

Universidad Rafael Landívar, Campus de Quetzaltenango
Facultad de Ingeniería
Inteligencia Artificial
Ing. Dhaby Xiloj

Documentación, proyecto I.A. 2019

Nombres y Carnets:

Cristian Mauricio Monroy Rosales, 15365-15
Edvin Gonzalo Felipe Cristóbal, 22733-15

Quetzaltenango, 26 de Abril de 2019

Fruta escogida para trabajar en el proyecto: Chile jalapeño

El chile jalapeño es una de las frutas más picantes de [Capsicum annuum](#) más extensamente cultivadas y consumidas en [América](#). El fruto del jalapeño es carnoso y alargado, alcanzando los 7 cm de largo y alrededor de 3 de ancho en la base. Se emplea tanto antes como después de la maduración; una parte importante de la producción total se destina al [secado](#), proceso tras el cual se lo conoce como chile chipotle (del [náhuatl](#) chilpochtli, que significa "chile ahumado"). Es una variedad de fruta medianamente picante, entre 2.500 y 8.000 puntos en la [escala Scoville](#), aunque la intensidad del sabor depende en gran medida de las características del terreno y de la variedad de semilla; las más habituales son conocidas como típico, meco y morita. Buena parte de la [capsaicina](#), el [alcaloide](#) que provoca la picazón, se concentra en las venas y [semillas](#) en el interior del fruto; retirarlas antes de su empleo proporciona un sabor más delicado.

El jalapeño es un chile picante; sus frutos son firmes, aromáticos, de buen sabor y de aspecto atractivo por lo cual tienen muy buena aceptación en el mercado, tanto en el nacional como en el extranjero.

Como los restantes cultivares de *C. annuum*, el jalapeño se planta habitualmente poco antes del comienzo de la estación húmeda, y lo favorecen las altas temperaturas. Normalmente se cosecha alrededor de 70 días tras la siembra, rindiendo entre 25 y 35 frutos por planta.

Las principales enfermedades que atacan a este cultivo es *Choanephora cucurbitarum*, mancha gris o tizón por alternaria.

¿Cómo se determina la maduración?

En este caso, el chile jalapeño se utiliza totalmente verde para su distribución en el mercado, por lo general, ya que es la etapa en don

de estos mismos se encuentran más frescos, y es cuando mejor degustan su sabor los compradores. Conforme partes de este fruto se van tornando rojas, empiezan su proceso de maduración, el fruto se torna rojo por partes, poco a poco, hasta que el fruto por completo se vuelve rojo. Hay que tomar en cuenta que entre más verde es el producto, más fresco es, entre más rojo es, menos fresco es y menos etapa de vida tiene. Los frutos pudieran madurar más rápido o pudieran demorar en su maduración, dependiendo de qué manera se maneje la distribución y qué procesos (y con qué cuidado) se lleven a cabo para su ensamblado y posterior trasbordo y venta de producto. El proceso de maduración también es proporcional a la temperatura del ambiente, entre menor ambiente hay en su entorno, con menor rapidez madura el chile jalapeño, pero si se encuentra el mismo en un ambiente cálido, tiende a madurar más rápido. La humedad en el ambiente también acelera el proceso de maduración.

La escala de maduración es la siguiente:



En donde el chile jalapeño más verde (el del extremo izquierdo de la imagen) sería la etapa 1, y el chile jalapeño más maduro (el del extremo derecho de la imagen) sería la etapa 5. Las etapas van enumeradas del lado izquierdo al derecho

¿Cómo determinar la calidad?

Tamaño:

Los chiles jalapeños se pueden distribuir en el mercado dependiendo del tamaño y de la tendencia del mercado. Los altos mandos de la empresa tienen la capacidad necesaria para observar qué tendencias hay en el momento para la venta del chile jalapeño en el mercado. En algunos momentos puede que las personas prefieran el chile jalapeño en su tamaño grande, o bien lo prefieran en trozos o con tamaños pequeños. En cuestiones así se fijan las personas que distribuyen el chile jalapeño para su venta. Esto básicamente influye en la percepción de calidad en las personas, aunque no fue un punto que se tomara en cuenta para la realización del proyecto.

Color:

Este es un factor para determinar la calidad del chile jalapeño. Como se mencionó arriba, normalmente cuando el jalapeño está verde es cuando se distribuye en el mercado, ya sea para ensamblado o simplemente para venderse en el mercado. Las personas prefieren realizar la venta así debido a que es cuando mejor pueden aprovechar el tiempo para vender el producto por lo mismo que apenas está iniciando su ciclo de vida. Entre más maduro está, el fruto está menos fresco, por tanto las personas tienen menos tiempo ya sea para venderlo en su estado natural o de forma ensamblada (que en este caso sería necesario realizar procesos propios de cada empresa, tales como transporte, transbordo, y entre otros).

Temporada:

En temporada de lluvia el producto abunda, pero es lo contrario cuando acá en Guatemala es verano, pues el producto escasea. En este caso, se aplica la ley de la oferta y la demanda; el producto cuando escasea es caro, pero cuando abunda es más barato.

Plagas:

Entre las plagas que pueden afectar el estado del chile jalapeño encontramos:

- **Pulga saltona.** Este insecto es muy pequeño, mide de 1.5 a 3.0 milímetros de largo, su forma es oval y es de color negro. Generalmente se localiza en las partes tiernas (cogollos) de las plantas. El daño que ocasiona, consiste en pequeños orificios redondos que atraviesan las hojas jóvenes, de tal manera que al desarrollarse

las hojas, también aumentan las dimensiones de los orificios, dando la apariencia de haber sido afectados por "tiro de munición"



- **Barrenillo o Picudo.** En estado adulto, este insecto es negro o café grisáceo y mide de 3 a 4 milímetros de longitud. La hembra deposita sus huevecillos en los botones florales o en los frutos pequeños. Del huevecillo sale un gusano sin patas, con cabeza café y mide aproximadamente 6 milímetros de largo; se alimenta de la masa de las semillas del centro del chile, lo que provoca que caiga antes de madurar y pierde así su valor comercial, además de contribuir a elevar las poblaciones de este insecto. Posteriormente, la larva se transforma en pupa y luego en adulto dentro del fruto caído. Los adultos o "picudos" barrenan o perforan el fruto con su pico y salen de él; de ahí su nombre de barrenillo.



- **Pulgón.** Es un insecto que mide 1.5 milímetros de largo, tiene cuerpo suave de tonalidad verde y puede o no presentar alas. Se localiza principalmente en el reverso de la hoja, en los brotes terminales y en

las partes sombreadas de los tallos y flores. Se alimenta de la savia de las plantas, las cuales se debilitan cuando las poblaciones son altas. Los pulgones alados son los más dañinos para el cultivo, por su habilidad para desplazarse, ya que transmiten enfermedades virosas, tales como "mosaicos" y el "enrollamiento de la hoja".



- **Gusanos trozadores.** Tienen un aspecto "grasoso"; son de color oscuro, piel suave y rechonchos, su tamaño varía de 3.6 a 5.0 centímetros de largo. Su comportamiento se caracteriza porque cuando son perturbados se enrollan fuertemente y fingen estar muertos; además, suelen esconderse cerca de la base de las plantas.



- **Gusano soldado.** El adulto es una palomilla de color café oscuro; la hembra deposita sus huevecillos sobre las hojas en forma de masas y las cubre con una sustancia gris. Las larvas son de color verde pálido y pueden llegar a medir hasta 4 centímetros de largo.



- **Mosquita blanca.** Los adultos miden 2 milímetros de longitud, son de color amarillento, con las alas cubiertas por un polvillo blanco. Las hembras depositan sus huevecillos en el envés de las hojas, las cuales tienen una tonalidad crema, las ninfas son planas, ovaladas y chupan la savia de las hojas. Cuando se presentan infestaciones severas de esta plaga, las plantas se vuelven amarillentas, se marchitan y finalmente mueren; además, se considera como un transmisor muy importante de enfermedades virosas.



- **Minador de la hoja.** Los adultos miden de 2 a 3 milímetros de longitud y son amarillentos con el dorso oscuro. Las hembras después de aparear, depositan sus huevecillos dentro de los tejidos de la hoja y las larvas emergen dos a tres días después, las cuales tienen una apariencia cilíndrica y miden 1.5 milímetros; al principio son incoloras y posteriormente se vuelven amarillentas al final de su desarrollo. Desde su emergencia se alimentan del tejido de las hojas, formando túneles irregulares que se amplían a medida que crece la larva.



- **Trips.** Aunque es una plaga de menor cuidado, se debe de tener precaución porque en ocasiones las poblaciones son fuertes pudiendo llegar a provocar abortos en las flores. El daño que ocasionan es al alimentarse de las flores ya que tienen el aparato bucal raspador chupador, incluso son transmisores de virus en diferentes cultivos.



Enfermedades:

Entre las enfermedades que pueden sufrir los chiles jalapeños encontramos:

- **Marchitez del chile.** Es la enfermedad principal que se presenta en la región y las condiciones ambientales que favorecen su desarrollo son una humedad del suelo alta y la presencia de temperaturas frescas. Los síntomas característicos de la marchitez del chile comienzan precisamente con un secamiento muy leve de la planta. Después de tres o cuatro días las plantas se marchitan completamente, observándose en el cuello un necrosamiento muy marcado y al efectuarse un corte transversal, en el tallo se aprecia una coloración café oscura.

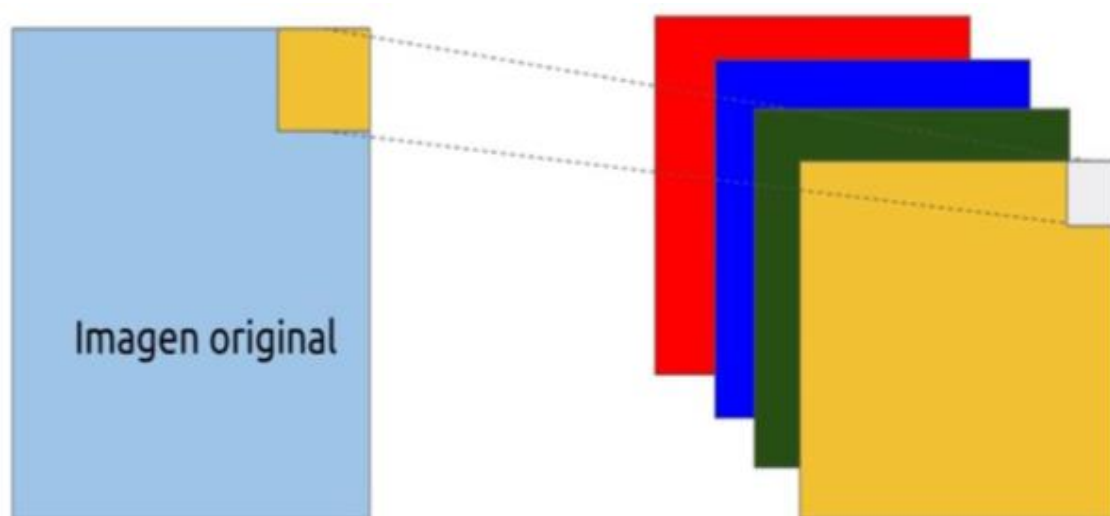


- **Mancha bacteriana.** Los síntomas de esta enfermedad se presentan como pequeñas manchas irregulares de color verde amarillento, que después se vuelven café, tanto en las hojas como en el fruto. Las condiciones que favorecen su desarrollo son las lluvias, días nublados, suelos excesivamente mojados y temperaturas de 24 a 29 grados centígrados; las manchas en las hojas llegan a ser tan numerosas, que al juntarse forman manchas grandes. Finalmente las hojas muy afectadas caen de la planta.



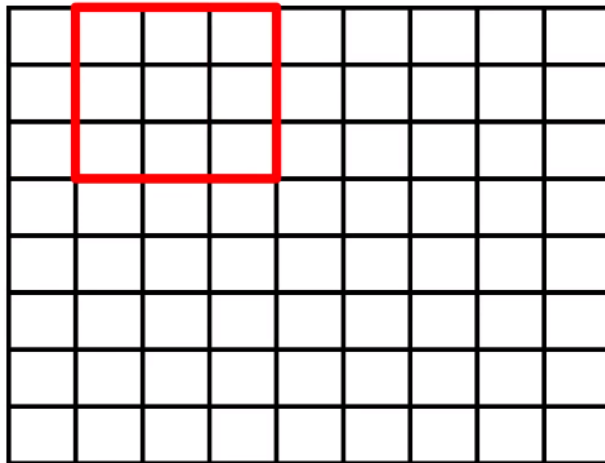
¿Cómo funciona una red neuronal convolucional?

- **Convoluciones**



Lo primero que una red neuronal convolucional utiliza son las denominadas 'convoluciones'. Estas convoluciones utilizan filtros de un determinado tamaño, que lo que hacen es recorrer toda la imagen de poco a poco.

Stride ó Paso

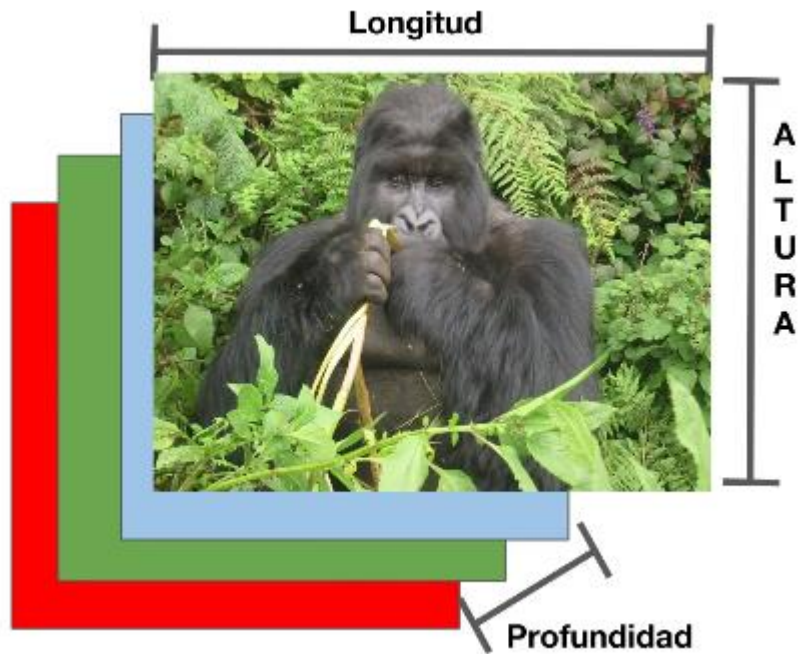


Stride=1

**El filtro
se
recorre
un pixel
a la vez.**

Los recorridos pueden ser de forma detenida o de forma rápida, es decir, puede saltarse varios píxeles a la vez, entre más grandes son estos saltos, más llega a reducir la imagen original la convolución.

Las convoluciones ayudan a disminuir el tamaño de la imagen. Estos recorridos aumentan la profundidad de la imagen, que básicamente quedaría así la misma:



Se observa en la imagen que la misma tiene una profundidad. Esto lo provocan los filtros que van pasando por la imagen. Estos filtros son necesarios pues cada filtro se encarga de hacer que la red neuronal detecte ciertos patrones, tales como que, por ejemplo, un filtro se encargue de detectar colores, otro filtro se encargue de detectar sombras, otro filtro se encargue de detectar formas, y entre otros ejemplos más que se podrían mencionar.

- **Pooling (Agrupación):**

Este método se enfoca en reducir aún más el tamaño de la imagen original, de dos formas que a continuación se van a detallar:

- Max Pooling: Este algoritmo usa un filtro de determinado tamaño, que se dedica a obtener el valor máximo de una matriz de valores, para formar una nueva matriz sólo con los valores más altos dentro del filtro en cada stride que haga.

Max Pooling

2	7	9	7
4	9	1	4
4	7	6	2
5	1	8	2

9	9
7	8

- Average Pooling: Este algoritmo usa un filtro, igualmente de determinado tamaño como el método de Max Pooling, con la diferencia de que obtiene un promedio de los valores que quedan en la matriz del filtro, almacenando los mismos

Average Pooling

2	7	9	7
4	9	1	4
4	7	6	2
5	1	8	2

5.5	5.25
4.25	4.5

Hay que mencionar que en los métodos anteriormente mencionados, no se toma en cuenta la profundidad obtenida y creada por los filtros convolucionales, por tanto, esto no se toca.

Luego de este proceso, que consistió en hallar una profundidad a la imagen, y en reducir la misma, se procede al crear la red neuronal a aplanar la imagen, y así mismo a hacer generadores de imágenes que tomen en cuenta que el objeto a analizar esté cerca o lejos, pero que la debe de detectar, inclusive tomando en cuenta si el objeto está inclinado.

Un apartado para explicar cómo funciona la CNN de forma matemática sería con lo siguiente:

Neuronas Convoluciones

En la fase de extracción de características, las neuronas sencillas de un perceptron son reemplazadas por procesadores en matriz que realizan una operación sobre los datos de imagen 2D que pasan por ellas, en lugar de un único valor numérico. La salida de cada neurona convolucional se calcula como:

$$Y_j = g \left(b_j + \sum_i K_{ij} \otimes Y_i \right)$$

Donde la salida Y_j de una neurona j es una matriz que se calcula por medio de la combinación lineal de las salidas Y_i de las neuronas en la capa anterior cada una de ellas operadas con el núcleo de convolucional K_{ij} correspondiente a esa conexión. Esta cantidad es sumada a una influencia b_j y luego se pasa por una función de activación $g(\cdot)$ no-lineal.

El operador de convolución tiene el efecto de filtrar la imagen de entrada con un núcleo previamente entrenado. Esto transforma los datos de tal manera que ciertas características (determinadas por la forma del núcleo) se vuelven más dominantes en la imagen de salida al tener estas un valor numérico más alto asignados a los pixeles que las representan. Estos núcleos tienen habilidades de procesamiento de imágenes específicas, como por ejemplo la detección de bordes que se puede realizar con núcleos que resaltan la gradiente en una dirección en particular. Sin embargo, los núcleos que son entrenados por una red neuronal convolucional generalmente son más complejos para poder estos extraer otras características más abstractas y no triviales.

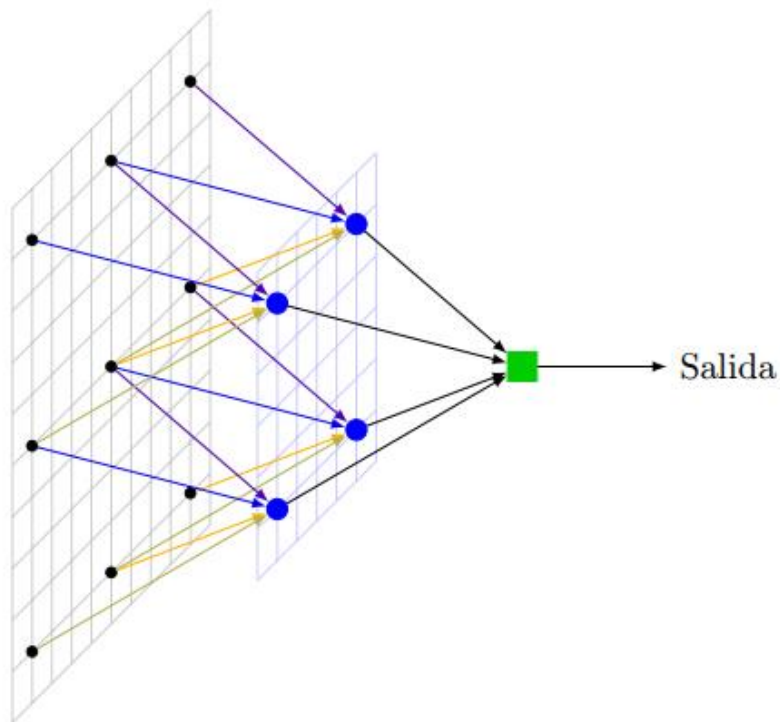
Neuronas de Clasificación

Después de una o más fases de extracción de características, los datos finalmente llegan a la fase de clasificación. Para entonces, los datos han sido depurados hasta una serie de características únicas para la imagen de entrada, y es ahora la labor de esta última fase el poder clasificar estas características hacia una etiqueta u otra, según los objetivos de entrenamiento.

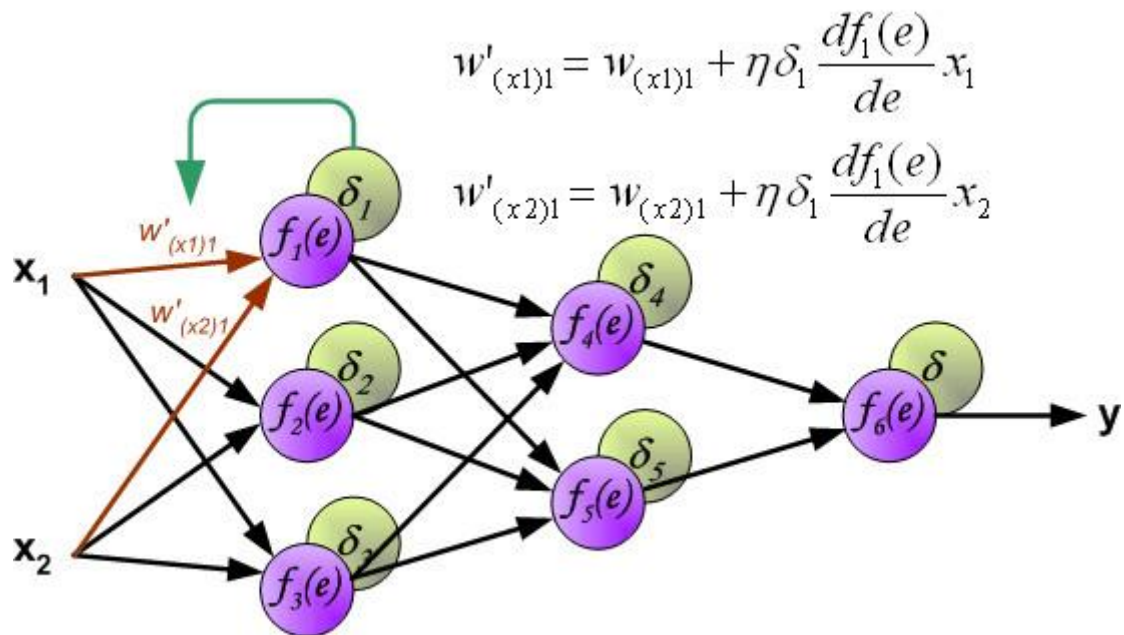
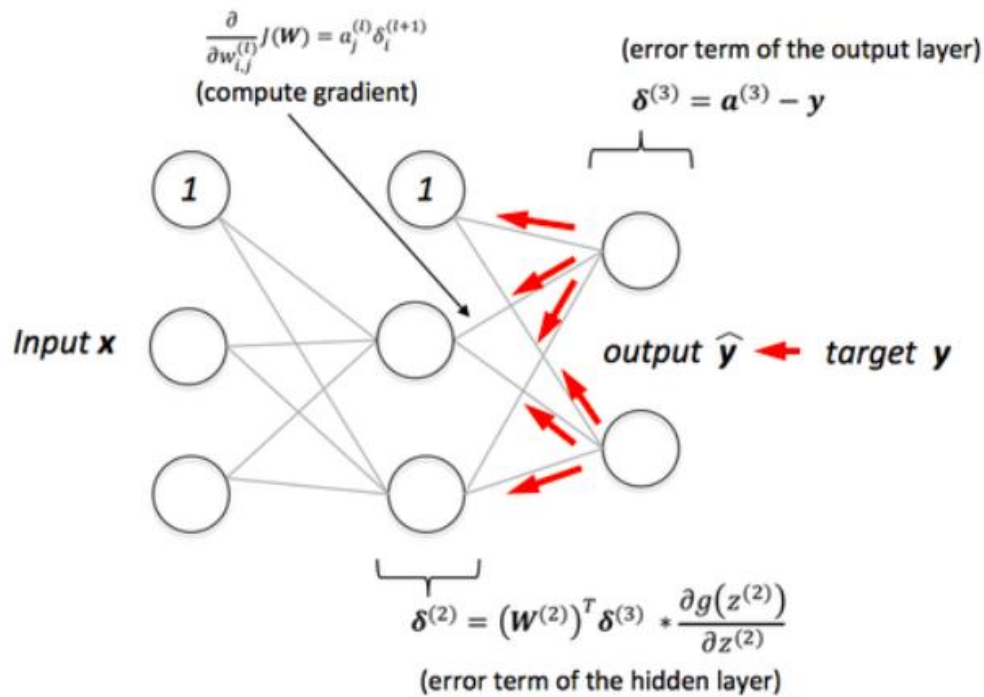
Las neuronas en esta fase funcionan de manera idéntica a las de un perceptron multicapas, donde la salida de cada una se calcula de esta forma:

$$y_j = g \left(b_j + \sum_i w_{ij} \cdot y_i \right)$$

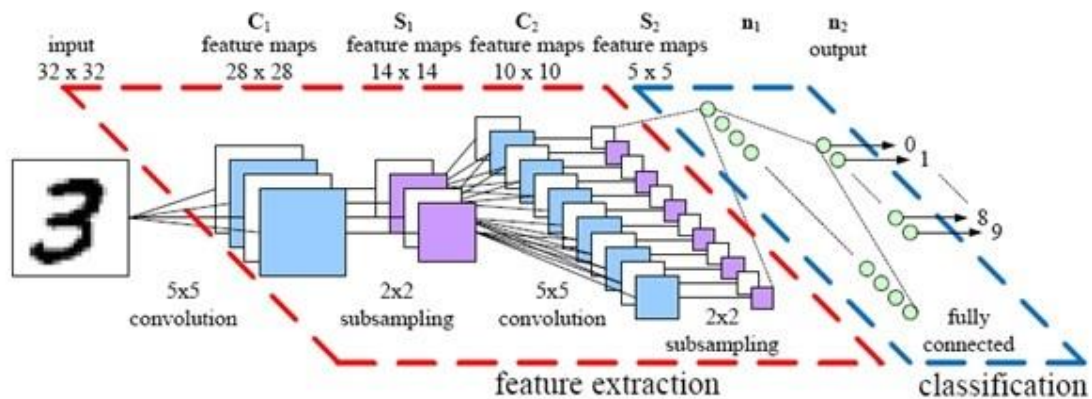
Donde la salida y_j de una neurona j es un valor que se calcula por medio de la combinación lineal de las salidas y_i de las neuronas en la capa anterior cada una de ellas multiplicadas con un peso w_{ij} correspondiente a esa conexión. Esta cantidad es sumada a una influencia b_j y luego se pasa por una función de activación $g(\cdot)$ no-lineal.



Algoritmo Propagación hacia atrás



Estas redes convolucionales la imagen de entrada en distintas capas para reducir en número de variables a tratar (pixeles) de la siguiente forma:



¿Cómo entrenamos y predecimos el chile jalapeño?

Antes de pasar a explicar el código tanto de entrenamiento como de predicción, hay que anticipar que el proyecto se codificó en Jupyter, mencionado esto, ahora sí se explica cada segmento de código.

- Entrenamiento:

```
import tensorflow as tf
import numpy as np
from skimage import io
from skimage import transform
import os
import matplotlib.pyplot as plt
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers.core import Dense, Flatten
# Dense permite utilizar las ultimas capas que son nodos o neuronas en un capa
# Flatten permite cambiar una matriz a un vector, que pueda ser entendida por las demas neuronas conectadas
from keras.optimizers import SGD # descenso del gradiente
from keras.layers.convolutional import Convolution2D, MaxPooling2D
```

Este trozo de código no es más que el importe de librerías, que sirven para convoluciones, para obtener las imágenes de las carpetas (para entrenar y para predecir), para guardar los pesos y modelos de la red neuronal y para crear la CNN.

```
def lectura_imagenes(directorio):
    # Vector de directorios, existente en la ruta indicada
    dirs = [ d for d in os.listdir(directorio) ]
    # Lista que va guardar las imagenes leidas
    images = []
    etiquetas = []
    i = 0
    for directory in dirs:
        nom_carpeta = os.path.join(directorio,directory)
        # Listado de nombres de imagenes, en una sola ruta
        list_nombre_imagenes = [ os.path.join(nom_carpeta,f)
                                for f in os.listdir(nom_carpeta)
                                if f.endswith(".jpg")]

        # Leyendo las imagenes
        for f in list_nombre_imagenes:
            images.append(io.imread(f))
            etiquetas.append(i)
            i += 1

    return images,etiquetas
```

En la función `lectura_imagenes` se cargan las imágenes de cada carpeta de la ruta especificada, y se guardan en listas.

```
# Directorios
images_entrenamiento,etiquetas = lectura_imagenes("images/entrenamiento")
#images_validacion = lectura_imagenes("images/validacion")
```

Acá se asigna la ruta a las dos variables (que se explicarán más a detalle adelante) de donde se encuentran almacenadas las imágenes

```
nclases = 7
etiquetas = np_utils.to_categorical(etiquetas,nclases)
#y_test = np_utils.to_categorical(y_test, nclases)
```

```
# Redimensionando Las imágenes a 32x32
images_entrenamiento = [ transform.resize(image,(50,50)) for image in images_entrenamiento]
#images_validacion = [ transform.resize(image,(32,32)) for image in images_validacion]
```

```
C:\Users\Felipe\Anaconda3\lib\site-packages\skimage\transform\_warps.py:105: UserWarning: The default mode, 'constant', will be
changed to 'reflect' in skimage 0.15.
  warn("The default mode, 'constant', will be changed to 'reflect' in "
C:\Users\Felipe\Anaconda3\lib\site-packages\skimage\transform\_warps.py:110: UserWarning: Anti-aliasing will be enabled by defa
ult in skimage 0.15 to avoid aliasing artifacts when down-sampling images.
  warn("Anti-aliasing will be enabled by default in skimage 0.15 to "
```

```
images_entrenamiento = np.array(images_entrenamiento)
images_entrenamiento.shape
```

En este trozo de código se redimensionan las imágenes de entrenamiento, luego se pasan estos datos a un array (tome en cuenta que un array no es lo mismo que una lista).

```
tam_lote = 15
iteraciones = 1
filtro = 3
num_colores = 3
ancho = 30
altura = 30
```

Esto no es más que la asignación de variables que se usarán para entrenar la red neuronal.

```
modelo = Sequential()
```

```
# Primera red convolucional
modelo.add(Convolution2D(64, kernel_size=(6,6), strides=(1,1), activation='relu',
                        input_shape=(50, 50, 3)))

modelo.add(MaxPooling2D(pool_size=(2,2), strides=2))
```

WARNING:tensorflow:From C:\Users\Felipe\Anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.

```
modelo.add(Convolution2D(100, kernel_size=(6,6), strides=(1,1), activation='relu'))
modelo.add(MaxPooling2D(pool_size=(2,2), strides=2))
```

```
# Segunda red convolucional
modelo.add(Convolution2D(50, kernel_size=(6,6), strides=(1,1), activation='relu'))
modelo.add(MaxPooling2D(pool_size=(2,2), strides=2))
```

En estos trozos de código se están formando las convoluciones, con los parámetros de los filtros que se usarán tanto en cada convolución como en cada max pooling, qué función de activación se usará, el tamaño de cada filtro, y qué tantos pixeles se saltará el filtro.

```
# Aplanamiento de las redes convolucionales, pasandolos a vectores
modelo.add(Flatten())
# NN con 50 neuronas
modelo.add(Dense(128, activation='relu'))
```

```
modelo.add(Dense(64, activation='relu'))
# Softmax con 10 neuronas de salida
modelo.add(Dense(nclasses, activation='softmax'))
```

```
# Optimizador, con gradiente descendiente
optimizador = SGD(lr=0.001)
modelo.compile(loss='categorical_crossentropy',
              optimizer=optimizador,
              metrics=['accuracy'])
```

```
# Entrenamiento
modelo.fit(images_entrenamiento, etiquetas, epochs=180, batch_size=5, verbose=2)
```

En este trozo de código se aplanan la información, y luego se crea la CNN, con tres capas, en donde la última capa sería la capa que indica los resultados obtenidos, ya sea una etapa de maduración o que no es un chile jalapeño, independientemente de la imagen de entrada que reciba la CNN. En sí se identifica durante el proceso si existe un chile jalapeño, si sí, entonces se procede a identificar su etapa de maduración, si no, una de las neuronas indicará que es una imagen cualquiera.

```
target_dir = "./model/"
if not os.path.exists(target_dir):
    os.mkdir(target_dir)

modelo.save('./model/model.h5')
modelo.save_weights('./model/pesos.h5')
```

En este trozo de código se guarda la información (en una carpeta llamada model) de la CNN y de sus pesos.

- Predicción:

```
import numpy as np
from keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.models import load_model
import matplotlib.pyplot as plt
```

C:\Users\Felipe\Anaconda3\lib\site-packages\h5py__init__.py:72: UserWarning: h5py is running against HDF5 1.10.2 when it was built against 1.10.3, this may cause problems
'{0}.{1}.{2}'.format(*version.hdf5_built_version_tuple)
Using TensorFlow backend.

```
longitud, altura = 50, 50
modelo = './v1.0/model/model.h5'
pesos_modelo = './v1.0/model/pesos.h5'
cnn = load_model(modelo)
cnn.load_weights(pesos_modelo)
```

Acá se importan las librerías, básicamente servirán para poder cargar los modelos y pesos del CNN, y las fotos de prueba para identificar la etapa de maduración de un chile jalapeño.

```
def predict(file):
    x = load_img(file, target_size=(longitud, altura))
    plt.imshow(x)
    plt.show()
    x = img_to_array(x)

    x = np.expand_dims(x, axis = 0)

    array = cnn.predict(x)
    result = array[0]
    answer = np.argmax(result)
    print("Answer: "+str(answer))
    if (answer == 0):
        print("pred: Etapa 1 \n Calidad: Muy alta (100%)")
    elif answer == 1:
        print("pred: Etapa 2 \n Calidad. Alta (85%)")
    elif answer == 2:
        print("pred: Etapa 3 \n Calidad: Media Alta (65%)")
    elif answer == 3:
        print("pred: Etapa 4 \n Calidad: Media (50%)")
    elif answer == 4:
        print("pred: Etapa 5 \n Calidad: Baja (30%)")
    elif answer == 5:
        print("pred: Etapa 6 \n Calidad: Muy baja (10%)")
    elif answer == 6:
        print("Precaución: El objeto que analizó la red neuronal detectó un objeto que no es un chile jalapeño")

    return answer
```


En esta función se predice la etapa y calidad del chile jalapeño, se hace una serie de procesos tales como convertir la imagen a array y luego se almacena el resultado en la variable answer, que es la que indica la etapa.

```
#predict('./images/Validacion/Etapa5/IMG_20190420_133245.jpg')  
predict('IMG_20190426_130217.jpg')
```

Acá sencillamente se llama a la función predict, se pone la ruta relativa de la imagen y ya predice sin problemas la madurez y calidad del chile jalapeño.

Las imágenes que utilizamos están en este link:

<https://drive.google.com/drive/folders/1w4u0XQzzD76hcHIwy13Zb5EhUwLekTB>

=