IPTC Seminar
Introduction to Deep Learning and Keras

July 15, 16 and 17, 2019
ETSIT UPM, Madrid

Introduction to Deep Learning & Keras
Day 3

iptc

Information Processing and Telecommunications Center

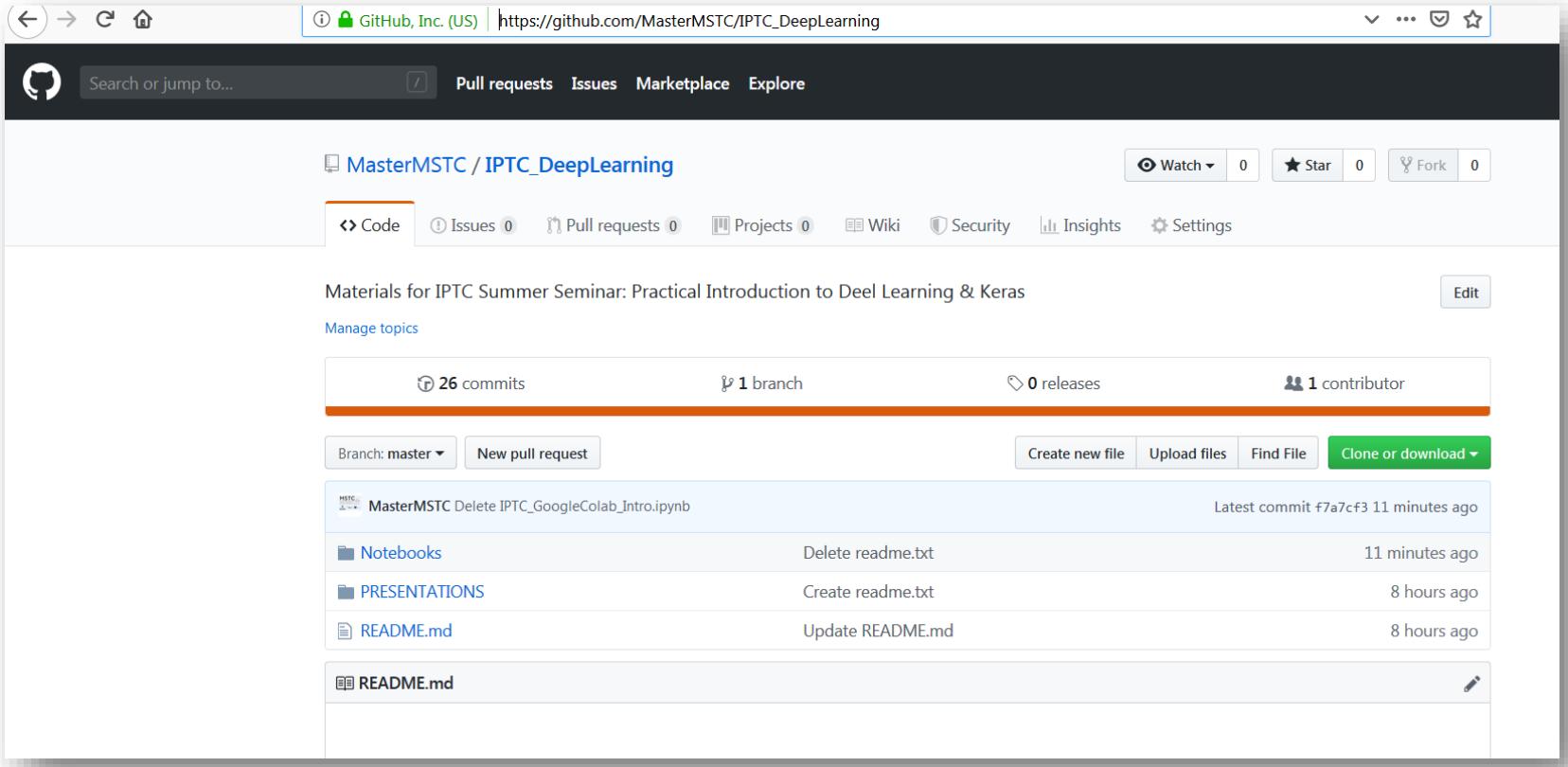
Prof. Luis A. Hernández Gómez
luisalfonso.hernandez@upm.es

E.T.S. Ingenieros de Telecomunicación
Universidad Politécnica de Madrid

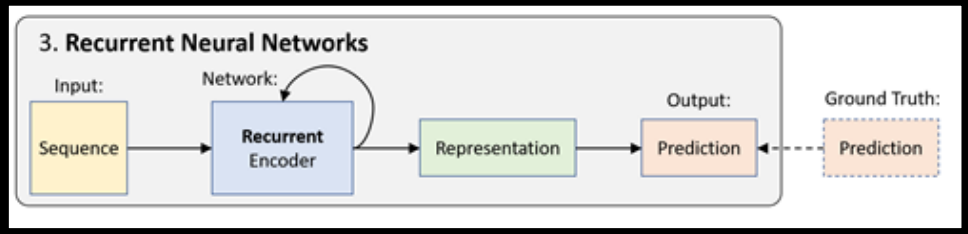

POLITÉCNICA
iptc
Information Processing and Telecommunications Center
Day 1 (July 15) : Intro to Deep Learning & Keras.
Feed Forward models
Day 2 (July 16) : Backpropagation. Convolutional Networks.
Transfer Learning. Data Augmentation.
Day 3 (July 17) : Recurrent Networks. Advanced Architectures.
GANs. Applications. Final discussion on AI
IPTC Seminar
Introduction to Deep Learning and Keras
July 15, 16 and 17, 2019
ETSIT UPM, Madrid

IPTC Summer Seminar MATERIALS

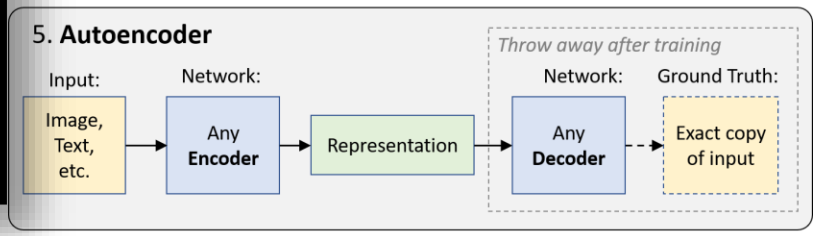
Please go to our MSTC GitHub:
https://github.com/MasterMSTC/IPTC_DeepLearning



Deep learning basics



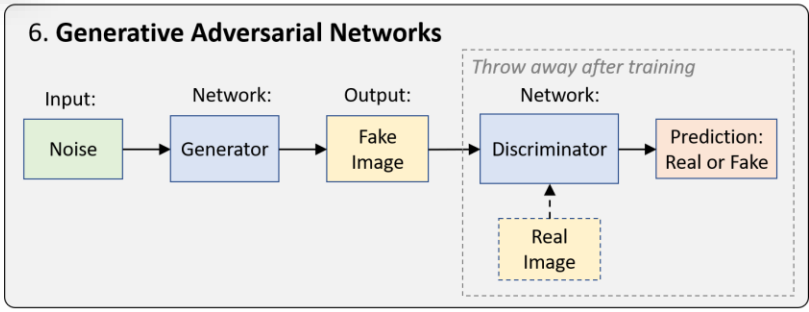
Unsupervised Learning

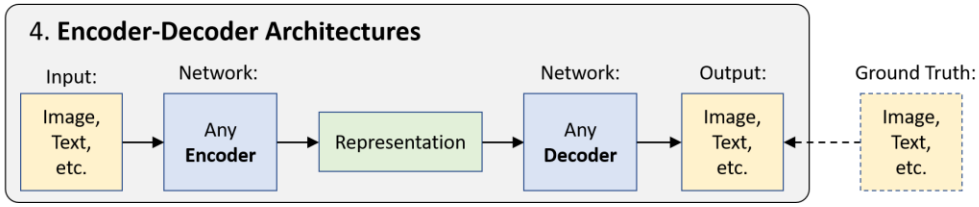


.... also

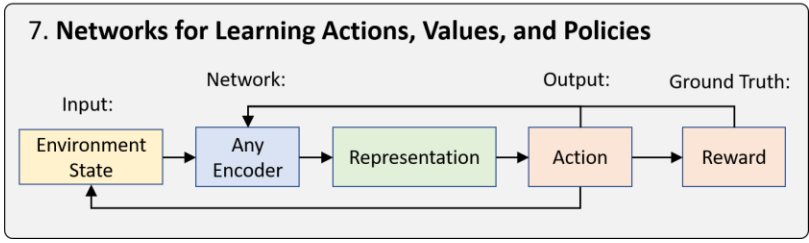
Generative Models

... (Reinforcement Learning)





Reinforcement Learning

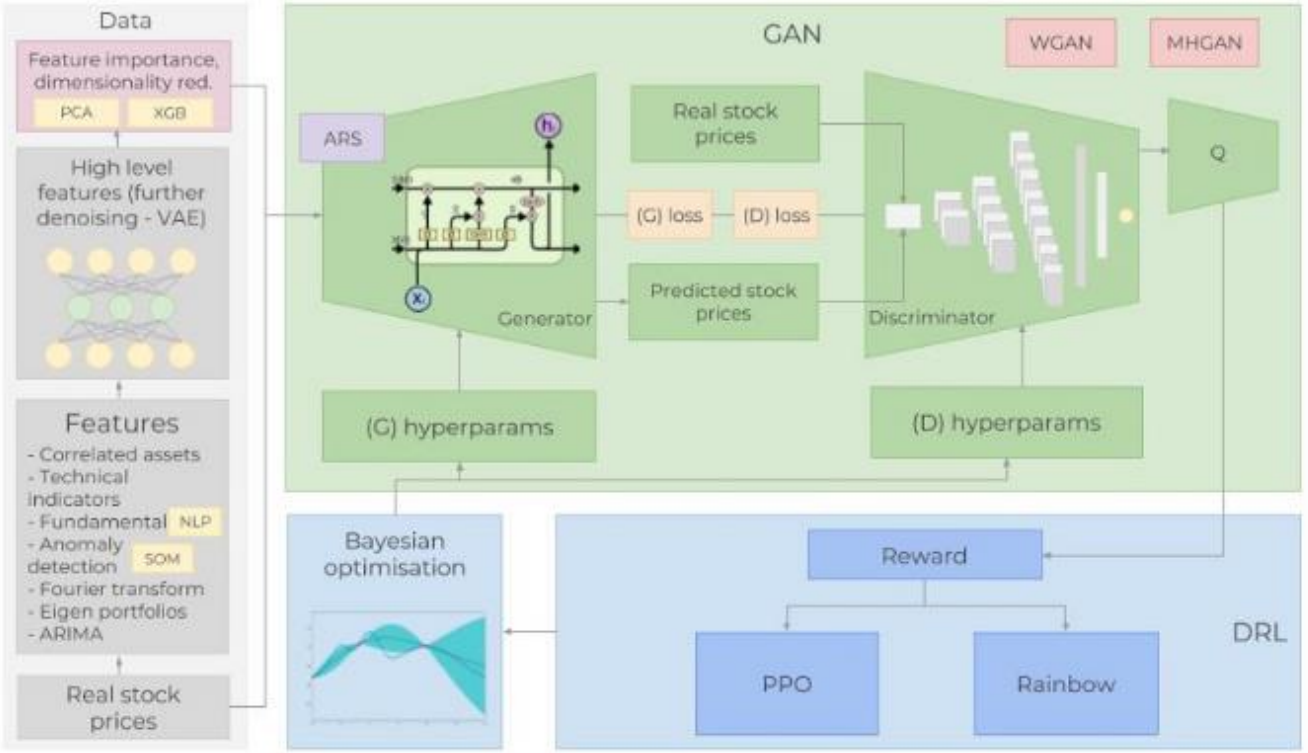


https://github.com/lexfridman/mit-deep-learning/blob/master/tutorial_deep_learning_basics/deep_learning_basics.ipynb

Using the latest advancements in deep learning to predict stock price movements

Boris B Follow

Jan 10 · 34 min read



Overview of the complete architecture.

https://towardsdatascience.com/aifortrading-2edd6fac689d

2.8. Extracting high-level features with Stacked Autoencoders

Before we proceed to the autoencoders, we'll explore an alternative activation function.

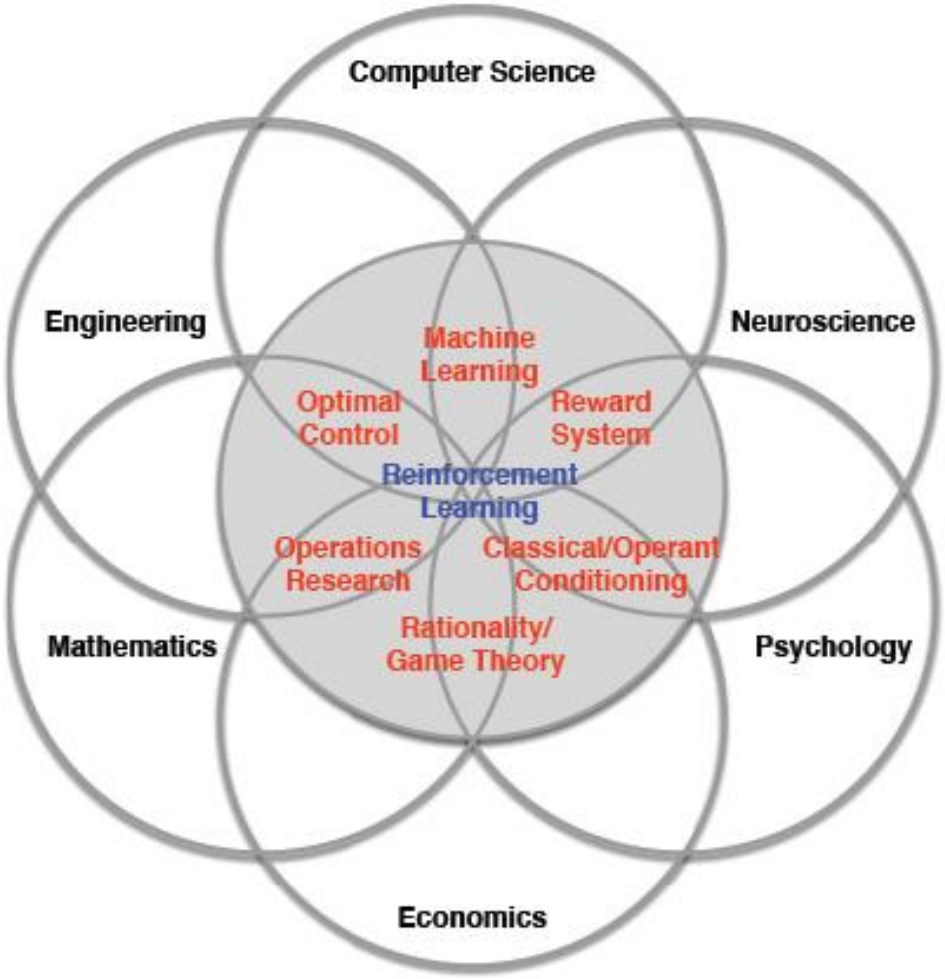
2.8.1. Activation function — GELU (Gaussian Error)

GELU — Gaussian Error Linear Unites was recently proposed — link. In the paper the authors show several instances in which neural networks using GELU outperform networks using ReLU as an activation. `gelu` is also used in **BERT**, the NLP approach we used for news sentiment analysis.

We will use GELU for the autoencoders.

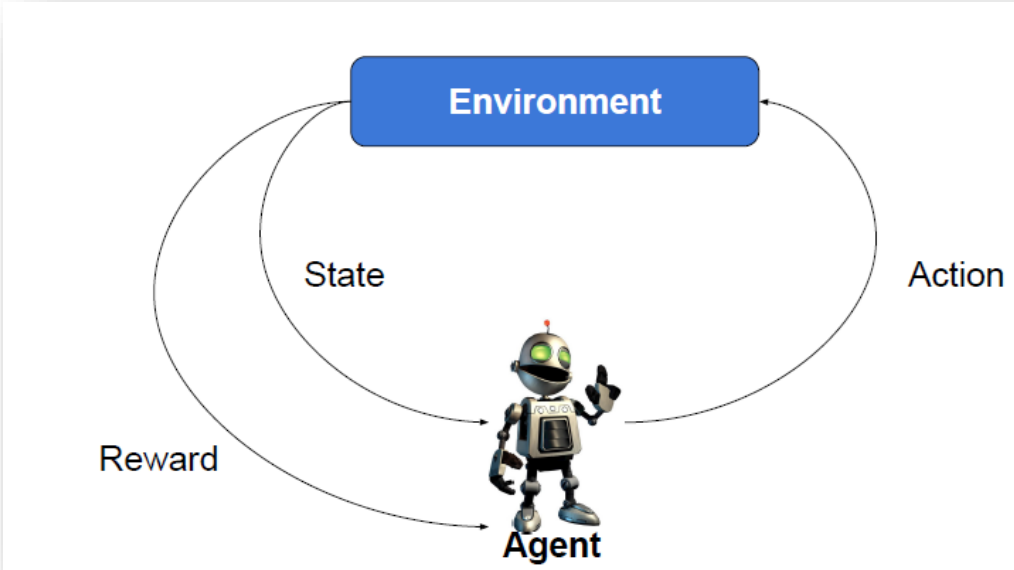
https://towardsdatascience.com/aifortrading-2edd6fac689d

Many Faces of Reinforcement Learning



Tutorial: Deep Reinforcement Learning
David Silver, Google DeepMind

What's Reinforcement Learning?

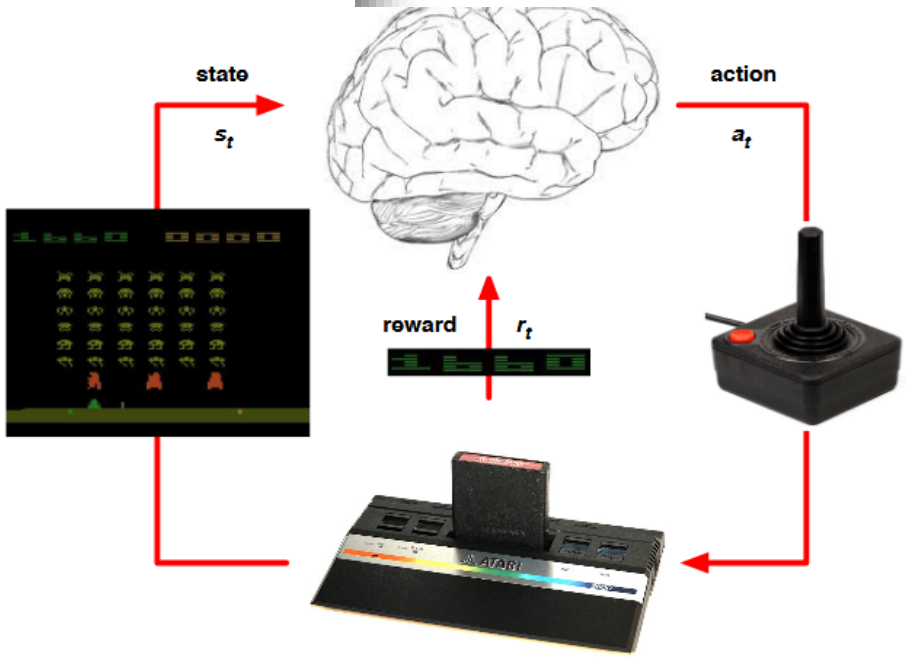


Gym

OpenAI

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

- Agent interacts with an environment and learns by maximizing a reward.
- No labels or any other supervision is given.



Major Components of an RL Agent

- An RL agent may include one or more of these components:
  - Policy: agent's behaviour function
  - Value function: how good is each state and/or action
  - Model: agent's representation of the environment

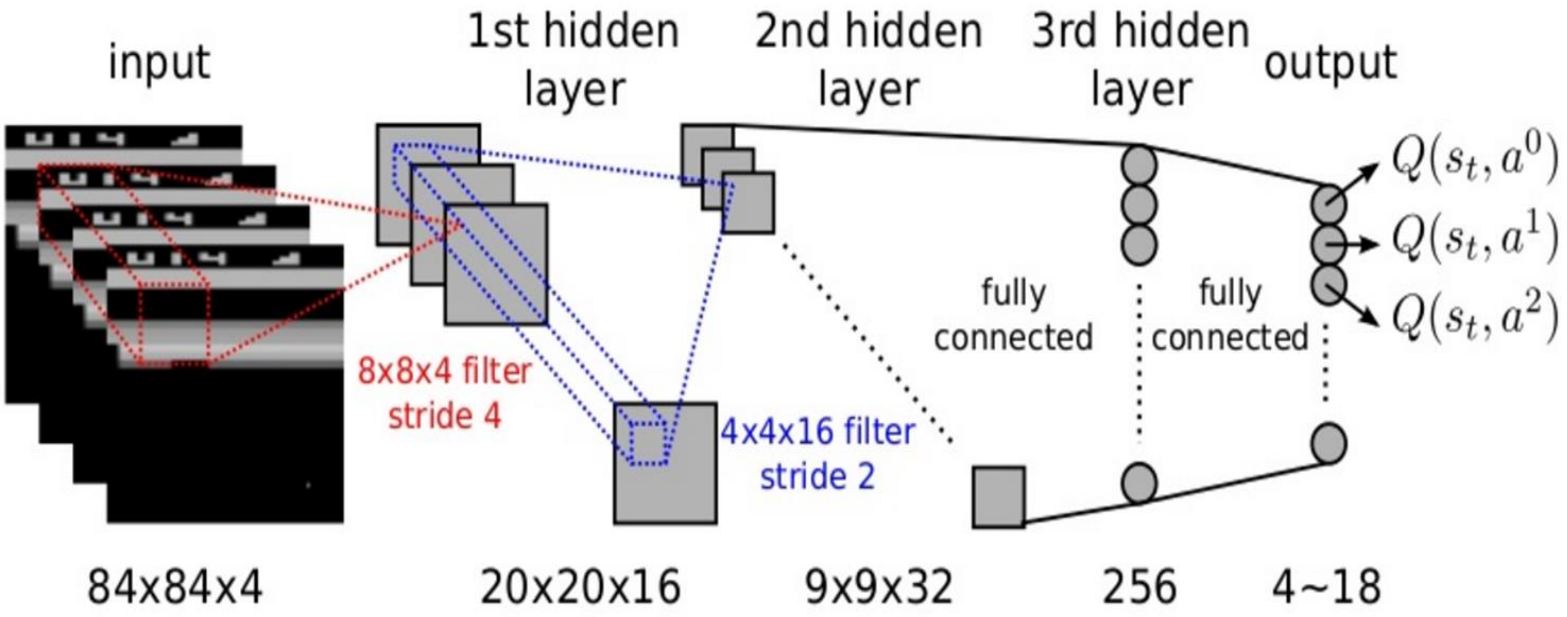
- Policy π is a behavior function selecting actions given states

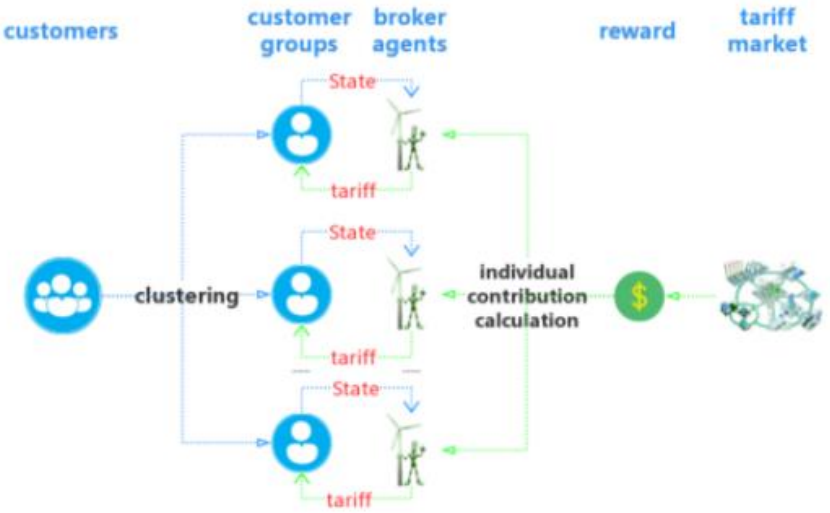
$$\boldsymbol{a = \pi(s)}$$

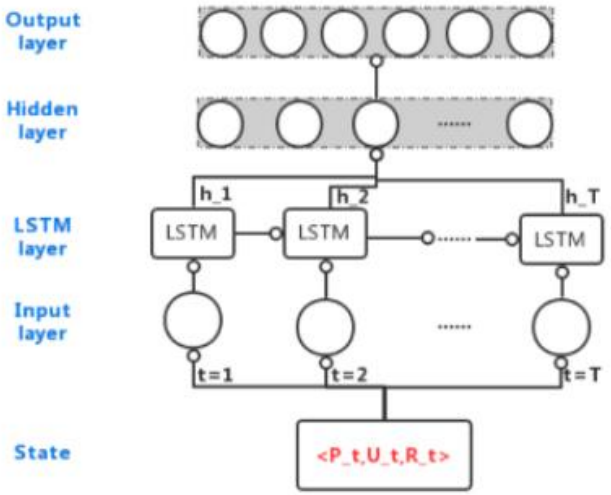
- Value Function $Q^{\pi}(s, a)$ is expected total reward from state $\boldsymbol{s}$ and action $\boldsymbol{a}$ under policy $\boldsymbol{\pi}$

$$\boldsymbol{Q^{\pi}(s,a) = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s, a]}$$

Example: Deep Q Learning (in Atari)



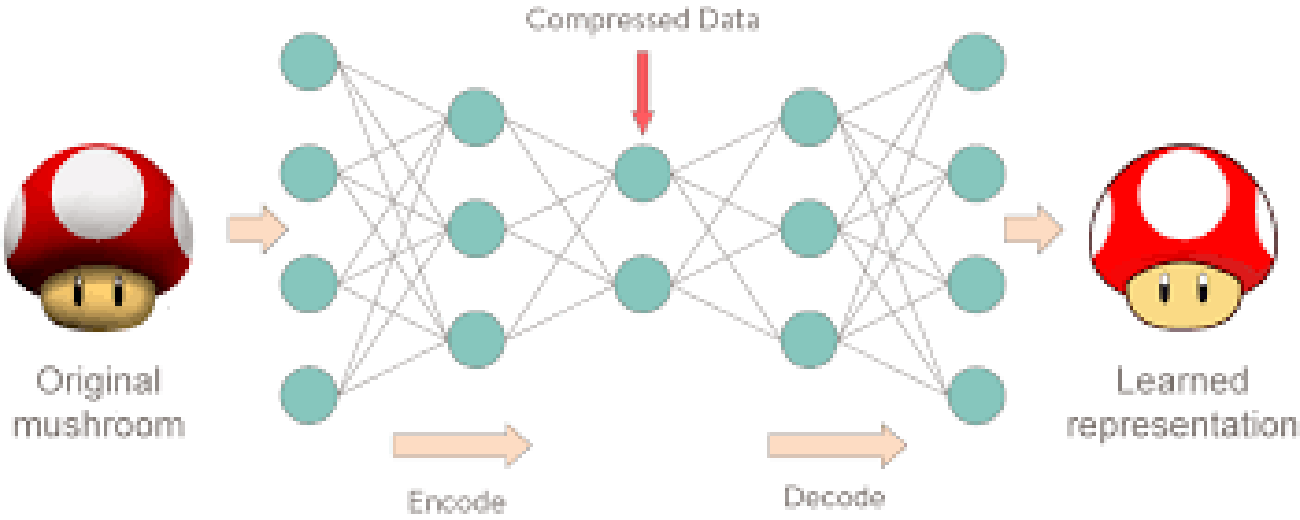




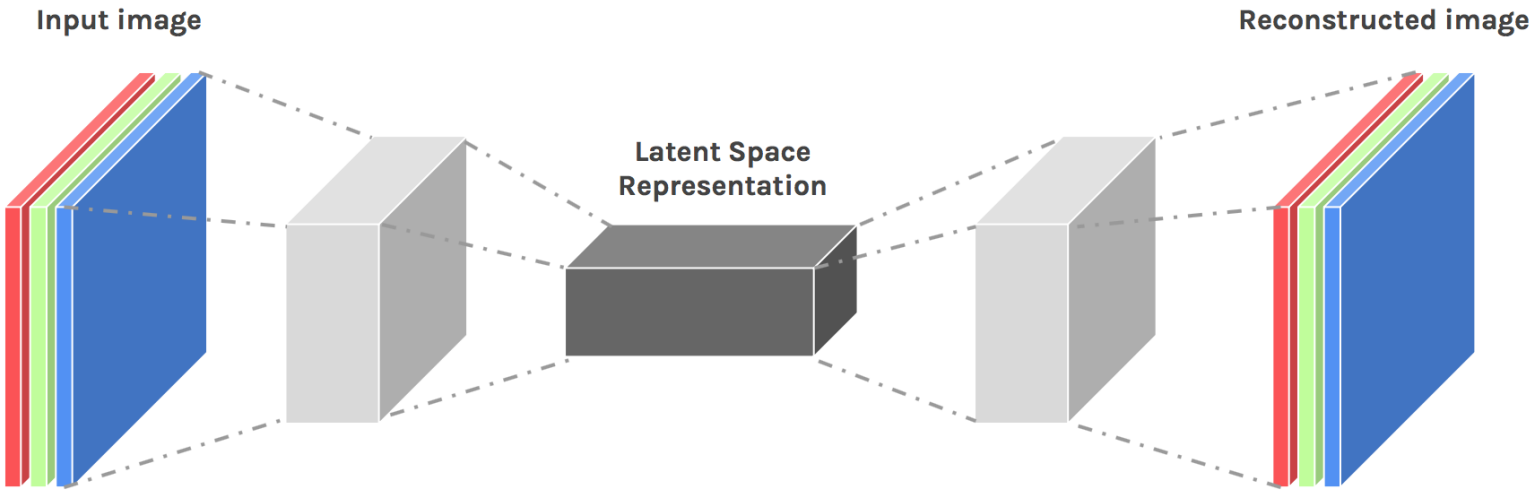
Deep Reinforcement Learning in Retail Pricing Strategy, Source : https://www.ijcai.org/proceedings/2018/0079.pdf

https://towardsdatascience.com/deep-learning-vs-deep-reinforcement-learning-algorithms-in-retail-industry-ii-9c17c83ecf2f

Autoencoders



You can use different Architectures,
for example Convolutional Autoencoder



To capture the distribution of the data

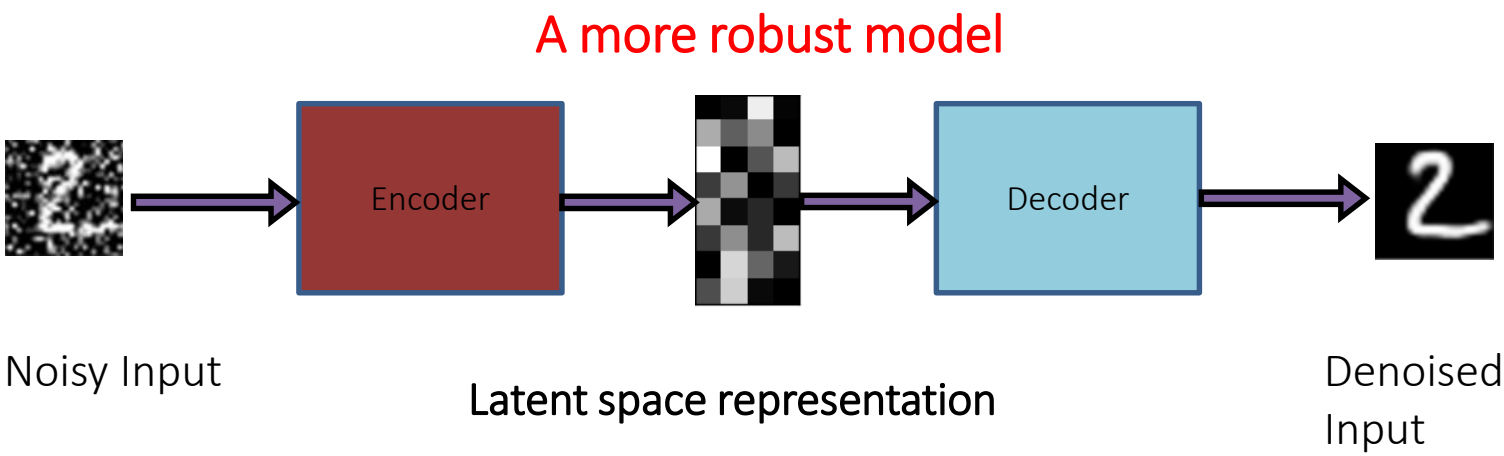
- Bottleneck
- Variational Autoencoders
  https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

iptc
Information Processing and Telecommunications Center

Denoising Autoencoders

Intuition:

- We still aim to encode the input and to NOT mimic the identity function.
- We try to undo the effect of *corruption* process stochastically applied to the input.

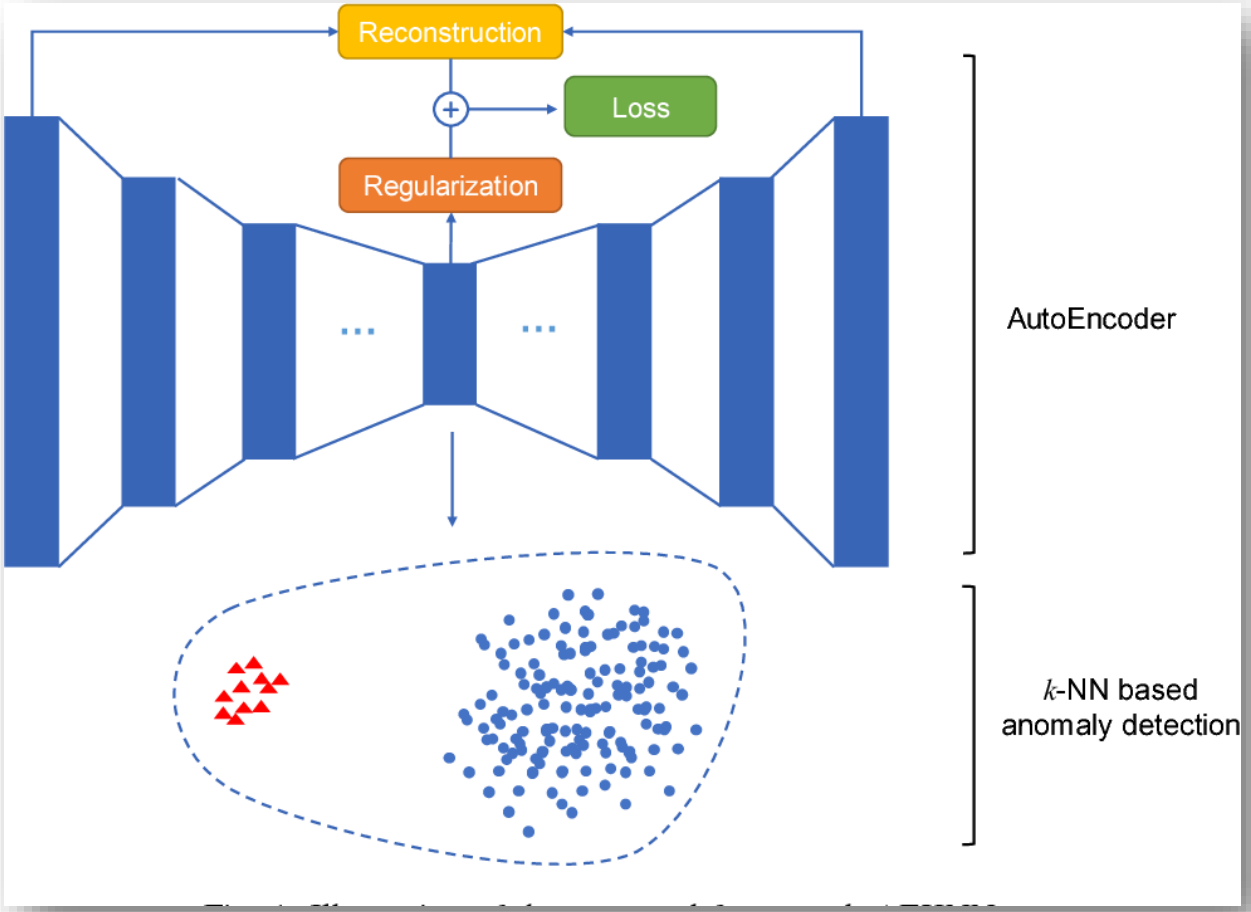


An Anomaly Detection Framework Based on Autoencoder and Nearest Neighbor

Jia Guo, Guannan Liu, +1 author Junjie Gavin Wu ·

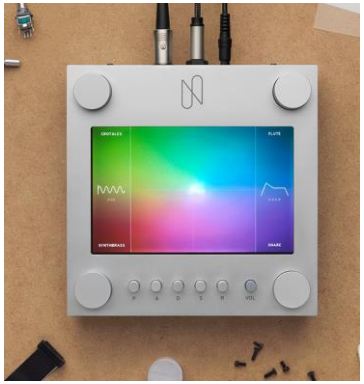
DOI: 10.1109/icsssm.2018.8464983

Published in 15th International Conference on Service Systems... 2018 ·



https://www.dataversity.net/fraud-detection-using-a-neural-autoencoder/

Generations of effects, synthesizers, etc.



Traditionally, a computer turns sound waves directly into numbers. To mix a bass and flute, you would add the numbers at each point along each sound wave. And that would sound like this – a bass playing with a flute:

NSynth also turns sound into numbers. But these numbers don't represent the sound wave itself. They represent a more abstract description of the sound. When you mix a bass and flute using these numbers, it sounds more like a new, hybrid bass-flute sound:

You also get interesting results playing non-instrument sounds through NSynth, like a cow's moo. Since NSynth is only trained on instruments, it tries to recreate the sound using only what it knows about how instruments behave:

https://experiments.withgoogle.com/ai/sound-maker/view/

Get Started | Blog | Topics | EBooks | Donate | FAQ | About | Contact

LSTM Model Architecture for Rare Event Time Series Forecasting

by **Jason Brownlee** on November 2, 2018 in **Deep Learning for Time Series**

Tweet | Share | Share

Time series forecasting with LSTMs directly has shown little success.

Welcome to Machine Learning Mastery!

Hi, I'm **Jason Brownlee**, PhD. I write tutorials to help developers (*like you*) get results with machine learning.

Read More

Uber Engineering

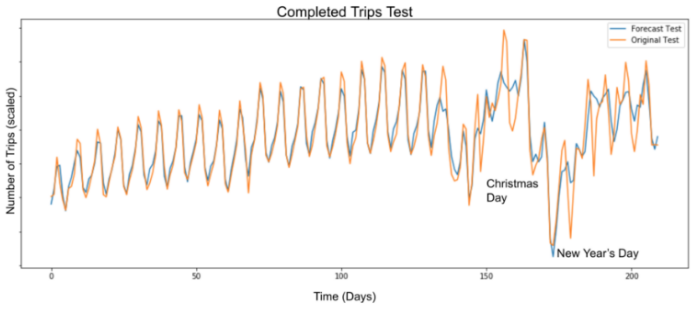
Blog | Research | Locations

Uber Data

Engineering Extreme Event Forecasting at Uber with Recurrent Neural Networks

Nikolay Laptev, Slawek Smyl, and Santhosh Shanmugam

June 9, 2017



Sign up for Uber Engineering updates:

Your email address

Subscribe

Popular Articles

Uber's Big Data Platform: 100+ Petabytes with Minute Latency

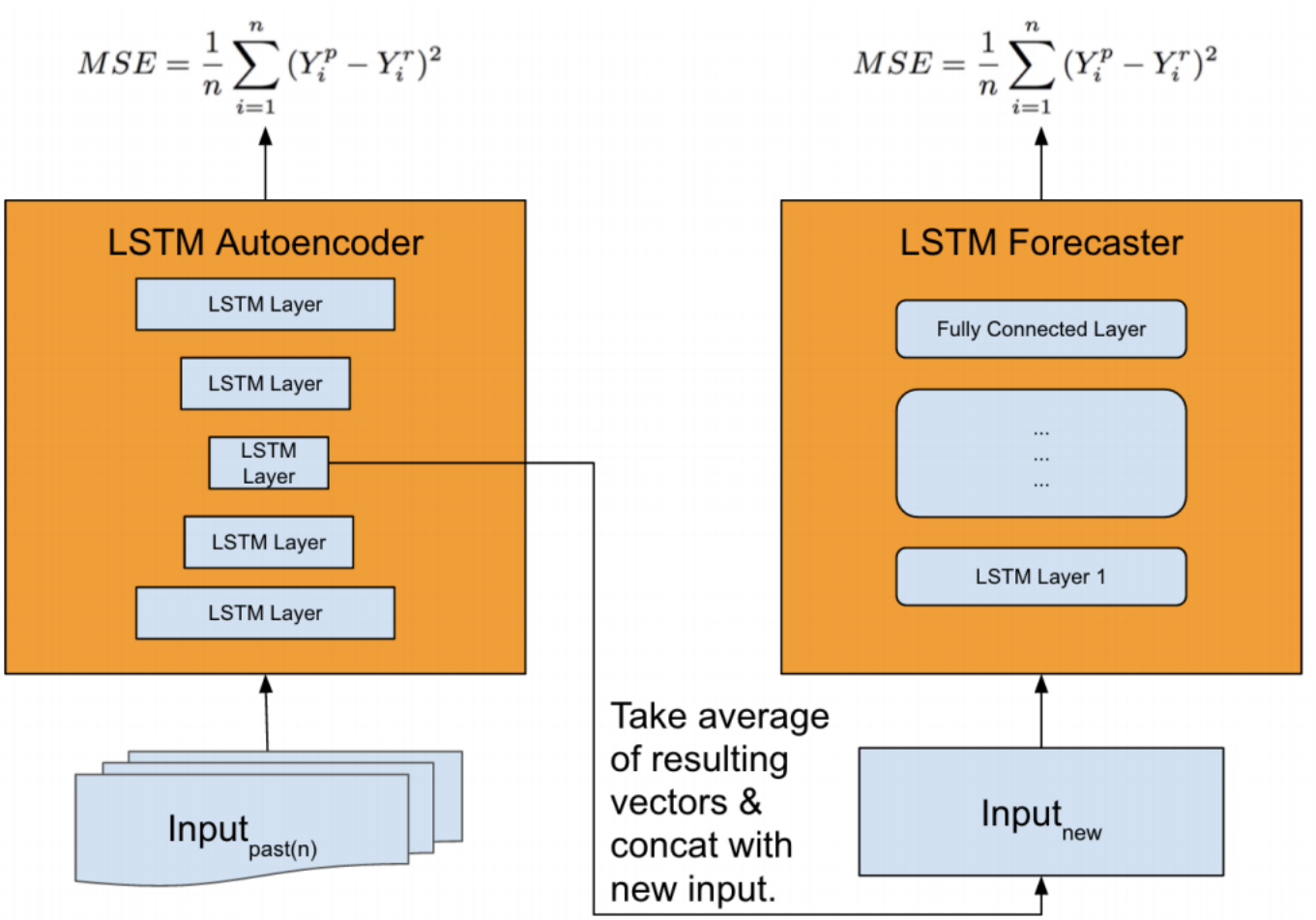
October 17, 2018

Meet Michelangelo: Uber's Machine Learning Platform

iptc Information Processing and Telecommunications Center

Overview of Feature Extraction Model and Forecast Model

Taken from “Time-series Extreme Event Forecasting with Neural Networks at Uber.”



Deep Reinforcement Learning

Some Refs

- Introduction to RL by David Silver
  http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf
  https://www.youtube.com/watch?v=2pWv7GOvuf0&list=PL7-jPKtc4r78-wCZcQn5IqyuWhBZ8fOxT

- Lex Fridman. MIT 6.S094: Deep Reinforcement Learning
  https://www.youtube.com/watch?v=MQ6pP65o7OM

- Harini Suresh.MIT 6.S191 Lecture 6: Deep Reinforcement Learning
  https://www.youtube.com/watch?v=xWe58WGWmlk&t=362s

- https://www.youtube.com/watch?v=lvoHnicueoE

Modelling sequences

- Applications areas are broad:
  - **Signals**: video, speech, music, sensors,....
  - **Time series**: financial series, log messages,..
  - **Sequence of characters**: Natural Language Processing, MIDI music...
  - **Sequences of observations**, actions, rewards: Reinforcement Learning (robots, bots, autonomous vehicles, dialog systems..)

Simple Examples: NLP & MUSIC

Text Prediction/Generation with Keras using LSTM: *Long Short Term Memory* networks

In this example we will work with the book: Alice's Adventures in Wonderland by Lewis Carroll.

We are going to learn the dependencies between characters and the conditional probabilities of characters in sequences so that we can in turn generate wholly new and original sequences of characters.

Adapted from:

Text Generation With LSTM Recurrent Neu

By Jason Brownlee

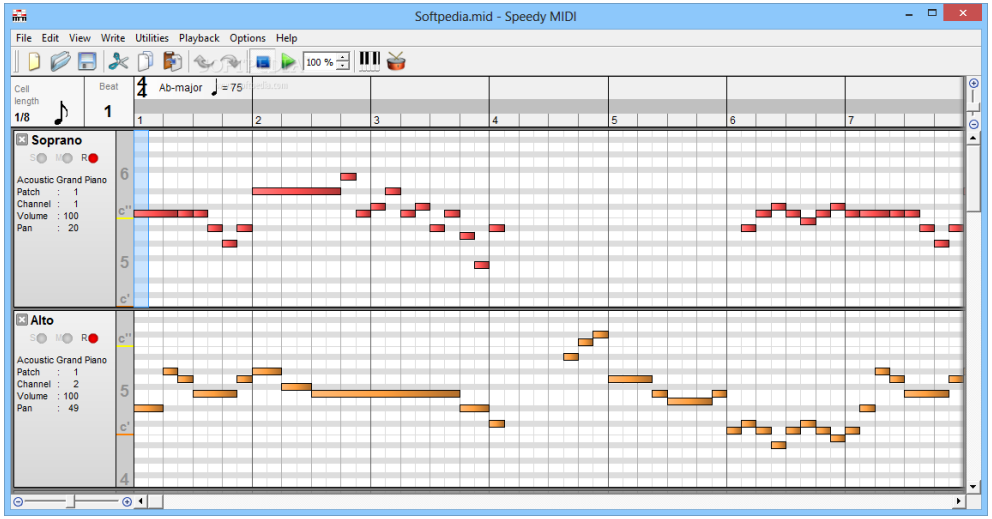
See example using the IMDB movie review sentiment prediction.

"IMDB dataset", a set of 50,000 highly-polarized reviews from the Internet Movie Database. They are split into 25,000 reviews for training and 25,000 reviews for testing, each set consisting in 50% negative and 50% positive reviews.

Just like the MNIST dataset, the IMDB dataset comes packaged with Keras. It has already been preprocessed: the reviews (sequences of words) have been turned into sequences of integers, where each integer stands for a specific word in a dictionary.

Keras

MIDI Music Generation



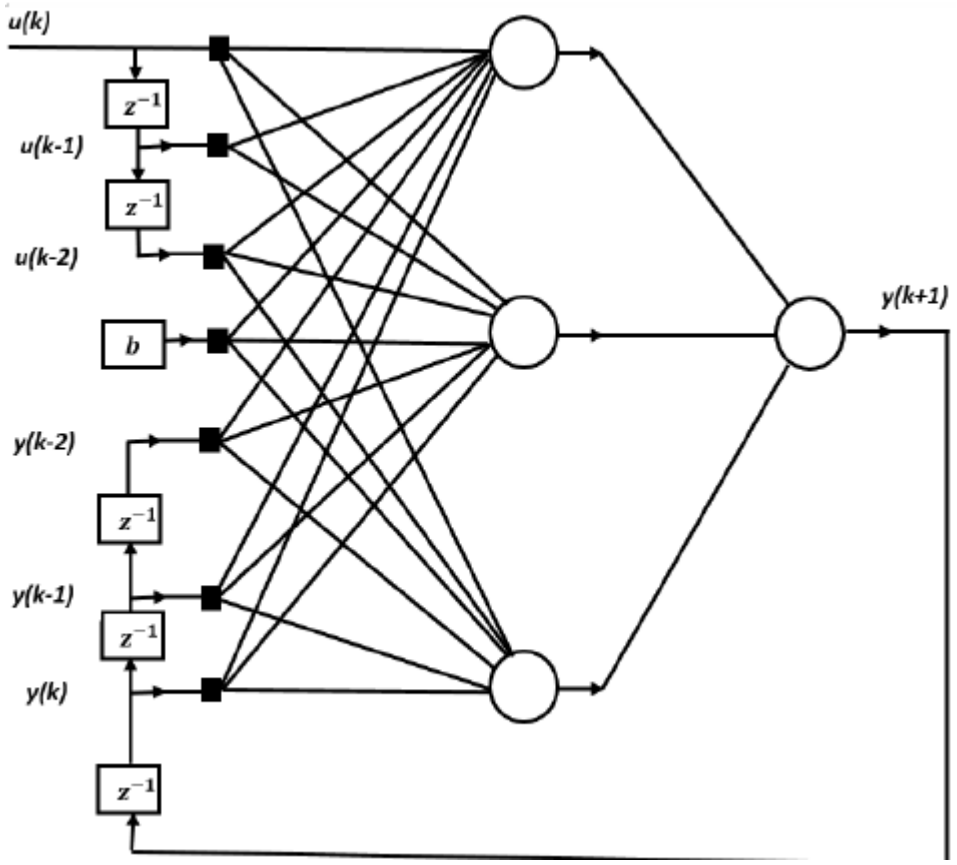
https://blog.floydhub.com/gpt2/

29 APRIL 2019 / DEEP LEARNING

How to Build OpenAI's GPT-2: "The AI That's Too Dangerous to Release"

Memoryless models for sequences

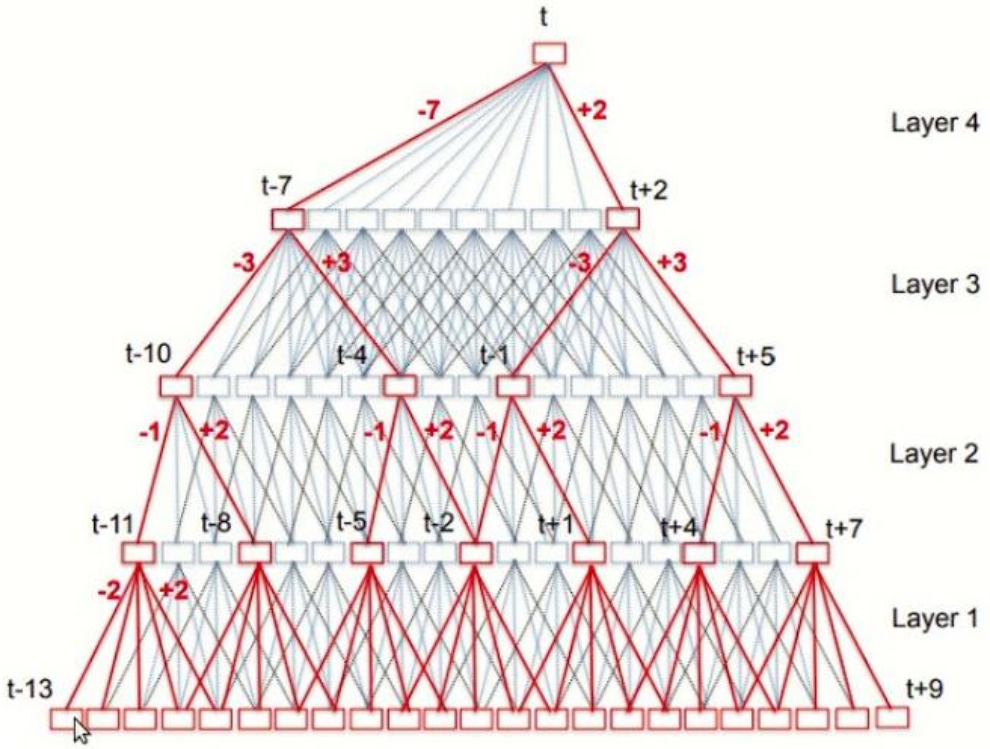
- **NARX** model: **N**onlinear **A**uto **R**egressive Model with e**x**ternal inputs.
- Another name: **TDNN** (time-delay neural network)



TDNN : Time Delay Neural Network

2.1 TDNN basics

1. Input patches consist of fixed number of feature vectors as input to TDNN
2. The TDNN processes them with fixed local temporal context
3. The context width increases as we go to upper layers

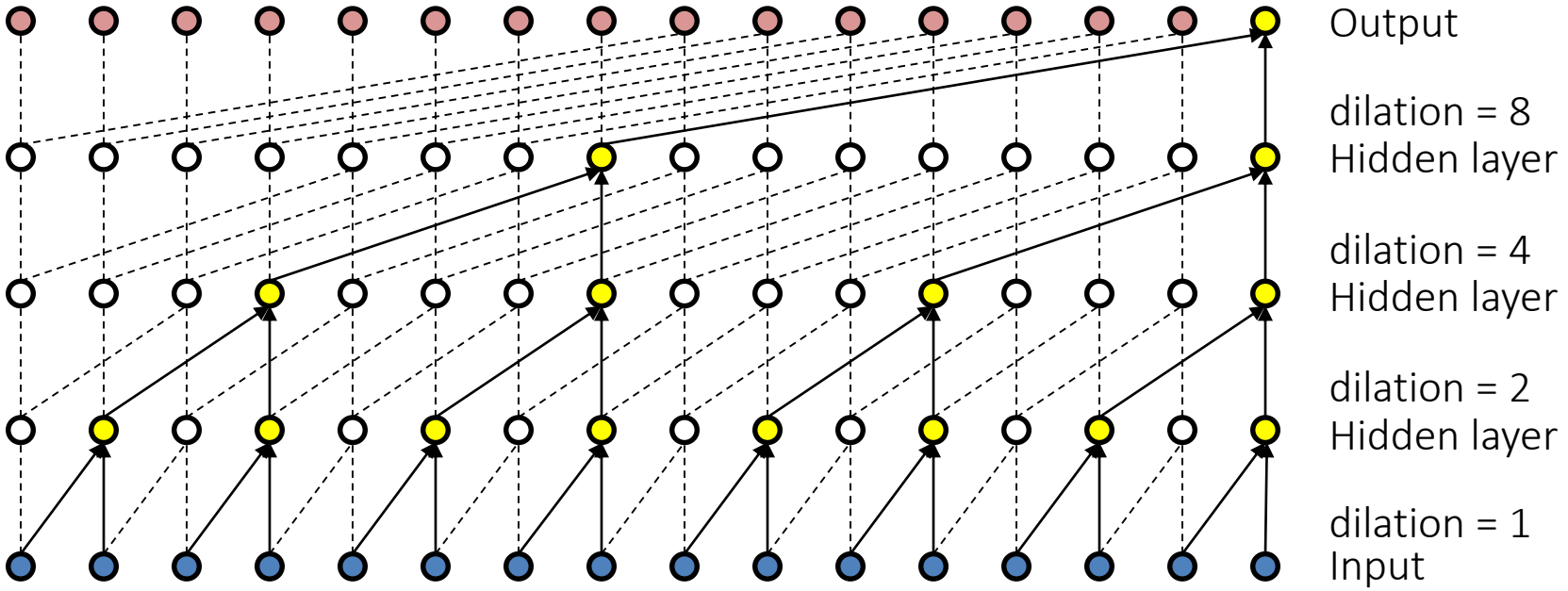


https://www.youtube.com/watch?v=hwSVs7muapQ

WaveNet: A Generative Model for Raw Audio

DeepMind

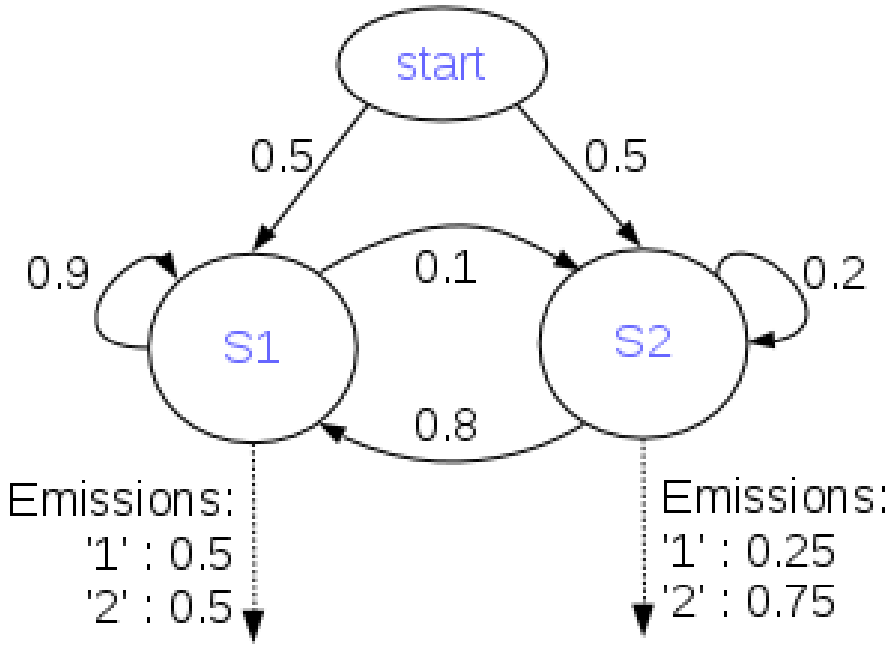
WaveNet models the conditional probability distribution $p(x_t|x_1, \ldots, x_{t-1})$ with a stack of dilated causal convolutions.



https://deepmind.com/blog/wavenet-generative-model-raw-audio/

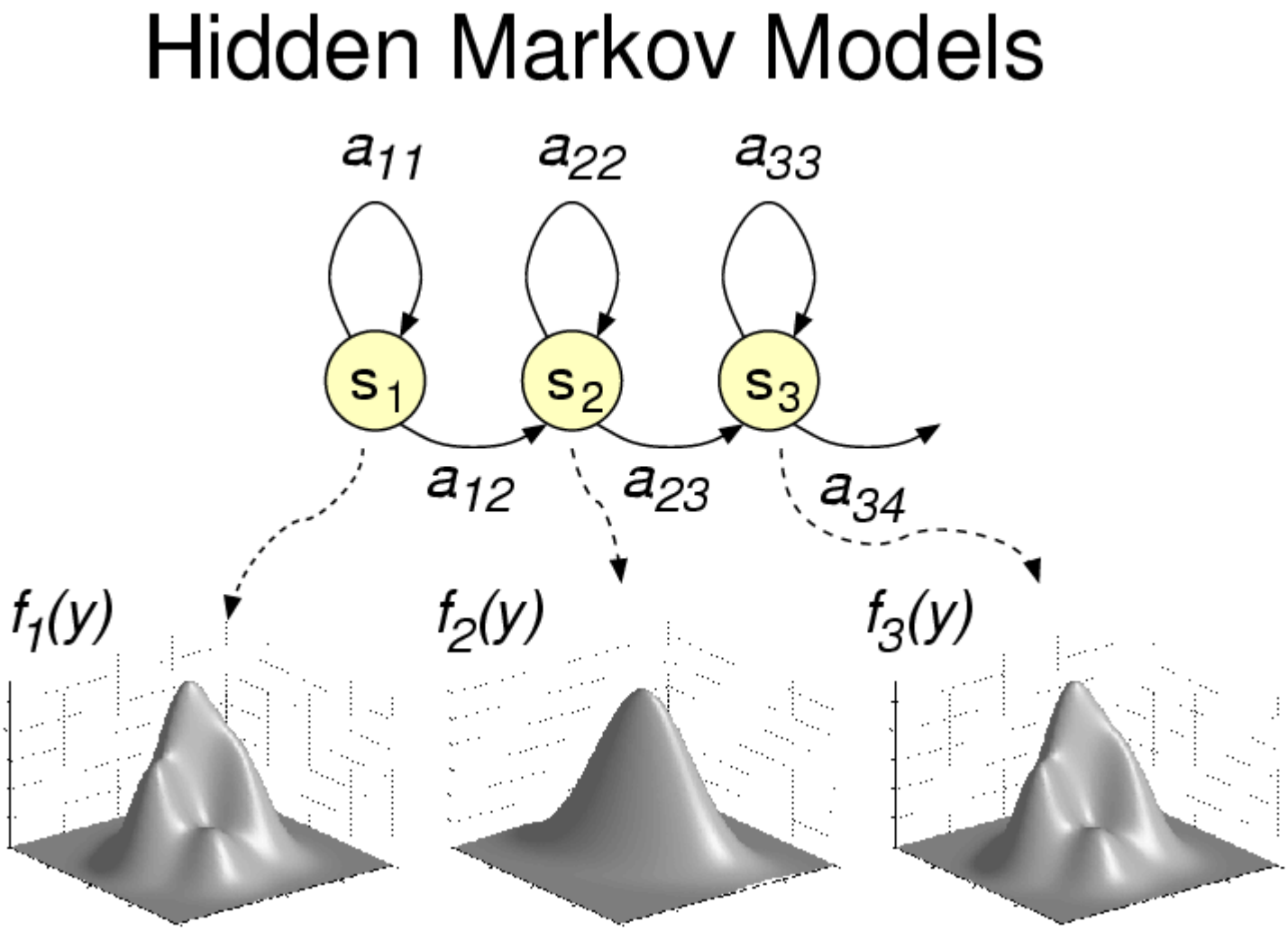
Dynamic Generative Models:
as for example Hidden Markov Models

- Hidden Markov Models have a discrete one-of-N hidden state.
- Transitions between states are stochastic and controlled by a transition matrix.
- The outputs produced by a state are stochastic.



From: Geoffrey Hinton
CSC2535 2013: Advanced Machine Learning

Traditional Automatic Speech Recognition : HMMs



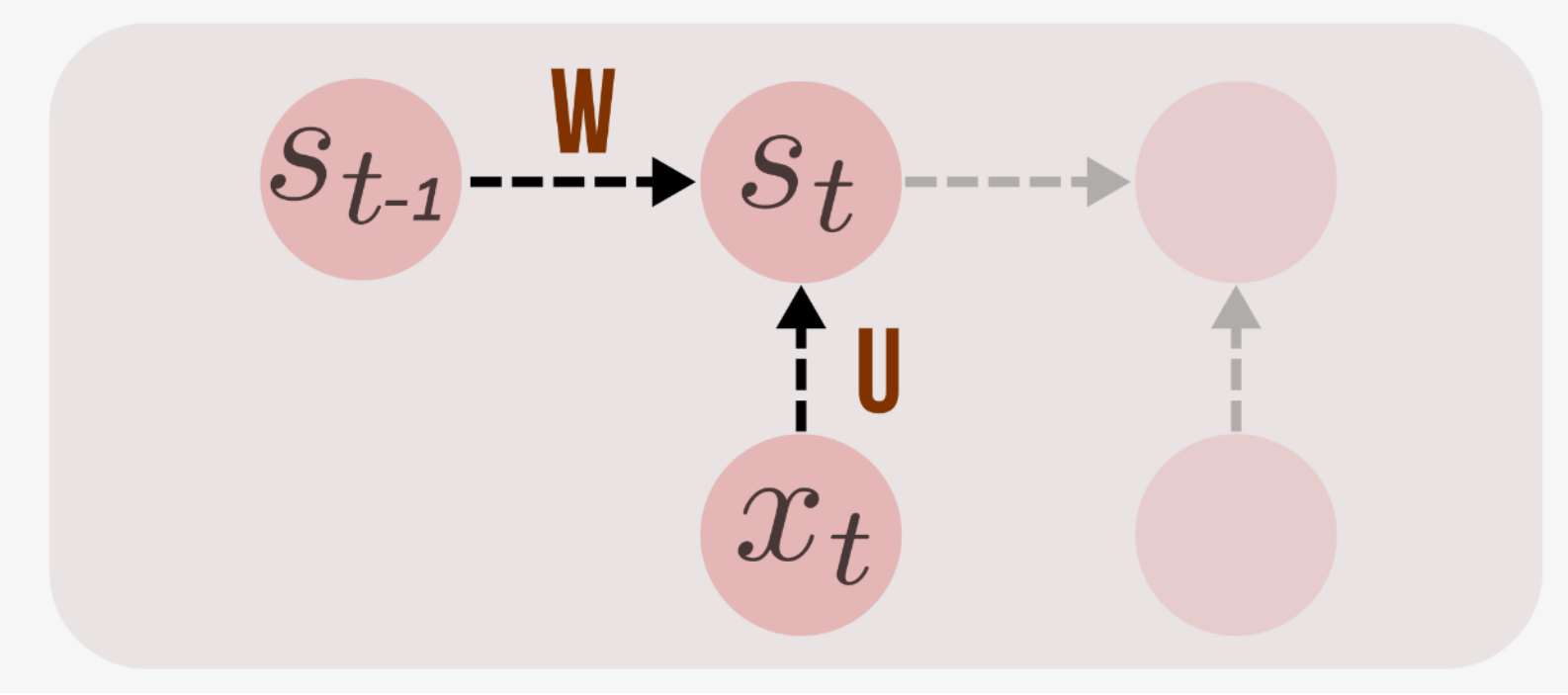
A fundamental limitation of HMMs

- Consider what happens when a hidden Markov model generates data.
  - At each time step it must select one of its hidden states.
  - So with N hidden states it can only remember log(N) bits about what it generated so far.

From: Geoffrey Hinton
CSC2535 2013: Advanced Machine Learning

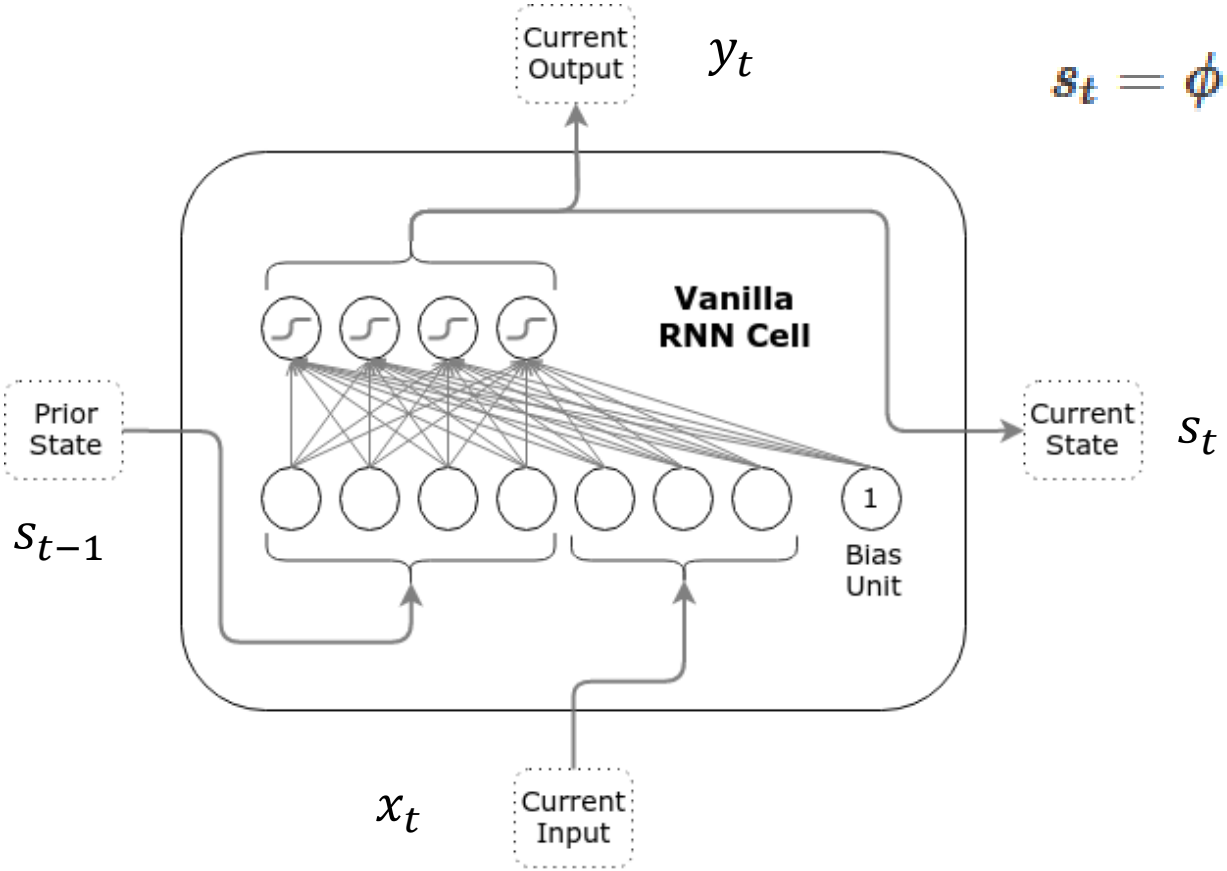
Recurrent Neural Networks (RNN)

RNNs combine the **input vector** with their **state vector** with a **fixed** (but learned) **function** to **produce a new state vector**.



$$s_t = tanh(Ux_t + Ws_{t-1}),$$

http://suriyadeepan.github.io/2017-02-13-unfolding-rnn-2/



$$s_t = \phi(W s_{t-1} + U x_t + b)$$

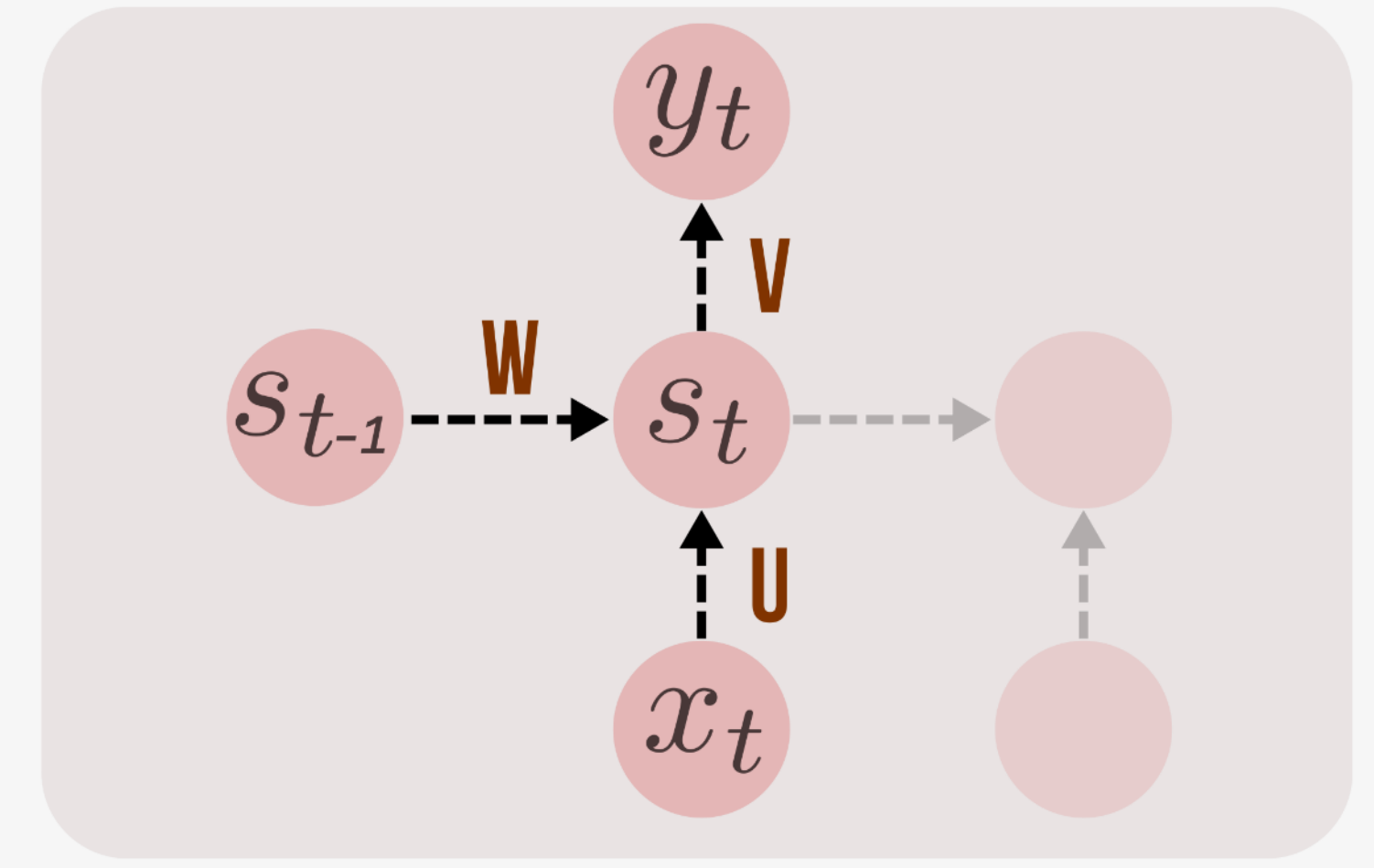
where:

- ϕ is the activation function (e.g., sigmoid, tanh, ReLU),
- $s_t \in \mathbb{R}^n$ is the current state (and current output),
- $s_{t-1} \in \mathbb{R}^n$ is the prior state,
- $x_t \in \mathbb{R}^m$ is the current input,
- $W \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^n$ are the weights and biases, and
- n and m are the state and input sizes.

OUTPUT y_t

logits = tf.matmul(state, V) + b

predictions = tf.nn.softmax(logits)



Recurrent Neural Networks (RNN) concepts

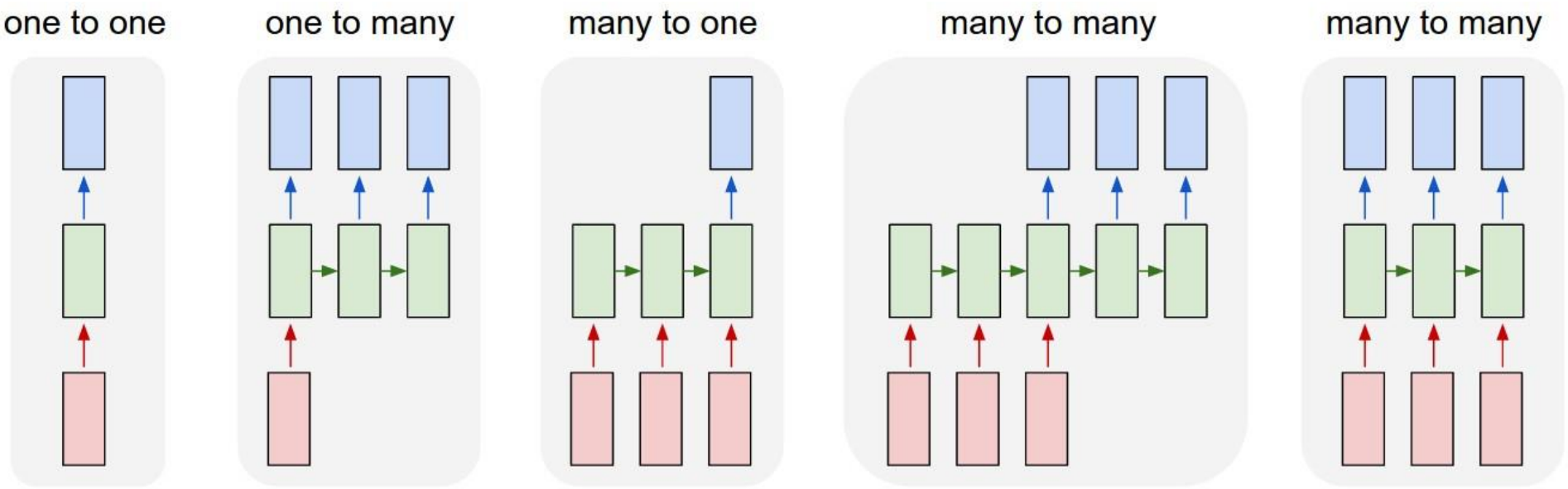


Image description using text

Emotion Recognition

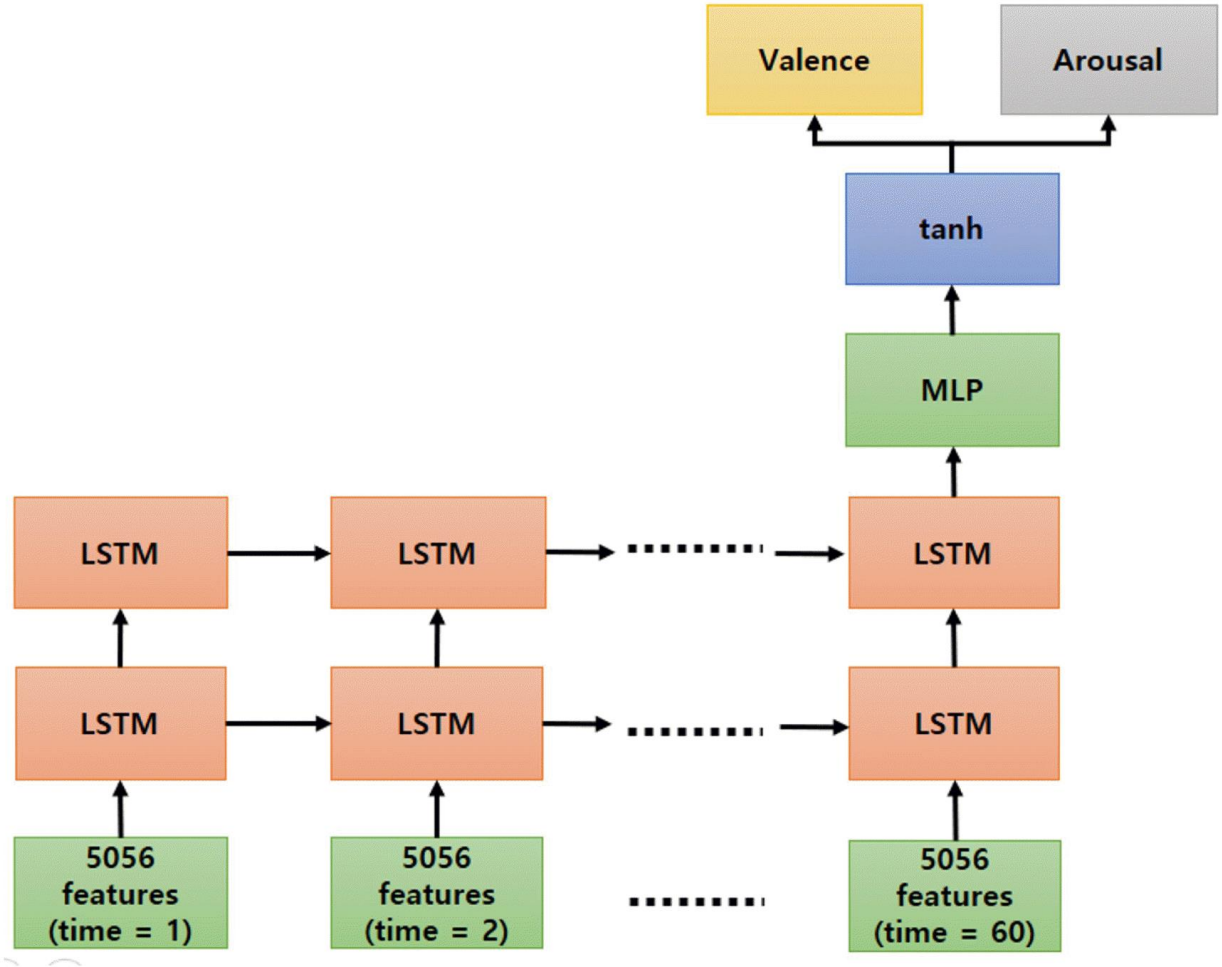
Language Model

Machine Translation

Video frame labelling

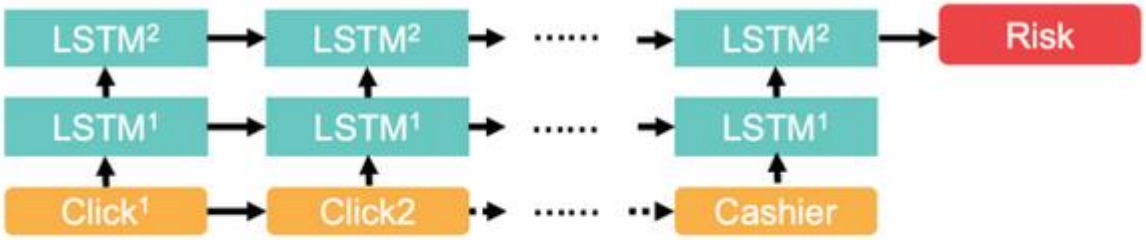
Phoneme Recognition

Emotion recognition



Deep-Learning Based Model for Emotional Evaluation of Video Clips
Byoungjun Kim, Joonwhoan Lee

Deep Learning (LSTM) to predict fraud/authentic user-order

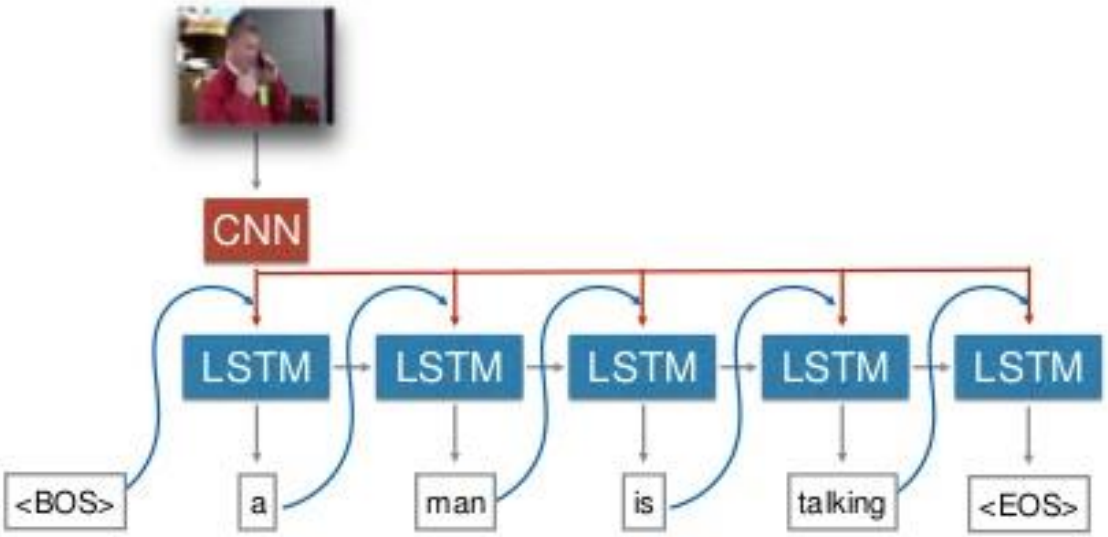


Clicks to detect Fraus , Source : http://ecmlpkdd2017.ijs.si/papers/paperID69.pdf

https://towardsdatascience.com/deep-learning-vs-deep-reinforcement-learning-algorithms-in-retail-industry-ii-9c17c83ecf2f

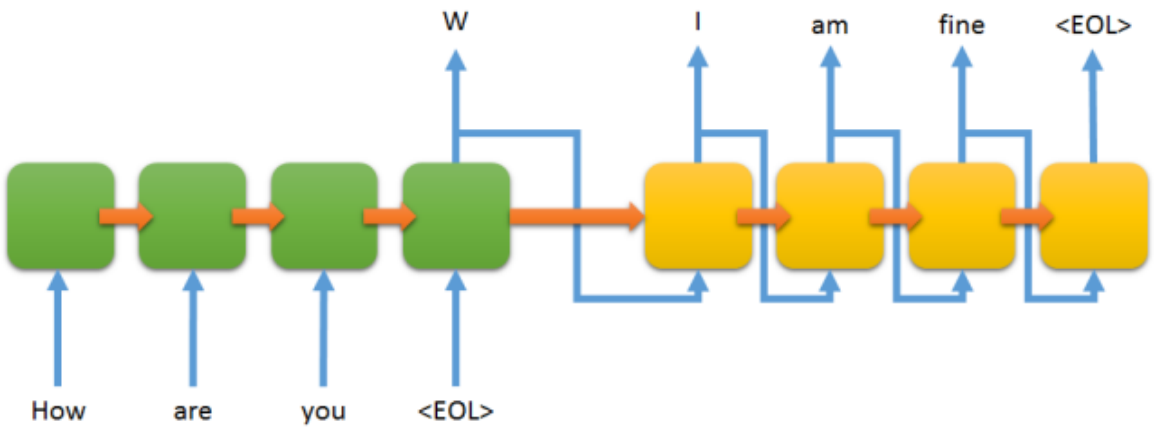
Image Description

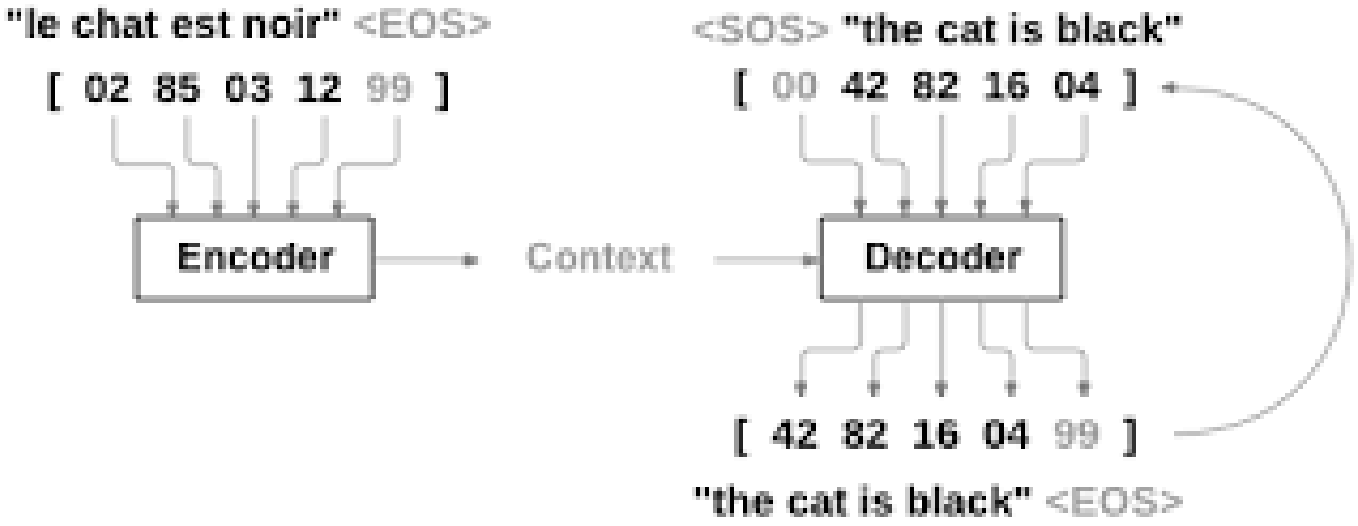
sequential output



- Jeff Donahue, CVPR Caffe Tutorial, June 6, 2015

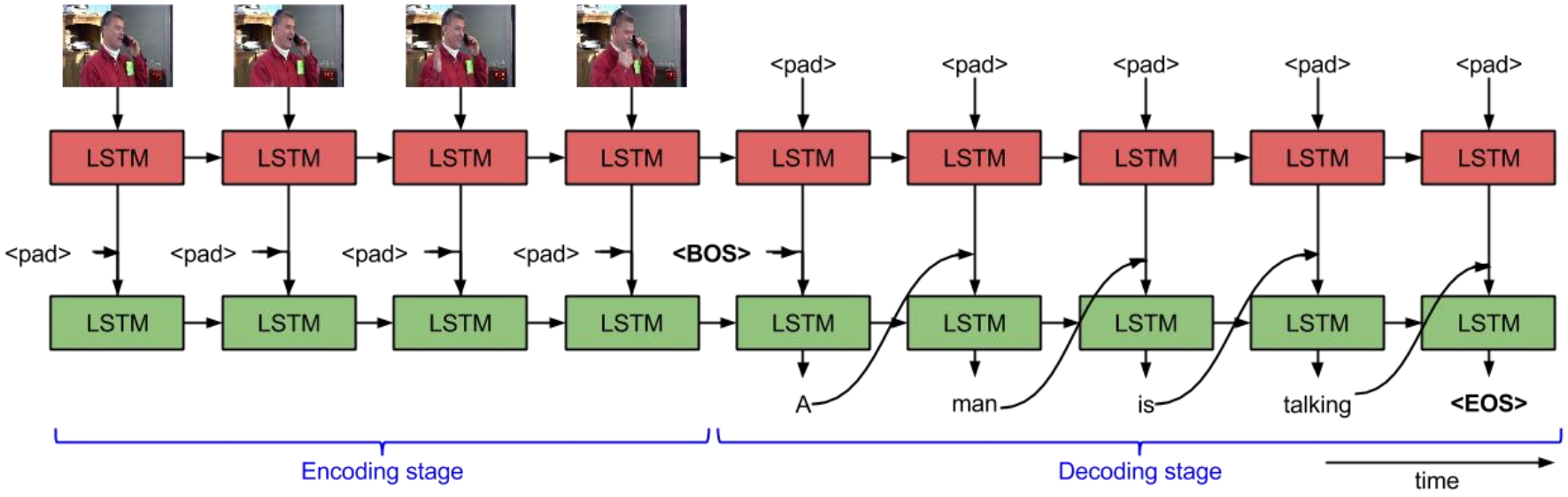
Machine translation, chatbots





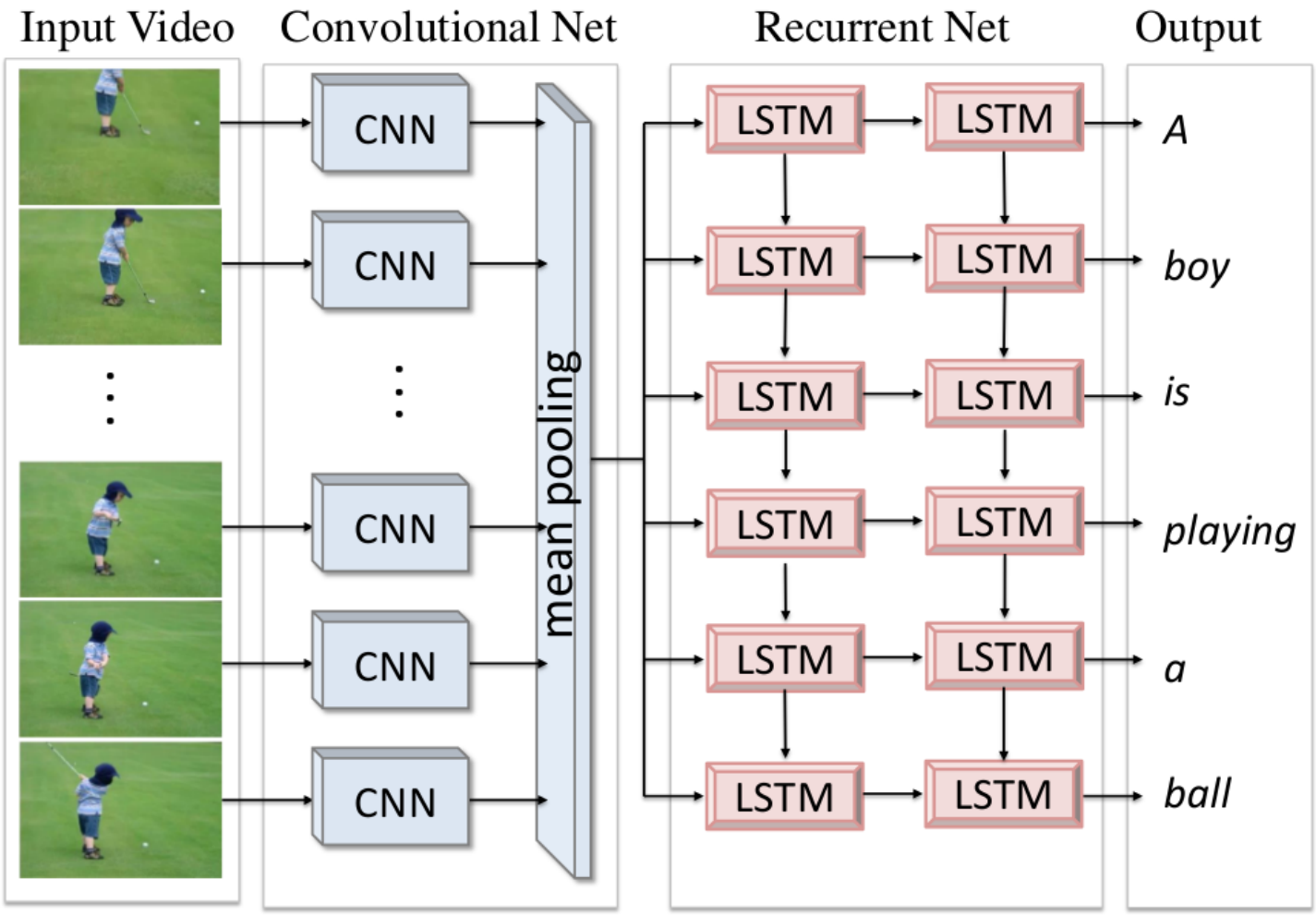
Sequence to Sequence - Video to Text

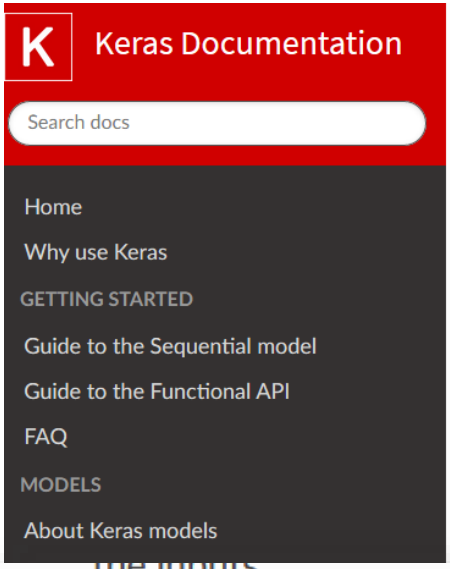
Subhashini Venugopalan, et al.



Translating Videos to Natural Language Using Deep Recurrent Neural Networks

Subhashini Venugopalan, et al.





- **go_backwards**: Boolean (default False). If True, process the input sequence backwards and return the reversed sequence.
- **stateful**: Boolean (default False). If True, the last state for each sample at index i in a batch will be used as initial state for the sample of index i in the following batch.
- **unroll**: Boolean (default False). If True, the network will be unrolled, else a symbolic loop will be used. Unrolling can speed-up a RNN, although it tends to be more memory-intensive. Unrolling is only suitable for short sequences.
- **reset_after**: GRU convention (whether to apply reset gate after or before matrix multiplication). False = "before" (default), True = "after" (CuDNN compatible).

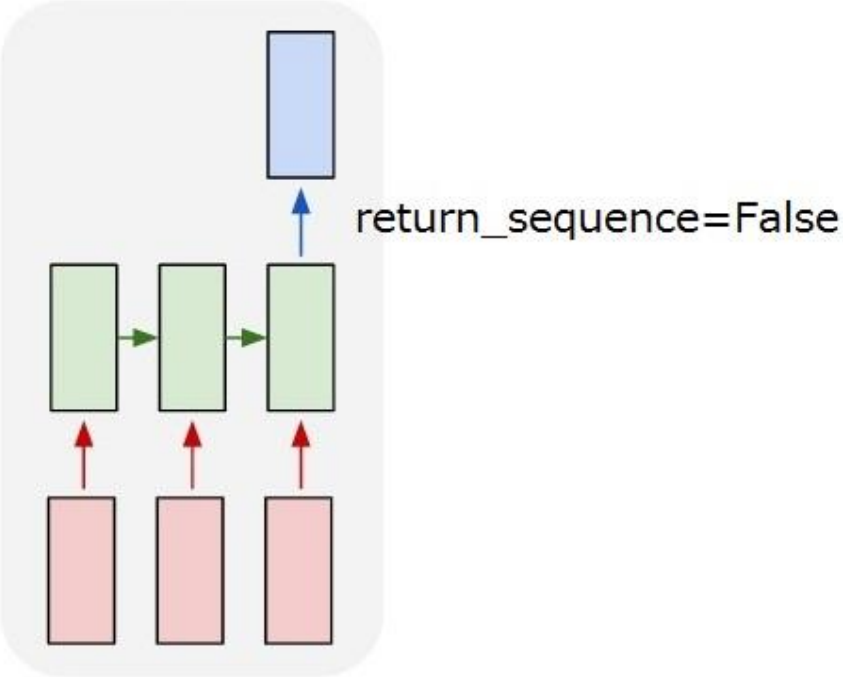
References

- Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
- On the Properties of Neural Machine Translation: Encoder-Decoder Approaches

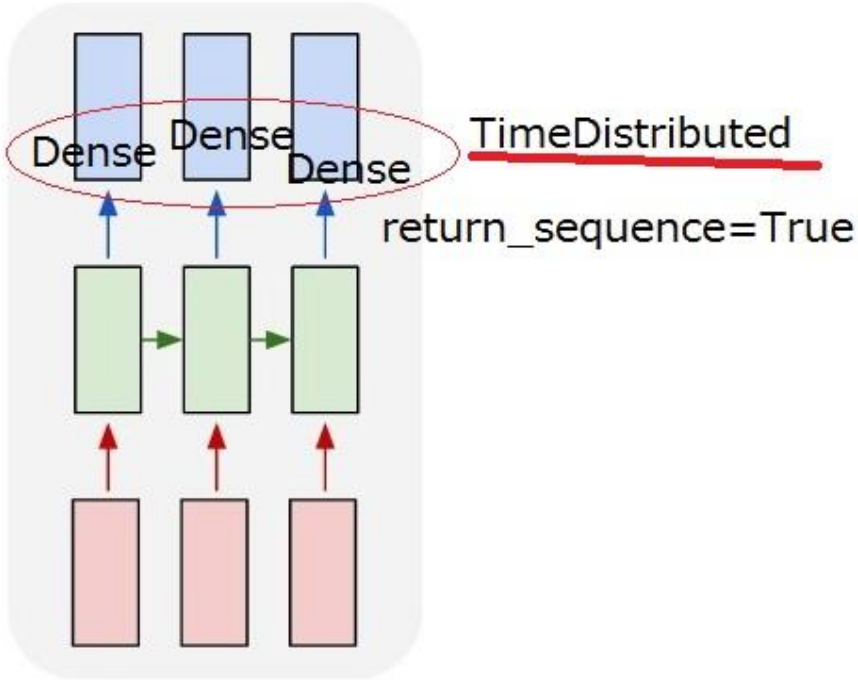
the inputs.

- **recurrent_dropout**: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the recurrent state.
- **implementation**: Implementation mode, either 1 or 2. Mode 1 will structure its operations as a larger number of smaller dot products and additions, whereas mode 2 will batch them into fewer, larger operations. These modes will have different performance profiles on different hardware and for different applications.
- **return_sequences**: Boolean. Whether to return the last output in the output sequence, or the full sequence.
- **return_state**: Boolean. Whether to return the last state in addition to the output. The returned elements of the states list are the hidden state and the cell state, respectively.
- **go_backwards**: Boolean (default False). If True, process the input sequence backwards and return the reversed sequence.

many to one



many to many (2)



CNN Long Short-Term Memory Networks

by **Jason Brownlee** on August 21, 2017 in **Long Short-Term Memory Networks**

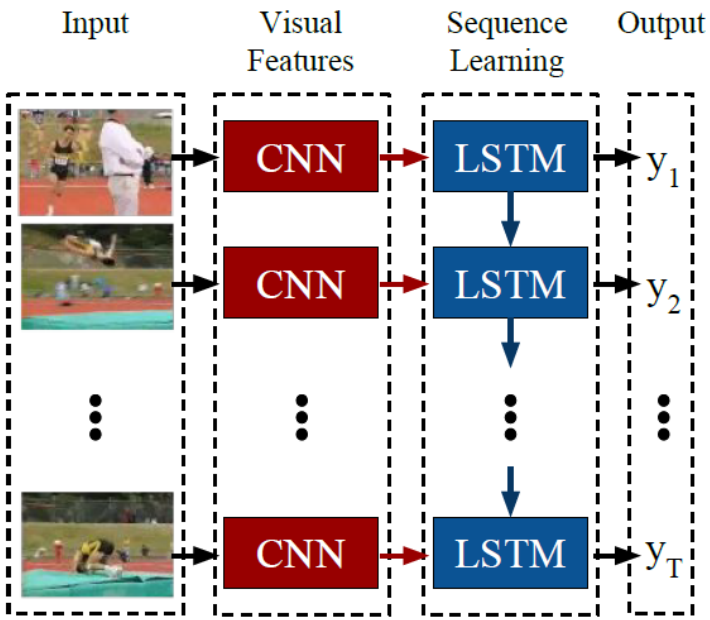
Tweet Share Share

Gentle introduction to CNN LSTM recurrent neural networks with example Python code.

Welcome to Machine Learning Mastery!

Hi, I'm **Jason Brownlee**, PhD. I write tutorials to help developers (*like you*) get results with machine learning.

Read More




```
model = Sequential()
# define CNN model
model.add(TimeDistributed(Conv2D(...))
model.add(TimeDistributed(MaxPooling2D(...)))
model.add(TimeDistributed(Flatten()))
# define LSTM model
model.add(LSTM(...))
model.add(Dense(...))
```

RNN essentially describe programs:

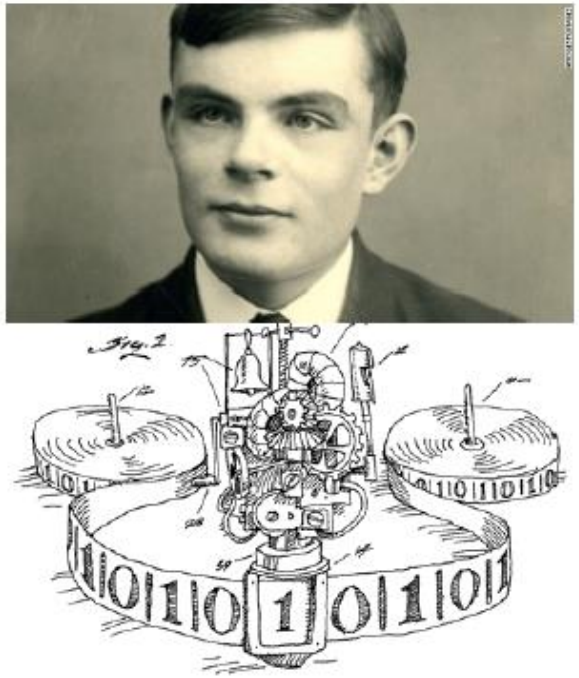
inputs + some internal variables.

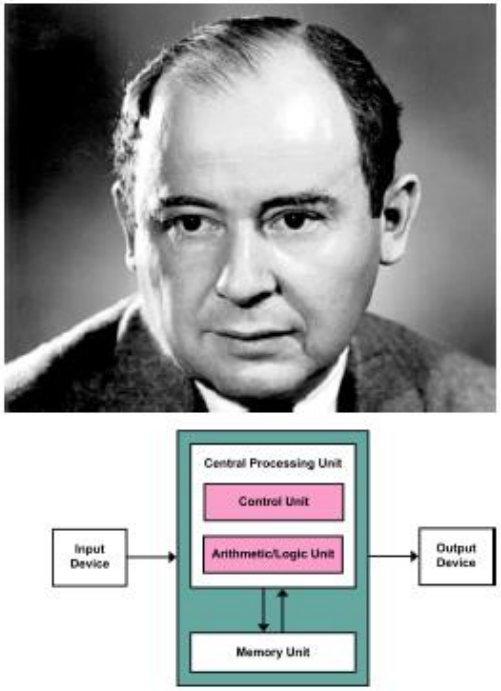
In fact, it is known that RNNs are Turing-Complete in the sense that they can to simulate arbitrary programs (with proper weights).

Neural Turing Machines

Can neural nets learn programs?

Alex Graves
Greg Wayne
Ivo Danihelka





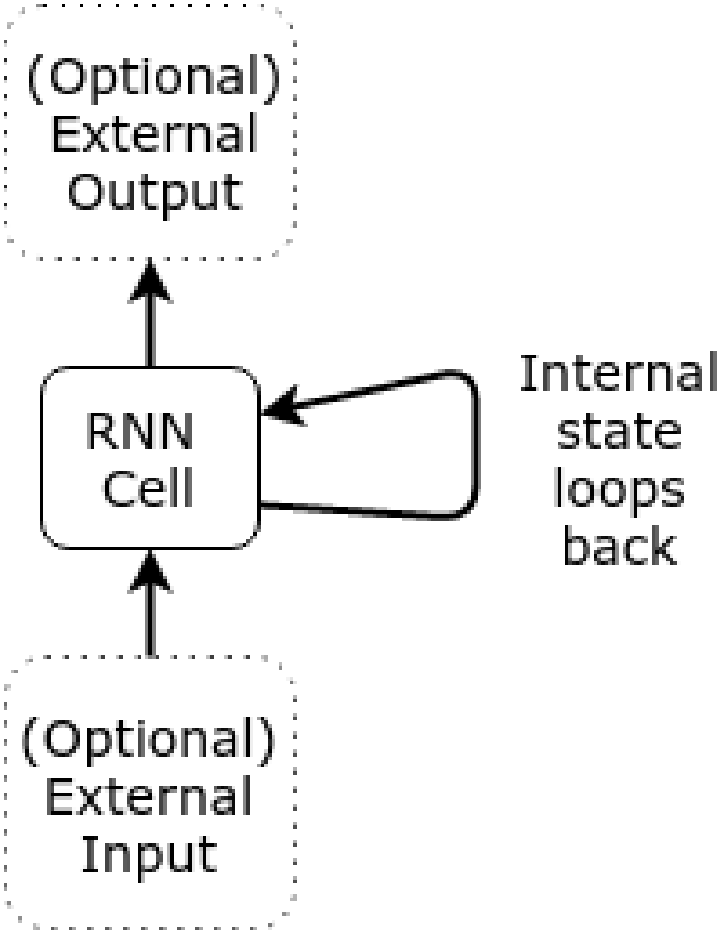
…path to **AI**??

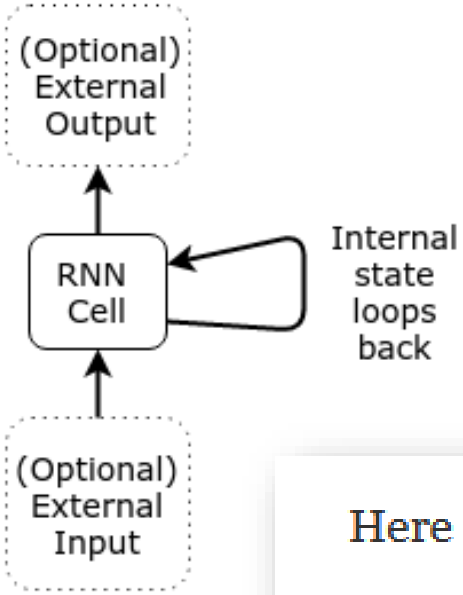
…. But " forget I said anything."

Andrej Karpathy
http://karpathy.github.io/2015/05/21/rnn-effectiveness

Written Memories: Understanding, Deriving and Extending the LSTM

http://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html



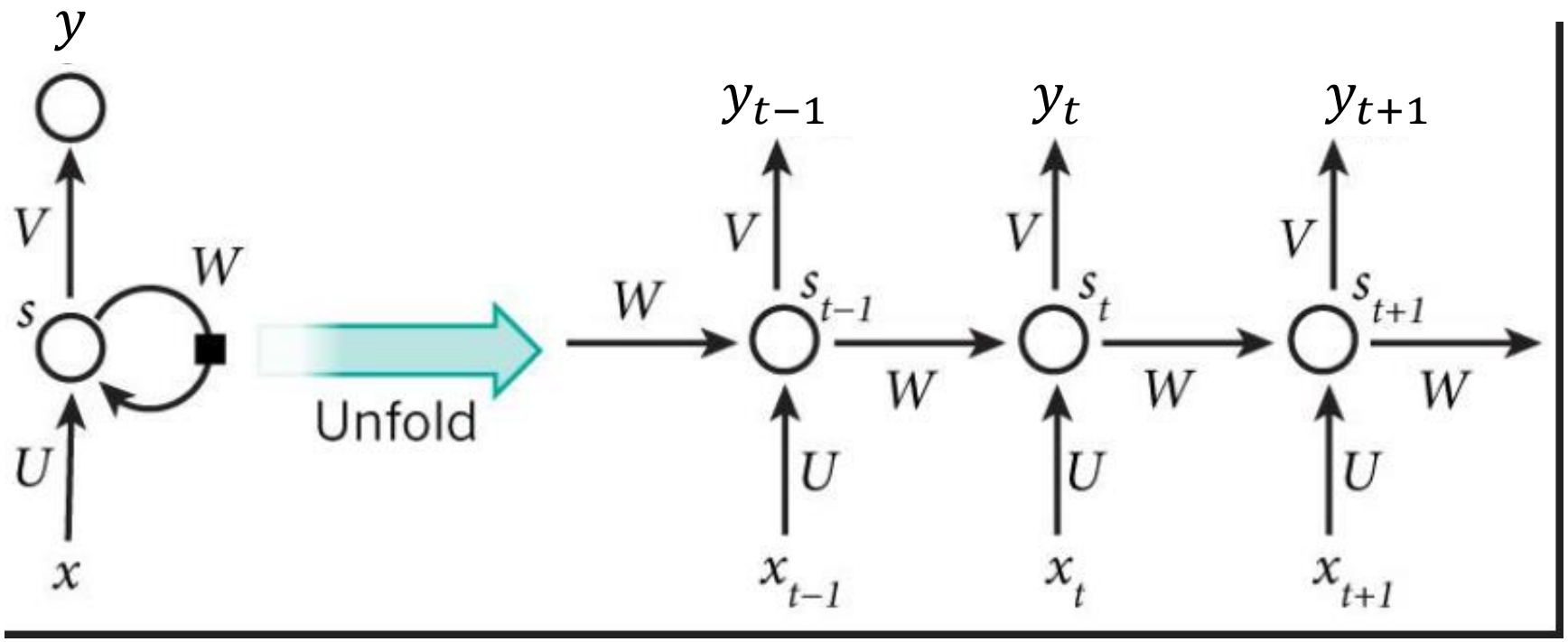


Here is the algebraic description of the RNN cell:

$$\begin{pmatrix} s_t \\ o_t \end{pmatrix} = f \begin{pmatrix} s_{t-1} \\ x_t \end{pmatrix}$$

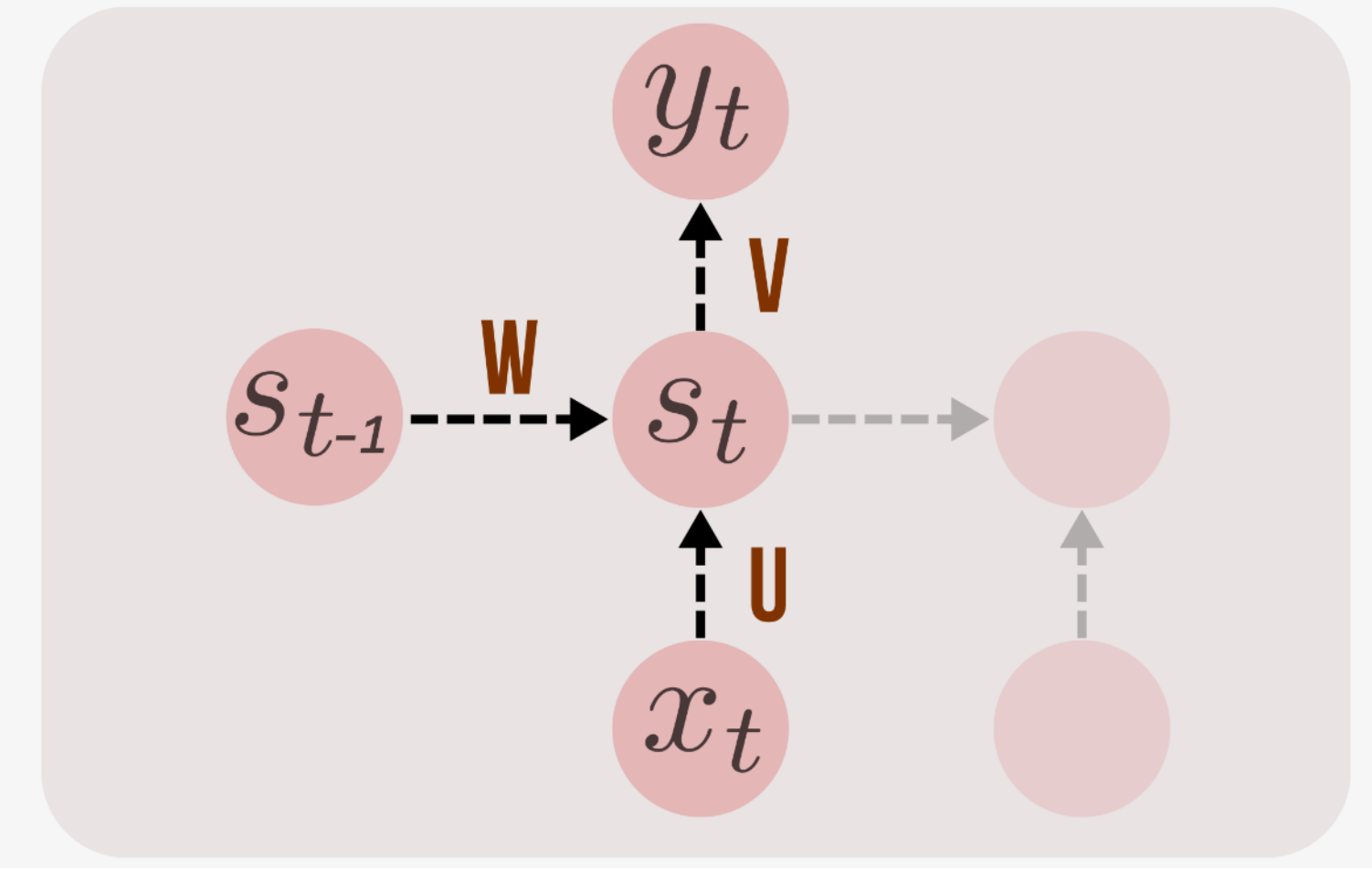
where:

- s_t and s_{t-1} are our current and prior states,
- o_t is our (possibly empty) current output,
- x_t is our (possibly empty) current input, and
- f is our recurrent function.


y
V
W
s
U
x
Unfold
W
y_{t-1}
V
s_{t-1}
U
x_{t-1}
W
y_t
V
s_t
U
x_t
W
y_{t+1}
V
s_{t+1}
U
x_{t+1}
W

TRAINING RNN: W, U, V ?

- **BPTT:** *Backpropagation Through Time*
- Truncated



Backpropagation Through Time (BPTT)

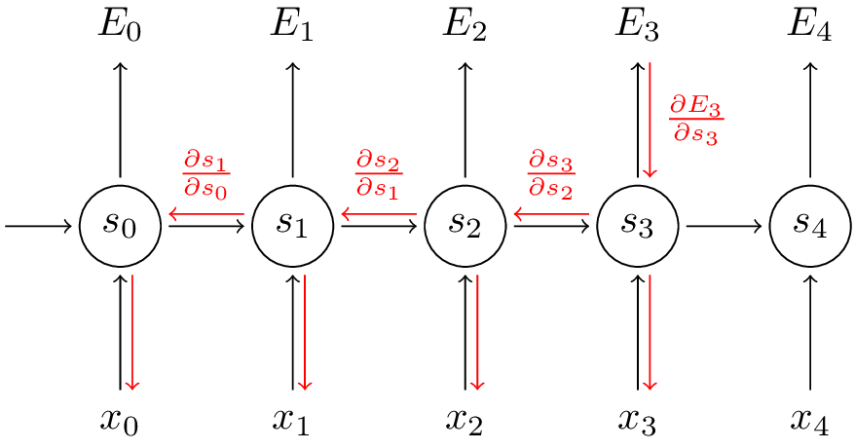
- Because the parameters are shared by all time steps in the network
- The gradient at each output depends not only on the calculations of the current time step, but also the previous time steps.

http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-2-implementing-a-language-model-rnn-with-python-numpy-and-theano/

Backpropagation Through Time (BPTT)

- But the recurrent net can be seen as a (very deep) feedforward net with shared weights
  - The forward pass builds up a stack of the activities of all the units at each time step.
  - The backward pass peels activities off the stack to compute the error derivatives at each time step.
  - After the backward pass we add together the derivatives at all the different times for each weight

$$\frac{\partial E_t}{\partial W} = \sum_{k=0}^{t} \frac{\partial E_t}{\partial y_y} \frac{\partial y_t}{\partial s_t} \frac{\partial s_t}{\partial s_k} \frac{\partial s_k}{\partial W}$$



Basic concepts: LSTM & GRU

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

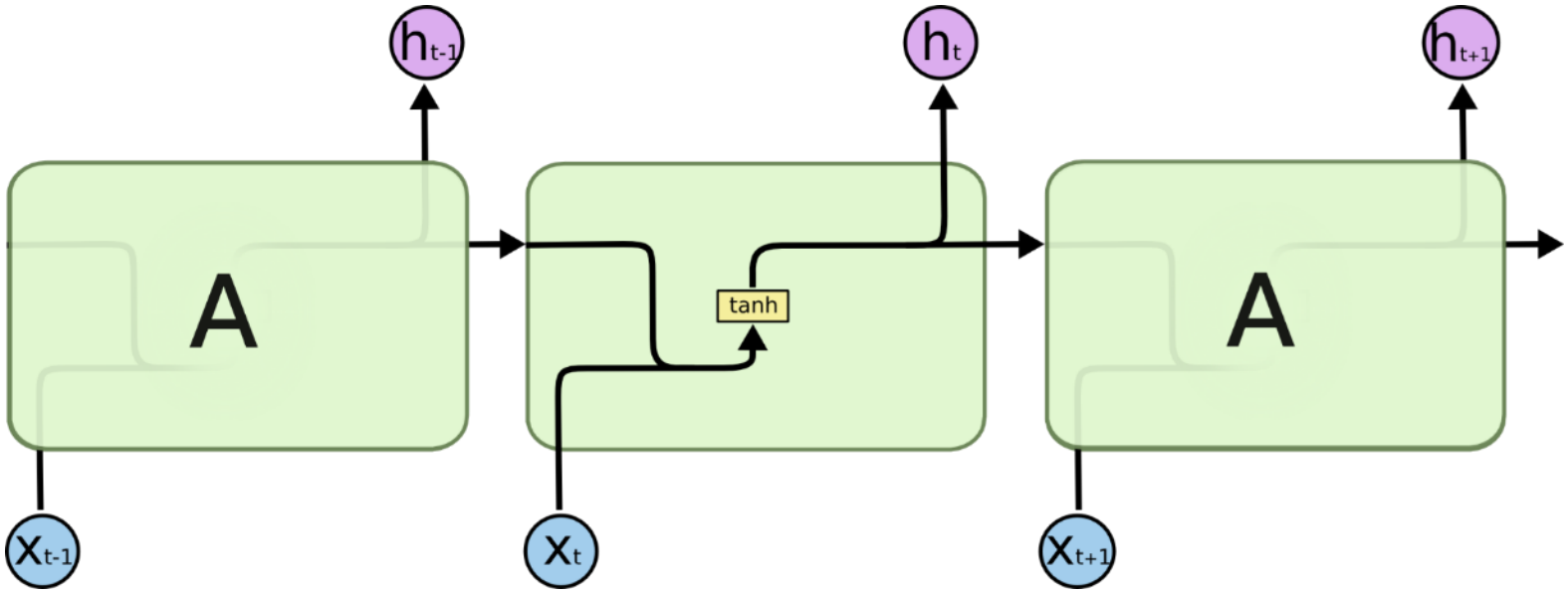
- **Vanishing gradients**: a problem?

In an RNN trained on long sequences (e.g. 100 time steps) the gradients can easily **explode** or **vanish.**

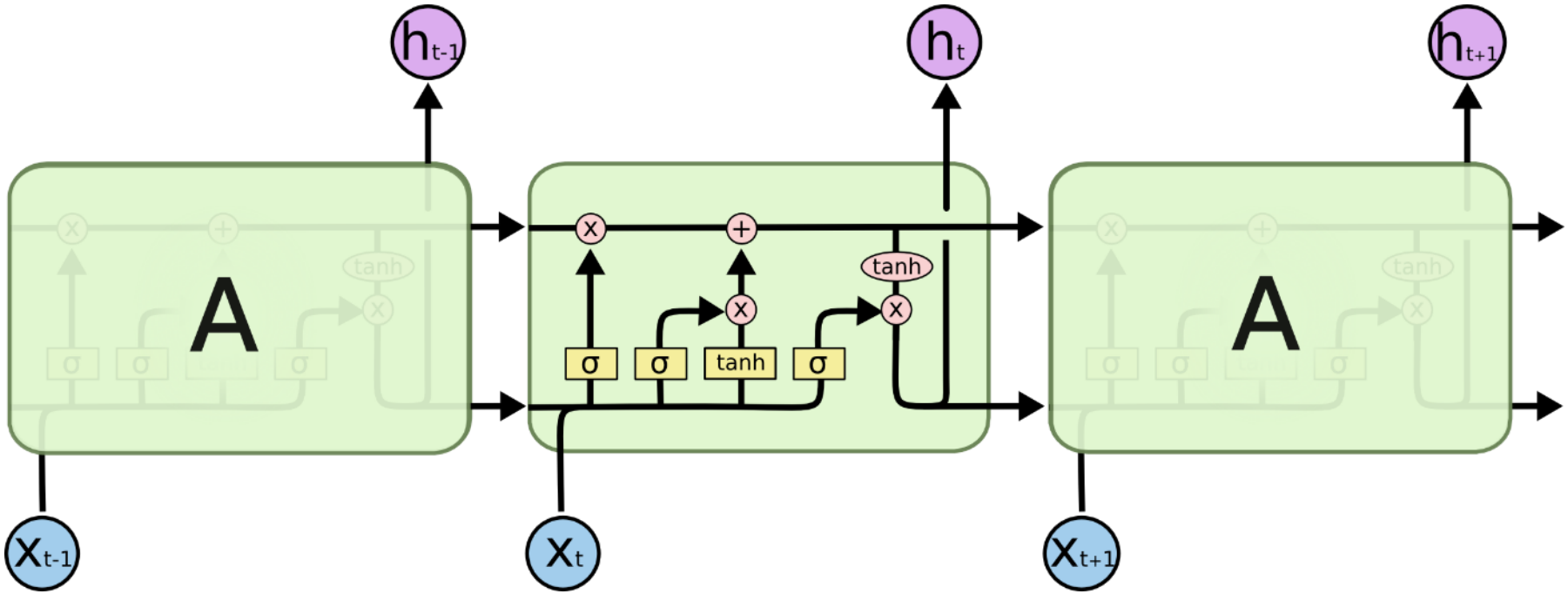
- Are RNNs capable of handling "**long-term dependencies**?"

Long Short Term Memory networks – LSTMs
Hocreiter and Schmidhuber (1997)

$NOTATION: states = h_t$

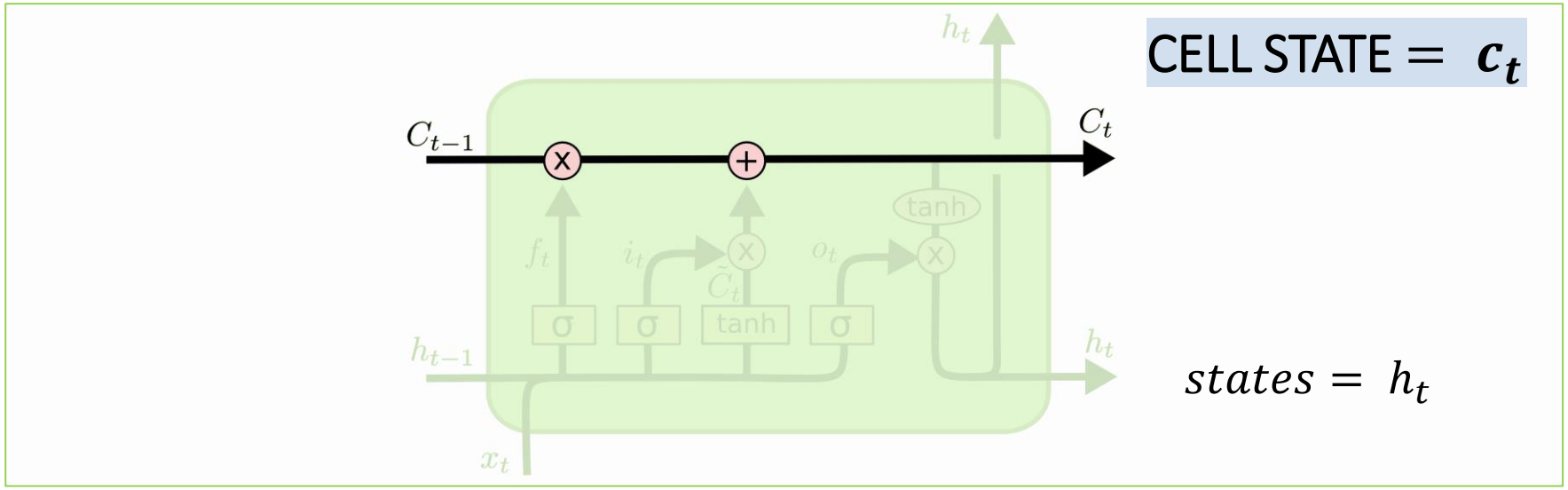


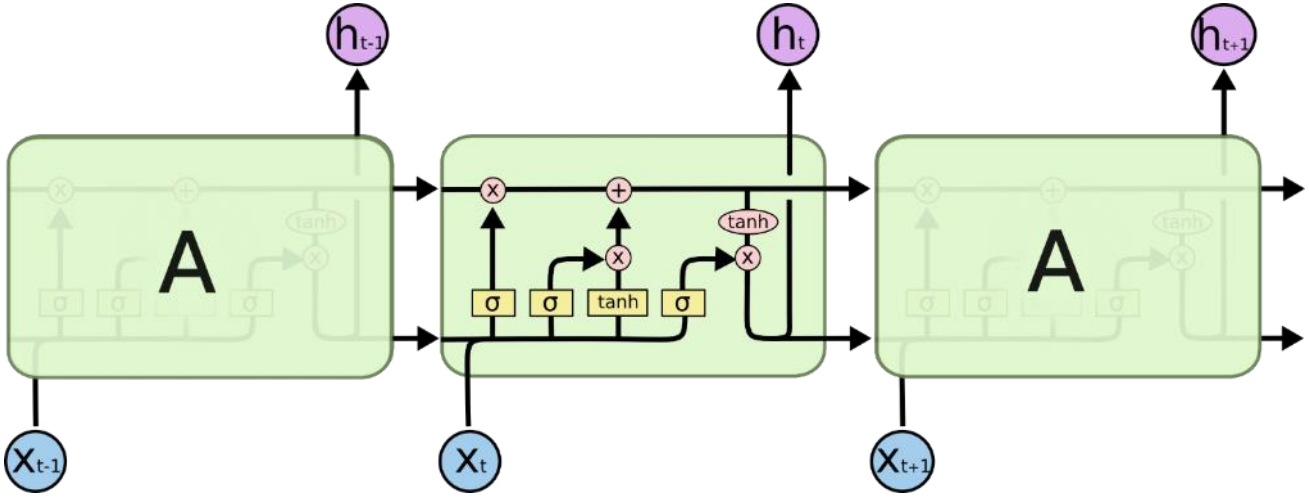
The repeating module in a standard RNN contains a single layer



The repeating module in LSTM contains four interacting layers

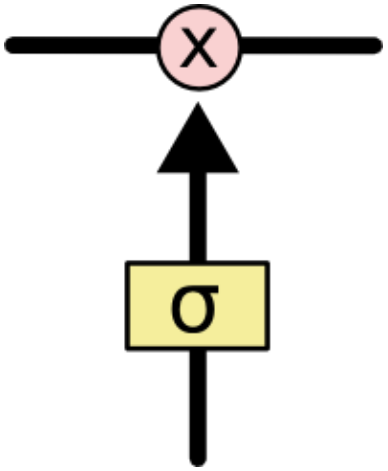
Core Idea Behind LSTM





Core Idea Behind LSTM

- The LSTM does have the ability to **remove or add information to the cell state,** carefully regulated by structures called *gates*.
- **Gates** are a way to optionally let information through.
- They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.



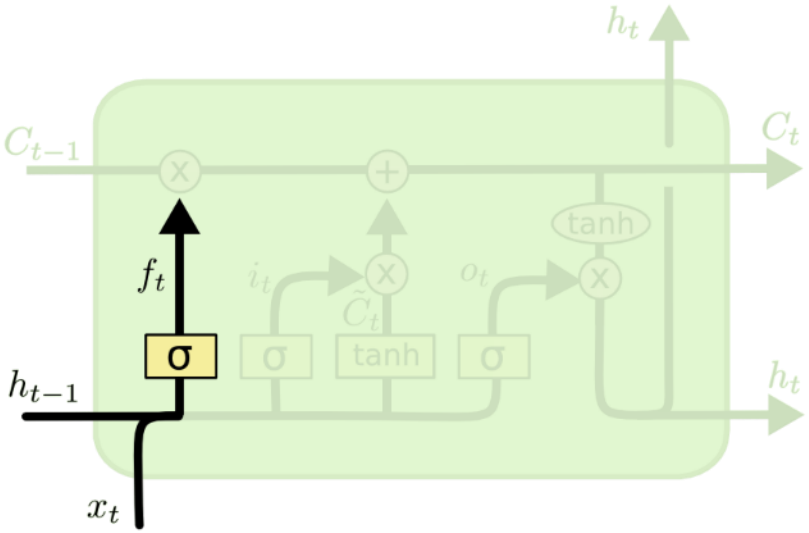
A value of zero means “let nothing through,” while a value of one means “let everything through!”

An LSTM has three of these gates, to protect and control the cell state

Step-by-Step LSTM Walk Through

Forget gate: The first step in our LSTM is to decide what information we're going to throw away from the cell state.

When we "see" something new relevant we decide forget….

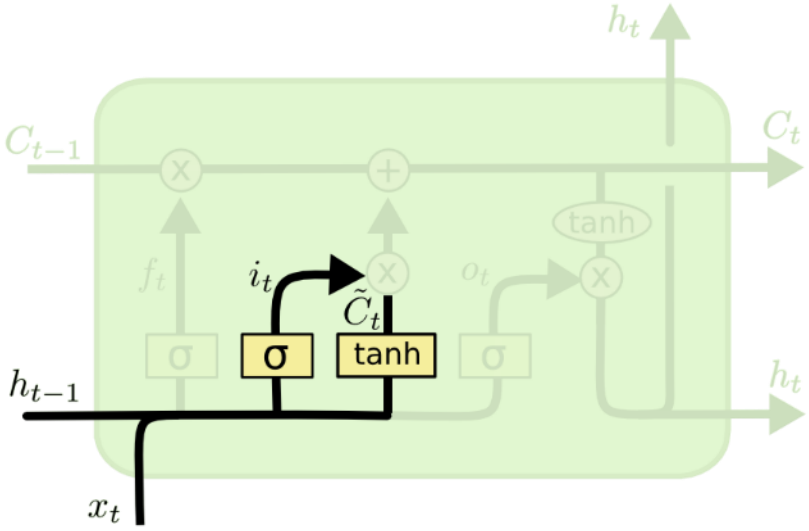


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Step-by-Step LSTM Walk Through

The next step is to decide what new information we're going to store in the cell state. This has two parts.

- First, a sigmoid layer called the "**Input gate layer**" decides which values we'll update.
- Next, a **tanh layer** creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state.



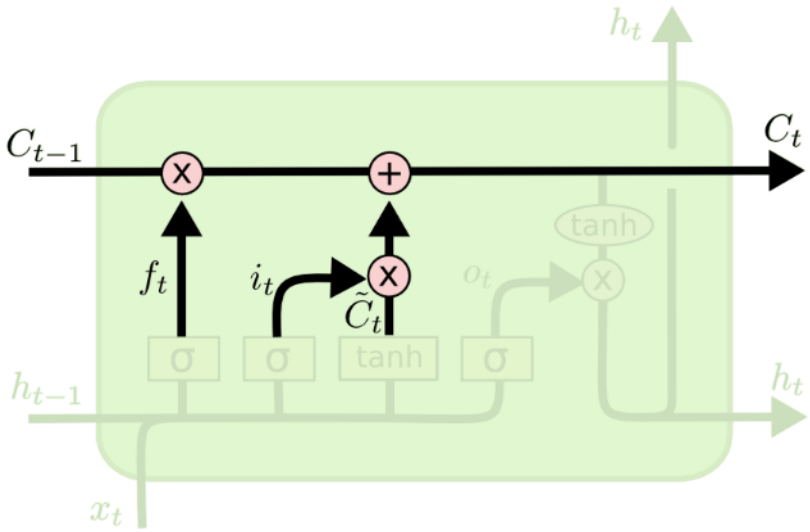
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Step-by-Step LSTM Walk Through

It's now time to update the old cell state, C_{t-1}, into the new cell state C_t

- The previous steps already decided what to do, we just need to actually do it.
- We multiply the old state by f_t(forget gate) then we add $i_t * \tilde{C}_t$

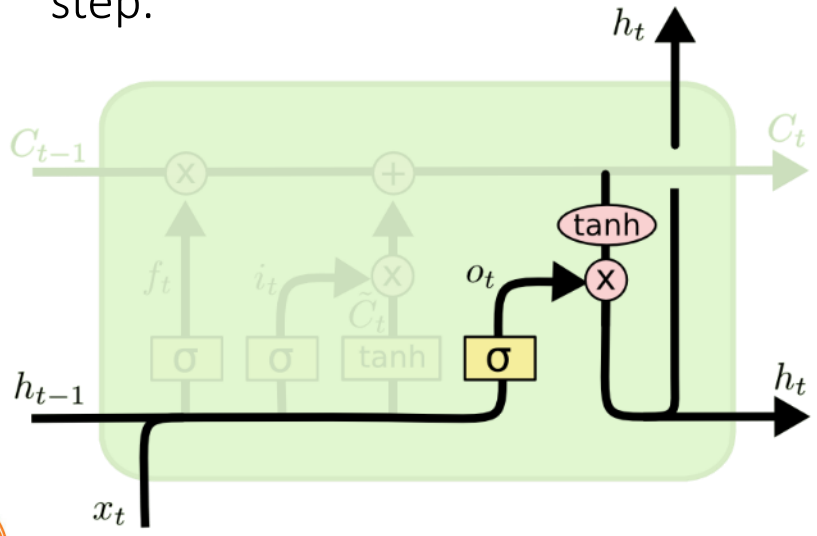


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Step-by-Step LSTM Walk Through

Finally, we need to decide **what we're going to output** h_t. This output will be based on our cell state, but will be a filtered version.

- First, we pass the previous hidden state and the current input into a sigmoid function.
- Then we pass the newly modified cell state to the tanh function. We multiply the tanh output with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state.
- The new cell state and the new hidden is then carried over to the next time step.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

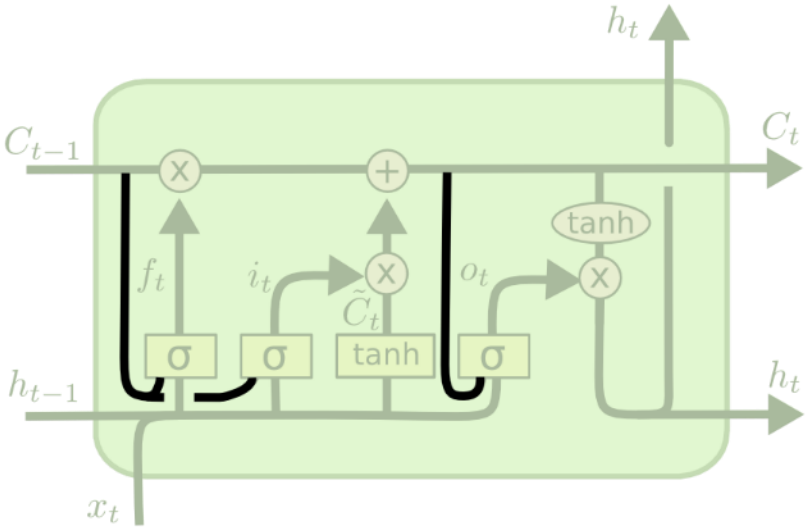
You can also see:

https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Variants on Long Short Term Memory

One popular LSTM variant, introduced by Gers & Schmidhuber (2000), is adding "peephole connections."

- This means that we let the gate layers look at the cell state.



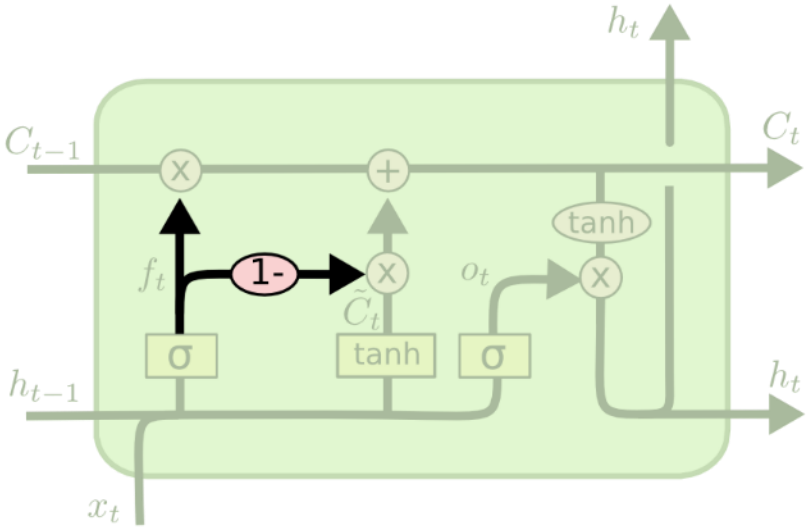
$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] + b_o\right)$$

The above diagram adds peepholes to all the gates, but many papers will give some peepholes and not others.

Variants on Long Short Term Memory

Another variation is to use coupled forget and input gates.

- We only input new values to the state when we forget something older



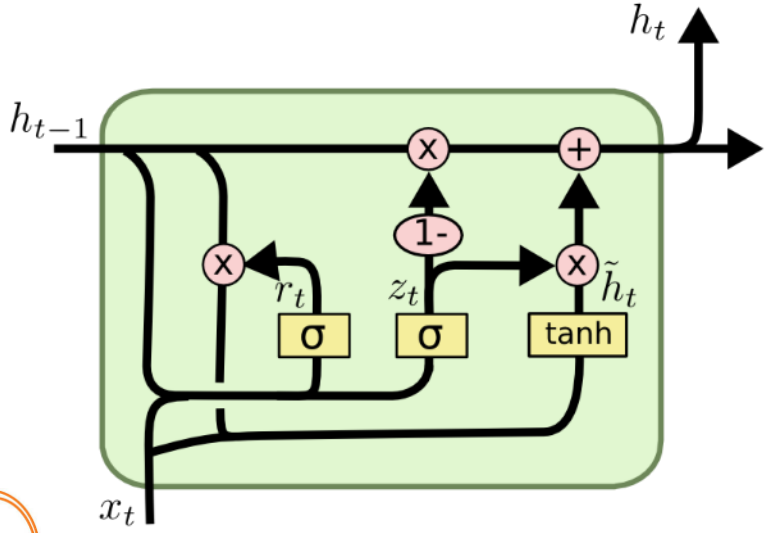
$$C_t = f_t * C_{t-1} + (1 - \boldsymbol{f_t}) * \tilde{C}_t$$

Variants on Long Short Term Memory

Gated Recurrent Unit, or GRU, introduced by Cho, et al. (2014).

- It combines the forget and input gates into a single "update gate."
- It also merges the cell state and hidden state, and makes some other changes.

The resulting model is simpler than standard LSTM models, and has been growing increasingly popular.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

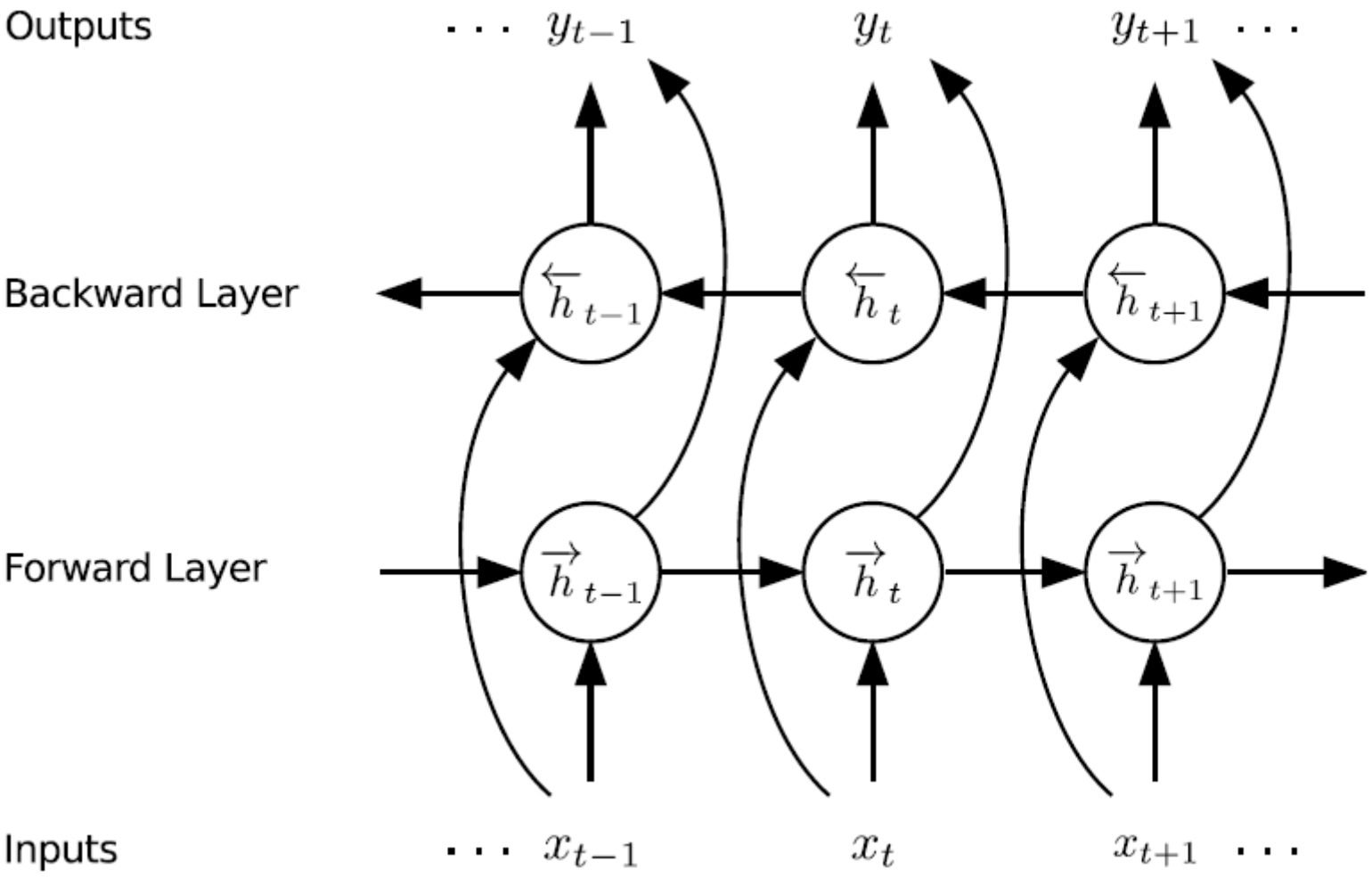
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Variants on Long Short Term Memory

These are only a few of the most notable LSTM variants.

See for example: **BLSTM** Bi-directional LSTM



Variants on Long Short Term Memory

Which of these variants is best? Do the differences matter?

- Greff, et al. (2015) do a nice comparison of popular variants, finding that they're all about the same.
- Jozefowicz, et al. (2015) tested more than ten thousand RNN architectures, finding some that worked better than LSTMs on certain tasks.

Simple Examples: NLP & MUSIC

Text Prediction/Generation with Keras using LSTM: *Long Short Term Memory* networks

In this example we will work with the book: Alice's Adventures in Wonderland by Lewis Carroll.

We are going to learn the dependencies between characters and the conditional probabilities of characters in sequences so that we can in turn generate wholly new and original sequences of characters.

Adapted from:

Text Generation With LSTM Recurrent Neu

By Jason Brownlee

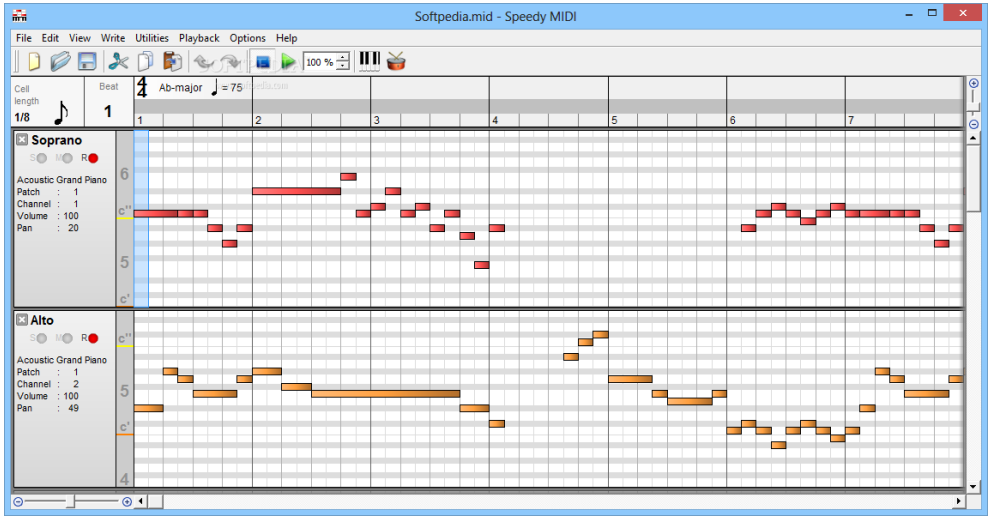
See example using the IMDB movie review sentiment prediction.

"IMDB dataset", a set of 50,000 highly-polarized reviews from the Internet Movie Database. They are split into 25,000 reviews for training and 25,000 reviews for testing, each set consisting in 50% negative and 50% positive reviews.

Just like the MNIST dataset, the IMDB dataset comes packaged with Keras. It has already been preprocessed: the reviews (sequences of words) have been turned into sequences of integers, where each integer stands for a specific word in a dictionary.

Keras

MIDI Music Generation



Simple Examples (I): NLP & MUSIC

IPTC_Keras_RNN_Text.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

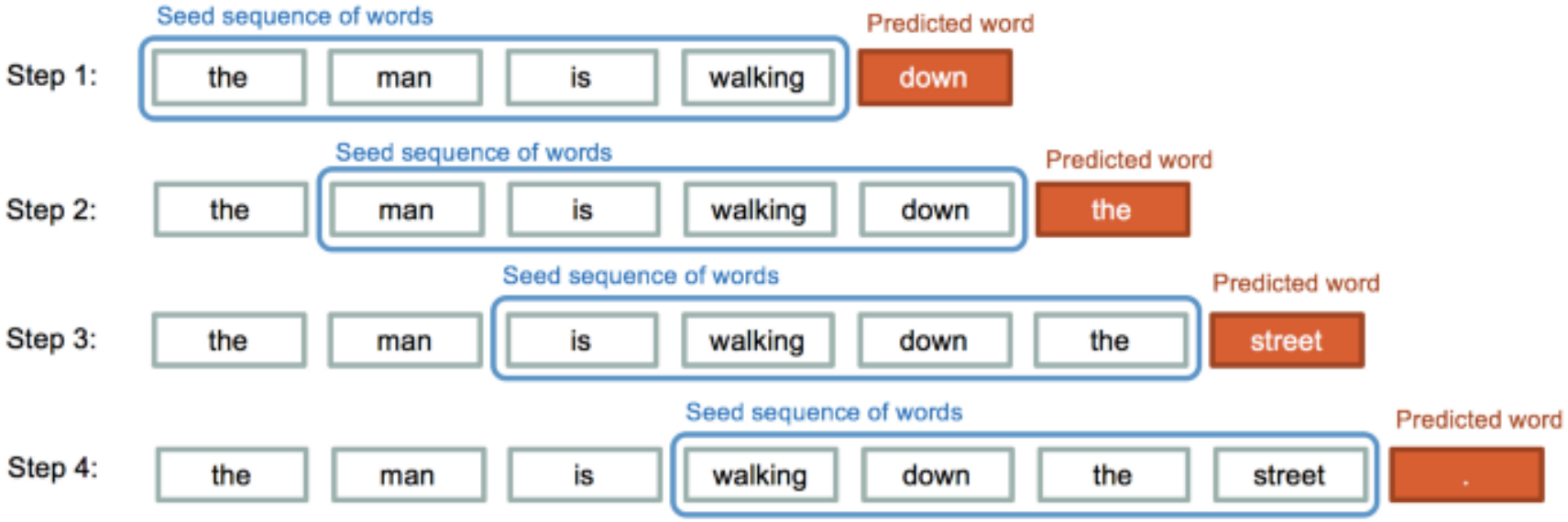
CÓDIGO TEXTO | CELDA CELDA

Text Prediction/Generation with Keras using LSTM: *Long Short Term Memory* networks

In this example we will work with the book: Alice's Adventures in Wonderland by Lewis Carroll.

We are going to learn the dependencies between characters and the conditional probabilities of characters in sequences so that we can in turn generate wholly new and original sequences of characters.

Google Colab: Reads a shared file in a Drive



https://medium.com/@david.campion/text-generation-using-bidirectional-lstm-and-doc2vec-models-1-3-8979eb65cb3a

OpenAI GPT-2 writes alternate endings for Game of Thrones

Chintan Trivedi in Towards Data Science Follow
May 23 · 12 min read ★

I trained the GPT-2 language model on GRRM's book series *"A Song of Ice and Fire"* and let it complete the HBO show's storyline. Can it do better than HBO's season 8 train-wreck?

https://towardsdatascience.com/openai-gpt-2-writes-alternate-endings-for-game-of-thrones-c9be75cd2425

Simple Examples (II): NLP & MUSIC

IPTC_Music_Generation_LSTM.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

CÓDIGO TEXTO CELDA CELDA

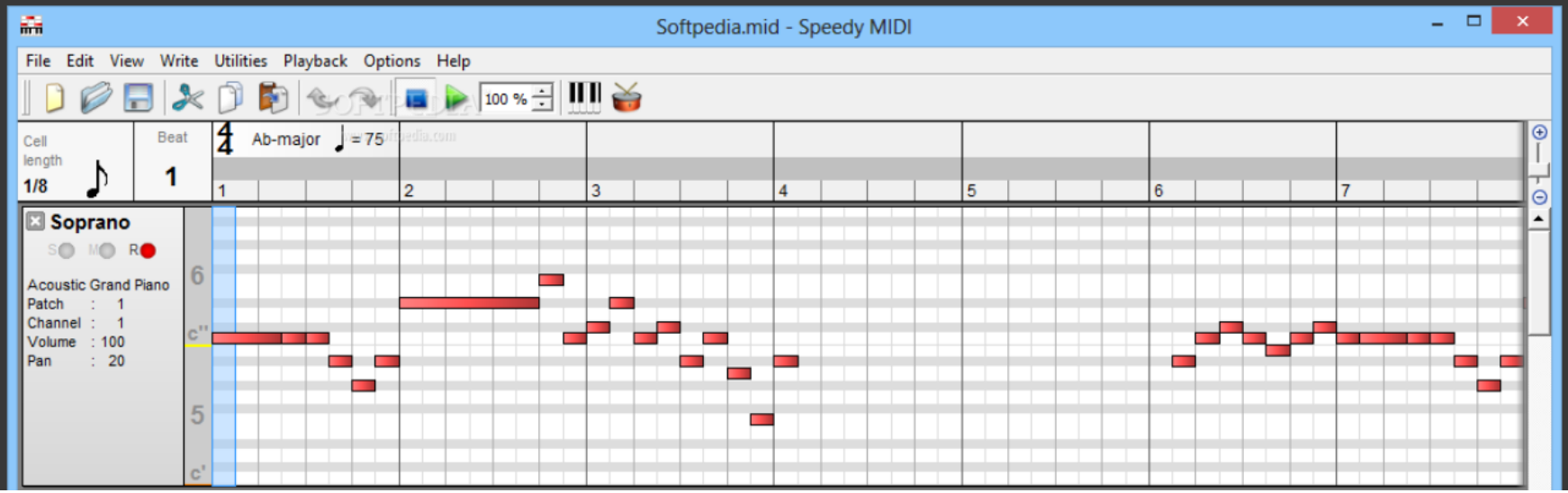
Google Colab:

- mounts your google drive

Music generation

To train our model, we will use MIDI files. MIDI files contain **information about a music composition**, not the music itself. They contain information about notation, pitch, velocity, vibrato, panning, and clock signals (which set tempo).

There are some synthesizers which are able to transform this composition information into a real audio track. Our model will learn from these **MIDI files** and will be able to generate new ones.



Simple Examples (III): NLP & MUSIC

Sentiment Analysis

See example using the IMDB movie review sentiment prediction.

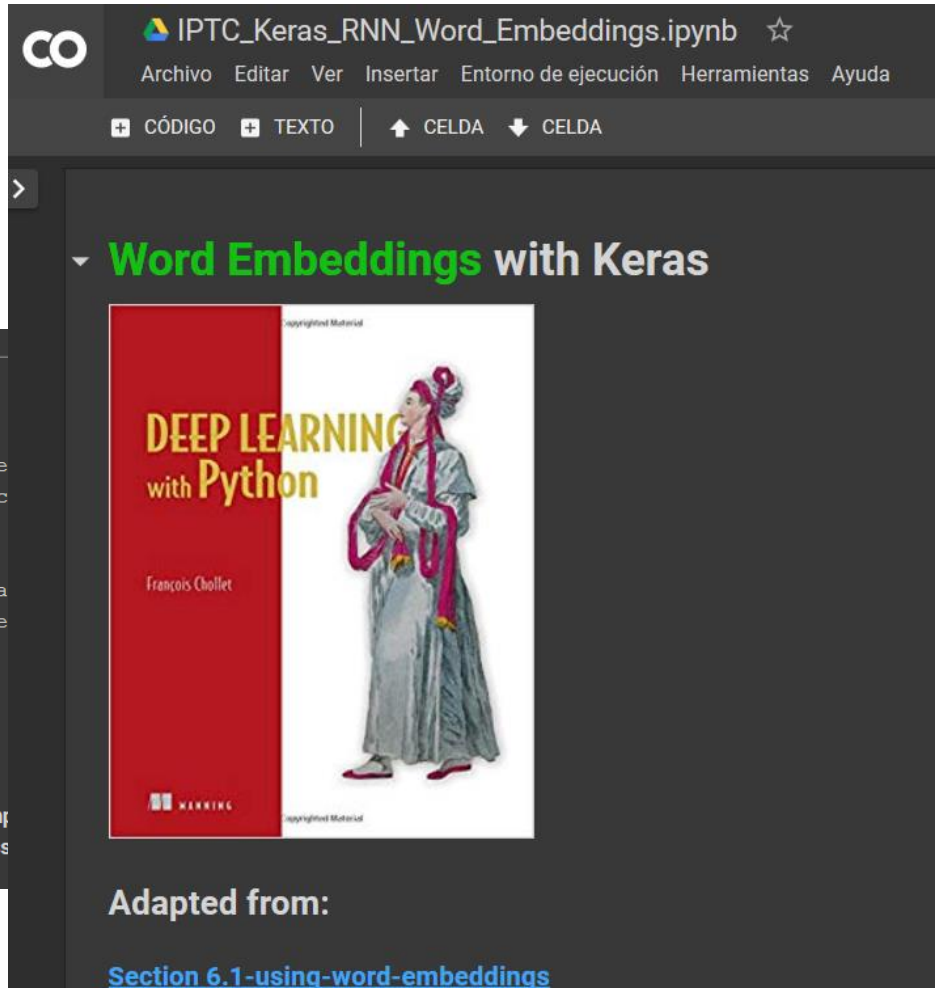
```
"IMDB dataset", a set of 50,000 highly-polarized reviews from the Inte
25,000 reviews for training and 25,000 reviews for testing, each set c
reviews.

Just like the MNIST dataset, the IMDB dataset comes packaged with Kera
reviews (sequences of words) have been turned into sequences of intege
word in a dictionary.
```

Let's quickly prepare the data.

- We will restrict the movie reviews to the top 10,000 most common words
- and cut the reviews after only 20 words.

Our network will simply learn 8-dimensional embeddings for each of the 10,000 words, turn the inp sequences (3D float tensor), flatten the tensor to 2D, and train a single Dense layer on top for class

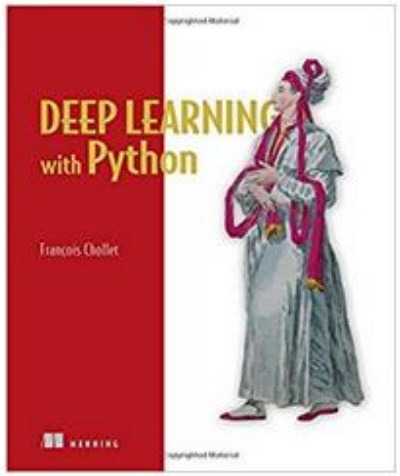


Embedding:

Indices by themselves, carry no **semantic** meaning

IPTC_Keras_RNN_Word_Embeddings.ipynb

Word Embeddings with Keras

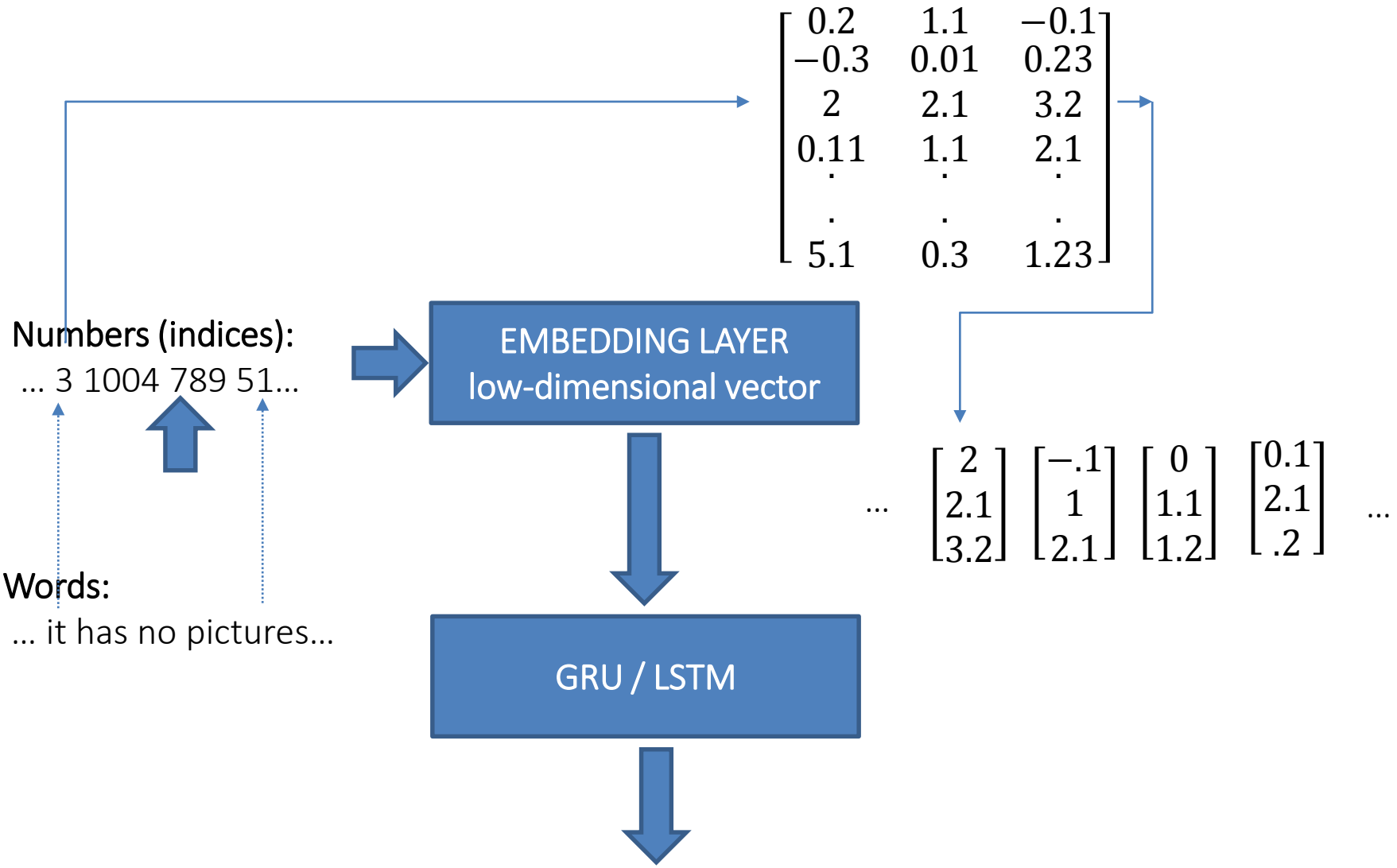


Adapted from:

6.1-using-word-embeddings

By François Chollet

Keras

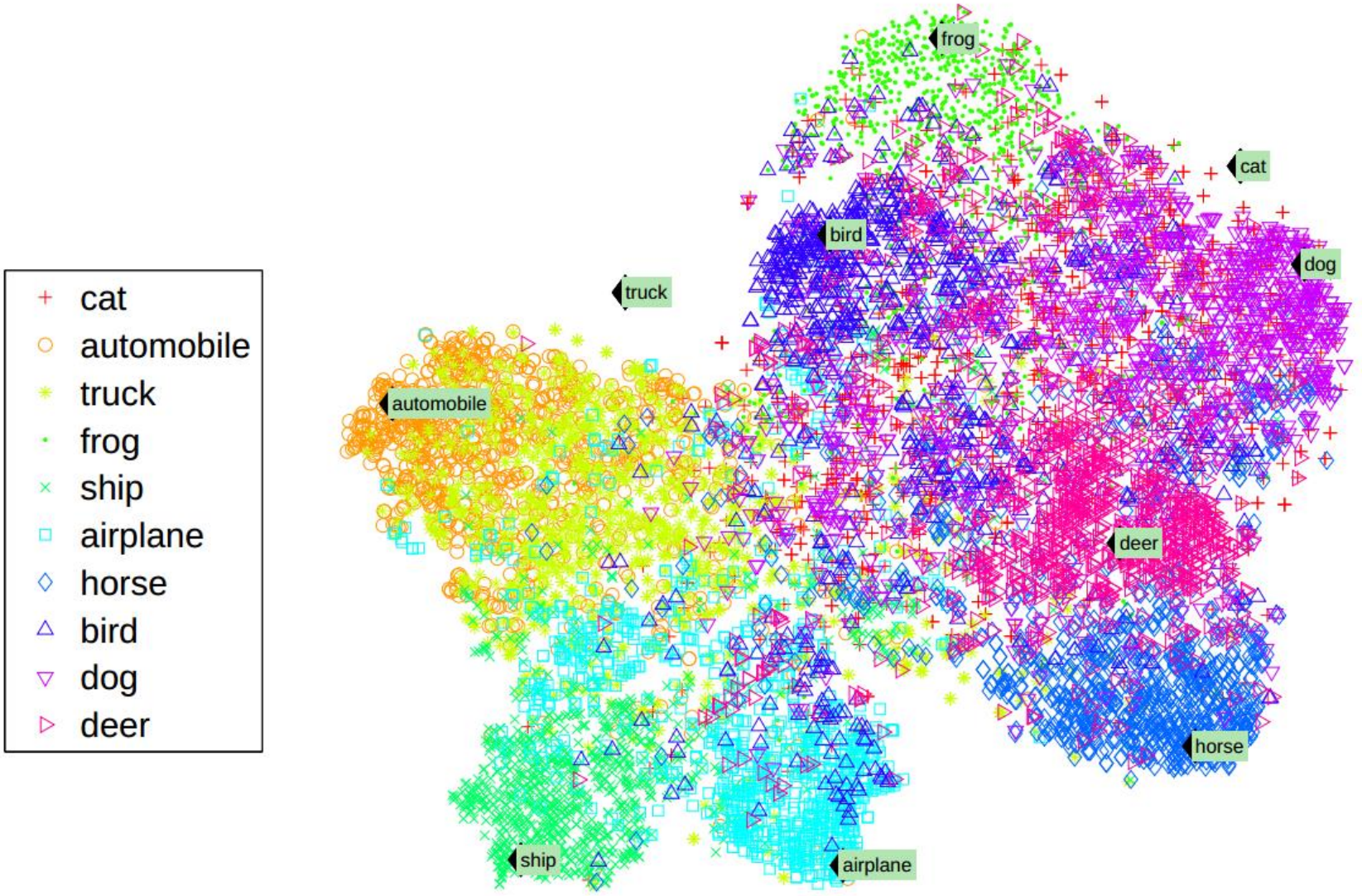


http://stackoverflow.com/questions/40184537/what-does-embedding-do-in-tensorflow

Embedings

- Indices by themselves, carry no semantic meaning
- This is where embedding comes in; more commonly known as *word vector* or *word embedding*.

http://sebastianruder.com/word-embeddings-1/



http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

See also:

Word2vec https://www.tensorflow.org/tutorials/word2vec

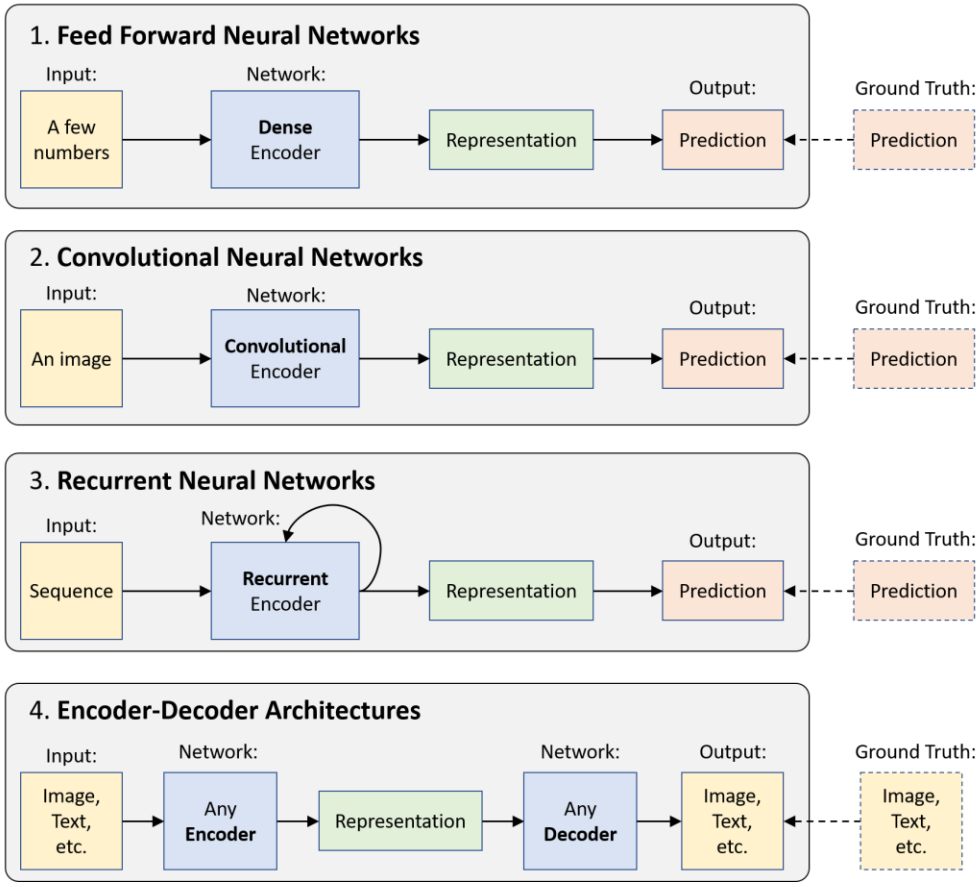
Vector space models (VSMs) represent (embed) words in a continuous vector space where semantically similar words are mapped to nearby points ('are embedded nearby each other')

Word2Vec

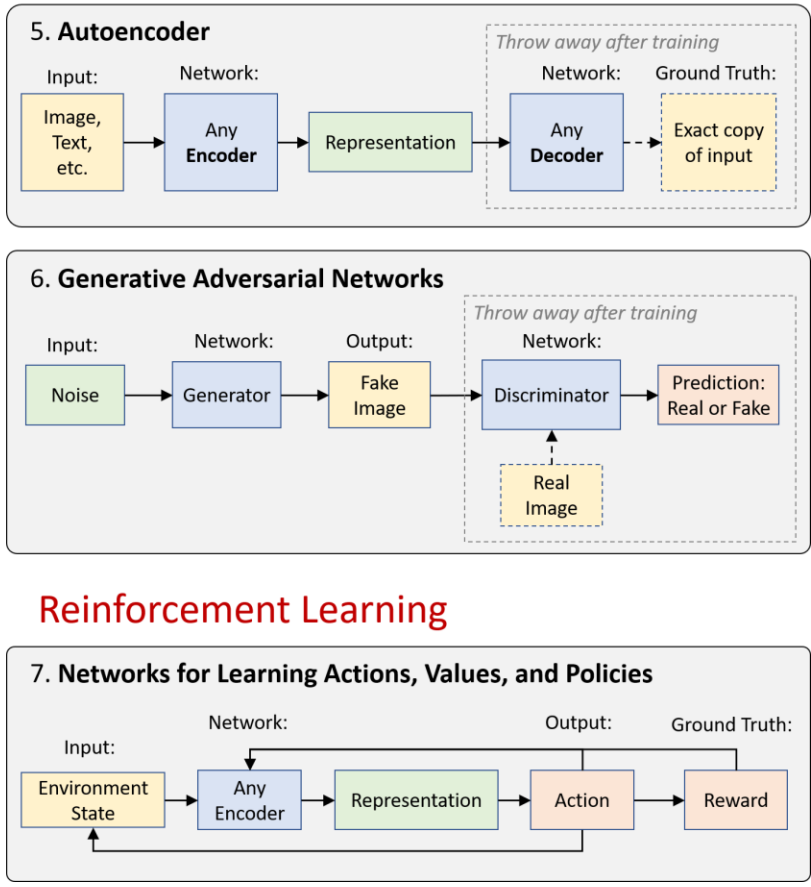
https://ronxin.github.io/wevi/

Deep learning basics

Supervised Learning



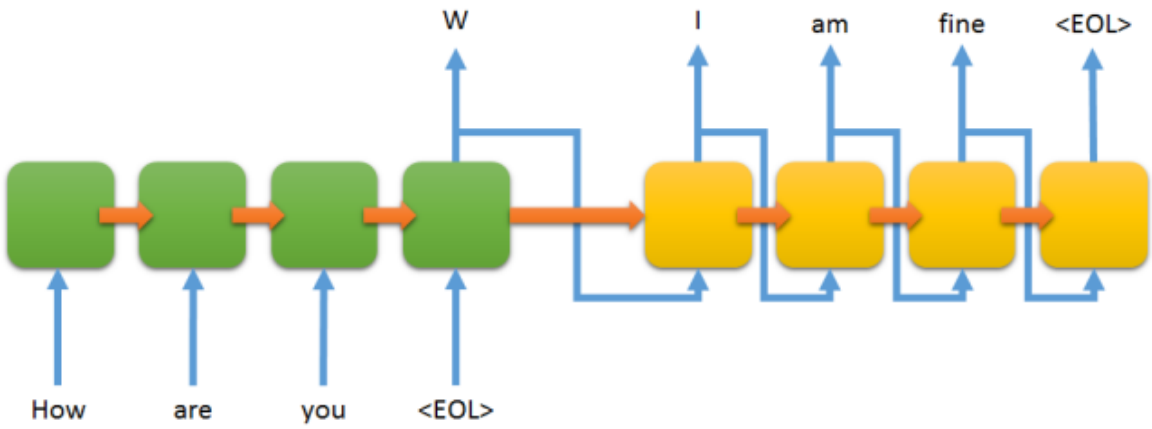
Unsupervised Learning

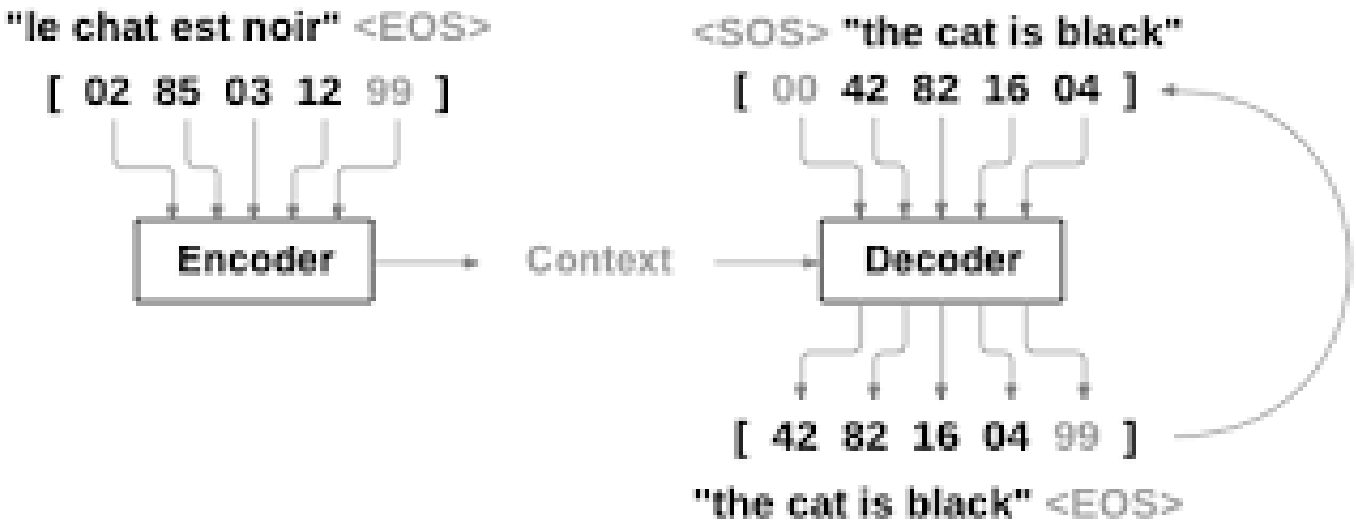


Reinforcement Learning

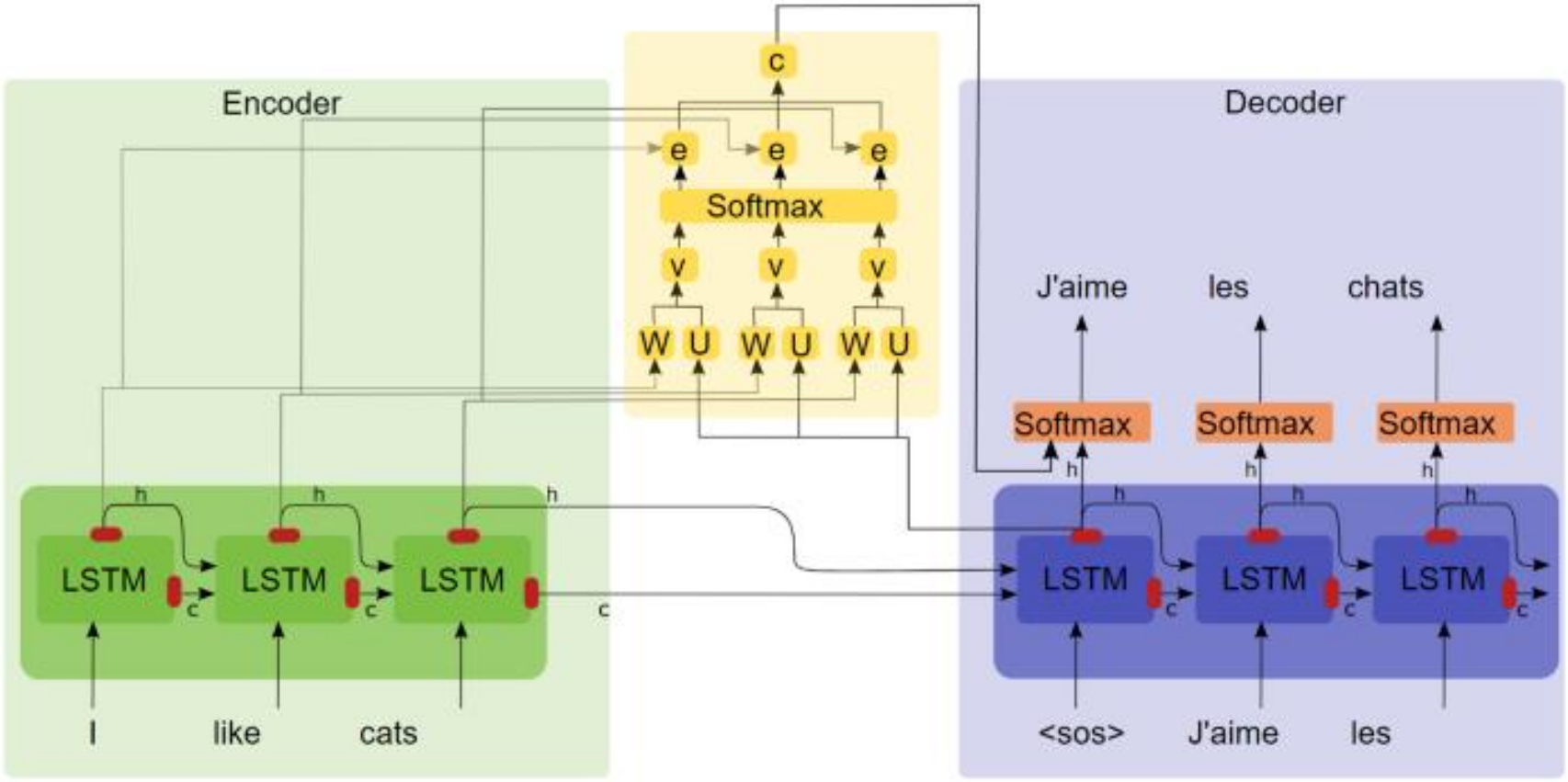
https://github.com/lexfridman/mit-deep-learning/blob/master/tutorial_deep_learning_basics/deep_learning_basics.ipynb

Machine translation, chatbots,…



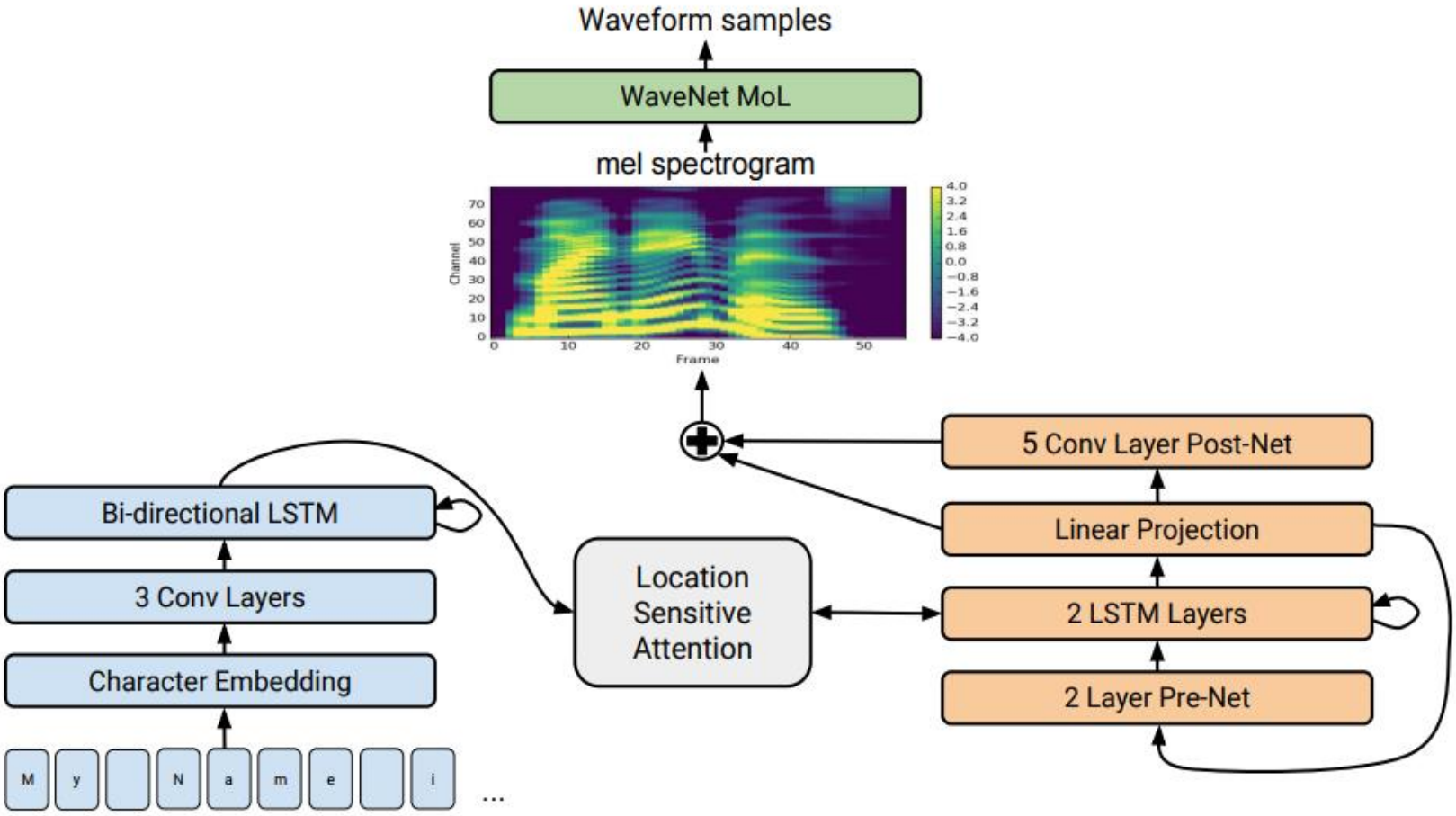


ATTENTION



https://towardsdatascience.com/light-on-math-ml-attention-with-keras-dc8dbc1fad39

END-TO-END Text-to-Speech

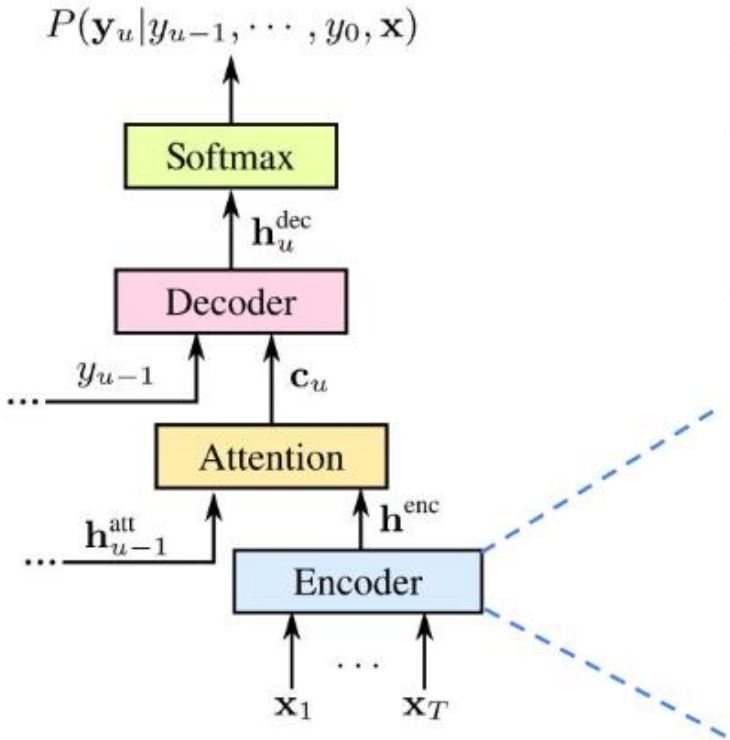


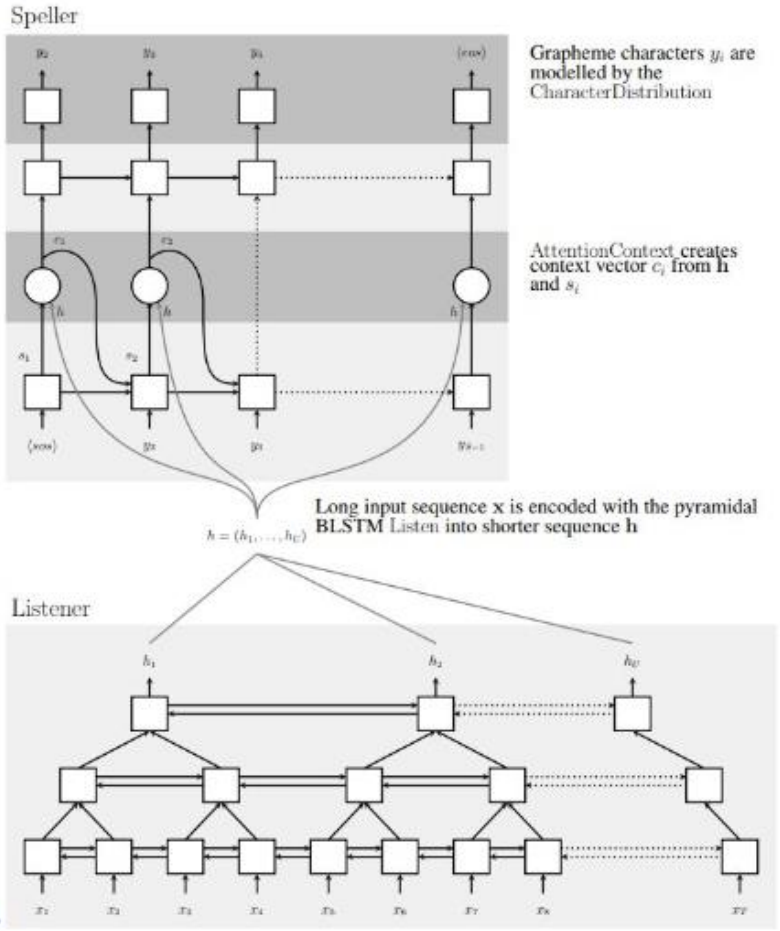
[Tacotron 2: Generating Human-like Speech from Text]

https://ai.googleblog.com/2017/12/tacotron-2-generating-human-like-speech.html

END-TO-END Automatic Speech Recognition

Attention-based Models





Reproduced from [Chan et al., 2015]

Google

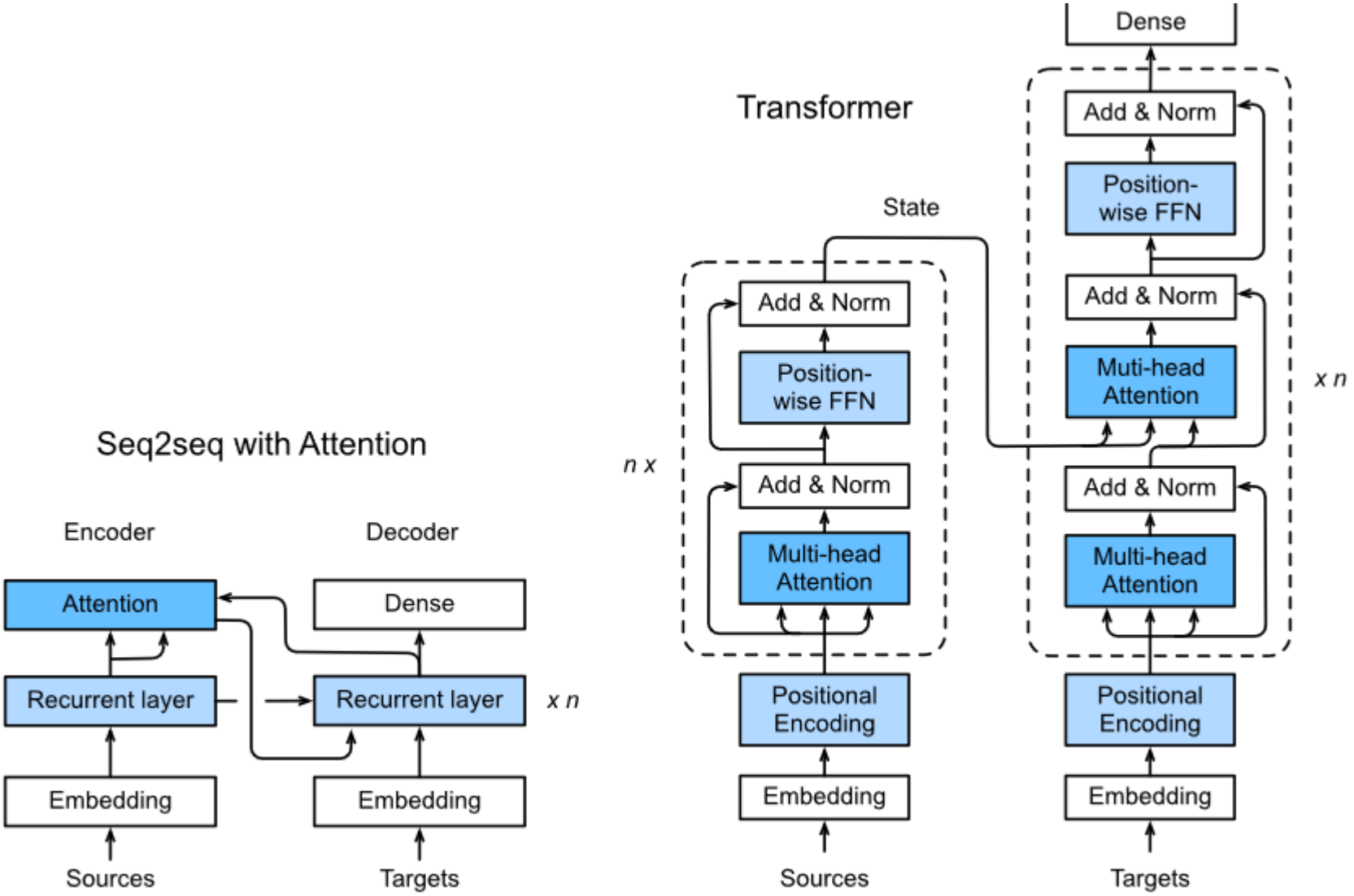
http://iscslp2018.org/images/T4_Towards%20end-to-end%20speech%20recognition.pdf

iptc
Information Processing and Telecommunications Center

Attention is all you need

Md Mehrab Tanjim

http://jalammar.github.io/illustrated-transformer/
https://nlp.stanford.edu/seminar/details/lkaiser.pdf



https://www.d2l.ai/_images/transformer.svg

References (I)

https://theneuralperspective.com/2016/10/04/05-recurrent-neural-networks-rnn-part-1-basic-rnn-char-rnn/

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-2-implementing-a-language-model-rnn-with-python-numpy-and-theano/

https://github.com/suriyadeepan/rnn-from-scratch

http://stats.stackexchange.com/questions/241985/understanding-lstm-units-vs-cells

http://monik.in/a-noobs-guide-to-implementing-rnn-lstm-using-tensorflow/

https://medium.com/@erikhallstrm/hello-world-rnn-83cd7105b767

http://suriyadeepan.github.io/2017-01-07-unfolding-rnn/

http://archive.eetindia.co.in/www.eetindia.co.in/VIDEO_DETAILS_700001601.HTM

References (II)

http://r2rt.com/styles-of-truncated-backpropagation.html

http://akkikiki.github.io/assets/LSTM+and+GRU.html

http://campuspress.yale.edu/yw355/deep_learning/

http://datascience.stackexchange.com/questions/12964/what-is-the-meaning-of-the-number-of-units-in-the-lstm-cell

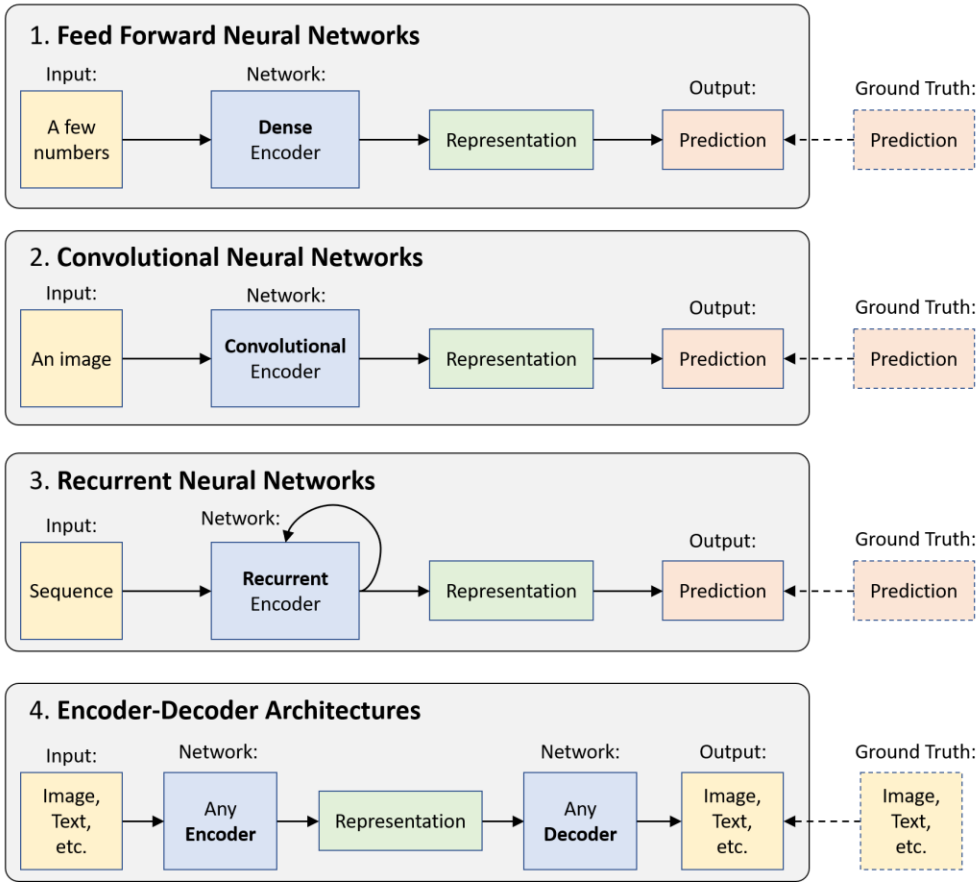
http://stackoverflow.com/questions/37901047/what-is-num-units-in-tensorflow-basiclstmcell

[Transformer] Attention Is All You Need | AISC Foundational

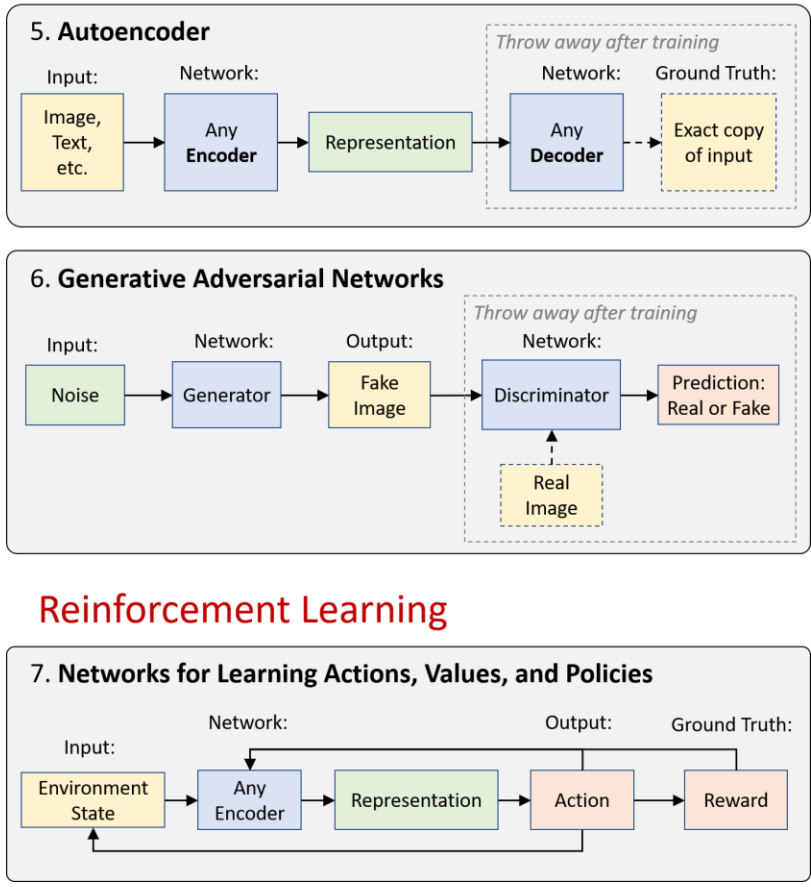
https://www.youtube.com/watch?v=S0KakHcj_rs

Deep learning basics

Supervised Learning



Unsupervised Learning



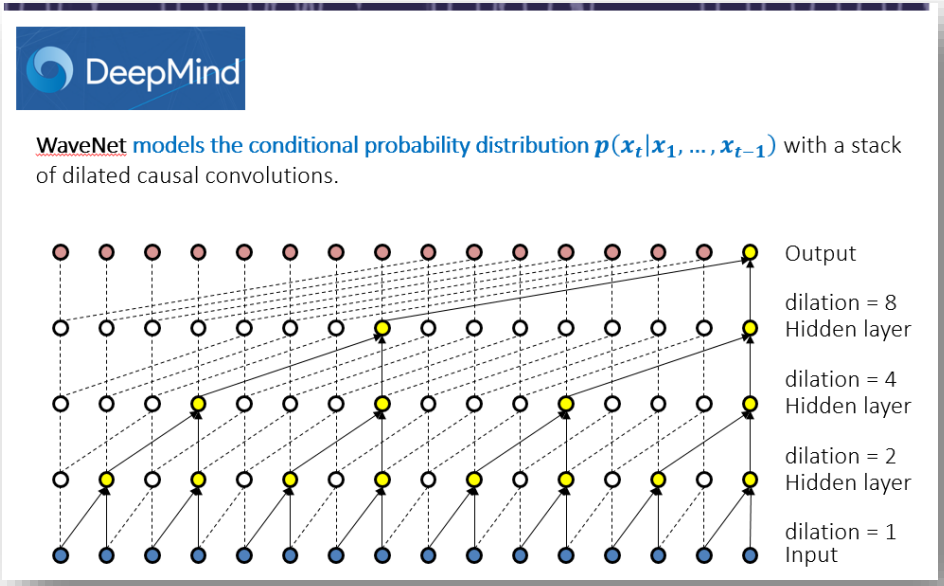
Reinforcement Learning

https://github.com/lexfridman/mit-deep-learning/blob/master/tutorial_deep_learning_basics/deep_learning_basics.ipynb

… there is much more to explore!

Remember that RNN are GENERATIVE models !!!

- Generative Adversarial Networks : GANs
- WaveNet …

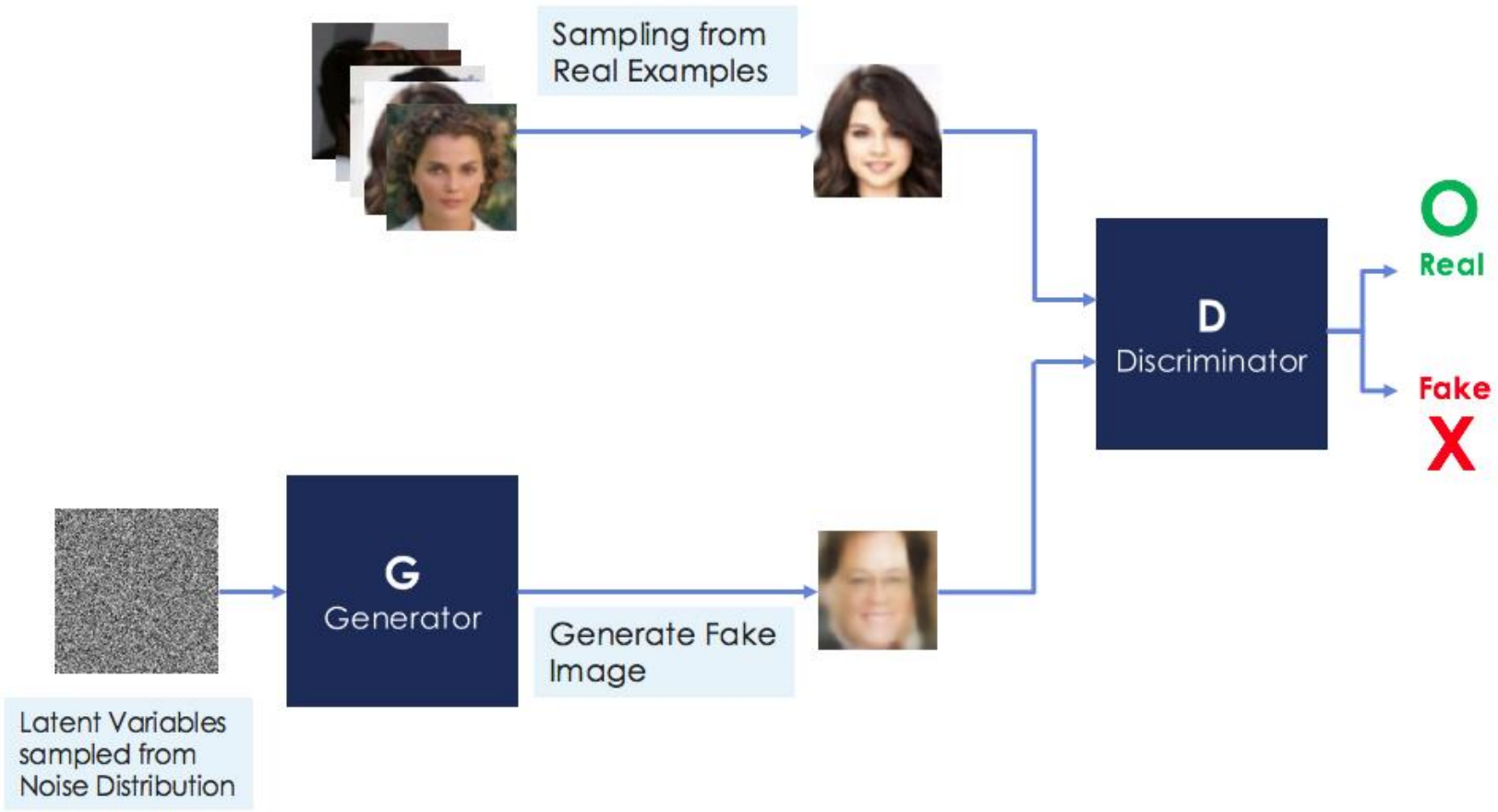


Generative Adversarial Networks GANs

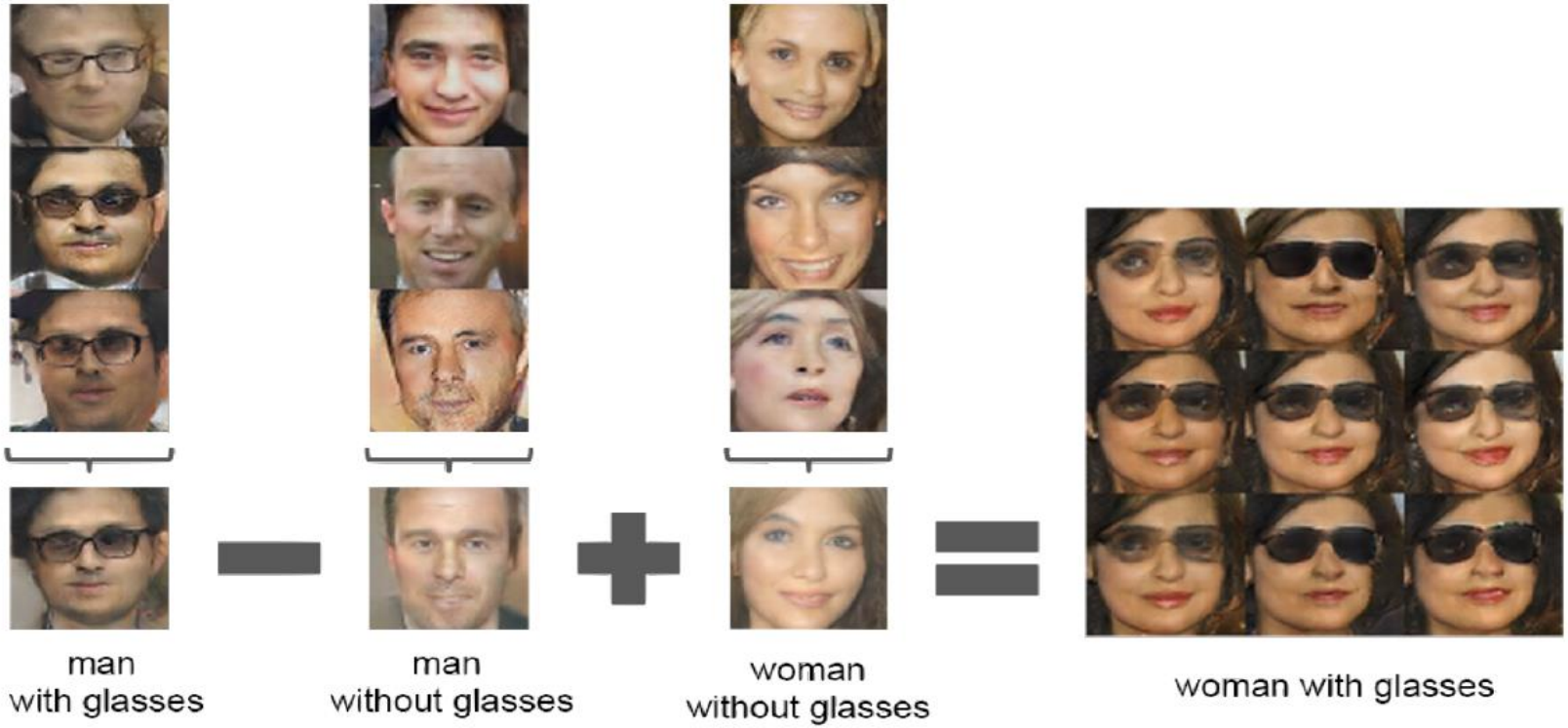
https://youtu.be/2edOMMREazo

https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2

Generative Adversarial Networks(GAN)



Latent vectors capture interesting patterns…



Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

https://youtu.be/2edOMMREazo

https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2

Text-to-Image Synthesis

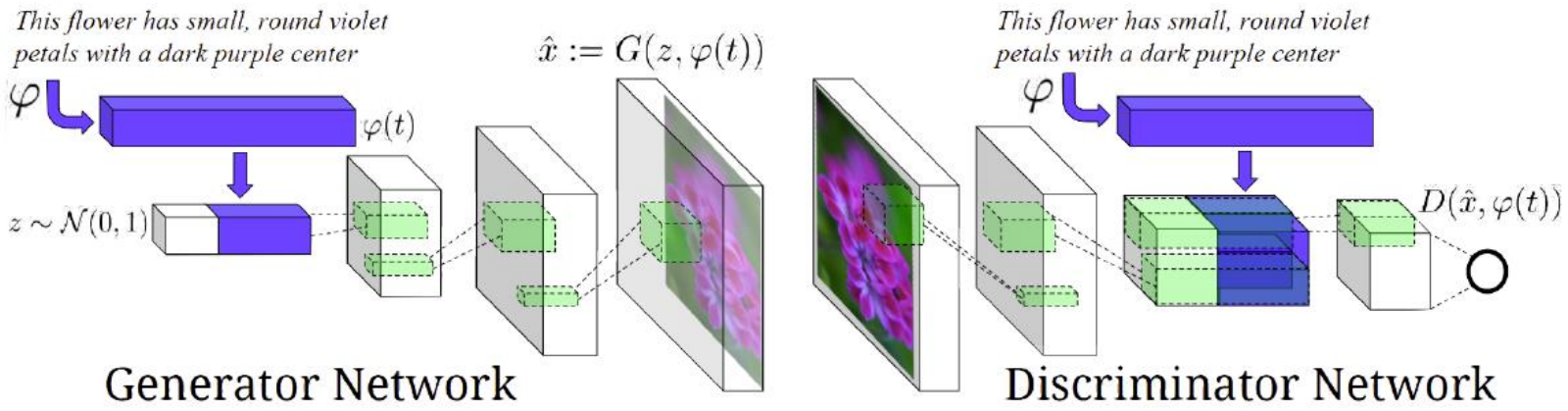


Figure 2 in the original paper.

Positive Example:
Real Image, Right Text

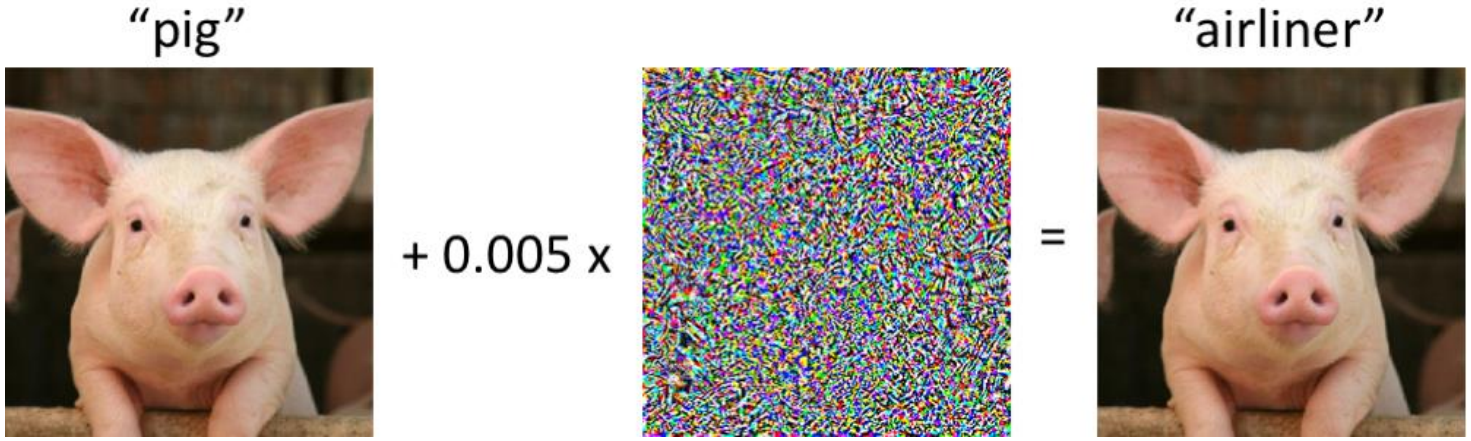
Negative Examples:
Real Image, Wrong Text
Fake Image, Right Text

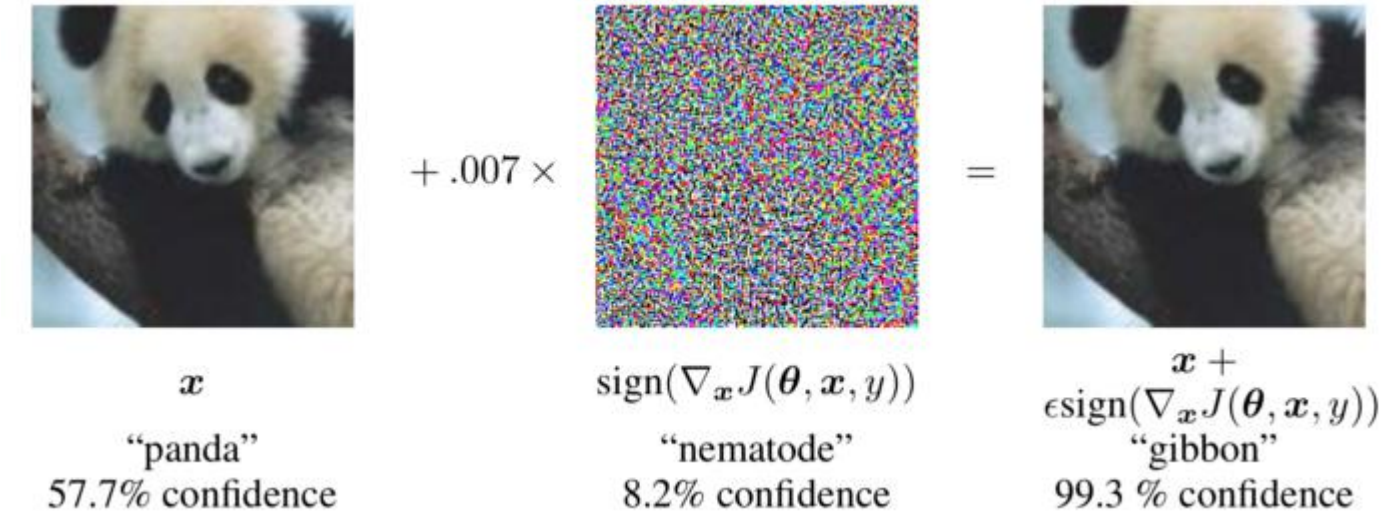
Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. "**Generative adversarial text to image synthesis**". ICML (2016).

- Image (Audio) Super Resolution
- Image-to-Image
- Visually-Aware Fashion Recommendation and DesignnWith GANs (Kang et al., 2017)
- Fraud / anomaly detection ,…

Some refs on GANs

- https://tryolabs.com/blog/2016/12/06/major-advancements-deep-learning-2016/
- https://blog.waya.ai/introduction-to-gans-a-boxing-match-b-w-neural-nets-b4e5319cc935#.6l7zh8u50
- https://en.wikipedia.org/wiki/Generative_adversarial_networks
- http://blog.aylien.com/introduction-generative-adversarial-networks-code-tensorflow/
- https://github.com/soumith/ganhacks





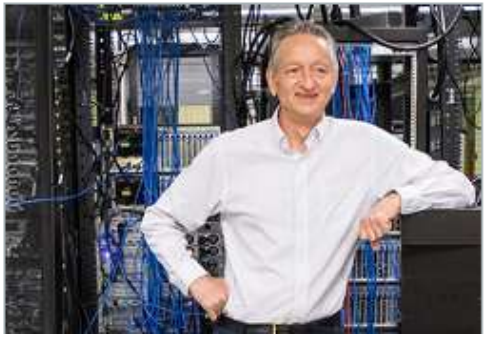
https://buzzrobot.com/4-ways-to-easily-fool-your-deep-neural-net-dca49463bd0

ACM NEWS

Artificial Intelligence Pioneer Says We Need to Start Over

By Axios
September 18, 2017
Comments

VIEW AS: SHARE:



In 1986, Geoffrey Hinton co-authored a paper that, three decades later, is central to the explosion of artificial intelligence (AI). But Hinton says his breakthrough method should be dispensed with, and a new path to AI found.

Speaking with Axios on the sidelines of an AI conference in Toronto on Wednesday, Hinton, a professor emeritus at the University of Toronto and a Google researcher, said he is now "deeply suspicious" of back-propagation, the workhorse method

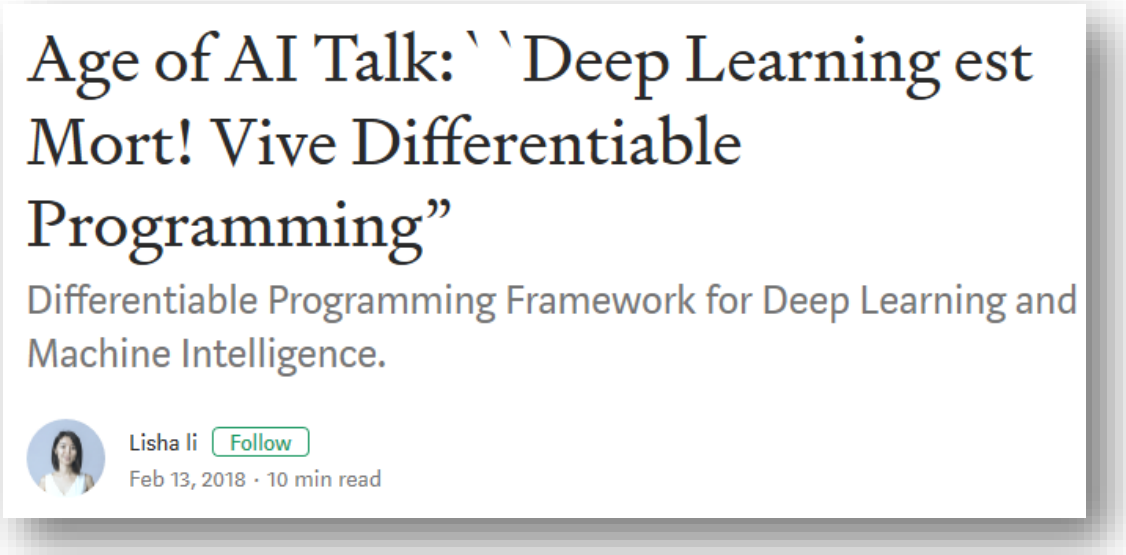
MORE NEWS & OPINIONS

Researchers Combine Technologies for New Digital

Deep Learning est mort. Vive Differentiable Programming!

Yann LeCun, Director of Facebook AI Research and the inventor of convolutional neural networks, took it.

https://www.facebook.com/yann.lecun/posts/10155003011462143



Age of AI Talk: ` `Deep Learning est Mort! Vive Differentiable Programming”

Differentiable Programming Framework for Deep Learning and Machine Intelligence.

Lisha li Follow

Feb 13, 2018 · 10 min read

https://techburst.io/deep-learning-est-mort-vive-differentiable-programming-5060d3c55074

https://medium.com/amplify-partners/age-of-ai-talk-deep-learning-est-morte-vive-differentiable-programming-6b1a1c9800d8

iptc Information Processing and Telecommunications Center

Demystifying Differentiable Programming: Shift/Reset the Penultimate Backpropagator

FEI WANG, Purdue University, USA
XILUN WU, Purdue University, USA
GREGORY ESSERTEL, Purdue University, USA
JAMES DECKER, Purdue University, USA
TIARK ROMPF, Purdue University, USA

Deep learning has seen tremendous success over the past decade in computer vision, machine translation, and gameplay. This success rests in crucial ways on *gradient-descent optimization* and the ability to "learn" parameters of a neural network by backpropagating observed errors. However, neural network architectures are growing increasingly sophisticated and diverse, which motivates an emerging quest for even more general forms of *differentiable programming*, where arbitrary parameterized computations can be trained by gradient descent. In this paper, we take a fresh look at automatic differentiation (AD) techniques, and especially aim to demystify the *reverse-mode* form of AD that generalizes backpropagation in neural networks.

https://arxiv.org/abs/1803.10228

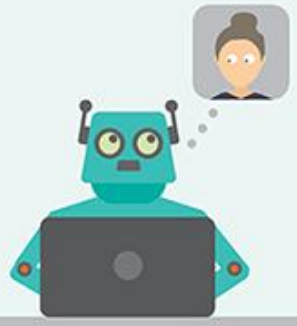
AGI: The forgotten science

Real AI – Human-level learning and understanding

Artificial General Intelligence (AGI)	Conventional AI
Focus on ***acquiring*** knowledge and skills	Focus on ***having*** knowledge and skills
Acquisition via learning	Acquisition via programming
General ability, using abstraction and context	Domain specific, rule-based and concrete
Ongoing cumulative, adaptive, grounded, self-directed learning	Relatively fixed abilities. Externally initiated improvements

Cognitive Computing vs. AI

COGNITIVE COMPUTING		ARTIFICIAL INTELLIGENCE
Machine learning, natural language processing, neural networks, deep learning, sentiment analysis	TECHNOLOGIES	Machine learning, natural language processing, neural networks, deep learning
Simulate human thought processes to assist humans in finding solutions to complex problems	CAPABILITIES	Find patterns in big data to learn and either reveal hidden information or deliver solutions to complex problems
Augment human capabilities	PURPOSE	Automate processes
Customer service, healthcare, industrial sector	INDUSTRIES	Finance, security, healthcare, retail, manufacturing, government



… and more pragmatic ML things…

Towards Data Science Follow

HOME DATA SCIENCE MACHINE LEARNING PROGRAMMING VISUALIZATION AI JOURNALISM PICKS | CONTRIBUTE

AutoKeras: The Killer of Google's AutoML

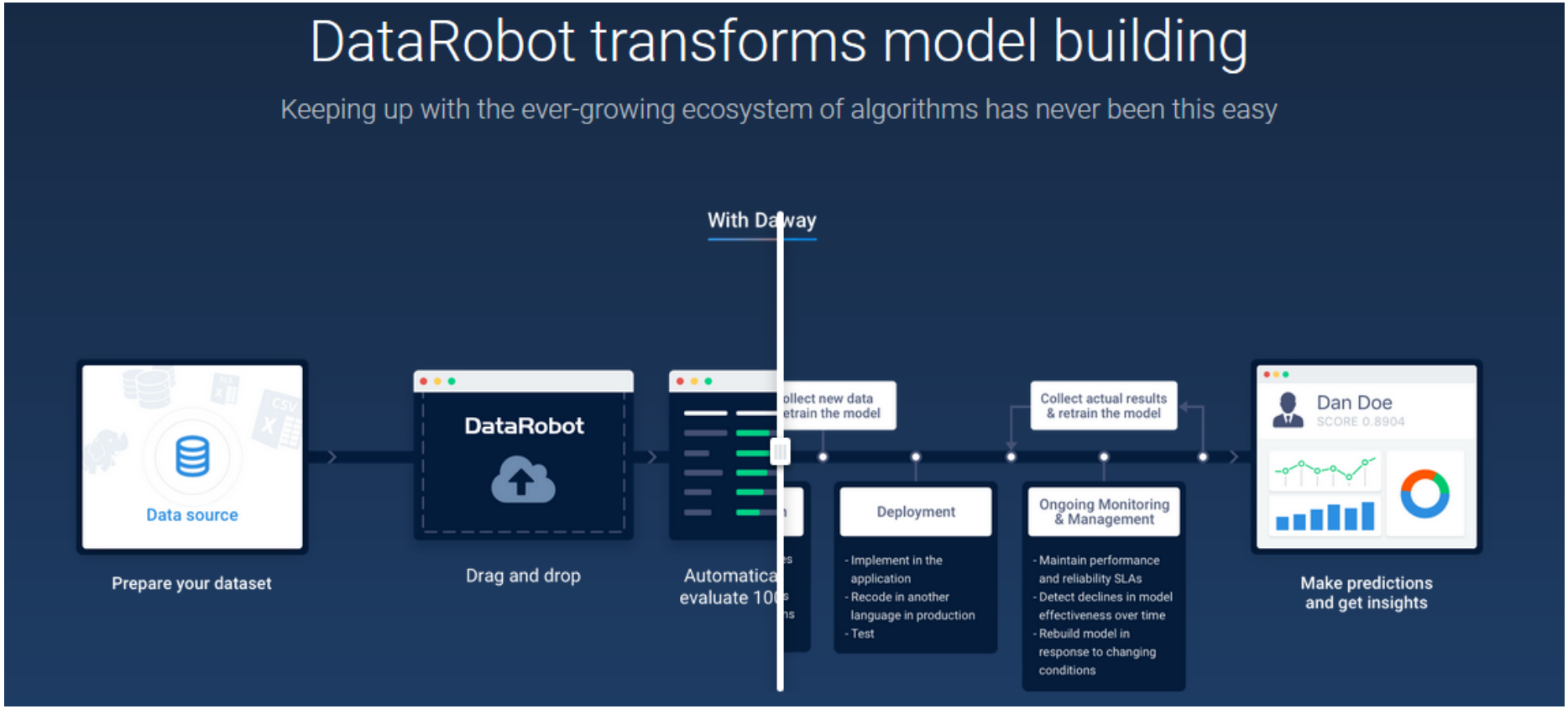
George Seif Follow

Jul 31, 2018 · 4 min read

AUTO KERAS

https://towardsdatascience.com/autokeras-the-killer-of-googles-automl-9e84c552a319

https://www.datarobot.com/



Ethics?

Engineers at Amazon, Google, and Microsoft are demanding the companies put ethics before profits

Would You Quit a Tech Project Over Ethical Concerns?

By KATHY PRETZ 2 November 2018

http://theinstitute.ieee.org/ieee-roundup/career-and-education/career-guidance/would-you-quit-a-tech-project-over-ethical-concerns

Why Schools Are Getting More Serious About Teaching Engineering Students About Ethics?

Harvard, MIT, and Stanford say it's more important than ever to recognize the ramifications of AI technology

By KATHY PRETZ 5 September 2018

http://theinstitute.ieee.org/ieee-roundup/blogs/blog/why-schools-are-getting-more-serious-about-teaching-engineering-students-about-ethics

Engineers at Amazon, Google, and Microsoft are demanding the companies put ethics before profits

Would You Quit a Tech Project Over Ethical Concerns?

By KATHY PRETZ 2 November 2018

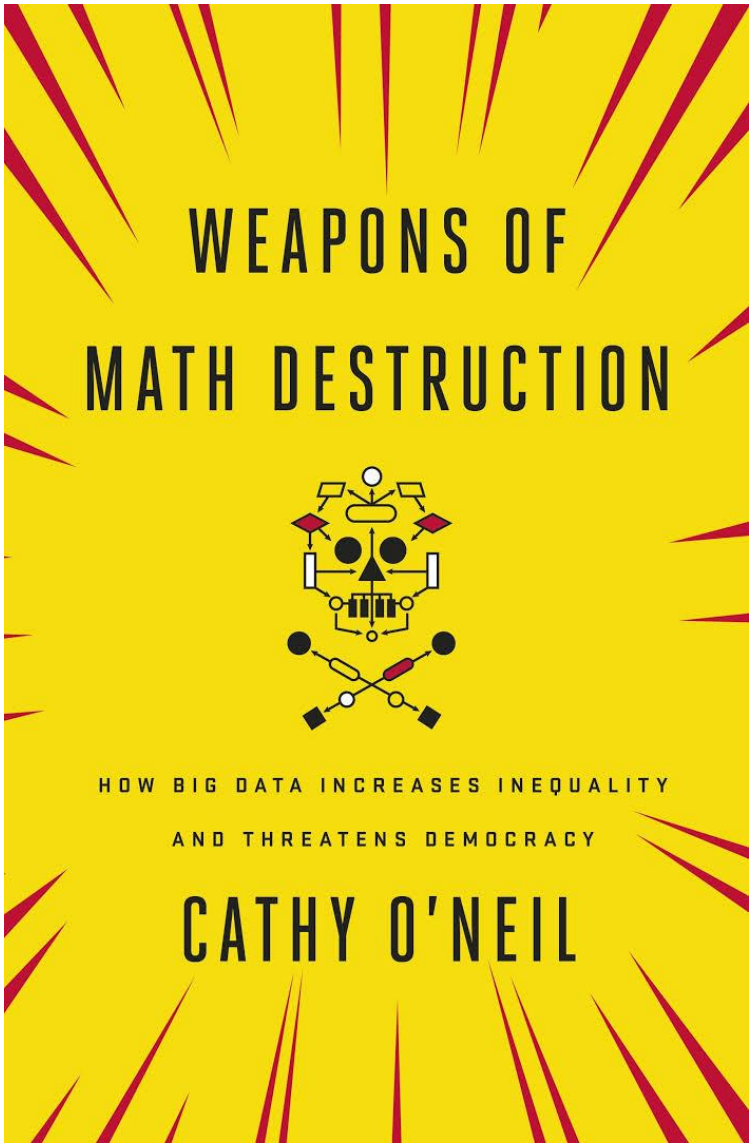
http://theinstitute.ieee.org/ieee-roundup/career-and-education/career-guidance/would-you-quit-a-tech-project-over-ethical-concerns

- **Ethics for Machines :** *what type of machines is acceptable for us to make (privacy, use of and relationship with technology,…).*

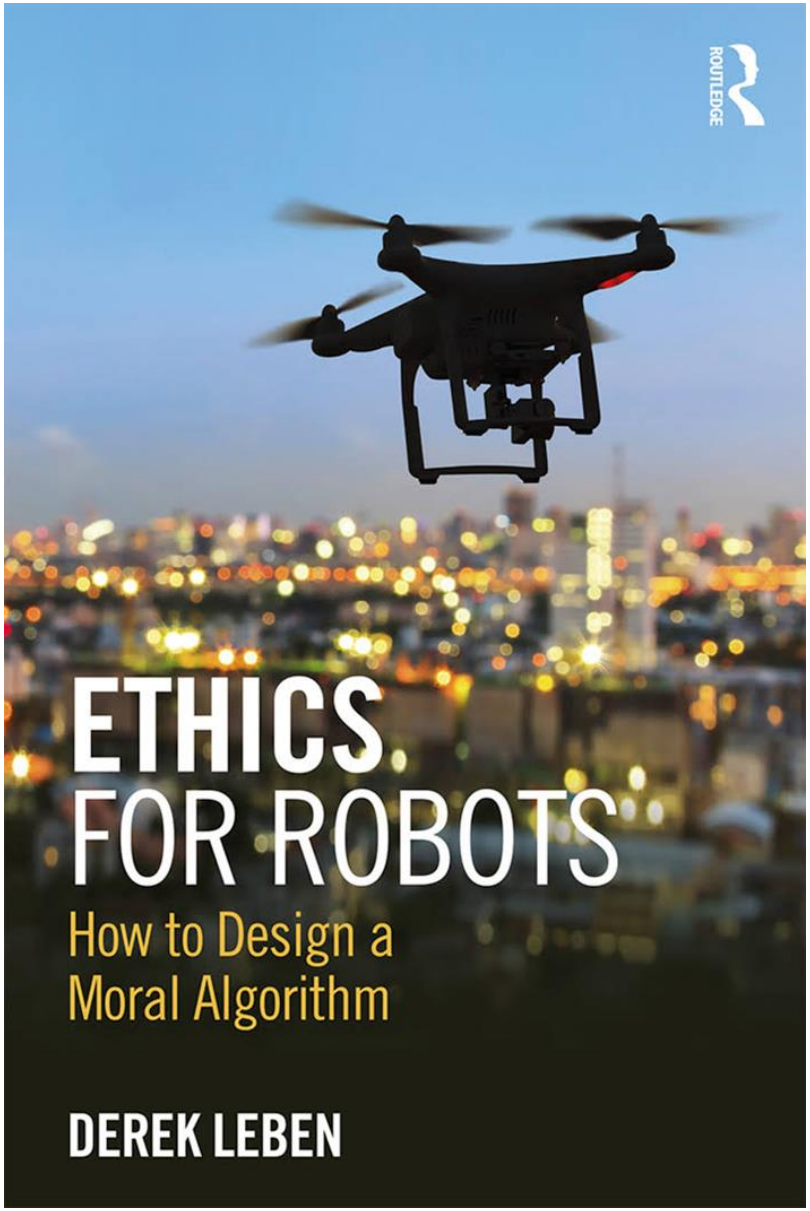
- **Ethics of Machines :** *how we want machines to be behaving when they are taking decisions autonomously.*

Machine ethics podcast episode with Derek Leben
Associate Professor of Philosophy at the University of Pittsburgh, Johnstown
https://www.youtube.com/watch?v=OjLA6lZEPW4

Ethics for Machines



Ethics of Machines



….or just BE GOOD!!!

That's all !

I hope this seminar will be useful to you

…we keep waiting for your amazing deep-applications..

THANK YOU!

luisalfonso.hernandez@upm.es