AI Literacy & DS Ethics

The Jackson Laboratory

Module 3: Generative Models & Transformers

Review

Fully connected Networks

- Built from multiple regression models combined together
- Can solve complex non-linear problems
- Still used as a fundamental part of many more advanced types of network
- Need to represent input data as a vector of fixed size

Review

Images can be thought of as matrixes of numbers

Convolutional neural networks

Strengths:

- Can efficiently summarize large input matrixes
- Can be reversed to (re)create an image

Limitations:

- No memory
- Limited field of view
- Fixed number of channels

Review

Tokenization

- By representing things like words with vectors of numbers we can input them into neural networks

Recurrent neural networks

Strengths:

- Adds memory to a neural network
- Incorporates sequential context
- A lot of things are sequential

Limitations:

- Computationally expensive
- Limited memory capacity and range
- Can’t learn relative importance of tokens

Day 4 review

What Is one new thing you learned yesterday?

What lingering questions does everyone have about yesterday’s material?

Learning Goals:

How do we Generate new things with a neural network and what does a state-of-the-art network look like?

You will be able to:

- Contrast generative vs. discriminative / predictive models; outline GAN generator–discriminator interplay.
- Explain at a high level how self-attention works and why it replaced recurrence for many tasks.
- Appreciate the power (and pitfalls) of large-scale pre-training through guest talk & games.

Generative Adversarial Networks

Not enough regression

Learning Goals:

How can we create a loss function for generated content?

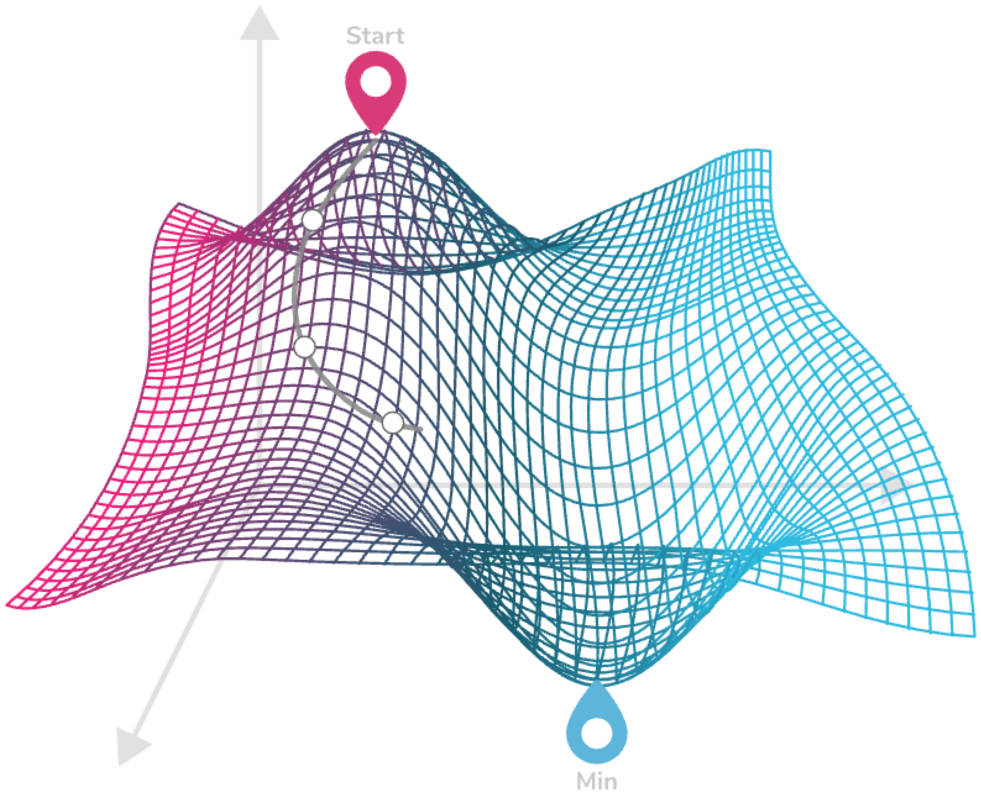
You will be able to:

- Contrast generative models with “predictive” models
- Describe how a discriminator network works
- Infer how adversarial learning allows training of generative models

Call Back: Loss Functions

A good loss function is necessary to train a neural network

- Think back to the treasure splitting game from day 1
- A loss function measures how different your output is from the desired output
- This is easy when you have a single right answer for each input



How do we quantify the quality of generated content?

Q: When generating something "new", how can we compute the loss?

- What are we trying to quantify?
  - Plausibility

- How can we measure that?
  - What if we could measure how good our generator is at tricking someone or something?

To consider: what implications does defining loss in this way have for AI Hallucinations

Can we turn generation into a prediction problem?

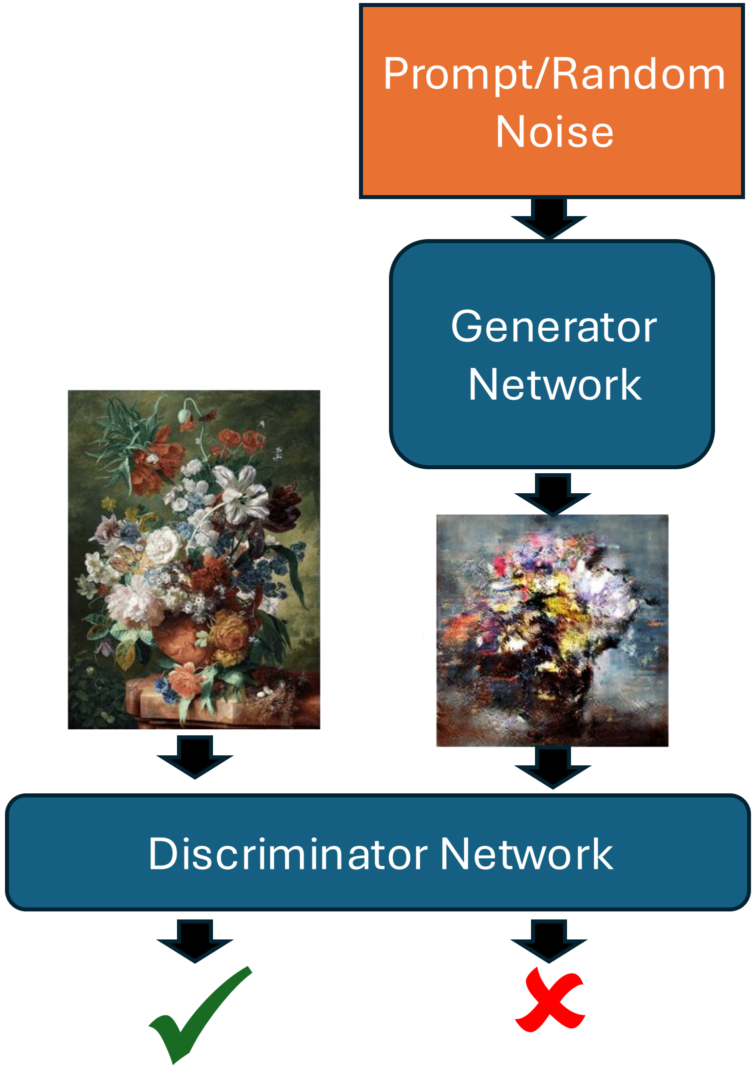
Generative Adversarial Networks have two parts:

- The Generator which creates something new

  Examples:

  - Deconvolutions to create an image
  - A Recurrent Network that chooses a plausible next word for a sentence

- The Discriminator which takes the generated output and a real example and tries to determine which is the fake



Prompt/Random Noise

Generator Network

Discriminator Network

✓ ✗

To discuss after the activity

This activity puts you in the role of generators and discriminators

Generators: what strategies did you use to improve the plausibility of the answers as the game went on?

Discriminators: were you basing your decision on anything other than plausibility?

Everyone: what implications would training a network in this way have, if there were no correct answer?

Data science balderdash

1. Split into pairs and decide on who will be the generator and who will be the discriminator. (in the case of an odd number of players one group should have 2 discriminators)

- Generators:
  2. take one of the stacks of index cards, on one side is a data science themed question on the other is the real answer.
  3. Make up a fake answer to the question
  4. Tell the discriminator(s) both your fake answer and the real answer
- Discriminators:
  5. Guess which answer is the real one

6. Keep playing more rounds until time is called or you run out of cards (optionally switching roles)

Activity Discussion

This activity puts you in the role of generators and discriminators

Generators: what strategies did you use to improve the plausibility of the answers as the game went on?

Discriminators: were you basing your decision on anything other than plausibility?

Everyone: what implications would training a network in this way have, if there were no correct answer?

Bonus points: these questions and answers were AI generated, were any of the "True answers" incorrect?

What are some major limitations of GANs?

1. Limited by the capabilities of the generator and discriminator architectures
2. Hard to train and can't be "transferred" easily

Only a limitation in some cases:

3. Correctness is only rewarded in so far as it makes the generated answer more plausible
4. Will always generate an output

vs

Review

Generative Adversarial Networks

- Allows quantifying loss for generated output
- Turns a generation problem into a classification problem
- Trained to produce plausible output
- Will always produce some form of output

Transformers & self-attention

Working with Sequences

Learning Goals:

What are Transformer Networks and how do they build on the networks discussed yesterday?

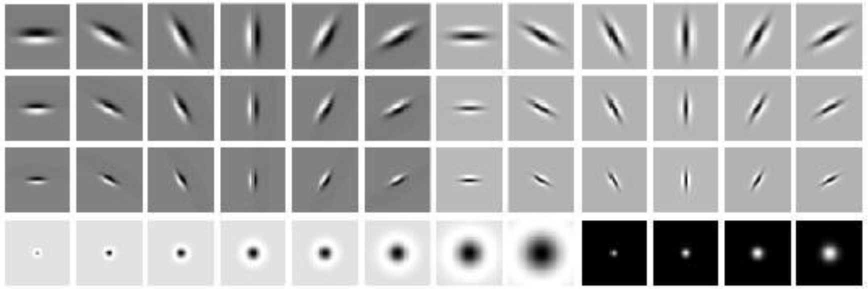
You will be able to:

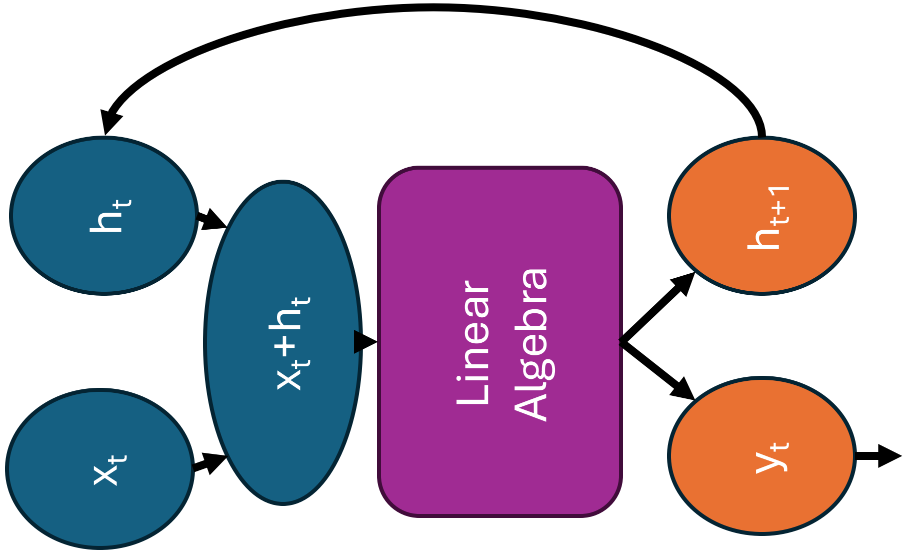
- Describe how an attention layer works
- Compare and contrast the effects of attention layers and those discussed on day two

Context is King

What do convolutions and recurrent layers have in common?

- Capture context more efficiently than fully connected layers.
- Convolutions summarize data by finding patterns.
- Recurrence summarizes sequences into a single data point via cumulative memory.
- Both focus on local context





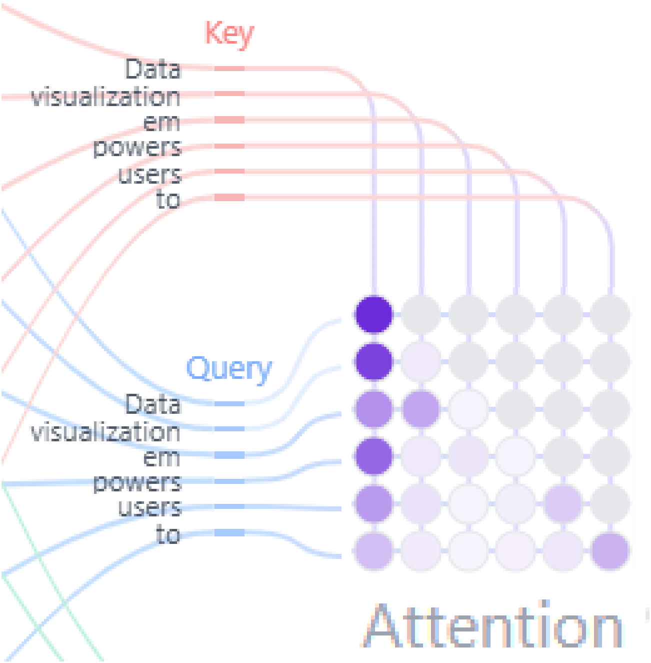
How can we broaden our context window?

Self attention captures the relationships between input tokens and has 3 steps:

1. Measure the similarity between tokens

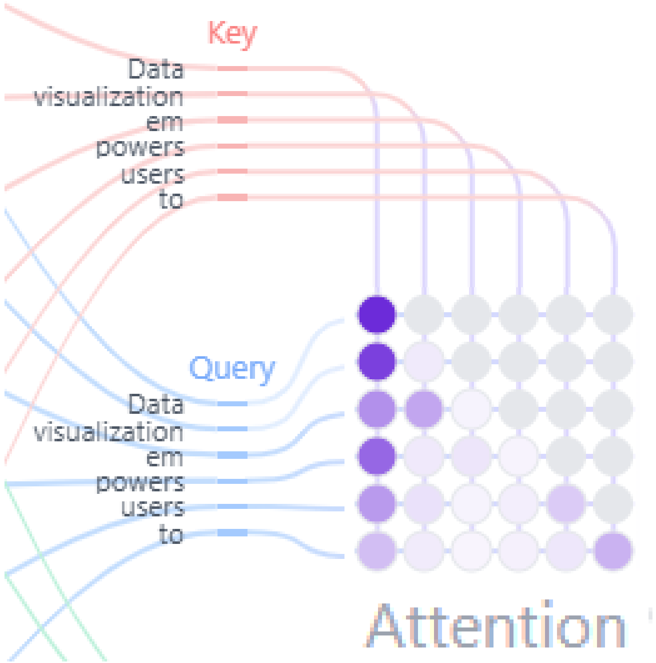
2. Scale the similarity using learned weights

3. Apply an activation function & regularize



What advantages does self attention provide?

- Can connect separated words if they are contextually related
- Multiple "attention heads" can capture many different relationships
- Each attention head can be calculated in parallel



To Discuss after the activity

We will be replaying the sentence game from yesterday but this time you will order the words based on “importance” to the sentence

What strategies did you use to order the words?

How did this compare to analyzing the sentence sequentially?

Does this way of ordering things introduce any new concerns?

Sentence Analysis Activity Redux

1. Break into small groups
2. Each group gets a stack of index cards
    - Set aside the top card
3. One member of each group sorts the words in order of importance
4. Everyone else Flip each of the remaining cards over one by one and try to guess the meaning of the sentence in as few cards as you can
5. Shuffle cards and pass to the next group.
6. Change rolls and repeat steps 3-5

Activity Discussion

We will be replaying the sentence game from yesterday but this time you will order the words based on “importance” to the sentence

What strategies did you use to order the words?

How did this compare to analyzing the sentence sequentially?

Does this way of ordering things introduce any new concerns?

Putting it all together

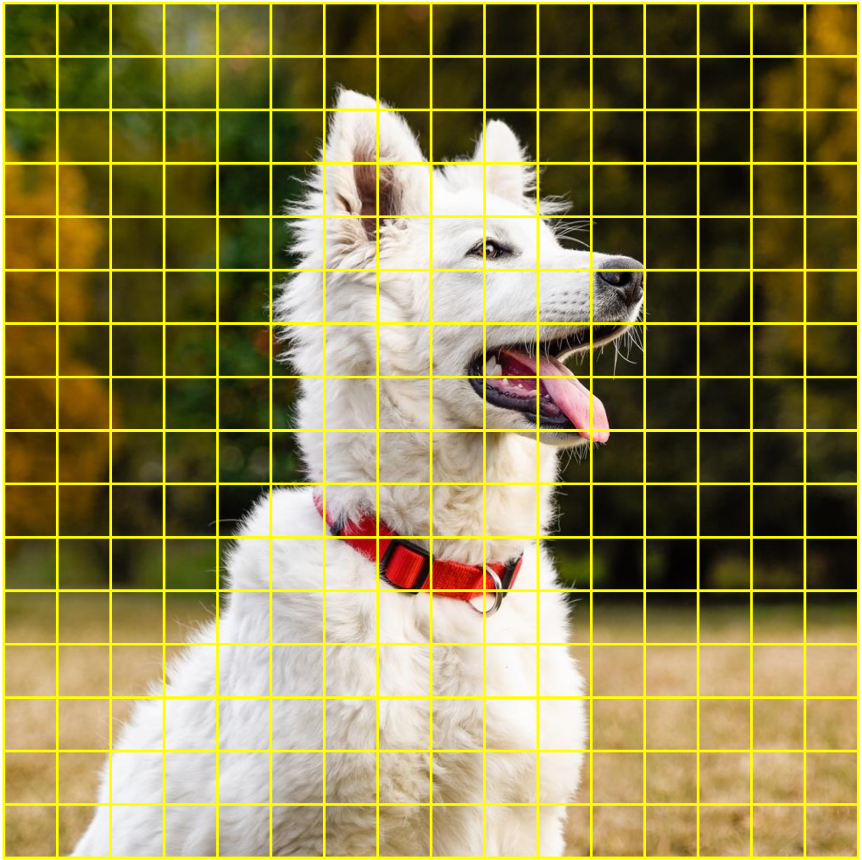
How do transformer networks work in practice?

https://poloclub.github.io/transformer-explainer/

Brief aside: Tokens aren't just for words

Vision transformers use the same self attention mechanism to improve image analysis

- Much like tokenizing a sentence an image is broken down into patches
- Convolutions can be used to turn each patch into a feature vector, or the patch can be flattened
- The feature vectors are used as tokens by the attention layer

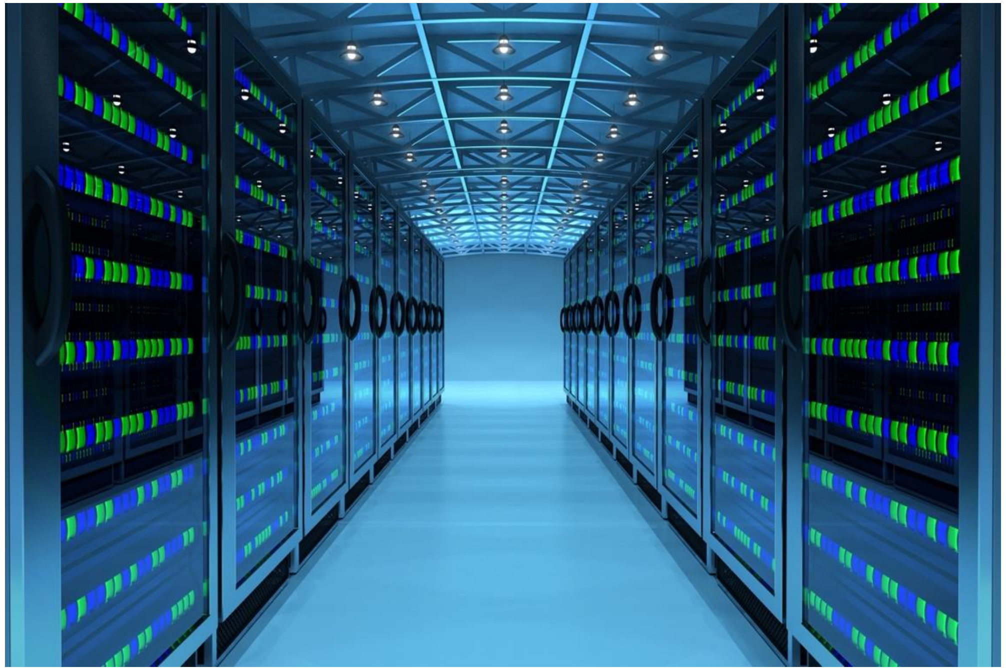


What are some major limitations of self attention layers?

1. Computationally expensive
    - While they are more parallelizable than RNNs transformers tend to take more total compute resources
2. Memory footprint
    - Larger models can have trouble fitting in working memory
3. Required training data
    - Transformers tend to need much larger training sets to learn the complex interactions they are designed to capture

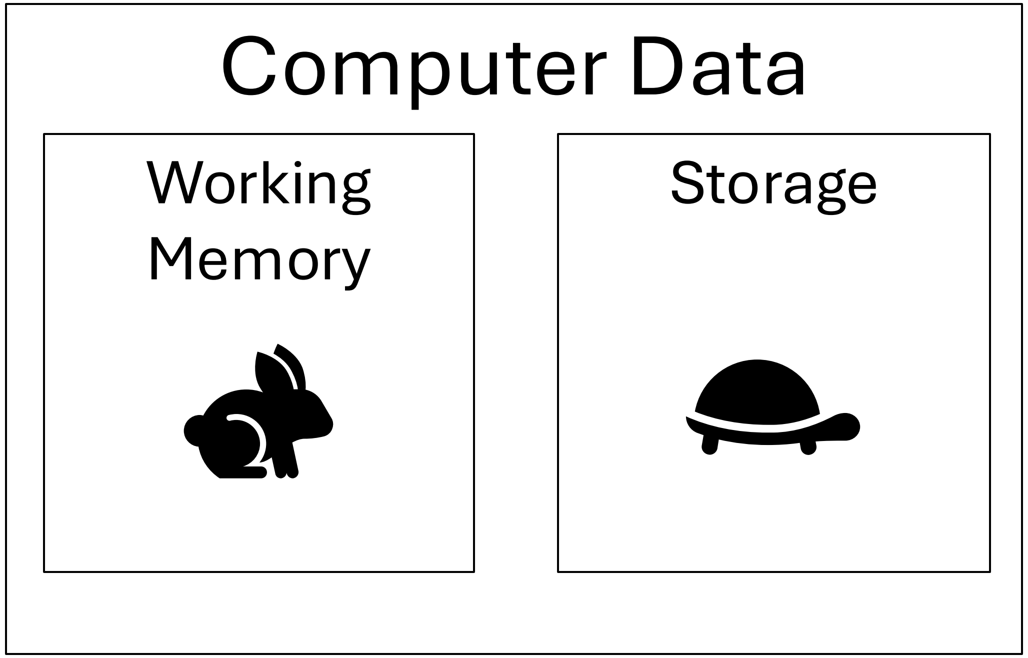
What are some major limitations of self attention layers?

1. Computationally expensive
   - While they are more parallelizable than RNNs transformers tend to take more total compute resources



What are some major limitations of self attention layers?

2. Memory footprint
   - Larger models can have trouble fitting in working memory
   - Running out of working memory Significantly slows down computation



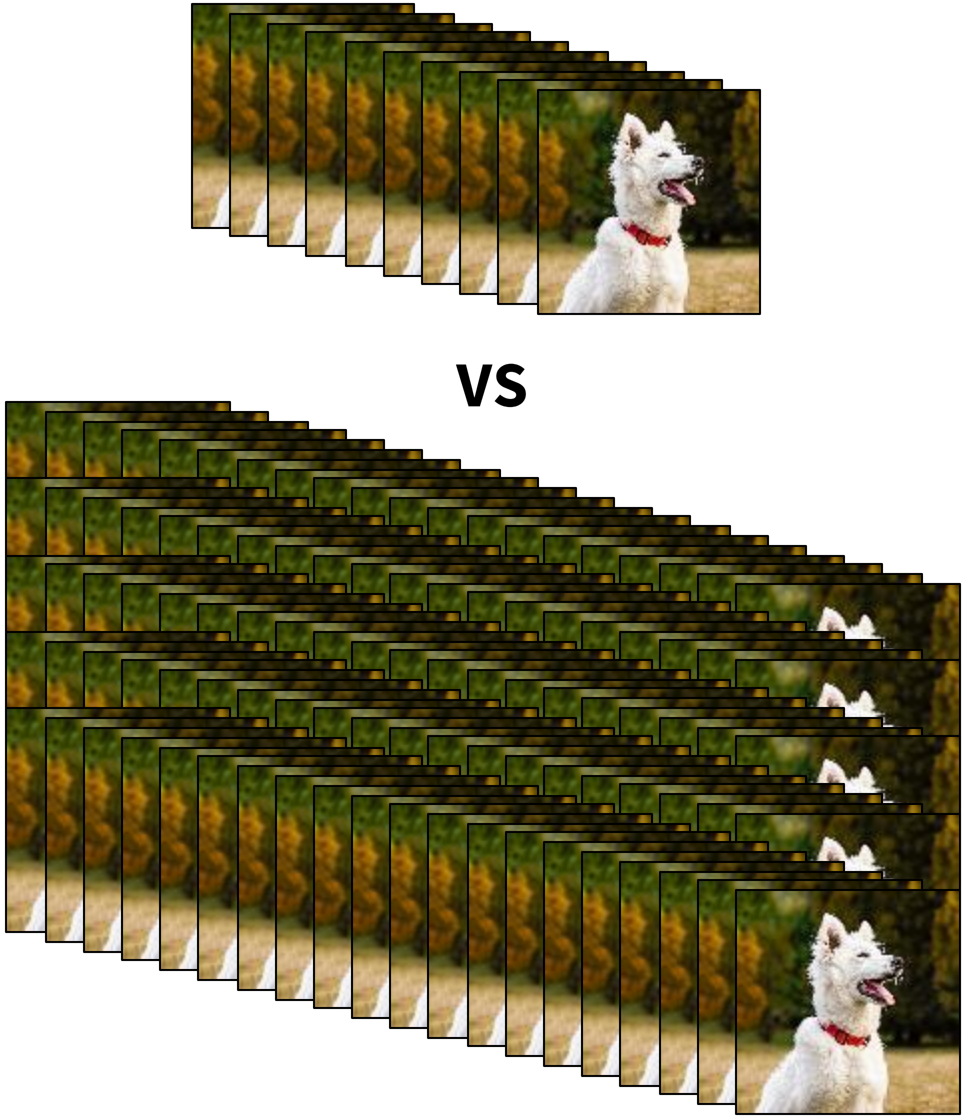
Computer Data

Working Memory

Storage

What are some major limitations of self attention layers?

1. Required training data
   - Transformers tend to need much larger training sets to learn the complex interactions they are designed to capture



Review

Transformers

Strengths:

- Wide context
- Highly parallelizable
- Broadly applicable

Limitations:

- Computationally expensive
- Memory footprint
- Required training data

Fun Fact: the T in ChatGP**T** stands for transformer

Day 5 review

What questions does everyone have about today’s material?

Next up: How LLMs Work