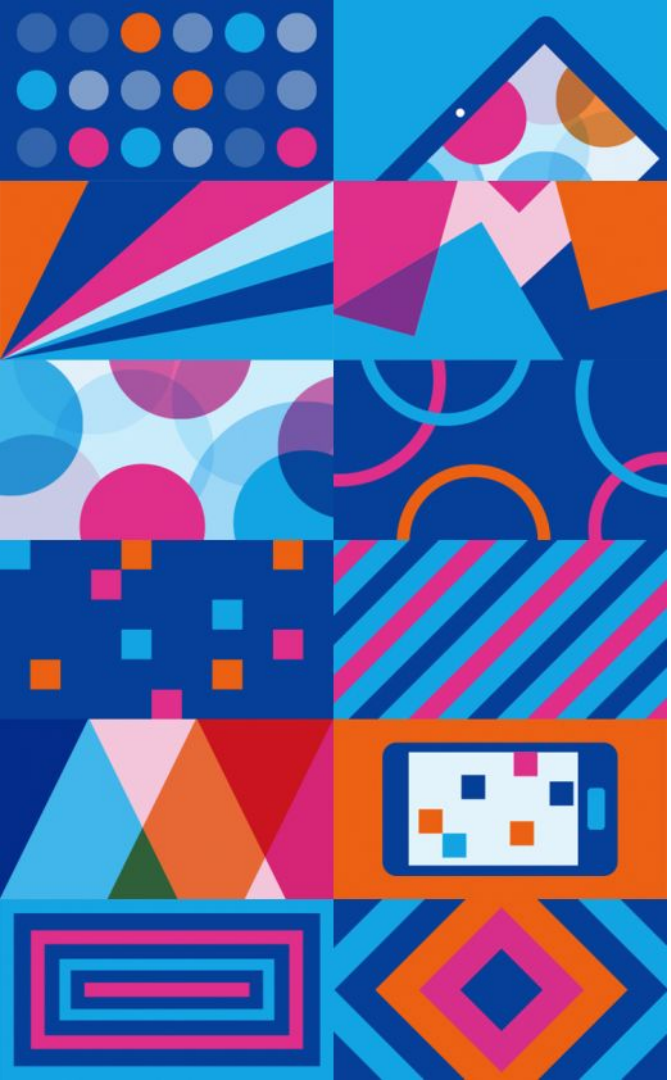


MEETUP PYDATA - 2019

MANAUS/AM

DEEP KNN

LUCAS FROTA

OCEAN | SAMSUNG | UEA UNIVERSIDADE DO ESTADO DO AMAZONAS

O QUE É DEEP LEARNING?

Copyright © Samsung Electronics 2017. All Rights Reserved

VAMOS IMAGINAR

Copyright © Samsung Electronics 2017. All Rights Reserved

PORÉM...

- Você tem poucos dados

Copyright © Samsung Electronics 2017. All Rights Reserved

PORÉM...

- Você tem poucos dados
- Não tem um supercomputador

Copyright © Samsung Electronics 2017. All Rights Reserved

PORÉM...

- Você tem poucos dados
- Não tem um supercomputador
- Nem dinheiro pra nuvem

Copyright © Samsung Electronics 2017. All Rights Reserved

PORÉM...

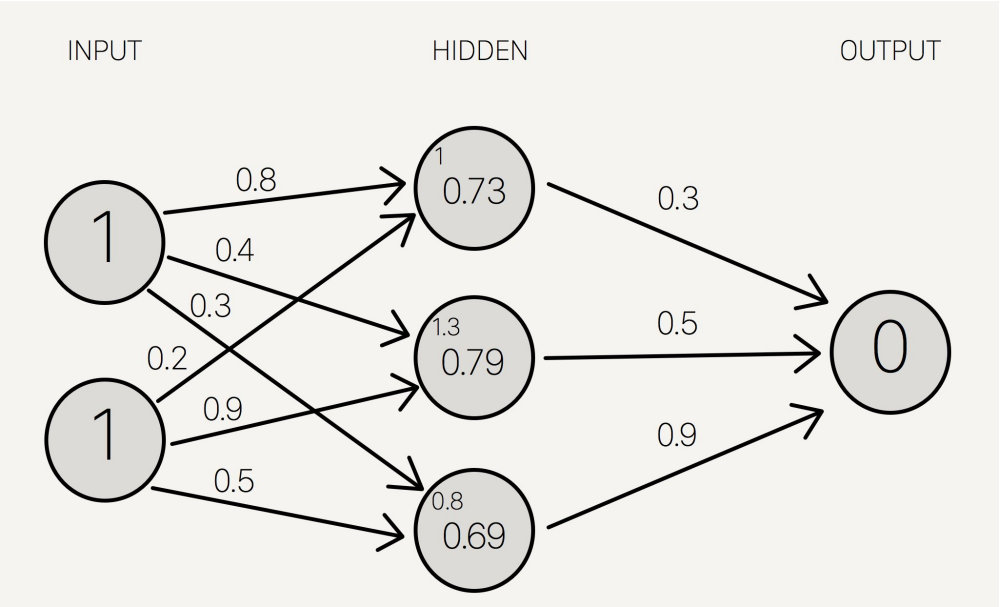
- Você tem poucos dados
- Não tem um supercomputador
- Nem dinheiro pra nuvem

Copyright © Samsung Electronics 2017. All Rights Reserved

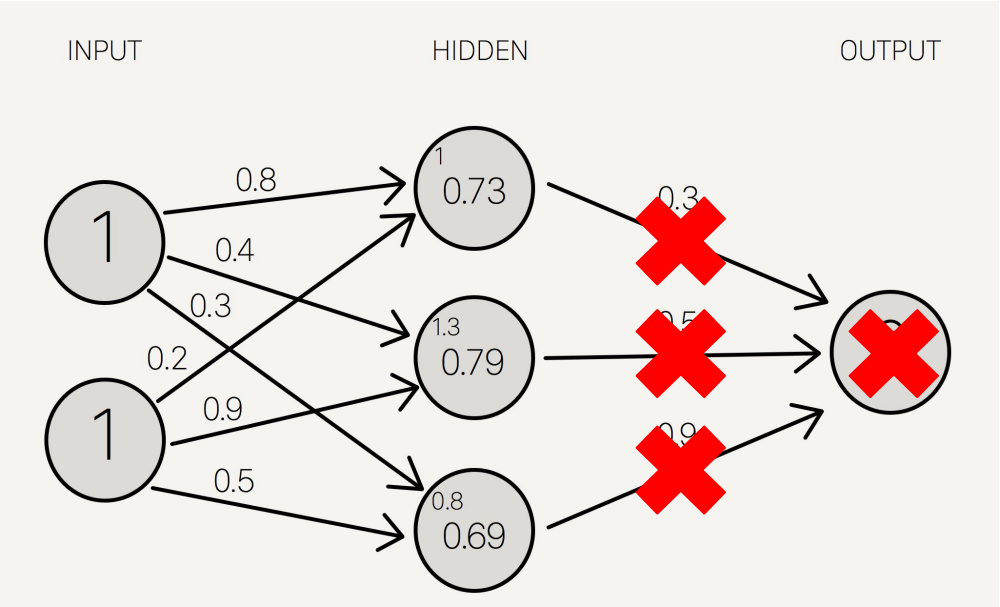
E SE?

Copyright © Samsung Electronics 2017. All Rights Reserved

MAS COMO?

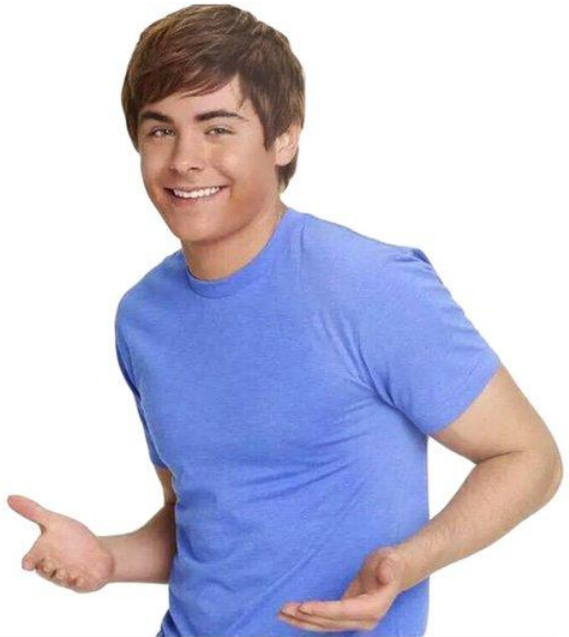

INPUT
HIDDEN
OUTPUT
1
1
0.8
0.4
0.3
0.2
0.9
0.5
1
0.73
1.3
0.79
0.8
0.69
0.3
0.5
0.9
0

MAS COMO?



MAS QUAL?

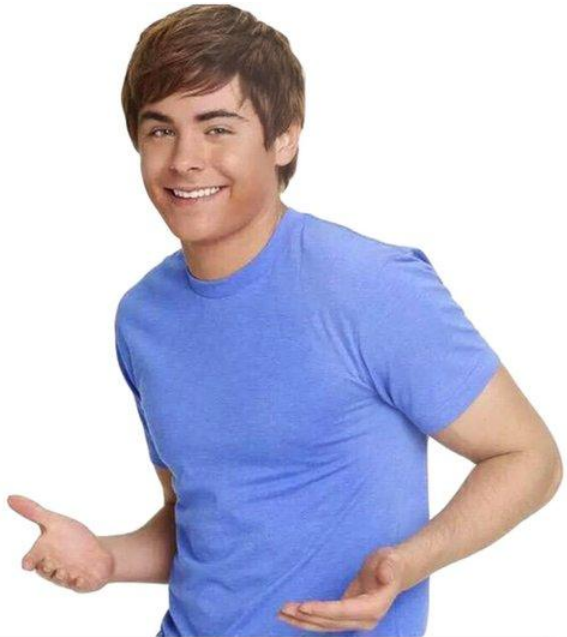
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-



Copyright © Samsung Electronics 2017. All Rights Reserved

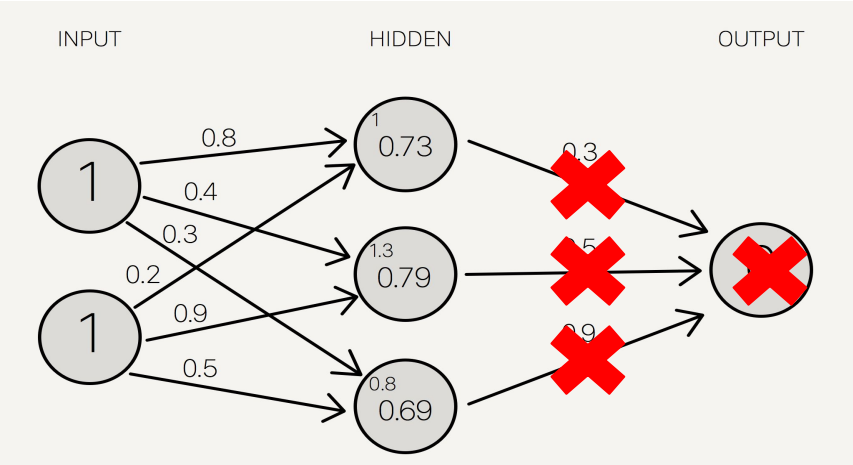
MAS QUAL?

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-



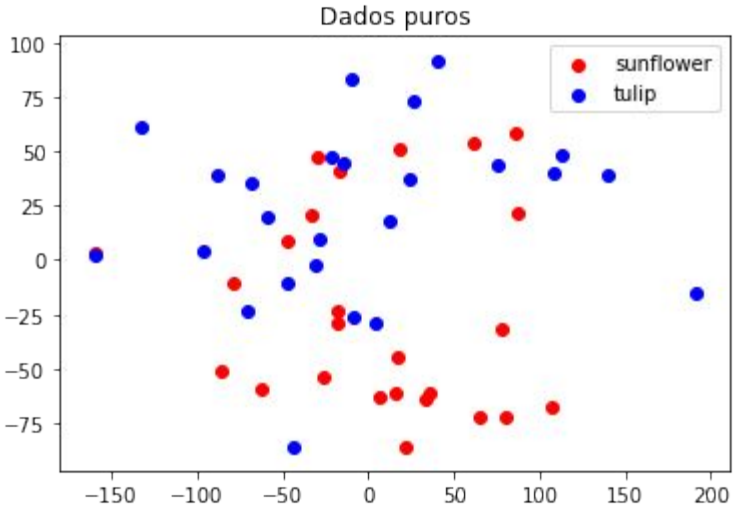
Copyright © Samsung Electronics 2017. All Rights Reserved

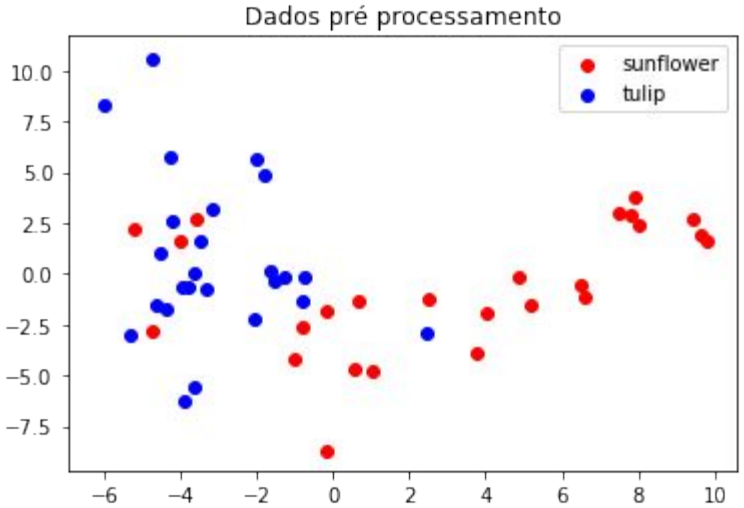
E COMO RETREINA ISSO??



Copyright © Samsung Electronics 2017. All Rights Reserved

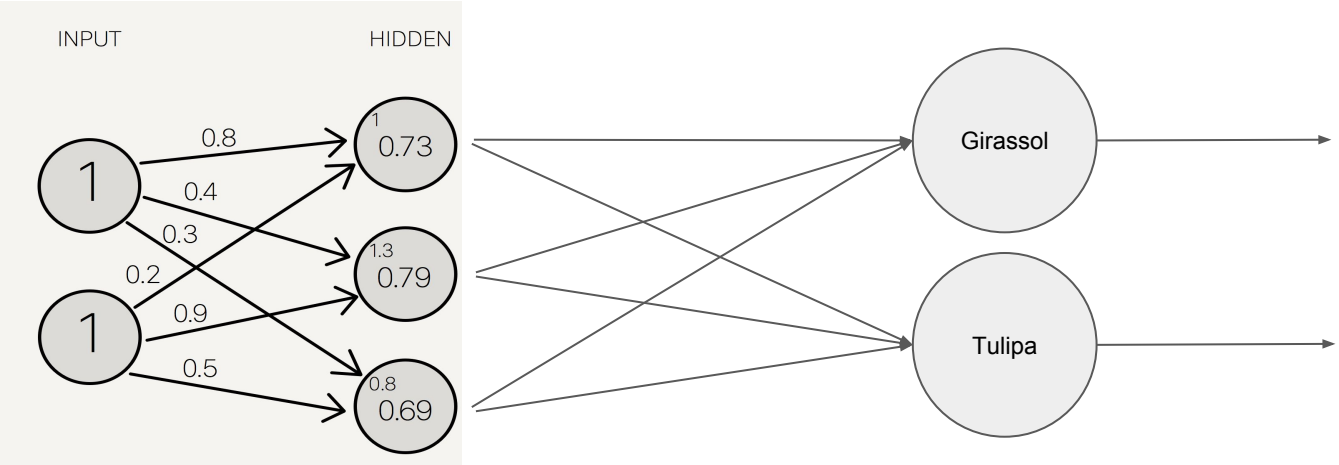
E POR QUE???





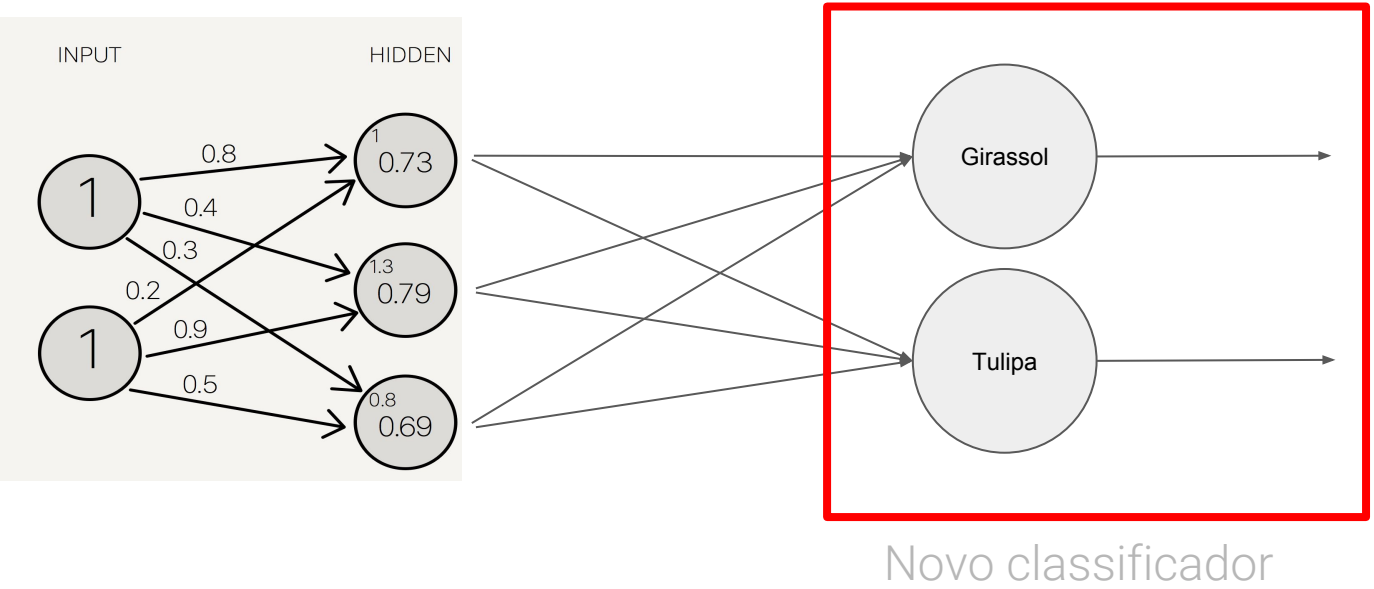
Copyright © Samsung Electronics 2017. All Rights Reserved

E COMO RETREINA ISSO??



Copyright © Samsung Electronics 2017. All Rights Reserved

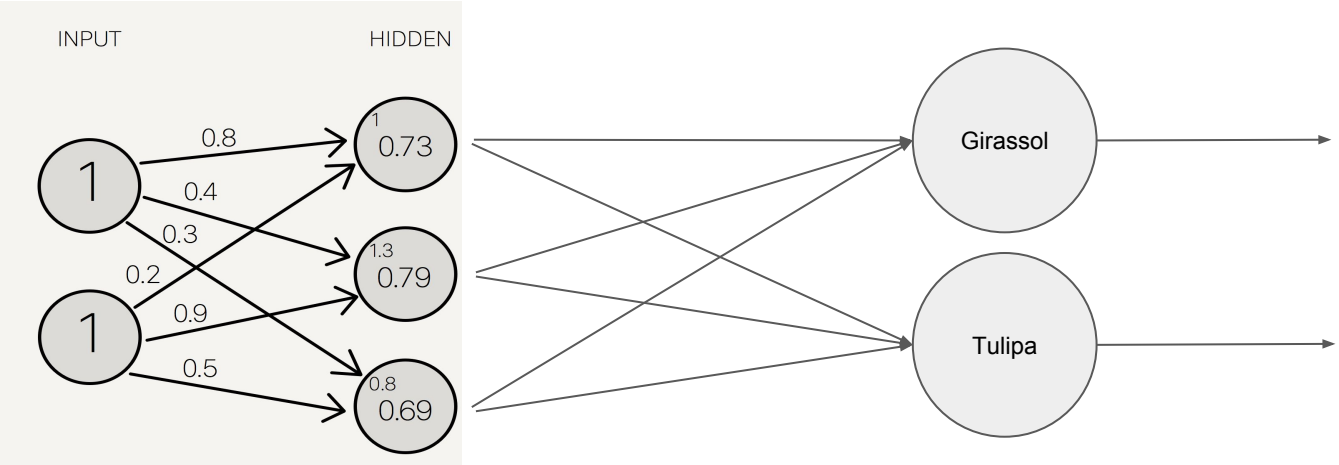
E COMO RETREINA ISSO??



Novo classificador

Copyright © Samsung Electronics 2017. All Rights Reserved

E COMO RETREINA ISSO??

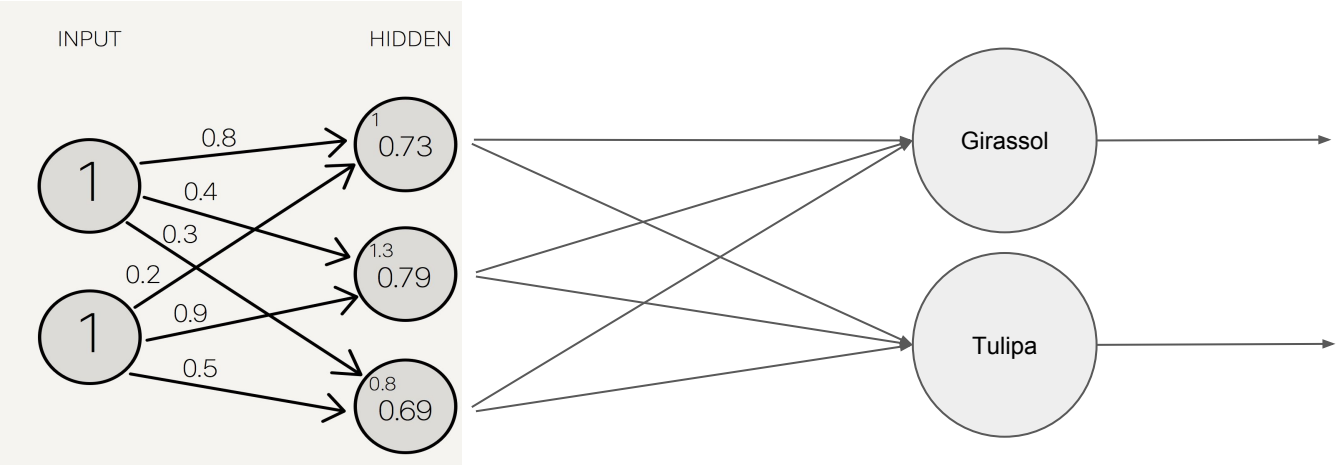


FAZ PREVISÃO

ATUALIZA OS PESOS

Copyright © Samsung Electronics 2017. All Rights Reserved

E COMO RETREINA ISSO??



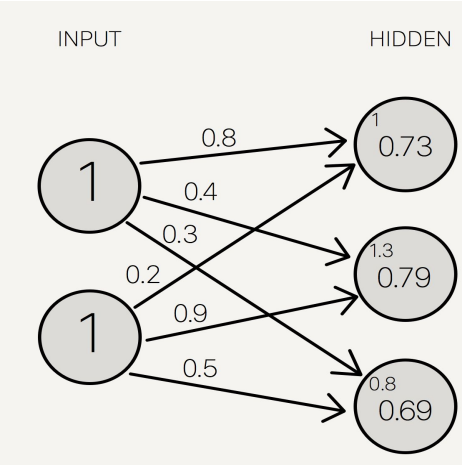
FAZ PREVISÃO

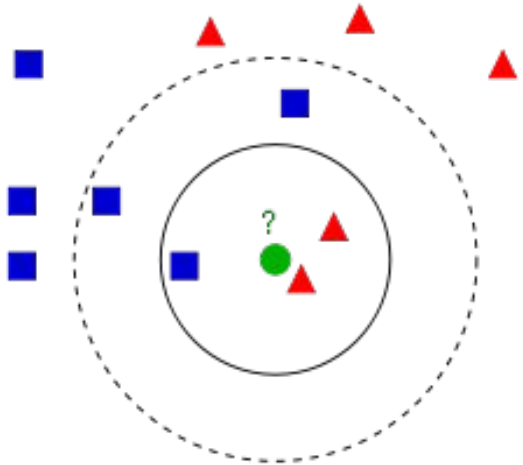
CAMADAS CONGELADAS

ATUALIZA OS PESOS

Copyright © Samsung Electronics 2017. All Rights Reserved

E COMO RETREINA ISSO??





FAZ PREVISÃO

TREINAR O KNN

Copyright © Samsung Electronics 2017. All Rights Reserved

E OS DADOS??

Copyright © Samsung Electronics 2017. All Rights Reserved

E OS DADOS??

COM LICENÇA SERÁ QUE POR ACASO VOCÊ NÃO TERIA UM DATASET DE FOTOS COM UMAS 100 MIL FOTOS DE CADA CLASSE O RGANIZADAS EM PASTAS COM OS NOMES DAS CLASSES?

Copyright © Samsung Electronics 2017. All Rights Reserved

E OS DADOS??

- 25 imagens de cada classe

Copyright © Samsung Electronics 2017. All Rights Reserved

OK VAMOS AO CÓDIGO

Copyright © Samsung Electronics 2017. All Rights Reserved

REDE NEURAL

Copyright © Samsung Electronics 2017. All Rights Reserved

HORA DE CORTAR O MODELO(RN)

```
comp_base_model = iv3.InceptionV3(weights='imagenet', include_top=False)
```

```
x = comp_base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(64, activation='relu')(x)
predictions = Dense(2, activation='softmax')(x)

comp_model = Model(inputs=comp_base_model.input, outputs=predictions)
```

```
for layer in comp_base_model.layers:
    layer.trainable = False
```

```
comp_model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
```

Copyright © Samsung Electronics 2017. All Rights Reserved

AGORA VAMOS TREINAR(RN)

```
%%time
comp_model.fit(X_train, raw_y_train, epochs=30)
```

```
50/50 [==============================] - 16s 315ms/step - loss: 0.0863 - acc: 0.9800
Epoch 23/30
50/50 [==============================] - 16s 315ms/step - loss: 0.1095 - acc: 0.9800
Epoch 24/30
50/50 [==============================] - 16s 314ms/step - loss: 0.0234 - acc: 1.0000
Epoch 25/30
50/50 [==============================] - 16s 315ms/step - loss: 0.0355 - acc: 1.0000
Epoch 26/30
50/50 [==============================] - 16s 318ms/step - loss: 0.0406 - acc: 1.0000
Epoch 27/30
50/50 [==============================] - 16s 318ms/step - loss: 0.0340 - acc: 1.0000
Epoch 28/30
50/50 [==============================] - 16s 317ms/step - loss: 0.0195 - acc: 1.0000
Epoch 29/30
50/50 [==============================] - 16s 317ms/step - loss: 0.0183 - acc: 1.0000
Epoch 30/30
50/50 [==============================] - 16s 317ms/step - loss: 0.0133 - acc: 1.0000
Wall time: 7min 52s

<keras.callbacks.History at 0x23c334e16d8>
```

Copyright © Samsung Electronics 2017. All Rights Reserved

AGORA VAMOS TREINAR(RN)

```
%%time
comp_model.fit(X_train, raw_y_train, epochs=30)
```

```
50/50 [==============================] - 16s 315ms/step - loss: 0.0863 - acc: 0.9800
Epoch 23/30
50/50 [==============================] - 16s 315ms/step - loss: 0.1095 - acc: 0.9800
Epoch 24/30
50/50 [==============================] - 16s 314ms/step - loss: 0.0234 - acc: 1.0000
Epoch 25/30
50/50 [==============================] - 16s 315ms/step - loss: 0.0355 - acc: 1.0000
Epoch 26/30
50/50 [==============================] - 16s 318ms/step - loss: 0.0406 - acc: 1.0000
Epoch 27/30
50/50 [==============================] - 16s 318ms/step - loss: 0.0340 - acc: 1.0000
Epoch 28/30
50/50 [==============================] - 16s 317ms/step - loss: 0.0195 - acc: 1.0000
Epoch 29/30
50/50 [==============================] - 16s 317ms/step - loss: 0.0183 - acc: 1.0000
Epoch 30/30
50/50 [==============================] - 16s 317ms/step - loss: 0.0133 - acc: 1.0000
Wall time: 7min 52s

<keras.callbacks.History at 0x23c334e16d8>
```

7 minutos e 52 segundos de treino

Copyright © Samsung Electronics 2017. All Rights Reserved

DEEP KNN

Copyright © Samsung Electronics 2017. All Rights Reserved

HORA DE CORTAR O MODELO(DEEPKNN)

```
1 base_model = iv3.InceptionV3(weights='imagenet')
```

```
1 deep_featuring_model = Model(inputs=base_model.input,
2                               outputs=base_model.get_layer('avg_pool').output)
```

Copyright © Samsung Electronics 2017. All Rights Reserved

AGORA VAMOS TREINAR(DEEPKNN)

```
%%time
features = deep_featuring_model.predict(X_train)
```

```
Wall time: 26.9 s
```

```
%%time
clf = KNeighborsClassifier(5)
clf.fit(features, y_train)
```

```
Wall time: 0 ns
```

Copyright © Samsung Electronics 2017. All Rights Reserved

AGORA VAMOS TREINAR(DEEPKNN)

```
%%time
features = deep_featuring_model.predict(X_train)
```

```
Wall time: 26.9 s
```

```
%%time
clf = KNeighborsClassifier(5)
clf.fit(features, y_train)
```

```
Wall time: 0 ns
```

27 segundos de treino

Copyright © Samsung Electronics 2017. All Rights Reserved

HORA DOS RESULTADOS

Copyright © Samsung Electronics 2017. All Rights Reserved

HORA DOS RESULTADOS

- 400 imagens de cada classe

```
%%time
features_test = deep_featuring_model.predict(X_test)
```

```
Wall time: 5min 48s
```

```
pred_y = clf.predict(features_test)
```

```
accuracy_score(y_test, pred_y)
```

```
0.8425
```

```
confusion_matrix(y_test, pred_y)
```

```
array([[324,  76],
       [ 50, 350]], dtype=int64)
```

```
%%timeit
clf.predict(deep_featuring_model.predict(single_image))
```

```
473 ms ± 4.74 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
%%time
comp_pred_y = comp_model.predict(X_test)
```

```
Wall time: 6min 9s
```

```
accuracy_score(new_comp_pred_y, y_test)
```

```
0.71625
```

```
confusion_matrix(new_comp_pred_y, y_test)
```

```
array([[391, 218],
       [  9, 182]], dtype=int64)
```

```
%%timeit
comp_model.predict(single_image)
```

```
667 ms ± 93.4 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

Copyright © Samsung Electronics 2017. All Rights Reserved

OCEAN | SAMSUNG | UEA UNIVERSIDADE DO ESTADO DO AMAZONAS

QUEM SOU EU ?

LUCAS FROTA
DESENVOLVEDOR
ANDROID

Email: lucv.frota@gmail.com
Linkedin: bit.ly/LucasFrotaLinkedin
Github: @Lucasfrota
Twitter: @LuscasFrota
Link do projeto: http://bit.ly/deepKnn

Copyright © Samsung Electronics 2017. All Rights Reserved