National University of Computer and Emerging Sciences

Deep Learning For Perception (CS4045)

Date: May 22nd 2024

Course Instructor(s)

Ms. Sumaiyah Zahid

Final Exam

Total Time: 3 Hours
Total Marks: 100
Total Questions: 7

Semester: SP-2024
Campus: Karachi
Dept: Computer Science

Student Name	Roll No	Section	Student Signature

Do not write below this line

Attempt all the questions.

CLO # 1: Student should be able to describe what Deep Learning is and the skill sets needed for Deep Learning

Q1: Write short answers in a maximum of 3 to 4 lines. [1*15=15 marks]

1. What do you understand by transfer learning? Name a few commonly used transfer learning models.
   Transfer learning is the process of transferring the learning from one model to another model without having to train it from scratch. It takes critical parts of a pre-trained model and applies them to solve new but similar machine-learning problems.
   Some of the popular transfer learning models are:
   VGG-16, BERT, GTP-3, Inception V3, XCeption
2. Can we have the same bias for all neurons of a hidden layer?
   Essentially, you can have a different bias value at each layer or at each neuron as well. However, it is best if we have a bias matrix for all the neurons in the hidden layers as well.A point to note is that both these strategies would give you very different results.
3. In a neural network, what if all the weights are initialized with the same value?
   In simplest terms, if all the neurons have the same value of weights, each hidden unit will get exactly the same signal. While this might work during forward propagation, the derivative of the cost function during backward propagation would be the same every time.In short, there is no learning happening by the network! What do you call the phenomenon of the model being unable to learn any patterns from the data? Yes, underfitting. Therefore, if all weights have the same initial value, this would lead to underfitting.
4. True or false? A language model usually does not need labels for its pretraining.

True: The pretraining is usually self-supervised, which means the labels are created automatically from the inputs (like predicting the next word or filling in some masked words).

5. Which of these types of models would you use for classifying text inputs according to certain labels? And why? (An encoder model, A decoder model, A seq-to-seq model)
   An encoder model: An encoder model generates a representation of the whole sentence which is perfectly suited for a task like classification.
6. What are the common data structures used in Deep Learning?
   Deep Learning goes right from the simplest data structures like lists to complicated ones like computation graphs. Here are the most common ones:
   List, Matrix, Dataframe, Tensors, Computation Graphs.
7. How is backpropagation different in RNN compared to ANN?
   In Recurrent Neural Networks (RNNs), backpropagation through time (BPTT) is used to update weights by unrolling the network over time.
8. Why don't we use Long Short-Term Memory Networks for smaller datasets or problems?
   LSTM networks have high computational complexity and parameter overhead, making them prone to overfitting on smaller datasets or simpler problems. Additionally, training LSTMs on small datasets may not effectively exploit their ability to capture long-term dependencies, leading to suboptimal performance compared to simpler models.
9. Explain the significance of the RELU activation function in a Convolution Neural Network.
   Moreover, RELU is a non-linear activation function. This operation is applied to each pixel and replaces all the negative pixel values in the feature map with zero.
10. Does the size of the feature map always reduce upon applying the filters? Explain why or why not.
    The size of the feature map in a CNN may not always reduce upon applying filters. Factors such as padding, stride, and border effects influence whether the spatial dimensions of the feature map decrease or remain the same.
11. List down the hyperparameters of a Pooling Layer.
    Filter size
    Stride
    Max or average pooling
12. Can we use CNN to perform Dimensionality Reduction? If Yes then which layer is responsible for dimensionality reduction particularly in CNN?
    Pooling layers
13. When would you use MLP, CNN, and RNN
    Multilayer Perceptrons, or MLPs for short are the classical type of neural network. They are very flexible and can be used generally to learn a mapping from inputs to outputs, however, they are perhaps more suited to classification and regression problems.
    Convolutional Neural Networks, or CNNs, were developed and are best used for image classification. But they can also be used generally with data that has a spatial structure, such as a sequence of words, and can be used for document classification.
    Recurrent Neural Network or RNNs, was developed for sequence prediction and is well suited for problems that have a sequence of input observations or a sequence of output observations. They are suitable for text data, audio data, and similar applications.

14. What are the applications of a Recurrent Neural Network (RNN)?
    Natural Language Processing (NLP), Time Series Analysis, Speech Recognition and Synthesis,Sequence-to-Sequence Learning
15. What are the limitations of the Transformer?
    Inefficiency in processing long sequences, lack of fine-grained interpretability, and reliance on large-scale pretraining data. Additionally, it struggles with capturing hierarchical structure and complex contextual information and requires significant computational resources for training large models.

CLO # 2: Students should be able to understand supervised and unsupervised methods of Deep Learning

Q2: **[10+3+2 = 15 marks]**

a) Consider the AlexNet architecture which consists of the following layers:

- Conv1: 96 filters of size 11×11, stride 4, padding 0 55×55×96
- (11*11*3*96)+96=34944
- Max-Pooling: Pool size 3×3 stride 2 27×27×96
- Conv2: 256 filters of size 5×5, stride 1, padding 2 27×27×256
- total parameters: (5 *5*96*256)+256= 614656
- Max-Pooling: Pool size 3×3, stride 2 13×13×256
- Conv3: 384 filters of size 3×3, stride 1, padding 1 13×13×384
- Conv4: 384 filters of size 3×3, stride 1, padding 1 13×13×384
- Conv5: 256 filters of size 3×3, stride 1, padding 1 13×13×256
- Max-Pooling: Pool size 3×3, stride 2 6×6×256

Given an input image of size 227×227×3 (height × width × channels):

1. Calculate the spatial dimensions (height, width, and channels) of the output after each convolutional and pooling layer. **Output= (Input +2p-k)/s +1**
2. Determine the total number of parameters in Conv1 and Conv2 layers.
   Num_params= [i* (f*f)*o] +o

b) You have a dataset D1 with 1 million labeled training examples for classification, and dataset D2 with 100 labeled training examples. Your friend trains a model from scratch on dataset D2. You decide to train on D1, and then apply transfer learning to train on D2. State one problem your friend is likely to find with his approach. How does your approach address this problem?
Friend is likely to see overfitting. The model is not going to generalize well to unseen data. By using transfer learning and freezing the weights in the earlier layers, you reduce the number of learnable parameters, while using the weights which have been pre-trained on a much larger dataset.

c) What role does the 'stride' parameter play in convolutional layers of text-based CNNs? How might adjusting the stride impact the model's ability to extract meaningful features from text?
It controls the distance between filter applications, impacting the granularity of feature extraction or window size.

CLO # 2: Students should be able to understand supervised and unsupervised methods of Deep Learning

Q3: **[10+5+5=20 marks]**

a) Suppose we have a Sequence-to-Sequence machine translation (MT) model from English to Dutch, where the hidden states for the encoder and decoder RNNs have a size of 4. We input the English sentence "Dragons eat apples too" into the MT model, and below are the values of the encoder hidden states we get from the model.

Name	Input Word	Hidden State
s1	Dragons	[0.8, 0.2, 0.2, 0.1]
s2	eat	[0.1, 0.7, 0.3, 0.2]
s3	apple	[0.0, 0.5, 0.4, 0.4]
s4	too	[0.2, 0.2, 0.0, 0.9]

Suppose the first word that the MT model generates is "Draken". And the decoder hidden state value for the word is h1 = [0.6, 0.2,0.3, 0.1]. Calculate the dot-product attention scores, attention weights, and the final attention output for the word "Draken". Recall that the definition of dot-product attention score is $E^t=[h_t^T s_1,....., h_t^T s_{N]}$

$e1=[0.6,0.2,0.3,0.1] \cdot [0.8,0.2,0.2,0.1]=0.6*0.8+0.2*0.2+0.3*0.2+0.1*0.1=0.59$

$e2=[0.6,0.2,0.3,0.1]\cdot[0.1,0.7,0.3,0.2]=0.6*0.1+0.2*0.7+0.3*0.3+0.1*0.2=0.31$

$e3=[0.6,0.2,0.3,0.1]\cdot[0.0,0.5,0.4,0.4]=0.6*0.0+0.2*0.5+0.3*0.4+0.1*0.4=0.26$

$e4=[0.6,0.2,0.3,0.1]\cdot[0.2,0.2,0.0,0.9]=0.6*0.2+0.2*0.2+0.3*0.0+0.1*0.9=0.25$

Attention Output = 0.313×[0.8,0.2,0.2,0.1] + 0.237×[0.1,0.7,0.3,0.2]+ 0.226×[0.0,0.5,0.4,0.4] + 0.222×[0.2,0.2,0.0,0.9]

Attention Output=[0.319,0.3851,0.224,0.3699]

b) Consider a single hidden layer RNN-based English-to-Spanish language translation model where Luong attention mechanism is employed. The maximum sentence length in the corpus is Le (Encoder length) and Ld (decoder length) respectively. The number of nodes in the encoder and decoder are Ne and Nd respectively. Source and destination vocabulary sizes are Vs and Vd respectively.

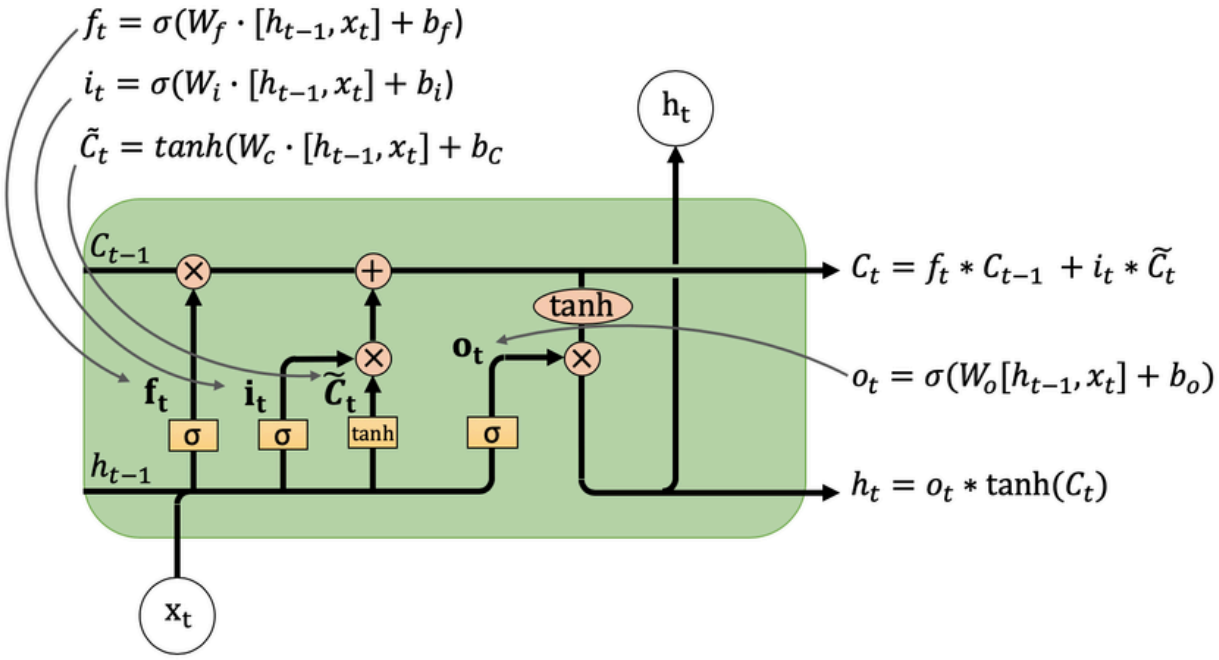
i) How many normalized attention weights wi(t) will be there? Le

ii) Assume raw attentional weights score(h_t, s_k)= $h_t^T Ws_k$ where s and h denote the hidden states in the encoder and decoder respectively. How many trainable parameters are there in the entire network?

N parameters =Ne×Ne + Ne +Nd ×(Nd+Vs)+ Ne×Nd+Nd + Vd ×Nd+Vd

Encoder + Decoder + Attention + Output

c) Write all the equations of LSTM Cell. The diagram is given for your reference.



CLO # 2: Students should be able to understand supervised and unsupervised methods of Deep Learning

Q4: **[10+5+5=20 marks]**

a) You are building a Transformer model to assist in medical diagnosis by analyzing patient symptoms and suggesting potential illnesses. The model takes as input a sequence of symptoms described by the patient and generates a list of possible illnesses. Consider the following scenario:

Input symptoms: "Fever, headache, fatigue"
Output diagnosis: "Common cold"
Given the embedding matrix for the symptoms and illnesses as follows:
"Fever": [0.2, 0.3, 0.4]
"headache": [0.5, 0.6, 0.7]
"fatigue": [0.8, 0.9, 1.0]
"Common cold": [0.3, 0.1, 0.5]
w_key = [[2, 0, 1], [1, 0, 3], [1, 1, 2]]
w_query = [[1, 0, 1], [1, 4, 2], [0, 1, 1]]
w_value = [[5, 2, 1], [2, 3, 0], [1, 1, 4]]
dk=2

1. Calculate masked self-attention for the above symptoms using the provided weight matrices. Show all steps clearly.
2. Calculate cross-attention for the above symptoms and output diagnosis using the provided weight matrices. Show all steps clearly.

Q=[0.5, 1.6, 1],[1.1,3.1, 1.9][1.7,4.6,2.8]]
K=[1.1, 0.4, 1.9],[2.3, 0.7, 3.7][3.5, 1, 5.5]]

V=[2,1.7, 1.8],[4.4,3.5,3.3][6.8,5.3,4.8]]
QKt=[3.09, 5.97, 8.85],[6.06, 11.73, 17.4],[9.03, 17.49, 25.95]]

b) Write short answers to the following questions:

1. In terms of the ViT (Vision Transformer) paper “An image is worth 16*16 words", what are sentences and words?
   "sentences" refer to image patches and "words" refer to tokens generated by dividing the patches into smaller embeddings.
2. What is the major difference between ViT and GPT? Though they both belong to the Transformers family. ViT= Encoder GPT = Decoder
3. Explain the major difference between GPT1, GPT2, and GPT3. Also, mention the number of trainable parameters in all versions.
   The major difference between GPT1, GPT2, and GPT3 lies in their size and scale, with GPT3 being the largest and most powerful, containing 175 billion parameters, compared to GPT2's 1.5 billion and GPT1's 110 million parameters.
4. What steps are required to convert GPT to ChatGPT?
   Supervised Fine-tuning, Reward Model, Proximal Policy Optimization
5. What is the difference between cross-attention and self-attention?
   Cross-attention involves attending to different input sequences, such as between encoder and decoder in sequence-to-sequence models, while self-attention attends to different positions within the same input sequence, capturing relationships between elements within the sequence itself.

c) Draw the architectural diagram of any BERT, DistilBert, BART, LLAVA, DETR, SegFormer, VideoMAE, or Whisper.
Many solutions exist

CLO # 2: Students should be able to understand supervised and unsupervised methods of Deep Learning

Q5: **[2.5*4 = 10 marks]**

Consider a GAN where, during training, the generator G manages to deceive the discriminator D completely such that $D(G(z))=0.5$ for all z.

1. Whether this indicate an optimal solution for the GAN? Justify your answer.
   No. It implies that the discriminator is unable to distinguish between real and fake samples, but it doesn't guarantee that the generated samples are of high quality or resemble the true data distribution.
2. What could be the potential pitfalls in this scenario?
   Mode collapse, Lack of diversity, Stagnation

Assume that after a few iterations of training a GAN, the discriminator D perfectly distinguishes real and fake samples, i.e., $D(x)=1$ for real samples x and $D(G(z))=0$ for generated samples $G(z)$.

3. Explain why this situation is problematic for the generator G?
   In this case, the generator fails to produce samples that resemble real data, and it receives no useful feedback from the discriminator to improve its performance.

4. What changes in the loss functions or training process would help the generator improve in this scenario?
   Use alternative loss functions like Wasserstein loss or hinge loss, which provide stronger gradients and can help mitigate mode collapse.

CLO # 1: Student should be able to describe what Deep Learning is and the skill sets needed for Deep Learning

Q6: **[1*5 +5=10 marks]**

a)Write short answers to the following questions:

1. What is the difference between PCA and autoconders?
   While both PCA and autoencoders perform dimensionality reduction, PCA is based on linear transformations and does not involve learning, whereas autoencoders are neural networks that learn to encode and decode data, allowing them to capture nonlinear relationships in the data.
2. What is meant by Latent Space Representation?
   Sol: The latent space is simply a representation of compressed data in which similar data points are closer together in space. For example, code is the latent space representation.
3. List main hyperparameters in autoencoders.
   Sol: Code Size, Number of Layers, Loss Function, Number of nodes per layer etc
4. Give two main applications of autoencoders
   • Anomaly detection
   • Data denoising (ex. images, audio)
   • Image inpainting
   • Information retrieval
5. What are Variational AutoEncoders?
   Variational AutoEncoders (VAEs) are generative models that use neural networks and variational inference to map data to a latent space, enabling the generation of new, similar data points by sampling from this space. They consist of an encoder, mapping data to a latent distribution, and a decoder, reconstructing data from samples drawn from this distribution. VAEs are adept at generating data that follows the distribution of the training data.

b) For the classical MNIST digit dataset complete the below code for AutoEncoders.

```
input_size = 784
hidden_size = 128
code_size = 32
input_img = Input(shape=(input_size,))
hidden_1 = Dense(hidden_size, activation='relu')(input_img)
code = Dense(code_size, activation='relu')(hidden_1)
hidden_2 = Dense(hidden_size, activation='relu')(code)
output_img = Dense(input_size, activation='sigmoid')(hidden_2)
autoencoder = Model(input_img, output_img)
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
autoencoder.fit(x_train,x_train, epochs=5)
```

CLO # 3: Students should be able to apply most important deep learning methods, using open-source tools

Q7: **[10 marks]**

You are part of a research team working on a project to classify satellite images into different land cover categories. The dataset consists of high-resolution satellite images captured from various geographic locations around the world. The goal is to classify these images into five categories: Urban, Forest, Water, Agriculture, and Desert.

Your team has encountered several challenges:

1. The dataset is highly imbalanced, with significantly more images of Urban areas than Desert areas.
2. The images have varying resolutions and quality due to different satellite sensors.
3. The presence of seasonal variations and weather conditions in the images makes classification more challenging.
4. Some categories have subtle distinctions that require the model to capture fine-grained details.

Which combination of architectures from the deep learning course would you use to solve this problem? Create an end-to-end architecture diagram. Your proposed architecture should address all the mentioned issues, and you should explain how each component of your solution specifically resolves each issue.

Note: many solutions exist

Sample Solution:

Input Image: High-resolution satellite images are fed into the pipeline.

Data Preprocessing: Normalize the images and resize them to a uniform size compatible with the CNN models.

Data Augmentation: Random Cropping, Horizontal/Vertical Flipping, Rotation, Scaling, Color Jittering: These techniques help increase the diversity of the dataset and make the model robust to variations in image quality and resolution.

CNN Models (CNN 1, CNN 2, CNN 3):

CNN 1 (e.g., ResNet): Known for handling varying resolutions and capturing detailed features.

CNN 2 (e.g., VGG): Effective for fine-grained feature extraction.

CNN 3 (e.g., EfficientNet): Balances model complexity and performance, optimized for different image resolutions.

Feature Extractors: Extract features from each CNN model. These features capture different aspects of the images due to the varied architectures of the CNN models.

Concatenate/Feature Fusion: Combine the features from the different CNN models to create a comprehensive feature representation. This step leverages the strengths of each model.

Temporal and Spatial Attention: Implement attention mechanisms to focus on important parts of the image and adjust for seasonal variations. This step ensures the model pays attention to

relevant spatial and temporal features, improving classification accuracy.
Classifier (MLP) with Imbalance Handling: A multi-layer perceptron (MLP) classifier that processes the fused features.
Output Classes: The final output layer classifies the images into one of the five categories: Urban, Forest, Water, Agriculture, Desert.

Imbalanced Dataset: Data augmentation and class imbalance handling in the classifier help balance the dataset and ensure fair training.

Varying Resolutions and Quality: Multiple CNN models, each designed to handle different image resolutions and quality variations, provide robust feature extraction.

Seasonal Variations and Weather Conditions: Temporal and spatial attention mechanisms allow the model to focus on relevant features despite seasonal changes and weather variations.

Subtle Distinctions Between Categories: The ensemble of different CNN models captures a wide range of features, and the feature fusion ensures that fine-grained details are considered during classification.