Pay less attention with lightweight and dynamic convolutions

Radek Bartyzal

Let's talk ML in Prague

11. 4. 2019

Classic Convolutions

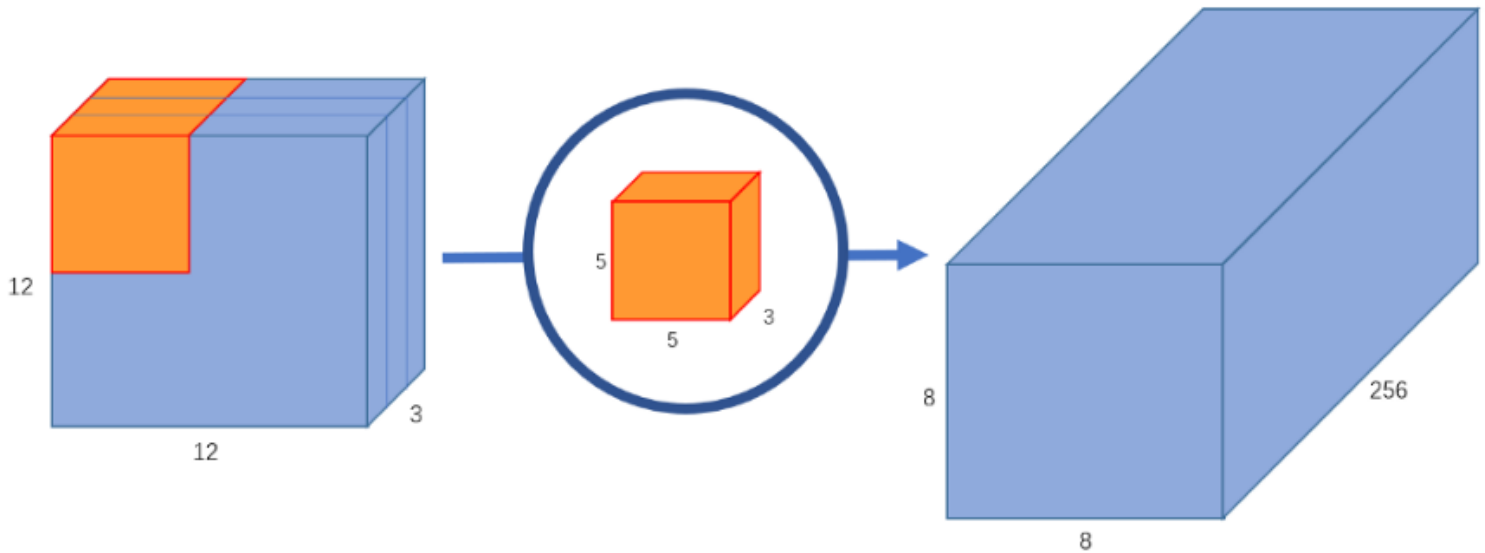


Figure: Each kernel has $k \times d$ weights. To have output of dim $= F$ we need F filters $\implies$ $F \times k \times d$ kernel weights.

Depthwise separable Convolutions

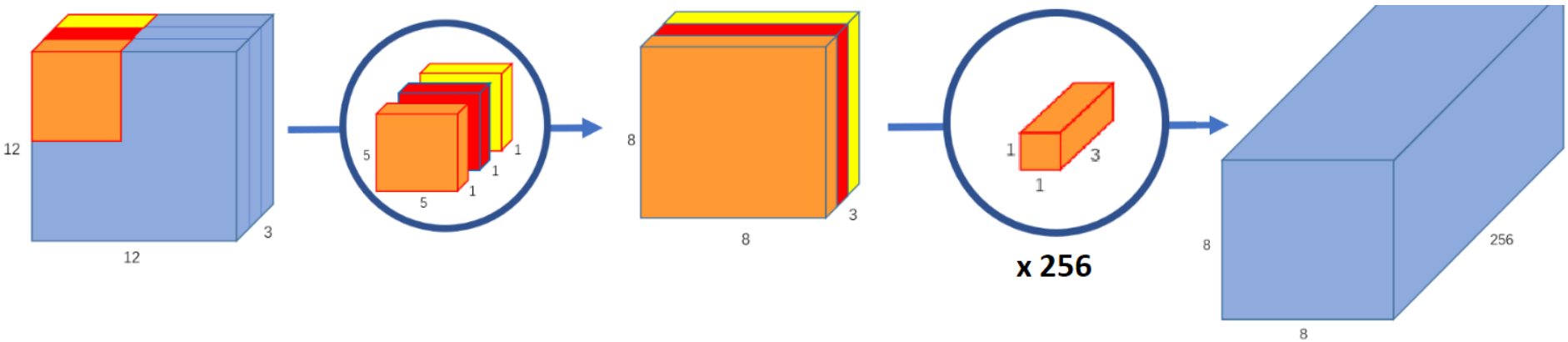


Figure: Use kernels of size $k \times 1$ and slide it over each channel separately = output has same number of channels as input. If we want different number of output channels (e.g. F) we can use F efficient $1 \times 1 \times d$ filters. Resulting number of kernel weights is either just $k \times 1 \times d$ or $k \times 1 \times d + 1 \times 1 \times d \times F = k \times d + F \times d$.

NLP case

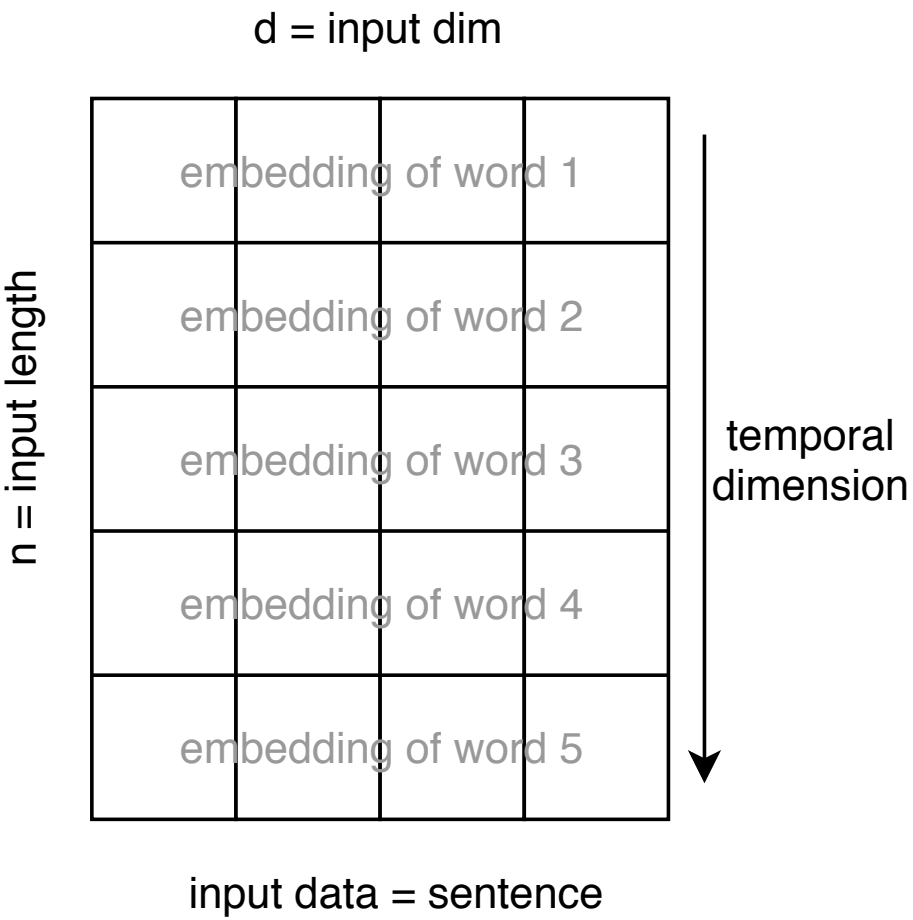
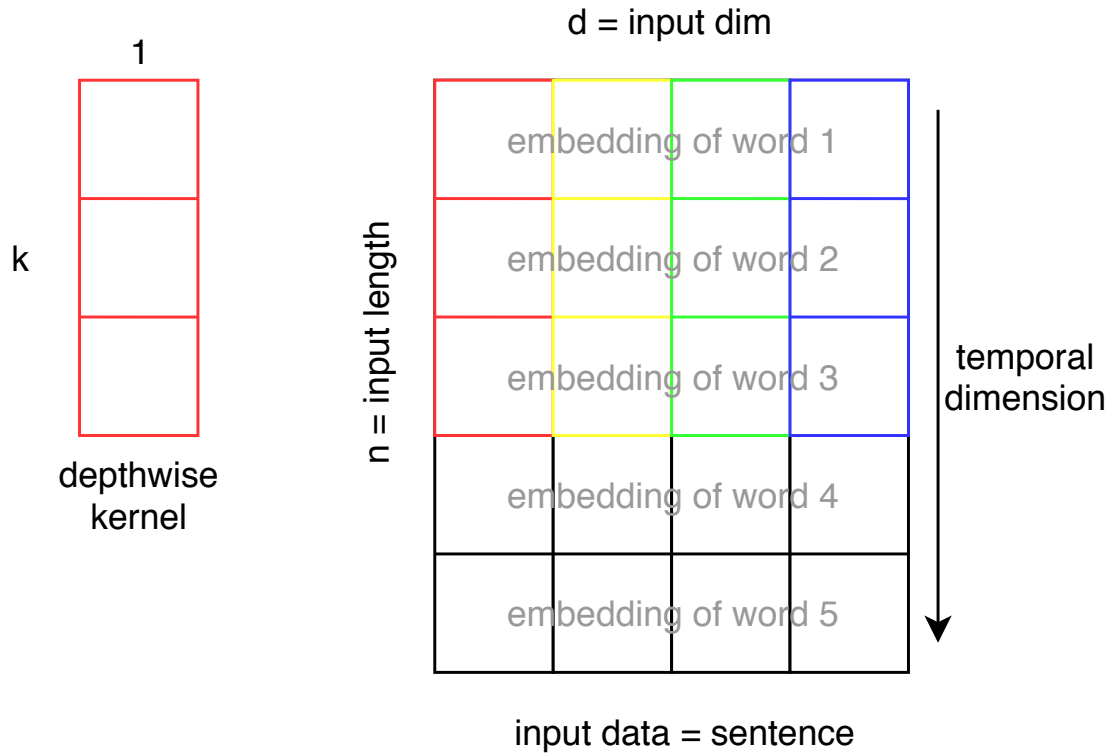


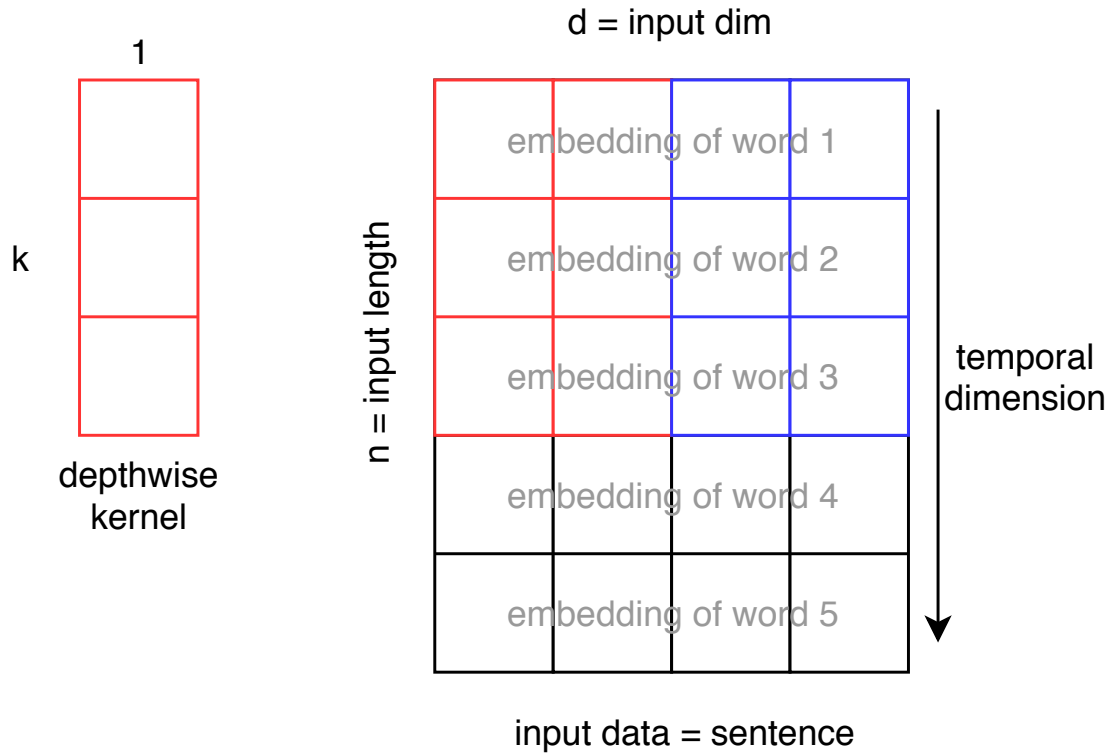
Figure: Input sentence as a matrix $X \in \mathbb{R}^{n \times d}$.

Depth-wise separable



- output dim $= d =$ input dim $=$ number of channels
- each channel has different kernel $= d$ kernels
- num kernel weights $= d \times k \times 1$

Lightweight convolutions



- output dim $= d =$ input dim $=$ number of channels
- $\frac{d}{H}$ channels share kernel weights $= H$ kernels
- num kernel weights $= H \times k \times 1$
- each kernel is softmaxed before being applied

Lightweight and dynamic convolutions

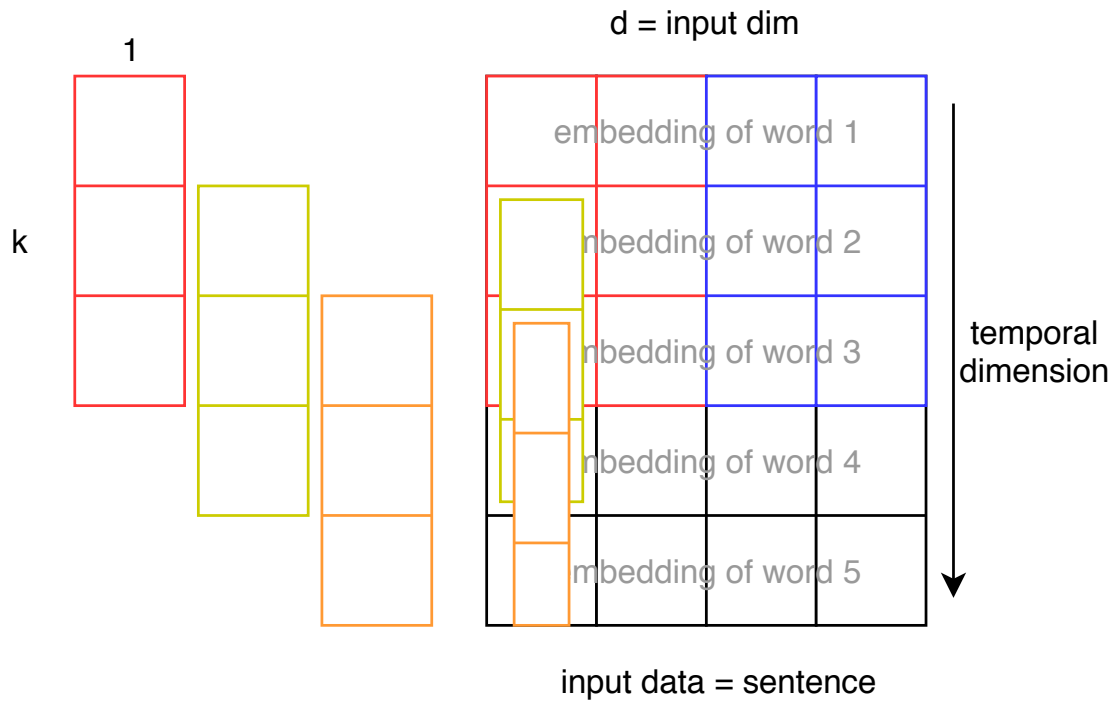
$$O_{i,c} = \text{DepthwiseConv}(X, W_{c,:}, i, c) = \sum_{j=1}^{k} W_{c,j} \cdot X_{(i+j-\lceil \frac{k+1}{2} \rceil),c}$$

$$\text{LightConv}(X, W_{\lceil \frac{cH}{d} \rceil,:}, i, c) = \text{DepthwiseConv}(X, \text{softmax}(W_{\lceil \frac{cH}{d} \rceil,:}), i, c)$$

$$\text{DynamicConv}(X, i, c) = \text{LightConv}(X, f(X_i)_{h,:}, i, c)$$

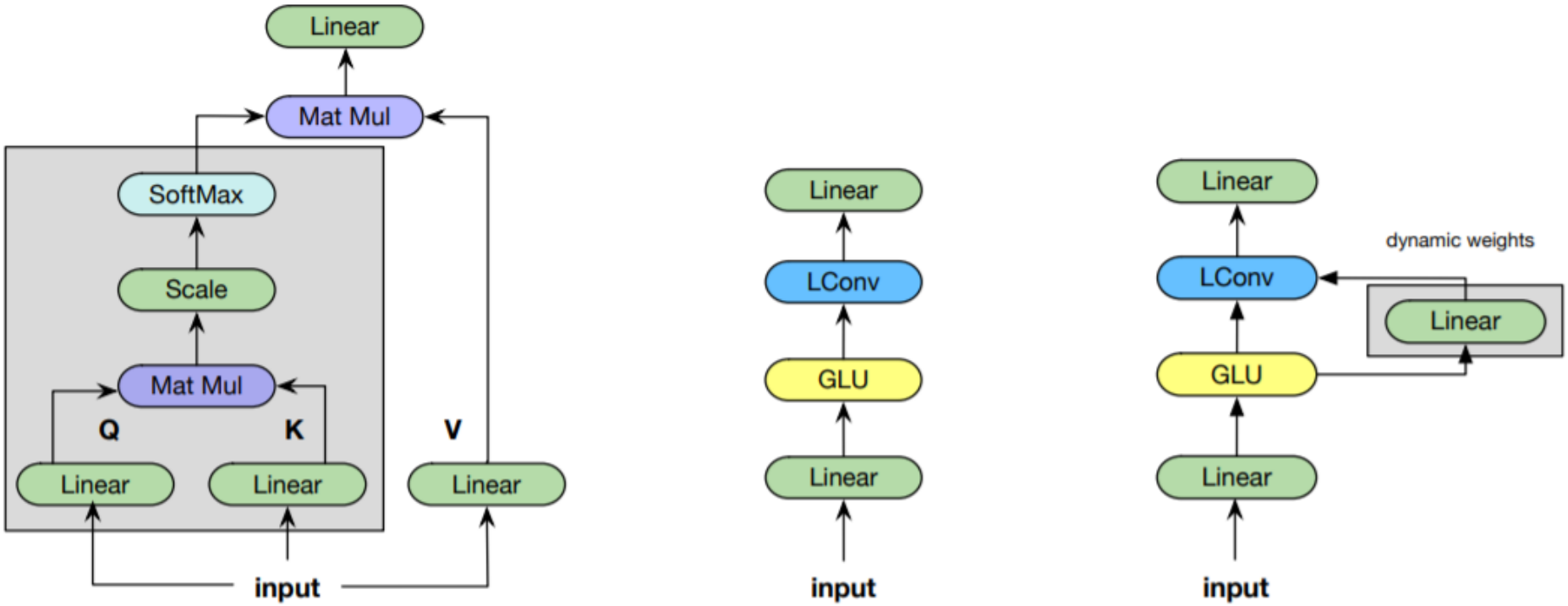
- dynamic = add dense layer based on k elements that generates the kernel = the kernel changes as is slides over the temporal dimension
- dynamic weights are a function of the current time-step only rather than the entire context

Dynamic convolutions



- kernel at each timestep is generated by a dense layer
- = as kernel slides over the temporal dimension it changes

Comparison to attention



(a) Self-attention (b) Lightweight convolution (c) Dynamic convolution

Lightweight and dynamic convolutions

- pay "attention" only to k surrounding elements
- ends with output projection to correct dim by linear layer

Experiments

- Transformer network - replace self-attention modules with LConv or Dynamic Conv
- LConv, DConv use less params = increase number of encoder blocks to $N = 7$ to match number of params
- $H = 16$
- encoder/decoder block kernels: $3, 7, 15, 31 \times 4$
- top 3 layers of decoder have kernel size 31
- We train three random initializations of a each configuration and report test accuracy of the seed which resulted in the highest validation BLEU
- machine translation, lang. modeling, summarization

Results

Model	Param (En-De)	WMT En-De	WMT En-Fr
Gehring et al. (2017)	216M	25.2	40.5
Vaswani et al. (2017)	213M	28.4	41.0
Ahmed et al. (2017)	213M	28.9	41.4
Chen et al. (2018)	379M	28.5	41.0
Shaw et al. (2018)	-	29.2	41.5
Ott et al. (2018)	210M	29.3	**43.2**
LightConv	202M	28.9	43.1
DynamicConv	213M	**29.7**	**43.2**

Figure: Machine translation accuracy in terms of BLEU for WMT En-De and WMT En-Fr on newstest2014.

Results

Model	Param (Zh-En)	IWSLT	WMT Zh-En
Deng et al. (2018)	-	33.1	-
Hassan et al. (2018)	-	-	24.2
Self-attention baseline	292M	34.4	23.8
LightConv	285M	34.8	24.3
DynamicConv	296M	**35.2**	**24.4**

Figure: Machine translation accuracy in terms of BLEU on IWSLT and WMT Zh-En.

Results

Model	Param	BLEU	Sent/sec
Vaswani et al. (2017)	213M	26.4	-
Self-attention baseline (k=inf, H=16)	210M	26.9 ± 0.1	52.1 ± 0.1
Self-attention baseline (k=3,7,15,31x3, H=16)	210M	26.9 ± 0.3	54.9 ± 0.2
CNN (k=3)	208M	25.9 ± 0.2	68.1 ± 0.3
CNN Depthwise (k=3, H=1024)	195M	26.1 ± 0.2	67.1 ± 1.0
+ Increasing kernel (k=3,7,15,31x4, H=1024)	195M	26.4 ± 0.2	63.3 ± 0.1
+ DropConnect (H=1024)	195M	26.5 ± 0.2	63.3 ± 0.1
+ Weight sharing (H=16)	195M	26.5 ± 0.1	63.7 ± 0.4
+ Softmax-normalized weights [LightConv] (H=16)	195M	26.6 ± 0.2	63.6 ± 0.1
+ Dynamic weights [DynamicConv] (H=16)	200M	26.9 ± 0.2	62.6 ± 0.4
Note: DynamicConv(H=16) w/o softmax-normalization	200M	diverges	
AAN decoder + self-attn encoder	260M	26.8 ± 0.1	59.5 ± 0.1
AAN decoder + AAN encoder	310M	22.5 ± 0.1	59.2 ± 2.1

Figure: Ablation on WMT English-German newstest2013. (+) indicates that a result includes all preceding features. Speed results based on beam size 4, batch size 256 on an NVIDIA P100 GPU.

Results

Model	Param	Valid	Test
2-layer LSTM-8192-1024 (Józefowicz et al., 2016)	–	–	30.6
Gated Convolutional Model (Dauphin et al., 2017)	428M	–	31.9
Mixture of Experts (Shazeer et al., 2017)	4371M †	–	28.0
Self-attention baseline	331M	26.67	26.73
DynamicConv	339M	26.60	**26.67**

Figure: Language modeling results on the Google Billion Word test set. + does not include embedding and softmax layers

Results

Model	Param	Rouge-1	Rouge-2	Rouge-l
LSTM (Paulus et al., 2017)	-	38.30	14.81	35.49
CNN (Fan et al., 2017)	-	39.06	15.38	35.77
Self-attention baseline	90M	39.26	15.98	36.35
LightConv	86M	39.52	15.97	36.51
DynamicConv	87M	**39.84**	**16.25**	**36.73**
RL (Celikyilmaz et al., 2018)	-	**41.69**	**19.47**	**37.92**

Figure: Results on CNN-DailyMail summarization. We compare to likelihood trained approaches except for Celikyilmaz et al. (2018).

Sources

1. Wu, Felix, et al. "Pay Less Attention with Lightweight and Dynamic Convolutions." arXiv preprint arXiv:1901.10430 (2019).
https://arxiv.org/abs/1901.10430

2. Medium blogpost on depthwise convolutions.
https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728