VST Lab1 Image Classification: Methodology Sharing

Presented by Po-Jui Su

Model Design

```python
class ClassificationModel(nn.Module):
    def __init__(self, num_classes=100):
        super(ClassificationModel, self).__init__()

        # Reduced number of filters in convolutional layers
        self.conv1 = nn.Conv2d(3, 3, kernel_size=3, stride=1, padding=1)  # 3 filters
        self.conv2 = nn.Conv2d(3, 6, kernel_size=3, stride=1, padding=1)  # 6 filters
        self.conv3 = nn.Conv2d(6, 8, kernel_size=3, stride=1, padding=1)  # 8 filters

        self.pool = nn.MaxPool2d(4, 4)  # Pooling to reduce the spatial dimensions by a factor of 4
        self.dropout = nn.Dropout(0.15)  # Dropout to reduce overfitting (15% dropout rate)

        # Fully connected layer size is adjusted for the output size of the conv3 layer (8 channels, 3x3)
        self.fc1 = nn.Linear(8 * 3 * 3, num_classes)  # 8 filters with 3x3 feature map size

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))  # After conv1 -> ReLU -> pool (224x224 -> 56x56)
        x = self.pool(F.relu(self.conv2(x)))  # After conv2 -> ReLU -> pool (56x56 -> 14x14)
        x = self.pool(F.relu(self.conv3(x)))  # After conv3 -> ReLU -> pool (14x14 -> 3x3)

        # Flatten the output for the fully connected layer
        x = x.view(-1, 8 * 3 * 3)  # Flatten the tensor (8 channels, 3x3 feature map size)

        x = self.dropout(x)  # Apply dropout to reduce overfitting
        x = self.fc1(x)  # Fully connected layer to get class scores

        return x
```

Model Compression skills:

(1) Reducing the number of filters.

(2) Using **larger kernel** pooling layers to decrease the size of the feature map.

(3) **Fully Connected layer** might be replaced by **Global Average Pooling** (But I failed to do this.)

Parameters Calculation:

conv1:

(3 x 3 x 3 + 1) x 3 Filters = 84

conv2:

(3 x 3 x 3 + 1) x 6 Filters = 168

conv3:

(6 x 3 x 3 + 1) x 8 Filters = 440

FC Layer:

(8 x 3 x 3 + 1) x 100 (num_classes) = 7300

Hyperparameter Settings and Training Strategies

(1) Hyperparameter Settings:

(a) (batch size , Epoch): (32 , 100)

(b) (optimizer , init_lr): (Adam , **1E-3**)

(c) Loss Function: CrossEntropy

(2) Training Strategies

(a) Use **Early Stopping** (trigger if no improvement f**or more than 10 epochs**).

(b) Use **ReduceLROnPlateau** to decrease the learning rate (**multiply 0.5** if no improvement **for more than 5 epochs**).

(c) Both of the above are based on **validation Top-5 accuracy**.