



Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

División de Electrónica y Computación

Departamento de Ciencias Computacionales

Ingeniería en Computación

Inteligencia Artificial II

Profesora: Arana Daniel, Nancy Guadalupe

I7040 – D01

Martes y Jueves 11:00 – 12:55

Actividad 03: Reporte de práctica del Adaline aplicado

Flores Camarena, Luis Manuel
214519661

César Arley Ojeda Escobar
216306568

Fecha: 26/09/2019

Predictor de divorcios

Utilizando el modelo del Adaline, se busco crear un predictor de divorcios, en base a un clasificador de decisiones y sentimientos, los cuales son representados en una escala de 0 a 4, según la afinidad de las personas hacia ciertas cuestiones o situaciones.

Este trabajo esta basado en la investigación del Dr. Yöntem y su equipo de la Universidad Nevşehir, en Turquía, sobre como se manifiestan los divorcios en su país. Utilizó un grupo de estudio de 170 participantes entre hombres y mujeres de diferentes edades, y les aplico una prueba de 54 preguntas las cuales pueden caracterizar los patrones que corresponden a las relaciones de pareja. Para más información sobre los tipos de personas que pertenecen al grupo de estudio, y las preguntas qué se realizaron, se recomienda visitar el articulo **DIVORCE PREDICTION USING CORRELATION BASED FEATURE SELECTION AND ARTIFICIAL NEURAL NETWORKS**.

Para la investigación original se aplicaron 54 preguntas al grupo de estudio, creando así un data set de entrenamiento con 170 registros y 54 atributos, incluyendo un ultimo atributo que define si la persona esta actualmente casada o se encuentra divorciada, sin embargo, y ajustándonos a la investigación original, se utilizaron solamente 6 atributos de los 54 originales, pues el documento indica que se realizó una clasificación con los atributos más significativos dejándonos solo con los 6 que se utilizaron en esta práctica, además, esto nos facilita el trabajo al momento de probar la neurona.

Finalmente tenemos un data set de 6 atributos o dimensiones para efectos de la práctica, más la clase a la que pertenecen cada uno de los registros y 170 registros de los cuales 84 son divorciados y 86 están casados. Este data set fungirá como nuestro set de entrenamiento.

Código

Se implemento una aplicación de escritorio con el lenguaje Python en su versión 3.7 y el módulo PyQt5 para el diseño de la interfaz.

Algoritmo del Adaline en Python

Inicializamos la clase Adaline

```
def __init__(self, entradas, clases, ratio_aprendizaje, epocas_maximas, pesos, theta, error_minimo):
    self.ERROR_MAX = 20

    self.entradas = np.array(entradas)
    self.salidas_deseadas = np.array(clases)
    self.ratio_aprendizaje = ratio_aprendizaje
    self.epocas_maximas = epocas_maximas
    self.pesos = np.array(pesos)
    self.theta = theta
    self.error_minimo = error_minimo
    self.erroses = []
```

Entrenamiento de la neurona

```
def entrenamiento(self, ventana):
    progreso = ventana.progreso
    activacion = self.f_activacion()

    resultados = self.salidas_deseadas - activacion
    gradiente = activacion * (1-activacion)
    # error = (np.sum(np.array(resultados**2)))/len(self.pesos)
    error = (np.sum(np.array(resultados**2)))/len(self.pesos)
    self.epoca_actual = 0
    carga = 100 / self.epocas_maximas

    if error < self.error_minimo:
        self.erroses.append(np.sum(resultados))
        self.cargarErrorGrafica(ventana)
        progreso.setValue(100)
        return True

    while self.epoca_actual < self.epocas_maximas and error > self.error_minimo:
        self.epoca_actual += 1
        i = 0
        progreso.setValue(self.epoca_actual * carga)
```

```
self.theta = self.theta + (self.ratio_aprendizaje * resultados[i] * gradiente[i] * 1)
for i, entrada in enumerate(self.entradas):
    self.pesos = self.pesos + (self.ratio_aprendizaje * resultados[i] * gradiente[i] * entrada)
    if error < self.error_minimo:
        progreso.setValue(100)
        return True

activacion = self.f_activacion()
resultados = self.salidas_deseadas - activacion
gradiente = activacion * (1-activacion)
error = (np.sum(np.array(resultados**2)))/len(self.pesos)
selferrores.append(error)
self.cargarErrorGrafica(ventana)
if error == 0 or error < self.error_minimo:
    progreso.setValue(100)
    return True

return False
```

Función de activación

```
def f_activacion(self):
    resultados = []
    net = np.dot(self.entradas, self.pesos) + self.theta
    # net = np.dot(self.entradas, self.pesos)
    for resultado in net:
        resultados.append(1 / (1 + math.exp(resultado*-1)))
    return np.array(resultados)
```

Interfaz de la aplicación

La interfaz se separa en dos partes, la parte de la derecha sirve para entrenar a la neurona, mientras que la parte de la izquierda sirve para usar la neurona ya entrenada, sin embargo esta parte solo se activara cuando la neurona haya convergido correctamente.

Adaline aplicado

Preguntas de prueba

1. Sé que podemos ignorar nuestras diferencias, incluso si a veces las cosas se ponen difíciles.

0

2. No compartimos tiempo en casa como compañeros

0

3. Creo que un día en el futuro, cuando mire hacia atrás, me dare cuenta que mi esposa y yo estamos en armonía uno con el otro.

0

4. Mi esposa y yo tenemos ideas similares sobre cómo debería ser el matrimonio.

0

5. Conozco las preocupaciones básicas de mi esposa.

0

6. Podemos empezar una pelea antes de saber lo que está pasando.

0

Resultados

Resultado:

Predecir

Controles

Ratio de aprendizaje

0.50000

Épocas máximas

1000

Error mínimo

0.01000

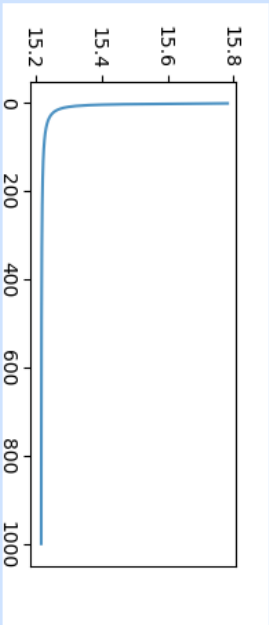
Entrenar

Información extra

El error mínimo alcanzado es: 15.2144

No convergió

Gráfica del error



Data set de entrenamiento

La siguiente tabla muestra una pequeña sección del data set completo. Se muestran los primeros 10 registros de personas divorciadas con la clase '1', y los últimos 10 registros de personas casadas con la clase '0'.

Atr1	Atr2	Atr3	Atr4	Atr5	Atr6	Class
2	0	1	0	0	3	1
4	0	4	4	1	4	1
2	3	3	3	2	3	1
2	3	4	3	1	4	1
2	1	0	1	2	1	1
0	2	0	1	2	2	1
3	3	2	3	3	3	1
1	1	4	3	3	4	1
2	4	3	3	3	2	1
1	2	3	2	1	2	1
...
0	0	0	0	0	1	0
0	0	0	0	0	1	0
1	1	1	1	0	0	0
0	2	0	0	0	0	0
0	0	1	0	1	0	0
1	2	0	0	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
1	0	0	0	1	2	0
0	0	0	0	0	1	0

Conclusión

Es interesante poder aplicar nuestro conocimiento en situaciones reales de la vida y ver como los algoritmos que en un momento pueden parecer muy sencillos, puedan realmente ser aplicados en sistemas que puedan incluso hacer predicciones y como con un data set de entrenamiento de unos cuantos cientos de registros se puede tener una efectividad de hasta el 98.82% [Yöntem, 2018].

Sin embargo y a partir de varios entrenamientos con diferentes ratios de aprendizaje, épocas máximas y errores mínimos podemos concluir que nuestro set de entrenamiento no es linealmente separable, pues nunca se logró llegar a la convergencia de la neurona Adaline, por lo que habrá que esperar a implementar una red neuronal que nos permita clasificar problemas que quedan fuera del campo de los linealmente separables.

Referencias

Yöntem, M , Adem, K , İlhan, T , Kılıçarslan, S. (2018). *DIVORCE PREDICTION USING CORRELATION BASED FEATURE SELECTION AND ARTIFICIAL NEURAL NETWORKS*. Nevşehir Hacı Bektaş Veli University SBE Dergisi, 9 (1), 259-273. Recuperado el 18 de septiembre del 2019 desde [\[https://dergipark.org.tr/en/download/article-file/748448\]](https://dergipark.org.tr/en/download/article-file/748448)

Dr. Mustafa Kemal Yöntem, D. K. (2018). *Machine Learnign Repository*. Obtenido de Divorce Predictors data set: Center for Machine Learning and Intelligent Systems Recuperado el 18 de septiembre del 2019 desde [\[https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set\]](https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set)