Convolutional Neural Nets

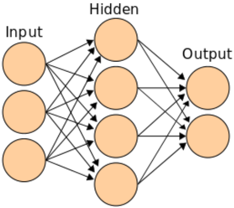
and Character-based Language Models

Jon Dehdari

February 1, 2016

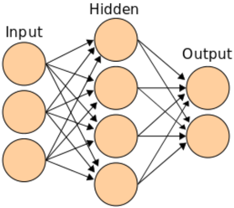
Too Connected!

- All the neural networks that we've seen so far are *fully connected* between each layer
- That is, every node in a layer is connected to every node in the next layer



Too Connected!

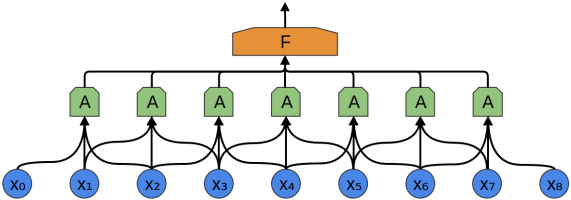
- All the neural networks that we've seen so far are *fully connected* between each layer
- That is, every node in a layer is connected to every node in the next layer



- This is fine for a small number of inputs, but can be problematic for a large number of inputs ($|\mathbf{x}| \cdot |\mathbf{h}|$)

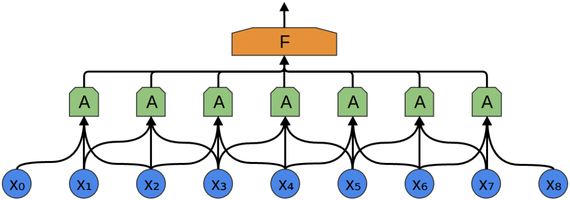
Convolutional Neural Network (CNN)

- Convolutional layers localize connections to a small window of input



Convolutional Neural Network (CNN)

- Convolutional layers localize connections to a small window of input

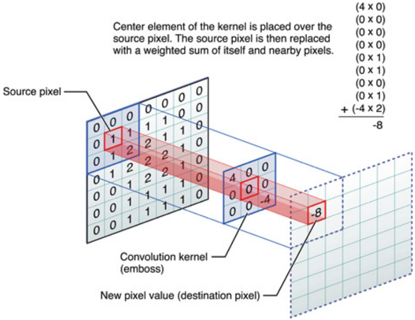


- For example, the input node x_3 connects to the second, third, and fourth nodes at the next layer, which is a normal dot-product + activation function

Image courtesy of Colah's awesome blog

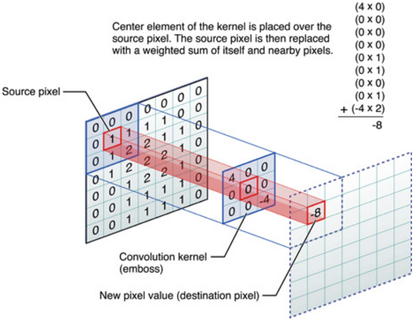
Convolutions

- A convolution *in this context* is just the dot product of a window of input and its weight matrix:



Convolutions

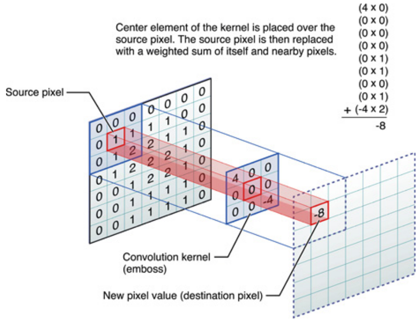
- A convolution *in this context* is just the dot product of a window of input and its weight matrix:



- It's also sometimes called a filter or kernel, because CNNs were originally popular with image processing

Convolutions

- A convolution *in this context* is just the dot product of a window of input and its weight matrix:



- It's also sometimes called a filter or kernel, because CNNs were originally popular with image processing
- It's common to use the same weight matrix at all positions (parameter sharing), because a cat is a cat regardless of where in the image it is!

Image courtesy of Apple

CNN Hyperparameters

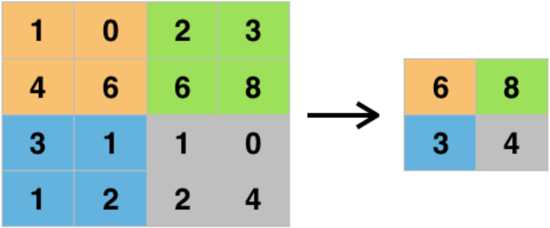
Window Size – number of inputs. Also called the (local) receptive field

Stride – amount of overlap between each window. A stride of 1 is densest and most common

Padding – default values for areas outside of the input. If used, usually 0

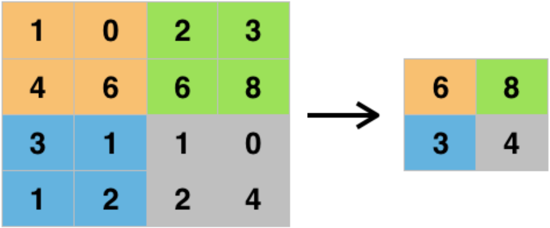
Pool Party!

- **Pooling** takes the maximum value ("max pooling"), arithmetic mean ("average pooling"), or L2 norm of a small block of nodes. Max pooling most common.



Pool Party!

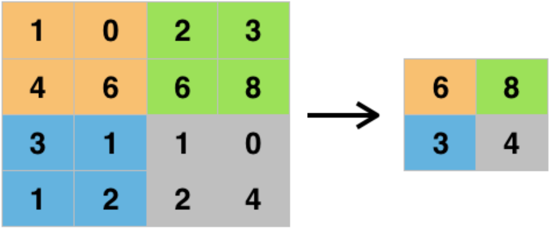
- **Pooling** takes the maximum value ("max pooling"), arithmetic mean ("average pooling"), or L2 norm of a small block of nodes. Max pooling most common.



- We can view pooling as "summarizing" a low-level area of input
- They allow small variations of input (translation invariance), and prevent overfitting

Pool Party!

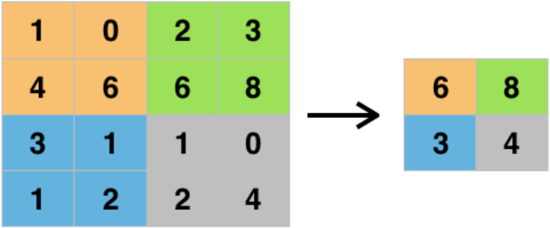
- **Pooling** takes the maximum value ("max pooling"), arithmetic mean ("average pooling"), or L2 norm of a small block of nodes. Max pooling most common.



- We can view pooling as "summarizing" a low-level area of input
- They allow small variations of input (translation invariance), and prevent overfitting
- Pooling layers are usually on top of a convolutional layer

Pool Party!

- **Pooling** takes the maximum value ("max pooling"), arithmetic mean ("average pooling"), or L2 norm of a small block of nodes. Max pooling most common.

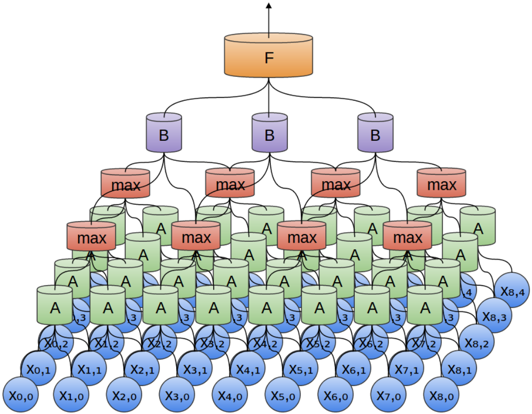


- We can view pooling as "summarizing" a low-level area of input
- They allow small variations of input (translation invariance), and prevent overfitting
- Pooling layers are usually on top of a convolutional layer
- Pooling is a type of **subsampling** / **down-sampling**

 Image courtesy of Wikipedia

Combining Convolutional Layers and Pooling

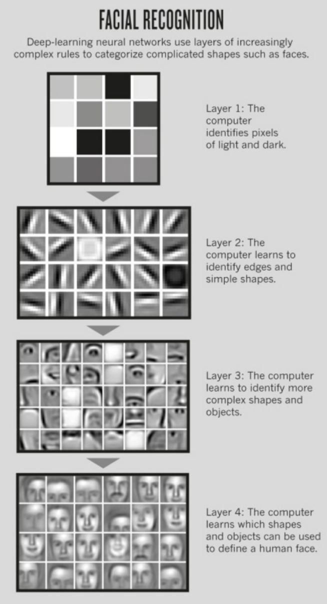
- We can combine a convolutional layer and max pooling:



- The 2-dim. input is blue, then a convolutional layer (green), then a max pooling layer (red), then another conv. layer (purple), and finally a fully-connected layer is orange

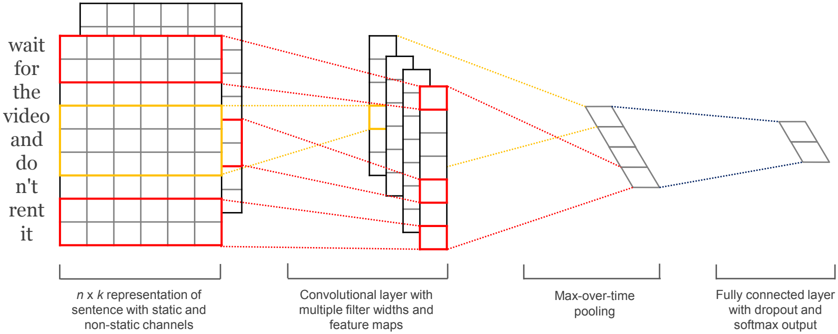
 Image courtesy of Colah's awesome blog

Face Detection Example



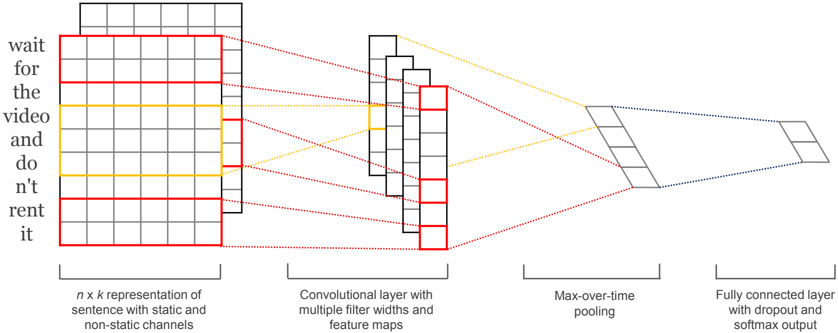
NLP Example: CNNs for Sentence Classification

- Yoon Kim (2014) used a CNN for sentence classification:



NLP Example: CNNs for Sentence Classification

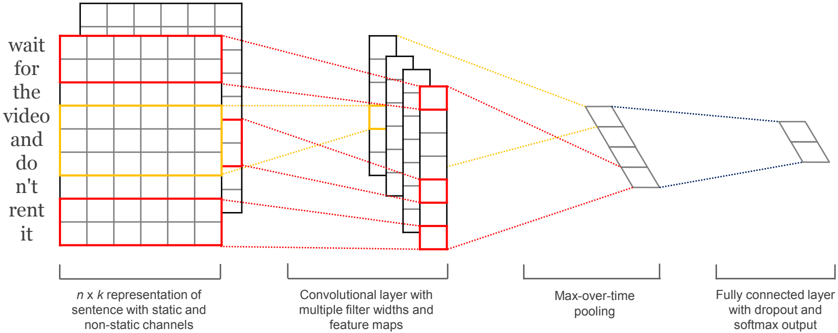
- Yoon Kim (2014) used a CNN for sentence classification:



- It used varying window sizes, maybe to approximate phrases

NLP Example: CNNs for Sentence Classification

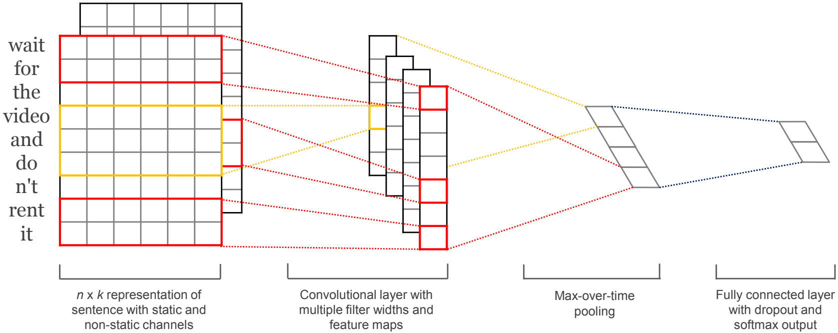
- Yoon Kim (2014) used a CNN for sentence classification:



- It used varying window sizes, maybe to approximate phrases
- It performed well in 6 different sentiment tasks, including movie & product reviews

NLP Example: CNNs for Sentence Classification

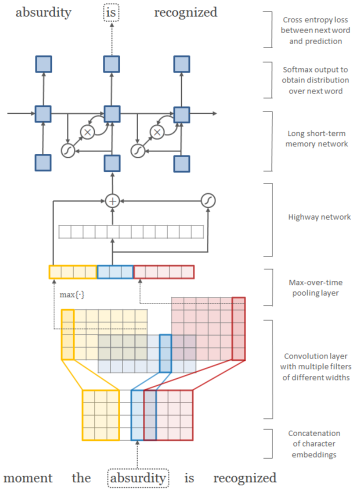
- Yoon Kim (2014) used a CNN for sentence classification:



- It used varying window sizes, maybe to approximate phrases
- It performed well in 6 different sentiment tasks, including movie & product reviews
- Interestingly, when word vectors were allowed to change (via backprop), they became better at discriminating antonyms

Character-based Neural Language Models

- Kim et al. (2016) made a nice character-based language model by combining convolutional layers, LSTMs, and highway networks (like LSTMs for deep networks):



Character-based Neural Language Models (cont'd)

	In Vocabulary					Out-of-Vocabulary		
	while	his	you	richard	trading	computer-aided	misinformed	looooook
LSTM-Word	although	your	conservatives	jonathan	advertised	–	–	–
	letting	her	we	robert	advertising	–	–	–
	though	my	guys	neil	turnover	–	–	–
	minute	their	i	nancy	turnover	–	–	–
LSTM-Char (before highway)	chile	this	your	hard	heading	computer-guided	informed	look
	whole	hhs	young	rich	training	computerized	performed	cook
	meanwhile	is	four	richer	reading	disk-drive	transformed	looks
	white	has	youth	richter	leading	computer	inform	shook
LSTM-Char (after highway)	meanwhile	hhs	we	eduard	trade	computer-guided	informed	look
	whole	this	your	gerard	training	computer-driven	performed	looks
	though	their	doug	edward	traded	computerized	outperformed	looked
	nevertheless	your	i	carl	trader	computer	transformed	looking

Table 6: Nearest neighbor words (based on cosine similarity) of word representations from the large word-level and character-level (before and after highway layers) models trained on the PTB. Last three words are OOV words, and therefore they do not have representations in the word-level model.

Further Reading

Overviews:

- https://colah.github.io/posts/2014-07-Conv-Nets-Modular
- https://colah.github.io/posts/2014-07-Understanding-Convolutions
- https://cs231n.github.io/convolutional-networks
- http://white.stanford.edu/teach/index.php/An_Introduction_to_Convolutional_Neural_Networks
- http://neuralnetworksanddeeplearning.com/chap6.html
- https://en.wikipedia.org/wiki/Convolutional_neural_network
- http://deeplearning.net/tutorial/lenet.html
- http://u.cs.biu.ac.il/~yogo/nnlp.pdf (§ 9)
- http://cs224d.stanford.edu/lectures/CS224d-Lecture13.pdf

Original Papers:

- LeCun, Yann, & Bengio, Yoshua. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361.10.
- Kim, Yoon. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of EMNLP-2014*. Doha, Qatar. ACL Antho: D14-1181. Software Link.
- Kim, Yoon, Yacine Jernite, David Sontag, Alexander M. Rush. 2016. Character-Aware Neural Language Models. In *Proceedings of AAAI-2016*. Phoenix, AZ, USA. arXiv:1508.06615. Software Link.