

Data Science CBA Challenge '17: Understanding the Amazon from Space

Nicolas Peretti

`nicoperetti.16@gmail.com`

Pablo Pastore

`pabloapast@gmail.com`

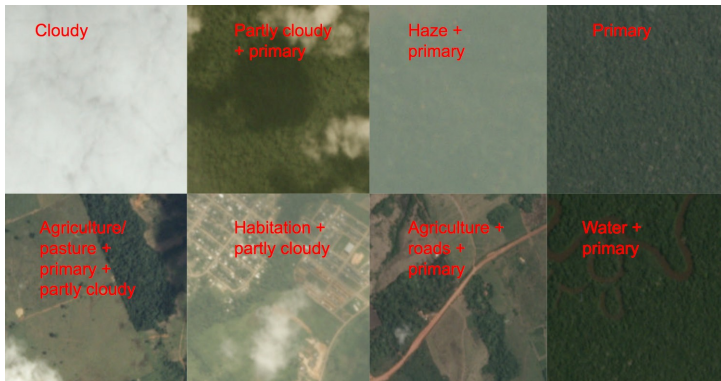
PP-CBA

14 de septiembre de 2017

- 1 Introducción
 - Descripción del problema
 - CNN
- 2 Datos
 - Análisis
 - Pre-procesamiento
- 3 Entrenamiento
- 4 Evaluación
- 5 Primeros Resultados
- 6 Alternativas
- 7 Herramientas

Introducción: Descripción del problema

- Objetivo: Medir y analizar el impacto de la deforestación a nivel global.
- Propuesta: Construir un algoritmo capaz de predecir el contenido en imágenes satelitales del amazonas.
- Datos:



- Poco conocimiento sobre el problema → Utilizar Redes Neuronales Convolucionales (**CNN**) para extraer una representación vectorial de la imagen.
- Desarrollar algoritmos de clasificación *multi-label*.

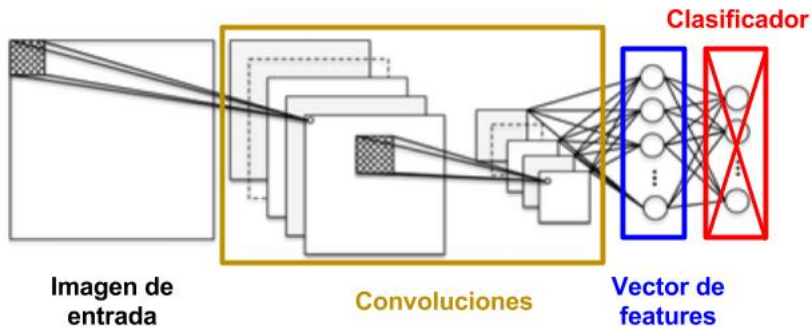


Figura: Red neuronal convolucional entrenada en ImageNet.

CNN: Freeze vs Tune

		Images per Class		
		L (1-20)	M (21-99)	H (≥ 100)
Cosine Distance	L (0.0-0.2)	Freeze	Try Freeze or Tune	Tune
	M (0.2-0.4)	Try Freeze or Tune	Tune	Tune
	H (0.4-1.0)	Try Freeze or Tune	Tune	Tune

Chu et al. *Best practices for fine-tuning visual classifiers to new domains*. ECCV'16.

- 40K imágenes anotadas para entrenamiento. Usaremos 80 %/20 % para entrenamiento/validación.
- 40K imágenes sin anotar para reportar resultados.

	image_name	tags
0	train_0	haze primary
1	train_1	agriculture clear primary water
2	train_2	clear primary
3	train_3	clear primary
4	train_4	agriculture clear habitation primary road

Figura: Imágenes con sus anotaciones.

Datos: Análisis

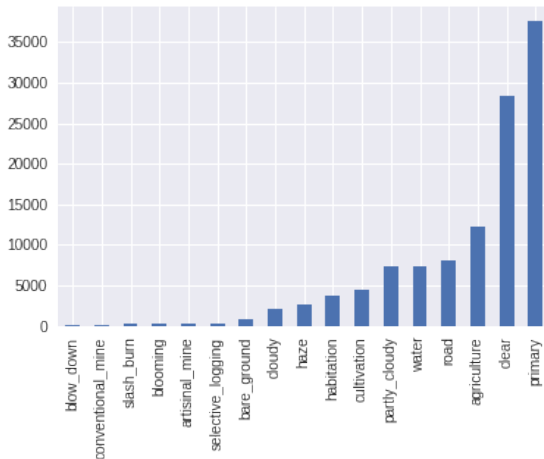


Figura: Cantidad de imágenes por etiqueta, datos desbalanceados, algunas clases con pocas imágenes.

Nos gustaría contar con mas datos de entrenamiento → **Data Augmentation**

- Se aplica *on the fly* durante el entrenamiento.
- Aplicar transformaciones aleatorias.
- Rotación, desplazamiento, zoom, recortes.
- También ayuda a evitar el sobreajuste (**overfitting**).

Datos: Pre-procesamiento



Figura: Imagen original: clear, agriculture, cultivation, primary, road, water.

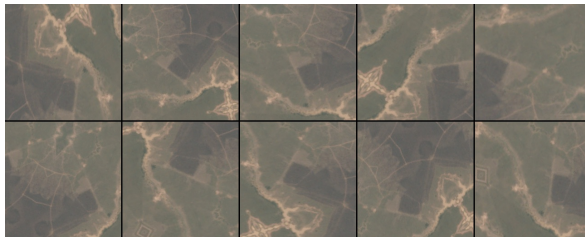
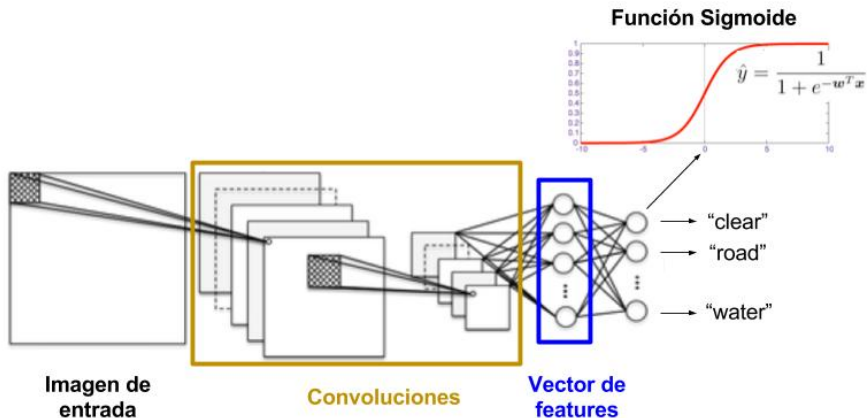


Figura: Con data augmentation.

Entrenamiento



Cross Entropy Loss

$$\mathcal{L}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Se evaluará con F_2 score, mide *accuracy* utilizando *precision* y *recall*. *Precision* es el ratio de positivos verdaderos (tp) sobre todos los positivos predichos ($tp + fp$). *Recall* es el ratio de positivos verdaderos (tp) sobre todos los positivos ($tp + fn$).

$$F_2 = (1 + \beta^2) \frac{pr}{\beta^2 p + r} \quad \text{donde} \quad p = \frac{tp}{tp + fp}, \quad r = \frac{tp}{tp + fn}, \quad \beta = 2.$$

Notar que F_2 le da mayor peso a *recall* sobre *precision*.

A las etiquetas de cada imagen las mapeamos a l^{17} , donde $l=0,1$
La salida del modelo propuesto es un vector $v \in R^{17}$, donde cada
 $0 < v_i < 1$

Para poder trabajar en el mismo dominio utilizamos la noción de umbral (thresholds)

$$F(x) = \begin{cases} 0, & x < thr \\ 1, & cc \end{cases}$$

Utilizamos las siguientes formas de setear este umbral.

- Uno global.
- Uno por clase.

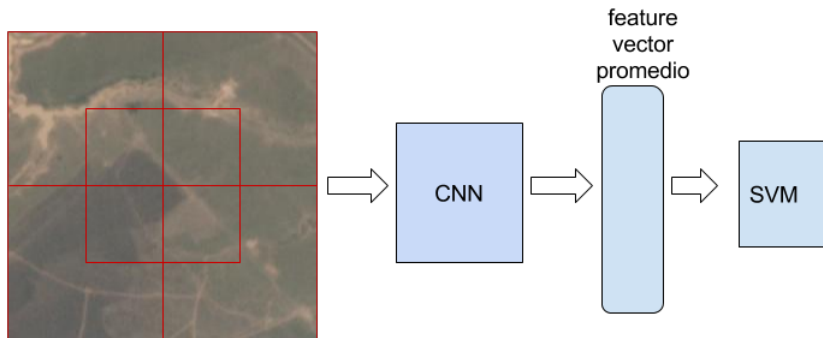
Primeros Resultados

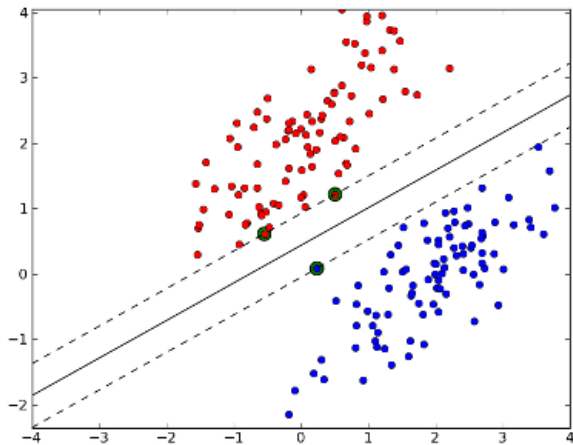
Arquitectura	F_2 Score		
	Val	Public LB	Private LB
SqueezeNet	0.6747	-	-
InceptionV4	0.9021	-	-
ResNet-50	0.9228	0.92452	0.92233
InceptionV3	0.9234	0.92472	0.92262
Xception	0.9233	0.92441	0.92295

Cuadro: Resultados y arquitecturas utilizadas durante el entrenamiento.

Alternativas: SVM

- 5 crops + imagen.
- Pipeline:





Modelo	parámetros	CNN	Crops	F2
LinearSVC	$C=1, \gamma = 1$	InceptionV3	Si	0.8901
SVC	rbf, $C=1000, \gamma = 0,01$	InceptionV3	Si	0.8980
SVC	rbf, $C=850, \gamma = 0,003$	InceptionV3	No	0.9011

Cuadro: Resultados.

Alternativas: Ensamble

Idea original: Corrección de errores hacia adelante (en inglés, **Forward Error Correction**)

Example (Repetition code)

Señal original:

1110110011

Codificada:

10,3 101011001111101100111110110011

Decodificado:

1010110011

1110110011

1110110011

Votación:

1110110011

Ensamble

Supongamos que tenemos 3 clasificadores binarios A, B, C todos con un 0.70 de *accuracy*.

3 predicciones correctas

$$0.7 * 0.7 * 0.7 \\ = 0.3429$$

2 predicciones correctas

$$0.7 * 0.7 * 0.3 \\ + 0.7 * 0.3 * 0.7 \\ + 0.3 * 0.7 * 0.7 \\ = 0.4409$$

Total

$$0.3429 + 0.4409 \\ = 0.7838$$

Example (3 clasificadores binarios fuertemente correlacionados)

Resultado esperado:

1111111111

Predicciones:

1111111100 = 0.8 accuracy

1111111100 = 0.8 accuracy

1011111100 = 0.7 accuracy

Votación:

1111111100 = 0.8 accuracy

Example (3 clasificadores binarios descorrelacionados)

Resultado esperado:

1111111111

Predicciones:

1111111100 = 0.8 accuracy

0111011101 = 0.7 accuracy

1000101111 = 0.6 accuracy

Votación:

1111111101 = 0.9 accuracy

+ Info: <https://mlwave.com/kaggle-ensembling-guide/>

Ensamble: Resultados

Arquitectura	F_2 Score		
	Val	Public LB	Private LB
SqueezeNet	0.6747	-	-
InceptionV4	0.9021	-	-
ResNet-50	0.9228	0.92452	0.92233
InceptionV3	0.9234	0.92472	0.92262
Xception	0.9233	0.92441	0.92295
Ensamble	0.92956	0.92982	0.92777

Cuadro: Resultado final utilizando ensamble.

Herramientas



pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Keras



NumPy



DEEPVISION

Muchas gracias!
Preguntas?