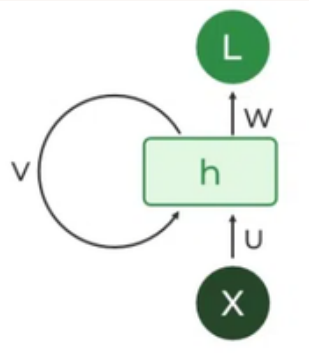
Types of Neural Networks

RNN

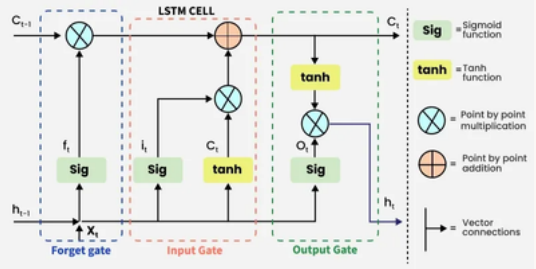
- A Recurrent Neural Network is a type of neural network that has a built-in memory and it remembers information from previous steps and uses it in future steps. The current prediction depends not just on the current input, but also on what the model has seen before. You can't parallelize the training easily.
- This makes RNNs slower to train than other models (like CNNs or Transformers).



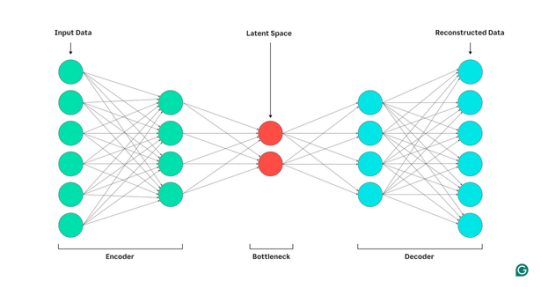
LSTM

Long Short Term Memory
LSTM is a special kind of neural network that's really good at remembering things for a long time better than a regular Long Short Term Memory
LSTM is a special kind of neural network that's really good at remembering things for a long time better than a regular RNN. It has three gates

1. Forget Gate - $f_t=\sigma(W_f \cdot [h_{t-1},x_t]+b_f)$ here, the forget gate looks at both the new input and what it already remembers. It then outputs a vector of numbers between 0 and 1:
   Values close to 1 → keep that memory
   Values close to 0 → forget that memory
2. Input Gate - Keeps what information should it keep a track of. It has two inputs, 1 from the forget gate and the other one is
   $\tilde{C}_t=\tanh(W_C \cdot [h_{t-1},x_t]+b_C)$. Here the goal is to add new information, reduce the impact of existing information or even remove some information.
3. Output gate: The output gate uses sigmoid to decide how much of the memory to show, and tanh to compress the memory into a manageable format.



Autoencoder



Encoder:

- Takes high-dimensional input (many features)
- Compresses it to fewer dimensions (latent representation)

Think of it as "learning the **essence** of the input"

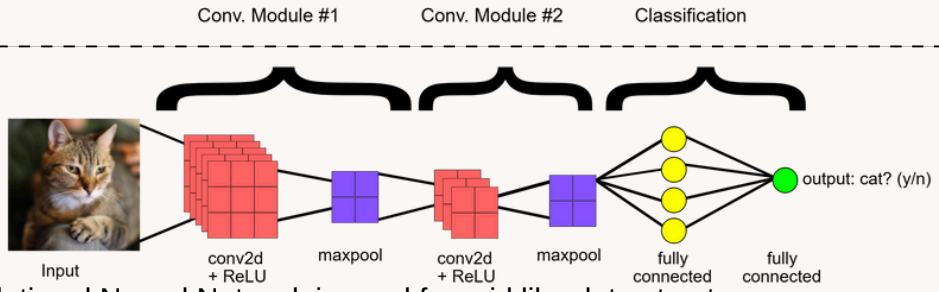
Bottleneck (Hidden Layer):

- The most compact version of the data
- Like squeezing info into a tight zip file

Decoder:

- Expands this compressed info back to the original form
- Reconstructs the input from the compressed "essence"

CNN



Convolutional Neural Network is used for grid like data structures typically for images. In this there are three layers

1. Convolution Layer
   - The CNN uses small filters (also called kernels) that slide over the image.
   - Each kernel is a small matrix (e.g., 3×3 or 5×5) of trainable weights.
   - As the kernel slides (this is called convolution), it multiplies its weights with the underlying pixels and sums the result to produce a feature map.
   - Each feature map learns to detect a specific feature (e.g., edges, textures, shapes).
   - Multiple kernels can be used in parallel to capture different features.
   - After convolution, we typically apply a non-linear function like ReLU to introduce non-linearity and help the network learn complex patterns.
2. Pooling (Subsampling) Layer
   - We shrink the picture by picking the most important number from small regions (Max Pooling) or taking an average.
   - This makes the image smaller and easier for the computer to handle.
3. Stacked Layers
   - Early layers learn basic things (edges), and deeper layers learn complex things (like eyes, wheels, or letters).
4. Flattening (Convert to 1D vector)
   - We stretch out the final smaller picture into a single row of numbers.

These numbers go into a normal neural network that decides what's in the image (e.g., cat, dog, car).