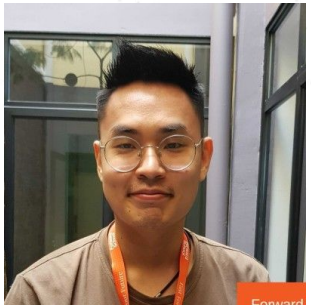
Image Captioning

Discover How AI Creates Text from Images

Boon Khai Yeoh
Junior AI Engineer

Boon Khai is a Deep Learning practitioner, who often takes part in the activities of developing the Artificial Intelligence courses in Skymind. He applies his engineering and computer science knowledge to leverage emerging technologies of Machine Learning to run performance tests for different software environments. He also works closely with Skymind Trainers to deliver knowledge and inspire future generation of practitioners.



Boon Khai Yeoh
Junior AI Engineer
CertifAI Sdn. Bhd.
www.linkedin.com/in/boon-khai-yeoh/

EDUCATION

Bachelor of Engineering (Mechatronics) with Honours
Universiti Tunku Abdul Rahman

What is Image Captioning?

- Image Captioning is the process of generating textual description of an image.
- It uses both Natural Language Processing and Computer Vision to generate captions.
- It is a multi-model technique involving both convolution and recurrent neural networks.

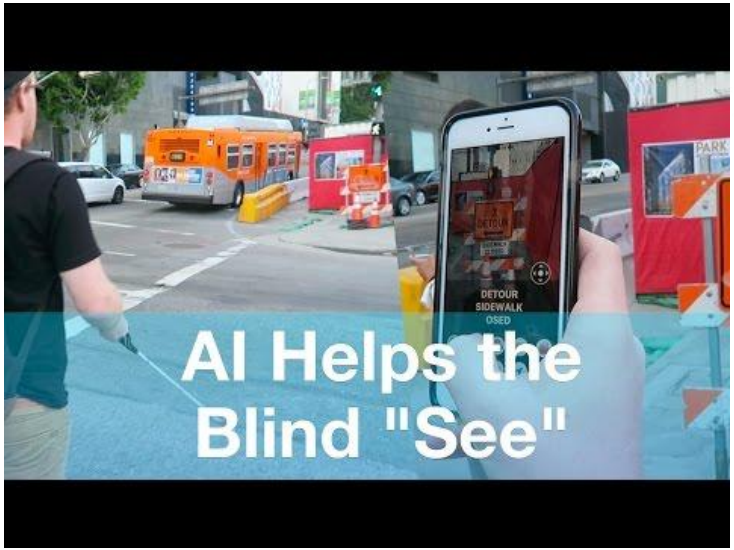
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

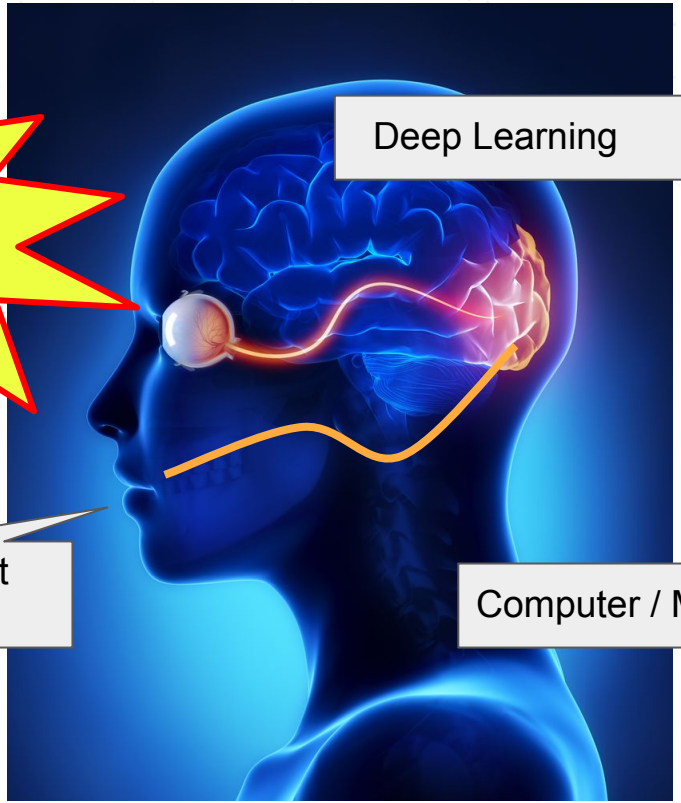
What are the possible applications?

- The main application of image captioning is automating the job of someone who interprets the images
- Assistance for the visually impaired



Source: https://www.youtube.com/watch?time_continue=1&v=FZBGjxQeP-A&feature=emb_logo

What will you will learn today?



Deep Learning

CNN - “eye” model

RNN - “language” model

Man in black shirt is playing guitar

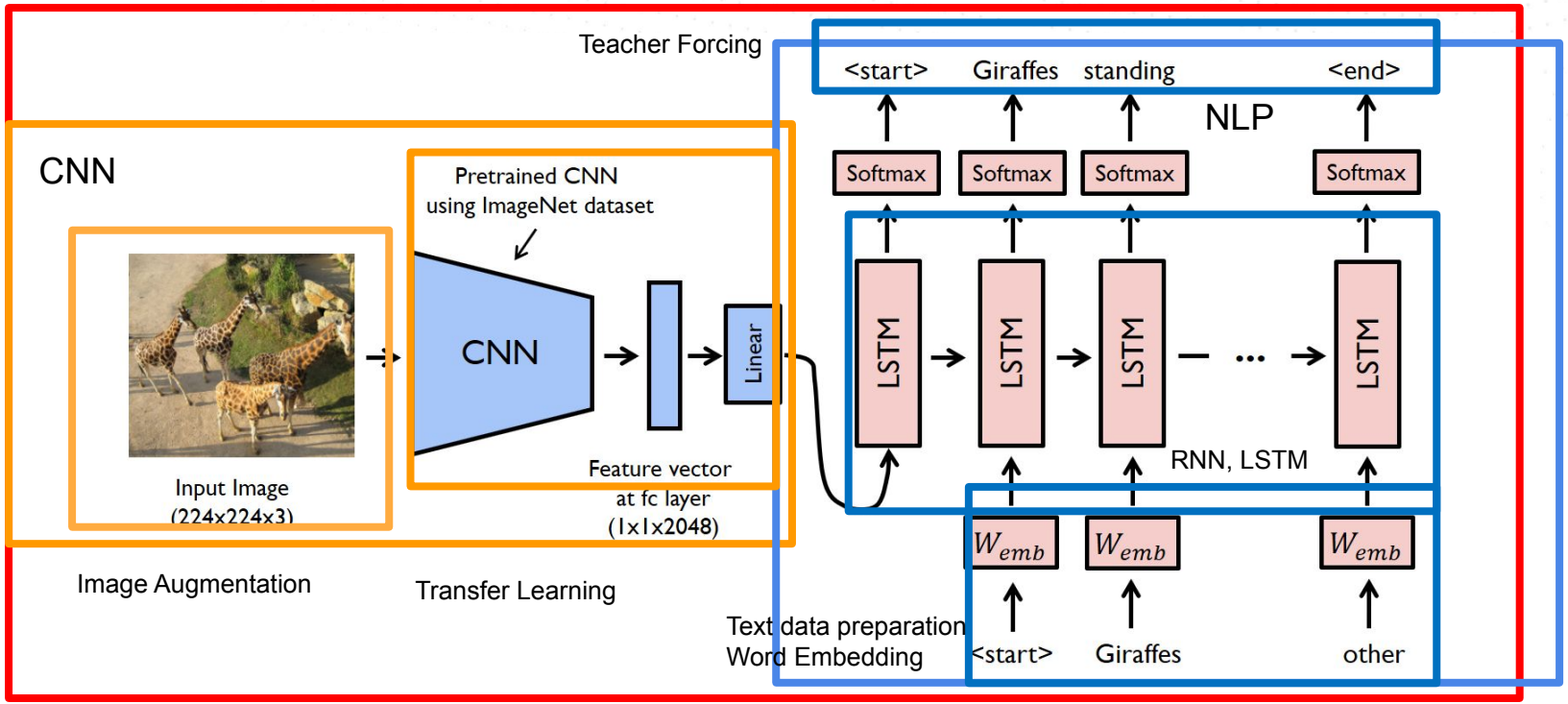
Computer / Machine

CERTIFAI
powered by Skymind

Captioning Model

- Captioning model relies on 2 main components, CNN and RNN.
    - CNN works by recognizing object in images.
    - RNN works with sequence data such as sequence of words.
- So by merging the two, you will get a model that can find patterns and images which will then generate captions related to the images.

What will you will learn today? Seq2Seq

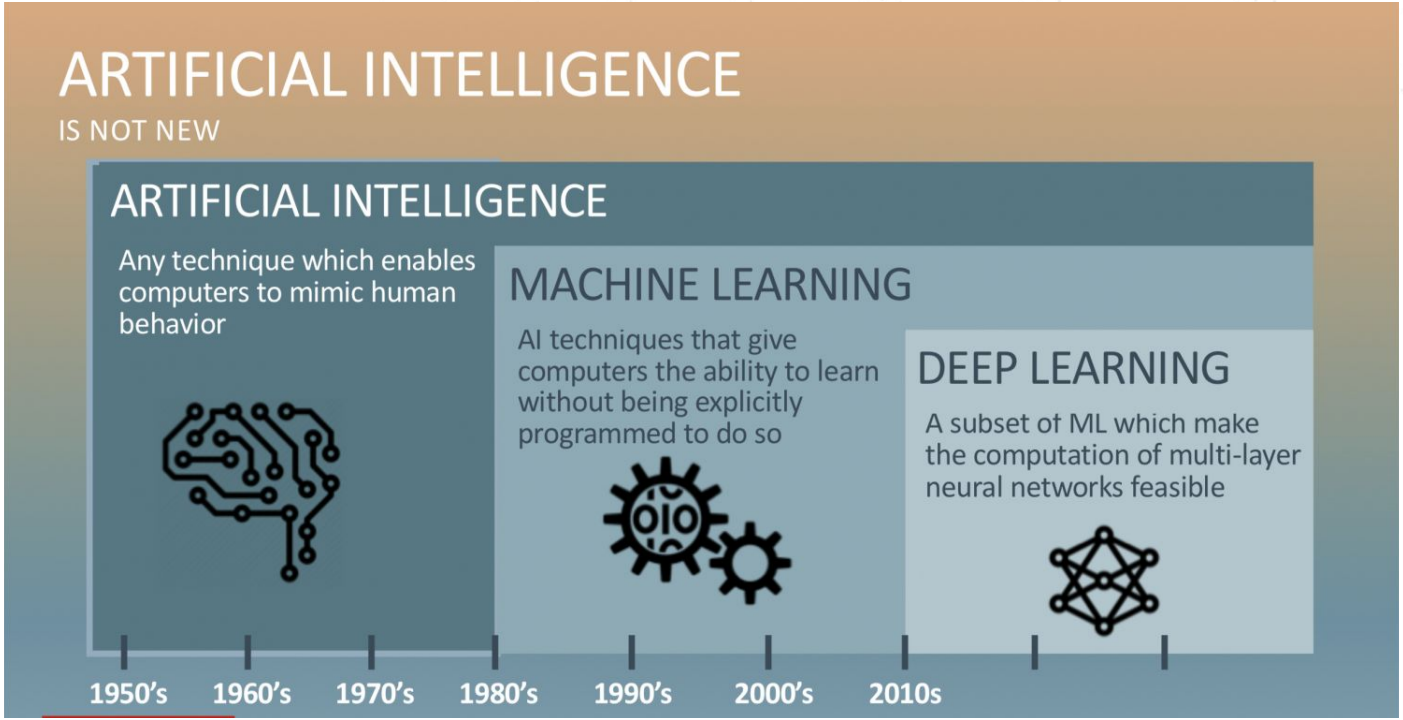


CERTIFAI
powered by Skymind

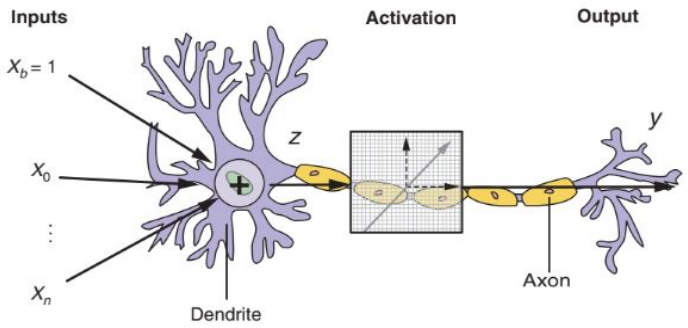
What will you will learn today?

- ANN
- Computer Vision
  - CNN
  - Image Augmentation
- NLP
  - RNN
  - LSTM
  - Text Data Preparation
- Bag of Tricks
  - Transfer learning
  - Teacher Forcing

What is Deep Learning?

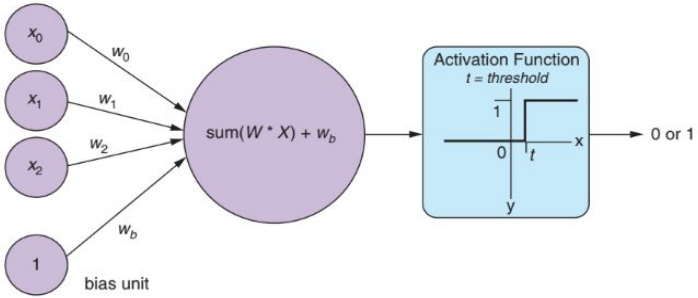


What are Neural Networks?



Neuron Cell

- Electrical signals flow through the dendrites to the nucleus.
- When cell reaches a certain level of electric charge, it discharges or **fires.**
- Electric signal is then passed through the axon.



Artificial Neuron

Based of a neuron cell:

- Input are being multiplied by **weights**.
- If the sum are above the **threshold** the output will be 1, or otherwise 0.

What are Neural Networks?

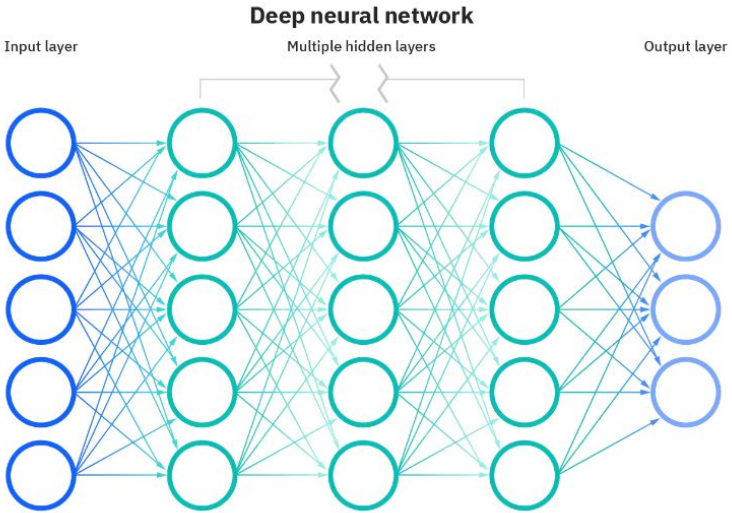
Also known as **artificial neural networks (ANN)**, are a subset of machine learning and the heart of deep learning.

Structure of Neural Network

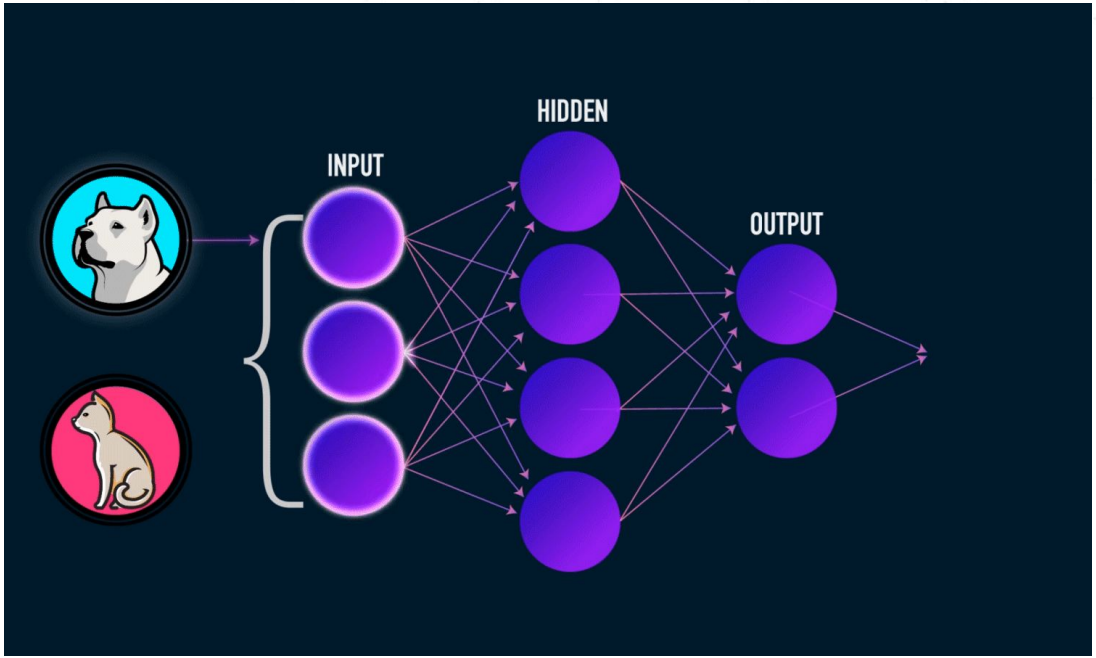
- One input layer
- One or more hidden layers
- One output layer

Each layer is comprised of **many Artificial Neurons**.

Each artificial neuron is connected and has its own associated **weight** and **activation functions**.



Neural Network in Action

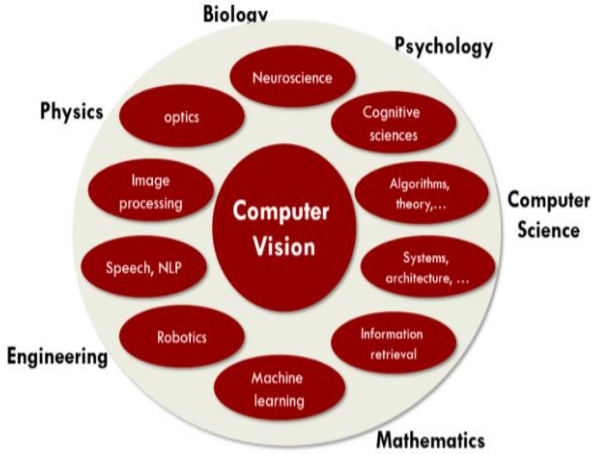


Computer Vision

What is Computer Vision?

"Computer vision is an interdisciplinary scientific field that deals with how computers can be made to **gain high-level understanding from digital images or videos**.

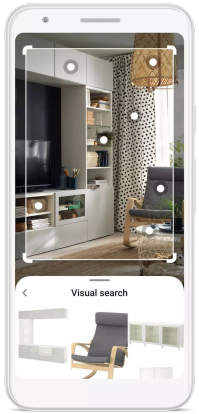
From the perspective of engineering, it seeks to **automate tasks that the human visual system can do**." (Wikipedia)

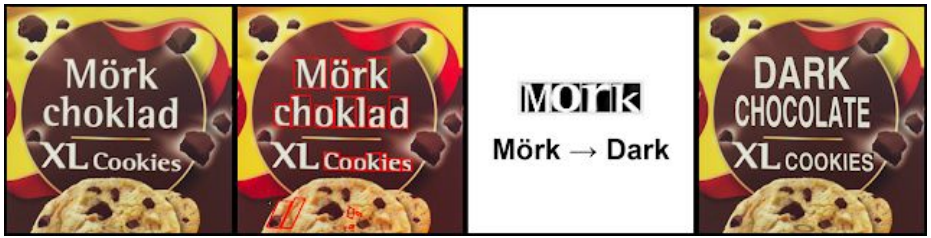


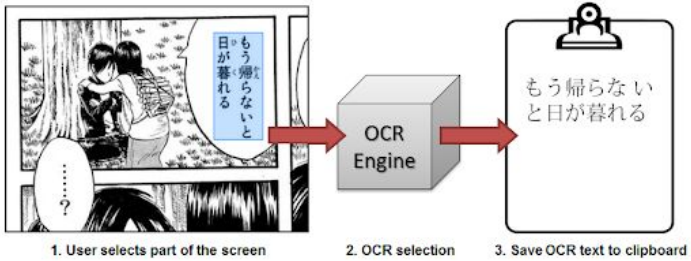
Computer Vision Tasks

- Image/Object recognition
- Object detection

- Image retrieval
- Pose estimation
- Optical Character Recognition (OCR)
- Facial recognition







CERTIFAI
powered by Skymind

Computer Vision Tasks - Recognition

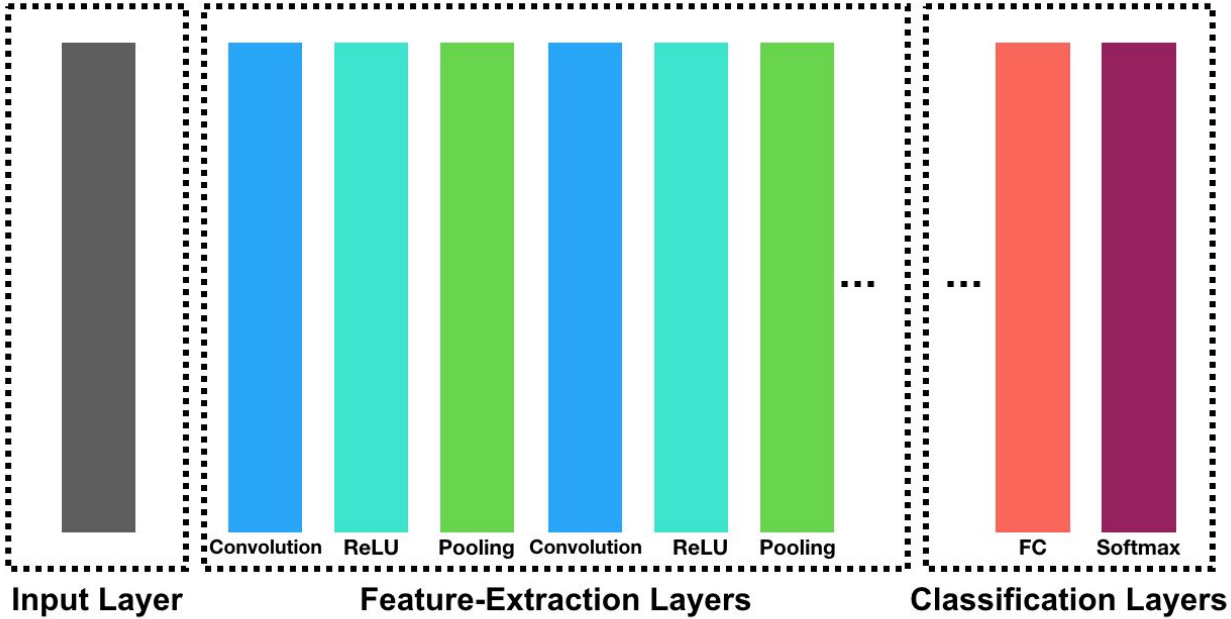
Sources: https://www.youtube.com/watch?time_continue=4&v=VOC3huqHrss&feature=emb_logo

Convolutional Neural Networks

CNN Architecture Overview

- Input layer
- Feature extraction (learning) layers
- Classification Layer

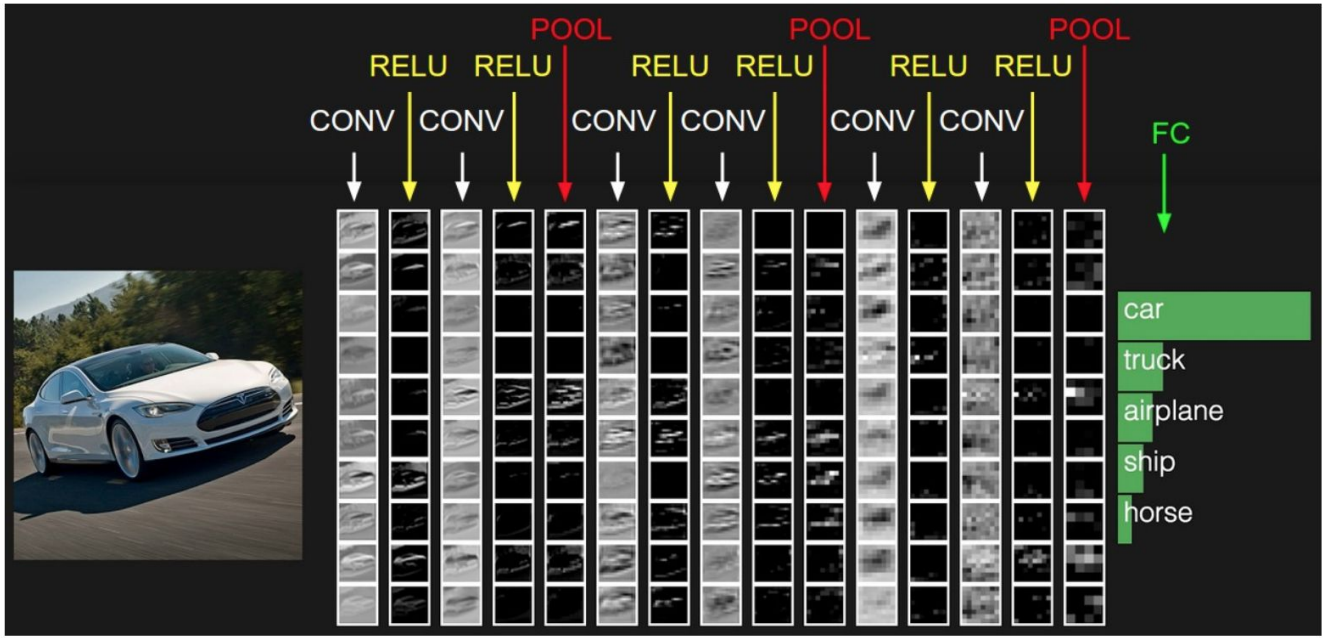
There are many variations of CNN architecture



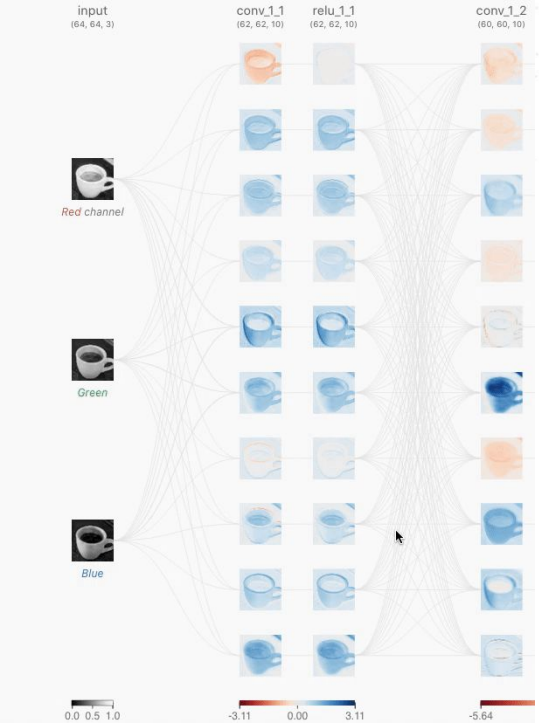
CERTIFAI
powered by Skymind

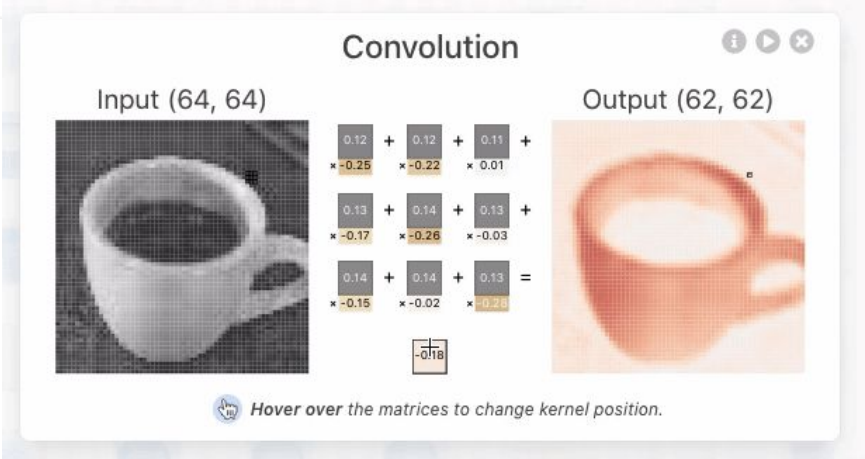
Convolutional Neural Network

An example architecture of ConvNet for Cifar-10 Classification



Convolutional Neural Network





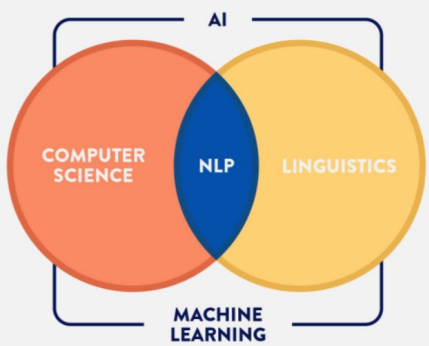
Simulation:
https://poloclub.github.io/cnn-explainer/

CERTIFAI
powered by Skymind

NLP

What is NLP?

- **Natural language processing (NLP)** is a branch of computer science, and more specifically, a branch of artificial intelligence (AI), **concerned with giving computers the ability to understand text and spoken words in much the same way that humans can.**
- NLP uses algorithms to:
  - Identify and extract natural language rules
  - Converting unstructured language data into a format that computers can understand.
  - Extract meaning associated with every sentence
  - Collect essential information from languages data
- The ultimate goal of NLP is to read, interpret, understand, and generate human languages in a valuable way.

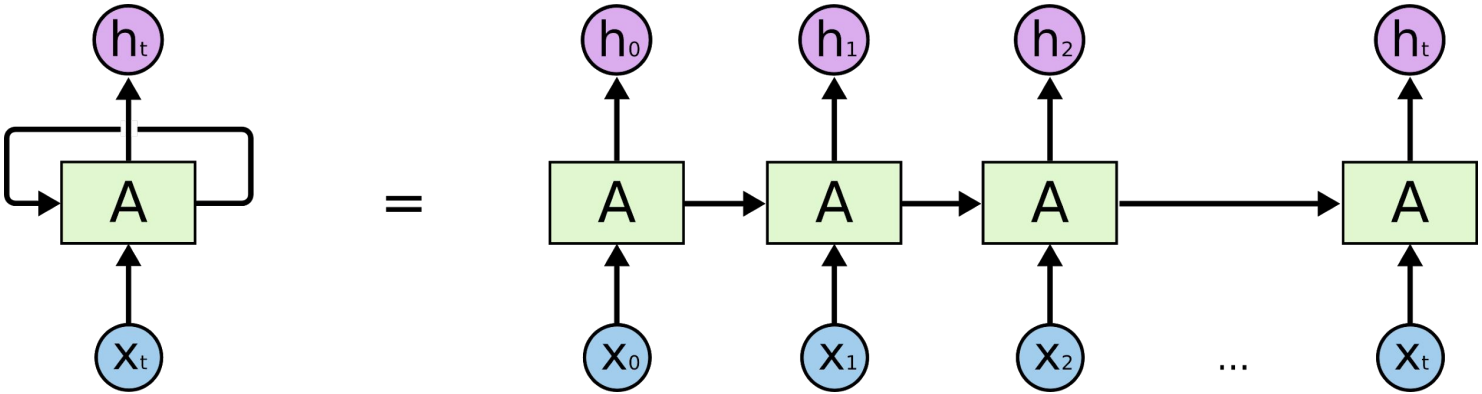


CERTIFAI
powered by Skymind

Recurrent Neural Network

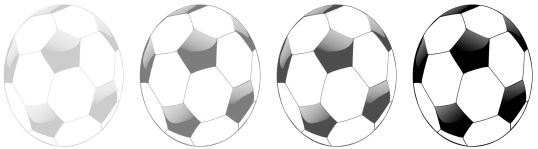
- RNN exhibits temporal dynamic behavior, which allows for information persistence in between layers.

- RNN includes a feedback loop that it uses to learn from sequences.
- Output at each time step is based on
  - current input, and
  - input at all previous time steps



Why Recurrent Neural Networks?

The best thing we can do is just guess if it's going to move left or right



We can probably see that it is moving towards to right based on the previous locations of the ball

Why Recurrent Neural Networks?

- A sequence modelling problem

"This morning I took the cat for a walk"

Given these words *predict what comes next*

" 1 2 3 4 5 6 7 8 9"

Given these numbers *predict what comes next*

Example

Source: https://www.youtube.com/watch?v=8BFzu9m52sc&t=2s

Example

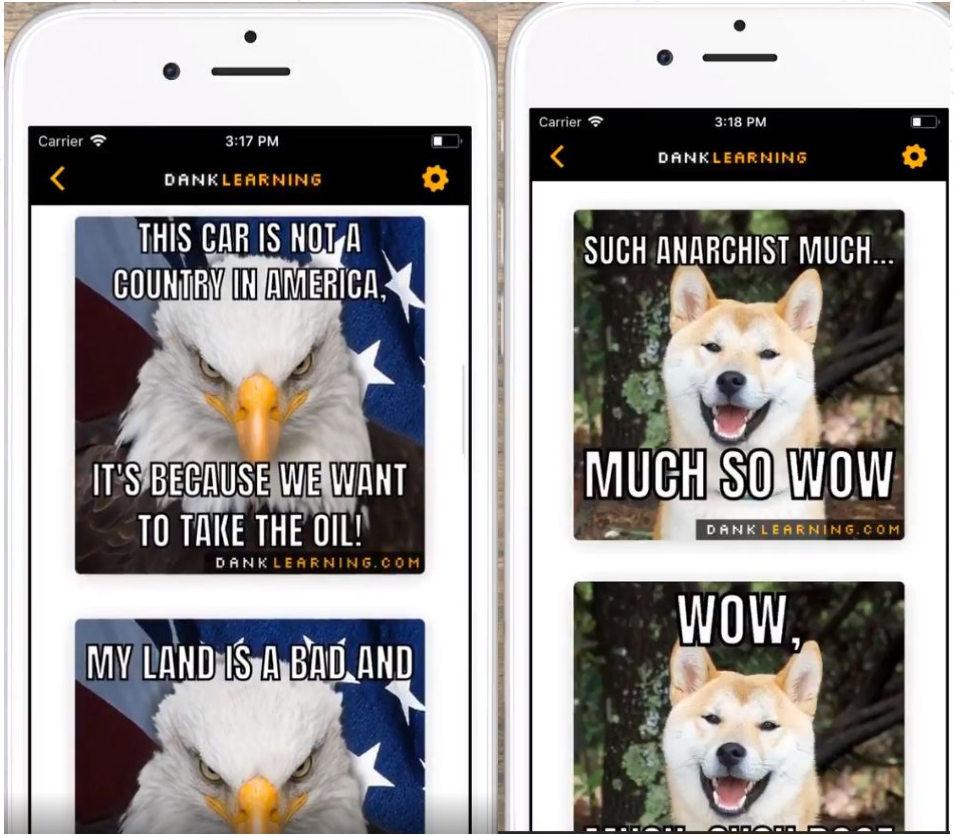
Dank Learning: Generating Memes Using Deep Neural Networks

Abel L. Peirson V
Department of Physics
Stanford University
alpv95@stanford.edu

E. Meltem Tolunay
Department of Electrical Engineering
Stanford University
meltem.tolunay@stanford.edu

Abstract

We introduce a novel meme generation system, which given any image can produce a humorous and relevant caption. Furthermore, the system can be conditioned on not only an image but also a user-defined label relating to the meme template, giving a handle to the user on meme content. The system uses a pre-trained Inception-v3 network to return an image embedding which is passed to



LSTM

Motivation of Long Short Term Memory (LSTM)

- LSTM is introduced to handle the vanishing or exploding gradient problem by introducing some gate units within.

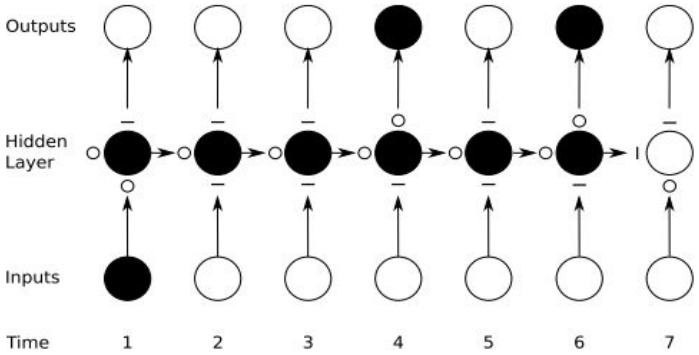
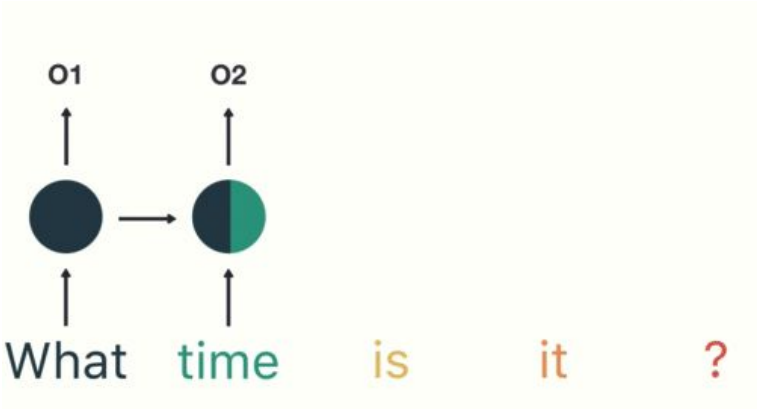


Figure 4.4: **Preservation of gradient information by LSTM.** As in Figure 4.1 the shading of the nodes indicates their sensitivity to the inputs at time one; in this case the black nodes are maximally sensitive and the white nodes are entirely insensitive. The state of the input, forget, and output gates are displayed below, to the left and above the hidden layer respectively. For simplicity, all gates are either entirely open ('O') or closed ('—'). The memory cell 'remembers' the first input as long as the forget gate is open and the input gate is closed. The sensitivity of the output layer can be switched on and off by the output gate without affecting the cell.

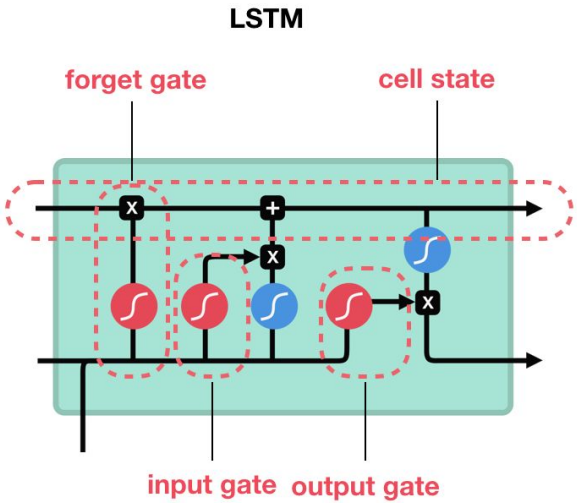


LSTM

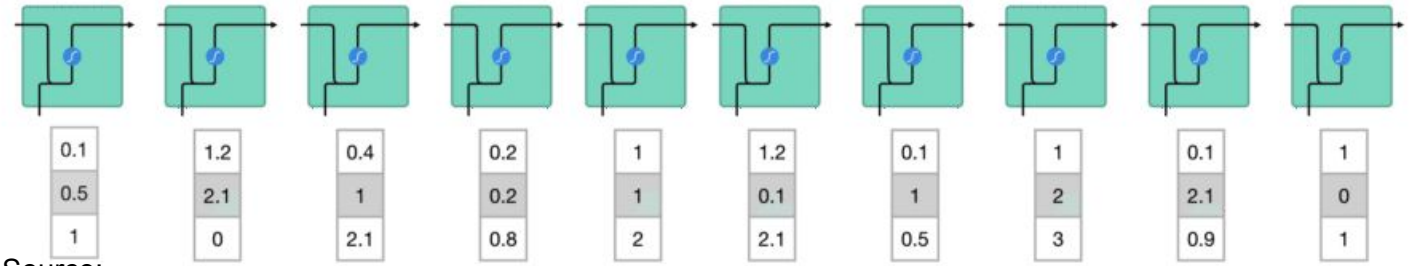
In order to add a new information, **RNN** transforms the existing information **completely by applying a function**.

LSTM networks can mitigate this:

- Make small modifications to the information by multiplications and additions.
- The information flows through a mechanism known as cell states.
- Selectively remember or forget things.



LSTM



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

CERTIFAI
powered by Skymind

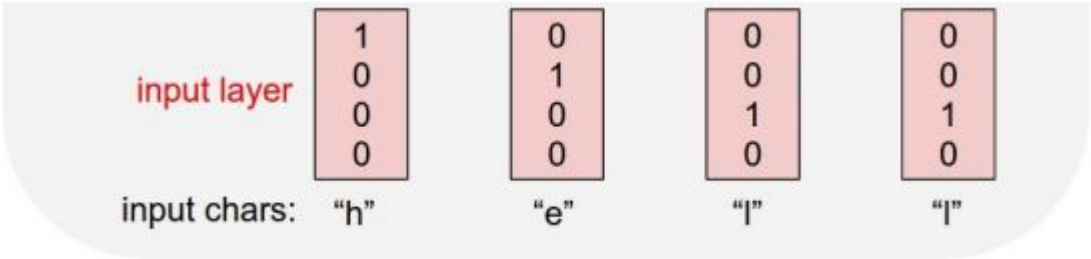
Example

Character Language Modelling

Character-level language model

Training

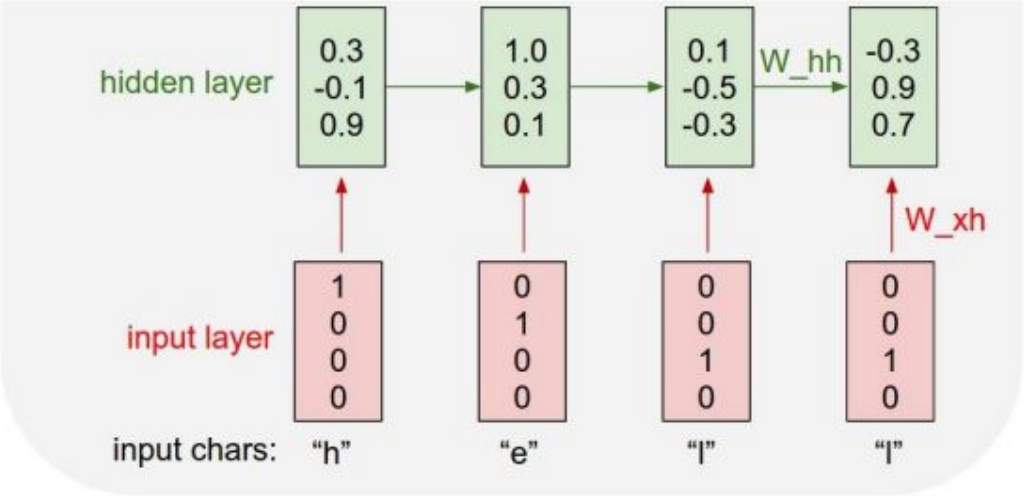
- Train our RNN to generate words, character-by-character
- Vocabulary: [h, e, l, o]
- Example training sequence: “hello”
- The word “hello” is encoded into 5 binary digits



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

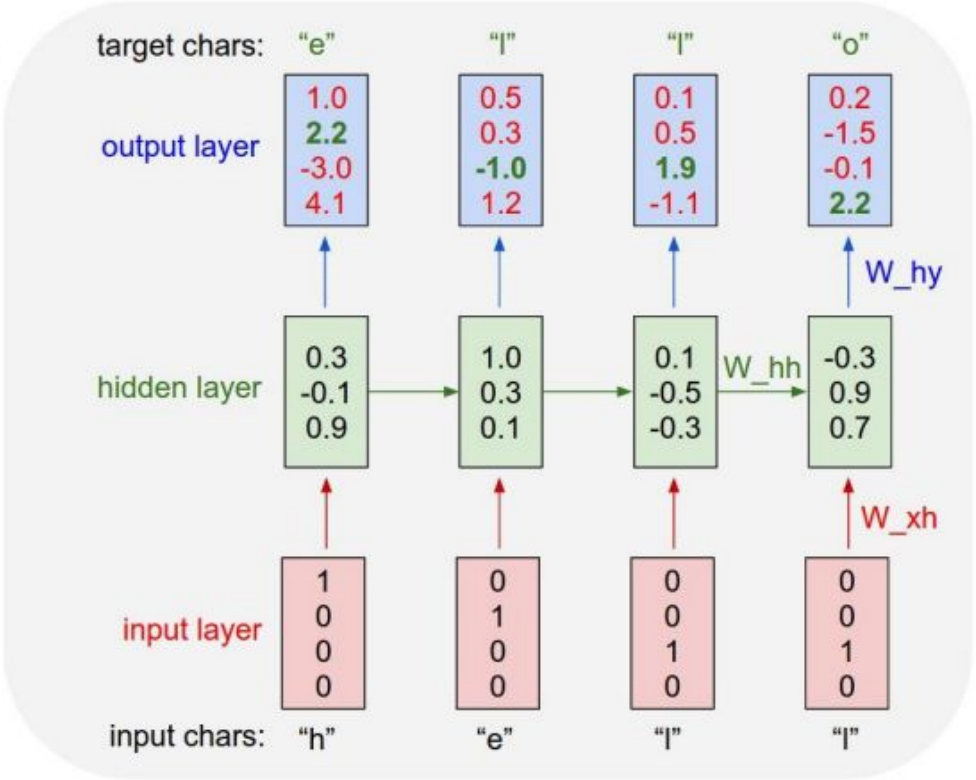
Character-level language model

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Character-level language model

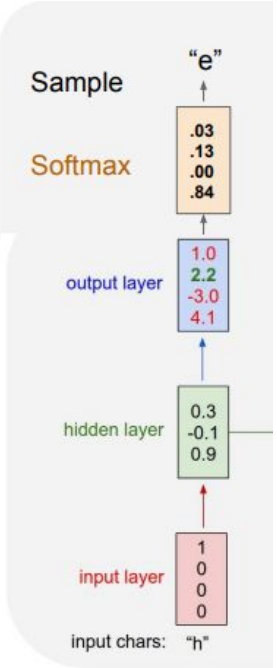


Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Character-level language model

Test / Character sampling

- Let our model synthesise a new word based on our trained model

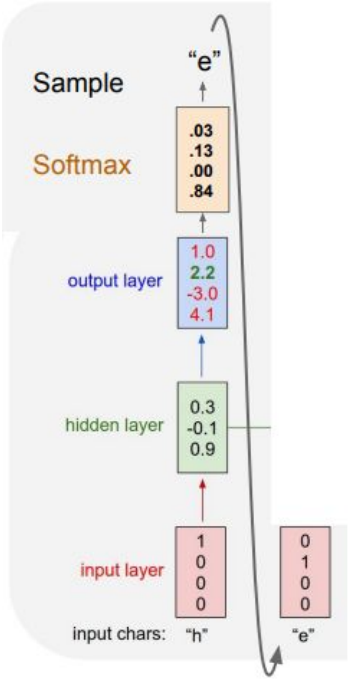


Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Character-level language model

Test / Character sampling

- Let our model synthesise a new word based on our trained model

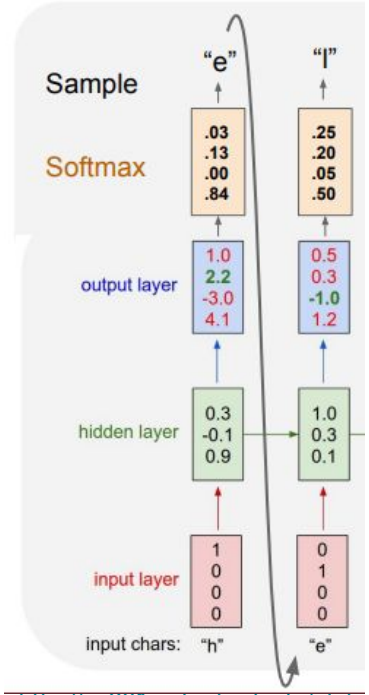


Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Character-level language model

Test / Character sampling

- Let our model synthesise a new word based on our trained model

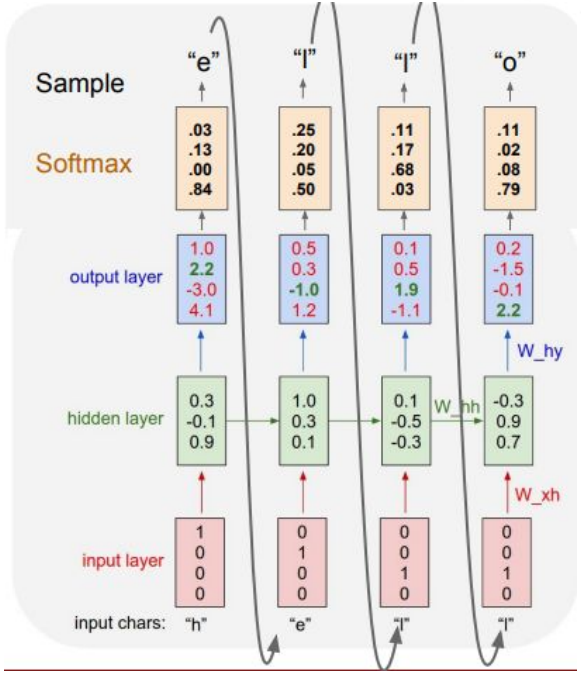


Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Character-level language model

Test / Character sampling

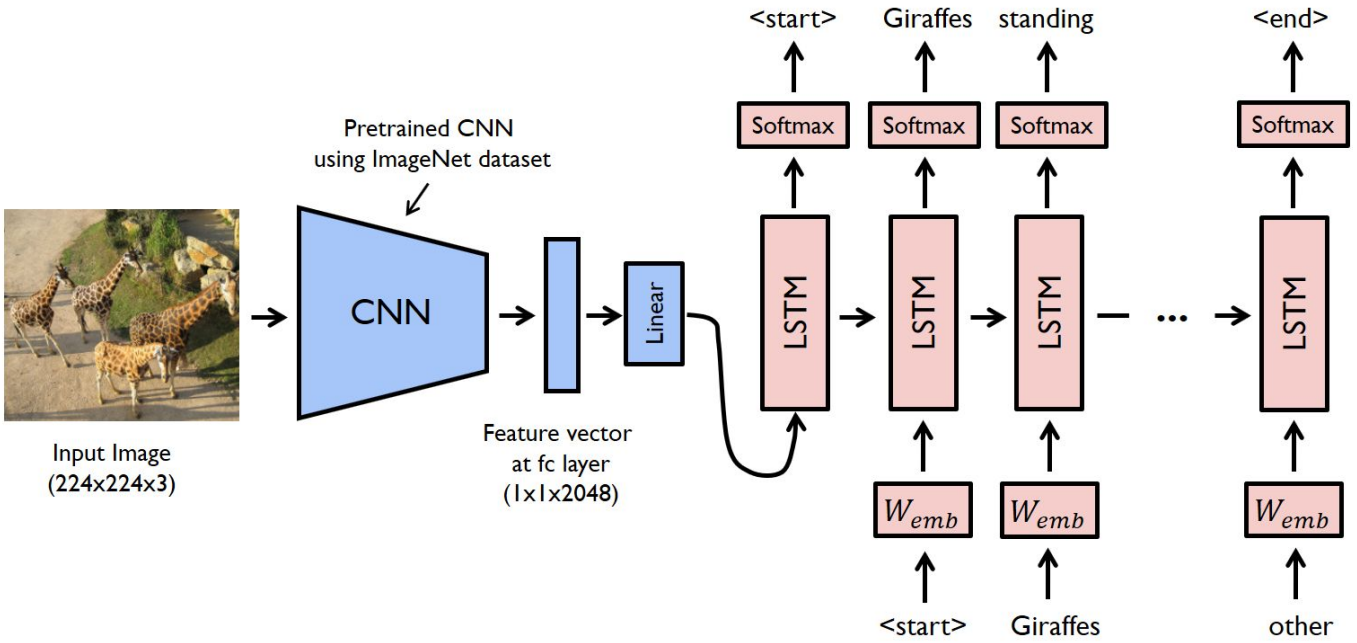
- Let our model synthesise a new word based on our trained model



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

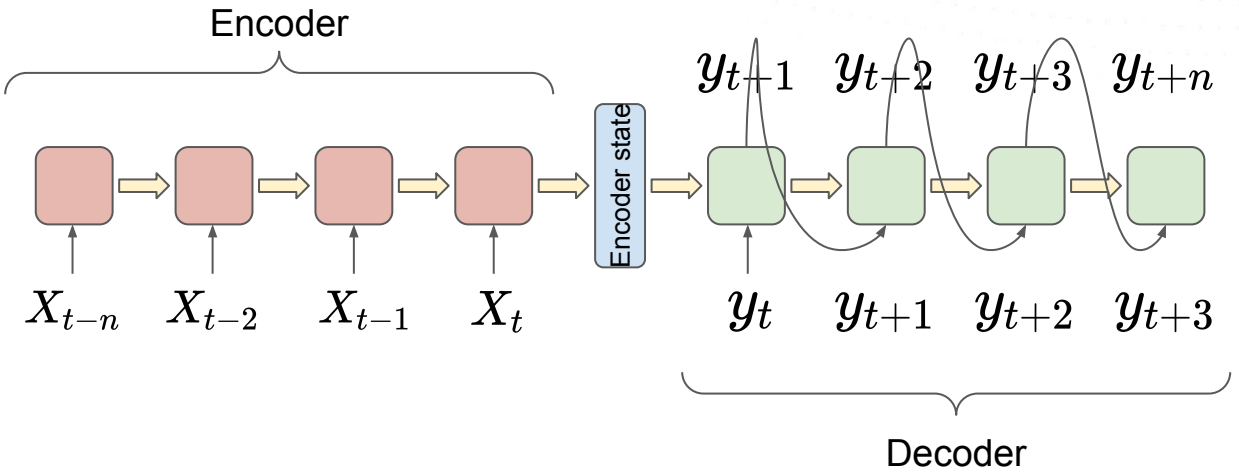
Seq2Seq

It comes to this…….



How does the Seq2seq model work?

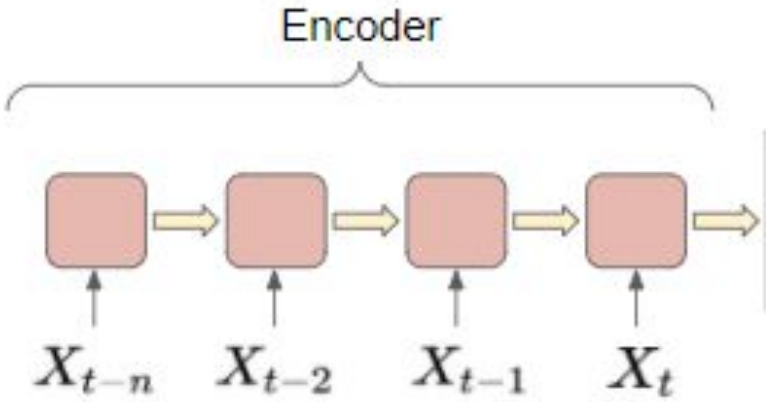
- The Seq2seq model architecture can be shown in the illustration below.
- The model consists of 3 parts : encoder, intermediate (encoder) vector and decoder



How does the Seq2seq model work?

Encoder

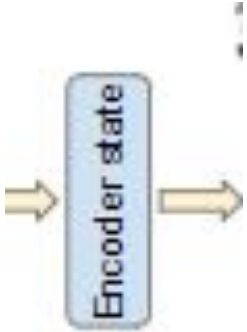
- A stack of several recurrent units (it can be RNN, LSTM or GRU) where each unit accepts a single element of the input sequence
- All the outputs will be discarded and only the internal states (hidden state) are preserved.
- The hidden states will be passed to the decoder



How does the Seq2seq model work?

Encoder Vector

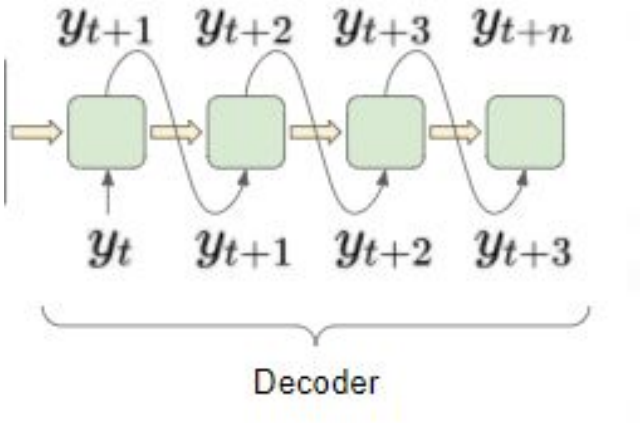
- This is the final hidden state produced from the model's encoder part.
- It acts as the initial hidden state of the decoder part of the model
- In practice, this does not exist and all we need to do is to input our hidden state from encoder to decoder



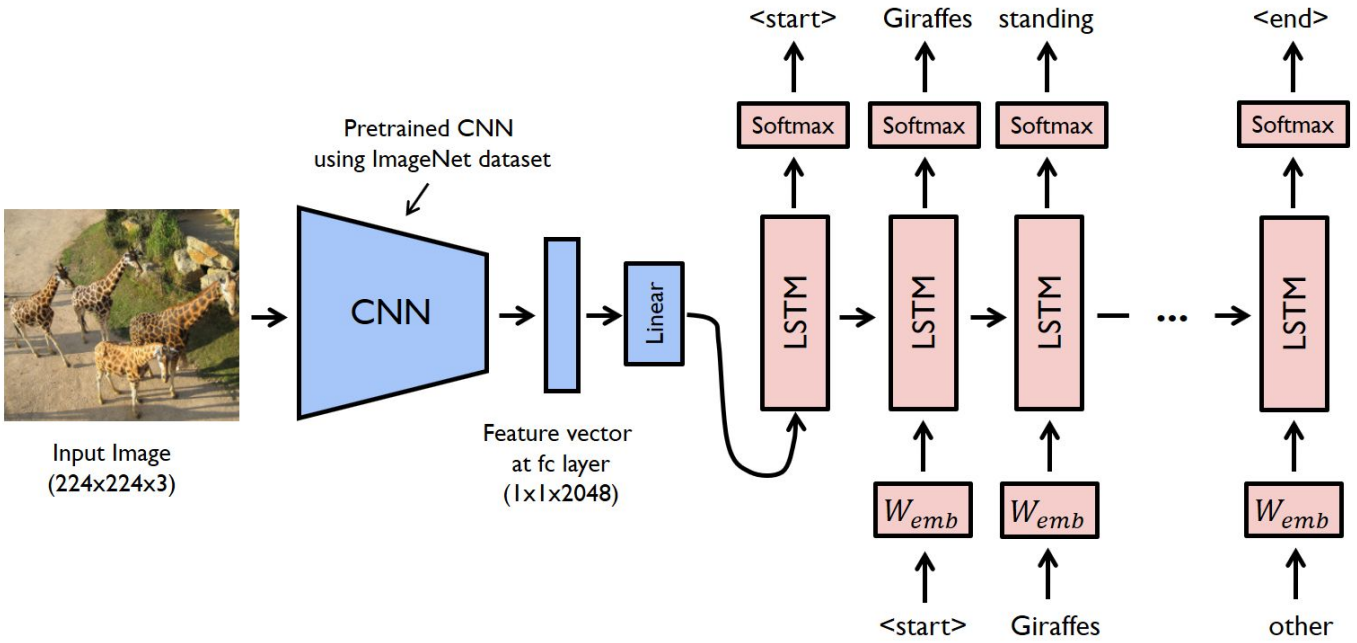
How does the Seq2seq model work?

Decoder

- Each recurrent unit accepts a hidden state from previous unit and output
- The initial states are initialised from the final states of Encoder



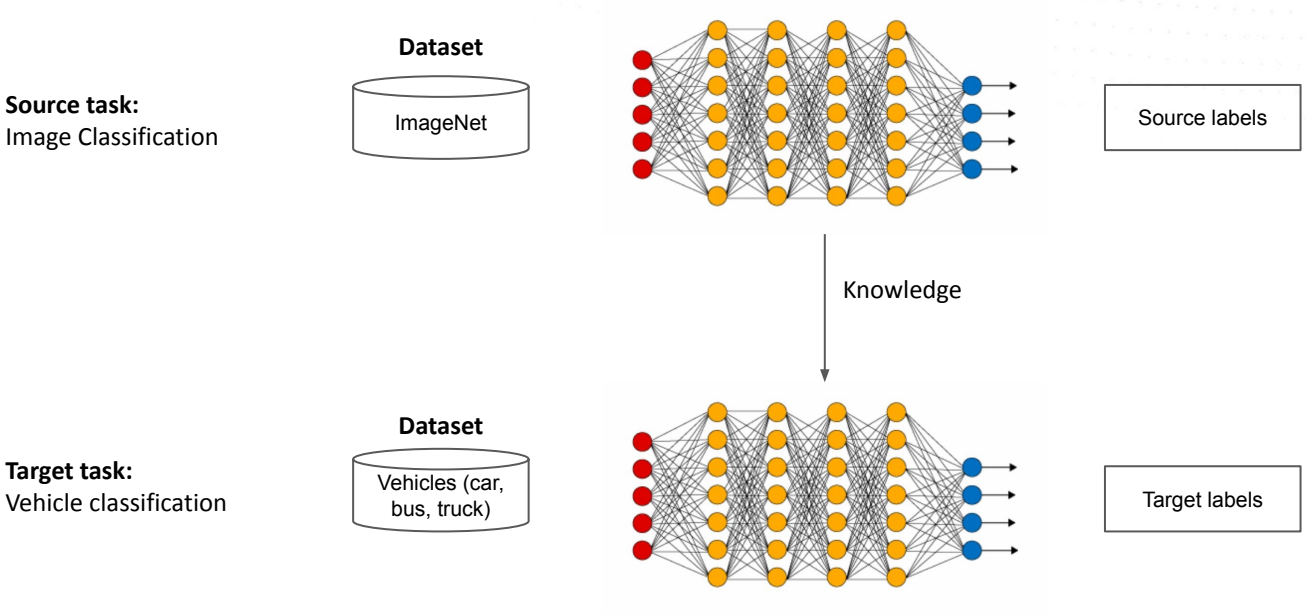
It comes to this…….



Bag of tricks

Transfer Learning

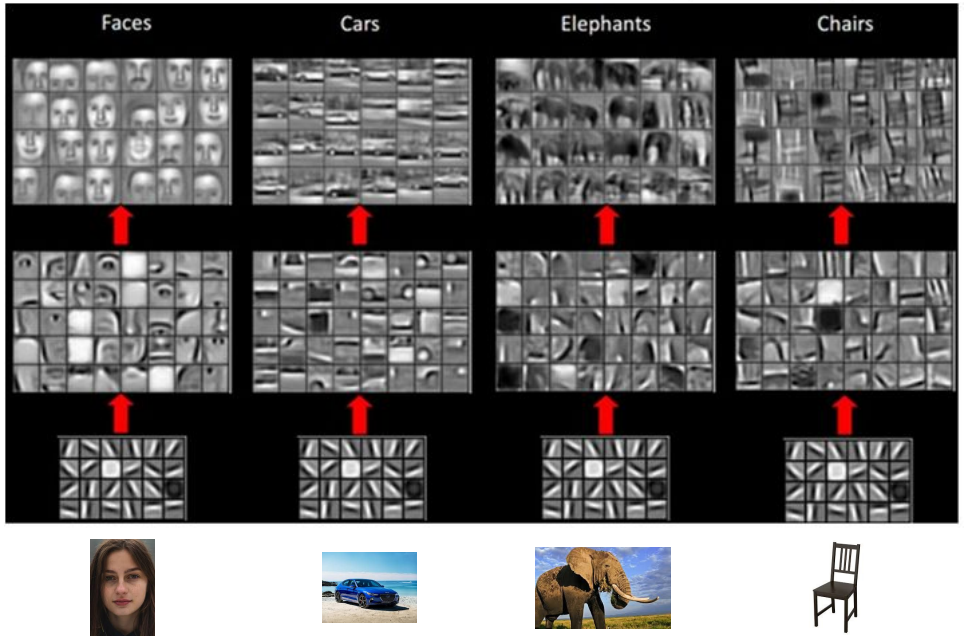
The most popular approach in deep learning is that it takes the network parameters trained on different source tasks and use it on related target tasks.



Convolutional Neural Network

CNN progressively construct higher-order features as layers deepens

- **Low level features**: Fundamental blocks of image (edges, lines, curves)
- **High level features:** Combination of low level features for more complicated representations



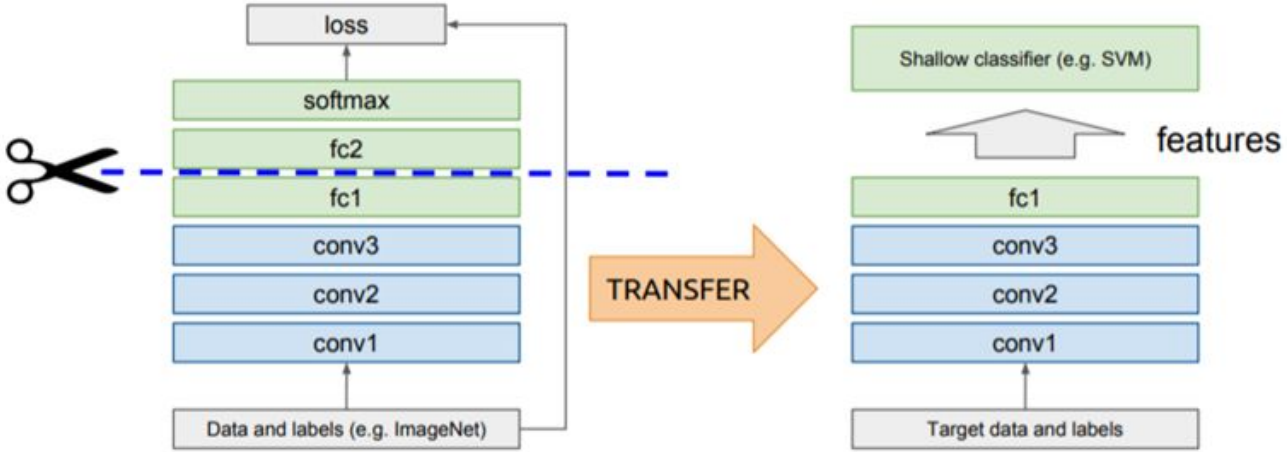
CERTIFAI
powered by Skymind

Transfer Learning

Transfer Learning Strategy

Pre-trained Models as Feature Extractors

Leverage the pre-trained models' weighted layers to extract features, the extracted features are then classified with shallow classifier like SVM

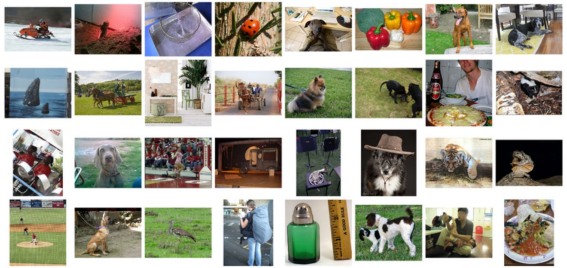


Transfer Learning

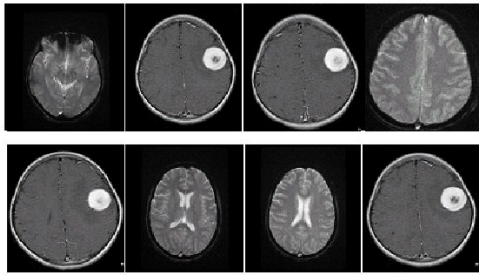
Transfer Learning Strategy

Fine Tuning Pre-trained Models for Domain Adaptation

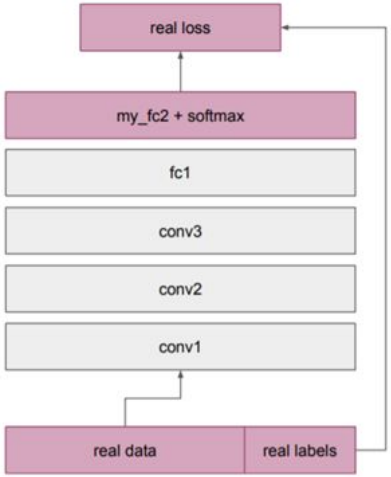
- Domain adaptation refers to the scenarios where the source distribution is different with target distribution. For example, *Imagenet dataset vs medical image dataset*.



Imagenet



Medical Imaging



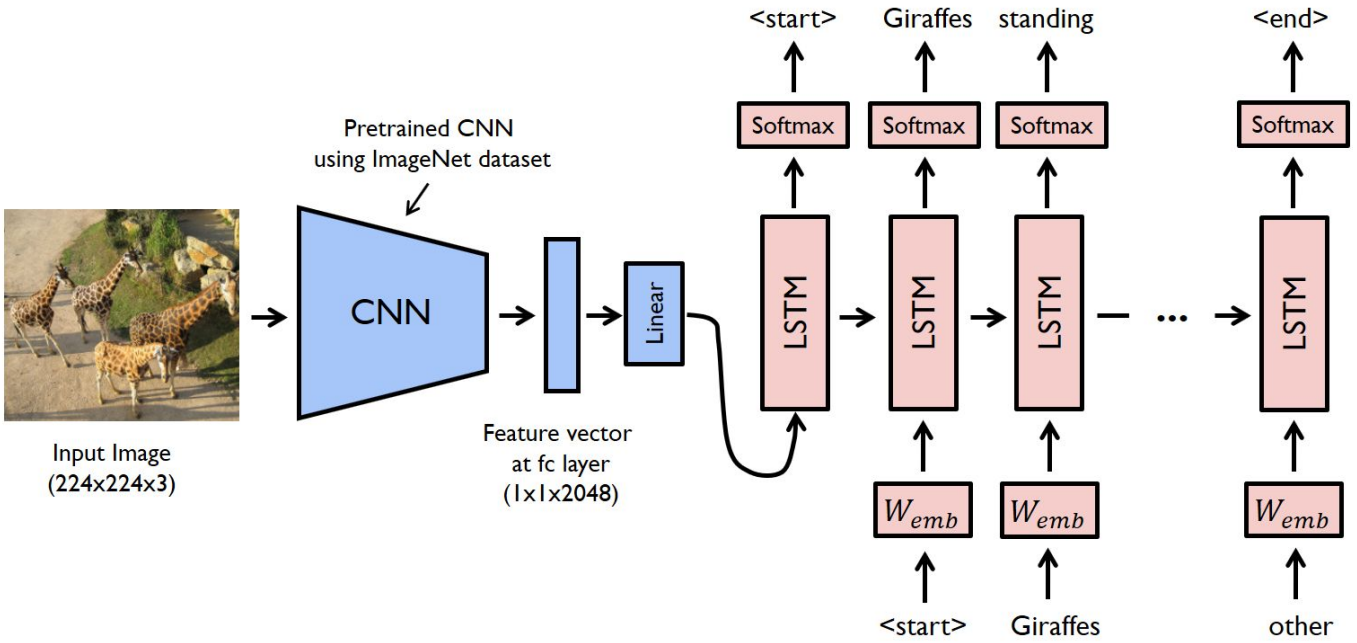
- For domain adaptation, the whole network requires fine tuning.

Transfer Learning

Overall Benefits of transfer learning

- Learning process can be faster
- Requires less training data
- Higher accuracy after training

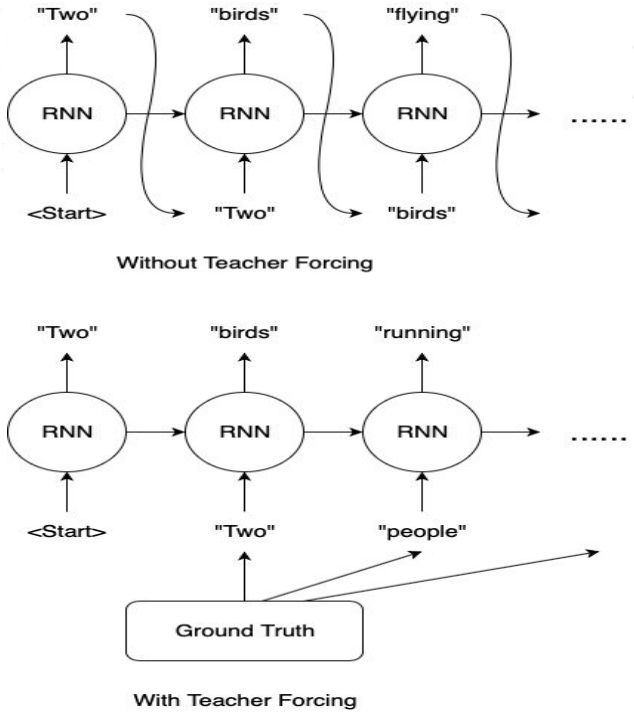
It comes to this…….



Teacher Forcing

Example:

- The ground truth caption for the image is “Two people reading a book”.
- Our model makes a mistake in predicting the 2nd word and we have “Two” and “birds” for the 1st and 2nd prediction respectively
- Without Teacher Forcing, we would feed “birds” back to our RNN to predict the 3rd word. Let’s say the 3rd prediction is “flying”. Even though it makes sense for our model to predict “flying” given the input is “birds”, it is different from the ground truth.
- if we use Teacher Forcing, we would feed “people” to our RNN for the 3rd prediction, after computing and recording the loss for the 2nd prediction.



Teacher Forcing

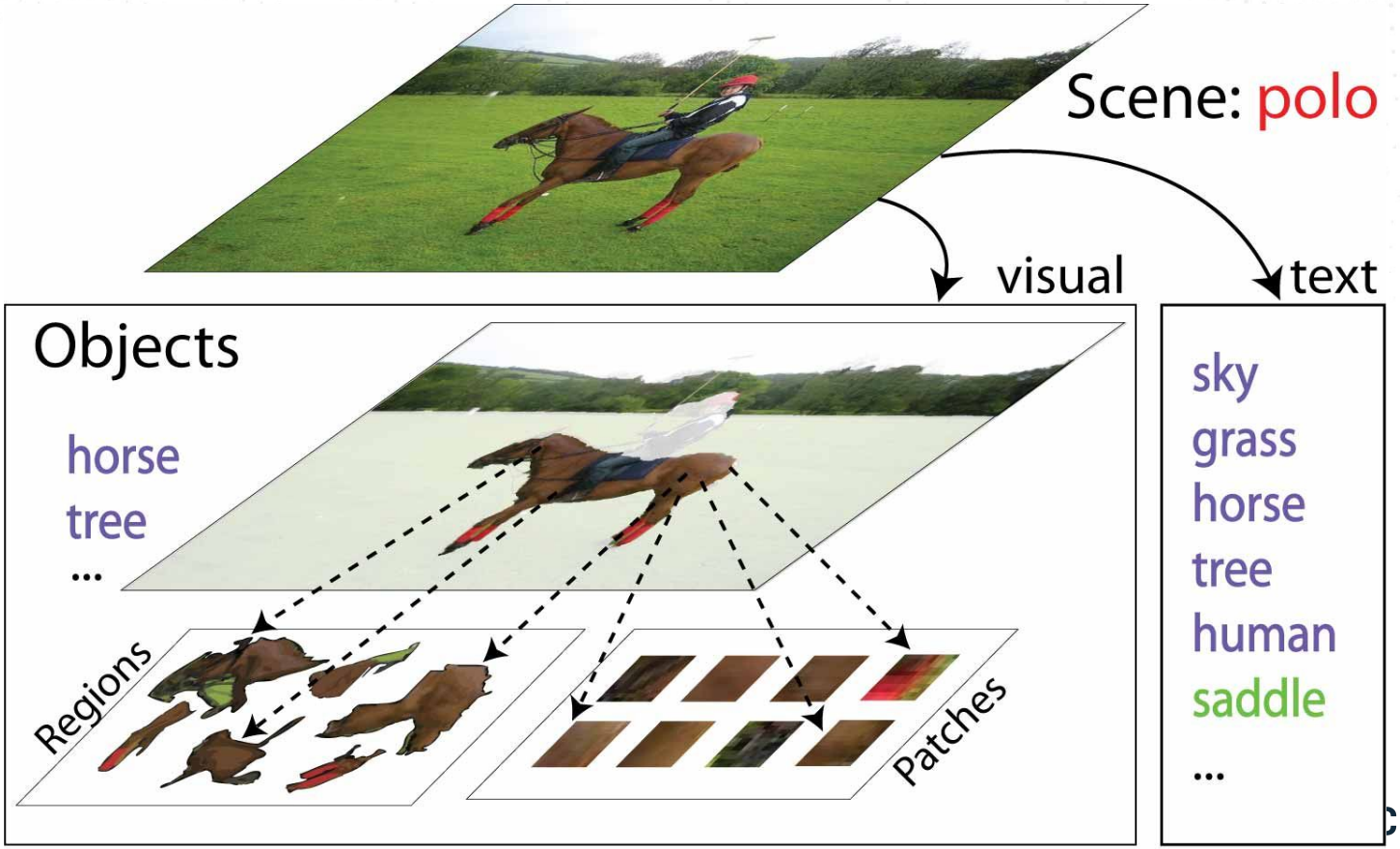
- Training with Teacher Forcing converges faster.
- At the early stages of training, the predictions of the model are very bad
- If we do not use Teacher Forcing, the hidden states of the model will be updated by a sequence of wrong predictions, errors will accumulate, and it is difficult for the model to learn from that

Appendix

Data Collection

- There are many open sources datasets:
  - **Flickr 8k**
  - Flickr 30k
  - MS Coco dataset

Computer Vision Tasks - Scene understanding



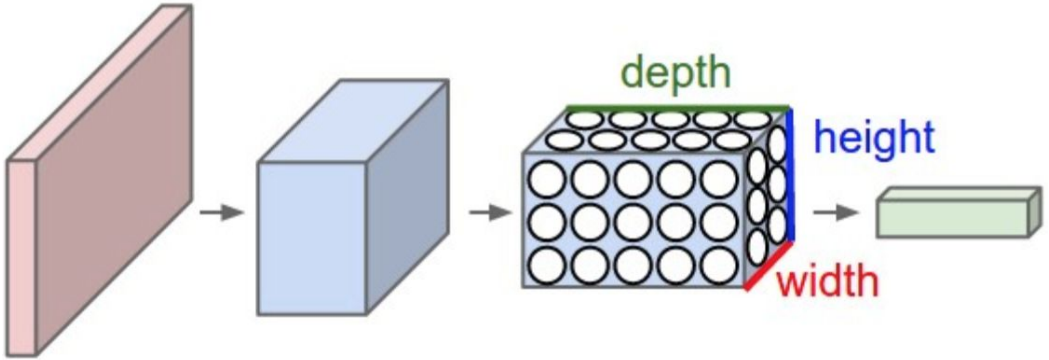
CNN Introduction

Convolutional networks perceive images as volumes (3 dimensional objects)

The layer of a CNN has neurons arranged in 3 dimensions

- width
- height
- depth

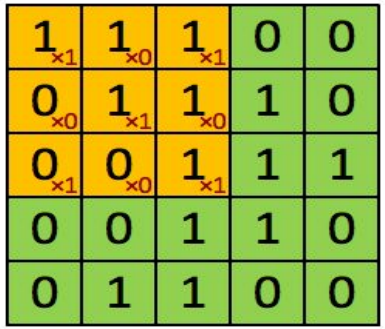
Input data



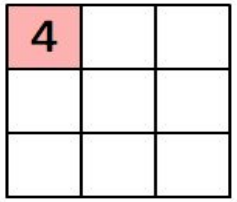
Neurons arranged in 3 dimensions, visualized in one of the layers

Features Extraction Layer - Convolutions

The role of the convolution is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.



Image



Convolved Feature

Input Layer

The input layer is 5x5 image indicated by the green colour.

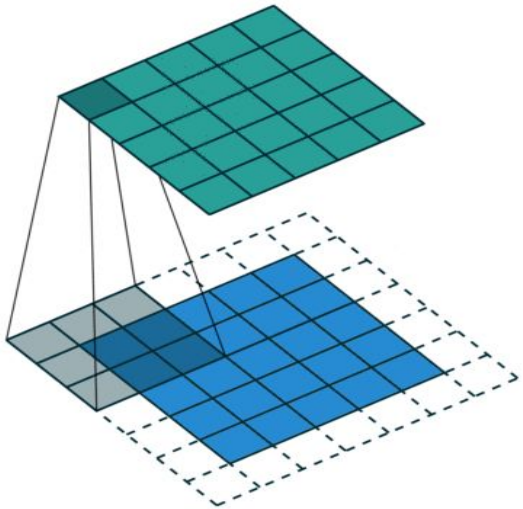
Kernel

The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel represented by the colour yellow. This Kernel size is 3x3.

Convoluted Layer

Convoluted layer size is 3x3 which contains the convolved features from the first input.

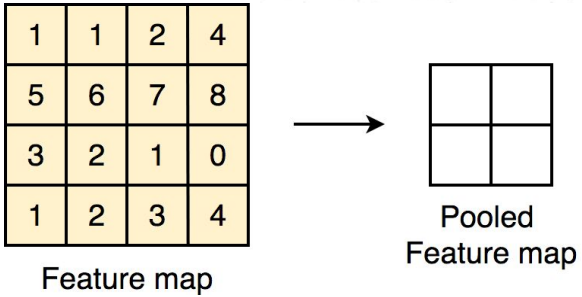
Features Extraction Layer - Convolutions



Objectives of convolution:

- Extract high-level features such as edges, from the input image.
- Conventionally, the first convolutional layer is responsible for capturing Low-Level features such as edges, color, gradient orientation, etc.
- With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

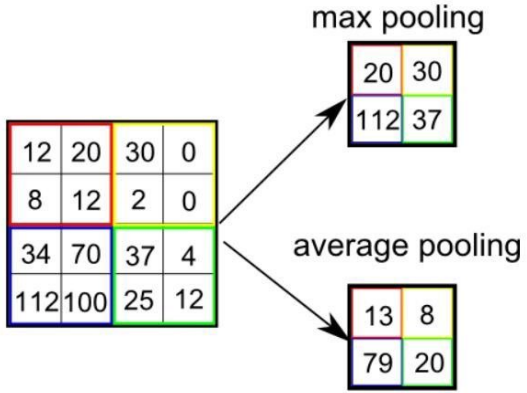
Features Extraction Layer - Pooling Layer



- Responsible for reducing the spatial size of the Convolved Feature.
- To decrease the computational power required to process the data through dimensionality reduction.
- Extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model.

There are two types of pooling:

- Max Pooling
- Average Pooling



CERTIFAI
powered by Skymind

What can NLP do?

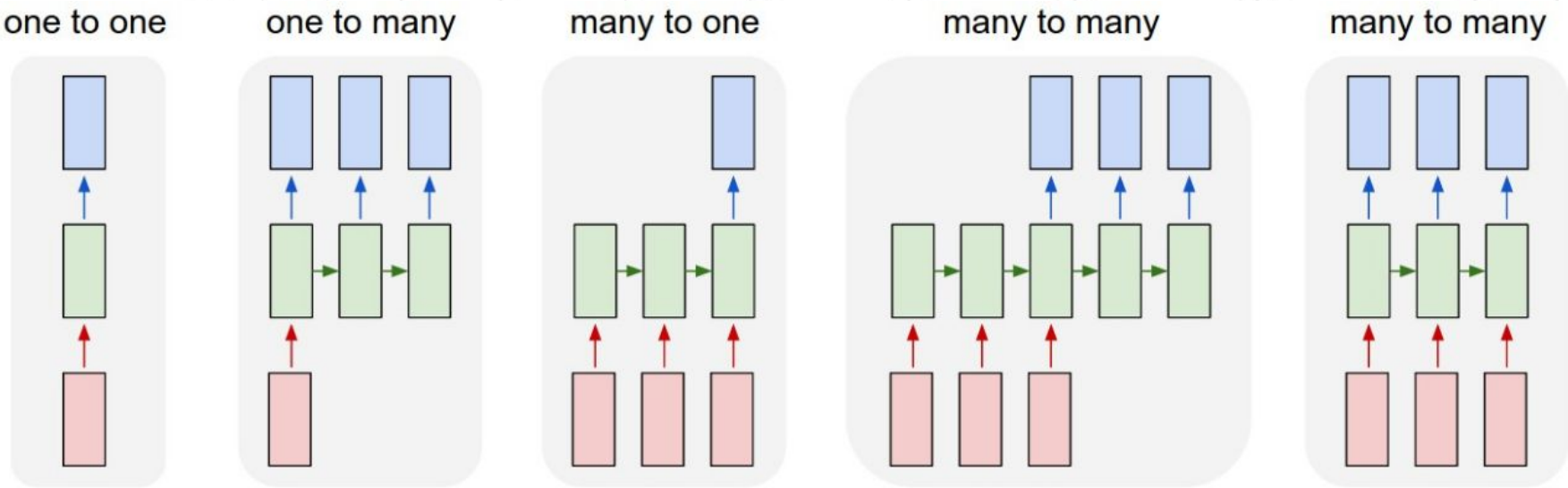
Speech Recognition (speech-to-text)	Part of Speech Tagging (grammatical tagging)	Named Entity Recognition (NEM)
• Converting voice data into text data. • Required for any application that follows voice commands or answers spoken questions.	• Process of determining the part of speech of a particular word or piece of text based on its use and context • **"**I can make a paper plane**" -> '**make' as verb in this context. • **"**What make of car do you own**"-> '**make' as noun in this context.	• Identifies words or phrases as useful entities • **"**Kentucky**" ->** location name • **"**Fred**" ->** mans name

What can NLP do?

Sentiment Analysis	Natural Language Generation (NLG)	Word Sense Disambiguation (WSD)
• Attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text. • "I am happy because I am learning NLP"-> positive • "I am learning NLP"-> neutral • "I am sad,I am not learning NLP"-> negative	• Task of putting structured information into human language. • Automated journalism	• Task to determine which meaning of the word is activated by the use of the word in a particular context. • "I can hear the **bass** sound." >> I can hear the sound **frequency** • "He likes to eat grilled **bass**" >>He likes to eat grilled **fish**. ** >> indicates as after WSD

CERTIFAI
powered by Skymind

RNN can operate on non-fixed size sequences of vectors (input and output)



Traditional Neural Network

Music Generation

Sentiment Classification

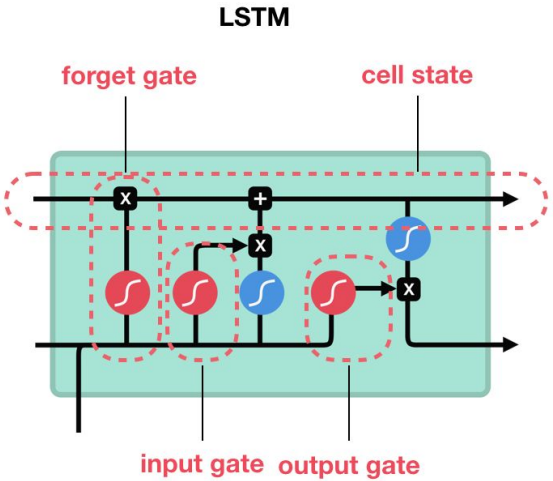
Machine Translation

Name Entity Recognition

Components of LSTM

Cell States:

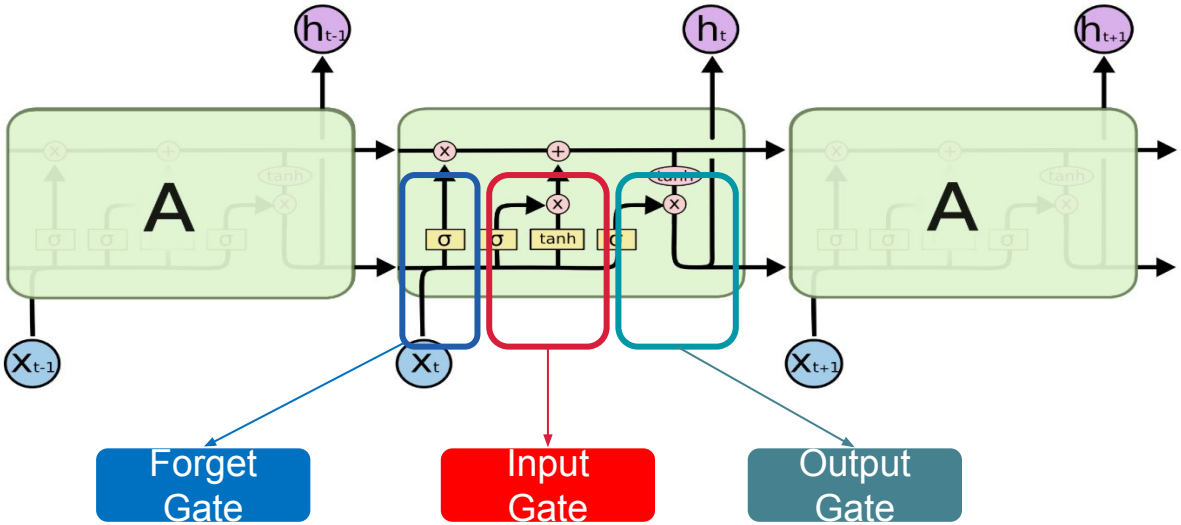
- the long term memory
- like a conveyor belt, information flows down the chain
- It is tasked to remember and forget information based on the context of the input.



Gates (input, forget, output):

- A way to optionally let information through
- Composed out of a sigmoid neural net layer and a pointwise multiplication operation

Components of LSTM



Forget Gate

Forget gate decides on which information to forget

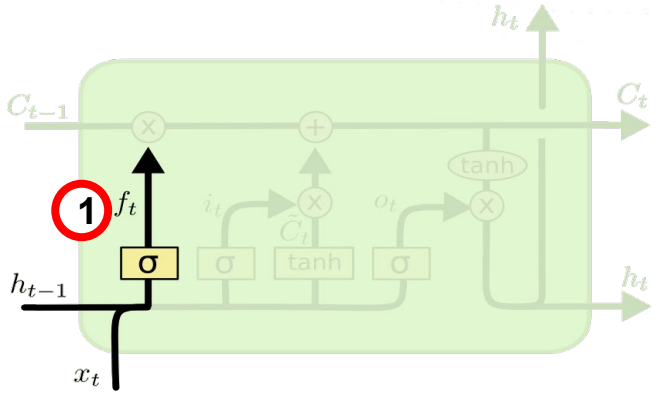
- Sigmoid function applied to weighted input and previous hidden state
- 1 to keep the information and 0 to forget

Forget gate equation

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

Output of this gate

$$= f_t * C_{t-1}$$



Input Gate

Input gate decides on which information to add

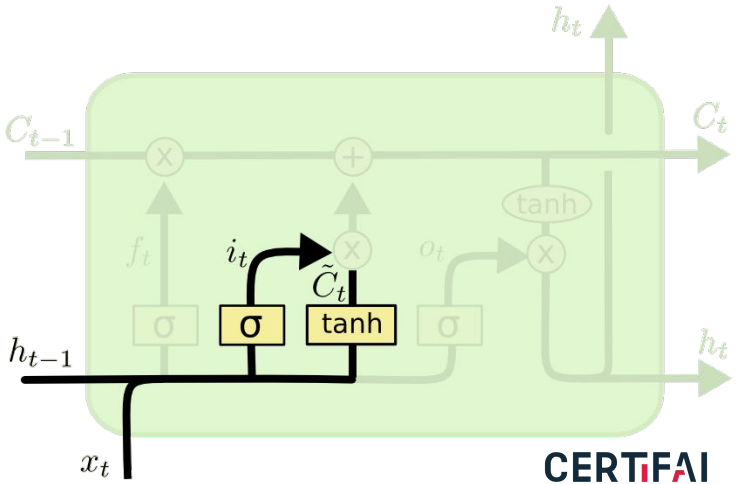
- Has two parts:
  - **(1) Sigmoid layer:** decides which input to emphasize now [0, 1]
  - **(2) Tanh layer:** decides which input to ignore [-1, 1]
- If only to add float number between [0, 1], a number will never be zero/ turned off. Hence the TanH activation

Input gate equation

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

Input Modulation gate equation

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



Cell State Update

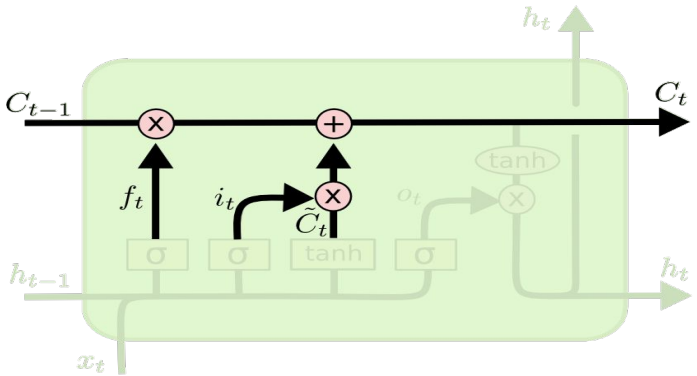
The two operations are combined to form a new cell state

- Multiply the old state by forgetting old information
- Add by multiplication as addition of new information

Notice that the equation of the cell state is a summation.

Cell state equation

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



Output Gate

Output of current time step, is the filtered version of the cell state

Has two stages

(1) Sigmoid layer decides which information to output

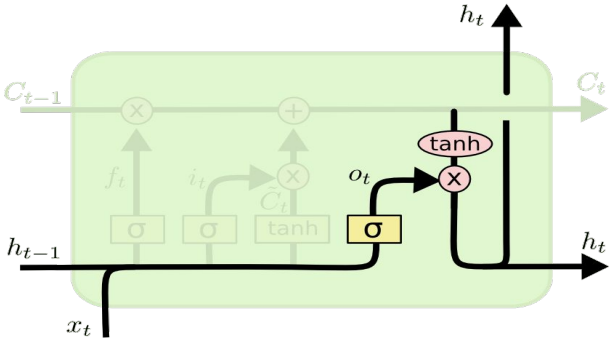
(2) Put cell state through tanh, and multiply with the output of sigmoid gate

Output gate equation

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

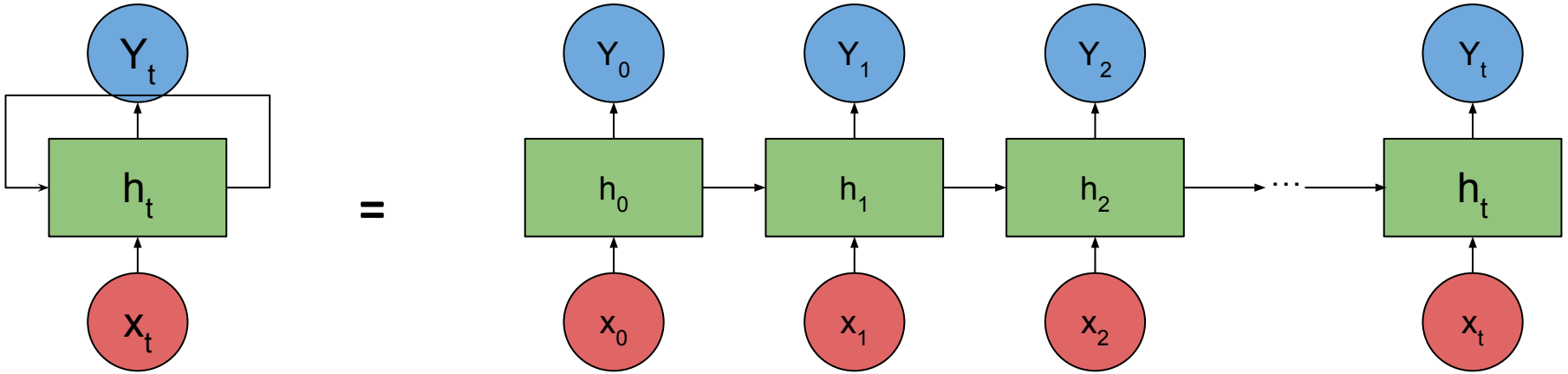
Current output equation

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



Recurrent Neural Network (many to many)

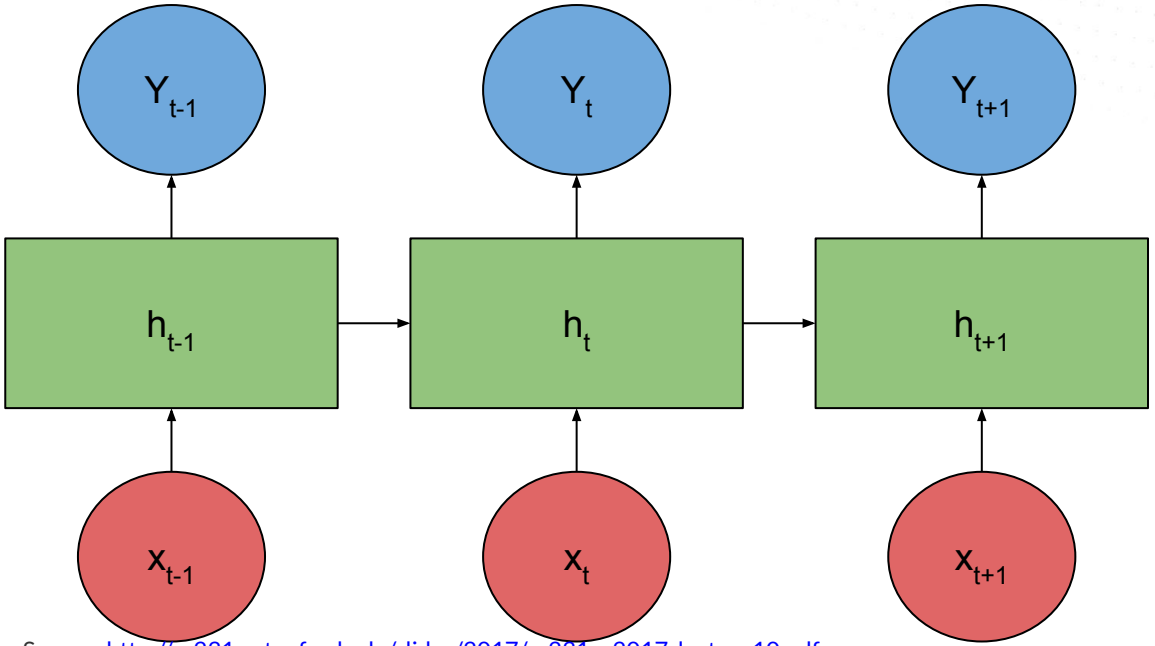
- **RNN includes a feedback loop that it uses to learn from sequences**
- Trained to generate sequences
- Output at each time step Y_t is based on
  - current input, and
  - input at all previous time steps



Rolled RNN

Unrolled RNN

How Recurrent Neural Network works



$$h_t = f_W(h_{t-1}, x_t)$$

new state

some function with parameters W

old state

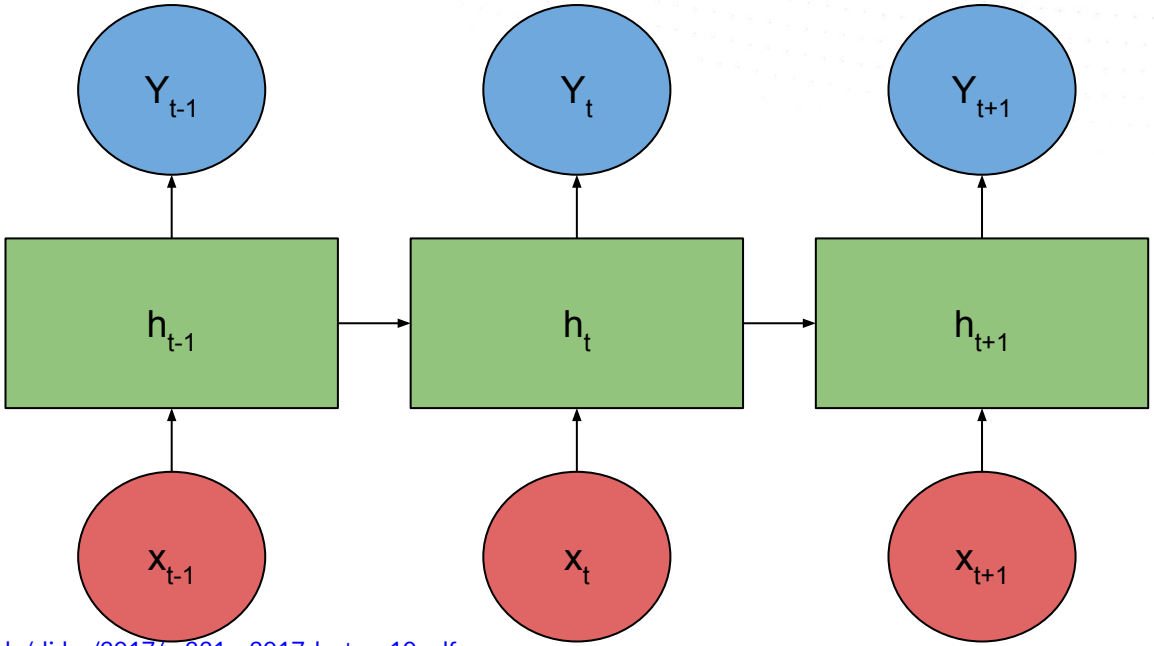
input vector at some time step

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Recurrent Neural Network (many to many)

How Recurrent Neural Network works

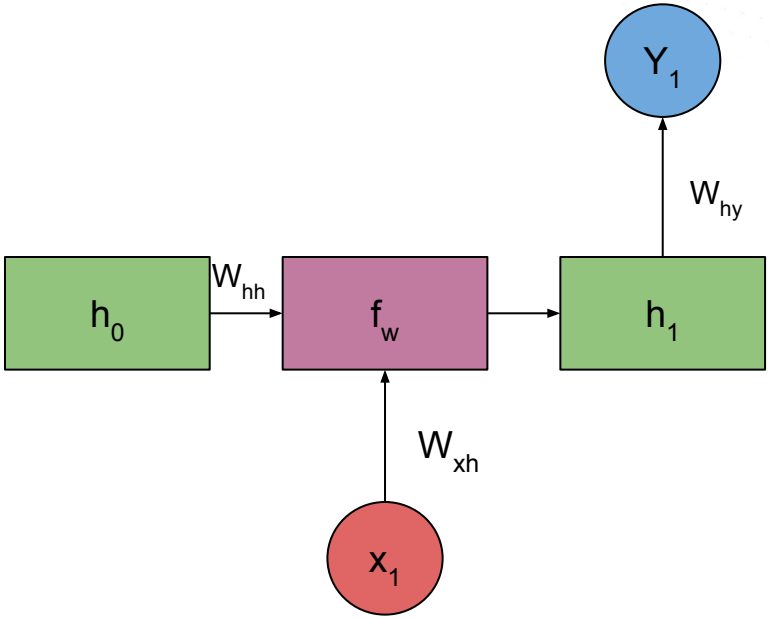
The hidden state **h** is a single vector.



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Recurrent Neural Network (many to many)

How Recurrent Neural Network works: Forward Propagation



$$h_t = f_W(h_{t-1}, x_t)$$

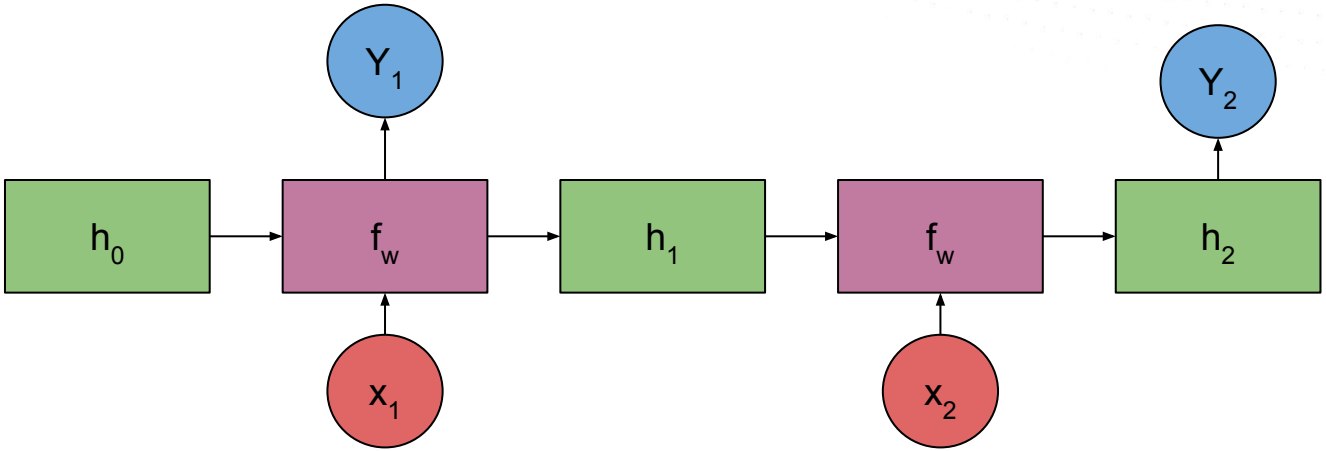
$$\downarrow$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

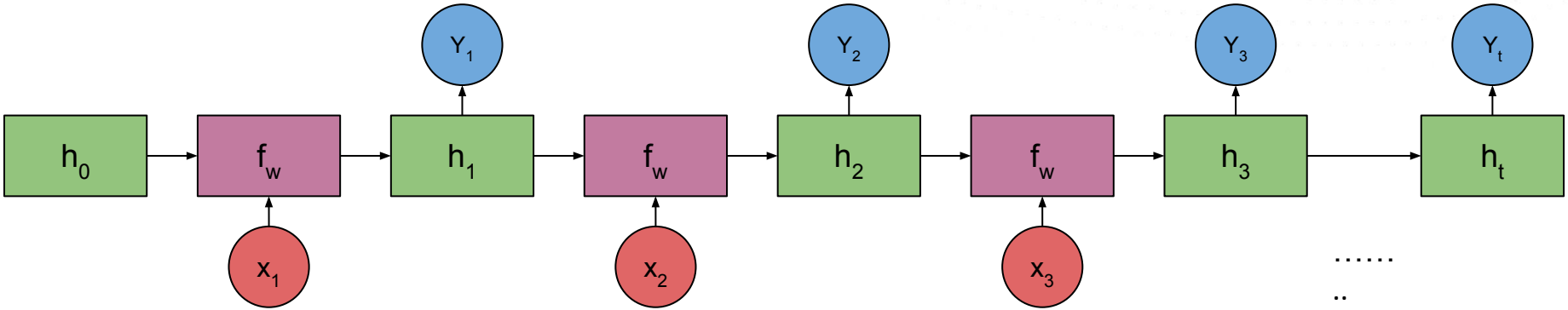
Recurrent Neural Network (many to many)

How Recurrent Neural Network works: Forward Propagation



Recurrent Neural Network (many to many)

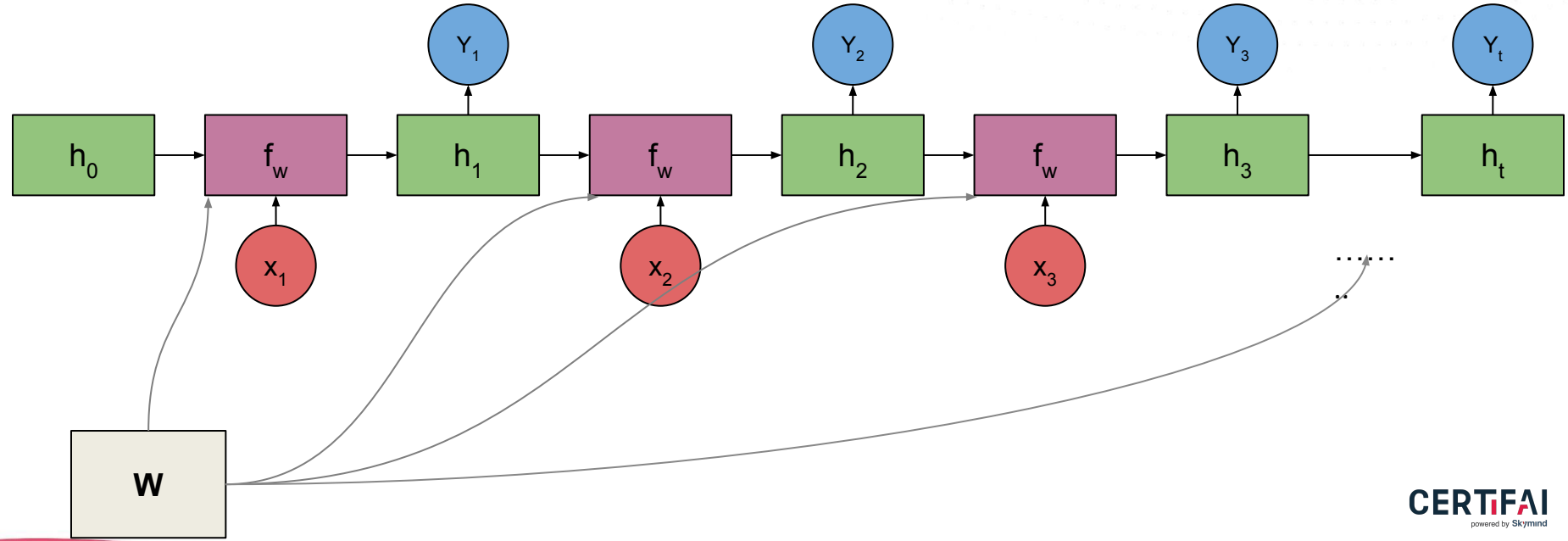
How Recurrent Neural Network works: Forward Propagation



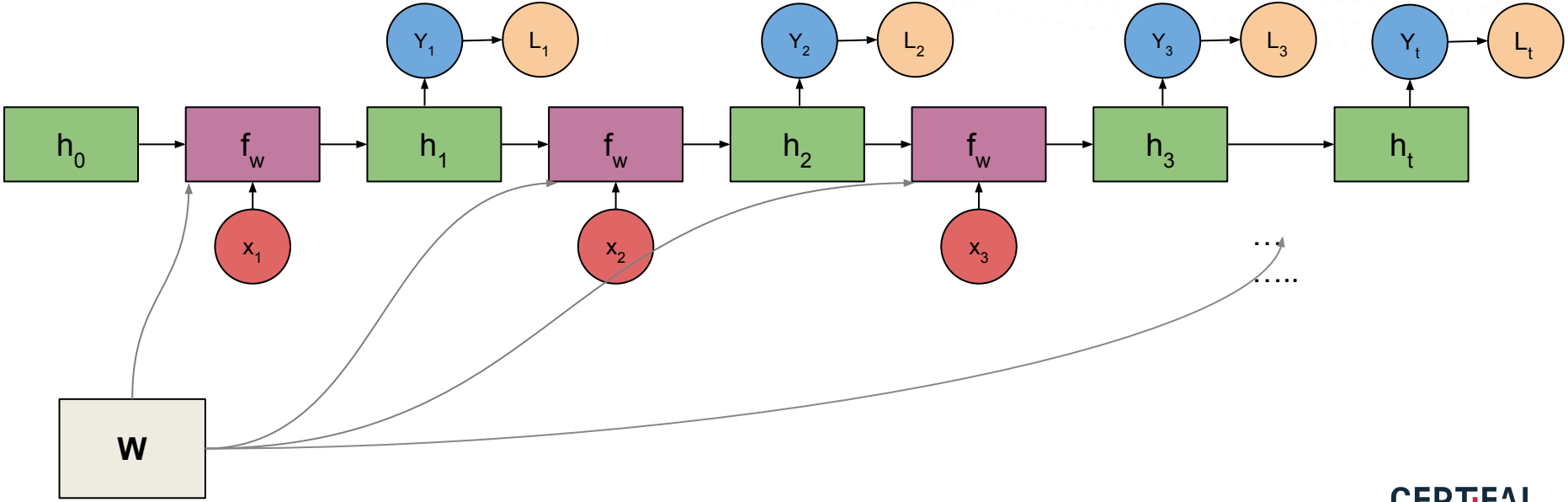
Recurrent Neural Network (many to many)

The same weight is re-used at every time-step

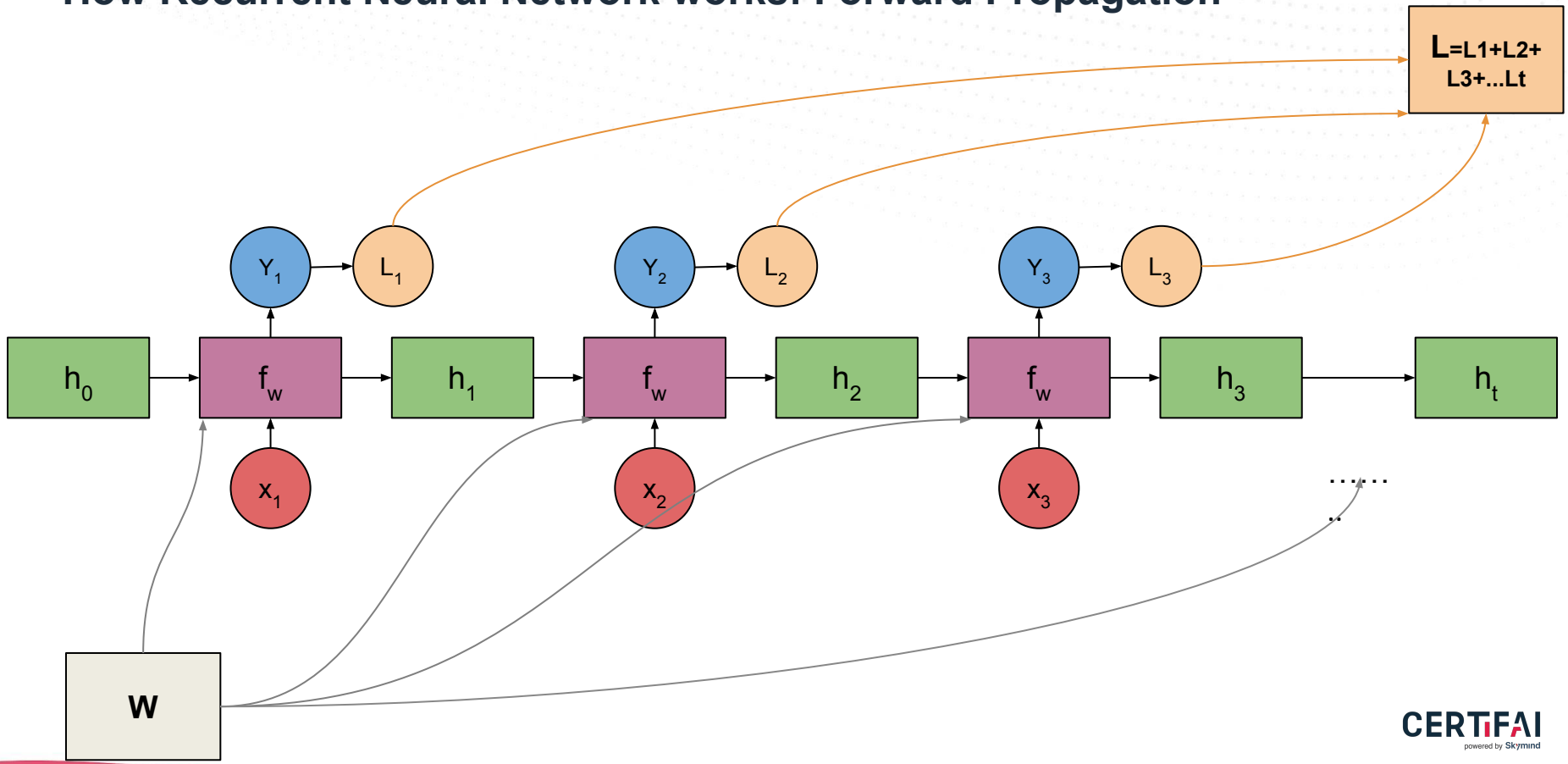
$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$



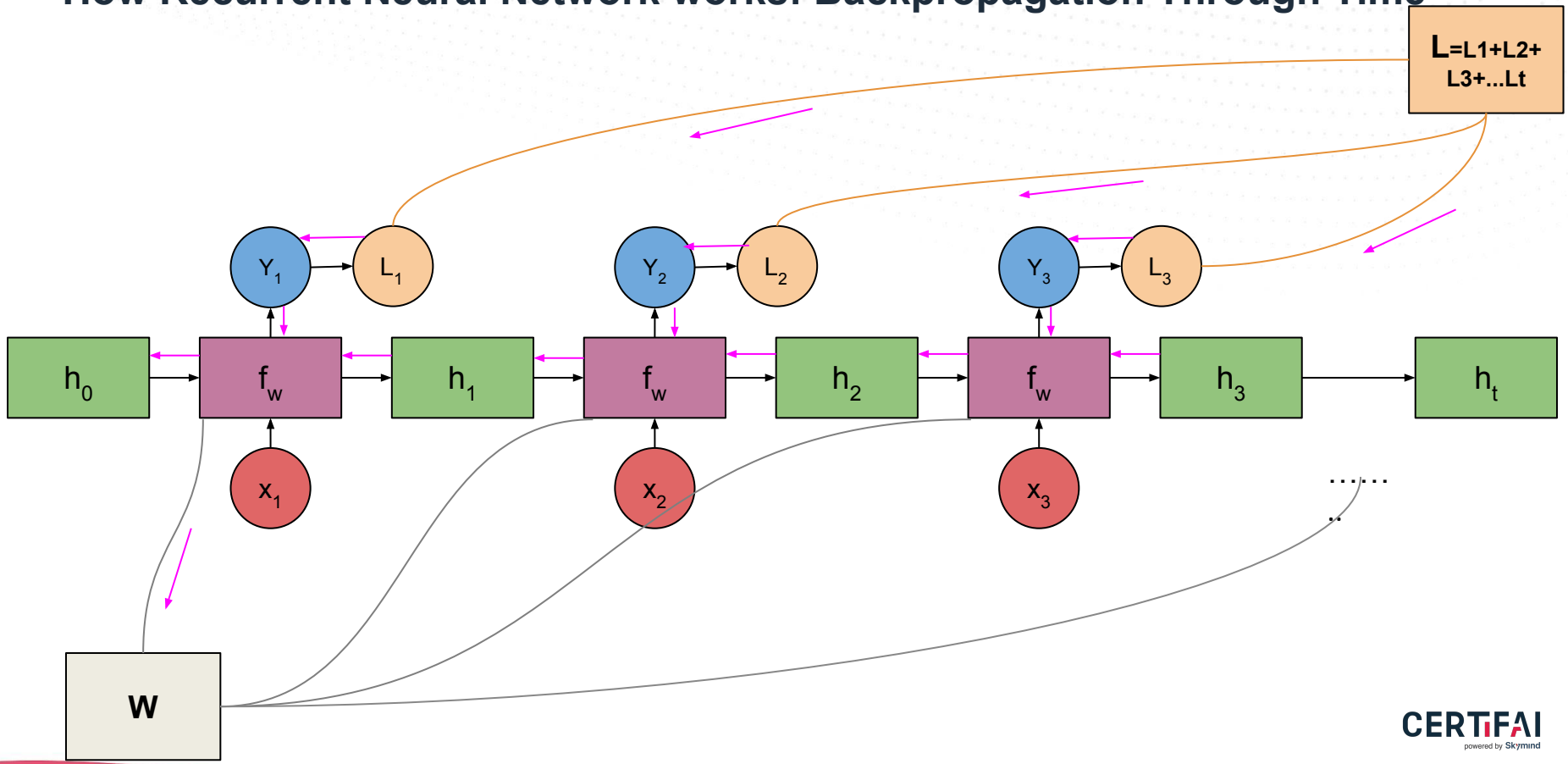
How Recurrent Neural Network works: Forward Propagation



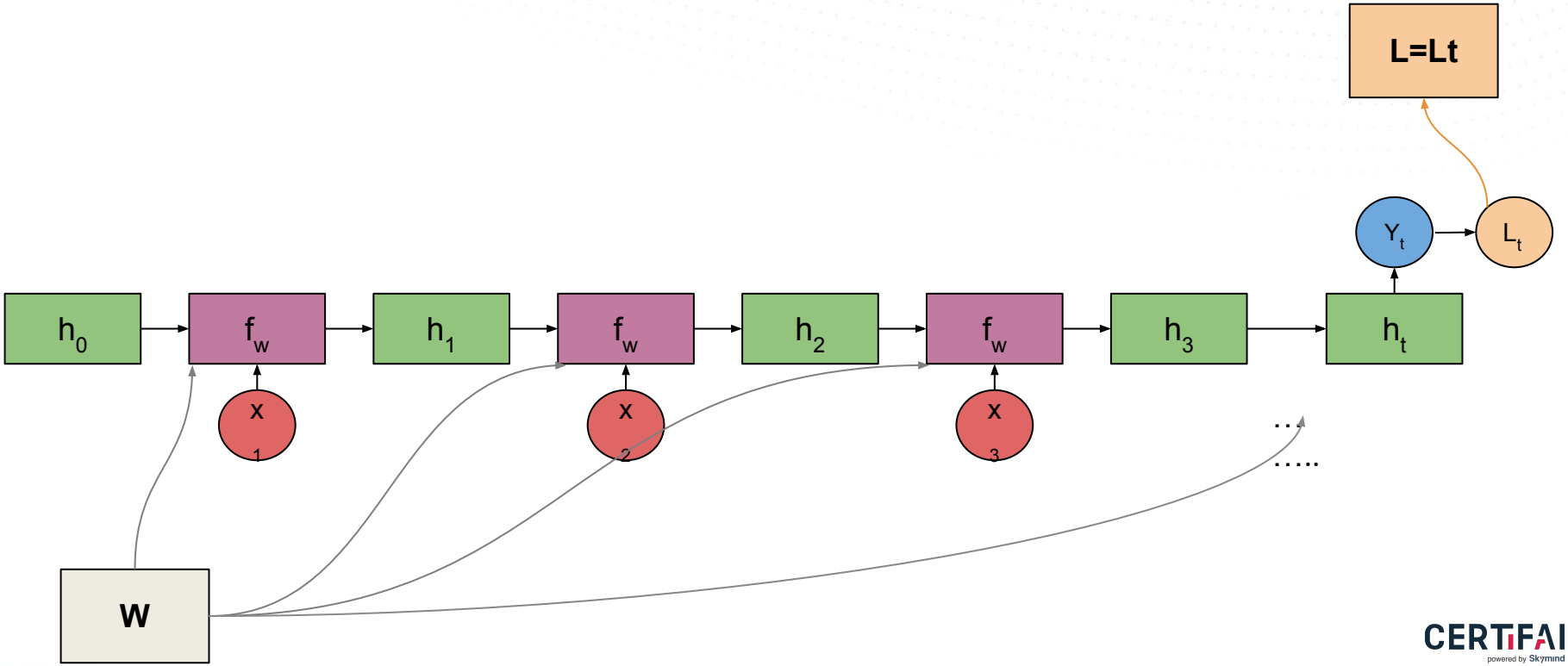
How Recurrent Neural Network works: Forward Propagation



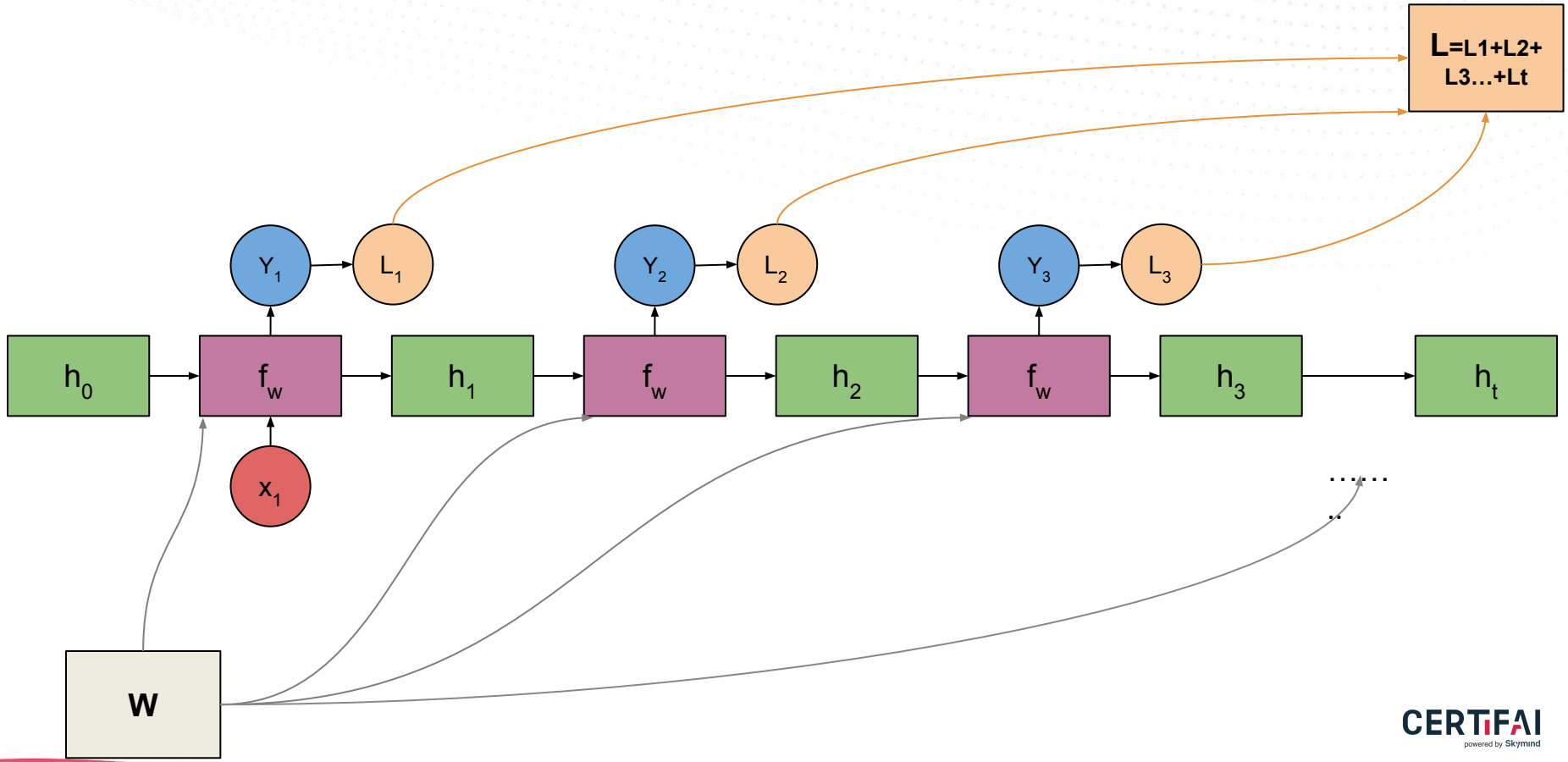
How Recurrent Neural Network works: Backpropagation Through Time



Recurrent Neural Network (many to one)

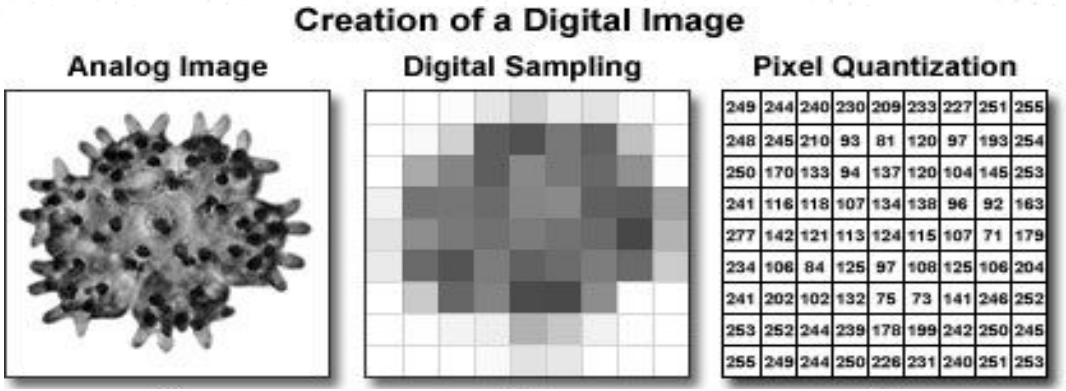


Recurrent Neural Network (one to many)



Text Data Preparation

How we represent text as in computer



Natural Language Processing → ['Natural', 'Language', 'Processing'] → [0.4, 0.12, 0.01, 0.64]

Tokenization **Work Embedding**

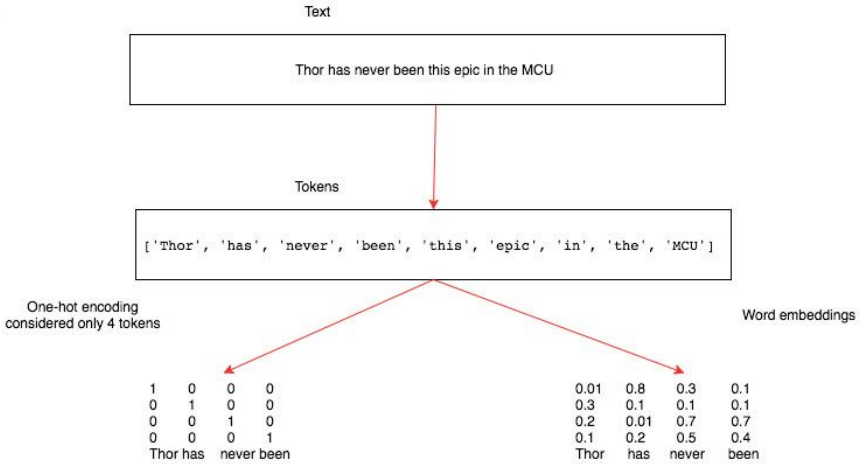
How we represent text as in computer

- In the context of NLP tasks, the text corpus refers to the set of texts used for the task.
- For example, if a model is built to analyse news articles, the text corpus would be the entire set of articles or papers we used to train and evaluate the model.
- The set of unique words used in the text corpus is referred to as the vocabulary

raw text corpus → processed text → tokenized text → corpus vocabulary → text representation

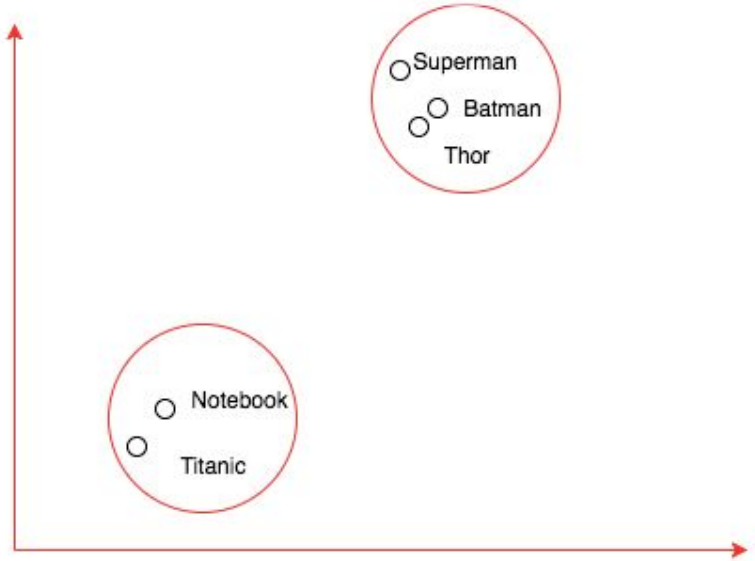
Word Embedding

- Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation.
- An embedding is a mapping of a discrete — categorical — variable to a vector of continuous numbers.
- Embedding refers to learning this mapping from one discrete type to a point in the vector space.



Word Embedding

- Word embedding provides a dense representation of a word filled with floating numbers.
- The vector dimension varies according to the vocabulary size.
- Common to use a word embedding of dimension size 50, 100, 256, 300, and sometimes 1,000
- The dimension size is a hyper-parameter that we need to play with during the training phase.



Word Embeddings Visualization: https://projector.tensorflow.org/

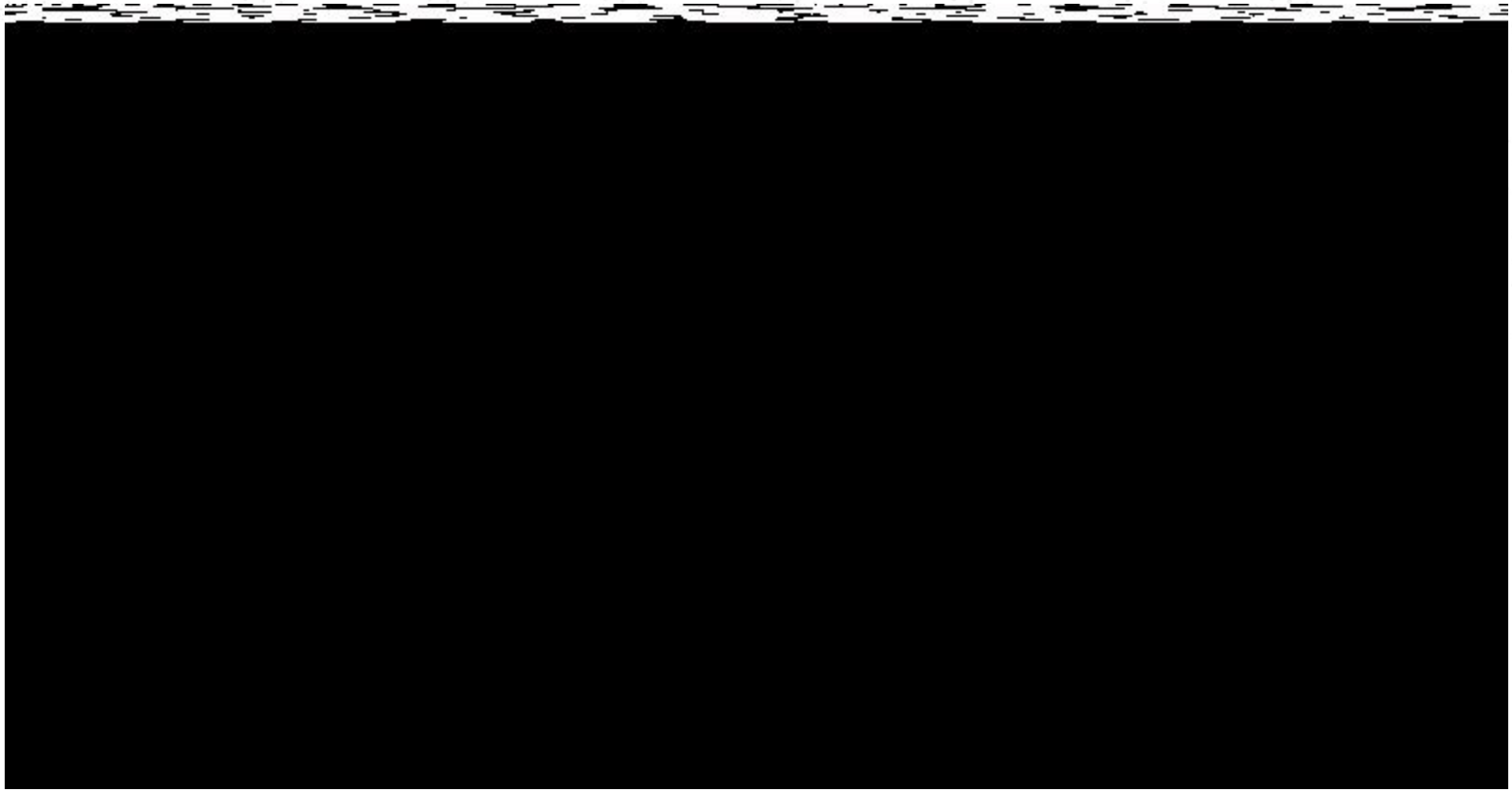


Image Data Preparation

Image Augmentation

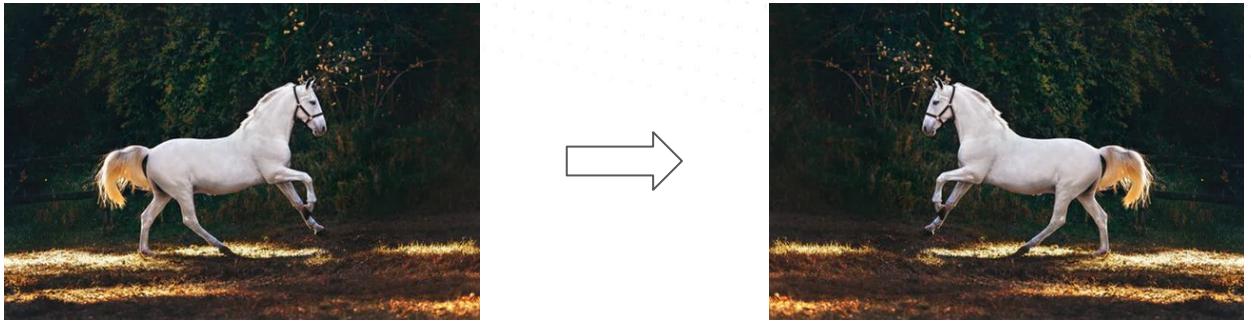
Image augmentation is a technique that can be used to **increase the size of a training sample**.

Image augmentation **randomly transform training examples**, reduce a models' dependence on certain properties, therefore **improving deep neural networks' capability for generalisation.**

The transformation are usually operations to alter the image such as: **flips, shifts, random crop, contrast or color adjustment.**

Image Augmentation

Horizontal flip



Vertical flip

CERTIFAI
powered by Skymind

CERTIFAI
powered by Skymind

Thank You

Follow & talk to us for more info!!

https://certifai.ai/

https://www.facebook.com/certifai

https://www.linkedin.com/company/certif-ai/

http://certifai.eventbrite.com