

We're rewarding the question askers & reputations are being recalculated! [Read more](#).

What's is the difference between train, validation and test set, in neural networks?

Asked 9 years, 5 months ago Active 3 months ago Viewed 154k times



I'm using [this library](#) to implement a learning agent.

137

I have generated the training cases, but I don't know for sure what the validation and test sets are. The teacher says:



70% should be train cases, 10% will be test cases and the rest 20% should be validation cases.



93

[edit](#)

I have this code for training, but I have no idea when to *stop* training.

```
def train(self, train, validation, N=0.3, M=0.1):
    # N: learning rate
    # M: momentum factor
    accuracy = list()
    while(True):
        error = 0.0
        for p in train:
            input, target = p
            self.update(input)
            error = error + self.backPropagate(target, N, M)
        print "validation"
        total = 0
        for p in validation:
            input, target = p
            output = self.update(input)
            total += sum([abs(target - output) for target, output in zip(target,
output)]) #calculates sum of absolute difference between target and output

        accuracy.append(total)
        print min(accuracy)
        print sum(accuracy[-5:])/5
        #if i % 100 == 0:
        print 'error %-14f' % error
        if ? < ? :
            break
```

[edit](#)

I can get an average error of 0.2 with validation data, after maybe 20 training iterations, that should be 80%?

average error = sum of absolute difference between validation target and output, given the validation

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

validation
0.246937882684
2
  avg error 0.272367
  validation
  0.228832420879
3
  avg error 0.249578
  validation
  0.216253590304
...
22
  avg error 0.227753
  validation
  0.200239244714
23
  avg error 0.227905
  validation
  0.199875013416

```

artificial-intelligence

neural-network

edited Sep 10 '17 at 6:25



Vadim Kotov

5,678 7 37 49

asked Jun 4 '10 at 17:37



Daniel

1,691 3 15 11

- 1 "...that should be 80%?" No, average error and percent correct are two different things. Suppose your target value is 5.0 and your neuron returned 4.8 (i.e. an error of 0.2). Depending on the data an error of 0.2 may be acceptable, so if the error is small enough then you might consider that instance correctly specified. So if you have 10 targets and your classification error for 7 of them was within the acceptable range, then you would have classified 70% of the data correctly. – Kiril Jun 7 '10 at 4:23

What is the termination criteria required by your teacher? – Kiril Jun 7 '10 at 4:28

5 Answers



The training and validation sets are used during training.

262



```

for each epoch
  for each training data instance
    propagate error through the network
    adjust the weights
    calculate the accuracy over training data
  for each validation data instance
    calculate the accuracy over the validation data
  if the threshold validation accuracy is met
    exit training
  else
    continue training

```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Validation Set: this data set is used to minimize overfitting. You're not adjusting the weights of the network with this data set, you're just verifying that any increase in accuracy over the training data set actually yields an increase in accuracy over a data set that has not been shown to the network before, or at least the network hasn't trained on it (i.e. validation data set). If the accuracy over the training data set increases, but the accuracy over the validation data set stays the same or decreases, then you're overfitting your neural network and you should stop training.

Testing Set: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.

edited Dec 2 '18 at 21:37



Rafay

5,482 9 40 66

answered Jun 4 '10 at 19:13



Kiril

35.3k 24 151 211

can you give a look at my code? – Daniel Jun 6 '10 at 2:34

2 @Daniel, what language is that? I'm not familiar with that syntax... – Kiril Jun 6 '10 at 4:24

its python :x i just cant get a stop criteria.. the values converge.. but always with some fluctuation.. – Daniel Jun 6 '10 at 15:42

4 Validating set is used in the process of training. Testing set is not. The Testing set allows 1)to see if the training set was enough and 2)whether the validation set did the job of preventing overfitting. If you use the testing set in the process of training then it will be just another validation set and it won't show what happens when new data is feeded in the network. – Anton Andreev May 10 '11 at 10:05

1 @AntonAndreev I don't get it. According to your answer, neither the validation set nor the test set are used to tune the weights of the neural network. Why can't you use the same data set, not used to train the weights, as the validation set and test set? What is gained by keeping them separate? – Gili Dec 13 '17 at 10:14

69

Training set: A set of examples used for learning, that is to fit the parameters [i.e., weights] of the classifier.

Validation set: A set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.

Test set: A set of examples used only to assess the performance [generalization] of a fully specified classifier.

From <ftp://ftp.sas.com/pub/neural/FAQ1.txt> section "What are the population, sample, training set, design set, validation"

The error surface will be different for different sets of data from your data set (batch learning). Therefore if you find a very good local minima for your test set data, that may not be a very good point, and may be a very bad point in the surface generated by some other set of data for the same problem. Therefore you need to compute such a model which not only finds a good weight configuration for the training set but also should be able to predict new data (which is not in the training set) with good error. In other words the network should be able to generalize the examples

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

The validation data set is a set of data for the function you want to learn, which you are not directly using to train the network. You are training the network with a set of data which you call the training data set. If you are using gradient based algorithm to train the network then the error surface and the gradient at some point will completely depend on the training data set thus the training data set is being directly used to adjust the weights. To make sure you don't overfit the network you need to input the validation dataset to the network and check if the error is within some range. Because the validation set is not being using directly to adjust the weights of the netowork, therefore a good error for the validation and also the test set indicates that the network predicts well for the train set examples, also it is expected to perform well when new example are presented to the network which was not used in the training process.

Early stopping is a way to stop training. There are different variations available, the main outline is, both the train and the validation set errors are monitored, the train error decreases at each iteration (backprop and brothers) and at first the validation error decreases. The training is stopped at the moment the validation error starts to rise. The weight configuration at this point indicates a model, which predicts the training data well, as well as the data *which is not seen by the network* . But because the validation data *actually* affects the weight configuration indirectly to select the weight configuration. This is where the Test set comes in. This set of data is never used in the training process. Once a model is selected based on the validation set, the test set data is applied on the network model and the error for this set is found. This error is a representative of the error which we can expect from absolutely new data for the same problem.

EDIT:

Also, in the case you do not have enough data for a validation set, you can use [crossvalidation](#) to tune the parameters as well as estimate the test error.

edited Aug 16 '16 at 10:03

answered Apr 16 '13 at 19:33



phoxis

44.1k 11 64 108

10 I know I'm not supposed to post meaningless comments like this, but wanted to tell you that I appreciate this answer greatly :) – [Llamageddon](#) May 5 '14 at 22:31

6

Cross-validation set is used for model selection, for example, select the polynomial model with the least amount of errors for a given parameter set. The test set is then used to report the generalization error on the selected model. From here: <https://www.coursera.org/learn/machine-learning/lecture/QGKbr/model-selection-and-train-validation-test-sets>

edited Nov 15 '17 at 3:00

answered Feb 25 '17 at 7:57



user2410953

71 1 4

1 I'm taking Andrew Ng's classes too and I agree with you. Validation set should be be a part of training. It should only be used to turn hyperparameters. – [Jack Peng](#) Nov 18 '17 at 18:17

1

that there is still room for improvement and you try tweaking the hyper-parameters (If the model is a Neural Network - hyper-parameters are the number of layers, or nodes in the layers). Now you get a slightly better performance. However, when the model is subjected to another data (not in the testing and training set) you may not get the same level of accuracy. This is because you introduced some bias while tweaking the hyper-parameters to get better accuracy on the testing set. You basically have adapted the model and hyper-parameters to produce the best model for that particular training set.

A common solution is to split the training set further to create a **validation set**. Now you have

- training set
- testing set
- validation set

You proceed as before but this time you use the validation set to test the performance and tweak the hyper-parameters. More specifically, you train multiple models with various hyper-parameters on the reduced training set (i.e., the full training set minus the validation set), and you select the model that performs best on the validation set.

Once you've selected the best performing model on the validation set, you train the best model on the full training set (including the validation set), and this gives you the final model.

Lastly, you evaluate this final model on the test set to get an estimate of the generalization error.

answered Aug 4 at 10:02



Aditya

220 3 14

In simple words define Training set, Test set, Validation set

-8

Training set: Is used for finding Nearest neighbors. Validation set: Is for finding different k which is applying to train set. Test set: Is used for finding the maximum accuracy and unseen data in future.

answered Jul 5 '18 at 14:16

Gaurav Pal

17 3