Sequence Modelling and Recurrent Neural Networks (RNNs)

Pattern Recognition

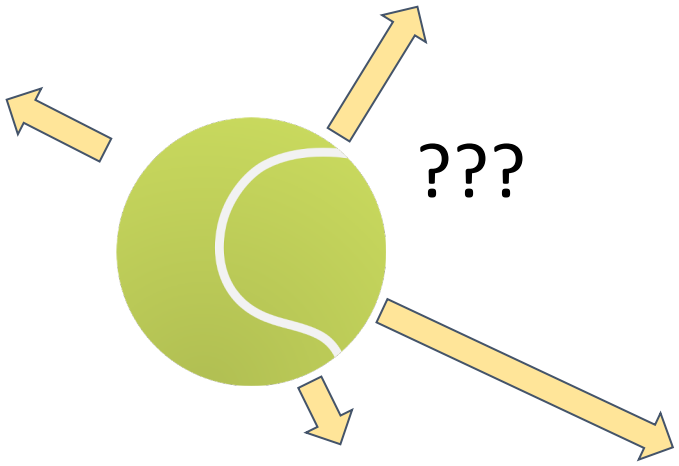
Dennis Madsen

Topic overview

- *Neural Networks (recap) and Deep Learning*
- *Improving DNN: Hyperparameter tuning, regularization, optimization*
- *Convolutional Neural Networks (CNN)*
- *CNN popular architectures*
- **Sequence Models/Recurrent neural networks (RNN)**
  - RNN architecture
  - Gated cells for long term dependencies
  - Natural Language Processing (NLP)
  - Transformer networks
- **Beyond the basics**
  - Neural networks as a generative model

Motivation

Given a single image instance of a ball - can we predict its direction?

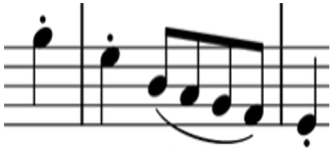


Motivation

Given a single image instance of a ball - can we predict its direction?

Given enough previous placements, it is possible to predict direction and magnitude

Examples of sequence data

Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	★☆☆☆☆
Machine translation	Vil du synge med mig?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

All of these problems can be addressed using supervised learning with labeled data

Source: https://www.coursera.org/learn/nlp-sequence-models/

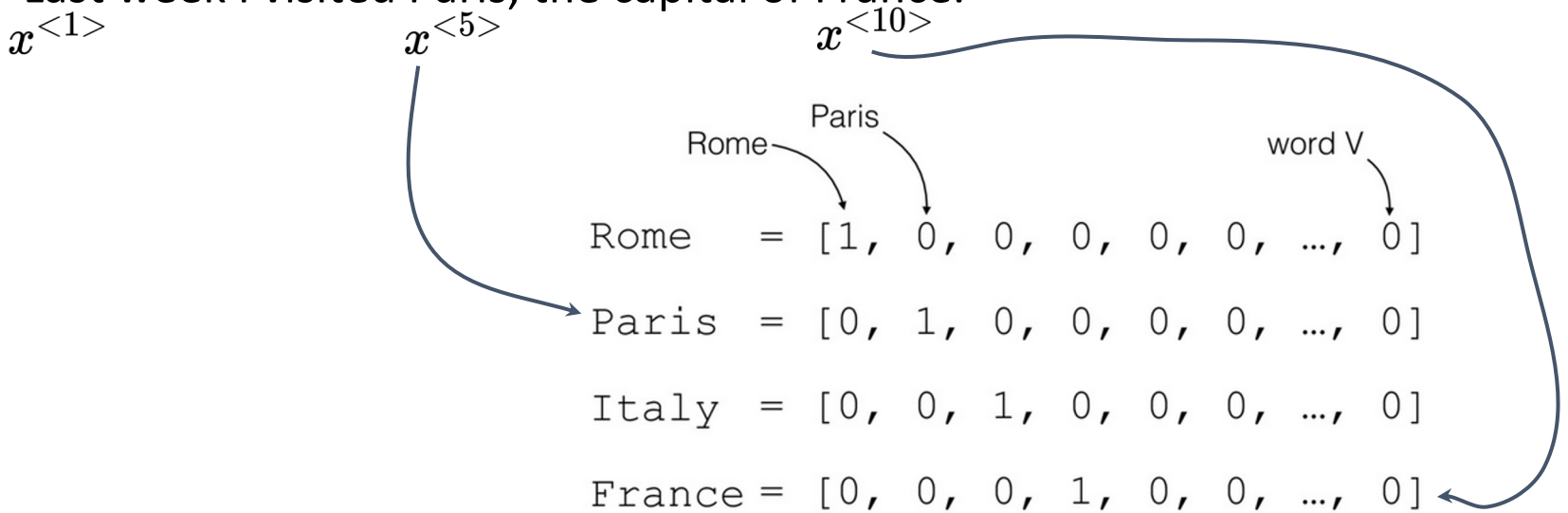
Words: One-hot representation

Words are represented as a one-hot feature representation.

The vocabulary is therefore fixed with words representing a single entry in a vector.

For commercial applications, vocabularies of 30-50.000 words are often used.

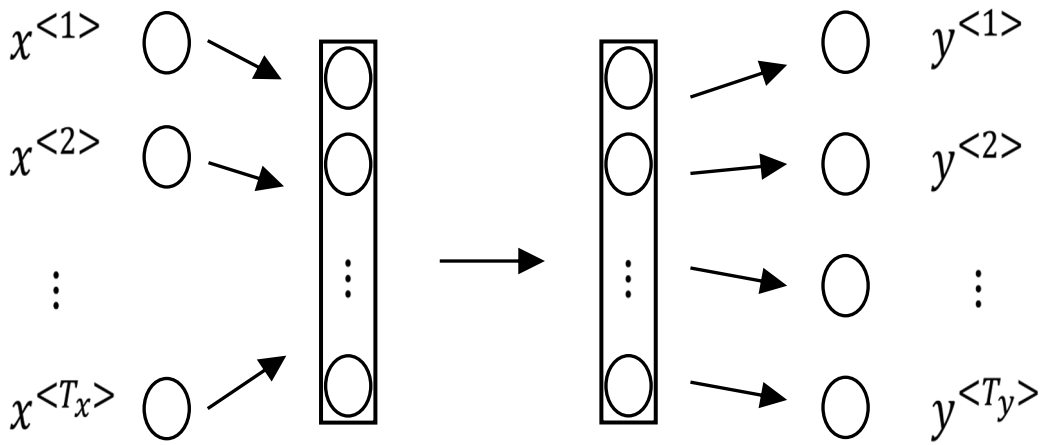
x: "Last week I visited Paris, the capital of France."



Why not a standard fully connected NN?

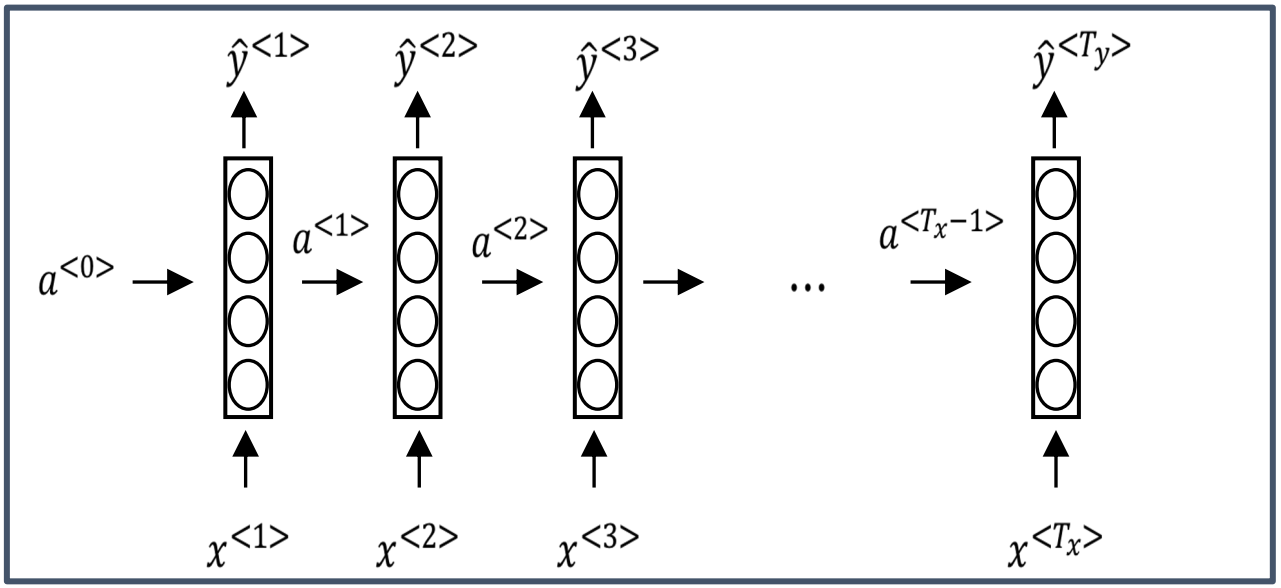
Problems:

- Input/outputs can be different lengths in different examples.
  - Example: language translation doesn't happen word to word.
- Does not share features across different locations (bag-of-words cannot be used).
  - The food was good, not bad at all (positive).
  - The food was bad, not good at all (negative).

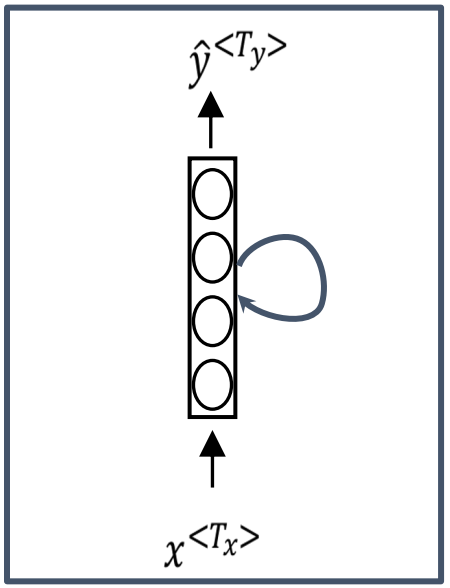


Source: https://www.coursera.org/learn/nlp-sequence-models/

Recurrent Neural network structure



Unrolled representation



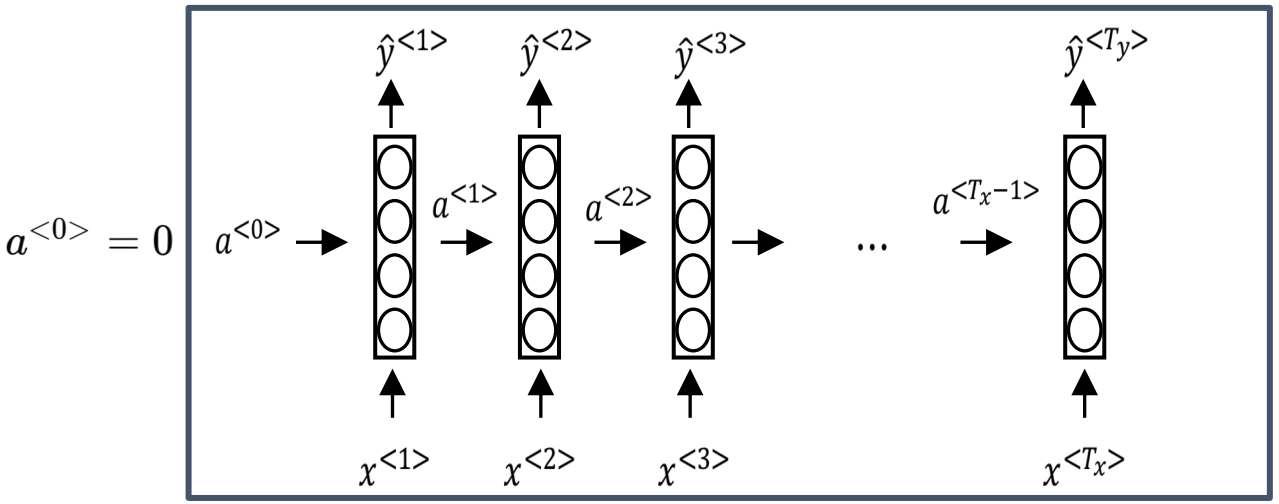
Compact representation

Source: https://www.coursera.org/learn/nlp-sequence-models/

RNN notation

- g() is the activation function such as: tanh, sigmoid, ReLU.

$$a^{<1>} = g(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a)$$
$$\hat{y}^{<1>} = g(W_{ya}a^{<1>} + b_y)$$



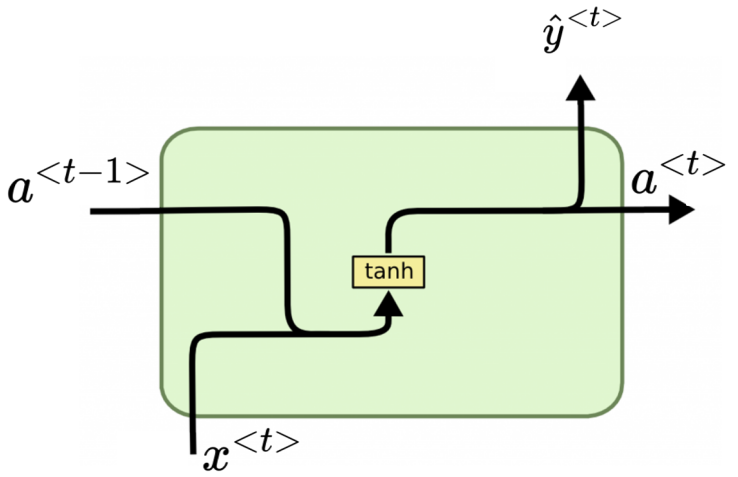
$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \qquad W_a = [W_{aa}|W_{ax}]$$
$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$$

- First subscript of w defines the output of the multiplication.
- Second subscript of w defines what it is being multiplied with.

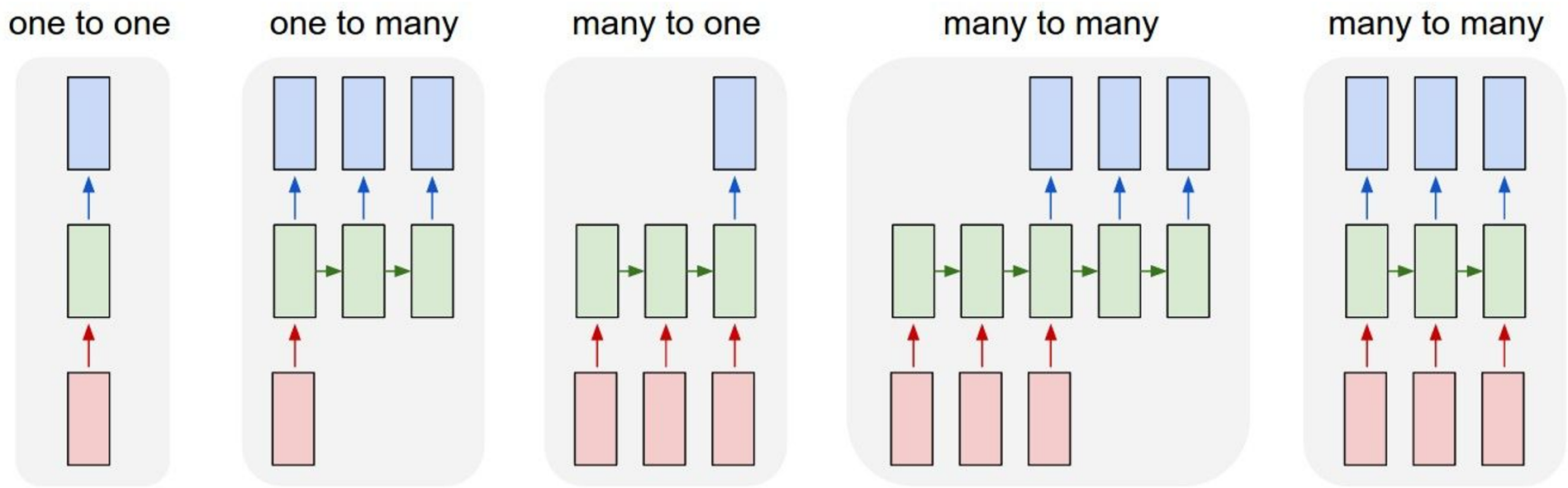
Source: https://www.coursera.org/learn/nlp-sequence-models/

Types of RNNs

- One to One: "Vanilla" neural network.
- One to Many: Music generation.
- Many to One: Sentiment classification.
- Many to Many: Translation.

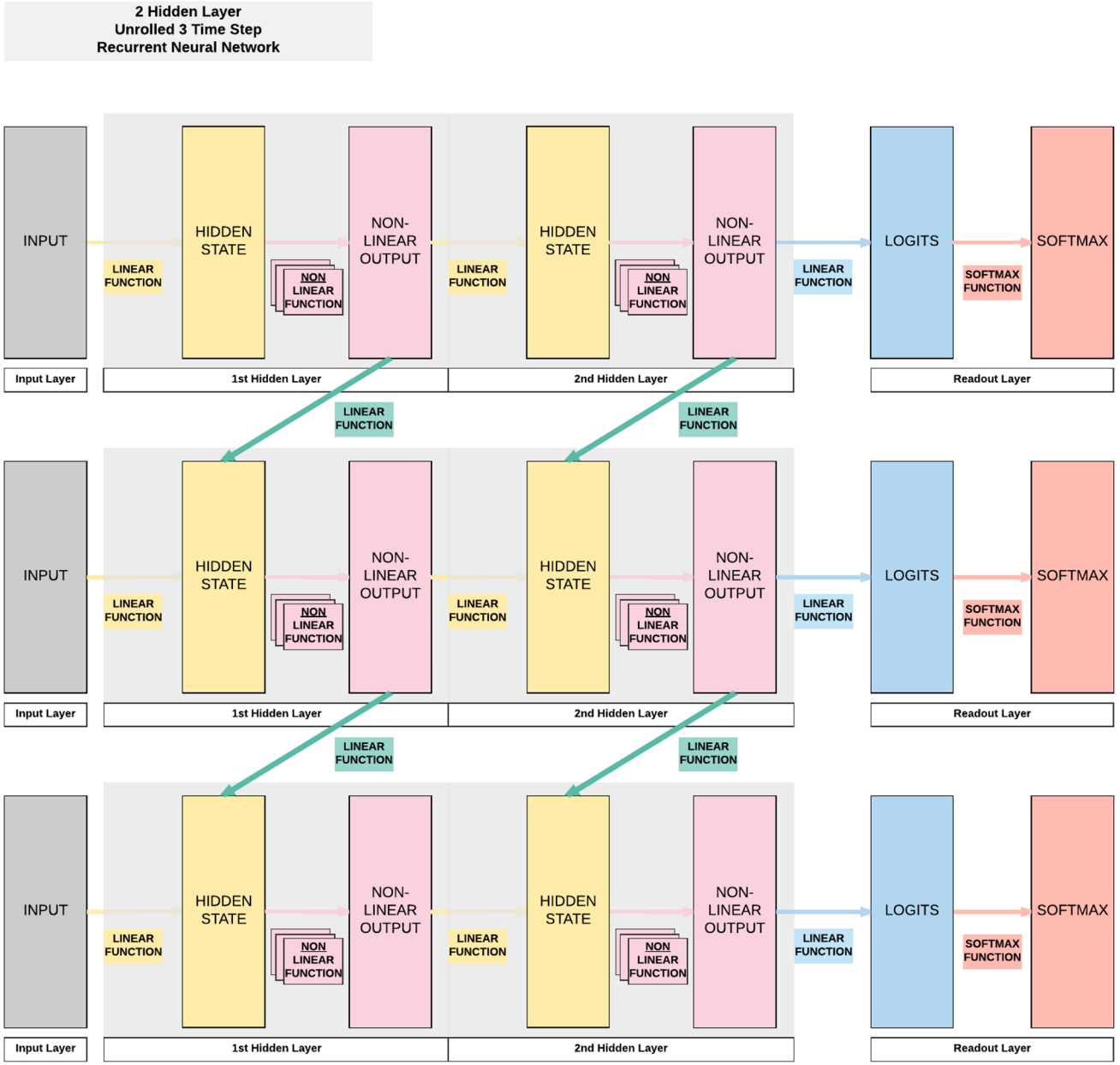


$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$



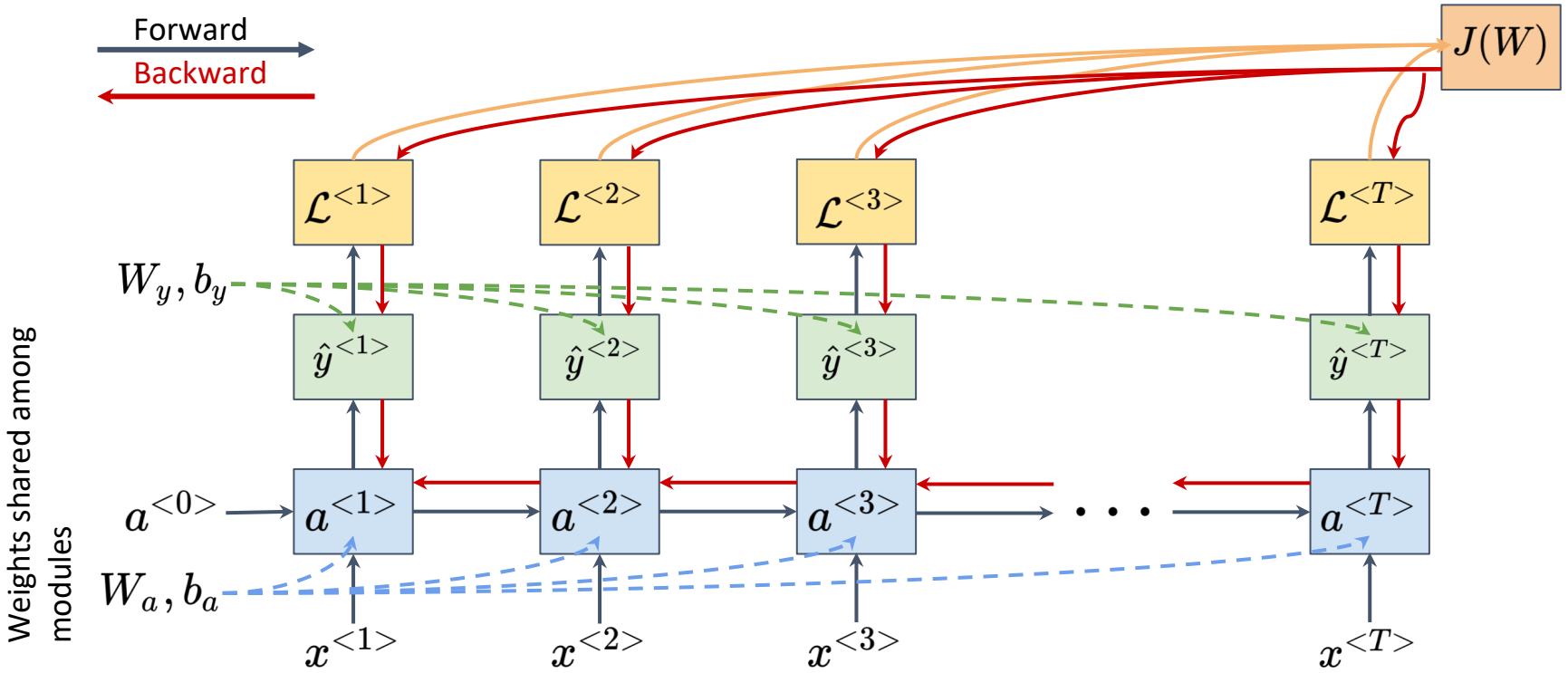
Source: http://karpathy.github.io/

Deep RNN example



Source: https://www.deeplearningwizard.com/deep_learning/

Backpropagation through time!



$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = y^{(i)} log(f(x^{(i)})) + (1 - y^{(i)}) log(1 - f(x^{(i)}))$$ Cross-entropy loss for each output

$$J(W) = \sum_{t=1}^{T} \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$ Total cost

Source: https://www.coursera.org/learn/nlp-sequence-models/

RNN problems - vanishing gradients

Long term feature dependencies are very difficult to learn with a standard RNN.

- *"The cat, which already ate …, **was** full."*
- *"The cats, which already ate …, **were** full."*

Vanishing/Exploding gradients

Solutions:

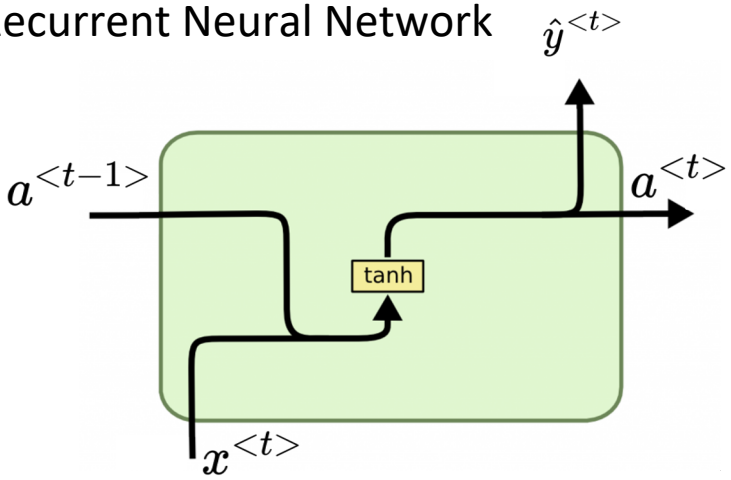
1. Using ReLU activation functions to prevent shrinking the gradients.
2. Initialize the weights to the identity matrix
    a. Biases still initialized to zero
3. Use a more complex recurrent unit with gates to control what information is passed through.

Gated cells
LSTM, GRU, etc.

Source: http://karpathy.github.io/

RNN

Recurrent Neural Network

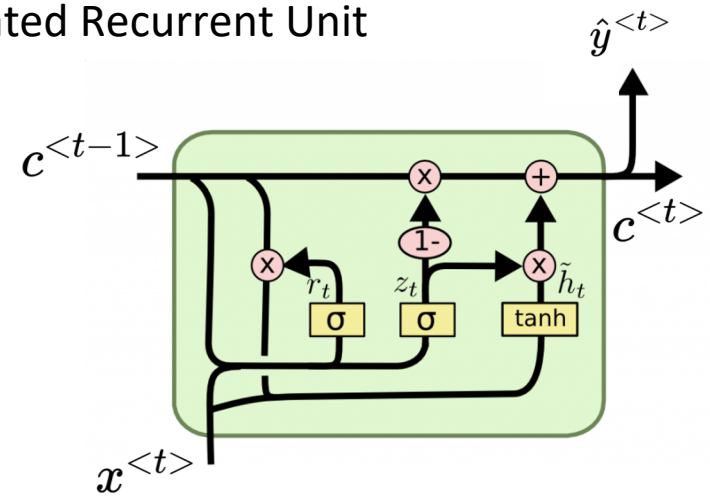


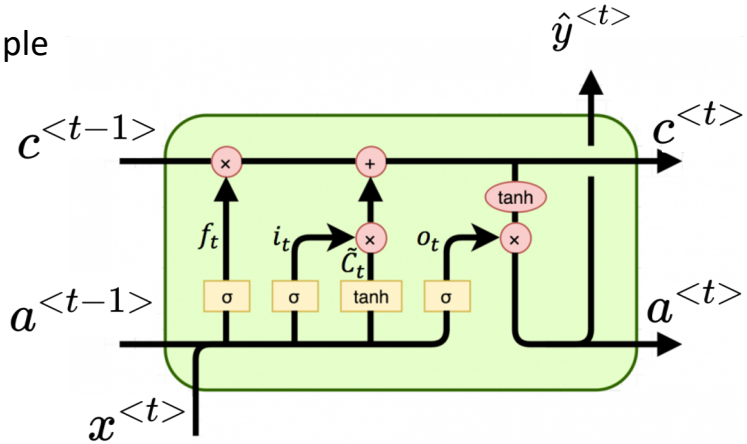
$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

g() = tanh is the above example

GRU

Gated Recurrent Unit





LSTM

Long-short term memory

Gated Recurrent Unit (GRU)

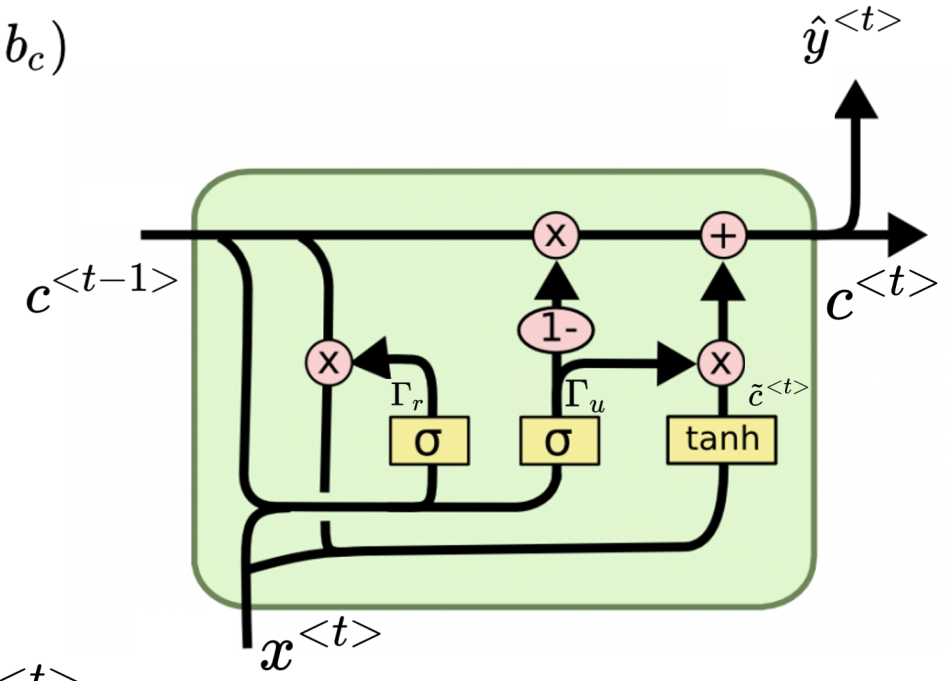
C = memory cell (remember singular or plural).

$$c^{<t>} = a^{<t>}$$
$$\tilde{c}^{<t>} = tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$$
$$\Gamma_u = \sigma(W_u[c^{t-1}, x^t] + b_u)$$
$$\Gamma_r = \sigma(W_r[c^{t-1}, x^t] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

$\Gamma_u = 1$ $\Gamma_u = 0$ $\Gamma_u = 1$

"The **cat,** which already ate …, **was** full."



Update gate: Γ_u decides when to update $c^{<t>}$

Relevance gate: Γ_r how relevant is the last feature in computing the next output?

Intuition: Gates are always on or off. They are modelled with a sigmoid function, so in practice very close to 0 or 1.

* is element-wise vector multiplication.

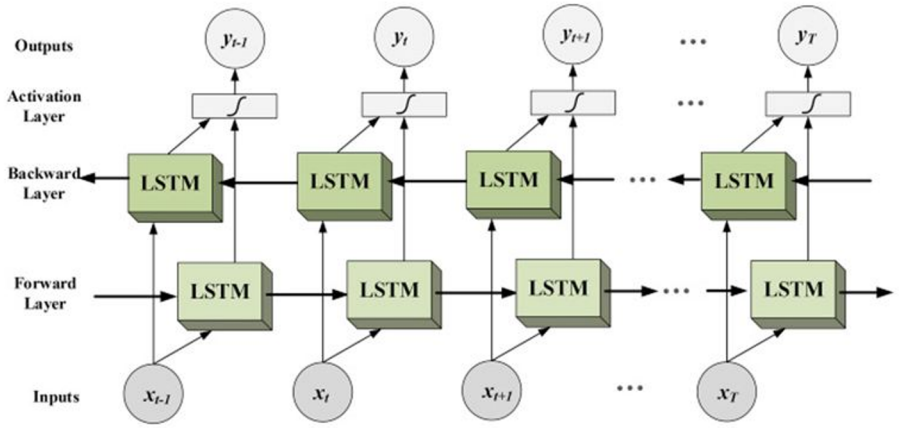
Gated cells problems

Forward propagating a sequence model only uses earlier information to predict context information.

Example, is Teddy a name?

- *He said, "Teddy Roosevelt was a great President."*
- *He said, "Teddy bears are on sale!"*

One solution is to use a *Bidirectional RNN* structure



Source: https://www.i2tutorials.com/

Still problematic - Words in context

Danish to english translation

- *Hans* is a name but does also mean *his.*
- *Ged* means *goat*, but often used when something went *wrong.*
- *Regner* can be both *rains* and *calculates/computes.*
- *På spanden* literally means *on the bucket* but often used to say *in trouble.*

There has been a goat in the budget because Hans is raining badly, so in short he is on the bucket for the rest of the month.

2020 update

A note on Natural Language Processing

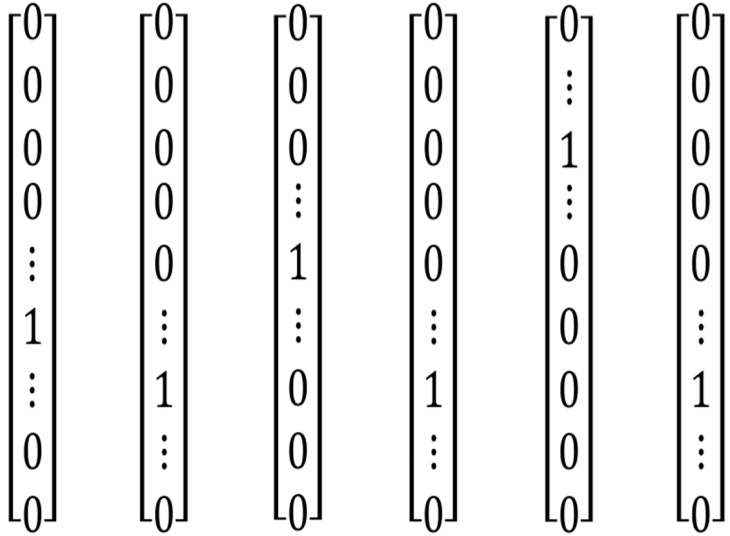
Generalizing from one example of "apple juice" to another of "orange juice" is not more intuitive than to "orange man".

I want a glass of orange _____ .

I want a glass of apple _____ .

1-hot representation.

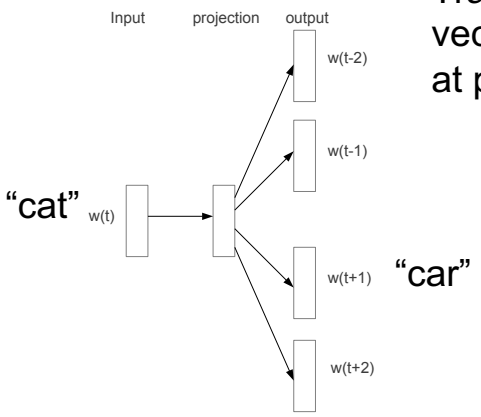
Man (5391) Woman (9853) King (4914) Queen (7157) Apple (456) Orange (6257)



	Man	Woman	King	Queen	Apple	Orange
Gender	-1.0	1.0	-0.95	0.97	0.0	0.01
Royal	0.0	0.0				
Age						
Food					0.95	0.97
...						

Source: https://www.coursera.org/learn/nlp-sequence-models/

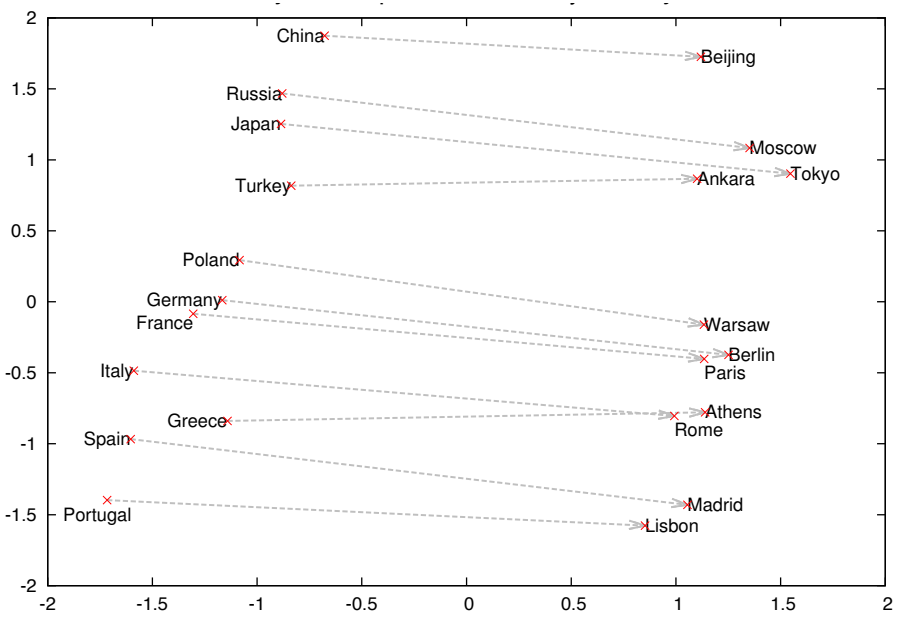
Word embedding word2vec



Training objective is to learn word vector representations that are good at predicting the nearby words.

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Model works as a lookup table



“Madrid”-”Spain”+”France” = “Paris” (closest vector)

Source: Distributed Representations of Words and Phrases and their Compositionality (NIPS2013)

Transformers

One of the main problems with Gated cells is their lack of parallelism.

- A sentence is processed one word at the time.
- Makes both training and inference slow.

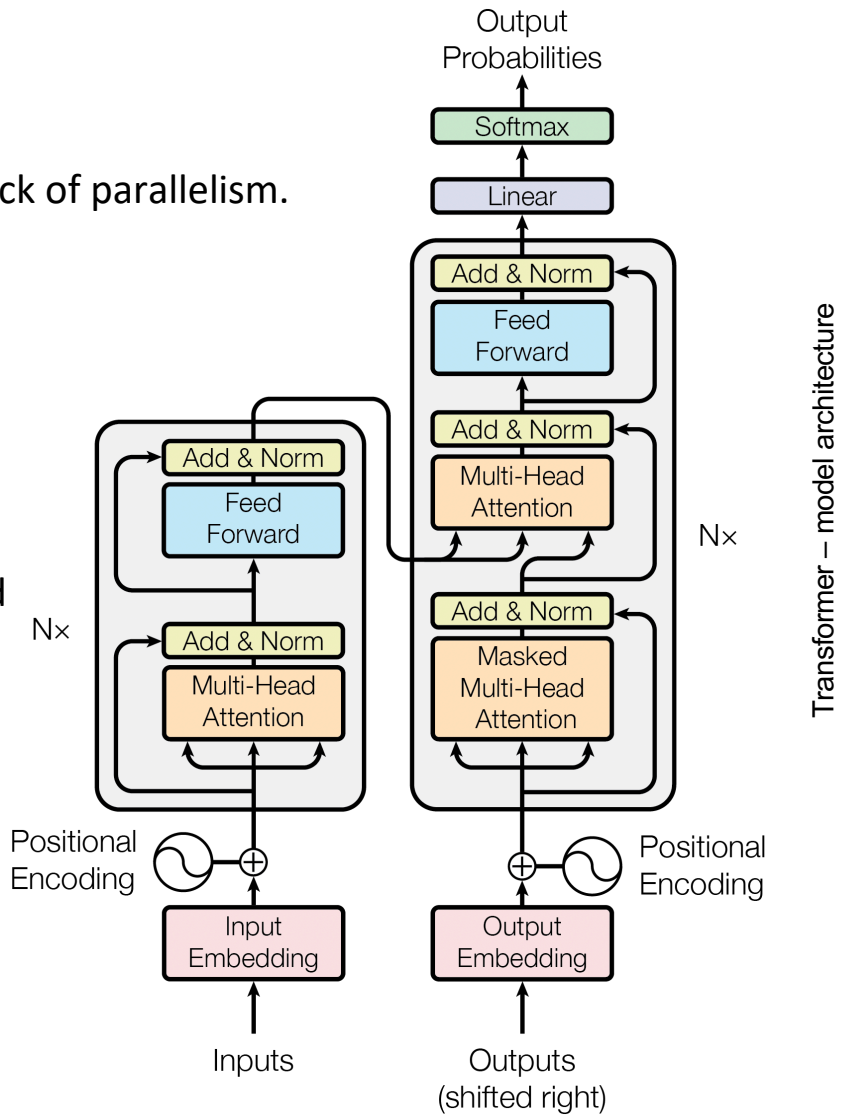
Transformers can process data in parallel

- Uses an encoder-decoder architecture
- 6 encoders + 6 decoders is setup in parallel
- Attention/importance for each word is computed

Sample sentence: “The cat is brown”

The	0.80	0.13	0.04	0.03
cat	0.2	0.62	0.08	0.10
is	0.05	0.25	0.42	0.28
brown	0.04	0.33	0.12	0.51

Average attention vectors over N parallel systems



Source: Attention is all you need (NIPS 2017)

Sequence modelling summary

- RNNs can be used to model sequence tasks.
- Model sequences are traditionally modelled via a recurrence relation.
- Training RNNs can be done with back-propagation through time and a gradient based optimizer.
- Gated cells like GRU let us model (reasonable) long-term dependencies.
- RNN networks suffer from slow training and inference time.
- Transformer networks works in parallel and can model very long-term dependencies.

Generative Models

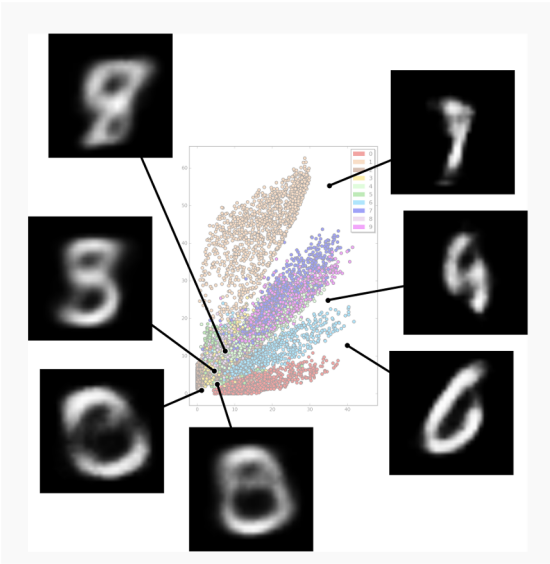
Variational Autoencoders (VAEs)

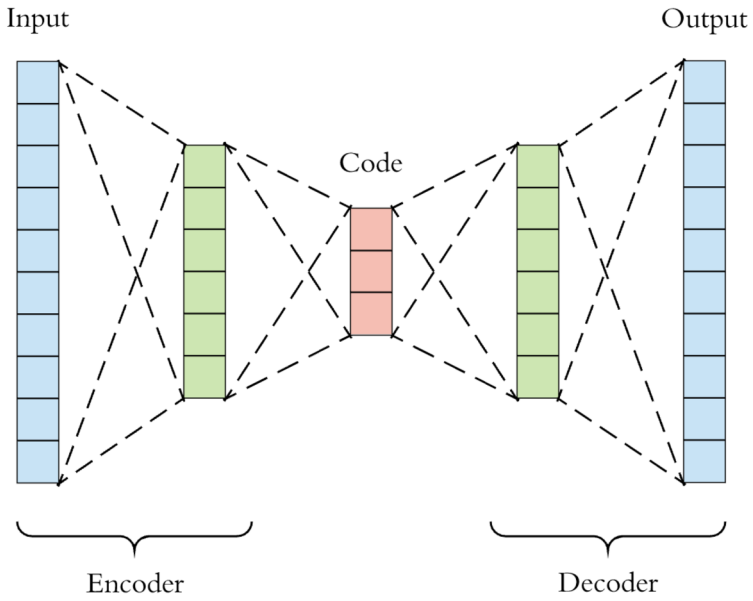
Generative Adversarial Networks (GANs)

Transformer network implementations (BERT, GPT-3)

Autoencoders

- Basic Autoencoder network
  - With linear activation functions, this is similar to Principal Component Analysis (PCA).





$$\phi : \mathcal{X} \to \mathcal{F}$$
$$\psi : \mathcal{F} \to \mathcal{X}$$
$$\phi, \psi = \arg\min_{\phi,\psi} \|X - (\psi \circ \phi)X\|^2$$

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2$$

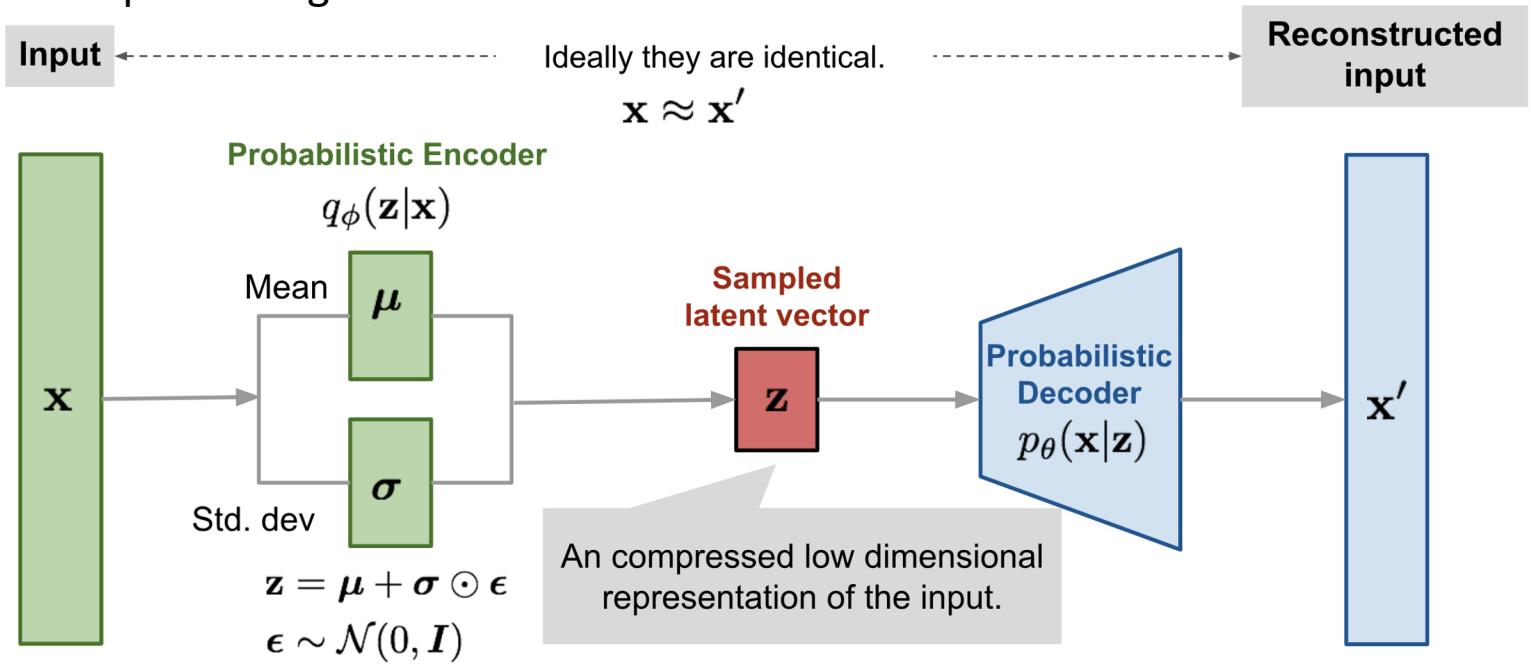
The basic autoencoder contains gaps in the latent space.

Latent space not well separated.

Source: https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Variational Autoencoders (VAE)

- Encoder is learning an approximation of the posterior distribution.
- Latent space is regularized to a standard normal distribution.

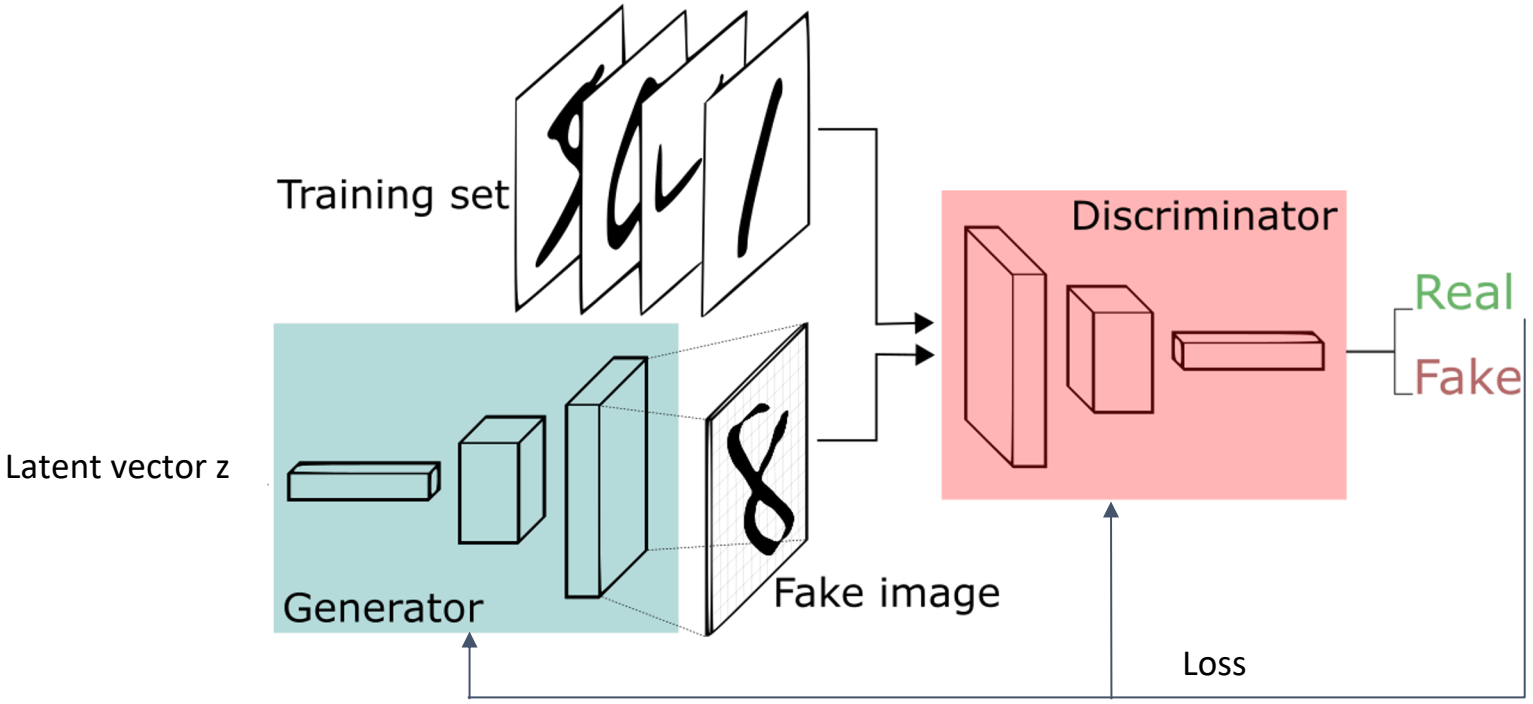


Means of MNIST digits

Source: https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Generative Adversarial Network (GAN)

- Generator objective: Fool the discriminator network by generating more real images.
- Discriminator objective: Become better in discriminating real and fake images



StyleGAN

Figure 2. Uncurated set of images produced by our style-based generator (config F) with the FFHQ dataset. Here we used a variation of the truncation trick [42, 5, 34] with $\psi = 0.7$ for resolutions $4^2 - 32^2$. Please see the accompanying video for more results.

Figure 11. Uncurated set of images produced by our style-based generator (config F) with the LSUN CAR dataset at 512 × 384. FID computed for 50K images was 3.27.

https://thispersondoesnotexist.com/

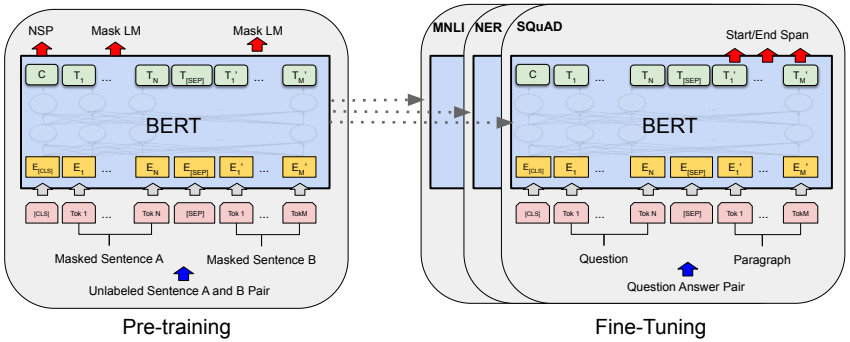
https://thiscatdoesnotexist.com/

Source: A Style-Based Generator Architecture for Generative Adversarial Networks, Tero et al. NVIDIA

Transformers

BERT (Google) – Bidirectional Encoder Representations for Transformers

- 340 million parameters (Large)
- Fine tune to specific task with additional output layer



GPT-3 (OpenAI) – Generative Pre-trained Transformer

- 175 billion parameters
- Can perform *specific* tasks without any special tuning by providing a few examples (less than 10):
- Translation
- Programmer
- Author

Source: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Source: Language Models are Few-Shot Learners

Transformers

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

The three settings we explore for in-context learning

Zero-shot

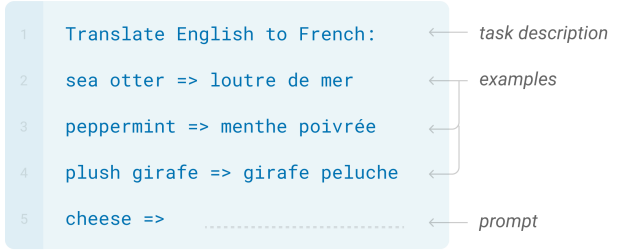
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

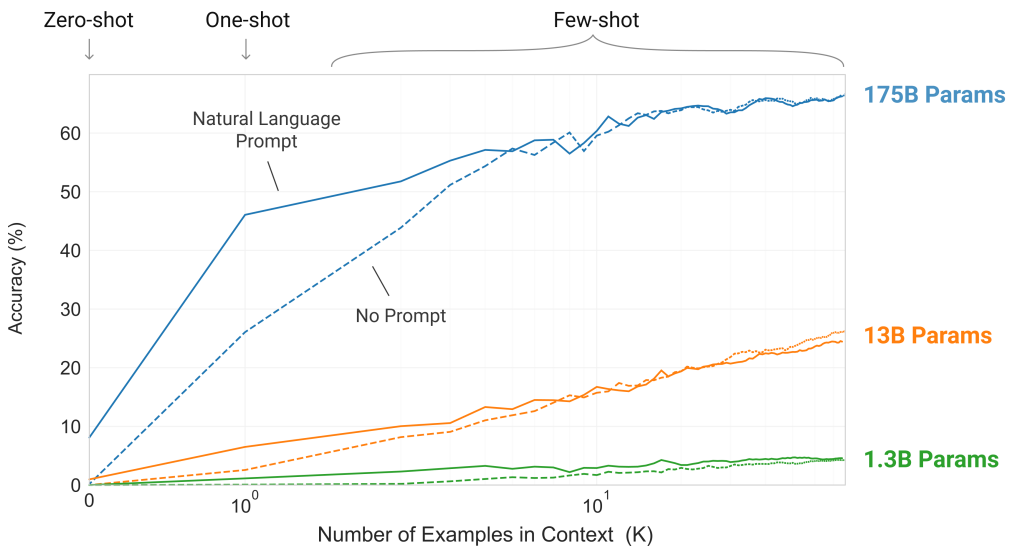
Few-shot

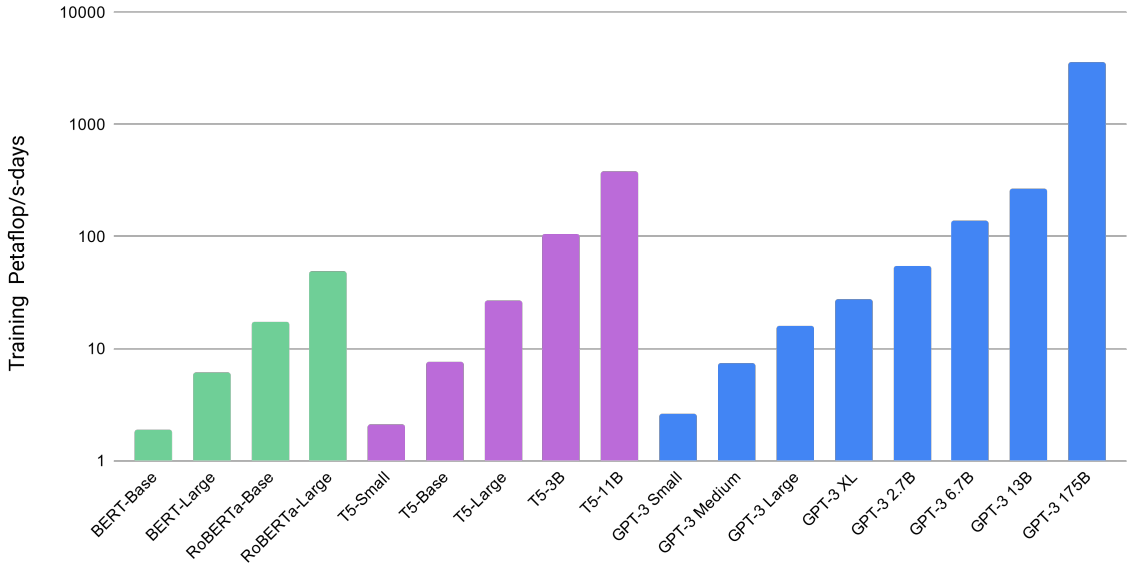
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Source: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Source: Language Models are Few-Shot Learners

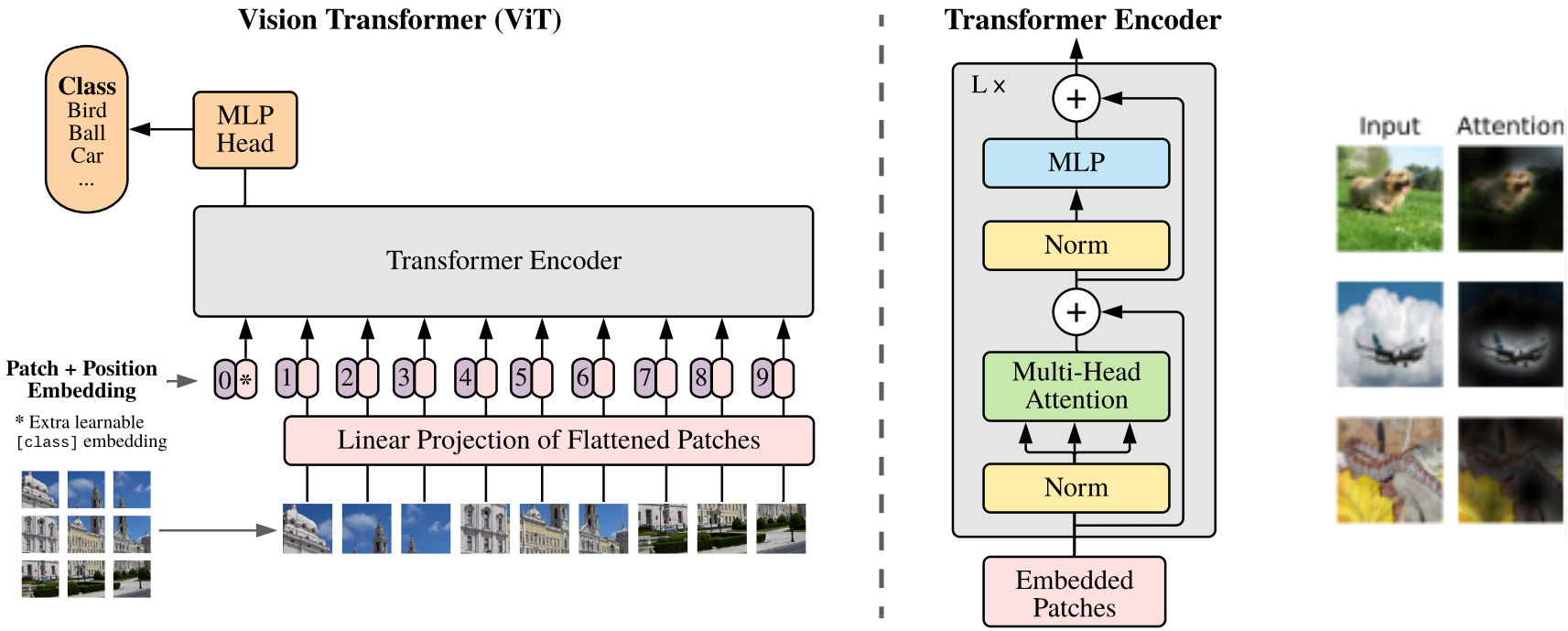
Transformers





Source: Language Models are Few-Shot Learners

Transformers for image classification



Source: An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, (2020 under review)

Deep learning outlook

- Deep learning is everywhere and will spread to even more areas in the years to come.
- Even though DL is popular, one should always analyze the problem at hand and pick the best tool.
- Still many open problems within DL:
  - Architecture understanding.
  - Reasoning capabilities, e.g. from context.
  - Robustness against adversarial attacks.
  - Fully unsupervised learning systems to avoid tedious labeling process.

Deep learning energy

Training artificial intelligence is an energy intensive process. New estimates suggest that the carbon footprint of training a single AI is as much as 284 tonnes of carbon dioxide equivalent – **five times the lifetime emissions of an average car.**

THE DAILY NEWSLETTER
Sign up to our daily email newsletter

NewScientist

News Technology Space Physics Health Environment Mind Video | Tours Events Jobs

Creating an AI can be five times worse for the planet than a car

TECHNOLOGY 6 June 2019

By Donna Lu

Server farms are used to train AIs
Tommy Lee Walker/Getty

Source: https://www.newscientist.com/article/2205779-creating-an-ai-can-be-five-times-worse-for-the-planet-than-a-car/

Deep learning detection robustness

Spot a pedestrian walking walking in front of a car coming with 20 MPH.

Only 40% of adult collisions in optimal conditions were avoided.

At night, the systems didn't even ping the driver to reduce speed.

Source:https://www.extremetech.com/extreme/299715-aaa-car-testing-shows-pedestrian-detection-tech-is-far-from-effective

Deep learning prediction

- Co-author of:
  - Learning representations by back-propagating errors (1986)
  - ImageNet Classification with Deep Convolutional Neural Networks (2012)
- Awarded the Turing Award together with Yann LeCun and Yoshua Bengio in 2019

- "I do believe DL is going to be able to do averything"
  - BUT: we need more breakthroughs like e.g. Transformers
  - We need scale (data + models)
    - Human brain: ~100 trillion parameters
    - GPT-3: 175 billion parameters (0.1% of the brain)

NOAH BERGER / AP

Artificial intelligence / Machine learning

AI pioneer Geoff Hinton: "Deep learning is going to be able to do everything"

Thirty years ago, Hinton's belief in neural networks was contrarian. Now it's hard to find anyone who disagrees, he says.

by **Karen Hao** November 3, 2020

Source: https://www.technologyreview.com/2020/11/03/1011616/ai-godfather-geoffrey-hinton-deep-learning-will-do-everything/

Credits

Books:

- https://www.deeplearningbook.org/
- http://neuralnetworksanddeeplearning.com/

Online Course from MIT:

- http://introtodeeplearning.com/

Online course from Stanford University:

- https://www.coursera.org/specializations/deep-learning?

Other

- cs231n.github.io
- appliedgo.net
- brohrer.github.io
- learnopencv.com