR)

تكون صائبة إذا تحقق شرط واحد على الأقل.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

4. الاستلزام الشرطي $p \rightarrow q$ (Implication)

القاعدة: الاستلزام خاطئ فقط عندما تكون المقدمة صحيحة والنتيجة خاطئة.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

10.3 قوانين المنطق (Logical Laws)

1. قوانين دي مورغان:

(q نفي) أو (p نفي) يكافئ (p و q) نفي

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

2. قانون التضاد:

$$(p \rightarrow q) \equiv (\neg q \rightarrow \neg p)$$

10.4 نظرية المجموعات (Set Theory)

المجموعة هي تجمع من العناصر المتمايضة.

- الاتحاد (Union): العناصر الموجودة في A أو B.
- التقاطع (Intersection): العناصر المشتركة فقط.
- الجداء الديكارتي: مجموعة الأزواج المرتبة (أساس جداول SQL).

10.5 العلاقات والدوال

أنواع الدوال:

1. دالة متباينة (One-to-One): لا يوجد عنصران لهما نفس الصورة (مثل رقم الهوية).
2. دالة شاملة (Onto): تغطي كامل المجال المقابل.
3. دالة تناظر (Bijective): متباينة وشاملة (لها دالة عكسية).

الفصل الحادي عشر

الرياضيات المتقطعة وهياكل البيانات الرياضية

11.1 المتتاليات والمتسلسلات البسيطة (Sequences and Series)

في تحليل الخوارزميات، غالباً ما نحتاج لعدّد الخطوات التي ينفذها البرنامج، وهنا تأتي أهمية المتتاليات.

المتتالية: (Sequence) هي دالة مجالها مجموعة الأعداد الصحيحة الموجبة. هي قائمة مرتبة من الأعداد.

• المتتالية الحسابية: (Arithmetic Progression) الفرق بين كل حد وتاليه ثابت (مثال: 1، 3، 5، 7).

• المتتالية الهندسية: (Geometric Progression) النسبة بين كل حد وتاليه ثابتة (مثال: 2، 4، 8، 16). هذا النمط يظهر بوضوح في حجم الذاكرة (RAM) وفي الأشجار الثنائية.

المتسلسلة: (Series) هي مجموع حدود المتتالية.

• أهمية خاصة: متسلسلة الجمع $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ تخبرنا بعدد المرات التي تتنفذ فيها حلقة تكرارية متداخلة (Nested Loop)، وهو أساس حساب التعقيد الزمني $O(n^2)$.

11.2 التوافيق والتباديل (Combinatorics)

علم العدّ (Counting) هو الركيزة الأساسية لحساب الاحتمالات وتقدير مساحة البحث في مشاكل الذكاء الاصطناعي.

مبدأ الضرب (Rule of Product)

إذا كانت هناك m طريقة لعمل شيء، و n طريقة لعمل شيء آخر، فإن هناك $m \times n$ طريقة لعملهما معاً.

• تطبيق: حساب عدد كلمات المرور الممكنة. إذا كانت كلمة المرور تتكون من 4 أرقام، فعدد

$$10 \times 10 \times 10 \times 10 = 10,000$$
 الاحتمالات هو

التباديل (Permutations)

عدد طرق ترتيب مجموعة من العناصر المتميزة حيث الترتيب مهم.

• القانون: $P(n, r) = \frac{n!}{(n-r)!}$

• مثال: ترتيب وصول حزم البيانات في الشبكة.

التوافيق (Combinations)

عدد طرق اختيار مجموعة فرعية من العناصر حيث الترتيب غير مهم.

• القانون: $C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$

• مثال: اختيار 3 ميزات (Features) من أصل 10 لتدريب نموذج تعلم آلي مبسط.

11.3 الرسوم البيانية (Graphs) والأشجار (Trees)

نظرية المخططات (Graph Theory) هي اللغة التي نستخدمها لنمذجة العلاقات المعقدة، من الشبكات الاجتماعية إلى شبكات الطرق.

الرسوم البيانية (Graphs)

يتكون المخطط $G = (V, E)$ من مجموعة من الرؤوس (Vertices) ومجموعة من الحواف (Edges) التي تربط بينها.

• مخطط غير موجه (Undirected): العلاقة ثنائية الاتجاه (مثل الصداقة على فيسبوك).

• مخطط موجه (Directed): العلاقة أحادية الاتجاه (مثل المتابعة على تويتر).

• مخطط موزون (Weighted): كل حافة لها قيمة (مثل المسافة بين مدينتين في خرائط جوجل).

الأشجار (Trees)

هي نوع خاص من المخططات متصلة ولا تحتوي على دورات (Cycles).

- الشجرة الثنائية (Binary Tree): كل أب له ابنان على الأكثر.
- أهمية الأشجار: تُستخدم في هيكلية ملفات النظام (File System Hierarchy) وفي خوارزميات البحث السريع، وهي الأساس لخوارزميات "الغابة العشوائية (Random Forest)" في تعلم الآلة.

11.4 مبادئ تعقيد الخوارزميات (Big-O Notation)

كيف نقيس كفاءة الخوارزمية؟ ليس بالثواني (لأن الأجهزة تختلف)، بل بمعدل نمو الوقت بالنسبة لحجم المدخلات n .

ترميز (Big-O الحد الأعلى للنمو):

1. $O(1)$ - ثابت: الوقت لا يتغير مهما زاد حجم البيانات (مثال: الوصول لعنصر في مصفوفة عبر الفهرس).
2. $O(\log n)$ - لوغاريتمي: الوقت يزيد ببطء شديد (مثال: البحث الثنائي Binary Search). هذه هي الكفاءة المثالية للبحث في البيانات الضخمة.
3. $O(n)$ - خطي: الوقت يزيد بنسبة طردية (مثال: البحث عن اسم في قائمة غير مرتبة).
4. $O(n^2)$ - تربيعي: الوقت يتضاعف بشكل مربع (مثال: خوارزميات الترتيب البسيطة مثل Bubble Sort غير مقبولة للبيانات الكبيرة).
5. $O(2^n)$ - أسّي: الوقت يتضاعف مع كل عنصر إضافي. هذه خوارزميات غير عملية إلا للمدخلات الصغيرة جداً.

11.5 تطبيقات الرياضيات المتقطعة في الشبكات والخوارزميات

1. أقصر مسار (Shortest Path): خوارزمية دايكسترا (Dijkstra) تستخدم نظرية المخططات الموزونة لتحديد أسرع طريق لتوصيل حزمة بيانات (أو سيارة توصيل طلبات).
2. التشفير (Cryptography): يعتمد تشفير RSA على نظريات الأعداد الأولية (Prime Numbers) وبقايا القسمة (Modular Arithmetic)، وهي فروع أصيلة في الرياضيات المتقطعة.

3. الشبكات الاجتماعية: تحليل "المركزية (Centrality) "في المخطط يساعد في معرفة الشخصيات الأكثر تأثيراً في شبكة اجتماعية.

الفصل الثاني عشر

الجبر الخطي والتحليل المتجهي

12.1 المتجهات (Vectors) في بعدين وثلاثة أبعاد

في علوم الحاسوب، لا ننظر للمتجه كسهم هندسي فقط، بل كقائمة مرتبة من الأرقام تمثل "نقطة بيانات" (Data Point) في فضاء متعدد الأبعاد.

التعريف: المتجه v هو كيان له مقدار (Magnitude) واتجاه (Direction).

• في البعدين \mathbb{R}^2 : $\mathbf{v} = [x, y]$

• في n أبعاد \mathbb{R}^n : $\mathbf{v} = [x_1, x_2, \dots, x_n]$

العمليات الأساسية:

1. الجمع: يتم بجمع العناصر المتقابلة. هندسياً، يمثل قاعدة المثلث.

2. الضرب القياسي (Dot Product): $u \cdot v = \sum u_i v_i$

○ أهمية القصوى: إذا كان حاصل الضرب القياسي صفراً، فإن المتجهين متعامدان (Orthogonal)، أي مستقلان تماماً. في الذكاء الاصطناعي، يُستخدم الضرب القياسي لحساب التشابه Cosine Similarity بين النصوص أو الصور.

12.2 المحددات (Determinants) والرتبة (Rank)

المحدد (Determinant)

قيمة عددية وحيدة للمصفوفة المربعة، يرمز لها بـ $|A|$ أو $\det(A)$.

• التفسير الهندسي: في البعدين، يمثل المحدد مساحة متوازي الأضلاع الذي يشكله عمودا المصفوفة. إذا كان المحدد صفراً، فهذا يعني أن المساحة انعدمت، وأن المصفوفة "تسحق" الفضاء إلى بُعد أقل، وبالتالي لا يوجد لها معكوس (Singular Matrix).

الرتبة (Rank)

هي عدد الصفوف (أو الأعمدة) المستقلة خطياً في المصفوفة.

- معنى الاستقلال الخطي: لا يمكن تكوين صف من خلال جمع أو ضرب الصفوف الأخرى. الرتبة تخبرنا بكمية "المعلومات الفريدة" الموجودة في المصفوفة.

12.3 أنظمة المعادلات الخطية وطريقة غاوس

حل جمل المعادلات الخطية $Ax = b$ هو المشكلة الأكثر تكراراً في الحسابات العلمية.

الحذف الغاوسي (Gaussian Elimination): خوارزمية منهجية لتحويل المصفوفة إلى الشكل الدرجي الصفي (Row Echelon Form)، مما يسهل حل المعادلات بالتعويض الخلفي.

- التطبيق: تستخدم في محاكاة الدوائر الكهربائية، التنبؤ الجوي، وموازنة المعادلات الكيميائية.

12.4 القيم والمتجهات الذاتية – (Eigenvalues & Eigenvectors)

هذا المفهوم هو جوهر خوارزميات ضغط البيانات وتحليلها.

عندما نضرب مصفوفة A في متجه v ، فإن المتجه الناتج عادة يغير اتجاهه ومقداره. لكن، هناك متجهات خاصة لا تغير اتجاهها، بل يتم فقط "تمطيها" أو "تقليصها".

- المتجه الذاتي v : هو ذلك المتجه الذي يحافظ على اتجاهه بعد التحويل الخطي.
- القيمة الذاتية λ : هي عامل القياس (Scaling Factor) الذي تمدد أو تقلص به المتجه.
- المعادلة $Av = \lambda v$
- تطبيق: خوارزمية PageRank من جوجل تعتمد على القيم الذاتية لترتيب صفحات الويب حسب الأهمية.

12.5 تطبيقات الجبر الخطي في الرسوميات الحاسوبية والذكاء الاصطناعي

1. الرسوميات الحاسوبية (Computer Graphics):

- كل حركة في لعبة فيديو ثلاثية الأبعاد (تدوير، تكبير، نقل) هي في الحقيقة عملية ضرب مصفوفة تحويل (Transformation Matrix) في إحداثيات نقاط الجسم.

2. تحليل المكونات الرئيسية:(PCA)

- تقنية لتقليل أبعاد البيانات (Dimensionality Reduction) تعتمد كلياً على القيم الذاتية والمتجهات الذاتية. تستخدم لضغط البيانات مع الحفاظ على أهم المعلومات (التباين).

3. تضمين الكلمات:(Word Embeddings)

- في معالجة اللغة الطبيعية(NLP) ، يتم تحويل الكلمات إلى متجهات في فضاء عالي الأبعاد، بحيث تكون الكلمات ذات المعنى المتقارب (مثل "ملك" و "أمير") قريبة من بعضها رياضياً (زاوية صغيرة بين المتجهين).

الفصل الثالث عشر

حساب التفاضل والتكامل لعلوم الحاسوب

13.1 الدوال الحقيقية ورسم المنحنيات (Real Functions)

في البرمجة، الدالة هي كود يستقبل مدخلات ويعيد مخرجات. في الرياضيات، المفهوم مشابه ولكنه أكثر تجريداً: الدالة $f(x)$ هي علاقة تربط كل عنصر x في المجال بعنصر وحيد y في المجال المقابل.

أهمية سلوك الدوال في علوم البيانات:

قبل تطبيق أي خوارزمية، يجب فهم "شكل" البيانات.

- الدوال الخطية: $f(x) = ax + b$ أساس الانحدار الخطي. (Linear Regression)
- الدوال الأسية: $f(x) = e^{ax}$ تصف النمو السريع (مثل انتشار الفيروسات أو تفاعل المنشورات).
- الدوال اللوجستية: (Sigmoid/Logistic) حيوية في الشبكات العصبية لأنها تحول أي قيمة مدخلة إلى رقم بين 0 و 1 (احتمال)، وتستخدم كدالة تنشيط. (Activation Function)

13.2 النهايات والاستمرارية (Limits and Continuity)

النهايات (Limits)

المفهوم الأساسي الذي يميز التفاضل والتكامل عن الجبر. النهاية $\lim_{x \rightarrow c} f(x) = L$ تعني دراسة سلوك الدالة عندما "يقترب" المتغير x من القيمة c دون أن يساويها بالضرورة.

- تطبيق: تعريف المشتقة يعتمد كلياً على النهايات. $(\lim_{h \rightarrow 0})$

الاستمرارية (Continuity)

الدالة المستمرة هي التي يمكن رسم منحناها دون رفع القلم عن الورق.

- الأهمية الحاسوبية: معظم خوارزميات التحسين (Optimization Algorithms) تتطلب أن تكون دالة الخطأ (Loss Function) مستمرة وقابلة للاشتقاق. الانقطاعات المفاجئة في البيانات قد تسبب فشل النموذج في التعلم.

13.3 التفاضل: القواعد والتطبيقات

التفاضل (Differentiation) هو دراسة معدل التغير. المشتقة $f'(x)$ تخبرنا بمدى حساسية المخرجات للتغيرات الطفيفة في المدخلات.

التفسير الهندسي: المشتقة هي ميل المماس (Slope of the tangent) لمنحنى الدالة عند نقطة معينة.

المشتقات الجزئية: (Partial Derivatives)

في الذكاء الاصطناعي، نتعامل مع دوال متعددة المتغيرات $f(x, y, z, \dots)$ (حيث المتغيرات هي أوزان الشبكة العصبية). المشتقة الجزئية $\partial f / \partial x$ تقيس معدل التغير بالنسبة لمتغير واحد فقط مع تثبيت البقية.

- التدرج: (Gradient ∇f) هو متجه يحتوي على جميع المشتقات الجزئية. يشير التدرج دائماً إلى اتجاه الزيادة القصوى للدالة.

13.4 التكامل: المفهوم الهندسي والتقنيات الأساسية

التكامل (Integration) هو العملية العكسية للتفاضل، ويمثل التراكم. (Accumulation)

التفسير الهندسي: التكامل المحدود $\int_a^b f(x) dx$ يحسب المساحة المحصورة تحت منحنى الدالة بين النقطتين a و b .

التطبيقات في الاحتمالات:

في المتغيرات العشوائية المتصلة، الاحتمال ليس نقطة، بل هو مساحة. لحساب احتمال وقوع حدث ما (مثلاً أن يكون طول الشخص بين 170 و 180 سم)، نقوم بحساب التكامل لدالة الكثافة الاحتمالية (PDF) في تلك الفترة.

13.5 تطبيقات التفاضل والتكامل في تحسين الأداء (Optimization)

هنا يكمن الجوهر الرياضي لتدريب الذكاء الاصطناعي.

هدف التعلم الآلي:

هو العثور على قيم المعاملات (Parameters) التي تجعل نسبة الخطأ في التنبؤ أقل ما يمكن. رياضياً، هذا يعني إيجاد النهاية الصغرى (Global Minimum) لدالة التكلفة. (Cost Function)

خوارزمية الانحدار التدرجي: (Gradient Descent)

1. نبدأ بقيم عشوائية للأوزان.
 2. نحسب التدرج (المشتقة) لدالة الخطأ عند تلك النقطة.
 3. نتحرك في الاتجاه المعاكس للتدرج (لأننا نريد النزول لأسفل الوادي، أي تقليل الخطأ).
 4. نكرر العملية حتى نصل إلى القاع (أقل خطأ ممكن).
- بدون التفاضل، لن نتمكن من معرفة "الاتجاه" الصحيح لتحسين النموذج، وسنضطر للتخمين العشوائي، وهو أمر مستحيل في الفضاءات عالية الأبعاد.

الفصل الرابع عشر

الاحتمالات والإحصاء وتحليل البيانات

14.1 مفاهيم أساسية في الاحتمال (Probability Concepts)

بينما يتعامل المنطق الرياضي (الذي ناقشناه سابقاً) مع اليقين (صواب/خطأ)، تتعامل نظرية الاحتمالات مع عدم اليقين (Uncertainty) في العالم الحقيقي، البيانات مشوشة (Noisy) وناقصة، وهنا تكمن قوة النموذج الاحتمالي.

الفضاء العيني (Sample Space - Ω) هو مجموعة جميع النتائج الممكنة لتجربة ما.

• مثال: عند رمي حجر نرد، $\Omega = \{1, 2, 3, 4, 5, 6\}$.

الحدث (Event): هو مجموعة فرعية من الفضاء العيني.

• مثال: حدث الحصول على رقم زوجي $E = \{2, 4, 6\}$. احتمال هذا الحدث

$$P(E) = \frac{|E|}{|\Omega|} = \frac{3}{6} = P(E) = \frac{|E|}{|\Omega|} = \frac{3}{6} = 0.5$$

الاحتمال الشرطي (Conditional Probability): هو احتمال وقوع حدث A بشرط أن الحدث B قد وقع بالفعل، ويرمز له بـ $P(A|B)$.

$$\bullet \text{ القانون } P(A|B) = \frac{P(A \cap B)}{P(B)}$$

• الأهمية القصوى: هذا المفهوم هو أساس نظرية بايز (Bayes' Theorem)، التي تعد الركيزة الأساسية لخوارزميات التصنيف (مثل Naive Bayes) وتحديث المعتقدات بناءً على أدلة جديدة في الذكاء الاصطناعي.

14.2 المتغيرات العشوائية والتوزيعات الشهيرة

المتغير العشوائي (Random Variable) هو دالة تحول نتائج التجربة العشوائية إلى أرقام حقيقية. (مثلاً: دالة تعطي القيمة 1 إذا كانت العملة "صورة" و 0 إذا كانت "كتابة").

1. المتغيرات المتقطعة (Discrete)

تأخذ قيماً منفصلة ومعدودة.

- توزيع برنولي (Bernoulli Distribution): لتجربة لها نتيجتان فقط (نجاح/فشل).
- التوزيع ذو الحدين (Binomial Distribution): لعدد النجاحات في n من المحاولات المستقلة.

2. المتغيرات المتصلة (Continuous)

تأخذ أي قيمة داخل فترة زمنية أو مكانية.

- التوزيع الطبيعي (Normal/Gaussian Distribution):
 - هو "ملك" التوزيعات الإحصائية. يتخذ شكل الجرس. (Bell Curve)
 - الأهمية: حسب مبرهنة النهاية المركزية (Central Limit Theorem)، فإن مجموع عدد كبير من المتغيرات العشوائية المستقلة يميل للتوزيع الطبيعي، مهما كان توزيعها الأصلي. هذا يفسر لماذا تتبع أطوال البشر، وأخطاء القياس، ودرجات الذكاء هذا التوزيع.
 - يتميز بمعاملين: المتوسط μ (مركز القمة) والانحراف المعياري σ (عرض الجرس).

14.3 المقاييس الإحصائية (Statistical Measures)

لوصف البيانات الضخمة، نحتاج لاختزالها في أرقام معبرة. تنقسم هذه المقاييس إلى نوعين:

أولاً: مقاييس النزعة المركزية (Central Tendency)

تخبرنا أين تتمركز البيانات.

1. المتوسط الحسابي (μ - Mean) مجموع القيم مقسوماً على عددها. حساس جداً للقيم الشاذة (Outliers).

2. الوسيط (Median): القيمة التي تتوسط البيانات بعد ترتيبها. هو المقياس الأفضل لرواتب الموظفين أو أسعار المنازل لأنه لا يتأثر بالقيم المتطرفة.

3. المنوال (Mode): القيمة الأكثر تكراراً.

ثانياً: مقياس التشتت (Dispersion)

تخبرنا بمدى انتشار البيانات حول المركز.

1. التباين: (σ^2 - Variance) متوسط مربعات انحراف القيم عن المتوسط.

2. الانحراف المعياري: (σ - Standard Deviation) الجذر التربيعي للتباين. يقاس بنفس وحدة البيانات الأصلية.

○ في علوم البيانات: البيانات ذات الانحراف المعياري العالي تعني تشتتاً كبيراً وعدم يقين، بينما الانحراف المنخفض يعني دقة واستقراراً.

14.4 الاستدلال الإحصائي (Statistical Inference)

علم الإحصاء ينقسم إلى "وصفي" (ما ذكرناه أعلاه) و"استدلالي". الإحصاء الاستدلالي يهدف لتعميم النتائج من عينة (Sample) صغيرة على مجتمع (Population) كامل.

فترات الثقة (Confidence Intervals)

بدلاً من القول "متوسط طول السكان هو 170 سم"، نقول "نحن واثقون بنسبة 95% أن المتوسط يقع بين 168 و 172 سم". هذا يعطي مجالاً للخطأ المحتمل.

اختبار الفرضيات (Hypothesis Testing)

منهجية لاتخاذ القرار: هل النتيجة التي حصلنا عليها حقيقية أم مجرد صدفة؟

1. الفرضية الصفرية: (H_0) نفترض عدم وجود تأثير أو فرق (مثلاً: الدواء الجديد لا يختلف عن القديم).

2. القيمة الاحتمالية: (P-value) هي احتمال رؤية النتائج الحالية إذا كانت الفرضية الصفرية صحيحة.

- إذا كانت P-value صغيرة جداً ($0.05 <$) ، نرفض الفرضية الصفرية ونقول إن النتيجة "ذات دلالة إحصائية". (Statistically Significant)

14.5 تطبيقات الاحتمالات والإحصاء في أمن المعلومات والذكاء الاصطناعي

1. اكتشاف الشذوذ: (Anomaly Detection)

- في الأمن السيبراني، يتم بناء "نموذج احتمالي" لسلوك المستخدم العادي (أوقات الدخول، حجم البيانات المنقولة). إذا قام المستخدم بنشاط احتماله ضئيل جداً (مثلاً $P < 0.001$) ، يطلق النظام إنذاراً باحتمالية وجود اختراق.

2. التصنيف البايزي: (Naive Bayes Classifier)

- خوارزمية بسيطة وفعالة جداً تستخدم نظرية بايز لتصنيف النصوص (مثل فلترة الرسائل المزعجة. Spam) تحسب احتمال أن تكون الرسالة Spam بناءً على وجود كلمات معينة فيها.

3. النمذجة التوليدية: (Generative Models)

- نماذج الذكاء الاصطناعي الحديثة (مثل التي تولد الصور) تعتمد على تعلم التوزيع الاحتمالي للبيانات، ثم أخذ عينات عشوائية من هذا التوزيع لإنتاج صور جديدة لم تكن موجودة من قبل.

الفصل الخامس عشر

الرياضيات التطبيقية في الذكاء الاصطناعي وعلوم البيانات

15.1 نظرة عامة على نماذج التعلم الآلي (Machine Learning Models)

التعلم الآلي ليس سحراً، بل هو نمذجة رياضية (Mathematical Modeling) لظواهر طبيعية. الهدف هو بناء دالة رياضية $f(x)$ يمكنها تقريب العلاقة بين المدخلات والمخرجات بأقل نسبة خطأ ممكنة.

رياضياً، تنقسم النماذج إلى فئتين رئيسيتين بناءً على طبيعة المخرجات:

1. نماذج التصنيف (Classification): حيث المخرج هو فئة متقطعة. (Discrete Class)

○ مثال: هل الرسالة "بريد مزعج" أم "عادي"؟ (0 أو 1). هنا نستخدم الاحتمالات والمنطق الفاصل.

2. نماذج الانحدار (Regression): حيث المخرج هو قيمة متصلة. (Continuous Value)

○ مثال: التنبؤ بسعر منزل بناءً على مساحته. هنا نستخدم الجبر الخطي والتفاضل لرسم "أفضل خط" يمر بالبيانات.

15.2 دور الجبر الخطي في الشبكات العصبية (Neural Networks)

الشبكة العصبية، مهما تعقدت، هي في جوهرها سلسلة ضخمة من عمليات ضرب المصفوفات.

التشريح الرياضي للطبقة الواحدة: لنفترض أن لدينا طبقة من الخلايا العصبية. العملية التي تحدث داخلها هي:

$$Z = W \cdot X + b$$

حيث:

- X : متجه المدخلات (بيانات الصورة مثلاً).
- W : مصفوفة الأوزان (Weights Matrix) التي تمثل قوة الروابط بين الخلايا. (هذا ما "تتعلمه" الشبكة).

• b : متجه الانحياز. (Bias Vector)

• Z : الناتج الخطي.

بعد ذلك، يتم تمرير Z عبر دالة تنشيط غير خطية (مثل ReLU أو Sigmoid) التي تحدثنا عنها في فصل التفاضل، لتتمكن الشبكة من فهم البيانات المعقدة. بدون الجبر الخطي، كنا سنضطر لحساب كل خلية عصبية بمعادلة منفصلة، مما يجعل التدريب مستحيلاً حسابياً. المصفوفات تسمح لنا بحساب ملايين الخلايا في عملية واحدة متوازية (Vectorization) على الـ GPU.

15.3 دور الاحتمالات والإحصاء في نماذج التنبؤ

الذكاء الاصطناعي لا يعطي إجابات مطلقة، بل احتمالية.

التنبؤ الاحتمالي: عندما يخبرك النموذج أن "هذه الصورة لقطة"، هو في الحقيقة يقول رياضياً:

$$P(\text{Cat}|\text{Image}) = 0.98$$

أي: "احتمال أن تكون القطة لقطة" بشرط رؤية هذه الصورة هو 98%.

دالة الخسارة (Loss Function) والإحصاء: كيف نعرف أن النموذج أخطأ؟ نستخدم مقاييس إحصائية.

• في الانحدار، نستخدم متوسط مربع الخطأ (MSE)، وهو تطبيق مباشر لمفهوم التباين (Variance).

• في التصنيف، نستخدم Cross-Entropy، وهو مقياس مستمد من نظرية المعلومات (Information Theory) والاحتمالات لقياس المسافة بين توزيعين احتماليين (توزيع تنبؤ النموذج وتوزيع الحقيقة).

15.4 أمثلة تطبيقية مبسطة لربط الرياضيات ببرامج عملية

لنربط المفاهيم النظرية بالتطبيق العملي (الخوارزميات):

1. الانحدار الخطي: (Linear Regression)

○ الرياضيات: إيجاد معادلة الخط $y = mx + c$.

○ التطبيق: استخدام "طريقة المربعات الصغرى (Least Squares) "من الجبر الخطي لتقليل المسافة العمودية بين النقاط والخط.

2. أقرب الجيران: (K-Nearest Neighbors - KNN)

○ الرياضيات: تعتمد كلياً على هندسة المسافات (مثل المسافة الإقليدية Euclidean Distance التي درسناها في المتجهات) لمعرفة من هم "جيران" النقطة الجديدة وتصنيفها مثلهم.

3. عنقدة: K-Means

○ الرياضيات: خوارزمية تكرارية تحسب "النقطة المركزية - Centroid) " وهو متوسط هندسي (لمجموعة نقاط، ثم تعيد توزيع النقاط بناءً على قربها من المركز الجديد.

15.5 أخلاقيات الذكاء الاصطناعي والاعتبارات الحسابية

الرياضيات لا تكذب، لكن البيانات قد تكون متحيزة.

1. التحيز الرياضي: (Mathematical Bias) إذا كانت بيانات التدريب تحتوي على تحيز إحصائي (مثلاً صور الأطباء كلها لرجال)، فإن النموذج سيتعلم أن $P(\text{Doctor}|\text{Woman}) \approx 0$. هذا ليس خطأ في الرياضيات، بل خطأ في "العينة الإحصائية" التي لم تمثل المجتمع بصدق.

2. التكلفة الحسابية: (Computational Cost) كلما زاد تعقيد النموذج الرياضي (عدد الأبعاد والمصفوفات)، زادت الحاجة لقوة معالجة و طاقة كهربائية. تدريب نموذج لغوي ضخم (LLM) يتطلب ضرب تريليونات المصفوفات، مما يستهلك طاقة تعادل تشغيل مدينة صغيرة. هنا يأتي دور الرياضيات في محاولة "تبسيط" النماذج (Model Compression) للحفاظ على الدقة مع تقليل الحسابات.

المحور الرابع: علوم البيانات والذكاء الاصطناعي والتطبيقات المتقدمة

الفصل السادس عشر

مدخل إلى علوم البيانات ودورة حياة مشروع البيانات

16.1 تعريف علوم البيانات وتطورها

علم البيانات (Data Science) هو مجال متعدد التخصصات يدمج بين الأساليب العلمية، العمليات الحسابية، والخوارزميات لاستخراج المعرفة والأفكار من البيانات الهيكلية وغير الهيكلية. تاريخياً، كان يُنظر للإحصاء على أنه علم تحليل البيانات الصغيرة. مع انفجار البيانات (Big Data) في العقد الماضي، لم تعد الأدوات التقليدية كافية، فولد "علم البيانات" كدمج بين:

1. الرياضيات والإحصاء: للنمذجة والتحليل.
2. علوم الحاسوب: للقدرة على المعالجة والتخزين والبرمجة.
3. الخبرة في المجال: (Domain Knowledge) فهم سياق البيانات (طب، مال، تسويق).

16.2 أدوار فريق علوم البيانات

في المشاريع الكبرى، لا يقوم شخص واحد بكل شيء. تتكامل الأدوار كالتالي:

1. مهندس البيانات: (Data Engineer)

- المهمة: "السابك" الرقمي. يبني الأنابيب (Pipelines) التي تنقل البيانات من المصادر، وتخزنها، وتجهزها بشكل نظيف وفعال. يركز على قواعد البيانات (SQL/NoSQL) والبنية التحتية السحابية.

2. محلل البيانات: (Data Analyst)

- المهمة: "المحقق". ينظر للبيانات السابقة (Historical Data) للإجابة عن أسئلة: "ماذا حدث؟ ولماذا؟". يستخدم SQL و Excel وأدوات التصوير (Tableau/PowerBI).

3. عالم البيانات: (Data Scientist)

- المهمة: "العراف" العلمي. يستخدم البيانات لبناء نماذج تنبؤية تجيب عن: "ماذا سيحدث مستقبلاً؟". يركز على خوارزميات التعلم الآلي والرياضيات المتقدمة.

4. مهندس تعلم الآلة: (ML Engineer)

- المهمة: تحويل نموذج عالم البيانات (الذي يعمل على جهازه الشخصي) إلى نظام إنتاج (Production System) قوي وقابل للتوسع لخدمة ملايين المستخدمين.

16.3 أنواع البيانات ومصادرها

1. البيانات المهيكلة: (Structured Data)

- بيانات منظمة جداً في جداول (صفوف وأعمدة).
- أمثلة: جداول Excel ، قواعد بيانات SQL ، سجلات المبيعات.
- سهولة التعامل: عالية جداً.

2. البيانات شبه المهيكلة: (Semi-Structured)

- لا تتبع جدولاً صارماً ولكن لها هيكلية داخلية ووسوم.
- أمثلة: ملفات JSON ، XML ، البريد الإلكتروني (يحتوي حقول ثابتة كالعنوان والتاريخ، ومتن رسالة غير مهيكلة).

3. البيانات غير المهيكلة: (Unstructured Data)

- الكتلة الأكبر من البيانات في العالم (80%+). ليس لها قالب محدد.
- أمثلة: الصور، مقاطع الفيديو، الملفات الصوتية، منشورات التواصل الاجتماعي، ملفات

PDF.

○ التحدي: تتطلب تقنيات ذكاء اصطناعي متقدمة (Deep Learning) لاستخراج المعلومات منها.

16.4 دورة حياة مشروع علوم البيانات (Data Science Lifecycle)

أي مشروع ناجح يتبع منهجية علمية مثل: (CRISP-DM)

1. فهم العمل: (Business Understanding) تحديد المشكلة بدقة. (مثال: نريد تقليل انسحاب العملاء بنسبة 5%).
2. جمع البيانات: (Data Collection) سحب البيانات من المصادر المختلفة.
3. فهم البيانات: (Data Understanding) استكشاف أولي لمعرفة طبيعة البيانات وجودتها.
4. إعداد البيانات: (Data Preparation) المرحلة الأطول (70-80% من الوقت). تشمل التنظيف والتحويل.
5. النمذجة: (Modeling) اختيار الخوارزمية وتدريب النموذج.
6. التقييم: (Evaluation) اختبار دقة النموذج وهل يلبي هدف العمل.
7. النشر: (Deployment) وضع النموذج في بيئة العمل الحقيقية.

16.5 تنظيف البيانات ومعالجتها (Data Cleaning)

البيانات الخام دائماً "قذرة". قاعدة "Garbage In, Garbage Out" تحكم هذا المجال.

- القيم المفقودة: (Missing Values) هل نحذف الصف؟ أم نملأ الفراغ بالمتوسط (Imputation)؟
- البيانات الشاذة: (Outliers) قيم متطرفة قد تكون خطأً (عمر شخص 200 سنة) أو حقيقة نادرة (عملية احتيال بنكي). التعامل معها يتطلب حذراً.
- التكرارات: (Duplicates) إزالة السجلات المكررة لضمان عدم تحيز النموذج.

16.6 الاستكشاف البصري للبيانات (EDA - Exploratory Data Analysis)

قبل تطبيق أي خوارزمية معقدة، يجب أن "نرى" البيانات.

- استخدام الرسوم البيانية (Histograms, Scatter Plots, Box Plots) لفهم التوزيعات والعلاقات.
- الهدف: اكتشاف الأنماط الخفية، وفهم الارتباطات (Correlations) بين المتغيرات (مثلاً: هل هناك علاقة بين درجة الحرارة ومبيعات الأيس كريم؟).

16.7 دراسة حالة مبسطة: التنبؤ بأسعار المنازل

لتوضيح الدورة الكاملة:

1. الهدف: بناء نموذج لتسعير العقارات.
2. البيانات: سجلات تاريخية للمنازل (المساحة، الموقع، عدد الغرف، السعر).
3. التنظيف: حذف المنازل ذات الأسعار الصفرية (أخطاء)، وتحويل النصوص (اسم الحي) إلى أرقام.
4. النمذجة: استخدام خوارزمية "الانحدار الخطي".
5. النتيجة: معادلة رياضية يمكنها تقدير سعر أي منزل جديد بمجرد إدخال مواصفاته.

الفصل السابع عشر

أساسيات تعلم الآلة (Machine Learning)

17.1 مقدمة في تعلم الآلة وأنواعه

تعلم الآلة هو مجال فرعي من الذكاء الاصطناعي يركز على تطوير خوارزميات تمكن الحواسيب من التعلم من البيانات واكتشاف الأنماط المخفية دون الحاجة لبرمجتها صراحةً لكل قاعدة (Explicit Programming). بدلاً من كتابة كود يقول: "إذا كانت الصورة تحتوي على شوارب وأذنين مدببتين فهي قطة"، نعطي الخوارزمية آلاف الصور للقطط ونتركها تستنتج القواعد بنفسها.

ينقسم تعلم الآلة تقليدياً إلى ثلاثة أنماط رئيسية:

1. التعلم تحت الإشراف: (Supervised Learning)

- الآلية: يتم تدريب النموذج على بيانات "موسومة" (Labeled Data)، أي أننا نعطي الحاسوب السؤال (المدخلات) والإجابة الصحيحة (المخرجات).
- الهدف: أن يتعلم النموذج الدالة التي تربط المدخلات بالمخرجات ليتنبأ بالإجابة لبيانات جديدة لم يرها من قبل.
- أمثلة: تصنيف البريد المزعج، تشخيص الأمراض، التنبؤ بأسعار الأسهم.

2. التعلم غير الخاضع للإشراف: (Unsupervised Learning)

- الآلية: البيانات هنا "غير موسومة" (لا توجد إجابات صحيحة مسبقة). يُترك النموذج وحيداً مع البيانات ليكتشف بنيتها الداخلية.
- الهدف: استكشاف الأنماط، التجمعات، أو العلاقات الخفية.
- أمثلة: تقسيم العملاء إلى شرائح تسويقية (Segmentation)، ضغط الصور، أنظمة التوصية (Recommender Systems).

3. تعلم التعزيز (Reinforcement Learning)

- الآلية: يتعلم "العميل (Agent)" عبر التفاعل مع بيئة معينة وتلقي "مكافآت" (Rewards) عند النجاح و"عقوبات (Penalties)" عند الخطأ.
- الهدف: تعظيم مجموع المكافآت على المدى الطويل.
- أمثلة: الروبوتات، السيارات ذاتية القيادة، وبرامج لعب الشطرنج (AlphaGo).

17.2 بيانات التدريب والتحقق والاختبار (Train/Validation/Test Split)

أخطر مشكلة في تعلم الآلة هي **الحفظ الزائد (Overfitting)**؛ حيث يقوم النموذج بحفظ بيانات التدريب عن ظهر قلب بدلاً من فهم القواعد العامة، مما يجعله يفشل فشلاً ذريعاً عند مواجهة بيانات جديدة. لتجنب ذلك، نقوم بتقسيم البيانات المتاحة إلى ثلاث مجموعات:

1. **مجموعة التدريب (Training Set) - حوالي 70-80%:** تُستخدم لتعليم النموذج (تعديل الأوزان).
2. **مجموعة التحقق (Validation Set) - حوالي 10-15%:** تُستخدم أثناء التدريب لضبط "المعاملات العليا (Hyperparameters)" واختيار أفضل نسخة من النموذج. هي بمثابة "اختبار تجريبي".
3. **مجموعة الاختبار (Test Set) - حوالي 10-15%:** تُعزل تماماً ولا يراها النموذج إطلاقاً أثناء التدريب. تُستخدم فقط في النهاية لتقييم الأداء الحقيقي للنموذج في العالم الواقعي.

17.3 نماذج التعلم تحت الإشراف

تنقسم المشاكل هنا إلى نوعين:

1. الانحدار (Regression)

عندما يكون المخرج المطلوب رقماً متصلاً (Continuous).

- **الانحدار الخطي (Linear Regression):** أبسط النماذج، يحاول رسم خط مستقيم يمر عبر البيانات.

- أشجار القرار (Decision Trees) للانحدار: تقسيم البيانات عبر سلسلة من الأسئلة للوصول لقيمة تقديرية.

2. التصنيف (Classification)

عندما يكون المخرج فئة محددة. (Categorical).

- الانحدار اللوجستي (Logistic Regression): رغم اسمه، هو خوارزمية تصنيف تحسب "احتمالية" الانتماء لفئة معينة (بين 0 و 1).
- آلات المتجهات الداعمة (Support Vector Machines - SVM): تبحث عن أفضل "حد فاصل (Hyperplane)" يفصل بين الفئات بأكبر هامش ممكن. قوية جداً في البيانات المعقدة.
- الغابات العشوائية (Random Forests): استخدام مئات من أشجار القرار معاً والتصويت على النتيجة النهائية (Ensemble Learning)، مما يعطي دقة عالية جداً ومناعة ضد الـ Overfitting

17.4 نماذج التعلم غير الخاضع للإشراف

1. العنقدة (Clustering)

تجميع البيانات المتشابهة معاً.

- K-Means: الخوارزمية الأشهر. تحدد K مراكز، وتوزع النقاط حول أقرب مركز لها.
- DBSCAN: تعتمد على كثافة النقاط، وقادرة على اكتشاف الأشكال غير المنتظمة واستبعاد الضوضاء.

2. تقليل الأبعاد (Dimensionality Reduction)

تبسيط البيانات المعقدة (ذات آلاف الميزات) إلى عدد أقل من الميزات مع الحفاظ على المعلومات الأساسية.

- **تحليل المكونات الرئيسية (PCA):** عملية رياضية (جبر خطي) تقوم بإسقاط البيانات على محاور جديدة تحافظ على أكبر قدر من التباين (Variance) مفيدة جداً لتسريع التدريب والتصوير البياني (Visualization).

17.5 مقاييس تقييم النماذج (Evaluation Metrics)

كيف نحكم على جودة النموذج؟ الدقة وحدها خادعة.

1. **مصفوفة الارتباك (Confusion Matrix):** جدول يوضح عدد التوقعات الصحيحة والخاطئة لكل فئة. (True Positive, True Negative, False Positive, False Negative)
2. **الدقة (Accuracy):** نسبة التوقعات الصحيحة الكلية. (خادعة إذا كانت البيانات غير متوازنة، مثلاً 99% من المرضى أصحاء).
3. **الدقة (Precision):** من بين كل ما توقع النموذج أنه "إيجابي"، كم واحد كان فعلاً إيجابياً؟ (مهمة لتقليل الإنذارات الكاذبة).
4. **الاستدعاء (Recall / Sensitivity):** من بين كل الحالات الإيجابية الحقيقية، كم واحدة اكتشفها النموذج؟ (حيوية في المجال الطبي؛ عدم تفويت مريض سرطان أهم من التشخيص الخاطئ لشخص سليم).
5. **F1-Score:** المتوسط التوافقي بين Precision و Recall. يعطي صورة متوازنة للأداء.
6. **ROC-AUC:** منحنى يقيس قدرة النموذج على التمييز بين الفئات عند عتبات تصنيف مختلفة. كلما اقتربت المساحة تحت المنحنى (AUC) من 1، كان النموذج مثالياً.

17.6 ضبط المعاملات وتقنيات Cross-Validation

- **المعاملات العليا (Hyperparameters):** هي إعدادات الخوارزمية التي لا يتعلمها النموذج بنفسه بل يضبطها المهندس) مثل عدد الأشجار في الغابة العشوائية، أو معدل التعلم (Learning Rate)
- **البحث الشبكي (Grid Search):** تجربة كل التوفيقات الممكنة للقيم للعثور على الأفضل (مكلفة حسابياً).

- **Cross-Validation (K-Fold):** تقنية لتقييم النموذج بمصدقية عالية. يتم تقسيم البيانات إلى K أجزاء (مثلاً 5). يتم تدريب النموذج 5 مرات، في كل مرة نستخدم جزءاً مختلفاً للاختبار والباقي للتدريب، ثم نأخذ متوسط النتائج. هذا يضمن أن التقييم لم يكن صدفة بسبب تقسيم معين للبيانات.

17.7 مثال تطبيقي باستخدام مكتبة Scikit-Learn

تعتبر scikit-learn المكتبة القياسية لتعلم الآلة التقليدي في بايثون. سير العمل النمطي (Pipeline) يكون كالتالي:

```
# 1. استدعاء المكتبات
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# 2. تقسيم البيانات
# X: الميزات (Features), y: الهدف (Target)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# 3. بناء النموذج (Instantiation)
model = RandomForestClassifier(n_estimators=100)

# 4. التدريب (Fitting)
model.fit(X_train, y_train)

# 5. التنبؤ (Prediction)
predictions = model.predict(X_test)

# 6. التقييم (Evaluation)
print("Accuracy:", accuracy_score(y_test, predictions))
```

أولاً: المعادلات الإحصائية في نموذج Random Forest

نموذج Random Forest ليس "معادلة واحدة" مثل الانحدار الخطي، بل هو مجموعة من الأشجار (Decision Trees)، وكل شجرة تُصدر تنبؤاً ثم تُدمج النتائج كالتالي:

1. التنبؤ العام: (Prediction)

لكل شجرة T_i من بين N أشجار:

$$\hat{y}_i = T_i(X)$$

ثم يتم حساب النتيجة النهائية بالأغلبية (للتصنيف) أو المتوسط (للانحدار):

$$\hat{y} = \text{mode}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N)$$

أو في حالة الانحدار: (Regression)

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N \hat{y}_i$$

ثانياً: معادلات حسابية لبيانات معدلات الرياضيات

نفترض أن الطالب لديه عدة مكونات تؤثر على نتيجته النهائية في مادة الرياضيات، مثل:

- الاختبار الأول T_1
- الاختبار الثاني T_2
- الواجبات H
- الحضور A
- النشاط P

يمكننا تمثيل معدل الرياضيات النهائي (M) بهذه المعادلة:

$$M = 0.4T_1 + 0.4T_2 + 0.1H + 0.05A + 0.05P$$

مثال تطبيقي

لو كانت القيم كالتالي:

$$T_1 = 70 \quad \bullet$$

$$T_2 = 80 \quad \bullet$$

$$H = 100 \quad \bullet$$

$$A = 90 \quad \bullet$$

$$P = 80 \quad \bullet$$

إذن:

$$M = 0.4(70) + 0.4(80) + 0.1(100) + 0.05(90) + 0.05(80)$$

$$M = 28 + 32 + 10 + 4.5 + 4 = 78.5$$

النتيجة: معدل الطالب في الرياضيات %78.5 =

ثالثاً: معادلات التقييم في النموذج

عند تقييم النموذج نستخدم دقة التصنيف: (Accuracy)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

حيث:

• TP: عدد الحالات التي تنبأ بها النموذج بـ "ناجح" وكانت صحيحة.

• TN: عدد الحالات التي تنبأ بها بـ "راسب" وكانت صحيحة.

• FP: تنبأ بـ "ناجح" ولكن الطالب راسب.

• FN: تنبأ بـ "راسب" ولكن الطالب ناجح.

الفصل الثامن عشر

الشبكات العصبية العميقة والتعلم العميق (Deep Learning)

18.1 البنية العامة للشبكات العصبية الاصطناعية (ANN)

الشبكة العصبية الاصطناعية هي نموذج حوسبي مستوحى من البنية البيولوجية للدماغ البشري. تتكون من وحدات معالجة بسيطة تسمى "عصبونات (Neurons)" تترابط فيما بينها لتشكل شبكة معقدة.

المعمارية الأساسية: (The Perceptron & MLP)

1. **طبقة الإدخال (Input Layer):** تستقبل البيانات الخام (بكسلات الصورة، كلمات النص). لا تقوم بأي معالجة.

2. **الطبقات الخفية (Hidden Layers):** هي "الصندوق الأسود" حيث يحدث السحر. كلما زاد عدد هذه الطبقات، أصبحت الشبكة "عميقة (Deep)"، وزادت قدرتها على فهم الميزات المعقدة والمجردة.

3. **طبقة الإخراج (Output Layer):** تعطي النتيجة النهائية (احتمالات التصنيف).

4. **الأوزان (Weights):** تمثل قوة الاتصال بين العصبونات. التدريب هو عملية تعديل هذه الأوزان.

5. **دوال التنشيط (Activation Functions):** مثل ReLU, Sigmoid, Tanh هي صمامات رياضية تقرر ما إذا كان العصبون سيطلق إشارة أم لا، وهي المسؤولة عن إدخال "اللاخطية" (Non-linearity) في الشبكة، مما يسمح لها بتعلم أنماط أعقد من مجرد خط مستقيم.

18.2 خوارزمية الانتشار العكسي (Backpropagation) والهبوط التدريجي

كيف تتعلم الشبكة؟ العملية تتم في دورتين:

1. **الانتشار الأمامي (Forward Propagation):** تمر البيانات من المدخلات إلى المخرجات، وتقوم الشبكة بإعطاء تخمين (غالباً خاطئ في البداية).

2. **حساب الخطأ (Loss Calculation):** مقارنة التخمين بالإجابة الصحيحة.

3. الانتشار العكسي: (Backpropagation) هنا تكمن عبقرية الرياضيات. نستخدم التفاضل) قاعدة السلسلة (Chain Rule) للعودة من المخرجات إلى الخلف، لنحسب "مسؤولية" كل وزن في الشبكة عن الخطأ الكلي.

4. تحديث الأوزان: (Weight Update) باستخدام خوارزمية الهبوط التدريجي (Gradient Descent)، نقوم بتعديل الأوزان قليلاً في الاتجاه المعاكس للخطأ.

بتكرار هذه العملية ملايين المرات، يقل الخطأ تدريجياً وتصبح الشبكة ذكية.

18.3 الشبكات الالتفافية (CNN) للرؤية الحاسوبية

الشبكات التقليدية تفشل مع الصور لأنها لا تدرك العلاقات المكانية (Spatial Relationships) بين البكسلات. الشبكات الالتفافية (Convolutional Neural Networks) حلت هذه المشكلة.

آلية العمل: بدلاً من النظر للصورة كقائمة أرقام مسطحة، تمرر الـ "CNN فلاتر (Filters/Kernels)" صغيرة على الصورة.

1. طبقة الالتفاف: (Convolution Layer) تكتشف الميزات البسيطة (حواف، خطوط، ألوان) في الطبقات الأولى، ثم تدمجها لتكتشف ميزات أعقد (عيون، عجلات) في الطبقات العميقة.

2. طبقة التجميع: (Pooling Layer) تقلل حجم الصورة (تأخذ الماكسيموم في كل منطقة) لتقليل الحسابات والتركيز على الميزات الأهم. (Feature Extraction)

3. الطبقة المتصلة بالكامل: (Fully Connected) في النهاية، تأخذ الميزات المستخلصة وتقوم بالتصنيف.

• تطبيقات: التعرف على الوجوه، تحليل الأشعة الطبية، السيارات ذاتية القيادة.

18.4 الشبكات المتتابعة (RNN, LSTM) للسلاسل الزمنية والنصوص

عندما يكون للبيانات "ترتيب" زمني (مثل كلمات جملة، أو أسعار أسهم يومية)، فإن الشبكات العادية تفشل لأنها لا تملك "ذاكرة".

الشبكات العصبية المتكررة: (RNN) تملك حلقة تكرارية تسمح بتمرير المعلومات من خطوة زمنية سابقة إلى الخطوة الحالية). المشكلة: تعاني من "نسيان" المعلومات القديمة جداً بسبب مشكلة تلاشي التدرج (Vanishing Gradient).

الذاكرة طويلة وقصيرة المدى: (LSTM) تطوير عقري للـ RNN. تحتوي الخلية العصبية فيها على "بوابات: (Gates) "

- بوابة النسيان: (Forget Gate) تقرر ما يجب حذفه من الذاكرة.
- بوابة الإدخال: (Input Gate) تقرر ما يجب حفظه.
- هذا يسمح لها بربط سياق بعيد (مثل تذكر جنس الفاعل في بداية فقرة طويلة لتحديد الضمير المناسب في نهايتها).

18.5 لمحة عن Transformers والنماذج اللغوية الكبيرة (LLMs)

في عام 2017، غيرت ورقة بحثية بعنوان "Attention is All You Need" وجه العالم.

آلية الانتباه: (Self-Attention) بدلاً من معالجة الكلمات بالترتيب) مثل (RNN ، يستطيع نموذج Transformer النظر للجملة كاملة في آن واحد، وحساب علاقة كل كلمة بكل الكلمات الأخرى. هذا يسمح بفهم عميق للسياق والترابط المعنوي بغض النظر عن المسافة بين الكلمات.

النماذج اللغوية الكبيرة: (LLMs) مثل GPT و BERT هي شبكات Transformer عملاقة تدربت على كميات مهولة من النصوص (الإنترنت بأكمله تقريباً).

- التدريب المسبق: (Pre-training) تتعلم اللغة بشكل عام (التنبؤ بالكلمة التالية).
- الضبط الدقيق: (Fine-tuning) تخصص لأداء مهام محددة.

18.6 متطلبات التدريب GPU/TPU

التعلم العميق "نهم" للموارد.

- المعالجة المتوازية: نحتاج لضرب ملايين المصفوفات في وقت واحد. المعالج المركزي (CPU) بطيء جداً لهذا، لذا نستخدم وحدات معالجة الرسومات (GPU) التي تحتوي آلاف الأنوية، أو وحدات TPU المخصصة من جوجل.

- **حجم الدُفعات (Batch Size):** عدد الأمثلة التي تعالجها الشبكة قبل تحديث الأوزان مرة واحدة. التوازن مطلوب: دفعة كبيرة جداً تستهلك الذاكرة (VRAM) ، ودفعة صغيرة جداً تجعل التدريب غير مستقر.

18.7 تحديات عملية **Overfitting** ، **Regularization** ، **Dropout**

كلما كانت الشبكة أعمق، زاد خطر حفظ البيانات. (**Overfitting**) **الحلول التقنية:**

1. **التنظيم (Regularization - L1/L2):** إضافة "عقوبة" رياضية في معادلة الخطأ تمنع الأوزان من أن تصبح كبيرة جداً.
2. **الإسقاط العشوائي (Dropout):** تقنية ذكية جداً؛ أثناء التدريب، نقوم بإطفاء نسبة عشوائية من العصبونات (مثلاً 50%) في كل خطوة. هذا يجبر الشبكة على عدم الاعتماد على عصبون واحد معين، بل توزيع المعرفة على كامل الشبكة، مما يجعلها أكثر قوة ومتانة. (**Robustness**)
3. **التوقف المبكر (Early Stopping):** مراقبة دقة النموذج على "بيانات التحقق"؛ بمجرد أن تبدأ الدقة بالتراجع، نوقف التدريب فوراً قبل أن يبدأ النموذج بالحفظ.

الفصل التاسع عشر

هندسة البيانات وبُنى الذكاء الاصطناعي في البيئات الواقعية

19.1 قواعد البيانات العلائقية SQL وغير العلائقية NoSQL

تخزين البيانات هو الخطوة الأولى في أي بنية تحتية. لا يوجد حل واحد يناسب الجميع.

1. قواعد البيانات العلائقية: (RDBMS - SQL)

- البنية: جداول صارمة (Schemas) وعلاقات محددة) مفتاح أجنبي. (Foreign Key)
- الخصائص: تضمن سلامة البيانات. (ACID Compliant)
- متى نستخدمها؟ للبيانات المالية، سجلات المستخدمين، وأي بيانات تتطلب دقة واتساقاً عالياً.
- أمثلة: PostgreSQL, MySQL, SQL Server.

2. قواعد البيانات غير العلائقية: (NoSQL)

- البنية: مرنة. (Schema-less) قد تكون وثائق (Documents) ، مفتاح-قيمة (Key-Value) ، أو رسوم بيانية. (Graphs)
- الخصائص: قابلة للتوسع الأفقي الهائل. (Scalability) مصممة للسرعة والبيانات غير المهيكلة.
- متى نستخدمها؟ لتخزين محتوى التواصل الاجتماعي، بيانات إنترنت الأشياء، أو الكاش السريع.
- أمثلة: MongoDB, Cassandra, Redis.

19.2 مستودعات البيانات (Data Warehouse) وبحيرات البيانات (Data Lake)

عندما تتناثر البيانات في عشرات القواعد، نحتاج لتوحيدها للتحليل.

1. مستودع البيانات: (Data Warehouse)

- مخزن مركزي للبيانات **المعالجة والمنظمة** (Structured). صمم خصيصاً للاستعلامات التحليلية السريعة (OLAP). البيانات هنا جاهزة للاستهلاك من قبل المحللين.
- أمثلة: Snowflake, Google BigQuery.

2. بحيرة البيانات: (Data Lake)

- مخزن ضخم للبيانات **الخام** بكل أشكالها (صور، نصوص، جداول) دون معالجة مسبقة. يحتفظ بالبيانات "كما هي" ليقوم علماء البيانات باستكشافها لاحقاً.
- أمثلة: Amazon S3, Azure Data Lake.
- Lakehouse: مفهوم حديث يدمج مزايا الاثنين.

19.3 أنابيب البيانات (ETL/ELT)

كيف تنتقل البيانات من المصدر إلى المستودع؟

1. ETL (Extract, Transform, Load):

- استخراج البيانات -> معالجتها وتنظيفها في خادم وسيط -> تحميلها للمستودع. (النهج التقليدي).

2. ELT (Extract, Load, Transform):

- استخراج -> تحميل فوراً للمستودع السحابي -> المعالجة داخله. (النهج الحديث بفضل قوة السحابة). يسمح بالسرعة والمرونة.
- أدوات (Apache Airflow): للجدولة والأتمتة.

19.4 المعالجة الموزعة (Hadoop, Spark)

عندما يكون حجم البيانات بالبيتابايت (Petabytes)، لا يوجد حاسوب واحد قادر على معالجتها.

- **Hadoop MapReduce** التقنية الرائدة التي قسمت البيانات ووزعت المعالجة على آلاف الأجهزة. (قديمة وبطيئة لأنها تكتب على القرص كثيراً).

- **Apache Spark** الجيل الجديد. يقوم بالمعالجة في الذاكرة (In-Memory Processing) مما يجعله أسرع بـ 100 مرة. هو المعيار الفعلي اليوم لمعالجة البيانات الضخمة وتدريب النماذج عليها.

19.5 أساسيات MLOps من النموذج التجريبي إلى الإنتاج

MLOps هو دمج لمفاهيم DevOps مع خصوصية تعلم الآلة. الهدف هو أتمتة دورة حياة النموذج.

- **إدارة التجارب (Experiment Tracking)**: تسجيل كل تجربة تدريب (الأوزان، الدقة، الكود) لضمان القابلية للتكرار (Reproducibility) أداة MLflow .
- **إصدارات البيانات (Data Versioning)**: البيانات تتغير. يجب أن نعرف بالضبط أي نسخة من البيانات تُرب عليها النموذج). أداة (DVC) .

19.6 نشر نماذج التعلم الآلي كخدمات

بمجرد تدريب النموذج، كيف يستخدمه الناس؟

1. **API (REST/gRPC)**: تغلف النموذج في خدمة ويب) باستخدام FastAPI أو Flask). يرسل التطبيق البيانات (JSON)، ويرد النموذج بالتنبؤ.
2. **Microservices**: يكون نموذج الذكاء الاصطناعي "خدمة صغيرة" مستقلة تتحدث مع باقي أجزاء النظام، مما يسهل تحديثه دون إيقاف النظام كاملاً.
3. **Edge Deployment**: تصدير النموذج بصيغة خفيفة (TFLite, ONNX) ليعمل مباشرة على الهواتف المحمولة دون إنترنت.

19.7 مراقبة النماذج (Model Monitoring)

البرمجيات التقليدية إما تعمل أو تتعطل. نماذج الذكاء الاصطناعي "تتلعن" بصمت.

- **انحراف البيانات (Data Drift)**: عندما تتغير طبيعة البيانات الحقيقية عن البيانات التي تدرب عليها النموذج (مثلاً: تغير عادات الشراء بعد الوباء).
- **انحراف المفهوم (Concept Drift)**: عندما تتغير العلاقة بين المدخلات والمخرجات.
- **الحل**: أنظمة مراقبة تنبه المهندسين عند تراجع الدقة لإعادة تدريب النموذج (Retraining) تلقائياً .

الفصل العشرون

أخلاقيات الذكاء الاصطناعي ومستقبل الحوسبة والبحث العلمي

20.1 مبادئ أخلاقيات الذكاء الاصطناعي

مع تزايد مستوى الاستقلالية واتخاذ القرار الذاتي في الأنظمة الذكية، لم يعد الذكاء الاصطناعي مجرد أداة تقنية، بل أصبح فاعلاً مؤثراً في القرارات الاجتماعية والاقتصادية والقانونية. وقد أدى ذلك إلى ظهور الحاجة إلى ميثاق أخلاقي عالمي ينظم تصميم وتطوير واستخدام هذه الأنظمة، ويضمن توافقها مع القيم الإنسانية الأساسية. وتتمحور أخلاقيات الذكاء الاصطناعي حول مجموعة من المبادئ الجوهرية، من أبرزها:

- **العدالة: (Fairness)** تهدف إلى ضمان عدم انحياز الأنظمة الذكية ضد أي فئة اجتماعية، سواء على أساس العرق أو الجنس أو الخلفية الثقافية أو الاقتصادية. ويتطلب تحقيق العدالة تصميم نماذج وخوارزميات تقلل من احتمالية التمييز غير المقصود في المخرجات.
- **الشفافية: (Transparency)** تقتضي تمكين المستخدم من معرفة أنه يتعامل مع نظام آلي، وفهم الأسس العامة التي بُني عليها القرار المتخذ. ويُعرف هذا التوجه بمفهوم الذكاء الاصطناعي القابل للتفسير (Explainable AI)، الذي يهدف إلى تقليل "الصندوق الأسود" في النماذج المعقدة.
- **المساءلة: (Accountability)** تطرح تساؤلاً قانونياً وأخلاقياً محورياً حول الجهة المسؤولة عند حدوث خطأ أو ضرر ناتج عن نظام ذكي، سواء أكان المطور، أو الشركة المالكة، أو المستخدم النهائي، وهو ما يتطلب أطراً قانونية واضحة.
- **الأمان والموثوقية:** يركز هذا المبدأ على ضمان أن تعمل الأنظمة الذكية ضمن حدود آمنة، وألا تسبب أضراراً مادية أو نفسية أو اجتماعية، مع التأكد من استقرار أدائها في الظروف المختلفة.

20.2 التحيز الخوارزمي (Algorithmic Bias)

على خلاف الاعتقاد الشائع، فإن الخوارزميات ليست كيانات محايدة بطبيعتها، بل تعكس طبيعة البيانات التي يتم تدريبها عليها. فإذا كانت البيانات التاريخية المستخدمة في مجال ما—مثل التوظيف أو القبول

الجامعي—منحازة لفئة معينة، فإن النموذج سيتعلم هذا التحيز ويعيد إنتاجه، حتى في غياب نية برمجية صريحة للتمييز. ويُعرف هذا السلوك بـ **التحيز الخوارزمي**.

وتتعدد أشكال هذا التحيز، من أبرزها:

- **تحيز التمثيل:** الناتج عن استخدام عينات بيانات غير شاملة لجميع الفئات.
- **تحيز القياس:** الذي ينشأ نتيجة أدوات أو معايير قياس غير دقيقة أو غير عادلة.
- **تحيز التأكيد:** حيث تعزز الخوارزميات أنماطًا موجودة مسبقًا في البيانات.

ويمكن الحل في اعتماد عمليات **تدقيق منهجية للبيانات والخوارزميات (Bias Audit)**، إلى جانب استخدام تقنيات رياضية وإحصائية تهدف إلى تقليل التحيز في المخرجات، وضمان عدالة أكبر في القرارات الآلية.

20.3 الخصوصية وحماية البيانات

في عصر يُشار فيه إلى البيانات على أنها “النفط الجديد”، أصبحت الخصوصية من أكثر القضايا عرضة للانتهاك، خاصة مع الاعتماد الواسع على جمع وتحليل البيانات الشخصية لتدريب نماذج الذكاء الاصطناعي. وقد أدى ذلك إلى تطوير تقنيات متقدمة تهدف إلى **الحفاظ على خصوصية المستخدمين** دون تعطيل قدرات التعلم الآلي.

ومن أبرز هذه التقنيات:

- **التعلم الفيدرالي (Federated Learning):** يعتمد على تدريب النماذج محليًا على أجهزة المستخدمين، مثل الهواتف الذكية، مع إرسال تحديثات الأوزان فقط إلى الخادم المركزي، دون نقل البيانات الشخصية نفسها، مما يقلل مخاطر تسريب المعلومات.
- **الخصوصية التفاضلية (Differential Privacy):** تقوم على إضافة ضوضاء رياضية محسوبة إلى البيانات أو النتائج الإحصائية، بما يمنع استنتاج معلومات دقيقة عن فرد بعينه، مع الحفاظ على القيمة التحليلية العامة للبيانات.

20.4 الذكاء الاصطناعي وسوق العمل

يثير الذكاء الاصطناعي مخاوف متزايدة تتعلق بفقدان الوظائف، وغالبًا ما يُختزل هذا النقاش في فكرة "استبدال الروبوتات للبشر". إلا أن الواقع أكثر تعقيدًا وتوازنًا، إذ يمكن تلخيص تأثير الذكاء الاصطناعي على سوق العمل في محورين رئيسيين:

- **الأتمة:** حيث تتجه الأنظمة الذكية إلى تولي المهام الروتينية والمتكررة، مما يؤدي إلى تراجع بعض الوظائف التقليدية.

- **التعزيز: (Augmentation)** في المقابل، يعمل الذكاء الاصطناعي على دعم الإنسان وتعزيز قدراته، وليس استبداله بالكامل. فالمختص الذي يستخدم أدوات الذكاء الاصطناعي بكفاءة سيكون أكثر قدرة على المنافسة من نظيره الذي يتجاهلها.

وفي هذا السياق، تبرز أهمية **المهارات المستقبلية**، مثل التفكير النقدي، والإبداع، والذكاء العاطفي، والقدرة على فهم البيانات والتعامل معها، بوصفها عناصر أساسية لضمان التكيف مع التحولات المتسارعة في سوق العمل.

20.5 الاستدامة والآثار البيئية

مع التوسع المتزايد في تطوير ونشر نماذج الذكاء الاصطناعي، ولا سيما النماذج اللغوية الكبيرة، برزت قضية الاستدامة البيئية بوصفها أحد التحديات الرئيسية المصاحبة لهذا التقدم التقني. إذ تتطلب عمليات تدريب هذه النماذج قدرًا هائلًا من الموارد الحاسوبية والطاقة الكهربائية، ما يؤدي إلى انبعاثات كربونية ملموسة تسهم في زيادة البصمة البيئية للتقنيات الرقمية. وتشير بعض التقديرات إلى أن تدريب نموذج لغوي واحد كبير قد ينتج انبعاثات كربونية تعادل تلك الصادرة عن عدة سيارات تعمل طوال عمرها التشغيلي، وهو ما يسلب الضوء على حجم التأثير البيئي المحتمل لهذا النوع من التقنيات.

استجابةً لهذه التحديات، ظهر توجه بحثي يُعرف بـ **الذكاء الاصطناعي الأخضر (Green AI)**، يركز على تحقيق كفاءة أعلى في استهلاك الطاقة والموارد، بدل الإقتران على السعي إلى تحسين الدقة والأداء دون اعتبار للتكلفة البيئية. ويقوم هذا التوجه على تطوير نماذج أصغر حجمًا وأكثر كفاءة، قادرة على تحقيق أداء مقبول أو مرتفع باستخدام موارد حاسوبية أقل. كما يشمل اعتماد أساليب تدريب محسنة، مثل تقليل عدد المعاملات غير الضرورية، وإعادة استخدام النماذج المدربة مسبقًا، وتحسين خوارزميات التعلم.

إضافةً إلى ذلك، يشجع الذكاء الاصطناعي الأخضر على تشغيل وتدريب النماذج في مراكز بيانات تعتمد على مصادر الطاقة المتجددة، مثل الطاقة الشمسية وطاقة الرياح، بهدف تقليل الانبعاثات الكربونية المرتبطة بالبنية التحتية الحاسوبية. ويمثل هذا التوجه خطوة أساسية نحو تحقيق توازن مستدام بين التقدم التقني والمسؤولية البيئية، بما يضمن استمرارية تطوير الذكاء الاصطناعي دون الإضرار بالبيئة أو استنزاف الموارد الطبيعية.

20.6 الأطر التشريعية والتنظيمية العالمية

• مع التوسع السريع في استخدام تقنيات الذكاء الاصطناعي وتغلغلها في مختلف القطاعات الحيوية، برزت الحاجة الملحة إلى أطر تشريعية وتنظيمية تضمن الاستخدام الآمن والمسؤول لهذه التقنيات، وتحدّ من مخاطرها المحتملة على الأفراد والمجتمعات. وقد اتجهت العديد من الدول والمنظمات الدولية إلى تطوير سياسات وقوانين تنظم تصميم وتطوير ونشر أنظمة الذكاء الاصطناعي، مع مراعاة الأبعاد الأخلاقية والقانونية والاقتصادية.

• يُعد قانون الذكاء الاصطناعي الأوروبي (EU AI Act) أول إطار تشريعي شامل على المستوى العالمي يُعنى بتنظيم الذكاء الاصطناعي بصورة منهجية. ويعتمد هذا القانون على نهج قائم على تقييم المخاطر، حيث يقوم بتصنيف تطبيقات الذكاء الاصطناعي إلى فئات متعددة بحسب مستوى الخطورة، تشمل التطبيقات ذات المخاطر المقبولة، والتطبيقات عالية الخطورة التي تتطلب التزامات صارمة تتعلق بالشفافية والسلامة والمسائلة، إضافة إلى التطبيقات المحظورة التي يُنظر إليها على أنها تشكل تهديدًا مباشرًا للحقوق الأساسية أو القيم الإنسانية. ويهدف هذا التصنيف إلى توفير إطار قانوني واضح يوازن بين حماية المستخدمين وتعزيز الثقة في تقنيات الذكاء الاصطناعي.

• وعلى الرغم من أهمية هذه الأطر التنظيمية في الحد من الاستخدامات الضارة وضمان الامتثال للمعايير الأخلاقية، فإن التحدي الرئيس يتمثل في تحقيق توازن دقيق بين التنظيم والحماية من جهة، وعدم خنق الابتكار من جهة أخرى. فالإفراط في القيود التشريعية قد يؤدي إلى إبطاء وتيرة البحث والتطوير، وتقليل القدرة التنافسية للشركات والمؤسسات البحثية، في حين أن غياب التنظيم قد يعرّض المجتمعات لمخاطر تقنية وقانونية جسيمة. ومن هنا، تسعى الأطر التشريعية الحديثة إلى تبني سياسات مرنة وقابلة للتحديث، تواكب التطور التقني السريع دون الإخلال بمبادئ السلامة والعدالة والمسؤولية.

20.7 آفاق البحث المستقبلية بعد 2026

إلى أين نتجه؟ آفاق مستقبل الذكاء الاصطناعي

يشهد الذكاء الاصطناعي تطورًا متسارعًا يتجاوز كونه أداة تقنية محدودة الوظائف، ليصبح مجالًا بحثيًا استراتيجيًا يعيد تشكيل مفاهيم الإدراك والمعالجة واتخاذ القرار. وفي هذا السياق، تتجه الأبحاث الحديثة نحو عدد من المسارات المستقبلية الواعدة، من أبرزها ما يأتي:

1. الذكاء الاصطناعي العام (Artificial General Intelligence – AGI)

يمثل الذكاء الاصطناعي العام الحلم الأكبر في مجال الذكاء الاصطناعي، إذ يهدف إلى تطوير أنظمة قادرة على امتلاك ذكاء مرن وشامل يماثل الذكاء البشري في قدرته على التعلم، والفهم، والتعميم، وحل المشكلات عبر مختلف المجالات دون الحاجة إلى تدريب متخصص لكل مهمة على حدة. وعلى خلاف أنظمة الذكاء الاصطناعي الضيق (Narrow AI) المستخدمة حاليًا، فإن أنظمة AGI يفترض أن تتمتع بالقدرة على التكيف مع البيئات المتغيرة، واكتساب مهارات جديدة بصورة ذاتية. ومع ذلك، لا يزال هذا المجال يواجه تحديات علمية وفلسفية وتقنية كبيرة، تتعلق بفهم طبيعة الذكاء نفسه، وضمان الأمان، والتحكم، والأبعاد الأخلاقية لاستخدام مثل هذه الأنظمة.

2. الذكاء الاصطناعي العصبي الرمزي (Neuro-Symbolic Artificial Intelligence)

يُعد الذكاء الاصطناعي العصبي الرمزي أحد الاتجاهات البحثية المتقدمة التي تسعى إلى تجاوز القيود القائمة في النماذج التقليدية. فهو يقوم على دمج قدرات الشبكات العصبية، التي تتميز بفاعليتها في التعلم من البيانات الضخمة والتعرف على الأنماط، مع تقنيات الذكاء الرمزي، التي تعتمد على القواعد المنطقية والاستدلال الصريح. ويهدف هذا الدمج إلى إنتاج أنظمة قادرة على التعلم بكفاءة عالية، مع الحفاظ على قابلية التفسير والشفافية في اتخاذ القرار، وهو ما يُعد مطلبًا أساسيًا في المجالات الحساسة مثل الطب، والقانون، والأنظمة الذكية الحرجة. ويُنظر إلى هذا التوجه على أنه خطوة مهمة نحو تطوير ذكاء اصطناعي أكثر موثوقية وفهمًا للسياق.

3. الحوسبة الحيوية (Biocomputing)

تمثل الحوسبة الحيوية تحولًا جذريًا في مفهوم الحوسبة التقليدية، إذ تعتمد على استغلال الأنظمة البيولوجية، مثل الحمض النووي (DNA) والخلايا الحية، في عمليات تخزين البيانات والمعالجة الحسابية. ويتميز DNA بكثافة تخزين هائلة تفوق الوسائط الرقمية التقليدية، إضافة إلى استهلاكه المنخفض للطاقة وطول عمره الافتراضي. أما المعالجة الحيوية، فتعتمد على التفاعلات البيوكيميائية لتنفيذ العمليات الحسابية بصورة متوازية. وعلى الرغم من أن هذا المجال لا يزال في مراحله البحثية الأولى، إلا أنه يحمل إمكانات مستقبلية كبيرة قد تسهم في حل مشكلات تتجاوز قدرات الحواسيب الكلاسيكية، خصوصًا في مجالات البيانات الضخمة، والذكاء الاصطناعي المتقدم، والعلوم الطبية.

- [1] **Abbas, A. (2025).** *Advanced Computational Mathematics for Robotics*. CRC Press.
- [2] **Aggarwal, C. C. (2018).** *Neural Networks and Deep Learning: A Textbook*. Springer.
- [3] **Alpaydin, E. (2020).** *Introduction to Machine Learning* (4th ed.). MIT Press.
- [4] **Axler, S. (2024).** *Linear Algebra Done Right* (4th ed.). Springer.
- [5] **Bacciu, D., et al. (2026 - Expected).** *The Handbook of Neuro-Symbolic Artificial Intelligence*. IOS Press.
- [6] **Bishop, C. M., & Bishop, H. (2023).** *Deep Learning: Foundations and Concepts*. Springer.
- [7] **Bondy, J. A., & Murty, U. S. R. (2018).** *Graph Theory*. Springer.
- [8] **Bostrom, N. (2016).** *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.
- [9] **Boyd, S., & Vandenberghe, L. (2018).** *Introduction to Applied Linear Algebra*. Cambridge University Press.
- [10] **Brookshear, G., & Brylow, D. (2019).** *Computer Science: An Overview*. Pearson.
- [11] **Chartrand, G., & Zhang, P. (2020).** *A First Course in Graph Theory*. Dover Publications.
- [12] **Chollet, F. (2021).** *Deep Learning with Python* (2nd ed.). Manning Publications.
- [13] **Cormen, T. H., et al. (2022).** *Introduction to Algorithms* (4th ed.). MIT Press.

- [14] **Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020).** *Mathematics for Machine Learning*. Cambridge University Press.
- [15] **Devlin, K. (2021).** *Introduction to Mathematical Thinking*. Keith Devlin.
- [16] **Domingos, P. (2015).** *The Master Algorithm*. Basic Books.
- [17] **Easley, D., & Kleinberg, J. (2019).** *Networks, Crowds, and Markets*. Cambridge University Press.
- [18] **Ford, M. (2021).** *Rule of the Robots: How Artificial Intelligence Will Transform Everything*. Basic Books.
- [19] **Gaddis, T. (2019).** *Starting Out with Programming Logic and Design*. Pearson.
- [20] **Gerstein, M. (2025).** *The Future of Biocomputing and DNA Storage*. Academic Press.
- [21] **Goodfellow, I., Bengio, Y., & Courville, A. (2016).** *Deep Learning*. MIT Press.
- [22] **Graham, R. L., Knuth, D. E., & Patashnik, O. (2017).** *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley.
- [23] **Grimmett, G., & Welsh, D. (2018).** *Probability: An Introduction*. Oxford University Press.
- [24] **Hastie, T., et al. (2017).** *The Elements of Statistical Learning*. Springer.
- [25] **Higham, N. J. (2020).** *Princeton Companion to Applied Mathematics*. Princeton University Press.
- [26] **James, G., et al. (2021).** *An Introduction to Statistical Learning with Applications in R/Python*. Springer.
- [27] **Kelleher, J. D. (2019).** *Deep Learning (The MIT Press Essential Knowledge series)*.

- [28] **Knuth, D. E. (2023).** *The Art of Computer Programming, Vol 4B*. Addison-Wesley.
- [29] **Kurose, J., & Ross, K. (2020).** *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.
- [30] **Lee, K. F. (2018).** *AI Superpowers: China, Silicon Valley, and the New World Order*. Houghton Mifflin.
- [31] **Lipschutz, S., & Lipson, M. (2017).** *Schaum's Outline of Discrete Mathematics*. McGraw-Hill.
- [32] **Mitchell, M. (2019).** *Artificial Intelligence: A Guide for Thinking Humans*. FSG.
- [33] **Murphy, K. P. (2022).** *Probabilistic Machine Learning: An Introduction*. MIT Press.
- [34] **Nielsen, M. A., & Chuang, I. L. (2016).** *Quantum Computation and Quantum Information*. Cambridge.
- [35] **Patterson, D. A., & Hennessy, J. L. (2020).** *Computer Organization and Design RISC-V Edition*. Morgan Kaufmann.
- [36] **Pearl, J., & Mackenzie, D. (2018).** *The Book of Why: The New Science of Cause and Effect*. Basic Books.
- [37] **Pratt, L. (2026 - Forthcoming).** *Neuro-Symbolic AI: The Next Frontier*. O'Reilly Media.
- [38] **Rosen, K. H. (2019).** *Discrete Mathematics and Its Applications* (8th ed.). McGraw-Hill.
- [39] **Russell, S., & Norvig, P. (2021).** *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- [40] **Silberschatz, A., et al. (2018).** *Operating System Concepts* (10th ed.). Wiley.
- [41] **Sommerville, I. (2020).** *Software Engineering* (10th ed.). Pearson.

- [42] **Spivak, M. (2018).** *Calculus* (4th ed.). Publish or Perish.
- [43] **Stallings, W. (2022).** *Computer Organization and Architecture* (11th ed.). Pearson.
- [44] **Stewart, J. (2020).** *Calculus: Early Transcendentals* (9th ed.). Cengage Learning.
- [45] **Strang, G. (2016).** *Introduction to Linear Algebra* (5th ed.). Wellesley-Cambridge Press.
- [46] **Strogatz, S. (2019).** *Infinite Powers: How Calculus Reveals the Secrets of the Universe*. HMH.
- [47] **Sutton, R. S., & Barto, A. G. (2018).** *Reinforcement Learning: An Introduction*. MIT Press.
- [48] **Tanenbaum, A. S., & Wetherall, D. J. (2021).** *Computer Networks* (6th ed.). Pearson.
- [49] **Trefethen, L. N., & Bau III, D. (2022).** *Numerical Linear Algebra*. SIAM.
- [50] **Vanschoren, J., & Rivolli, A. (2025).** *Automated Machine Learning: Methods and Systems*. Cambridge University Press.
- [51] **Wasserman, L. (2016).** *All of Statistics: A Concise Course in Statistical Inference*. Springer.
- [52] **Winston, P. H. (2023).** *Make It Smart: A Guide to Modern AI*. MIT Press.
- [53] **Zaki, M. J., & Meira Jr, W. (2020).** *Data Mining and Machine Learning*. Cambridge.
- [54] **Zhong, L. (2024).** *Neuromorphic Computing: Principles and Architectures*. Springer.
- [55] **Arora, S., & Barak, B. (2016).** *Computational Complexity: A Modern Approach*. Cambridge.

- [56] **Jurafsky, D., & Martin, J. H. (2024).** *Speech and Language Processing* (3rd ed.). Prentice Hall.
- [57] **Karpathy, A. (2025).** *Large Language Models from Scratch*. Technical Press.
- [58] **Ng, A. (2023).** *Machine Learning Yearning*. DeepLearning.AI.
- [59] **Tegmark, M. (2017).** *Life 3.0: Being Human in the Age of Artificial Intelligence*. Knopf.
- [60] **Wolfram, S. (2023).** *What Is ChatGPT Doing ... and Why Does It Work?.* Wolfram Media.
- [61] **Hinton, G. E., & Salakhutdinov, R. R. (2015).** *Multilayer Perceptron Architectures in Deep Learning*. Journal of Machine Learning Research.
- [62] **LeCun, Y., Bengio, Y., & Hinton, G. (2015).** *Deep Learning*. Nature, 521(7553), 436-444.
- [63] **Silver, D., et al. (2016).** *Mastering the game of Go with deep neural networks and tree search*. Nature.
- [64] **Vaswani, A., et al. (2017).** *Attention is All You Need*. Advances in Neural Information Processing Systems.
- [65] **Brown, T., et al. (2020).** *Language Models are Few-Shot Learners*. arXiv preprint arXiv:2005.14165.
- [66] **Marcus, G. (2020).** *The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence*. arXiv. (بحث حول الذكاء العصبي الرمزي).
- [67] **Hitzler, P., & Sarker, M. K. (2022).** *Neuro-Symbolic Artificial Intelligence: The State of the Art*. IOS Press.
- [68] **Churchland, P. S. (2019).** *Conscience: The Origins of Moral Intuition*. W. W. Norton & Company.

- [69] **Goyal, A., & Bengio, Y. (2022).** *Inductive Biases for Deep Learning of Higher-Level Cognitive Structures*. Proceedings of the Royal Society A.
- [70] **O'Neil, C. (2016).** *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown.
- [71] **Mayer-Schönberger, V., & Cukier, K. (2017).** *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Eamon Dolan Books.
- [72] **Harari, Y. N. (2018).** *21 Lessons for the 21st Century*. Spiegel & Grau.
- [73] **D'Ignazio, C., & Klein, L. F. (2020).** *Data Feminism*. MIT Press.
- [74] **Bender, E. M., & Gebru, T. (2021).** *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?*. FAccT '21.
- [75] **Bommasani, R., et al. (2021).** *On the Opportunities and Risks of Foundation Models*. arXiv.
- [76] **Garvie, M. (2025).** *The Biology of Algorithms: Biocomputing in Practice*. BioTech Press.
- [77] **Jordan, M. I. (2019).** *Artificial Intelligence—The Revolution Hasn't Happened Yet*. Harvard Data Science Review.
- [78] **Zhang, C., et al. (2021).** *Understanding Deep Learning (Still) Requires Rethinking Generalization*. Communications of the ACM.
- [79] **Lighthill, J. (2023 - Reprinted).** *Artificial Intelligence: A General Survey*. Science Research Council.
- [80] **Schmidhuber, J. (2015).** *Deep Learning in Neural Networks: An Overview*. Neural Networks, 61, 85-117.
- [81] **He, K., et al. (2016).** *Deep Residual Learning for Image Recognition*. CVPR.

- [82] **Devlin, J., et al. (2019).** *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL.
- [83] **Mnih, V., et al. (2015).** *Human-level control through deep reinforcement learning*. Nature.
- [84] **Hochreiter, S. (2025 - Expected).** *Long Short-Term Memory: The Next Decade*. Springer.
- [85] **Radford, A., et al. (2019).** *Language Models are Unsupervised Multitask Learners*. OpenAI Blog.
- [86] **Ramesh, A., et al. (2021).** *Zero-Shot Text-to-Image Generation*. ICML.
- [87] **Jumper, J., et al. (2021).** *Highly accurate protein structure prediction with AlphaFold*. Nature.
- [88] **Deng, L., & Yu, D. (2014 - Updated 2016).** *Deep Learning: Methods and Applications*. Foundations and Trends in Signal Processing.
- [89] **Goodfellow, I. (2017).** *NIPS 2016 Tutorial: Generative Adversarial Networks*. arXiv.
- [90] **Sze, V., et al. (2017).** *Efficient Processing of Deep Neural Networks: A Tutorial and Survey*. Proceedings of the IEEE.

ثانياً: المراجع باللغة العربية

- [91] إبراهيم، محمد. (2019) مقدمة في علوم الحاسوب وتكنولوجيا المعلومات. دار الكتب العلمية.
- [92] أبو السعود، أحمد. (2021) الذكاء الاصطناعي: ثورة في حياتنا. دار المنهل.
- [93] الببلاوي، سحر. (2018) أساسيات البرمجة بلغة بايثون. مكتبة الأنجلو المصرية.
- [94] الجميل، علي. (2022) الخوارزميات وهياكل البيانات: دليل عملي. دار الفكر العربي.
- [95] الحسن، ياسين. (2023) الرياضيات المتقطعة لعلماء الحاسوب. دار اليازوري العلمية.
- [96] الخالدي، فهد. (2020) نظم التشغيل: المفاهيم والتطبيقات. دار المسيرة.
- [97] الراوي، علي. (2024) مستقبل الحوسبة الحيوية وتخزين البيانات في DNA. المجلة العربية للعلوم والبحث العلمي.
- [98] الزهراني، سعيد. (2017) أمن المعلومات والشبكات. دار وجوه.
- [99] السامرائي، قاسم. (2019) تاريخ الرياضيات وتطورها عبر العصور. دار الرافدين.
- [100] الشناوي، طارق. (2021) أساسيات الجبر الخطي وتطبيقاته البرمجية. دار النهضة العربية.
- [101] العبيدي، ناصر. (2018) هندسة البرمجيات المعاصرة. دار صفاء للنشر.
- [102] الفارس، منى. (2025) الذكاء الاصطناعي التوليدي: الآفاق والتحديات. دار الكتاب الجامعي.
- [103] القحطاني، عبد الله. (2022) التحول الرقمي والأمن السيبراني في الوطن العربي. مركز دراسات الوحدة العربية.
- [104] المالكي، إبراهيم. (2020) مبادئ الإحصاء الحيوي باستخدام الحاسب. دار الخريجي.
- [105] النجار، فايز. (2017) نظم المعلومات الإدارية: منظور إداري. دار الحامد.
- [106] توفيق، عادل. (2023) التعلم العميق: من النظرية إلى التطبيق. دار المعارف.
- [107] جاسم، عمار. (2024) الرياضيات الحاسوبية المتقدمة. دار دجلة.
- [108] حسين، نبيل. (2016) معمارية الحاسوب الرقمي. المكتبة الأكاديمية.
- [109] حمدي، محمود. (2021) علم البيانات: المبادئ والتطبيقات. دار رؤية.
- [110] خليل، إبراهيم. (2019) التحليل العددي وتطبيقاته الهندسية. دار القدس.
- [111] راضي، حامد. (2022) المنطق الرقمي وتصميم الدوائر المنطقية. دار الكتب الوطنية.

- [112] سالم، رشيد. (2018) *إنترنت الأشياء: الواقع والمستقبل*. دار المعرفة.
- [113] سليمان، محمود. (2020) *التفاضل والتكامل في البرمجة*. دار الثقافة للنشر.
- [114] صادق، محمد. (2023) *موسوعة الأمن الرقمي والذكاء الاصطناعي*. المكتب العربي

للمعارف.

- [115] ظافر، أحمد. (2017) *قواعد البيانات: المفاهيم والتصميم*. دار غيداء.
- [116] عباس، حسن. (2021) *الحوسبة السحابية: التقنيات والخدمات*. دار المناهج.
- [117] عبد العزيز، تامر. (2019) *الذكاء الاصطناعي والتعلم الآلي*. دار السحاب.
- [118] علي، محمود. (2022) *التشفير وحماية البيانات الرقمية*. دار الفجر للنشر.
- [119] عمران، يوسف. (2024) *النماذج الرياضية في الاقتصاد الرقمي*. دار الروافد.
- [120] عوض، منصور. (2020) *برمجة الروبوتات والذكاء الاصطناعي*. مكتبة جريز.
- [121] فرج، خالد. (2018) *تطوير تطبيقات الهواتف الذكية*. دار العلم والإيمان.
- [122] كامل، مجدي. (2023) *عصر الحوسبة الكمية*. دار الشروق.
- [123] لطفي، أماني. (2021) *الرياضيات المالية والحاسوب*. دار المنهل.
- [124] محمد، إيهاب. (2020) *أساسيات شبكات الحاسب اللاسلكية*. دار العلوم.
- [125] محمود، طارق. (2019) *التفكير الحسابي وحل المشكلات*. دار غيداء.
- [126] مصطفى، رامي. (2025) *أخلاقيات الذكاء الاصطناعي في القرن الحادي والعشرين*. دار

النهضة.

- [127] منصور، علاء. (2022) *معالجة اللغات الطبيعية باستخدام الحاسوب*. دار الكتب الجامعية.
- [128] ياسين، عمر. (2021) *النماذج الخطية والارتباط الإحصائي*. دار المسيرة.
- [129] يوسف، خالد. (2024) *تقنية البلوكشين (Blockchain) وتطبيقاتها*. دار المستقبل.
- [130] الخوري، أنطوان (2026 - متوقع) *مستقبل التفاعل بين الإنسان والآلة*. دار الفكر.
- [131] الطه، عاصم. (2019) *النمذجة والذكاء الاصطناعي في خدمة المجتمع*. مجلة جامعة دمشق

للعلوم الأساسية.

- [132] بكري، ريم. (2021) *النماذج العصبية الرمزية: نحو ذكاء اصطناعي قابل للتفسير*. المركز العربي للأبحاث.

- [133] درويش، أشرف. (2020) *الحوسبة السحابية وإنترنت الأشياء وتطبيقاتها*. المكتبة

الأكاديمية.

- [134] سمير، تامر. (2018) هندسة الميكاترونكس والذكاء الاصطناعي. دار الكتب والوثائق المصرية.
- [135] الحديدي، محمد. (2022) مبادئ المنطق الرياضي لبرمجة الحاسب. دار المعارف.
- [136] الشوارة، علي. (2023) الأمن السيبراني والحروب الرقمية. دار الخليج للنشر.
- [137] العاني، نزار. (2017) فلسفة الذكاء الاصطناعي والحاسوب. دار المرتضى.
- [138] عبد الهادي، محمد. (2021) استخراج البيانات والتعلم الآلي في المكتبات. دار المصرية اللبنانية.
- [139] إسماعيل، خالد. (2025) التحليل الرياضي المتقدم لعلماء الحاسوب. دار الفجر.
- [140] بدر، محمد. (2019) الخوارزميات الجينية وتطبيقاتها في البحث العملي. دار الكتب العلمية.
- [141] رضوان، أحمد. (2020) النظم الرقمية المدمجة والذكاء الاصطناعي. دار الكتاب الحديث.
- [142] صالح، محمود. (2022) الشبكات العصبية الاصطناعية وتطبيقاتها في التنبؤ. دار الوفاء.
- [143] غالب، حسن. (2016) مدخل إلى لغة R والتحليل الإحصائي. دار النشر للجامعات.
- [144] قاسم، نهى. (2023) اللغويات الحاسوبية والمعالجة الآلية للغة العربية. دار النهضة العربية.
- [145] مرعي، نبيل. (2021) المدخل الشامل لعلوم الحاسوب. مؤسسة الأهرام للنشر.
- [146] نزيه، فؤاد. (2018) تصميم واجهات المستخدم وتجربة المستخدم. (UI/UX) دار السحاب.
- [147] واصل، علي. (2024) العملات الرقمية والبنوك المركزية: رؤية تقنية. دار الروافد.
- [148] يوسف، هشام. (2020) المعالجة المتوازية والحوسبة فائقة الأداء. دار المعارف.
- [149] حماد، سمير. (2025) أساسيات البرمجة الكمية لغير المختصين. دار العلم.
- [150] النجار، سليم (2026 - متوقع). تطور النظم الخبيرة والذكاء الاصطناعي الهجين. المكتب العربي للمعارف.