Avoiding Overfitting:

« dropout »

```
class    Classifier(nn.module):
    def __init__(self):
        ______
        ______
        ______
        self.dropout = nn.Dropout(p=0.2)

    def forward(self, x):
        ____________
        ____________
        x = self.dropout( F.relu (self.fc1(x)))
        x = self.dropout( . . .
        ______
        x = F.log_softmax(self.fc4(x), dim=1)

        return x
```

0.2
drop probability

apply dropout at every hidden layer

no dropout for output layer

Note: when we do validation, we don't want any dropouts, so:

```
with torch.no_grad():

    model.eval()  → turns off dropouts

model.train()  → revert to train mode
```