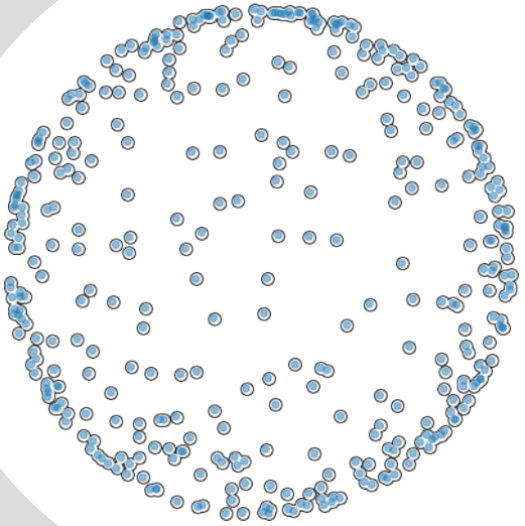
WiSe 2024/25

Machine Learning for Data Science

Lecture by G. Montavon

FREIE UNIVERSITÄT BERLIN



Lecture 11b

Neural Networks (cont.)

Outline

Neural Networks for Images

- The Neocognitron
- Convolutional Neural Networks
- Applications

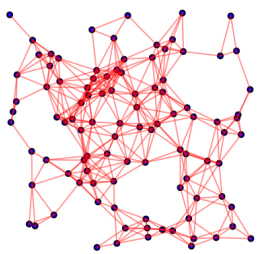
Neural Networks for Text

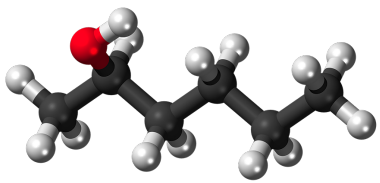
- Vector representation of text
- Recursive Neural Networks (RecNNs)
- Other architectures

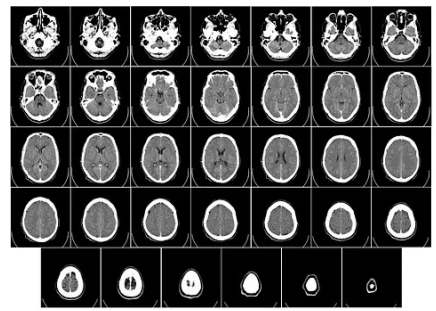
Graph Neural Networks

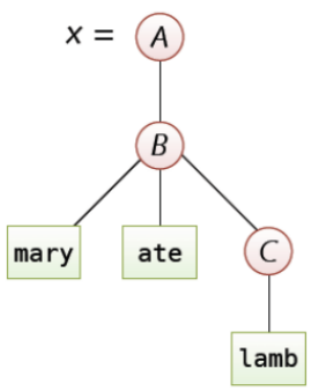
- Structure of a Graph Neural Network
- Applications
- Relation to CNNs and RecNNs.

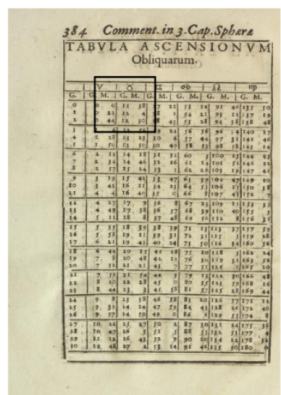
Today's Lecture: NNs for Structured Data





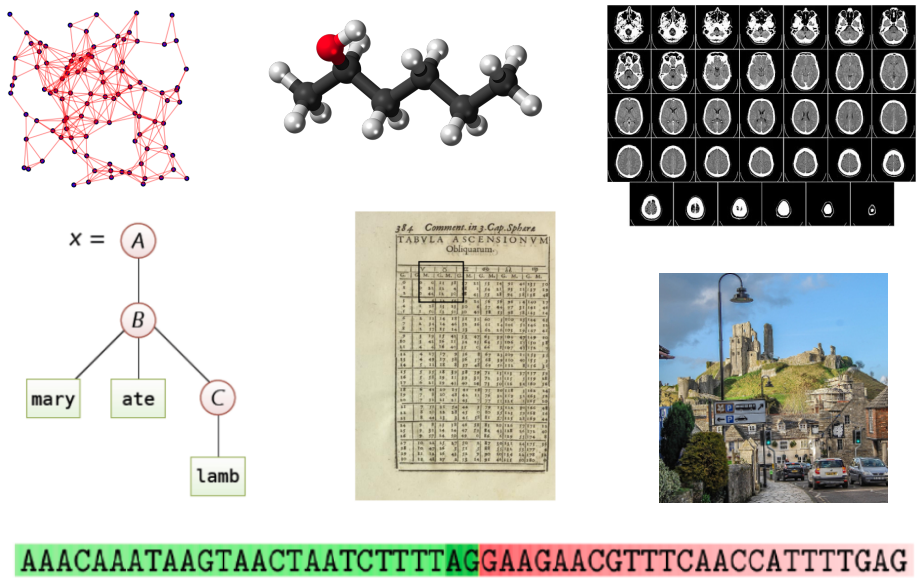






AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG

Why NNs for Structured Data?



Two Goals:

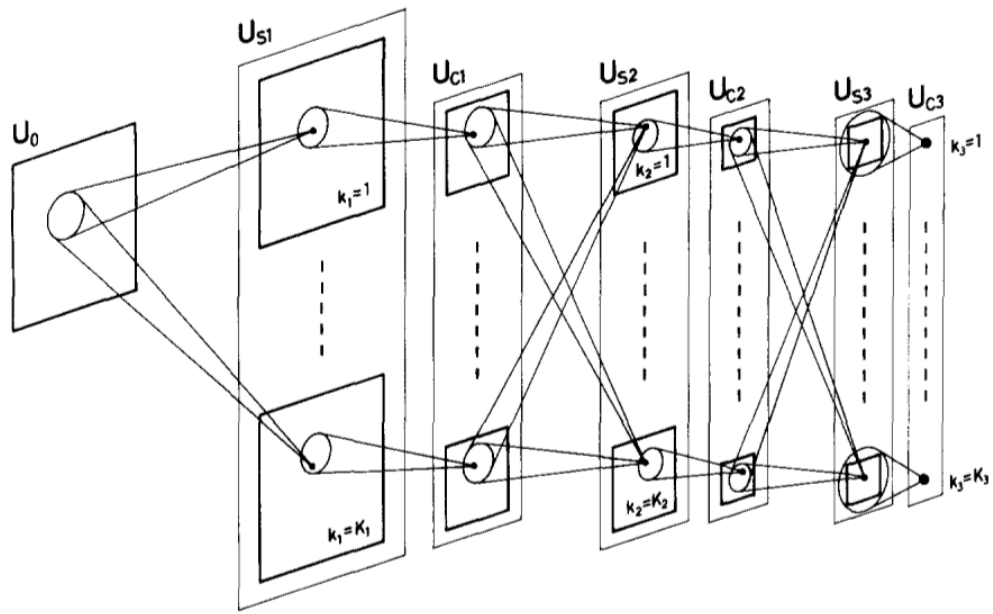
- *Technical:* Being able to process inputs that are not real-valued and cannot be represented as a fixed-size vector, e.g. text, DNA sequences.
- *Statistical:* Structured data (e.g. images) often come with specific invariances (e.g. translation invariance), that we would like the model to capture in order to generalize well.

Part 1

Neural Networks for Images

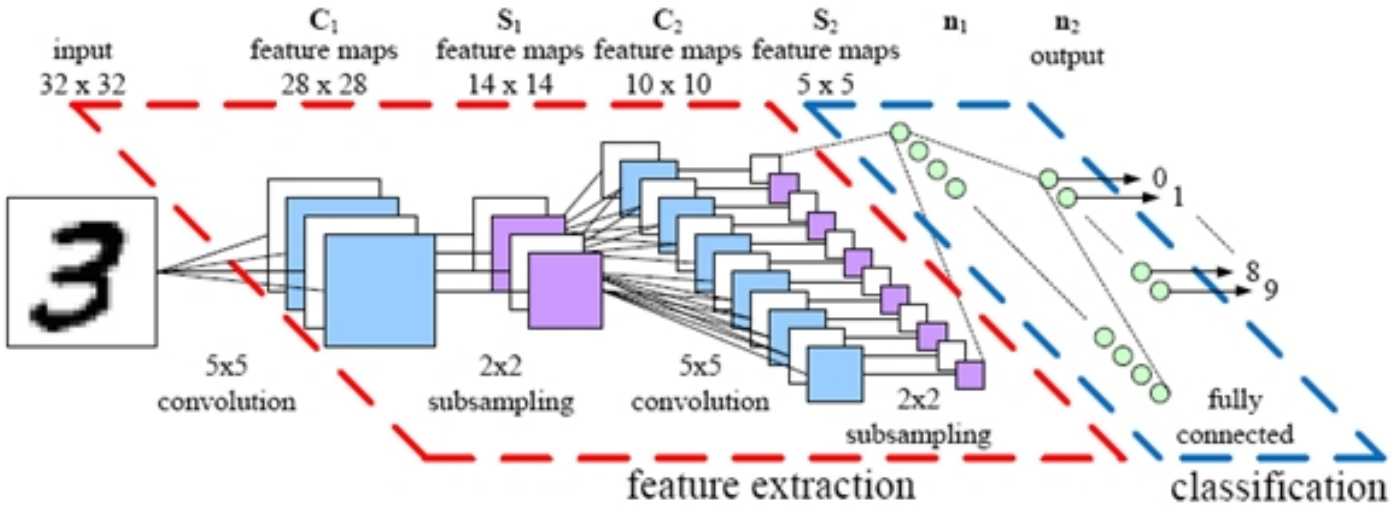
The Neocognitron (1979)

The Neocognitron [2] is an architecture for representing images, and which is designed in a way that the produced representation is invariant to a translation of the input image.



The Neocognitron consists of an alternation of 'simple cells' (convolutions) and 'complex cells' (pooling), which progressively build the desired representation. It was a precursor of modern convolutional neural network architectures.

Convolutional Neural Nets (1989, 1998, 2012)



- The Convolutional Neural Network (CNN) [4] is also composed of convolution and pooling layers, and comes with a way of backpropagating the error through the convolution/pooling layers.
- CNNs are nowadays state-of-the-art (by a wide margin) on many image recognition tasks (e.g. ILSVRC 2012).
- CNN training is resource-hungry. Often, people use pretrained models.

Standard vs. Convolutional Neural Networks

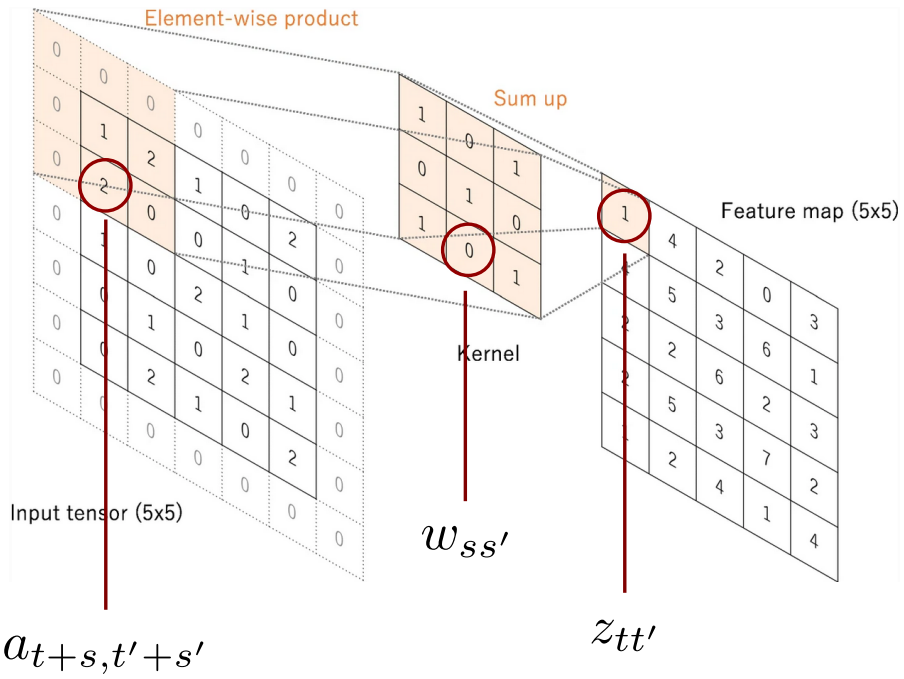
Standard neural network:

- Neurons and weights are scalars, i.e. $a_i, w_{ij} \in \mathbb{R}$.
- Mapping between two layers is a sum of products: $z_j = \sum_i w_{ij} a_i + b_j$

Convolutional neural network:

- Neurons and weights are 2D arrays, i.e. $\boldsymbol{a}^{(i)} \in \mathbb{R}^{W \times H}$, $\boldsymbol{w}^{(ij)} \in \mathbb{R}^{W' \times H'}$.
- Mapping between two layers is a sum of 2D cross-correlations: $\boldsymbol{z}^{(j)} = \sum_i \boldsymbol{w}^{(ij)} \star \boldsymbol{a}^{(i)} + \boldsymbol{b}^{(j)}$
- Convolution layers are interleaved with pooling layers in order to progressively reduce the spatial resolution and simultaneously increase the number of neurons.

Understanding the CNN (1)



adapted from Yamashita et al. 2018
Convolutional neural networks: an overview [...]

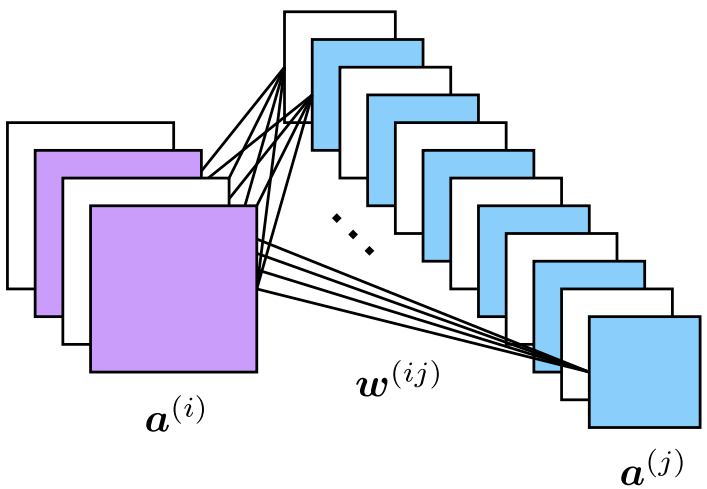
Step 1: The Cross-Correlation

- A *cross-correlation* slides a kernel (or filter) through the whole input tensor and records responses at each location.

$$z_{tt'} = [w \star a]_{tt'}$$
$$= \sum_{s=0}^{2} \sum_{s'=0}^{2} w_{ss'} \cdot a_{t+s,t'+s'}$$

- Zero-padding can be applied to the input signal in order to produce an output signal of same size as the input.

Understanding the CNN (2)



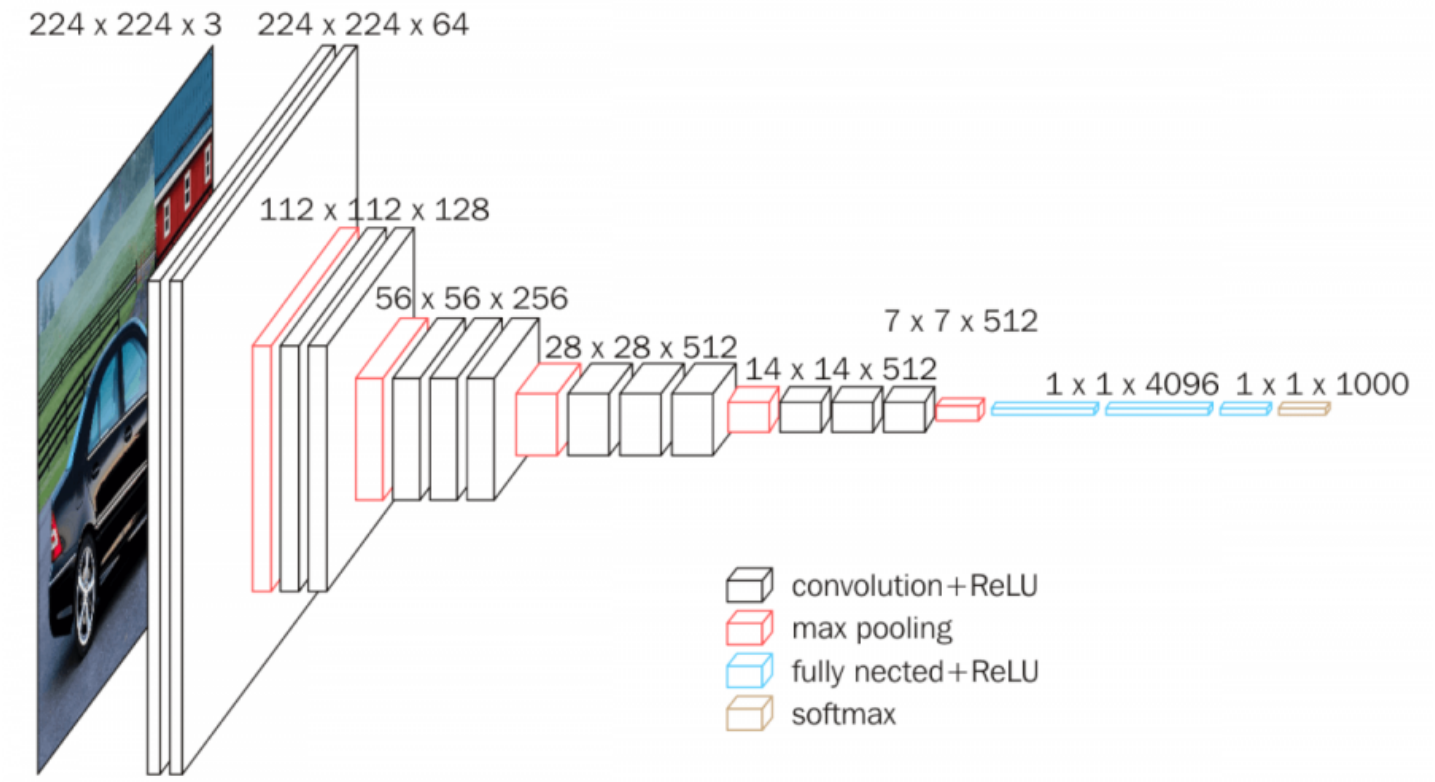
Step 2: The Convolution Layer

- A *convolution layer* has its output given by a sum of cross-correlations associated to each incoming neurons:

$$\boldsymbol{a}^{(j)} = g\big(\textstyle\sum_i \boldsymbol{w}^{(ij)} \star \boldsymbol{a}^{(i)}\big)$$

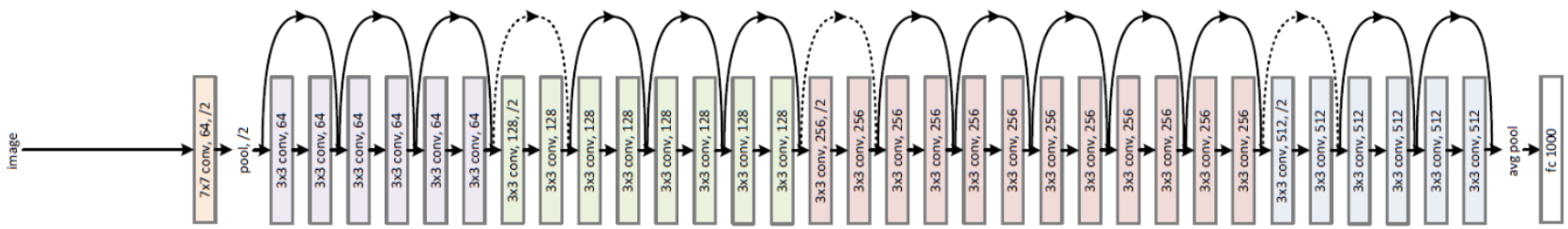
- The number of trainable parameters in a convolution layer is given by (# input neurons $\times$ # output neurons $\times$ filter size).

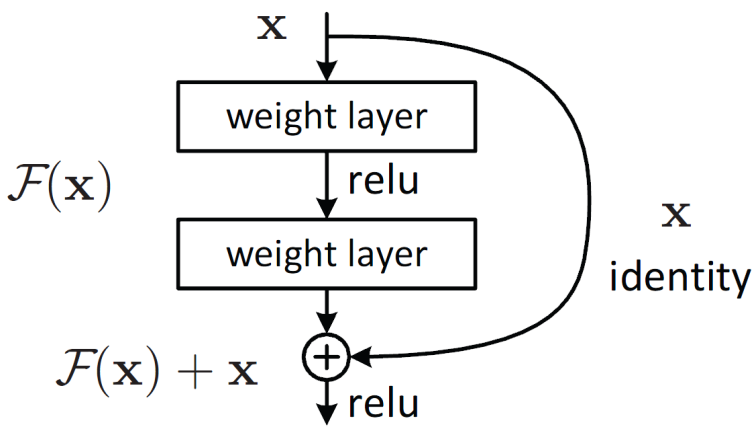
The VGG-16 Architecture



- Multiple blocks of two or three consecutive convolutions (each with kernel size 3×3). Blocks are interleaved by max-pooling layers.
- The spatial resolution is progressively reduced, which allows to use more neurons in the top layers.

The ResNet Architecture





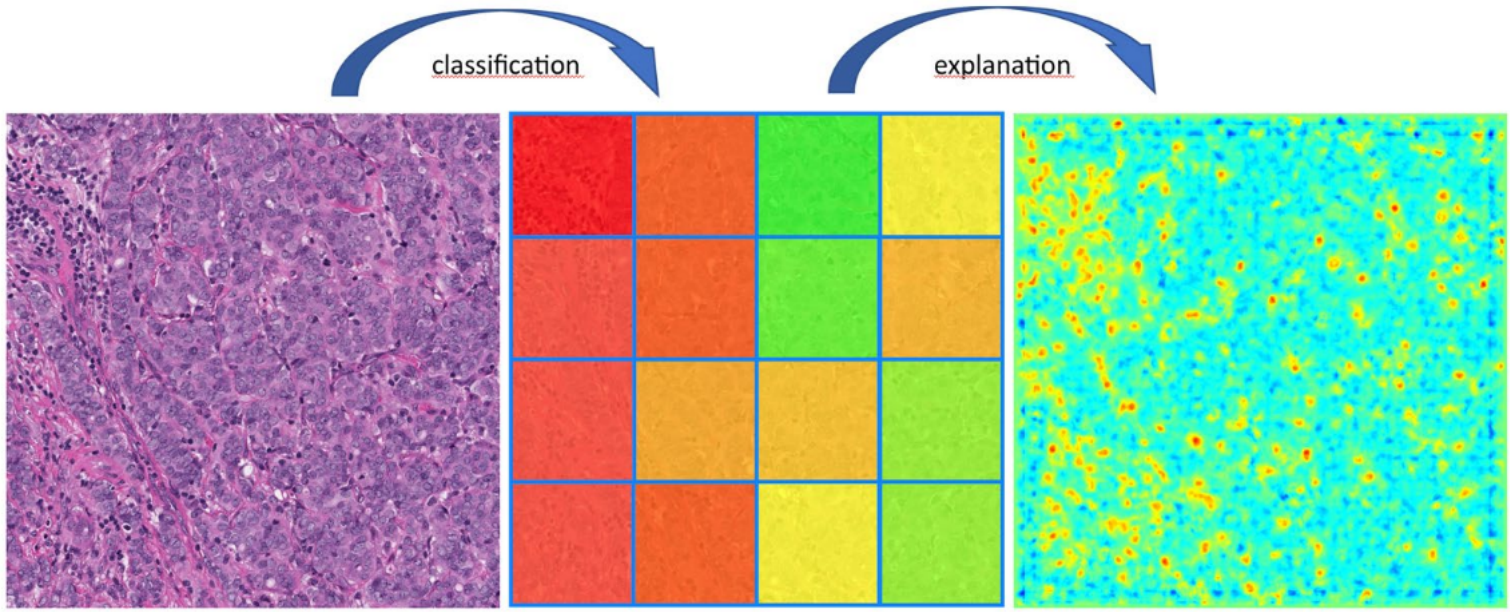
- Deeper than VGG-16 (ResNets have between 34 and a hundred of layers depending on the variants).
- Includes skip-connections (identity) every two or three layers in order to make prediction/training more stable.

Application of CNNs: ImageNet Classification

Source: Krizhevsky et al. (2012) ImageNet Classification with Deep Convolutional Neural Networks

- Ground-truth prediction very often in the top-5 predicted classes (among 1000 predicted classes!).
- Errors are often similar to errors a human could do (e.g. similar class, or co-located pattern of another class on the same image).

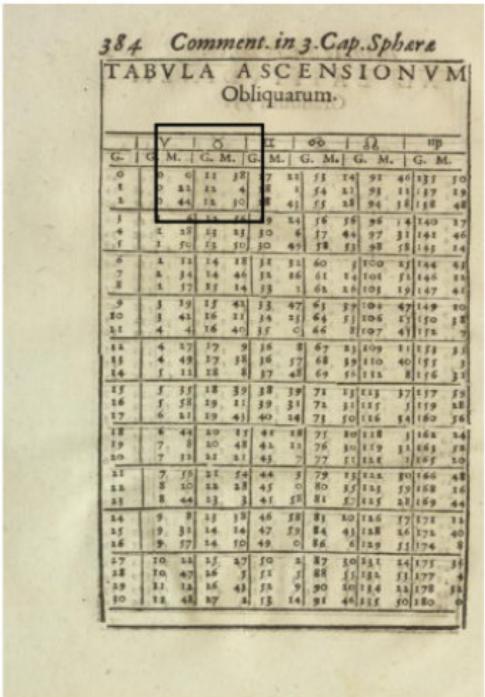
Application of CNNs: Histopathological Images



Klauschen et al. (2018) Scoring of tumor-infiltrating lymphocytes: from visual estimation to machine learning

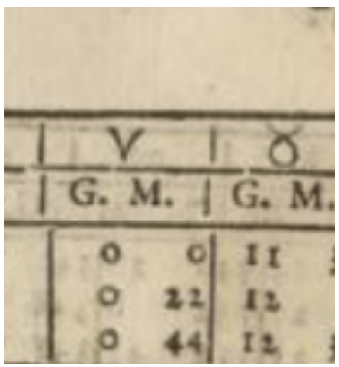
- Histopathological images can be very large ($\sim$ gigapixel). Convolutional neural networks are trained and applied on small patches.
- Patch-level predictions can be sharpened spatially by using pixel-wise explanation techniques such as LRP (cf. Lecture 12).

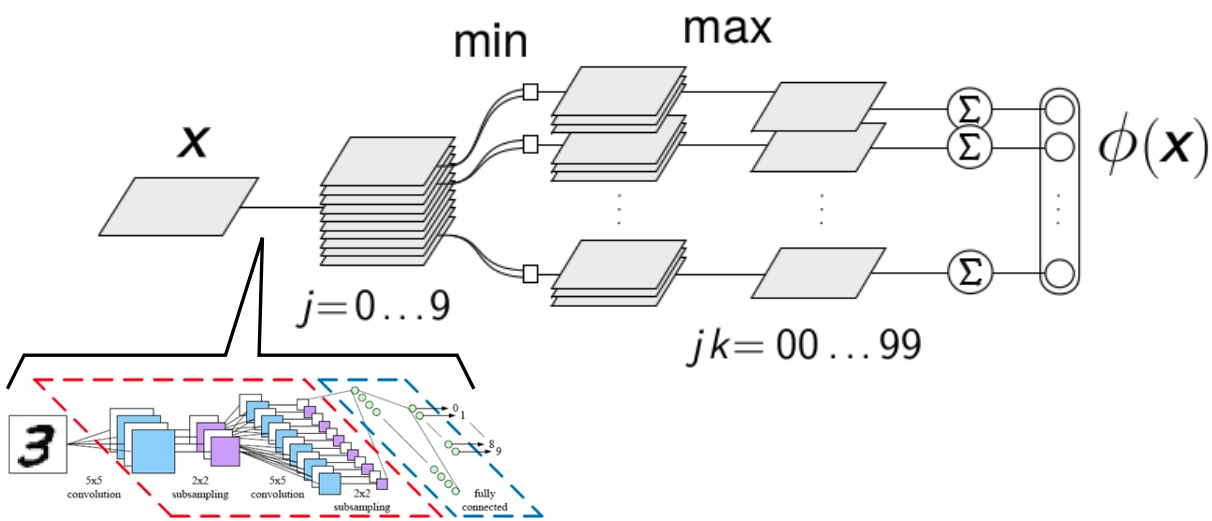
Specialized Architectures for Table Images



- Predicting similarity between scanned books content (e.g. from the early-modern era), is a recent direction for making historical inferences.
- The type of data we would like to predict is very specialized, and standard convnet architectures for image recognition do not work well at predicting this data.
- We need ad-hoc neural network architectures that take into consideration what types of features are predictive of table similarity, and also that make an efficient usage of data and labels.

The ‘Bigram Network’ [1]





- Compound architecture composed of a digit recognizer followed by soft-logic operations to extract histogram of digit bigrams.
- Only the digit recognizer is trained, the rest is hard-coded. This avoids the necessity of having full-table labels.

Part 2

Neural Networks for Text

Structured Neural Networks for Text

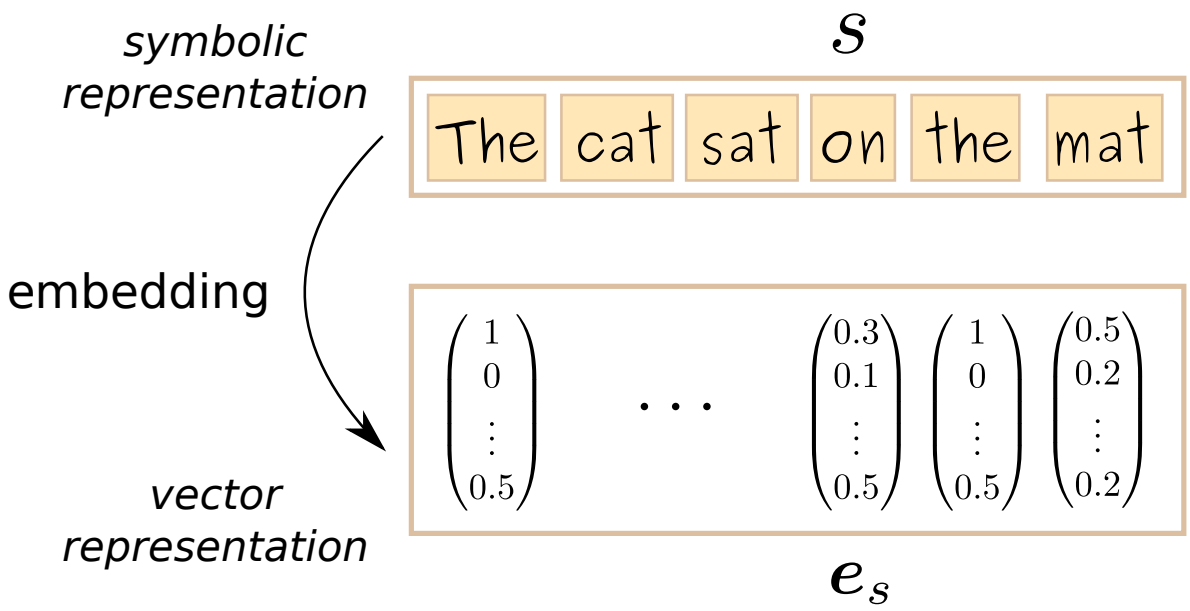
Image classification

- Images are arrays of pixels.
- Pixels p of an image are given by their RGB components (i.e. $x_p \in [0,1]^3$). Therefore, they can be readily given as input to the neural network.

Text classification

- Texts are collections of words.
- Words w are abstract symbols that do not have a numerical representation. $\Rightarrow$ They need to be embedded into one such representation so that they can be given as input to a neural network.

Vector Embedding of a Sentence



Types of Vector Embedding

1. **One-hot encoding:** Represent the word with indicator functions:

$$e[w] = \left(1_{\{w=\text{'a'}\}}, 1_{\{w=\text{'able'}\}}, 1_{\{w=\text{'about'}\}}, \dots\right) \in \mathbb{R}^{\#\text{words}}$$

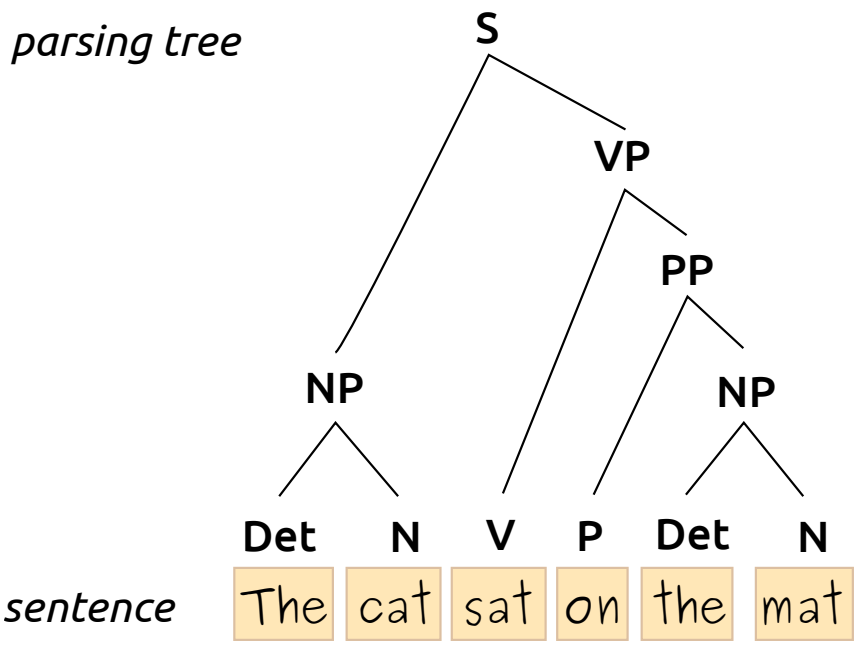
2. **kPCA embedding:** Structured kernel $k(w, w')$ measures similarity between words (e.g. co-occurrence or lexical similarity). Embedding is obtained by diagonalizing the Gram matrix: $K = U\Lambda U^\top$, and defining:

$$e[w] = U_{w,:d} \odot (\lambda_1^{0.5}, \dots, \lambda_d^{0.5}) \in \mathbb{R}^d$$

3. **Learned embedding (e.g. Word2Vec):** Start with a randomly initialized embedding $e[w] \in \mathbb{R}^d$ and learn the embedding on the supervised task directly or on some related unsupervised task:

$$e[w] \leftarrow e[w] - \gamma \cdot \partial\mathcal{E}/\partial e[w].$$

Recursive Neural Networks [5]

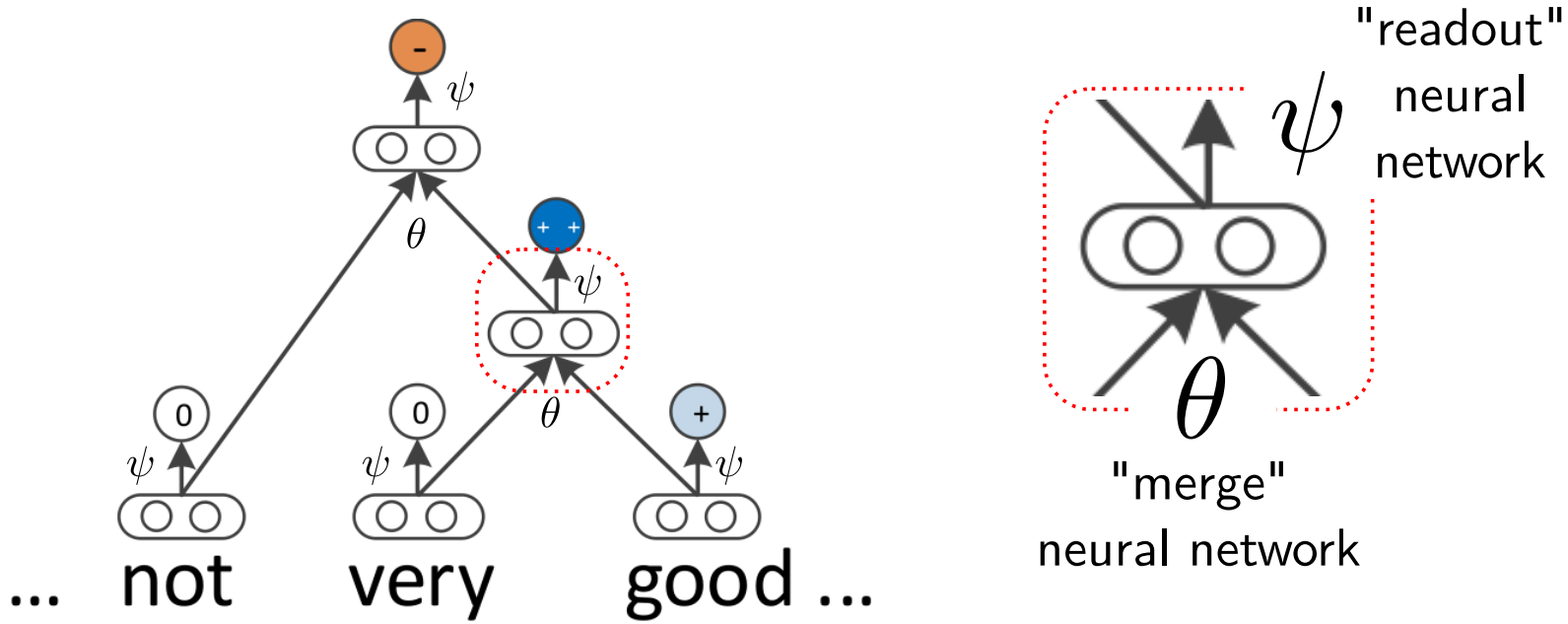


Ideas:

- Make use of the parsing tree of a sentence to produce a better text classification model.
- Build the representation by recursively applying a neural network from the leaves of the tree up to the root.

RecNNs for Sentiment Analysis

The recursive neural network (RecNN) consists of two subnetworks: "merge" and "readout". Both are used at multiple locations in the parse tree.

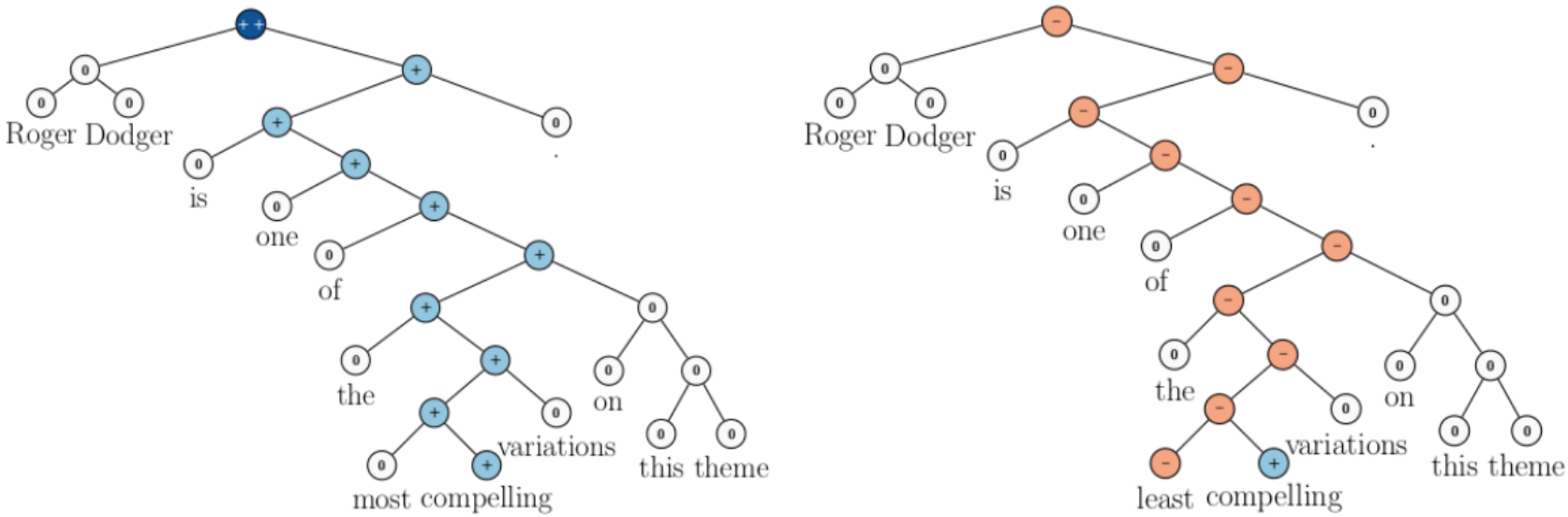


Based on Socher et al. (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

It can be applied to any sentence that comes with a binary parse tree.

RecNNs for Sentiment Analysis

Applying the readout function on each node allows to visualize how the sentiment builds up in the recursive neural network.



Source: Socher et al. (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Other Architectures for Text

Recurrent Neural Networks (LSTMs):

- Special neural networks that process the text as a sequence.
- They receive text as input (sequences of word vectors) and can also generate text (sequence of word vectors).
- They found application e.g. for text translation.
- They do not require a parse tree associated to the sentence.

Transformers:

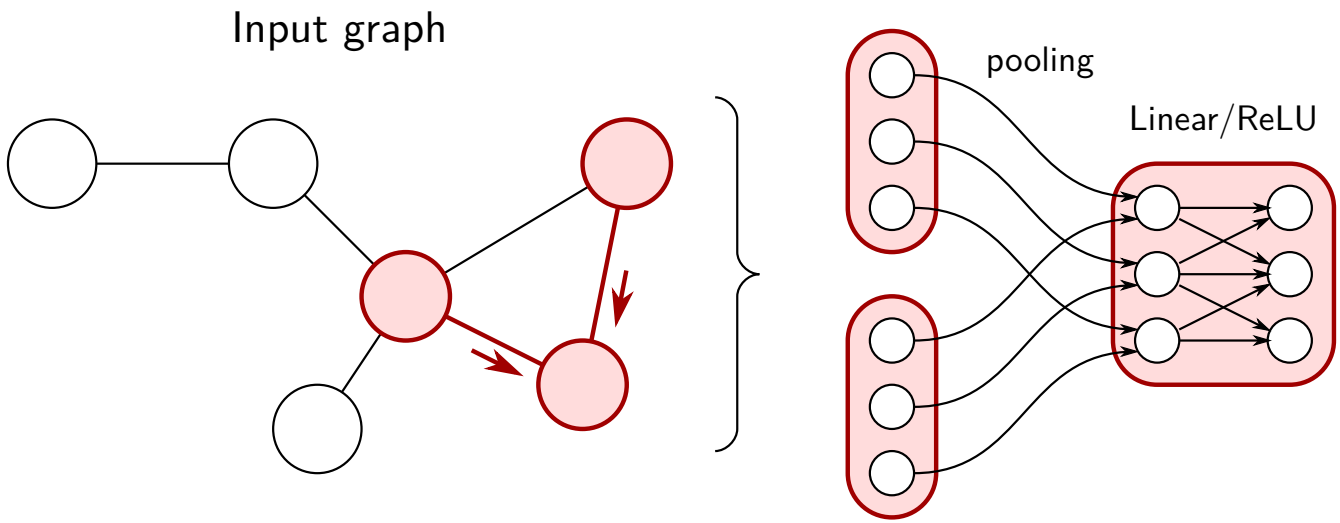
- Special neural networks that work with input sequences and facilitate the modeling of long-range dependencies in the sequence (via a mechanism called attention).
- Transformer architectures have become state-of-the-art for a broad range of text processing tasks. They are the basis of large language models such as BERT, GPT-3, etc.

Part 3

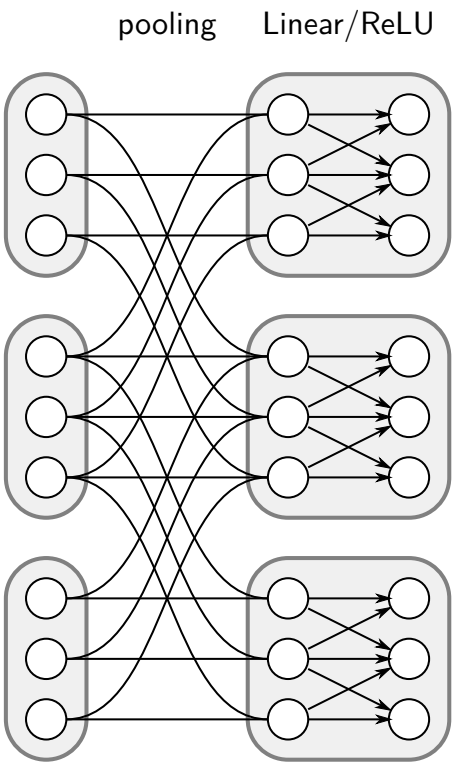
Graph Neural Networks

Graph Neural Networks

- Graph neural networks (GNNs) are networks that are specialized for classifying graphs. Graphs are general data representation, that include pixel lattices, trees, sequences as special cases.
- GNNs receive a graph as input and implement at each layer a propagation step along its edges (via a pooling function).



Graph Neural Networks



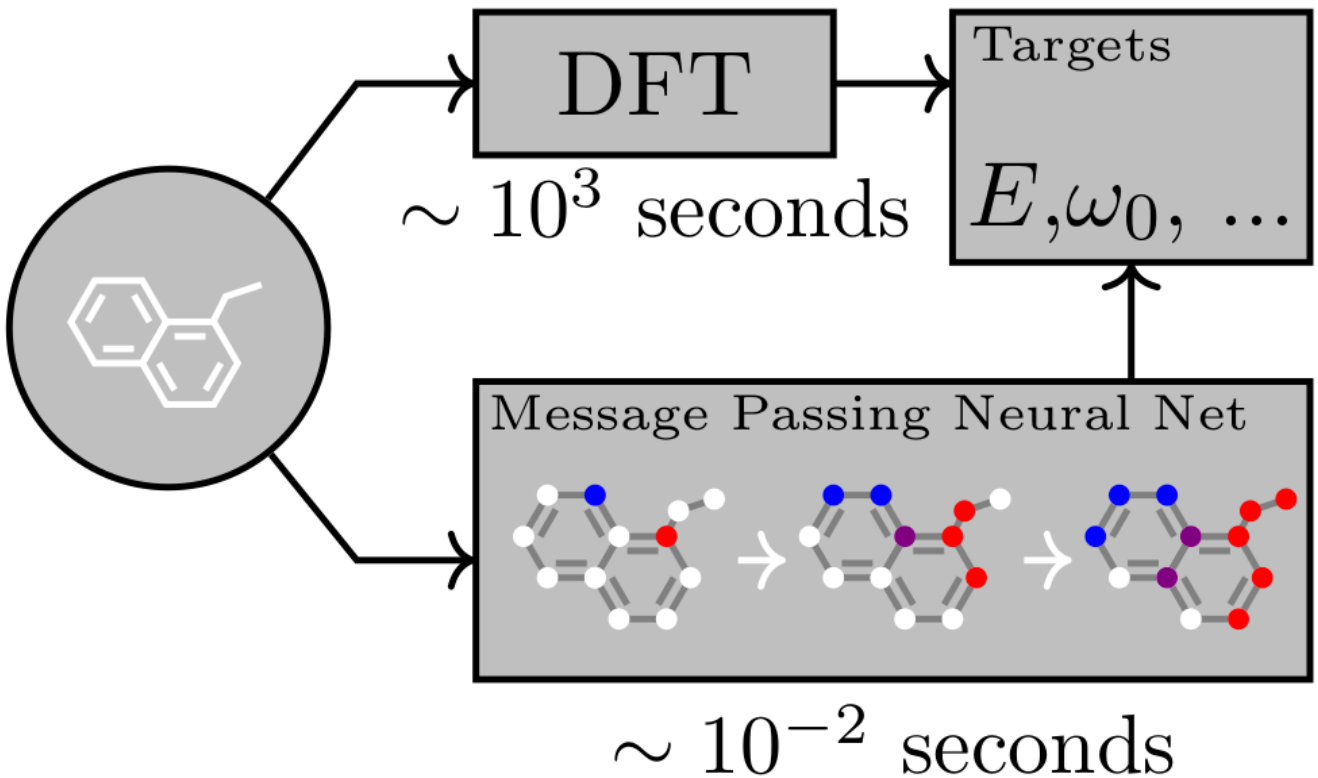
- Consider a graph of N nodes. We denote the representation of the graph at layer l as $\boldsymbol{H}_l = (\boldsymbol{h}_1^{(l)}, \ldots, \boldsymbol{h}_N^{(l)})$ a matrix of dimensions $N \times d$.
- Define $\boldsymbol{\Lambda}$ of size $N \times N$ a connectivity matrix, e.g. a normalized version of the graph adjacency matrix.
- Define $\boldsymbol{W}_l$ a matrix of size $d \times d'$, which we use to map nodes representations from one layer to another.
  The graph convolutional network [3] defining the following layer-wise mapping:

$$\boldsymbol{A}_l = \boldsymbol{\Lambda}\boldsymbol{H}_l \qquad \text{(pooling)}$$
$$\boldsymbol{H}_{l+1} = \rho(\boldsymbol{A}_l\boldsymbol{W}_l) \qquad \text{(linear/ReLU)}$$

  where ρ is a ReLU function applied element-wise.

Applications of GNNs

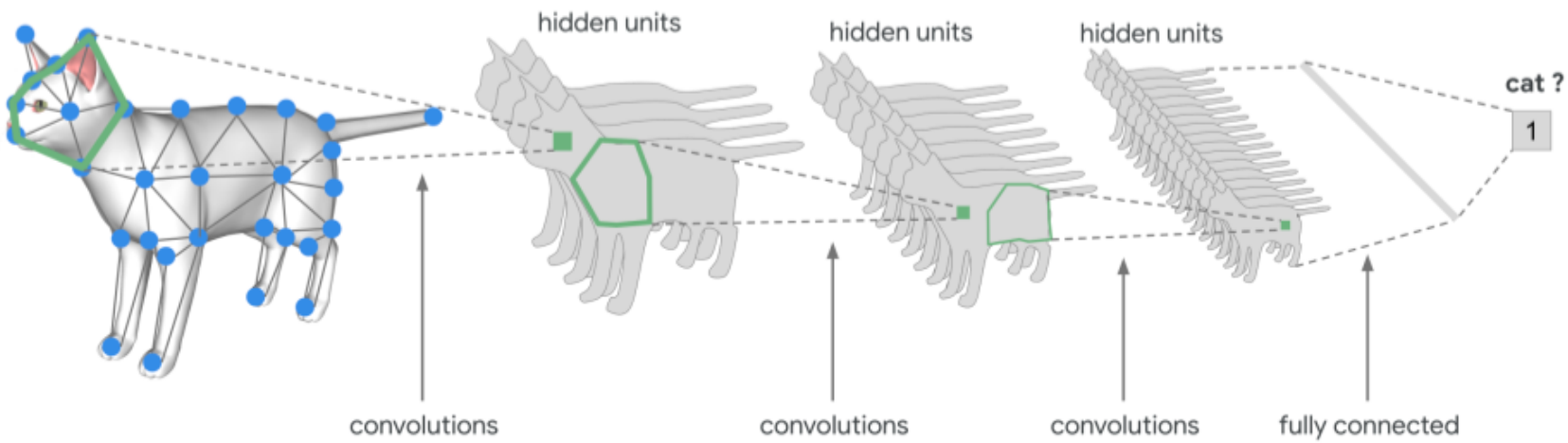
Predicting atomization energy to bypass expensive DFT simulations:



Source: Gilmer et al. (2017) Neural Message Passing for Quantum Chemistry

Applications of GNNs

Recognizing objects from meshes:



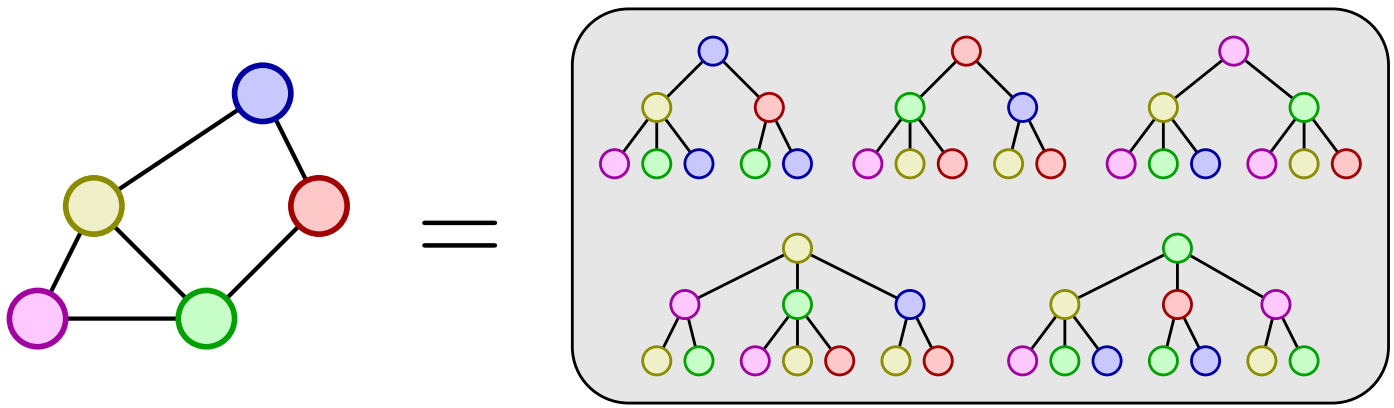
Source: https://colab.research.google.com

Relation between GNNs and Recursive Nets

Graph neural networks models can be expanded as recursive neural networks applying to trees. Each tree represents a search of depth L into the graph starting from a given node, and where L is the number of layers of the graph neural network.

graph neural network

collection of recursive networks



Relation between GNNs and Convolution Layers

Observation:

- The convolution layer can be seen as a special case of a graph neural network where the graph is a two-dimensional lattice.

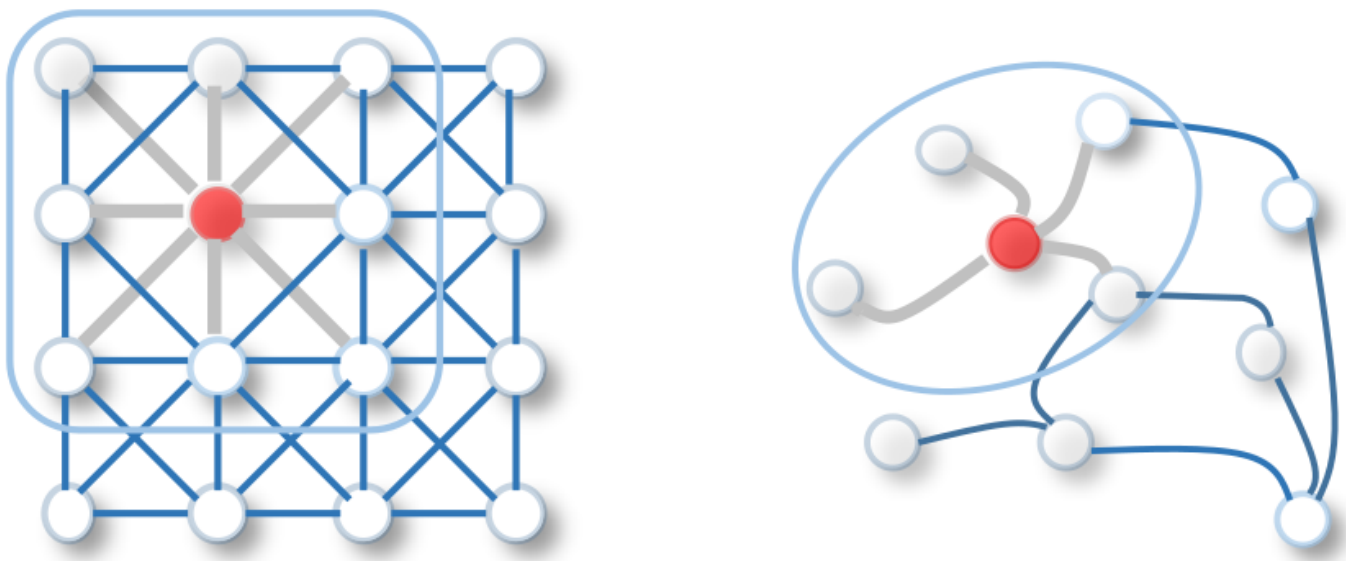


Image source: Wu et al. 2019, A Comprehensive Survey on Graph Neural Networks

Summary

Summary

- Structured networks are a generalization of standard neural networks that allows to handle various real-world data (e.g. images, text, graphs). In terms of applications, an analogy can be made between structured neural networks and structured kernels.
- Convolutional neural networks are a widely used approach for predicting image data. They operate by progressively extracting more complex and useful features while at the same time filtering out exact pixel-wise information.
- Recursive neural networks enable the processing of data (e.g. text) by following the underlying graph structure of that data (e.g. its parse tree, encoding the grammatical structure).
- Graph neural networks generalize the application of neural networks to any graph-structured input (including trees and pixel lattices). Convolutional neural networks and recursive neural networks can be seen as a special case of graph neural networks.

References

O. Eberle, J. Büttner, F. Kräutli, K. Müller, M. Valleriani, and G. Montavon.

Building and interpreting deep similarity models.
CoRR, abs/2003.05431, 2020.

K. Fukushima.

Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.
Biological Cybernetics, 36(4):193–202, 1980.

T. N. Kipf and M. Welling.

Semi-supervised classification with graph convolutional networks.
In *ICLR (Poster)*. OpenReview.net, 2017.

Y. LeCun.

Generalization and network design strategies.
In *Connectionism in Perspective*. Elsevier, 1989.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts.

Recursive deep models for semantic compositionality over a sentiment treebank.
In *EMNLP*, pages 1631–1642. ACL, 2013.