



Aprendizaje Automático Profundo (Deep Learning)

Dr. Facundo Quiroga - Dr. Franco Ronchetti

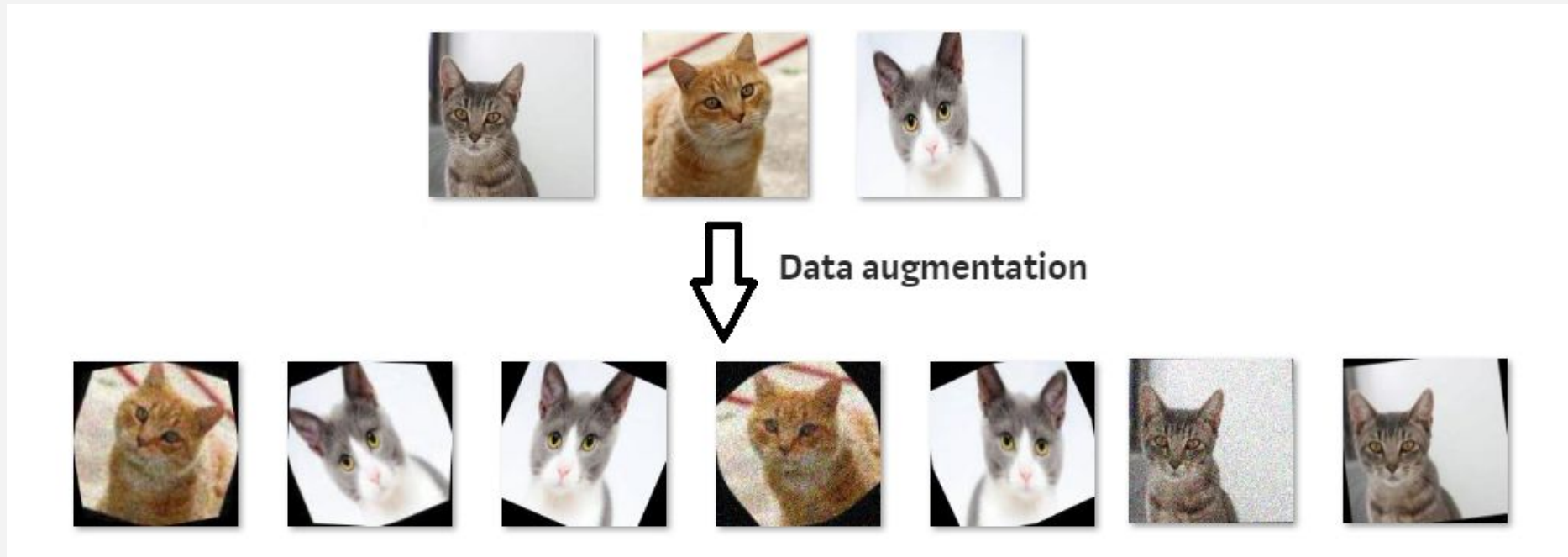


Data Augmentation

(Aumentación de datos)

Aumentación de Datos

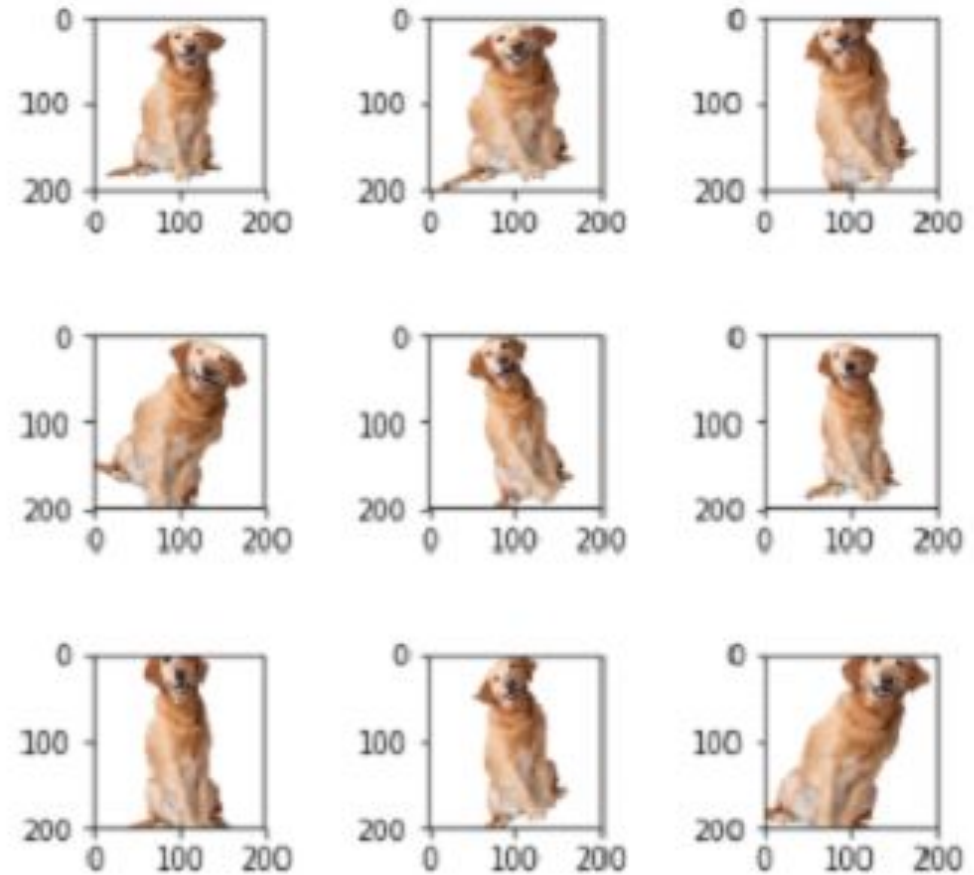
- Generar nuevos ejemplos
 - Definir transformaciones **útiles**
 - Transformar los ejemplos disponibles



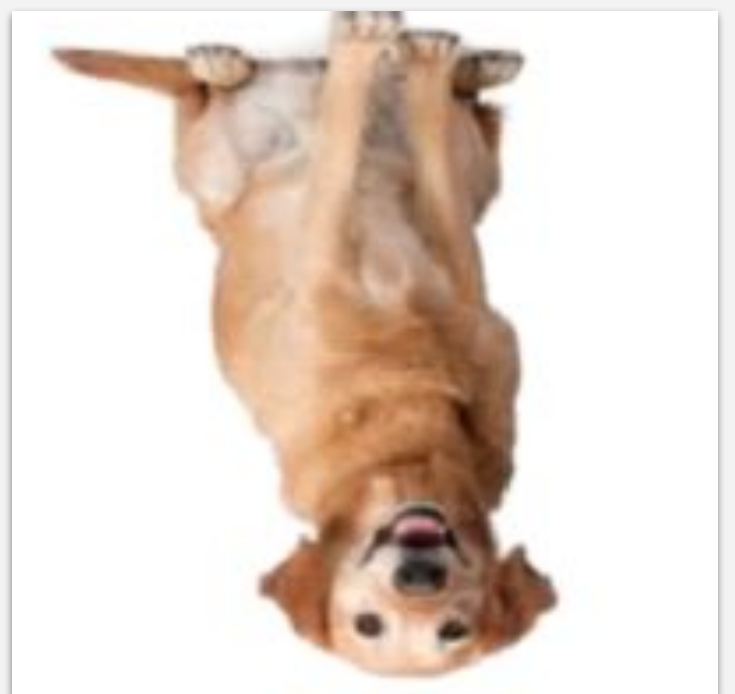
Rotaciones entre -30° y 30°



Augmented Images



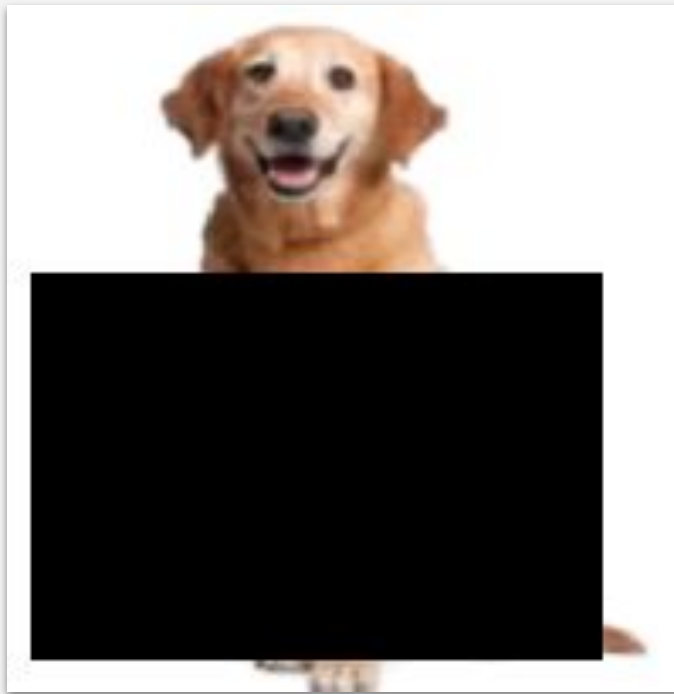
Flips horizontales (y verticales?)



Crops (recortes)



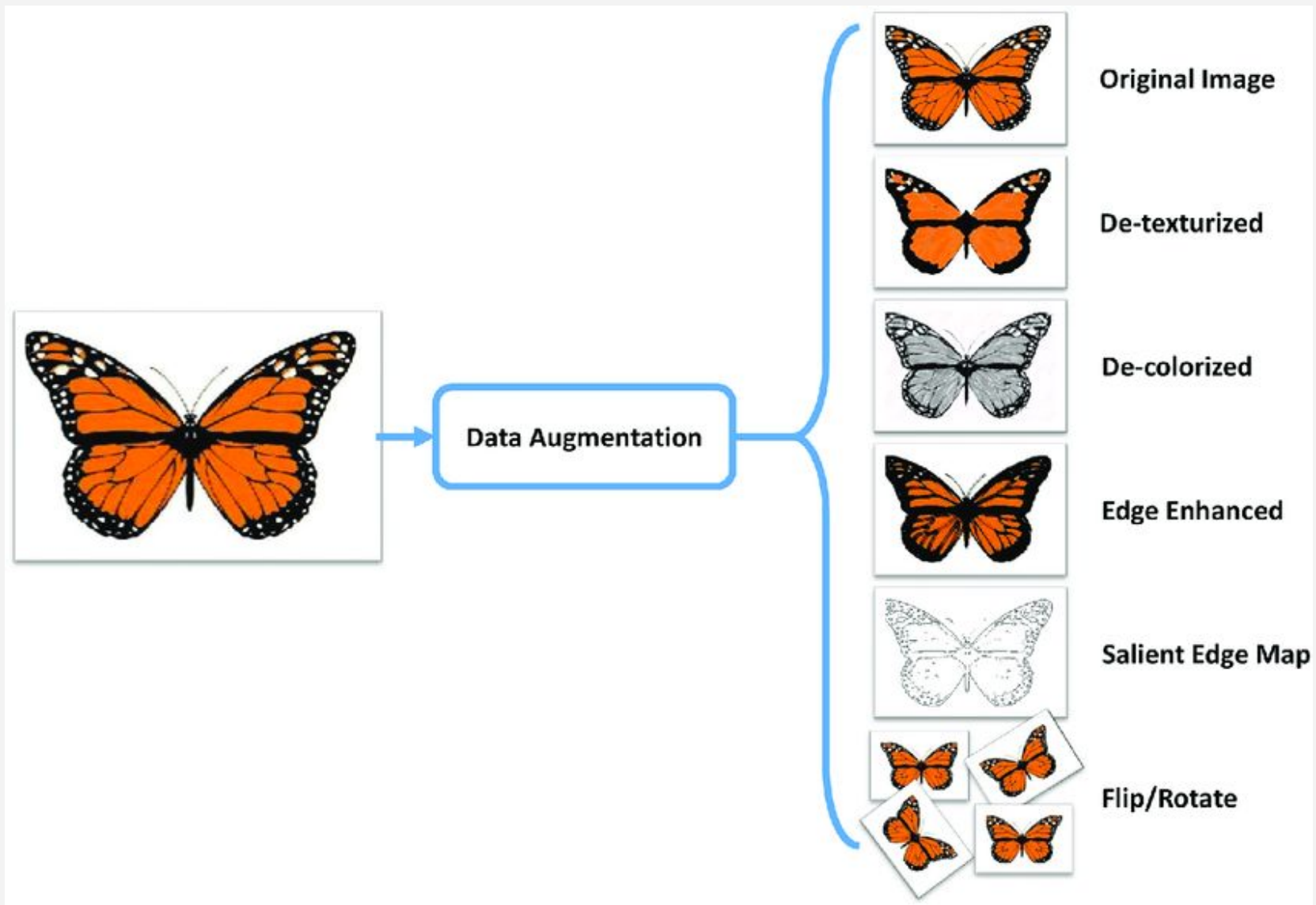
Oclusiones Parciales



Cambios de fondo



Cambios de brillo/color/textura



Explosión combinatoria de transformaciones

Data Augmentation:

a. No augmentation (= 1 image)



b. Flip augmentation (= 2 images)



c. Crop+Flip augmentation (= 10 images)



Aumentación de Datos + Keras + Imágenes.

- ImageDataGenerator con rotaciones de $-20^{\circ}/+20^{\circ}$, flip horizontales aleatorios, y escalado del valor de los píxeles.

```
x,y=load_samples()  
datagen = ImageDataGenerator(rotation_range=20,  
                             rescale=1./255,  
                             horizontal_flip=True,  
                             )  
  
# generador que transforma a medida que genera batches  
generator = datagen.flow(x,y,batch_size=32)  
...  
model.fit_generator(generator,  
                    steps_per_epoch=len(generator)/32, ...)
```

Aumentación de Datos + Keras + Imágenes.

- ImageDataGenerator: transformaciones *entrenables*: normalización

```
x,y=load_samples()  
# Centrar y dividir por media y desviación estándar  
datagen = ImageDataGenerator(featurewise_center=True,  
                             featurewise_std_normalization=True,  
                             )  
datagen.fit(x) # calcula la media y desviación  
generator = datagen.flow(x,y,batch_size=32)  
...  
model.fit_generator(generator,  
                    steps_per_epoch=len(generator)/32, ...)
```


Aumentación de Datos + Keras + Imágenes.

- ImageDataGenerator: transformaciones *entrenables* y conjuntos *train* y *test*

```
x_train,y_train,x_test,y_test=load_samples()  
# Centrar y dividir por media y desviación estándar  
datagen = ImageDataGenerator(featurewise_center=True,  
                             featurewise_std_normalization=True)  
datagen.fit(x_train) # calcula media y desviación con x_train  
train_generator = datagen.flow(x_train,y_train,batch_size=32)  
model.fit_generator(train_generator,  
                   steps_per_epoch=len(generator)/32, ...)  
# se utiliza el mismo para test sin volver a llamar a fit  
test_generator = datagen.flow(x_test,y_test,batch_size=32)  
model.evaluate_generator(test_generator)
```

Aumentación de Datos + Keras + Imágenes.

- ImageDataGenerator parámetros ([más detalles](#)):

```
keras.preprocessing.image.ImageDataGenerator(  
# normalización  
featurewise_center=False, featurewise_std_normalization=False,  
samplewise_center=False, samplewise_std_normalization=False,  
zca_whitening=False, zca_epsilon=1e-06,  
# transformaciones geométricas  
rotation_range=0, width_shift_range=0.0,  
height_shift_range=0.0, shear_range=0.0, zoom_range=0.0,  
horizontal_flip=False, vertical_flip=False, rescale=None,  
fill_mode='nearest', cval=0.0, # 2 opciones de las tr. geom.  
# Colores  
channel_shift_range=0.0, brightness_range=None,  
preprocessing_function=None) # transformación custom
```

Aumentación de Datos - NLP

- Data Augmentation
- Natural con imágenes
- No tanto con otros dominios
- Para Procesamiento de Lenguaje Natural (NLP):
 - Reemplazar por sinónimos
 - Insertar palabras
 - Sustituir palabras/frases
 - Borrar palabras

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A <i>lamentable</i> , superior human comedy played out on the <i>backward</i> road of life.
RI	A sad, superior human comedy played out on <i>funniness</i> the back roads of life.
RS	A sad, superior human comedy played out on <i>roads</i> back <i>the</i> of life.
RD	A sad, superior human out on the roads of life.