ECG782: MULTIDIMENSIONAL DIGITAL SIGNAL PROCESSING

ATTENTION AND TRANSFORMERS

http://www.ee.unlv.edu/~b1morris/ecg782

OUTLINE

- Overview of State of CV/AI
- Recurrent Neural Networks
- Attention
- Transformers

STATE OF COMPUTER VISION/AI

- Reached the point of real solutions
  - Images in → understanding out
- Killer apps can rely on AI/CV solutions
- Advances come quickly

- No dream of just using vision for DARPA Grand or Urban Challenges (2007)

Sep 24, 2014

https://xkcd.com/1425

DRIVERS FOR INNOVATION

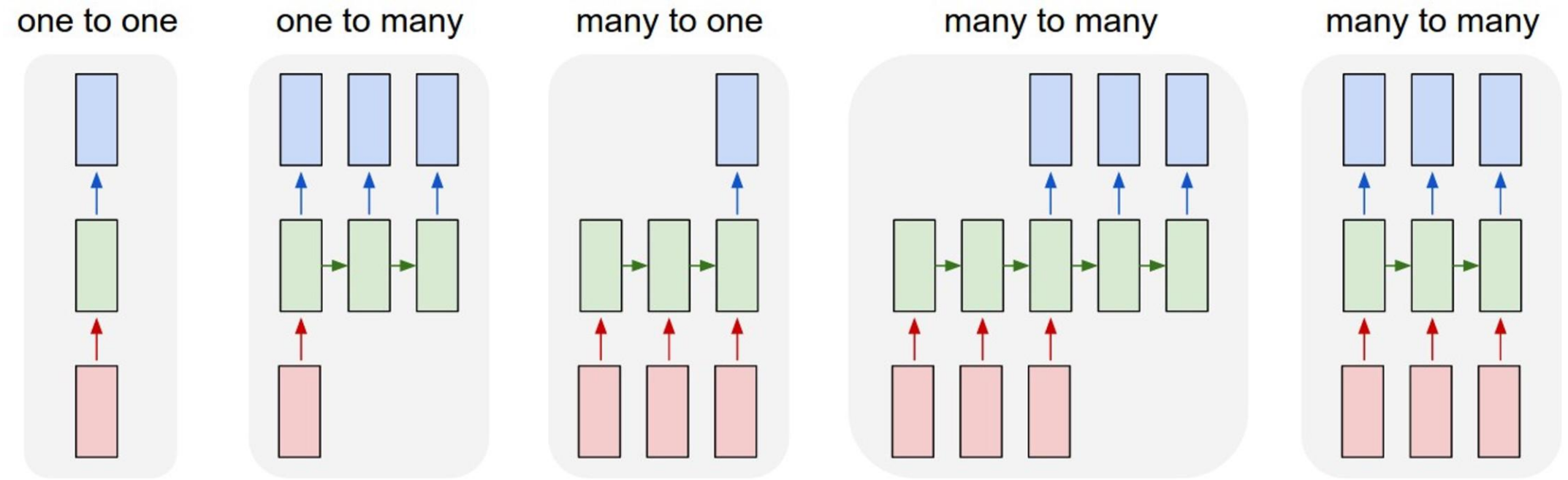
- Open access culture in CV community
- Strong review process
  - Review process taken very seriously, associated pride → high quality reviews, thoughtful decision making process, rebuttal "conversation"
  - Preprint servers (arXiv) rapidly advertise
- Large benchmark datasets
  - ImageNet changed the world – direct comparison for algorithms, democratization of research (not limited just to ivory towers)
- High quality code repositories
  - Code expected to be viewed by others and used. Sure, self-serving because you get more citations/references when you make code available but again makes comparison possible/fair → leads to innovations as parallel approaches to build off of others
- Abundance of frameworks and tools to make vision and AI computationally feasible

OUTLINE

- Overview of State of CV/AI
- Recurrent Neural Networks
- Attention
- Transformers

RNN BACKGROUND

- Up to now, we have assumed fixed-size inputs and outputs
- How can we handle more complicated scenarios (such as sequential processing)?



Vanilla NN

Video Captioning
Image → seq of words

Action Prediction
Seq frames → action class

Video Captioning
Seq frames → seq of words

Video Classification
Seq frames → seq class

RNN MOTIVATION

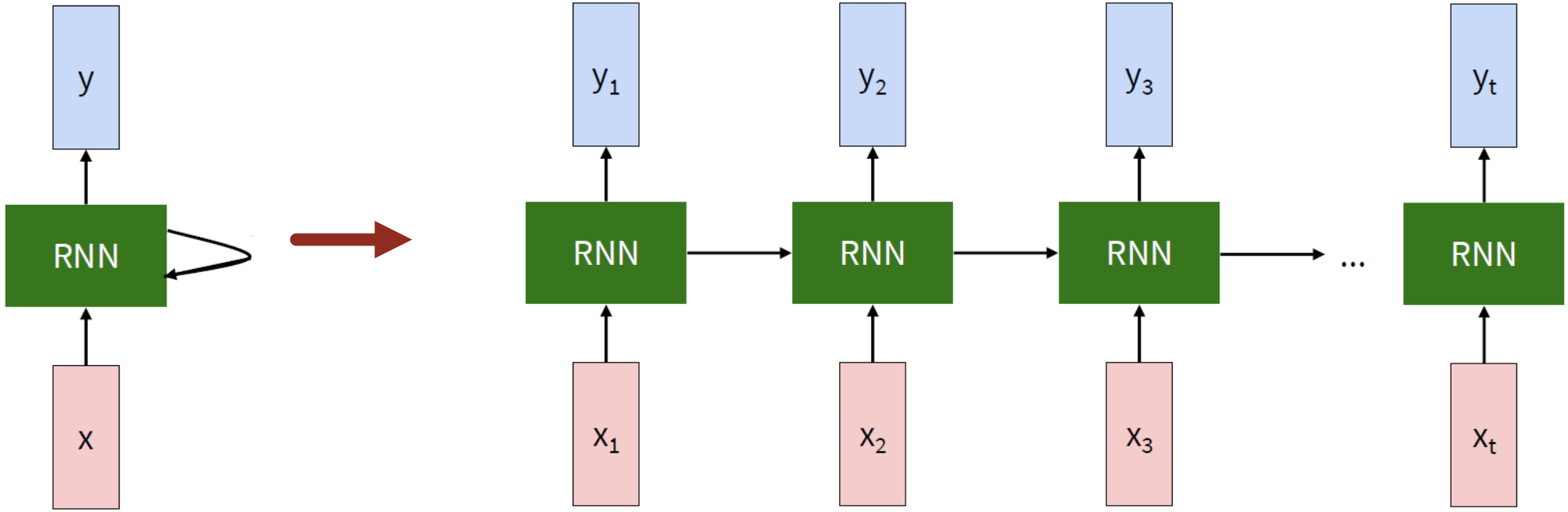
- These tasks require models that can:
  - Handle variable-length input and output sequences
  - Preserve temporal structure and order
  - Capture long-range dependencies
- Considerations:
  - Long-range dependencies: how to learn which past inputs are relevant
  - Parallelizability: can model be parallelized across time steps
  - Compute/memory use: how does this scale with sequence length
  - Inductive bias: how well do model capture temporal/locality structure

RECURRENT NEURAL NETWORK (RNN)

- RNNs have "internal state" that is updated as the sequence is processed

- Unrolled RNN

Recurrence: internal state influences future output



RNN EQUATIONS

- Hidden state update

$$h_t = f_W(h_{t-1}, x_t)$$

  - New state
  - Input at time t
  - Old state
  - Function with parameters W

- Output decode

$$y_t = f_D(h_t)$$

  - Output
  - New state
  - Function with different parameters D

RNN SUMMARY

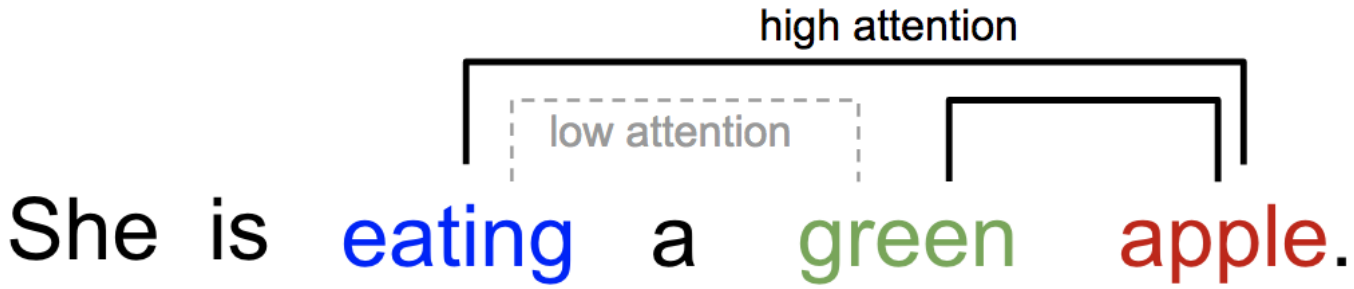
- Advantages
  - Can process inputs of any length
  - Each step can use information from all previous steps (in theory)
  - Model size is fixed, regardless of sequence length
  - Shared weights across time → enforces temporal consistency
- Disadvantages
  - Slow training due to sequential/recurrent computation
  - Hard to capture long-term dependencies
  - Vanishing/exploding gradients
    - LSTM/GRU variants use gating mechanism to control flow of info over time

OUTLINE

- Overview of State of CV/AI
- Recurrent Neural Networks
- Attention
- Transformers

ATTENTION

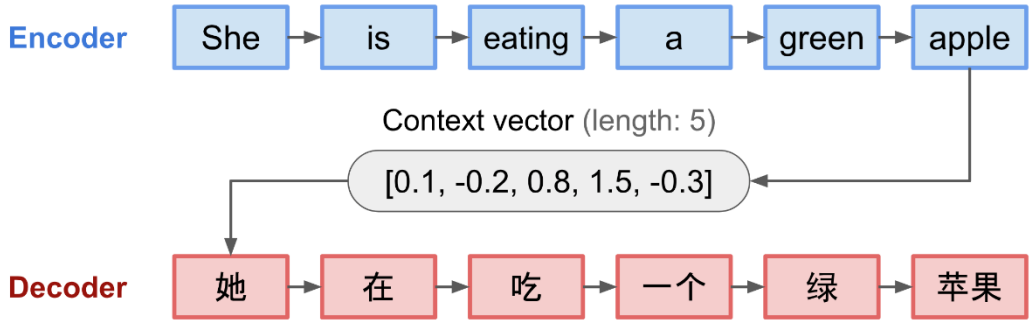
- What we should attend to or focus on to answer a question (start with NLP)



- Seeing word eating leads to expectation of a food soon in the sentence

SEQ2SEQ LIMITATIONS

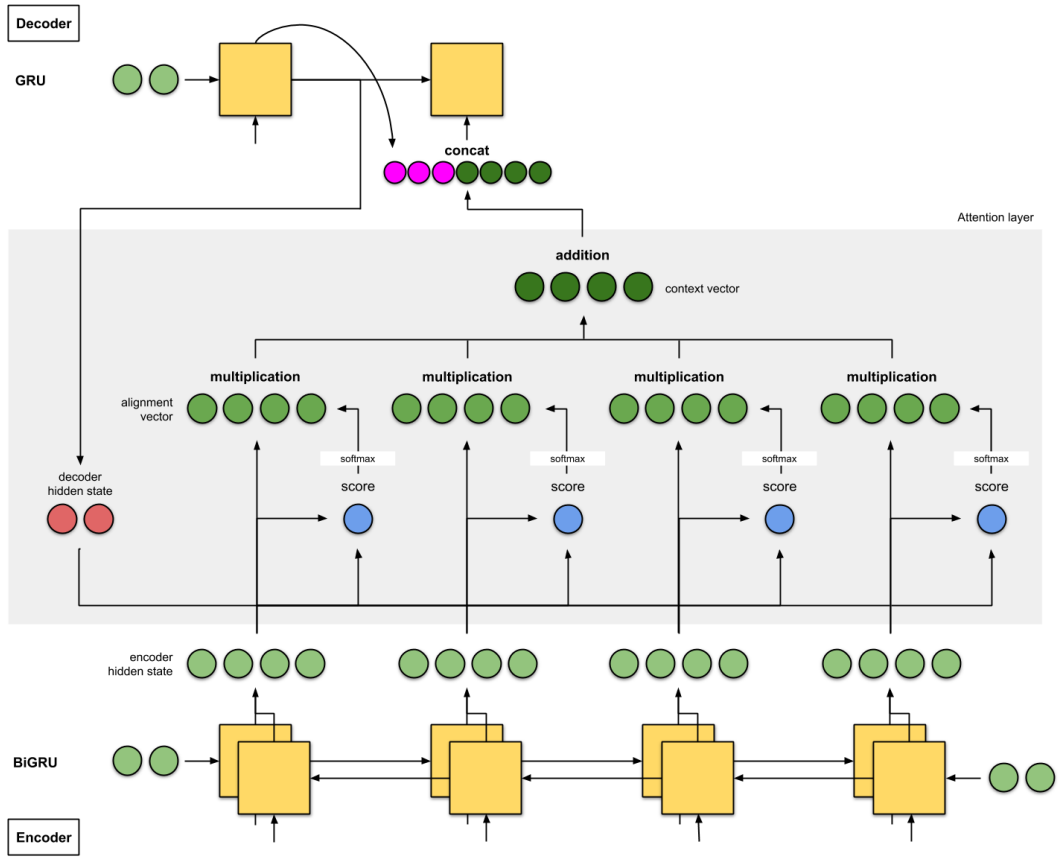
- Use of RNN to encode a sentence (e.g. for translation)



- Full sentence summarized/embed into fixed length context vector
- Read input word by word, then generate output word by word
  - Must do full translation based on summary of everything read
- Tends to forget for longer sequences
  - Try to do a translation after reading a long paragraph

ATTENTION LAYER I

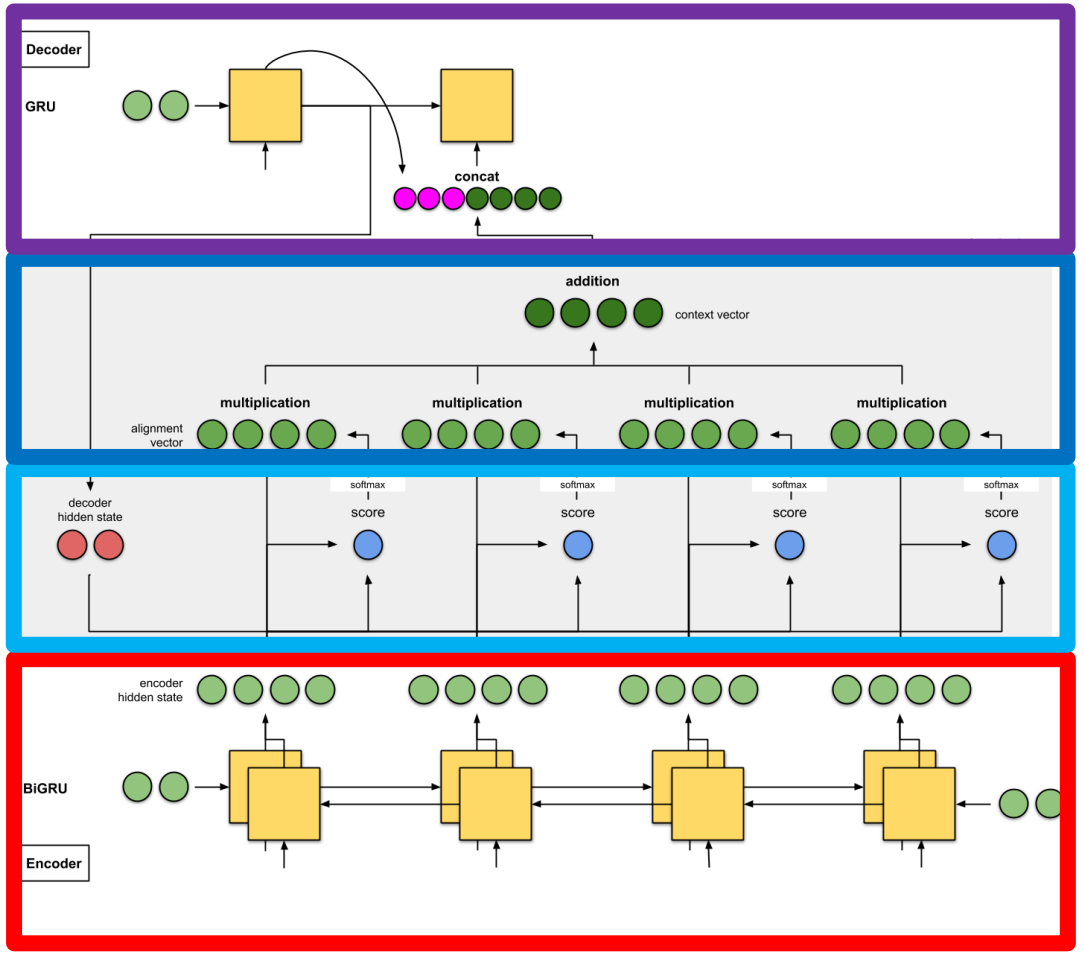
- Incorporate encodings from each time step (word)
  - Focus on useful parts of input (alignment)
- Learn alignment from data



Bahdanau et. al. 2015

ATTENTION LAYER II

- Encoder to embed input and provide hidden state/representation
- Attention weights computed at each time step
  - Score between decoder hidden state and all input hidden states
- Context vector to combine inputs
- Decoder makes prediction on weighted sum of inputs (context)



Bahdanau et. al. 2015

ATTENTION LAYER III

- Length n input and length m output

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]$$

$$\mathbf{y} = [y_1, y_2, \ldots, y_m]$$

- Input encodings

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i^\top; \overleftarrow{\boldsymbol{h}}_i^\top]^\top, i = 1, \ldots, n$$

- Output encoding $(t = 1, \ldots, m)$

$$\boldsymbol{s}_t = f(\boldsymbol{s}_{t-1}, y_{t-1}, \mathbf{c}_t)$$

- Context vector for output y_t

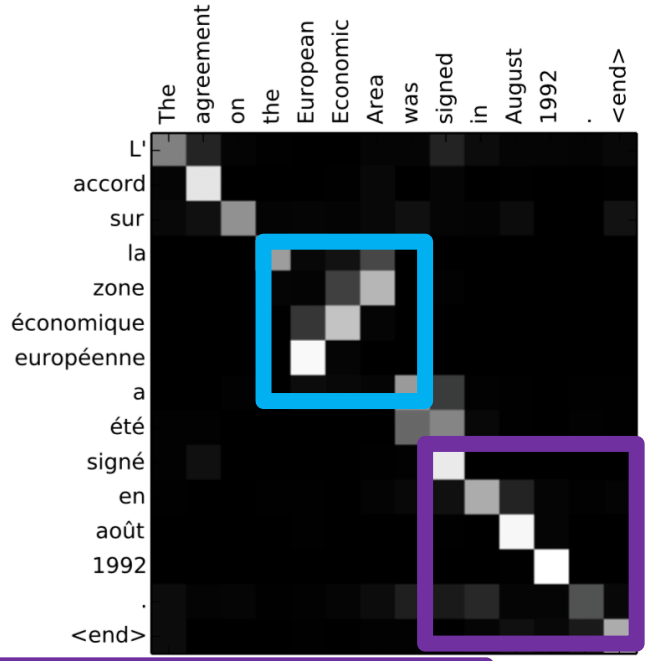
$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i$$

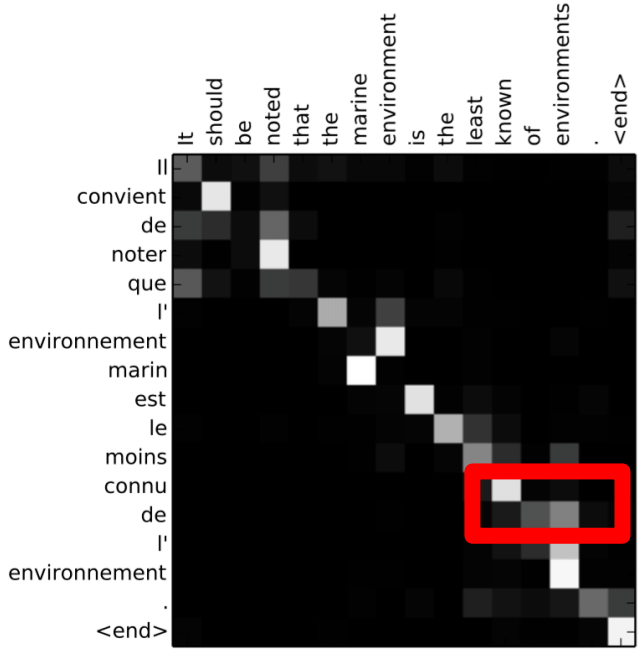
$$\alpha_{t,i} = \text{align}(y_t, x_i)$$

$$= \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))}$$

- $\alpha_{t,i}$ - how well words y_t and x_i are aligned (contribution)
- Score – function (e.g. dot product) for similarity
  - Other functions can be learned

VISUALIZATION OF ATTENTION





- One-to-one mapping
- Ordering of words matters
- Output can depend on more than one input word

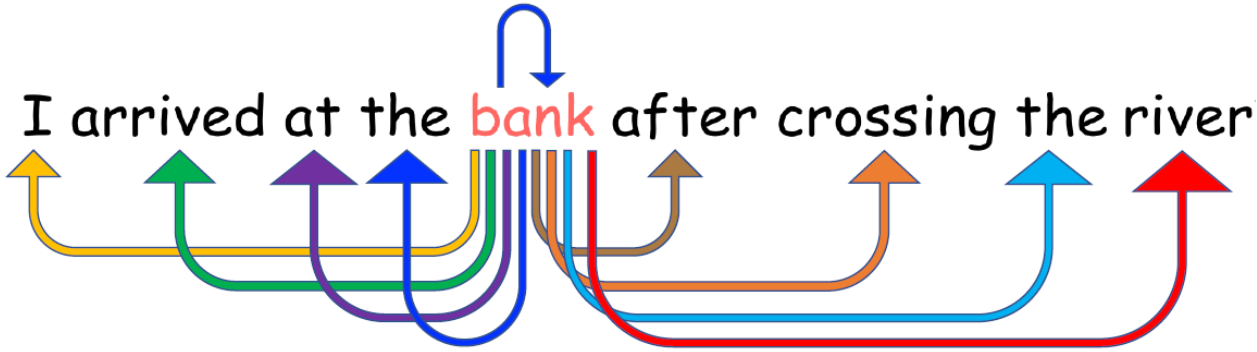
OUTLINE

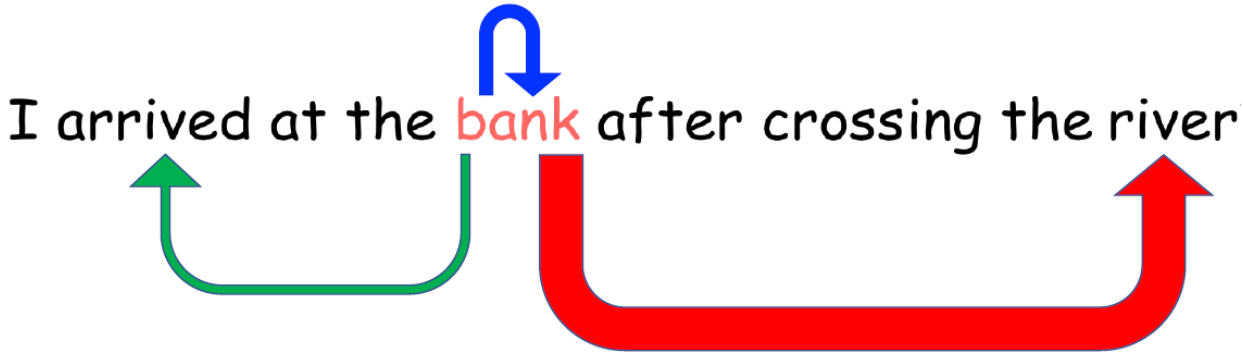
- Overview of State of CV/AI
- Recurrent Neural Networks
- Attention
- Transformers

TRANSFORMER OVERVIEW

- Avoid recurrence/convolution (attention is all you need)
  - Faster to train
  - Constant number of steps for processing (do not wait until end of sentence)
- Attention for relationship within sentence
- Meaning of bank doesn't have to wait until seeing river

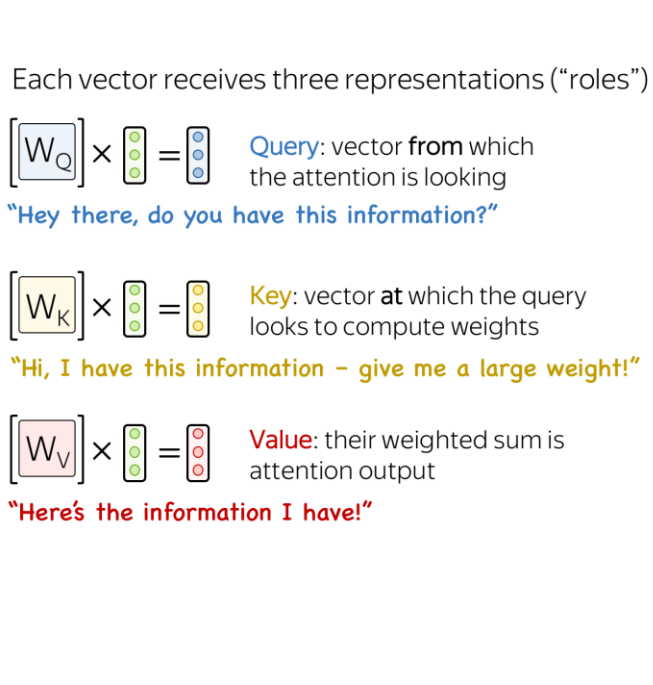
- Key idea: self attention
  - Compare "tokens"

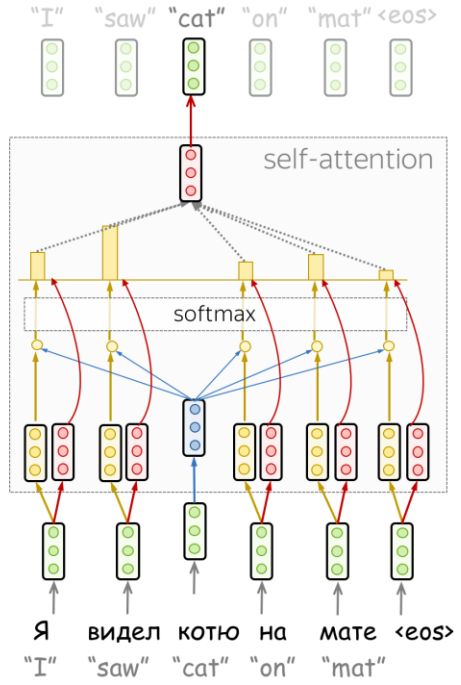




TRANSFORMER – QUERY/KEY/VALUE

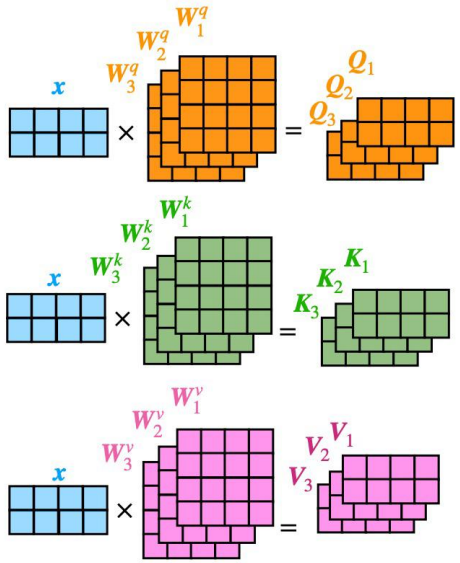
- Three learnable embeddings for each token
- Query – asking for information to better understand itself
- Key – response to query request → used to compute attention weights
- Value – give information → used to compute context



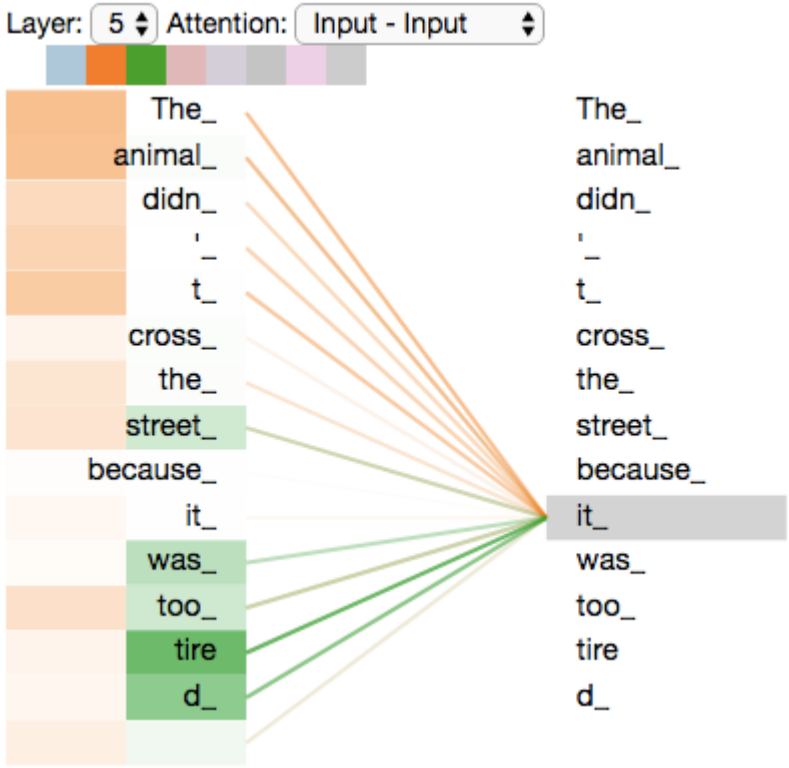


TRANSFORMER MULTI-HEAD ATTENTION

- Enable each token to relate to other tokens in multiple ways
- Use multiple self-attention mechanisms with their own query, key, and value matrices



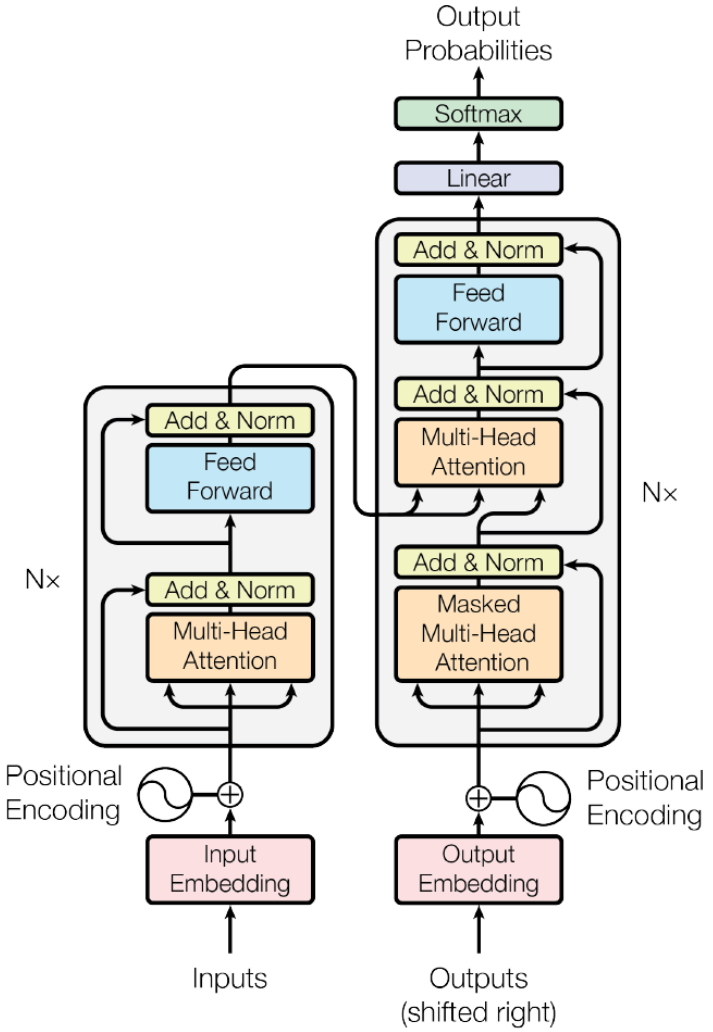
- Orange head – it = animal



- Green head – it = tired

TRANSFORMER MODEL

- Multi-head attention (self attention) in encoder (left) and decoder (right)
- Residual connections and layer normalization – gradient propagation and convergence
- Feed forward – process info
- Positional encoding – explicitly add position infomation

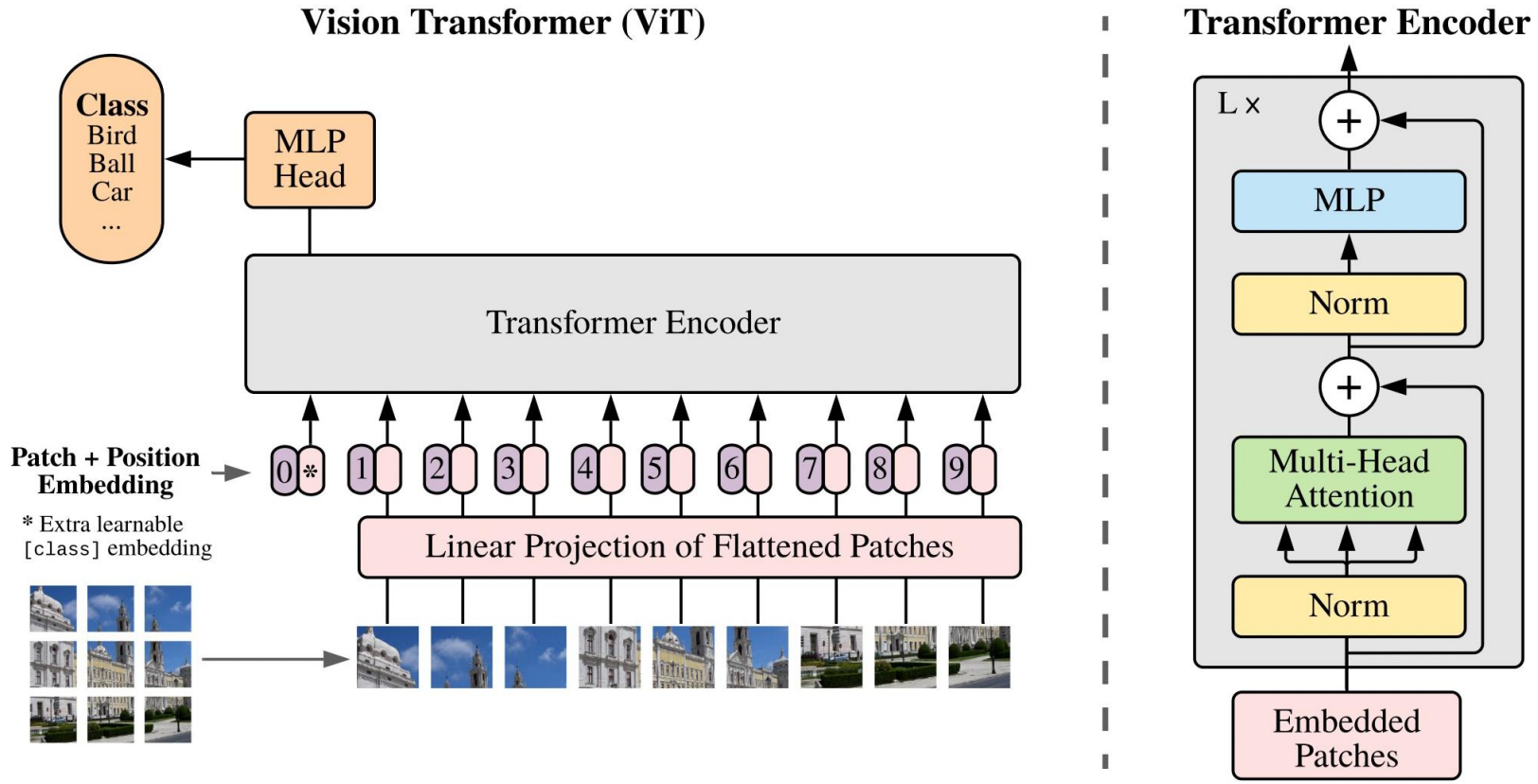


Vaswai et al. NeurIPS 2017

VISUAL ATTENTION

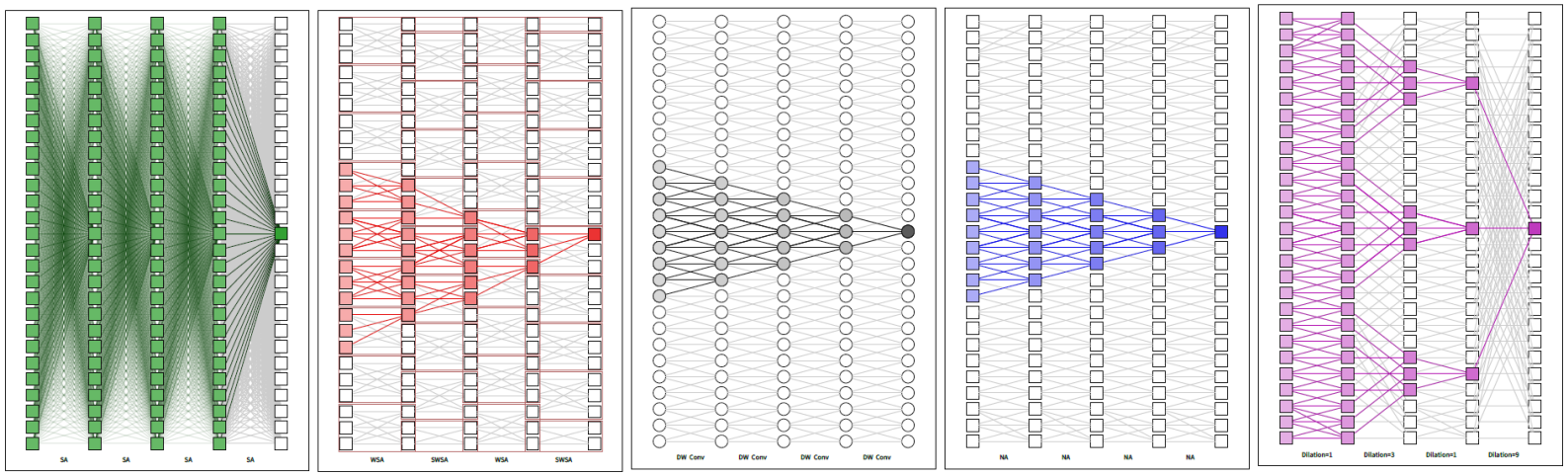
VISION TRANSFORMER (VIT)

- How can Transformer success be used in vision tasks?
  - E.g. words = image patches



Dosovitskiy et al. ICLR 2021

POPULAR VARIANTS



ViT.
Complexity: $\mathcal{O}(n^2 d)$,
RF $= n$.

Swin.
Complexity: $\mathcal{O}(ndk)$,
RF $= \ell k$.

ConvNeXt.
Complexity: $\mathcal{O}(ndk)$,
RF $= \ell(k-1)+1$.

NAT.
Complexity: $\mathcal{O}(ndk)$,
RF $= \ell(k-1)+1$.

DiNAT.
Complexity: $\mathcal{O}(ndk)$,
RF $\in [\ell(k-1)+1, k^{\ell}]$.

- Handle self-attention cost over image pixels scale
- Shifted window (Swin) – hierarchical feature maps for scale, local attention
- Dilated Neighborhood Attention (DiNAT) – dilated computation for larger receptive field → global context

TRANSFORMERS AND COMPUTER VISION

- Top performer is many vision benchmarks (e.g. ImageNet classification)
- Everyone is converting their models to Transformer variants
- CVPR2022 papers
  - Zhai, Scaling Vision Transformer
  - Chavan, Vision Transformer Slimming: Multi-Dimension Searching in Continuous Optimization Space

REFERENCES

- Attention is All You Need, Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomes, Kaizer, Polosukhin, 2017
- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, Dosovitskiy et al., 2021