

기초인공지능 4차 과제

20171637 박성우

MLP

Input layer 와 output layer 를 포함해서 hidden layer 5개를 추가해서 총 7개의 layer 를 쌓았다. 먼저 image 를 문제 1번에서 구현한 flatten 함수를 통해 flatten 시키고 Pytorch 에 nn.linear 모델을 사용해서, 첫 hidden layer 는 input size 그대로 하였고, 두번째 hidden layer 부터 5번째 hidden layer 는 노드 수를 100개로 줄였다. 그리고 마지막 layer 에서 output size 로 바꾸어 주었다.

```
nn.Linear(input_size, input_size),
self.activation_func,
nn.Linear(input_size, 100),
self.activation_func,
nn.Linear(100, 100),
self.activation_func,
nn.Linear(100, 100),
self.activation_func,
nn.Linear(100, 100),
self.activation_func,
nn.Linear(100, output_size),
```

CNN

CNN 을 구성하는 데에 있어 문제에서 요구한대로 Conv2d 와 MaxPool 을 사용해서 구성하였고, FC층을 구성하기에 앞서 먼저 view 함수로 flatten 해주었고 사이즈를 5x5x24 -> 300 -> 300 -> 10 으로 4개의 층을 가지도록 구성하였다.

```

self.activation_func=active_func
##### Write your Code Here #####
self.conv1 = nn.Conv2d(input_channel, 12, kernel_size=3)
self.conv2 = nn.Conv2d(12, 12, kernel_size=3)
self.conv3 = nn.Conv2d(12, 24, kernel_size=3)
self.conv4 = nn.Conv2d(24, 24, kernel_size=3)
self.pool = nn.MaxPool2d(2, 2)
self.fc1 = nn.Linear(5*5*24, 300)
self.fc2 = nn.Linear(300, 300)
self.fc3 = nn.Linear(300, 100)
self.fc4 = nn.Linear(100, 10)
#####

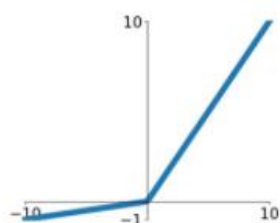
def forward(self, output):
##### Write your Code Here #####
output = self.activation_func(self.conv2(self.conv1(output)))
output = self.pool(output)
output = self.activation_func(self.conv4(self.conv3(output)))
output = self.pool(output)
output = output.view(output.size(0), -1)
output = self.activation_func(self.fc1(output))
output = self.activation_func(self.fc2(output))
output = self.activation_func(self.fc3(output))
output = self.fc4(output)
return output
#####

```

Activation function

Activation function 으로는 LeakyReLU 함수를 선택했다. 이 함수는

Activation Functions



Leaky ReLU

$$f(x) = \max(0.01x, x)$$

이렇게 생겼는데 이 함수를 선택한 이유는 기존 음수 부분에서도 기울기를 주어서 vanishing gradient 문제를 해결할 수 있고 기존의 RELU 가 가지고 있는 dying relu 문제를 해결할 수 있어서 선택하였다. Leaky ReLU 가 기존의 ReLU 보다 느리다고 알려져 있지만 CIFAR10 의 dataset 은 큰 편이 아니기에 괜찮다고 판단하였다.

마지막으로 최종 accuracy 이다.

