9. Convolutional Neural Networks

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

0.0/3.0 puntos (calificable)

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)),
        )

```

Presione ESC y después TAB o haga clic afuera del editor de código para salir

Sin Responder

Enviar Ha realizado 0 de 20 intentos

Discusión

Ocultar Discusión

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

Add a Post

Mostrar todas las publicaciones por actividad reciente

Some interesting read about dropout layer This article gives a very good explanation about what is dropout and why we need it in our nn... Pinned Community TA	2
A convolutional layer with 64 filters of size 3×3 Hi, We do have 32 channels, 13x13 layer followed by a 64-filter of size 3x3. Does that mean, e...	6
How does converting 32 features to 64 features work? I understand how to convert a single image input into 32 feature maps. But how does it work ...	8
What exactly is the grader expecting? I succesfully ran my model locally with an "Accuracy on test set": 0.9903846153846154 But w...	12
The grader showed initially a 2.66/3.00 points. Now it shows 0.0/3.0 why? I ran all the 10 layers and got accuracy of 0.976% on the test. The training had accuracy of 0.9...	1
Suggestion! I think this part of the course requires more exercises on how to compute the dimensionality ...	1

	A PyTorch tutorial – deep learning in Python Hope it helps answer some questions: [Pytorch tutorial deep learning][1] Good luck! [1]: https...	1
	[STAFF] Project-3 grades mixed up!! Until few hours back, my project-3 score was showing more than 60% (solved until Problem#...	3
?	Flatten is not defined what should the input of flatten be or is this the point of the exercise because every thing els...	3
	Dimensions Discrepancy input image: 28 x 28 x 1 After first 3x3 conv layer, result is 26 x 26 x 32 After ReLu, result is 26...	4
	A fully connected layer with n neurons I am clueless about how to implement a "fully connected layer" in torch. I looked it up, but di...	7
	It is too slow, how to make it works with CUDA? torch.cuda.is_available() #returns True, so: torch.cuda.set_device(0) #before torch.manual_se...	3
	[STAFF] minor issue in train_model() `train_model()` in `train_utils.py` takes a `n_epochs` parameter (defaulted to `n_epochs=3...	1
?	[Concept Check] Image Dimensions - Steps Please, anyone can check my steps? I'm reviewing the Project 3 (I finished last week) #Questio...	1
	Is the nnet_cnn.py running OK out of the box? I see this error when trying to run the code: Epoch 1: 0%| | 0/1687 [00:00<?, ?it/s] Traceback (...	2
	Error I ran into the following error, and honestly I think it's very specific to this case, because I didn'...	5
	Visualization of CNN If you ever ask how visually CNN works, here is a beautiful explanation which I found useful: ...	1
☑	[STAFF] Need guidance on how to calculate proper # of parameters at each CNN Layer Hi, I'm geting following error and not sure how to calculate the proper # parameters when I h...	6
?	[Staff]is the grader down now?	3