Deploying an ML model

ML Meet 6 @SSN Coding Clubb

Scan this QR code to ask questions anonymously, or just use MS Teams. The link is also in the chat.

Quick Recap!

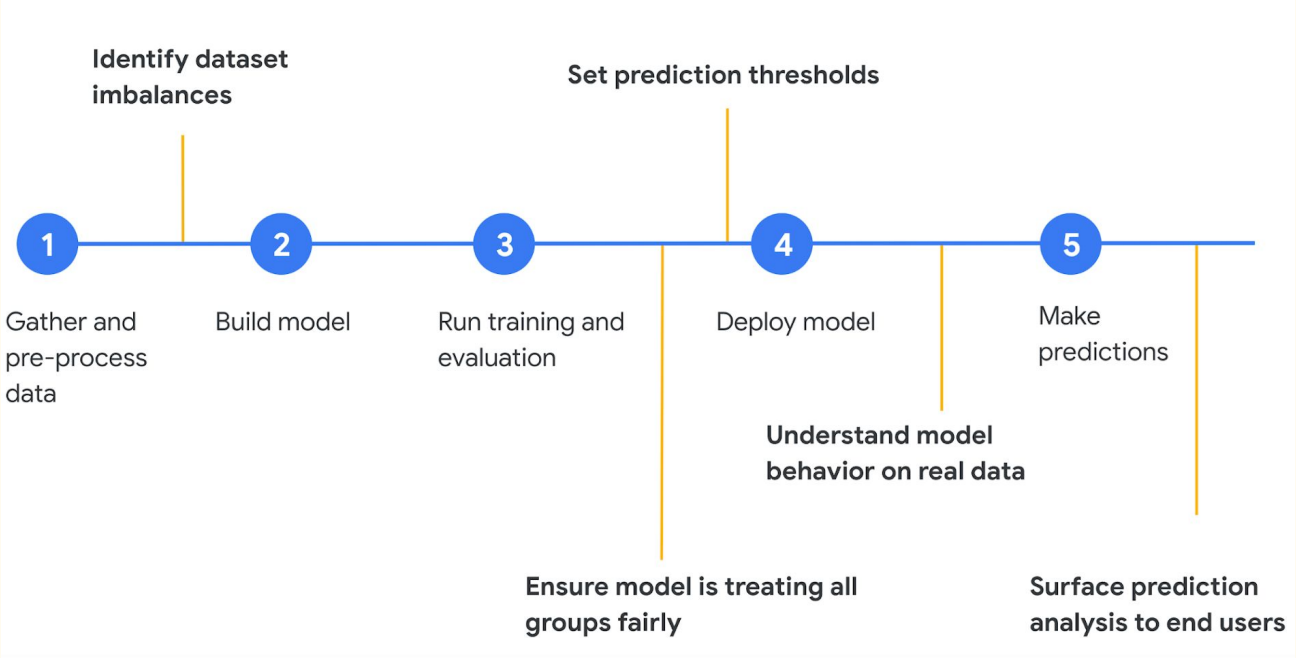
The Sudoku Solver

1. Found the largest square in the image
2. Unskewed the image
3. Extracted the cells
4. Decided if each cell contains a digit or not
5. Applied an ML model like CNN to find the digits (a classifier, that is)
6. Solve the sudoku

Now, what? We have to deploy this model so that any one of us can use it!

Deploy?

- Making the ML model available for real use, often on the web or as an app.



Why deploy?

In order to start using a model for practical decision-making, it needs to be effectively deployed into production.

If you cannot reliably and quickly get practical insights using your model, then the impact of the model is severely limited.

The question now

Should you make use of a **client-based approach** or **server-based approach** for deployment? What do these mean in the context of Machine Learning?

Server-side ML

- Model hosted on remote server and I/O shared between the client and server while predicting
- Ensures that ML can be used by any client (web, mobile) regardless of their computational power, since the inference is carried out server side.
- Most popular

Advantages

- Powerful servers can be used, allows to extend complex models
- Independent of the client's platform. Eg: A webapp and an Android app can use the same endpoints
- Hardware limitations of end-user devices can be disregarded

Disadvantages

- Expensive and Powerful servers need to be maintained and monitored.
- Network-induced latency might cause inference to be even slower than Client-side implementations.

Client-side ML

- A part of the model (say, weights) or the model configuration file may be imported to the client, where it is run locally
- Theoretically, lower latency
- Developing in recent years

Advantages

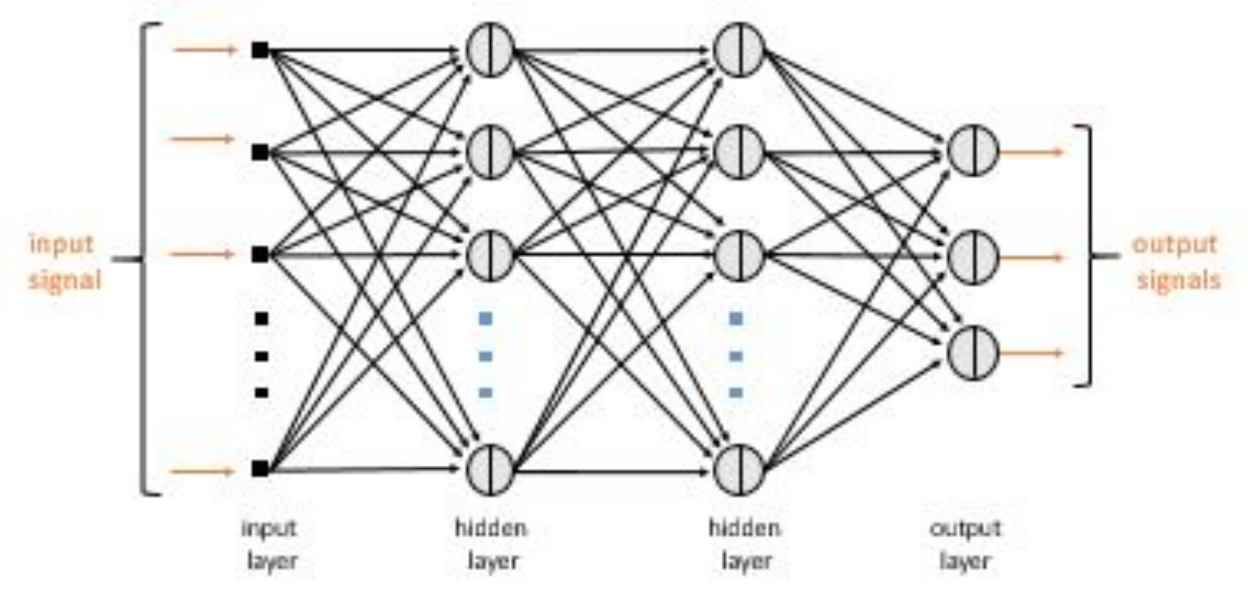
- Cheaper. We don’t need servers that are always available
- Delegate work to the client-side!
- Many frameworks available now, although research still continues
- An internet connection is required only to fetch the model files, once.

Disadvantages

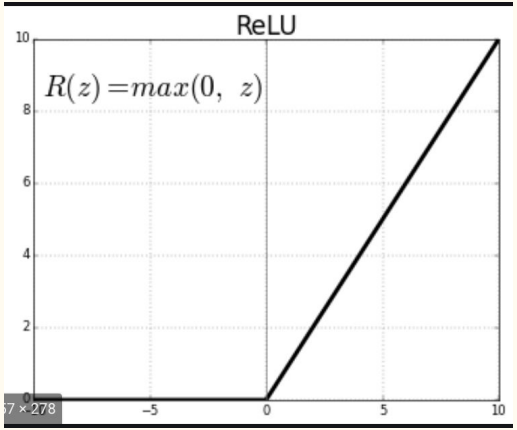
- Hardware dependent. The framework/implementation is based on client platform
- The hardware dependence could affect the latency inversely. Low power devices will have huge wait times.

Developing the neural network

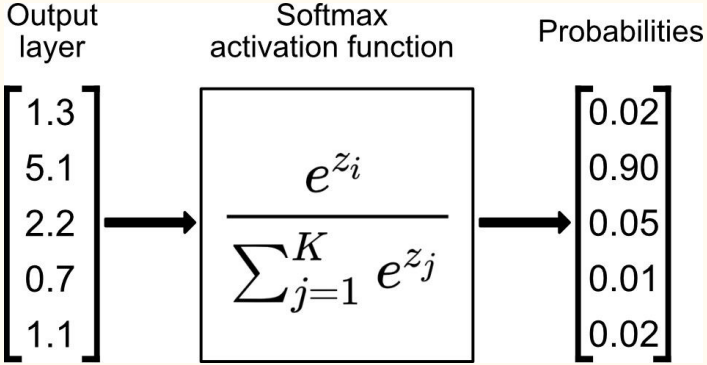
- Iris dataset
- Multi-Layer Perceptron (MLP) with two hidden layers



Activation functions



Rectified Linear Unit
(ReLU) activation function



Softmax activation function

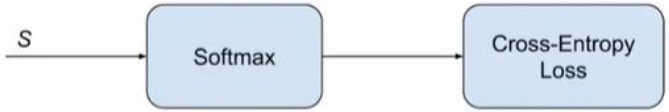
Loss function

Categorical cross-entropy loss (or softmax loss)

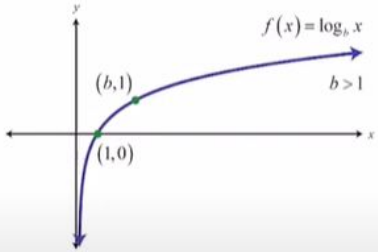
- It is a Softmax activation plus a cross-entropy loss

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) =$$

Sp is the positive class



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$



- Example:

True Label: Rabbit

Prediction: Dog = 1, Cat = 4, Rabbit = 8, Squirrel = 2

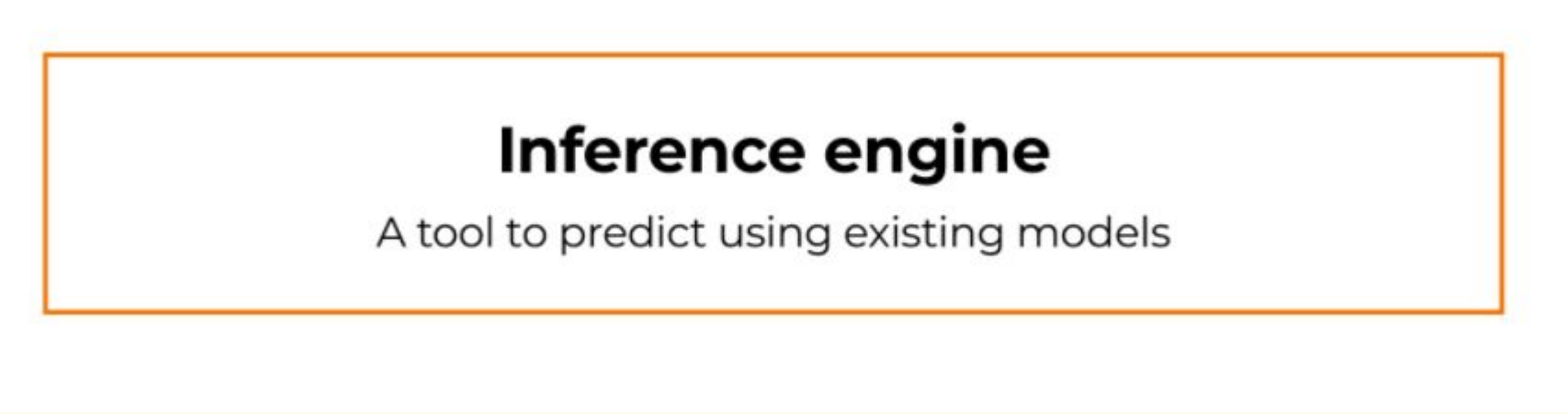
Softmax : D = e^1/SUM, C = e^4/SUM, R = e^8/SUM, S = e^2/SUM

CE Loss = - (0 * ln(D) + 0 * ln(C) + 1 * ln(R) + 0 * ln(S))

= - (0 + 0 + (-?) + 0)

Client side ML - Tensorflow.js

- Framework for training and executing machine learning models with JavaScript
- Your application doesn't have to send data
- Low latency
- Useful for time-sensitive applications, e.g., real-time object detection



Setting up a local HTTP server

```
npm install -g http-server
```

- This command is used to install the http-server package using the node package manager

```
http-server
```

- This command will start the http server

Cloud services for deploying ML models

Thank you