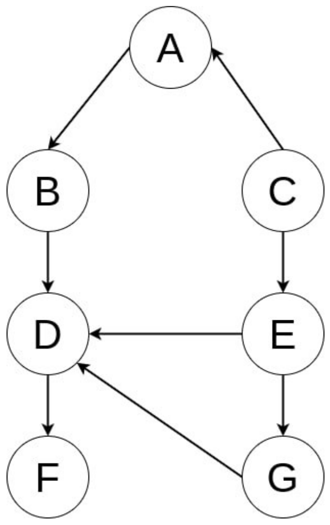
Assignment 1 — Introduction to AI

Note: Your solution for this assignment should have two parts—**a pdf document and code files**.

- Have a **single** pdf document that shows your solution for different questions (show either numerical values if the question asks for it, and/or theoretical justification as required). Include in this pdf, the code you wrote for the solution for the respective question (if coding is required).
- Upload your real code files that you used to solve the particular question. Make sure your code is neatly organized per question, runs correctly, and has comments that highlight the part you implemented so that I can easily understand it
- Combine your solution pdf and code files in a single zip folder and upload it on the eLearn assignment folder
- Solution should be typeset using a professional software (word, keynote, latex etc). Figures should also be made using software such as power point. **No handwritten solutions are allowed and will not be graded.**

Question 1 [10 points]:

Consider the following Bayesian network:



Write True/False for the following conditional independence statements. Justify clearly your answer by showing active/blocked trails as necessary and appropriate rules for them to be active/blocked. [No coding required for this question. Each sub-question has **2 points**]

1. A ⊥ G |{F}
2. A ⊥ G | {E}

3. $A \perp E \mid \{C\}$

4. $A \perp E \mid \{C,D\}$

5. $A \perp D \mid \{B,E\}$

Question 2 [10 points]

The Ministry of Health purchased COVID-19 tests for distribution to their population. They are produced by three different manufacturers: A, B and C. Each citizen will receive two tests at home and the manufacturer of each test is chosen independently and identically distributed. Therefore, a citizen may receive two tests from the same manufacturer or from two different ones. Each manufacturer is equally likely to be chosen (P=1/3).

The tests' ability to provide accurate results fluctuates. In other words, they have different chances of producing *true positive* (TP) and *true negative* (TN) results. A True positive occurs when a test is positive and the patient has COVID19. True negatives, on the other hand, are tests in which the result is negative and the patient does not have the illness. The manufacturers describe their tests in the following way:

*A: P(TP) = 0.7 and P(TN) = 0.99
*B: P(TP) = 0.8 and P(TN) = 0.95
*C: P(TP) = 0.9 and P(TN) = 0.91

Assume that before taking the tests, the prior probability of having covid is 0.3.
A person can self-test either using one test kit or both the test kits.

(i) Draw the Bayes net corresponding to this setup. Explain what each random variable of this Bayes represents, and show the domain of each random variable **[3 points]**
Hint: There should be 5 random variables. You can use a table like below to describe random variables and their interpretation:

Variable Name	Domain (Set of Values)	Interpretation *(intuitive explanation of what the variable represents)*

(ii) Write conditional probabilities (numerical values) associated with each node of this Bayes net. As there are 5 variables, please specify one conditional probability table (CPT) for each variable **[2 points]**

(iii) Are the results of the two tests dependent or independent given the evidence that the Covid Status is known? Justify your answer. **[1 point]**

(iv) Assume you took both tests at home. After being tested twice in a matter of minutes, the first test was positive and the second negative. What is the probability that you actually have COVID19? Show your analytical computations (without using pgmpy toolbox) **[4 points]**

Question 3 [10 points]

In this question below, we will construct a small Bayesian network in the pgmpy toolbox. This network models the relationship between *yellow fingers (Y)*, *smoking (S)*, *cancer (C)*, *skin burn (B)*, *radiation (R)*, *solar flares (F)*, *Weakened Immune System (I)*, abusing *alcohol (A),* and *using a cellphone (P)*. In this model, smoking can cause yellow fingers and cancer. Solar flares or using a cellphone can cause radiation. Radiation can cause cancer and skin burn. Alcohol abuse can both cause cancer or weaken the immune system, and a weakened immune system can cause cancer.

Consider "0" represents a variable being a false, "1" represents a variable being true. Each variable in this problem is binary, i.e. can only take two values (0 or 1).

The prior probability of smoking P(S=1) is 0.1. The prior probability of solar flares P(F=1) is 0.001. The prior probability of using a cellphone P(P=1) is 0.999. The prior probability of alcohol abuse P(A) is 0.1.

Conditional probabilities for skin burn (B) are given below:

R	P(B=1|R)
0	0.02
1	0.2

Conditional probabilities for Weakened Immune System (I) are given below:

A	P(I=1|A)
0	0.05
1	0.3

Conditional probabilities for radiation (R) are given below:

F	P	P(R=1|F,P)
0	0	0.01
0	1	0.05
1	0	0.2
1	1	0.3

Conditional probabilities table for cancer C are given as:

S	A	I	R	P(C=1|A,I,R,S)
0	0	0	0	0.1
0	0	0	1	0.5
0	0	1	0	0.2
0	0	1	1	0.6
0	1	0	0	0.3
0	1	0	1	0.6

0	1	1	0	0.4
0	1	1	1	0.8
1	0	0	0	0.2
1	0	0	1	0.6
1	0	1	0	0.3
1	0	1	1	0.7
1	1	0	0	0.4
1	1	0	1	0.7
1	1	1	0	0.5
1	1	1	1	0.9

The conditional probability table for yellow fingers (Y) is given as:

S	P(Y=1|S)
0	0.04
1	0.95

Implement the above Bayes net with the specified conditional probabilities into pgmpy (Install it from http://pgmpy.org/).

- **Show screenshot of your code denoting the implementation in pgmpy and attach as part of your solution pdf.** [**2.5 points**]
  Should show screenshot of the code. Otherwise -1 point.

Answer the following questions. You can use functions implemented in pgmpy to compute the numerical answers as required for some of the below questions as appropriate. Also show snippets of your code used in solution pdf. [Each sub-part has **1.5 points**]

1. Draw the Bayesian network clearly showing the nodes and arrows showing relationship among all the variables (you can use drawing tools in PowerPoint or Keynote)
2. What is the probability of radiation given Cancer = 1 (show values for R=0, 1)?

3. What is the probability of cancer given the patient has skin burn, yellow fingers and abuses alcohol (show values for C=0, 1)

4. Are Smoking and skin burn independent given that cancer is present? Justify your answer.

5. What is the probability of cancer (=1) if you never abused alcohol or used a cellphone?

Question 4 [10 points]

In this question, we will use an ANN classifier to classify the handwritten digits (from the MNIST dataset). In a slight twist, to check the robustness of ANN-based methods, we corrupt MNIST images via adding some noise. We will use the "**mnist_corrupted/zigzag**" dataset available via tfds (https://www.tensorflow.org/datasets/catalog/mnist_corrupted).

You can build your solution on top of the python notebook covered in class to classify the standard MNIST dataset. The solution should perform the following tasks:

a) Load the **mnist_corrupted/zigzag** dataset via tfds. Load both training and testing datasets separately. **[2 points]** (refer to this page to learn how to load data from tfds: https://www.tensorflow.org/datasets/overview#as_numpy_tfdsas_numpy)
b) Create an ANN with 1 input layer and **at least** one hidden layer, with at least 2 nodes per layer. You can use relu or any other activation function for hidden layers. You are free to create additional hidden layers or increase the number of nodes to maximize the final accuracy. You are encouraged to systematically explore the different hyper-parameter configurations. Your objective is to get an overall accuracy of 90% or more on the testing dataset. **[2 points]**
c) Create 1 output layer. What is the size of output layer? What should be the activation function for the output layer? **[1 points]**
d) Compile and train the neural network with the appropriate loss function (what should be the loss function type?) **[2 points]**
e) Plot the training loss (as given by the Keras API) on the y-axis against the epoch number **[1 point]** (refer to this page to see how to extract the history: https://www.tensorflow.org/guide/keras/train_and_evaluate)
f) What is the final accuracy for different classes (the proportion of correctly classified instances of each class) and overall accuracy on the testing data? **[1 point]**
g) For total accuracy (on testing data):
    1. >=90% **[0.5 point],**
    2. >=95% **[additional 0.5 points]**

For this question, please fill up your code using the **Q4_template_MNISTCorrupted.ipynb** file. Also, please limit yourselves to using standard dense ANNs (rather than using advanced CNNs).

Remark: in this introductory course, we only split the data into two sets (training and testing), without worrying about any overfitting the hyperparameters (such as the number of nodes, etc.). A more rigorous approach would require the use of a "validation set", but we leave this to more advanced courses. Please do not use a validation set for this introductory assignment.

Question 5 [10 points]

In this question, we will learn how to use transfer learning in the context of CNNs. More hints to solve this question are available at: https://www.tensorflow.org/tutorials/images/transfer_learning . In particular,

- You will first create and train a CNN using the MNIST dataset. The CNN we use is the LeNet-5 which was one of the first successful CNN proposed by Yann LeCun (https://en.wikipedia.org/wiki/LeNet)
- You will then use this trained CNN, and adapt it to the **binary_alpha_digits** dataset (https://www.tensorflow.org/datasets/catalog/binary_alpha_digits).

We have provided you a code skeleton (Q5_vision_lenet.ipynb) to help you finish this problem. You are free to change this file as required.

Please perform the following tasks in your solution:

a) Create the LeNet-5 CNN architecture using Keras API (see code skeleton for the number and types of layers to create). Train the model on the MNIST dataset. **[3 points]**
b) What is the accuracy of your trained LeNet-5 model on the MNIST training dataset? Try to get an accuracy above 90%. **[0.5 points]**
c) Download the **binary_alpha_digits** dataset using tfds, and split the dataset into 20% testing data and 80% training data. **[1 point]**
d) As the dimension of images in the binary_alpha_digits are different from the image size in MNIST dataset, upscale images in binary_alpha_digits to match the image size in MNIST dataset using OpenCV. This is required as we would like to use the LeNet trained using the MNIST dataset for binary_alpha_digits. **[2 points]**
e) Remove the final output layer of LeNet you have trained on MNIST (to do this, please check the flag "include_top" in Keras and the tensorflow link for transfer learning noted earlier) **[0.5 point]**
f) After removing the final output layer, extend your trained LeNet model by adding at least one hidden layer (dense, convolution, max pooling or any other type of layer). Also attach one final output layer. In this part, you are free to explore and decide how many hidden layers to add, their type, the number of nodes in each layer and the activation function yourself. Keep in mind, the *output layer* must have the appropriate number of nodes and activation function that matches the given task. **[1.5 points]**
g) Train the model and show accuracy on the testing dataset (of **binary_alpha_digits)**. You can either fix all the weights of your MNIST-trained LeNet model and train only the layers you have added, or train the whole network again. Choose the setting that gives you higher accuracy given the computational resources. Check link https://keras.io/getting-started/faq/#how-can-i-freeze-keras-layers.
Try to **achieve** a testing data accuracy of **50% or more** (you can report the **best** over multiple **run**s). Please make sure that in your submitted jupypter notebook, **logs** show your best run. **Note:** some variation between runs is expected, with *the true accuracy*

being somewhere in between. You are **not** required to *reliably* get 50 percent accuracy over *all runs*, but try to demonstrate from the log files that one run achieved 50 percent. **[1.5 points]**

For each part (a)-(g), **copy and paste the code snippets (as applicable)** in your solution pdf that shows how you're doing the particular tasks. **Do not** copy and paste your complete code (only the *relevant* parts for each question).

Please use tensorflow and Keras libraries for this question (rather than pytorch or any other deep learning packages) as our testing setup only supports these environments. You may also have to install the opencv package for doing some image processing for this question (opencv should already be installed if you use Google Collab).

Programming hints: Make sure to save your network's parameters while training every once in a while (search online to find out how you can do it in Keras!). That way if your program crashes, you don't have to re-train from scratch. How many epochs are sufficient for convergence? You may need to enter some threshold condition so that training stops when that condition is satisfied. It is not good (or necessary) to train your network until the loss is zero (that may never happen). Furthermore, this may overfit the training dataset, and may lead to poor accuracy on the testing dataset.

Useful pointers for running on Colab:

To use GPU for training, you will need to enable GPUs for the notebook (*Runtime->Change runtime type->Hardware accelerator->GPU*). Please change the runtime type to CPU when you are not using GPU resources as GPUs will be prioritized for users who have recently used fewer resources.

Remark: in this introductory course, we only split the data into two sets (training and testing), without worrying about any overfitting the hyperparameters (such as the number of nodes, etc.). A more rigorous approach would require the use of a "validation set", but we leave this to more advanced courses. Please do not use a validation set for this introductory assignment.