

Redes, simulación y aprendizaje automático

Juan Seguro

Pregunta central

¿Cómo convirtieron los ordenadores programables las redes y los modelos en sistemas ejecutables capaces de simular procesos complejos y aprender representaciones internas ajustables a partir de datos?

Contexto histórico y evolutivo

La computación programable y la teoría de la información crearon dos abstracciones de posguerra: los procedimientos podían ejecutarse mediante máquinas generales y la comunicación podía cuantificarse independientemente del medio físico. Una tercera línea trató unidades interactuantes como redes. En 1943 Warren McCulloch y Walter Pitts representaron neuronas idealizadas como elementos lógicos. En 1953 Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller y Edward Teller demostraron un potente método de Monte Carlo en un ordenador electrónico, mostrando cómo la simulación estocástica podía explorar sistemas difíciles de resolver analíticamente. El perceptrón de Frank Rosenblatt de 1958 convirtió los pesos adaptativos en elemento central de un clasificador entrenable. En 1986 Rumelhart, Hinton y Williams mostraron cómo la retropropagación podía entrenar redes multicapa para formar representaciones internas.

¿Qué podían saber las comunidades antes de esta transición?

Las matemáticas podían describir ecuaciones diferenciales, modelos probabilísticos y redes, y los dispositivos analógicos podían simular sistemas restringidos. PIO-082 aportó computación programable y PIO-084 información cuantificable. La limitación era la escala práctica y la adaptabilidad: muchos sistemas acoplados eran analíticamente intratables, mientras los programas fijos exigían que los humanos especificaran explícitamente cada transformación.

¿Qué cambió?

La simulación se convirtió en un modo de inferencia científica

El método de Metropolis de 1953 utilizó muestreo aleatorio y reglas de aceptación para explorar computacionalmente distribuciones de equilibrio. Un modelo podía investigarse ahora mediante experimentos ejecutables repetidos, creando evidencia numérica donde las soluciones en forma cerrada eran inaccesibles.

Las redes se convirtieron en estructuras computacionales entrenables

McCulloch y Pitts mostraron que redes de unidades simplificadas podían implementar funciones lógicas. El perceptrón de Rosenblatt de 1958 añadió una regla adaptativa de aprendizaje, haciendo que el rendimiento dependiera del ajuste de pesos impulsado por datos y no solo de lógica codificada manualmente.

Las representaciones internas podían aprenderse a través de capas

La retropropagación de 1986 hizo entrenables en la práctica redes multicapa para muchas tareas al propagar información de error a través de conexiones diferenciables. El cambio epistémico importante fue que algunas características intermedias útiles ya no tenían que ser especificadas de antemano por el programador.

Los materiales se convirtieron en infraestructura computacional

Los dispositivos semiconductores transformaron desde finales de la década de 1940 la densidad, velocidad y fiabilidad de conmutación. Este hilo material, anticipado en PIO-074, hizo físicamente viables grandes redes digitales y, más tarde, sistemas de aprendizaje automático.

¿Por qué pudo surgir esta capacidad?

La transición requirió máquinas programables, teoría de la información, probabilidad, análisis numérico, electrónica de semiconductores, memorias mayores, conjuntos de datos y comunidades dispuestas a tratar los experimentos numéricos como herramientas científicas.

También se nutrió de la cibernética, metáforas neurocientíficas, investigación operativa y aprendizaje estadístico.

¿Qué formas posteriores de conocimiento hizo posibles?

La simulación se volvió central en modelado climático, de materiales, molecular, de ingeniería y económico. El aprendizaje automático permitió reconocimiento de patrones, predicción y sistemas de lenguaje y visión, mientras los modelos de redes se convirtieron en herramientas para estudiar comunicación, biología y sistemas sociales.

Nuevos límites y riesgos

Una simulación hereda los supuestos de su modelo, método numérico y condiciones iniciales. El aprendizaje automático puede ajustar correlaciones sin comprensión causal, amplificar sesgos, filtrar datos sensibles y producir resultados cuya base interna sea difícil de interpretar. Los modelos grandes también concentran recursos computacionales y consumo energético. Por ello, el rendimiento predictivo no puede sustituir validación, procedencia ni rendición de cuentas.

¿Qué cambió en la arquitectura del conocimiento?

El ordenador se convirtió en un **medio experimental para modelos ejecutables y representaciones adaptativas**. El conocimiento podía generarse ejecutando un sistema formal, comparando sus salidas con datos y cambiando iterativamente parámetros o pesos aprendidos.

Pérdida, exclusión e incertidumbre

Las narrativas actuales pueden proyectar la «IA» moderna hacia atrás sobre cualquier modelo de redes anterior. Las neuronas de McCulloch-Pitts, los perceptrones, la simulación Monte Carlo y la retropropagación resolvían problemas distintos y surgieron en comunidades diferentes. El aprendizaje automático actual es una síntesis histórica estratificada, no una única invención ni un camino inevitable.

Mapa de dependencias

- INDIRECTO ← PIO-074 Redes eléctricas y sistemas tecnológicos
- DIRECTO ← PIO-082 Turing y la computación programable
- DIRECTO ← PIO-084 Shannon y la información cuantificable
- INDIRECTO → PIO-091 Ciencia del clima y conocimiento de un planeta acoplado

Referencias

1. McCulloch, W. S. & Pitts, W. «A Logical Calculus of the Ideas Immanent in Nervous Activity.» *Bulletin of Mathematical Biophysics* 5, 115–133 (1943). DOI: 10.1007/BF02478259.
2. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. «Equation of State Calculations by Fast Computing Machines.» *Journal of Chemical Physics* 21(6), 1087–1092 (1953). DOI: 10.1063/1.1699114.
3. Rosenblatt, F. «The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.» *Psychological Review* 65(6), 386–408 (1958). DOI: 10.1037/h0042519.
4. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. «Learning Representations by Back-propagating Errors.» *Nature* 323, 533–536 (1986). DOI: 10.1038/323533a0.