Break New Ground

GPU não é luxo, é necessidade!

Waslley Souza
Solution Engineer

Blog: waslleysouza.com.br
LinkedIn: waslleysouza
Github.com/waslleysouza
waslley.souza@oracle.com

ORACLE

Waslley Souza

- Solution Engineer na Oracle Brasil
- Entusiasta e curioso sobre Python, ML e DL
- Palestrante e escritor nas horas vagas
- Oracle ACE
- Dog Lover!

BASEADO EM FATOS REAIS

~~UM FILME DE ROMAN POLANSKI~~

Uma história de Waslley Souza

Era uma vez um desenvolvedor ....

Que descobriu o ...

TensorFlow

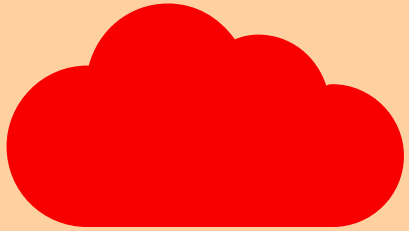
Mas nem tudo foram flores ...

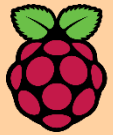
Achou uma alternativa mais simples ...

E evoluiu!

Keras

TensorFlow





Raspberry Pi

Flask

Mas, precisava da teoria ...

Blogs

TensorFlow

UDACITY

YouTube

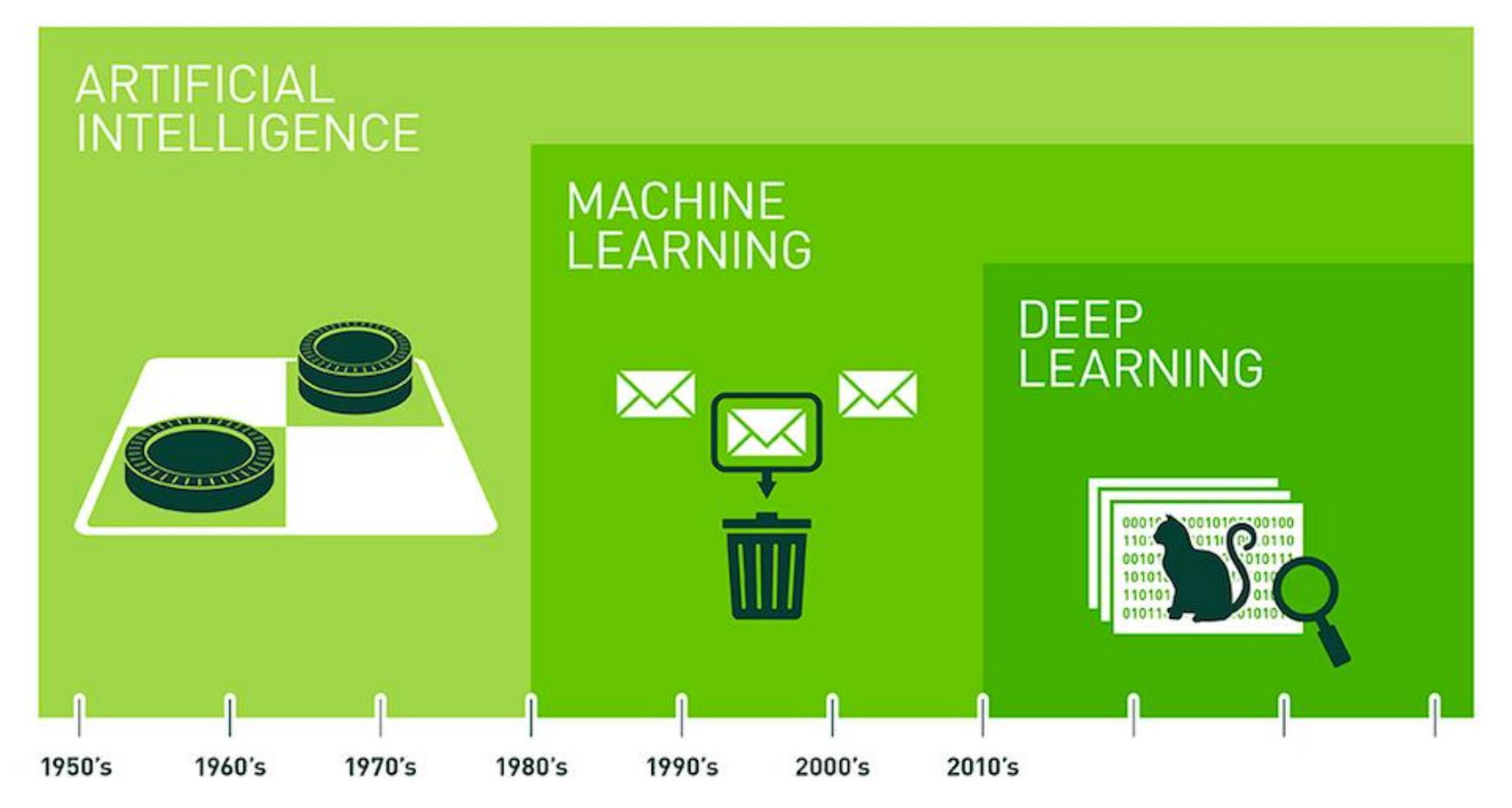
Udemy

DataCamp

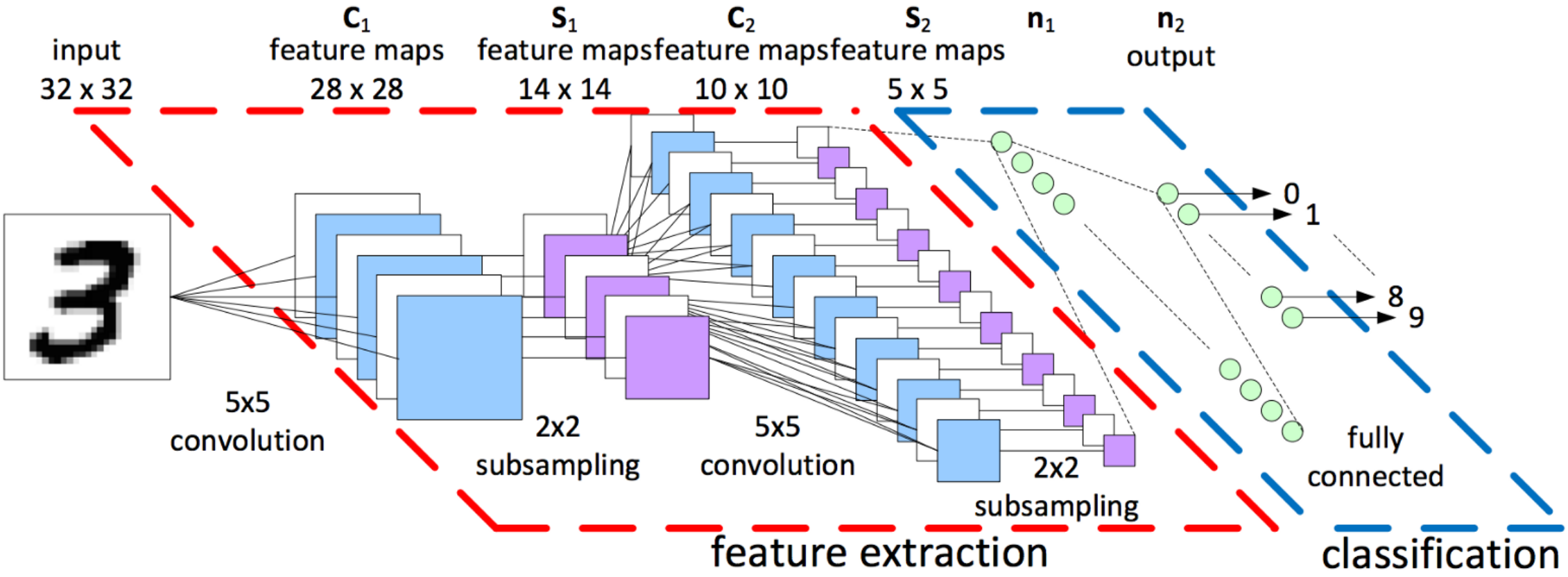
kaggle

coursera

Conheceu IA ...


ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
DEEP LEARNING
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Aprendeu Redes Neurais Convolucionais ...



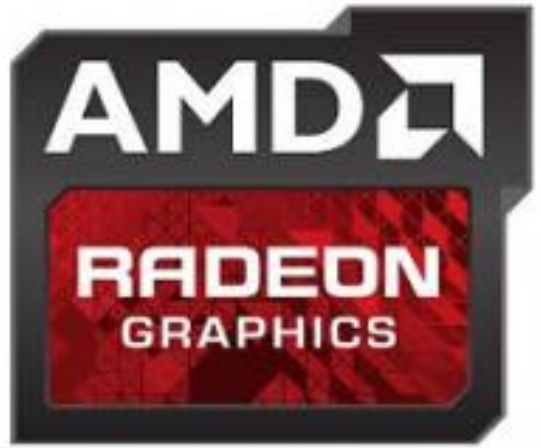
Mas a vida é feita de decepções!

Começando a estudar DL

Tentando treinar uma CNN com > 2M de parâmetros

https://www.youtube.com/watch?v=-P28LKWTzrI

GPU?? Bora ver o que é isso!



Best GPU For Mining

Ele descobriu algumas opções ...

Ambientes Compartilhados

- Google Colab (12hs, TPU??)
- Kaggle Kernels (9hs)

Tesla K80
Cuda Cores: 2496
Memória: 12GB
Performance: 8 TFLOPS

Ambientes Pagos

- **Oracle Cloud** e outras nuvens ($ ou $$$$$)

Tesla V100
Cuda Cores: 5120
Memória: 16GB
Performance: 15 TFLOPS

Comprar

- Computador Gamer ($$$)
- eGPU + GPU ($$$)

RTX 2070
Cuda Cores: 2304
Memória: 8GB
Performance: 7.5 TFLOPS

https://medium.com/@_NicT_/colaboratorys-free-gpu-72ebc9272933
https://www.kaggle.com/docs/kernels#technical-specifications

E qual foi o resultado?

Premissas

- CPU i5-8350U (4 cores e 8 threads), 32GB
- eGPU Sonnet + RTX2070 8GB (20% de perda / sem usar TensorCores)
- Python 3.6 + Tensorflow 2
- Classificação binária
- Treinamento: epochs=10, batch_size=20
- github.com/waslleysouza/tensorflow2

https://ark.intel.com/content/www/br/pt/ark/products/124969/intel-core-i5-8350u-processor-6m-cache-up-to-3-60-ghz.html
https://egpu.io/forums/mac-setup/pcie-slot-dgpu-vs-thunderbolt-3-egpu-internal-display-test/

kaggle Search Competitions Datasets Kernels Discussion Learn ...


Dogs vs. Cats

Create an algorithm to distinguish dogs from cats

215 teams · 5 years ago

Overview Data Kernels Discussion Leaderboard Rules Team

Data Description

The training archive contains 25,000 images of dogs and cats. Train your algorithm on these files and predict the labels for test1.zip (1 = dog, 0 = cat).

A note on hand labeling

Per the rules and spirit of this contest, please do not manually label your submissions. We work hard to fair and fun contests, and ask for the same respect in return.

https://www.kaggle.com/c/dogs-vs-cats/data

Modelo de Teste 1

Imagens 50x50 + 5 hidden layers = ~307k parâmetros

```
Found 20000 images belonging to 2 classes.
Found 3000 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
```

```
inputs = tf.keras.Input(shape=(50,50,3))
x = layers.Conv2D(32, (3,3), activation=tf.nn.relu)(inputs)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(64, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(128, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(128, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Flatten()(x)
x = layers.Dense(512, activation=tf.nn.relu)(x)
outputs = layers.Dense(1, activation=tf.nn.sigmoid)(x)
model = tf.keras.Model(inputs, outputs)
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 50, 50, 3)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 48, 48, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 22, 22, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 9, 9, 128)         73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 2, 2, 128)         147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 1, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 512)               66048
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 307,393
Trainable params: 307,393
Non-trainable params: 0
_________________________________________________________________
```

Modelo de Teste 1

Imagens 50x50 + 5 hidden layers = ~307k parâmetros

CPU (23,3 m / média = 2,3 m)

```
Epoch 1/10
1000/1000 [==============================] - 124s 124ms/step - loss: 0.6582 - accuracy: 0.5967 - val_loss: 0.6106 - val_acc
uracy: 0.6508
Epoch 2/10
1000/1000 [==============================] - 141s 141ms/step - loss: 0.5312 - accuracy: 0.7351 - val_loss: 0.4815 - val_acc
uracy: 0.7614: 7s - los
Epoch 3/10
1000/1000 [==============================] - 148s 148ms/step - loss: 0.4523 - accuracy: 0.7875 - val_loss: 0.4706 - val_acc
uracy: 0.7870 20s - loss: 0.4569 - acc - ETA: 19s - ETA: 6s - l - ETA - ETA: 1s - loss: 0.4525 - accuracy: 0.78 - ETA: 1s -
loss:
Epoch 4/10
1000/1000 [==============================] - 142s 142ms/step - loss: 0.3952 - accuracy: 0.8186 - val_loss: 0.4223 - val_acc
uracy: 0.8066
Epoch 5/10
1000/1000 [==============================] - 141s 141ms/step - loss: 0.3446 - accuracy: 0.8479 - val_loss: 0.3985 - val_acc
uracy: 0.8200
Epoch 6/10
1000/1000 [==============================] - 147s 147ms/step - loss: 0.3015 - accuracy: 0.8698 - val_loss: 0.3794 - val_acc
uracy: 0.8298
Epoch 7/10
1000/1000 [==============================] - 153s 153ms/step - loss: 0.2629 - accuracy: 0.8881 - val_loss: 0.4080 - val_acc
uracy: 0.8208 - loss: 0.2629 - accuracy
Epoch 8/10
1000/1000 [==============================] - 134s 134ms/step - loss: 0.2258 - accuracy: 0.9043 - val_loss: 0.4318 - val_acc
uracy: 0.8276
Epoch 9/10
1000/1000 [==============================] - 133s 133ms/step - loss: 0.1879 - accuracy: 0.9226 - val_loss: 0.4441 - val_acc
uracy: 0.8198
Epoch 10/10
1000/1000 [==============================] - 136s 136ms/step - loss: 0.1602 - accuracy: 0.9347 - val_loss: 0.5188 - val_acc
uracy: 0.8178
```

GPU (13,3 m / média = 1,3 m)

```
Epoch 1/10
1000/1000 [==============================] - 103s 103ms/step - loss: 0.6902 - accuracy: 0.5270 - val_loss: 0.6769 - val_acc
uracy: 0.5726oss: 0.6933 - accuracy: 0.50 - ETA: 1:07 - loss: 0.6933 - accuracy - ETA: 1:07 - -  - ETA: 49s - loss: 0.6931
- accuracy:  - ETA: 49s - loss: 0.6931 - accurac - ETA: 48s - loss: 0.6931 - accuracy: 0.511 - ETA: 48s  - ETA: 40s - loss:
0.6931 - accuracy: 0.51 - ETA: 40s - ETA: 37s - loss: 0.6931 - accuracy - ETA: 36s - loss: 0.693 - ETA: 30s - loss: 0.6927
- acc - ETA: 29s - loss: 0.6928 - accu - ETA: 28s - loss: 0.6928 - ETA:  - ETA: 19s - loss: 0.6923 - accu - ETA: 18s - loss
: 0.6921 - accuracy: 0.515 - ETA: 18s - loss: 0.6921 - accuracy: 0.51 - ETA: 18s - loss: 0.6921 -  - ETA: 16s - loss: 0.691
8 - -  - ETA: 11s - loss: 0.6913 - accuracy - ETA: 10s - lo - ETA: 9s - loss: 0.6911 - accuracy: 0. - ETA: 8s - loss: 0.691
1 - accura - ETA: 8s - loss: 0.6910 - ac - ETA: 1s - loss: 0.6903 - accura - ETA: 1s -
Epoch 2/10
1000/1000 [==============================] - 94s 94ms/step - loss: 0.6374 - accuracy: 0.6356 - val_loss: 0.5945 - val_accur
acy: 0.6820
Epoch 3/10
1000/1000 [==============================] - 95s 95ms/step - loss: 0.5335 - accuracy: 0.7278 - val_loss: 0.4824 - val_accur
acy: 0.7668
Epoch 4/10
1000/1000 [==============================] - 94s 94ms/step - loss: 0.4644 - accuracy: 0.7770 - val_loss: 0.4646 - val_accur
acy: 0.7816 - loss: 0.4689 - accura - ETA:  - ETA: 4s - loss:
Epoch 5/10
1000/1000 [==============================] - 96s 96ms/step - loss: 0.4082 - accuracy: 0.8116 - val_loss: 0.4135 - val_accur
acy: 0.8090 - ETA: 56s - loss: 0.4039 - accuracy: 0.8 - ETA: 56s - loss: 0.4027 - accu - ETA: 55s - loss: 0.4040 - accur -
ETA: 51s - loss: 0.4041 - accuracy: 0.815  - ETA: - ETA: 34s - loss: 0.4061 - accuracy: - ET  - ETA: 26s - loss: 0.4069 - a
ccuracy: 0 - ETA: 3s - loss: 0.4 - ETA: 0s - loss: 0.4081 -
Epoch 6/10
1000/1000 [==============================] - 94s 94ms/step - loss: 0.3668 - accuracy: 0.8335 - val_loss: 0.4241 - val_accur
acy: 0.8054
Epoch 7/10
1000/1000 [==============================] - 95s 95ms/step - loss: 0.3266 - accuracy: 0.8572 - val_loss: 0.4540 - val_accur
acy: 0.7960
Epoch 8/10
1000/1000 [==============================] - 101s 101ms/step - loss: 0.2977 - accuracy: 0.8715 - val_loss: 0.4110 - val_acc
uracy: 0.8206- ac - ETA: 49s - loss: 0.2911 - accuracy:  - ETA: 48s - loss: 0.292 - ETA: 23s - loss: 0.2932  - ETA: 22s - l
o - - ETA: 15s - loss: 0.2943 - accuracy: - ETA: 15s - loss: 0.2 - ETA: 2s - loss: - ETA: 0s - loss: 0.297
Epoch 9/10
1000/1000 [==============================] - 103s 103ms/step - loss: 0.2586 - accuracy: 0.8913 - val_loss: 0.4106 - val_acc
uracy: 0.8112s - loss: 0.2433 - accuracy: 0.89 - ET - ETA: 47s - loss: 0.2489 - - ETA: 41s - loss - ETA: 39s - loss: 0.2471
- accur - ETA: 38s - loss: 0.2476 - ETA: 36s  - ETA:  - ETA: 11s - loss: 0. - ETA: 4s - loss: 0.2577 - accuracy: 0.89 - ETA
: 3s - loss: 0.2576 - ac - ETA: 3s -
Epoch 10/10
1000/1000 [==============================] - 104s 104ms/step - loss: 0.2267 - accuracy: 0.9050 - val_loss: 0.4588 - val_acc
uracy: 0.8262
```

Modelo de Teste 2

Modelo 1 + imagens 70x70 + aumento no número de neurônios = ~933k parâmetros

```
inputs = tf.keras.Input(shape=(70,70,3))
x = layers.Conv2D(64, (3,3), activation=tf.nn.relu)(inputs)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(64, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(128, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(256, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Flatten()(x)
x = layers.Dense(512, activation=tf.nn.relu)(x)
outputs = layers.Dense(1, activation=tf.nn.sigmoid)(x)
model = tf.keras.Model(inputs, outputs)
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 70, 70, 3)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 68, 68, 64)        1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 34, 34, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 14, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 5, 5, 256)         295168
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 512)               524800
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 933,057
Trainable params: 933,057
Non-trainable params: 0
```

Modelo de Teste 2

Modelo 1 + imagens 70x70 + aumento no número de neurônios = ~933k parâmetros

CPU (38,1 m / média = 3,8 m)

```
Epoch 1/10
1000/1000 [==============================] - 232s 232ms/step - loss: 0.6850 - accuracy: 0.5489 - val_loss: 0.6450 - val_acc
uracy: 0.6286
Epoch 2/10
1000/1000 [==============================] - 213s 213ms/step - loss: 0.6078 - accuracy: 0.6659 - val_loss: 0.5655 - val_acc
uracy: 0.7088
Epoch 3/10
1000/1000 [==============================] - 224s 224ms/step - loss: 0.5279 - accuracy: 0.7332 - val_loss: 0.4879 - val_acc
uracy: 0.7660
Epoch 4/10
1000/1000 [==============================] - 223s 223ms/step - loss: 0.4488 - accuracy: 0.7898 - val_loss: 0.4663 - val_acc
uracy: 0.7778
Epoch 5/10
1000/1000 [==============================] - 227s 227ms/step - loss: 0.3893 - accuracy: 0.8251 - val_loss: 0.4252 - val_acc
uracy: 0.8086
Epoch 6/10
1000/1000 [==============================] - 240s 240ms/step - loss: 0.3367 - accuracy: 0.8494 - val_loss: 0.3690 - val_acc
uracy: 0.8322
Epoch 7/10
1000/1000 [==============================] - 226s 226ms/step - loss: 0.2923 - accuracy: 0.8727 - val_loss: 0.3727 - val_acc
uracy: 0.8408
Epoch 8/10
1000/1000 [==============================] - 232s 232ms/step - loss: 0.2479 - accuracy: 0.8949 - val_loss: 0.3813 - val_acc
uracy: 0.8472
Epoch 9/10
1000/1000 [==============================] - 230s 230ms/step - loss: 0.2137 - accuracy: 0.9132 - val_loss: 0.3933 - val_acc
uracy: 0.8454
Epoch 10/10
1000/1000 [==============================] - 243s 243ms/step - loss: 0.1800 - accuracy: 0.9251 - val_loss: 0.4014 - val_acc
uracy: 0.8430
```

GPU (18,1 m / média = 1,8 m)

```
Epoch 1/10
1000/1000 [==============================] - 120s 120ms/step - loss: 0.6929 - accuracy: 0.5132 - val_loss: 0.6921 - val_acc
uracy: 0.5172
Epoch 2/10
1000/1000 [==============================] - 101s 101ms/step - loss: 0.6782 - accuracy: 0.5598 - val_loss: 0.6266 - val_acc
uracy: 0.6476
Epoch 3/10
1000/1000 [==============================] - 103s 103ms/step - loss: 0.5884 - accuracy: 0.6855 - val_loss: 0.5194 - val_acc
uracy: 0.7422
Epoch 4/10
1000/1000 [==============================] - 100s 100ms/step - loss: 0.5014 - accuracy: 0.7555 - val_loss: 0.4859 - val_acc
uracy: 0.7594
Epoch 5/10
1000/1000 [==============================] - 124s 124ms/step - loss: 0.4510 - accuracy: 0.7886 - val_loss: 0.4898 - val_acc
uracy: 0.7590
Epoch 6/10
1000/1000 [==============================] - 114s 114ms/step - loss: 0.4096 - accuracy: 0.8119 - val_loss: 0.4447 - val_acc
uracy: 0.7992
Epoch 7/10
1000/1000 [==============================] - 109s 109ms/step - loss: 0.3674 - accuracy: 0.8346 - val_loss: 0.4511 - val_acc
uracy: 0.7996TA: 3s
Epoch 8/10
1000/1000 [==============================] - 115s 115ms/step - loss: 0.3265 - accuracy: 0.8563 - val_loss: 0.4382 - val_acc
uracy: 0.8046
Epoch 9/10
1000/1000 [==============================] - 103s 103ms/step - loss: 0.2852 - accuracy: 0.8752 - val_loss: 0.4964 - val_acc
uracy: 0.7964c - ETA: 26s - l - ETA: 0s - loss: 0.286
Epoch 10/10
1000/1000 [==============================] - 102s 102ms/step - loss: 0.2442 - accuracy: 0.8955 - val_loss: 0.5044 - val_acc
uracy: 0.7974
```

Modelo de Teste 3

Modelo 2 + imagens 120x120, aumento no número de hidden layers = ~1,8M parâmetros

```
inputs = tf.keras.Input(shape=(120,120,3))
x = layers.Conv2D(32, (3,3), activation=tf.nn.relu)(inputs)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(64, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(128, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(256, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Conv2D(512, (3,3), activation=tf.nn.relu)(x)
x = layers.MaxPooling2D((2,2))(x)
x = layers.Flatten()(x)
x = layers.Dense(512, activation=tf.nn.relu)(x)
outputs = layers.Dense(1, activation=tf.nn.sigmoid)(x)
model = tf.keras.Model(inputs, outputs)
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 120, 120, 3)]     0
_________________________________________________________________
conv2d (Conv2D)              (None, 118, 118, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 59, 59, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 57, 57, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 28, 28, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 26, 26, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 13, 13, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 11, 11, 256)       295168
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 256)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 3, 512)         1180160
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 1, 1, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 512)               262656
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 1,831,745
Trainable params: 1,831,745
Non-trainable params: 0
_________________________________________________________________
```

Modelo de Teste 3

Modelo 2 + imagens 120x120, aumento no número de hidden layers = ~1,8M parâmetros

CPU (92 m / média = 9,2 m)

```
Epoch 1/10
1000/1000 [==============================] - 568s 568ms/step - loss: 0.6578 - accuracy: 0.6019 - val_loss: 0.6035 - val_acc
uracy: 0.6644
Epoch 2/10
1000/1000 [==============================] - 560s 560ms/step - loss: 0.5343 - accuracy: 0.7301 - val_loss: 0.4691 - val_acc
uracy: 0.7728
Epoch 3/10
1000/1000 [==============================] - 595s 595ms/step - loss: 0.4270 - accuracy: 0.7999 - val_loss: 0.4319 - val_acc
uracy: 0.7880
Epoch 4/10
1000/1000 [==============================] - 535s 535ms/step - loss: 0.3427 - accuracy: 0.8490 - val_loss: 0.3513 - val_acc
uracy: 0.8426
Epoch 5/10
1000/1000 [==============================] - 545s 545ms/step - loss: 0.2874 - accuracy: 0.8747 - val_loss: 0.3541 - val_acc
uracy: 0.8560
Epoch 6/10
1000/1000 [==============================] - 530s 530ms/step - loss: 0.2432 - accuracy: 0.8971 - val_loss: 0.3169 - val_acc
uracy: 0.8598
Epoch 7/10
1000/1000 [==============================] - 535s 535ms/step - loss: 0.2071 - accuracy: 0.9115 - val_loss: 0.3137 - val_acc
uracy: 0.8612
Epoch 8/10
1000/1000 [==============================] - 556s 556ms/step - loss: 0.1787 - accuracy: 0.9263 - val_loss: 0.3087 - val_acc
uracy: 0.8796
Epoch 9/10
1000/1000 [==============================] - 668s 668ms/step - loss: 0.1468 - accuracy: 0.9397 - val_loss: 0.3454 - val_acc
uracy: 0.8774
Epoch 10/10
1000/1000 [==============================] - 455s 455ms/step - loss: 0.1249 - accuracy: 0.9488 - val_loss: 0.3759 - val_acc
uracy: 0.8720
```

GPU (21,8 m / média = 2,1 m)

```
Epoch 1/10
1000/1000 [==============================] - 130s 130ms/step - loss: 0.6921 - accuracy: 0.5121 - val_loss: 0.6815 - val_acc
uracy: 0.5556s - loss: 0.693 - E - ETA: 33s - loss: 0.6 - ETA: 31s - loss - ETA: 28s - loss: 0.6935 - accuracy: 0.50 - ETA:
28s - loss: 0.69
Epoch 2/10
1000/1000 [==============================] - 115s 115ms/step - loss: 0.6306 - accuracy: 0.6336 - val_loss: 0.5688 - val_acc
uracy: 0.6960s: 0 - ETA: 1:06 - loss: 0.6 - ETA: 1:03 - loss: 0.6616 -  - ETA: 1:02 - loss: 0.661 -
Epoch 3/10
1000/1000 [==============================] - 123s 123ms/step - loss: 0.4943 - accuracy: 0.7572 - val_loss: 0.4803 - val_acc
uracy: 0.7858
Epoch 4/10
1000/1000 [==============================] - 125s 125ms/step - loss: 0.4025 - accuracy: 0.8124 - val_loss: 0.3845 - val_acc
uracy: 0.8242ETA: 13s - loss: 0.4064 - ac - ETA: 12s - loss: 0.4058 - accuracy: - ETA: 11s - ETA: 6s - loss: 0.4050 - accur
acy: 0. - ETA: 3s - loss: 0.4038 - ac - ETA: 2s - loss: 0.4034 - accura - ETA: 2s - loss: 0.4034 - accu - ETA: 1s - l
Epoch 5/10
1000/1000 [==============================] - 130s 130ms/step - loss: 0.3245 - accuracy: 0.8559 - val_loss: 0.3733 - val_acc
uracy: 0.8338
Epoch 6/10
1000/1000 [==============================] - 119s 119ms/step - loss: 0.2668 - accuracy: 0.8842 - val_loss: 0.3373 - val_acc
uracy: 0.8574
Epoch 7/10
1000/1000 [==============================] - 129s 129ms/step - loss: 0.2237 - accuracy: 0.9042 - val_loss: 0.2902 - val_acc
uracy: 0.8810acy: 0.907 - ETA: 56s - loss: 0.2174 - accuracy:  - ETA: 55s - loss: 0.2184 - accu - ETA: 54s - loss: 0.2191 -
accuracy: 0.90 - ETA: 54s - lo - ETA: 47s - loss: 0.2212  - ETA: 41s -  - ETA: 5s - loss: 0.2231 -  - ETA:  - ETA: 0s - los
s: 0.2237 - accuracy
Epoch 8/10
1000/1000 [==============================] - 197s 197ms/step - loss: 0.1884 - accuracy: 0.9221 - val_loss: 0.3147 - val_acc
uracy: 0.8754s: 0.1700 - accuracy: 0. - ETA: 2:26 - loss: 0.1746 - accuracy - ETA: 2:27 - loss: 0.1715 - accuracy: 0.92 - E
TA: 2:27 - loss: 0.1738 -  - ETA: 2:28 - loss: - ETA: 2:25 - loss: 0.1807  - ETA: 2:24 - loss: 0.1813 -  - ETA: 2:23 - loss
: - ETA: 2:20 - los - ETA: 2:18 - loss: 0.178 - ETA: 2:17 - loss: 0.1783 - accuracy: 0. - ETA: 2:17 - loss: 0.1779 - accura
cy: 0.92 - ETA: 2:17 - loss: 0.1786 - accuracy: 0. - ETA: 2:16 - loss: 0.1803 - accuracy:  - ETA: 1:21 - loss: 0.1785 - acc
uracy:  - ETA: 1:21 - los - ETA: 1:18 - loss: 0.1798 - accuracy: 0.92 - ETA: 1:18 - l - ETA: 1:07 - loss: 0.1801 - accura -
- ETA: 57s - loss: 0.1844 - accurac - ETA: 55s - loss: 0.1843  - ETA: 52  - ETA: 4s - loss: 0.1889 - accuracy:  - ETA: 3s -
loss: 0.1892 - accura - ETA: 2s - l
Epoch 9/10
1000/1000 [==============================] - 127s 127ms/step - loss: 0.1523 - accuracy: 0.9368 - val_loss: 0.3308 - val_acc
uracy: 0.8734
Epoch 10/10
1000/1000 [==============================] - 118s 118ms/step - loss: 0.1325 - accuracy: 0.9452 - val_loss: 0.3965 - val_acc
uracy: 0.8732
```

.. e viveu feliz ~~para sempre!~~ até agora!

	1 época *		10 épocas *		
	GPU	CPU	GPU	CPU	
Modelo 1 (300k)	1,3	2,3	13,3	23,3	***+50%***
Modelo 2 (900k)	1,8	3,8	18,1	38,1	***+100%***
Modelo 3 (1,8M)	2,1	9,2	21,8	92	***+300%***

* Valores em minutos

USE THE RIGHT TOOL FOR THE JOB


ORACLE
Cloud
Visite nosso Estande
Faça um Trial
Ganhe um Brinde
Faça um Hands on
Ganhe outro Brinde
Obrigado!
Waslley Souza
Blog: waslleysouza.com.br
LinkedIn: waslleysouza
Github.com/waslleysouza
waslley.souza@oracle.com