

**Ease of Use for tensorflow** – provides flexibility

**Ease of Use for Pytorch** – it’s **dynamic computation graph** approach developed more flexible model compare to tensorflow

**What is a dynamic computation graph?**

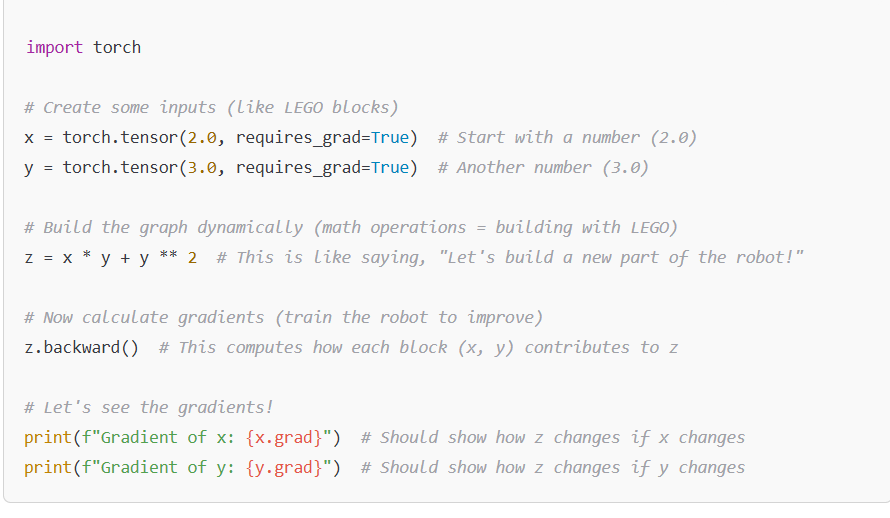
Imagine you're building something cool with LEGO blocks, like a robot. You don't know exactly what your robot will look like until you start building it, and you can change it as you go along—maybe you add wings or make its arms longer. This way of building is flexible because you decide what to do step by step.

Now, a **dynamic computation graph** in PyTorch works like that! It's like a flexible blueprint for math problems. Instead of having a fixed plan for solving a problem (like a recipe you can't change), PyTorch lets you create and change the steps of the plan as you go.

If you're training a robot (a neural network) to learn something, PyTorch builds this plan while you work, and it figures out how to improve the robot by looking at its mistakes. And if you want to try a different way of building the robot, you can change the plan right away.

This is super useful because it’s like making your LEGO robot smart and ready to adapt to new ideas while you're still building it!

Here's a simple example of a **dynamic computation graph** in PyTorch:



**Ease of Use for Keras** – develop train model with minimal no of code, High level API runs top of tensorflow

**Ease of Use for scikit-learn –** Focuses on traditional ml algorithm classification, regression, clustering

**PERFORMANCE OF EACH:**

1. Tensorflow’s static computation graph(SCG) helps develop large model with high performance and scalability.
2. Pytorch has DCG allows efficient memory usage and better performance, it may not scale as well compare to tensorflow
3. Keras integrated with tensorflow so it has better performance for deep learning task compare to only tensorflow.
4. Scikit may not give good performance like tensorflow, because it follows only the traditional method, it is good for efficient classification tasks

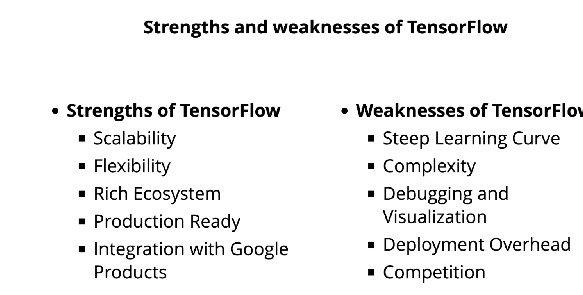
**COMMUNITY AND ECOSYSTEM**

1. Tensorflow has good C and E
2. Pytorch has growing C and E on computer vision
3. Keras has same good C and E like tensorflow
4. Scikit also has good C and E.

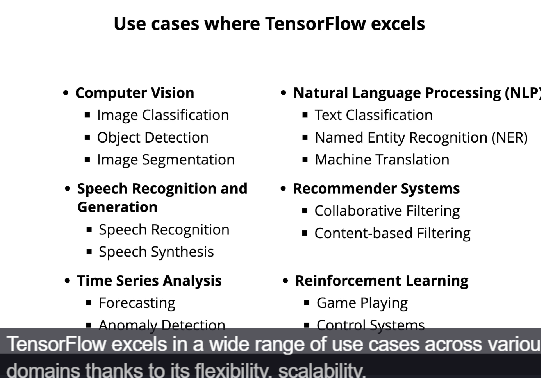
**MODEL DEPLOMENT**

1. Tensorflow and keras model can be easily deployed in mobile based app
2. Pytorch, scikit need more manual work to do deployment for mobile

But four of this can be deployed in any online platform

****

****

****