AI and Environmental Impact

Floris van der Flier

Wageningen University, Bioinformatics Group

08-07-2024

GPT-2, OpenAI, February 2019

ChatGPT, OpenAI, November 2022

Make a Video, Meta, September 2022

Sora, OpenAI, Febrauary 2024

Surfaces and Interfaces 46 (2024) 104081

Contents lists available at ScienceDirect

Surfaces and Interfaces

journal homepage: www.sciencedirect.com/journal/surfaces-and-interfaces

ELSEVIER

The three-dimensional porous mesh structure of Cu-based metal-organic-framework - aramid cellulose separator enhances the electrochemical performance of lithium metal anode batteries

Manshu Zhang [a,1], Liming Wu [a,1], Tao Yang [b], Bing Zhu [a], Yangai Liu [a,*]

[a] Beijing Key Laboratory of Materials Utilization of Nonmetallic Minerals and Solid Wastes, National Laboratory of Mineral Materials, School of Materials Science and Technology, China University of Geosciences, Beijing100083, China
[b] College of Materials & Environmental Engineering, Hangzhou Dianzi University, Hangzhou 310036, China

ARTICLE INFO

ABSTRACT

Lithium metal, due to its advantages of high theoretical capacity, low density and low electrochemical reaction potential, is used as a negative electrode material for batteries and brings great potential for the next generation of energy storage systems. However, the production of lithium metal dendrites makes the battery life low and poor safety, so lithium dendrites have been the biggest problem of lithium metal batteries. This study shows that the larger specific surface area and more pore structure of Cu-based metal-organic-framework - aramid cellulose (CuMOF-ANFs) composite separator can help to inhibit the formation of lithium dendrites. After 110 cycles at 1 mA/cm^2, the discharge capacity retention rate of the Li-Cu battery using the CuMOF-ANFs separator is about 96 %. Li-Li batteries can continue to maintain low hysteresis for 2000 h at the same current density. The results show that CuMOF-ANFs composite membrane can inhibit the generation of lithium dendrites and improve the cycle stability and cycle life of the battery. The three-dimensional (3D) porous mesh structure of CuMOF-ANFs separator provides a new perspective for the practical application of lithium metal battery.

1. Introduction

Certainly, here is a possible introduction for your topic:Lithium-metal batteries are promising candidates for high-energy-density rechargeable batteries due to their low electrode potentials and high theoretical capacities [1,2]. However, during the cycle, dendrites forming on the lithium metal anode can cause a short circuit, which can affect the safety and life of the battery [3–9]. Therefore, researchers are indeed focusing on various aspects such as negative electrode structure [10], electrolyte additives [11,12], SEI film construction [13,14], and collector modification [15] to inhibit the formation of lithium dendrites.

chemical stability of the separator is equally important as it ensures that the separator remains intact and does not react or degrade in the presence of the electrolyte or other battery components. A chemically stable separator helps to prevent the formation of reactive species that can further promote dendrite growth. Researchers are actively exploring different materials and designs for separators to enhance their mechanical strength and chemical stability. These efforts aim to create separators that can effectively block dendrite formation, thereby improving the safety and performance of lithium-ion batteries. While there are several research directions to address the issue of dendrite formation, using a separator with high mechanical strength and chem-

Price of a GFLOP

Year	Cost (USD)
1945	130 trillion
1984	100 million
1997	57,000
2012	0.97
2023	0.01

Scaling Laws for Neural Language Models, Kaplan et al., 2024
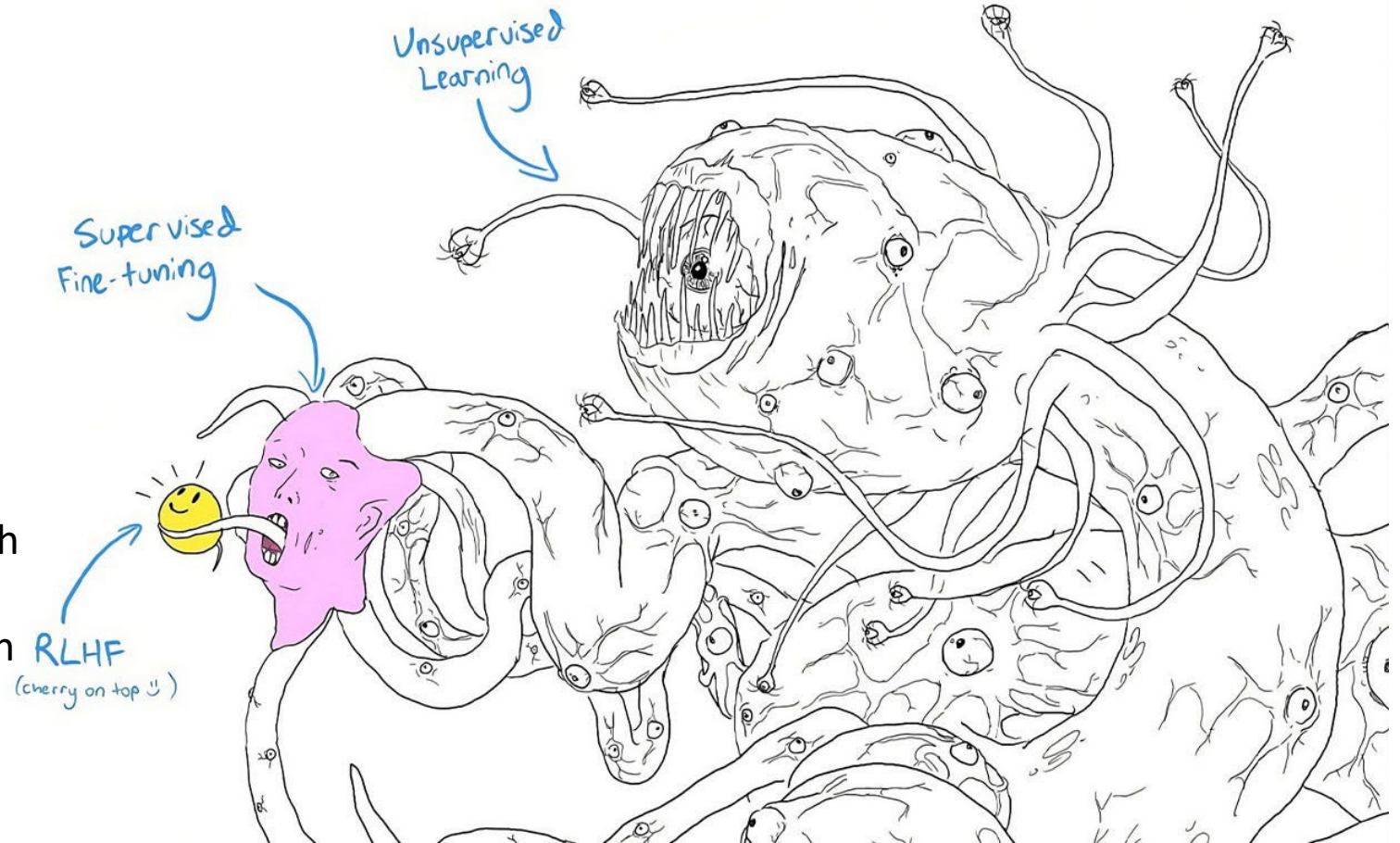
The birth of a virtual assistant

Unsupervised learning
- Next word prediction
- Alignment problem

Supervised finetuning
- Train on conversations between users and human acting as chatbot
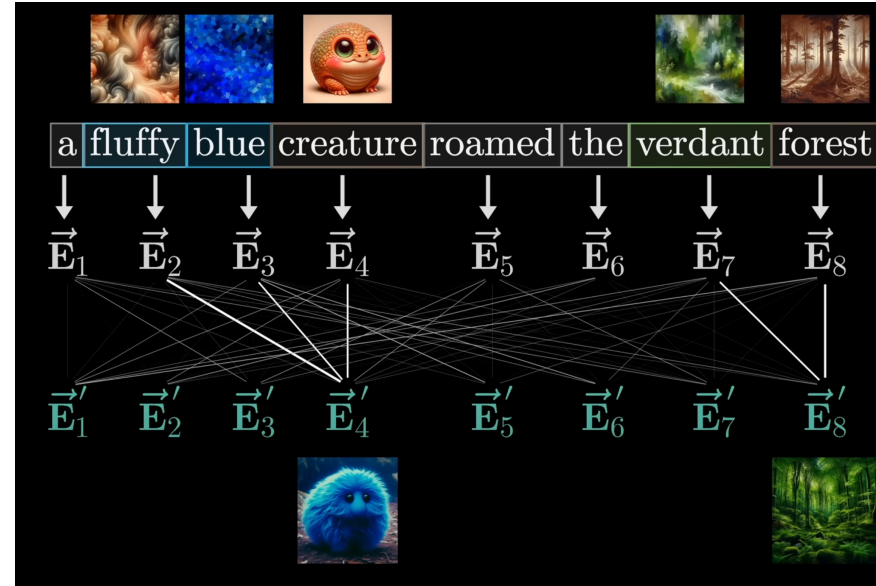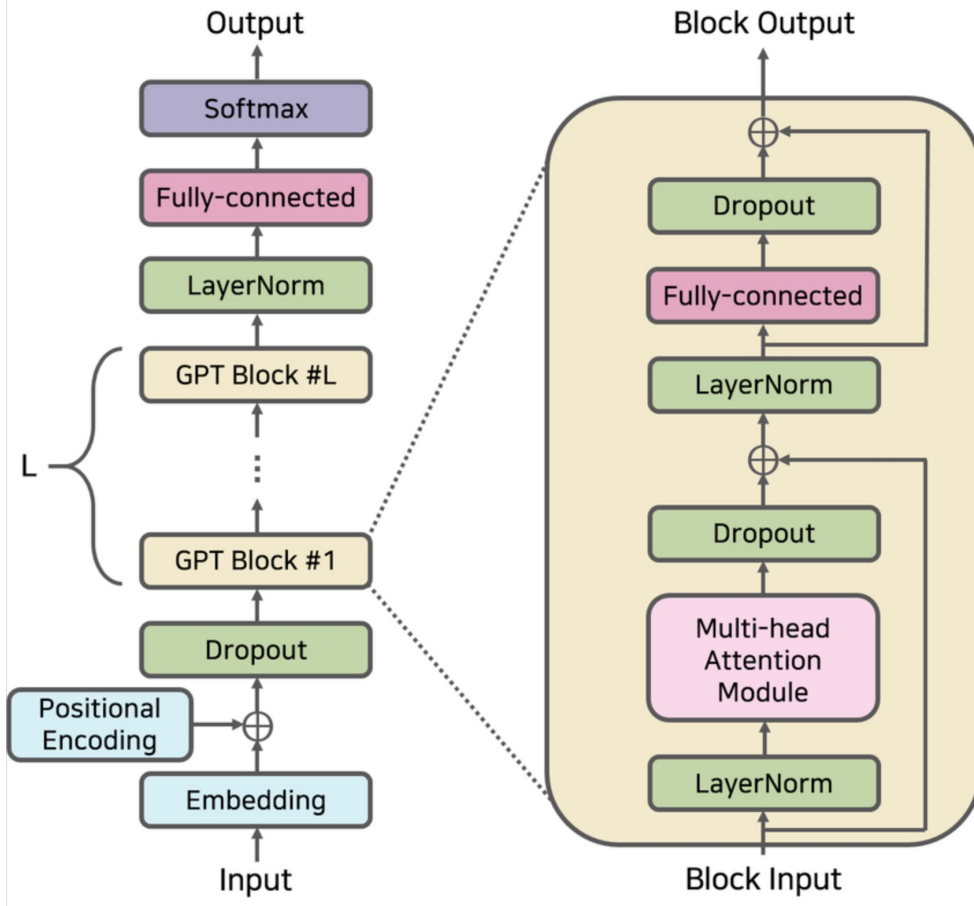- State distribution shift.

RLHF
- Encourage chatbot behavior with a reward function
- Train reward model using human labelled responses
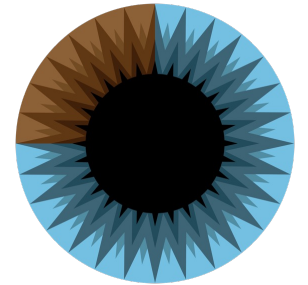- Train ChatGPT to generate responses with high reward
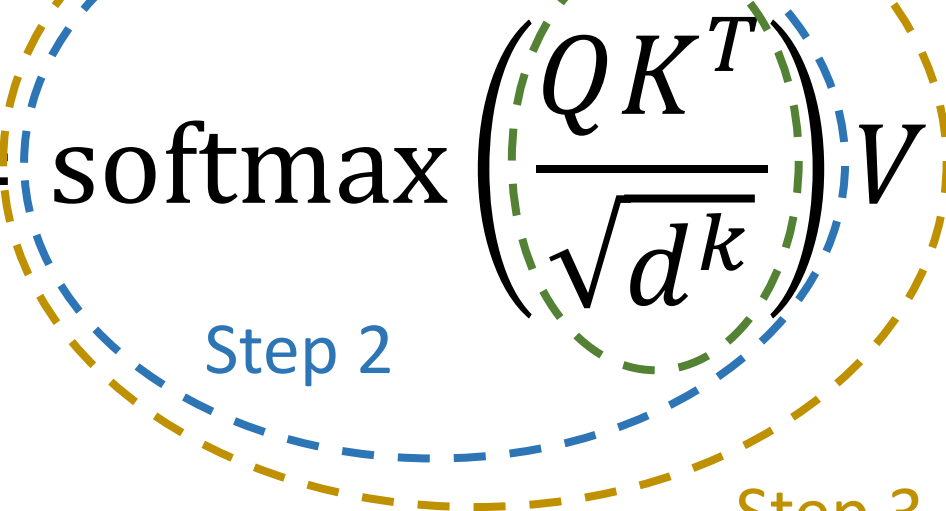
Attention is all you need

3Blue1Brown

Query embeddings
Representation of search term

Key embeddings
Representation indicating if what to look for is present.

Value embeddings
Representation of what to retrieve

$$\text{Attention}(K, Q, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d^k}}\right)V$$

Step 1

Step 2

Step 3

Exercise 1

Implement the three missing lines in the forward method according to the Attention equation

```
### IMPLEMENT STEP 1
attention = ...

# Masking attention scores that connect seen tokens to tokens not yet observed (only done in gpt-style models)
mask = torch.triu(torch.ones(context_length, context_length, dtype=torch.bool, device=q.device), diagonal=1)
attention = attention.masked_fill(mask, -torch.inf)

### IMPLEMENT STEP 2
# Calculate attention scores over the column dimension (i.e. rows sum to 1)
attention = ...

# Randomly masking out some attention scores (not important for today)
attention = self.attention_dropout(attention)

### IMPLEMENT STEP 3
pred = ...
```

```python
### IMPLEMENT STEP 1
# Calculate the attention matrices
# Tensor transformation of k and query to attention matrix
# (batch_size, n_heads, L, head_size) x (batch_size, n_heads, head_size, L) -> (batch_size, n_heads, L, L)
attention = (q @ k.transpose(-2, -1)) / math.sqrt(head_embedding_dim)

# Masking attention scores that connect seen tokens to tokens not yet observed
mask = torch.triu(torch.ones(context_length, context_length, dtype=torch.bool, device=q.device), diagonal=1)
attention = attention.masked_fill(mask, -torch.inf)

### IMPLEMENT STEP 2
# Calculate attention scores over the column dimension (i.e. rows sum to 1)
attention = F.softmax(attention, dim=-1)

# Randomly masking out some attention scores (not important for today)
attention = self.attention_dropout(attention)

### IMPLEMENT STEP 3
# weight outputs with calculated attention
# (B, n_heads, L, L) x (B, n_heads, L, head_dim) -> (B, n_heads, L, head_dim)
pred = attention @ v
```

Emissions =

Carbon intensity   ×   Energy consumption
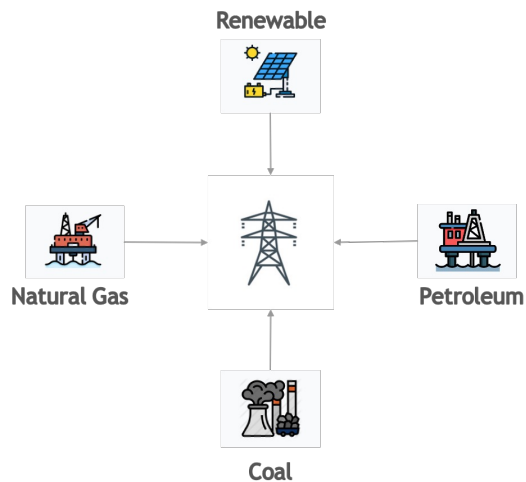
Energy Source	Carbon Intensity (kg/MWh)
Coal	995
Petroleum	816
Natural Gas	743
Geothermal	38
Hydroelectricity	26
Nuclear	29
Solar	48
Wind	26

`EmissionsTracker()`

`measure_power_secs=15`

GPU: `pynvml`
CPU: `Intel Power Gadget`
RAM: 3 watts per 8GB
Apple Silicon: `powermetrics`

```
[ ]   1 random_forest_tracker = EmissionsTracker(
      2     project_name="random_forest",
      3     output_file="random_forest_emissions.csv",
      4     output_dir=".",
      5     save_to_file=True,
      6 )
```

```
  1 data = fetch_california_housing()
  2 X = data.data
  3 y = data.target
  4
  5 # Split the dataset into training and testing sets
  6 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
  7
  8 # Fit a Random Forest model
  9 model = RandomForestRegressor(n_estimators=100, random_state=42)
 10 with random_forest_tracker as tracker:
 11     model.fit(X_train, y_train)
 12
 13 y_pred = model.predict(X_test)
 14 # Calculate r2 score
 15 r2 = r2_score(y_test, y_pred)
 16 print(f'R2 score: {r2:.2f}')
 17
 18 # Calculate Spearman correlation
 19 corr, _ = spearmanr(y_test, y_pred)
 20 print(f'Spearman correlation: {corr:.2f}')
 21
 22 # Plot scatter of predictions vs ground truth
 23 plt.scatter(y_test, y_pred, alpha=0.5, s=2)
 24 plt.xlabel('Ground Truth')
 25 plt.ylabel('Predictions')
 26 plt.show()
```

Exercise 2
Study the example above and try to modify the GPT training code to track the emissions