UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

DOBLE GRAU DE MATEMÀTIQUES I ENGINYERIA INFORMÀTICA

Treball final de grau

UN NOU MARC PER ANALITZAR TRANSFORMERS DE VISIÓ. APLICACIÓ A L'ANÀLISI D'IMATGES DE MENJAR

Autor: David Rial Figols

Directors :	**Dra. Petia Radeva i Dr. Joan Carles Tatjer**
Tutor:	**Marcos Mejía Córdova**
Realitzat a:	**Departament de Matemàtiques i Informàtica**

Barcelona, 24 de gener de 2022

Abstract

The field of Deep Learning is constantly evolving to optimize the models that are developed and achieve a specific task with the best possible accuracy. It was in 2017 when Vaswani introduced a new neural network structure that would allow for a new evolution: transformers. Based on the concept of attention, introduced in 2014, transformers were able to quickly impose themselves on all-natural language processing tasks. It was not until 2020 that transformers, applied to image-related tasks started to be competitive. Furthermore, within less than two years they have been able to overcome the previous models of neural networks architectures, to end up prevailing with the best results. Among these image tasks, the problem of image classification stands out, which is to assign each image a label that describes it, as it is a problem that has historically been used to describe the evolution of Deep Learning and the progress made.

Despite being at the forefront of all the tasks mentioned, transformers, especially image transformers, are still a black box as to why they learn or what makes one transformer better than another. It is for this reason that this work is based on the study of transformers. Specifically, this paper aims to introduce the transformers and the basics needed to understand how they work, and then to understand and investigate how transformers learn dedicated to image classification, looking for their similarities and differences, and how they are characterized.

In this work, we have proposed a comparative framework for transformers dedicated to the problem of image classification, based on the results achieved with the transformers ViT, BEiT, DeiT, Swin and CSWin on the set of food images Food-101. This comparison framework is based on the different properties and evolution of the various weight matrices that make up these transformers, aided by the unique values of the matrices and their rules and ranges. At the same time, a form of training is also suggested to make it faster in a specific data set, reducing the time by 33% without losing accuracy.

2020 Mathematics Subject Classification. 68T07

Resum

El camp del Deep Learning está evolucionant constantment per tal d'optimitzar els models desenvolupats i aconseguir realitzar una tasca concreta amb la millor precisió possible. És al 2017 quan Vaswani va presentar una nova arquitectura de xarxa neuronal que permetria una nova evolució: els transformers. Basats en el concepte d'atenció introduït al 2014, els transformers van aconseguir rápidament imposar-se en totes les tasques de processament de llenguatge natural. No va ser fins al 2020 que van desenvolupar-se transformers dedicats a tasques amb imatges els quals fossin realment competitiu, però també han estat capaços de superar, en menys de dos anys, els models anteriors de xarxes neuronals per acabar imposant-se amb els millors resultats. Dins d'aquestes tasques amb imatges, destaca el problema de classificació d'imatges que consisteix en assignar a cada imatge una etiqueta que la descrigui. La classificació d'imatges és un problema que històricament ha servit per descriure l'evolució del Deep Learning i els avenços realitzats. En aquest moment, s'ha convertit en la línia repte de tot el Deep Learning ontenint els millors resultats en diferentsproblemes i datatses de Visió per Computador.

Tot i el fet d'estar al capdavant en totes les tasques mencionades, els transformers, sobretot els transformers dedicats a imatges, són encara una caixa negra respecte al motiu pel qual aprenen o què comporta que un transformer sigui millor a un altre. És per aquest motiu que aquest treball es basa en l'estudi dels transformers. Concretament, l'objectiu d'aquest treball és introduir els transformers i les nocions bàsiques necessàries per entendre el seu funcionament per, a continuació, entendre i investigar sobre com aprenen els transformers dedicats a la classificació d'imatges, buscant les seves similituds i diferències, i en què es caracteritzen.

En aquest treball, hem plantejat un nou marc per explorar el procés d'aprenentatge dels transformers que ens serveix per fer comparació dels transformers dedicats al problema de la classificació d'imatges. Hem estudiat cinc dels transformers més potents en aquest moment, especialment els transformers ViT, BEiT, DeiT, Swin i CSWin i la seva aplicació sobre un dataset d'imatges de menjar, nomenat Food-101. Aquest marc de comparació es basa en les diferents propietats i l'evolució de les diverses matrius de pesos que conformen aquests transformers, ajudant-se dels valors singulars de les matrius i de les seves normes i rangs. Alhora, es proposa una forma d'entrenament per tal de fer-lo més ràpid en un conjunt de dades específic, aconseguint reduir el temps en un 33% sense perdre precisió. Els resultats sobr eel reconeixement de menjar són molt prometedors.

Resumen

El campo del Deep Learning está evolucionando constantemente para optimizar los modelos desarrollados y conseguir realizar una tarea concreta con la mejor precisión posible. Es en 2017 cuando Vaswani presentó una nueva estructura de red neuronal que permitiría una nueva evolución: los transformers. Basados en el concepto de atención introducido en 2014, los transformers consiguieron rápidamente imponerse en todas las tareas de procesamiento de lenguaje natural. No fue hasta el 2020 cuando aparecieron transformers dedicados a tareas con imágenes que fueran realmente competitivos, pero también han sido capaces de superar, en menos de dos años, los modelos anteriores en las redes neuronales para imponerse con los mejores resultados. Dentro de estas tareas con imágenes, destaca el problema de clasificación de imágenes que consiste en asignar a cada imagen una etiqueta que la describa, puesto que es un problema que históricamente ha servido para describir la evolución del Deep Learning y los avances realizados.

A pesar de estar al frente en todas las tareas mencionadas, los transformers, sobre todo los transformers dedicados a imágenes, son todavía una caja negra respecto al motivo por el que aprenden o qué comporta que un transformer sea mejor a otro. Por este motivo, este trabajo se basa en el estudio de los transformers. Concretamente, el objetivo de este trabajo es introducir los transformers y las nociones básicas necesarias para entender su funcionamiento para, a continuación, entender e investigar sobre cómo aprenden los transformers dedicados a la clasificación de imágenes, buscando sus similitudes y diferencias , y en que se caracterizan.

En este trabajo, hemos planteado un marco de comparación de los transformers dedicados al problema de la clasificación de imágenes, basándonos en los resultados conseguidos con los transformers ViT, BEiT, DeiT, Swin y CSWin en el conjunto de imágenes de comida Food -101. Este marco de comparación se basa en las distintas propiedades y la evolución de las diversas matrices de pesos que conforman estos transformers, ayudándose de los valores singulares de las matrices y de sus normas y rangos. Asimismo, también se propone una forma de entrenamiento para hacerlo más rápido en un conjunto de datos específico, consiguiendo reducir el tiempo en un 33% sin perder precisión.

Agraïments

En la realització d'aquest treball, he tingut la inestimable ajuda de moltes persones les quals m'han ajudat en tots els aspectes a tirar-lo endavant. És per aquest motiu que els vull agrair a tots el suport moral i acadèmic que m'han proporcionat.

Primer de tot, als dos directors d'aquest projecte: la Dra. Petia Radeva i el Dr. Joan Carles Tatjer. La Dra. Petia Radeva va ser qui em va proposar i engrescar en aquest projecte. Ella m'ha posat en contacte amb la resta de persones involucrades en aquest projecte i m'ha transmès els nervis i emoció inherents en la recerca acadèmica quan es busca anar més enllà. Per altra banda, el Dr. Joan Carles Tatjer m'ha aportat la seva visió més teòrica i m'ha motivat a intentar definir amb una vessant matemàtica els diversos conceptes treballats.

Agrair també al meu tutor Marcos Mejía per la seva paciència i coneixements. Amb la seva ajuda en la implementació i la seva disponibilitat a quasi qualsevol hora, m'ha ajudat a no desanimar-me quan el codi no funcionava i a motivar-me per seguir amb el projecte.

No puc acabar aquests agraïments sense mencionar també la meva família, per aguantar els meus nervis i estrès i donar-me un suport constant i necessari, i a la empresa Unica360 per posar-me totes les facilitats possibles per poder realitzar aquest treball.

Per últim, agrair al Dr. Vicent Ribas, al doctorant Giuseppe Pezzano i a tots els meus companys i coneguts que m'han ajudat a polir aquest treball. Tots ells han aportat i millorat aquest treball i han ajudat a que avui aquest treball de final de grau sigui una realitat.

Índex

1 Introducció

Tot seguit s'introdueix l'objecte d'estudi d'aquest treball, juntament amb les tasques que s'han desenvolupat per portar-lo a terme. Finalment, s'elabora l'estructura de la memòria.

El projecte

Actualment, les xarxes neuronals estan presents en el nostre dia a dia, des de les que estan presents en les xarxes socials fins a les que podem trobar en quasi qualsevol dispositiu electrònic actual. Són tan importants que fins i tot els mitjans de comunicació es fan ressò de cada avenç que es realitza en el camp del Deep Learning. És per aquest motiu que en aquest projecte aprofundirem en l'avenç més recent: els transformers.

Des de la publicació, ja històrica, del famós article "*Attention is all you need*"[56], els transformers han revolucionat tots els camps del Deep Learning (DL), des del processament del llenguatge natural, com pot ser el GPT-3 [6], fins a, més recentment, el camp de la visió. Aquests, aprofitant la ingent quantitat d'imatges disponibles a la web, optimitzen el seu rendiment per aconseguir millorar els seus resultats en detecció d'objectes i segmentació i classificació d'imatges, entre d'altres.

El focus principal d'aquest projecte és explorar l'aplicació dels transformers des del punt de vista de la seva convergència i aplicabilitat per atacar el problema del reconeixement d'imatges de menjar.

Motivació

El Deep Learning és una ciència que es troba en constant evolució, una evolució que sovint avança més ràpidament que les bases matemàtiques que fomenten les arquitectures dels seus models. Aquest és un camp que evoluciona a partir de resultats i d'idees de grans científics que, al desenvolupar-se i veure que donen resultats competitius, s'accepten i provoquen un avanç al següent nivell. Tot i les millores fruit d'aquesta evolució, és necessari ordenar i fonamentar tots els nous conceptes, per tal d'aplicar-los en el següent avenç amb certesa i coneixement de la correcta evolució.

Per altra banda, el problema de classificació d'imatges de menjar és un problema complex degut a les similituds inherents en molts menjars similars. Al mateix temps, té implicacions mèdiques relacionats amb la nutrició. Tot plegat, fa que sigui un problema interessant, exigent i ideal per la realització d'aquest treball.

Per tots aquests motius, en aquest treball els objectius són:

1. Assolir els coneixements necessaris en els quals se sostenen els transformers.
2. Donar una definició matemàtica del concepte d'atenció, un concepte essencial en els transformers, i explicar d'una forma clara i didàctica els conceptes i arquitectures dels principals transformers dedicats a la classificació d'imatges.
3. Proposar un marc per analitzar el procés d'aprenentatge dels transformers de visió.
4. Proposar heurístiques per optimitzar el procés d'aprenentatge dels transformers basades en formes de "congelar" alguns dels seus blocs. Demostrar que aquestes eines proporcionen transformers més estables i menys adients a l' "overfitting".

5. Demostrar que aquest marc serveix per a avaluar els diferents transformers utilitzant 5 dels transformers més potents i populars en aquest moment.

6. Entrenar i executar els diferents transformers mencionats, analitzant els seus resultats sobre el reconeixement d'imatges de menjar i comparar els resultats obtinguts en els entrenaments dels transformers.

Planificació del projecte

Aquest projecte va ser plantejat al setembre del 2021 i ha estat desenvolupat al llarg d'aquests mesos fins a la data d'entrega, 24 de gener del 2022.

Per començar en el projecte es va plantejar una primera etapa per entrar en contacte amb les xarxes neuronals, els transformers i l'entorn PyTorch, ja que són els conceptes indispensables per realitzar aquest projecte. Per entrar-hi en contacte, seria primordial el llibre "Dive Into Deep Learning"[71], un llibre en línia ideal per entrar en contacte amb el camp del DL ja que t'ofereix, apart dels conceptes teòrics, codi executable i exercicis per tal que, durant cada apartat, puguis no tan sols entendre els conceptes, sinó també aplicar-los.

Una vegada els coneixements inicials estiguessin establerts, investigaríem sobre quins transformers eren més interessants per aquest projecte basant-nos en les seves característiques i la seva utilització actual dins del DL. I un cop escollits els transformers, els exploraríem i executaríem sobre el conjunt d'imatges Food-101 per analitzar la seva convergència i evolució durant el procés d'aprenentatge basant-nos a diferents mètriques.

Finalment, s'encararia la recta final del semestre analitzant els resultats i realitzant els diversos experiments per acabar el projecte extraient les conclusions i validant-les.

Estructura de la Memòria

L'estructura de la memòria en aquest projecte està diferenciada en dues parts principals: una part teòrica en la qual es defineixen els diferents conceptes que utilitzen els transformers i s'introdueixen els diferents transformers que s'analitzen en aquest treball, així com les diferents mètriques per analitzar la seva convergència i evolució durant l'aprenentatge; i una part pràctica i descriptiva en la qual s'executen els cinc transformers i es realitzen els experiments que es plantegen en l'apartat 4.

Entrant en detall, els apartats amb els quals s'organitza aquesta memòria són:

2. **Estat de l'art**: breu explicació del problema de la classificació d'imatges i de la història i evolució dels transformers.

3. **Background teòric**: definició de tots els conceptes necessaris per entendre aquest treball i introducció dels diferents transformers que s'utilitzen al llarg del projecte.

4. **Un nou marc per analitzar el procés d'aprenentatge dels transformers de visió**. Es divideix en tres parts principals:

    4.1. **Les nostres hipòtesis sobre el procés d'aprenentatge dels transformers**: presentació dels reptes i les hipòtesis.

5. **Implementació de l'entorn per analitzar els transformers**: explicació de com s'ha elaborat el codi del projecte, juntament amb el conjunt d'imatges, els paràmetres i els models pre-entrenats utilitzats. Finalment, presentació i análisi dels resultats obtinguts.

6. **Conclusions**: valoració dels objectius assolits i la feina realitzada.

7. **Treball futur**: proposta de futures línies d'investigació que es podrien desenvolupar a continuació d'aquest treball.

8. **Bibliografia**: recopilació dels articles i materials utilitzats.

2 Estat de l'art

En aquest apartat s'introduirà el problema de la classificació d'imatges i es presentaran alguns dels últims avenços en aquest problema. A continuació, s'explicaran breument les xarxes convolucionals (CNN) per seguidament posar èmfasis en com van sorgir els transformers i la seva evolució fins l'actualitat, ressaltant a on han arribat actualment en el camp del Deep Learning i, en concret, en el camp de la Visió Artificial (VA).

2.1 Classificació d'imatges

Classificar imatges ha estat, des de l'inici de la VA, un tema molt transcendent en el qual cada innovació ha aportat i ha implicat grans avenços. El problema en sí es redueix en una simple premissa: etiquetar cada imatge d'un conjunt de dades amb la seva categoria corresponent.

Si mirem el cronograma del conjunt de dades ImageNet [15], un conjunt de 14.197.122 imatges publicat al 2009, observem com al 2011, el model AlexNet [34] obtenia una precisió de 63.3% utilitzant una xarxa CNN. 8 anys després, CoAtNet-7 [13] ha obtingut una precisió de 90.88% i 7 models més han obtingut, tots en aquest últim any, una precisió superior al 90%. El fet que d'aquest top 8, la meitat són transformers i només dos [45] no implementen de cap manera el concepte d'atenció ja ens indica la gran presència dels transformers en aquest problema.

Rank	Model	Acc[2]
1	**CoAtNet-7** [13]	**90.88**
2	**ViT-G/14** [70]	**90.45**
3	**CoAtNet-6/14** [13]	**90.45**
4	**ViT-MoE-15B** [49]	**90.35**
5	**MetaPseudo Labels** [45]	**90.2**
6	**SWinV2-G** [40]	**90.17**
7	**Florence-CoSwin-H** [69]	**90.05**

Rank	Model	Acc
1	**ViT-H/14** [18]	**99.50±0.06**
2	**CaiT-M-36U 224**[55]	**99.4**
3	**CvT-W24** [64]	**99.39**
4	**BiT-L** [32]	**99.37**
5	**CeiT-S** [68]	**99.1**
6	AutoFormer-S—384 [10]	99.1
7	**TNT-B** [24]	**99.1**

Rank	Model	Acc
1	EffNet-L2 [20]	96.08
2	**SWin-L + ML-Decoder** [48]	**95.1**
3	**ViT-H/14** [18]	**94.55±0.04**
4	**ViT-B-16** [47]	**94.09**
5	**CvT-W24** [64]	**93.90±0.05**
6	**ViT-L16** [18]	**93.51**
7	BiT-L [32]	93.51

Taula 1: Millors resultats en ImageNet (dalt a l'esquerra), CIFAR-10 (dalt a la dreta) i CIFAR-100 (a baix) [15][33]

2.2 Xarxes Convolucionals (CNN)

Les xarxes convolucionals han estat dominant en molts problemes del camp del Deep Learning relacionats amb imatges, com pot ser el problema de classificació d'imatges, des del desenvolupament d'AlexNet [34] al 2012. Inspirades per processos biològics [21], les xarxes convolucionals són xarxes neuronals directes amb pesos compartits i interaccions disperses, és a dir, la majoria de pesos són 0 [7].

[2]Acc és una reducció d'Accuracy, precisió en català. És la métrica estàndard per avaluar la qualitat d'un model en la tasca de classificació d'imatges

Tal com hem comentat, no va ser fins a la publicació de la xarxa AlexNet que les xarxes convolucionals no van dominar completament. Aquesta xarxa revolucionària va demostrar, per primera vegada, que les característiques i valors de les xarxes obtinguts en el procés d'entrenament superaven als valors definits aribitràriament. Utilitzant tan sols 8 capes convolucionals, va imposar-se guanyant clarament a la resta de competidors en la ImageNet Large Scale Visual Recognition Challenge 2012 [50].

A partir de la victòria aclaparadora d'AlexNet, les CNN van evolucionar ràpidament, adaptant-se a cada vegada capacitats més grans de GPU, que els permetia tenir més paràmetres i millorar el seu rendiment. VGG-19, Inception V3 i fins a la més recent, EfficientNet-L2 [51][53][20], entre d'altres, han aconseguit mantenir les CNN en el seu domini incorporant nous paràmetres, escalant i uniformitzant les capes convolucionals, etc.

Fins i tot, cal destacar que a data de la realització d'aquest treball, el model que ha obtingut una major precisió en el conjunt d'imatges ImageNet ha estat el CoAtNet-7 [13], una xarxa neuronal que unifica els conceptes de convolució propi de les CNN amb el concepte d'atenció propi dels transformers. Aquest transformer intercala capes convolucionals amb capes d'atenció per acabar obtenint una precisió de 90.88% en ImageNet.

2.3 Transformers

Els transformers, unes xarxes neuronals introduïdes només fa 4 anys (l'any 2017 [56]), van revolucionar des del moment de la seva aparició en el camp del Deep Learning. Des de llavors, han anat sorgint diferents transformers els quals han anat optimitzant el seu funcionament i millorant el seu rendiment.

Aquesta estructura proposada per Vaswani implementava el concepte d'atenció [3], un concepte que s'estava començant a desenvolupar, però que fins aleshores únicament s'aplicava juntament amb xarxes neuronals recurrents. En canvi, Vaswani proposava una arquitectura basada en el concepte d'atenció, independitzant-se de les xarxes recurrents i aconseguint obtenir informació sobre dependències globals d'una manera paral·lelitzable.

2.3.1 Transformers de visió

En el camp de la visió artificial, les xarxes neuronals convolucionals eren els models dominants fins a l'aparició dels transformers. Per aquest motiu, en el moment que Vaswani [56] va publicar l'estructura dels transformers, poc temps més tard van sorgir models de CNN que intentaven afegir d'alguna forma el concepte d'atenció: afegint una capa d'autoatenció a nivell espacial [60] o substituint capes convolucionals per blocs atencionals [62]. Tanmateix, els transformers acabarien demostrant que es podrien independitzar de les CNN començant quan Dosovitskiy a finals de 2020 va publicar la primera versió d'un transformer aplicat a imatges[18]. Els autors allà van comprovar que el concepte d'atenció també es podia aplicar en imatges obtenint resultats competitius. Avui en dia, els transformers de visió s'estan aplicant a tots els camps de la visió artificial (tal i com es pot comprovar en la figura 1), on hi podem observar una classificació d'alguns dels transformers de visió utilitzats actualment, diferenciant-los segons la tasca per la qual han estat dissenyats i segons les seves característiques.

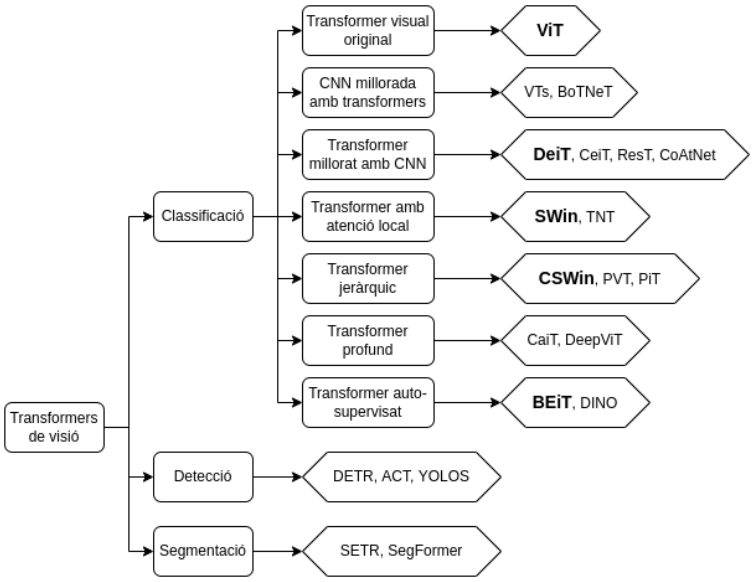

Figura 1: Classificació reduïda dels principals transformers de visió [39]

El problema de detecció d'objectes es basa en identificar objectes en una imatge, indicant la seva posició. DETR [8] va ser el primer transformer de visió dedicat a la detecció d'objectes. Amb una estructura encoder - decoder, rep com a input el resultat d'introduir la imatge en una CNN juntament amb la informació posicional. Tanmateix, rápidament van sorgir nous transformers, com el ACT [74], que intenta millorar la convergència del DETR reduint alhora el cost computacional a partir d'eliminar informació redundant en l'encoder. El YOLOS és un altre exemple de transformer de detecció d'objecte i en el seu cas es reestructura l'arquitectura del DETR per tal d'eliminar el decoder i, al mateix temps que millorar el DETR, generalitzar el seu ús a altres tasques visuals.

D'altra banda, en segmentació d'imatges, una tasca que consisteix en dividir una imatge en tots els múltiples objectes diferents que hi apareixen, també s'han aconseguit grans resultats. SETR [75] s'inspira amb el ViT i segueix la mateixa estructura en l'encoder. També utilitza diverses tècniques a nivell de píxel per acabar demostrant un gran rendiment en la tasca de segmentació d'imatges. Per altra banda, el SegFormer [65] fa ús d'un canvis en l'estructura del transformer amb l'objectiu de millorar el seu rendiment en la segmentació, com per exemple una estructura jeràrquica.

Per últim, en classificació d'imatges també tenim un seguit de transformers, els quals s'han classificat a partir de les seves característiques [39]. Dins d'aquest grup de transformers, destaquem el ViT [18], el primer transformer dedicat a la classificació d'imatges que va aconseguir obtenir uns resultats competitius. Les CNN millorades amb transformers [63][52], modifiquen algunes capes convolucionals per capes atencionals per tal d'intentar combinar els beneficis dels transformers en les CNN. Per altra banda, els transformers millorats amb CNN [54][68][73][13] ho enfoquen del punt de vista oposat: modificar un transformer per tal d'afegir-li una capa convolucional i combinar els beneficis de les CNN en el transformer. Un altre grup són els transformers amb atenció local [24][41], transformers que intenten enriquir la informació del voltant de cada patch que s'obté a través dels blocs atencionals. Continuant amb els grups, els transformers jeràrquics [17][59][26] modifiquen l'estructura del transformer ViT per tal que no totes les capes atencionals del

transformer necessitin la mateixa resolució d'imatge i d'aquesta manera aconsegueixen reduir el cost computacional al mateix temps que presten atenció també a característiques més petites de la imatge. El penúltim grup, els transformers profunds [55][77] augmenten la profunditat del transformer per tal que tingui en compte representacions més complexes [25]. Finalment els transformers auto-supervisats [4][9] canvien el seu mètode d'entrenament per un entrenament auto-supervisat, amb l'objectiu de reproduir la millora en els resultats obtinguts en els transformers auto-supervisats dedicats a tasques de processament de llenguatge natural.

En resum, tots aquests transformers estan superant les xarxes neuronals convolucionals (CNN), que eren les xarxes dominants en la visió artificial. Podem dir que fins aquest moment les CNNs van revolucionar així tots els camps de la visió artificial tal com podem observar en les taules 2 - 1.

Rank	Model	Box AP[3]
1	**Florence-CoSwin-H** [69]	**62**
2	**GLIP** [36]	**60.8**
3	**Soft Teacher+SWin-L** [66]	**60.7**
4	**DyHead** [12]	**60.3**
12	Cascade Eff-B7 NAS-FPN [22]	57.0

Taula 2: Millors resultats en detecció d'objectes en el conjunt de dades COCO [38]

Rank	Model	Validation mIoU[4]
1	**SWinV2-G** [40]	**59.9**
2	**SeMask-L MSFaPN-Mask2Former** [29]	**58.2**
3	**SeMask-L FaPN-Mask2Former** [29]	**58.0**
4	**SeMask-L Mask2Former** [29]	**57.5**
20	ResNeSt-200 [72]	48.36

Taula 3: Millors resultats en segmentació semàntica en el conjunt de dades ADE20K [76]

[3]Box AP és una métrica per mesurar la qualitat en problemes de detecció d'objectes
[4]mIoU és una métrica per mesurar la qualitat en problemes de segmentació d'imatges

3 Background teòric

En aquest apartat es comença definint què s'entén per xarxa neuronal i continua amb els conceptes relacionats directament amb els transformers, explicant el seu funcionament. Seguidament, es presenten els transformers de visió que s'utilitzaran en aquest treball i es finalitza l'apartat definint alguns conceptes matemàtics els quals s'utilitzaran per analitzar la convergència dels transformers i el seu procés d'aprenentatge.

3.1 Xarxes neuronals

Les xarxes neuronals, també conegudes com a xarxes neuronals artificials (ANN), són sistemes adaptatius artificials inspirats pel funcionament del cervell humà [42]. Igual que en el nostre cervell, una xarxa neuronal està formada per unitats bàsiques de processament de la informació, les neurones abstractes, les quals treballen conjuntament per tal d'aprendre i optimitzar una tasca.

Per poder definir concretament què entenem com a neurona necessitem definir uns conceptes previs:

Definició 3.1. *Anomenem **funció d'activació** d'una xarxa neuronal a la funció que, a partir de l'input d'una neurona, genera el seu output.*

Exemple 3.2. La funció ReLU [7] és una funció d'activació utilitzada en alguns dels blocs dels transformers. Es defineix de la següent forma:

$$\begin{aligned} ReLU : \mathbb{R} &\to \mathbb{R}^+ \\ x &\mapsto ReLU(x) = max(0, x) \end{aligned}$$

Observació 3.3. Una funció d'activació pot ser lineal o no lineal. Tanmateix, en xarxes neuronals és un requisit que sigui una funció d'activació no lineal per tal que la xarxa pugui aproximar qualsevol funció. [44]

Definició 3.4. *Anomenem **vector de pesos** al vector d'una neurona de mida igual al nombre d'inputs.*

Ara si, ja podem donar una definició formal d'una neurona abstracte.

Definició 3.5. *Una **neurona abstracte** està formada per quatre elements (x, w, φ, y) on $x^T = (x_0, ..., x_n) \in \mathbb{R}^n$ és el vector d'entrada o input, $w^T = (w_0, ..., w_n) \in \mathbb{R}^n$ és el vector de pesos, amb $x_0 = -1$ i $w_0 = b$, el biaix, i $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ és una funció d'activació que defineix la funció de sortida $y = \varphi(x^T w) = \varphi(\sum_{i=1}^n w_i x_i)$*

Les neurones s'agrupen en capes on cada capa rep una senyal d'entrada, la processa i genera una senyal de sortida. D'aquesta manera, una xarxa neuronal rep una senyal d'entrada, que anomenem input, la processa, és a dir, l'envia a la primera capa, la qual envia la seva sortida com a entrada per la següent capa i així successivament fins a l'última capa.

Definició 3.6. *Anomenem **xarxa neuronal** a un conjunt de capes formades per neurones la qual rep un conjunt de senyals d'entrada, input, i genera una senyal de sortida, output.*

Notació 1. *Anomenem paràmetres d'una xarxa neuronal al conjunt dels diferents vectors de pesos.*

Un concepte també recurrent en les xarxes neuronals és el concepte de logit:

Definició 3.7. *Anomenem **logits** a l'output no normalitzat d'una xarxa neuronal.*

3.1.1 La funció de Loss d'una xarxa neuronal

Inicialment una xarxa neuronal no té coneixement sobre com resoldre cap tasca. Per tant, una xarxa neuronal ha d'aprendre a realitzar de la millor manera possible la tasca que desitgem que ens resolgui. Per tal que la xarxa neuronal tingui una manera de mesurar la seva eficiència en la tasca en concret, li indiquem una funció de loss, la qual haurà d'intentar minimitzar.

Definició 3.8. *Una **funció de loss** (o pèrdua) és una funció objectiu que mesura la proximitat entre les prediccions d'una xarxa neuronal i el seu valor real.*

Observació 3.9. Les funcions de *loss també es poden anomenar funcions de cost o d'error.*

3.1.2 La funció de loss Cross-entropy

Una de les funcions de loss més populars és la funció de cross-entropy. Per poder-la definir, primer necessitem definir el concepte de funció negativa logarítmica de versemblança.

Definició 3.10. *Sigui q una densitat en $\mathbb{R}$. Definim la **funció negativa de versemblança** per $l_q(x) = \ln q(x)$.*

Observació 3.11. La funció negativa de versemblança compleix les següents propietats:

1. Negativa: $l_q(x) \leq 0$, $\forall x$.
2. $\forall q_1, q_2$ densitats independent, $l_{q_1 q_2} = l_{q_1}(x) + l_{q_2}(x)$
3. $\lim_{q(x)\to 0}(l_q(x)) = -\infty$

Definició 3.12. *Siguin p i q dues densitats en $\mathbb{R}$. Definim la funció de **cross-entropy** de p respecte q com $S(p,q) = \mathbb{E}^p[-l_q] = -\int_{\mathbb{R}} p(x)\ln q(x)dx$ [7].*

3.1.3 Xarxes neuronals supervisades i auto-supervisades

Les xarxes neuronals són molt variades i poden tenir moltes categories diferents, tanmateix, les podem englobar en dos tipus diferents: xarxes supervisades i auto-supervisades o no supervisades. Aquests dos tipus es diferencien en les seves etapes d'entrenament.

Definició 3.13. *Diem que una xarxa neuronal està en **fase d'entrenament** quan la xarxa està aprenent a realitzar la tasca per la qual ha estat dissenyada.*

Definició 3.14. *Diem que una xarxa neuronal és **supervisada** quan durant les seves etapes d'entrenament es coneix el resultat de la seva funció objectiu.*

Definició 3.15. *Diem que una xarxa neuronal és **auto-supervisada** o **no supervisada** quan no es coneix el resultat de la seva funció objectiu.*

3.1.4 Xarxes neuronals directes (FNN)

Dins de la multitud de tipus de xarxes neuronals, les xarxes neuronals directes han aconseguit fer-se un lloc en el camp del Deep Learning degut a la seva simplicitat, ja que la informació només circula en una direcció.

Definició 3.16. *Sigui* $U_l = \{1, 2, ..., d^{(l)}\}$, $0 \leq l \leq L$, *i considerem* $\phi^{(l)} : \mathbb{R} \to \mathbb{R}$ *la seqüència de funcions d'activació i* $\alpha_1, ..., \alpha_L$ *una seqüència de funcions afins,* $\alpha_l : \mathcal{F}(U_{l-1}) \to \mathcal{F}(U_l)$, *on* $\mathcal{F}(U) = \{f : U \to \mathbb{R}\}$ *és el conjunt de funcions reals definides en el conjunt* U. *Aleshores la* ***xarxa neuronal directa o Feedforward Neural Network*** *(FNN) és la seqüència d'aplicacions* $f_0, ...f_L$ *tals que*

$$f_l = \phi^{(l)} \circ \alpha_l \circ f_{l-1},\ 1 \leq l \leq L$$

amb f_0 *donada. [7]*

3.1.5 Multi-layer Perceptrons

Una xarxa neuronal *perceptró multicapa* (MLP) és un model de xarxa neuronal directa supervisada.

Definició 3.17. *Anomenem* ***perceptró*** *a una neurona en la qual el seu input només pot ser 0 o 1 i té com a funció d'activació la funció Heaviside:*

$$\varphi(x) = \begin{cases} 0,\ si\ x < 0 \\ 1,\ si\ x \geq 0 \end{cases}$$

Observació 3.18. Seguint la notació descrita en la definició de neurona, l'output d'un perceptró és:

$$y = \varphi(x^T w - b) = \begin{cases} 0, si \sum_{i=1}^{n} w_i x_i < b \\ 1, si \sum_{i=1}^{n} w_i x_i \geq b \end{cases}$$

Definició 3.19. *Una xarxa MLP és una FNN supervisada formada per perceptrons.*

3.2 Optimitzadors de l'aprenentatge de les xarxes neuronals

Com ja hem mencionat anteriorment, inicialment una xarxa neuronal no sap com resoldre un problema. És a mesura que entrenem la xarxa que aquesta aprèn com resoldre'l. També sabem que aquest procés d'aprenentatge consisteix en minimitzar la funció de loss ja que aquesta permet a la xarxa qualificar la seva resolució del problema. En aquest procés de minimització de la funció de loss és on participa un optimitzador.

Definició 3.20. *Anomenem* ***optimitzador*** *a aquell algoritme que ens modifica els paràmetres de la xarxa neuronal per tal que la funció de loss es minimitzi.*

En la majoria d'optimitzadors es fa ús del gradient de la funció de loss per tal de minimitzar-la ja que ens permet obtenir informació sobre com evoluciona la funció.

Definició 3.21. *El* ***gradient*** *d'una funció* $f : \mathbb{R}^n \to \mathbb{R}^n$ *és un vector format per les seves derivades parcials:*

$$\nabla f(x_1, ..., x_n) = (\frac{\partial f}{\partial x_1}(x_1, ..., x_n), ..., \frac{\partial f}{\partial x_n}(x_1, ..., x_n))^T$$

Exemple 3.22. Un dels algoritmes més famosos i coneguts és el **descens del gradient**, un mètode iteratiu que busca el mínim de la funció de loss en funció del seu gradient amb la fórmula:

$$x^{k+1} = x^k - \eta \cdot (\frac{\nabla L(x^k)}{||\nabla L(x^k)||})$$

on k és la iteració, η un paràmetre que serveix per regular quant ens desplacem en cada iteració cap al mínim, L és la funció de loss i $x = (x_1, ..., x_n)$ és el conjunt de paràmetres dels quals s'ha de calcular el gradient.

Un algorisme derivat del descens del gradient és el **descens del gradient estocàstic** (SGD), molt més eficient i que consisteix en realitzar el descens del gradient en cada subconjunt de dades que iterem, en comptes de fer-ho sobre tot el conjunt de dades en cada època d'entrenament.

3.2.1 El mètode d'optimització Adam

Adam és un optimitzador presentat al 2014 per Diederik i Ba. [31] amb l'objectiu de combinar els avantatges dels altres dos optimitzadors més populars en en el moment de la publicació de l'article: AdaGrad [19] i RMSProp [27].

Per entendre com funciona l'optimitzador Adam, primer hem de definir alguns conceptes.

Definició 3.23. *Sigui X una variable aleatòria contínua amb densitat $p(x)$ en un espai $\mathcal{X}$, aleshores per cada funció vectorial $\phi : \mathcal{X} \to \mathbb{R}^n$ definim el $\phi-$moment de X com:*

$$\mu_\phi(X) = \mathbb{E}[\phi(X)] = \int_{\mathcal{X}} \phi(x)p(x)dx$$

Observació 3.24.

1. Si $\phi(x) = x$, aleshores el ϕ-moment és la mitjana. Aquest moment l'anomenem primer moment.
2. Si $\phi(x) = x^2$, aleshores el ϕ-moment és la variància no centrada. Aquest moment l'anomenem segon moment.

Notació 2. *Denotarem per m^k i v^k els primers i segons moments respectivament en la iteració k d'un conjunt de variables aleatòries.*

L'algorisme Adam comença considerant que en la iteració $k = 0$, els dos primers moments són vectors nuls i els actualitza amb la fórmula:

$$m^{k+1} = \beta_1 m^k + (1 - \beta_1)\nabla L(x^{k+1})$$
$$v^{k+1} = \beta_2 v^k + (1 - \beta_2)(\nabla L(x^{k+1}))^2$$

on β_1,$\beta_2 \in [0, 1)$ són dos constants i $(\nabla L(x^{k+1}))^2$ és el quadrat de cada element de la funció de loss.

Finalment, la fórmula que implementa l'Adam és:

$$x^{k+1} = x^k - \eta \frac{\hat{m}^k}{\sqrt{|\hat{v}^k|}+\epsilon}$$

on $\epsilon > 0$ és un nombre molt petit que serveix per evitar la divisió per 0, η és un paràmetre que serveix per regular quant ens desplacem en cada iteració cap al mínim i $\hat{m}^k$ i $\hat{v}^k$ han estat calculats de la següent manera:

$$\hat{m}^k = \frac{m^k}{1-\beta_1^k};\ \hat{v}^k = \frac{v^k}{1-\beta_2^k}$$

Fins ara hem definit i introduït conceptes generals en les xarxes neuronals. A continuació es defineixen els principals conceptes relacionats amb els transformers.

3.3 Atenció

El concepte d'Atenció ha estat la gran revolució dels transformers i és el principal motiu rere el gran èxit dels transformers. Per introduir la funcionalitat de l'atenció d'una manera més informal, podríem considerar que li transmet al transformer la informació sobre a quines dades ha de prestar més atenció.

Definició 3.25. *Sigui $x \in \mathbb{R}^n$ un vector. Definim la funció n-dimensional com $softmax(x) = z$, amb $z_j = \frac{e^{x_j}}{||e^x||}$, $j = 1, ..., n$. Normalment, es considera la norma L^1, és a dir, $||e^x|| = \sum_{i=1}^{n} e^{x_i}$.*

Definició 3.26. *Siguin $X \in \mathbb{R}^{n_x \times d_y}$ i $Y \in \mathbb{R}^{n_y \times d_y}$. Aleshores definim la matriu* query*, matriu* key *i matriu* value *com les matrius resultants de les operacions:*

$$Q = XW^Q,\ K = YW^K,\ V = YW^V$$

on $W^Q \in \mathbb{R}^{d_x \times d^k}$, $W^K \in \mathbb{R}^{d_y \times d^k}$, $W^V \in \mathbb{R}^{d_y \times d^v}$ són matrius lineals i amb $d^k, d^v \in \mathbb{R}$ [39].

Definició 3.27. *Sigui $Q \in \mathbb{R}^{d^k}$, $K \in \mathbb{R}^{d^k}$ i $V \in \mathbb{R}^{d^v}$ les matrius query, key i value respectivament. Siguin $K = \{k_1, ..., k_n\} \in K$ i $V = \{v_1, ..., v_n\} \in V$ un conjunt de keys i de values respectivament, i $q \in Q$ una query. Sigui $a : Q \times V \to R$ una funció de semblança. Definim l'atenció com a l'aplicació Atenció$(q, K, V) := \sum_{i=1}^{N} softmax_i(\{a(q, k_i)\}i)v_i$ [57] [3].*

Notació 3. *Com tots els vectors q els processem paral·lelament, unificarem la definició d'atenció per poder passar tots els vectors q com a files de l'espai query:*

$$Atenció(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

Definició 3.28. *Definim l'auto-atenció com l'atenció quan els inputs X, Y per obtenir les matrius query, key i value són iguals, és a dir, quan $X = Y$.*

Observació 3.29. Un bloc d'atenció té tres matrius de pesos W^Q, W^K, W^V, rep com a input els dos tensors X i Y i retorna com a output el resultat d'aplicar l'aplicació Atenció en les matrius query, key i value obtingudes a partir de l'input.

3.4 Multi-head Self Attention

Després de realitzar certes proves, Vaswani [56] es va adonar que un bloc era massa restrictiu i que necessitava una manera d'ajuntar múltiples blocs atencionals independents entre ells per poder obtenir resultats competitius. D'aquesta idea van sorgir els Multi-head Self-Attention blocks (MHSA).

Definició 3.30. *Siguin* $\forall i \in \{1, ..., h\}$, Q_i, K_i, V_i *les matrius query, key, value de cada bloc atencional obtinguts a partir de les respectives matrius lineals* $W^{Q_i} \in \mathbb{R}^{d_x^k}$, $W^{K_i} \in \mathbb{R}^{d_x^k}$, $W^{V_i} \in \mathbb{R}^{d_x^v}$, *aleshores:*

$$MHSA(Q, K, V) = Concat(Z_1, ...Z_h)W0$$

on Q, K, V *són les matrius resultant de concatenar les matrius* Q_i, K_i, V_i, $W0 \in \mathbb{R}^{hd_v \times d_x}$ *és la matriu de projecció i* $Z_i = Attention(Q_i, K_i, V_i)$, $\forall i \in \{1, ..., h\}$.

Un cop explicat l'atenció i el bloc atencional, introduirem un concepte que incorpora únicament el transformer Deit, la destil·lació de coneixement.

3.5 Destil·lació de coneixement

Destil·lació de coneixement és un tipus d'entrenament de xarxes neuronals publicat per Hinton G. [28] en el qual s'aprofita una xarxa neuronal ja entrenada, que anomenem mestra, per tal de guiar i millorar l'eficiència de l'entrenament en la xarxa neuronal que entrenem, que anomenem alumna. Un cop tenim la mestra i l'alumna, definim una funció objectiu la qual buscarem minimitzar, igual que amb la funció de loss.

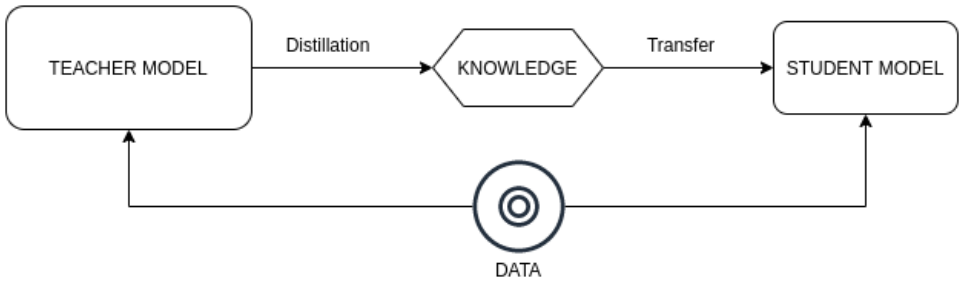

Figura 2: Diagrama de la relació entre la xarxa mestra i l'alumna [23]

Hi ha dos tipus principals de destil·lació: suau i forta de classe.

3.5.1 Destil·lació suau

Aquesta destil·lació busca minimitzar la divergència de Kullback-Leibler [35] entre les funcions softmax de la mestra i de l'alumna, ja que aquesta és una bona indicadora de la similitud entre les dues funcions.

La funció objectiu en aquesta destill·lació és:

$$\mathcal{L}_{suau} = (1 - \lambda)\mathcal{L}_{CE}(\psi(Z_S), y) + \lambda\tau^2 KL(\psi(Z_S/\tau), \psi(Z_t/\tau)) \tag{3.2}$$

on Z_t són els logits de la mestra, Z_S els logits de l'alumna, τ un paràmetre que anomenem temperatura per la destil·lació, λ el coeficient que normalitza la divergència de Kullback-Leibler, que anomenem KL, $\mathcal{L}_{CE}$ la funció de cross-entropy sobre les classes de les imatges y i ψ la funció softmax. [54]

3.5.2 Destil·lació forta de classe

Aquesta destil·lació, introduïda per Touvron H. [54] equipara la classe donada pel conjunt de dades amb la classe obtinguda amb la mestra.

La funció objectiu en aquesta destil·lació és:

$$\mathcal{L}_{hard} = 0.5\mathcal{L}_{CE}(\psi(Z_S), y) + 0.5\mathcal{L}_{CE}(\psi(Z_S), y_t)$$

on Z_t són els logits de la mestra, Z_S els logits de l'alumna, $\mathcal{L}_{CE}$ la funció de cross-entropy sobre les classes de les imatges y i ψ la funció softmax [54].

3.6 Positional Encoding

Un dels punts forts dels transformers és que totes les parts en les que se separa l'input (que anomenem patches) es processen paral·lelament i de la mateixa forma. Així doncs, sembla obvi que d'alguna manera li hem de fer saber al transformer quina posició té cada patch dins de l'input original.

En els primers transformers, s'afegia un vector posició a cada patch de tal manera que, les matrius de pesos aprenien a tenir en compte la posició, en cas que fos necessari [56]. Aquesta manera d'afegir un vector posició fixat s'anomena **positional encoding**.

Exemple 3.31. Les funcions sinus i cosinus solen ser usades com a valors pels vectors de posició, de la següent manera:

$$PE_{(pos,i)} = \begin{cases} sin(pos \cdot w_k) & \text{if } i = 2k \\ cos(pos \cdot w_k) & \text{if } i = 2k+1 \end{cases}$$

amb $w_k = \frac{1}{10000^{2k/d}}$, $k = 1, ..., d/2$ i on pos és la posició de l'element i i és la posició del valor dins del vector de positional encoding.

Tanmateix, alguns transformers, en lloc d'aplicar un vector fixat per passar la informació sobre la posició, passen un vector el qual també és entrenat. Aquesta manera d'afegir un vector posició que també s'entrena s'anomena "positional embedding".

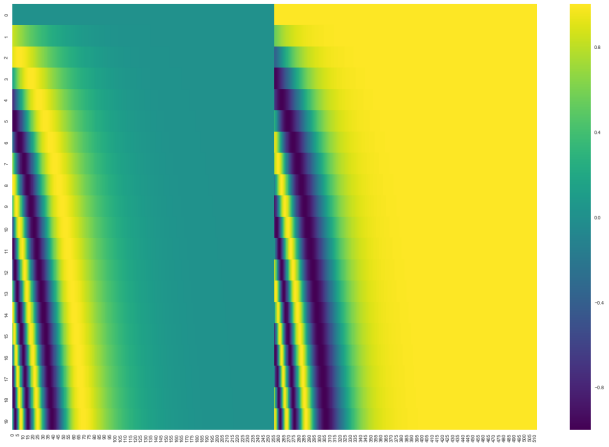

Figura 3: Visualització dels valors del positional encoding en un input de 20 elements i amb un positional encoding de dimensió 512. La partició que es veu a la meitat del gràfic és deguda a que en la primera meitat, els valors són generats per la funció sinus i en la segona meitat per la funció cosinus. [1]

3.7 Transformers

Arribats en aquest punt, ja tenim tots els conceptes necessaris per a poder definir els transformers i explicar com Vaswani, en l'article *Attention is all you need* [56] va presentar una arquitectura que en poc temps ha eclipsat tot el camp del processament de llenguatge natural i, també a ritmes vertiginosos, va eclipsant tots els altres camps del Deep Learning.

Si ens mirem l'arquitectura dels transformers a gran nivell, veurem que la podríem resumir com una composició de dos blocs principals, l'**encoder** i el **decoder**, els quals estan formats per blocs MHSA i FNN, juntament amb algunes capes de normalització i unes capes de codificació posicional.

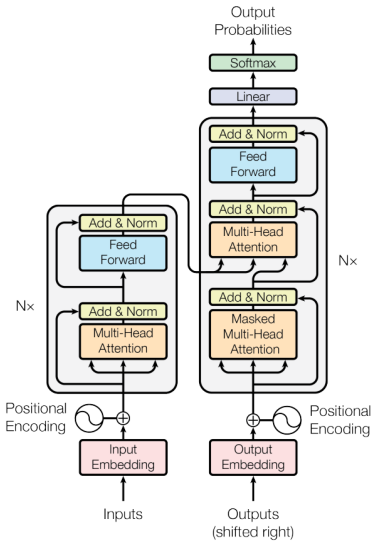

Figura 4: Estructura d'un transformer [56]

Per explicar els transformers a un nivell més baix, seguirem el procés des que entra l'input fins que obtenim l'output. Com que el transformer original va ser pensat per processament de llenguatge natural, suposarem que l'input és una frase, per exemple, "El transformer és una arquitectura revolucionària".

Primer de tot, separem la frase per paraules i, per tant, ens queda un vector amb 6 paraules o tokens. A continuació li afegim la codificació posicional triada, que en el transformer original són les funcions sinus i cosinus explicades en l'exemple [3.31].

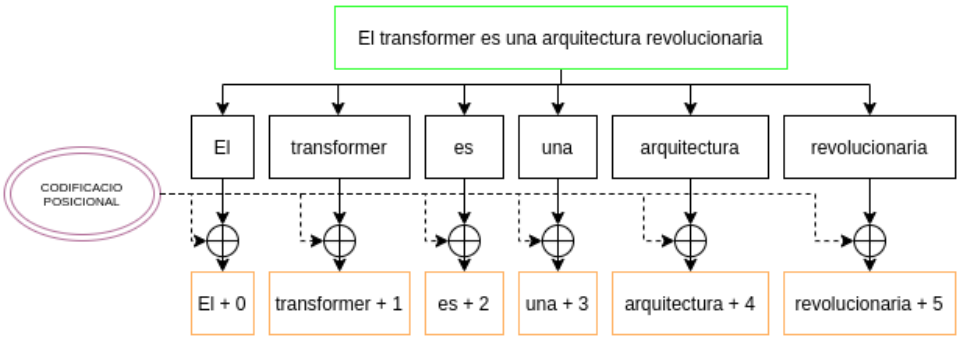

Figura 5: Passos previs a l'encoder. La codificació posicional l'hem simplificat amb nombres enters representant l'ordre de les paraules.

En aquest punt, ja tenim l'entrada per l'encoder.

3.7.1 Encoder

En el transformer original, l'encoder estava format per sis capes iguals figura, cada una de les quals formada per un bloc MHSA i una FNN. Així doncs, l'input que rep l'encoder es passa en paral·lel al bloc MHSA i obtenim, per cada token x_i^1, un vector z_i^1 amb la informació sobre l'atenció.

En aquest punt, abans de la capa FNN, trobem una sub-capa de normalització en la qual sumem l'output del bloc MHSA amb l'input de l'encoder, és a dir, sumem els vector x_i^1 amb els vectors z_i^1 per obtenir un vector, diguem-li x_i^2, el qual ja sí és l'input per la FNN.

Un cop obtingut l'output de la FNN, que l'anomenem z_i^2, aquest entra en una altra sub-capa de normalització en la qual sumem z_i^2 amb x_i^2 per obtenir x_i^3 i normalitzar-lo. Aquest x_i^3 és l'output de la primera capa de l'encoder i podem observar com té les mateixes dimensions que l'input que rep aquesta capa.

Així doncs, aquest procediment es reprodueix per cada capa, sent l'output d'una capa, l'input de la posterior fins que arriba al final de l'encoder.

3.7.2 Decoder

Igual que en l'encoder, en el transformer original el decoder estava format per sis capes iguals, amb la diferència que cada capa està formada per les dues mateixes sub-capes dels

encoders més una sub-capa de MHSA que rep com a input, en comptes dels outputs de l'encoder, els outputs anteriors del transformer.

Tot i així, les capes similars a l'encoder sí que reben com a input part de l'output de l'encoder, concretament les matrius K i V que ajudaran al decoder a prestar atenció als tokens corresponents de l'input.

3.8 Transformers de visió

Vista la gran revolució dels transformers en el camps de processament de llenguatge natural, els transformers aplicats en els camps de la visió artificial no van fer-se esperar. Aquests transformers són els que anomenem transformers de visió. En aquest apartat, introduirem cinc d'aquests transformers de visió, els quals estan dedicats al problema de la classificació d'imatges.

3.8.1 Visual Transformer (ViT)

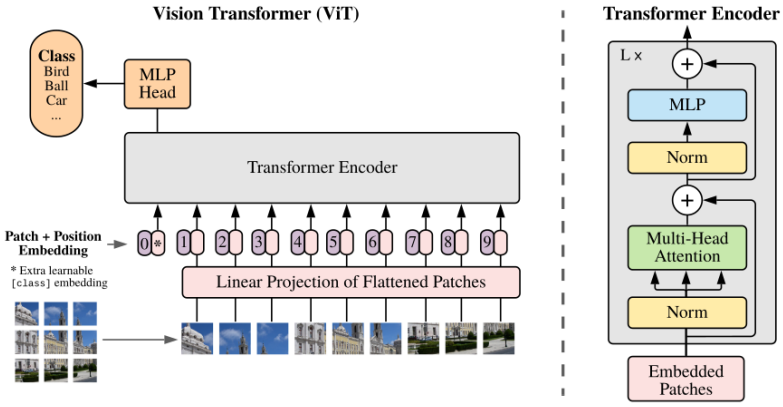

Figura 6: Estructura del ViT

Presentat a l'octubre del 2020, el Visual Transformer (ViT) [18] va ser el primer transformer de visió dedicat a la classificació d'imatges. Buscant que l'estructura fos el més semblant possible a l'estructura del transformer original publicat per Vaswani [56], simplement parteix la imatge d'entrada en trossos que no se solapen i aleshores fan una projecció a cada tros per transformar-lo en un vector. D'aquesta manera, se simula que cada vector és un dels tokens en el transformer original. Per tal de realitzar la classificació, ViT afegeix un token de classe i el tracta com si fos un tros de la imatge. Aleshores, aquest token serà el que s'utilitzarà a l'hora de fer la classificació.

Aquest transformer aprèn de manera supervisada i necessita de grans quantitats d'imatges per poder ser entrenat correctament. És aquí on apareix un dels problemes principals del ViT: necessita d'una gran quantitat d'imatges classificades per tal de poder ser competitiu. Un dels altres inconvenients del ViT és que, al partir la imatge en trossos, és poc sensible als detalls petits de la imatge.

Envers la resta de transformers de visió, el ViT destaca per ser el més simple d'es-

tructura sense que això li comporti pèrdua d'eficiència, tal i com podem observar en els resultats que ha aconseguit. Concretament, aquest transformer ha aconseguit resultats competitius com el model amb més precisió en el conjunt de dades CIFAR-10 [33] amb una precisió de 99.50±0.06% o el tercer model amb més precisió en el conjunt de dades CIFAR-100 [33]. Tanmateix, necessita de grans conjunts de dades d'entrenament per poder obtenir aquests resultats competitius.

El fet que tingui una estructura tant simple també ha afavorit que s'hagi pres de base a molts dels transformers de visió posteriors, els quals també han obtingut resultats excepcionals, com poden ser els altres transformers d'aquest treball: DeiT, BEiT, SWin i CSWin.

3.8.2 Data-efficient image transformer (DeiT)

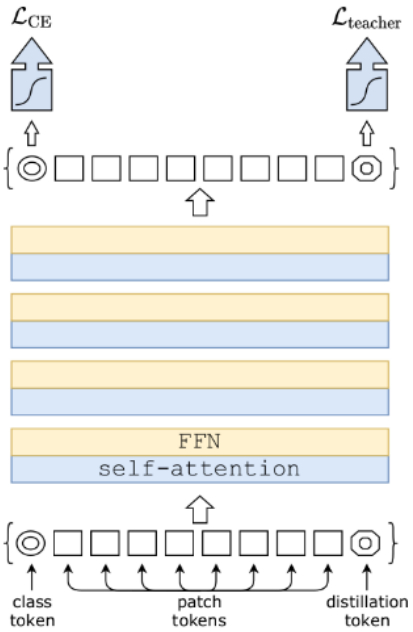

Figura 7: Estructura del DeiT

Presentat al desembre del 2020, el DeiT [54] intenta solucionar el problema del ViT amb la necessitat de grans conjunts de dades d'entrenament. Per aconseguir-ho, aplica un procés de destil·lació forta de classe amb un classificador d'imatges precís que fa de mestre, afegint un token de desil·lació. L'objectiu d'aquest token és similar al token de classe present també en el ViT, és a dir, indicar el màxim possible la classe correcta. Tanmateix, aquest pren per imatge correcta la classe indicada per la xarxa mestra.

Contrari al que es pugui pensar en un primer moment, el DeiT obté uns millors resultats que la xarxa mestra, com és el cas de la RegNetY-16GF, la qual obté una precisió de 82.9% en el conjunt ImageNet, envers el 85.2% de precisió del DeiT.

A més, aquest transformer ha aconseguit resultats competitius com el sisè model amb més precisió en el conjunt de dades CIFAR-10 amb una precisió de 99.1% o una precisió de 90.8% en el conjunt CIFAR-100.

3.8.3 Bidirectional Encoder representation from image Transformers (BEiT)

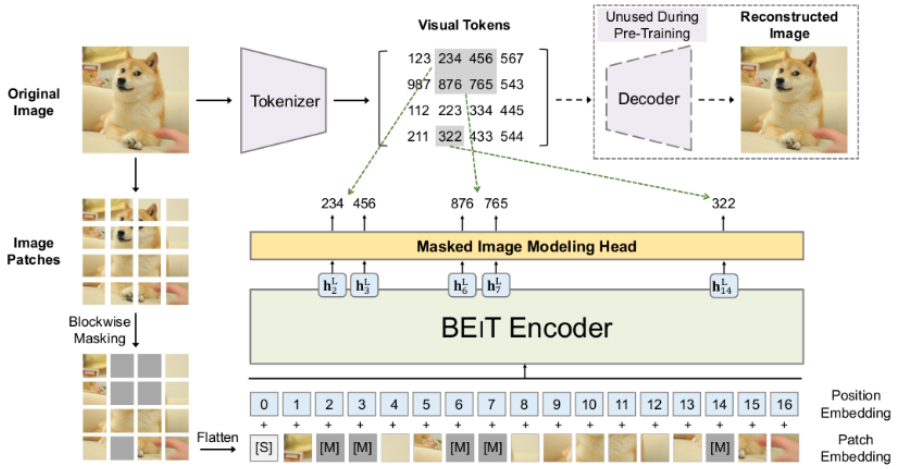

Figura 8: Estructura del BEiT

Presentat al juny del 2021, el BEiT [4] emula el transformer BERT que està dedicat a text [16]. En aquest transformer, en l'entrenament, cada imatge és representada de dues maneres diferents: en un vector de tokens visuals i en trossos similar en el ViT. Respecte la primera representació, destacar que de cada imatge s'obté una matriu de 14x14 tokens visuals, cada un, representant la part corresponent de la imatge. Per obtenir aquests tokens, s'utilitza un vocabulari de 8192 paraules i el tokenitzador d'imatges publicat per Ramesh A. [46].

Aleshores, durant l'entrenament, s'emmascaren alguns dels trossos de la imatge, és a dir, es canvien els valors reals d'aquells trossos per un valor constant, i l'objectiu del transformer és obtenir tots els tokens visuals de l'altra representació, inclosos els emmascarats.

Aquest transformer ha aconseguit resultats competitius tant en segmentació semàntica, amb una 5a posició en el conjunt d'imatges ADE20k, com en classificació d'imatges, amb una 14a posició en el conjunt d'imatges ImageNet o una 7a posició si prenem el top5 de precisió.

3.8.4 Shifted Windows Transformer (Swin)

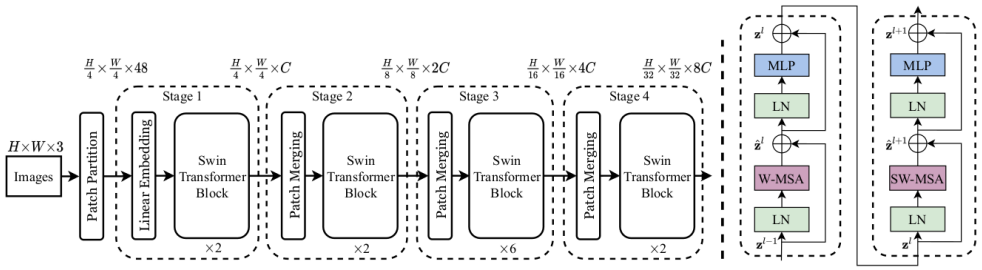

Figura 9: Estructura del SWin

Presentat al març del 2021, el Shifted Windows Transformer [41] implementa un sistema jeràrquic i un sistema de finestra desplaçable similar a un mòdul de desplaçament temporal (TSM) [37]. El transformer comença amb unes finestres de mida reduïda que permeten al transformer tenir en compte detalls més petits, per anar augmentant la mida de les finestres per tenir una visió més global de la imatge. D'aquesta manera, comparant amb el ViT, el SWin transformer té en consideració detalls més petits de la imatge i, alhora, és més adaptable i òptim per diferents mides d'imatge ja que el cost computacional a l'augmentar la resolució de les imatges passa a ser de quadràtic en el ViT a lineal en el Swin.

El Swin Transformer ha obtingut molt bons resultats tant en segmentació semàntica com en detecció d'objectes. En classificació d'imatges, el SWin justament quan el van introduir va obtenir una vint-i-vuitena posició en el rànquing de models amb més precisió en el conjunt de dades ImageNet, amb una precisió de 87.3%. Tanmateix, versions més recents d'aquest transformer han arribat a obtenir la sisena posició en el mateix rànquing amb una precisió de 90.17% [40].

3.8.5 Cross-Shaped Windows Transformer (CSWin)

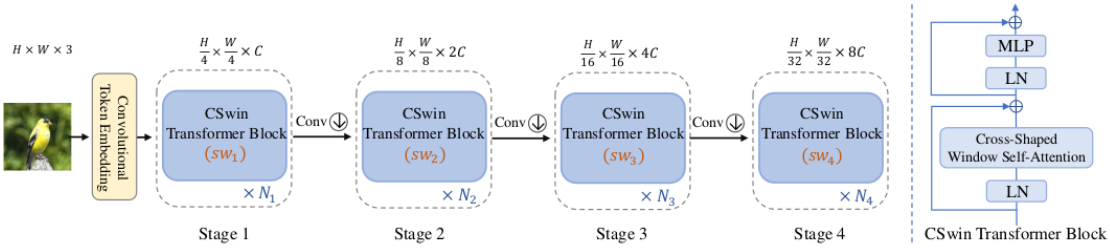

Figura 10: Estructura del CSWin

Presentat al juliol del 2021, el CSWin transformer[17] és un transformer que no està enfocat únicament en la classificació d'imatges. De manera similar al SWin transformer, implementa un sistema jeràrquic i un sistema similar al TSM [37] però que, en comptes de tenir finestres de mida quadrada, divideixen la imatge en tires horitzontals i verticals.

A més, també implementa un nou sistema per codificar la posició anomenat Codificació Posicional millorada Localment (LePE).

La LePE es diferencia de la resta de codificacions posicionals que han aparegut en la resta de transformers perquè incorporen la informació posicional dins de cada bloc del transformer i no només abans d'entrar en tots els blocs. Alhora, Dong fa la hipòtesi que, per cada input del transformer, la informació posicional és la dels entorns propers a l'element que s'esta analitzant. D'aquesta manera, aconsegueixen reduir el cost computacional i la fórmula queda:

$$Attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d}}V + DWConv(V)$$

on DWConv fa referència a un operador de convolució en profunditat [11].

Aquest transformer, al ser més recent, ha estat provat en pocs conjunts de dades i amb poques versions. Tot i així, ha obtingut una precisió de 87.5% en el conjunt de dades ImageNet, fet que col·loca aquest transformer en la vint-i-sisena posició del rànquing de models en aquest conjunt de dades.

Un cop definits els conceptes relacionats amb els transformers i haver-los introduït, es defineixen alguns conceptes matemàtics que s'utilitzaran a l'hora de comparar i analitzar els resultats.

3.9 Normes matricials

Definició 3.32. *Sigui E un espai vectorial sobre $\mathbb{R}$. Anomenem* ***norma*** *a E a l'aplicació:*

$$\begin{aligned} ||\cdot|| : E &\to \mathbb{R}^+ \\ x &\mapsto ||x|| \end{aligned}$$

tal que compleix les següents propietats:

1. $||x|| = 0 \iff x = 0$
2. $||cx|| = |c|||x||$, $\forall x \in E$ *i* $\forall c \in \mathbb{R}$
3. $||x + y|| \leq ||x|| + ||y||$, $\forall x, y \in E$

Definició 3.33. *Diem que dues normes $||\cdot||_a$ i $||\cdot||_b$ són* ***equivalents*** *si existeixen constants $k_2 \geq k_1 > 0$ tals que:*

$$k_1||x||_a \leq ||x||_b \leq k_2||x||_a, \forall x \in \mathbb{R}^n$$

Proposició 3.34. *La propietat d'equivalència entre normes és transitiva.*

Demostració: Suposem que tenim dues normes $||\cdot||_a$ i $||\cdot||_b$ equivalents a una tercera norma $||\cdot||$ per constants $0 < k_1^a \leq k_2^a$ i $0 < k_1^b \leq k_2^b$, respectivament. Aleshores:

$$\begin{aligned} k_1^a||x|| \leq ||x||_a \leq k_2^a||x|| \\ k_1^b||x|| \leq ||x||_b \leq k_2^b||x|| \end{aligned}$$

És fàcil veure que $\frac{k_1^b}{k_2^a}||x||_a \leq ||x||_b \leq \frac{k_2^b}{k_1^a}||x||_a$. □

Teorema 3.35. *Totes les normes a $\mathbb{R}^n$ són equivalents.*

Demostració [30]: Es defineix la norma-1 com $||x||_1 = \sum_{i=1}^{n} |\alpha_i|$. És fàcil comprovar com compleix totes les condicions de norma.

Primer de tot, demostrarem que ens és suficient amb considerar només x tal que $||x||_1 = 1$. És a dir, provarem que $k_1||x||_1 \leq ||x||_a \leq k_2||x||_1$ és cert per tot $x \in \mathbb{R}^n$ per algun k_1 i k_2. El cas per $x = 0$ és trivial. Consideren $x \neq 0$. Dividint per $||x||_1$ obtenim la condició:

$$k_1 \leq ||u||_a \leq k_2, \ (1)$$

on $u = \frac{x}{||x||_1}$ té norma $||u||_1 = 1$.

A continuació, provarem que tota norma $|| \cdot ||_a$ és una funció contínua en $\mathbb{R}^n$ sota la tipologia induïda per la norma $|| \cdot ||_1$. És a dir, provarem que $\forall \epsilon > 0$, $\exists \delta > 0$ tal que

$$||x - x'||_1 < \delta \implies | \ ||x||_a - ||x'||_a \ | < \epsilon$$

Per la propietat de desigualtat triangular de les normes, tenim que:

$$||x||_a - ||x'||_a = ||x' + (x - x')||_a - ||x'||_a \leq ||x - x'||_a$$
$$||x'||_a - ||x||_a = ||x' - (x - x')||_a - ||x||_a \leq ||x - x'||_a$$

i per tant,

$$| \ ||x||_a - ||x'||_a \ | \leq ||x - x'||_a$$

Ara, prenem $(e_1, ..., e_n)$ la nostra base de $\mathbb{R}^n$ i escrivim $x = \sum_{i=1}^{n} \alpha_i e_i$ i $x' = \sum_{i=1}^{n} \alpha'_i e_i$. Per la desigualtat triangular tenim:

$$||x - x'||_a \leq \sum_{i=1}^{n} |\alpha_i - \alpha'_i| \cdot ||e_i||_a \leq ||x - x'||_1 \max_i ||e_i||_a$$

Finalment, triant $\delta = \frac{\epsilon}{\max_i ||e_i||_a}$, obtenim que:

$$||x - x'||_1 < \delta \implies | \ ||x||_a - ||x'||_a \ | \leq ||x - x'||_a < \epsilon$$

Ja per acabar, apliquem el teorema dels valors extrems (una funció contínua en un conjunt compacte té un màxim i un mínim en el conjunt). Sigui

$$k_1 = \min_{||u||_1 = 1} ||u||_a$$
$$k_2 = \max_{||u||_1 = 1} ||u||_a$$

Com que $u \neq 0$ per $||u||_1 = 1$, obtenim que $k_2 \geq k_1 > 0$ i que $k_1 \leq ||u||_a \leq k_2$ que és el que necessitàvem a (1). □

Definició 3.36. *Una **norma matricial** és una norma en l'espai vectorial $\mathbb{R}^{n \times n}$ que, a més de les propietats de norma, és sub-multiplicativa, és a dir:*

$$||AB|| \leq ||A||||B||$$

Exemples 3.37. Sigui $A = (a_{ij})_{1\leq i\leq n, 1\leq j\leq m} \in \mathbb{R}^{n\times m}$. Alguns exemples de normes matricials són:

1. Norma sub-1: $||A||_1 = \max\limits_{1\leq j\leq m}(\sum\limits_{i=1}^{n}|a_{ij}|)$
2. Norma euclidiana: $||A||_2 = (\rho(A^TA))^{1/2}$, on ρ indica el radi espectral, que és el màxim dels mòduls dels valors propis de la matriu.
3. Norma del màxim: $||A||_\infty = \max\limits_{1\leq i\leq n}(\sum\limits_{j=1}^{m}|a_{ij}|)$
4. Norma de Frobenius: $||A||_F = \sqrt{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}||a_{ij}|^2} = \sqrt{\sum\limits_{i=1}^{min(m,n)}\sigma_i^2}$, on σ_i són els valors singulars de la matriu.

3.10 Descomposició en valors singulars (SVD)

La descomposició en valors singulars d'una matriu A és la factorització $A = U\Sigma V^T$, on U i V són matrius ortonormals i Σ és una matriu diagonal la qual té per elements els valors singulars d'A.

Definició 3.38. *Sigui A una matriu $\mathbb{R}^{m\times n}$, és a dir, de m files i n columnes, λ és un **valor propi** d'A si i només si existeix un vector v no nul tal que $Av = \lambda v$.*

Observació 3.39. Sigui A una matriu $\mathbb{R}^{m\times n}$, aleshores $\lambda \geq 0$ $\forall$ valor propi λ d'A^TA.

Definició 3.40. *Sigui A una matriu $\mathbb{R}^{m\times n}$, sigui $k = min(m,n)$ i siguin $\lambda_1, ..., \lambda_k$ els valors propis de la matriu A^TA ordenats de tal manera que $\lambda_1 \geq ... \geq \lambda_k \geq 0$. Aleshores els **valors singulars** de la matriu A són els valors $\sigma_1 \geq ... \geq \sigma_k \geq 0$ tal que $\sigma_i = \sqrt{\lambda_i}$, $1 \leq i \leq k$.*

Teorema 3.41. *Tota matriu $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ es pot factoritzar com:*

$$A = U\Sigma V^T$$

on U és una matriu amb columnes ortogonals de dimensió $m \times n$, V és una matriu amb columnes ortogonals de dimensió $n \times n$ i Σ és una matriu diagonal $n \times n$ amb els valors singulars d'A a la diagonal.

Demostració [14]: Utilitzarem inducció sobre m i n: assumim que la descomposició SVD existeix per matrius $(m-1)\times(n-1)$ i provem que existeix també per $m\times n$. També assumim $A \neq 0$, ja que aleshores la descomposició SVD és trivial. Com hem suposat que $m \geq n$, així que el primer cas per la inducció és el cas $n = 1$

Cas n=1: Escrivim $A = U\Sigma V^T$ amb $U = \frac{A}{||A||_2}$, $\Sigma = ||A||_2$ i $V = 1$. Cas inductiu: Escollim v tal que $||v||_2 == 1$ i $||A||_2 = ||Av||_2|0$. Sabem que existeix aquesta v per la definció de $||A||_2 = \max\limits_{||V||_2=1}||Av||_2$. Sigui $u = \frac{Av}{||Av||_2}$ un vector unitari. Escollim $\tilde{U}$ i $\tilde{V}$ tal

que $U = [u, \tilde{U}]$ és una matriu ortogonal $m \times m$ i $V = [v, \tilde{V}]$ és una matriu ortogonal $n \times n$. Aleshores, podem escriure:

$$U^T AV = \begin{bmatrix} u^T \\ \tilde{U}^T \end{bmatrix} \cdot A \cdot \begin{bmatrix} v & \tilde{V} \end{bmatrix} = \begin{bmatrix} u^T Av & u^T A\tilde{V} \\ \tilde{U}^T Av & \tilde{U}^T A\tilde{V} \end{bmatrix} \tag{3.3}$$

Aleshores,

$$u^T Av = \frac{(Av)^T (Av)}{||Av||_2} = \frac{||Av||_2^2}{||Av||_2} = ||Av||_2 = ||A||_2 \equiv \sigma \tag{3.4}$$

i $\tilde{U}^T Av = \tilde{U}^T u||Av||_2 = 0$. Podem afirmar que $u^T A\tilde{V} = 0$ ja que, si fos el contrari, $\sigma = ||A||_2 = ||U^T AV||_2 \geq ||[1, 0..., 0]U^T AV||_2 = ||[\sigma | u^T A\tilde{V}]||_2 > \sigma$.

Per tant, $\tilde{U}AV = \begin{bmatrix} \sigma & 0 \\ 0 & \tilde{U}^T A\tilde{V} \end{bmatrix} = \begin{bmatrix} \sigma & 0 \\ 0 & \tilde{A} \end{bmatrix}$. Ara apliquem la hipòtesi d'inducció a $\tilde{A}$ per obtenir $\tilde{A} = U_1 \Sigma_1 V_1^T$, on U_1 té dimensions $(m-1) \times (n-1)$, Σ té dimensions $(n-1) \times (n-1)$ i V_1 té dimensions $(n-1) \times (n-1)$. Per tant,

$$U^T AV = \begin{bmatrix} \sigma & 0 \\ 0 & U_1 \Sigma_1 V_1^T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^T \tag{3.5}$$

o, dit d'una altra forma:

$$A = (U \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}) \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix} (V \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix})^T \tag{3.6}$$

la qual és la descomposició que volíem demostrar. □

4 Un nou marc per analitzar el procés d'aprenentatge dels transformers de visió

L'objectiu d'aquest capítol és el d'explicar la part pràctica del treball. Primer es presenten les hipòtesis plantejades i els objectius plantejats en aquesta part pràctica. A continuació, s'exposen els detalls de la implementació i com s'ha desenvolupat el codi. Finalment, es presenten els resultats i se'n fa un anàlisi.

4.1 Les nostres hipòtesis sobre el procés de l'aprenentatge dels transformers

Els transformers, i en especial els transformers de visió, són sense dubte una de les famílies de xarxes neuronals més recents i amb més projecció en l'actualitat. Tanmateix, al ser tan recents encara hi ha poca informació respecte la utilitat i eficiència de cada model.

És per aquest motiu que, en aquest part del treball, l'objectiu principal es basa en analitzar-los i proposar un marc per comparar els transformers. Per poder il·lustrat el procés, els aplicarem sobre un conjunt d'imatges de menjar (Food-101), executant-los de la mateixa forma i, d'aquesta manera, simplificar la forma d'adaptar un model qualsevol de transformer per un conjunt d'imatges específic.

Entrant en detall, volem analitzar com aprèn cada un dels transformers de visió per, a posteriori, obtenir conclusions generals sobre els transformers de visió. Per analitzar-los, observarem com es modifiquen els blocs principals durant l'aprenentatge, com afecta la posició del bloc en l'aprenentatge i com varien durant el temps. Per il·lustrar aquest anàlisi, mostrarem el procés d'aprenentatge amb un conjunt de gràfics que ens permetran observar el comportament dels seus valors singular, les seves normes, els rangs i la seva variació.

Entrant en detall, la primera pregunta que ens hem realitzat és si totes les matrius dels transformers s'adapten (aprenen) al nostre conjunt de dades, o pel contrari algunes no tenen canvis i per tant no depenen de les dades. I en el cas de les matrius que s'adapten al nostre conjunt de dades, si és necessari que s'hi adaptin, és a dir, si tenen un impacte considerable en l'eficiència del transformer.

Per altra banda, la segona pregunta està relacionada amb la posició de la matriu dins del transformer es dir si té una implicació en l'eficiència del transformer. És a dir, els blocs inicials o finals són els quins s'adapten més al nostre conjunt de dades i tenen una implicació més directa en l'aprenentatge?!

Finalment, també estudiarem si les matrius dels transformers que tenen més impacte en l'aprenentatge de la xarxa són aquelles les quals tenen grans variacions durant l'entrenament. Per aquest motiu, plantegem dos criteris diferents per tal de decidir quan una matriu té canvis suficientment petits com per bloquejar-la:

1. Basant-nos en el canvi de les matrius tenint en compte l'evolució dels seus elements individuals a través de la següent fórmula:

$$C_1^k = \frac{1}{n^2} \sum_{i,j=0}^{n} |x_{i,j}^{k+1} - x_{i,j}^{k}| \tag{4.1}$$

   on k indica l'època d'entrenament i $x_{i,j}$ indica l'element en la fila i i columna j de la matriu per analitzar.

2. Basant-nos en el canvi de les matrius tenint en compte la seva incertesa, és a dir, utilitzant l'evolució de la suma total de la matriu juntament amb la desviació típica:

$$C_2^k = (\frac{1}{n^2}|\sum_{i,j=0}^{n} x_{i,j}^{k+1} - x_{i,j}^{k}|, \sqrt{\frac{\sum_{i,j=0}^{n}(x_{i,j}^{k+1} - x_{i,j}^{k}) - \bar{X})^2}{n^2}}) \tag{4.2}$$

on k indica l'època d'entrenament, $x_{i,j}$ indica l'element en la fila i i columna j i $\bar{X}$ és la mitjana dels valors $x_{i,j}^{k+1} - x_{i,j}^{k}$ $\forall i, j$.

5 Implementació de l'entorn per analitzar els transformers

Respecte la implementació, utilitzem el llenguatge de programació Python, juntament amb la biblioteca de programari de codi obert PyTorch. Aquesta biblioteca ens permet treballar amb matrius i tensors i utilitzar la plataforma de computació paral·lela CUDA.

Per obtenir els diferents transformers, fem ús de la biblioteca Timm [61], la qual conté la majoria de transformers de visió que utilitzem en aquest treball. En el cas del transformer CSWin hem decidit utilitzar el repositori oficial ja que és tant recent que, a data de la realització d'aquest treball, encara no ha estat afegit en la biblioteca Timm. Per poder-lo executar de manera anàloga a la resta de transformers, hem fet les modificacions corresponents a la biblioteca Timm per tal que el model hi estigui present.

Finalment, farem ús del descens del gradient estocàstic present en la biblioteca Timm, en el mòdul *optim*, que obtindrem a partir del mètode proporcionat *create_optimizer_v2*.

5.1 Dataset

Com ja hem comentat en apartats anteriors, en aquest treball hem seleccionat el conjunt de dades Food-101 per tal de realitzar els diferents anàlisis. Aquest conjunt de dades està format per 101 categories diferents de menjar i cada categoria conté 1000 imatges.

Figura 11: Conjunt de dades Food-101 [5]

Food-101 està disponible de manera totalment gratuïta en la plataforma Kaggle i el fitxer original està estructurat amb dos directoris principals: un amb les imatges i l'altre conté metadades, de les quals les més importants són dos fitxers de text que indiquen quines imatges són del conjunt de validació i quines del conjunt d'entrenament.

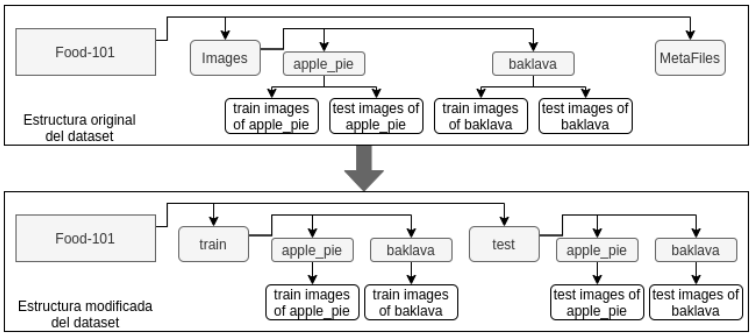

Figura 12: Reestructuració del conjunt d'imatges Food-101 [5]

Tenint aquesta estructura, el primer que vam realitzar va ser reestructurar el directori amb les imatges per tal de tenir una estructura més estàndard i uniforme tal i com s'indica en la figura [12], assegurant d'aquesta forma que en totes les proves que realitzaríem, les imatges d'entrenament i de validació fossin idèntiques.

5.2 Configuració

En totes les xarxes neuronals hi ha un seguit de paràmetres que s'estableixen de tal manera que la xarxa pot obtenir un resultat diferent amb uns valors diferents en aquests paràmetres. Abans d'entrar en detall amb els parámetres utilitzats, mencionar que les configuracions realitzades poden no ser òptimes ja que l'objectiu no és obtenir la millor precisió possible en cadascun dels transformers, sinó realitzar una comparació equitativa entre els diferents transformers de visió.

Durant les múltiples execucions que hem realitzat, hem utilitzat un *learning rate* de 10^{-3} en el SGD i una mida de batch de 32, és a dir, hem separat les imatges en grups de 32.

A més, hem realitzat dos processos de *data augmentation* [43]: un pel conjunt d'imatges d'entrenament i l'altre pel conjunt d'imatges de validació. Aquests dos processos ens han servit per reduir encara més el problema d'*overfitting* [67], el qual ocorre quan el model s'ajusta a les imatges de l'entrenament a partir de característiques concretes que no tenen relació causal amb la tasca per la qual s'entrena la xarxa.

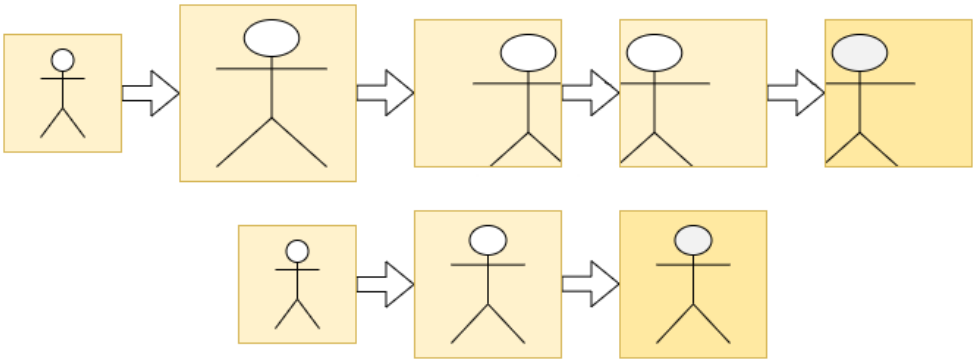

Figura 13: Processos de *data augmentation* realitzats. A la part superior, el procés realitzat en el conjunt d'imatges d'entrenament. A la part inferior, el procés realitzat en el conjunt d'imatges de test.

D'altra banda, s'utilitzen els models *base* pre-entrenats amb ImageNet i amb imatges de resolució 224×224 disponibles en la biblioteca Timm, excepte el model del CSWin, el qual s'utilitza el disponible en el repositori oficial del CSWin. És necessari destacar que s'utilitzen aquests models pre-entrenats degut a la necessitat d'una quantitat ingent de dades per tal d'entrenar els transformers i és per aquest motiu que s'utilitzen aquests models pre-entrenats per adaptar-los a conjunts d'imatges insuficientment grans.

Finalment, per reforçar la hipòtesis sobre la importància de la posició del bloc dins del transformer, hem realitzat unes execucions bloquejant, ja des d'un inici, el 33% dels blocs inicials o el 33% dels blocs finals.

5.3 Mètriques per analitzar el canvi de les matrius

Amb l'análisi plantejat es poden fer servir diferents mètriques com poden ser els canvis de les matrius (figures 4.1, 4.2) i les normes, els rangs i els valors singulars de les diferents matrius. En aquest treball fem un anàlisi de les diferents mètriques i hem decidit utilitzar la primera en els experiments, ja que aquesta mètrica tendeix a 0 i descriu d'una manera més clara la convergència de les matrius.

Un cop decidida la mètrica, hem utilitzat dos criteris per tal de decidir quines matrius bloquejàvem, els quals hem escollit els seus llindars empíricament. En el primer (eq. 4.1), hem situat el llindar en un valor de $5 \cdot 10^{-4}$ i en el segon (eq. 4.2), hem situat el primer llindar en $5 \cdot 10^{-3}$ i el llindar de la desviació típica al 10^{-1}.

5.4 El procés de l'aprenentatge dels transformers

En aquest apartat, es presenta un anàlisi descriptiu sobre els objectius plantejats de la part práctica, basant-se en un seguit de gràfics i taules que serviran de suport als nostres resultats.

5.4.1 L'aprenentatge dels transformers i les matrius dels pesos

La primera pregunta que es tracta és si totes les matrius participen al procés d'aprenentatge d'igual manera, és a dir si totes les matrius s'adapten al conjunt de dades i de quina manera. Per respondre aquesta pregunta, començarem comentant algunes similituds que hem observat. En el gràfic 14 tenim representada l'evolució de les normes dels diferents blocs que formen el transformer ViT. En aquest gràfic, cada color representa un bloc diferent, tal i com es pot observar en la llegenda, l'eix horitzontal indica l'etapa de l'entrenament i l'eix vertical el valor de la norma en qüestió. Tal i com es pot observar, en tots els casos les normes acaben convergint i són molt estables. En alguns blocs, observem com en les primeres etapes hi ha un creixement considerable en la majoria de blocs, sent els blocs 0 i 1 els únics que tenen una disminució de la norma. Aquesta disminució també s'aprecia en la primera capa dels dos transformers jeràrquics analitzats, el Swin i el CSWin, tanmateix en la resta de matrius W_Q i W_k les normes són invariants en el temps. Respecte el creixement que observem en les matrius W_v i $W0$, també el podem observar en la majoria dels transformers incloses les matrius MLP.

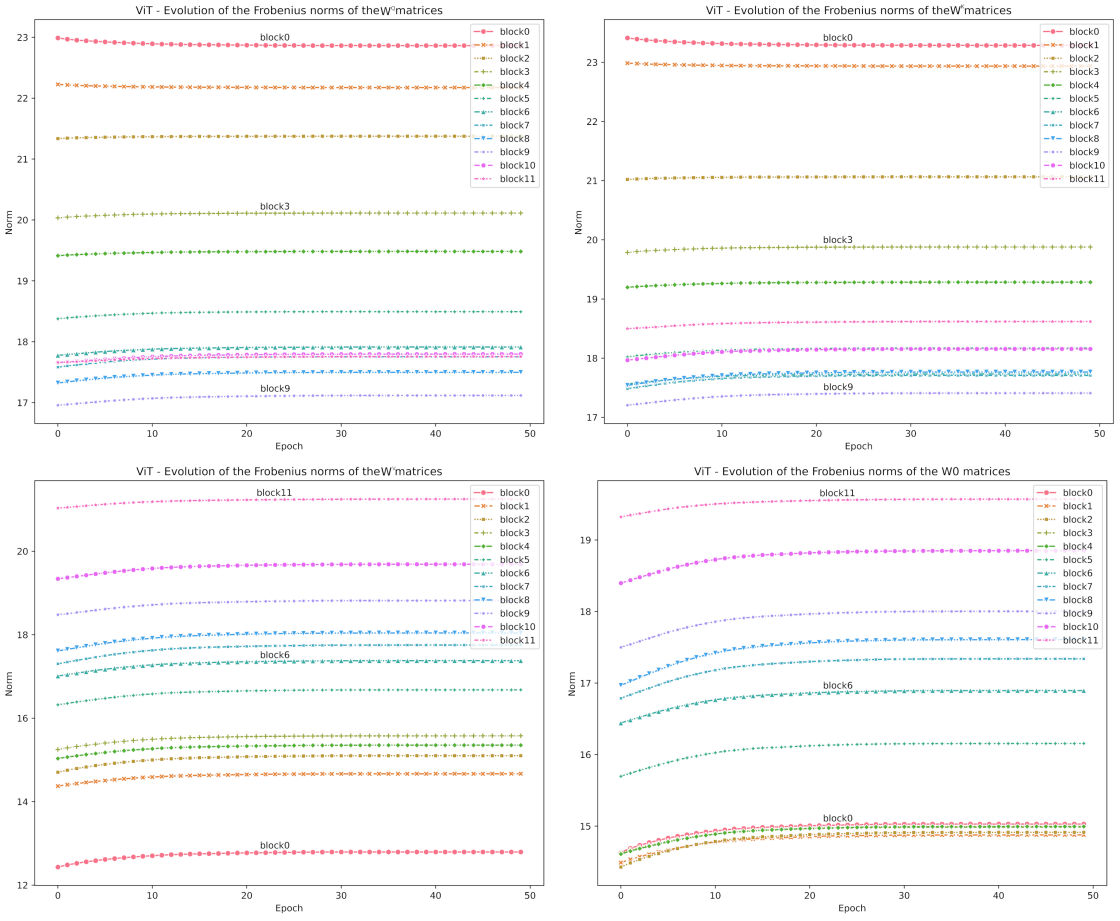

Figura 14: Normes de Frobenius de les matrius W_Q, W_k, W_vi $W0$ del transformer ViT durant l'entrenament amb Food-101.

En els gràfics de la figura 15 també podem observar com, tal i com era esperable, degut al comportament del descens del gradient estocàstic, les matrius acaben convergint

en tots els blocs. En aquests gràfics, cada color representa un bloc diferent seguint la llegenda, l'eix horitzontal indica l'etapa de l'entrenament i l'eix vertical indica la suma de totes les diferències de la matriu en valor absolut, és a dir, en l'eix vertical es mostra el resultat de $\sum_{i,j=0}^{n} |x_{i,j}^{k+1} - x_{i,j}^{k}|$, on $x_{i,j}^{k}$ indica l'element de la fila i i columna j de la matriu en l'etapa k de l'entrenament. Fixem-nos que alguns d'aquests blocs tenen diferències considerablement més petites que la resta de blocs, fet que ens reforça la hipòtesi que no tots els blocs s'adapten al nostre conjunt d'imatges de la mateixa manera. Finalment, fixem-nos també que les matrius W_Q i W_k tenen aproximadament la meitat de diferències que les altres dues matrius. En la resta de transformers, també observem les mateixes característiques, destacant encara més el fet que algunes matrius tenen significantment menys variacions.

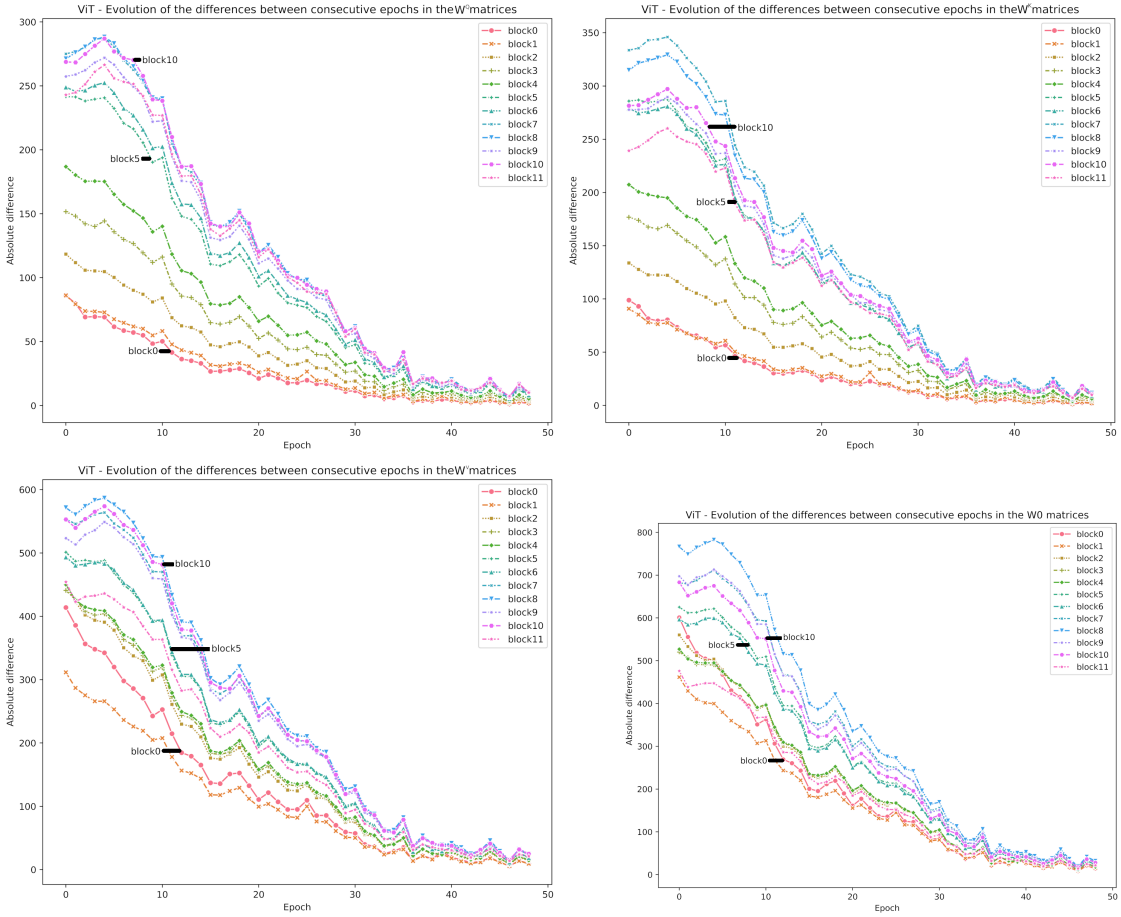

Figura 15: Evolució de les matrius W_Q, W_k, W_v i $W0$ en el transformer ViT durant l'entrenament amb Food-101. En l'eix vertical es mostra la suma de les diferències en valor absolut de cada matriu en la època indicada en l'eix horitzontal i la mateixa matriu en l'época posterior.

Podem afirmar que no totes les matrius s'adapten al nostre conjunt d'imatges, o almenys que no totes les matrius tenen la mateixa importància en l'entrenament també a partir de les gràfiques de la figura 16, en les quals apreciem que els valors singulars d'algunes matrius augmenten considerablement envers altres matrius en les qual no apreciem

canvis substancials. En aquestes gràfiques s'hi representa l'evolució dels valors singulars de les matrius atencionals i MLP de manera que cada color diferent representa una etapa d'entrenament seguint la llegenda, l'eix vertical mostra el valor del valor singular i l'eix horitzontal mostra la posició del valor singular en la matriu Σ de la descomposició SVD. Tal i com podem observar, en les matrius W_v i $W0$ s'observa un augment substancial en els valors singulars, principalment en els 60 primers valors singulars i, en canvi, els canvis en les matrius MLP, i sobretot en les matrius W_Q i W_k, són insignificants. Aquest fet també és apreciable en la resta de transformers, sent les matrius W_v i $W0$ les que tenen un augment més considerable dels valors singulars.

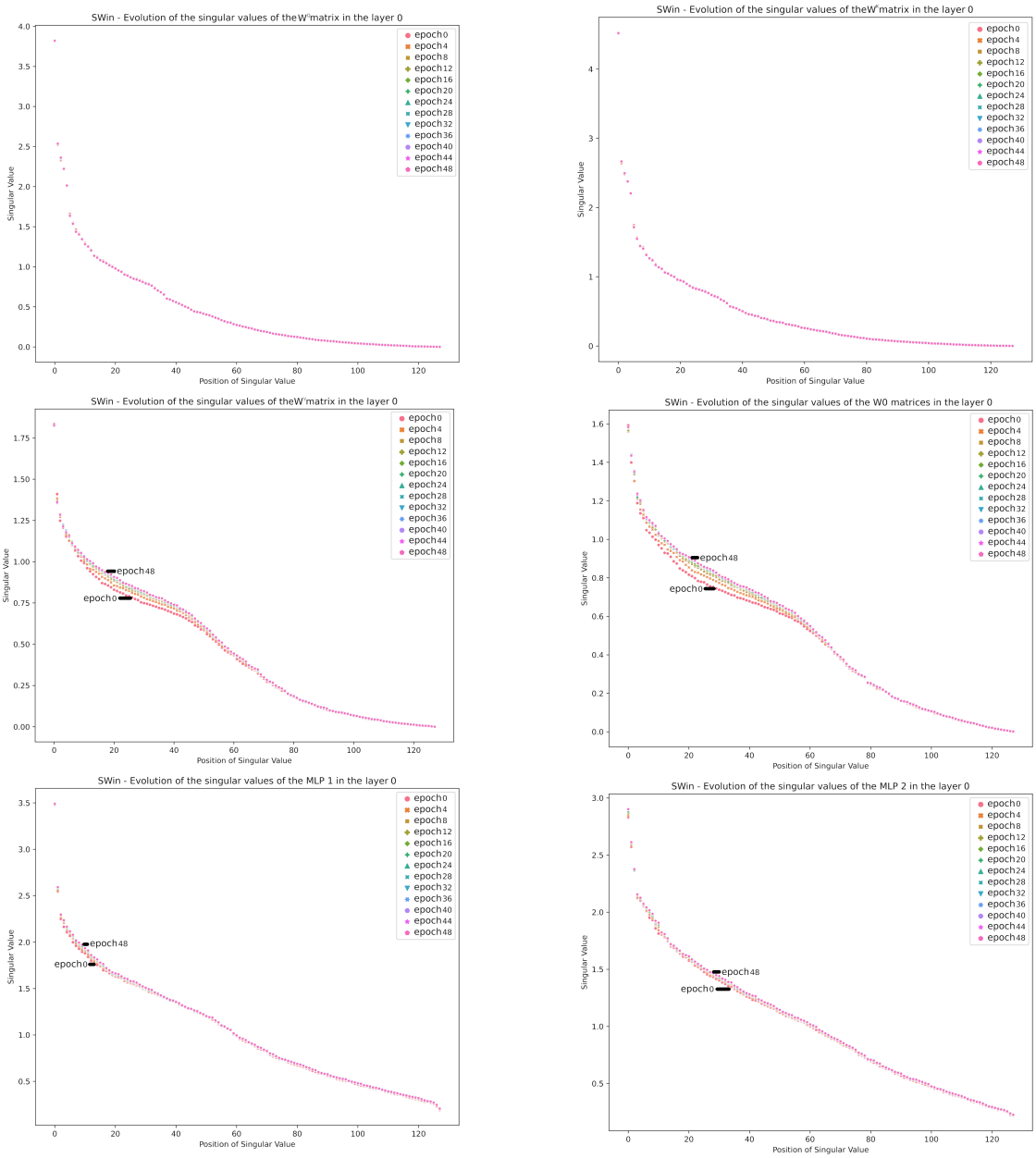

Figura 16: Evolució dels valors singulars de les matrius dels blocs atencionals i MLP de la layer 0 del transformer SWin durant l'entrenament amb Food-101.

En conclusió, observem que les matrius W_Q i W_k tenen, independentment del transformer de visió, un comportament similar en tots els aspectes: evolució dels valors singulars 16, diferència entre els blocs 15 i evolució de les normes 14, etc. Alhora, en alguns casos podem apreciar com aquest comportament difereix del de les matrius W_v i W_0, com es pot observar en les figures 14 i 16.

Per altra banda, també concluim que no totes les matrius participen d'igual manera en el procés d'aprenentatge, és a dir, algunes matrius són més invariants durant el procés d'entrenament, principalment entre diferents blocs, però també entre tipus de matriu, com podem observar en la figura 15 en la qual apreciem que les matrius W_v i $W0$ tenen el doble de diferències en valor absolut que les matrius W_Q i W_k, fet que ens indica que aquestes dues matrius són molt més importants en el procés d'aprenentatge que no pas W_Q i W_k.

5.4.2 Com afecta la posició del bloc?

Com podem visualitzar en els gràfics de la figura 17, en general els blocs superiors tenen valors singulars superiors excepte en els valors més alts, els quals pertanyen als blocs inferiors. En aquesta gràfica l'eix d'abscisses indica la posició del valor singular en la matriu Σ de la descomposició SVD de la matriu, l'eix d'ordenades indica el valor del valor singular i els colors són els diferents blocs que formen el transformer BEiT. Aquesta característica la podem observar en tots els transformers de visió que hem utilitzat en aquest anàlisi i, en conseqüència, podem afirmar que els blocs superiors tenen valors singulars més regulars i els blocs inferiors valors singulars més desiguals. Finalment, fixem-nos que en aquest gràfic també observem la conclusió de l'apartat anterior: les matrius W_Q i W_k són molt similars entre elles envers les altres dues matrius que conformen el bloc atencional.

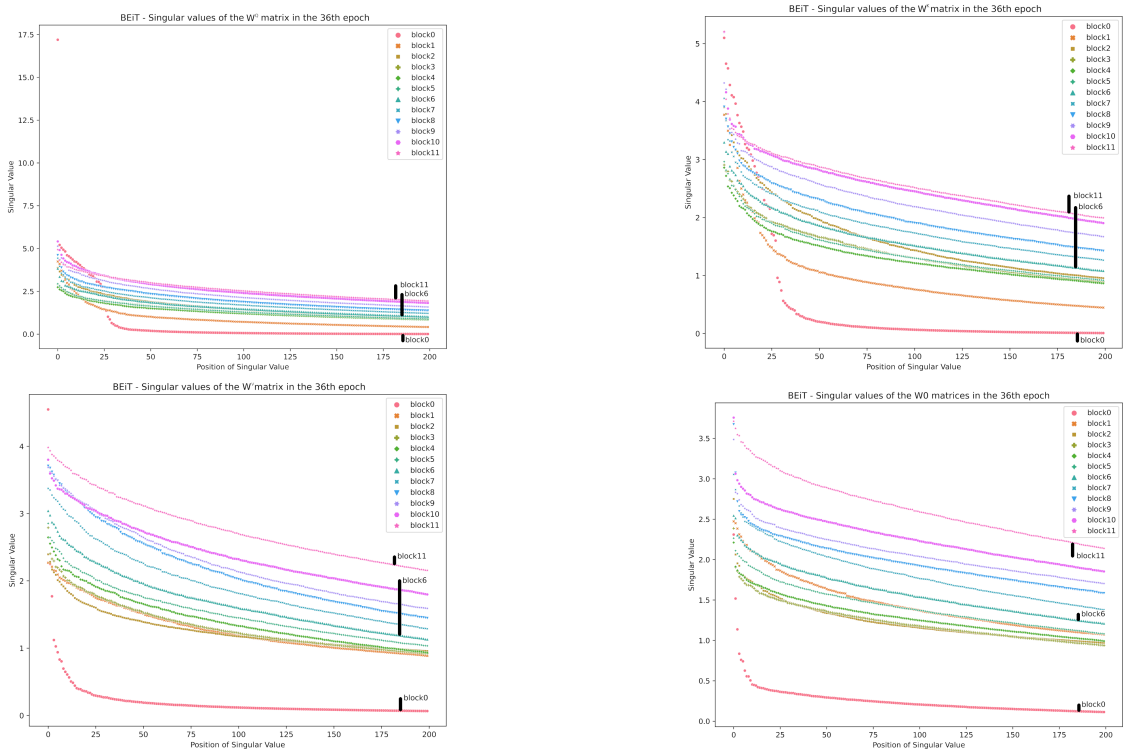

Figura 17: Valors singulars de les matrius W_Q, W_k, W_vi $W0$ en l'etapa 36 de l'entrenament del transformer BEiT amb Food-101.

El fet que en general, els blocs superiors tenen valors singulars superiors, excepte en els valors singulars més alts, també és observable en la figura [14], en la qual veiem com les normes de les matrius W_Q i W_k van disminuint a mesura que augmentem la posició del bloc i les matrius W_v i $W0$ augmenten, ja que la norma serà decreixent entre els blocs si la major part dels valors singulars més grans pertanyen als blocs inferiors, independentment de l'ordre en els valors singulars més petits. Aquesta també és una propietat que trobem en tots els transformers, amb l'excepció del transformer BEiT, en el qual els blocs inicials tenen menys valors singulars alts en comparació a la resta dels transformers i, per tant, la norma dels blocs superiors també és superior a la norma dels inferiors.

Encara més en general, aquesta característica també l'observem en els blocs MLP tal i com podem observar en la figura 18 on els blocs superiors també tenen normes superiors. En aquest gràfic, cada color representa un bloc diferent, tal i com es pot observar en la llegenda, l'eix horitzontal indica l'etapa de l'entrenament i l'eix vertical el valor de la norma en qüestió. Per tant, podem afirmar que sí que és una característica comuna en els transformers de visió.

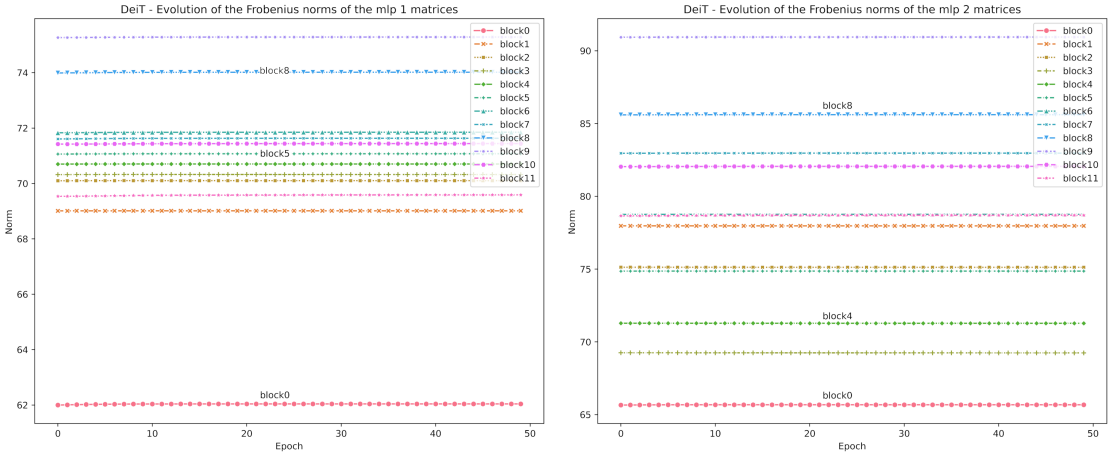

Figura 18: Normes de Frobenius de les matrius MLP del transformer DeiT durant l'entrenament amb Food-101.

Finalment, juntament amb les gràfiques de la figura 15 podem afirmar com els blocs inicials són els quins s'adapten en menys mesura al nostre conjunt de dades. Juntament amb els resultats obtinguts per Kaiming [25], aquest fet ens fa afirmar que els blocs inicials presten atenció a característiques més generals i observables de les imatges, mentres que els blocs superiors atenen a característiques més complexes i un nivell d'abstracció més elevat, fent que siguin aquests últims els que tinguin més importància en l'adaptació a un conjunt d'imatges.

Vistes les conclusions obtingudes, hem realitzat un gràfic que mostra l'evolució de les precisió en el transformer ViT sense cap matriu bloquejada, amb les matrius dels blocs inicials bloquejades i amb les matrius dels blocs finals bloquejades.

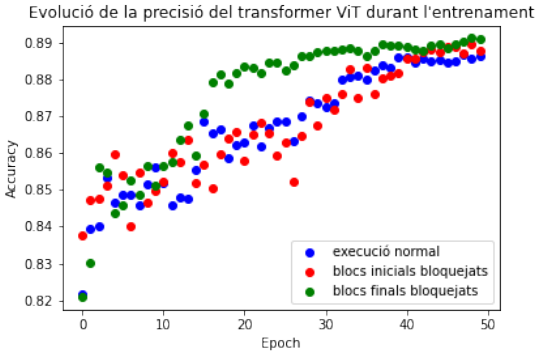

Figura 19: Evolució de la precisió del transformer ViT durant l'entrenament amb Food-101 amb cap matriu bloquejada, les matrius dels blocs inicials bloquejades i amb les matrius dels blocs finals bloquejades.

En aquest gràfic, cada color indica a quina execució pertany, l'eix horitzontal indica l'etapa d'entrenament i l'eix vertical la precisió del transformer. Tal i com podem observar, destaca el fet que bloquejant els blocs inicials, la precisió del transformer convergeix molt més ràpidament i bloquejant els blocs finals, li costa més convergir. Aquest fet reafirma

les conclusions obtingudes, és a dir, que els blocs finals són els més importants a l'hora d'adaptar-se al nostre conjunt d'imatges.

Tenint unes conclusions clares i validades, en el següent apartat aprofundirem en quin criteri seguir per bloquejar les matrius i quin efecte comporta en l'entrenament.

5.5 Optimització del procés d'aprenentatge dels transformers

En aquest apartat, es presenten els resultats dels diferents experiments realitzats per tal de poder reforçar les conclusions obtingudes en l'anàlisi de l'apartat anterior. Concretament es fa èmfasis en investigar si les matrius amb més variació són les que realment tenen més impacte en l'entrenament.

Amb les execucions sense bloquejar cap matriu, hem obtingut la taula 4, la qual ens ha servit per escollir els criteris 1 i 2 com a mètriques per tal de bloquejar els blocs. Aquesta taula ha estat realitzada a través del paràmetre dels transformers que conté les matrius W_Q, W_k i W_v i les hem avaluat juntes degut a que l'arquitectura dels diferents transformers utilitzat no permet el bloqueig independent d'aquestes matrius. Per tant, és més adequat avaluar les diferent mètriques amb les tres matrius juntes.

Entrant en detall, en aquesta taula $Transf$ indica el transformer, $Block$ el bloc atencional corresponent, $Total$ indica la diferència entre la mètrica en la primera etapa d'entrenament i la última, $Mean$ la mitjana, Min el mínim de la mètrica en tot l'entrenament, Var la desviació típica, $C1$ els valors de la fórmula del criteri 1, $C2$ els valors del primer element de la fórmula del criter 2, $Norm$ el valor de la norma de Frobenius, $SVAvg$ la mitjana dels valors singulars i, finalment, $Rank$ el rang de la matriu.

Transf.	Block	C1 Total	C1 Mean	C1 Min	C1 Var	C2 Total	C2 Mean	C2 Min	C2 Var
ViT	1	1.759e-3	1.764e-4	8.971e-6	1.518e-4	2.591e-7	1.437e-7	9.154e-11	1.698e-7
ViT	6	4.384e-3	4.774e-4	2.661e-5	3.915e-4	2.132e-6	1.862e-7	1.057e-10	2.319e-7
ViT	11	5.119e-3	4.829e-4	3.337e-5	3.845e-4	2.281e-6	1.342e-7	3.662e-9	1.324e-7
BEiT	1	1.09e-3	9.836e-5	1.014e-5	1.190e-4	6.463e-7	1.581e-7	1.519e-9	2.871e-7
BEiT	6	3.030e-3	2.166e-4	3.313e-5	2.073e-4	4.575e-7	9.386e-8	6.448e-9	1.179e-7
BEiT	11	1.957e-3	8.289e-5	1.213e-5	8.544e-5	2.332e-7	3.230e-8	6.961e-10	3.720e-8
DeiT	1	3.853e-4	2.889e-5	5.755e-6	3.184e-5	2.830e-7	2.696e-8	6.275e-11	3.936e-8
DeiT	6	2.683e-3	1.591e-4	3.339e-5	1.651e-4	3.299e-7	4.945e-8	3.600e-10	5.062e-8
DeiT	11	3.545e-3	1.675e-4	3.410e-5	1.709e-4	6.158e-7	5.215e-8	1.025e-9	6.432e-8
SWin	1[5]	1.276e-3	1.555e-4	4.787e-5	9.810e-5	1.469e-6	1.304e-7	2.397e-9	1.257e-7
SWin	6	1.935e-3	2.254e-4	9.225e-5	1.196e-4	1.027e-6	1.217e-7	4.675e-11	1.523e-7
SWin	11	1.769e-3	1.756e-4	9.733e-5	6.703e-5	2.460e-7	1.027e-7	1.012e-8	8.427e-8
CSWin	1[4]	6.403e-4	6.000e-5	4.262e-6	6.165e-5	5.857e-8	4.917e-8	3.407e-10	7.893e-8
CSWin	6	8.593e-4	7.138e-5	6.050e-6	6.634e-5	8.224e-8	2.626e-8	1.997e-9	3.521e-8
CSWin	11	1.041e-3	7.683e-5	7.400e-6	6.939e-5	5.327e-7	2.506e-8	1.468e-10	3.283e-8

[5]Per la realització d'aquesta taula, s'ha agafat els blocs de la capa 3 tant del transformer SWin com CSWin.

Transf.	Block	Norm Total	Norm Mean	Norm Var	SV Avg Total	SV Avg Mean	SV Avg Min	SV Avg Var	Rank Total	Rank Mean
ViT	1	0.055	35.10	0.016	5.050e-3	0.815	0.810	1.323e-3	0	768
ViT	6	0.387	30.55	0.099	0.012	0.970	0.960	3.060e-3	0	768
ViT	11	0.256	33.32	0.068	8.570e-3	1.128	1.121	2.278e-3	0	768
BEiT	1	0.084	31.68	0.018	2.719e-3	0.837	0.834	5.893e-4	0	768
BEiT	6	0.070	47.52	0.016	2.216e-3	1.503	1.502	5.357e-4	0	768
BEiT	11	0.021	80.92	5.739e-3	7.348e-4	2.723	2.723	1.989e-4	0	768
DeiT	1	-8.087e-4	55.53	3.089e-4	2.334e-4	1.382	1.382	5.478e-5	0	768
DeiT	6	0.039	62.16	8.330e-3	1.286e-3	1.869	1.868	2.864e-4	0	768
DeiT	11	0.050	60.70	0.011	1.753e-3	2.000	1.999	3.998e-4	0	768
SWin	1[6]	0.024	48.00	6.453e-3	1.265e-3	1.597	1.596	3.367e-4	0	512
SWin	6	0.041	44.48	0.012	1.815e-3	1.684	1.682	5.017e-4	0	512
SWin	11	0.031	45.25	7.974e-3	1.410e-3	1.730	1.729	3.770e-4	0	512
CSWin	1[4]	0.019	21.17	5.214e-3	1.243e-3	0.858	0.857	3.271e-4	0	384
CSWin	6	0.036	21.62	7.888e-3	2.023e-3	0.935	0.934	4.687e-4	0	384
CSWin	11	0.063	21.41	0.015	3.126e-3	0.954	0.951	7.622e-4	0	384

Taula 4: Comparativa de les mètriques C1, C2, norma, mitjana de valors singulars i rang de les matrius de pesos W_Q, W_k i W_v concatenades en el Food-101.

En la taula, podem observar com els valors dels dos criteris són considerablement més petits, però, alhora, són els quins tenen, en percentatge, més variació, ja que els valors de la mitjana i el mínim són molt similars en la resta de mètriques. Aquest fet va lligat a l'observat en les gràfiques de la figura 15, ja que s'observa clarament la variació de la mètrica. També podem apreciar com el rang és màxim en tots els blocs i durant tot l'entrenament, fet que fa descartar aquesta mètrica.

Encara sobre la taula 4, també és important ressaltar el fet que podem comprovar com, encara que normes de les matrius W_Q i W_k en el transformer ViT disminueixen en les primeres etapes de l'entrenament, es compensa amb la norma de la matriu W_v i, per aquest motiu, acaba sortint un resultat positiu en $NormTotal$.

Seguint els criteris descrits a la planificació i amb els paràmetres descrits a la implementació, hem obtingut la següent taula, on *C1* i *C2* fan referència als criteris 1 i 2 respectivament, *Acc* fa referència a la precisió del model (en valors de 0 a 1), *Time* al temps que ha tardat en fer les 50 èpoques d'entrenament en segons i *Avg* la mitjana de totes les execucions realitzades sense bloquejar cap matriu.

Model	Avg Acc	C1 Acc	C2 Acc
ViT	0.8858	0.8878	**0.8970**
BEiT	**0.9057**	0.9017	0.9013
DeiT	**0.8759**	0.8635	0.8654
SWin	0.9110	**0.9162**	0.9056
CSWin	0.8951	**0.8960**	0.8941

Model	Avg Time	C1 Time	C2 Time
ViT	71632	**44932**	49545
BEiT	96995	**60843**	66267
DeiT	74730	48873	**43617**
SWin	114172	**74059**	84121
CSWin	126716	114998	**81982**

Taula 5: Comparativa dels resultats en Food-101 obtinguts sense modificacions i amb les modificacions dels experiments.

A partir de la taula elaborada, s'observa com seguint el criteri 1 s'ha perdut, de mitjana, un 0.166% de precisió, però s'ha reduït el temps en un 30.7% aproximadament. D'altra banda, seguint el criteri 2, s'han obtingut resultats similars, perdent, de mitjana, un 0.22% però reduint el temps en un 33.15%. Aquests resultats permeten afirmar que

[6]Per la realització d'aquesta taula, s'ha agafat els blocs de la capa 3 tant del transformer SWin com CSWin.

realment, les matrius que tenen grans variacions durant l'entrenament són aquelles que influeixen en gran mesura en l'adaptació d'un transformer a un conjunt de dades específic.

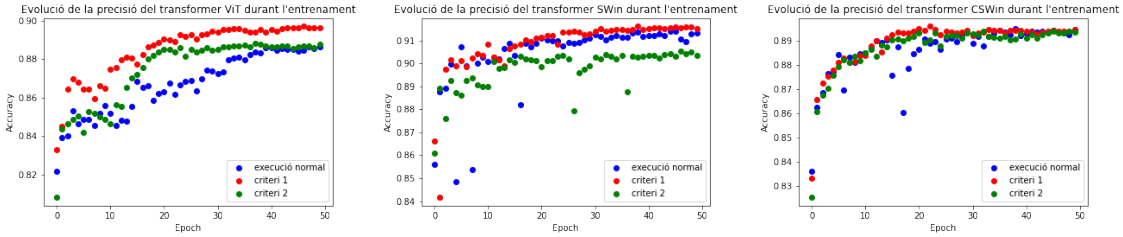

Figura 20: Evolució dels la precisió dels transformers ViT, SWin i CSWin durant l'entrenament amb Food-101 en la millor execució sense bloquejar cap matriu i en les dues execucions realitzades amb els dos criteris plantejats.

A més, en la figura 20 s'observa com el fet de bloquejar les matrius amb menys canvis també afavoreix a que el transformer s'estabilitzi amb menys èpoques, fet que indueix a una reducció de les épocques necessàries per adaptar els transformers al conjunt d'imatges específic. Aquest fet, juntament amb la millora del temps considerable que es pot observar en les taules de la figura 5, ens porta a afirmar que en entrenaments amb conjunts d'imatges relativament petits, es podrien bloquejar algunes matrius optimitzant algun dels criteris utilitzats en aquest treball amb l'objectiu d'accelerar l'entrenament i facilitar l'adopció dels transformers en tasques en les quals no es precisa d'una GPU suficientment potent.

Finalment, destacar que en els experiments les matrius W_Q, W_k i W_v s'han bloquejat en el mateix moment, és a dir, no es pot bloquejar la matriu W_Q sense bloquejar les altres dues matrius. Aquest impediment és degut a la implementació dels diferents transformers, els quals agrupen aquestes tres matrius en un mateix paràmetre. Aquest fet, ha comportat que no podem extreure conclusions respecte les diferències de les tres matrius en aquest apartat.

6 Conclusions

A la finalització d'aquest treball, s'han aconseguit els principals objectius que s'havien plantejat a l'inici d'aquest treball.

Primer de tot, s'han assolit els coneixements necessaris en els qual es fomenten els transformers i, com a tal, s'ha plasmat de la manera més formal possible en els apartats d'Estat de l'Art i Background teòric. Personalment, estic satisfet amb el nivell aconseguit que, lluny de ser perfecte, considero que m'ha permés entendre realment la rellevància i innovació que suposen tots els diferents models analitzats. Alhora m'ha permès anar més enllá en els objectius plantejats incialment, motivant-me per, a la finalització d'aquest treball, seguir avançant en el desenvolupament d'aquest treball amb l'objectiu de publicar un article.

Respecte els objectius més pràctics, s'ha proposat un marc teóric per analitzar el procés d'aprenentatge dels transformers de visió el qual s'ha demostrat la seva utilitat en els diferents experiments realitzats. Al mateix temps, s'ha demostrat l'impacte de les nostres heurístiques en la optimització del procés d'aprenentatge dels transformers i els ha fet més estables. Aquestes heurístiques es podrien optimitzar per cada transformer i per tant, aquesta reducció del temps d'execució podria inclús augmentar.

Finalment, s'han entrenat i executat d'una manera anàloga els diferents transformers que s'havien plantejat, obtenint resultats competitius. De fet, els resultats aconseguits superen en precisió els models anteriors ja que el millor resultat publicat en la classificació de Papers With Code, TWIST [58], té una precisió de 89.3%, una precisió inferior a la màxima de 91.62% obtinguda pel transformer SWin utilitzant el criteri 1 per bloquejar dinàmicament les matrius.

7 Treball futur

Els objectius i resultats d'aquest treball no han aspirat a ser el punt final d'una investigació, sinó just el contrari. Com a tal, aquest treball acaba destacant les possibles continuacions que es podrien realitzar.

Primer de tot, destacar que, encara que el marc de comparació sigui funcional, no es pot considerar que sigui exhaustiu i, per tant, es poden investigar noves propietats de les matrius. Per exemple, estenent les observacions realitzades als vectors de biaix, analitzant també la seva convergència, si també hi ha biaixos que s'adapten més al conjunt de dades que d'altres, etc. D'altra banda, també es pot estendre incloent més transformers de visió dels analitzats en aquest treball.

Seguidament, també es pot realitzar l'análisi que s'ha realitzat en aquest treball amb els paràmetres òptims per cada transformer. En aquest treball, obtenir la millor precisió possible en cada transformer no ha estat un dels objectius, sinó que era realitzar una comparació dels resultats obtinguts amb els mateixos paràmetres en tots els transformers. Podria ser que el comportament variés de formes diferents depenent del transformer? Aquesta és una pregunta que, al seu temps, pot obrir més línies d'investigació.

Finalment, una altra línia d'investigació es podria obrir amb un anàlisi més profund sobre els valors singulars de les matrius. Per exemple, es podria mirar quina informació aporten els valors singulars més propers a 0 i, en cas que no aportessin una informació substancial o fins i tot tinguessin implicacions en el soroll de la xarxa, es podria intentar aproximar aquests valors singulars a 0 per tal d'optimitzar els càlculs i fer més robust el transformer. En l'annex es pot observar un primer análisi d'aquesta línia d'investigació proposada.

Referències

[1] Jay Alammar, *The illustrated transformer*, 2018.

[2] Josh Alman and Virginia Vassilevska Williams, *A refined laser method and faster matrix multiplication*, CoRR **abs/2010.05846** (2020).

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, *Neural machine translation by jointly learning to align and translate*, 2016.

[4] Hangbo Bao, Li Dong, and Furu Wei, *Beit: BERT pre-training of image transformers*, CoRR **abs/2106.08254** (2021).

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool, *Food-101 – mining discriminative components with random forests*, Computer Vision – ECCV 2014 (Cham) (David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, eds.), Springer International Publishing, 2014, pp. 446–461.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, *Language models are few-shot learners*, CoRR **abs/2005.14165** (2020).

[7] Ovidiu. Calin, *Deep learning architectures a mathematical approach / by ovidiu calin.*, 2020.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, *End-to-end object detection with transformers*, CoRR **abs/2005.12872** (2020).

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, *Emerging properties in self-supervised vision transformers*, CoRR **abs/2104.14294** (2021).

[10] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling, *Autoformer: Searching transformers for visual recognition*, CoRR **abs/2107.00651** (2021).

[11] Francois Chollet, *Xception: Deep learning with depthwise separable convolutions*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[12] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang, *Dynamic head: Unifying object detection heads with attentions*, CoRR **abs/2106.08322** (2021).

[13] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan, *Coatnet: Marrying convolution and attention for all data sizes*, CoRR **abs/2106.04803** (2021).

[14] James W Demmel, *Applied numerical linear algebra*, SIAM, 1997.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *ImageNet: A Large-Scale Hierarchical Image Database*, CVPR09, 2009.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *BERT: pre-training of deep bidirectional transformers for language understanding*, CoRR **abs/1810.04805** (2018).

[17] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo, *Cswin transformer: A general vision transformer backbone with cross-shaped windows*, CoRR **abs/2107.00652** (2021).

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, CoRR **abs/2010.11929** (2020).

[19] John Duchi, Elad Hazan, and Yoram Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research **12** (2011), no. 61, 2121–2159.

[20] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur, *Sharpness-aware minimization for efficiently improving generalization*, CoRR **abs/2010.01412** (2020).

[21] Kunihiko Fukushima, *Neocognitron: A hierarchical neural network capable of visual pattern recognition*, Neural Networks **1** (1988), no. 2, 119–130.

[22] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph, *Simple copy-paste is a strong data augmentation method for instance segmentation*, CoRR **abs/2012.07177** (2020).

[23] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao, *Knowledge distillation: A survey*, CoRR **abs/2006.05525** (2020).

[24] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang, *Transformer in transformer*, CoRR **abs/2103.00112** (2021).

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, CoRR **abs/1512.03385** (2015).

[26] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh, *Rethinking spatial dimensions of vision transformers*, CoRR **abs/2103.16302** (2021).

[27] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky, *Neural networks for machine learning lecture 6e*, 2012.

[28] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean, *Distilling the knowledge in a neural network*, ArXiv **abs/1503.02531** (2015).

[29] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi, *Semask: Semantically masked transformers for semantic segmentation*, 2021.

[30] Steven G. Johnson, *Notes on the equivalence of norms.*

[31] Diederik P. Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, 2017.

[32] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby, *Large scale learning of general visual representations for transfer*, CoRR **abs/1912.11370** (2019).

[33] Alex Krizhevsky, *Learning multiple layers of features from tiny images*, (2009), 32–33.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[35] S. Kullback and R. A. Leibler., *On information and sufficiency*, Ann. Math. Statist.

[36] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao, *Grounded language-image pre-training*, 2021.

[37] Ji Lin, Chuang Gan, and Song Han, *Temporal shift module for efficient video understanding*, CoRR **abs/1811.08383** (2018).

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, *Microsoft coco: Common objects in context*, Computer Vision – ECCV 2014 (Cham) (David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, eds.), Springer International Publishing, 2014, pp. 740–755.

[39] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He, *A survey of visual transformers*, CoRR **abs/2111.06091** (2021).

[40] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo, *Swin transformer V2: scaling up capacity and resolution*, CoRR **abs/2111.09883** (2021).

[41] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, *Swin transformer: Hierarchical vision transformer using shifted windows*, CoRR **abs/2103.14030** (2021).

[42] Warren S. McCulloch and Walter Pitts, *A logical calculus of the ideas immanent in nervous activity*, The Bulletin of Mathematical Biophysics **5** (1943), no. 4, 115–133.

[43] Agnieszka Mikołajczyk and MichałGrochowski, *Data augmentation for improving deep learning in image classification problem*, 05 2018, pp. 117–122.

[44] Michael A. Nielsen, Determination Press, 2015.

[45] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le, *Meta pseudo labels*, CoRR **abs/2003.10580** (2020).

[46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever, *Zero-shot text-to-image generation*, Proceedings of the 38th International Conference on Machine Learning (Marina Meila and Tong Zhang, eds.), Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 8821–8831.

[47] Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik-Manor, *Imagenet-21k pretraining for the masses*, CoRR **abs/2104.10972** (2021).

[48] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben Baruch, and Asaf Noy, *Ml-decoder: Scalable and versatile classification head*, CoRR **abs/2111.12933** (2021).

[49] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby, *Scaling vision with sparse mixture of experts*, CoRR **abs/2106.05974** (2021).

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, International Journal of Computer Vision (IJCV) **115** (2015), no. 3, 211–252.

[51] Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv 1409.1556 (2014).

[52] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani, *Bottleneck transformers for visual recognition*, CoRR **abs/2101.11605** (2021).

[53] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, *Rethinking the inception architecture for computer vision*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, *Training data-efficient image transformers & distillation through attention*, CoRR **abs/2012.12877** (2020).

[55] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou, *Going deeper with image transformers*, CoRR **abs/2103.17239** (2021).

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, CoRR **abs/1706.03762** (2017).

[57] James Vuckovic, Aristide Baratin, and Remi Tachet des Combes, *A mathematical theory of attention*, 2020.

[58] Feng Wang, Tao Kong, Rufeng Zhang, Huaping Liu, and Hang Li, *Self-supervised learning by estimating twin class distributions*, CoRR **abs/2110.07402** (2021).

[59] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao, *Pyramid vision transformer: A versatile backbone for dense prediction without convolutions*, CoRR **abs/2102.12122** (2021).

[60] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He, *Non-local neural networks*, CoRR **abs/1711.07971** (2017).

[61] Ross Wightman, *Pytorch image models*, https://github.com/rwightman/pytorch-image-models, 2019.

[62] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, *CBAM: convolutional block attention module*, CoRR **abs/1807.06521** (2018).

[63] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda, *Visual transformers: Token-based image representation and processing for computer vision*, CoRR **abs/2006.03677** (2020).

[64] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang, *Cvt: Introducing convolutions to vision transformers*, CoRR **abs/2103.15808** (2021).

[65] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo, *Segformer: Simple and efficient design for semantic segmentation with transformers*, CoRR **abs/2105.15203** (2021).

[66] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu, *End-to-end semi-supervised object detection with soft teacher*, CoRR **abs/2106.09018** (2021).

[67] Xue Ying, *An overview of overfitting and its solutions*, Journal of Physics: Conference Series **1168** (2019), 022022.

[68] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu, *Incorporating convolution designs into visual transformers*, CoRR **abs/2103.11816** (2021).

[69] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang, *Florence: A new foundation model for computer vision*, CoRR **abs/2111.11432** (2021).

[70] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer, *Scaling vision transformers*, CoRR **abs/2106.04560** (2021).

[71] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, *Dive into deep learning*, arXiv preprint arXiv:2106.11342 (2021).

[72] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander J. Smola, *Resnest: Split-attention networks*, CoRR **abs/2004.08955** (2020).

[73] Qing-Long Zhang and Yubin Yang, *Rest: An efficient transformer for visual recognition*, CoRR **abs/2105.13677** (2021).

[74] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong, *End-to-end object detection with adaptive clustering transformer*, CoRR **abs/2011.09315** (2020).

[75] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang, *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers*, CoRR **abs/2012.15840** (2020).

[76] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, *Scene parsing through ade20k dataset*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5122–5130.

[77] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng, *Deepvit: Towards deeper vision transformer*, CoRR **abs/2103.11886** (2021).

A ANNEXOS

A.1 Análisi del procediment de simplificació dels valors singulars

Sabem que $Q = XW_q$ i $K = XW_k$ són matrius reals i on W_Q i W_k són matrius reals quadrades. Volem mirar de simplificar la operació $softmax(QK^T) = softmax(XW_qW_k^TX^T)$, suposant que W_Q i W_k tenen molts valors singulars propers a 0.

Sabem que tota matriu té descomposició SVD[1], és a dir, tota matriu $A \in \mathbb{R}^{n\times m}$ es pot descomposar de la següent forma:

$$A = ULV^T$$

on V^T és la matriu transposada de V (en nombres complexes seria la matriu conjugada transposada, però en reals és la transposada), on V és una matriu de dimensió $m \times m$ que té per columnes els vectors singulars d'A per la dreta i U una matriu $n \times n$ composta pels vectors singulars d'A per l'esquerra. Finalment, L és una matriu rectangular diagonal de dimensió $n \times m$ que té per elements els valors singulars d'A (per definició positius o 0).

Considerem la següent notació per la matriu W_Q (idem per W_k):

$$W_q = U_qL_qV_q^T$$

on totes les matrius tenen dimensió $n \times n$.

Posant tota l'operació, tenim:

$$softmax(QK^T) = softmax(XU_qL_qV_q^TV_kL_kU_k^TX^T)$$

Observar que en la fórmula anterior hem utilitzat la igualtat $L_k^T = L_k$ ja que la matriu és diagonal.

Considerem que la matriu W_Q i W_k tenen q_1 i k_1 valors singulars diferents de 0 respectivament. Aleshores, considerem $m = max(q_1, k_1)$. Seguint amb la notació:

$$U_q = \begin{bmatrix} U_{q11} & U_{q12} \\ U_{q21} & U_{q22} \end{bmatrix},\ L_q = \begin{bmatrix} L_{q1} & 0 \\ 0 & L_{q2} \end{bmatrix},\ V_q^T = \begin{bmatrix} V_{q11}^T & V_{q21}^T \\ V_{q12}^T & V_{q22}^T \end{bmatrix}.$$

on les matrius L_{q1} U_{q11} i V_{q11} són matrius $m \times m$, U_{q21}, V_{q21}, U_{q12}^T, V_{q12}^T tenen dimensions $(n-m) \times m$, és a dir, $n-m$ files i m columnes, i les matrius L_{q2} U_{q22} i V_{q22} tenen dimensions $(n-m) \times (n-m)$. A més, les matrius L_{q1} i L_{q2} són matrius diagonals.

Seguint la nomenclatura anterior, tenim:

$$V_q^TV_k = \begin{bmatrix} V_{q11}^TV_{k11} + V_{q21}^TV_{k21} & V_{q11}^TV_{k12} + V_{q21}^TV_{k22} \\ V_{q12}^TV_{k11} + V_{q22}^TV_{k21} & V_{q12}^TV_{k12} + V_{q22}^TV_{k22} \end{bmatrix}.$$

Continuem multiplicant per L_q i L_k:

$$L_qV_q^TV_kL_k = \begin{bmatrix} L_{q1}(V_{q11}^TV_{k11} + V_{q21}^TV_{k21})L_{k1} & 0 \\ 0 & 0 \end{bmatrix}$$

Finalment, multipliquem per U_q i U_k^T:

$$U_q L_q V_q^T V_k L_k U_k^T = \begin{bmatrix} U_{q11} B U_{k11}^T & U_{q11} B U_{k21}^T \\ U_{q21} B U_{k11}^T & U_{q21} B U_{k21}^T \end{bmatrix}$$

on $B = L_{q1}(V_{q11}^T V_{k11} + V_{q21}^T V_{k21}) L_{k1}$.

Observació: Podem destacar que no podem suposar $A = VLV^{-}1$ ja que aquesta descomposició no és possible per totes les matrius reals (sí és possible si treballem en complexos). Per tant, la descomposició que es necessària per aquest treball és la que ens dona la descomposició SVD.

Cost Computacional Respecte la complexitat de la descomposició SVD, és pot veure que realitzar la descomposició SVD de dues matrius quadrades de dimensió n té un cost de $O(n^3)$, cost superior al de multiplicar aquestes dues matrius, el qual, utilitzant l'algorisme de Williams i Alman [2], té un cost de $O(n^{2.3728596})$. Per tant, es pot fàcilment demostrar que el cost computacional per les diferents matrius és:

1. Descomposar SVD: $O(n^3)$
2. Obtenir B: $O(4 * n^{2.3728596} + n^2)$
3. Obtenir $W_q W_k = U_q L_q V_q^T V_k L_k U_k^T$: $O(6 * n^{2.3728596})$