Case Study Solution

by

Olagoke Lukman Olabisi

August 2018

Preface

This processbook describes the solution to the home case study at K&K. Parallel to this a Jupyter Notebook (in Python) will be submitted as a coding implementation solution to the case study. Reference to the coding solution will be made at the appropriate section of this process book.

Each main question in the case study is solved under a separate chapter. It is hoped that such an approach will facilitate easy reading and ensure appropriate considerations is given to important key points.

Contents

Chapter 1

Solution to Question 1

1.1 Question 1

Lets say you want to reproduce an animals function (e.g. visual recognition of food) with a computer algorithm. What kinds of methods/indicators could you use to compare the performance of your algorithm with the animals brain circuits which provided that same function? You have the freedom to explore different definitions of performance (time, complexity, energy consumption, physical size of the device performing the operations, etc.)

1.2 Introduction

It is known that half of non-human primate neocortex is devoted to visual processing[1]. Thus we have a robust working solution in such organisms which could be investigated and competitively compared with any other algorithmic solutions that replicate such (visual) function. This would form the fundamental motivation for the solution to this question. In particular since the visual system would be extensively used as a case study.

1.2.1 Visual Function: Core Object Recognition

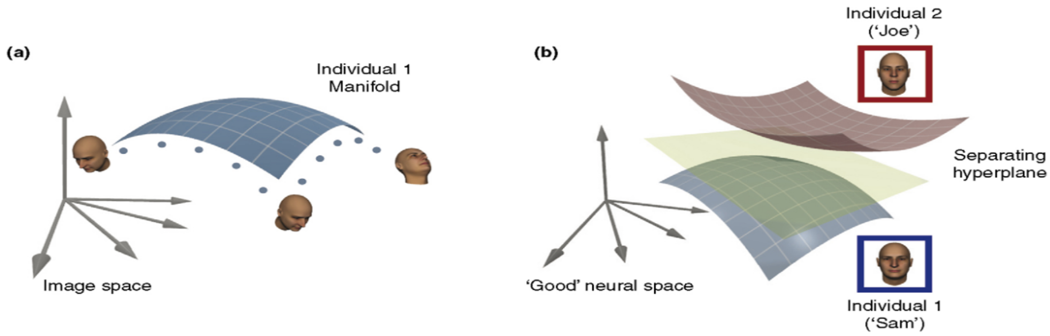
In other to make this comparison more tractable we define what aspect of the visual functions in organisms we are interested in replicating through a computational algorithm. Typically, vision accomplishes many tasks besides object recognition, including object tracking, segmentation, obstacle avoidance, object grasping, etc [2]. Here, the focus will be on core object recognition as defined in [2]: *the ability to assign labels to particular objects and doing so over a range of identity preserving transformations (e.g., changes in objects position, size, pose, and background context), and without any object-specific precuing.*

1.2.2 Computational Process for Object recognition

In order to model this object recognition for organism and compare to an algorithm that replicate such function it is necessary we define what computational process underlie this object recognition function .

As discussed in [2], one can take the view object recognition computationally (fig 1.1) as the problem of finding operations that progressively transform retinal representation into a new form of representation followed by the application of relatively simple decision function. Thus in this context we have reduced the problem into one of data representation and re-representation combined with the use of simple decision functions to validate such representation.

Figure 1.1: Simplified Object Recognition Computational Process for the Brain

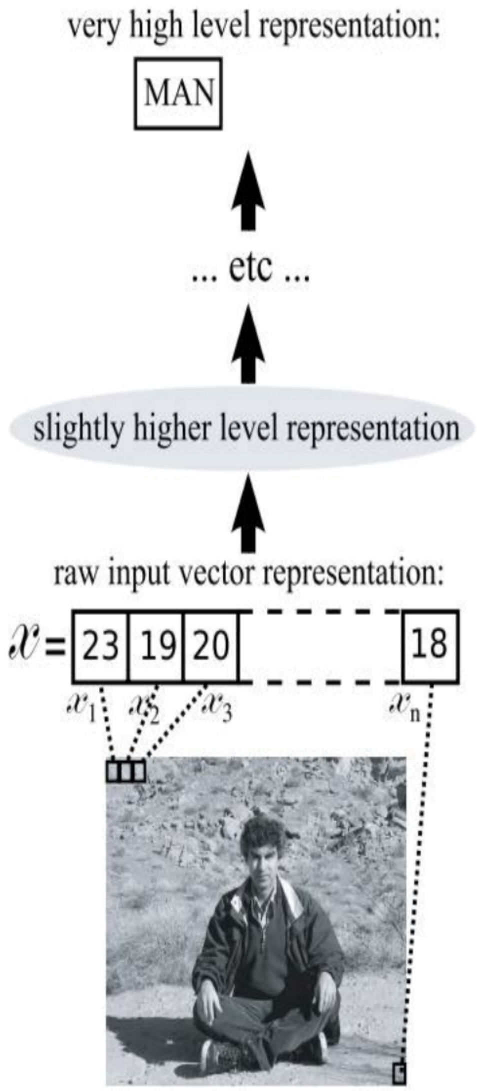


[Source : credit to [2].
For a givem neuronal population , each cardinal axis is one neurons activity. The dimensionality of the space is equal to the number of neurons. Higher dimensional space not shown.**(A)** An image of an object (say a face) is one point in retinal image space. As the faces pose is varied, the point travels along curved paths in the space, and all combinations of left/right and up/down pose (two degrees of freedom) **(A)** Same illustration but in a 2 image neuronal space.]

Computationally, the task is then to find an algorithmic approach that can efficiently allow for the representation and re-representation of data with a combined capability to apply decision functions. Base on this fact, it can be said that the aim of algorithmic approach would be to have different intermediate abstraction or representation layers of the object before a final decision layer where the object can be identified -based on the features extracted from intermediate layers pf abstraction.

Indeed such computational approach exists in the form a learning algorithm called Deep Neural Net. In [3], hierarchical correspondence of spatio-temporal cortical dynamics of human visual object recognition and deep Neural net was shown. To put this into perspective consider the figure below:

Figure 1.2: Deep Net Computational representation



[Source : [4]]

Similar to the visual cortex, the aim of deep net for this discussion is to take the raw input image and transform it gradually into higher levels of representation ; representing more and more abstract functions of the raw input [4]. From the figure, it is clear that many low level and high level abstraction will be needed to identify an object (say a Man). The aim of the algorithm would be to learn layers of abstractions based on perceived relevant statistical properties.[4].

1.3 Performance Comparison

In the discussion above we established the similarity of the visual cortex and the deep net architecture. How do can one compare the performance of visual cortex with Neural net on core object recognition task?

1.3.1 ROC or AUROC Curve

One way to compare performance is to use ROC curve. According to wikipaedia: A receiver operating characteristic curve, i.e. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

$$TPR = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$FPR = \frac{TrueNegative}{TrueNegative + FalsePositive}$$

A typical scenario would involve a animal trained on normal object recognition task. Performance can then be measured by presenting two object images or videos on a computer screen. The similarity between a object pair can be judged on the following scale:

- It is the same object(maybe);
- same object (maybe)
- Answer not known;
- They are different objects (maybe);
- They are same people (sure);

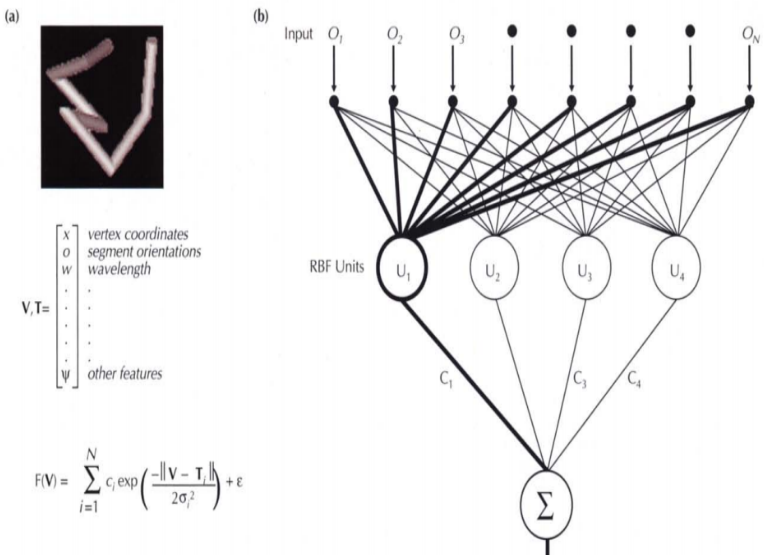
From the results of the answers above we compute the ROC curve for the aninal. Similarly, from training a deep net architecture we can compute the ROC curve. The ROC curve for the animal is then compared with that of the neural net. AUROC is the Area Under the Receiver Operating Characteristic curve.

The the cross-modal performance analysis framework (CMPA) has been used in [5] to analyse performance on face recognition tasks. It is also based on ROC curve comparison but more robust.

1.3.1.1 Example Case Study For ROC

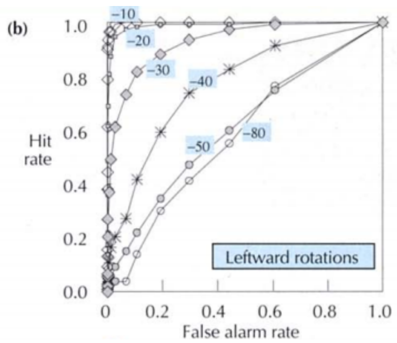
The case study here is adapted from [6], where Deep Neural Net was campared with vision system in primate.

Figure 1.3: Neural Network Topology



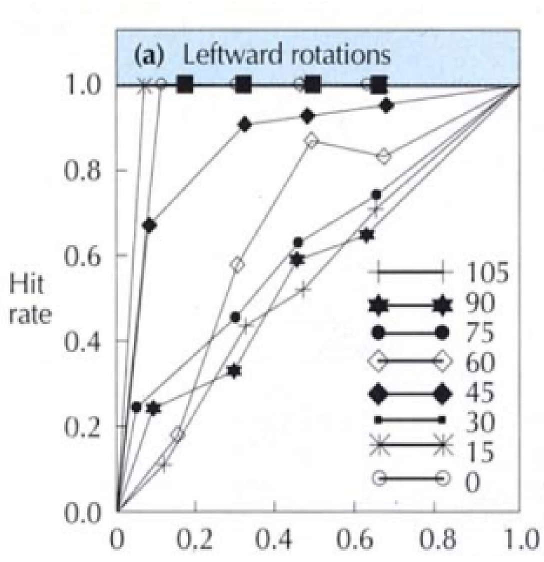
[Source : credit to [6]. **(a)** Object view is represented as a vector of features .**(b)** An example of an RBF architecture used for training]

Figure 1.4: ROC Curve for Neural Network Topology



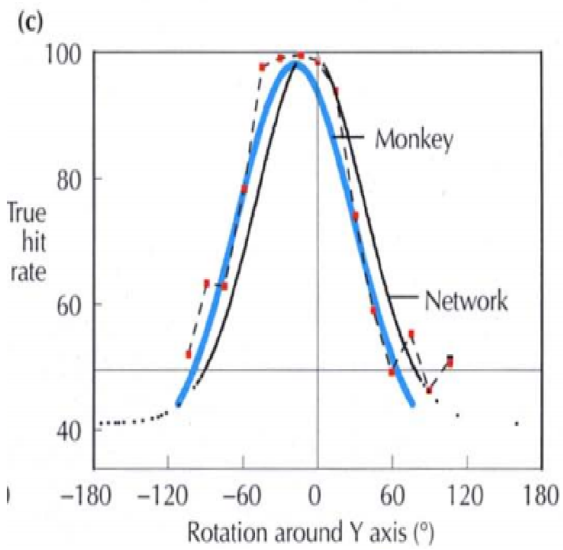
[Source : credit to [6]. **(a)** ROC curves generated after training on leftward rotated objects.

Figure 1.5: ROC Curve for Monkey Training



[Source : credit to [6]. **(a)** ROC curves generated after training a monkey. Animals are trained using standard conditioning techniques with positive reinforcement to perform a task. The xaxis represent the false alarm rate

Figure 1.6: Recognition Performance for Different object views



[Source : credit to [6]. Each red square represents the area under the corresponding ROC curve. The solid blue line models the data with a single Gaussian function, the thin black line is simulated data

1.3.2 Energy consumption

For any computational system, be it biologically-based, microprocessor-based, etc., the cost-per-computation or computational energy can be measured in terms of the energy required (in units of Joules) to perform the computation [7].

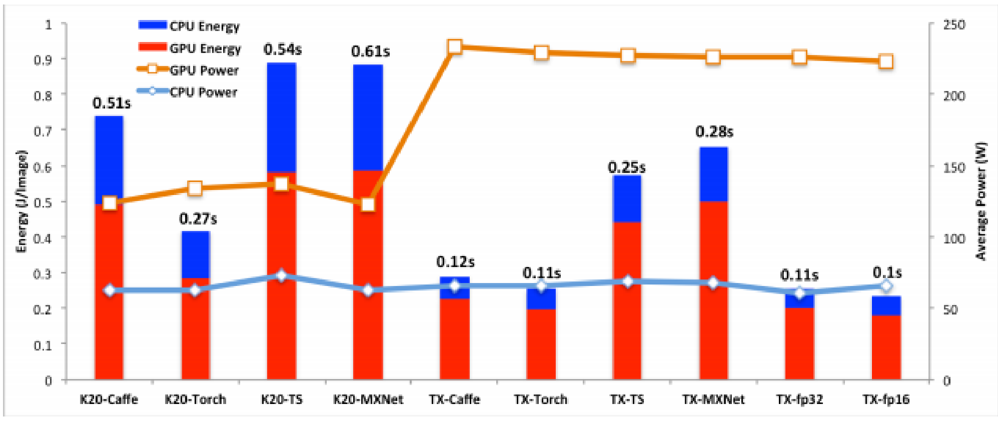
A typical microprocessor chip can perform about 10 million operations per second and uses about 1 watt of power. In round numbers, it costs about 10-7 joules to do one operation on such a chip. The ultimate silicon technology as of today will dissipate on the order of 10-9 joules of energy for each operation at the single chip level[7].

The brain for example, on the other hand, has about 1015 synapses. A nerve pulse arrives at each synapse on the average of 10 times per second. So, roughly, the brain accomplishes 1016 complex operations per second. Since the power dissipation is a few watts, each operation costs only 10-16 joules!

The brain is more efficient, by a factor of 10 million, than the best digital technology that we can hope to attain [7]. One reason for the inefficiency in computation energy is due to the way devices are used in a system[7]. In a typical silicon implementation, we switch about 104 transistors to do one operation[7].

The current trend is to use GPU for most deep net architecture training. And as fig 3.5 shows the results of this analysis on the Caffe framework, which can be configured to either use a native GPU implementation or rely on the cuDNN library. A total of 500 iterations of forward and backward propagation with a batch size of 128 was tested. The bars represent the energy consumption per image processed (left y-axis). Specifically, the bottom (red) and top (blue) part of each bar indicate the energy consumption of GPU and CPU, respectively. The two lines show the average power consumption (in Watts) of GPU and CPU (idle); the power consumption scale is on the right y-axis. On top of each bar we report the execution time of a single iteration. In the experiments used two Nvidia GPUs: a K20m and a Titan X (shown as K20 and TX along the x-axis, respectively) and four networks (AlexNet, OverFeat, VGG_A and GoogleNet)[8].

Figure 1.7: Deep Net Computational representation



[Source : credit to [8].]

As observed from the plot, this is far less efficient compared to visual cortex operations in the brain. It is hoped that hard-ware specifically tailored to neural net (example neuromorphic processors) would be more energy efficient.

1.3.3 Processing Time And Time Complexity

As discussed in [9] model processing times for deep net are currently competitive with primate behavioral reaction times. This however still requires an in-depth research.

Heuristically, the time complexity of neural net is proportional to the number of nodes, number of weights and the memory usage but it really depends on the kind of neural network - taking into considering forward and back propagation. Heuristically, we expect the runtime to scale linearly in the number of input nodes, quadratically in the number of hidden states, and linearly in the number of output nodes. Analysing the complexity of neural system or specifically the neural cortex would be a daunting task given the gap in our knowledge of how neurons precisely function. But we know definitely that neurons have very fast reaction time[10].

1.3.4 Representational Performance

This section is due to [9]. In [9] an extension of **kernel analysis** , was used to measure the accuracy of a representation performance. It was concluded that , the latest DNNs rival the representational performance of IT cortex

To evaluate the neural representation, multi-unit and single-unit neural activity from awake behaving rhesus macaques during passive fixation [9] were recorded. This was done using large scale multi-electrode arrays placed in either IT cortex or visual area V4 [9].

To create a neural feature vector for assessment of representation of performance, each image (1960 images in total) was presented for 100 ms. This is followed by measuring the normalized, background subtracted firing-rate in a window from 70 ms to 170 ms post image onset, averaged over 47 repetitions[9].

To evaluate neural model, kernel analysis was carried out. Kernel analysis evaluates the efficacy of the representation by measuring how the precision of the category regression problem changes as we allow the complexity of the regression function to increase [9]. Intuitively, more effective representations will achieve higher precision at the same level of complexity because they have removed irrelevant variability from the original representational space.

For further facts on this section see [9]

1.4 Conclusion

Though the focus has been on the visual representation, the same performance criteria can be extended to other functions of the brain.

Chapter 2

Solution to Question 2

2.1 Question 2

K&K is creating an artificial biological nose (for detection of compounds in the air, as commonly associated with trained canines) with on-the-chip neurons. One of the key steps to create this artificial nose is to transform neurons into biosensors (BS). There are thousands of different biosensors, each responding in a different manner to different compounds. For this, we design an experiment where some cells are expressing specific sensors and can produce a signal when these BS are bound to a compound. By comparing the response of the sensors (BS1, BS2, BS3, etc) to different compounds (A, B, C, etc), the relative affinity of a given sensor-compound pair can be determined.
SIMPLIFIED PROTOCOL

1. A population of cells is kept in an incubator and maintained for several days before they are dispensed into the wells of a plate.

2. For each well, the cell population is given treatments to transform them into one of twelve different types of biosensor

3. The plate is inserted into the machine and a certain volume of liquid compounds is dispensed into each well.

4. The machine measures the response of the cells to the injected solution and writes the data to a file. The entire protocol is performed twice for each compound, A through Z, and the resulting data is uploaded into a database.

Data for twelve specific biosensors was queried and aggregated into a file called OutputDATA.csv. In the tab Data you will find the signal (in arbitrary units) produced by the sensors when they are in presence of the compounds. We performed the experiment twice to have an idea of the experimental variance (that is why there are two values corresponding to each compound - biosensor pair). It is important for lab automation tasks to be able to analyze various kinds of data during operation so that specific conditions trigger further inspection of the data or other downstream processes. This exercise is an example of such data analysis, and should not require any biology knowledge that is not mentioned here.

1. What normalization would you perform on this data? Normalize the data according to the method of your choice. Explain your reasoning for this method and be sure to incorporate the normalized data in your explanation.
2. By analyzing the data with the method of your choice (Excel, python, R, etc.) describe the different relationships that you observe and the different biosensor characteristics that emerge. Implement statistical tests and do not hesitate to use graphs and tables and to present your results.

2.1.1 Solution

Please find solution to this question in the attached Zip file. The zip file contains 4 files:

- casestudy.ipynb - the main Jupyter Notebook solution
- Docker File
- Data set (excel)
- Readme file

There are 2 ways to run the notebook:

1. Run the Jupyter notebook by itself
2. Using the Docker file to build and run docker image. Helpful instructions on this is in the readme file. This option is more flexible and requires one to have

docker installed. All needed package are bundled in docker image that ensures the notebook runs as produced by the person who solved it. It avoids package compatibility issues.

The read me file gives notes on how to build and run the docker image using the docker file. After running the docker file copy paste the link in any browser to launch the notebook. Docker file has been added to ensure the required libraries are all bundled in each run. This also helps avoid library compatibility issues and achieve consistency of results as discussed earlier.

2.2 Conclusion

This section documents the solution to question 2.

Chapter 3

Solution to Question 3

3.1 Question

After several repetitions of the process discussed in Question 2, it appears that there is often large variability within and between experiments. This makes it difficult to reach clear conclusions or produce consistent results. However, for a company which plans to produce thousands of biological machines for critical applications, clear and consistent results are absolutely necessary.

One approach to this problem is to employ automation, extensive data collection, and machine learning - -this is where you come in. Use the following example and your own background knowledge to guide your responses:

Suppose that you are trying to measure the influence of certain compounds – A through Z – on a population of cells living on a 96 well plate, such as:

- Odor response intensity or pattern
- Population growth rate and longevity
- Receptor expression efficiency
- Network structural development

After several days of experimentation, you analyze your samples with a plate reader and find that, while some wells show promising results, many more are either inconclusive or even go against your hypothesis. You repeat your experiment, taking extra care to treat each well the same way, yet the second batch seems just as variable as the first.

1. What factors might affect your results? Suppose you can use multiple smart devices to measure whatever data you like during the experiment (pH, temperature, microscope videos, electrical signals, etc.) Choose a few and consider the type of data they collect.

2. Employing high-throughput automation of experiments allows for results to be generated on a much larger scale. Given the amount and diverse nature of the data collected, how would you design a database system to integrate the information across multiple devices and/or experiments? Represent your ideas using concept maps, schema diagrams, text and/or other methods you prefer

3. What platforms and tools would you use to implement this data infrastructure? How would you make it amenable to the kind of queries and analysis performed in Question 2?

3.2 Solution Q3.1

Given the fact that the readings were carefully taken, the following factors might be affecting the results:

1. **Random noise** might be a major factor affecting the results. For this we might have several statistical methods for denoising the collected readings.

2. Another factor is the test methodology. If the test methodology is faulty then the results might be inconsistent. In particular, it might be possible there are experimental variables or effects that were not accounted for which might have caused some cell wells to give inaccurate results.

3. Another factor would be the **bias** of the experimenter. This has been observed in various data analysis problem. While we have our hypothesis, consideration should be given to what the data tells which might at times be counter intuitive.

We might therefore need to check if enough **control** has been set up such that only intended factors are tested. Bias might cause us to :

- Reject null hypothesis when it is true
- Failing to reject the null hypothesis when it is false

4. There might also be Measurement error due to the instruments used in measuring signals from each well. The units of measurement should also be checked across devices to ensure consistency. Correct labelling of well plates and compounds should be checked.

3.2.1 Smart Devices And Data

The following is a non-exhaustive list of smart devices that might be useful:

- NeuLog Colorimeter Sensor: The inexpensive NeuLog Colorimeter Sensor measures Beer's Law or changes in concentration over time for reaction rate studies. The colorimeter sensor measures light transmitted by 3 different color LEDs through a solution, turning on the LEDs and measuring the received light that passes through the solution. Requires connection hardware
- NeuLog UVB Sensor: The NeuLog Ultraviolet Light UVB Sensor measures UV light in the UVB wavelength range of 280 to 320 nm. This represents only 2% of the total UV radiation, but is the most commonly measured.
- PASCO Wireless Temperature Sensor PS-3201: This durable, wireless, high-resolution sensor features a stainless steel temperature probe for the most demanding of applications. It can be used in a wide array of experiments and activities as it measures small but significant temperature changes produced by chemical reactions, convection currents, and even skin temperatures. The rugged sensor housing/handle is rated for temperatures of -10 C to 40 C.
- PASCO Wireless Colorimeter And Turbidity Sensor - PS-3215: The Wireless Colorimeter and Turbidity Sensor simultaneously measures the absorbance and transmittance of six different wavelengths. Each wavelength represents a region of the ROYGBIV color wheel and can be displayed graphically using PASCO software.

The colorimeter can be used to study concentrations of solutions, the rates of chemical reactions and more. The Wireless Colorimeter and Turbidity Sensor is also resistant to splashes and spills, ensuring years of use in labs. It was also designed to draw minimal amounts of power which gives it a long battery life and reduces how often it needs to be recharged. Calibration is also fast and easy, eliminating wasted time during the lab activity.

- PASCO Wireless pH Sensor - PS-3204 : This sensor measures the pH of a solution as discrete measurements or as a continuous reading. Use the probe to study water quality, test household solutions, or perform high-resolution acid-base titrations with ease.

- PASCO Wireless Conductivity Sensor - PS-3210: Wirelessly measure both conductivity and dissolved solids. Highly accurate with a fast response time and featuring built-in temperature compensation. The Wireless Conductivity Sensor measures the electrical conductivity of a water solution. Investigate the properties of solutions, perform conductometric titrations, model and measure water quality and more.

- Wireless Spectrometer - PS-2600 : The PASCO Wireless Spectrometer is specifically designed for introductory spectroscopy experiments. The Bluetooth and USB connectivity enable use with tablets and computers. With this one apparatus, one can measure intensity, absorbance, transmittance and fluorescence, making this a powerful and intuitive tool for your spectroscopy needs. Includes 10 plastic cuvettes and lids. PASCO's FREE Spectrometry Software (download link below) walks you through the process of calibration, performing a full spectrum scan, determining the optimum wavelength for study, and concentration and kinetics experiments

- ThermoFisher Eutech Ion Selective Combination Electrode: for measurement of specific ions in aqueous solutions.

- Eutech TDSTestr11 Dual Range TDS Tester: Eutech offers meters that allow the direct reading of Total Dissolved Solids (TDS) values. Eutech Waterproof TDSTestr11 is a dual range meter with selectable or auto-ranging options. Waterproof, anti-roll design, large display screens, user-replaceable sensor, adjustable tds factor for more accurate measurements, and the ability to switch between C and F are just some options of this meter.

- Nanolive Holotomographic Microscope: Tomographic Microscope to look instantly inside living cells in 3D. Useful for Observe and measure in real-time the effect of compounds and stimuli applied cells. Also helps to Analyze cell behaviour in in vivo-like systems. Observing how they organize into 3D matrix or on 3D surfaces

The following links were useful for answering this question 3:

- `https://nanolive.ch/applications/`
- `https://www.pasco.com/index.cfm`
- `https://neulog.com/products/`

3.3 Solution Q3.2

Given the needs of the experimental setup the following task might be needed to be carried out :

- Batch processing of data sources at rest.
- Real-time processing of data in motion
- Interactive exploration of data
- Predictive analytics and machine learning

This summarizes what Q2 and part of Q3 is trying to achieve. A solution that ensures ease of collaboration among different team will be desirable. In addition, replicability of results and automation of processing flow should be considered in the solution proposal.

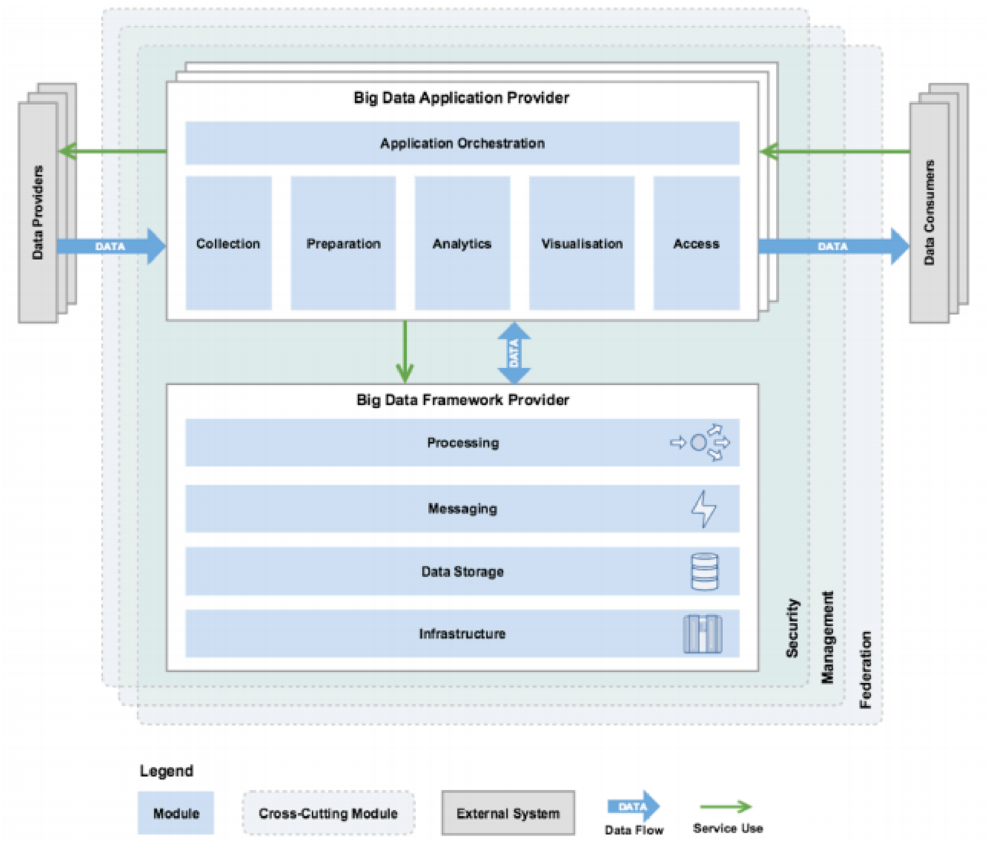
A **big data** architecture would be necessary because it is needed to :

- Store and process data in volumes that are large

- Transform unstructured data for analysis and reporting
- Capture, process, and analyze unbounded streams of data in real time, or with low latency
- ensure scalability,replicability and seamless integration.
- Automation of processing pipeline
- Support for schema on read

As the figure below shows, the big data architecture incorporates whatever data storage , flow processing and analysis is needed:

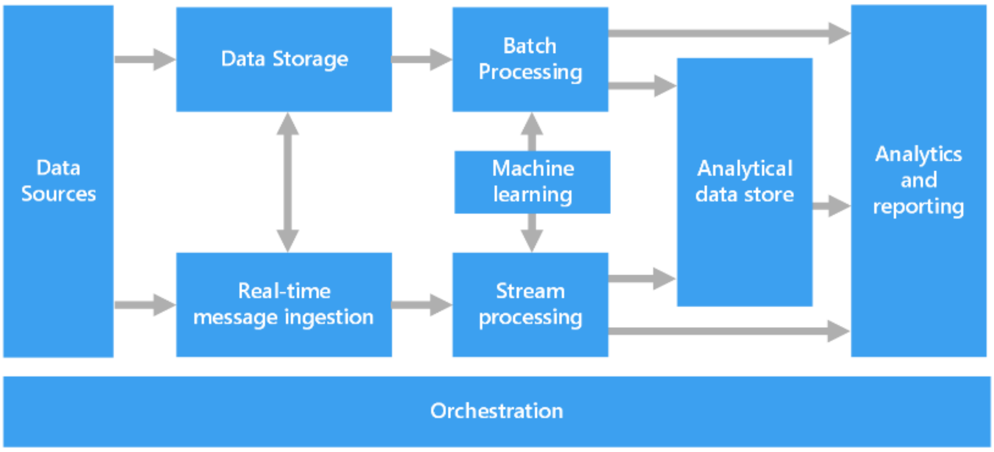
Figure 3.1: Big Data Architecture Overall View



[Source : credit to [11].]

3.3.1 Model Architecture

Figure 3.2: Big Data Architecture Core Module



A few comments on this might be necessary:

1. Data source come from the experiments. It can be structured or unstructured data set. Semantic integrability should be ensured by giving consideration to schema mapping while setting up the data set. This ensures integrability across platforms.

2. Data storage: Data would typically be stored in a distribute file system which can large data set in different formats. Options for implementing this storage include: Azure Data Lake Store or blob containers in Azure Storage

3. Batch processing: this entails processing data files using long-running batch jobs to filter, aggregate, and otherwise prepare the data for analysis. Options here include : Options include running U-SQL jobs in Azure Data Lake Analytics, using Hive, Pig, or custom Map/Reduce jobs in an HDInsight Hadoop cluster, or using Java, Scala, or Python programs in an HDInsight Spark cluster.

4. Analytical data store: This involves preparing data for analysis and then serve the processed data in a structured format that can be queried using analytical tools. HBase, and Spark SQL, can be used to serve data for analysis

5. Analysis and reporting: . The goal of big data solutions is typically to provide insights into the data through analysis and reporting. To enable users to analyze the data, the architecture may include a data modeling layer, such as a multidimensional OLAP cube. Analysis and reporting can also take the form of interactive

data exploration by data scientists or data analysts. For these scenarios, platforms such as Azure services that support analytical notebooks, such as Jupyter, enabling these users to leverage their existing skills with Python or R.

3.4 Solution Q3.3

This will be discussed under the following sub-headings:

3.4.1 What types of data is being analyzed?

If it is Excel, a relational database like Postgres, MySQL, Amazon Redshift or BigQuery will fit the needs. If data is unstructured a non-relational (NoSQL) database like Hadoop or Mongo would be the good option. If processing might involve doing a large amount of text mining, language processing, or image processing, then one need to use non-relational data stores.

Figure 3.3: Comparison of Data Base

Choosing a Database		
Criteria	Relational	Non-relational
Type of Data	Structured	Unstructured
Would fit in massive	Excel Sheet	Word Doc
The schema	Stays the same	Changes often
Works well with data like	User data, Inventory	Email content, Photos, Video
For analysis like	User paths, Funnel analysis	Text mining, Language processing
Can query with	SQL	MapReduce, Python

3.4.2 How much data is being processed?

The chart below can help in deciding base on size of data.

Figure 3.4: Data Size

Database Options by Scale				
Data Size	< 1TB	2TB-64TB	64TB-2PB	#allofthedata
Database that's a good fit	Postgres MySQL	Amazon Aurora	Amazon Redshift Google BigQuery	Hadoop

3.4.3 Third party echo system

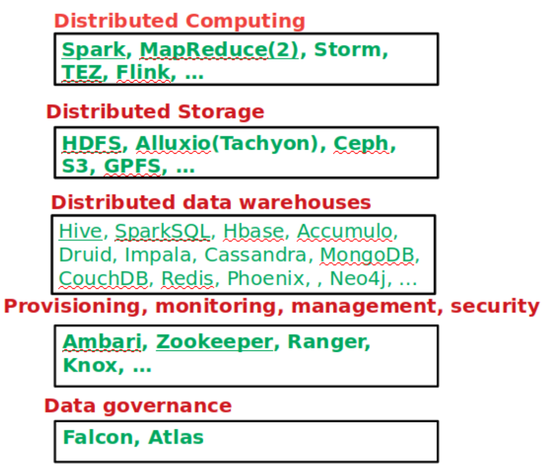
Google Cloud Storage. Amazon S3 and Microsoft Azure might be a viable cloud option to reduce down time in managing the big data echo system.

This way engineers can focus on improving productivity of services and critical data analysis important to the compnay.

3.4.4 Task And Typical Platform

Lastly the chat below shows the typical task of the architecture and the platform that can be used :

Figure 3.5: Task and Platform



The following links were useful for answering this question 3:

- `https://segment.com/blog/choosing-a-database-for-analytics/`
- `https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/`

3.5 Conclusion

While setting up the data base the data base schema should be checked to ensure semantic operability across platforms. This will enable deployment of solution across heterogeneous platform. Typically XML could be as data definition language. XML can ensure syntactic interoperability, but the semantic aspect requires human effort and should be taken into consideration in the architectural pipeline.

Bibliography

[1] DJ Felleman and DC Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, N.Y. : 1991)*, 1(1):147, 1991. ISSN 1047-3211. doi: 10.1093/cercor/1.1.1-a. URL `https://doi.org/10.1093/cercor/1.1.1-a`.

[2] JamesJ. DiCarlo, Davide Zoccolan, and NicoleC. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415 – 434, 2012. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2012.01.010. URL `http://www.sciencedirect.com/science/article/pii/S089662731200092X`.

[3] Radoslaw M. Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 2016.

[4] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2 (1):1–127, January 2009. ISSN 1935-8237. doi: 10.1561/2200000006. URL `http://dx.doi.org/10.1561/2200000006`.

[5] P. Jonathon Phillips and Alice J. O'Toole. Comparison of human and computer performance across face recognition experiments. *Image and Vision Computing*, 32(1):74 – 85, 2014. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2013.12.002. URL `http://www.sciencedirect.com/science/article/pii/S0262885613001741`.

[6] N.K. Logothetis, J. Pauls, H.H. Blthoff, and T. Poggio. View-dependent object recognition by monkeys. *Current Biology*, 4(5):401 – 414, 1994. ISSN 0960-9822. doi: https://doi.org/10.1016/S0960-9822(00)00089-0. URL `http://www.sciencedirect.com/science/article/pii/S0960982200000890`.

[7] Mohamad H. Hassoun. *Fundamentals of Artificial Neural Networks.* MIT Press, Cambridge, MA, USA, 1st edition, 1995. ISBN 026208239X.

[8] Da Li, Xinbo Chen, Michela Becchi, and Ziliang Zong. Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus, 10 2016.

[9] Charles F Cadieu, Ha Hong, Daniel LK Yamins, Nicolas Pinto, Diego Ardila, Ethan A Solomon, Najib J Majaj, and James J DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10(12):e1003963, 2014.

[10] James J DiCarlo and David D Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.

[11] John Klein, Ross Buglak, David Blockow, Troy Wuttke, and Brenton Cooper. A reference architecture for big data systems in the national security domain. In *Big Data Software Engineering (BIGDSE), 2016 IEEE/ACM 2nd International Workshop on*, pages 51–57. IEEE, 2016.