



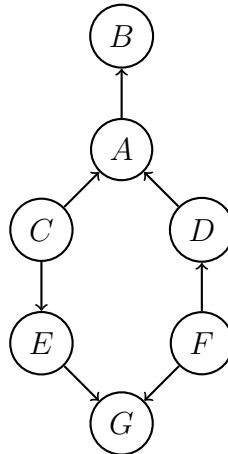
Introduction to AI Assignment 1

Vdovenko Anton

February 2021

Question 1

Consider the following Bayesian network:



Write True/False for the following conditional independence statements. Justify clearly your answer. For example, show the active/blocked trails as necessary and the reason for them to be active or blocked (e.g., use rules such as cascade, common cause or v-structure).

We need to look through all trails either they are blocked or not given 3 above rules. If all trails are blocked we can conclude independence.

1) $C \perp F \mid \{E, G\}$

- Trail $C \rightarrow A \leftarrow D \leftarrow F$ is **blocked** based on **v-structure** rule for (C, A, D) , because neither A nor its descendants are given.
- Trail $C \rightarrow E \rightarrow G \leftarrow F$ is **active** based on **v-structure** rule for (E, G, F) , because G is given.

Answer: **FALSE**

2) $C \perp F \mid \{D, G\}$

- Trail $C \rightarrow A \leftarrow D \leftarrow F$ is **blocked** based on **cascade** rule for (A, D, F) , where D is given and **v-structure** rule for (C, A, D) , because neither A nor its descendants are given.
- Trail $C \rightarrow E \rightarrow G \leftarrow F$ is **active** based on **v-structure** rule for (E, G, F) , because G is given.

Answer: **FALSE**

3) $E \perp D \mid B$

- Trail $\textcircled{E} \leftarrow \textcircled{C} \rightarrow \textcircled{A} \leftarrow \textcircled{D}$ is **active** based on **v-structure** rule for (C,A,D), because B which is a descendant of A is given.
- Trail $\textcircled{E} \rightarrow \textcircled{G} \leftarrow \textcircled{F} \rightarrow \textcircled{D}$ is **blocked** based on **v-structure** rule for (E,G,F), because nor G or its descendants are given.

Answer: **FALSE**

4) $E \perp D \mid \{\}$

- Trail $\textcircled{E} \leftarrow \textcircled{C} \rightarrow \textcircled{A} \leftarrow \textcircled{D}$ is **blocked** based on **v-structure** rule for (C,A,D), because nor A or its descendants are given.
- Trail $\textcircled{E} \rightarrow \textcircled{G} \leftarrow \textcircled{F} \rightarrow \textcircled{D}$ is **blocked** based on **v-structure** rule for (E,G,F), because nor G or its descendants are given.

Answer: **TRUE**

5) $A \perp F \mid \{C, D\}$

- Trail $\textcircled{A} \leftarrow \textcircled{D} \leftarrow \textcircled{F}$ is **blocked** based on **cascade** rule for (A,D,F) given D.
- Trail $\textcircled{A} \leftarrow \textcircled{C} \rightarrow \textcircled{E} \rightarrow \textcircled{G} \leftarrow \textcircled{F}$ is **blocked** based on **v-structure** rule for (E,G,F), because nor G or its descendants are given and **comon cause** rule for (A,C,E) given C.

Answer: **TRUE**

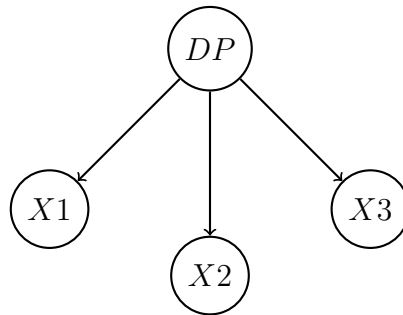
Question 2

A box contains three different dices. Each dice has 4 faces with numbers 1 through 4 written over them. The properties of the dices are as follows:

- Dice 1 (say D1) is a fair dice with each face equally likely to come.
- Dice 2 (say D2) is a biased dice with outcome 1 likely with probability 0.5. The rest of the outcomes have equal probability (i.e., $P(D2=2) = P(D2=3) = P(D2=4)$).
- Dice 3 (say D3) is also a biased dice with outcome 2 likely with probability 0.5. The rest of the outcomes have equal probability of coming up.

We pick one dice randomly (that is, each of the dice in the box has equal chance of being picked up), and then that dice is rolled three times to generate outcomes X1, X2 and X3.

1) 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.



Variable Name	Domain	Interpretation
DP	{D1,D2,D3}	Pick of one of three given dice
X1	{1,2,3,4}	First throw of dice
X2	{1,2,3,4}	Second throw of dice
X3	{1,2,3,4}	Third throw of dice

2) Write conditional probabilities (numerical values) associated with each node of the above Bayes net.

DP			X1 D	1	2	3	4
Dice 1	Dice 2	Dice 3	Dice 1	1/4	1/4	1/4	1/4
1/3	1/3	1/3	Dice 2	1/2	1/6	1/6	1/6
			Dice 3	1/6	1/2	1/6	1/6

X2 D	1	2	3	4	X3 D	1	2	3	4
Dice 1	1/4	1/4	1/4	1/4	Dice 1	1/4	1/4	1/4	1/4
Dice 2	1/2	1/6	1/6	1/6	Dice 2	1/2	1/6	1/6	1/6
Dice 3	1/6	1/2	1/6	1/6	Dice 3	1/6	1/2	1/6	1/6

3) Assume that the observed outcome was $X_1=1$, $X_2=3$, $X_3=2$. Calculate which dice(D1 or D2 or D3) was most likely to have been picked from the box. Show numerical calculations(based on conditional probabilities) to justify your answer.

Joint distribution table. (Same for all X):

X	1	2	3	4
Dice 1	1/12	1/12	1/12	1/12
Dice 2	1/6	1/18	1/18	1/18
Dice 3	1/18	1/6	1/18	1/18

- $Pr(D1 | X_1 = 1, X_2 = 3, X_3 = 2) = 1/12 * 1/12 * 1/12 = 1/1728$
- $Pr(D2 | X_1 = 1, X_2 = 3, X_3 = 2) = 1/6 * 1/18 * 1/18 = 1/1944$
- $Pr(D3 | X_1 = 1, X_2 = 3, X_3 = 2) = 1/18 * 1/18 * 1/6 = 1/1944$

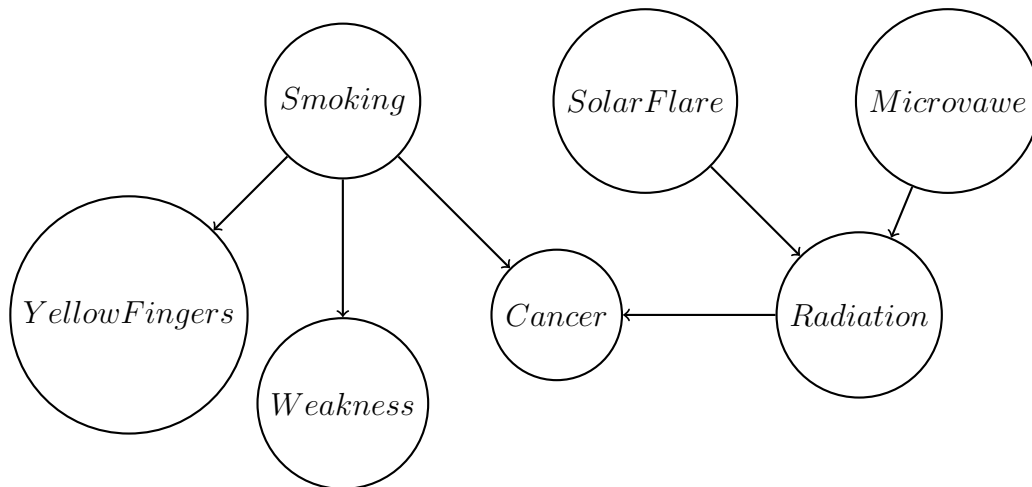
Now to get final answer we need to scale above probabilities to 1.

- $D1 = 1/1728 * (1/(1/1728 + 1/1944 + 1/1944)) = 0.36$
- $D2 = 1/1944 * (1/(1/1728 + 1/1944 + 1/1944)) = 0.32$
- $D3 = 1/1944 * (1/(1/1728 + 1/1944 + 1/1944)) = 0.32$

Answer: **$Pr(\text{Dice 1}) = 0.36$** is most probable one given $X_1=1$, $X_2=3$, $X_3=2$ specified sequence of outcomes.

Question 3

1) Draw the Bayesian network clearly showing the nodes and arrows showing relationship among all the variables.



Bayesgraph using pgmpy:

```
# Prior probability for Smoking P(S)
cpd_S = TabularCPD(variable='Smoking', variable_card=2, values=[[0.85],
                                                                [0.15]])

# Prior probability for Solar_Flare P(F)
cpd_F = TabularCPD(variable='Solar_Flare', variable_card=2, values=[[0.
99], [0.01]])

# Prior probability for Microvawe P(M)
cpd_M = TabularCPD(variable='Microvawe', variable_card=2, values=[[0.05]
                                                                , [0.95]])

# Conditional probability for Weakness or P(W| S)
cpd_W = TabularCPD(variable='Weakness',variable_card=2, values = [[0.8,0
.1],[0.2,0.9]],
                    evidence = ['Smoking'],
                    evidence_card=[2])

# Conditional probability for Radiation or P(R|F,M)
cpd_R = TabularCPD(variable='Radiation', variable_card=2, values = [[0.9
, 0.8, 0.8, 0.1],
[0.1, 0.2, 0.2, 0.9]],
                    evidence = ['Solar_Flare',"Microvawe"],
                    evidence_card=[2,2])

# Conditional probability for Cancer or P(C|S,R)
cpd_C = TabularCPD(variable='Cancer', variable_card=2, values = [[0.9, 0
.4, 0.7, 0.1],
[0.1, 0.6, 0.3, 0.9]],
                    evidence = ['Smoking',"Radiation"],
                    evidence_card=[2,2])

# Conditional probability for Yellow_Fingers or P(Y|S)
cpd_Y = TabularCPD(variable='Yellow_Fingers', variable_card=2, values =
[[0.89, 0.2],
[0.11, 0.8]],
                    evidence = ['Smoking'],
                    evidence_card=[2])
```

2) What is the probability of cancer given weakness = 1?

```
# Compute probability of Cancer given only Weakness
phi_query = infer.query(['Cancer'], evidence={'Weakness':1}, joint =
                        False)

factor = phi_query['Cancer']
print('Probability of Cancer given Weakness')
print(factor)
```

Answer: $\Pr(C = 1 \mid W = 1) = 0.2983$

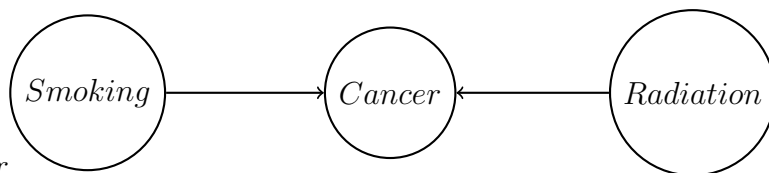
3) What is the probability of smoking given cancer=1?

```
# Compute probability of Smoking given only Cancer
phi_query = infer.query(['Smoking'], evidence={'Cancer':1}, joint =
                        False)

factor = phi_query['Smoking']
print('Probability of Smoking given Cancer')
print(factor)
```

Answer: $\Pr(S = 1 \mid C = 1) = 0.27$

4) Are Smoking and Radiation independent given cancer? Justify your answer.



Smoking \perp Radiation \mid Cancer

Answer: The trail is **active** due to **v-structure** rule and hence Smoking and Radiation given Cancer are **not independant**.

5) What is the probability of cancer if you never use a microwave = 0?

```
# Compute probability of Cancer given no Microwave
phi_query = infer.query(['Cancer'], evidence={'Microwave':0}, joint =
                        False)

factor = phi_query['Cancer']
print('Probability of Cancer given no Microwave')
print(factor)
```

Answer: $\Pr(C = 1 \mid M = 0) = 0.182$

Question 4

In this question it is required to create an artificial neural based classifier to classify the fashion mnistdataset. The solution you provide should preform following tasks:

a) Download the fashion mnistdataset

```
mnist = tf.keras.datasets.fashion_mnist
# Load training data, labels; and testing data and their true labels
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

b,c) Create and ANN with 1 input layer, at least two hidden layer with at least 10 nodes per layer. Create 1 output layer.

```
# b) ANN
model = tf.keras.Sequential()
model.add(layers.Flatten(input_shape=(28, 28)))
model.add(layers.Dense(256, activation=tf.nn.relu))
model.add(layers.Dense(256, activation=tf.nn.relu))
model.add(layers.Dense(256, activation=tf.nn.relu))
model.add(layers.Dense(128, activation=tf.nn.relu))
model.add(layers.Dense(128, activation=tf.nn.relu))
model.add(layers.Dense(128, activation=tf.nn.relu))
model.add(layers.Dense(64, activation=tf.nn.relu))
model.add(layers.Dense(32, activation=tf.nn.relu))
# c) Output layer
model.add(layers.Dense(10, activation=tf.nn.softmax))
```

The size of output layer is equal to number of classes for classification, which is 10. Because the classification is not binary we need to use softmax function as activation fuction to make prediction at final layer.

d) Compile and train the neural network with the appropriate loss fuction.

```
# Compile
model.compile(optimizer="Adam",
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
# Train
epochs = 50
hist = model.fit(train_images, train_labels,
                validation_split = 0.4, epochs=epochs,
                callbacks=[early_stopping_monitor])
```

We have more than 2 classes to predict, hence we use sparce categorical crossentropy to compute the crossentropy loss between the labels and predictions.

e) Final accuracy on the testing data.

```
test_loss, test_acc = model.evaluate(test_images, test_labels)
print('Test accuracy:', round(test_acc,4))
```

Test accuracy: 0.8819.

Question 5

You will need to use a pre-trained network (MobileNetV2) to do image classification on the CIFAR-10 dataset. Your solution must do the below tasks:

a,b,c) Download the pre-trained MobileNetV2 network from Keras Applications. Remove the final output layer of the downloaded network. After removing the final output layer, extend the MobileNetV2 model by adding at least one hidden layer (dense, convolution or any other type of layer). Also attach one final output layer.

```
# Download the pre-trained model using keras applications API
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2

model = tf.keras.Sequential()
mob = MobileNetV2(input_shape = (96,96,3), include_top=False)
model.add(mob)
# Extend the pre-trained model
model.add(layers.Conv2D(filters=64, kernel_size=(3,3), padding="same",
                        activation="relu"))
model.add(layers.Conv2D(filters=64, kernel_size=(3,3), padding="same",
                        activation="relu"))
model.add(layers.MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(layers.Conv2D(filters=128, kernel_size=(3,3), padding="same",
                        activation="relu"))
model.add(layers.Conv2D(filters=128, kernel_size=(3,3), padding="same",
                        activation="relu"))

model.add(layers.Flatten())
model.add(layers.Dense(units=128, activation="relu"))
model.add(layers.Dense(units=64, activation="relu"))
# Final output layer
model.add(layers.Dense(units=10, activation="softmax"))
```

1) Write how you extended the MobileNetV2 model (how many layers you added, what type of layers, how many nodes per layer, their activation function etc).

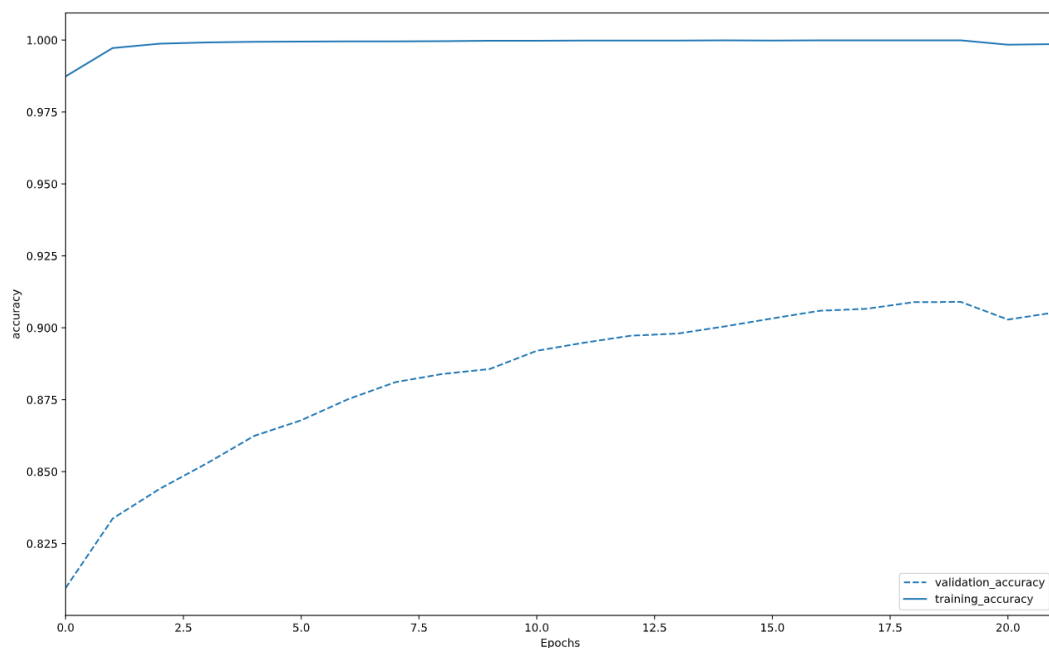
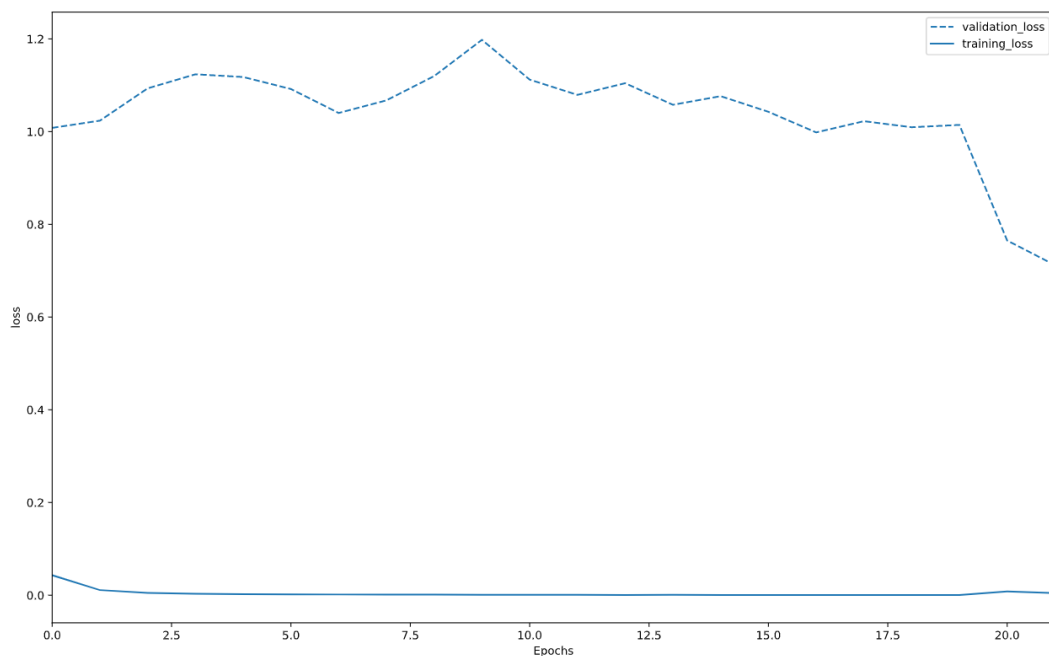
After MobileNetV2 I add 2 Conv2D layers with 64 filters 3*3 size following by MaxPool2D layer with pooling size 2*2 and strides(2,2), after that 2 more Conv2D layers with 128 filters same filter size as before following by Flatten layer which then outputs data to Dense layer with 128 neurons, which then outputs data to the next Dense layer with 64 neurons and at last output layer with 10 neurons one for each class. The activation function for all layers was relu, except for output layer where softmax function needs to be used and layers such as MaxPool2D and Flatten which require no activation function. I used VGG16 architecture to come up with idea how to extend MobileNetV2.

d) Add the appropriate loss function, compile and train the modified network.

```
# Compile
model.compile(optimizer="Adam",
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train
epochs = 200
patience = 10
hist_3 = model.fit(resized_train_images,
                  train_labels, validation_split = 0.4, epochs=epochs,
                  callbacks=[early_stopping_monitor, model_checkpoint],
                  batch_size = 512, shuffle = True)
```

2) Plot the loss function value with respect to the epoch number on the training data. How did you decide when to terminate training?



I used list of callbacks to save best model at the end of epoch using validation accuracy as metric and also for stopping the training if validation accuracy was not increasing due to specified patience variable (number of epochs with no improvement of metric).

```
# ModelCheckpoint
model_checkpoint=tf.keras.callbacks.ModelCheckpoint('
                                                    Ass_1_Question_5_model_4.h5',
                                                    save_weights_only=False, save_best_only = True)

# EarlyStopping
patience = 2
early_stopping_monitor = EarlyStopping(
    monitor='val_accuracy',
    min_delta=0,
    patience=patience,
    verbose=0,
    mode='auto',
    baseline=None,
    restore_best_weights=True
)
```

3) Show the accuracy of the trained classifier over the entire testing dataset.

```
test_loss, test_acc = model.evaluate(resized_test_images, test_labels)
print('Test accuracy:', test_acc)
```

Test accuracy: 0.9076