

Introdução

O objetivo do tutorial é implementar um simples sistema de reconhecimento de comandos por voz usando os exemplos do TensorFlow com alguns ajustes em uma placa da NVIDIA Jetson Nano. Tal placa possui uma GPU incluída, o que a torna ideal para executar programas que fazem uso de redes neurais.

![Alt]("./img/jetson nano.png")

Imagem do JETSON

Update de libs e packages

```
$ sudo apt update
$ sudo apt upgrade
```

Instalar Pip3

```
$ sudo apt install python3-pip
$ sudo pip3 install -U pip
```

Instalar dependências do TensorFlow

```
$ sudo apt install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev
zip libjpeg8-dev
$ sudo pip3 install -U numpy==1.16.1 future==0.17.1 mock==3.0.5
h5py==2.9.0 keras_preprocessing==1.0.5 keras_applications==1.0.6 enum34
futures testresources setuptools protobuf
```

Instalar Tensorflow

```
$ sudo pip3 install --pre --extra-index-url
https://developer.download.nvidia.com/compute/redist/jp/v42 tensorflow-gpu
$ sudo pip3 install -U pip
```

Testar instalação do TensorFlow

```
$ python3
```

```
>>> import tensorflow as tf
```

O comando acima deve executar sem nenhum erro se o TensorFlow tiver sido instalado corretamente.

Download do exemplo do TensorFlow

O TensorFlow possui um exemplo em seu repositório de utilização de redes neurais para reconhecimento de comandos por voz. Nesse tutorial, utilizaremos esse mesmo exemplo com algumas adaptações, uma vez que o modelo fornecido recebe como input um arquivo de áudio **.wav**. Nosso objetivo é criar um programa que recebe um stream de áudio que deve ser processado continuamente, assim como é visto em assistentes de voz (Google e Siri).

O primeiro passo é clonar o repositório raiz do TensorFlow.

```
$ git clone https://github.com/tensorflow/tensorflow.git
```

Esse comando pode demorar um pouco para ser executado, uma vez que o repositório é grande. Vá tomar um café! ☕

Vá até a pasta do exemplo que queremos executar.

```
$ cd tensorflow/tensorflow/examples/speech_commands/
```

Note que na pasta existem diversos arquivos, os principais são:

- `train.py` : utilizado para treinar o modelo.
- `freeze.py` : utilizado para compilar o modelo treinado.
- `label_wav.py` : utilizado para reconhecer um comando dado um arquivo `.wav` de input e um modelo previamente treinado