2023 DNN EndSem Makeup

1. Consider a combined Lenet-5 and a single-layer RNN based visual captioning system that is trained to generate the sequence of small characters corresponding to ten digits, 0 to 9. For example, an input image of '7' will generate the output character sequences s, e, v, e, n. An input image that is not of a digit will generate the output n, o, n, e. Assuming one hot representation is used for both input and output,

   (a) What is the minimum number of input nodes and minimum number of output node required in RNN? [1]
   Unique characters are 15. Also start of word and end of word tokens are needed. So, minimum no. of i/o nodes =17

   (b) Assuming linear combinations of the output of last convolution layer (after subsampling and unrolling) is used to initialize the RNN hidden layer, how many trainable parameters will be needed, excluding the CNN convolution parameters? Assume 50 hidden nodes are used in RNN. Show all steps clearly. No attention is used. [4]
   Lenet-5 feature vector size post unrolling = 5*5*16=400
   Training weights for computing linear combinations of Lenet-5 feature vectors to RNN hidden units = 400*50+50.
   Total number of trainable parameters for RNN excluding Lenet-5 parameters =20050 +(50+17) * 50+50+50* 17+17

   (c) Over how many time steps, does the loss function need to be evaluated during training? [1]
   The longest character sequences is 6, including the end of word token. So, loss function needs to be evaluated over 6 time steps.

2023 DNN EndSem Makeup

2. Compute the dimensions of W^Q, W^K, W^V, W^O if the input dimensions and attention dimensions are given as 1024 and the query dimensions is 256 and 512 key and value dimensions and there are 6 multi-heads. Compute the dimensions of the attention vector. [5]
One mark each

$$\text{Dimensions of } W^Q = (h \times d) \times d_q = (6 \times 1024) \times 256$$
$$\text{Dimensions of } W^K = (h \times d) \times d_k = (6 \times 1024) \times 512$$
$$\text{Dimensions of } W^V = (h \times d) \times d_v = (6 \times 1024) \times 512$$
$$\text{Dimension of attention vector} = d_v = 6 \times 512$$
$$\text{Output Projection Matrix} W^O = h \times d_v \times d = 6 \times 512 \times 1024$$

3. Consider a simplified version of the NiN architecture with one NiN block followed by global average pooling and a sigmoid output layer. Each NiN block consists of the following layers: 1x1 convolutional layer with 32 filters, two separate 3x3 convolutional layers, each with 64 filters, two separate 5x5 convolutional layers with 24 filters, ReLU activation function applied after each convolutional layer. Assume that the input to the NiN architecture is a greyscale image with dimensions of 512x512 pixels. Calculate the following:

   (a) The number of parameters (weights and biases) in one NiN block. [4]

   (b) The total number of parameters in the entire NiN architecture. [1]

   Rubric

   - Number of parameters in one NiN block:

   $$\text{1x1 Convolutional layer } Weights + bias = 1 \times 1 \times 1 \times 32 + 32 = 64 \quad [1]$$
   $$\text{3x3 Convolutional layer } Weights + bias = 3 \times 3 \times 1 \times 64 + 64 = 640$$
   $$\text{For three 3x3 Convolutional layers } = 2 \times 640 = 1280 \quad [1]$$
   $$\text{5x5 Convolutional layer } Weights + bias = 5 \times 5 \times 1 \times 24 + 24 = 624$$
   $$\text{For two 5x5 Convolutional layers } = 2 \times 624 = 1248 \quad [1]$$
   $$\text{Total parameters in onw NiN block } = 64 + 1280 + 1248 = 2592 \quad [1]$$

   - Since the NiN architecture consists of one NiN block followed by global average pooling and a softmax output layer, the NiN block contains 47230 parameters. [1]

4. Two historians approach you for your deep learning expertise. They want to classify images of historical objects into 3 classes depending on the time they were created: Antiquity ($y = 0$), Middle Ages ($y = 1$) and Modern Era ($y = 2$)

(A) Class: Antiquity (B) Class: Middle Ages (C) Class: Modern Era

You come up with a CNN classifier as given below. Fill in the values of xxAxx to xxFxx, such that line 1 results in 501x501x64 and line 2 results in 246x246x128. Line 3 reduce the output size by half. Write the statement for global average pooling in place of xxFxx. All the answers should be written with the correct syntax. [8]

```
cnnModel = models.Sequential()
cnnModel.add(layers.Conv2D(64, (11,11), activation='relu',
input_shape=(1024,1024,3) ))
cnnModel.add(layers.MaxPooling2D((2,2)))
cnnModel.add(layers.Conv2D(xxAxx, (7,7), activation='relu')) #line 1
cnnModel.add(layers.Dropout(xxBxx))
cnnModel.add(layers.MaxPooling2D((2,2)))
cnnModel.add(layers.Conv2D(128, xxCxx, activation='relu')) #line 2
cnnModel.add(layers.Dropout(0.3))
cnnModel.add(layers.MaxPooling2D(xxDxx) # line 3
cnnModel.add(layers.Conv2D(256, (3,3), xxExx))
cnnModel.add(layers.MaxPooling2D((2,2)))
cnnModel.add(layers.Dropout(0.2))
cnnModel.add(layers.Flatten())
# xxFxx
cnnModel.add(layers.Dense(xxGxx, activation='xxHxx' ))
```

One mark each

- xxAxx
  cnnModel.add(layers.Conv2D(64, (7,7), activation='relu'))
- xxBxx
  cnnModel.add(layers.Dropout(0.3))
- xxCxx
  cnnModel.add(layers.Conv2D(128, (5,5), activation='relu'))
- xxDxx
  cnnModel.add(layers.MaxPooling2D((2,2)))
- xxExx
  cnnModel.add(layers.Conv2D(256, (3,3), activation='relu'))
- xxFxx - global average pooling
- xxGxx and xxHxx
  cnnModel.add(layers.Dense(3, activation='softmax'))

5. A deep learning researcher is designing an GRU-based neural network for weather forecasting. The researcher plans to use a single-layer GRU with specific architectural parameters. The GRU layer is fed with 24 numeric features of the last 15 days. and a hidden size of 128. The neural network is supposed to provide an 5 numeric output for the next 5 days.

   (a) Based on these specifications, calculate the total number of parameters required for this GRU layer. Show Gate wise necessary calculations. [4]

   (b) If the researcher adds a bidirectional GRU, how will the number of parameters get affected? Is this addition justified for the application. [2]

(a) One mark each

$$\text{Parameters for cell state} = n_i \times n_h + n_h \times n_h + n_h = 24 \times 128 + 128 \times 128 + 128$$
$$\text{Parameters for reset gate} = n_i \times n_h + n_h \times n_h + n_h = 24 \times 128 + 128 \times 128 + 128$$
$$\text{Parameters for update gate} = n_i \times n_h + n_h \times n_h + n_h = 24 \times 128 + 128 \times 128 + 128$$
$$\text{Parameters for output projection} = n_h \times n_o + n_o = 128 \times 5 + 5$$

2023 DNN EndSem Makeup

(b) If we add a bidirectional GRU layer, the number of parameters will double because we have forward and backward passes, effectively doubling the weights and biases. Therefore, we need to multiply the total number of parameters by 2. Reset Gate and Update Gate: Total parameters for each gate: $2 \times (24 \times 128 + 128 \times 128) + 2 \times 128$
No. Future prediction does not affect the past weather conditions.