Week 3

- A way of creating web-based machine learning could be making a model in python, exporting it as JSON, and using it in a browser
- Using Browser based training have lots of compromises
- Instead of using browser based training, it's better to use existing trained models
- Could also use scripts to convert your pretrained models to make them usable in Java Script

Toxicity Classifier

- Import required packages

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/toxicity"></script>
```

- Code for Toxicity Classifier:

```
<!DOCTYPE html>
<html>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/toxicity"></script>
<script>
    const threshold = 0.9;
    toxicity.load(threshold).then(model => {
        const sentences = ['you suck'];
        model.classify(sentences).then(predictions => {
            console.log(predictions);
            for(i=0; i<7; i++){
                if(predictions[i].results[0].match){
                    console.log(predictions[i].label +
                                " was found with probability of " +
                                predictions[i].results[0].probabilities[1]);
                }
            }
        });
});
</script>
<head>
</head>
<body>
</body>
</html>
```

Mobile-Net

- List of supported objects in Mobile-Net: LINK

- To import mobile net

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"> </script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/mobilenet@1.0.0"> </script>
```

- Code for Mobile Net:

```
<!DOCTYPE html>
<html>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"> </script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/mobilenet@1.0.0"> </script>
<head>
</head>
<body>
    <img id="img", src="coffee.jpg"></img>
    <div id="output" style="font-family:courier;font-size:24px;height=300px"></div>
</body>
<script>
    const img = document.getElementById('img');
    const outp = document.getElementById('output');
    mobilenet.load().then(model => {
        model.classify(img).then(predictions => {
            console.log(predictions);
            for(var i = 0; i<predictions.length; i++){
                outp.innerHTML += "<br/>" + predictions[i].className + " : " + predictions[i].probability;
            }
        });
    });
</script>
</html>
```

Converting Models to JSON

- Python Code and Jupyter notebook code:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/169c3e65-2cb6-4bfe-90b6-234cb9000b9f/convertingToJSON.ipynb

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/943f18c7-38e2-4cc3-b5c5-c0e996263319/convertingtojson.py

Install Wget

The exercise for this week uses a package called **Wget** to download a filtered version of the Dogs vs. Cats dataset available on Kaggle.

If you run the notebook and get an error message similar to "**wget: command not found**" when trying to download the dataset, you probably don't have **Wget** installed. If this happens, you will need to install **Wget** on your system**. Wget** is free and is available for various platforms including Windows, Mac OS, and Linux. Below, we will go over the installation process of each of these.

Install Wget on Windows

1. Go to https://eternallybored.org/misc/wget/
2. Download the **wget.exe** file from the links provided. You can download the latest version of **wget** for either 32-bit or 64-bit systems.
3. If prompted, click **Run** or **Save**.
4. If you chose **Save**, double-click the downloaded file to start installing.

Exercise

- Code:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5affa4da-e9b6-4cd3-88c5-1e55407bab4b/tfjs_week3_exercise.py

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/09565867-0523-478d-8d8b-23c8a0a1a65e/TFJS_Week3_Exercise.ipynb