Advanced Deep Learning: Recurrent Networks

Anders Søgaard

coastal

Course outline

Goal 1: Quick tour of recent developments in deep learning

Goal 2: Inspiration for thesis/research projects

Week	Lecturer	Subject	Literature	Assignment
1	Stefan	Introduction to Neural Networks.	d2l 2.1-2.5, 2.7, 11.5.1, **slides**	
2	Stefan	CNNs; FCNs; U-Nets. Data augmentation; invariance; regularization e.g. dropout	d2l 6, 7, **slides**	Assignment 1 (May 10)
3	Anders/Phillip	May 9 (A): RNNs May 11 (P): Transformers	d2l 8 Transformers: d2l 10.5-10.7 + Vaswani et al. (2017)	Assignment 2 (May 17)
4	Phillip/Anders	May 16 (P): Representation and Adversarial Learning May 18 (A): A Learning Framework + Self-supervised Learning + Contrastive Learning	Autoencoders: tba GANs: tba Self-supervised learning: blog post Contrastive learning: Dor et al. (2018)	
5	Anders	May 23-25: Applications of Representation, Adversarial and Contrastive Learning	GANs: Lample et al. (2018) Autoencoders: Chandar et al. (2011) Contrastive learning: Yu et al. (2018) Examples: Goodfellow et al. (2015) DynaBench: Talk by Douwe Kiela (Facebook, now HuggingFace)	Assignment 3 *[MC on Representation Learning/1p Report on Adversarial Learning]* (May 31)
6	Anders	May 30-June 1: Interpretability	Literature: Søgaard (2022)	
7	Anders	June 8: Interpretability (Note: June 6 off)	Literature: Feng and Boyd-Graber (2018) ; Jiang and Senge (2021)	
8	Anders	June 13-15: Best Practices	Literature: Dodge et al. (2019) and Raji et al. (2021)	Assignment 4 *[MC on Interpretability; 1p Report on Best Practices]* (June 21)

Course outline

Goal 1: Quick tour of recent developments in deep learning

Goal 2: Inspiration for thesis/research projects

Stefan: U-Nets+CNNs (1-2)

Phillip and I: a) RNNs and Transformers and how to train them (3-5); **b)** Interpretability and Best Practices (6-8)

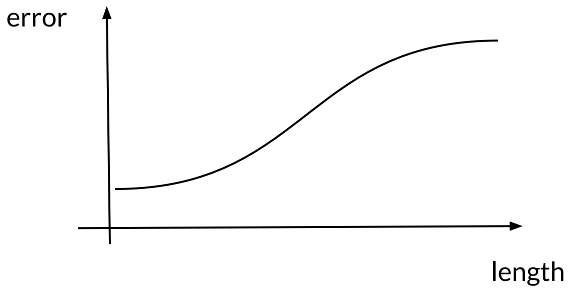
Week	Lecturer	Subject	Literature	Assignment
1	Stefan	Introduction to Neural Networks.	d2l 2.1-2.5, 2.7, 11.5.1, **slides**	
2	Stefan	CNNs; FCNs; U-Nets. Data augmentation; invariance; regularization e.g. dropout	d2l 6, 7, **slides**	Assignment 1 (May 10)
3	Anders/Phillip	May 9 (A): RNNs May 11 (P): Transformers	d2l 8 Transformers: d2l 10.5-10.7 + Vaswani et al. (2017)	Assignment 2 (May 17)
4	Phillip/Anders	May 16 (P): Representation and Adversarial Learning May 18 (A): A Learning Framework + Self-supervised Learning + Contrastive Learning	Autoencoders: tba GANs: tba Self-supervised learning: blog post Contrastive learning: Dor et al. (2018)	
5	Anders	May 23-25: Applications of Representation, Adversarial and Contrastive Learning	GANs: Lample et al. (2018) Autoencoders: Chandar et al. (2011) Contrastive learning: Yu et al. (2018) Examples: Goodfellow et al. (2015) DynaBench: Talk by Douwe Kiela (Facebook, now HuggingFace)	Assignment 3 *[MC on Representation Learning/1p Report on Adversarial Learning]* (May 31)
6	Anders	May 30-June 1: Interpretability	Literature: Søgaard (2022)	
7	Anders	June 8: Interpretability (Note: June 6 off)	Literature: Feng and Boyd-Graber (2018) ; Jiang and Senge (2021)	
8	Anders	June 13-15: Best Practices	Literature: Dodge et al. (2019) and Raji et al. (2021)	Assignment 4 *[MC on Interpretability; 1p Report on Best Practices]* (June 21)

Exercise (May 17)

a) Multiple choice (Google Form)
b) 1 page report: Compare RNNs and LSTMs on $a^n b^n$ and $a^n b^n c^n$
c) ~~Code piece: RNN implementation~~

Exercise (May 17)

a) Multiple choice (Google Form)
b) 1 page report: Compare RNNs and LSTMs on $a^n b^n$ and $a^n b^n c^n$
~~c) Code piece: RNN implementation~~



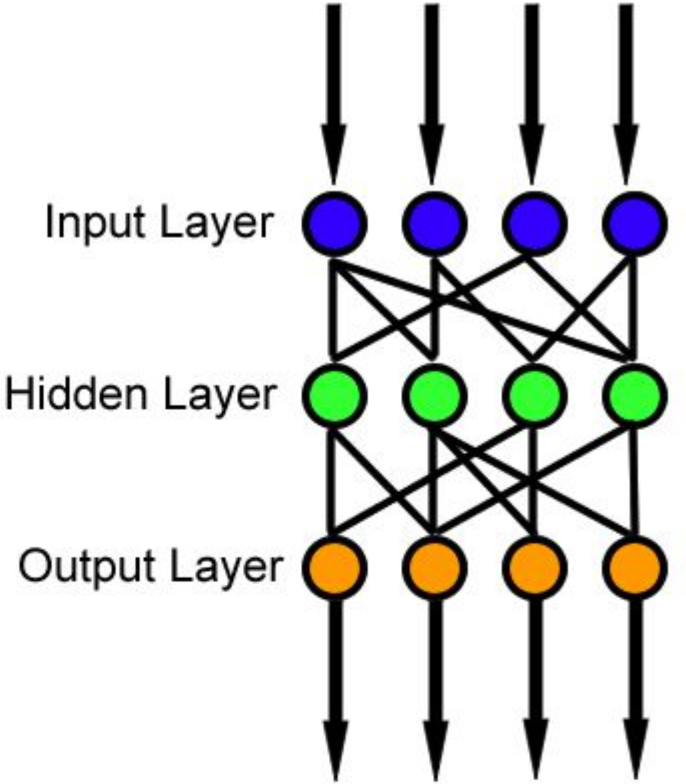
Today

a) The Deep Learning Landscape
b) Applications
c) Recurrency
d) Historical context

Landscape

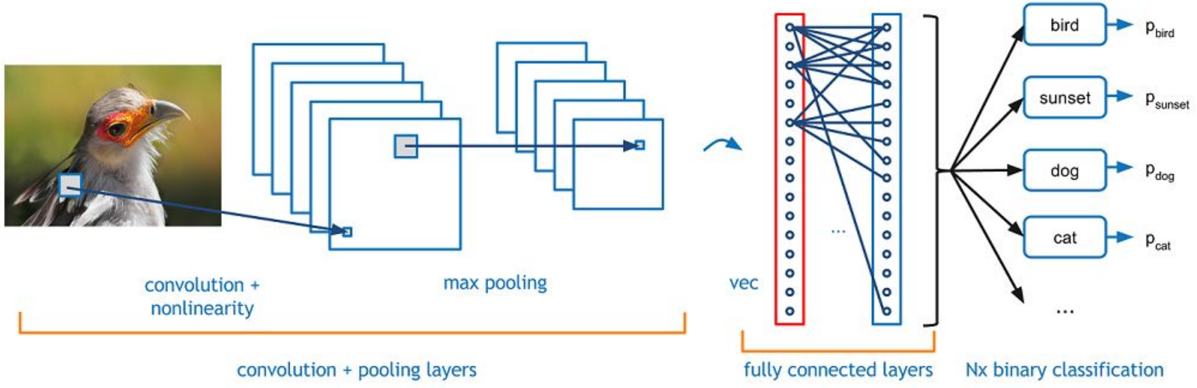
Feed-forward Networks

FFNs are networks of perceptrons or logistic regressors feeding into other perceptrons or logistic regressors. In the so-called *forward* step, that's the whole story. For training, you need to do a more complicated *backward* step to compute the appropriate weight updates. This process is called *back-propagation*.



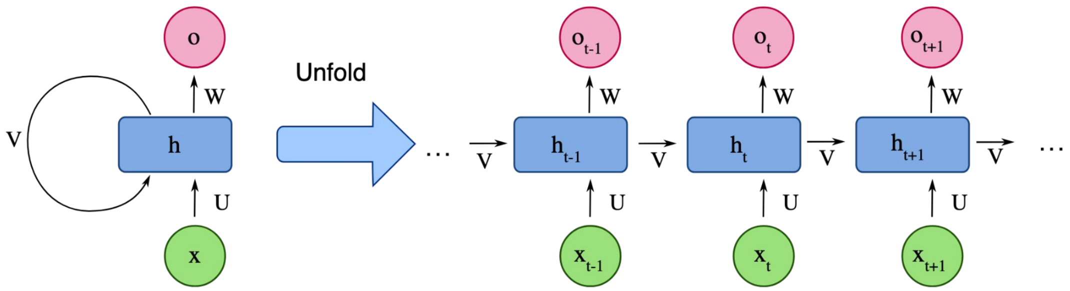
Convolutional Neural Networks

Adds special early layers to FNNs that account for the invariance properties of images.



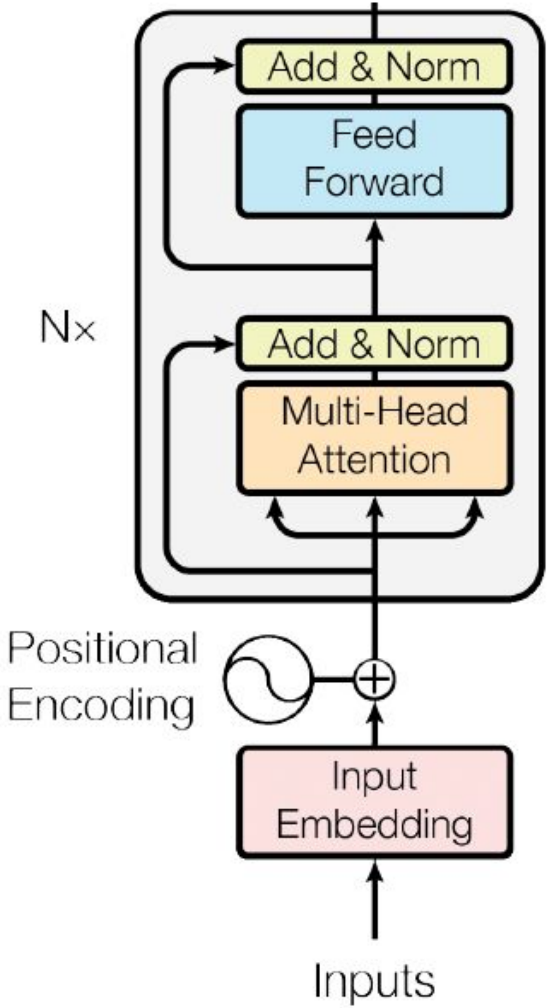
Recurrent Neural Networks

You can think of an RNN as a FNN with *n* layers used to process a sequence of *n* tokens, but in which the *n* layers are all the same. **Note:** For a deep k-layer RNN, this would be *kn* layers with the *n* combinations of *k* layers being the same.



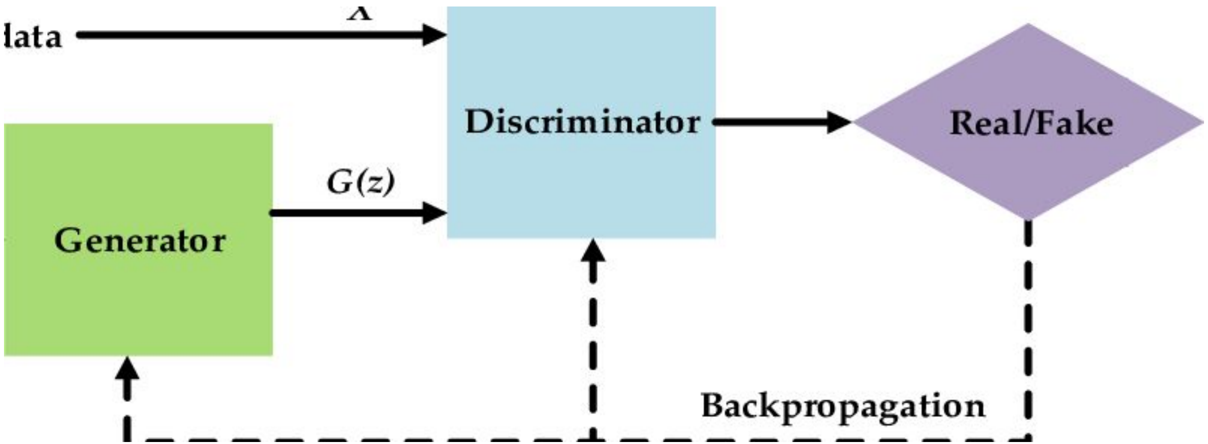
Transformers

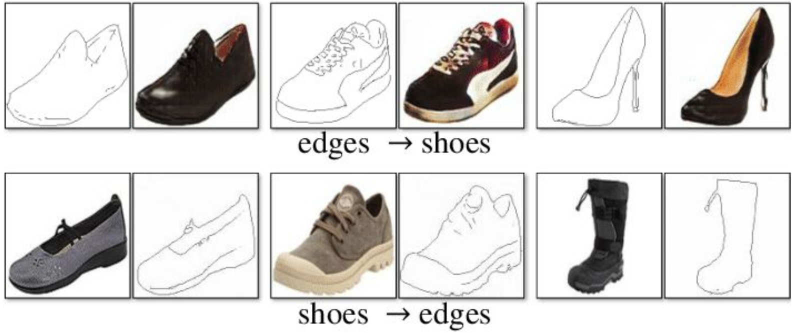
Improvement over RNNs, specialized for GPUs. Processes tokens in parallel, but considers their interaction with all other tokens (making inference quadratic-time).



Generative Adversarial Networks

Combines two networks, e.g., a FNN and an RNN to transform data points from one distribution so they look like data points from another distribution.

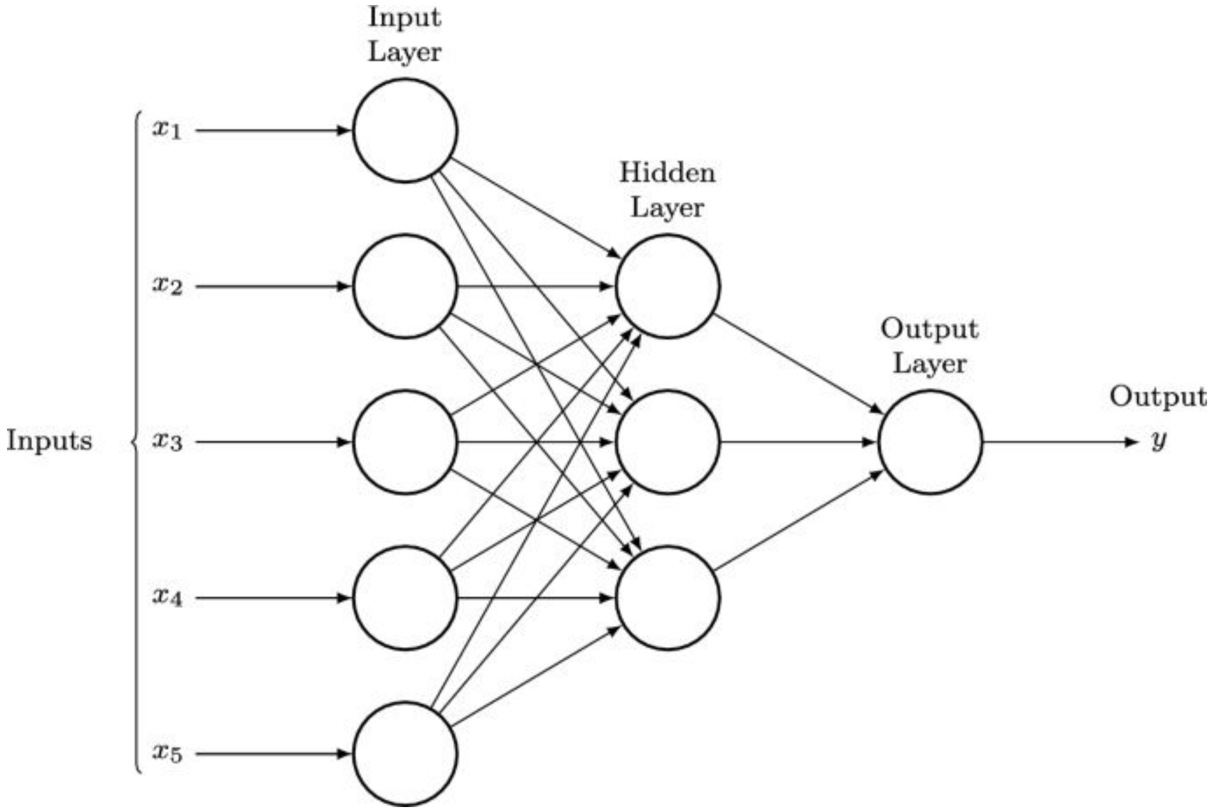




Applications

Sequential data

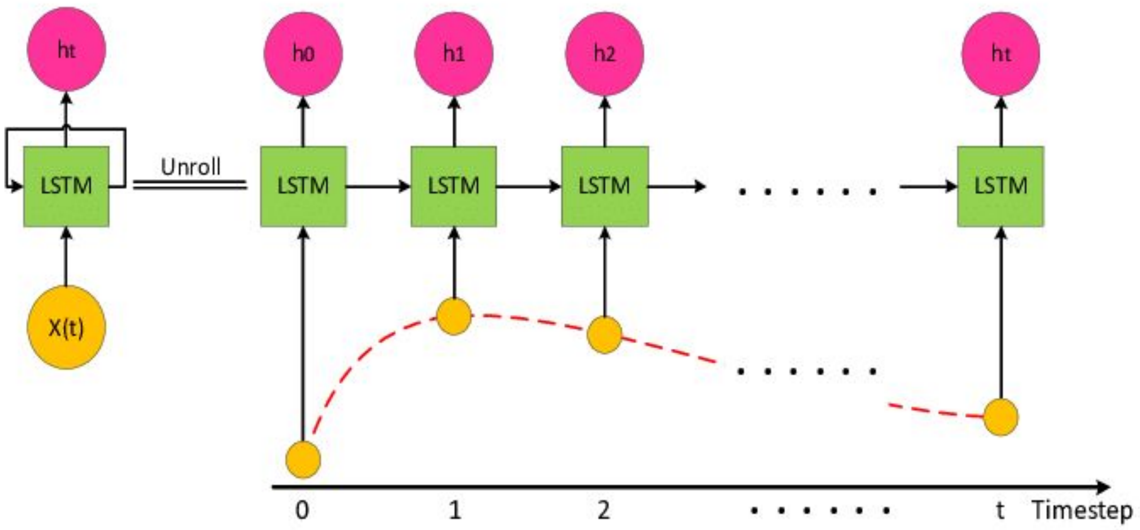
- Bag-of-*n*-grams with feed-forward networks are limited by a poor bias-variance trade-off.
- So are 1D CNNs.
- Pre-deep models with slightly better trade-offs: HMMs, CRFs, and HMM-Perceptrons.

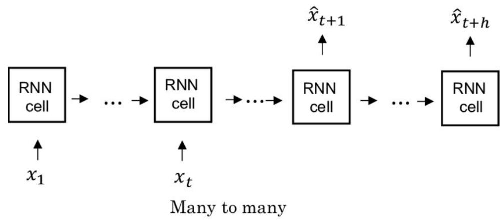


Time series data

Predicting trends or developments over time.

a) Sales
b) Climate
c) Socio-economic variables
d) …



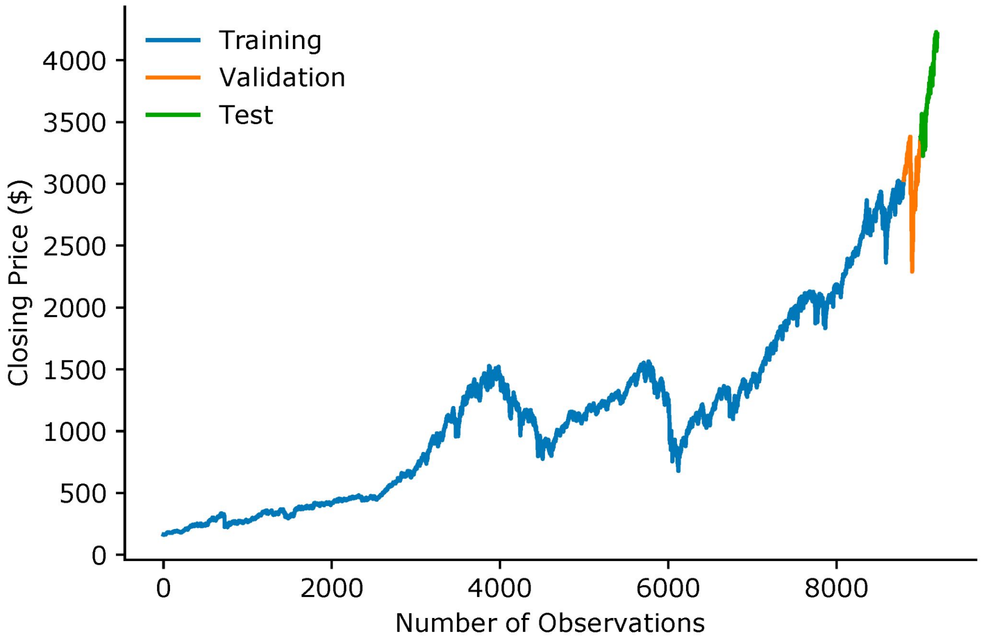


—

Time series data

Predicting trends or developments over time.

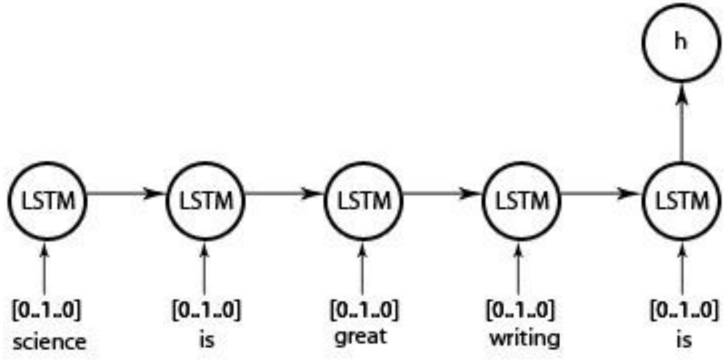
a) Sales
b) Climate
c) Socio-economic variables
d) …

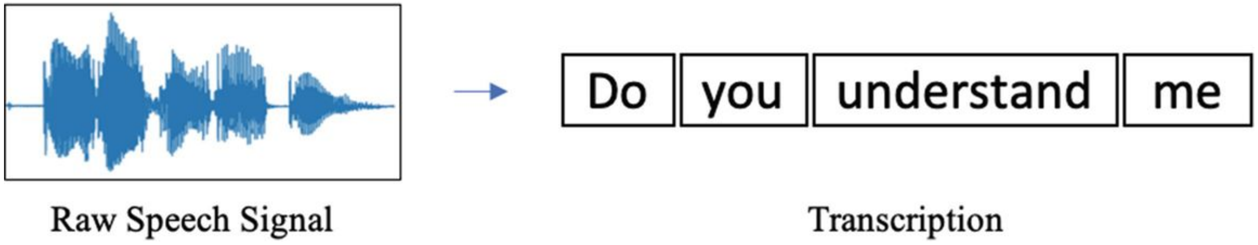


Sentence or document classification

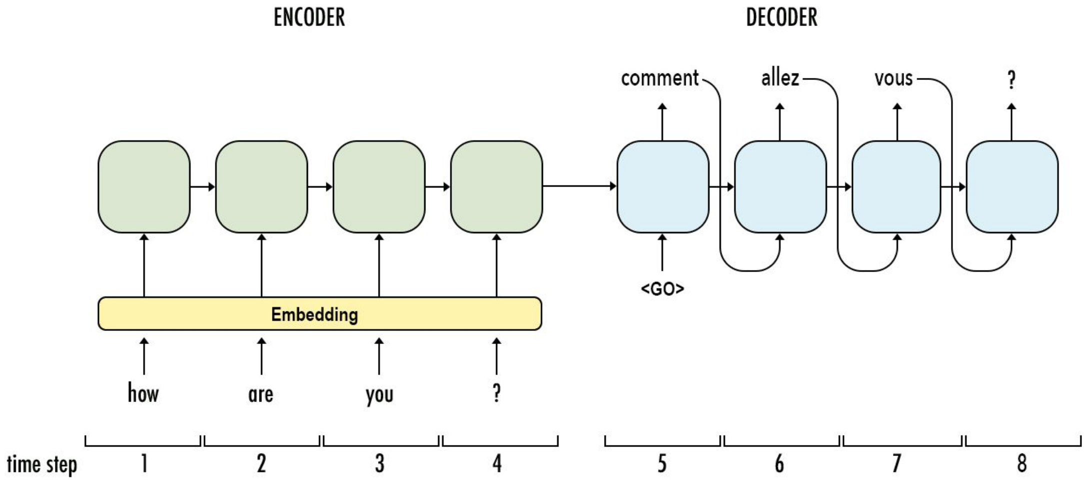
Sentences and documents come in different lengths.

a) Sparse, e.g., few sentences of length 54
b) Unbounded, e.g., may always potentially see a longer sentence

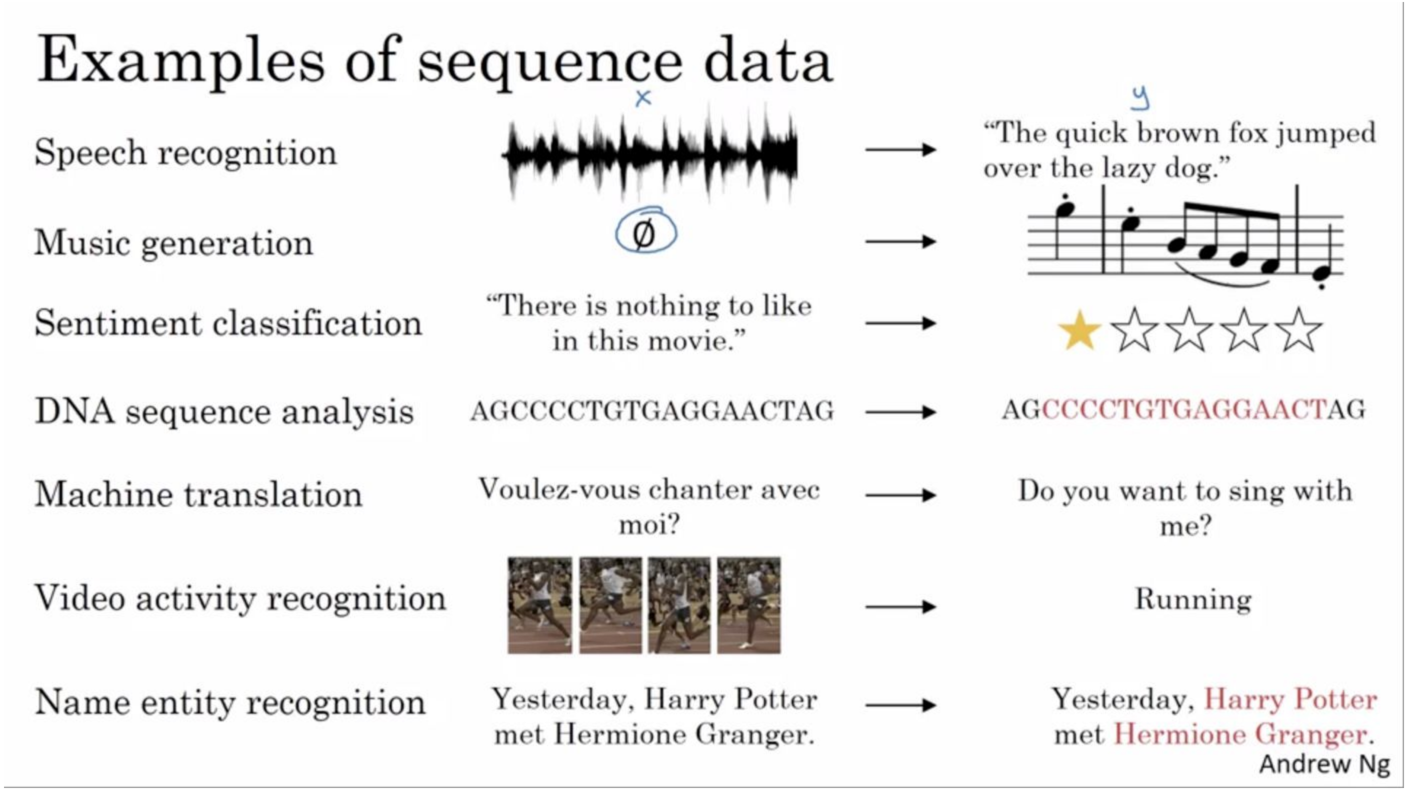




Speech recognition



Machine translation

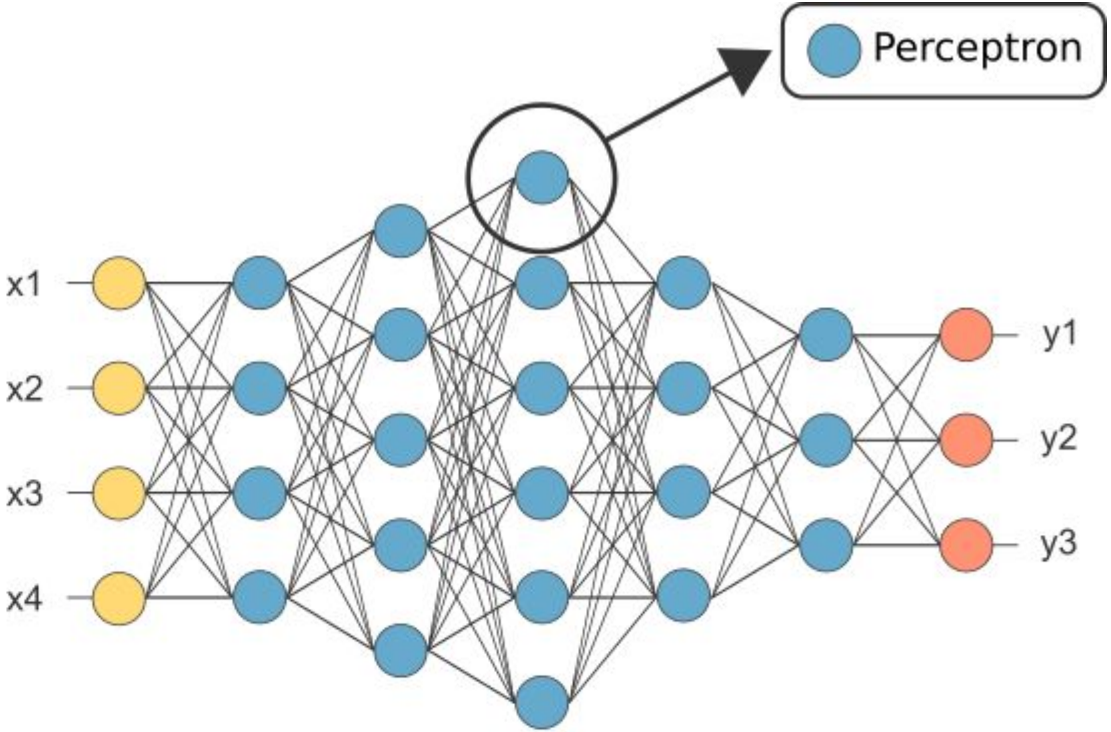


Other examples

Recurrency

Feed Forward

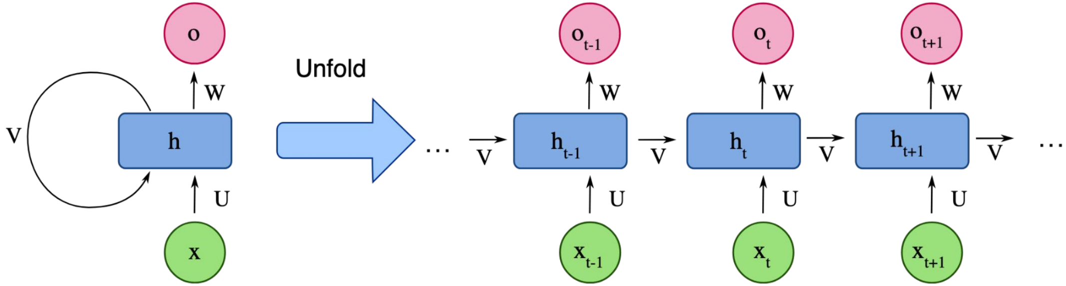
a) Interconnected neurons
b) *m* many layers
c) But can't model *n* length sequences in an unbiased way



—

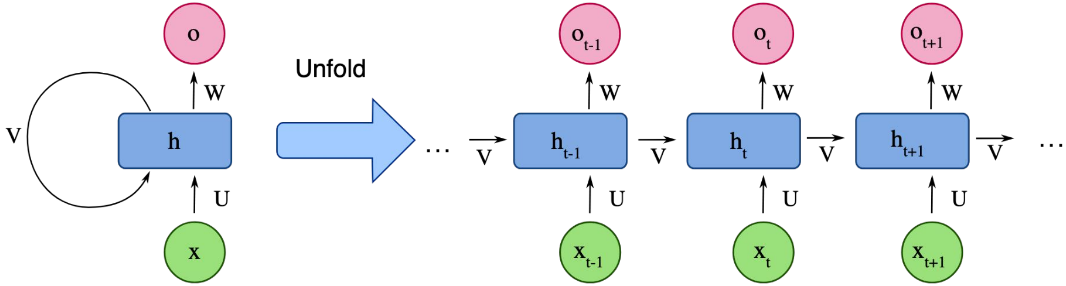
Recurrent

But what if we used some of the hidden layers *recurrently*?



Recurrent

But what if we used some of the hidden layers *recurrently*?



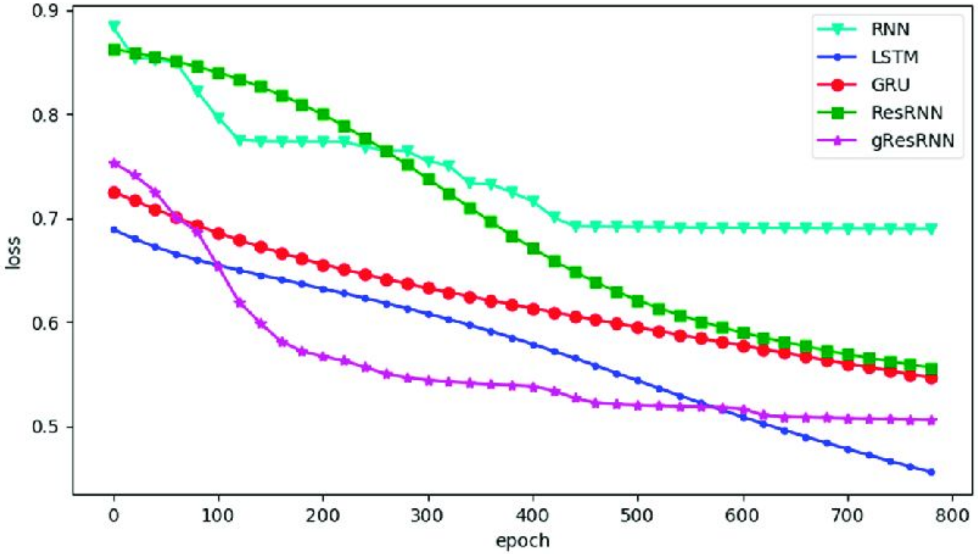
$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \mathrm{softmax}(\boldsymbol{o}^{(t)})
\end{aligned}
$$

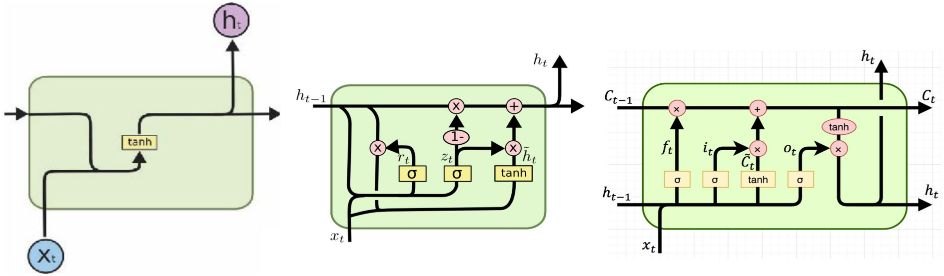
—

Beyond RNNs

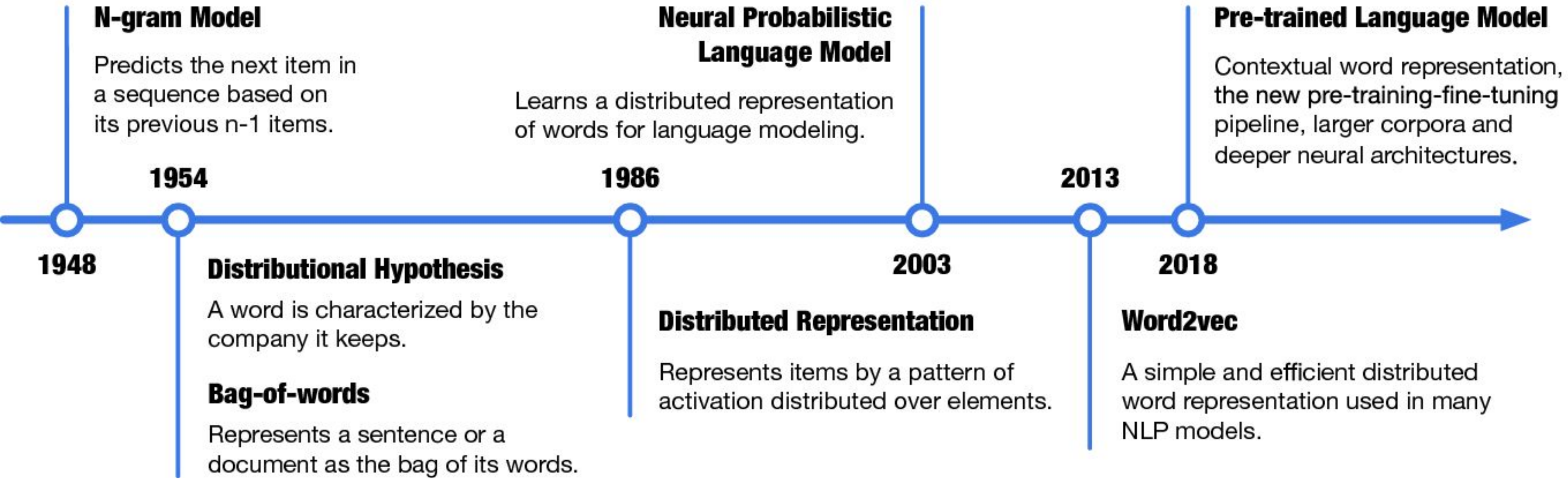
a) RNNs are basically stacked FNNs with parameter sharing.
b) GRUs introduce update gates.
c) LSTMs also introduce forget and output gates.

More control prevents vanishing gradients and enables more efficient induction of long-distance dependencies.

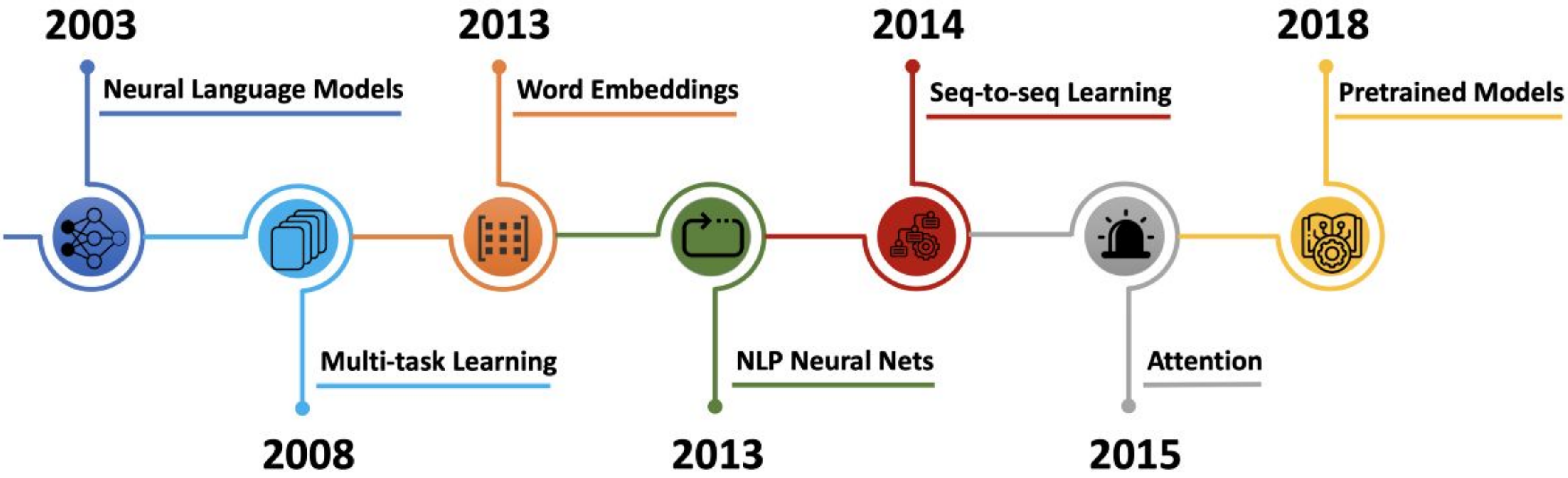




Historical context



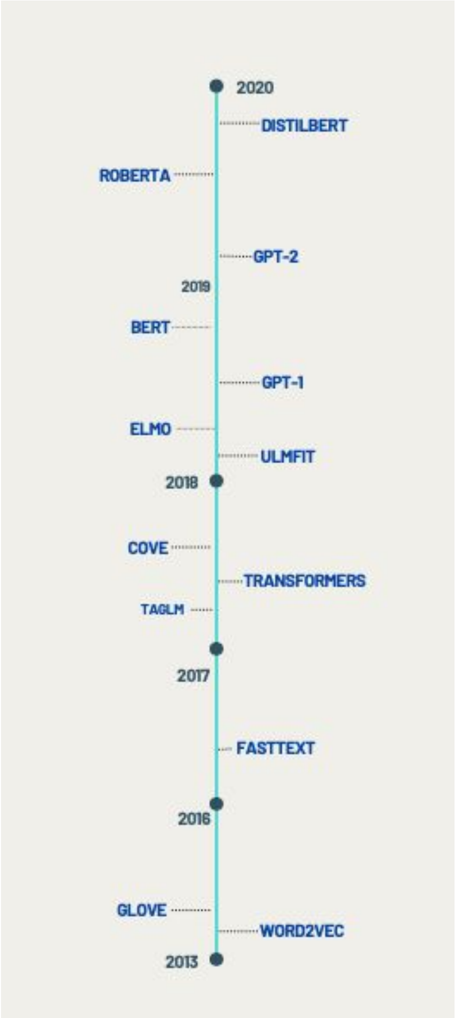
History of NLP



History of NLP

Pretrained models

a) Early generations (w2v, fastText, etc) built on feed forward networks
b) Intermediate models were based on RNNs (ELMO, GPT-1, etc)
c) Later generations (GPT-3, BART, Pegasus, RoBERTa, etc) are based on Transformers (**next time**)



	RNNs	GRUs/LSTMs	Transformers
NLP	2010	2013	2017
Computer Vision	(Video)	(Video)	ViT (2021) TrOCR (2021)

History of NLP and Computer Vision