



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Redes Neuronales

Trabajo práctico 1

Parser

Resumen

Entrenamiento de Redes Neuronales

Integrante	LU	Correo electrónico
Negri, Franco	893/13	franconegri2004@hotmail.com
?, ?	?/?	?@?.com

Palabras claves:

TP

Índice

1. Introduccion	2
2. Desarrollo	2
2.1. Preprocesamiento De Los Datos	2
2.2. Implementacion Del algoritmo	2
2.3. Experimentación sobre datos de Diagnostico de Cancer	3
2.3.1. Convergencia del algoritmo	3
2.3.2. Performance Vs Learning Rate	3
2.3.3. Performance Vs Cantidad De iteraciones	4
2.4. Performance Regreción Lineal Calor	4
3. Conclusiones	5

1. Introduccion

En este trabajo se intentara utilizar tecnicas de redes neuronales profundas con la intención de resolver dos problemas diferentes:

En el primer problema, contaremos con características de imágenes de células y por otro lado con el diagnóstico final de si resultaron ser cáncer o no cáncer.

Para el segundo problema intentaremos predecir la carga de calefacción y la carga de refrigeración de un edificio a partir de ocho características dadas de un edificio.

Para ambos problemas utilizaremos una red neuronal con retro-propagación del error a la que se entrenara con instancias de estos datos para ver si son generalizables. Además se intentaran buscar los parámetros adecuados (cantidad de neuronas, cantidad de capas, learning rate, momentum) para mejorar la performance de la misma.

2. Desarrollo

2.1. Preprocesamiento De Los Datos

Con el objetivo de darle mayor estabilidad matemática a los datos y quitar algo del ruido que puedan tener, realizamos un preprocesamiento de los mismos. El mismo consistió en quitar los outliers dentro del set de datos y luego normalizarlos con media 0 y varianza 1.

2.2. Implementacion Del algoritmo

El algoritmo que implementamos consistió en una red neuronal profunda con retropropagación del error. Este, descrito de manera informal consiste en introducir una instancia del problema, calcular la salida, compararla con la salida esperada y retropropagar el error, con el objetivo de modificar las matrices de pesos y minimizar la diferencia. El sistema implementado se dice ser *online* ya que los pesos de las matrices son ajustados luego de retropropagar el error de cada instancia en vez de hacerlo al final de la época.

A fin de tener una medida de como va aprendiéndola red neuronal, luego de presentarle a la red todas las instancias una vez, calculamos la norma de la diferencia entre las respuestas obtenidas y las esperadas. Consideraremos esta nuestra "norma del error" la tomaremos como una métrica adecuada para saber cuan buenos resultados devuelve nuestra red.

Como optimización adicional se agregó un término de momentum. La idea tras esto consiste en darle a cada peso de la matriz una "inercia" que le permita continuar avanzando en la dirección en la que avanzó en la iteración anterior. El objetivo consiste en disminuir las oscilaciones con cada pequeño cambio en la matriz.

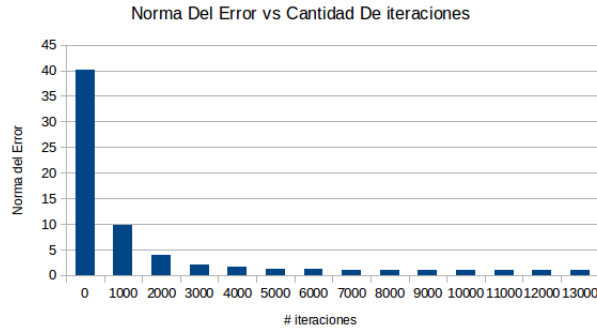
2.3. Experimentación sobre datos de Diagnostico de Cancer

2.3.1. Convergencia del algoritmo

Para este problema utilizaremos una neurona en la última capa con función de activación logística. Para ser consistentes con esto, las salidas esperadas adoptarán los valores 1 si resulto ser cáncer maligno, y 0 si resulto ser benigno. La experimentación consistirá en buscar los parámetros óptimos que nos den la mayor tasa de aciertos para nuestro data set.

Como primera instancia comprobaremos que la red converge efectivamente a la solución esperada. Para eso tomamos un learning rate de 0,01 y con 1300 épocas graficamos la norma del error

cada 1000 iteraciones:



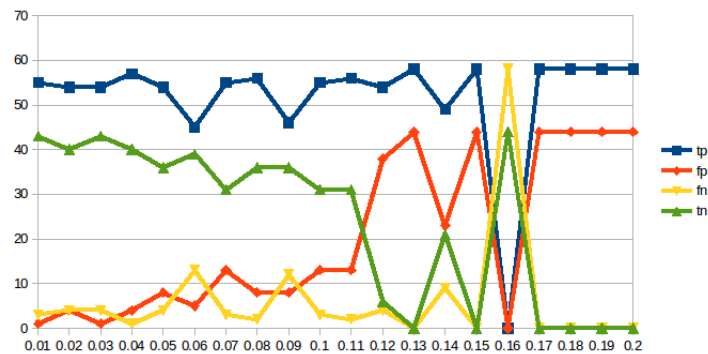
El grafico muestra que, efectivamente, nuestro algoritmo minimiza la diferencia entre las soluciones obtenidas y las deseadas, minimizando así tambien la norma del error.

2.3.2. Performance Vs Learning Rate

Para evitar el sobreajuste las experimentaciones todas las experimentaciones desde este punto se realizarán con la siguiente metodología. Se dividirá el set de datos de entrenamiento provisto por la catedra en dos sets de datos distintos. Uno se utilizará para entrenar la red, mientras que el otro se medirán que tan buenos fueron los resultados obtenidos. Con esto se espera reducir el overfitting que la red pueda generar y comprobar de manera mas acertada que tan buenos resultados dará la red sobre datos reales del problema.

Ademas, para visualizar los resultados mas claramente utilizaremos una matriz de confución, que nos permitirá dicernir entre falsos positivos (fp), falsos negativos(fn) y instancias clasificadas correctamente (tp y tn).

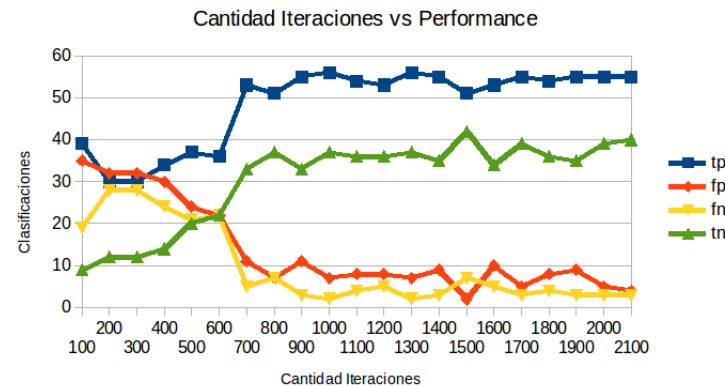
En esta sección queremos experimentar es como se comporta el algoritmo al variar el learning rate. Para ello, dejamos constantes las cantidad de epocas de entrenamiento en 10000 y variamos el learning desde 0,01 hasta 0,2 aumentando de a 0,01



El grafico muestra varios resultados interesantes. En el rango $[0,01,0,11]$ puede verse que la red arroja buenos resultados, obteniendo en general un 90 % de clasificaciones correctas. Pasado ese rango puede verse como los falsos positivos y los falsos negativos sufren un incremento muy significativo, llegando al final a casi un 50 % de clasificaciones incorrectas. Consideramos que para estos casos la red neuronal divergió, posiblemente debido a que multiplicar el gradiente por un valor muy grande lleva a "pasarnos" del minimo y por lo tanto reduciendo la precisión del algoritmo.

2.3.3. Performance Vs Cantidad De iteraciones

Como siguiente paso buscaremos analizar el comportamiento de la red para distintas cantidades de iteraciones, viendo si existe algun cambio significativo en este sentido. Para ello dejamos fijo el learning rate fijo en 0,01 ya que consideramos que este nos da los resultados mas aceptables y variamos la cantidad de iteraciones.



Para una cantidad menor a 600 iteraciones puede observarse que la red neuronal tiene una performance mala obteniendo un porcentaje de falsos negativos y falsos positivos cercano al 50 %. Ya con 700 iteraciones en adelante la performance del algoritmo mejora drasticamente, rondando el porcentaje de falsos negativos y falsos positivos al rededor de 10 % y 20 %. Para 2000 y 2100 iteraciones obtenemos un 7 % de falsos negativos.

2.4. Performance Regreción Lineal Calor

Por cuestiones de tiempo este ejercicio no fue probado de manera adecuada

3. Conclusiones

El tema esta copado, pero no nos dio el tiempo. =(