– IN5550 –
Neural Methods in Natural Language Processing

CNNs, Part 1: Introduction and background

Erik Velldal

Language Technology Group (LTG)
University of Oslo

Example text classification tasks

Positive or negative polarity?

- *The food was expensive but hardly impressive.*
- *The food was hardly expensive but impressive.*

Example text classification tasks

Positive or negative polarity?

- *The food was expensive but hardly impressive.*
- *The food was hardly expensive but impressive.*

- Strong local indicators of class,
- some ordering constraints,
- but independent of global position.
- In sum: a small set of relevant n-grams could provide strong features.

Example text classification tasks

Positive or negative polarity?

- *The food was expensive but hardly impressive.*
- *The food was hardly expensive but impressive.*

- Strong local indicators of class,
- some ordering constraints,
- but independent of global position.
- In sum: a small set of relevant n-grams could provide strong features.

Many text classification tasks have similar traits:...

- topic classification
- authorship attribution
- spam detection
- abusive language
- subjectivity classification
- question type detection ...

Rewind: MLPs + embeddings as inputs

- Embeddings have benefits over discrete feature vectors; can make use of unlabeled data + information sharing across features.
- MLPs can learn certain feature combinations, but there are limitations when it comes to learning ordering and positional information.
- We still lack power for representing sentences and documents.

Rewind: MLPs + embeddings as inputs

- Embeddings have benefits over discrete feature vectors; can make use of unlabeled data + information sharing across features.
- MLPs can learn certain feature combinations, but there are limitations when it comes to learning ordering and positional information.
- We still lack power for representing sentences and documents.
- Summing or averaging (CBOW)? Gives a fixed-length representation, but no information about order or structure.

Rewind: MLPs + embeddings as inputs

- Embeddings have benefits over discrete feature vectors; can make use of unlabeled data + information sharing across features.
- MLPs can learn certain feature combinations, but there are limitations when it comes to learning ordering and positional information.
- We still lack power for representing sentences and documents.
- Summing or averaging (CBOW)? Gives a fixed-length representation, but no information about order or structure.
- Concatenation? For longer sequences, this could easily blow up the parameter space for a fully connected layer.

Rewind: MLPs + embeddings as inputs

- Embeddings have benefits over discrete feature vectors; can make use of unlabeled data + information sharing across features.
- MLPs can learn certain feature combinations, but there are limitations when it comes to learning ordering and positional information.
- We still lack power for representing sentences and documents.
- Summing or averaging (CBOW)? Gives a fixed-length representation, but no information about order or structure.
- Concatenation? For longer sequences, this could easily blow up the parameter space for a fully connected layer.
- Bag-of-n-grams or n-gram embeddings?
- Potentially wastes many parameters; only a few n-grams relevant.
- Data sparsity issues + does not scale to higher order n-grams.

Feature extractors

- Need for specialized NN architectures that extract higher-level features:
- E.g. CNNs and RNNs
- Learns intermediate representations that are then plugged into additional layers for prediction.
- Pitch: layers and architectures are like Lego bricks that plug into each-other – mix and match.
- This week: convolutional neural networks.
- Allows for efficiently modeling relevant n-grams.

Roots: CNNs and image recognition

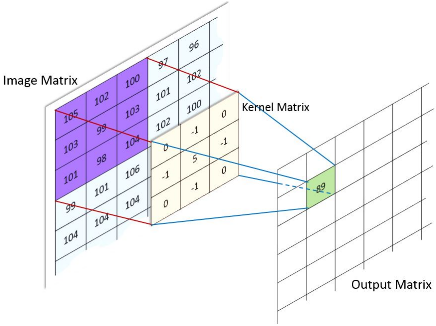
- Evolved in the 90s in the fields of signal processing and computer vision.
- Proved great for object recognition, independent of position in image.
- These roots are witnessed by the terminology associated with CNNs.

Roots: CNNs and image recognition

- Evolved in the 90s in the fields of signal processing and computer vision.
- Proved great for object recognition, independent of position in image.
- These roots are witnessed by the terminology associated with CNNs.
- A convolution operation is defined on the basis of a kernel or filter: a matrix of weights.
- The size of the filter referred to as the receptive field.
- Several standard convolution operations are available for image processing: Blurring, sharpening, edge detection, etc:
- `https://en.wikipedia.org/wiki/Kernel_(image_processing)`

2d convolutions for image processing

- The output of an image convolution is computed as follows:
  - Slide the filter matrix across every pixel.
  - For each pixel, compute the matrix convolution operation:
  - Multiply each element of the filter matrix with its corresponding element of the image matrix, and sum the products.
  - Edges requires special treatment (e.g. zero-padding or reduced filter).
- Each pixel in the resulting filtered image is a weighted combination of its neighboring pixels in the original image.



Convolutions and CNNs

- Convolutions are also used for feature extraction for ML models.
- Forms the basic building block of CNNs.
- But then we want to learn the weights of the filter,
- and typically apply a non-linear activation function to the result,
- and use several filters.

Convolutions and CNNs

- Convolutions are also used for feature extraction for ML models.
- Forms the basic building block of CNNs.
- But then we want to learn the weights of the filter,
- and typically apply a non-linear activation function to the result,
- and use several filters.

CNNs in NLP:

- Convolution filters can also be used for feature extraction from text:
- 'n-gram detectors'.
- Pioneered by Collobert, Weston, Bottou, et al. (2008, 2011) for various tagging tasks, and later by Kalchbrenner et al. (2014) and Kim (2014) for sentence classification.
- A massive proliferation of CNN-based work in the field since.

CNNs: an overview

- AKA convolution-and-pooling architectures or ConvNets.

CNNs explained in three lines

- A convolution layer extracts n-gram features across a sequence.
- A pooling layer then samples the features to identify the most informative ones.
- These are then passed to a downstream network for prediction.

- We'll spend the rest of the lecture fleshing out the details.