Module 2: Understanding AI Models

In this module we're going to get our hands dirty examining the nuts and bolts of some example AI models. The models we're going to look at today are Large Language Models (LLM), Image Classification, and Sound Classification. These examples are intended to spark your creativity and intuition about what *you* can do with AI but they are in no way exhaustive. There are many, MANY different AI models that can do everything from spot odd activity in credit card transactions to removing objects in photos and filling them back in with generated content. We can train the models with human input labeling the data (supervised learning), we can allow the models to find patterns and associations on their own (unsupervised learning), or we can design models that reinforce desired outcomes, allowing agents to learn on their own (reinforcement learning).

The big takeaway here is that AI models are spectacular at finding, highlighting, and exploring patterns in data, especially with datasets too large for humans to manage well. In the case of generative AI, as we'll see in the upcoming section on LLMs, they can even take the patterns they've discovered and recreate them!

Module 2 Objectives:

By the end of this module, you will be able to:

1. Differentiate between three real-world use cases of AI training and deployment, including what dataset was used and what type of model was trained.
2. Understand what an AI model is, identify major types (image classification, sound classification, large language model), and kinds of data needed for each.
3. Explain the difference between supervised, unsupervised, and reinforcement learning.

Big Language

As we touched on in the last module, generative text models have gone from a pipe dream to big business. Let's cut to the chase and explore how they actually work. Keep in mind that to a large extent, the principles that underlie the workings for LLMs are the same for all machine learning (ML) applications.

Large Language Models from Scratch - https://youtu.be/lnA9DMvHtfI [1]

Large Language Models: Part 2 - https://youtu.be/YDiSFS-yHwk [2]

Now let's tinker with an LLM, HuggingChat. **(As for the previous module, please note that current USDA guidance prohibits using externally hosted generative AI tools for work purposes, so limit your interactions to non-work topics.)**

HuggingChat - https://huggingface.co/chat/

1. Use the model for a few minutes.
2. Using what you learned in the videos above, try to get the model to make mistakes. What did you try, and why?
3. Was the model easy to "trick"? Why do you think that is? What can we do to make models like these more robust?
4. How could you use an LLM in your work?

A Machine You Can Teach

Onto something completely different; Image Classification. We're going to use Google's Teachable Machine to train an image classification model. Yes, *we*, right now, are going to train a model. Ready?

Start here: https://teachablemachine.withgoogle.com/

- Watch the videos "What is Teachable Machine?", "Gather Samples", "Train your model", and "Export your model".
- Navigate here: https://teachablemachine.withgoogle.com/train
- Select Image Project.
- Select the Standard image model option.
- Using the instructions from the videos in step 1, create your own image classifier! You can upload images or use your webcam. We recommend using your webcam, as that is a much faster way to get your model up and running. **(Again, please note that you may not use work-related images for this purpose.)**
- Play around. Try making a model that recognizes if your face is in the frame or not, then add more classes. Experiment with using more samples or fewer samples per class and see how that affects the model's confidence when making predictions. Another thing to try is experimenting with variance in your samples: Try a sample set where you stay very still versus one where you move around a bit. How does that affect your model's confidence?
- Under the Training section, click Advanced. You'll see three parameters: Epochs, Batch Size, and Learning Rate. If you hover your mouse over the (?) icon, you'll get a description of what each of these things does. Here are simple overviews:
    - Epochs: The number of times you feed your entire dataset through the training algorithm. More is usually better, up to a point.
    - Batch Size: How many samples are grouped (batched!) together and fed through your training algorithm simultaneously.
    - Learning Rate: This is the size of each "step" we take towards our solution. Too large and we risk shooting past the solution, too small and the solution may take too long or too many compute resources to find.
- Select the "Under the hood" button at the bottom of the Training section. A sidebar will pop up. You can look at the Vocab subject, but what we're really looking for are the Accuracy per Class, Confusion Matrix, Accuracy per Epoch, and Loss per Epoch. For Accuracy per Class and Confusion Matrix, you'll need to click the Calculate buttons.
    - Accuracy per Class: A measure of how well a ML model performs at classifying each individual class in a multi-class problem. It is calculated by dividing the number of correctly classified instances for each class by the total number of instances in that class.

The Language of AI

If you're panicky about the terms, that's normal. We're just touching on them now, but we'll go into more depth in later courses.

  - Confusion Matrix: A table that is used to evaluate the performance of a ML model. It shows the number of times the model correctly classified an instance and the number of times it incorrectly classified an instance. The Confusion Matrix can be used to calculate various metrics, such as accuracy, precision, and recall (we'll explain these in more depth in later courses).
  - Accuracy per Epoch: A measure of how well a ML model performs at classifying instances after each epoch of training. It is calculated by dividing the number of correctly classified instances after each epoch by the total number of instances.
  - Loss per Epoch: A measure of how well a ML model performs at predicting the correct output after each epoch of training. It is calculated by averaging the loss over all the instances in the training data.
- Now that you've had a chance to play with the model, how could you use an Image Classification model in your work?

Can You Hear Me Now?

Finally, we'll test out a sound classifier. This model works similarly to the Image Classifier you just used. Follow the steps outlined above to train, test, and explore your model. As before, while you're working with this model consider how you might use this kind of model in your work.

References

[1] Graphics in 5 Minutes. "Large Language Models from Scratch." YouTube, 16 July 2022, youtu.be/lnA9DMvHtfI

[2] Graphics in 5 Minutes. "Large Language Models: Part 2." YouTube, 16 July 2022, https://youtu.be/YDiSFS-yHwk