Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Técnicas de inteligencia artificial

Reporte: Ejemplos del perceptrón multicapa y redes de propagación hacia adelante

Docente: Abraham Sánchez López

Alumno	Matrícula
Taisen Romero Bañuelos	202055209

Perceptrón multicapa y redes de propagación hacia adelante

Ciertamente el perceptrón multicapa tiene similitud con la propagación hacia adelante, aunque eso no fue lo único que me pareció interesante. Usando el sentido común supuse que el modelo tenía dificultades distinguiendo entre versicolor y virginica porque las características entre ambos tipos son similares, tipo que el largo de sus pétalos sean muy similares (de hecho, creo que en alguna práctica de minería se hizo una observación así). Pero, ¿esto se refleja en los pesos obtenidos?, es decir, si sólo nos basamos en los pesos que distinguen la clasificación de una categoría y otra, podríamos suponer que esta dificultad para distinguirlos se debe a que tienen pesos similares (análogo a las características con valores similares). Para revisar eso hice una tabla comparativa de los pesos hacia las salidas.

Consideremos que: setosa (o1), versicolor (o2), virginica (o3), sesgo (b) y neuronas ocultas (h_n).

```
> pesos_nn2
  Conexion setosa versicolor virginica
1    sesgo  19.92      12.96    -31.85
2       h1 -52.35       0.67     51.84
3       h2  25.18      -8.92    -17.48
```

```
> pesos_nn6
  Conexion setosa versicolor virginica
1    sesgo   6.05       0.25     -6.21
2       h1  28.12      -3.76    -24.85
3       h2  -1.50       4.62     -3.16
4       h3 -33.99       2.77     31.54
5       h4  24.82      12.57    -38.13
```

Si bien a primera vista no hay similitudes muy notables podemos extraer información interesante de esas tablas comparativas. Primero, nn.6 presenta valores más equilibrados y menos contrastantes entre versicolor y virginica, lo que podría generar activaciones similares. Y con nn.2 vemos pesos más diferenciados, especialmente en la neurona h1, donde versicolor y virginica tienen pesos casi opuestos. Al parecer la forma en que se distribuyen los pesos (y no sólo su valor) también afecta en la clasificación. Es posible que el modelo nn.6 generaliza mejor los datos por el número de entradas que tiene.

De hecho, la modificación de los pesos influye directamente en los resultados de salida, y esto se ve reflejado en la práctica de forward propagation. Yo cambié los valores de la siguiente forma:

```
> # ----> Modificando los valores de los pesos
> # W0,0: 0.2
> # W0,1: 0.4
> # W0,2: -0.3
> # W0,3: -0.6
> # W1,0: 0.007
> # W1,1: -0.05
```

Las modificaciones dieron como resultado un mejor error.

```
> print(N2.1.error)
           [,1]
[1,] 0.1227763
>
> print(N2.0.error)
          [,1]
[1,] 0.124325
```

Los cambios en los pesos los hice de forma aleatoria pero siguiendo una tendencia a hacer pesos más pequeños. Si bien esto no quiere decir que hubo un proceso de aprendizaje de parte del modelo, si nos hace darnos cuenta en qué dirección hay que modificar los pesos para disminuir el gradiente. Esta combinación afortunada nos hace notar que hay un margen de mejora y también nos ayuda a corroborar lo del perceptrón, los pesos pueden tener un gran impacto al permitir (o no) que el modelo aprenda más rápido. Quizá podríamos modificar los parámetros de las redes guiándonos por el peso para no hacer combinaciones aleatorias (como lo hice en forward propagation).