

## Proyecto 2 - (Un)Confused Terminator



Año 1984. Terminator es enviado por Skynet para matar a Sarah Connor y así evitar que desactive Skynet en un futuro próximo. Una vez que Terminator llega al pasado, busca en la guía telefónica el número y dirección de Sarah Connor, pero hay varias Sarah Connor, algunas escritas a mano y con otro font, y no tiene la inteligencia necesaria para entender escritura sin importar el font usado o quien lo escribió. Terminator está confundido, no sabe qué hacer. Pobre Terminator :/ ocupamos ayudarlo.

Uno de los primeros problemas resueltos usando redes neuronales ha sido la detección de texto en imágenes. Cerca de 1989, Yann Lecun fue uno de los primeros en resolver este problema cuando cursaba su PhD, al hacer uso de back-prop para entrenar una red de convolución.

En este proyecto, programaremos nuestra primera red neuronal desde cero, haciendo uso del conocimiento adquirido en el curso sobre modelos lineales, no-linealidades (funciones de activación), funciones de loss y retropropagación.

La red será entrenada con el set de datos MNIST, mismo usado por Lecun en sus primeros trabajos en redes neuronales. La idea del proyecto es que la red automáticamente reconozca números del 0 al 9, escritos a mano, sin importar la letra de la persona o quién la escribió.

### Red Neuronal - Perceptrón Multicapa (MLP)

Crearemos una red neuronal desde cero, con dos capas ocultas. La red claramente tiene 10 clases de salida, dado que el set de datos a usar tiene 10 clases del 0 al 9. Las capas oculta tendrá diferentes tamaños como parte de los experimentos. Tamaños de 128, 256, 512, 1024 son recomendados para pruebas.

Usaremos Cross Entropy para calcular la pérdida total al final del modelo. Las capas de neuronas utilizarán ReLU como función de activación.

Para poder optimizar la función de pérdida, recurriremos a una implementación de back-prop no general, es decir, no se espera que programen código generalista de grafos para cualquier función compuesta, sino solamente el código de gradientes para exclusivamente lo necesario para el MLP. Esto se traduce en que deben averiguar las ecuaciones de derivadas parciales para la neurona, ReLU, y cross entropy/softmax.

### Dropout

Deben programar la técnica de Dropout para evitar overfitting, desabilitando durante el entrenamiento un porcentaje aleatorio de las neuronas de las capas ocultas, para forzar a la red a aprender otros caminos.

## Lenguaje de Programación

Se debe programar en Python haciendo uso de numpy para las operaciones vectoriales. Para almacenar los pesos de la red se puede usar Pickle (una vez entrenada, pueden guardarlos en un diccionario directo a un archivo a través de pickle).

Se debe crear una interfaz, donde sea posible escoger un archivo de pesos de la red y poder poner imágenes a prueba, es decir, escoger una imagen de un número escrito a mano, con fondo blanco, y predecir su versión digital o clase. De nosotros depende que Terminator encuentre a Sarah Connor! :)

## Datos

Usaremos un set de datos llamado **NMIST** que puede ser encontrado aquí:

<http://yann.lecun.com/exdb/mnist/>

Este set de datos es de baja dimensionalidad lo que permite correr una red fácilmente en un CPU sin necesidad de GPU. El set de datos incluye 60k fotos para entrenamiento, todas de 28x28 pixels en blanco y negro.

## Entrenamiento

Deben tomar batches de imágenes de las 60k fotos de entrenamiento, calcular el forward de la red con cada batch, calcular el loss, y usando las derivadas parciales calcular cuánto modificar los pesos de la red.

## Experimentos

Se espera que hagan al menos 3 experimentos, y discutan sobre los resultados obtenidos:

1. Capas ocultas cada vez más grande: se espera que prueben con diversos tamaños de capa oculta y probar cuál obtiene mayor exactitud y minimiza mejor la pérdida.
2. Una capa oculta versus dos capas ocultas.
3. Un tercer experimento interesante que se les ocurra.

## Documento en Latex / PDF

Deben crear en latex (y entregar los fuentes de latex) un documento donde describen los métodos usados así como los experimentos. El documento debe tener la estructura de un paper normal (abstract, introducción, trabajo relacionado, metodología, experimentos, resultados, conclusiones).

Es importante mostrar la foto implícita en el W que corresponde a cada una de las 10 clases. Sería esperable que se vea una imagen “promedio” de cada una de las clases, representativa de los números del 0 al 9.

El documento debe seguir el template de la IEEE para publicaciones en ingeniería el cual pueden encontrar aquí:

<https://www.ieee.org/documents/ieee-latex-conference-template.zip>

## Evaluación

Tarea	Puntaje Máximo
Programación de MLP (Neuronas, Log Loss, ReLU, etc)	20
Programación del entrenamiento (derivadas, modificación de los pesos)	20
Dropout	5
Forma Paper (Intro, Metodología, Referencias, Citas, etc)	5
Experimento 1 (ejecución, discusión, gráficos)	15
Experimento 2 (ejecución, discusión, gráficos)	15
Experimento 3 (innovación, ejecución, discusión, gráficos)	20

