

Proyecto 1: Parte B

Curso: Inteligencia Artificial

4 de mayo de 2017

Instituto Tecnológico de Costa Rica
Sede Interuniversitaria de Alajuela
Prof. Fabián Fallas Moya

1. Introducción

Esta parte del proyecto está orientada a hacer todo un algoritmo de Redes Neuronales para el problema de clasificación de imágenes. Su tarea será descargar el dataset, dividirlo en training-set, validation-set y test-set. Para finalmente implementar el algoritmo de Feedforward Backpropagation y ver un prototipo corriendo de manera adecuada.

1.1. El dataset

El dataset lo deben descargar de la dirección: <http://yann.lecun.com/exdb/mnist/> . Una vez descargado deben visualizar las imágenes y llevar a cabo todo el desarrollo del algoritmo basados en ese dataset. Es su tarea investigar todo lo necesario con respecto a ese dataset.

1.2. Preprocesamiento

Además no olvide que estamos trabajando con escala de grises, verifique que esto se cumpla en el dataset, de lo contrario debe hacer esta conversión. También verifique que el tamaño de todas las imágenes es el mismo, ya que necesitamos que el input sea igual para todas las imágenes.

2. Implementación

Deben de hacer la implementación de un algoritmo de Redes Neuronales Feedforward Backpropagation. Deben aplicar todo lo visto en clase y durante la revisión (cada grupo tendrá cerca de 20 minutos para la revisión) deben defender y explicar su algoritmo. Además durante la revisión se solicitará hacer cambios (por ejemplo de parámetros, pesos, variables) para ver el comportamiento del algoritmo.

2.1. Lenguaje de programación

Deben utilizar Python para el desarrollo de este proyecto. Pueden utilizar cualquier biblioteca para el manejo de álgebra lineal (sobretudo para multiplicación de matrices) pero todo el algoritmo debe estar programado (así como se hizo para Octave). No se permite el uso de bibliotecas que ya traigan implementadas las Redes Neuronales.

2.2. Prototipo

2.2.1. Proporciones

Deben crear un prototipo y su correspondiente interfaz, en la cual podamos cargar el dataset y podamos definir las proporciones del training set, validation set y test set. Por lo tanto esta proporción la podrá definir el usuario. De esta manera podremos hacer pruebas exhaustivas manejando diferentes proporciones. Por ejemplo: si el usuario define la proporción de 60-20-20, entonces nuestro training va a realizarse con el 60 % de los datos, el validation con el 20 % y el test con el restante 20 %; otra proporción interesante podría ser 30-30-40, en donde nuestro training va a realizarse con el 30 % de los datos, el validation con el 30 % y el test con el restante 40 %; y así sucesivamente.

2.2.2. Arquitectura de la Red Neuronal

Recuerden que la arquitectura la define el programador, pero este no será el caso. La arquitectura será definida por el usuario. La cantidad de neuronas de entrada no va a variar, dado que el input será siempre el mismo; al igual que la cantidad de neuronas de salida. Lo que el usuario sí podrá definir es la cantidad de capas ocultas (hidden layers) y la cantidad de unidades (neuronas) en cada capa (hidden units). De esta manera también podremos ir comparando resultados con diferentes arquitecturas. Lo mismo podremos hacer con el aspecto de el parámetro de regularización.

2.2.3. Learning Curves

También tendremos la opción de correr nuestra Red Neuronal de manera incremental, esto es: si el dataset es de 5000 imágenes, podremos decirle que corra cada 50 imágenes. Por ejemplo: que corra con 50 imágenes, con 100, con 150, con 200, etc. Y los resultados se van guardando (los errores), esto con el objetivo de poder graficar las curvas de aprendizaje vistas en clase y de esta manera poder concluir si nuestra red padece de *high bias* o *high variance*. Estos datos se almacenarán en la BD que se explica a continuación.

2.2.4. Base de Datos de Resultados

Cuando elegimos una proporción y una arquitectura, los resultados los vamos a poder almacenar en una Base de Datos. Puede ser una Base de Datos Transaccional, el requisito es que se encuentre en la nube (Internet). Por lo tanto va a hacer una ventana en la interfaz para poder conectarnos con esa BD y probar la conexión, como el asunto de Internet falla en la Sede, debemos tener la capacidad de conectarnos a una BD local en caso de que falle el acceso a Internet.

2.2.5. Reportes de Diagnóstico

Con los datos que se van almacenando en la BD, vamos a poder hacer reportes para visualizar el comportamiento del Bias y el Variance (así como lo hicimos en la última clase). Cuando vayan a hacer la presentación del proyecto, lleven muchos datos para poder visualizar estos reportes.

Estos reportes estarán disponibles en línea para que sean accesibles desde cualquier parte del planeta. Para este propósito, desarrollarán un pequeño módulo web, hecho en **PHP** y lo publicarán en un servidor en la nube (Internet) para poder acceder a esos reportes. Hay muchos disponibles, por ejemplo www.hostinger.es (nos da una opción para una BD MySQL y un servidor Apache para correr PHP), ahí pueden tener su BD y web site en un solo lugar. Hay otros «*hosting services*», pueden utilizar el que gusten o consultar por otros más.

2.2.6. Github

Para todo el proyecto, deben utilizar un administrador de versiones para el código. En este caso utilizarán Github para que el código fuente esté en la nube. Lo bueno de Github es que pueden trabajar desde sus casas y todos en conjunto van creando el proyecto en la nube; cada integrante del grupo va aportando al código fuente y van haciendo «commits», los cuales serán revisados y la idea es que cada integrante del grupo tenga un número similar de commits cuando sea la entrega final del proyecto (esto dará la seguridad de que todos trabajaron por igual).

3. Revisión

Se tendrán 2 puntos de control (revisiones antes de la entrega final). El primero será el día 11 de mayo, en el cual deben mostrar avances en todas las etapas del proyecto. El siguiente punto de control será el día 18 de mayo.

La revisión final del proyecto será el 25 de mayo. Para esta revisión se asignarán citas de revisión.

3.1. Sanciones

Cualquier indicio (y posterior comprobación) de fraude académico será sancionado con las mayores consecuencias que indique el reglamento estudiantil del ITCR.

Todos los integrantes del grupo deben trabajar, si se comprueba que alguno no colaboró del todo; todos los integrantes del grupo serán sancionados.

3.2. Tabla de evaluación

En el primer punto de control se les entregará la tabla de evaluación, con cada rubro.