Interpreting neural network models

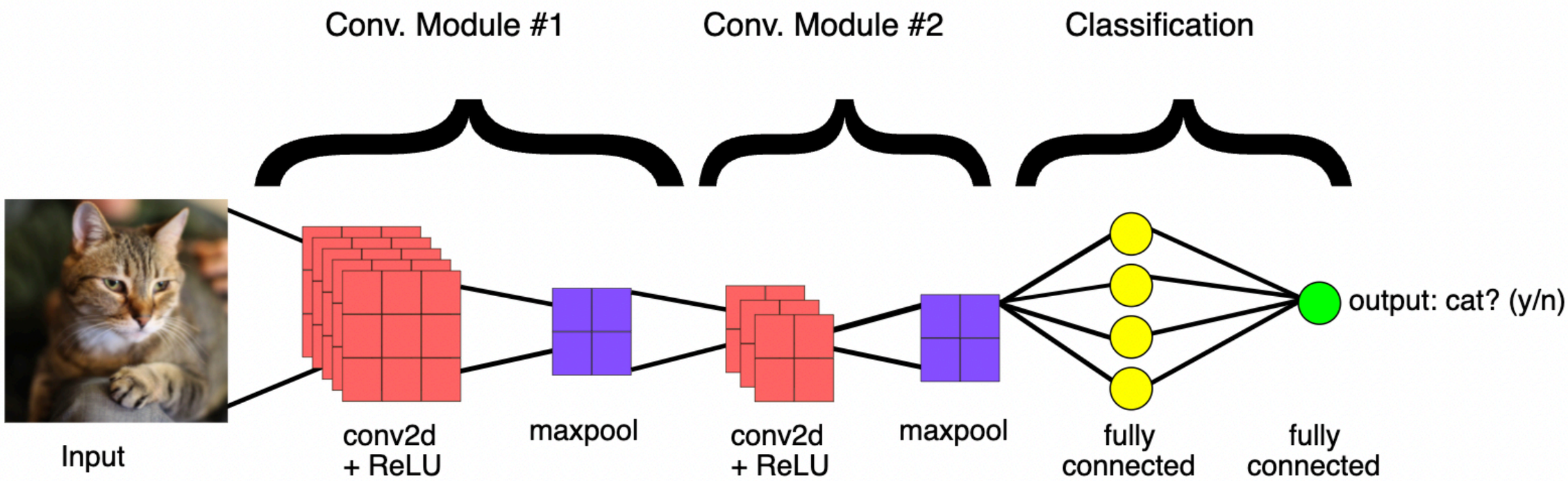
Interpretability is the degree to which a human can understand the cause of a decision.

Why interpret?

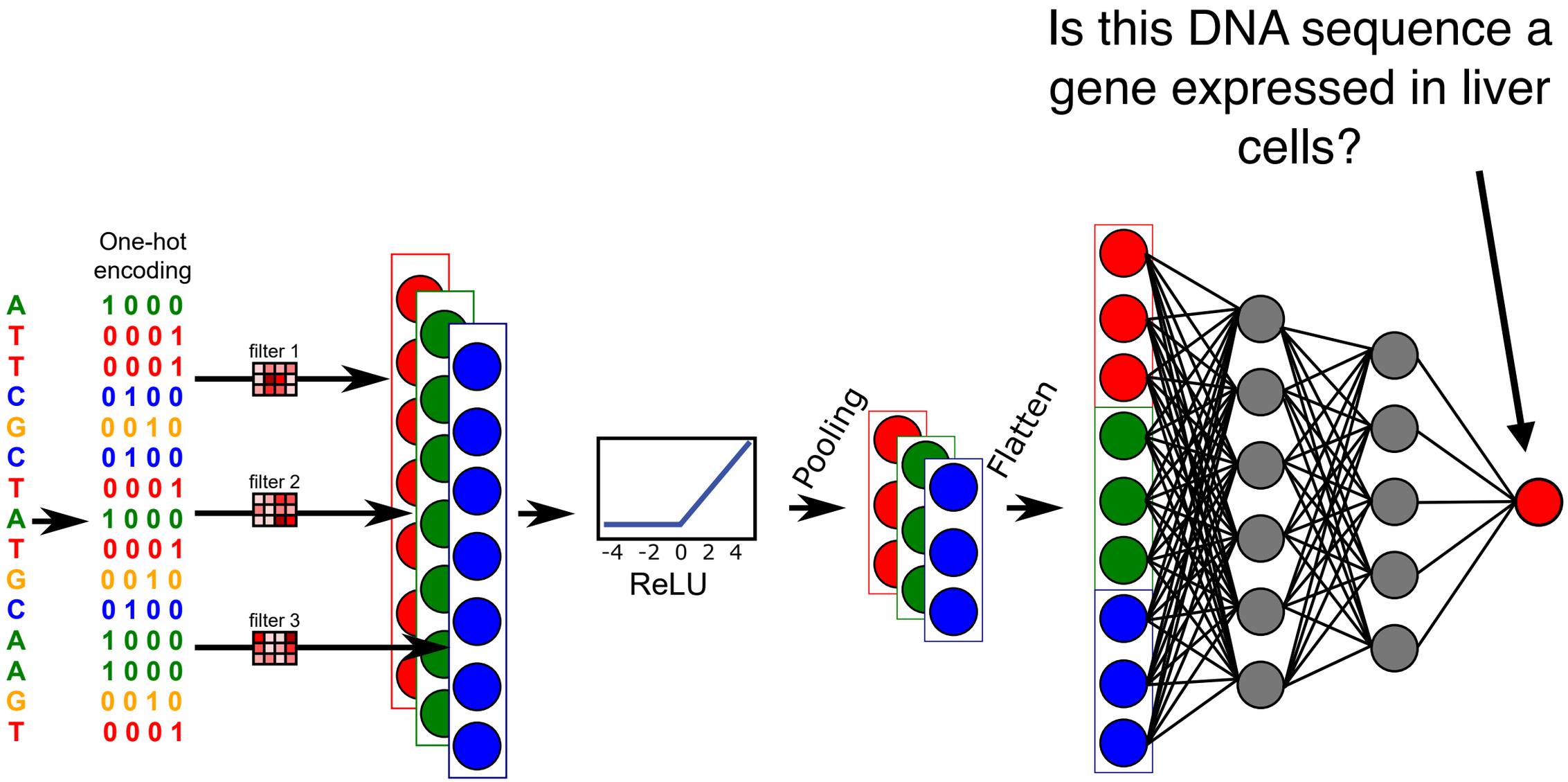
Local interpretation: Explain a model's prediction on a single input example.

Global interpretation: Explain how a model makes predictions across all examples.

Example model: Predicting contents of an image



Example model: Predicting gene expression from DNA sequence



Novakovsky et al, In Press

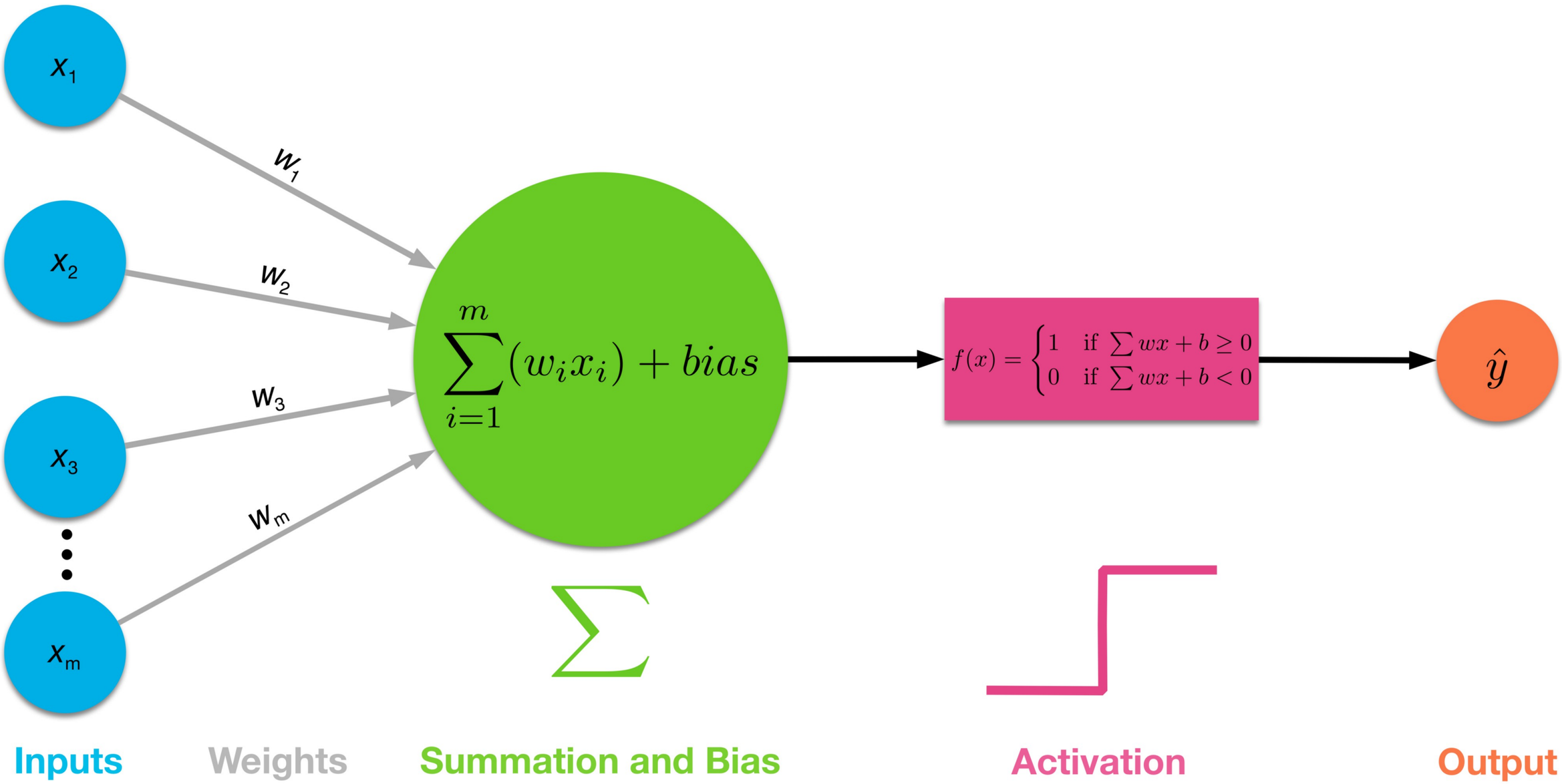
Outline

- Model-based interpretation
- Saliency map
- Understanding interactions

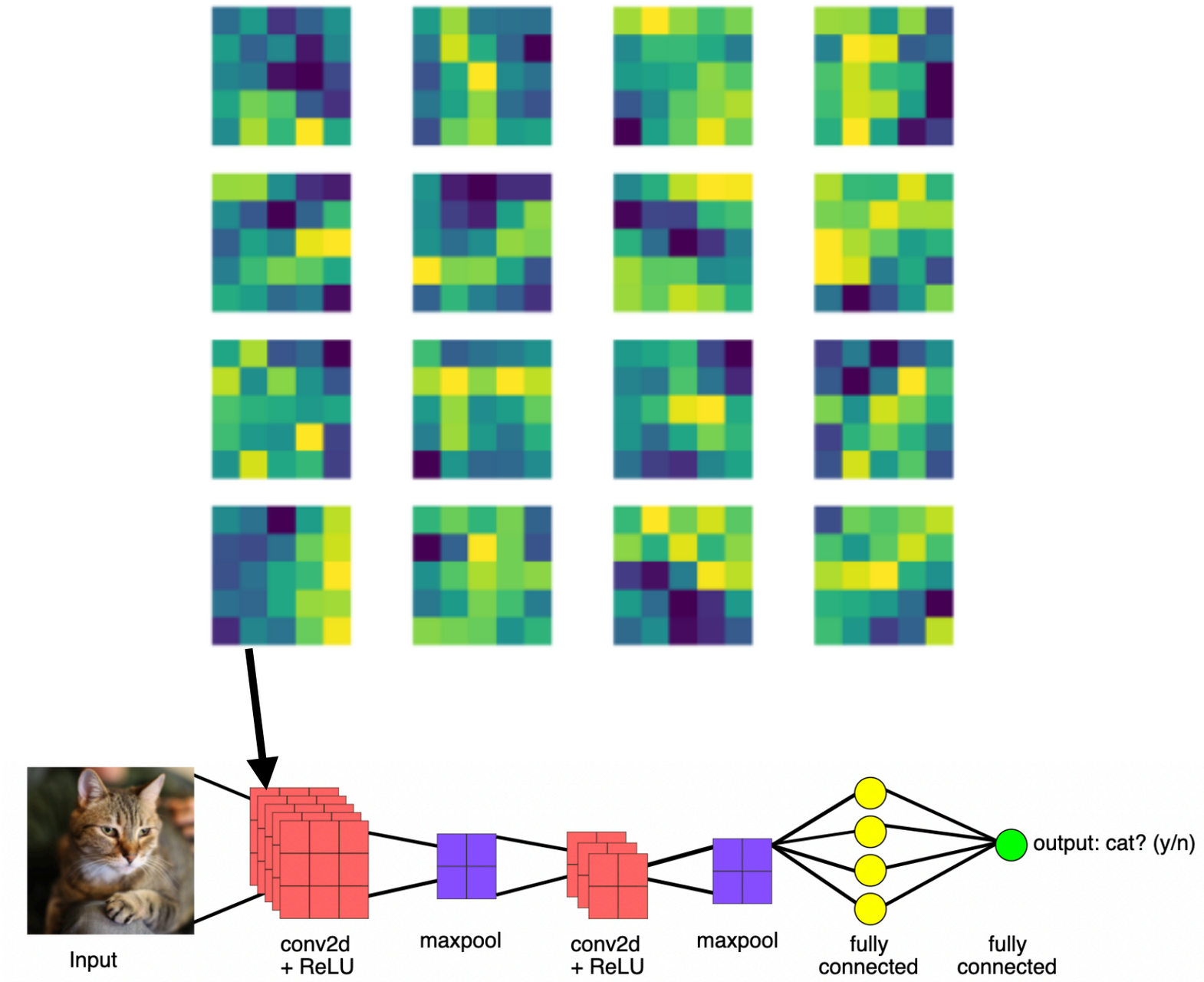
Outline

- Model-based interpretation
- Saliency map
- Understanding interactions

Understanding simple neural networks

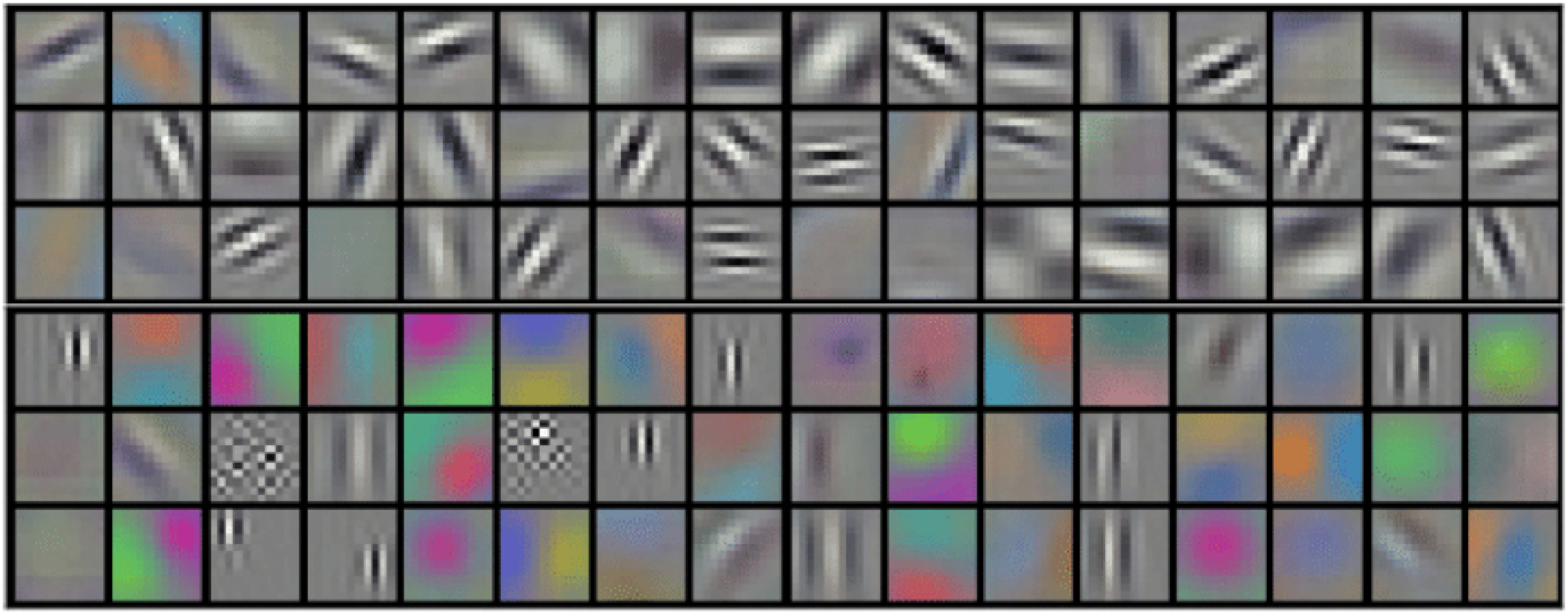


Inspecting individual nodes



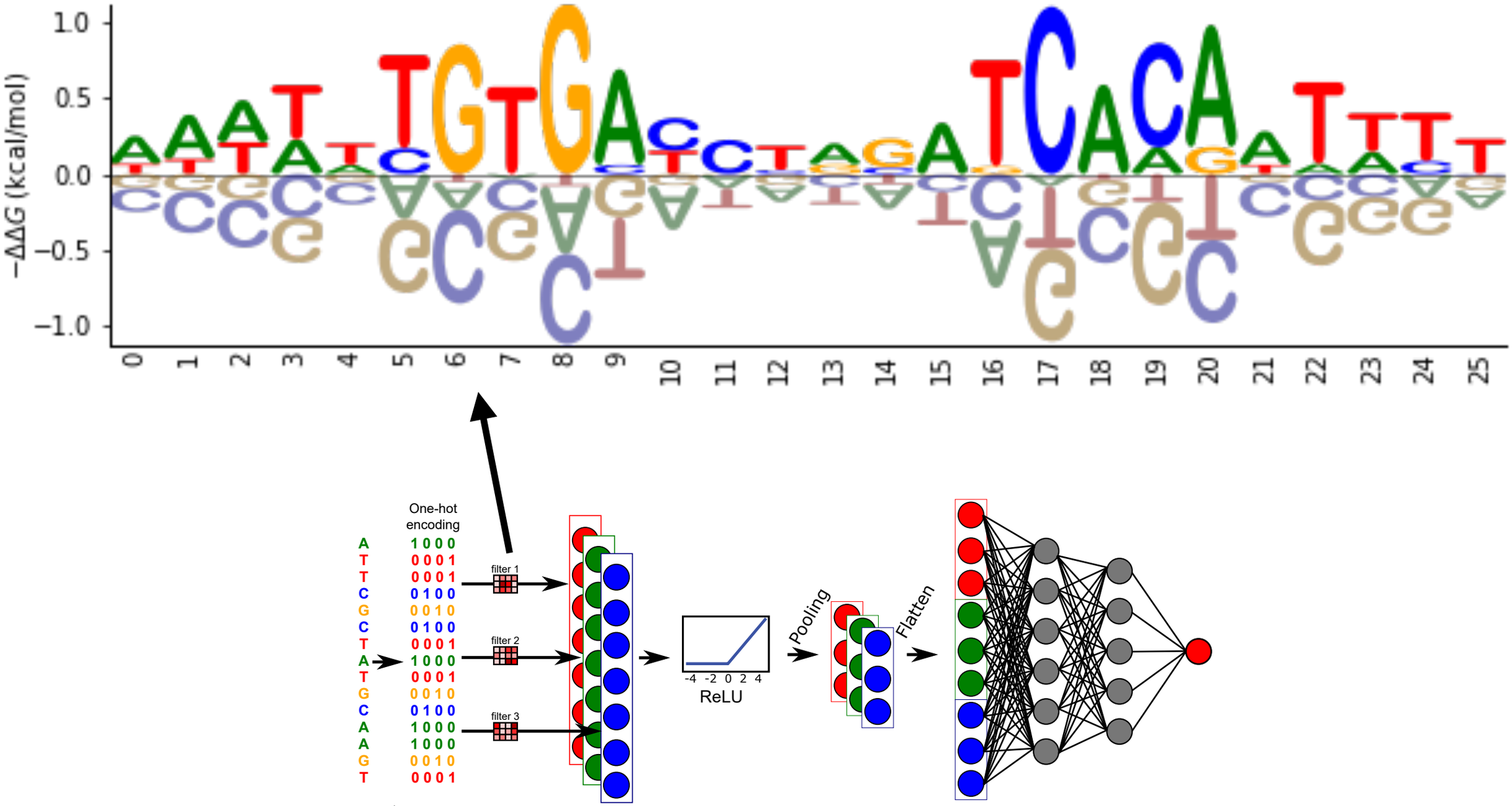
https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e

Inspecting individual nodes



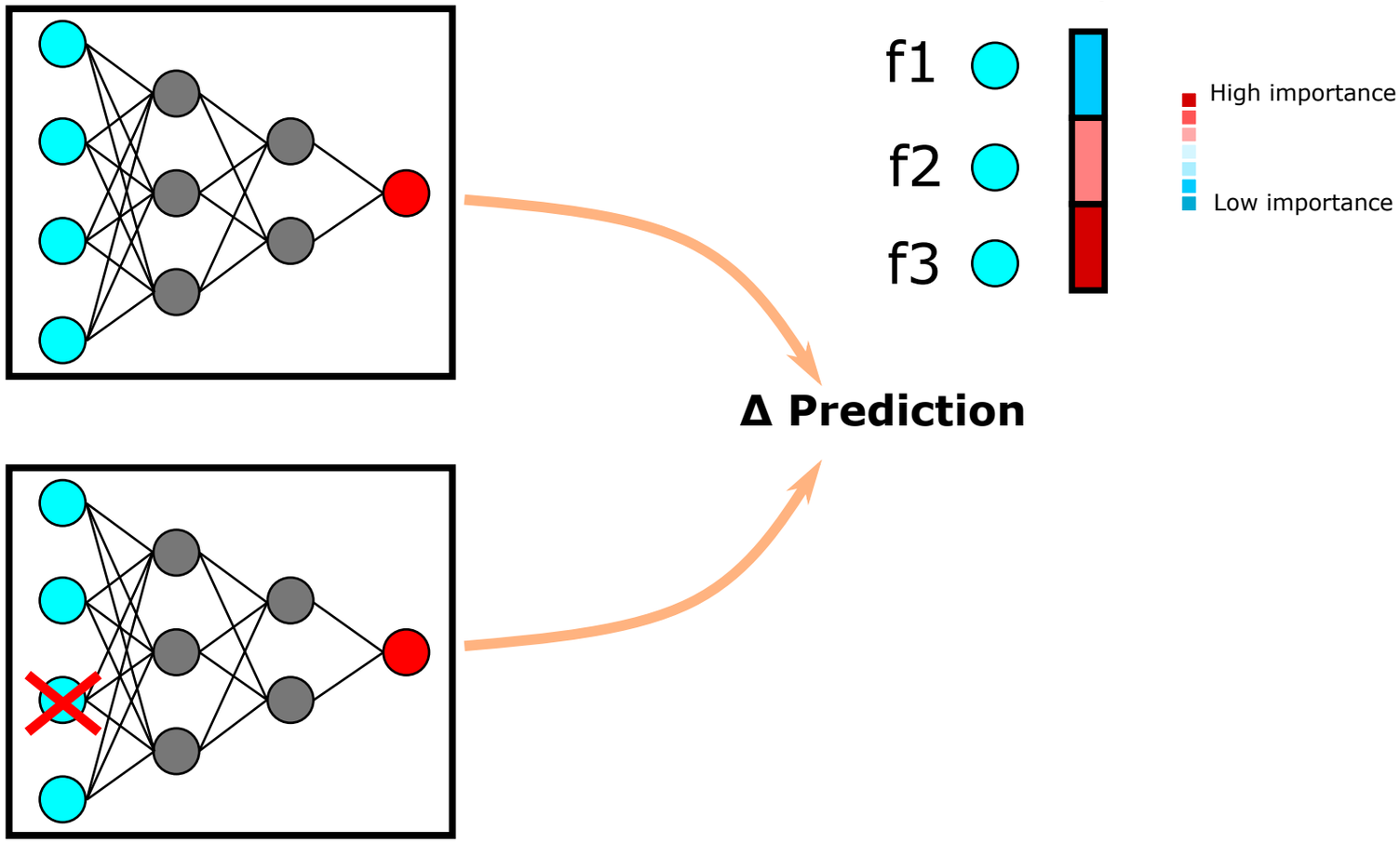
DOI: 10.1109/MAPR.2018.8337517

Inspecting individual nodes



DOI: 10.1109/MAPR.2018.8337517

What impact does each node have on the final prediction?



Novakovsky et al, In Press

Visualizing a hidden node

Idea: Use gradient descent to find the input that maximizes the activation of a given node.

https://christophm.github.io/interpretable-ml-book/

Visualizing a hidden node

Idea: Use gradient descent to find the input that maximizes the activation of a given node.

Maximize

https://christophm.github.io/interpretable-ml-book/

Visualizing a hidden node

Idea: Use gradient descent to find the input that maximizes the activation of a given node.

Maximize Minimize

https://christophm.github.io/interpretable-ml-book/

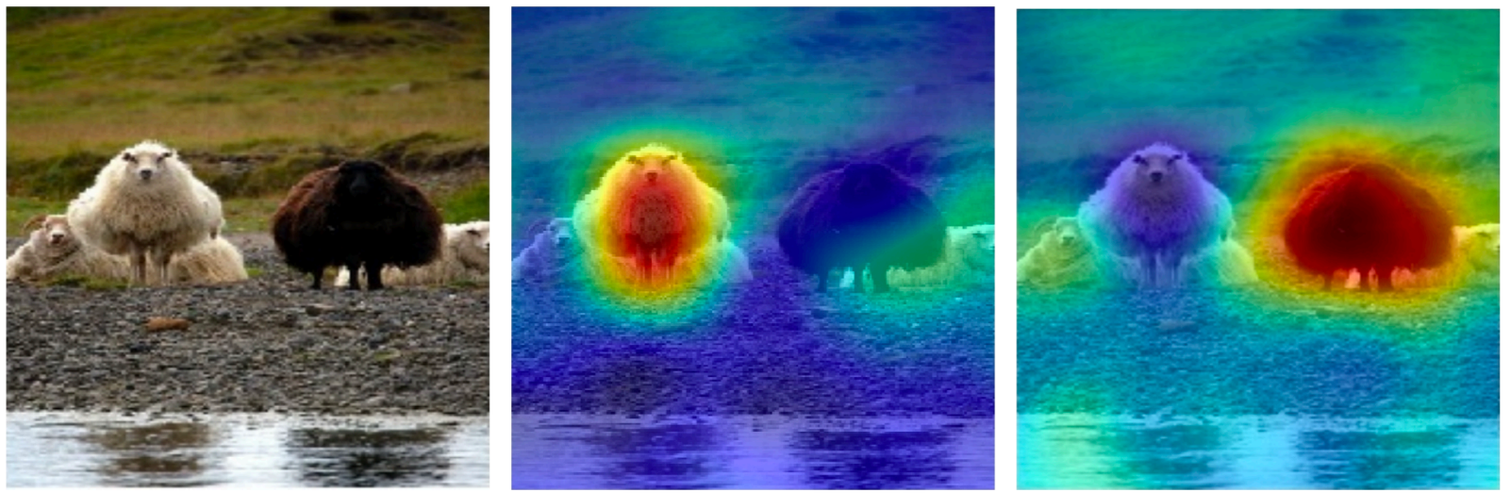
Questions about model-based interpretation

Outline

- Model-based interpretation
- Saliency map
- Understanding interactions

Saliency map

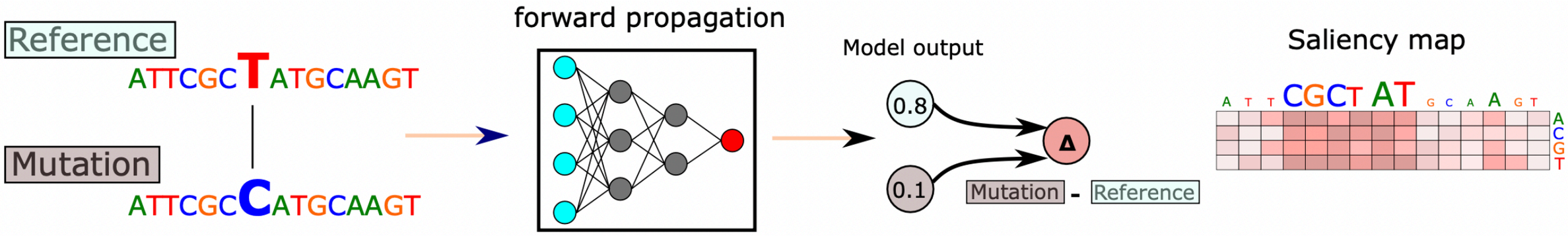
Saliency map: An estimate of the importance of each input feature.



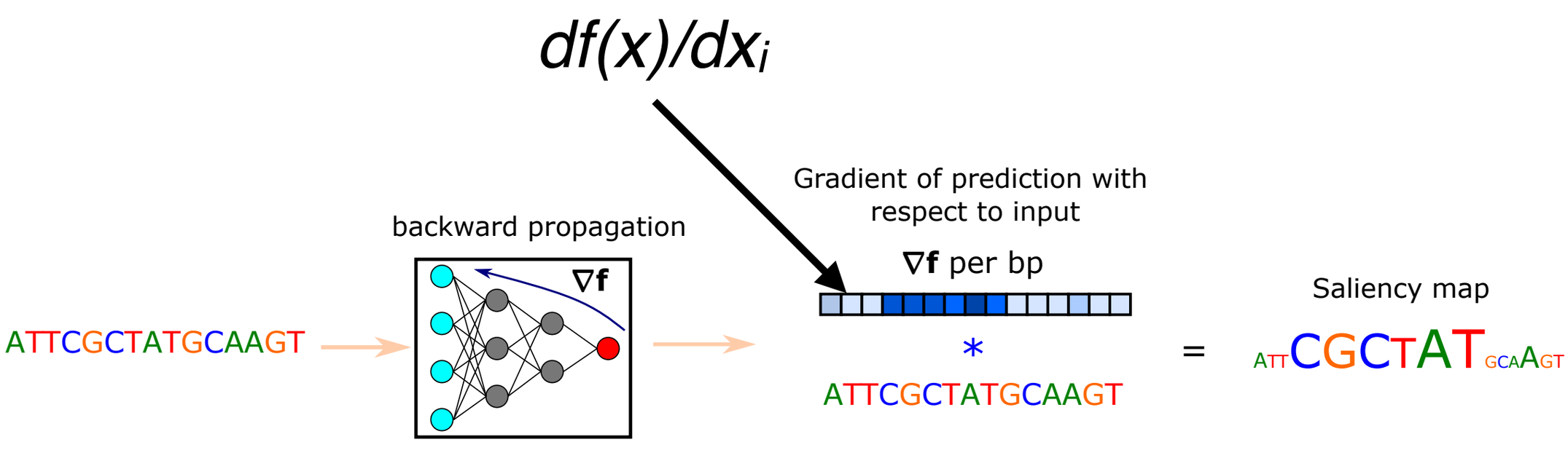
(a) Sheep - 26%, Cow - 17% (b) Importance map of '*sheep*' (c) Importance map of '*cow*'

(d) Bird - 100%, Person - 39% (e) Importance map of '*bird*' (f) Importance map of '*person*'

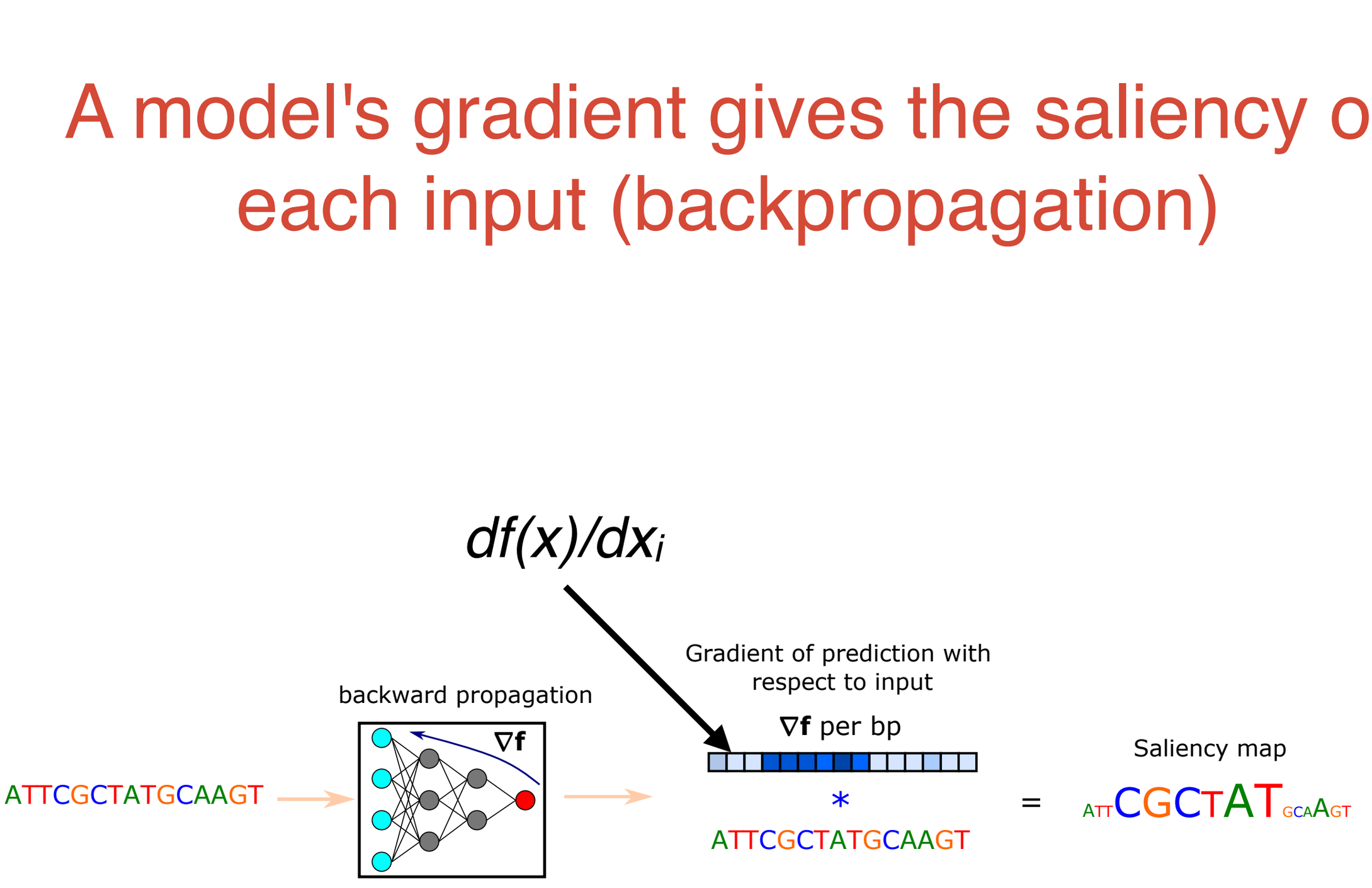
Interpretation through perturbation



A model's gradient gives the saliency of each input (backpropagation)

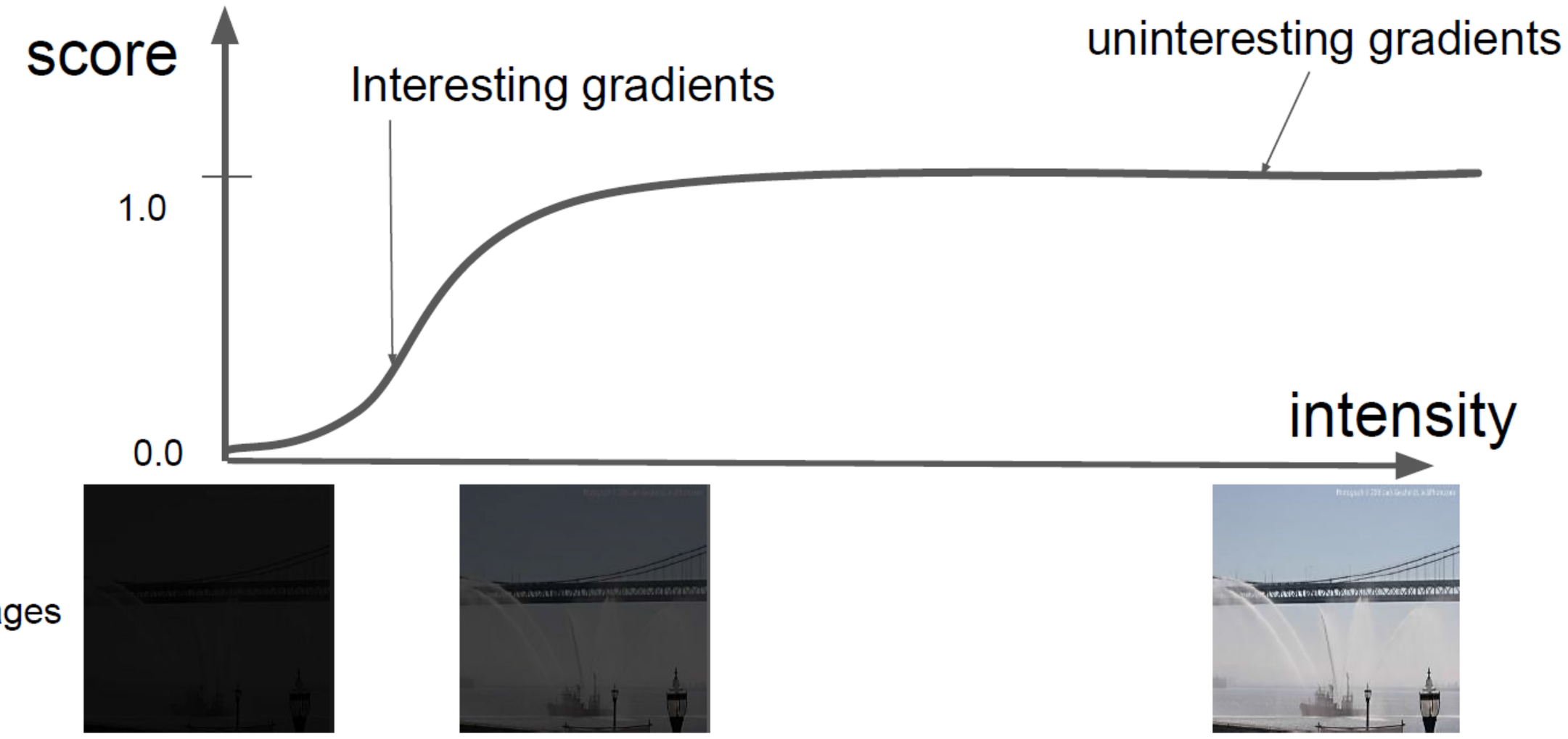


A model's gradient gives the saliency of each input (backpropagation)



Problem: Saturating nonlinearities.
There are many solutions: Integrated Gradients, DeepLIFT, DeepSHAP, ...

Integrated gradients

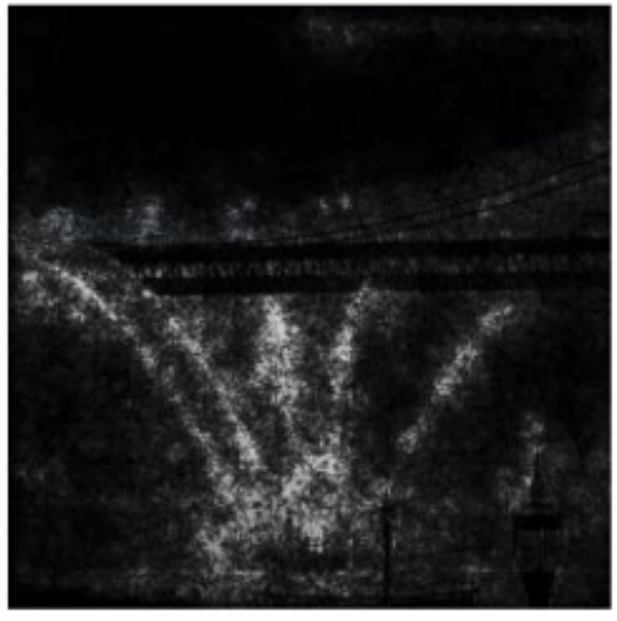


Integrated gradients

$$IG(input, base) ::= (input - base) * \int_{0\ -1} \nabla F(\alpha * input + (1-\alpha) * base)\ d\alpha$$

Original image

Integrated Gradients



DeepLIFT

$$x_i \times \frac{\partial Y}{\partial x_i}$$

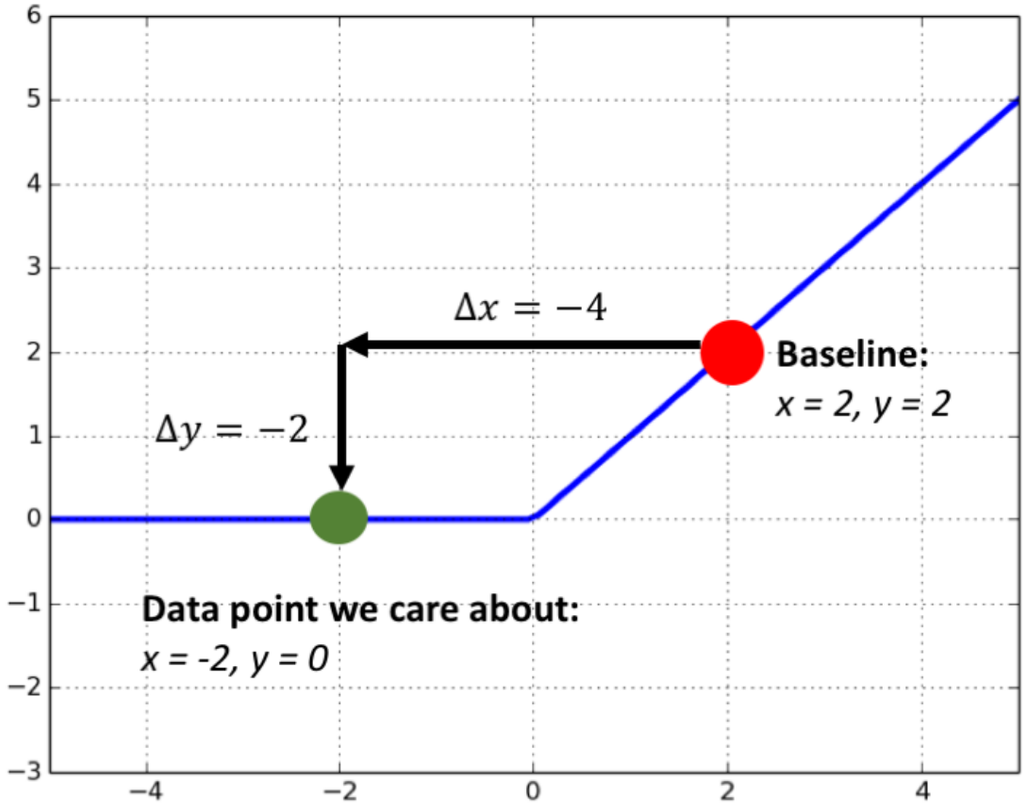
DeepLIFT

$$x_i \times \frac{\partial Y}{\partial x_i} \rightarrow (x_i - x_i^{baseline}) \times \frac{Y - Y^{baseline}}{x_i - x^{baseline}}$$

DeepLIFT

$$x_i \times \frac{\partial Y}{\partial x_i} \rightarrow (x_i - x_i^{baseline}) \times \frac{Y - Y^{baseline}}{x_i - x^{baseline}}$$

$$y = ReLU(x) = max(0, x)$$



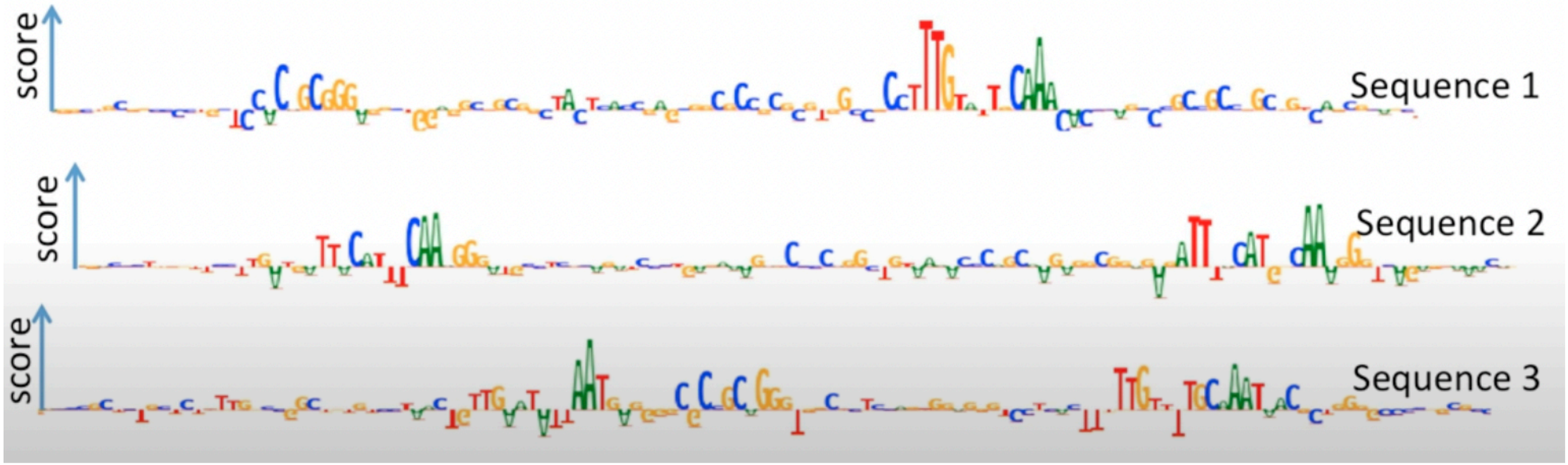
1. Calculating the slope

$$\frac{\Delta y}{\Delta x} = \frac{-2}{-4} = 0.5$$

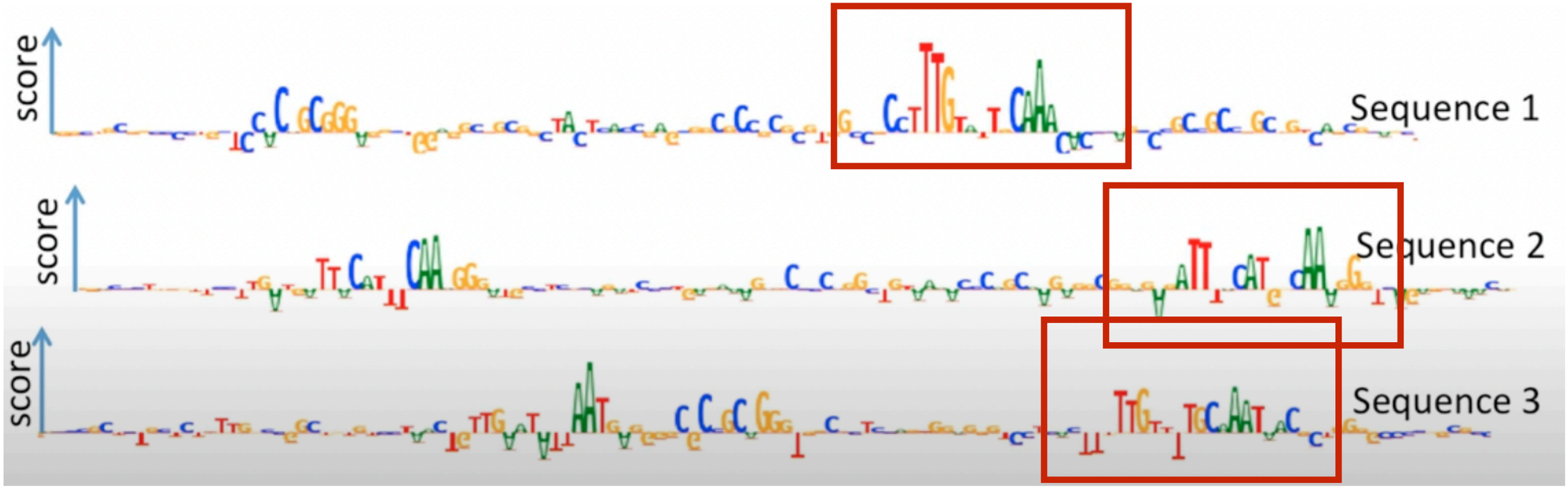
2. Finding the feature importance

$$\Delta x \times \frac{\Delta y}{\Delta x} = -4 \times 0.5 = -2$$

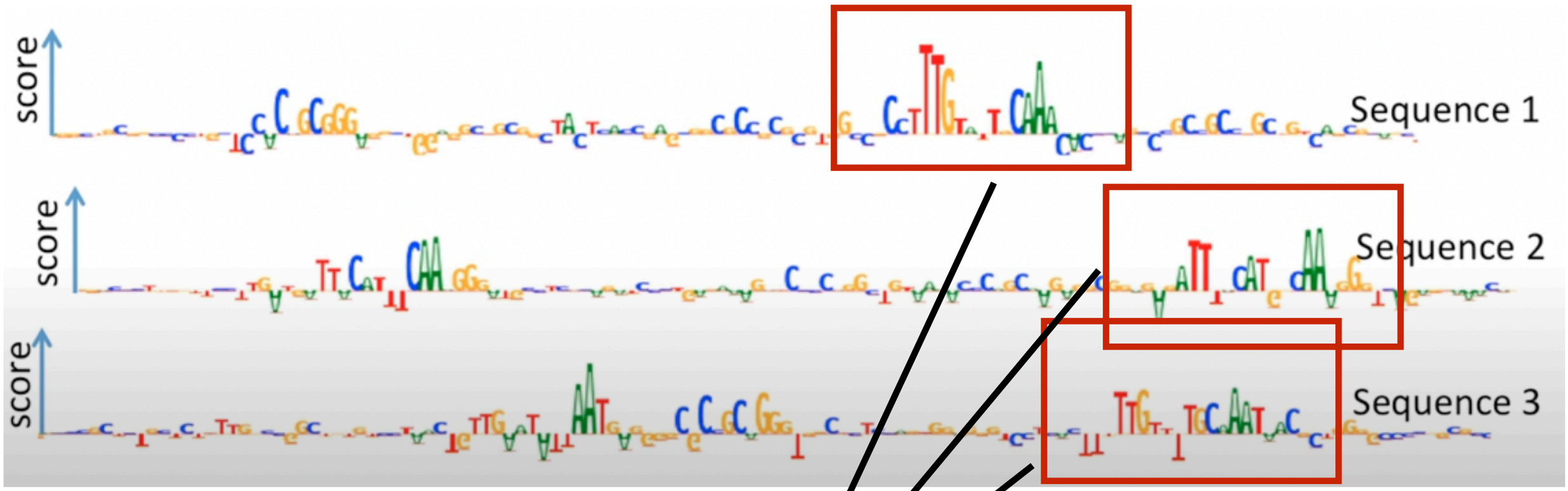
Saliency: Local to global



Saliency: Local to global



Saliency: Local to global



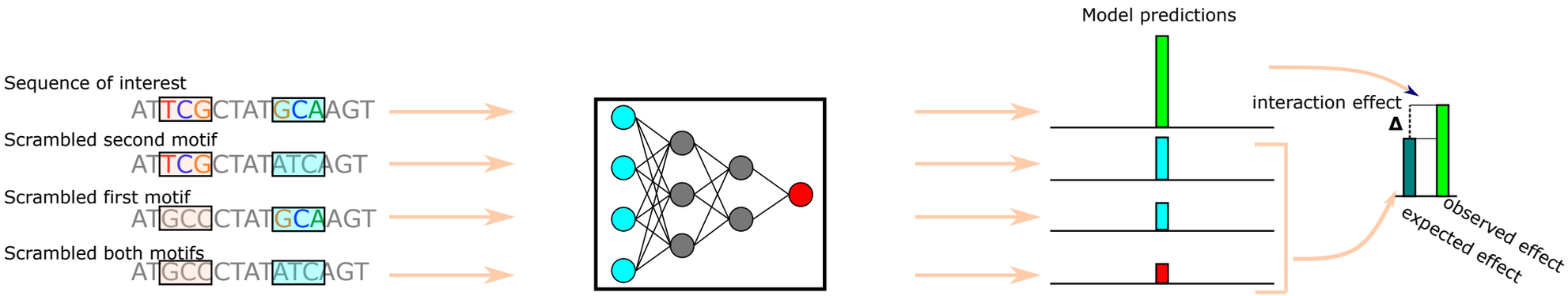
Cluster

DOI: 10.1038/s41588-021-00782-6

Outline

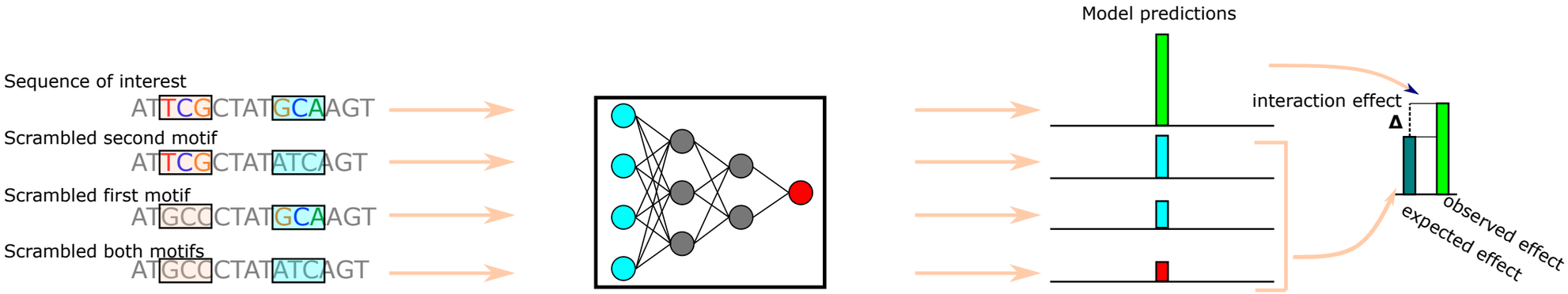
- Model-based interpretation
- Saliency map
- Understanding interactions

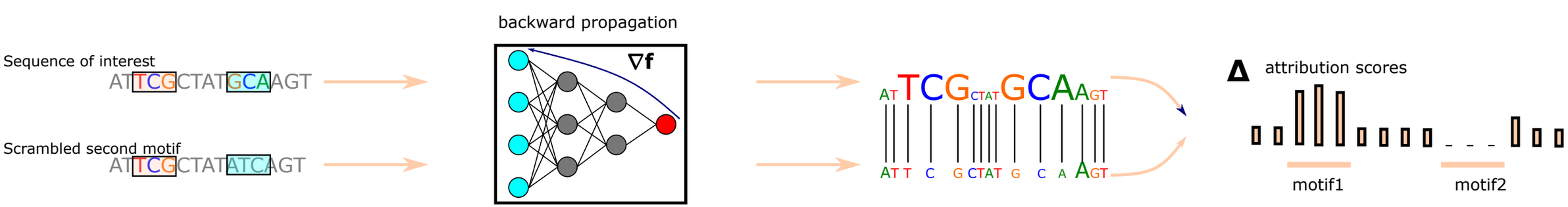
Multiple perturbation for identifying interactions



Novakovsky et al, In Press

Multiple perturbation for identifying interactions



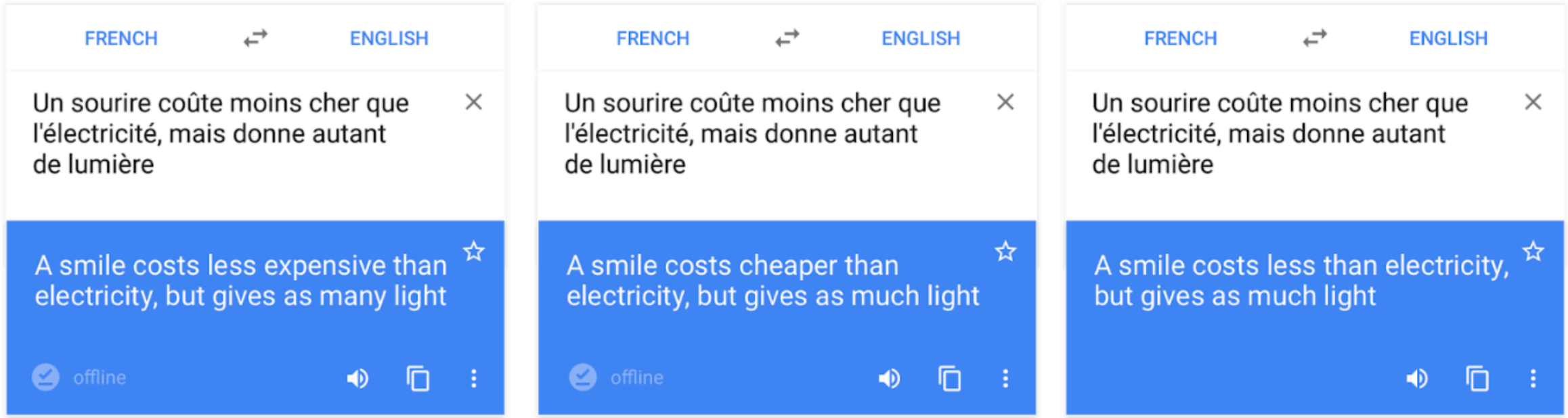


Novakovsky et al, In Press

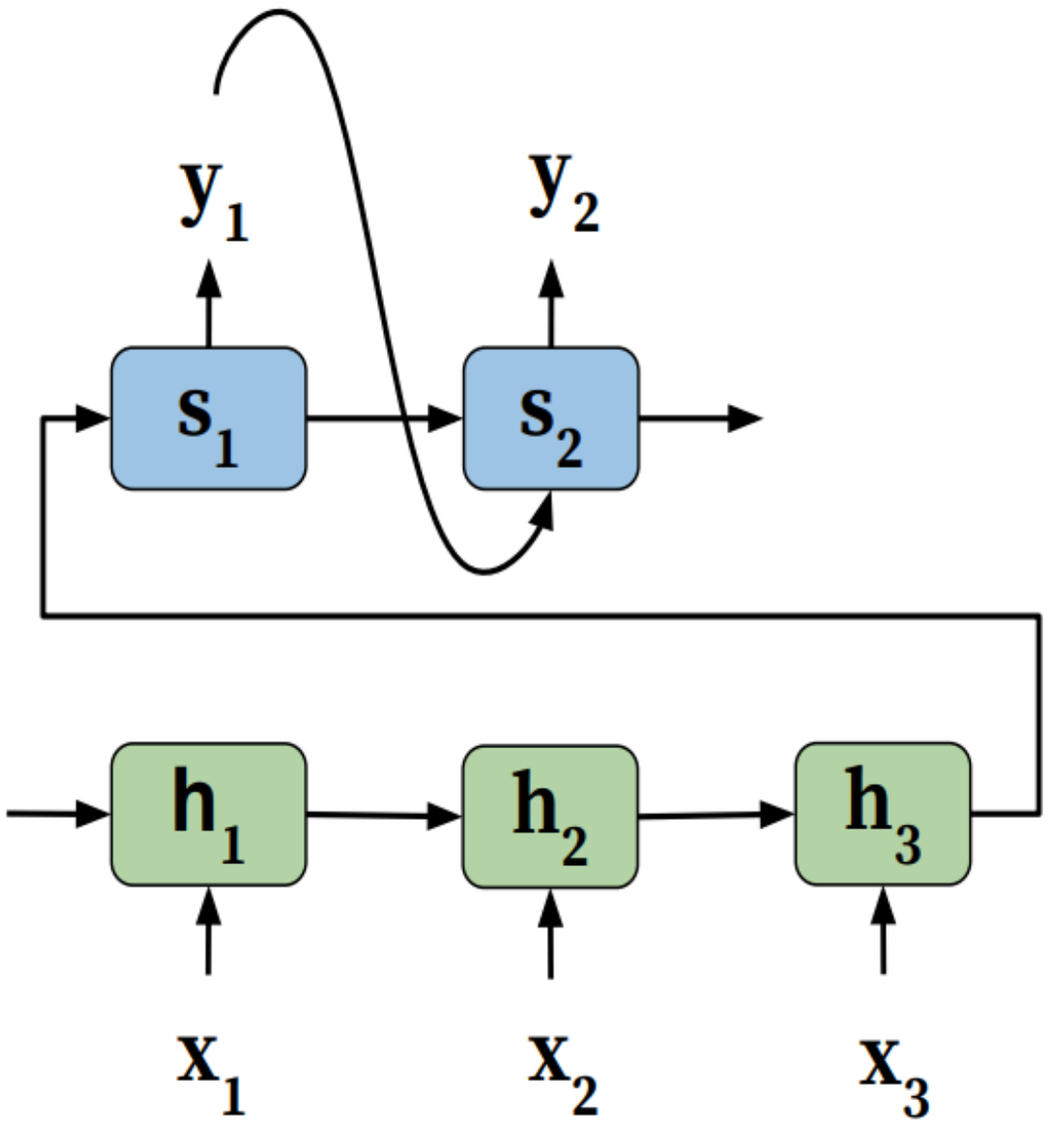
Limitations of interpretability

Attention

Motivating problem: translation

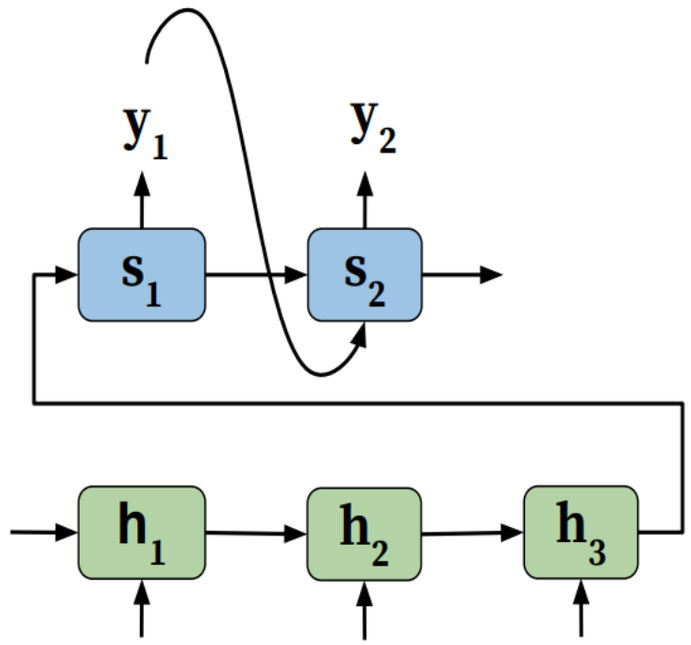


Idea #1: Recurrent neural network



Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Idea #1: Recurrent neural network



Encode $h_i = f(x_i, h_{i-1})$

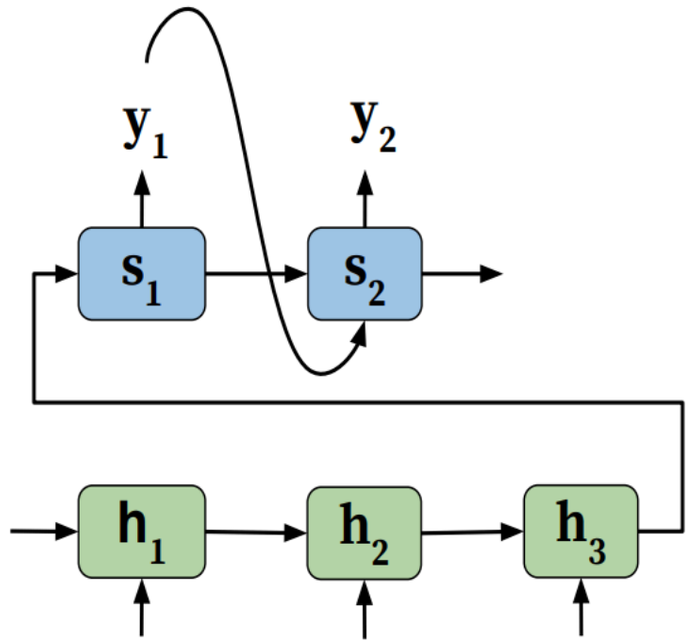
Context $c = h_T$

Decode $s_j = f(s_{j-1}, y_{j-1}, c)$

Generate $y_j = g(y_{j-1}, s_j, c)$

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

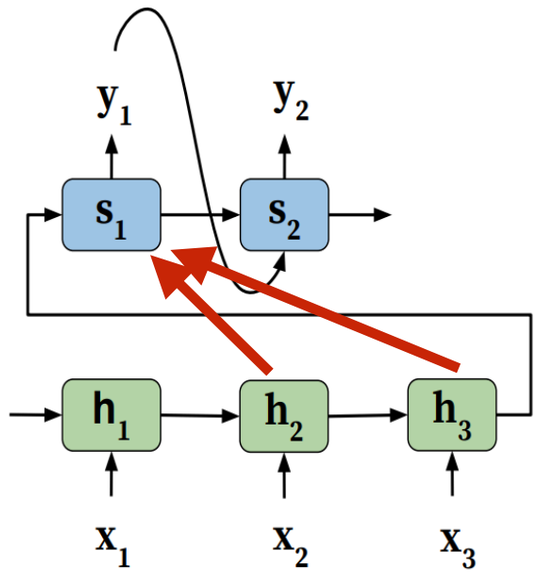
Idea #1: Recurrent neural network



		# parameters
Encode	$h_i = f(x_i, h_{i-1})$	*(X+H)×H×L*
Context	$c = h_T$	*0*
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$	*(H+Y+H)×H×L*
Generate	$y_j = g(y_{j-1}, s_j, c)$	*(Y+H+H)×Y×L*

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Idea #2: Give the network access to more hidden states



parameters

Encode	$h_i = f(x_i, h_{i-1})$	
Context	$c = h_T$ ~~X~~ $h_{T-M:T}$	
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$	$(H+Y+MH)\times H\times L$
Generate	$y_j = g(y_{j-1}, s_j, c)$	

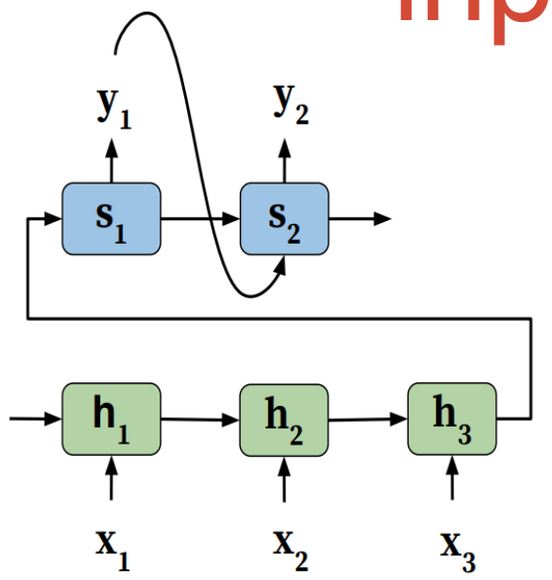
Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Aside: What makes one architecture better than another?

- Complexity vs. generalizability tradeoff.
- Inductive bias.
- Computational efficiency.

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Idea #3: Do some computation for each input, then average



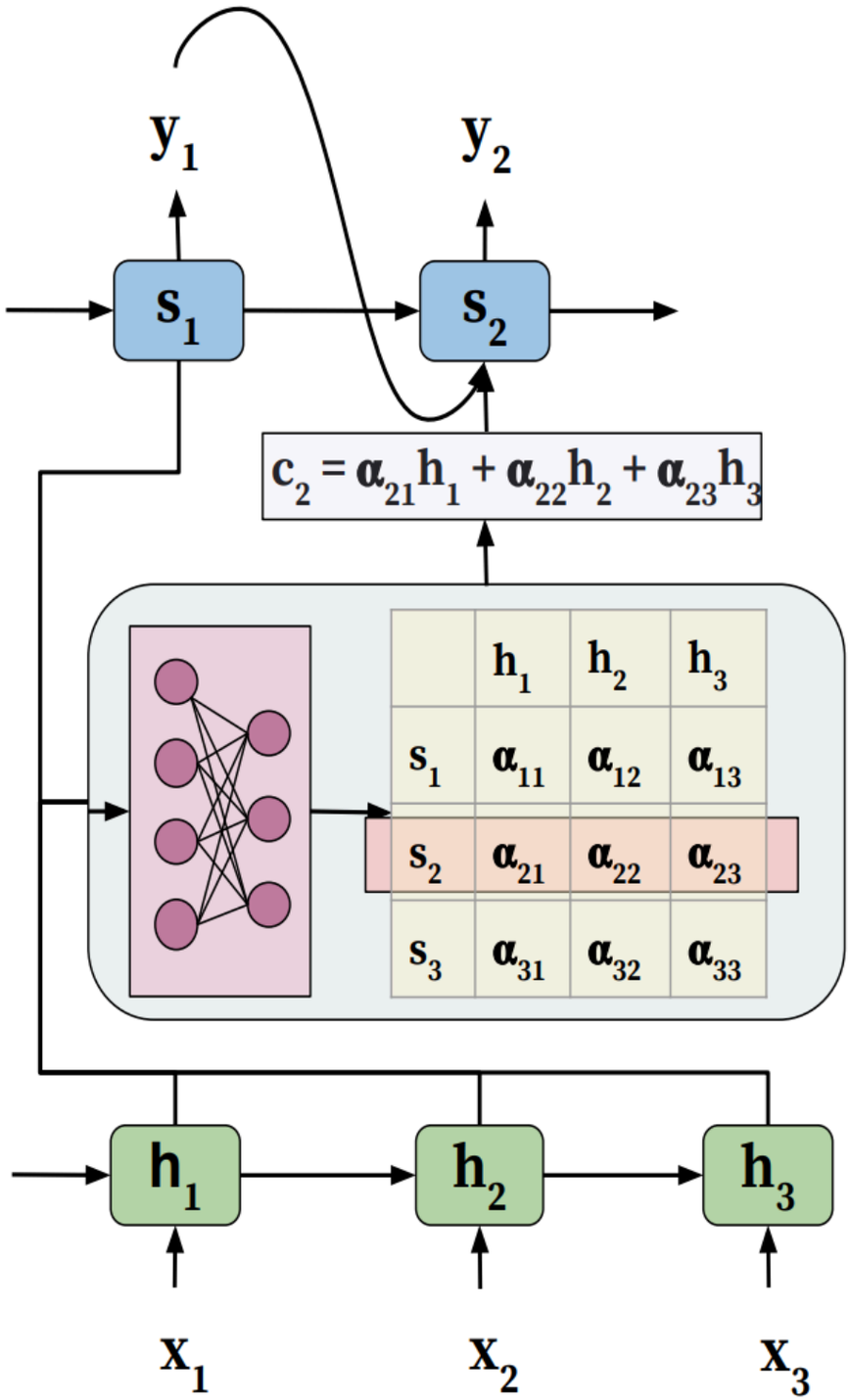
parameters

Encode $h_i = f(x_i, h_{i-1})$

Decode $s_j = \cancel{f(s_{j-1}, y_{j-1}, c)}$

$$\sum_{i=1}^{T} f(s_{j-1}, y_{j-1}, h_i)$$ *(H+Y+H)×H×L*

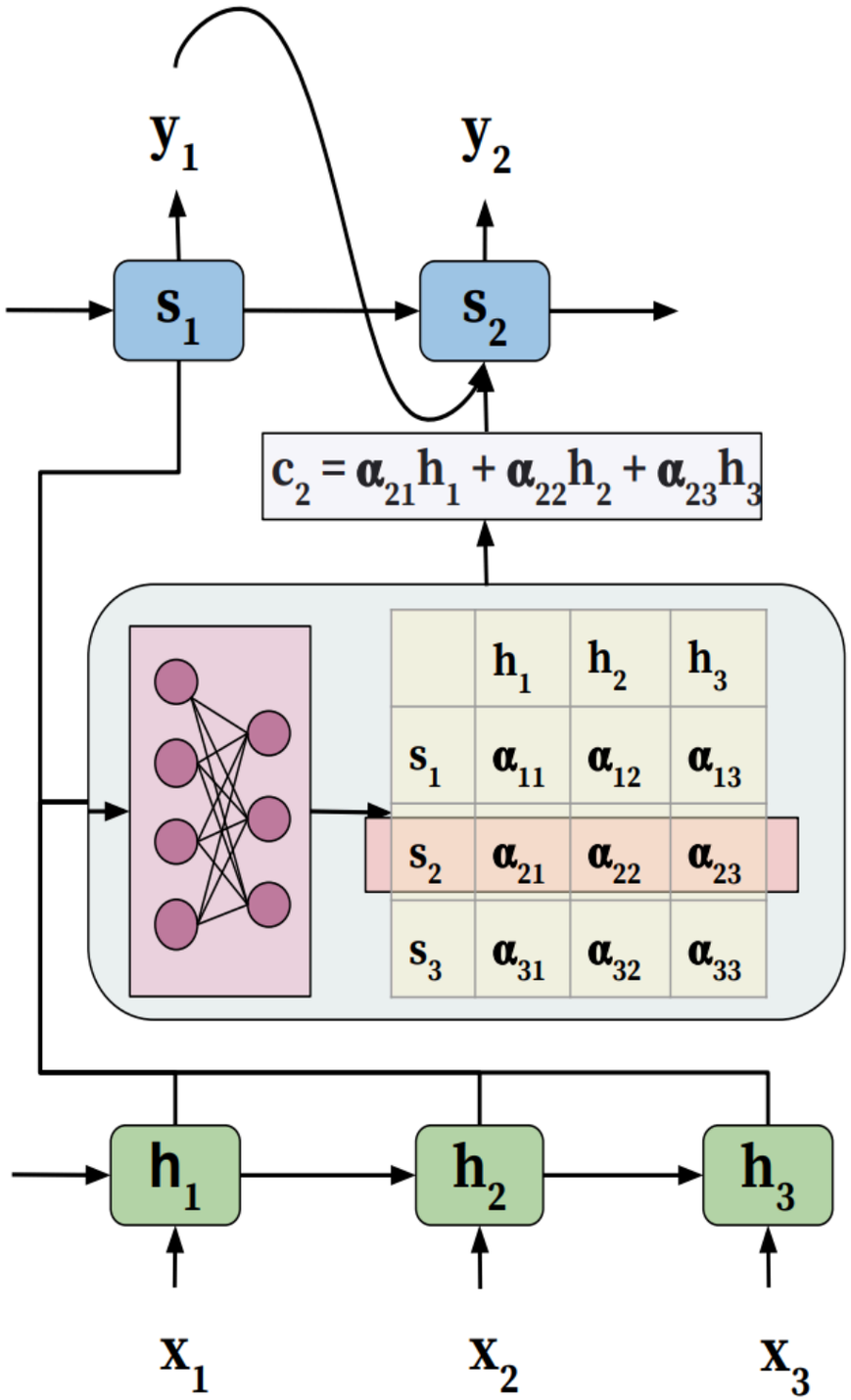
Attention



Function	Traditional Encoder-Decoder
Encode	$h_i = f(x_i, h_{i-1})$
Context	$c = h_T$
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$
Generate	$y_j = g(y_{j-1}, s_j, c)$

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

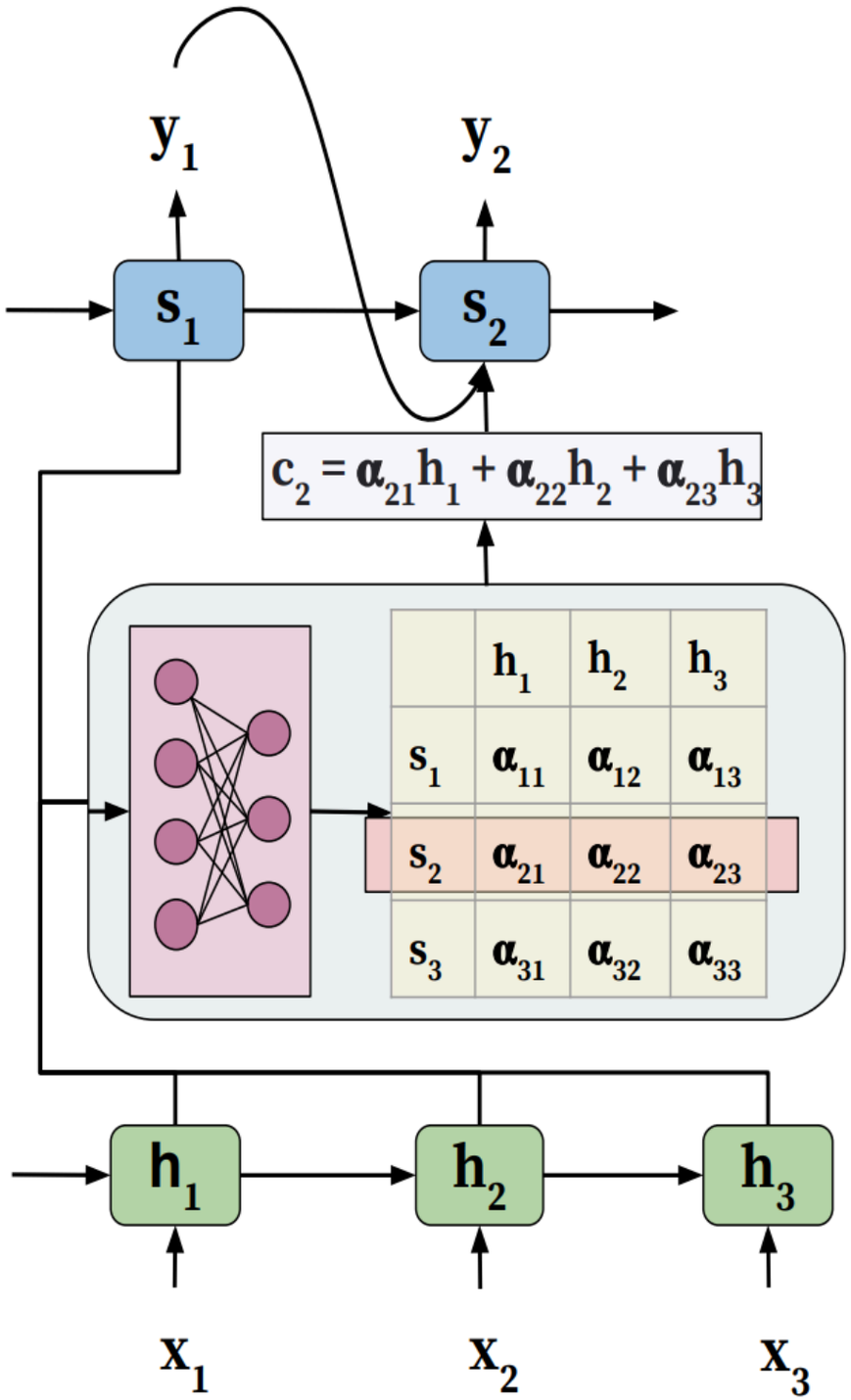
Attention



Function	Traditional Encoder-Decoder	Encoder-Decoder with Attention
Encode	$h_i = f(x_i, h_{i-1})$	$h_i = f(x_i, h_{i-1})$
Context	$c = h_T$	
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$	$s_j = f(s_{j-1}, y_{j-1}, c_j)$
Generate	$y_j = g(y_{j-1}, s_j, c)$	$y_j = g(y_{j-1}, s_j, c_j)$

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

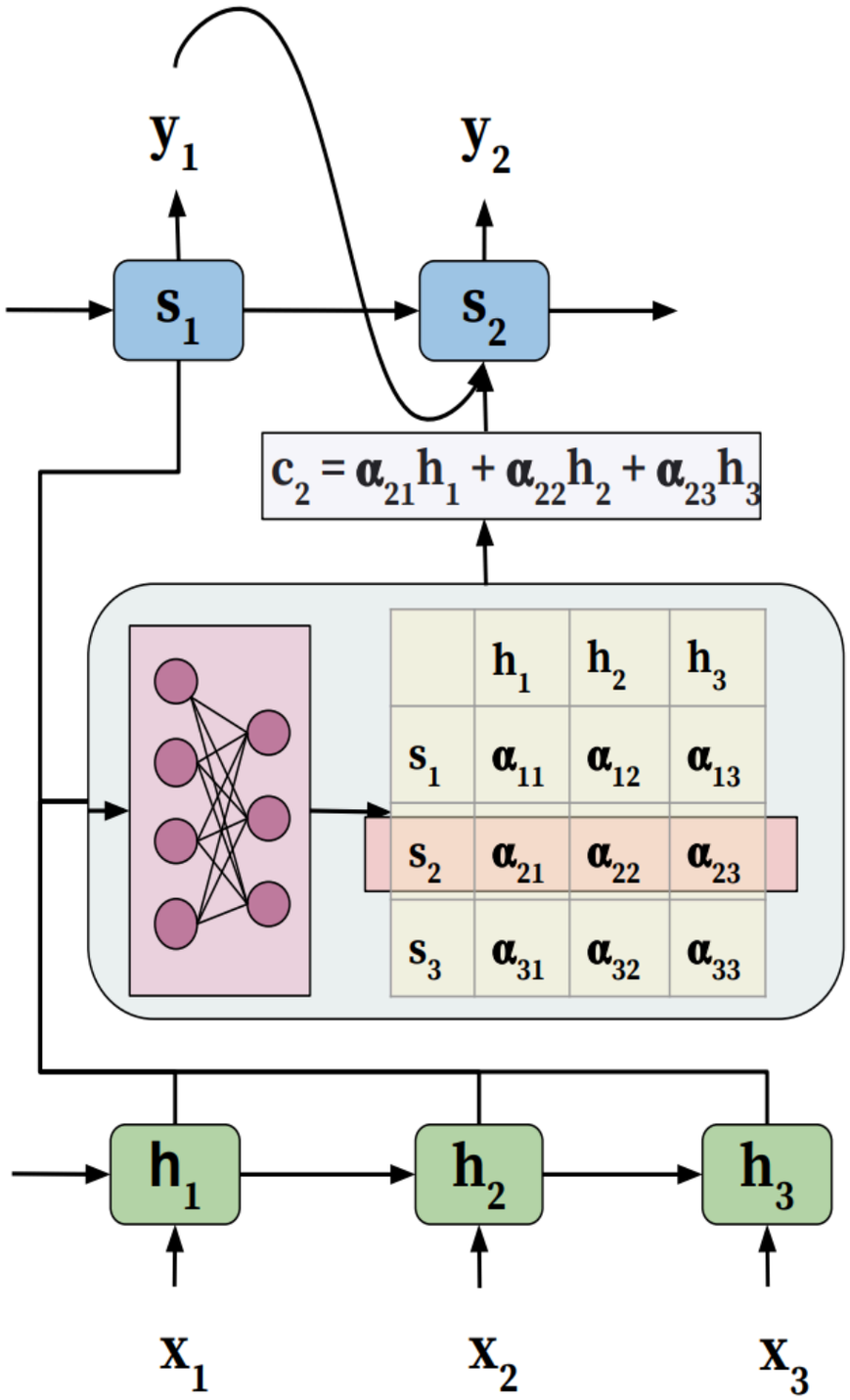
Attention



Function	Traditional Encoder-Decoder	Encoder-Decoder with Attention
Encode	$h_i = f(x_i, h_{i-1})$	$h_i = f(x_i, h_{i-1})$
Context	$c = h_T$	
		$\alpha_{ij} = p(e_{ij})$
		$e_{ij} = a(s_{j-1}, h_i)$
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$	
Generate	$y_j = g(y_{j-1}, s_j, c)$	$y_j = g(y_{j-1}, s_j, c_j)$

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

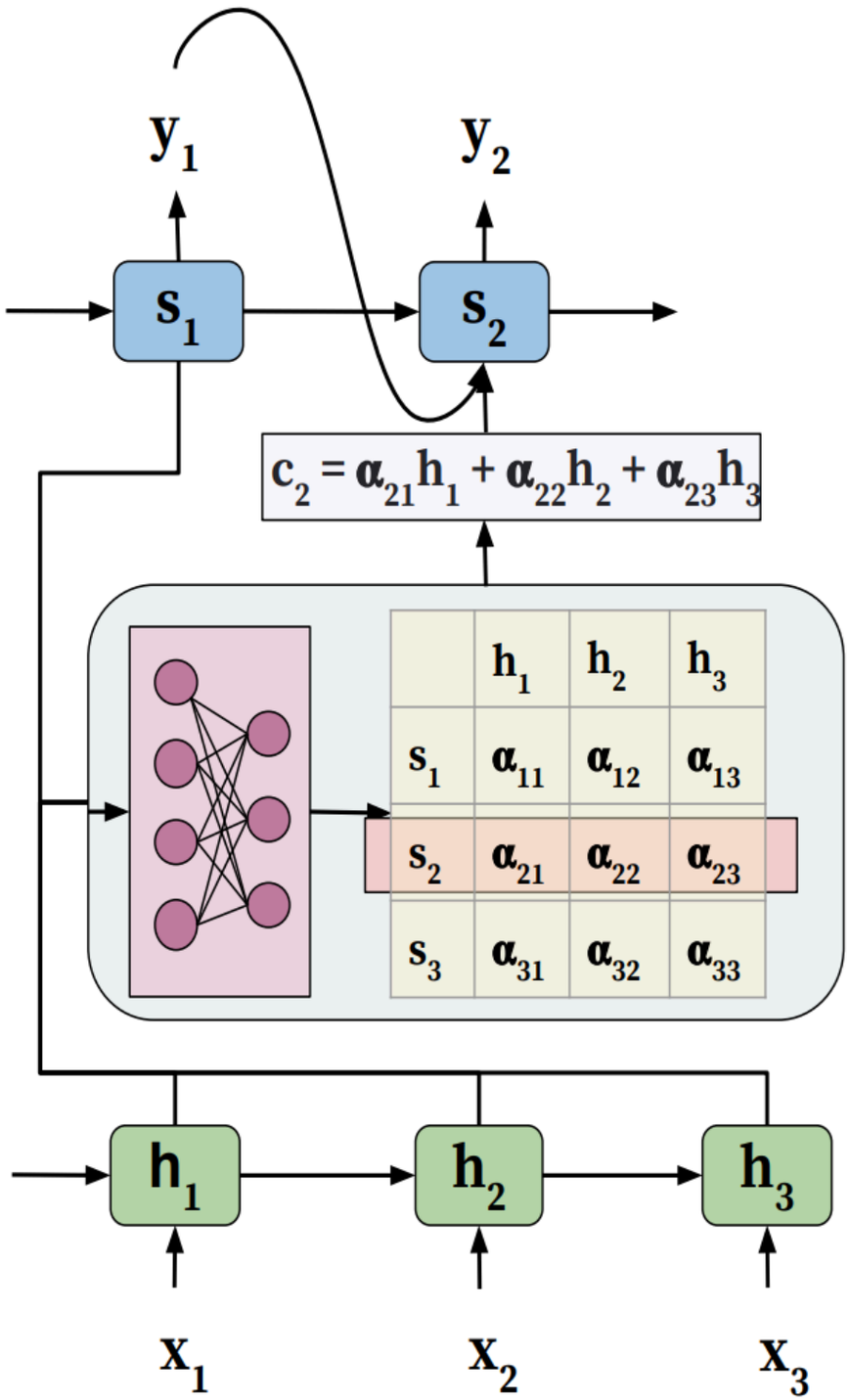
Attention



Function	Traditional Encoder-Decoder	Encoder-Decoder with Attention
Encode	$h_i = f(x_i, h_{i-1})$	$h_i = f(x_i, h_{i-1})$
Context	$c = h_T$	$c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$
		$\alpha_{ij} = p(e_{ij})$
		$e_{ij} = a(s_{j-1}, h_i)$
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$	$s_j = f(s_{j-1}, y_{j-1}, c_j)$
Generate	$y_j = g(y_{j-1}, s_j, c)$	$y_j = g(y_{j-1}, s_j, c_j)$

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Attention



Function	Encoder-Decoder with Attention	# parameters
Encode	$h_i = f(x_i, h_{i-1})$	$(X+H)\times H\times L$
Context	$c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$	0
	$\alpha_{ij} = p(e_{ij})$	0
	$e_{ij} = a(s_{j-1}, h_i)$	$(H+H)\times H\times L$
Decode	$s_j = f(s_{j-1}, y_{j-1}, c_j)$	$(H+Y+H)\times H\times L$
Generate	$y_j = g(y_{j-1}, s_j, c_j)$	

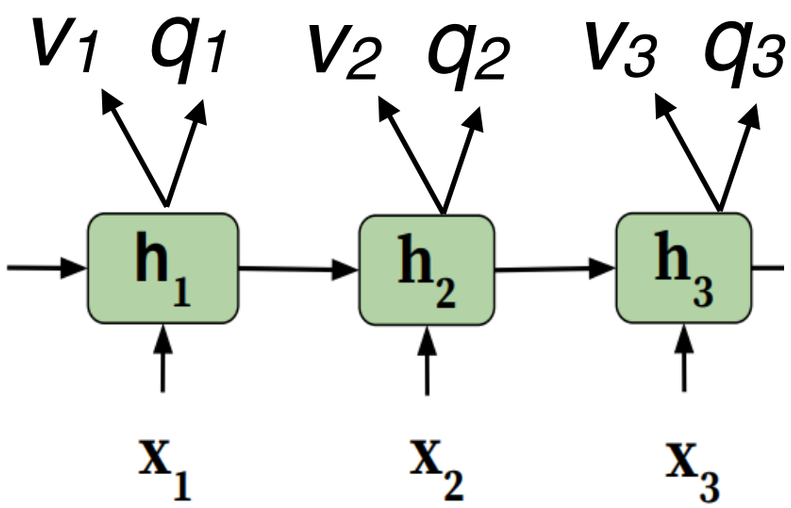
Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Outline

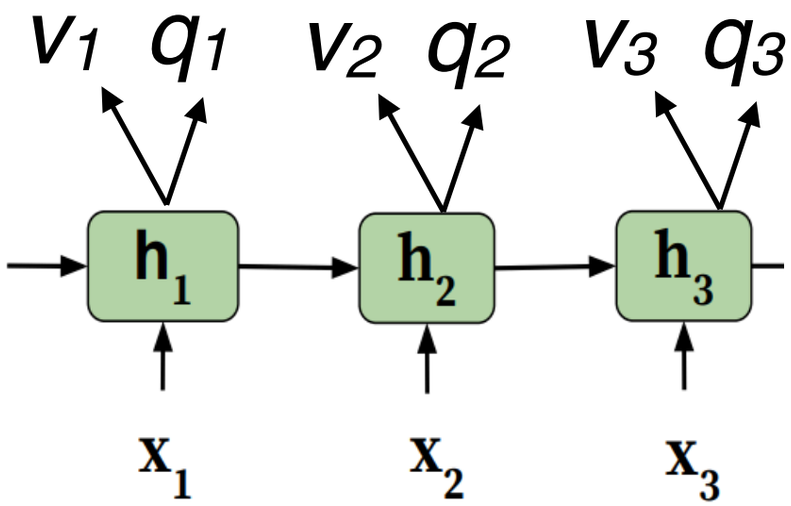
- Attention variants
- Self-attention and transformers
- Applications

Generalized attention

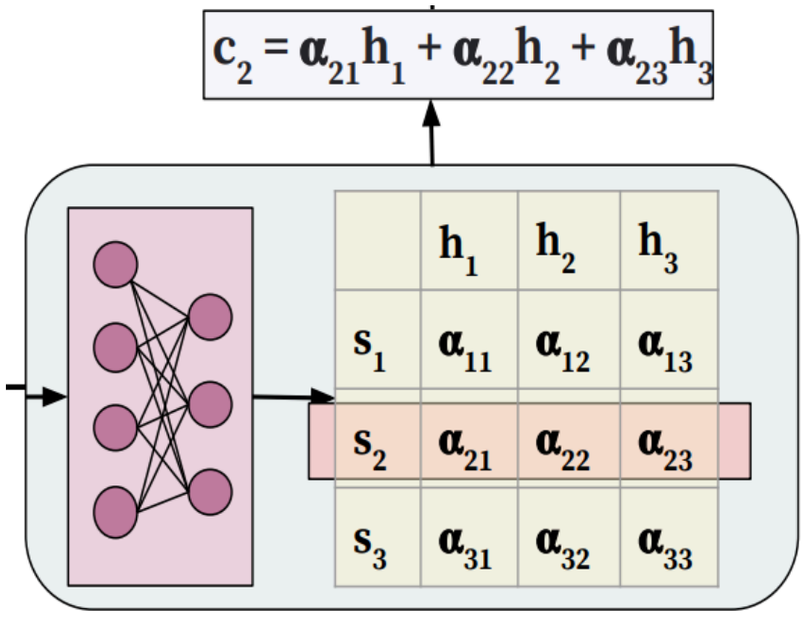
Generalized attention



Generalized attention

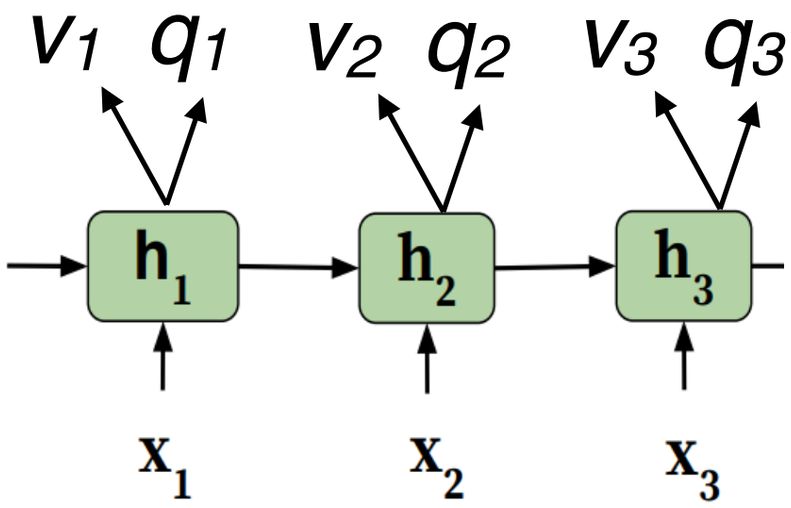


Function	Equation
similarity	$a(k_i, q) = sim(k_i, q)$
dot product	$a(k_i, q) = q^T k_i$
scaled dot product	$a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$

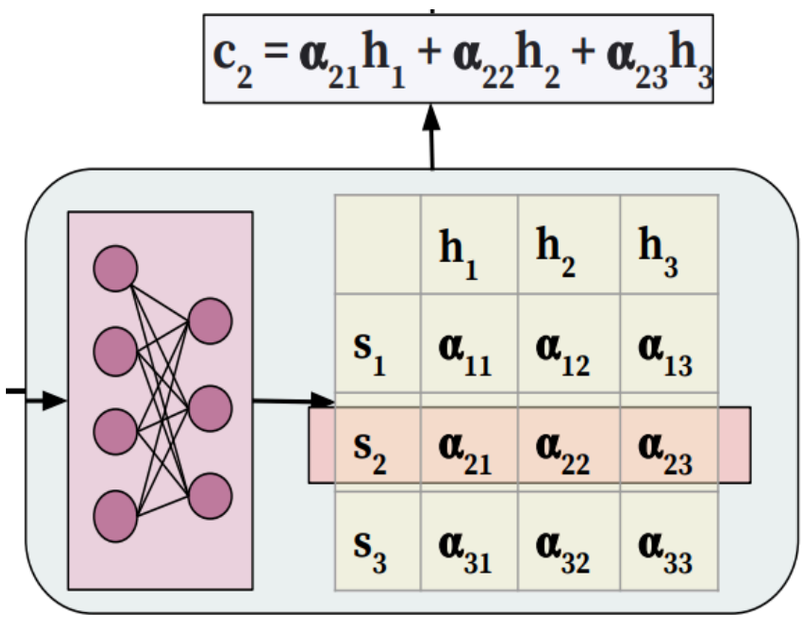


Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Generalized attention

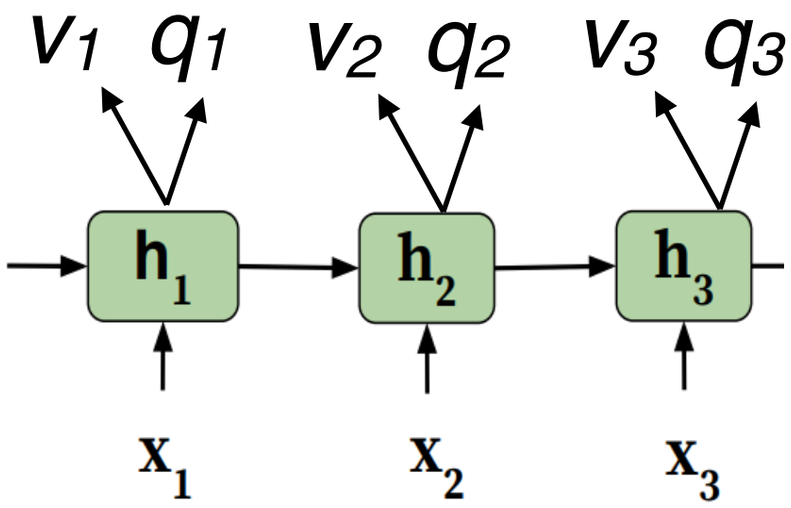


Function	Equation
similarity	$a(k_i, q) = sim(k_i, q)$
dot product	$a(k_i, q) = q^T k_i$
scaled dot product	$a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$
general	$a(k_i, q) = q^T W k_i$
biased general	$a(k_i, q) = k_i(Wq + b)$

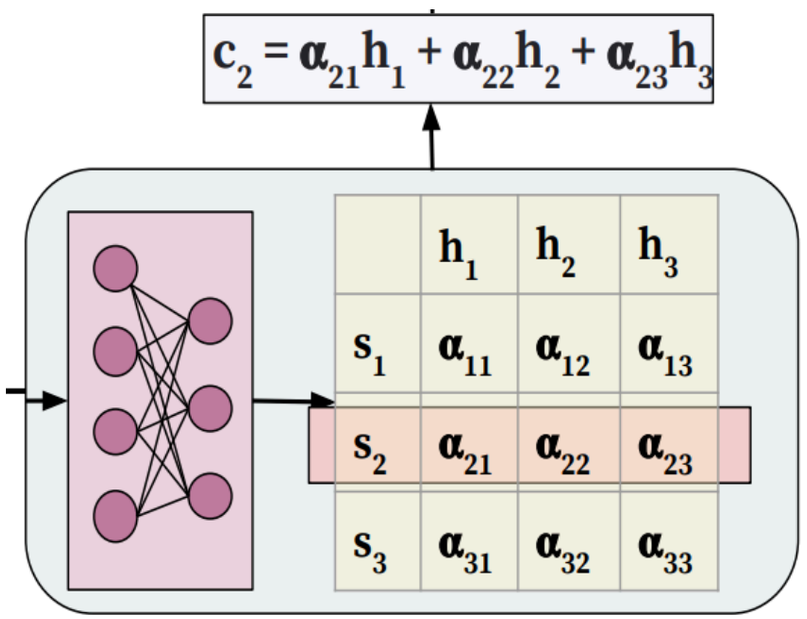


Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Generalized attention



Function	Equation
similarity	$a(k_i, q) = sim(k_i, q)$
dot product	$a(k_i, q) = q^T k_i$
scaled dot product	$a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$
general	$a(k_i, q) = q^T W k_i$
biased general	$a(k_i, q) = k_i(Wq + b)$
activated general	$a(k_i, q) = act(q^T W k_i + b)$
deep	$a(k_i, q) = w_{imp}^T E^{(L-1)} + b^L$ $E^{(l)} = act(W_l E^{(l-1)} + b^l$ $E^{(1)} = act(W_1 k_i + W_0 q) + b^l$



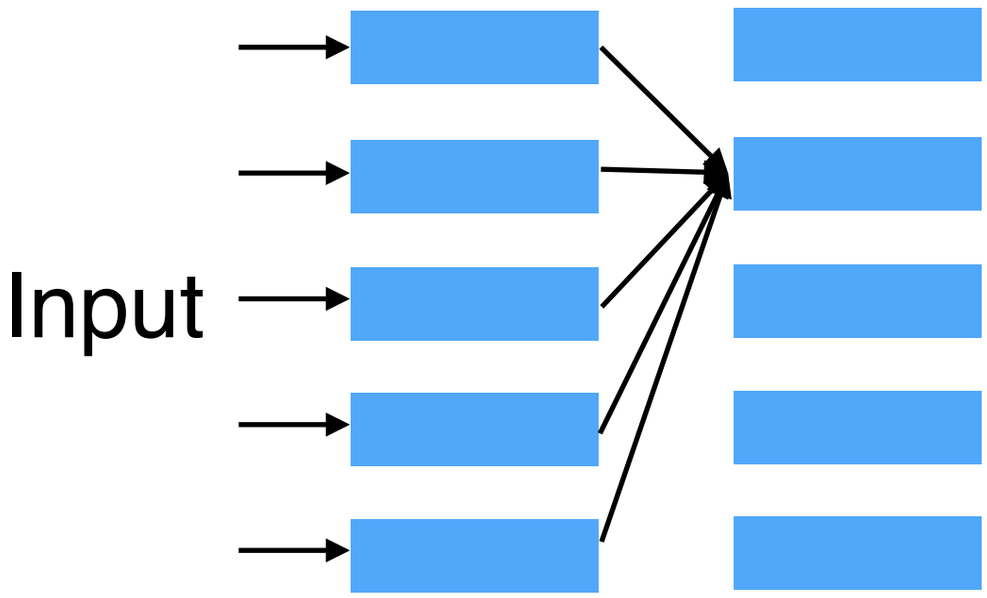
Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1-32.

Hard vs. soft attention

Outline

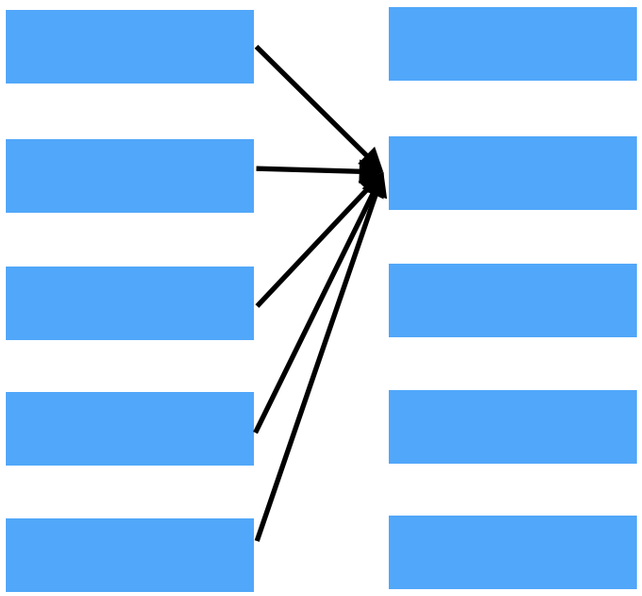
- Attention variants
- Self-attention and transformers
- Applications

Self-attention



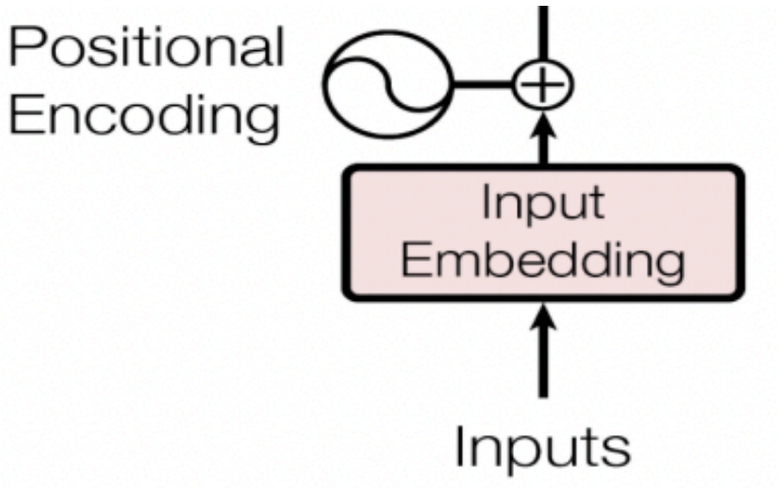
$$h_j^{(2)} = \sum_{i=1}^{T} \alpha_{ij} h_i^{(1)}$$

Positional encoding

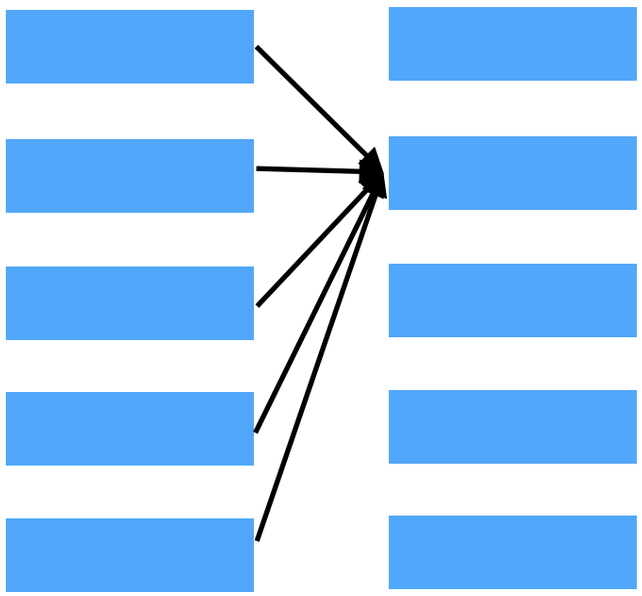


How to represent relative position:

- Idea #1: One-hot encoding of distance.

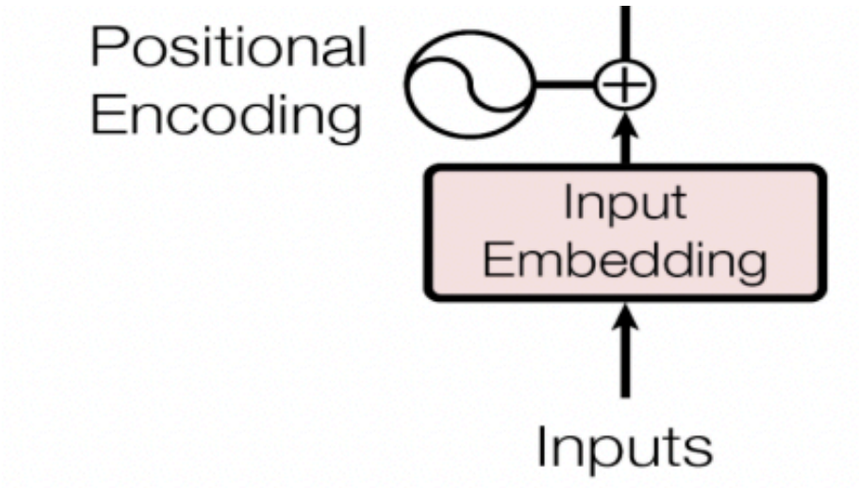


Positional encoding

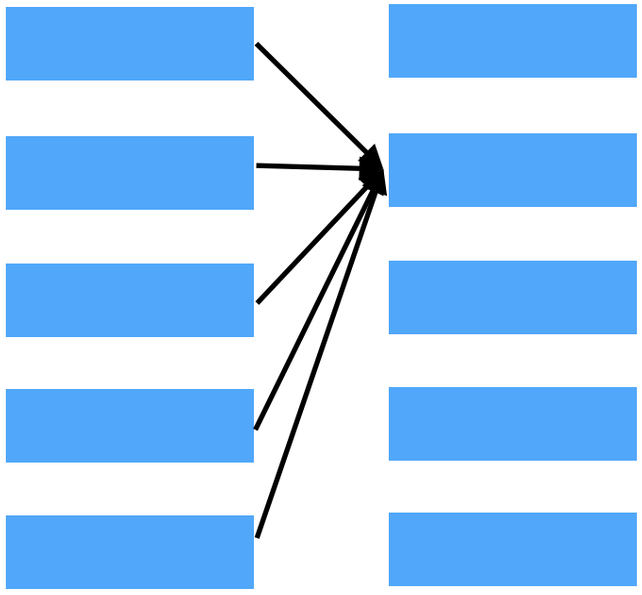


How to represent relative position:

- Idea #1: One-hot encoding of distance.
- Idea #2: Binary encoding.

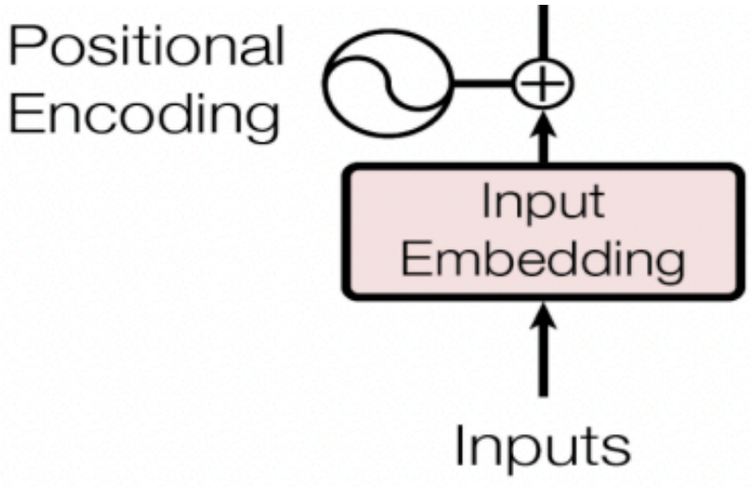


Positional encoding



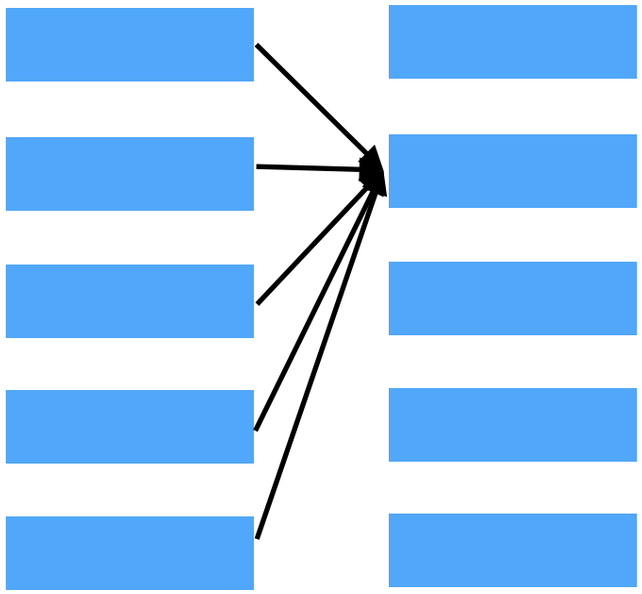
How to represent relative position:

- Idea #1: One-hot encoding of distance.
- Idea #2: Binary encoding.
- Idea #3: Sinusoids of different frequencies.



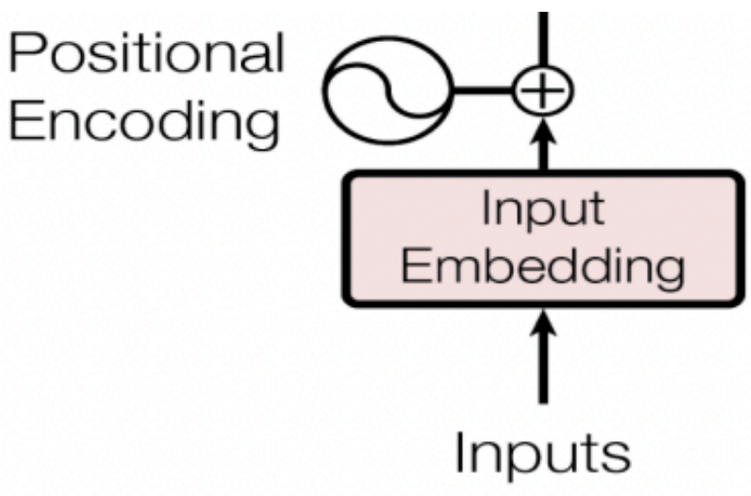
$$\overrightarrow{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

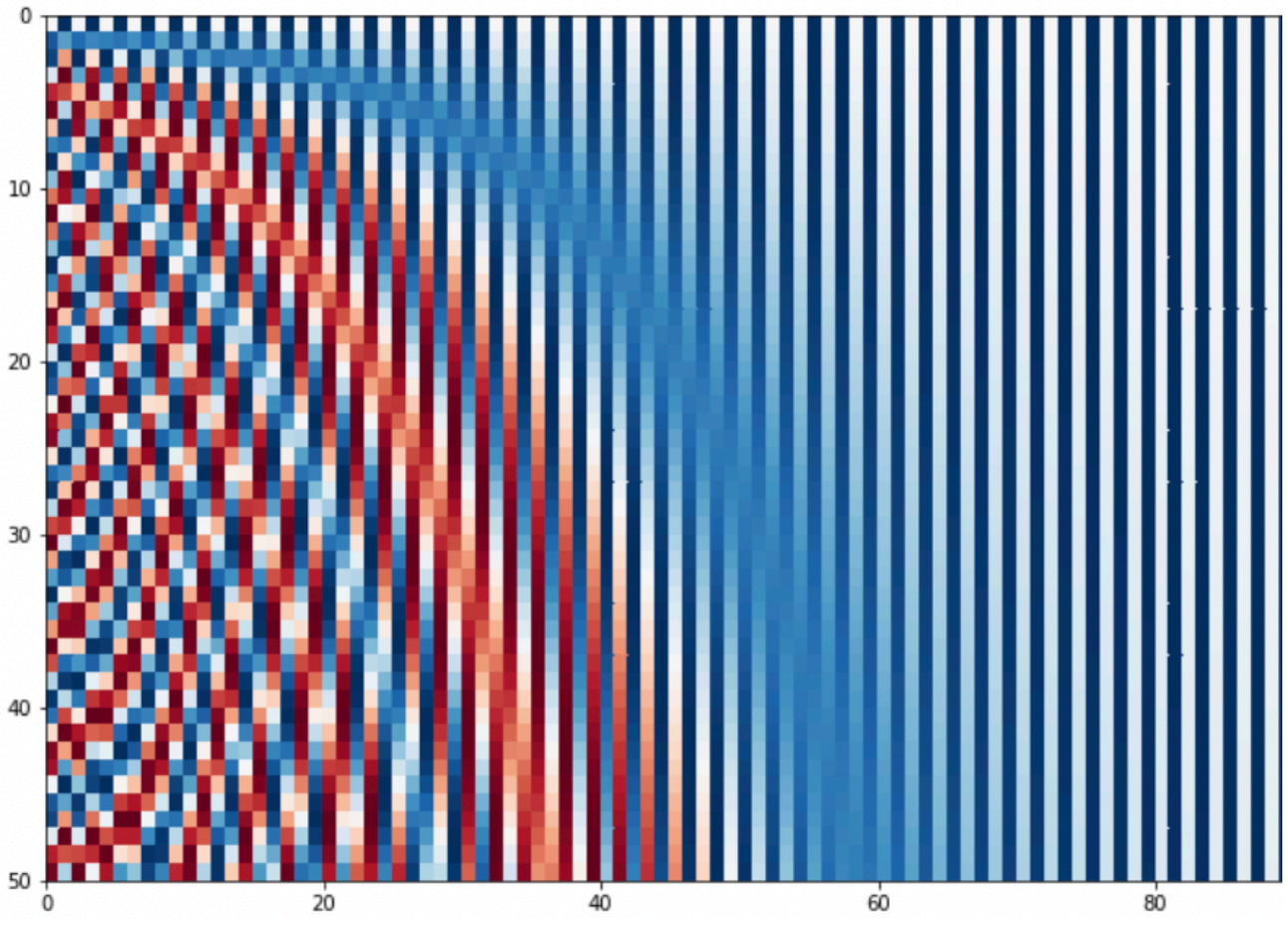
Positional encoding



How to represent relative position:

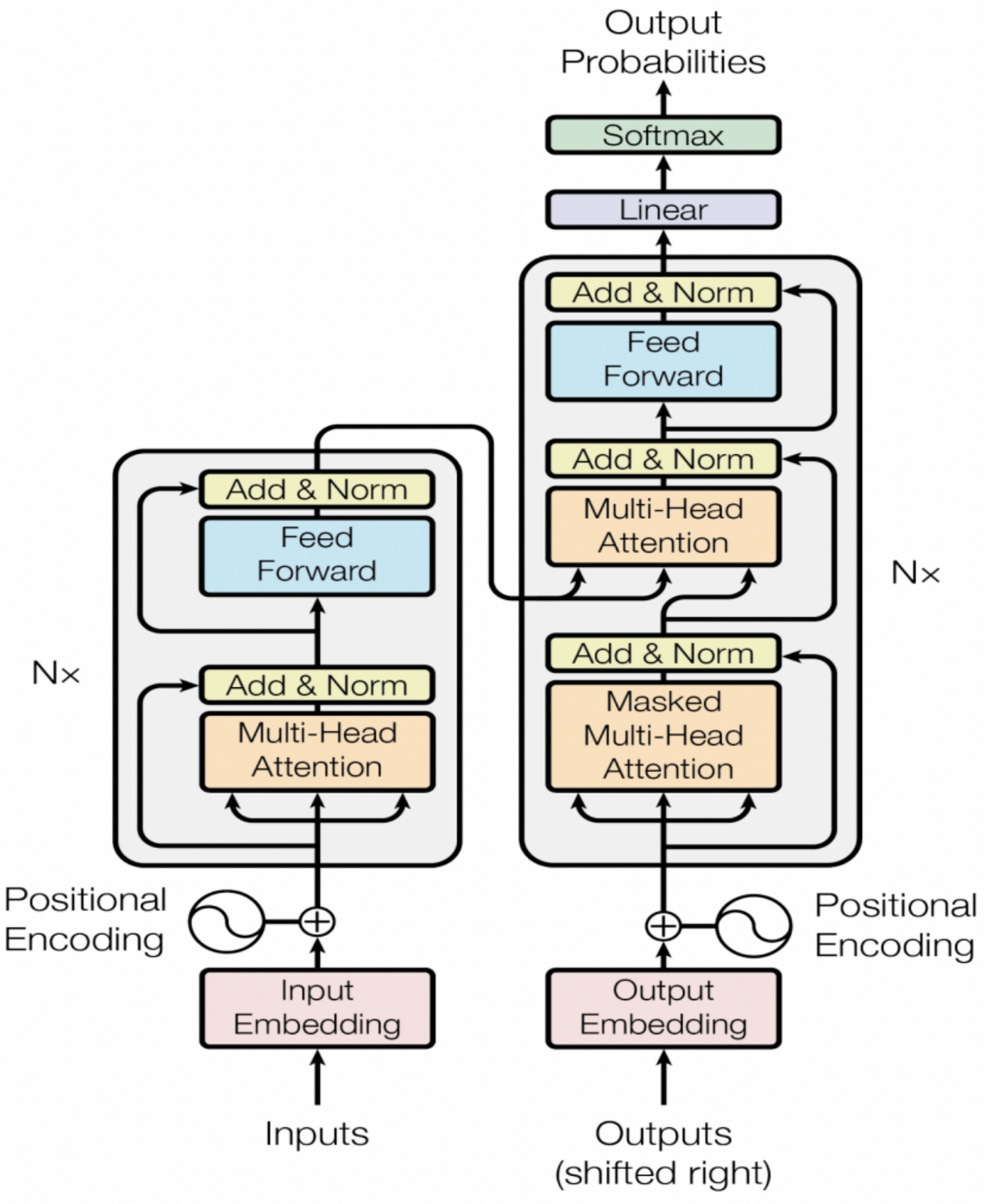
- Idea #1: One-hot encoding of distance.
- Idea #2: Binary encoding.
- Idea #3: Sinusoids of different frequencies.





https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Transformer

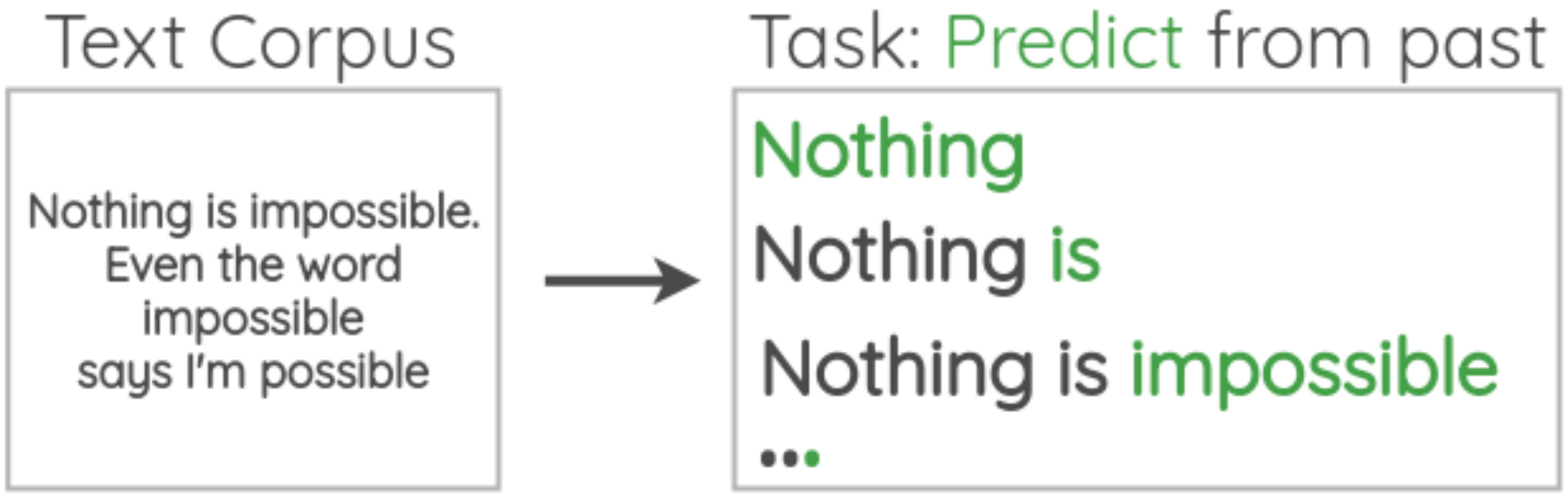


Vaswani et al, 2017

Outline

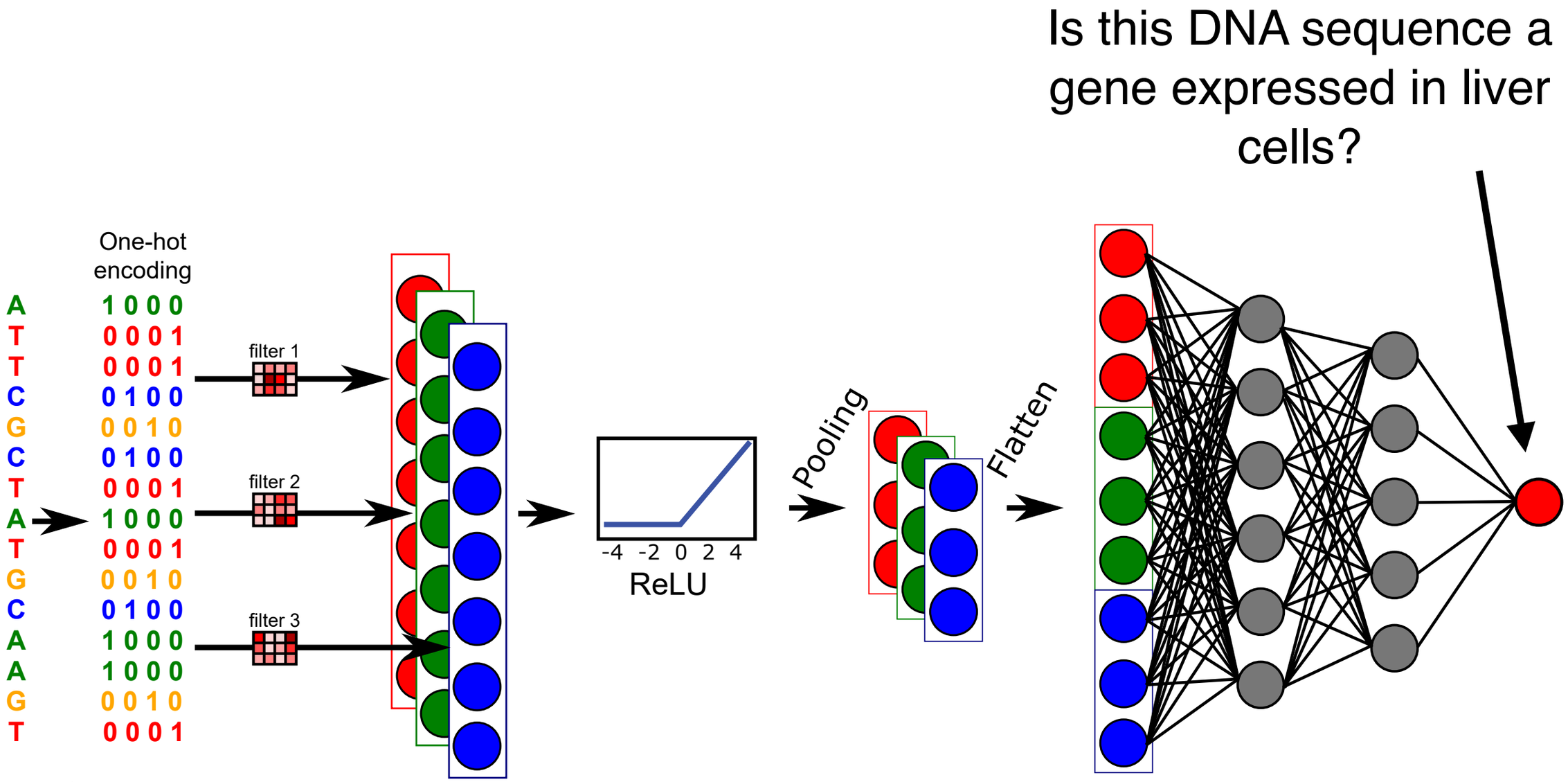
- Attention variants
- Self-attention and transformers
- Applications

Self-supervised pre-training of language models



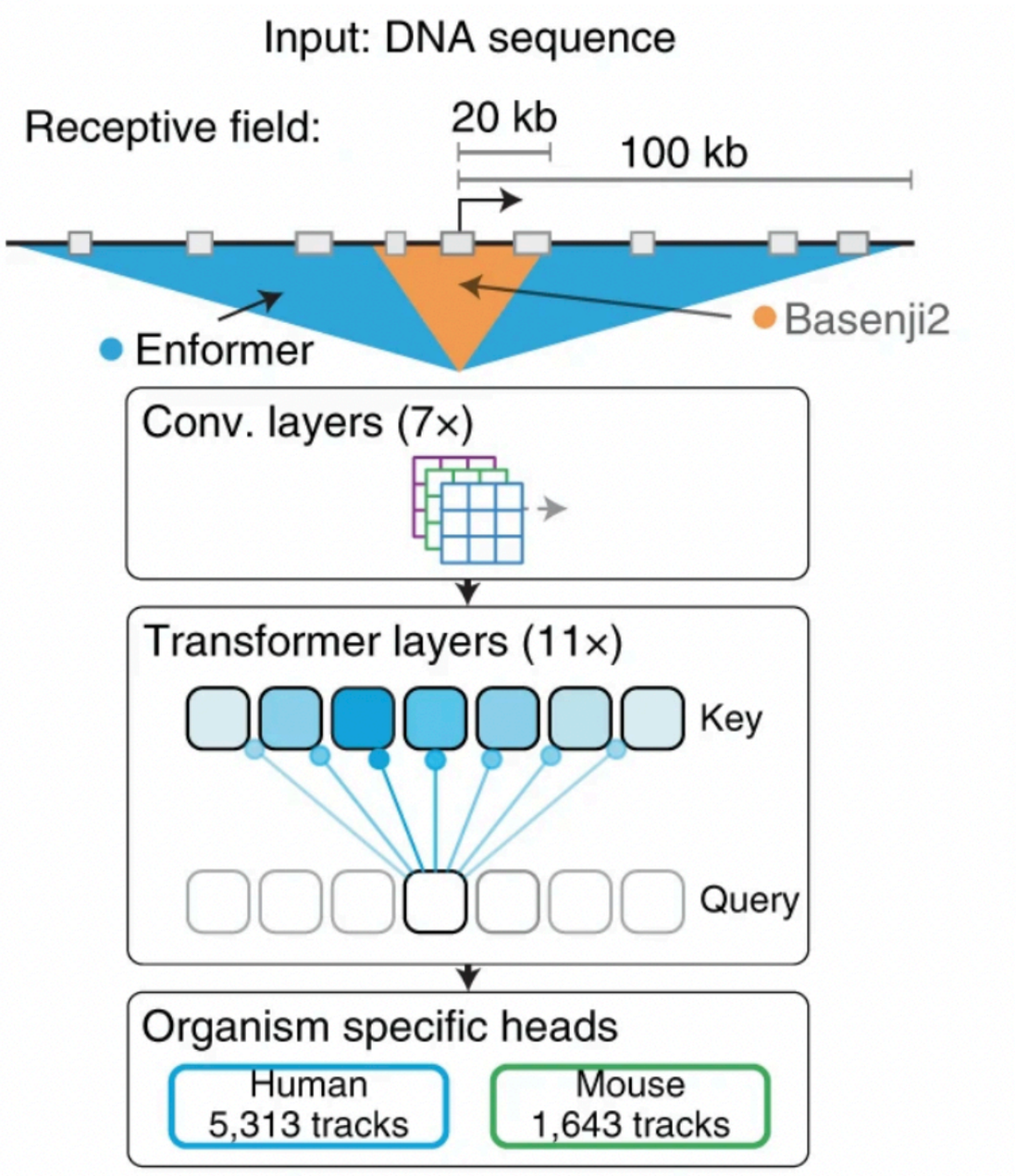
https://amitness.com/2020/05/self-supervised-learning-nlp/

Predicting gene expression



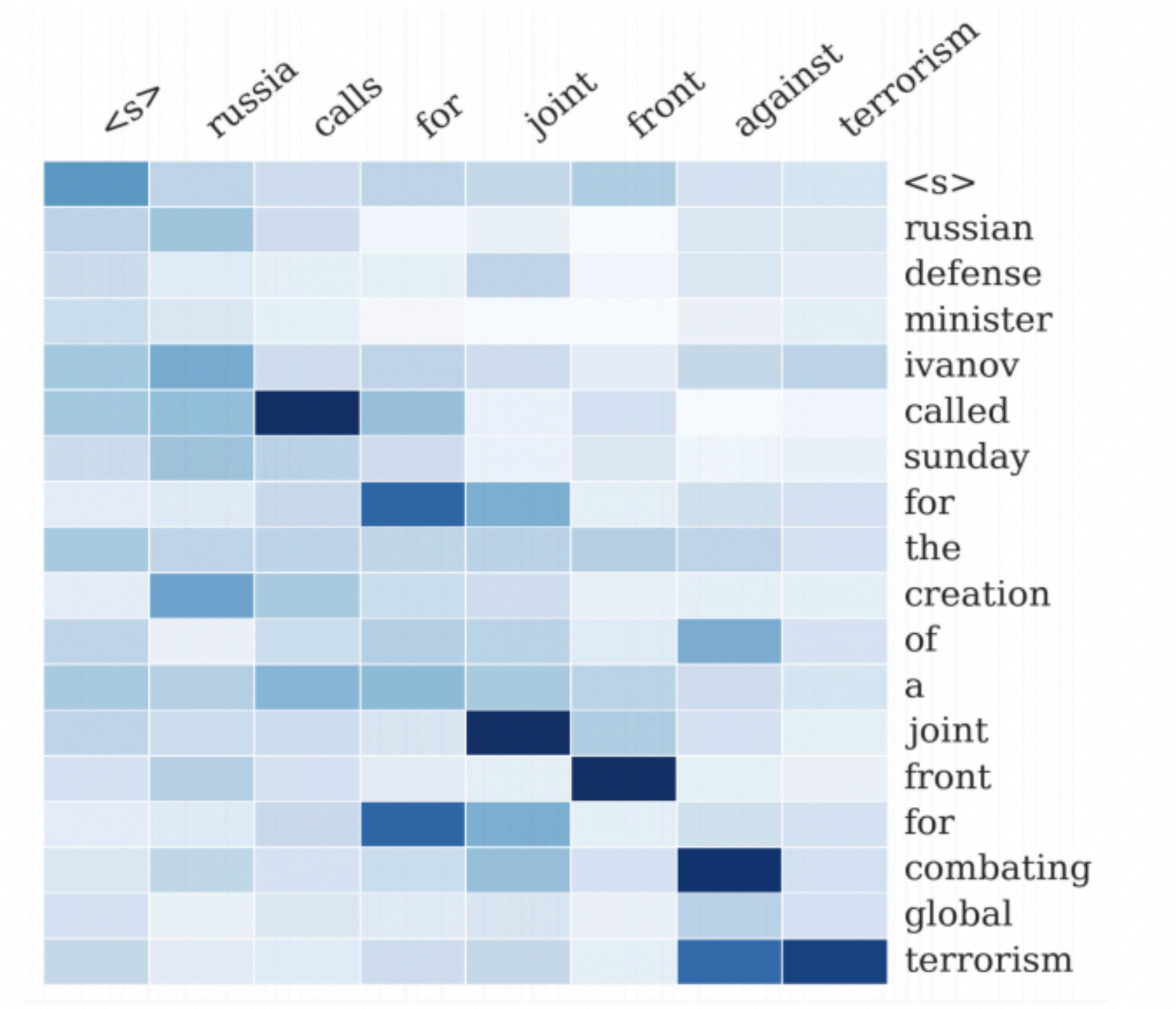
Novokovsky et al, in press

Predicting gene expression



Avsec, Ž., Agarwal, V., Visentin, D. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nat Methods* **18**, 1196–1203 (2021). https://doi.org/10.1038/s41592-021-01252-x

Attention can be interpretable



Questions and future directions: Attention

- How can we trade off complexity, generalizability and computation?
- In each application, what attention architectures are effective?
- In what settings are attention weights provide a useful explanation?

Questions and future directions: Interpretation

- Do interpretations report spurious associations present in the training set?
- How can interpretation account for un-identifiability of machine learning?
- What interpretation approaches are relevant for a given goal?
- How can we aggregate gradient-based saliency into a global (or large-scale local) interpretation?

Questions and future directions: Interpretation

- Do interpretations report spurious associations present in the training set?
- How can interpretation account for un-identifiability of machine learning?
- What interpretation approaches are relevant for a given goal?
- How can we aggregate gradient-based saliency into a global (or large-scale local) interpretation?

Speaking

- Content
- Visual aids
  - Slides
  - Schematics
  - Data figures
- Delivery

Presentations should have a main thesis

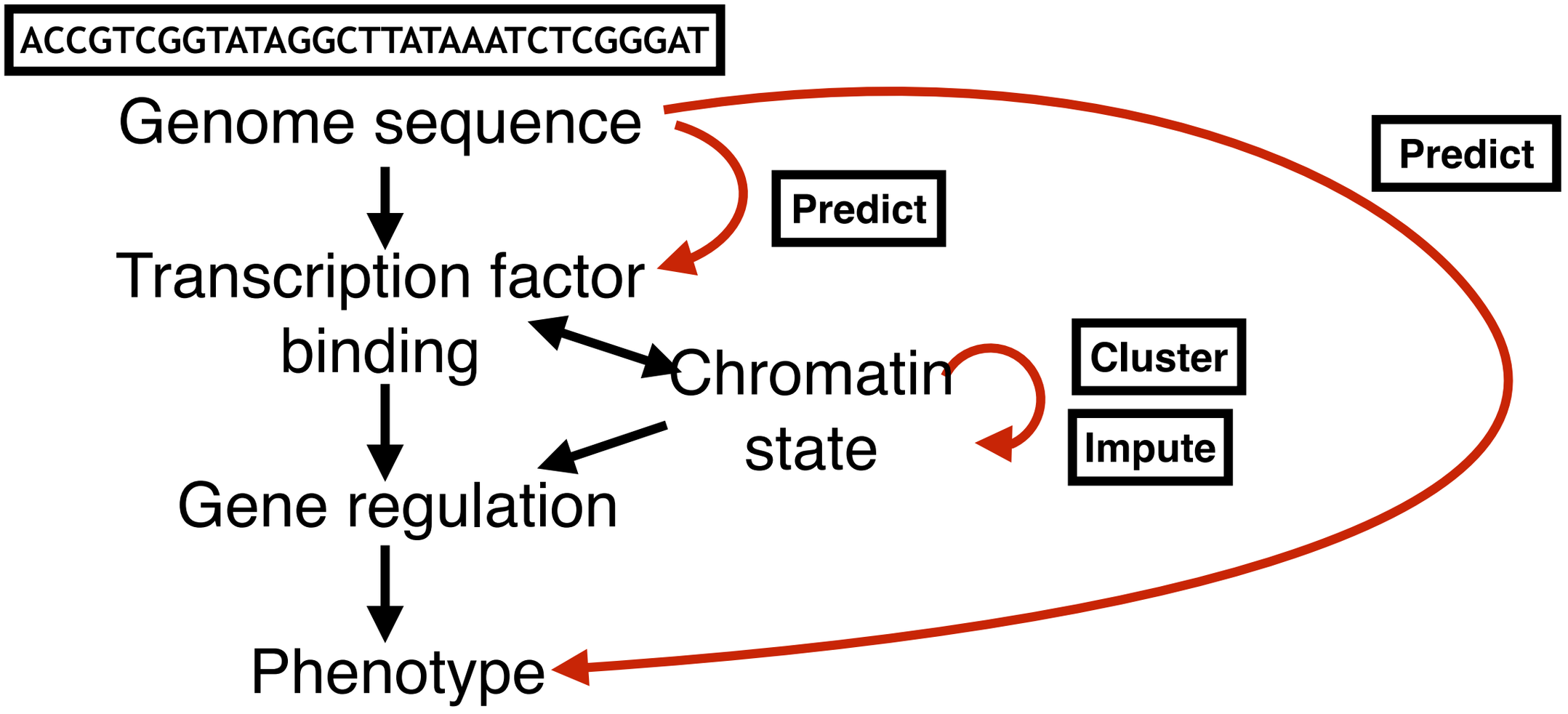
Err on the side of too much background

Re-engage the audience using a home slide

Outline

- Introduction
- Methods
- Results
- Conclusion

Machine learning methods for the genotype-phenotype relationship, gene regulation and epigenomics



Text should be in full sentences

Text should be in full sentences

"Method A crash rate too high"

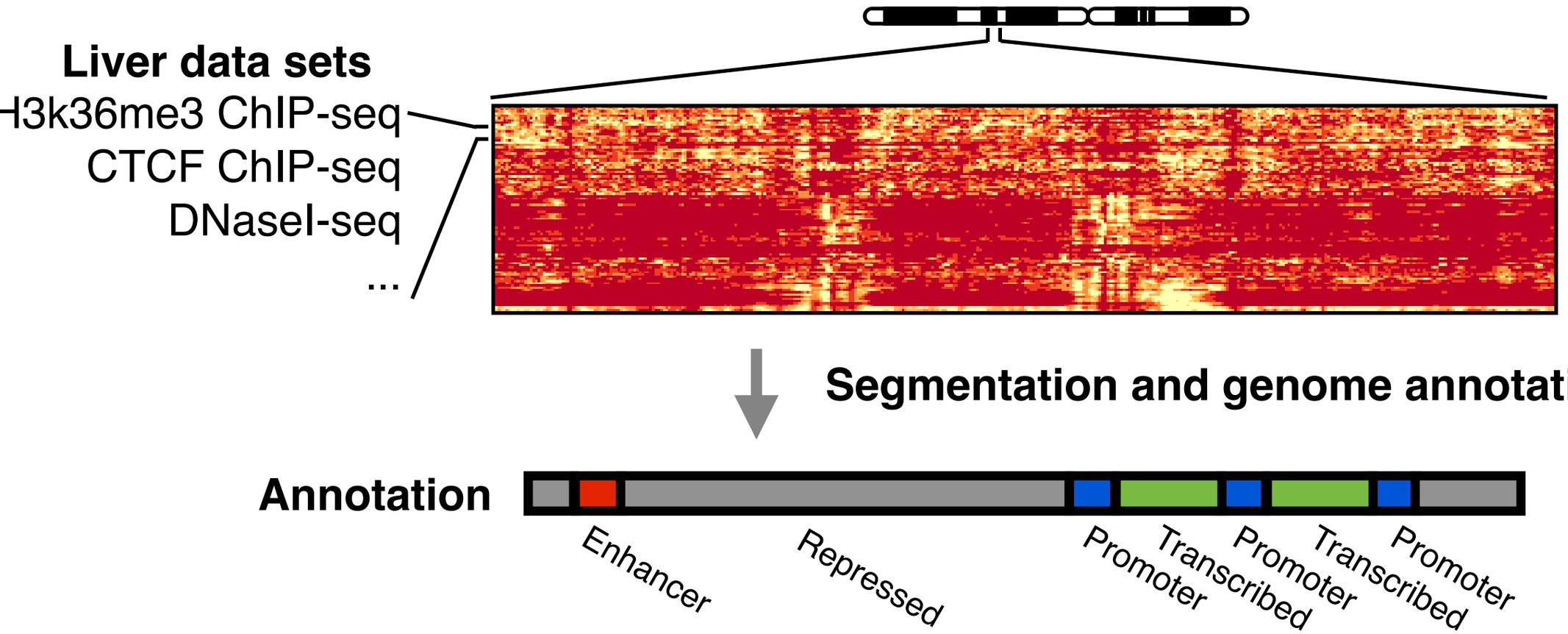
- Slide titles: Use a full sentence explaining the main point of the slide.
- Notation, acronyms: Re-introduce every time you use it.
- Animate in each slide element as you present it.
- Make every slide element legible (font size 18+)

Reduce text by translating it to schematics

Reduce text by translating it to schematics

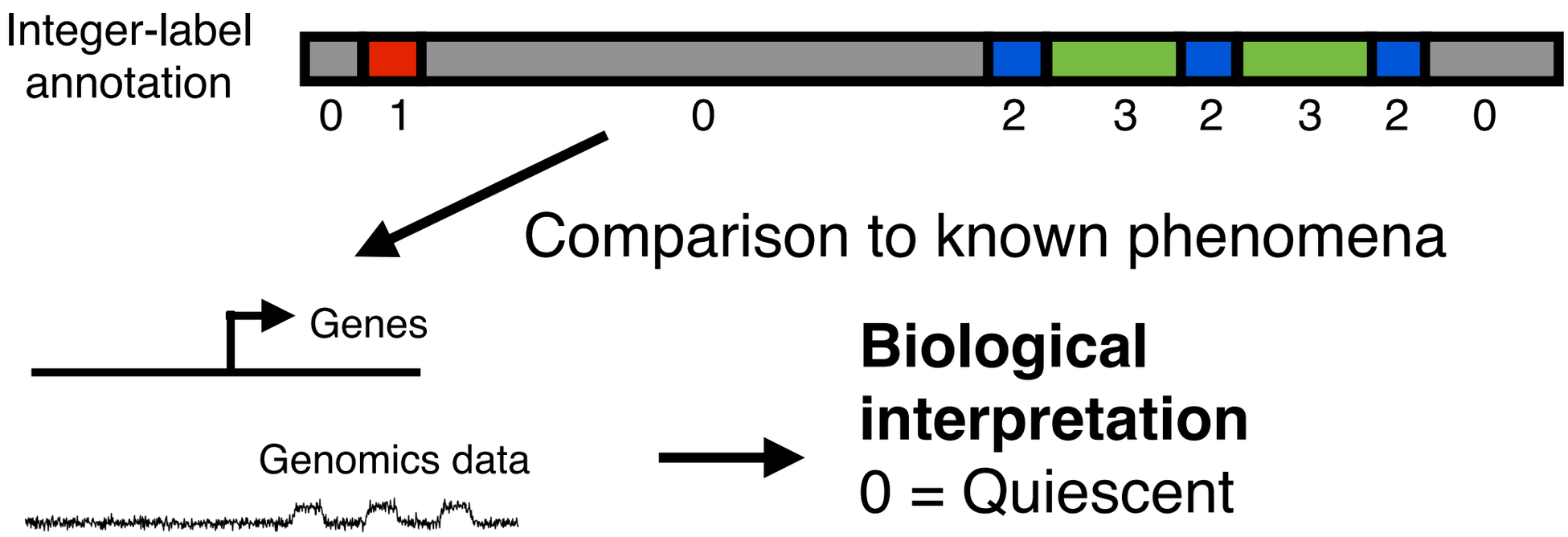
A class of methods known as *semi-automated genome annotation* (SAGA) algorithms are widely used to perform such integrative modeling of diverse genomics data sets. These algorithms take as input a collection of genomics data sets from a particular cell type. They output (1) a set of integer *state labels*, such that each state label putatively corresponds to a type of genomic activity (such as active promoter, active transcription or repressed region), and (2) a partition of the genome and annotation of each genomic segment with one state label. These methods are "semi-automated" because a human performs a functional interpretation of the state labels after the annotation process. In this interpretation step, the human assigns an *interpretation term* to each state label, such as "Promoter" or "Repressed", indicating its putative function.

Segmentation and genome annotation (SAGA) algorithms partition and label the genome on the basis of genomics data sets



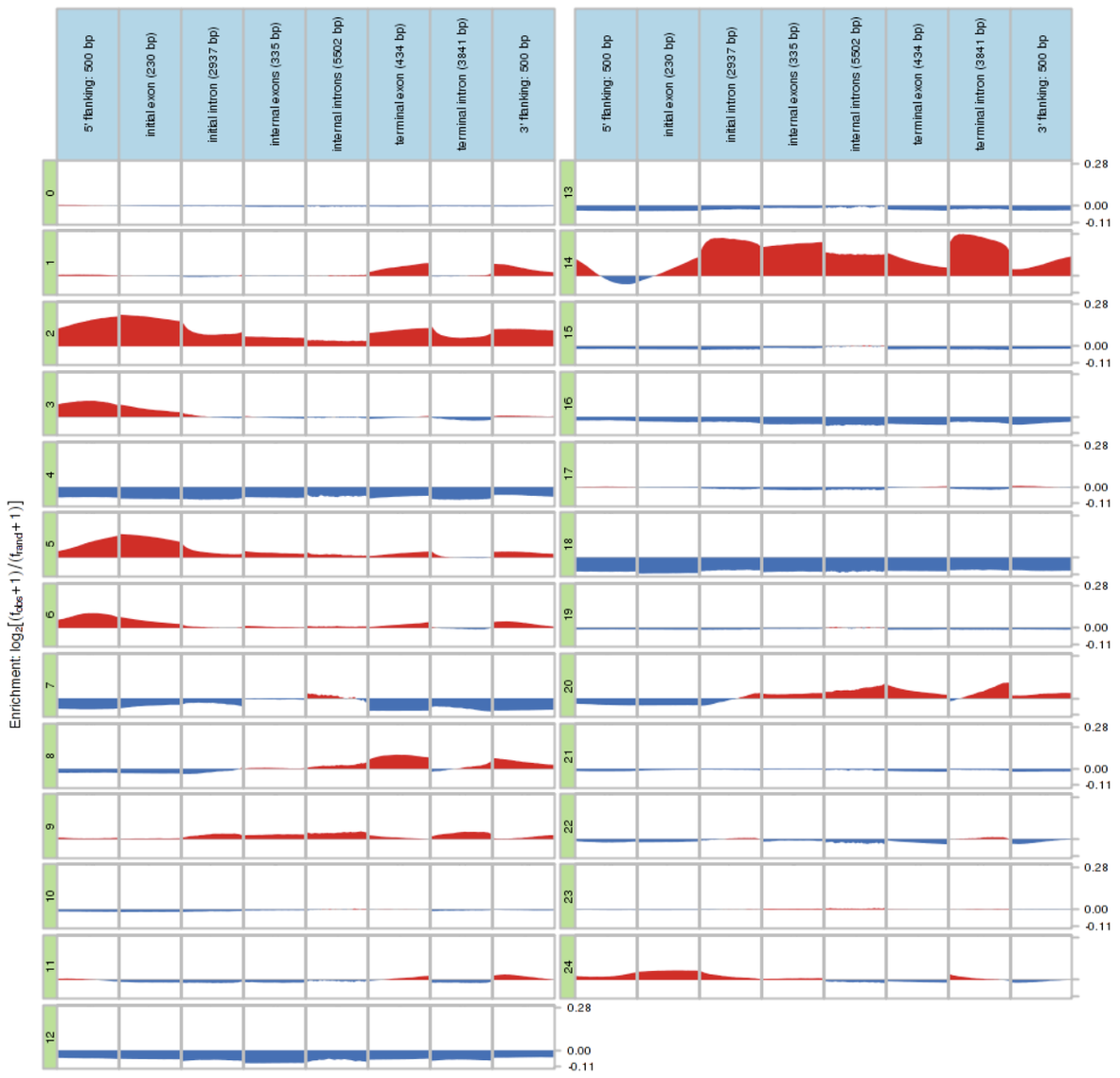
ChromHMM: Ernst, J. and Kellis, M. *Nature Biotechnology*, 2010
Segway: Hoffman, M et al. *Nature Methods*, 2012

What biological phenomenon does each unsupervised label correspond to?

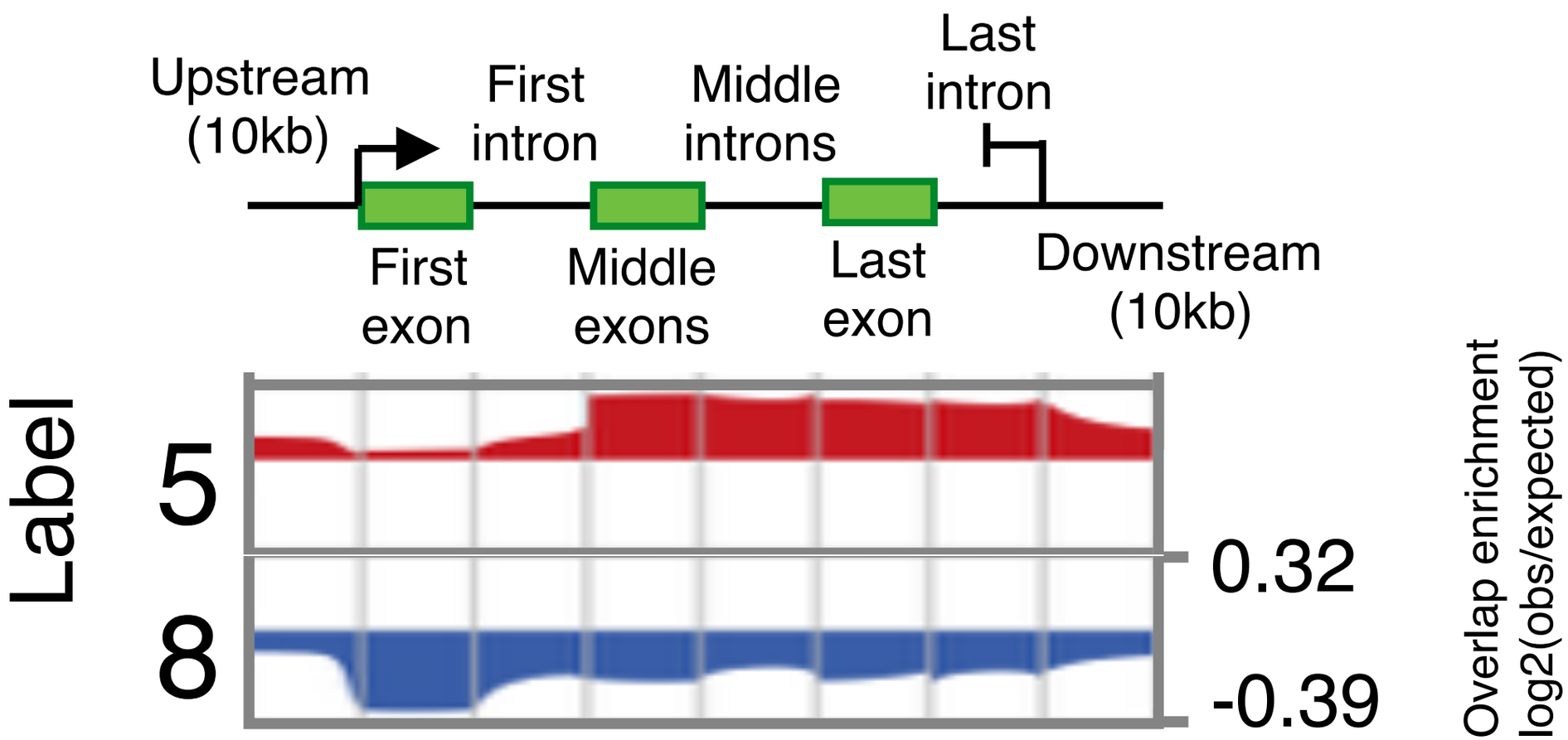


Figures should make a point

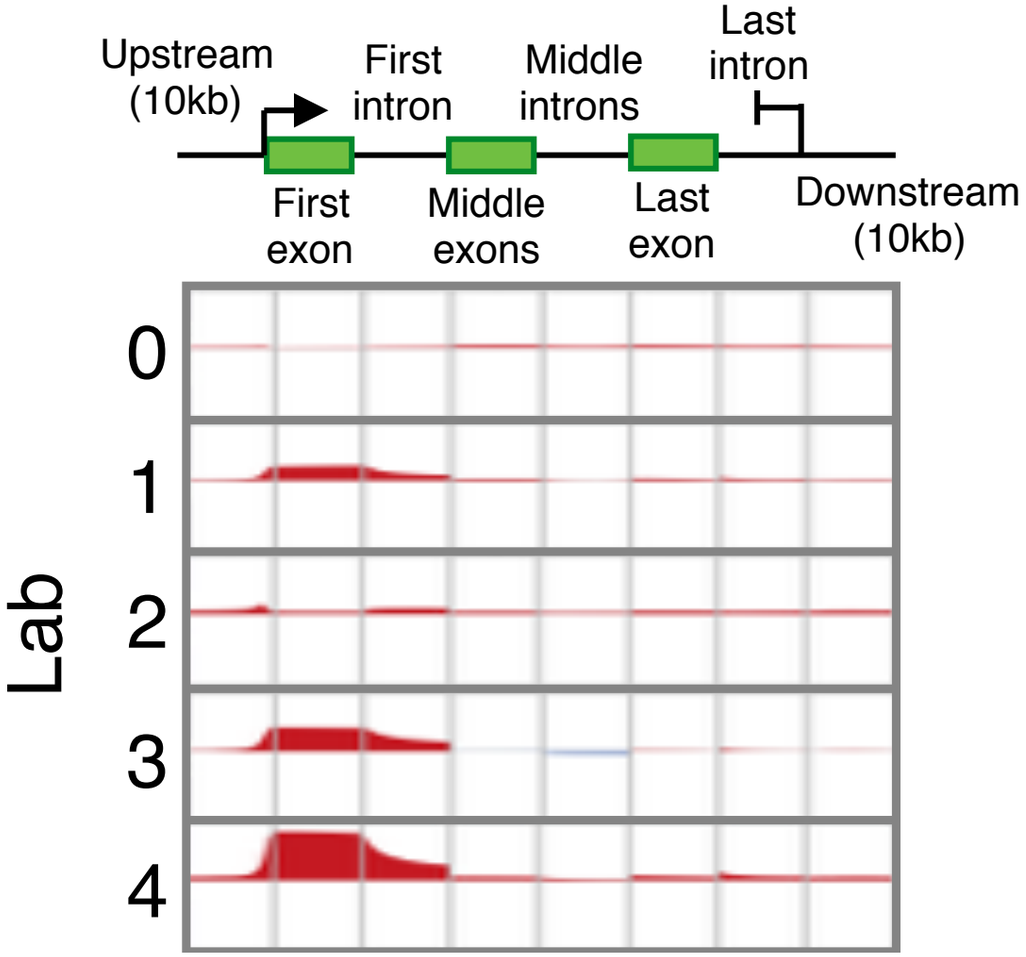
Walk the audience through each figure

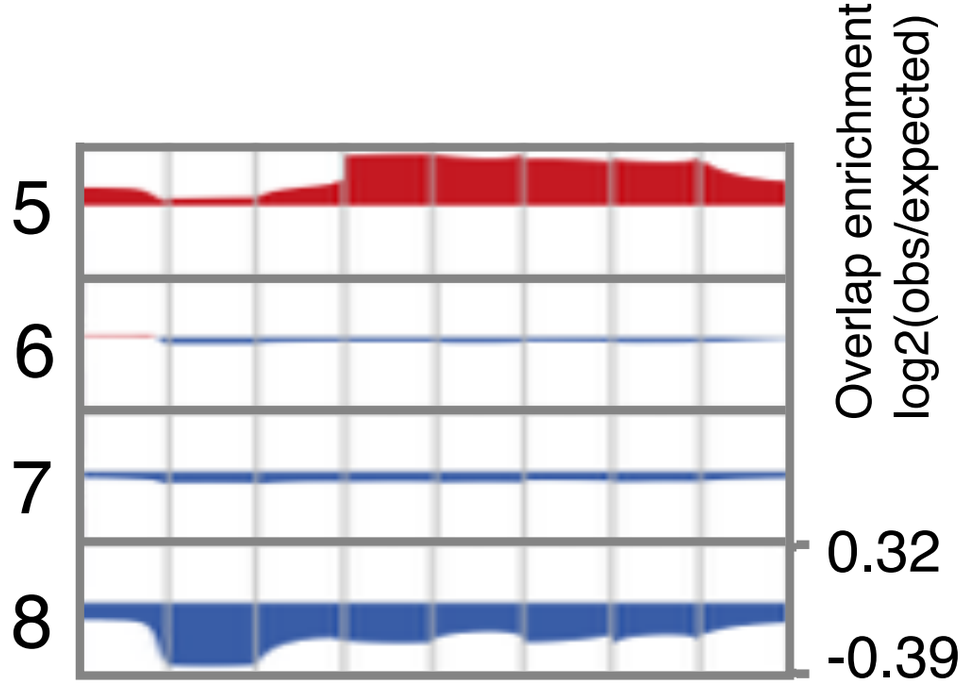


What types of genomic elements did the algorithm find?

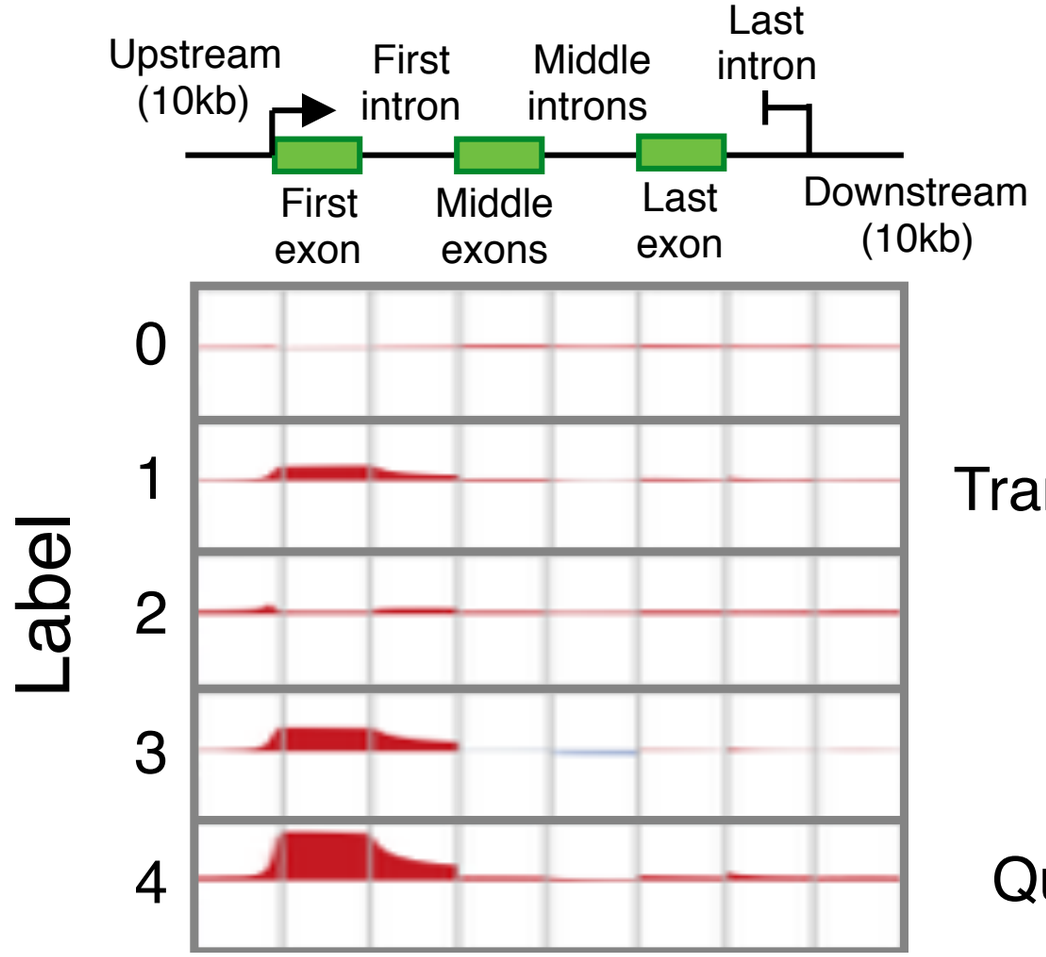


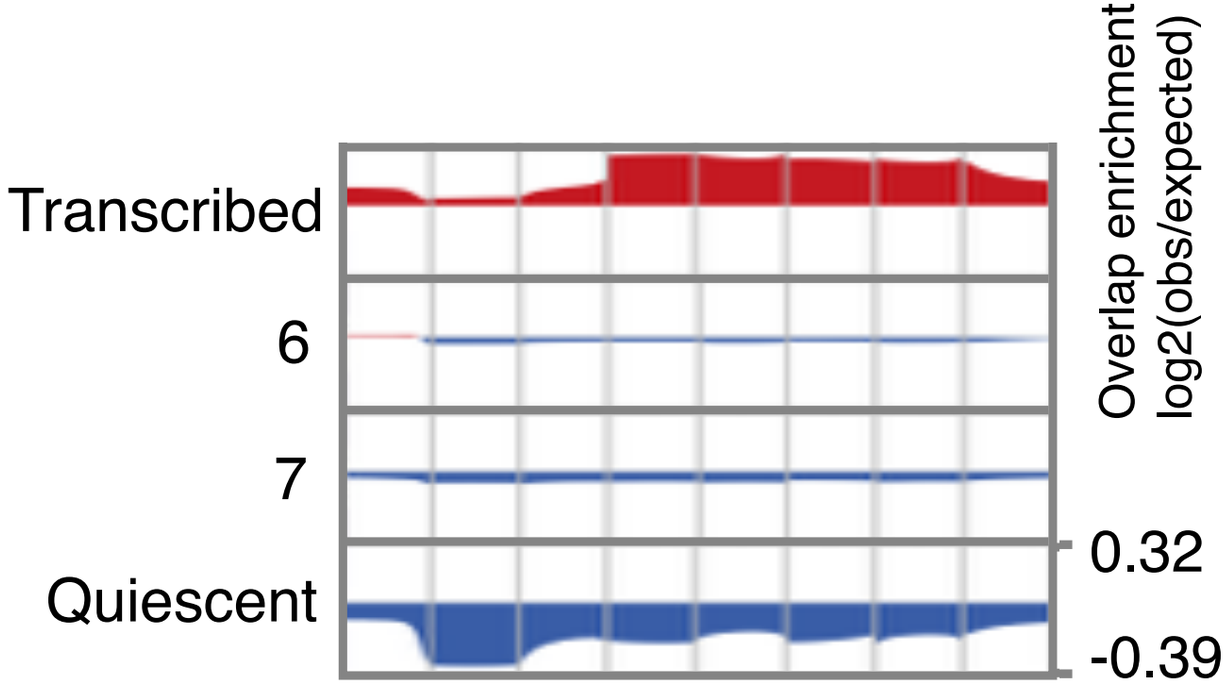
What types of genomic elements did the algorithm find?





What types of genomic elements did the algorithm find?





Practice your delivery

Administrivia