

Clasificación de Cactáceas

Universidad Nacional Autónoma de México

Facultad de Ciencias

REDES NEURONALES

González Aguila Elias¹

Eliasgonzalezaguila08@ciencias.unam.mx¹

Junio 6, 2022.

Resumen

Las cactáceas son unas plantas endémicas del continente americano, con cerca de mil 400, México alberga cerca de 670 especies de las cuales cerca de 520 son endémicas del país. El grupo Cactácea es el más extenso, el objetivo de este artículo es crear una arquitectura de red neuronal para la clasificación de algunas de estas especies, seleccionamos 6 especies y creamos una red neuronal convolucional para clasificar estas imágenes y vimos su precisión y pérdida así como otros factores, después proponemos una técnica de aumento de datos para algunas clases y comparamos con el modelo inicial y notamos las diferencias en la predicción de imágenes llegando hasta a un 77 por ciento de predicción.

Carpeta en GB

<https://github.com/EliasGonzalezAguila/RedesNeuronales.git>

1. Introducci3n

Cact3ceas

La familia Cact3cea agrupa a una gran diversidad de plantas de formas globosas, cilndricas y esf3ricas. La familia es originaria del continente americano, constituida por cerca de 2000 especies, distribuidas desde el norte de Canad3 hasta la Patagonia, y desde el nivel del mar, en dunas costeras, hasta 5100 msnm. Las cact3ceas viven en diversos ecosistemas, incluyendo las selvas tropicales, pero la mayor3a de ellas se desarrollan en ambientes 3ridos y semi3ridos.

En M3xico las zonas secas abarcan cerca del 60 de la superficie del pa3s en d3nde viven principalmente estos ejemplares. Que se encuentran en 4 grandes grupos, que son: *Pereskioideae*, especies que tienen hojas bien formadas, *Opuntioideae*, especies con tallos en forma ovalada como una raqueta de tenis, d3nde se encuentran por ejemplo, los nopales, *Maihuenioideae* especies end3micas de Argentina y Chile y *Cactoideae* d3nde se encuentran la mayor3a de especies del pa3s.

Todos estos grupos son de gran importancia ecol3gica ya que cumplen funciones importantes dentro de sus ecosistemas, indicadores del bienestar de los ecosistemas, fuentes de alimentos para animales y humanos, algunas llegan a estar presentes en la cultura de ciertas regiones.

Familia Cactoideae

La familia Cactoideae es una de las familias m3s extensas entre las plantas con flores, la cual se puede reconocer por una combinaci3n de caracter3sticas propias como son la presencia de tallos suculentos de forma globosa o columnar, generalmente verdes, una tendencia a no presentar hojas, y tallos con espinas de diversas formas, ta-

ma3os y colores. Para la clasificaci3n de cact3ceas son centraremos principalmente en esta familia por sus distintivas caracter3sticas antes mencionadas. Las especies seleccionadas son:

- *Astrophytum Asterias* (1)
- *Lophophora Williamsii* (2)
- *Mammillaria Bocasana* (3)
- *Mammillaria Elongata* (4)
- *Pelecyphora Aselliformis* (5)
- *Turbincarpus Valdezianus* (6)

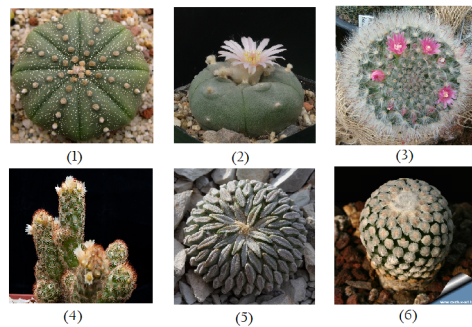


Figura 1: Especies

Ya que cuentan con una variedad de formas distintas entre s3, y para poner una mayor presi3n al sistema de aprendizaje escogimos algunos ej3mples que se parecen entre s3, como lo son: *Pelecyphora Aselliformis* y *Turbincarpus Valdezianus*.

Problemas Generales

Los principales problemas que enfrentan las cact3ceas actualmente son, el cambio de usos del suelo provoca que los ambientes naturales sean completamente transformados, en 3reas agr3colas, ganaderas o urbanizaci3n provocando la perdida de espacio y poblaciones de especies. La introducci3n de especies ex3ticas que compiten por los

recursos. La colecta directa para forraje para el ganado, y para la obtención de alimento. Algunas especies son buscadas para obtener materias primas, Algunas otras son muy apreciadas por los coleccionistas y son buscadas por su rareza, generando un tráfico ilegal, lo que ha llevado a ponerlas en riesgo. En la actualidad se encuentran numerosas especies de la familia colocadas dentro de la Norma Oficial Mexicana NOM-059-SEMARNAT-2001 de especies de flora y fauna silvestres en México en categorías de riesgo.

Cultivo de cactáceas

El cultivo de estas especies puede funcionar como una manera de reducir algunos problemas planteados anteriormente, el uso ornamental de estas especies está muy presente en México y el mundo. El cultivo a gran escala es algo común de ver en nuestro país y países como Tailandia que cuentan con una gran variedad de especies híbridadas y especies puras genéticamente.

Por lo que tener una manera sofisticada de clasificar rápidamente algunas especies para personas que no cuenten con una experiencia previa en el cultivo de cactáceas es de suma importancia. Por lo que se propone un método usando redes neuronales para clasificar las especies anterior mencionadas.

2. Objetivos

El objetivo principal de este artículo es crear un modelo de red neuronal para clasificación de cactáceas. Divida en los siguientes objetivos particulares.

- Base de imaganes
- Pre procesamiento de imaganes
- Arquitectura de red

- Entrenamiento modelo
- Aumento de datos.
- Aumento de datos proporcional.

3. Metodología

Base de imaganes

Para realizar poder hacer la red neuronal necesitamos una base de imágenes de las especies, estas imágenes fueron obtenidas de nuestra base de datos de los cultivos que tenemos y de una carpeta en Drive publica y otras de internet con un total de especies de:

Especies	imágenes
Astropytum Asterias	181
Lophophora Williamsii	298
Mammillaria Bocasana	204
Mammillaria Elongata	192
Pelecypora Aselliformis	184
Turbiniacarpus Valdezianus	223
Total de imágenes	1282

Cuadro 1: Total de imágenes

Pre procesamiento de imágenes

Una vez teniendo las bases de datos de las imágenes por carpeta de cada especie, procedemos a renombrarlas para su mejor manejo y cargarlas al entorno de trabajo. Las imágenes por su naturaleza cuentan con un tamaño distinto entre ellas, por lo que necesitamos que todas las imágenes estén en un mismo tamaño a la hora de entrenar y las redimensionamos a (300, 300). Además, notamos que algunas pocas imágenes cuentan con filtros es decir cuentan con un (300, 300, 4) y todas deben contener las dimensiones (300, 300, 3) para que no tengamos errores, por lo que procedemos a identificar estas imágenes no RGB con un algoritmo y las eliminamos disminuyendo muy poco el data-

set.

```
img_total=img1+img2+img3+img4
print('Numero total de imagenes:', len(img_total), 'Dimensiones de las imagenes:', img_total[0].shape)

et_1=[0]*len(img1) #Agregamos una lista de longitud de las imagenes de los tipos de rocas
et_2=[1]*len(img2)
et_3=[2]*len(img3)
et_4=[3]*len(img4)
et_5=[4]*len(img5)
et_6=[5]*len(img6)
etiquetas=[et_1+et_2+et_3+et_4+et_5+et_6] #Creamos las etiquetas
print('Numero total de etiquetas:', len(etiquetas), 'tipos de etiquetas:', list(dict.fromkeys(etiquetas)),
      'etiquetas vs img:', len(etiquetas)-len(img_total))

Numero total de imagenes: 1200 Dimensiones de las imagenes: (300, 300, 3)
Numero total de etiquetas: 1200 tipos de etiquetas: [0, 1, 2, 3, 4, 5] etiquetas vs img: True
```

Figura 2: imaganes procesadas

Las matrices de imágenes son introducidas en una lista y se les asocia un diccionario con el nombre de cada especie, para su posterior entrenamiento. Las matrices de imágenes son introducidas en una lista y se les asocia un Con la función *train test split* creamos nuestros conjuntos de entrenamiento y prueba para posteriormente ingresarlos en el modelo de IA.

```
from sklearn.model_selection import train_test_split # creamos los conjuntos de test y entrenamiento
#corta el 20% de imagenes de test y almacenamos al azar el test y prueba
x_entre,x_test,y_entre,y_test = train_test_split(x,y,test_size=0.25)

print('dimensiones datos de entrenamiento: ', x_entre.shape, y_entre.shape)
print('dimensiones de datos de test: ', x_test.shape, y_test.shape)

#Normalizamos nuestras muestras de entrenamiento
x_entre = x_entre/255
x_test = x_test/255
y_entre[y_entre==1]=11
y_test[y_test==1]=11

from tensorflow.keras.utils import to_categorical #importamos para nuestras salidas de la red
y_entre_hot = to_categorical(y_entre)
y_test_hot = to_categorical(y_test)

dimensiones datos de entrenamiento : (951, 300, 300, 3) (951,)
dimensiones de datos de test: (238, 300, 300, 3) (238,)
```

Figura 3: imaganes prueba y entrenamiento

Modelo

Para el tratamiento de imágenes con redes neuronales es común usar convolucionales, que detectan patrones aplicando distintos filtros a las imágenes y extrayendo las características de cada grupo de imágenes y son muy efectivas en lo que hacen, por lo que usaremos una red convolucional con Keras.

Considerando una imagen de entrada de (300,300,3) con capas Conv2D, de 32,32,64 y 128 filtros, una capa Max-Pooling2D y apagamos algunas neuronas, con una función de activación relu y finalmente una capa densa con activación softmax que se usa comúnmente

para clasificación múltiple.

```
def arquitectura_modelo(activation,input_shape,num_classes,dropout):
    model = Sequential()

    model.add(keras.layers.Conv2D(32,(2,2), activation = activation, input_shape = input_shape))
    model.add(keras.layers.MaxPooling2D(pool_size = (2,2)))
    model.add(keras.layers.Dropout(dropout))
    model.add(keras.layers.Conv2D(32,(2,2), activation = activation, input_shape = input_shape))
    model.add(keras.layers.MaxPooling2D(pool_size = (2,2)))
    model.add(keras.layers.Dropout(dropout))
    model.add(keras.layers.Conv2D(64,(2,2), activation = activation))
    model.add(keras.layers.MaxPooling2D(pool_size = (2,2)))
    model.add(keras.layers.Dropout(dropout))
    model.add(keras.layers.Conv2D(128,(2,2), activation = activation))
    model.add(keras.layers.MaxPooling2D(pool_size = (2,2)))
    model.add(keras.layers.Dropout(dropout))
    model.add(keras.layers.Flatten())
    model.add(keras.layers.Dense(128, activation=activation))
    model.add(keras.layers.Dense(num_classes, activation='softmax'))
    return model
```

Figura 4: Modelo

4. Procedimiento

Entrenamiento modelo Para realizar el entrenamiento consideramos un optimizador Adam, perdida de categorical crossentropy y una métrica de accuracy. A unas 28 épocas para obtener los las graficas y la precisión del modelo, además de probar con los datos de test su desempeño final.

Cabe resaltar que son demasiadas variables, cambiamos la arquitectura de red, las capas y la cantidad de filtros para obtener una arquitectura adecuada. Otra limitante fue la RAM disponible de Colab, ya que por la cantidad de procesamiento de información se llegaba a desconectar limitando el desarrollo de la red neuronal, así como la cantidad de datos asociados al procesamiento. Con estas consideraciones procede dimos a entrenar el modelo y se obtuvo lo siguiente.

5. Resultados y Analisis

Modelo 1 (6 clases)

Para las 6 categorías obtuvimos un desempeño poco preciso con cerca del 33 por ciento de precisión que es demasiado poco, esto se atribuye principalmente a la poca cantidad de datos dispo-

nibles para cada clase, por lo que consideramos implementar una técnica conocida como aumento de datos, ya que, si tenemos dataset pequeños, podemos simular más imágenes a partir de una original, realizando rotaciones, reflexiones de manera aleatoria (Algoritmo en el repositorio)

Con los datos aumentados disminuyendo las clases a la mitad y considerando la misma arquitectura de red y variables obtenemos el modelo 2.

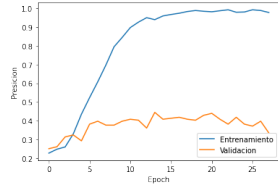


Figura 5: Presición M1

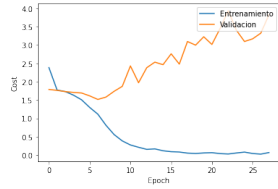


Figura 6: Perdida M1

Modelo 2 (3 clases aumentado)
Para esta parte seleccionamos las especies, *Lophophora Williamsii*, *Mammillaria Bocasana* y *Pelecypora Ase-lliformis*, aplicamos el aumento de datos a estas imágenes con un total de 1370 imágenes y realizamos los mismos procedimientos anteriores mencionados, obteniendo una precisión con los datos de test de 77 por ciento.

Modelo 3 (3 clases aumentado proporcional)

En la prueba anterior notamos que los datos no son proporcionales y se entrena con más datos de una especie que de

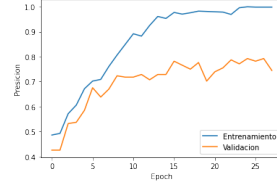


Figura 7: Presición M2

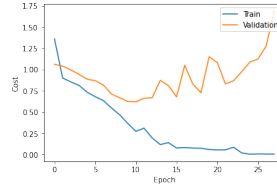


Figura 8: Perdida M2

otra, por lo que ahora aumentamos los datos, pero de una manera proporcional con un total de 1483 al realizar el test tenemos un 77 por ciento también.

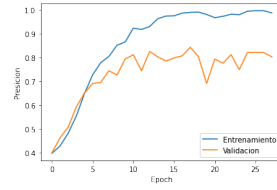


Figura 9: Presición M3

Comparación de modelos

La siguiente tabla contiene los datos de los 3 modelos para realizar una comparación entre ellos y notar sus desempeños.

6. Conclusiones

Consideramos que la cantidad de imágenes fue poca y se necesito de más imágenes y mayor capacidad computacional para lograr un mejor entrenamiento y resultados, pero fue capaz de detectar patrones esta versión de IA por lo que se puede usar para clasifi-

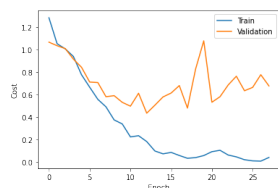


Figura 10: Perdida M3

Modelo	total	Entre	Test
1	1269	951	318
2	1370	1027	343
3	1483	1112	371

Cuadro 2: Datos

cación reducida de cactáceas sin embargo es necesario entrenarla con más especies e imágenes para que sea competitiva.

El aumento de datos y disminución de clases resulto efectivo a la hora de mejorar el modelo pasando de un 33 porciento a un 77 porciento, hace falta realizar mas pruebas con el aumento de datos proporcional, ya que los resultados mostrados en este articulo no son concluyentes.

Explorar otro tipo de técnicas como lo es la transferencia de aprendizaje y seguir variando los parámetros de nuestra arquitectura de red para obtener mejores resultados.

Referencias

- (1) Bravo, H. y H. Sánchez-Mejorada. 1978-1991. Las Cactáceas de México, 3 vols. Universidad Nacional Autónoma de México, México
- (2) Ken March. 1988. Cactus y suculentas. Ed. Blume. España
- (3) Reyes, S.J. 1997. Cultivo y

Modelo	Presición	Perdida	Clases
1	0.3364	3.8715	6
2	0.7725	1.3164	3
3	0.7789	1.0390	3

Cuadro 3: Datos Entrenamiento

propagación como plantas de ornato. En suculentas mexicanas: cactáceas. Conabio. México

(4)Gulli, Antonio, Deep Learning with Keras, Packt Publishing Ltd, 2017

(5)Montaño Moreno, Juan, Redes neuronales artificiales aplicadas al análisis de datos, Universitat de les Illes Balears, 2017,