12/1 AI Meeting

Topics of Discussion:

- **Lack of normalization of dot products in computing attention coefficients:** Let Q, K, and V represent the query, key, and value matrices, respectively. Let D_k be the number of columns of Q. The pre-softmax attention coefficient $a_{p,h}$ is given by

  $$\frac{q_p' k_h}{\sqrt{D_k}} \quad \text{(dot product of the p-th query and k-th key vector).}$$

  Considering that attention coefficients are intended to identify similarities between queries and keys to return weighted value representations, it is surprising that they are not computed with a normalized dot product

  $$\frac{q_p' k_h}{\sqrt{q_p' q_p}\sqrt{k_h' k_h}}$$

  to return a value between -1 and 1, inclusive. For instance, the dot product of q_p and $2q_p$ will be twice the dot product of q_p with q_p, even though their similarity, controlling for magnitude, is the same. Without normalization, higher magnitude vectors may tend to have higher attention coefficients because of their magnitudes as opposed to similarities. It is interesting that they divide by $\sqrt{D_k}$ to address the vanishing gradient problem, justifying it by the assumption that elements of vectors are iid, such that dividing by $\sqrt{D_k}$ makes the magnitude of the dot product have unit variance with an expected value of 0. This may be reinterpreted as performing a similar function to normalization, where dividing every attention coefficient by the same scalar adjusts the likelihoods of values in a way similar to temperature scaling.

- **What will happen when Key/Value matrices have the same set of parameters?** These matrices will then need to both learn a useful value representation that can also function as key vectors appropriately matching to queries. Queries will compute attention scores based on direct similarities with values rather than keys corresponding to values. Consider the movie example: each movie has an embedding, e.g., genre, critic score, year, actors, etc., which are standardized. A person inputs a query specifying their preferences for each of these categories. The key and value vectors consist of all movie embeddings, and the method of computing attention scores returns the value corresponding to the key with the highest attention score—effectively selecting the movie that has the greatest dot product with the query, resembling the closest match to the requested fictitious movie. Hard to think through an interpretable answer to what will happen if this is done to an LLM. Think about this as a regularization?

- **Word Embeddings Aren't an Encoder:** I previously misunderstood word embeddings as being generated from an initial encoder. However, this doesn't make sense since word embeddings are specific to individual words, whereas an encoder processes an entire sequence of words to give a single word a new representation.
- **Space/Time complexity of transformer when N D and when D N**
- **(Not LLM) Went through the Bias Variance Decomposition of MSE**

Experiment Ideas:

1. Conduct GPT-3 training with the same key/value parameters.
2. Conduct GPT-3 training using normalized dot products (consider alternatives to softmax).