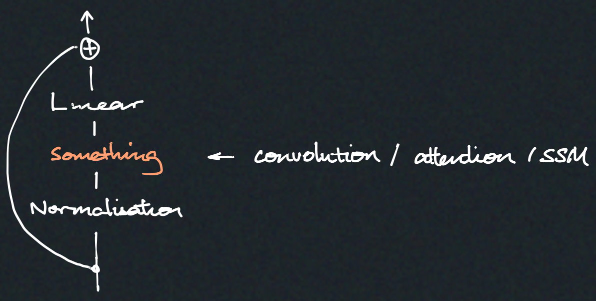
Lec 16 Long Seq Modelling

Deep sequence models very useful

Sequence model: seq $\xrightarrow{f_\theta(\cdot)}$ seq

RNN, CNN, Neural ODEs, Attention



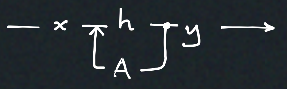
Baselines

RNN — recurrence cells
+ inherently causal
- slow training
- vanishing gradient

Attention
+ parallelisable
- quadratic time training
  └ finite context window

→ Selective State Spaces
+ parallel, fast, linear
+ long context
+ performance

SSM



signal processing
continuous, diffEQ
like RNN with continuous time,
big hidden state,
no activation,
linear

$h'(t) = \overset{\square}{A}h(t) + \overset{\|}{B}x(t)$

state transition

transform input

$y(t) = \overset{\square}{C}h(t) + \overset{\square}{D}x(t)$

project back to 1D

skip connection

Biased for continuous data, less good for text

▷ Discretised

$h = \bar{A}h + \bar{B}x$
$y = \bar{C}h + \bar{D}x$

* Issues
- no parallelisation along seq length
- large hidden (expensive)
- harder to train, given known future

▷ Convolution version of SSM

Equivalent to convolution

$y(t) = x(t) * K(t)$

convolution kernal defined by A, B, C, D

+ Fast fourrier transform, near linear time

SSSM

Linear Time Invarant — params invariant through time

SSSM — State Space Seq Model
SSSSM — Structured State Space Seq Model

Things viewed as SSM:

- RNN → state is one fixed-sized vec
  efficient, maybe too strong of compression on history
- Attention → state is cache of entire history
  attend to all past key & value
  good history, bad performance

Better model

Compress, selectively remember relevant information

Doing the selection: parametrise update func on current input

Make efficient: need to tailor to GPU hardware