

Tipos de capas en redes neuronales

En el **diseño de redes neuronales artificiales** existen diferentes tipos de capas, cada una con un propósito y funcionamiento específico. El uso y combinación de estos tipos de capas permiten que las redes aprendan patrones complejos a partir de los datos. A continuación, te presento un resumen de los **tipos principales de capas** que puedes encontrar en redes neuronales modernas, tanto para visión, procesamiento de lenguaje, tabulares, etc.:

1. Capa Densa o Fully Connected (Dense Layer)

- **Descripción:** Cada neurona está conectada a todas las neuronas de la capa anterior.
- **Usos:** Clasificación, regresión, redes MLP.
- **Ejemplo (Keras):**

```
from tensorflow.keras.layers import Dense
Dense(128, activation='relu')
```

2. Capa de Convolución (Convolutional Layer)

- **Descripción:** Extraen características espaciales usando filtros (kernels) que barren la entrada.
- **Usos:** Visión por computador, procesamiento de imágenes y señales.
- **Variantes:** Conv1D, Conv2D, Conv3D.
- **Ejemplo:**

```
from tensorflow.keras.layers import Conv2D
Conv2D(32, kernel_size=(3,3), activation='relu')
```

3. Capa de Pooling (Pooling Layer)

- **Descripción:** Reduce la dimensión de los datos y la cantidad de parámetros, manteniendo la información más importante.
- **Tipos:** MaxPooling, AveragePooling.
- **Ejemplo:**

```
from tensorflow.keras.layers import MaxPooling2D  
MaxPooling2D(pool_size=(2,2))
```

4. Capa de Normalización (Normalization Layer)

- **Descripción:** Mejora la estabilidad y velocidad del entrenamiento normalizando la activación.
- **Tipos:** BatchNormalization, LayerNormalization.
- **Ejemplo:**

```
from tensorflow.keras.layers import BatchNormalization  
BatchNormalization()
```

5. Capa Recurrente (Recurrent Layer)

- **Descripción:** Procesan datos secuenciales manteniendo información en el tiempo.
- **Tipos:** SimpleRNN, LSTM, GRU.
- **Usos:** Procesamiento de lenguaje, series temporales.
- **Ejemplo:**

```
from tensorflow.keras.layers import LSTM  
LSTM(64)
```

6. Capa de Dropout

- **Descripción:** Técnica de regularización que “apaga” aleatoriamente neuronas en el entrenamiento, evitando overfitting.
- **Ejemplo:**

```
from tensorflow.keras.layers import Dropout
Dropout(0.5)
```

7. Capa de Embedding

- **Descripción:** Convierte datos categóricos o secuenciales (como palabras) en vectores densos de menor dimensión.
- **Usos:** NLP, recomendadores.
- **Ejemplo:**

```
from tensorflow.keras.layers import Embedding
Embedding(input_dim=10000, output_dim=64)
```

8. Capa de Flatten

- **Descripción:** "Aplana" un tensor de varias dimensiones en un vector (útil para conectar CNNs a capas densas).
- **Ejemplo:**

```
from tensorflow.keras.layers import Flatten
Flatten()
```

9. Capas de Activación

- **Descripción:** Aplican funciones como ReLU, sigmoid, softmax. Pueden ser independientes o estar incluidas en otras capas.
- **Ejemplo:**

```
from tensorflow.keras.layers import Activation
Activation('relu')
```

10. Capas Específicas (para arquitecturas avanzadas)

- **Attention Layer:** Implementa mecanismos de atención para destacar partes relevantes de la entrada.

- **Residual/Add Layer:** Permiten conexiones directas entre capas distantes (ResNet, transformers).
 - **Concatenate/Merge:** Combinan varias entradas o salidas.
 - **Capas Custom (Lambda):** Para operaciones personalizadas.
-

¿Cómo combinarlas?

La arquitectura depende de la tarea:

- **Visión por computador:** Conv2D + Pooling + Flatten + Dense.
- **Texto/series temporales:** Embedding + LSTM/GRU + Dense.
- **Tabulares:** Dense + Normalization + Dropout.