Convolutional Methods For Text

Tal Perry
@thetalperry
me@talperry.com

What We’ll cover today

- **Background**

  Things we want to do in NLP

  Refresher on LSTMs & why they’re gr8

- **Convolutions**

  What is a convolution

  1D convolution in Tensorflow

  Receptive Field vs Gradients

    Residual Connections

    Dilated convolutions

  Up/Down Sampling

    Pooling

    Deconvolutions

Workshop

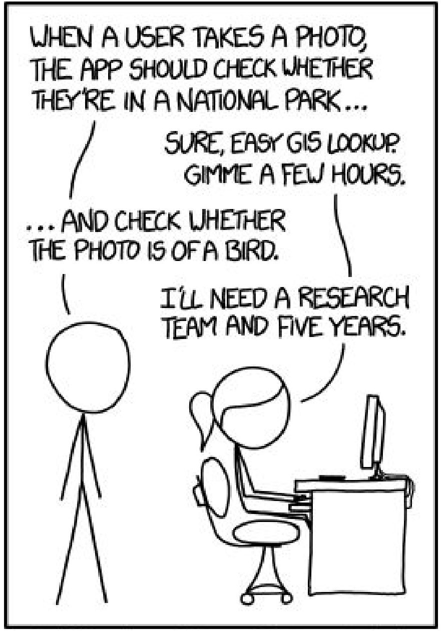
Restore Punctuation with convolutions

Who am I

- Tal Perry
- Data Scientist at Citi
- I do “Deep Learning” on chats
- Mostly work at the character level ⇒ My sequences are long
- I like to build stuff

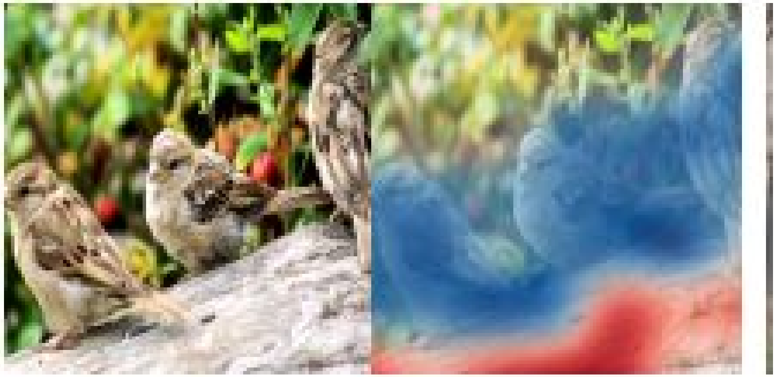
Why you should care *About deep learning*

~2013


WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

~2016

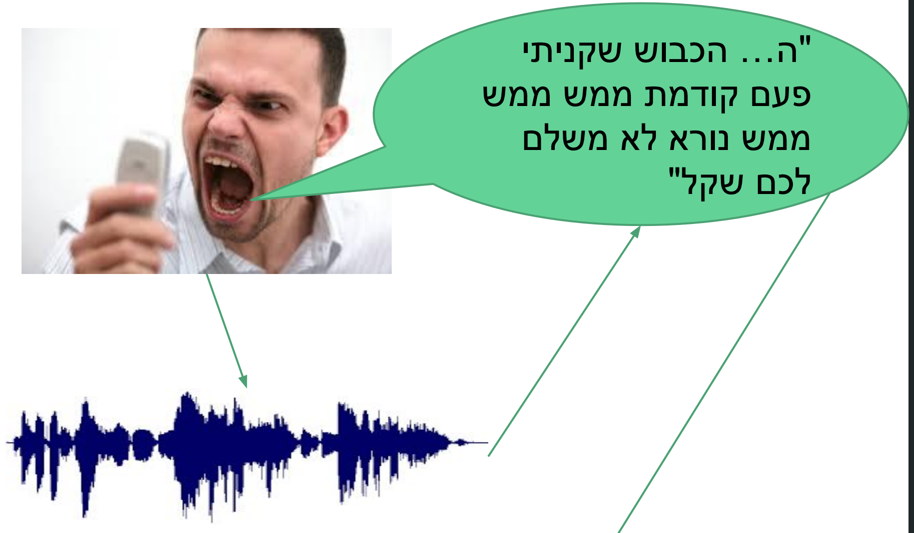


two birds sitting on top of a tree branch.

Why you should care *About convolutions*

- Convolutions are FAST
- Fast training means fast exploration
- Easier to put in production and hit SLAs
-

Things we need to do in NLP



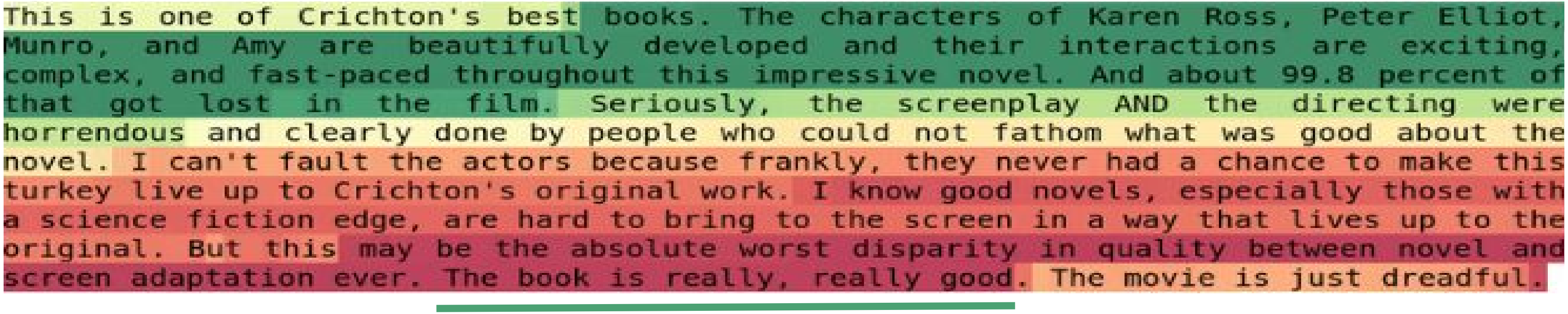
"ה... הכבוש שקניתי
פעם קודמת ממש ממש
ממש נורא לא משלם
לכם שקל"

Client_id	Product	Sentiment
7	Pickled Cabbage	Negative
7	Israeli Occupation	Neutral

- Tasks
  - Translation
  - Entity Recognition
  - Sentiment Analysis
  - Imputation
  - Dependency Parsing
- Why is this hard ?
  - “The rules” are ambiguous
  - People don’t follow them
  - All data is dirty

Two Technical Requirements

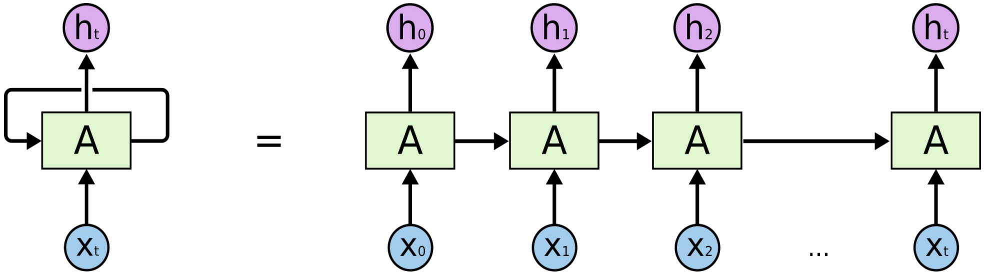
- Handle arbitrary input/output lengths
- Capture dependencies at multiple time scales

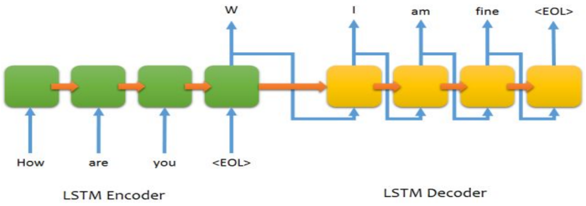


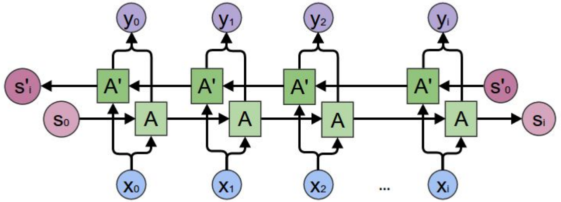
This is one of Crichton's best books. The characters of Karen Ross, Peter Elliot, Munro, and Amy are beautifully developed and their interactions are exciting, complex, and fast-paced throughout this impressive novel. And about 99.8 percent of that got lost in the film. Seriously, the screenplay AND the directing were horrendous and clearly done by people who could not fathom what was good about the novel. I can't fault the actors because frankly, they never had a chance to make this turkey live up to Crichton's original work. I know good novels, especially those with a science fiction edge, are hard to bring to the screen in a way that lives up to the original. But this may be the absolute worst disparity in quality between novel and screen adaptation ever. The book is really, really good. The movie is just dreadful.

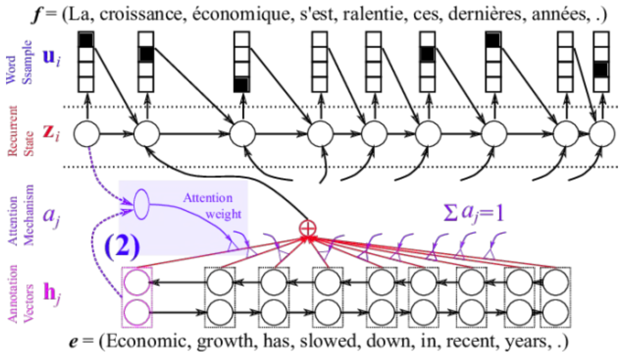
A Quick Refresher on RNNs

```python
class RNN():
    def __init__(self,hidden_size):
        self.W_hh = np.random.rand(hidden_size,hidden_size)
        self.W_xh = np.random.rand(hidden_size, hidden_size)
        self.W_hy = np.random.rand(hidden_size, hidden_size)
        self.h = np.zeros(hidden_size)
    def step(self,x):
        #update the hidden state
        self.h = np.tanh(np.dot(self.W_hh, self.h) + np.dot(self.W_xh, x))
        # compute the output vector
        y = np.dot(self.W_hy, self.h)
        return y
```

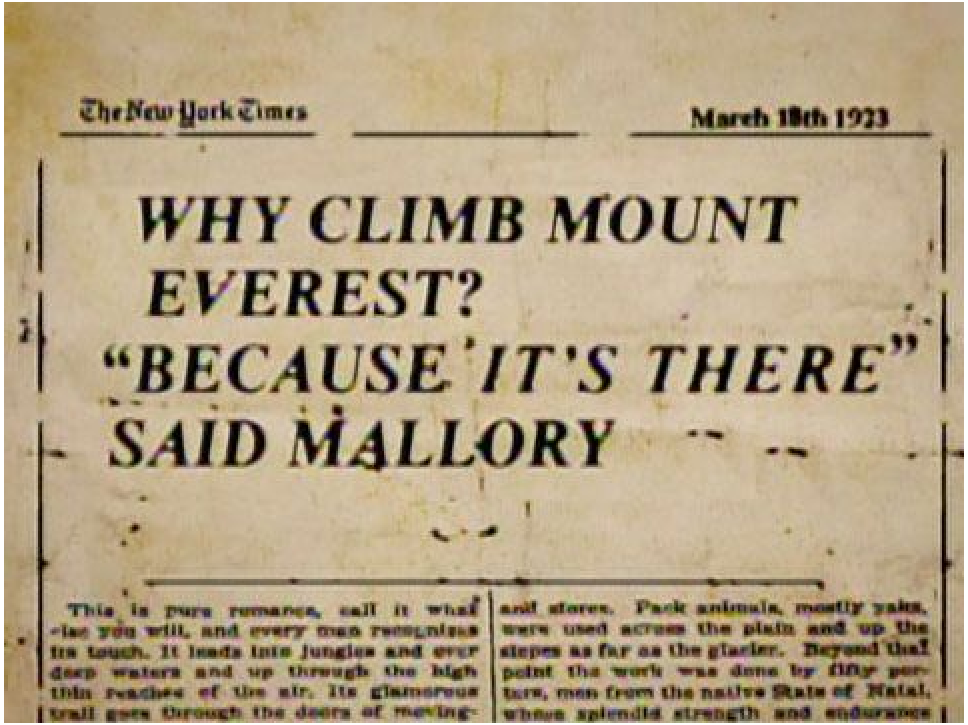








So why use Convolutions ?



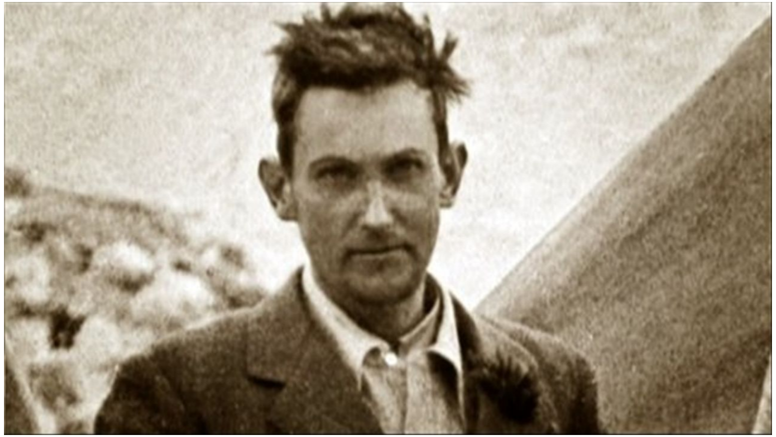
The New York Times

March 18th 1923

WHY CLIMB MOUNT EVEREST?
"BECAUSE IT'S THERE"
SAID MALLORY

This is pure romance, call it what -ise you will, and every man recognizes its touch. It leads into jungles and over deep waters and up through the high thin reaches of the air. Its glamorous trail goes through the doors of moving-

and stores. Pack animals, mostly yaks, were used across the plain and up the slopes as far as the glacier. Beyond that point the work was done by fifty porters, men from the native State of Natal, whose splendid strength and endurance



So why use Convolutions ?

- They’re very fast and data efficient
- CNNs work in parrelel and make better use of GPUs
- CNNs process information hierarchical instead of sequentially making it easier to capture complex relations

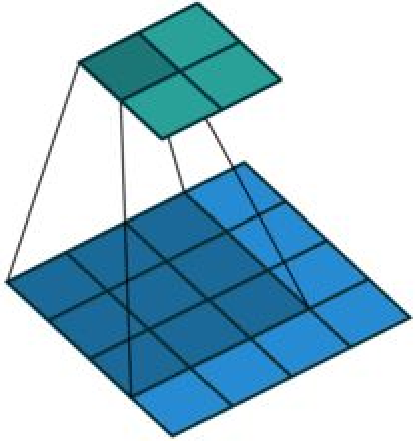
“When the CNN and the best RNN of similar size are trained in the same way, the CNN outperforms it by 1.5 BLEU.... The FAIR CNN model is computationally very efficient and is nine times faster than strong RNN systems”

- A novel approach to neural machine translation

Facebook AI Research

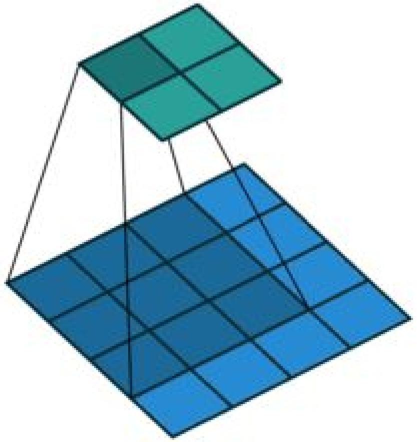
Convolution Refresher

- Convolutions apply a “linear map” on patches of the input and output a scalar at each step
- Convolutions work on the Height, Width and Depth dimensions
- We usually run many convolutions on the same patch to learn multiple features
- If the depth is 1 (grey scale image) Convolutions are a matrix product
- Otherwise we need to do a tensor product
- Instead of thinking of tensor products you can think of Fourier Transform
- $x * y = DTFT^{-1}\left[DTFT\{x\} \cdot\ DTFT\{y\}\right],$



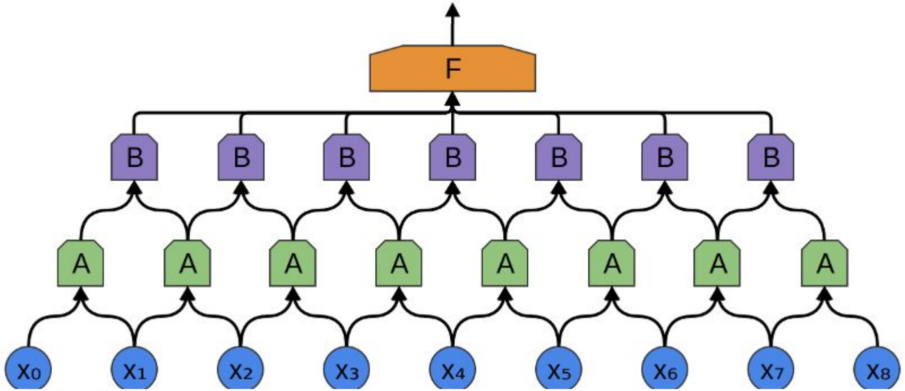
Convolution Refresher

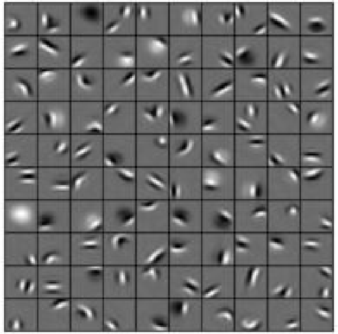
```
 def _conv2d(x,height,width,input_channels,num_convolutions):
     filter_ = tf.get_variable("conv_filter",
                            shape=[height, # The height of our patch
                                  width,  # The width of our patch
                                  input_channels, # How many channels/featurs are in the input
                                  num_convolutions, # How many different convolutions do we want to run
                                  ])
     convolved = tf.nn.conv2d(x, filter=filter_, strides=[1,1,1,1],) #Apply the convolution
     return convolved

def _conv1d(x,width,input_channels,num_convolutions):
    x = tf.expand_dims(x,axis=1)
    filter_ = tf.get_variable("conv_filter",
                           shape=[1, width, input_channels,num_convolutions])                    ])

    convolved = tf.nn.conv2d(x, filter=filter_, strides=[1,1,1,1],) #Apply the convolution
    convolved = tf.squeeze(x,axis=1)
    return convolved
```

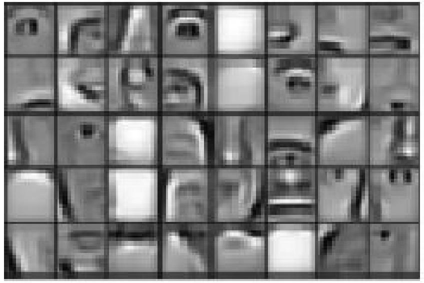


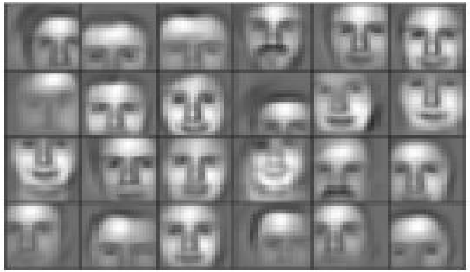
Receptive Fields and Hierarchy

Receptive Fields and Hierarchy









Receptive Fields and Hierarchy

What has 4 letters, sometimes 9 letters, but never has 5 letters.

In text we usually want to get everything into the receptive field

It's hard to increase our receptive fields

- Using large filters is computationally expensive and harder to train
- Naively stacking layers gives us only a linear growth in receptive field
- Naively stacking layers introduces Vanishing/Exploding gradients

Me when my network won't converge

Me when I get NaN in my gradients

Two Solutions

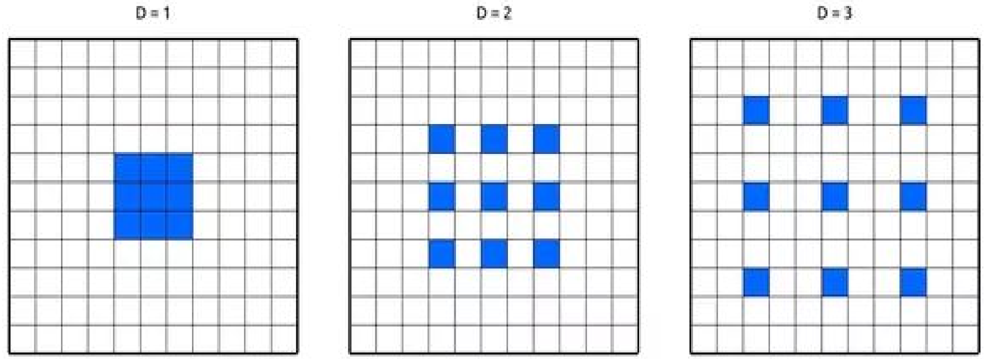
- Don’t stack convolutions naively.
  - Use Dilated convolutions to grow exponentially
  - Or residual connections to tame your gradients
  - Or both

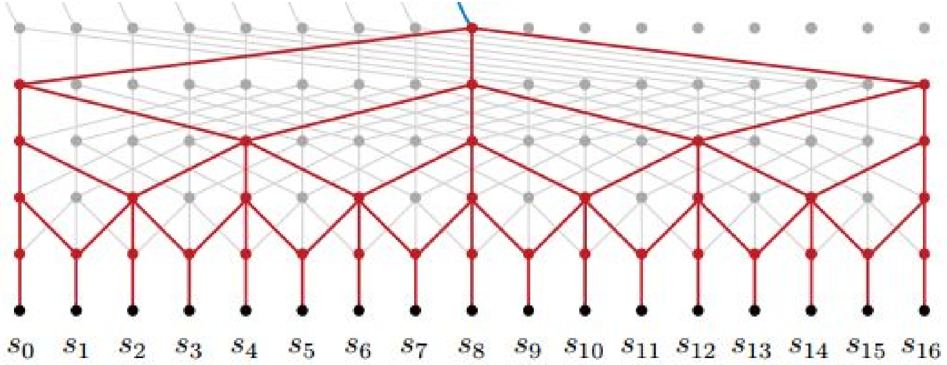
Me with dilated convolutions

Me with informative well behaved gradients

Dilated Convolutions

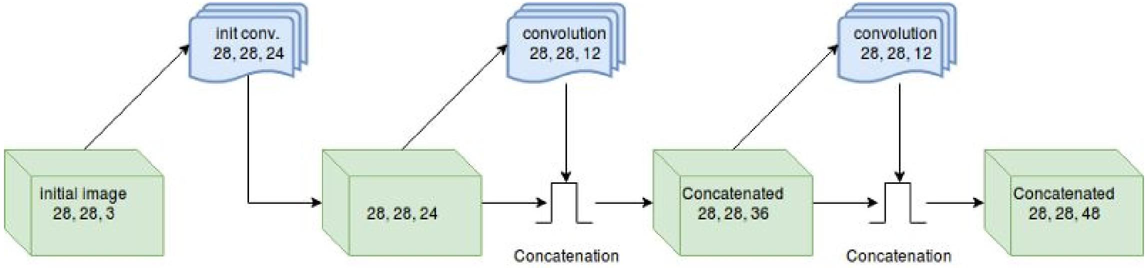
- With dilated convolutions, we skip over parts of the input within a single application of the convolution.
- By stacking larger dilations we can get an **exponential growth in receptive field with only linear depth**
- tf.nn.atrous_conv2d





Residual Connections

- If we add direct connections between distant layers we can prevent vanishing gradients
- Then we can stack as many layers as we like. This gives us a more fine grained hierarchy.
- Multiple ways to do this
    - Add the input to the conv output after non linearity (ResNet)
    - Concatenate the input to the conv output after non



init conv.
28, 28, 24

convolution
28, 28, 12

convolution
28, 28, 12

initial image
28, 28, 3

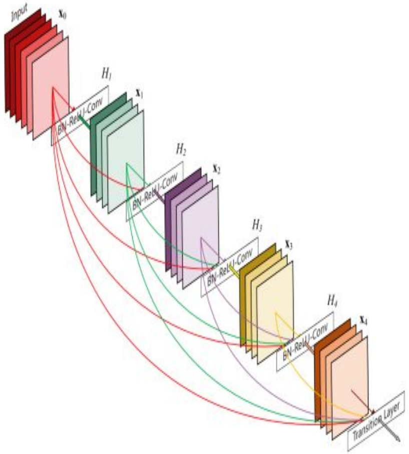
28, 28, 24

Concatenation

Concatenated
28, 28, 36

Concatenation

Concatenated
28, 28, 48



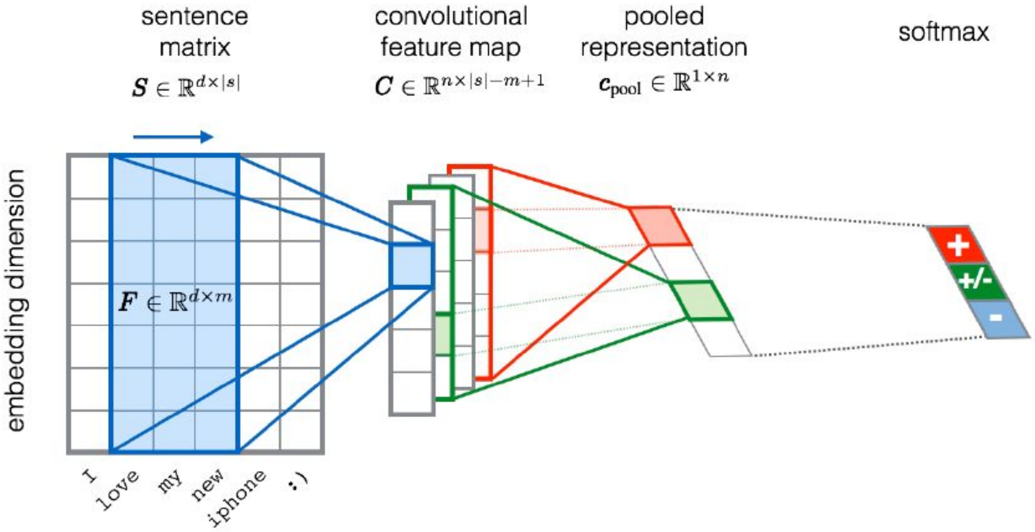
Changing variable length outputs

Changing variable length outputs

- Sometimes we need to change the length of our input
  - Translation “*בראשית*” ⇒ *“In the begining”*
  - Imputation
  - Auto Encoding
  - Classification

Reducing Sequences

- **If your input has a bound on its length and you need to reduce to a vector**
  Classification, Sentiment, Paraphrase detection
  - Use pooling
  - Consider adding residual layers between poolings


sentence matrix
$S \in \mathbb{R}^{d \times |s|}$
convolutional feature map
$C \in \mathbb{R}^{n \times |s|-m+1}$
pooled representation
$c_{\text{pool}} \in \mathbb{R}^{1 \times n}$
softmax
embedding dimension
$F \in \mathbb{R}^{d \times m}$
I
love
my
new
iphone
:)
+
+/-
-

Reducing Sequences

If there is no bound on |s1| and target is a vector / scalar
Document classification, Document Embedding.

- I don’t know what to do
- Use an LSTM
- Tell me what you know ?

Reducing Sequences

If you know that a|s2| <= |s1| <=b|s2|
Entity recognition, POS, punctuation

- Maintain the same length representation
- Use padding on source and target.
- Use Tensorflows sequence mask and sequence loss functions
- Optionally apply the mask between layers

This works if you use 0 as your padding id

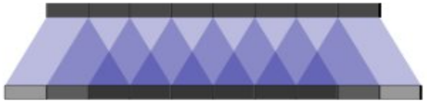
```
def _loss(self,logits,targets):
    lengths = tf.reduce_sum(tf.sign(targets),axis=1)
    maxlen = self.original.get_shape().as_list()[1]
    mask = tf.sequence_mask(lengths,dtype=tf.float32,maxlen=maxlen)
    loss = tf.contrib.seq2seq.sequence_loss(logits=logits,
                                            targets=self.original,
                                            weights = mask
                                            )
```

Changing variable length outputs

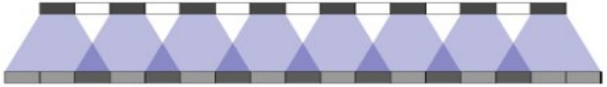
Upsampling with “deconvolutions”

Deconvolutions are convolutions in “reverse” (transpose

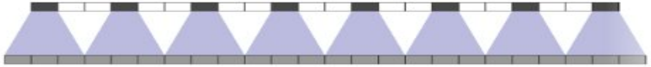
- Our width dictates how many steps our input will spreac to
- Striding in deconvolutions adds fake holes to the input. Use it to control overlapping/ reason about growth
- Add regular convolutions in the end.
- *tf.nn.conv2d_transpose*



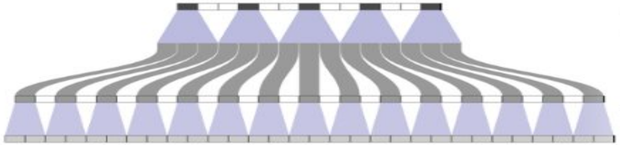
Convolution of stride 1 and width 3



A deconvolution with stride 2 and width 3



A deconvolution with stride 3 and width 3



Stacking two deconvolutional layers one after the other. The top layer is stride 3 and width 3, while the second layer is stride 2 and width 2. This grew our sequence length from 5 to 30, a factor of 6.

Practice Time

You’re going to restore punctuation and capitalisation in text using convolutions

- Repo https://github.com/talolard/CMFT_PyCon2017
- Choose 1 or two questions you want to tackle (There are 5)
- Batteries included, just do the convolution part
- Much easier (for the model) if you just do capitalisation.
- Things I hope you’ll learn
    - How to adjust tensorflow ops for 1 dimensional data
    - Reason about sequence lengths and tensor shapes
    - Implement the things we learnt
    -
    -

the name xubuntu is a portmanteau of xfce and ubuntu

The name Xubuntu is a portmanteau of Xfce and Ubuntu .

Thank you

Tal Perry
me@talperry.com
@thetalperry

Questions ?