同济大学大模型创新实践课外发实验结果汇总报告

2151140 王谦

0.关于部分截图并非在线截图

部分实验在执行完之后就把notebook删除了，但是保存了执行完成的ipynb文件，因此部分内容是使用vscode打开的截图，但是内容和结果没有问题。

1.昇腾AI大模型入门

1.1 基于MindSpore的手写体数字识别

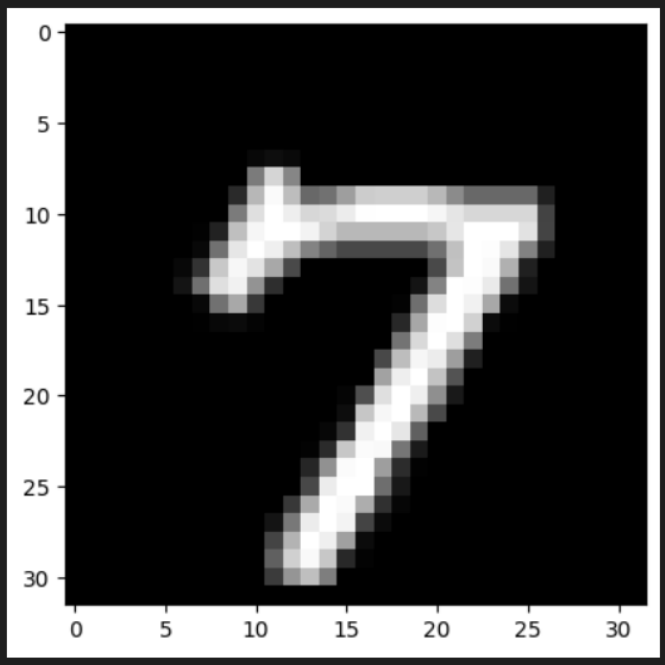
读取任意一个数据内容，观察打印结果：

```
data_next = dataset_train.create_dict_iterator(output_numpy=True).__next__()
print('Batch Size/通道数/图像长/宽：', data_next['image'].shape)
print('图像的标签样式：', data_next['label'])

plt.figure()
plt.imshow(data_next['image'][1,...].squeeze(), cmap="gray")
plt.grid(False)
plt.show()
```

[9]

```
Batch Size/通道数/图像长/宽： (32, 1, 32, 32)
图像的标签样式： [1 7 2 4 3 5 7 6 3 3 2 7 8 8 2 9 6 4 1 7 7 6 2 7 8 9 1 2 1 0 6 7]
```



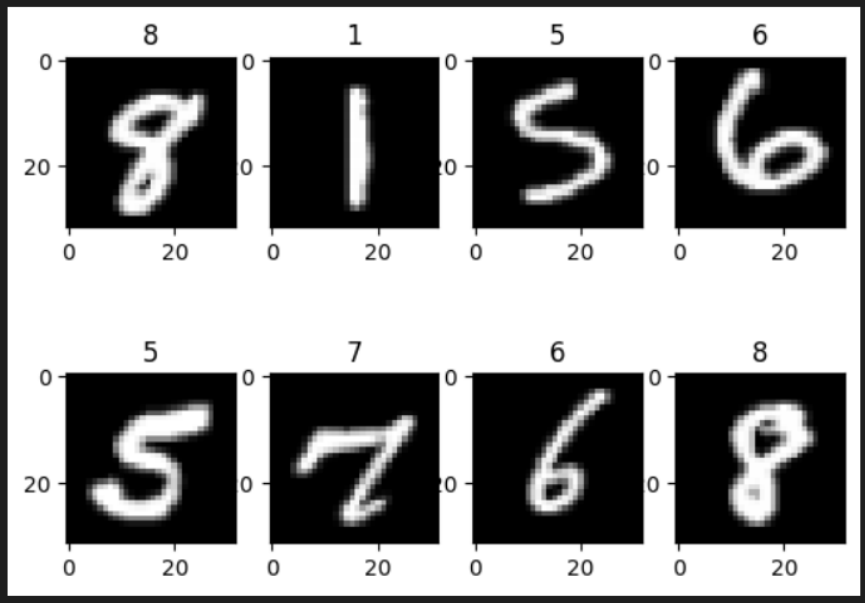
预测可视化输出：

```python
# 将模型参数存入parameter的字典中，采用load_checkpoint接口加载模型参数
param_dict = ms.load_checkpoint("./save_direct.ckpt")
# 重新定义一个LeNet5神经网络
net = network
# 将参数加载到网络中
ms.load_param_into_net(net, param_dict)
model = Model(net)
data_test = dataset_eval.create_dict_iterator()
data = next(data_test)
images = data["image"].asnumpy()
labels = data["label"].asnumpy()

# 使用函数model.predict预测image对应分类
output = model.predict(ms.Tensor(data['image']))
pred = np.argmax(output.asnumpy(), axis=1)

plt.figure()
for i in range(1, 9):
    plt.subplot(2, 4, i)
    plt.imshow(images[i-1].squeeze(), cmap="gray")
    plt.title(pred[i-1])
plt.show()
```

[13]



2.Transformer

2.1 基于Mindspore-Ascend的Transformer实现“德译英”翻译

数据集遍历迭代，逐个epoch打印训练的损失值和评估精度：

数据集遍历迭代，一次完整的数据集遍历成为一个epoch。我们逐个epoch打印训练的损失值和评估精度，并通过`save_checkpoint`保存评估精度最高的ckpt文件（transformer.ckpt）到home_path/.mindspore_examples/transformer.ckpt。

```
from mindspore import save_checkpoint

num_epochs = 10
best_valid_loss = float('inf')
ckpt_file_name = './transformer.ckpt'


for i in range(num_epochs):
    train_loss = train(train_iterator, i)
    valid_loss = evaluate(valid_iterator)

    if valid_loss < best_valid_loss:
        best_valid_loss = valid_loss
        save_checkpoint(model, ckpt_file_name)
```

[29]

```
Train   :   0%|          | 0/226 [00:00<?, ?step/s]
|
Train   : 100%|██████████| 226/226 [02:03<00:00,  1.83step/s, loss=4.52, epoch=000]
Evaluate:  88%|████████▌ | 7/8 [00:00<00:00,  7.38step/s, loss=3.44]
/
Evaluate: 100%|██████████| 8/8 [00:03<00:00,  2.23step/s, loss=3.48]
Train   : 100%|██████████| 226/226 [01:05<00:00,  3.43step/s, loss=3.07, epoch=001]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.33step/s, loss=2.55]
Train   : 100%|██████████| 226/226 [01:06<00:00,  3.42step/s, loss=2.41, epoch=002]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.38step/s, loss=2.16]
Train   : 100%|██████████| 226/226 [01:05<00:00,  3.46step/s, loss=2.02, epoch=003]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.30step/s, loss=1.94]
Train   : 100%|██████████| 226/226 [01:05<00:00,  3.46step/s, loss=1.76, epoch=004]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.20step/s, loss=1.81]
Train   : 100%|██████████| 226/226 [01:04<00:00,  3.48step/s, loss=1.56, epoch=005]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.23step/s, loss=1.73]
Train   : 100%|██████████| 226/226 [01:03<00:00,  3.54step/s, loss=1.39, epoch=006]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.07step/s, loss=1.67]
Train   : 100%|██████████| 226/226 [01:03<00:00,  3.56step/s, loss=1.25, epoch=007]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.12step/s, loss=1.64]
Train   : 100%|██████████| 226/226 [01:04<00:00,  3.50step/s, loss=1.13, epoch=008]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  6.73step/s, loss=1.61]
Train   : 100%|██████████| 226/226 [01:06<00:00,  3.39step/s, loss=1.02, epoch=009]
Evaluate: 100%|██████████| 8/8 [00:01<00:00,  7.05step/s, loss=1.59]
```

以测试数据集中的第一组语句为例进行测试：

以测试数据集中的第一组语句为例，进行测试。

```
example_idx = 0

src = test_dataset[example_idx][0]
trg = test_dataset[example_idx][1]
pred_trg = inference(src)

print(f'src = {src}')
print(f'trg = {trg}')
print(f"predicted trg = {pred_trg}")
```

[32]

```
src = ['ein', 'mann', 'mit', 'einem', 'orangefarbenen', 'hut', ',', 'der', 'etwas', 'anstarrt', '.']
trg = ['a', 'man', 'in', 'an', 'orange', 'hat', 'starring', 'at', 'something', '.']
predicted trg = ['a', 'man', 'in', 'an', 'orange', 'hat', 'is', '<unk>', 'something', '.']
```

BLUE得分：

BLEU得分

双语替换评测得分（bilingual evaluation understudy，BLEU）为衡量文本翻译模型生成出来的语句好坏的一种算法，它的核心在于评估机器翻译的译文
译文进行比较，计算出各个片段的的分数，并配以权重进行加和，基本规则为：

1. 惩罚过短的预测，即如果机器翻译出来的译文相对于人工翻译的参考译文过于短小，则命中率越高，需要施加更多的惩罚；
2. 对长段落匹配更高的权重，即如果出现长段落的完全命中，说明机器翻译的译文更贴近人工翻译的参考译文；

BLEU的公式如下：

$$exp(min(0, 1 - \frac{len(\text{label})}{len(\text{pred})})\Pi_{n=1}^{k} p_n^{1/2^n})$$

- `len(label)`：人工翻译的译文长度
- `len(pred)`：机器翻译的译文长度
- `p_n`：n-gram的精度

我们可以调用`nltk`中的`corpus_bleu`函数来计算BLEU。

```
from nltk.translate.bleu_score import corpus_bleu

def calculate_bleu(dataset, max_len=50):
    trgs = []
    pred_trgs = []

    for data in dataset[:10]:

        src = data[0]
        trg = data[1]

        pred_trg = inference(src, max_len)
        pred_trgs.append(pred_trg)
        trgs.append([trg])

    return corpus_bleu(trgs, pred_trgs)

bleu_score = calculate_bleu(test_dataset)

print(f'BLEU score = {bleu_score*100:.2f}')
```

[33]

```
BLEU score = 45.87
```

3.Bert&GPT

3.1 基于Mindspore-Ascend的bert模型微调与推理实验

模型构建训练：

```
# start training
trainer.run(tgt_columns="labels")
```

[14]

```
The train will start from the checkpoint saved in 'checkpoint'.
Epoch 0:   0%|          | 0/302 [00:00<?, ?it/s]
/
Epoch 0: 100%|▉▉▉▉▉▉▉▉▉| 301/302 [03:24<00:00,  1.99it/s, loss=0.34207344]
-
Epoch 0: 100%|▉▉▉▉▉▉▉▉▉| 302/302 [03:29<00:00,  1.44it/s, loss=0.34212732]
Checkpoint: 'bert_emotect_epoch_0.ckpt' has been saved in epoch: 0.
Evaluate:  97%|▉▉▉▉▉▉▉▉▉| 33/34 [00:04<00:00,  7.78it/s]
|
Evaluate: 100%|▉▉▉▉▉▉▉▉▉| 34/34 [00:15<00:00,  2.22it/s]
Evaluate Score: {'Accuracy': 0.9398148148148148}
---------------Best Model: 'bert_emotect_best.ckpt' has been saved in epoch: 0.---------------
Epoch 1: 100%|▉▉▉▉▉▉▉▉▉| 302/302 [02:39<00:00,  1.90it/s, loss=0.19011086]
Checkpoint: 'bert_emotect_epoch_1.ckpt' has been saved in epoch: 1.
Evaluate: 100%|▉▉▉▉▉▉▉▉▉| 34/34 [00:04<00:00,  6.91it/s]
Evaluate Score: {'Accuracy': 0.9685185185185186}
---------------Best Model: 'bert_emotect_best.ckpt' has been saved in epoch: 1.---------------
Epoch 2: 100%|▉▉▉▉▉▉▉▉▉| 302/302 [02:37<00:00,  1.91it/s, loss=0.1225593]
The maximum number of stored checkpoints has been reached.
Checkpoint: 'bert_emotect_epoch_2.ckpt' has been saved in epoch: 2.
Evaluate: 100%|▉▉▉▉▉▉▉▉▉| 34/34 [00:04<00:00,  7.02it/s]
Evaluate Score: {'Accuracy': 0.9787037037037037}
---------------Best Model: 'bert_emotect_best.ckpt' has been saved in epoch: 2.---------------
Epoch 3: 100%|▉▉▉▉▉▉▉▉▉| 302/302 [02:37<00:00,  1.92it/s, loss=0.088487014]
The maximum number of stored checkpoints has been reached.
Checkpoint: 'bert_emotect_epoch_3.ckpt' has been saved in epoch: 3.
Evaluate: 100%|▉▉▉▉▉▉▉▉▉| 34/34 [00:05<00:00,  6.74it/s]
Evaluate Score: {'Accuracy': 0.9824074074074074}
---------------Best Model: 'bert_emotect_best.ckpt' has been saved in epoch: 3.---------------
Epoch 4: 100%|▉▉▉▉▉▉▉▉▉| 302/302 [02:38<00:00,  1.91it/s, loss=0.06332478]
The maximum number of stored checkpoints has been reached.
Checkpoint: 'bert_emotect_epoch_4.ckpt' has been saved in epoch: 4.
Evaluate: 100%|▉▉▉▉▉▉▉▉▉| 34/34 [00:05<00:00,  6.48it/s]
Evaluate Score: {'Accuracy': 0.9888888888888889}
---------------Best Model: 'bert_emotect_best.ckpt' has been saved in epoch: 4.---------------
Loading best model from 'checkpoint' with '['Accuracy']': [0.9888888888888889]...
---------------The model is already load the best model from 'bert_emotect_best.ckpt'.---------------
```

模型验证：

模型验证

将验证数据集加再进训练好的模型，对数据集进行验证，查看模型在验证数据上面的效果，此处的评价指标为准确率。

```
evaluator = Evaluator(network=model, eval_dataset=dataset_test, metrics=metric)
evaluator.run(tgt_columns="labels")
```

[15]

```
Evaluate:  97%|▉▉▉▉▉▉▉▉▉| 32/33 [00:04<00:00,  7.28it/s]
-
Evaluate: 100%|▉▉▉▉▉▉▉▉▉| 33/33 [00:15<00:00,  2.14it/s]
Evaluate Score: {'Accuracy': 0.9083011583011583}
```

模型推理：

模型推理

遍历推理数据集，将结果与标签进行统一展示。

[20]
```
dataset_infer = SentimentDataset("data/infer.tsv")
```

[17]
```
def predict(text, label=None):
    label_map = {0: "消极", 1: "中性", 2: "积极"}

    text_tokenized = Tensor([tokenizer(text).input_ids])
    logits = model(text_tokenized)
    predict_label = logits[0].asnumpy().argmax()
    info = f"inputs: '{text}', predict: '{label_map[predict_label]}'"
    if label is not None:
        info += f" , label: '{label_map[label]}'"
    print(info)
```

[18]
```
from mindspore import Tensor

for label, text in dataset_infer:
    predict(text, label)
```

```
inputs: '我 要 客观', predict: '中性' , label: '中性'
inputs: '靠 你 真是 说 废话 吗', predict: '消极' , label: '消极'
inputs: '口嗅 会', predict: '中性' , label: '中性'
inputs: '每次 是 表妹 带 窝 飞 因为 窝路痴', predict: '中性' , label: '中性'
inputs: '别说 废话 我 问 你 个 问题', predict: '消极' , label: '消极'
inputs: '4967 是 新加坡 那 家 银行', predict: '中性' , label: '中性'
inputs: '是 我 喜欢 兔子', predict: '积极' , label: '积极'
inputs: '你 写 过 黄山 奇石 吗', predict: '中性' , label: '中性'
inputs: '一个一个 慢慢来', predict: '中性' , label: '中性'
inputs: '我 玩 过 这个 一点 都 不 好玩', predict: '消极' , label: '消极'
inputs: '网上 开发 女孩 的 QQ', predict: '中性' , label: '中性'
inputs: '背 你 猜 对 了', predict: '中性' , label: '中性'
inputs: '我 讨厌 你 ， 哼哼 哼 。 。', predict: '消极' , label: '消极'
```

自定义推理数据集：

自定义推理数据集

自己输入推理数据，展示模型的泛化能力。

[19]
```
predict("家人们咱就是说一整个无语住了 绝绝子叠buff")
```

```
inputs: '家人们咱就是说一整个无语住了 绝绝子叠buff', predict: '中性'
```

3.2 基于Mindspore-Ascend的gpt模型文本分类实验

训练结果：

```
[13]
trainer.run(tgt_columns="labels")

The train will start from the checkpoint saved in 'checkpoint'.
Epoch 0:   0%|          | 0/2500 [00:00<?, ?it/s]
|
Epoch 0: 100%|██████████| 2500/2500 [47:00<00:00,  1.13s/it, loss=0.2794315]
Checkpoint: 'gpt_imdb_finetune_epoch_0.ckpt' has been saved in epoch: 0.
Evaluate: 100%|██████████| 2500/2500 [06:03<00:00,  6.87it/s]
Evaluate Score: {'Accuracy': 0.9398}
---------------Best Model: 'gpt_imdb_finetune_best.ckpt' has been saved in epoch: 0.---------------
Loading best model from 'checkpoint' with '['Accuracy']': [0.9398]...
---------------The model is already load the best model from 'gpt_imdb_finetune_best.ckpt'.---------------
```

精度校验结果：

```
[14]
evaluator = Evaluator(network=model, eval_dataset=dataset_test, metrics=metric)
evaluator.run(tgt_columns="labels")

Evaluate: 100%|██████████| 6250/6250 [14:30<00:00,  7.18it/s]
Evaluate Score: {'Accuracy': 0.92412}
```

3.3 【课外拓展】基于MindSpore的GPT2文本摘要

模型训练：

```
[20]
trainer.run(tgt_columns="labels")

The train will start from the checkpoint saved in 'checkpoint'.
Epoch 0:   0%|          | 0/3 [00:00<?, ?it/s]
-
[ERROR] CORE(1772,ffffa6f350b0,python):2024-07-11-14:11:50.321.222 [mindspore/cor
[ERROR] CORE(1772,ffffa6f350b0,python):2024-07-11-14:11:50.321.748 [mindspore/cor
[ERROR] CORE(1772,ffffa6f350b0,python):2024-07-11-14:11:50.322.264 [mindspore/cor
[ERROR] CORE(1772,ffffa6f350b0,python):2024-07-11-14:11:50.322.304 [mindspore/cor
\
Epoch 0:  67%|▌      | 2/3 [01:48<00:45, 45.68s/it, loss=10.080887]
/
Epoch 0: 100%|██████████| 3/3 [02:01<00:00, 40.60s/it, loss=9.918955]
Checkpoint: 'gpt2_summarization_epoch_0.ckpt' has been saved in epoch: 0.
```

模型推理：

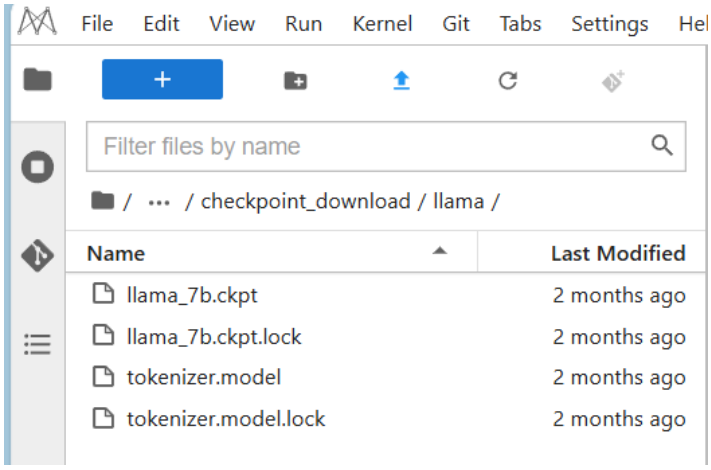
```
[23]
print(next(batched_test_dataset.create_tuple_iterator(output_numpy=True)))
```

```
[array([[ 101, 3173, 1290, 5381, 1266,  776,  130, 3299, 8153, 3189, 4510,
         8020, 6381, 5442, 3864, 7208, 8021, 1392, 3175, 7770, 2428, 1068,
         3800, 4638, 3330, 3378, 3378, 5023,  782, 2487, 1960, 3428, 8153,
         3189,  677, 1286, 1762, 1266,  776, 2356, 3862, 3895, 1277, 3791,
         7368,  671, 2144, 2146, 1161, 8024, 3791, 7368,  809, 2487, 1960,
         5389, 1146, 1166, 1161, 1905, 6158, 1440,  782, 3330, 3378, 3378,
         3300, 3309, 2530, 1152, 8108, 2399,  510, 4374, 3378, 5023,  125,
          782, 3300, 3309, 2530, 1152, 8110, 2399, 5635,  124, 2399, 8020,
         5353, 1152,  124, 2399, 8021,  679, 5023, 4638, 3300, 3309, 2530,
         1152,  511, 3428,  816, 2146, 1161, 1400, 8024, 5596, 6380, 5381,
         5018,  671, 3198, 7313, 4324, 2157,  683, 6393, 3330, 1921,  671,
         3791, 2526, 7560, 7309,  510, 3173, 7319, 1355, 6241,  782, 1065,
         1469, 2526, 2360, 2526, 2360,  511, 1065, 1469, 2526, 2360, 6134,
         4850, 1161, 1152, 1922, 7028, 8024, 3330, 2157, 5507, 2137,  833,
          677, 6401,  511, 1369, 2945,  776, 1290, 3198, 2845, 6381, 5442,
         2476, 3902, 4386,  523, 3330, 3378, 3378, 2526, 2360, 7357, 3364,
         6134, 4850, 6206,  677, 6401,  524, 7357, 3364, 6134, 4850, 8024,
         6206,  677, 6401, 8024, 1780, 2898, 3187, 5389, 6796, 2844, 8024,
          679, 5543,  809, 1366,  897, 2137, 5389, 8024, 6206, 4692, 2145,
         6225, 6395, 2945,  511, 5632,  791, 2399,  123, 3299, 3428, 1355,
          809, 3341, 8024, 6821, 6629, 3428,  816,  912, 6822, 1057, 1062,
          830, 6228, 7029,  511, 2215, 1071, 1762, 6822, 1057, 1385, 3791,
         4923, 2415,  809, 1400, 8024, 6821, 6629, 3428,  816, 3291, 3221,
         3797, 4073,  679, 3171,  511, 3330, 3378, 3378, 5023,  782, 4638,
         6121,  711, 3221, 2487, 1960, 6820, 3221, 2069, 2035, 8043, 6158,
 ...
         8024, 3766, 3300,  680, 6158, 2154,  782, 1355, 4495, 2595, 1068,
         5143,  100,  511, 2190, 3634, 3791, 7368, 6134, 4850, 8024, 6006,
         4197,  794, 6158, 2154,  782, 1079, 6175,  677, 3766, 3300,  102]],
       dtype=int64), array(['法院：李天一案判定李10年刑期属从轻处罚；李律师称要上诉：不能以口供定罪'], dtype='<U36')]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

4.LLaMa&Glm大模型

4.1 基于Mindspore-Ascend的llama7b模型推理与微调实验

下载文件：



pipeline_infer.py推理脚本直接推理：

```
2024-07-11 15:20:23,371 - mindformers - WARNING - The user passed the custom defined activation function True. If the user want to enable shard for the activatio
n cell, the user should set the shard for each primitives in the cell.
2024-07-11 15:20:26,232 - mindformers - WARNING - The user passed the custom defined activation function True. If the user want to enable shard for the activatio
n cell, the user should set the shard for each primitives in the cell.
2024-07-11 15:20:28,271 - mindformers - INFO - start to read the ckpt file: 13476850247
2024-07-11 15:22:15,490 - mindformers - INFO - weights in ./checkpoint_download/llama/llama_7b.ckpt are loaded
[{'text_generation_text': ['I love china, because it is so beautiful and delicate. I love to use it for my wedding.\nI love china, because it is so beautiful and
 delicate. I love to use it for my wedding. I love china, because it']}]
(MindSpore) [ma-user mindformers]$
```

数据预处理：

ma-user@notebook-fafa30cl × alpaca_data.json ×

Filter...


```
▼ root: [] 10 items
  ▼ 0:
      instruction: "What is the smallest bridge in the world?"
      input: ""
      output: "The bridge of a nose."
  ▼ 1:
      instruction: "What does 'thank you dog' mean?"
      input: ""
      output: "This meme is from a video on Bilibili. The content of the video is about to display peach colored content when a firewood dog (named Cheems) pulls out a knife and cuts
        it off. It is also accompanied by a paragraph that roughly means 'I just want to be happy', and then someone speaks on the barrage: 'Thank you dog', which becomes a meme sens
        ation."
  ▶ 2:
  ▶ 3:
  ▶ 4:
  ▶ 5:
  ▶ 6:
  ▶ 7:
  ▶ 8:
  ▶ 9:
```


训练：

ma-user@notebook-fafa30cl × alpaca_data.json ×

```
e, loss_scale: 65536.0
2024-07-11 16:01:10,039 - mindformers - INFO - Per sink_size step time: 11364.213 ms, per step time: 2272.843 ms, avg loss: 1.014
2024-07-11 16:01:21,396 - mindformers - INFO - Epoch:[  7/ 20], step:[    5/    5], loss:[2.803/2.803], time:11329.703 ms, lr:[7.5466e-05], overflow cond: False,
 loss_scale: 65536.0
2024-07-11 16:01:21,397 - mindformers - INFO - Per sink_size step time: 11357.257 ms, per step time: 2271.451 ms, avg loss: 2.803
2024-07-11 16:01:32,753 - mindformers - INFO - Epoch:[  8/ 20], step:[    5/    5], loss:[2.690/2.690], time:11332.352 ms, lr:[6.8195244e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:01:32,754 - mindformers - INFO - Per sink_size step time: 11356.890 ms, per step time: 2271.378 ms, avg loss: 2.690
2024-07-11 16:01:44,113 - mindformers - INFO - Epoch:[  9/ 20], step:[    5/    5], loss:[0.697/0.697], time:11335.179 ms, lr:[6.0448376e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:01:44,114 - mindformers - INFO - Per sink_size step time: 11359.619 ms, per step time: 2271.924 ms, avg loss: 0.697
2024-07-11 16:01:55,472 - mindformers - INFO - Epoch:[ 10/ 20], step:[    5/    5], loss:[0.032/0.032], time:11333.325 ms, lr:[5.2428113e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:01:55,473 - mindformers - INFO - Per sink_size step time: 11358.604 ms, per step time: 2271.721 ms, avg loss: 0.032
2024-07-11 16:02:06,836 - mindformers - INFO - Epoch:[ 11/ 20], step:[    5/    5], loss:[0.049/0.049], time:11338.125 ms, lr:[4.434431e-05], overflow cond: Fals
e, loss_scale: 65536.0
2024-07-11 16:02:06,837 - mindformers - INFO - Per sink_size step time: 11363.820 ms, per step time: 2272.764 ms, avg loss: 0.049
2024-07-11 16:02:18,192 - mindformers - INFO - Epoch:[ 12/ 20], step:[    5/    5], loss:[0.231/0.231], time:11330.456 ms, lr:[3.6408495e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:02:18,193 - mindformers - INFO - Per sink_size step time: 11356.301 ms, per step time: 2271.260 ms, avg loss: 0.231
2024-07-11 16:02:29,565 - mindformers - INFO - Epoch:[ 13/ 20], step:[    5/    5], loss:[1.677/1.677], time:11346.498 ms, lr:[2.8828328e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:02:29,567 - mindformers - INFO - Per sink_size step time: 11372.926 ms, per step time: 2274.585 ms, avg loss: 1.677
2024-07-11 16:02:40,930 - mindformers - INFO - Epoch:[ 14/ 20], step:[    5/    5], loss:[0.009/0.009], time:11338.365 ms, lr:[2.180215e-05], overflow cond: Fals
e, loss_scale: 65536.0
2024-07-11 16:02:40,931 - mindformers - INFO - Per sink_size step time: 11364.053 ms, per step time: 2272.811 ms, avg loss: 0.009
2024-07-11 16:02:52,295 - mindformers - INFO - Epoch:[ 15/ 20], step:[    5/    5], loss:[0.002/0.002], time:11335.824 ms, lr:[1.5513804e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:02:52,296 - mindformers - INFO - Per sink_size step time: 11365.086 ms, per step time: 2273.017 ms, avg loss: 0.002
2024-07-11 16:03:03,664 - mindformers - INFO - Epoch:[ 16/ 20], step:[    5/    5], loss:[0.092/0.092], time:11340.351 ms, lr:[1.0127848e-05], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:03:03,665 - mindformers - INFO - Per sink_size step time: 11368.605 ms, per step time: 2273.721 ms, avg loss: 0.092
2024-07-11 16:03:15,024 - mindformers - INFO - Epoch:[ 17/ 20], step:[    5/    5], loss:[0.066/0.066], time:11334.468 ms, lr:[5.785209e-06], overflow cond: Fals
e, loss_scale: 65536.0
2024-07-11 16:03:15,024 - mindformers - INFO - Per sink_size step time: 11358.928 ms, per step time: 2271.786 ms, avg loss: 0.066
2024-07-11 16:03:26,391 - mindformers - INFO - Epoch:[ 18/ 20], step:[    5/    5], loss:[0.003/0.003], time:11341.792 ms, lr:[2.5995075e-06], overflow cond: Fal
se, loss_scale: 65536.0
2024-07-11 16:03:26,392 - mindformers - INFO - Per sink_size step time: 11367.233 ms, per step time: 2273.447 ms, avg loss: 0.003
2024-07-11 16:03:37,761 - mindformers - INFO - Epoch:[ 19/ 20], step:[    5/    5], loss:[0.002/0.002], time:11343.785 ms, lr:[6.540715e-07], overflow cond: Fals
e, loss_scale: 65536.0
2024-07-11 16:03:37,762 - mindformers - INFO - Per sink_size step time: 11370.013 ms, per step time: 2274.003 ms, avg loss: 0.002
2024-07-11 16:03:49,128 - mindformers - INFO - Epoch:[ 20/ 20], step:[    5/    5], loss:[0.002/0.002], time:11341.863 ms, lr:[0.], overflow cond: False, loss_sc
ale: 65536.0
2024-07-11 16:08:56,712 - mindformers - INFO - Per sink_size step time: 318948.696 ms, per step time: 63789.739 ms, avg loss: 0.002
2024-07-11 16:08:56,718 - mindformers - INFO - .........Training Over!.............
(MindSpore) [ma-user scripts]$
```

使用微调后的模型推理：

```
ma-user@notebook-fafa30cl ×   alpaca_data.json ×

### Instruction:
Who is Huang Heyang?

### Response: Famous Young Scholars from Ten Miles and Eight Villages.
2024-07-11 16:18:48,745 - mindformers - INFO - output result is: [{'text_generation_text': ['Below is an instruction that describes a task. Write a response that
 appropriately completes the request.\n\n### Instruction:\nTell me about alpacas.\n\n### Response: They are cute and friendly.']}]
2024-07-11 16:18:48,746 - mindformers - INFO - output result is saved at: text_generation_result.txt
2024-07-11 16:18:48,746 - mindformers - INFO - .........Predict Over!............
Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
Tell me about alpacas.

### Response: They are cute and friendly.
2024-07-11 16:18:53,240 - mindformers - INFO - output result is: [{'text_generation_text': ['Below is an instruction that describes a task. Write a response that
 appropriately completes the request.\n\n### Instruction:\nWhere is Beijing?\n\n### Response: in the capital city of China.']}]
2024-07-11 16:18:53,240 - mindformers - INFO - output result is saved at: text_generation_result.txt
2024-07-11 16:18:53,240 - mindformers - INFO - .........Predict Over!............
Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
Where is Beijing?

### Response: in the capital city of China.
2024-07-11 16:18:56,171 - mindformers - INFO - output result is: [{'text_generation_text': ['Below is an instruction that describes a task. Write a response that
 appropriately completes the request.\n\n### Instruction:\nWhere does afternoon come before morning in the world?\n\n### Response: in the dictionary.']}]
2024-07-11 16:18:56,172 - mindformers - INFO - output result is saved at: text_generation_result.txt
2024-07-11 16:18:56,172 - mindformers - INFO - .........Predict Over!............
Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
Where does afternoon come before morning in the world?

### Response: in the dictionary.
2024-07-11 16:19:02,524 - mindformers - INFO - output result is: [{'text_generation_text': ["Below is an instruction that describes a task. Write a response that
 appropriately completes the request.\n\n### Instruction:\nTell me about banana.\n\n### Response: It's sweet and high in potassium."]}]
2024-07-11 16:19:02,524 - mindformers - INFO - output result is saved at: text_generation_result.txt
2024-07-11 16:19:02,524 - mindformers - INFO - .........Predict Over!............
Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
Tell me about banana.

### Response: It's sweet and high in potassium.
(MindSpore) [ma-user mindformers]$

M: 0.27/96 GB | NPU: 0% | HBM: 0% | EVS: 38.12/128 GB | NET: ↑ 0.03 / ↓ 0.05 Kb/s   ma-user@notebook-fafa30cb-84ea-46d5-a7b9-c68c2be704a0:~/work/llama_lab/mindform
```

4.2 基于Mindspore-Ascend的Glm2_6b模型推理与微调实验

使用autoclass.py推理脚本推理：

```
e will be **SAMPLE**.
2024-07-11 15:30:15,547 - mindformers[mindformers/generation/text_generator.py:724] - INFO - total time: 76.193
81070137024 s; generated tokens: 186 tokens; generate speed: 2.441143162257601 tokens/s
[{'text_generation_text': ['如何提高肺活量？ 肺活量是指在不限时间的情况下，一次最大吸气后再尽最大能力所呼出的气
体量，是反映人体生长发育水平的重要机能指标之一，是衡量肺的功能的重要标志。为了提高肺活量，我们可以采取以下措施
：\n\n1. 坚持参加有氧运动，如慢跑、游泳、深呼吸、快速步行等，可以有效地提高肺活量。\n\n2. 坚持参加集体体育锻炼
，可以提高肺活量。\n\n3. 坚持参加集体体育活动，可以提高肺活量。\n\n4. 坚持参加集体劳动，可以提高肺活量。\n\n5.
坚持参加集体游戏，可以提高肺活量。\n\n6. 坚持参加集体读书，可以提高肺活量。\n\n7. 坚持参加集体旅游，可以提高肺
活量。\n\n8. ']}]
(MindSpore) [ma-user mindformers]$
```

启动训练：

```
                                        | 16.33894 samples/s/p  0:00:40 }
2024-07-11 15:39:54,030 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 70/150], st
ep:[    1/    1], loss: 1.974, per_step_time: 489ms, lr: 0.0027333333, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:39:54,030 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   46.7% |
                                        | 16.34382 samples/s/p  0:00:39 }
2024-07-11 15:39:55,014 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 72/150], st
ep:[    1/    1], loss: 1.849, per_step_time: 489ms, lr: 0.0026666666, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:39:55,015 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   48.0% |
                                        | 16.33769 samples/s/p  0:00:38 }
2024-07-11 15:39:55,999 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 74/150], st
ep:[    1/    1], loss: 1.728, per_step_time: 489ms, lr: 0.0026000002, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:39:56,000 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   49.3% |
                                        | 16.34143 samples/s/p  0:00:37 }
2024-07-11 15:39:56,984 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 76/150], st
ep:[    1/    1], loss: 1.610, per_step_time: 489ms, lr: 0.0025333331, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:39:56,984 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   50.7% |
                                        | 16.34948 samples/s/p  0:00:36 }
2024-07-11 15:39:57,969 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 78/150], st
ep:[    1/    1], loss: 1.491, per_step_time: 489ms, lr: 0.0024666667, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:39:57,969 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   52.0% |
                                        | 16.33309 samples/s/p  0:00:35 }
2024-07-11 15:39:58,954 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 80/150], st
ep:[    1/    1], loss: 1.370, per_step_time: 489ms, lr: 0.0024, overflow cond: False, loss_scale: 65536.0
2024-07-11 15:39:58,954 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   53.3% |
                                        | 16.33719 samples/s/p  0:00:34 }
2024-07-11 15:39:59,946 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 82/150], st
ep:[    1/    1], loss: 1.249, per_step_time: 493ms, lr: 0.0023333333, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:39:59,947 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   54.7% |
                                        | 16.21386 samples/s/p  0:00:33 }
2024-07-11 15:40:00,942 - mindformers[mindformers/core/callback/callback.py:314] - INFO - { Epoch:[ 84/150], st
ep:[    1/    1], loss: 1.133, per_step_time: 495ms, lr: 0.0022666666, overflow cond: False, loss_scale: 65536.
0
2024-07-11 15:40:00,942 - mindformers[mindformers/core/callback/callback.py:324] - INFO -   56.0% |
                                        | 16.15161 samples/s/p  0:00:32 }
```

查看微调后的模型权重：

```
(MindSpore) [ma-user scripts]$cd /home/ma-user/work/mindformers/output/checkpoint/rank_0
(MindSpore) [ma-user rank_0]$ll
total 12740216
-r-------- 1 ma-user ma-group 13042350695 Jul 11 15:41 glm2-6b-ptuning2_rank_0-75_2.ckpt
-rw------- 1 ma-user ma-group     3616988 Jul 11 15:39 glm2-6b-ptuning2_rank_0-graph.meta
-rw------- 1 ma-user ma-group          89 Jul 11 15:41 meta.json
(MindSpore) [ma-user rank_0]$
```

启动推理脚本：

```
2024-07-11 15:50:11,889 - mindformers[mindformers/generation/text_generator.py:478] - INFO - total time: 4.473
40236663818 s; generated tokens: 20 tokens; generate speed: 4.47043232257086 tokens/s
好志宏是谁
志宏老师是一个领导，虽然一头白头发，但也是非常帅气，性格开朗。
2024-07-11 15:50:11,915 - mindformers[mindformers/generation/text_generator.py:1099] - WARNING - When do_sampl
 is set to False, top_k will be set to 1 and top_p will be set to 0, making them inactive.
2024-07-11 15:50:11,915 - mindformers[mindformers/generation/text_generator.py:1103] - INFO - Generation Confi
 is: {'max_length': 1024, 'max_new_tokens': None, 'num_beams': 1, 'do_sample': False, 'use_past': False, 'temp
rature': 1, 'top_k': 0, 'top_p': 1.0, 'repetition_penalty': 1, 'encoder_repetition_penalty': 1.0, 'renormalize
logits': False, 'pad_token_id': 0, 'bos_token_id': None, 'eos_token_id': 2, '_from_model_config': True}
2024-07-11 15:50:11,915 - mindformers[mindformers/generation/text_generator.py:176] - INFO - The generation mo
e will be **GREEDY_SEARCH**.
2024-07-11 15:50:15,611 - mindformers[mindformers/generation/text_generator.py:478] - INFO - total time: 3.696
0510597229 s; generated tokens: 17 tokens; generate speed: 4.599560745338281 tokens/s
陈秀鸿是谁
陈秀鸿是一个老师,讲课风格也是yyds，风格生动有趣.
2024-07-11 15:50:15,636 - mindformers[mindformers/generation/text_generator.py:1099] - WARNING - When do_sampl
 is set to False, top_k will be set to 1 and top_p will be set to 0, making them inactive.
2024-07-11 15:50:15,636 - mindformers[mindformers/generation/text_generator.py:1103] - INFO - Generation Confi
 is: {'max_length': 1024, 'max_new_tokens': None, 'num_beams': 1, 'do_sample': False, 'use_past': False, 'temp
rature': 1, 'top_k': 0, 'top_p': 1.0, 'repetition_penalty': 1, 'encoder_repetition_penalty': 1.0, 'renormalize
logits': False, 'pad_token_id': 0, 'bos_token_id': None, 'eos_token_id': 2, '_from_model_config': True}
2024-07-11 15:50:15,637 - mindformers[mindformers/generation/text_generator.py:176] - INFO - The generation mo
e will be **GREEDY_SEARCH**.
2024-07-11 15:50:21,341 - mindformers[mindformers/generation/text_generator.py:478] - INFO - total time: 5.703
627365112305 s; generated tokens: 26 tokens; generate speed: 4.558554223233415 tokens/s
闫伟是谁
闫伟老师是一个领导，有点像梁源，就是小鬼太酷辣那个，是一个笛子能手.
2024-07-11 15:50:21,366 - mindformers[mindformers/generation/text_generator.py:1099] - WARNING - When do_sampl
 is set to False, top_k will be set to 1 and top_p will be set to 0, making them inactive.
2024-07-11 15:50:21,366 - mindformers[mindformers/generation/text_generator.py:1103] - INFO - Generation Confi
 is: {'max_length': 1024, 'max_new_tokens': None, 'num_beams': 1, 'do_sample': False, 'use_past': False, 'temp
```