10707
Deep Learning

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu

Sequence Model / Transformers

Slides borrowed from ICML Tutorial

Seq2Seq ICML Tutorial

Oriol Vinyals and Navdeep Jaitly

@OriolVinyalsML | @NavdeepLearning

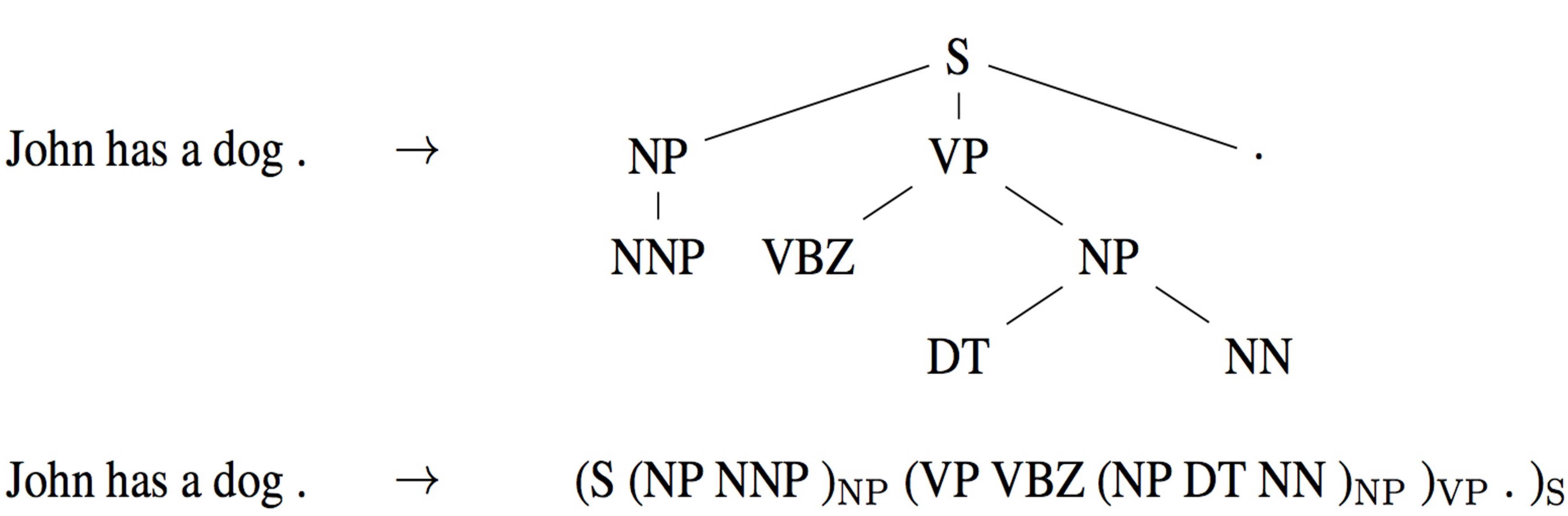
Site: https://sites.google.com/view/seq2seq-icml17

Sydney, Australia, 2017

Applications

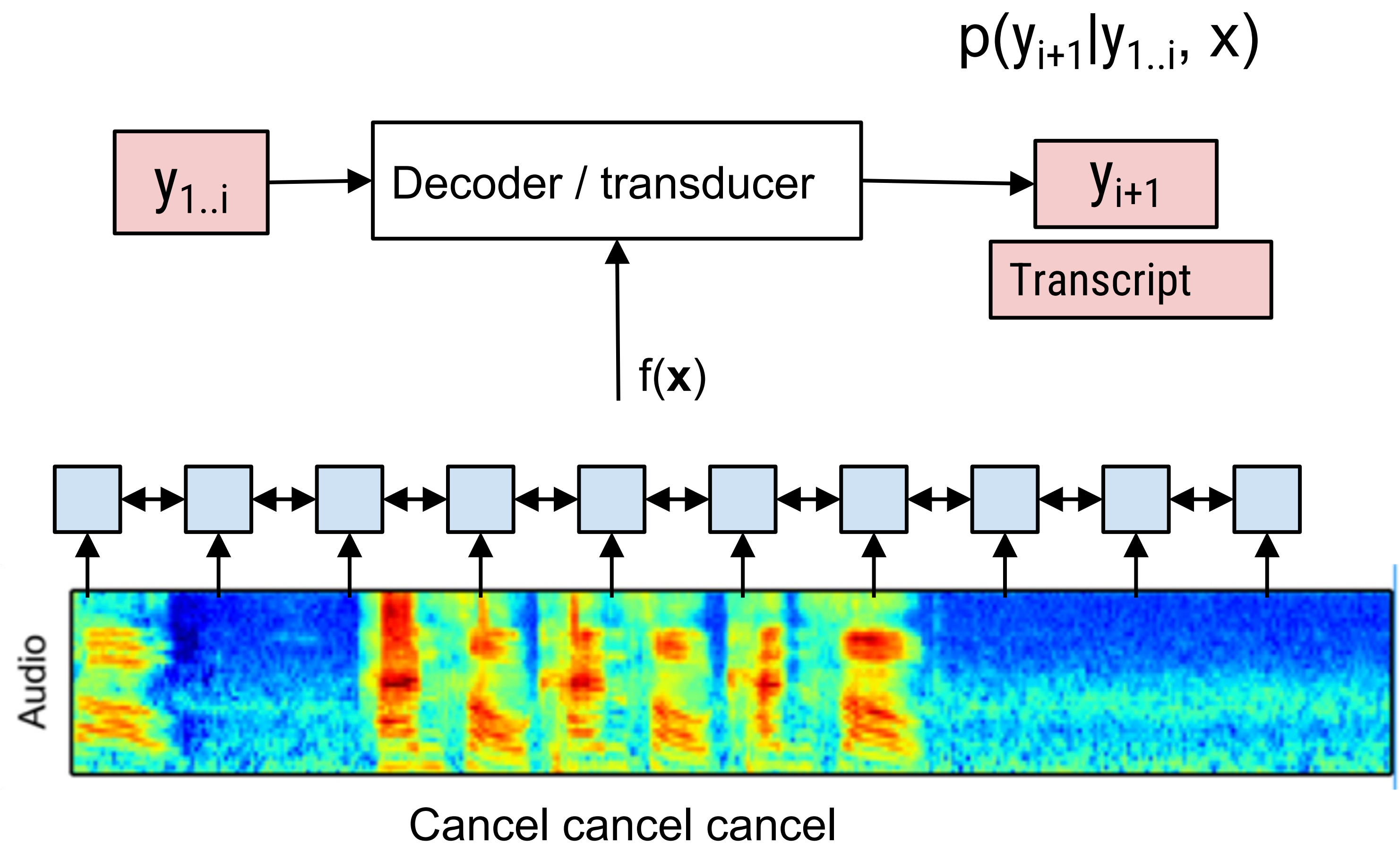
Sentence to Constituency Parse Tree

1. Read a sentence
2. Flatten the tree into a sequence (adding (,))
3. “Translate” from sentence to parse tree



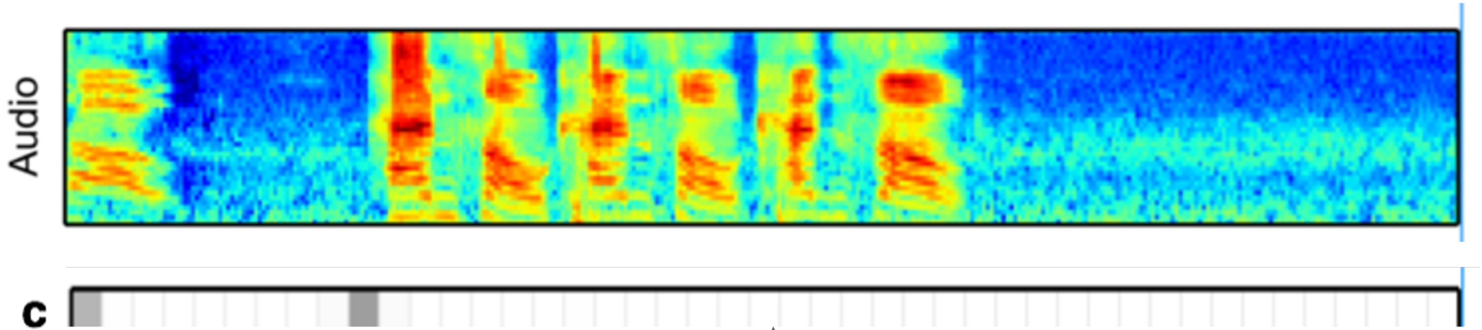
Vinyals, O., et al. “Grammar as a foreign language.” *NIPS (2015).*

Speech Recognition



Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell.*" *ICASSP (2015).*

Attention Example



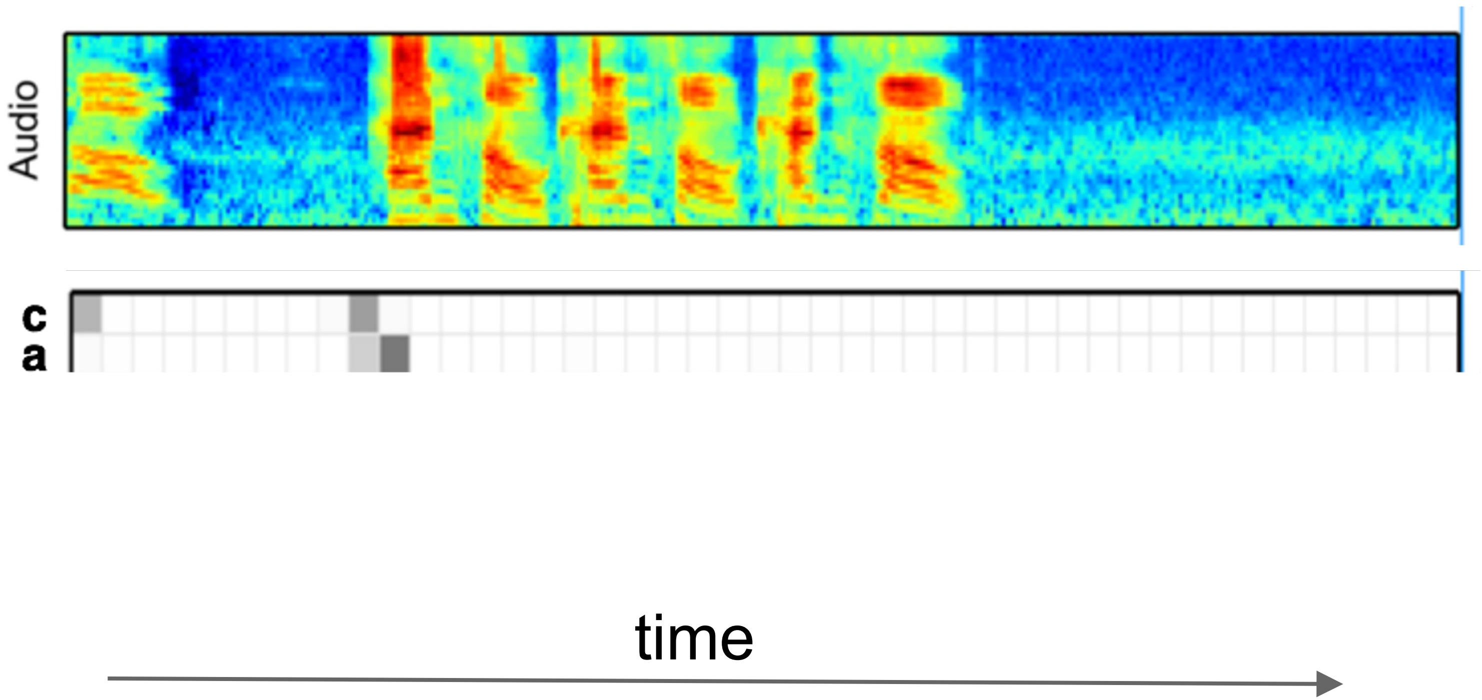
prediction derived from “attending” to segment of input

Attention vector - where the model thinks the relevant information is to be found

time

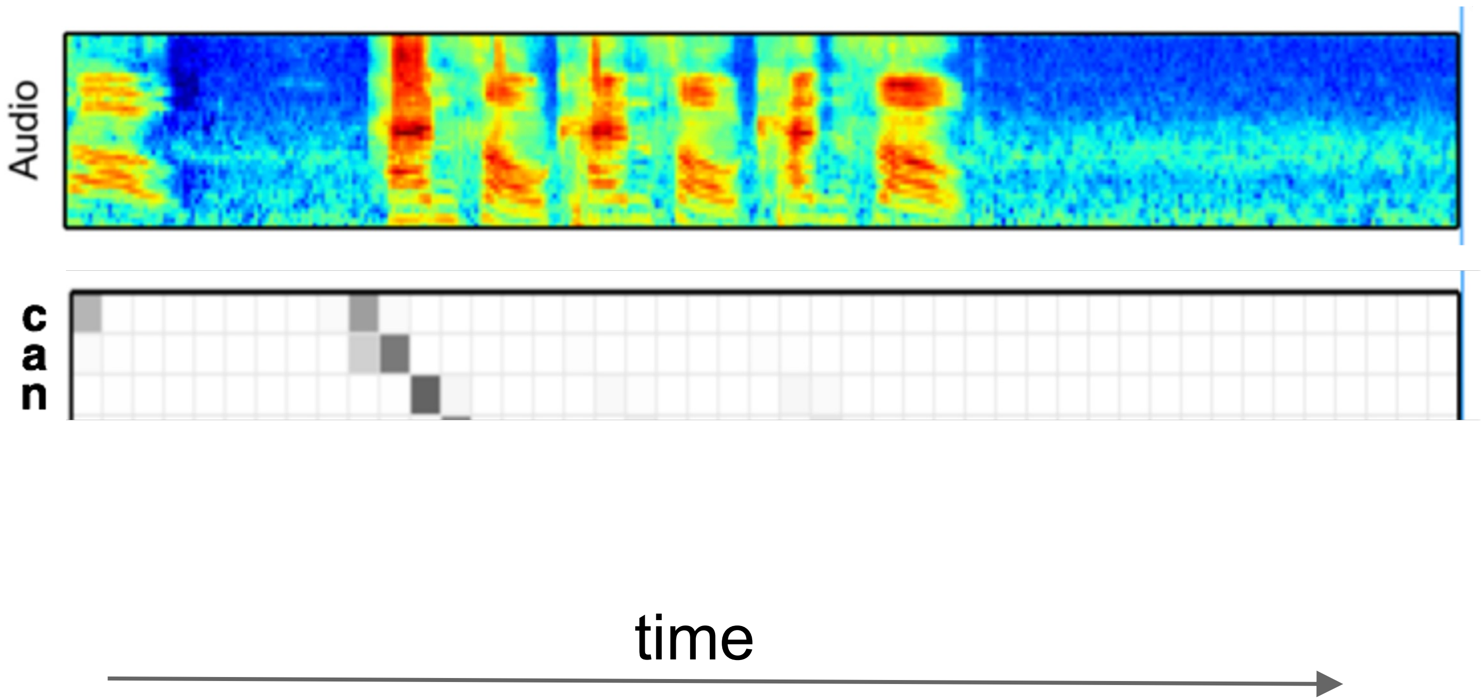
Chan, W., Jaitly, N., Le, Q., Vinyals, O. “*Listen Attend and Spell*.” *ICASSP (2015).*

Attention Example



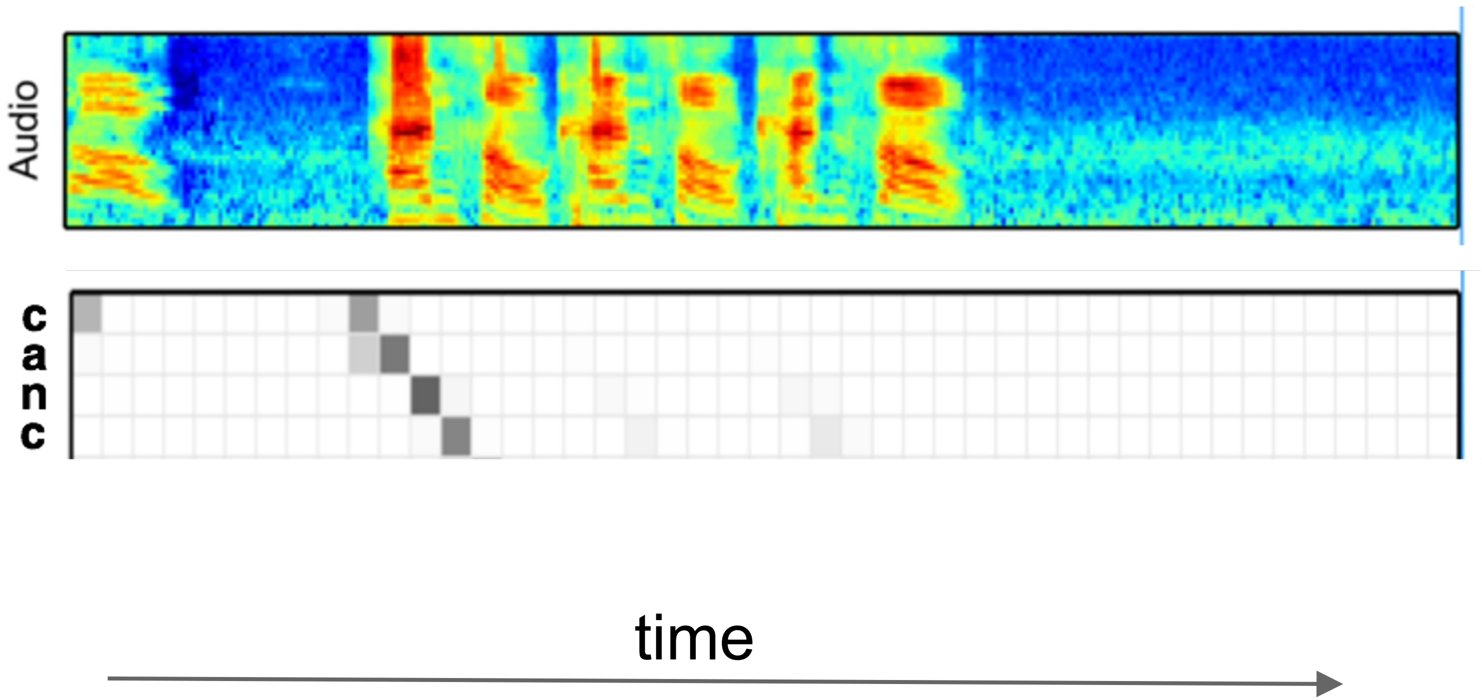
Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell.*" *ICASSP (2015).*

Attention Example



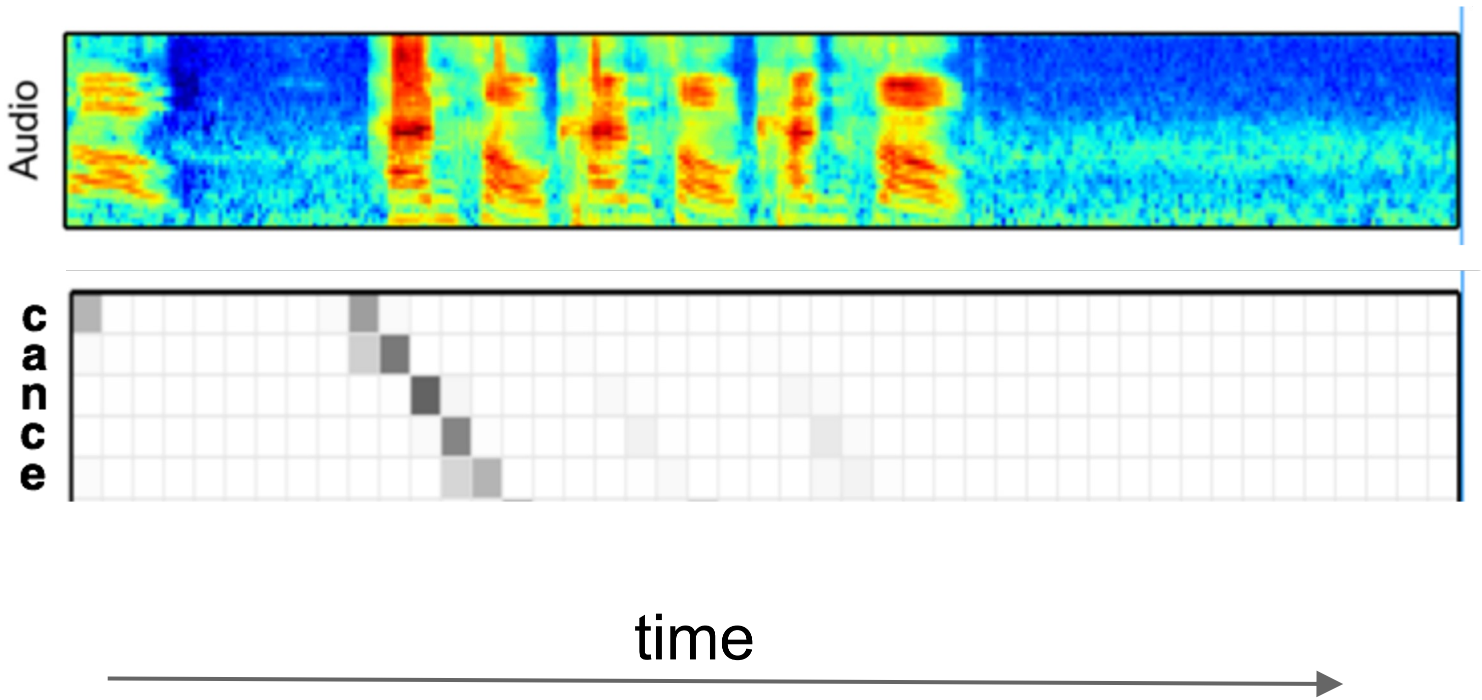
Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell*." *ICASSP (2015).*

Attention Example



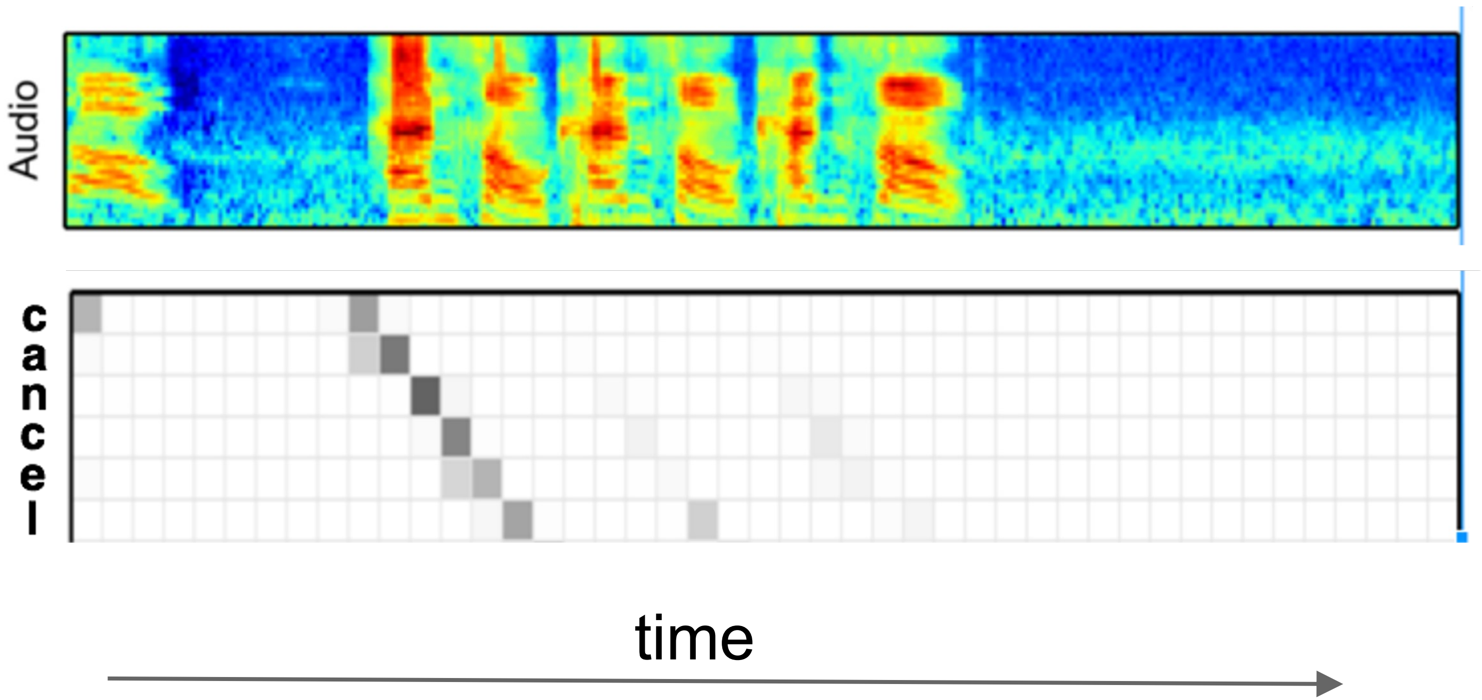
Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell.*" *ICASSP (2015).*

Attention Example



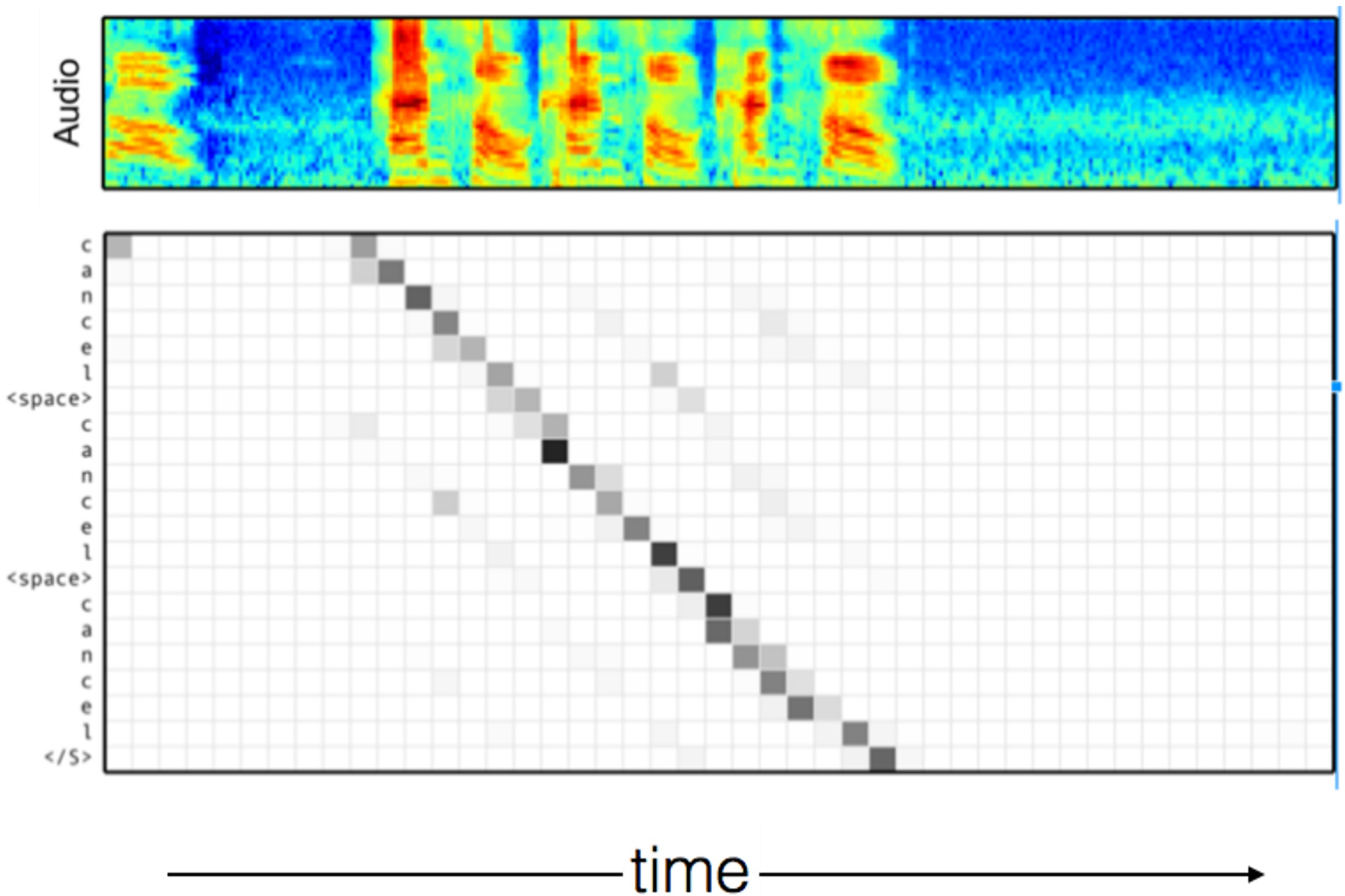
Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell.*" *ICASSP (2015).*

Attention Example



Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell.*" *ICASSP (2015).*

Attention Example



Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell*." *ICASSP (2015).*

Caption Generation with Visual Attention

A man riding a horse in a field.

Xu et al, ICML 2015

Caption Generation with Visual Attention

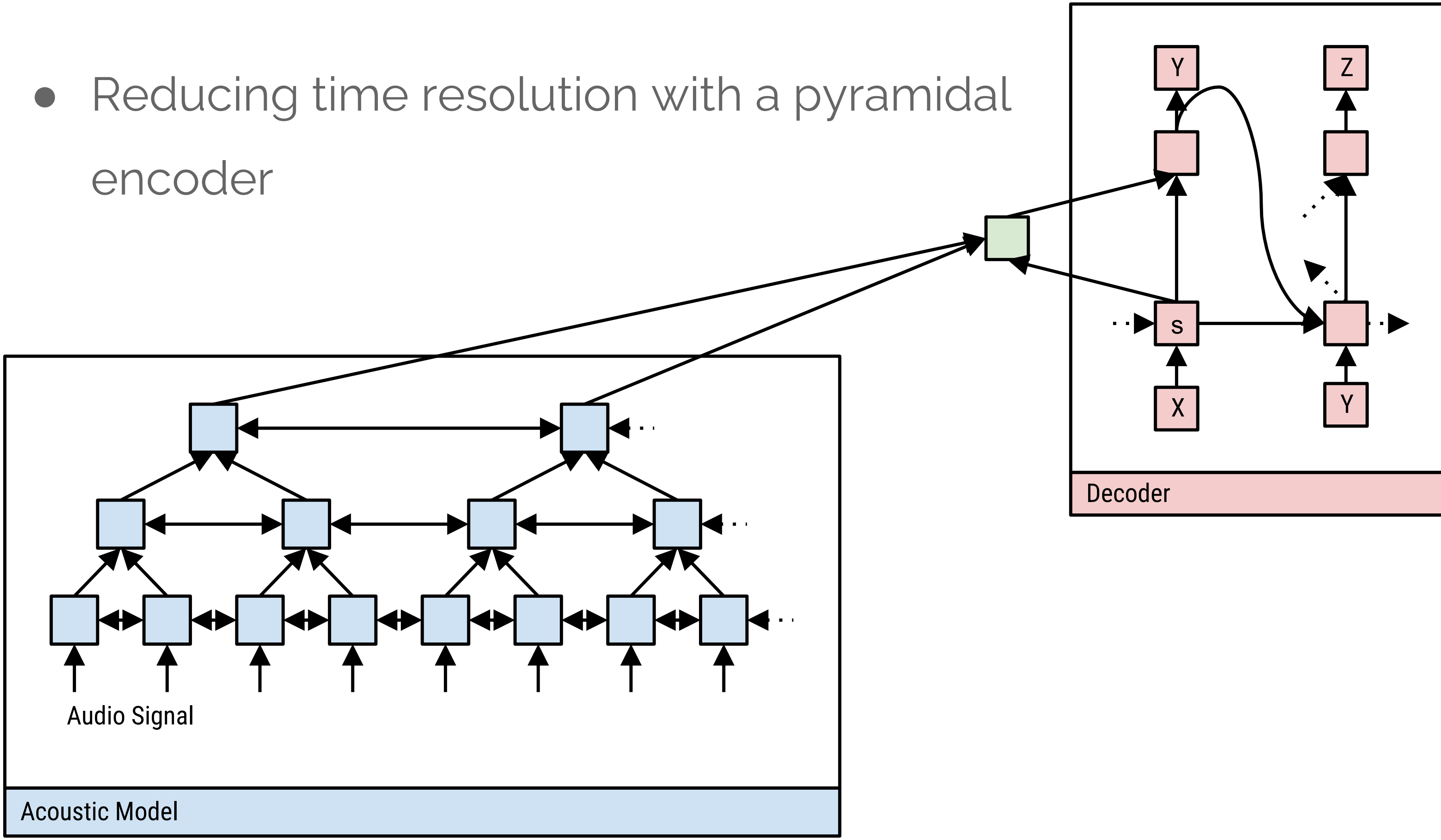
A woman holding a clock in her hand.

A large white bird standing in a forest.

Xu et al, ICML 2015

Listen Attend and Spell (LAS)

- Reducing time resolution with a pyramidal encoder



Chan, W., Jaitly, N., Le, Q., Vinyals, O. "*Listen Attend and Spell.*" *ICASSP (2015).*

LAS Results

Beam	Text	LogProb	WER
Truth	call aaa roadside assistance	-	-
1	call aaa roadside assistance	-0.5740	0.00
2	call triple a roadside assistance	-1.5399	50.0
3	call trip way roadside assistance	-3.5012	50.0
4	call xxx roadside assistance	-4.4375	25.0

Lip Reading

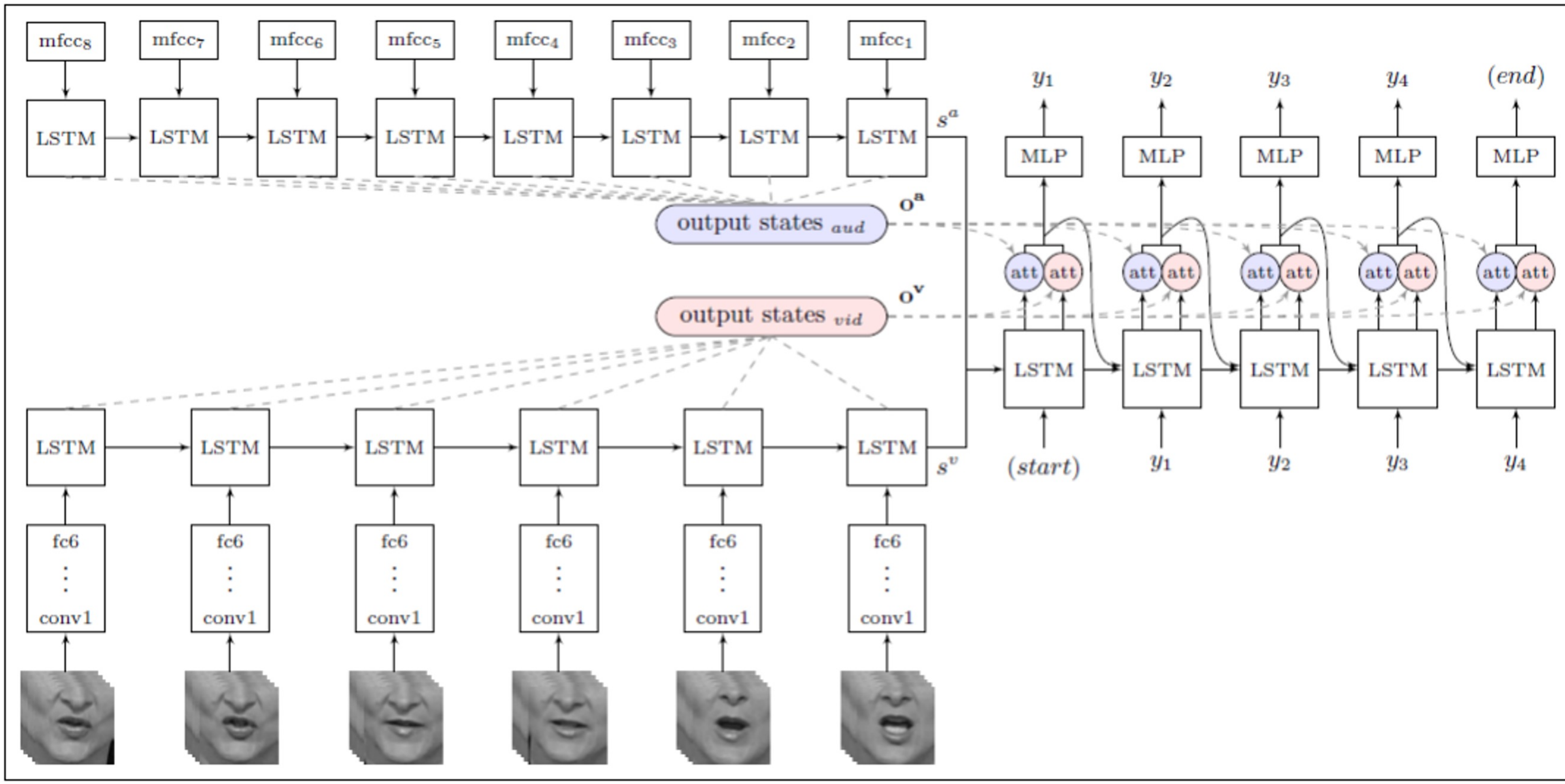
Channel	Series name	# hours	# sent.
BBC 1 HD	News†	1,584	50,493
BBC 1 HD	Breakfast	1,997	29,862
BBC 1 HD	Newsnight	590	17,004
BBC 2 HD	World News	194	3,504
BBC 2 HD	Question Time	323	11,695
BBC 4 HD	World Today	272	5,558
All		4,960	118,116

http://www.robots.ox.ac.uk/~vgg/data/lip_reading/

1. Chung, J., et al. "Lip reading sentences in the wild." *CVPR* (2017).
2. Assael, Y., et al. "Lipnet: Sentence-level lipreading." *arxiv* (2016).

Lip Reading

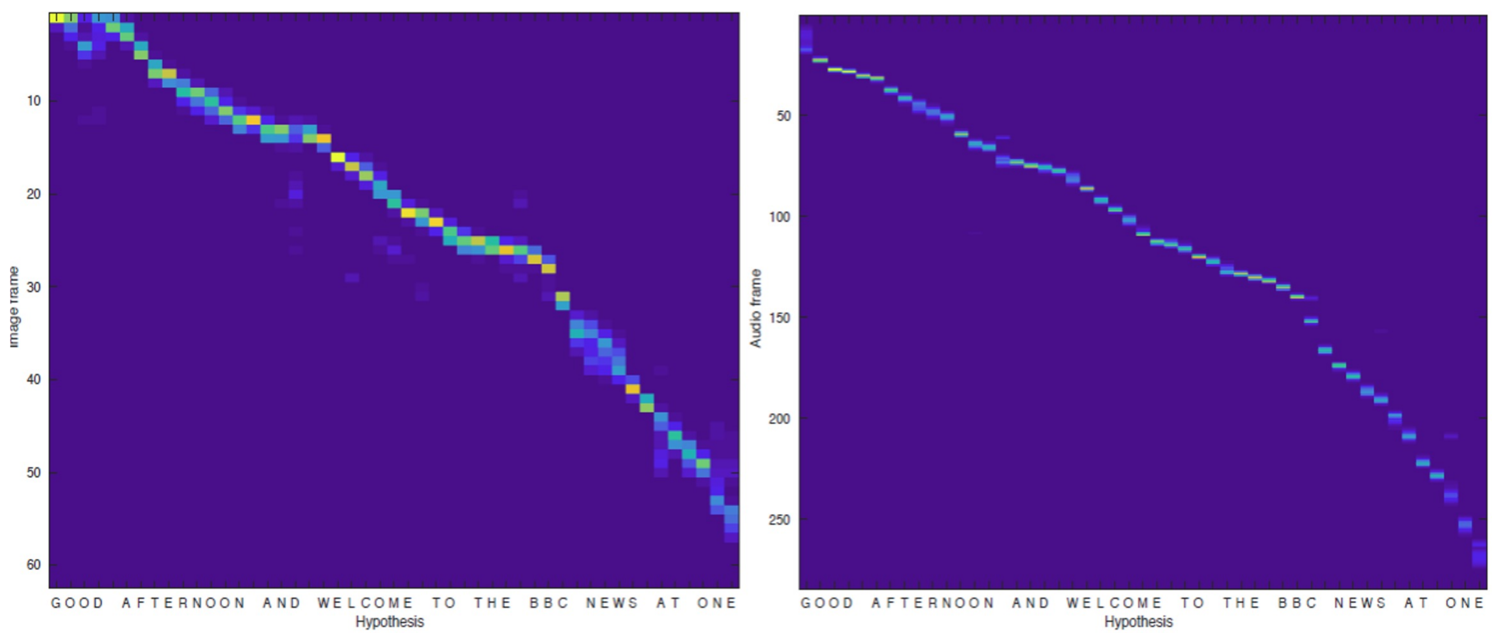
Separate embedding and attention for audio and visual streams



Chung, J., et al. "Lip reading sentences in the wild." *CVPR* (2017).

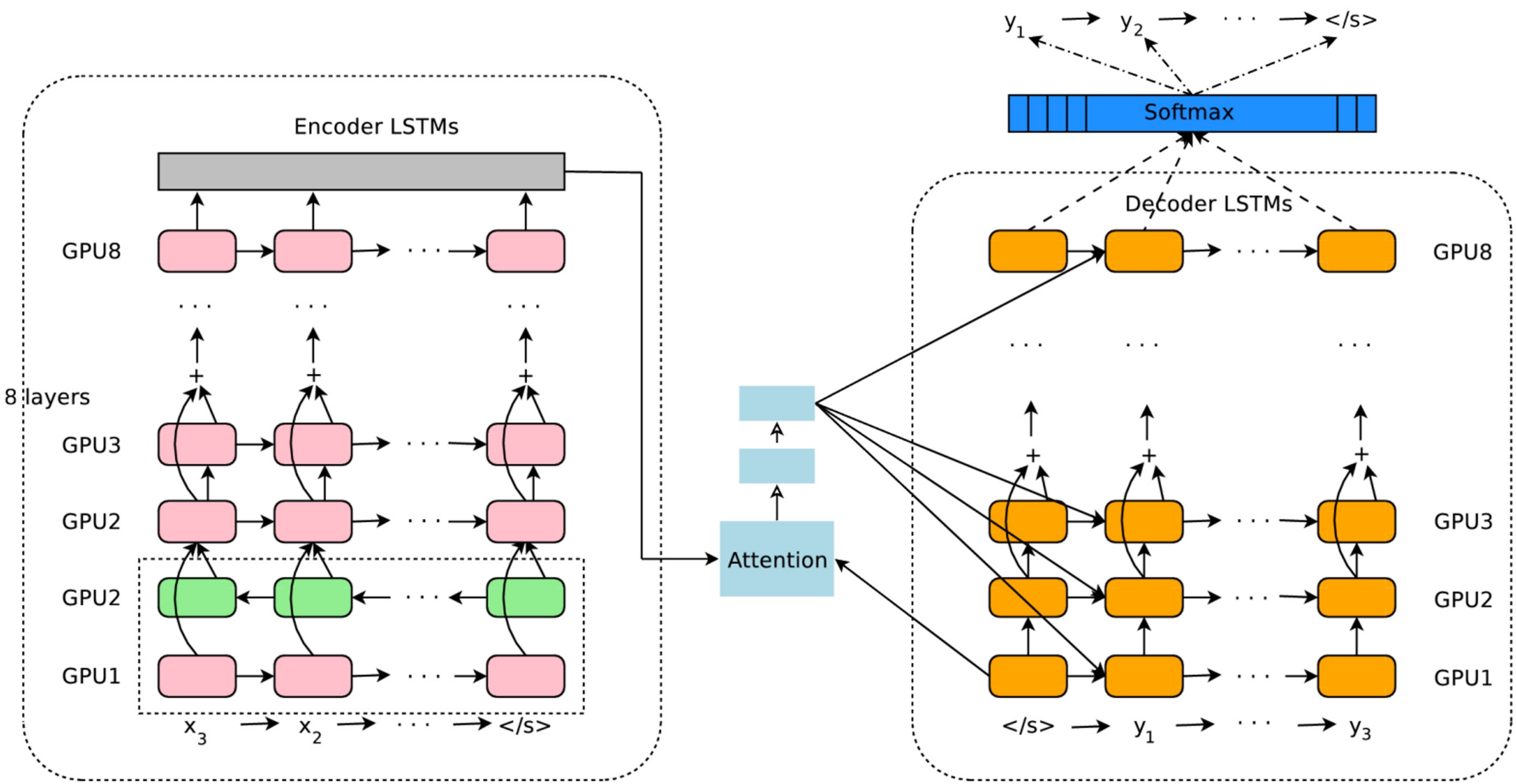
Lip Reading

Method	SNR	CER	WER	BLEU†
Lips only				
Professional‡	-	58.7%	73.8%	23.8
WAS	-	59.9%	76.5%	35.6
WAS+CL	-	47.1%	61.1%	46.9
WAS+CL+SS	-	42.4%	58.1%	50.0
WAS+CL+SS+BS	-	39.5%	50.2%	54.9



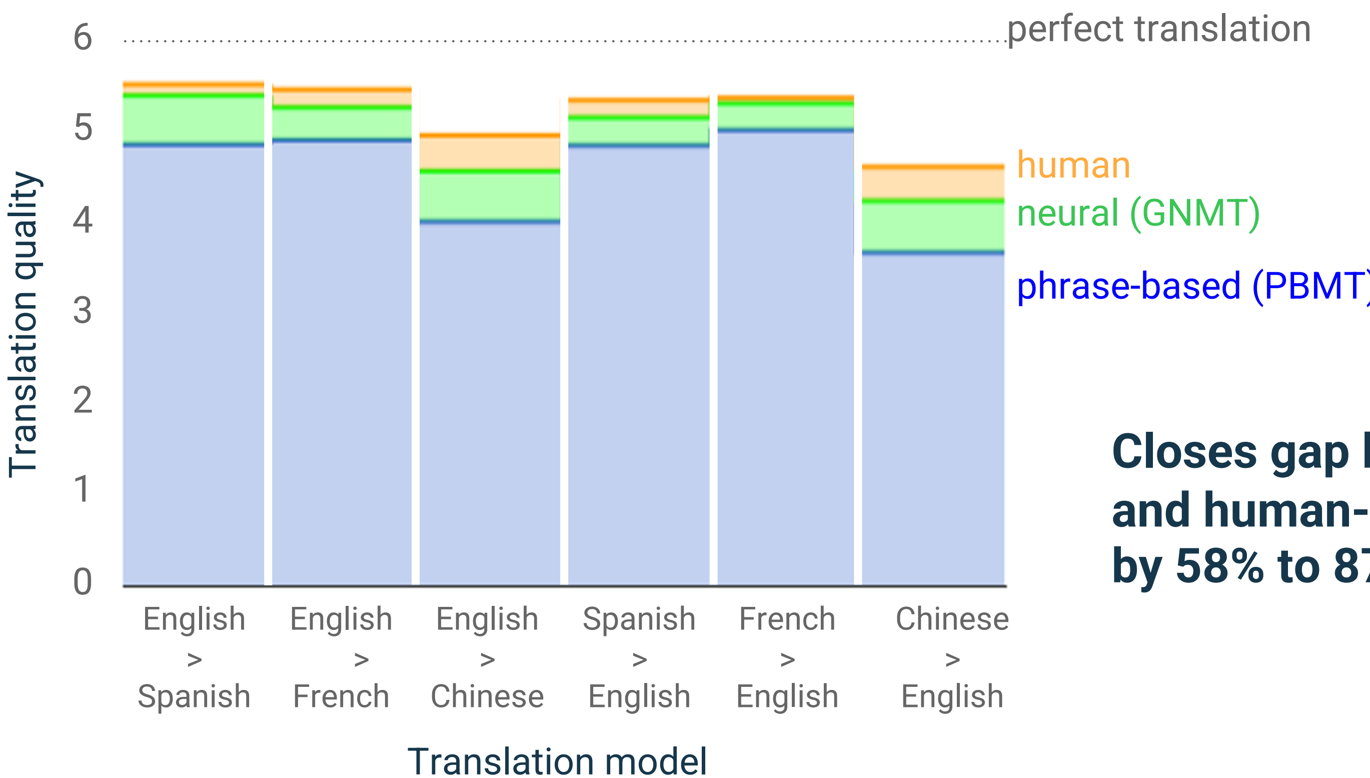
Chung, J., et al. "Lip reading sentences in the wild." *CVPR* (2017).

Google Neural Machine Translation System



Wu, Y., et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." *arxiv (2016).*

Google Neural Machine Translation System



Closes gap between old system and human-quality translation by 58% to 87%

research.googleblog.com/2016/09/a-neural-network-for-machine.html

Loss Functions

Loss Functions

- Cross Entropy
- Scheduled Sampling [1]
- Expected Loss [2]
- Augmented Loss [3]
- Sequence to Sequence as a beam search optimization [4]
- Learning decoders with different loss function [5]

1. Bengio, S., et al. “Scheduled sampling for sequence prediction with recurrent neural networks.” *NIPS (2015).*
2. Ranzato, M., et al. “Sequence level training with recurrent neural networks.” *ICLR (2016).*
3. Norouzi, M., et al. “Reward augmented maximum likelihood for neural structured prediction.” *NIPS (2016).*
4. Wiseman, S., Rush, A. “Sequence-to-sequence learning as beam-search optimization.” *EMLP (2016).*
5. Gu, J, Cho, K and Li, V.O.K. "Trainable greedy decoding for neural machine translation." *arXiv preprint arXiv:1702.02429* (2017).

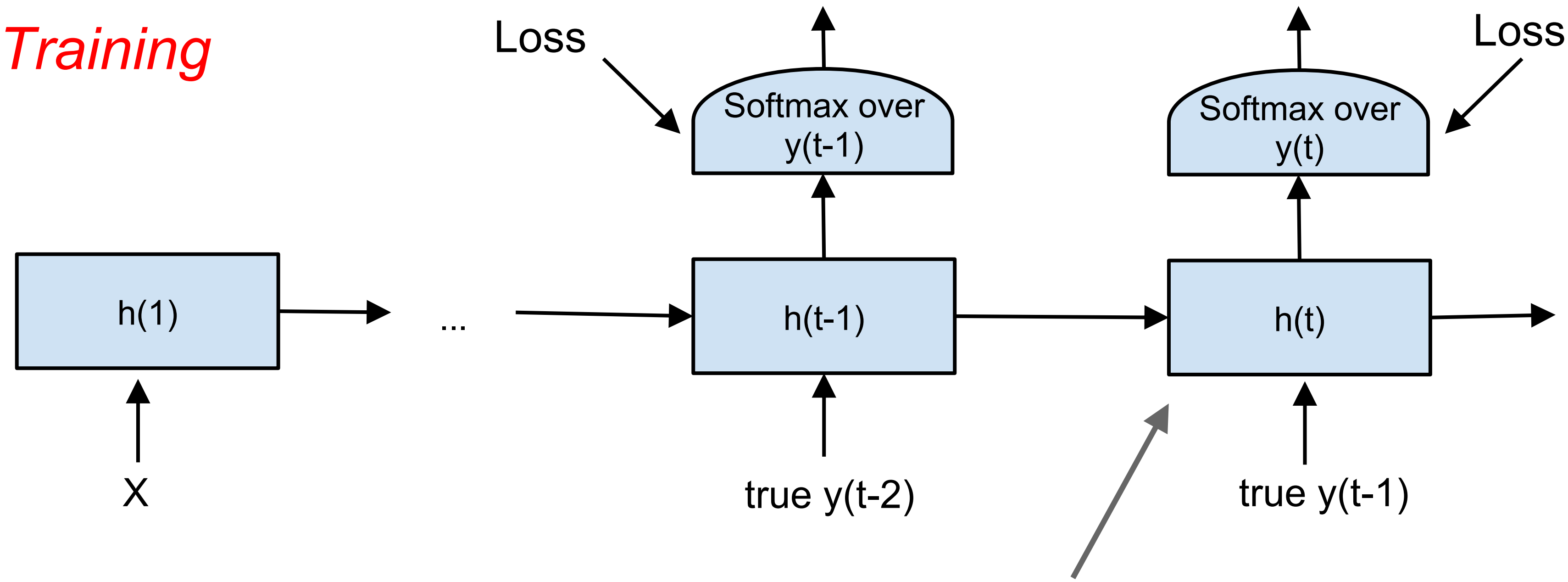
Cross Entropy (Negative Log Likelihood) Loss

- Log Likelihood, by chain rule is sum of next step log likelihoods

$$\log p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{N} \log p(y_i|y_{<i}, \mathbf{x})$$

- Supervised classification for each time step
  - depends on input, past outputs, which are known during training

Training and Inference Mismatch

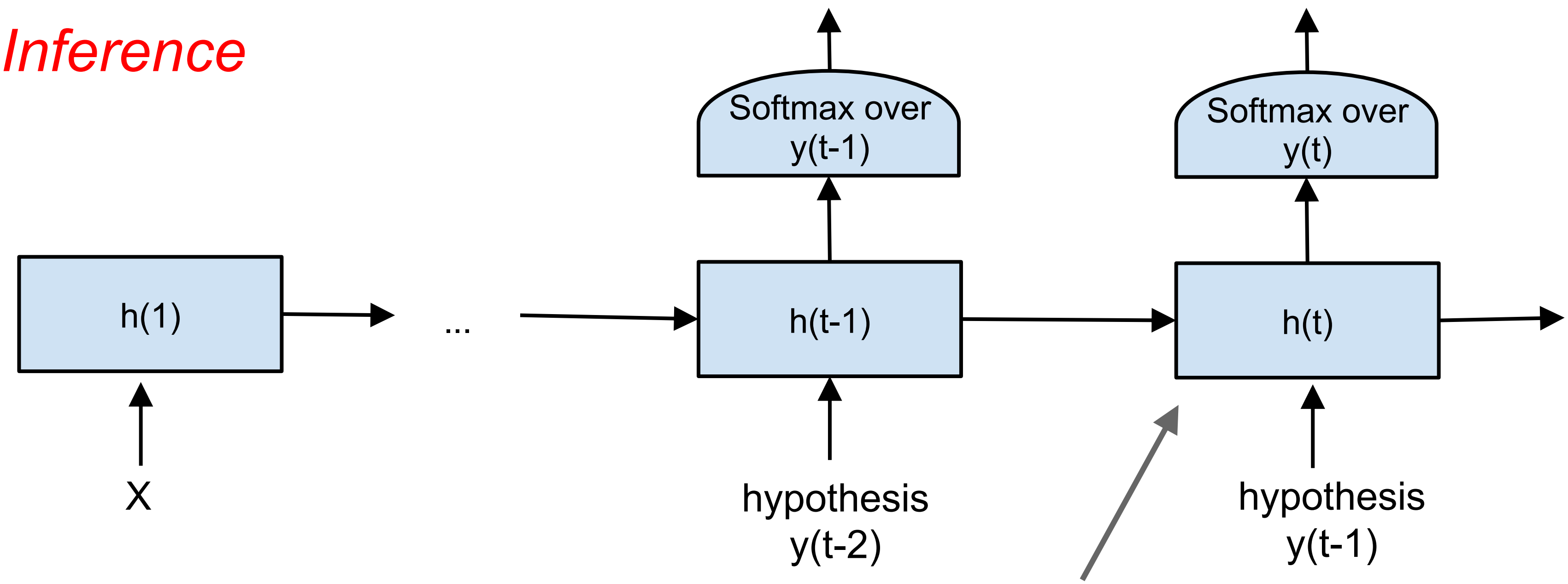


$$P(y_t|h_t) \text{ with } h_t = f(h_{t-1}, y_{t-1}; \theta)$$

Bengio, S., et al. "Scheduled sampling for sequence prediction with recurrent neural networks." *NIPS (2015).*

Training and Inference Mismatch

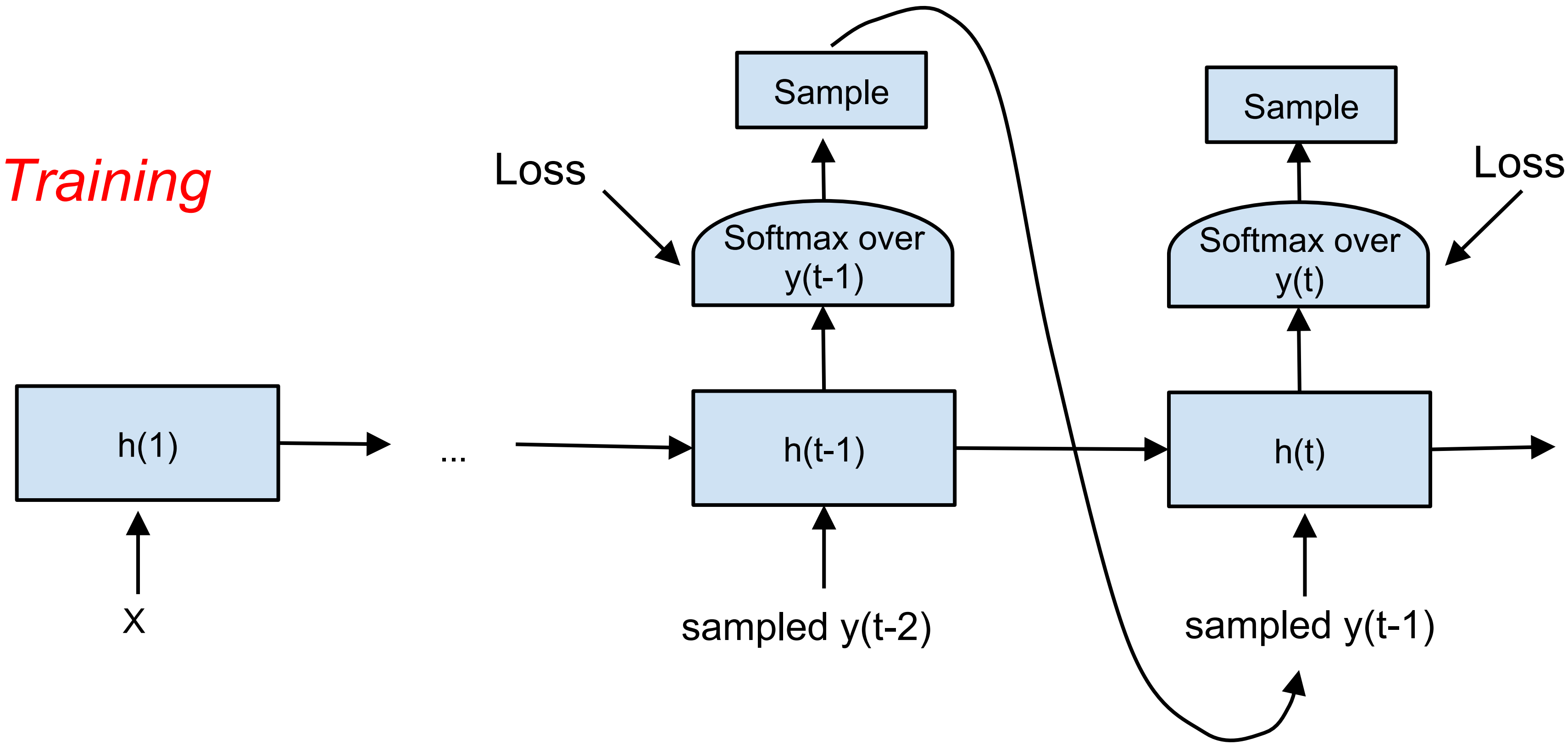
Inference



$$P(y_t|h_t) \text{ with } h_t = f(h_{t-1}, y_{t-1}; \theta)$$

Bengio, S., et al. "Scheduled sampling for sequence prediction with recurrent neural networks." *NIPS (2015).*

Scheduled Sampling



$$P(y_t|h_t) \text{ with } h_t = f(h_{t-1}, \hat{y}_{t-1}; \theta)$$

Bengio, S., et al. "Scheduled sampling for sequence prediction with recurrent neural networks." *NIPS (2015).*

Scheduled Sampling

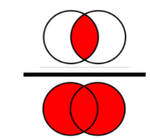
Machine Translation Model	Bleu-4	Meteor	Cider
Baseline	28.8	24.2	89.5
Baseline with dropout	28.1	23.9	87.0
Scheduled sampling	**30.6**	**24.3**	**92.1**

Parsing Model	F1
Baseline LSTM with dropout	87.00
Scheduled sampling with dropout	**88.68**

Speech Recognition Model	WER
LAS + LM Rescoring	12.6
LAS + **Sampling** + LM Rescoring	10.3

Bengio, S., et al. "Scheduled sampling for sequence prediction with recurrent neural networks." *NIPS (2015).*

Rewards (-loss) used in Structured Prediction

TASK	REWARD	
Classification	0/1 rewards	$r(\mathbf{y}, \mathbf{y}^*) = \mathbb{1}[\mathbf{y} = \mathbf{y}^*]$
Segmentation	Intersection over Union	$r(\mathbf{y}, \mathbf{y}^*) = \cap(\mathbf{y}, \mathbf{y}^*) / \cup(\mathbf{y}, \mathbf{y}^*)$ 
Speech Recognition	Edit Distance	$r(\mathbf{y}, \mathbf{y}^*)$ = (#*d* + #*i* + #*s*) I N T E * N T I O N | | | | | | | | | | * E X E C U T I O N d s s i s
Machine Translation	BLEU	

Expected reward (-loss)

Given a dataset of input output pairs, $\mathcal{D} \equiv \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)*})\}_{i=1}^{N}$

learn a conditional distribution $p_\theta(\mathbf{y} \mid \mathbf{x})$ that minimizes

expected loss:

$$\mathcal{L}_{\mathrm{RL}}(\boldsymbol{\theta}) = \sum_{(\mathbf{x}, \mathbf{y}^*) \in \mathcal{D}} - \sum_{\mathbf{y} \in \mathcal{Y}} p_\theta(\mathbf{y} \mid x)\ r(\mathbf{y}, \mathbf{y}^*)$$

Sample from the ***model*** *distribution*

Difficult / Impossible to train from scratch!!

Mixed Incremental Cross-Entropy Reinforce (MIXER)

- Gradually interpolate from Cross-Entropy to Expected Loss

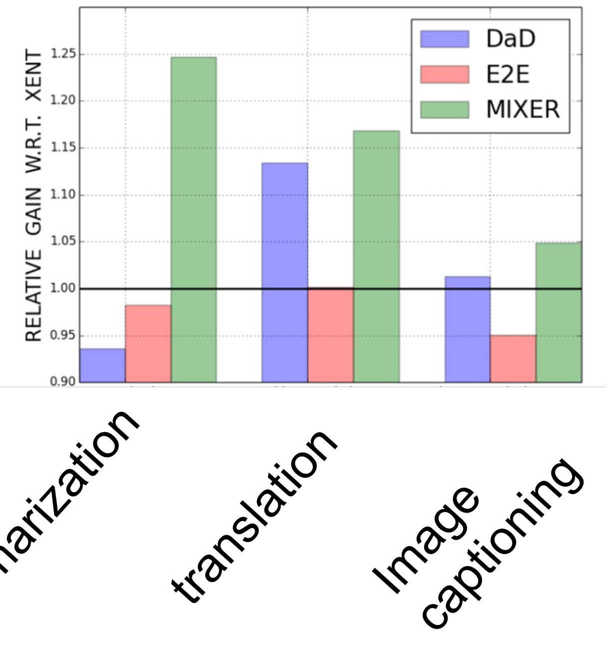
Data: a set of sequences with their corresponding context.
Result: RNN optimized for generation.
Initialize RNN at random and set N^{XENT}, $N^{\text{XE+R}}$ and Δ;
for $s = T, 1, -\Delta$ **do**
  if $s == T$ **then**
    train RNN for N^{XENT} epochs using XENT only;
  else
    train RNN for $N^{\text{XE+R}}$ epochs. Use XENT loss in the first s steps, and REINFORCE (sampling from the model) in the remaining $T - s$ steps;
  end
end

More expected loss optimization as training proceeds

Ranzato, M., et al. "Sequence level training with recurrent neural networks." *ICLR (2016).*

Mixed Incremental Cross-Entropy Reinforce (MIXER)

TASK	XENT	DAD	E2E	MIXER
summarization	13.01	12.18	12.78	**16.22**
translation	17.74	20.12	17.77	**20.73**
image captioning	27.8	28.16	26.42	**29.16**



Ranzato, M., et al. "Sequence level training with recurrent neural networks." *ICLR (2016).*

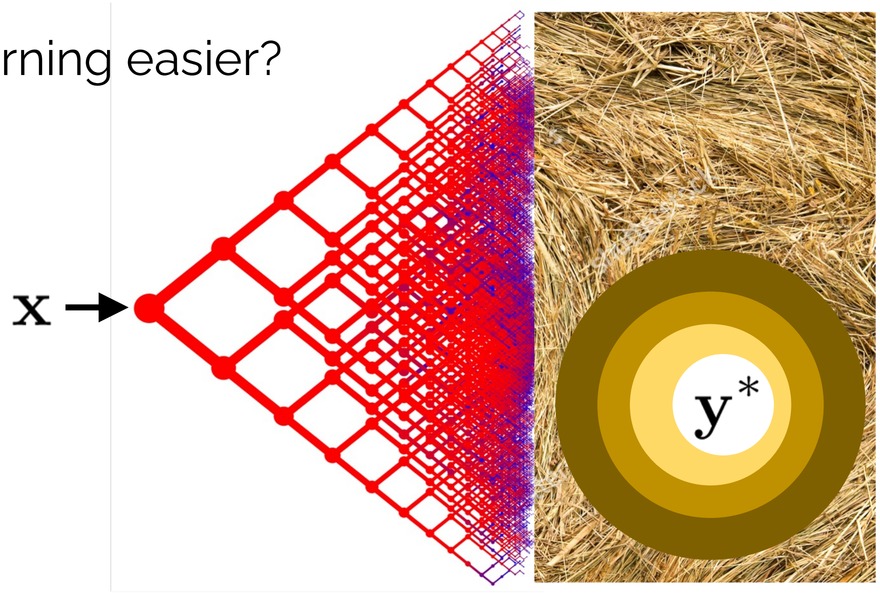
Reward Augmented Maximum Likelihood (RML)

Finding the *right output sequence*, for tasks like speech recognition or machine translation is like finding *a needle in a haystack*.

It is very risky to shoot *only* for the *true target*.

What if we expand the targets to make learning easier?

E.g. by inserting, deleting random words...



Norouzi, M., et al. "Reward augmented maximum likelihood for neural structured prediction." *NIPS (2016).*

Reward Augmented maximum likelihood (RML)

$$\mathcal{L}_{\mathrm{RML}}(\boldsymbol{\theta};\tau) = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} \left\{ -\sum_{\mathbf{y}\in\mathcal{Y}} q(\mathbf{y} \mid \mathbf{y}^*;\tau) \log p_\theta(\mathbf{y} \mid \mathbf{x}) \right\}$$

Optimal $p_\theta(\mathbf{y} \mid \mathbf{x})$:

*Sample from the **reward** distribution, irrespective of the model*

$$q(\mathbf{y} \mid \mathbf{y}^*;\tau) = \frac{1}{Z(\mathbf{y}^*,\tau)} \exp\left\{ r(\mathbf{y},\mathbf{y}^*)/\tau \right\}$$

$$\mathcal{L}_{\mathrm{RML}}(\boldsymbol{\theta};\tau) = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} D_{\mathrm{KL}}\left(q(\mathbf{y} \mid \mathbf{y}^*;\tau) \parallel p_\theta(\mathbf{y} \mid \mathbf{x})\right) + \text{constant}$$

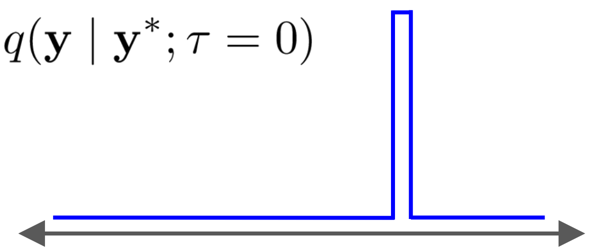
Norouzi, M., et al. "Reward augmented maximum likelihood for neural structured prediction." *NIPS (2016).*

RML - Impact of temperature τ

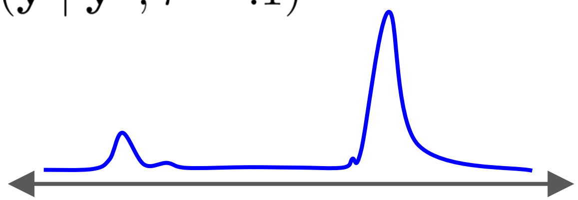
Temperature impacts spread of distribution we sample from

Cross Entropy Targets → More spread

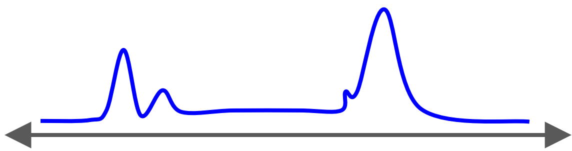
$q(\mathbf{y} \mid \mathbf{y}^*; \tau = 0)$



$q(\mathbf{y} \mid \mathbf{y}^*; \tau = .1)$



$q(\mathbf{y} \mid \mathbf{y}^*; \tau = .2)$

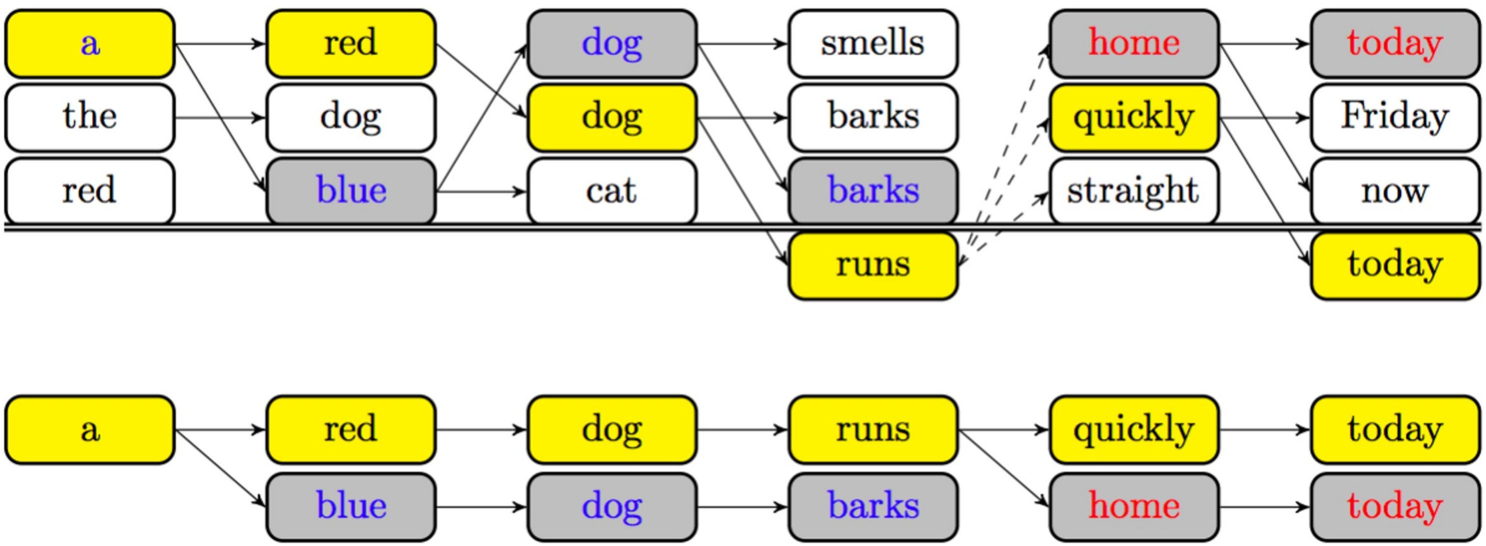


$$\mathcal{L}_{\mathrm{RML}}(\boldsymbol{\theta}; \tau) = \sum_{(\mathbf{x}, \mathbf{y}^*) \in \mathcal{D}} D_{\mathrm{KL}}\left(q(\mathbf{y} \mid \mathbf{y}^*; \tau) \parallel p_\theta(\mathbf{y} \mid \mathbf{x})\right) + \mathrm{constant}$$

Norouzi, M., et al. “Reward augmented maximum likelihood for neural structured prediction.” *NIPS (2016).*

Margin Loss

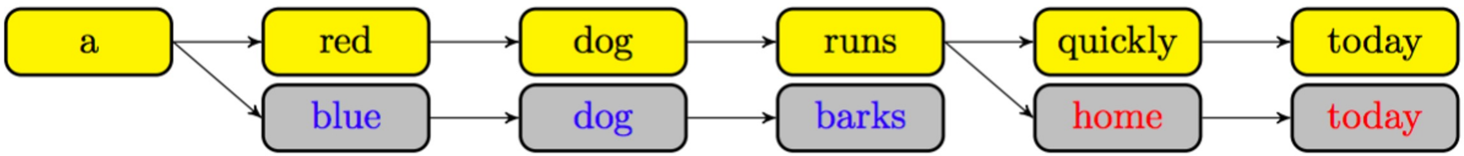
- Perform beam search until correct hypothesis falls out of the beam
- Restart beam whenever there is a violation
- Extract correct hypothesis and competing hypotheses



Wiseman, S., Rush, A. “Sequence-to-sequence learning as beam-search optimization.” *EMLP (2016).*

Margin Loss

- Add a margin score for all time steps where the correct hypothesis is not better than the Kth best hypothesis by a certain margin



$$\mathcal{L}(f) = \sum_{t=1}^{T} \Delta(\hat{y}_{1:t}^{(K)}) \left[1 - f(y_t, \boldsymbol{h}_{t-1}) + f(\hat{y}_t^{(K)}, \hat{\boldsymbol{h}}_{t-1}^{(K)})\right]$$

Loss for error; 0 when margin constraint is satisfied

Score function for prediction of current output

Score function for prediction of K^{th} best output

Wiseman, S., Rush, A. "Sequence-to-sequence learning as beam-search optimization." *EMLP (2016).*

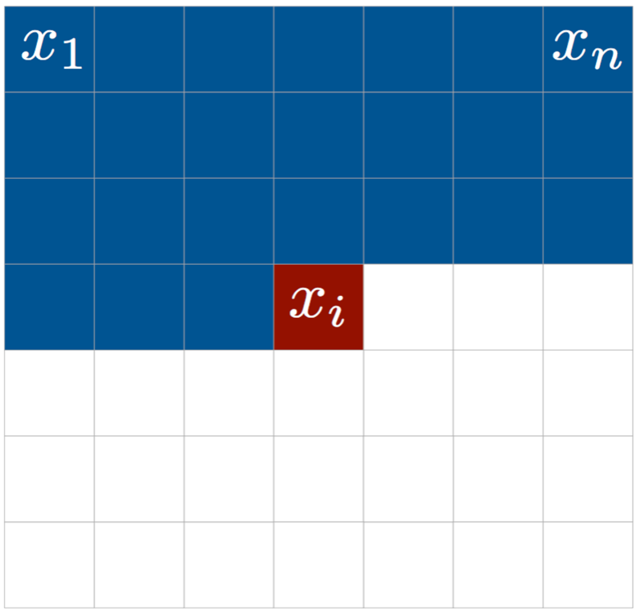
Margin Loss

	Machine Translation (BLEU)		
	$K_{te} = 1$	$K_{te} = 5$	$K_{te} = 10$
seq2seq	22.53	24.03	23.87
BSO, SB-Δ	**23.83**	**26.36**	**25.48**
XENT	17.74	20.10	20.28
DAD	20.12	22.25	22.40
MIXER	20.73	21.81	21.83

Wiseman, S., Rush, A. "Sequence-to-sequence learning as beam-search optimization." *EMLP (2016).*

Autoregressive Generative Models

Pixel RNN Model

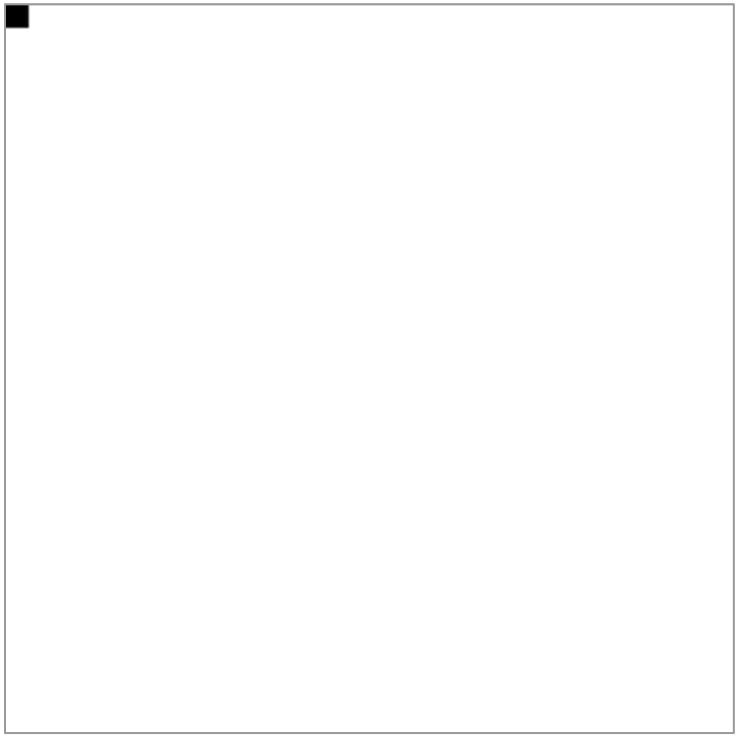


$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

- Fully visible
- Similar to language models with RNNs
- Model pixels with Softmax

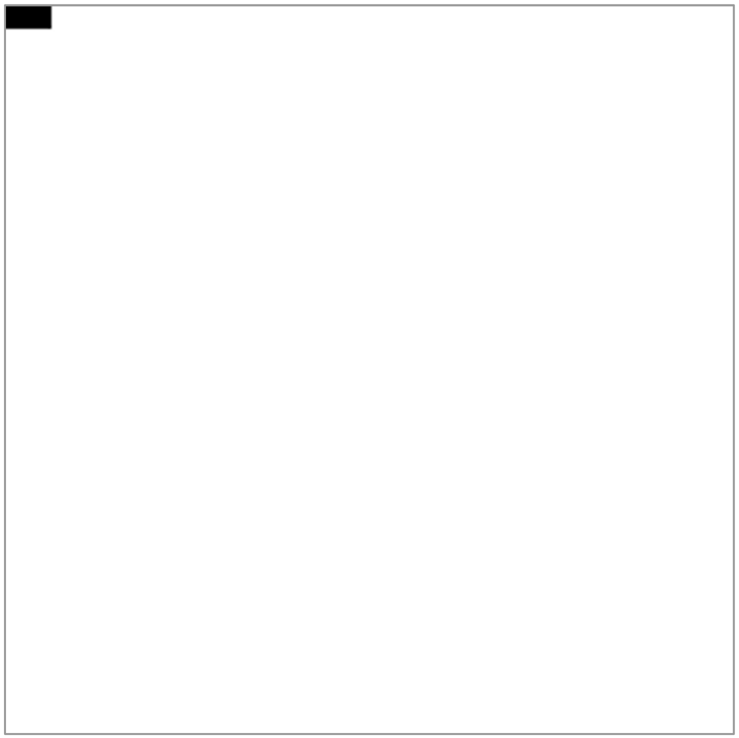
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



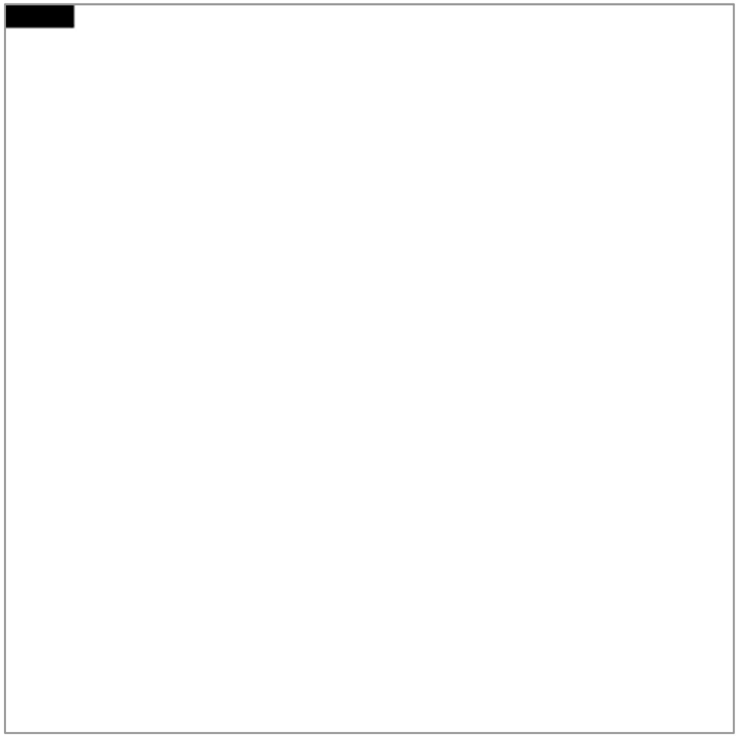
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



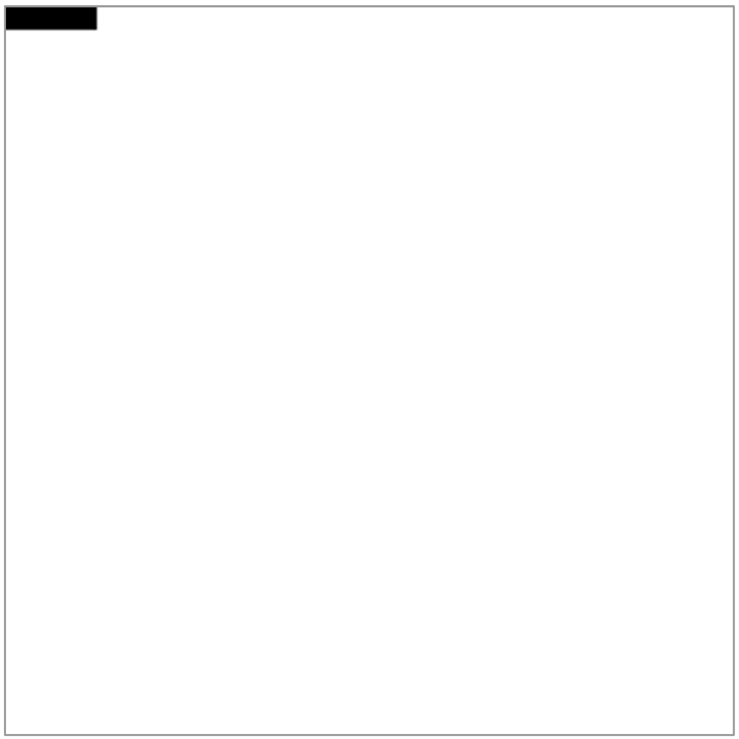
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



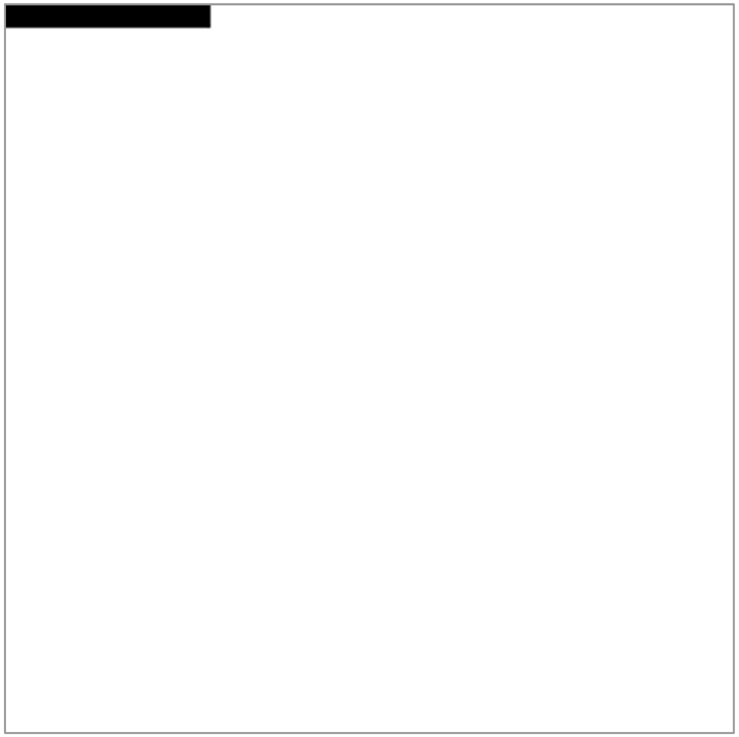
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



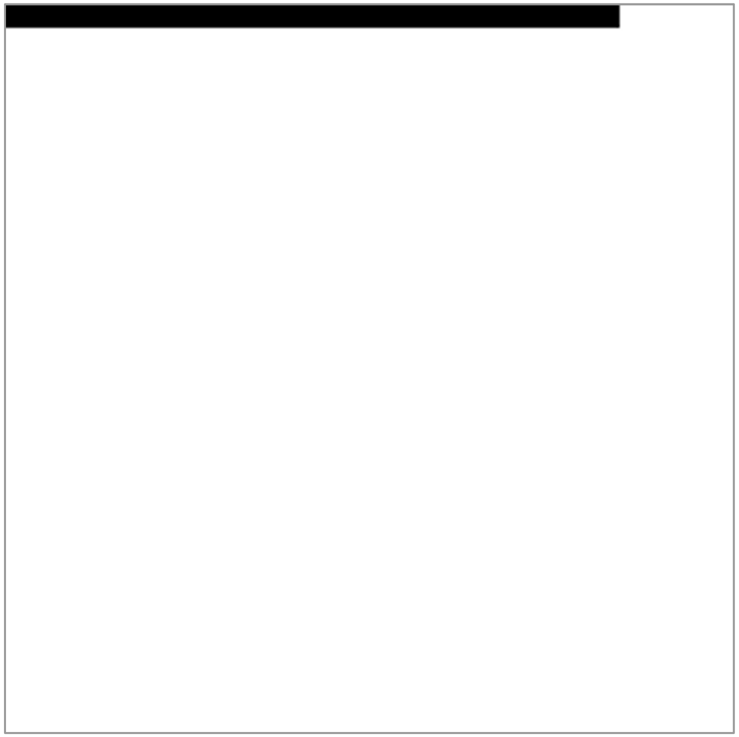
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



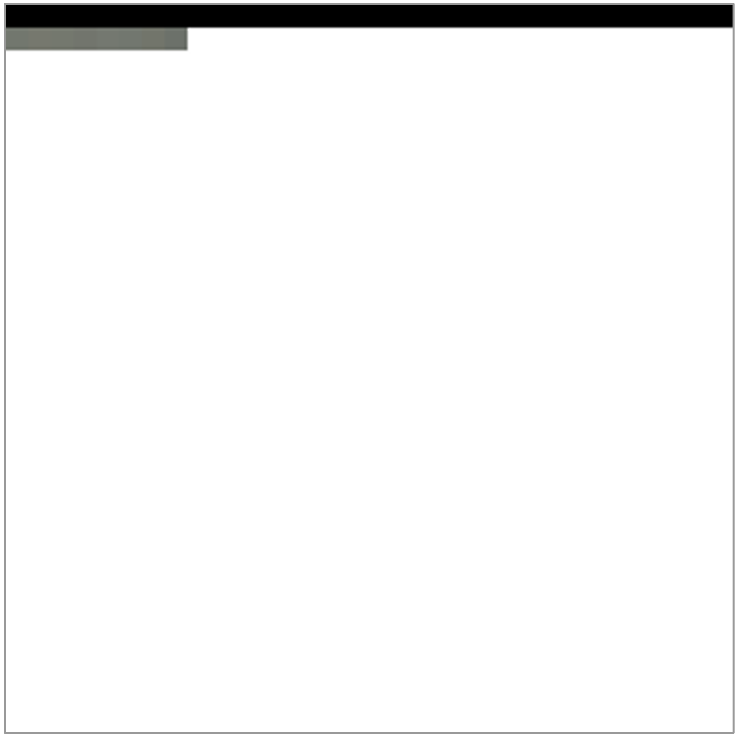
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



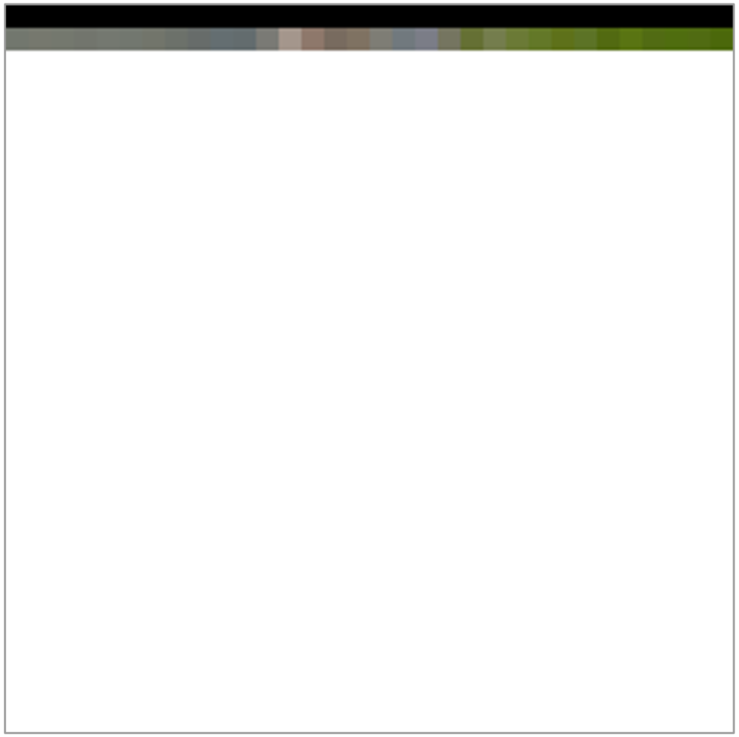
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



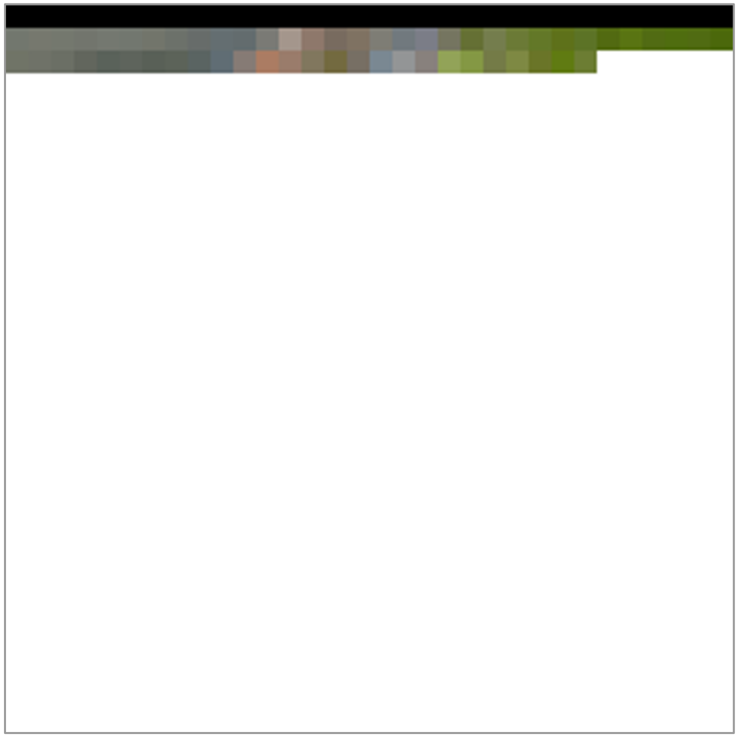
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



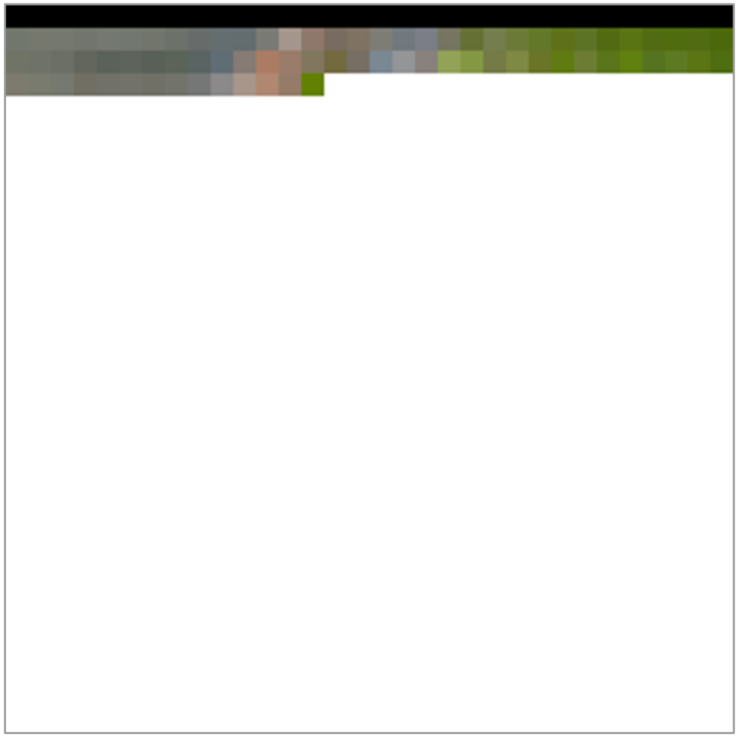
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling

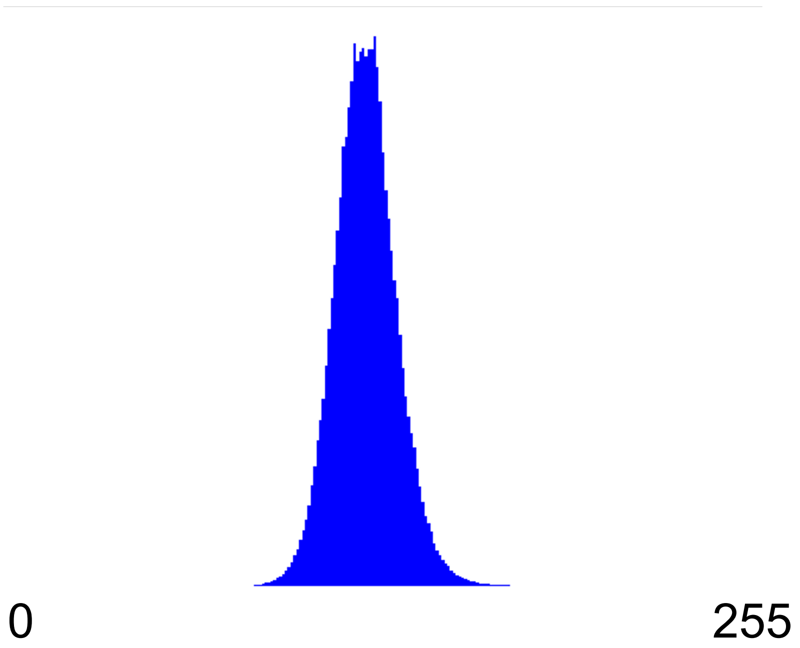


van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling

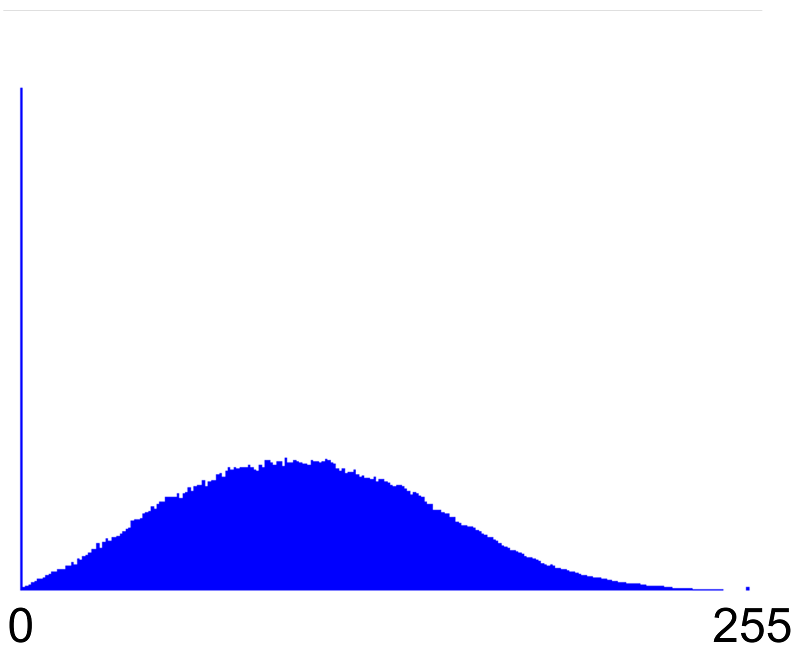
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



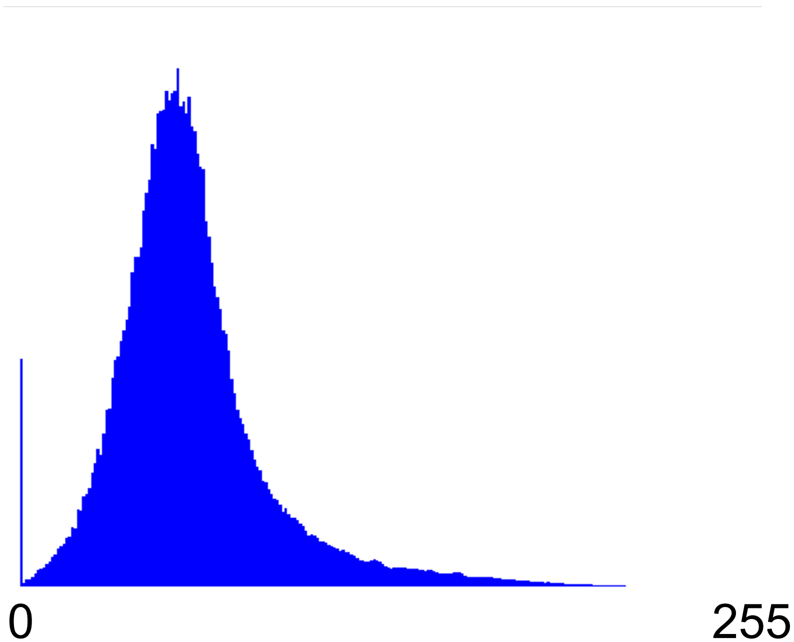
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



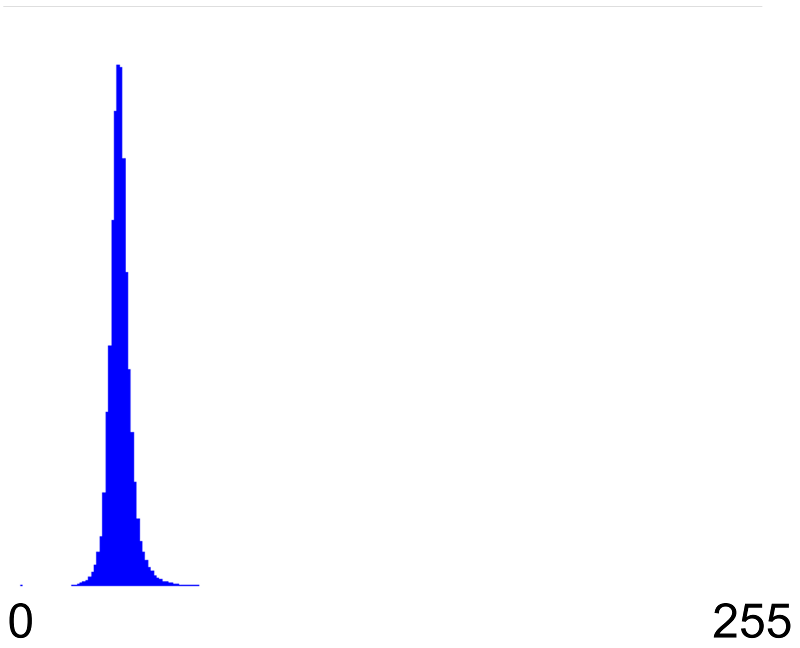
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Softmax Sampling



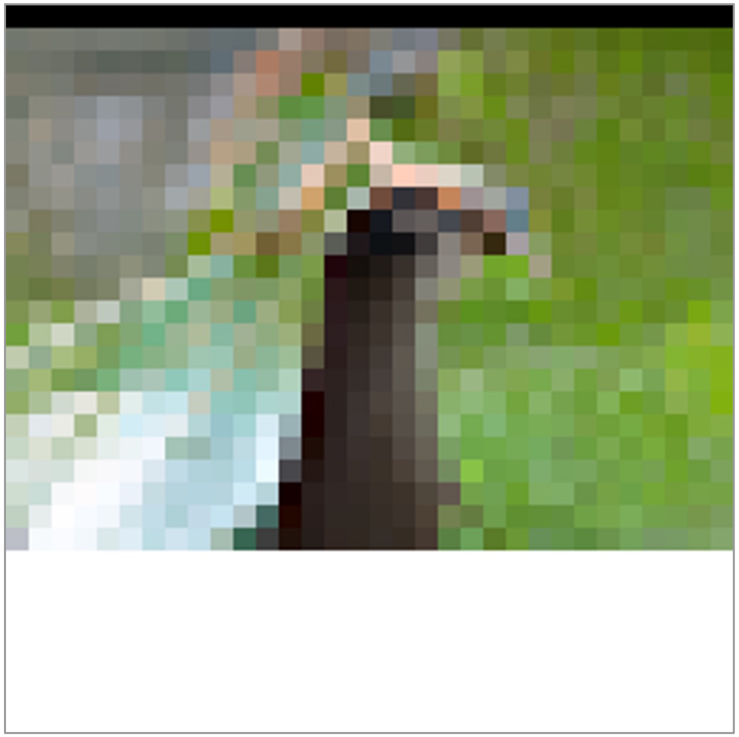
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

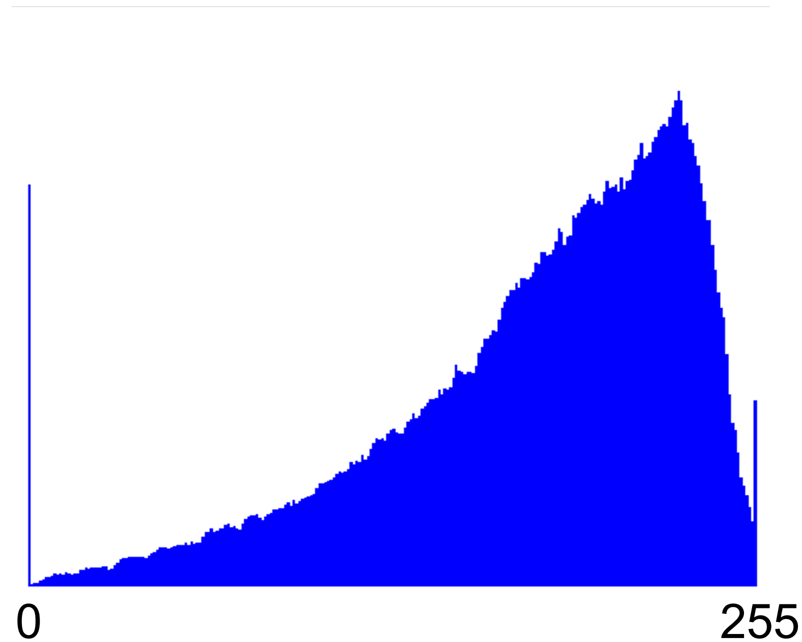
Softmax Sampling



van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

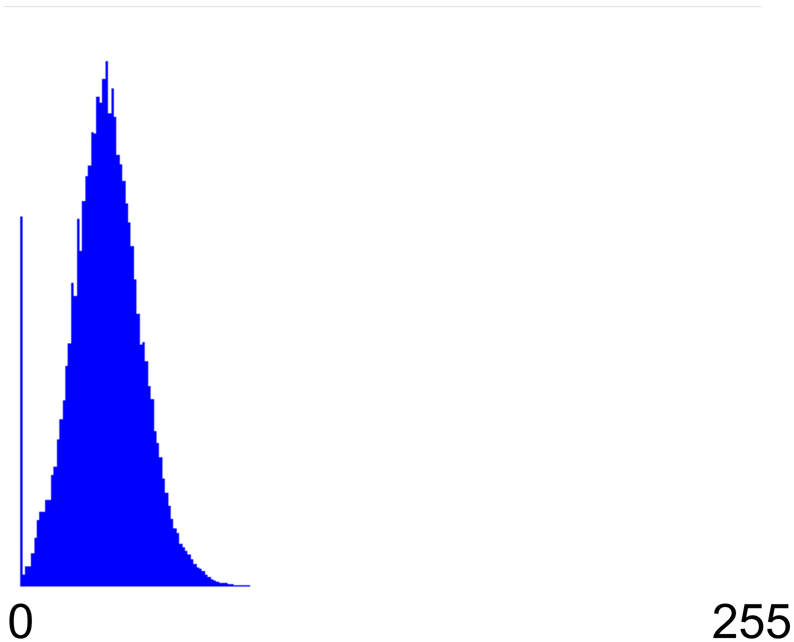
Softmax Sampling





van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

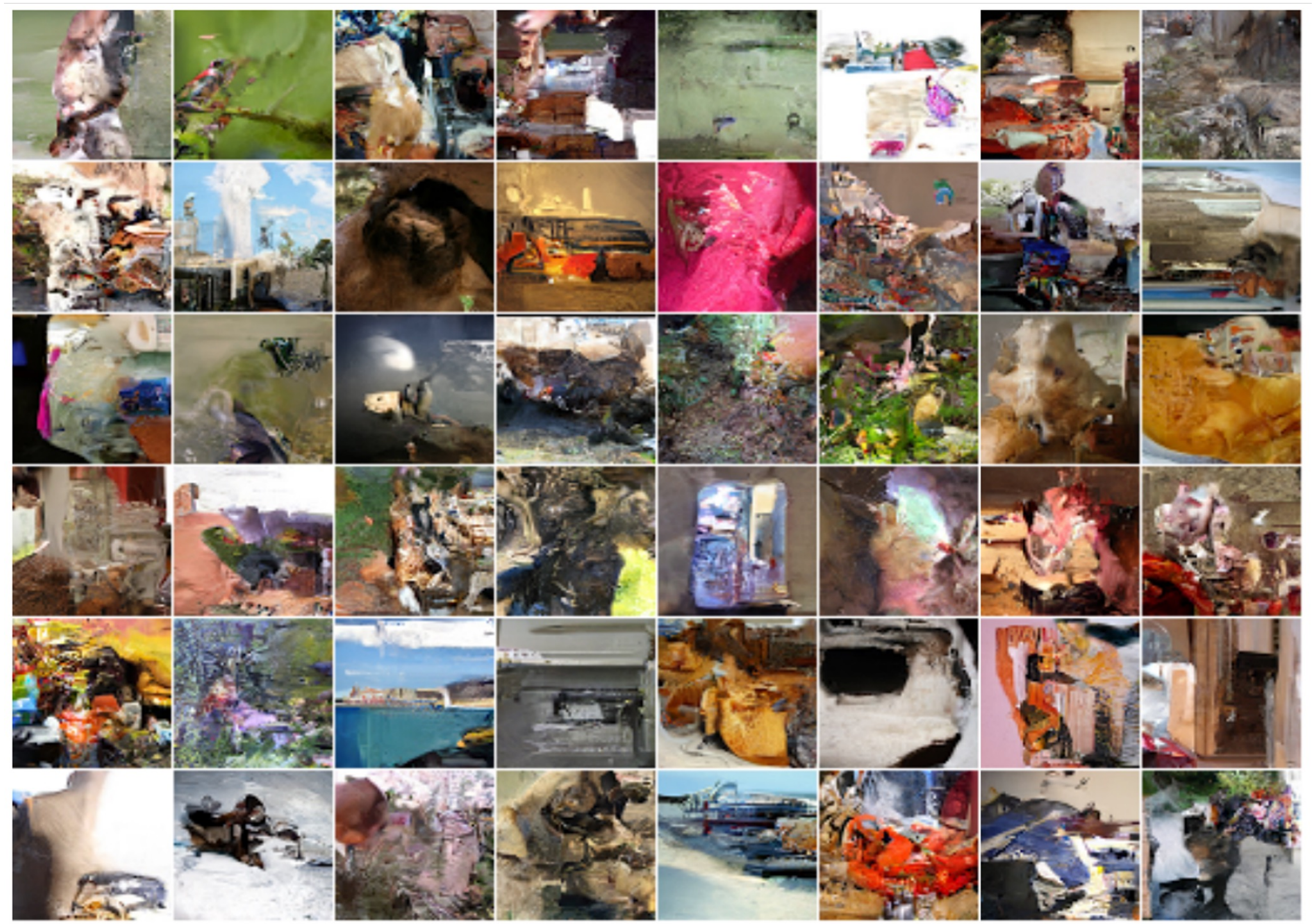
Softmax Sampling



van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Pixel RNN

Sequence of Words == Sequence of Pixels



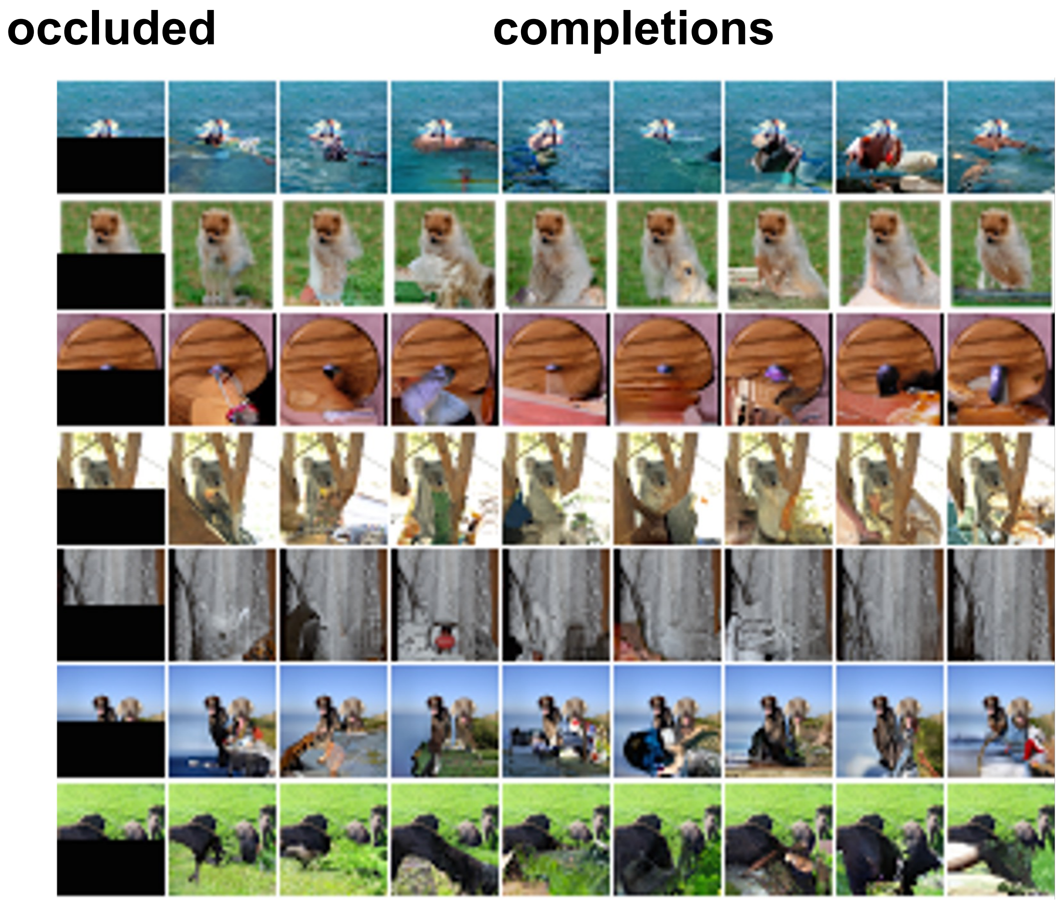
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Pixel RNN

occluded

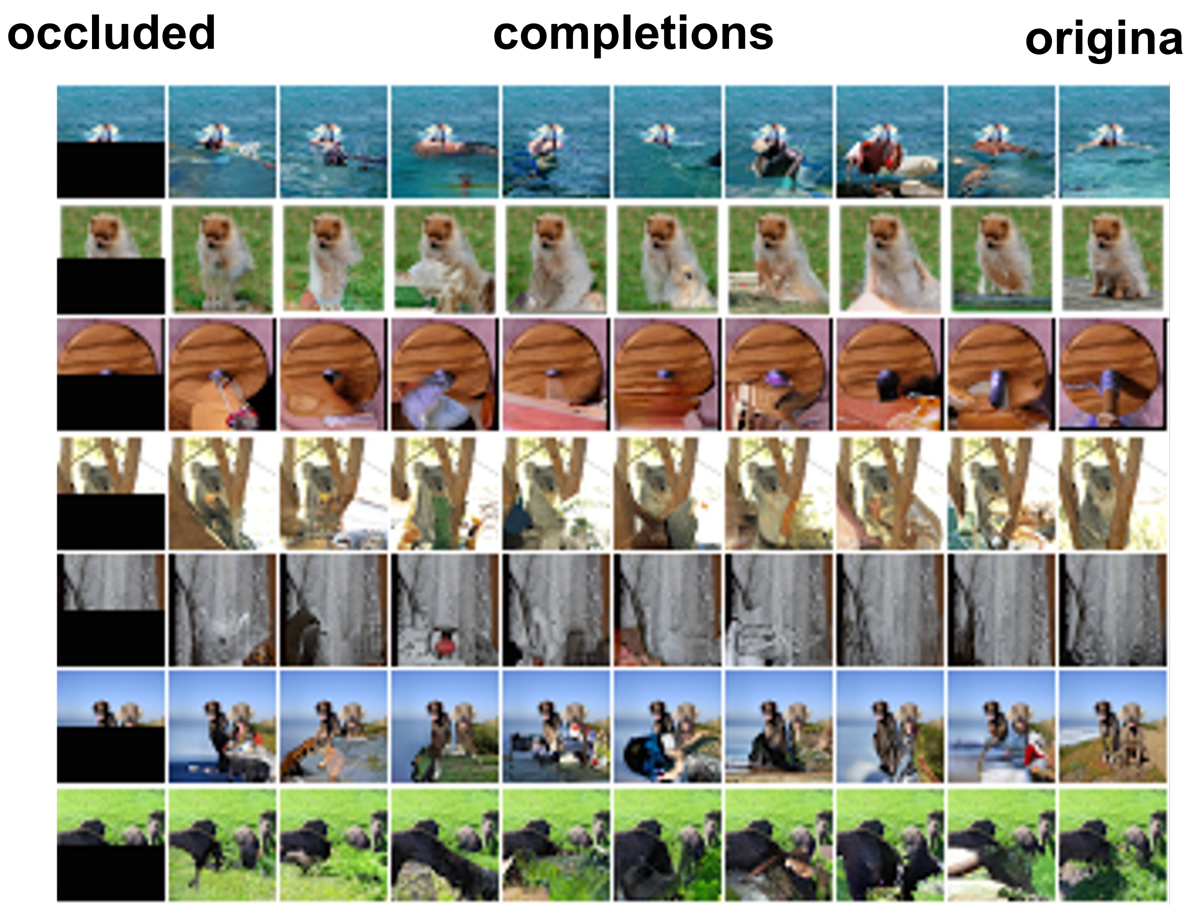
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Pixel RNN



van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Pixel RNN

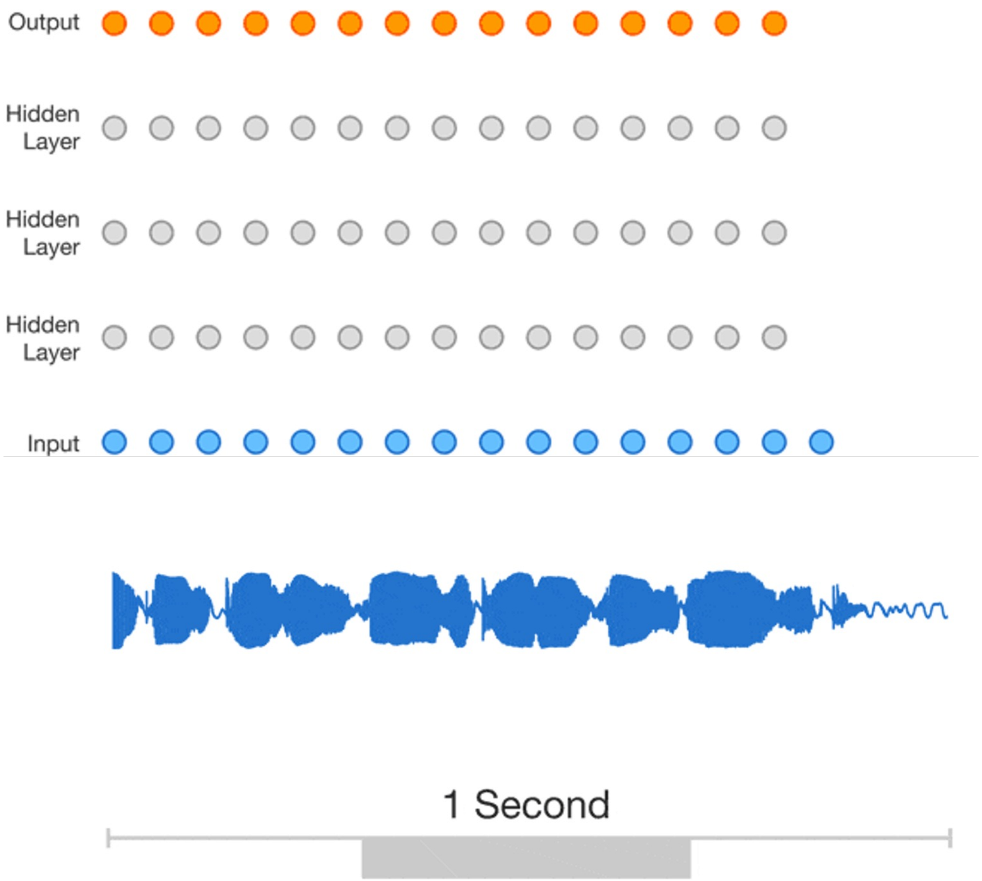


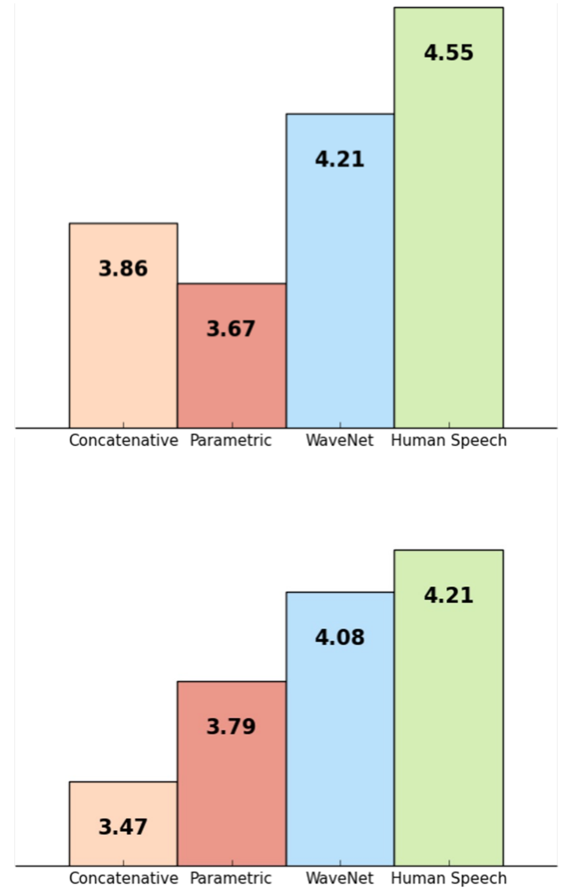
van den Oord, A., et al. “Pixel Recurrent Neural Networks.” *ICML (2016).*

Conditional Pixel CNN

van den Oord, A., et al. "Conditional Pixel CNN." *NIPS (2016).*

WaveNets

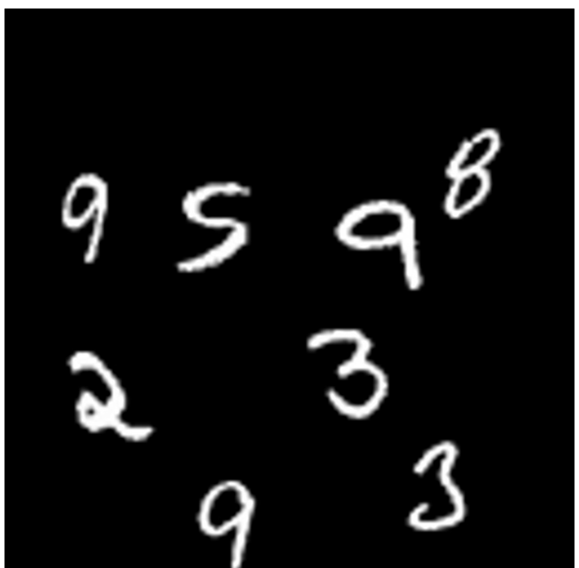




van den Oord, A., et al. "WaveNet: A Generative Model for Raw Audio." *arxiv (2016).*

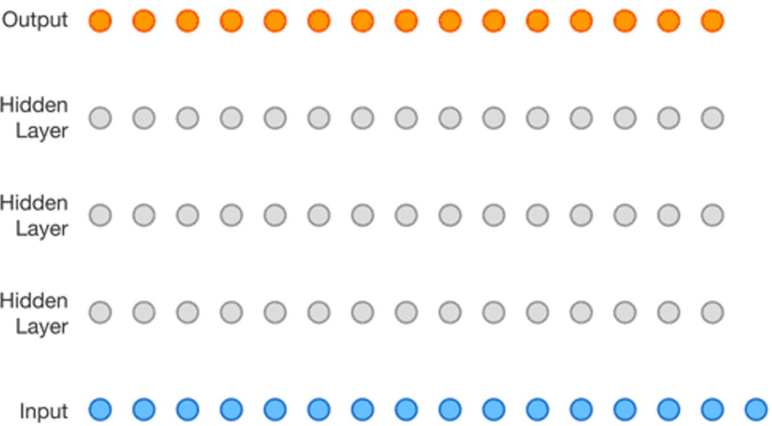
Video Pixel Network (VPN)

Model	Test
(Shi et al., 2015)	367.2
(Srivastava et al., 2015a)	341.2
(Brabandere et al., 2016)	285.2
(Patraucean et al., 2015)	179.8
Baseline model	110.1
VPN	**87.6**
Lower Bound	86.3

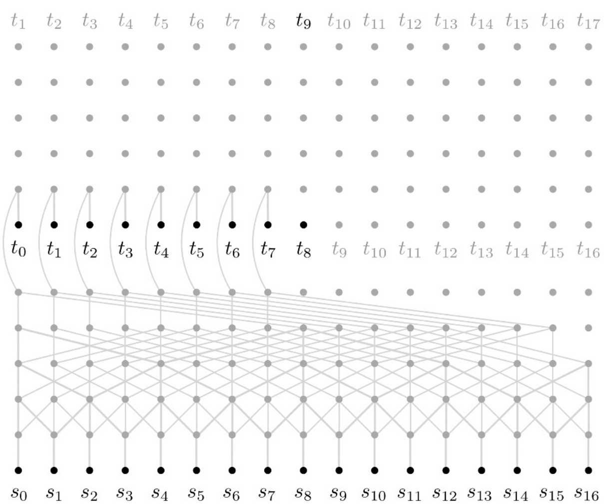


Kalchbrenner, N., et al. "Video Pixel Networks." ***ICML (2017)***.

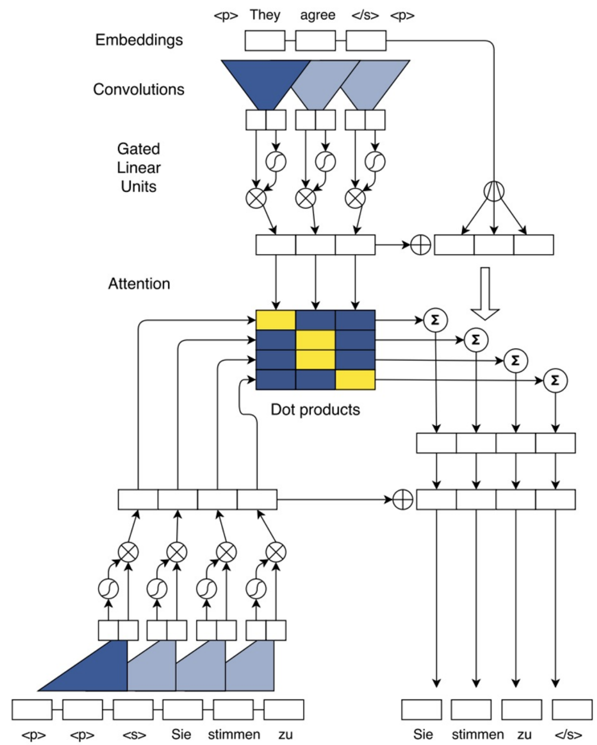
New Architectures



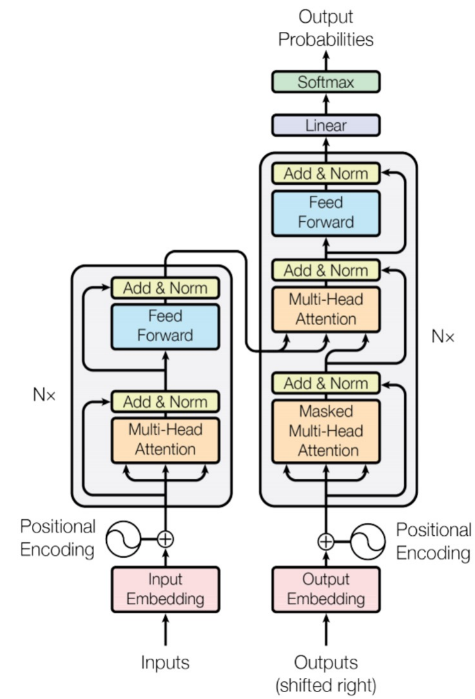
Wavenet, van den Oord, et al, 2016



Bytenet, Kalchbrenner, et al, 2016



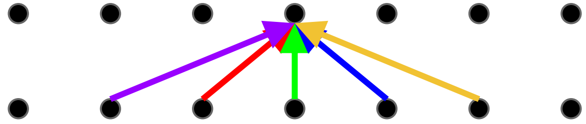
Conv seq2seq, Gehring, et al, 2017



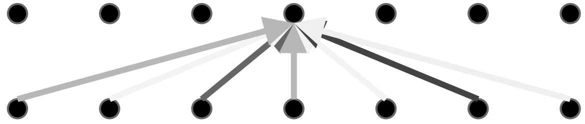
Att is all you need, Vaswani, et al, 2017

Self-Attention

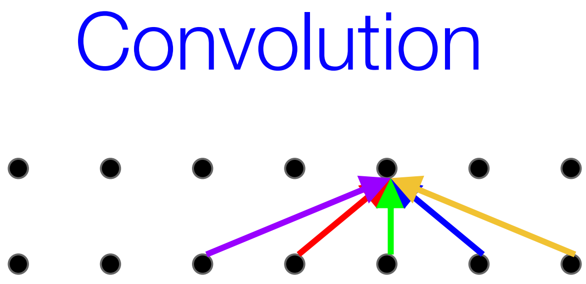
Convolution

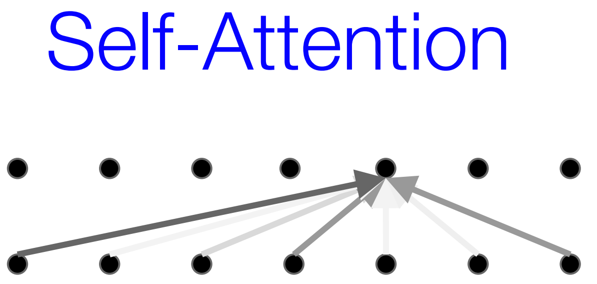


Self-Attention



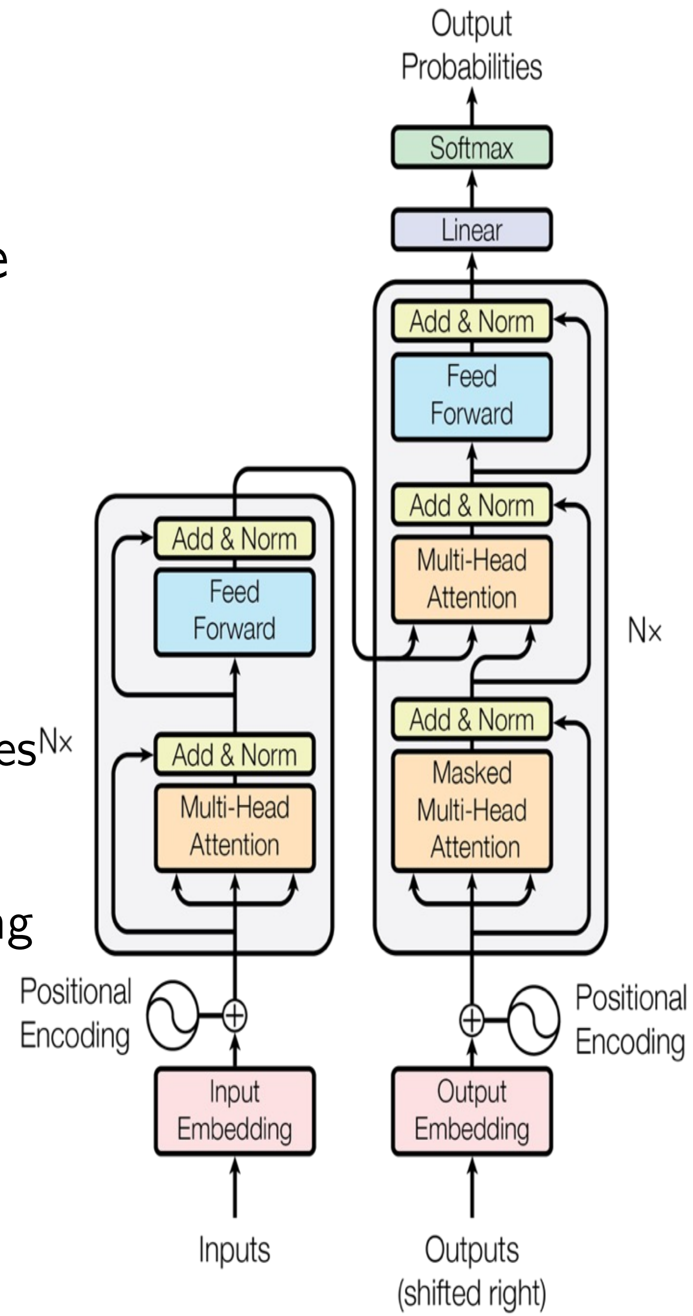
Self-Attention





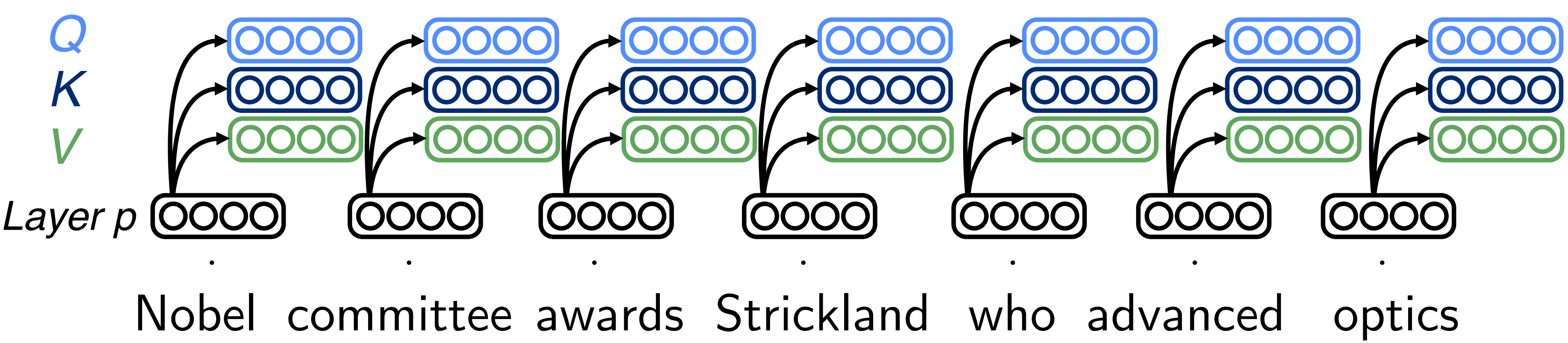
Transformer Networks

- Originally designed for Neural Machine Translation
- Input/Output Embedding Layer:
  - Lookup table from discrete tokens to continuous word representations
- Positional Encoding
  - Adding temporal information into sequences
- Encoder/ Decoder
  - Performing Sequence-to-Sequence Modeling
  - Core: Scaled Dot-Product Attention Mechanism
- Output Probability Layer
  - Lookup table from continuous word representations to discrete tokens



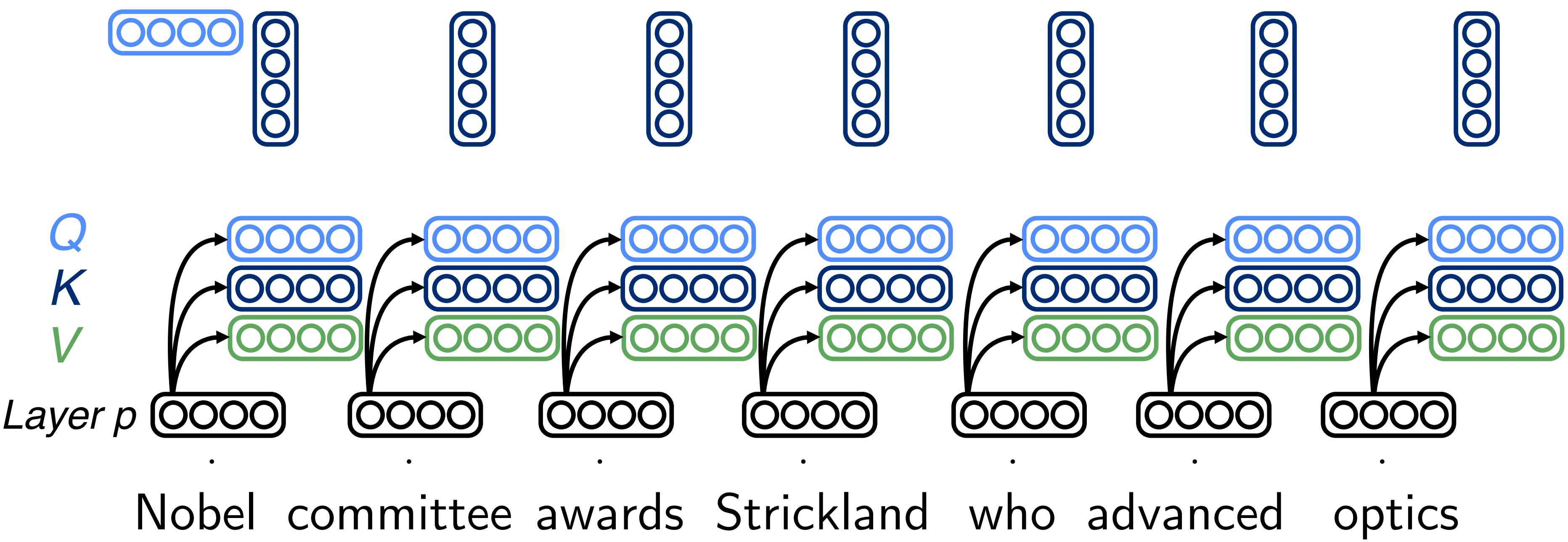
Self Attention

[Vaswani et al. 2017], Slides borrowed form Emma Strubell



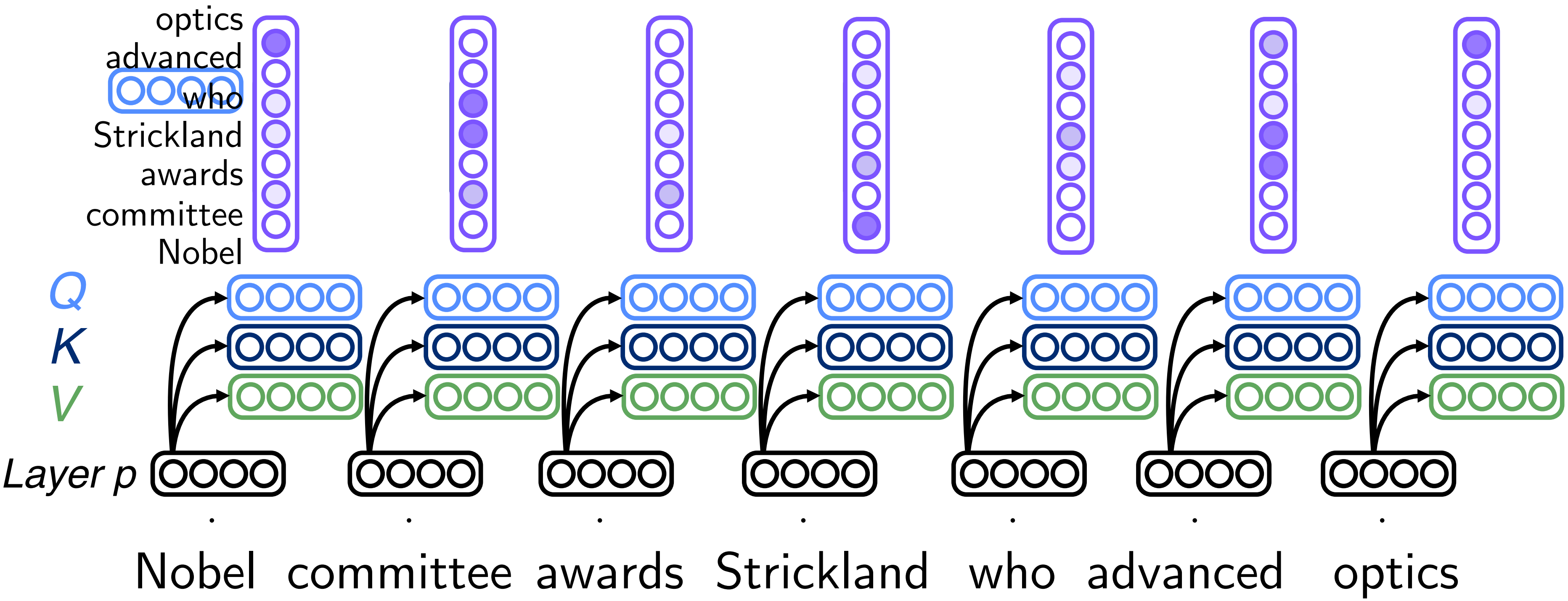
Self Attention

[Vaswani et al. 2017],
Slides borrowed form
Emma Strubell



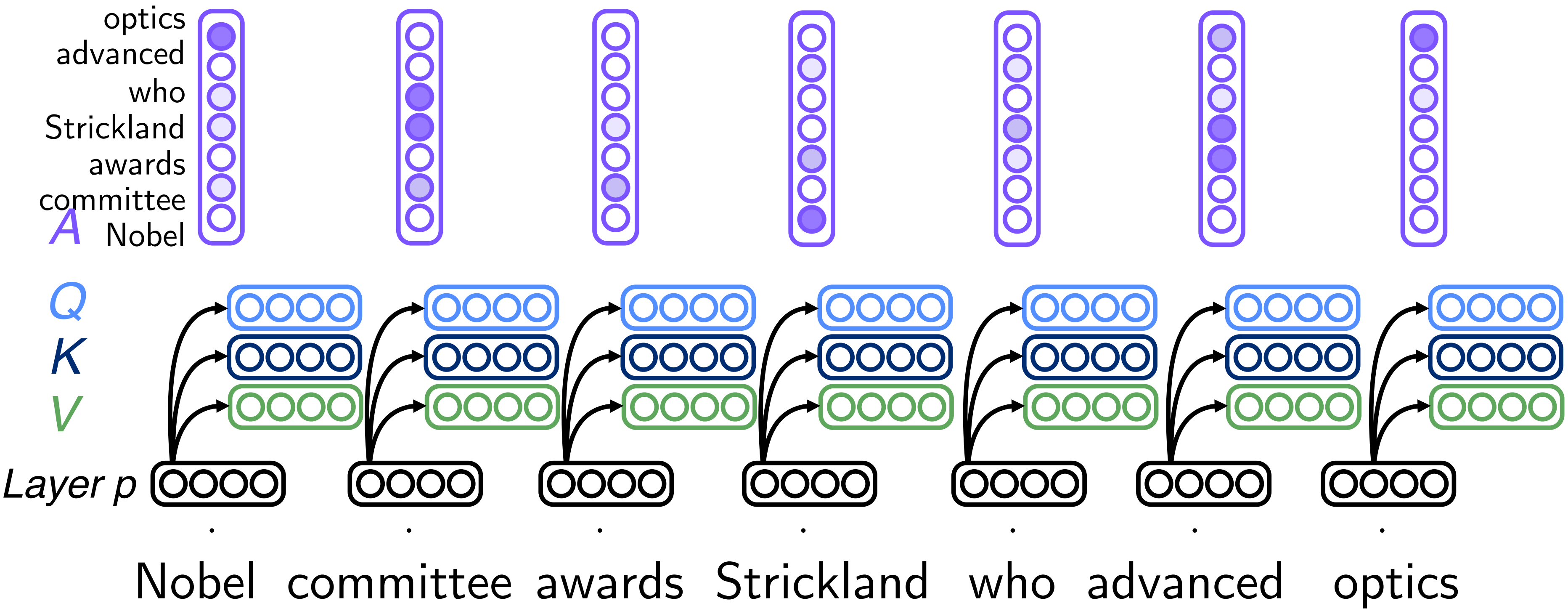
Self Attention

[Vaswani et al. 2017],
Slides borrowed form
Emma Strubell



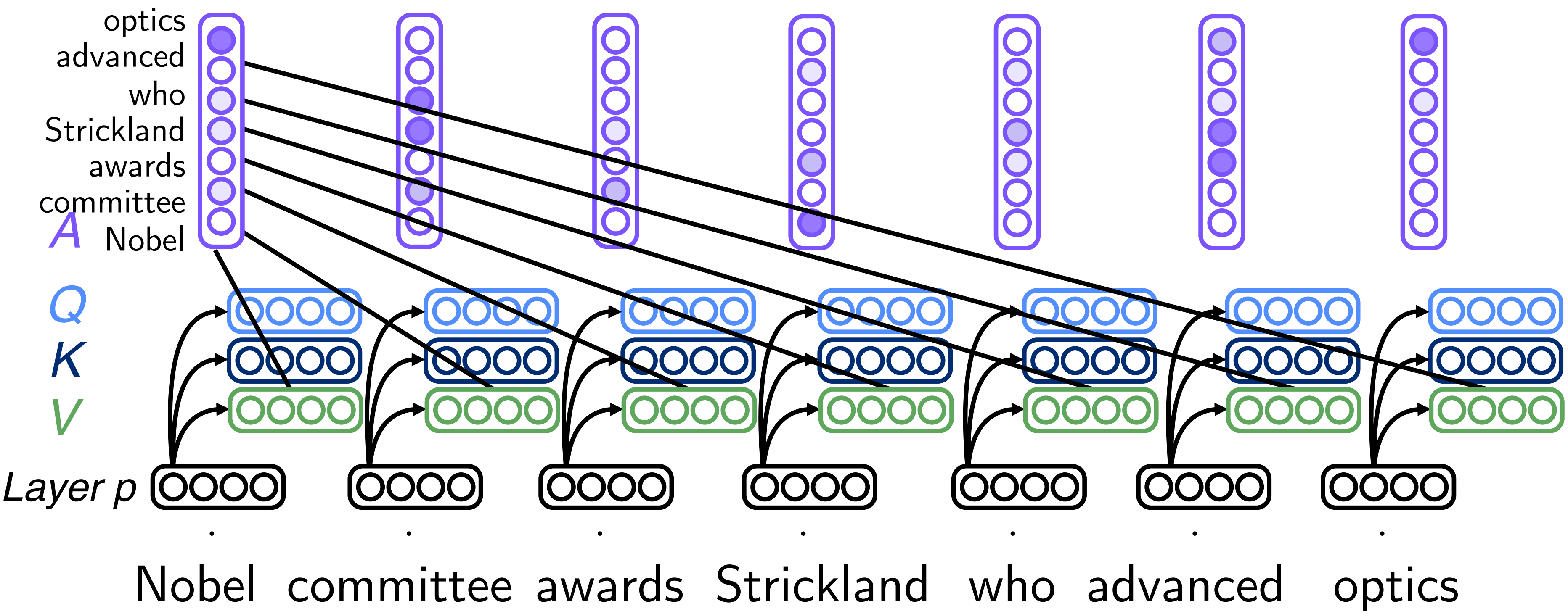
Self Attention

[Vaswani et al. 2017], Slides borrowed form Emma Strubell



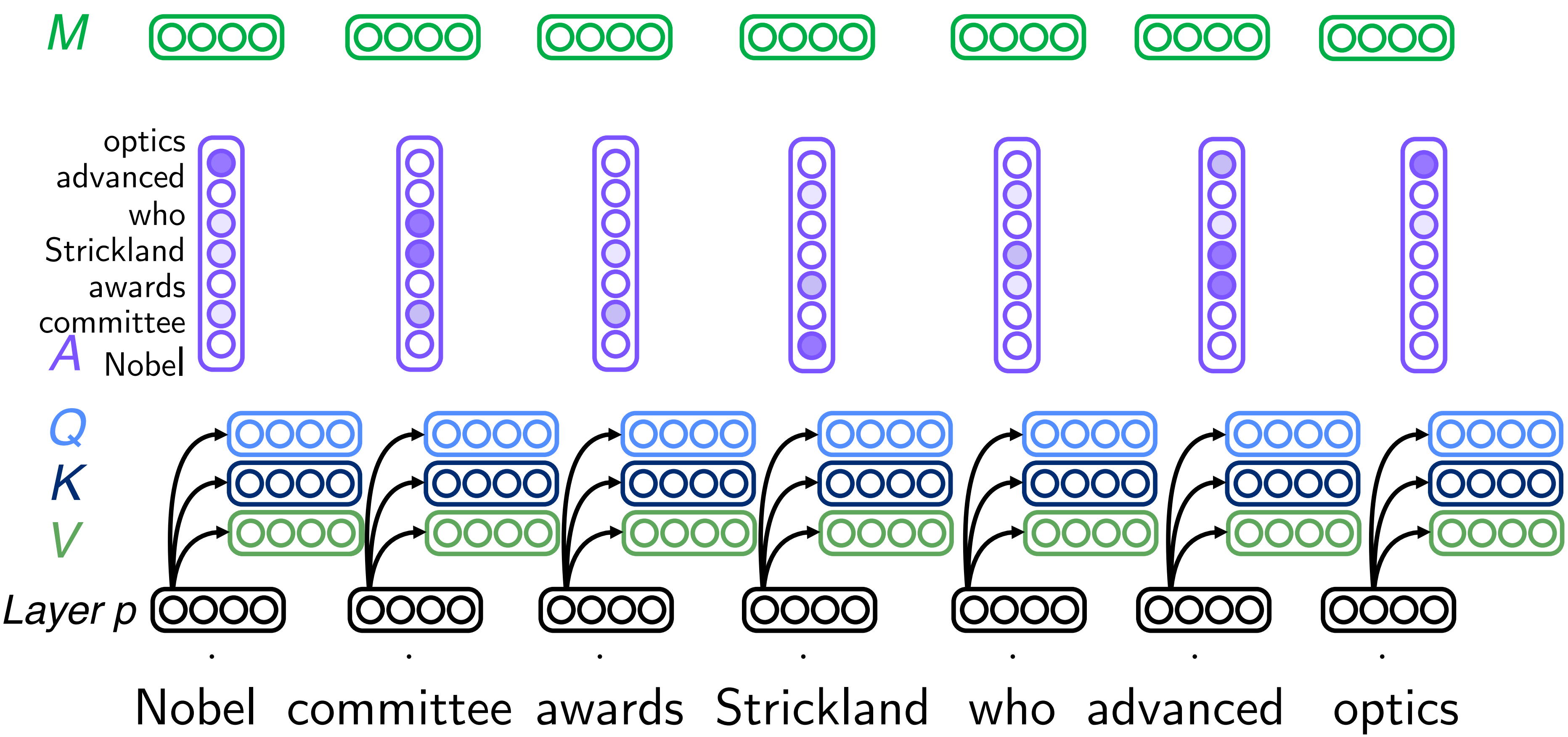
Self Attention

[Vaswani et al. 2017], Slides borrowed form Emma Strubell



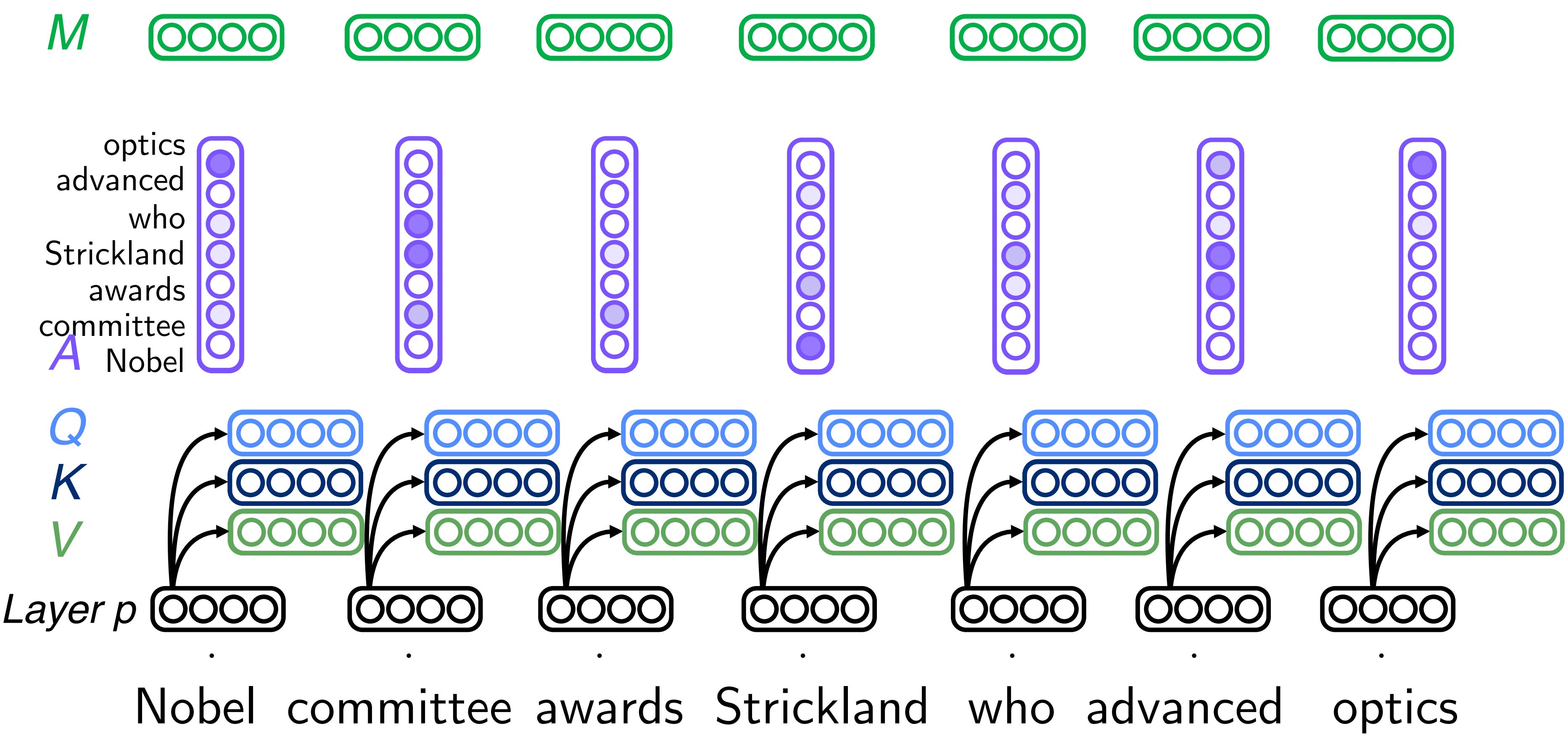
Self Attention

[Vaswani et al. 2017],
Slides borrowed form
Emma Strubell



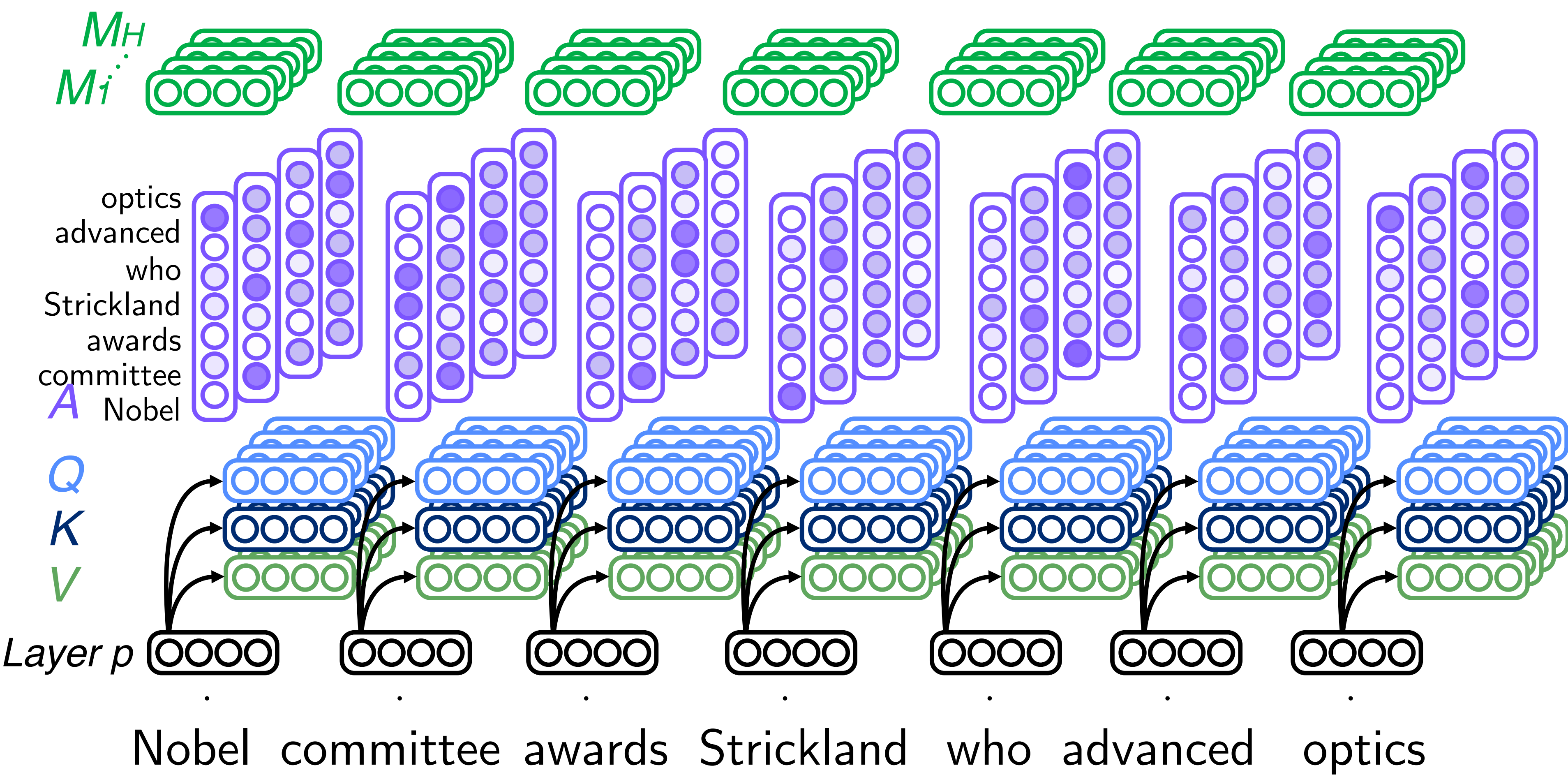
Self Attention

[Vaswani et al. 2017], Slides borrowed form Emma Strubell



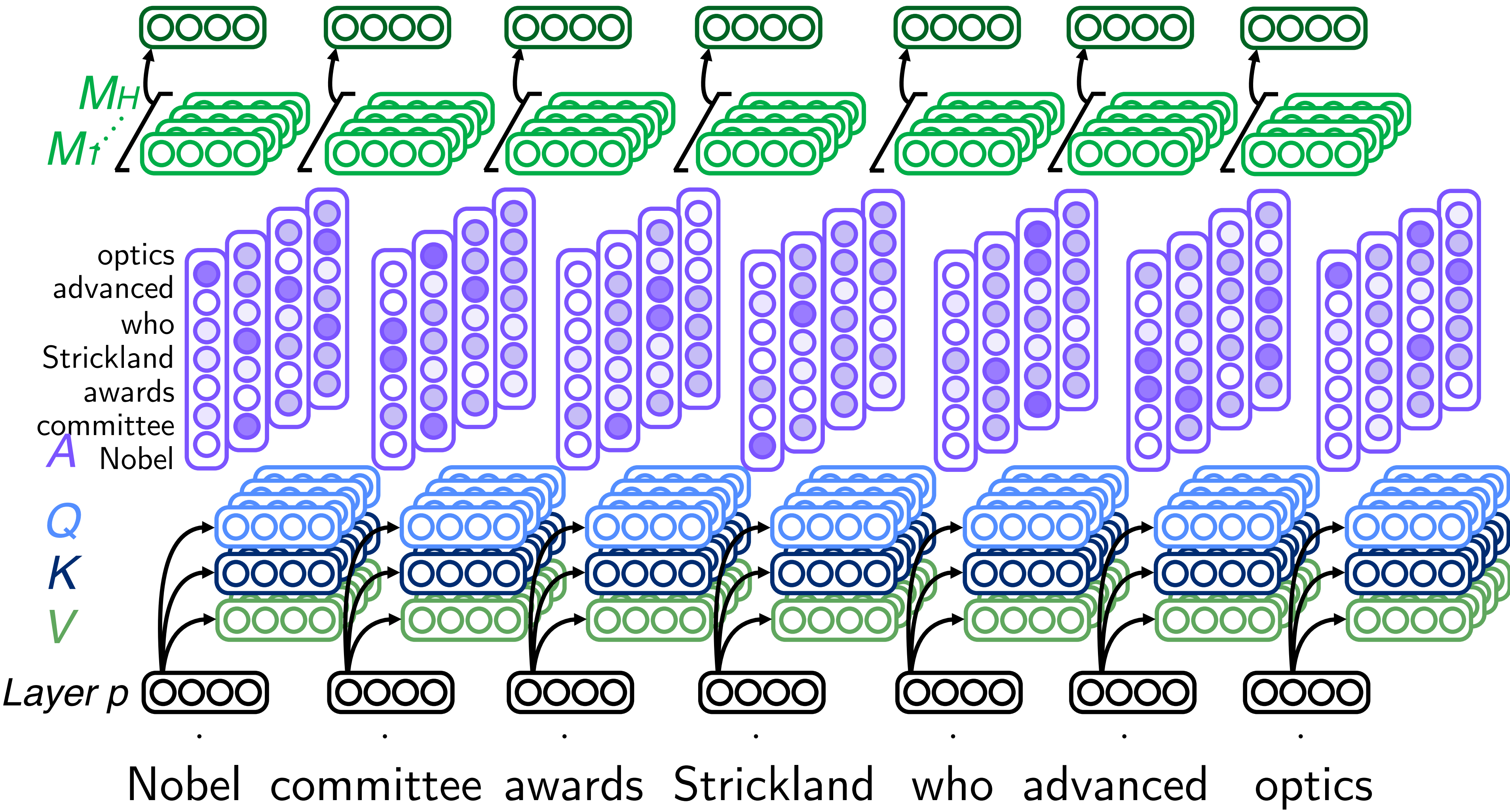
Multi-Head Self Attention

[Vaswani et al. 2017],
Slides borrowed form
Emma Strubell



Multi-Head Self Attention

[Vaswani et al. 2017], Slides borrowed form Emma Strubell



Transformer Networks

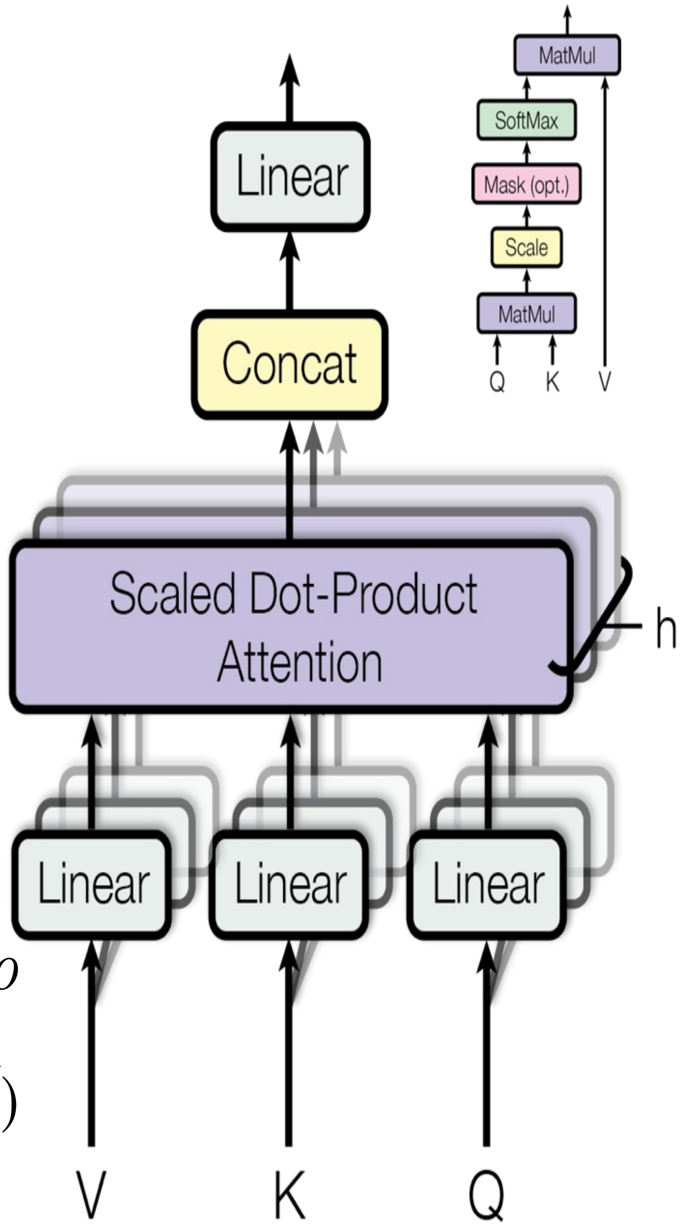
- Core: Scaled Dot-Product Attention Mechanism
  - Also called Single-Head Attention

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V$$

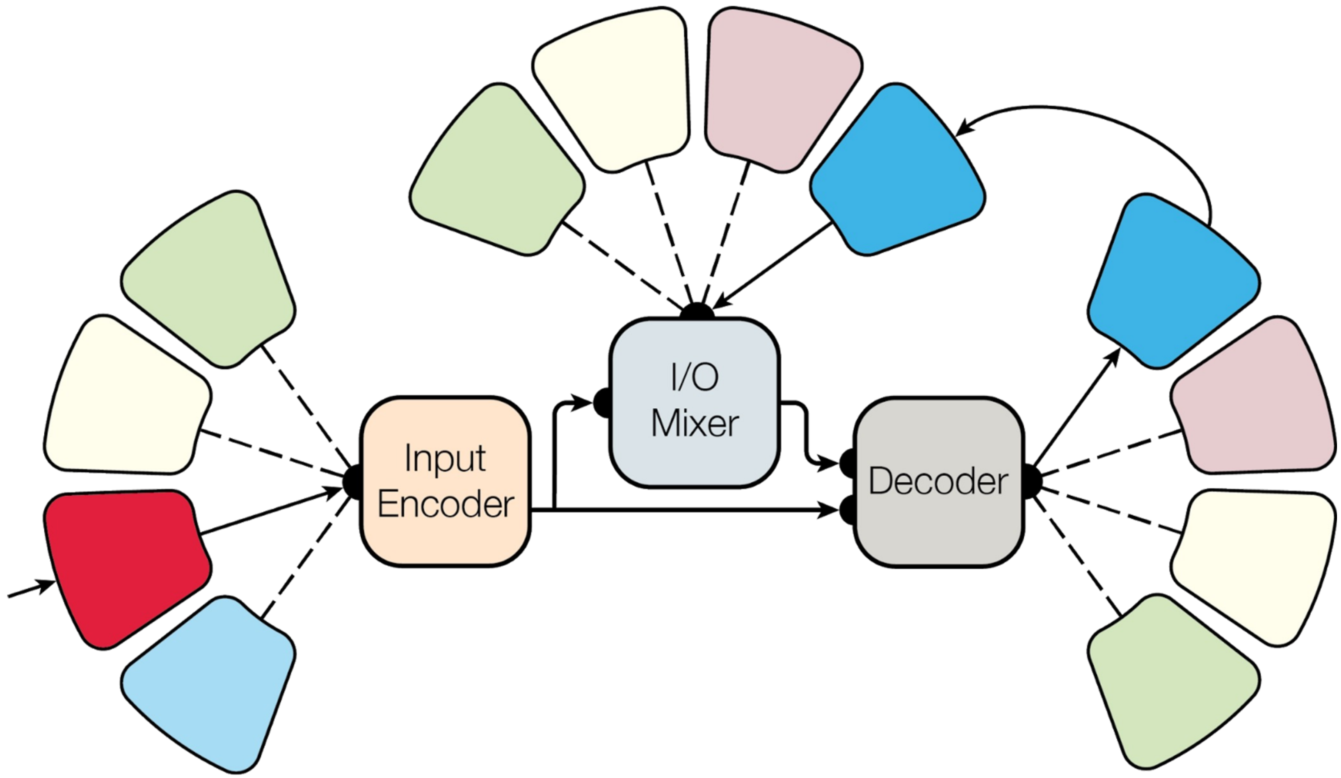
- Multi-Head Attention
  - Consider multiple attention hypothesis

$$\text{MutliHead}(Q, K, V) = \text{Concat}(\text{head}_1, \cdots, \text{head}_h)W^O$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

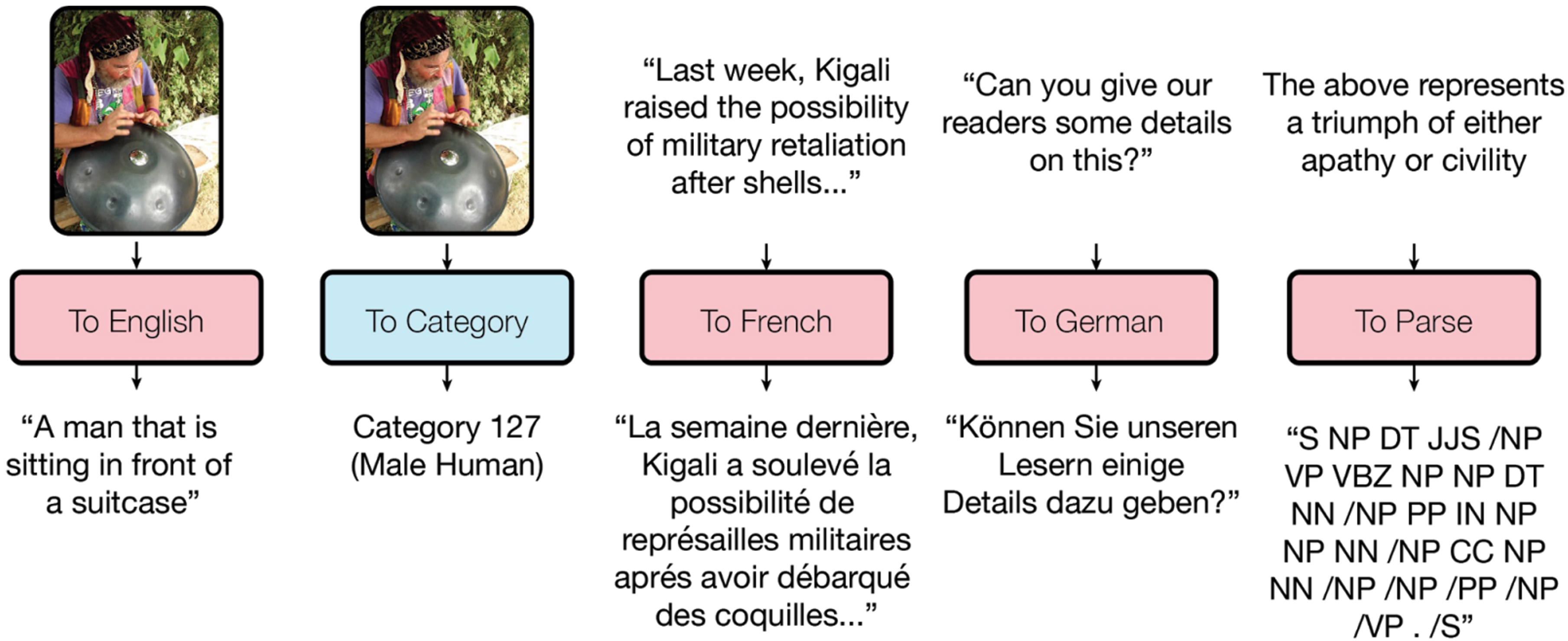


MultiModel



Kaiser, L., et al. “One Model To Learn Them All.” *arxiv (2017).*

MultiModel



Kaiser, L., et al. “One Model To Learn Them All.” *arxiv (2017).*