CS 6476 PS4

Name
GT Email
GT ID

Part 1: Standard Scaler: Why did we use StandardScaler instead of looping over all the dataset twice for mean and standard deviation? Why a simple loop will not be a good choice in a deployed production grade ML system?

<text answer here>

Part 1: Why do we normalize our data (0 mean, unit standard deviation)?

<text answer here>

Part 3: Loss function. Why did we need a loss function?

<text answer here>

Part 3: [2 pt] KL-divergence is defined as

$$D_{KL}(P||Q) = \mathbb{E}_{X\sim P}\left[\log\frac{P(x)}{Q(x)}\right]$$

Cross-entropy is defined as

$$H(P,Q) = -\mathbb{E}_{X\sim P}\log Q(x)$$

How would you write cross-entropy in terms of KL-divergence

<text answer here>

Part 5: Training SimpleNet

<Loss plot here>

<Accuracy plot here>

Final training accuracy value:

Final validation accuracy value:

Part 6: Simple Segmentation Net

Class Index	Class name	Simple Segmentation Net Class IoU
0	Building	
1	Tree	
2	Sky	
3	Car	
4	SignSymbol	
5	Road	
6	Pedestrian	
7	Fence	
8	Column_Pole	
9	Sidewalk	
10	Bicyclist	

Validation mIoU:

Number of Epochs: ____

Part 6: Simple Segmentation Net

Paste a figure of the generated semantic segmentation from Colab. It should be a 2x3 grid, with ground truth on the top row, and your predictions on the bottom row.

Part 6: Simple Segmentation Net

Which classes have the lowest mIoU? Why might they be the most difficult? Provide an example RGB image from Camvid that illustrates your point.

Conclusion: briefly discuss what you have learned from this project.

<Text solution here>

Theory: Neural Nets

1. Given are the layers of a neural network. Calculate the total number of learnable parameters for each layer and in total.

```
(conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
(maxpool1): MaxPool2d(kernel_size=2, stride=2, padding=0)
(conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
(maxpool2): MaxPool2d(kernel_size=2, stride=2, padding=0)
(fc1): Linear(in_features=3200, out_features=10, bias=True)
(softmax): Softmax(dim=1)
```

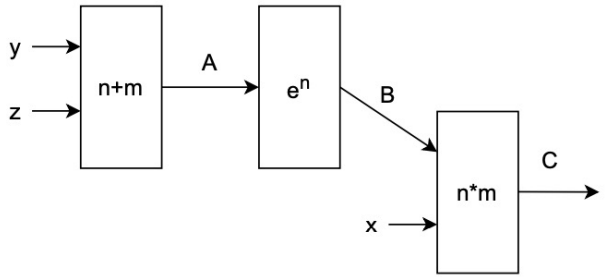
a) Conv 1: <answer here>
b) Conv 2: <answer here>
c) FC1: <answer here>
d) Total (sum all 3): <answer here>

Theory: Neural Nets (cont.)

Here is the computational graph for the function $f(x, y, z) = x * e^{y+z}$. We want to compute the partial derivative of this function with respect to each input. Here are the three intermediate functions

$A = y + z$ $B = e^A$ $C = x * B$

From these we build the feed forward computation graph, with inputs n, m. The top input is n and the bottom is m.



a. What is the correct expression for $\frac{\partial C}{\partial x}$?

Type equation here.

b. What is the correct expression for $\frac{\partial C}{\partial B}$?

Type equation here.

c. What is the correct expression for $\frac{\partial B}{\partial A}$?

Type equation here.

d. What is the correct expression for $\frac{\partial A}{\partial y}$?

Type equation here.

e. What is the correct expression for $\frac{\partial C}{\partial y}$?

Type equation here.

EC1.1: Screenshot of your get_data_augmentation_transforms()

<Screenshot here if attempted; do not delete the slide if not attempted>

EC1: Training to solve overfitting

<Loss plot here>

<Accuracy plot here>

Final training accuracy value:

Final validation accuracy value:

EC2: PSPNet

Class Index	Class name	PSPNet Class IoU
0	Building	
1	Tree	
2	Sky	
3	Car	
4	SignSymbol	
5	Road	
6	Pedestrian	
7	Fence	
8	Column_Pole	
9	Sidewalk	
10	Bicyclist	

Validation mIoU:

Number of Epochs: ____

EC2: PSPNet

Paste a figure of the generated semantic segmentation from Colab. It should be a 2x3 grid, with ground truth on the top row, and your predictions on the bottom row.