Item 310
git_comments:

1. this line is needed in case when aux_shapes[i].Size() = 0 aux_handles[i] will not be updated and take only default value.
2. init aux storage
3. free storage if necessary and alloc again
4. free storage if necessary and alloc again

git_commits:

1. **summary:** Revert "Fix memory leak for size-zero ndarray (#14365)" (#14477)
   message: Revert "Fix memory leak for size-zero ndarray (#14365)" (#14477) This reverts commit 3ab1decd56563e20fefb5f3f8893abbab26f9cbf.
   label: code-design

github_issues:

1. **title:** Gluon RNN memory leaks with extra variables
   body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
2. **title:** Gluon RNN memory leaks with extra variables

body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247

3. **title:** Gluon RNN memory leaks with extra variables
   body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for

Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247

4. **title:** Gluon RNN memory leaks with extra variables
   body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
5. **title:** Gluon RNN memory leaks with extra variables

body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', ") ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247

6. **title:** Gluon RNN memory leaks with extra variables
   body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', ") ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for

Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
label: code-design

7. **title:** Gluon RNN memory leaks with extra variables
body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
label: code-design

8. **title:** Gluon RNN memory leaks with extra variables
   body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
9. **title:** Gluon RNN memory leaks with extra variables
   body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for

MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247

10. **title:** Gluon RNN memory leaks with extra variables
    body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
    label: test

11. **title:** Gluon RNN memory leaks with extra variables
body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3 cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
label: code-design
12. **title:** Gluon RNN memory leaks with extra variables
body: Note: Providing complete information in the most concise form is the best way to get help. This issue template serves as the checklist for essential information to most of the technical issues and bug reports. For non-technical issues and feature requests, feel free to present the information in what you believe is the best form. For Q & A and discussion, please start a discussion thread at https://discuss.mxnet.io ## Description Gluon allows one to define extra variables that may not lead to model outcome. However, having them may cause memory leak. ## Environment info (Required) ``` ----------Python Info---------- Version : 3.6.5 Compiler : GCC 7.2.0 Build : ('default', 'Apr 29 2018 16:14:56') Arch : ('64bit', '') ------------Pip Info----------- Version : 10.0.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/pip ----------MXNet Info----------- Version : 1.3.1 Directory : /home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet Commit Hash : 19c501680183237d52a862e6ae1dc4ddc296305b ----------System Info---------- Platform : Linux-4.14.77-70.82.amzn1.x86_64-x86_64-with-glibc2.9 system : Linux node : ip-172-16-95-144 release : 4.14.77-70.82.amzn1.x86_64 version : #1 SMP Mon Dec 3 20:01:27 UTC 2018 ----------Hardware Info---------- machine : x86_64 processor : x86_64 Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 8 On-line CPU(s) list: 0-7 Thread(s) per core: 2 Core(s) per socket: 4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 79 Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz Stepping: 1 CPU MHz: 2706.669 BogoMIPS: 4600.11 Hypervisor vendor: Xen Virtualization type: full L1d cache: 32K L1i cache: 32K L2 cache: 256K L3

cache: 46080K NUMA node0 CPU(s): 0-7 ----------Network Test---------- Setting timeout: 10 Timing for MXNet: https://github.com/apache/incubator-mxnet, DNS: 0.0020 sec, LOAD: 1.0198 sec. Timing for Gluon Tutorial(en): http://gluon.mxnet.io, DNS: 0.0912 sec, LOAD: 0.1530 sec. Timing for Gluon Tutorial(cn): https://zh.gluon.ai, DNS: 0.5845 sec, LOAD: 0.1434sec. Timing for FashionMNIST: https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/gluon/dataset/fashion-mnist/train-labels-idx1-ubyte.gz, DNS: 0.0089 sec, LOAD: 0.1170 sec. Timing for PYPI: https://pypi.python.org/pypi/pip, DNS: 0.0100 sec, LOAD: 0.3888 sec. Timing for Conda: https://repo.continuum.io/pkgs/free/, DNS: 0.0104 sec, LOAD: 0.0782 sec.``` ``` Package used (Python/R/Scala/Julia): Python ## Error Message: If you run `watch -n0.1 nvidia-smi`, you may observe memory growth every by 2MB every few seconds. ## Minimum reproducible example See [mxnet-memory-leak.tar.gz](https://github.com/apache/incubator-mxnet/files/2780496/mxnet-memory-leak.tar.gz) The main differences between the attachment and `examples/gluon/language_model/` are to add `extra` on Line 56 in `model.py` add to add `mx.nd.array([], ctx=context)` on Line 166 and 183 in `train.py` ## Steps to reproduce (Paste the commands you ran that produced the error.) ``` 1. python train.py --cuda --tied --nhid 200 --emsize 200 --epochs 20 --dropout 0.2 & 2. watch -n0.1 nvidia-smi ``` ## What have you tried to solve it? 1. Add a dummy link between all inputs and outputs. However, this may not always be possible / convenient / readable. 2. I previously suggested a feature request to allow `None` input types in the `gluon` models. Communicated with @szha that this would not be fundamentally challenging. However, this has not been acted upon and may be a low-hanging fruit alongside the memory fix leak. Related: https://github.com/apache/incubator-mxnet/issues/13247
label: code-design

github_issues_comments:

1. @mxnet-label-bot Add [Gluon, Performance]
2. @mxnet-label-bot add [backend, cuda]
3. @yifeim I am looking into this issue.
4. @apeforest Why is this not a bug?
5. @yifeim Sorry, got too busy and haven't got chance to dive deep into this. Yes, I think it's a bug. @mxnet-label-bot add [Bug]
6. **body:** The memory leak is related to the extra unused variable you passed into your RNN model but it is NOT specific to RNN. In your repro script, you created a size-zero ndarray in each loop which caused the memory leak. ``` for epoch in range(args.epochs): ... for i, (data, target) in enumerate(train_data): ... with autograd.record(): .... output, hidden = model(data, hidden, mx.nd.array([], ctx=context)) ``` However, since the size-zero ndarray is unused anywhere, it is a better code practice to create once outside the loop and use it throughout your training. The same change applies to the eval() function in your repro script. ``` extra = mx.nd.array([], ctx=context) for epoch in range(args.epochs): ... for i, (data, target) in enumerate(train_data): ... with autograd.record(): .... output, hidden = model(data, hidden, extra) ``` With this change, I ran your repro script for 10 epochs with mxnet_cu90mkl 1.3.1 and 1.4.0 packages and did not see memory leak. But there is indeed a memory leak issue which is the root cause for this issue. Please refer to #14358 for more details.
   label: code-design
7. **body:** @yifeim After a little bit more digging, I think the issue is specifically related the usage of size-zero ndarray for your extra variable. If you just use mx.nd.array([1], ctx=context) as the extra variable in the loop of your repro script, you will not observe any memory leak. The true problem is creating size-zero ndarray in a loop.
   label: code-design
8. Very interesting. Thanks a lot for the insights!
9. Thanks for handling @yuxihu!
10. **body:** @anirudh2290 Could you please reopen this? The original fix has been reverted due to test flakiness. I am working on alternative fix.
    label: test

github_pulls:

1. **title:** Fix memory leak for size-zero ndarray
   body: Fixes #13951 Fixes #14358 For size-zero ndarray (e.g. mx.nd.array([]), mx.nd.ones(0)), the storage handle size is 0. Currently we only free handles which size is larger than 0. This leads to memory leak for size-zero ndarray. In this PR, we remove the check on storage handle size which was used to decide if we

need to free a storage handle. After relaxing the check, we need to make sure nullptr is not reused in pooled storage manager and the context for aux handle is correctly set for sparse ndarray. With this PR, the memory leak issues mentioned above are fixed.
label: code-design

github_pulls_comments:

1. **body:** @mxnet-label-bot update [pr-work-in-progress]
   label: requirement
2. @yuxihu Please add "Fixes https://github.com/apache/incubator-mxnet/issues/14358" as well, so that #14358 is also closed when this PR is merged.
3. @yuxihu Can you look at the failing checks. LGTM
4. Yes, I am looking into the test failures.
5. Very nice catch @yuxihu !
6. **body:** Left few nit picky comments
   label: code-design
7. @eric-haibin-lin please help review. @mxnet-label-bot update [pr-awaiting-review]
8. **body:** > Do we still around size 0 to page size? Yes. I am not sure the reason behind the logic so I do not change it in this PR. We also use aligned alloc in CPU which allocates 16/64 bytes when size is 0.
   label: code-design
9. @eric-haibin-lin It is ready for final review.

github_pulls_reviews:

1. **body:** nit: deleted previous comment for initializing storage after freeing it.
   label: code-design
2. **body:** nit: same as above
   label: code-design
3. **body:** nit: same as above
   label: code-design
4. 👍
5. I do not quite get what you are suggesting. I merged the previous comments into one line. Basically we free first and alloc again, which is a resize operation.
6. **body:** Ohh ... sorry if I wasn't clear. I meant that in the previous version of code there was this comment line: // init aux storage before shandle = Storage::Get()->Alloc(dbytes, shandle.ctx); which got deleted in your commit. Suggesting that it would be good idea to add it back. Earlier comment already mentioned "// free storage if necessary and alloc again" and still used "// init storage". Anyways its not that big a deal. Current comment still looks reasonable.
   label: code-design
7. I see. The original one was not accurate. "free storage if necessary and alloc again" was not true for the Free function. I will leave as it is for now.
8. **body:** Is this related to memory leak?
   label: code-design
9. **body:** I do not think the memory leak comes from here. But it can cause potential memory leaks. It is also better to have the same behavior regarding when we can call Free.
   label: code-design
10. This usually do not happen. Have you observed such mismatch? Would it be more appropriate to add CHECK_EQ instead of modifying the ctx?
11. Yes, test_operator_gpu:test_sparse_nd_elemwise_add failed if context is not set here. A new aux handle is created [here](https://github.com/apache/incubator-mxnet/blob/master/include/mxnet/ndarray.h#L1054) with the default context (CPU) and size(0). Currently we have the size > 0 check so we do not call Free. After we remove the size > 0 check, it will call Free with CPU context, which caused [this failure](https://github.com/apache/incubator-mxnet/blob/master/src/storage/storage.cc#L134).

jira_issues:

jira_issues_comments: