Estrategias de decodificacion

jzazooro

December 10, 2023

1 Descripción Detallada de LLM

Los modelos de lenguaje como GPT-2 (y su sucesor, GPT-3) no generan texto directamente, sino que producen lo que se llaman "logits". Los logits son los puntajes o puntuaciones asociados con cada posible palabra o token en el vocabulario del modelo.

Para generar texto, el modelo de lenguaje toma como entrada un contexto inicial (una secuencia de palabras) y calcula la probabilidad de que cada palabra en su vocabulario sea la siguiente palabra más probable en función de ese contexto. Esta probabilidad se calcula asignando un puntaje a cada palabra en el vocabulario, y esos puntajes son precisamente los logits.

Los logits no son texto real, son valores numéricos que representan la "confianza" del modelo en cada posible palabra siguiente. Estos puntajes se convierten en probabilidades a través de una función llamada "softmax", que convierte los logits en una distribución de probabilidad sobre el vocabulario. Las palabras con probabilidades más altas son más probables de ser elegidas como la siguiente palabra en la secuencia generada por el modelo.

Por lo tanto, aunque el texto generado parece fluir de manera natural y coherente, en realidad está determinado por una serie de cálculos matemáticos que asignan puntajes a cada palabra en función del contexto dado.

Los modelos de lenguaje, como GPT-2 o GPT-3, se basan en arquitecturas de aprendizaje profundo, específicamente en redes neuronales llamadas redes neuronales transformadoras (Transformers). Estos modelos utilizan capas de atención y capas completamente conectadas para procesar y comprender el texto.

Cuando se alimenta al modelo con una secuencia de palabras, este realiza una serie de cálculos matemáticos complejos para generar logits para cada palabra en el vocabulario. Estos cálculos se realizan a través de múltiples capas y operaciones que se entrenan utilizando grandes cantidades de texto con el objetivo de predecir la siguiente palabra en una secuencia.

Una vez que se obtienen los logits para cada palabra en el vocabulario, se aplica la función softmax, que convierte estos logits en probabilidades. La palabra con la probabilidad más alta se selecciona como la siguiente palabra en la secuencia generada por el modelo.

Es importante tener en cuenta que los modelos de lenguaje como GPT-2 no comprenden el texto de la misma manera que lo hace un humano. No tienen un entendimiento semántico o conceptual de las palabras o frases; en cambio, aprenden patrones estadísticos en los datos con los que se entrenan. Estos modelos pueden generar texto coherente y contextualmente relevante debido a la gran cantidad de datos que han procesado durante el entrenamiento.

Los logits son valores numéricos que representan la "confianza" del modelo en cada posible palabra en su vocabulario dada una secuencia de entrada. Estos valores numéricos se obtienen después de que el modelo ha procesado el contexto dado y ha calculado la probabilidad de cada palabra como la siguiente en la secuencia.

Después de obtener los logits para cada palabra en el vocabulario, se aplica la función softmax para convertir estos valores en una distribución de probabilidad. La función softmax escala estos logits para que se conviertan en probabilidades, de modo que la suma de todas las probabilidades para las posibles palabras sea igual a 1. Esto se logra al aplicar una exponencial a los logits y normalizar los resultados.

Una vez que se tienen estas probabilidades para cada palabra en el vocabulario, se selecciona la palabra con la probabilidad más alta como la siguiente palabra en la secuencia generada por el modelo. Esta palabra se elige en función de la lógica del modelo, que busca predecir la palabra más probable en función del contexto proporcionado por la secuencia de entrada.

Este proceso se repite iterativamente para generar una secuencia continua de palabras que forman el texto final. Cada vez que se elige una palabra como la siguiente en la secuencia, el modelo utiliza esa palabra como parte del contexto para predecir la siguiente palabra, y así sucesivamente.

Es importante tener en cuenta que aunque los logits representan la confianza del modelo en cada palabra, la generación de texto no es completamente determinista. Los modelos de lenguaje a menudo introducen cierto grado de aleatoriedad o variabilidad para producir textos más diversos. Esto puede manifestarse en la elección de palabras menos probables basadas en sus logits, lo que ayuda a generar textos más interesantes y diversos.

2 Proceso de Generación de Texto

El proceso de generación de texto en un modelo de lenguaje como GPT-2 involucra varios pasos, desde la entrada de texto hasta la predicción del siguiente token (palabra o símbolo) en la secuencia generada. Aquí está el proceso paso a paso:

1. Preprocesamiento de texto:

   Cuando se ingresa un texto al modelo, este se convierte en una representación numérica comprensible para la red neuronal. Esto implica tokenizar el texto (dividirlo en partes más pequeñas, como palabras o subpalabras) y convertir cada token en un vector numérico.

2. Codificación posicional:

   Para preservar el orden de las palabras en la entrada, se agrega información sobre la posición relativa de cada token. Esto es esencial para que el modelo comprenda la secuencia y la relación entre las palabras.

3. Paso a través de las capas del modelo:

   El texto codificado se pasa a través de múltiples capas de atención y redes neuronales. Estas capas realizan cálculos complejos para procesar y comprender la información contextual. Cada capa procesa la entrada y la transforma, extrayendo información relevante en cada paso.

4. Cálculo de logits:

   Después de pasar por las capas del modelo, se calculan los logits para cada token en el vocabulario. Estos logits representan la puntuación o "confianza" del modelo en la probabilidad de que cada token sea el siguiente en la secuencia.

5. Aplicación de softmax:

   Los logits se transforman en probabilidades mediante la función softmax. Esta función convierte los logits en una distribución de probabilidad sobre el vocabulario, donde la suma de todas las probabilidades es igual a 1.

6. Selección del siguiente token:

   Se elige el siguiente token en la secuencia basándose en las probabilidades calculadas. Generalmente, se selecciona el token con la probabilidad más alta como el siguiente en la secuencia generada.

7. Generación continua:

   El token seleccionado se agrega a la secuencia generada, y se utiliza junto con el contexto existente para predecir el siguiente token. Este proceso se repite iterativamente hasta que se alcance una longitud deseada para la secuencia generada o se cumpla una condición de finalización predefinida.

Este proceso se lleva a cabo de manera iterativa y recursiva, generando texto palabra por palabra basado en el contexto proporcionado por la secuencia anterior. La generación resultante es una predicción del modelo sobre cuál podría ser la secuencia de texto más coherente y probable dada la entrada proporcionada.

Token a ID de token: Cuando se tokeniza un texto, cada palabra o símbolo se asigna a un identificador único conocido como "ID de token".

ID de token a token: La conversión inversa implica tomar un ID de token y devolver el token correspondiente desde el vocabulario. En otras palabras, recuperar la palabra o símbolo a partir de su ID de token.

El proceso de conversión entre tokens y sus identificadores únicos permite al modelo representar y manejar texto en forma numérica, lo que es fundamental para su procesamiento y generación de texto.

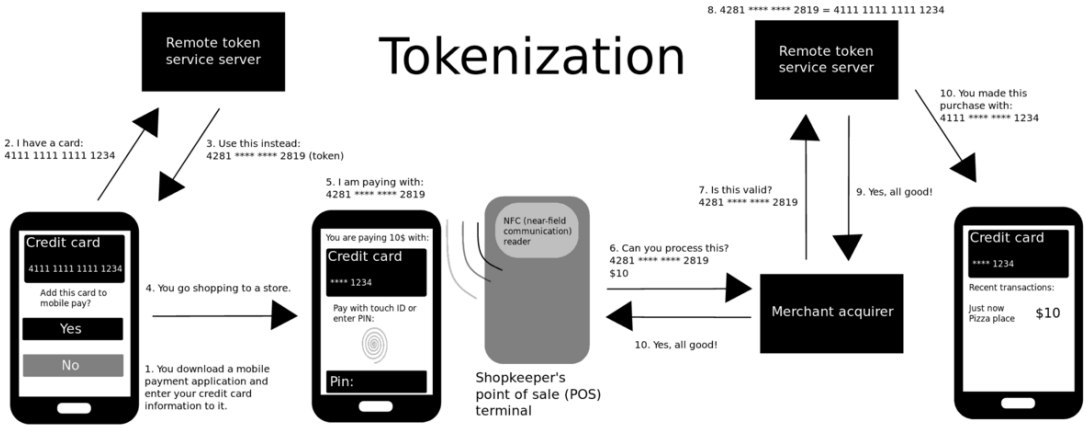

Figure 1:

La representación de token a ID y viceversa es esencial en el procesamiento de lenguaje natural. Este mapeo bidireccional permite al modelo comprender y manipular texto como secuencias numéricas, facilitando así su capacidad para procesar, generar y trabajar con datos textuales de manera efectiva y eficiente.

graphicx

3 Estrategias de Decodificación

La búsqueda codiciosa (greedy search) y la búsqueda de haz (beam search) son algoritmos utilizados en la generación de texto por modelos de lenguaje, como GPT-2 o GPT-3, para predecir la siguiente palabra en una secuencia.

- Búsqueda codiciosa (Greedy search): En este enfoque, el modelo selecciona la palabra con la probabilidad más alta como la siguiente en la secuencia en cada paso de generación. Esto significa que en cada punto de decisión, se elige la palabra más probable según las predicciones del modelo en ese momento. Aunque es simple y eficiente, la búsqueda codiciosa tiende a producir secuencias que pueden ser localmente óptimas en cada paso, pero no necesariamente generan la mejor secuencia globalmente coherente y diversa. Puede quedarse atascada en una palabra común y no explorar otras posibilidades menos probables pero potencialmente más adecuadas para el contexto más amplio.

- Búsqueda de haz (Beam search): En contraste, la búsqueda de haz es un enfoque más avanzado. En lugar de elegir una sola palabra en cada paso, la búsqueda de haz considera múltiples opciones simultáneamente. Mantiene un conjunto (haz) de las mejores N predicciones en cada paso, donde N es un parámetro definido por el usuario (generalmente entre 5

y 10). El modelo genera múltiples secuencias parciales y selecciona las N secuencias más prometedoras según las probabilidades calculadas. Luego, estas secuencias se utilizan para predecir la siguiente palabra, y el proceso se repite hasta alcanzar un punto de finalización o longitud deseada. A través de este enfoque, la búsqueda de haz puede explorar un conjunto más amplio de posibilidades y producir secuencias más diversas y coherentes en comparación con la búsqueda codiciosa.

La búsqueda de haz tiende a generar resultados más variados y potencialmente mejores, pero también es más computacionalmente intensiva debido a la consideración de múltiples caminos. Ambos enfoques tienen sus ventajas y desventajas, y su elección depende del equilibrio entre eficiencia computacional y calidad de las predicciones que se desee obtener en la generación de texto.

El muestreo con top-k y el muestreo de núcleo son técnicas utilizadas en la generación de texto para controlar la diversidad y la fluidez de las predicciones de un modelo de lenguaje, como GPT-2 o GPT-3. Ambos métodos se emplean para equilibrar la generación de texto entre la coherencia y la variedad.

- Muestreo con top-k: Esta técnica limita las opciones de palabras a considerar en cada paso de generación. Se selecciona un valor "k" que representa el número máximo de tokens más probables permitidos en cada paso. El modelo calcula las probabilidades para todas las palabras en el vocabulario y retiene solo las k palabras más probables. Luego, se elige una palabra de este conjunto reducido utilizando las probabilidades normalizadas. Esto permite al modelo mantener cierta diversidad en su generación, ya que en cada paso se consideran solo las k palabras más probables, lo que puede evitar que el modelo se "atasque" en las predicciones más comunes.

- Muestreo de núcleo (nucleus sampling): A diferencia del muestreo con top-k, el muestreo de núcleo establece un "núcleo" o un umbral en las probabilidades acumulativas de las palabras en el vocabulario. En lugar de limitar explícitamente el número de opciones, el modelo selecciona palabras hasta que la suma acumulativa de las probabilidades de esas palabras excede un cierto umbral predefinido (denotado como "p"). Este método permite una mayor variabilidad en las predicciones al ajustar dinámicamente la cantidad de tokens candidatos en función de sus probabilidades acumulativas, lo que da como resultado una selección más flexible y adaptativa de palabras.

Muestreo de núcleo (nucleus sampling): A diferencia del muestreo con top-k, el muestreo de núcleo establece un "núcleo" o un umbral en las probabilidades acumulativas de las palabras en el vocabulario. En lugar de limitar explícitamente el número de opciones, el modelo selecciona palabras hasta que la suma acumulativa de las probabilidades de esas palabras excede un cierto umbral predefinido (denotado como "p"). Este método permite una mayor variabilidad en las predicciones al ajustar dinámicamente la cantidad de tokens candidatos en función de sus probabilidades acumulativas, lo que da como resultado una selección más flexible y adaptativa de palabras.

4 Hiperparámetros y su Manipulación

Estos son parámetros clave que se utilizan en la generación de texto con modelos de lenguaje como GPT-2 o GPT-3 para ajustar la diversidad y la calidad de las predicciones.

- Temperatura (Temperature): Este parámetro controla la aleatoriedad en las predicciones del modelo. Se usa con la función softmax durante la generación de texto y ajusta la distribución de probabilidad sobre las palabras candidatas. Una temperatura más baja tiende a priorizar las palabras con las probabilidades más altas, lo que conduce a predicciones más seguras y a menudo más repetitivas. Por otro lado, una temperatura más alta suaviza la distribución de probabilidad, lo que permite una exploración más amplia de las palabras candidatas y genera predicciones más diversas y creativas.

- numbeams: Este parámetro se usa específicamente en la búsqueda de haz (beam search). Controla la cantidad de "haces" o caminos que el modelo considera durante la generación de texto. Cuanto mayor sea el número de haces, más secuencias alternativas generará el modelo en paralelo. Un valor más alto de numbeams tiende a mejorar la coherencia y calidad de las predicciones al considerar múltiples opciones, pero también puede aumentar el costo computacional.

- topk: Este parámetro, como se discutió previamente, limita la cantidad de tokens más probables que se consideran en cada paso de generación. Define el tamaño del conjunto de palabras candidatas que el modelo evalúa para tomar su siguiente decisión. Un valor más alto de topk permite una mayor diversidad al considerar más palabras, mientras que un valor más bajo tiende a producir predicciones más seguras y menos diversas.

- topp: También conocido como "núcleo" o muestreo de núcleo (nucleus sampling), este parámetro establece un umbral en las probabilidades acumulativas de las palabras en el vocabulario. En lugar de limitar explícitamente la cantidad de opciones, topp determina las opciones disponibles basándose en la acumulación de las probabilidades de las palabras en orden descendente. Este método permite una mayor variabilidad en las predicciones, ya que ajusta dinámicamente la cantidad de tokens candidatos en función de sus probabilidades acumulativas.

Estos parámetros son fundamentales para controlar la calidad, la coherencia y la diversidad en la generación de texto mediante modelos de lenguaje, permitiendo ajustar el equilibrio entre la exploración creativa y la producción de secuencias más precisas y coherentes. Experimentar con estos valores puede ser crucial para obtener resultados óptimos dependiendo del caso de uso específico.

Por ejemplo, al generar texto con una temperatura baja y un topk alto, el modelo podría producir una salida altamente coherente pero limitada en

términos de variedad y creatividad. En contraste, una temperatura alta y un topp elevado podrían generar una salida más diversa y creativa, pero posiblemente menos coherente en términos de estructura gramatical. Experimentar con estos parámetros te permite ajustar el equilibrio entre coherencia y diversidad en la salida del modelo según tus necesidades específicas.

5 Reflexión y Conclusiones

Manipular parámetros como temperatura, numbeams, topk y topp impacta la diversidad y coherencia del texto generado por modelos de lenguaje como GPT-2 o GPT-3. Estos ajustes permiten equilibrar entre predicciones más seguras y coherentes o predicciones más diversas y creativas, adaptándose a necesidades específicas de generación de texto.