

Neural Nets MVPs

Data Science Immersive

Objectives for today

- Identify how each of the most popular types of NNs is used
- Know what parameters to optimise in which functions
- Be aware of the data prep necessary for each type


True across types

While we think of these “models” they are actually “layers”. Multiple layer types can be used in one network.

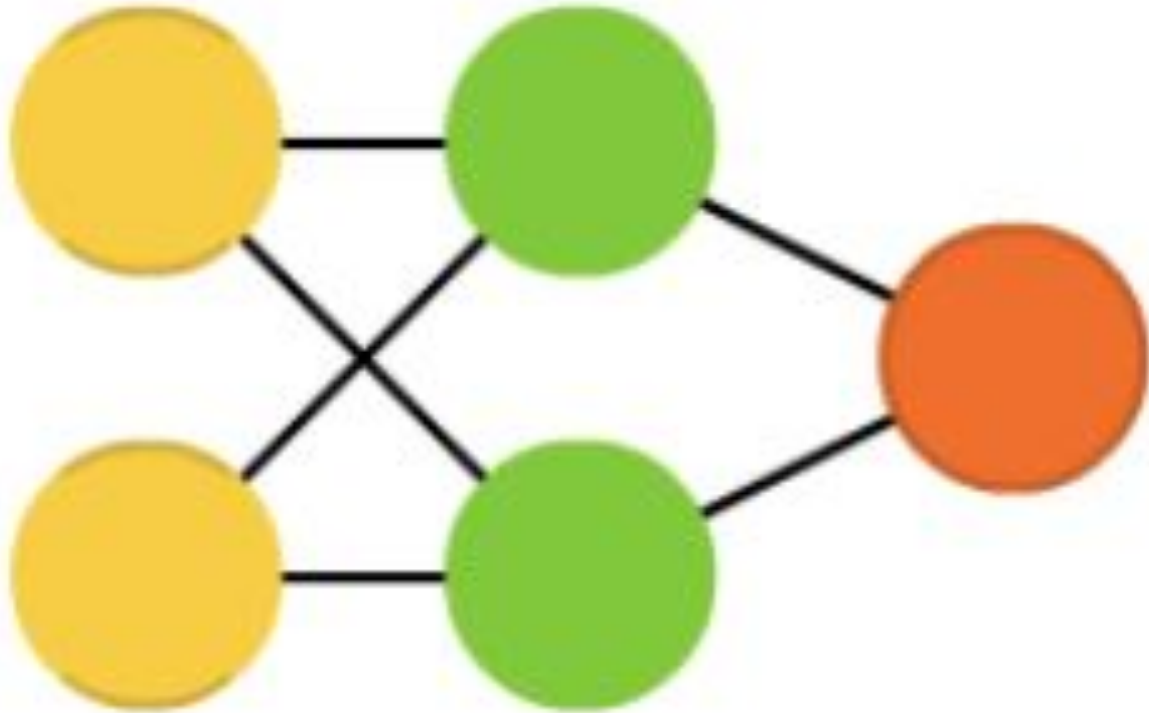
All use `Sequential()` at the start.

All use `.compile()` and `.fit()` at the end.

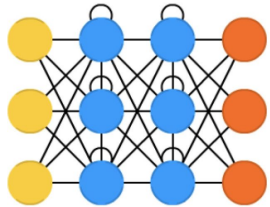
Neural Networks (Layers) types

Layers	When to use	Parameters	Keras functions	Data prep	Representation
Dense	Classification & Regression	epochs batch_size layers and nodes	<u>Dense</u>	Scale continuous	

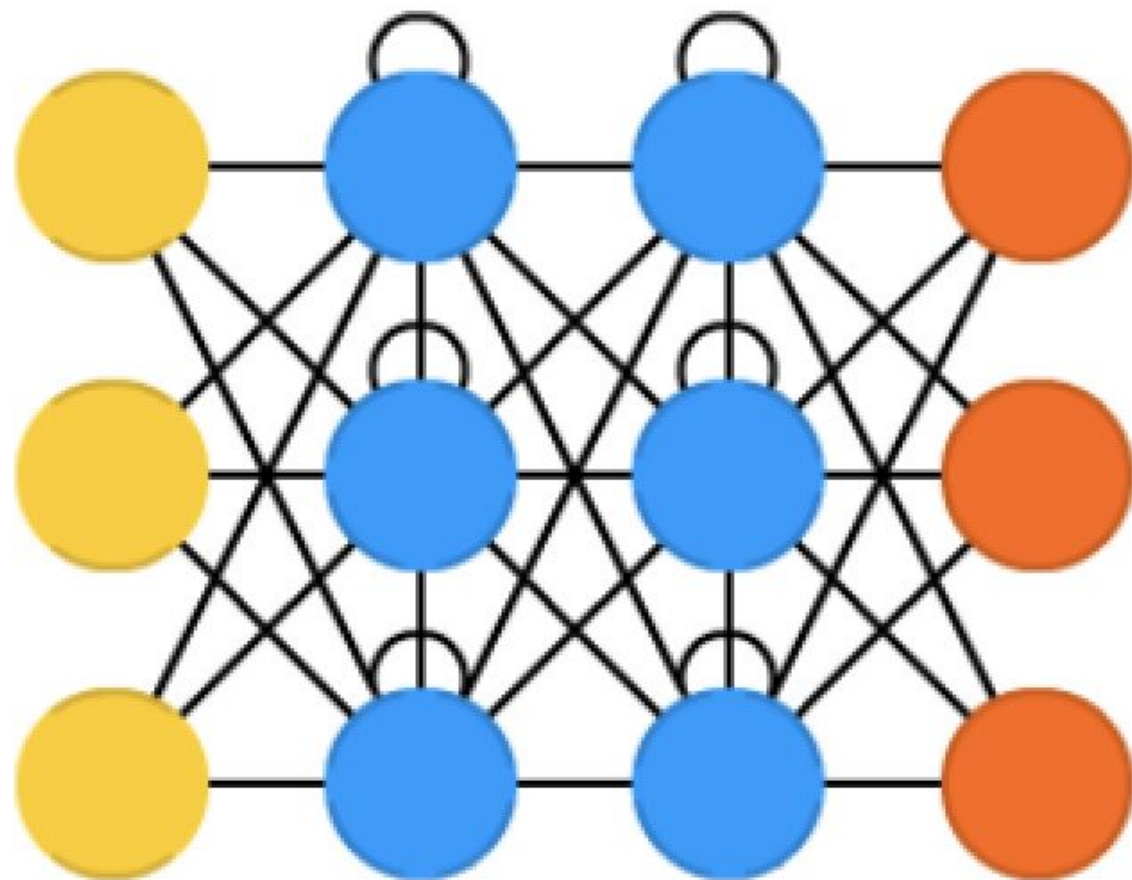
Dense layers



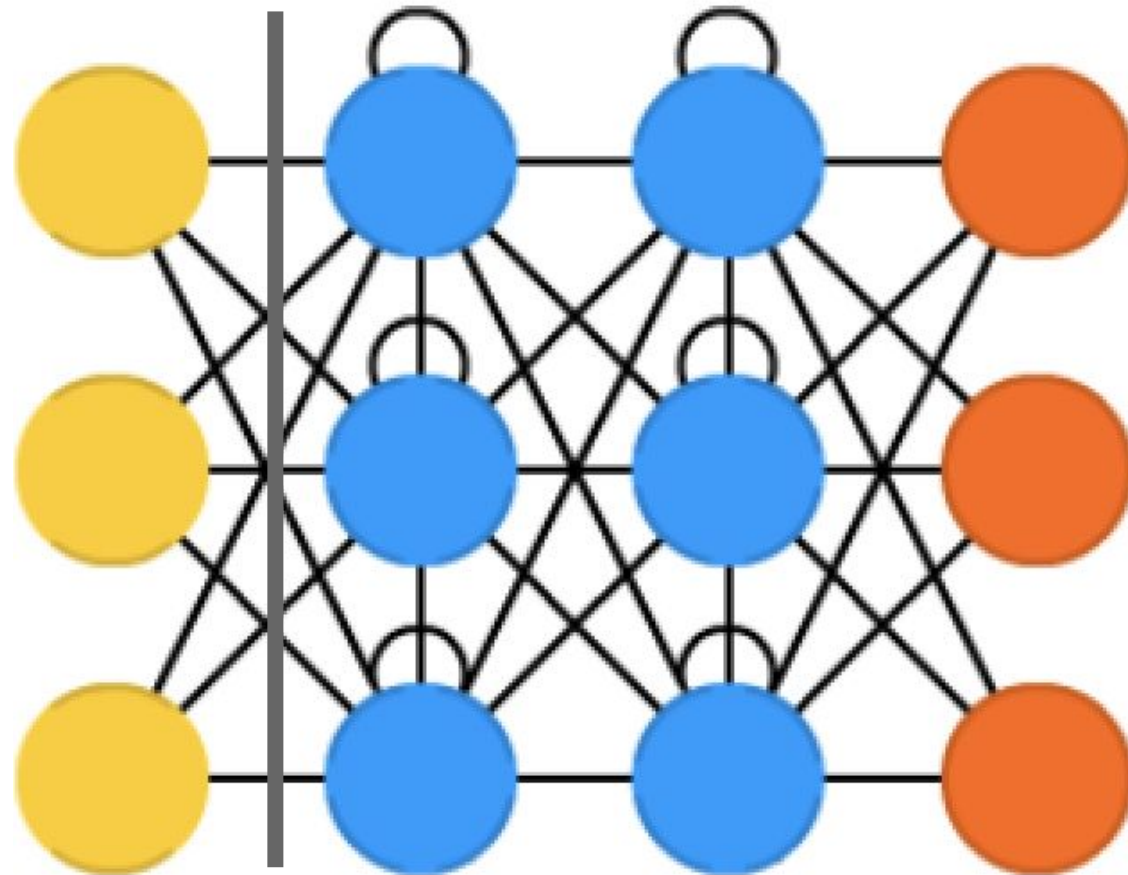
Neural Networks (Layers) types

Layers	When to use	Parameters	Keras functions	Data prep	Representation
Dense	Classification & Regression	epochs batch_size layers and nodes	Dense	Scale continuous	
RNNs	Anything for which the sequence of data is important	seq_length output_dim	Embedding SimpleRNN	Tokenizing One-hot target Embeddings	

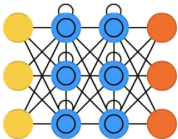
RNNs



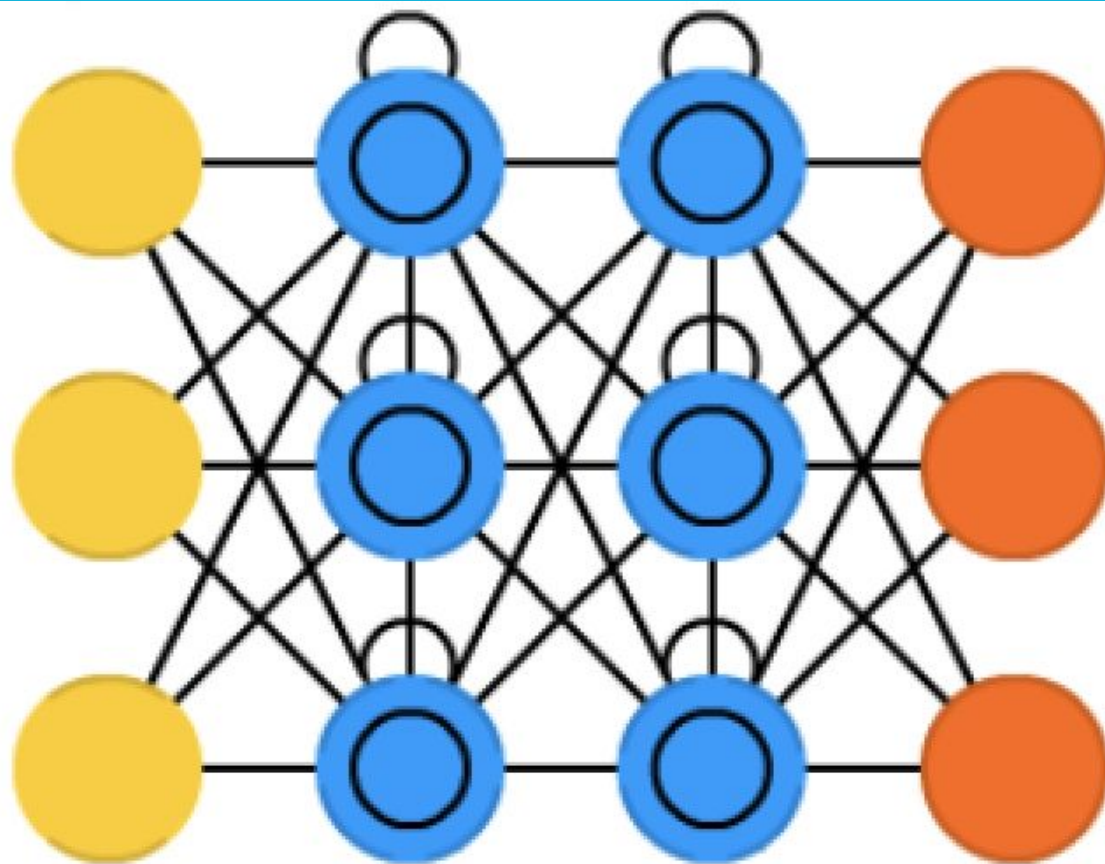
Embeddings



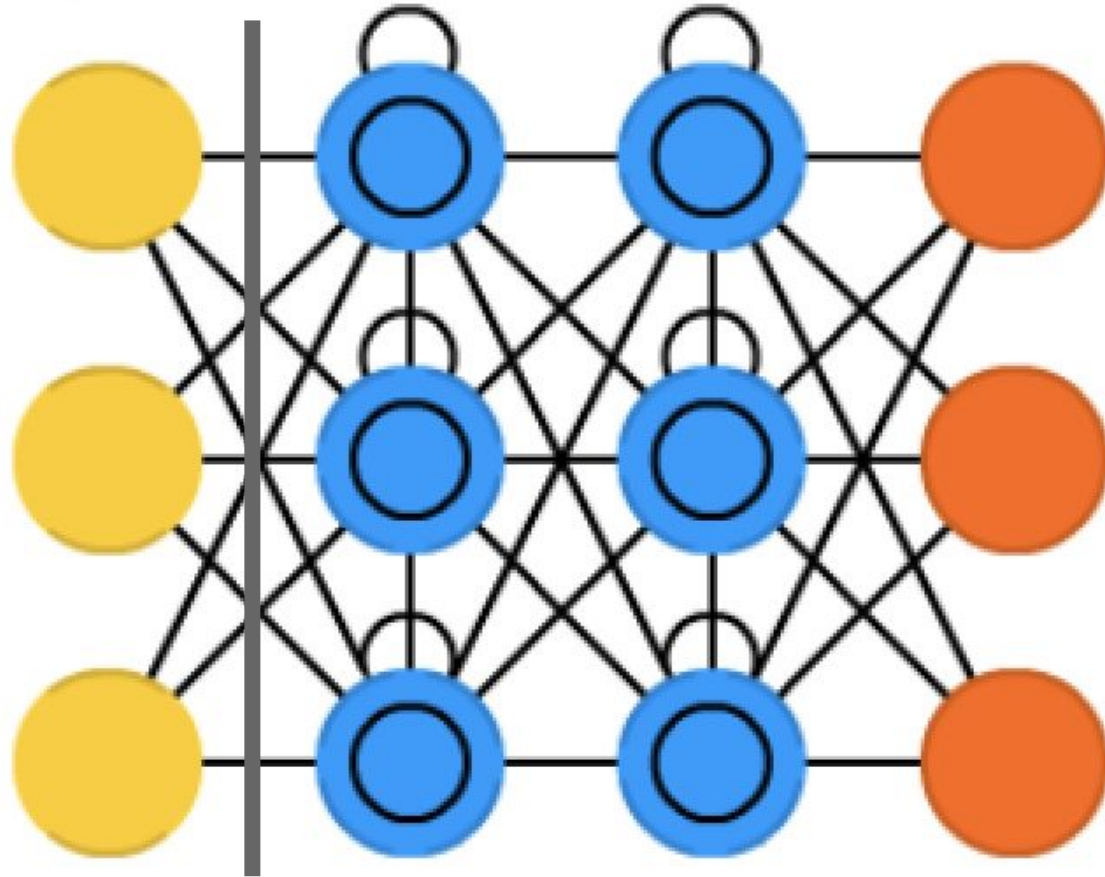
Neural Networks (Layers) types

Layers	When to use	Parameters	Keras functions	Data prep	Representation
Dense	Classification & Regression	epochs batch_size layers and nodes	Dense	Scale continuous	
RNNs	Anything for which the sequence of data is important	seq_length output_dim	Embedding SimpleRNN	Tokenizing One-hot target Embeddings	
LSTM	When items far in the past are relevant	seq_length output_dim	<u>LSTM</u>	Tokenizing One-hot target Embeddings	

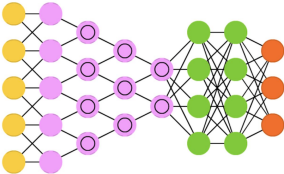
LSTM



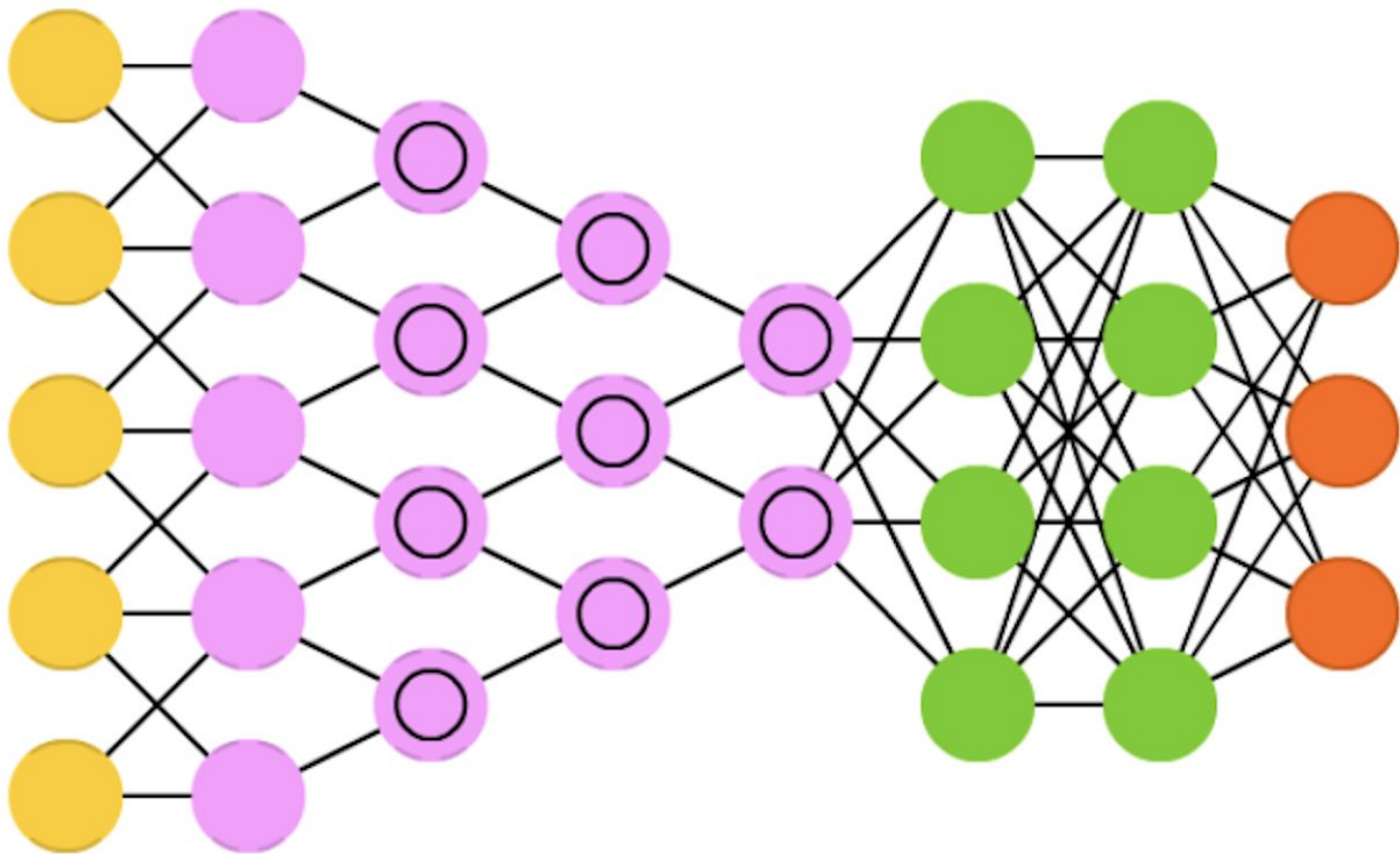
Embeddings



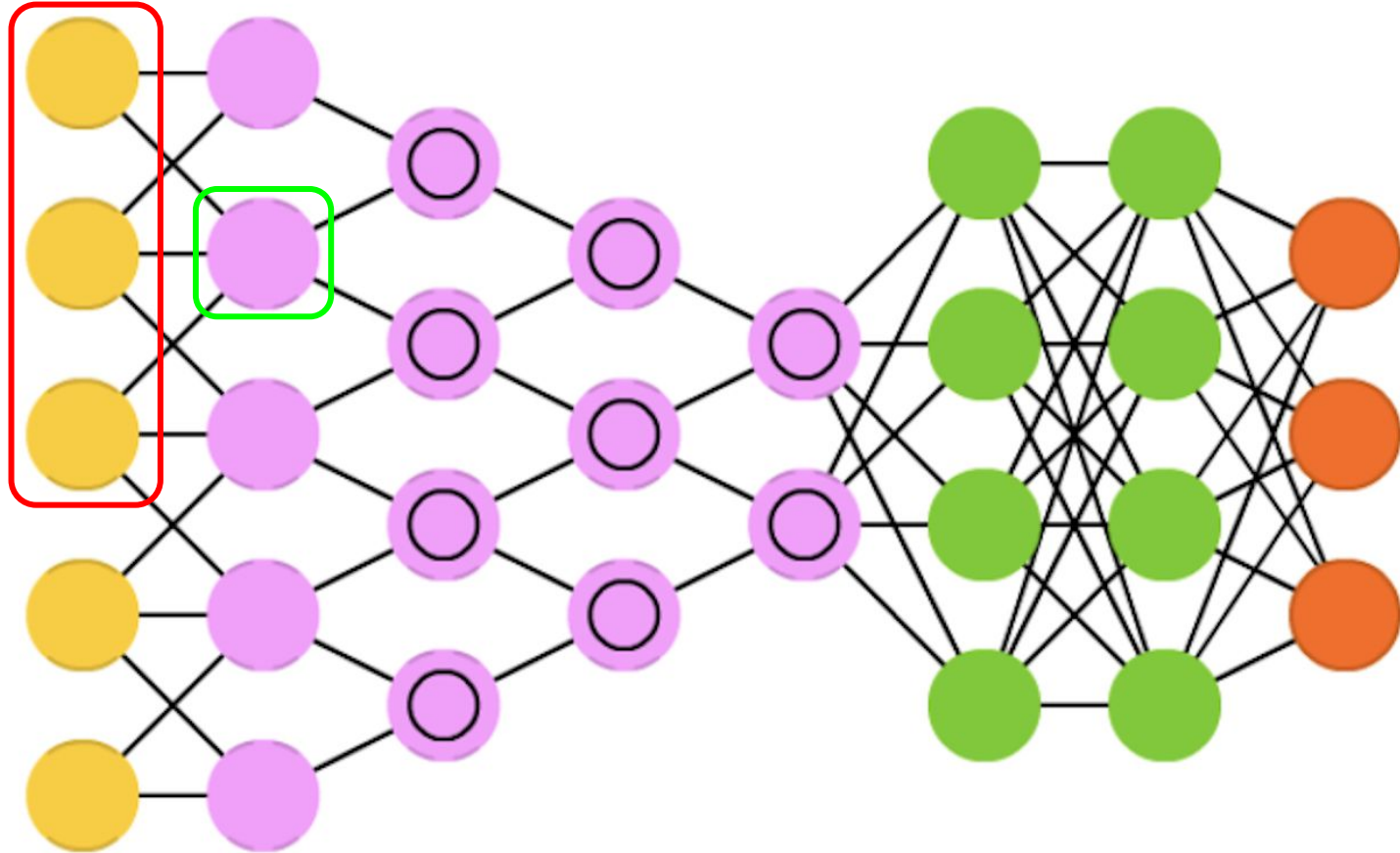
Neural Networks (Layers) types

Layers	When to use	Parameters	Keras functions	Data prep	Representation
Dense	Classification & Regression	epochs batch_size layers and nodes	Dense	Scale continuous	
RNNs	Anything for which the sequence of data is important	seq_length output_dim	Embedding SimpleRNN	Tokenizing One-hot target Embeddings	
LSTM	When items far in the past are relevant	seq_length output_dim	LSTM	Tokenizing One-hot target Embeddings	
CNNs	Image recognition	kernel_size strides pool_size	Conv2D MaxPooling2D Flatten	Image to tensor of integers Uniform aspect ratio	

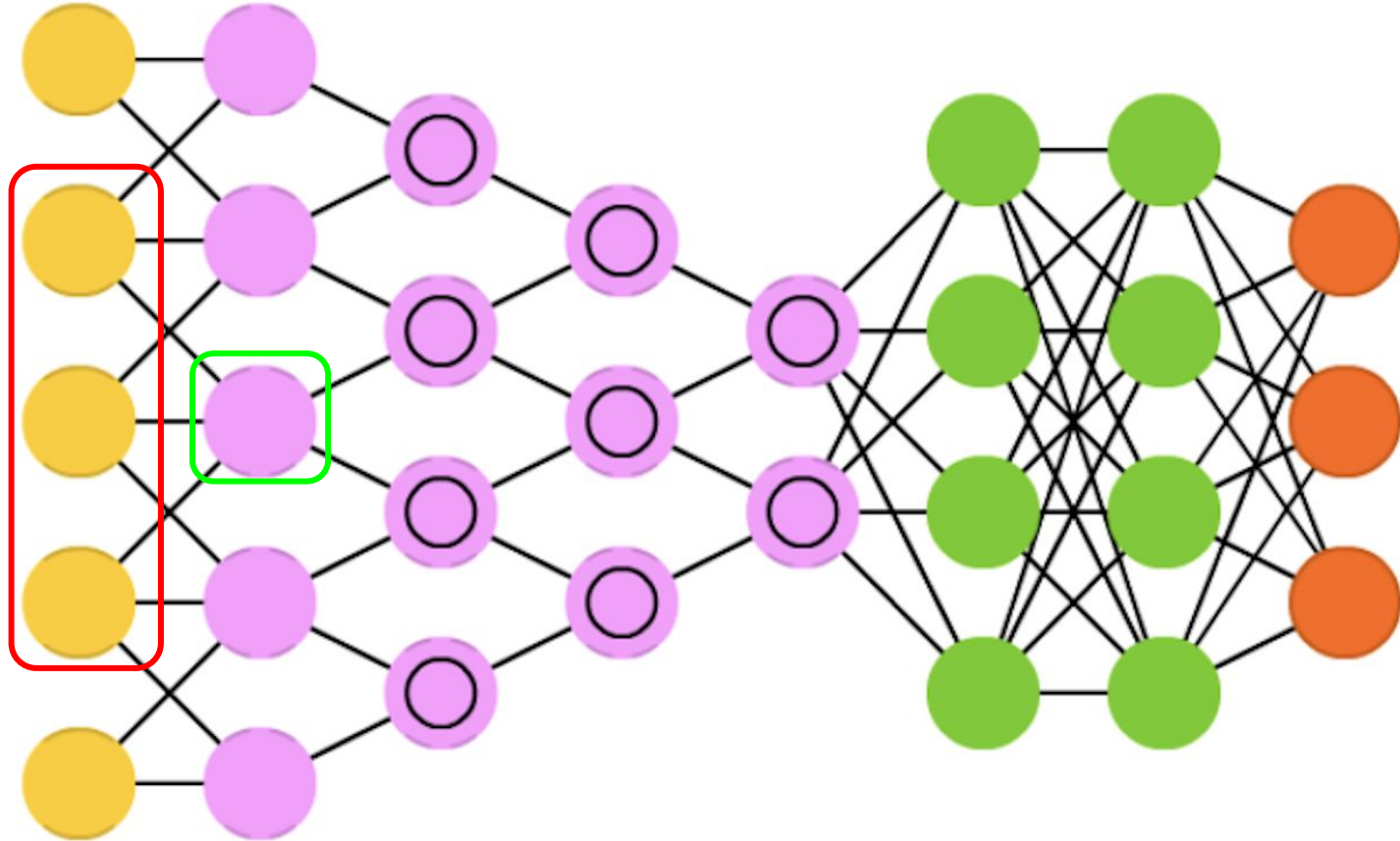
CNNs



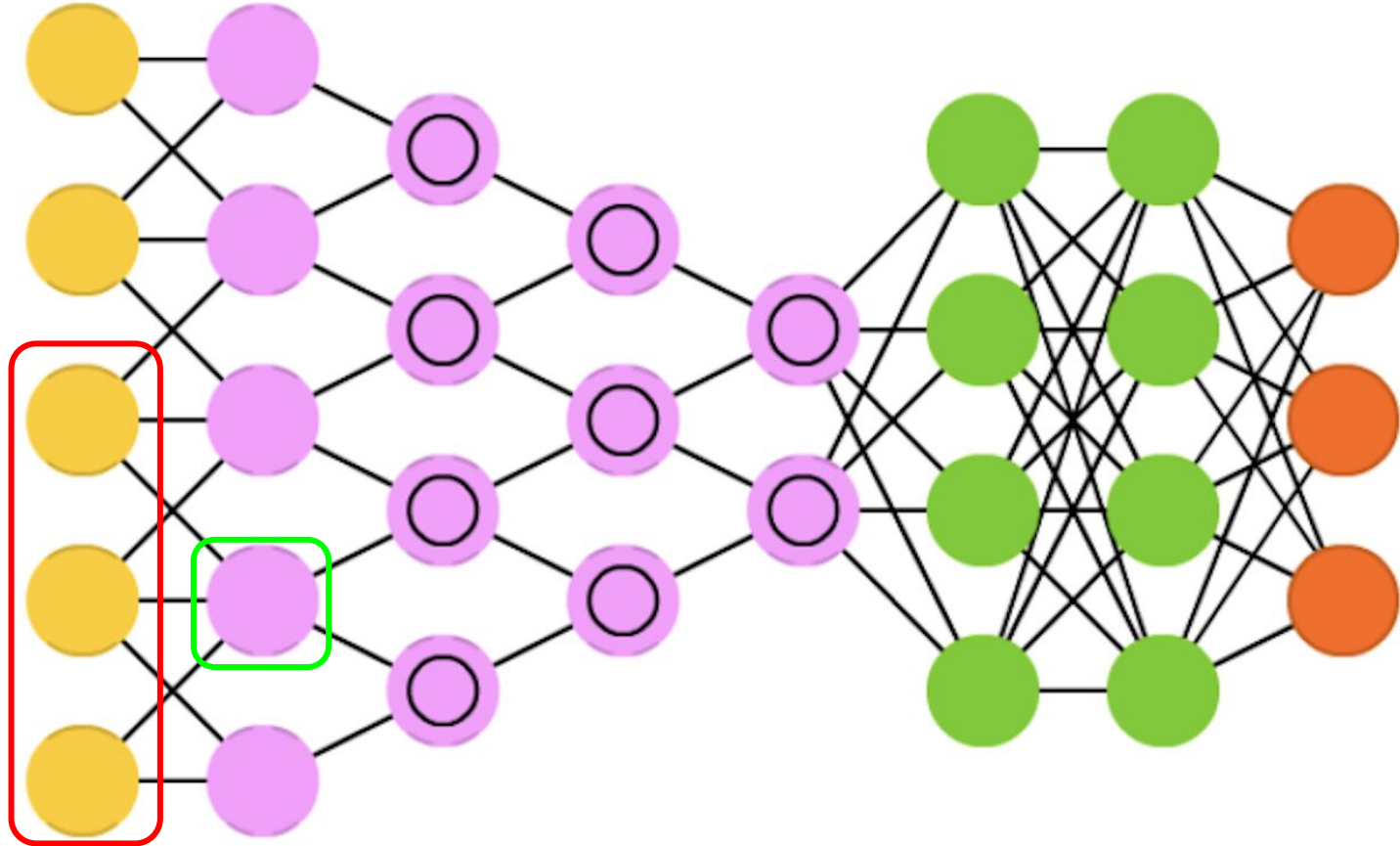
Convolutional layer



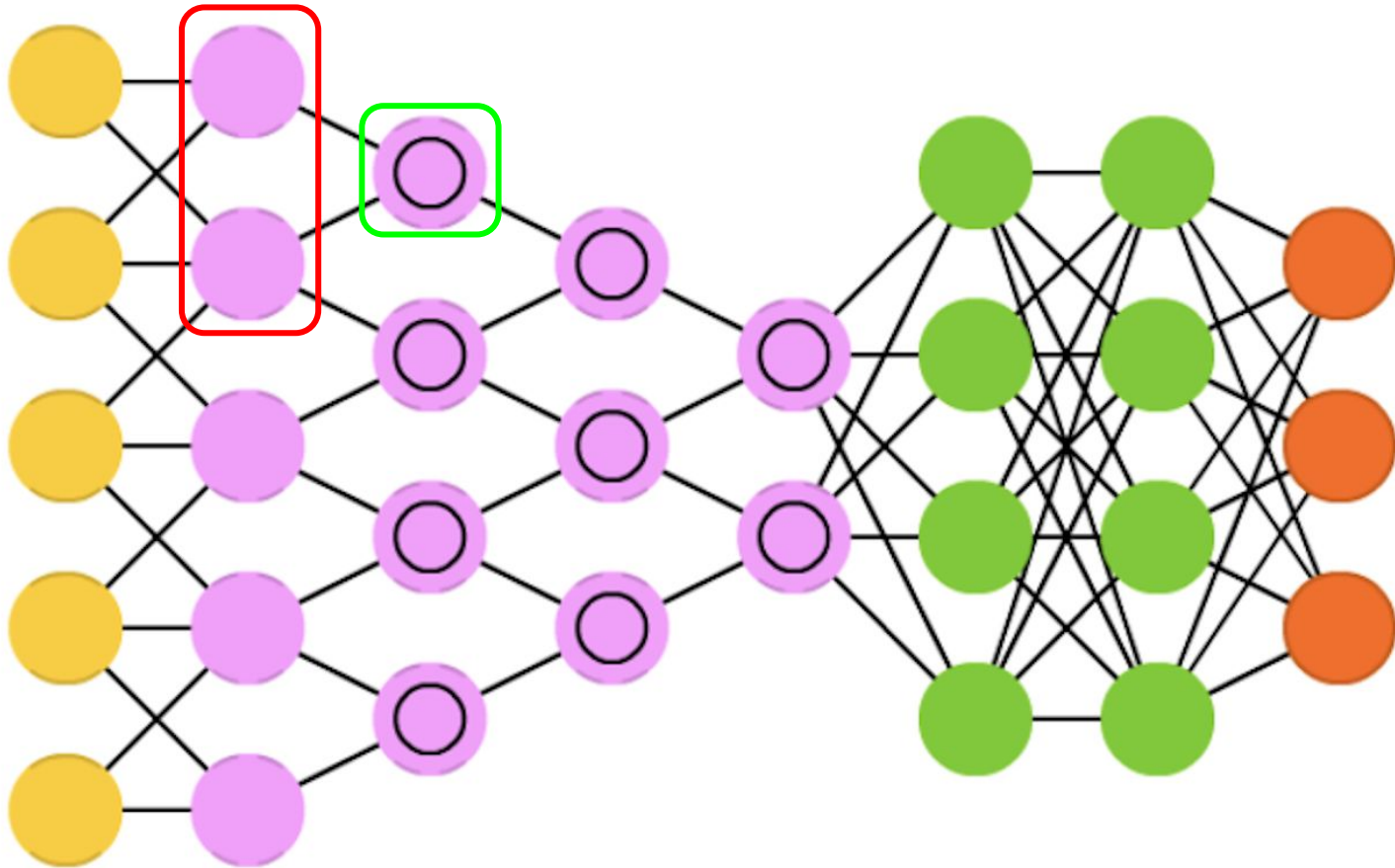
Convolutional layer



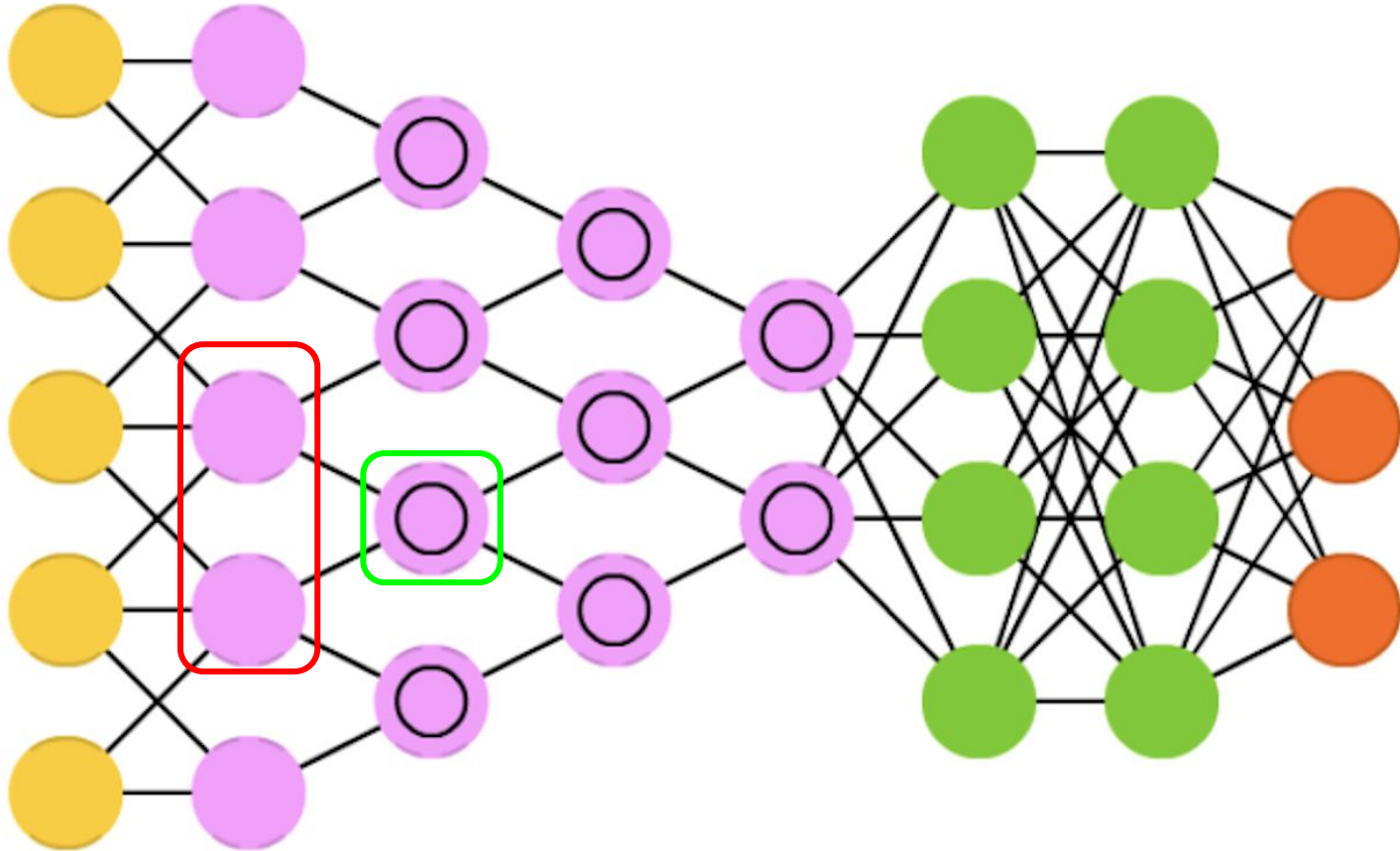
Convolutional layer



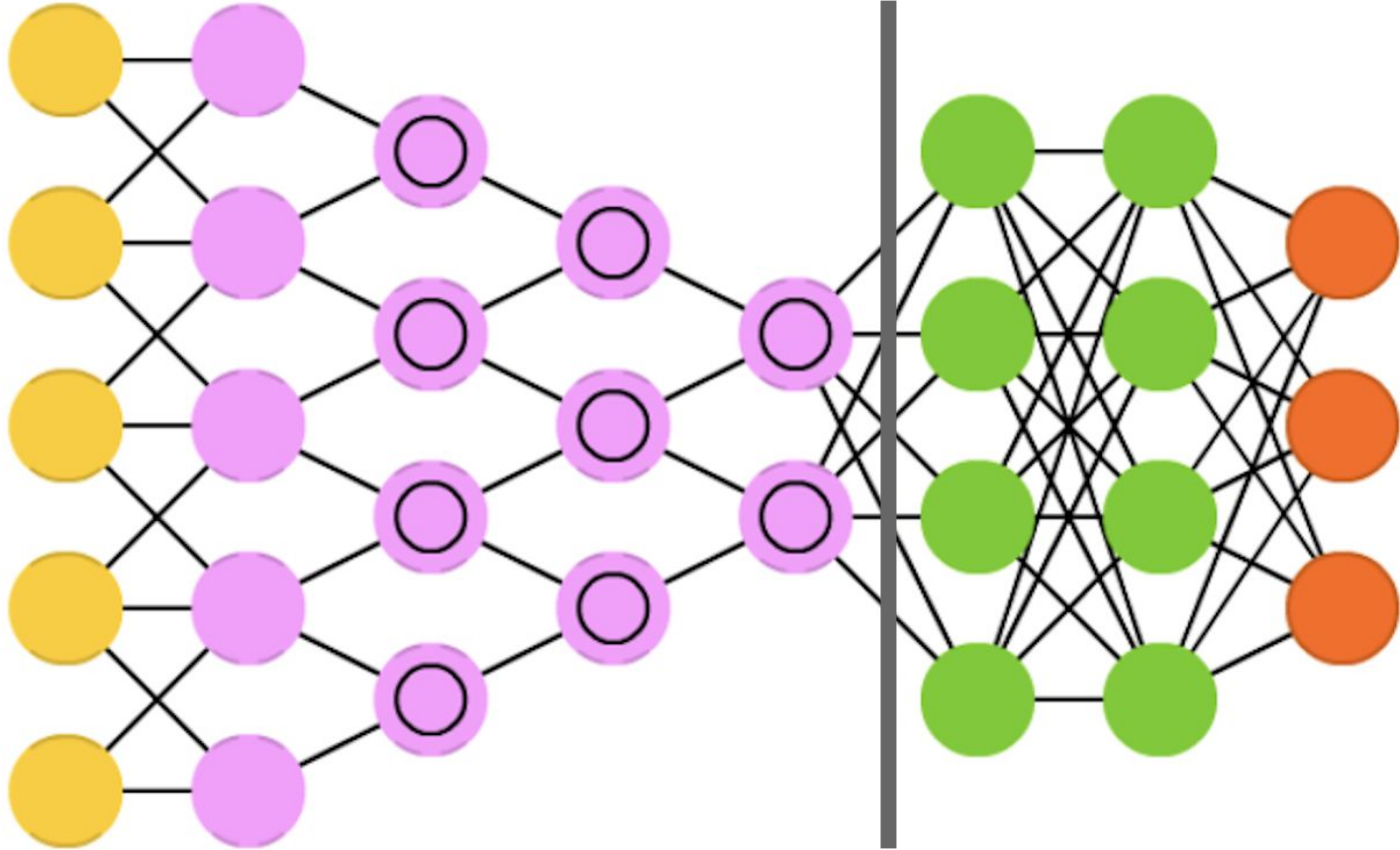
Pooling layer



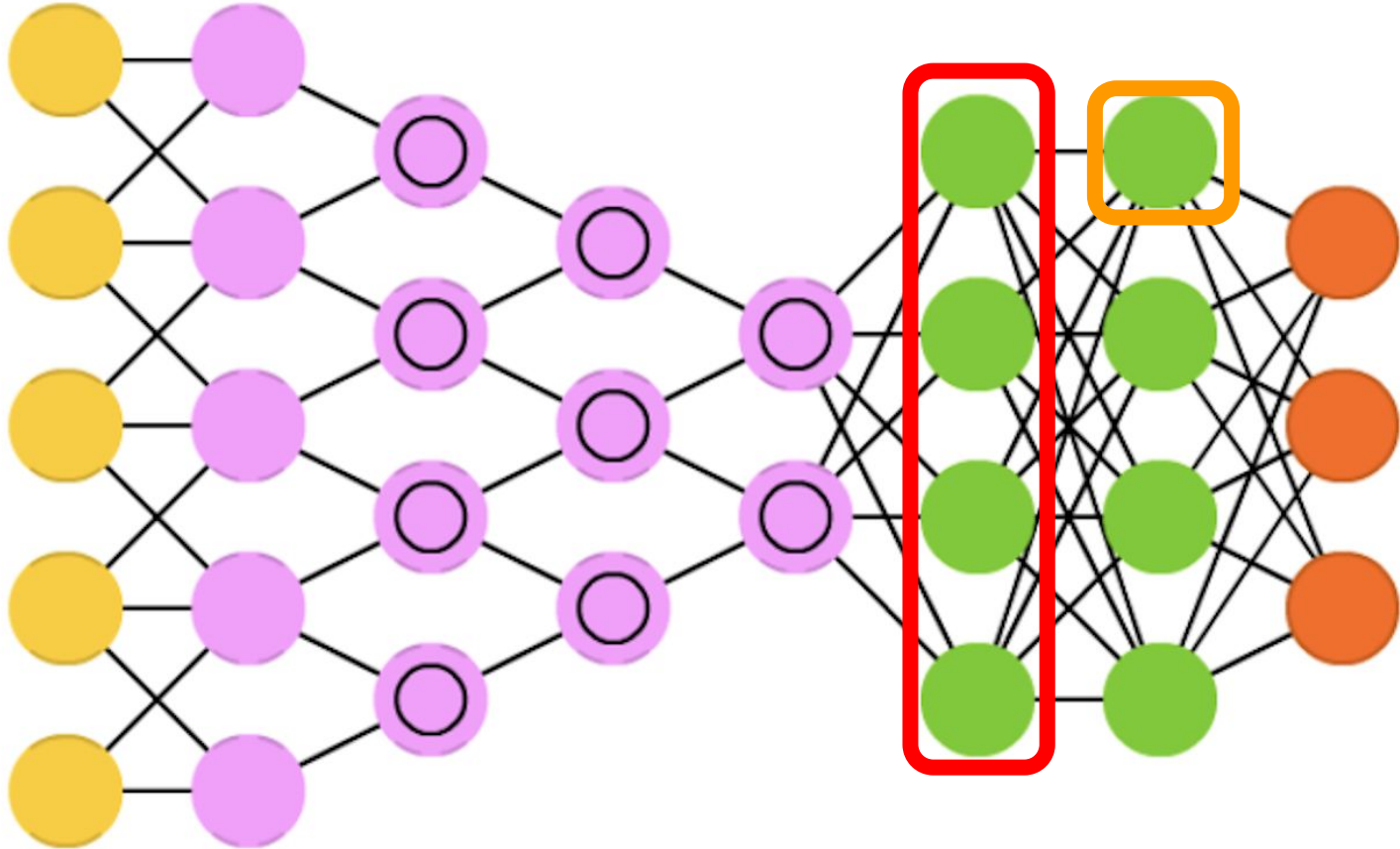
Pooling layer



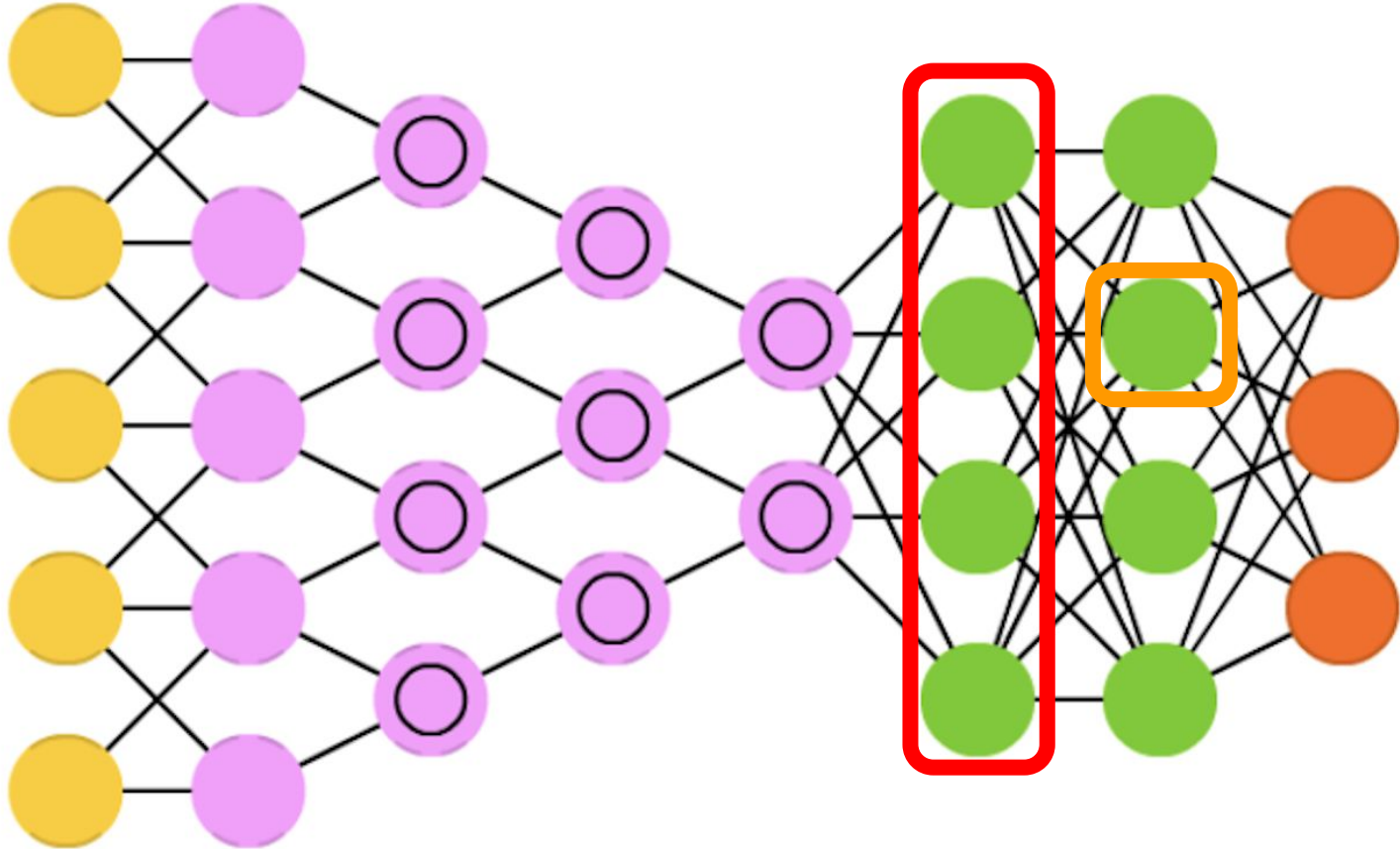
Flatten layer



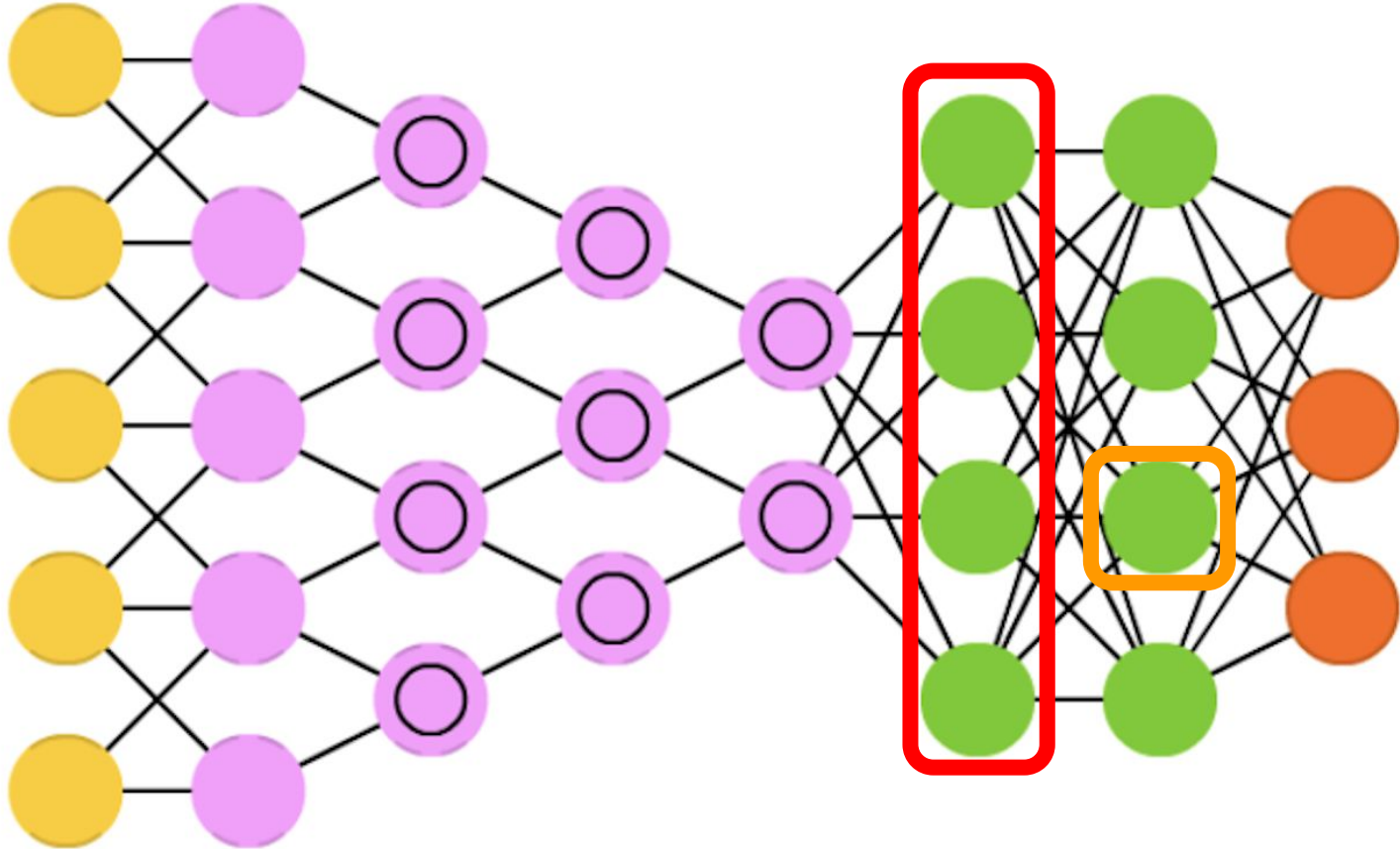
Dense layer



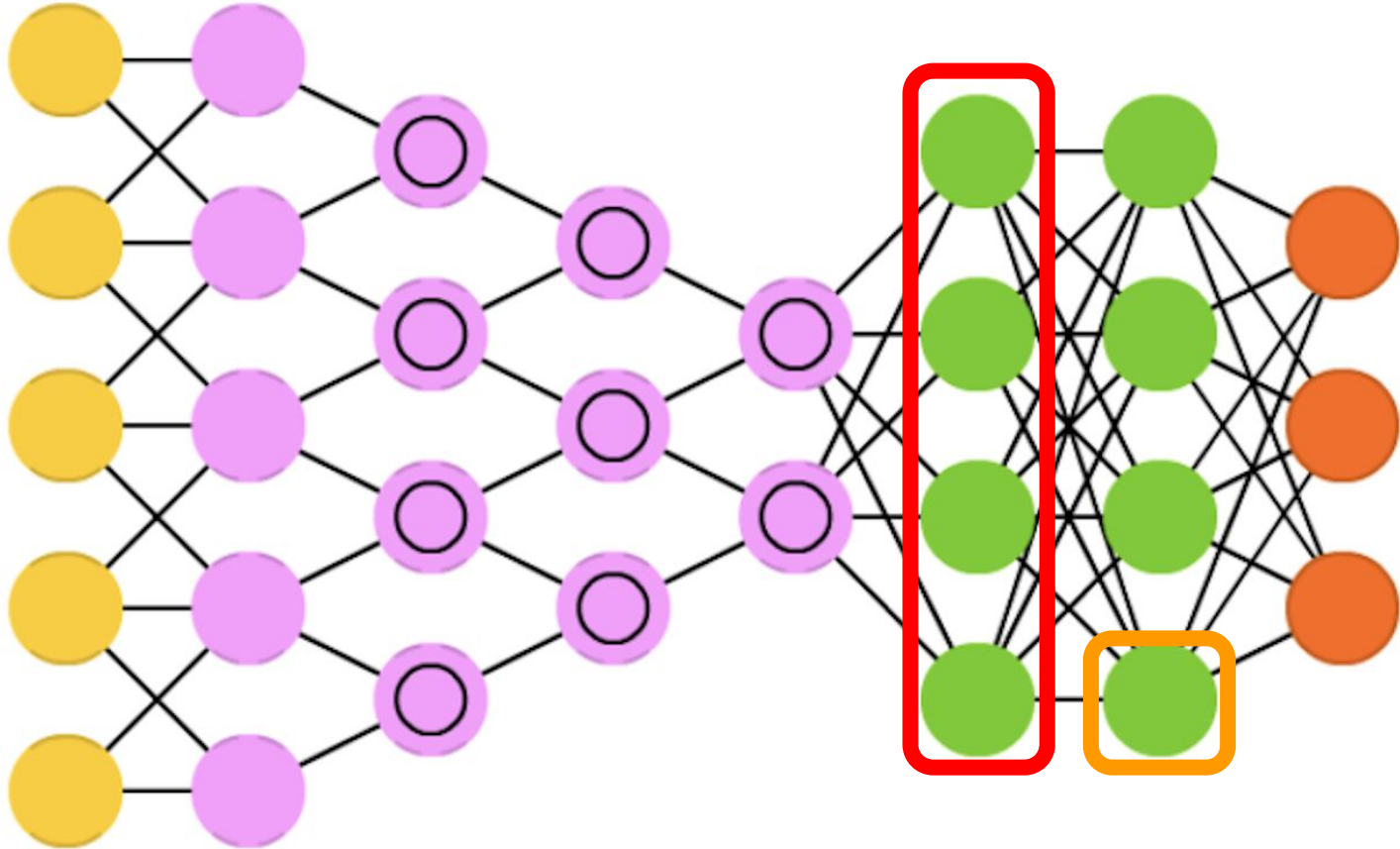
Dense layer



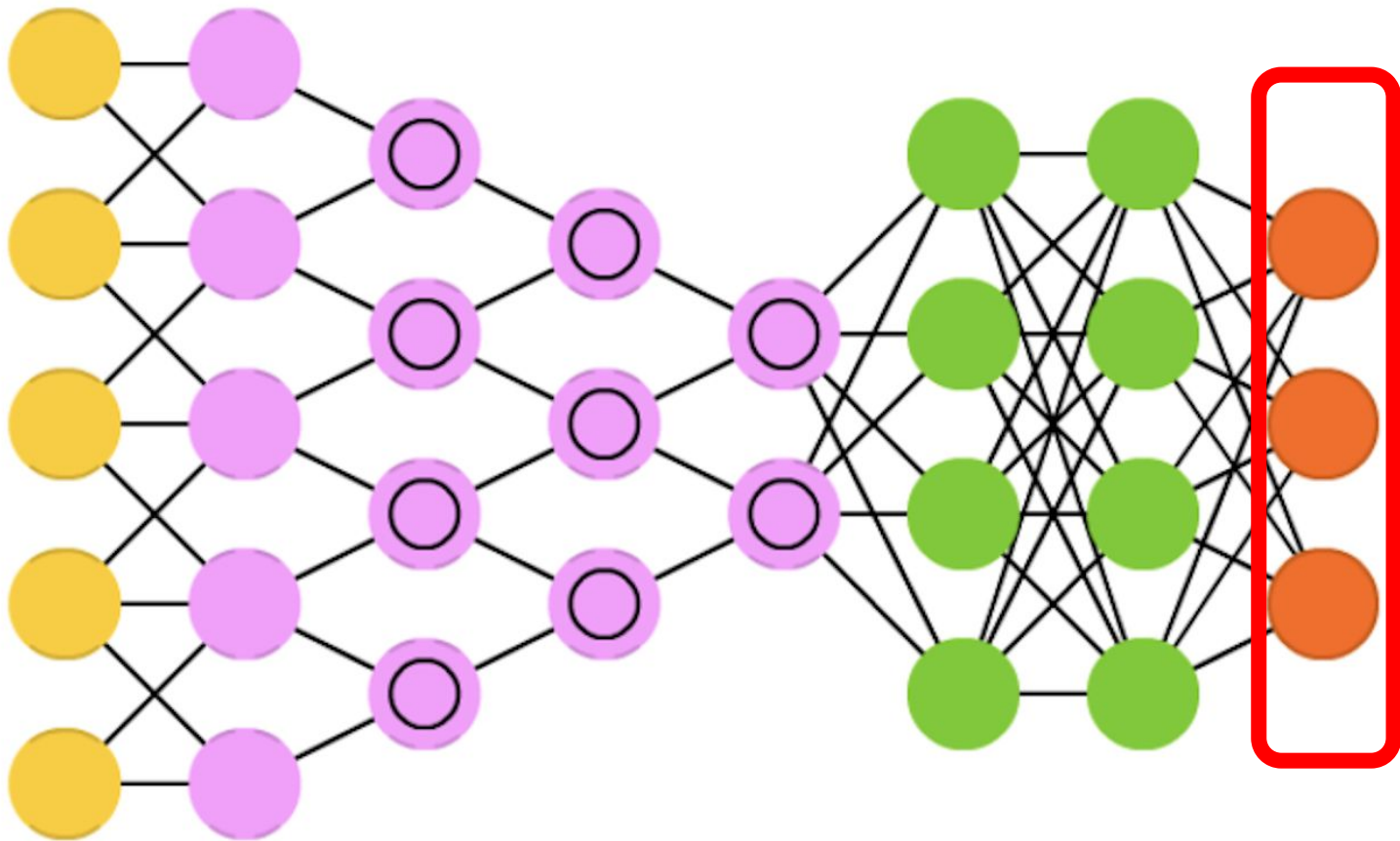
Dense layer




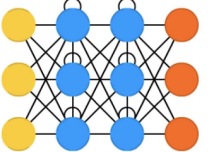
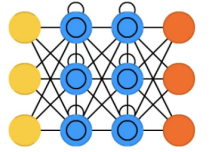
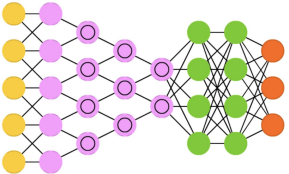
Dense layer



Output layer w/ $\text{soft}(\text{arg})\text{max}$



Neural Networks (Layers) types

Layers	When to use	Parameters	Keras functions	Data prep	Representation
Dense	Classification & Regression	epochs batch_size layers and nodes	Dense	Scale continuous	
RNNs	Anything for which the sequence of data is important	seq_length output_dim	Embedding SimpleRNN	Tokenizing One-hot target Embeddings	
LSTM	When items far in the past are relevant	seq_length output_dim	LSTM	Tokenizing One-hot target Embeddings	
CNNs	Image recognition	kernel_size strides pool_size	Conv2D MaxPooling2D Flatten	Image to tensor of integers Uniform aspect ratio	

From Keras creator

“Not all problems can be solved;
just because you've assembled examples of inputs X and targets Y
doesn't mean X contains enough information to predict Y”