Sapienza Training Camp 2021

Building an Image Search Engine

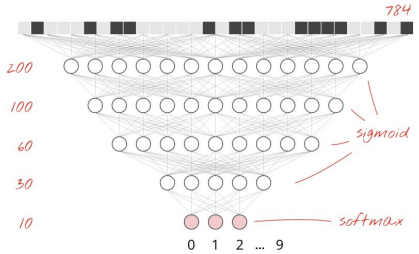
2 - 4 September, 2021

Recap: What did we learn so far?

Recap: Dense and convolutional layers

- Dense layer:

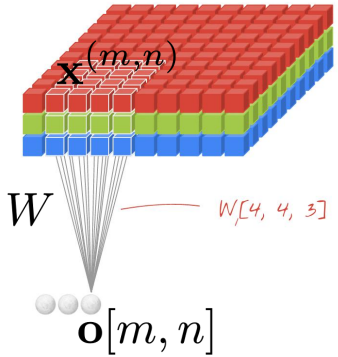
$$\mathbf{o} = \mathrm{g}(W\mathbf{x} + w_0), \quad \text{where } \mathbf{x} \in \mathbb{R}^d$$



- Convolutional layer:

$$\mathbf{o}[m,n] = \mathrm{g}(W\mathbf{x}^{(m,n)} + w_0), \quad \text{where } \mathbf{x}^{(m,n)} = \mathbf{x}[m:m+D, n:n+D]$$

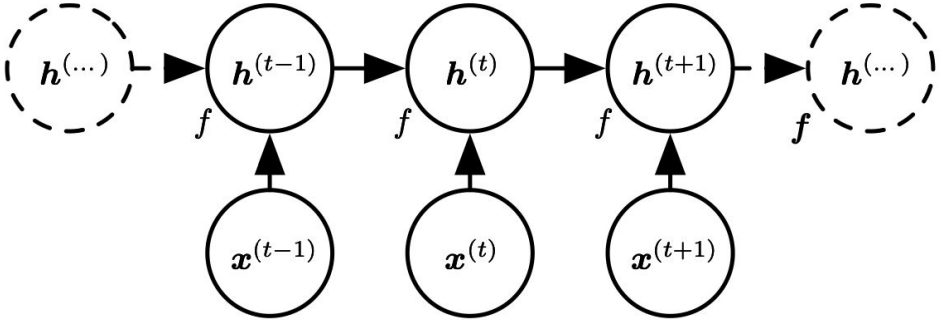
  - weight sharing: same weights used for all local windows



Images from "TensorFlow, Keras and deep learning, without a PhD"

Recap: Recurrent layer

- Recurrent layer
  - weight sharing across time steps

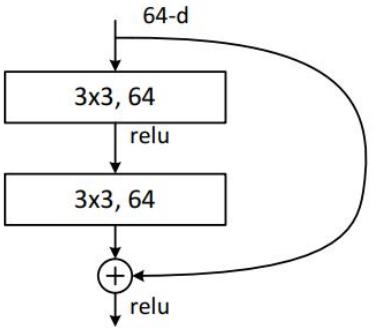


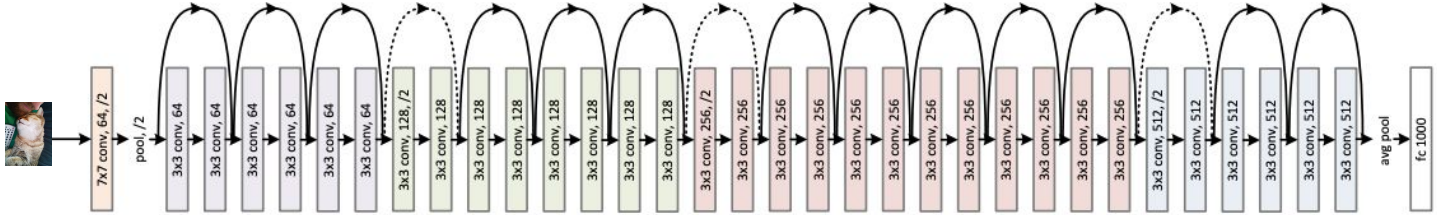
$$\mathbf{h}^{(t)} = \mathrm{g}(W\mathbf{h}^{(t-1)} + U\mathbf{x}^{(t)} + w_0)$$

$$\text{where } \mathbf{h}^{(t)} \in \mathbb{R}^K$$

Image from "https://www.deeplearningbook.org/slides/10_rnn.pdf"

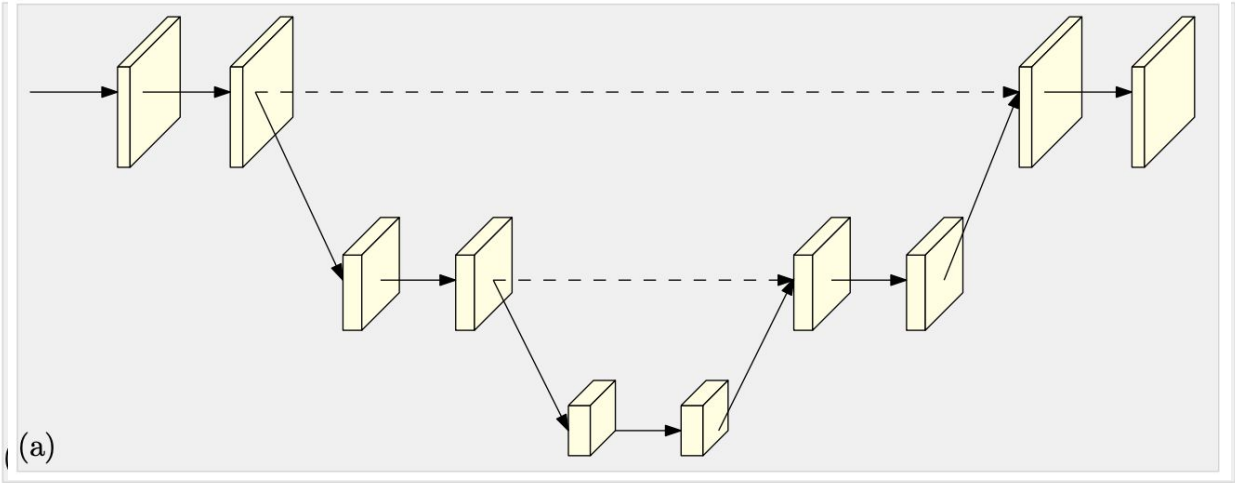
Recap: Neural network architectures

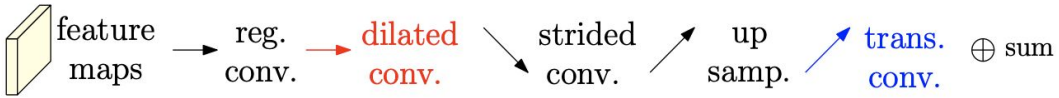




He et al., Deep Residual Learning for Image Recognition, CVPR'16

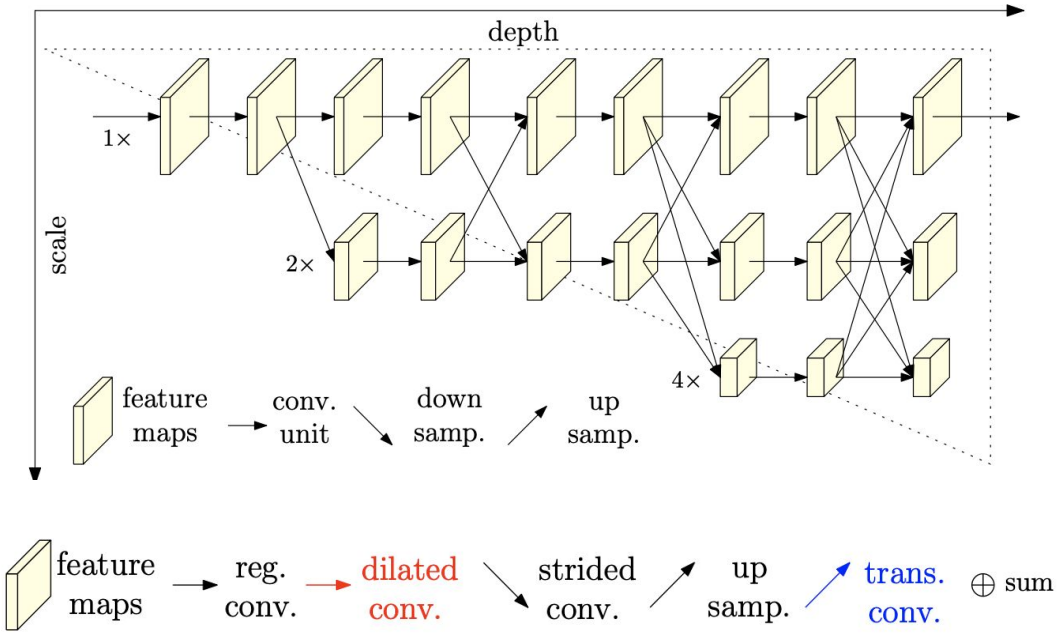
Hourglass CNN





Newell et al., Stacked Hourglass Networks for Human Pose Estimation, ECCV'16

HRNet



Wang et al., Deep high-resolution representation learning for visual recognition, PAMI'20

Roadmap

Image from the photo collection

I_k

Neural Network (CNN)

Image encoding

φ_k

Recurrent neural network (RNN)

Image description

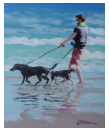
T_k

= “dog next to person on the bike near street crossing”

Define similarity function. Order images according to similarity to the query.

Q

Query: “person walking with a dog on the beach”



$$\text{sim}(Q, T_1) > \text{sim}(Q, T_2)$$

More complex RNN units: GRU and LSTM

```
class RNN_Decoder(tf.keras.Model):
  def __init__(self, embedding_dim, units, vocab_size):
    super(RNN_Decoder, self).__init__()
    self.units = units

    self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
    self.gru = tf.keras.layers.GRU(self.units,
                                   return_sequences=True,
                                   return_state=True,
                                   recurrent_initializer='glorot_uniform')
    self.fc1 = tf.keras.layers.Dense(self.units)
    self.fc2 = tf.keras.layers.Dense(vocab_size)

    self.attention = BahdanauAttention(self.units)
```

GRU?

Difficulty of training RNNs

- Hard to retain long term dependencies in the state vector

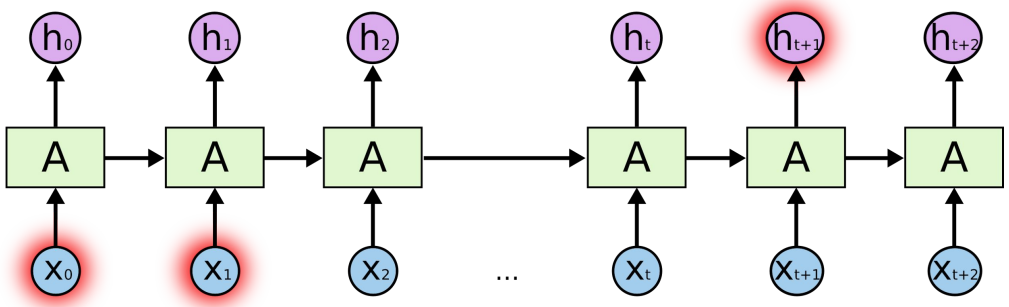
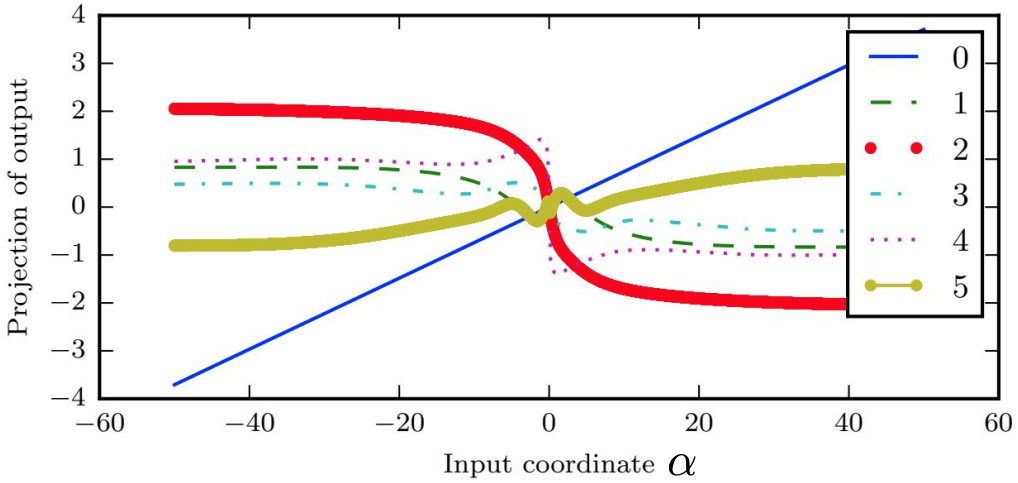


Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Difficulty of training RNNs

- Vanishing/exploding gradients

$\left\langle \mathbf{r}, \mathbf{h}^{(t)} \right\rangle$

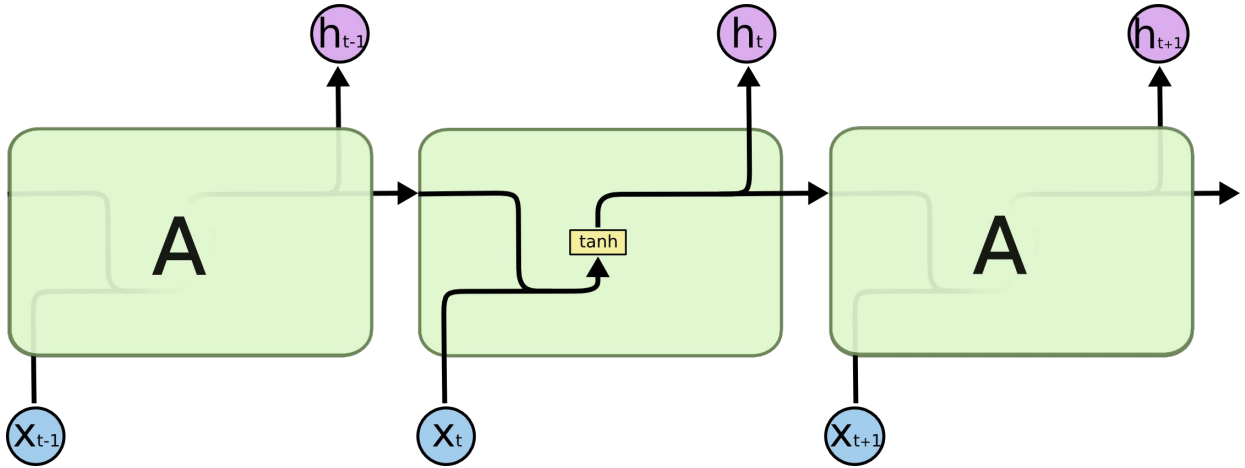


$$\mathbf{h}^{(0)} = \alpha \bar{\mathbf{r}}$$

$$\mathbf{h}^{(t)} = \tanh(W\mathbf{h}^{(t-1)} + w_0)$$

Image from "https://www.deeplearningbook.org/slides/10_rnn.pdf"

Simple RNN unit



$$\mathbf{h}^{(t)} = \tanh(W\mathbf{h}^{(t-1)} + U\mathbf{x}^{(t)} + w_0)$$

Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)

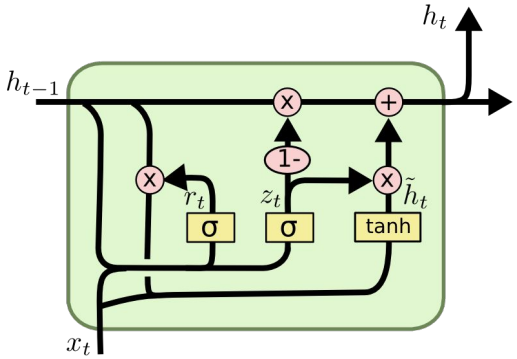
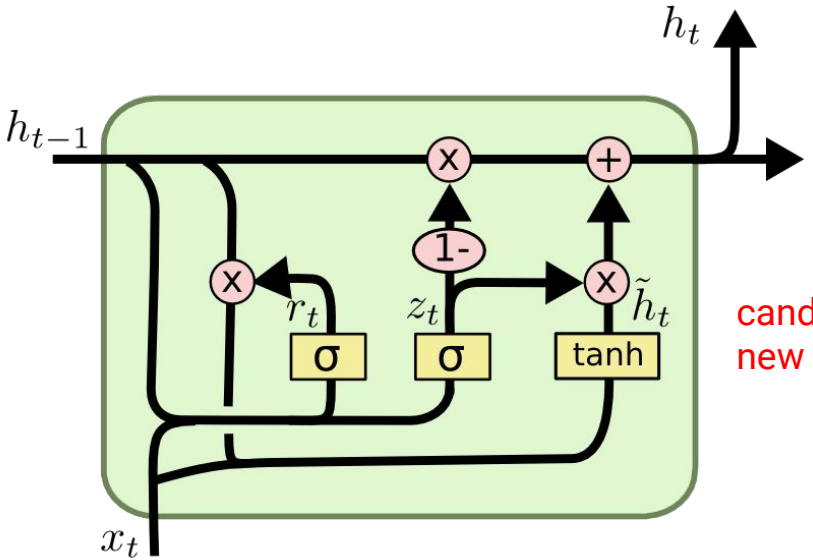


Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

candidate

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

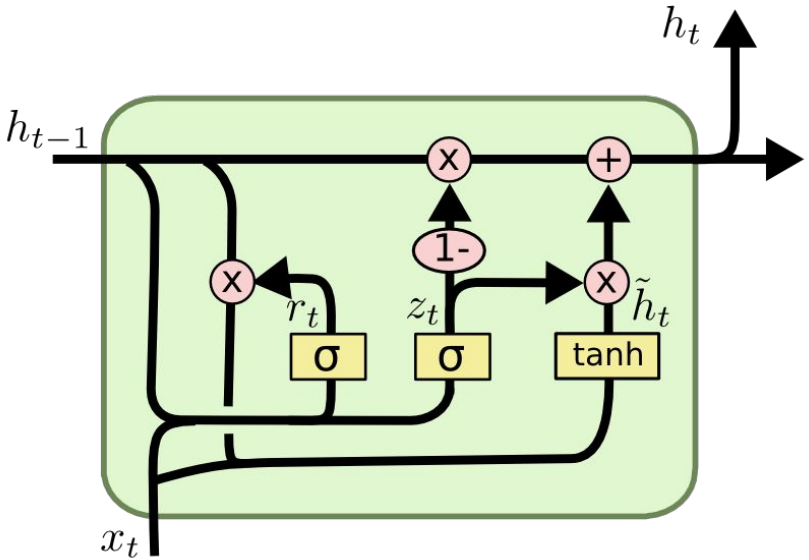
new state

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

recall the skip connections in the ResNet!

Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$ update gate

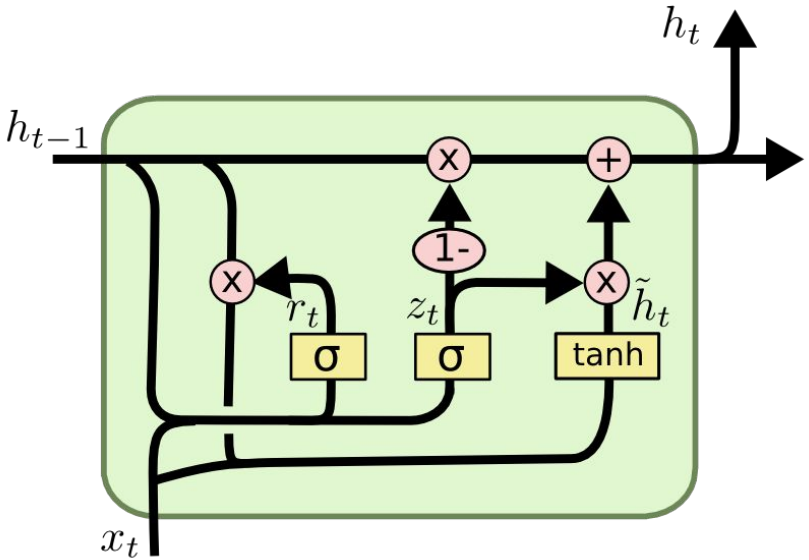
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

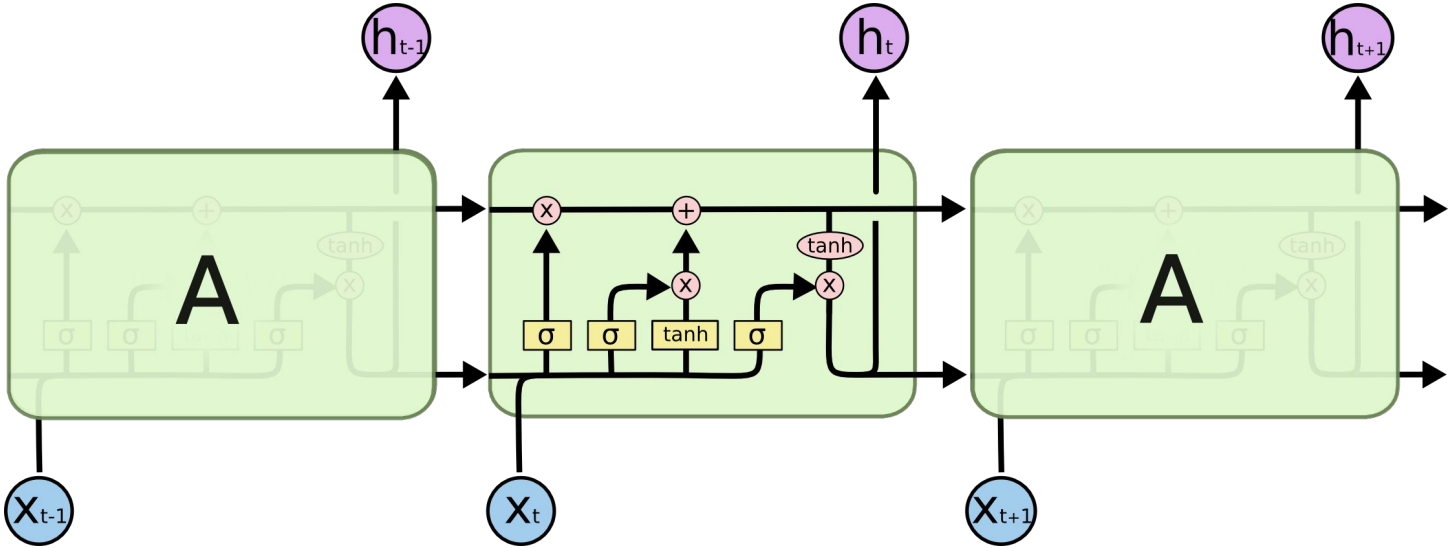
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$ reset gate

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-term Memory (LSTM)



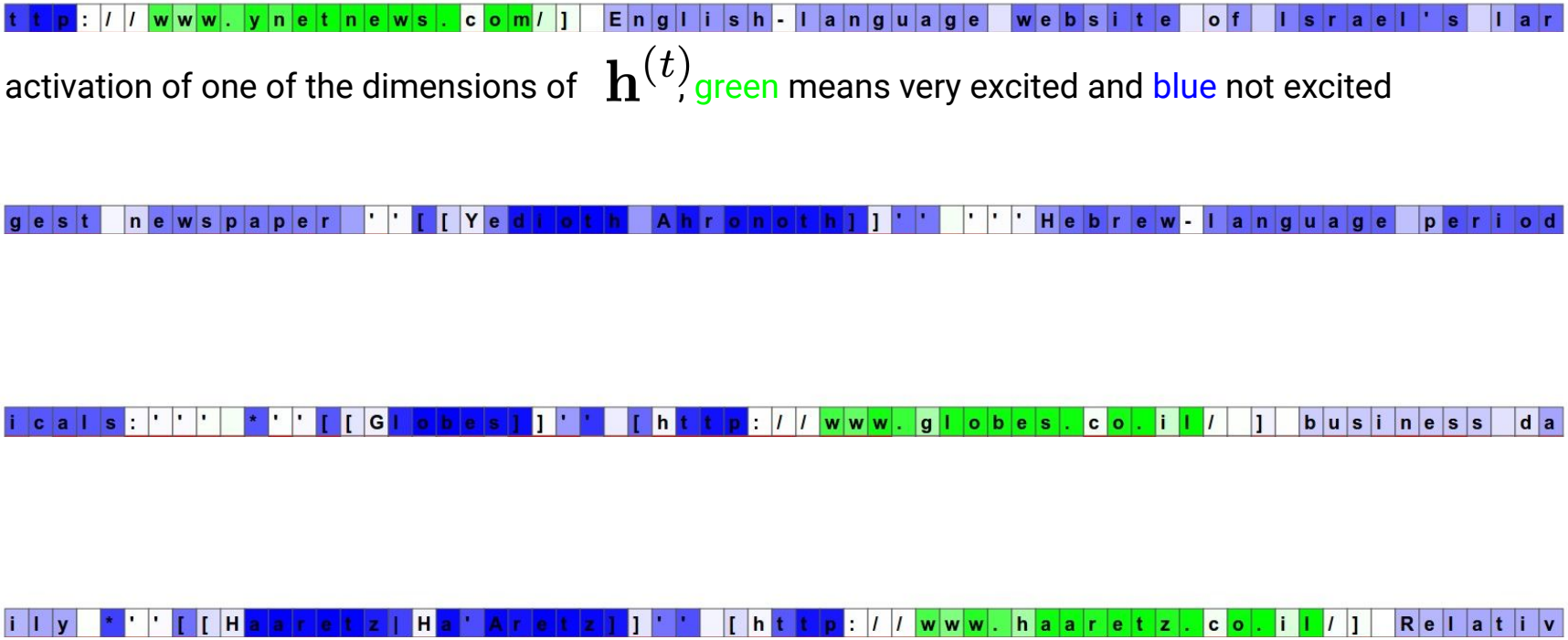
Hochreiter&Schmidhuber, Long short-term Memory, Neural Computation, 1997

Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Visualizing the hidden state of an RNN

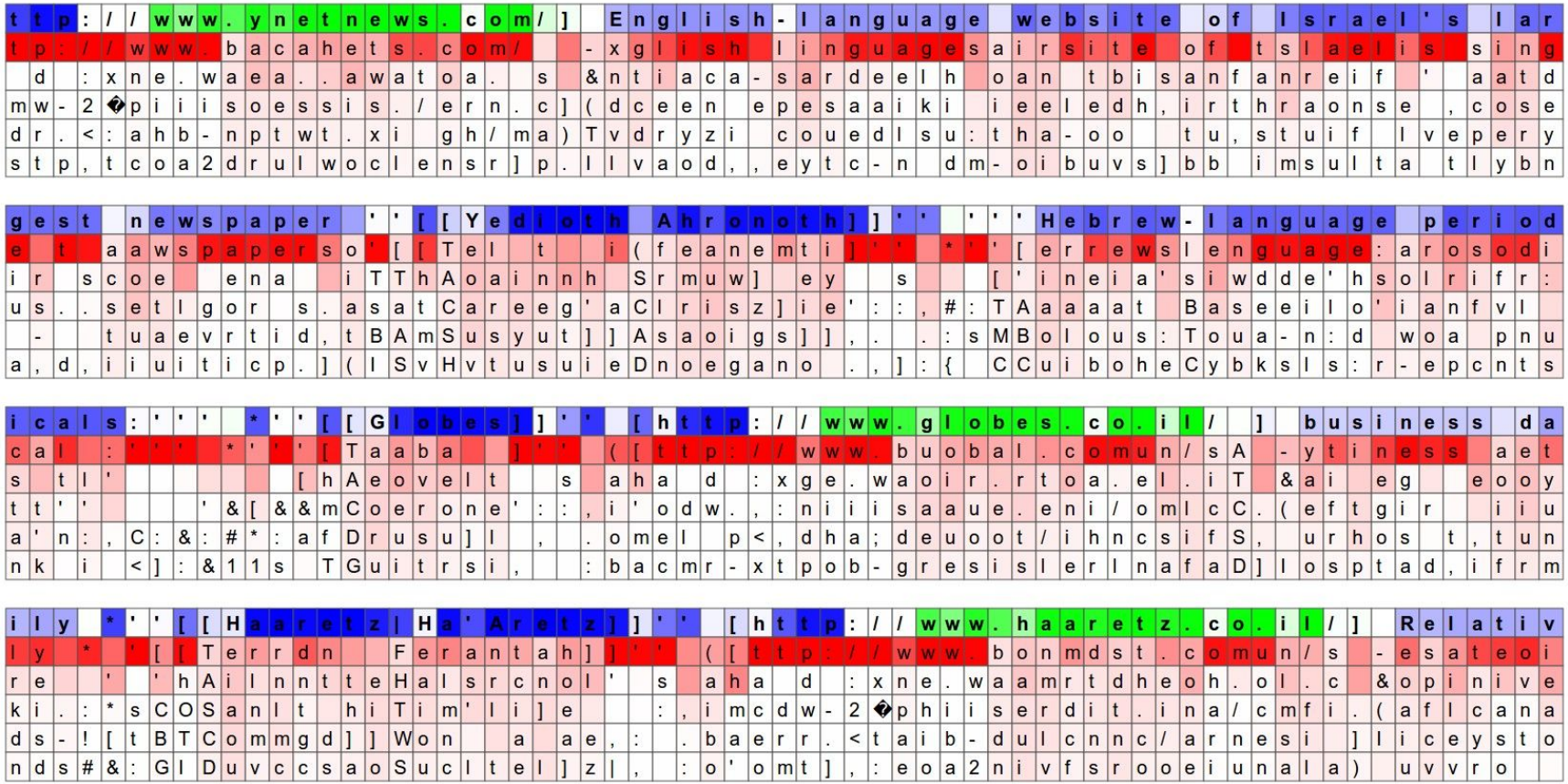
- Character level RNN with LSTM units trained on a large corpora of text (wikipedia)
  - output is a single character
  - RNN has to learn to form words, punctuation and mark-up
- http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Visualizing RNN state



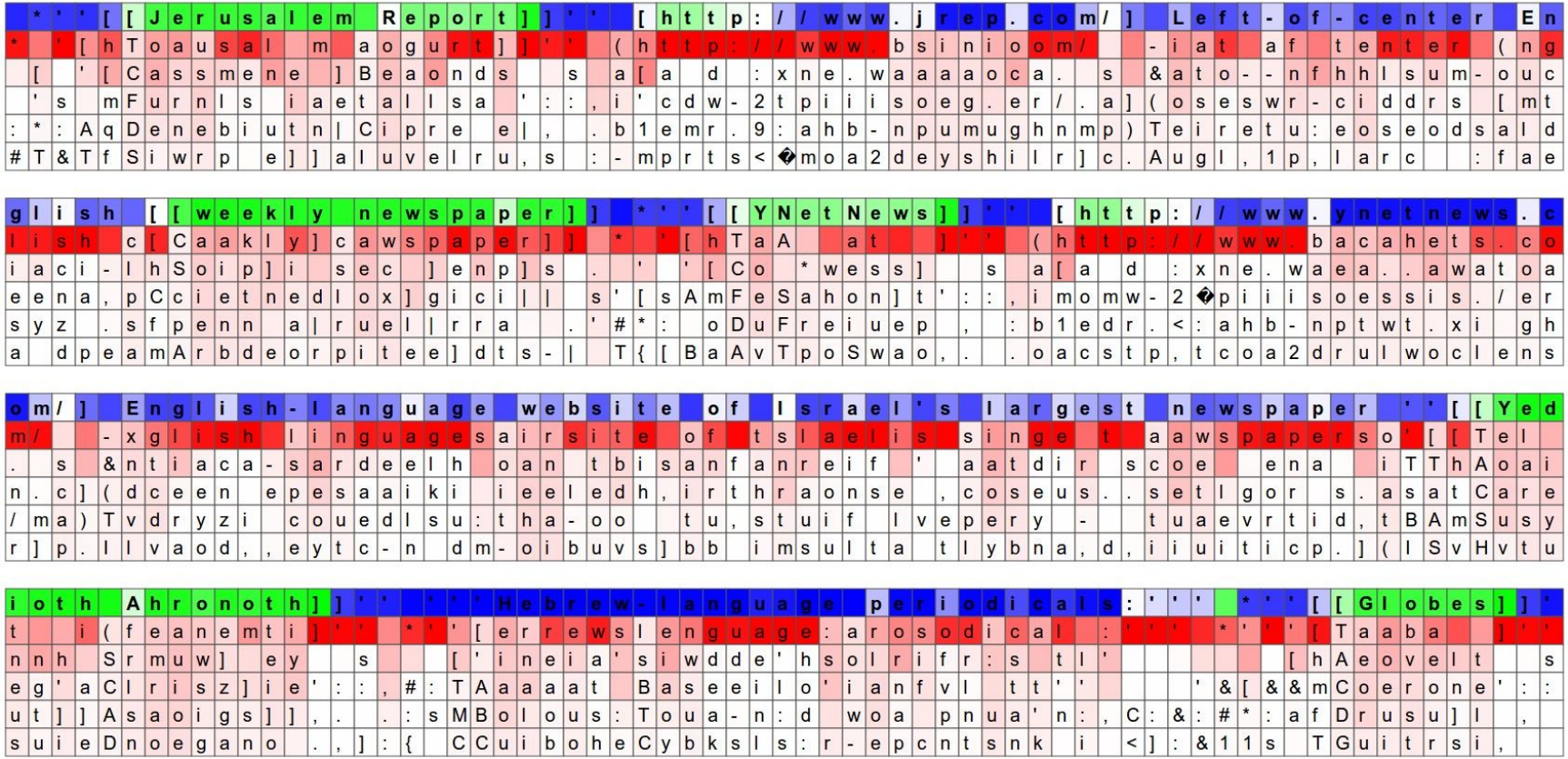
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Visualizing RNN state



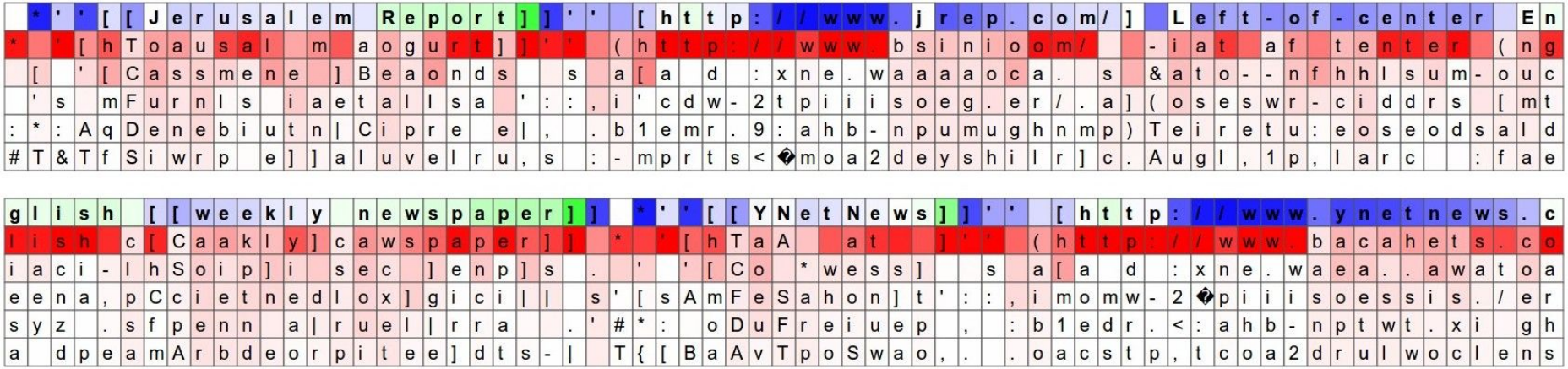
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Visualizing RNN state



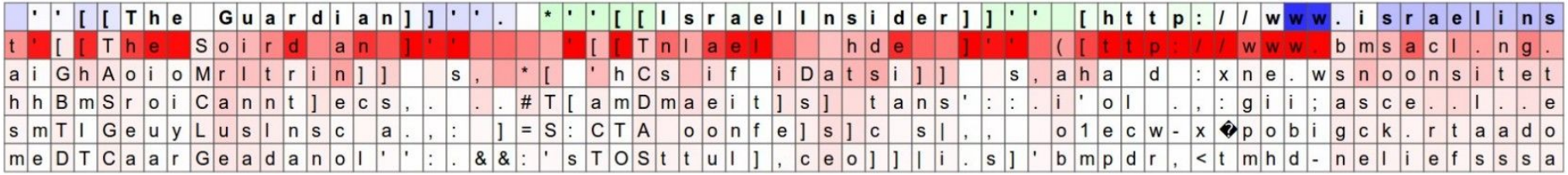
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Visualizing RNN state



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Visualizing RNN state



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Roadmap

Image from the photo collection I

Neural Network (CNN)

Image encoding φ

Recurrent neural network (RNN)

Image description T_k = “dog next to person on the bike near street crossing”

Q Query: “person walking with a dog on the beach”

Define similarity function. Order images according to similarity to the query.

$$\text{sim}(Q, T_1) > \text{sim}(Q, T_2)$$

Image captioning model

Image from the photo collection

I

Neural Network (CNN)

Image encoding

φ

Recurrent neural network (RNN)

T_k

Image captioning model

Image from the photo collection

I

Inception CNN

Image encoding

φ

Recurrent neural network (RNN)

T_k

Image captioning model

Image from the photo collection

I

Inception CNN

Image encoding

φ

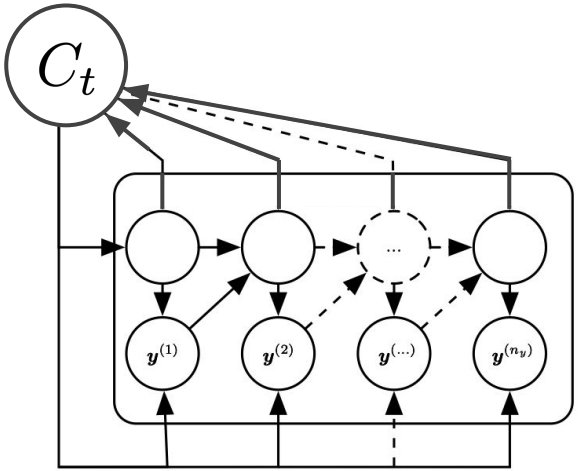


Image captioning model

Image from the photo collection

I

Inception CNN

Image encoding

φ

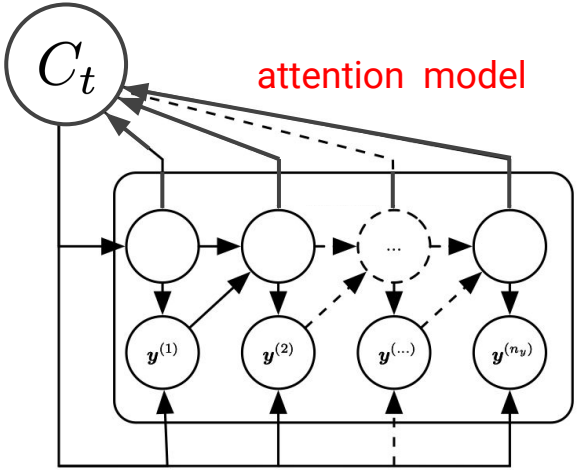


Image captioning model

Image from the photo collection

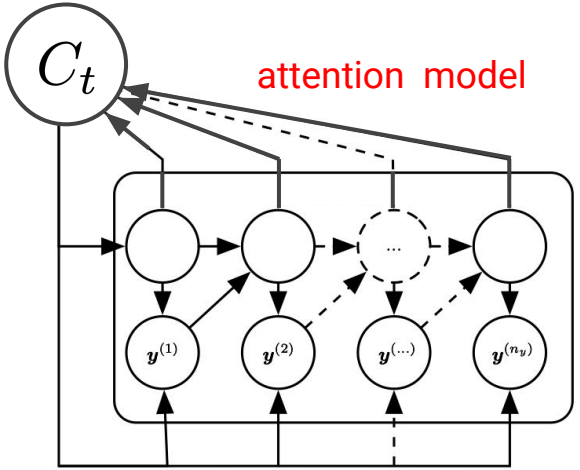
I

Inception CNN

Image encoding

φ

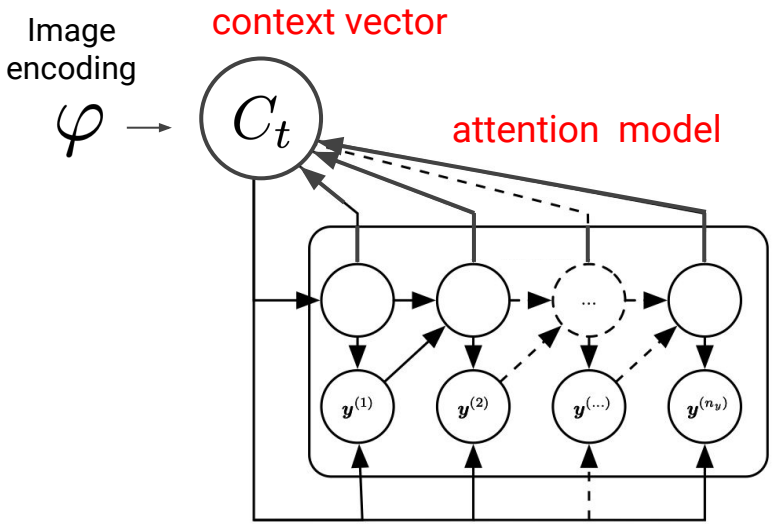
context vector



Attention model

image features at location j

$$e_{tj} = v_a^T \tanh(W_a \mathbf{h}^{(t-1)} + U_a \varphi_j)$$

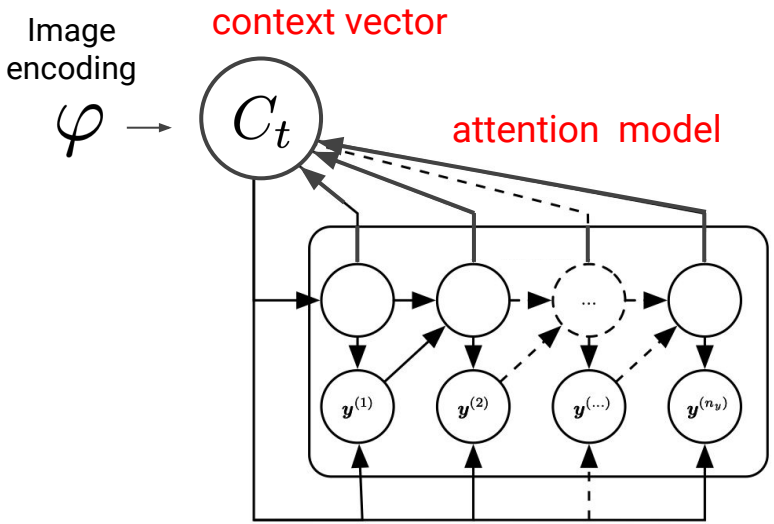


Attention model

image features at location j

$$e_{tj} = v_a^T \tanh(W_a \mathbf{h}^{(t-1)} + U_a \varphi_j)$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_k \exp(e_{tk})}$$



Attention model

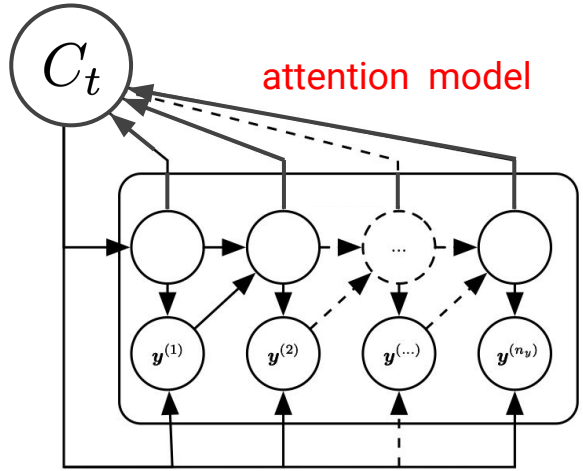
image features at location j

$$e_{tj} = v_a^T \tanh(W_a \mathbf{h}^{(t-1)} + U_a \varphi_j)$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_k \exp(e_{tk})}$$

$$\mathbf{C}_t = \sum_j \alpha_{tj} \varphi_j$$

Image encoding φ →



Bahdanau et al, Neural Machine Translation by Jointly Learning to Align and Translate, ICLR'15

Recurrent neural networks

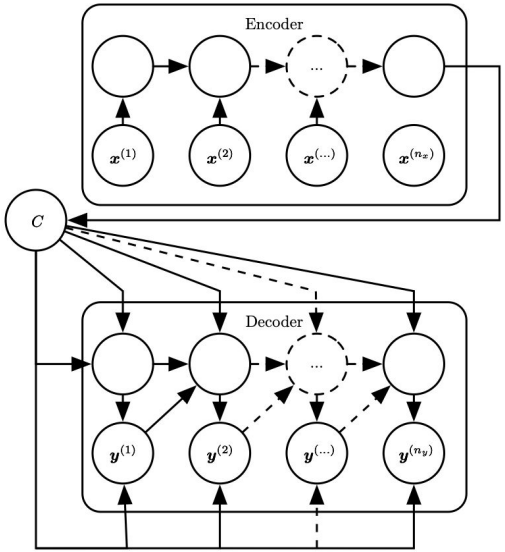
Image captioning model from:

https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/text/image_captioning.ipynb

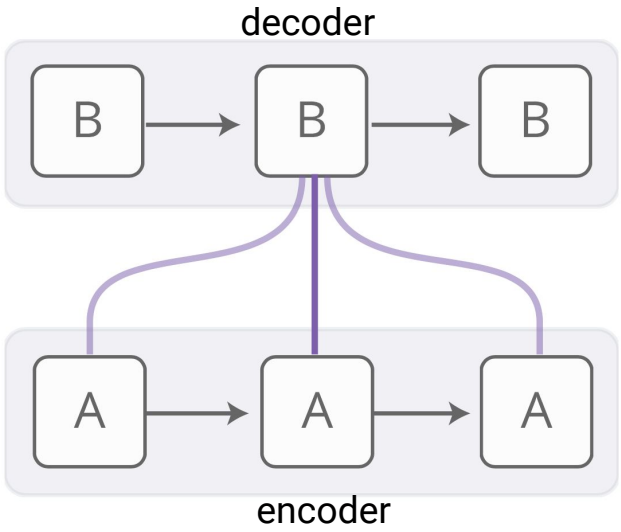
```
class RNN_Decoder(tf.keras.Model):
  def __init__(self, embedding_dim, units, vocab_size):
    super(RNN_Decoder, self).__init__()
    self.units = units

    self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
    self.gru = tf.keras.layers.GRU(self.units,
                                   return_sequences=True,
                                   return_state=True,
                                   recurrent_initializer='glorot_uniform')
    self.fc1 = tf.keras.layers.Dense(self.units)
    self.fc2 = tf.keras.layers.Dense(vocab_size)

    self.attention = BahdanauAttention(self.units)
```

Bahdanau et al, Neural Machine Translation by Jointly Learning to Align and Translate, ICLR'15

Machine translation with attention



Machine translation as a sequence-to-sequence model



Machine translation with attention

Images from https://distill.pub and https://www.deeplearningbook.org

Machine translation with attention

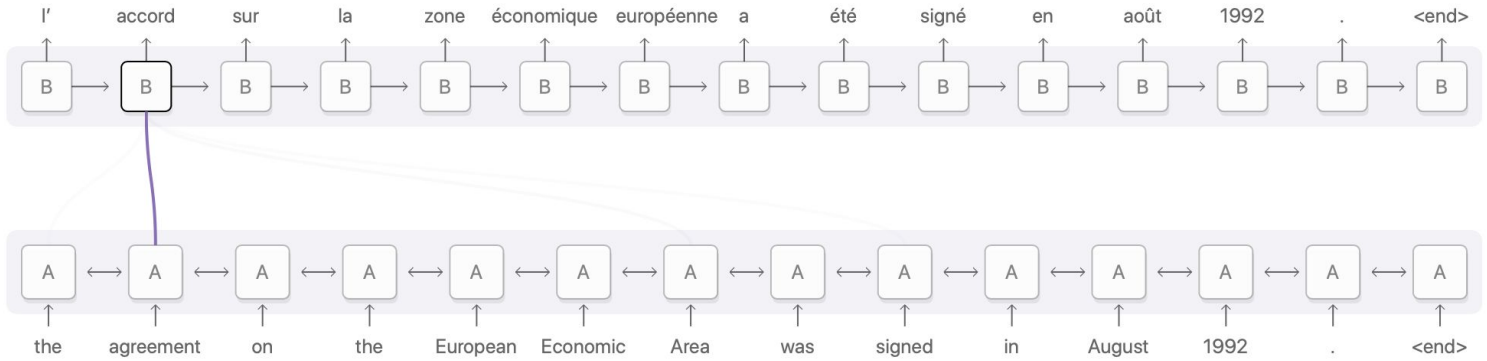


Diagram derived from Fig. 3 of Bahdanau, *et al.* 2014

https://distill.pub/2016/augmented-rnns/#attentional-interfaces

Machine translation with attention

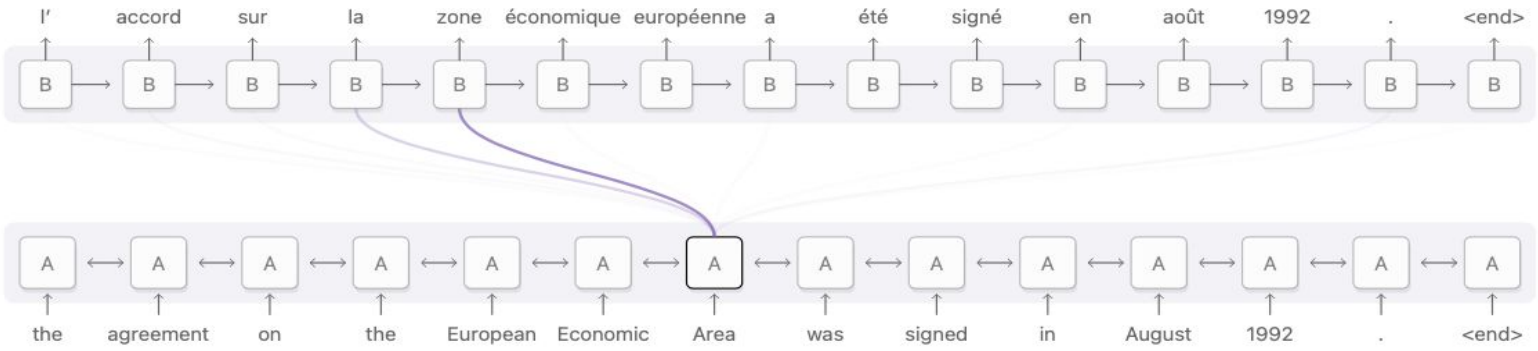


Diagram derived from Fig. 3 of Bahdanau, *et al.* 2014

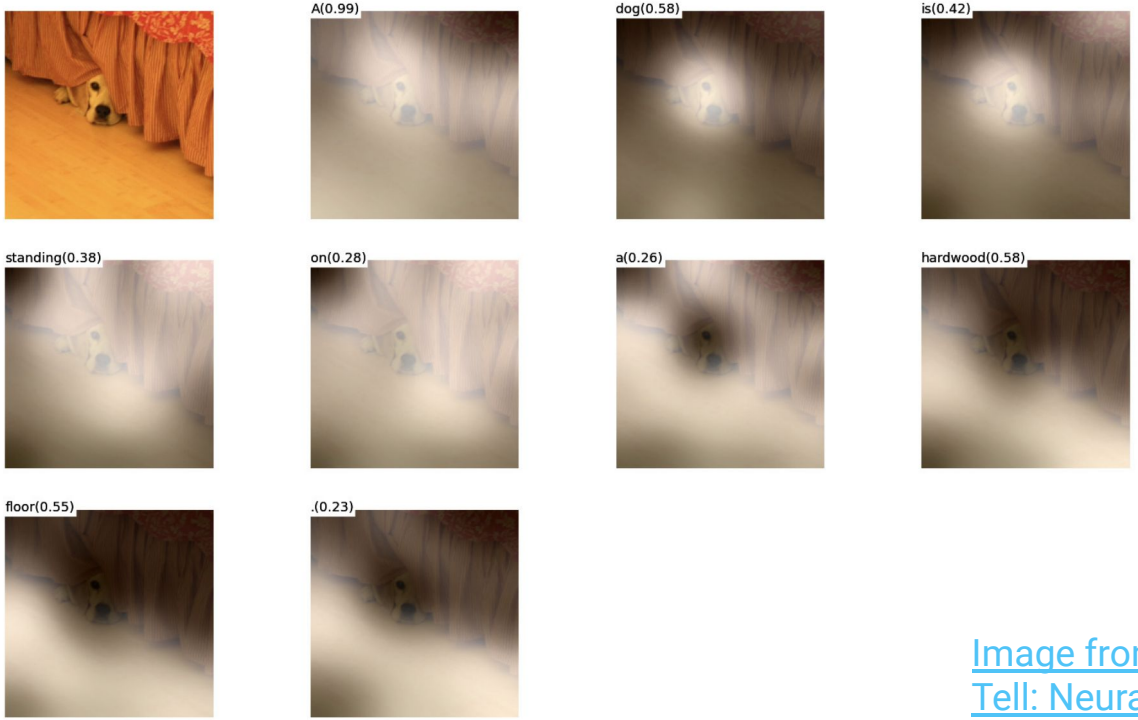
https://distill.pub/2016/augmented-rnns/#attentional-interfaces

Image captioning with attention

(b) A woman is throwing a frisbee in a park.

Image from “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”

Image captioning with attention



(b) A dog is standing on a hardwood floor.

[Image from “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”]

Image captioning with attention

(b) A woman holding a clock in her hand.

[Image from “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”]

Congratulations! We did it!

Image from the photo collection I

Neural Network (CNN)

Image encoding φ

Recurrent neural network (RNN)

Image description T_k = “dog next to person on the bike near street crossing”

Q Query: “person walking with a dog on the beach”

Define similarity function. Order images according to similarity to the query.

$$\text{sim}(Q, T_1) > \text{sim}(Q, T_2)$$

Thank you for your attention!

Take a quiz!