

**UNIVERSIDAD SAN IGNACIO DE LOYOLA**

**FACULTAD DE INGENIERÍA**

**Proyecto 1: OCR – Single Layer Perceptron**

Curso: Inteligencia Artificial

Bloque: FC-SMVINF07A1N

Instructor: Fernando Mauricio Jiménez Motte

Equipo:

Alzamora Deza, André Israel

Quiche Merizalde, Guillermo Josué

Quesada Chumbes, Alexis Anderson

García Alejandro, Mario Ricardo

Lima – Perú

2016

**Resumen Ejecutivo**

En la actualidad las diversas técnicas de inteligencia artificial permiten al programador diseñar un modelo el cual le permita insertar a una inteligencia artificial a diferentes campos en los cuales se trata de aprender y superar al experto humano, algunos de los campos pueden ser: Mercados financieros. Física, Transito, etc.

Pero todos los campos tienen en diferentes medidas algo en común lo que es el reconocimiento de patrones los cuales ha sido motivo de platica en muchas de las conferencias actuales relacionadas a la inteligencia artificial. Tratando de usar los ejemplos anteriores los sistemas inteligentes en el mercado financiero tendrían que aprender y reconocer patrones de cambio en diferentes aspectos para poder predecir los movimientos monetarios y valores futuros de cantidades enormes de dinero.

El presente proyecto está limitado lamentablemente al reconocimiento individual de siete (7) letras del alfabeto español, aunque se proyecta un escalamiento con un incremento exponencial de la capacidad de este proyecto, con la búsqueda de patrones de escritura escritos de palabras y textos completos, pero claramente esto no se podría lograr con la arquitectura básica que veremos en este perceptron.

**Introducción**

El enfoque primordial de este proyecto es el desarrollo de un proyecto el cual reciba diferentes diseños de letras obtenidas por una documente “.xlsx” o mejor dicho por una tabla en Microsoft Excel la cual forme una letra de una fuente de libre decisión por el usuario, se tiene que aclarar que por motivos didácticos y de tiempo se escaló a un diseño el cual contemple la letra con una capacidad máxima de 9x7 pixel.

Para el inicio del proyecto se le otorgo a nuestra red neuronal una serie de diferentes fuentes (diseño de letras) las cuales cada una comprendía 7 letras (A, B, C, D, E, F, G), esto será el inicio del Patter Recognition por parte de nuestro sistema inteligente.

La implementación de sistemas inteligente otorga la posibilidad de brindar a las computadoras algunas capacidades de los seres humanos, se podría decir que en un futuro no tan lejano las computadoras habrán podido embeber todas las capacidades humanas.

Luego de la entrega de data y el análisis realizado por la red neurona viene la fase de testing la cual me permite entregarle fuentes las cuales no fueron parte en el área de aprendizaje o entrenamiento. Finalmente, al entregarle nuevas fuentes la red será capaz de clasificarla, por eso estamos embebiendo 2 características humanas al computador “Pattern Recognition and Patter Clasification”.

**Modelamiento**

El presente diseño consiste en un modelo OCR (Optical Character Recognition) trabajando bajo una arquitectura Single Layer Neuronal Network, debido a la necesidad de procesar la data recibida solo una iteración para poder clasificarla.

Se tiene que mencionar que en esta versión del diseño de no se podrá apreciar un uso completo de una GUI para que el usuario pueda comunicarse de manera apropiada con la red neuronal.

W637

W11

X63

Y7

Y2

Y1

X3

X2

X1

Ilustración número 1 – Arquitectura del OCR

**Análisis**

Por el punto de vista matemático el algoritmo de perceptron hace un uso adecuado de las diversas herramientas matemáticas para poder realizar el aprendizaje y memorización de datos por parte de la red neuronal lo cual permite posteriormente la clasificación de caracteres.

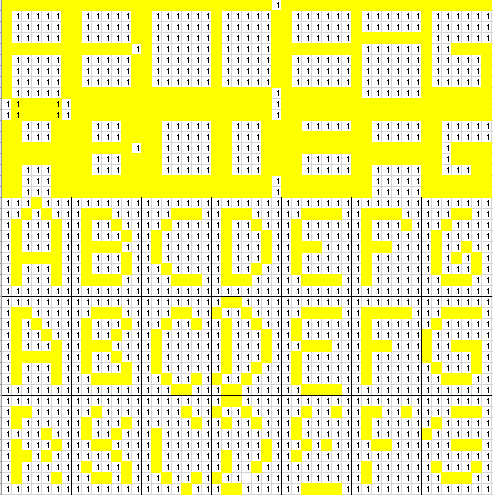
Se decidió ir por una programación en binario, pero con una investigación posterior se comprobó que una codificación en bipolar sería mucho más conveniente al mejorar la tolerancia a fallas del sistema sin mencionar que otorgaría una mejor velocidad al OCR para el aprendizaje.

Algunos de los datos que se ven recurrentes en el algoritmo son: Peso de las conexiones (Matriz), BIAS (Vector), UMBRAL (Vector).

Ahora con respecto al algoritmo en si se tiene que mencionar que el perceptron necesita el uso constante de matrices y vectores para sus cálculos la matriz principal en nuestra opinión y la cual nos permitirá a nosotros realizar la clasificación de las letras es la matriz de pesos.

La matriz de pesos originalmente esta inicializada en ceros como una matriz de 63\*7 donde la cantidad de filas son la cantidad de entradas que recibe la neurona y las columnas son la cantidad de salidas que podría llegar a tener. Esta matriz se actualizará constantemente cada vez que se le entrega a la red una fuente la cual no haya sido evaluada y las compara hallando la salida de esa fuente con el target u objetivo a mapear la cual también será una matriz en bipolar la cual que tendrá con dimensiones 7x7 y será una matriz de identidad.

Para poder realizar una labor más óptima en el aprendizaje se decidió por brindarle a la red un Bach de datos la cual lleva todas las fuentes y letras correspondientes.



Luego de que la red neuronal recorriera todo el Bach nos retorna una matriz de pesos completamente diferente la cual variara por cada letra y fuente que se le otorgue.

Ilustración número 2 – Bach de datos

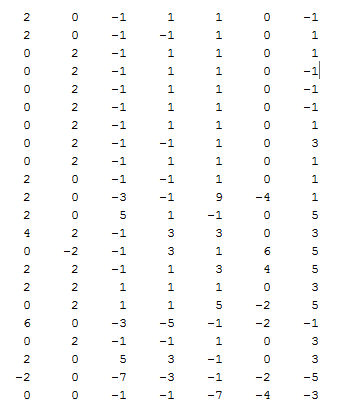


Ilustración número 3 – Sección de Nueva Matriz de Pesos

**Programación**

Se buscó separar la data del código y por eso se optó por colocar toda la data en un paquete los cuales se encuentran separados.

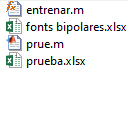
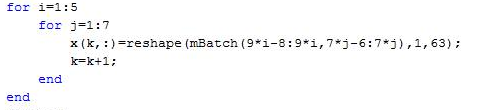
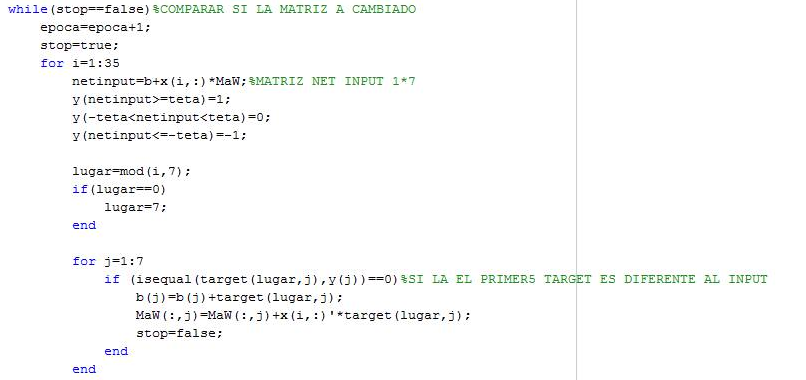


Ilustración número 4 – Paquetes de datos

Se decidió primordialmente recorrer el Bach y guardarlo en una nueva matriz la cual será ajustada para tener las dimensiones 35x63, lo que significa que cada fila será una letra de una fuente.



Luego se inicia un bucle en el cual la condición principal es que si los pesos no cambian no se detenga las iteraciones.

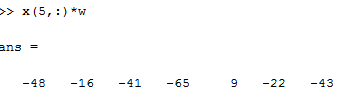


Finalmente, el método de testeo se realizó de la siguiente manera:

1. w=entrenar('fonts bipolares.xlsx')
2. x=prue()
3. x(5,:)\*w

Esta última línea de consulta nos permite ingresar la 5 letra de las fuentes de prueba las cuales son enviadas con el método prue().

Se verá que el resultado dará:



Y se podrá apreciar que el 5 número del vector es positivo y todos los demás son negativos por lo tanto se puede decir con seguridad que la quinta letra de las fuentes de prueba es una F.

Conclusiones

* Para concluir se puede decir que el sistema puede reconocer las letras ingresadas considerando un factor de ruido incluido en algunas de las fuentes otorgadas,
* También se tiene que decir que se puede escalar a un OCR el cual tenga una mejor capacidad de reconocimiento, al ser este un perceptron simple todavía no puede aprender de los errores cometidos y el margen de error dado la cantidad de fuentes y la cantidad de pixel en cada letra no son muy altos.

**References**

* Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms and applications*. Prentice Hall.
* Monge, M. A. (2008). *Diseño de una Arquitectura para una Red Neuronal Artificial Perceptron Multicapa sobre Un FPGA Aplicada al Reconocimiento de Caracteres*. Pontificia Universidad Católica del Perú: Lima (Perú). Recuperado de https://drive.google.com/folderview?id=0B1t\_MpVwOiCIUk1IOWx0dFh1dDQ&usp=sharing\_eid&ts=57197f04.
* Ordoñez, J. P. (2001). *Reconocimiento Óptico de Caracteres (OCR) con Redes Neuronales Estado del Arte.* Recuperado de https://drive.google.com/folderview?id=0B1t\_MpVwOiCIUk1IOWx0dFh1dDQ&usp=sharing\_eid&ts=57197f04.