CMPT 733 – Big Data Programming II

Deep Learning II

Instructor	Steven Bergner
Course website	https://coursys.sfu.ca/2025sp-cmpt-733-g1/pages/

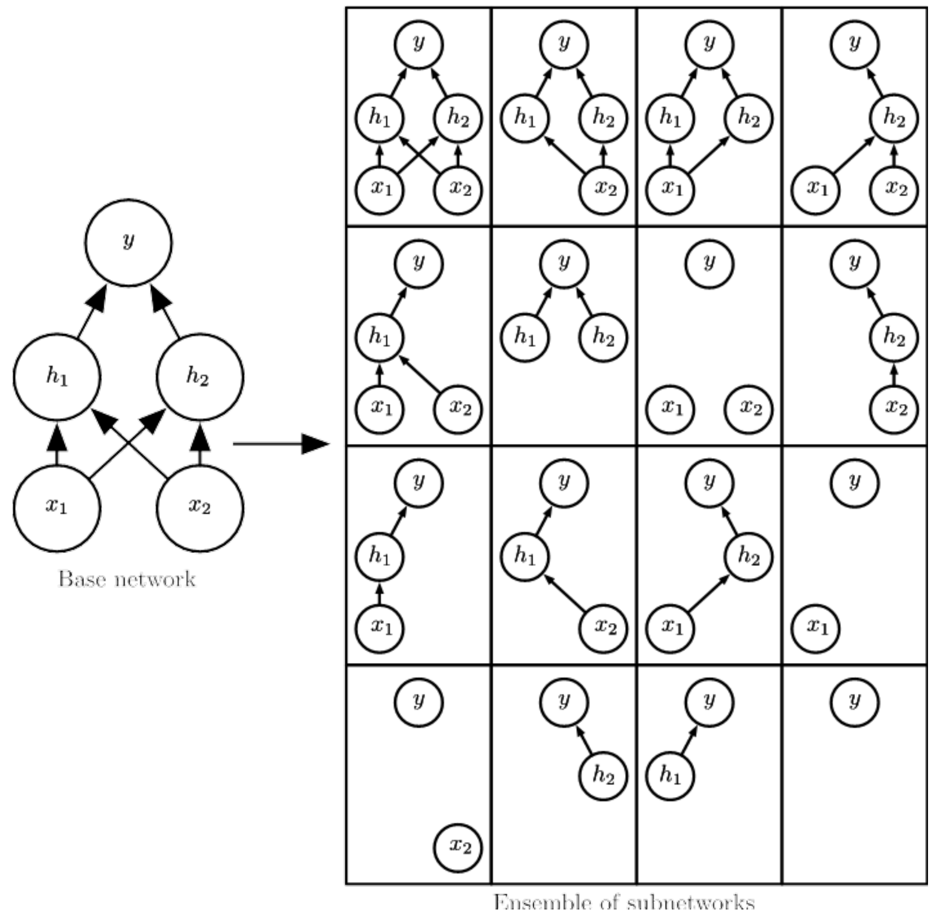
Overview

- Recap: Overfitting remedies
- Deep learning for sequences
- Natural language processing, e.g.
  - Sentiment analysis
  - Word embeddings
- Visualization for Deep Learning

Strategies against Overfitting (short recap)

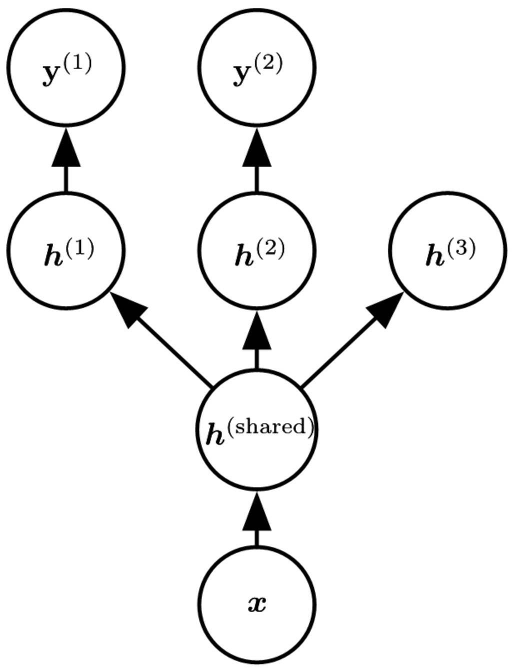
Dropout

- Random sample of connection weights is set to zero
- Train different network model each time
- Learn more robust, generalizable features



[Goodfellow, Bengio, Courville 2016]

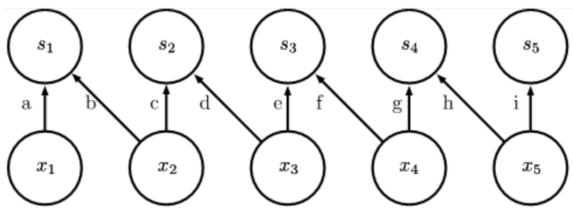
Multitask learning



- Shared parameters are trained with more data
- Improved generalization error due to increased statistical strength
- Missing components of y are masked from the loss function

[Goodfellow, Bengio, Courville 2016]

Types of connectivity



Local connection: like convolution, but no sharing

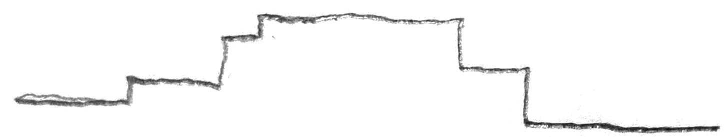
$$\begin{bmatrix} a & b & & & & \\ & & c & d & & \\ & & & & e & f \\ & & & & & \ddots \end{bmatrix}$$

$$\begin{bmatrix} a & b & & & \\ & a & b & & \\ & & a & b & \\ & & & & \ddots \end{bmatrix}$$

$$\begin{bmatrix} a & b & c & d & \cdots \\ h & i & j & k & \cdots \\ o & p & q & r & \cdots \\ & & & & \ddots \end{bmatrix}$$

[Goodfellow, Bengio, Courville 2016]

Convolution calculation illustrated



$$[1\ 1\ 2\ 4\ 5\ 5\ 5\ 3\ 0\ 0]$$
$$*\ [-1\ 1]$$

Choosing architecture family

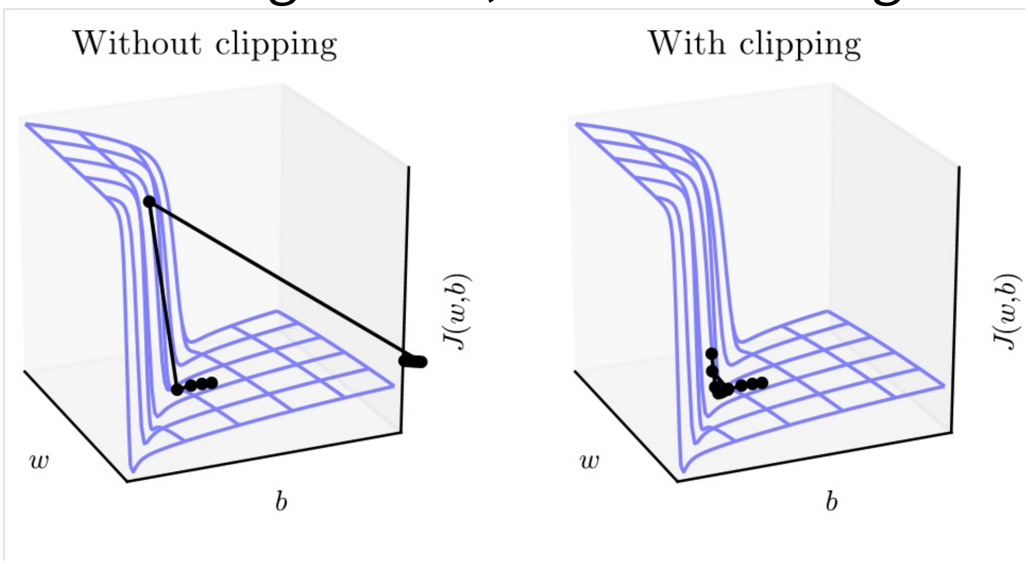
- No structure → fully connected
- Spatial structure → convolutional
- Sequential structure → recurrent

Optimization Algorithm

- Lots of variants address choice of learning rate
- See Visualization of Algorithms
- AdaDelta and RMSprop often work well

Gradient Clipping

- Add learning rate time gradient to update parameters
- Believe direction of gradient, but not its magnitude



[Goodfellow, Bengio, Courville 2016]

Development strategy

- Identify needs: High accuracy or low accuracy?
- Choose metric
  - Accuracy (% of examples correct), Coverage (% examples processed)
  - Precision TP/(TP+FP), Recall TP/(TP+FN)
  - Amount of error in case of regression
- Build end-to-end system
  - Start from baseline, e.g. initialize with pre-trained network
- Refine driven by data

Software for Deep Learning

Current Frameworks

- Tensorflow / Keras
- PyTorch
- DL4J
- Caffe (superseded by Caffe2, which is merged into PyTorch)
- And many more
- Most have CPU-only mode but much faster on NVIDIA GPU

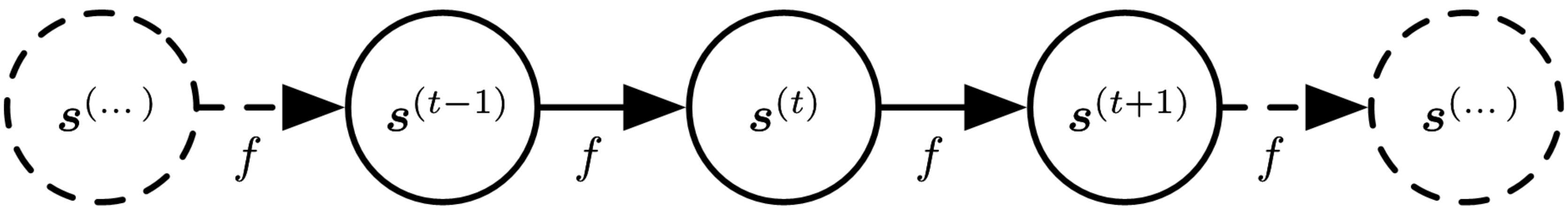
Recap: Choosing architecture family

- No structure → fully connected
- Spatial structure → convolutional
  - Adjacency or order of inputs has meaning
- Sequential structure → recurrent

Sequence Modeling with Recurrent Nets

Classical Dynamical Systems

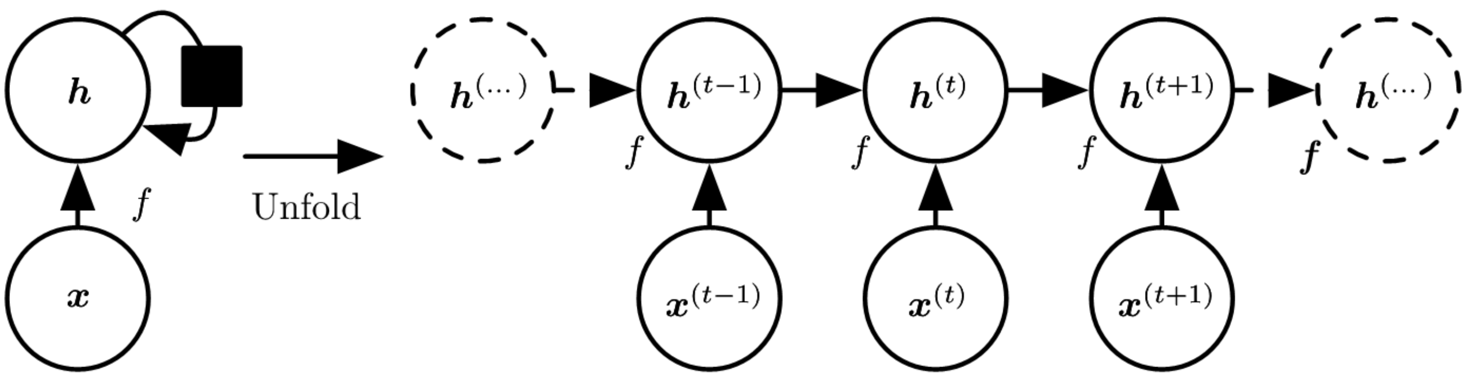
- Recurrent network models a dynamical system that is updated in discrete steps over time
- Function *f* takes input from time *t* to output at time *t+1*
- Rules persist across time



[Goodfellow, Bengio, Courville 2016]

Unfolding Computation Graphs

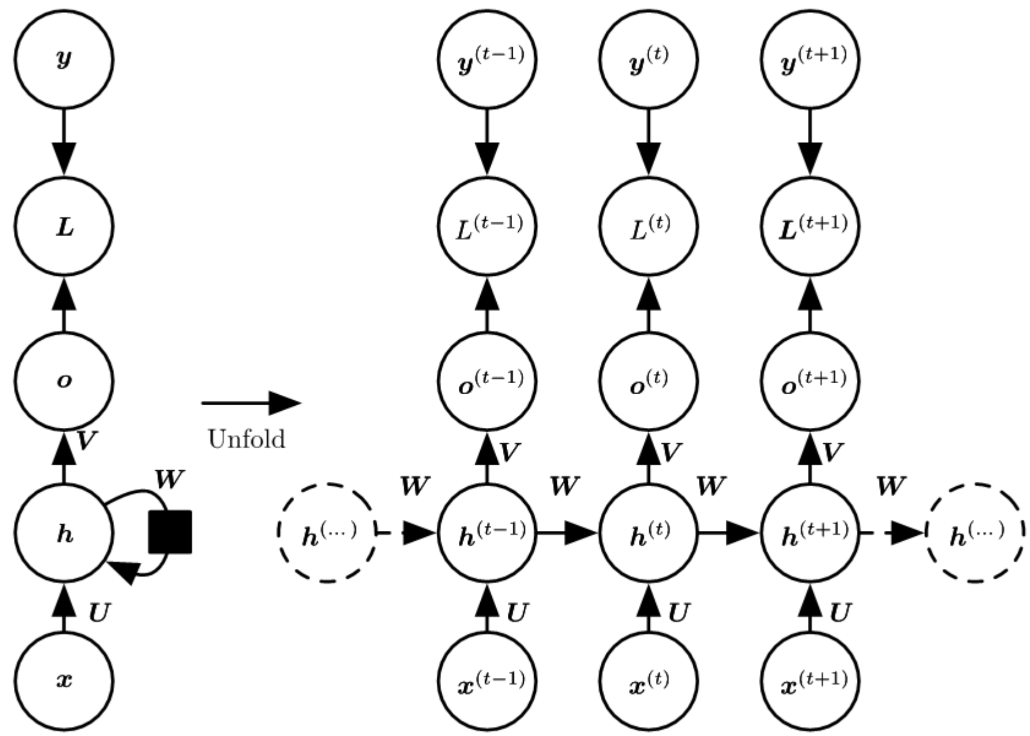
- Recurrent graph can be unfolded, where hidden state h is influencing itself
- Backprop through time is just backprop on unfolded graph



[Goodfellow, Bengio, Courville 2016]

Recurrent Hidden Units

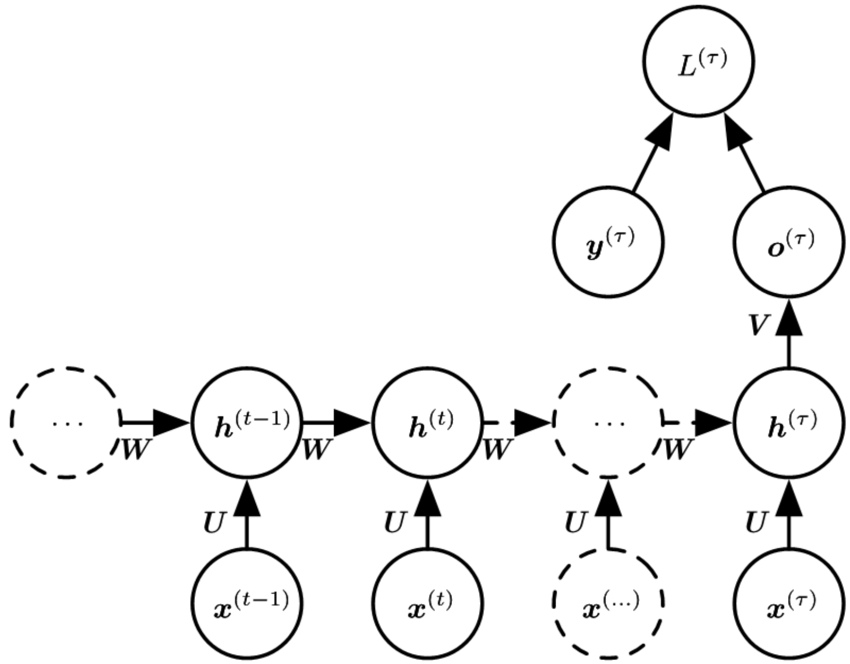
- Can have more than one layer



Sequence Input, Single Output

Example

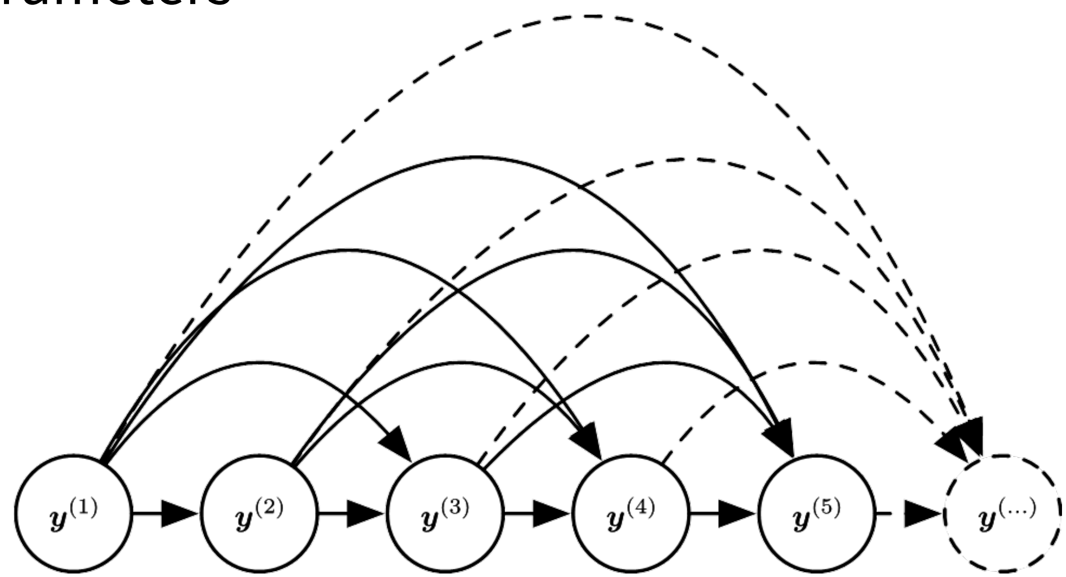
Sentiment analysis of text



[Goodfellow, Bengio, Courville 2016]

Fully Connected Graphical Model

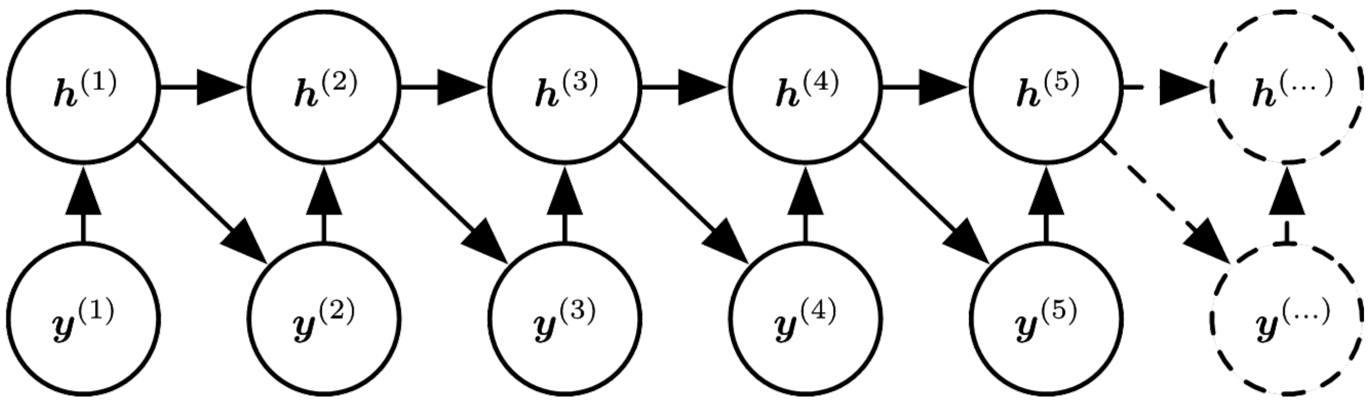
- Too many dependencies among variables, if each has its own set of parameters



[Goodfellow, Bengio, Courville 2016]

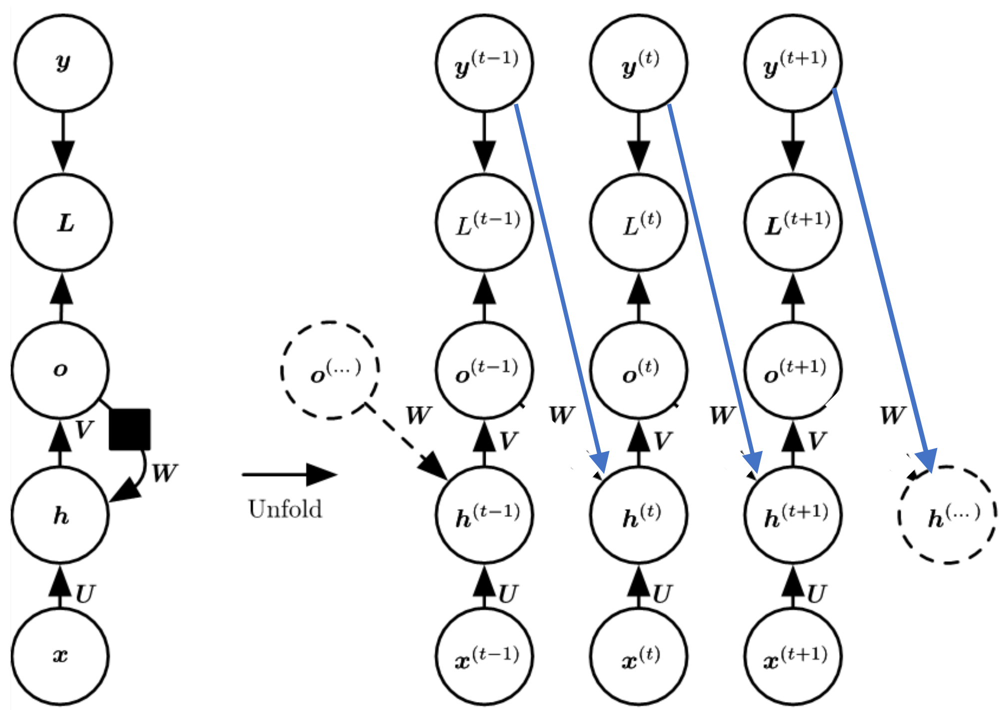
RNN Graphical Model

- Organize variables according to time with single update rule
- Finite set of relationships may extend to infinite sequences
- *h* acts as "memory state" summarizing relevant history



Recurrence only through output

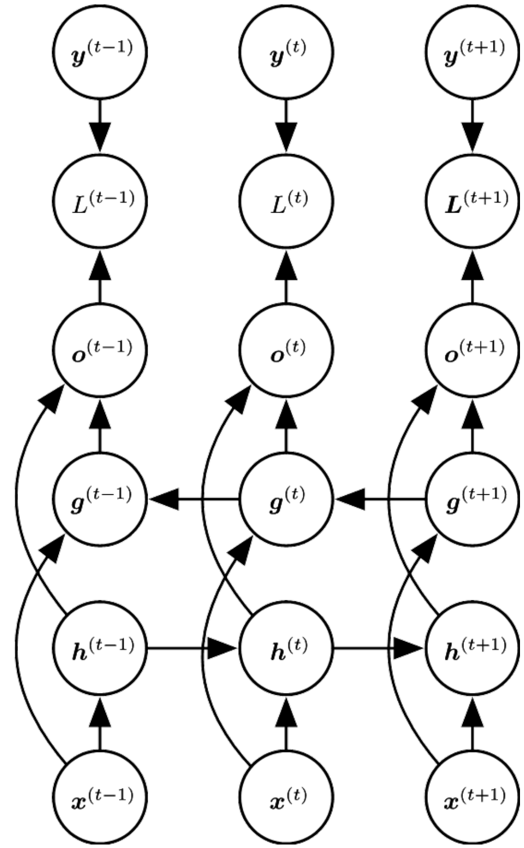
- Avoid backprop through time
- Mitigation: Teacher forcing
  - Use actual or expected output from the training dataset at current time y(t) as input o(t) to the next time step, rather than generated output
  - Backprop stops when it reaches y(t-1) via o(t-1)



[Goodfellow, Bengio, Courville 2016]

Bidirectional RNN

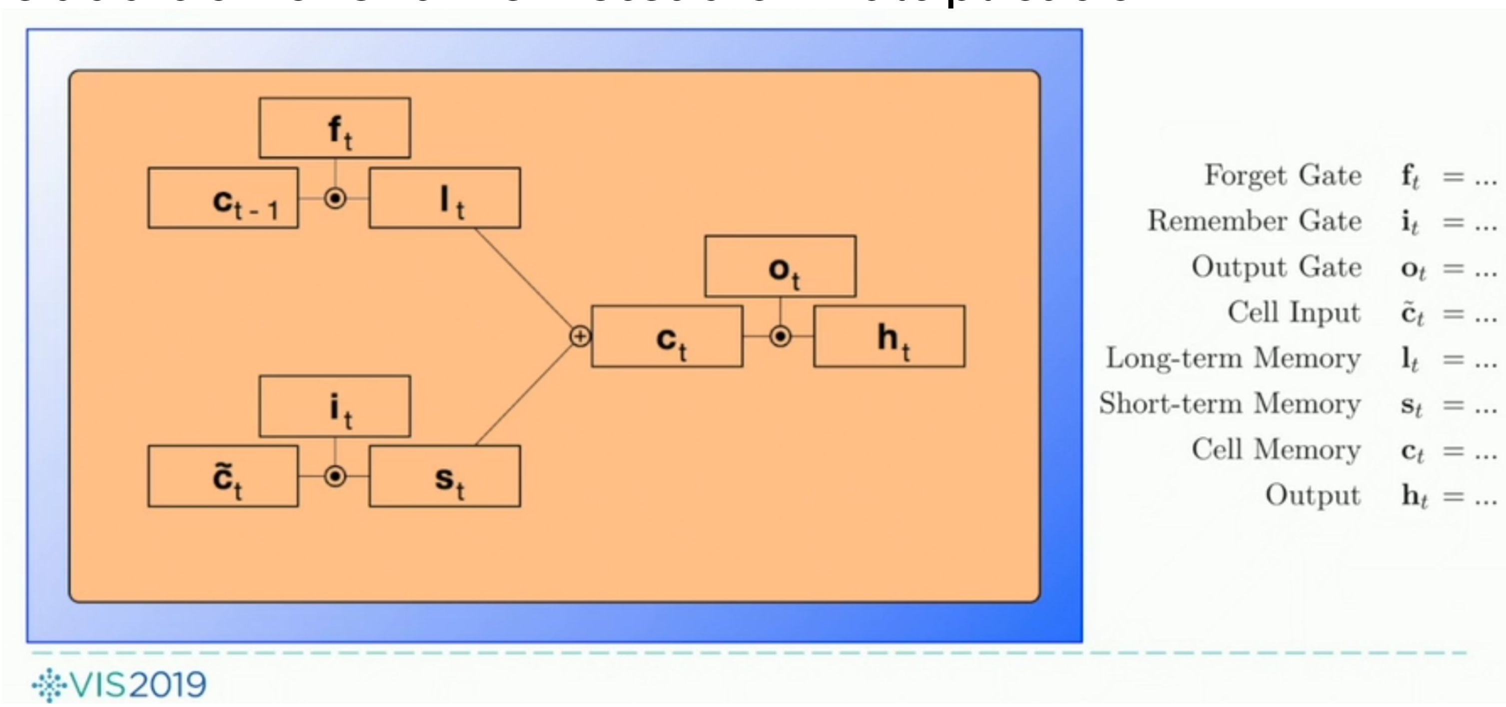
- Later information may be used to reassess previous observations



[Goodfellow, Bengio, Courville 2016]

LSTMs

- Use addition over time instead of multiplication



[Sawatzky, Bergner, Popowich, 2019]

Further Architectures

- Transformers
- Deep Reinforcement Learning

Excellent explanation of Attention

Karpathy's NanoGPT

https://www.youtube.com/watch?v=kCc8FmEb1nY&t=1s

NanoGPT implementation

https://github.com/karpathy/nanoGPT

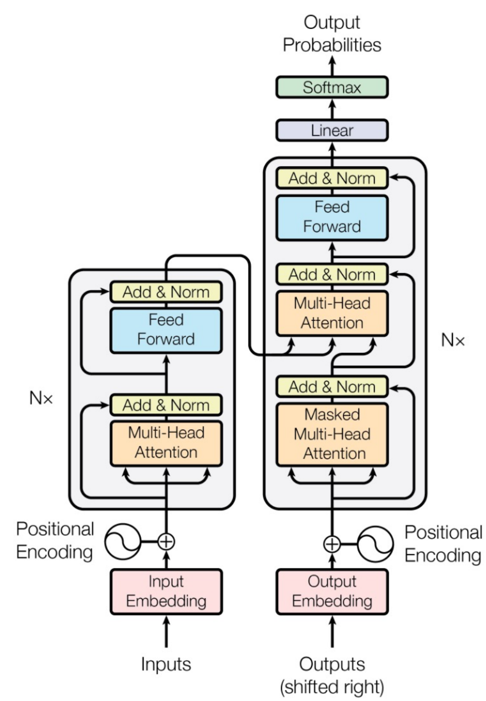


Figure 1: The Transformer - model architecture.

Generative language models are passing exams

Exam	GPT-4	GPT-4 (no vision)	GPT-3.5
Uniform Bar Exam (MBE+MEE+MPT)	298 / 400 (~90th)	298 / 400 (~90th)	213 / 400 (~10th)
LSAT	163 (~88th)	161 (~83rd)	149 (~40th)
SAT Evidence-Based Reading & Writing	710 / 800 (~93rd)	710 / 800 (~93rd)	670 / 800 (~87th)
SAT Math	700 / 800 (~89th)	690 / 800 (~89th)	590 / 800 (~70th)
Graduate Record Examination (GRE) Quantitative	163 / 170 (~80th)	157 / 170 (~62nd)	147 / 170 (~25th)
Graduate Record Examination (GRE) Verbal	169 / 170 (~99th)	165 / 170 (~96th)	154 / 170 (~63rd)
Graduate Record Examination (GRE) Writing	4 / 6 (~54th)	4 / 6 (~54th)	4 / 6 (~54th)
USABO Semifinal Exam 2020	87 / 150 (99th - 100th)	87 / 150 (99th - 100th)	43 / 150 (31st - 33rd)
USNCO Local Section Exam 2022	36 / 60	38 / 60	24 / 60
Medical Knowledge Self-Assessment Program	75 %	75 %	53 %
Codeforces Rating	392 (below 5th)	392 (below 5th)	260 (below 5th)
AP Art History	5 (86th - 100th)	5 (86th - 100th)	5 (86th - 100th)
AP Biology	5 (85th - 100th)	5 (85th - 100th)	4 (62nd - 85th)

Visualization for DL

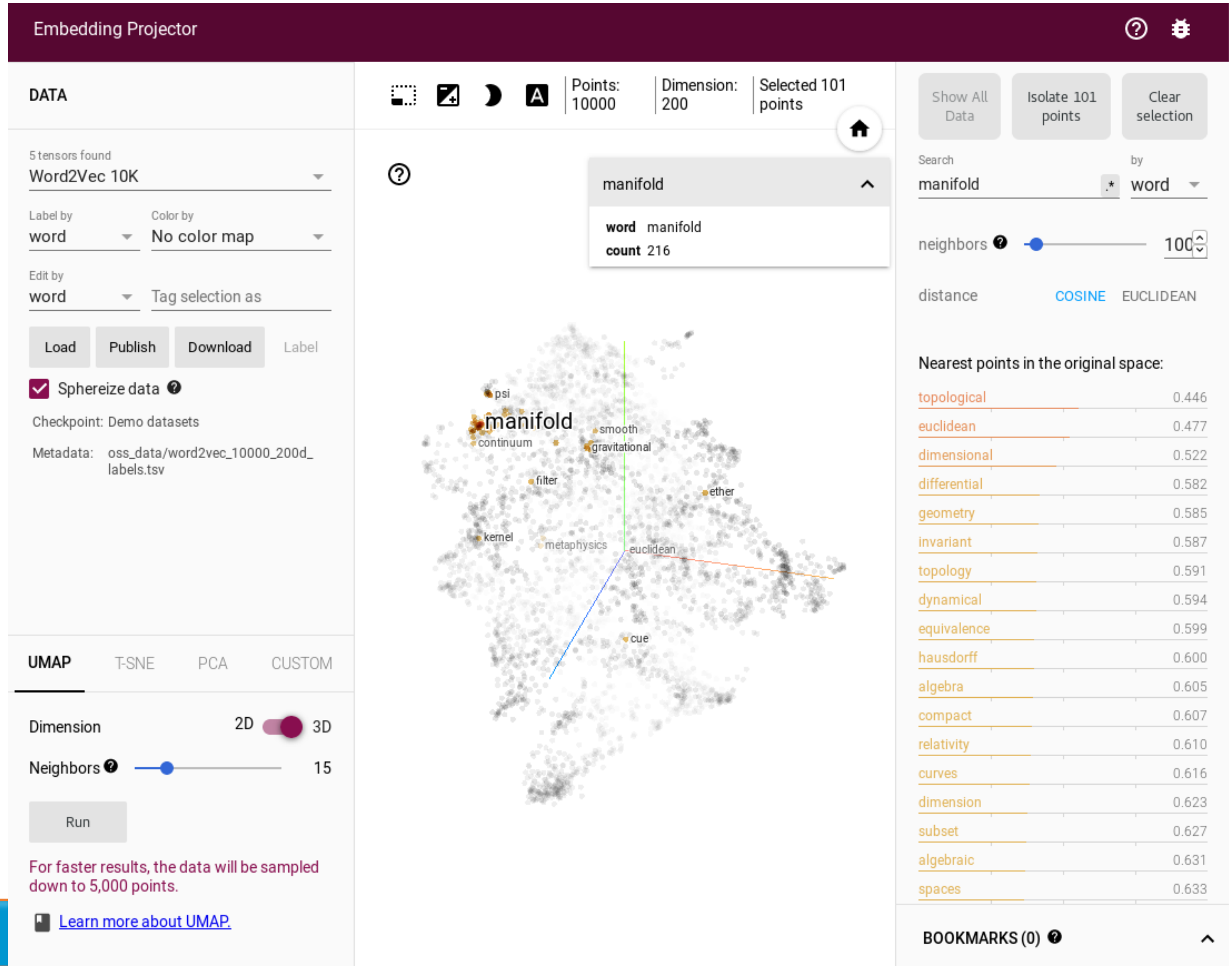
- **Tensorboard: Visualizing Learning**
- How to use t-SNE efficiently
- UMap

Model visualization

- **LSTM-Vis**: http://lstm.seas.harvard.edu/client/index.html
- Video demo
- Building blocks of interpretability

U-MAP

- Embed high-dimensional points in screen coordinates



Sources

- I. Goodfellow, Y. Bengio, A. Courville "Deep Learning" MIT Press 2016 [link]
- Zhang et al. "Dive into Deep Learning" [link]