CS4215 Quantitative Performance Evaluation for Computing systems
Assignment 1

Marco, Schouten
m.schouten-4@student.tudelft.nl
Student Number: 5352908

September 24, 2021

1 Exercise 1, 2-way ANOVA

	DoF	Sum of Squares	Mean Square	F value	F critical	P
A	2	80	40.00	4.80	3.89	0.029
B	3	100	33.33	4,00	3.50	0.035
AB	6	40	6.67	0.8	3.00	0.588
Error	12	100	8.33	/	/	/
Total	23	320	/	/	/	/

1. First we find "a" and "b".

$$MS_A = SS_A/(a-1) \rightarrow a = (SS_A/MS_A) + 1 \rightarrow a = (80/40) + 1 = 3$$
$$MS_B = SS_B/(b-1) \rightarrow B = (SS_B/MS_B) + 1 \rightarrow b = (100/33.33) + 1 = 4$$

Then we know the DoF of A, B, and AB:

$$DoF(A) = a - 1 = 2$$
$$DoF(B) = b - 1 = 3$$
$$DoF(AB) = (a-1)(b-1) = 6$$

Now we find "r" using the ANOVA decomposition:

Remember $V_{Error} = ab(r-1)$:

$$\begin{aligned} V_{tot} &= V_A + V_B + V_{AB} + V_{Error} \\ 23 &= 2 + 3 + 6 + 12 * (r-1) \\ 23 &= 2 + 3 + 6 + 12r - 12 \\ 24 &= 12r \\ r &= 2 \end{aligned}$$

Therefore: Replicates = 2; DoF(Error) = 12, $MS_E = SS_E/DoF(Error) = 100/12 = 8.33$.

Now to Compute the F-Values and compare it with F-table.
F-value of A = $MS_A/MS_{error} = 40/8.33 = 4.80$
F-value of B = $MS_B/MS_{error} = 33.33/8.33 = 4.00$
F-value of AB = $MS_AB/MS_{error}6.67/8.33 = 0.8$

2. the F-computed statistics should be compared to the respective F-table values. We want the F statistics to fall outside the significance region (to reject the null hypothesis, hence proving the model is good).

   if $\forall factor : F_{computed} > F_{table}$ then they are significant. In our specific case, the interaction AB is not significant.

3. P-values are computed depending on F-computed and DoF. Note that p-value of AB is far off the requirement of being smaller than 0.05.

4. To fit a regression model, we can include the significant factors A and B (and excluding the insignificant factor AB to have a simpler model).

$$y_{ijk} = \mu + \alpha_j + \beta_i + \gamma_{ij} + e_{ijk} \tag{1}$$
$$\rightarrow y_{ijk} = \mu + \alpha_j + \beta_i + e_{ijk} \tag{2}$$

5. We can conclude that this experiment higher order interaction could be omitted. Null hypothesis is not rejected for AB.

2 Exercise 2, 2-way ANOVA, Random Forest

Since the p-values for Core and Estimator and their interaction and all less than 0.05, this means that every factor has a statistically significant effect on the computational training time for Random Forest classifier.

In statistical terms, the Null Hypothesis is rejected *"there's no difference between the training time for the different number of cores and different number of estimators"*. Hence the model is good.

	numeber of cores		
Number of Estimators	1	2	8
100	4.393	2.506	1.374
	4.653	2.537	1.236
	4.977	2.568	1.278
200	9.449	5.314	3.050
	9.561	5.330	2.628
	10.062	5.431	2.683
400	18.220	10.284	5.774
	17.534	10.651	4.627
	18.603	9.795	4.392
500	24.812	11.910	5.572
	23.472	11.896	5.329
	24.597	12.298	5.409

Table 1: Experimental Data Y

	sum sq	df	F	$PR(> F)$
core	686.64	2	2209.46	6.17e-28
estimator	672.68	3	1443.03	3.17e-27
core&estimator	217.49	6	233.28	4.64e-20
Residual	3.73	24	NaN	NaN

Table 2: 2-way ANOVA analysis

3 Exercise 3, fractional factorial design 2^{k-p}

1. Main Effects:

$$q_A = \frac{\sum y_i x_A}{n_{experiments}} = \frac{100 + 120 - 15 - 10 - 40 - 20 + 30 + 50}{8} = 26.875$$
$$q_B = \frac{\sum y_i x_B}{n_{experiments}} = \frac{-100 + 120 - 15 + 10 - 40 + 20 - 30 + 50}{8} = 1.8750$$
$$q_C = \frac{\sum y_i x_C}{n_{experiments}} = \frac{-100 - 120 + 15 + 10 - 40 - 20 + 30 + 50}{8} = -21.8750$$
$$q_D = \frac{\sum y_i x_D}{n_{experiments}} = \frac{-100 - 120 - 15 - 10 + 40 + 20 + 30 + 50}{8} = -13.1250$$

2. Percentage of Variation Explained:
   Now we compute SSA, SSB,SSC,SSD, SST. using the formula
   $k = 3, 2^k = 8, r = 1$

   $q_0 = (100 + 120 + 15 + 10 + 40 + 20 + 30 + 50)/8 = 385/8 = 48.1250$

$SSO = 8 * q_0^2 = 8 * 2316.0 = 18528$
$SS_Y = \sum_i y_i^2 = 100^2 + 120^2 + 15^2 + 10^2 + 40^2 + 20^2 + 30^2 + 50^2 = 30125$
$SS_T = SSY - SS0 = 30125 - 18528 = 11597$

$SS_j = 2^k r q_j^2 = 8 * q_j^2$
$SS_A = 5778.1248$
$SS_B = 28.125$
$SS_C = 3828.125$
$SS_D = 1378.125$

	% Variation
A	$\frac{SSA}{SST} = 49.82\%$
B	$\frac{SSB}{SST} = 0.24\%$
C	$\frac{SSC}{SST} = 33.01\%$
D	$\frac{SSD}{SST} = 11.88\%$

3. Confounding: From the table we can find a generator I = ACD generator. From which we can compute all the other possible confounding
{ first order: A = CD, B = ABCD, C = AD, D = AC
second order: AB = BCD, AC = D, AD = C, BC = ABD, BD = ABC, CD = A
third order: ABC = BD, fourth order: ABCD = B }

4. Since we have 8 Experiments, we can have as much as 8 unknown variables, so by increasing the number of factors, (I would start the search by including second order interaction could improve the system, if needed third and fourth order interaction). Otherwise, by preserving the number of unknowns, one could replace low effect (like q_B) with higher effects.

5. Resolution tells us the order of effects that are confounded. By default we want to have higher order interaction to be confounded. In the given table, resolution is 3.

4 Exercise 4, Operational Laws

In this exercise we have a closed system with two subsystems: cpu and cloud. Ans is $N^Q_{cpu} = T^Q_{cpu} * X_{cpu}$

$$\begin{aligned}\text{I am looking for Throughput in CPU}\\ \textit{From Schematics: } V_{cpu} = 1 + 0.5V_{cpu} \rightarrow V_{cpu} = 2\\ V_{cloud} = V_{cpu} * 15 = 30\\ D_{cloud} = V_{cloud}S_{cloud} = 30 * 0.01 = 0.3\\ \textit{BottleNeck Law: } X_{tot} = \frac{\rho_{cloud}}{D_{cloud}} = \frac{0.4}{0.3} = 1.33\\ \textit{Forced Flow Law: } \mathbf{X_{cpu} = V_{cpu} * X_{tot} = 2 * 1.33 = 2.66}\end{aligned}$$

$$\begin{aligned}\text{I am looking for Time in CPU queue}\\ X_{cloud} = V_{cloud} * X_{tot} = 30 * 1.33 = 39.9\\ T_{cloud} = N_{cloud}/X_{cloud} = 20/39.9 = 0.501\\ \textit{slide28: } T_{cpu} = T - T_{cloud} - Z = 50 - 0.501 - 3 = 46.4990\\ \textit{Slide 21: } S_{cpu} = \rho_{cpu}/X_{cpu} = 0.7/2.66 = 0.263\\ \textit{Slide 21: } \mathbf{T^Q_{cpu} = T_{cpu} - S_{cpu} = 46.4990 - 0.263 = 46.2360}\end{aligned}$$

Therefore: answer is $\mathbf{N^Q_{cpu} = T^Q_{cpu} * X_{cpu} = 46.2360 * 2.66 = 122.9878}$

5 Exercise 5, Bottleneck

First we assess the baseline performance:

1. $$\begin{aligned}D^{old}_{cpu} = B_{cpu}/C = 800/100 = 8\\ D^{old}_{disk} = B_{disk}/C = 1500/100 = 15\end{aligned}$$

   The bottleneck device is D_{max} i.e. the disk. Given that they are the same price, Marty should buy the new Disk.

2. $$\begin{aligned}V_{disk} = C_{disk}/C = 500/100 = 5\\ V_{slow} + V_{fast} = 5\end{aligned}$$

We know that the best Split happens when D_i are equal.

$$D_{slow} = D_{fast}$$
$$S_{slow}V_{slow} = V_{fast}S_{fast}$$
$$S_{slow} = B/C = 1500/500 = 3$$

We also know that the faster Disk is 3 times faster than the old slow one

$$S_{fast} = S_{slow}/3$$

Now we can solve the following system of linear equations:

$$3V_{slow} = V_{fast}V_{slow} + V_{fast} = 5$$

Then we obtain values:

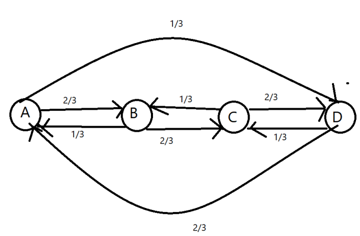
$$V_{slow} = 5/4$$
$$V_{fast} = 15/4$$

The ratio needed to maintain this split is given by $V_{slow}/V_{fast} = 1/3$

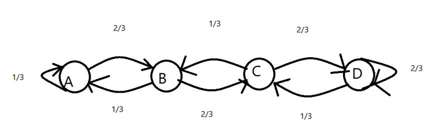
6 Exercise 6, Time Reversibility Equations

1. Figures

P1



P2



2. Using Time-Reversibility Equations, for P1:

$$A = 1/3B + 2/3D$$
$$1/3A + 2/3B = 1/3C + 2/3D$$
$$1/3A + 2/3C = D$$
$$A + B + C + D = 1$$

By solving those 4 equations with Gauss Reduction Method for Linear Equations we obtain the stationary probability vector: $\Pi = [1/4, 1/4, 1/4, 1/4]$

Using Time-Reversibility Equations, for P2:

$$
\begin{aligned}
2/3A &= 1/3B \\
1/3C &= 2/3B \\
2/3C &+ 2/3B + 1/3D \\
A + B + C + D &= 1
\end{aligned}
$$

By solving those 4 equations with Gauss Reduction Method for Linear Equations we obtain the stationary probability vector: $\Pi = [1/15, 2/15, , 4/15, 8/15]$

I have checked this results using also Balance Equations, and I obtained the same results.
Balance Equations P1:

$$
\begin{aligned}
A &= 1/3B + 2/3D \\
B &= 2/3A + 1/3C \\
C &= 2/3B + 1/3D \\
A + B + C + D &= 1
\end{aligned}
$$

Balance Equations P2:

$$
\begin{aligned}
A &= 1/3A + 1/3B \\
B &= 2/3A + 1/3C \\
C &= 2/3B + 1/3D \\
A + B + C + D &= 1
\end{aligned}
$$

3. By Definition (according to Karl Sigman 2009): If for an irreducible Markov chain with transition matrix P, there exists a probability solution π to the "time-reversibility" set of equations $\pi_i P_{i,j} = \pi_j P_{j,i}$ for all pairs of states i, j, then the chain is time-reversible.

   Therefore, called time reversible if the reverse-time stationary Markov chain has the same distribution as the forward-time stationary Markov chain. Hence it is reasonable that the rates are the same in both directions.

7 Exercise 7, Paper Review

7.1 Question (Q1)

Paper title: Habitat: A Runtime-Based Computational Performance Predictor for Deep Neural Network Training. [1]

Url : `https://www.usenix.org/conference/atc21/presentation/yu`

Authors: Geoffrey X. Yu, University of Toronto/Vector Institute; Yubo Gao, University of Toronto; Pavel Golikov and Gennady Pekhimenko, University of Toronto/Vector Institute.

Publication venue : 2021 USENIX Annual Technical Conference (USENIX ATC 21),

7.2 Question (Q2) Summary. What is this paper about, and what contributions does it make?

In this paper, the authors have proposed an innovative tool that helps to choose the most suitable GPU for training a deep net according to the user's needs. These recommendation important because hardware accelerator are expensive and the user may prefer to be informed about costs and performances in order to make a better choice according to his or her needs. Habitat suggests these recommendation by predicting computational time and costs for GPU which are not available; so the user can choose which one to rent .

This is possible because computations in DNN are highly repetitive, thus making predictions feasible.

Predictions depends on the chosen DNN architecture, hardware considered and software libraries used during training (e.g., cuDNN [74], cuBLAS [77]).

Habitat workflow operates in three steps: (1) Profile experimental data on an available GPU, (2) computes prediction using "Wave Scaling" or "Multi Layer Perceptrons" (3) evaluate overall training execution time.

At the core, predictions for non available GPUs depend on three scaling factors: Memory Bandwidth, wave size, clock frequency.

Lastly, they proposed two different case studies that ranked a few GPUs according to cost and performances tuned for specific DNN models.

This research addresses a problem of critical importance and provides a tool that offers quantitative suggestions. This is a huge step in this direction as current methods rely on heuristics which can have errors in predictions up to 49% , additionally getting measurements is tedious for different models, and this tool solves both problems.

7.3 Question (Q3) What strengths does this paper have?

1. This paper is very well argued and structured making and proving its points clearly. All the sections and subsections are relevant, and explain in a concise manner all the theoretical insights and design decisions.

2. The system is well designed, the accuracy yields much better results than the previous literature's work.Their model had up to 20% improvement

in predictions compared to other modeling techniques. The average error of 11.8% on ResNet-50, Inception v3, the Transformer, GNMT, and DCGAN.

3. It is also open source project and the applicability is feasible to every ML practitioner. Additionally it supports PyTorch library, making it even easier to integrate in one's project.

4. The authors have tested this tool on a variety of state-of-the-art networks architecture suited for a variety of domains, making its prediction relevant for real-world use. For example: Image Classification (ResNet [CVPR'16], VGG [ICLR'15], AlexNet [NeurIPS '12] Machine Translation (Transformer [NeurIPS'17] Seq2Seq NMT [NeurIPS'14]), Object Detection YOLO [CVPR'16] SSD [ECCV'16]), Speech Recognition (Deep Speech 2 [ICML'16] End-to-End w/ RNNs [ICML'14])

5. Considering Wave Scaling and MLP as factors, the authors proved that they are significant (thus rejecting null hypothesis). The split is around 46% for Wave Scaling and 54 % for MLP.

7.4 Question (Q4)

1. The case studies are a good way of testing the tool to real world settings. Though I think that they should test this tool in more scenarios. A 2^{k-p} factorial design with different architectures and models could yield better insights and extends this experimentation findings, providing a better grounding for ML practitioners.

2. Diminishing returns for MLP design features are not very well argued, but left vague. I would have preferred a quantitative analysis of those factors.

7.5 Question (Q5) How can this work be further extended or improved?

The paper has promising results, the way forward would be to extend this framework to TPUs (v1, v2, v3) and Emerging Accelerators like Cerebras WSE, Habana Gaudi, AWS Trainium. But perhaps more theoretical understanding of those accelerators can help fine tune the Wave Scaling mechanism, or finding a different innovative ways to compute the performances.

For Example, TPU rely on TPU compiles which may change the operations, thus contradicting the initial hypothesis on which Habitat is based. Would be interesting to learn about the difference in the compiler, and making appropriate changes to extend the system.

Or simply, to build an interface dedicated just for TPU architectures (note that AWS Traininum uses systolic arrays like Google's v2 and v3) using MLP and relying the statistical analysis of MLP parameters to fractional factorial design and ANOVA analysis. In this way, by renting a V1, one could infer about performances of V3 and Trainium to investigate the trade-offs between performances and cost and decided whether it is worth to switch hardware accelerator.

References

[1] Geoffrey X. Yu, Yubo Gao, Pavel Golikov, and Gennady Pekhimenko. Habitat: A runtime-based computational performance predictor for deep neural network training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 503–521. USENIX Association, July 2021.