9. Convolutional Neural Networks

Bookmark this page

Project due Nov 5, 2020 05:29 IST *Completed*

Next, we are going to apply convolutional neural networks to the same task. These networks have demonstrated great performance on many deep learning tasks, especially in computer vision.

You will be working in the files `part2-mnist/nnet_cnn.py` and `part2-mnist/train_utils.py` in this problem

Convolutional Neural Networks

3.0/3.0 points (graded)

We provide skeleton code `part2-mnist/nnet_cnn.py` which includes examples of some (**not all**) of the new layers you will need in this part. Using the PyTorch Documentation, complete the code to implement a convolutional neural network with following layers in order:

- A convolutional layer with 32 filters of size 3×3
- A ReLU nonlinearity
- A max pooling layer with size 2×2
- A convolutional layer with 64 filters of size 3×3
- A ReLU nonlinearity
- A max pooling layer with size 2×2
- A flatten layer
- A fully connected layer with 128 neurons
- A dropout layer with drop probability 0.5
- A fully-connected layer with 10 neurons

Note: We are not using a softmax layer because it is already present in the loss: PyTorch's `nn.CrossEntropyLoss` combines `nn.LogSoftMax` with `nn.NLLLoss`.

Without GPU acceleration, you will likely find that this network takes quite a long time to train. For that reason, we don't expect you to actually train this network until convergence. Implementing the layers and verifying that you get approximately 93% **training accuracy** and 98% **validation accuracy** after one training epoch (this should take less than 10 minutes) is enough for this project. If you are curious, you can let the model train longer; if implemented correctly, your model should achieve >99% **test accuracy** after 10 epochs of training. If you have access to a CUDA compatible GPU, you could even try configuring PyTorch to use your GPU.

After you successfully implement the above architecture, copy+paste your model code into the codebox below for grading.

Grader note:: If you get a NameEror “Flatten" not found, make sure to unindent your code.

Available Functions: You have access to the `torch.nn` module as `nn` and to the `Flatten` layer as `Flatten`; No need to import anything.

```
model = nn.Sequential(
          nn.Conv2d(1, 32, (3, 3)),
          nn.ReLU(),
          nn.MaxPool2d((2, 2)),
          nn.Conv2d(32, 64, (3,3)),
          nn.ReLU(),
          nn.MaxPool2d((2, 2)),
          Flatten(),
          nn.Linear(1600,128),
          nn.Dropout(p = 0.5),
          nn.Linear(128,10)
```

```
        )

```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

See full output

```
CORRECT
```

See full output

Submit You have used 3 of 25 attempts

Discussion

Hide Discussion

Topic: Unit 3 Neural networks (2.5 weeks):Project 3: Digit recognition (Part 2) / 9. Convolutional Neural Networks

Add a Post

Show all posts by recent activity

Hint: Viewing details of the model
Using the $\to rch\sum mary$ module you can output some useful information from each layer in the model, including output sizes af... 19
Pinned

Answer ungraded
I submitted my answer so it says 1 out of 25 attempts but the checker says 'unanswered' and I received 0/3.0 points for this problem... 4

? Dimension question
How do we get 1600 as input dimension after applying Flatten()? 2

Locally correct, grader gives error
My code run locally correct and give an accuracy of 99 after 10 epochs. Someone please explain the reason of grader error 5

Grader Processing Time?
I submitted my code but grader is still processing, what's expected time of completion? 15

☑ [python basics] Not sure when the CNN.forward being called in the code
Hi sorry about my basic question: in the file train_utils.py, when a the class model is called, but it doesn't specifically call CNN.forwar... 3

GPU Support
Hi Everyone, If you have CUDA compatible GPU and you wander hot to user full power of your GPU here are some small twiks: ##Ch... 1

? About the indent of Flatten..
I have tried varied indent and without indent case, well, it just doesn't work.... 11

use Flatten() instead of nn.Flatten()
The question does note to use Flatten(). I was initially using nn.Flatten(), but what we need to use is Flatten(). 5

? error - Operands could not be broadcast together with shapes (32, 26, 26) (32,)
this is the error I got when I ran the nnet_cnn.py model = nn.Sequential(nn.Convd2d(1, 32, (3, 3))) I tried add model step by step tha... 6

? operands could not be broadcast together with shapes
I don't know what I am missing here but I keep getting "operands could not be broadcast together with shapes" whenever I use any ... 2

Are we losing information here?
The output of the second convolution layer are 11x11 matrices, but we are converting them into 5x5 matrices with the second pooli... 5

? What do arguments in those layers mean? 7

‹ Previous

Next ›

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate