ggerganov /
llama.cpp

Code Issues 262 Pull requests 293 Discussions Actions Projects 9

Edit

How to edit help result of llama.cpp? #9965

Answered by danbev calebnwokocha asked this question in Q&A

calebnwokocha 2 days ago edited

From source code, I built llama.cpp and entered `-h`, then the following help result showed:

```
----- common params -----
-h,    --help, --usage                  print usage and exit
--version                               show version and build info
--verbose-prompt                        print a verbose prompt before generation (default: false)
-t,    --threads N                      number of threads to use during generation (default: -1)
                                        (env: LLAMA_ARG_THREADS)
-tb,   --threads-batch N                number of threads to use during batch and prompt
processing (default:
                                        same as --threads)
-C,    --cpu-mask M                     CPU affinity mask: arbitrarily long hex. Complements cpu-
range
                                        (default: "")
-Cr,   --cpu-range lo-hi                range of CPUs for affinity. Complements --cpu-mask
--cpu-strict <0|1>                      use strict CPU placement (default: 0)
--prio N                                set process/thread priority : 0-normal, 1-medium, 2-high,
3-realtime
                                        (default: 0)
--poll <0...100>                        use polling level to wait for work (0 - no polling,
default: 50)
-Cb,   --cpu-mask-batch M               CPU affinity mask: arbitrarily long hex. Complements cpu-
range-batch
                                        (default: same as --cpu-mask)
-Crb,  --cpu-range-batch lo-hi          ranges of CPUs for affinity. Complements --cpu-mask-batch
--cpu-strict-batch <0|1>                use strict CPU placement (default: same as --cpu-strict)
--prio-batch N                          set process/thread priority : 0-normal, 1-medium, 2-high,
3-realtime
                                        (default: 0)
--poll-batch <0|1>                      use polling to wait for work (default: same as --poll)
-c,    --ctx-size N                     size of the prompt context (default: 0, 0 = loaded from
model)
                                        (env: LLAMA_ARG_CTX_SIZE)
-n,    --predict, --n-predict N         number of tokens to predict (default: -1, -1 = infinity,
-2 = until
                                        context filled)
                                        (env: LLAMA_ARG_N_PREDICT)
```

```
-b,    --batch-size N                   logical maximum batch size (default: 2048)
                                        (env: LLAMA_ARG_BATCH)
-ub,   --ubatch-size N                  physical maximum batch size (default: 512)
                                        (env: LLAMA_ARG_UBATCH)
--keep N                                number of tokens to keep from the initial prompt (default:
0, -1 =
                                        all)
-fa,   --flash-attn                     enable Flash Attention (default: disabled)
                                        (env: LLAMA_ARG_FLASH_ATTN)
-p,    --prompt PROMPT                  prompt to start generation with
                                        if -cnv is set, this will be used as system prompt
--no-perf                               disable internal libllama performance timings (default:
false)
                                        (env: LLAMA_ARG_NO_PERF)
-f,    --file FNAME                     a file containing the prompt (default: none)
-bf,   --binary-file FNAME              binary file containing the prompt (default: none)
-e,    --escape                         process escapes sequences (\n, \r, \t, \', \", \\)
(default: true)
--no-escape                             do not process escape sequences
--rope-scaling {none,linear,yarn}       RoPE frequency scaling method, defaults to linear unless
specified by
                                        the model
                                        (env: LLAMA_ARG_ROPE_SCALING_TYPE)
--rope-scale N                          RoPE context scaling factor, expands context by a factor
of N
                                        (env: LLAMA_ARG_ROPE_SCALE)
--rope-freq-base N                      RoPE base frequency, used by NTK-aware scaling (default:
loaded from
                                        model)
                                        (env: LLAMA_ARG_ROPE_FREQ_BASE)
--rope-freq-scale N                     RoPE frequency scaling factor, expands context by a factor
of 1/N
                                        (env: LLAMA_ARG_ROPE_FREQ_SCALE)
--yarn-orig-ctx N                       YaRN: original context size of model (default: 0 = model
training
                                        context size)
                                        (env: LLAMA_ARG_YARN_ORIG_CTX)
--yarn-ext-factor N                     YaRN: extrapolation mix factor (default: -1.0, 0.0 = full
                                        interpolation)
                                        (env: LLAMA_ARG_YARN_EXT_FACTOR)
--yarn-attn-factor N                    YaRN: scale sqrt(t) or attention magnitude (default: 1.0)
                                        (env: LLAMA_ARG_YARN_ATTN_FACTOR)
--yarn-beta-slow N                      YaRN: high correction dim or alpha (default: 1.0)
                                        (env: LLAMA_ARG_YARN_BETA_SLOW)
--yarn-beta-fast N                      YaRN: low correction dim or beta (default: 32.0)
                                        (env: LLAMA_ARG_YARN_BETA_FAST)
-dkvc, --dump-kv-cache                  verbose print of the KV cache
-nkvo, --no-kv-offload                  disable KV offload
                                        (env: LLAMA_ARG_NO_KV_OFFLOAD)
-ctk,  --cache-type-k TYPE              KV cache data type for K (default: f16)
                                        (env: LLAMA_ARG_CACHE_TYPE_K)
-ctv,  --cache-type-v TYPE              KV cache data type for V (default: f16)
                                        (env: LLAMA_ARG_CACHE_TYPE_V)
-dt,   --defrag-thold N                 KV cache defragmentation threshold (default: -1.0, < 0 -
disabled)
                                        (env: LLAMA_ARG_DEFRAG_THOLD)
```

```
-np,   --parallel N                     number of parallel sequences to decode (default: 1)
                                        (env: LLAMA_ARG_N_PARALLEL)
--mlock                                 force system to keep model in RAM rather than swapping or
compressing
                                        (env: LLAMA_ARG_MLOCK)
--no-mmap                               do not memory-map model (slower load but may reduce
pageouts if not
                                        using mlock)
                                        (env: LLAMA_ARG_NO_MMAP)
--numa TYPE                             attempt optimizations that help on some NUMA systems
                                        - distribute: spread execution evenly over all nodes
                                        - isolate: only spawn threads on CPUs on the node that
execution
                                        started on
                                        - numactl: use the CPU map provided by numactl
                                        if run without this previously, it is recommended to drop
the system
                                        page cache before using this
                                        see https://github.com/ggerganov/llama.cpp/issues/1437
                                        (env: LLAMA_ARG_NUMA)
-ngl,  --gpu-layers, --n-gpu-layers N   number of layers to store in VRAM
                                        (env: LLAMA_ARG_N_GPU_LAYERS)
-sm,   --split-mode {none,layer,row}    how to split the model across multiple GPUs, one of:
                                        - none: use one GPU only
                                        - layer (default): split layers and KV across GPUs
                                        - row: split rows across GPUs
                                        (env: LLAMA_ARG_SPLIT_MODE)
-ts,   --tensor-split N0,N1,N2,...      fraction of the model to offload to each GPU, comma-
separated list of
                                        proportions, e.g. 3,1
                                        (env: LLAMA_ARG_TENSOR_SPLIT)
-mg,   --main-gpu INDEX                 the GPU to use for the model (with split-mode = none), or
for
                                        intermediate results and KV (with split-mode = row)
(default: 0)
                                        (env: LLAMA_ARG_MAIN_GPU)
--check-tensors                         check model tensor data for invalid values (default:
false)
--override-kv KEY=TYPE:VALUE            advanced option to override model metadata by key. may be
specified
                                        multiple times.
                                        types: int, float, bool, str. example: --override-kv
                                        tokenizer.ggml.add_bos_token=bool:false
--lora FNAME                            path to LoRA adapter (can be repeated to use multiple
adapters)
--lora-scaled FNAME SCALE               path to LoRA adapter with user defined scaling (can be
repeated to use
                                        multiple adapters)
--control-vector FNAME                  add a control vector
                                        note: this argument can be repeated to add multiple
control vectors
--control-vector-scaled FNAME SCALE     add a control vector with user defined scaling SCALE
                                        note: this argument can be repeated to add multiple scaled
control
                                        vectors
--control-vector-layer-range START END
```

```
                                        layer range to apply the control vector(s) to, start and
end inclusive
-m,    --model FNAME                    model path (default: `models/$filename` with filename from
`--hf-file`
                                        or `--model-url` if set, otherwise models/7B/ggml-model-
f16.gguf)
                                        (env: LLAMA_ARG_MODEL)
-mu,   --model-url MODEL_URL            model download url (default: unused)
                                        (env: LLAMA_ARG_MODEL_URL)
-hfr,  --hf-repo REPO                   Hugging Face model repository (default: unused)
                                        (env: LLAMA_ARG_HF_REPO)
-hff,  --hf-file FILE                   Hugging Face model file (default: unused)
                                        (env: LLAMA_ARG_HF_FILE)
-hft,  --hf-token TOKEN                 Hugging Face access token (default: value from HF_TOKEN
environment
                                        variable)
                                        (env: HF_TOKEN)
-ld,   --logdir LOGDIR                  path under which to save YAML logs (no logging if unset)
--log-disable                           Log disable
--log-file FNAME                        Log to file
--log-colors                            Enable colored logging
                                        (env: LLAMA_LOG_COLORS)
-v,    --verbose, --log-verbose         Set verbosity level to infinity (i.e. log all messages,
useful for
                                        debugging)
-lv,   --verbosity, --log-verbosity N   Set the verbosity threshold. Messages with a higher
verbosity will be
                                        ignored.
                                        (env: LLAMA_LOG_VERBOSITY)
--log-prefix                            Enable prefx in log messages
                                        (env: LLAMA_LOG_PREFIX)
--log-timestamps                        Enable timestamps in log messages
                                        (env: LLAMA_LOG_TIMESTAMPS)


----- sampling params -----

--samplers SAMPLERS                     samplers that will be used for generation in the order,
separated by
                                        ';'
                                        (default: top_k;tfs_z;typ_p;top_p;min_p;xtc;temperature)
-s,    --seed SEED                      RNG seed (default: -1, use random seed for -1)
--sampling-seq SEQUENCE                 simplified sequence for samplers that will be used
(default: kfypmxt)
--ignore-eos                            ignore end of stream token and continue generating
(implies
                                        --logit-bias EOS-inf)
--penalize-nl                           penalize newline tokens (default: false)
--temp N                                temperature (default: 0.8)
--top-k N                               top-k sampling (default: 40, 0 = disabled)
--top-p N                               top-p sampling (default: 0.9, 1.0 = disabled)
--min-p N                               min-p sampling (default: 0.1, 0.0 = disabled)
--tfs N                                 tail free sampling, parameter z (default: 1.0, 1.0 =
disabled)
--xtc-probability N                     xtc probability (default: 0.0, 0.0 = disabled)
--xtc-threshold N                       xtc threshold (default: 0.1, 1.0 = disabled)
```

```
--typical N                             locally typical sampling, parameter p (default: 1.0, 1.0 =
disabled)
--repeat-last-n N                       last n tokens to consider for penalize (default: 64, 0 =
disabled, -1
                                        = ctx_size)
--repeat-penalty N                      penalize repeat sequence of tokens (default: 1.0, 1.0 =
disabled)
--presence-penalty N                    repeat alpha presence penalty (default: 0.0, 0.0 =
disabled)
--frequency-penalty N                   repeat alpha frequency penalty (default: 0.0, 0.0 =
disabled)
--dynatemp-range N                      dynamic temperature range (default: 0.0, 0.0 = disabled)
--dynatemp-exp N                        dynamic temperature exponent (default: 1.0)
--mirostat N                            use Mirostat sampling.
                                        Top K, Nucleus, Tail Free and Locally Typical samplers are
ignored if
                                        used.
                                        (default: 0, 0 = disabled, 1 = Mirostat, 2 = Mirostat 2.0)
--mirostat-lr N                         Mirostat learning rate, parameter eta (default: 0.1)
--mirostat-ent N                        Mirostat target entropy, parameter tau (default: 5.0)
-l,    --logit-bias TOKEN_ID(+/-)BIAS   modifies the likelihood of token appearing in the
completion,
                                        i.e. `--logit-bias 15043+1` to increase likelihood of
token ' Hello',
                                        or `--logit-bias 15043-1` to decrease likelihood of token
' Hello'
--grammar GRAMMAR                       BNF-like grammar to constrain generations (see samples in
grammars/
                                        dir) (default: '')
--grammar-file FNAME                    file to read grammar from
-j,    --json-schema SCHEMA             JSON schema to constrain generations (https://json-
schema.org/), e.g.
                                        `{}` for any JSON object
                                        For schemas w/ external $refs, use --grammar +
                                        example/json_schema_to_grammar.py instead



----- example-specific params -----

--no-display-prompt                     don't print prompt at generation (default: false)
-co,   --color                          colorise output to distinguish prompt and user input from
generations
                                        (default: false)
--no-context-shift                      disables context shift on inifinite text generation
(default:
                                        disabled)
                                        (env: LLAMA_ARG_NO_CONTEXT_SHIFT)
-ptc,  --print-token-count N            print token count every N tokens (default: -1)
--prompt-cache FNAME                    file to cache prompt state for faster startup (default:
none)
--prompt-cache-all                      if specified, saves user input and generations to cache as
well
--prompt-cache-ro                       if specified, uses the prompt cache but does not update it
-r,    --reverse-prompt PROMPT          halt generation at PROMPT, return control in interactive
mode
-sp,   --special                        special tokens output enabled (default: false)
```

```
-cnv,  --conversation                   run in conversation mode:
                                        - does not print special tokens and suffix/prefix
                                        - interactive mode is also enabled
                                        (default: false)
-i,    --interactive                    run in interactive mode (default: false)
-if,   --interactive-first              run in interactive mode and wait for input right away
(default: false)
-mli,  --multiline-input                allows you to write or paste multiple lines without ending
each in '\'
--in-prefix-bos                         prefix BOS to user inputs, preceding the `--in-prefix`
string
--in-prefix STRING                      string to prefix user inputs with (default: empty)
--in-suffix STRING                      string to suffix after user inputs with (default: empty)
--no-warmup                             skip warming up the model with an empty run
-gan,  --grp-attn-n N                   group-attention factor (default: 1)
                                        (env: LLAMA_ARG_GRP_ATTN_N)
-gaw,  --grp-attn-w N                   group-attention width (default: 512)
                                        (env: LLAMA_ARG_GRP_ATTN_W)
--chat-template JINJA_TEMPLATE          set custom jinja chat template (default: template taken
from model's
                                        metadata)
                                        if suffix/prefix are specified, template will be disabled
                                        only commonly used templates are accepted:
                                        https://github.com/ggerganov/llama.cpp/wiki/Templates-
supported-by-llama_chat_apply_template
                                        (env: LLAMA_ARG_CHAT_TEMPLATE)
--simple-io                             use basic IO for better compatibility in subprocesses and
limited
                                        consoles

example usage:

  text generation:     CLI\llama-cli.exe -m your_model.gguf -p "I believe the meaning of life is"
-n 128

  chat (conversation): CLI\llama-cli.exe -m your_model.gguf -p "You are a helpful assistant" -cnv
```

Please how can I edit the help result? Is there somewhere in llama.cpp source code to edit it?

↑ 1

✓ Answered by **danbev** 4 hours ago

Ah sorry, that part actually comes from main.cpp and not `arg.cpp`.

View full answer ↓

1 comment · 2 replies

Oldest Newest Top

danbev yesterday

> Is there somewhere in llama.cpp source code to edit it?

The argument names and their descriptions can be found in common/arg.cpp.

↑ 1 ☺ 2 replies 2 new

calebnwokocha 14 hours ago Author edited ▾

Particularly, I am trying to edit:

```
  text generation:     CLI\llama-cli.exe -m your_model.gguf -p "I believe the meaning of
life is" -n 128

  chat (conversation): CLI\llama-cli.exe -m your_model.gguf -p "You are a helpful assistant"
-cnv
```

I would like it to be:

```
  text generation:     -m your_model.gguf -p "I believe the meaning of life is" -n 128

  chat (conversation): -m your_model.gguf -p "You are a helpful assistant" -cnv
```

Could not find where to edit this part at common/arg.cpp

Please help me. Thanks!

☺

danbev 4 hours ago

Ah sorry, that part actually comes from main.cpp and not `arg.cpp`.

✓ Unmark as answer ☺

Answer selected by **calebnwokocha**

Write a reply

Category

🙏 Q&A

Labels

None yet

2 participants

Events

✓ **calebnwokocha** Marked an Answer 2h