

# Organizational

- Homework: **You nearly did it!**
  - Homework 7 released (you will have until Dec 6<sup>th</sup>)
  - No more extensions
- Midterm Wednesday 11/29 4.30 to 5.45

# Transformers



(Image credit: Paramount Pictures)

# Transformers

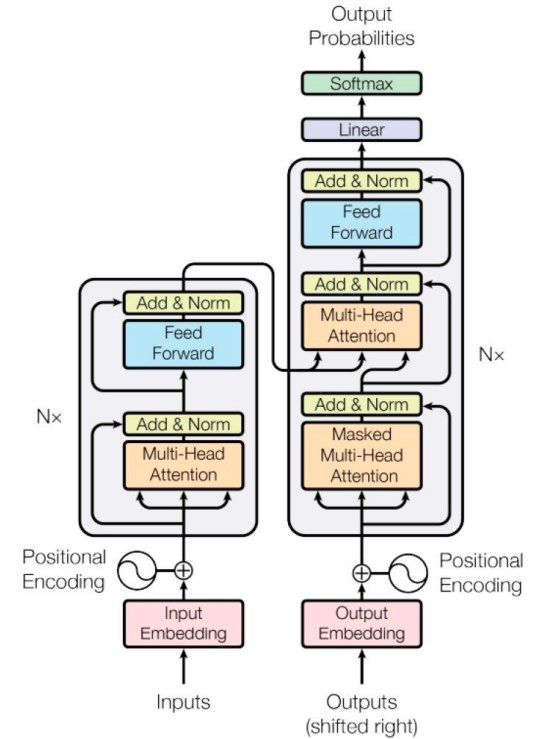
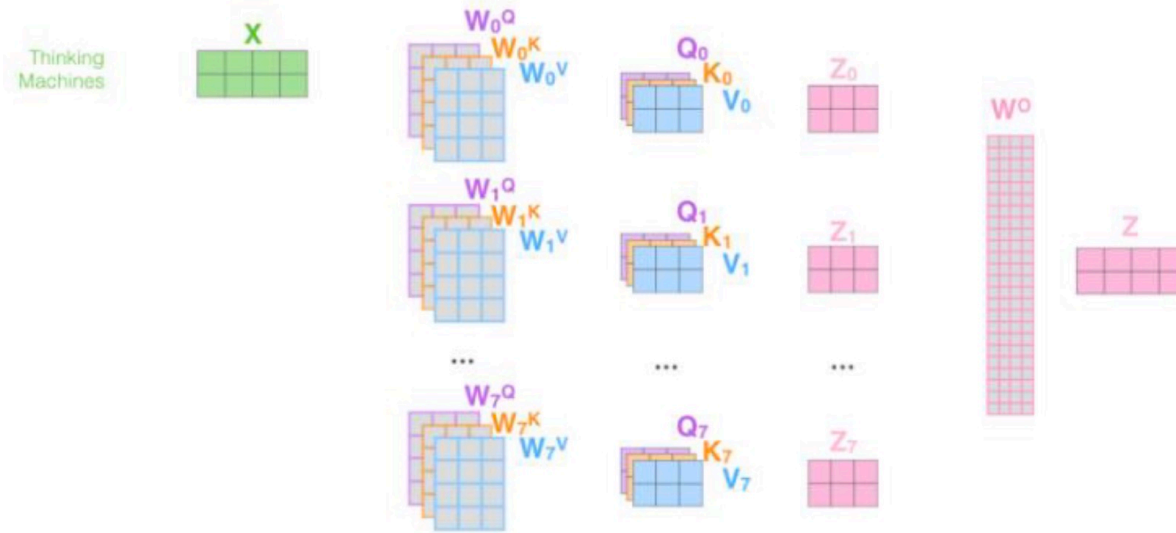
1) This is our input sentence\*

2) We embed each word\*

3) Split into 8 heads. We multiply  $X$  or  $R$  with weight matrices

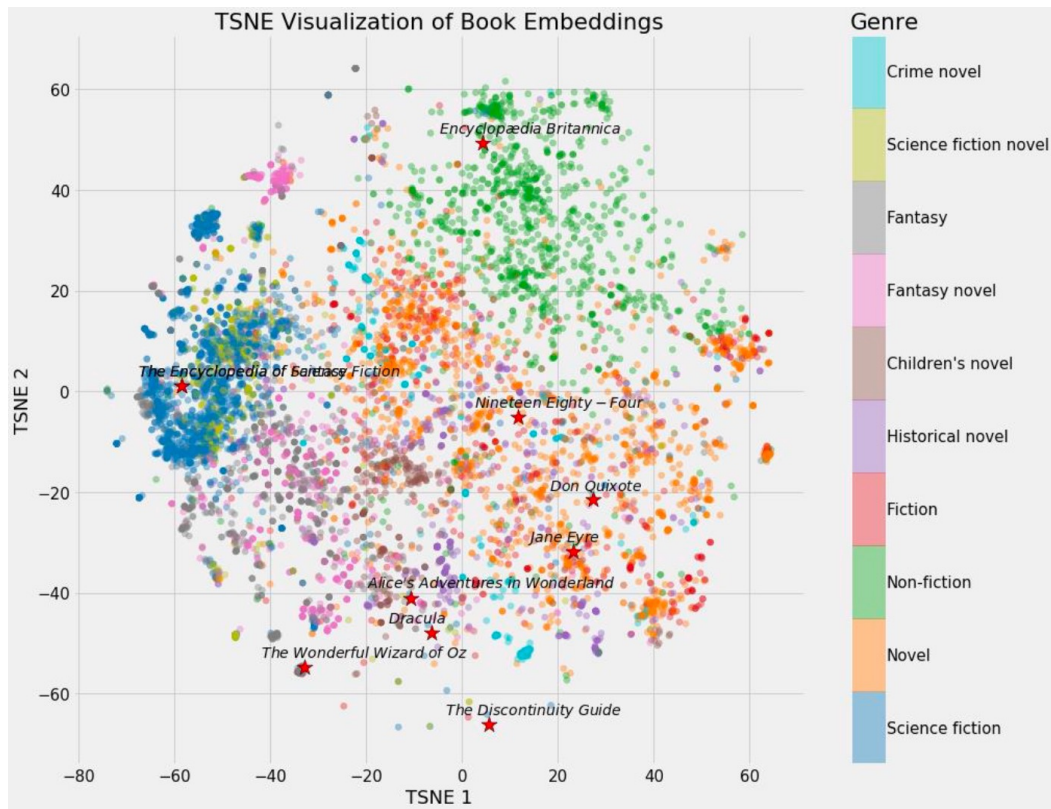
4) Calculate attention using the resulting  $Q/K/V$  matrices

5) Concatenate the resulting  $Z$  matrices, then multiply with weight matrix  $W^O$  to produce the output of the layer

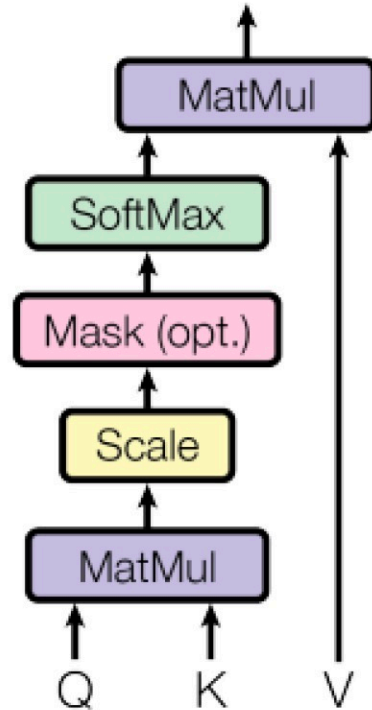


# Embedding

- From one-hot encoding (image patches) to a continuous space.
- Dimensionality reduction.
- Semantic Clusters



# Attention



$$\text{attention}(q, k, v) = \text{Softmax}(q * k^T) * v$$

The diagram shows the matrix operations for the attention formula. A purple 2x3 matrix labeled Q is multiplied by an orange 3x2 matrix labeled K<sup>T</sup>. The result is divided by  $\sqrt{d_k}$  and then passed through a softmax function. This is then multiplied by a blue 2x3 matrix labeled v to produce the final result, a pink 2x3 matrix labeled z.

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) \times v = z$$

# What exactly is Attention?

- Feature aggregation for sequence.
- How do RNN do feature aggregation? -> sum them up.
- How do attention do feature aggregation? -> weighted sum.

$$\text{attention}(q, k, v) = \text{Softmax}(q * k^T) * v$$

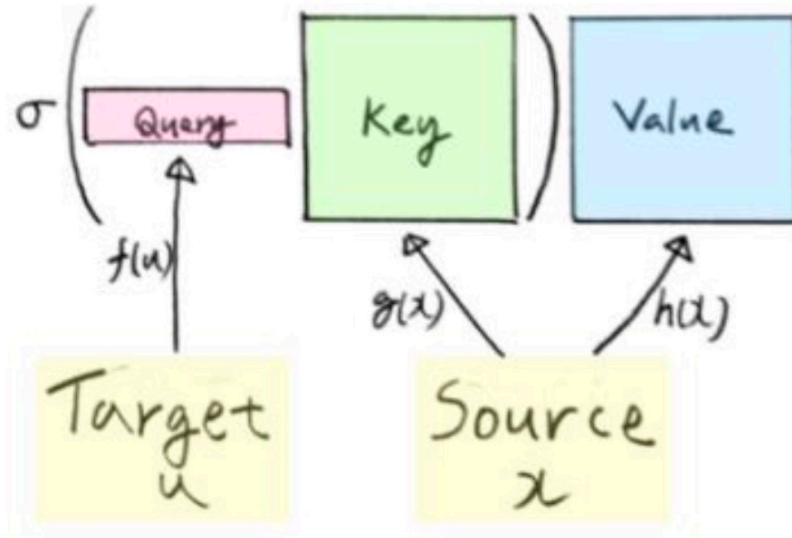
Weight

Weighted Sum

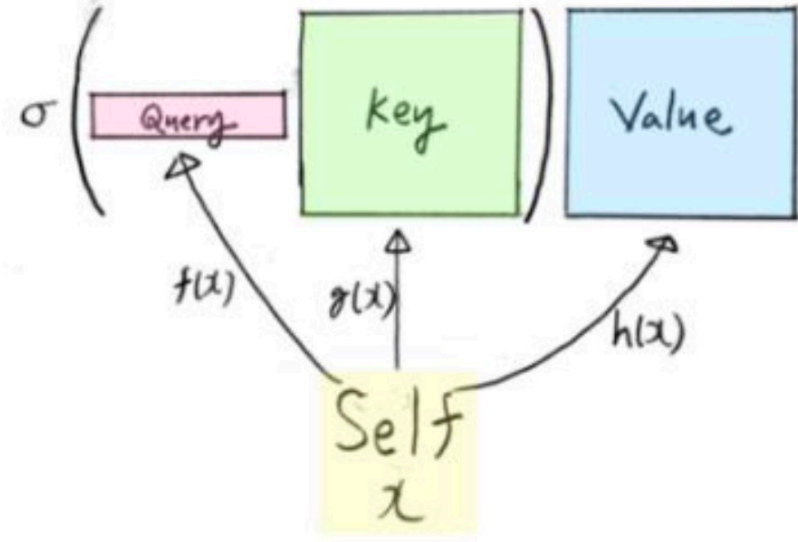


# Several Attentions

## Cross-attention

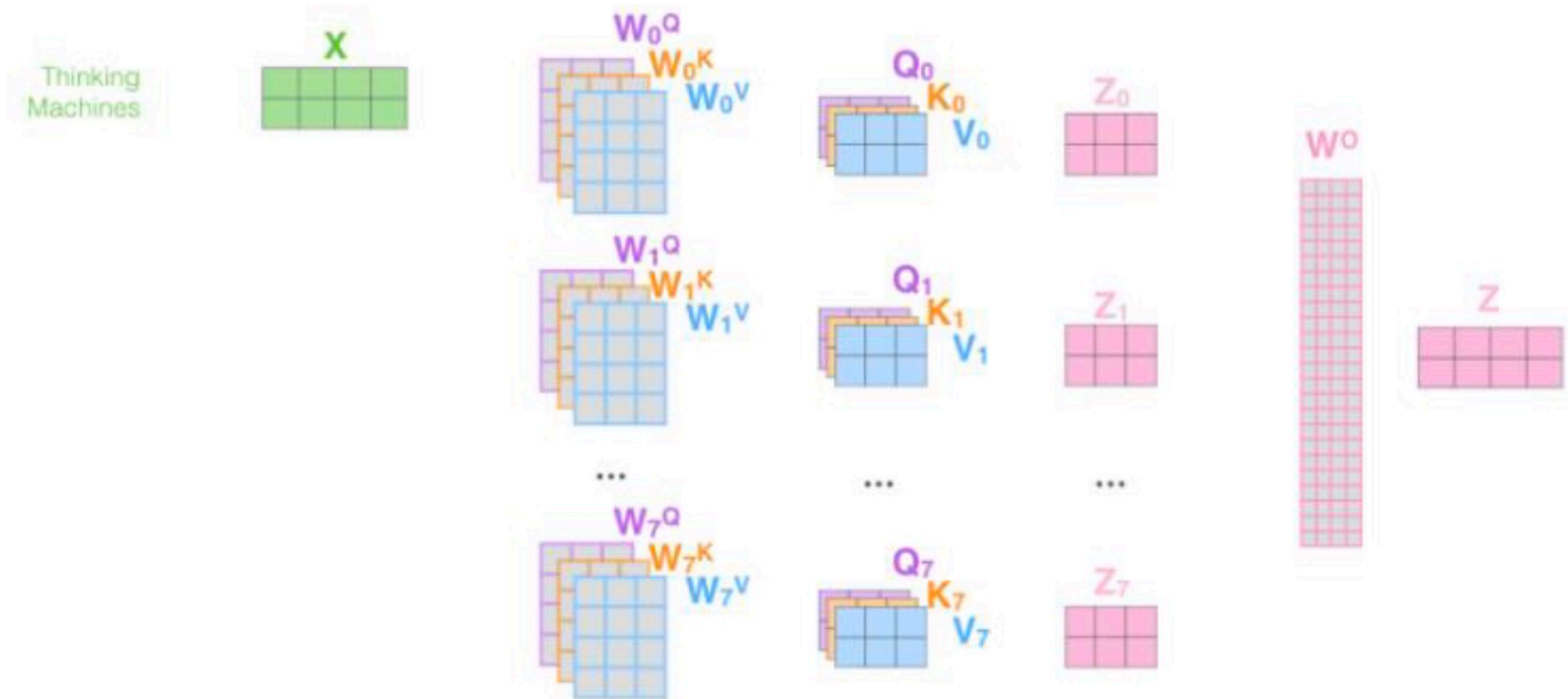


## Self-attention



# Several Attentions

## Multi-head Attention





# Transformers

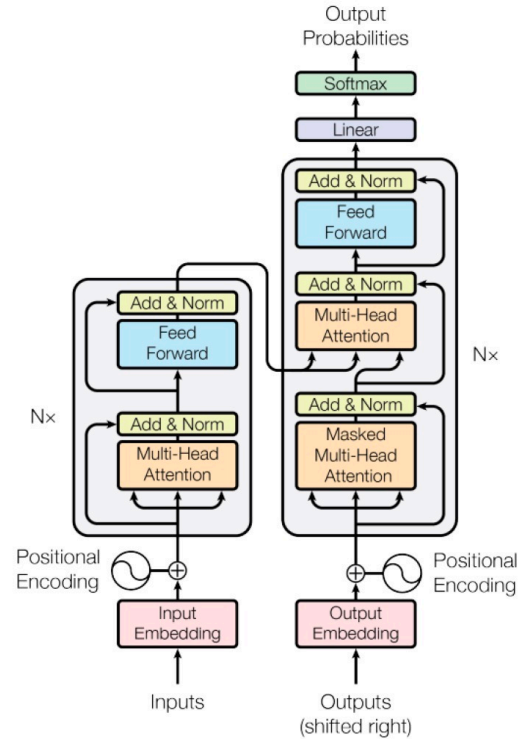
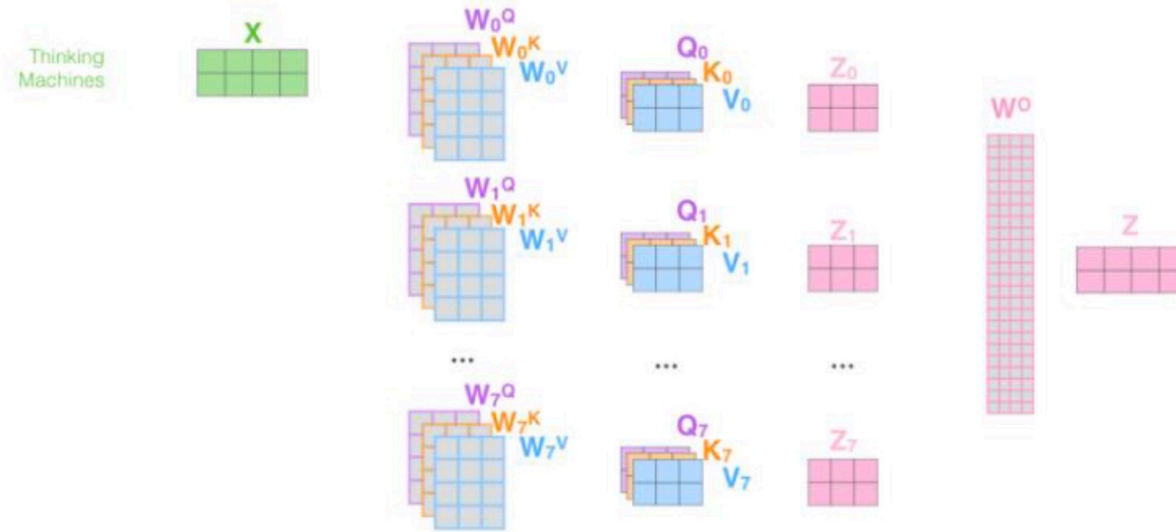
1) This is our input sentence\*

2) We embed each word\*

3) Split into 8 heads. We multiply  $X$  or  $R$  with weight matrices

4) Calculate attention using the resulting  $Q/K/V$  matrices

5) Concatenate the resulting  $Z$  matrices, then multiply with weight matrix  $W^O$  to produce the output of the layer



# Transformer vs RNN

- Transformer can do parallel training.
- Transformer keeps the form of the whole sequence.

## Challenges that the Transformers are facing

- Broken Cybertron
- $O(n^2)$  complexity for attention.
- $O(n^2)$  complexity during inference-time.
- Take look at Retentive Network if you are interested in this.

# HW7

- Three **bonus** problems: +1% towards hw per problem
- Released, check Piazza
- Due: Wednesday, Dec 6<sup>th</sup> by 11:59 pm
- Submit: a zip to Gradescope (Entry Code **BBVDNN**)

# Q1. Generative Adversarial Networks

Training a GAN to generate MNIST image

1. Define a generator
2. Define a discriminator
3. Minimax optimization
4. Goal: generate reasonable image of inception score  $\geq 1.5$

## Q2. Adversarial Attacks

1. Implement a gradient-based attack method
2. Generate adversarial example for MNIST
3. Generate adversarial example for CIFAR-10
4. Generate adversarial example for a CIFAR-alike real world photo

More fun (optional):

Play with a physical attack method on an object detection algorithm!

## Q3. Word Embedding

1. Implement two word embedding methods:
  - a. CBOW (Continuous Bag of Words)
  - b. Skip-gram
2. Train the model to generate good embeddings
3. Use the embeddings to evaluate words similarity

Balancing your efforts on

bonus problems ( $\leq 3\%$  of 50%)

VS.

the final project (25%)