

Redes Neuronales

Introducción completa para alumnos familiarizados con IA tradicional

Jesús Emmanuel Martínez García

Introducción

- Una **red neuronal** es un sistema computacional inspirado en las redes neuronales biológicas.
- Aprende funciones a partir de datos, capturando **relaciones complejas y no lineales**.
- Ventajas sobre modelos tradicionales:
 - Representación automática de características
 - Escalabilidad con grandes volúmenes de datos
 - Capacidad para resolver problemas complejos



Historia y Hitos Clave

- **1943** – McCulloch & Pitts: primer modelo de neurona artificial.
- **1949** – Hebb: regla de aprendizaje Hebbiana.
- **1958** – Rosenblatt: perceptron.
- **1969** – Minsky & Papert: limitaciones del perceptron.
- **1986** – Rumelhart, Hinton & Williams: backpropagation.
- **1998** – LeCun: LeNet.
- **2006** – Hinton et al.: Deep Belief Nets.
- **2012** – Krizhevsky: AlexNet, revolución con GPUs.

Conceptos Fundamentales

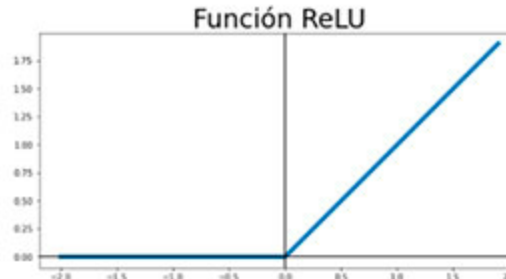
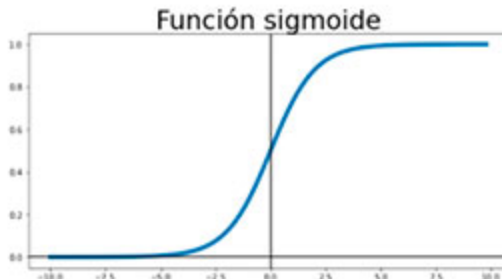
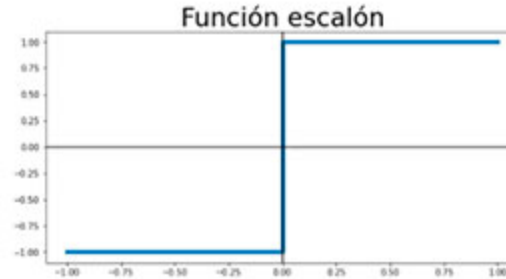
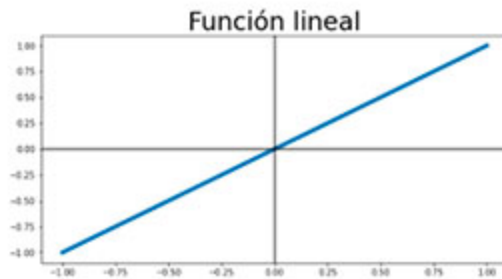
- **Neurona artificial (perceptrón)**
 - $z = w \cdot x + b \rightarrow a = \phi(z)$
 - Entradas (x), pesos (w), sesgo (b), función de activación (ϕ)
- **Capa**
 - Entrada, oculta(s), salida
 - Conectividad: fully-connected, convolucional, recurrente

- **Arquitectura**

- Disposición y tipo de capas + funciones de activación + entrenamiento

- **Funciones de activación**

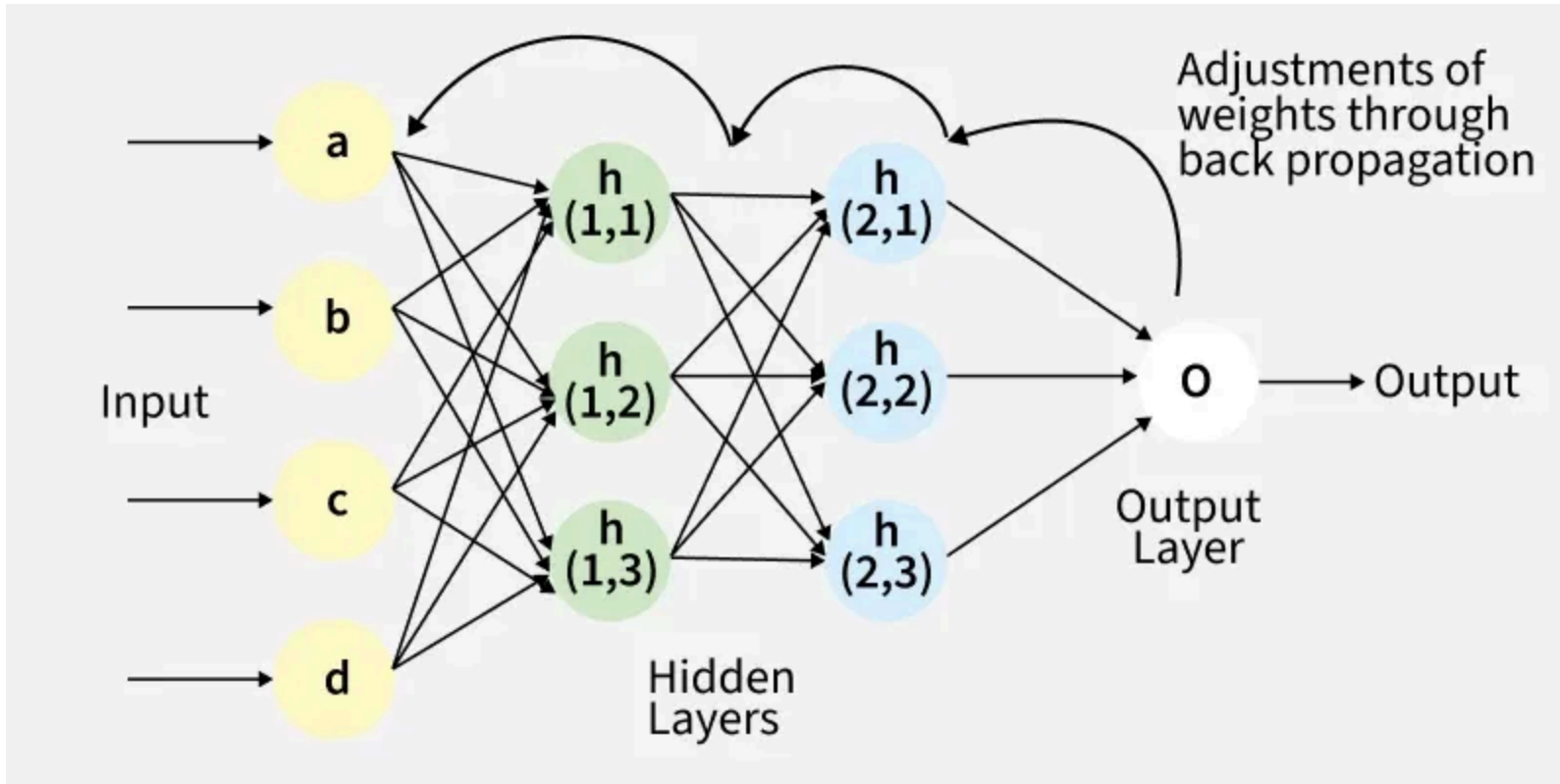
- Lineal, Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax



Entrenamiento y Backpropagation

- **Forward pass** → cálculo de salida
- **Cálculo de pérdida** → función objetivo
- **Backpropagation** → ajuste de pesos mediante gradiente descendente
- Optimizadores: SGD, Momentum, RMSprop, Adam
- Regularización: L1/L2, Dropout, Batch Normalization, Early stopping

Backpropagation

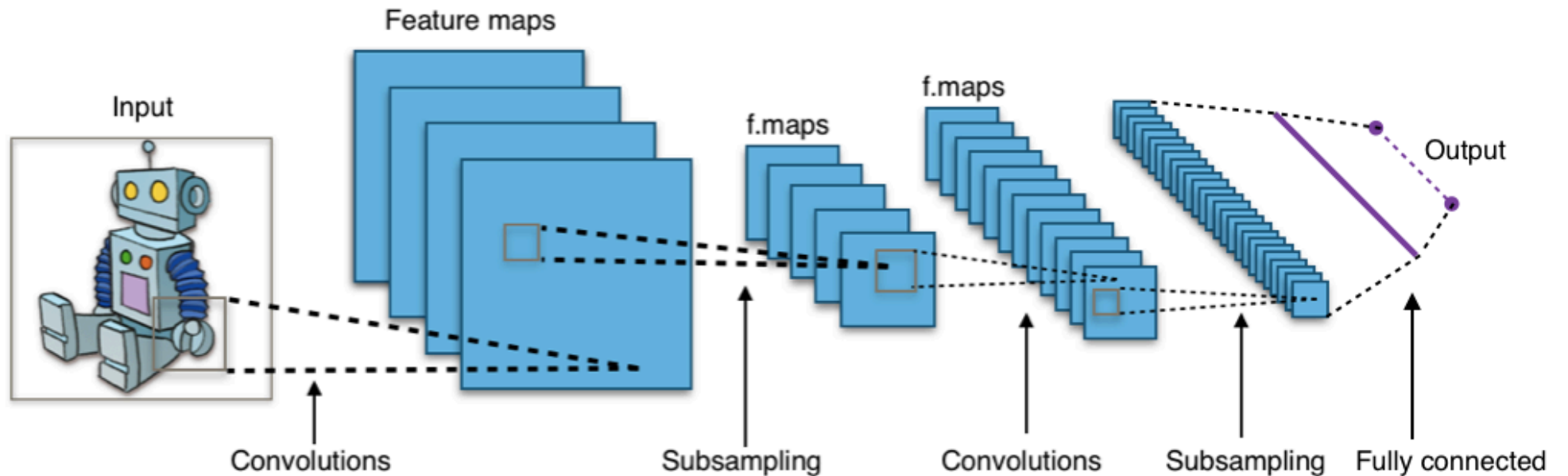


Arquitecturas Comunes

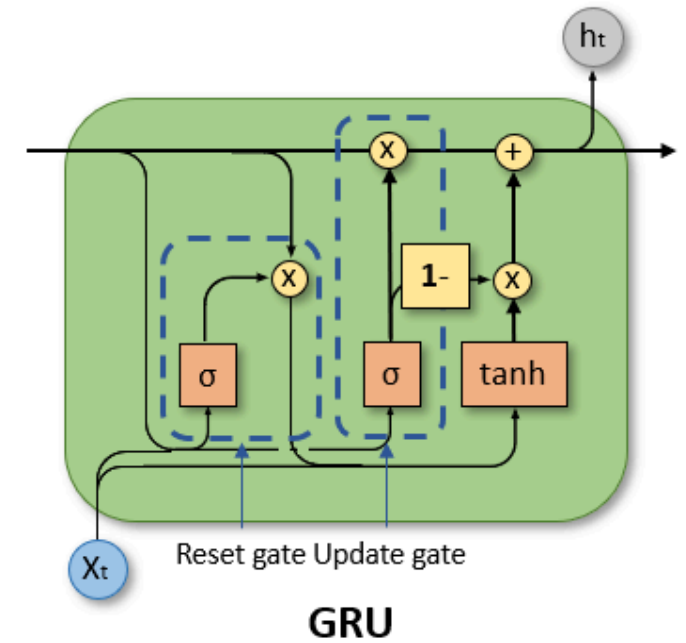
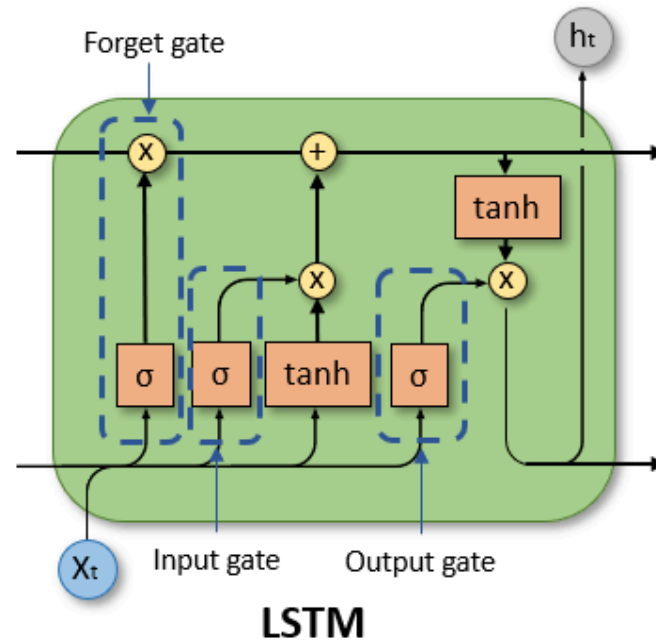
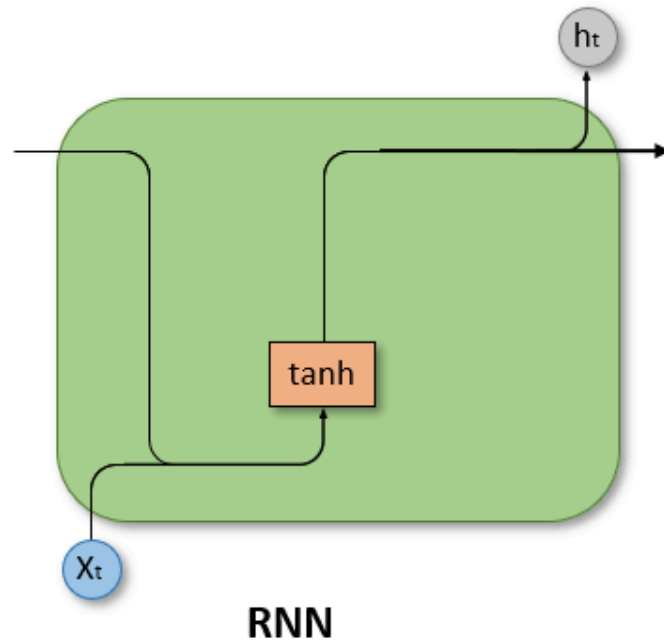
- MLP (Multi-Layer Perceptron)
 - Datos tabulares, clasificación/regresión

- **CNN (Convolutional Neural Network)**

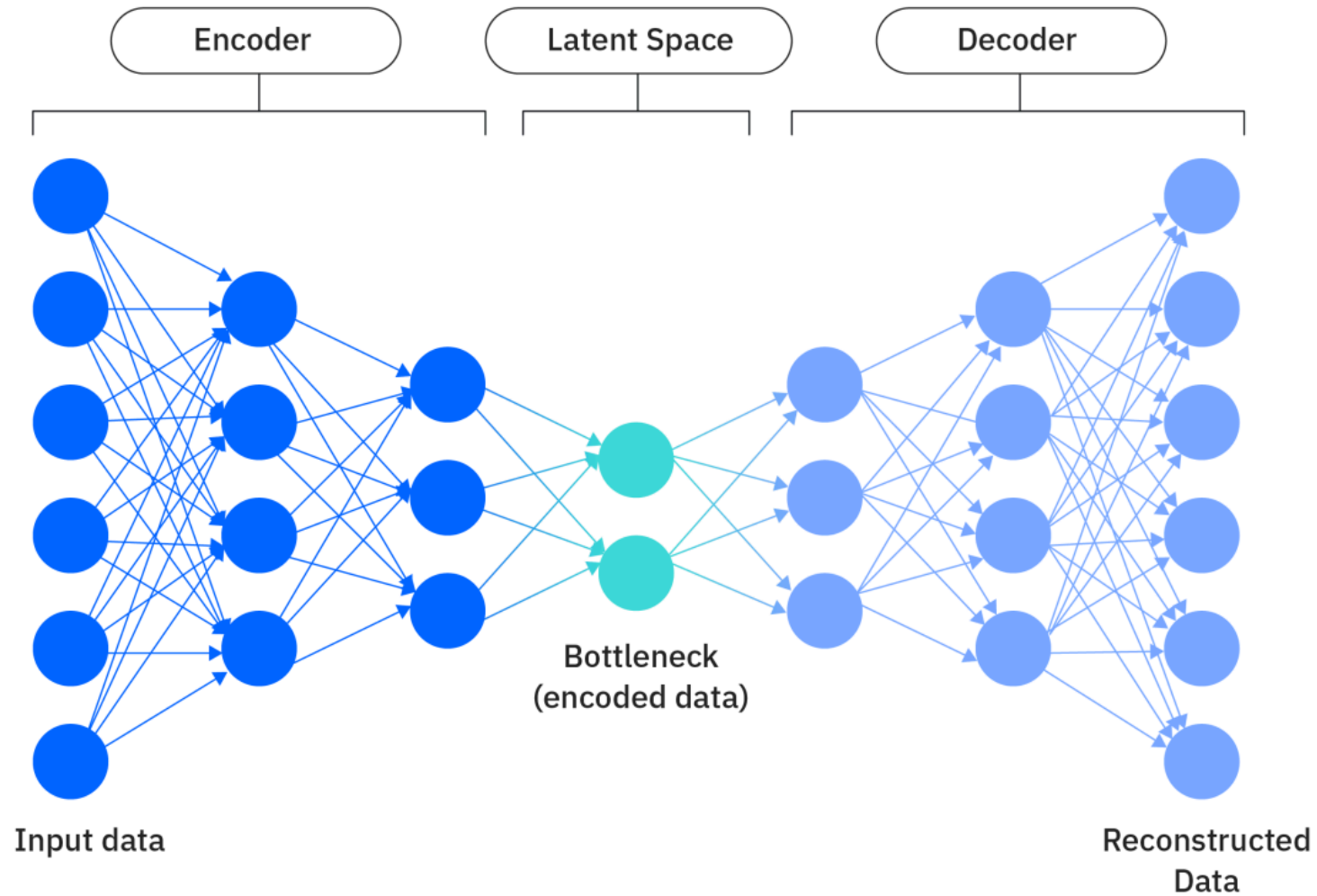
- Imágenes, audio, visión por computadora
- Bloques: convolución → activación → pooling → fully connected



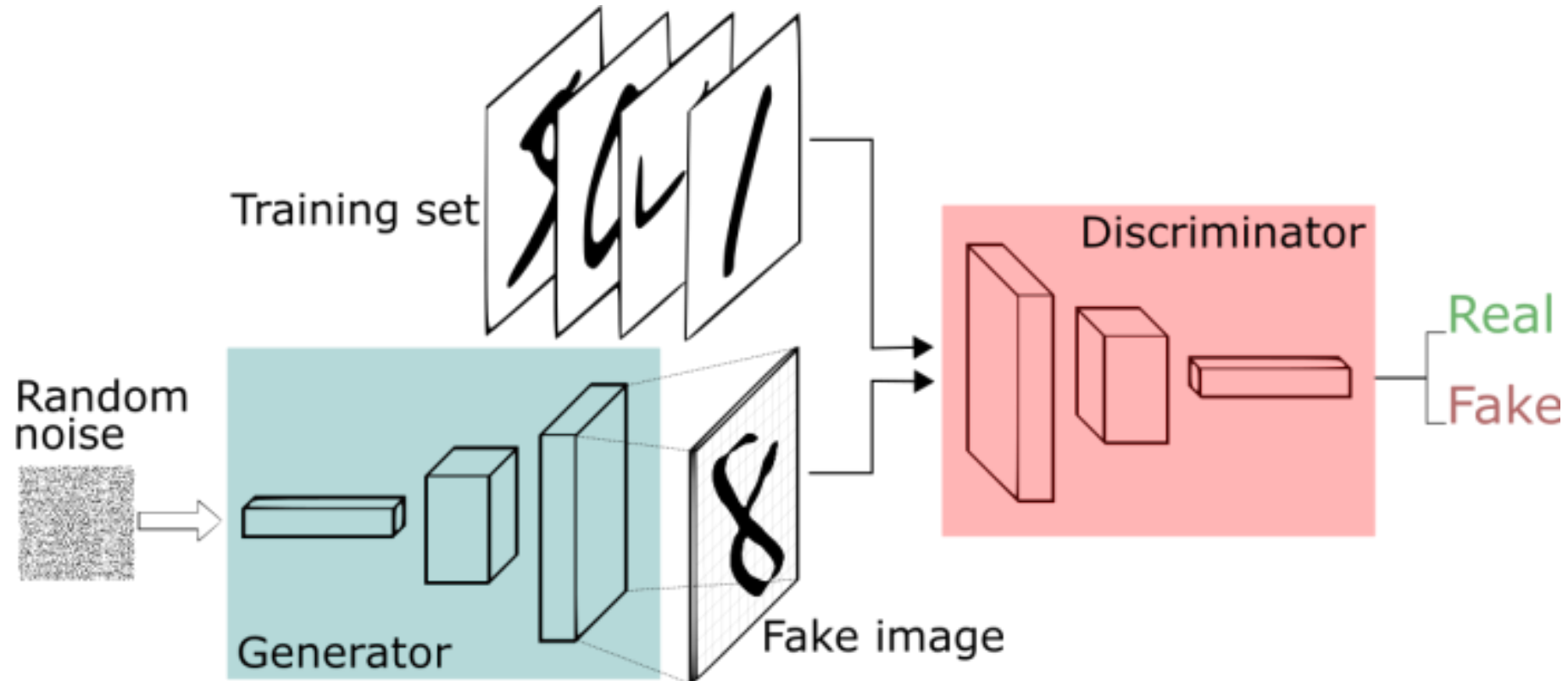
- RNN, LSTM, GRU → series temporales, secuencias, texto



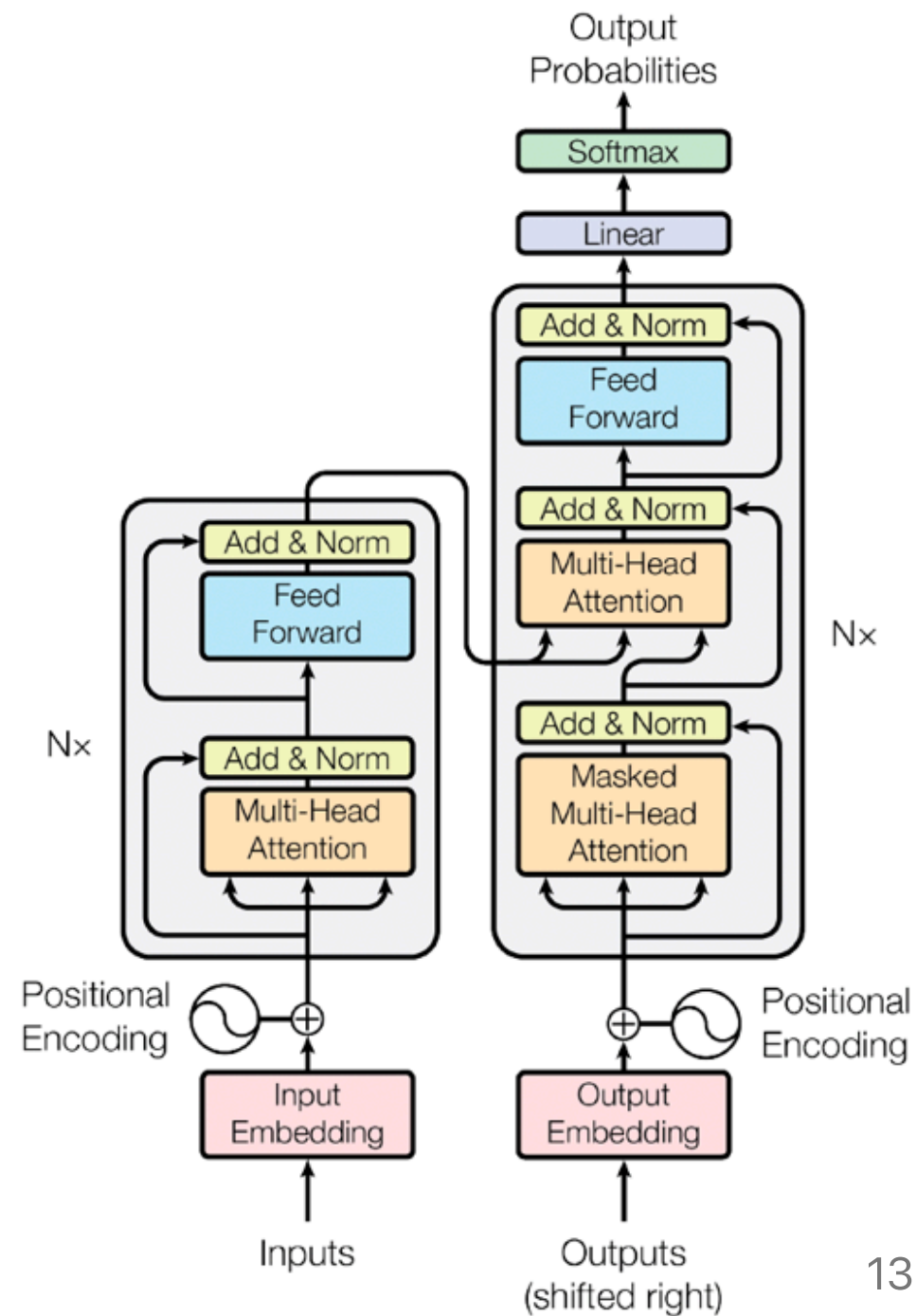
- **Autoencoders** → compresión y detección de anomalías



- **GANs** → generación de datos (imágenes, augmentations)



- **Transformers y atención** → NLP, visión (ViT, BERT, GPT)





Matemáticas Esenciales

- Derivadas parciales y gradiente
- Regla de la cadena aplicada al backpropagation
- Funciones de pérdida y sus derivadas (MSE, cross-entropy)
- Ejemplo numérico: forward + backward de red 2-1-1



Entrenamiento Práctico

- Preparación de datos: normalización, augmentations
- División: train / val / test
- Parámetros: batch size, epochs, learning rate
- Callbacks: early stopping, checkpoints
- Validación cruzada si aplica



Métricas y Evaluación

- Clasificación: accuracy, precision, recall, F1, ROC-AUC
- Regresión: RMSE, MAE, R^2
- Métricas avanzadas: mAP, BLEU/ROUGE, perplexity

Problemas Comunes

- Overfitting vs underfitting
- Vanishing / exploding gradients
- Saturación de activaciones
- Learning rate inapropiado

Interpretabilidad

- Saliency maps, Grad-CAM, LIME, SHAP
- Consideraciones éticas y explicabilidad del modelo

Recursos y Herramientas

- Frameworks: TensorFlow/Keras, PyTorch, JAX
- Librerías: scikit-learn, alumentations, Hugging Face
- Plataformas: Colab, Kaggle, AWS/GCP/Azure



Ejemplos de Flujo de Trabajo

- Clasificación de imágenes con CNN
- Predicción de series temporales con LSTM/GRU
- Transfer learning con redes preentrenadas (ResNet, EfficientNet)

Bibliografía

- McCulloch & Pitts (1943)
- Rosenblatt (1958)
- Minsky & Papert (1969)
- Rumelhart, Hinton & Williams (1986)
- LeCun (1998)
- Hinton et al. (2006)
- Krizhevsky et al. (2012)