**Downloaded Model (with bnb)**  
Parametres:  
trainable params: 31891466 || all params: 183632906 || trainable%: 17.36696689862328

Accuracy: .09

**After Training**  
  
Parametres:  
trainable params: 786432 || all params: 184419338 || trainable%: 0.42643684145531424

Accuracy: .64

**Loaded – (bnb: True, floa16: True, 4bit: True, merge: False)**  
  
Parametres:  
trainable params: 0 || all params: 184419338 || trainable%: 0.0

Accuracy: .12

**Loaded – (bnb: True, floa16: True, 4bit: True, merge: True)**  
  
Parametres:  
trainable params: 0 || all params: 208798730 || trainable%: 0.0

Accuracy: Expected CUDA got CPU

**Loaded – (bnb: False, floa16: False, 4bit: False, merge: True)**  
  
Parametres:  
trainable params: 0 || all params: 335152138 || trainable%: 0.0

Accuracy: .19

**Loaded – (bnb: False, floa16: False, 4bit: False, merge: False)**  
  
Parametres:  
trainable params: 0 || all params: 335938570 || trainable%: 0.0

Accuracy: .10

**Test**

model\_new\_full = BertForSequenceClassification.from\_pretrained("ManuelAlv/IMDB\_Classify\_Bart\_adapters")

Parametres:  
trainable params: 0 || all params: 335938570 || trainable%: 0.0

Accuracy: .10

model\_new\_full = BertForSequenceClassification.from\_pretrained("ManuelAlv/IMDB\_Classify\_Bart\_adapters",

                                                               quantization\_config=bnb\_config,

                                                               torch\_dtype=torch.bfloat16,

                                                              load\_in\_4bit=True,

                                                               num\_labels = 10,

                                                               device\_map={"":0})

Parametres:  
trainable params: 0 || all params: 184419338 || trainable%: 0.0

Accuracy: .10

model\_new\_full = PeftModel.from\_pretrained(model\_original, model\_new\_full)

**CHANGED BY MERGING THE ACTUAL MODEL**

**WORKS**from peft import PeftConfig, PeftModel

# Load the model

peft\_model = "ManuelAlv/IMDB\_Classify\_Bart\_adapters"

config = PeftConfig.from\_pretrained(peft\_model)

model\_original = BertForSequenceClassification.from\_pretrained(

    config.base\_model\_name\_or\_path,

    quantization\_config=bnb\_config,

    torch\_dtype=torch.bfloat16,

    load\_in\_4bit=True,

    num\_labels = 10,

    device\_map={"":0}

    )

tokenizer = AutoTokenizer.from\_pretrained(config.base\_model\_name\_or\_path)

model\_new\_full = PeftModel.from\_pretrained(model\_original, peft\_model)