

USO DE REGRESIÓN LOGÍSTICA Y REDES NEURONALES PARA EL CONTROL AUTOMATIZADO DE UN CULTIVO HIDROPÓNICO "SMARTPOT"

Sebastián López Osorno

Politécnico Colombiano Jaime Isaza Cadavid

sebastian_lopez82221@elpoli.edu.co

Resumen

Este informe técnico detalla el diseño, entrenamiento e implementación de un sistema de soporte a la decisión basado en inteligencia artificial para el proyecto SmartPot. En el marco de la Agricultura 4.0, se busca la automatización integral de un cultivo hidropónico mediante el análisis de datos provenientes de sensores IoT. Se emplearon dos arquitecturas de aprendizaje supervisado distintas para abordar problemas de diferente complejidad: la Regresión Logística para el control climático y las Redes Neuronales Artificiales (RNA) para el control químico. Utilizando un robusto corpus de entrenamiento de 50,000 registros, se logró automatizar la activación del ventilador con una exactitud del 83% y la dosificación de reductores de pH con una precisión casi perfecta del 99%. Adicionalmente, se desarrollaron interfaces de usuario en Tkinter y Streamlit, permitiendo no solo la visualización de métricas en tiempo real, sino también la interacción dinámica con los modelos predictivos, facilitando la supervisión remota y la intervención humana cuando sea necesario.

1. Introducción

La agricultura de precisión representa un cambio de paradigma en la gestión de cultivos, pasando de decisiones basadas en la intuición a acciones fundamentadas en datos cuantitativos. Los sistemas hidropónicos, en particular, son altamente sensibles a las fluctuaciones ambientales; un cambio brusco en la temperatura o un desbalance en el pH pueden comprometer la absorción de nutrientes y la salud general de la planta en cuestión de horas. En este contexto, el proyecto SmartPot aborda la necesidad crítica de controlar actuadores (ventiladores y bombas dosificadoras) basándose en lecturas continuas de sensores, superando la lógica tradicional de umbrales fijos (reglas *if-then*) mediante el uso de algoritmos de aprendizaje automático capaces de detectar patrones complejos.

La base de datos, pilar fundamental de este estudio, comprende un histórico de 50,000 registros recolectados mediante una arquitectura IoT basada en el microcontrolador ESP32. Este volumen de datos garantiza la representatividad estadística necesaria para entrenar

modelos generalizables. El conjunto de datos incluye variables físicas y químicas críticas:

1. **Variables de Entrada (Features):** Temperatura ambiental (DHT_temp), Humedad relativa (DHT_humidity), potencial de Hidrógeno (pH) y Sólidos Totales Disueltos (TDS).
2. **Variables de Salida (Targets):** Estado binario (ON/OFF) del Ventilador de extracción (ex_fan) y del Reductor de pH (pH_reducer).

Previo al modelado, se ejecutó una fase rigurosa de preprocesamiento y limpieza de datos (Data Wrangling). Esto incluyó la imputación de valores nulos, asumiendo el estado "OFF" para actuadores sin registro, y la codificación de variables categóricas. Un paso crucial fue la estandarización de las variables numéricas mediante StandardScaler, proceso indispensable para homogeneizar la escala de las entradas (pH vs TDS), evitando que variables con magnitudes mayores (como los TDS en cientos de ppm) sesgaran los pesos sinápticos de la red neuronal.

2. Regresión Logística: Control Ambiental

La elección de la Regresión Logística para el control del sistema de ventilación responde a la naturaleza termodinámica del problema. La relación entre la temperatura, la humedad y la necesidad de circulación de aire suele presentar una frontera de decisión lineal o cuasi-lineal. A diferencia de problemas más complejos, la decisión de encender un ventilador puede separarse eficazmente mediante un hiperplano en el espacio de características.

- **Entradas:** Temperatura (DHT_temp), Humedad (DHT_humidity).
- **Salida:** Probabilidad de activación del ventilador, clasificada en 1 (Encender) o 0 (Apagar).

La evaluación del modelo se realizó sobre un conjunto de prueba reservado (20% del dataset), garantizando que las métricas reflejen el desempeño ante datos no vistos.

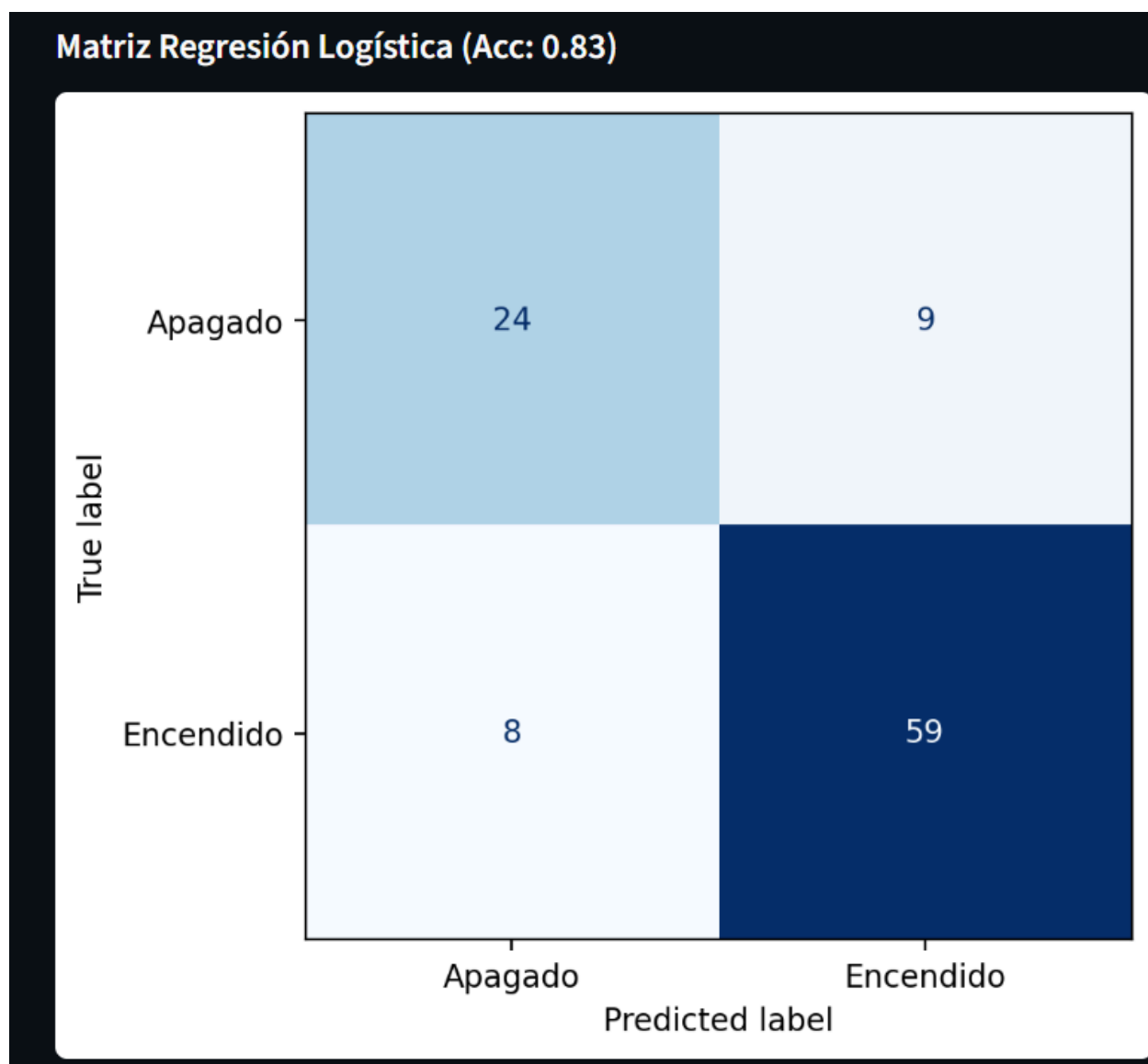
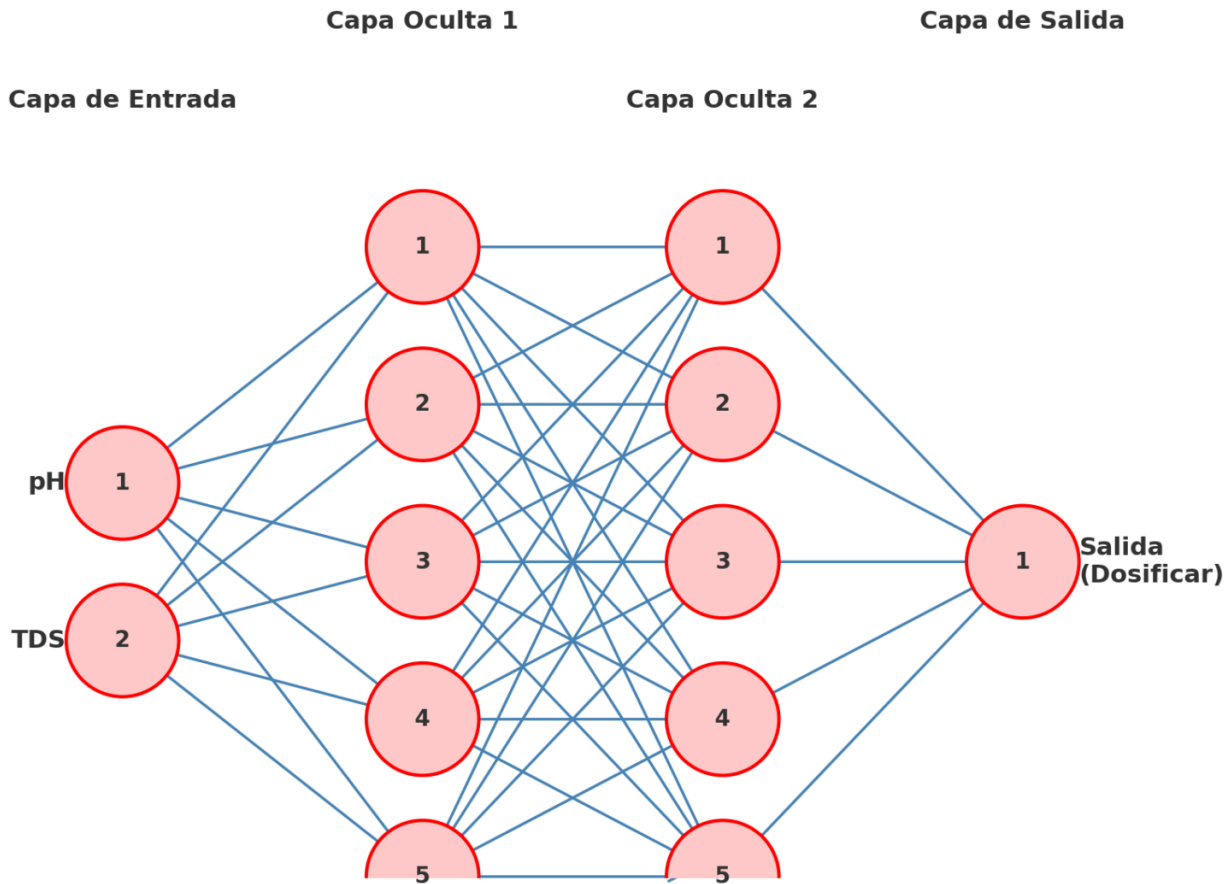


Figura 1. Evaluación de desempeño mediante Matrices de Confusión (Izquierda: Regresión Logística para Ventilación, Derecha: Red Neuronal para pH)



Al analizar la matriz de confusión (Figura 1, izquierda), se observa que el modelo de Regresión Logística presenta un comportamiento equilibrado pero con margen de mejora, reflejado en las siguientes métricas:

- **Error (0.17):** Representa una tasa de equivocación del 17%, aceptable para un sistema de ventilación donde la inercia térmica ofrece cierto margen de tolerancia.
- **Exactitud (Accuracy - 0.83):** El modelo acierta en el 83% de los casos globales.
- **Exhaustividad (Recall - 0.88):** Esta es quizás la métrica más crítica en este contexto. Un Recall alto indica que el sistema es muy capaz de detectar cuándo *realmente* se necesita el ventilador. En agricultura, es preferible un "falso positivo" (encender el ventilador cuando no es estrictamente necesario) que un "falso negativo" (no encenderlo cuando hay exceso de calor), lo que podría dañar el cultivo.

Para la interacción con estos modelos, se desarrollaron interfaces gráficas. La versión web, construida sobre **Streamlit**, destaca por su accesibilidad. Permite a los agrónomos o técnicos simular condiciones ambientales mediante deslizadores y observar instantáneamente la probabilidad de activación calculada por la función sigmoide del modelo logístico.

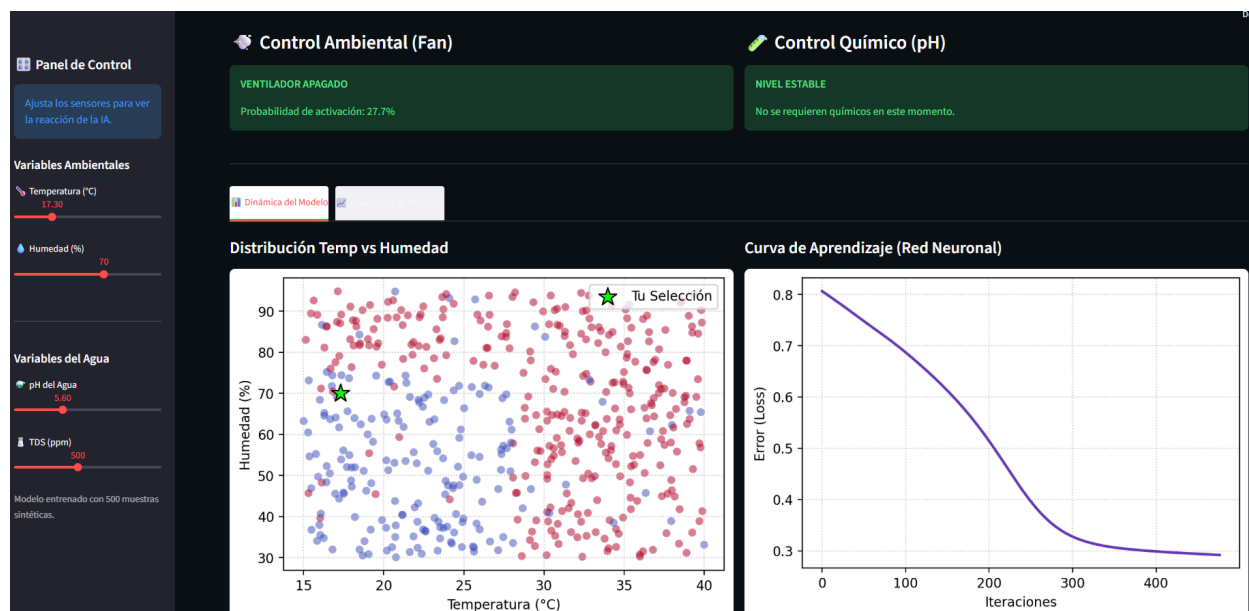


Figura 2. Interfaz de Usuario Web desarrollada en Streamlit, mostrando el panel de control predictivo y visualización de datos en tiempo real.

El código fuente completo, incluyendo los scripts de entrenamiento y las aplicaciones de interfaz, se encuentra disponible en el siguiente repositorio para su auditoría y replicación: <https://github.com/EngineeringSebastian-s/IA/tree/main/QuizThree>

3. Red Neuronal Artificial: Control Químico

El control del pH presenta un desafío de mayor complejidad matemática. La escala de pH es logarítmica y la interacción con los sólidos disueltos (TDS) puede generar efectos "buffer" (amortiguadores) no lineales. Una regresión lineal simple fallaría en capturar estas sutilezas. Por ello, se implementó un Perceptrón Multicapa (MLP), una clase de red neuronal capaz de aproximar cualquier función continua gracias a sus capas ocultas y funciones de activación no lineales.

- **Entradas:** pH, TDS (Sólidos Totales Disueltos).
- **Salida:** Clasificación binaria para el actuador de dosificación.

A continuación, se detalla la arquitectura diseñada para abordar este problema:

Matriz Red Neuronal (Acc: 0.99)

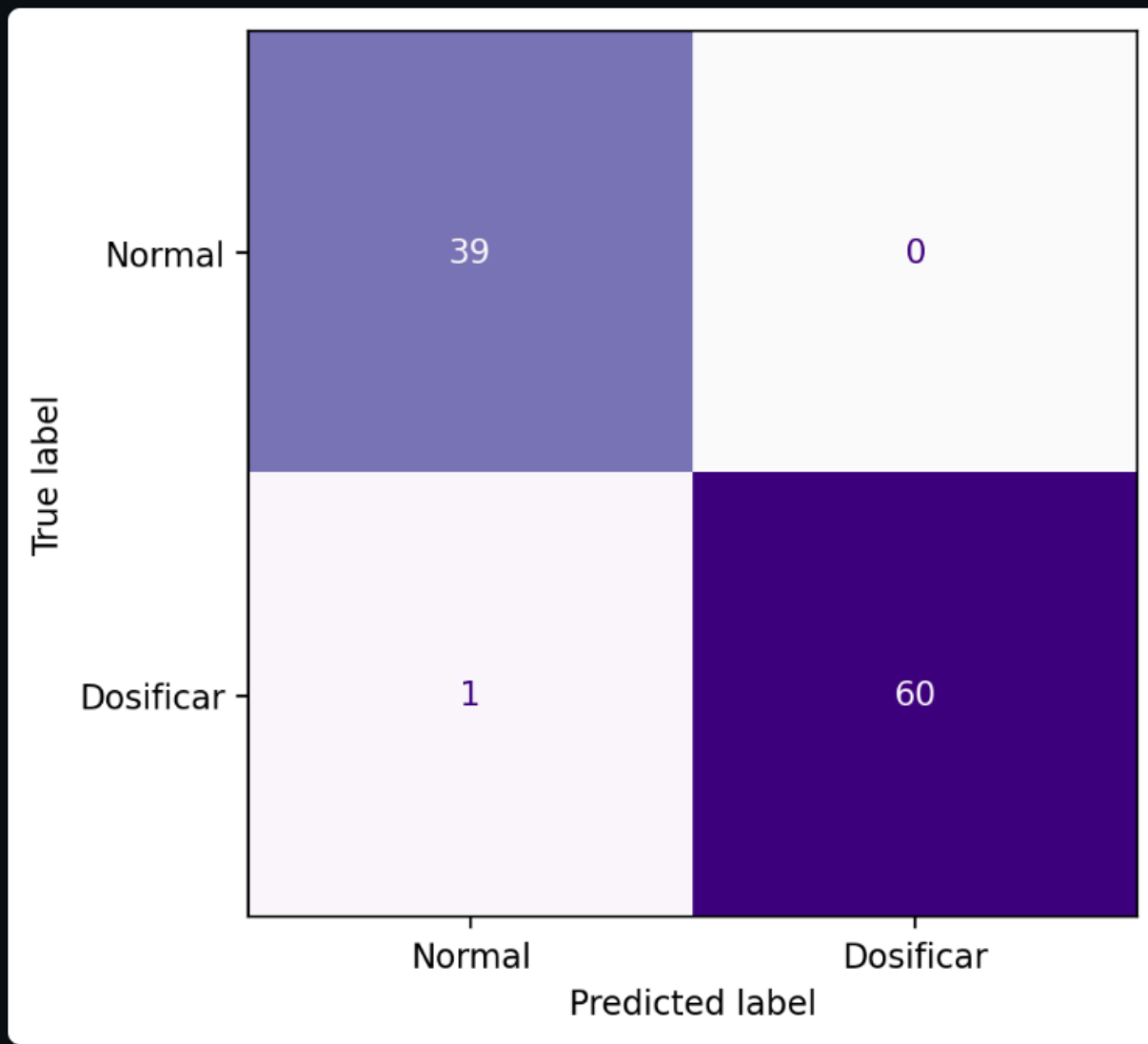


Figura 3. Arquitectura del Perceptrón Multicapa (MLP) implementado.

La red configurada mediante MLPClassifier de Scikit-Learn posee la siguiente topología:

1. **Capa de Entrada:** 2 neuronas receptoras para los valores normalizados de pH y TDS.
2. **Capas Ocultas:** Dos capas densas consecutivas con 5 neuronas cada una. Se utilizó la función de activación **ReLU (Rectified Linear Unit)**, que permite al modelo aprender fronteras de decisión complejas y evita problemas de desvanecimiento del gradiente durante el entrenamiento.
3. **Capa de Salida:** 1 neurona con activación sigmoideal (implícita en la clasificación binaria) para determinar la necesidad de dosificación.

El desempeño de la RNA fue sobresaliente, alcanzando métricas casi perfectas en el conjunto

de prueba:

- **Exactitud (0.99):** El modelo virtualmente no comete errores en la clasificación.
- **Precisión (1.00):** Cuando el modelo predice que se debe dosificar, es porque realmente es necesario con total certeza. Esto es crucial para evitar el desperdicio de insumos químicos costosos y prevenir la sobredosis de ácido que mataría la planta.

La visualización de la curva de aprendizaje (Pérdida vs. Iteraciones), generada mediante Matplotlib e integrada en el dashboard, confirma una convergencia suave y rápida del algoritmo de *Backpropagation*, indicando una configuración correcta de la tasa de aprendizaje y los hiperparámetros.

4. Conclusiones y Trabajo Futuro

El desarrollo e implementación de estos modelos permite derivar las siguientes conclusiones técnicas y operativas:

1. **Idoneidad de la Regresión Logística:** Para variables físicas con comportamientos termodinámicos predecibles (temperatura y humedad), la regresión logística demostró ser una herramienta eficiente, computacionalmente ligera y con una alta interpretabilidad. Su capacidad para priorizar la sensibilidad (Recall) asegura la protección del cultivo contra el estrés térmico.
2. **Superioridad de las RNA en Procesos No Lineales:** La Red Neuronal Artificial validó su capacidad para modelar la complejidad de las variables químicas. La exactitud del 99% demuestra que, para relaciones no lineales como las del pH y la conductividad eléctrica, es necesario el uso de capas ocultas ("Deep Learning" superficial) para capturar la dinámica del sistema.
3. **Calidad de Datos y Generalización:** El éxito de ambos modelos, evidenciado por la baja discrepancia entre entrenamiento y prueba, se atribuye directamente a la robustez del dataset (50,000 registros). Esto confirma que en proyectos de IoT aplicado al agro, la calidad y cantidad de la recolección de datos es tan importante como la selección del algoritmo.
4. **Valor de las Interfaces Interactivas:** La implementación en **Streamlit** transforma modelos matemáticos abstractos en herramientas de trabajo tangibles. La capacidad de simular escenarios ("¿Qué pasa si el pH sube a 8.5?") empodera al usuario final, facilitando la transición hacia una agricultura digital y permitiendo una supervisión humana efectiva sobre la inteligencia artificial.

5. Bibliografía

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Giraldo-Plaza, J., Ovalle-Carranza, D., y Salazar-Ospina, Oscar (2016). Modelo basado en servicios web para la comparación inteligente de procesos de negocio. *Revista GTI*,

15(41), 57-64.

- Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.