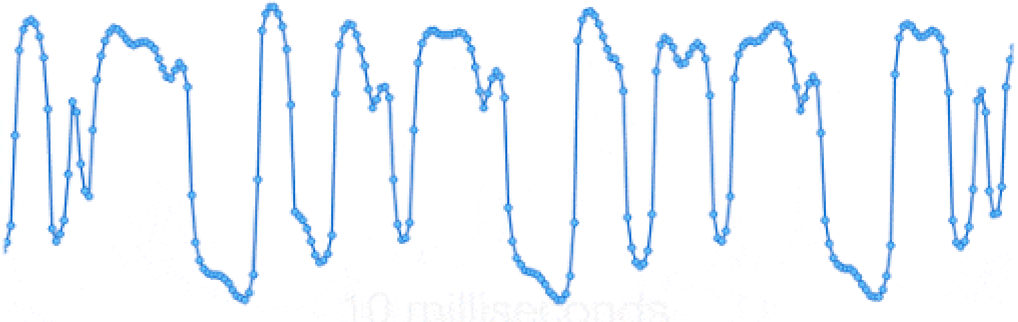
An implementation of WaveNet

May 2017

Vassilis Tsiaras
Computer Science Department
University of Crete

Motivation

- In September 2016, DeepMind presented WaveNet.
- WaveNet is a deep generative model of raw audio waveforms.
- It is able to generate speech which mimics any human voice and which sounds more natural than the best existing Text-to-Speech systems.
- WaveNet directly models the raw waveform of the audio signal, one sample at a time.



- By modelling the waveforms, WaveNet can model any kind of audio, including music.
- DeepMind published a paper about WaveNet, which does not reveal all the details of the network.
- We built an implementation of WaveNet based on partial information about their architecture.
- This attempt revealed the computational requirements of WaveNet. Also the new software will be used to investigate the properties of these networks and their potential applications.

WaveNet architecture – Pre-processing

- The joint probability of a speech waveform $\mathrm{x} = x_1 x_2 \cdots x_T$ can be written as

$$p(x) = \prod_{t=1}^{T} p(x_t | x_1, \dots, x_{t-1})$$

- WaveNet represents $p(x_t | x_1, \dots, x_{t-1})$ with a categorical distribution where x_t falls into one of a number of bins (usually 256).

- Raw audio, y_t, is first transformed into x_t, where $-1 < x_t < 1$, using an μ-law transformation

$$x_t = sign(y_t) \frac{\ln(1 + \mu |y_t|)}{\ln(1 + \mu)}$$

where $\mu = 255$

- Then x_t is quantized into 256 values and encoded to one-hot vectors.

- **Example:**

$-2.2, -1.43, -0.77, -1.13, -0.58, -0.43, -0.67, \dots$
signal

→

$-0.7, -0.3, 0.2, -0.1, 0.4, 0.6, 0.3, \dots$
μ-law transformed

→

$0, 1, 2, 1, 2, 3, 2, \dots$
quantized into 4 bins

→

bin 0	1	0	0	0	0	0	0
bin 1	0	1	0	1	0	0	0
bin 2	0	0	1	0	1	0	1
bin 3	0	0	0	0	0	1	0

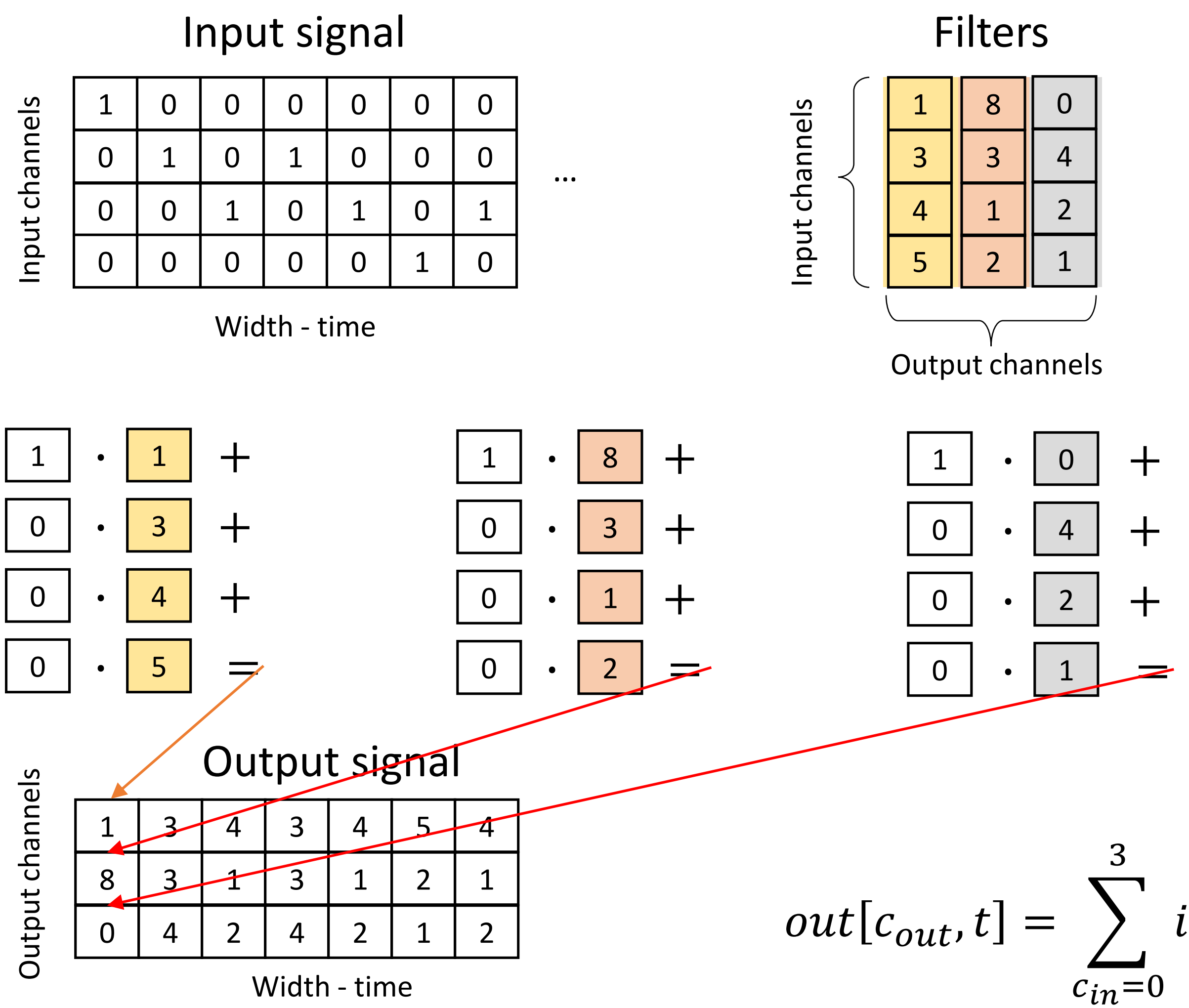
…

one-hot vectors

→ *Input to WaveNet*

WaveNet architecture – 1×1 Convolutions

- 1×1 convolutions are used to change the number of channels. They do not operate in time dimension
- Example of a 1×1 convolution with 4 input channels, and 3 output channels



Input signal

Input channels

1	0	0	0	0	0	0	
0	1	0	1	0	0	0	...
0	0	1	0	1	0	1	
0	0	0	0	0	1	0	

Width - time

Filters

Input channels

1	8	0
3	3	4
4	1	2
5	2	1

Output channels

$1 \cdot 1 +$
$0 \cdot 3 +$
$0 \cdot 4 +$
$0 \cdot 5 =$

$1 \cdot 8 +$
$0 \cdot 3 +$
$0 \cdot 1 +$
$0 \cdot 2 =$

$1 \cdot 0 +$
$0 \cdot 4 +$
$0 \cdot 2 +$
$0 \cdot 1 =$

Output signal

Output channels

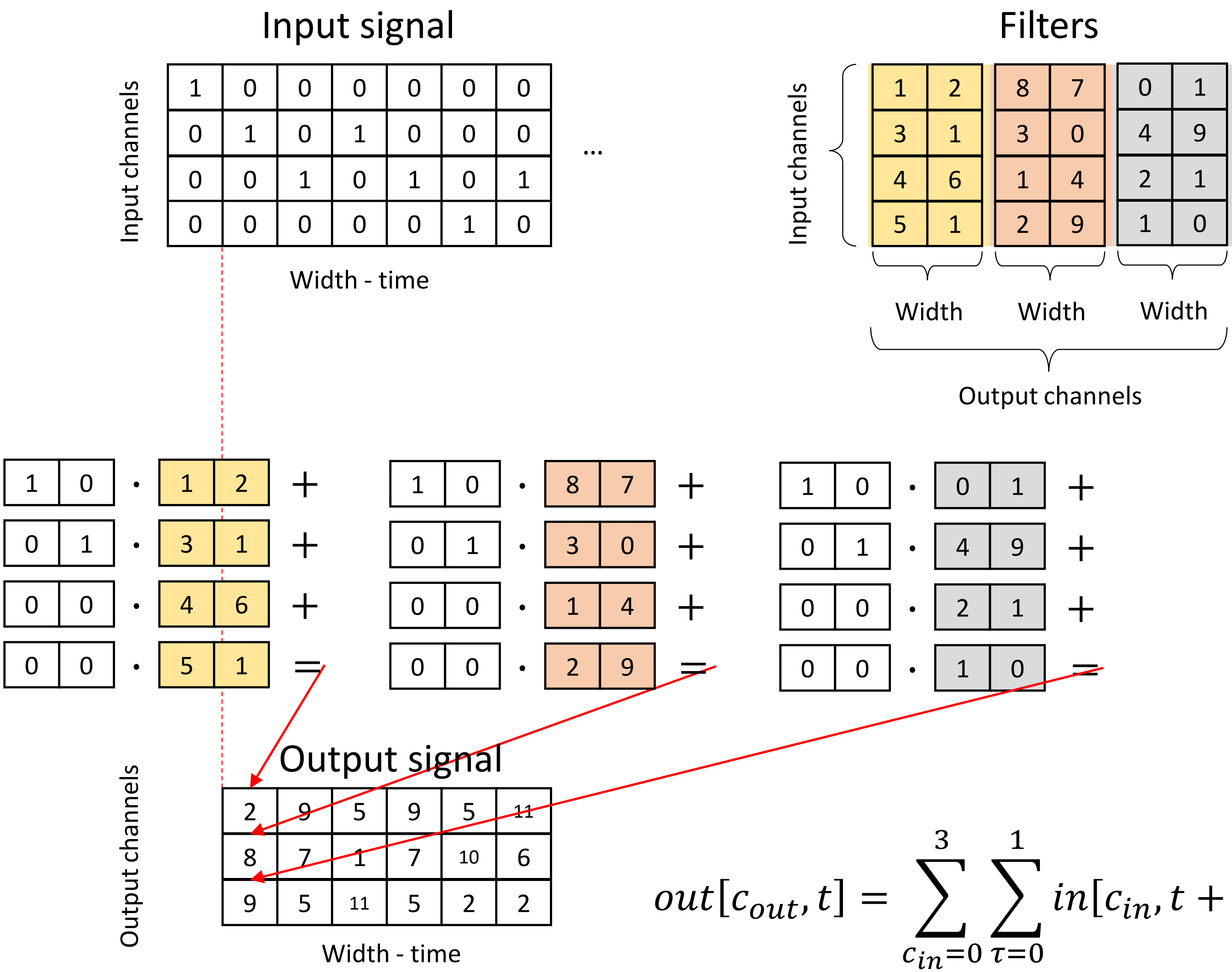
1	3	4	3	4	5	4
8	3	1	3	1	2	1
0	4	2	4	2	1	2

Width - time

$$out[c_{out}, t] = \sum_{c_{in}=0}^{3} in[c_{in}, t] \cdot filter[c_{out}, c_{in}]$$

WaveNet architecture – Causal convolutions

- Example of a **causal** convolution of width 2, 4 input channels, and 3 output channels



Input signal

1	0	0	0	0	0	0	…
0	1	0	1	0	0	0	
0	0	1	0	1	0	1	
0	0	0	0	0	1	0	

Filters

1	2	8	7	0	1
3	1	3	0	4	9
4	6	1	4	2	1
5	1	2	9	1	0

1	0	·	1	2	+	1	0	·	8	7	+	1	0	·	0	1	+
0	1	·	3	1	+	0	1	·	3	0	+	0	1	·	4	9	+
0	0	·	4	6	+	0	0	·	1	4	+	0	0	·	2	1	+
0	0	·	5	1	=	0	0	·	2	9	=	0	0	·	1	0	=

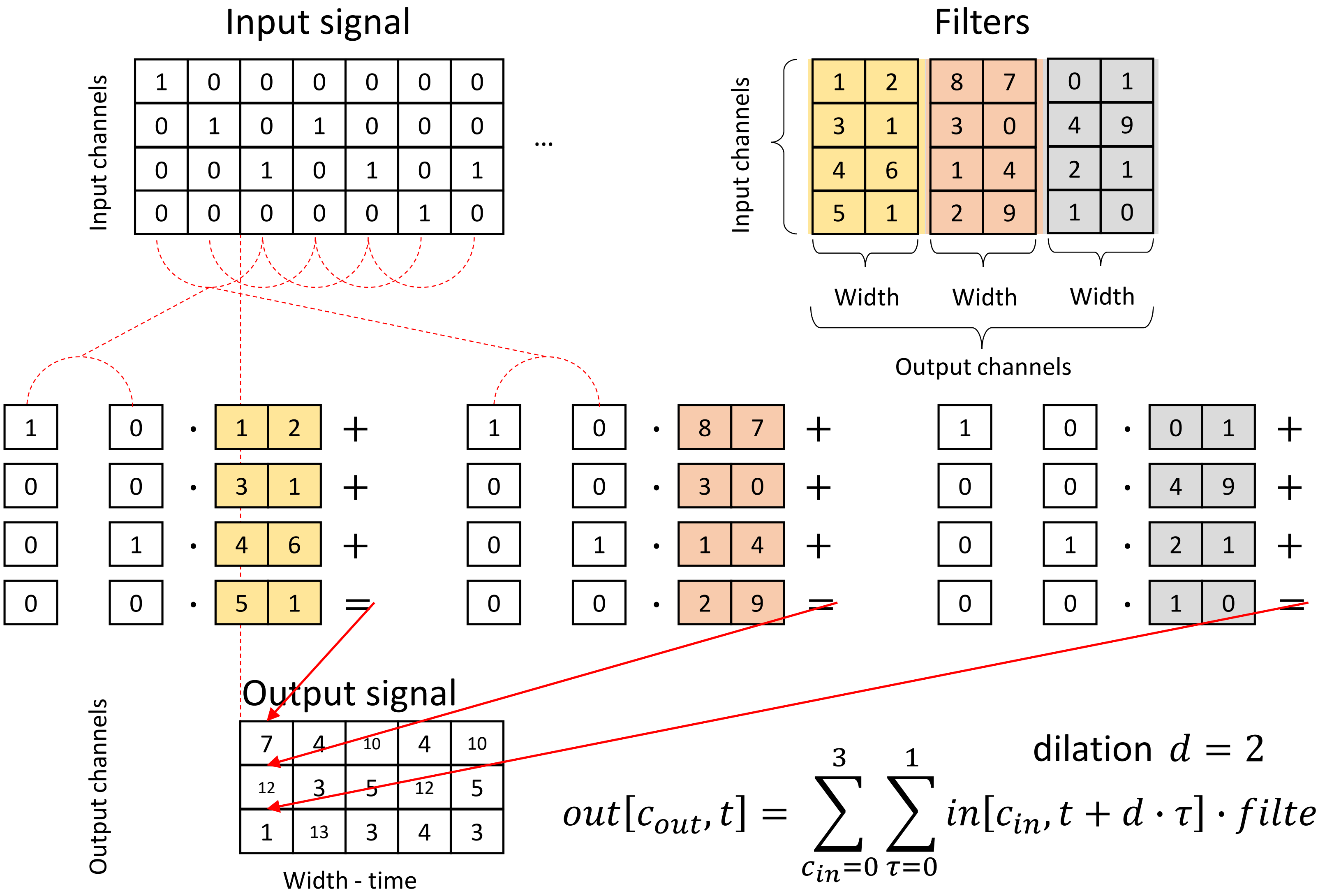
Output signal

2	9	5	9	5	11
8	7	1	7	10	6
9	5	11	5	2	2

$$out[c_{out}, t] = \sum_{c_{in}=0}^{3} \sum_{\tau=0}^{1} in[c_{in}, t+\tau] \cdot filter[c_{out}, c_{in}, \tau]$$

WaveNet architecture – Dilated convolutions

- Example of a **causal dilated** convolution of width 2, dilation 2, 4 input channels, and 3 output channels. Dilation is applied in time dimension



Input signal

1	0	0	0	0	0	0
0	1	0	1	0	0	0
0	0	1	0	1	0	1
0	0	0	0	0	1	0

Filters

1	2	8	7	0	1
3	1	3	0	4	9
4	6	1	4	2	1
5	1	2	9	1	0

Output signal

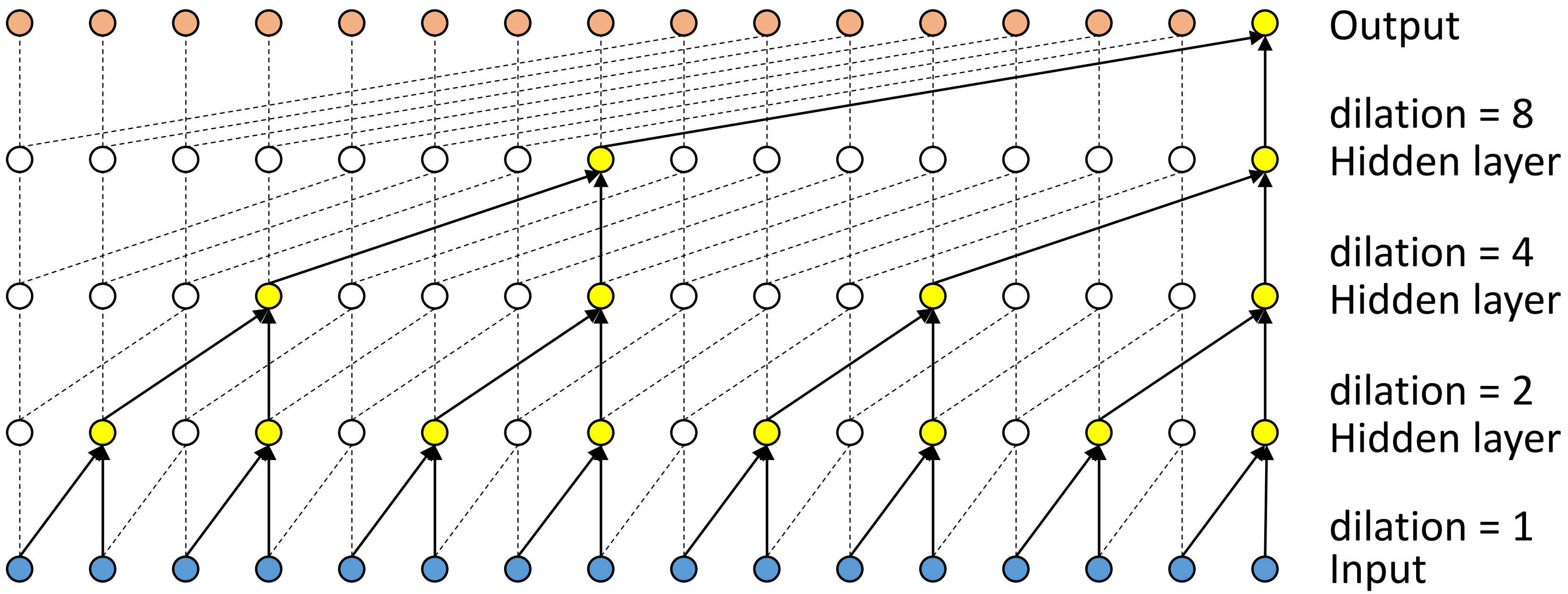
7	4	10	4	10
12	3	5	12	5
1	13	3	4	3

dilation $d = 2$

$$out[c_{out}, t] = \sum_{c_{in}=0}^{3} \sum_{\tau=0}^{1} in[c_{in}, t + d \cdot \tau] \cdot filter[c_{out}, c_{in}, \tau]$$

WaveNet architecture – Dilated convolutions

- WaveNet models the conditional probability distribution $p(x_t|x_1, \dots, x_{t-1})$ with a stack of dilated causal convolutions.

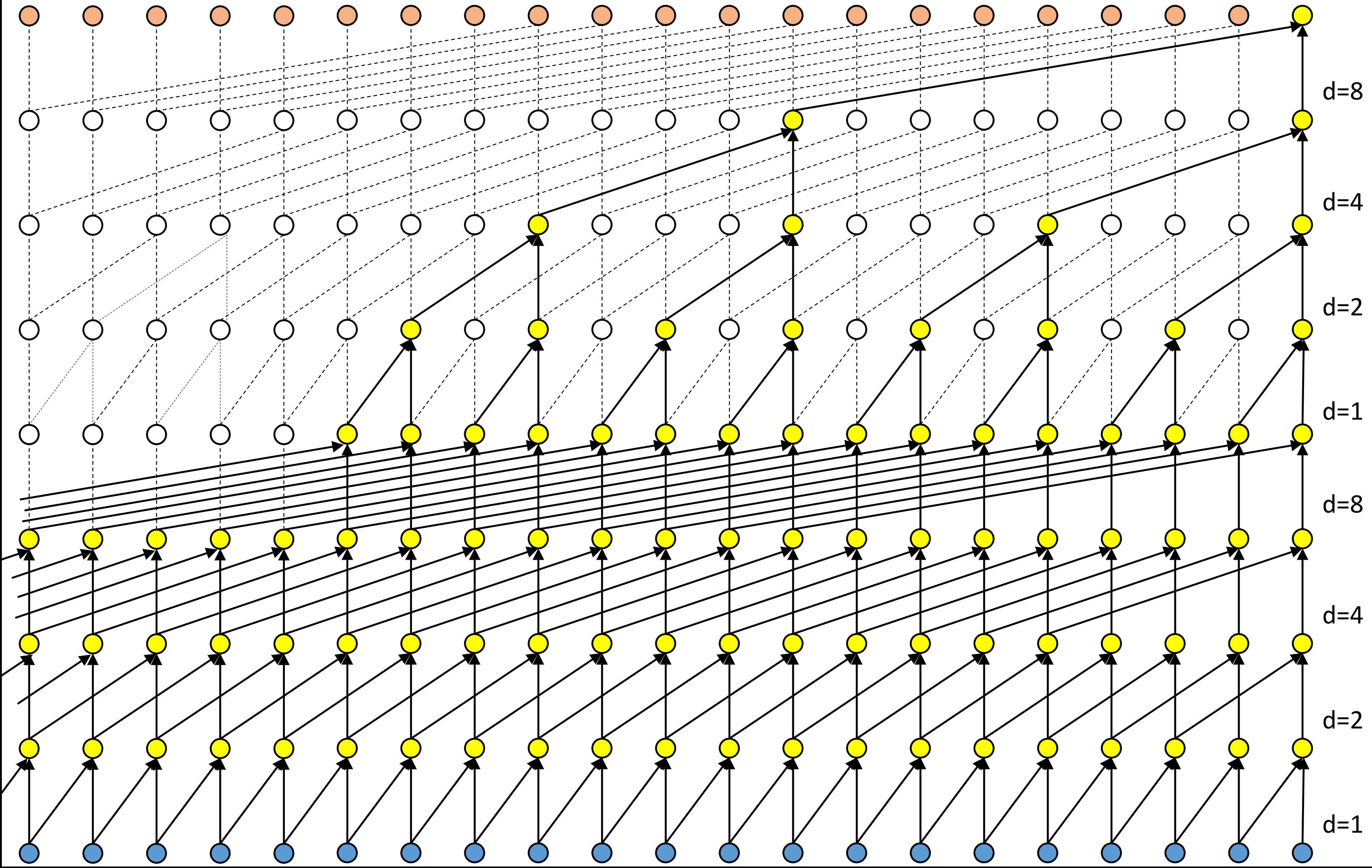


Visualization of a stack of dilated causal convolutional layers

- Stacked dilated convolutions enable very large receptive fields with just a few layers.
- In WaveNet, the dilation is doubled for every layer up to a certain point and then repeated: 1, 2, 4, ..., 512, 1, 2, 4, ..., 512, 1, 2, 4, ..., 512, 1, 2, 4, …, 512, 1, 2, 4, …, 512

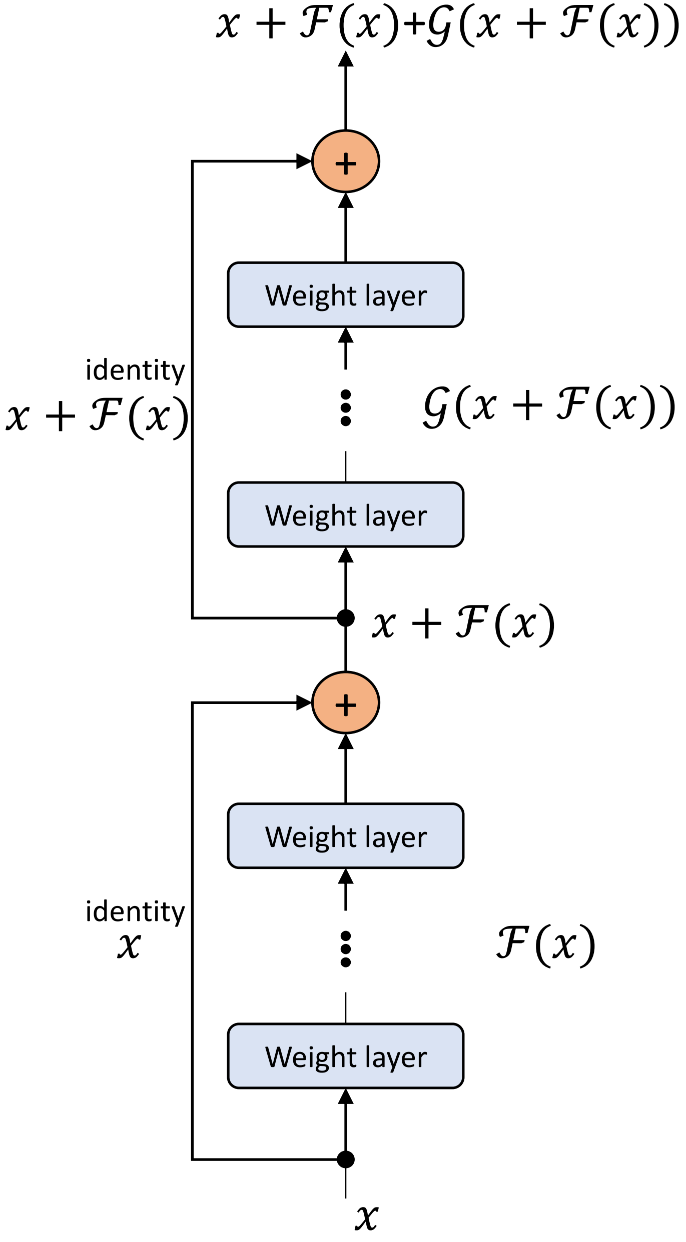
WaveNet architecture – Dilated convolutions

- Example with dilations 1,2,4,8,1,2,4,8



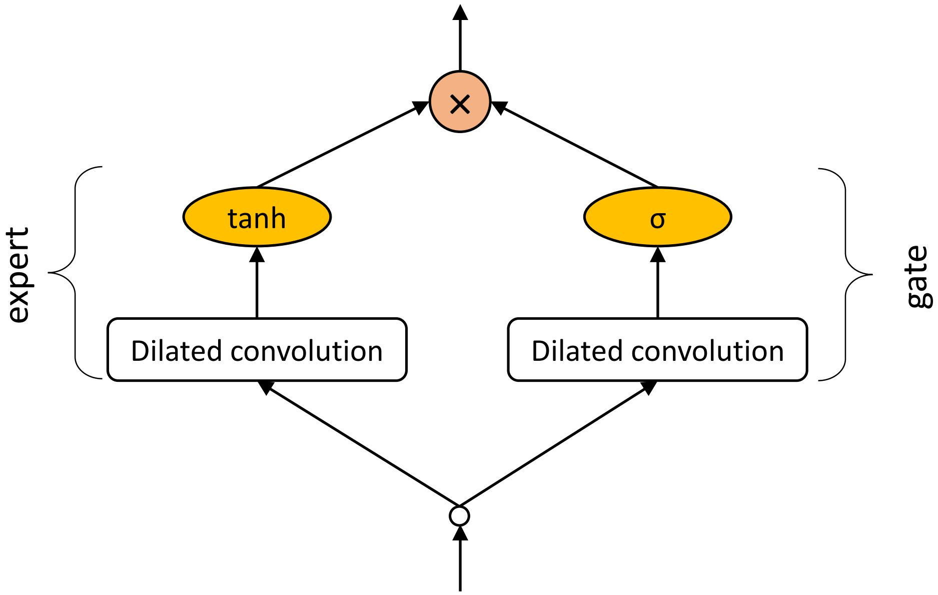
WaveNet architecture – Residual connections

- In order to train a WaveNet with more than 30 layers, residual connections are used.
- Residual networks were developed by researchers from Microsoft Research.
- They reformulated the mapping function, $x \rightarrow f(x)$, between layers from $f(x) = \mathcal{F}(x)$ to $f(x) = x + \mathcal{F}(x)$.
- The residual networks have identity mappings, x, as skip connections and inter-block activations $\mathcal{F}(x)$.
- Benefits
  - The residual $\mathcal{F}(x)$ can be more easily learned by the optimization algorithms.
  - The forward and backward signals can be directly propagated from one block to any other block.
  - The vanishing gradient problem is not a concern.



WaveNet architecture – Experts & Gates

- WaveNet uses gated networks.
- For each output channel an expert is defined.
  - Experts may specialize in different parts of the input space
- The contribution of each expert is controlled by a corresponding *gate* network.
- The components of the output vector are mixed in higher layers, creating mixture of experts.



WaveNet architecture – Post-processing

- WaveNet assigns to an input vector x_t a probability distribution using the softmax function.

$$h(z)_j = \frac{e^{z_j}}{\sum_{c=1}^{256} e^{z_c}}, \quad j = 1, \dots, 256$$

Channels

.6	.2	.1	.1	0
.2	.5	.1	.6	.1
.1	.2	.7	.2	.1
.1	.1	.1	.1	.8

time

WaveNet output:
probabilities from softmax

- The loss function used is the mean (across time) cross entropy.

$$H(in, out) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{256} in(c,t)\log(out(c,t))$$

WaveNet – Audio generation

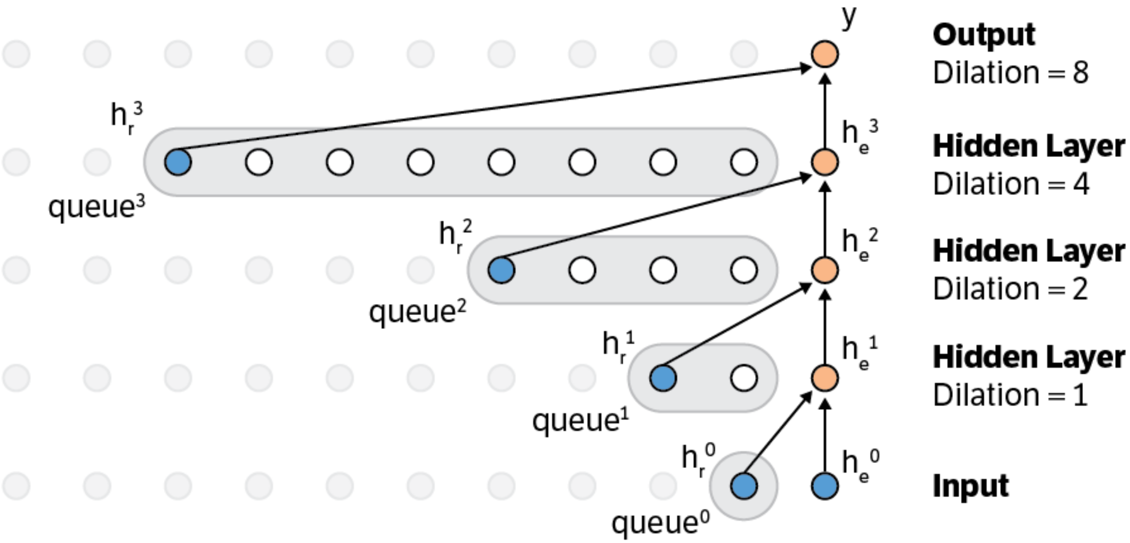
- After training, the network is sampled to generate synthetic utterances.
- At each step during sampling a value is drawn from the probability distribution computed by the network.
- This value is then fed back into the input and a new prediction for the next step is made.
- The output, $out,$ of the network is scaled back to speech with the inverse μ-law transformation.

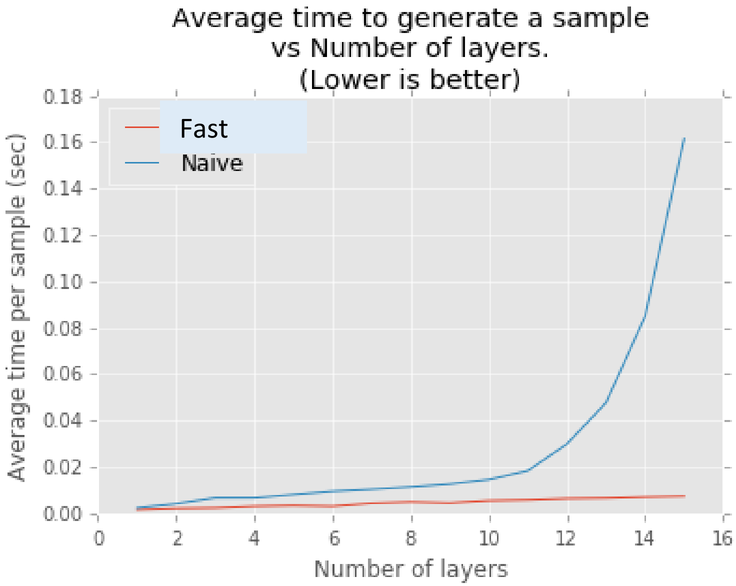
$$u = 2\frac{out}{\mu} - 1$$ From $out \in \{0,1,2,\dots,255\}$ to $u \in [-1,1]$

speech= $\frac{sign(u)}{\mu}\left((1+\mu)^u - 1\right)$ Inverse μ-law transform

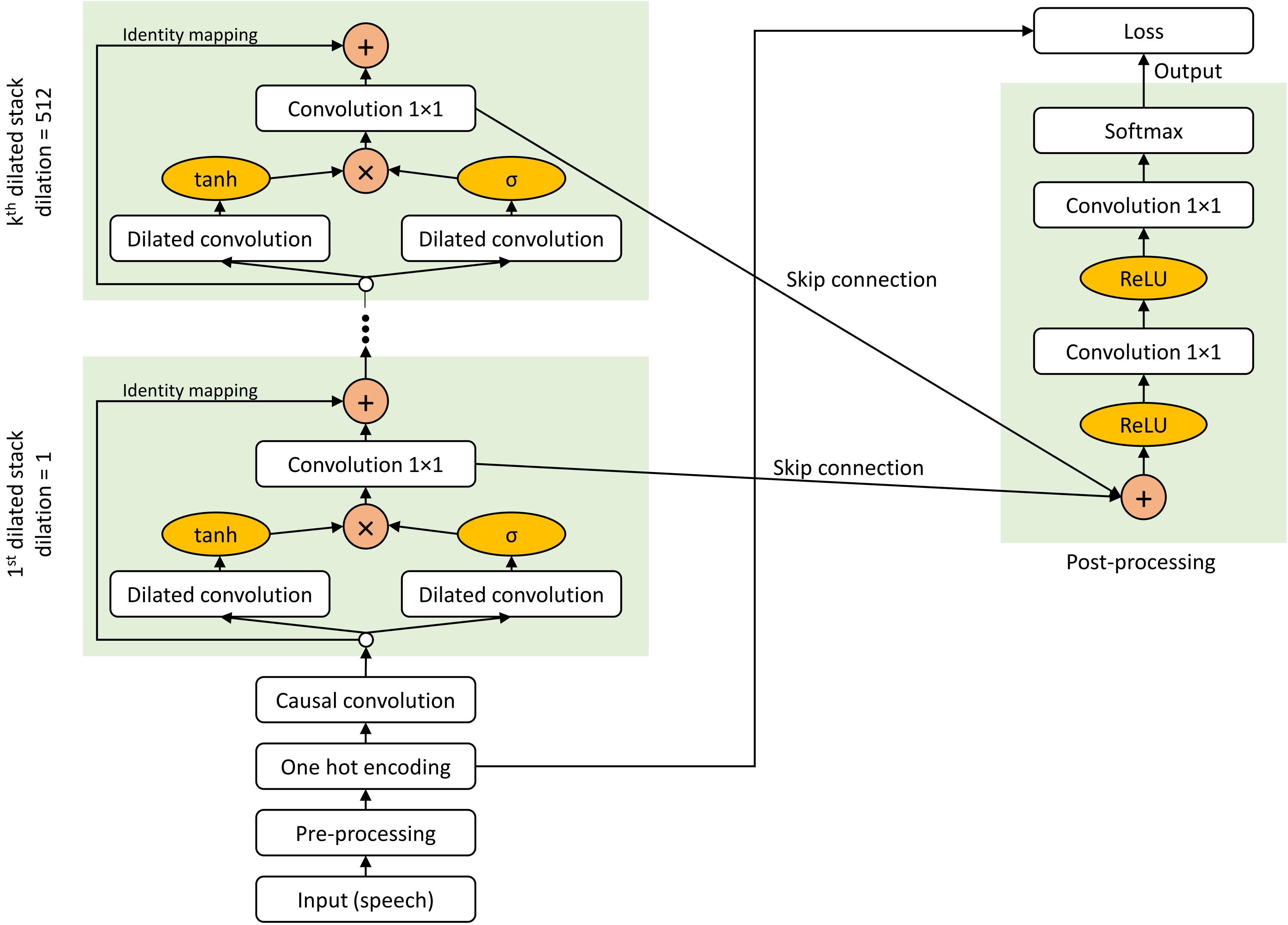
Fast WaveNet – Audio generation

- A naïve implementation of WaveNet generation requires time $O(2^L)$, where L is the number of layers.
- Recently, Tom Le Paine et al. have published their code for fast generation of sequences from trained WaveNets.
- Their algorithm uses queues to avoid redundant calculations of convolutions.
- This implementation requires time $O(L)$.

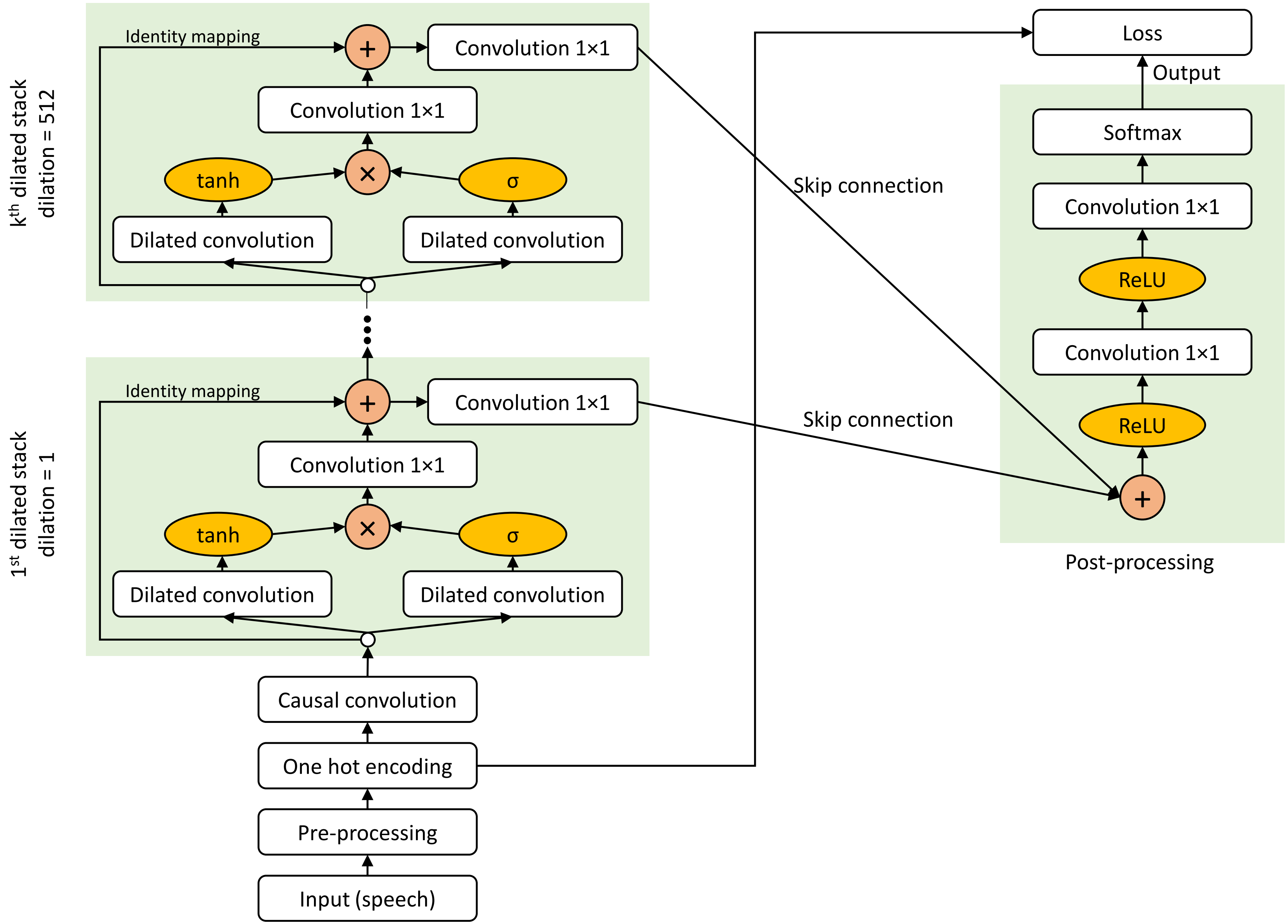




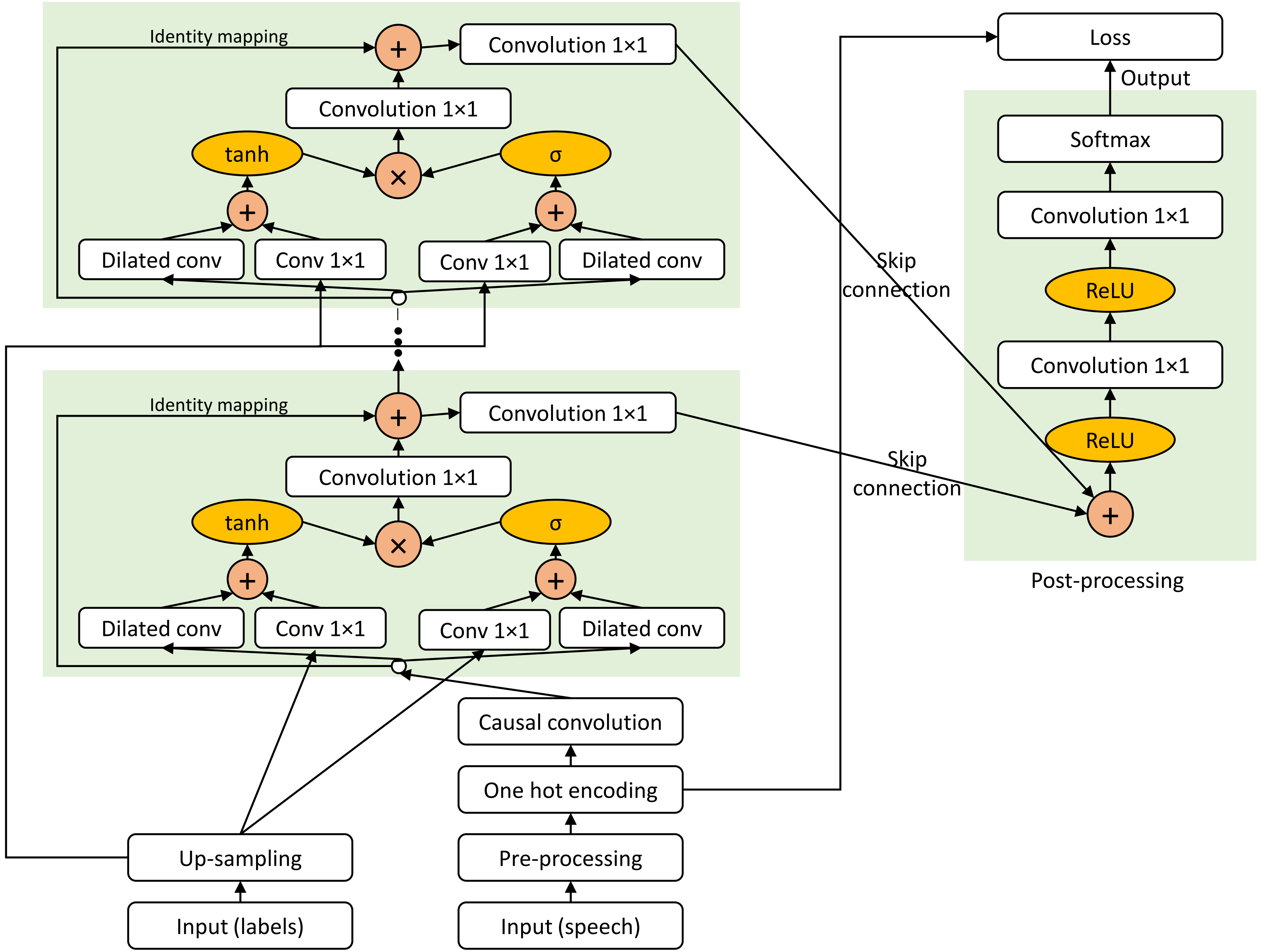
Basic WaveNet architecture - DeepMind



Basic WaveNet architecture – Un. Crete

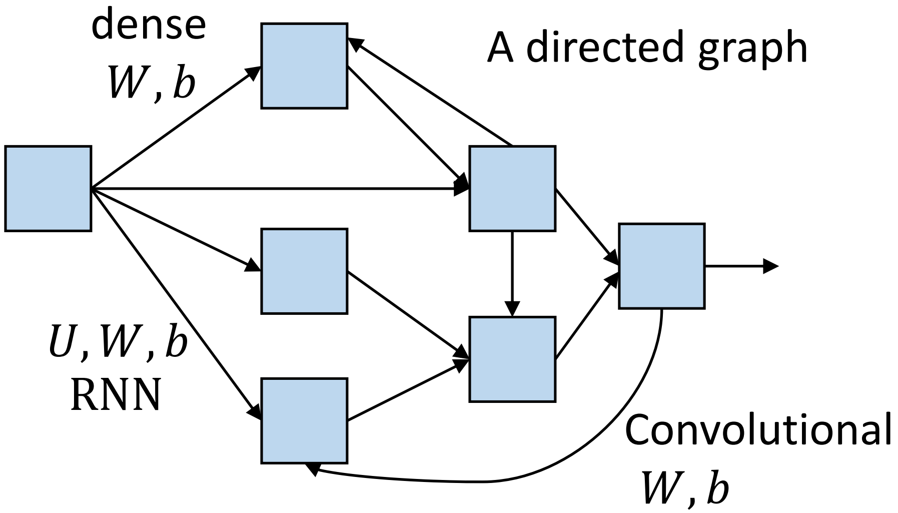


WaveNet architecture for TTS – Un. Crete



An implementation of WaveNet

- The NNARC library, which we build in the University of Crete, supports network architectures which are directed graphs.



- Due to this support the integration of WaveNet into NNARC was straight-forward.
- The only new components were the dilated causal convolutional layer and the data reader.