RNN
layer

$$y_t^T = f\left(x_t^T H + v^T + y_{t-1}^T G\right)$$

mixes within current input feature vector

mixes within previous output feature vector

if G was identity matrix then current output is previous output plus a perturbation

NN
(single input)
layer

$$y^T = f\left(x^T H + v^T\right)$$

mixes within input feature vector

NN
(batch input)
layer

$$Y^T = f\left(X^T H + 1 v^T\right)$$

mixes within input feature vectors
does not mix across input feature vectors

Self attention
(single headed)
layer

$$Y^T = A^T X^T H,$$

mixes across input feature vectors

mixes within input feature vectors

$$A^T = \underset{\text{row}}{\text{softmax}}\left(\frac{X^T W_q W_k^T X}{\text{scalar}}\right)$$

self attention is function of the weighted product $X^T W X$

each row in A^T is a pmf

Encoder-attention-decoder
(linear weighting)
layer

this ↖

$$y^T = a^T X^T,$$

used by the decoder

mixes across input feature vectors

$$a^T = \text{softmax}\left(\frac{s^T W^T X}{\text{scalar}}\right)$$

s^T comes from the decoder

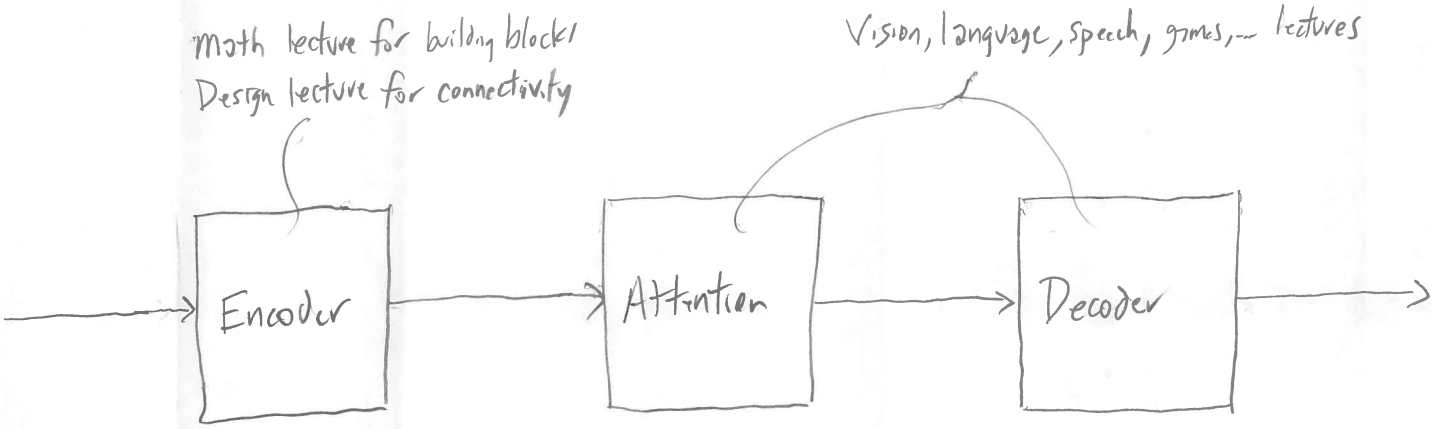
many variations on this theme

CNN
(not batched)
layer

$$Y^{3D} = f\left(H^{4D} \circledast X^{3D} \oplus v^{1D}\right)$$

$H^{4D} \circledast X^{3D}$ will be implemented via lowering to matrix matrix multiplication

Encoder – attention – decoder architecture



Examples:

Input image → ResNet50 with a FPN → Features → RPN followed by RoI extraction → RoIs → Classifier and bounding box regressor → Object detection

English words → Transformer → Features → Linear weighted attention → Attended features → Transformer with masking → French words