Assignment 1

Your Name email@uregina.ca
200XXYYZZ

This is an example of how to cite: Darwiche (2013) proposed a new approach for inference in Bayesian networks (BNs) (Pearl, 1988) based on partial differentiation called *Arithmetic Circuits* (Darwiche, 2013).

This template is organized as follows. In Section 1, ...

1. Example Section

A *joint probability distribution* (JPD) is function P on the Cartesian product V of the variable domains such that the following two conditions hold:

$$0 \leq P(v) \leq 1.0 \text{ for each configuration } v \in V; \text{ and} \tag{1}$$

$$\sum_{v \in V} P(v) = 1.0 \tag{2}$$

Referring to Equation 2 here.

Table 1: Table (a) provides the prior probability of variable A and Table (b) provides the conditional probability of B given A.

(a)

A	$P(A)$
1	0.3
0	0.7

(b)

A	B	$P(B\|A)$
1	1	0.1
1	0	0.9
0	1	0.8
0	0	0.2

Referring to Table 1 (a) here.

1.1 Example Subsection

This is how you use math items inline: $(\theta_a \star \lambda_a)$, $(\theta_{ab} \star \lambda_b)$, $(\theta_{a\bar{b}} \star \lambda_{\bar{b}})$, $(\lambda_b \star \theta_{\bar{a}b})$, $(\lambda_{\bar{b}} \star \theta_{\bar{a}\bar{b}})$, and $(\lambda_{\bar{a}} \star \theta_{\bar{a}})$.

Referring to Figure 1.

```
import numpy as np
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Layer, Activation
import time
import tensorflow as tf

f = open("results.csv", "w")


INPUT_SIZE = 10
OUTPUT_SIZE = INPUT_SIZE
nb_class = 3

batch_size = 128
nb_epoch = 40

np.random.seed(123)

X_train = np.random.rand(INPUT_SIZE, nb_class)
Y_train = np.random.rand(OUTPUT_SIZE, nb_class)

X_test = np.random.rand(INPUT_SIZE)
Y_test = np.random.rand(OUTPUT_SIZE)

for i in range(1,51):

    start_time = time.time()

    model = Sequential()
    model.add(Dense(INPUT_SIZE, input_shape=(nb_class,)))
    model.add(Activation('linear'))
    model.add(Dense(OUTPUT_SIZE))
    model.add(Activation('linear'))
    model.compile(loss='categorical_crossentropy', optimizer='rmsprop')

    final_time = time.time()
    diff_time = final_time - start_time

    f.write(str(i)+","+str(diff_time)+","+"\n")

f.close()
```

Table 2: Comparison of parallel and serial solutions for ACs with 50 runs each

Solution	Time average	Standard deviation
Serial	0.1610	0.0536
Parallel	0.0434	0.0082

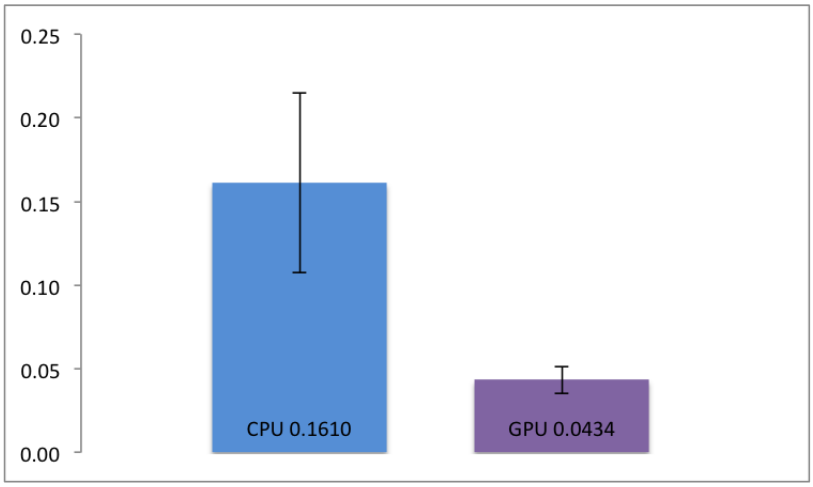

Figure 1: Comparison between CPU and GPU implementation of AC compiling.

2. Conclusion

In nec hendrerit arcu. Pellentesque leo libero, fringilla consectetur sapien eget, placerat finibus justo. Aliquam id sapien in eros lacinia euismod. Sed consectetur eros quis dui vestibulum pulvinar. Suspendisse ligula lacus, blandit interdum convallis sit amet, vulputate eget quam. Curabitur justo nunc, efficitur vitae fringilla eget, suscipit vitae sem. Etiam ultricies, nibh dictum convallis viverra, ipsum magna vehicula nibh, eu elementum purus est ac risus. Suspendisse sollicitudin auctor urna vel aliquet. In at eros et elit mollis lacinia. Fusce auctor leo id metus porttitor, vulputate semper erat congue. Donec rutrum erat non mauris convallis, id feugiat velit facilisis. Sed interdum magna sit amet mauris elementum pellentesque.

References

Adnan Darwiche. A differential approach to inference in bayesian networks. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI2000)*, 2013.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.