What’s in a good representation?

John Tan Chong Min

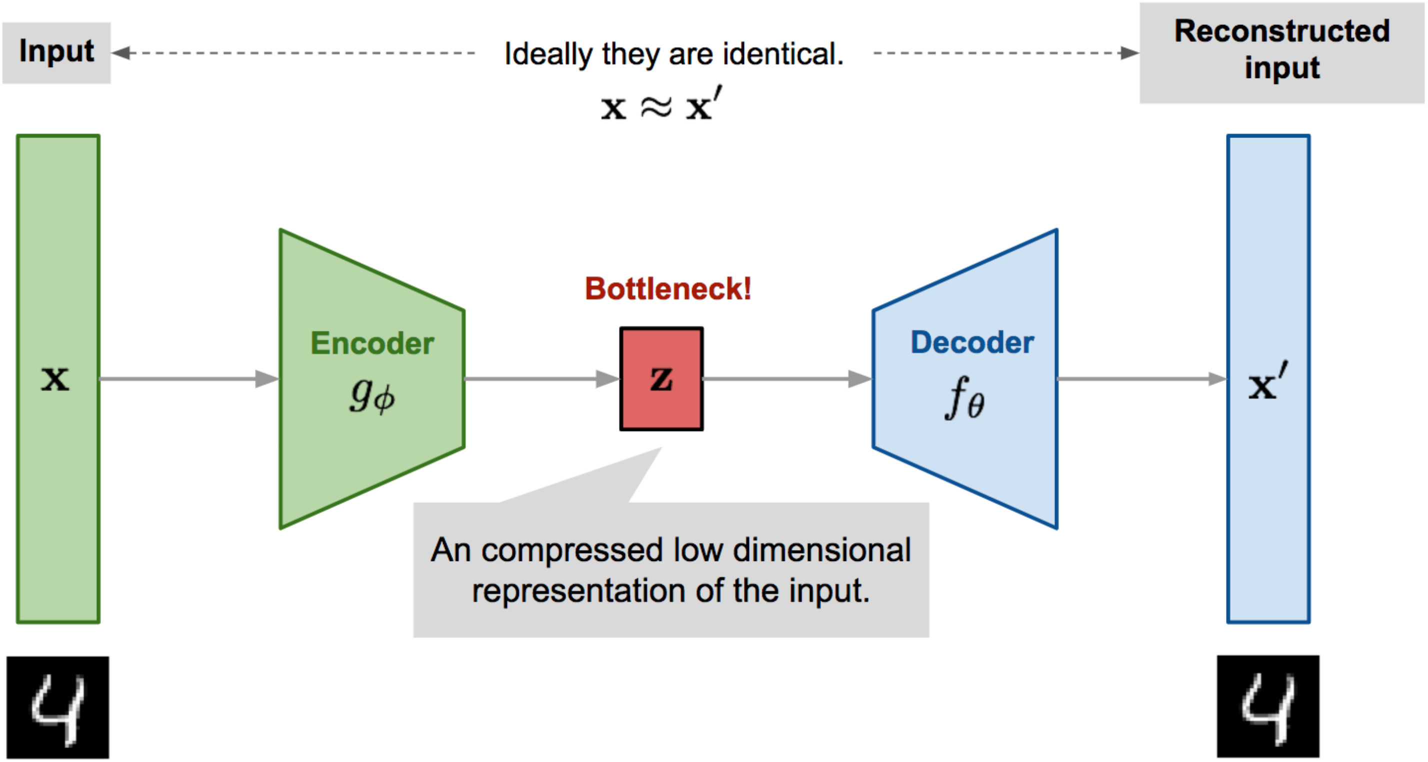
Aim

- Find a way to encode a suitable representation to perform decision making

- Such representation can also be how we store memories for use in the future

Autoencoders: Representation via Reconstruction

- Prioritises output clarity – may not disentangle well in latent space



Input

Ideally they are identical.

$\mathbf{x} \approx \mathbf{x}'$

Reconstructed input

Bottleneck!

An compressed low dimensional representation of the input.

Do you need to predict everything? (Recap)

- Some things in input space are not important to understand for your goals

Transformers: Representation via Prediction

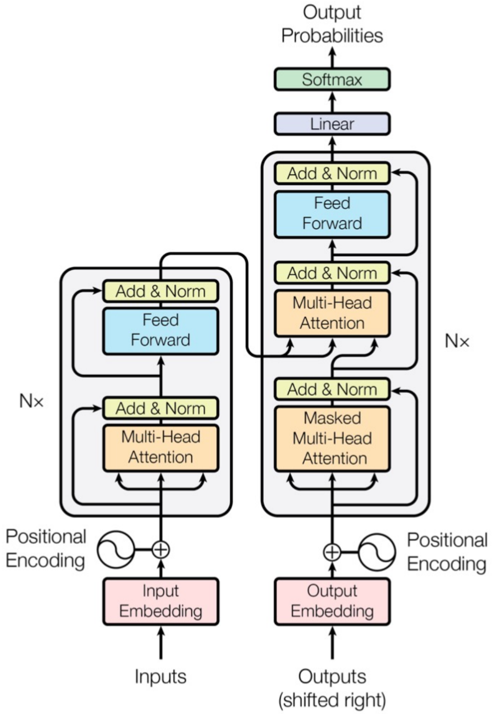
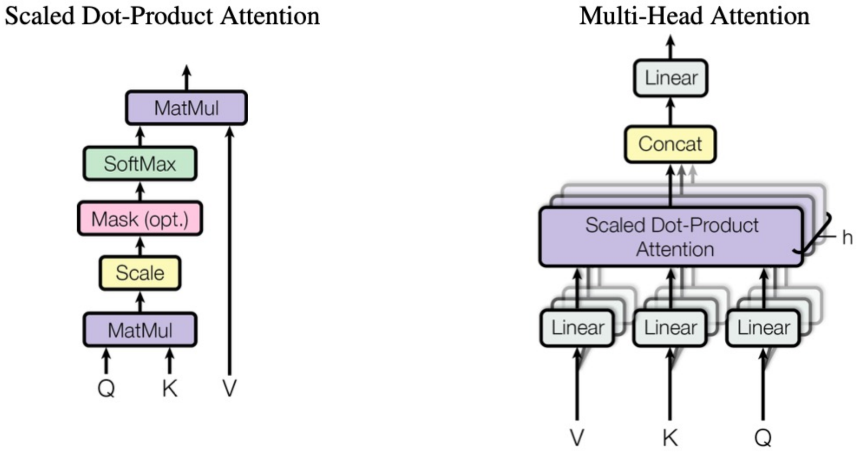


Figure 1: The Transformer - model architecture.


Scaled Dot-Product Attention

Multi-Head Attention

Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

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

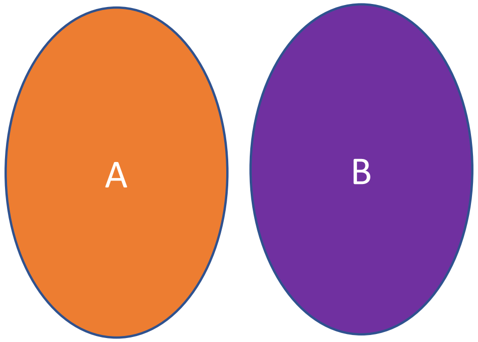
Taken from: Attention is all you need. Vaswani et al. (2017)

Large Self-Supervised Learning

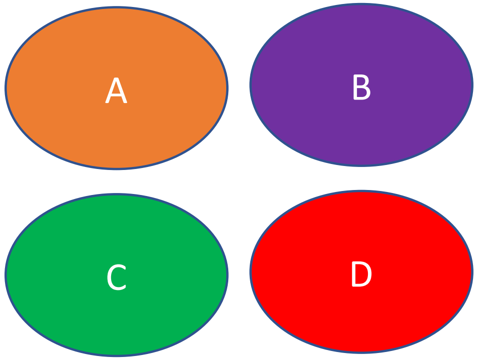
- Self-supervised learning helps to learn better manifolds across large data

- Can work zero-shot on a new sample

Manifold of having only 2 classes

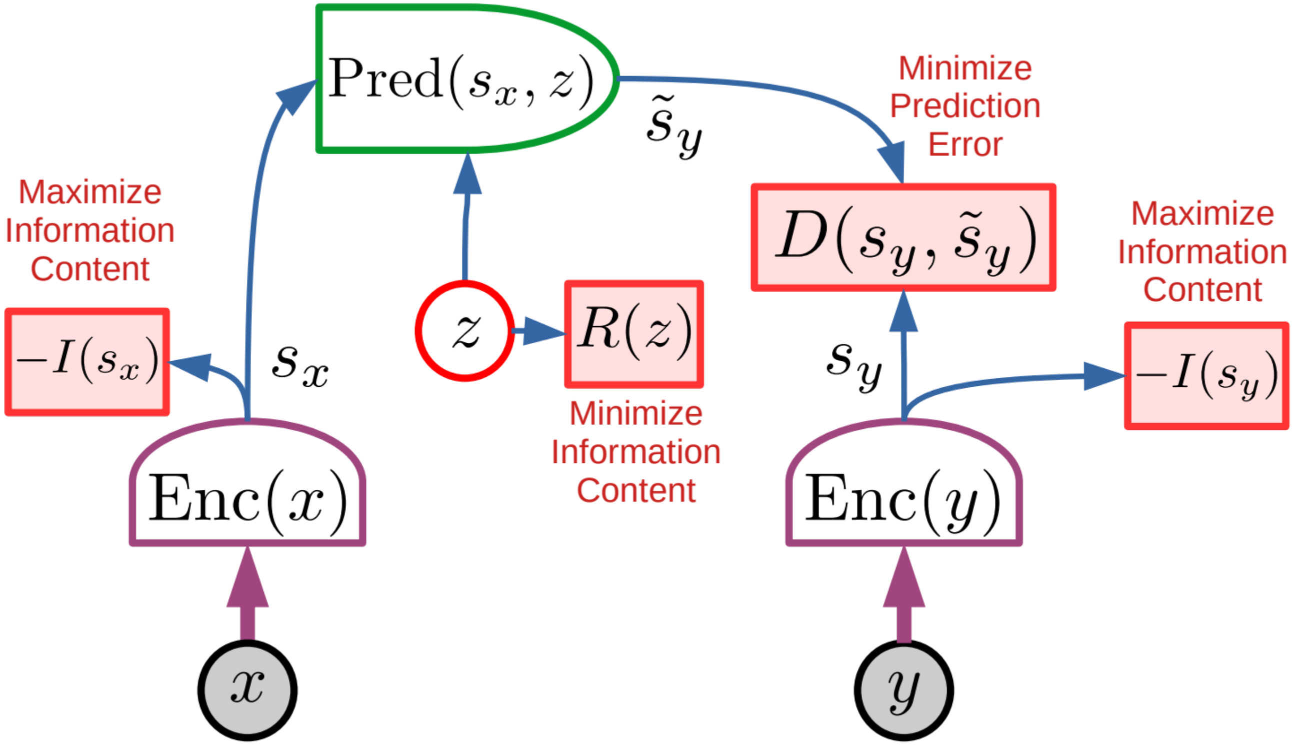


Manifold of having 4 classes



JEPA - Only use whatever is necessary to predict

- Prediction is done in latent space



$\mathrm{Pred}(s_x, z)$

$\tilde{s}_y$

Minimize
Prediction
Error

$D(s_y, \tilde{s}_y)$

Maximize
Information
Content

$-I(s_x)$

s_x

z

$R(z)$

Minimize
Information
Content

s_y

Maximize
Information
Content

$-I(s_y)$

$\mathrm{Enc}(x)$

$\mathrm{Enc}(y)$

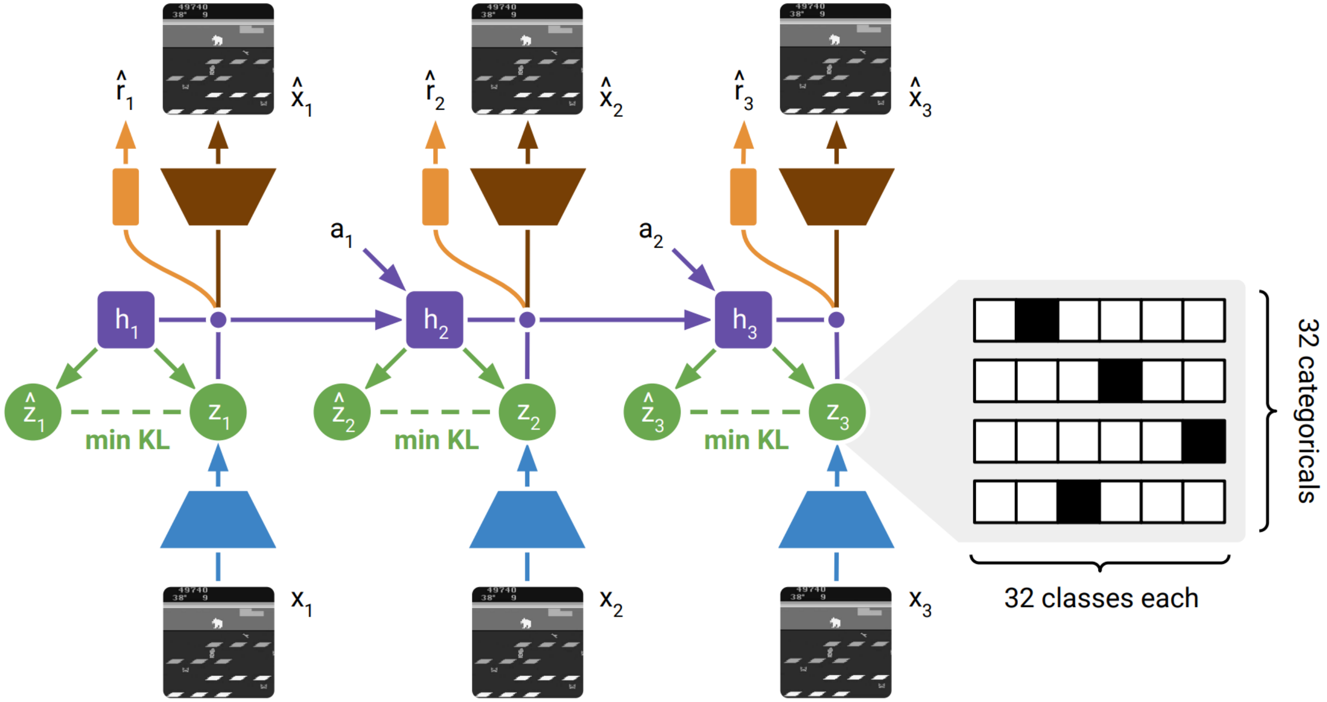
x

y

A Path towards Autonomous Machine Intelligence. Yann LeCun. 2022.

World Modelling

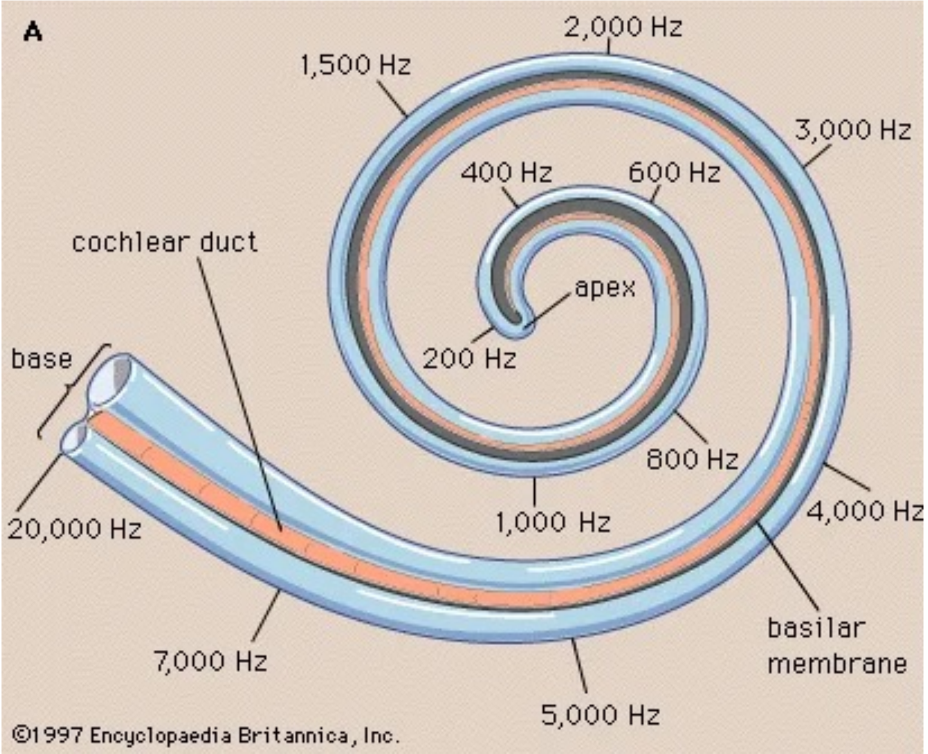
- Use hidden representations for prediction
- Latent space is **discrete** using latent space of categorical variables



Mastering Atari with Discrete World Models. Hafner et al. 2022.

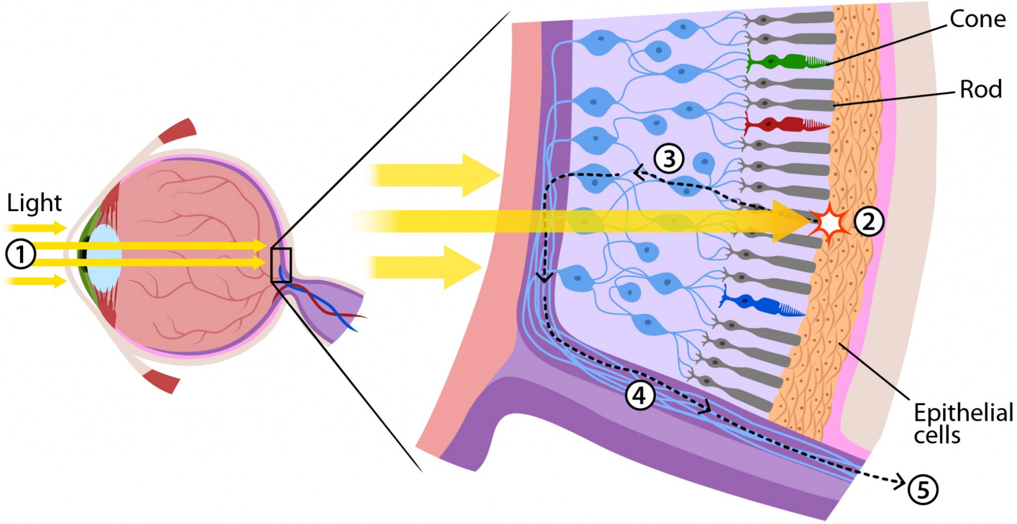
Natural Fixed Biases: Faster learning by constraints

- Sound: frequency in cilia

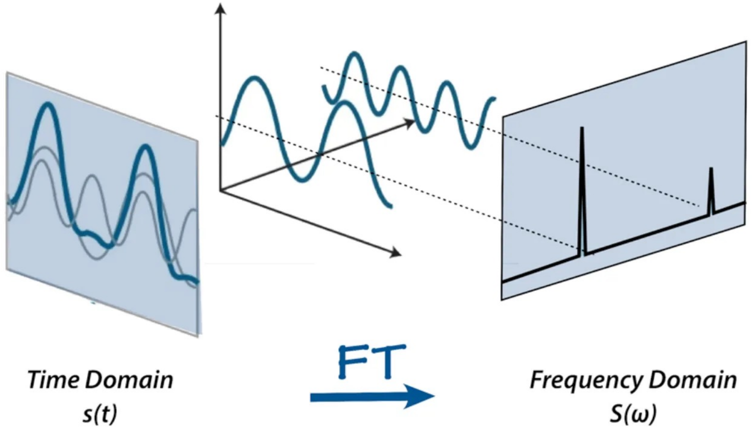


©1997 Encyclopaedia Britannica, Inc.

- Vision:
  - Local patches
  - Cones for Red, Green, Blue
  - Rods for Black and White

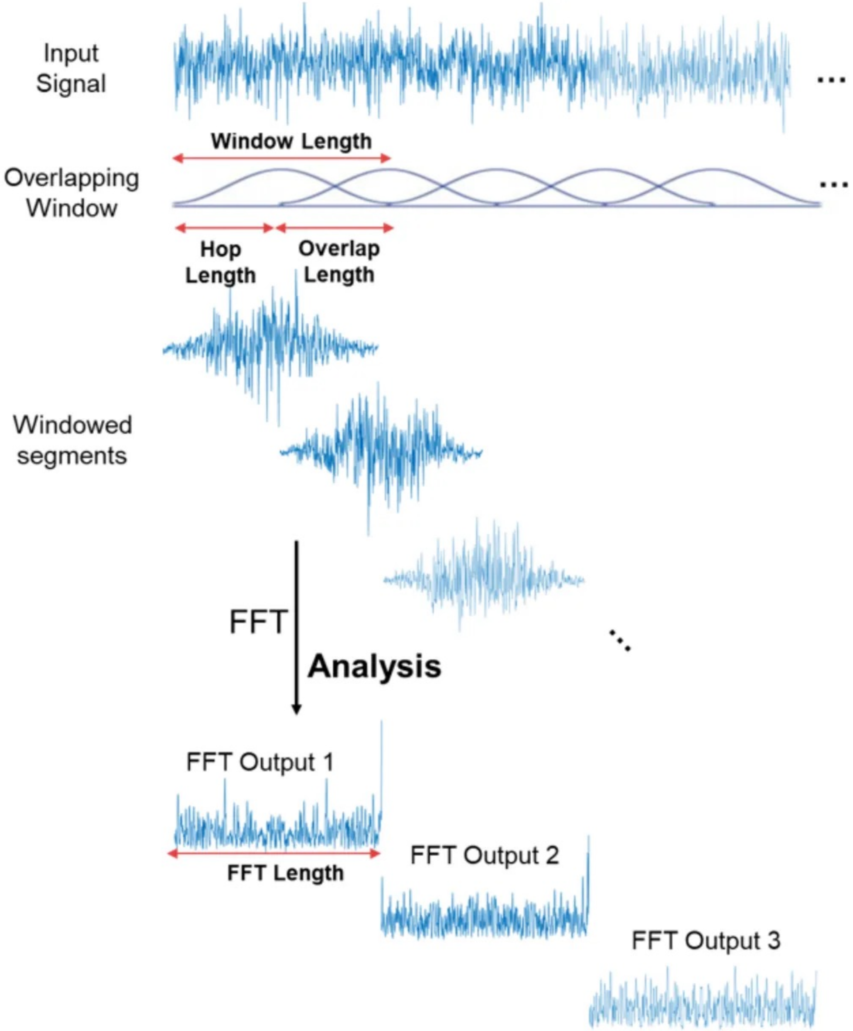


Audio – Freq Modelling



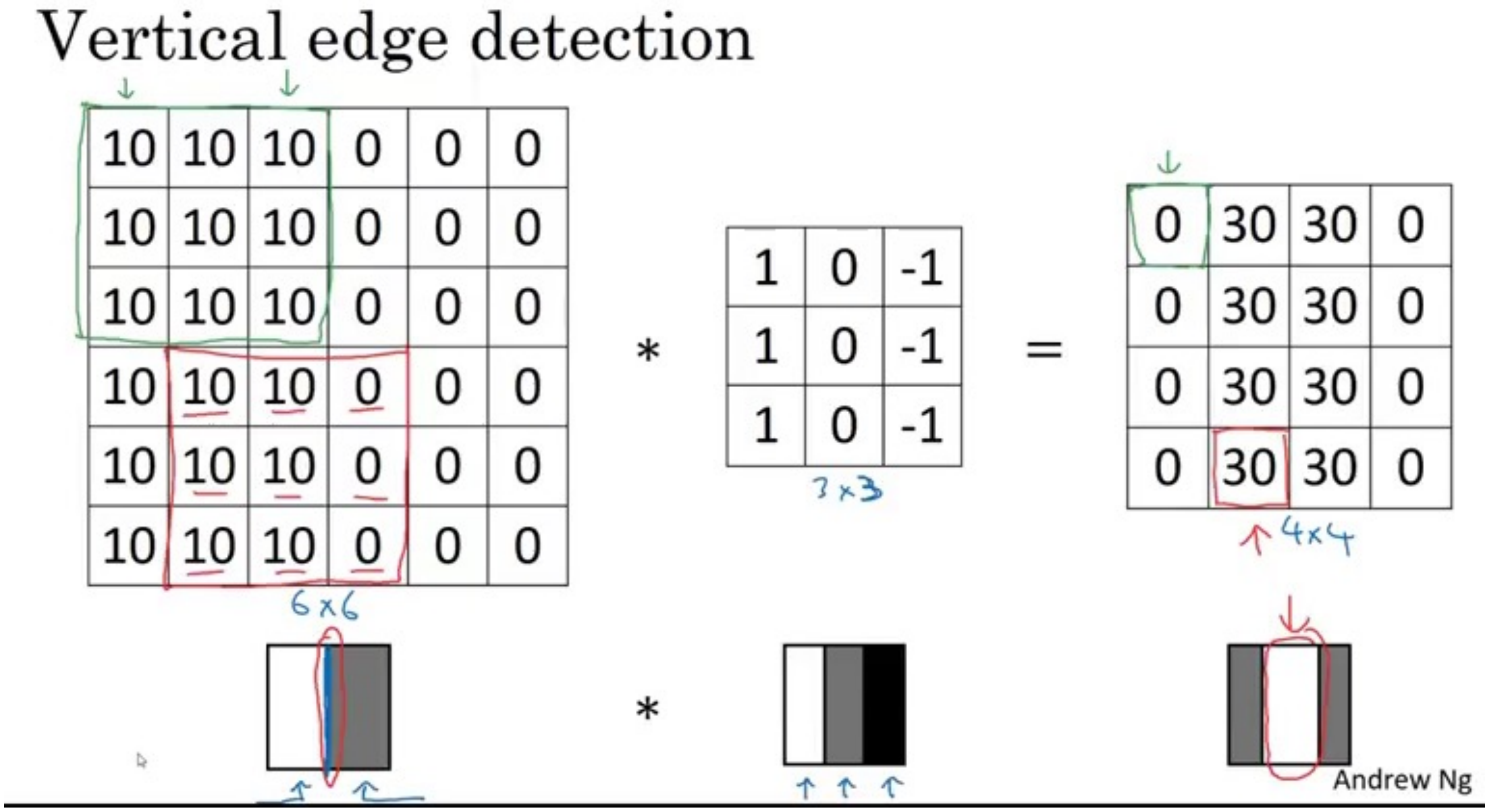
Audio signal converted into frequencies

Images taken from: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53



Use overlapping window to model waveform over time

Vision – Pixel Proximity

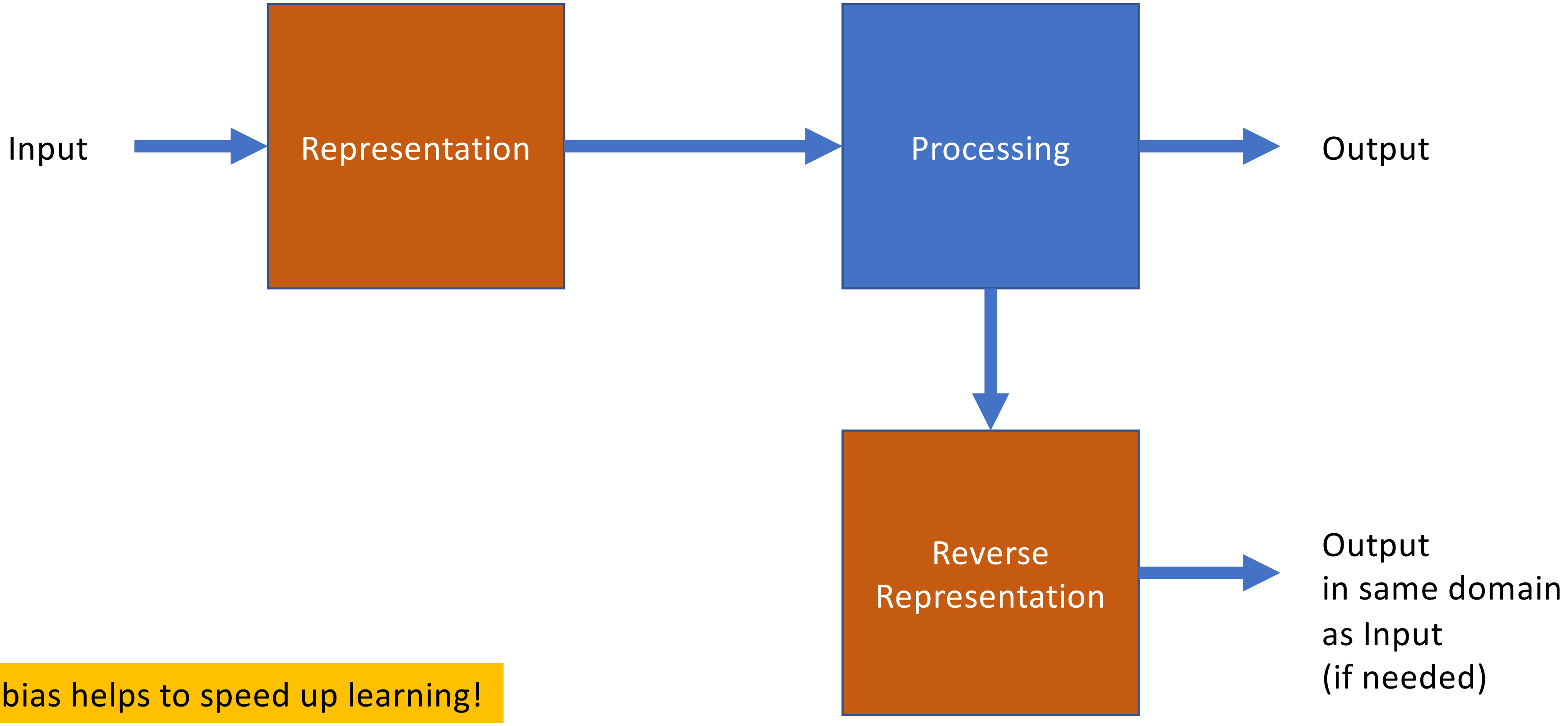


Vertical edge detection

$$\underset{6\times 6}{\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix}} * \underset{3\times 3}{\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}} = \underset{4\times 4}{\begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}}$$

Andrew Ng

Information Pipeline – Bias for Representation (Recap)



Input

Representation

Processing

Output

Reverse Representation

Output
in same domain
as Input
(if needed)

Fixed bias helps to speed up learning!

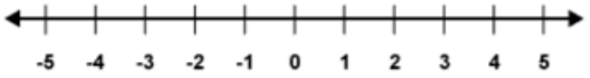
Vector Representation

Initial Reference + Movement

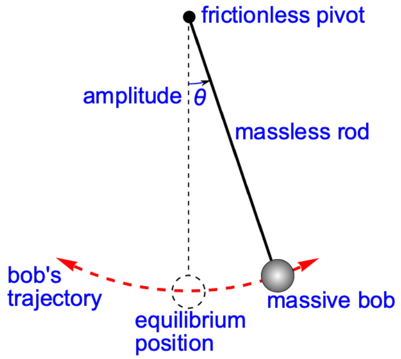
Could aid with generalization

Vector Representation of memory (my view)

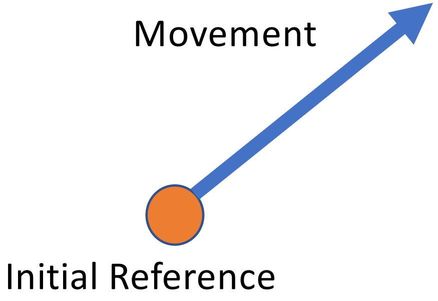
- Initial Reference + Movement



- Naturally represented by vectors



- Fits nicely with concepts
    - **Science:** Initial position/state of object + movement
    - **Movement:** Starting position + action
    - **Geography:** Start from one country and move to another
    - **History:** Starting time and moving up or down timeline
    - **Mathematics:** Starting number and moving up or down number line
    - **Reinforcement Learning:** Start state, action, end state



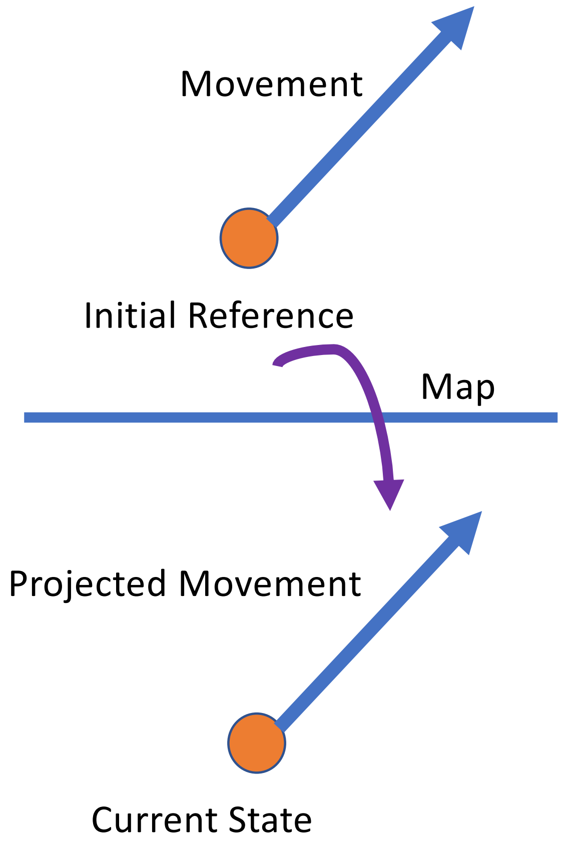
How we generalize from memory (my view)

- When we retrieve memories, the initial reference can be adjusted to suit the situation

- That way, the movement obtained can be applied to the new situation to get the desired outcome

- We can probably simulate various outcomes by retrieving from memory, applying the movement to current situation

- Current vectors don't disentangle between reference and movement – may be required for better memory referencing?



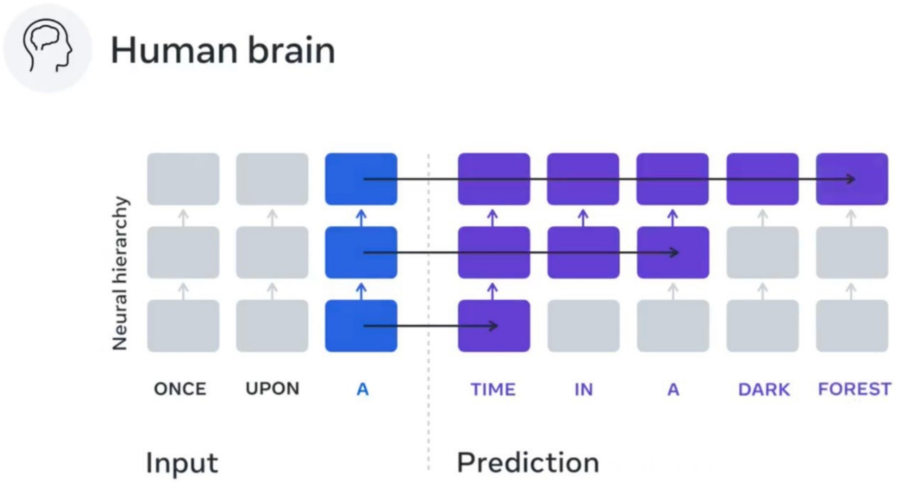
The next level

Hierarchical Prediction

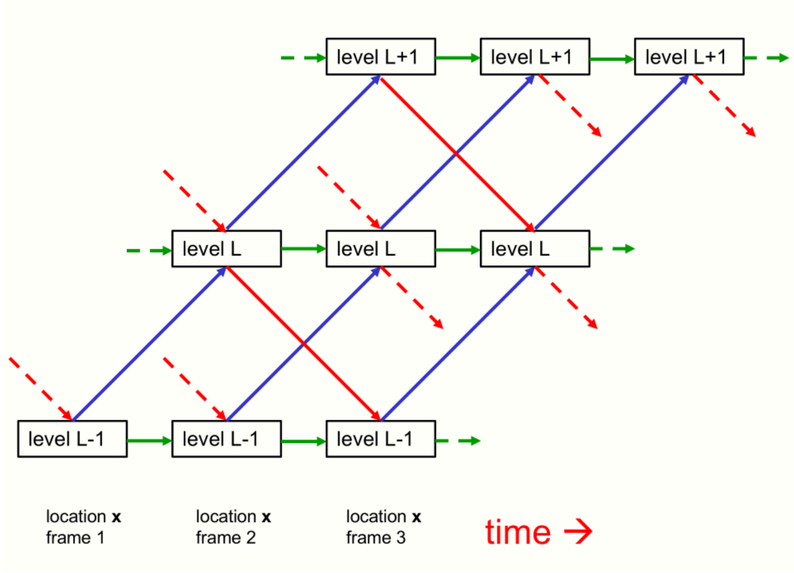
Top level forms basis for interpretation of the bottom

Hierarchical Prediction is the future

- Hierarchical prediction of more than just next token, but broader prediction at higher levels
- Higher level prediction can be more abstract and less detailed than lower levels



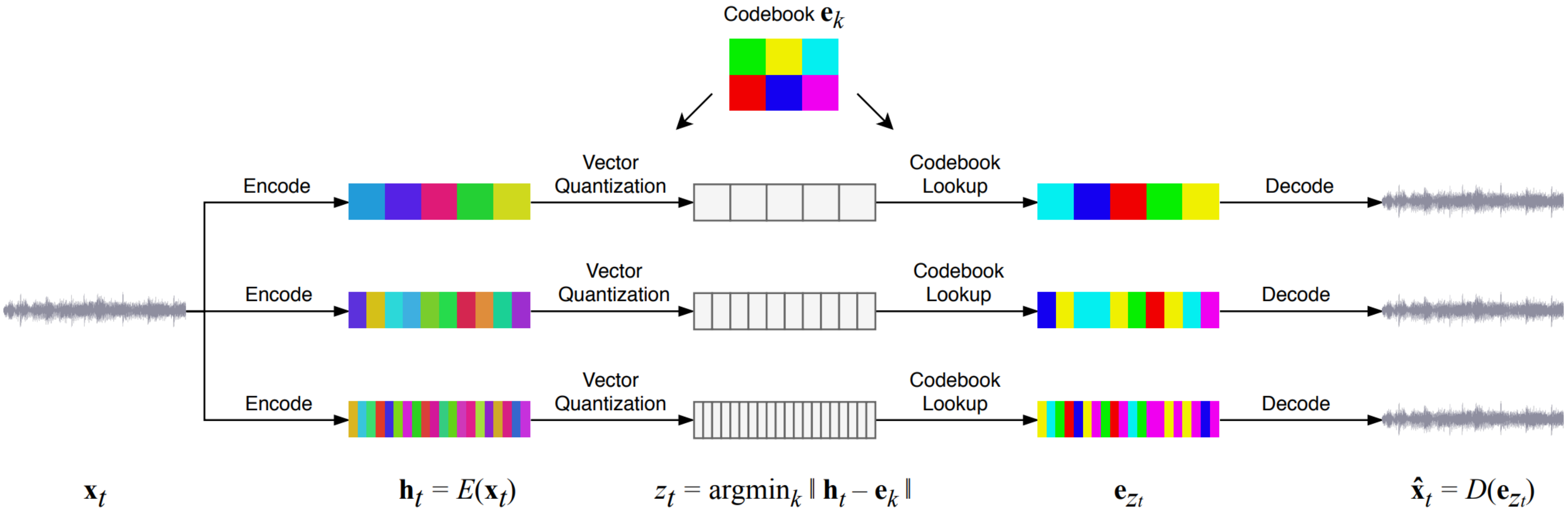
Evidence of a predictive coding hierarchy in the human brain listening to speech. Caucheteux. 2022. Nature Human Behaviour.



How to represent part-whole hierarchies in a neural network. Hinton. 2021.

Hierarchical Prediction – Jukebox (OpenAI)

- Hierarchical Quantization of sound from coarse-grained input to fine-grained layers
- Codebook entries are **finite**



Codebook $\mathbf{e}_k$

Encode

Vector Quantization

Codebook Lookup

Decode

Encode

Vector Quantization

Codebook Lookup

Decode

Encode

Vector Quantization

Codebook Lookup

Decode

$\mathbf{x}_t$

$\mathbf{h}_t = E(\mathbf{x}_t)$

$z_t = \text{argmin}_k \| \mathbf{h}_t - \mathbf{e}_k \|$

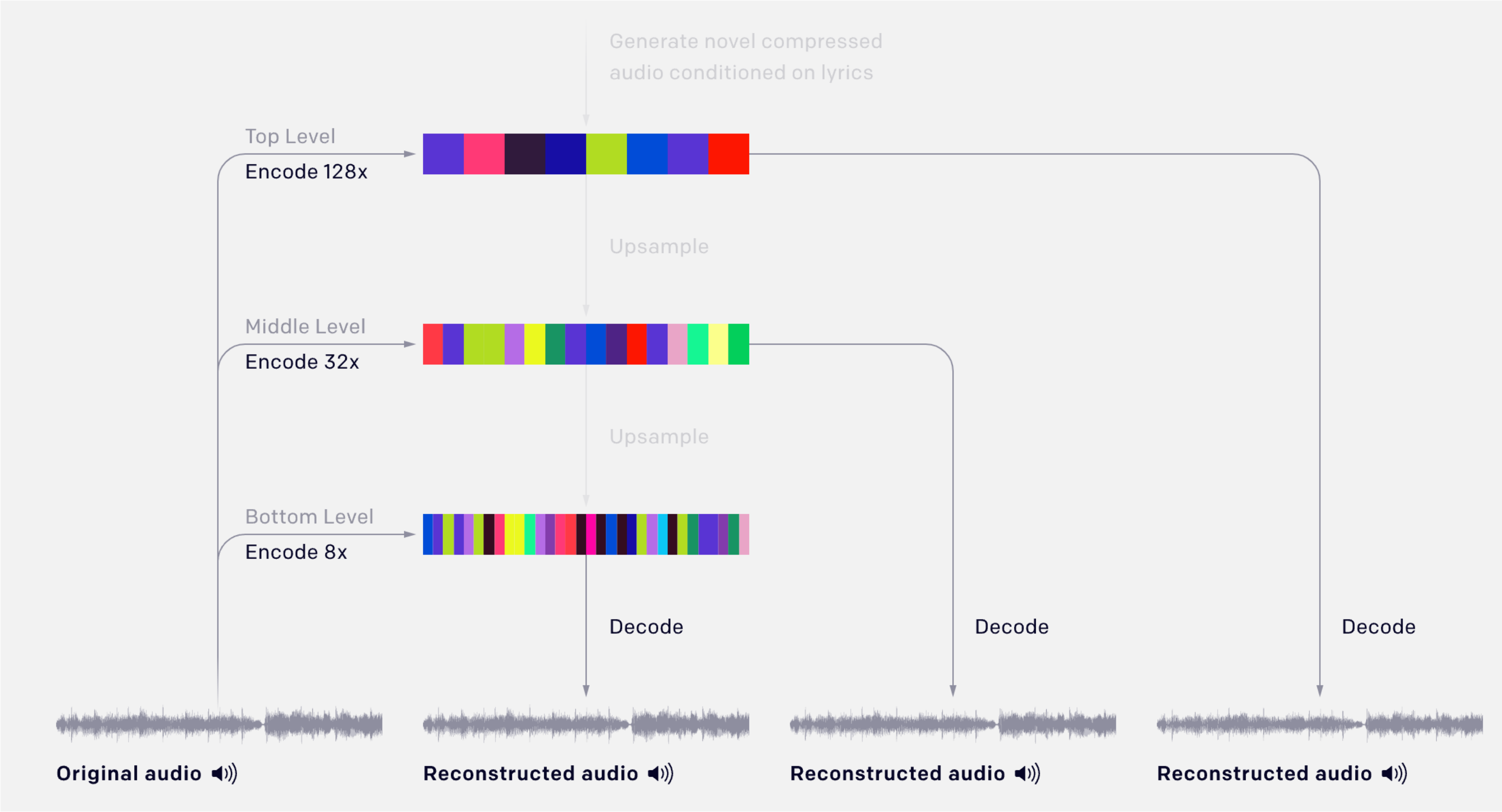
$\mathbf{e}_{z_t}$

$\hat{\mathbf{x}}_t = D(\mathbf{e}_{z_t})$

Jukebox: A Generative Model for Music. Dhariwal et. al. 2020.

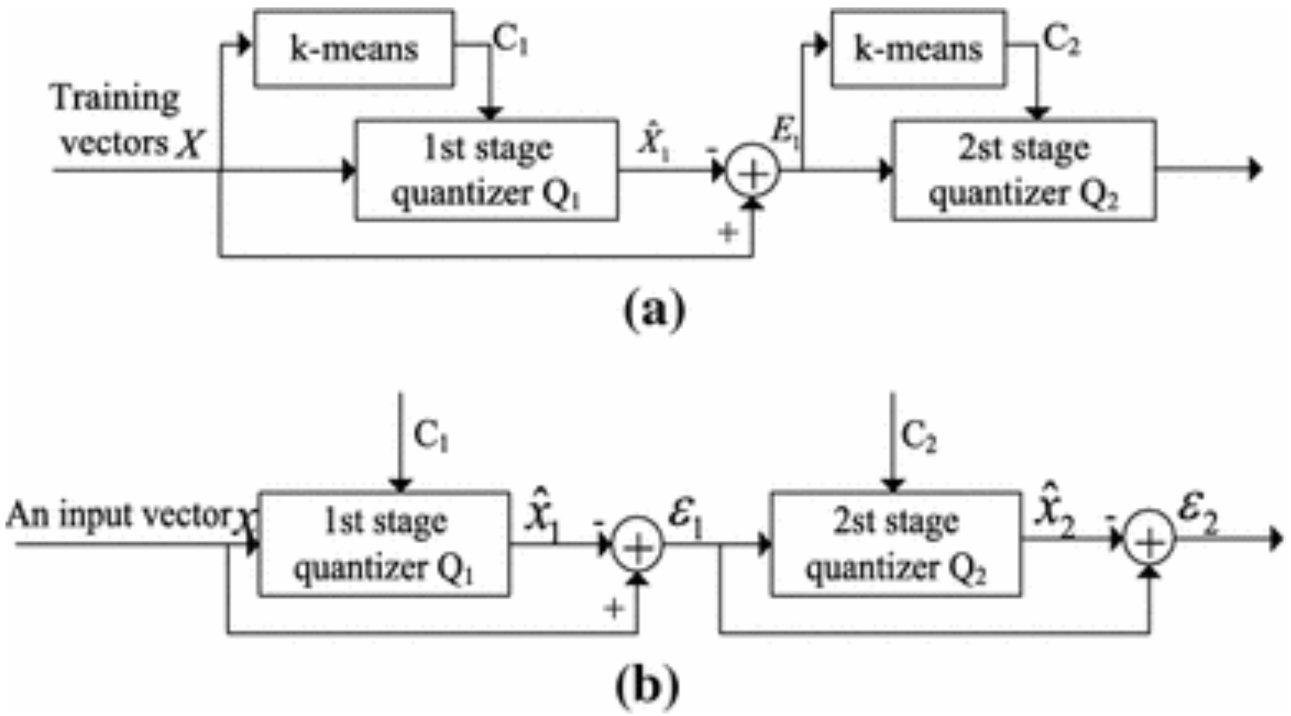
Hierarchical Prediction – Jukebox (OpenAI)

- Conditional generation of coarse-grained input to fine-grained layers



Residual Vector Quantization

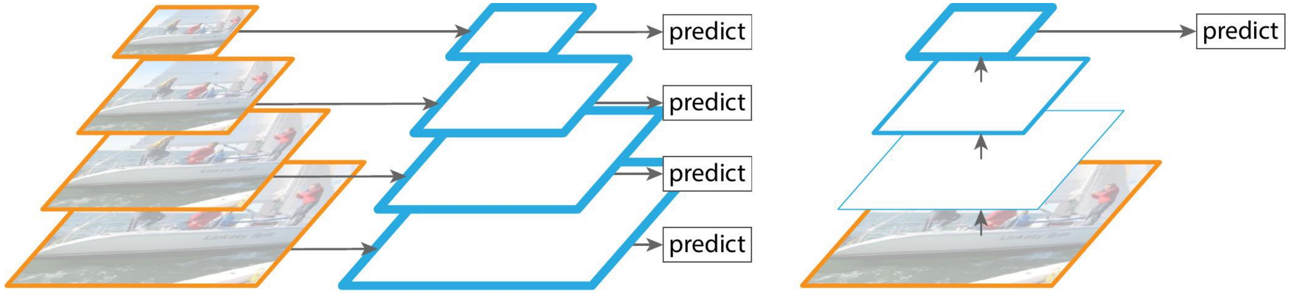
- Hierarchical Quantization from coarse-grained input to fine-grained layers
- Akin to compositionality?



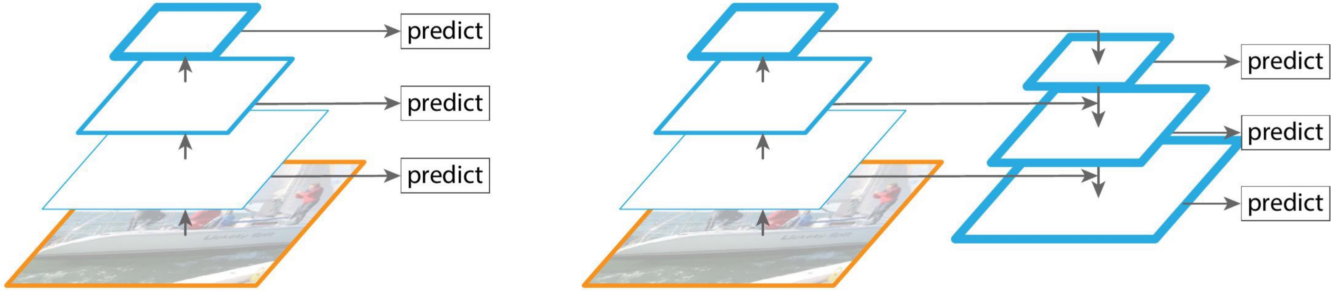
Optimized residual vector quantization for efficient approximate nearest neighbor search. Ai et al. 2015.

Hierarchical Prediction - Feature Pyramid Network

- Hierarchical prediction from coarse-grained image to fine-grained image



(a) Featurized image pyramid (b) Single feature map

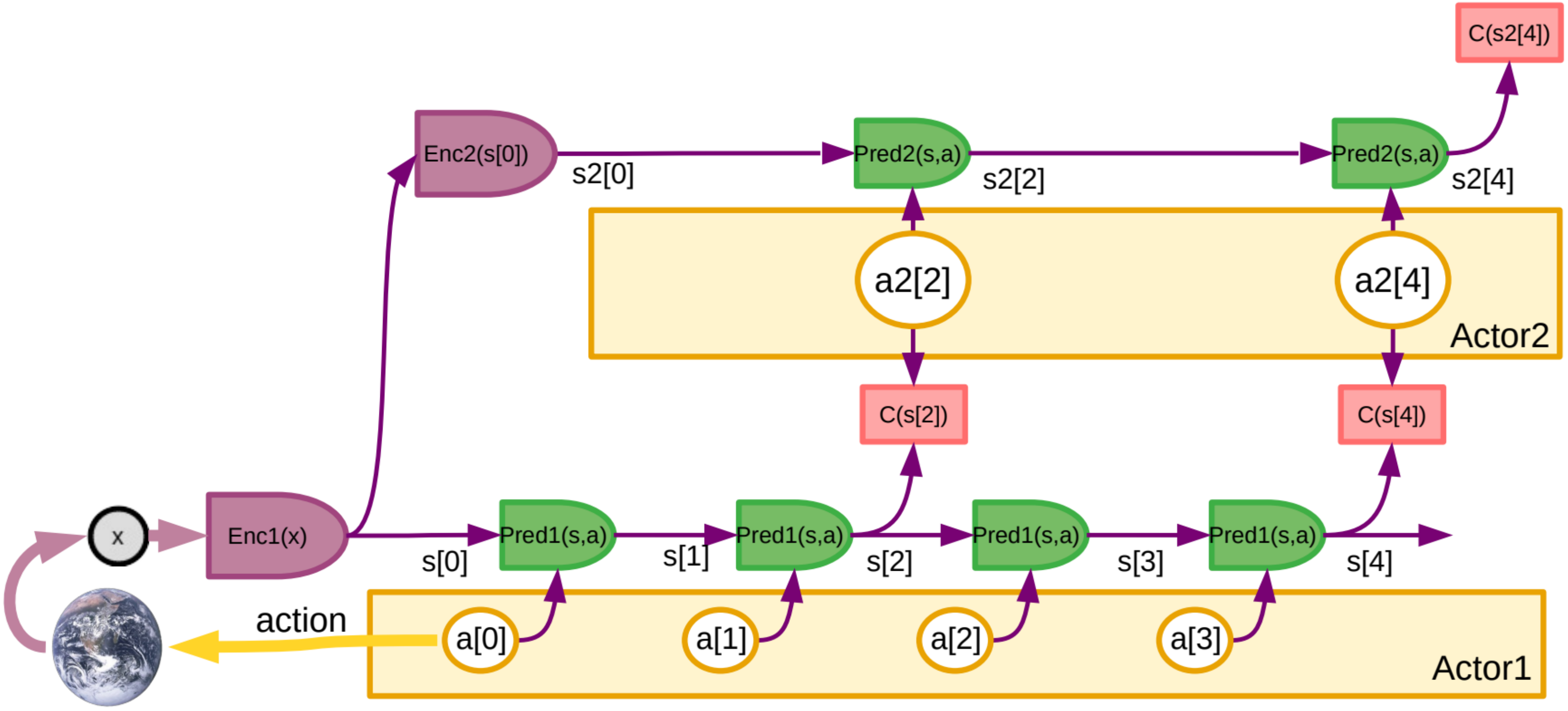


(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

Feature Pyramid Networks for Object Detection. Lin et al. 2017.

Hierarchical JEPA

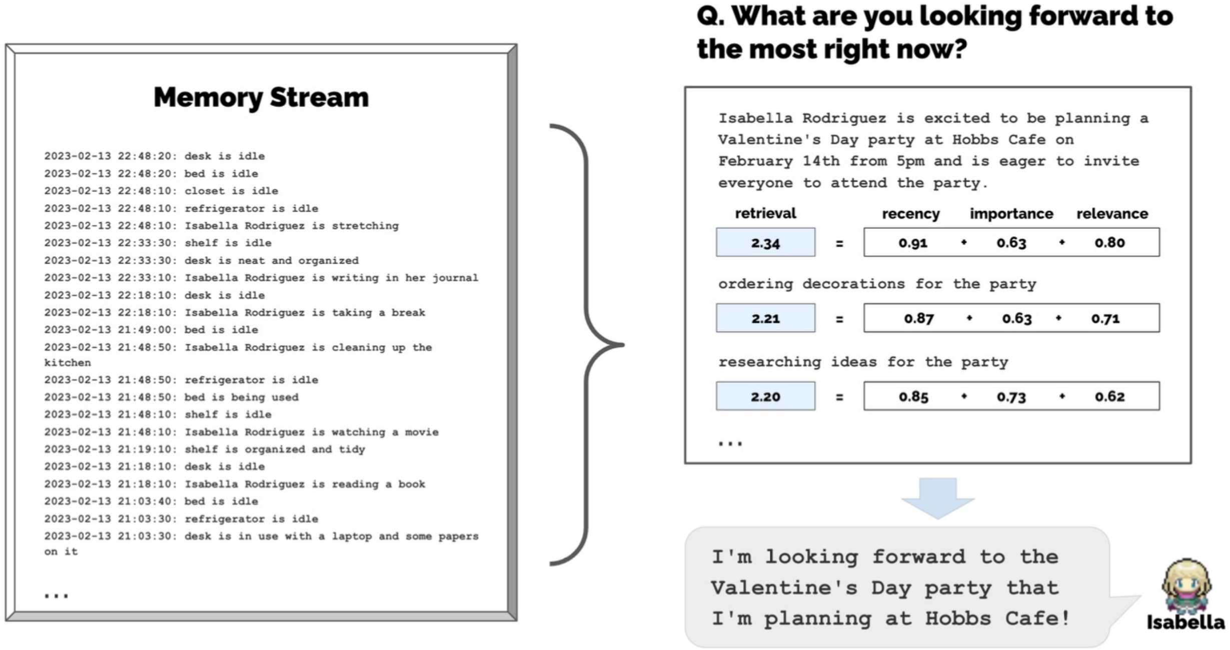
- Hierarchical prediction of actions from the highest level action to the lowest level action



A Path towards Autonomous Machine Intelligence. Yann LeCun. 2022.

Hierarchical Action Prediction

- Hierarchical prompting of actions from broad action to specific actions



Memory Stream

```
2023-02-13 22:48:20: desk is idle
2023-02-13 22:48:20: bed is idle
2023-02-13 22:48:10: closet is idle
2023-02-13 22:48:10: refrigerator is idle
2023-02-13 22:48:10: Isabella Rodriguez is stretching
2023-02-13 22:33:30: shelf is idle
2023-02-13 22:33:30: desk is neat and organized
2023-02-13 22:33:10: Isabella Rodriguez is writing in her journal
2023-02-13 22:18:10: desk is idle
2023-02-13 22:18:10: Isabella Rodriguez is taking a break
2023-02-13 21:49:00: bed is idle
2023-02-13 21:48:50: Isabella Rodriguez is cleaning up the
kitchen
2023-02-13 21:48:50: refrigerator is idle
2023-02-13 21:48:50: bed is being used
2023-02-13 21:48:10: shelf is idle
2023-02-13 21:48:10: Isabella Rodriguez is watching a movie
2023-02-13 21:19:10: shelf is organized and tidy
2023-02-13 21:18:10: desk is idle
2023-02-13 21:18:10: Isabella Rodriguez is reading a book
2023-02-13 21:03:40: bed is idle
2023-02-13 21:03:30: refrigerator is idle
2023-02-13 21:03:30: desk is in use with a laptop and some papers
on it

...
```

Q. What are you looking forward to the most right now?

Isabella Rodriguez is excited to be planning a Valentine's Day party at Hobbs Cafe on February 14th from 5pm and is eager to invite everyone to attend the party.

retrieval		recency		importance		relevance
2.34	=	0.91	+	0.63	+	0.80

ordering decorations for the party

retrieval		recency		importance		relevance
2.21	=	0.87	+	0.63	+	0.71

researching ideas for the party

retrieval		recency		importance		relevance
2.20	=	0.85	+	0.73	+	0.62

...

I'm looking forward to the Valentine's Day party that I'm planning at Hobbs Cafe!

Figure 6: The memory stream comprises a large number of observations that are relevant and irrelevant to the agent's current situation. Retrieval identifies a subset of these observations that should be passed to the language model to condition its response to the situation.

Generative Agents: Interactive Simulacra of Human Behavior. Joon et al. 2022

Sequential Prompting

- You are a day-to-day activity planner. Here are the requirements:
- Objectives:
  - Buy **bread** for tomorrow's breakfast
  - Give a **2 hour lecture on Large Language Models** and prompting
  - Have time for **breakfast, lunch, dinner**
  - Have a **one hour games session** at night
  - Have time for some **3 hour group discussion** with students in the afternoon
- Constraints:
  - Only awake from 8am to 10pm, and the rest of the time will be sleeping

- Give the broad overview of the plan for the day
- Give a detailed breakdown of the plan, highlighting any special requirements or preparations and things to take note of

LLM’s answer (ChatGPT - Broad)

- Here's a broad overview of the day's plan:

8:00 AM - Wake up and get ready.

8:30 AM - Have breakfast.

9:00 AM - Go out and buy bread for tomorrow's breakfast.

9:30 AM - Prepare for the 2-hour lecture on Large Language Models and prompting.

10:00 AM - Start the lecture.

12:00 PM - Finish the lecture and take a break for lunch.

12:30 PM - Have lunch.

1:00 PM - Gather with students for a 3-hour group discussion.

4:00 PM - End the group discussion and take a short break.

4:15 PM - Review and prepare for the evening game session.

5:00 PM - Start the one-hour game session.

6:00 PM - Wrap up the game session.

6:15 PM - Take a break and relax.

7:00 PM - Prepare and have dinner.

8:00 PM - Engage in personal activities or leisure time.

10:00 PM - Start winding down for bedtime.

LLM’s answer (ChatGPT – Detailed)

- Now, let's break down the plan with more details, considering the special requirements and preparations:

8:00 AM - Wake up and get ready:

1. Set an alarm to wake up at 8:00 AM.
2. Ensure you have a comfortable sleep environment to get a good night's rest.

8:30 AM - Have breakfast:

1. Plan a nutritious and fulfilling breakfast to start the day.
2. Consider having bread or other breakfast options along with your preferred choices.

9:00 AM - Go out and buy bread for tomorrow's breakfast:

1. Make a shopping list to ensure you don't forget anything.
2. Check nearby grocery stores or bakeries for fresh bread.
3. If necessary, bring cash or payment methods for the purchase.

9:30 AM - Prepare for the 2-hour lecture on Large Language Models and prompting:

1. Gather your lecture materials, including slides, notes, and any relevant examples.
2. Review the content to ensure you're well-prepared to deliver the lecture effectively.

10:00 AM - Start the lecture:

1. Allocate sufficient time for the lecture, keeping in mind that it should last for approximately 2 hours.
2. Engage with the audience, encourage questions, and facilitate discussions as appropriate.

Hierarchical Representation of Actions

- Hierarchical chunking of sets of commonly done actions can help reduce sequence length to goal state

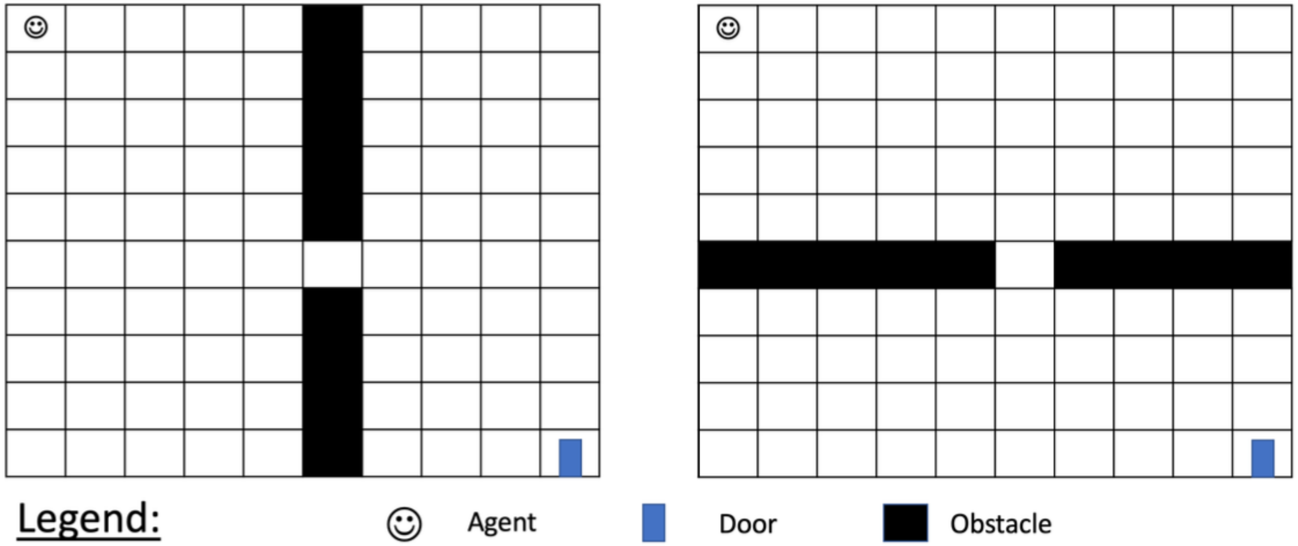


Figure 6: A sample maze environment of size 10x10. By default, the agent's start state is at the top left and the door is at the bottom right, but it can be varied. **(Left)** Obstacles before episode 50 form a vertical wall with a gap in the center across the mid-point. **(Right)** Obstacles after episode 50 from a horizontal wall with a gap in the center across the mid-point.

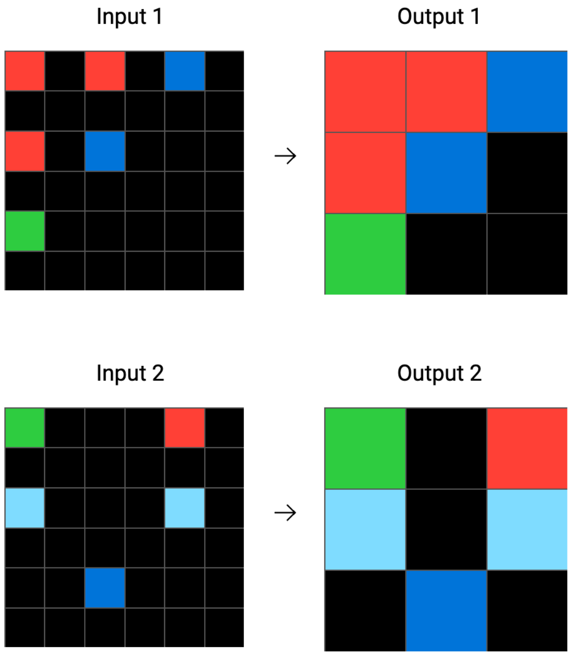
Hierarchical Memory Referencing in Text (my idea)

- Abstraction and Reasoning Corpus (ARC) Challenge

- **Broad Intent:** Reduce the input grid to a smaller size
  - Can reference/recite similar broad intents from memory to refine broad intent

- **Detailed Steps (conditioned on Broad Intent):** Remove every other square from the row and columns of the grid
  - Can reference/recite similar detailed steps from memory to refine detailed steps

- Execution: Perform the detailed steps on the test input to get the answer



Transformers:
Can a Transformer perform hierarchical generation?

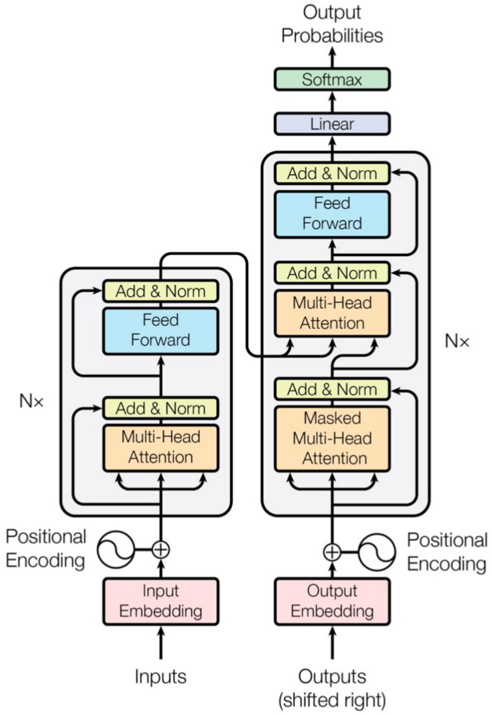
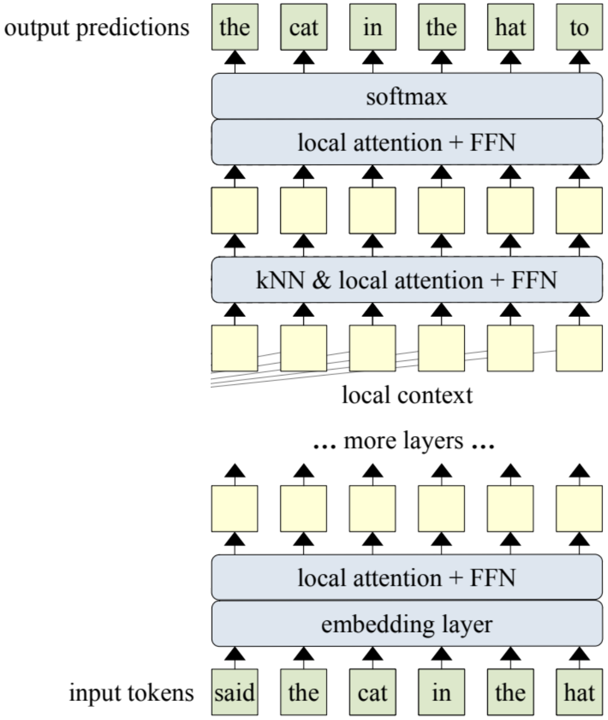


Figure 1: The Transformer - model architecture.

Attention is all you need. Vaswani et al. (2017)



Memorizing Transformers. Wu et al. 2022.

Questions to Ponder

- Do we learn from experience, or from natural fixed bias? Or both?
- Should we do prediction whereby we map back to input space (like tokens in Transformers), or should we just predict the latent space? What are the benefits and drawbacks?
- Should we use hierarchical generation? Is our brain hierarchical or more flat like what Jeff Hawkins proposes in “Thousand Brains Theory”?
- How do we structure hierarchical abstractions? Could chunking or vector quantization help?
- For a given problem, how would we know which is the right hierarchy of abstraction to use?
- Should we represent latent space as continuous or discrete? Would an unbounded length of discrete tokens be sufficient to represent continuous spaces?