

TFG – Arquitectura de computadors i sistemes operatius

FITA#05: Disseny de l'arquitectura de xarxes neuronals

Gestió del projecte a GitHub:

Branca del repositori:

<https://github.com/UOC-Assignments/uoc.tfg.jbericat/tree/FITA%2305>

Dashboard de seguiment de les tasques associades a la fita:

<https://github.com/UOC-Assignments/uoc.tfg.jbericat/projects/5>

Estudiant: Jordi Bericat Ruz

Professor col·laborador: Daniel Rivas Barragan

Semestre: Tardor 2021/22 (Aula 1)

Versió: ESBORRANY_v2

Índex

5 - Disseny de l'arquitectura de xarxes neuronals	1
5.1 - Tasques d'investigació i recerca	1
5.2 - Estructuració de l'algorisme de DL i configuració d'hyperparàmetres	2
5.2.1 – Identificació / definició de les mètriques de rendiment.....	2
5.2.2 - Establiment / disseny d'un model base (baseline model)	6
5.2.3 - Preparació de les dades per a l'entrenament del model.....	7
5.2.3.1 - Pre-processament de les imatges del dataset	7
5.2.3.2 - Divisió dels dataset en grups.....	9
5.3 - Avaluació del model i interpretació del seu rendiment	11

5 - Disseny de l'arquitectura de xarxes neuronals

5.1 - Tasques d'investigació i recerca

Donat que la implementació d'un model DCNN específic per a resoldre el problema proposat a la PdC d'aquest projecte no és una tasca gens trivial i que a més sobrepassa en complexitat els objectius marcats per al TFG, aleshores es decideix fer recerca de diferents models DCNN i metodologies d'entrenament i implementació ja existents per a avaluar si ens poden ser d'utilitat i així evitar haver de fer un disseny simplificat de xarxa neuronal profunda. En aquest sentit, s'ha trobat que els capítols 4 i 5 de la referència bibliogràfica **LLIBRE DL** ens poden servir de guió per a d'una banda preparar les dades (*datasets* imatges) per a ser processades pel model DCNN, i de l'altra per a avaluar i seleccionar una arquitectura DCNN ja existent per al seu ús en la PdC.

A més, s'ha complementat la informació obtinguda de la font anterior amb les que ofereix el següent enllaç:

<https://iq.opengenus.org/precision-recall-sensitivity-specificity/>

5.2 - Estructuració de l'algorisme de DL i configuració d'hyperparàmetres

5.2.1 – Identificació / definició de les mètriques de rendiment¹

Definir les mètriques que s'utilitzaran per a mesurar el rendiment del model durant el seu entrenament és una tasca imprescindible per a poder monitoritzar si els canvis que es van realitzant en la configuració dels paràmetres a les diferents iteracions o *epoch* influeixen d'una manera o altra en la precisió (*accuracy*) de les prediccions realitzades pel model².

Per a establir doncs un marc en el qual puguem definir aquestes mètriques es representarà el rendiment del model DCNN mitjançant una “**taula de confusió**” (mètode de representació de mesures d'encert en la predicció utilitzat habitualment durant l'entrenament de models de ML i DL).

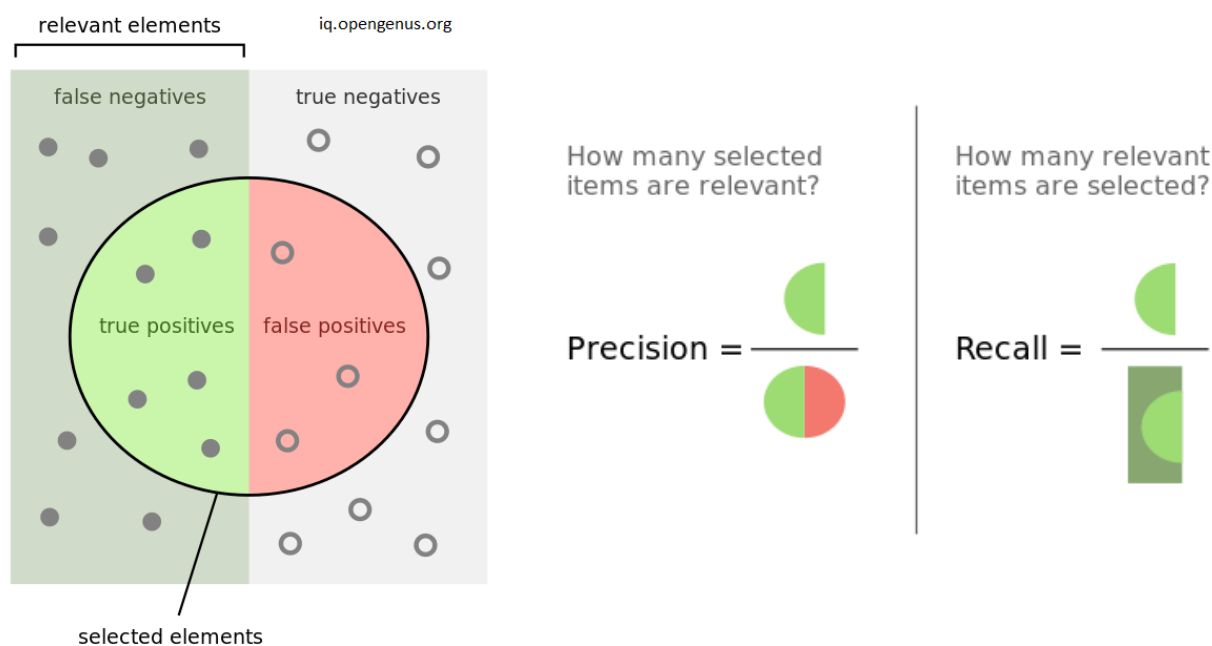
Simplificant, l'objectiu al representar el rendiment del model mitjançant d'una taula de confusió és el de poder centrar la mesura de l'encert de les prediccions en funció de, o bé la proporció de falsos positius (falses deteccions d'incendis) o bé de la de falsos negatius (no detecció d'un incendi). Així en funció de la importància de que es produeixi un o altre cas en el projecte de DL que s'estigui desenvolupant, caldrà tenir més cura de controlar els falsos positius (mesurats mitjançant la *precisió*), o bé els falsos negatius (mesurats mitjançant la *sensitivitat* o *recall*)

iq.opengenus.org		Predicted Class	
		NO	YES
Actual Class	NO	True Negative (TN)	False Positive (FP)
	YES	False Negative (FN)	True Positive (TP)

Imatge obtinguda de: <https://iq.opengenus.org/precision-recall-sensitivity-specificity/>

¹ <https://iq.opengenus.org/precision-recall-sensitivity-specificity/>

² Pàgina 148 del llibre de DL → “Defining the model evaluation metric is a necessary step because it will guide your approach to improving the System. Without clearly defined mètrics, it can be difficult to tell whether changed to a ML System result in progress or not”.



Imatges obtingudes de: <https://iq.opengenus.org/precision-recall-sensitivity-specificity/>

Així doncs, fetes les consideracions anteriors, en aquest punt cal decidir quina de les possibles falses prediccions pot implicar “pitjors” conseqüències a l'hora de predir si una imatge conté o no característiques pròpies d'un incendi forestal (objectiu central de la PdC d'aquest TFG).

Concretament, només es poden produir dos tipologies de falses prediccions:

- **Errada de tipus I: Fals positiu (*False Positive* o *FP*)**

Un “fals positiu” significa que s'ha reconegut / identificat un incendi a una imatge però en realitat no ho és (p.e. en el cas de la PdC, es podria donar el cas que un cos o objecte que emeti calor es detecti com la característica d'un incendi, per exemple, una pila de compost orgànic)

- **Errada de tipus II: Fals negatiu (*False Negative* o *FN*)**

Un fals negatiu significa que, havent-t'hi un incendi forestal a la imatge processada per l'algorisme d'aprenentatge profund, aquest no ha reconegut cap característica pròpia d'aquests.

Per tant, si tenim en compte que la detecció d'un incendi a les seves fases inicials és un factor crític que pot implicar la seva extinció abans que aquest causi danys a major escala, aleshores podem

concloure que **tindrà pitjors conseqüències que es produeixi una predicció errònia de tipus II (FN)** que no pas de tipus I (FP). Podem determinar doncs que la mètrica de rendiment que ens permetrà identificar millor, o tenir una visió més precisa sobre la proporció d'incendis no detectats (FN) serà la basada en la **sensitivitat (*sensitivity / recall*)** o rati de positius veritaders (*True Positives o TP*). Dit d'una altra manera, la sensitivitat com a mètrica de rendiment ens permetrà saber amb millor exactitud que la precisió quantes vegades s'ha produït un FN durant cada iteració (*epoch*) d'entrenament del model, fet que ens ajudarà a fer un reajustament més precís dels paràmetres (pesos de les arestes del graf) i en conseqüència a obtenir un model final completament entrenat que mostri un rendiment més acurat (*model performance & accuracy*).

Podem calcular la sensitivitat de les dades mesurades amb la següent expressió:

$$\text{sensitivitat} = \frac{TP}{TP + FN}$$

Un cop realitzat l'estudi anterior per a determinar la millor manera de mesurar el rendiment del model a l'hora de resoldre el problema binari de classificar imatges en funció de si contenen o no característiques d'incendis, passem a ara a estudiar el cas en el què el resultat de resoldre el problema de classificació ha de ser la categorització d'un incendi en funció de les seves característiques, tal com es defineix a la següent taula:

Categoria d'incendi (Intensitat x Mida)	Localitzat	Moderadament estès	Molt estès
Intensitat Baixa	A	B	C
Intensitat Moderada	D	E	F
Intensitat Alta	G	H	I

Abans però caldrà fer notar que, tenint en compte que s'està desenvolupant una PdC, per a simplificar considerarem les següents prioritats (essent "A" el nivell o categoria menys crítica i "I" el nivell o categoria més crítica) sense tenir en compte altres factors, com podria ser la proximitat a zones habitades:

$$A < B < C < D < E < F < G < H < I$$

En aquest nou escenari podem definir:

- **Errada de tipus I: Fals positiu (*False Positive* o *FP*)**

Ara, un “fals positiu” significa que havent processat una imatge que conté un incendi de categoria inferior (o bé sense cap incendi), el model n’ha identificat un, i de categoria superior si la imatge en contenia característiques.

- **Errada de tipus II: Fals negatiu (*False Negative* o *FN*)**

Ara, un fals negatiu significa que, havent processat una imatge que conté un incendi de categoria superior, el model n’ha identificat un de categoria inferior, o bé no n’ha identificat cap.

Observem que, de la mateixa manera que passava amb el problema de classificació binari, la mètrica més adequada serà la que ens permeti tenir una visió més acurada dels falsos negatius (FN) que es produeixin durant l’entrenament del model, és a dir, la **sensitivitat** de les mesures. Com a conseqüència o *side-effect* però, tindrem que amb aquesta mètrica no tindrem dades tant precises de la quantitat de falsos positius que es produeixin. Tanmateix, a la pràctica en el cas que ens ocupa això només podria suposar que en un moment donat algun dels drons de l’eixam intenti informar de / apagar un incendi inexistent, o bé que s’hi dediquin més recursos dels necessaris per a gestionar la situació, com a conseqüència d’un entrenament poc acurat del model. Així doncs, un cop realitzades les primeres mesures amb la mètrica seleccionada (sensitivitat o *recall*) contra el *dataset* d’exploració (veure [secció 5.3](#)), si s’observa un rendiment pobre en el cas de falsos positius el que caldrà fer serà tornar a fer l’entrenament del model utilitzant una mètrica menys conservadora, però que ens doni una visió més precisa dels falsos positius (sota risc de què es comencin a detectar casos de falsos negatius, cas en el que s’hauria d’avaluar si paga la pena el canvi de mesura de rendiment donades les fatals conseqüències que pot tenir **una sola** predicció errònia de tipus II en un incendi forestal). En aquest sentit, la mètrica basada en *F-Score*³ és una possibilitat que cal tenir en compte i que s’estudiarà en funció dels resultats (*accuracy*) de les mesures basades en la sensitivitat (*recall*).

Podem calcular el *F-Score* de les dades mesurades de la següent manera:

³ Refs: Punt 4.1.4, pàg. 149 llibre DL

$$r = \text{sensitivitat} = \frac{TP}{TP + FN}$$

$$p = \text{precisió} = \frac{TP}{TP + FP}$$

$$F - \text{Score} = \frac{2pr}{p + r}$$

5.2.2 - Establiment / disseny d'un model base (*baseline model*)

5.2.3 - Preparació de les dades per a l'entrenament del model

Abans de procedir a realitzar l'entrenament del model escollit a la secció anterior ([5.2.2 – Disseny d'un model base](#)), hem de:

- a) Pre-processar els dataset d'imatges “en cru o *raw*” que s'han generat com a resultat de les accions realitzades a la secció 4 d'aquest document
- b) Dividir els mateixos dataset en 3 grups: Entrenament, validació i explotació

Tot seguit es detalla en què consisteixen les accions de preparació anterior, així com la manera com s'aplicaran a les dades de què disposem per a la PdC.

5.2.3.1 - Pre-processament de les imatges del dataset

El pre-processament i “neteja” de les dades que alimentaran el model de xarxes neuronals es un pas previ prescindible quan es tracta d'entrenar models DL, però important si es tracta de refinar al màxim la seva velocitat d'aprenentatge i rendiment. Per tant aplicarem les estratègies més habituals de preprocessament a les imatges dels dataset de la PdC.

a) Imatges en color vs. escala de grisos

Coneixent que per a representar una imatge en color RGB amb una estructura de dades calen tres matrius (una per a cada color base), mentre que una imatge en escala de grisos la podem representar amb una sola matriu, aleshores podem concloure que utilitzant imatges en escala de grisos “estalviarem” complexitat computacional a l'hora d'entrenar el model, ja que la quantitat de paràmetres que haurem de configurar serà molt menor (això és, els pesos de les d'arestes del graf que formen els nodes de les diferents capes ocultes de la xarxa neuronal profunda). En el cas d'aquesta PdC, com que ja estem utilitzant imatges que simulen visió IR tèrmica ([veure secció 3.x.x.x](#)) i aquestes estan convertides al B/N per a aconseguir una simulació més fidel, aleshores no haurem de prendre cap acció i simplement considerarem que les imatges ja han sigut pre-processades en aquest sentit. Cal notar que la utilització d'imatges en escala de grisos sempre serà convenient si es poden reconèixer les seves característiques utilitzant la visió humana, per tant en el cas dels dataset d'aquesta PdC clarament ens podrem beneficiar de l'estalvi en complexitat.

b) Redimensionament de les imatges

Quan estem entrenant xarxes convolucionals profundes (DCNN), igual com passa amb les xarxes de perceptrons multicapa (MLP), hem de tenir cura de que totes les imatges dels dataset tinguin la mateixa mida, ja que la capa d'entrada (input Layer) de la xarxa neuronal contindrà tants nodes com píxels conté la imatge. En el nostre cas, estem utilitzant imatges amb una resolució relativament alta (640x512) --→ **AQUÍ S'HA D'HAVER DECIDIT PRIMER QUINA DCNN S'UTILITZARÀ (SECCIÓ 5.2.2), PERQUÈ EN FUNCIÓ D'AIXÒ HAUREM DE REDUÏR O NO LA MIDA DE LES IMATGES (MENTRE MAJOR MIDA DE LES IMATGES, MÉS PROFUNDA HAURÀ DE SER LA XARXA NEURONAL I MAJORS RECURSOS EN TERMES DE GPU NECESSITAREM PER A PROCESSAR CADA IMATGE):**

https://www.researchgate.net/post/Which_Image_resolution_should_I_use_for_training_for_deep_neural_network

“So the rule of thumb is use images about 256x256 for ImageNet-scale networks and about 96x96 for something smaller and easier. I have heard that in kaggle people train on 512x512 sometimes, but you will need to compromise on something. Or just buy gpu cluster. If you train fully convolutional networks like Faster RCNN you can take much bigger images (say 800x600) because you have batch size = 1.”

c) Estandardització de les dades

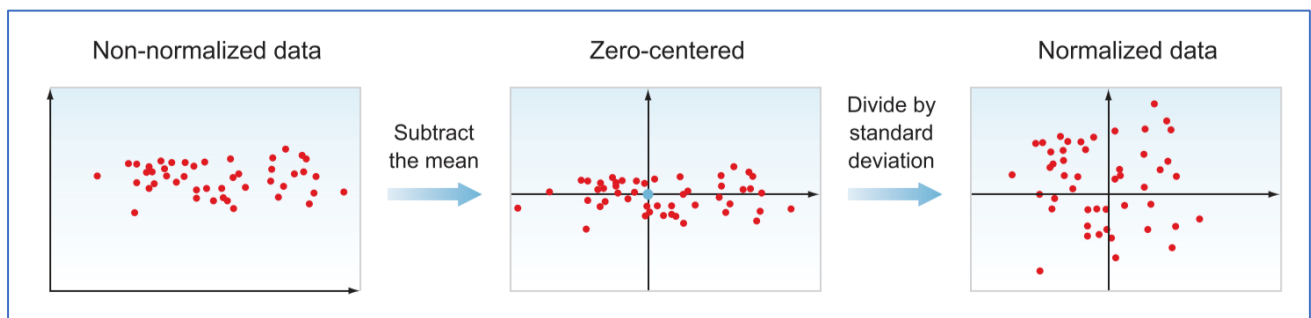
La estandardització o normalització de dades en termes estadístics permet, donat un conjunt de mostres (en el cas que ens ocupa, entenem com a mostres els píxels d'una imatge que podem representar en una variable amb valors x_1, x_2, \dots, x_n), localitzar amb la menor complexitat computacional temporal possible una dada en relació amb la seva mitjana, així com comparar els valors de les observacions de conjunts diferents, o dit d'una altra manera, comparar el valor dels píxels corresponents a imatges diferents⁴.

Per a calcular el valor estandarditzat o Z-Score d'una observació (píxel) donada la variable $x = x_1, x_2, \dots, x_n$, amb mitjana \bar{x} i desviació típica s_x , només cal restar la mitjana al valor de la observació i dividir-lo per la desviació típica, tal que:

⁴ Referències bibliogràfiques: Pàgs. 44-46, materials de l'assignatura “Estadística”, mòdul “Estadística descriptiva: Introducció a l'anàlisi de dades”, Àngel J. Gil Estallo (UOC)

$$x_i \xrightarrow{\text{Estandarditzar}} z_i = \frac{x_i - \bar{x}}{s_x}$$

D'aquesta manera, si estandarditzem les dades obtindrem com a resultat que el valor de tots els píxels de cada imatge es trobarà en el rang [0,1] de valors, que a efectes pràctics ens permetrà accelerar el procés d'aprenentatge de la xarxa neuronal⁵. La següent figura mostra la distribució uniforme de les dades aconseguida un cop realitzat el procés d'estandarització:



Imatge obtinguda de [Ref. Llibre DL], pàg. 155

Implementació:

TO-DO

d) Augmentació de les imatges

TO-DO

5.2.3.2 - Divisió dels dataset en grups

Un cop s'hagin aplicat les diferents tècniques de pre-processament d'imatges als dataset per tal d'adaptar-los als requeriments de l'arquitectura DCNN, el que haurem de fer és dividir cada dataset en tres grups o subconjunts diferents;

⁵ P. 154 Llibre DL → "Although not required, it is preferred to normalize the pixel values to the range of 0 to 1 to boost learning performance and make the network converge faster."

- a) Grup d'entrenament (*training dataset*) → 75% - 80%
- b) Grup de validació (*validation dataset*) → 15% - 20%
- c) Grup d'explotació (*test dataset*) → 15% - 20%

Bloc XX: Divisió dels dataset i la seva proporció

De manera que utilitzarem el subconjunt d'entrenament a) per a entrenar el model tot ajustant el pes de les arestes que connecten les capes ocultes (hidden layers) de la DCNN; el subconjunt de validació b) per a contrastar el rendiment del model **durant l'entrenament** (de manera periòdica, cada certes *epoch* o iteracions d'entrenament) contra un subconjunt de mostres “desconegut” per al model per a comprovar el nivell de generalització del coneixement adquirit pel model durant la etapa d'entrenament i realitzar els ajustos corresponents als pesos de les arestes per a millorar el seu rendiment; i finalment, el subconjunt d'explotació c) ens servirà com a “prova de camp” per a avaluar el rendiment final del model **un cop completat l'entrenament**.

Pel que fa la proporció de mostres dels dataset que s'assignen a cada subconjunt tindrem en compte que, per norma general, si el dataset està comprés per una quantitat relativament gran de mostres (de l'ordre dels milions) aleshores amb un 1% de les mostres per als subconjunts de validació i d'explotació ja n'hi haurà prou, de manera que reservarem tota la resta per a efectuar l'entrenament i així aconseguir una major precisió de les prediccions del model. En el cas de la PdC que estem desenvolupant però, els dataset contindran una quantitat d'imatges molt inferior (de l'ordre dels centenars o milers), de manera que els ratis habitualment utilitzats per a l'entrenament de models ML seran més convenients (veure **bloc XX**).

5.3 - Avaluació del model i interpretació del seu rendiment