Deep Learning

for

Natural Language Processing

whoami

Valerio Basile
Postdoc researcher at University of Turin
Content-Centered Computing group
Hate Speech Monitoring group
http://valeriobasile.github.io/
http://hatespeech.di.unito.it/

Outline

Intro: NLP, Machine Learning and Neural Networks

Deep learning: motivation and models

More deep learning: extra features, resources

Conclusion: practice, questions

“Il deep learning è una roba da matematici più che da informatici”

– Anonimo, Torino 2019

Natural Language Processing

Automatic processing of human language

Applications: machine translation, speech recognition/synthesis, conversational agents, information extraction, ...

Natural Language Processing

Tasks:
tokenization
tagging (e.g., part-of-speech)
parsing (e.g., syntactic parsing)
classification (e.g., sentiment)
...

Machine Learning

learn structure from data
rather than
code structure in the program

Thanks, WWW! (ML existed in the 70s already)

You will still write code though.

Machine Learning

Unsupervised Machine Learning
Learn structure from data alone.
Examples: clustering, word embeddings.

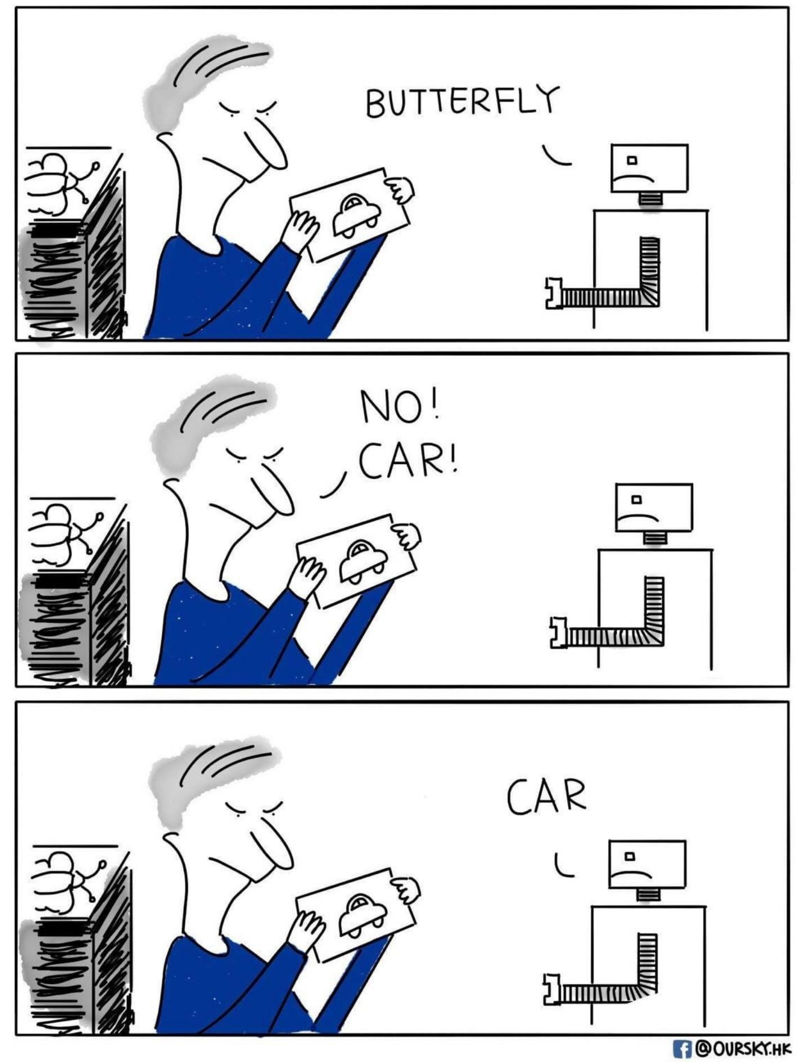
Supervised Machine Learning
Learn structure from annotated data.
Examples: linear regression, support vector machine, neural networks.

Supervised Machine Learning

Learn from annotated (labeled) data.

Annotation comes from experts, distant supervision, crowdsourcing, gamification, ...

The goal is to generalize, not overfit.



Supervised ML models

Naive Bayesian
Linear Regression
Logistic Regression
Conditional Random Fields
Decision trees
Random Forest
Support Vector Machine
Neural Networks
...

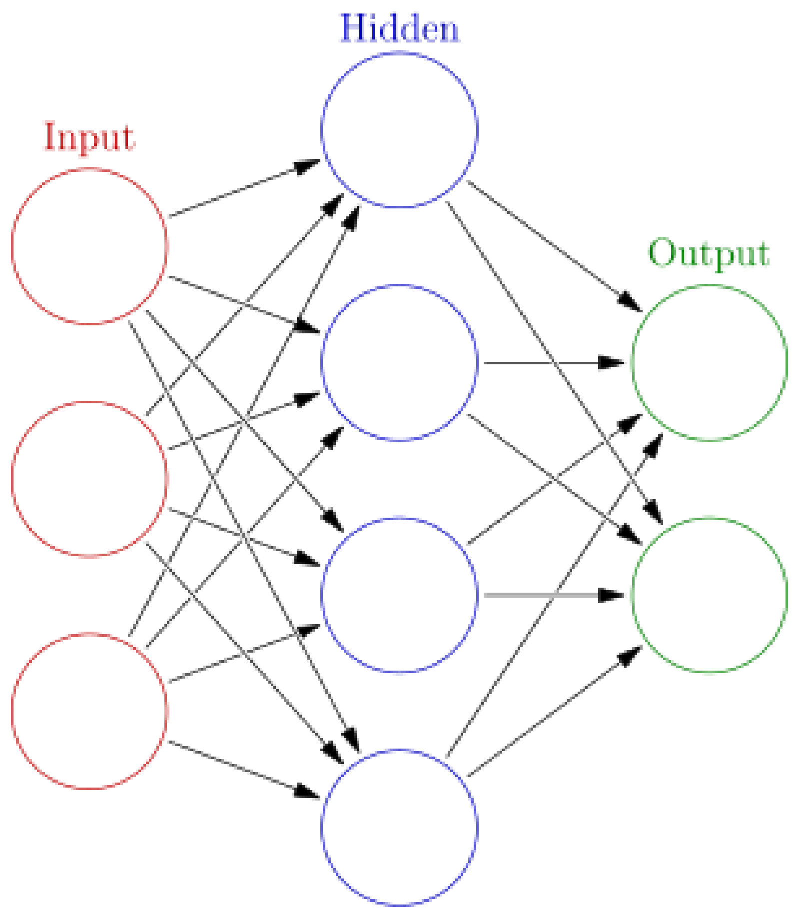
Supervised ML models

training: data+labels → model

prediction: data+model → labeled data

Neural Network

Artificial neural networks are a computational model inspired by biological neural networks

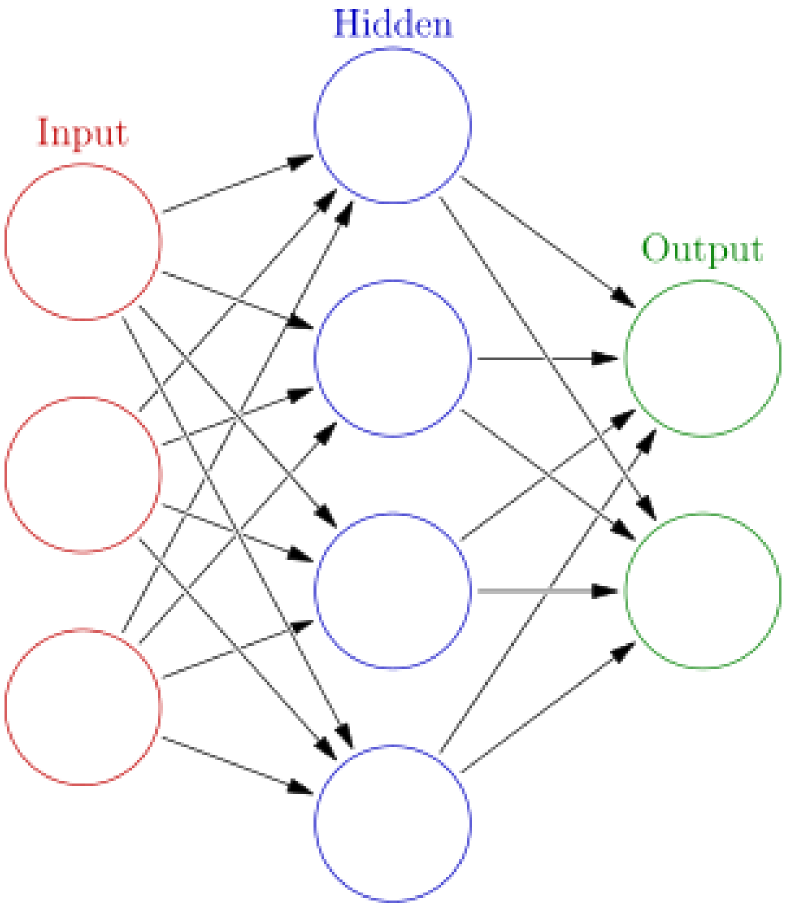


Feedforward

Input is encoded in input nodes (i.e., features)

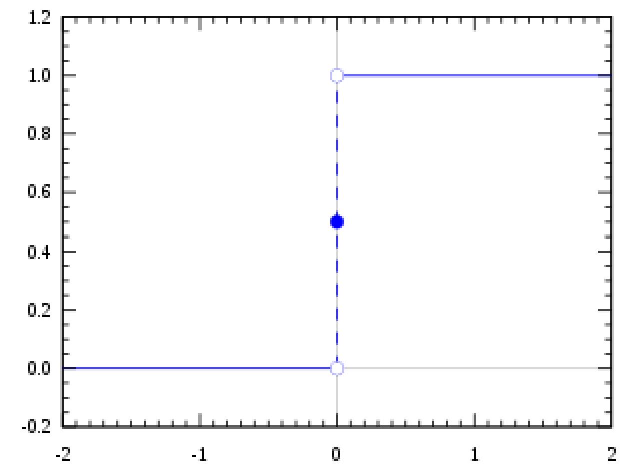
Edges have weights

Output is decoded from output nodes



Activation function

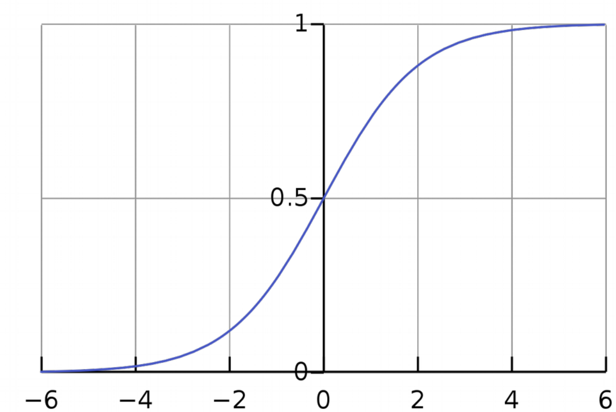
Input is multiplied by weight and fed to an activation function



Simple: step function

Better: smooth version (sigmoid), e.g. logistic function

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}}$$



Backpropagation

Crucial for learning.

Needs a defined loss function to measure the error between prediction and ground truth.
Example: squared Euclidean distance

Weights, initialized randomly, are adjusted with gradient descent

Gradient Descent

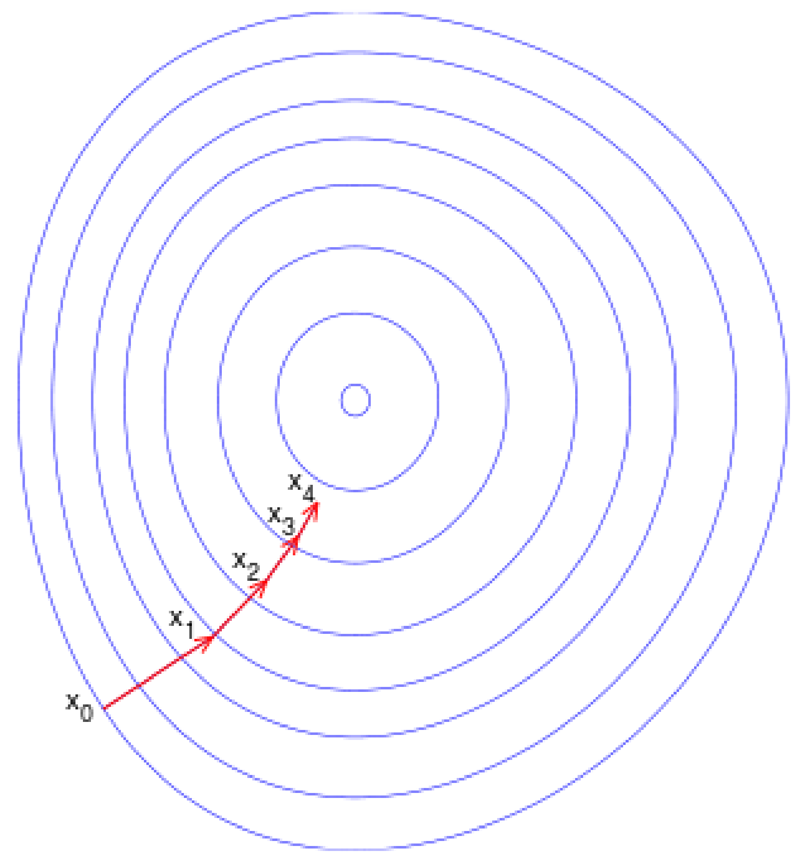
Highly popular optimization technique

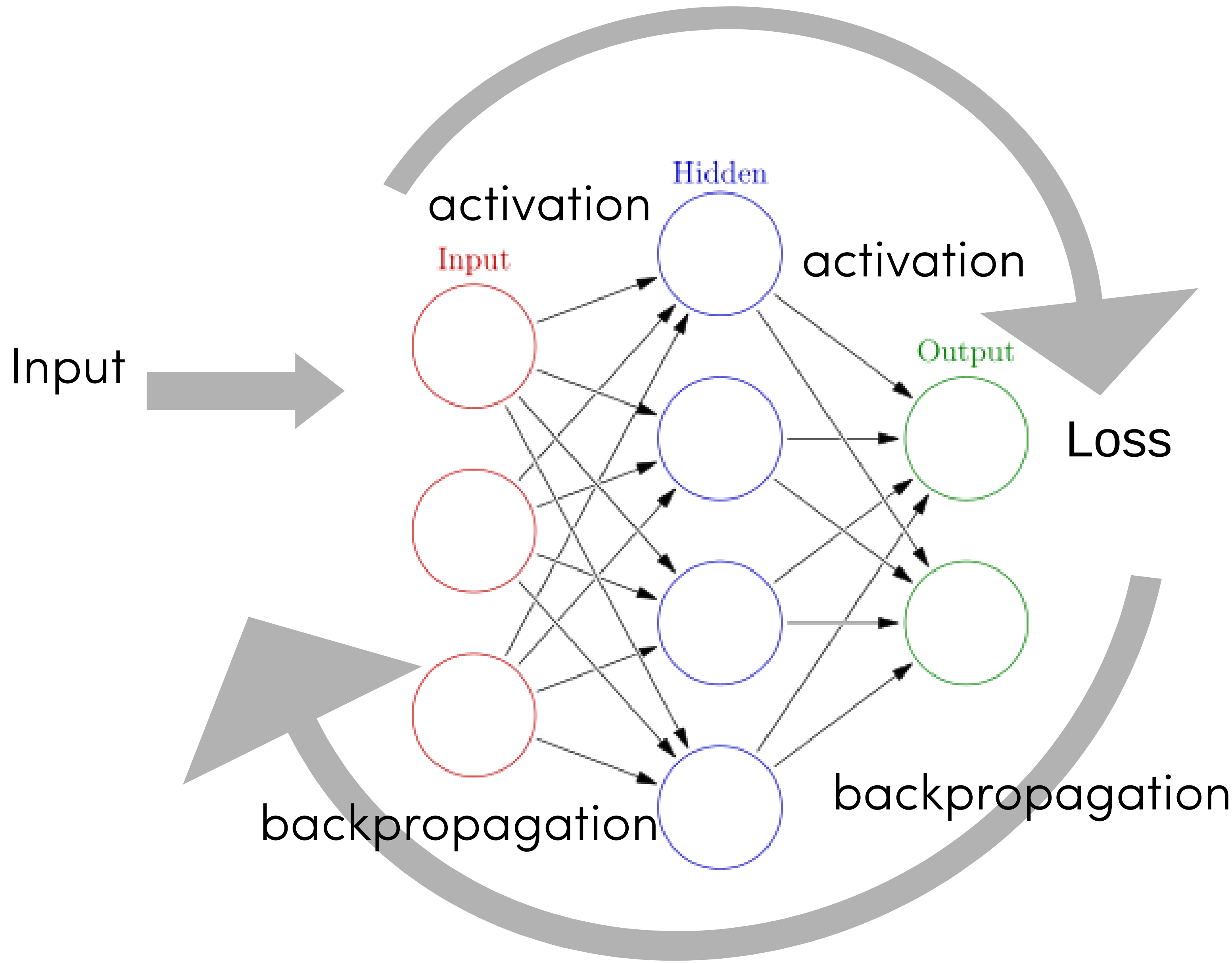
$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

new weight

old weight

loss func.




Input
activation
Hidden
Input
activation
Output
Loss
backpropagation
backpropagation

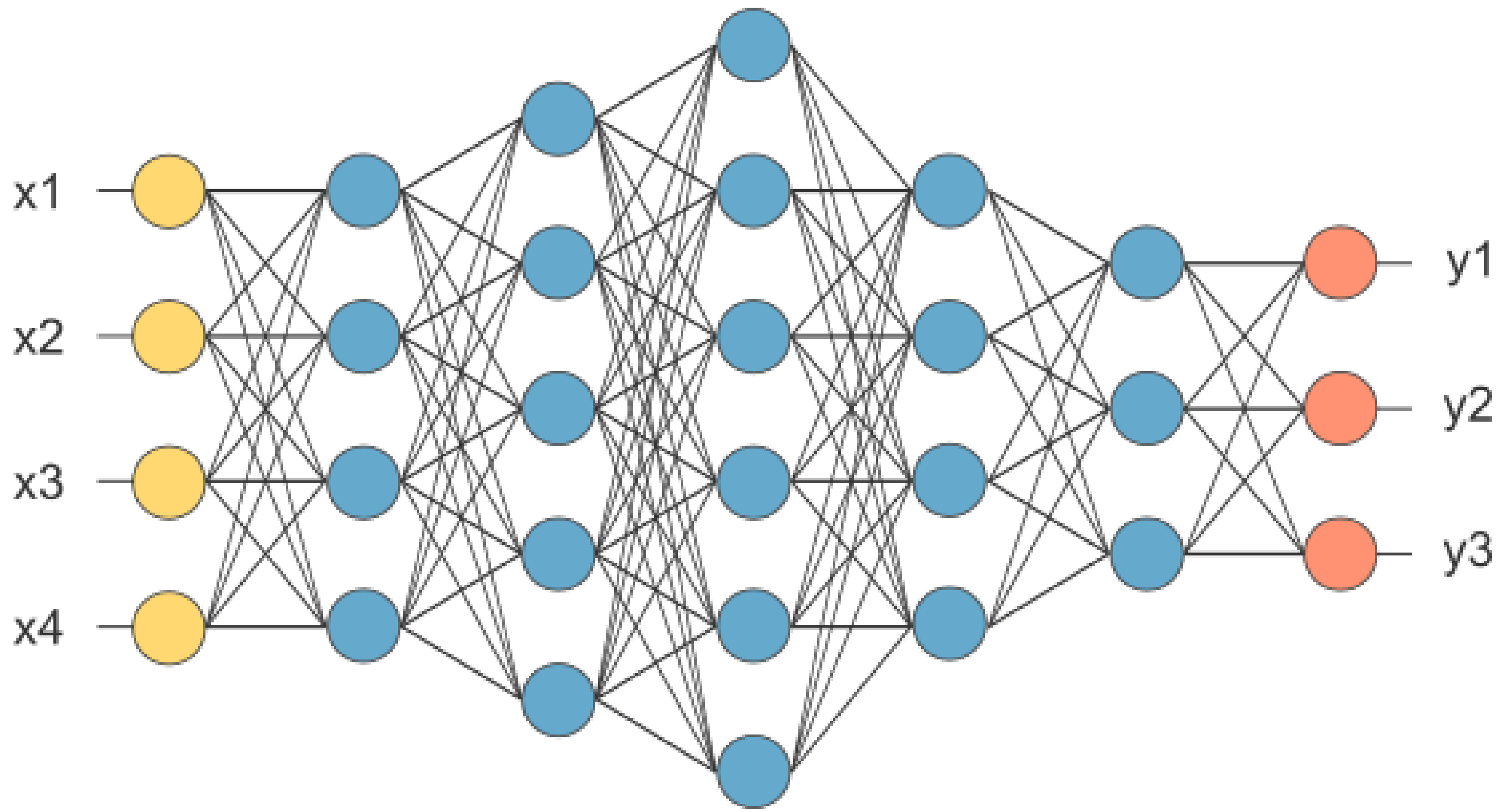
Deep Learning

Why deep?

Deep Learning

Why deep?

More layers.



Deep Learning

Why more layers?

Deep Learning

Why more layers?

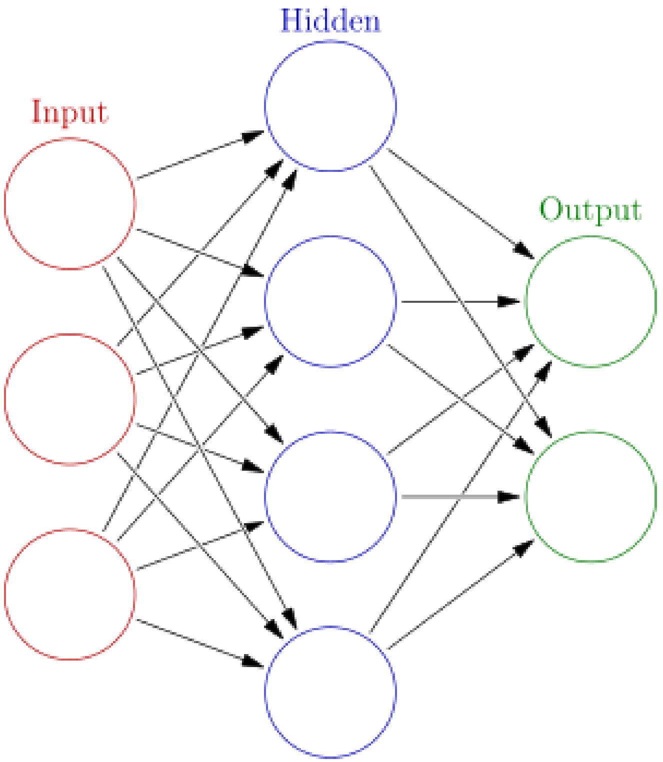
Capturing interactions between features

Learning high-level representations

https://youtu.be/pfFyZY1RPZU?t=1761

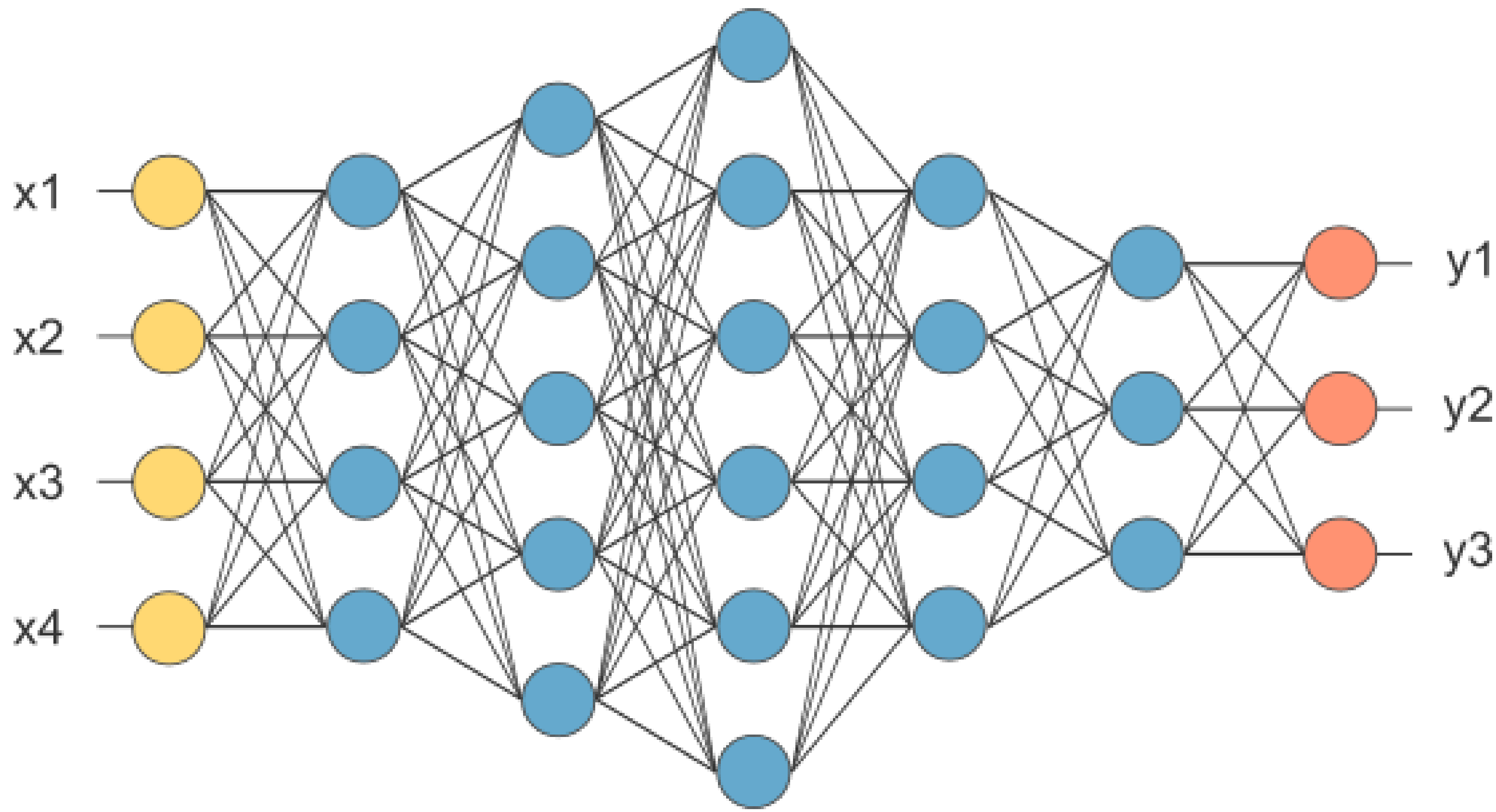
Multilayer Perceptron

Confusingly called *multilayer*
even with 1 hidden layer



Deep Multilayer Perceptron

Many hidden layers
Not necessarily of the same size



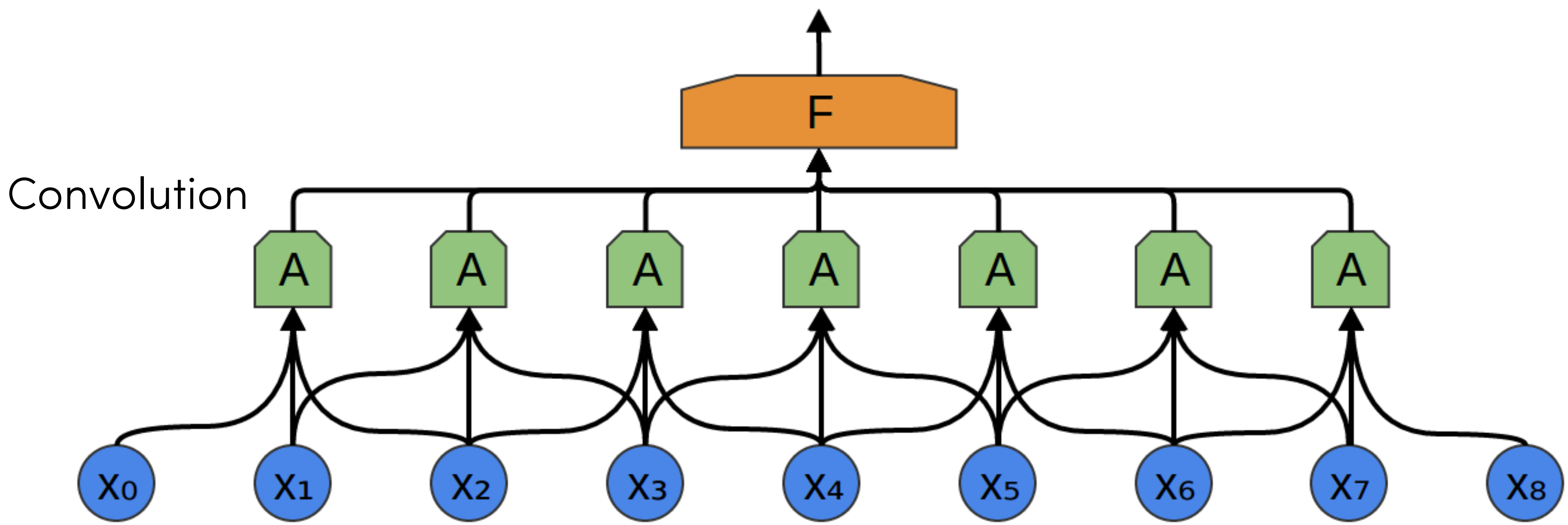
Convolutional Neutral Network

Uses copies of the same neurons (parameter sharing) to learn local features

Based on the convolution matrix operation

Gained popularity from
computer vision and voice recognition

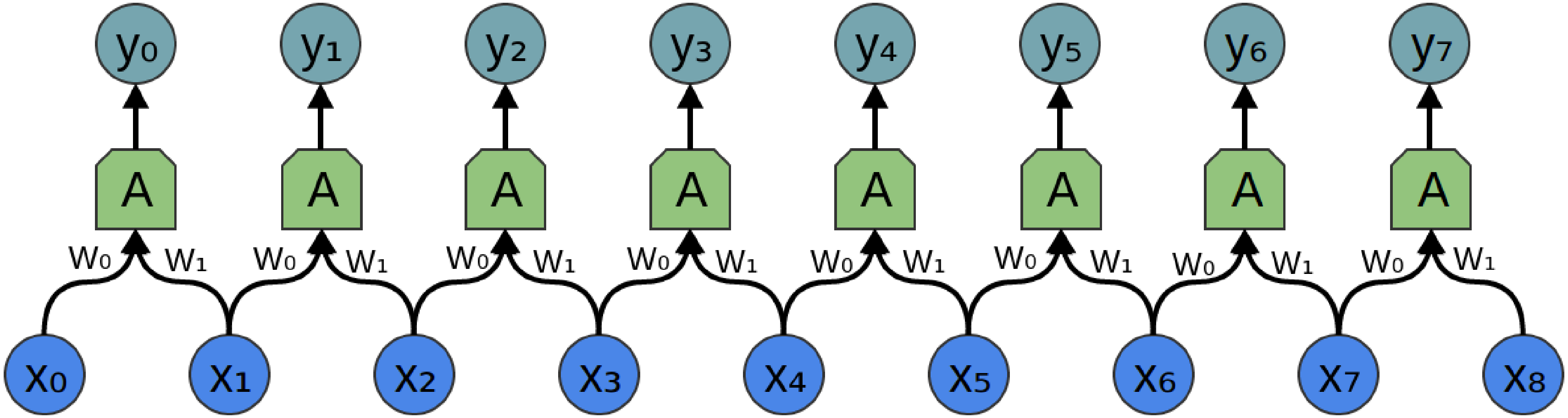
Convolutional Neural Network



The size of the convolution is a parameter

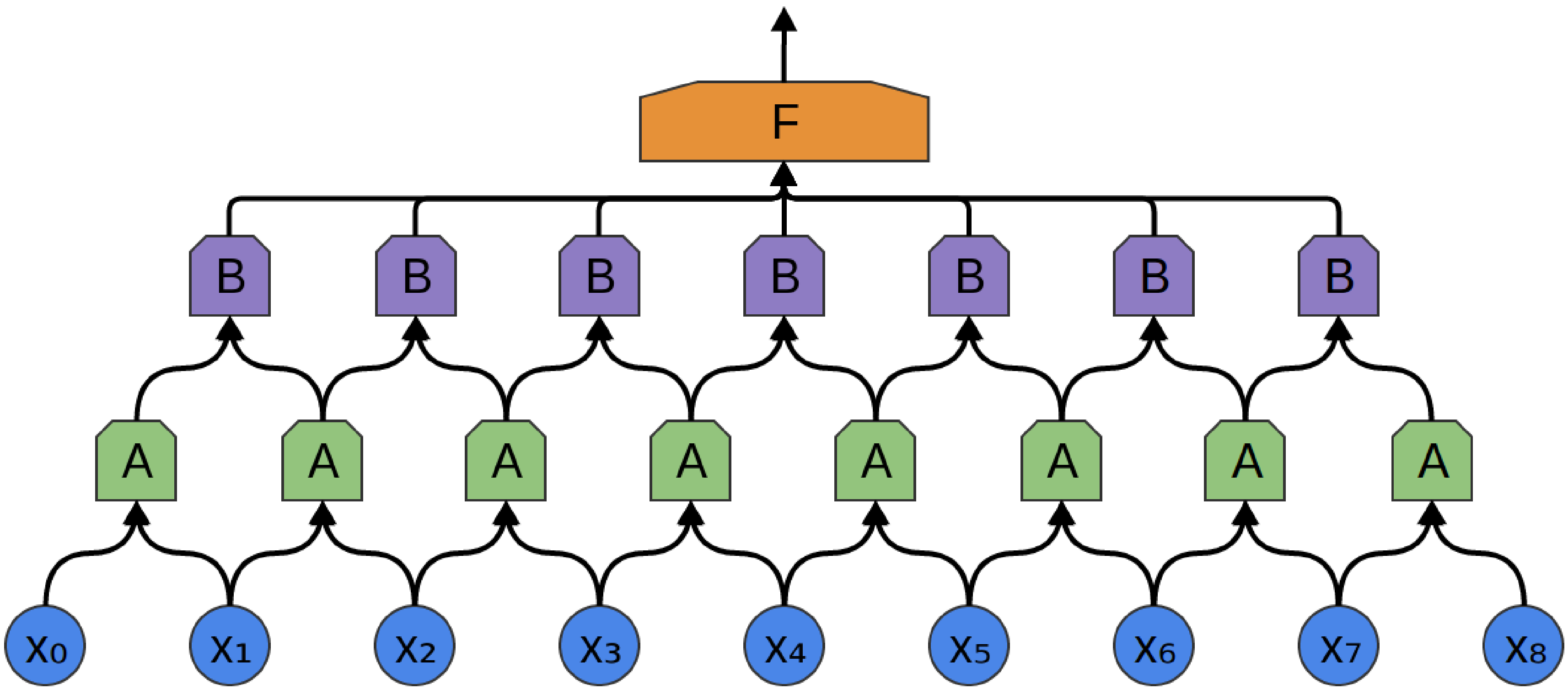
Images from http://colah.github.io

Convolutional Neural Network



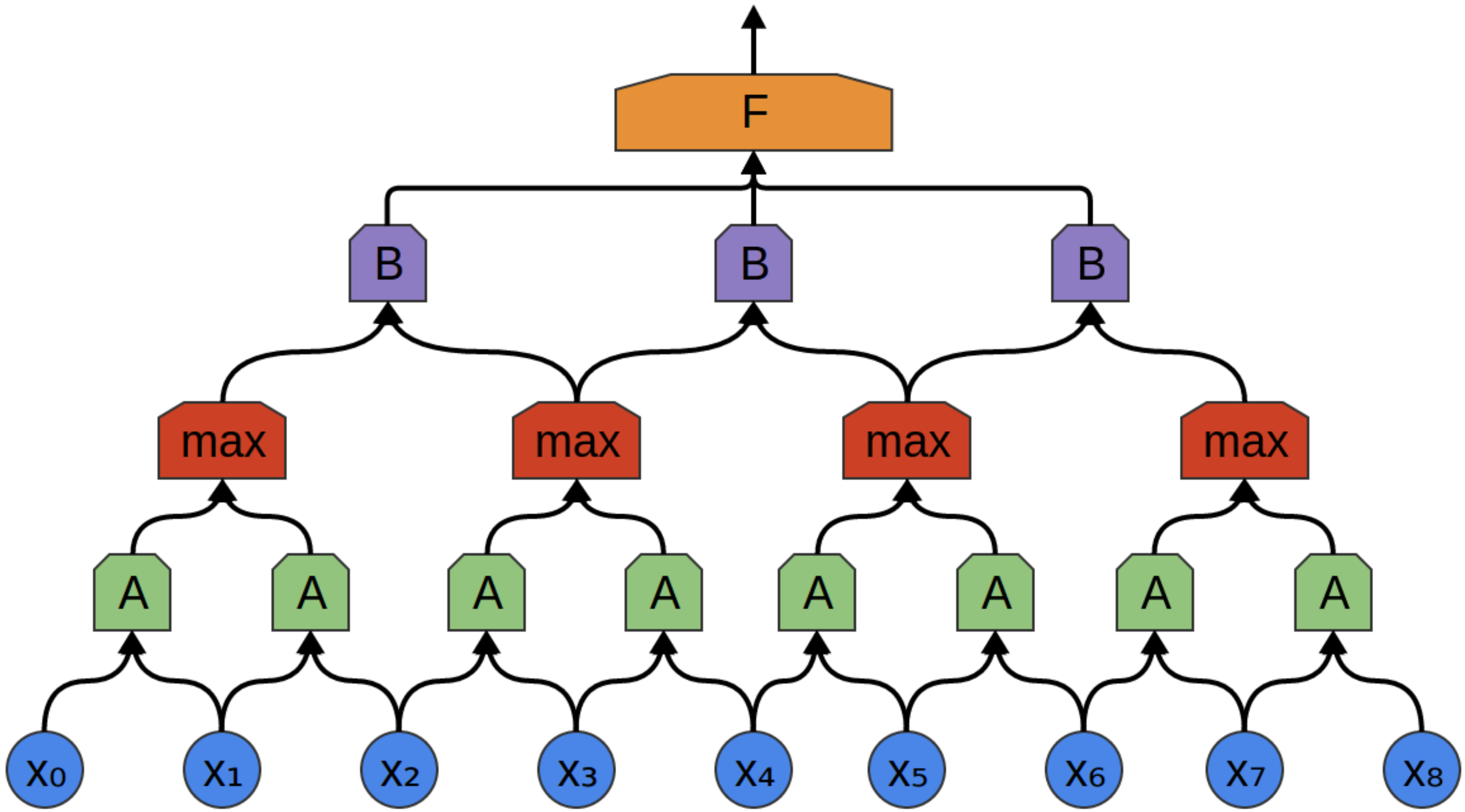
Notice the parameter sharing

Convolutional Neural Network



Convolution layers can be stacked

Convolutional Neural Network

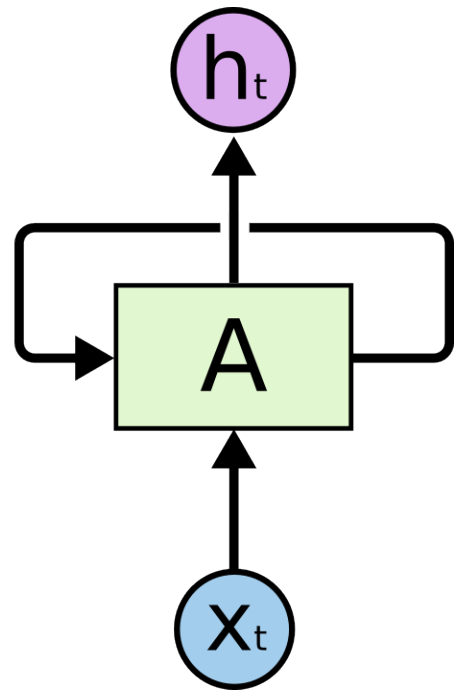


Pooling layer to further reduce complexity

Convolutional Neural Network

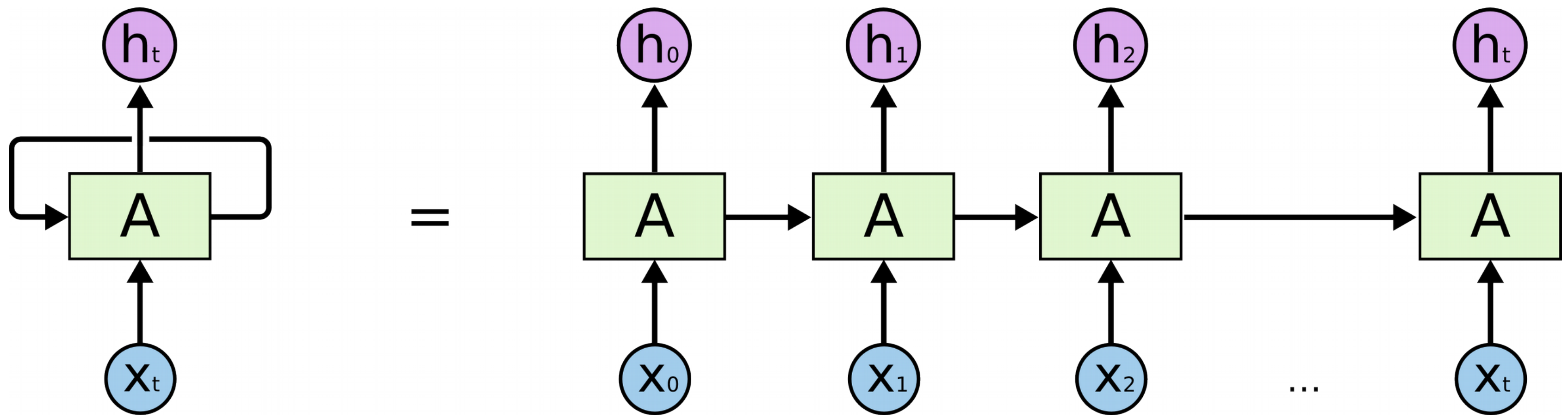
Applied to Natural Language Processing, CNNs can learn features of multiword expressions and phrases.

Recurrent Neural Network



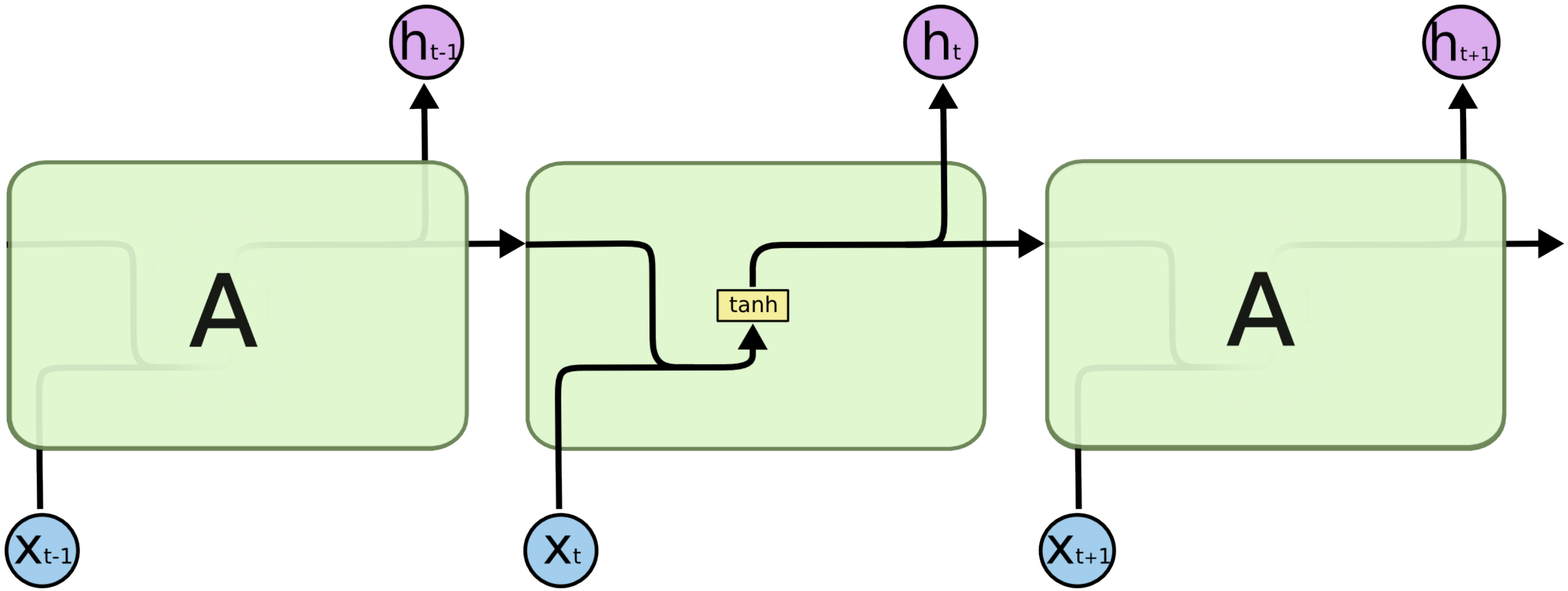
Neural Network with loops

Recurrent Neural Network



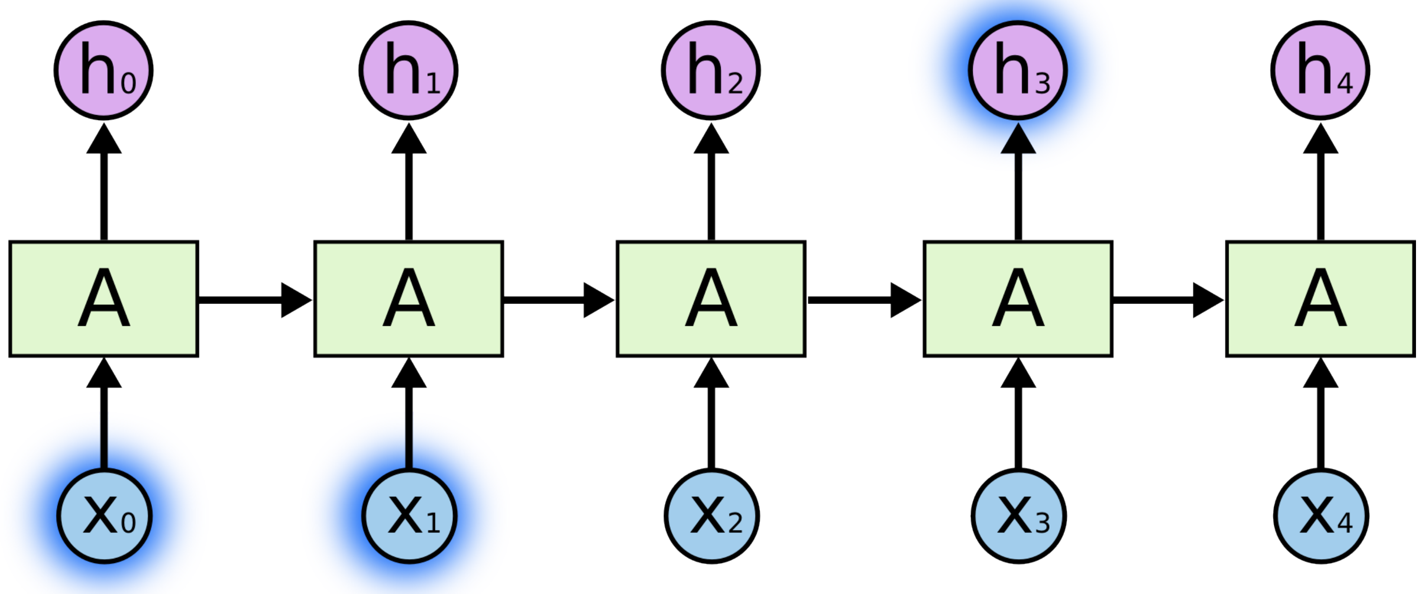
RNNs work on time sequences
(like, e.g., a sentence)

Recurrent Neural Network



Inside the RNN unit

Recurrent Neural Network



I grew up in Naples... I speak fluent Neapolitan

Recurrent Neural Network

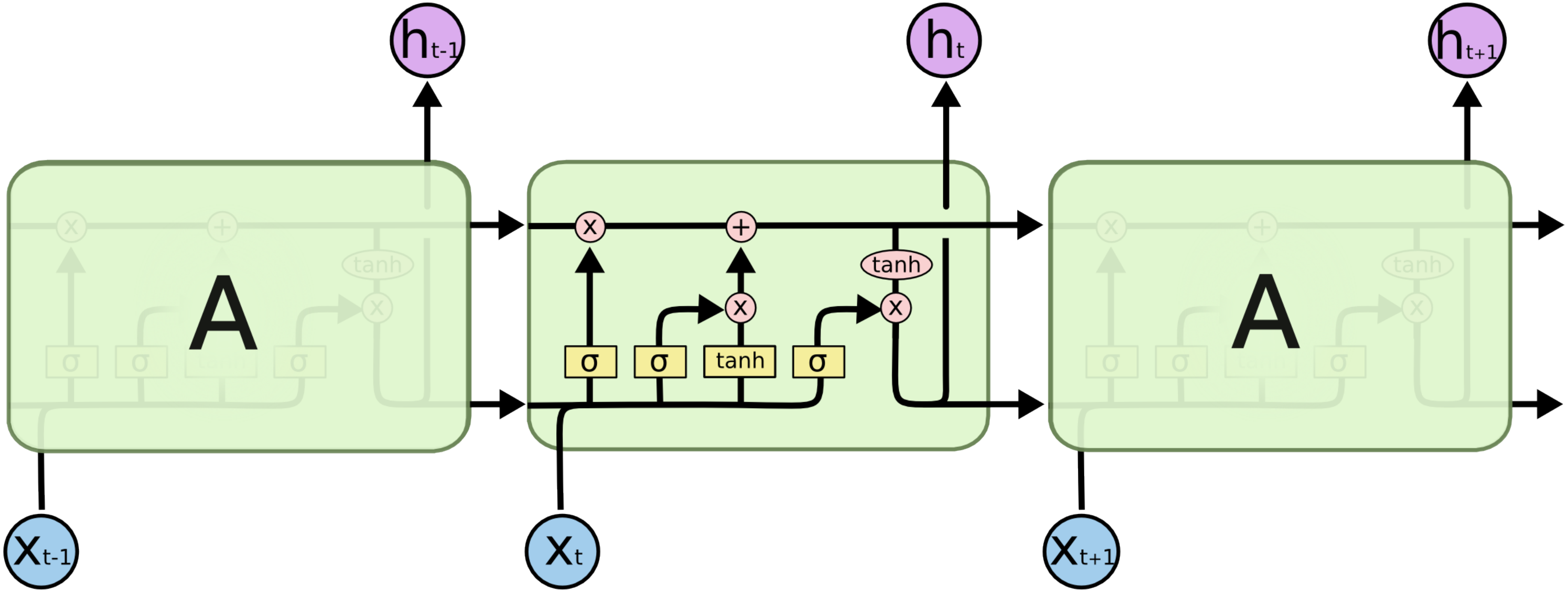
Applied to Natural Language Processing, RNNs can learn features from the order of constituent words, syllabes, characters, phonemes

Long Short-term Memory Network

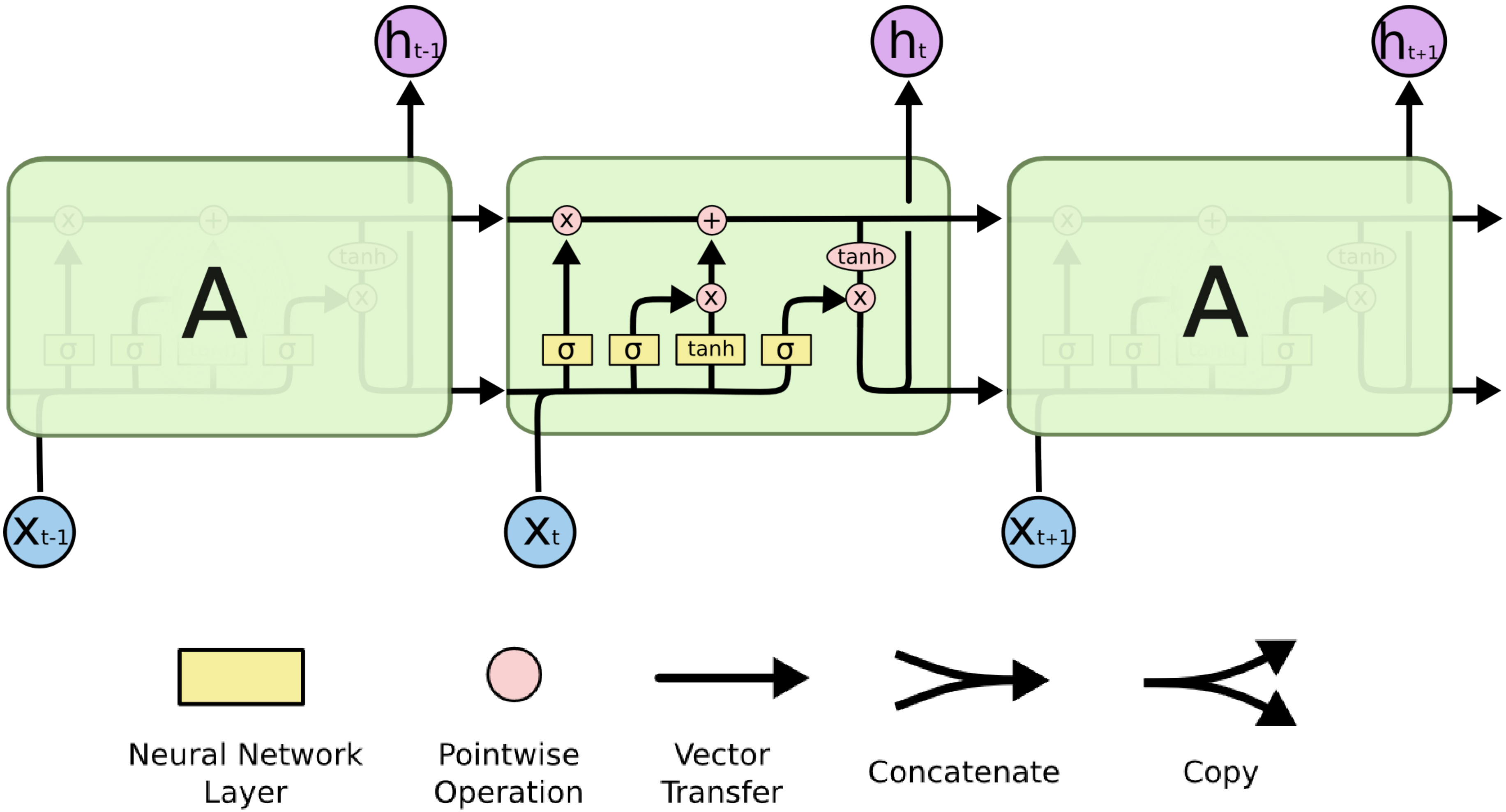
Long-term dependencies are important, but they can be long → memory problem (vanishing gradient)

Solution?

Long Short-term Memory Network

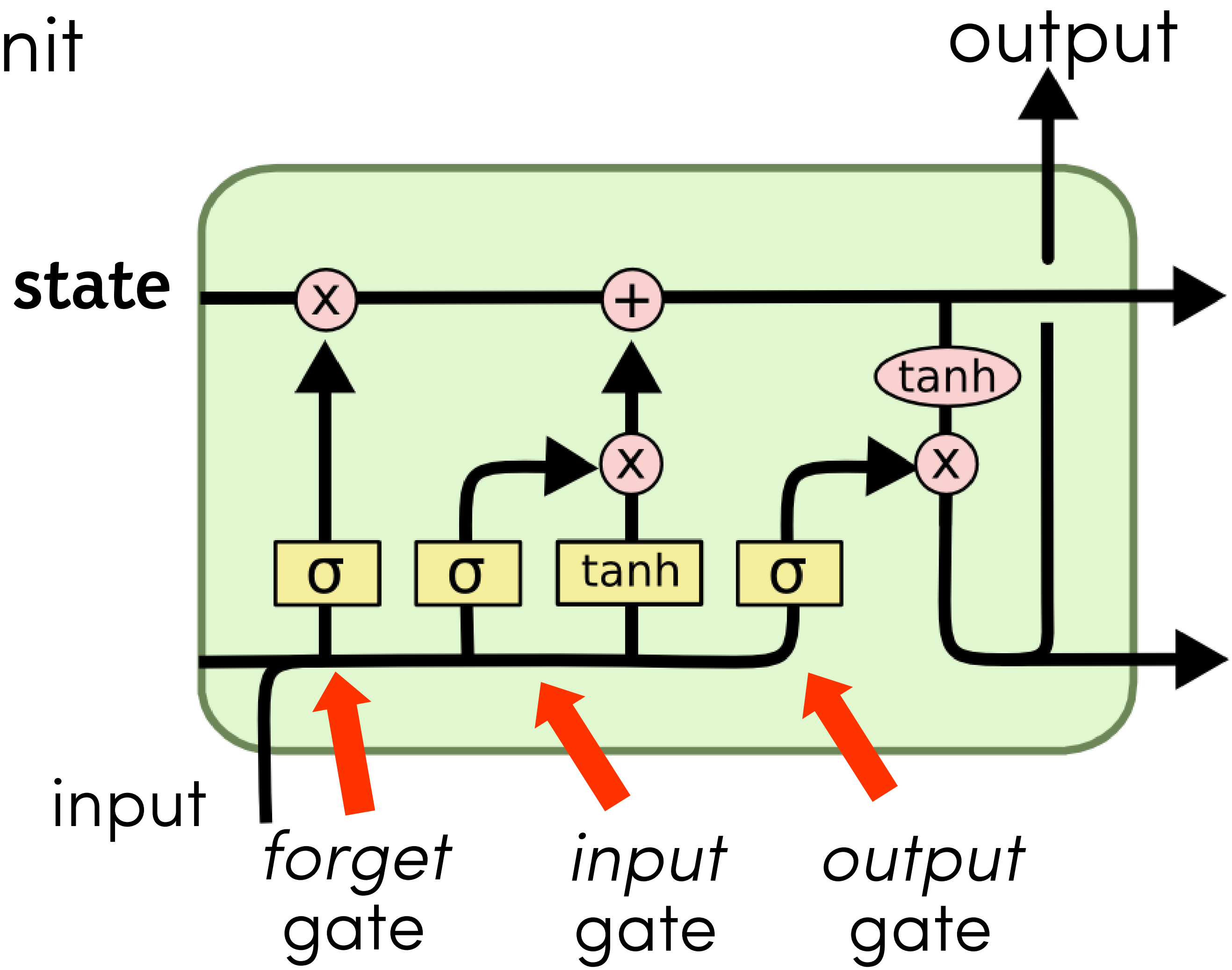


Long Short-term Memory Network



Long Short-term Memory Network

LSTM unit

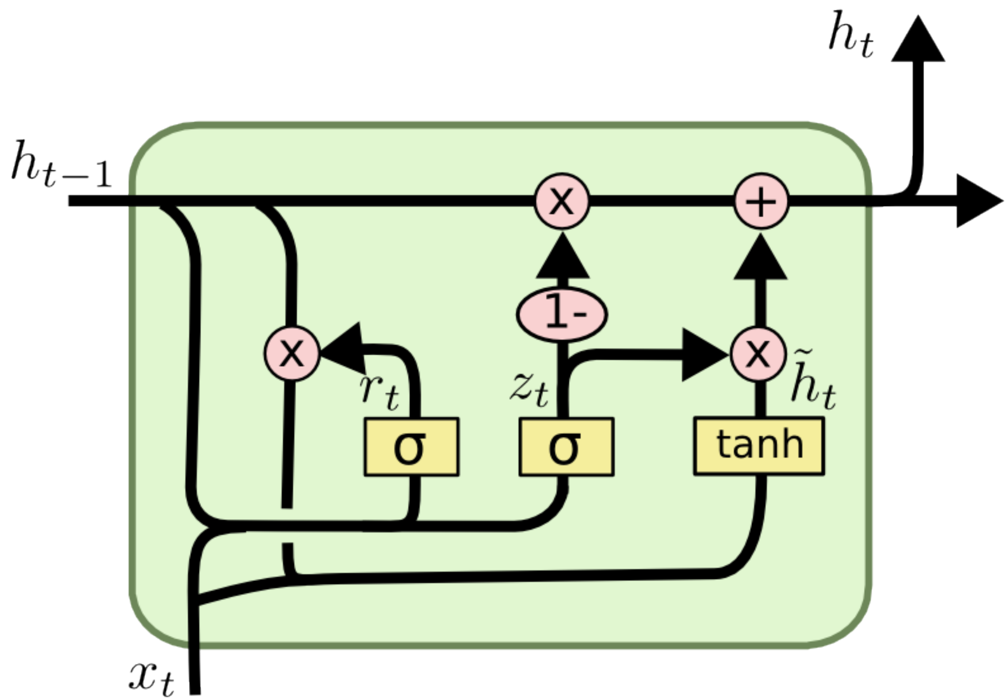


Long Short-term Memory Network

LSTM and their bidirectional variant (BiLSTM) are the state of the art in most NLP tasks

Often paired with pre-trained word embeddings

Gated Recurrent Unit



Simpler version of LSTM
combines the forget and input gates
merges the cell state and hidden state

More deep models

New architectures, variants, embeddings, preprocessing, benchmarks, ... every day

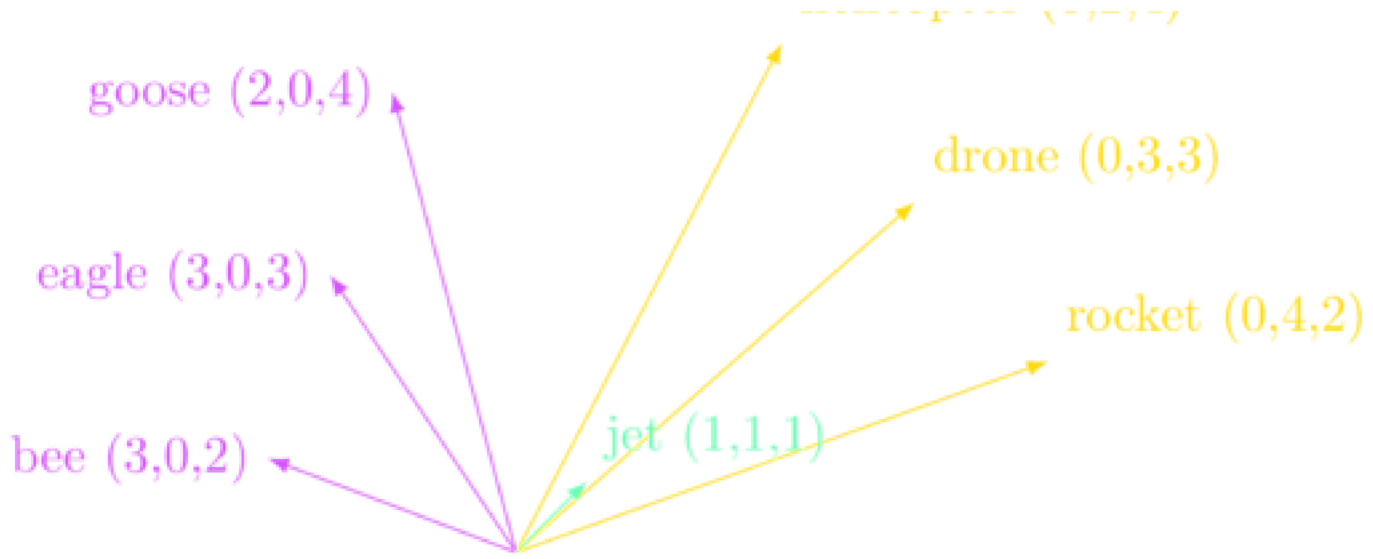
https://modelzoo.co

http://www.asimoviinstitute.org/neural-network-zoo

Word Embeddings

High-dimensional representations of words

Based on the distributional hypothesis
(Harris, 1954; Firth, 1957)



Word Embeddings

Originally based on counting co-occurrences
(Latent Semantic Analysis, Random Indexing, ...)

Recently based on predicting co-occurrences
(word2vec, GloVe)

More and more context-dependant
(ELMo, Google's BERT)

Overfitting

Dropout

delete random connections → better generalization

Regularization

learn less → learn better

$$L(x, y) = \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2$$

where $h_\theta x_i \ = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3 \ + \theta_4 x_4^4$

$$L(x, y) \equiv \sum_{i=1}^{n} (y_i \ - h_\theta(x_i)\ )^2 \boxed{+ \lambda \sum_{i=1}^{n} \theta_i^2}$$

Usual techniques

more data, smaller model, plot the learning curve

Attention

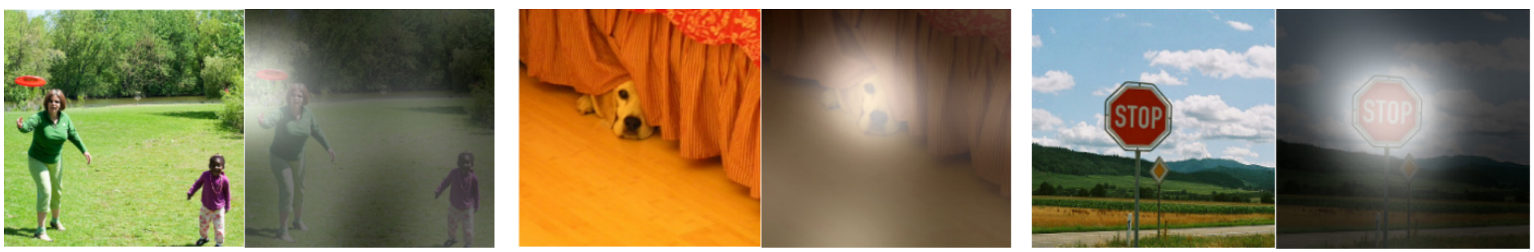
A neural attention mechanism equips a neural network with the ability to focus on a subset of its inputs

Extra vector between layers:

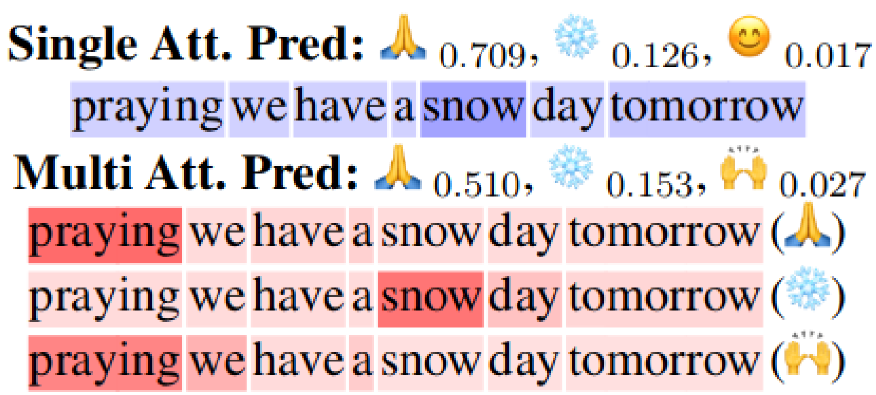
$$a=f_{\phi}(x)$$
$$g=a\odot z$$

Attention

Attention mechanisms compute a mask which is used to multiply features



A woman is throwing a frisbee in a park. A dog is standing on a hardwood floor. A stop sign is on a road with a mountain in the background.



Software

General-purpose:

- Tensorflow: first-level API by Google
- Theano: first-level, open-source
- Torch: GPU-oriented, open-source
- Keras: Python, high-level based on Tensorflow

Application-specific: word2vec, gensim, sklearn, ... (many many more)

Of course, write your own! (with GNU/Octave, if you dare)

Resources

Books, e.g.:

- *Neural Network Methods for Natural Language Procesing*, Yoav Goldberg (2017)
- *Deep Learning*, Ian Goodfellow, Yoshua Bengio, Aaron Courville, Francis Bach (2016)

Courses:

- *Machine Learning* by Andrew Ng on Coursera
- *Deep learning specialization* on coursera

Limitations of Deep Learning

Deep learning is machine learning.
Machine learning needs data.
Data needs human judgment.

Deep learning does not solve overfitting.

Deep learning is not automatically AI.

Practice

https://www.mathworks.com/help/deeplearning/examples/deep-learning-speech-recognition.html