

DESIGN DECISION AND RESULTS ON hw3_pytorch_cifar.py

Optimizer

I tried 2 different optimizers, SGD and Adam. For both of them, I tried turning on and off their internal optimizing modes (nesterov and amsgrad). I also tried different learning rates: [0.05, 0.01, 0.001]. I did all the possible combination of the previous optimizers. Results were as follows:

SGD; nesterov:False; lr:0.05. Accuracy:(67.08)
SGD; nesterov:False; lr:0.01. Accuracy:(67.20)
SGD; nesterov:False; lr:0.001. Accuracy:(48.84)
SGD; nesterov:True; lr:0.05. Accuracy:(66.92)
SGD; nesterov:True; lr:0.01. Accuracy: (66.60)
SGD; nesterov:True; lr:0.001. Accuracy: (49.71)

Adam; amsgrad:False; lr:0.05. Accuracy: (10.00)
Adam; amsgrad:False; lr:0.01. Accuracy:(10.00)
Adam; amsgrad:False; lr:0.001. Accuracy: (68.46)
Adam; amsgrad:True; lr:0.05. Accuracy (10.00)
Adam; amsgrad:True; lr:0.01. Accuracy: (10.00)
Adam; amsgrad:True; lr:0.001. Accuracy: (68.85)

We observe that, for this network architecture, the best optimizer is Adam; amsgrad:True; lr:0.001, with accuracy (68.85).

Network architecture

After deciding over the best optimizer, I moved to optimizing my network. After some iterations, I found that an optimal election of network architecture was:

- 4 convolution layers, with relu activations for each of them. I tried different filter sizes in them. Finally choosing (3,2,4 and 2) for filter sizes, and different output_channels (32,64,28,64), gave good results.
- max_pool with dimension 2
- dropouts of 0.25 and 0.5
- 2 fully connected layers

With this network, and the previously selected best optimizer, I got 7323 / 10000 correct (73.23 accuracy)

```
felipe@felipe-Ubuntu:~/2020-Winter/CV/cnn-from-scratch$ python3 gradient_checker.py
```

```
Testing Conv2d:
```

```
Shape Match = True
```

```
Result Match = True
```

```
Result Match = False
```

```
Result Match = False
```

```
Result Match = False
```

```
Testing Linear:
```

```
Shape Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Testing MaxPool2d:
```

```
Shape Match = True
```

```
Result Match = True
```

```
Result Match = False
```

```
Testing ReLU:
```

```
Shape Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Testing BatchNorm1d:
```

```
Shape Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Result Match = True
```

```
Testing cross_entropy_loss_with_softmax:
```

```
Shape Match = True
```

```
Result Match = True
```

```
Result Match = True
```