Самостоятельно обучить классификатор текстов на примере 20newsgroups
На примере 20 newsgroups попробовать разные параметры для сверток для классификации текстов

In [47]:

```
import pandas as pd
import numpy as np
from sklearn.datasets import fetch_20newsgroups
```

возьмем только 4 категории из 20 доступных в наборе данных

In [48]:

```
categories = ['alt.atheism', 'soc.religion.christian',
              'comp.graphics', 'sci.med']
```

In [49]:

```
train = fetch_20newsgroups(subset='train',
   categories=categories, shuffle=True, random_state=42)
```

In [50]:

```
train.target_names
```

Out[50]:

```
['alt.atheism', 'comp.graphics', 'sci.med', 'soc.religion.christian']
```

In [51]:

```
print(len(train.data))
len(train.filenames)
```

```
2257
```

Out[51]:

```
2257
```

In [52]:

```
print(train.target_names[train.target[6]])
train.target[:10]
```

```
soc.religion.christian
```

Out[52]:

```
array([1, 1, 3, 3, 3, 3, 3, 2, 2, 2], dtype=int64)
```

In [53]:

```
for i in train.target[:10]:
    print(train.target_names[i])
```

```
comp.graphics
comp.graphics
soc.religion.christian
soc.religion.christian
soc.religion.christian
soc.religion.christian
soc.religion.christian
sci.med
sci.med
sci.med
```

Выведем первые строки первого загруженного файла

In [ ]:

```

```

In [54]:

```
columns=['data', 'target']
```

In [55]:

```
df = pd.DataFrame(columns= columns)
```

In [56]:

```
# a = {'data': "\n".join(train.data[0].split("\n")),
#      'target': train.target[1]}
# df = df.append(a, ignore_index=True)
# df.iloc[0,0]
```

In [57]:

```
"\n".join(train.data[2].split("\n"))
```

Out[57]:

```
"From: djohnson@cs.ucsd.edu (Darin Johnson)\nSubject: Re: harrassed at work,
could use some prayers\nOrganization: =CSE Dept., U.C. San Diego\nLines: 63
\n\n(Well, I'll email also, but this may apply to other people, so\nI'll pos
t also.)\n\n>I've been working at this company for eight years in various\n>
engineering jobs.  I'm female.  Yesterday I counted and realized that\n>on s
even different occasions I've been sexually harrassed at this\n>company.\n\n
>I dreaded coming back to work today.  What if my boss comes in to ask\n>me
some kind of question...\n\nYour boss should be the person bring these probl
ems to.  If he/she\ndoes not seem to take any action, keep going up higher a
nd higher.\nSexual harrassment does not need to be tolerated, and it can be
an\nenormous emotional support to discuss this with someone and know that\nt
hey are trying to do something about it.  If you feel you can not\ndiscuss t
his with your boss, perhaps your company has a personnel\ndepartment that ca
n work for you while preserving your privacy.  Most\ncompanies will want to
deal with this problem because constant anxiety\ndoes seriously affect how e
ffectively employees do their jobs.\n\nIt is unclear from your letter if you
have done this or not.  It is\nnot inconceivable that management remains ign
orant of employee\nproblems/strife even after eight years (it's a miracle if
they do\nnotice).  Perhaps your manager did not bring to the attention of\nh
igher ups?  If the company indeed does seem to want to ignore the\nentire pr
oblem, there may be a state agency willing to fight with\nyou.  (check with
a lawyer, a women's resource center, etc to find out)\n\nYou may also want t
o discuss this with your paster, priest, husband,\netc.  That is, someone yo
u know will not be judgemental and that is\nsupportive, comforting, etc.  Th
is will bring a lot of healing.\n\n>So I returned at 11:25, only to find tha
t ever single\n>person had already left for lunch.  They left at 11:15 or s
o.  No one\n>could be bothered to call me at the other building, even though
my\n>number was posted.\n\nThis happens to a lot of people.  Honest.  I beli
eve it may seem\nto be due to gross insensitivity because of the feelings yo
u are\ngoing through.  People in offices tend to be more insensitive while\n
working than they normally are (maybe it's the hustle or stress or...)\nI've
had this happen to me a lot, often because they didn't realize\nmy car was b
roken, etc.  Then they will come back and wonder why I\ndidn't want to go (t
his would tend to make me stop being angry at\nbeing ignored and make me lau
gh).  Once, we went off without our\nboss, who was paying for the lunch :-)
\n\n>For this\n>reason I hope good Mr. Moderator allows me this latest indul
gence.\n\nWell, if you can't turn to the computer for support, what would\nw
e do?  (signs of the computer age :-)\n\nIn closing, please don't let the ha
teful actions of a single person\nharm you.  They are doing it because they
are still the playground\nbully and enjoy seeing the hurt they cause.  And y
ou should not\naccept the opinions of an imbecile that you are worthless - m
uch\nwiser people hold you in great esteem.\n-- \nDarin Johnson\ndjohnson@uc
sd.edu\n  - Luxury!  In MY day, we had to make do with 5 bytes of swap...\n"
```

In [58]:

```
for i in range(len(train.data)):
    a = {'data': "\n".join(train.data[i].split("\n")),
    'target': train.target[i]}
    df = df.append(a, ignore_index=True)
```

In [59]:

```
df.head()
```

Out[59]:

	data	target
0	From: sd345@city.ac.uk (Michael Collier)\nSubj...	1
1	From: ani@ms.uky.edu (Aniruddha B. Deglurkar)\...	1
2	From: djohnson@cs.ucsd.edu (Darin Johnson)\nSu...	3
3	From: s0612596@let.rug.nl (M.M. Zwart)\nSubjec...	3
4	From: stanly@grok11.columbiasc.ncr.com (stanly...	3

In [60]:

```
df = df.astype({"target": np.int64})
```

In [61]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2257 entries, 0 to 2256
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   data    2257 non-null   object
 1   target  2257 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 35.4+ KB
```

In [62]:

```
from string import punctuation
from stop_words import get_stop_words
from pymorphy2 import MorphAnalyzer
import re
```

In [63]:

```
sw = set(get_stop_words("en"))
exclude = set(punctuation)
morpher = MorphAnalyzer()

def preprocess_text(txt):
    txt = str(txt)
    txt = "".join(c for c in txt if c not in exclude)
    txt = txt.lower()
    txt = re.sub("not\s*", "not", txt)
    txt = [morpher.parse(word)[0].normal_form for word in txt.split() if word not in sw]
    return " ".join(txt)

df['data'] = df.data.apply(preprocess_text)
# df_val['text'] = df_val['text'].apply(preprocess_text)
# df_test['text'] = df_test['text'].apply(preprocess_text)
```

In [64]:

```
dfn = df.head(5)
```

In [65]:

```
num = []
```

In [66]:

```
for i in df['data']:
#     print(i, '\n')
    num.append(len(i))

print(max(num))
```

```
43693
```

In [67]:

```
tr = int(len(train.data)* 0.6)
tv = int(len(train.data)* 0.2)
print(tr, tv)
```

```
1354 451
```

In [68]:

```
df['target'].unique()
```

Out[68]:

```
array([1, 3, 2, 0], dtype=int64)
```

In [69]:

```
df_train = df.loc[:tr]
df_test = df.loc[tr:tv+tr]
df_val = df.loc[tv+tr:]

print(df_train.shape)
print(df_test.shape)
print(df_val.shape)
```

```
(1355, 2)
(452, 2)
(452, 2)
```

In [70]:

```
train_corpus = " ".join(df_train["data"])
train_corpus = train_corpus.lower()
```

In [71]:

```
import nltk
from nltk.tokenize import word_tokenize
nltk.download("punkt")

tokens = word_tokenize(train_corpus)
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\VoronkovSergey\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [72]:

```
tokens_filtered = [word for word in tokens if word.isalnum()]
```

In [92]:

```
max_words = 2000
max_len = 4000
num_classes = 4
```

In [93]:

```
from nltk.probability import FreqDist
dist = FreqDist(tokens_filtered)
tokens_filtered_top = [pair[0] for pair in dist.most_common(max_words-1)]
```

In [94]:

```
tokens_filtered_top[10:]
```

Out[94]:

```
['071',
 'sd345cityacuk',
 'subject',
 'images',
 'nntppostinghost',
 'hampton',
 'organization',
 'lines',
 '14',
 'anyone',
 'know',
 'good',
 'way',
 'standard',
 'pc',
 'applicationpd',
 'utility',
 'convert',
 'tifimgtga',
 'format',
 'also',
 'like',
 'hpgl',
 'plotter',
 'please',
 'response',
 'correct',
 'group',
 'thanks',
 'advance',
 'programmer',
 'computer',
 'unit',
 'mpcollierukaccity',
 'tel',
 '4778000',
 'x3769',
 'london',
 'fax',
 '4778565',
 'ec1v',
 '0hb']
```

In [95]:

```
vocabulary = {v: k for k, v in dict(enumerate(tokens_filtered_top, 1)).items()}
```

In [96]:

```
import numpy as np
def text_to_sequence(text, maxlen):
    result = []
    tokens = word_tokenize(text.lower())
    tokens_filtered = [word for word in tokens if word.isalnum()]
    for word in tokens_filtered:
        if word in vocabulary:
            result.append(vocabulary[word])
    padding = [0]*(maxlen-len(result))
    return padding + result[-maxlen:]
```

In [97]:

```
tokens = word_tokenize(df_train["data"][0].lower())
```

In [98]:

```
tokens_filtered = [word for word in tokens if word.isalnum()]
```

In [99]:

```
tokens_filtered[:5]
```

Out[99]:

```
['sd345cityacuk', 'michael', 'collier', 'subject', 'converting']
```

In [100]:

```
result = []
```

In [101]:

```
x_train = np.asarray([text_to_sequence(text, max_len) for text in df_train["data"]], dtype=
x_test = np.asarray([text_to_sequence(text, max_len) for text in df_test["data"]], dtype=np
x_val = np.asarray([text_to_sequence(text, max_len) for text in df_val["data"]], dtype=np.i
```

In [102]:

```
import random
import torch
import torch.nn as nn

seed = 0

random.seed(seed)
np.random.seed(seed)
torch.manual_seed(seed)
torch.cuda.manual_seed(seed)
torch.backends.cudnn.deterministic = True
```

In [103]:

```
from torch.utils.data import DataLoader, Dataset

class DataWrapper(Dataset):
    def __init__(self, data, target=None, transform=None):
        self.data = torch.from_numpy(data).long()
        if target is not None:
            self.target = torch.from_numpy(target).long()
        self.transform = transform

    def __getitem__(self, index):
        x = self.data[index]
        y = self.target[index] if self.target is not None else None

        if self.transform:
            x = self.transform(x)

        return x, y

    def __len__(self):
        return len(self.data)
```

In [104]:

```
class Net(nn.Module):
    def __init__(self, vocab_size=20, embedding_dim = 128, out_channel = 128, num_classes =
        super().__init__()
        self.embedding = nn.Embedding(vocab_size, embedding_dim)
        self.conv = nn.Conv1d(embedding_dim, out_channel, kernel_size=3)
        self.relu = nn.ReLU()
        self.linear = nn.Linear(out_channel, num_classes)

    def forward(self, x):
        output = self.embedding(x)
        #                       B  F  L
        output = output.permute(0, 2, 1)
        output = self.conv(output)
        output = self.relu(output)
        output = torch.max(output, axis=2).values
#         print(output.shape)
        output = self.linear(output)
#         print(output.shape)

        return output
```

In [105]:

```
# result_loss = pd.DataFrame(columns= ['epochs', 'batch', 'lr','optimizer' , 'criterion', '
# result_loss
```

In [132]:

```
model = Net(vocab_size=max_words)

print(model)
print("Parameters:", sum([param.nelement() for param in model.parameters()]))

# Training
epochs = 10
batch_size = 128
print_batch_n = 100
learning_rate = 0.001
criterion = nn.BCEWithLogitsLoss()
# criterion = nn.CrossEntropyLoss()
# optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate, momentum=0.9)
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

opt = str(optimizer).split()[0]
```

```
Net(
  (embedding): Embedding(2000, 128)
  (conv): Conv1d(128, 128, kernel_size=(3,), stride=(1,))
  (relu): ReLU()
  (linear): Linear(in_features=128, out_features=1, bias=True)
)
Parameters: 305409
```

In [133]:

```



model.train()
#model = model.cuda()

# optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
optimizer = optimizer
criterion = criterion





train_dataset = DataWrapper(x_train, df_train['target'].values)
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True,drop_last=True

val_dataset = DataWrapper(x_val, df_val['target'].values)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=True,drop_last=True)

loss_history = []

for epoch in range(1, epochs + 1):
    print(f"Train epoch {epoch}/{epochs}")
    for i, (data, target) in enumerate(train_loader):
        optimizer.zero_grad()

        # data = data.cuda()
        # target = target.cuda()

        # compute output
        output = model(data)

        # compute gradient and do SGD step
#         print(target.float().view(-1, 1))
        loss = criterion(output, target.float().view(-1, 1))
        loss.backward()

        optimizer.step()

        if i%print_batch_n == 0:
            loss = loss.float().item()
            print("Step {}: loss={}".format(i, loss))
            loss_history.append(loss)
            result_loss = result_loss.append({'epochs': epochs, 'batch': batch_size,'optimi
                                               'lr': learning_rate, 'loss': loss},
                                               ignore_index= True)
```

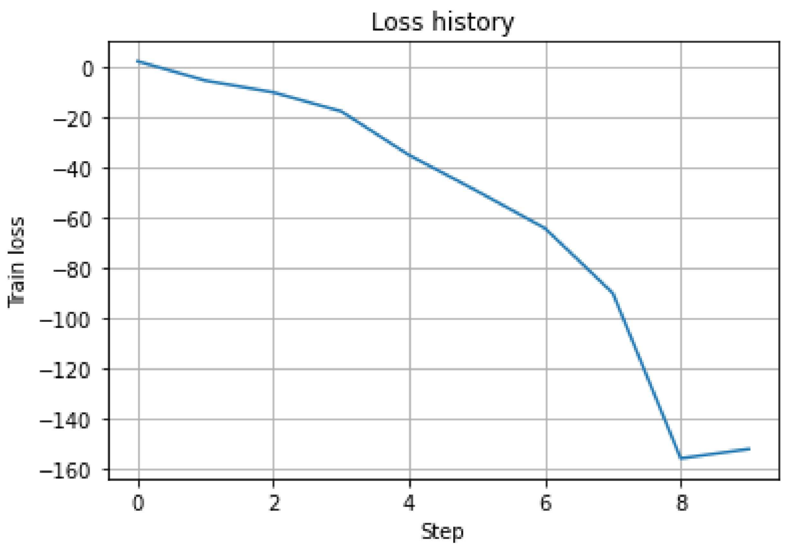
```
Train epoch 1/10
Step 0: loss=2.337507963180542
Train epoch 2/10
Step 0: loss=-5.361674785614014
Train epoch 3/10
Step 0: loss=-10.034090042114258
Train epoch 4/10
Step 0: loss=-17.586673736572266
Train epoch 5/10
Step 0: loss=-35.064857482910156
Train epoch 6/10
Step 0: loss=-49.338539123535156
```

```
Train epoch 7/10
Step 0: loss=-64.16010284423828
Train epoch 8/10
Step 0: loss=-90.03060913085938
Train epoch 9/10
Step 0: loss=-155.832275390625
Train epoch 10/10
Step 0: loss=-152.12391662597656
```

In [135]:

```
import matplotlib.pyplot as plt
%pylab inline
plt.title('Loss history')
plt.grid(True)
plt.ylabel('Train loss')
plt.xlabel('Step')
plt.plot(loss_history);
```

```
Populating the interactive namespace from numpy and matplotlib
```



In [136]:

```
result_loss.shape
```

Out[136]:

```
(40, 6)
```

In [137]:

```
result_loss.to_csv('loss.csv')
```

In [138]:

```
result_loss = pd.read_csv('loss.csv', index_col=0)
```

In [139]:

```
print(result_loss.head(60))
```

```
    epochs  batch     lr optimizer            criterion          loss
0       10    128  0.010      Adam  BCEWithLogitsLoss()  8.954821e-01
1       10    128  0.010      Adam  BCEWithLogitsLoss() -1.450427e+02
2       10    128  0.010      Adam  BCEWithLogitsLoss() -7.637862e+02
3       10    128  0.010      Adam  BCEWithLogitsLoss() -2.201274e+03
4       10    128  0.010      Adam  BCEWithLogitsLoss() -3.925404e+03
5       10    128  0.010      Adam  BCEWithLogitsLoss() -9.787680e+03
6       10    128  0.010      Adam  BCEWithLogitsLoss() -1.813418e+04
7       10    128  0.010      Adam  BCEWithLogitsLoss() -3.093842e+04
8       10    128  0.010      Adam  BCEWithLogitsLoss() -3.722721e+04
9       10    128  0.010      Adam  BCEWithLogitsLoss() -5.718064e+04
10      10    512  0.100       SGD  BCEWithLogitsLoss()  4.467733e-01
11      10    512  0.100       SGD  BCEWithLogitsLoss() -4.771332e+01
12      10    512  0.100       SGD  BCEWithLogitsLoss() -2.826782e+03
13      10    512  0.100       SGD  BCEWithLogitsLoss() -3.534390e+06
14      10    512  0.100       SGD  BCEWithLogitsLoss() -2.220668e+17
15      10    512  0.100       SGD  BCEWithLogitsLoss()           NaN
16      10    512  0.100       SGD  BCEWithLogitsLoss()           NaN
17      10    512  0.100       SGD  BCEWithLogitsLoss()           NaN
18      10    512  0.100       SGD  BCEWithLogitsLoss()           NaN
19      10    512  0.100       SGD  BCEWithLogitsLoss()           NaN
20      10    512  0.010       SGD  BCEWithLogitsLoss()  9.541477e-01
21      10    512  0.010       SGD  BCEWithLogitsLoss() -2.293588e+00
22      10    512  0.010       SGD  BCEWithLogitsLoss() -7.309615e+00
23      10    512  0.010       SGD  BCEWithLogitsLoss() -2.168841e+01
24      10    512  0.010       SGD  BCEWithLogitsLoss() -6.925711e+01
25      10    512  0.010       SGD  BCEWithLogitsLoss() -2.354380e+02
26      10    512  0.010       SGD  BCEWithLogitsLoss() -7.313280e+02
27      10    512  0.010       SGD  BCEWithLogitsLoss() -2.488165e+03
28      10    512  0.010       SGD  BCEWithLogitsLoss() -1.128388e+04
29      10    512  0.010       SGD  BCEWithLogitsLoss() -1.067501e+05
30      10    128  0.001      Adam  BCEWithLogitsLoss()  2.337508e+00
31      10    128  0.001      Adam  BCEWithLogitsLoss() -5.361675e+00
32      10    128  0.001      Adam  BCEWithLogitsLoss() -1.003409e+01
33      10    128  0.001      Adam  BCEWithLogitsLoss() -1.758667e+01
34      10    128  0.001      Adam  BCEWithLogitsLoss() -3.506486e+01
35      10    128  0.001      Adam  BCEWithLogitsLoss() -4.933854e+01
36      10    128  0.001      Adam  BCEWithLogitsLoss() -6.416010e+01
37      10    128  0.001      Adam  BCEWithLogitsLoss() -9.003061e+01
38      10    128  0.001      Adam  BCEWithLogitsLoss() -1.558323e+02
39      10    128  0.001      Adam  BCEWithLogitsLoss() -1.521239e+02
```

In [140]:

```
loss_1 = result_loss.head(10)
loss_2 = result_loss.iloc[10:20]
loss_2 = loss_2.reset_index(drop=True)
loss_3 = result_loss.iloc[20:30]
loss_3 = loss_3.reset_index(drop=True)
loss_4 = result_loss.iloc[30:40]
loss_4 = loss_4.reset_index(drop=True)
loss_4.head(20)
```

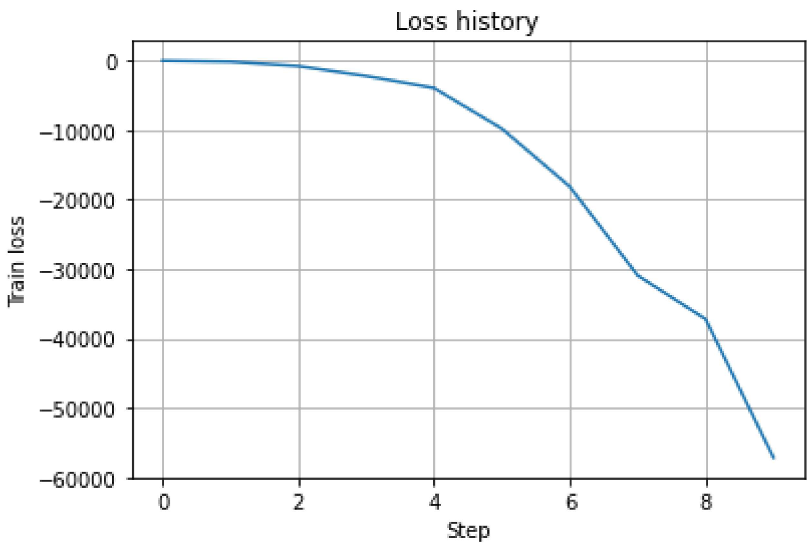
Out[140]:

	epochs	batch	lr	optimizer	criterion	loss
0	10	128	0.001	Adam	BCEWithLogitsLoss()	2.337508
1	10	128	0.001	Adam	BCEWithLogitsLoss()	-5.361675
2	10	128	0.001	Adam	BCEWithLogitsLoss()	-10.034090
3	10	128	0.001	Adam	BCEWithLogitsLoss()	-17.586674
4	10	128	0.001	Adam	BCEWithLogitsLoss()	-35.064857
5	10	128	0.001	Adam	BCEWithLogitsLoss()	-49.338539
6	10	128	0.001	Adam	BCEWithLogitsLoss()	-64.160103
7	10	128	0.001	Adam	BCEWithLogitsLoss()	-90.030609
8	10	128	0.001	Adam	BCEWithLogitsLoss()	-155.832275
9	10	128	0.001	Adam	BCEWithLogitsLoss()	-152.123917

In [141]:

```
import matplotlib.pyplot as plt
%pylab inline
plt.title('Loss history')
plt.grid(True)
plt.ylabel('Train loss')
plt.xlabel('Step')
plt.plot(loss_1['loss']);
```

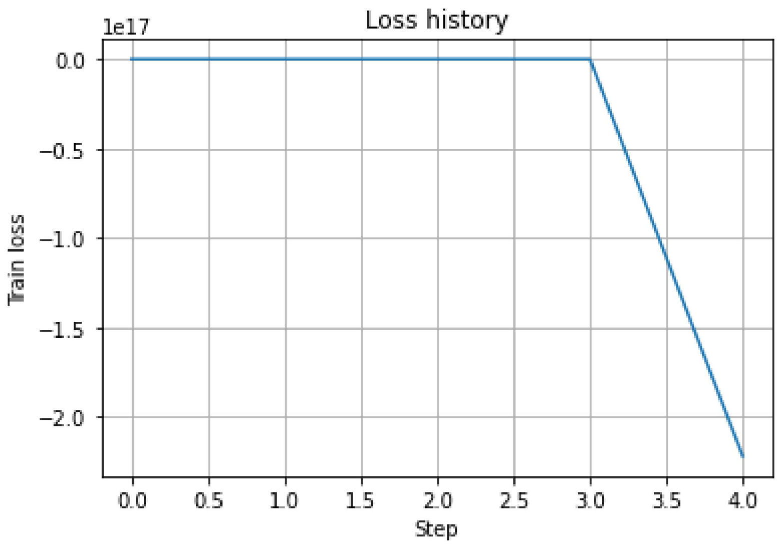
```
Populating the interactive namespace from numpy and matplotlib
```



In [142]:

```
import matplotlib.pyplot as plt
%pylab inline
plt.title('Loss history')
plt.grid(True)
plt.ylabel('Train loss')
plt.xlabel('Step')
plt.plot(loss_2['loss']);
```

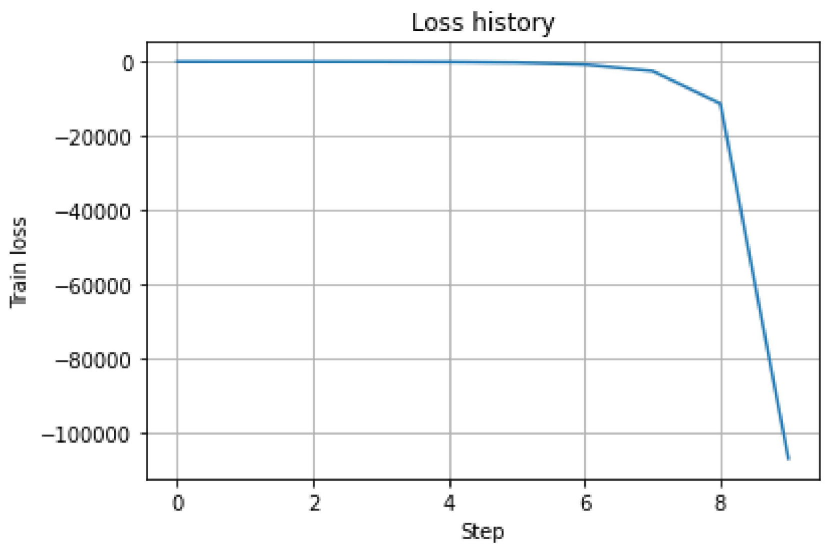
```
Populating the interactive namespace from numpy and matplotlib
```



In [143]:

```
import matplotlib.pyplot as plt
%pylab inline
plt.title('Loss history')
plt.grid(True)
plt.ylabel('Train loss')
plt.xlabel('Step')
plt.plot(loss_3['loss']);
```

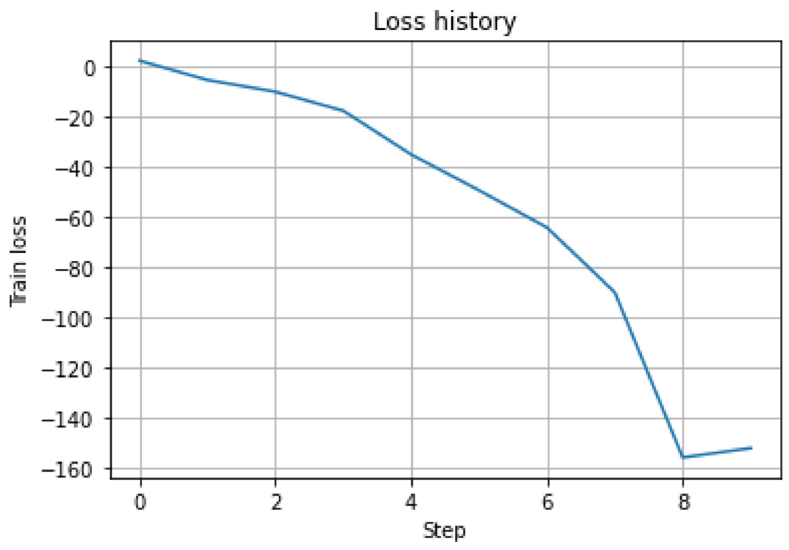
```
Populating the interactive namespace from numpy and matplotlib
```



In [144]:

```
import matplotlib.pyplot as plt
%pylab inline
plt.title('Loss history')
plt.grid(True)
plt.ylabel('Train loss')
plt.xlabel('Step')
plt.plot(loss_4['loss']);
```

```
Populating the interactive namespace from numpy and matplotlib
```



In [ ]:

```
```