

IA-Project

June 17, 2019

1 Proyecto Inteligencia Artificial

1.1 Integrantes: Alexandra Guamán - Diego Cabrera

1.1.1 Redes Neuronales

1.1.2 Definition

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Neural networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification and regression.

1.1.3 Pygame

Pygame is a python wrapper for SDL, written by Pete Shinnars. What this means is that, using pygame, you can write games or other multimedia applications in Python that will run unaltered on any of SDL's supported platforms (Windows, Unix, Mac, BeOS and others).

1.1.4 Example of Application

This project consists of training a neural network, with a corpus 946 digit samples, obtained from several images. The purpose of the project is to predict the digit that is drawn on a grid.

```
In [1]: import pygame
import tkinter
from tkinter import ttk
from Utilities import Utilities
import numpy as np
from Red import Red

datos=[]
n=[]
```

v=0

```
class DigitsVisualizer():

    def __init__(self, path = './corpus/digits-database.data'):
        super().__init__()
        self.path = path
        self.utilities = Utilities()
        self.indices = self.utilities.generate_indices()
        self.WIDTH = 800
        self.HEIGHT = 600
        self.red=Red()
        self.initialize()

    def initialize(self):
        cont1=21
        cont2=53
        for i in range(1):
            n=self.utilities.get_digit(cont1,cont2)
            n=np.reshape(n,-1)
            print(n)
            m=self.utilities.get_digit(cont2,cont2+1)
            v=m[0]
            n=np.append(n,v)
            cont1+=1
            cont2+=1
            cont1=cont1+32
            cont2=cont1+32
            #print(n[1024])
            datos.append(n)
        self.generate_grid()

    def generate_grid(self):
        pygame.init()
        NEGRO = (0, 0, 0)
        BLANCO = (255, 255, 255)
        VERDE = (0,255,0)
        GRIS=(200,200,200)
        DIMENSION_VENTANA = [800, 700]
        pantalla = pygame.display.set_mode(DIMENSION_VENTANA)
        LARGO = round(800/34)
        ALTO = round(600/34)
        MARGEN = 1
        self.grid = []
        for fila in range(32):
            self.grid.append([])
            for columna in range(32):
```

```

        self.grid[fila].append(0)

self.grid[1][31] = 0
hecho = False
pos=[]
x=(800/2)-75
reloj = pygame.time.Clock()
pantalla.fill(NEGRO)
pygame.font.init()
pygame.draw.rect(pantalla,(100,100,100),(x-5,615,160,60))
pygame.draw.rect(pantalla,GRIS,(x,620,150,50))
basicfont = pygame.font.SysFont(None, 26)
text = basicfont.render('ENVIAR DATOS', True, NEGRO, (200, 200, 200))
textrect = text.get_rect()
textrect.centerx = 400
textrect.centery = 640
pantalla.blit(text, textrect)
pygame.display.flip()

while not hecho:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            hecho = True
        elif evento.type == pygame.MOUSEBUTTONDOWN:
            pos = pygame.mouse.get_pos()
            columna = pos[0] // (LARGO + MARGEN)
            fila = pos[1] // (ALTO + MARGEN)
            if pos[1] < 593:
                self.grid[fila][columna] = 1
            if x+150 > pos[0] > x and 620+50 > pos[1] > 620:
                self.boton()

    for fila in range(32):
        for columna in range(32):
            color = BLANCO
            if self.grid[fila][columna] == 1:
                color = NEGRO
            pygame.draw.rect(pantalla,
                             color,
                             [(MARGEN+LARGO) * columna + MARGEN,
                              (MARGEN+ALTO) * fila + MARGEN,
                              LARGO,
                              ALTO])

    reloj.tick(100)
    pygame.display.flip()
pygame.quit()

```

```
def boton(self):
    self.red.inicio(datos)
    print(self.grid)

if __name__=="__main__":
    dv = DigitsVisualizer()
```

pygame 1.9.6

Hello from the pygame community. <https://www.pygame.org/contribute.html>

Inicio Red Neuronal

[0 0 0 ... 0 0 0]