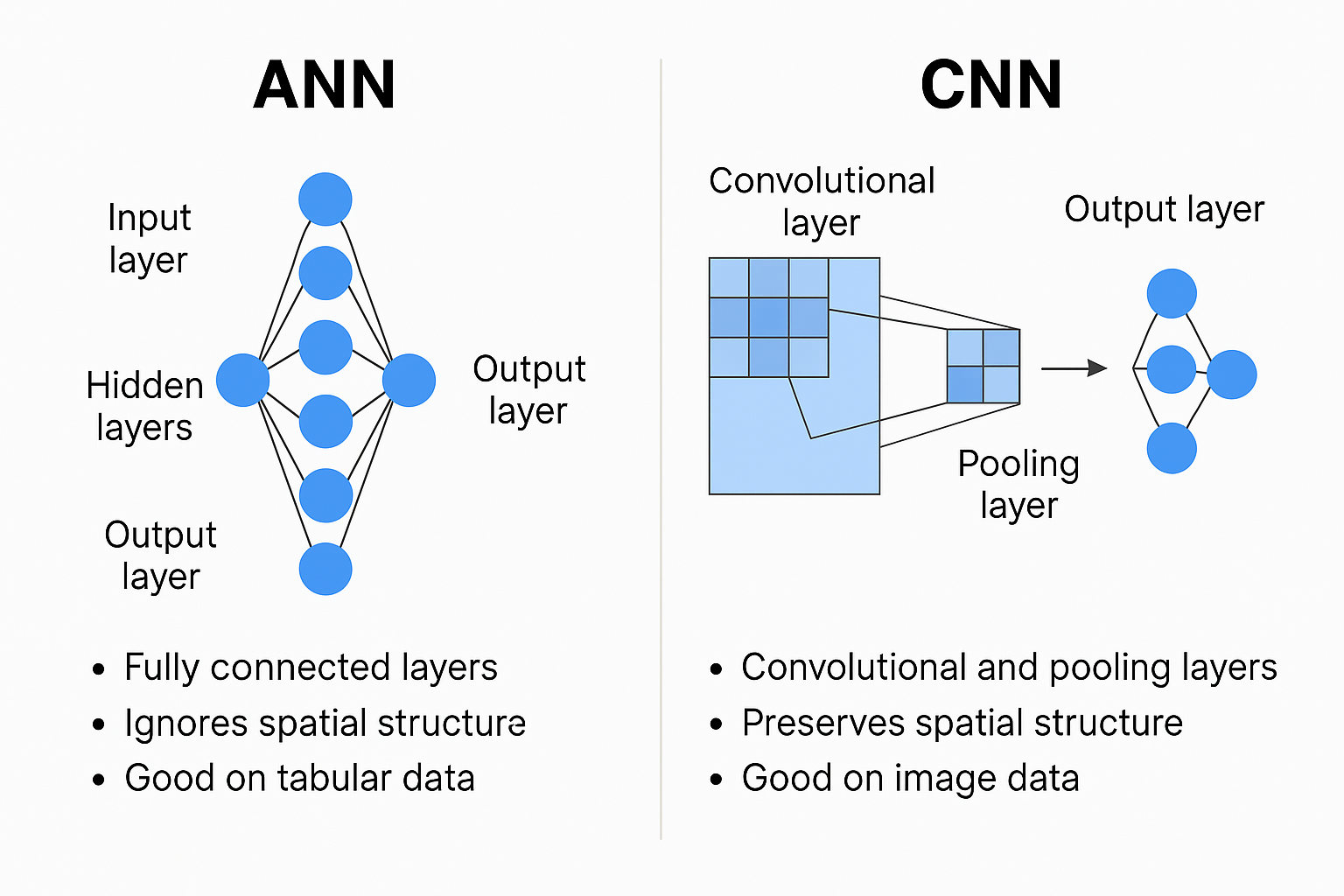
======================================================================

**ANN vs CNN**

======================================================================



1. **Definition**

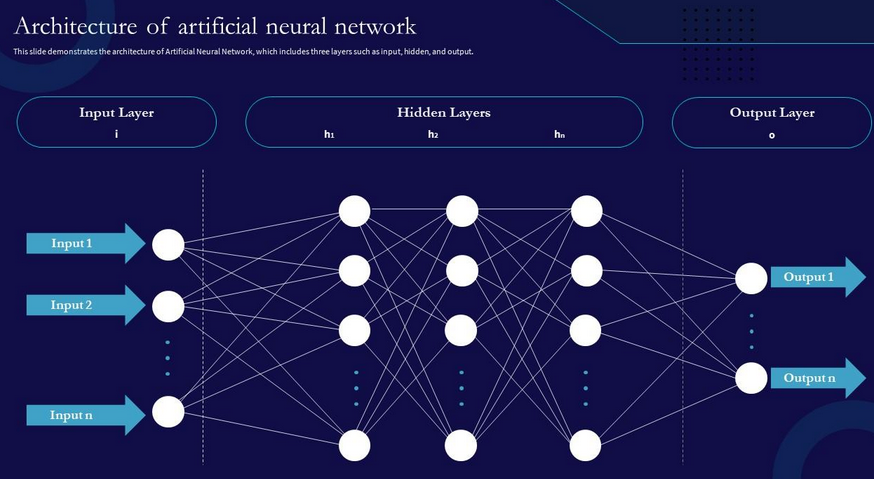
| **Topic** | **ANN (Artificial Neural Network)** | **CNN (Convolutional Neural Network)** |
| --- | --- | --- |
| Purpose | General-purpose network for tabular or structured data | Specialized for image, video, or spatial data |
| Structure | Fully connected layers | Convolution + Pooling layers followed by dense layers |

**2. Architecture Differences**

A. ANN Architecture

* Input layer → Hidden layers (Dense) → Output layer
* Every neuron is connected to every neuron in the previous and next layer (Fully Connected).
* Does not preserve spatial structure.

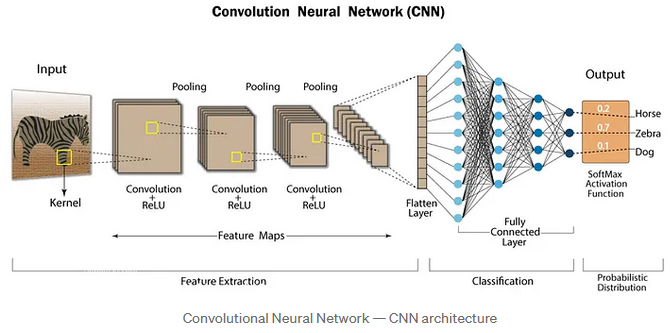
Example: Flatten → Dense(128) → Dense(64) → Dense(1 or Softmax)



B. CNN Architecture

* Input layer (image) → Convolutional layer(s) → Pooling layer(s) → Fully connected layer(s) → Output
* Uses filters (kernels) to detect features like edges, shapes, textures.
* Preserves spatial relationships (e.g., location of a feature in an image).

Example: Conv2D → MaxPooling → Conv2D → Flatten → Dense → Output



**3. Feature Extraction**

| Feature | ANN | CNN |
| --- | --- | --- |
| Manual Feature Engineering | Often required | Automatically learns spatial features |
| Input Shape | 1D (requires flattening) | 2D or 3D (e.g., images with height, width, channels) |
| Translation Invariance | No | Yes (due to convolution and pooling) |

**4. Parameters and Computation**

| Feature | ANN | CNN |
| --- | --- | --- |
| Parameters | Large (dense layers = high number of weights) | Fewer (due to weight sharing in filters) |
| Computation | High for image data | Optimized for image data |
| Overfitting | High risk on large input like images | Less risk due to fewer parameters |

**5. Use Cases**

| ANN | CNN |
| --- | --- |
| Tabular data (house prices, customer churn) | Image classification (cats vs dogs) |
| Time series (with preprocessing) | Object detection, face recognition |
| Text data (after vectorization) | Medical imaging, handwriting recognition |

**6. Performance**

| Aspect | ANN | CNN |
| --- | --- | --- |
| Accuracy on image tasks | Usually poor | Excellent |
| Training speed | Slower for images | Faster due to shared weights |
| Feature learning | Shallow | Hierarchical (learns from edges → patterns → objects) |

**7. Advantages**

ANN

* Simpler architecture for general tasks.
* Good for structured and small datasets.

CNN

* Automatically learns spatial hierarchies.
* Excellent at detecting features like edges, textures, shapes.
* Efficient for high-dimensional inputs (like 2D/3D images).

**Visual Analogy**

* ANN: Like every person in a room talking to everyone else.
* CNN: Like a person scanning small patches of an image (reusing the same “eye” or filter) and remembering important features.