Performance Check for Different Models

Model (Version-7)

class FNNv7(nn.Module):  
 def \_\_init\_\_(self, input\_dim, hidden\_dim, output\_dim):  
 super(FNNv7, self).\_\_init\_\_()  
 self.fc1 = nn.Linear(input\_dim, hidden\_dim)  
 self.relu1 = nn.ReLU()  
 self.dropout1 = nn.Dropout(0.2) *# Adjust dropout rate to 0.2* self.fc2 = nn.Linear(hidden\_dim, hidden\_dim // 2) *# Use integer division //* self.relu2 = nn.ReLU()  
 self.dropout2 = nn.Dropout(0.3) *# Adjust dropout rate to 0.3* self.fc3 = nn.Linear(hidden\_dim // 2, hidden\_dim // 4) *# Use integer division //* self.relu3 = nn.ReLU()  
 self.dropout3 = nn.Dropout(0.3) *# Adjust dropout rate to 0.3* self.fc4 = nn.Linear(hidden\_dim // 4, output\_dim)  
  
 def forward(self, x):  
 x = self.fc1(x)  
 x = self.relu1(x)  
 x = self.dropout1(x)  
 x = self.fc2(x)  
 x = self.relu2(x)  
 x = self.dropout2(x)  
 x = self.fc3(x)  
 x = self.relu3(x)  
 x = self.dropout3(x)  
 x = self.fc4(x)  
 return x

Summary:

* Fully Connected Layers: 4
* Activation Function: ReLU
* Variation of Dropout: 0.2, and 0.3
* Variation of the number of neurons in hidden layers

Performance Check:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Epochs | Batch | Learning | Loss | Optim | Hidden  Dim | Train  Loss | Test  Loss | Train Acc | Test Acc |
| 80% | 200 | 8 | 0.01 | CrossEntropy | Adam | 512 |  |  |  |  |