Сверточные сети для последовательностей

Шабан Махмуд, БПМИ171

Зачем, если есть RNN?

Проблемы RNN

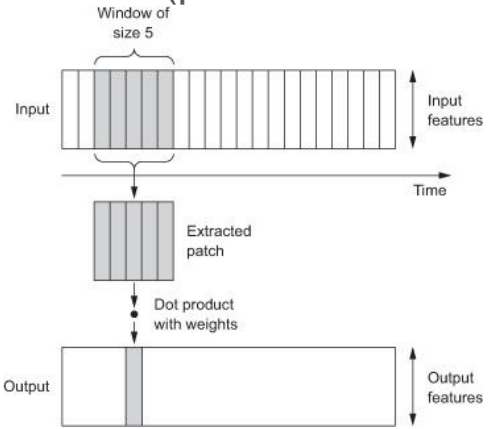
- Не параллелятся -> медленные
- Требуют много памяти для длинных последовательностей
- “Капризные” на практике (взрывы градиентов, затухание, прочее)

Плюсы CNN

- Работают значительно быстрее RNN
- Хорошо подходят для простых задач: классификация текстов, предсказание временных рядов
- Хорошо подходят для построения “представлений” предложений, нет необходимости в словаре эмбеддингов, более компактные представления

Концепты сверточных сетей с точки зрения последовательностей

- 1D свёртки - линейная комбинация нескольких соседних объектов
- Интуиция такая же: поиск локальных паттернов, независимо от положении в предложении (к примеру, отрицание чего либо (“не круто”) и т.д.)
- Pooling - абсолютно так же, уменьшаем размер входа, сохраняя наиболее важную информацию; также помогает избавиться от проблемы переменной длины предложения
- Каналы - вместо разных цветов могут быть разные представления текста (разные эмбеддинги, разные языки, и т.д.)

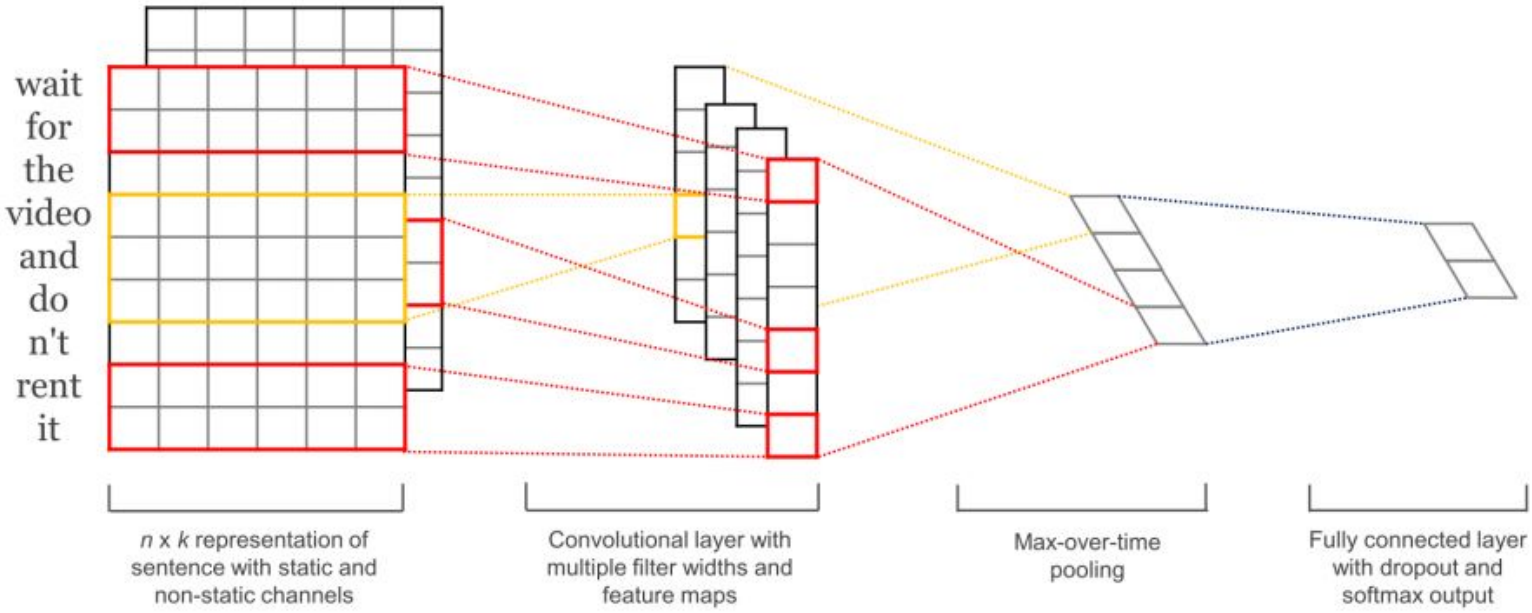


Важно понимать!

- 1D свёртки используют НЕ одномерные фильтры
- Фильтр 1D свёртки имеет размерность (w, d)
- w - ширина свёртки, т.е. кол-во объектов, указывается нами
- d - глубина свёртки, т.е. длина вектора-объекта, фиксирована
- На выходе имеем вектора размерности, равной количеству фильтров.

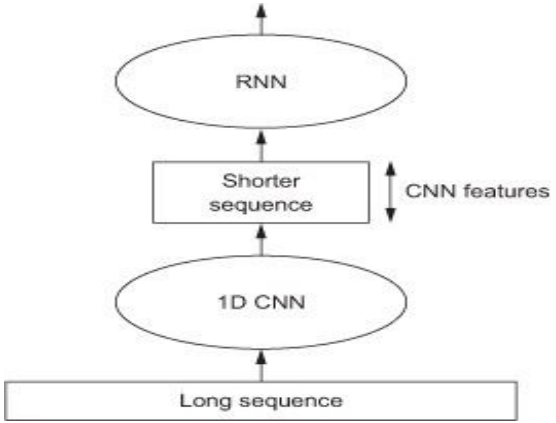
Пример применения

- Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification
- SOTA на нескольких классификационных датасетах на то время



CNN + LSTM

- Проблема 1D CNN - нет чувствительности к позиции элементов последовательности, что бывает критично (например, в задаче предсказания температуры).
- Совместим скорость CNN и чувствительность к порядку RNN
- Применяем одномерную сверточную сеть на последовательности - получаем гораздо менее длинную последовательность с высокоуровневыми признаками
- На полученной последовательности - GRU или LSTM
- Работает в 2 раза быстрее и почти так же по качеству, как “ванильная” RNN



Temporal Convolutional Networks

- Универсальная архитектура, обгоняющая GRU/LSTM на широком спектре задач
- Гибко параметризуемый receptive field
- Малое потребление памяти
- Поддержка ввода переменного размера

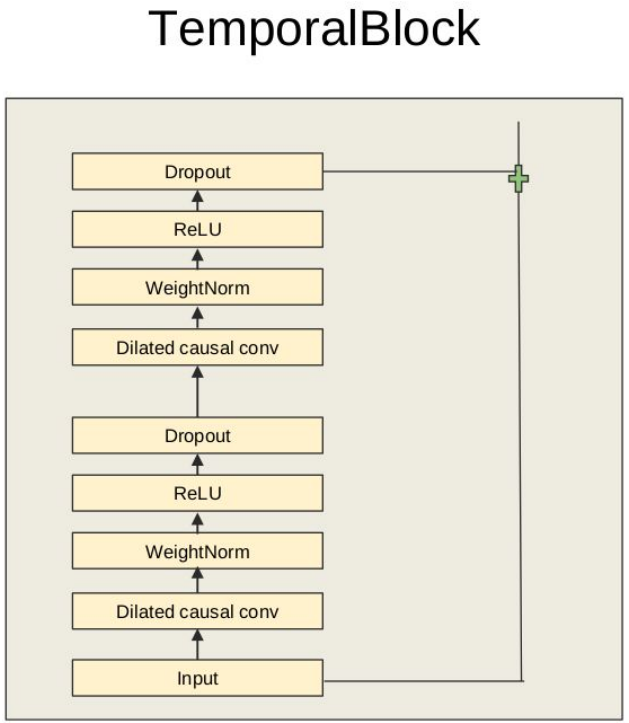
Вкратце: 1D FCN + causal convolutions

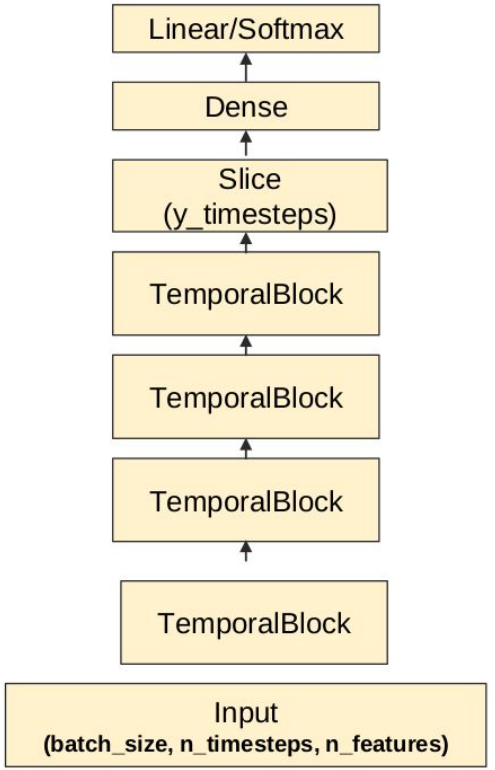
Temporal Convolutional Networks

Table 1. Evaluation of TCNs and recurrent architectures on synthetic stress tests, polyphonic music modeling, character-level language modeling, and word-level language modeling. The generic TCN architecture outperforms canonical recurrent networks across a comprehensive suite of tasks and datasets. Current state-of-the-art results are listed in the supplement. ^{h} means that higher is better. $^{\ell}$ means that lower is better.

Sequence Modeling Task	Model Size ($\approx$)	Models			
		LSTM	GRU	RNN	**TCN**
Seq. MNIST (accuracy^{h})	70K	87.2	96.2	21.5	**99.0**
Permuted MNIST (accuracy)	70K	85.7	87.3	25.3	**97.2**
Adding problem T=600 (loss$^{\ell}$)	70K	0.164	**5.3e-5**	0.177	**5.8e-5**
Copy memory T=1000 (loss)	16K	0.0204	0.0197	0.0202	**3.5e-5**
Music JSB Chorales (loss)	300K	8.45	8.43	8.91	**8.10**
Music Nottingham (loss)	1M	3.29	3.46	4.05	**3.07**
Word-level PTB (perplexity$^{\ell}$)	13M	**78.93**	92.48	114.50	88.68
Word-level Wiki-103 (perplexity)	-	48.4	-	-	**45.19**
Word-level LAMBADA (perplexity)	-	4186	-	14725	**1279**
Char-level PTB (bpc$^{\ell}$)	3M	1.36	1.37	1.48	**1.31**
Char-level text8 (bpc)	5M	1.50	1.53	1.69	**1.45**

Temporal Convolutional Networks



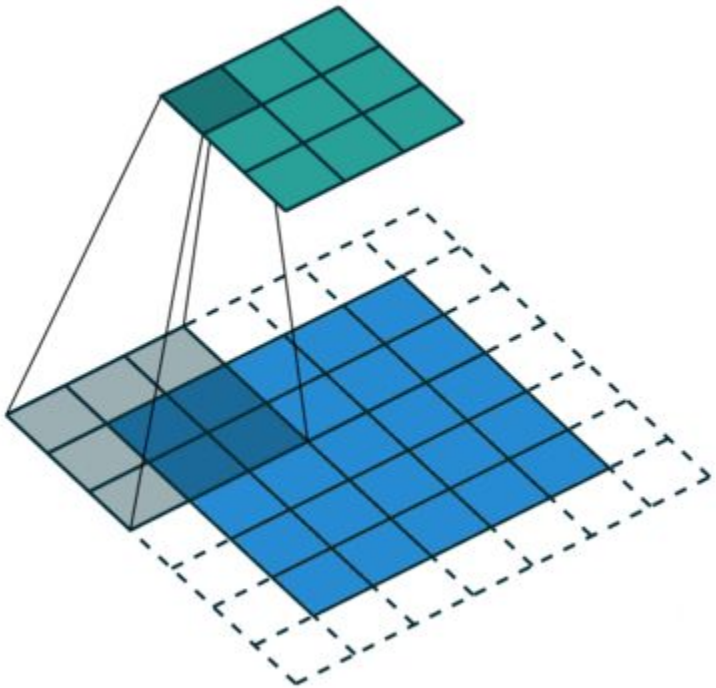


Каузальные (causal) свёртки

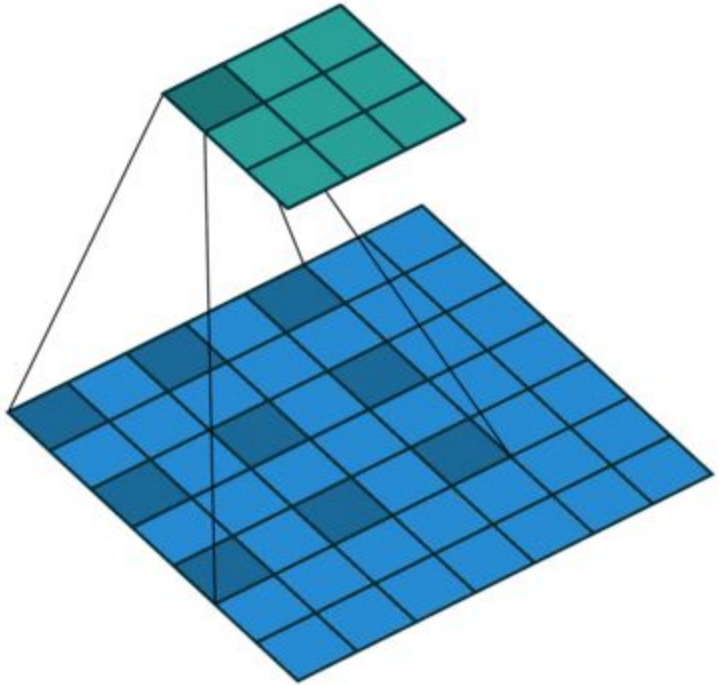
- При работе с временными рядами мы не можем “заглядывать в будущее”
- Необходимо реализовать свёртку, которая выводит элемент последовательности b[i] только на основе a[i], a[i - 1], …
- Проще всего это сделать паддингом: слева добавляем количество нулей, равное ширине свёртки - 1, справа не добавляем ничего

Разреженные (dilated) свёртки

dilation_rate=1



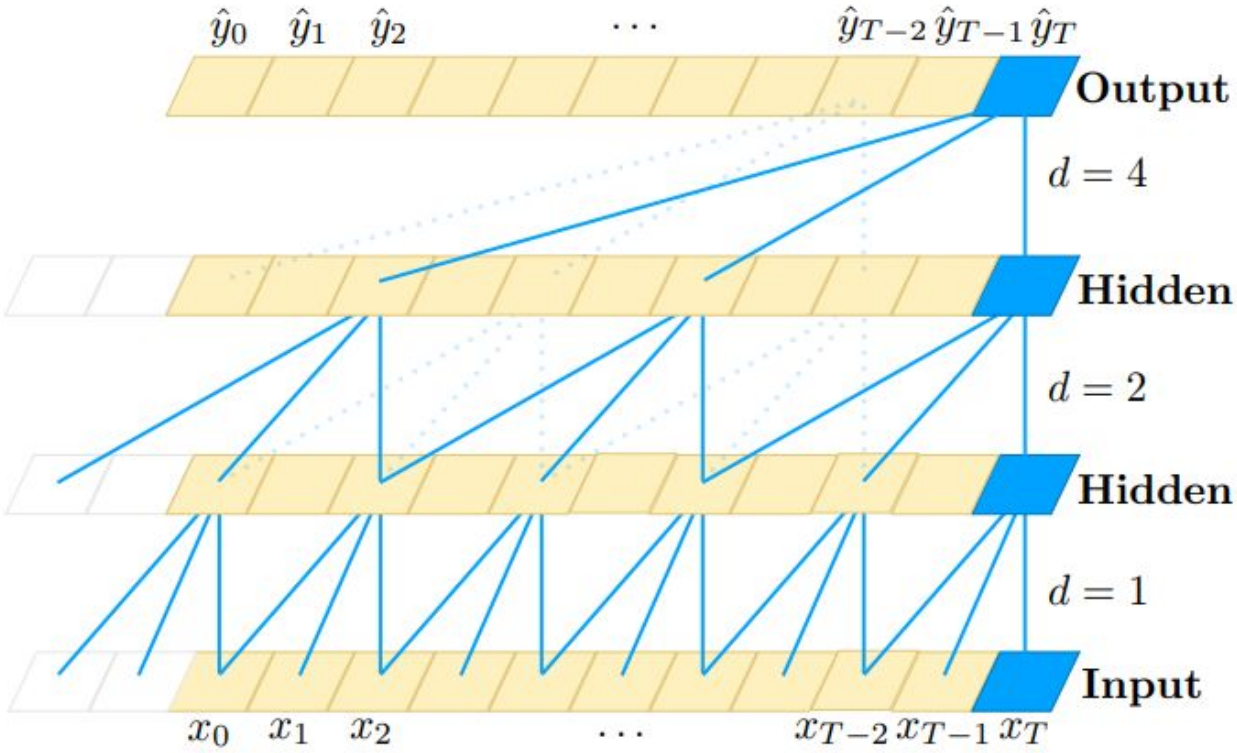
dilation_rate=2



Разреженные (dilated) свёртки

- В чём суть такого разреживания?
- Ответ простой: увеличение receptive field
- Обычно используется так: каждый последующий сверточный слой имеет dilation_rate в 2 раза больше предыдущего
- Такой трюк позволяет нам эффективно смотреть на большое количество объектов последовательности - reception field растет экспоненциально относительно глубины сети (при обычных свертках линейно).

Разреженные (dilated) свёртки

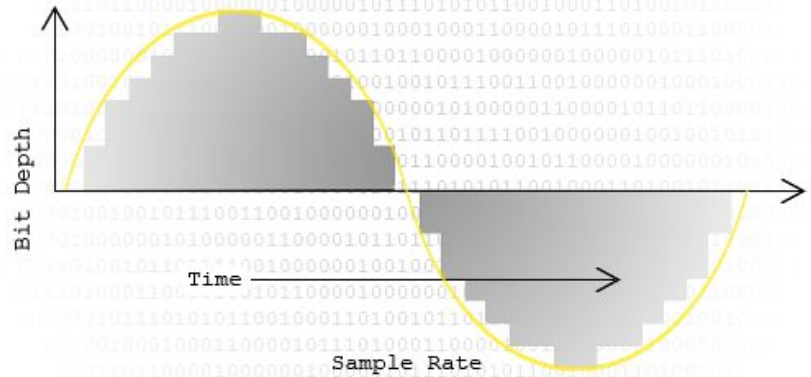


Недостатки TCN

- Необходимость хранить последовательность целиком в процессе обучения - особенность сверточных сетей
- Проблемы с domain adaptation: обученную модель будет сложно переиспользовать, если данные требуют учитывания более длинного контекста (ибо это параметризует архитектуру модели)

“Сырой” звук

- Имеем дело с сигналом
- Это, внезапно, временной ряд (ибо мы строим приближение волн)
- Разрядность задает количество значений, которое звук может принимать в конкретный момент времени
- Частота дискретизации задаёт количество событий в секунду (например, 44.1 kHz задаёт 44100 событий в секунду)

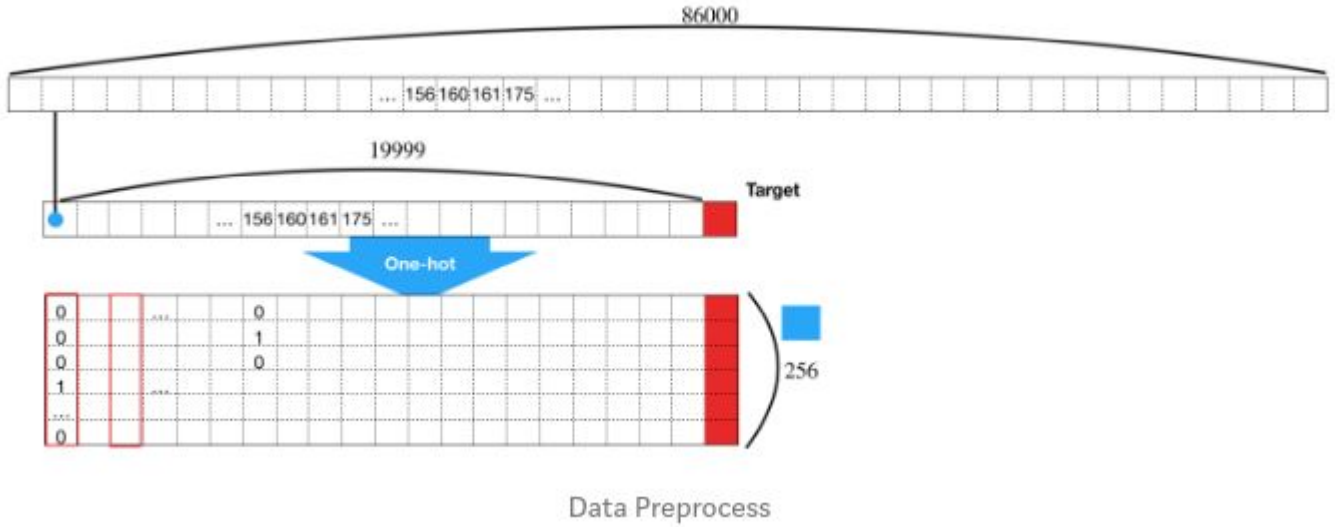


WaveNet (DeepMind, 2016)

- WaveNet является моделью для генерации “сырого” звука
- Задача сводится к предсказанию временных рядов
- Проблема: ОЧЕНЬ много данных
- Решение: разреженные каузальные свертки, которые мы уже разобрали
- Это помогает решить проблему необходимости учета огромного контекста

WaveNet (DeepMind, 2016)

- Будем обучаться с размером батча 20000 (выберем непрерывный кусок из данных)
- Классы кодируем в one-hot



WaveNet (DeepMind, 2016)

- k - количество разреживаний
- Архитектура подобна ResNet - та же логика с residual connections, блоки последовательные, k штук

WaveNet (DeepMind, 2016)

2.4 RESIDUAL AND SKIP CONNECTIONS

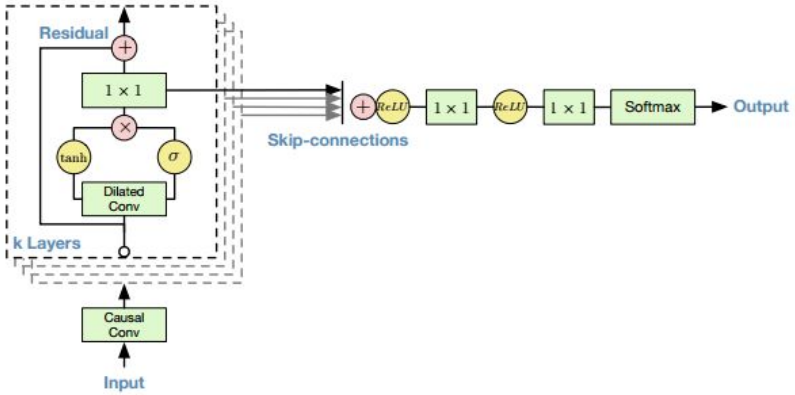
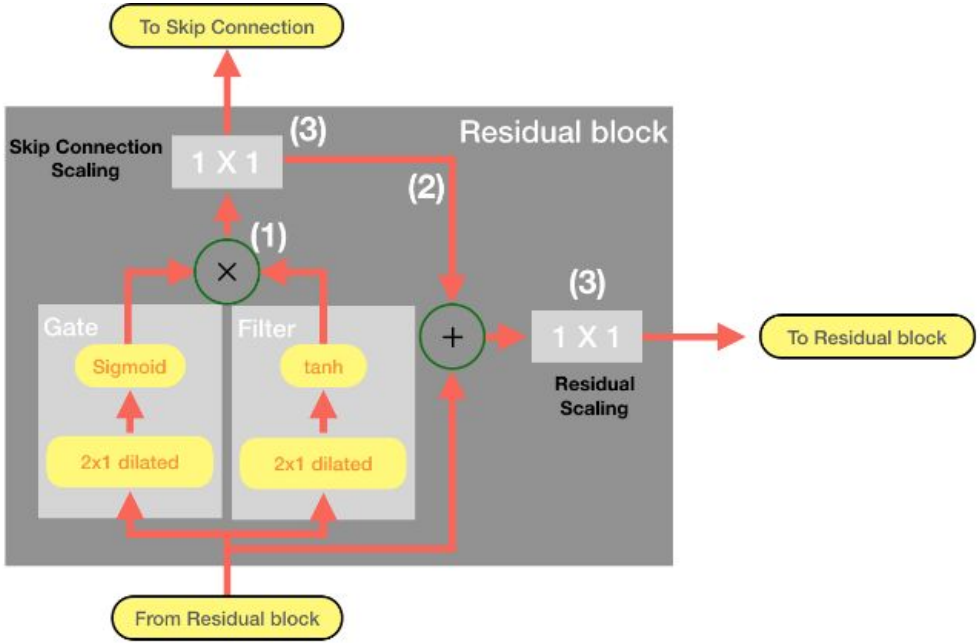


Figure 4: Overview of the residual block and the entire architecture.

Both residual (He et al., 2015) and parameterised skip connections are used throughout the network, to speed up convergence and enable training of much deeper models. In Fig. 4 we show a residual block of our model, which is stacked many times in the network.

WaveNet (DeepMind, 2016)



Выводы

- **1D CNN** - быстрая альтернатива RNN для последовательностей, отлично подходит для простых задач, однако плохо подходит для трудных по причине нечувствительности к порядку.
- **TCN** - архитектура исключительно на свертках, оказывающая серьезную конкуренцию RNN благодаря концепту **разреженных каузальных сверток**.
- **WaveNet** - мощная архитектура для генерации аудио, актуальная по сегодняшний день.

Спасибо за внимание!

Вопросы

1. Каким образом происходит слияние архитектур CNN и RNN? Какой профит от использования слияния подходов для последовательности?
2. В чем смысл (принцип работы) causal свёрток? Почему при работе с предсказаниями они подходят лучше обычных свёрток?
3. Что такое разреженные свёртки? Чем они отличаются от обычных сверток с увеличенным шагом? Какое у них преимущество перед обычными свертками?

Источники

1. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling (https://arxiv.org/pdf/1803.01271.pdf)
2. Understanding Convolutional Neural Networks for NLP (http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/)
3. WaveNet: A Generative Model for Raw Audio (https://arxiv.org/pdf/1609.03499.pdf)
4. Francois Chollet “Deep Learning with Python”