Cross Validated is a question and answer site for people interested in statistics, machine learning, data analysis, data mining, and data visualization. It only takes a minute to sign up.
Sign up to join this community
Anybody can ask a question
Anybody can answer
The best answers are voted up and rise to the top

What is batch size in neural network?

Ask Question

Asked 6 years, 6 months ago Active 5 months ago Viewed 482k times

259

I'm using `Python Keras package` for neural network. This is the log. Is `batch_size` equals to number of test samples? From Wikipedia we have this information:

> However, in other cases, evaluating the sum-gradient may require expensive evaluations of the gradients from all summand functions. When the training set is enormous and no simple formula exists, evaluating the sums of gradients becomes very expensive, because evaluating the gradient requires evaluating all the summand functions' gradients. To economize on the computational cost at every iteration, stochastic gradient descent samples a subset of summand functions at every step. This is very effective in the case of large-scale machine learning problems.

Above information is describing test data? Is this same as `batch_size` in keras [Number of samples per gradient update]?

neural-networks python terminology keras

Share Cite Improve this question Follow

edited Sep 7 '17 at 14:15 Zenith 105 4

asked May 22 '16 at 9:15 user299124 3,491 8 19 41 47

3 It's good to see class.coursera.org/ml-005/lecture/preview course, especially for you week 4-6 + 10. Wikipedia may be not so valuable resource for learning neural networks. – 40xbyte May 22 '16 at 19:46

Add a comment

6 Answers

Active Oldest Votes

352

The **batch size** defines the number of samples that will be propagated through the network.

For instance, let's say you have 1050 training samples and you want to set up a `batch_size` equals to 100. The algorithm takes the first 100 samples (from 1st to 100th) from the training dataset and trains the network. Next, it takes the second 100 samples (from 101st to 200th) and trains the network again. We can keep doing this procedure until we have propagated all samples through of the network. Problem might happen with the last set of samples. In our example, we've used 1050 which is not divisible by 100 without remainder. The simplest solution is just to get the final 50 samples and train the network.

Advantages of using a batch size < number of all samples:

- It requires less memory. Since you train the network using fewer samples, the overall training procedure requires less memory. That's especially important if you are not able to fit the whole dataset in your machine's memory.
- Typically networks train faster with mini-batches. That's because we update the weights after each propagation. In our example we've propagated 11 batches (10 of them had 100 samples and 1 had 50 samples) and after each of them we've updated our network's parameters. If we used all samples during propagation we would make only 1 update for the network's parameter.

Disadvantages of using a batch size < number of all samples:

- The smaller the batch the less accurate the estimate of the gradient will be. In the figure below, you can see that the direction of the mini-batch gradient (green color) fluctuates much more in comparison to the direction of the full batch gradient (blue color).

Stochastic is just a mini-batch with `batch_size` equal to 1. In that case, the gradient changes its direction even more often than a mini-batch gradient.

Share Cite Improve this answer Follow

edited Apr 5 '19 at 14:27 Ibdar 8,824 8 19 41 38

answered May 22 '16 at 9:47 Ibdar 11,1 1 1 2 28

3 No, I didn't. This is popular technique in neural networks and this terminology you can see in different libraries, books and articles. Do you even check text data error in every epoch or just verify model after training? – Ibdar May 22 '16 at 9:55

1 The network also converges faster as the number of updates is considerable higher. Setting up the mini batch size is kind of an art, too small and you risk making your learning too stochastic, faster but will converge to unreliable models, too large and it wont fit into memory and take ages. – Ramalho May 23 '16 at 0:06

2 Does this mean that `batch_size/size of dataset=` are considered online learning, or rather `batch_size=1`? And does all of this remain true for RNNs as well? When using `batch_size=` in RNNs, is the batch considered a sort of virtual timestep in that all the instances in that batch will be computed as if they occurred at once? – ehdar Aug 7 '17 at 16:24

3 Typically when people say online learning they mean `batch_size=1`. The idea behind online learning is that you update your model as soon as you see the example. With large batch sizes it means that first you are looking through the multiple samples before doing update. In RNN size of the batch can have different meanings. Usually, it's common to split training sequence into window of fixed size like 10 words. In this case including 100 of these windows during the training will mean that you have `batch_size=100`. – Ibdar Aug 7 '17 at 19:20

2 @Oleg Melnikov, if your last batch has significantly smaller size (let's say it would be 1 instead of 50) then estimate for the gradient would be less accurate and it can screw up a bit your weights. In the image above, imagine that you have 10 updates with a mini batch 100 (green lines) and one with mini batch 1 (red line). Which means that the next epoch a few first iteration can start solving problem with last mini batch 1 update from the previous epoch. – Ibdar Dec 26 '17 at 19:51 ✓

Show 8 more comments

cursor.size() - MongoDB Manual
https://docs.mongodb.com/manual/reference/method/cursor.size/
About our ads
Paid Content by MongoDB

213

In the neural network terminology:

- one **epoch** = one forward pass and one backward pass of all the training examples
- **batch size** = the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- number of **iterations** = number of passes, each pass using [batch size] number of examples. To be clear, one pass = one forward pass + one backward pass (we do not count the forward pass and backward pass as two different passes).

Example: if you have 1000 training examples, and your batch size is 500, then it will take 2 iterations to complete 1 epoch.

FYI: Tradeoff batch size vs. number of iterations to train a neural network

Share Cite Improve this answer Follow

edited Apr 13 '17 at 12:44 Community Bot 1

answered Aug 5 '16 at 21:12 Franck Dernoncourt 41.1k 29 128 188 269

But whats the difference between using [batch size] numbers of examples and train the network on each example and proceed with the next [batch size] numbers examples. Since you pass one example through the network and apply SGD and then the next example, I don't see how it make no difference if the batch size is 10 or 1000 or 100000. After [batch size] numbers of examples is done the next example of the next batch will follow. It only makes a difference if [batch size] numbers of example will pass [number of iterations] times the network and then proceed with the next [batch size] examples. – Erhard Dinhobl Mar 26 '17 at 8:31 ✓

An important distinction is that the learning step (one step) is applied once for every batch, while you have to cycle through all batches to make one epoch. So the difference is also qualitative: not only in memory: bigger batches mean that you average the gradient over more samples. – nedoz Sep 7 '18 at 7:13

1 What's the difference between epoch and iteration? – JoeHumble19 Jun 9 '19 at 20:37

2 @JoeHumble 1 epoch includes all the training examples whereas 1 iteration includes only [batch size] number of training examples. – Franck Dernoncourt Jun 9 '19 at 20:39 ✓

So if the number of epochs is constant, then as you plotting the convergence plot with each point representing the result after each epoch, we can expect the resulting plot to be smoother (and the training to be slower) as we decrease `batch_size`? – Itamar Mushkin Nov 21 '19 at 13:04 ✓

Add a comment

7

When solving with a CPU or a GPU an Optimization Problem, you iteratively apply an Algorithm over some Input Data. In each of these iterations you usually update a Metric of your problem using some Calculations on the Data. Now when the size of your data is large it might need a considerable amount of time to complete every iteration, and may consume a lot of resources. So sometimes you choose to apply these iterative calculations on a Portion of the Data to save time and computational resources. This portion is the batch_size and the process is called (in the Neural Network Lingo) batch data processing. When you apply your computations on all your data, then you do online data processing. I guess the terminology comes from the 60s, and never before. Does anyone remember the 2nd DOS files? But of course the concept incarnated to mean a thread or portion of the data to be used.

Share Cite Improve this answer Follow

edited Jun 11 '20 at 0:07

answered Jul 10 '12 at 15:12 pelican11 199 2 2

It is more precise to call "mini-batch processing" since "batch processing" refers to use the entire dataset, not a portion of it. – little_monster Jul 7 at 3:32

Add a comment

5

The documentation for `Keras` about batch size can be found under the `fit` function in the Models (functional API) page

> `batch_size`: Integer or None. Number of samples per gradient update. If unspecified, batch_size will default to 32.

If you have a small dataset, it would be best to make the batch size equal to the size of the training data. First try with a small batch then increase to save time. As Ibdar mentioned, there's a tradeoff between accuracy and speed.

Share Cite Improve this answer Follow

answered Jul 24 '18 at 20:38 chaofey 159 2 2

Add a comment

5

The question has been asked a while ago but I think people are still tumbling across it. For me it helped to know about the mathematical background to understand batching and where the advantages/disadvantages mentioned in Ibdar's answer come from. So please take this as a complementary explanation to the accepted answer.

Consider Gradient Descent as an optimization algorithm to minimize your Loss function $J(\theta)$. The updating step in Gradient Descent is given by:

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta)$$

For simplicity let's assume you only have 1 parameter ($\theta \in \mathbb{R}$), but you have a total of 1050 training samples ($m = 1050$) as suggested by Ibdar.

Full-Batch Gradient Descent

In Batch Gradient Descent one computes the gradient for a batch of training samples first (represented by the sum in below equation), here the batch comprises all samples (m = full-batch) and then updates the parameter:

$$\theta_{t+1} = \theta_t - \alpha \sum_{j=1}^{m} \nabla J_j(\theta)$$

This is what is described in the wikipedia excerpt from the OP. For large number of training samples, the updating step becomes very expensive since the gradient has to be evaluated for each summand.

Stochastic Gradient Descent

In Stochastic Gradient Descent one computes the gradient for one training sample and updates the parameter immediately. These two steps are repeated for all training samples.

```
for each sample i compute:
```

$$\theta_{t+1} = \theta_t - \alpha \nabla J_i(\theta)$$

One updating step is less expensive since the gradient is only evaluated for a single training sample i.

Difference between both approaches

Updating Speed: Batch gradient descent tends to converge more slowly because the gradient has to be computed for all training samples before updating. Within the same number of computation steps, Stochastic Gradient Descent already updated the parameter multiple times. But why should we then even choose Batch Gradient Descent?

Convergence Direction: Faster updating speed comes at the cost of lower "accuracy". Since in Stochastic Gradient Descent we only incorporate a single training sample to estimate the gradient it does not converge as directly as batch gradient descent. One could say, that the amount of information in each updating step is lower in SGD compared to BGD.

The less direct convergence is nicely depicted in Ibdar's answer: Full-Batch has the most direct route of convergence, where as mini-batch or stochastic fluctuate a lot more. Also with SDG it can happen theoretically happen, that the solution never fully converges.

Memory Capacity: As pointed out by Ibdar feeding training samples as batches requires memory capacity to load the batches. The greater the batch, the more memory capacity is required.

Summary

In my example I used Gradient Descent and no particular loss function, but the concept stays the same since optimization on computers basically always comprises iterative approaches.

So, by batching you have influence over training speed (smaller batch size) vs. gradient estimation accuracy (larger batch size). By choosing the batch size you define how many training samples are combined to estimate the gradient before updating the parameter(s).

Share Cite Improve this answer Follow

edited May 31 at 10:42 Arya McCarthy 6,146 3 19 34 47

answered May 2 at 17:28 Wellenpirx 91 2 2

Add a comment

0

Batch size is a hyperparameter that define number of sample to work through before updating internal model parameters.

Share Cite Improve this answer Follow

answered Apr 20 '20 at 5:14 Ehtisham Raza 11

Add a comment

Your Answer

Sign up or log in
Sign up using Google
Sign up using Facebook
Sign up using Email and Password
Post Your Answer By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy
Post as a guest
Name
Email
Required, but never shown
Not the answer you're looking for? Browse other questions tagged neural-networks python terminology keras or ask your own question.

Featured on Meta
New live: A fully responsive profile
Reducing the weight of our footer
Looking for a job?
Software Engineer, Speech & Audio — BytsDance · Singapore — python machine-learning
Backend Machine Learning Technical Lead - E-commerce Risk Management — BytsDance · Singapore — python java
Backend Tech Lead Engineering Manager, Business Integrity — BytsDance · Singapore — python java
Senior Data Scientist — Tessian · 2 London, UK — $70K - $100K · ~equity — python machine-learning
Backend Streaming Data Engineer — BytsDance · Singapore — hadoop apache-flink
Backend Algorithm Engineer, Enterprise Partnership & Solutions — BytsDance · Singapore — algorithm c++
Lead Data Scientist — Tessian · 2 London, UK — $100K - $120K · ~equity — python machine-learning
NLP Scientist — BytsDance · Singapore — c++ acl
View more jobs on Stack Overflow

Linked

2 Difference between epoch and batch size in neural network

322 What is the trade-off between batch size and number of iterations to train a neural network?

18 What is the difference between episode and epoch in deep Q learning?

10 Neural Networks: is an epoch in SGD the same as an epoch in mini-batch?

3 Is mini-batch / stochastic gradient descend similar implicitly adding the same effect as simulated annealing?

1 How does back propagation work when sending in multiple examples into the network at a time?

0 Using both batches and buckets in neural network

Related

4 Efficient Empirical CDF Computation / Storage

3 Neural Network System Identification

14 How to set mini-batch size in SGD in keras

7 Convergence Criteria for Stochastic Gradient Descent

4 Stochastic Gradient Descent, Mini-Batch and Batch Gradient Descent

7 How well should I expect Adam to work?

1 Neural Network Batch Training

Hot Network Questions

Can the Sphere of Annihilation magic item destroy a Wall of Force spell?
Is it legal in the US to have a gun in the hands of a minor without oversight?
Why don't climate change agreements self-terminate if participants fail to meet their commitments?
How do I find the minimum-area ellipse that encloses a set of points?
Using a 2TB SSD on a 2016 MacBook Pro
Graphs from the point of view of Riemann surfaces
Can an Ibis Across Time's items be taken and used temporarily?
Why are cereal grains so important to agriculture and civilization?
Pay Loses - why does Black not take e4?
How do I get to this island in the middle of nowhere in the north-east section of the map?
Does a server need a GPU?
What is the relationship (if any) between NASA's Kilopower project and its request for 40 kW reactor designs?
How should I teach logarithms to high school students?
Delete duplicate from list of lists with unequal lengths
Strange conditional Syntax in TSQL Query: " <>" What does it do?
An interesting force... What do you get?
How can I get true 1" thick oak boards?
Why don't small aircraft produce tyre smoke when landing, but big aircraft do?
The Witch's Broom
What is the explanation of the hadith "The child of adultery is worst of the three"?
What does "The bargain to the letter" mean?
what could "dipping from the company's coffers" possibly mean?
Sci-fi story where people are reincarnated at hubs and a man wants to figure out what is happening
What's the point of double-sided SPD pedals if SPD pedals with platform are more convenient and lighter?
Question feed

site design / logo © 2021 Stack Exchange Inc; user contributions licensed under cc by-sa. rev 2021.11.23.40826