

1 Start coding or [generate](#) with AI.

```
1 !pip install git+https://github.com/shumingma/transformers.git
2
```

```
Collecting git+https://github.com/shumingma/transformers.git
  Cloning https://github.com/shumingma/transformers.git to /tmp/pip-req-build-_914h5_r
  Running command git clone --filter=blob:none --quiet https://github.com/shumingma/transformers.git /tmp/pip-req-build-_914h5_r
  Resolved https://github.com/shumingma/transformers.git to commit 4a01efe84d0120dc2545ff2de445082400d87407
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (0.30.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers==4.52.0.dev0) (2025.3.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers==4.52.0.dev0) (4.13.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (2025.1.31)
Building wheels for collected packages: transformers
  Building wheel for transformers (pyproject.toml) ... done
  Created wheel for transformers: filename=transformers-4.52.0.dev0-py3-none-any.whl size=11413940 sha256=e4bee5136b7d8098b43ceb969d92a596d75e7e74590d4bc99c8271240cad91c1
  Stored in directory: /tmp/pip-ephem-wheel-cache-32mew_b4/wheels/2c/2a/7c/3be0c30fb51a7becc4bcb536739ae9ed9cc7e633fbbfaf63b
Successfully built transformers
Installing collected packages: transformers
  Attempting uninstall: transformers
    Found existing installation: transformers 4.51.3
    Uninstalling transformers-4.51.3:
      Successfully uninstalled transformers-4.51.3
Successfully installed transformers-4.52.0.dev0
```

Collecting git+https://github.com/shumingma/transformers.git Cloning https://github.com/shumingma/transformers.git to /tmp/pip-req-build-_914h5_r Running command git clone --filter=blob:none --quiet https://github.com/shumingma/transformers.git /tmp/pip-req-build-_914h5_r Resolved https://github.com/shumingma/transformers.git to commit 4a01efe84d0120dc2545ff2de445082400d87407 Installing build dependencies ... done Getting requirements to build wheel ... done Preparing metadata (pyproject.toml) ... done Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (3.18.0) Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (0.30.2) Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (2.0.2) Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (24.2) Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (6.0.2) Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (2024.11.6) Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (2.32.3) Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (0.21.1) Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (0.5.3) Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers==4.52.0.dev0) (4.67.1) Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers==4.52.0.dev0) (2025.3.2) Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers==4.52.0.dev0) (4.13.2) Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (3.4.1) Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (3.10) Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (2.3.0) Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers==4.52.0.dev0) (2025.1.31) Building wheels for collected packages: transformers Building wheel for transformers (pyproject.toml) ... done Created wheel for transformers: filename=transformers-4.52.0.dev0-py3-none-any.whl size=11413940 sha256=e4bee5136b7d8098b43ceb969d92a596d75e7e74590d4bc99c8271240cad91c1 Stored in directory: /tmp/pip-ephem-wheel-cache-32mew_b4/wheels/2c/2a/7c/3be0c30fb51a7becc4bcb536739ae9ed9cc7e633fbbfaf63b Successfully built transformers Installing collected packages: transformers Attempting uninstall: transformers Found existing installation: transformers 4.51.3 Uninstalling transformers-4.51.3: Successfully uninstalled transformers-4.51.3 Successfully installed transformers-4.52.0.dev0

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

✓ [/content/bitnet-b1.58-2B-4T/configuration_bitnet.py](#)

```

1 #/content/bitnet-b1.58-2B-4T/configuration_bitnet.py
2
3
4
5 # coding=utf-8
6 # Copyright 2022 EleutherAI and the HuggingFace Inc. team. All rights reserved.
7 #
8 # This code is based on EleutherAI's GPT-NeoX library and the GPT-NeoX
9 # and OPT implementations in this library. It has been modified from its
10 # original forms to accommodate minor architectural differences compared
11 # to GPT-NeoX and OPT used by the Meta AI team that trained the model.
12 #
13 # Licensed under the Apache License, Version 2.0 (the "License");
14 # you may not use this file except in compliance with the License.
15 # You may obtain a copy of the License at
16 #
17 #     http://www.apache.org/licenses/LICENSE-2.0
18 #
19 # Unless required by applicable law or agreed to in writing, software
20 # distributed under the License is distributed on an "AS IS" BASIS,
21 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
22 # See the License for the specific language governing permissions and
23 # limitations under the License.
24 """ LLaMA model configuration"""
25
26 from transformers.configuration_utils import PretrainedConfig
27 from transformers.utils import logging
28
29
30 logger = logging.get_logger(__name__)
31
32 LLAMA_PRETRAINED_CONFIG_ARCHIVE_MAP = {}
33
34
35 class BitnetConfig(PretrainedConfig):
36     r"""
37     This is the configuration class to store the configuration of a [`BitnetModel`]. It is used to instantiate an LLaMA
38     model according to the specified arguments, defining the model architecture. Instantiating a configuration with the
39     defaults will yield a similar configuration to that of the LLaMA-7B.
40     Configuration objects inherit from [`PretrainedConfig`] and can be used to control the model outputs. Read the
41     documentation from [`PretrainedConfig`] for more information.
42
43     Args:
44         vocab_size (`int`, *optional*, defaults to 32000):
45             Vocabulary size of the LLaMA model. Defines the number of different tokens that can be represented by the
46             `inputs_ids` passed when calling [`BitnetModel`]
47         hidden_size (`int`, *optional*, defaults to 4096):
48             Dimension of the hidden representations.
49         intermediate_size (`int`, *optional*, defaults to 11008):
50             Dimension of the MLP representations.
51         num_hidden_layers (`int`, *optional*, defaults to 32):
52             Number of hidden layers in the Transformer decoder.
53         num_attention_heads (`int`, *optional*, defaults to 32):
54             Number of attention heads for each attention layer in the Transformer decoder.
55         num_key_value_heads (`int`, *optional*):
56             This is the number of key_value heads that should be used to implement Grouped Query Attention. If
57             `num_key_value_heads=num_attention_heads`, the model will use Multi Head Attention (MHA), if
58             `num_key_value_heads=1` the model will use Multi Query Attention (MQA) otherwise GQA is used. When
59             converting a multi-head checkpoint to a GQA checkpoint, each group key and value head should be constructed
60             by meanpooling all the original heads within that group. For more details checkout [this
61             paper](https://arxiv.org/pdf/2305.13245.pdf). If it is not specified, will default to
62             `num_attention_heads`.
63         hidden_act (`str` or `function`, *optional*, defaults to `"silu"`):
64             The non-linear activation function (function or string) in the decoder.
65         max_position_embeddings (`int`, *optional*, defaults to 2048):

```

```

65         The maximum sequence length that this model might ever be used with. Bitnet 1 supports up to 2048 tokens,
66         Bitnet 2 up to 4096, CodeBitnet up to 16384.
67     initializer_range (`float`, *optional*, defaults to 0.02):
68         The standard deviation of the truncated_normal_initializer for initializing all weight matrices.
69     rms_norm_eps (`float`, *optional*, defaults to 1e-06):
70         The epsilon used by the rms normalization layers.
71     use_cache (`bool`, *optional*, defaults to `True`):
72         Whether or not the model should return the last key/values attentions (not used by all models). Only
73         relevant if `config.is_decoder=True`.
74     pad_token_id (`int`, *optional*):
75         Padding token id.
76     bos_token_id (`int`, *optional*, defaults to 1):
77         Beginning of stream token id.
78     eos_token_id (`int`, *optional*, defaults to 2):
79         End of stream token id.
80     pretraining_tp (`int`, *optional*, defaults to 1):
81         Experimental feature. Tensor parallelism rank used during pretraining. Please refer to [this
82         document](https://huggingface.co/docs/transformers/main/perf_train_gpu_many#tensor-parallelism) to understand more about it
83         necessary to ensure exact reproducibility of the pretraining results. Please refer to [this
84         issue](https://github.com/pytorch/pytorch/issues/76232).
85     tie_word_embeddings (`bool`, *optional*, defaults to `False`):
86         Whether to tie weight embeddings
87     rope_theta (`float`, *optional*, defaults to 10000.0):
88         The base period of the RoPE embeddings.
89     rope_scaling (`Dict`, *optional*):
90         Dictionary containing the scaling configuration for the RoPE embeddings. Currently supports two scaling
91         strategies: linear and dynamic. Their scaling factor must be a float greater than 1. The expected format is
92         `{"type": strategy name, "factor": scaling factor}`. When using this flag, don't update
93         `max_position_embeddings` to the expected new maximum. See the following thread for more information on how
94         these scaling strategies behave:
95         https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/. This is an
96         experimental feature, subject to breaking API changes in future versions.
97     attention_bias (`bool`, defaults to `False`, *optional*, defaults to `False`):
98         Whether to use a bias in the query, key, value and output projection layers during self-attention.
99     attention_dropout (`float`, *optional*, defaults to 0.0):
100         The dropout ratio for the attention probabilities.
101
102     """
103     python
104     >>> from transformers import BitnetModel, BitnetConfig
105     >>> # Initializing a LLaMA llama-7b style configuration
106     >>> configuration = BitnetConfig()
107     >>> # Initializing a model from the llama-7b style configuration
108     >>> model = BitnetModel(configuration)
109     >>> # Accessing the model configuration
110     >>> configuration = model.config
111     """
112
113     model_type = "llama"
114     keys_to_ignore_at_inference = ["past_key_values"]
115
116     def __init__(
117         self,
118         vocab_size=32000,
119         hidden_size=4096,
120         intermediate_size=11008,
121         num_hidden_layers=32,
122         num_attention_heads=32,
123         num_key_value_heads=None,
124         hidden_act="silu",
125         max_position_embeddings=2048,
126         initializer_range=0.02,
127         rms_norm_eps=1e-6,
128         use_cache=True,
129         pad_token_id=None,
130         bos_token_id=1,
131         eos_token_id=2,
132         pretraining_tp=1,
133         tie_word_embeddings=False,
134         rope_theta=10000.0,
135         rope_scaling=None,
136         attention_bias=False,
137         attention_dropout=0.0,
138         weight_bits=1,
139         input_bits=8,
140         **kwargs,
141     ):
142         self.vocab_size = vocab_size
143         self.max_position_embeddings = max_position_embeddings

```

```

142     self.hidden_size = hidden_size
143     self.intermediate_size = intermediate_size
144     self.num_hidden_layers = num_hidden_layers
145     self.num_attention_heads = num_attention_heads
146
147     # for backward compatibility
148     if num_key_value_heads is None:
149         num_key_value_heads = num_attention_heads
150
151     self.num_key_value_heads = num_key_value_heads
152     self.hidden_act = hidden_act
153     self.initializer_range = initializer_range
154     self.rms_norm_eps = rms_norm_eps
155     self.pretraining_tp = pretraining_tp
156     self.use_cache = use_cache
157     self.rope_theta = rope_theta
158     self.rope_scaling = rope_scaling
159     self._rope_scaling_validation()
160     self.attention_bias = attention_bias
161     self.attention_dropout = attention_dropout
162     self.weight_bits = weight_bits
163     self.input_bits = input_bits
164
165     super().__init__(
166         pad_token_id=pad_token_id,
167         bos_token_id=bos_token_id,
168         eos_token_id=eos_token_id,
169         tie_word_embeddings=tie_word_embeddings,
170         **kwargs,
171     )
172
173     def _rope_scaling_validation(self):
174         """
175         Validate the `rope_scaling` configuration.
176         """
177         if self.rope_scaling is None:
178             return
179
180         if not isinstance(self.rope_scaling, dict) or len(self.rope_scaling) != 2:
181             raise ValueError(
182                 "`rope_scaling` must be a dictionary with with two fields, `type` and `factor`, "
183                 f"got {self.rope_scaling}"
184             )
185         rope_scaling_type = self.rope_scaling.get("type", None)
186         rope_scaling_factor = self.rope_scaling.get("factor", None)
187         if rope_scaling_type is None or rope_scaling_type not in ["linear", "dynamic"]:
188             raise ValueError(
189                 f"`rope_scaling`'s type field must be one of ['linear', 'dynamic'], got {rope_scaling_type}"
190             )
191         if rope_scaling_factor is None or not isinstance(rope_scaling_factor, float) or rope_scaling_factor <= 1.0:
192             raise ValueError(f"`rope_scaling`'s factor field must be a float > 1, got {rope_scaling_factor}")

```

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

```


1 import torch
2 from transformers import AutoModelForCausalLM, AutoTokenizer
3
4 model_id = "rakmik/bitnetrun"
5
6 # Load tokenizer and model
7 tokenizer = AutoTokenizer.from_pretrained(model_id)
8 model = AutoModelForCausalLM.from_pretrained(
9     model_id

```

```

9     model.to(device=device),
10     torch_dtype=torch.bfloat16
11 )
12
13 # Apply the chat template
14 messages = [
15     {"role": "system", "content": "You are a helpful AI assistant."},
16     {"role": "user", "content": "Who is Napoleon Bonaparte?"},
17 ]
18 prompt = tokenizer.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)
19 chat_input = tokenizer(prompt, return_tensors="pt").to(model.device)
20
21 # Generate response
22 chat_outputs = model.generate(**chat_input, max_new_tokens=128)
23 response = tokenizer.decode(chat_outputs[0][chat_input['input_ids'].shape[-1]:], skip_special_tokens=True) # Decode only the response part
24 print("\nAssistant Response:", response)
25

```

 /usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
 The secret `HF_TOKEN` does not exist in your Colab secrets.
 To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as secret.
 You will be able to reuse this secret in all of your notebooks.
 Please note that authentication is recommended but still optional to access public models or datasets.

```

warnings.warn(
tokenizer_config.json: 100%          50.8k/50.8k [00:00<00:00, 1.55MB/s]
tokenizer.json: 100%                9.09M/9.09M [00:00<00:00, 12.0MB/s]
special_tokens_map.json: 100%       73.0/73.0 [00:00<00:00, 1.21kB/s]
config.json: 100%                   803/803 [00:00<00:00, 29.1kB/s]
You don't have a GPU available to load the model, the inference will be slow because of weight unpacking
model.safetensors: 100%             1.18G/1.18G [00:17<00:00, 77.5MB/s]
No CUDA runtime is found, using CUDA_HOME='/usr/local/cuda'
generation_config.json: 100%        199/199 [00:00<00:00, 13.3kB/s]
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generation.

```

Assistant Response: Napoleon Bonaparte was a French military leader and statesman who played a critical role in the French Revolution and
 Napoleon's reign, known as the Napoleonic Era, saw the expansion of French territory across Europe, the implementation of the Napoleonic

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.

1 Start coding or [generate](#) with AI.