En aquesta part pràctica, us explicaré els passos que vaig seguir per aprendre a fer les xarxes neuronals des de 0 i les dificultats amb les que em vaig trobar. Al final del treball, a l’annex, podreu trobar una còpia del codi de la xarxa neuronal.

Els principals objectius d’aquest treball de recerca eren crear una xarxa neuronal i, si fos possible, crear-la sense cap ajuda de llibreries. Per aconseguir aquests objectius, la millor manera de fer-ho seria buscar una idea senzilla per la xarxa neuronal, ja que si la idea fos massa complexa, seria molt difícil aprendre tota la teoria de xarxes neuronals i de programació per aconseguir una cosa tant complicada.

Una de les hipòtesis a demostrar, era que un alumne de segon de batxillerat pot crear una xarxa neuronal des de zero; i això també ho discutirem aquí.

Al principi del treball, tenia una metodologia molt clara: aprendre gràcies a les plataformes de YouTube i Coursera i fer la xarxa neuronal amb el llenguatge de programació JavaScript.

La plataforma Coursera, és una pàgina web de cursos online. En aquesta pàgina, universitats de tot el món publiquen els seus cursos sobre el tema que vulguin. Per publicar i monetitzar els cursos, les universitats ho poden fer de tres maneres: publicar un curs completament gratuït, publicar un curs que només puguis veure o publicar el curs i que només hi pugui accedir la gent que el compra. Al acabar aquests cursos, l’estudiant rep un diploma de la universitat conforme ha fet el curs. En el curs gratuït, no has de pagar per rebre el diploma; en el curs en que només pots veure les classes, no pots fer exàmens ni rebre el diploma si no pagues; i en el curs en el que pagues per accedir-hi, reps el diploma al acabar-lo.

**Coneixements previs**

Abans de començar a explicar la metodologia de la creació de la xarxa neuronal, cal destacar que no partia d’uns coneixements nuls en quant a programació. A mitjans de 4rt d’ESO (fa 2 anys), vaig descobrir els llenguatges de programació, i el meu interès en ells van fer que comencés a aprendre de forma autodidacta. Durant tot un any, vaig estar navegant internet en busca d’exemples i tutorials per aprendre tant com pogués dels llenguatges de programació.

El primer llenguatge de tots que vaig descobrir va ser un anomenat “Processing”. Aquest llenguatge era molt visual i fàcil d’entendre. Però al ser un llenguatge de programació seguia uns bàsics que segueixen tots els llenguatges. Per tant ja tenia una bona base per enfrontar-me a altres llenguatges més sofisticats. El següent llenguatge que més em va interessar, va ser C++, un llenguatge molt utilitzat avui en dia. No vaig aprofundir molt, però si que vaig aprendre suficient com per utilitzar els coneixements més endavant.

Per últim, em vaig interessar en JavaScript. Vaig buscar i fer diversos cursos on-line d’aquest llenguatge. Vaig aconseguir un bon nivell.

En resum, al començar el treball de recerca, tenia una base sòlida de llenguatges de programació i havia aprofundit en C++ i JavaScript; però de xarxes neuronals, no n’havia sentit a parlar, encara.

**Primer intent autodidacta**

Per començar el treball de recerca, tant la part teòrica com la part pràctica, havia de saber que eren les xarxes neuronals i com funcionaven. Per començar a aprendre, vaig decidir buscar cursos on-line, ja que en podia trobar que estiguessin bé i fossin gratuïts. Finalment, vaig començar un curs sobre xarxes neuronals a una pàgina anomenada “Coursera”.

El curs estava pensat per fer durant onze setmanes, dedicant-se tres hores a la setmana. Les primeres setmanes del curs eren introducció, i parlaven d’on venien les xarxes neuronals i hi havia classes d’àlgebra bàsica i matrius per qui ho necessités. D’aquestes primeres setmanes, vaig acabar fent només les classes de matrius, que eren tres vídeos de 20 min cada un. Cal destacar que un concepte de les matrius estava explicat de manera errònia, cosa que va aportar confusió més endavant.

La resta de setmanes del curs, les vaig fer per sobre, ja que vaig veure que aquest curs no anava enlloc. Vaig acabar aprenent que era *feedforward* i en total, vaig extreure dotze pàgines d’apunts a mà amb molts diagrames i dibuixos.

**Primeres idees**

Al “acabar” l’anterior curs, ja sabia què eren les xarxes neuronals, tot i que tenia poques idees de que podien fer. Tot i això, era moment de triar una idea per la meva futura xarxa neuronal. Directament, vaig descartar qualsevol idea que tingués a veure amb reconeixement d’imatges o de veu, ja que era conscient de que eren coses de nivell avançat i requerien de màquines potents per treballar, cosa de la que no disposava. Idees de nivell avançat les vaig descartar principalment per temps: tenia un estiu per aprendre la programació i la teoria necessària, i sabia que no era possible. I exemples que requerissin de màquines potents els vaig descartar, ja que treballava des del meu portàtil, que era poc potent, i fer projectes ambiciosos en ell significava emprenyar-se per aplicacions que no responen o per temps de càrrega massa grans.

Al final vaig tenir la primera idea: fer una xarxa neuronal a la que li diguessis un parell de pel·lícules que t’agraden, i a través de factors comuns (director, durada, tema...), et proposés una o dues que et poguessin agradar.

Encara no sabia com es podien programar les xarxes neuronals artificials, però sí que sabia que, per començar el projecte, necessitaria una base de dades gran amb pel·lícules i les seves característiques principals. Com que omplir bases de dades és complicat i tediós, vaig dissenyar i crear una pàgina web amb un petit qüestionari que cada persona a qui li compartís ompliria amb la seva pel·lícula preferida.

Començant a investigar com podia fer aquest programa, vaig descobrir que no ho podria fer amb una xarxa neuronal artificial, sinó que es feia amb un altre tipus de programa. En aquest punt, vaig abandonar la idea.

**Primers intents de programació**

En aquell moment, m’havia quedat sense idea, per tant, vaig començar a fer la part teòrica. Avançant, va arribar un moment en el que vaig acabar l’apartat del perceptró, i vaig pensar que seria una bona idea fer-ne un; tot i que no contava completament com a xarxa neuronal. Com a exemple, vaig utilitzar el mateix que a l’apartat abans mencionat: un perceptró que resolgués portes lògiques. Com que la intenció era fer un programa senzill i no perdre molt temps creant-lo, vaig decidir fer-ho en C++, sent la primera raó el fe de que feia molt que no l’utilitzava i era un dels meus preferits. També va ser perquè era un llenguatge eficient i no necessitava cap aspecte visual per fer el perceptró.

La creació del programa del perceptró no va comportar molts problemes, ja que era un programa senzill i vaig poder evitar treballar amb matrius gràcies a la poca quantitat de dades que havia de processar. A més a més, ja que tots els programes de programació tenen un sistema per operar amb portes lògiques, vaig poder automatitzar el procés d’aprenentatge del perceptró amb poques línies de codi més.

Tot i que el programa, teòricament, havia de funcionar, no ho feia; i vaig estar dies canviant trossos de programa i revisant una i altre vegada per detectar algun error, però no vaig poder descobrir perquè el perceptró no funcionava, ja que tornava resultats erronis desprès d’haver estat entrenat.

**La segona idea**

En quan em vaig informar més sobre les xarxes neuronals més senzilles vaig veure que la gran majoria classificaven. Això em va ser de gran ajuda, ja que ara podia tornar a pensar sobre la idea de la meva xarxa neuronal senzilla, ja que hi havia moltes coses que es podien classificar. Tenia clar que volia que fos molt visual i que el que fos que classifiqués la xarxa, havia de dependre de la percepció humana, ja que així m’assegurava que el que feia la xarxa no ho podria fer qualsevol altre programa. Pensant en aquesta idea, vaig acabar pensant en colors, ja que es podien classificar de moltes maneres.

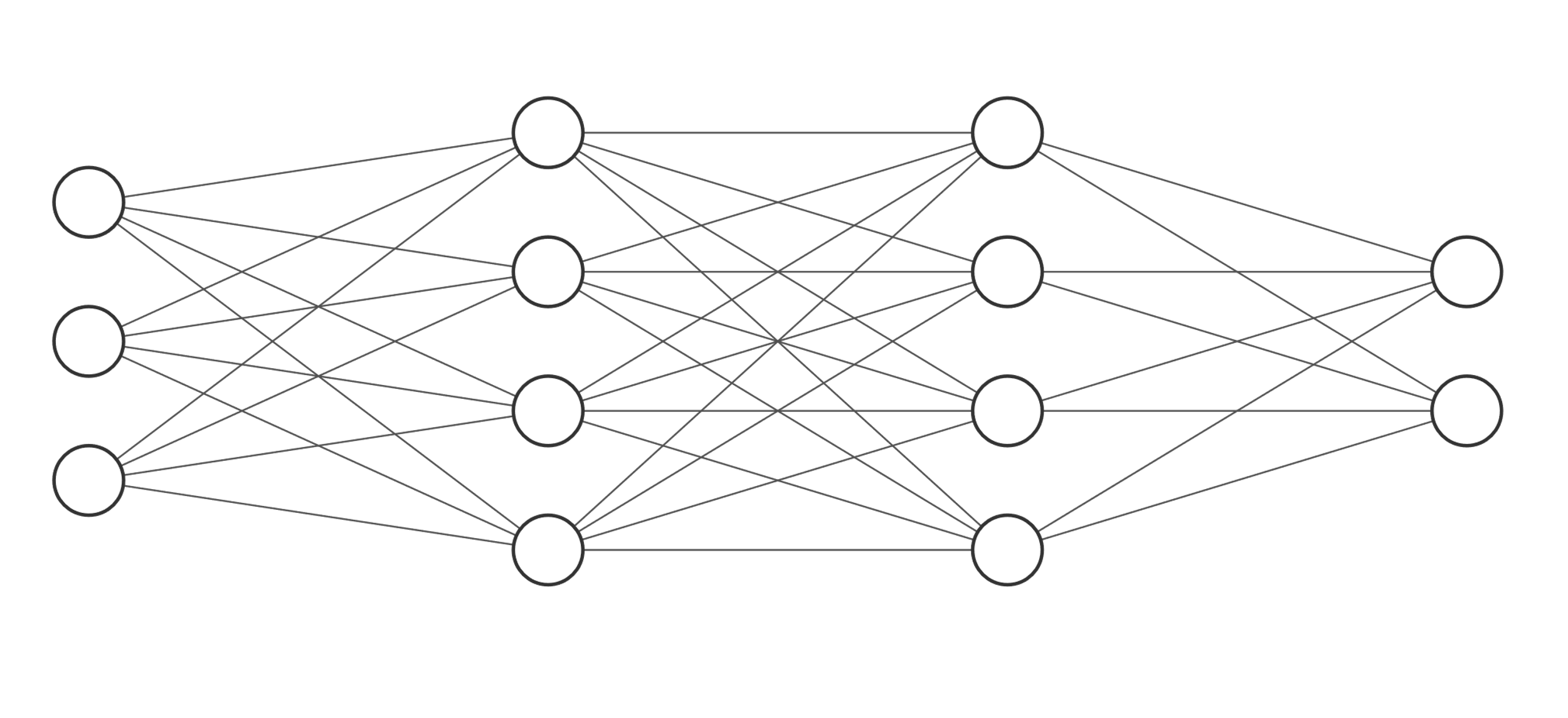
Finament, vaig arribar a la meva idea final: la xarxa neuronal classificaria els colors segons si eren clars o foscos.

**Desenvolupament de la segona idea**

Un cop ja tenia la idea triada, havia de passar al desenvolupament de la xarxa neuronal. A l’hora de crear una xarxa neuronal, cal tenir dos factors en compte: la seva estructura i la seva presentació.

Per la estructura de la xarxa, havia de triar el nombre de columnes i el nombre de neurones de cada columna. Per la columna d’entrada, triar el nombre de neurones era fàcil: tres; ja que la xarxa llegiria el color en rgb, per tant, rebria tres dades diferents. I per triar les neurones de sortida tampoc era molt difícil: dues; una per indicar el percentatge de probabilitat de que el color fos clar i l’altre per fosc.

El fet de triar el nombre de columnes i neurones de la capa oculta ja era més difícil. Els factors a tenir en compte a l’hora de triar és que quantes més columnes, més complicat seria crear-la i més temps tardaria, però els resultats serien molt més precisos; mentre que si posava poques columnes la xarxa seria més imprecisa però aniria més ràpid i seria més fàcil de fer. Al final em vaig quedar en un punt intermedi: dues columnes de quatre neurones cada una.



**Plantejament de la programació**

Ara que ja tenia la idea i que estava segur de que es podia dur a terme amb una xarxa neuronal, m’havia de plantejar tots els aspectes de com ho faria: des de com funcionaria la xarxa neuronal a com interactuaria l’usuari amb ella. Per començar per algun lloc, vaig decidir començar per la part tècnica abans que la visual.

Com hem vist a la teoria, la xarxa havia de constar de dues parts o processos, el *feedforward*, en el que donava la resposta que creia; i el *backpropagtion*, a on es corregia gracies a l’usuari si era necessari.

***Feedforward***

El primer pas que havia de prendre per començar a fer la xarxa neuronal era pensar en com fer l’algoritme amb el que la xarxa tornaria una resposta. Per fer això, la xarxa necessitava unes senyals d’entrada, i com hem quedat abans, aquestes serien un color en rgb. En aquest punt, ja sabia per on començar: havia de fer un sistema que em generés un color de manera aleatòria i que l’usuari pogués interactuar per activar la xarxa neuronal.

Per fer això vaig decidir crear un requadre que seria del color que es triés aleatòriament, i un botó que, cada cop que fos clicat, canviés el color. Fer això no va ser complicat, ja que qualsevol llenguatge de programació té una funció que permet escollir un número aleatori entre dos nombres determinats; i per pintar un requadre d’un cert color, no era més complicat que dues línies de codi.

En el moment en que vaig acabar de fer el sistema que escollia els colors, ja podia començar a programar la xarxa neuronal, ja que la senyal d’entrada era tot el que de moment necessitava.

Com hem vist a la part teòrica, el que fa la xarxa neuronal per donar una resposta és fer varies multiplicacions de matrius entre les dades d’entrada i els pesos; el que significava que necessitaria una manera d’operar amb matrius. Tots els llenguatges de programació tenen unes matemàtiques bàsiques incorporades, com podira ser: suma, resta, multiplicació, divisió, funcions trigonomètriques... però no n’hi ha cap que tingui incorporat un sistema d’operacions amb matrius.

Per aconseguir-ho, normalment faria ús d’una llibreria externa, una sèrie de funcions que algú va programar que em permetrien operar amb matrius. Utilitzar això tenia dos majors inconvenients: el primer es que estaria directament negant la meva hipòtesis, que afirmava poder crear la xarxa neuronal artificial sense ajuda de cap llibreria, i el segon inconvenient era que hauria d’aprendre les funcions de la llibreria, saber com utilitzar-les i haver de tractar amb errors per causa del funcionament de la llibreria. És per això que vaig decidir crear el meu propi codi que em permetria crear i operar amb matrius. En aquell moment no podia programar-ho sense ajuda, ja que no tenia els coneixements necessaris; així que vaig decidir seguir un tutorial de YouTube, fet per Daniel Shiffman.

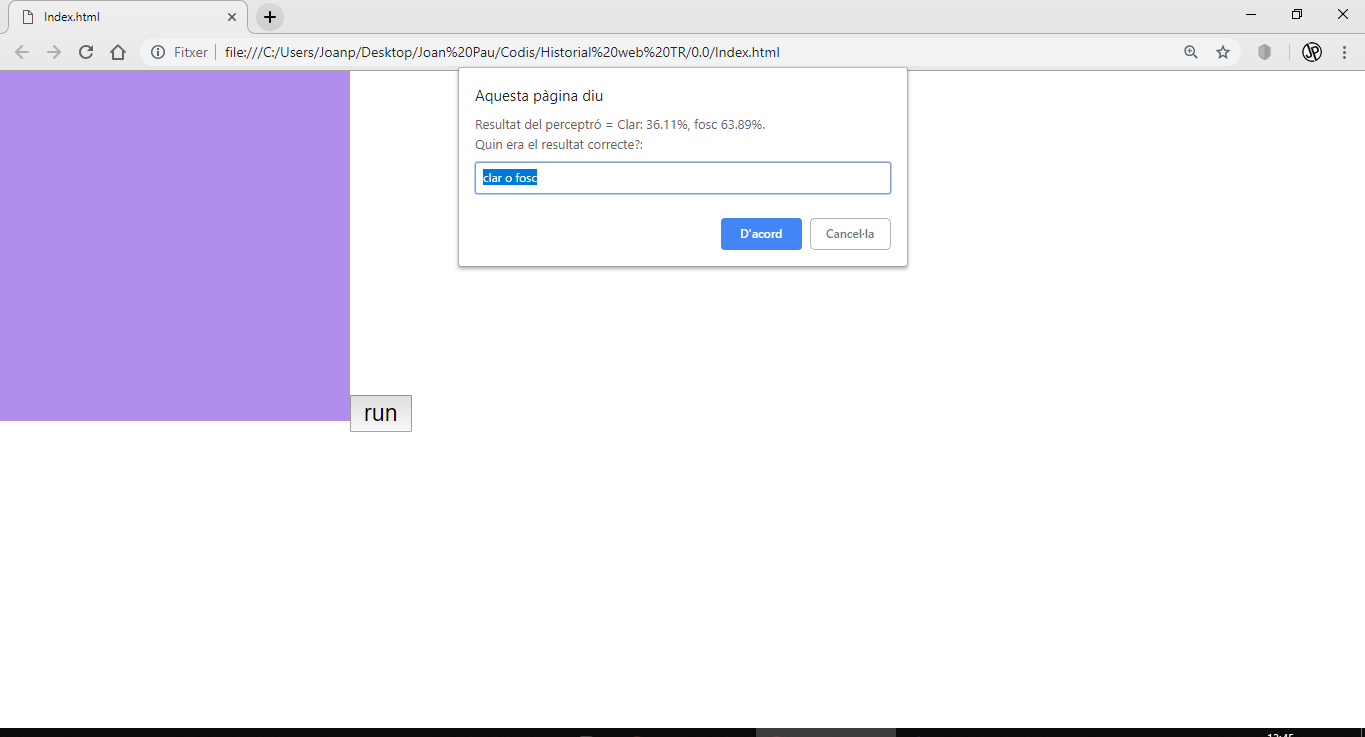
Al acabar aquest tutorial tenia un programa que e permetia operar i crear matrius i que sabia com funcionava i exactament quines funcions tenia i com funcionava. Aquesta petita avantatge em va ser molt útil ja que em va fer més fàcil la detecció d’errors en el futur i la implementació de noves funcions en cas de necessitat.

En aquest moment ja vaig començar a programar la xarxa neuronal amb el seu primer algoritme: el *feedforward.* En acabar, ja tenia un programa que agafava un color aleatori en rgb i em deia si era clar o si era fosc. Una cosa a tenir en compte del programa en aquest punt és que els pesos, que són l’element que fan que la xarxa funcioni millor o pitjor, es triaven també de manera aleatoria cada cop que l’usuari entrava a la web de la xarxa neuronal. En aquest punt no era un problema, però més endavant sí que ho serà.

***Backpropagation***

Ara que a tenia el primer dels dos algoritmes de la xarxa, podia continuar amb el següent, el que li permetria a la xarxa aprendre i funcionar. Com que estava treballant amb l’aprenentatge supervisat, el primer que havia de fer era preguntar a l’usuari quina era la resposta correcta. Per fer això, feia que quan la xarxa ja tingués la resposta, apareixes una notificació que preguntés per una resposta a l’usuari.

En quan vaig tenir allò fet, vaig donar per feta la primera versió de la xarxa, i vaig decidir aparcar el tema de l’algoritme de *backpropagation* ja que era molt complicat i necessitava informar-me més. Mentrestant, aniria retocant i afegint aspectes visuals per millorar l’experiència de l’usuari a la web de la xarxa. En aquest punt, la web tenia la forma següent:



**Els pesos**

Com s’ha mencionat abans, cada cop que l’usuari entrava a la pàgina web, els pesos es generaven de manera aleatòria. Si volia continuar amb el desenvolupament de la xarxa, havia de canviar això, ja que sinó, la xarxa mai podria aprendre i fer l’algoritme que entrenava la xarxa seria inútil ja que tot l’entrenament fet es perdria. Això significava que, abans de continuar fent la xarxa, havia de trobar una solució a aquest problema i aplicar-la.

La solució era relativament senzilla: cada cop que l’usuari entrenés la xarxa, els pesos canviarien de valor, i en aquell moment hauria de copiar el valor dels pesos en un arxiu per després, el següent cop extreure les dades del fitxer i donar aquells valors als pesos. D’aquesta manera, els pesos es guardarien cada cop que es tornés a iniciar la pàgina web.

Per guardar els valors dels pesos, es pot fer de dues maneres: localment o al núvol. La avantatge de fer-ho localment, és que no requereix d’accés a internet i el fitxer es guardaria en un lloc segur: el disc dur del meu ordinador. La principal desavantatge de fer-ho d’aquesta manera és que la pàgina web no podria ser publicada, ja que després, per agafar els valors dels pesos des de qualsevol ordinador que no fos el meu, no ho podria fer perquè el fitxer no existiria.

És per això que vaig decidir fer-ho a núvol. D’aquesta manera, els valors estarien en una base de dades o en un fitxer al núvol i podrien ser accedits per qualsevol persona que entrés a la web des de un dispositiu que no fos el meu ordinador. Per guardar els pesos en una base de dades a núvol, vaig utilitzar la aplicació de *firebase*.

***Firebase***

*Firebase* és una aplicació gratuïta de Google que et permet crear una base de dades al núvol i escriure i llegir dades en ella. Aquesta base de dades està preparada per poder ser utilitzada des de aplicacions de mòbil, pàgines web, videojocs, i moltes plataformes més.

La meva intenció amb aquesta plataforma era que en el moment en que l’usuari es connectés a la pàgina web, la xarxa copiés les dades de la base de dades per actualitzar el valor dels pesos i que cada cop que fos entrenada i els pesos canviessin de valor, s’enviessin els nous valors a la base de dades.