An Old Friend

Assignment 3

- LSTMs and Seq2Seq
- We’ve heard concerns about time
- It is really simple

Q1

- Very trivial variant on the previous assignment
- Estimated time to complete = 5 mins

Q2

- Trivial (but interesting) Seq2Seq Problem
- Estimated time to complete (less than 1 hr)
  - (coding + training)

Due in 2 Weeks

BUT GET STARTED

ASAP

Recap

So far

MLP

- MLP
  - Vector -> Vector using matrix multiplications
  - Nonlinear activations

CNN

- Linear operation (local weighted sum)
- Cuts down on parameters
- Diverse applications
  - Images
  - Audio
  - Text (assignment)

LSTM

- Sequence modeling workhorse
- Cell traverses a sequence
- Encoded a variety of sequences

Encoder-Decoder Framework

- Creative LSTM use
- You will implement this in an assignment
- Applications in:
  - Machine translation
  - TTS
  - STT

Attention

- Easing the long-dependence issue
- Combined with a seq2seq framework
- Newer patterns:
  - Just use attention
  - https://arxiv.org/abs/1706.03762
  - No conv / recurrence bullshit

Repurposing CNN

- Agenda
  - Use CNNs for sequence problems

Why?

- Ideas?

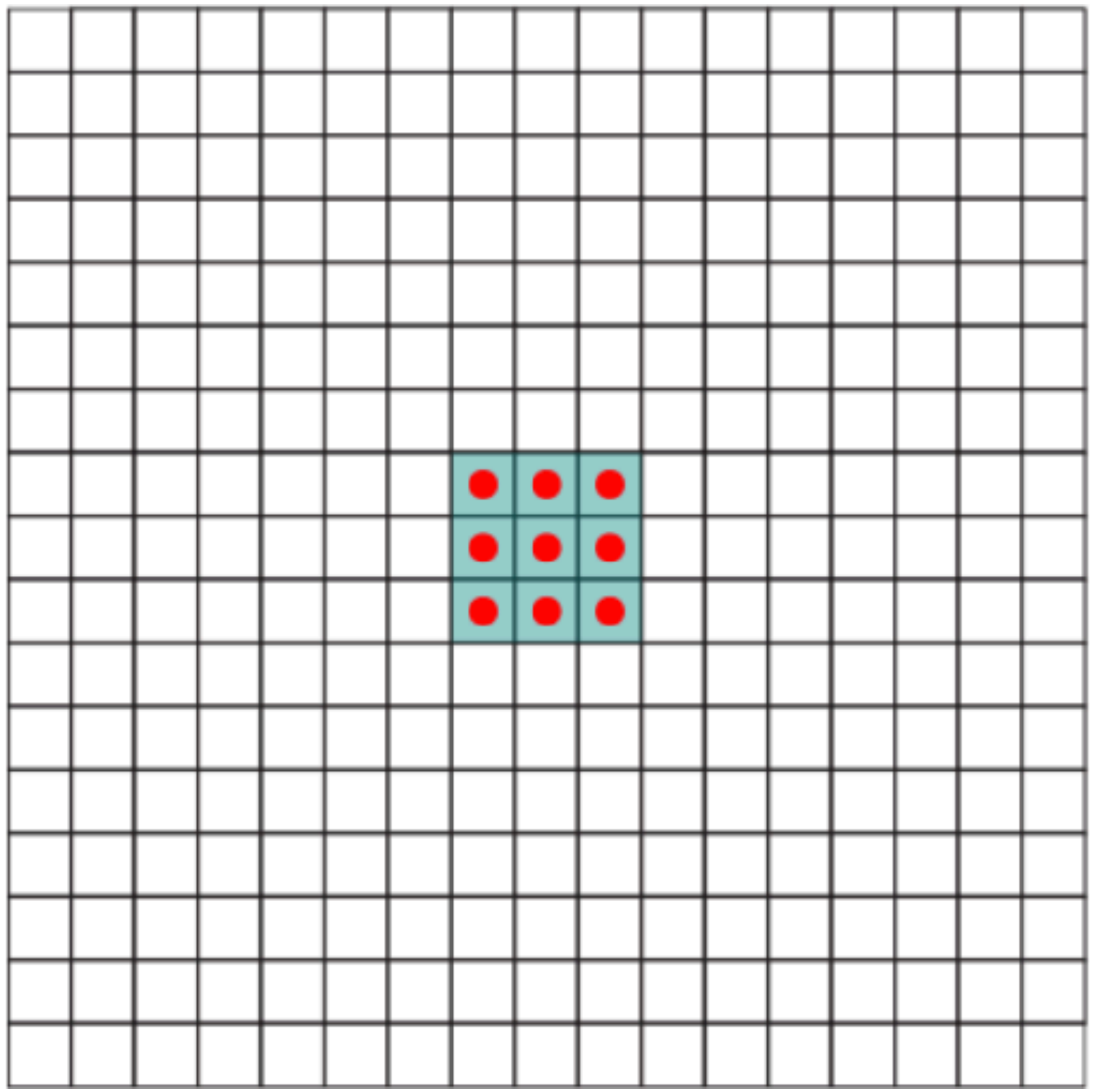
Big Idea

- RNNs and LSTMs (in particular)
- Long dependence (sort of)

Simple Tweak

- Dilated CNNs
- In a certain configuration

A Kernel



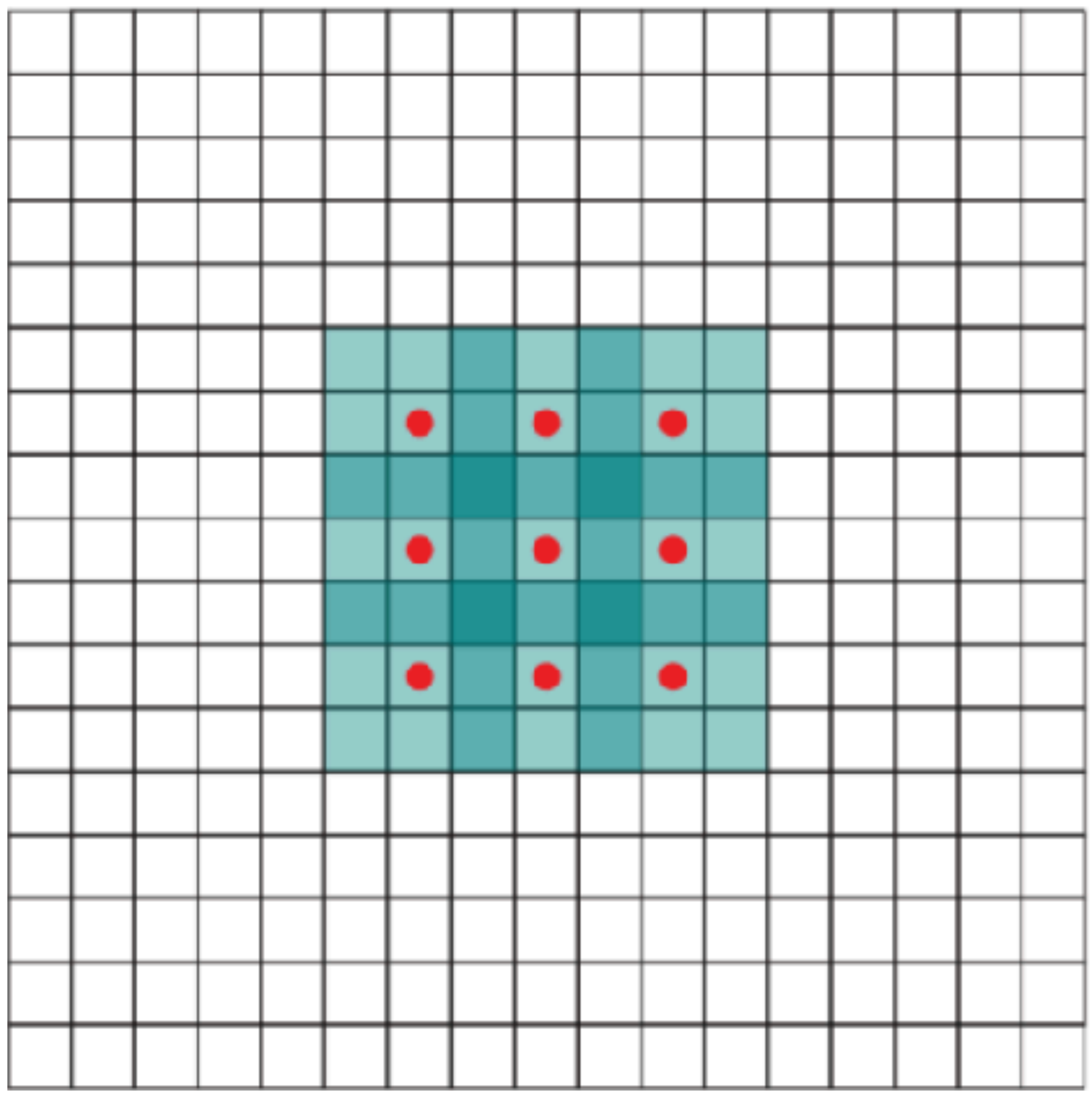
This is

- A 3x3 kernel
- The reds are weights

With Dilation Rate 2

- See every 2nd pixel

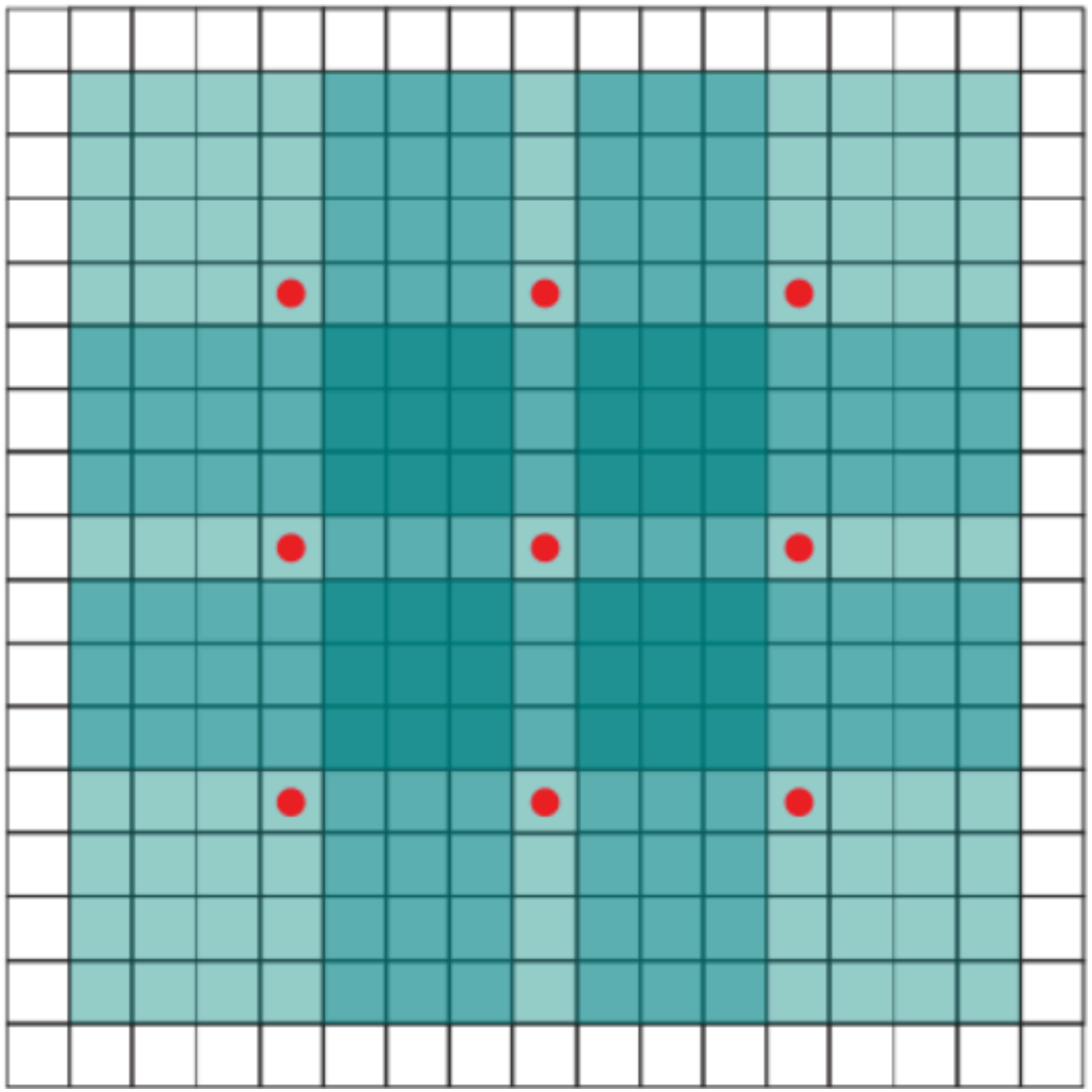
Dilation=2



The Kernel

- “Sees” a 7x7 region
- But same # of parameters - 9

Dilation=4



The Kernel

- “Sees” 15x15 region
- 9 parameters

We Can

- Rapidly grow the kernel size - exponentially
- And thus the “receptive field”
- Keep parameter count growth linear

From Prev Presentation

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67
Output channels								
Basic	C	C	C	C	C	C	C	C
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	C

Low Parameter Count

- Desired loss at the top
- Per-pixel classification or regression

Applications

- Segmentation
- Maybe chunking of text
  - (not sure if it has been tried but seems like a fit)

What next?

- 2 Additional Modules

From A Previous Presentation

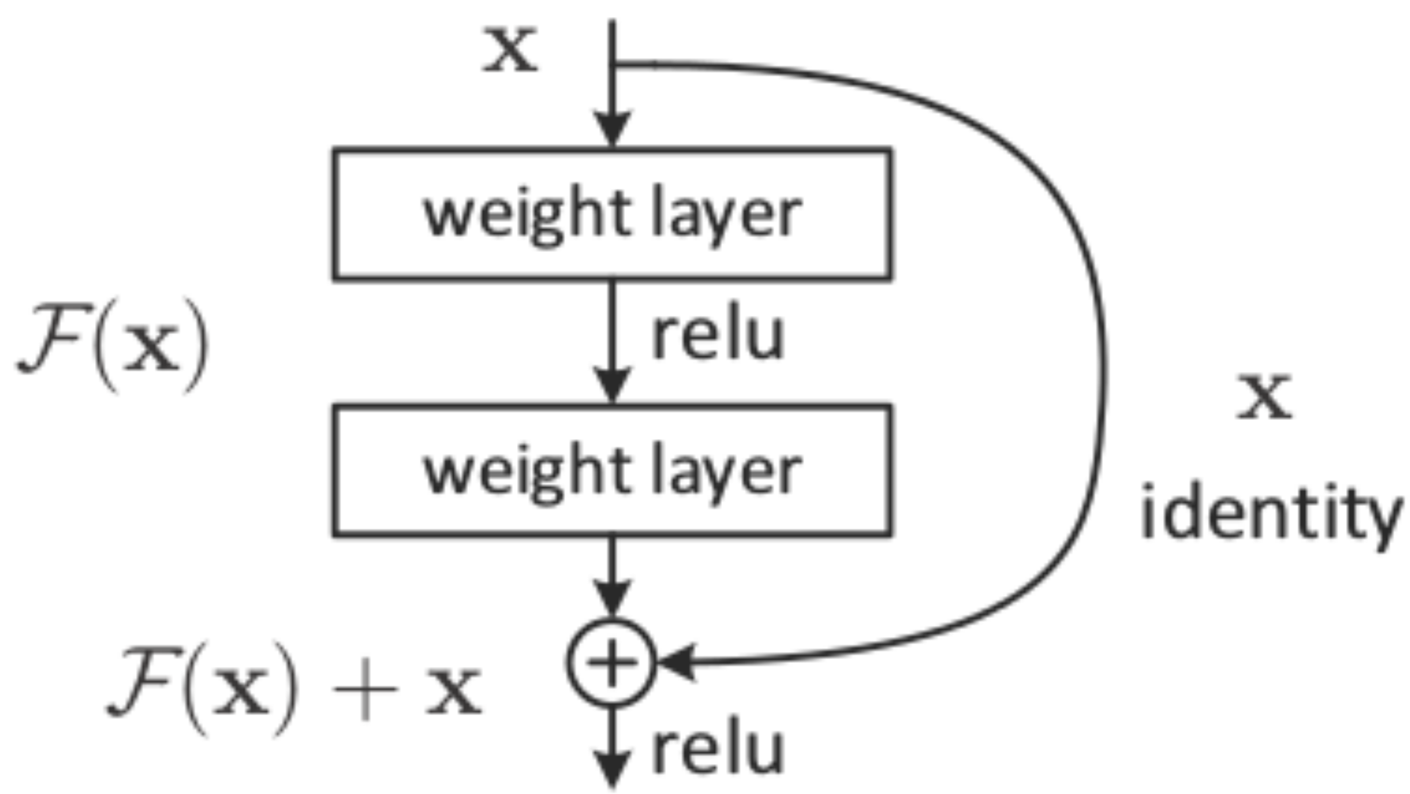


Figure 2. Residual learning: a building block.

Skip Connections!

Computational Cost

- ***VERY*** low

Patterns

- Each dilated convolution
  - Stride 1 (so output size = input size)
  - Group = dilated convs + skip connections

So Far

- Residual blocks
- Dilations
- Segmentation / regression

What Next?

- Generative Models

You’ve Generated

- Text

Distribution Over

- Next character given current sequence

What About

- Audio

Audio

- Signal
- Typically sampled at 16000, 32000, 44100 Hz.
- i.e for 1s of audio you have 16000 (?) values

“Very” Long Dependence

- Typically Run MFCC or STFT
- Then pass these representations to LSTM

Wavenet

- Work With Raw Audio
- A Generative Model of raw audio

How?

- Ideas
- Hint: Combine ingredients we discussed
- You want long dependence

Generative Model

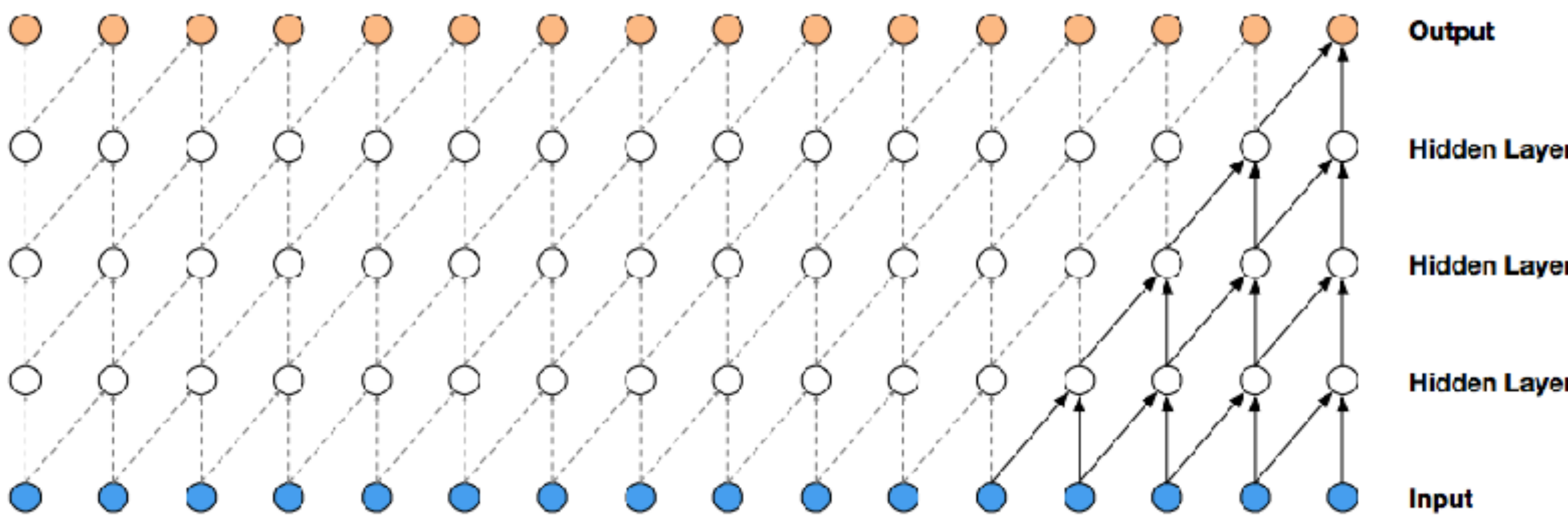
- Next audio sample
- Conditioned on previous audio samples

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

p

- Causal Convolution

huh?


Output
Hidden Layer
Hidden Layer
Hidden Layer
Input

Training

- In parallel
- Why?

Generating

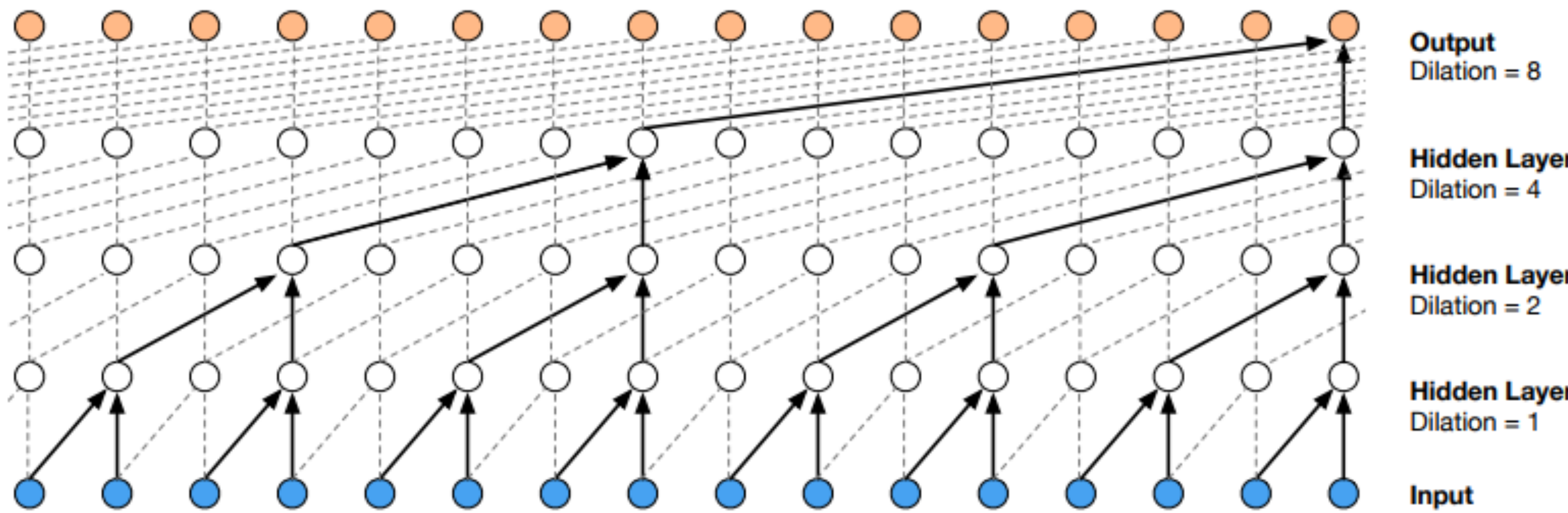
- Must be sequential
- In fact 1 sample at a time

2nd Ingredient

- Dilations

Dilation

- Sorts out long dependence easily (as we know)


Output
Dilation = 8
Hidden Layer
Dilation = 4
Hidden Layer
Dilation = 2
Hidden Layer
Dilation = 1
Input

Adding Text

- How?

TCN + Dilation

- Recently
  - Empirically better than RNNs (LSTMs etc)
  - Maybe - let us see
- Very competitive with RNNs