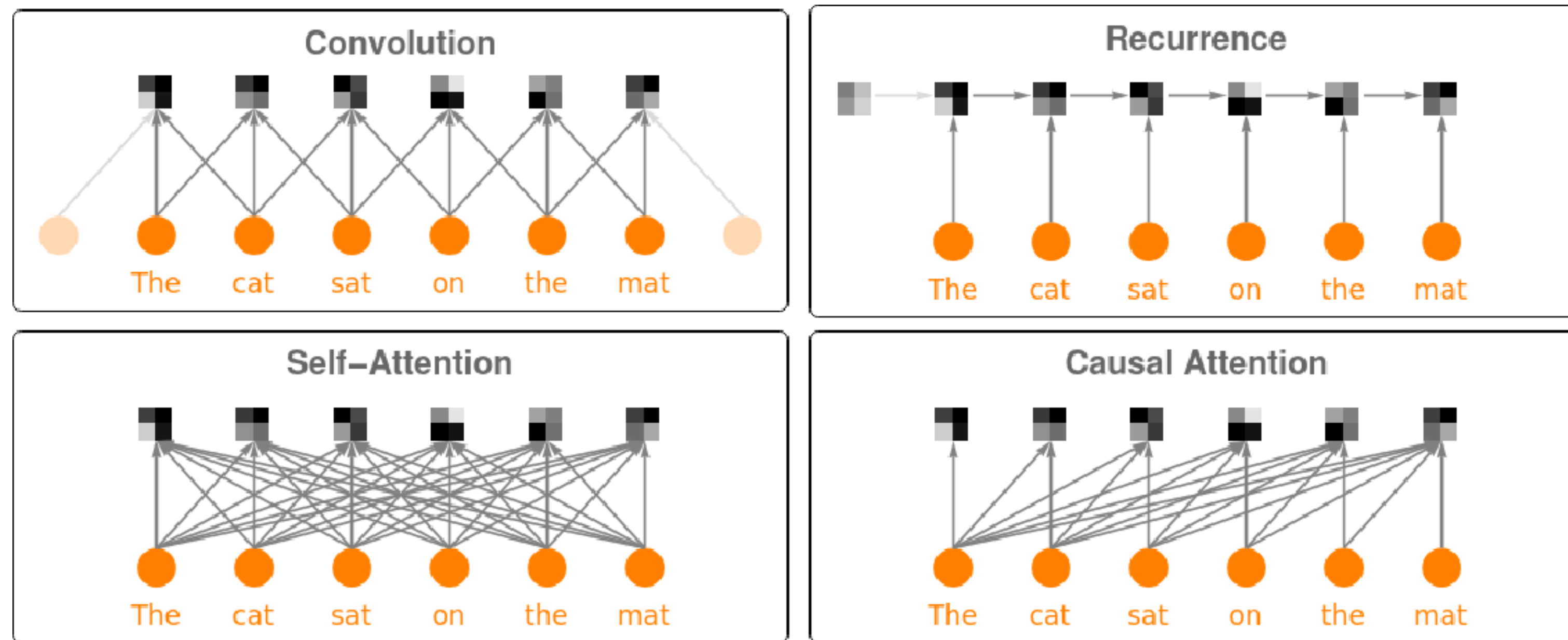


Neural language modeling: advantages

- Polysemy
 - Word2Vec and other static language models fail to capture polysemy. Having a single embedding per token limits our ability to capture the meaning of words. In fact, our training procedure, whereby identical words in different contexts are constrained to a single representation, introduces ambiguity into the model.
- Linearity
 - Word2Vec, and other similar static word embedding models such as GLOVE, are linear models. In contrast, neural networks are sequences of linear transformations separated by non-linearities; more layers translates to more representational power. Without non-linearities, no modeling capacity is gained by adding extra layers.
- Maximizing the context that a model can consider is an exercise that is largely computational in nature! The DL community has borrowed many of its most successful ideas from the field of discrete-time signal processing, and have built highly efficient numerical methods that utilize modern hardware (GPUs, TPUs).
 - CNNs
 - RNNs
- More recently (2017), a mechanism called **attention** has changed what is possible with language models. Transformer networks, based on this idea, **explicitly** represent contextual dependencies not only at the input layer but arbitrarily deep into the network. Transformers are the basis behind SOTA methods in NLP and vision! Methods based purely on attention come with the theoretical disadvantage of being able to only consider finite context windows; in practice we can make n large enough (10^3) for practical applications, and this context size ends up being larger than what is practical with convolutions and recurrent connections, anyway.

Neural language modeling: capturing context

- The three dominant classes of neural network architecture each differ in how context is captured:
 - Convolutional neural networks (CNNs)
 - Recurrent neural networks (RNNs)
 - Transformer networks (i.e., attention based)



Credit: Wolfram