EE/CprE/SE 492 Bi-Weekly Report 1
February 19 - March 4
Team sdmay23-05
Project Title: Livestock Well-being Identification (to be changed)
Client/Advisor: Dr. Ashraf Gaffar

Team Members:

Asad Abdalla - Dev Ops Manager
Rashed Alyammahi - Backend Manager
Mohammed Elbermawy - Client Interaction
Yannick Fumukani - Frontend Manager
Richard Gonzalez - Scrum Master / Team Organizer
Meet Patel - Advisor Interaction
Adam Sweiger - Status Reporter

Summary of Past Two Weeks

Over the past two weeks, the team set up an Amazon Sagemaker Notebook instance in AWS, used this instance to train the machine learning image classification model, then documented the process and results of training in the cloud compared to training locally on the team's virtual machine. The purpose of this documentation is to provide steps to guide readers through the process of training a machine learning model in the cloud and to analyze the performance in terms of cost and time required for training.

Accomplishments from Past Two Weeks

- Asad Abdalla:
  - Explored two approaches to train the model: using AWS SageMaker, below are the tasks that were performed
  - Trained the model using AWS SageMaker
  - Checked the price and specifications of AWS SageMaker to ensure that it meets the project requirements.
  - Uploaded the data to an S3 bucket "Just test".
  - Created an AWS SageMaker notebook instance and launched Jupyter Notebook to write the code for the model.
  - Configured the training job by specifying the hyperparameters, instance type, and other settings.
  - Ran the training job and monitored the progress, adjusting the settings as needed.
  - Reviewed the metrics to assess the performance
- Rashed Alyammahi:
  - Assisted in initial configuration of Sagemaker instance in AWS

    - Brainstormed potential causes to Keras error on AWS by output tracing all the parameters in the fit() function and model object
    - Worked on the documentation summarizing the background nature of the ISIC dataset and which particular subsets of the dataset was utilized to train the model
    - Concisely documented the process of creating a Sagemaker notebook, installing the appropriate packages for the environment, moving data into the environment, and running the Python script for training the data to the model. Provided all screenshots of the process
    - Documented the performance differences between each Epoch iteration between the VM and AWS and speculated on causes of the differences between GPU and CPU utilization
    - Discussed interpretation of model results, including the significance of output parameters from the training results and the effect of training sample size and epoch volume on accuracy
- Mohammed Elbermawy:

  Communicated with AWS customer service to explore the available services that could be utilized to move the model training from the local VM to the AWS cloud. During the conversation, we explored AWS SageMaker, a service provided by AWS that offers pre-built algorithms, frameworks, and tools used to build, train, and deploy machine learning models quickly. Additionally, AWS SageMaker provides pre-configured instances that can be used for model training. AWS customer service also assisted us in configuring EC2, which is a virtual server that provides compute capacity in the cloud. Used EC2 to run the training jobs for the model. Overall, the assistance received from AWS customer service was instrumental in transitioning the model training from a local VM to the AWS cloud.

- Yannick Fumukani:
    - Researched different AWS services related to AI for modeling and training, such as Sagemaker, Rekognition, Comprehend, and DeepLens.
    - Explored some AWS Sagemaker features:
      Sagemaker allows users to easily scale up or down the resources needed to train and deploy AI models; it's cost-effective, and flexible since you can pay as you go and integrate with other AWS services such as S3, EC2 and 3rd party tools, such as Jupyter.
    - Run the AI model on our VM.

- Richard Gonzalez: Researched how AWS can be utilized to train some given models. Set up AWS account for the team to utilize Sagemaker. Set up notebook instance for the senior design team and configured the JupyterLab instance to run our specific models. Worked within the team to get past errors faced and successfully trained and documented running a specific model on AWS. Was able to experiment with different instances with access to more CPU power and set the correct one for our model. Set up individual accounts for Sagemaker on AWS and distributed those throughout the team. Actively managing AWS utilization is an ongoing process.

- Meet Patel: Worked with the team to set up AWS training model. Researched the prediction and accuracy of the model. Brainstormed the potential errors for AWS Sagemaker.
- Adam Sweiger:
    - The python code for the AI model works as intended without errors when training on the VM, but upon training in Sagemaker Notebook, a Keras error appeared that prevented the model from training. Researched the error and attempted multiple fixes before finding one that worked. It turned out that one of Keras' API functions called in the code requires a different input format depending on whether the model is being trained using a CPU or GPU. The VM uses a GPU for training, while the Sagemaker Notebook instance uses a CPU, explaining why the error occurred in Sagemaker but not on the VM. The solution that resolved the issue was changing the Keras backend variable for image data format to be set to channels_last instead of channels_first.
    - Another issue encountered while training in Amazon Sagemaker is that the code would find extra images to train and sometimes try to give the model 3 classes of images to train on instead of 2. This would have severe negative impacts on the accuracy and performance of the model. Discovered that JupyterLab inside Sagemaker Notebook was automatically generating hidden files in the same directory as the dataset of images, causing the problem to occur. To solve the problem, deleted all of the hidden files every time they were discovered. In the future, it would be a good idea to find a way to prevent JupyterLab from generating these files.
    - Trained the AI model using the Amazon Sagemaker instance. Initially used a Jupyter Notebook to execute the code but found that the kernel would occasionally die in the middle of training, which would waste a lot of time. To resolve this, researched ways to run code using the command line in Sagemaker. Opened a terminal in Sagemaker and activated the conda environment containing TensorFlow and other libraries required to run the code. Once in the conda environment, execution is achieved by running 'python3 filename.py'.
    - Documented results of training AI model in Sagemaker and analyzed the performance compared to local training. Training was much slower on the cloud than on the team's virtual machine, largely due to the significant gap in available computing resources between the two environments. The team may want to

explore using more expensive Sagemaker instances in the future to allow for training on larger datasets.

Individual Contributions

Team Member	Contributions	Weekly Hours	Cumulative Hours
Asad Abdalla	Amazon Sagemaker model training and documentation	3	6
Rashed Alyammahi	Assisted in Sagemaker configuration and bug fixing, documented process and analysis of results	6	10
Mohammed Elbermawy	Assisted in the setup of the cloud computing environment, as well as training the model on both the cloud and the VM, and then assisted in the documenting of the results from both settings.	6	9
Yannick Fumukani	Learned about AWS Sagemaker, trained the AI model on the VM, and helped with the documentation	5	8
Richard Gonzalez	Setup AWS and environment. Worked through getting models trained on AWS.	8	11
Meet Patel	Learned about AWS Sagemaker, assisted in training the model	4	9
Adam Sweiger	Modified AI model code to fix AWS errors, trained model on AWS and VM, documented results of training	8	15

Plans for the Upcoming Two Weeks

The team will be meeting with the advisor, Dr. Gaffar, on Monday to discuss project progress and upcoming tasks. One potential task that might be pursued is using a more expensive Amazon

Sagemaker instance with more computing power to train the model on a larger dataset because the team is currently operating within the AWS free tier. Doing so would allow the team to train on more data in less time. Another potential area of interest is developing a simple user interface that takes an image of a skin lesion as user input and runs a benign vs. malignant prediction on it using the trained model. In addition, the team may develop more machine-learning models related to skin cancer that can be trained on the ISIC dataset.