Capstone project

Requirements are **Horizontal** [deployment/operationalization and real application centered, Dev + MLops + ML] and **Vertical** [centered on in depth understanding and exploration of deep learning - basically model training]. They refer to different aspects of the **Capstone project** that needs to be addressed. They help provide a comprehensive and well-rounded understanding of the project. Below is a detailed breakdown of the **horizontal** and **vertical** requirements.

Horizontal

Horizontal requirements focus on the deployment, operationalization, and real-world application of the project. They encompass a combination of development (Dev), machine learning operations (MLOps), and machine learning (ML) itself. In essence, these requirements involve:

1. Development (Dev): This aspect is about creating the software infrastructure, tools, and applications to support the machine learning model. It may include developing APIs, front-end interfaces, and other components that help users interact with the model.

2. MLOps: Machine Learning Operations (MLOps) is the process of managing and automating the lifecycle of a machine learning model. It includes steps like model versioning, deployment, monitoring, and maintenance. MLOps aims to establish a smooth and efficient workflow that ensures the consistent performance of the model in a production environment.

3. ML: This aspect focuses on the design, training, and evaluation of the machine learning model. It includes selecting the appropriate algorithms, data preprocessing, feature engineering, and model validation.

Vertical

Vertical requirements emphasize a deep understanding and exploration of deep learning techniques, primarily focusing on model training. These requirements are more specialized, as they delve into the specifics of the chosen deep learning approach. These requirements involve:

1. ***Model architecture***: Understanding the structure and design of the deep learning model, such as the number of layers, types of layers, and their connections. This can include exploring various architectures like Convolutional Neural Networks (CNNs), Graph Neural Network (GNNs), Recurrent Neural Networks (RNNs), or Transformers, depending on the problem being addressed.
2. ***Hyperparameter tuning***: Experimenting with different hyperparameters, like learning rate, batch size, or the number of layers, to optimize the model's performance. This step involves techniques like grid search, random search, or Bayesian optimization to find the optimal set of hyperparameters for the model.

3. ***Loss functions and optimization algorithms***: Understanding and selecting the most suitable loss functions and optimization algorithms for the specific problem, ensuring the model converges and learns effectively from the training data.
4. ***Regularization and generalization***: Implementing strategies to avoid overfitting, such as dropout, weight decay, or early stopping. This helps ensure that the model can generalize well to new, unseen data.
5. ***Model evaluation and validation***: Assessing the performance of the model on validation and test data using appropriate metrics, such as accuracy, F1-score, or mean squared error, depending on the problem type.

Combined

By addressing both **horizontal** and **vertical** requirements in a **Capstone project**, aim is for you to demonstrate the ability to not only develop an effective deep learning model but also deploy and operationalize it in a real-world setting. This holistic approach ensures that the project is both theoretically sound and practically applicable.

Levels

A brief description of the **levels** to pass the **Capstone project**:

Minimal (7): Complete two deep learning tutorials covering different topics (e.g., GANs and CNNs). Make additional improvements or tuning to the models, such as improving accuracy or reducing execution time. Alternatively, participate in two Kaggle competitions and achieve better results than a random guess (e.g., accuracy above 50% for binary classification).

Minimal+ (8): Complete two deep learning tutorials on different topics and provide significant additional improvements or variations. For example, if a tutorial uses one framework, reimplement it in another framework while improving accuracy or execution time. Alternatively, submit work to two Kaggle competitions and rank within the top 20% of the leaderboard for each competition.

Medium (9): Choose an area of interest not covered in class and implement a machine learning or deep learning model. Alternatively, compete in three Kaggle competitions and rank within the top 15% of the leaderboard for each competition.

Max (10): Develop a usable end-to-end system with an ML/DL model at its core. The model should not be covered in class and should be entirely created by you. The system doesn't need to be unique or completely original but should be functional.

Highest: Implement a research paper, verifying its results, and essentially peer-reviewing the process and findings. Provide a detailed report on your own discoveries and observations. This level demonstrates your ability to understand and reproduce the research work of others, as well as critically analyze their results.

Exceptional: Write an original research paper authored by you. This level showcases your ability to conduct novel research, contribute to the field of machine learning or deep learning, and effectively communicate your findings.

For all levels (except the two highest (purple), additional requirements include:

- Providing a report (which can be in the same notebook or optionally through GitHub repository) explaining the rationale behind the steps you have taken. This report should demonstrate your understanding of the concepts and techniques used in the project.

- Sharing your code on Google Colab or in a notebook on GitHub (preferably Colab). If you choose to provide a separate report from the code, ensure that the code is minimally annotated and presentable as a project for a job application.

Some ideas from the previous groups:

- Image classification from a minimal training set: given two images one with an item and another without can we train a DNN/CNN to classify them correctly and how can we achieve that with minimal resources).
- Stock market prediction model evaluation - comparative analysis between various known models.
- Anomaly detection for time series data.
- Music classification (genre) / music generation.
- Take your own photos and create style transfer / deep dreaming project and give a gift to your friends
- Shoping list printer that can identify produce missing in the fridge and automatically give you a shopping list.