

## **Primera reunión (23-02-22):**

bases de datos: mnist, fashion mnist

redes densa y convolucionales, binaria (parte positiva: trabajan con entrada y salida binaria, hay que modificar las operaciones internas) (modificarlas), prioridad nn normales (si da tiempo convolucional)

entender de que va el tema

revisión histórica, estudiar algoritmos, e implementar, 3 algoritmos a comparar

para la memoria:

diagrama de clases, analisis de riesgos, presupuestos,

metodo agil o clasico (puedo elegir ambas,justificarlo)

ref: revistas, libros, pagina web (directamente tiene que ir separada, NUNCA WIKIPEDIA NI BLOGS).

justificar todo los métodos/clases/software

leer papers (que manda Rocío) y escoger metodología

llevar control de horas de trabajo (timecamp)

hacer diagrama de gantt y contrastarlo al final (ver tutorial de como hacer estos diagramas)

análisis de riesgo (se tiene que hacer al inicio, aunque se suele hacer al final) mirar tfgs

para las operaciones de la red neuronal mejor usar op simples, sumas, restas, etc...

## **Primera semana:**

### **To-Do List:**

Entender diagramas de gantt y cómo hacerlo

Estudiar Papers que envíe Rocío (esperando)

Estudiar metodologías de desarrollo de software (clásica o ágil)

Ver como hacer análisis de riesgo, mirar los tfgs antiguos.

### **Como hacer diagrama de gantt**

Listar todas las tareas a realizar, inicio, fin y duración, completarla lo máximo posible (pensar las partes de las que se va a componer el TFG)

Crear un gráfico de barras

Añadir fechas y duraciones al gráfico de barras (consultar [web](#))

### **Tareas a realizar en un futuro**

- Resumen del TFG (español e inglés)
- Redactar la introducción (hablar sobre la computación neuromórfica, sus motivaciones y beneficios y sobre los estudios que voy a realizar)
- Justificación del proyecto (por qué lo hago, ¿qué utilidad tiene?)
- INCLUIR ESTRUCTURA QUE VA A SEGUIR LA MEMORIA
- Definir los objetivos en la memoria (marcar cada uno de los objetivos que tengo que alcanzar para completar el TFG)
- Por cada objetivo definir la lista de tareas necesarias para su consecución
- Definir la metodología que voy a llevar a cabo para desarrollar el proyecto.
- Realizar diagrama de gantt

### **Metodología ágil:**

- Método iterativo y flexible
- Se compone de fases llamadas sprints, en cada sprint se termina una pieza del proyecto completa y funcional
- Mayor control y flexibilidad para afrontar cambios y riesgos
- Se pueden hacer cambios en la planificación y en los requisitos de forma rápida y controlada
- Mayor contacto con el cliente
- Mejor software que funcione que mucha documentación
- Interacción entre los individuos muy importante
- Comunicación con el cliente es vital
- Establecer un plan es necesario pero es más importante tener buena capacidad de respuesta
- Product Backlog:** requisitos del producto a desarrollar, también se estima el tiempo necesario para realizar dichos requisitos. Evoluciona a lo largo del todo el desarrollo.
- Scrum Master:** líder del equipo, se encarga de hacer cumplir la metodología
- Product owner:** Encargado de gestionar el desarrollo, organiza y prioriza las funcionalidades
- Equipo de desarrollo:** se puede subdividir en equipos

### **-Ciclo de vida de un sprint:**

- Planificación: se define el plan a seguir. Qué se va a hacer y cuanto se va a tardar. Con esto y el backlog se crea el sprint backlog
- Reunión diaria: seguimiento del trabajo y se planifica lo que se hará mañana
- Revisión de la iteración: Tras terminar el sprint se hace una revisión, viendo que objetivos se alcanzaron y que problemas han surgido
- Retrospectiva de la iteración: se revisan los puntos a mejorar y los puntos fuertes del equipo para ser más productivos.
- Replanificación: Junto al cliente se revisa la planificación teniendo en cuenta posibles cambios surgidos durante el sprint.

**Metodología waterfall:**

- definición
- diseño
- codificación
- de pruebas
- mejora

**Dudas:**

-Al no ser un producto software tal cual se puede aplicar la metodología scrum???, pues hay que investigar mucho y cuando se haya hecho la investigación me pondré con la implementación con tensorflow y keras.

**Segunda reunión (2-03-22):****Esqueleto del TFG**

- Buscar algoritmos de redes neuronales (bibliografía)
- Calcular el orden (importante para el orden)
- Elegir un par para hacer pruebas (no más de 3) (sencillo, eficiente, complejo)
- Elegir un ganador e implementarlo (buscarlo o implementarlo)
- Pruebas para testear (si hay tiempo añadir ruido de los componentes)
- Análisis de los resultados
- Redactar motivos por los que he seleccionado la metodología scrum. resultados

**Recomendaciones:**

algoritmo original, pero puedo proponer una modificación

**To-Do List:**

- Elegir plantilla para memoria (Hecho)
- Buscar algoritmos de entrenamiento de redes neuronales (Done)
- Hacer Diagrama de Gantt (Done)
- Análisis de riesgos (Pendiente -> se traspasa a la siguiente semana)
- Justificar el por qué voy a usar metodología scrum (Pendiente -> se traspasa a la siguiente semana)
- Establecer los sprint a realizar (Pendiente -> se traspasa a la siguiente semana)
- Establecer objetivos y tareas para conseguirlos (Done)

**Objetivos del TFG:**

**Objetivo general:** Encontrar el mejor algoritmo para entrenar redes neuronales cuantizadas en circuitos neuromórficos.

- OBJ1. Recopilar algoritmos de entrenamiento de redes neuronales.

- OBJ2. Encontrar la complejidad computacional de cada uno de los algoritmos de entrenamiento estudiados.
- OBJ3. Comparar 3 de los algoritmos estudiados
- OBJ4. Adaptar el mejor de los algoritmos para que se ejecute en circuitos neuromórficos.
- OBJ5. Estudiar el comportamiento del mejor algoritmo en los circuitos neuromórficos.

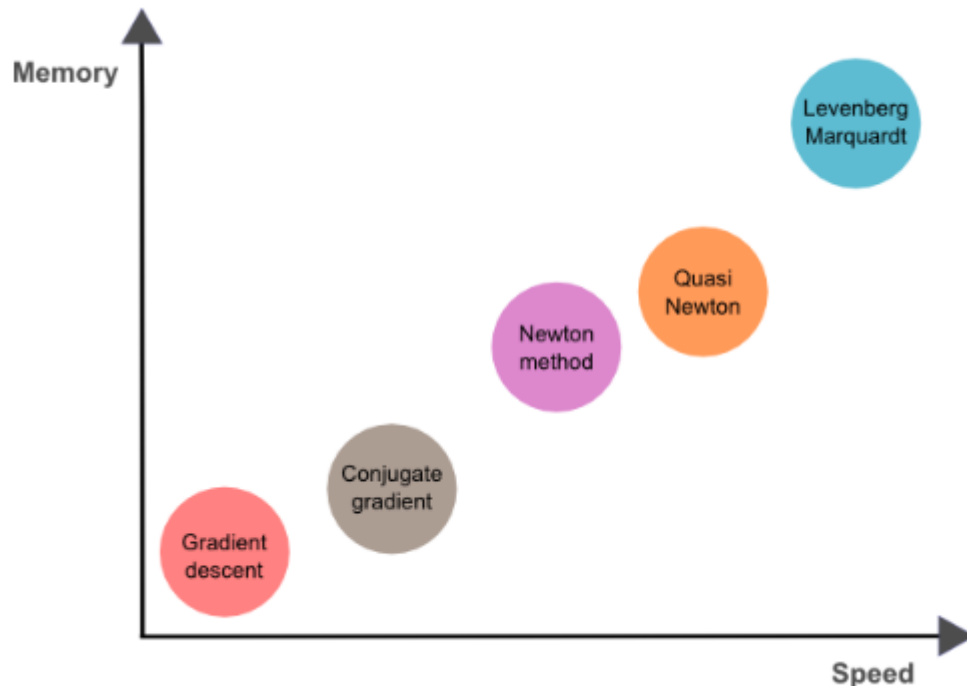
#### **Tareas del TFG:**

1. (OBJ1) Buscar los algoritmos existentes para el entrenamiento de redes neuronales cuantizadas
  - a. Buscar en internet o en libros los algoritmos de entrenamiento de redes neuronales.
  - b. Comprender el funcionamiento de los algoritmos y redactar una explicación de cada uno de ellos.
2. (OBJ2) Realizar una análisis de la complejidad computacional de cada uno de los algoritmos encontrados.
3. (OBJ3) Comparar el rendimiento de 3 de los algoritmos más interesantes que haya recopilado.
  - a. Implementar los 3 algoritmos.
  - b. Ejecutar y tomar medidas de las ejecuciones de los algoritmos.
  - c. Realiza el análisis de los resultados.
4. (OBJ4) Implementar el mejor algoritmo en una red neuronal cuantizada.
5. (OBJ5) Análisis del mejor algoritmo.
  - a. Ejecutarlo varias veces en las bases de datos
  - b. Tomar medidas
  - c. Estudiar las medidas y sacar conclusiones.

#### **Lista de algoritmos de entrenamiento (MODIFICACIONES DEL BACKPROP):**

- Gradient descent (stochastic, mini-batch, momentum, nesterov accelerated gradient, adagrad, adadelta, adam): se trata de un algoritmo que ajusta los pesos de la red retropropagando los errores cometidos, para ello hace uso de la derivada del error y de la regla de la cadena).
- Conjugate Gradient
- newton method (**puede llegar a diverger, si se encuentra en un maximo se para**): este método aproxima localmente la función que esta optimizando haciendo uso del polinomio de taylor de segundo orden, la raíz de dicho polinomio se usa para calcular el tamaño y la dirección en la que calcular el paso. Usa la matriz hessiana por lo tanto es un algoritmo de entrenamiento de segundo orden.

- quasi newton (**puede llegar a diverger????, si se encuentra en un maximo se para**): realiza lo mismo que el método de newton pero no usa la matriz hessiana debido al coste computacional. Para ello aproxima la matriz hessiana mediante un proceso iterativo.
- levenberg marquardt



Hasta ahora la información la he sacado de

<https://www.datasciencecentral.com/5-algorithms-to-train-a-neural-network/>

### Alternativas a backprop:

- Hebbian learning (aprendizaje no supervisado)
- Delta rule (aprendizaje supervisado) (es para perceptron)
- difference target propagation <https://arxiv.org/pdf/1412.7525.pdf> (pendiente de estudio)
- dtp mejora <https://arxiv.org/pdf/2006.14331.pdf> (pendiente de estudio)
- hsic bottleneck <https://arxiv.org/pdf/1908.01580v3.pdf> : hace uso de la teoría de la información para el entrenamiento de redes neuronales. Se basa en la correlación entre variables. La idea es hacer que la entrada no esté correlada con la representación interna, y que la representación interna esté altamente correlada con la salida.  
explicación  
<https://towardsdatascience.com/hsic-bottleneck-an-alternative-to-back-propagation-36e951d4582c>
- synthetic gradient:

<https://deepmind.com/blog/article/decoupled-neural-networks-using-synthetic-gradients> : alternativa novedosa al backpropagation que hace uso de estructuras auxiliares llamadas módulos para el entrenamiento, son redes muy pequeñas de 1 o 2 capas de profundidad. Estos módulos se encargan de proporcionar a las capas gradientes artificiales los cuales son estimaciones hechas por los módulos, a su vez, estos módulos se irán entrenando con el gradiente real para poder ir mejorando las estimaciones.

-feedback alignment: <https://www.nature.com/articles/ncomms13276.pdf> : se trata de emplear backpropagation pero sin propagar el error con los mismos pesos que en el feedforward. Este algoritmo hace uso de matrices aleatorias para el backpropagation consiguiendo buenos resultados. El motivo de tan buenos resultados es que la dirección en la que apunta el gradiente con los pesos reales y los aleatorios no difieren en más de 90°, además dicha diferencia se estrecha conforme avanza el entrenamiento dándose lo que se conoce como alineamiento.

### **Tercera reunión (09-03-22):**

descripción corta algoritmos

descripción más extensa de los algoritmos a implementar

#### **to do list:**

- Redactar introducción (done)
- Redactar tareas y objetivos (done) (pendiente de revisión)
- Aprender cómo hacer las referencias a la bibliografía. (done)
- Redactar algoritmos de entrenamiento (corto los más complejos, más extensos los interesantes)
- Justificar el por qué voy a usar metodología scrum
- Establecer los sprint a realizar (establecer tras la reunión del martes)
- Análisis de riesgos
- BUSCAR DELTA RULE
- Para la explicación del backpropagation explicar los 3 algoritmos que lo componen: gradient descent (explicar en el preliminar), cálculo de sensibilidad y (done)

#### **preguntas a realizar para la cuarta reunión:**

- ¿Qué debo de poner en la especificación de los requisitos?
- ¿Debo de seguir la estructura que establece la plantilla del tfg? (me parece que tiene más sentido la estructura llevada a cabo por los tfg que envió)
- Estructura de plantilla tfg e indicaciones en la web del tfg no coinciden, ¿cual debo seguir? Especificar qué tengo que tratar en cada uno de los apartados.
- ¿Dónde colocar la revisión del estado del arte? ¿Que comentario?
- Estoy yendo a buen ritmo o tengo que trabajar más

- No puedo acceder a la publicación de hsic bottleneck (pedírselo a Rocío)
- No encuentro el pseudocódigo del synthetic gradient ¿qué hago?
- No encuentro el pseudocódigo del feedback alignment

#### **aprendido durante esta semana:**

- No dejar la memoria para redactar tras trabajar, hay que ir redactándola conforme se avanza.
- Dividir el tiempo entre las distintas partes del tfg, no dedicarse todo el rato a una cosa
- Hacer la revisión del estado del arte en la introducción
- Solo leer la introducción/abstract sobre el tema que quiero hablar
- Cuando diga algo justificarlo buscando un abstract que trate sobre dicho tema

#### **to do list introducción:**

- Decir por qué es importante el tema (mejor si hago referencias a papers sobre el tema)
- ventajas, inconvenientes y que voy a hacer para mi tfg
- estructura de la memoria: es mejor si lo redacto en un párrafo en vez de hacer una lista
- capítulo preliminar con las herramientas comunes a todos los algoritmos

#### **Problemas backprop:**

- Feedforward and backward block

## **cuarta reunión (15-03-22)**

- juntar estado del arte y preliminares

estructura

capítulo 1: íntro, justificación, objetivos y estructura de proyecto

capítulo 2: estado del arte, preliminares y algoritmos

capítulo 3: especificación requisitos y planificación y hacer historias de usuario (preguntarán en la defensa): para cumplir objetivo tengo que pensar en los requisitos, explico la planificación en función del estado del arte,

implementación

pruebas

planificación final (comparar con la inicial)

enlace hsic Wan-Duo Kurt Ma, JP Lewis, and W Bastiaan Kleijn. The hsic bottleneck: Deep learning without back-propagation.

In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 5085–5092, 2020

BUSCAR SYNTHETIC GRADIENT (PONER EL DE LA CONFERENCIA)

## **quinta semana (29-03-22): analisis de complejidad**

### **objetivos:**

- tener el pseudocódigo de al menos 1 de los 2 algoritmos restantes

### **to do list:**

- terminar de explicar los algoritmos restantes (done)
- terminar pseudocodigo (done)
- analizar complejidad algorítmica de los algoritmos (done) (falta synthetic)
- realizar recopilatorio acerca del trabajo realizado para luego poder recordarlo bien
- hacer panel con las tareas pendientes

### **problemas:**

- no hay pseudocódigo ni de synthetic gradient ni de feedback alignment
- fa y sa no dicen que eficiencia computacional tiene
- codigo de hsic muy mal explicado
- hsic explica una complejidad algorítmica que luego no implementa y no explica de donde aparece

### **soluciones:**

- mirar código de los repositorios de hsic
- sacar pseudocódigos restantes a partir de los papers, hacerlo de manera aproximada
- añadir enlace <https://www.sciencedirect.com/science/article/pii/S0893608021002732>

### **preguntas a realizar**

- es la complejidad algorítmica del calculo del primer gradiente  $O(n)???$   
BACKPROPAGATION
- preguntar si es correcta la forma en la que he escrito el codigo del synthetic gradient y si esta bien calculada la complejidad.
- cuando explique un problema tengo que encontrar un paper que describa solo el problema, ejemplo: vanishing gradients
- para los problemas del backprop he mirado este [enlace](#). lo puedo incluir? o quedaría mal? tengo que buscar los problemas en papers y referenciar dichos papers.
- no puedo acceder a <https://www.worldscientific.com/doi/abs/10.1142/S0129065791000261>

### **aprendido durante esta semana**



-0.1307 y 0.3081 es la media y desviación típica de la base de datos mnist

## **sexta reunion (06-04-22)**

hemos hablado sobre cuantización, vamos a realizar una aproximación, los cálculos se realizan de manera exacta pero los tensores se almacenan de manera cuantizada.

### **To do list:**

- sacar código cuantizado
- hacer análisis comparativo
- redactar sprints

### **Problemas:**

-si modifica el forward method se modifica el computational graph y por lo tanto se modifica el grad\_fn. resultado: no se realiza bien los cálculos hacia atrás.

añadir: <https://discuss.pytorch.org/t/torch-round-gradient/28628/5> (de donde saqué la idea de crear una función especial cuyo gradiente sea la identidad)

añadir: <https://pytorch.org/blog/quantization-in-practice/> (consultado para aprender teoría de cuantización)

### **Dudas:**

- ¿Qué función de cuantización debo usar?
- ¿Si cuantizo dos veces un número se queda igual que en la primera cuantización? Hacer comparación
- Que es el scale factor en la operación de cuantización, como lo obtengo?

## **septima reunion (20-04-22)**

### **To Do List:**

-hacer pruebas con valores maximos y minimos globales y locales

1. maximo y minimo global y local. global: entreno y miro los pesos maximos y minimos. local: lo mismo pero por capa, cuantizaciones distintas. Hacer prueba con los dos algoritmos

-BUSCAR:

atlas de activacion: visualización de que mira cada algoritmo. o INTERPRETABILIDAD DE MODELOS.

- Revisar dni y hsic: ofrecen malos resultados
- modularizar bases de datos

### **Dudas:**

- ¿Se puede hacer la cuantización y luego descuantizar para usar los valores en flotante?
- ¿Qué valores uso para la cuantización? Los pesos con los del entrenamiento del modelo sin cuantizar, pero ¿y los tensores?(actualmente estoy haciendo el producto y cuantizo en función del resultado) ¿y los gradientes? (sigo el mismo proceso)
- El entrenamiento que realizamos es fake ya que los rangos los estamos conociendo de un entrenamiento que hemos hecho sin cuantización. ¿Cómo podemos hacer este entrenamiento sin requerir un entrenamiento previo?
- Usamos cuantizaciones distintas, pero si lo implementamos en un modelo real solo vamos a tener  $2^{n\_bits}$  valores que podremos representar. Dichos valores se pueden interpretar de forma distinta o son los mismos x valores para toda la red, tanto para activaciones como para pesos, como para los propios tensores.
- Cuanto más entreno peores son los valores que aparecen, con los pesos globales, con los pesos locales el asunto mejora. Ambos se estancan y no entrenan.
- Cuantizar data y target

### **Problemas:**

- No funciona la cuantización en fashionmnist

### **octava reunion (28-04-22)**

Se preguntó sobre como hacer la memoria.  
Se estimaron las fechas en las que terminar experimentación y redacción de memoria.  
experimentación: fecha límite 18 mayo  
memoria: fecha límite 15-22 junio

### **novena reunion (4-05-22)**

Reunión en la que se ha comentado la cuantización final que se tendrá que aplicar.  
Corregimos la actual cuantización: no hace falta cuantizar los tensores.

### **duda:**

- para la actualización de los pesos resto con el gradiente normal y luego cuantizo? o tengo que cuantizar lo que vaya a usar en el sgd.

-sirve la simulación que estamos realizando, es suficiente el trabajo que estamos realizando? no se quedará algo corto al quedarse tan simple la experimentación y habiendo realizado tan poco desarrollo software.

-¿Tenía que haber estudiado otras maneras de realizar cuantizaciones? ¿otras formas de entrenamiento?

### **decima reunion (11-05-22)**

Se habló sobre si el trabajo era suficiente, se volvió a explicar la cuantización que se está realizando (se cuantizan solo los pesos) se mostró que los resultados son muy malos y bajo esas condiciones los algoritmos son incapaces de aprender.

Me ha dicho que se debe de experimentar con pesos locales, y tengo que sacar la estadística de cómo varían las capas.

Para los experimentos fijar un parámetro y cambiar el resto.

Probar con otras arquitecturas para ver si se puede mitigar la precisión con estructuras más grandes.

#### **Problemas:**

-ver porque salen valores de loss tan elevados en algunas ejecuciones

### **undecima reunion (18-05-22)**

he comentado que los pesos varían mucho pues intentan compensarse

hacer pruebas 4-8 bits con la misma arquitectura , ver donde esta la frontera busco arquitectura más grande y hago la misma prueba, con una sola cuantizacion. ver los valores de los pesos cuando está en global para ver si se van a los extremos. juntar maximos y minimos capas y bias

tareas:

- pruebas con arquitecturas más grandes
- ver los valores de los pesos cuando hay arquitectura global

Dudas:

-he usado la guía y he traducido ciertas partes, tengo que tener miedo del copyright?

### **duodecima reunion (23-05-22)**

se ha mostrado resultados y se remarca que hay que hacer la redacción, falta probar arquitecturas más grandes

dudas:

es cuantización o cuantificación.

A Survey on Methods and Theories of Quantized Neural Networks solo está en arxiv lo menciono

Cuando hago mención de este paper

hace falta que cite en una parte del texto la fuente que he usado

si no se cita hay que meterlo en la bibliografía??? solo se mete lo que se cita

## **decimotercera reunion (1-06-22)**

preguntar que sueldo aproximado tiene un recién graduado y que sueldo tendría que poner por su trabajo.

preguntar por las cosas que tengo que especificar sobre los memristores

## **decimocuarta reunión (1-07-22)**

ver un poco de la metodología

## **decimocuarta reunión (1-07-22)**

revisión memoria

ver que resultados mostrar

expresar más o menos mis valoraciones

cosas a resolver:

-grafico pesos

-incluir paper de inviabilidad biológica: Backpropagation and the brain

se realizan nuevos experimentos