Transfer Learning and Fine-Tuning Large Language Models

What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.

What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.

What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.

What is Transfer Learning?

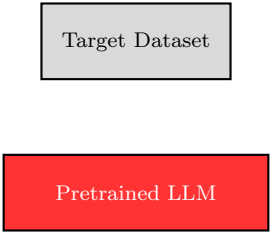
- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.
- Example: Using a language model trained on Wikipedia to write news articles.

What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.
- Example: Using a language model trained on Wikipedia to write news articles.

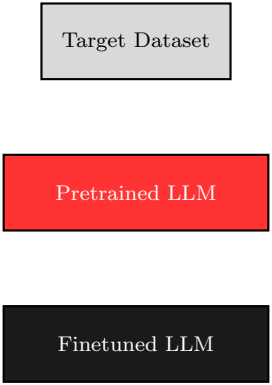
What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.
- Example: Using a language model trained on Wikipedia to write news articles.



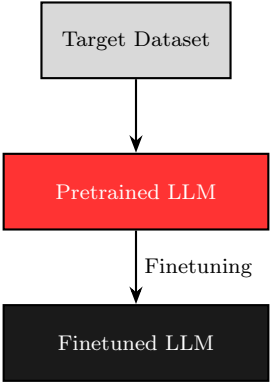
What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.
- Example: Using a language model trained on Wikipedia to write news articles.



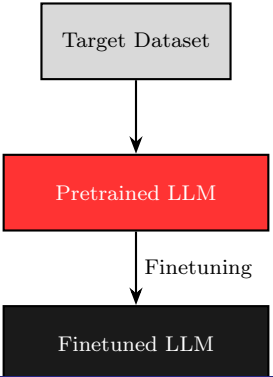
What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.
- Example: Using a language model trained on Wikipedia to write news articles.



What is Transfer Learning?

- Transfer Learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task.
- It allows leveraging pre-trained knowledge from large datasets to solve new, but related problems.
- Reduces training time, data requirements, and computational cost.
- Example: Using a language model trained on Wikipedia to write news articles.



Connection with Fine-Tuning LLMs

- Large Language Models (LLMs) are typically pre-trained on massive text corpora.

Connection with Fine-Tuning LLMs

- Large Language Models (LLMs) are typically pre-trained on massive text corpora.
- Fine-tuning adjusts the pre-trained LLM to perform specific tasks or adapt to specialized datasets.

Connection with Fine-Tuning LLMs

- Large Language Models (LLMs) are typically pre-trained on massive text corpora.
- Fine-tuning adjusts the pre-trained LLM to perform specific tasks or adapt to specialized datasets.
- Fine-tuning is a form of Transfer Learning where the base knowledge is refined for the new task.

Connection with Fine-Tuning LLMs

- Large Language Models (LLMs) are typically pre-trained on massive text corpora.
- Fine-tuning adjusts the pre-trained LLM to perform specific tasks or adapt to specialized datasets.
- Fine-tuning is a form of Transfer Learning where the base knowledge is refined for the new task.
- This approach significantly improves performance on domain-specific problems.

Fine-Tuning DistilGPT-2

- DistilGPT-2 is a smaller, faster version of GPT-2 that retains much of the performance.

Fine-Tuning DistilGPT-2

- DistilGPT-2 is a smaller, faster version of GPT-2 that retains much of the performance.
- Fine-tuning DistilGPT-2 involves training it on a custom dataset to specialize it.

Fine-Tuning DistilGPT-2

- DistilGPT-2 is a smaller, faster version of GPT-2 that retains much of the performance.
- Fine-tuning DistilGPT-2 involves training it on a custom dataset to specialize it.
- Example steps:
  - Load pre-trained DistilGPT-2.

Fine-Tuning DistilGPT-2

- DistilGPT-2 is a smaller, faster version of GPT-2 that retains much of the performance.
- Fine-tuning DistilGPT-2 involves training it on a custom dataset to specialize it.
- Example steps:
  - Load pre-trained DistilGPT-2.
  - Prepare a text dataset.

Fine-Tuning DistilGPT-2

- DistilGPT-2 is a smaller, faster version of GPT-2 that retains much of the performance.
- Fine-tuning DistilGPT-2 involves training it on a custom dataset to specialize it.
- Example steps:
  - Load pre-trained DistilGPT-2.
  - Prepare a text dataset.
  - Use a Trainer to fine-tune the model.

Fine-Tuning DistilGPT-2

- DistilGPT-2 is a smaller, faster version of GPT-2 that retains much of the performance.
- Fine-tuning DistilGPT-2 involves training it on a custom dataset to specialize it.
- Example steps:
  - Load pre-trained DistilGPT-2.
  - Prepare a text dataset.
  - Use a Trainer to fine-tune the model.
- This makes DistilGPT-2 effective for resource-limited applications.

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

- **Step 2: Prepare Dataset**

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

- **Step 2: Prepare Dataset**

Prepare a small custom dataset using Python dictionary and load it with:
`dataset = Dataset.from_dict(data)`

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

- **Step 2: Prepare Dataset**

Prepare a small custom dataset using Python dictionary and load it with:
`dataset = Dataset.from_dict(data)`

- **Step 3: Load and Configure Tokenizer**

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

- **Step 2: Prepare Dataset**

Prepare a small custom dataset using Python dictionary and load it with:
`dataset = Dataset.from_dict(data)`

- **Step 3: Load and Configure Tokenizer**

`tokenizer = GPT2Tokenizer.from_pretrained('distilgpt2')`
Add padding token: `tokenizer.pad_token = tokenizer.eos_token`

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

- **Step 2: Prepare Dataset**

Prepare a small custom dataset using Python dictionary and load it with:
`dataset = Dataset.from_dict(data)`

- **Step 3: Load and Configure Tokenizer**

`tokenizer = GPT2Tokenizer.from_pretrained('distilgpt2')`
Add padding token: `tokenizer.pad_token = tokenizer.eos_token`

- **Step 4: Load Model**

Fine-Tuning DistilGPT2: Steps 1–4

- **Step 1: Import Required Libraries**

```
from datasets import Dataset
from transformers import GPT2Tokenizer, AutoModelForCausalLM, Trainer,
TrainingArguments, DataCollatorForLanguageModeling
```

- **Step 2: Prepare Dataset**

Prepare a small custom dataset using Python dictionary and load it with:
`dataset = Dataset.from_dict(data)`

- **Step 3: Load and Configure Tokenizer**

`tokenizer = GPT2Tokenizer.from_pretrained('distilgpt2')`
Add padding token: `tokenizer.pad_token = tokenizer.eos_token`

- **Step 4: Load Model**

`model = AutoModelForCausalLM.from_pretrained('distilgpt2')`
Resize token embeddings: `model.resize_token_embeddings(len(tokenizer))`

Fine-Tuning DistilGPT2: Steps 5–7

- **Step 5: Tokenize Dataset**

Fine-Tuning DistilGPT2: Steps 5–7

- **Step 5: Tokenize Dataset**

Write a tokenization function using max length and padding.
Apply tokenization: `dataset.map(...)`

Fine-Tuning DistilGPT2: Steps 5–7

- **Step 5: Tokenize Dataset**

Write a tokenization function using max length and padding.
Apply tokenization: `dataset.map(...)`

- **Step 6: Prepare Data Collator**

Fine-Tuning DistilGPT2: Steps 5–7

- **Step 5: Tokenize Dataset**

Write a tokenization function using max length and padding.
Apply tokenization: `dataset.map(...)`

- **Step 6: Prepare Data Collator**

Use `DataCollatorForLanguageModeling` with `mlm=False` for causal language modeling.

Fine-Tuning DistilGPT2: Steps 5–7

- **Step 5: Tokenize Dataset**

Write a tokenization function using max length and padding.
Apply tokenization: `dataset.map(...)`

- **Step 6: Prepare Data Collator**

Use `DataCollatorForLanguageModeling` with `mlm=False` for causal language modeling.

- **Step 7: Define Training Arguments**

Fine-Tuning DistilGPT2: Steps 5–7

- **Step 5: Tokenize Dataset**

Write a tokenization function using max length and padding.
Apply tokenization: `dataset.map(...)`

- **Step 6: Prepare Data Collator**

Use `DataCollatorForLanguageModeling` with `mlm=False` for causal language modeling.

- **Step 7: Define Training Arguments**

Set output directory, batch size, number of epochs, save steps, and logging steps using `TrainingArguments`.

Fine-Tuning DistilGPT2: Steps 8–10

- **Step 8: Initialize Trainer**

Fine-Tuning DistilGPT2: Steps 8–10

- **Step 8: Initialize Trainer**

Pass model, training arguments, dataset, and data collator to the `Trainer`.

Fine-Tuning DistilGPT2: Steps 8–10

- **Step 8: Initialize Trainer**

Pass model, training arguments, dataset, and data collator to the `Trainer`.

- **Step 9: Train the Model**

Fine-Tuning DistilGPT2: Steps 8–10

- **Step 8: Initialize Trainer**

Pass model, training arguments, dataset, and data collator to the `Trainer`.

- **Step 9: Train the Model**

Call `trainer.train()` to start fine-tuning.

Fine-Tuning DistilGPT2: Steps 8–10

- **Step 8: Initialize Trainer**

Pass model, training arguments, dataset, and data collator to the `Trainer`.

- **Step 9: Train the Model**

Call `trainer.train()` to start fine-tuning.

- **Step 10: Save Fine-Tuned Model**

Fine-Tuning DistilGPT2: Steps 8–10

- **Step 8: Initialize Trainer**

Pass model, training arguments, dataset, and data collator to the `Trainer`.

- **Step 9: Train the Model**

Call `trainer.train()` to start fine-tuning.

- **Step 10: Save Fine-Tuned Model**

Save the trained model and tokenizer using `model.save_pretrained(...)` and `tokenizer.save_pretrained(...)`

Reach PostNetwork Academy

Website

www.postnetwork.co

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

LinkedIn Page

www.linkedin.com/company/postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

LinkedIn Page

www.linkedin.com/company/postnetworkacademy

GitHub Repositories

www.github.com/postnetworkacademy

Thank You!