

# Data augmentation and data generators

A nice party trick

Filippo Biscarini  
Senior Scientist  
CNR, Milan (Italy)

Nelson Nazzicari  
Senior Scientist  
CREA, Lodi (Italy)



# What is data augmentation?



- Feed to your network “new” training data, derived algorithmically
- Deep neural network are always data-hungry
- No data sample is completely “used”
- Computers are stupid



# What is data augmentation?



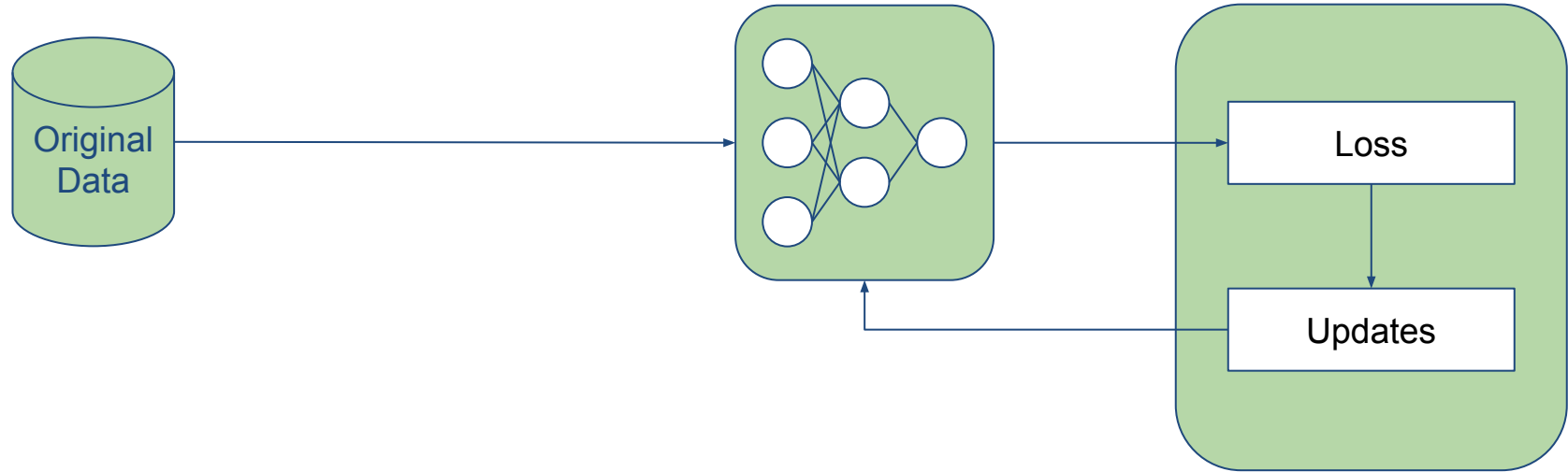
A cat



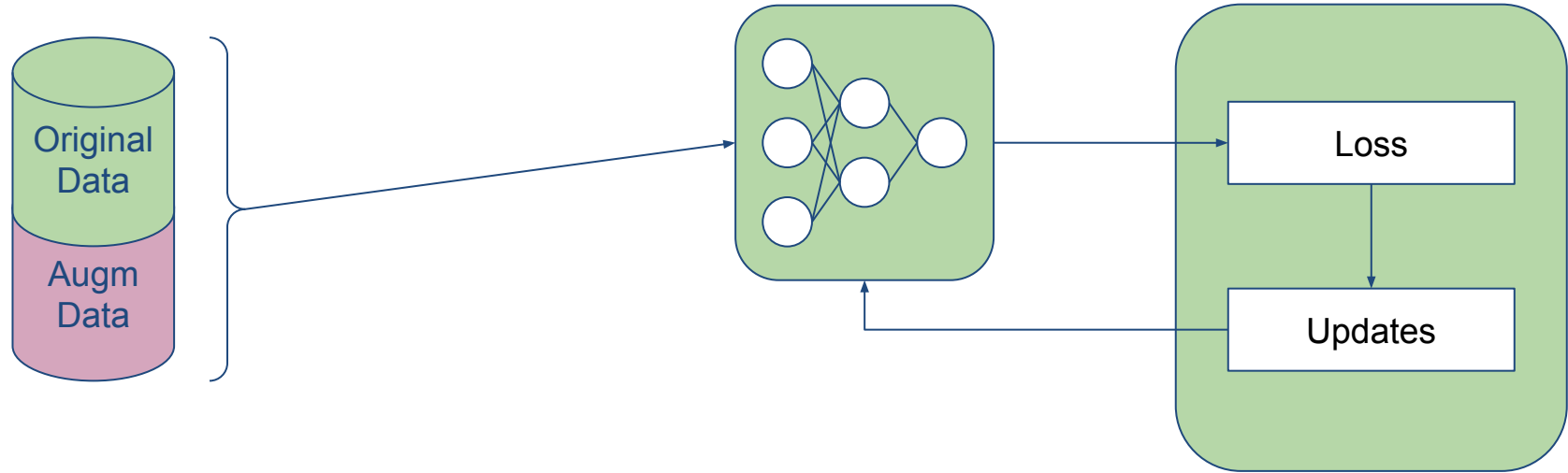
A completely different cat



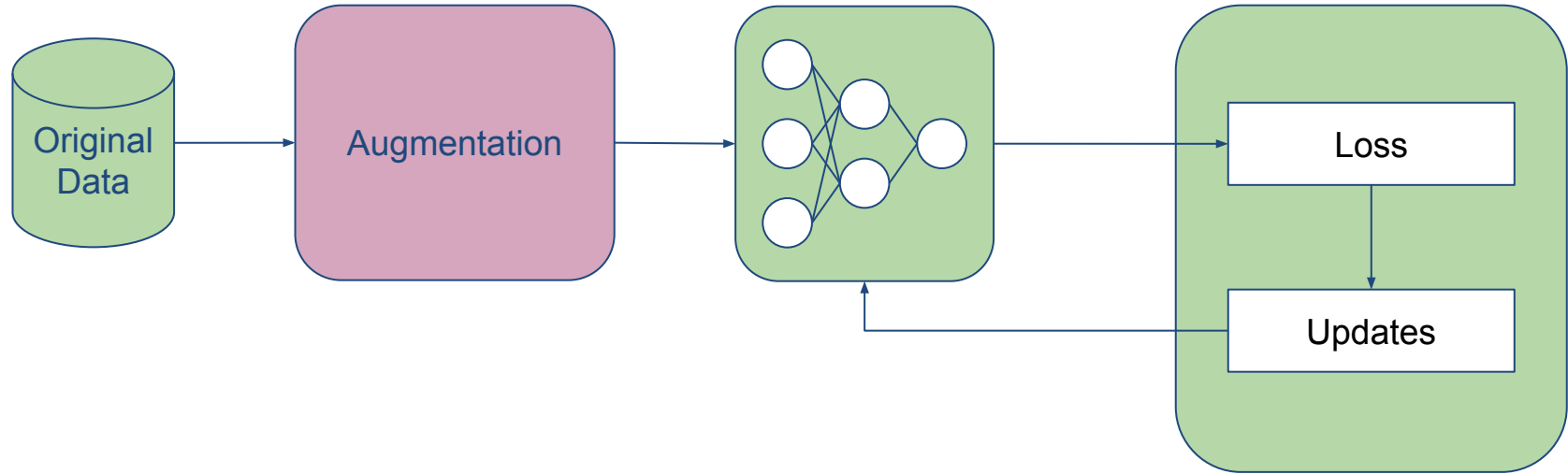
# Data augmentation - training baseline



# Data augmentation #1: offline

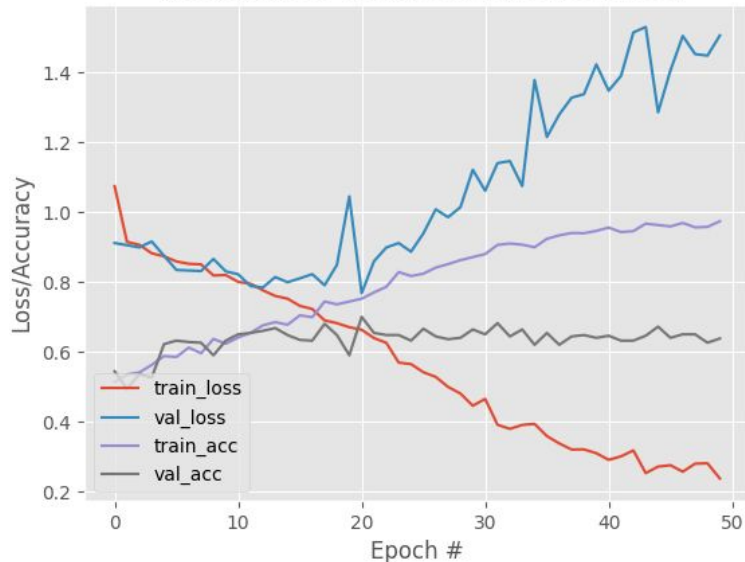


# Data augmentation #2: on the fly

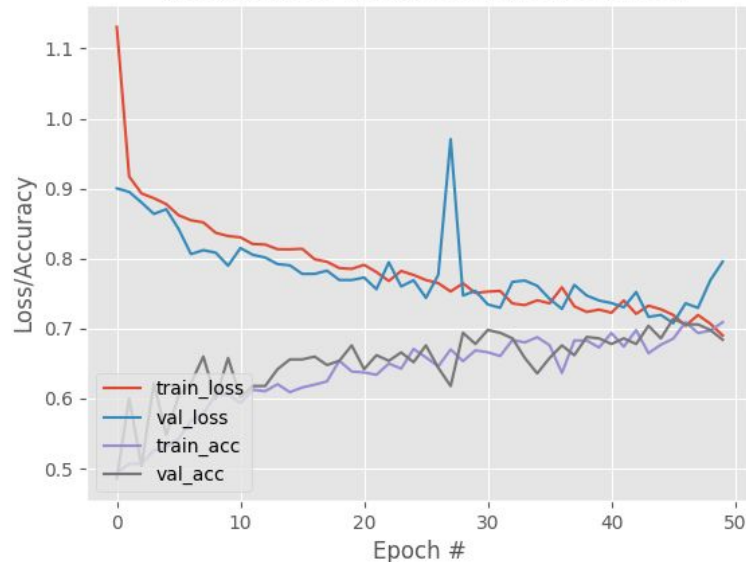


# Data augmentation effect

Training Loss and Accuracy on Dataset



Training Loss and Accuracy on Dataset



Credit: <https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/>



# On-the-fly data augmentation in keras

## 1) Organize your data properly

```
my_big_data_folder
├── train_set
│   ├── class_1
│   │   ├── Img1.jpg
│   │   ├── Img2.jpg
│   │   └── Img3.jpg
│   ├── class_2
│   │   ├── Img4.jpg
│   │   ├── Img5.jpg
│   │   └── Img6.jpg
│   ├── class_3
│   │   ├── Img7.jpg
│   │   ├── Img8.jpg
│   │   └── Img9.jpg
│   └── ...
├── test_set
│   └── <same structure, different images>
└── val_set
    └── <same structure, different images>
```





# Data augmentation in keras

## 2) Instantiate two/three ImageDataGenerator

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    horizontal_flip=True, vertical_flip=True,
    rotation_range=10, width_shift_range=0.2, height_shift_range=0.2,
    ...
)

val_datagen = ImageDataGenerator(rescale=1./255)
```



# Data augmentation in keras

## 3) Give the data to the generator

```
train_generator = train_datagen.flow_from_directory(  
    directory = 'my_big_data_folder/train_set',  
    target_size = image_shape,  
    batch_size = batch_size,  
    class_mode = 'categorical'  
    ...  
)
```

```
val_generator = val_datagen.flow_from_directory(  
    directory = 'my_big_data_folder/val_set',  
    target_size = image_shape,  
    batch_size = 5,      #ATTENTION HERE  
    class_mode = 'categorical'  
    ---  
)
```



# Data augmentation in keras

## 4) Train the model

```
history = model.fit(  
    x = train_generator,  
    validation_data = val_generator,  
    epochs = 50,  
    ...  
)
```



# Data augmentation in keras

- Not only from directory:
  - `<your_generator>.flow_from_dataframe(...)`
- Not only images...
  - `from keras.preprocessing.sequence import TimeseriesGenerator`
  - `keras.preprocessing.text...`
- ...but images have way more options



# Take home message

- Data augmentation is “free”
  - Extra computational burden is usually minimal
- It does NOT increase the training data size
  - Unless you explicitly do so (offline vs on-the-fly)
- It helps your network to generalize better
- Allows for more training epochs
- It's almost always a good idea



# [REF]

- Keras image data preprocessing:  
<https://keras.io/api/preprocessing/image/>
- The different kinds of data augmentation, implemented in a detailed example:  
<https://www.pyimagesearch.com/2019/07/08/keras-image-datagenerator-and-data-augmentation/>
- A gallery of image augmentation:  
<https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>

