Fine-tuning DeepSeek-R1 (distilled Llama) and running it locally 👉

```python
from IPython.display import Markdown
import ollama

response = ollama.chat(model="deepseek_finetuned_model",
                       messages=[{"role": "user",
                                  "content": "How to add chart to a document?"},
                       ])

Markdown(response.message.content)
```

Finetuned model

To add a chart to a document, follow these steps:

1. **Insert a Table**: Start by inserting a table into the document. You can do this using the 'Table' tool in most word processors.
2. **Insert Data**: Add data into the table. Ensure that your data is properly formatted and organized before adding the chart.
3. **Choose a Chart Type**: Select the type of chart you want to create from the available options (e.g., bar chart, pie chart, line graph, etc.).
4. **Edit the Chart Data**: Add the necessary data points and formatting to the chart using the chart editor that appears once the chart is selected.
5. **Format the Table and Chart Together**: Make sure the table and chart work well together by adjusting alignment, spacing, and other design elements as needed.

For more detailed instructions, you may want to consult a guide or use a tool such as Microsoft Word's chart features.

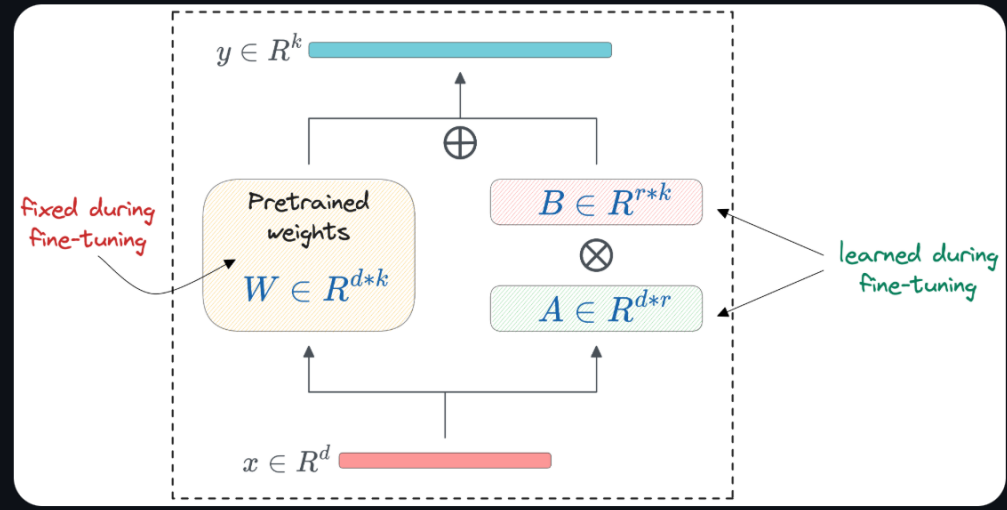
1) Load model

deepseek

```
# pip install unsloth

from unsloth import FastLanguageModel
import torch

MODEL = "unsloth/DeepSeek-R1-Distill-Llama-8B-unsloth-bnb-4bit"

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = MODEL,
    max_seq_length = 2048,
    dtype = None,
    load_in_4bit = True,
)
```

2) Define LoRA config

```
model = FastLanguageModel.get_peft_model(
    model,
    r = 4,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj"],
    use_gradient_checkpointing = "unsloth",
    lora_alpha = 16,
    lora_dropout = 0,
    bias = "none",
    use_rslora = False,
    loftq_config = None
)
```



3) Prepare dataset

```
from datasets import load_dataset
from unsloth import to_sharegpt
from unsloth import standardize_sharegpt

dataset = load_dataset("vicgalle/alpaca-gpt4", split = "train")

dataset = to_sharegpt(
    dataset,
    merged_prompt = "{instruction}[[\nYour input is:\n{input}]]",
    output_column_name = "output",
    conversation_extension = 3,
)

dataset = standardize_sharegpt(dataset)
```

4) Define Trainer

```
from trl import SFTTrainer
from transformers import TrainingArguments

trainer = SFTTrainer(model = model,
                     tokenizer = tokenizer,
                     train_dataset = dataset,
                     ...
                     args = TrainingArguments(
                            per_device_train_batch_size = 2,
                            gradient_accumulation_steps = 4,
                            max_steps = 60,
                            learning_rate = 2e-4,
                            ...
                            optim = "adamw_8bit",
                            weight_decay = 0.01,
                     ))
```

5) Train

```
trainer_stats = trainer.train()
```

```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs = 1
   \\   /|    Num examples = 52,002 | Num Epochs = 1
0^0/ \_/ \    Batch size per device = 2 | Gradient Accumulation steps = 4
\        /    Total batch size = 8 | Total steps = 60
 "-____-"     Number of trainable parameters = 3,407,872
```

[60/60 16:37, Epoch 0/1]

Step	Training Loss	Step	Training Loss		Step	Training Loss
1	1.903300	11	1.548500		51	1.266000
2	1.925000	12	1.584900		52	1.279400
3	1.821100	13	1.344100		53	1.275300
4	2.015100	14	1.437000		54	1.185500
5	2.090400	15	1.470300	....	55	1.206000
6	1.788900	16	1.413800		56	1.245900
7	1.760200	17	1.313900		57	1.229400
8	1.672000	18	1.476200		58	1.234900
9	1.697000	19	1.355700		59	1.179900
10	1.522400	20	1.336700		60	1.228300

6) Export to Ollama


```
# install Ollama
!curl -fsSL https://ollama.com/install.sh | sh

# Save model and tokenizer
model.save_pretrained_gguf("model", tokenizer)

# create a fine-tuned model
!ollama create deepseek_finetuned_model -f ./model/Modelfile
```

Run it locally with Ollama

```python
from IPython.display import Markdown
import ollama

response = ollama.chat(model="deepseek_finetuned_model",
                       messages=[{"role": "user",
                                  "content": "How to add chart to a document?"},
                       ])

Markdown(response.message.content)
```

Finetuned model

To add a chart to a document, follow these steps:

1. **Insert a Table**: Start by inserting a table into the document. You can do this using the 'Table' tool in most word processors.
2. **Insert Data**: Add data into the table. Ensure that your data is properly formatted and organized before adding the chart.
3. **Choose a Chart Type**: Select the type of chart you want to create from the available options (e.g., bar chart, pie chart, line graph, etc.).
4. **Edit the Chart Data**: Add the necessary data points and formatting to the chart using the chart editor that appears once the chart is selected.
5. **Format the Table and Chart Together**: Make sure the table and chart work well together by adjusting alignment, spacing, and other design elements as needed.

For more detailed instructions, you may want to consult a guide or use a tool such as Microsoft Word's chart features.